

Novel cluster maintenance protocol for efficient satellite integration in MANETs

Monia Hamdi* and Laurent Franck †

Télécom Bretagne/Institut Télécom, Toulouse, France

Xavier Lagrange‡

Télécom Bretagne/Institut Télécom, Rennes, France

Mobile ad hoc networks (MANETs) are a promising technology for emergency communications requiring easily deployable and dynamically adaptable topologies. However, MANETs can experience severe impairments because of node motion or scarce node density. The limited availability of communications due to partitioning compromises the rescue and recovery operations. Satellites may help in this respect by setting up temporary back up links. K -hop clustering techniques allow to select the nodes, called *clusterheads* providing access to the satellite communication, while optimizing the network load. However, the selection of clusterheads has to be monitored and maintained.

The overhead in signaling introduced by the cluster maintenance has an important impact on the network performance. When addressing maintenance, most papers in the literature apply a preventive approach consisting in minimizing the number of maintenance events. This is done either by forming clusters that are supposed to be stable or by temporarily relaxing constraints on the cluster structure. Such a strategy raises two issues. First the actual cost of the required maintenance (e.g., in terms of signaling) has never been properly evaluated. Second, it is questionable to assume that a strong clustering policy will mitigate the impairments of maintenance. Indeed, maintenance is often the result of topology changes that are not under control. Finding a tradeoff between the cluster structure relevance and the induced maintenance overhead is not yet investigated. This article proposes a new maintenance policy allowing to minimize the number of exchanged signaling messages.

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 and the remaining operative parts of the network become ineffective and overloaded. Therefore, there is a need for mobile technologies being independent of pre-established infrastructures and where functions such as routing and resource management are the result of spontaneous cooperation among terminals.¹

In MANETs, terminals self-organize and create temporary topologies.² Hence, a MANET has great potential in emergency and rescue operations because of its instant deployment and reconfiguration capabilities.

Due to terminal mobility and dynamic topology changes, network partitioning may occur. The network is then split into separate and unconnected groups. In such a situation, satellite communications may be of a great use to bridge these unconnected islands. Once partitioning is detected, the nodes select a subset of terminals that will provide access to satellite communication. The problem is to choose these nodes in a distributed manner without a centralized control. From this standpoint, that issue is similar to an already developed technique called *clustering*.

The network is divided into virtual groups of mobile nodes called clusters. A clusterhead is elected among mobile hosts to be the local coordinator. We adopt an approach based on the k -hop clustering algorithm

*PhD Student, MicroWave Department, Monia.Hamdi@telecom-bretagne.eu.

†Prof, MicroWave Department, Laurent.Franck@telecom-bretagne.eu.

‡Prof, RSM Department, Xavier.Lagrange@telecom-bretagne.eu.

called KCMBC (K-hop Compound Metric Based Clustering).³ The KCMBC algorithm takes into account link expiration in order to keep the cluster structure as stable as possible and avoid frequent re-affiliations. This property is computed for each link and represents the expected amount of time in seconds before this link disappears. Therefore, the nodes that are able to maintain longer connections with their neighbors are more likely to be selected as clusterheads.

All clustering algorithms impose several primary constraints defining the relevance of the cluster structure. However, as time goes by, node mobility results in the violation of these constraints. Cluster maintenance is consequently needed in order to cope with the node mobility and guarantee the consistency of the cluster structure.

Cluster maintenance assumes that the nodes, by means of message exchanges, cooperate in order to detect and react to cluster structural changes. If the number of exchanged messages is not adequately tuned, the signaling overhead for the cluster maintenance increases drastically, resulting in network performance degradation. Existing low-maintenance clustering schemes aim to guarantee relevant cluster structure while minimizing cluster maintenance cost. They apply preventive approaches consisting in minimizing the number of maintenance events. But the actual cost of the required maintenance has never been properly evaluated. Cluster maintenance in these proposals relies on periodic clusterhead state broadcast.

Our formulation contrasts with prior work in that we propose to minimize the explicit control message by piggybacking cluster information in HELLO packets, that are already sent by KCMBC for neighborhood monitoring. Moreover, the dense MANET assumption, usually adopted in the literature implying a full connectivity in the network, simplifies the design of protocols and applications. We consider more challenging environments where the topology dynamics may result in a partitioned network. Our contribution addresses those changes including the merging and splitting of partitions and their impact on cluster maintenance.

The remainder of this paper is organized as follows. The works related to cluster maintenance are outlined in Section II. The proposed maintenance algorithm is described in section III. Section IV presents a thorough quantitative evaluation of the proposal, while conclusive remarks and directions of ongoing research end the paper.

II. Cluster Maintenance Related Works

Most low-maintenance clustering protocols address the maintenance overhead issue by avoiding maintenance events, i.e., by creating stable clusters. By preserving the current cluster structure as long as possible and minimizing re-clustering and re-affiliation, the cluster maintenance overhead can be reduced accordingly. Grouping nodes with low relative mobility with respect to each other into the same cluster minimizes the impact of the node movement on the cluster structure.⁴ SCA⁵ achieves the overhead reduction by releasing some clustering constraints. When two clusterheads move into one hop apart, instead of immediate cluster merging, the authors propose to ensure that one cluster is a subset of another cluster, in order to avoid rippling effects.

In order to simplify maintenance, most of clustering algorithms focus on one-hop clustering where nodes are at most single-hop distant from clusterheads. Thus, several papers propose a detailed description of the exchanged messages, in regards with one-hop clustering. In FWCA,⁶ the clusterhead is responsible for the maintenance and consulted in every decision. Node affiliation to the cluster implements a four-way handshake mechanism with a *join request* by the node, a *welcome* or a *non-welcome* response by the clusterhead, a *join confirmation request*, and finally an *acceptance acknowledgment*. Moreover, a *leave* message announces the node departure to the clusterhead. Accordingly, each clusterhead maintains a node table wherein the information about its cluster members is stored. Since the clusterhead is consulted for every maintenance decision, FWCA creates a localized bottleneck, especially in highly mobile environments.

Similarly, k -hop clustering proposals focus solely on the preservation of the cluster properties regardless of the produced amount of signaling packets. K -CONID⁷ is a k -hop clustering algorithm that considers four cases modifying the cluster structure in the presence of topological changes: (i) a node switches on; (ii) a clusterhead switches off; (iii) link outage between two nodes belonging to the same cluster; (iv) the distance hop count between two clusterheads is lower than k . The main objective of this maintenance proposal is to cope with the cluster topology changes. The description of the exchanged messages and their impact on the network performance are not covered.

G. Venkataraman et. al.⁸ extend to k -hop clusters the work done for cluster maintenance in FWCA. They provide a detailed description for the required messages to deal with new node arrival, node leaving and

cluster merging and splitting. Moreover, all nodes within the same cluster are notified only upon a new node arrival or a previous member departure. This allows to reduce the maintenance signaling in comparison with k -CONID where the disappearance of a link may lead to a notification broadcast. The work of G. Venkataraman et. al describes accurately the exchanged messages but the impact of the cluster maintenance overhead in terms of network load is not covered.

The first work that proposes to study different aspects of the cluster maintenance independently of the cluster formation is the work of P. Bellavista et. al.⁹ They investigate the effect of the node mobility on the cluster maintenance quality. Moreover, compared to the already presented maintenance schemes, their proposal produces the lowest signaling overhead, since it is based only on periodical node state broadcast from the clusterheads. Their work will be designated as *Periodical Broadcast*.

All of these clustering algorithms require that the nodes send repeatedly explicit clustering messages to build and maintain the cluster structure. Differently from these algorithms, our contribution inserts additional cluster information in the existing HELLO packets and uses on-demand clustering control messages. This new method will be designated as the *Passive Maintenance* and will be compared to the Periodical Broadcast.

III. Periodical Broadcast and Passive Maintenance

In this work, the network may be partitioned resulting in several unconnected groups. For functional reasons, it is required to have one clusterhead per partition. In order to cope with the topology dynamics in such a challenging environment, the resulting maintenance guidelines are as follows: each clusterhead should be able to detect the presence of other clusterheads in its partition, each cluster member to detect the loss of its clusterhead and the presence of neighboring clusters and orphan nodes trigger a new cluster formation. Periodical Broadcast will be first presented and then our proposal called Passive Maintenance is described.

III.A. Periodical Broadcast

Periodical Broadcast is modified as follows so to match pur requirements. The clusterhead selection is based on expiration time in accordance of the operation of KCMBC. The primary characteristics of the maintenance procedure remain unchanged.

Each clusterhead broadcasts a state REFRESH in its k -hop neighborhood every T_{ref} seconds. Each node maintains a clusterhead table, containing the identifier and the expiration time of all the clusterheads within the k -hop neighborhood. A timeout field is associated with each table entry. Upon the receipt of a REFRESH message, a node updates its clusterhead table.

Due to node mobility, three cases may occur:

1. A cluster member does not receive the REFRESH message from its clusterhead: if the entry timeout expires, it triggers a new cluster formation.
2. A clusterhead receives a REFRESH message from another clusterhead: if its expiration time is lower than the received one, it resigns as clusterhead and becomes a cluster member.
3. A node is associated with a clusterhead (CH_A) and receives a REFRESH message from another clusterhead (CH_B): if the expiration time of the CH_B is greater than the expiration time of CH_A , the node changes its cluster affiliation.

The timeout value (T_{out}) will be used to detect the partition splitting and/or the clusterhead migration. If a node does not receive any REFRESH message, it means that there is no clusterhead in the partition and a new cluster formation ought to be triggered. Moreover, the three cases mentioned above confirm that Periodical Broadcast complies with the maintenance guidelines mentioned above.

III.B. Passive Maintenance

Passive Maintenance aims at minimizing the amount of explicit control messages by piggybacking cluster information in HELLO packets. In KCMBC, each node sends a HELLO message every *Hello_Period* seconds. This message includes the clusterhead identifier, the hop count toward the clusterhead, and the node position (for expiration time computing). In KCMBC, the HELLO messages are used for the neighborhood discovering and the host/link activation management.

This contribution uses the information included in HELLO messages to distribute clusterhead information beyond direct neighborhood. It adds an extra field describing a time stamp ($ch_last_sent_hello_msg$). This time stamp represents the last known time the clusterhead has sent a HELLO message and is associated for each node with a timer.

The use of the time stamp in this method assumes that the nodes are time synchronous. Moreover, the rate of cluster information update becomes correlated with HELLO packet period because of piggybacking. This is not considered harmful since both HELLO packets and cluster information are driven by the same phenomenon: node motion.

The concept of cluster consistency is defined from two conditions:

1. Any node should be associated with one clusterhead.
2. In each partition, only one node assumes the role of the clusterhead.

Node mobility tends to transgress these rules. After the splitting of existing partitions or a clusterhead leaves its cluster (i.e., migration), a group of nodes becomes unconnected to its clusterhead. The partition merging may result in several clusterheads in the same partition. In order to ensure cluster integrity, the Passive Maintenance uses, dedicated on-demand clustering control messages: CH_REQ and CH_REPLY. In the next three paragraphs, we will describe the operation of Passive Maintenance where two classes of topology changes are distinguished: clusterhead disconnection and cluster aggregation.

III.B.1. Clusterhead Disconnection

Clusterhead disconnection denotes the fact that the clusterhead and a group of its cluster members are in disjointed partitions. As already stated, the timeout value (T_{out}) will be used to detect this situation. Periodically, each node checks the $ch_last_sent_hello_msg$ value. If the difference between the current time and the former parameter value exceeds T_{out} seconds, the node assumes a possible a clusterhead disconnection. Because of the node mobility, false timeout may occur. In order to cope with this problem and avoid unnecessary re-clustering, the node broadcasts a clusterhead request (CH_REQ). If the clusterhead is not actually disconnected, it will respond to this request by sending a clusterhead reply (CH_REPLY). If the node does not receive any CH_REPLY, the clusterhead disconnection is confirmed and a new cluster formation is triggered.

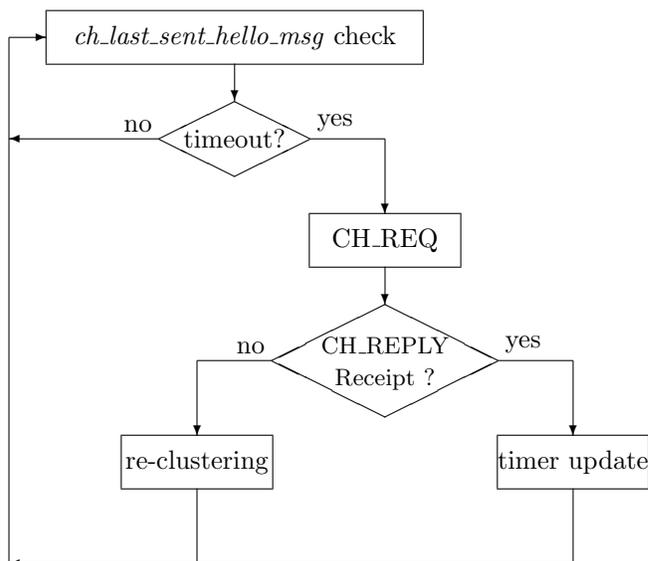


Figure 1. Flow chart of the clusterhead disconnection.

III.B.2. Cluster Aggregation

The cluster aggregation denotes the fact that several clusterheads are in the same partition. By means of HELLO messages, each node is aware of the clusterhead of its neighbors. If a node detects more than d neighbors affiliated to a different cluster, it broadcasts in the cluster a CH_REQ message. A clusterhead receiving this request will respond by sending CH_REPLY. The former message comprises the expiration time of the clusterhead. As a result, if several clusterheads are in the partition, the nodes will be able to select the one with the highest expiration time.

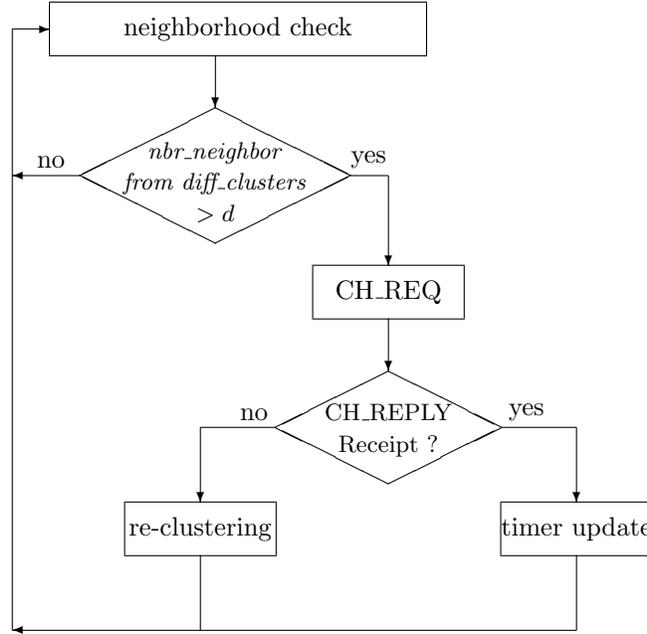


Figure 2. Flow chart of the cluster aggregation.

With both clusterhead disconnection and cluster aggregation, several nodes may send quasi-simultaneously CH_REQs. To avoid broadcast storms, a node is allowed to send or relay only one CH_REQ during a period of *Hello_Period* seconds. Similarly, a clusterhead is allowed to send only one CH_REPLY during a period of *Hello_Period* seconds. In addition, simultaneous CH_REQs may result in packet collision. To tackle this problem, a node u waits for a backoff interval ($T_{backoff}$) before sending a CH_REQ. In this work,

$$T_{backoff} = ID_u/N,$$

where ID_u is the identifier of the node u and N is the network size..

III.B.3. Parameter Initialization

The aim of this phase is the initialization of the *ch_last_sent_hello_msg* value. In the aftermath of the network partitioning, the nodes trigger the cluster formation at time t_0 . The first HELLO message sent by the clusterhead at time $t_0 + \Delta t$ needs a period of time to be spread in the cluster and reach the nodes in the cluster border. In order to avoid false timeout, the initial value of the time stamp, denoted *ch_last_sent_hello_msg_0* should be a function of the distance hop count between the node and its clusterhead.

The clusterhead selection procedure, Max-Min,¹⁰ featured by KCMBC allows each node to be aware of the distance hop count toward its clusterhead.³ This hop count will be denoted *hop_count*. Therefore,

$$ch_last_sent_hello_msg_0 = t_0 + hop_count \times Hello_Period.$$

As a conclusion, the on-demand messages allow the Periodical Broadcast to comply with the maintenance requirements mentioned in the beginning of this section. The CH_REQ and the CH_REPLY messages allow

to check the presence of the clusterheads and confirm the clusterhead disconnection and/or the cluster aggregation.

Finally, the protocol has been formalized using PetriNets so to ensure that properties such as liveness (absence of deadlocks) and reversibility (home state) are fulfilled.

IV. Simulation results

A specific mobility model whose aim is to mimic group motion behavior during a forest fire-fighting mission is used in this article.¹¹ The clustering algorithm KCMBC, the Periodical Broadcast and the Passive Maintenance protocols are implemented in a discrete event simulator. Each simulation set lasts for 10000 s and results are averaged over ten repetitions with different random seeds. Table 1 summarizes the simulation parameters.

Parameter	Value
Number of nodes	37
Average node speed (m/s)	4.4
Average node degree	5.7
Playground size ($m \times m$)	1000 \times 1000
Radio range (m)	[70, 100]
HELLO & REFRESH period (s)	1
k	12
Simulation duration (s)	10000

Table 1. Simulation parameters.

IV.A. Performance Comparison

Both Passive Maintenance and Periodical Broadcast do not continually ensure an accurate clustered topology. The consistency requirements mentioned in paragraph III.B can be temporarily violated. So we define the term of clustering inconsistency to indicate the percentage of time during which these conditions are not met. This measure has a relevant meaning only when the network is partitioned.

A node can trigger re-clustering, if it detects more than D_r orphan neighbors. In case of cluster aggregation, if a node has more than d neighbors affiliated to different clusters, it broadcasts a CH_REQ. The clusterhead disconnection is detected by means of the timeout parameter (T_{out}). For protocol performance comparison, we set appropriately input parameters so that both maintenance schemes offer the same clustering inconsistency level. Table 2 shows the input parameters and Fig. 3 represents the clustering inconsistency level as a function of radio range.

Parameter	Periodical Broadcast	Passive Maintenance
D_r	1	1
d	not defined	1
T_{out} (s)	3	2

Table 2. Model input parameters.

The messages include CH_REQ and CH_REPLY for Passive Maintenance and REFRESH messages for Periodical Broadcast. It includes also the overhead induced by re-clustering situations. In order to assess both schemes based on the signaling load, two standpoints are adopted: header and payload overhead. The header overhead is induced by the supporting networking protocol (e.g., UDP and IP) and is directly related to the number of sent messages, whatever the size. The payload overhead is related to the size of the payload (e.g., the information in a REFRESH message) and therefore does not account for the header size. Fig. 4 shows how the number of messages sent per node varies according to the radio range. The reported results

show that Passive Maintenance outperforms Periodical Broadcast by up to 75% for radio range equal to 70 m and up to 84% for radio range equal to 100 m . This is due to the design of Passive Maintenance which makes it possible to avoid useless transmission when the cluster is still in a stable state. Moreover, should the signaling load be kept equal for Periodical Broadcast and Passive Maintenance, the latter would display a lower inconsistency level.

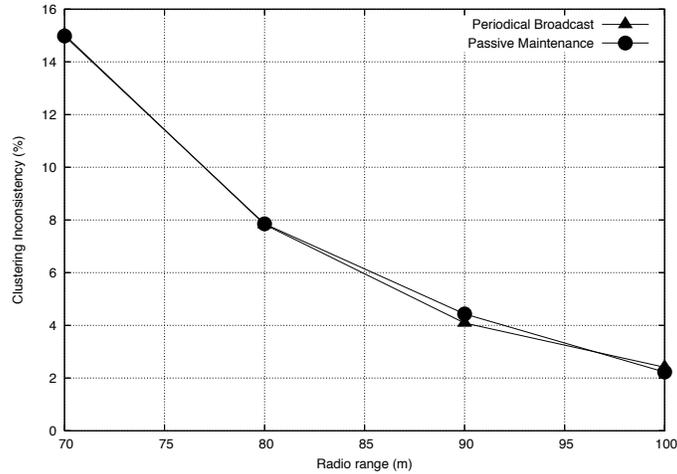


Figure 3. Clustering inconsistency level as a function of the radio range.

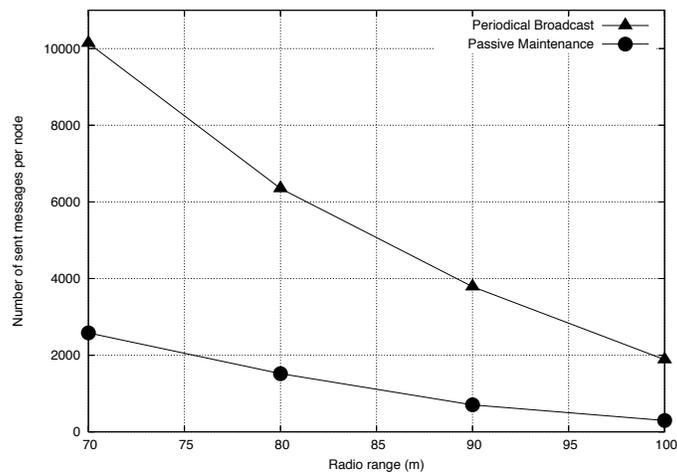


Figure 4. Periodical Broadcast and Passive Maintenance performance comparison with regards of the number of sent messages per node (during the whole simulation). The messages taken into account are: REFRESH (Periodical), CH_REQ, CH_REPLY (Passive), KCMBC messages and HELLO (both).

For payload overhead, we consider the type of transmitted data and the resulting size in bytes. For the Passive Maintenance, there is a supplementary HELLO overhead because it uses an additional field (the time stamp). Fig 5 shows that Passive Maintenance outperforms Periodical Broadcast by up to 45% for a radio range equal to 70 m and up to 51% for a radio range equal to 100 m .

With both metrics, the Passive Maintenance introduces less overhead than the Periodical Broadcast. However, the gap is more important when considering header overhead. Large maintenance overhead consumes an important portion of the network bandwidth and drains quickly node energy. Hence, the passive maintenance approach improves the performance and the lifetime of the network.

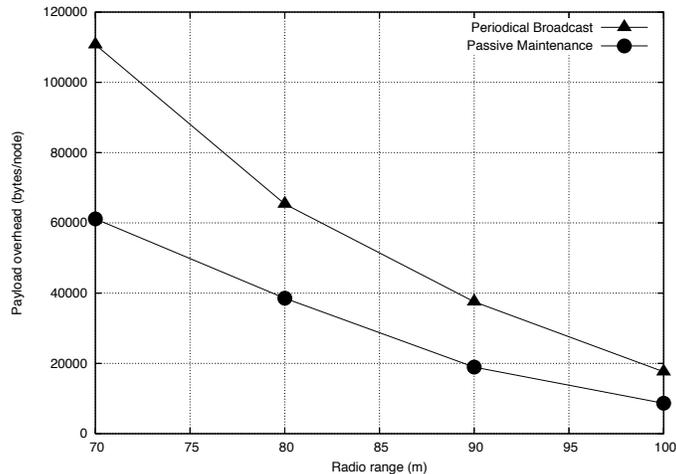


Figure 5. Periodical Broadcast and Passive Maintenance performance comparison with regards of the payload overhead. The messages taken into account are: REFRESH (Periodical), CH_REQ, CH_REPLY (Passive), KCMBC messages and HELLO (both).

IV.B. Evaluation of Passive Maintenance

The consistency of the k -hop clustering depends mainly on the D_r , d and T_{out} values. The rest of the section is devoted to the evaluation of these dependence relationships in Passive Maintenance. We set the radio range to 70 m .

First, we compare how D_r and d affect the inconsistency level. Fig. 6(a) shows that regardless of the T_{out} value, inconsistency level increases as d grows. This can be explained by the fact the nodes cannot detect cluster aggregation if d is large. Similarly, by increasing D_r (Fig. 6(b)), the cluster topology is less consistent since the nodes cannot trigger re-clustering.

However, it is possible to observe that D_r has a greater deteriorating effect than d . For instance, for $T_{out} = 4s$, inconsistency rises from 21.6% to 36.4% (+14.8%) when d shifts from 2 to 4, while it rises from 19.1% to 63.7% (+44.6%) when D_r shifts similarly from 2 to 4. This can be explained by the use of entry timeout. Even if nodes cannot detect the presence of neighbors affiliated to different cluster (large d), the T_{out} parameter will be useful in this case. Nodes that join another cluster are disconnected from their clusterhead and as the timer expires, they send CH_REQ and the cluster topology is again consistent.

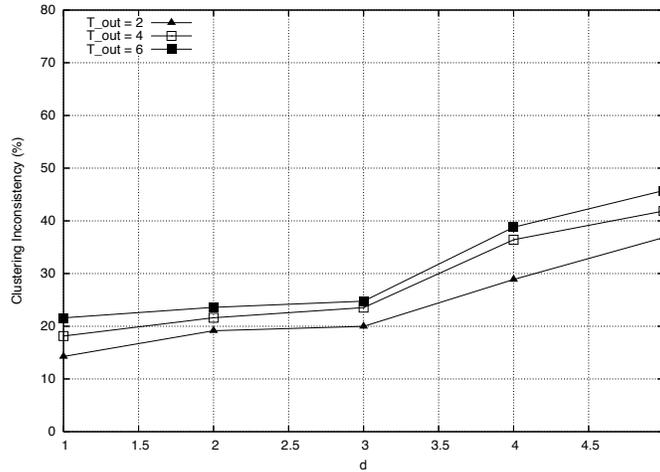
Properly setting d and D_r depends first on the scenario requirements and then on the most appropriate tradeoff between consistency and maintenance overhead since decreasing d and D_r generates more signaling data.

The second part of this section deals with the T_{out} parameter. Tuning this parameter depends not only on maintenance overhead and clustering inconsistency but also on false clusterhead disconnection defined as the amount of false timeouts over the total timeouts. Figure 7 indicates that by increasing T_{out} , false clusterhead disconnections subsequently decrease. This is due to the fact that a large T_{out} helps to cope with node mobility effect. It is interesting to notice that by setting T_{out} to 6 s , we can attain 6.4% of false clusterhead disconnection instead of 20.6% if we set T_{out} to 5 s and the maintenance overhead decreases by 94 messages per node (from 1541 to 1447). In the same time, we increase inconsistency only by 0.56% (from 21.04% to 21.6%).

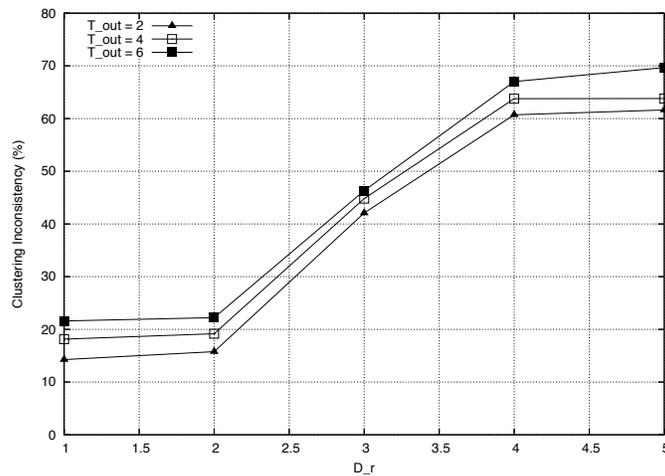
In summary, the inconsistency level tends to decrease along with decrease of d , D_r and T_{out} . But the achieved gain costs more maintenance overhead and false timeouts. The most suitable tradeoff is determined by properly tuning the protocol parameters according to deployment scenario and mobility model.

V. Conclusion

In this article, we proposed a clustering-based approach for satellite and terrestrial network integration for emergency deployment scenario. Cluster maintenance tackles cluster topology changes but it may introduce a large communication overhead. We proposed a novel cluster maintenance protocol that minimizes



(a)



(b)

Figure 6. Clustering inconsistency level as a function of d (a) and D_r (b).

communication overhead so as to save bandwidth and energy.

In this protocol, called Passive Maintenance, cluster information is piggybacked in the HELLO messages used by the clustering algorithm KCMBC. The performance of our protocol, Passive Maintenance, was compared to the widely used maintenance protocol, Periodical Broadcast. Simulation results shows that Passive Maintenance is less resource consuming than Periodical Broadcast for an equivalent quality of service. As part of future work, we will further investigate the impact of the proposed maintenance procedure on the traffic routing since clusterhead handover means routing table change.

References

- ¹Gyoda, K., Nguyen, N. H., Okada, K., and Takizawa, O., "Analysis of Ad Hoc Network Performance in Emergency Communication Models," *22nd International Conference on Advanced Information Networking and Applications*, 2008.
- ²Perkins, C. E., *Ad Hoc Networking*, Addison-Wesley, 2001.
- ³Leng, S., Zhang, Y., Chen, H. H., Zhang, L., and Liu, K., "A Novel k-Hop Compound Metric Based Clustering Scheme for Ad Hoc Wireless Networks," *IEEE Transactions on Wireless Communications*, Vol. 8, No. 1, 2009.
- ⁴Zhang, Y. and Ng, J. M., "A Distributed Group Mobility Adaptive Clustering Algorithm for Mobile Ad Hoc Networks," *IEEE Communications Society*, Vol. 32, 2008, pp. 189–202.
- ⁵Liu, K., Su, J., Zhang, J., Liu, F., and Gong, C., "A Novel Stable Cluster Protocol for Mobile Ad Hoc Networks," *IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*, Vol. 2, 2005, pp. 1328 – 1332.

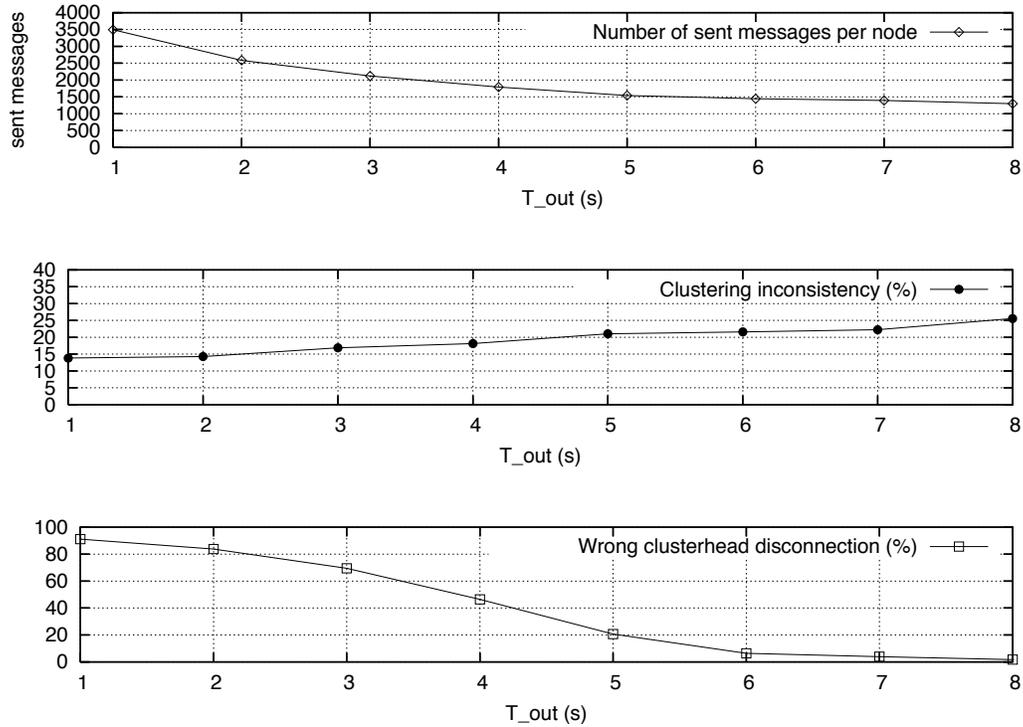


Figure 7. Dependence on T_{out} ($d = 1, D_r = 1$).

⁶El-Bazzal, Z., Kadoch, M., Agba, B. L., Gagnon, F., and Bennani, M., “A Flexible Weight Based Clustering Algorithm in Mobile Ad hoc Networks,” *International Conference on Systems and Networks Communications, ICSNC '06*, 2006.

⁷G. Chen, F. G. N., Gonzalez, J. S., and Stojmenovic, I., “Connectivity Based k-hop Clustering in Wireless Networks,” *Proceedings of the 35th Hawaii International Conference on System Sciences*, 2002.

⁸Venkataraman, G., Emmanuel, S., and Thambipillai, S., “A Novel Distributed Cluster Maintenance Technique for High Mobility Ad-hoc Networks,” *1st International Symposium on Wireless Communication Systems*, 2004.

⁹Bellavista, P. and Magistretti, E., “How Node Mobility Affects k-hop Cluster Quality in Mobile Ad Hoc Networks: a Quantitative Evaluation,” *IEEE Symposium on Computers and Communications*, 2008.

¹⁰Amis, A., Prakash, R., Vuong, T., and Huynh, D., “Max-min d-cluster formation in wireless ad hoc networks,” *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Vol. 1, 2000, pp. 32–41 vol.1.

¹¹Franck, L., Hamdi, M., and Rodriguez, C. G., “Topology Modelling Of Emergency Communication Networks: Caveats And Pitfalls,” *TIEMS (The International Emergency Management Society) Workshop*, 2011.