Gateway Placement in Hybrid MANET-Satellite Networks

Monia Hamdi, Laurent Franck
Telecom Bretagne, MicroWave Department
Site of Toulouse, France
Email:{Monia.Hamdi, Laurent.Franck}@telecom-bretagne.eu

Xavier Lagrange
Telecom Bretagne, RSM Department
Site of Rennes, France
Email: Xavier.Lagrange@telecom-bretagne.eu

***Abstract*—Mobile ad hoc networks (MANETs) are self-organized and dynamically reconfigurable wireless networks. It is a promising technology for emergency scenarios where communication systems should be able to operate in potentially adverse environments. However, MANETs experience severe impairments because of node motion or scarce node density yielding network partitioning. Satellite communications may help in this respect by setting up recovery links.**

In this article, we study the problem of selecting for each network partition a MANET node called gateway that will provide access to the satellite capacity. A key challenge is that nodes are mobile resulting in frequent topology changes. We address this challenge by re-using the mechanisms of clustering techniques initially designed to solve scalability issues in MANETs.

keywords: mobile ad hoc network, k-hop clustering, connectivity recovery, emergency communication, satellite, gateway placement

I. Introduction

The services offered by traditional wireless systems such as cellular networks depend on established infrastructures. In the aftermath of a disaster, communication infrastructures may be totally destroyed. There is a need for mobile technologies being independent of infrastructure where network operation is the result of cooperation among terminals.

In MANETs, terminals self-organize yielding temporary topologies. MANETs display a great potential in emergency and rescue operations because of their instant deployment and reconfiguration capabilities. However, due to frequent topology changes, network partitioning may also occur. The network is then split into unconnected groups. In such a situation, satellite communications may be of a great use to bridge these unconnected islands. On the one hand, MANET nodes - called gateways - in each partition should be elected to offer access to the satellite capacity. On the other hand, a minimal number of gateways should be activated so to save satellite and terrestrial ressources. In the literature, this problem is known as the *Gateway Placement Problem*.

Similar issues exist in wireless mesh networks (WMNs), however the solutions proposed are not designed for a mobile environment [12], [11], [3]. We propose the use of a mechanism called *clustering* which was originally designed to cope with scalability issues in MANETs.

The remainder of this paper is organized as follows. Existing gateway placement approaches are outlined in Section II. The behavior of the clustering algorithm is evaluated and analyzed in Section III. Finally, Section IV concludes this paper.

II. Related Work

A. k - hop Clustering

Originally, the use of clustering techniques was proposed to solve scalability issues in large MANETs [8]. The network is divided into virtual groups of mobile nodes called clusters. A clusterhead is elected among mobile hosts to be the local coordinator. In k-hop clustering, each node is at most k-hop distant from its clusterhead.

We start by describing clustering algorithms and then explain the choice of the most suitable clustering algorithm for our target deployment scenario. MaxMin [1] is the pioneering algorithm in this field. Cluster formation is composed of two phases (FloodMax and FloodMin) and each phase comprises k rounds of message exchanges. MaxMin inspired several later works. The KCMBC algorithm [9], based on MaxMin introduces an expiration time metric to take into account the impact of node mobility.

Unlike MaxMin, k-lowestID and k-CONID [6] rely on clustering request and clustering decision flooding. Huang [7] improves these algorithms by introducing clusterhead back up mechanisms and a weight factor based on link quality. But like its ancestors, it results in broadcast storms.

In DSCAM [2], the cluster-based network structure forms a (r,k)-Dominating Set, where r is the minimum number of clusterheads per node and k is the maximum number of hops between a node and its clusterhead. The construction of an initial dominating set is based on node identifiers.

As already stated, the node properties impact the suitability of a node to serve as satellite access point. Therefore, the approaches where clusterhead selection is not based on inherent node properties are excluded. In addition to that, [9] proves that MinMax-based approaches outperform k-LowestID- based approaches with respect to cluster formation overhead. As a result of the previous analysis, the KCMBC algorithm is selected for this study.

The performance of the KCMBC algorithm has been already simulated and validated in a mobile environment [9]. However, prior works assumed dense networks where each node can communicate over multi-hop paths with any other node of the network. It is the first time a clustering algorithm is evaluated

978-1-4673-1881-5/12/$31.00 ©2012 IEEE

in a scarce-density network where partitioning may occur. These partitions are dynamic, nodes can leave a partition to join another one. Our aim is to analyze the KCMBC behavior in this context and re-use the mechanism of clustering to address gateway placement.

B. KCMBC Overview

The KCMBC algorithm comprises three main steps. The first step is node metric computation using the degree and the so called expiration time. The purpose of this metric is to assess whether nodes will be able to maintain longer connections with their neighbors and therefore be eligible clusterheads. The second step is the clusterhead election based on the FloodMax and FloodMin protocols and the third step is the cluster maintenance where the cluster structure is updated according to nodes joining and leaving the cluster.

During the second step (clusterhead election), each node broadcasts its metric value to the *k*- hop neighbors and upon reception of other nodes metric proceeds with the election of a candidate clusterhead. This decision is then broadcasted back to the *k*- hop neighbors. Various mechanisms are then used to make sure that there is a consensus among nodes as to which node serves as clusterhead.

III. Simulation

A. Network Model

The simulation scenario relies on a specific mobility model called FireMobility describing group motion behavior during forest fighting operations [5]. FireMobility was designed following interviews with French Civil Protection personnel and field guides. This model yields a hierarchical network organization where fire-fighting forces are arranged into 4 intervention groups plus a command car, each group being composed of 1 command car, 4 water tank trucks and 4 firemen pairs yielding a total of 37 network nodes. These nodes are dispatched on a rectangular playground of 1000 m x 1000 m facing and flanking a fire. As the fire moves, groups are re-dispatched so to comply with safety rules and the tactics of fire-fighting. The average node speed is 4.4 m/s. In [5], the characteristics of FireMobility are compared to those of Random Walk and Reference Region Group Mobility.

All nodes are equipped with WLAN devices displaying a communication range from 70 to 100 m. Each simulation is conducted for 10000 s and then repeated 10 times with a different pseudo-random seed. The results are averaged.

B. Results and Analysis

We have run the KCMBC clustering algorithm against the network topology every time a network partition was detected. However before proceeding with the actual evaluation of KCMBC, it has been necessary to measure the characteristics of partitioning (e.g., lifetime) so to tune the functioning of KCMBC. This will be the topic of the next subsection. Furthermore, while running the clusterhead election should not be an issue (the network is considered to be stable during this phase that lasts for about a second), there are concerns about the further maintenance of the cluster integrity. Indeed, it may happen that a clusterhead gets isolated or that other nodes join or leave the cluster impacting significantly its topology. The study of the topology dynamics of a formed cluster and its impact on the maintenance is the second topic covered in this contribution.

1) Validation of the proposed architecture and the use of KCMBC: A first task is to assess the necessity of using satellites for network connectivity recovery by examining the partitioning lifetime distribution with a communication radio range equal to 70 m, which is the worst case.

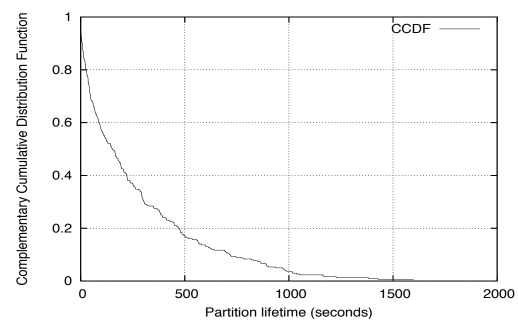

Fig. 1: Complementary Cumulative Distribution Function (CCDF) of the partition lifetime for a radio range of 70 m

Figure 1 reveals that over 60% of partitioning events have a lifetime greater than 100 s and 19% of the network partitioning events have a lifetime greater than 500 s. FireMobility is a group mobility model where the nodes are often clustered and the update of the node positions takes place only in the case of safety distance transgression. If network connectivity is lost, partitioning may last for a long period, hence the need to recover from these long lasting partitioning events.

As pointed out by [12], tactical networks usually deploy a single gateway in each part of the network. It is therefore assumed that one cluster should be defined per partition. In oder to meet this requirement, the parameter *k* in KCMBC corresponding to the maximum number of hops between a cluster member and its clusterhead should be properly tuned. This parameter is bounded by the diameter of network partitions. Figure 2 shows the partition diameter distribution for different radio communication ranges in FireMobility. According to simulation results, *k* is set to the value of 12 so to ensure that one clusterhead per partition is sufficient to meet the requirements of this scenario.

The following paragraphs address the impact of topology changes on the clusters once there are formed.

2) Cluster Maintenance: This part describes the topology changes in a partitioned network, evaluates how KCMBC responds to these events and proposes guidelines for implementing cluster maintenance. Indeed, as shown later, KCMBC was not initially designed to cope with partitioning. It is therefore required to modify or add new mechanisms to KCMBC.

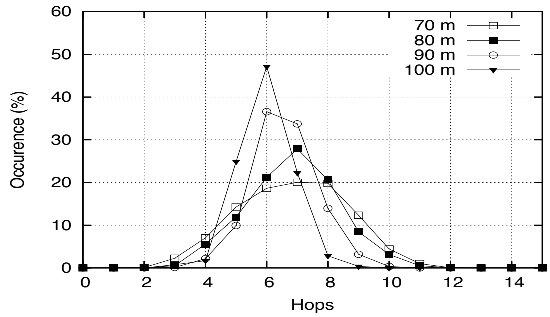

Fig. 2: Distribution of partition diameter as a function of the radio range.

Three cases represent the possible evolution of a network partition: splitting, merging and node migration. For each case, the topology evolution is illustrated, the KCMBC behavior is analyzed and simulation results based FireMobility are presented.

a) Splitting and Merging: Splitting occurs when two groups of nodes, initially located in the same partition, move away from each other and form two different partitions (Figure 3). The nodes, located in a partition where there is no clusterhead, trigger reclustering. In this context, reclustering consists in a new clusterhead election from this subset of so-called orphaned nodes. According to KCMBC design, if an orphan node detects more than d orphan neighbors, all those orphans attempt to trigger a new cluster formation. This rule guarantees that there is at least one clusterhead per partition. However, KCMBC does not detail how to detect the loss of a clusterhead.

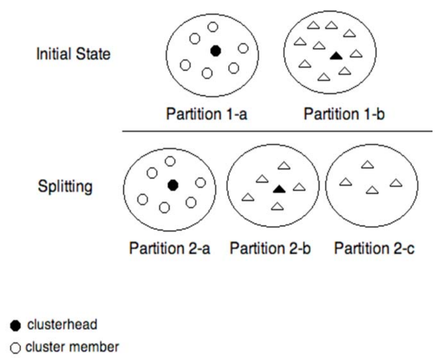

Fig. 3: Splitting: partition 1-b is split in two subsequent partitions.

Partitions may also move toward each other to form a single partition (merging). After merging, full network connectivity may be recovered. In this case, the use of satellite, hence clustering is no more required. The network may also remain partitioned (Figure 4). In KCMBC, two clusterheads upon partition merge, keep their status, unless they become close neighbors. This is contradictory to a functional requirement in this work which is to have only one clusterhead per partition. One of the clusterhead should therefore resign. According to the value of k chosen above, the diameter of the partition resulting from the merging of two partitions is lower than k. If one of the two clusterheads loses its clusterhead status, previous cluster members are still within k hops from the other clusterhead. In order to manage partition merging, a clusterhead should therefore be able to detect the presence of other clusterheads in the same partition.

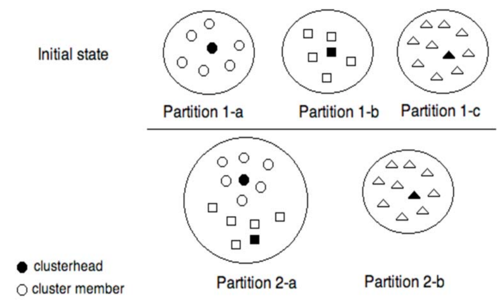

Fig. 4: Merging: partitions 1-b and 1-c merge, but the network is still partitioned.

Before describing the third case (node migration), the next paragraphs show how often splitting and merging happen in the FireMobility scenario. A partitioning event is defined as the splitting of a previously connected topology into several partitions. Splitting and merging occurrence is defined as the amount of partition splitting and merging respectively during a partitioning event. Figure 5 shows that splitting and merging occurrences are approximately similar. During a partitioning event there is a continuous oscillation between splitting and merging. It can also be noticed that the lower the radio range, the more dynamic the topology. For a radio range lower than 80 m, partition merging and splitting occur at least once every partitioning event. As a result, the cluster maintenance guidelines given above in the case of a general network also apply to FireMobility: each clusterhead should detect the presence of other clusterheads in its partition, each cluster member should detect the loss of its clusterhead and orphan nodes should trigger reclustering.

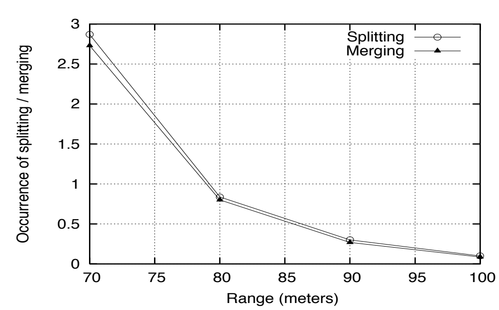

Fig. 5: Average number of splitting and merging events during a partitioning event as a function of the radio range.

The previous paragraphs show that because of partition merging, several clusterheads may be located in the same partition. As already stated, KCMBC does not support in its standard design the detection of multiple clusterheads in a single partition. This may be an important drawback depending on how often it happens in our scenario. In order to assess the magnitude of this phenomena, we define clusterhead diversity occurrence as the ratio between the duration several clusterheads are observed in the same partition and the total partitioning lifetime. The total partitioning lifetime is considered instead of the simulation duration because the clustering procedure is only relevant when the network is partitioned. If the standard maintenance procedure described in KCMBC is applied, Figure 6 shows that during a partitioning event, several clusterheads are observed in the same partition for at least 20% of the time. As a result, the standard maintenance procedure in KCMBC does not totally meet the needs of our scenario.

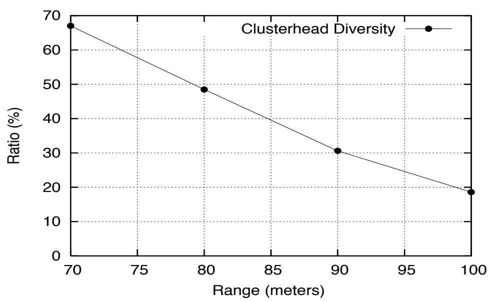

Fig. 6: Clusterhead diversity occurrence as a function of the radio range.

b) Node Migration: Because of the mobility, a node may move to a different partition hence cluster. The node may be a cluster member (Figure 7(a)) that joins an already formed cluster. In KCMBC, if a node loses the path to its former clusterhead and detects a new neighbor (i.e., because of new link), it joins the cluster to which its new neighbor belongs. This assumes that nodes communicate the identifier of their clusterhead to their neighbors. The migrating node may also be a clusterhead (Figure 7(b)). Finally, if a clusterhead has no more affiliated neighbors, it joins another cluster. Otherwise, it keeps its clusterhead status. The latter case is similar to merging, resulting in a partitioned network. A clusterhead should therefore be able to detect the presence of other clusterheads in its partition.

Clusterhead migration is similar to partition merging in regard to the cluster maintenance. On the other hand, cluster member migration is different and its significance in the scenario has to be measured. It is defined as the number of cluster members leaving their partitions during a partitioning event. Figure 8 shows that cluster migration happens on the average twice during a partitioning event for a radio range of 80 m. Consequently the maintenance procedure should make it possible for nodes leaving their partition to detect the neighboring clusters in their new partitions.

In KCMBC, each node includes the identifier of its clusterhead in neighborhood sensing messages called "Hello" messages (one per second in this work). However, to be able to make relevant decision, the information sent by a node to its neighbors should also be updated. For instance, if a node

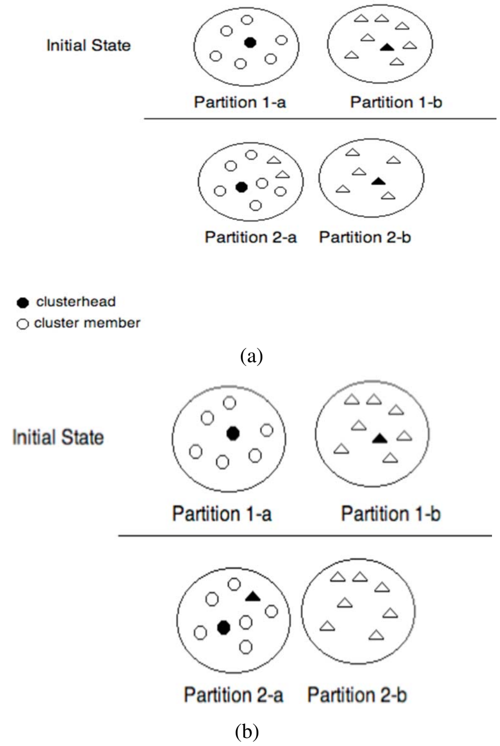

Fig. 7: Node migration: cluster member migration (a), clusterhead migration (b).

leaves its cluster and joins another cluster, the information sent in earlier "Hello" messages is no more topical. Moreover, if a node loses its clusterhead status, its affiliated members have to be informed. The KCMBC authors suppose that each node has the required information to make relevant decisions without specifying the underlying signaling messages.

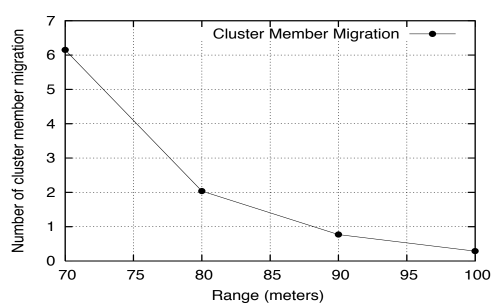

Fig. 8: Average number of cluster member migrations during a partitioning event as a function of the radio range.

As a conclusion, the maintenance procedure in KCMBC does not meet all the needs of the scenario using the FireMobility model. For partition splitting or clusterhead migration, KCMBC does not detail how a node detects the loss of its clusterhead. Furthermore, in case of partition merging, clusterheads keep their status, which transgresses the functional requirement of one clusterhead per partition. This study allows to highlight additional cluster maintenance requirements: each clusterhead should be able to detect the presence of other clusterheads in the partition, each cluster member to detect the loss of its clusterhead and the presence of neighboring clusters and orphan nodes to trigger reclustering.

c) Updated maintenance: We have proposed in [10] a new maintenance procedure together with two companion signaling called Periodical Broadcast and Passive Maintenance. Periodical Broadcast is a signaling protocol proposed by Bellavista in [4]. Passive Maintenance is our protocol optimized for KCMBC. In order to assess the efficiency of this updated maintenance procedure, a metric called cluster inconsistency is defined as the ratio between the duration maintenance rules are violated (e.g., more than one clusterhead exists in a cluster) to the duration of the partitioning event. Figure 9 shows the cluster inconsistency as a function of the radio range. While both protocols display similar performance, the cost in signaling messages is reduced for Passive Maintenance as shown in Figure 10. It is also interesting to node that the updated maintenance procedure performs well even when compared to the only case of cluster diversity occurrence.

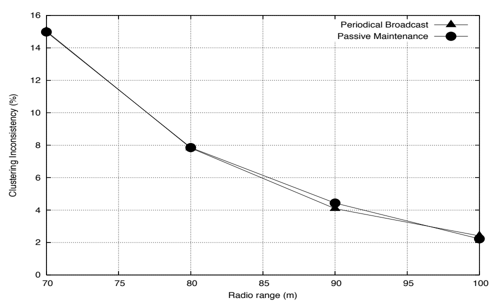

Fig. 9: Inconsistency of clusters as a function of the radio range.

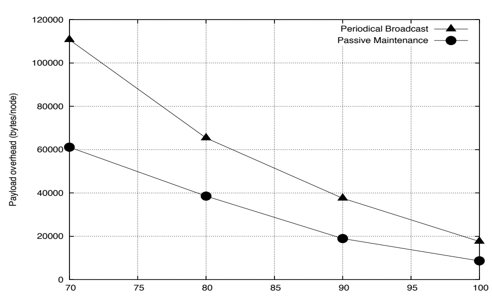

Fig. 10: Signaling overhead (in bytes per node for a total duration of 10000 s) as a function of the radio range.

IV. Conclusion

In this contribution, we address the issue of coping with network partitions in MANETs during emergency situations. The scenario selected is based on forest firefighting operations. Our initial analysis shows that in such a situation, 50% of the partitions last for more than 146 s, calling for the set up of satellite links to bridge unconnected network partitions. However, the nodes hosting these satellite links must be selected so to minimize the economical cost and optimise the network operation.

We propose to use a clustering technique called KCMCB and assess that it is fit for identifying the nodes serving as satellite gateways. However, we also show that in a context where partitioning is highly dynamic, KCMBC does not completely fulfill the requirements of cluster maintenance. The contribution ends with proposals in order to extend KCMBC and meet the requirements of the application.

References

[1] A. D. Amis, R. Prakash, T. H.P. Vuong, and D. T. Huynh. Max-min d-cluster formation in wireless ad hoc networks. *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2000.

[2] V. S. Anitha and M. P. Sebastian. Scenario-based diameter-bounded algorithm for cluster creation and management in mobile ad hoc networks. *13th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications*, 2009.

[3] B. Aoun, R. Boutaba, Youssef Iraqi, and G. Kenward. Gateway placement optimization in wireless mesh networks with qos constraints. *Selected Areas in Communications, IEEE Journal on*, 24(11):2127 – 2136, nov. 2006.

[4] P. Bellavista and E. Magistretti. How node mobility affects k-hop cluster quality in mobile ad hoc networks: a quantitative evaluation. In *IEEE Symposium on Computers and Communications*, 2008.

[5] L. Franck, M. Hamdi, and C. G. Rodriguez. Topology modelling of emergency communication networks: Caveats and pitfalls. *TIEMS (The International Emergency Management Society) Workshop*, 2011.

[6] F. G. Nocetti G. Chen, J. S. Gonzalez, and I. Stojmenovic. Connectivity based k-hop clustering in wireless networks. *Proceedings of the 35th Hawaii International Conference on System Sciences*, 2002.

[7] T. C. Huang, L. C. Shiu, and H.C. Ke. A double-manager k-hop clustering algortihm in mobile ad hoc networks. *Fourth International Conference on Computer and Information Technology*, 2004.

[8] P. Krishna, M. Chatterjee, N. H. Vaidya, and D. K. Pradhan. A cluster-based approach for routing in ad hoc networks. *Proceedings of the 2nd Symposium on Mobile and Location-Independent Computing*, 1995.

[9] S. Leng, Y. Zhang, H. H. Chen, L. Zhang, and Ke. Liu. A novel k-hop compound metric based clustering scheme for ad hoc wireless networks. *IEEE Transactions on Wireless Communications*, 8(1), 2009.

[10] L. Franck M. Hamdi and X. Lagrange. Novel cluster maintenance protocol for efficient satellite integration in manets. *Proceedings of the 29th AIAA International Communications Satellite Systems Conference*, 2011.

[11] J. Robinson, M. Uysal, R. Swaminathan, and E. Knightly. Adding capacity points to a wireless mesh network using local search. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 1247 –1255, april 2008.

[12] Bo Xing, M. Deshpande, S. Mehrotra, and N. Venkatasubramanian. Gateway designation for timely communications in instant mesh networks. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pages 564 –569, 29 2010-april 2 2010.