Scalable and Reliable Searching in Unstructured Peer-to-Peer Systems

by

Efstratios Ioannidis

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Abstract

Scalable and Reliable Searching
in Unstructured Peer-to-Peer Systems

Efstratios Ioannidis
Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto
2009

The subject of this thesis is searching in unstructured peer-to-peer systems. Such systems have been used for a variety of different applications, including file-sharing, content distribution and video streaming. These applications have been very popular; they contribute to a large percentage of today's Internet traffic and their users typically number in the millions.

By searching, we refer to the process of locating content stored by peers. Searching in unstructured peer-to-peer systems poses a challenge because of high churn: both the topology and the content stored by peers can change quickly as peers arrive and depart, while the network formed under this churn process can be arbitrary at any point in time. As a result, a search mechanism must operate without any a priori assumptions on this dynamic topology.

Ideally, a search mechanism should be *scalable*: as, typically, peers have limited bandwidth, the traffic generated by queries should not grow significantly as the peer population increases. Moreover, a search mechanism should also be *reliable*: if certain content is in the system, searching should locate it with reasonable guarantees. These two goals can be conflicting, as generating more queries increases a mechanism's reliability but decreases its scalability. Hence, a fundamental question regarding searching in unstructured systems is whether a mechanism can exhibit both properties, despite the network's dynamic and arbitrary nature.

In this thesis, we show this is indeed the case, by proposing a novel mechanism that is both scalable and reliable. This is shown under a mathematical model that captures the evolution of both network and content in an unstructured system, but is also verified through simulations.

To the best of our knowledge, this is the first provably scalable and reliable search mechanism for unstructured peer-to-peer systems.

In addition to the above problem, we also consider a *hybrid* peer-to-peer system, in which the peer-to-peer network co-exists with a central server. The purpose of this hybrid architecture is to reduce the server's traffic by delegating part of it to its clients —*i.e.*, the peers: a peer wishing to retrieve certain content first propagates a query over the peer-to-peer network, and downloads the content from the server only if the query fails. This hybrid architecture can be used to partially decentralize a content distribution server, a search engine, an online encyclopedia, *etc.*

The trade-off between scalability and reliability translates, in the hybrid case, to a trade-off between the peer and the server traffic loads. We propose a search mechanism under which both loads remain bounded as the peer population grows. This is surprising, and has an important implication: one can construct hybrid peer-to-peer systems that can handle traffic generated by a large (unbounded) peer population, even when both the server and peer bandwidth capacities are limited. Again, this is proved under a model capturing the hybrid system's dynamic nature and verified through simulations. To the best of our knowledge, our work is the first to show that hybrid systems with such properties exist.

In the memory of my father.

Acknowledgements

I am deeply grateful to my advisor, Peter Marbach, for his invaluable guidance and support. Working with him has been an inspiration; everything I can claim to know about research I have learned from him. I can only hope that, in the years to come, I will carry with me some of his infectious enthusiasm, his insight, and his insatiable desire to find, and solve, those problems that challenge us the most.

I am also indebted to the members of my supervisory committee for their insightful and thorough criticism of this thesis. Derek Corneil's careful reviewing of this work has been a great help to me from the very beginning, and I owe him much for his feedback and comments. Allan Borodin's observations have always been spot-on, and I was very fortunate to have his perspective and his advice. I cannot thank Avner Magen enough for sharing with me his knowledge on expander graphs —a topic with which, as he once told me, I could very easily lure him into a conversation. This work would not have been placed in the proper context without Jörg Liebeherr's deep understanding of peer-to-peer systems; on a personal level, I cannot overstate the effect that his support and encouragement has had on me. Finally, a special thanks to Don Towsley, the external reviewer of this thesis, both for his detailed comments and for the numerous different directions and applications that he saw in this work.

I was very lucky to be surrounded by an amazing group of colleagues —Sam Hasinoff, Faisal Qureshi, Philipp Hertel, Kleoni Ioannidou, Alan Skelley, Xiaoyang Guan, Larry Zhang, Felix Wong, Amin Tootoonchian, Geoff Salmon, Monia Ghobadi, and all the members of the networking group; you have all been great friends, and I will always have a fond memory of my years in Toronto because of you. Many thanks also to everyone I met and worked with at the Thomson lab in Paris, especially Augustin Chaintreau and Laurent Massoulié; the lab has been, and will continue to be, one of the most exciting places young networking researchers can find themselves in.

Last, but not least, I would like to say a warm, heartfelt thank you to my family back in Greece and to my wife, Yan Wang; none of this would have been possible without your unconditional love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The subject of this thesis is searching in unstructured peer-to-peer systems. In general, peer-to-peer systems are computer networks created to allow *peers* —*i.e.*, the participants in the network— to share their resources, such as bandwidth, storage capacity and computational power. Typically, the network formed by the connections among these peers is highly dynamic: as time progresses, new peers may arrive and establish new connections, while existing peers may depart. Moreover, all operations performed by peers are executed in the absence of a central authority and, in this sense, peer-to-peer systems are distributed systems.

Peer-to-peer systems have been used for a variety of different applications. The application we focus on in thesis is a *file-sharing* application. In this application, the peer-to-peer system is deployed to enable peers to share their files. More specifically, in such systems, peers (a) make their own files publicly available and (b) use the peer-to-peer network to locate and download files shared by other peers. Such file-sharing applications (*e.g.*, Gnutella [SRS08], Kazaa [LKR05] and eDonkey [HKL$^+$06]) have been very popular and, at their peaks, their user population numbered in the millions [SRS08, LKR05].

Peer-to-peer systems have also been deployed to distribute content (BitTorrent [QS04, NRZ$^+$07]) and to stream video (*e.g.*, PPlive [HLL$^+$07], SopCast [sop]). In such applications, content originates from a single source; incoming peers can obtain the content both from the source and from other peers that have already downloaded it, perhaps partially. Such systems have also been very successful, to the extent that in the past few years they have contributed to a large percentage (more than 40%) of all Internet traffic [PDGM06, ipo]. More generally, any application designed to distribute and maintain content over a dynamic community of users can benefit from the peer-to-peer paradigm. For example, peer-to-peer systems have been proposed to maintain distributed, decentralized versions of an encyclopedia [Buz], a web-search

engine [YaC], a torrent search engine [Tri], *etc.*

Searching plays a key role in all of the above peer-to-peer systems. For example, in file-sharing peer-to-peer systems, to obtain the copy of a file a peer must first locate another peer that stores this file. Searching is implemented through a *query propagation mechanism*: a peer interested in a file creates a query message with the file's name, which is then propagated over the network of connections among peers. If one of these messages reaches a peer storing the requested file, the peer that started the query is notified and, subsequently, downloads a copy of the file. Searching through query propagation can similarly be applied to the scenario of a peer-to-peer web or torrent search engine (where the content sought for is a site or torrent relevant to a query), a peer-to-peer encyclopedia (where the content sought for is an encyclopedia entry), *etc.* In peer-to-peer content distribution and video streaming applications, searching can be used to locate peers that have already downloaded partial content [FR08].

In all cases, searching over a peer-to-peer system poses a fundamental challenge, as the query propagation mechanism needs to cope with the system's dynamic nature. Both the network topology and the content present in the system evolve through time, as peers arrive and depart. A query propagation mechanism should successfully locate the data it seeks (files, websites, partial content, *etc.*) in spite of the joint evolution of network and content. Moreover, it should do so by exchanging as few messages as possible. In particular, one would like to design query propagation mechanisms that are *scalable*: given that, typically, the bandwidth available to peers is limited, the query traffic they have to handle should not grow significantly as their numbers increase. In addition, query propagation mechanisms should also be *reliable*, in the sense that, if the content is in the system, a search will be able to locate it with reasonable guarantees. These two goals can be conflicting: for example, flooding the entire network with query messages is reliable but unscalable. On the other hand, sending a query message to, *e.g.*, a constant number peers can scale very well but is unreliable, as it may not guarantee that content can be located even if it is widely available in the system. The challenge therefore posed by searching in peer-to-peer systems is establishing a favourable trade-off between these two goals, given that a peer-to-peer system can be highly dynamic. Ideally, one would like to design a mechanism that exhibits both good scalability and good reliability properties.

The peer-to-peer systems that have been designed to meet this challenge can be classified into two different categories: structured and unstructured. The main premise behind structured peer-to-peer systems is that peers are organized so that the network they form exhibits a predetermined structure. The graph structures used in such systems include the by now well-known Chord graph [SMK+01], de Bruijn graphs [LKRG03, KK03, NW03], small-world

graphs [MBR03], butterfly graphs [MNR02], *etc.* In contrast, in unstructured systems peers do not actively try to maintain any underlying structure [SRS08, LKR05, HKL+06]. As a result, the network they form can be arbitrary, at any point in time.

The main advantage of the structured over the unstructured approach is that a priori knowledge of the underlying structure can be exploited to locate content efficiently: a requested file in a structured file-sharing system can be located with only a small number of message exchanges —typically, $O(\text{poly}\log n)$, where n the number of peers in the system. On the other hand, the maintenance of the graph structure incurs a considerable additional traffic load on peers. For example, maintaining the network in Chord [SMK+01] requires that $\Theta\left(\log^2 n\right)$ messages are exchanged at each arrival and departure. Additional messages also need to be exchanged because structured systems redistribute content, both after a new arrival and prior to a peer's departure. Moreover, most structured systems require graceful peer departures, *i.e.*, peers must perform certain operations to assure that the graph structure is maintained after they leave. In general, however, peers may fail or depart abruptly. The network is very sensitive to such departures and, in this sense, structured systems are not robust. To achieve robustness, additional "stabilization" operations need to be performed periodically to recover the underlying structure [CCR03, BR03, LSG+04], thereby incurring additional maintenance costs.

In contrast, since unstructured systems do not require the maintenance of a specific graph structure, very little maintenance traffic is incurred during peer arrivals and departures —both graceful and abrupt. There are additional drawbacks to structured approaches, such as limited query expressiveness [CRB+03] and vulnerability under attacks [SENB07]. Nonetheless, the large maintenance costs under churn and the lack of robustness are the main reasons why the interest in unstructured approaches has remained high even after the introduction of structured peer-to-peer systems in the field; in particular, many real-life applications (including Gnutella, Kazaa and eDonkey) have adopted the unstructured approach in their design.

In spite of these low maintenance costs, coping with the arbitrary nature of the network in unstructured peer-to-peer systems makes the design of query propagation mechanisms for such systems difficult. Although several models of the evolution of an unstructured network have been proposed (see also Section 4.3), such models have not been used to analyze the behaviour of query propagation mechanisms on such networks. Instead, the analysis of mechanisms proposed in recent years has mostly focused on static behaviour, *e.g.*, by modelling the network as a static random graph (see also Section 2.3). Most importantly, the following fundamental question remains open: is there a query propagation mechanism for unstructured peer-to-peer systems that is both scalable (in the sense that the query traffic it generates scales well with

the system's size) and reliable (in the sense that files available in the system can be reliably located)?

In this thesis, we answer the above question in the affirmative. In particular, we propose a model that accounts for the time-varying behaviour of both the network topology and the content stored by peers in an unstructured system. We then construct a novel query propagation mechanism and show that it is both scalable and reliable; this result is shown analytically through our proposed model and is also verified through simulations. To the best of our knowledge, our work is the first to discuss the scalability and reliability of a query propagation mechanism under a model capturing the joint evolution of both network and content in an unstructured peer-to-peer system.

For the sake of concreteness, the above results are presented in the context of a file-sharing application. Nonetheless, our model and our analysis are quite general; our results and methods can be used to understand the scalability and reliability of searching in other unstructured peer-to-peer applications as well. In particular, our analysis covers two different variants of the unstructured peer-to-peer system discussed above: a *pure* peer-to-peer system, which is essentially the system described so far, and a *hybrid* system, in which a central server co-exists with the peer-to-peer network. The hybrid system is motivated by partially decentralizing a server-based content-distribution service (*e.g.*, an online encyclopedia, a search engine, a partial-content tracker, *etc.*) and, as such, can give useful insight on the non-filesharing peer-to-peer applications we discussed above.

In the remainder of this chapter, we give a more detailed presentation of the issues of scalability and reliability as they arise in the hybrid and the pure case. Moreover, we present two query propagation mechanisms that will have a central role in our analysis, namely, the random walk and the expanding ring. We then give an overview of the main contributions of this thesis, and conclude with an outline of the thesis organization.

1.1 Two System Variants and Two Query Propagation Mechanisms

Two different variants of an unstructured peer-to-peer system are covered in this thesis: hybrid and pure peer-to-peer systems. Moreover, our results will involve two query propagation mechanisms called the random walk and the expanding ring. Below, we give a brief description of these two system variants and these two query propagation mechanisms.

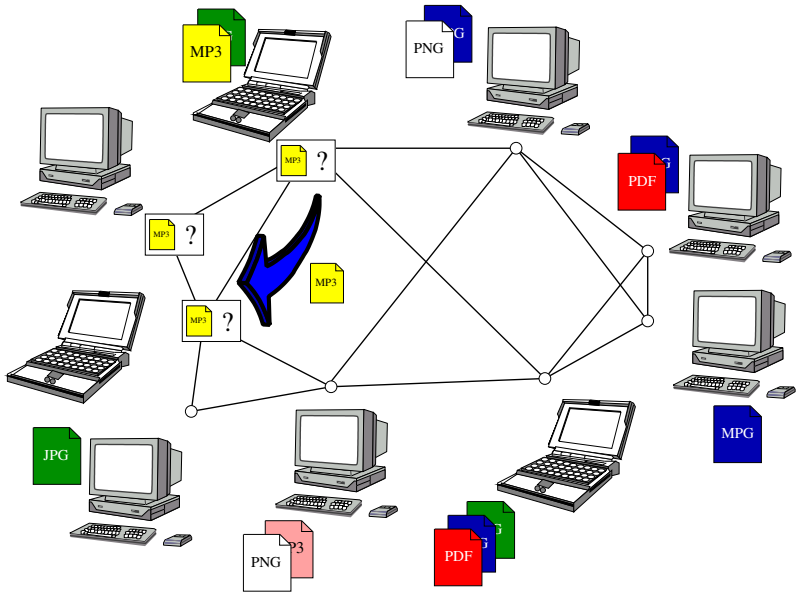


Figure 1.1: A pure peer-to-peer system. Peers connect to each other and form a network with the purpose of sharing files. This network is used to propagate queries: when a query reaches a peer storing the requested file, this peer responds to the one that issued the query and provides it with a copy of the file.

1.1.1 Pure vs. Hybrid Peer-to-Peer Systems

The system we refer to as a pure peer-to-peer system is essentially the one presented in the beginning of this chapter. As illustrated in Figure 1.1, peers form a network, which they use to search for files stored by other peers. When a query reaches a peer that has the file, this peer notifies the one that issued the query and lets it download a copy. A peer that downloads such a copy shares it for the remainder of the time it stays in the system. As we have already discussed, the main challenge posed by the design of query propagation mechanisms for such a system is the following. Ideally, a query propagation mechanism should not only be scalable, in the sense that the traffic it imposes on peers does not grow fast with the system size, but also reliable, in the sense that files that are present in the system should be likely to be located. Hence, the question we wish to answer in the context of unstructured, pure peer-to-peer systems is whether a search mechanism having both of the above properties exists.

In this thesis, we also consider another system architecture which we call hybrid, as it consists of both a peer-to-peer network and a central server. The server stores files and peers act as clients of the server, by requesting and retrieving these files. The purpose of deploying

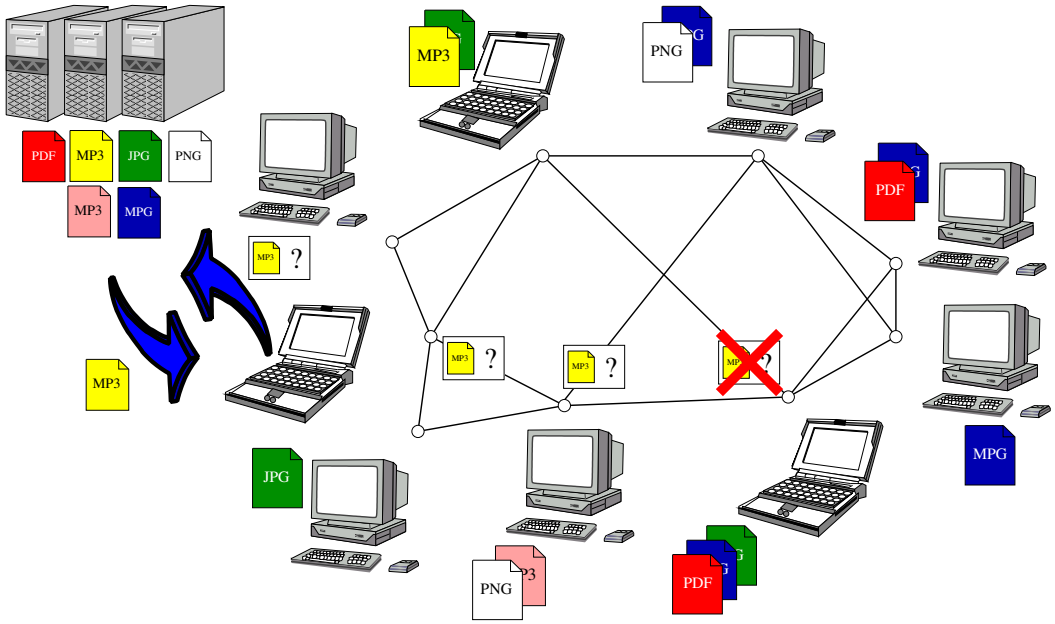


Figure 1.2: A hybrid peer-to-peer system. The network formed by peers coexists with a central server. Queries are first propagated over the peer-to-peer network. In this example, a query is dropped before reaching a peer that stores the requested file. As a result, the search fails and the peer that issued the query requests the file directly from the central server.

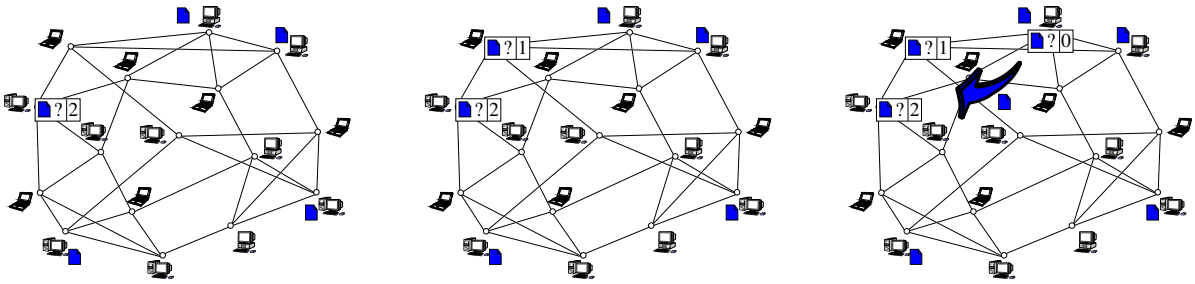
the hybrid architecture is to reduce the traffic at the server by delegating a fraction of it to the server's clients —*i.e.*, the peers. In particular, a peer wishing to retrieve a particular file first propagates a query over the peer-to-peer network. If another peer that has already downloaded this file is reached, then the file is downloaded without the intervention of the server. If, on the other hand, the query fails to locate such a peer, the file is downloaded directly from the server. This hybrid architecture is illustrated in Figure 1.2.

Though in this thesis we focus on file-sharing, there are many traditional client-server applications that could benefit from the above architecture. Some examples can be drawn from the beginning of this chapter: a hybrid peer-to-peer system could be used to partially decentralize a search engine, an online encyclopedia or a partial content tracker. More generally, a hybrid approach could be used to alleviate the traffic load sustained by any centralized content distribution system.

Hybrid systems are, by definition, reliable: every file can eventually be retrieved from the central server, so every query is guaranteed to succeed. However, the scalability of a hybrid system is determined not only by the query traffic imposed on peers but also by the query traffic

that eventually reaches the server. For a given hybrid system, we would like to understand (a) how fast the bandwidth available at the server has to grow as the number of peers increases, and (b) how the traffic load imposed on individual peers grows as a function of the peer population. There is an inherent trade-off between these two quantities, similar to the trade-off between scalability and reliability in the pure peer-to-peer case: increasing the number of messages one propagates within the peer-to-peer network increases the load on peers but also decreases the load on the server, as it increases the likelihood that the file is located within the peer-to-peer network.

Ideally, we would like to design our system so that both the server and the peer traffic loads are low. Low server traffic is indeed the reason for deploying the hybrid architecture in the first place. On the other hand, peers typically have limited bandwidth resources, so our design should avoid overwhelming them with query traffic. Hence, the question we wish to answer in the context of hybrid systems is whether there exist query propagation mechanisms that scale well in terms of both traffic loads.



(a) (b) (c)

Figure 1.3: Example of a random walk search. In (a), the peer initiating a search sends a message to one of its neighbours, chosen uniformly at random. The message time-to-live field is initialized to 2 and decremented with every transmission. In (b), as the peer receiving the message does not store the requested item, it decrements the value of the time-to-live field and forwards the message to one of its neighbours. The peer in (c) has the item, and thus responds to the source peer by providing it with the requested item.

1.1.2 Random Walk and Expanding Ring

The two main query propagation mechanisms that are analyzed in this thesis are the *random walk* [LCC+02, GMS04] and the *expanding ring* [LCC+02, TK06]: in particular, in the hybrid case we only focus on these two mechanisms, while in the pure case we also consider a variant

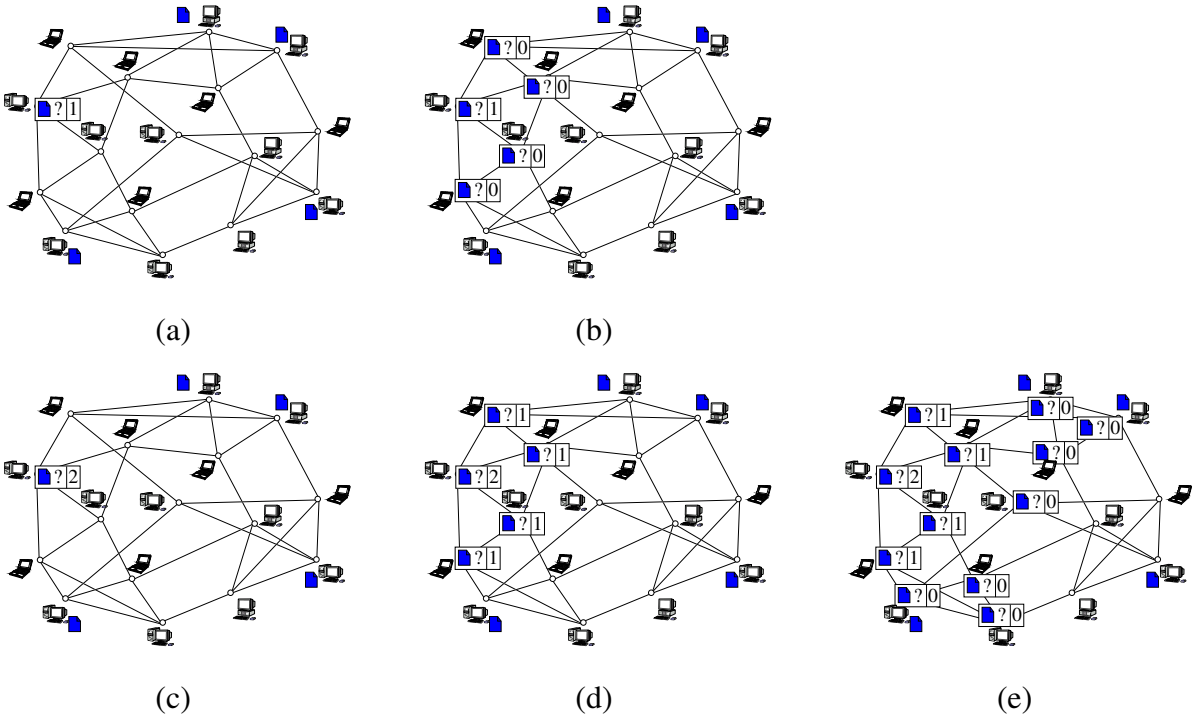


Figure 1.4: Example of an expanding ring search. The expanding ring in this example consists of two simple flooding stages, shown in figures (a)-(b) and (c)-(e), respectively. In the first stage, the time-to-live field is initialized to 1, so only the one-hop neighbours are reached. In the second stage, the time-to-live field is set to 2, so peers within two-hops from the source are reached.

of the random walk. The discussion below serves as a brief introduction to the random walk and the expanding ring, both of which are illustrated in Figures 1.3 and 1.4, respectively. A full description and a more formal treatment of both can be found in Chapter 5, while the random walk variant discussed in the pure peer-to-peer case can be found in Chapter 6.

In simple words, in the random walk mechanism, a peer initiating a search forwards a query message to one of its neighbours, chosen uniformly at random. If the peer's neighbour does not have a copy of the requested content, it forwards the query to one of its own neighbours, chosen again uniformly at random. This process is repeated until either the content is located or the query message expires. The expiration is determined by a time-to-live mechanism: each query message carries a time-to-live field, which is initialized to a predetermined value and is decremented with each transmission. When the time-to-live field becomes zero, the message is no longer propagated.

In the expanding ring mechanism, the peer initiating the search forwards a message to *all*

its neighbours. Again, this process is repeated by neighbours that do not have the content: any peer receiving a query message and not having the requested content forwards the message to all its neighbours. Similarly to the random walk, a time-to-live mechanism is used to limit the query propagation: query messages cease to be forwarded either when the content is located or when their time-to-live becomes zero. However, if a query expires and the file has not been found, the peer initiating the query can increase the expiration threshold and repeat the search. Increasing the expiration threshold increases the number of peers reached, hence the name "expanding ring".

Compared to the random walk, the expanding ring propagates messages far more aggressively. As such, in general, it generates more traffic while locating the requested content faster ([LCC+02], see also Chapter 5). Both mechanisms are appealing because they are simple and easy to implement in a distributed fashion. Moreover, neither requires a priori knowledge of the network, which makes them quite appropriate for unstructured systems. As a result, many of the proposed query propagation mechanisms for such systems are variants of the above two mechanisms (see also Section 2.3).

1.2 Contributions

The contributions of this thesis can be summarized as follows:

(a) We propose a model for the analysis of unstructured peer-to-peer systems. Our model describes the joint evolution of both the network topology and the content present in the system.

(b) We analyze the scalability of the random walk and the expanding ring query propagation mechanisms in hybrid peer-to-peer systems and show that both have excellent scalability properties. For the random walk, we show that both the traffic load at the server and the traffic load at peers remains bounded as the system size grows. For the expanding ring, we show that both of these loads grow very slowly with the system size, to the extent that, for all practical purposes, they can be considered bounded.

(c) We analyze the scalability and reliability of the random walk and the expanding ring query propagation mechanisms in pure peer-to-peer systems, and show that these two mechanisms cannot be both scalable and reliable. Furthermore, we propose a novel mechanism that indeed exhibits both of these two properties.

Below, we give a more detailed description of each of these contributions.

Model

The first contribution of this thesis is to propose a model for the analysis of an unstructured peer-to-peer system that captures the dynamic nature of both the overlay network and the content stored by peers.

Our model is motivated by real measurement studies on unstructured peer-to-peer networks, described in detail in Section 2.2. It differs from earlier models used to analyze the behaviour of query propagation mechanisms in unstructured systems by capturing the evolution of the system as time progresses. In particular, the joint evolution of the overlay network and the content stored by peers is described by a Markov process. This joint Markov process is determined by arrivals and departures of peers, by how they request and share files and by the connection protocol they use to construct the unstructured network.

Previous work on searching in unstructured systems, presented in Section 2.3, assumed that the overlay network was a static random graph and that files were distributed uniformly at random. Several Markovian models, described in Section 4.3 of this thesis, have been proposed in the past to describe the evolution of the overlay network of an unstructured peer-to-peer system. However, such models have not been used so far to analyze the behaviour of query propagation mechanisms.

We note that these earlier models can in fact be seen as special cases of our general model. Because of this, several of our results obtained under our general Markovian model can be applied to these specific models as special cases (see also Sections 5.2 and 6.2).

Hybrid Peer-to-Peer Systems

Our analysis formally shows that a hybrid peer-to-peer system, with an unstructured overlay, scales extremely well when the query propagation mechanism is the random walk or the expanding ring. In particular, we obtain the following two results.

First, we show that if a random walk with a time-to-live proportional to the peer population size is used, the traffic loads at both the server and at individual peers stay bounded as the peer population grows. This result, stated formally in Theorem 5.1, implies that it is possible to construct hybrid peer-to-peer systems that can handle query traffic generated by a large (unbounded) number of peers even when the bandwidth capacities of both the server and the peers are limited.

Second, we show that if an expanding ring with a logarithmic (in the peer population size) time-to-live is used, the traffic loads at both the server and at individual peers will grow very slowly as the peer population grows. The rate of growth of both traffic loads is so slow that both can effectively be considered constant, for all practical purposes. Hence, this result, which is stated formally in Theorem 5.2, implies that a hybrid, unstructured peer-to-peer system using an expanding ring also scales very well. This is important because, compared with the random walk, the expanding ring leads to much shorter query response times.

Both of the above results are shown under our mathematical model. Moreover, they are further validated through simulations in which several of our modelling assumptions are relaxed. To the best of our knowledge, our work is the first to formally characterize the scalability of such hybrid peer-to-peer systems and, in particular, to show that such systems can indeed scale very well.

To prove the above results, we characterize the server traffic load and average traffic load per peer in terms of several system parameters, including the popularity of a file and the topology of the overlay network. Because of the generality of these characterizations, our analysis can be used to address more general questions than the ones we posed above; we present several such extensions in more detail in the end of our discussion on hybrid systems (Section 5.7).

Pure Peer-to-Peer Systems

We also use our proposed model to analyze the scalability and reliability of search mechanisms in pure peer-to-peer systems. In particular, we obtain the following two results.

Our first result is negative: we show that neither the random walk nor the expanding ring are simultaneously scalable and reliable. This is stated formally in Theorem 6.1 and again proved under our mathematical model and validated through simulations. Intuitively, for both the random walk and the expanding ring, a system designer can choose an initial time-to-live value so that the mechanism is scalable or reliable, but there is no value guaranteeing that the mechanism exhibits both of these properties. This behaviour occurs because of the inability of these mechanisms to deal with queries for absent files, *i.e.*, files that are not in the system. In particular, reducing the time-to-live decreases the traffic generated by such queries, thereby improving the system's scalability. However, it also makes the system more unreliable, leading to low query success rates. We note that queries for absent files have been observed to occur frequently in real-life peer-to-peer systems (see also Section 2.2.3).

Our second result is positive: we construct a simple, easy to implement query propagation

mechanism that is both scalable and reliable. This mechanism is based on the idea that the absence of a file can be detected through query failures. In particular, a peer that fails to locate a file considers this file to be absent from the system, from that point on. In this sense, the "absence of evidence" of a file (*i.e.*, the failure to find it through a search) is treated as "evidence of absence" (*i.e.*, evidence that the file is not present in the system). This information can then be used to constrain queries: a peer halts the propagation of queries for files it considers absent from the system.

Our work shows that a random walk mechanism that uses "evidence of absence", as described above, is scalable, as queries for unavailable files are stopped early on. Most importantly, it is also reliable: if a file is present sufficiently often, almost all queries for it are guaranteed to succeed. These two properties are stated formally in Theorem 6.2, which is also proved under our mathematical model and further validated through simulations.

To the best of our knowledge, this is the first formal demonstration that the random walk and the expanding ring are not both scalable and reliable. Most importantly, the random walk using "evidence of absence" is also the first search mechanism for unstructured peer-to-peer networks has been shown to have both of the above two properties. For this reason, these two contributions are of fundamental importance in understanding the scalability and reliability of pure, unstructured peer-to-peer systems.

1.3 Thesis Overview

The remainder of this thesis is organized as follows.

In Chapter 2, we give an overview of existing peer-to-peer systems and of related work in the area. In particular, we discuss how modern unstructured systems are designed and review measurement studies of such systems. Our main focus with respect to related work is on search (*i.e.*, query propagation) mechanisms, and on models employed to analyze their performance. We note that several of our modelling assumptions in Chapter 4 are motivated by the experimental observations presented in this chapter (see also Section 2.2.4). One such observation, that will play a crucial role in our analysis, is that the overlay network of unstructured systems is "almost" regular (*i.e.*, the peer degree distribution is concentrated around a constant that does not depend on the system size).

Chapter 3 serves as a technical preliminary to the rest of the thesis. We first give a brief introduction to random walks on finite graphs. In doing so, we also review a few fundamental results about Markov chains and Markov processes, which is helpful as both are extensively

used throughout this thesis. We then relate the random walk on a graph to a quantity called the relaxation time of the graph. Finally, we introduce expander graphs —namely, graphs with small relaxation times. Almost all regular graphs are expanders (see Section 3.3.2); this, along with their relationship to random walks, is the reason why expanders will have a recurring role in our analysis.

Chapter 4 consists of an in-depth presentation of our model of an unstructured peer-to-peer system, used in both the pure and the hybrid setting. In particular, we describe how peers arrive and depart and how they request and publish data. We also define how the overlay network can change under system churn; more specifically, we model these changes through a Markov process whose state space is a subset of all regular graphs of a given size.

Our analysis of a hybrid peer-to-per system is presented in Chapter 5. We start with the hybrid case as it is simpler. We show that a hybrid system can be represented as a Markov process that does not depend on the query propagation mechanism used and establish conditions under which this process is ergodic. We then use this process to compute the average traffic load per peer and the server load for the random walk and the expanding ring. Finally, we discuss several possible immediate extensions of our results, as well as open problems on hybrid systems.

The pure peer-to-peer setting is presented in Chapter 6. We first focus on the analysis of the random walk and the expanding ring, and show that these mechanisms cannot be both scalable and reliable. We then analyze the behaviour of the random walk that uses "evidence of absence", and show that it is both scalable and reliable. Interestingly, the scalability of this mechanism is an immediate implication of the scalability of the hybrid system. Proving reliability requires more effort; we use a mean field limit method to obtain this result. Finally, as in Chapter 5, we conclude by discussing possible extensions and open questions.

Chapter 2

Background and Related Work

This chapter has the following three goals. The first is to describe how unstructured peer-to-peer applications are designed in practise. This aims to provide insight on how these systems behave as well as on the type of constraints accounted for in their design. The second goal is to review experimental measurement studies of such systems, and to use these studies to understand properties like, *e.g.*, their population size, peer churn, and the topology of the network formed by peers. Finally, as the focus of this thesis is on search in unstructured systems, our third goal is to review the state of the art in query propagation mechanisms proposed for such systems.

To that end, in the first section of this chapter we give an overview of the unstructured file-sharing system called Gnutella. Our description of Gnutella consists of a presentation of the Gnutella overlay network as well as of the search mechanism employed by peers to locate files. Our focus on this particular system is due to the fact that it is not proprietary and, as a result, its behaviour and properties are known and well documented.

In the second section of this chapter we present experimental measurement studies of unstructured peer-to-peer file-sharing systems (including Gnutella, Kazaa, eDonkey and Napster). The measurements discussed can be categorized as follows: (a) measurements of the peer population dynamics, describing how frequently peers arrive and depart and how the population size changes through time, (b) measurements of the topology of the peer-to-peer network and (c) measurements of the popularity and availability of files in the system. Note that all three of these quantities can affect the behaviour of a query propagation mechanism over an unstructured network; therefore, an analysis of such a mechanism needs to take into account how these quantities behave in practise. In our work, this will be captured by basing several of the modelling assumptions that we make in Chapter 4 on the measurement studies presented in

this section.

In the third and final section of this chapter we present related work in the area of search in unstructured peer-to-peer systems. We discuss proposed query propagation mechanisms for peer-to-peer systems and the analytical results known to hold for these mechanisms. The mechanisms we discuss can be grouped into passive replication mechanisms, in which files are only replicated by peers that request and download them, and proactive replication mechanisms, in which files can be proactively replicated at peers that do not request them. We conclude this section by discussing mechanisms proposed to deal with queries for files that are not present in the system.

2.1 An Overview of the Gnutella File-Sharing System

In this section, we overview the Gnutella peer-to-peer file-sharing system [SRS08]. Because both the Gnutella protocol and several of its implementations (*e.g.*, the LimeWire client [SRS08]) are open-source, both the connection protocol followed by Gnutella peers and the query propagation mechanism they deploy are well documented (see, *e.g.*, [SRS08, DWD04, AC08a, LHH+04, SGG02a] and many others). Far fewer studies (and of a more limited scope) exist for proprietary file-sharing networks like Kazaa [LRW03, LKR05] and eDonkey [HKLM04, HKL+06]. Though our main focus is on Gnutella, for reasons of completeness we will contrast the behaviour of Gnutella to these two systems, whenever the behaviour of the latter two is known.

2.1.1 The Gnutella Overlay Network

In this section, we describe the network formed by Gnutella peers, also known as the *overlay network* of the system. The first widely released version of Gnutella (Gnutella 0.4) adopted a "flat" architecture, in which all peers were equal [SGG02a, CRB+03]. In contrast, the most recent version of Gnutella (Gnutella 0.6) adopts a hierarchical, two-tier architecture. In this architecture, peers are classified into two different categories, namely *super-peers* and *leaf-peers*. Super-peers can connect to other super-peers and to leaf-peers, while leaf-peers can "attach" themselves only to super-peers. As a result, the overlay network consists of two distinct levels: a higher level, comprising the super-peers, which connect to each other and to leaf peers, and a lower lever, comprising the leaf-peers, which are connected only to peers of the higher level.

Typically, a peer arriving in the system becomes a super-peer only if it satisfies certain requirements; for example, a super-peer must have a high-bandwidth Internet connection and not be behind a firewall [Limb]. A leaf-peer can connect to at most three super-peers [SRS08]. Moreover, super-peers allow no more than 30 leaf-peers to be attached to them [SRS08,Limb]). Similarly, super-peers maintain no more than 32 connections to other super-peers [SRS08, Limb].

The exact overlay network topology in Gnutella is determined by the connection protocol followed by super-peers and leaf-peers. This consists of a set of rules dictating peer operation when new peers enter the network, existing peers depart, or connections are abruptly dropped. Below, we briefly outline the connection protocol followed by Gnutella peers.

To begin with, each peer (super-peer or leaf-peer) maintains a cache of known hosts, *i.e.*, a list of IP addresses that correspond to super-peers known to be online [Gnu]. At any point in time, a peer is connected to only a subset of the hosts maintained in this list.

This list is replenished frequently, using two methods. The first is proactive: every peer can discover new peers by sending "ping" messages [Gnu]. By the protocol specification, a peer receiving a "ping" message is required to respond by providing its own cache of known hosts. This information can be added to the local cache, even if it is not used to establish more connections. Moreover, failure to respond to a "ping" message signals that a host has become inactive, and should thus be removed from the cache.

The second method is reactive: a super-peer can monitor query traffic it forwards, and learn the IP addresses of other super-peers issuing queries or query responses [Gnu]. As query traffic is propagated only among super-peers, this method is only useful to super-peers. Leaf peers can benefit from it only indirectly, by sending "ping" messages to a super-peer.

This cache of known hosts is stored locally and is thus available even when a peer is off-line —though it may, of course, become outdated. A peer wishing to enter the network can consult it before establishing its connections. For example, the Gnutella 0.6 RFC outlines the following three ways for an incoming peer to discover active peers in the network: [Gnu]

1. An incoming peer can discover peers by contacting a well-known web server that maintains a cache of active peers. Use of this method is discouraged; it should only be employed as a last resort, when the methods below fail.

2. An incoming peer can attempt to contact hosts discovered during previous online activity that are stored in the peer's local cache.

3. Once a peer has established a connection to other active peers, it may "ping" them to discover more hosts and establish more connections.

We note that the same protocol is used by both leaf-peers and super-peers when they enter the system, though a leaf connects to only a few super-peers (at most three).

The two-tier overlay architecture is also adopted by Kazaa [LKR05] and eDonkey [HKL+06]. In particular, it has been experimentally observed that, in Kazaa, a leaf-peer can connect to only one super-peer, a super-peer can connect to about 150 leaf-peers [LKR05], while the super-peer to super-peer connections are around 45-50 [LKR05]). Finally, peers in Kazaa also maintain a cache replenished through "ping" messages, and bootstrapping involves use of this cache in a manner similar to the one described above [LKR05].

2.1.2 The Gnutella Query Propagation Mechanism

The first version of Gnutella (Gnutella 0.4) used a simple flooding mechanism to propagate queries over the "flat" overlay formed by peers [CRB+03,Rit01]. According to this mechanism, the header of each query contains a time-to-live field that is initialized to a predetermined value. A peer receiving a query checks if any of the names of its files matches the query terms. If some do, it sends the list of the results to the peer that initiated the search. It then decrements the time-to-live field of the query. If the resulting value is larger than zero, the peer forwards the query to all its neighbours, which repeat the above process. If not, the query is dropped.

It was noted from early on that this mechanism generated considerable traffic over the network, rendering the first version of Gnutella unscalable [Rit01]. The query propagation mechanism used in modern Gnutella (Gnutella 0.6), presented below, is a variant of the expanding ring mechanism, and exploits the two-tier architecture to reduce the number of messages propagated. In particular, according to the Gnutella 0.6 protocol specification [Fis03], query traffic is propagated only among super-peers. After attaching itself to a super-peer, a leaf informs the super-peer of all the files it shares. In turn, every super-peer maintains a table with all files stored by leaf-peers attached to it [Lima]. A query generated by a leaf-peer is handed over to the super-peer(s) to which the leaf is attached, and is forwarded over the super-peer overlay network. A super-peer can resolve incoming queries without ever forwarding them to its leaves, by simply consulting the table of leaf contents it maintains.

The propagation over the (higher level) network formed by super-peers takes place as follows. When a super-peer receives a query from a leaf-peer for a file its other leaves do not store, it creates a query message and forwards it to all its neighbouring super-peers. The mes-

sage is propagated according to the simple flooding algorithm presented above. In particular, the message header contains a time-to-live field, that is initialized at a small value (*e.g.*, 1). A super-peer receiving a query message checks if any of the leaves attached to it can resolve the query, by looking at its table of leaf-peer contents. If the query can be resolved, it responds to the super-peer that initiated the query by sending the list of results. After doing so, it decrements the time-to-live field of the query message and, if it is not equal to zero, it also forwards the query message to all its neighbours. The message is thus forwarded until its time-to-live becomes zero.

The super-peer that initiated the query waits for a predetermined period. If, by the expiration of the waiting period it has not received at least 50 results, it initializes a new query with a higher time-to-live value than the one previously used. To compute how the time-to-live should be increased, a peer first estimates how many super-peers it has reached so far using the following heuristic:

$$\sum_{i=0}^{\mathrm{TTL}-1} (d-1)^i \tag{2.1}$$

where $d = 32$ and TTL the initial value of time-to-live field. It then computes the "density" of results for this query, defined as the number of results it has received so far divided by the number computed from (2.1). The time-to-live is then increased to a value such that the number of peers reached, according to (2.1), multiplied by the density, exceeds 50 results.

This process is repeated until either the super-peer receives 50 results, or the time-to-live value exceeds the value 4. The latter number is chosen heuristically, given that (2.1) exceeds 200,000 for $\mathrm{TTL} > 4$, considered by the system designers as the maximum number of super-peers that should be queried [Fis03]. As each leaf is connected to (at most) three super-peers, the above implies that, should such results exist, a leaf-peer receives roughly 50-150 results, per query; this number is deemed satisfactory for most users [Fis03].

2.1.3 Discussion on the Gnutella Peer-to-Peer System

The defining characteristic of unstructured systems is that the overlay network is, at any point in time, assumed to be arbitrary. As a result, a query propagation mechanism cannot rely on the fact that the overlay network exhibits a certain structure that can be exploited during searching. In practise, the overlay graph of Gnutella is a two-tier graph, and therefore does have an underlying structure. This structure is obviously exploited during a search, as query propagation is only limited to the network formed by super-peers. Moreover, the heuristic

(2.1) is based on the assumption that the overlay network is almost a regular graph (*i.e.*, a graph in which all peers have the same degree), as the number of connections per super-peer is approximately 32. Apart from the "almost" regularity assumption however, no additional assumptions are made on the topology of the graph representing connections among super-peers.

Having a hierarchical architecture aims at making searching more efficient. Intuitively, fewer query messages need to be propagated over such a system compared to a "flat" architecture, as queries never need to reach any leaf-peers. In this sense, a query forwarded over the super-peer network "covers more ground" faster than a query propagated over a "flat" architecture. There is however an inherent trade-off in this approach: maintaining the table of leaf contents incurs additional traffic at super-peers.

The problem with hierarchical architectures is that they cannot scale indefinitely. As noted above, super-peers do not connect to more than a limited number of leaves. As a result, to accommodate the growth of the overall population, the size of the super-peer network will have to grow proportionally to the total number of peers. This leads to the same scalability issues that arise in a "flat" architecture. Allowing an unlimited number of leaves per super-peer or increasing the levels of the hierarchy does not solve the problem: the traffic incurred at the top-level super-peers while maintaining their table of leaf contents or while receiving query traffic from lower-level peers would then grow with the system size. Moreover, increasing the number of leaves or the levels in the hierarchy makes the network less robust: top-level peers are responsible for maintaining connectivity for a non-negligible fraction of the peer population.

In conclusion, a two-tier system only ameliorates the problem of scalability, without solving it for all possible peer population sizes. For this reason, the analysis presented in this thesis focuses on a "flat" architecture. We note however that, since addressing scalability in a two-tier network essentially reduces to addressing the scalability of a "flat" system, namely, the one formed by super-peers, our analysis can be applied to such a system as well, with appropriate modifications. In particular, our model of the overlay graph can be seen directly as a model of the top-level network formed by super-peers (see also Section 5.7.2).

2.2 Peer-to-Peer Network Measurements

In this section, we give an in-depth account of several key properties of unstructured systems, as observed by experimental measurements. The measurement studies we overview cover three

different properties of an unstructured peer-to-peer system. The first property is the evolution of peer population, *i.e.*, the process describing how peers arrive and depart, and how the peer population changes both within short and long-term periods of time. The second property is the overlay graph, *i.e.*, the time-evolving graph representing the network of connections established among peers. The final and third property is the evolution of content in the system, captured by the popularity of files in the system —*i.e.*, how often they are requested— as well as by their availability —*i.e.*, how many replicas of each file are present in the system.

All three of these properties can affect the behaviour of a query propagation mechanism over the peer-to-peer network. The peer population dynamics determine how the system size increases through time and as well as how volatile this quantity is. The connections among peers are used precisely to propagate queries, so how quickly a query "spreads" over the network is highly dependent on the topology of the overlay graph. Finally, a file's popularity determines the frequency of queries for this file, while its availability determines how quickly such queries can be resolved. As a result, any analysis of a query propagation mechanism of an unstructured peer-to-peer system needs to account for these three quantities. In particular, it needs to capture how these behave in real unstructured peer-to-peer applications. For this reason, as we mentioned in the beginning of this chapter, several of our modelling assumptions (appearing in Chapter 4) will be based on observations derived from the measurements presented below.

2.2.1 Peer Population Dynamics

Peer-to-peer networks can grow significantly over time, potentially reaching population sizes on the order of millions. For example, in November 2000, the number of online peers in Gnutella was approximately 2,000; within 6 months, this number had grown to approximately 50,000 peers [RIF02]. By April 2004, the number of peers had grown to 400,000 [SRS08], and the growth continued until January 2006, by when the total peer population had surpassed 3 million [RSR06]. Similar growth patterns have been observed for other networks, like Kazaa [LRW03] and Napster [SGG02a].

Typically, this growth occurs over long-term periods of time, such as months or years [SRS08, RIF02]. Over shorter periods of time, such as days [DWD04] or weeks [SGG02b, SR04], the population size is relatively stable, tending to oscillate around a fixed value. For example, Deschenes *et al.* [DWD04] measured the size of the Gnutella peer population during different hours of the day between April 10th and April 16th of 2003. The authors observed a variation

in the population size of about $\pm 10\%$, during a day (averaged over the seven-day measurement period).

The variation of the population size throughout a week-long period is also small. Saroiu *et al.* [SGG02b] measured the Gnutella population size between May 6th and May 14th, 2001. Throughout the entire measurement period, the size remained approximately equal to 10,000 peers, with a maximum variation of $\pm 17\%$. Note that, although the population size did not vary considerably, the system was in fact under high churn: approximately 1.2 million distinct peers were observed throughout the seven-day measurement period. Similar measurements were conducted by Stutzbach and Rejaie [SR04] in the week between August 31st and September 8th, 2004. The population size was, at all times, approximately 85,000 peers[1], with a variation of about $\pm 18\%$. Again, the system's churn was very high: the total number of distinct peers observed in [SR04] during this one-week period was approximately 3 million.

In conclusion, unstructured peer-to-peer systems do experience growth and, therefore, scalability is a property that must be considered in their design. This growth however happens over long-term periods of time (*e.g.*, months or years); over short-term periods of time (such as days or weeks), although the consistency of the peer population may change significantly, its size remains relatively stable.

2.2.2 Overlay Network

The *overlay graph* of a peer-to-peer system is the graph representing the connections among peers. This graph is typically changing rapidly under churn: the network evolves as new peers arrive and old ones depart, as dictated by the connection protocol followed by peers (see also Section 2.1.1). As a result, the topology of the overlay graph can change drastically in a matter of hours. For Gnutella, it has been observed that the edge set of two different snapshots of the network taken two hours apart can differ by as much as 48% [SRS08]. Similarly, the average duration of a connection between super-peers in the Kazaa network has been measured to be close to 11 minutes, with more than 32% of such connections lasting less than 30 seconds [LKR05].

Although peer connections are rapidly changing, the number of connections a super-peer maintains is relatively stable [LKR05, SRS08]. This is because, as discussed in Section 2.1.1,

[1]The figures of [SR04] include only the super-peer population. The figures of [DWD04] and [SGG02b] are over total population sizes: at the time the latter two measurements were conducted, super-peers had not yet been introduced to the Gnutella network.

once connections are dropped, a peer can establish new connections through its cache of known hosts. In particular, recent studies [SRS08] show that the distribution of degrees in Gnutella is highly concentrated around 32, the limit value set by the protocol.

In particular, Stutzbach *et al.* [SRS08], studied the degree distribution of the overlay graph formed by Gnutella super-peers. The authors obtained four different snapshots of the network, taken between 09/27/04 and 02/02/05, and observed that the peer degree was concentrated around 32, in all four measurements. The fraction of peers with a degree between 24 and 32 was 57%-69%. Only 6%-12% of peers had a degree higher than 32 —these were peers using software not complying with the protocol [SRS08].

Stutzbach *et al.* also observed that the degree is correlated with a peer's age. Therefore, peers having a small degree were observed to be in a "transient" state: they had recently entered the system, and were trying to establish more connections. Within a few minutes, most peers had established a large number of connections, which they subsequently maintained for as long as they remained in the system. The same stable degree behaviour has been observed in the Kazaa system as well [LKR05]: the number of super-peer to super-peer connections converges quickly to about 45-50, while each super-peer increases its connections to leaf-peers until the latter stabilize to about 150.

The above suggest that, in the above networks, most peers have approximately the same degree. Note that this is consistent with a connection protocol followed by peers like the one described in Section 2.1.1.

2.2.3 File Popularity and Availability

By *file popularity*, we refer to the fraction of new peers that request the file when they join the system. By *file availability*, we refer to the fraction of peers in the network that have a copy of the file. The following two observations were made on real unstructured peer-to-peer systems. First, the popularity and availability of a file varies little within short time periods such as a day or a week [SZR07, AC08b], though both can change significantly over longer periods of time [SZR07, HKL$^+$06]. Second, there is almost no correlation between the popularity and the availability of a file: some files are very popular (*i.e.*, a large fraction of new peers request them) but hardly available (*i.e.*, only a small fraction of peers have a copy), while others are unpopular but widely replicated [AC08b].

In particular, Stutzbach *et al.* [SZR07] measured the availability of 50,000 randomly selected files in Gnutella over a one-day period and observed its change; for all files, the fraction

of peers storing it did not change more than 1% throughout the entire day. Similar observations were made during a one-week interval in Gnutella [SZR07] and a four-week interval in eDonkey [HKL+06]. However, over a period of a year (06/2005 to 06/2006), the availability of the 10 most available files in Gnutella changed by as much as 50% [SZR07]. Similar observations have been made about file popularity. Acosta and Chandra [AC08b] monitored 2.5 million queries generated by peers in Gnutella over an 8-day period in April 2007. The authors grouped queries occurring during the same one-hour period together, and ranked them according to their frequency (within the one-hour period). They observed that the set of queries, and their relative ranking, varied little throughout the one week period.

Queries for files that are not in the system are very common in practise. In a different study [AC08c] taking place during October 2006 and April 2007, Acosta and Chandra took several snapshots of the Gnutella peer-to-peer system and monitored both queries issued by peers, as well as files available in the network. They observed that roughly half of the queries (44% in Oct. 2006 and 55.6% in Apr. 2007) could not be matched with any file in the system.

In general, unstructured peer-to-peer systems exhibit a much lower query success rate, close to 10% [ZCSK07]. This low success rate can be attributed to many reasons. For example, time-to-live fields are typically set to such low values that a large fraction of queries for available files do not succeed [LHH+04, LHSH04, AC08c]. Failures may also occur due to poor query resolution mechanisms and misspellings, both in submitted queries and file names; Zaharia *et al.* [ZCSK07] show that employing spelling-error correction mechanisms can improve the query success rate from 10% to as much as 23.5%. Nonetheless, the study of Acosta and Chandra suggests that a significant fraction of queries fail simply because the files sought for are not in the system.

Another important observation made by Acosta and Chandra [AC08c] is that there is almost no correlation between the popularity of a file and its availability in the system. In particular, the authors ranked files both according to their popularity and their availability, and observed a high discrepancy between the two rankings for the vast majority of files. This discrepancy was manifested both by files that were popular but rare as well as by files that were widely available but infrequently requested.

The fact that there is almost no correlation between the popularity and the availability of a file seems, at first glance, puzzling and counterintuitive: one would expect file availability to be proportional to file popularity, as the more popular a file is the more it should be replicated in the system. One possible explanation that comes to mind is “flash crowds”, *i.e.* spikes in the number of peers suddenly requesting a file. In this case, file popularity could change

drastically over a short period of time (such as minutes or hours), predating the change in availability. However, this is not corroborated by the above studies: First, it has been observed that both popularity and the availability of a file vary little within periods such as a day or a week [SZR07, AC08b]. Second, a flash crowd would also be noticed through an abrupt change in the population size, which, as discussed in Section 2.2.1, was also absent from several studies [DWD04, SGG02b, SR04].

A more plausible explanation is to accept that there is a discrepancy between the number of peers that actually "publish" a file, *i.e.*, bring it in the system with the purpose of sharing it, and the number of peers that request the file. One reason why a file that is very popular may be actually published by only a small fraction of incoming peers is "free riding": peers may not share content they download. Note that, for a file to be available in the system, it has to be brought into the system (published) by at least one peer. If only a small fraction of peers bring a file into the system when they join, queries for this file are likely to fail, and only a small fraction of peers will be able to locate and download the file. As a result, the availability of this file will be small, even if it is very popular and a large fraction of peers request it.

2.2.4 Key Insights from Peer-to-Peer Measurements

To summarize, the measurement studies presented above offer us the following key insights:

- The peer population grows over long-term periods of time (*e.g.*, months or years); over short-term periods of time (such as days or weeks), although the system may be under high churn, its size remains relatively stable.

- The degree distribution of the overlay graph is concentrated around a constant that does not depend on the system size. This is a direct result of the peers' connection behaviour: there exists an upper bound on the number of connections a peer maintains, while dropped connections are replaced by new connections with peers selected from a cache of known hosts.

- A discrepancy can exist between the availability and popularity of a file: a file can be requested often but be poorly replicated and vice-versa. In fact, a large fraction of queries can be for files that are actually not present in the system.

We will use these key insights in Chapter 4, when devising our model of an unstructured peer-to-peer system.

2.3 Proposed Search Mechanisms for Unstructured Systems

In this section, we give an overview of proposed search mechanisms for unstructured peer-to-peer systems. Roughly, these search mechanisms can be classified into two main categories: (a) search mechanisms based on passive replication, and (b) search mechanisms based on proactive replication. Passive replication mechanisms assume that a file is replicated only at peers that request it and download it. In contrast, in proactive replication, a peer can proactively replicate a file at other peers, even if these peers have not requested it. The cooperation of peers in storing such files is a prerequisite for all proactive replication mechanisms.

As proactive replication increases the availability of a file in the system, search mechanisms that use it generate less query traffic compared with mechanisms that use passive replication. This of course comes with an additional traffic cost incurred by the act of replicating files. We note however that, as such mechanisms can only replicate files that are in the system, they still generate a large amount of query traffic for unavailable files.

2.3.1 Search Mechanisms using Passive Replication

Most search mechanisms with passive replication are variants of the basic *random walk* [LCC+02, GMS04] and *expanding ring* mechanisms [LCC+02, TK06] (see also Chapter 5 for a detailed description of these mechanisms). This is because these two mechanisms are simple, easy to implement in a distributed manner, and work on arbitrary overlay network topologies. Lv *et al.* [LCC+02] were the first to propose the random walk and the expanding ring query propagation mechanisms as alternatives to the simple flooding mechanism used by Gnutella 0.4. Lv *et al.* also provide an analysis of the random walk mechanism, in which they approximate it with uniform sampling: that is, the authors assume that the sequence of peers visited by a random walk is similar to a sequence obtained by sampling independently and uniformly among all peers in the network. Under this assumption, it is easy to see then that a file will be located in k steps with probability

$$(1-\alpha)^{k-1}\alpha$$

where α is the availability of the file (*i.e.*, the fraction of peers storing it). Of course, uniform sampling is only a heuristic approximation of a real random walk, as it completely ignores the impact of the network topology.

As we discuss in Chapter 3, several properties of a random walk on a graph can be concisely described in terms of a quantity called the *relaxation time* of the graph. Gkantsidis *et al.*

[GMS04] were the first to exploit this in the context of searching in peer-to-peer networks. In particular, the authors give bounds that relate the relaxation time of a graph to its cover time, *i.e.*, the number of hops required to visit all vertices of the graph through a random walk. Our work follows a similar approach, as we also characterize the behaviour of the random walk through bounds expressed in terms of the relaxation time (see, *e.g.*, Theorems 3.4 and 3.5). Gkantsidis *et al.* were also the first to note that, since random regular graphs are very likely to have small relaxation times, the overlay network of an unstructured system should also be very likely to exhibit this property, thereby making the above bounds useful.

Gkantsidis *et al.* also compared through simulations the query hit-rate (*i.e.*, the number of results obtained) of the random walk, simple flooding and uniform sampling mechanisms. Their simulations were performed over several different fixed topologies and for several different file availabilities. They found that, on average, a random walk discovers more results than a flooding mechanism that uses the same number of messages.

Though the random walk performs very well in terms of incurred query traffic, it can lead to poor performance in terms of query response times (delays). Several variants of the random walk have been proposed to address this. One of the first already appears in the early paper of Lv *et al.* [LCC+02]: it was shown experimentally, for both different topologies and different file availabilities, that running several random walks in parallel generates less traffic than an expanding ring, while having comparable performance in terms of delay.

Based on the observations of Lv *et al.*, Chawathe *et al.* [CRB+03] designed and implemented an unstructured peer-to-peer system in which queries are propagated through multiple independent random walks. The system also employs congestion control, as it biases a walk to avoid forwarding queries to peers that already sustain a high query traffic load. Chawathe *et al.* [CRB+03] showed experimentally that their system incurs lower traffic at peers compared to other mechanisms, including simple random walks and simple flooding.

In a different study, Gkantsidis *et al.* [GMS05] allow peers to index the files stored at their neighbours. This is equivalent to giving a random walk the ability to perform a one-step "look-ahead" at no additional message cost. The authors show that, in a random graph in which a few peers have a large ($\Theta(\sqrt{n})$) degree, a random walk with a one-step look-ahead can reduce the time to reach $\Omega(n)$ distinct peers from $\Theta(n)$ to $O(\sqrt{n})$, with high probability.

Puttaswamy *et al.* [PSZ08] prove a similar result over a two-tier system: if a super-peer sends its table of leaf contents to all other super-peers within a two-hop distance, a random walk over the super-peer network can discover all files in the system within a number of steps that is sub-linear (in terms of the total population, including leaf-peers). The authors prove

this by assuming that the top-level network formed by super-peers is a random regular graph, and by using the fact that the cover time of a random d-regular graph is $\Theta(n \log n)$, with high probability.

Terpstra *et al.* [TKLB07] and Gkantsidis *et al.* [GMS05] propose a query propagation based on message budgets, that generalizes the notion of the usual time-to-live mechanism used to constrain query traffic. Each query is initially assigned a total message budget, very similar to an initial time-to-live value. Queries can be forwarded to more than one neighbour, chosen at random. The query's budget is decreased by one by every peer receiving a query, and is split among the neighbours to which it is forwarded. This works similarly to a time-to-live mechanism, but allows greater flexibility. Splitting the budget unevenly is used to perform congestion control by Terpstra *et al.* [TKLB07]. Gkantsidis *et al* [GMS05] claim that a search can become more efficient by giving a higher budget to "well positioned" peers. For example, a query forwarded over a "bridge" edge, connecting two large subnetworks, should receive a higher budget than an edge leading back to the subnetwork.

The above search mechanisms are far more sophisticated than the mechanisms appearing in this thesis. Nonetheless, all of the analytical methods employed in the above works assume that the overlay graph is a static random graph, and that files are placed in the network uniformly at random. In this sense, these works do not capture system churn. In particular, neither the dynamic nature of the overlay graph nor the dynamic nature of the set of peers sharing a file is incorporated in the above models. In contrast, our model, presented in detail in Chapter 4, complements the above works by accounting for the time-variant behaviour of a real peer-to-peer system.

2.3.2 Search Mechanisms using Proactive Replication

An approach frequently used to reduce query traffic in unstructured peer-to-peer systems is to proactively replicate files [TKLB07, CS02, FRA+05, MBSM05, PSZ08]. This approach increases a file's availability, thereby also decreasing the traffic generated by queries for the file. On the other hand, there is a fundamental trade-off between query traffic and traffic created because of replication.

This trade-off was investigated by Terpstra *et al.* [TKLB07] and Ferreira *et al.* [FRA+05]. Terpstra *et al.* establish a relationship that describes the trade-off between query message traffic and the number of replicas necessary to guarantee query success [TKLB07]. In particular, in the system they propose, both queries and replicas are propagated through the budget-based

propagation mechanism presented in the previous section. Terpstra *et al.* show that, assuming that both replicas and queries are placed independently and uniformly in the network, the probability that a query succeeds is a function of the product $s \cdot r$, where s the budget of the query propagation and r the budget of the replica propagation.

For a network consisting of n peers, Ferreira *et al.* [FRA+05] show that if $\sqrt{n}$ replicas of each file are generated through a random walk, the number of query messages to locate a file will be $O(\sqrt{n \log n})$. This is also shown under the assumption that both replicas and queries are placed independently and uniformly among peers. Morselli *et al.* [MBSM05] again locate a file within $O(\sqrt{n \log n})$ by creating $\sqrt{n}$ replicas of a file. The authors assume that both file and peer identifiers can be hashed to a common key space. The replicas are then placed at peers whose identifiers (keys) are close to the key of the file being replicated. Subsequent walks propagating queries for a file are biased toward nodes with a similar key.

Typically, peers have storage capacity constraints, which limit the possible number of replicas that can be placed in a network. If the storage capacity constraints at each peer are known, an interesting optimization problem is how to replicate different files in order to minimize the aggregate query traffic (over all files). Intuitively, popular files that are requested very often should be replicated at more peers than unpopular files.

This question was studied in the context of the random walk by Cohen and Shenker [CS02]. Approximating again a random walk with uniform sampling, they showed that the aggregate traffic is minimized if a file is replicated in proportion to the square root of its request rate. More precisely, if a file i is requested with probability p_i, the average message cost per query is minimized if the number of replicas of the file is proportional to $\sqrt{p_i}$.

Using an operating point argument, Cohen and Shenker also show that square root replication can be achieved through a mechanism called *path replication*: if a file is back-propagated over the path that the random walk followed during the search, and each peer in this path caches the file, the replication rate of each file will be equal to the square root of their request rate.

Tewari and Kleinrock [TK06] extend this work by showing that, if the search mechanism is the expanding ring, it is optimal to replicate files proportionally to their request rates, as opposed to their square roots. Their work assumes that the overlay graph has a constant fan-out (*i.e.*, every peer set A has exactly $h|A|$ outgoing edges, where h is a constant).

Almost all of above works, with the exception of [MBSM05], approximate a random walk with uniform sampling, and assume that file copies are distributed uniformly at random. In this sense, our comments regarding the effect of churn in the system's behaviour appearing in the end of Section 2.3.1 also apply here. Extending the above results over a model that incorporates

churn, like ours, is an interesting open question; we discuss this in more detail in Sections 5.7 and 6.6.

2.3.3 Dealing with Absent Files

The search mechanisms based on passive and proactive replication discussed above are typically analyzed under the assumption that the file sought for is present in the system. As we discuss in Chapter 6, such mechanisms can lead to high query traffic if the file is unavailable. As a result, for all of the above mechanisms, reducing the traffic generated by queries for unavailable files is a fundamental open question.

One proposed approach to achieve this and, to the best of our knowledge, the only approach that has been presented in the literature, is to use an architecture complementing the unstructured peer-to-peer network with a structured system [LHH+04, LHSH04, CCR04, AC08c, ZK08]. In simple words, a relatively stable subset of the peers in the unstructured network forms a structured subnetwork. Queries for files that are very likely to be in the system are served by the entire unstructured peer-to-peer network. On the other hand, queries for rare files are directed to the structured peer-to-peer network. This takes advantage of the best of both structured and unstructured approaches: searching over the unstructured network gives good performance for widely available files, while searching through the structured network gives reliable, efficient handling of rare files, that may even not be present in the system.

One challenge in the above approach is how to decide, at the time a query is issued, whether a file is widely available or rare. Depending on the outcome of this decision, a peer issuing a query should propagate it either through the unstructured or the structured network, respectively.

In [LHH+04, LHSH04], each peer monitors its local query traffic, keeping track of popular terms and inferring file availability from term popularity. This has a potential pitfall: as discussed in Section 2.2.3, files requested often are not necessarily also replicated often. Zaharia *et al.* [ZK08] propose a gossipping mechanism for obtaining global statistics on the availability of shared files. In particular, each peer constructs a *synopsis* of its available files, which includes the name of each file and a randomly generated number, called the CT value. Peers periodically send their synopsis to a random neighbour. Moreover, a peer receiving a synopsis merges it with its own; if a file appears in both synopses, its CT value is set to be the maximum of the two. After many rounds, files with many replicas are more likely to have higher CT values, and thus the synopses resemble (randomized) histograms of the availability of files. To

keep synopses short, files with small CT values can be dropped.

The mechanism proposed in our work for dealing with files not in the system is simpler than the ones discussed above. First, it does not require an additional infrastructure for serving queries for rare files —it guarantees the scalability of the system irrespectively of whether a file is rare or widely available. Second, it does not require to proactively collect and maintain information about file availability: As we will see in Chapter 6, using the "absence of evidence" as "evidence of absence", the system automatically adapts to the presence or absence of files, while guaranteeing that query traffic is low.

2.4 Summary

The connection protocol used by peers in unstructured peer-to-peer systems leads to overlay graphs whose degree distributions are concentrated around a constant value, that does not depend on the system size. In all other respects, the overlay graph may be arbitrary; in this sense, the main challenge in unstructured systems, as discussed in Section 2.3, is to design efficient query propagation mechanisms that cope precisely with the arbitrary nature of the overlay graph.

Several such mechanisms have been proposed; nonetheless, approaches to analyze such mechanisms have mostly focused on static random graphs, or have eschewed modelling the overlay graph altogether. The work presented in the subsequent chapters of this thesis aims at addressing this issue, by encompassing the variability of an unstructured overlay in the system's model.

Chapter 3

Random Walks and Expander Graphs

The goal of this thesis is to characterize the performance of searching in unstructured peer-to-peer systems. In particular, we wish to understand when a query propagation mechanism is scalable (*i.e.*, the query traffic it incurs does not grow significantly with the peer population size) and when it is reliable (*i.e.*, files that are present in the system can be located with reasonable guarantees).

Our results focus on two query propagation mechanisms, namely, the random walk and the expanding ring. To formally assess the scalability of these mechanisms, we need to be able to compute the number of message transmissions required to locate a file during a search. Similarly, to assess their reliability, we need to be able to to compute the probability that a given search is successful in locating a file. This chapter is a technical preliminary: it introduces the mathematical tools that will allow us to relate the above two quantities to (a) the overlay network formed by peers and (b) the number copies of the requested file, at the time a search takes place.

The remainder of this chapter is organized as follows. First, in Section 3.1, we give a formal definition of a random walk on a finite graph. We define two different versions: the first is a discrete-time version, where the time between transitions of the walk is constant, and the second is a continuous-time version, where the time between transitions is exponentially distributed. This formal presentation also allows us to briefly review a few basic concepts from the theory of Markov chains and Markov processes. This is useful, as Markov chains and Markov processes appear frequently throughout our analysis.

In Section 3.2, we introduce and define the relaxation time of a graph. We provide several bounds that establish a relationship between this quantity and the hitting time of a random walk; the latter, viewed from the perspective of a random walk as a query propagation mechanism, is

precisely the number of message transmissions required to find a file. As a result, the bounds in Section 3.2 can be used to characterize the probability a random walk succeeds in locating a file, as well the expected number of message transmissions this requires.

In Section 3.3, we discuss the fundamental properties of expanders, *i.e.*, graphs with a small relaxation time. It is a well known fact that such graphs are abundant: as we will see in Section 3.3.2, almost all regular graphs are expanders. This will play an important role in our analysis, not least because, as already stated in Chapter 2, the overlay graph of an unstructured network is "almost" regular. Moreover, having a small relaxation time will be especially useful in light of the fact that our bounds on the hitting time (*i.e.*, the number of transmissions required to locate a file) depend on this quantity.

In Section 3.3.3, we introduce the expansion ratio of a graph and explore its properties. This quantity will play a similar role in our analysis of the expanding ring mechanism to the one played by the relaxation time in our analysis of the random walk. In particular, in Chapter 5, we will use the results of Section 3.3.3 to relate the expansion ratio of the overlay graph to both the probability that the expanding ring locates a file as well as the number of message transmissions required to do so (see also Section 5.5).

3.1 Random Walks on Finite Graphs

In this section, we give a brief review of some fundamental properties of random walks on finite graphs. All the results mentioned in this section are classic; our main reference on random walks is the book by Aldous and Fill [AF], while our main reference for Markov chains and Markov processes is the book by Gallager [Gal96].

3.1.1 Random Walks and Markov Chains

A random walk on a graph captures the movement of a particle —the "walker"— that starts from a vertex of the graph and moves to a neighbour chosen uniformly at random. This process is repeated for several steps; at each step, the walker moves to a neighbouring vertex, chosen again uniformly at random. The neighbour selection is independent of the path followed by the walker prior to its arrival at the current vertex. In particular, the walker has no memory of the vertices it has visited so far, and may visit the same vertex more than once.

In this thesis, we use the random walk as a mechanism for propagating queries over a peer-to-peer network. In particular, the graph on which the walk takes place is the overlay network

formed by peers, and the transition of the walker over an edge corresponds to the transmission of a message along the link between two adjacent peers. When viewed as such a query propagation mechanism, the random walk is appealing due to its inherent simplicity: forwarding decisions are based only on "local" information (namely, each peer's list of neighbours). No additional state information needs to be maintained by peers or transmitted along with the message.

Random Walk on an Unweighted Graph

To formally define a random walk, consider a simple, undirected graph $G(V, E)$, whose vertex set V is finite and whose edge set is E. Recall that the *degree* d_i of vertex $i \in V$ is the number of edges incident to i, *i.e.*,

$$d_i = |\{(i, j) \in E\}|.$$

A *discrete-time random walk* —or, simply, a *random walk*— on G is a sequence of random variables $X(0), X(1), X(2), \ldots$ such that

$$\begin{aligned} p_{ij} &\equiv \mathbf{P}(X(t+1)=j \mid X(t)=i) \\ &= \mathbf{P}(X(t+1)=j \mid X(t)=i, X(t-1)=k_{t-1}, \ldots, X(0)=k_0) \\ &= \begin{cases} \frac{1}{d_i}, & \text{if } (i,j) \in E, \\ 0, & \text{otherwise} \end{cases} \qquad \text{for all } t \geq 0 \text{ and all } i, j \in V. \end{aligned} \tag{3.1}$$

In other words, the random walk $\{X(t)\}_{t\in\mathbb{N}}$ is a *Markov chain* with state space V and transition probabilities p_{ij}, $i, j \in V$, given by (3.1). The second equality in (3.1) indicates that the random walk has the *Markov property*: conditioned on the current vertex $X(t)$, the next vertex $X(t+1)$ is independent of $X(0), X(1), \ldots, X(t-1)$. Finally, the last equality implies that the next vertex is chosen uniformly at random among the neighbours of the current vertex.

To fully define the sequence $\{X(t)\}_{t\in\mathbb{N}}$, in addition to the transition probabilities in (3.1), we also need to determine $X(0)$, the vertex from which the walk starts —*i.e.*, the initial state of the Markov chain. For $i \in V$, we denote by $\mathbf{P}_i(\cdot)$ the probability (of some event) given that the walk starts from vertex i (*i.e.*, $X(0) = i$). Moreover, for ρ a probability distribution over V, we denote by $\mathbf{P}_\rho(\cdot)$ the probability given that $X(0)$ is distributed according to ρ.

Random Walk on a Weighted Graph or a Multi-Graph

Though our focus is on random walks on unweighted graphs, our analysis will occasionally require results from a more general setting, namely, random walks on weighted graphs. In

such walks, the choice of the next vertex may be biased. In particular, assume that the edges of the graph $G(V, E)$ are assigned weights according to a positive weight function

$$w : E \to \mathbb{R}_+,$$

that maps

$$(i, j) \mapsto w_{ij} > 0, \quad (i, j) \in E.$$

Moreover, let

$$w_i = \sum_{(i,j)\in E} w_{ij} \tag{3.2}$$

be the total weight of all edges incident to i. Then, a random walk on the weighted graph G is a Markov chain $\{X(t)\}_{t\in\mathbb{N}}$ with state space V and transition probabilities

$$p_{ij} = \begin{cases} \frac{w_{ij}}{w_i}, & \text{if } (i, j) \in E, \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

In other words, in random walks over a weighted graph, a "heavier" edge is more likely to be followed than a "lighter" edge.

Note that we do not allow the weights w_{ij} to be zero. A walk on a graph with zero-weight edges can be reduced to the above definition by removing such edges from the edge set E, effectively treating them as "absent". Moreover, the random walk on an unweighted graph G follows from the above (more general) definition by assuming that all edges have the same (positive) weight (*e.g.*, $w_{ij} = 1$ for all $(i, j) \in E$).

The definition of random walks can be easily extended to multi-graphs (*i.e.*, graphs in which the E is a multi-set). However, any random walk on an unweighted multi-graph MG is equivalent to a random walk on a simple, weighted graph G. Graph G can be constructed from the multi-graph MG as follows. First, every simple edge of MG is added to G and a weight equal to one is assigned to it. Second, for any multi-edge in MG with multiplicity higher than one, the corresponding simple edge is added to G, and is assigned a weight equal to the multiplicity of the multi-edge.

A similar construction reduces a random walk on a weighted multi-graph to a random walk over a simple, weighted graph: the weight of a simple edge is set to be equal to the sum of the weights of the corresponding multi-edges it replaces. For this reason, we only discuss random walks on weighted, simple graphs, keeping in mind that random walks on multi-graphs are equivalent to such walks.

Stationary Distribution and Ergodicity

Given a Markov chain with space V, a distribution π over V is called *stationary* if it satisfies the following system of equations:

$$\pi P = \pi \tag{3.4}$$

where $P = [p_{ij}]_{i,j \in V}$ is the transition matrix of the chain. Equations 3.4 are called the *balance equations* of the chain. A stationary distribution has the property that

$$\mathbf{P}_\pi(X(t) = j) = \pi_j, \qquad \text{for all } t \geq 0, j \in V.$$

In other words, if $X(0)$ is distributed according to π, so is $X(t)$, $t \geq 1$. This is a consequence of the Markov property: The matrix $P^{(t)} = [\mathbf{P}_i(X(t) = j)]_{i,j \in V}$ can be written as

$$P^{(t)} = \underbrace{P \cdot P \cdot \ldots \cdot P}_{t \text{ times}}$$

and, hence, by the definition of the stationary distribution

$$\pi P^{(t)} = \pi, \qquad \text{for all } t \geq 0.$$

Below, we discuss the conditions under which a stationary distribution exists.

In a Markov chain with state space V, a state $j \in V$ is *accessible* from state $i \in V$ if there is a finite number of steps after which the chain starting at i reaches j with positive probability, *i.e.*,

$$\exists\, t \geq 0 \text{ such that } \mathbf{P}_i(X(t) = j) > 0.$$

A Markov chain is *irreducible* if j is accessible from i for all pairs $i, j \in V$. The *period* of a state i is the greatest common divisor of the times $t \in \mathbb{N}$ for which $\mathbf{P}_i(X(t) = i) > 0$. Intuitively, a chain starting at i returns back to i only at times t that are multiples of i's period. A chain is *aperiodic* if all states have period one. If the Markov chain is irreducible, all states in V have the same period [Gal96]. A chain that is both irreducible and aperiodic is called *ergodic*.

Ergodic chains have a unique stationary distribution, which is also the "steady state" distribution, *i.e.*, the limit distribution of the chain for large t.

Theorem 3.1 (Ergodic Theorem for Markov Chains [Gal96,AF])**.** *Consider an ergodic Markov chain $\{X(t)\}_{t \in \mathbb{N}}$ with state space V and transition probabilities p_{ij}, $i, j \in V$. Then, there exists*

a probability distribution π over V such that, for all $i \in V$ and for all initial states $j \in V$,

$$\pi_i = \lim_{t\to\infty} \mathbf{P}_j(X(t) = i) = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t} \mathbb{1}_{X(\tau)=i}, \tag{3.5}$$

*where the last equality holds almost surely (*a.s.*). Moreover, π is the unique (up to a multiplicative factor) solution of the balance equations* (3.4).

The existence of the first limit in (3.5) implies that, for large t, the distribution of $X(t)$ converges to a limit distribution. Moreover, the second equality in (3.5) implies that, for each $i \in V$, the limit probability π_i is asymptotically equal to the fraction of times the chain is in state i. Finally, the last statement of the theorem implies that the limit distribution is also the unique stationary distribution of the chain: if $X(0)$ is distributed according to π, so are $X(t)$, $t \geq 1$.

A random walk on a weighted (undirected) graph is irreducible if and only if the graph is connected. Moreover, it is aperiodic if and only if the graph is non-bipartite. Hence, the random walk on G is ergodic if and only if G is connected and non-bipartite. If the random walk is ergodic, the unique stationary probability distribution of the walk is

$$\pi_i = \frac{w_i}{\sum_{j\in V} w_j}, \qquad i \in V. \tag{3.6}$$

This can be shown by checking that (3.6) indeed satisfies the balance equations (3.4); uniqueness is implied by the ergodic theorem.

In the unweighted case, (3.6) becomes

$$\pi_i = \frac{d_i}{\sum_{j\in V} d_j} = \frac{d_i}{2|E|}, \qquad i \in V. \tag{3.7}$$

An (unweighted) graph is called *regular* if all its vertices have the same degree. For $d \geq 1$, a graph is called *d-regular* if all its vertices have degree d. Note that, if G is unweighted and regular, the distribution (3.7) is the uniform distribution over V.

Theorem 3.3, presented below, implies that the necessary and sufficient condition for the stationary distribution given by (3.6) to be unique is that graph G is connected. However, if G is bipartite, the first limit of (3.5) does not exist: the set of vertices visited by the walker at odd times is distinct from the set visited at even times. In this sense, the walk never reaches a "steady state". If, on the other hand, the graph is disconnected, the system of equations (3.4) has more than one linearly independent solutions (as many as its connected components). Moreover, both limits in (3.5) depend on the initial state $X(0)$, as the walker never leaves the connected component from which it starts.

All irreducible random walks on weighted graphs are *reversible*: the stationary distribution π, given by (3.6), and the transition probabilities p_{ij} satisfy the equations

$$\pi_i p_{ij} = \pi_j p_{ji}, \qquad \text{for all } i, j \in V. \tag{3.8}$$

It is interesting to note that, *any* irreducible reversible chain can be represented as a random walk on a weighted, connected graph (possibly with self-loops) [AF]. To see this, let p_{ij}, $i, j \in V$, be the transition probabilities of an irreducible, reversible Markov chain, and let π be any positive solution of $\pi P = \pi$. Then, this chain is identical to a random walk on a graph G in which the weight of edge (i, j) is $\pi_i p_{ij}$.

In other words, the set of irreducible, reversible Markov chains is identical to the set of random walks on weighted, connected graphs, and, in particular, the set of ergodic reversible Markov chains is identical to the set of ergodic random walks. Therefore, the theory presented in this chapter, pertaining to random walks on graphs, can also be interpreted as a theory over reversible Markov chains.

3.1.2 Continuous-Time Random Walks and Markov Processes

In a continuous-time random walk [AF], the walker moves again from a vertex of the graph to one of its neighbours according to (3.3). However, the time that the walker spends on each vertex is not deterministic: it is exponentially distributed with mean one. There are several reasons why we are interested in continuous-time random walks. The most important is technical. As we discuss in Section 3.2, several quantities that will be of interest in this thesis (*e.g.*, the distributions of hitting times) are easier to express for continuous-time random walks. Moreover, as we discuss below, the ergodic theorem applies to continuous-time random walks in a wider class of graphs than its discrete-time equivalent. Finally, continuous-time walks are of particular interest in the context of this thesis because, in a peer-to-peer system, the time required to transmit a message may not necessarily be deterministic. In this sense, continuous-time walks are more appropriate for modelling the propagation of queries in a such a system.

Markov Processes

Before we formally define a continuous-time random walk, we briefly review some fundamental properties of Markov processes. A *Markov process* [Gal96] $\{Y(t)\}_{t \in \mathbb{R}_+}$ with state space V and transition rates $q_{ij} \geq 0$, $i, j \in V$, is defined as follows: Given that the process is in

state i, the time until the next transition is (a) independent of earlier transition epochs and (b) exponentially distributed with rate

$$q_i = \sum_{k \in V} q_{ik}.$$

Moreover, given that current state is i, the probability that the next state is j is

$$p_{ij} = \frac{q_{ij}}{q_i}. \tag{3.9}$$

I.e., if one ignores the time between transitions, the states visited are described by a Markov chain $\{X(t)\}_{t \in \mathbb{N}}$ with transition probabilities given by (3.9). The chain $\{X(t)\}_{t \in \mathbb{N}}$ is called the *embedded Markov Chain* of the Markov process $\{Y(t)\}_{t \in \mathbb{R}_+}$.

We define the transition matrix Q of a Markov process $\{Y(t)\}_{t \in \mathbb{R}_+}$ as the matrix whose i, j-th element is

$$\begin{cases} q_{ij}, & \text{if } i \neq j \\ q_{ii} - q_i & \text{if } i = j. \end{cases} \tag{3.10}$$

Note that, in general, the aggregate transition rates q_i from each state $i \in V$ may be different. When all aggregate rates q_i are equal, the Markov process is called *uniformized* [Gal96]. Uniformized Markov processes can be related to their embedded Markov chains through the following formula:

$$Y(t) = X(N(t))$$

where $N(t)$ is a Poisson process with rate $q = q_i$, $i \in V$.

Definition of Continuous-Time Random Walk

Formally, a continuous-time random walk on a weighted graph $G(V, E)$ is a Markov process $\{Y(t)\}_{t \in \mathbb{R}_+}$ with state space V and transition rates

$$q_{ij} = \begin{cases} \frac{w_{ij}}{w_i}, & \text{if } (i, j) \in E, \\ 0, & \text{otherwise,} \end{cases}$$

where $w : E \to \mathbb{R}_+$ is the weight function and

$$w_i = \sum_{(i,j) \in E} w_{ij}$$

is the total weight of edges incident to i, as in (3.2). Note that, by definition, the continuous-time random walk on a graph G is uniformized, as the aggregate transition rate q_i from every state is one. Moreover, the corresponding embedded Markov chain is none other than the (discrete-time) random walk on G.

Stationary Distribution and Ergodicity

Let again Q be the transition matrix of a Markov process, as defined by (3.10). A distribution π over V is called stationary if it satisfies the following system of equations

$$\pi Q = 0, \tag{3.11}$$

which we call the balance equations of the Markov process. A stationary distribution of a Markov process again has the property that

$$\mathbf{P}_\pi(Y(t) = j) = \pi_j, \qquad \text{for all } t \in \mathbb{R}_+, j \in V.$$

In other words, if $Y(0)$ is distributed according to π, so will $Y(t)$ for all $t > 0$. Showing this is not as simple as in the case of Markov chains: we refer the reader to Gallager [Gal96, Section 6.2, Chapter 6, pp. 192-195] for a proof of this statement.

An important property of uniformized Markov processes is the following: it can be easily verified that, if all aggregate rates are identical, any solution of the balance equations of the embedded Markov chain is also a solution of the balance equations of the Markov process, and vice versa [Gal96, Section 6.3, Chapter 6, pp. 195-196]. As a result, if the embedded Markov chain has a stationary distribution, this distribution will also be a stationary distribution of the Markov process —which is not true, in general, for non-uniformized processes.

A Markov process is called ergodic if and only if its embedded chain is irreducible. The ergodic theorem extends to Markov processes as follows.

Theorem 3.2 (Ergodic Theorem for Markov Processes [Gal96, AF])**.** *Consider an ergodic Markov process $\{Y(t)\}_{t\in\mathbb{R}_+}$ with state space V and transition rates q_{ij}, $i, j \in V$. Then, there exists a probability distribution π over V such that, for all $i \in V$ and for all initial states $j \in V$,*

$$\pi_i = \lim_{t\to\infty} \mathbf{P}_j(Y(t) = i) = \lim_{t\to\infty} \frac{1}{t} \int_{\tau=0}^{t} \mathbb{1}_{Y(\tau)=i} d\tau, \tag{3.12}$$

*where the last equality holds almost surely (*a.s.*). Moreover, π is the unique (up to a multiplicative factor) solution of balance equations* (3.11).

A continuous-time random walk on a graph G is ergodic if and only if the graph G is connected. In particular, the limit distribution of a continuous-time random walk exists even if G is bipartite, and the corresponding discrete-time walk on G is not aperiodic. Moreover, because continuous-time random walks are uniformized, any solution of the balance equations

(3.11) of the Markov process is also a solution of the balance equations (3.4) of the embedded discrete-time random walk. As a result, the stationary distribution π is again given by (3.6), *i.e.*,

$$\pi_i = \frac{w_i}{\sum_{j \in V} w_j}, \qquad i \in V.$$

The above already illustrates an inherent advantage of the continuous-time random walks over traditional, discrete-time walks: "continuizing" [AF] a walk allows us to consider a wider class of graphs while still maintaining ergodicity. We discuss additional advantages in the next section.

3.2 Relaxation Time and Random Walks

In this section, we introduce the relaxation time of a graph, a quantity that determines, in many different ways, the behaviour of a random walk on the graph. Again, though our focus will be on unweighted graphs, we introduce the theory in the (more general) weighted setting. Our main reference for the relaxation time and its relationship to random walks is Aldous and Fill [AF]: all the statements presented below can either be found in this book or be derived from results appearing in it. For statements of the latter category, the derivation is included in the present text.

3.2.1 Graph Spectrum

Consider a connected graph $G(V, E)$, paired with a positive weight function $w : E \to \mathbb{R}_+$. Let again p_{ij}, $i, j \in V$, be the transition probabilities of the random walk on G, defined by (3.3), and denote by $P = [p_{ij}]_{i,j \in V}$ the corresponding transition matrix.

The following can be stated about the spectrum of the transition matrix P.

Theorem 3.3 (Perron-Frobenius, [Gal96])**.** *Let P be the transition matrix of an irreducible Markov chain. Then the following hold:*

1. $\lambda = 1$ *is an eigenvalue of P and has multiplicity one,* i.e.*, there exists a unique (up to a multiplicative factor) eigenvector ν such that*

$$\nu P = \nu.$$

Moreover, $\nu > 0$, i.e.*, all entries of ν are positive.*

2. *For any other eigenvalue λ' of P, $|\lambda'| \leq \lambda$.*

Theorem 3.3 implies the existence and uniqueness of the stationary distribution π of a random walk on a graph G provided that the graph G is connected. The stationary distribution is again given by (3.6), and it is positive on all vertices in V. Moreover, Theorem 3.3 implies that all eigenvalues of P have a magnitude that is less than or equal to one. The fact that the random walk on G is reversible, also implies that all eigenvalues of P are real:

Lemma 3.1. *Let P be the transition matrix of an irreducible, reversible Markov chain. Then, all eigenvalues of P are real.*

The proof can be found in Chapter 3, Section 4, pp. 16–19, of Aldous and Fill [AF]. The above lemma, along with Theorem 3.3, imply that the eigenvalues of P can be ordered as follows

$$1 = \lambda_1 > \lambda_2 \geq \ldots \geq \lambda_n \geq -1$$

where $n = |V|$. The fact that $\lambda_1 \neq \lambda_2$ is implied by the fact that, if G is connected, the left eigenvector is unique (up to a multiplicative factor) and, therefore, the eigenvalue $\lambda_1 = 1$ has multiplicity one.

It is important to note that the graph G is bipartite if and only if $\lambda_n = -1$ [AF, Chu97]. In particular, if $\lambda_n > -1$, then the random walk on G is ergodic, and the stationary distribution is also the limit distribution of the random walk.

The difference $1 - \lambda_2$, between the first and second largest eigenvalue of the transition matrix P, is called the *spectral gap*[1] of the graph G. The *relaxation time* τ_G of the graph G [AF] is defined as the inverse of its spectral gap:

$$\tau_G = \frac{1}{1 - \lambda_2(G)}. \tag{3.13}$$

If G is disconnected, we define the relaxation time τ_G to be infinite. This is consistent with (3.13) as, for a disconnected graph, the multiplicity of eigenvalue 1 is greater than one and, hence, $\lambda_2 = 1$.

The relaxation time τ_G is a quantity that will have a recurring role throughout our analysis. In the following, we present several well-known results that establish the strong relationship between the relaxation time and the random walk on G.

[1] We adopt the notation of Aldous and Fill [AF] and Chung [Chu97], in which the eigenvalues of the transition matrix P are used in defining the spectral gap, rather than the eigenvalues of the adjacency matrix of G [HLW06]. This simplifies the isoperimetric inequalities appearing in Section 3.3.3.

3.2.2 Mixing Time

Perhaps the most fundamental property of the relaxation time is that it quantifies how "quickly" the distribution of the random walker converges to the steady state, if the latter exists. Suppose again that the graph G is connected, and that P is the transition matrix of the random walk on G. Recall that, by Theorem 3.3, a stationary distribution π exists, given by (3.6).

Consider the continuous-time random walk $\{Y(t)\}_{t\in\mathbb{R}_+}$ on G, and let

$$p_{ij}^{(t)} = \mathbf{P}_i(Y(t) = j), \quad i, j \in V$$

be the probability that the random walk starting from i is in vertex j at time t. Let

$$D_i(t) = \sum_{j\in V} |p_{ij}^{(t)} - \pi_j| = ||\mathbf{P}_i(Y(t) = \cdot) - \pi||_1$$

be the l_1 distance of the distribution of the random walk after t steps from the stationary distribution[2] and

$$D(t) = \max_{i\in V} D_i(t),$$

i.e., the maximum distance from the stationary distribution over all possible initial vertices. The quantity $D(t)$ is a decreasing function [AF, Lemma 20, Chapter 2, pp. 16], and, by the ergodic theorem, converges to zero as t tends to infinity. We are interested in computing how quickly this convergence takes place. In particular, given some $\epsilon > 0$, we are interested in

$$\tau_\epsilon = \inf\{t : D(t) \leq \epsilon\}.$$

Typically, the *mixing time* of the random walk is defined as τ_ϵ for some given constant $0 < \epsilon < 1$ (*e.g.*, Aldous and Fill [AF] use $\epsilon = e^{-1}$). The following theorem then holds

Theorem 3.4. *Let G be a connected, weighted graph. Then,*

$$\tau_\epsilon \leq \tau_G \left(-\log \epsilon + \frac{1}{2} \log \frac{1}{\pi_*} \right) \tag{3.14}$$

where $\pi_* = \min_{i\in V} \pi_i$ and τ_G is the relaxation time of the graph.

Theorem 3.4 is essentially Lemma 23 in Chapter 4 of Aldous and Fill [AF]. A similar statement can be made about the discrete-time random walk (see [HLW06, Section 3.1.1, Chapter 3, pp. 26]), if one substitutes

$$\max(\lambda_2, |\lambda_n|)$$

[2]The *variation distance* between the two distributions is equal to $\frac{1}{2}D_i(t)$ [AF, Lemma 19, Chapter 2, pp. 15].

for λ_2 in the definition (3.13) of the relaxation time. This discrepancy between the two versions of the random walk is due to the sensitivity of the discrete-time walk on the bipartiteness of the graph. For example, graphs that are "almost" bipartite (*i.e.*, can become bipartite after the removal of only a few, low-weight edges) will have a last eigenvalue λ_n close to -1: this is true because the eigenvalue λ_n is a locally-Lipschitz function of the edge-weights of the graph. This "almost"' bipartiteness may affect the convergence of the discrete-time walk: the "memory" of which of the two "almost" bipartitions the walk starts from takes a long time to die out, and this is captured by the dependence of the mixing time to λ_n. On the other hand, it has no effect in continuous-time.

In conclusion, small relaxation times imply fast convergence of the (continuous-time) random walk to the stationary distribution. Hence, continuous-time random walks on graphs that have small relaxation times "mix" fast. The same can be said about the corresponding discrete-time random walks, provided that $|\lambda_n| < \lambda_2$.

3.2.3 Hitting Times

The relaxation time is also related to the time required for a random walk to reach a subset of the vertices $A \subset V$. This quantity, formally called the hitting time of set A, is of particular interest in the context of our work. From our perspective of the graph as a peer-to-peer network, assuming that the set A is the set of peers that have the copy of a file, the hitting time expresses the number of query messages needed to find the file, if queries are propagated according to a random walk.

The results on hitting times we present here agree with our main observation regarding the relationship between the relaxation time and the "mixing" time. In graphs with a small relaxation time, random walks converge quickly to their stationary distribution; the same can be said, in general, about hitting times: a small relaxation time also implies small hitting times.

Consider a random walk on a weighted, connected graph G, and let π be the stationary distribution of the random walk on G. Let again $\{X(t)\}_{t\in\mathbb{N}}$ be the random walk on G and $\{Y(t)\}_{t\in\mathbb{R}_+}$ the continuous-time random walk on G. In the following, we denote by $\mathbb{E}_i[\cdot]$, $i \in V$, the expectation of a random variable, given that the random walk $\{X(t)\}_{t\in\mathbb{N}}$ (or, $\{Y(t)\}_{t\in\mathbb{R}_+}$) starts at state i. Similarly, we denote by $\mathbb{E}_\rho[\cdot]$ the expectation of a random variable given that $X(0)$ (or $Y(0)$) is distributed according to the probability distribution ρ.

Given a set $A \subset V$, the *hitting time* of set A is the random variable

$$T_A = \inf\{t \in \mathbb{N} : X(t) \in A\}.$$

Similarly, the hitting time of set A for the continuous-time walk can be defined as

$$\tilde{T}_A = \inf\{t \in \mathbb{R}_+ : Y(t) \in A\}.$$

Although these two random variables have distinct probability distributions, they have the same expectation:

Lemma 3.2. *For any initial distribution ρ on V*

$$\mathbb{E}_\rho[\tilde{T}_A] = \mathbb{E}_\rho[T_A]$$

Proof. A proof of the lemma can be found in Section 5.3 of Chapter 2 in Aldous and Fill [AF]. We give an alternate proof, based on Wald's equality.

Recall that

$$Y(t) = X(N(t))$$

where $\{N(t)\}_{t\in\mathbb{R}_+}$ is a Poisson process with rate one. Let $\{S_n\}_{n\in\mathbb{N}}$ be the arrival process associated with the counting process $\{N(t)\}_{t\in\mathbb{R}_+}$, such that

$$S_n \leq t \text{ if and only if } N(t) \geq n.$$

Then,

$$\tilde{T}_A = S_{T_A}.$$

The random variable T_A is a stopping rule on the joint probability space of $\{S_n, X(n)\}_{n\in\mathbb{N}}$, hence, by Wald's equality [Gal96, Theorem 3, Chapter 3, pp. 66]

$$\mathbb{E}_\rho[\tilde{T}_A] = \mathbb{E}[T_A] \cdot \bar{Y}$$

where $\bar{Y} = \mathbb{E}[S_{n+1} - S_n] = 1$ is the mean time between two transitions. □

The following theorem establishes a relationship between the expected hitting time of a set and the relaxation time of the graph. If π the stationary distribution of the random walk and $A \subseteq V$, we define the distribution π_A as

$$(\pi_A)_i = \begin{cases} \frac{\pi_i}{\pi(A)}, & i \in A \\ 0, & i \notin A, \end{cases}$$

where

$$\pi(A) = \sum_{j\in A} \pi_j.$$

The distribution π_A can be seen as the stationary distribution, conditioned on the event that the random walk is in a state in A. Theorem 3.5 indicates that, conditioned on $X(0)$ starting outside A, the expected hitting time of a set A can be bounded in terms of $\pi(A)$ and τ_G in a simple way.

Theorem 3.5 ([AF]). *Let $G(V, E)$ be a connected, weighted graph and $A \subset V$ a non-empty set of vertices. Then, for $A^c = V \setminus A$,*

$$\frac{1}{\pi(A)} - 1 \leq \mathbb{E}_{\pi_{A^c}}[T_A] \leq \frac{\tau_G}{\pi(A)} \tag{3.15}$$

where τ_G is the relaxation time of G.

Proof. The theorem is proved by Aldous and Fill [AF, Lemma 17, Chapter 3, pp. 24] for irreducible, reversible chains, for the case where A is a singleton. The case of general sets A can be reduced to the singleton case by considering a *collapsed chain* [AF, Chapter 2, Section 7.3, pp. 26-27], in which the set A is collapsed to a single state α. Formally, the collapsed chain is a chain with state space $A^c \cup \{\alpha\}$, and transition probabilities

$$\begin{aligned}
p'_{ij} &= p_{ij}, \quad i, j \in A^c \\
p'_{i\alpha} &= \sum_{k \in A} p_{ik}, \quad i \in A^c \\
p'_{\alpha i} &= \frac{1}{\pi(A)} \sum_{k \in A} \pi_k p_{ki}, \quad i \in A^c \\
p_{\alpha\alpha} &= \frac{1}{\pi(A)} \sum_{k \in A} \sum_{l \in A} \pi_k p_{kl}
\end{aligned}$$

If the original chain is irreducible and reversible, so will the collapsed chain. Moreover, the collapsed chain has a steady state distribution π' such that

$$\pi'_\alpha = \pi(A) \quad \text{and} \quad \pi'_j = \pi_j, j \in A^c.$$

The theorem therefore follows[3] by applying the result on the collapsed chain and using the contraction principle [AF, Proposition 44, Chapter 4, pp. 37], according to which the relaxation time of the collapsed chain is no greater than τ_G. □

Recalling that, if G is a regular graph, the stationary distribution is uniform, we obtain the following corollary:

[3] An alternate proof of the upper bound can be derived from Corollary 34 and equation (87), in page 42 of Chapter 3 in Aldous and Fill [AF]. See also Proposition 21 for page 28 of the same chapter. Here, τ_G is defined differently but still refers to the same quantity.

Corollary 3.1. *Let $G(V, E)$ be a connected, unweighted, regular multi-graph and $A \subset V$ a non-empty subset of vertices. Then, for $A^c = V \setminus A$ and u_{A^c} the uniform distribution over A^c,*

$$\frac{n}{|A|} - 1 \leq \mathbb{E}_{u_{A^c}}[T_A] \leq \frac{\tau_G n}{|A|} \tag{3.16}$$

where τ_G is the relaxation time of G and $n = |V|$.

There are several interesting observations regarding the above bounds. First, by Lemma 3.2, both Theorem 3.5 and Corollary 3.1 hold for the discrete-time as well as the continuous-time random walk on G. Second, smaller relaxation times make the upper-bound on (3.15) tighter. This agrees with the intuition we mentioned earlier that, the smaller the relaxation time, the faster the random walk will hit set A. A final important observation is that both bounds in (3.16) do not depend on the set A itself, but on its cardinality. In other words, (3.16) bounds the hitting time based on *how many* vertices are in A, as opposed to *which vertices* are in the set.

In the case of the continuous-time random walk, we can give a more detailed description of the hitting time. In particular, the following theorem states that hitting times are "almost" exponentially distributed[4]. This statement is not true for the discrete-time walk[5]; as a result, Theorem 3.6 presents an additional advantage of considering a continuous-time random walk over a discrete-time version.

Theorem 3.6 ([AF])**.** *Let $G(V, E)$ be a connected, weighted graph and $A \subset V$. Then, for $A^c = V \setminus A$,*

$$(1 - 2\tau_G \pi(A))\, e^{-2\pi(A)t} \leq \mathbf{P}_{\pi_{A^c}}(\tilde{T}_A > t) \leq e^{-\frac{\pi(A)t}{\tau_G}} \tag{3.17}$$

where τ_G is the relaxation time of G.

Proof. The upper bound is an immediate corollary of Eq. (68) in Proposition 21 of Chapter 3 of Aldous and Fill [AF] (see remark (b) on the proposition, page 29 of the same chapter).

The lower bound can also be derived from a result in Aldous and Fill: Theorem 43 [AF, Chapter 3,pp 56] states that

$$P_\pi(\tilde{T}_A > t) \geq \left(1 - \frac{\tau_G}{\mathbb{E}_{\alpha_A}[\tilde{T}_A]}\right) e^{-\frac{t}{\mathbb{E}_{\alpha_A}[\tilde{T}_A]}}, \qquad t > 0 \tag{3.18}$$

[4]See also Proposition 23, pp. 30 of Chapter 3, in Aldous and Fill [AF]. This result is of a similar nature, as it bounds the L_∞ distance of the c.d.f. of the hitting time from the c.d.f. of an exponential r.v.

[5]See *e.g*, Chapter 2, Section 4.3, pp. 19-20, of Aldous and Fill [AF].

where α_A the *quasi-stationary distribution* on A^c [AF, Section 6.5, chapter 3, pp. 39-42]. From Eq. (87), Chapter 3, pp. 42 of [AF],

$$\mathbb{E}_{\alpha_A}[\tilde{T}_A] \geq \mathbb{E}_{\pi_{A^c}}[T_A].$$

The lower bound therefore follows from (3.18) and the fact that

$$\mathbb{E}_{\pi_{A^c}}[T_A] \geq \frac{1}{2\pi(A)} \tag{3.19}$$

for all non-empty A. To see this, observe that, for $\pi(A) \geq \frac{1}{2}$,

$$\mathbb{E}_{\pi_{A^c}}[T_A] \geq 1 \geq \frac{1}{2\pi(A)},$$

while, for $\pi(A) < \frac{1}{2}$,

$$\frac{1}{\pi(A)} - 1 > \frac{1}{2\pi(A)},$$

and, therefore, (3.19) follows by the lower bound of (3.15). □

Note that both bounds in (3.17) depend on τ_G: the smaller τ_G is, the tighter these bounds become. Again, using the fact that the stationary distribution is uniform on regular graphs, we get the following corollary.

Corollary 3.2. *Let $G(V, E)$ be a connected, unweighted regular multi-graph and $A \subset V$. Then, for $A^c = V \setminus A$ and u_{A^c} the uniform distribution over A^c,*

$$\left(1 - 2\tau_G \frac{|A|}{n}\right) e^{-\frac{2|A|t}{n}} \leq \mathbf{P}_{u_{A^c}}(\tilde{T}_A > t) \leq e^{-\frac{|A|t}{n\tau_G}} \tag{3.20}$$

where τ_G is the relaxation time of G and $n = |V|$.

3.3 Expander Graphs

In the remainder of this chapter, we will introduce the notion of expander graphs. We give a formal definition below; roughly, an expander graph is a graph whose relaxation time is bounded, *i.e.*, it does not grow as the size of the graph increases. In the previous section, we presented several results that relate the relaxation time of a graph to two different properties on the random walk, namely, the mixing times and hitting times. These results imply that random walks over expander graphs have many interesting properties that will be particularly useful to our analysis.

Expander graphs are a very well studied concept that has applications in a variety of fields in computer science (see [HLW06] for an overview). An important fact is that they are abundant; as discussed in Section 3.3.2 below, almost all regular graphs are expanders. Since the overlay network in an unstructured peer-to-peer system is an almost regular graph, expanders are of great interest in the context of peer-to-peer systems.

So far, we have discussed random walks on weighted graphs, as this was necessary in order to derive several of the bounds in Section 3.2. In the remainder of this thesis however, our focus will be on unweighted graphs and multi-graphs. For this reason, and for the sake of concreteness, we will restrict our discussion on unweighted expanders. For a discussion of, *e.g.*, isoperimetric inequalities in the weighted setting, we refer the reader to Chung [Chu97].

Again, most of the results that we present here are classic. Our main references for expander graphs and isoperimetric inequalities are the books by Chung [Chu97] and by Hoory, Linial and Widgerson [HLW06]. Our discussion on the abundance of regular expanders is based on the aforementioned book by Hoory *et al.* [HLW06], on the survey by Wormald [Wor99] and on Friedman's proof on Alon's second eigenvalue conjecture [Fri03].

3.3.1 Definition of Expander Graphs

Consider a sequence $\{G_n\}_{n\geq n_0}$ of (unweighted) graphs or multi-graphs, where the size of graph $G_n(V_n, E_n)$ is $|V_n| = n$. Let d_{G_n} be the maximum degree of graph G_n, *i.e.*,

$$d_{G_n} = \max_{i\in V_n} d_i.$$

The sequence $\{G_n\}_{n\geq n_0}$ is called a *bounded degree sequence* if there exists a constant $d > 0$ such that

$$d_{G_n} \leq d \qquad \text{for all } n \geq n_0.$$

Let

$$\{\tau_n\}_{n\geq n_0} \equiv \{\tau_{G_n}\}_{n\geq n_0}$$

be the sequence of relaxation times corresponding to the graph sequence $\{G_n\}_{n\geq n_0}$. We will say that the sequence $\{G_n\}_{n\geq n_0}$ is an *expander family* if the following hold

(a) For all $n \geq n_0$, G_n is a connected graph.

(b) $\{G_n\}_{n\geq n_0}$ is a bounded degree sequence.

(c) The relaxation time sequence $\{\tau_n\}_{n\geq n_0}$ is asymptotically bounded (in n), *i.e.*,

$$\tau_n = O(1). \tag{3.21}$$

Hence, an expander family is a graph sequence that has "small" relaxation times, that do not become unbounded with n, the size of the graphs in the sequence. Given that, by (3.13), the relaxation time of a graph is the inverse of its spectral gap, expander families can also be characterized by the fact that the sequence of second largest eigenvalues λ_2 is asymptotically bounded away from one.

The "expander" property is, by definition, a property of a sequence of graphs and not a property exhibited by a single graph. However, it is common in practise to say that a graph G_n is an expander, effectively referring to some implicit, unambiguous sequence of graphs $\{G_n\}_{n\geq n_0}$, parameterized by their size n. We will also adopt this practise, wherever the underlying sequence is implied or can be derived from context.

Theorem 3.4 has the following immediate corollary:

Corollary 3.3. *Assume that $\{G_n\}_{n\geq n_0}$ is an expander family. Then, given $\epsilon > 0$, the mixing time τ_ϵ of a random walk on G_n is*

$$\tau_\epsilon = O(\log n).$$

Note that, since $\{G_n\}_{n\geq n_0}$ are connected and have a bounded degree, $\pi_* = \min_{i\in V}\pi_i = \Theta\left(\frac{1}{n}\right)$. In other words, random walks over expanders mix fast: within $\Theta(\log n)$ steps, the distribution of the random walk gets arbitrarily close to the stationary distribution of the walk. A similar statement can be made about hitting times, based on Theorem 3.5.

Corollary 3.4. *Assume that $\{G_n\}_{n\geq n_0}$ is an expander family, and consider a sequence of vertex sets $\{A_n\}_{n\geq n_0}$, where $A_n \subseteq V_n$. Then,*

$$\mathbb{E}_{u_{A^c}}[T_{A_n}] = \Theta\left(\mathbb{E}_{\pi_{A_n^c}}[T_{A_n}]\right)$$

and

$$\mathbb{E}_{\pi_{A_n^c}}[T_{A_n}] = \Theta\left(\frac{n}{|A_n|}\right).$$

From Lemma 3.2, the above corollary holds for both for the discrete-time and the continuous-time random walk. It is interesting to interpret Corollary 3.4 in the context of searching in peer-to-peer systems. Suppose that G_n is the overlay graph of a peer-to-peer system, and that A_n is the set of peers storing a certain file locally. Assume that a search is initiated at a peer chosen

uniformly outside the set of peers storing the file, and that the query is propagated according to a random walk. Then, loosely speaking, if the overlay graph of the system is an expander, the time it takes to locate the file does not depend on where the peers storing it are positioned in the network. Instead, it will simply be proportional to the fraction $n/|A_n|$.

3.3.2 Random Regular Graphs are Expanders

The previous section suggests that the behaviour of a random walk over expander graphs is well understood. In the following, we will present several results that indeed attest to the existence of such graphs. In doing so, we will deviate from our treatment so far of deterministic graph sequences and will instead talk about *random graphs*. The most fundamental result we present, and which will be used throughout this thesis, is that almost all d-regular graphs are expanders (see Theorem 3.7 below).

Resorting to the theory of random graphs is not strictly necessary to address the issue of existence of expander graphs: there are known deterministic sequences of graphs that are expanders (see, *e.g.*, [HLW06]). However, the "random graph" point of view is far more useful in the context of our work in unstructured peer-to-peer systems. This will become apparent in Section 4.2, where we introduce our model of an overlay graph.

Random Graphs

Let $\mathbb{G}_n$ be the set of all undirected graphs of size n, assuming (without loss of generality) that the vertex set of a graph of size n is $V_n = \{1, \ldots, n\}$. A random graph of size n is a random variable taking values in $\mathbb{G}_n$. We list several simple examples below:

$\mathcal{G}_{n,M}$: Erdös-Renyi Graphs (1st variant). The random graph sampled uniformly from all graphs of n vertices and M edges.

$\mathcal{G}_{n,p}$: Erdös-Renyi Graphs (2nd variant). A random graph generated as follows: for $i, j \in V_n$, the edge (i, j) is added to the edge set of the graph with probability p, independently of all other edges in the graph.

$\mathcal{G}_{n,d}$: Random d-Regular Graphs. Recall that a d-regular graph is a graph in which every vertex has degree d. Let $\mathbb{G}_{n,d}$ be the set of all d-regular graphs of size n. Then $\mathcal{G}_{n,d}$ is a random graph sampled uniformly from the set $\mathbb{G}_{n,d}$.

$\mathcal{CG}_{n,d}$: Random Connected d-Regular Graphs. Let $\mathbb{CG}_{n,d}$ be the set of all connected d-regular graphs of size n. Then $\mathcal{CG}_{n,d}$ is a random graph sampled uniformly from the set $\mathbb{CG}_{n,d}$.

Note that $\mathbb{G}_{n,d}$ is non-empty if and only if nd is even. This is true because the number of edges in the graph must be equal to $\frac{nd}{2}$.

Typically, one is interested in sequences of random graphs $\{\mathcal{G}_n\}_{n\geq n_0}$, for certain $n \geq n_0$. Such sequences may not be defined for all $n \geq n_0$: For example, for sequences of d-regular graphs, only n such that nd is an even number should be considered.

Given a random graph $\mathcal{G}_n$, denote by $\mathbf{P}^{\mathcal{G}_n}$ the probability measure induced on $\mathbb{G}_n$:

$$\mathbf{P}^{\mathcal{G}_n}(A) = \mathbf{P}(\mathcal{G}_n \in A), \qquad A \subseteq \mathbb{G}_n.$$

Then, given a sequence of events $A_n \subset \mathbb{G}_n$, we say that A_n occurs *asymptotically almost surely* (*a.a.s.*) if

$$\lim_{n\to\infty} \mathbf{P}^{\mathcal{G}_n}(A_n) = \lim_{n\to\infty} \mathbf{P}(\mathcal{G}_n \in A_n) = 1.$$

If $\mathcal{G}_n$ is not defined for all n but only for an infinite subsequence of $\mathbb{N}$ (*e.g.*, for n even), we assume that n only takes the corresponding values in the above limit.

We will also say that A_n occurs *with high probability* (*w.h.p.*) if

$$\mathbf{P}^{\mathcal{G}_n}(A_n) = 1 - o\left(\frac{1}{n}\right),$$

i.e., A_n occurs *a.a.s.* and the probability it occurs converges to one faster than $1 - \frac{1}{n}$.

A celebrated result, first conjectured by Noga Alon [Alo86] and proved by Joel Friedman [Fri03], is that expanders are abundant among d-regular graphs: a random d-regular graph is an expander *a.a.s.*.

Theorem 3.7 (Alon's Second Eigenvalue Conjecture [Fri03])**.** *A random d-regular graph $\mathcal{G}_{n,d}$ is an expander* a.a.s.*. In particular, let $\lambda_2(G_n)$ be the second largest eigenvalue of $\mathcal{G}_{n,d}$. Then, for any fixed $\epsilon > 0$ and any integer $d \geq 3$*

$$\lim_{n\to\infty} \mathbf{P}^{\mathcal{G}_{n,d}}\left(\lambda_2(G_n) \leq 2\frac{\sqrt{d-1}}{d} + \epsilon\right) = 1,$$

where n is restricted to values such that nd is even.

Therefore, for any $\epsilon > 0$, asymptotically, almost all d-regular graphs have a relaxation time that is bounded by $(1 - 2\frac{\sqrt{d-1}}{d} - \epsilon)^{-1}$ and are, therefore, expanders *a.a.s.*

A somewhat striking observation about the above result is that a random d-regular graph is not simply an expander. In fact, it is a "very good" expander: the following theorem states that the second largest eigenvalue of a regular graph cannot be smaller than $2\frac{\sqrt{d-1}}{d}$:

Theorem 3.8 (Alon-Boppana [Alo86])**.** *For any d-regular graph G_n,*

$$\lambda_2(G_n) \geq 2\frac{\sqrt{d-1}}{d} - O\left(\log_d n\right)^{-1}.$$

Therefore, Theorem 3.7 states that the second eigenvalue of most regular graphs is not only bounded away from one but, in fact, it is also very close to the lower bound given by the Alon-Boppana theorem.

Random Multi-Graphs

It will be useful to extend the notion of random graphs to random multi-graphs. As the set of multi-graphs of size n is not finite, it is usually simpler (see, *e.g.*, [Wor99]) to restrict the sample space to multi-graphs of a finite degree. We will confine our discussion to a more limited class, namely regular multi-graphs. We denote by $\mathbb{MG}_{n,d}$ the set of d-regular multi-graphs of size n, again defined only for nd even. Then, a random d-regular multi-graph is a random variable taking values in $\mathbb{MG}_{n,d}$.

Below we give several examples of random multi-graphs that will be of interest in the context of our work. We first remind the reader of the definitions of Hamiltonian cycles and perfect matchings. Given a graph G with size n, a *Hamiltonian cycle* [Wes01] is a spanning cycle of G, *i.e.*, a subgraph whose vertex set is the entire vertex set of G and is isomorphic to the n-cycle graph. A *matching* [Wes01] is a subset of E containing edges that share no endpoints. A matching M is called *perfect* if every vertex in V is an endpoint of one edge in M [Wes01]. A k-factor is a k-regular spanning subgraph of G —hence, a perfect matching is essentially a 1-factor of G.

$\mathcal{MH}_{n,d}$**:** $d/2$ **Random Hamiltonian Cycles.** Assume that $d > 2$ and even. Graphs $\mathcal{H}_{n,d}$ are constructed as follows: pick $d/2$ Hamiltonian cycles uniformly at random from the set of all Hamiltonian cycles in $\mathbb{G}_n$. Then, construct a sample of $\mathcal{MH}_{n,d}$ by superimposing these $d/2$ Hamilton cycles. Note that graph $\mathcal{MH}_{n,d}$ is a random, d-regular multi-graph, as an edge might appear in more than one cycle.

There is an alternative, but equivalent, definition of $\mathcal{MH}_{n,d}$. Given a d-regular multi-graph, where d is even, a complete Hamiltonian decomposition of G is a partition of its edge set into $\frac{d}{2}$ Hamiltonian cycles. Let $\mathbb{MH}_{n,d} \subset \mathbb{MG}_{n,d}$ be the subset of d-regular multi-graphs that have a complete Hamiltonian decomposition. Then, $\mathcal{MH}_{n,d}$ is a random variable sampled uniformly from $\mathbb{MH}_{n,d}$.

$\mathcal{MI}_{n,d}$**:** d **Random Perfect Matchings.** Assume that n is even. Graphs $\mathcal{MI}_{n,d}$ are constructed as follows: pick d perfect matchings uniformly at random from the set of all perfect matchings in $\mathbb{G}_n$. Then, construct a sample of $\mathcal{MI}_{n,d}$ by superimposing these d perfect matchings. Again, the resulting graph is a random, d-regular multi-graph, as the matchings may not necessarily be disjoint.

An alternative, but equivalent, definition of $\mathcal{MI}_{n,d}$ is as follows. A 1-factorization of a d-regular graph is a partition of its edge set to d 1-factors (perfect matchings). Let $\mathbb{MI}_{n,d} \subset \mathbb{MG}_{n,d}$ be the subset of d-regular multi-graphs that have a 1-factorization. Then, $\mathcal{MI}_{n,d}$ is a random variable sampled uniformly from $\mathbb{MI}_{n,d}$.

There are several reasons why the random multi-graphs $\mathcal{MH}_{n,d}$ and $\mathcal{MI}_{n,d}$ are of interest in the context of our work. As we will see in Section 4.3, these random multi-graphs will arise in the context of overlay graphs for peer-to-peer systems. An additional reason to focus on these particular multi-graphs is that they too are expanders *a.a.s.*. In fact, for both random graphs, the probability that the relaxation time is bounded can be described more precisely than in Theorem 3.7.

Theorem 3.9 (Friedman [Fri03])**.** *Fix a real* $\epsilon > 0$ *and a positive even integer* $d > 3$*. Let* $\lambda_2(\mathcal{MH}_{n,d})$ *be the second largest eigenvalue of the random multi-graph* $\mathcal{MH}_{n,d}$*. Then, there is a constant,* c*, such that*

$$\mathbf{P}\left(\lambda_2(\mathcal{MH}_{n,d}) \leq 2\frac{\sqrt{d-1}}{d} + \epsilon\right) \geq 1 - \frac{c}{n^{\lceil\sqrt{d-1}\rceil-1}} \tag{3.22}$$

Moreover, (3.22) *holds for* $\lambda_2(\mathcal{MI}_{n,d})$*, the second largest eigenvalue of the random multi-graph* $\mathcal{MI}_{n,d}$*, for all* $d > 3$ *and for* n *restricted to even values.*

Therefore, asymptotically, almost all multi-graphs that have a Hamiltonian decomposition have a relaxation time that is bounded by $(1 - 2\frac{\sqrt{d-1}}{d} - \epsilon)^{-1}$. Moreover, the same bound holds, asymptotically, for almost all multi-graphs that have a 1-factorization.

The following corollary follows immediately from Theorem 3.9 and the definition of "with high probability":

Corollary 3.5. *For* $d > 5$*,* $\mathcal{MH}_{nd}$ *and* $\mathcal{MI}_{nd}$ *are expanders* w.h.p.

Restrictions to $\mathbb{G}_{n,d}$

Given a random d-regular multi-graph $\mathcal{G}_n \in \mathbb{MG}_{n,d}$ and a subset $\mathbb{S}_{n,d} \subseteq \mathbb{MG}_{n,d}$, we call the *restriction* of $\mathcal{G}_n$ to $\mathbb{S}_{n,d}$ as $\mathcal{G}_n$ conditioned on the event $\mathcal{G}_n \in \mathbb{S}_{n,d}$. *I.e.*, if $\mathcal{G}'_n$ is the restriction

of $\mathcal{G}_n$ to $\mathbb{S}_{n,d}$, then $\mathcal{G}'_n$ is a random variable taking values in $\mathbb{S}_{n,d}$ whose distribution is given by:

$$\mathbf{P}(\mathcal{G}'_n \in A) = \frac{\mathbf{P}(\mathcal{G}_n \in A)}{\mathbf{P}(\mathcal{G}_n \in \mathbb{S}_{n,d})}, \quad A \subset \mathbb{G}_{n,d}.$$

In general, for the restriction to be meaningful, we require that the probability $\mathbf{P}(\mathcal{G}_n \in \mathbb{S}_{n,d})$ is bounded away from zero (for all n).

Note that $\mathbb{G}_{n,d}$ is a subset of $\mathbb{MG}_{n,d}$, as a simple graph is also a multi-graph. We will denote with $\mathcal{H}_{n,d}$ the restriction of $\mathcal{MH}_{n,d}$ to $\mathbb{G}_{n,d}$. Then, $\mathcal{H}_{n,d}$ is a random graph sampled uniformly from $\mathbb{H}_{n,d} \subset \mathbb{G}_{n,d}$, the set of d-regular graphs that have a complete Hamiltonian decomposition. Similarly, we denote by $\mathcal{I}_{n,d}$ the restriction of $\mathcal{MI}_{n,d}$ to $\mathbb{G}_{n,d}$; again, $\mathcal{I}_{n,d}$ is a random graph sampled uniformly from $H_{n,d} \subset \mathbb{G}_{n,d}$. We note that both of these restrictions are meaningful:

Lemma 3.3 (Wormald [Wor99])**.** *Fix $d \geq 3$. Then, there exists a constant $\epsilon > 0$ such that*

$$\mathbf{P}(\mathcal{MH}_{n,d} \in \mathbb{G}_{n,d}) > \epsilon \quad \textit{and} \quad \mathbf{P}(\mathcal{MI}_{n,d} \in \mathbb{G}_{n,d}) > \epsilon$$

for all $n > d$.

The following is an immediate corollary of the above lemma and Theorem 3.9:

Corollary 3.6. *Fix a real $\epsilon > 0$ and a positive even integer $d > 3$. Let $\lambda_2(\mathcal{H}_{n,d})$ be the second largest eigenvalue of the random graph $\mathcal{H}_{n,d}$. Then, there is a constant, c, such that*

$$\mathbf{P}\left(\lambda_2(\mathcal{H}_{n,d}) \leq 2\frac{\sqrt{d-1}}{d} + \epsilon\right) \geq 1 - \frac{c}{n^{\lceil\sqrt{d-1}\rceil - 1}} \tag{3.23}$$

Moreover, (3.23) *holds for $\lambda_2(\mathcal{I}_{n,d})$, the second largest eigenvalue of the random graph $\mathcal{I}_{n,d}$, for all $d \geq 3$ and for n restricted to even values.*

Interestingly, by restricting the multi-graphs $\mathcal{MH}_{n,d}$ and $\mathcal{MI}_{n,d}$ to $\mathbb{G}_{n,d}$ one can generate, asymptotically, almost all d-regular graphs. More precisely, if $\mathcal{G}_n$ and $\mathcal{G}'_n$ are two random graphs on $\mathbb{G}_n$, their induced probability measures $\mathbf{P}^{\mathcal{G}_n}$ and $\mathbf{P}^{\mathcal{G}'_n}$ will be called *contiguous* if, for every $A_n \in \mathbb{G}_n$,

$$\lim_{n\to\infty} \mathbf{P}^{\mathcal{G}_n}(A_n) = 1 \qquad \text{iff} \qquad \lim_{n\to\infty} \mathbf{P}^{\mathcal{G}'_n}(A_n) = 1.$$

That is, the event A_n occurs *a.a.s.* in the probability space defined by $\mathcal{G}_n$ if and only if it occurs *a.a.s.* also in the space defined by $\mathcal{G}'_n$. Then, the following theorem holds:

Theorem 3.10 (Wormald [Wor99])**.** *For even $d > 3$, $\mathbf{P}^{\mathcal{G}_{n,d}}$ is contiguous to $\mathbf{P}^{\mathcal{H}_{n,d}}$. For even $n > 0$ and for every $d \geq 3$, $\mathbf{P}^{\mathcal{G}_{n,d}}$ is contiguous to $\mathbf{P}^{\mathcal{I}_{n,d}}$.*

In other words, the restrictions $\mathcal{H}_{n,d}$ and $\mathcal{I}_{n,d}$ characterize "typical" d-regular graphs. In fact, Theorem 3.10 implies the following:

Corollary 3.7. *For even $d > 3$, $\mathcal{G}_{n,d}$ has a complete Hamiltonian decomposition* a.a.s. *For even $n > 0$, $\mathcal{G}_{n,d}$ has a 1-factorization* a.a.s.

Keeping in mind that, nd must be even for a d-regular graph of size n to exist, we see that the above two cases capture all possible cases of d-regular graphs.

The fact that $\mathcal{G}_{n,d}$ has a complete Hamiltonian decomposition implies that, for d even, $\mathcal{G}_{n,d}$ is connected *a.a.s.* (or, $\mathbf{P}^{\mathcal{G}_{n,d}}$ is contiguous to $\mathbf{P}^{\mathcal{CG}_{n,d}}$). In fact, a much stronger statement is true.

Theorem 3.11 ([Wor99, Bol01, Łuc92])**.** *For any $d \geq 3$, $\mathcal{G}_{n,d}$ is d-connected* a.a.s.

In other words, at least d vertices need to be removed from a graph $\mathcal{G}_{n,d}$ in order to disconnect it, *a.a.s.*

Label Independent Random Graphs

Recall that a *permutation* $\sigma : V \to V$ is a 1-1 and onto mapping (*i.e.*, a bijection) from set V to itself. Two graphs $G(V, E)$ and $G'(V, E')$ are called *isomorphic* if $|E| = |E'|$ and there exists a permutation $\sigma : V \to V$ such that

$$(i, j) \in E \text{ if and only if } (\sigma(i), \sigma(j)) \in E', \text{ for all } i, j \in V.$$

The permutation $\sigma : V \to V$ is called an *isomorphism* from G to G' [Wes01].

A permutation $\sigma : V_n \to V_n$ defines a bijection $\hat{\sigma} : \mathbb{MG}_{n,d} \to \mathbb{MG}_{n,d}$ that maps $G \mapsto \hat{\sigma}(G)$ in a way that σ is an isomorphism from G to $\hat{\sigma}(G)$. That is, if E is the edge set of G and E' is the edge set of $\hat{\sigma}(G)$,

$$(i, j) \in E \text{ if and only if } (\sigma(i), \sigma(j)) \in E', \text{ for all } i, j \in V.$$

A random graph $\mathcal{G} \in \mathbb{MG}_{n,d}$ is *label-independent* if all isomorphic graphs are equally likely, *i.e.*,

$$\mathbf{P}(\mathcal{G} = G) = \mathbf{P}\left(\mathcal{G} = \hat{\sigma}(G)\right), \tag{3.24}$$

for all $G \in \mathbb{MG}_{n,d}$ and for all permutations $\sigma : V_n \to V_n$. Alternatively, $\mathcal{G}$ is label-independent if the probability measure $\mathbf{P}^{\mathcal{G}}$ is invariant under all bijections $\hat{\sigma}$. In general, for $\mathbb{S}_{n,d} \subseteq \mathbb{MG}_{n,d}$, a random graph $\mathcal{G}$ taking values only in $\mathbb{S}_{n,d}$ is label-independent if

$$\hat{\sigma}(\mathbb{S}_{n,d}) = \mathbb{S}_{n,d}, \quad \text{for all permutations } \sigma : V_n \to V_n$$

i.e., $\mathbb{S}_{n,d}$ is closed under all mappings $\hat{\sigma}$, and (3.24) holds.

All random graphs we have discussed so far are label-independent. Intuitively, a random graph that is label-independent exhibits a certain symmetry, in the sense that exchanging the labels of two vertices has no effect on the definition of the random graph.

3.3.3 Isoperimetric Inequalities

As noted by Hoory *et al.* [HLW06], an attractive feature of expanders is that there are several very different ways of defining them. We have defined expander graphs as graphs with a bounded relaxation time; however, expander graphs can also be seen as graphs with "sparse cuts", as we describe below.

Let $G(V, E)$ be an unweighted graph with size $n = |V|$. Given a set $A \subseteq V$, let $A^c = V \setminus A$ be its complement. We define the *neighbourhood* $\Gamma(A)$ of A as the set of vertices adjacent to a vertex in A, *i.e.*,

$$\Gamma(A) = \{j \in V : \exists i \in A \text{ s.t. } (i, j) \in E\}.$$

We define the *vertex boundary* $\delta(A)$ of A as the set of vertices in A^c that are adjacent to A:

$$\delta(A) = \{j \in A^c : \exists i \in A \text{ s.t. } (i, j) \in V\} = \Gamma(A) \setminus A.$$

We define the *edge boundary* ∂A of A as the set of edges connecting A and its complement:

$$\partial A = \{(i, j) \in E : i \in A \text{ and } j \in A^c\}.$$

Finally, we define the *volume* of set A as the sum of the degrees of its vertices:

$$\text{vol}(A) = \sum_{i \in A} d_i.$$

Edge Expansion

The *edge expansion ratio* [Chu97] of the graph G, which is also known as the *Cheeger constant* of G, is defined as follows:

$$h_G = \min_{A \subset V} \frac{|\partial A|}{\min\left(\text{vol}(A), \text{vol}(A^c)\right)}. \tag{3.25}$$

Intuitively, the ratio in (3.25) is minimized by the set with the smallest fraction of outgoing edges, in proportion to the set's volume. The *isoperimetric number* of G [Chu97] is defined as follows:

$$h'_G = \min_{A \subset V} \frac{|\partial A|}{\min\left(|A|, |A^c|\right)}.$$

If $d_{\max}, d_{\min}$ the maximum and minimum degrees in G, respectively, these two quantities can be easily related to each other:

$$d_{\min} h_G \leq h'_G \leq d_{\max} h_G \tag{3.26}$$

In particular, if G is d-regular,

$$h_G = \frac{1}{d} h'_G = \min_{A \subset V} \frac{|\partial A|}{d \min\left(|A|, |A^c|\right)}.$$

The edge expansion can be related to the relaxation time of the graph G through the following inequalities.

Theorem 3.12 (Isoperimetric Inequalities for Edge Expansion [Chu97])**.** *Let G be a connected graph. Then*

$$\frac{h_G^2}{2} < \tau_G^{-1} \leq 2h_G, \tag{3.27}$$

where h_G is the edge expansion of G and τ_G is the relaxation time of G.

The above theorem implies that an equivalent definition of expanders can be given in terms of the edge expansion ratio of a graph, as opposed to its relaxation time.

Corollary 3.8. *Let $\{G_n\}_{n \geq n_0}$ be a bounded degree sequence of connected graphs and denote by $\{h_n\}_{n \geq n_0}$ and $\{h'_n\}_{n \geq n_0}$, the sequences of edge expansions and isoperimetric numbers, respectively, corresponding to $\{G_n\}_{n \geq n_0}$. Then, the following are equivalent:*

- *$\{G_n\}_{n \geq n_0}$ is an expander family.*
- *$h_n = \Omega(1)$.*
- *$h'_n = \Omega(1)$.*

Informally, Corollary 3.8 suggests that in expander graphs every set of vertices A with size $|A| \leq n/2$ has a large edge boundary, at least proportional to $|A|$.

Vertex Expansion

The *vertex expansion ratio* [Chu97] of G is defined as follows:

$$g_G = \min_{A \subset V} \frac{\text{vol}(\delta(A))}{\min\left(\text{vol}(A), \text{vol}(A^c)\right)}. \tag{3.28}$$

Note that, if G is a regular graph, the vertex expansion ratio becomes

$$g_G = \min_{A \subset V} \frac{|\delta(A)|}{\min\left(|A|, |A^c|\right)}.$$

Given that

$$|\delta(A)| \leq |\partial A| \leq \text{vol}(\delta(A)),$$

it is easy to see that

$$h_G \leq g_G \leq d_G h_G \tag{3.29}$$

where d_G is the maximum degree of G. The above inequalities, along with (3.27) can be used to relate the vertex expansion ratio to the relaxation time. The upper bound of τ_G^{-1} presented in the following theorem is derived thus; the lower bound is tighter that the one resulting from (3.27) and (3.29).

Theorem 3.13 (Isoperimetric Inequalities for Vertex Expansion [Chu97])**.** *Let $G(V, E)$ be a connected graph. Then*

$$\frac{g_G^2}{4d_G - 2d_G g_G} < \tau_G^{-1} \leq 2g_G, \tag{3.30}$$

where g_G is the vertex expansion of G, $d_G = \max_{i \in V} d_i$, and τ_G is the relaxation time of G.

Again, (3.30) suggests an equivalent definition of expander graphs, given in terms of the vertex expansion ratio:

Corollary 3.9. *Let $\{G_n\}_{n \geq n_0}$ be a bounded degree sequence of connected graphs and denote by $\{g_n\}_{n \geq n_0} \equiv \{g_{G_n}\}_{n \geq n_0}$, the sequence of vertex expansions corresponding to $\{G_n\}_{n \neq n_0}$. Then, the following are equivalent:*

- *$\{G_n\}_{n \geq n_0}$ is an expander family.*
- *$g_n = \Omega(1)$.*

Informally, Corollary 3.9 suggests that, in expander graphs, every set of vertices A with size $|A| \leq n/2$ has a large vertex boundary, at least proportional to $|A|$.

Vertex Expansion of Small Sets

The vertex expansion ratio of a graph can be no more than one; this is because it is an upper bound on the ratio $|\delta(A)|/|A|$ for $|A| = \frac{n}{2}$. However, this is not the typical value of this ratio for smaller sets. In particular, for any d-regular graph G, define

$$g_G(k) = \min_{A \subset V, |A| \leq k} \frac{|\delta(A)|}{|A|}, \quad k \leq \frac{n}{2}$$

as the expansion ratio restricted to sets with size less than k, and a modified version

$$g'_G(k) = \min_{A \subset V, |A| < k} \frac{|\Gamma(A)|}{|A|}, \quad k \leq \frac{n}{2}.$$

Note that $g_G(\frac{n}{2}) \leq 1$ while $g'_G(\frac{n}{2}) \leq 2$. The following theorem by Kahale gives a method for bounding $g'_G(k)$ for small k, when the graph is an expander.

Theorem 3.14 (Kahale [Kah95, HLW06])**.** *There exists an absolute constant $c > 0$ such that any d-regular graph G satisfies the following inequality for all $\epsilon > 0$*

$$g'_G(\epsilon n) \geq \frac{d}{2} \cdot \left(1 - \sqrt{1 - 4\frac{d-1}{d^2 \lambda_2(G)^2}}\right) \cdot \left(1 - \frac{c \log d}{\log\left(\frac{1}{\epsilon}\right)}\right).$$

where $\lambda_2(G)$ is the second largest eigenvalue of G.

To see how Theorem 3.14 can be used, consider a "good" expander, that has

$$\lambda_2(G) = \frac{2\sqrt{d-1}}{d} + o(1).$$

Then, by taking ϵ to be arbitrarily small, one can get $g'_G(\epsilon n)$ to be arbitrarily close to $\frac{d}{2}$, which is much larger than 2, the strict upper bound for large sets. In particular, Theorem 3.9 along with Kahale's theorem immediately imply the following:

Corollary 3.10. *Fix a positive even integer $d > 3$. Then, for every $\delta > 0$ there exists an $\epsilon > 0$ such that*

$$\mathbf{P}\left(g'_{\mathcal{H}_{n,d}}(\epsilon n) \geq \frac{d}{2}(1-\delta)\right) \geq 1 - O\left(\frac{1}{n^{\lceil\sqrt{d-1}\rceil - 1}}\right). \tag{3.31}$$

Moreover, (3.31) *holds also for $g'_{\mathcal{I}_{n,d}}(\epsilon n)$, for all $d \geq 3$ and for n restricted to even values.*

For random d-regular graphs, the lower bound of $g_G(\epsilon n)$ can be further improved from 1 to arbitrarily close to $d-2$, *a.a.s.*, which happens to be tight. There is a caveat however: the closer the bound is to $d-2$, the slower the *a.a.s.* convergence, as stated by the following theorem.

Theorem 3.15 (Hoory, Linial, Widgerson [HLW06])**.** *Let $d \geq 3$ be a fixed integer. Then, for every $\delta > 0$ there exists an $\epsilon > 0$ such that*

$$\mathbf{P}\left(g_{\mathcal{G}_{n,d}}(\epsilon n) \geq d - 2 - \delta\right) \geq 1 - O\left(\frac{1}{n^{\frac{\delta}{2}}}\right).$$

The above theorem can be extended to any probability measure that is contiguous to $\mathbf{P}^{\mathcal{G}_{n,d}}$ and whose support is a subset of $\mathbb{G}_{n,d}$.

Corollary 3.11. *Let $d \geq 3$ be a fixed integer, and let $\mathbb{S}_{n,d} \subseteq \mathbb{G}_{n,d}$ be such that the uniform distribution over $\mathbb{S}_{n,d}$ is contiguous to the uniform distribution over $\mathbb{G}_{n,d}$. Then Theorem 3.15 also holds for a graph chosen uniformly from $\mathbb{S}_{n,d}$.*

Proof. Let $\mathcal{G}$ be the random graph sampled uniformly from $\mathbb{S}_{n,d}$. Given, $\delta > 0$, take $\epsilon > 0$ as in Theorem 3.15 and define

$$A(\delta, n) = \{G \in \mathbb{G}_{n,d} \text{ s.t. } g_G(\epsilon n) < d - 2 - \delta\}.$$

The random variable $\mathcal{G}$ is uniformly distributed over $\mathbb{S}_{n,d} \subset \mathbb{G}_{n,d}$. Hence

$$\mathbf{P}\left(g_{\mathcal{G}}(\epsilon n) < d - 2 - \delta\right) = \frac{\mathbf{P}^{\mathcal{G}_{n,d}}(A(\delta, n) \cap \mathbb{S}_{n,d})}{\mathbf{P}^{\mathcal{G}_{n,d}}(\mathbb{S}_{n,d})} \leq \frac{\mathbf{P}^{\mathcal{G}_{n,d}}(A(\delta, n))}{\mathbf{P}^{\mathcal{G}_{n,d}}(\mathbb{S}_{n,d})}.$$

By the contiguity assumption, the denominator of the r.h.s. converges to one while, by Theorem 3.15, the numerator converges to zero as $O\left(\frac{1}{n^{\frac{\delta}{2}}}\right)$, and the corollary follows. □

In particular, Theorems 3.10 and 3.11 imply that Theorem 3.15 also holds if $\mathbb{S}_{n,d}$ is one of the sets $\mathbb{H}_{n,d}$ (the set of d-regular graphs with a complete Hamiltonian decomposition), $\mathbb{I}_{n,d}$ (the set of d-regular graphs with a 1-factorization), or $\mathbb{CG}_{n,d}$ (the set of connected d-regular graphs).

3.4 Summary

Several quantities of interest relating to the random walk over an undirected graph can be succinctly described through bounds involving the graph's relaxation time. These bounds are of most interest when the graph is an expander —*i.e.*, its relaxation time is bounded. Expander graphs are abundant, as (asymptotically) almost all regular graphs are expanders. Finally, graphs with bounded relaxation time also have large vertex and edge expansion ratios; this allows us to give an alternative definition (and interpretation) of what an expander graph is.

In the next chapter, we present the model that we use in our analysis. As we will see in Section 4.2.2, the abundance of expander graphs will reappear during our discussion of the nature of overlay graphs in unstructured peer-to-peer systems.

In Table 3.1, we provide a brief summary of the notation introduced so far that will be used again in later chapters.

Table 3.1: SUMMARY OF NOTATION APPEARING IN CHAPTER 3

$\mathbf{P}_i(\cdot)$, $\mathbf{P}_\rho(\cdot)$	The probability of an event defined over a Markov chain or a Markov process, given that the initial state is i or distributed according to ρ, respectively.
$\mathbb{E}_i(\cdot)$, $\mathbb{E}_\rho(\cdot)$	The expectation of a random variable defined over a Markov chain or a Markov process, given that the initial state is i or distributed according to ρ, respectively.
$\Gamma(A)$	The neighbourhood of a vertex set A.
$\delta(A)$	The vertex boundary of a vertex set A.
$\partial(A)$	The edge boundary of a vertex set A.
d_G	The maximum degree of graph G.
τ_G	The relaxation time of graph G.
h_G	The edge expansion ratio of graph G.
g_G	The vertex expansion ratio of graph G.
$\lambda_2(G)$	The second largest eigenvalue of graph G.
$\mathbb{G}_{n,d}$	The set of d-regular graphs of size n.
$\mathbb{H}_{n,d}$	The set of d-regular graphs of size n having a complete Hamiltonian decomposition.
$\mathbb{I}_{n,d}$	The set of d-regular graphs of size n having a 1-factorization.
$\mathbb{CG}_{n,d}$	The set of connected d-regular graphs of size n.
$\mathbb{MG}_{n,d}$	The set of d-regular multi-graphs of size n.
$\mathbb{MH}_{n,d}$	The set of d-regular multi-graphs of size n having a complete Hamiltonian decomposition.
$\mathbb{MI}_{n,d}$	The set of d-regular multi-graphs of size n having a 1-factorization.
$\mathbf{P}^{\mathcal{G}}$	The probability measure induced by random graph $\mathcal{G}$.

Chapter 4

Model

In this chapter, we propose our mathematical model of an unstructured peer-to-peer system. We will use this model to formally derive our main results —*i.e.*, to obtain a formal characterization of the scalability and reliability of the query propagation mechanisms considered in this thesis. Before presenting our model, we note that, to make it tractable, we inevitably have to make certain simplifying assumptions. For this reason, in Chapters 5 and 6, apart from formally deriving our results, we will also verify them though simulations; several of our simplifying assumptions will be removed when performing these simulations.

In spite of these assumptions, our model is still quite descriptive. More specifically, there are three key components of an unstructured peer-to-peer system that are captured by this model. The first is the process describing the peer population dynamics —*i.e.*, how peers arrive and depart. The second is the overlay graph —*i.e.*, the network of connections among peers— and how it evolves through time due to churn. The third is the evolution of content, determined by how often and when peers request or bring files into the system.

Several of our modelling assumptions are directly motivated by the measurement studies we presented in Chapter 2 (see also Section 2.2.4). In particular:

- We assume that the system size (*i.e.*, the number of peers in the system) is constant as time progresses: although the system is subject to churn, each departing peer is immediately replaced by a new peer, thus maintaining the population size fixed. As discussed in Section 2.2.1, the population size in real peer-to-peer systems changes slowly, over long-term periods of time (*e.g.*, within months or years). Over short-term periods (*e.g.*, days or weeks), the population size appears to oscillate around an operating point. Our modelling assumption that the system size is fixed aims to capture the behaviour at such an operating point.

- We assume that the overlay graph representing the network is regular at all times (*i.e.*, every peer has the same number of connections). This aims to capture the fact that, in the real overlays described in Section 2.2.2, the degree distribution is concentrated around a constant value, that did not depend on the system size.

- We consider files that are requested very often but brought in the system very rarely, as well as files requested rarely but brought in the system often. This is motivated by the fact that, as observed in Section 2.2.3, in real peer-to-peer systems there exists a discrepancy between the popularity of a file and its availability.

We note that our model captures the dynamic nature of both the overlay graph and the content stored by peers. As such, it departs considerably from models used so far to analyze the behaviour of query propagation mechanisms in unstructured systems, discussed in Section 2.3. In particular, rather than modelling the overlay as a random graph or assuming that files are distributed uniformly at random, we capture their joint evolution as a Markov process. This Markov process is determined by how peers arrive, depart, request and share files and, last but not least, connect to each other in order to form the time-variant overlay graph.

Markovian models have been proposed in the past to describe the evolution of the overlay graph of an unstructured peer-to-peer system —although, to the best of our knowledge, our work is the first to use such a model to analyze the behaviour of query propagation mechanisms. We give a detailed description of previously proposed Markovian models in Section 4.3. Our work extends these models in two ways. First, our model captures the joint evolution of the overlay *and* the files shared by peers. Second, it generalizes existing models by describing a much wider class of overlay graphs. As a result, several of the theorems appearing in Chapters 5 and 6, which are proved under our general model, immediately follow (as special cases) for these previously proposed models. Whenever such statements can be derived, this is noted explicitly in the text (see also Sections 5.2 and 6.2).

The remainder of this chapter is structured as follows. We first describe how peers arrive and depart (Section 4.1). We then describe how the overlay graph can change under system churn (Sections 4.2 and 4.3). Finally, we conclude by describing the process under which peers request and publish data (Section 4.4) in both a pure and a hybrid peer-to-peer system.

We do not describe in this chapter the query propagation mechanism employed by peers or the metrics we use to evaluate such a mechanism's performance (and, in particular, its scalability and reliability). As we will use different mechanisms and different metrics for the pure and hybrid cases, we leave the presentation of these aspects of our model for Chapters 5 and

6, respectively.

4.1 Peer Population Dynamics

We begin our model's definition by first describing the system churn, *i.e.*, the process characterizing how new peers arrive and old ones depart. As we will see, our standing modelling assumption is that the number of peers in the system remains constant as time progresses. The system is subject to churn, as the peer population changes through time; however, each departing peer is immediately replaced by a new peer, thus maintaining the population size fixed.

As mentioned above, this assumption is motivated by the measurement studies presented in Section 2.2.1. In particular, the systems that we aim to capture are systems in which the population size changes slowly over a long-term period of time (*e.g.*, within months or years); over short-term periods (*e.g.*, days or weeks), the population size oscillates around an operating point, which we model through parameter n, the (fixed) system size.

Instead of peers being immediately replaced when they leave, an alternative —but equivalent— way of viewing our system is the following: new peers arrive according to a Poisson process and "oust" old peers, chosen uniformly at random. Each of these two equivalent definitions of churn provides useful insight on how the peer population evolves in our model, so we discuss both of them in more detail below.

4.1.1 Immediate Replacement Viewpoint

Our peer-to-peer network consists of a constant number of peers, which we denote by n, that does not change as time progresses. In particular, we assume that initially (at time 0), the system consists of precisely n peers. Each peer stays in the system for an exponentially distributed time (lifetime) with mean $1/\mu$, independent of the lifetimes of all other peers. Whenever a peer departs, it is immediately replaced by a new peer that again has an exponentially distributed lifetime with mean $1/\mu$, independent of the lifetimes of all other peers that are, or have been, in the system. Because every departing peer is immediately replaced, the total number of peers remains constant and equal to n. We refer to n as the *system size* and treat it as a parameter of our model.

The fact that departing peers are immediately replaced by newly arriving peers allows us to simplify the system's representation at a given point in time. More formally, we label each of the initial peers in the system as $1, \ldots, n$ and denote by V_n the set $\{1, \ldots, n\}$. We say that peer

B is a *successor* of peer A if B replaces A upon A's departure. Similarly, we say that A is a *predecessor* of B if B is A's successor. We can associate each $i \in V_n$ with an infinite sequence of consecutive peers, where (a) the first peer of this sequence is the initial peer labelled i and (b) the $k+1$-th peer appearing in the sequence is the successor of the k-th peer. We call such a sequence, whose first peer is $i \in V_n$, the i-th *successor sequence*.

The n successor sequences correspond to n independent Poisson processes whose rates are equal to μ. These processes describe the epochs at which peers in each sequence are replaced by their successors. The content stored by a successor peer may differ from the content of its predecessor. Similarly, the connections of the successor peer (forming the overlay graph) may differ from its predecessor. We describe in detail how such changes may take place in Sections 4.4 and 4.2, respectively.

To simplify our notation, we identify each peer in the system, at some arbitrary time $t \geq 0$, with the successor sequence it belongs to. In particular, note that, for each $i \in V_n$ and for every time $t \geq 0$, there is exactly one peer present in the system that belongs to the i-th successor sequence: the unique "descendant" of the initial peer labelled i that is present in the system at time t. We refer to this peer as "the i-th peer in the system at time t" or, simply, "peer i at time t", where $i \in V_n$.

4.1.2 Poisson Arrivals Viewpoint

As we mentioned above, there is an alternative (and equivalent) way of viewing the above system. The system again consists of precisely n peers initially, and new peers arrive according to a Poisson process with rate $n \cdot \mu$. Whenever a new peer arrives, it replaces a peer chosen uniformly at random among all existing peers. That is, each of the existing peers is replaced with probability $1/n$. This process is equivalent to the one we described above. In particular, each peer stays in the system for an exponentially distributed time with mean $1/\mu$, which is independent of the lifetimes of all other peers.

More formally, peers arrive according to a Poisson process $\{N(t)\}_{t\in\mathbb{R}_+}$ with rate μ, where $N(t)$ is the number of arrivals up to and including time $t \geq 0$. At each arrival, one of the n successor sequences is chosen uniformly at random. Denote by

$$\{I(t)\}_{t\in\mathbb{N},t>0}, \quad I(t) \in V_n,$$

the choice made at the t-th departure/arrival epoch. We call $\{I(t)\}_{t\in\mathbb{N},t>0}$ the *churn sequence*. Intuitively, $I(t)$ captures which peer (labelled according to the successor sequence it belongs

to) is replaced at the t-th departure/arrival epoch. In our model, $I(t)$ are independent and uniformly distributed over V_n, *i.e.*,

$$\mathbf{P}(I(t) = i) = \frac{1}{n}, \qquad i \in V_n, t \geq 1.$$

For $i \in V_n$, let $\{N^i(t)\}_{t\in\mathbb{R}_+}$ be the counting process of the epochs at which the i-th peer is replaced (*i.e.*, $N^i(t)$ is the number of times that the i-th peer has been replaced up to and including time $t \geq 0$). It is a fundamental property of the Poisson process (see, *e.g.*, [Gal96, Chapter 2, Section 2.3, pp. 38-41]) that $\{N^i(t)\}_{t\in\mathbb{R}_+}$ are *independent* Poisson processes with rates μ. As a result, the lifetimes of peers under the above process are independent and exponentially distributed with mean $1/\mu$, and the above system is equivalent to the one defined in Section 4.1.1.

4.2 Overlay Graph

In our model, each peer maintains a number of connections to other peers, thus forming an overlay graph. Newly arriving peers establish connections to existing peers, while departing peers abolish their connections; as a result, the overlay graph is time-variant. We assume that the overlay graph can change only during departure/arrival epochs: between any two consecutive epochs, the overlay graph remains static.

In general, the changes incurred on the graph at a departure/arrival epoch are the result of the connection protocol followed by departing and arriving peers. We assume that the protocol followed by peers is such that, at any point in time, the overlay graph formed by peers is a d-regular multi-graph, where d is a constant that does not depend on n. This assumption is motivated by the connection protocols described in Section 2.1 and by the measurement studies of real overlays appearing in Section 2.2.2.

More formally, recall that $\mathbb{MG}_{n,d}$ is the set of d-regular multi-graphs, as defined in Chapter 3. Then, the overlay graph at any point in time in our model is represented as a multi-graph $\{G(t)\}_{t\in\mathbb{R}_+}$, where $G(t)$ takes values in some set $\mathbb{S}_{n,d} \subseteq \mathbb{MG}_{n,d}$. The vertex set of the graph is $V_n = \{1, \ldots, n\}$: in other words, we define the overlay graph as a graph over the set of the n peers in the system, labelled according to their respective successor sequences. In effect, every successor sequence corresponds to a single vertex of the graph $G(t)$. Although, under the above definition, the vertex set V_n is constant through time, the edge set is time-variant. The edges incident to vertex i, $i \in V_n$, at time t correspond to the connections maintained by the i-th peer at time t.

As stated above, changes in the graph may happen only at the departure/arrival epochs. Hence, the overlay graph can be equivalently represented by a sequence $\{G(t)\}_{t\in\mathbb{N}}$, corresponding to the graph immediately after the t-th departure/arrival epoch. Note that the time between two consecutive changes is exponentially distributed with mean $(n\mu)^{-1}$.

We assume that $\{G(t)\}_{t\in\mathbb{R}_+}$ is a Markov process, whose state space is $\mathbb{S}_{n,d}$ and whose transitions are determined by the connection protocol followed by peers. Below, we describe how we model these transitions for a general system, without focusing on any specific connection protocol. Specific connection protocols and the Markov processes $\{G(t)\}_{t\in\mathbb{R}_+}$ that they lead to are an interesting subject in their own right and have been studied in the past [LS03, CDG05, FGMS06, Tay81, MS05, GMS04]. For this reason, we discuss such protocols in detail separately, in Section 4.3.

4.2.1 Churn-Driven Markovian Graph Models

Under the assumption that $\{G(t)\}_{t\in\mathbb{R}_+}$ is a Markov process in $\mathbb{S}_{n,d}$, the transition rate from each state is $n\mu$. As a result, the process is uniformized, and the embedded Markov chain of the process $\{G(t)\}_{t\in\mathbb{N}}$ has the same stationary distribution (should one exist) as $\{G(t)\}_{t\in\mathbb{R}_+}$. Keeping this in mind, we focus on describing the transitions of the embedded chain $\{G(t)\}_{t\in\mathbb{N}}$.

In a real peer-to-peer system, the change on the graph after an arrival or a departure event depends on which peer departs or arrives. The arriving or departing peer usually runs an appropriate connection protocol, like the one outlined in Section 2.1.1, and the graph is altered only "locally", at the neighbourhood of this peer. For this reason, the Markov process describing how the overlay graphs evolve should depend on the churn process, describing which peers arrive or depart. In the context of our model, where departure/arrival events occur simultaneously, the churn sequence $\{I(t)\}_{t\in\mathbb{N},t>0}$ should determine the transitions of Markov chain $\{G(t)\}_{t\in\mathbb{N}}$.

More formally, recall from Section 4.1.2 that the churn sequence $\{I(t)\}_{t\in\mathbb{N},t>0}$, where $I(t) \in V_n$, is a sequence of i.i.d. random variables capturing at which vertex —or, successor sequence— the t-th simultaneous departure/arrival event occurs. In particular, in our model, $I(t)$ are independent and uniformly distributed over V_n, *i.e.*,

$$\mathbf{P}(I(t) = i) = \frac{1}{n}, \qquad i \in V_n, t \in \mathbb{N}, t > 0.$$

We assume that the Markov chain $\{G(t)\}_{t\in\mathbb{N}}$ is driven by the above process $\{I(t)\}_{t\in\mathbb{N},t>0}$ as follows. For every pair of graphs G and G' in $\mathbb{S}_{n,d}$, and for every $i \in V_n$ we define the

conditional transition probability $p^i_{GG'}$ as the probability that a transition from G to a graph G' occurs, conditioned on the event that the vertex on which the departure/arrival event takes place is the vertex i. ***I.e.***,

$$\begin{aligned} p^i_{GG'} &\equiv \mathbf{P}(G(t+1) = G' \mid I(t+1) = i, G(t) = G) \\ &= \mathbf{P}(G(t+1) = G' \mid I(t+1) = i, I(t) \in \cdot, \ldots, I(1) \in \cdot, \\ &\qquad G(t) = G, G(t-1) \in \cdot, \ldots, G(0) \in \cdot) \end{aligned}$$

In other words, the evolution of $\{G(t)\}_{t\in\mathbb{N}}$ depends on which vertex a departure/arrival event occurs next. Given that the current graph is G, and that a departure/arrival event occurs at vertex i, $p^i_{GG'}$ gives the probability that the new graph will be G'. The change in the graph is therefore determined by which peer is being replaced.

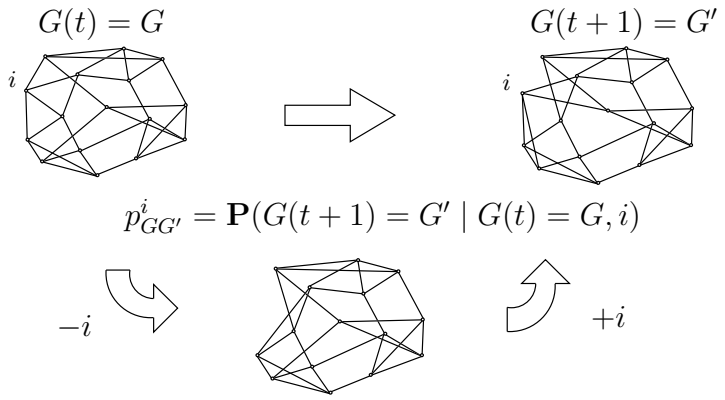


Figure 4.1: A transition of the overlay graph $G(t)$. The departure of peer i from G leads to an "intermediate" graph of size $n-1$, and the arrival of the new peer replacing i leads to a new graph G' of size n. The transition probability p^i_{GG} is the probability that the final graph is G', given that the initial graph is G and the peer being replaced is i. Note that our model characterizes the initial graph G and the final graph G' but not the intermediate graph.

The probabilities $p^i_{GG'}$ depend on the protocol used by the departing peer when exiting the system and the protocol used by its successor to connect to the remaining peers. This is illustrated in Figure 4.1: the departure of a peer from G leads to an "intermediate" graph of size $n-1$, and the arrival of the new peer leads to a new graph G' of size n. Note that, in our analysis, we care about the initial graph G and the final graph G' but not the intermediate graph. In Section 4.3, we give examples of protocols followed by peers during their arrival and their departure, as well as the probabilities $p^i_{GG'}$ that they imply.

We call the above model a *churn-driven Markovian graph model*, as the churn sequence $\{I(t)\}_{t\in\mathbb{N},t>0}$ drives the transitions of $\{G(t)\}_{t\in\mathbb{N}}$. Note that the (marginal) process $\{G(t)\}_{t\in\mathbb{N}}$ is in fact a Markov chain —and, thus, we correctly refer to this model as a Markovian graph model. In particular, $\{G(t)\}_{t\in\mathbb{N}}$ exhibits the Markov property, and its transition probability matrix is $P = [p_{GG'}]_{G,G'\in\mathbb{S}_{n,d}}$ where

$$p_{GG'} = \sum_{i\in V_n} \mathbf{P}(I = i) \cdot p^i_{GG'} = \sum_{i\in V_n} \frac{1}{n} \cdot p^i_{GG'}.$$

4.2.2 Unstructured Systems in Our Model

In this thesis, we are interested in analyzing the behaviour of unstructured peer-to-peer systems. As discussed in Chapter 2, the main premise behind unstructured systems is that the overlay graph can, at any point in time, be an arbitrary graph. Having defined a mathematical model of the overlay graph, we can make this statement more precise.

Assume that the churn-driven Markov chain $\{G(t)\}_{t\in\mathbb{N}}$ is irreducible and that its state space is $\mathbb{S}_{n,d} \subseteq \mathbb{G}_{n,d}$ (*i.e.*, it is restricted to simple regular graphs). We say that the peer-to-peer system is unstructured if *the stationary distribution of* $\{G(t)\}_{t\in\mathbb{N}}$ *is contiguous to the uniform distribution over* $\mathbb{G}_{n,d}$.

The reasoning behind this definition is quite intuitive. If the stationary distribution of the overlay graph is concentrated over (or considerably biased toward) a small subset of $\mathbb{G}_{n,d}$, this would suggest that, in fact, the peer-to-peer system exhibits a certain structure. This structure could —and should— be exploited to search more efficiently for items stored in the system.

For example, a churn-driven Markov chain could be used to describe the evolution of, *e.g.*, the overlay graph of a structured system like Chord [SMK+01]. Of course, one would need to determine the transition probabilities $p^i_{GG'}$ implied by the connection protocol used by Chord. In any case, the stationary distribution of such a chain would be concentrated around graphs exhibiting the Chord-graph structure; this, in turn, is exploited by the search mechanism of the Chord peer-to-peer system.

However, the standing assumption in the literature on unstructured systems is that the overlay topology can be arbitrary, and that query propagation mechanisms should be designed to operate without any prior assumptions on the peer topology. For this reason, according to our definition, a system is unstructured precisely when its distribution is almost uniform (in the sense of contiguity) over the set of all d-regular graphs.

Our model allows the overlay graph to be a d-regular multi-graph. If the state space $\mathbb{S}_{n,d}$ indeed includes multi-graphs (that is, $\mathbb{S}_{n,d} \subseteq \mathbb{MG}_{n,d}$ and $\mathbb{S}_{n,d} \setminus \mathbb{G}_{n,d} \neq \emptyset$) we can similarly

say that the peer-to-peer system is unstructured if the stationary distribution of $\{G(t)\}_{t\in\mathbb{N}}$ is contiguous to the uniform distribution over $\mathbb{MG}_{n,d}$. Though our analysis will indeed incorporate multi-graph models, our main focus will be on simple graphs, as those are more often encountered in practise.

The above definition of an unstructured system has several immediate consequences, that are implied by the results presented in Section 3.3.2. First, Theorem 3.11 implies that the overlay graph is k-connected *a.a.s.* This is interesting, because it suggests that the overlay graph is robust to random failures of peers. Most importantly however, Theorem 3.7 implies that the overlay graph is an expander *a.a.s.* This is a very useful property, given the multitude of results presented in Chapter 3 about such graphs.

More formally, assume again that $\{G(t)\}_{t\in\mathbb{N}}$ is irreducible, and denote by

$$\nu_G, \quad G \in \mathbb{S}_{n,d},$$

its unique stationary distribution. By uniformity, this is also the unique stationary distribution of $\{G(t)\}_{t\in\mathbb{R}_+}$. We denote by τ_n the relaxation time of a random graph sampled from $\mathbb{S}_{n,d}$ according to the distribution ν_G. That is, τ_n is a random variable in $\mathbb{R}_+ \cup \{+\infty\}$ whose cumulative distribution function is given by

$$\mathbf{P}(\tau_n \leq t) = \sum_{G\in\mathbb{S}_{n,d}} \nu_G \mathbb{1}_{\tau_G \leq t}, \quad t \in \mathbb{R}_+ \cup \{+\infty\}. \tag{4.1}$$

We call τ_n the *steady state* relaxation time of $\{G(t)\}_{t\in\mathbb{N}}$. Note that, if $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic, then

$$\mathbb{E}[\tau_n] = \sum_{G\in\mathbb{S}_{n,d}} \tau_G \nu_G = \lim_{t\to\infty} \mathbb{E}[\tau_{G(t)}] = \lim_{t\to\infty} \frac{1}{t} \sum_{s=0}^{t} \tau_{G(t)}, \quad \textit{a.s.} \tag{4.2}$$

by the key renewal theorem [Gal96].

Following our notation of Section 3.3.2, we say that the overlay graph is an expander *a.a.s.* if τ_n is bounded *a.a.s.*, *i.e.*, there exists a constant $\bar{\tau} \geq 1$ such that

$$\mathbf{P}(\tau_n \leq \bar{\tau}) = 1 - o(1).$$

Similarly, we say that the overlay graph is an expander *w.h.p.* if there exists a constant $\bar{\tau} \geq 1$ such that

$$\mathbf{P}(\tau_n \leq \bar{\tau}) = 1 - o\left(\frac{1}{n}\right).$$

Theorem 3.7 immediately implies that an unstructured system, under our model, has an expander overlay graph *a.a.s.* In this sense, the expander property follows as a natural consequence of the contiguity to the uniform distribution over d-regular graphs.

Moreover, Corollary 3.6 gives us two probability measures that are contiguous to the uniform distribution over all d-regular graphs and have bounded relaxation time *w.h.p.* Hence, any overlay graph whose stationary distribution is given by these two probability measures will be an expander *w.h.p.* We further elaborate on this issue in Section 4.3, where we present overlay graphs with the above stationary distributions.

4.2.3 Vertex Reversibility and Vertex Balance

Below, we define two technical properties of a churn-driven Markov model that will play an important role in our analysis of Chapters 5 and 6. They capture two different notions of "symmetry" in a churn-driven Markovian graph model —these notions are not identical, as one property is weaker than the other. We note that these two properties hold for all the models of peer-to-peer overlay graphs [LS03, CDG05, FGMS06, Tay81, MS05, GMS04] that we review in Section 4.3.

Vertex Reversibility

Suppose that $\{G(t)\}_{t\in\mathbb{N}}$ is a churn-driven Markovian graph model, and that it is irreducible. By the definition of reversibility, $\{G(t)\}_{t\in\mathbb{N}}$ will be reversible if its stationary distribution ν satisfies the following equations

$$\nu_G \sum_{i\in V_n} \mathbf{P}(I=i)\cdot p^i_{GG'} = \nu_{G'} \sum_{i\in V_n} \mathbf{P}(I=i)\cdot p^i_{G'G}, \qquad \text{for all } G, G' \in \mathbb{S}_{n,d}.$$

In our work, we require at times that the Markov chain exhibits a stronger property. We say that a churn-driven Markovian graph model is *vertex reversible* if its stationary distribution ν satisfies the following set of equations

$$\nu_G p^i_{GG'} = \nu_{G'} p^i_{G'G}, \qquad \text{for all } G, G' \in \mathbb{S}_{n,d} \text{ and for every } i \in V_n. \tag{4.3}$$

Obviously, vertex reversibility implies the reversibility of $\{G(t)\}_{t\in\mathbb{N}}$.

The reason why we introduce this stronger notion of reversibility will become apparent later, in Section 5.3. To gain some insight as to why we refer to this quantity as "vertex" reversibility, consider a Markov chain $\{G^i_n(t)\}_{t\in\mathbb{N}}$ over $\mathbb{S}_{n,d}$ with transition probabilities $p^i_{GG'}$.

Intuitively, $\{G_n^i(t)\}_{t\in\mathbb{N}}$ describes transitions when the peer being replaced is always the peer at vertex i, *i.e.*, $I(t) = i$ for all $t \geq 1$. This is therefore a "restricted to vertex i" version of the process $\{G(t)\}_{t\in\mathbb{N}}$.

Eq. (4.3) implies that the chain $\{G_n^i(t)\}$ is reversible, for every i; this is why we refer to $\{G(t)\}_{t\in\mathbb{N}}$ as "vertex" reversible, if it satisfies (4.3). Note that the chain $\{G_n^i(t)\}_{t\in\mathbb{N}}$ may not be irreducible, even if $\{G(t)\}_{t\in\mathbb{N}}$ is; as a result, although Eq. (4.3) implies that ν is a stationary distribution of $\{G_n^i(t)\}$, it may not necessarily be unique. If $\{G_n^i(t)\}_{t\in\mathbb{N}}$ is irreducible and $\{G(t)\}_{t\in\mathbb{N}}$ is vertex reversible, then ν is also the unique stationary distribution of $\{G_n^i(t)\}$.

Vertex Balance

Suppose that $\{G(t)\}_{t\in\mathbb{N}}$ is a churn-driven Markovian graph model, and that it is irreducible. By Theorem 3.3, there exists a unique stationary distribution ν over $\mathbb{S}_{n,d}$ that satisfies the balance equations

$$\sum_{G'\in\mathbb{S}_{n,d}} \nu'_G \sum_{i\in V_n} \mathbf{P}(I = i) \cdot p^i_{G'G} = \nu_G, \qquad \text{for all } G \in \mathbb{S}_{n,d}.$$

Again, we require at times that the Markov chain exhibits a stronger property than simple irreducibility. We say that an irreducible churn-driven Markovian graph model is *vertex balanced* if its stationary distribution ν satisfies the following system of equations

$$\sum_{G'\in\mathbb{S}_{n,d}} \nu_{G'} p^i_{G'G} = \nu_G, \qquad \text{for all } G \in \mathbb{S}_{n,d} \text{ and for every } i \in V_n. \tag{4.4}$$

Obviously, any distribution that satisfies the equations (4.4) also satisfies the balance equations of $\{G(t)\}_{t\in\mathbb{N}}$ and, thus, it is a stationary distribution of the churn-driven chain.

The notion of vertex balance is weaker than the notion of vertex reversibility. In particular,

Lemma 4.1. *Every vertex reversible chain is also vertex balanced.*

Proof. Assume that a chain is vertex reversible. Then (4.3) holds. We thus have that

$$\sum_{G'\in\mathbb{S}_{n,d}} \nu_{G'} p^i_{G'G} = \sum_{G'\in\mathbb{S}_{n,d}} \nu_G p^i_{GG'} = \nu_G \sum_{G'\in\mathbb{S}_{n,d}} p^i_{GG'} = \nu_G$$

for all $G \in \mathbb{S}_{n,d}$ and every $i \in V_n$. *I.e.*, the stationary distribution ν satisfies (4.4), and the chain is vertex balanced. □

The converse is not necessarily true; in fact, a vertex balanced chain may not even be reversible.

Again, the reason why we introduce this notion will become apparent later, also in Section 5.3. To gain some insight as to why we refer to this quantity as "vertex" balance we can again consider the Markov chain $\{G_n^i(t)\}_{t\in\mathbb{N}}$, which describes transitions when the peer being replaced is always vertex i. Recall that this is a chain over $\mathbb{S}_{n,d}$ with transition probabilities $p^i_{GG'}$. Eq. (4.4) implies that the stationary distribution of $\{G(t)\}_{t\in\mathbb{N}}$ also satisfies the balance equations of every chain $\{G_n^i(t)\}_{t\in\mathbb{N}}$, for $i \in V$; this is why we refer to $\{G(t)\}_{t\in\mathbb{N}}$ as "vertex" balanced. Again, as the chain $\{G_n^i(t)\}_{t\in\mathbb{N}}$ may not be irreducible, ν may not be the unique solution of its balance equations. If, however, $\{G_n^i(t)\}_{t\in\mathbb{N}}$ is irreducible and $\{G(t)\}_{t\in\mathbb{N}}$ is vertex balanced, then ν is also the unique stationary distribution of $\{G_n^i(t)\}$.

4.2.4 Markovian Graph Models Independent of Churn

In general, one may assume that the Markov chain describing transitions of the overlay graph does not depend on the churn sequence. Such a chain can be thought of as the special case of a churn-driven Markov chain over $\mathbb{S}_{n,d} \subseteq \mathbb{MG}_{n,d}$ whose conditional transition probabilities satisfy the following equations:

$$p^i_{GG'} = p_{GG'}, \quad \text{for all } i \in V_n,\ G, G' \in \mathbb{S}_{n,d}, \tag{4.5}$$

i.e., the transition of the overlay graph does not depend on the peer being replaced. Eq. (4.5) implies that every Markovian graph model independent of the churn sequence is, by definition, vertex reversible. Although mathematically sound, such models are not realistic. In particular, they do not represent the behaviour of a peer-to-peer system in which an arriving or a departing peer executes a connection protocol, thereby affecting the overlay graph only on this peer's neighbourhood.

The Independent Graph Model

An even more restricted case of a Markovian graph model is one in which, at each transition, the overlay graph is sampled independently from $\mathbb{S}_{n,d}$. We call this the *independent graph model*. Formally, let again $\{G(t)\}_{t\in\mathbb{N}}$ be the graph after the t-th arrival departure event. Then, under the independent graph model, $\{G(t)\}_{t\in\mathbb{N}}$ is an i.i.d. sequence of random d-regular multigraphs whose vertex set is V_n. That is, each $G(k)$, $k \geq 0$ is sampled from a distribution over the set $\mathbb{S}_{n,d} \subseteq \mathbb{MG}_{n,d}$, independently of previous samples.

Expressed as a churn-driven Markovian model, the independent model has transition prob-

abilities that satisfy the equations

$$p^i_{GG'} = p_{G'}, \quad \text{for all } i \in V_n,\ G, G' \in \mathbb{S}_{n,d}.$$

As such, it is trivially ergodic, vertex reversible, and its stationary distribution is p_G, $G \in \mathbb{S}_{n,d}$.

The Static Graph Model

Perhaps the simplest and most restricted Markovian model is the *static graph model*. In this model the overlay graph can be represented at any point in time as a static graph, *i.e.*

$$G(t) = G \in \mathbb{MG}_{n,d}, \text{for all } t \in \mathbb{N}.$$

In a system modelled by the above process, whenever a peer departs, it is immediately replaced by a new peer that connects to precisely the same neighbourhood. Expressed as a churn-driven Markovian graph model, it is an independent graph model, in which the graph is sampled from a distribution whose support is only the set $\mathbb{S}_{n,d} = \{G\}$.

4.3 Examples of Markovian Graph Models

The churn-driven Markovian model we described above is fairly generic, and encompasses several models of unstructured systems proposed in literature. The purpose of this section is to present these previous models in detail, and indicate their relationship to our general framework. Almost all the models and results presented below are prior work [LS03, CDG05, FGMS06, Tay81, MS05, GMS04]. The only two exceptions are the churn-driven switch model and the switch model restricted to $\mathbb{MI}_{n,d}$; both are presented at the end of Section 4.3.2.

4.3.1 Law and Siu Model

The first model we discuss was first proposed by Law and Siu [LS03], and can be described in terms of a distributed connection protocol followed by peers as they arrive in (or, depart from) a peer-to-peer system.

We assume that, at any point in time, the overlay graph of the peer-to-peer system consists of $d/2$ superimposed Hamiltonian cycles. In other words, the overlay graph belongs to $\mathbb{MH}_{n,d}$, the set of d-regular multi-graphs having a complete Hamiltonian decomposition, as defined in Section 3.3.2. For each Hamiltonian cycle k, where $k = 1, \ldots, d/2$, peer i is connected to two

other peers, its predecessor $pred_k(i)$ and its successor $succ_k(i)$ (not to be confused with the successors and predecessors in a successor sequence).

When a new peer i' enters the system, it joins each of the d cycles as follows: for each cycle k, $k = 1, \ldots, d$, peer i' picks a node j at random and becomes its successor on the cycle, while also becoming $succ_k(j)$'s predecessor. Departures are handled in a way that reverses the above process: when a node i leaves, each of its d predecessors re-connects to the respective successor of i, thereby closing the "opening" in the cycle created by i's departure.

Observe that, if the overlay graph is a d-regular multi-graph and has a complete Hamiltonian decomposition prior to an arrival or a departure, it will maintain these properties after the arrival or departure takes place. An example of an arrival in which the incoming peer follows the Law and Siu connection protocol is shown in Figure 4.2.

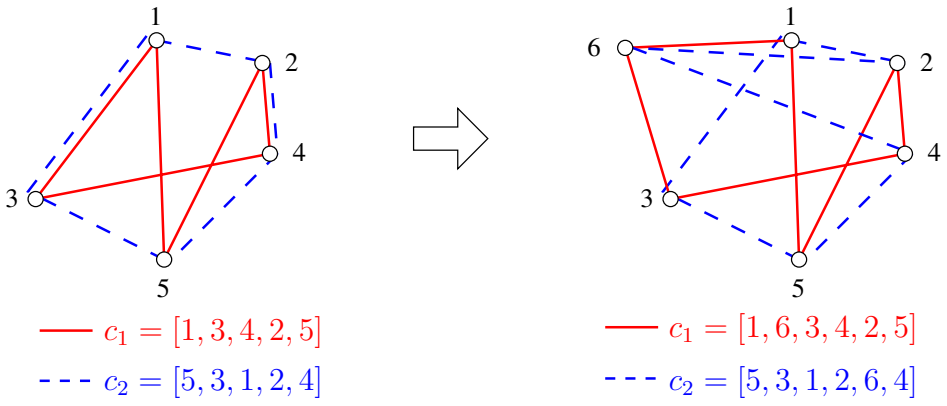


Figure 4.2: An example of the Law and Siu connection protocol [LS03] for $d = 4$. At each point in time, the overlay graph consists of two Hamiltonian cycles, c_1 and c_2. For each of the two cycles, an arriving peer (here, peer 6) chooses two existing peers at random (here 1 and 2, respectively) and becomes their successor. When departing, the process is reversed: whenever a peer departs, each of its predecessors connect to its respective successors. As a result, the overlay graph has, at all times, a complete Hamiltonian decomposition.

Using the Law and Siu connection protocol at the simultaneous departure/arrival events of our model yields a churn-driven Markovian model with state space $\mathbb{S}_{n,d} = \mathbb{MH}_{n,d}$. Each Hamiltonian cycle can be represented as a permutation σ_k over the vertex set V_n, where, $k = 1, \ldots, d/2$. Under this convention, the edge set of the overlay graph can be represented as $[\sigma_1, \ldots, \sigma_{d/2}]$.

A departure/arrival event at peer i can be effectively seen as an "operation" that changes σ_k in the following way: first, peer i is removed from σ_k completely, resulting in a permutation

over $V_n \setminus \{i\}$. Then, i is re-inserted in this permutation at one of the n different positions possible, with equal probability, resulting in a new permutation σ'_k over V_n. Formally, the probability that σ_k is transformed to σ'_k when a departure arrival event takes place at peer i is

$$p^i_{\sigma_k\sigma'_k} = \begin{cases} \frac{1}{n}, & \text{if } \sigma'_k \text{ can be constructed from } \sigma_k \text{ by moving } i \text{ to a different position} \\ & \text{in the permutation, while leaving all other elements unchanged, and} \\ 0, & \text{otherwise.} \end{cases} \tag{4.6}$$

Then, conditioned on the event that a departure/arrival event occurs at the i-th peer (*i.e.*, $I(t) = i$), the transition probabilities of the churn-driven Markov chain can be written as

$$p^i_{GG'} = p^i_{\sigma_1\sigma'_1} \cdot p^i_{\sigma_2\sigma'_2} \cdot \ldots \cdot p^i_{\sigma_{d/2}\sigma'_{d/2}}$$

where $[\sigma_1, \ldots, \sigma_{d/2}]$ the edge set of G and $[\sigma'_1, \ldots, \sigma'_{d/2}]$ the edge set of G', and $p^i_{\sigma_k\sigma'_k}$ given by (4.6).

It is easy to show that the resulting embedded Markov chain $\{G(t)\}_{t\in\mathbb{N}}$ is irreducible, as any permutation σ_k can be transformed to any other permutation σ'_k by a sequence of "operations" outlined above. Moreover, it is aperiodic, as self-transitions occur with non-zero probability. Hence, $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic. Moreover, its transition probabilities satisfy the property that

$$p_{GG'} = p_{G'G}, \quad \text{for all } G, G' \in \mathbb{MIH}_{n,d}.$$

As a result, its unique stationary distribution is the uniform distribution over $\mathbb{MIH}_{n,d}$. Moreover, the above equalities also imply that $\{G(t)\}_{t\in\mathbb{N}}$ is vertex reversible and, hence, it is also reversible and vertex balanced.

The Law and Siu protocol is appealing because it can be implemented in a distributed way, provided that arriving peers can select $d/2$ peers uniformly at random. In particular, peers do not need to have full knowledge of the Hamiltonian cycles they belong to: the protocols for arrivals and departures can be implemented as long as each peer maintains the list of its predecessors and its successors. *E.g.*, when peer i is contacted by a new peer j, requesting to become i's successor in a certain cycle k, peer i can respond by providing j with $succ_k(i)$, and setting j as its new successor. Peer j can then contact i's former successor and inform it that j will be its new predecessor. Similarly, peer i can depart by appropriately notifying its successors and predecessors in each cycle. Moreover, the number of messages exchanged per arrival and departure are $O(d)$. As a result, the average maintenance traffic that a peer has to sustain is small (it is $\mathrm{O}(1)$, in n). Finally, another reason why the above connection protocol

is appealing is that, as implied by Theorem 3.9, graphs sampled uniformly from $\mathbb{MH}_{n,d}$ are expanders *w.h.p.* for $d > 5$. As we will see, the fact that the stationary distribution of the chain $\{G(t)\}_{t\in\mathbb{N}}$ exhibits this property will be very useful.

It is assumed in the definition of the above connection protocol that an arriving peer can somehow obtain a uniform sample of $d/2$ peers in the system. In reality, this is no simple matter. In a real peer-to-peer system, peers could "bootstrap" to the system by first contacting a server, that could provide them with such a random sample. This solution is unappealing however, as it requires the existence of this central server, which has to handle $\Theta(n)$ messages per second. Gkantsidis *et al.* [GMS04] suggest maintaining $d/2$ servers, that sample peers infrequently by performing random walks, which mitigates the problem but only by a constant factor. In systems like Gnutella, peers maintain a cache of peers they had seen the last time they entered the system; this is not guaranteed to always work, however, as these peers may not be present in the system at the time the new peer enters, and the resulting sample obtained thus may not necessarily be uniform.

Law and Siu surpass this obstacle by performing random walks on the overlay graph. Assuming that each incoming peer can "bootstrap" by connecting to at least one active peer, it can initiate $d/2$ random walks over the overlay. If these random walks are stopped after performing $\Theta(\log(n))$ steps, by Theorem 3.4, their distribution will be close to uniform, precisely because the overlay graph is an expander. Interestingly, Law and Siu show that if $G(0)$ is sampled uniformly from $\mathbb{MH}_{n,d}$, and all subsequent joins sample the $d/2$ peers through random walks, the resulting graph after any number of join operations will still be an expander, with high probability. There is a caveat however; the number of messages generated per arrival by using such random walks will be $\Theta(d \log n)$, so the average maintenance traffic load per peer will also be of the order of $\log n$.

4.3.2 Switch Model

Given two edges (i, j) and (k, ℓ), where i, j, k, ℓ are distinct, a *switch* [CDG05, FGMS06] is an operation shown in Figure 4.3. This operation removes edges (i, j) and (k, ℓ) and replaces them by (i, ℓ) and (k, j), provided that the resulting graph remains simple. The switch operation preserves the degree of each vertex and, as a result, using such operations, we can define a Markovian graph model over $\mathbb{G}_{n,d}$.

Though the switch operations have been used to model changes in a peer-to-peer overlay [CDG05], such operations do not immediately correspond to changes due to arrivals and

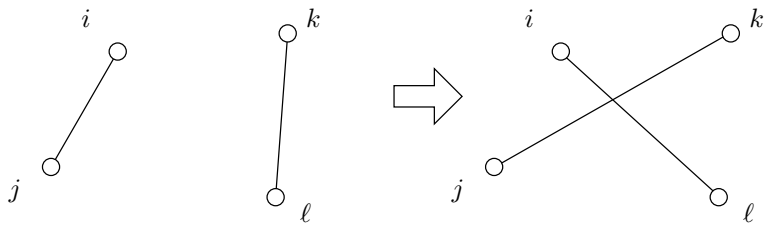


Figure 4.3: An example of a switch between two edges.

departures incurred by a specific connection protocol followed by peers. Nonetheless, Feder *et al.* [FGMS06] claim that existing peer-to-peer systems occasionally use the switch operation to "randomize" their existing connections.

Formally, a switch Markov chain is a Markovian graph model with state space $\mathbb{S}_{n,d} = \mathbb{G}_{n,d}$ and the following transition probabilities. With probability $1/2$, a transition from $G(V_n, E) \in \mathbb{G}_{n,d}$ is back to $G(V_n, E)$ (*i.e.*, the graph remains unaltered). Otherwise,

1. select two non-adjacent edges (i, j) and (k, ℓ) uniformly at random,
2. generate a perfect matching M on the vertices i, j, k, ℓ and
3. if $M \cap E = \emptyset$ then delete the edges (i, j), (k, ℓ) and replace them with M; otherwise, do nothing (*i.e.*, return back to G).

Cooper *et al.* [CDG05] show that the above Markov chain $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and reversible, and that its stationary distribution is uniform over $\mathbb{G}_{n,d}$. As such, the above overlay graph is unstructured, according to the definition we gave in Section 4.2.2. Although this is not important in our analysis, it is interesting to note that Cooper *et al*. also show that the mixing time τ_ϵ (see Section 3.2.2) of the above chain is of the order of $O\left(d^{17}n^7 \log(dn\epsilon^{-1})\right)$, *i.e.*, polynomial in n.

An application of the switch operation can disconnect the graph; restricting switch operations to only those that do not disconnect the graph leads to an ergodic chain with a stationary distribution that is uniform over the connected graphs in $\mathbb{G}_{n,d}$ [Tay81]. By Theorem 3.11, the restricted chain is also unstructured, according to our definition. Feder *et al.* [FGMS06] show that this chain too has a polynomial mixing time $\tau'_\epsilon = O\left(d^{34}n^{36}\tau_\epsilon\right)$, where τ_ϵ the mixing time of the unconstrained chain over all d-regular graphs.

We note that, by Theorem 3.7, both the simple switch model and its restriction to connected d-regular are expanders *a.a.s.*

A Churn Driven Switch Model

The above model is not churn-driven, in the strict sense. Of course, it can be converted to a churn-driven graph model in the trivial way, by assuming that $p^i_{GG'} = p_{GG'}$, for all i. A more interesting churn-driven Markov chain can be constructed by requiring that one of the edges be incident to the peer being replaced. In this churn-driven Markov chain, $p^i_{GG'}$ can be described as follows:

First, with probability $1/2$, a transition from $G(V_n, E) \in \mathbb{G}_{n,d}$ leaves G unaltered. Moreover, with probability $1/2$:

1. Select one of the edges (i, j) adjacent to i uniformly at random among the d edges of i,
2. select one of the edges (k, ℓ) uniformly at random from the $nd/2 - 2(d-1)$ edges not adjacent to (i, j),
3. generate a perfect matching M on the vertices i, j, k, ℓ and
4. if $M \cap E = \emptyset$ then delete the edges (i, j), (k, ℓ) and replace them with M; otherwise, do nothing (*i.e.*, return back to G).

The above churn-driven process indeed describes the switch chain.

Lemma 4.2. *If the churn sequence $\{I(t)\}_{t\in\mathbb{N},t>0}$ is a sequence of independent and uniform random variables over V_n, the above churn-driven Markov chain $\{G(t)\}_{t\in\mathbb{N}}$ has the same transition probabilities as the switch chain.*

Proof. To show this, it suffices to prove that, given that i is uniform over V_n, the pair of edges (i, j) and (k, ℓ) selected by the churn-driven chain is uniformly distributed among all non-adjacent pairs. This is indeed true. For each edge in $e \in E$, there are exactly $nd/2 - 2(d-1)$ edges non-adjacent to e. Therefore the total number of pairs of non-adjacent edges is

$$\frac{nd}{2} \cdot \left(\frac{nd}{2} - 2(d-1) \right) \cdot \frac{1}{2}.$$

On the other hand, the probability that peer i is replaced is $\frac{1}{n}$, while the probability that, given that peer i is replaced, the edges selected are (i, j), (k, ℓ) is

$$\frac{1}{d} \cdot \frac{1}{\frac{nd}{2} - 2(d-1)},$$

A pair of non-adjacent edges (i, j), (k, ℓ) may be selected when one of i, j, k, ℓ is being replaced. Thus, any such given pair is chosen with probability

$$4 \cdot \frac{1}{n} \cdot \frac{1}{d} \frac{1}{\frac{nd}{2} - 2(d-1)},$$

which is uniform. □

A Switch Model Restricted to $\mathbb{MI}_{n,d}$

Provided that n is even, one can use the switch operation to define a Markovian graph model with state space $\mathbb{MI}_{n,d}$, the set of d-regular graphs with a 1-factorization, as defined in Section 3.3.2.

A graph in $\mathbb{MI}_{n,d}$ consists of d superimposed perfect matchings; each one can be seen as a graph in $\mathbb{G}_{n,1}$ —*i.e.*, a 1-regular graph. We can define a Markov chain over $\mathbb{MI}_{n,d}$ by applying independent switch operations to each of the d matchings. Equivalently, the Markov chain can be seen as d independent switch chains, each having $\mathbb{G}_{n,1}$ as a state space.

By Cooper *et al.*, each of these chains is ergodic and has a uniform stationary distribution over all perfect matchings. The independence of transitions can be used to show that the stationary distribution of the joint chain in $\mathbb{MI}_{n,d}$ has a product form. The product distribution however is the uniform distribution over $\mathbb{MI}_{n,d}$ (as this is, in effect, $\mathbf{P}^{\mathcal{MI}_{n,d}}$, where $\mathcal{MI}_{n,d}$ the random multi-graph introduced in Section 3.3.2).

Note that, by Theorem 3.9, the switch model restricted to $\mathbb{MI}_{n,d}$ is an expander *w.h.p.* if $d > 5$.

4.3.3 Flip Model

As Feder *et al.* [FGMS06] point out, the assumption that peers can switch one of their edges with any random peer in the network is quite strong. Mahlmann and Schindelhauer [MS05] introduce a *flip* operation, seen in Figure 4.4, that resembles a switch but exhibits a certain "locality". The flip differs from a switch in that, for a switch between two edges to take place, it is required that one of the endpoints of the first edge be adjacent to one of the endpoints of the second edge (or, that the two edges are connected by a path of length one).

Contrary to the switch operation, a flip operation cannot disconnect the overlay graph. Mahlmann and Schindelhauer show that the Markov chain defined using the flip operation instead of the switch operation is an ergodic, reversible Markov chain over $\mathbb{CG}_{n,d}$, the set of connected d-regular graphs. Moreover, its stationary distribution is uniform over $\mathbb{CG}_{n,d}$.

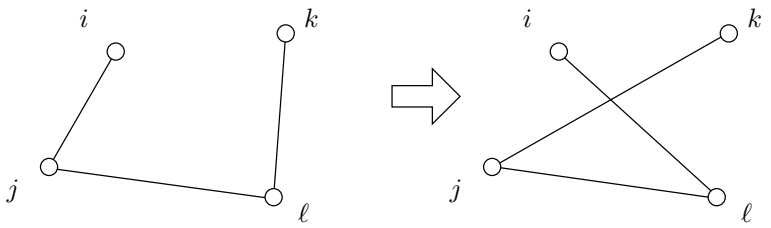


Figure 4.4: An example of a flip between two edges. Contrary to a switch, this operation can take place only if one of the vertices incident to edge (i, j) is adjacent to one of the vertices incident to (k, ℓ).

As noted earlier, the switch chain constrained to connected graphs has a polynomial mixing time. One would expect that the "locality" of the flip operation makes the mixing time of the flip chain much worse. Feder *et al.* [FGMS06] showed however that it too is polynomial; in fact, the mixing time τ_ϵ'' of the flip chain satisfies

$$\tau_\epsilon'' = O\left(d^7 n^9 \tau_\epsilon'\right),$$

where τ_ϵ' the mixing time of the switch chain constrained to connected graphs. A tighter bound on τ_ϵ'' was later obtained by Cooper et al. [CDH09].

Again, by Theorems 3.7 and 3.11, the overlay graph defined by the flip operation is unstructured and also an expander *a.a.s.*.

4.3.4 On Almost-Uniform Stationary Distributions

The Markovian graph models we described in Section 4.3 were conceived as models of the overlay graph of an unstructured peer-to-peer system [LS03, FGMS06, CDG05, MS05]. All of them lead to overlay graphs that are expanders *a.a.s.*, under the definition on the steady state relaxation time we gave in Section 4.2.2. This is not coincidental; as discussed in Section 3.3.2, any probability measure that is contiguous to sampling uniformly over $\mathbb{G}_{n,d}$ leads to a bounded relaxation time *a.a.s.* Simply put, as long as the stationary distribution of the Markov process $\{G(t)\}_{t\in\mathbb{N}}$ is uniform over a "large enough" set $\mathbb{S}_{n,d}$ (in the sense of contiguity to uniform sampling over $\mathbb{G}_{n,d}$), the resulting overlay graph will be an expander *a.a.s.*

Observing the above models can give us intuition on how connection protocols can yield an "almost uniform" stationary distribution. Roughly, it suffices that the connection protocol, followed during arrivals and departures, exhibits the following two properties. First, it is "rich enough", in the sense that repeated arrivals and departures can transform any graph $G \in \mathbb{G}_{n,d}$

to any possible graph from a large subset of $\mathbb{G}_{n,d}$. Second, it exhibits a certain "symmetry": *E.g.*, it suffices that

$$p_{GG'} = p_{G'G}, \text{for all } G, G' \in \mathbb{S}_{n,d}$$

for the stationary distribution to be uniform over $\mathbb{S}_{n,d}$.

An important and interesting observation is that both of these two requirements can hold even if the operations during an arrival or a departure are "local". An intuitive example of this is the flip model: its distribution is uniform over a large subset of $\mathbb{G}_{n,d}$ although changes are indeed "local". In particular, only edges that are within one-hop distance can be "flipped". Choosing edges that are far apart (as in the switch model) may speed up the convergence of the Markov process to its stationary distribution; however, this does not play a role on whether the stationary distribution is uniform over a large set of graphs. In some sense, the above observation is analogous to the fact that the random walk of every connected, d-regular graph has a uniform distribution over the vertex set (see Eq. (3.7) and the discussion right below it), irrespectively of whether it mixes fast or not.

4.4 Content Evolution

In this section, we describe the process through which peers request or share data. We refer to files as data items (or, simply, items), to stress that the data shared can have a generic form. For example, the items shared can be answers to a data-base query, web search results, *etc.*

A newly arriving peer may issue requests for data items stored in the peer-to-peer system. Moreover, peers may also publish data items at the time they arrive, *i.e.*, make them publicly available and share them with other peers. Below, we describe how these two processes, *i.e.*, requesting data items and publishing them, are modelled in our work. An important feature of our model is that it allows items to be requested very often but published rarely, and vice-versa. As discussed below, such behaviour captures the discrepancy between file popularity and availability observed in measurements of real peer-to-peer systems.

We present two different models. The first describes our assumptions in a pure peer-to-peer system, where no server exists. The second describes our assumptions in a hybrid system, in which a server coexists with the peer-to-peer system. The server stores all possible data items that may be requested by a peer. As discussed in Chapter 1, in the pure peer-to-peer system, if a query fails to locate a copy of the requested data item, the peer issuing it does not retrieve a copy. In the hybrid system, a peer can redirect its query to the server; as a result, peers always

retrieve the data items they request.

Our model of the data request and publishing processes will differ only slightly between the pure and hybrid cases; nonetheless, for the sake of clarity, we present them separately.

4.4.1 Pure Peer-to-Peer System

Data Item Popularity and Availability

Recall, from Section 2.2.3, that in an unstructured peer-to-peer system there might be a discrepancy between a data item's "popularity", *i.e.*, the fraction of peers that request it, and its "availability", *i.e.* the fraction of peers that share it. To capture the above aspect of unstructured peer-to-peer systems in our analysis, we associate two quantities with each item: the *publishing probability* q and the *request probability* p. The request probability models precisely the item's popularity, as it also equals the fraction of new peers that request it. The publishing probability captures the fraction of peers that bring the item into the system when they enter; as such, an item's availability is at least q but can be as high as $q + p$ (in expectation), depending on the outcomes of searches.

In our analysis, we treat all arriving peers as new peers; in reality, peers that publish the data item can in fact be returning peers, that downloaded it sometime in the past. In this sense, q can be interpreted as the likelihood that a peer downloaded the item in the past *and* is willing to share it.

More precisely, we assume that, in a system of size n, peers may issue queries for M_n distinct data items, that may or may not be present in the system at the time a new peer arrives. A data item j, $j = 1, \ldots, M_n$, is brought into system (*i.e.*, published) by a new peer with a probability q_n^j and is requested by an incoming peer with a probability p_n^j. Thus, for each item $j = 1, \ldots, M_n$, an arriving peer does exactly one of the following actions:

(a) It brings (publishes) the item j, with probability q_n^j, or

(b) it requests the item j, with probability p_n^j, or

(c) it does neither of the above, with probability $1 - q_n^j - p_n^j$.

We assume that each peer performs one of the above actions independently of any other event, *e.g.*, of the choices made by other peers, the choices the same peer makes with respect to other data items, the items in the system at the time of its entry, *etc.* Note that we allow both the

publishing probability q_n^j and the request probability p_n^j to be functions of the system size n; this dependence is discussed in more detail below.

We assume that a peer that successfully retrieves an item shares it for the remainder of the time that it spends in the system. Note that the expected number of items an incoming peer publishes is given by $\sum_{j=1}^{M_n} q_n^j$, while the expected number of queries it issues is $\sum_{j=1}^{M_n} p_n^j$. Moreover, at any point in time, the expected number of peers in the system that publish item j is nq_n^j, while the expected number of peers that request it is np_n^j. Thus, the overall expected number of peers storing item j at any given time is at most $n(q_n^j + p_n^j)$.

In our analysis, we will assume that query propagations are conducted independently; for example, a peer searching for k data items will initiate k different random walks. As a result, without loss of generality, we can focus in our analysis on a single item; this is because the metrics we are interested in can be evaluated on a per-item basis. For example, the total traffic load on a peer generated by all M_n types of queries can be obtained by summing the individual loads generated per item. For this reason, we will omit the index j and denote the request and the publishing probabilities by p_n and q_n, respectively.

When performing a single item analysis, we refer to a peer as *positive* if it has a copy of the item, and as *null* if it does not have the item. In a pure peer to peer system, a peer requesting the item can become either null or positive, depending on whether the query succeeds in locating a positive peer or not.

Dependence on System Size

Recall that the size parameter n captures the growth of the peer-to-peer system, occurring over long periods of time (*e.g.*, months or years). Therefore, allowing probabilities p_n and q_n to be functions of n aims to capture the long-term changes of the popularity or the availability of a data item, as the system size grows. To gain some intuition as to what this dependence might mean, recall that, in a system of size n, the expected number of peers that requested the item upon their arrival is equal to np_n, and the expected number of peers that bring the item into the system is nq_n. If, *e.g.*, $q_n = 0.01$, then 1% of all peers publish the item, in expectation. The expected number of peers that brings the item into the system is then $0.01n$ and grows linearly with the system size. If $q_n = c/n$, where $c > 0$ a constant, only c peers bring the item into the system, in expectation. For example, this would be the case if the item is published only by a group of limited size, that does not grow as the overall population in the network increases. In the case where q_n decays faster than $1/n$, *e.g.*, $q_n = 1/n^2$, the expected number of peers that

bring the item into the system *decreases* as the system size grows, indicating that fewer peers publish it. Similar observations can be made regarding p_n with respect to the number of peers that request the item.

To avoid trivial system behaviour, our standing assumption throughout our analysis will be that each of the three events described above (requesting, publishing, or simply ignoring a data item) have a non-zero probability of occurring, *i.e.*,

$$0 < p_n < 1, \qquad 0 < q_n < 1, \qquad 0 < p_n + q_n < 1.$$

The assumption of $q_n > 0$ is of particular significance in the pure peer-to-peer case. In a system in which no peer publishes the data item there will be a time (finite, with probability one) after which no peer will ever store the item. After this time, all subsequent queries for the item will fail. Hence, the requirement that $q_n > 0$ is essential for the system behaviour to not be trivial.

Request Epochs

For simplicity, we assume that all data items are requested by a peer immediately when it enters the system. It would be more realistic to allow peers to issue their requests at some random epoch within the time they spend in the system. However, our assumption greatly simplifies our analysis without considerably affecting our results. For example, in Section 5.6, we simulate a system in which all queries are issued at times chosen uniformly at random within a peer's lifetime. This system's behaviour is very similar to the behaviour of a system described by our model in which the peer lifetimes are one half of their true value. In this sense, if request epochs were chosen uniformly at random, metrics such as the average traffic load per peer would only be altered only by a constant factor of two. As our focus is on the asymptotic behaviour of such metrics in terms of n, such an effect can essentially be ignored.

4.4.2 Hybrid Peer-to-Peer System

Our modelling assumptions for the hybrid case are almost identical to the ones employed in the pure peer-to-peer system. Again, we assume that, in a system of size n, the server stores M_n data items, and that an incoming peer requests or publishes an item j with probability p_n^j and q_n^j, respectively. We again focus, without loss of generality, on a single item. As such, in the single item analysis, we will drop the index j, simply referring to the request and publishing probabilities as p_n and q_n. Again, in the single item analysis, we refer to a peer as positive if

it has a copy of the item and as null, otherwise. A peer requesting the item always becomes positive in a hybrid peer-to-peer system, as it can always retrieve the item from the server.

The only major difference in the modelling assumptions on the data request and publishing processes that we make in the hybrid case is that we allow the publishing probability q_n to be zero. In particular, although we assume that

$$0 < p_n < 1, \qquad 0 < q_n + p_n < 1,$$

we also consider cases in which $q_n = 0$ and no peer publishes the item. Contrary to the pure peer-to-peer case, this assumption does not lead to a degenerate system in the hybrid setting. This is because of the existence of the server, which always has a copy of the item.

In some sense, the case in which $q_n = 0$ is a "worst-case" scenario in the hybrid model: a system in which peers publish data items will sustain both lower query traffic on the peers and the server compared to a system with $q_n = 0$. We further elaborate on this issue in Section 5.7.5.

4.5 Summary

Many of our modelling assumptions (*i.e.*, the invariability of the system size, the differentiation between request and publishing probabilities, and the regularity of the overlay graph) are directly motivated by the results of the measurement studies presented in Section 2.2. Our model takes into consideration the variability of a peer-to-peer system through time. Most importantly, our model leads to a concise and general definition of an unstructured system, expressed in terms of the stationary distribution of the overlay graph.

In the next two chapters, we will show how the above model can be used to discuss the scalability of search mechanisms in a hybrid system, and the scalability and reliability of search mechanisms in a pure peer-to-peer system. A summary of the notation of Chapter 4 that will be used in these two chapters can be found in Table 4.1.

Table 4.1: SUMMARY OF MODEL PARAMETERS APPEARING IN CHAPTER 4

n	The system size.
$1/\mu$	The expected peer lifetime.
$\{G(t)\}_{t\in\mathbb{R}_+}$	The overlay graph process.
$\{G(t)\}_{t\in\mathbb{N}}$	The embedded Markov chain of the overlay graph process.
$\mathbb{S}_{n,d}$	The state space of $\{G(t)\}_{t\in\mathbb{R}_+}$.
$p^i_{GG'}$	The transition probabilities of $\{G(t)\}_{t\in\mathbb{N}}$, conditioned on $I(t)=i$.
$p_{GG'}$	The (unconditional) transition probabilities of $\{G(t)\}_{t\in\mathbb{N}}$.
τ_n	The steady state relaxation time of the overlay graph.
V_n	The set of vertices $\{1,2,\ldots,n\}$.
$\{I(t)\}_{t\in\mathbb{N},t>0}$	The churn sequence.
d	The degree of the overlay graph.
p_n	The request probability.
q_n	The publishing probability.

Chapter 5

Hybrid Peer-to-Peer System

Having introduced our model, we turn our attention to the analysis of a hybrid peer-to-peer system[1]. In this chapter, we will use our model of Chapter 4 to analyze the scalability of two query propagation mechanisms, namely, the random walk and the expanding ring, in the hybrid setting.

The most important property of the hybrid setting is that the peer-to-peer system coexists with a central server. The purpose of this hybrid architecture is to assist the server in the distribution of content by delegating part of its traffic load to the peers. In particular, incoming peers are interested in retrieving some of the content stored at the server. However, instead of requesting this content directly, they first propagate a query over the peer-to-peer network. If the query reaches a peer that has already retrieved the requested content, it is resolved without the intervention of the server. If, however, the query fails, it is redirected to the server.

We note that, as content can always be retrieved from the server, a hybrid system is inherently reliable (*i.e.*, a query always succeeds at retrieving the requested content). This motivates our focus on scalability, captured by the query traffic load at peers and the query traffic load at the server. Ideally, we would like to design our system so that both of these two loads are low. This is because achieving a low server traffic load is the reason for deploying the hybrid architecture in the first place while, on the other hand, query traffic should not overwhelm peers, as their bandwidth is typically limited.

We formally show that a hybrid peer-to-peer system whose query propagation mechanism is the random walk scales extremely well in terms of both of the above two traffic loads. In particular, a random walk that stops after a time that is proportional to the peer population

[1] An earlier version of the work presented in this chapter appeared in [IM08].

incurs a traffic load at the server and at individual peers that stays bounded as the peer population grows. This result is surprising and has a very important implication: it shows that it is possible to construct a hybrid peer-to-peer system that can handle query traffic generated by a large (unbounded) number of peers even when the bandwidth capacities of both the server and individual peers are limited.

We also show a similar result for a hybrid system whose query propagation mechanism is the expanding ring. In particular, if an expanding ring with a stopping time that is logarithmic in the size of the peer population is used, both the server load and the traffic load at individual peers grows very slowly as the peer population grows. The rate of growth is very small, to the extent that both traffic loads can effectively be considered constant, for all practical purposes. As such, our result implies that a hybrid system using an expanding ring also exhibits excellent scalability properties. Most importantly, as the stopping time of the expanding ring is logarithmic in the peer population size, the system also exhibits much lower query response times than the random walk.

The remainder of Chapter 5 is structured as follows. In Section 5.1, we briefly review our model of the hybrid peer-to-peer system. The full description of this model can be found in Chapter 4; Section 5.1 serves as a reminder of the model's basic assumptions. In addition, this section includes a detailed description of the random walk and the expanding ring query propagation mechanisms as well as the metrics we use to evaluate their performance. The latter are the server traffic load, the average traffic load per peer and the expected query response time.

The chapter's two main results (namely, Theorems 5.1 and 5.2) are stated in Section 5.2. Their proofs appear later in the text (in Sections 5.4 and 5.5, respectively). The focus of Section 5.2 is on discussing their importance and their implications.

In Section 5.3, we show that the evolution of the hybrid system is described by a Markov process that does not depend on the query propagation mechanism used. We establish the conditions under which (a) this process is ergodic and (b) the limits describing the average traffic load per peer and the server traffic load, as defined in Section 5.1, indeed exist.

The analysis of the random walk and the expanding ring is presented in Sections 5.4 and 5.5, respectively. In both cases, we first treat the average load per peer and the server load separately. We then combine the results on each of these two performance metrics to prove our two main theorems.

In Section 5.6, we validate our analysis through simulations. Even though several of our modelling assumptions are relaxed in these simulations, the behaviour we observe is consistent

with the one predicted by our theoretical analysis.

Finally, in Section 5.7, we discuss several possible immediate extensions of our results, as well as open problems. Our main focus is on generalizations to non-expander topologies and to query propagation mechanisms other than the random walk and the expanding ring.

5.1 Model

In this section, we briefly summarize the key assumptions of the model introduced in Chapter 4. We then describe the two query propagation mechanisms that we will analyze, namely, the random walk and the expanding ring, as well as the metrics that we employ to evaluate their performance.

Recall, from Chapter 4, that we assume that the system consists of n peers, at any point in time. Each peer stays in the system for an exponentially distributed time with mean $1/\mu$ and, upon departure, it is immediately replaced by a new peer; this keeps the system size constant and equal to n at all times.

The overlay graph of the peer-to-peer system evolves according to a churn-driven Markovian graph model $\{G(t)\}_{t\in\mathbb{R}_+}$, whose state space is some set $\mathbb{S}_{n,d} \subseteq \mathbb{MG}_{n,d}$ (see also Section 4.2.1). We denote by τ_n the relaxation time of a random graph sampled from $\mathbb{S}_{n,d}$ according to the stationary distribution of $\{G(t)\}_{t\in\mathbb{N}}$. Recall that τ_n is a random variable in $\mathbb{R}_+ \cup \{\infty\}$ whose distribution is given by (4.1).

Arriving peers may issue requests for data items (*e.g.*, files, websites, *etc.*) on the server or publish their own data items. Without loss of generality, we focus on the dynamics of a system in which the server stores —and peers share— only a single data item (see also Section 4.4). Incoming peers request the item with probability p_n and publish it with probability q_n, where

$$0 < p_n < 1, \qquad 0 < p_n + q_n < 1.$$

In our analysis below, we will first assume that $q_n = 0$, *i.e.*, none of the peers publishes the data item. The case where $q_n > 0$ is covered in Section 5.7.5; as discussed there, assuming $q_n = 0$ gives a worst-case performance of the system in terms of both the average traffic load at peers and the traffic load in the server.

Peers that request the data item do so immediately when they enter the network. A query issued by an incoming peer (the source) is first propagated over the peer-to-peer network. If the query succeeds in locating another peer storing the requested item, the source peer obtains

a copy without the intervention of the server. If, on the other hand, the query fails, the source peer retrieves the item directly from the server.

We refer to peers that have a copy of the item as *positive* and to peers that do not as *null*; a fundamental property of the hybrid system is that peers that request the item always become positive after the conclusion of a search.

We will analyze two query propagation mechanisms in the above system: the random walk and the expanding ring. Both are described in detail in Section 5.1.1. The three metrics of interest to us are the average traffic load per peer ρ_n, the server traffic load ρ_n^0, and the query response time D_n, all of which are defined in Section 5.1.2.

For both the random walk and the expanding ring, we assume that the total query response time is negligible compared to the lifetime of a user. In effect, we decouple the propagation of a query from the rest of the dynamics of the system: queries are instantaneous when viewed in the timescale determined by user arrivals and departures. As a result, both the overlay graph and the set of positive peers remain unchanged during the propagation of a query.

The assumption that the system does not change during the propagation of a query is not realistic. In particular, for systems with very long response times we expect this assumption to be violated and our model to be inapplicable. Nonetheless, as we show through our numerical study in Section 5.6, the results obtained under our model remain valid for a very wide range of values of n even when the topology changes while queries are propagated. Similarly, in a real peer-to-peer system, the system size would not be constant, but would oscillate around an operating value. We also relax this assumption in our numerical study and verify that our results are still valid for a system in which the system size is time-variant.

5.1.1 Random Walk and Expanding Ring

Our analysis in this chapter will focus on two query propagation mechanisms, namely, the random walk and the expanding ring. Below, we give a detailed description of both mechanisms. We note once more that, irrespective of the mechanism used, the system is assumed to be static during a query propagation, and that this assumption will be relaxed in in Section 5.6.

Random Walk

We have already discussed in great detail the mathematical concepts underlying the random walk in Chapter 3; here, we give a more systems-oriented perspective.

In the random walk query propagation mechanism, the peer issuing a query, which we call the *source* peer, chooses one of its neighbours in the overlay graph at random and forwards a query message to it. A peer that receives a query message checks if it can resolve the query, *i.e.*, whether it stores the item requested locally. If it has the item, it notifies the source peer. If not, it forwards the message to one of its neighbours, chosen again uniformly at random. No information is maintained about the peers that receive the query and a peer may receive the same query message more than once.

We assume that the time it takes to transmit a message is exponentially distributed, with mean $\delta > 0$. To restrict the number of transmissions, we employ a *time-to-live* mechanism. We consider two different versions of the time-to-live mechanism: a *hop-constrained* version, in which the number of hops of the random walk cannot exceed a certain value, and a *delay-constrained* version, in which the query response time (or, the delay) cannot exceed a certain value. The main reasons behind using two different mechanisms are technical[2]; in practise, the two mechanisms should not differ considerably with respect to the performance metrics we consider. For example the average traffic load per peer generated by a random walk with either time-to-live mechanism should be of the same order (in the number of peers in the system).

Hop-Constrained Random Walk. In a hop-constrained random walk, the header of the query message contains an integer value field, called the time-to-live field. This is set by the source peer to an initial value, which we denote by TTL. Before re-transmitting a message, a peer first decreases the value in the time-to-live field. The message is thus transmitted until either a peer storing the requested item is reached, or the TTL field becomes zero. If a peer storing the item is reached, this peer responds to the source of the query by providing it with the item. If, instead, the TTL field becomes zero, the source peer is notified that the search for the data item has failed. In the hybrid system, the failure of the search implies that the source peer will redirect its request to the server. We note that, in a pure peer-to-peer system, the peer that issued the query would simply fail to get the item.

When using a hop-constrained random walk, the number of message transmissions during one search can be no more than TTL. Note that it can be less than TTL; a peer having the item will cease the propagation of the query. Moreover, the query response

[2]Introducing the delay-constrained version of the random walk allows us to use Theorem 3.6 for bounding the probability a query does not locate a positive peer; similar bounds are hard to obtain in a closed form for the hop-constrained random walk (see also Section 3.2.3).

time is a random variable; however its expectation is no more than $\delta \cdot \mathrm{TTL}$; this is implied by the fact that the expectation of the sum of TTL random variables is equal to the sum of their expectations.

Delay-Constrained Random Walk. In a delay-constrained random walk, a query message is dropped when it has been in the system for $\delta \cdot \mathrm{TTL}$ time, where $\delta > 0$ the mean time to transmit a query. A message may be dropped during a transmission, in which case, the query is considered failed even if the recipient has a copy of the data item. The time that a query has spent in the system can be monitored by, *e.g.*, adding a time-stamp on the header of the message that indicates the time it was issued, along with the maximum response time $\delta \cdot \mathrm{TTL}$ —this assumes that peers are synchronized. Alternately, if peers are not synchronized, the header field can contain the time a query has been in the system, and each peer can increment this field appropriately before forwarding the message.

The source peer waits for a response for a period of $\delta \cdot \mathrm{TTL}$ time units. If this period expires and no response is received, the propagation is considered failed. Obviously, when using a delay-constrained random walk, the query response time can be no more than $\delta \cdot \mathrm{TTL}$ time units. On the other hand, the expected number of message transmissions is no more than TTL. Proving this requires some work; we refer the reader to Lemma 5.14 for a proof of this statement.

For both the hop-constrained and the delay-constrained random walk, we refer to TTL as the *stopping time* of the random walk. Moreover, for both of the above two versions of the random walk, we assume that TTL is a system parameter that can depend on the system size. We denote by TTL_n its value for a system of size n; that is, TTL_n is the maximum number of hops under the hop-constrained walk in a system of n peers, and $\delta \cdot \mathrm{TTL}_n$ is the maximum query response time for a delay constrained random walk in a system of n peers.

Allowing TTL_n to be a function of the system size implies that n is a-priori known to the source peer, so that it can initialize the header of the query message appropriately. As we will see, peers do not need to know the precise value of n —an estimate that is linear in n, no matter how far from the exact value (*e.g.*, $0.01n$ or $100n$), suffices for all the results we obtain. There are known distributed algorithms for obtaining such an estimate at a small overhead (see, *e.g.*, [GKLM07]). As the systems that we consider are systems of slow growth, and inaccuracy can be tolerated, such algorithms can be executed infrequently (*e.g.*, once a day); new peers can learn the computed value from their neighbours, when they enter the system.

The larger the value of the stopping time the more likely that the random walk will succeed in locating a positive peer (*i.e.*, a peer storing the item). As a result, in a hybrid system, a large value of TTL_n tends to reduce the query traffic that reaches the server while also increasing the peer traffic. Considering the function TTL_n as a design parameter, we study how TTL_n should scale as the peer population grows in order to achieve a desired trade-off between the traffic load at the server and at individual peers.

We note that in a pure peer-to-peer system the TTL_n establishes a similar trade-off between the average traffic load per peer and the query success rate (*i.e.*, the probability a query succeeds), as opposed to the traffic load on the server.

Expanding Ring

The second query propagation mechanism that we consider is an expanding ring mechanism [LCC+02]. Essentially, this is a sequence of "simple flooding" searches in which the time-to-live is incremented at each iteration; we describe what we mean by simple flooding below.

In simple flooding, a peer sends a query packet to *all* its neighbours. A peer that receives a query message forwards it only if it has not already received it in the past. A time-to-live field is used in the header of the message, and it is decremented every time a peer forwards a query packet (*i.e.*, the query propagation is hop-constrained). A peer ceases to forward a packet when the time-to-live field becomes zero.

In the expanding ring mechanism, the query propagation is implemented by flooding in several stages; in each stage, the source peer sets the time-to-live value of the query message to a higher value than the value it used in the previous stage. In particular, the peer that issues the query first searches by simple flooding with a time-to-live field initialized to one. If the query is not successful within this hop threshold, the peer increments the time-to-live by one and repeats the search. This process is repeated until either the query is resolved, or the hop threshold exceeds a given value TTL.

Just as we did for the random walk mechanism, we call TTL the *stopping time* of the expanding ring. We treat TTL as a design parameter and allow it to depend on the system size n. Again, we denote the stopping time in a system with size n by TTL_n. Similarly to the corresponding threshold in the random walk, in a hybrid system TTL_n determines the trade-off between the peer traffic and the traffic directed to the server.

In our analysis of the random walk mechanism, we assumed that the transmission of a query packet is exponentially distributed with mean δ time units. For the expanding ring mechanism,

we will assume that the transmission time of a query packet is non-random and is exactly δ time units. This assumption is again only made for technical reasons: the random walk is easier to analyze in the continuous-time realm, whereas the expanding ring is easier to analyze when hops are deterministic. In practise, both transmission models should behave similarly, both qualitatively and quantitatively.

5.1.2 Performance Metrics

We consider three different performance metrics: the average traffic load per peer, the server traffic load and the query response time. Below, we give a formal definition of these metrics in terms of our model.

Average Load per Peer

We assume that each query message that a peer receives has a cost of one unit, which accounts for the bandwidth required to receive and forward the query packet. Irrespective of the query propagation mechanism used, we define the traffic load at a peer to be the expected message cost (*i.e.*, the expected number of query messages) the peer handles per unit time. In our analysis, we focus on the *average load per peer* ρ_n, which we define as the average traffic load, in messages per second, over all peers in the system.

More formally, we associate with each vertex $i \in V_n$ of the overlay graph G_n (or, with each successor sequence) a counting process

$$\{M_i(t)\}_{t\in\mathbb{R}_+}$$

that corresponds to the number of query messages that peers at vertex i have received up to and including time t. For example, for the random walk mechanism, $\{M_i(t)\}_{t\in\mathbb{R}_+}$ is the number of times a walk has passed through or has terminated at vertex i. We formally define the load ρ_n^i at vertex i as the time-average load at i, which is

$$\rho_n^i = \lim_{t\to\infty} \frac{M_i(t)}{t}.$$

In general, the above limit may not necessarily exist. Nonetheless, assuming that the process describing our system is ergodic, the above limit exists *a.s.* and, by Blackwell's Theorem [Gal96], it will also be equal to

$$\rho_n^i = \lim_{\delta\to 0}\lim_{t\to\infty} \frac{\mathbb{E}[M_i(t+\delta)] - \mathbb{E}[M_i(t)]}{\delta}$$

i.e., the expected instantaneous traffic load at vertex i, in steady state.

We define the *average query traffic load per peer* as

$$\rho_n = \sum_{i=1}^{n} \frac{\rho_i}{n}.$$

Ideally, we would like the above load to be bounded by a constant independent of n, irrespective of the request probability p_n or the publishing probability q_n. This would suggest that the load incurred on peers due to query traffic can be sustained even if peers have a fixed amount of bandwidth resources, that does not grow with the system size. This would indicate that the system scales as the number of peers increases. In practise, cases in which ρ_n grows slowly in n are also of interest. For example, keeping in mind that system growth occurs at long-term periods of time, it may be plausible that the bandwidth resources available to peers grow with time (albeit slowly). This, for example, can occur if faster connections become more affordable over the long-term period of time within which the system size evolves. Nonetheless, even in the bounded resource scenario, a system in which ρ_n grows slowly may still be appealing, as the region of values of n for which the system can operate without exceeding the peers' bandwidth capacity may be large.

Server Load

Describing the scalability of a hybrid peer-to-peer system requires us to consider both the traffic incurred on peers as well as the traffic load the server. This load can be defined in a similar manner: assuming that each query message that is sent to the server incurs a cost of one unit, the *server load* ρ_n^0 can be defined as the query message cost per unit time on the server.

Formally, let

$$\{M_0(t)\}_{t \in \mathbb{R}_+}$$

be the counting process describing the number of messages received by the server up to and including time t. Alternatively, $\{M_0(t)\}_{t \in \mathbb{R}_+}$ can be seen as the number of times a query failed to be satisfied within the peer-to-peer system (*i.e.*, a query propagation failed to locate a positive peer). We formally define the server load ρ_n^0 as the time average of $M_0(t)$, which is

$$\rho_n^0 = \lim_{t \to \infty} \frac{M_0(t)}{t}.$$

Once more, we note that the above limit may not necessarily exist. Assuming that the process describing our system is ergodic, the above limit does exist *a.s.* and, by Blackwell's Theorem

[Gal96], it will also be equal to

$$\rho_n^0 = \lim_{\delta \to 0} \lim_{t \to \infty} \frac{\mathbb{E}[M_0(t+\delta)] - \mathbb{E}[M_0(t)]}{\delta}$$

i.e., the expected instantaneous traffic load at the server, in steady state.

Our goal is to design a system in which both the server traffic load ρ_n^0 and the average traffic load per peer ρ_n grow slowly in n. Ideally, we would like both to be bounded in n, irrespective of the request probability p_n or the publishing probability q_n. A bounded server load would imply that the server can support queries from a growing peer base without needing to expand its infrastructure. In this sense, if both ρ_n and ρ_n^0 are bounded, the hybrid system indeed serves its purpose of alleviating the server load without overloading the peers of the system.

In practise, cases in which the server load grows slowly in n are also interesting. Assuming that system growth occurs over a long term period of time, a slowly growing load indicates, *e.g.*, that the company maintaining the server can invest in upgrading its infrastructure at a slow pace.

Query Response Time

Our view of scalability focuses on the use of system resources; for this reason, we address the scalability of the system in terms of the traffic loads ρ_n and ρ_n^0. The query response time, which we define formally below, is a metric that measures the *quality* of the service offered (namely, searching) as perceived by the peer issuing the query.

Let $\{D(t)\}_{t \in \mathbb{N}, t>0}$ be the process describing the response time of the t-th query propagation. We are interested in computing the time-average query response time D_n, *i.e.*,

$$D_n = \lim_{t \to \infty} \frac{1}{t} \sum_{s=1}^{t} D(s).$$

As usual, if the system is ergodic, the above limit will exist almost surely and will equal

$$D_n = \lim_{t \to \infty} \mathbb{E}[D(t)].$$

That is, an alternative definition of D_n is the expected response time in "steady state".

Dependence on System Size

Our analysis amounts to characterizing the dependence of the performance of the above metrics on n, the system size parameter. As we will see, our analysis essentially consists of two

steps. In the first step, we fix n and consider a system evolving through time according to the aforementioned model. For this system, we compute the time-average value of the metric we are interested in (*e.g.*, the average traffic load per peer) as a function of n. Having established the dependence of this quantity on n, in the second step, we investigate how this metric behaves as n tends to infinity.

The systems that we model have a roughly constant operating size over short-term periods of time, while growing in the long-term. Hence, our first step captures the time-average (or, expected) behaviour of the system over such a short-term period of time. On the other hand, the second step captures long-term scalability, as it describes how the system will behave, *e.g.*, within a year, when the size of the system has grown considerably.

5.2 Main Results

In this section we formally state and discuss our two main results, namely, Theorems 5.1 and 5.2. Intuitively, these two theorems show that an unstructured hybrid system whose search mechanism is either the random walk or the expanding ring scales extremely well. We note again that the proofs are given later in the text (in Sections 5.4 and 5.5, respectively); this section aims at illustrating their importance and their implications.

5.2.1 Random Walk

Our first main result is that, if the overlay graph is an expander *w.h.p.*, the delay-constrained random walk on a hybrid system scales very well as the peer population increases. As discussed in Section 4.2.2, this suggests that hybrid systems with unstructured topologies have excellent scalability properties.

Theorem 5.1. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and vertex balanced, and that there exists a constant $\bar{\tau}$ such that*

$$\mathbf{P}(\tau_n \leq \bar{\tau}) = 1 - o\left(\frac{1}{n}\right),$$

where τ_n the relaxation time of a graph sampled from the stationary distribution of $\{G(t)\}_{t\in\mathbb{N}}$. Then, both the average traffic load per peer and the server traffic load for a delay-constrained random walk with $\mathrm{TTL}_n = \Theta(n)$ are bounded in n, irrespectively of p_n.

The proof can be found in Section 5.4.4. The theorem states that, if the overlay graph is an expander *w.h.p.*, both the traffic load on the server *and* the average traffic load per peer of a

random walk with $\mathrm{TTL}_n = \Theta(n)$ will remain bounded by a constant, as the peer population increases. Hence, such a hybrid system indeed serves the purpose of alleviating the traffic load at the server without creating a considerable burden on the participating peers. In fact, the server can accommodate an unlimited number of peers, even if both the server and the peers only have a limited (*i.e.*, bounded) amount of bandwidth resources.

The expanders presented in Section 3.3.2 and the churn-driven Markovian models of Section 4.3 give us several examples of overlay graphs that satisfy the conditions of Theorem 5.1. To begin with, both the Law and Siu model (Section 4.3.1) and the switch model restricted over $\mathbb{MI}_{n,d}$ (Section 4.3.2) are ergodic, vertex balanced churn-driven chains, whose stationary distribution yields expanders, *w.h.p.*, if $d > 5$. More generally, Theorem 3.9 and Corollary 3.6 imply that the conditions of the theorem hold for ergodic, vertex balanced overlay graphs $\{G(t)\}_{t\in\mathbb{N}}$ whose stationary distribution is uniform over the state space $\mathbb{S}_{n,d}$, where $\mathbb{S}_{n,d}$ is one of the sets $\mathbb{MH}_{n,d}$, $\mathbb{MI}_{n,d}$, $\mathbb{H}_{n,d}$ or $\mathbb{I}_{n,d}$ (defined in Section 3.3.2).

Most importantly, as long as the conditions of Theorem 5.1 on ergodicity and vertex reversibility hold, the result on the scalability of the random walk can be extended to any overlay graph that is an expander *w.h.p.* In this sense, given the abundance of expanders among d-regular graphs, the above theorem applies to the overlay graph resulting from any connection protocol yielding a uniform distribution over a large enough set of d-regular graphs.

Though the result presented above focuses on the case the the overlay graph is an expander, and characterizes the conditions under which ρ_n and ρ_n^0 are bounded, our analysis can be extended to cover more general cases. In particular, in our analysis, we obtain exact (*i.e.,* not only asymptotic) bounds on ρ_n and ρ_n^0, in terms of the request probability p_n, the relaxation time τ_n and the stopping time TTL_n (see Theorems 5.4 and 5.5). These bounds can be used, *e.g.*, to characterize the afforementioned traffic loads if the overlay graph is not an expander, or to characterize the conditions under which one of the two loads grows slowly, *e.g.*, as $O(\log n)$. We discuss such extensions in more detail in Section 5.7.

5.2.2 Expanding Ring

Theorem 5.1 suggests that the random walk with $\mathrm{TTL}_n = \Theta(n)$ has excellent scalability properties over a peer-to-peer system with an expander overlay. Nonetheless, the above random walk has an inherent disadvantage; in the worst case, its query response time is $\delta \cdot \mathrm{TTL}_n$, *i.e.*, linear in the number of peers in the system. This motivates us to also look at the expanding ring mechanism. Our second main result is that the expanding ring over an unstructured system

also scales very well as the number of peers increases.

Theorem 5.2. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and vertex balanced, that its state space is $\mathbb{S}_{n,d} \subseteq \mathbb{G}_{n,d}$, where $d \geq 3$, and that its stationary distribution is uniform over $\mathbb{S}_{n,d}$ and contiguous to the uniform distribution over $\mathbb{G}_{n,d}$. Then, there exists a $\mathrm{TTL}_n = \Theta\left(\log_{(d-1)} n\right)$ such that the expanding ring has*

$$\rho_n = O\left(n^{\frac{\log(d-1)}{\log(d-3)}-1}\right)$$

and

$$\rho_n^0 = O\left(n^{1-\frac{\log(d-3)}{\log(d-1)}}\right)$$

irrespectively of p_n.

Theorem 5.2 is proved in Section 5.5.4. Note that the bounds on both the average load-per-peer and the server load we obtain are not constant in n. Instead, both bounds grow as fractional powers of the population size. However, the exponents of these bounds are very small. For example, for $d = 32$, the value used by Gnutella super-peers (see Section 2.2), the above bounds are

$$\rho_n = O\left(n^{0.0199}\right) \quad \text{and} \quad \rho_n^0 = O\left(n^{0.0195}\right).$$

Most importantly, as the exponents decrease with d, a system designer can make them arbitrarily small, by choosing the appropriate degree of the overlay graph. In short, the bounds or ρ_n and ρ_n^0 grow very slowly in n, to the extent that, for practical purposes, both traffic loads can be considered constant.

The above result is achieved for a $\mathrm{TTL}_n = \Theta\left(\log_{(d-1)} n\right)$. It is easy to see that the worst case response time of an expanding ring with the above TTL_n is $O\left(\log^2_{(d-1)} n\right)$ (see also Lemma 5.12). This implies that the expanding ring mechanism not only is scalable, but also exhibits a much improved query response time compared to the random walk.

The assumptions of Theorem 5.2 hold for the switch model of the overlay graph, presented in section 4.3.2, as well as the flip model, presented in Section 4.3.3. Moreover, the theorem holds if $\mathbb{S}_{n,d}$ is either $\mathbb{G}_{n,d}$, $\mathbb{H}_{n,d}$, $\mathbb{I}_{n,d}$, or $\mathbb{CG}_{n,d}$, as defined in Section 3.3.2. In general, the theorem holds for any (ergodic, vertex balanced) Markov process $\{G(t)\}_{t\in\mathbb{N}}$ that is unstructured, *i.e.*, its stationary distribution is uniform over a large enough (in the sense of contiguity) subset of $\mathbb{G}_{n,d}$.

Again, our analysis (see Theorems 5.6, 5.7, and 5.9) covers more cases than the ones discussed above. In particular, it can be extended to obtain bounds on the traffic incurred by the

expanding ring over general overlay graph topologies, even ones that are not expanders *w.h.p*. Such an extension can again be found in Section 5.7.

5.3 A Markov Process Representation of the Hybrid System

An interesting property of the hybrid system is that its evolution is characterized by a unique Markov process, that does not depend on the query propagation mechanism used by peers. Roughly speaking, both the overlay graph and of the set of positive peers in the system evolve in a manner that does not depend on the outcome of a search; this is a simple consequence of the fact that the requested item can always be retrieved by redirecting a query to the server.

More precisely, the evolution of the hybrid system can be described by a Markov process $\{A(t), G(t)\}_{t\in\mathbb{R}_+}$, where $A(t) \subset V_n$ is the set of positive peers (*i.e.*, that have a copy of the item) at time t and $G(t) \in \mathbb{S}_{n,d}$ is the overlay graph of the system at time t. Then, the Markov process $\{A(t), G(t)\}_{t\in\mathbb{R}_+}$ does not in any way depend on the query propagation mechanism used by peers. For this reason, before we present our analysis of the random walk and expanding ring query propagation mechanisms (appearing in Sections 5.4 and 5.5, respectively) we first give an in-depth description of the Markov process $\{A(t), G(t)\}_{t\in\mathbb{R}_+}$.

5.3.1 The Set of Positive Peers and the Overlay Graph

The Marginal Processes $\{G(t)\}_{t\in\mathbb{R}_+}$ and $\{A(t)\}_{t\in\mathbb{R}_+}$

The process $\{G(t)\}_{t\in\mathbb{R}_+}$ is a Markovian graph process in $\mathbb{S}_{n,d} \subseteq \mathbb{MG}_{n,d}$, driven by the churn sequence $\{I(t)\}_{t\in\mathbb{N},t>0}$, as described in Section 4.1.2. For the sake of completeness, we restate the basic assumptions behind the definition of such a process. First, the process is uniformized, and the departure rate from each state is $n\mu$. The transition probabilities of its embedded Markov chain $\{G(t)\}_{t\in\mathbb{N}}$ are as follows: Conditioned on

$$G(t) = G, \qquad G \in \mathbb{S}_{n,d},$$

i.e., the graph at the t-th simultaneous departure/arrival epoch is G, and on

$$I(t+1) = i, \qquad i \in V,$$

i.e., the event that the peer replaced at the $t+1$-th departure/arrival epoch is the i-th peer, the probability that the overlay graph at the $t+1$-th departure/arrival epoch is $G' \in \mathbb{S}_{n,d}$ is given

by

$$p^i_{GG'},$$

where, in general, $p^i_{GG'}$ depends on the connection protocol employed by peers. Given that $\{I(t)\}_{t\in\mathbb{N},t>0}$ are i.i.d. and uniform over V_n, the transition probabilities of $\{G(t)\}_{t\in\mathbb{N}_+}$ are

$$p_{GG'} = \sum_{i\in V_n} \frac{1}{n} p^i_{GG'}.$$

The set of positive peers $\{A(t)\}_{t\in\mathbb{R}_+}$ is a Markov process in 2^{V_n}, also driven by the churn sequence $\{I(t)\}_{t\in\mathbb{N},t>0}$. It evolves as follows: conditioned on $I(t+1) = i$, at the $t+1$-th simultaneous departure/arrival event, peer i is replaced by a new peer that requests the item with probability p_n, independently of previous requests and of the evolution of $\{G(t)\}_{t\in\mathbb{R}_+}$. Note that a peer that requests the item always retrieves it (by potentially requesting it from the server) and, therefore, always becomes positive. As a result, the evolution of $A(t)$ does not depend on the query propagation mechanism used or on whether a query propagation succeeds in locating a positive peer.

The process $\{A(t)\}_{t\in\mathbb{R}_+}$ is also uniformized, as the departure rate from every state is $n\mu$. Its embedded Markov chain $\{A(t)\}_{t\in\mathbb{N}}$ captures the set of positive peers right after the t-th departure/arrival epoch. Its transition probabilities are as follows. Conditioned on

$$A(t) = A, \qquad A \subseteq V_n,$$

i.e., the set of peers having the item at the t-th departure/arrival epoch, and on

$$I(t+1) = i, \qquad i \in V,$$

i.e., the event that the peer replaced at the $t+1$-th departure/arrival epoch is the i-th peer, the probability that the set at the $t+1$-th departure/arrival epoch is $A' \subset V_n$ is given by

$$p^i_{AA'} = \begin{cases} p_n, & \text{if } A' = A \cup \{i\} \\ (1-p_n), & \text{if } A' = A \setminus \{i\} \\ 0, & \text{o.w.} \end{cases} \tag{5.1}$$

Note that the chain includes self-transitions from A to A: if $i \in A$ then $A \cup \{i\} = A$ and if $i \notin A$ then $A \setminus \{i\} = A$.

Given that $\{I(t)\}_{t\in\mathbb{N},t>0}$ are i.i.d. and uniform over V_n, the transition probabilities of $\{A(t)\}_{t\in\mathbb{N}}$ are given by

$$p_{AA'} = \sum_{i\in V_n} \frac{1}{n} p^i_{AA'}.$$

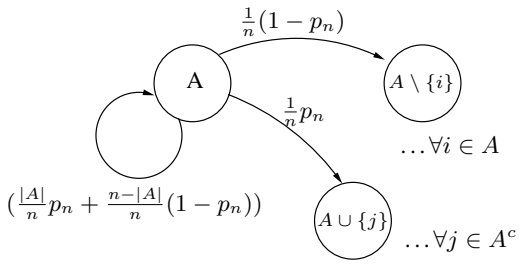


Figure 5.1: The Markov chain $\{A(t)\}_{t\in\mathbb{N}}$, *i.e.*, the set of positive peers at the t-th departure/arrival epoch. A transition from a set A to $A\cup\{j\}$, where $j \in A^c$, indicates that the null peer j is replaced by a peer that requested the item. Such transitions occur with probability $\frac{1}{n}p_n$. This is so because a new peer requests the item with probability p_n and every request always leads to acquiring the item, irrespectively of the topology and the query propagation mechanism used. On the other hand, a transition to a set $A \setminus \{i\}$, where $i \in A$, indicates that the positive peer i is replaced by a peer that does not request the item. Such transitions occur with rate $\frac{1}{n}(1-p_n)$. Finally, all other events leave the set A unaltered.

These transition probabilities are illustrated in Figure 5.1.

The cardinality $\{|A(t)|\}_{t\in\mathbb{N}_+}$ of $\{A(t)\}_{t\in\mathbb{N}_+}$ (*i.e.*, the number of positive peers) is also a Markov chain, described in Figure 5.2. This is because the transitions of this process are also Markovian: given that, at the t-th departure/arrival epoch, k peers are positive, the number of positive peers at the $t+1$-th epoch does not depend on the number of positive peers at the epochs $0, 1, \ldots, t-1$.

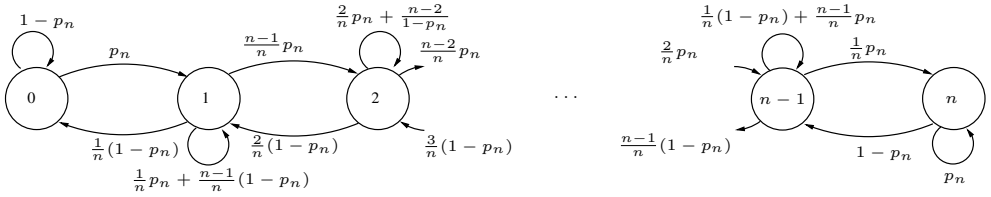


Figure 5.2: The Markov chain $\{|A(t)|\}_{t\in\mathbb{N}}$, *i.e.*, the number of positive peers. If k peers have the item, where $k = 0, 1, \ldots, n$, transitions to $k+1$ happen with probability $\frac{n-k}{n}p_n$, as this is the probability with which one of the null peers is replaced by a peer that requests the item. Transitions to $k-1$ happen with rate $\frac{k}{n}p_n$, as this is the probability with which one of the positive peers is replaced by a null peer.

Recall that our standing assumption is that $0 < p_n < 1$. Under this assumption, it can easily be checked that both $\{A(t)\}_{t\in\mathbb{N}}$ and $\{|A(t)|\}_{t\in\mathbb{N}}$ are ergodic and reversible. The stationary distribution of $\{A(t)\}_{t\in\mathbb{N}}$ (which, by uniformity, is also the stationary distribution of $\{A(t)\}_{t\in\mathbb{R}_+}$)

is

$$\nu_A = p_n^{|A|}(1-p_n)^{n-|A|}, \quad A \subseteq V_n. \tag{5.2}$$

Eq. (5.2) can be derived without the balance equations, merely by observing that the probability that a given peer is positive is equal to the probability it requests the item (namely, p_n) and that requests are independent. Similarly, the stationary distribution of $\{|A(t)|\}_{t\in\mathbb{N}}$ (and $\{|A(t)|\}_{t\in\mathbb{R}_+}$) is binomial:

$$\nu_k = \binom{n}{k} p_n^k (1-p_n)^{n-k}, \quad 0 \le k \le n. \tag{5.3}$$

The Joint Process $\{A(t), G(t)\}_{t\in\mathbb{R}_+}$

The processes $\{G(t)\}_{t\in\mathbb{R}_+}$ and $\{A(t)\}_{t\in\mathbb{R}_+}$ are *not* independent. The same is true for their embedded Markov chains $\{A(t)\}_{t\in\mathbb{N}}$ and $\{G(t)\}_{t\in\mathbb{N}}$. The reason is that both are driven by the same churn sequence $\{I(t)\}_{t\in\mathbb{N},t>0}$; the transitions of each chain happen jointly, depending on which peer is replaced.

In particular, the joint process $\{A(t), G(t)\}_{t\in\mathbb{R}_+}$ is a Markov process over $2^{V_n} \times \mathbb{S}_{n,d}$ that is also driven by $\{I(t)\}_{t\in\mathbb{N},t>0}$. It is also a uniformized Markov process, with aggregate transition rate $n\mu$. The transitions of its embedded Markov chain $\{A(t), G(t)\}_{t\in\mathbb{N}}$ can be described as follows. Conditioned on

$$A(t) = A, \qquad A \subseteq V_n,$$

i.e., the set of positive peers at the t-th simultaneous departure/arrival epoch, on

$$G(t) = G, \qquad G \in \mathbb{S}_{n,d},$$

i.e., the graph at the t-th simultaneous departure/arrival epoch is G, and on

$$I(t+1) = i, \qquad i \in V,$$

i.e., the event that $t+1$-th departure/arrival event is the i-th peer, the probability that at the $t+1$-th departure/arrival epoch the set is $A' \subset V_n$ and that the overlay graph is $G' \in \mathbb{S}_{m,d}$ is

$$p^i_{AA'} p^i_{GG'},$$

which captures the fact that, conditioned on $I(t+1)$, the transitions of $A(t)$ and $G(t)$ are independent. Given that $\{I(t)\}_{t\in\mathbb{N},t>0}$ are i.i.d. and uniform over V_n, the transition probabilities of $\{A(t), G(t)\}_{t\in\mathbb{N}}$ are

$$p_{(A,G)\to(A',G')} = \sum_{i\in V_n} \frac{1}{n} p^i_{AA'} p^i_{GG'}.$$

Although the chains $\{A(t)\}_{t\in\mathbb{N}}$ and $\{G(t)\}_{t\in\mathbb{N}}$ evolve jointly and are not independent, in "steady state" they do become independent! More precisely, the stationary distribution of the joint chain $\{A(t), G(t)\}_{t\in\mathbb{N}}$ has a product form, as stated in Theorem 5.3, which we present below. Before we state this theorem, we first establish that the ergodicity of $\{G(t)\}_{t\in\mathbb{N}}$ implies the ergodicity of the joint chain $\{A(t), G(t)\}_{t\in\mathbb{N}}$.

5.3.2 Irreducibility, Aperiodicity and Ergodicity.

To begin with, the irreducibility of $\{G(t)\}_{t\in\mathbb{N}}$ implies the irreducibility of $\{A(t), G(t)\}_{t\in\mathbb{N}}$.

Lemma 5.1. *The joint chain $\{A(t), G(t)\}_{t\in\mathbb{N}}$ is irreducible if and only if the marginal Markov chain $\{G(t)\}_{t\in\mathbb{N}}$ is irreducible.*

Proof. Suppose first that $\{G(t)\}_{t\in\mathbb{N}}$ is irreducible, and consider two states (A, G) and (A', G') in $2^{V_n} \times \mathbb{S}_n$. Then, since $\{A(t)\}_{t\in\mathbb{N}}$ is irreducible, there is a finite sequence of departure/arrival events, and corresponding requests of the item, that transforms A to A'. For example, these can be $|A \setminus A'|$ events, in which every peer in A but not in A' is replaced by a peer that does not request the item, followed by an additional $|A' \setminus A|$ events, in which every peer in A' but not in A is replaced by a peer requesting the item. Since $0 < p_n < 1$, each of these events occur with a non-zero probability: the former occur with probability $\frac{1}{n}(1 - p_n)$ and the latter occur with probability $\frac{1}{n}p_n$. Suppose that the graph at the end of this sequence of events is G''. Then, as $\{G(t)\}_{t\in\mathbb{N}}$ is irreducible, there is a finite sequence of events with non-zero probability in which G'' is transformed to to G' without altering $A(t)$: essentially, if a peer in A' is replaced, it is replaced by a peer that request the item and, similarly if a peer in $V_n \setminus A'$ is replaced, it is replaced by a peer that does not request the item. Hence, state (A', G') is accessible from state (A, G). As A, A', G, G' are arbitrary, the joint chain $\{A(t), G(t)\}_{t\in\mathbb{R}_+}$ is irreducible.

On the other hand, the existence of a non-zero probability path from (A, G) to (A', G') in $\{A(t), G(t)\}_{t\in\mathbb{N}}$ implies the existence of a non-zero probability path from G to G' of the same length in $\{G(t)\}_{t\in\mathbb{N}}$. In particular, suppose that a path

$$(A_1, G_1), (A_2, G_2), \ldots, (A_\ell, G_\ell)$$

exists, such that

$$(A_1, G_1) = (A, G) \qquad \text{and} \qquad (A_\ell, G_\ell) = (A', G')$$

and

$$p_{(A_kG_k)\to(A_kG_{k+1})} = \sum_{i\in V_n} \frac{1}{n} p^i_{A_kA_{k+1}} p^i_{G_kG_{k+1}} > 0.$$

for all $k = 1, \ldots, \ell - 1$. This implies that

$$\sum_{i \in V_n} p^i_{G_k G_{k+1}} > 0,$$

for all $k = 1, \ldots, \ell - 1$. Hence the path

$$G_1, G_2, \ldots, G_\ell$$

is a non-zero probability path of $\{G(t)\}_{t \in \mathbb{N}}$. Hence, if (A', G') is accessible from from (A, G) in $\{A(t), G(t)\}_{t \in \mathbb{N}}$, then G' is accessible from G in $\{G(t)\}_{t \in \mathbb{N}}$. As this holds for every (A, G) and every (A', G'), if $\{A(t), G(t)\}_{t \in \mathbb{N}}$ is irreducible, then so will be $\{G(t)\}_{t \in \mathbb{N}}$. □

The same statement holds about aperiodicity.

Lemma 5.2. *The joint chain $\{A(t), G(t)\}_{t \in \mathbb{N}}$ is aperiodic if and only if the marginal Markov chain $\{G(t)\}_{t \in \mathbb{N}}$ is aperiodic.*

Proof. Suppose first that $\{G(t)\}_{t \in \mathbb{N}}$ is aperiodic. We will prove that the joint chain $\{A(t), G(t)\}_{t \in \mathbb{N}}$ is also aperiodic by contradiction. Suppose that a state (A, G) of $\{A(t), G(t)\}_{t \in \mathbb{N}}$ has period $m > 1$. Consider a non-zero probability path $\mathcal{P}$ on $\{G(t)\}_{t \in \mathbb{N}}$ from G back to G and assume that it has length $\ell > 0$. That is, the path is of the form

$$G_1, G_2, \ldots, G_\ell$$

where

$$G_1 = G_\ell = G$$

and

$$p_{G_k G_{k+1}} = \sum_{i \in V_n} \frac{1}{n} p^i_{G_k G_{k+1}} > 0$$

for all $k = 1, \ldots, \ell - 1$. This implies that,

$$\text{for all } k = 1, \ldots, \ell - 1, \text{ there exists an } i \text{ such that } p^i_{G_k G_{k+1}} > 0 \tag{5.4}$$

Then, there is a non-zero probability path $\mathcal{P}'$ in $\{A(t), G(t)\}_{t \in \mathbb{N}}$ from (A, G) back to (A, G) that also has length ℓ: This path is simply the path in which transitions occur as in $\mathcal{P}$ while the set A remains unchanged. *I.e.*, $\mathcal{P}'$ is

$$(A, G_1), (A, G_2), \ldots, (A, G_\ell)$$

where

$$p_{AG_i,AG_{i+1}} = \sum_{i\in V_n} \frac{1}{n} p^i_{AA} p^i_{G_iG_{i+1}} = \sum_{i\in A} \frac{1}{n} p_n \cdot p^i_{G_iG_{i+1}} + \sum_{i\notin A} \frac{1}{n}(1-p_n)\cdot p^i_{G_iG_{i+1}}.$$

As $0 < p_n < 1$, (5.4) implies that the above are positive for all $k = 1, \ldots, \ell - 1$.

As (A, G) has period $m > 1$, ℓ must be divisible by m. As the path $\mathcal{P}$ is arbitrary, it follows that the length of every path on $\{G(t)\}_{t\in\mathbb{N}}$ from G back to G is divisible by $m > 1$. Hence, $\{G(t)\}_{t\in\mathbb{N}}$ is not aperiodic, a contradiction.

To show the "only if" direction, assume that $\{A(t), G(t)\}_{t\in\mathbb{N}}$ is aperiodic. In the proof of Lemma 5.2, we showed that the existence of a non-zero probability path from (A, G) to (A', G') in $\{A(t), G(t)\}_{t\in\mathbb{N}}$ implies the existence of a non-zero probability path from G to G' of the same length in $\{G(t)\}_{t\in\mathbb{N}}$. Applying this to paths from (A, G) back to (A, G), we have that, for every such path in $\{A(t), G(t)\}_{t\in\mathbb{N}}$ of length ℓ, there exists a path from G back to G in $\{G(t)\}_{t\in\mathbb{N}}$ also of length ℓ. Hence, if G has a period $m > 1$, so will (A, G). Therefore, if $\{G(t)\}_{t\in\mathbb{N}}$ is not aperiodic, neither is $\{A(t), G(t)\}_{t\in\mathbb{N}}$, a contradiction. □

The equivalence of the ergodicity of the marginal and the joint chain follows as an immediate corollary of the above two theorems.

Corollary 5.1. *The joint Markov chain $\{A(t), G(t)\}_{t\in\mathbb{N}}$ is ergodic if and only if the marginal Markov chain $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic.*

5.3.3 Stationary Distribution and Reversibility

Having established that the irreducibility of $\{G(t)\}_{t\in\mathbb{N}}$ is necessary and sufficient for the irreducibility of $\{A(t), G(t)\}_{t\in\mathbb{N}}$, we turn our attention to the stationary distribution of $\{A(t), G(t)\}_{t\in\mathbb{N}}$. As mentioned above, it turns out that this distribution has a simple product form; in other words, in "steady state", the set of positive peers is independent of the overlay graph.

Theorem 5.3. *Assume that the marginal Markov chain $\{G(t)\}_{t\in\mathbb{N}}$ is irreducible. Then, the joint chain $\{A(t), G(t)\}_{t\in\mathbb{R}_+}$ has a unique stationary distribution given by*

$$\nu_{A,G} = \nu_A \cdot \nu_G, \qquad A \subseteq V_n, G \in \mathbb{S}_{n,d}$$

where ν_G the stationary distribution of the marginal chain $\{G(t)\}_{t\in\mathbb{N}}$ and ν_A the stationary distribution of the marginal chain $\{A(t)\}_{t\in\mathbb{N}}$, given by (5.2)*.*

Proof. As $\{G(t)\}_{t\in\mathbb{N}}$ is irreducible, Theorem 3.3 implies that a unique stationary distribution ν_G exists such that

$$\nu_{G'} = \sum_{G\in\mathbb{S}_{n,d}} \nu_G \sum_{i\in V_n} \frac{1}{n} p^i_{GG'}, \qquad \text{for all } G' \in \mathbb{S}_{n,d}.$$

We show below that $\nu_A \cdot \nu_G$ satisfies the balance equations of $\{A(t), G(t)\}_{t\in\mathbb{N}}$.

Given $A' \subseteq V_n$, $G' \in \mathbb{S}_{n,d}$, we have that

$$\begin{aligned}
&\sum_{G\in\mathbb{S}_{n,d}} \sum_{A\subseteq V_n} \nu_G \nu_A p_{(A,G)\to(A',G')} \\
&\quad= \sum_{G\in\mathbb{S}_{n,d}} \sum_{A\subseteq V_n} \nu_G p_n^{|A|}(1-p_n)^{n-|A|} \sum_{i\in V_n} \frac{1}{n} p^i_{AA'} p^i_{GG'} \\
&\quad= \frac{1}{n} \sum_{G\in\mathbb{S}_{n,d}} \nu_G \Bigg[\sum_{A\neq A', A=A'\setminus\{i\}} p_n^{|A|}(1-p_n)^{n-|A|}\cdot p_n \cdot p^i_{GG'} + \\
&\qquad\qquad \sum_{A\neq A', A=A'\cup\{j\}} p_n^{|A|}(1-p_n)^{n-|A|}\cdot (1-p_n)\cdot p^j_{GG'} + \\
&\qquad\qquad \sum_{A=A', i\in A'} p_n^{|A|}(1-p_n)^{n-|A|}\cdot p_n p^i_{GG'} + \\
&\qquad\qquad \sum_{A=A', j\notin A'} p_n^{|A|}(1-p_n)^{n-|A|}\cdot (1-p_n) p^j_{GG'} \Bigg] \\
&\quad= \frac{1}{n} \sum_{G\in\mathbb{S}_{n,d}} \nu_G \Bigg[\sum_{i\in A'} p_n^{|A'|-1}(1-p_n)^{n-|A'|+1}\cdot p_n \cdot p^i_{GG'} + \\
&\qquad\qquad \sum_{j\notin A'} p_n^{|A'|+1}(1-p_n)^{n-|A'|-1}\cdot (1-p_n)\cdot p^j_{GG'} + \\
&\qquad\qquad \sum_{i\in A'} p_n^{|A'|}(1-p_n)^{n-|A'|}\cdot p_n p^i_{GG'} + \\
&\qquad\qquad \sum_{j\notin A'} p_n^{|A'|}(1-p_n)^{n-|A'|}\cdot (1-p_n) p^j_{GG'} \Bigg] \\
&\quad= \frac{1}{n} \sum_{G\in\mathbb{S}_{n,d}} \nu_G \Bigg[\sum_{i\in A'} p_n^{|A'|-1}(1-p_n)^{n-|A'|} p^i_{GG'} + \sum_{j\notin A'} p_n^{|A'|}(1-p_n)^{n-|A'|} p^j_{GG'} \Bigg] \\
&\quad= \frac{1}{n} \sum_{G\in\mathbb{S}_{n,d}} \nu_G \sum_{i\in V_n} p_n^{|A|}(1-p_n)^{n-|A|} p^j_{GG'} \\
&\quad= p_n^{|A'|}(1-p_n)^{n-|A'|} \sum_{G\in\mathbb{S}_{n,d}} \nu_G \sum_{i\in V_n} \frac{1}{n} p^j_{GG'} \\
&\quad= p_n^{|A'|}(1-p_n)^{n-|A'|} \nu_{G'} = \nu_{A'}\nu_{G'}.
\end{aligned}$$

Hence the product $\nu_A \cdot \nu_G$ is a stationary distribution of $\{A(t), G(t)\}_{t\in\mathbb{R}_+}$. Lemma 5.1 implies that $\{A(t), G(t)\}_{t\in\mathbb{N}}$ is irreducible and, hence, the uniqueness of the stationary distribution is implied by the Perron-Frobenius Theorem (Theorem 3.3). □

Interestingly, contrary to ergodicity, the reversibility of $\{G(t)\}_{t\in\mathbb{N}}$ is not sufficient to prove the reversibility of $\{A(t), G(t)\}_{t\in\mathbb{N}}$. Instead, the stronger notion of vertex reversibility, which we introduced in Section 4.2.3, is required.

Lemma 5.3. *Assume that the marginal Markov chain $\{G(t)\}_{t\in\mathbb{N}}$ is irreducible. Then, the joint Markov chain $\{A(t), G(t)\}_{t\in\mathbb{N}}$ is reversible if and only if the marginal Markov chain $\{G(t)\}_{t\in\mathbb{N}}$ is vertex reversible.*

Proof. Since $\{G(t)\}_{t\in\mathbb{N}}$ is irreducible, by Theorem 5.3 so is $\{A(t), G(t)\}_{t\in\mathbb{N}}$, and its stationary distribution is given by $\nu_A\nu_G$. Recall from Section 4.2.3 that the vertex reversibility of $\{G(t)\}_{t\in\mathbb{N}}$ implies that its stationary distribution satisfies (4.3), *i.e.*,

$$\nu_G p^i_{GG'} = \nu_{G'} p^i_{G'G}, \qquad \text{for every } i \in V_n,\ G, G' \in \mathbb{S}_{n,d}.$$

The necessary and sufficient condition for the reversibility of $\{A(t), G(t)\}_{t\in\mathbb{N}}$ is that

$$\nu_A\nu_G p_{(A,G)\to(A',G')} = \nu_{A'}\nu_{G'} p_{(A',G')\to(A,G)}, \qquad A, A' \subseteq V_n,\ G, G' \in \mathbb{S}_{n,d}. \tag{5.5}$$

This is obviously always true if $A = A'$ and $G = G'$. Suppose first that $A \neq A'$. Then, given that $0 < p_n < 1$, by (5.1), $p^i_{AA'} > 0$ if and only if $A' = A \cup \{i\}$ or $A' = A \setminus \{i\}$. Therefore, (5.5) holds if and only if it holds for the two cases where A and A' differ by one element. If $A' = A \cup \{i\}$, we have

$$\nu_A\nu_G p_{(A,G)\to(A',G')} \overset{(5.2),(5.1)}{=} p_n^{|A|}(1-p_n)^{n-|A|}\nu_G \frac{1}{n} p_n p^i_{GG'}$$

while

$$\nu_{A'}\nu_{G'} p_{(A',G')\to(A,G)} \overset{(5.2),(5.1)}{=} p_n^{|A|+1}(1-p_n)^{n-|A|-1}\nu_{G'} \frac{1}{n}(1-p_n) p^i_{G'G}$$

Hence, (5.5) holds for $A' = A \cup \{i\}$, $A' \neq A$, if and only if

$$\nu_G p^i_{GG'} = \nu_{G'} p^i_{G'G}$$

for all $G, G' \in \mathbb{S}_{nd}$ and for all $i \notin A$.

If $A' = A \setminus \{i\}$ then, similarly,

$$\nu_A\nu_G p_{(A,G)\to(A',G')} \overset{(5.2),(5.1)}{=} p_n^{|A|}(1-p_n)^{n-|A|}\nu_G \frac{1}{n}(1-p_n) p^i_{GG'}$$

while

$$\nu_{A'}\nu_{G'}p_{(A',G')\to(A,G)} \overset{(5.2),(5.1)}{=} p_n^{|A|-1}(1-p_n)^{n-|A|+1}\nu_{G'}\sum_{i\in V_n}\frac{1}{n}p_n p^i_{G'G}$$

Hence, (5.5) holds for $A' = A \setminus \{i\}$, $A' \neq A$, if and only if

$$\nu_G p^i_{GG'} = \nu_{G'} p^i_{G'G}$$

for all $G, G' \in \mathbb{S}_{nd}$ and for all $i \in A$.

Combining the two above statements, we get that (5.5) holds for all $A \neq A'$ and for all G, G' if an only if $\{G(t)\}_{t\in\mathbb{N}}$ is vertex reversible.

Finally, suppose that $A = A'$ but $G \neq G'$. Then, (5.5) holds if and only if

$$\nu_A\nu_G\sum_{i\in V_n}\frac{1}{n}p^i_{AA}p^i_{GG'} = \nu_A\nu_{G'}\sum_{i\in V_n}\frac{1}{n}p^i_{AA}p^i_{G'G},$$

which is immediately implied for all $A \subseteq V_n$ and for all $G, G' \in \mathbb{S}_{n,d}$ by the vertex reversibility of $\{G(t)\}_{t\in\mathbb{N}}$. Hence, (5.5) holds for all $A = A'$ and for all G, G' if $\{G(t)\}_{t\in\mathbb{N}}$ is vertex reversible.

In conclusion, (5.5) holds for all $A, A' \subseteq V_n$ and for all $G, G' \in \mathbb{MG}_{n,d}$ if and only if $\{G(t)\}_{t\in\mathbb{N}}$ is vertex reversible. □

5.3.4 Rewards Over the Joint Chain

There are several quantities of interest in our analysis that can be represented as functions on the state space of the chain $\{A(t), G(t)\}_{t\in\mathbb{N}}$. In renewal theory, such functions are typically referred to as *reward* functions [Gal96]. One example is the number of query messages propagated over the peer-to-peer network at the t-th departure/arrival epoch. Another is the indicator function of the event that a query reaches the server at the time of a departure/arrival epoch. Both of these rewards can be represented as instances of the class of reward functions we discuss below.

Let $\{R(t)\}_{t\in\mathbb{N}}$ be a random reward function over $\{A(t), G(t)\}_{t\in\mathbb{N}}$ taking values in $\mathbb{R}$, having the following form:

$$R(t) = \begin{cases} 0, & \text{if no request is issued at the } t\text{-th epoch,} \\ r_{i,A,G}, & \text{if a request is issued at the } t\text{-th epoch,} \end{cases}$$

where, for all t at which requests are issued, $r_{i,A,G}$ are independent random variables whose distribution depends on

$i \in V_n$: the peer that issues the request,

$A \subset V_n$: the set of positive peers at the time the request is issued and

$G \in \mathbb{S}_{n,d}$: the overlay graph at the time the request is issued.

The following lemma allows us to characterize the limit of the *ensemble average* of the reward $R(t)$

$$\lim_{t\to\infty} \mathbb{E}[R(t)]$$

as well as the limit of the *time average* of the reward $R(t)$

$$\lim_{t\to\infty} \frac{1}{t} \sum_{s=0}^{t} R(s)$$

provided that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and exhibits the vertex balance property, as defined in Section 4.2.3.

Lemma 5.4. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic. Then, there exists an $R_n \in \mathbb{R}$ such that*

$$R_n = \lim_{t\to\infty} \mathbb{E}[R(t)] = \lim_{t\to\infty} \frac{1}{t} \sum_{s=0}^{t} R(s), \quad \text{a.s.}$$

Moreover, if $\{G(t)\}_{t\in\mathbb{N}}$ is vertex balanced:

$$R_n = p_n \cdot \sum_{G\in\mathbb{S}_{n,d}} \nu_G \sum_{A\subsetneq V} \frac{n-|A|}{n} p_n^{|A|}(1-p_n)^{n-1-|A|} \mathbb{E}_{u_{A^c}}[r_{A,G}], \tag{5.6}$$

where $\mathbb{E}_{u_{A^c}}[r_{A,G}] = \frac{1}{|A^c|}\sum_{i\in A^c} \mathbb{E}[r_{i,A,G}]$ is the expected value of $r_{i,A,G}$ conditioned on i being distributed uniformly outside the set A.

We will call R_n the *steady state* reward of the chain. The intuition behind Eq. (5.6) is that, if $\{G(t)\}_{t\in\mathbb{N}}$ is vertex balanced, the peer requesting the item is positioned uniformly outside the set of positive peers.

Proof of Lemma 5.4. Since $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic, by Corollary 5.1 so will be $\{A(t), G(t)\}_{t\in\mathbb{N}}$. By the key renewal theorem [Gal96], the limits

$$\lim_{t\to\infty} \mathbb{E}[R(t)] \quad \text{and} \quad \lim_{t\to\infty} \frac{1}{t} \sum_{s=0}^{t} R(s)$$

exist and are equal to

$$R_n = \sum_{A \subseteq V_n} \sum_{G \in \mathbb{S}_{n,d}} \mathbb{E}[R \mid A, G] \nu_{A,G} \tag{5.7}$$

where, for all t,

$$\mathbb{E}[R \mid A, G] \equiv \mathbb{E}[R(t) \mid A(t) = A, G(t) = G], \qquad A \subseteq V_n, G \in \mathbb{S}_{n,d}, \tag{5.8}$$

is the expected reward at a state (A, G) (note that it does not depend on t as $r_{i,A,G}$ does not), and

$$\nu_{A,G}, \qquad A \subseteq V_n, G \in \mathbb{S}_{n,d},$$

the unique stationary probability distribution of $\{A(t), G(t)\}_{t \in \mathbb{N}}$ (which exists by ergodicity). The expected reward $\mathbb{E}[R \mid A, G]$ can be rewritten as

$$\mathbb{E}[R \mid A, G] = \sum_{A' \subseteq V_n} \sum_{i \in V_n} \mathbb{E}[R \mid A, G, A', i] \mathbf{P}(A(t-1) = A', I(t) = i \mid A(t) = A, G(t) = G) \tag{5.9}$$

where

$$\mathbb{E}[R \mid A, G, A', i] = \mathbb{E}[R(t) \mid A(t) = A, G(t) = G, A(t-1) = A', I(t) = i].$$

Moreover,

$$\mathbf{P}(A(t\text{–}1) = A', I(t) = i \mid A(t) = A, G(t) = G) = \sum_{G' \in \mathbb{S}_{n,d}} \mathbf{P}(A(t\text{–}1) = A', G(t\text{–}1) = G', I(t) = i \mid A(t) = A, G(t) = G)$$

and

$$\mathbf{P}(A(t\text{–}1) = A', G(t\text{–}1) = G', I(t) = i \mid A(t) = A, G(t) = G) = \mathbf{P}(A(t) = A, G(t) = G \mid A(t\text{–}1) = A', G(t\text{–}1) = G', I(t) = i) \cdot \frac{\mathbf{P}(A(t\text{–}1) = A', G(t\text{–}1) = G', I(t) = i)}{\mathbf{P}(A(t) = A, G(t) = G)}$$

By the definition of the churn-driven Markov chain $\{A(t), G(t)\}_{t \in \mathbb{N}}$,

$$\mathbf{P}(\mid A(t) = A, G(t) = G \mid A(t-1) = A', G(t-1) = G', I(t) = i) = p^i_{A'A} p^i_{G'G}$$

Moreover, in steady state,

$$\mathbf{P}(A(t) = A, G(t) = G) = \nu_{A,G}$$

and

$$\mathbf{P}(A(t-1)=A', G(t-1)=G', I(t)=i) = \nu_{A',G'}\frac{1}{n}$$

as $I(t)$ is sampled independently of $A(t-1)$ and $G(t-1)$. This gives us that

$$\mathbf{P}(A(t\text{–}1)=A', I(t)=i \mid A(t)=A, G(t)=G) = \frac{1}{\nu_{A,G}}\frac{1}{n}p^i_{A'A}\sum_{G'\in\mathbb{S}_{n,d}} p^i_{G'G}\nu_{A',G'}. \tag{5.10}$$

As $\{G(t)\}_{t\in\mathbb{N}}$ is irreducible, Theorem 5.3 implies that

$$\nu_{A,G} = \nu_A \cdot \nu_G, \qquad A \subseteq V_n, G \in \mathbb{S}_{n,d}$$

where ν_A,ν_G the stationary distributions of the marginal chains $\{A(t)\}_{t\in\mathbb{N}}$ and $\{G(t)\}_{t\in\mathbb{N}}$, respectively. Hence, (5.10) becomes

$$\mathbf{P}(A(t\text{–}1)=A', I(t)=i \mid A(t)=A, G(t)=G) = \frac{\nu_{A'}}{\nu_{A,G}}\frac{1}{n}p^i_{A'A}\sum_{G'\in\mathbb{S}_{n,d}} \nu_{G'}p^i_{G'G} = \frac{\nu_{A'}}{\nu_{A,G}}\frac{1}{n}p^i_{A'A}\nu_G \tag{5.11}$$

because

$$\sum_{G'\in\mathbb{S}_{n,d}} \nu_{G'}p^i_{G'G} = \nu_G$$

as $\{G(t)\}_{t\in\mathbb{N}}$ is vertex balanced. Using for the r.h.s. of (5.11) in (5.9) yields

$$\mathbb{E}[R \mid A, G] = \sum_{A'\subseteq V_n}\sum_{i\in V_n}\mathbb{E}[R \mid A, G, A', i]\frac{\nu_{A'}}{\nu_{A,G}}\frac{1}{n}p^i_{A'A}\nu_G$$

and, by (5.7),

$$R_n = \sum_{A'\subseteq V_n}\sum_{G\in\mathbb{S}_{n,d}}\nu_{A'}\nu_G\sum_{A\subseteq V_n}\sum_{i\in V_n}\mathbb{E}[R \mid A, G, A', i]\frac{1}{n}p^i_{A'A}$$

Recall from (5.1) that

$$p^i_{A'A} = \begin{cases} p_n, & \text{if } A = A'\cup\{i\} \\ (1-p_n), & \text{if } A = A'\setminus\{i\} \\ 0, & \text{o.w.} \end{cases}$$

This gives us

$$\begin{aligned} R_n = &\sum_{A'\subseteq V_n}\sum_{G\in\mathbb{S}_{n,d}}\nu_{A'}\nu_G\cdot \\ &\Big[\sum_{i\in A'}\frac{1}{n}p_n\mathbb{E}[R \mid A', G, A', i] + \sum_{i\in A'}\frac{1}{n}(1-p_n)\mathbb{E}[R \mid A'\setminus\{i\}, G, A', i] + \\ &\quad\sum_{j\notin A'}\frac{1}{n}p_n\mathbb{E}[R \mid A'\cup\{j\}, G, A', j] + \sum_{j\notin A'}\frac{1}{n}(1-p_n)\mathbb{E}[R \mid A', G, A', j]\Big] \end{aligned}$$

On the other hand, by the definition of the reward function $R(t)$

$$\mathbb{E}[R \mid A', G, A', i] = \mathbb{E}[r_{i,A'\setminus\{i\},G}]$$

while

$$\mathbb{E}[R \mid A' \setminus \{i\}, G, A', i] = 0,$$

as, by definition, R is zero when no request is issued. Similarly, for $j \notin A'$,

$$\mathbb{E}[R \mid A' \cup \{j\}, G, A', j] = \mathbb{E}[r_{i,A',G}]$$

while, again

$$\mathbb{E}[R \mid A', G, A', j] = 0,$$

as, by definition, R is zero when no request is issued. The above yield

$$\begin{aligned}
R_n &= \sum_{A'\subseteq V_n} \sum_{G\in\mathbb{S}_{n,d}} \nu_{A'}\nu_G \Big[\sum_{i\in A'} \frac{1}{n} p_n \mathbb{E}[r_{i,A'\setminus\{i\},G}] + \sum_{j\notin A'} \frac{1}{n} p_n \mathbb{E}[r_{j,A',G}] \\
&= p_n \sum_{G\in\mathbb{S}_{n,d}} \nu_G \frac{1}{n} \sum_{A\subsetneq V_n} \sum_{j\notin A} \mathbb{E}[r_{A,G,j}](\nu_{A\cup\{j\}} + \nu_A) \\
&\overset{(5.2)}{=} p_n \sum_{G\in\mathbb{S}_{n,d}} \nu_G \frac{1}{n} \sum_{A\subsetneq V_n} \sum_{j\notin A} \mathbb{E}[r_{j,A,G}] \left(p_n^{|A|+1}(1-p_n)^{n-|A|-1} + p_n^{|A|}(1-p_n)^{n-|A|} \right) \\
&= p_n \sum_{G\in\mathbb{S}_{n,d}} \nu_G \frac{1}{n} \sum_{A\subsetneq V_n} \sum_{j\notin A} \mathbb{E}[r_{j,A,G}] p_n^{|A|}(1-p_n)^{n-|A|-1} \\
&= p_n \sum_{G\in\mathbb{S}_{n,d}} \nu_G \sum_{A\subsetneq V_n} \frac{n-|A|}{n} p_n^{|A|}(1-p_n)^{n-|A|-1} \mathbb{E}_{u_{A^c}}[r_{A,G}]
\end{aligned}$$

□

Average Load per Peer

Recall from Section 5.1.2 that the average traffic load per peer is defined as

$$\rho_n = \sum_{i=1}^{n} \frac{\rho_i}{n} = \lim_{t\to\infty} \frac{1}{n} \sum_{i=1}^{n} \frac{M_i(t)}{t},$$

where $\{M_i(t)\}_{t\in\mathbb{R}_+}$ the number of query messages that peers at vertex i have received up to and including time t. As we noted in Section 5.1.2, the limit above may not necessarily exist unless our system is ergodic. Below, we show that under the ergodicity conditions we have

established above, the limit indeed exists and can be described in terms of a reward function that satisfies Lemma 5.4.

Define $\{C(t)\}_{t\in\mathbb{N}}$ as the following reward function

$$C(t) = \begin{cases} 0, & \text{if no request is issued at the } t\text{-th epoch,} \\ C_{i,A,G}, & \text{if a request is issued at the } t\text{-th epoch,} \end{cases} \tag{5.12}$$

where $C_{i,A,G}$ is the number of messages generated by a query propagation given that, at the time the query is issued, the source peer is i, the set of positive peers is A and the overlay graph is G. Then, the following lemma holds:

Lemma 5.5. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic. Then, the average traffic load per peer ρ_n exists* a.s. *and is equal to*

$$\rho_n = \mu \cdot C_n$$

where

$$C_n = \lim_{t\to\infty} \mathbb{E}[C(t)] = \lim_{t\to\infty} \frac{1}{t} \sum_{s=0}^{t} C(s), \quad \text{a.s.}$$

and $\{C(t)\}$ the reward defined in (5.12)*.*

Proof. If $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic, then, by Lemma 5.4, C_n exists. Let $\{N(t)\}_{t\in\mathbb{R}_+}$ be the number of departure/arrival epochs up to and including time t. Observe then that, for any $t \in \mathbb{R}_+$,

$$\sum_{i=1}^{n} M_i(t) = \sum_{s=0}^{N(t)} C(s).$$

Therefore,

$$\begin{aligned} \rho_n &= \frac{1}{n} \cdot \lim_{t\to\infty} \sum_{i=1}^{n} \frac{M_i(t)}{t} = \frac{1}{n} \cdot \lim_{t\to\infty} \frac{\sum_{s=0}^{N(t)} C(s)}{t} \\ &= \frac{1}{n} \cdot \lim_{t\to\infty} \frac{N(t)}{t} \cdot \lim_{t\to\infty} \frac{\sum_{s=1}^{N(t)} C(s)}{N(t)} = \frac{1}{n} \cdot n\mu \cdot C_n, \quad \textit{a.s.} \end{aligned}$$

The first limit exists by the strong law of large numbers because $\{N(t)\}_{t\in\mathbb{R}_+}$ is a Poisson process with rate $n\mu$. The second limit exists because C_n exists and

$$\lim_{t\to\infty} N(t) = \infty, \quad \textit{a.s.},$$

again, by the strong law of large numbers. Hence, ρ_n exists *a.s.*, and the lemma follows. □

In conclusion, Lemma 5.5 proves the existence of the average traffic load per peer ρ_n, as defined in Section 5.1.2. It also reduces the computation of ρ_n to computing the steady state reward C_n. The latter can be computed for all query propagation mechanisms we discuss through Lemma 5.4.

Server Load

Recall from Section 5.1.2 that the traffic load on the server is defined as

$$\rho_n^0 = \lim_{t\to\infty} \frac{M_0(t)}{t},$$

where $\{M_0(t)\}_{t\in\mathbb{R}_+}$ the number of query messages the server has received up to and including time t. Again, as noted in Section 5.1.2, the limit above may not necessarily exist unless the system is ergodic. Below we show that, just as the average load per peer, under the ergodicity conditions we have established above, this limit too exists (*a.s.*) and can be described in terms of a reward function that satisfies Lemma 5.4.

Define $\{R(t)\}_{t\in\mathbb{N}}$ as the following reward function

$$R(t) = \begin{cases} 0, & \text{if no request is issued at the } t\text{-th epoch,} \\ r_{i,A,G}, & \text{if a request is issued at the } t\text{-th epoch,} \end{cases} \tag{5.13}$$

where $r_{i,A,G}$ is one if the query propagation reaches the server and zero otherwise, given that, at the time the query is issued, the source peer is i, the set of positive peers is A and the overlay graph is G. Then, the following lemma holds:

Lemma 5.6. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic. Then, the server traffic load per ρ_n^0 exists* a.s. *and is equal to*

$$\rho_n^0 = n\mu \cdot R_n$$

where

$$R_n = \lim_{t\to\infty} \mathbb{E}[R(t)] = \lim_{t\to\infty} \frac{1}{t} \sum_{s=0}^{t} R(s), \quad \text{a.s.}$$

and $\{R(t)\}$ the reward defined in (5.13).

Proof. If $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic, then, by Lemma 5.4, C_n exists. Let $\{N(t)\}_{t\in\mathbb{R}_+}$ be the number of departure/arrival epochs up to and including time t. Observe that, for all $t \in \mathbb{R}_+$,

$$M_0(t) = \sum_{s=0}^{N(t)} R(s)$$

where $R(t)$ the reward in (5.13). Thus,

$$\begin{aligned} \rho_n^0 &= \lim_{t\to\infty} \frac{M_0(t)}{t} = \lim_{t\to\infty} \frac{\sum_{s=0}^{N(t)} R(s)}{t} \\ &= \lim_{t\to\infty} \frac{N(t)}{t} \cdot \lim_{t\to\infty} \frac{\sum_{s=1}^{N(t)} R(s)}{N(t)} = n\mu \cdot R_n, \quad \textit{a.s.} \end{aligned}$$

The first limit exists by the strong law of large numbers because $\{N(t)\}_{t\in\mathbb{R}_+}$ is a Poisson process with rate $n\mu$. The second limit exists because R_n exists and

$$\lim_{t\to\infty} N(t) = \infty, \quad \textit{a.s.},$$

again, by the strong law of large number. Hence, ρ_n^0 exists *a.s.*, and the lemma follows. □

Lemma 5.6 proves the existence of the server load per peer ρ_n^0, as defined in Section 5.1.2. It also shows that the computation of ρ_n^0 can be reduced to the computation the steady state reward R_n, as defined above. Just like C_n in Section 5.3.4, the latter can be computed for all query propagation mechanisms we are interested in through Lemma 5.4.

5.4 Random Walk Mechanism

Having established conditions under which our system is ergodic and the limits describing our two metrics of interest (ρ_n and ρ_n^0) exist, we turn our attention to the case where the query propagation mechanism is the delay-constrained random walk. The main result of this section is Theorem 5.1, stated in Section 5.2. It states that if (a) the overlay graph is an expander *w.h.p.*, and (b) TTL_n is proportional to the number of peers n, both the average load per peer and the server load generated by the delay-constrained random walk mechanism will be bounded. As discussed in Section 5.2, this suggests that a random walk on an unstructured peer-to-peer network can be used to significantly alleviate the traffic at the server —to the extent that having constant server bandwidth is sufficient— without imposing a significant burden on the peers.

The remainder of Section 5.4 is dedicated to the proof of this result. We first derive an upper bound on the query response time in terms of the request probability p_n, the stopping time TTL_n, and the steady relaxation time τ_n of the overlay graph (Section 5.4.1). This result yields an upper bound on the average traffic load per peer —again, in terms of the above three quantities (Section 5.4.2). We also derive upper and lower bounds on the server traffic load (Section 5.4.3); finally, applying the above bounds to the case where $\mathrm{TTL}_n = \Theta(n)$ and the overlay is an expander yields Theorem 5.1 (Section 5.4.4).

Our proofs rely on the bounds on the hitting time of a random walk presented in Chapter 3. Intuitively, under the assumption that the overlay graph is vertex balanced, each query propagation starts uniformly outside the set of positive peers (*i.e.*, peers having the data item). Because of this, Theorems 3.5 and 3.6 can be used to relate the time to hit the set of positive peers to the relaxation time of the overlay graph.

We note that the general bounds we derive (in terms of p_n, TTL_n, and τ_n) can be used to address more general cases than the one we consider here, including non-expander graphs and non-constant traffic loads. Extensions of this kind can be found in Section 5.7.

5.4.1 Expected Query Response Time

We begin our analysis with a characterization of the query response time of the delay-constrained random walk. The reason is that, as discussed in the next section, this can be related to the number of message transmissions of a query propagation and, thus, to the traffic load at peers.

Let $\{D(t)\}_{t\in\mathbb{N}}$ be the process describing the response time of the t-th query propagation. We are interested in computing the steady state expected query response time D_n, *i.e.*,

$$D_n = \lim_{t\to\infty} \mathbb{E}[D(t)] = \lim_{t\to\infty} \frac{1}{t}\sum_{s=1}^{t} D(s)$$

where the last equality holds *a.s.* The following lemma computes this quantity.

Lemma 5.7. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and vertex balanced. Then, for any constant $\bar{\tau} > 1$, the steady state expected response time of a query that is issued with probability p_n under the delay-constrained random walk propagation mechanism is such that:*

$$D_n \leq \delta \cdot \left[\min\left(2\bar{\tau}/p_n, \mathrm{TTL}_n\right) + \mathrm{TTL}_n(1-p_n)^{n-1} + \mathrm{TTL}_n p_n(1-\phi_n)\right].$$

where $\phi_n = \mathbf{P}(\tau_n \leq \bar{\tau})$ and τ_n the steady state relaxation time of $\{G(t)\}_{t\in\mathbb{N}}$, defined by (4.1)*.*

An important observation to make here is that not all queries have response time $\delta \cdot \mathrm{TTL}_n$ (which is the worst-case response time). For example, queries that are issued with constant probability (*e.g.*, $p_n = 0.1$) will experience a constant expected delay, if the overlay graph is an expander *w.h.p.*

Proof of Lemma 5.7. Let $\{N_r(t)\}_{t\in\mathbb{N}}$ be the number of requests for the item up to and including the t-th departure/arrival epoch. Then, by the law of large numbers,

$$\lim_{t\to\infty} \frac{N_r(t)}{t} = p_n, \quad \textit{a.s.}$$

Recall that the sequence $\{D(t)\}_{t\in\mathbb{N}}$ is the delay of the t-th query propagation. Define a reward function $\bar{D}(t)$ on the Markov chain $\{A(t), G(t)\}$ as follows

$$\bar{D}(t) = \begin{cases} D(N_r(t)), & \text{if the a request takes place at the } t\text{-th departure/arrival epoch} \\ 0, & \text{otherwise.} \end{cases}$$

The process $\{\bar{D}(t)\}_{t\in\mathbb{N}}$ can be seen as an extension of $\{D(t)\}_{t\in\mathbb{N}}$, which takes values only on departure/arrival epochs where the item is requested, to all departure/arrival epochs. Then,

$$\begin{aligned}\lim_{t\to\infty}\frac{\sum_{s=0}^{t}\bar{D}(s)}{t} &= \lim_{t\to\infty}\frac{N_r(t)}{t}\cdot\frac{\sum_{s=0}^{t}\bar{D}(s)}{N_r(t)}\\ &= \lim_{t\to\infty}\frac{N_r(t)}{t}\lim_{t\to\infty}\cdot\frac{\sum_{s=0}^{t}\bar{D}(s)}{N_r(t)}\\ &= p_n\cdot\lim_{t\to\infty}\frac{\sum_{s=0}^{N_r(t)}D(s)}{N_r(s)} = p_nD_n, \quad \textit{a.s.}\end{aligned}$$

where the last equality holds because

$$\lim_{t\to\infty}N_r(t)=\infty, \quad \textit{a.s.}$$

by the law of large numbers. Hence, to compute D_n, it suffices to compute

$$\bar{D}_n=\lim_{t\to\infty}\frac{\sum_{s=0}^{t}\bar{D}(s)}{t}=\lim_{t\to\infty}\mathbb{E}[\bar{D}(t)].$$

Let now $T_{i,A,G}$ be the time it takes a random walk starting from vertex $i\in V_n$ to hit set $A\subset V_n$ over the graph $G\in\mathbb{S}_{n,d}$. By convention, we allow A to be the empty set, defining $T_{i,\emptyset,G}\equiv\infty$. Then, $\bar{D}(t)$ can be written as

$$\bar{D}(t)=\begin{cases}0, & \text{if no request is issued at the } t\text{-th epoch},\\ \min(T_{i,A,G},\delta\mathrm{TTL}_n), & \text{if a request is issued at the } t\text{-th epoch},\end{cases}$$

where $T_{i,A,G}$ are independent among different requests. Therefore, Lemma 5.4 applies; by (5.6)

$$\begin{aligned}\bar{D}_n &= p_n\cdot\sum_{G\in\mathbb{S}_{n,d}}\nu_G\sum_{A\subsetneq V_n}\frac{n-|A|}{n}p_n^{|A|}(1-p_n)^{n-1-|A|}\mathbb{E}_{u_{A^c}}[\min(T_{A,G},\delta\mathrm{TTL}_n)]\\ &\le p_n\delta\mathrm{TTL}_n(1-\phi_n)+p_n\cdot\sum_{G\in\mathbb{S}_{n,d},\tau_G\le\bar{\tau}}\nu_G\sum_{A\subsetneq V_n}\frac{n-|A|}{n}p_n^{|A|}(1-p_n)^{n-1-|A|}\mathbb{E}_{u_{A^c}}[\min(T_{A,G},\delta\mathrm{TTL}_n)]\end{aligned}$$

where

$$\phi_n=\mathbf{P}(\tau_n\le\bar{\tau})$$

and

$$\mathbb{E}_{u_{A^c}}[\min(T_{A,G},\delta\mathrm{TTL}_n)]=\frac{1}{|A^c|}\sum_{i\in A^c}\mathbb{E}[\min(T_{i,A,G},\delta\mathrm{TTL}_n)].$$

For $A=\emptyset$, we have

$$\mathbb{E}_{u_{V_n}}[\min(T_{\emptyset,G},\mathrm{TTL}_n)]=\delta\mathrm{TTL}_n.$$

For $A \neq \emptyset$, by Corollary 3.1 of Chapter 3, we have that

$$\delta \cdot \left(\frac{n}{|A|} - 1 \right) \leq \mathbb{E}_{u_{A^c}}[T_{A,G}] \leq \delta \cdot \frac{\tau_G n}{|A|}$$

where τ_G the relaxation time of G. By the concavity of the $\min$ operator,

$$\mathrm{E}_{u_{A^c}}[\min(T_{A,G}, \delta \mathrm{TTL}_{\max})] \leq \min(\mathrm{E}_{u_{A^c}}[T_{AG}], \delta \mathrm{TTL}_{\max}).$$

The lemma therefore follows by substituting the above bounds and carrying out the calculations. In particular,

$$\begin{aligned}
\bar{D}_n &\leq p_n \delta \mathrm{TTL}_n (1 - \phi_n) + p_n (1 - p_n)^{n-1} \delta \mathrm{TTL}_n + \\
&\quad p_n \left(\sum_{G \in \mathbb{S}_{n,d}, \tau_G \leq \bar{\tau}} \nu_G \sum_{A \subsetneq V_n, A \neq \emptyset} \frac{n - |A|}{n} p_n^{|A|} (1 - p_n)^{n-|A|-1} \min \left(\frac{\delta \tau_G n}{|A|}, \delta \mathrm{TTL}_n \right) \right) \\
&\leq p_n \delta \mathrm{TTL}_n (1 - \phi_n) + p_n (1 - p_n)^{n-1} \delta \mathrm{TTL}_n + \\
&\quad \min \left(p_n \delta \bar{\tau} \sum_{A \subsetneq V_n, A \neq \emptyset} \frac{n - |A|}{n} p_n^{|A|} (1 - p_n)^{n-|A|-1} \frac{n}{|A|}, p_n \delta \mathrm{TTL}_n \right)
\end{aligned}$$

again, by the concavity of the $\min$ operator. On the other hand,

$$\begin{aligned}
&\sum_{A \subsetneq V_n, A \neq \emptyset} \frac{n - |A|}{n} p_n^{|A|} (1 - p_n)^{n-|A|-1} \frac{n}{|A|} = \\
&\qquad = \sum_{|A|=1}^{n-1} \frac{n!}{|A|!(n - |A|)!} \frac{(n - |A|)}{n} \left[p_n^{|A|} (1 - p_n)^{n-|A|-1} \right] \frac{n}{|A|} \\
&\qquad = \sum_{k=1}^{n-1} \frac{n!}{k \cdot k!(n - k - 1)!} \left[p_n^k (1 - p_n)^{n-k-1} \right] \\
&\qquad \leq \sum_{k=1}^{n-1} 2 \frac{n!}{(k+1)!(n - k - 1)!} \left[p_n^k (1 - p_n)^{n-k-1} \right] \\
&\qquad = \frac{2}{p_n} \sum_{k'=2}^{n} \frac{n!}{(k')!(n - k')!} \left[p_n^{k'} (1 - p_n)^{n-k'} \right] \\
&\qquad = \frac{1}{p_n} \cdot 2 \cdot \left(1 - (1 - p_n)^n - (1 - p_n)^{n-1} n p_n \right) \\
&\qquad = \begin{cases} \Theta\left(1/p_n\right), & p_n = \Omega\left(1/n\right) \\ O\left(n\right), & p_n = o\left(1/n\right) \end{cases}
\end{aligned}$$

which concludes the proof of Lemma 5.7. □

5.4.2 Average Load per Peer

Having discussed the expected response time of a query, we turn our attention to the average traffic load per peer ρ_n. The following result relates ρ_n to p_n, the request probability of the item, to TTL_n, the stopping time of the random walk, and to τ_n, the relaxation time of the overlay graph.

Theorem 5.4. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and vertex balanced. Then, for any constant $\bar{\tau} > 1$, the average traffic load per peer ρ generated by queries issued with probability p_n is such that*

$$\rho_n \leq \mu \cdot \left\{ \min\left[2\bar{\tau}, \mathrm{TTL}_n p_n\right] + \mathrm{TTL}_n p_n (1-p_n)^{n-1} + \mathrm{TTL}_n p_n (1-\phi_n) \right\}.$$

where $\phi_n = \mathbf{P}(\tau_n \leq \bar{\tau})$ and τ_n the steady state relaxation time of $\{G(t)\}_{t\in\mathbb{N}}$, defined by (4.1).

Proof. From Lemma 5.5, the average traffic load per peer is

$$\rho_n = \mu \cdot C_n$$

where

$$C_n = \lim_{t\to\infty} \mathbb{E}[C(t)] = \lim_{t\to\infty} \frac{1}{t} \sum_{s=0}^{t} C(s), \quad \textit{a.s.}$$

and $\{C(t)\}$ the following reward function:

$$C(t) = \begin{cases} 0, & \text{if no request is issued at the } t\text{-th epoch,} \\ C_{i,A,G}, & \text{if a request is issued at the } t\text{-th epoch,} \end{cases}$$

where $C_{i,A,G}$ is the number of messages generated by a query propagation given that, at the time the query is issued, the source peer is $i \in V_n$, the set of positive peers is $A \subset V_n$ and the overlay graph is $G \in \mathbb{S}_{n,d}$.

As in the proof of Lemma 5.7, let $D_{i,A,G}$ be the query response time under the same conditions as above (i,A and G). We will show that

$$\mathbb{E}[D_{i,A,G}] = \delta \mathbb{E}[C_{i,A,G}] \tag{5.14}$$

Consider a continuous-time random walk on G that starts on i and has a mean transition time δ. Assume that the random walk is unconstrained (*i.e.*, does not stop after TTL_n transmissions or when the set of positive peers is reached), and let $N_{i,G}(t)$ be the number of transitions

made by the walk up to and including time $t > 0$. Note that this is a Poisson process with rate $\frac{1}{\delta}$. Moreover, by definition,

$$N_{i,G}(D_{i,A,G}) = C_{i,A,G}.$$

Observe that $N_{i,G}(D_{i,A,G}) + 1$ is a stopping rule [Gal96] of the random walk. Therefore, by Wald's identity,

$$\mathbb{E}[N_{i,G}(D_{i,A,G}) + 1] \cdot \delta = \mathbb{E}[S_{N_{i,G}(D_{i,A,G})+1}],$$

Where S_k, $k \in \mathbb{N}$, is the epoch of the k-th transition of the walk. We have that

$$\mathbb{E}[S_{N_{i,G}(D_{i,A,G})+1}] = \mathbb{E}[D_{i,A,G}] + \mathbb{E}[y(D_{i,A,G})]$$

where $y(t)$ is the remaining time until the first transition after time t; that is, $y(D_{i,A,G})$ can be either (a)the time until the first transition after time TTL_n, or (b)the time until the first transition after the walk has reached (hit) the set of positive peers. In either case, by the memoryless property, $y(D_{i,A,G})$ is exponentially distributed with mean δ. This gives us

$$\mathbb{E}[S_{N_{i,G}(D_{i,A,G})+1}] = \mathbb{E}[D_{i,A,G}] + \delta$$

and, therefore,

$$\mathbb{E}[N_{i,G}(D_{i,A,G})]\delta = \mathbb{E}[D_{i,A,G}].$$

Eq. (5.14) thus follows since $\mathbb{E}[N_{i,G}(D_{i,A,G})] = \mathbb{E}[C_{i,A,G}]$.

Lemma 5.4 and (5.14) imply that

$$C_n = D_n/\delta,$$

where D_n the expected query response time of Lemma 5.7. Theorem 5.4 therefore follows from Lemma 5.7. □

5.4.3 Server Load

We conclude our analysis of the random walk mechanism by discussing the behaviour of the traffic load ρ_n^0, incurred at the server. Like Theorem 5.4, the following result relates ρ_n^0 to p_n, the request probability of the item, to TTL_n, the stopping time of the random walk, and to τ_n, the relaxation time of the overlay graph.

Theorem 5.5. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and vertex balanced. Then, for any constant $\bar{\tau} > 1$, the server traffic load ρ_n^0 generated by queries issued with probability p_n is such that*

$$\rho_n^0 \leq \mu\big[np_n(1-p_n+p_ne^{-\frac{\mathrm{TTL}_n}{n\bar{\tau}}})^{n-1}\phi_n + np_n(1-\phi_n)\big] \quad \textit{and}$$
$$\rho_n^0 \geq \mu\big[np_n(1-p_n+p_ne^{-\frac{2\mathrm{TTL}_n}{n}})^{n-1}(1-2p_n\bar{\tau})\phi_n\big],$$

where $\phi_n = \mathbf{P}(\tau_n \leq \bar{\tau})$ and τ_n the steady state relaxation time of $\{G(t)\}_{t\in\mathbb{N}}$, defined by (4.1)*.*

Proof. By Lemma 5.6,

$$\rho_n^0 = n\mu \cdot R_n \tag{5.15}$$

where

$$R_n = \lim_{t\to\infty}\mathbb{E}[R(t)] = \lim_{t\to\infty}\frac{1}{t}\sum_{s=0}^{t}R(s), \quad \textit{a.s.}$$

and $\{R(t)\}$ the reward defined in (5.13). Let $T_{i,A,G}$ be the time it takes a delay-constrained random walk starting at i to hit set A over a graph G. Then,

$$R(t) = \begin{cases} 0, & \text{if no request is issued at the } t\text{-th departure/arrival epoch} \\ \mathbb{1}_{T_{i,A,G}>\delta\mathrm{TTL}_n}, & \text{if a request is issued at the } t\text{-th departure/arrival epoch,} \end{cases}$$

where $T_{i,A,G}$ are independent among different requests. By Lemma 5.4,

$$R_n = p_n\sum_{G\in\mathbb{S}_{n,d}}\nu_G\Bigg[\sum_{A\subsetneq V_n}\frac{n-|A|}{n}p_n^{|A|}(1-p_n)^{n-1-|A|}\mathbf{P}_{u_{A^c}}(T_{A,G}>\delta\mathrm{TTL}_n)+(1-p_n)^{n-1}\Bigg]$$

where

$$\mathbf{P}_{u_{A^c}}(T_{A,G}>t) = \frac{1}{|A^c|}\sum_{i\in A^c}\mathbf{P}(T_{i,A,G}>t)$$

is the probability that a random walk starting uniformly outside set A will reach A at a time greater than t.

On the other hand, by Corollary 3.2 of Chapter 3, we have

$$\left(1-2\tau_G\frac{|A|}{n}\right)e^{-\frac{2|A|t}{n\delta}} \leq \mathbf{P}_{u_{A^c}}(T_{A,G}>t) \leq e^{-\frac{|A|t}{n\tau_G\delta}} \tag{5.16}$$

where τ_G the relaxation time of G. The above bounds give us

$$\begin{aligned}
R_n &\leq \sum_{G\in\mathbb{S}_{n,d}} p_n\nu_G\Bigg[\sum_{A\subsetneq V_n}\frac{n-|A|}{n}p_n^{|A|}(1-p_n)^{n-|A|-1}e^{-\frac{|A|\mathrm{TTL}_n}{n\tau_G}}\Bigg]\\
&= \sum_{G\in\mathbb{S}_{n,d}} p_n\nu_G\Bigg[\sum_{|A|=0}^{n-1}\binom{n-1}{|A|}p_n^{|A|}(1-p_n)^{n-1-|A|}e^{-\frac{|A|\mathrm{TTL}_n}{n\tau_G}}\\
&= \sum_{G\in\mathbb{S}_{n,d}} \nu_G p_n(1-p_n+p_ne^{-\frac{\mathrm{TTL}_n}{n\tau_G}})^{n-1}\\
&\leq p_n(1-p_n+p_ne^{-\frac{\mathrm{TTL}_n}{n\bar{\tau}}})^{n-1}\phi_n + p_n\cdot 1\cdot(1-\phi_n)
\end{aligned}$$

and

$$\begin{aligned}
R_n &\geq p_n\Bigg[\sum_{A\subsetneq V_n}\frac{n-|A|}{n}p_n^{|A|}(1-p_n)^{n-|A|-1}e^{-\frac{2|A|\mathrm{TTL}_n}{n}}\left(1-\frac{2|A|\bar{\tau}}{n}\right)\Bigg]\phi_n\\
&\quad + 0\cdot(1-\phi_n)\\
&= p_n\phi_n\Bigg[\sum_{|A|=0}^{n-1}\binom{n-1}{|A|}p_n^{|A|}(1-p_n)^{n-1-|A|}e^{-\frac{2|A|\mathrm{TTL}_n}{n}}\left(1-\frac{2|A|\bar{\tau}}{n}\right)\Bigg]\\
&\geq \phi_n p_n(1-p_n+p_ne^{-\frac{2\mathrm{TTL}_n}{n}})^{n-1}(1-2p_n\bar{\tau}).
\end{aligned}$$

where the last step follows by "Chebychev's Other Inequality", [AF, Chapter 3, page 23]:

$$\mathbb{E}[Xe^{-Xt}] \leq \mathbb{E}[X]\mathbb{E}[e^{-Xt}].$$

Theorem 5.5 therefore follows by substituting the above bounds in Eq. (5.15). □

5.4.4 Proof of Theorem 5.1

We are now ready to prove our main result for the random walk mechanism, namely, Theorem 5.1. First, an immediate implication of Theorem 5.4 is that if the overlay graph is an expander (*i.e.*, τ_n is bounded) with high probability, then the average traffic load per peer will be bounded, as long as TTL_n grows no faster than linearly in n:

Corollary 5.2. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and vertex balanced, and that there exists a constant $\bar{\tau}$ such that*

$$\phi_n = \mathbf{P}(\tau_n \leq \bar{\tau}) = 1 - o\left(\frac{1}{n}\right).$$

Then, if $\mathrm{TTL}_n = O(n)$, the average traffic load per peer is bounded, i.e.*,*

$$\rho_n = O(1),$$

irrespectively of p_n.

Proof. The first term of the bound in Theorem 5.4 is bounded as $\bar{\tau}$ is a constant. The third term is $o(1)$ as $\mathrm{TTL}_n = O(n)$ and $1 - \phi_n = o\left(\frac{1}{n}\right)$. Finally, the second term is of the order of

$$\mathrm{TTL}_n p_n (1-p_n)^{n-1} \leq \mathrm{TTL}_n p_n e^{-(n-1)p_n} = \frac{\mathrm{TTL}_n}{n-1}(n-1)p_n e^{-(n-1)p_n}$$

which is bounded because xe^{-x} is a bounded function.

Note that, if $p_n = o\left(\frac{1}{n}\right)$, the average traffic load per peer is in fact decreasing: the first term is decreasing as $\mathrm{TTL}_n p_n = o(1)$, and the second term is decreasing, as $\lim_{x\to\infty} xe^{-x} = 0$. □

To understand the intuition behind the above result assume that the overlay graph is an expander *w.h.p.* and that $\mathrm{TTL}_n = O(n)$. Theorem 5.4 implies that queries can be categorized under two regimes: frequent queries, for popular items that are requested with probability $p_n = \Omega(1/\mathrm{TTL}_n)$, and infrequent queries, for items that are requested with probability $p_n = o(1/\mathrm{TTL}_n)$. Intuitively, if p_n is high, and the item is popular, peers generate many queries for an item. For example, if p_n is constant, the number of queries generated by peers per second is linear in n. On the other hand, because peers store and share the items they request, a popular item will be widely available within the peer-to-peer network. If the overlay graph is an expander *w.h.p.*, queries for such items are served within approximately $1/p_n$ hops, where p_n is the request probability of the item. Thus, Theorem 5.4 implies that frequent queries generate no more than a constant amount of traffic per peer: although such queries are issued often (with rate $\mu n p_n$), they are served within a small number of hops ($O(1/p_n)$), and the overall traffic they generate at each peer is small.

Unpopular items, whose request probabilities are $o(1/\mathrm{TTL}_n)$, are not widely replicated. As a result, queries for such items may require many ($O(\mathrm{TTL}_n)$) message transmissions. Nonetheless, such queries do not occur as often as frequent queries; in fact, as $p_n = o(1/\mathrm{TTL}_n)$, the overall traffic load per peer they generate is decreasing.

Theorem 5.5 also has an interesting implication in the case of expander overlays. In particular, if the overlay graph is an expander (*i.e.*, τ_n is bounded) with high probability, then the average traffic load per peer will be bounded, as long as TTL_n grows no slower than linearly in n:

Corollary 5.3. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and vertex balanced, and that there exists a constant $\bar{\tau}$ such that*

$$\phi_n = \mathbf{P}(\tau_n \leq \bar{\tau}) = 1 - o\left(\frac{1}{n}\right).$$

Then, if $\mathrm{TTL}_n = \Omega(n)$, the traffic load at the server is bounded, i.e.*,*

$$\rho_n^0 = O(1)$$

irrespectively of p_n.

Proof. The second term in the upper bound of ρ_n^0 in Theorem 5.5 is decreasing, as $n(1-\phi_n) = o(1)$. The first term is

$$\begin{aligned} np_n(1-p_n+p_ne^{-\frac{\mathrm{TTL}_n}{n\bar{\tau}\delta}})^{n-1} &\leq np_n(1-p_n+p_ne^{-\frac{cn}{n\bar{\tau}\delta}})^{n-1}, \quad \text{for large } n \\ &\leq np_ne^{-C(\bar{\tau},\delta,c)(n-1)p_n} \end{aligned}$$

which is bounded as xe^{-x} is bounded, and the corollary follows.

Note that the "worst-case" load is for $p_n = \Theta\left(\frac{1}{n}\right)$: for such p_n, ρ_n^0 is upper-bounded by a constant (it is actually $\Theta(1)$, by the lower bound of Theorem 5.5). For $\lim_{n\to\infty} np_n = 0$ or $\lim_{n\to\infty} np_n = \infty$, the traffic load is decreasing, as xe^{-x} converges to zero if x tends to either zero or infinity. □

Some intuition on Corollary 5.3 can again be gained by considering popular and unpopular items separately. If the overlay graph is an expander *w.h.p.* and $\mathrm{TTL}_n = \Omega(n)$, popular items are very likely to be located within the peer-to-peer system. As a result, the probability that a query for a popular item reaches the server is quite low and, even though queries for such items are very frequent, the overall traffic they incur on the server is small. On the other hand, if queries for a item are infrequent, peers storing the item are less likely to be in the system when a query takes place and such queries are more likely to reach the server. However, as such queries are not frequent to begin with, the overall traffic load they contribute to the server is again small.

Theorem 5.1 immediately follows from Corollaries 5.2 and 5.3. In particular, the above two corollaries imply that, if the overlay graph is an expander *w.h.p.*, and $\mathrm{TTL}_n = \Theta(n)$, then *both* the average load per peer *and* the server load will be bounded, irrespectively of p_n.

5.5 Expanding Ring Mechanism

As we saw in the previous section, the choice of TTL_n determines the expected query response time D_n of a delay-constrained random walk search, as D_n is of the order of TTL_n in the worst case (over all probabilities p_n). For example, if we choose TTL_n to be proportional to n in order to keep the server load and the load per peer bounded as n grows, then the query response time will grow linearly in n for queries for unpopular items. On the other hand, items that are requested with probability $\omega(1/n)$ will have sub-linear expected response times. In the worst case however, while the random walk mechanism leads to good performance in terms of

the load at the server and at individual peers, it might have a poor performance in terms of the query response time.

This motivates our study of the expanding ring mechanism. Our main result —namely, Theorem 5.2— states that the expanding ring with a logarithmic stopping time TTL_n yields an average load per peer and a server load that are almost constant in n. In addition, the query response time under the aforementioned mechanism is $O\left(\log^2 n\right)$, *i.e.*, considerably smaller than the one observed under the random walk.

The remainder of this section focuses on the derivation of the above result. We first introduce *distance layered search* (Section 5.5.1), a class of search mechanisms for which peers are visited in the order of their distance from the query source. In particular, we show that the message cost of an expanding ring can be upper-bounded by the message cost of a distance layered search on the same graph. This bound allows us to derive upper bounds on both the average traffic load per peer (Section 5.5.2) and the server traffic load (Section 5.5.3) in terms of the request brobability of a query. Investigating the suprema of these bounds over all p_n gives us Theorem 5.2 (Section 5.5.4).

Our proofs rely on Theorem 3.15 and its corollary (Corollary 3.11), describing the vertex expansion over small sets in random graphs whose distributions satisfy precisely the assumptions of Theorem 5.2. By definition, such graphs are expanders *a.a.s.* Nonetheless, our analysis can be extended to more general graphs using Kahale's Theorem (Theorem 3.14) and its Corollary 3.10; such an extension is presented in Section 5.7.

5.5.1 Distance Layered Search

We begin our analysis of the expanding ring by first briefly describing a search mechanism called *distance layered search*, a special case of which is the well-known breadth first search algorithm [Wes01]. As we show below, the message cost of the expanding ring query propagation mechanism can be expressed in terms of the message cost of a distance layered search; this motivates our interest in this mechanism.

Given a graph $G(V_n, E)$, a distance layered search (or, simply, DLS) starting at vertex i is a search in which the vertices of the graph are visited in an order of increasing distance from i. Formally, a DLS is a finite sequence of vertices

$$j_1, j_2, \ldots, j_n$$

such that

(1) $j_1 = i$, *i.e.*, the search starts at vertex i,

(2) $j_k \neq j_\ell$ for all $k \neq \ell$, *i.e.*, no vertex is visited twice, and

(3) If $1 \leq k \leq \ell \leq n$, then $dist(i, j_k) \leq dist(i, j_\ell)$, where $dist(i, j)$ the edge distance between vertices i and j.

There are more than one DLS sequences starting from i, as the distance from i defines only a partial ordering among vertices. Assuming that $V_n = \{1, \ldots, n\}$, a typical way of breaking ties among vertices at equal distance from i is visiting a vertex with smaller label first. The resulting search satisfies the following condition in addition to (1)–(3):

(4) If $1 \leq k < \ell \leq n$ and $dist(i, j_k) = dist(i, j_\ell)$ then $j_k < j_\ell$ (under the natural ordering of $\{1, \ldots, n\}$).

For concreteness, we will assume in the following that ties are always broken according to the above rule, and that the DLS sequence is the unique sequence satisfying conditions (1)–(4). This is not however necessary for any of the results presented below, which hold for any valid DLS sequence satisfying conditions (1)–(3).

We will view DLS as a query propagation mechanism: formally, the DLS sequence starting at $i \in V_n$ determines the order in which peers receive the query message sent by vertex i. We will consider a stopped version, in which transmissions occur until either $\mathrm{TTL}_n \leq n$ vertices are visited or the data item is located. More precisely, let $A \subseteq V_n$ be the set of positive peers at the time a query is issued and consider a DLS sequence $j_1, \ldots, j_n$. If A is non-empty, let

$$k \equiv \min\{\ell : j_\ell \in A\}$$

be the first vertex in the DLS sequence belonging to A. If A is empty, define

$$k \equiv n.$$

Then, for $\mathrm{TTL}_n \leq n$, the number of vertices visited by the stopped DLS (*i.e.*, that receive the query message) will be $\min(k, \mathrm{TTL}_n + 1)$. In other words, the vertices visited will be the subsequence

$$j_1, j_2, \ldots, j_{\min(k, \mathrm{TTL}_n + 1)}$$

of the DLS sequence.

Contrary to the random walk, DLS is deterministic. Moreover, DLS is not distributed: assuming that transmissions are sequential, peers need to know the entire topology of the graph

to decide when it is their turn to transmit. Because of this, implementing DLS in a real peer-to-peer network is challenging; our interest in DLS however stems from its relationship to the expanding ring mechanism, which is explored in detail below, through Lemma 5.9 . For this reason, we proceed by analyzing the traffic load generated by DLS.

The Number of Message Transmissions of a DLS search

The stationary distribution of $\{A(t), G(t)\}_{t\in\mathbb{N}}$ is such that a peer is positive with probability p_n and null with probability $1-p_n$, independently of other peers. Since DLS passes through vertices only once, the steady state probability that it requires k transmissions before it encounters a positive peer (or, that it visits k peers, including the source peer), will be

$$(1-p_n)^{k-1}p_n.$$

This is indeed true, and it is stated more formally in the following Lemma. In particular, the lemma establishes that, in steady state, the number C of transmissions the DLS with a stopping time TTL_n is distributed as a truncated geometric random variable:

$$\mathbf{P}(C=k) = \begin{cases} (1-p_n)^{k-1}p_n, & \text{if } 1 \le k < \mathrm{TTL}_n \\ (1-p_n)^{\mathrm{TTL}_n-1}, & \text{if } k = \mathrm{TTL}_n \\ 0, & \text{if } j > \mathrm{TTL}_n. \end{cases}$$

Moreover, the above distribution act does not depend on the topology of the overlay graph. In fact, apart from the requirement that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and vertex reversible, the stationary distribution of $\{G(t)\}_{t\in\mathbb{N}}$ does appear in the lemma.

Lemma 5.8. *Consider the following reward function, defined over the Markov chain* $\{A(t), G(t)\}_{t\in\mathbb{N}}$*:*

$$R(t) = \begin{cases} 0, & \textit{if no request is issued at the } t\textit{-th epoch}, \\ f(C_{i,A,G}), & \textit{if a request is issued at the } t\textit{-th epoch}, \end{cases}$$

where $f : \mathbb{R} \to \mathbb{R}$ *a real function and* $C_{i,A,G}$ *the number of message transmissions of a DLS starting at vertex* i*, given that the set of positive peers is* A *and that the graph is* G*. Then, if* $\{G(t)\}_{t\in\mathbb{N}}$ *is ergodic and vertex balanced,*

$$R_n = p_n \cdot \left[f(\mathrm{TTL}_n)\cdot(1-p_n)^{\mathrm{TTL}_n-1} + \sum_{k=1}^{\mathrm{TTL}_n-1} f(k)(1-p_n)^{k-1}p_n \right]$$

$$= \lim_{t\to\infty} \mathbb{E}[R(t)] = \lim_{t\to\infty} \frac{1}{t}\sum_{s=0}^{\infty} R(s), \quad \text{a.s.}$$

Proof. The reward $\{R(t)\}_{t\in\mathbb{N}}$ is of the form appearing in Lemma 5.4 —note though that $C_{i,A,G}$ is deterministic. Hence,

$$\begin{aligned} R_n &= p_n \cdot \sum_{G\in\mathbb{S}_{n,d}} \nu_G \sum_{A\subsetneq V_n} \frac{n-|A|}{n} p_n^{|A|}(1-p_n)^{n-1-|A|} \sum_{i\in A^c} \frac{1}{n-|A|} f(C_{i,A,G}) \\ &= p_n \cdot \sum_{G\in\mathbb{S}_{n,d}} \nu_G \sum_{i\in V_n} \frac{1}{n} \sum_{A\subsetneq V_n, A\not\ni i} p_n^{|A|}(1-p_n)^{n-1-|A|} f(C_{i,A,G}) \end{aligned}$$

Conditioned on the graph G and the source vertex i, the DLS starting at i over G is the unique deterministic sequence

$$J_{i,G} = \{j_1, j_2, \ldots, j_n\}$$

satisfying the conditions (1)-(4) we stated above to define a (lexicographic) DLS. Let

$$J_{i,G}(k) = \{j_1, j_2, \ldots, j_k\}, \quad k = 1, \ldots, n \tag{5.17}$$

be the subsequence including the k first elements of $J_{i,G}$. The event $C_{i,A,G} = k$, for $1 \le k \mathrm{TTL}_n$ can be written as

$$J_{i,G}(k) \cap A = \emptyset, j_{k+1} \in A,$$

while $C_{i,A,G} = \mathrm{TTL}_n$ is the event

$$J_{i,G}(\mathrm{TTL}_n) \cap A = \emptyset.$$

Thus, R_n can be rewritten as

$$R_n = p_n \cdot \sum_{G\in\mathbb{S}_{n,d}} \nu_G \sum_{i\in V_n} \frac{1}{n} \sum_{k=1}^{\mathrm{TTL}_n} f(k) \cdot \nu_{C_{i,A,G}=k} \tag{5.18}$$

where

$$\begin{aligned} \nu_{C_{i,A,G}=k} &= \begin{cases} \sum_{A: i\notin A} \mathbb{1}_{A\cap J_{i,G}(k)=\emptyset \wedge J_{i,G}(k+1)\neq\emptyset} p_n^{|A|}(1-p_n)^{n-1-|A|}, & \text{if } 1 \le k < \mathrm{TTL}_n, \\ \sum_{A: i\notin A} \mathbb{1}_{A\cap J_{i,G}(\mathrm{TTL}_n)=\emptyset} p_n^{|A|}(1-p_n)^{n-1-|A|}, & \text{if } k = \mathrm{TTL}_n. \end{cases} \\ &= \begin{cases} (1-p_n)^{k-1} p_n, & \text{if } 1 \le k < \mathrm{TTL}_n, \\ (1-p_n)^{\mathrm{TTL}_n - 1}, & \text{if } k = \mathrm{TTL}_n. \end{cases} \end{aligned}$$

The last equality can be shown by counting the number of sets that satisfy the conditions in the summation. Intuitively, however, the distribution over the sets A is such that an element $j \neq i$

is included with probability p_n (and excluded with probability $(1-p_n)$), independently of other elements. This yields the simple expressions above. The theorem follows by substituting the above quantities for $\nu_{C_{i,A,G}=k}$ in (5.18), and noticing that, after the substitution, the summand does not depend on i. □

Expanding Ring and DLS

Let $G(V_n, E) \in \mathbb{G}_{n,d}$ be a connected d-regular graph of size n. Recall, from Section 3.3.3 that

$$g_G(k) = \min_{A \subset V_n, |A| \leq k} \frac{|\delta(A)|}{|A|}, \quad k \leq \frac{n}{2}$$

and

$$g'_G(k) = \min_{A \subset V_n, |A| < k} \frac{|\Gamma(A)|}{|A|}, \quad k \leq \frac{n}{2}$$

are the vertex expansion for small sets and its modified version.

Lemma 5.9. *Let $G(V_n, E)$ be a d-regular graph of size n, where $d \geq 3$. Fix an $0 < \epsilon \leq \frac{1}{2}$, and let*

$$C \equiv C_{i,A,G}$$

be the number of message transmissions of an expanding ring query propagation mechanism with

$$\mathrm{TTL}_n = \log_{(d-1)}(\epsilon n - 1) + 2 - \log_{(d-1)} d, \tag{5.19}$$

given that the query propagation starts from a vertex $i \in V_n$, the set of positive peers is $A \subset V_n$ and the overlay graph is G. Similarly, let

$$C' \equiv C'_{i,A,G}$$

be the number of message transmissions of a DLS with

$$\mathrm{TTL}'_n = (d+1)g'_G(\epsilon n)^{\mathrm{TTL}_n - 1} - 1 = (d+1)g'_G(\epsilon n)^{\log_{(d-1)}(\epsilon n - 1) + 1 - \log_{(d-1)} d} - 1, \tag{5.20}$$

under the same conditions i, A and G. Then, the following hold:

(a) If the DLS locates the data item within the peer-to-peer system, then so will the expanding ring.

(b) The expanding ring will terminate within k stages, where

$$\log_{(d-1)} C' + 1 - \log_{(d-1)} d \leq k \leq \log_{g'_G(\epsilon n)}(C' + 1) + 2 - \log_{g'_G(\epsilon n)}(d+1).$$

(c) The message costs C, C' of the two mechanisms are related by:

$$C \leq \frac{d^2(d-1)^{2-\log_{g'_G(\epsilon n)}(d+1)}}{d-2} \cdot (C'+1)^{\log_{g'_G(\epsilon n)}(d-1)} - \frac{d^2}{d-2}$$

Finally, the lemma still holds if $g'_G(\epsilon n)$ is replaced by $1 + g_G(\epsilon n)$ in the definition of TTL'_n as well as in statements (b) and (c).

Proof. Let $B(j) \subset V_n$ be the set of vertices reached by an expanding ring with $\mathrm{TTL}_n = j$. By definition $B(0) = \{i\}$ and $B(1) = \{i\} \cup \Gamma(\{i\})$. For $j \geq 1$, we have that

$$B(j+1) = B(j) \cup \delta(B(j)) = \Gamma(B(j)), \tag{5.21}$$

where the latter equality holds because $B(j) \subset \Gamma(B(j))$ for all $j \geq 1$, by the definition of the expanding ring. Hence, if $j \geq 1$ and $|B(j)| \leq \epsilon n$,

$$\frac{|B(j+1)|}{|B(j)|} \geq g'_G(\epsilon n)$$

and, by induction on $j \geq 1$

$$|B(j)| \geq (d+1) \cdot g'_G(\epsilon n)^{j-1} \tag{5.22}$$

provided that $|B(j-1)| < \epsilon n$. On the other hand, for every $j \geq 1$,

$$\frac{|B(j+1) \setminus \{i\}|}{|B(j) \setminus \{i\}|} \leq d-1$$

as peers other than i do not transmit to the peer that forwarded them the query. Thus,

$$|B(j)| \leq 1 + d \cdot (d-1)^{j-1}. \tag{5.23}$$

From (5.22) and (5.23) we get that

$$(d+1) \cdot g'_G(\epsilon n)^{j-1} \leq |B(j)| \leq 1 + d \cdot (d-1)^{j-1} \tag{5.24}$$

provided that $1 + d \cdot (d-1)^{j-2} \leq \epsilon n$. For $j = \mathrm{TTL}_n = \log_{(d-1)}(\epsilon n - 1) + 2 - \log_{(d-1)} d$,

$$1 + d(d-1)^{\log_{(d-1)}(\epsilon n - 1) + 2 - \log_{(d-1)} d - 2} = \epsilon n$$

Hence, the total number of vertices reached with the above value of TTL_n is

$$|B(\mathrm{TTL}_n)| \geq (d+1) \cdot g'_G(\epsilon n)^{\log_{(d-1)}(\epsilon n - 1) + 2 - \log_{(d-1)} d - 1} = \mathrm{TTL}'_n + 1. \tag{5.25}$$

Eq. (5.25) implies that if the DLS with TTL'_n succeeds in locating the data item, so will the expanding ring with the given TTL_n. This proves statement (a).

Suppose now that data item is located by the DLS at message cost C', thus covering $C'+1$ vertices. Then, the expanding ring locates the item in k stages, such that

$$S(k) \geq C' + 1.$$

Eq. (5.24) implies that

$$k \geq \log_{(d-1)} C' + 1 - \log_{(d-1)} d. \tag{5.26}$$

Moreover, since the expanding ring locates the item at k and not $k-1$ stages, we have that

$$S(k-1) \leq C' + 1,$$

or, by eq. (5.24),

$$k \leq \log_{g'_G(\epsilon n)}(C'+1) + 2 - \log_{g'_G(\epsilon n)}(d+1). \tag{5.27}$$

Suppose, on the other hand, that the data item is not located by the DLS; then, the DLS transmits $C' = \mathrm{TTL}'_n$ messages. Assume again that the expanding ring terminates after k stages, either locating the item or not. As it must cover at least $C'+1$ vertices to locate the item, we still have that

$$S(k) \geq C' + 1$$

and, therefore, (5.26) still holds. Moreover, k cannot be more than TTL_n. By (5.20) we have that

$$\mathrm{TTL}_n = \log_{g'_G(\epsilon n)}(\mathrm{TTL}'_n + 1) + 1 - \log_{g'_G(\epsilon n)}(d+1)$$

which implies that

$$k \leq \mathrm{TTL}_n = \log_{g'_G(\epsilon n)}(\mathrm{TTL}'_n + 1) + 1 - \log_{g'_G(\epsilon n)}(d+1)$$

so (5.27) again holds. This proves statement (b).

The number of transmissions C of the expanding ring that terminates after k steps are

$$\sum_{j=1}^{k}(|B(j)|-1) \leq C \leq d \cdot \sum_{j=1}^{k}(|B(j)|-1). \tag{5.28}$$

To see this, observe that if $C(j)$ the number of transmissions on the j-th stage then

$$C(j) \geq |B(j)| - 1$$

as every peer visited, except the source i, receives the message at least once. On the other hand, no peer transmits the same message at the same stage more than once. Hence, each peer visited cannot receive the message more than d times and, thus,

$$C(j) \leq d \cdot (|B(j)| - 1).$$

Summing up over j yields (5.28). From (5.23) we have that

$$\sum_{j=1}^{k}(|B(j)| - 1) \leq \sum_{j=1}^{k}\left[1 + d(d-1)^{j-1} - 1\right] = d\frac{(d-1)^k - 1}{d-2} \tag{5.29}$$

The last bound, along with (5.28) and the bound (5.27) on k, give

$$\begin{aligned} C &\leq d^2 \frac{(d-1)^{\log_{g'_G(\epsilon n)}(C'+1)+2-\log_{g'_G(\epsilon n)}(d+1)} - 1}{d-2} \\ &= \frac{d^2(d-1)^{2-\log_{g'_G(\epsilon n)}(d+1)}}{d-2} \cdot (C'+1)^{\log_{g'_G(\epsilon n)}(d-1)} - \frac{d^2}{d-2} \end{aligned}$$

which proves statement (c).

Finally, the fact that we could have used $1 + g_G(\epsilon n)$ instead of $g'_G(\epsilon n)$ follows by (5.21), and is an immediate implication of the fact that, for $j \geq 1$,

$$|B(j+1)| = |B(j)| + |\delta(B(j))|.$$

□

5.5.2 Average Load per Peer

Recall from Section 5.2.2 that our analysis of the expanding ring mechanism will be on a overlay graphs $\{G(t)\}_{t\in\mathbb{R}_+}$ whose state space is restricted to simple graphs, and whose stationary distribution is contiguous to the uniform distribution over $\mathbb{G}_{n,d}$. The latter property is satisfied by the uniform distribution over any one of the following sets: $\mathbb{G}_{n,d}$, the set of d-regular (simple) graphs, $\mathbb{CG}_{n,d}$, the set of connected d-regular graphs, $\mathbb{H}_{n,d}$, the set of d-regular graphs having a complete Hamiltonian decomposition, and $\mathbb{I}_{n,d}$, the set of d-regular graphs that have a 1-factorization.

Under this assumption, the average traffic load per peer can be bounded according to the following theorem.

Theorem 5.6. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and vertex balanced, that its state space is $\mathbb{S}_{n,d} \subseteq \mathbb{G}_{n,d}$, where $d \geq 3$, and that its stationary distribution is uniform over $\mathbb{S}_{n,d}$ and*

contiguous to the uniform distribution over $\mathbb{G}_{n,d}$. Then, for every $\delta > 0$ there exists a $\mathrm{TTL}_n = \Theta\left(\log_{(d-1)} n\right)$ such that average traffic load per peer of the expanding ring is

$$\rho_n = O\left(n^{1-\frac{\delta}{2}} p_n\right) + \begin{cases} O\left(\left(\frac{1}{p_n}\right)^{\alpha(d,\delta)-1}\right), & \text{if } p_n = \Omega\left(\frac{1}{n}\right) \\ O\left(n^{\alpha(d,\delta)} p_n\right), & \text{if } p_n = o\left(\frac{1}{n}\right) \end{cases} \tag{5.30}$$

where

$$\alpha(d,\delta) = \log_{(d-1-\delta)}(d-1).$$

Moreover, irrespectively of p_n,

$$\rho_n = O\left(n^{1-\frac{\delta}{2}} p_n\right) + O\left(n^{\alpha(d,\delta)-1}\right).$$

Proof. We prove Theorem 5.6 only for the case where $\mathbb{S}_{n,d} = \mathbb{G}_{n,d}$. The case where $\mathbb{S}_{n,d} \subset \mathbb{G}_{n,d}$ can be dealt with the same way, by using Corollary 3.11 instead of Theorem 3.15 wherever the latter appears below.

Our proof requires two auxiliary lemmas. The first gives an upper bound the worst-case message cost of an expanding ring search.

Lemma 5.10. *The number of query messages propagated by an expanding ring mechanism over a d-regular graph is no more than*

$$d^2 \frac{(d-1)^{\mathrm{TTL}_n} - 1}{d-2},$$

where TTL_n the stopping time of the expanding ring.

Proof. This bound can be derived from Eq. (5.28) and (5.29). Eq. (5.28) and (5.29) imply that the number of transmissions of the expanding ring that terminates after k steps are less than or equal to $d^2 \frac{(d-1)^k - 1}{d-2}$. The lemma therefore follows by taking $k = \mathrm{TTL}_n$. □

Statement (c) of Lemma 5.9 bounds the message cost of an expanding ring search in terms of the message cost of a DLS, raised to a certain (possibly fractional) power. By Lemma 5.8, the message cost of a DLS is distributed according to a truncated geometric random variable. These two observations motivate the following technical lemma, which characterizes the asymptotic behaviour of the fractional moments of a sequence of truncated geometric random variables. The lemma is a simple consequence of the properties of truncated geometric random variables; we include its proof only for the convenience of the reader. We use the shorthand $f_n \sim g_n$ for $\lim_{n\to\infty} \frac{f_n}{g_n} = 1$.

Lemma 5.11. *Let X_n be a sequence of truncated geometric random variables,* i.e.*, a sequence of integer random variables with distribution*

$$\mathbf{P}(X_n = k) = \begin{cases} (1-p_n)^{k-1}p_n, & \text{if } 1 \le k < \xi_n \\ (1-p_n)^{\xi_n - 1}, & \text{if } k = \xi_n \\ 0, & \text{if } k > \xi_n. \end{cases} \tag{5.31}$$

where $0 < p_n < 1$ and $\lim_{n\to\infty} \xi_n = \infty$. Then, for any constant $\alpha > 0$, the following two statements hold:

(a)

$$\mathbb{E}[(X_n)^\alpha] \sim \begin{cases} p_n^{-\alpha}\Gamma(\alpha+1), & \text{for } p_n = \Omega\left(\frac{1}{\xi_n}\right) \\ \xi_n^\alpha, & \text{for } p_n = o\left(\frac{1}{\xi_n}\right) \end{cases}$$

where

$$\Gamma(\alpha) \equiv \int_0^\infty t^{\alpha-1}e^{-t}dt$$

the Gamma function.

(b) Irrespectively of p_n,

$$\mathbb{E}[(X_n)^\alpha] = O\left(\xi_n^\alpha\right),$$

and this bound is tight: for $p_n = 1/\xi_n$,

$$\mathbb{E}[(X_n)^\alpha] = \Theta\left(\xi_n^\alpha\right).$$

Proof. We have that

$$\begin{aligned} \mathbb{E}[(X_n)^\alpha] &= \sum_{j=1}^{\xi_n - 1} j^\alpha (1-p_n)^{j-1}p_n + \xi_n^\alpha (1-p_n)^{\xi_n - 1} \\ &\sim \int_{x=0}^{\xi_n - 1} x^\alpha (1-p_n)^{x-1}p_n dx + \xi_n^\alpha (1-p_n)^{\xi_n - 1} \end{aligned}$$

by the monotonicity of the summand for large j. We have that

$$\begin{aligned} \mathbb{E}[(X_n)^\alpha] &\sim \int_{x=0}^{\xi_n - 1} x^\alpha (1-p_n)^{x-1}p_n dx + \xi_n^\alpha (1-p_n)^{\xi_n - 1} \\ &= p_n(-\log(1-p_n))^{-1-\alpha}\Gamma(1+\alpha) \\ &\quad - \alpha p_n(-\log(1-p_n))^{-1-\alpha}\Gamma(\alpha, -\log(1-p_n)(\xi_n - 1)) \\ &\quad + \frac{p_n}{\log(1-p_n)}(\xi_n - 1)^\alpha (1-p_n)^{\xi_n - 1} + \xi_n^\alpha (1-p_n)^{\xi_n - 1} \end{aligned}$$

where

$$\Gamma(\alpha) = \int_0^\infty t^{\alpha-1}e^{-t}dt$$

the Gamma function and

$$\Gamma(\alpha, z) = \int_z^\infty t^{\alpha-1}e^{-t}dt.$$

For $p_n < 1$, $\log(1-p_n) \sim -p_n$ so

$$\mathbb{E}[[X_n]^\alpha] \sim p_n^{-\alpha}\left(\Gamma(\alpha+1) - \alpha\Gamma(\alpha, p_n\xi_n)\right)$$

Therefore, for $p_n\xi_n = \Omega(1)$,

$$\mathbb{E}[A^\alpha] \sim p_n^{-\alpha}\Gamma(\alpha+1).$$

For $p_n\xi_n \to 0$, we have that

$$\left(\Gamma(\alpha+1) - \alpha\Gamma(\alpha, p_n\xi_n)\right) \to 0.$$

By L'Hospital's rule, the order of convergence is $\left(\Gamma(\alpha+1) - \alpha\Gamma(\alpha, z)\right) \sim z^\alpha$ and, therefore, for $p_n\xi_n \to 0$,

$$\mathbb{E}[A^\alpha] \sim \xi_n^\alpha.$$

This proves the first part of the lemma. For the second part, the upper bound follows immediately, as

$$\mathbb{E}[(X_n)^\alpha] = \sum_{j=1}^{\xi_n-1} j^\alpha(1-p_n)^{j-1}p_n + \xi_n^\alpha(1-p_n)^{\xi_n-1} \tag{5.32}$$

$$= \xi_n^\alpha\left(\sum_{j=1}^{\xi_n-1}\left(\frac{j}{\xi_n}\right)^\alpha(1-p_n)^{j-1}p_n + (1-p_n)^{\xi_n-1}\right) \tag{5.33}$$

$$\leq \xi_n^\alpha\left(\sum_{j=1}^{\xi_n-1} 1\cdot(1-p_n)^{j-1}p_n + (1-p_n)^{\xi_n-1}\right) \leq \xi_n^\alpha\cdot 1. \tag{5.34}$$

On the other hand, if $p_n = \Theta(1/\xi_n)$, for large enough n

$$\mathbb{E}[(X_n)^\alpha] \geq \xi_n^\alpha(1-\frac{c}{\xi_n})^{\xi_n-1}$$

and

$$\lim_{\xi_n\to\infty}(1-\frac{c}{\xi_n})^{\xi_n-1} = e^{-c},$$

as $\lim_{n\to\infty}\xi_n = \infty$.

□

From Theorem 3.15 in Chapter 3, for every $\delta > 0$, there exists an $\epsilon > 0$ such that

$$\mathbf{P}\left(1 + g_{\mathcal{G}_{n,d}}(\epsilon n) \geq c(d,\delta)\right) \geq \phi_n. \tag{5.35}$$

where

$$c(d,\delta) = d - 1 - \delta$$

and

$$\phi_n = 1 - O\left(n^{-\frac{\delta}{2}}\right).$$

Take the TTL_n to be

$$\mathrm{TTL}_n = \log_{(d-1)}(\epsilon n - 1) + 2 - \log_{(d-1)} d. \tag{5.36}$$

From Lemma 5.5, the average traffic load per peer is

$$\rho_n = \mu \cdot C_n \tag{5.37}$$

where C_n the steady state expectation

$$C_n = \lim_{t\to\infty} \mathbb{E}[C(t)] = \lim_{t\to\infty} \frac{1}{t}\sum_{s=0}^{t} C(s), \quad \textit{a.s.}$$

of the reward function

$$C(t) = \begin{cases} 0, & \text{if no request is issued at the } t\text{-th epoch,} \\ C_{i,A,G}, & \text{if a request is issued at the } t\text{-th epoch,} \end{cases}$$

where $C_{i,A,G}$ is the number of messages generated by a query propagation given that, at the time the query is issued, the source peer is i, the set of positive peers is A and the overlay graph is G. From Lemma 5.4,

$$\begin{aligned} C_n &= p_n \cdot \sum_{G\in\mathbb{S}_{n,d}} \nu_G \sum_{A\subsetneq V} p_n^{|A|}(1-p_n)^{n-1-|A|} \sum_{j\in A^c} \frac{1}{n}\mathbb{E}[C_{i,A,G}] \\ &\overset{\text{(5.35),Lemma 5.10}}{\leq} p_n(1-\phi_n)W_n^1 + p_n \cdot W_n^2 \end{aligned} \tag{5.38}$$

where

$$W_n^1 \equiv d\frac{(d-1)^{\mathrm{TTL}_n} - 1}{d-2} \overset{(5.36)}{=} \Theta(n) \tag{5.39}$$

and

$$W_n^2 \equiv \sum_{G\in\mathbb{S}_{n,d}, 1+g_G(\epsilon n) > c(d,\delta)} \nu_G \sum_{A\subsetneq V} p_n^{|A|}(1-p_n)^{n-1-|A|} \sum_{j\in A^c} \frac{1}{n}\mathbb{E}[C_{i,A,G}].$$

By Lemma 5.9(c), we have that

$$W_n^2 \leq \sum_{G\in\mathbb{S}_{n,d},1+g_G(\epsilon n)>c(d,\delta)} \nu_G \sum_{A\subsetneq V} p_n^{|A|}(1-p_n)^{n-1-|A|} \sum_{j\in A^c} \frac{1}{n}\mathbb{E}[f(G, C'_{i,A,G})],$$

where $C'_{i,A,G}$ the number of messages generated by a DLS with

$$\mathrm{TTL}'_n = (d+1)(1+g_G(\epsilon n))^{\mathrm{TTL}_n - 1} - 1$$

and

$$f(G,C') = \frac{d^2(d-1)^{2-\log_{(1+g_G(\epsilon n))}(d+1)}}{d-2} \cdot (C'+1)^{\log_{(1+g_G(\epsilon n))}(d-1)} - \frac{d^2}{d-2}.$$

Since $1+g_G(\epsilon n) < d-1$,

$$\mathrm{TTL}'_n \leq (d+1)(d-1)^{\log_{(d-1)}(\epsilon n-1)+1-\log_{(d-1)} d} - 1 = \frac{(d+1)(d-1)}{d}\cdot(\epsilon n-1) - 1$$

and, for $1+g_G(\epsilon n) \geq c(d,\delta)$,

$$f(G,C') \leq \frac{d^2(d-1)^2}{d-2} \cdot 2^{\log_{c(d,\delta)}(d-1)} (C')^{\log_{c(d,\delta)}(d-1)} \tag{5.40}$$

This gives

$$W_n^2 \leq \frac{d^2(d-1)^2}{d-2}\cdot 2^{\alpha(d,\delta)} \sum_{G\in\mathbb{S}_{n,d},1+g_G(\epsilon n)>c(d,\delta)} \nu_G \sum_{A\subsetneq V} p_n^{|A|}(1-p_n)^{n-1-|A|} \sum_{j\in A^c} \frac{1}{n}(C''_{i,A,G})^{\alpha(d,\delta)},$$

where $C''_{i,A,G}$ the message cost under a DLS with stopping time

$$\xi_n \equiv \mathrm{TTL}''_n = \frac{(d+1)(d-1)}{d}\cdot(\epsilon n-1) - 1 = \Theta(n)$$

and

$$\alpha(d,\delta) \equiv \log_{c(d,\delta)}(d-1) = \log_{(d-1-\delta)}(d-1).$$

As in the proof of Lemma 5.8, it can be shown that

$$\sum_{A\subsetneq V} p_n^{|A|}(1-p_n)^{n-1-|A|} \sum_{j\in A^c} \frac{1}{n}(C''_{i,A,G})^{\alpha(d,\delta)} = \mathbb{E}[(C'_{i,A,G})^{\alpha(d,\delta)}]$$

where $C''_{i,A,G}$ a truncated geometric r.v. whose distribution is given by (5.31) with ξ_n given above. Note that the above quantity does not depend on G. Hence, Lemma 5.11 along with the bound (5.35) imply

$$W_n^2 \leq \frac{d^2(d-1)^2}{d-2}\cdot 2^{\alpha(d,\delta)}\phi_n \cdot \begin{cases} \left(\frac{1}{p_n}\right)^{\alpha(d,\delta)} \Gamma\left(1+\alpha(d,\delta)\right), & \text{if } p_n = \Omega\left(\frac{1}{\xi_n}\right) \\ \xi_n^{\alpha_{d,\delta}}, & \text{if } p_n = o\left(\frac{1}{\xi_n}\right) \end{cases} \tag{5.41}$$

From (5.38),(5.39),(5.41), and (5.37) we have that

$$\rho_n = O\left(np_n(1-\phi_n)\right) + \begin{cases} O\left(\left(\frac{1}{p_n}\right)^{\alpha(d,\delta)-1}\right), & \text{if } p_n = \Omega\left(\frac{1}{\xi_n}\right) \\ O\left(p_n\xi_n^{\alpha(d,\delta)}\right), & \text{if } p_n = o\left(\frac{1}{\xi_n}\right) \end{cases}$$
$$= O\left(n^{1-\frac{\delta}{2}}p_n\right) + \begin{cases} O\left(\left(\frac{1}{p_n}\right)^{\alpha(d,\delta)-1}\right), & \text{if } p_n = \Omega\left(\frac{1}{\xi_n}\right) \\ O\left(p_n\xi_n^{\alpha(d,\delta)}\right), & \text{if } p_n = o\left(\frac{1}{\xi_n}\right) \end{cases}$$

as

$$1-\phi_n = O\left(n^{-\frac{\delta}{2}}\right).$$

This proves the first statement of Theorem 5.6. The second statement follows from part (b) of Lemma 5.11. □

5.5.3 Server Load

A similar statement to Theorem 5.6 can be proved, under the same assumptions, for the server traffic load.

Theorem 5.7. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and vertex balanced, that its state space $\mathbb{S}_{n,d} \subseteq \mathbb{G}_{n,d}$, where $d \geq 3$, and that its stationary distribution is uniform over $\mathbb{S}_{n,d}$ and contiguous to the uniform distribution over $\mathbb{G}_{n,d}$. Then, for every $\delta > 0$ there exists a $\mathrm{TTL}_n = \Theta\left(\log_{(d-1)} n\right)$ such that the server traffic load under the expanding ring is*

$$\rho_n^0 = O\left(n^{1-\frac{\delta}{2}}p_n\right) + O\left(np_n(1-p_n)^{\xi_n}\right) \tag{5.42}$$

where

$$\xi_n = \Theta\left(n^{1/\alpha(d,\delta)}\right)$$

and

$$\alpha(d,\delta) = \log_{(d-1-\delta)}(d-1).$$

In particular, the stopping time TTL_n is the same as the TTL_n appearing in Theorem 5.6, for the same $\delta > 0$.

Proof. Again, we prove only the case where $\mathbb{S}_{n,d} = \mathbb{G}_{n,d}$, as the proof of the case where $\mathbb{S}_{n,d} \subset \mathbb{G}_{n,d}$ is almost identical: it too can be dealt with by using Corollary 3.11 instead of Theorem 3.15, wherever the latter appears below.

As in the proof of Theorem 5.6, let TTL_n be

$$\mathrm{TTL}_n = \log_{(d-1)}(\epsilon n - 1) + 2 - \log_{(d-1)} d. \tag{5.43}$$

where $\epsilon > 0$ is such that

$$\mathbf{P}\left(1 + g_{\mathcal{G}_{n,d}}(\epsilon n) \geq c(d,\delta)\right) \geq \phi_n. \tag{5.44}$$

where

$$c(d,\delta) = d - 1 - \delta$$

and

$$\phi_n = 1 - O\left(n^{-\frac{\delta}{2}}\right).$$

Such an $\epsilon > 0$ exists for every $\delta > 0$ by Theorem 3.15 in Chapter 3.

From Lemma 5.6, the average traffic load per peer is

$$\rho_n^0 = n\mu \cdot R_n \tag{5.45}$$

where R_n the steady state expectation

$$R_n = \lim_{t\to\infty} \mathbb{E}[R(t)] = \lim_{t\to\infty} \frac{1}{t}\sum_{s=0}^{t} R(s), \quad \textit{a.s.}$$

of the reward function

$$R(t) = \begin{cases} 0, & \text{if no request is issued at the } t\text{-th epoch,} \\ r_{i,A,G}, & \text{if a request is issued at the } t\text{-th epoch,} \end{cases}$$

and $r_{i,A,G}$ is one if the query propagation reaches the server and zero otherwise, given that, at the time the query is issued, the source peer is i, the set of positive peers is A and the overlay graph is G. Note that, for the expanding ring, $r_{i,A,G}$ is deterministic. From Lemma 5.4,

$$\begin{aligned} R_n &= p_n \cdot \sum_{G\in\mathbb{S}_{n,d}} \nu_G \sum_{A\subsetneq V} p_n^{|A|}(1-p_n)^{n-1-|A|} \sum_{j\in A^c} \frac{1}{n} r_{i,A,G} \\ &\overset{(5.44)}{\leq} p_n(1-\phi_n)\cdot 1 + p_n \cdot Z_n \end{aligned} \tag{5.46}$$

where

$$Z_n = \sum_{G\in\mathbb{S}_{n,d}, 1+g_G(\epsilon n)>c(d,\delta)} \nu_G \sum_{A\subsetneq V} p_n^{|A|}(1-p_n)^{n-1-|A|} \sum_{j\in A^c} \frac{1}{n} r_{i,A,G}.$$

By Lemma 5.9(a), we have that

$$r_{i,A,G} \leq r'_{i,A,G}$$

where $r'_{i,A,G}$ is one if a DLS with

$$\mathrm{TTL}'_n = (d+1)(1+g_G(\epsilon n))^{\mathrm{TTL}_n - 1} - 1$$

fails to locate the item within the peer-to-peer system and zero otherwise (given that, at the time the query is issued, the source peer is i, the set of positive peers is A and the overlay graph is G). Note that $r'_{i,A,G}$ too is deterministic. For $g'_G(\epsilon n) \geq c(d,\delta)$,

$$r'_{i,A,G} \leq r''_{i,A,G}$$

where $r''_{i,A,G}$ is one if a DLS with stopping time

$$\xi_n \equiv \mathrm{TTL}''_n = (d+1)c(d,\delta)^{\mathrm{TTL}_n - 1} - 1 \leq \mathrm{TTL}'_n$$

fails to locate the item within the peer-to-peer system and zero otherwise. Hence,

$$Z_n \leq \sum_{G \in \mathbb{S}_{n,d}, 1+g_G(\epsilon n) > c(d,\delta)} \nu_G \sum_{A \subsetneq V} p_n^{|A|}(1-p_n)^{n-1-|A|} \sum_{j \in A^c} \frac{1}{n} r''_{i,A,G},$$

As in the proof of Lemma 5.8, it can be shown that

$$\sum_{A \subsetneq V} p_n^{|A|}(1-p_n)^{n-1-|A|} \sum_{j \in A^c} \frac{1}{n} r''_{i,A,G} = (1-p_n)^{\xi_n}$$

Note that the above quantity does not depend on G. Hence, by (5.44),

$$Z_n \leq \phi_n \cdot (1-p_n)^{\xi_n} \tag{5.47}$$

From (5.45),(5.46), and (5.47), we have that

$$\rho_n^0 = O\left(np_n(1-\phi_n)\right) + O\left(np_n(1-p_n)^{\xi_n}\right).$$

Theorem 5.7 follows by observing that

$$1 - \phi_n = O\left(n^{-\frac{\delta}{2}}\right)$$

and

$$\begin{aligned}\xi_n &= (d+1)c(d,\delta)^{\log_{(d-1)}(\epsilon n - 1) + 1 - \log_{(d-1)} d} - 1 \\ &= (d+1)c(d,\delta)^{1-\log_{(d-1)} d} \cdot (\epsilon n - 1)^{\log_{(d-1)} c(d,\delta)} = \Theta\left(n^{1/\alpha(d,\delta)}\right),\end{aligned}$$

where

$$\alpha(d,\delta) = \log_{(d-1-\delta)}(d-1). \quad \square$$

5.5.4 Proof of Theorem 5.2

We are now ready to prove Theorem 5.2, our main result for the expanding ring mechanism. To begin with, Theorem 5.6 immediately implies the following asymptotic upper bound on the average traffic load per peer, which is the first statement in Theorem 5.2.

Corollary 5.4. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and vertex balanced, that its state space is $\mathbb{S}_{n,d} \subseteq \mathbb{G}_{n,d}$, where $d \geq 3$, and that its stationary distribution is uniform over $\mathbb{S}_{n,d}$ and contiguous to the uniform distribution over $\mathbb{G}_{n,d}$. Then, there exists a $\mathrm{TTL}_n = \Theta\left(\log_{(d-1)} n\right)$ such that the expanding ring has*

$$\rho_n = O\left(n^{\frac{\log(d-1)}{\log(d-3)}-1}\right) \tag{5.48}$$

irrespectively of p_n.

Proof. The corollary follows from Theorem 5.6 by taking $\delta = 2$. In particular, for $\delta = 2$,

$$n^{1-\frac{\delta}{2}} p_n = O\left(1\right).$$

Hence, we can focus on the second term of (5.30) of Theorem 5.6. The worst case upper-bound on this term, over all p_n, is for $p_n = \Theta\left(\frac{1}{n}\right)$. In this case the load is $\rho_n = O\left(n^{\alpha(d,\delta)-1}\right)$ and the corollary follows as $\alpha(d,\delta) = \log_{(d-1-\delta)}(d-1)$. □

We note again that, as discussed in Section 5.2.2, the above bound on ρ_n grows very slowly in n. The second statement in Theorem 5.2 follows similarly, as a corollary of Theorem 5.7.

Corollary 5.5. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and vertex balanced, that its state space is $\mathbb{S}_{n,d} \subseteq \mathbb{G}_{n,d}$, where $d \geq 3$, and that its stationary distribution is uniform over $\mathbb{S}_{n,d}$ and contiguous to the uniform distribution over $\mathbb{G}_{n,d}$. Then, there exists a $\mathrm{TTL}_n = \Theta\left(\log_{(d-1)} n\right)$ such that the expanding ring has*

$$\rho_n^0 = O\left(n^{1-\frac{\log(d-3)}{\log(d-1)}}\right) \tag{5.49}$$

irrespectively of p_n. In particular, the stopping time TTL_n is the same as the TTL_n appearing in Corollary 5.4, for the same $\delta > 0$.

Proof. Again, by taking $\delta = 2$, the first term in (5.42) is $O\left(1\right)$ and, hence, we can focus on the second term. The worst case over all p_n is for $p_n = \Theta\left(\frac{1}{\xi_n}\right)$, which gives

$$\rho_n^0 = O\left(n\frac{1}{\xi_n}(1-\frac{1}{\xi_n})^{\xi_n}\right) = O\left(\frac{n}{\xi_n}\right) = \Theta\left(n^{1-\frac{1}{\alpha(d,\delta)}}\right).$$

and the corollary follows. □

Again, as discussed in Section 5.2.2, the above bound also grows very slowly in n. Theorem 5.2 follows immediately from Corollaries 5.4 and 5.5. Moreover, it is easy to bound the query response time by observing that it can be no more than

$$\delta \cdot \frac{\mathrm{TTL}_n(\mathrm{TTL}_n + 1)}{2}.$$

In particular, the following lemma holds.

Lemma 5.12. *The (worst-case) response time of a query propagated by an expanding ring is* $O\left(\mathrm{TTL}_n^2\right)$. □

Comparing the expanding ring with $\mathrm{TTL}_n = \Theta\left(\log_{d-1}(n)\right)$ and the random walk with $\mathrm{TTL}_n = \Theta\left(n\right)$, we see that the response time of the expanding ring mechanism is of a considerably smaller order, while giving traffic loads that grow very slowly in n. Hence, the expanding ring mechanism considerably reduces the response time compared to the random walk mechanism, without jeopardizing the scalability of the system.

5.6 Numerical Study

To illustrate the validity of our analytical results, we conducted a numerical study in which we relaxed several of our modelling assumptions. First, instead of assuming that the system size is fixed, in our simulations we let it vary as time evolves. Second, instead of assuming that the overlay graph is static during query propagation, our simulated system can change while a query is being propagated. Finally, we allow queries to take place at times chosen uniformly at random within a peer's lifetime, as opposed to when a peer initially enters the system.

We restricted our numerical evaluation to overlay graphs that are expanders, motivated by the fact that unstructured peer-to-peer networks have this property, as discussed in Section 4.2.2. To create an overlay graph that is an expander *w.h.p.*, we use the Law and Siu connection protocol [LS03] that was presented in detail in Section 4.3.1.

Even though we considerably relaxed several modelling assumptions of Section 4, our analytical results of Sections 5.4 and 5.5 predict remarkably well the behaviour of the simulated system. Overall, the simulation results suggest that our analytical model indeed captures the important features of hybrid peer-to-peer systems, and that, despite the simplifying assumptions made, the correct insight on the scalability of such systems is gained. In particular, the simulations confirm our main result that a system with a random walk or an expanding ring has excellent scalability properties, in terms of the traffic incurred at both the server and at individual peers.

5.6.1 Simulation Setup

As mentioned above, in our simulations we let the number of peers vary over time. In particular, we let peers arrive according to a Poisson process with rate λ and stay in the system for exponentially distributed times with mean $1/\mu$ equal to 20 minutes. To scale the system, we repeated our simulations with different arrival rates λ, ranging between $1,000 \times \mu$ and $500,000 \times \mu$. As a result, the expected number of peers in the system in each of our experiments, given by $n = \lambda/\mu$, scales between a thousand and half a million nodes.

We let peers join and leave the peer-to-peer network according to the Law and Siu connection protocol [LS03] which was outlined in detail in Section 4.3.1. Recall that this protocol leads to a graph that is an expander with high probability, and is easy to implement in a distributed manner. We use a degree $d = 16$ in the simulations presented here, although also conducted experiments with values as low as $d = 4$ and obtained similar results.

Finally, in addition to the case where an arriving peer requests the data item immediately upon its arrival, we also consider the case where requests are issued at a time uniformly chosen among a peer's lifetime. In either case, requests occur with probability p_n. We repeated our simulations for different request probabilities p_n given by $p_n = 0.5$, $p_n = 0.5\log(1000)/\log(n)$, $p_n = 0.5\sqrt{1000/n}$, $p_n = 0.5(1000/n)^{1/3}$, $p_n = 0.5 \cdot 1000/n$, $p_n = 0.5(1000)^2/n^2$, $p_n = 0.5(1000)^{2.5}/n^{2.5}$ and $p_n = (1000)^3/n^3$. We start each simulation in steady state: if the expected number of peers is n and the request probability is p_n, we start with a system populated with n peers, each one storing a copy of the data item with probability p_n. We observe the system for 20 million arrivals and measure the traffic load at the server, the traffic load at peers and the query response time.

5.6.2 Random Walk Mechanism

We first illustrate the results obtained in Section 5.4 for the random walk query propagation mechanism. Queries in our simulations are propagated according to a random walk, where the one-hop transmission delay of a query is exponentially distributed with mean $\delta = 20$ milliseconds. Queries are redirected to the server after a time period of $\delta \mathrm{TTL}_n$, where $\mathrm{TTL}_n = n$ (and n the expected number of peers in the system).

Our analysis in Section 5.4 suggests that, if TTL_n is linear and the overlay graph is an expander, then queries can be grouped in to two categories: frequent queries, for popular items that have request probabilities $p_n = \Omega(1/n)$, and infrequent queries, for unpopular items that have request probabilities $p_n = o(1/n)$. We first present the results for frequent queries.

(a) Average Load per Peer ρ_n (b) Response Time D_n

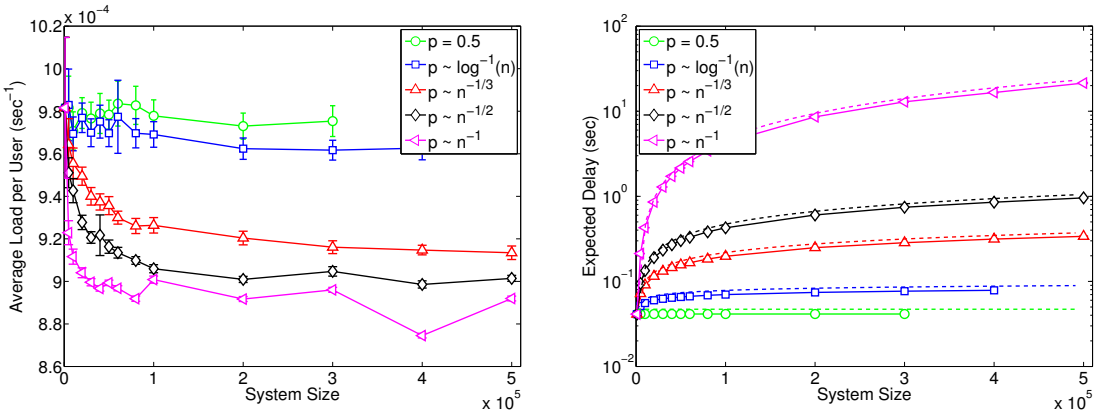


Figure 5.3: The average traffic load per peer and the query response time for frequent queries, under a delay-constrained random walk with $\mathrm{TTL}_n = n$. The average load per peer is plotted with 98% confidence intervals, which were too small to plot for the query response time. In all cases, the traffic load per peer is bounded, as predicted by Theorem 5.4. The dashed lines next to each query response time plot show the upper bound on the query response time, as predicted by Lemma 5.7.

Figure 5.3(a) shows the average traffic load ρ_n per peer, with 98% confidence intervals, for the request probabilities p_n that decrease no faster than $1/n$. The observed values of ρ confirm Theorem 5.4 of Section 5.4: the traffic loads generated for all request probabilities $p_n = \Omega(n)$ are constant, close to one query every thousand seconds. This is indeed the behaviour predicted for frequent queries.

In Figure 5.3(b), we plot the query response times for frequent queries. The 98% confidence intervals are too small to be displayed on the same graph as the observed values. We also plot with dashed lines approximations of the respective bounds on the response time obtained by Lemma 5.7. To compute these bounds, we use the results from Figure 5.3(a) to estimate the relaxation time of the Law and Siu network. In particular, from Theorem 5.4, we have that $\rho_n \approx \bar{\tau}\mu$ for high request probabilities. Hence, we can use the observed value of ρ_n in Figure 5.3(a) to estimate the relaxation time as $\bar{\tau} = \rho/\mu$. We estimated it to be $\bar{\tau} = 1.176$ and then used this value to approximate the bound on the response time, as expressed by Lemma 5.7. These approximate bounds correctly predict the response times both qualitatively and quantitatively; in particular, the delays grow as $1/p_n$. This indicates that our model predicts system behaviour quite accurately, in spite of the simplifying assumptions it employs.

For all $p_n = \Omega(1/n)$ and throughout the 20 million departure events of our simulations,

we observed zero traffic at the server. This might seem surprising at first, however this is also consistent with our theoretical analysis. Using $\tau = 1.176$, Theorem 5.4 gives that, for queries with request probabilities $p_n = \Omega(1/n)$, the server traffic load is less of 10^{-120} queries per second. Such a traffic load is too small to be observed from our measurements.

Figure 5.4 shows the average traffic load per peer, the query response time and the server load for infrequent queries, *i.e.*, queries for items request probabilities p_n that decay faster than $1/n$. In each of the above figures, we also plot with dashed lines the theoretical bounds in Section 5.4 obtained by applying Theorem 5.4, Theorem 5.5 and Lemma 5.7, respectively, with $\tau = 1.176$. For Figure 5.4(a), the 98% confidence intervals are too small to be plotted along with the observed values. The theoretical bounds for the age for $p_n = \Theta(1/n^{2.5})$ and $p_n = \Theta(1/n^3)$ are too close to TTL_n to be distinguishable, so only the latter appears in Figure 5.4. We note that, contrary to the frequent query regime, we do observe a traffic load on the server. In all cases, our theoretical analysis correctly predicts the behaviour of all three metrics, both qualitatively and quantitatively.

In our model, we assumed that the network is effectively static during query propagation. As for infrequent queries the delays are quite large, this modelling assumption is clearly violated in our simulations. However, as we see from Figure 5.4, our analytical results still correctly predict all three measured quantities, both qualitatively and quantitatively. This suggests that the model used for our analysis indeed captures the important features of the system and provides correct insight on its behaviour. As predicted by the analytical results, the above simulations show that (a) the system scales well in terms of both the server and peer traffic loads and that (b) the query response times can be large.

Next, we investigated the case in which the queries where not issued upon the arrival of a new peer in the system, but at a time uniformly chosen among the peer's lifetime. Comparing Figure 5.3(a) to Figure 5.5(a), we see that the traffic load is again bounded by a constant (close to 1 query every 580 seconds), though it has increased roughly by a factor of two. Intuitively, obtaining the data item at a time uniformly chosen from within a peer's lifetime reduces the expected time that a peer shares the item to 10 minutes, *i.e.*, half the peer lifetime (20 minutes), with a corresponding increase in the experienced delay and the traffic loads.

In Figure 5.5(b) we plot the observed delays along with the theoretical bounds from Lemma 5.7 for a lifetime of 10 minutes and $\tau = 1.176$. Similarly, in Figure 5.6 we plot the average traffic load per peer, the server load, and the query response time for infrequent queries. Again, we also plot the theoretical upper bounds for a lifetime of 10 minutes and $\tau = 1.176$, as predicted by Theorem 5.4, Theorem 5.5 and Lemma 5.7, respectively. The results further attest

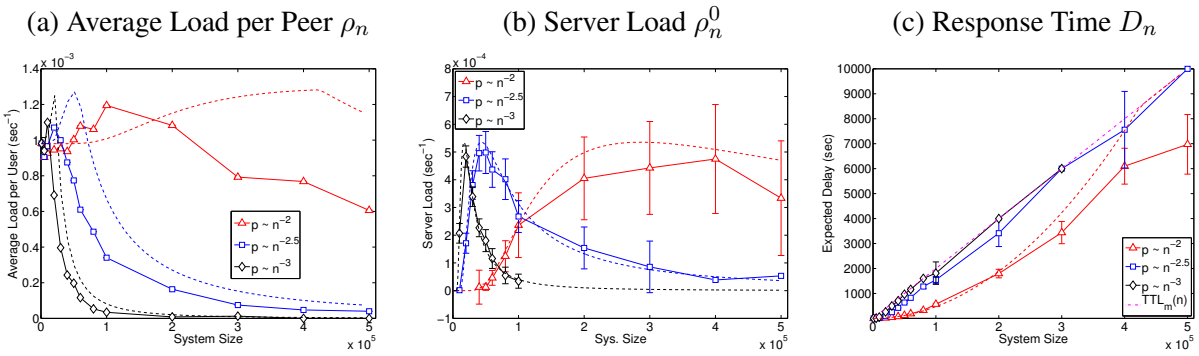


Figure 5.4: The average traffic load per peer, the server load, and the query response time for infrequent queries, under a delay-constrained random walk with $\mathrm{TTL}_n = n$. The server load and the response time are plotted with 98% confidence intervals, which were too small to plot for average load per peer. In all three cases, we also plot the theoretical upper bounds, as predicted by Theorems 5.4, 5.5, and Lemma 5.7, respectively. In all three cases, our simulations agree very closely with the results predicted by our model.

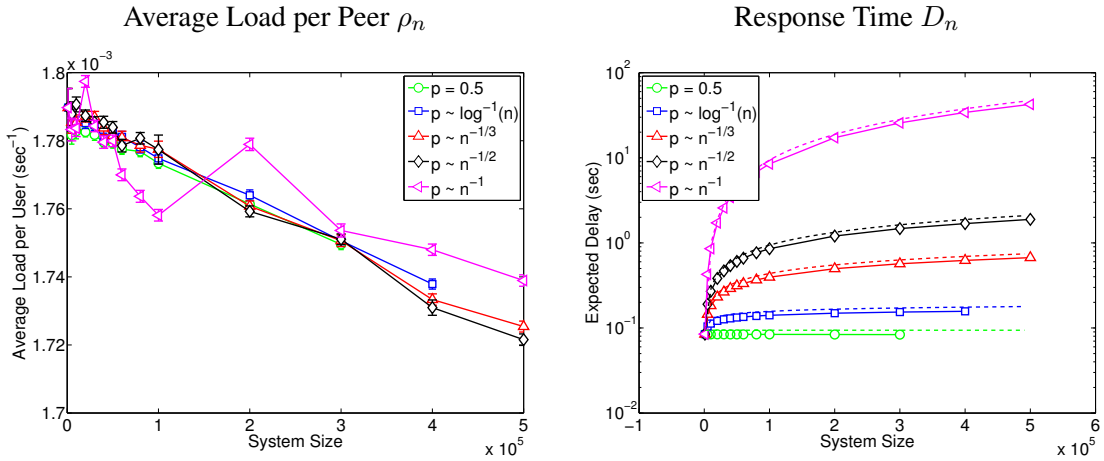


Figure 5.5: The average load per peer and the response time for frequent queries occurring at a time chosen uniformly within a peer's lifetime, with 98% confidence intervals. In (a), the traffic loads observed are again bounded by a constant, though they are twice higher than the corresponding loads for queries issued at the beginning of peer lifetimes (*c.f.* Fig. 5.3). The dashed lines in (b) are the response times predicted by our model for a peer lifetime equal to 10 minutes, *i.e.*, half the actual lifetime of peers.

to the fact that the system has a behaviour no worse that the one represented in our model, where the mean lifetime has been reduced to 10 minutes. We note that, in these simulations too, we did not observe any traffic load at the server when queries were frequent.

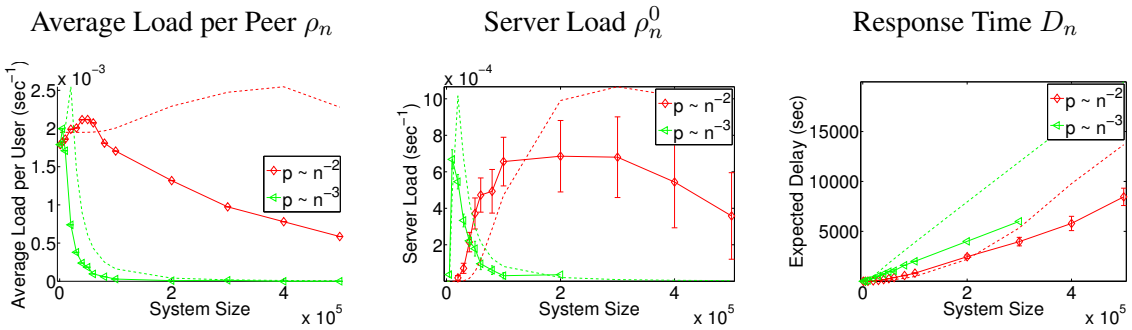


Figure 5.6: The average traffic load per peer, the server load, and the query response time for infrequent queries, occurring at a time chosen uniformly within a peer's lifetime, with 98% confidence intervals. The dashed lines correspond to the same metrics calculated under our model for a peer lifetime equal to 10 minutes, *i.e.*, half the actual lifetime of peers.

5.6.3 Expanding Ring Mechanism

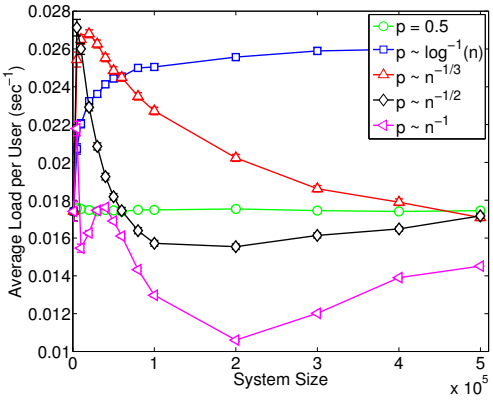
Next, we investigate numerically the expanding ring query propagation mechanism. We repeat the above experiments for the expanding ring with $\mathrm{TTL}_n = \delta \log_{16} n$.

Figures 5.7(a) and 5.7(b) present the average traffic load per peer for frequent and infrequent queries, respectively. Comparing Figures 5.3(a) and 5.7(a), we see that the peer loads are 16 to 20 times larger than the ones observed under the random walk. This is not surprising, as the expanding ring propagates messages more aggressively.

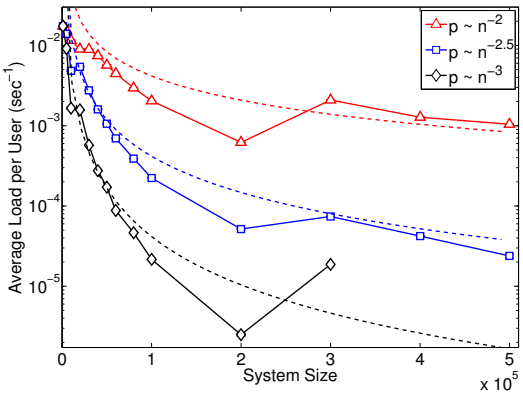
Theorem 5.6 bounds for the average load at peers ρ by a quantity that grows as a (fractional) polynomial of n. We do not plot these theoretical bounds as the constants involved in (5.30) are quite loose. Nonetheless, although the asymptotic behaviour of the observed loads in Figure 5.7(a) is not immediately clear, all but one of these loads do not grow in in n —in fact, most have an initial decaying behaviour, similar to the one observed for the random walk in Figure 5.3(a). Moreover, the only load that appears to be increasing (for $p_n = \Theta\left(\frac{1}{\log n}\right)$) grows very slowly. All in all, the simulation results shown in Figure 5.7(a) suggest that the traffic load per peer for frequent queries is either bounded or growing slowly; this agrees with our conclusion that the expanding ring is scalable for $p_n = \Omega\left(1/n\right)$.

For request probabilities $p = o\left(1/n\right)$, our model correctly predicts the asymptotic behaviour of the load on the peers and the load on the server, as seen in Figures 5.7(b) and 5.7(c). However, we note that we overestimate the behaviour for low values of n and that our bounds are not as close as the ones obtained for the same probabilities under the random walk (Figures 5.4(a) and 5.4(b), respectively).

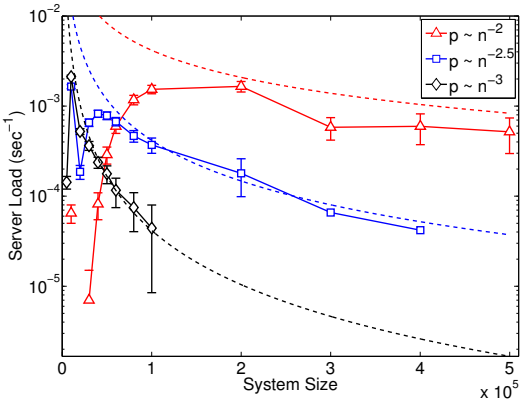
(a) Average Load per Peer ρ_n, $p_n = \Omega\left(\frac{1}{n}\right)$



(b) Average Load per Peer, $p_n = o\left(\frac{1}{n}\right)$



(c) Server Load ρ_n^0, $p_n = o\left(\frac{1}{n}\right)$



(d) Response Time D_n

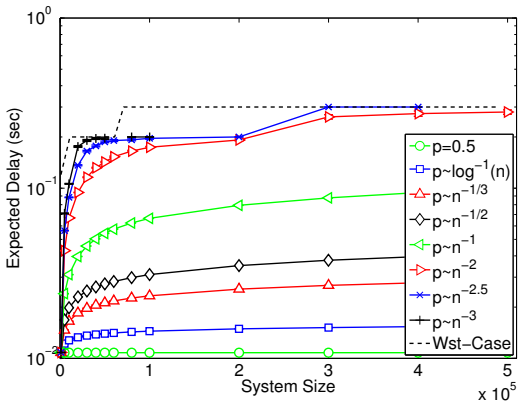


Figure 5.7: Performance metrics for the expanding ring. Figures (a) and (b) show the average load per peer for frequent and infrequent queries respectively. Figure (c) shows the server load for infrequent queries —no server traffic was observed for frequent queries. Figure (d) shows the response times for both frequent and infrequent queries. In (b)-(d), the theoretical bounds are also plotted with dashed lines.

As expected, the expanding ring considerably outperforms the random walk in terms of query response times. Fig. 5.7 shows the observed response times as well as the worst-case delay of Lemma 5.12. Response times are considerably smaller (by 2 to 5 orders of magnitude) than the respective ones under the random walk (Figures 5.3 and 5.4). In addition, the response times are close to the worst-case bound of Lemma 5.12 only for $p = o\left(1/n\right)$.

5.7 Extensions and Open Questions

5.7.1 General Overlay Topologies

Our two main results, namely, Theorem 5.1 and Theorem 5.2, require that the overlay graph be uniformly distributed (in steady state) over a large enough subset of regular graphs and, as a result, be an expander *a.a.s.* As discussed in Section 4.2.2, most connection protocols yielding a rich enough family of regular overlay graphs will have this property, and several examples have already been presented in Section 4.3. Moreover, overlay graphs not having this property, *e.g.*, because their distribution is concentrated over a small subset of regular graphs, exhibit a certain structure. Therefore, this structure should be exploited by the query propagation mechanism, and a mechanism that does so would be preferable to the random walk or the expanding ring.

Nonetheless, it is interesting to understand how the random walk and the expanding ring would behave if the above assumption did not hold and the overlay graph was not an expander. Our analysis can immediately be extended to this case as well.

In particular, consider an ergodic, vertex balanced, churn-driven Markov chain $\{G(t)\}_{t\in\mathbb{N}}$, whose stationary probability is such that

$$\mathbf{P}(\tau_n \leq f_n) = 1 - o\left(1/n\right),$$

for some positive sequence f_n with

$$\limsup f_n = +\infty.$$

Then, our entire analysis on the delay-constrained random walk mechanism extends to this case. In particular, Theorems 5.4 and 5.5, describing ρ_n and ρ_n^0, respectively, hold if $\bar{\tau}$ replaced by f_n.

Interestingly, the above results imply that, if f_n is *not* bounded, we cannot simultaneously bound both ρ_n and ρ_n^0. The following theorem is the analogue of Theorem 5.1:

Theorem 5.8. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and vertex balanced, and that there exists a sequence f_n such that*

$$\mathbf{P}(\tau_n \leq f_n) = 1 - o\left(\frac{1}{n}\right),$$

where τ_n the relaxation time of a graph sampled from the stationary distribution of $\{G(t)\}_{t\in\mathbb{N}}$. Then, for a delay-constrained random walk with $\mathrm{TTL}_n = \Theta\left(n\right)$,

$$\rho_n = O\left(\min(f_n, n)\right)$$

and

$$\rho_n^0 = O\left(\min(f_n, n)\right),$$

irrespectively of p_n.

In other words, for $\mathrm{TTL}_n = \Theta(n)$, the worst-case bounds (over all p_n) we obtain for both ρ_n^0 and ρ_n grow with n. A coupling argument can be used to show that smaller values of TTL_n can only increase the worst-case bound on ρ_n^0, while larger values of TTL_n can only increase the worst-case bound on ρ_n. As a result, if the overlay graph is not an expander *w.h.p.*, there is no TTL_n for which ρ_n and ρ_n^0 are both bounded, irrespectively of p_n.

We can also extend our analysis of the expanding ring. Assuming, as above, that the relaxation time can be bounded by a sequence f_n *w.h.p.*, Kahale's Theorem (Theorem 3.14) yields a bound on the expansion of small sets, also *w.h.p.*. As in Corollary 3.10, the bound will be on the modified vertex expansion $g'_G(\epsilon n)$, rather than $g_G(\epsilon n)$. Our analysis on the expanding ring can be applied as is —note that Lemma 5.9(c) holds if $1 + g_G(\epsilon n)$ is replaced by $g'_G(\epsilon n)$. The following theorem is the analogue of Theorem 5.2:

Theorem 5.9. *Assume that* $\{G(t)\}_{t\in\mathbb{N}}$ *is ergodic and vertex balanced, that its state space is* $\mathbb{S}_{n,d} \subseteq \mathbb{G}_{n,d}$ *and that there exists a sequence* f_n *such that*

$$\mathbf{P}(\tau_n \leq f_n) = 1 - o\left(\frac{1}{n}\right),$$

where τ_n *the relaxation time of a graph sampled from the stationary distribution of* $\{G(t)\}_{t\in\mathbb{N}}$. *Then, for every* $\delta > 0$ *there exists a* $\mathrm{TTL}_n = \Theta\left(\log_{(d-1)} n\right)$ *such that the expanding ring has*

$$\rho_n = O\left(n^{\beta(d,\delta)-1}\right)$$

and

$$\rho_n^0 = O\left(n^{1-\frac{1}{\beta(d,\delta)}}\right),$$

irrespectively of p_n*, where*

$$\beta(d,\delta) = \frac{\log(d-1)}{\log\left[\frac{d}{2}\left(1 - \sqrt{1 - 4\frac{d-1}{d^2\left(1-\frac{1}{f_n}\right)}}\right)\cdot(1-\delta)\right]}.$$

There are two important things to notice about Theorem 5.9. First, it allows us to extend the result of Theorem 5.2 not only to general graphs, but also to expander topologies whose stationary distribution is not necessarily contiguous to the uniform distribution over $\mathbb{G}_{n,d}$, as required by Theorem 5.2.

The second important observation is that, even if $\{G(t)\}_{t\in\mathbb{N}}$ is an expander with high probability, Theorem 5.9 will give looser bounds than the ones appearing in Theorem 5.2. The reason is that, by definition, $\beta(d,\delta) > \log_{d/2}(d-1)$. Hence, using Theorem 5.9, we cannot achieve exponents as small as the ones of Theorem 5.2. Nonetheless, if the overlay graph is a "good" expander, and $\beta(d,\delta) \approx \log_{d/2}(d-1)$, the exponents of the bounds in Theorem 5.9 decrease in d, and can therefore be made arbitrarily small by the system designer.

Intuitively, in the case of expanders, Theorem 3.15 suggests that the modified vertex expansion ratio for small sets can be brought arbitrarily close to $d-1$, by potentially paying a price in terms the rate of the *a.a.s.* convergence (*i.e.*, in the rate with which $\phi_n = \mathbf{P}(\tau_n \leq f_n)$ converges to zero). This suggests that the bounds of Theorem 5.9, obtained by applying Kahale's Theorem, are not tight and can be further improved.

5.7.2 General Query Propagation Mechanisms

A rather obvious, and quite interesting, open area of research is extending the above results to other query propagation mechanisms, such as k-parallel random walks [LCC+02, AAK+08], non-backtracking random walks [ABLS07], budget-based forwarding [TKLB07, GMS05], *etc.* In all of these cases Lemmas 5.5 and 5.6 can be used to reduce the computation of the loads ρ_n and ρ_n^0 to steady state rewards over the Markov chain $\{G(t)\}_{t\in\mathbb{N}}$. From a system designer's perspective, the fact that the performance of hybrid systems is already good under the expanding ring and the random walk mechanisms is very promising. One expects that the performance of the system would improve under the above (more sophisticated) mechanisms.

The analysis of DLS extends to any query propagation mechanism that is *one-pass*, *i.e.*, visits every vertex only once. In particular, Lemma 5.8 holds for all such mechanisms, including, *e.g.*, depth first search (DFS) [Wes01]. This suggests that a DFS over a d-regular graph with stopping time $\mathrm{TTL}_n = \Theta(n)$ will also yield a bounded traffic load on both the server and peers. DFS can be implemented in a distributed way simply by "backtracking"; this will guarantee that each edge is not visited more than twice, or, that each peer visited will exchange no more than $2d$ messages.

Lemma 5.8 implies that the above implementation of a DFS on a regular graph incurs traffic loads of the same order as the random walk on an expander overlay. However, there are differences in the constants involved: The average traffic load per peer incurred by DFS will be roughly $2d$ times the traffic load incurred by the random walk. Moreover, the DFS needs to maintain state information, which makes it more vulnerable to graph changes during query

propagation. In short, if the overlay graph is an expander *w.h.p.*, the random walk can achieve a much better traffic load than the DFS, without maintaining any state information.

Investigating the performance of a system that incorporates non-passive replication is an open problem. An interesting case is combining a random walk with *path-replication*, as introduced by Lv *et al.* [LCC+02], and discussed in Section 2.3.2. In such a random walk, each peer forwarding a query message obtains a copy of the data item, as opposed to only the source peer. A simple coupling argument can be used to show that path replication increases the availability of a item, compared to the simple random walk mechanism, while the incurred traffic can be no more than twice the traffic generated by the simple random walk mechanism. This immediately implies that the random walk with path replication will generate both a bounded average traffic load per peer and a bounded server traffic load.

As discussed in Section 2.3.2, by approximating the random walk with uniform sampling, Cohen and Shenker [CS02] use an operating point argument to claim that path-replication leads to a replication rate of the data item that is optimal, with respect to the average (over all items) query response time of the random walk. It would be extremely interesting both from a practical and a theoretical perspective, to extend the above results in the context of our model.

Peers may not be willing to store data items that they have not requested; Moreover, assuming that data items are far larger (*e.g.*, in bytes) than queries, the bandwidth required for implementing path replication may be prohibitive. An alternative is to allow peers forwarding a query to store indexing information, in the form of a pointer to either the peer that requested the item or the peer providing it. Such indexes may be used to shorten subsequent queries for the item received by the peer. Such a system would behave differently than a system with path replication, as indexes may become stale, upon the departure of the peer storing the item. This would also be quite interesting to analyze, both from a practical and theoretical perspective.

Finally, the model we proposed can be adapted to address searching over a hierarchical, two-tier peer-to-peer system. In particular, our overlay graph can be seen as the top-tier overlay of such a system. To fully capture a two-tier system's behaviour, we can extend our model so that (a) the contents of a (super)-peer correspond to an index of the contents of its leaf peers, and (b) requests originate from the leaf-peers. These lead to the following two modifications of our current model. The first issue requires modelling leaf-peer departures with self-loops in $\{G(t)\}_{t\in\mathbb{N}}$, that leave the graph structured unchanged, while (potentially) altering the contents indexed by the super-peer. The second issue may result to queries that are served locally (*i.e.*, not propagated over the system), when the content is stored at other leaf-peers attached to a super peer. Given that the above self-loops do not affect the ergodicity, balance or steady state

properties of the chain $\{G(t)\}_{t\in\mathbb{N}}$, and that the number of leaf-peers per super-peer is constant, the above modifications should not have a significant impact on the asymptotic performance of our system.

5.7.3 Optimal Query Propagation Mechanisms

Assume $\{G(t)\}_{t\in\mathbb{N}}$ is an ergodic churn-driven Markov chain, whose distribution is not necessarily contiguous to the uniform distribution over $\mathbb{G}_{n,d}$. Given two query propagation mechanisms $\mathcal{A}$ and $\hat{\mathcal{A}}$, let ρ_n and $\hat{\rho}_n$ be the average traffic load per peer under each mechanism for a given request probability p_n. Similarly, let ρ_n^0, $\hat{\rho}_n^0$ be the server traffic load under each of the two mechanisms, respectively. We can then define a partial ordering between query propagation mechanisms in terms of their scalability. In particular, we write $\mathcal{A} \preceq_\rho \hat{\mathcal{A}}$ if, for any p_n,

$$\text{if } \hat{\rho}_n = O(1), \text{ then } \rho_n = O(1).$$

Similarly, we write $\mathcal{A} \preceq_{\rho^0} \hat{\mathcal{A}}$ if, for any p_n,

$$\text{if } \hat{\rho}_n^0 = O(1), \text{ then } \rho_n^0 = O(1).$$

Finally, we write $\mathcal{A} \preceq \hat{\mathcal{A}}$ if $\mathcal{A} \preceq_\rho \hat{\mathcal{A}}$ and $\mathcal{A} \preceq_{\rho^0} \hat{\mathcal{A}}$. Given an overlay graph model $\{G(t)\}_{t\in\mathbb{N}}$, we say that a query propagation mechanism $\mathcal{A}$ is *optimal* if, for every other query propagation mechanism $\mathcal{B}$,

$$\mathcal{A} \preceq \mathcal{B}.$$

Theorem 5.1 immediately implies the following corollary.

Corollary 5.6. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and vertex balanced, and that there exists a constant $\bar{\tau}$ such that*

$$\mathbf{P}(\tau_n \leq \bar{\tau}) = 1 - o\left(\frac{1}{n}\right),$$

where τ_n the relaxation time of a graph sampled from the stationary distribution of $\{G(t)\}_{t\in\mathbb{N}}$. Then, the delay-constrained random walk with $\mathrm{TTL}_n = \Theta(n)$ is an optimal query propagation mechanism.

The results in Section 5.7.1 suggest that, if the overlay graph is not an expander *w.h.p.*, the random walk may not be optimal. Again, this is not surprising, as a query propagation mechanism over a structured system should exploit the structure of the overlay graph. An interesting open problem is therefore the following:

Optimality in a Hybrid System. Given an ergodic, churn-driven Markov chain $\{G(t)\}_{t\in\mathbb{N}}$, find an optimal query propagation mechanism (if one exists).

We note that, in practise, the overlay graph is determined through a connection protocol followed by peers. Therefore, we first need to determine the stationary distribution of the overlay graph, resulting from the connection protocol, before attempting to answer the above question.

5.7.4 Unbounded Traffic Loads

Our focus so far has been on the conditions under which both the traffic load at peers and the server load are bounded. In practise, cases in which ρ_n or ρ_n^0 grow slowly in n are also of interest. For example, due to technological advancements, peers may be able to afford faster connections over the long-term period of time during which the system size increases. On the server side, a slowly growing load indicates, *e.g.*, that the company maintaining it can invest in upgrading its infrastructure at a slow pace.

Lemma 5.7 and Theorems 5.4 and 5.5 describe the query response time and the peer and server loads, respectively, for all possible request probabilities, stopping times and topologies (the latter captured by the relaxation time τ_n and discussed in Section 5.7.1). Because of this, our results can be generalized to discuss cases under which the above traffic loads are not necessarily bounded.

In this section, we illustrate this by studying the effect of sub-linear stopping times on the system. For simplicity, we focus on the case where the overlay graph is an expander, although the discussion below can easily be extended to the case where τ_n is not bounded *w.h.p.* Intuitively, reducing TTL_n to a sub-linear value has the effect of increasing the proportion of queries that reach the server, while reducing the load at peers as well as the worst case delay they experience. Understanding such a trade-off is interesting because, on one hand, the server may be able to tolerate unbounded traffic (as discussed above) and, on the other hand, peers can benefit from reductions on their loads and their response times.

Theorem 5.4 suggests that the worst-case load at the server, over all request probabilities p_n, is for $p_n = \Theta\left(1/\mathrm{TTL}_n\right)$. In this case, the traffic load is $\rho_n^0 = \Theta\left(n/\mathrm{TTL}_n\right)$. On the other hand, for $\mathrm{TTL}_n = O\left(n\right)$ queries with $p_n = \Omega\left(1/\mathrm{TTL}_n\right)$, *i.e.*, frequent queries, generate a bounded average load per peer, while queries with $p_n = o\left(1/\mathrm{TTL}_n\right)$, *i.e.*, infrequent queries, generate a decreasing load per peer ($\rho_n = o\left(1\right)$).

Therefore, reducing TTL_n has the effect of increasing the queries for which the load per peer will decrease as n increases. In general, for a given sub-linear TTL_n, all items with a

popularity less than $1/\mathrm{TTL}_n$ will lead to (a) a decreasing load per peer as the peer population grows and (b) to a server load that scales as n/TTL_n in the worst case over all probabilities p_n. This result is important as it implies one can achieve a trade-off between the load at the server and at individual peers by appropriately choosing TTL_n.

To illustrate the above, we consider two numerical examples.

1. If TTL_n is constant, and the query is redirected to the server after a number of steps that does not depend on the total number of peers, the server load will grow as np_n: a large fraction of queries will therefore be directed to the server as the number of peers increases.

2. Suppose that, for all items, the item popularity p_n is a fractional power of of n, *i.e.*, $p_n = 1/n^c$ for some $0 < c < 1$, and thus the expected number of peers requesting a given item (given by np_n) grows slower than linearly in n. For this case, the load at individual peers can decrease as the peer population grows if TTL_n is chosen appropriately. For example, if $\mathrm{TTL}_n = n^{c'}$, $0 < c' \leq c$, the load per peer will be decreasing for all items such that $p_n = o(1/n^{c'})$. Moreover, the worst-case response time will be no worse than $n^{c'}$. However, this comes at the cost of incurring a load at the server which scales as $n^{1-c'}$.

5.7.5 Non-Zero Publishing Probabilities

Our analysis extends immediately to the case where $q_n > 0$, *i.e.*, peers publish the data item in the system. It is easy to show that a system with $q_n > 0$ will always have better performance than a system with $q_n = 0$.

Lemma 5.13. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic. Given p_n,q_n such that*

$$0 < p_n < 1, \quad 0 < p_n + q_n < 1, \qquad \textit{for all } n \in \mathbb{N},$$

let ρ_n, ρ_n^0 and D_n be the average traffic load per peer, the server traffic load and the query response time for a system with the above request and publishing probabilities, under the random walk with a given TTL_n. Let $\bar{\rho}_n$,$\bar{\rho}_n^0$ and $\bar{D}_n$ be the average traffic load per peer, the server traffic load and the query response time for a system with the same request probability, the same stopping time and with $q_n = 0$. Then

$$\rho_n \leq \bar{\rho}_n, \qquad \rho_n^0 \leq \bar{\rho}_n^0, \qquad D_n \leq \bar{D}_n,$$

for all $n \in \mathbb{N}$. Moreover, the above statement also holds if the query propagation mechanism is the expanding ring.

The lemma can be shown using a simple coupling argument: intuitively, for every sample path of the Markov process with $q_n > 0$, we can construct a sample path of a Markov process identical to the sample path of the original process, except that the item is never published. The set of positive peers in the coupled process will always be a subset of positive peers in the system with $q_n > 0$. As a result, all three metrics we consider will be higher for the coupled chain. The lemma then follows by observing that the coupled process evolves as a system with $q_n = 0$.

Lemma 5.13 implies that the analysis presented so far, for the case $q_n = 0$, is a worst-case scenario in terms of the metrics ρ_n,ρ_n^0 and D_n over all possible values of q_n. In particular, Lemma 5.13 implies the following generalization of Theorem 5.1:

Theorem 5.10. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and vertex balanced, and that there exists a constant $\bar{\tau}$ such that*

$$\mathbf{P}(\tau_n \leq \bar{\tau}) = 1 - o\left(\frac{1}{n}\right).$$

Then, both the average traffic load per peer and the server traffic load for a delay-constrained random walk with $\mathrm{TTL}_n = \Theta(n)$ are bounded in n, irrespectively of p_n and q_n.

In general, the bounds obtained by applying by Lemma 5.13 are not tight. In particular, if $p_n = o(q_n)$, the performance of the system with respect to the above three metrics will be much better compared to the performance of the system with $q_n = 0$. Our analysis can be easily extended to get a precise description of the above traffic loads. In particular, the following results can be shown about the random walk, using the same methods that we used in the case where $q_n = 0$.

Theorem 5.11. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and vertex balanced. Then, for any constant $\bar{\tau} > 1$, the server traffic load ρ_n^0 generated by queries for a item requested with probability p_n and published with probability q_n is such that*

$$\rho_n^0 \leq \mu\big[\big(np_n(1-(p_n+q_n)+(p_n+q_n)e^{-\frac{\mathrm{TTL}_n}{n\bar{\tau}\delta}}\big)^{n-1}\phi_n + np_n(1-\phi_n)\big] \quad \textit{and}$$
$$\rho_n^0 \geq \mu np_n\big(1-(p_n+q_n)+(p_n+q_n)e^{-\frac{2\mathrm{TTL}_n}{n\delta}}\big)^{n-1}(1-2p_n\bar{\tau})\phi_n,$$

where $\phi_n = \mathbf{P}(\tau_n \leq \bar{\tau})$ and τ_n the steady state relaxation time of $\{G(t)\}_{t\in\mathbb{N}}$, defined by (4.1).

Theorem 5.12. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and vertex balanced. Then, for any constant $\bar{\tau} > 1$, the average traffic load per peer ρ generated by queries for a item requested with probability p_n and published with probability q_n is such that*

$$\rho_n \leq \mu \left[\min\left(\frac{p_n}{p_n + q_n} \bar{\tau}, \mathrm{TTL}_n p_n \right) + \mathrm{TTL}_n p_n (1 - (p_n + q_n))^{n-1} + \mathrm{TTL}_n p_n (1 - \phi_n) \right].$$

where $\phi_n = \mathbf{P}(\tau_n \leq \bar{\tau})$ and τ_n the steady state relaxation time of $\{G(t)\}_{t\in\mathbb{N}}$, defined by (4.1).

One interesting implication of these results is the following. If the overlay graph is an expander *w.h.p.*, the average traffic load per peer in a system with $q_n = 0$ is bounded by a constant for popular items (with $p_n = \Omega(1/n)$) and decreasing for unpopular items (with $p_n = o(1/n)$). In a system where $q_n > 0$, items that are popular generate a traffic load of the order of

$$\rho_n = O\left(\frac{p_n}{p_n + q_n} \right).$$

Suppose that $p_n = o(q_n)$, *i.e.*, the number of peers publishing the data item exceeds, asymptotically, the number of peers requesting it. Then, the average traffic load per peer can in fact be decreasing even if the item is popular. For example, in a system where $p_n = \Theta\left(n^{-1/2}\right)$ and $q_n = \Theta(1)$, Theorem 5.12 implies that the average traffic load per peer will be

$$\rho_n = O\left(n^{-1/2}\right),$$

i.e., it will decrease with the system population, even if the expected number of peers requesting it increases with n.

Finally, similar observations can be made about the expanding ring. In particular, Lemma 5.13 implies that Theorem 5.2 also holds in the case where $q_n > 0$.

5.7.6 Multiple Data Items

Our main results, given by Theorem 5.1 and 5.2, give bounds for the traffic generated by queries for a single data item. To compute the aggregate (over all data items) traffic load per peer or at the server, one would have to sum the individual loads incurred for each item.

If the number of data items stored at the server is bounded (*i.e.*, the server provides no more than a constant number of data items), Theorems 5.1 and 5.2 extend immediately to the aggregate traffic loads. This case is of practical interest: keeping in mind that growth captures the long-term evolution of the system, it is plausible that the user base grows while the content provided by the server remains the same.

Nonetheless, it would be interesting to understand the behaviour of the system when the number of items provided by the server increases in n. We will again try to characterize the conditions under which the aggregate load (per peer and at the server) remains bounded by a constant; however, our discussion below can be easily extended to loads increasing in n.

As in Section 4.4, assume that the server stores M_n distinct data items. Moreover, a data item j, $j = 1, \ldots, M_n$, is requested by an incoming peer with a probability p_n^j. Recall that the expected number of items requested by a peer is

$$\sum_{j=1}^{M_n} p_n^j.$$

For several reasons, it is natural to assume that the above quantity is bounded in n, *i.e.*, there exists a constant $B > 0$ s.t.

$$\sum_{j=1}^{M_n} p_n^j < B, \qquad \text{for all } n. \tag{5.50}$$

To begin with, peers should not request more than a constant number of items because they have limited resources, both in terms of bandwidth and storage. In this sense, a single peer could not accommodate the retrieval of so many items, let alone the query traffic this would incur. In particular, if (5.50) does not hold, it is impossible to bound the average traffic load per peer even if only one query message is transmitted at each search.

An alternative and, perhaps, more natural interpretation of (5.50) is the following: the number of items asked by a peer should not depend on the number of items stored by the server just as the number of books checked out from a library by a library member does not depend on the size of the library.

Without loss of generality, we can assume that, for all n,

$$p_n^1 \geq p_n^2 \geq \ldots \geq p_n^{M_n}. \tag{5.51}$$

In doing so, we slightly change the meaning of the request probabilities p_n^j. Keeping in mind that different values of n correspond to different (long-term) periods in the evolution of the system, we can treat p_n^j as the request probability of the j-th most popular item when the operating size is n; a given item may have a different ranking at different periods, as it loses popularity and new items become more popular. In this sense, a re-labelling of the items is required at each time, for (5.51) to be true.

The above assumption is consistent with the analysis we have presented so far. Moreover, it is not strictly necessary for the results we present below: in particular, Theorem 5.13 and

Corollary 5.7 below hold even if (5.51) does not, and the request probabilities retain their original meaning. Nonetheless, it is easier to explain the intuition behind these results assuming the above ordering.

Let ϱ_n, ϱ_n^0 be the aggregate (over all items) traffic load per peer and at the server, respectively. Then, Theorems 5.4 and 5.5 immediately imply the following:

Theorem 5.13. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and vertex balanced. Then, for any constant $\bar{\tau} > 1$,*

$$\varrho_n \leq \mu \cdot \sum_{j=1}^{M_n} \left\{ \min\left[\bar{\tau}, \mathrm{TTL}_n p_n^j\right] + \mathrm{TTL}_n p_n^j (1-p_n^j)^{n-1} + \mathrm{TTL}_n p_n^j (1-\phi_n) \right\}.$$

and

$$\varrho_n^0 \leq \mu \sum_{j=1}^{M_n} \left[n p_n^j (1 - p_n^j + p_n^j e^{-\frac{\mathrm{TTL}_n}{n\bar{\tau}}})^{n-1} \phi_n + n p_n^j (1-\phi_n) \right]$$

where $\phi_n = \mathbf{P}(\tau_n \leq \bar{\tau})$ and τ_n the steady state relaxation time of $\{G(t)\}_{t\in\mathbb{N}}$, defined by (4.1)*.*

As a result, the following corollary holds.

Corollary 5.7. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and vertex balanced, that there exists a constant $\bar{\tau} > 0$ such that*

$$\mathbf{P}(\tau_n \leq \bar{\tau}) = 1 - o\left(\frac{1}{n}\right),$$

and two constants $M > 0$, $B > 0$ such that

$$\sum_{j=M}^{M_n} n p_n^j < B, \quad \textit{for large enough } n. \tag{5.52}$$

Then, if $\mathrm{TTL}_n = \Theta(n)$, both the aggregate (over all items) traffic load at the server and at each peer is bounded, i.e.*,*

$$\varrho_n = O(1), \quad \textit{and} \quad \varrho_n^0 = O(1).$$

The condition (5.52) is stronger than the condition (5.50). Assuming that request probabilities are ordered, it states that the probabilities of the least requested items, constituting the tail of (5.50), not only fall, but to so quickly enough so that np_n^j is summable over $j \geq M$, uniformly in n. Moreover, (5.52) implies that the $M-1$ most popular items can have *e.g.*, constant request probabilities, whereas all remaining items should have a decreasing probability, that decays very fast. To illustrate this statement, we consider an example in which items are ranked according to the Zipf "distribution".

An Example of Aggregate Loads Under Zipf File Rankings

Assume that p_n^j are given by

$$p_n^j = c_n j^{-\alpha}$$

where $c_n < 1$ and $\alpha > 0$. We consider the following cases. If $0 < \alpha < 1$, then for every constant $M > 0$,

$$\sum_{j=M}^{M_n} n c_n j^{-\alpha} = n c_n \Theta\left(M_n^{1-\alpha}\right),$$

which is bounded if and only if

$$c_n = O\left(\frac{1}{n M_n^{1-\alpha}}\right).$$

Hence, if $0 < \alpha < 1$, both ϱ_n and ϱ_n^0 will be bounded if

$$p_n^j = O\left(\frac{1}{n M_n^{1-\alpha} j^\alpha}\right),$$

for all but finitely many items. If M_n is unbounded in n, these request probabilities decrease faster than $1/n$.

Similarly, for $\alpha = 1$, one can show that both ϱ_n and ϱ_n^0 will be bounded if

$$p_n^j = O\left(\frac{1}{n \log M_n j^\alpha}\right).$$

for all but finitely many items. Finally, if $\alpha > 1$, the request probabilities are summable, and the sufficient condition for scalability becomes

$$p_n^j = O\left(\frac{1}{n j^\alpha}\right).$$

for all but finitely many items. In other words, for $\alpha > 1$, the total number of items does not affect the rate of decay necessary for scalability. In particular, the latter case allows ϱ_n and ϱ_n^0 to be bounded even if $M_n = \infty$ for all n: the number of items served by the server can be infinite!

A similar discussion applies for the aggregate traffic loads under the expanding ring mechanism, using the bounds appearing in Theorems 5.6 and 5.7. Of course, we cannot hope for the above loads to be bounded, even if the total number of items M_n is.

5.8 Summary

Our theoretical analysis and our numerical study suggest that hybrid peer-to-peer systems with desirable scalability properties can be constructed based on the random walk and the expanding ring. In particular, our analysis shows that a system in which both the traffic load at the server and the traffic load at peers can be bounded irrespective of the system size, a result corroborated by our simulations.

An interesting future extension of this work would be to investigate the scalability of different query propagation mechanisms, including variants of the above mechanisms that include path replication or caching of data items. In the next chapter we pursue a different direction, by applying our model to a pure peer-to-peer system, where no server exists.

Chapter 6

Pure Peer-to-Peer System

In this chapter, we consider the pure peer-to-peer system architecture[1]. The main difference from the hybrid system discussed in Chapter 5 is that no server exists; as a result, queries can fail, and a data item may not always be retrieved successfully by the peer that requested it. We will again focus on the scalability of the random walk and expanding ring mechanisms, captured here by the query traffic load they generate on peers alone. In addition, as queries can now fail, we will also be interested in the reliability of these mechanisms; as we will see, the latter will be captured by the query success rate, *i.e.*, the probability that, given that a query takes place, it succeeds in locating the requested data item.

It has been observed [LCC+02, YGM02, TK06] that both the random walk and the expanding query propagation mechanisms scale very well, in terms of the traffic load they generate at peers, when searching for data items that are available in the network —*i.e.*, that are stored by at least one peer. However, both mechanisms generate considerable query traffic when the item requested is not stored by any peer. We show this formally in Section 6.3, as well as through simulations in Section 6.5.

Unfortunately, searches for items that are not in the system are very common in practise. For example, as discussed in Section 2.2, Acosta and Chandra [AC08c] took several snapshots of the Gnutella peer-to-peer system and monitored both the queries issued by peers, as well as the data items present in the system. The authors observed that roughly half of the queries (between 44% and 55.6%) could not be matched to *any* item. A consequence of this behaviour is that unstructured peer-to-peer systems using the random walk or expanding ring query propagation mechanisms do not scale, precisely because of the large traffic generated by queries for

[1] An earlier version of the work presented in this chapter appeared in [IM09].

absent data items.

One approach used in practise to reduce the amount of query traffic generated by searches for unavailable data items is to initialize the time-to-live field of query messages to a small value (see also our discussion in Sections 6.3 and 6.5.2). However, searches with a small initial time-to-live value will only reach a small fraction of the peers in the system; as a result, queries are likely to be unsuccessful, even if the requested item is actually available in the network. Measurement studies on actual unstructured peer-to-peer networks [LHH+04, LHSH04, CCR04, AC08c] indicate that time-to-live fields are typically initialized to such low values that a large fraction of queries for available data items are unsuccessful (see also Section 2.2). A consequence of this behaviour is that the query success rate in unstructured peer-to-peer systems is very small, typically close to 10% [ZCSK07]. In other words, reducing the time-to-live of query messages can make queries very unreliable.

The reason why a short time-to-live value leads to low success rates is that it limits the "search radius" of both searches for items that are in the network and items that are not. This suggests that a better approach would be to only limit queries for unavailable items without affecting queries for items in the system. This would make the system scalable (by eliminating long, unsuccessful searches) without affecting the success rate of queries for available items.

Therefore, a fundamental question in the design of query propagation mechanisms for unstructured networks is whether this can be done. In other words, *is it possible to limit searches for unavailable items without reducing the query success rate for items that are in the system*? The goal of this chapter is to study this issue. Surprisingly, we find that this can indeed be done.

The following provides some intuition as to why this is the case. If peers knew a priori whether an item is in the system or not, limiting searches for items not in the system would be trivial: peers simply need not search for such an item, as they have no chance of finding it. Of course, the catch is that it is not clear how to determine whether an item is in the system before searching for it. On the other hand, *after* searching for an item, a peer has more information about this item than it did before: a failed search suggests that the item is likely to not be in the system. In this sense, the outcome of a search can help in determining whether an item is available or not.

This observation leads us to consider the following mechanism. If a peer fails to locate an item, it treats the failed query as "evidence of absence" of the item. It then stores this information, for as long as it remains in the system, and shares it with other peers. That is, whenever it receives a query for this same item, it stops the propagation and informs the

peer that issued the query that the item is (likely to be) unavailable. The peer that receives this information will treat this again as "evidence of absence" of the item and share this information with other peers in the same manner. Basically, the system treats the *absence of evidence* (the failure to locate the item) as *evidence of absence* (evidence that the item is not available).

The above mechanism indeed succeeds at stopping searches for items that are not in the system early on, without jeopardizing the query success rate. More precisely, we show that the proposed mechanism is (a) scalable, *i.e.*, it keeps the query traffic load generated by searches for any item bounded at each peer, and (b) reliable, *i.e.*, it is able to locate all items that are brought into the system *w.h.p.* We obtain this result both through a formal analysis, based on the mathematical model of unstructured peer-to-peer networks that we introduced in Chapter 4, as well as by numerical results, presented in Section 6.5. To the best of our knowledge, this is the first time that a search mechanism for unstructured peer-to-peer networks has been shown to be both scalable and reliable.

The remainder of this chapter is organized as follows. In Section 6.1, we review the main assumptions employed by our model of a pure peer-to-peer system, and present our proposed mechanism (that is both scalable and reliable) in detail. In Section 6.2, we outline and discuss our main results, namely, Theorems 6.1 and 6.2. These two theorems are proved in Sections 6.3 and 6.4, respectively. In Section 6.5, we validate our results through a numerical study and, finally, we discuss possible extensions and open questions in Section 6.6.

6.1 Model

We again use, for our analysis, the model introduced in detail in Chapter 4. Below, we summarize once more our main assumptions, focusing mostly on similarities and differences between this and the hybrid case. We then describe in more detail the search mechanism that uses "evidence of absence". Finally, we conclude this section with the performance metrics that we will use in our analysis.

In our model, as in the hybrid case, at any point in time, the system consists of n peers, which stay in the system for exponentially distributed times with mean $1/\mu$. Upon departure, a peer is immediately replaced by a new peer, thus keeping the system size constant. Again, without loss of generality, we first focus on the dynamics of a system in which only one data item is shared among peers; the multiple item case will be covered in Section 6.6.5. Incoming peers request the item with probability p_n (the request probability) and publish it with proba-

bility q_n, (the publishing probability), where

$$0 < p_n < 1, \quad 0 < q_n < 1, \quad 0 < p_n + q_n < 1, \tag{6.1}$$

Note that, contrary to the hybrid case, we will require a non-zero publishing probability. If $q_n = 0$, the pure peer-to-peer system is not ergodic: eventually, no peer in the system will have the item.

As in the hybrid case, we assume that the total query response time is negligible compared to the lifetime of a user: we decouple the propagation of a query from the rest of the dynamics of the system, assuming that queries are instantaneous when viewed in the timescale determined by user arrivals and departures.

Some of our results (Theorems 6.3 and 6.5) are proved under the usual assumption that the overlay graph $\{G(t)\}_{t\in\mathbb{N}}$ right after the t-th departure/arrival event is an ergodic, churn-driven Markov chain, whose state space is $\mathbb{S}_{n,d} \subseteq \mathbb{MG}_{n,d}$. However, for some of our results (Theorems 6.4 and 6.6), we will require a much stronger assumption, namely, that $\{G(t)\}_{t\in\mathbb{N}}$ is an independent graph sequence, as defined in Section 4.2.4. *I.e.*, $\{G(t)\}_{t\in\mathbb{N}}$, the overlay graph at different simultaneous departure/arrival epochs, is a sequence of i.i.d. random variables sampled from a set $\mathbb{S}_{n,d}$ according to a given distribution. As any independent graph sequence is an ergodic, vertex balanced, churn-driven Markov chain, any result proved under the general model (and, in particular, Theorems 6.3 and 6.5) also hold under this (stronger) assumption.

Again, we denote by τ_n the steady state relaxation time of $\{G(t)\}_{t\in\mathbb{N}}$, *i.e.*, the relaxation time of a random graph sampled from $\mathbb{S}_{n,d}$ according to the stationary distribution of $\{G(t)\}_{t\in\mathbb{N}}$. Recall, from Section 4.2.1, that τ_n is a random variable in $\mathbb{R}_+ \cup \{\infty\}$ whose distribution is given by (4.1).

The assumption that $\{G(t)\}_{t\in\mathbb{N}}$ is an i.i.d. sequence of graphs is unrealistic. It introduces more "randomness" into the overlay graph topology than one encounters in a real system: as arrivals and departures affect the network only "locally", overlay topologies between subsequent departure/arrival epochs might overlap and, therefore, are not independent. A churn-driven Markovian graph model thus gives a more accurate depiction of the evolution of the overlay graph in a real system. However, such a system is considerably harder to analyze than in the hybrid case, as discussed in Sections 6.3.1 and 6.3.3.

In our numerical study of Section 6.5, we validate our results by relaxing our assumption that $\{G(t)\}_{t\in\mathbb{N}}$ is an i.i.d. sequence. In particular, in our simulations, the overlay graph is not sampled independently, but evolves according to the Law and Siu connection protocol presented in Section 4.2. Also, as in the simulations of Chapter 5 and contrary to the above

modelling assumptions, the simulated system might change during query propagations and its size is time-variant, rather than fixed to an operating value n.

6.1.1 Absence of Evidence as Evidence of Absence

Our analysis will again focus on the query propagation mechanisms defined in Section 5.1.1, namely, the random walk and the expanding ring. We also introduce a variant of the random walk mechanism, which we call a *random walk using evidence of absence*.

To define the precise operation of this search mechanism, we use the following notation. For a given data item, we distinguish among three different types of peers: *positive* peers, which are peers that have a copy of the item, *negative* peers, which are peers that believe that the item is not in the system, and *null* peers, which are peers that neither have the item nor believe it is not in the system. Peers that publish the item, *i.e.*, that already have a copy of the item when they enter the network, are always positive. Peers that request the item can become either positive or negative, depending on the outcome of the search. Finally, peers indifferent to the item, that neither request it or bring it in the system, are null.

The search mechanism is illustrated in Figure 6.1, and works as follows. In order to locate a data item, peers propagate a query over the overlay network using a random walk mechanism (either hop-constrained or delay-constrained). For example, a peer issuing a query (the source peer) chooses one of its d neighbours in the overlay network at random and forwards a query packet to it. When a query reaches a positive peer, *i.e.*, a peer that has a copy of the item, the positive peer notifies the source peer and allows it to download a copy of the item, thus converting it to a positive peer. When a query reaches a negative peer, the negative peer responds by informing the source peer that the item is (likely) not in the system, thus converting it to a negative peer. In either case, the query propagation ceases upon reaching the positive or negative peer. When a query reaches a null peer, the null peer continues to forward the query by choosing one of its d neighbours at random, and sending the query to it, as long as the query has not expired. In the latter case, the null peer terminates the query and informs the source peer about the query's failure. This, in turn, also converts the source peer to a negative peer.

We call this mechanism a random walk using evidence of absence because, as discussed earlier, this mechanism uses the absence of evidence (*i.e.*, the inability to locate the item) as evidence of the absence of a particular item. Note that the "bad news" of the absence of the item are exchanged among peers in precisely the same way as copies of items are exchanged among peers.

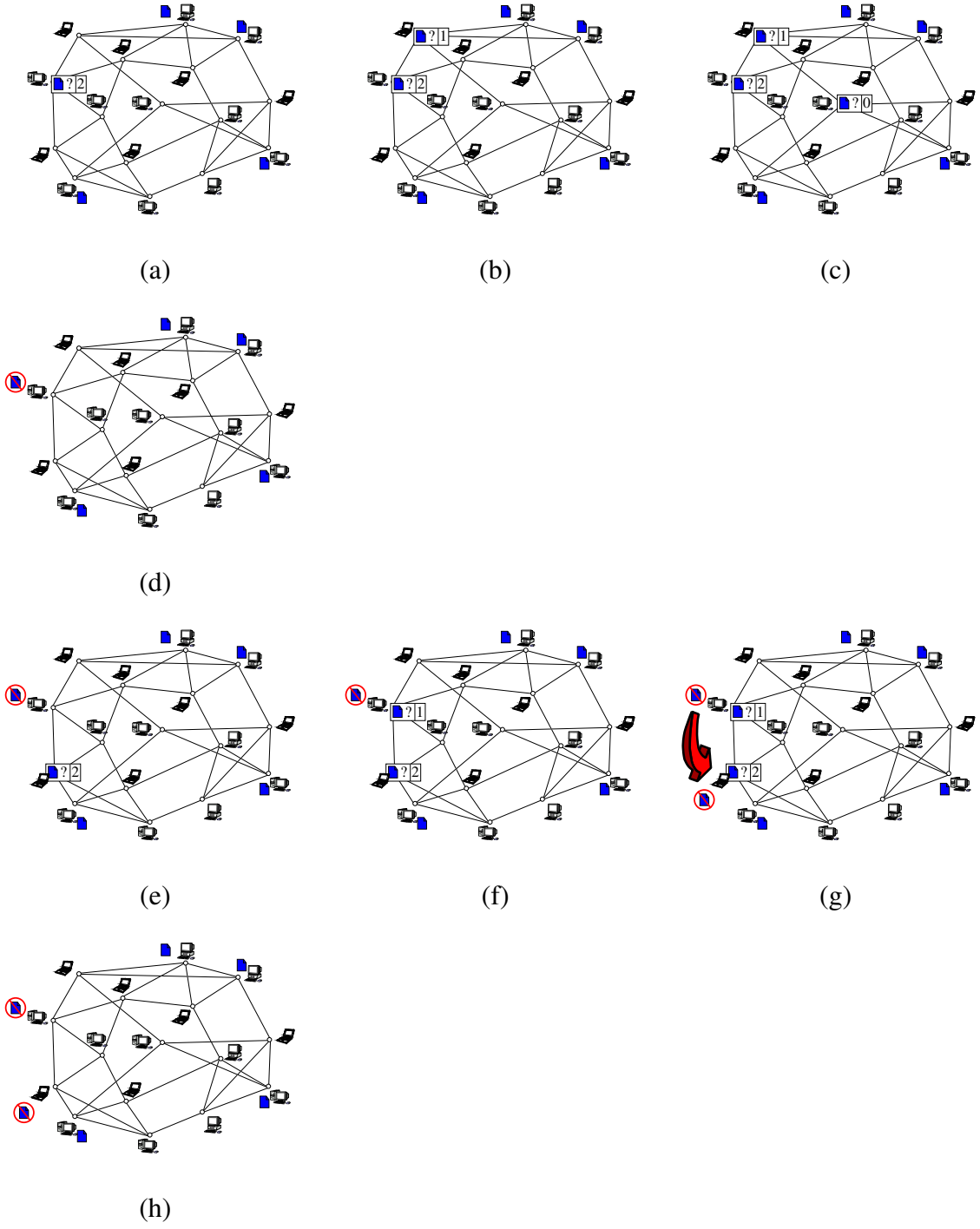


Figure 6.1: Example of random walk search using evidence of absence. In (a)-(c), a query is propagated, but fails to locate the item. As a result, in (d), the peer that initiated the query becomes negative, *i.e.*, it believes that the item is not in the system. In (e)-(g), a second peer requests the item and the search is stopped by the negative peer: it responds by informing the second peer that the item is not in the system. After this, the second peer is also converted to a negative peer, as shown in (h).

The above scheme can be generalized to an arbitrary query propagation mechanism, with appropriate modifications. For example, if an expanding ring is used, a peer can increase the flooding radius until it receives a response from a positive or a negative peer. The source peer can become positive, if it receives a positive reply, or negative, if it receives a negative reply or no reply at all. In general, in a mechanism in which a source peer may receive more than one replies, it can become negative if and only if it does not receive a positive reply, *etc.*

6.1.2 Performance Metrics

As in the hybrid case, we capture the scalability of our system by characterizing how the average traffic load per peer ρ_n behaves in terms of the system size (see also Section 5.1.2). However, the scalability of a pure peer-to-peer system must be taken into consideration in conjunction with the probability that queries fail. For example, one can trivially make a pure peer-to-peer system scalable by not propagating any query. In such a scheme, ρ_n is trivially bounded (it is zero!), but no query succeeds in retrieving the data item. In other words, we would like a pure peer-to-peer system to not only be scalable, but also reliable, in the sense that queries for an item have a high likelihood of succeeding.

For this reason, when looking at a pure peer-to-peer system, we are also interested in the query success rate, *i.e.*, the probability that, given that a query is issued, the peer that initiated it succeeds in retrieving the data item requested. Note that this quantity is interesting only in a pure peer-to-peer system as, in a hybrid systems, all queries are eventually answered (by potentially retrieving the item from the server). Below, we give a formal definition of this quantity in terms of our model.

Query Success Rate

Consider a reward function $\{R(t)\}_{t\in\mathbb{N},t>0}$ defined over the sequence of queries issued in the system as follows

$$R(t) = \begin{cases} 1, & \text{if the data item is retrieved at the } t\text{-th query and} \\ 0, & \text{otherwise.} \end{cases}$$

We are interested in the time-average query success rate γ_n, defined as

$$\gamma_n = \lim_{t\to\infty} \frac{1}{t} \sum_{s=1}^{t} R(s).$$

Again, the above limit may not necessarily exist. If the system is ergodic, then the key renewal theorem implies that it does exist *a.s.*, and that it equals

$$\gamma_n = \lim_{t\to\infty} \mathbb{E}[R(t)] = \lim_{t\to\infty} \mathbf{P}(R(t) = 1).$$

That is, under proper ergodicity assumptions, γ_n can be defined, equivalently, as the steady state probability that a query succeeds.

Definitions of Scalable and Reliable Query Propagation Mechanisms

To summarize, for a given a data item with request probability p_n and publishing probability q_n, we are interested in the performance of a query propagation mechanism in terms of the average load per peer ρ_n and the query success rate γ_n.

We will say that a query propagation mechanism over a pure peer-to-peer system is *scalable* if the average load per peer ρ_n stays bounded as the system size n increases, *i.e.*, we have

$$\rho_n = O\left(1\right). \tag{6.2}$$

for all possible combinations of p_n and q_n. In addition, we will say that a query propagation mechanism is *reliable* if queries for data items that are brought sufficiently often in the system are almost always successful. More precisely, we will say that a search mechanism is reliable if

$$\text{if } q_n = \omega\left(\frac{1}{n}\right) \text{ then } \lim_{n\to\infty} \gamma_n = 1, \tag{6.3}$$

for all possible values of p_n.

Keeping in mind that nq_n is the expected number of peers in the system that publish the item —*i.e.*, already had a copy of the item when they entered the system— $q_n = \omega\left(\frac{1}{n}\right)$ means the expected number of peers in the system that publish the item increases with n (see also Section 4.4); however, this increase can be arbitrarily slow. For example, the expected number of peers in the system that publish the item can grow as slowly as $\log\log(n)$, or even slower. The above definition is therefore just slightly weaker than requiring that queries are successful if there exists at least one peer in the system that publishes the item, in expectation.

6.2 Main Results

This section includes the formal statement of our two main results for pure peer-to-peer systems, namely, Theorems 6.1 and 6.2. The first result states that the random walk and the

expanding ring are not scalable and reliable, while the second result states that the random walk with evidence of absence exhibits both properties. The proofs of these two statements are presented in Sections 6.3 and 6.4, respectively; the discussion below aims at illustrating the implications of these results, as well as the intuition behind them.

6.2.1 Random Walk and Expanding Ring

Our first main result is a negative result, the proof of which can be found in Section 6.3:

Theorem 6.1. *Assume that* $\{G(t)\}_{t\in\mathbb{N}}$ *is an i.i.d. sequence of graphs sampled from a label-independent distribution. Them, there is no* TTL_n *such that the hop-constrained random walk is both scalable and reliable. Moreover, the same statement holds for the expanding ring.*

The intuition behind the above theorem is that the hop-constrained random walk and the expanding ring can be either scalable or reliable, but not both. In particular, if $\mathrm{TTL}_n = \omega(1)$, neither mechanism will be scalable: there exists a pair of p_n and q_n such that the average traffic load per peer ρ_n will be unbounded as n increases. On the other hand, if $\mathrm{TTL}_n = O(1)$, both the random walk and the expanding ring are scalable. However, in this case, neither of them is reliable, in the sense of (6.3). On the contrary, queries are not guaranteed to succeed even if a very large fraction of peers publishes the item: as long as the number of peers that publish the item, in expectation, grows slower than linear in n, (*i.e.*, $q_n = o(1)$) the query success rate will not converge to one.

The statement regarding scalability (Theorem 6.3) is stated and proved in Section 6.3 separately from the statement regarding reliability (Theorem 6.4). We note that the independent graph model is only required for the reliability result; the analysis of scalability is proved for a generic churn-driven Markovian model. Moreover, both statements are validated in Section 6.5 through a numerical case study, that is conducted under a churn-driven Markovian setting.

6.2.2 Random Walk Using Evidence of Absence

Our second main result, proved in Section 6.4, is positive:

Theorem 6.2. *Assume that* $\{G(t)\}_{t\in\mathbb{N}}$ *is an* i.i.d. *sequence of graphs sampled from a label-independent distribution. Moreover, assume there exists a constant* $\bar{\tau}$ *such that*

$$\mathbf{P}(\tau_n \leq \bar{\tau}) = 1 - o\left(\frac{1}{n}\right),$$

where τ_n the relaxation time of a graph sampled from the (stationary) distribution of $\{G(t)\}_{t\in\mathbb{N}}$. Then, the delay-constrained random walk with $\mathrm{TTL}_n = \Theta(n)$ using evidence of absence is both scalable and reliable.

It is interesting to contrast this result with the performance of the hop-constrained random walk and the expanding ring in which peers do not share information about failed queries. By Theorem 6.1, such mechanisms cannot be both scalable and reliable, no matter what the TTL_n value used. In fact, to guarantee scalability, one needs to relinquish a high success rate for all queries except for items that have a constant publishing probability $q_n = \Omega(1)$. In this light, the fact that the random walk using evidence of absence leads to a bounded average load per peer for all possible functions p_n and q_n, and queries succeed as long as $q_n = \omega(1/n)$, is quite remarkable.

Again, though the statement about reliability (6.6) requires that $\{G(t)\}_{t\in\mathbb{N}}$ is an i.i.d. sequence, the scalability of the proposed mechanism (Theorem 6.5) in proved under the more general churn-driven Markovian model. As a result, both the Law and Siu model (Section 4.3.1) and the switch model restricted over $\mathbb{MI}_{n,d}$ (Section 4.3.2) satisfy the conditions guaranteeing the scalability of the above mechanism. In any case, both the statement regarding scalability and the statement regarding reliability are validated through our numerical case study of Section 6.5, that is conducted under the general churn-driven Markovian setting.

6.2.3 Some Intuition Behind Theorem 6.2

The main technical challenge of this chapter is the proof of the reliability result of Theorem 6.2, appearing in Section 6.6. The greatest hurdle in showing that the proposed mechanism is reliable amounts to proving that one can indeed treat the "absence of evidence" as "evidence of absence". In other words, one has to show that when the item is sufficiently often brought into the system and $q_n = \omega(1/n)$, then false negatives (*i.e.*, queries that failed even though the item is actually in the system) are very rare, and searches for such an item succeed *a.a.s.* We try to give insight into why this is the case by describing a simple system with some key similarities to our pure peer-to-peer system.

Consider a bin consisting of n balls where balls are either blue, red, or white. At constant intervals $t = 1, 2, 3...$, one ball is removed from the bin, and immediately replaced by either Alice, Bob, or Charlie, as follows. Let $p, q \in (0, 1]$ be such that $p + q < 1$. Then, with probability q, Alice replaces the removed ball with a blue ball, and with probability $1 - p - q$ Charlie replaces it with a white ball. Otherwise, Bob replaces the removed ball either with a

blue or red ball. The probability that Bob puts in a red ball is equal to the ratio of the number of red balls to the total number of red and blue balls in the bin.

Let's now just focus on the ratio of blue to red balls in the bin. Clearly, Charlie does not affect this ratio as she only puts in white balls. Furthermore, Bob also does (on average) not change the ratio of red to blue balls that are currently in the bin. However, Alice always puts in a blue ball, and hence changes the ratio between red and blue balls always into the favour of blue balls. Because of this, if the process goes on forever there will eventually only be white and blue balls in the bin, but no red balls.

We can map the situation to our search mechanism by letting positive peers be blue balls, negative peers be red balls, and null peers be white balls; and let q_n and p_n be the publishing and request probabilities of a given data item. With probability q_n, a new peer that enters the system will publish the item and become a positive peer. Hence, such a peer corresponds to the action of Alice in the above analogy, and puts a blue ball into the bin.

With probability p_n, a new peer will request the item and become a positive or negative depending the outcome of the search. In Section 6.4.3, we show that having $q_n = \omega(1/n)$ guarantees that searches will terminate by either hitting a positive or negative peer *a.a.s.* Hence, such a request corresponds to the action of Bob, and puts a blue or red ball in the bin. Moreover, a ball is chosen according to the current ratio of positive and negative peers (red and blue balls) in the system. Finally, with probability $1 - p_n - q_n$ a new peer will not request the item and become a null peer, which corresponds to the action of Charlie.

Using the above analogy, if a sufficiently large number of peers bring the item into the system (*i.e.* if $q_n = \omega(1/n)$) then, eventually, there will only be positive (blue balls) and null (white balls) peers in the system, and all negative peers (red balls) will die out. Furthermore, as for $q_n = \omega(1/n)$ all searches will always end by either hitting a positive or a negative peer, searches will always terminate by hitting a positive peer (blue ball) and hence be successful.

Formalizing the argument presented above requires additional effort to address that almost all searches find a non-null peer asymptotically (as n tends to infinity). As we will see, to do so, in our proof of Theorem 6.6, we use the mean field limit method developed by Benaïm and Le Boudec [BL08], described in detail in Section 6.3.3.

6.3 Random Walk and Expanding Ring

In this section, we analyze a pure peer-to-peer system that does not use evidence of absence. In such a system, upon the failure of a query, the peer issuing the request does not obtain a copy

of the item and simply becomes null. As discussed in Section 6.2, the first of our two main results (namely, Theorem 6.1) is that, in such a system, the random walk and the expanding ring cannot be both scalable and reliable.

This section is devoted to the proof of this statement. In particular, we first characterize the conditions under which the general churn-driven Markovian model is ergodic (Section 6.3.1). We then show our main result regarding the scalability of the random walk and the expanding ring (Section 6.3.2): as long as the stopping time TTL_n of these mechanisms grows with n, these mechanisms are not scalable.

To characterize the reliability of these mechanisms, we depart from our general churn-driven Markovian model and assume that the overlay consists of a sequence of i.i.d. graphs. We describe the implications of this assumption in detail (Section 6.3.3); in particular, we show that such a system can be represented as a *mean field interaction model* [BL08]. We then show that, under the i.i.d. assumption, the random walk and the expanding ring with $\mathrm{TTL}_n = O(1)$ will not be reliable (Section 6.3.4). Finally, combining this reliability result with the scalability result discussed above yields Theorem 6.1 (Section 6.3.5).

Characterizing the reliability of the random walk and the expanding ring is the most technically challenging part of our proof. In particular, we employ the mean field limit method developed by Benaïm and Le Boudec [BL08]. The main advantage of this method is in approximating the system's evolution by a deterministic process; this is discussed in more detail in Section 6.3.3.

6.3.1 Ergodicity

We first discuss the evolution of the pure peer-to-peer system under the assumption that $\{G(t)\}_{t\in\mathbb{R}_+}$ is a general churn-driven Markovian graph process. Recall that peers can be either positive or null in a system where evidence of absence is not used. We denote by $A_+(t) \subseteq V$ the set of positive peers at time t. Then, under the assumption that $\{G(t)\}_{t\in\mathbb{R}_+}$ is a churn-driven Markovian process with state space $\mathbb{S}_{n,d} \subseteq \mathbb{MG}_{n,d}$, as described in Section 4.2.1, the process $\{A_+(t), G(t)\}_{t\in\mathbb{R}_+}$ is a Markov process whose state space is $2^{V_n} \times \mathbb{S}_{n,d}$; transitions occur at Poisson epochs with rate $n\mu$, and depend only on the previous state.

We denote the embedded Markov chain of the joint process by $\{A_+(t), G(t)\}_{t\in\mathbb{N}}$. As the Markov process is uniformized, a stationary distribution of the embedded chain will also be a stationary distribution of the Markov process and vice-versa.

Conditioned on $\{G(t)\}_{t\in\mathbb{N}}$, the transitions of $\{A_+(t)\}_{t\in\mathbb{N}}$ are as follows. At the t-th depar-

ture/arrival epoch, the peer being replaced publishes the item in the system with probability q_n. With probability p_n, it initiates a search: A query is propagated over the graph $G(t)$; according to the outcome of the search, the peer will become either positive or null. Finally, with probability $1 - p_n - q_n$, the peer simply becomes null.

A property that proved to be very useful in the analysis of the hybrid system was that the status of a given peer (*i.e.*, positive or null) was independent of the status of other peers. Unfortunately, this property is not true in the pure peer-to-peer system: Intuitively, conditioning on a peer i being positive biases the probability that a query succeeds, thereby increasing the likelihood that a peer $j \neq i$ is also positive. Moreover, the stationary distribution of the joint process $\{A_+(t), G(t)\}_{t\in\mathbb{R}_+}$ does not necessarily have a simple product form as the one appearing in Theorem 5.3. Intuitively, conditioning on the overlay graph gives us some information on the likelihood that a search will succeed; in turn, this also affects the distribution of the set of positive peers.

In fact, the process $\{A_+(t)\}_{t\in\mathbb{R}_+}$ may not even be a Markov process! All of the above are implications of the fact that, contrary to the hybrid system, the transitions of process $\{A_+(t)\}_{t\in\mathbb{R}_+}$ depend on the outcome of a query propagation, which, in turn, depends on $\{G(t)\}_{t\in\mathbb{R}_+}$.

Nonetheless, under our standing assumption that

$$0 < p_n < 1, \quad 0 < q_n < 1, \quad 0 < p_n + q_n < 1,$$

we can still prove that the ergodicity of the joint chain $\{A_+(t), G(t)\}_{t\in\mathbb{N}}$ is equivalent to the ergodicity of $\{G(t)\}_{t\in\mathbb{N}}$.

Lemma 6.1. *The joint Markov chain $\{A_+(t), G(t)\}_{t\in\mathbb{N}}$ is ergodic if and only if the marginal Markov chain $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic.*

Sketch of proof. The proof of is almost identical to the proof of the corresponding statement for the hybrid system (Corollary 5.1). Essentially, for any two sets $A, A' \subseteq V_n$, we can construct a non-zero probability path from A to A' by adding elements with probability q_n and removing elements with probability $1 - p_n - q_n$. Using this idea, the steps followed in the proof of Lemmas 5.1 and 5.2 can be replicated almost verbatim, without involving any paths in which requests take place. □

Hence, if $\{G(t)\}_{t\in\mathbb{N}}\}$ is ergodic, the average traffic load per peer ρ_n and the query success rate γ_n exist (that is, the limits defined in Sections 5.1.2 and 6.1.2, respectively, exist *a.s.*). For

the average traffic load per peer in particular, we can define $\{C(t)\}_{t\in\mathbb{N}}$ as the following reward function

$$C(t) = \begin{cases} 0, & \text{if no request is issued at the } t\text{-th epoch,} \\ C_{i,A,G}, & \text{if a request is issued at the } t\text{-th epoch,} \end{cases} \tag{6.4}$$

where $C_{i,A,G}$ is the number of messages generated by a query propagation given that, at the time the query is issued, the source peer is i, the set of positive peers is A and the overlay graph is G. Then, the following lemma, which is an analogue of Lemma 5.5 in Chapter 5, holds:

Lemma 6.2. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic. Then, the average traffic load per peer ρ_n exists* a.s. *and is equal to*

$$\rho_n = \mu \cdot C_n$$

where

$$C_n = \lim_{t\to\infty} \mathbb{E}[C(t)] = \lim_{t\to\infty} \frac{1}{t} \sum_{s=0}^{t} C(s), \quad \text{a.s.}$$

and $\{C(t)\}$ the reward defined in (6.4)*.*

The proof is identical to the proof of Lemma 5.5.

6.3.2 Scalability

In this section, we discuss the scalability of a general class of search mechanisms. This class consists of the mechanisms that exhibit the following property: conditioned on the fact that a query fails, the expected number of messages generated by a query propagation is unbounded (*i.e.*, $\omega(1)$). This class includes both the hop-constrained and the delay constrained random walk with $\mathrm{TTL}_n = \omega(1)$, as well as the expanding ring with $\mathrm{TTL}_n = \omega(1)$ as, upon the failure of a query, all three mechanisms generate $\Omega(\mathrm{TTL}_n)$ messages, in expectation.

More formally, let $C_{i,A,G}$ be the number of messages generated by a certain query propagation mechanism, given that, at the time the query is issued, the source peer is i, the set of positive peers is A and the overlay graph is G. As we have already seen in the case of a hybrid system, $C_{i,A,G}$ is a random variable for both the hop-constrained and the delay-constrained random walk, while it is deterministic for the expanding ring. We denote by

$$\mathbb{E}[C_{i,A,G} \mid \text{failure}]$$

the expected message cost of a search conditioned on the event that the query fails to locate the item. Let $\mathcal{C}$ be the class of mechanisms that satisfy the property

$$\mathbb{E}[C_{i,A,G} \mid \text{failure}] = \omega(1),$$

i.e., the message cost of a failed query is unbounded.

Lemma 6.3. *Class $\mathcal{C}$ includes the hop-constrained and the delay constrained random walk with $\mathrm{TTL}_n = \omega(1)$, as well as the expanding ring with $\mathrm{TTL}_n = \omega(1)$.*

Proof. For both the hop-constrained and the delay-constrained random walk $\mathbb{E}[C_{i,A,G} \mid \text{failure}] = \mathrm{TTL}_n$. For the expanding ring mechanism, $\mathbb{E}[C_{i,A,G} \mid \text{failure}] > \mathrm{TTL}_n$. □

We will show that no mechanism belonging to $\mathcal{C}$ can be scalable. The proof relies on the fact that there exists a publishing probability q_n that is so small so that almost all queries fail, irrespectively of both the query propagation mechanism used and of the request probability p_n. As a result, for such q_n, almost all queries of a mechanism belonging to $\mathcal{C}$ will generate an unbounded number of messages. By Lemma 6.2, the average traffic load ρ_n of such a mechanism will be unbounded for this q_n and for p_n, *e.g.*, a constant. Hence, no such mechanism can be scalable, based on the definition of scalability given in Section 6.1.2.

We begin our analysis by proving the existence of the publishing probability under which almost all queries are guaranteed to fail. To obtain this probability, we consider a query propagation mechanism that has the following property: as long as a copy of the item exists in the system, any query under this mechanism succeeds in locating the item. We call a mechanism that satisfies this property a *maximal success rate* mechanism.

One way to implement a maximal success rate mechanism in a connected network is by crawling the entire overlay graph. Of course, that would be very inefficient; moreover, crawling the entire network may not even be possible if the overlay graph is disconnected.

Nonetheless, even if we cannot implement the above mechanism, it is well defined, in the sense that we can fully describe the transitions of $\{A_+(t), G(t)\}_{t\in\mathbb{N}}$ under this mechanism. Moreover, calling it a maximal success rate mechanism is indeed justified: the following lemma, whose proof uses a coupling argument, states that a maximal success rate mechanism achieves the maximum query success rate possible.

Lemma 6.4. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic, and let γ_n^* be the query success rate under the maximal success rate mechanism. Then, the query success rate γ_n under an arbitrary query propagation mechanism is such that*

$$\gamma_n \leq \gamma_n^*.$$

Proof. Recall that $A_+(t) \subset V_n$, $t \in \mathbb{R}_+$, is the set of positive peers in the system. As discussed in Section 6.3.1, for any query propagation mechanism, $\{A_+(t), G(t)\}_{t\in\mathbb{R}_+}$ is a Markov process. We construct a process $\{A^*_+(t), G^*(t)\}_{t\in\mathbb{R}_+}$ coupled to $\{A_+(t), G(t)\}_{t\in\mathbb{R}_+}$ as follows. First, $G^*(t) = G(t)$, *i.e.*, the transitions of the overlay graph are identical for both processes. Second, $A^*_+(0) = A_+(0)$, *i.e.*, both processes start with the same set of positive peers at time 0.

The evolution of A^*_+ is coupled to the evolution of A_+ as follows. First, whenever a departure/arrival event occurs in A_+, the same event occurs in A^*_+; *i.e.*, whenever a peer $i \in V_n$ is replaced in $\{A_+(t), G(t)\}_{t\in\mathbb{R}_+}$, the same peer is replaced in $\{A^*_+(t), G(t)\}_{t\in\mathbb{R}+}$. If i publishes the item in A_+ or it becomes null, it also does so in A^*_+. If i requests the item, then, in $\{A^*_+, G(t)\}_{t\in\mathbb{R}_+}$, it becomes positive as long as there is another peer in the system that is positive. Note that, in contrast, in $\{A_+, G(t)\}_{t\in\mathbb{R}_+}$, peer i becomes positive according to the outcome of the propagation mechanism (it may become positive or null).

Thus, these two Markov processes are defined in a joint probability space and the joint process $\{A_+(t), A^*_+(t), G(t)\}_{t\in\mathbb{R}_+}$ is a Markov process. At any point in time, $A_+(t) \subseteq A^*_+(t)$; this can be shown by induction on the transitions of the joint chain. As a result, any query that succeeds in $\{A_+(t), G(t)\}_{t\in\mathbb{R}_+}$ will also succeed in $\{A^*_+(t), G(t)\}_{t\in\mathbb{R}_+}$.

Let γ_n, γ^*_n the query success rates of the marginal processes $\{A_+(t), G(t)\}_{t\in\mathbb{R}_+}$ and $\{A^*_+(t), G(t)\}_{t\in\mathbb{R}_+}$, respectively. That is, as defined in Section 6.1.2,

$$\gamma_n = \lim_{t\to\infty} \frac{1}{t} \sum_{s=1}^{t} R(s),$$

where

$$R(t) = \begin{cases} 1, & \text{if the item is retrieved at the } t\text{-th query of } \{A_+(t), G(t)\}_{t\in\mathbb{R}_+} \text{ and} \\ 0, & \text{otherwise,} \end{cases}$$

and

$$\gamma^*_n = \lim_{t\to\infty} \frac{1}{t} \sum_{s=1}^{t} R^*(s),$$

where

$$R^*(t) = \begin{cases} 1, & \text{if the item is retrieved at the } t\text{-th query of } \{A^*_+(t), G(t)\}_{t\in\mathbb{R}_+} \text{ and} \\ 0, & \text{otherwise.} \end{cases}$$

Lemma 6.1 implies that both γ_n and γ^*_n exist *a.s.* Since $A_+(t) \subseteq A^*_+(t)$ for all $t \in \mathbb{R}$, we have that $R(t) \leq R^*(t)$ for all $t \in \mathbb{N}$. As a result,

$$\gamma_n \leq \gamma^*_n.$$

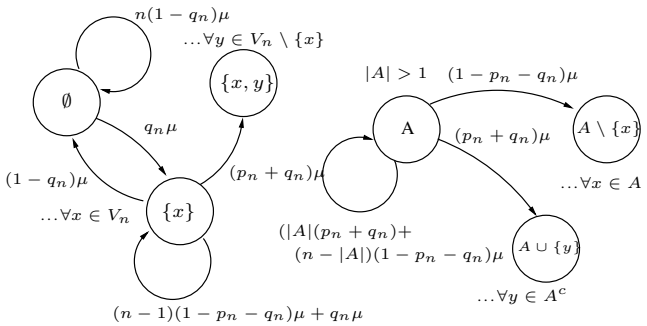


Figure 6.2: The Markov process $\{A^*_+(t)\}_{t\in\mathbb{R}_+}$, the set of positive peers under the maximal success rate algorithm.

Finally, the lemma follows by observing that the marginal chain $\{A^*_+(t), G(t)\}_{t\in\mathbb{R}_+}$ evolves as a system using the maximal success rate algorithm. □

Hence, γ^*_n can be used to obtain an upper bound on γ_n for *any* search mechanism. Interestingly, γ^*_n is straightforward to compute, as the Markov process describing the evolution of the system under the maximal success rate mechanism is fairly simple.

Lemma 6.5. *Let γ^*_n be the steady state probability that a query succeeds under the maximal success rate mechanism. Then*

$$\gamma^*_n = 1 - \left(\frac{q_n}{\alpha_n(1-\alpha_n)^{n-1}} + \frac{p_n}{\alpha_n}\right)^{-1} \tag{6.5}$$

where $\alpha_n = p_n + q_n$.

Proof. Let again $A^*_+(t) \subseteq V_n$ denote the set of positive peers under the maximal success rate algorithm. Then, both $\{A^*_+(t)\}_{t\in\mathbb{R}_+}$ and its cardinality $\{|A^*_+(t)|\}_{t\in\mathbb{R}_+}$ (*i.e.* the number of positive peers) are ergodic Markov processes, whose transition rates can be seen in Figures 6.2 and 6.3, respectively. To see that both of these processes are Markovian, observe that $\{A^*(t)\}_{t\in\mathbb{N}}$ evolves independently of $\{G(t)\}_{t\in\mathbb{N}}$, and that the transitions of $\{|A^*(t)|\}_{t\in\mathbb{N}}$ depend only only on $|A^*(t)|$. Ergodicity is implied by our standard assumptions on p_n,q_n.

Let $\alpha_n = p_n + q_n$. Let ν_k be the steady state probability that there are k peers in the system. Solving the steady state equations for the Markov process in Figure 6.3 gives

$$\nu_0 = \left(\frac{q_n}{(1-q_n)\alpha_n(1-\alpha_n)^{n-1}} + \frac{p_n}{\alpha_n(1-q_n)}\right)^{-1} \tag{6.6a}$$

$$\nu_k = \frac{\binom{n}{k}\alpha_n^k(1-\alpha_n)^{n-k}}{1+\frac{p_n}{q_n}(1-\alpha_n)^{n-1}}, \text{ for } 1 \le k \le n. \tag{6.6b}$$

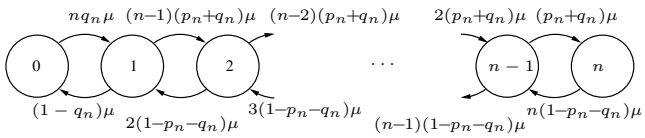


Figure 6.3: The Markov process $\{|A^*_+(t)|\}_{t\in\mathbb{R}_+}$, the number of positive peers under the maximal success rate algorithm.

By verifying that they satisfy the steady state equations, or simply by symmetry (as sets of the same size should have the same probability), we can show that the steady state probabilities of $\{A^*_+(t)\}_{t\in\mathbb{R}_+}$ are

$$\nu_A = \frac{1}{\binom{n}{|A|}}\nu_{|A|}, \qquad A \subseteq V_n \tag{6.7}$$

and thus can be derived from (6.6).

Consider now the embedded chain of the Markov process in Fig. 6.3. The original Markov process is uniformized [Gal96] (*i.e.*, the aggregate departure rate from a state is $n\mu$). The steady state probabilities of the embedded Markov chain are thus also given by (6.6). Failures occur at transitions from state 0 back to itself (searches that happened when no peers had the item), which happen with probability p_n, and transitions from state 1 to 0 (searches that originated from a peer that replaced the single peer that had the item), with probability p_n/n. Then the steady state probability of failure is equal to the probability that such transitions take place divided by the probability that a search takes place, namely p_n. The steady state probability of such a failed transition is equal by renewal theory to $\nu_0 p_n + \frac{1}{n}\nu_1 p_n$. (see *e.g.*, Gallager [Gal96, page 178]). The lemma then follows from equation (6.6). □

One can check from (6.5) that, if $q_n = e^{-n^2}$, for any p_n,

$$\lim_{n\to\infty} \gamma^*_n = 0.$$

Given that γ^*_n is an upper bound for the success rate of any mechanism the above has the following implication on the scalability of any mechanism in in $\mathcal{C}$:

Theorem 6.3. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is an ergodic churn-driven Markov chain. Then, no query propagation mechanism in $\mathcal{C}$ is scalable.* I.e.*, for every mechanism such that*

$$\mathbb{E}[C_{i,A,G} \mid \textit{failure}] = \omega(1),$$

there exist p_n, q_n such that

$$\rho_n = \omega(1).$$

Proof. Pick $q_n = e^{-n^2}$ so that, for any p_n,

$$\lim_{n\to\infty} \gamma_n^* = 0.$$

As a result, by Lemma 6.4, if a query propagation mechanism with

$$\mathbb{E}[C_{i,A,G} \mid \text{failure}] = \omega(1)$$

is used, the expected message cost per query will be

$$C_n \geq (1-\gamma_n^*)\mathbb{E}[C_{i,A,G} \mid \text{failure}] = \omega(1).$$

Thus, for p decaying slower than $1/\mathbb{E}[C_{i,A,G} \mid \text{failure}]$ (*e.g.*, for p_n constant), Lemma 6.2 implies that $\rho_n = \omega(1)$. □

An important observation is that our proof does not depend on the topology of the overlay graph: the only property of $\{G(t)\}_{t\in\mathbb{N}}$ used is its ergodicity, which we require for γ_n and ρ_n to exist *a.s.* In this sense, the proof simply relies on the fact that there is a "natural" upper bound on the success rate, irrespectively of the query propagation mechanism used and the evolution of the overlay graph.

6.3.3 Independent Graph Model and the Mean Field Limit Method

To prove Theorem 6.1 we will require a stronger assumption than the ergodicity of $\{G(t)\}_{t\in\mathbb{N}}$. In particular, we will require that the overlay graph evolves according to the independent graph model, as presented in Section 4.2.4. That is, $\{G(t)\}_{t\in\mathbb{N}}$ is a sequence of i.i.d. random variables sampled from a set $\mathbb{S}_{n,d} \subseteq \mathbb{MG}_{n,d}$ according to a given distribution. Here, we briefly discuss the advantage of making this assumption, from a modelling perspective.

To begin with, if $\{G(t)\}_{t\in\mathbb{N}}$ is an i.i.d. sequence, it is an ergodic, vertex-reversible (thus vertex-balanced) churn-driven chain and, therefore, by Lemma 6.1, the joint process $\{A_+(t), G(t)\}_{t\in\mathbb{N}}$ is ergodic. The most important benefit of assuming that $\{G(t)\}_{t\in\mathbb{N}}$ is an i.i.d. sequence is that it implies that the marginal process $\{A_+(t)\}_{t\in\mathbb{R}_+}$ is also a Markov process. Note that, as discussed in Section 6.3.1, this is not necessarily true in the general case. The embedded chain $\{A_+(t)\}_{t\in\mathbb{N}}$ of the marginal process is also ergodic, under our standard assumptions on p_n and q_n. In fact, it is simpler for our analysis to focus on the marginal process $\{A_+(t)\}_{t\in\mathbb{R}_+}$ and ignore the evolution of $\{G(t)\}_{t\in\mathbb{R}_+}$ altogether; the randomness of the overlay graph can be accounted for when describing the transition probabilities of the embedded chain $\{A_+(t)\}_{t\in\mathbb{N}}$.

Nonetheless, even if $\{A_+(t)\}_{t\in\mathbb{R}_+}$ is a Markov process, its stationary distribution is far from obvious, and highly depends on the query propagation mechanism. As in the general case, and contrary to a hybrid peer-to-peer system, peers are *not* positive independently of other peers in the system. This is one of the reasons that motivates the use of a mean field interaction model in our analysis, as we discuss below.

Mean Field Interaction Model

There is one more useful property that arises under the assumption that $\{G(t)\}_{t\in\mathbb{N}}$ is an i.i.d. sequence of graphs. Essentially, the rates of transitions between states do not depend on the actual sets $A_+(t)$ but on their cardinalities $|A_+(t)|$, *i.e.*, on the number of positive peers in the system. This implies that the Markov process is a *mean field interaction model*, as defined by Benaïm and Le Boudec [BL08], and described in more detail below.

Let $\vec{X}(t)$ be the characteristic vector of the sets $A_+(t)$, $t \in \mathbb{N}$. That is, $\vec{X}(t) \in \{0,1\}^n$, is a vector of size n indicating which peers are positive right after the t-th departure arrival epoch, *i.e.*:

$$X_i(t) = \begin{cases} 1, & \text{if } i \in A_+(t) \\ 0, & \text{o.w.} \end{cases} \text{ for all } t \in \mathbb{N}.$$

Note that, $\{\vec{X}(t)\}_{t\in\mathbb{N}}$ is an ergodic Markov chain over $\{0,1\}^n$: it is essentially a different representation the Markov chain $\{A_+(t)\}_{t\in\mathbb{N}}$. Let

$$P = [p_{\vec{x}\vec{x}'}]_{\vec{x},\vec{x}'\in\{0,1\}^n}$$

be the transition probability matrix of the chain $\{\vec{X}(t)\}_{t\in\mathbb{N}}$, *i.e.*,

$$p_{\vec{x}\vec{x}'} = \mathbf{P}(\vec{X}(j+1) = \vec{x}', \mid \vec{X}(j) = \vec{x}).$$

The embedded chain $\{\vec{X}(t)\}_{t\in\mathbb{N}}$ is called a *mean field interaction model* [BL08] if P is invariant under permutations. That is, for any permutation $\sigma : V_n \to V_n$, and for all $\vec{x}, \vec{y}, \vec{x}', \vec{y}' \in \{0,1\}^n$

$$p_{\vec{x}\vec{x}'} = p_{\hat{\sigma}(\vec{x})\hat{\sigma}(\vec{x}')}$$

where $\vec{y} = \hat{\sigma}(\vec{x})$ has coordinates

$$y_i = x_{\sigma^{-1}(i)}, \quad \text{for all } i \in V_n.$$

In our model, this property indeed holds for the hop-constrained random walk and the expanding ring, when the i.i.d. graphs $\{G(t)\}_{t\in\mathbb{N}}$ are label independent.

Lemma 6.6. *Assume that the i.i.d. sequence $\{G(t)\}_{t\in\mathbb{N}}$ is sampled from a label independent distribution. If the query propagation mechanism is the hop-constrained random walk, the delay-constrained random walk or the expanding ring, then $\{\vec{X}(t)\}_{t\in\mathbb{N}}$ is a mean field interaction model.*

Proof. Note first that the next transition happens at a peer chosen uniformly from V_n. Hence, the invariability of the kernel P under permutations holds if the new arriving peer publishes the data item or immediately becomes null. We therefore focus on transitions that are due to data item requests. We have that,

$$\mathbf{P}(\vec{X}(t+1)=\vec{x}' \mid \vec{X}(t)=\vec{x}) = \\ \sum_{G\in\mathbb{S}_{n,d}}\sum_{i\in V_n}\frac{1}{n}\mathbf{P}(\vec{X}(t+1)=\vec{x}' \mid \vec{X}(t)=\vec{x},\mathcal{G}=G,I(t+1)=i)\cdot\mathbf{P}(\mathcal{G}=G)$$

where $\mathcal{G}$ is a label-independent random graph in $\mathbb{S}_{n,d}$. Observe that, for any permutation $\sigma : V_n \to V_n$:

$$\mathbf{P}(\vec{X}(j+1)=\vec{x}' \mid \vec{X}(j)=\vec{x},\mathcal{G}=G,I(t+1)=i) = \\ \mathbf{P}(\vec{X}(j+1)=\hat{\sigma}(\vec{x}') \mid \vec{X}(j)=\hat{\sigma}(\vec{x}),\mathcal{G}=\hat{\sigma}(G),I(t+1)=\sigma(i))$$

where $\hat{\sigma}(G)$ the isomorphic graph to G induced by the permutation σ. For the random walk, this holds because any path followed by random walk on the "non-permuted" version has a one-to-one and onto correspondence to a path on the permuted version that occurs with the same probability (the latter holds because the random walker does not make a decision based on the index of the virtual peer it resides on). For the expanding ring, this holds because the propagation is deterministic, and does not depend on the labels of the vertices. On the other hand,

$$\mathbf{P}(\mathcal{G}=G)=\mathbf{P}(\mathcal{G}=\hat{\sigma}(G))$$

as $\mathcal{G}$ is label independent. These two observations, along with the fact that, by the definition of label independence, $\hat{\sigma}(\mathbb{S}_{n,d})=\mathbb{S}_{n,d}$, yield the lemma. □

An implication of Lemma 6.6 is that $\{\sum_{i=1}^{n} X_i(t)\}_{t\in\mathbb{N}} = \{|A_+(t)|\}_{t\in\mathbb{N}}$ is also a Markov chain, and the continuous-time process $\{|A_+(t)|\}_{t\in\mathbb{R}_+}$, is a Markov process [BL08].

The Mean Field Limit Method

As discussed above, although assuming that $\{G(t)\}_{t\in\mathbb{N}}$ is and i.i.d. sequence implies that $\{A_+(t)\}_{t\in\mathbb{R}_+}$ is a Markov process, the stationary distribution of this process is far from obvious. Nonetheless, knowing that $\{\vec{X}(t)\}_{t\in\mathbb{N}}$ is a mean field model gives us a method to compute

the stationary distribution for large enough n. The method we can use is the *mean field limit method*, and it is outlined by Benaïm and Le Boudec in [BL08]. The discussion below aims to give some intuition into what the mean field limit method is, and how it can be used to compute the query success rate in our system.

Let

$$M_n(t) = \frac{1}{n}\sum_{i=1}^{n} X_i(t) = \frac{1}{n}|A_+(t)|, \qquad t \in \mathbb{N}$$

be the fraction of positive peers in the system right after the t-th departure/arrival epoch. Recall from the previous section that $\{|A_+(t)|,\ t \in \mathbb{N}$ is a Markov chain and, therefore, so is $\{M_n(t)\}_{t\in\mathbb{N}}$. The mean field limit method indicates that, if $\{M_n(t)\}$ satisfies certain conditions, its trajectory can be arbitrarily close (as n tends to infinity) to the trajectory of a deterministic process. Although the system evolves in a random manner, its dynamics can be estimated asymptotically by a deterministic process

More precisely, let

$$F_n(m) = \mathbb{E}[M_n(t+1) - M_n(t) \mid M_n(t) = m]$$

be the *mean drift* of the process $\{M_n(t)\}_{t\in\mathbb{N}}$. Suppose that the following assumptions hold

A1 There exists an $\epsilon_n > 0$ and a function $F : [0,1] \to \mathbb{R}$ such that $\lim_{n\to\infty} \epsilon_n = 0$ and

$$\lim_{n\to\infty} \epsilon_n^{-1} F_n(m) = F(m)$$

uniformly in $[0,1]$.

A2 The function $\epsilon_n^{-1}F_n(m)$ is Lipschitz continuous uniformly in n. ***I.e.***, there exists a constant $L > 0$ such that, for all $m, m' \in [0,1]$,

$$\epsilon_n^{-1}|F_n(m) - F_n(m)| \le L|m - m'|$$

where L does not depend on n.

We define a continuous time process $x_n : \mathbb{R}_+ \to [0,1]$ in terms of the discrete time process $M_n : \mathbb{N} \to [0,1]$ as follows. Let $\tau_k = k\epsilon_n$, for $k \in \mathbb{N}$. Then, for all $k \in \mathbb{N}$,

$$x_n(\tau_k) = M_n(k), \text{ and}$$
$$x_n(\tau_k + s) = M_n(k) + s\frac{M_n(k+1) - M_n(k)}{\tau_{k+1} - \tau_k}, \text{ for } 0 < s < \frac{1}{n}.$$

Observe that x_n essentially evolves as M_n, only it does so at a different "time-scale": $M_n(t) = x_n(t\epsilon_n)$, for t an integer. On real intervals $[t\epsilon_n, (t+1)\epsilon_n]$, x_n is linearly interpolated between the values $M_n(t+1)$ and $M_n(t)$.

Theorem 1 of Benaïm and Le Boudec [BL08] implies that, if the above assumptions hold, $x_n(t)$ can be arbitrarily close, as n goes to infinity, to a process that evolves deterministically. In particular, we define the *flow* $\Phi : \mathbb{R} \times [0,1] \to [0,1]$, mapping $(\tau, m) \mapsto \Phi_\tau(m)$, as the solution of the following ODE

$$\Phi_0(m) = m, \qquad \frac{d\Phi_\tau(m)}{d\tau} = F(\Phi_\tau(m)) \tag{6.8}$$

where F is the asymptotic mean drift given by assumption A1. *I.e.*, $\Phi_\tau(m)$ is the solution of the ODE (6.8) with initial condition m. Assume that $x_n(0) = m$, for some $m \in [0,1]$. Then, Theorem 1 of Benaïm and Le Boudec [BL08] implies that

$$\lim_{n\to\infty} \mathbf{P}(\sup_{0\le\tau\le T} |x_n(\tau) - \Phi_\tau(m)|) = 0$$

for all $T > 0$. Hence, as n goes to infinity, the probability that the trajectory of x_n strays away from the trajectory of the solution of the ODE (6.8) (provided that they have the same initial state) goes to zero.

In other words, the mean field limit method allows us to describe the *dynamics* of our system with arbitrary precision. Though this is a powerful tool and a remarkable result on its own, we are more interested in the stationary distribution of the Markov process $\{M_n(t)\}_{t\in\mathbb{R}_+}$, rather than how it evolves through time. Interestingly, the mean field method gives us a way to compute the stationary distribution of $M^n(t)$, for large n, in terms of the stationary points of the ODE (6.8). In particular, assume that the equation

$$F(m) = 0$$

has a unique solution m^* in $[0,1]$. Then, if the fraction of positive peers in the system is m^*, the mean drift of the system will be zero, asymptotically. In this sense, m^* is an asymptotic "operating point" of the chain $M_n(t)$.

Let ν_n be the stationary distribution of M_n. Then, Corollary 2 of Benaïm and Le Boudec [BL08] implies that ν_n converges weakly to δ_{m^*}, the Dirac measure on m^*. *I.e.*, for any continuous function h on $[0,1]$,

$$\lim_{n\to\infty} \mathbb{E}[h(M_n)] = \lim_{n\to\infty} \int_{[0,1]} h(m)\nu_n(dm) = h(m^*) \tag{6.9}$$

In other words, the expectation (and the time-average, by ergodicity) of any quantity that can be expressed as a reward over M_n can be computed, for large n, by assuming that the fraction of peers in the system is always equal to the operating point m^*. It turns out that, because of the label-independence of $\{G(t)\}_{t\in\mathbb{N}}$, the probability that a random walk or an expanding ring successfully locates a positive peer can indeed be expressed as a function of the fraction of positive peers in the system. As a result, we can express γ_n, the query success rate, in terms of the operating point m^*, for large enough n.

The above discussion is meant to give some intuition as to what the mean field method is and how it can be used to compute the query success rate. The actual statements of the above results by Benaïm and Le Boudec [BL08] are proved for a far more general setting than the one we discussed. This will suffice for our analysis of the random walk and the expanding ring in this section. However, to prove the results for the random walk using evidence of absence appearing in Section 6.4, we will need to further extend the more results of Benaïm and Le Boudec. A formal discussion of the general model by Benaïm and Le Boudec, as well as the extension of the results appearing in [BL08], can be found in Appendix A.

6.3.4 Reliability

Theorem 6.3 implies that the random walk and the expanding ring propagation mechanism will be scalable only if TTL_n is bounded. However, queries with $\mathrm{TTL}_n = O(1)$ will reach very few peers, and are very likely to fail. In fact, the following theorem states that such a system will be very unreliable.

Theorem 6.4. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is an i.i.d. sequence of graphs sampled from a label-independent distribution. For a hop-constrained random walk search with $\mathrm{TTL}_n = O(1)$, queries are guaranteed to be successful,* i.e.*, we have*

$$\lim_{n\to\infty} \gamma_n = 1$$

for all possible values of p_n, only if

$$q_n = \Omega(1).$$

Moreover, the same statement holds for the expanding ring with $\mathrm{TTL}_n = O(1)$.

Proof. For the convenience of the reader, we first give a brief outline of the three main steps followed by our proof.

Step 1. We first show that the query success rate γ_n is an increasing function of p_n and q_n. Intuitively, the more peers request or bring the data item in the system, the more likely that any given peer stores it and, therefore, the more likely a given query succeeds. This is stated formally in Corollary 6.1 and proved through a simple coupling argument.

Step 2. The above result implies that, in the limit, the query success rate in a system where $q_n \to 0$ or $p_n \to 0$ will be no more than the query success rate of a system in which these are constant in n (say, equal to c_q and c_p, respectively). In the this case, we show that an upper bound on γ_n can be obtained as a solution of an equation involving c_q, c_p, and c_0, the maximum number of peers visited by a search. The latter quantity is constant in n, as $\mathrm{TTL}_n = O(1)$. The equation that characterizes γ_n is stated in Lemma 6.9, and is proved using the mean field method of Benaïm and Le Boudec [BL08].

Step 3. Our final step combines the previous two results as follows. First, we show that one can make γ_n arbitrarily small by taking $q_n = c_q$ to be small enough (but constant in n). As the query success rate of for any q_n that converges to zero must be lower (asymptotically) than the query success rate when q_n is constant, this implies that if $q_n \to 0$, then so does γ_n. The lemma then follows by contradiction, as any sequence $q_n = \Omega(1)$ will contain a subsequence that is $o(1)$.

Below, we present our proof, broken down in the above three main steps.

Step 1: Monotonicity in p_n, q_n. We thus begin with the first step in our proof, *i.e.*, showing that γ_n is increasing in p_n and q_n. We first obtain the following stochastic domination result, obtained through coupling: If either the publishing probability or the request probability of a data item increases, the item will become more widely available.

Lemma 6.7. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is an i.i.d. sequence, and consider a data item that is published with a probability q_n and requested with probability p_n. Let $\mathbf{P}_{p_n,q_n}(|A_+| > k)$, $k = 0, \ldots, n$, be probability that there are more than k positive peers in the system in steady state, under a hop-constrained random walk mechanism. Then, if $q'_n \geq q_n$ and $p'_n \geq p_n$,*

$$\mathbf{P}_{p'_n,q'_n}(|A_+| > k) \geq \mathbf{P}_{p_n,q_n}(|A_+| > k). \tag{6.10}$$

Moreover, the same statement holds for the expanding ring mechanism.

Proof. We prove the statement for the hop-constrained random walk mechanism, as the proof for the expanding ring is identical. We will use again a coupling argument, as in the proof of

Lemma 6.4. Note first that it suffices to prove (6.10) for the following two cases: (a) $q'_n \geq q_n$ and $p'_n = p_n$ and (b) $q'_n = q_n$ and $p'_n \geq p_n$.

We first prove (6.10) in case (a). Consider a $q'_n \geq q_n$, for all $n \in \mathbb{N}$. Construct a process $\{A'_+(t)\}_{t\in\mathbb{R}_+}$ coupled to $\{A_+(t)\}_{t\in\mathbb{R}_+}$ as follows. First, both processes start with the same set of positive peers at time 0. Whenever a departure/arrival event occurs in A_+, the same event occurs in A'_+. *I.e.*, whenever a peer $i \in V_n$ is replaced in A_+, so is the same peer i in A'_+. If the peer i requests or publishes the item in A_+ , it also does so in A'_+. Moreover, we assume that, if i requests the item, the same random walk conducted over A_+ is conducted over A'_+: that is, the graph sampled is the same in both cases, and the choices made by the random walker are also made by the walker in A'_+. However, the walk in A'_+ may stop earlier than A_+, if it encounters a positive peer not existing in A_+ (this can happen because $A_+(t) \subseteq A'_+(t)$, as we discuss below).

If i enters the system in A_+ and immediately becomes null (*i.e.*, neither publishes nor requests the item), then, in A'_+, one of the following two events can take place: (a) peer i publishes the item, with probability $\frac{q'_n - q_n}{1-(p_n+q_n)}$, or (b) it becomes null.

Again, these two Markov processes are defined in a joint probability space, and the joint process $\{A_+(t), A'_+(t)\}_{t\in\mathbb{R}_+}$ is a Markov process. The marginal chain $A'_+(t)$ evolves as a process with publishing probability q'_n and request probability p_n. Moreover, at any point in time, $A_+(t) \subseteq A'_+(t)$. Statement (6.10) therefore follows by ergodicity, as $\mathbf{P}_{p_n,q_n}(|A_+| > k)$ in steady state is equal to the time average of the function $\mathbb{1}_{|A_+|>k}$.

A similar coupling argument can be used to prove (6.10) under case (b), where by $q'_n = q_n$ and $p'_n \geq p_n$: we need to similarly "convert" some peers that are null in the original chain to peers that request the item in the coupled chain. If such conversions happen with probability $\frac{p'_n - p_n}{1-(p_n+q_n)}$, the paths of the coupled chain will have the desirable marginal distribution. [*Lemma 6.7*] □

An immediate corollary of Lemma 6.7 is that increasing either p_n or q_n will improve the reliability of the system. This concludes the first step of our proof.

Corollary 6.1. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is an i.i.d. sequence sampled from a label-independent distribution. Then, for both the hop-constrained random walk and the expanding ring, γ_n is increasing in p_n and q_n.*

Proof. The steady state probability that a query succeeds is can be expressed as

$$\gamma_n = \sum_{k=0}^{n} \gamma_k \pi_k$$

where π_k the steady state probability that there are k positive peers in the system and γ_k the probability that a query succeeds given that there are k positive peers in the system. By Lemma 6.7, if either p_n or q_n is increased, the number of positive peers in steady state stochastically dominates the number of positive peers in steady state under the previous values of p_n,q_n. On the other hand, γ_k is an non-decreasing function of k, and the corollary follows. [*Cor. 6.1*] □

Step 2: Constant p_n,q_n. Armed with the above result, we turn our attention to the second step in our proof, where we assume that $\mathrm{TTL}_n = O(1)$ and p_n and q_n are constant. More specifically, assume that there exist constants $c_{\mathrm{TTL}} \geq 1$, $c_p > 0$ and $c_q > 0$ such that

$$\mathrm{TTL}_n \leq c_{\mathrm{TTL}}, \quad p_n = c_p, \quad q_n = c_q.$$

An implication of the bound on TTL_n is that, for both the hop-constrained random walk and the expanding ring, there exists a constant c_0 such that the number of peers visited by the query propagation mechanism does not exceed c_0. For the random walk, $c_0 = c_{\mathrm{TTL}}$; for the expanding ring

$$c_0 = d^2 \frac{(d-1)^{c_{\mathrm{TTL}}} - 1}{d-2}$$

by Lemma 5.10.

Corollary 6.1 implies that the query success rate for a system in which $\lim_{n\to\infty} q_n = 0$ or $\lim_{n\to\infty} p_n = 0$ can only be lower than the success rate of a system in which both are constant. Thus, by characterizing the latter case, we can obtain an upper bound on the query success rate when q_n converges to zero.

Following the notation in Section 6.3.3, let

$$M_n(t) = \frac{1}{n}\sum_{i=1}^{n} X_i(t) = \frac{1}{n}|A_+(t)|, \qquad t = 0, 1, 2, \ldots$$

be the fraction of positive peers in the system right after the t-th departure/arrival epoch. Recall from Section 6.3.3 that $|A_+(t)|$, $t = 0, 1, \ldots$ is a Markov chain and, therefore, so is $M_n(t)$.

Denote by $s_n(m)$ the probability that, if a fraction m of all peers are positive when a query is initiated, the random walk reaches a positive peer. Then, $s_n(m)$ can be upper-bounded as follows:

Lemma 6.8.

$$s_n(m) \leq \begin{cases} 1 - (1 - m\frac{1}{1-\frac{c_0}{n}})^{c_0}, & m < 1 - \frac{c_0}{n} \\ 1, & m \geq 1 - \frac{c_0}{n} \end{cases} \tag{6.11}$$

Proof. To see this, suppose that the number of positive peers is $k = nm$. To begin with, observe that a random walk that visits j peers (excluding the source peer that initiated the query), where $j > (n-1) - k$, will visit at least one positive peer. Consider the case where a random walk visits $j \leq (n-1) - k$ peers, excluding the source peer. Recall that the overlay graph is label-independent, and that the random walk ignores peer indices. Therefore, given that k out of the $n-1$ peers excluding the source are positive, the probability that, if the random walk that visits j peers, none of them are positive, is

$$\frac{\binom{(n-1)-k}{(n-1)-k-j}}{\binom{n-1}{(n-1)-j}} = \frac{\frac{((n-1)-k)!}{((n-1)-k-j)!j!}}{\frac{(n-1)!}{((n-1)-j)!j!}}$$
$$= \frac{((n-1)-k)((n-1)-k-1)\cdots((n-1)-k-(j-1))}{(n-1)(n-2)\ldots((n-1)-(j-1))}$$
$$= (1-\frac{k}{n-1})(1-\frac{k}{n-2})\ldots(1-\frac{k}{n-j}) \geq (1-\frac{k}{n-j})^j$$

Finally, to obtain (6.11), observe that a random walk with $\mathrm{TTL} \leq c_0$ can visit at most c_0 peers, excluding the source. [Lemma (6.8)] □

As in Lemmas 6.4 and 6.7, we can use a coupling argument to construct a process $\bar{M}_n(t)$ in which the probability a query succeeds is given by the r.h.s. of (6.11), and this process satisfies

$$\bar{M}_n(t) \geq M_n(t), \quad t = 0, 1, \ldots.$$

In particular, this process evolves as $M_n(t)$, unless a query fails. If, when the failed query was issued, the fraction of positive peers was m, the query in $\bar{M}_n(t)$ will succeed with probability $\frac{\bar{s}_n(m)-s_n(m)}{1-s_n(m)}$, where $\bar{s}_n(m)$ the r.h.s. of (6.11). Moreover, as in Corollary 6.1, if γ_n, $\bar{\gamma}_n$ the steady state probability a query succeeds in $M_n(t)$ and $\bar{M}_n(t)$ respectively, then

$$\gamma_n \leq \bar{\gamma}_n.$$

Therefore, to obtain an upper bound on γ_n, we can focus on the chain $\bar{M}_n(t)$ and the behaviour of $\bar{\gamma}_n$. The following lemma concludes the second step of our proof, as it characterizes the limit of $\bar{\gamma}_n$ as the solution of an equation involving c_q, c_p and c_0.

Lemma 6.9. *Let $\bar{\gamma}_n$ be the steady state probability that a query locates a positive peer in $\bar{M}_n$. Then,*

$$\lim_{n\to\infty} \bar{\gamma}_n = 1 - (1-m^*)^{c_0} \tag{6.12}$$

where $m^* \in [0,1]$ satisfies

$$c_q + c_p(1-(1-m^*)^{c_0}) - m^* = 0. \tag{6.13}$$

Proof. Our proof relies on the analysis of Benaïm and Le Boudec [BL08]. In particular, we show that $\bar{M}_n$ can be characterized in terms of a deterministic process for large enough n. This allows us to relate the steady state behaviour of $\bar{M}_n$ to the stationary points of the deterministic process, yielding the lemma.

We define the mean drift of $\bar{M}_n$ as

$$F_n(m) = \mathbb{E}[\bar{M}_n(t+1) - \bar{M}_n(t) \mid \bar{M}_n(t) = m], \qquad m \in [0,1] \tag{6.14}$$

i.e., $F_n(m)$ is the expected change in the fraction of positive peers at the next transition, conditioned on the fact that the fraction of positive peers is m. This is

$$\begin{aligned} F_n(m) &= \frac{1}{n}\left[(+1)\cdot(1-m)[c_q + c_p\bar{s}_n(m)] + (-1)\cdot m(1-c_q-c_p\bar{s}_n(m-\frac{1}{n}))\right] \\ &= \frac{1}{n}\left[c_q + (1-m)\bar{s}_n(m) + m\bar{s}_n(m-\frac{1}{n}) - m\right] \end{aligned} \tag{6.15}$$

as the probability that a search is initiated by a (formerly) positive peer is m.

The following then holds:

Lemma 6.10. *1.* $\lim_{n\to\infty} \dfrac{F_n(m)}{\frac{1}{n}} = F(m) = c_q + c_p(1-(1-m)^{c_0}) - m$ *uniformly in* $[0,1]$.

2. *The number of peers that may change their state (to positive or null) at each transition is no more than a constant (actually, 1).*

3. *If* $c_0 > 1$, $nF_n(m)$ *is Lipschitz continuous uniformly in* n, i.e.*, for all* $m, m' \in [0,1]$

$$|n(F_n(m) - F_n(m'))| \leq L|m - m'|$$

where L *does not depend on* n*, for large enough* n.

Proof. 1. The point-wise convergence is follows immediately from (6.15) and the definition of $\bar{s}_n(m)$ in (6.11). To prove that the convergence is uniform, it suffices to show that $\bar{s}_n(m)$ converges uniformly to

$$\bar{s}(m) = 1 - (1-m)^{c_0}$$

The statement then follows as the mean drift can be expressed as the sum and/or composition of functions that converge uniformly.

Let $\xi = 1 - \frac{c_0}{n}$. For $m \in [0, \xi]$,

$$|\bar{s}_n(m) - \bar{s}(m)| = |(1-m)^{c_0} - (1 - m\frac{1}{\xi})^{c_0}| \leq C(c_0)m|1 - \frac{1}{\xi}| \overset{m\leq\xi}{\leq} C(c_0)\frac{c_0}{n}.$$

where $C(c_0)$ a constant depending only on c_0, because as $(1-x)^{c_0}$ has a bounded derivative in $[0, 1]$. For $m \in [\xi, 1]$

$$|\bar{s}_n(m) - \bar{s}(m)| = (1-m)^{c_0} \leq \left(\frac{c_0}{n}\right)^{c_0}.$$

Hence,

$$\max_{m\in[0,1]} |\bar{s}_n(m) - \bar{s}(m)| \leq \max(C(c_0)\frac{c_0}{n}, \left(\frac{c_0}{n}\right)^{c_0})$$

and the statement follows.

2. It is obvious.

3. It suffices to show that $\bar{s}_n(m)$ is Lipschitz uniformly on n. Observe that $\bar{s}_n(m)$ is everywhere differentiable for $c_0 > 0$: the only "knee" might appear at $m = 1 - \frac{c_0}{n}$, but

$$\frac{d}{dm}\left[1 - (1 - m\frac{1}{1-\frac{c_0}{n}})^{c_0}\right] = \frac{c_0}{1-\frac{c_0}{n}}(1 - m\frac{1}{1-\frac{c_0}{n}})^{c_0-1},$$

which is zero at $m = 1 - \frac{c_0}{n}$, provided that $c_0 > 1$. On the other hand, the above also implies that the magnitude of the derivative is bounded by $\frac{c_0}{1-\frac{c_0}{n}} \leq 2c_0$ for $n > 2c_0$, therefore Lipschitz continuity follows by the mean value theorem. Uniformity is implied by the fact that the above bound on the derivative does not depend on n for large enough n. [*Lemma 6.10*] □

The above lemma implies that chain $\bar{M}_n$ satisfies assumptions H1a,H2a and H3a of Benaïm and Le Boudec [BL08]. As a result, Theorems 1 and 3 of Benaïm and Le Boudec apply. In particular, $\bar{M}_n$ can be approximated by a deterministic process whose dynamics are determined by

$$F(m) = c_q + c_p(1 - (1-m)^{c_0}) - m. \tag{6.16}$$

More precisely, we define a continuous time process $x_n : \mathbb{R}_+ \to [0, 1]$ in terms of the discrete time process $\bar{M}_n : \mathbb{N} \to [0, 1]$ as follows. Let $\tau_k = \frac{k}{n}$, for $k \in \mathbb{N}$. Then, for all $k \in \mathbb{N}$,

$$x_n(\tau_k) = \bar{M}_n(k), \text{ and}$$
$$x_n(\tau_k + s) = \bar{M}_n(k) + s\frac{\bar{M}_n(k+1) - \bar{M}_n(k)}{\tau_{k+1} - \tau_k}, \text{ for } 0 < s < \frac{1}{n}.$$

Observe that x_n essentially evolves as $\bar{M}_n$, only it does so at a different "time-scale": $\bar{M}_n(t) = x_n(\frac{t}{n})$, for t an integer. On real intervals $[t\frac{1}{n}, (t+1)\frac{1}{n}]$, x_n is linearly interpolated between the values $\bar{M}_n(t+1)$ and $\bar{M}_n(t)$.

Lemma 6.10 and Theorem 1 of Benaïm and Le Boudec [BL08] imply that $x_n(t)$ can be arbitrarily close, as n goes to infinity, to a process that evolves deterministically according to (6.16). In particular, we define the *flow* $\Phi : \mathbb{R} \times [0,1] \to [0,1]$, mapping $(\tau, m) \mapsto \Phi_\tau(m)$, as the solution of the following ODE

$$\Phi_0(m) = m, \qquad \frac{d\Phi_\tau(m)}{d\tau} = F(\Phi_\tau(m)) \tag{6.17}$$

where F is the asymptotic mean drift given by (6.16). *I.e.*, $\Phi_\tau(m)$ is the solution of the above ODE with initial condition m. Then, Lemma 6.10 and Theorem 1 of Benaïm and Le Boudec imply the following:

Lemma 6.11. *For all $T > 0$ there exist constants $C_1(T)$, $C_2(T)$ and a random variable $B_n(T)$ in [0,1] such that*

$$\sup_{0 \le \tau \le T} |x_n(\tau) - \Phi_\tau(m)| \le C_1(T)(B_n(T) + |x_n(0) - m|)$$

where

$$\mathbb{E}[|B_n(t)|^2] \le \frac{C_2(t)}{n}.$$

The above implies (through Chebychev's inequality), that, as n goes to infinity, the probability that the trajectory of x_n strays away from the trajectory of the ODE (6.17) (provided that they have the same initial state) goes to zero.

The above result can be used to obtain a characterization of the steady state probability of $\bar{M}$ in terms of the asymptotic behaviour of the flow Φ_τ (as τ goes to infinity). In particular, Corollary 2 of Benaïm and Le Boudec [BL08] (or, Corollary 3.2 in Benaïm [Ben98]) implies the following:

Lemma 6.12. *Let ν_n be the steady state distribution of $\bar{M}_n$. Then, ν_n converges weakly to δ_{m^*}, the Dirac measure on m^*, where $m^* \in [0,1]$ the unique solution of the equation*

$$F(m) = 0, \quad m \in [0,1],$$

and F is given by (6.16)*. In particular, for any continuous function h on $[0,1]$,*

$$\lim_{n\to\infty} \mathbb{E}[h(\bar{M}_n)] = \lim_{n\to\infty} \int_{[0,1]} h(m)\nu_n(dm) = h(m^*) \tag{6.18}$$

Proof. By Theorem 3 of Benaïm and Le Boudec [BL08] states that any weak limit point of the steady state probabilities ν_n is has a compact support, included on the Birkhoff center of the flow Φ. For any $c_0 > 1$, $c_p > 0$, $c_q > 0$, equation

$$F(m) = 0$$

has a unique solution m^* in $[0, 1]$. Moreover, it is strictly positive in $[0, m^*)$ and strictly negative in $(m^*, 1]$. As a result, Φ_τ has a unique stationary point, namely m_*, and the only invariant measure of the flow Φ_τ is the Dirac measure on m^*. The lemma therefore follows from Corollary 2 of Benaïm and Le Boudec [BL08]. [*Lemma 6.12*] □

Lemma 6.12 can be used to describe the steady state probability that a query succeeds in $\bar{M}_n$. In particular, we have

$$\bar{\gamma}_n = \int \left[m\bar{s}_n(m - \frac{1}{n}) + (1-m)\bar{s}_n(m) \right] \nu_n(dm).$$

The function

$$m\bar{s}_n(m - \frac{1}{n}) + (1-m)\bar{s}_n(m)$$

converges uniformly to

$$\bar{s}(m) = 1 - (1-m)^{c_0}.$$

Thus, for all $\epsilon > 0$, and for large enough n,

$$\left| \bar{\gamma}_n - \int \bar{s}(m)\nu_n(dm) \right| < \epsilon/2.$$

On the other hand, by Lemma 6.12, for large enough n

$$\left| \int \bar{s}(m)\nu_n(dm) - \bar{s}(m^*) \right| < \epsilon/2.$$

Hence,

$$\lim_{n\to\infty} \bar{\gamma}_n = \bar{s}(m^*).$$

which proves Lemma 6.9. [*Lemma 6.9*] □

Step 3: Completing the proof. The third (and final) step of our proof is given below. Having shown Lemma 6.9, we can combine it with Corollary 6.1 to prove Theorem 6.4. To see this, for $c_0 \geq 1$, consider a system where $q_n = c_q$ and $p_n = c_p < \frac{1}{c_0}$. Let

$$f(c_q, m) = c_q + c_p(1 - (1-m)^{c_0}) - m.$$

Then,

$$\frac{\partial}{\partial m}\left[c_q + c_p(1-(1-m)^{c_0}) - m\right] = c_p c_0 (1-m)^{c_0-1} - 1.$$

The above quantity is negative for all $m \in [0,1]$, as $c_p c_0 < 1$. Thus, the implicit function theorem implies that m^*, that satisfies

$$f(c_q, m^*) = 0$$

can be expressed as $m^* = g(c_q)$, where g a continuous function. For $c_q = 0$, the m^* can easily be seen to be equal to 0. Thus, the continuity of g implies that

$$\lim_{c_q \to 0} m^* = \lim_{c_q \to 0} g(c_q) = 0.$$

Suppose now that we have a system in which $\mathrm{TTL}_n \leq c_0$, $p_n = c_p < \frac{1}{c_0}$ and $\lim_{n\to\infty} q_n = 0$. Pick an arbitrary $\epsilon > 0$. Then, the steady state query success rate of the above system is upper bounded (by Lemma 6.7) by the query success rate of a system in which $q_n = \epsilon > 0$, for large enough n. By taking ϵ to small enough, we can make m^* to be arbitrarily close to 0, thus also making the asymptotic query success rate arbitrarily small (by (6.12)). This implies that in the above system, in which $\mathrm{TTL}_n = O(1)$ and $q_n = o(1)$, the query success rate is asymptotically upper-bounded by a number that is arbitrarily small; hence, it converges to zero. Theorem 6.4 therefore follows by contradiction, as for any sequence q_n that is not $\Omega(1)$ there exists a subsequence that is $o(1)$. [*Thm. 6.4*] □

An important observation is that our proof of Thm. 6.4 relies on the underlying topology only insofar as the graph sampled during searches is label-independent. Moreover, the only properties of the hop-constrained random walk and the expanding ring used are the following two: First, these mechanisms ignore peer labels, and second, if TTL_n is constant, then no more than a constant number of peers can be reached by the mechanism. In this sense, the same proof would apply to any search mechanism that exhibits the above two properties.

The condition that $q_n = \Omega(1)$ requires that the expected number of peers that publish the item grows linearly with the system size n, in order to guarantee that queries succeed. This indeed means that the system is very unreliable. For example, suppose that $q_n = 1/n^{0.1}$. Then, a large number of peers bring the item into the system: the expected number of peers that publish the item *grows* as $n^{0.9}$, almost linearly in n. However, even though the number of peers guaranteed to have the item is so high, the random walk mechanism with a bounded TTL_n cannot locate the item reliably.

Note that the condition $q_n = \Omega(1)$ is a necessary condition for $\lim_{n\to\infty} \gamma_n = 1$, but it is not sufficient. In particular, it is possible that even if q_n is equal to a constant, the query success rate does not converge to one — it may, in fact, converge to non-zero probability other than one.

6.3.5 Proof of Theorem 6.1

To prove Theorem 6.1, recall first that Theorem 6.3 holds for any mechanism that incurs an unbounded expected message cost, in the worst case (*i.e.*, when a query fails to locate the item). The random walk and the expanding ring are two such mechanisms, and the following corollary thus holds:

Corollary 6.2. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is an ergodic churn-driven Markov chain. Then, the hop-constrained random walk, the delay-constrained random walk and the expanding ring incur a bounded average load per peer*

$$\rho_n = O(1)$$

for all possible functions p_n and q_n if and only if

$$\mathrm{TTL}_n = O(1).$$

Proof. Recall that for both the hop-constrained and the delay-constrained random walk

$$\mathbb{E}[C_{i,A,G} \mid \text{failure}] = \mathrm{TTL}_n. \tag{6.19}$$

For the expanding ring mechanism,

$$\mathrm{TTL}_n < \mathbb{E}[C_{i,A,G} \mid \text{failure}] \leq d^2 \frac{(d-1)^{\mathrm{TTL}_n} - 1}{d-2} \tag{6.20}$$

where the upper-bound follows from Lemma 5.10. Note that, from (6.19) and the lower bound in (6.20), all three mechanisms satisfy the conditions of Theorem 6.3 for $\mathrm{TTL}_n = \omega(1)$.

Suppose first that the sequence TTL_n is not $O(1)$. Then, it contains a subsequence that is $\omega(1)$. Hence, from Theorem 6.3, there exist p_n, q_n for which ρ_n is unbounded. This proves the "only if" direction.

On the other hand, if $\mathrm{TTL}_n = O(1)$, (6.19) and (6.20) imply that $\mathbb{E}[C_{i,A,G} \mid \text{failure}] = O(1)$. Thus, from Lemma 6.2, ρ_n is bounded for all possible functions p_n and q_n, and the "if" direction also holds. □

Theorem 6.3 therefore implies that, in order to bound the average load per peer under a random walk or an expanding ring, TTL_n has to be of the order $O(1)$.

On the other hand, Theorem 6.4 implies that a system in which $\mathrm{TTL}_n = O(1)$ cannot be reliable. Note that the assumption on $\{G(t)\}_{t\in\mathbb{N}}$ appearing in Theorem 6.4 is stronger than the assumption in Theorem 6.3, because every independent graph model is also an ergodic churn-driven Markovian model. As a result, Theorems 6.3 and 6.4 imply Theorem 6.1: the hop-constrained random walk and the expanding ring cannot be both scalable and reliable, under the definitions we gave in Section 6.1.2. To achieve a bounded query traffic load per peer, the TTL value has to be so low that most queries fail, even for data items that are widely available (see also our discussion in Section 1.1.1).

6.4 Random Walk Using Evidence of Absence

In this section we present the proof of our second main result, namely Theorem 6.2. The theorem states that a random walk mechanism using evidence of absence is both scalable and reliable.

The remainder of this section is structured as follows. We first show that the random walk using evidence of absence is scalable (Section 6.4.1). This result holds under the general churn-driven Markovian graph model and is a consequence of the following very useful fact: the union of positive and negative peers (*i.e.*, peers that have the file and peers that believe it is absent, respectively) in the pure peer to peer system evolves in an identical manner as the set of positive peers in a hybrid system.

We then strengthen our assumption on the overlay graph: we restrict our model to the independent graph model, and discuss the implications of this assumption (Section 6.4.2). Under this model, we show that the random walk using evidence of absence is reliable (Section 6.4.3). Finally, combining the scalability result with our result on reliability yields Theorem 6.2 (Section 6.4.4).

As in our analysis of the random walk and the expanding ring, the most technically challenging part of this proof is our result on reliability; we will again employ the mean field limit method of Benaïm and Le Boudec [BL08].

6.4.1 Equivalence to a Hybrid System and Scalability

To prove that our proposed mechanism is scalable, we first focus the evolution of the pure peer-to-peer system under the assumption that $\{G(t)\}_{t\in\mathbb{R}_+}$ is a general churn-driven Markovian graph process. Recall that some peers in the system, which we call negative, believe that the item is not present in the system. Such peers respond to queries for the item by letting the source peer know of the item's absence, thereby converting it to a negative status.

In general, at any point in time, a peer in the system can be either positive, negative, or null. We denote by $A_+(t) \subseteq V$ and $A_-(t) \subseteq V$ the sets of positive and negative peers at time t. Then, under the assumption that $\{G(t)\}_{t\in\mathbb{R}_+}$ is a churn-driven Markovian process with state space $\mathbb{S}_{n,d} \subseteq \mathbb{MG}_{n,d}$, as described in Section 4.2.1, the process $\{A_+(t), A_-(t), G(t)\}_{t\in\mathbb{R}_+}$ is a Markov process whose state space is a subset of $2^{V_n} \times 2^{V_n} \times \mathbb{S}_{n,d}$ (note that $A_+(t) \cap A_-(t) = \emptyset$); transitions happen at Poisson epochs with rate $n\mu$, and depend only on the previous state.

We denote the embedded Markov chain of the joint process by $\{A_+(t), A_-(t), G(t)\}_{t\in\mathbb{N}}$. As the Markov process is uniformized, a stationary distribution of the embedded chain will also be a stationary distribution of the Markov process.

Conditioned on $\{G(t)\}_{t\in\mathbb{N}}$, the transitions of $\{A_+(t), A_-(t)\}_{t\in\mathbb{R}_+}$ are as follows. At each simultaneous departure/arrival event, the peer being replaced according to the churn process brings a copy of the item in the system with probability q_n. With probability p_n, it initiates a search: A query is propagated over a graph among the peers in V_n, sampled according to the distribution ν_G from all graphs $G \in \mathbb{S}_{n,d}$. According to the outcome of the search, the peer becomes either positive or negative. Finally, with probability $1 - p_n - q_n$, the peer simply becomes null.

Just as in the case where evidence of absence was not used, the marginal process $\{A_+(t), A_-(t)\}_{t\in\mathbb{R}_+}$ may not necessarily be Markovian, and the stationary distribution $\{A_+(t), A_-(t), G(t)\}_{t\in\mathbb{R}_+}$ does not have a simple product form. Moreover, the event that a given peer is positive is not independent of the event that other peers are positive, and the same holds for the event that a peer is negative. Finally, contrary to the case where evidence of absence is not used, the ergodicity of $\{G(t)\}_{t\in\mathbb{N}}$ does not necessarily imply the ergodicity of the joint chain.

Nonetheless, a system using evidence of absence, irrespectively of the query propagation mechanism used, has an intrinsic similarity to a hybrid system. For all $t \in \mathbb{R}_+$, denote by

$$A(t) = A_+(t) \cup A_-(t)$$

the set of peers that are either positive or negative. Then, $\{A(t)\}_{t\in\mathbb{R}_+}$ evolves in an identical

manner as the set of positive peers in a hybrid peer to peer system with the same publishing and request probabilities. In particular, $\{A(t)\}_{t\in\mathbb{R}_+}$ is a Markov process, irrespectively of the query propagation mechanism used and the topology of the overlay graph, and satisfies the following equivalent of Corollary 5.1 and Lemma 5.3:

Lemma 6.13. *Assume that the marginal Markov chain $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic. Then, the joint chain $\{A(t), G(t)\}_{t\in\mathbb{R}_+}$ is ergodic, and has a unique stationary distribution given by*

$$\nu_{A,G} = \nu_A \cdot \nu_G, \qquad A \subseteq V_n, G \in \mathbb{S}_{n,d}$$

where ν_G the stationary distribution of the marginal chain $\{G(t)\}_{t\in\mathbb{N}}$ and ν_A the stationary distribution of the marginal chain $\{A(t)\}_{t\in\mathbb{N}}$, given by

$$\nu_A = (p_n + q_n)^{|A|}(1 - p_n - q_n)^{n-|A|}, \quad A \subseteq V_n. \tag{6.21}$$

The proof is identical to the proofs of Corollary 5.1 and Lemma 5.3. The above property is very useful, as it allows us to duplicate all of the results appearing in Chapter 5, in the context of a hybrid system, for any property or performance metric of our pure peer-to-peer system that can be expressed through a reward function on the process $\{A(t), G(t)\}_{t\in\mathbb{R}_+}$, as opposed to the process $\{A_+(t), A_-(t), G(t)\}_{t\in\mathbb{R}_+}$.

The average traffic load per peer and the query response time are two such metrics. As a result, all statements regarding the above two metrics that appear in Chapter 5 with respect to the random walk mechanism extend to the random walk in a pure system using evidence of absence. In fact, the following theorem follows immediately from Theorem 5.10.

Theorem 6.5. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is an ergodic and vertex balanced churn-driven Markov chain, and that there exists a constant $\bar{\tau}$ such that*

$$\mathbf{P}(\tau_n \leq \bar{\tau}) = 1 - o\left(\frac{1}{n}\right),$$

where τ_n the relaxation time of a graph sampled from the stationary distribution of $\{G(t)\}_{t\in\mathbb{N}}$. Then, the average traffic load per peer generated by a delay-constrained random walk with $\mathrm{TTL}_n = \Theta(n)$ that uses evidence of absence is bounded in n, irrespectively of p_n and q_n.

Hence, if the overlay graph is an expander with high probability, the pure peer-to-peer system that uses evidence of absence is scalable, according to the definition we gave in Section 6.1.2.

Note that not only the results on the random walk but also the results on the expanding ring can also be extended to the pure peer-to-peer case. In particular, Theorem 5.2 and Lemma 5.13 give a non-constant, but slowly growing, bound on the average traffic load per peer under the expanding ring mechanism using evidence of absence.

6.4.2 Independent Graph Model

Like Theorem 6.4, Theorem 6.6 is proved under the assumption that the overlay graph evolves according to the independent graph model, as presented in Section 4.2.4. That is, $\{G(t)\}_{t\in\mathbb{N}}$ is a sequence of i.i.d. random variables sampled from a set $\mathbb{S}_{n,d} \subseteq \mathbb{MG}_{n,d}$ according to a given distribution.

Similarly to the case presented in Section 6.3.3, if $\{G(t)\}_{t\in\mathbb{N}}$ is an i.i.d. sequence, the marginal process $\{A_+(t), A_-(t)\}_{t\in\mathbb{N}}$ is Markov process; again, this does not necessarily hold in the general case, as discussed in Section 6.4.1. Moreover, under our standard assumptions on p_n and q_n, $\{A_+(t), A_-(t), G(t)\}_{t\in\mathbb{N}}$ is ergodic. We will again focus on the marginal process $\{A_+(t), A_-(t)\}_{t\in\mathbb{R}_+}$ and ignore the evolution of $\{G(t)\}_{t\in\mathbb{R}_+}$ altogether; we account for $G(t)$ when describing the transition probabilities of the marginal chain $\{A_+(t), A_-(t)\}_{t\in\mathbb{N}}$. As in the case without evidence of absence, the stationary distribution of $\{A_+(t), A_-(t)\}_{t\in\mathbb{N}}$ (and, by uniformity, of $\{A_+(t), A_-(t)\}_{t\in\mathbb{R}_+}$) is far from obvious, and highly depends on the query propagation mechanism.

The Markov process $\{A_+(t), A_-(t)\}_{t\in\mathbb{N}}$ is also a mean field interaction model, according to the definion of Benaïm and Le Boudec [BL08]. In particular, let $\vec{X}(t)$ be a characteristic vector representing the distinct sets $A_+(t)$ and $A_-(t)$, $t \in \mathbb{N}$. That is, let $\vec{X}(t) \in \{\text{ø}, +, -\}^n$, be a vector of size n representing whether each peer is null, positive or negative right after the t-th departure arrival epoch, *i.e.*:

$$X_i(t) = \begin{cases} +, & \text{if } i \in A_+(t) \\ -, & \text{if } i \in A_-(t), \text{ and} \\ \text{ø}, & \text{o.w.} \end{cases} \qquad \text{for all } t \in \mathbb{N}.$$

Then $\{\vec{X}(t)\}_{t\in\mathbb{N}}$ is an ergodic Markov chain over $\{\text{ø}, +, -\}^n$. Let

$$P = [p_{\vec{x}\vec{x}'}]_{\vec{x},\vec{x}'\in\{\text{ø},+,-\}^n}$$

be the transition probability matrix of the chain $\{\vec{X}(t)\}_{t\in\mathbb{N}}$, *i.e.*

$$p_{(\vec{x},\vec{x}')} = \mathbf{P}(\vec{X}(j+1) = \vec{x}' \mid \vec{X}(j) = \vec{x}).$$

Recall that the embedded chain $\{\vec{X}(t)\}_{t\in\mathbb{N}}$ is a mean field interaction model [BL08] if P is invariant under permutations. That is, for any permutation $\sigma : V_n \to V_n$, and for all $\vec{x}, \vec{y}, \vec{x}', \vec{y}' \in \{0,1\}^n$

$$p_{\vec{x}\vec{x}'} = p_{\hat{\sigma}(\vec{x})\hat{\sigma}(\vec{x}')}$$

where $\vec{y} = \hat{\sigma}(\vec{x})$ has coordinates

$$y_i = x_{\sigma^{-1}(i)}, \quad \text{for all } i \in V_n.$$

The following lemma then holds:

Lemma 6.14. *Assume that the i.i.d. sequence $\{G(t)\}_{t\in\mathbb{N}}$ is sampled from a label independent distribution. If the query propagation mechanism is the delay-constrained random walk, then $\{\vec{X}(t)\}_{t\in\mathbb{N}}$ is a mean field interaction model.*

The proof is identical to the proof of Lemma 6.6 in Section 6.3.3. Again, the above lemma implies that we can use the mean field limit method to compute fraction of positive and negative peers in the system in steady state:

$$M_n^+(t) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{(X_i(t)=+)} = \frac{1}{n}|A_+(t)|, \quad M_n^-(t) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{(X_i(t)=-)} = \frac{1}{n}|A_-(t)|,$$

where $t \in \mathbb{N}$. In particular, the mean field limit method of [BL08] gives us a way to approximate the trajectory of the process $\{M_n^+(t), M_n^-(t)\}_{t\in\mathbb{N}}$ by a deterministic flow, that can be obtained through the solution of a system of ODEs. In turn, the stationary distribution of the above chain can computed, asymptotically, in terms of the stationary points of the above ODE; this is the methodology that we follow in order to compute asymptotic behaviour of the query success rate γ_n in the next section.

6.4.3 Reliability

We will now show that, if the item is reliably brought into the system, the query success rate converges to one. In the remainder of this section, we make the following three implicit assumptions: first, we assume that the query propagation mechanism is a delay-constrained random walk with $\mathrm{TTL}_n = \Theta(n)$. Second, we assume that $\{G(t)\}_{t\in\mathbb{N}}$ is a sequence of i.i.d. random graphs sampled from some set $\mathbb{S}_{n,d} \subseteq \mathbb{MG}_{n,d}$ according to a label-independent distribution. Third, we assume that this distribution is such that $G(t)$ is an expander-graph with high

probability. That is, if τ_n is the relaxation time of a graph sampled from $\mathbb{S}_{n,d}$, there exists a constant $\bar{\tau} \geq 1$ such that

$$\phi_n = \mathbf{P}(\tau_n \leq \bar{\tau}) = \mathbf{P}(\tau_{G(t)} \leq \bar{\tau}) = 1 - o\left(\frac{1}{n}\right).$$

We omit these three assertions from the statements of the lemmas appearing in this section, keeping in mind that all three are assumed to hold.

Our proof regarding the reliability of the random walk using evidence of absence consists of two parts. In the first part we show that, if $q_n = \omega\left(\frac{1}{n}\right)$, almost all queries find either a positive or a negative peer, and very few expire (because they exceeded the TTL). In the second part of our proof we show that, conditioned on the fact that a non-null peer is found, this peer will be a positive peer *a.a.s.* These two statements will then imply the reliability of the random walk using evidence of absence.

The proof of the first part relies on the fact that, most of the time, the fraction of peers that are either positive or negative is close to $p_n + q_n$.

Lemma 6.15. *Let γ_n^{any} be the steady state probability that a query reaches a non-null peer before it expires. If $q_n = \omega\left(\frac{1}{n}\right)$, then $\lim_{n\to\infty} \gamma_n^{any} = 1$.*

Proof. Let z be the fraction of peers that are either positive or negative (*i.e.*, that are non-null). From Lemma 6.13, in steady state, each peer is positive or negative with probability $p_n + q_n$, independently of other peers. Hence, in steady state, we have the following Chernoff bound:

$$\mathbf{P}\left(\left|z - \left(p_n + q_n\right)\right| > \delta(p_n + q_n)\right) \leq e^{-\Theta\left(\delta^2\right) n(p_n+q_n)} \tag{6.22}$$

for all $0 < \delta < 1$.

Let $s_n(z)$ be the probability that a query finds a non-null peer, conditioned on the fact that the fraction peers that are non-null at the time the query was initiated is $z \in [0, 1]$. Condition on the graph G sampled from $\mathbb{S}_{n,d}$. Then, as the query starts uniformly from outside the set of peers that are not positive or negative, by Lemma 3.2 this probability is no less than

$$1 - e^{-\frac{\mathrm{TTL}_n z}{\tau_G}}$$

where τ_G the relaxation time of the graph G sampled from $\mathbb{S}_{n,d}$. Hence, there exists a constant $\bar{\tau} \geq 1$ such that

$$s_n(z) \geq \phi_n(1 - e^{-\frac{\mathrm{TTL}_n z}{\bar{\tau}}}) \tag{6.23}$$

and

$$\phi_n = \mathbf{P}(\tau_n \leq \bar{\tau})$$

a function s.t. $\lim_{n\to\infty} \phi_n = 1$. Let π_z be the steady state probability that the fraction of positive or negative peers is z. The new query starts from a peer that replaced a positive or negative peer with probability z. Thus, we have that

$$\gamma_n^{any} = \sum_z \pi_z[(1-z))s_n(z) + zs_n(z-\frac{1}{n})]$$
$$\overset{(6.22),(6.23)}{\geq} \left(1-e^{-\Theta(\delta^2)n(p_n+q_n)}\right)\phi_n\left(1-e^{-\frac{\mathrm{TTL}_n\left((p_n+q_n)(1-\delta)-\frac{1}{n}\right)}{\tau_1}}\right)$$

For $\mathrm{TTL}_n = \Theta(n)$ and $p_n + q_n = \omega\left(\frac{1}{n}\right)$, the r.h.s. converges to one as n tends to infinity. □

The above lemma tells us that most queries will in fact locate a non-null peer. Let γ_n^+ be the steady state probability that, given that a query reaches a non-null peer, the peer reached is positive. The second part of our proof shows that γ_n^+ also converges to one.

Lemma 6.16. *Let γ_n^+ be the steady state probability that, given that a query reaches a non-null peer, the peer reached is positive. If $q_n = \omega\left(\frac{1}{n}\right)$, then*

$$\lim_{n\to\infty} \gamma_n^+ = 1.$$

We present the proof of Lemma 6.16 below. The main idea behind our argument is that γ_+ is proportional to the fraction of positive peers in the system. Hence, to prove our result, we study the dynamics of the positive peers in the system, and show that this stochastic process admits a mean field approximation, in the sense of Benaïm and Le Boudec [BL08]. This allows us to show that the fraction of positive peers over the total number of non-null peers converges to one, in expectation, as the number of peers increases. Thus, so does γ_n^+.

Lemmas 6.15 and 6.16 immediately imply that the random walk that uses evidence of absence is reliable, according to the definition we gave in Section 6.1.2.

Theorem 6.6. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is an* i.i.d. *sequence of graphs sampled from a label-independent distribution. Moreover, assume there exists a constant $\bar{\tau}$ such that*

$$\mathbf{P}(\tau_n \leq \bar{\tau}) = 1 - o\left(\frac{1}{n}\right),$$

where τ_n the relaxation time of a graph sampled from the (stationary) distribution of $\{G(t)\}_{t\in\mathbb{N}}$. Then the random walk search using evidence of absence is reliable. I.e.*, if the item is published sufficiently often, so that*

$$q_n = \omega\left(\frac{1}{n}\right)$$

then the item will be located reliably, i.e.,

$$\lim_{n\to\infty} \gamma_n = 1$$

for all p_n.

Proof. Follows immediately from Lemmas 6.15 and 6.16 as $\gamma_n = \gamma_n^+ \cdot \gamma_n^{any}$. □

Note that the restriction on q_n is much weaker than the restriction on q_n needed in Theorem 6.4 for the traditional random walk with a TTL to be reliable.

The remainder of this section is dedicated to the proof of Lemma 6.16. Because of its highly technical nature we begin with an outline of the major steps carried out.

Proof of Lemma 6.16. Our proof will follow the following four major steps:

Step 1. In the first step of our proof, we define two processes describing the evolution of positive and negative peers in our system. The first is the fraction of positive peers in the system over the total population of positive and negative peers. The second process we consider is the fraction of positive and negative peers, rescaled by a factor of $(p_n + q_n)^{-1}$. For both processes, we characterize in detail their mean drifts.

Note that the first process captures the probability that, given that a query takes place, and a positive or negative peer is successfully located, this peer will be a positive peer. Hence, the steady state behaviour of this process will characterize γ_n^+.

Step 2. Having defined the mean drifts of the above two processes, in the second step of our proof we consider the case where $q_n = \Omega(p_n)$. We then show that, in this case, the evolution of the system can be approximated by a mean field limit, in the sense of Benaïm and Le Boudec [BL08]. This allows us to characterize the weak limits of the stationary distributions of these two processes; in particular, we are able to show that the stationary distribution of the first process converges to a Dirac distribution on the value 1 —in other words, as n tends to infinity, almost all non-null peers are positive. This in turn implies that $\gamma_n^+ \to 1$.

Step 3. In the third step of our proof, we focus in the case $q_n = o(p_n)$. We again approximate the evolution of the first process and, as in the previous step, we show that the weak limit of its stationary distribution is the Dirac density on 1 and, as a result, $\gamma_n^+ \to 1$.

The difference from the second step is that, in this case, it is not possible apply the mean field approach of Benaïm and Le Boudec directly. The reason is that the two processes

we consider evolve at different timescales: the second process is essentially much faster than the first one. To address this, we use the averaging principle [Kif04], according to which the slow process can be approximated by a process that "sees" the fast process as if it were in steady state.

Step 4. Finally, we conclude our proof by showing that, since $\gamma_n^+ \to 1$ for the above two cases, it will converge to one for all possible values of q_n and p_n.

In the remainder of this section, we present the proof of Lemma 6.16, broken down into the above four major steps.

Step 1: Two processes describing our system. Following the notation of Section 6.4.2, for a system of size n, let

$$M_n^+(t) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{X_i(t)=+} = \frac{1}{n}|A_+(t)|, \quad M_n^-(t) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{X_i(t)=-} = \frac{1}{n}|A_-(t)|,$$

where $t \in \mathbb{N}$, be the fractions of positive and negative peer in the system, respectively, right after the t-th arrival/departure epoch. Recall that $G(t)$ is sampled from a label-independent distribution over $\mathbb{S}_{n,d}$, and that the random walk ignores the labels of virtual peers. This implies that queries that terminate because they locate a non-null peers are "unbiased": the probability that a query reaching a non-null peer reaches a positive peer is

$$\frac{M_n^+(t)}{M_n^+(t) + M_n^-(t)}.$$

For this reason, we turn our attention to how this quantity evolves in our system. We define

$$\hat{M}_n^+(t) = \frac{M_n^+(t)}{M_n^+(t) + M_n^-(t)} \quad \text{and} \quad \hat{M}_n^{\mathbb{Z}}(t) = \frac{M_n^+(t) + M_n^-(t)}{p_n + q_n} \tag{6.24}$$

where, if $M_n^+(t) + M_n^-(t) = 0$ we let $\hat{M}_n^+(t) = 0$, by definition. Then, $\{\hat{M}_n^+(t), \hat{M}_n^{\mathbb{Z}}(t)\}_{t\in\mathbb{N}}$ is a Markov chain whose state space is $[0,1] \times \mathbb{R}_+$.

Note that $\{\hat{M}_n^{\mathbb{Z}}(t)\}_{t\in\mathbb{N}}$ is also a Markov chain. In particular, we already know many things about this chain. First, we know its steady state distribution. By (6.21), the probability that there are k non-null peers in the system in steady state is binomial, given by

$$\pi_k = \binom{n}{k}\big(1 - (p_n + q_n)\big)^{n-k}\big(p_n + q_n\big)^k. \tag{6.25}$$

Moreover, we also know that the steady state probability that $\hat{M}_n^{\mathbb{Z}}(t) \in [1-\delta, 1+\delta]$, $0 < \delta < 1$, can be bounded by the Chernoff bound in (6.22).

The *mean drift* of the chain $\{\hat{M}_n^+(t), \hat{M}_n^{\mathbb{Z}}(t)\}$ is:

$$F_n^+(\hat{h}, \hat{z}) = \mathbb{E}\left[\hat{M}_n^+(t+1) - \hat{M}_n^+(t) \mid \hat{M}_n^+(t) = \hat{h}, \hat{M}_n^{\mathbb{Z}}(t) = \hat{z}\right]$$
$$F_{\mathbb{Z}}^n(\hat{h}, \hat{z}) = \mathbb{E}\left[\hat{M}_n^{\mathbb{Z}}(t+1) - \hat{M}_n^{\mathbb{Z}}(t) \mid \hat{M}_n^+(t) = \hat{h}, \hat{M}_n^{\mathbb{Z}}(t) = \hat{z}\right]$$

We begin by characterizing the transitions of chain $\{\hat{M}_n^+, \hat{M}_n^{\mathbb{Z}}\}$ by describing $\{\hat{M}_n^+(t+1), \hat{M}_n^{\mathbb{Z}}(t+1)\}$, conditioned on $\{\hat{M}_n^+(t), \hat{M}_n^{\mathbb{Z}}(t)\} = \{\hat{h}, \hat{z}\}$.

It is easier to describe these transitions in terms of $\{M_n^+(t), M_n^-(t)\} = \{h, g\}$ rather than $\{\hat{M}_n^+(t), \hat{M}_n^{\mathbb{Z}}(t)\} = \{\hat{h}, \hat{z}\}$. We will thus use h and g to denote the fractions of positive and negative peers in the system, respectively, keeping in mind that we can express $\hat{h}, \hat{z}$ in terms of h, g using (6.24). We will denote with $s_n(z)$ the probability that a query locates a non null peer, given that there are $h + g$ non-null peers at the time the query was initiated.

Given that

$$\left\{\hat{M}_n^+(t), \hat{M}_n^{\mathbb{Z}}(t)\right\} = \left\{\frac{h}{h+g}, \frac{h+g}{p_n+q_n}\right\},$$

the following transitions can take place:

1. A negative peer is replaced by a positive peer. Positive peers increase by one, while negative peers decrease by one. This occurs with probability

   $$g\left[q_n + p_n s_n (h + g - \frac{1}{n}) \frac{h}{h+g-\frac{1}{n}}\right]$$

   and leads to state

   $$\left\{\frac{h+\frac{1}{n}}{h+g}, \frac{h+g}{p_n+q_n}\right\}.$$

   The drift is thus

   $$\left\{+\frac{1}{n}\frac{1}{(h+g)}, 0\right\}.$$

2. A positive peer is replaced by a negative peer. Positive peers decrease by one, while negative peers increase by one. This occurs with probability

   $$h\left[p_n\big(1 - s_n(h + g - \frac{1}{n})\big) + p_n s_n (h + g - \frac{1}{n}) \frac{g}{h+g-\frac{1}{n}}\right]$$

   and leads to state

   $$\left\{\frac{h-\frac{1}{n}}{h+g}, \frac{h+g}{p_n+q_n}\right\}.$$

   The drift is thus

   $$\left\{-\frac{1}{n}\frac{1}{(h+g)}, 0\right\}.$$

3. A negative peer is replaced by a null peer. Negative peers decrease by one, while positive peers remain the same. This occurs with probability

$$g\left[1-(p_n+q_n)\right]$$

and leads to state

$$\left\{\frac{h}{h+g-\frac{1}{n}}, \frac{h+g-\frac{1}{n}}{p_n+q_n}\right\}.$$

If $h+g>\frac{1}{n}$, the drift is thus

$$\left\{+\frac{1}{n}\frac{h}{(h+g)(h+g-\frac{1}{n})}, -\frac{1}{n(p_n+q_n)}\right\}.$$

Otherwise, if $h+g=\frac{1}{n}$, the drift is

$$\left\{0, -\frac{1}{n(p_n+q_n)}\right\}.$$

4. A positive peer is replaced by a null peer. Positive peers decrease by one, while negative peers remain the same. This occurs with probability

$$h\left[1-(p_n+q_n)\right]$$

and leads to state

$$\left\{\frac{h-\frac{1}{n}}{h+g-\frac{1}{n}}, \frac{h+g-\frac{1}{n}}{p_n+q_n}\right\}.$$

If $h+g>\frac{1}{n}$, the drift is thus

$$\left\{-\frac{1}{n}\frac{g}{(h+g)(h+g-\frac{1}{n})}, -\frac{1}{n(p_n+q_n)}\right\}.$$

Otherwise, if $h+g=\frac{1}{n}$, the drift is

$$\left\{-1, -\frac{1}{n(p_n+q_n)}\right\}.$$

5. A null peer is replaced by positive peer. Positive peers increase by one, while negative peers remain unchanged. This occurs with probability

$$(1-(h+g))\left[q_n+p_n s_n(h+g)\frac{h}{h+g}\right]$$

and leads to state

$$\left\{\frac{h+\frac{1}{n}}{h+g+\frac{1}{n}}, \frac{h+g+\frac{1}{n}}{p_n+q_n}\right\}.$$

If the $h+g>0$, the drift is thus

$$\left\{+\frac{1}{n}\frac{g}{(h+g)\left(h+g+\frac{1}{n}\right)}, +\frac{1}{n(p_n+q_n)}\right\}.$$

Otherwise, if $h+g=0$, the drift is

$$\left\{+1, -\frac{1}{n(p_n+q_n)}\right\}.$$

6. A null peer is replaced by a negative peer. Negative peers increase by one, while positive peers remain unchanged. This occurs with probability

$$(1-(h+g))\left[p_n\big(1-s_n(h+g)\big)+p_n s_n(h+g)\frac{g}{h+g}\right]$$

and leads to state

$$\left\{\frac{h}{h+g+\frac{1}{n}}, \frac{h+g+\frac{1}{n}}{p_n+q_n}\right\}.$$

If $h+g>0$, the drift is thus

$$\left\{-\frac{1}{n}\frac{h}{(h+g)\left(h+g+\frac{1}{n}\right)}, +\frac{1}{n(p_n+q_n)}\right\}.$$

Otherwise, if $h+g=0$, the drift is

$$\left\{0, -\frac{1}{n(p_n+q_n)}\right\}.$$

From the above we get that, if $h+g>\frac{1}{n}$ (or, $\hat{z}>\frac{1}{n(p_n+q_n)}$), the mean drift $F_n^+(\hat{h},\hat{z})$ of $\hat{M}_n^+$ is

$$\begin{aligned}
F_n^+(\hat{h},\hat{z}) &= \mathbb{E}\big[\hat{M}_n^+(t+1)-\hat{M}_n^+(t)\mid \hat{M}_n^+(t)=\frac{h}{h+g}=\hat{h}, \hat{M}_n^{\text{z}}(t)=\frac{h+g}{p_n+q_n}=\hat{z}\big] \\
&= \frac{1}{n}\frac{1}{h+g+\frac{1}{n}}(1-(h+g))\left[\frac{g}{h+g}q_n-\frac{h}{h+g}p_n\big(1-s_n(h+g)\big)\right] \\
&\quad+\frac{1}{n}\frac{1}{h+g}(h+g)\left[\frac{g}{h+g}q_n-\frac{h}{h+g}p_n\big(1-s_n(h+g-\frac{1}{n})\big)\right] \\
&= \frac{1}{n}\Bigg[\frac{1}{(p_n+q_n)\hat{z}+\frac{1}{n}}(1-(p_n+q_n)\hat{z})\left[q_n-\Big(q_n+p_n\left[1-s_n\big((p_n+q_n)\hat{z}\big)\right]\Big)\hat{h}\right] \\
&\quad+\frac{1}{(p_n+q_n)\hat{z}}(p_n+q_n)\hat{z}\left[q_n-\left(q_n+p_n\left(1-s_n\left((p_n+q_n)\hat{z}-\frac{1}{n}\right)\right)\right)\hat{h}\right]\Bigg]
\end{aligned} \tag{6.26}$$

For $h+g=\frac{1}{n}$ (or, $\hat{z}=\frac{1}{n(p_n+q_n)}$), the mean drift $F_n^+(\hat{h},\hat{z})$ of $\hat{M}_+^n$ is

$$F_n^+(\hat{h},\hat{z})=\frac{gq_n}{n(h+g)}-\frac{hp_n}{n(h+g)}-h[1-(p_n+q_n)]=(h+g)q_n-h=\frac{q_n}{n}-\frac{\hat{h}}{n} \tag{6.27}$$

If $h+g=0$ (or, $\hat{z}=0$), the mean drift of $\hat{M}_n^+$ is

$$F_n^+(\hat{h},\hat{z})=q_n \tag{6.28}$$

On the other hand, the mean drift of $\hat{M}_n^{\mathbb{z}}$ can be easily shown to be

$$F_{\mathbb{z}}^n(\hat{h},\hat{z})=\mathbb{E}\left[\hat{M}_n^{\mathbb{z}}(t+1)-\hat{M}_n^{\mathbb{z}}(t)\mid \hat{M}_n^+(t)=\hat{h},\hat{M}_n^{\mathbb{z}}(t)=\hat{z}\right]=\frac{1}{n}[1-\hat{z}]. \tag{6.29}$$

Note that neither the mean drift of $\hat{M}_n^{\mathbb{z}}$ nor the transitions of $\hat{M}_n^{\mathbb{z}}$ actually depend on $\hat{h}$. We can consider the evolution of non-null peers as determined by the marginal $\hat{M}_n^{\mathbb{z}}$. In turn, $\hat{M}_n^+$ is driven by the evolution of $\hat{M}_n^{\mathbb{z}}$: a transition in $\hat{M}_n^{\mathbb{z}}$ determines the transitions in $\hat{M}_n^+$, and $\hat{M}_n^{\mathbb{z}}$ is an external "forcing" variable in the evolution of $\hat{M}_n^+$.

We will consider the following cases: $q_n=\Omega(p_n)$, and $q_n=o(p_n)$.

Step 2: The case $q_n=\Omega(p_n)$. We then have that $\liminf\frac{q_n}{p_n+q_n}\geq c>0$. W.l.o.g. we will assume that $q_n=\omega(p_n)$, *i.e.*, $c=1$. If $q_n=\Theta(p_n)$, then the analysis below still holds by taking $\epsilon_n=\frac{c}{n}$.

Consider two continuous time processes x_n^+ and $x_n^{\mathbb{z}}$, defined in terms of the discrete time processes $\hat{M}_+$ and $\hat{M}_{\mathbb{z}}$ as follows: For $\epsilon_n=\frac{1}{n}$, let $\tau_k=k\cdot\epsilon_n$ for $k\in\mathbb{N}$. Then, for all $k\in\mathbb{N}$,

$$\begin{aligned}&x_n^+(\tau_k)=\hat{M}_n^+(k),\text{ and}\\&x_n^+(\tau_k+s)=\hat{M}_n^+(k)+s\frac{\hat{M}_n^+(k+1)-\hat{M}_n^+(k)}{\tau_{k+1}-\tau_k},\text{ for } 0<s<\epsilon_n.\end{aligned}$$

and

$$\begin{aligned}&x_n^{\mathbb{z}}(\tau_k)=\hat{M}_n^{\mathbb{z}}(k),\text{ and}\\&x_n^{\mathbb{z}}(\tau_k+s)=\hat{M}_n^{\mathbb{z}}(k)+s\frac{\hat{M}_n^{\mathbb{z}}(k+1)-\hat{M}_n^{\mathbb{z}}(k)}{\tau_{k+1}-\tau_k},\text{ for } 0<s<\epsilon_n.\end{aligned}$$

We first show that, if the process $x_n^{\mathbb{z}}(t)$ starts close enough to 1, it will stay close to 1 with a high probability.

Lemma 6.17. *Assume $x_n^{\mathbb{z}}(0)\in[1-\delta,1+\delta]$ for some $0<\delta<0.5$. Then, for any $T>0$,*

$$\lim_{n\to\infty}\mathbf{P}(\sup_{0\leq t\leq T}|x_n^{\mathbb{z}}(t)-1|>2\delta)=0.$$

Proof. We first give a mean field representation of $\hat{M}_n^{\mathbb{z}}(t)$. Let $K > 0$ be a large enough constant, such that $K > 1 + 3\delta$. Consider the chain $\bar{M}_n^{\mathbb{z}}(t)$, which is $\hat{M}_n^{\mathbb{z}}(t)$ *observed* only in the set $[0, K]$. Formally (see also Aldous and Fill [AF], Chapter 2, Section 7.1), let

$$S_0 = \min\{t \geq 0 \ : \ \hat{M}_n^{\mathbb{z}}(t) \leq K\}$$
$$S_j = \min\{j > S_{j-1} \ : \ \hat{M}_n^{\mathbb{z}}(j) \leq K\}$$

and define

$$\bar{M}_n^{\mathbb{z}}(t) = \hat{M}_n^{\mathbb{z}}(S_t), \qquad t = 0, 1, \ldots$$

An important thing to observe is that, in $[0, K)$, the chain $\hat{M}_n^{\mathbb{z}}(t)$ is indistinguishable from $\bar{M}_n^{\mathbb{z}}(t)$. In particular, all transitions of the chain $\bar{M}_n^{\mathbb{z}}(t)$ in $[0, K)$ are the same as the possible transitions of $\hat{M}_n^{\mathbb{z}}(t)$ in $[0, K)$. The only difference is in state K: transitions leading outside the set in $\hat{M}_n^{\mathbb{z}}$ are leading back to K in $\bar{M}_n^{\mathbb{z}}$. In particular, the mean drift of $\bar{M}_n^{\mathbb{z}}$ is given by (6.29), except for $\bar{z} = K$, whereby the drift is $-\frac{\bar{z}}{n}$.

For $\epsilon_n = \frac{1}{n}$, we again define a continuous time process $y_n : \mathbb{R}_+ \to [0, 1]$ in terms of the discrete time process $\bar{M}_n^{\mathbb{z}} : \mathbb{N} \to [0, 1]$ in the same way that $x_n^{\mathbb{z}}$ was defined in terms of $\hat{M}_n^{\mathbb{z}}$. In particular, let $\tau_k = k \cdot \epsilon_n$ for $k \in \mathbb{N}$. Then, for all $k \in \mathbb{N}$,

$$y_n(\tau_k) = \bar{M}_n^{\mathbb{z}}(k), \text{ and}$$
$$y_n(\tau_k + s) = \bar{M}_n^{\mathbb{z}}(k) + s\frac{\bar{M}_n^{\mathbb{z}}(k+1) - \bar{M}_n^{\mathbb{z}}(k)}{\tau_{k+1} - \tau_k}, \text{ for } 0 < s < \epsilon_n.$$

The fact that the chains $\hat{M}_n^{\mathbb{z}}(t)$ and $\bar{M}_n^{\mathbb{z}}(t)$ are identical in $[0, K)$ implies the following. If $x^n(0) \in [1 - \delta, 1 + \delta]$, then

$$\mathbf{P}(\sup_{0 \leq t \leq T} |x_n^{\mathbb{z}}(t) - 1| < 2\delta) = \mathbf{P}(\sup_{0 \leq t \leq T} |y_n(t) - 1| < 2\delta)$$

if $y_n(0) = x^n(0)$. To see this, observe that the above probability is equal to the probability that the time either chain exits $[1 - 2\delta, 1 + 2\delta]$ is less than T. These exit times are the same in these two chains. As a result, to prove the lemma, it suffices to show that, conditioned on $y_n(0) \in [1 - \delta, 1 + \delta]$,

$$\lim_{n \to \infty} \mathbf{P}(\sup_{0 \leq t \leq T} |y_n(t) - 1| < 2\delta) = 0.$$

I.e., if y_n starts close enough to 1, it will stay close to 1 with a high probability.

We define a flow $\Phi : \mathbb{R} \times [0, 1] \to [0, 1]$, mapping $(\tau, m) \mapsto \Phi_\tau(m)$, as the solution of the following ODE

$$\Phi_0(m) = m, \qquad \frac{d\Phi_\tau(m)}{d\tau} = F(\Phi_\tau(m)) \tag{6.30}$$

where

$$F(\hat{z}) = \begin{cases} 1 - \hat{z}, & \hat{z} \in [0, K) \\ -\hat{z}, & \hat{z} = K \end{cases}. \tag{6.31}$$

From (6.29) we have that, for $\epsilon_n = 1/n$ and for $\bar{F}_n^{\mathbb{Z}}$ the mean drift of $\bar{M}_n^{\mathbb{Z}}$,

$$\frac{\bar{F}_n^{\mathbb{Z}}(\bar{z})}{\epsilon_n} = F(\hat{z})$$

where F as in (6.31). On the other hand,

$$\mathbb{E}\left[\left|\epsilon_n^{-1}\left[\hat{M}_n^{\mathbb{Z}}(t+1) - \hat{M}_n^{\mathbb{Z}}(t)\right]\right|^2 \mid \hat{M}_n^{+}(t) = \hat{h}, \hat{M}_n^{\mathbb{Z}}(t) = \hat{z}\right] \leq \frac{1+\hat{z}}{p_n + q_n} \tag{6.32}$$

We would like to show that $y_n(t)$ can be arbitrarily closely approximated by the flow defined by (6.30), as in Benaïm and Le Boudec [BL08]. In particular, we would like to show that

(*) For all $T > 0$ there exists a constant $C_1(T)$ and a random variable $B^n(T)$ in [0,1] such that

$$\sup_{0 \leq \tau \leq T} |y_n(\tau) - \Phi_\tau(m)| \leq C_1(T)(B^n(T) + |x^n(0) - m|)$$

where

$$\lim_{n \to \infty} \mathbb{E}[|B^n(t)|^2] = 0.$$

The lemma would then follow: The probability that the supremum of $|y_n(\tau) - \Phi_\tau(m)|$ for $\tau \in [0, T]$ exceeds δ goes to zero (by Chebychev's inequality). Moreover, $\Phi_\tau(x)$ for $x \in [1-\delta, 1+\delta]$ stays in this interval for all $\tau \in [0, \infty]$.

However, $y_n(t)$ fails to satisfy the assumptions of Benaïm and Le Boudec [BL08] in two ways. First, the second moment at each transition is not bounded: as p_n, q_n may decrease with n, (6.32) implies that the second moment at each transition can become unbounded. On the other hand, the functions $\epsilon_n^{-1} F_n^{\mathbb{Z}}$ are not Lipschitz continuous: they are discontinuous at $z = K$.

The fact that the second moment is not bounded can be dealt with by observing that

$$\frac{\epsilon_n}{p_n + q_n} = \frac{1}{n(p_n + q_n)} = o(1)$$

as $q_n = \omega(1/n)$. In Appendix A, we prove that even if the second moments at each transition are not bounded, a mean field limit exists in the sense of (*): the error term B^n converges to zero, as long as the second moment of each rescaled drift is $o(\epsilon_n^{-1})$. This is indeed the case, by (6.32) for $q_n = \omega(1/n)$.

The discontinuity can be dealt with as follows: The process $\bar{M}^z$ is a simple random walk on $[0, K]$ (in particular, it is reversible). Increasing the upwards drift from a state $x \in (1 + 2\delta, K]$ can only increase the probability that the chain spends more time in the interval $[1 + \delta, K]$. As a result, if we increase the positive drift of (6.31) so that $F(K) = 1 - K$, by increasing the time the chain spends in state K, we can only increase the probability the chain is above $1+2\delta$. Such a chain has a Lipschitz continuous drift, and thus can be used obtain a mean field model. The resulting flow has the same derivative as Φ in the interval $[1 - \delta, 1 + \delta]$, so we can show that the random process starting in $[1 - \delta, 1 + \delta]$ will stay within the interval $[1 - 2\delta, 1 + 2\delta]$. We can use this result thus to bound the probability that the original process $y_n(t)$ goes above $1 + 2\delta$.

Similarly, decreasing the upwards drift from a state $x \in [1 - 2\delta, K]$ will only increase the probability that the chain spends more time in the interval $[0, 1 - 2\delta]$. As a result, to obtain an upper bound on the probability that $y_n(t)$ goes below $1 - 2\delta$, we can increase the negative drift in a set of states of in an interval $[K - \epsilon, K]$. In particular, we can increase the probabilities of negative transitions so that, in $[K - \epsilon, \epsilon]$ is $F(\hat{z}) = (1 - \hat{z})f(\hat{z})$ where $f(z)$ a positive, infinitely differentiable function in $[K - \epsilon, K]$, such that $f(K - \epsilon) = 1$, $f(K) = \frac{K}{K-1}$, and the derivative of F exists and is bounded in all of $[0, 1]$. Creating a continuous drift thus will increase the probability that the system is below $1 - \delta$. The new chain has a mean field limit, and the resulting flow is again the same in the interval $[1 - \delta, 1 + \delta]$. Hence, we can use the mean field limit to bound the probability that the original chain $y_n(t)$ goes below $1 - 2\delta$, and the lemma follows. [*Lemma 6.17*] □

Let $I_\delta = [1 - \delta, 1 + \delta]$. We define two flows

$$\Phi : \mathbb{R} \times I_{2\delta} \to \times I_{2\delta} \quad \text{and} \quad \Psi : \mathbb{R} \times [0, 1] \to [0, 1]$$

that map

$$(\tau, \hat{z}) \mapsto \Phi^n_\tau(\hat{z}) \quad \text{and} \quad (\tau, \hat{h}) \mapsto \Psi_\tau(\hat{h})$$

and satisfy the following system of ODEs

$$\frac{d\Phi_\tau(\hat{z})}{d\tau} = 1 - \Phi_\tau(\hat{z}), \qquad \Phi_0(\hat{z}) = \hat{z}, \tag{6.33a}$$

$$\frac{d\Psi_\tau(\hat{h})}{d\tau} = \frac{1}{\Phi_\tau(\hat{z})}\left(1 - \Psi_\tau(\hat{h})\right), \qquad \Psi_0(\hat{h}) = \hat{h}. \tag{6.33b}$$

Lemma 6.18. *Assume that $x^z_n(0) \in [1 - \delta, 1 + \delta]$, for some $0 < \delta < 0.5$. Then, for any $T > 0$, and for any $\varepsilon > 0$,*

$$\lim_{n\to\infty} \mathbf{P}\left(\sup_{0\le t\le T} |x^+_n(t) - \Psi_t(\hat{h})| > \varepsilon\right) = 0.$$

where $\hat{h} = x_n^+(0)$ *and* $\Psi_t(\hat{h})$ *the solution of the system of ODEs* (6.33).

Proof. Let

$$A_n = \{ \sup_{0 \le t \le T} |x_n^+(t) - \Psi_t(\hat{h})| > \varepsilon \}$$

be the event that the trajectory of x_n^+ strays far away from Ψ_t. Moreover, let

$$E_n = \{ \sup_{0 \le t \le T} |x_n^{\scriptscriptstyle\mathbb{Z}}(t) - 1| < 2\delta \}$$

be the event that $x_n^{\scriptscriptstyle\mathbb{Z}}(t)$ is inside the set $[1 - 2\delta, 1 + 2\delta]$. We have that

$$\mathbf{P}(A_n) = \mathbf{P}(A_n \mid E_n)\mathbf{P}(E_n) + \mathbf{P}(A_n \mid E_n^c)\mathbf{P}(E_n^c) \le \mathbf{P}(A_n \mid E_n)\mathbf{P}(E_n) + \mathbf{P}(E_n^c)$$

On the other hand, by Lemma 6.17,

$$\lim_{n \to \infty} \mathbf{P}(E_n) = 1.$$

We can thus focus on the probability that A_n takes place *conditioned* on $x_n^{\scriptscriptstyle\mathbb{Z}}$ being inside the set $[1 - 2\delta, 1 + 2\delta]$. Then, the drift of M_n^+ is given by (6.26). Recall, from eq. (6.23) in the proof of Lemma 6.15, that $s_n(z)$ is bounded from below as follows:

$$s_n(z) \ge \phi_n (1 - e^{-\frac{\mathrm{TTL}_n z}{\bar{\tau}}})$$

where $\bar{\tau}$ a constant and

$$\phi_n = 1 - o\left(\frac{1}{n}\right)$$

a function s.t. $\lim_{n \to \infty} \phi_n = 1$. For this reason,

$$\lim_{n \to \infty} s_n((p_n + q_n)\hat{z}) \ge \lim_{n \to \infty} \phi_n (1 - e^{-\frac{\mathrm{TTL}_n (p_n + q_n)\hat{z}}{\tau_1}}) = 1$$

uniformly in $\hat{z}$, for $\hat{z} \in I_{2\delta}$. As a result, for $\epsilon_n = 1/n$,

- $\lim_{n \to \infty} \dfrac{F_n^+(\hat{h}, \hat{z})}{\epsilon_n} = \dfrac{1}{\hat{z}}(1 - \hat{h})$, and $\lim_{n \to \infty} \frac{F_n^{\scriptscriptstyle\mathbb{Z}}(\hat{h}, \hat{z})}{\epsilon_n} = (1 - \hat{z})$, uniformly in $[0, 1] \times I_{2\delta}$.
- The second moments of the drifts are

  $$\mathbb{E}\left[\left(\epsilon_n^{-1} \left| \hat{M}_n^+(t+1) - \hat{M}_n^+(t) \right| \right)^2 \mid \hat{M}_n^+(t) = \hat{h}, \hat{M}_n^{\scriptscriptstyle\mathbb{Z}}(t) = \hat{z} \right] \le \frac{2}{(1 - \delta)(p_n + q_n)}$$

  and

  $$\mathbb{E}\left[\left(\epsilon_n^{-1} \left| \hat{M}_n^{\scriptscriptstyle\mathbb{Z}}(t+1) - \hat{M}_n^{\scriptscriptstyle\mathbb{Z}}(t) \right| \right)^2 \mid \hat{M}_n^+(t) = \hat{h}, \hat{M}_n^{\scriptscriptstyle\mathbb{Z}}(t) = \hat{z} \right] \le 2\frac{1}{p_n + q_n}$$

- Both $\frac{F_n^+(\hat{h},\hat{z})}{\epsilon_n}$ and $\frac{F_n^{\mathbb{Z}}(\hat{h},\hat{z})}{\epsilon_n}$ are Lipschitz uniformly in n.

The joint process $\{x_n^+, x_n^{\mathbb{Z}}\}$, therefore satisfies conditions H1a and H3a of Benaïm and Le Boudec [BL08] but not H2a: the second moments of the scaled drifts are not bounded by a constant. As in the proof of Lemma 6.17, we can use the analysis presented in Appendix A to address this. In particular, Theorem A.1 implies that even if the moments are unbounded, a mean field limit exists, as long as the moments are $o(\epsilon_n)$. Indeed, of $q_n = \omega(1/n)$, both moments are $o(\epsilon_n)$ as $\epsilon_n = 1/n$ and

$$\lim_{n\to\infty} \frac{1}{n(p_n + q_n)}.$$

Therefore, Theorem A.1 implies that

$$\lim_{n\to\infty} \mathbf{P}\left(\sup_{0\le t\le T} |x_n^+(t) - \Psi_t(\hat{h})| > \varepsilon\right) = 0.$$

and the lemma follows. [*Lemma 6.18*] □

The above convergence result yields Lemma 6.16, as stated in the following lemma. We note that this lemma is a weaker version of Theorem 3 and Corollary 2 of Benaïm and Le Boudec [BL08].

Lemma 6.19. *Let ν_n^+ be the steady state distribution of $\hat{M}_n^+$. Then, ν_n^+ converges weakly to δ_1, the Dirac measure on* 1. *In particular,*

$$\lim_{n\to\infty} \gamma_n^+ = \lim_{n\to\infty} \mathbb{E}[\hat{M}_n^+] = 1 \tag{6.34}$$

Proof. Our proof follows the proof of Corollary 3.2 of Benaïm. Let $g : [0,1] \to \mathbb{R}$ be a continuous function. Fix $T > 0$ and let ν_n^+ be an invariant measure of of the chain $\hat{M}_n^+$. By invariance

$$\int_{[0,1]} g(x)\nu_n^+(dx) = \int_{[0,1]\times\mathbb{R}_+} \mathbb{E}_{\hat{h},\hat{z}}[g(x_n^+(T))]\nu_n(d\hat{h} \times d\hat{g}) \tag{6.35}$$

where $\mathbb{E}_{\hat{h},\hat{z}}[\cdot]$ the expectation conditioned on the starting point being $\{\hat{h},\hat{z}\}$ and ν_n the joint steady state probability (measure) of $\{\hat{M}_n^+, \hat{M}_n^{\mathbb{Z}}\}$. On the other hand, for $\nu_n^{\mathbb{Z}}$ the (marginal) invariant measure of $\hat{M}_n^{\mathbb{Z}}$,

$$\begin{aligned}
&\int_{[0,1]\times\mathbb{R}_+} \mathbb{E}_{\hat{h},\hat{z}}[g(x_n^+(T))]\nu_n(d\hat{h} \times d\hat{z}) = \int_{\mathbb{R}_+} \left(\int_{[0,1]} \mathbb{E}_{\hat{h},\hat{z}}[g(x_n^+(T))]\nu_n^+(d\hat{h})\right) \nu_n^{\mathbb{Z}}(d\hat{z}) \\
&= \int_{\mathbb{R}\setminus I_\delta} \left(\int_{[0,1]} \mathbb{E}_{\hat{h},\hat{z}}[g(x_n^+(T))]\nu_n^+(d\hat{h})\right) + \int_{I_\delta} \left(\int_{[0,1]} \mathbb{E}_{\hat{h},\hat{z}}[g(x_n^+(T))]\nu_n^+(d\hat{h})\right) \nu_n^{\mathbb{Z}}(d\hat{z}) \\
&\le ||g||\nu_n^{\mathbb{Z}}(\mathbb{R}_+ \setminus I_\delta) + \int_{I_\delta} \left(\int_{[0,1]} \mathbb{E}_{\hat{h},\hat{z}}[g(x_n^+(T))]\nu_n^+(d\hat{h})\right) \nu_n^{\mathbb{Z}}(d\hat{z})
\end{aligned}$$

where $I_\delta = [1-\delta, 1+\delta]$ and $||g||$ the L_∞ norm of g in $[0,1]$. However, by the Chernoff bound in (6.22)

$$\lim_{n\to\infty} \nu_n^+(\mathbb{R}_+ \setminus I_\delta) = 0.$$

As a result, for any $\varepsilon > 0$ we have that, for n large enough

$$\left| \int_{[0,1]\times\mathbb{R}} \mathbb{E}_{\hat{h},\hat{z}}[g(x_n^+(T))]\nu_n(d\hat{h}\times d\hat{g}) - \int_{I_\delta} \left(\int_{[0,1]} \mathbb{E}_{\hat{h},\hat{z}}[g(x_n^+(T))]\nu_n^+(d\hat{h}) \right) \nu_n^{\mathbb{z}}(d\hat{z}) \right| \leq \varepsilon. \tag{6.36}$$

On the other hand, as g is continuous over the compact set $[0,1]$, it is uniformly continuous over $[0,1]$. Let $\Psi_\tau(\hat{h})$ be the flow defined by the system ODEs (6.33). Then, by the uniform continuity of g on $[0,1]$ there exists an $\alpha > 0$ such that $|u - \Psi_T(\hat{h})| \leq \alpha$ implies that $|g(u) - g(\Psi_T(\hat{h}))| \leq \varepsilon$. Thus, for n large enough, we have that

$$\left| \mathbb{E}_{\hat{h},\hat{z}} \left[g(x_n^+(T)) - g(\Psi_T(\hat{h})) \right] \right| \leq \varepsilon + 2||g|| \mathbf{P}_{\hat{h},\hat{z}} \left(|x_n^+(T) - \Psi_T(x)| \geq \alpha \right) \tag{6.37}$$

Let ν^+ be a limit point of ν_n^+ in the weak topology, *i.e.* a measure such that

$$\lim_{n\to\infty} \int f(\hat{h})\nu_n^+(d\hat{h}) = \int f(\hat{h})\nu^+(d\hat{h})$$

for all continuous $f : [0,1] \to \mathbb{R}$. Lemma 6.18 and inequalities (6.35), (6.36) and (6.37) imply that

$$\left| \int_{[0,1]} g(\Psi_T(\hat{h}))\nu^+(dx) - \int_{[0,1]} g(\hat{h})\nu^+(d\hat{h}) \right| \leq 2\varepsilon.$$

Since T, g and ε are arbitrary, this implies that ν^+ is an invariant measure of the flow Ψ_T.

However, the unique invariant measure of Ψ_T is the Dirac measure on 1. This implies the first part of the lemma. The second part follows immediately by applying weak convergence to the identity function $g(\hat{h}) = \hat{h}$. [*Lemma 6.19*] □

We have thus proved Lemma 6.16 for the case where p_n decays faster than q_n. We know turn our attention to the second case.

Step 3: The case $q_n = o(p_n)$. In this case,

$$\lim_{n\to\infty} \frac{q_n}{q_n + p_n} = 0.$$

The argument that we used in Case 1 unfortunately breaks down. In particular, if we use a scaling factor of $\epsilon_n = \frac{1}{n}$, although process $\hat{M}_n^{\mathbb{z}}$ converges to a mean field limit, process $\hat{M}_n^+$

does not: its drift converges uniformly to zero. Essentially, the two processes evolve at different time scales: within the time scale that $\hat{M}_n^{\mathbb{Z}}$ evolves, $\hat{M}_n^+$ remains relatively constant.

A way to address this issue is to use two different scaling factors. We can again define two continuous time processes x_n^+ and $x_n^{\mathbb{Z}}$, in terms of the discrete time processes $\hat{M}_n^+$ and $\hat{M}_n^{\mathbb{Z}}$ as follows. Let

$$\epsilon_n^+ = \frac{q_n}{n(p_n + q_n)} \quad \text{and} \quad \epsilon_n^{\mathbb{Z}} = \frac{1}{n}.$$

Let $\tau_k = k \cdot \epsilon_n^+$ and $\zeta_k = k \cdot \epsilon_n^{\mathbb{Z}}$, for $k \in \mathbb{N}$. Then, for all $k \in \mathbb{N}$,

$$x_n^+(\tau_k) = \hat{M}_n^+(k), \text{ and} \tag{6.38a}$$

$$x_n^+(\tau_k + s) = \hat{M}_n^+(k) + s\frac{\hat{M}_n^+(k+1) - \hat{M}_n^+(k)}{\tau_{k+1} - \tau_k}, \text{ for } 0 < s < \epsilon_n^+. \tag{6.38b}$$

and

$$x_n^{\mathbb{Z}}(\zeta_k) = \hat{M}_n^{\mathbb{Z}}(k), \text{ and} \tag{6.39a}$$

$$x_n^{\mathbb{Z}}(\zeta_k + s) = \hat{M}_n^{\mathbb{Z}}(k) + s\frac{\hat{M}_n^{\mathbb{Z}}(k+1) - \hat{M}_n^{\mathbb{Z}}(k)}{\tau_{k+1} - \tau_k}, \text{ for } 0 < s < \epsilon_n^{\mathbb{Z}}. \tag{6.39b}$$

Observe that, again, x_n^+ is a rescaled version of $\hat{M}_n^+$, as $\hat{M}_n(t) = x_n^+(t\epsilon_n^+)$, for t an integer. On intervals $[t\epsilon_+, (t+1)\epsilon_+]$, x_n^+ is linearly interpolated between the values $\hat{M}_n^+(t+1)$ and $\hat{M}_n^+(t)$. A similar observation can be made about $x_n^{\mathbb{Z}}$. The important difference however is that $x_n^{\mathbb{Z}}$ has a different scaling factor than x_n^+. In particular, when x_n^+ is at a stage t of its evolution, $x_n^{\mathbb{Z}}$ is at a stage $\frac{t}{\epsilon_n'}$, where

$$\epsilon_n' = \frac{\epsilon_n^+}{\epsilon_n^{\mathbb{Z}}} = \frac{q_n}{p_n + q_n} \to 0$$

as $n \to \infty$. This prohibits us from re-using the approach of Case 1. In particular, we cannot assume that, within an interval $[0, T]$, in which the process x_n^+ can be approximated by a deterministic flow, $x_n^{\mathbb{Z}}$ will remain in a neighbourhood of 1. Within $[0, T]$, $x_n^{\mathbb{Z}}$ evolves in the interval $[0, \frac{T}{\epsilon_n'}]$. Therefore, it will escape the interval $[1 - \delta, 1 + \delta]$ with high probability. Thus, to describe the behaviour of x_n^+, we need to take the full dynamics of $\hat{M}_n^{\mathbb{Z}}$ into account.

The key idea behind doing so is in observing that $x_n^{\mathbb{Z}}$ is a fast "forcing" variable, whose evolution (captured by the drifts in eq. (6.29)) does not depend on x_n^+. Therefore, the classical averaging principle (see Kifer [Kif04] or Sanders and Verhurst [SV85]) applies. In particular, as $x_n^{\mathbb{Z}}$ evolves much faster than x_n^+, x_n^+ sees $x_n^{\mathbb{Z}}$ as if it were in steady state. More formally, let $\underline{M}_n$ be a process with drifts

$$\underline{F}_n(\hat{h}) = \int_{\mathbb{R}} F_n^+(\hat{h}, \hat{z}) \nu_n^{\mathbb{Z}}(d\hat{z}) \tag{6.40}$$

where $\nu_n^{\mathbb{Z}}(\hat{z})$ the invariant measure of $\hat{M}_n^{\mathbb{Z}}$. Define $\underline{x}_n$ in terms of $\underline{M}_n$ in the same way that x_n^+ was defined in terms of $\hat{M}_n^+$ in (6.38). Then, the averaging principle states that, for a fixed $T > 0$ and any $\alpha > 0$

$$\lim_{n\to\infty} \mathbf{P}\left(\sup_{0\le t\le T} |x_n^+(t) - \underline{x}_n(t)| > \alpha\right) = 0 \tag{6.41}$$

Hence, to describe the dynamics of $x_n^+(t)$, it suffices to describe the dynamics of $\underline{x}_n(t)$.

Lemma 6.20. *Let ν_n^+ be the steady state distribution of $\hat{M}_n^+$. Then, ν_n^+ converges weakly to δ_1, the Dirac measure on 1.*

Proof. Recall that the drift of $F_n^+(\hat{h},\hat{z})$ is given (for different values of $\hat{z}$) by eq. (6.26), (6.27) and (6.28). As a result, we have that, for $\hat{z} > \frac{1}{n(p_n+q_n)}$, and $\epsilon_n^+ = \frac{q_n}{n(p_n+q_n)}$,

$$\begin{aligned}\frac{F_n^+(\hat{h},\hat{z})}{\epsilon_n^+} = \frac{1}{\hat{z}+\frac{1}{n(p_n+q_n)}}(1-(p_n+q_n)\hat{z})\left[1-\left(1+\frac{p_n\left[1-s_n\big((p_n+q_n)\hat{z}\big)\right]}{q_n}\right)\hat{h}\right] \\ + \frac{1}{\hat{z}}(p_n+q_n)\hat{z}\left[1-\left(1+\frac{p_n\left(1-s_n\left((p_n+q_n)\hat{z}-\frac{1}{n}\right)\right)}{q_n}\right)\hat{h}\right]\end{aligned}$$

On the other hand, for $\hat{z} = \frac{1}{n(p_n+q_n)}$

$$\frac{F_n^+(\hat{h},\hat{z})}{\epsilon_n^+} \le \frac{(p_n+q_n)}{q_n}$$

and, for $\hat{z} = 0$

$$\frac{F_n^+(\hat{h},\hat{z})}{\epsilon_n^+} \le n(p_n+q_n)$$

Pick a δ such that $0 < \delta < 1$, and let $I_\delta = [1-\delta, 1+\delta]$. Recall from the Chernoff bound (6.22) that

$$\mathbf{P}\left(|x_n^{\mathbb{Z}} - 1| \ge \delta\right) = e^{-\Theta(\delta^2)n(p_n+q_n)} \tag{6.42}$$

It is easy to check, using the above formulae for the mean drifts and the above Chernoff bound, that if $q_n = \omega(1/n)$ then

$$\lim_{n\to\infty} \max_{\hat{z}\in\mathbb{R}\setminus I_\delta} \frac{F_n^+(\hat{h},\hat{z})}{\epsilon_n^+}\mathbf{P}\left(|x_n^{\mathbb{Z}} - 1| \ge \delta\right) = 0$$

uniformly in $\hat{h}$. In other words, the mean drift contributed to $\underline{F}$ (through eq. (6.40)) by $\hat{z}$ outside I_δ is negligible. We can thus focus on the drift contributed by $\hat{z} \in I_\delta$.

Recall from the proof of Lemma 6.15 that

$$s_n(z) \geq \phi_n(1 - e^{-\frac{\mathrm{TTL}_n z}{\bar{\tau}}})$$

where $\bar{\tau}$ a constant and

$$\phi_n = 1 - o\left(\frac{1}{n}\right)$$

a function s.t. $\lim_{n\to\infty} \phi_n = 1$. As a result, for $\hat{z} \in I_\delta$, we get that

$$\lim_{n\to\infty} \frac{p_n(1 - s_n((p_n+q_n)\hat{z} \pm \frac{1}{n(p_n+q_n)}))}{q_n} \leq \lim_{n\to\infty} \frac{p_n}{q_n}\left[1 - \phi_n(1 - e^{-2\mathrm{TTL}_n(p_n+q_n)/\bar{\tau}})\right] = 0$$

for $\phi_n = 1 - o\left(\frac{1}{n}\right)$, $q_n = \omega\left(\frac{1}{n}\right)$ and $\mathrm{TTL}_n = \Theta(n)$. As a result, for $\hat{z} \in I_\delta$,

$$\lim_{n\to\infty} \frac{F_n^+(\hat{h}, \hat{z})}{\epsilon_n^+} = \frac{1}{\hat{z}}(1 - \hat{h})$$

uniformly in $(\hat{h}, \hat{z}) \in [0,1] \times I_\delta$. Hence, the drift contributed to $\underline{F}_n$ by $\hat{z}$ in I_δ can be lower-bounded by

$$\frac{1}{1+\delta} - \frac{\hat{h}}{1-\delta}.$$

Putting the above results together, we get that $\underline{x}_n$ stochastically dominates a process in which

$$\lim_{n\to\infty} \underline{F}_n(\hat{h}) = \frac{1}{1+\delta} - \frac{\hat{h}}{1-\delta}$$

Finally, the second moments can be bounded as follows. For $\hat{z} > \frac{1}{n(p_n+q_n)}$, the second moment of $(\epsilon_n^+)^{-1}\left[\hat{M}_n^+(t+1) - \hat{M}_n^+(t)\right]$ is then given by

$$\begin{aligned}
&\mathbb{E}\left[\left|(\epsilon_n^+)^{-1}\left[\hat{M}_n^+(t+1) - \hat{M}_n^+(t)\right]\right|^2 \mid \hat{M}_n^+(t) = \hat{h}, \hat{M}_n^{\mathrm{z}}(t) = \hat{z}\right] \leq \\
&\frac{(p_n+q_n)^2}{q_n^2}\Bigg[2 \cdot \frac{1}{(p_n+q_n)^2(\hat{z} + \frac{1}{n(p_n+q_n)})} \cdot q + \\
&\quad 2 \cdot \frac{1}{(p_n+q_n)^2(\hat{z} + \frac{1}{n(p_n+q_n)})} p_n\left(1 - s_n\left((p_n+q_n)\hat{z} - \frac{1}{n}\right)\right) + \\
&\quad \frac{\hat{h}(1-\hat{h})}{(p_n+q_n)(z - \frac{1}{n(p_n+q_n)})}\Bigg] \\
&\quad \leq \frac{\frac{2}{q_n} + \frac{2p_n\left(1 - s_n\left((p_n+q_n)\hat{z} - \frac{1}{n}\right)\right)}{q_n^2} + \frac{p_n+q_n}{q_n^2}}{z - \frac{1}{n(p_n+q_n)}}
\end{aligned}$$

For $\hat{z} = \frac{1}{n(p_n+q_n)}$, the second moment of $(\epsilon_n^+)^{-1}\left[\hat{M}_n^+(t+1) - \hat{M}_n^+(t)\right]$ is then

$$\mathbb{E}\Big[\left|(\epsilon_n^+)^{-1}\left[\hat{M}_n^+(t+1) - \hat{M}_n^+(t)\right]\right|^2 \mid \hat{M}_n^+(t) = \hat{h}, \hat{M}_n^{\mathbb{Z}}(t) = \hat{z}\Big] \leq$$
$$\leq \frac{n^2(p_n+q_n)^2}{q_n^2}[gq_n + hp_n + h[1-(p_n+q_n)] \leq \frac{n^2(p_n+q_n)^2}{q_n^2}\left(\frac{q_n}{n} + \frac{\hat{h}}{n}\right) \leq 2\frac{n(p_n+q_n)^2}{q_n^2}$$

while, for $\hat{z} = 0$, the second moment of $(\epsilon_n^+)^{-1}\left[\hat{M}_n^+(t+1) - \hat{M}_n^+(t)\right]$ is

$$\mathbb{E}\Big[\left|(\epsilon_n^+)^{-1}\left[\hat{M}_n^+(t+1) - M_n^+(t)\right]\right|^2 \mid \hat{M}_n^+(t) = \hat{h}, \hat{M}_n^{\mathbb{Z}}(t) = \hat{z}\Big] =$$
$$= \frac{n^2(p_n+q_n)^2}{q_n^2}q = \frac{n^2(p_n+q_n)^2}{q_n}.$$

For $\epsilon_n^{\mathbb{Z}} = \frac{1}{n}$,

$$\mathbb{E}\left[\left|(\epsilon_n^{\mathbb{Z}})^{-1}\left[\hat{M}_n^{\mathbb{Z}}(t+1) - \hat{M}_n^{\mathbb{Z}}(t)\right]\right|^2 \mid \hat{M}_n^+(t) = \hat{h}, \hat{M}_n^{\mathbb{Z}}(t) = \hat{z}\right] \leq \frac{1+\hat{z}}{p_n+q_n}$$

In the case of the moments, it is easy to check using the above formulae and the Chernoff bound (6.42) that, for all $\hat{h}$,

$$\int_{\mathbb{R}_+} \mathbb{E}\left[\left|(\epsilon_n^{\mathbb{Z}})^{-1}\left[\hat{M}_n^{\mathbb{Z}}(t+1) - \hat{M}_n^{\mathbb{Z}}(t)\right]\right|^2 \mid \hat{M}_n^+(t) = \hat{h}, \hat{M}_n^{\mathbb{Z}}(t) = \hat{z}\right] \nu^{\mathbb{Z}}(d\hat{z}) \leq \beta_n$$

such that

$$\lim_{n\to\infty} \epsilon_n^+ \beta_n = 0.$$

These two observations, along with the Lipschitz continuity of the $\underline{F}_n$ drifts in $\hat{h}$ imply (from Theorem A.1) that $\underline{x}_n$ stochastically dominates a process that has a mean field model. The averaging principle implies through (6.41) that so does $\hat{x}_n^+$. Using the same argument as in Lemma 6.19, we can show that the steady state distribution of $\hat{x}_n^+$ stochastically dominates (in the limit) the Dirac measure on $\frac{1-\delta}{1+\delta}$. As δ can be arbitrarily small, this implies that the steady state distribution converges weakly to the Dirac measure on 1, and the lemma follows.

[*Lemma 6.20*] □

The above lemma is an analog of Lemma 6.19. In particular, as in Lemma 6.19, this implies that $\lim_{n\to\infty} \gamma_n^+ = 1$, by using the identity function in the weak convergence of the measures.

Step 4: Completing the proof. We have thus shown that γ_n^+ converges to one in the cases where $q_n = \Omega(p_n)$ and $q_n = o(p_n)$. This in fact will imply its convergence for all p_n: since the convergence occurs if $q_n = \Omega(p_n)$ or $q_n = o(p_n)$, it suffices to show that it also holds if

$$\text{neither } q_n = \Omega(p_n) \text{ nor } q_n = o(p_n). \tag{6.43}$$

For the sake of contradiction, suppose that (6.43) holds and that

$$\liminf \gamma_n^+ = \ell < 1.$$

Then, $\{\gamma_n^+\}_{n\in\mathbb{N}}$ has a convergent subsequence $\{\gamma_{n_k}^+\}_{k\in\mathbb{N}}$ such that

$$\lim_{k\to\infty} \gamma_{n_k}^+ = \ell.$$

Consider now the sequence

$$\left\{ \frac{q_{n_k}}{q_{n_k} + p_{n_k}} \right\}_{k\in\mathbb{N}}.$$

As neither $q_n = \Omega(p_n)$ nor $q_n = o(p_n)$ hold, the $\liminf$ and the $\limsup$ of this sequence may differ (and, in fact, may be 0 and 1, respectively). Nonetheless, by the Bolzano-Weierstrass theorem, $\left\{ \frac{q_{n_k}}{q_{n_k}+p_{n_k}} \right\}_{k\in\mathbb{N}}$ has a convergent subsequence, say for n_{k_j}, $j \in \mathbb{N}$. Since

$$\lim_{j\to\infty} \frac{q_{n_{k_j}}}{q_{n_{k_j}} + p_{n_{k_j}}} = c \in [0,1]$$

exists, either $q_{n_{k_j}} = \Omega\left(p_{n_{k_j}}\right)$ or $q_{n_{k_j}} = o\left(p_{n_{k_j}}\right)$. In both cases, the lemma's hypothesis implies that

$$\lim_{j\to\infty} \gamma_{n_{k_j}}^+ = 1.$$

This means that sequence $\gamma_{n_k}^+$ has a subsequence that converges to a limit other than ℓ, a contradiction. Hence, $\liminf \gamma_n^+ = 1$, and γ_n^+ converges to one for all possible p_n,q_n. [*Lemma 6.16*] □

6.4.4 Proof of Theorem 6.2

Theorem 6.2 follows from Theorem 6.5 and Theorem 6.6. In particular, the fact that the random walk using evidence of absence with TTL_n is scalable follows from the fact that an i.i.d. sequence of graphs is also an ergodic and vertex balanced Markov chain; hence, a system satisfying the assumptions of Theorem 6.2 also satisfies the assumptions of Theorem 6.5. Finally, the statement regarding reliability follows directly from Theorem 6.6.

6.5 Numerical Study

In this section, we present a numerical study to illustrate the validity of the theorems presented in the previous sections. To do that, we relax several of our modelling assumptions. In particular (a) we no longer assume that the system size is fixed, (b) we no longer assume that the

overlay graph evolution is described by an i.i.d. sequence, and (c) we allow the overlay network to change even during the propagation of queries.

We note that, in spite of the simplifying assumptions we made, our analysis correctly predicts the behaviour of the pure peer-to-peer system, and agrees with the simulation results, both qualitatively and quantitatively.

6.5.1 Simulation Setup

In our simulations, new peers arrive according to a Poisson process of rate λ. The lifetime of each peer is exponentially distributed with mean $1/\mu = 20$ minutes while the time to transmit a query is exponentially distributed with mean $\delta = 20$ milliseconds. We repeated our simulations for different arrival rates λ, between 10,000μ and 500,000μ. As a result, the average system size $n = \lambda/\mu$ is scaled in each experiment from ten thousand to half a million peers.

To connect peers during arrivals and departures, we implemented the connection protocol defined by Law and Siu [LS03], as described in Section 4.3.1. We again use $d = 16$ in our simulations.

We present results for two different systems: a system in which a traditional delay-constrained random walk with is used, and one in which the walk uses evidence of absence.

6.5.2 Random Walk

Scalability When a traditional delay-constrained random walk is used, queries for unavailable data items can generate an unbounded traffic load. Figure 6.4(a) shows the traffic load with and without evidence of absence for $p_n = 0.3$, $q_n = e^{-n}$ and $\mathrm{TTL}_n = 0.01n$. In both simulations, all queries generated failed. In the simple random walk the traffic load grows linearly, reaching 0.9609sec^{-1} (57.6 queries per minute) when the system size is 400,000 peers. In contrast, the traffic load when evidence of absence used is bounded: for all sizes, it varies between $1.217\text{E}{-3}\text{sec}^{-1}$ and $1.218\text{E}{-3}\text{sec}^{-1}$ (about 4.38 queries per hour). This happens precisely because negative peers stop queries quickly.

Reliability Decreasing the TTL_n (*e.g.*, to $\sqrt{n}$) improves the traffic load incurred by the simple random walk, but does not make it constant in n. This can be seen in Figure 6.5, where the traffic load under the traditional random walk with sub-linear TTL_n is plotted. Most importantly, using sub-linear TTL_n makes the system unreliable. We illustrate this in Figure 6.4(b), where query success rates for the simple random walk with $TTL(n) = 0.1\sqrt{n}$, $\mathrm{TTL}_n = \log(n)$

and $\mathrm{TTL}_n = 4$ are plotted. In all three experiments, $q_n = \omega\left(\frac{1}{n}\right)$. We see that the query success rate goes to zero even if an increasing number of peers brings the item into the system.

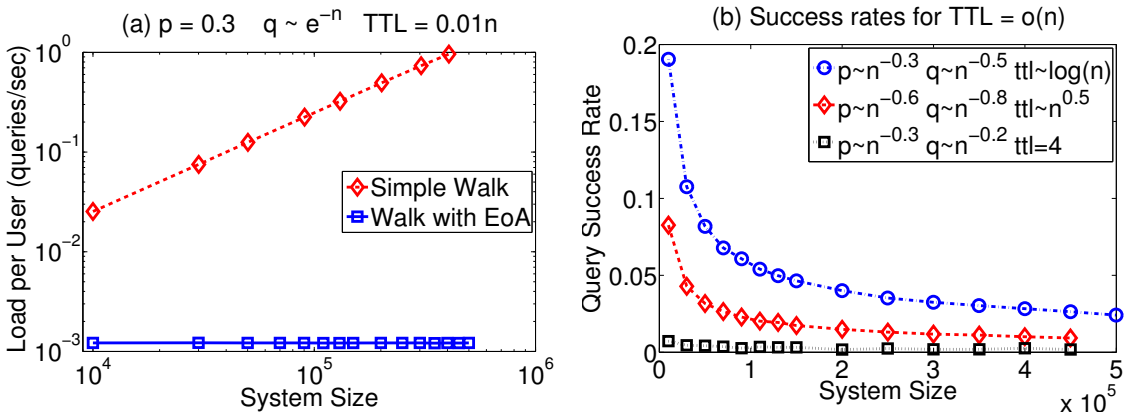


Figure 6.4: Traffic load and success rate of the delay-constrained random walk. In (a), the traffic load generated for a data item that is brought rarely into the system grows linearly in n, while using evidence of absence reduces it to a constant. In (b), the TTL_n grows slower than linear; as a result, even when the item is brought reliably into the system, most queries for it fail.

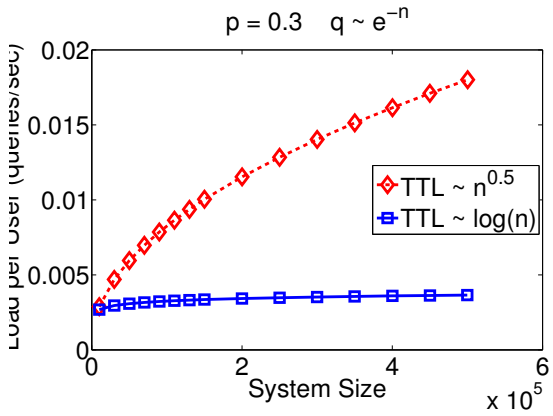


Figure 6.5: Traffic load of the delay-constrained random walk with sub-linear TTL_n. The traffic load generated for a data item that is brought rarely into the system grows sub-linearly in n, compared to the case when TTL_n is linear (*c.f.* Figure 6.4(a)).

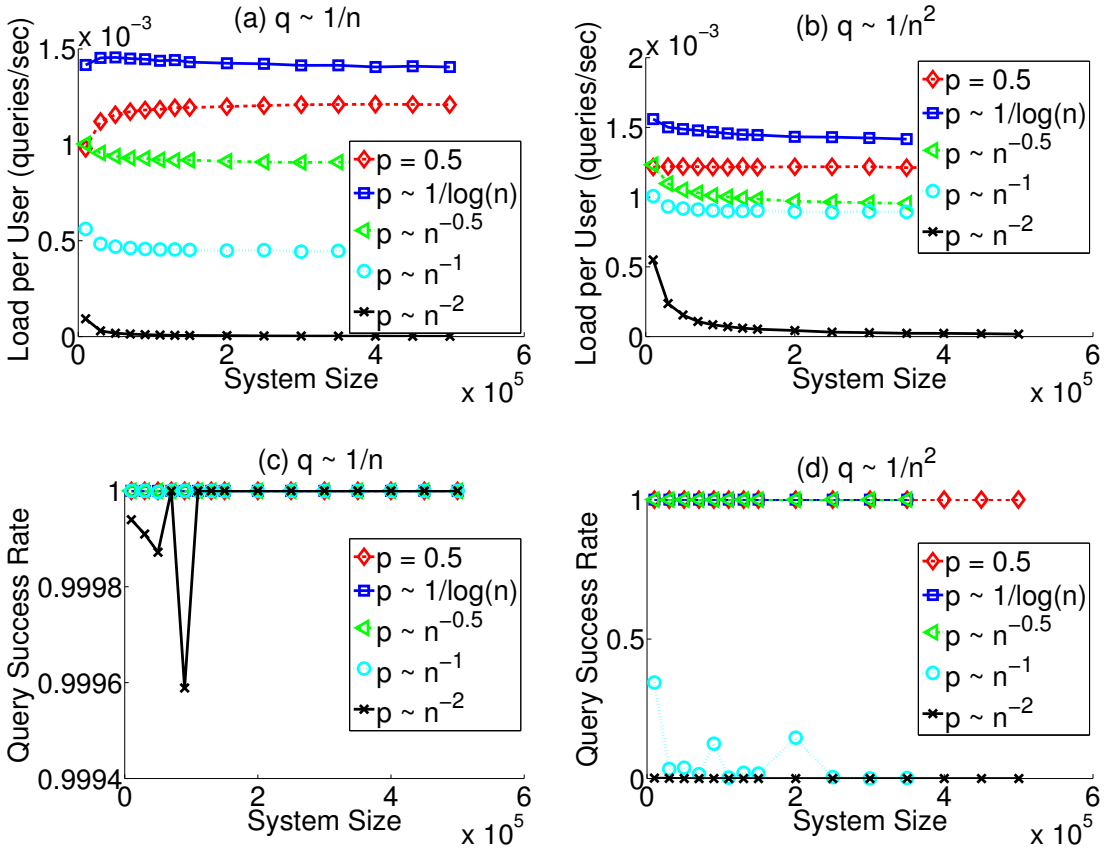


Figure 6.6: Traffic loads and success rates for data items brought into the system with publishing probabilities $q_n = 1000/n$ and $q_n = 1000^2/n^2$. Queries for the items brought into the system with probability $q_n = 1000/n$ almost always succeed, which is not true for $q_n = 1000^2/n^2$. In all cases however, the traffic load per peer remains bounded as the system size grows.

6.5.3 Random Walk Using Evidence of Absence

Scalability In Figures 6.6(a) and 6.6(b), we plot the average traffic load per peer for two different publishing probabilities, $q_n = 1000/n$ and $q_n = 1000^2/n^2$, respectively. In both case, we take the time-to-live to be $\mathrm{TTL}_n = 0.01n$, *i.e.*, linear in n. For $q_n = 1000/n$, the expected number of peers bringing the item into the system are 1000. This number does not grow with n and, therefore, the experiments in Figure 6.6(a) are "border line" in terms availability of the item. Moreover, note that any larger q_n, such as $q_n = 1/\sqrt{n}$, can only improve performance, by reducing the traffic load and increasing the success rate. In the case $q_n = 1000^2/n^2$, the item is not brought reliably into the system, as the expected number of peers that publish it decreases with n.

In each graph, we plot the traffic loads with the following request probabilities: $p_n =$

0.3, $p_n = \log(1000)/\log(n)$, $p_n = \sqrt{1000}/\sqrt{n}$, $p_n = 1000/(n)$ and $p_n = 1000^2/n^2$. In Figure 6.6(a), in all cases except the last, the average traffic load per peer is bounded; for $p_n = 1000^2/n^2$, we observe a decreasing traffic load. We observe an almost identical behaviour in Figure 6.6(b): the only difference is that, for $p_n = \Theta(1/n)$, the average traffic load per peer is now 1 query every 1000 seconds as opposed to 1 query every 2000 seconds.

Over all, the traffic load generated in all experiments does not grow with the system size, both when the system is reliably brought into the system and when it is not.

Reliability In Figures 6.6(c) and (d) we show the query success rate of the same experiments. When the item is brought reliably into the system (*i.e.*, for $q_n = 1000/n$), the query success rate was virtually 1 for all p_n and for all n: as seen in Figure 6.6(c), almost all queries succeeded (*i.e.*, located positive peers). Note that , in Figure 6.6(d), as $q_n = o(1/n)$, we are not guaranteed to find the item by Theorem 6.6. However, we observe that for high request probabilities we still can locate the item reliably; this is not true for $p_n = 1000/n$ and $p_n = 1000^2/n^2$, as almost all queries fail in this case.

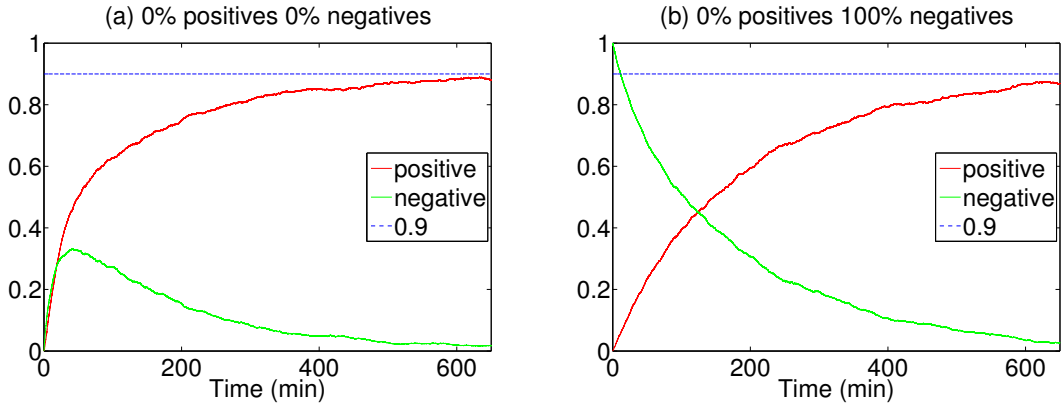


Figure 6.7: Traces of the fractions of positive and negative peers. In (a), all peers are initially null, while in (b) all peers are initially negative. In both cases, positive peers dominate and negative peers vanish.

System Dynamics In Fig. 6.7 we show how the fraction of positive and negative peers evolves as time progresses during a simulation. In this simulation, $p = 0.8$, $q = 0.1$, $\mathrm{TTL} = 100$ and $n = 10,000$. In Fig. 6.7(a), the simulation starts with all peers being null. Initially, negative peers grow faster than positive peers (as most queries fail); however, within 20 minutes the positive peers have surpassed the negative peers, which peak at 33 minutes and then start to

decay. In Fig. 6.7(b), the same experiment is repeated starting from a system where all peers are negative. As we see in Fig. 6.7, the effect of the initial state eventually dies out: although negative peers initially outnumber the positive peers, the number positive peers surpasses the number of negative peers within 2 hours.

In both simulations, eventually the positive peers prevail and their population converges to 90%, *i.e.*, to $p_n + q_n$, indicating that most queries succeed and every peer that requests the item becomes a positive peer.

6.6 Extensions and Open Questions

6.6.1 General Overlay Topologies

Both of the scalability results presented in this chapter (namely, Theorems 6.3 and 6.5) are proved under the general churn-driven Markovian graph model, while our reliability results (Theorems 6.4 and 6.6) required the (stronger) independent graph model. An interesting open problem is to extend the reliability results to the general model, especially since the simulations of Section 6.5 suggest that our observations are correct even in the churn-driven Markovian graph model. The challenge in obtaining such an extension lies in applying the mean field limit method to the above setting. In particular, the system in this case would no longer be a mean field interaction model, as defined by Benaïm and Le Boudec [BL08].

We note that, in general, the discussion appearing in Section 5.7.1 about non-expander topologies holds here as well. In particular, we can again consider an ergodic, vertex balanced, churn-driven Markov chain $\{G(t)\}_{t\in\mathbb{N}}$, whose stationary probability is such that

$$\mathbf{P}(\tau_n \leq f_n) = 1 - o\left(1/n\right),$$

for some positive sequence f_n with

$$\limsup f_n = +\infty.$$

Theorem 6.5 extends in a similar way to the analogue of 5.8, simply because the random walk using evidence of absence is equivalent to a walk in a hybrid system. In particular, the random walk using evidence of absence will not be scalable for all p_n, q_n, if the overlay graph is not an expander.

Theorem 6.6 can also be extended, combining the general framework of Section 5.7.1 with the analysis of the pure peer-to-peer system. Again, the random walk using evidence of absence

will not be reliable, in the sense of (6.3): the expected number of peers publishing the item would have to grow faster than

$$\min(f_n, n)$$

to guarantee that the query success rate converges to one.

6.6.2 General Query Propagation Mechanisms

The discussion appearing in Section 5.7.2 also applies here: investigating different query propagation mechanisms is an open problem. The path-replication scheme of Cohen and Shenker [LCC+02] are particularly interesting to investigate, since the original results about optimality of square root replication, discussed in Section 2.3.2, were stated in a pure peer-to-peer system context.

One challenge posed by path replication is that it can drastically increase the availability of a data item. For example, a long search (of the order of n) can increase the number of positive peers from constant to proportional to n. In this sense, the fraction of positive peers in a system with path replication is very volatile. Unfortunately, this does not have a mean field limit, and cannot be approximated by a deterministic process which is a solution of an ODE, as the latter are smooth functions.

An even more interesting question is how path replication would interact with evidence of absence. Our preliminary results show that replicating both data items and "evidence of absence", by turning all peers visited by a failed query to negative, can considerably harm the reliability of the system, and should thus be avoided. Instead, it is preferable to only replicate data items only.

6.6.3 Optimal Query Propagation Mechanisms

We can define a notion of an optimal mechanism in a pure peer-to-peer system, in a similar way as in Section 5.7.3. Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is an ergodic churn-driven Markov chain, whose distribution is not necessarily contiguous to the uniform distribution over $\mathbb{G}_{n,d}$. Given two query propagation mechanisms $\mathcal{A}$ and $\hat{\mathcal{A}}$, let ρ_n and $\hat{\rho}_n$ be the average traffic load per peer under each mechanism for a given request probability p_n and a given publishing probability q_n. Similarly, let γ_n, $\hat{\gamma}_n$ be the query success rates under each of the two mechanisms, respectively. We can then define a partial ordering between query propagation mechanisms as follows: we

write $\mathcal{A} \preceq_\rho \hat{\mathcal{A}}$ if, for any p_n, q_n,

$$\text{if } \hat{\rho}_n = O(1), \text{ then } \rho_n = O(1).$$

Similarly, we write $\mathcal{A} \preceq_\gamma \hat{\mathcal{A}}$ if, for any p_n and any $q_n = \omega\left(\frac{1}{n}\right)$,

$$\text{if } \lim_{n\to\infty} \hat{\gamma}_n = 1, \text{ then } \lim_{n\to\infty} \gamma_n = 1.$$

Finally, we write $\mathcal{A} \preceq \hat{\mathcal{A}}$ if $\mathcal{A} \preceq_\rho \hat{\mathcal{A}}$ and $\mathcal{A} \preceq_\gamma \hat{\mathcal{A}}$. Given an overlay graph model $\{G(t)\}_{t\in\mathbb{N}}$, we say that a query propagation mechanism $\mathcal{A}$ is *optimal* if, for every other query propagation mechanism $\mathcal{B}$,

$$\mathcal{A} \preceq \mathcal{B}.$$

Theorem 6.2 immediately implies the following corollary.

Corollary 6.3. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is an* i.i.d. *sequence of graphs sampled from a label-independent distribution. Moreover, assume there exists a constant $\bar{\tau}$ such that*

$$\mathbf{P}(\tau_n \leq \bar{\tau}) = 1 - o\left(\frac{1}{n}\right),$$

where τ_n the relaxation time of a graph sampled from the (stationary) distribution of $\{G(t)\}_{t\in\mathbb{N}}$. Then, the delay-constrained random walk with $\mathrm{TTL}_n = \Theta(n)$ that uses evidence of absence is an optimal query propagation mechanism.

As in the hybrid system, the results in Section 6.6.1 suggest that, if the overlay graph is not an expander *w.h.p.*, the random walk may not necessarily be optimal. Again, this is not surprising, as a query propagation mechanism over a structured system should exploit the structure of the overlay graph. Therefore, the optimality problem can be posed as follows in the pure setting.

Optimality in a Pure System. Given an ergodic, churn-driven Markov chain $\{G(t)\}_{t\in\mathbb{N}}$, find an optimal query propagation mechanism (if one exists).

Again, note that, in practise, the overlay graph is determined through a connection protocol followed by peers. Therefore, we first need to determine the stationary distribution of the overlay graph, resulting from the connection protocol, before attempting to answer the above question.

6.6.4 Stronger Notions of Reliability

Defining reliability through (6.3) is not unjustified. In particular, it seems reasonable to require that at least a constant number of peers publish the data item. Nonetheless, both the theoretical analysis and the simulation results (see, *e.g.*, Figure 6.6(c)) suggest that even if $q_n = o\left(\frac{1}{n}\right)$, the query success rate can converge to one, at least for high publishing probabilities p_n.

This motivates us to define a stronger notion of reliability. Recall from Section 6.3.2 that γ_n^* is the query success rate achieved under the maximal success rate mechanism. In such a mechanism, a query succeeds if at least one positive peer exists at the time it is initiated.

We say that a query propagation mechanism as *strongly reliable* if for all p_n, q_n

$$\lim_{n\to\infty} \gamma_n = 1 \text{ if } \lim_{n\to\infty} \gamma_n^* = 1. \tag{6.44}$$

I.e., if (and only if) the maximal success rate mechanism can locate an item reliably, so can the mechanism in question. Note that the "only if" direction is implied for any mechanism by Lemma 6.4. It would be interesting to see if any of the discussed mechanisms (the random walk with $\mathrm{TTL}_n = n$, the random walk using evidence of absence, *etc.*) is strongly reliable.

In general, we can use a relationship like (6.44) to define a partial ordering of arbitrary mechanisms with respect to reliability of a stronger form than $\preceq_\gamma$, described in the previous section. Given two query propagation mechanisms $\mathcal{A}$ and $\hat{\mathcal{A}}$ with query success rates γ_n and $\hat{\gamma}_n$, we can say that $\mathcal{A}$ is *more reliable* than $\mathcal{B}$ if, for all p_n *and all* q_n,

$$\text{if } \lim_{n\to\infty} \hat{\gamma}_n = 1 \text{ then } \lim_{n\to\infty} \gamma_n = 1 \tag{6.45}$$

It would be interesting to see, *e.g.*, if a random walk that does not use evidence of absence is more reliable than the same walk that uses evidence of absence, or if they are equally reliable (it terms of the partial ordering defined by (6.45)).

Unfortunately, our methods fall short of providing answers to such questions. The reason is that the existence of a mean field limit cannot be guaranteed in our model for $q_n = o\left(\frac{1}{n}\right)$ and, thus, we cannot describe the system behaviour in this regime. Intuitively, in a system where the expected number of peers publishing the item is decreasing, peers bring the item into the system very rarely. As a result, the system often empties out between two consecutive publications. This, in turn, implies that it cannot be well approximated by the solution of an ODE with a unique stationary point, as it may exhibit an oscillating behaviour.

6.6.5 Multiple Items

Our discussion of the traffic load incurred by multiple data items appearing Section 5.7.6 also applies for the random walk with evidence of absence in the pure setting. This is precisely because of the equivalence of this system to the hybrid system, as discussed in Section 6.4.1.

In particular, assume that there are M_n distinct data items in a pure peer-to-peer system of size n. Moreover, assume that a data item j, $j = 1, \ldots, M_n$, is requested by an incoming peer with a probability p_n^j and published with probability q_n^j. Let ϱ_n be the aggregate (over all items) traffic load per peer, under the delay-constrained random walk with evidence of absence.

An immediate implication of our results (Theorem 6.5) is that ϱ_n is bounded if the number of data items served by the system stays constant (bounded), *i.e.*, if $M_n = O(1)$. This is not a necessary condition, however: the following theorem is the analogue of Theorem 5.13 of Section 5.7.6.

Theorem 6.7. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and vertex balanced. Then, for any constant $\bar{\tau} > 1$,*

$$\varrho_n \leq \mu \cdot \sum_{j=1}^{M_n} p_n^j \left\{ \min\left[\frac{\bar{\tau}}{p_n^j + q_n^j}, \mathrm{TTL}_n \right] + \mathrm{TTL}_n (1 - p_n^j - q_n^j)^{n-1} + \mathrm{TTL}_n (1 - \phi_n) \right\}.$$

where $\phi_n = \mathbf{P}(\tau_n \leq \bar{\tau})$ and τ_n the steady state relaxation time of $\{G(t)\}_{t\in\mathbb{N}}$, defined by (4.1).

An immediate implication of the above theorem is that Corollary 5.7 of Section 5.7.6 applies as is to the pure peer-to-peer scenario: ϱ_n if all but finitely many items have rapidly decreasing request rates. There is however another condition which is also sufficient for scalability in the pure setting.

Corollary 6.4. *Assume that $\{G(t)\}_{t\in\mathbb{N}}$ is ergodic and vertex balanced, that there exists a constant $\bar{\tau} > 0$ such that*

$$\mathbf{P}(\tau_n \leq \bar{\tau}) = 1 - o\left(\frac{1}{n}\right),$$

and two constants $M > 0$, $B > 0$ such that

$$\sum_{j=M}^{M_n} \frac{p_n^j}{p_n^j + q_n^j} < B, \quad \textit{for large enough } n. \tag{6.46}$$

Then, if $\mathrm{TTL}_n = \Theta(n)$,

$$\varrho_n = O(1).$$

Combining Corollaries 5.7 and 6.4, we see that the system will have a bounded aggregate load if all but finitely many items are either (a) well published, in the sense that q_n is much larger than p_n, to the extent that $\frac{p_n^j}{p_n^j+q_n^j}$ are summable uniformly in n, or (b) are rarely requested, to the extent that np_n^j are summable uniformly in n. In other words, only finitely many items that are not well published *and* are requested often can be supported by the pure peer-to-peer system.

6.7 Summary

Our results show that a random walk using "evidence of absence" incurs a bounded load per peer, by limiting the query traffic generated by data items that are not in the system. This is an immediate implication of the equivalence of the above system to a hybrid system, in terms of the query traffic generated. Most importantly, the random walk using "evidence of absence" does so without jeopardizing the reliability of the system: every item that is brought in the system sufficiently often is guaranteed to be located *a.a.s.* Investigating the use of "evidence of absence" under different query propagation mechanisms, including under proactive replication, remains an interesting open question.

Chapter 7

Conclusions

Our work suggests that unstructured peer-to-peer systems have excellent scalability properties. In the hybrid setting, our work shows that an unstructured system can be deployed to reduce the traffic load at a server without overwhelming its clients. In the context of pure systems, our work has identified a query propagation mechanism that is both scalable, in the sense that the traffic incurred on each peer is bounded, and reliable, in the sense that queries for items published sufficiently often are guaranteed to succeed. These results are proved analytically under the model we defined in Chapter 4 and are also validated through simulations.

One of the strengths of our model is that it allows us to give a general definition of unstructured systems, without focusing on the connection protocol followed by peers: we can simply define unstructured systems in terms of the stationary distribution of their overlay graph. In particular, we say that a peer-to-peer system is unstructured if its overlay graph has a stationary distribution that is "almost" (in the sense of contiguity) uniform over all d-regular graphs. The results presented in this thesis can be seen as immediate consequences of the fact that such graphs are be expanders *a.a.s.* As discussed in Section 4.3, there are several examples of Markovian chains modelling the evolution of an overlay graph that have this property. Moreover, these examples are neither unique nor rare.

Although this is a direction not pursued in this thesis, our model is general enough to capture the behaviour of systems that are not unstructured: the connection protocol followed by peers can be such that the stationary distribution of the overlay graph is not contiguous to the uniform distribution over d-regular graphs. Such an overlay graph may not necessarily be an expander *a.a.s.* Our results of Sections 5.7.1 and 6.6.1 suggest that the random walk and the expanding ring over such overlays may not exhibit the nice scalability properties outlined above.

This is not surprising; if the overlay graph's stationary distribution is non-uniform over all d-regular graphs, it exhibits a certain structure. Rather than using a simple random walk (or, an expanding ring), a system designer should develop a query propagation mechanism that takes this structure into consideration, and exploits it when searching for data items. In particular, designing optimal query propagation mechanisms for different topologies is an interesting open problem, as discussed in Sections 5.7.3 and 6.6.3.

We have identified several possible extensions of this work, as well as open problems for both the hybrid and pure setting, in Sections 5.7 and 6.6, respectively. Of these, an open problem of particular importance from both a practical and a theoretical perspective is the analysis of non-passive replication schemes, such as path replication [CS02]. Overcoming the difficulties in analyzing such systems, which were described in Section 6.6.2, is interesting future work.

Bibliography

[AAK+08] Noga Alon, Chen Avin, Michal Koucky, Gady Kozma, Zvi Lotker, and Mark R. Tuttle. Many random walks are faster than one. In *SPAA*, pages 119–128, 2008.

[ABLS07] Noga Alon, Itai Benjamini, Eyal Lubetzky, and Sasha Sodin. Non-backtracking walks mix faster. *Communications in Contemporary Mathematics*, 9:585–603, 2007.

[AC08a] William Acosta and Surendar Chandra. Exploiting the properties of query workload and file name distributions to improve p2p synopsis-based searches. In *INFOCOM Mini-Symposium*, 2008.

[AC08b] William Acosta and Surendar Chandra. On the need for query-centric unstructured peer-to-peer overlays. In *HotP2P*, 2008.

[AC08c] William Acosta and Surendar Chandra. Understanding the practical limits of the Gnutella p2p system: An analysis of query terms and object name distributions. In *MMCN*, 2008.

[AF] David Aldous and Jim Fill. Reversible Markov Chains and Random Walks on Graphs. Monograph in preparation. `http://www.stat.berkeley.edu/~aldous/RWG/book.html`. Accessed on 29/12/2008.

[Alo86] Noga Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986.

[Ben98] Michel Benaïm. Recursive algorithms, urn processes and chaining number of chain recurrent sets. *Ergod. Th. & Dynam. Sys.*, 18, 1998.

[BL08] Michel Benaïm and Jean-Yves Le Boudec. A class of mean field interaction models for computer and communication systems. Technical Report LCA-REPORT-2008-010, EPFL, 2008.

[Bol01] Béla Bollobás. *Random Graphs*. Cambridge University Press, 2001.

[BR03] Charles Blake and Rodrigo Rodrigues. High availability, scalable storage, dynamic peer networks: Pick two. In *Ninth Workshop on Hot Topics in Operating Systems (HotOS-IX)*, pages 1–6, Lihue, Hawaii, May 2003.

[Buz] Buzm: Peer-to-peer wiki. `http://www.buzm.com`. Accessed on 23/12/2008.

[CCR03] Miguel Castro, Manuel Costa, and Antony Rowstron. Performance and dependability of structured peer-to-peer overlays. Technical Report MSR-TR2003 -94, Microsoft Research, 2003.

[CCR04] Miguel Castro, Manuel Costa, and Antony Rowstron. Peer-to-peer overlays: structured, unstructured or both? Technical Report MSR-TR-2004-73, Microsoft Research, 2004.

[CDG05] Colin Cooper, Martin Dyer, and Catherine Greenhill. Sampling regular graphs and a peer-to-peer network. In *SODA*, pages 980–988, 2005.

[CDH09] Colin Cooper, Martin Dyer, and Andrew J. Handley. The flip markov chain and a randomizing p2p protocol. In *PODC*, 2009.

[Chu97] Fan Rong K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[CRB+03] Yatin Chawathe, Sylvia Ratnasamy, Lee Breslau, Nick Lanham, and Scott Shenker. Making Gnutella-like p2p systems scalable. In *SIGCOMM*, 2003.

[CS02] Edith Cohen and Scott Shenker. Replication strategies in unstructured peer-to-peer networks. *SIGCOMM Comput. Commun. Rev.*, 32(4):177–190, 2002.

[DWD04] David G. Deschenes, Scott D. Weber, and Brian D. Davison. Crawling Gnutella: Lessons learned. Technical Report LU-CSE-04-005, Lehigh University, 2004.

[FGMS06] Tomás Feder, Adam Guetz, Milena Mihail, and Amin Saberi. A local switch markov chain on given degree graphs with application in connectivity of peer-to-peer networks. In *FOCS*, 2006.

[Fis03] A. Fisk. Gnutella dynamic query protocol v0.1, 2003. `http://www9.limewire.com/developer/dynamic_query.html`. Accessed on 29/12/2008.

[FR08] Charles P. Fry and Michael K. Reiter. Really truly trackerless bittorrent. Technical Report CMU-CS-06-148, Carnegie Mellon University, 2008.

[FRA+05] Ronaldo A. Ferreira, Murali Krishna Ramanathan, Asad Awan, Ananth Grama, and Suresh Jagannathan. Search with probabilistic guarantees in unstructured peer-to-peer networks. In *P2P*, 2005.

[Fri03] Joel Friedman. A proof of Alon's second eigenvalue conjecture. In *STOC '03*, pages 720–724, New York, NY, USA, 2003. ACM Press.

[Gal96] Robert G. Gallager. *Discrete Stochastic Processes*. Kluwer, Boston, 1996.

[GKLM07] Ayalvadi J. Ganesh, Anne-Marrie Kermarrec, Erwan Le Merrer, and Laurent Massoulié. Peer counting and sampling in overlay networks based on random walks. *Journal of Distributed Computing*, 20(4), 2007.

[GMS04] Christos Gkantsidis, Milena Mihail, and Amin Saberi. Random walks in peer-to-peer networks. In *INFOCOM*, 2004.

[GMS05] Christos Gkantsidis, Milena Mihail, and Amin Saberi. Hybrid search schemes for unstructured peer-to-peer networks. In *INFOCOM*, 2005.

[Gnu] Bootstrapping, Gnutella 0.6 RFC, `http://rfc-gnutella.sourceforge.net/developer/testing/bootstrapping.html`. accessed on 29/12/2008.

[HKL+06] Sidath B. Handurukande, Anne-Marie Kermarrec, Fabrice Le Fessant, Laurent Massoulié, and Simon Patarin. Peer sharing behaviour in the eDonkey network, and implications for the design of server-less file sharing systems. *SIGOPS Oper. Syst. Rev.*, 40(4):359–371, 2006.

[HKLM04] Sidath B. Handurukande, Anne-Marie Kermarrec, Fabrice Le Fessant, and Laurent Massoulié. Exploiting semantic clustering in the eDonkey p2p network. In *EW11: Proceedings of the 11th workshop on ACM SIGOPS European workshop*, 2004.

[HLL+07] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross. A measurement study of a large-scale p2p iptv system. *IEEE Transactions on Multimedia*, 2007.

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the AMS*, 43(4):439–561, October 2006.

[IM08] Stratis Ioannidis and Peter Marbach. On the design of hybrid peer-to-peer systems. In *SIGMETRICS*, 2008.

[IM09] Stratis Ioannidis and Peter Marbach. Absence of evidence as evidence of absence: A simple mechanism for scalable peer-to-peer search. In *INFOCOM*, 2009.

[ipo] Ipoque internet studies. `http://www.ipoque.com/resources/internet-studies/`. Accessed on 31/08/2009.

[Kah95] Nabil Kahale. Eigenvalues and expansion of regular graphs. *Journal of the ACM*, 42(5):1091–1106, 1995.

[Kif04] Yuri Kifer. Averaging principle for fully coupled dynamical systems and large deviations. *Ergod. Th. & Dynam. Sys.*, 24, 2004.

[KK03] M. Frans Kaashoek and David R. Karger. Koorde: A simple degree-optimal distributed hash table. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, 2003.

[LCC+02] Qin Lv, Pei Cao, Edith Cohen, Kai Li, and Scott Shenker. Search and replication in unstructured peer-to-peer networks. In *ICS*, 2002.

[LHH+04] Boon Thau Loo, Joseph M. Hellerstein, Ryan Huebsch, Scott Shenker, and Ion Stoica. Enhancing p2p file-sharing with an internet-scale query processor. In *VLDB*, 2004.

[LHSH04] Boon Thau Loo, Ryan Huebsch, Ion Stoica, and Joseph M. Hellerstein. The case for a hybrid p2p search infrastructure. In *IPTPS*, 2004.

[Lima] QRP. How Gnutella works. `http://wiki.limewire.org/index.php?title=QRP`. Accessed on 29/12/2008.

[Limb] Ultrapeers. How Gnutella works. `http://wiki.limewire.org/index.php?title=Ultrapeers`. Accessed on 29/12/2008.

[LKR05] Jian Liang, Rakesh Kumar, and Keith W. Ross. The Kazaa overlay: A measurement study. *Computer Networks (Special Issue on Overlays)*, 2005.

[LKRG03] Dmitri Loguinov, Anuj Kumar, Vivek Rai, and Sai Ganesh. Graph-theoretic analysis of structured peer-to-peer systems: routing distances and fault resilience. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 395–406. ACM Press, 2003.

[LRW03] Nathaniel Leibowitz, Matei Ripeanu, and Adam Wierzbicki. Deconstructing the Kazaa network. In *Proceedings of the The Third IEEE Workshop on Internet Applications*, 2003.

[LS03] Ching Law and Kai-Yeung Siu. Distributed construction of random expander networks. In *INFOCOM*, 2003.

[LSG+04] Jinyang Li, Jeremy Stribling, Thomer M. Gil, Robert Morris, and Frans M. Kaashoek. Comparing the performance of distributed hash tables under churn. In *Proc. IPTPS*, 2004.

[Łuc92] Tomasz Łuczak. Sparse random graphs with a given degree sequence. *Random Graphs*, 2:165–182, 1992.

[MBR03] Gurmeet Singh Manku, Mayank Bawa, and Prabhakar Raghavan. Symphony: Distributed hashing in a small world. In *Proc. 4th USENIX Symposium on Internet Technologies and Systems*, 2003.

[MBSM05] Ruggero Morselli, Bobby Bhattacharjee, Aravind Srinivasan, and Michael A. Marsh. Efficient lookup on unstructured topologies. In *PODC*, 2005.

[MNR02] Dahlia Malkhi, Moni Naor, and David Ratajczak. Viceroy: A scalable and dynamic emulation of the butterfly. In *Proceedings of the 21st ACM Symposium on Principles of Distributed Computing*, 2002.

[MS05] Peter Mahlmann and Christian Schindelhauer. Peer-to-peer networks based on random transformations of connected regular undirected graphs. In *SPAA*, 2005.

[NRZ+07] Giovanni Neglia, Giuseppe Reina, Honggang Zhang, Don Towsley, Arun Venkataramani, and John Danaher. Availability in bittorrent systems. In *INFOCOM*, 2007.

[NW03] Moni Naor and Udi Wieder. Novel architectures for p2p applications: the continuous-discrete approach. In *SPAA '03: Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures*, pages 50–59, New York, NY, USA, 2003. ACM Press.

[PDGM06] Marcell Perényi, Trang Dinh Dang, András Gefferth, and Sándor Moln'ar. Identification and analysis of peer-to-peer traffic. *Lecture Notes in Computer Science*, 3976, 2006.

[PSZ08] Krishna P.N. Puttaswamy, Alessandra Sala, and Ben Y. Zhao. Searching for rare objects using index replication. In *INFOCOM*, 2008.

[QS04] Dongyu Qiu and R. Srikant. Modeling and performance analysis of bittorrent-like peer-to-peer networks. In *SIGCOMM*, pages 367–378, New York, NY, USA, 2004. ACM.

[RIF02] Matei Ripeanu, Adriana Iamnitchi, and Ian Foster. Mapping the Gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. *IEEE Internet Computing*, 6(1), 2002.

[Rit01] Jordan Ritter. Why Gnutella can't scale. no, really. http://www.darkridge.com/ jpr5/doc/gnutella.html, 2001.

[RSR06] Amir H. Rasti, Daniel Stutzbach, and Reza Rejaie. On the long-term evolution of the two-tier Gnutella overlay. In *INFOCOM*, 2006.

[SENB07] Moritz Steiner, Taoufik En-Najjary, and Ernst W. Biersack. Exploiting KAD: Possible uses and misuses. *Computer Communication Review*, 37(5), 2007.

[SGG02a] Stefan Saroiu, Krishna P. Gummadi, and Steven D. Gribble. A measurement study of peer-to-peer file sharing systems. In *MMCN*, 2002.

[SGG02b] Stefan Saroiu, Krishna P. Gummadi, and Steven D. Gribble. Measuring and analyzing the characteristics of napster and Gnutella hosts. *Multimedia Systems Journal*, 8(5), 2002.

[SMK+01] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM*. ACM Press, 2001.

[sop] Sopcast. `http://www.sopcast.com/`. Accessed on 25/06/2009.

[SR04] Daniel Stutzbach and Reza Rejaie. Towards a better understanding of churn in peer-to-peer networks. Technical report, Univ. of Oregon, 2004.

[SRS08] Daniel Stutzbach, Reza Rejaie, and Subhabrata Sen. Characterizing unstructured overlay topologies in modern p2p file-sharing systems. *IEEE/ACM Transactions on Networking*, 16(2), 2008.

[SV85] Jan A. Sanders and Ferdinand Verhurst. *Averaging Methods in Nonlinear Dynamical Systems*. Springer-Verlag, 1985.

[SZR07] Daniel Stutzbach, Shanyu Zhao, and Reza Rejaie. Characterizing files in the modern Gnutella network. *Multimedia Systems*, 13(1), 2007.

[Tay81] R. Taylor. Constrained switchings in graphs. In *Combinatorial Mathematics VIII*, volume 884, 1981.

[TK06] Saurabh Tewari and Leonard Kleinrock. Proportional replication in peer-to-peer networks. In *INFOCOM*, 2006.

[TKLB07] Wesley W. Terpstra, Jussi Kangasharju, Christof Leng, and Alejandro P. Buchman. BubbleStorm: Resilient, probabilistic and exhaustive peer-to-peer search. In *SIGCOMM*, 2007.

[Tri] Tribler: The fastest way of social file sharing. `http://www.tribler.org/`. Accessed on 23/12/2008.

[Wes01] Douglas B. West. *Introduction to Graph Theory*. Prentice Hall, 2nd edition, 2001.

[Wor99] Nicholas C. Wormald. Models of random regular graphs. *Surveys in Combinatorics*, 276:239–298, 1999.

[YaC] YaCy. Distributed web search. `http://yacy.net/`. Accessed on 23/12/2008.

[YGM02] Beverly Yang and Hector Garcia-Molina. Improving search in peer-to-peer networks. In *ICDCS*, 2002.

[ZCSK07] Matei A. Zaharia, Amit Chandel, Stefan Saroiu, and Srinivasan Keshav. Finding content in file-sharing networks when you can't even spell. In *IPTPS*, 2007.

[ZK08] Matei Zaharia and Srinivasan Keshav. Gossip-based search selection in hybrid peer-to-peer networks. *J. Concurrency and Computation: Practice and Experience*, 20(2), Feb. 2008.

Index

Appendix A

An Extension of the Mean Field Limit Theorem of Benaïm and Le Boudec

In the following, we extend Theorem 1 of Benaïm and Le Boudec [BL08]. Our extension involves relaxing Hypothesis H2a (Assumption (3) below). In particular, we do not require that the rescaled drifts have second moments bounded by a constant; instead, these can simply be bounded by functions $\beta(\varepsilon)$, where ε the scaling factor, such that

$$\lim_{\varepsilon \to 0} \varepsilon \beta(\varepsilon) = 0.$$

Our proof follows closely the steps taken by Benaïm and Le Boudec [BL08] and by Benaïm in [Ben98].

Let $\mathcal{R}$ be a finite set. Let Δ be a compact convex subset of $\mathbb{R}^d$, $d \in \mathbb{N}$, with a non-empty interior. For each $\varepsilon > 0$, let $Z^\varepsilon = \{M^\varepsilon(t), R^\varepsilon(t)\}_{t \in \mathbb{N}}$ be a discrete time Markov chain whose state space is $\Delta \times \mathcal{R}$, such that:

$$M^\varepsilon(t+1) - M^\varepsilon(t) = \varepsilon G^\varepsilon(t+1) \tag{A.1}$$

and

$$\mathbf{P}\left(G^\varepsilon(t+1) \in dx, R^\varepsilon(t+1) = j' \mid M^\varepsilon(t) = m, R^\varepsilon(t) = j\right) = K^\varepsilon_{jj'}(m)\nu^\varepsilon_{m,j}(dx) \tag{A.2}$$

where, for all $m \in \Delta$, $\varepsilon > 0$, and $j \in \mathcal{R}$, we have that $K^\varepsilon(m)$ is a transition matrix on $\mathcal{R}$ and $\nu^\varepsilon_{m,j}$ is a probability measure in $\mathbb{R}^d$ whose support is $(\Delta - m)/\varepsilon$.

Let

$$f^\varepsilon_j(m) = \int_{\mathbb{R}^d} x \nu^\varepsilon_{m,j}(dx).$$

We make the following assumptions

Assumption 1 (Uniform convergence of f_j^ε to f_j)**.** *For all $j \in \mathcal{R}$, there exists an $f_j : \mathbb{R}^d \to \mathbb{R}^d$ such that f_j^ε converges to f_j uniformly in Δ.* I.e.*, for all $\delta > 0$, there exists $\varepsilon_0 > 0$ such that for all $0 < \varepsilon < \varepsilon_0$ and for all $m \in \Delta$, $||f_j^\varepsilon(m) - f_j(m)|| < \delta$.*

Assumption 2 (Uniform convergence of K^ε to K)**.** *For each m, there exists an indecomposable transition matrix $K(m)$ such that K^ε converges to K uniformly in Δ.* I.e.*, for all $\delta > 0$, there exists $\varepsilon_0 > 0$ such that for all $0 < \varepsilon < \varepsilon_0$ and for all $m \in \Delta$, $||K^\varepsilon(m) - K(m)|| < \delta$ where $|| \cdot ||$ a matrix norm (*e.g.*, the operator norm).*

Assumption 3 (G^ε has a bounded second moment)**.** *For all $m \in \mathbb{R}^d$ and for small enough $\varepsilon > 0$*

$$\int_{\mathbb{R}^d} ||x||^2 \nu_{m,j}^\varepsilon(dx) \leq \beta(\varepsilon)$$

where $\lim_{\varepsilon \to 0} \varepsilon\beta(\varepsilon) = 0$.

Assumption 4 (Lipschitz continuity)**.** *The maps K^ε and f_j^ε are Lipschitz continuous, uniformly in ε.* I.e.*,*

$$||K^\varepsilon(m) - K^\varepsilon(m')|| + ||f_j^\varepsilon(m) - f_j^\varepsilon(m')|| \leq L||m - m'||$$

where L does not depend on ε.

Since $K(m)$ is indecomposable, there exists a unique invariant probability measure $\pi(m)$ such that

$$\pi(m) = \pi(m)K(m).$$

Let

$$F(m) = \sum_{j \in \mathcal{R}} \pi_j(m) f_j(m).$$

Assumptions 1, 2 and 4 imply that $K(m)$, $f_j(m)$ are Lipschitz continuous on $\mathbb{R}^d$. Furthermore, since $\pi(m)$ depends smoothly on the coefficients of $K(m)$, the implicit function theorem implies that F is Lipschitz continuous on $\mathbb{R}^d$. It is also bounded, as a continuous function defined on a compact set Δ. Hence F is *completely integrable*, *i.e.*, there exists a unique flow

$$\Phi : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$$

that maps (τ, x) to $\Phi_\tau(x)$ and satisfies

$$\Phi_0(x) = x, \quad \frac{d\Phi_\tau(x)}{dt} = F(\Phi_\tau(x)). \tag{A.3}$$

We define a continuous-time process $\hat{M}^\varepsilon : \mathbb{R}_+ \to \Delta^\varepsilon$ in terms of $M : \mathbb{N} \to \Delta^\varepsilon$ as follows. Let $\tau_k = k\varepsilon$, $k \in \mathbb{N}$. Then, for all $k \in \mathbb{N}$,

$$\hat{M}^\varepsilon(\tau_k) = M^\varepsilon(k), \text{ and}$$
$$\hat{M}^\varepsilon(\tau_k + s) = M^\varepsilon(k) + s\frac{M^\varepsilon(k+1) - M^\varepsilon(k)}{\tau_{k+1} - \tau_k}, \text{ for } 0 < s < \varepsilon.$$

Observe that $\hat{M}^\varepsilon$ is a rescaled version of M^ε, as $M^\varepsilon(t) = \hat{M}^\varepsilon(t\varepsilon)$ for t an integer. On intervals $[t\varepsilon, (t+1)\varepsilon]$, $\hat{M}^\varepsilon$ is linearly interpolated between the values $M^\varepsilon(k+1)$ and $M^\varepsilon(k)$. Note that $\hat{M}^\varepsilon$ lives in Δ by the convexity of Δ.

Theorem A.1 (Benaïm and Le Boudec, [BL08]). *For any $T > 0$, there exist constants $C_1(T)$ and $C_2(T)$ and a random variable $B^\varepsilon(T)$ such that*

$$\sup_{0 \le \tau \le T} ||\hat{M}^\varepsilon(\tau) - \Phi_\tau(x)|| \le C_1(T)[B^\varepsilon(T) + ||\hat{M}^\varepsilon(0) - x||]$$

and

$$\mathbb{E}\left[||B^\varepsilon(T)||^2\right] \le C_2(T)\varepsilon\beta(\varepsilon)$$

Proof. Let

$$U^\varepsilon(t+1) = G^\varepsilon(t+1) - F(M^\varepsilon)$$

so that

$$M^\varepsilon(t+1) - M^\varepsilon(t) = \varepsilon[F(M^\varepsilon(t)) + U^\varepsilon(t+1)], \tag{A.4}$$

for $t \in \mathbb{N}$. We can bound the "error" term $B^\varepsilon(T)$ of the theorem in terms of U^ε.

Lemma A.1 (Benaïm [Ben98]). *For all $T > 0$ there exists $C_1(T) > 0$ such that*

$$\sup_{0 \le \tau \le T} ||\hat{M}^\varepsilon(\tau) - \Phi_\tau(x)|| \le C_1(T)\left[B^\varepsilon(T) + ||\hat{M}^\varepsilon(0) - x||\right]$$

where

$$B^\varepsilon(T) = \varepsilon \sup_{0 < t \le \lceil \frac{T}{\varepsilon} \rceil} ||\sum_{i=1}^{t} U^\varepsilon(i)||.$$

Proof. Recall that $\tau_k = k\varepsilon$, $k \in \mathbb{N}$. Define a piecewise constant version of $\hat{M}^\varepsilon$ as

$$\underline{M}^\varepsilon(\tau_k + s) = \hat{M}^\varepsilon(\tau_k)$$

for all $k \in \mathbb{N}$ and for $0 \leq s < \varepsilon$. Similarly, define

$$\underline{U}^{\varepsilon}(\tau_k + s) = U^{\varepsilon}(k+1)$$

for all $k \in \mathbb{N}$ and for $0 \leq s < \varepsilon$. Finally, define the "inverse" of $k \to \tau_k$ as $k : \mathbb{R}_+ \to \mathbb{N}$ where $k(\tau) = \sup\{i \geq 0 : \tau \geq \tau_i\}$

Under this notation, (A.4) implies that

$$\hat{M}^{\varepsilon}(\tau) = \hat{M}^{\varepsilon}(0) + \int_0^{\tau} [F(\underline{M}^{\varepsilon}(s)) + \underline{U}^{\varepsilon}(s)]\, ds$$

or

$$\hat{M}^{\varepsilon}(\tau) = \hat{M}^{\varepsilon}(0) + \int_0^{\tau} F(\hat{M}^{\varepsilon}(s))ds + A_1(\tau) + A_2(\tau)$$

where

$$A_1(\tau) = \int_0^{\tau} [F(\underline{M}^{\varepsilon}(s)) - F(\hat{M}^{\varepsilon}(s))]ds \quad \text{and}$$
$$A_2(\tau) = \int_0^{\tau} \underline{U}(s)ds$$

Flow Φ_τ satisfies

$$\Phi_\tau(x) = x + \int_0^{\tau} F(\Phi_s(x))ds.$$

Hence

$$||\hat{M}^{\varepsilon}(\tau) - \Phi_\tau(x)|| \leq \int_0^{\tau} ||F(\hat{M}^{\varepsilon}(s)) - F(\Phi_s(x))|| + ||A_1(\tau)|| + ||A_2(\tau)|| + ||\hat{M}^{\varepsilon}(0) - x||$$

Recall that F is Lipschitz and bounded in Δ. Therefore $||F(\hat{M}^{\varepsilon}(s)) - F(\Phi_s(x))|| \leq L||\hat{M}^{\varepsilon}(s) - \Phi_s(x)||$, for some constant L, and

$$||\hat{M}^{\varepsilon}(\tau) - \Phi_\tau(x)|| \leq L\int_0^{\tau} ||\hat{M}^{\varepsilon}(s) - \Phi_s(x)|| + ||A_1(\tau)|| + ||A_2(\tau)|| + ||\hat{M}^{\varepsilon}(0) - x||.$$

Grönwall's lemma therefore gives that

$$||\hat{M}^{\varepsilon}(\tau) - \Phi_\tau(x)|| \leq H(\tau) + \int_0^{\tau} LH(\tau)e^{L(\tau - s)}ds,$$

where $H(\tau) = ||A_1(\tau)|| + ||A_2(\tau)|| + ||\hat{M}^{\varepsilon}(0) - x||$. Therefore,

$$\sup_{0 \leq \tau \leq T} ||\hat{M}^{\varepsilon}(\tau) - \Phi_\tau(x)|| \leq C(T)\left(\sup_{0 \leq \tau \leq T} ||A_1(\tau)|| + \sup_{0 \leq \tau \leq T} ||A_2(\tau)|| + ||\hat{M}^{\varepsilon}(0) - x||\right) \tag{A.5}$$

where $C(T)$ a (deterministic) function of T, not depending on ε.

We have that

$$\sup_{0\leq\tau\leq T}||A_2(\tau)||\leq\sup_{0\leq\tau\leq T}||\varepsilon\sum_{i=1}^{k(\tau)+1}U^\varepsilon(i)||\leq B^\varepsilon(T) \tag{A.6}$$

where

$$B^\varepsilon(s)=\varepsilon\sup_{0<t\leq\lceil s/\varepsilon\rceil}||\sum_{i=1}^{t}U^\varepsilon(i)||.$$

On the other hand,

$$\begin{aligned}||\underline{M}^\varepsilon(s)-\hat{M}^\varepsilon(s)||&=||\int_{\tau_{k(s)}}^{s}F(\underline{M}^\varepsilon(u))+\underline{U}^\varepsilon(s)ds||\\&\leq\varepsilon K+||\int_{\tau_{k(s)}}^{s}\underline{U}^\varepsilon(s)ds||\end{aligned} \tag{A.7}$$

for some $K>0$ as F is bounded. Furthermore,

$$\begin{aligned}||\int_{\tau_{k(s)}}^{s}\underline{U}^\varepsilon(s)ds||&\leq||\int_{0}^{\tau_{k(s)}}\underline{U}^\varepsilon(s)ds||+||\int_{0}^{s}\underline{U}^\varepsilon(s)ds||\\&\leq\varepsilon||\sum_{i=1}^{k(s)}U^\varepsilon(i)||+\varepsilon||\sum_{i=1}^{k(s)+1}U^\varepsilon(i)||\leq 2B^\varepsilon(s).\end{aligned}$$

Therefore, inequality (A.7) gives

$$\sup_{0\leq s\leq\tau}||\underline{M}^\varepsilon(s)-\hat{M}^\varepsilon(s)||\leq\varepsilon K+2B^\varepsilon(\tau)$$

From the Lipschitz continuity of F we have

$$||[F(\underline{M}^\varepsilon(s))-F(\hat{M}^\varepsilon(s))||\leq L||\underline{M}^\varepsilon(s)-\hat{M}^\varepsilon(s)||$$

and, hence,

$$\sup_{0\leq\tau\leq T}||A_1(\tau)||\leq LT\left(\varepsilon K+2B^\varepsilon(\tau)\right) \tag{A.8}$$

for some constant L. The lemma thus follows from inequalities (A.6), (A.8) and (A.5) □

From Assumption 2, K^ε is indecomposable for small enough ε. We let π^ε denote the invariant probability of K^ε and

$$F^\varepsilon(m)=\sum_i\pi_i^\varepsilon(m)f_i^\varepsilon(m).$$

Let

$$U^{\varepsilon}(t) = \sum_{i=1}^{3} U^{i,\varepsilon}(t)$$

where

$$\begin{aligned} U^{1,\varepsilon} &= F^{\varepsilon}(M^{\varepsilon}(t)) - F(M^{\varepsilon}(t)), \\ U^{2,\varepsilon} &= G^{\varepsilon}(t+1) - f^{\varepsilon}_{R^{\varepsilon}(t+1)}(M^{\varepsilon}(t)), \text{ and} \\ U^{3,\varepsilon} &= f^{\varepsilon}_{R^{\varepsilon}(t+1)}(M^{\varepsilon}(t)) - F^{\varepsilon}(M^{\varepsilon}(t)). \end{aligned}$$

Then

$$B^{\varepsilon}(T) \leq \sum_{i=1}^{3} B^{i,\varepsilon}(T)$$

where

$$B^{i,\varepsilon(T)} = \varepsilon \sup_{0<t\leq\lceil\frac{T}{\varepsilon}\rceil} || \sum_{j=1}^{t} U^{i,\varepsilon}(j)||.$$

By Assumptions 1 and 2, and the fact that π^{ε} depend smoothly on K^{ε}, we have that for all $\delta > 0$, there exists $\varepsilon_0 > 0$ such that for all $0 < \varepsilon < \varepsilon_0$ and for all $m \in \Delta$, $||F^{\varepsilon}(m) - F(m)|| \leq \delta$. Hence, given $\delta > 0$, we have for $0 < \varepsilon < \varepsilon_0$

$$\begin{aligned} B^{1,\varepsilon} &= \varepsilon \sup_{0<t\leq\lceil\frac{T}{\varepsilon}\rceil} \left|\left| \sum_{j=1}^{t} [F^{\varepsilon}(M^{\varepsilon}(j)) - F(M^{\varepsilon}(j))] \right|\right| \\ &\leq \varepsilon \sup_{0<t\leq\lceil\frac{T}{\varepsilon}\rceil} \sum_{j=1}^{t} ||F^{\varepsilon}(M^{\varepsilon}(j)) - F(M^{\varepsilon}(j))|| \\ &\leq \epsilon\lceil T/\epsilon\rceil\delta \end{aligned}$$

Hence $\lim_{\epsilon\to 0} B^{1,\varepsilon} = 0$ with probability one. Note that

$$\mathbb{E}[G^{\varepsilon}(t+1) \mid M^{\varepsilon}(t) = m, R^{\varepsilon}(t) = j] = \mathbb{E}[f^{\varepsilon}_{R(t+1)}(m) \mid R^{\varepsilon}(t) = 1]$$

so $\sum_{i=1}^{t} U^{2,\varepsilon}$ is a martingale. On the other hand

$$\mathbb{E}[||G^{\varepsilon}(t+1)||^2 \mid M^{\varepsilon}(t) = m, R^{\varepsilon}(t) = j] = \sum_{i\in\mathcal{R}} K^{\varepsilon}_{ji}(m) \int_{\mathbb{R}^d} ||x||^2 \nu^{\varepsilon}_{m,i}(dx) \leq \beta(\varepsilon).$$

Hence $\mathbb{E}\left[\left|\left|\sum_{i=1}^{t} U^{2,\varepsilon}(i)\right|\right|^2\right] \leq 4t\beta(\varepsilon)$. Doob's L^p maximal inequality implies that there is a constant C such that

$$\mathbb{E}\left[\left(\sup_{0\leq k\leq t}\left|\left|\sum_{i=1}^{t} U^{2,\varepsilon}\right|\right|\right)^2\right] \leq C\mathbb{E}\left[\left|\left|\sum_{i=1}^{t} U^{2,\varepsilon}(i)\right|\right|^2\right] \leq 4Ct\beta(\varepsilon)$$

and, thus,

$$\mathbb{E}\left[||B^{2,\varepsilon}(T)||^2\right]\varepsilon^2 \leq 4C\left\lceil\frac{T}{\varepsilon}\right\rceil\beta(\varepsilon) = O\left(\varepsilon\beta(\varepsilon)\right).$$

Finally, bounding $U^{3,\varepsilon}$ can be done in precisely the same way as in Benaïm and Le Boudec [BL08]; we thus refer the reader to their proof. □