DISTRIBUTED CLUSTERING FOR MULTI-HOP WIRELESS NETWORKS

Stefano Basagni

Erik Jonsson School of Engineering and Computer Science
The University of Texas at Dallas
e-mail: basagni@utdallas.edu

SUMMARY

Obtaining a hierarchical organization of a network is a well-known and studied problem of distributed computing. It has been proven effective, for instance, in minimizing the amount of storage for communication information (e.g., routing and multicast tables), optimizing the use of the network bandwidth, distributing resources throughout the network, etc.

In the case of *multi-hop networks*, i.e., wireless networks in which possibly all nodes can be mobile, partitioning the nodes into groups (*clusters*) is similarly important. In addition, *clustering* may be used for controlling the spatial reuse of the shared channel (e.g., in terms of time division or frequency division schemes), for minimizing the amount of routing and control data, as well as for building and maintaining cluster-based *virtual* network architectures.

The notion of cluster organization has been used for multi-hop networks since their appearance (when these networks where often called "multi-hop packet radio networks"). Extensive references can be found, e.g., in [1].

The methods used so far for obtaining physical clustering in multi-hop networks all implement some distributed *greedy algorithm* for finding a set of nodes that act as coordinators of the clustering process (*clusterheads*). Once the clusterheads are selected, clusters are defined by associating each non-clusterhead node (called an *ordinary node*) with a clusterhead following a specific rule. For instance, in the "maximum degree first" approach of Gerla et al. [2], a node with maximum *degree* (i.e., with the maximum number of neighbors) is selected as a clusterhead, then a cluster is formed by that node and all its neighbors.[1] The same procedure is repeated among the nodes not assigned to a cluster yet, until all nodes belong to a cluster. In the "lowest ID first" method, used in Ephremides et al. [3] and in Gerla et al. [4], node IDs are used to choose the clusterheads.

Here, we describe a *Distributed Clustering Algorithm* (DCA) that generalizes the previous approaches by allowing the choice of the clusterheads based on a generic *weight* (a real number ≥ 0) associated to each node: The bigger the weight of a node, the better that node for the role of clusterhead. Our algorithm has the following main advantages, not available in previous solutions:

1. By representing with the weights node mobility-related parameters, we can choose for the role of clusterhead those nodes that are better suited for that role. For instance, when the weight of a node is inversely proportional to its speed, the less mobile nodes are selected to be clusterheads. Since these nodes either do not move or move slower than the other nodes, their cluster is guaranteed to have a longer life, and consequently the overhead associated with the cluster maintenance in the mobile environment is minimized.

2. Weights can dynamically change in time, thus, adapting to changing nodes' conditions. For example, a clusterhead whose transmission range is impaired by its current location, could lower its weight and trigger a cluster reorganization with the following election of the currently "best suited" node to the role of clusterhead.

Furthermore, we outline how to derive bounds on the DCA time and message complexities. The main results, here, are that we achieve optimal message complexity, and we prove that the time complexity of the DCA is bounded by a network parameter that depends on the network *topology* (that may change due to nodes mobility) rather than on the *size* of the network, i.e., the invariant number of its nodes (as in [4]). Preliminary simulation results demonstrate that the combination of the new weight-based clusterhead selection mechanism and the topology-dependent upper bound on the DCA time complexity yields a *logarithmic* upper bound on the DCA time complexity, thus stating the "average case" efficiency of the DCA.

The Distributed Clustering Algorithm

The DCA is executed at each node with the sole knowledge of the executing node's unique identifier, ID, its weight, and the IDs and the weights of its neighbors (namely, the algorithm is distributed and relies only on local information). We also make the following common operational assumptions: *a*) A message sent by a node is received correctly within a finite time (a *step*) by *all* its neighbors, and *b*) Network topology does not change during the algorithm execution.[2] For the

[1] Two "ad hoc" nodes are *neighbors* if they can hear each other's transmissions.

[2] A detailed discussion on how to release these assumptions can be found, e.g., in [4].

sake of simplicity, from now on we will consider that each node has a different weight.

Except for the initial routine (procedure *Init*), the algorithm is message driven: a specific procedure will be executed at a node depending on the reception of the corresponding message. We use only two types of messages: CH(v), used by a node v to make its neighbors aware that it is going to be a clusterhead, and JOIN(v, u), with which a node v communicates to its neighbors that it will be part of the cluster whose clusterhead is node u. *The main idea is that a node decides which role to assume only when all its neighbors with bigger weights have decided their own roles.*

Every node starts the execution of the algorithm at the same time, running the procedure *Init*. Only those nodes that have the biggest weight among all the nodes in their neighborhood will send a CH message ("init nodes"). Given the nature of the weights (real numbers), there always exists at least a node v that transmits the message CH(v). All the other nodes just wait to receive a message. Then, we have the following two message triggered procedures:

- *On receiving* CH. On receiving a CH message from a neighbor u, node v checks if it has received from *all* its neighbors z such that $w_z > w_u$, a JOIN(z, x) message, x. In this case, v will not receive a CH message from these z, and u is the node with the biggest weight in v's neighborhood that has sent a CH message. Thus, v joins u, and quits the algorithm execution (it already knows the cluster to which it belongs, i.e., its clusterhead). If there is still at least a node z, $w_z > w_u$, that has not sent a message yet, node v records that u sent a CH message, and keeps waiting for a message from z.
- *On receiving* JOIN. On receiving a JOIN(u, t) message, node v checks if it has previously sent a CH message. If this is the case, it checks if node u wants to join v's cluster ($v = t$). Then, if all v's neighbors z such that $w_z < w_v$ have communicated their willingness to join a cluster, v quits the execution of the DCA. Notice that, in this case, node v does not care about its neighbors y (if any) such that $w_y > w_v$, because these nodes have surely joined a node x such that $w_x > w_v$ (thus permitting v to be a clusterhead). If node v has not sent a CH message, before deciding what its role is going to be, it needs to know what *all* the nodes z such that $w_z > w_v$ have decided for themselves. If v has received a message from all such nodes, then it checks the nature of the messages received. If they are all JOIN messages, this means that all those neighbors z have decided to join a cluster as ordinary nodes. This implies that now v is the node with the biggest weight among the nodes (if any) that have still to decide what to do. In this case, v will be a clusterhead. At this point, v also checks if each neighbor y such that $w_y < w_v$ has already joined another cluster. If this is the case, v quits the algorithm execution: it will be the clusterhead of a cluster with a single node. Alternatively, if v has received at least a CH message from z, then it joins the cluster of the neighbor with the biggest weight that sent a CH message, and quits the execution of the DCA (a node always quits the algorithm execution as soon as it sends a JOIN message).

As mentioned earlier, in order to decide whether it is going to be a clusterhead or an ordinary node, each node waits for the decision of all the neighboring nodes with bigger weight. This "waiting time" of each node can be defined as a function of the *distance* of a node from one of the init nodes. Clearly, this "blocking distance," D_b, depends on the current topology of the network rather than on the number n of the nodes of the network. By the total ordering induced on the nodes by their weights, and through a simple inductive argument, we can prove the following result:

Theorem 1 *Each node of the network sends exactly one message within $D_b + 1$ steps.*

Since every ordinary node quits the execution of the DCA after sending a message, and since each clusterhead quits it as soon as it has received a JOIN from his neighboring ordinary nodes, it follows that:

Corollary 1 *The DCA terminates within $D_b + 1$ steps.*

This upper bound on the time complexity of the DCA improves on the $O(n)$ upper bound presented in [4] in all those cases in which $D_b \ll n$.

The following corollary states the (optimal, since each node needs to send at least a message) message complexity of the DCA.

Corollary 2 *The message complexity of the DCA is n.*

Detailed proofs, numerical results, and the proofs of correctness and of other properties of DCA-based clustering can be found in [5].

REFERENCES

(1) R. Ramanathan and M. Steenstrup, "Hierarchically-organized, multihop mobile wireless networks for quality-of-service support," *Mobile Networks & Applications*, vol. 3, pp. 101–119, June 1998.

(2) M. Gerla and J. T.-C. Tsai, "Multicluster, mobile, multimedia radio network," *Wireless Networks*, vol. 1, no. 3, pp. 255–265, 1995.

(3) A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," *Proceedings of the IEEE*, vol. 75, pp. 56–73, January 1987.

(4) C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *Journal on Selected Areas in Communications*, vol. 15, pp. 1265–1275, September 1997.

(5) S. Basagni, "Distributed and mobility-adaptive clustering for ad hoc networks," Tech. Rep. UTD/EE-02-98, Erik Jonsson School of Engineering and Computer Science, The University of Texas at Dallas, July 1998.