A dynamic, distributed and scalable admission control algorithm for QoS provision in multi-hop wireless ad hoc networks

by

Aibin Guo

A thesis submitted in partial fulfillment of
the requirements for the degree of

Master of Science

Electrical Engineering

at

Northeastern University
Boston, Massachusetts

2003

Northeastern University

Graduate School of Engineering

Thesis Title: A dynamic, distributed and scalable admission control algorithm for QoS provision in multi-hop wireless ad hoc networks

Author: Aibin Guo

Department: Electrical and Computer Engineering

Approved for Thesis Requirements of the Master of Science degree

Thesis committee: ______________________________

Thesis Advisor: A. Bruce McDonald

____

Thesis Reader:

____

Thesis Reader:

____

Chairman of Department: Fabrizio Lombardi

Graduate School Notified of Acceptance:

____

Director of the Graduate School

____

Date

To myself.

ACKNOWLEDGMENTS

I thank the many people who have done lots of nice things for me.

DISCARD THIS PAGE

TABLE OF CONTENTS

DISCARD THIS PAGE

LIST OF TABLES

DISCARD THIS PAGE

LIST OF FIGURES

Figure Page

ABSTRACT

A novel idea of admission control for QoS provision in multi-hop wireless ad hoc networks is presented in the thesis. It is dynamic, distributed, scalable, which is crucial for multi-hop wireless ad hoc networks.

Blah blah blah

Chapter 1

Analysis

1.1 The problem

For multi-hop ad hoc wireless data networks, i.e. packet-radio networks, the wireless channel is a shared resource. Everyone wants to transmit over the same channel needs to contend the media, but can not transmit at the same time, otherwise, collision will occur and garbled signal will be received. Although dual-band or multi-band devices may use the channel more efficiently, such as some commercial equipments support both IEEE 802.11a (operating at 5G Hz) and IEEE 802.11b (operating at 2.4G Hz) and they can auto sense the channel, there are still some issues besides the cost. First, normally such devices sense the channel during the startup and then only use one channel most of the time, they do not switch channel in a very frequent manner. When different nodes use the same channel, the potential collision is still there. Second, even the device has separate interfaces to separate channels, how to coordinate the channel allocation will become another hard problem, i.e. graph-coloring problem. So, in the paper, we only consider devices based on single channel access interface.

The media access control (MAC) protocol is essential in a multi-hop mobile ad hoc network that all nodes share a common broadcast channel. The most important differences between the wireless LAN and the MAC protocol of most wired networking applications is the impossibility to detect collisions. With the receiving and sending antennas immediately next to each other, a station is unable to see any signal but its own. As a result, the complete packet will be sent before the incorrect checksum reveals that a collision has happened. So, Carrier Sense Multiple Access with Collision Detection (CSMA/CD) is inappropriate to be used in wireless ad hoc networks. Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is used prevent collisions at the moment they are most likely to happen, i.e. when the transmit starts. All clients are forced to wait for a random number of timeslots and sense the medium, before starting a transmission. If the medium is sensed to be busy, a back-off procedure is initiated. Mobile ad hoc networks are different with wireless networks with base-station. The base-station is in the sight of every mobile node. However, for multi-hop ad hoc networks, individual mobile node may not in the transmission range of the other. Without appropriate MAC protocol, when two mobile nodes, which are out of transmission range of each other, want to transmit packets to the same third node at the same time, the collision will occur at the third node although the channel is free at the two source nodes. This is called the hidden terminal problem [ref]. IEEE 802.11 Distributed Coordination Function (DCF) [ref] is introduced to address the hidden terminal problem when there is no base-station. Basically, when a node wants to transmit, it sends out a very short Request-To-Send (RTS) packet. Short packet reduces the collision probability. The RTS packet includes source node address, destination node address, and the channel usage duration, which is called Network Allocation Vector (NAV). All nodes except the destination node hearing the RTS will defer their transmission for the period of NAV. Similarly, the destination node responds with a Clear-To-Send (CTS) short packet with the adjusted NAV if its channel is clear. All nodes hearing the CTS also back off. Then, the source sends out the real data packet (DATA), and the destination responds with a short Acknowledgement (ACK) packet. By doing RTS-CTS-DATA-ACK, the probability of collision is greatly reduced. Currently, IEEE 802.11 DCF is the most commonly used MAC protocol. One drawback is that it causes exposed terminal problem [ref], which make the protocol loses some efficiency. However, in order to achieve the maximum efficiency, complicated scheduling and coordinating algorithms are needed, which seems unreal for a dynamic network with dynamic traffic.

[Insert figures to show hidden and exposed terminal problems]

The admission control is hard not only because of the dynamic traffic, but also because of the difficulty to have a clear picture of the available bandwidth and network capacity. Essentially, every node has to be able to know how much available bandwidth it has before it can accept incoming QoS based traffic. Due to the infrastructureless feature, or limited infrastructure, for ad hoc networks, this has to be done distributed and independently. The intrinsic problem is that it is inappropriate for a node judging the available bandwidth only by the traffic it is receiving and transmitting. If other nodes in its transmission range are transmitting, then it is obvious that the bandwidth consumed by its neighbors will have effect on its own available bandwidth because they share the common channel. We call this kind traffic as induced traffic. When a node uses a certain amount of bandwidth, it virtually imposes similar amount traffic on all its neighbors' available bandwidth. That why we say the network throughput for a IEEE 802.11b wireless network is much lower than the maximum transmitting rate 11M bps, no need to mention the control packets overhead and the exposed terminal problem. It is easy to know the induced traffic has effect on the neighbors. But how the neighbors are affected, how much they are affected, there is any effect on two hop neighbors, or even three hop neighbors, all these questions have not been answered. In the paper, we give a detailed analysis of these and get the mathematic results on the available bandwidth estimation. Then, we use the result to help us address the dynamic QoS admission control problem. Given an ad hoc network, although the traffic is dynamic, a node still knows how much bandwidth it is uses for receiving and transmitting. It can easily know the instantaneous traffic by querying the MAC interface or keep a network layer statistics. Then, it can calculate the running average of the bandwidth it uses. For QoS traffic, such as real time video traffic, it is easy to predict how much bandwidth it will use in the future because QoS traffic follows the specified QoS parameters. For best effort traffic, a running average is good enough as the statistical result. Besides, best effort traffic is more network performance tolerant. However, we do not want to starve the best effort traffic. So, we keep QoS traffic with higher priority but still allocate a minimum bandwidth for best effort traffic. We'd like to mention that different MAC protocols have different effects or results. Given the specific MAC protocol, we can always know how the neighbors share the common channel. For us, we consider the IEEE 802.11 DCF scheme. Also, we assume every node knows all its current neighbors. It is easy to be done by exchanging beacon packets. Then, every node exchanges a certain amount of information with its neighbors, such as neighbor list, bandwidth usage. For simplicity, we ignore the traffic overhead of such information exchange when we do the analysis.

1.2 Mathematic Analysis

1.2.1 Definitions

neighbor When we say node j is a neighbor of node i, it means j is in the transmission range of node i. We consider all nodes have the same transmission range r, and the coverage area is just a circle with radius r.

induced traffic For node i, when other nodes in its transmission range are transmitting, it is obvious that node i can not transmit at the same time. The bandwidth consumed by its neighbors will have effect on node i's available bandwidth because they share the common channel. We call this kind traffic as induced traffic.

virtual link When node i and node j are neighbors of each other, we say node i and j can form a virtual link, which means they can transmit to each other. The link capacity of a virtual link (i,j) means the maximum bandwidth can be used by node i to transmit to node j. The available bandwidth of virtual link (i,j) means the amount of bandwidth available for node i to transmit to node j. Normally, node i's available bandwidth is smaller than the link capacity minus the bandwidth already used by node i because of the induced traffic caused by other nodes. Also, without specifying the MAC protocol, the induced traffic may have different effect on the two nodes, i and j, it is possible that the available bandwidth of link (i,j) is different with the available bandwidth of link (j,i). However, for IEEE 802.11 DCF, the available bandwidth on link (i,j) is the same as on link (j,i), which is showed in the following sections.

1.2.2 Notations

$N(i)$ All neighbors of node i (not including i itself).

$N_1(i)$ Same as $N(i)$.

$N_2(i)$ $\forall j \neq i : j \notin N(i); j \in N(\forall k : k \in N(i))$, i.e. all 2-hop neighbors, but not one-hop neighbors, of node i.

$N_3(i)$ $\forall j \neq i : j \notin N(i); j \notin N_2(i); j \in N(\forall k : k \in N_2(i))$, i.e. all 3-hop neighbors, but not one-hop neighbors, nor two-hop neighbors, of node i.

$N(i) \cup N(j)$ $\forall k | k \in N(i)\, or\, k \in N(j)$, i.e. all the nodes in the combined coverage map of node i and node j.

$vlink(i, j)$ the virtual link from node i to node j

$T(i, j)$ The amount of real traffic on virtual link $vlink(i, j)$, i.e. the amount of real traffic from node i to node j, (in the unit of Mbps)

$T_{incuded}(i, j)$ The amount of the induced traffic over virtual link $vlink(i, j)$.

$L(i, j)$ The virtual link capacity between node i and node j

$B_{available}(i, j)$ The available bandwidth from node i to node j

1.2.3 Some Notes

Because unicast traffic is the most dominant traffic in the network, we only consider unicast traffic in the paper. However, it is not hard to expand our results to the multicast and broadcast situation.

In order to simplify the mathematic expression, when we calculate the total induced traffic over virtual link $vlink(i, j)$, we treat the existing traffic between node i and node j, which are neighborhood, as a kind of induced traffic. By doing this, it also simplifies the procedure to estimate the available bandwidth over virtual link $vlink(i,j)$, which is our final goal. So, the formula to calculate the available bandwidth over $vlink(i,j)$:

$$B_{available}(i,j) = L(i,j) - T_{induced}(i,j) - Ti,j - Tj,i$$

becomes:

$$B_{available}(i,j) = L(i,j) - T_{induced}(i,j)$$

1.2.4 Mathematic Analysis

Proposition 1.1 (induced traffic 1) $\forall j,k \in N(i)$, if node j is transmitting to i, then node k can not transmit to any node at the same time.

Proof Please refer to Figure 1.1. It is obvious because, otherwise, the signal will collide at node i.

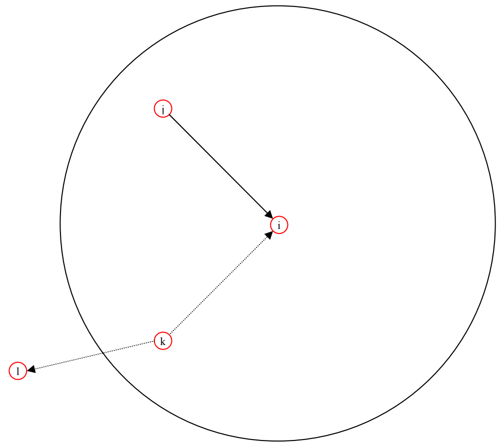

Figure 1.1 Directly induced traffic 1

Proposition 1.2 $\forall j,k \in N(i)$, if node i is transmitting to j, then node k can not receive any signal other than from node i at the same time.

Proof Please refer to Figure 1.2. It is obvious because, otherwise, the signal at node k will collide.

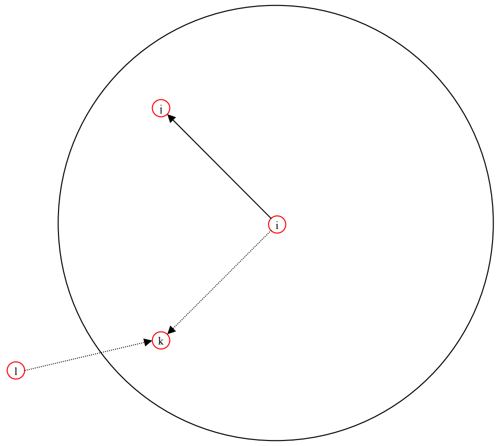

Figure 1.2 Directly induced traffic 2

Proposition 1.3 Specific to IEEE 802.11 DCF, $\forall j, k \in N(i)$, if node j is transmitting to i, then node k can not receive any signal other than from node j at the same time.

Proof Please refer to Figure 1.3. The restriction is imposed by RTS/CTS/NAV scheme, because when node i respond node j with CTS, node k will know the NAV and back-off. So, node k can not receive even it is out the transmission range of node j. [More detail about RTS/CTS/NAV scheme]

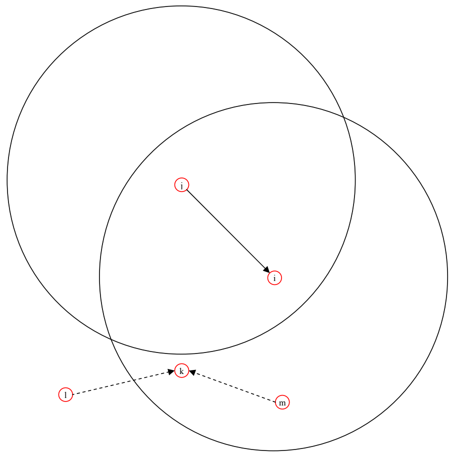

Figure 1.3 Indirectly induced traffic 1

Proposition 1.4 Specific to IEEE 802.11 DCF, $\forall j, k \in N(i)$, if node i is transmitting to j, then node k can not transmit to any node at the same time.

Proof Please refer to Figure 1.4. The restriction is imposed by RTS/CTS/NAV scheme, because when node i send RTS to node j, node k will know the NAV and back-off.

Theorem 1.1 Specific to IEEE 802.11 DCF, $\forall i, j; j \in N(i), i \in N(j)$, if node i is transmitting to j, then:

- $\forall k \in N(i)$, k can not transmit to, nor receive from, any node.
- $\forall l \in N(j)$, l can not transmit to, nor receive from, any node.

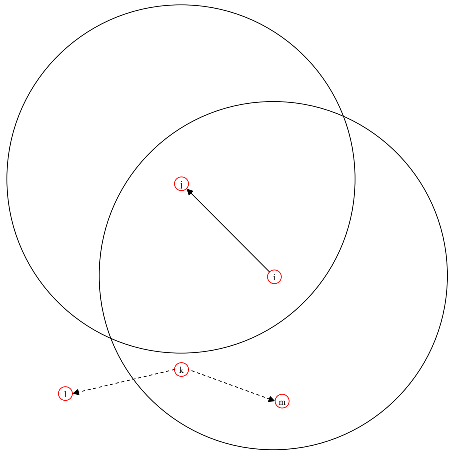

Figure 1.4 Indirectly induced traffic 2

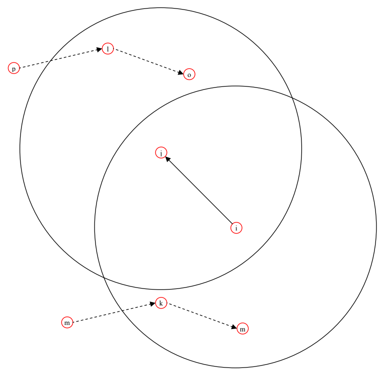

Figure 1.5 Induced Traffic

Proof Please refer to Figure 1.5. This is obvious from Proposition 1, 2, 3, and 4.

Corollary 1.1 Specific to IEEE 802.11 DCF, , if node i is transmitting to j, then:

- $\forall k \in N_2(i)$, k can not transmit to, nor receive from, any node that $\in N(i)$.
- $\forall l \in N_2(j)$, l can not transmit to, nor receive from, any node that $\in N(j)$.

When node i is transmitting to node j, another i's neighbor, k, can not transmit nor receive. That means k lost a certain amount of available bandwidth because of the bandwidth used by virtual link $vlink(i, j)$. From another viewpoint, the induced traffic on virtual link $vlink(i, j)$ includes all the bandwidth used by their neighbors, and under certain circumference, includes the bandwidths used by two-hop or even three-hop neighbors. The following theorem shows how we differentiate the induced traffic and how to calculate the total induced traffic of virtual link $vlink(i, j)$.

Figure 1.6 Induced traffic example

Theorem 1.2

$$T_{induced}(i,j) = \sum_{\substack{\forall k \in N(i) \cup N(j) \\ \forall l}} T(k,l) + \sum_{\substack{\forall k \in N(i) \cup N(j) \\ \forall l \notin N(i) \cup N(j)}} T(l,k) \quad [1.1]$$

Proof We divide the induced traffic into two categories. The first category is all the traffic within $N(i) \cup N(j)$. The second category is all the traffic between $N(i) \cup N(j)$ and all other nodes. All other type traffic has no effect on the virtual link $vlink(i, j)$, i.e. it does not induce any virtual load on $vlink(i, j)$. Because all these traffic, and only these traffic, have effect on the virtual link $vlink(i, j)$, we just add up all these traffic.

Because for a particular node, both the outgoing traffic and the incoming traffic may induce traffic on other nodes' virtual links, it is necessary to consider them all. However, for any two nodes, k and l, both within the combined coverage map of node i and node j, if there is any direct traffic between k and l, the incoming traffic of node k from node l is the same traffic as the outgoing traffic of node l to node k, vice versa. That means we can not count the same traffic twice.

Hence, the first category traffic can be described as:

$$T^1_{induced}(i,j) = \sum_{\substack{\forall k \in N(i) \cup N(j) \\ \forall l \in N(i) \cup N(j)}} T(k,l)$$

And the second category traffic can be described as:

$$T^2_{induced}(i,j) = \sum_{\substack{\forall k \in N(i) \cup N(j) \\ \forall l \notin N(i) \cup N(j)}} T(k,l) + \sum_{\substack{\forall k \in N(i) \cup N(j) \\ \forall l \notin N(i) \cup N(j)}} T(l,k)$$

Note: For $T^2_{induced}(i,j)$, the first item is different with the second item. The first item means all the outgoing traffic from any node within the combined coverage area of node i and node j to any node beyond the combined coverage area. The second item means all the incoming traffic from any node beyond the combined coverage area of node i and node j to the combined coverage area of node i and node j.

Hence:

$$\begin{aligned} T_{induced}(i,j) &= T^1_{induced}(i,j) + T^2_{induced}(i,j) \\ &= \sum_{\substack{\forall k \in N(i) \cup N(j) \\ \forall l \in N(i) \cup N(j)}} T(k,l) + \sum_{\substack{\forall k \in N(i) \cup N(j) \\ \forall l \notin N(i) \cup N(j)}} T(k,l) + \sum_{\substack{\forall k \in N(i) \cup N(j) \\ \forall l \notin N(i) \cup N(j)}} T(l,k) \\ &= \sum_{\substack{\forall k \in N(i) \cup N(j) \\ \forall l}} T(k,l) + \sum_{\substack{\forall k \in N(i) \cup N(j) \\ \forall l \notin N(i) \cup N(j)}} T(l,k) \end{aligned}$$

Note of Theorem 1.2: The first item means the sum of all the outing traffic for any node $k \in N(i) \cup N(j)$, the destination node is insignificant. The second item means the sum of all the traffic from any node $l \notin N(i) \cup N(j)$ to node $k \in N(i) \cup N(j)$

Corollary 1.2

$$T_{induced}(i,j) = \sum_{\substack{\forall k \in N(i) \cup N(j) \\ \forall l}} T(l,k) + \sum_{\substack{\forall k \in N(i) \cup N(j) \\ \forall l \notin N(i) \cup N(j)}} T(k,l)$$

Under certain circumferences, for two different nodes, $j, k \in N(i)$, the available bandwidth on virtual link $vlink(i, j)$ and on virtual link $vlink(i, k)$ can be different. The reason is that the corresponding neighbors of node j and k may generate different amount of the induced traffic on $vlink(i, j)$ and $vlink(i, k)$. The following example, Figure 1.7, says that: a, b, c, d, and e are five equally separated nodes on a line in order with distance 0.8r (r is the transmission range). Although a and c are both b's neighbors, the traffic on $vlink(d, e)$ is considered as induced traffic on $vlink(b, c)$, but not on $vlink(b, a)$.

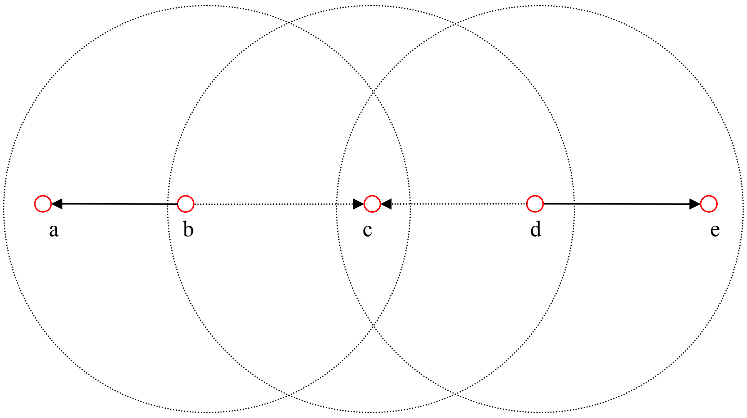

Figure 1.7 Different effects

By theorem 2, we know how to calculate the induced traffic on virtual link $vlink(i, j)$. In order to calculate the induced traffic, we need to know the following information:

- All the nodes $\in N(i) \cup N(j)$
- For each node $\in N(i) \cup N(j)$, the bandwidth usage, which includes both the incoming and the outgoing traffic, where the incoming traffic is used to calculate item 2 in theorem 2, i.e. the sum of the traffic from nodes that $\notin N(i) \cup N(j)$ to nodes that $\in N(i) \cup N(j)$.

We propose the calculation of induced traffic on virtual link $vlink(i, j)$ is performed at node i. So, the calculation of the induced traffic on virtual link $vlink(j, i)$ is performed at node j. However, for IEEE 802.11 DCF, they are the same. So, before we carry out the calculation on node i, we can query node j to see if the induced traffic has been calculated and if it is up-to-date. It is also desirable to update node j after node i finished the calculation.

Every node calculates the induced traffic on the virtual links from itself to each of its neighbors. It is performed independently in a *distributed* way. No global information is needed and no centralized control. It is *scalable* and does not depend on the network size. All we need to know is one-hop and two-hop neighbor's information. From a node's viewpoint, it needs n (the number of neighbors) calculation because it has n neighbors, i.e. n virtual links. However, globally, we only need l (the total number of the virtual links in the whole network) calculations, which is a half of the first method. [Express more clearly]

Here we assume, node i knows:

- For any node $j \in N(i)$, the traffic on virtual link $vlink(i, j)$, i.e. the outgoing traffic on each link.
- For any node $j \in N(i)$, the traffic on virtual link $vlink(j, i)$, i.e. the incoming traffic on each link

Now, in order to calculate $T_{induced}(i, j)$ for $\forall j \in N(i)$, the information node i needs to keep:

- One-hop neighbor list, $N(i)$, which can be obtained by exchanging beacon signals.
- Two-hop neighbor list, $N_2(i)$, which can be obtained by querying node i's neighbors.
- The bandwidth usage of each of its neighbors.

1.3 Performance Analysis

In this section, we evaluate the performance and complexity of deploying induced traffic calculation algorithm to wireless ad hoc networks, which includes the storage, computation and communication complexity.

1.3.1 Storage

The storage space needed for a node to carry out the induced traffic calculation depends on the number of its one-hop and two-hop neighbors. We assume each node has at most n neighbors. We can use either array or linked list data structure to organize the information. We use node i as an example.

1. First of all, the most fundamental information unit is the node information, which could be either one-hop neighbor node or two-hop neighbor node of node i. The node information should include its neighbor list and the traffic information. More specifically, whether it is incoming traffic or outgoing traffic and the traffic amount. The index key is the node ID. The following fields are included:

   - neighbor node ID
   - traffic direction, i.e. incoming or outgoing
   - traffic amount. It means no traffic if it is 0.

2. One-hop neighbor list, which stores the neighborhood node IDs and the traffic information between node i and its neighbors. Basically, this is just a special case of the previous item.

3. Two-hop neighbor list, which includes node IDs that are node i's neighbors' neighbors, but not node i's neighbors. The list can be formed by querying the previous two items. As Figure 1.8 showed, the star nodes are two-hop neighbors of node i.

1.3.2 Computation

The following theorem is derived from Theorem 1.2 to facilitate the induced traffic calculation.

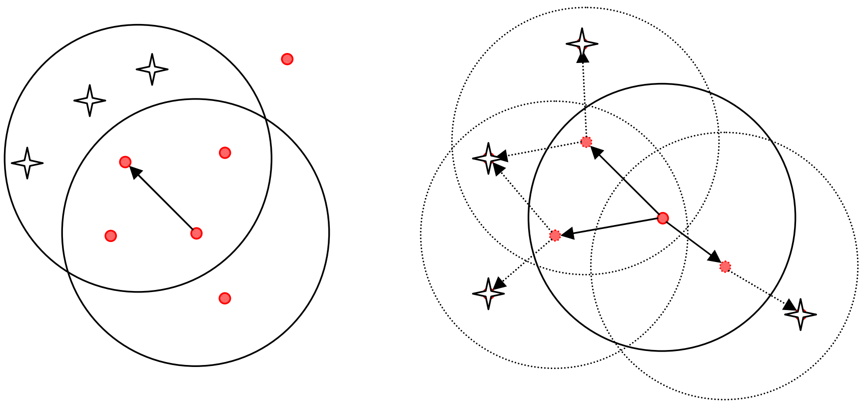

Figure 1.8 Two-hop neighbors

Theorem 1.3

$$T_{induced}(i,j) = \sum_{\substack{\forall k \in N(i) \\ \forall l}} T(k,l) + \sum_{\substack{\forall k \in N(j), k \notin N(i) \\ \forall l \notin N(i)}} T(k,l) + \sum_{\substack{\forall k \in N(i) \\ \forall l \notin N(i)}} T(l,k) + \sum_{\substack{\forall k \in N(j), k \notin N(i) \\ \forall l \notin N(i) \cup N(j)}} T(k,l)$$

Proof

$$\begin{aligned}
T_{induced}(i,j) &= \sum_{\substack{\forall k \in N(i) \cup N(j) \\ \forall l}} T(k,l) + \sum_{\substack{\forall k \in N(i) \cup N(j) \\ \forall l \notin N(i) \cup N(j)}} T(l,k) \\
&= \sum_{\substack{\forall k \in N(i) \\ \forall l}} T(k,l) + \sum_{\substack{\forall k \in N(j), k \notin N(i) \\ \forall l}} T(k,l) + \sum_{\substack{\forall k \in N(i) \\ \forall l \notin N(i) \cup N(j)}} T(l,k) + \sum_{\substack{\forall k \in N(j), k \notin N(i) \\ \forall l \notin N(i) \cup N(j)}} T(l,k) \\
&= \sum_{\substack{\forall k \in N(i) \\ \forall l}} T(k,l) + \sum_{\substack{\forall k \in N(j), k \notin N(i) \\ \forall l \notin N(i)}} T(k,l) + \sum_{\substack{\forall k \in N(j), k \notin N(i) \\ \forall l \in N(i)}} T(k,l) + \\
&\quad \sum_{\substack{\forall k \in N(i) \\ \forall l \notin N(i) \cup N(j)}} T(l,k) + \sum_{\substack{\forall k \in N(j), k \notin N(i) \\ \forall l \notin N(i) \cup N(j)}} T(l,k) \\
&= \sum_{\substack{\forall k \in N(i) \\ \forall l}} T(k,l) + \sum_{\substack{\forall k \in N(j), k \notin N(i) \\ \forall l \notin N(i)}} T(k,l) + \sum_{\substack{\forall l \in N(j), l \notin N(i) \\ \forall k \in N(i)}} T(l,k) + \\
&\quad \sum_{\substack{\forall k \in N(i) \\ \forall l \notin N(i) \cup N(j)}} T(l,k) + \sum_{\substack{\forall k \in N(j), k \notin N(i) \\ \forall l \notin N(i) \cup N(j)}} T(l,k) \\
&= \sum_{\substack{\forall k \in N(i) \\ \forall l}} T(k,l) + \sum_{\substack{\forall k \in N(j), k \notin N(i) \\ \forall l \notin N(i)}} T(k,l) + \sum_{\substack{\forall k \in N(i) \\ \forall l \notin N(i)}} T(l,k) + \sum_{\substack{\forall k \in N(j), k \notin N(i) \\ \forall l \notin N(i) \cup N(j)}} T(l,k)
\end{aligned}$$

The result of Theorem 1.3 shows a different way on how to divide the different induced traffic that affects the virtual link $vlink(i,j)$. For item 1 and item 3, they are only related to node i. That means they are the common components for all the neighbors of node i. For example, node k is another neighbor of node i, in order to calculate the induced traffic over virtual link $vlink(i,k)$, the same information, item 1 and item 3, has to be included. For item 2 and item 4, they are related to the specific neighbor, i.e. node j, or node k. The physical meaning of the four items is obvious:

- Item 1 means the outgoing traffic of node i's neighbors.
- Item 3 means the incoming traffic of node i's neighbors from any node beyond $N(i)$.

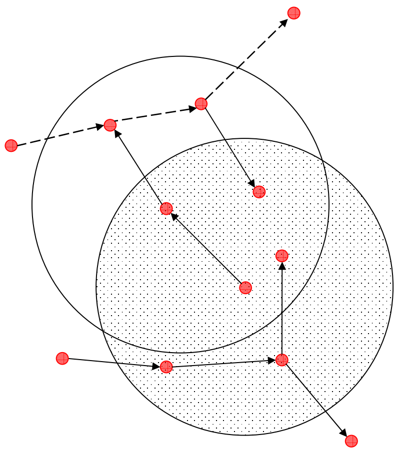

Figure 1.9 Another view of the different categories of the induced traffic

- Item 2 means the outgoing traffic of node j's neighbors, but not i's, to any node beyond N(i)
- Item 4 means the incoming traffic of node j's neighbors, but not i's, from any node beyond $N(i) and N(j)$.

Figure 1.9 shows the four categories of traffic. All the traffic with solid lines belongs to item 1 and 3, while all the traffic with dashed lines belongs to item 2 and item 4.

The computation of item 1 and item 3 is straightforward. To calculate item 2, first we get the list of nodes that are node j's neighbors but not node i's neighbors, which is a part of node i's two-hop neighbor list. Then, for those nodes, we only need to count their outgoing traffic to the nodes that $\notin N(i)$. The calculation of item 4 is similar with the calculation of item 2.

1.3.3 Communication

As we know from the previous section, in order to carry out the calculation of the induced traffic of node i, we need a complete picture of node i's one-hop and two-hop neighbors, which includes their neighbor list and traffic information. The one-hop neighbor information is easy to get because node i can talk to its one-hop neighbor directly. There are two ways to get two-hop neighbor information. One way is to ask node i's one-hop neighbors. Because every node i's two-hop neighbor is one-hop neighbor of a particular node i's one-hop neighbor. Node i's one-hop neighbors have node i's two-hop information. The other way is to establish the communication directly between node i and it's two-hop neighbors. This can only be done after node i has a complete picture of the network topology as far as two-hop away. The second method can be used to exchange traffic information.

Theorem 1.4

LIST OF REFERENCES