Smart and Balanced Clustering for MANETs

Luís Conceição and Marilia Curado

Dept. Informatics Engineering, Centre for Informatics and Systems,
University of Coimbra
{lamc,marilia}@dei.uc.pt

Abstract. Clustering is the most widely used performance solution for Mobile Ad Hoc Networks (MANETs), enabling their scalability for a large number of mobile nodes. The design of clustering schemes is quite complex, due to the highly dynamic topology of such networks. A numerous variety of clustering schemes have been proposed in literature, focusing different characteristics and objectives. In this work, a fully distributed and clusterhead-free clustering scheme is proposed, namely Smart and Balanced Clustering for MANETs (SALSA). The scheme introduces a new cluster balancing mechanism and a best clustering metric, aiming to provide a reduced maintenance overhead. SALSA was evaluated and compared with the Novel Stable and Low-maintenance Clustering Scheme (NSLOC), featuring topologies with up to 1000 nodes and velocities of 20 meters per second. Results confirmed the performance efficiency of the new scheme, providing stability and low maintenance overhead, even in the largest networks.

Keywords: MANET, distributed clustering, mobility, stability, large networks

1 Introduction

With the evolution of wireless technologies, there has been an increasingly wide utilization of mobile devices. Mobile networks have become particularly attractive in the recent years due to their flexibility at considerable low costs. Wireless is indeed one of the nominated communication technologies of the future, since it has the potential to allow the connection of all types of mobile devices.

MANETs are autonomous systems, capable of self deployment and maintenance, not requiring infrastructure support for their operation. As a result, the topology of such networks is very dynamic, especially due to the unpredictable behavior of the nodes involved. In this context, numerous clustering schemes were developed, following different approaches and objectives, such as stability, low maintenance overhead or energy efficiency. Each one attempts to obtain the best efficiency by varying the characteristics of the system, like the usage of clusterheads and gateways, the maximum hop distance between nodes and the location awareness. However, to the best of our knowledge, there is only one clustering scheme aiming at providing a fully distributed cluster structure with no clusterheads, namely Novel Stable and Low-maintenance Clustering Scheme

(NSLOC) [1]. This work, proposes an evolution of the NSLOC algorithm, attempting to further improve its performance, by reducing the control overhead. To accomplish this goal, SALSA introduces a new cluster balancing mechanism and a best clustering metric, capable of choosing suitable joining clusters.

The rest of this document is organized as follows. Section 2 discusses the related work. Section 3 describes the SALSA clustering scheme. Section 4 performs an evaluation of SALSA and, finally, Section 5 concludes the article.

2 Related Work

Clustering algorithms can be classified according to different characteristics and objectives [2]. One of the common features in clustering schemes is the utilization of clusterheads (CH) and most of the proposed schemes rely on centralized nodes to manage the clusters structure. The utilization of gateway (GW) nodes is also another important characteristic that is present in the majority of clustering schemes. Other properties of clustering schemes concern the single-hop or multi-hop environments, the multi-homing (MH) support, embedded routing capabilities and location awareness.

Combining the possible characteristics, each proposed clustering scheme attempts to accomplish a specific objective. The Stable Clustering Algorithm (SCA) [3] aims at supporting large MANETs containing nodes moving at high speeds by reducing re-clustering operations and stabilizing the network as long as possible. To meet these requirements, the algorithm is based on the quick adaptation to the changes of the network topology and reduction of clusterhead reelections. In order to avoid a high frequency of clusterheads reelection, the algorithm initially chooses the nodes that best meet some required metrics such as, energy, mobility, connectivity and communication range. The Stability-based Multi-hop Clustering Protocol (SMCP) [4] also builds the cluster structure according to the node connectivity quality. Moreover, this scheme introduces a new methodology (clustercast mechanism) with the purpose of limiting the broadcast of less significant control messages. The K-hop Clustering Protocol (KhCP) [5] protocol is specifically designed to cluster dense MANETs, as it delimits the cluster formation at a specified k-hop distance. In this protocol, clusters are formed on a circle basis, whereas the clusterhead, at the start point, is the centre of the circle.

A weight-based clustering scheme, named Distributed Weighted Clustering Algorithm (DWCA), was proposed with the objective to extend the lifetime of the network, by creating a distributed clustering structure [6]. The election of clusterheads is based on the weight value of nodes, which is calculated according to their number of neighbors, speed and energy. The Enhanced Performance Clustering Algorithm (EPCA) [7] is also a weight based clustering solution. Once more, the weight parameters are only taken into account for the selection of the clusterhead.

The Connectivity-based Clustering Scheme (CCS) [8] has the purpose of improving the effectiveness, reliability and stability of MANETs. In contrast with

Table 1: Comparison of clustering schemes

	CH	**GW**	**1/n-hop**	**MH**	**Main Objective**
SCA (2007)	Yes	Yes	2-hops max.	No	Large MANETs with high-speed nodes
SMCP (2005)	Yes	Yes	n-hop	No	Stable cluster formation
KhCP (2006)	Yes	No	n-hop	Yes	Limited overhead for dense networks
DWCA (2006)	Yes	Yes	1-hop	No	Stability of the network
EPCA (2010)	Yes	No	n-hop	No	Performance, with trusting node mechanism
CCS (2008)	Yes	No	n-hop	No	Effectiveness, reliability and stability
EEMC (2007)	Yes	No	n-hop	No	Distributed power consumption, limited control message flooding
TEDMC (2008)	Yes	Yes	1-hop	No	Stability, relying on trust values and residual energy of nodes
OCRP (2007)	Yes	Yes	1-hop	No	Merge clustering phase with routing discovery and data transmission
OCR (2007)	Yes	Yes	n-hop	Yes	Light control overhead, establishing the cluster structure and routing paths simultaneously
ODGM_GN (2008)	Yes	No	n-hop	No	Build clusters as foundation for variable types of routing protocols
EWDCA (2010)	Yes	No	n-hop	No	Maintain stable cluster structure with lowest number of clusters
NSLOC (2010)	No	Yes	n-hop	No	Provide stable cluster structure with low control overhead

most schemes, this solution ignores mobility and energy parameters, focusing only in the cluster organization to achieve its objectives. In order to provide effectiveness and low maintenance, it utilizes a technique of maintaining clusterheads separated by a significant hop distance. Therefore, the probability that two clusterheads come into each other's transmission range is reduced, decreasing the number of re-clustering operations. Concerning the reliability objective, an intra-connection degree is used to measure the connection quality between a node and the possible clusters that it can join.

The Energy Efficient Mobility-sensitive Clustering (EEMC) [9] presents a solution for energy balancing. The main objective of this scheme is to extend the lifetime of the network, by distributing the load amongst nodes and also regarding their mobility. The Trust-related and Energy-concerned Distributed MANET Clustering (TEDMC) [10] is also a scheme driven by energy concerns. TEDMC considers that the most important nodes are the clusterheads, and therefore it elects them according to their trust level and residual energy. In order to keep information about the trust level of nodes, this algorithm maintains and periodically exchanges a reputation rank table, which contains a reputation value and the unique identification of the last node to assign the value in question. Furthermore, TEDMC is substantially different from KhCP, as it only allows 1-hop clusters, thus being less suitable for dense networks.

There are also clustering schemes capable of performing route discovery, such as the On-Demand Clustering Routing Protocol (OCRP) and On-Demand Routing-based Clustering (ORC) [11,12]. These schemes are capable of building cluster structures and routing paths on-demand. In these schemes, only the nodes that are necessary to satisfy a routing path are bounded to the cluster structure. The On-Demand Group Mobility-Based Clustering with Guest Node [13] provides a solution with the main purpose of building a cluster structure capable of supporting several types of routing protocols with identical efficiency. Furthermore, it relies in a guest node approach in order to introduce arriving nodes to the network.

The Efficient Weighted Distributed Clustering Algorithm (EWDCA) [14] has the major concern of providing scalability for MANETs, by taking into consideration several weight parameters: connectivity, residual battery power, average mobility and distance between nodes. These parameters are used only to elect the most suitable clusterhead, in order to keep an optimal number of clusters, thus providing as much scalability as possible. A Novel Stable and Low-maintenance Clustering Scheme (NSLOC) [1] is a fully distributed clustering scheme, with the main goal of simultaneously provide a low maintenance overhead and network stability. NSLOC scheme employs a completely distributed approach, not relying on clusterheads, in contrast to most well known clustering schemes.

Table 1 shows the main characteristics of the analyzed clustering schemes. One of the main reasons clusterheads are so utilized is due to the simplicity that they provide to the clustering algorithm. Centralizing the power of management on only one node results in a less complex algorithm thus, becoming easier and faster to implement. Nonetheless, clusterheads carry big disadvantages, as they represent bottlenecks and uneven energy consumption in the network, due to the centralized management decisions.

3 Smart and Balanced Clustering for MANETs

SALSA is a fully distributed clustering scheme designed to operate in MANETs. The main purpose of this scheme is to build stable clusters aiming to significantly reduce the control overhead, thus providing a light hierarchical structure for routing. This proposal is designed to build a cluster topology in a distributed fashion, meaning that each node in the network will have the same role, not relying on centralized points, like clusterheads.

This scheme introduces a new load-balancing algorithm, which acts progressively along time. During execution, SALSA analyzes the current size of clusters and distributes nodes across them, in order to maintain well balanced clusters. Before the maximum capacity of a cluster is reached, it starts to assign nodes to neighbor clusters or, in cases where this operation is not possible, builds a new cluster to receive excess nodes.

With this new scheme, it was intended to reduce, even further, the clustering control overhead. This objective was mainly accomplished by utilizing small and purpose-drive specific messages. As a result, the proposed scheme utilizes five different types of messages, ensuring in most cases, a significant decrease in the amount of transmitted traffic, when compared to NSLOC.

3.1 Node States

In SALSA, nodes can be in one of three distinct states, namely *Unclustered*, *Clustered* and *Clustered-GW*, as shown in Figure 1.

The *Unclustered* state typically represents a temporary role, as the node is waiting to be assigned to a cluster. In this state, when the node discovers neighbors, it waits a predefined period of time in order to calculate the best candidate cluster to join.

Nodes in the *Clustered* state usually represent the majority of nodes on the network, whereas all in-range nodes must belong to its cluster. Thus, the communication with foreign nodes (i.e. nodes assigned to a different cluster) is performed through gateway nodes.

Finally, the *Clustered-GW* state is assigned to nodes that have in-range foreign nodes, i.e. they must have direct connectivity with at least one different cluster. Thus, they are responsible of forwarding inter-cluster maintenance messages and typically are located on the edge of clusters.

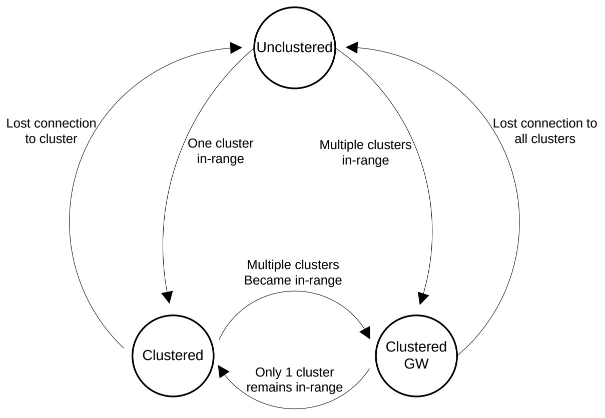

Unclustered
Lost connection to cluster
One cluster in-range
Multiple clusters in-range
Lost connection to all clusters
Multiple clusters Became in-range
Clustered
Clustered GW
Only 1 cluster remains in-range

Fig. 1: Node states

State Transitions. The *Unclustered* state occurs on two different situations:

1. Node isolation - in this case the node does not have any in-range neighbor nodes, therefore cannot create or be assigned to a cluster
2. Cluster transition - the management of clusters occasionally requires nodes to change clusters, due to cluster balancing. In this phase, nodes can be unassigned from a cluster.

Unclustered to Clustered This state occurs when a node becomes aware of an in-range cluster or an unclustered node. In the first situation, the node joins the cluster automatically. However, if the node only detects unclustered nodes, a new cluster is created to adopt the unclustered nodes.

Unclustered to Clustered-GW This transition is similar to the previous, but more than one cluster is discovered. Firstly, the node calculates which is the best, taking several parameters into account: number of in-range nodes for each cluster and the size of clusters. The greater the number of in-range nodes, the stronger connection to the cluster. However, if the size of the cluster is high, possibly close to the maximum allowed, this cluster would be a bad choice. To measure this trade-off, a new metric is utilized (1), namely the best clustering metric (BC).

$$BC = AP + \frac{IRN}{2} \tag{1}$$

The AP value is defined as the number of the available positions in the cluster until it reaches the maximum allowed, i.e. the difference between the maximum allowed number of nodes per cluster and the current number of assigned nodes. The IRN parameter is the number of in-range nodes belonging to the cluster. As a result, the cluster with the higher BC value is automatically chosen by the node.

Clustered to Clustered-GW This transition occurs when a node becomes aware of more clusters, excluding its own.

Clustered-GW to Clustered Whenever a clustered gateway node loses connection with all its foreign clusters, it automatically transits to a normal clustered state.

Clustered/Clustered-GW to Unclustered A node becomes unclustered when willingly disconnects from the network or loses connection with all its neighbor nodes. When this situation occurs, it is necessary to verify the consistency of the cluster, i.e. guarantee that all home nodes can communicate with each other.

3.2 Maintenance Information and Messages

This subsection describes the information that each node maintains and the messages utilized. There are two tables providing insight of the network topology, namely the *NODE_TABLE* and the *CLUSTER_TABLE.NODE_TABLE* keeps all the information about neighbor and home nodes, as described in Table 2.

SALSA relies on multiple, small purpose-driven, messages to manage the cluster structure. All messages contain one common field, *Type_ID*, which uniquely identifies the message type that is being transmitted. Apart from this field, all the messages contain different sets of fields, suitable to their purpose, as follows:

– *Ping* - periodic broadcast message, allowing nodes to discover their neighborhood

Table 2: Node maintenance information

Information	**Description**
Node_ID	Unique identifier of the node
State	Current state of the node (*Unclustered*, *Clustered* or *Clustered-GW*)
C-Degree	Value to determine the connection type towards this node. Value ranges from 0 to 5, whereas 0 represents a non-neighbor (therefore merely a home node), 1 denotes a lost connection towards this node and finally, 2-5 values represent the quality of the connection, being 5 the best possible connection.
Alive	Boolean value, determining whether the node is responding or not

- *Hello* - provide the structure of the cluster to member nodes
- *Lost Hello* - broadcasted when a node loses connection with a neighbor home node, informing member nodes, that do not have direct connection, about a possible disconnected node. This event triggers a process in order to verify if the node is still connected via other nodes, namely alive check process. At the end of this process, if it is verified that the node is in fact disconnected, it is necessary to verify if the cluster is still consistent, which implies the utilization of the following described message (*Alive Hello*)
- *Alive Hello* - upon the trigger of an alive check process, to verify the consistency of the cluster, i.e. guarantee that all nodes inside the cluster are capable of communicating with each other. In most situations the cluster remains consistent; however there are rare cases in which the cluster becomes partitioned in two clusters. In this particular situation, both clusters have the same identifier, thus it becomes imperative to change it.
- *Switch Hello* - used when a cluster identifier becomes inconsistent and it is necessary to change the *Cluster_ID* for their nodes.

3.3 Complexity Analysis

In this section, an analysis of the overhead introduced by SALSA is performed. The scheme operations can be classified as follows:

- Overhead due to *Ping* messages (OH_{Pg})
- Overhead due to Cluster Formation (OH_{CF})
- Overhead due to Cluster Maintenance (OH_{CM})

As previously described, SALSA utilizes a *Ping* message mechanism so nodes are able to discover their neighborhood. Thus, since the broadcast of these messages is constant during the execution of the algorithm, it must be analyzed aside from the remaining operations. The network model of SALSA to the analysis of the clustering overhead relies on the following parameters:

- N = the number of nodes in the entire network
- M = predefined constant, defining the maximum allowed number of nodes per cluster

- t_{ping} = predefined period of time for *Ping* message broadcast
- $t_{formation}$ = predefined period of time for initial cluster formation (join or cluster creation)
- t_{join} = predefined period of time for node join operation
- t_{change} = predefined period of time for node cluster change
- t_{alive} = predefined period of time to determine status of nodes

***Ping* Overhead.** In SALSA, *Ping* messages are broadcasted periodically. This process implies an overhead of $t_{ping}N$ messages per time step. Since t_{ping} is a predefined constant by the algorithm, the overhead of the *Ping* message is $OH_{Pg} = O(N)$ per time step.

Cluster Formation Overhead. In the cold start of SALSA, where all nodes in the network are unclustered, each node waits a predefined period of time, whether to create a new cluster or to join a recently created one. Thus, before a node being assigned to the cluster structure, it must wait at least a $t_{formation}$ period of time. Following this procedure, several *Hello* messages will be broadcasted to it, providing the necessary information about its cluster. The number of *Hello* messages broadcasted is equal to the number of 1-hop neighbors of the node, inside its cluster. Thus, in a worst case scenario, a recently clustered node will receive M *Hello* messages. The *Hello* messages are broadcasted simultaneously, and therefore it takes only 1 time step for this process, which adds up to $t_{formation} + 1$. Analyzing the complexity for the entire network, the overhead is $(t_{formation} + 1M)N$, resulting in $O(MN)$. However, since the formation process only occurs once during the entire execution, and not constantly as for *Ping* messages, the total overhead is $OH_{CF} = O(1)$.

Cluster Maintenance Overhead. The maintenance of clusters is divided in two main routines, namely the joining of a new node and the leaving of a node. This two events are responsible for triggering all the operations to manage the cluster structure.

Joining of New Node When a node joins a cluster two operations may be triggered, namely the auto-balancing of clusters or the creating of a new cluster, due to the imposed maximum nodes per cluster. In most cases, the node simply joins a cluster without requiring these operations, however for the complexity analysis, the worst case scenario must be considered. When a node wishes to join the cluster structure, it waits a predefined period of time t_{join} in order to discover the neighborhood, and to choose the most suitable cluster. Upon choosing its cluster, the node assigns itself to it and receives an *Hello* message from a member node, similarly to the initial phase of cluster formation. Thus, the join operation alone as a complexity of $t_{join}N$ for the entire network, and an overhead of $O(N)$ per time step.

The auto-balancing mechanism may be triggered once a node joins a cluster, which requires a node to be assigned to a different cluster. In this process, the node waits a random amount of time, no longer than a predefined period t_{change}. When this time expires, the node emits an *Hello* message, informing its former members that its no longer assigned to that cluster. This process implies a time complexity of $(t_{change}+1)N$ which results in an overhead of $O(N)$ per time step.

The creation of a new cluster is also an operation that can be triggered by the join cluster operation, when auto-balancing is not possible. This operation, is executed before the new node joins the cluster. Since the operation does not affect the topology of existing clusters, the message complexity does not exist, since the existence of the new cluster is broadcasted using *Ping* messages. In short, it will only cost the period of time t_{join}, resulting in an overhead of $O(1)$ per time step. Summarizing, the overhead of joining of new node is $O(N)+O(N)+O(1)$ which results in $O(N)$.

Leaving of a Node When a clustered node detects that has no longer connection to one of its member neighbors, it broadcasts a *Lost_Hello* message. Upon the reception of this message, each node waits a predefined period of time t_{alive} and broadcasts an alive message. This process, results in a message complexity of $t_{alive}MN$, which implies an overhead of $O(N)$, since M is a constant predefined by the algorithm. After this process, as the cluster may lose its consistency, a *Switch_Hello* message is broadcasted to build two new clusters.In the worst case scenario, M messages are broadcasted, resulting in complexity of $(t_{alive}M + M)N$, with an overhead of $O(N)$.

Total Maintenance Overhead As analyzed above, the overhead of joining of a new node is $O(N)$ and the leaving of a node is also $O(N)$, which results in a maintenance overhead of $OH_{CM} = O(N)$.

Total Clustering Overhead. Summarizing this analysis, the total overhead is denoted by $OH_C = OH_{Pg} + OH_{CF} + OH_{CM}$, which results in $OH_C = O(N) + O(1) + O(N)$. Consequentially, SALSA has a total clustering overhead of $O(N)$ per time step.

4 Simulation Evaluation

To properly examine the effectiveness of SALSA, a simulation evaluation, driven by the main objectives of the scheme, was performed using the OPNET Modeler [15]. Therefore, the main purpose of this simulation evaluation is to assess the stability and low overhead capabilities of SALSA. To accomplish this objective, a set of different simulation environments, featuring the network size and speed of nodes, were defined.

4.1 Environment and Parameters

The performance of clustering schemes is strongly influenced by the scenarios under which they are evaluated. For instance, a better performance is expected for low-density networks (i.e. low quantity of nodes per Km^2) or with nodes moving at low speeds. The scenarios used for SALSA evaluation were selected

Table 3: Simulation parameters

Fixed-value parameters

Simulator	OPNET Modeler 16.0
Field Size (m^2)	5000×5000
Node mobility algorithm	Random Waypoint Model
Pause time (s)	50
Transmission range (m)	150
Bandwidth (Mbps)	11
Simulation time (s)	900

Variable-value parameters

Network size (number of nodes)	200; 400; 600; 800; 1000
Node maximum speed (m/s)	0; 5; 10; 15; 20

in such a way that they represent, as much as possible, realistic scenarios. For this specification, the evaluation parameters can be divided in two groups, the fixed-value and the variable-value parameters, according to whether their value changes for different simulation scenarios (Table 3).

Given the enormous quantity of different possible scenarios that the combination of parameters provides, only the most significant were chosen. In particular, the parameters that most influence the scalability of the network are the network size (number of nodes) and the maximum speed that nodes can achieve.

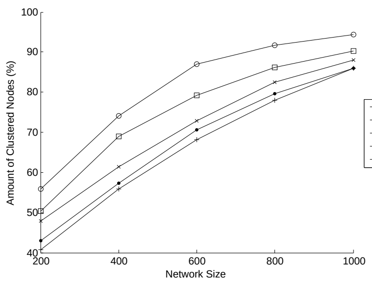
Considering the vast application that clustering can have and that this simulation study aims to evaluate a generic scenario, a specific node mobility pattern, like Group Mobility, Freeway or Manhattan models would not be suitable [16]. Thus, a random model, the Random Waypoint, was preferred. Also, for a simulation of 900 seconds, a 50 second pause time was chosen.

4.2 Results

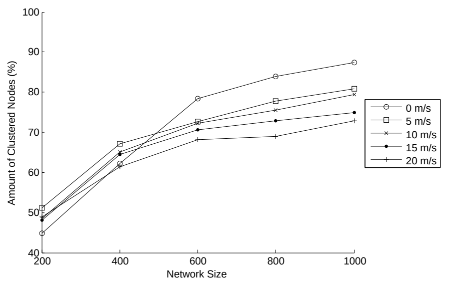
This section presents the obtained results from the SALSA simulation. As previously mentioned, SALSA is a completely new algorithm, based on the NSLOC scheme. Thus, the discussion of the following results will be conducted according to the results obtained in NSLOC.

Number of Clustered Nodes. This metric provides the number of nodes that are associated with the cluster structure.

Figure 2a shows the percentage of clustered nodes for the different network sizes and node speeds in SALSA. The percentage of clustered nodes for large networks is bigger than for smaller networks. Naturally, this occurrence is strongly tied with the density of the network, i.e. the probability of a node being communication in-range with another is greater for networks with more nodes. When



(a) SALSA

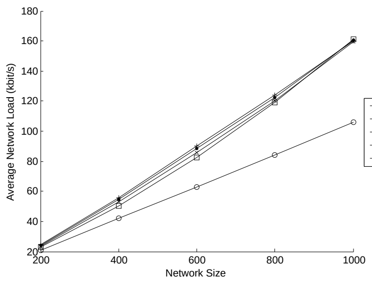


(b) NSLOC

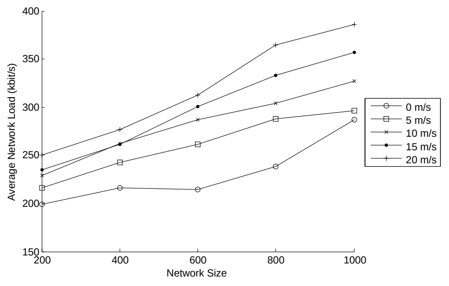
Fig. 2: Amount of clustered nodes (in percentage)

compared to NSLOC scheme (Figure 2b), the new proposal is capable of assign more nodes to the cluster structure, specially in bigger networks. This slight difference is due to the new node state transition specification, as it analyses the most suitable clusters based on the best clustering metric.

Network Load. The network load represents the received and transmitted traffic in the entire cluster structure. This metric translates the overall weight of the network.



(a) SALSA



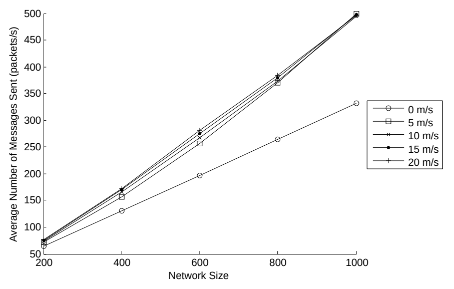
(b) NSLOC

Fig. 3: Average network load

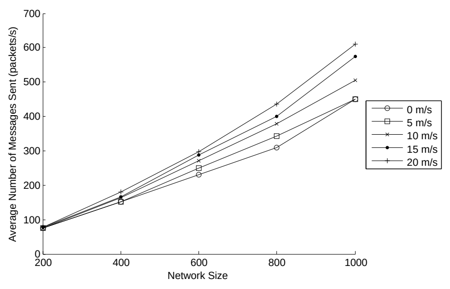
Figure 3a and Figure 3b show the average network load, for different velocities and network sizes, for SALSA and NSLOC, respectively. As shown in the charts, SALSA handles clustering with a significantly lower overhead. In fact, the biggest difference between the two schemes lies exactly on the amount of traffic necessary to handle clustering. As described in Section 3, SALSA only sends the exact required information, utilizing specific message types. As a result, the amount of traffic necessary to manage the cluster structure is quite lower than in NSLOC.

Another important aspect to retain about these results is the consistency of the amount of traffic. In NSLOC, the amount of traffic increases significantly as the maximum speed of nodes increase. SALSA, on the other hand, utilizes almost the same amount of traffic for different speeds, excluding the case when nodes are static. Furthermore, for a network size of 1000 nodes, the average network load for the different speeds seems to converge, utilizing almost the same amount of traffic, meaning that SALSA is sustainable.

Number of Messages. The number of messages required by the clustering scheme to operate, increases with the size of the network. Figure 4a shows the average number of control messages sent by SALSA, for the different network sizes and node speeds. As expected, the shape of this chart is quite similar with the network load metric, since the number of messages sent is strongly tied with the overall network load. Figure 4b shows evaluation results of NSLOC. The



(a) SALSA



(b) NSLOC

Fig. 4: Average number of messages sent

interesting fact tough, is that the average number of messages sent by SALSA is not much lower than the ones sent by NSLOC, in spite of the significant difference that obtained in the average network load results. Once more, this fact is due to the smaller, and more specific, messages utilized by the new scheme.

Other Evaluation Metrics. In addition to the previous evaluations, there are important metrics that must not be ignored, such as the balancing of clusters and their stability. However, due to the lack of space the complete results are not presented here. Concerning the first, results are quite acceptable, having for instance, an average of 23 clusters for the 1000 sized network with static nodes (0 m/s). In this scenario, a average total of 944 nodes were clustered, resulting in an average of around 41 nodes per cluster. Moreover, in the 1000 sized network with a node maximum speed of 20 m/s, an average of 32 clusters were created, whereas around 858 nodes were clustered. This represents an average of 27 nodes per cluster, which can be acceptable due to the high velocity. To be noted that in all the evaluations, SALSA was configured with a maximum number of 50 allowed nodes per cluster.

The stability of clusters can be measured according to the amount of time that nodes belong to a cluster, without suffering re-clustering operations. For this analysis, a cluster stability metric is utilized, which defines for a number of initial parameters, a stability time (ST), from which nodes are considered to be stable (2).

$$ST = k \times \frac{r \times p}{v \times d} \tag{2}$$

where r is the transmission range of nodes, p is the pause time, v the average of node speed (mean value of minimum and maximum speed), d the density of nodes (number of nodes per Km^2) and finally, k represents an arbitrary constant, equal in all simulation executions.

For the network size of 1000 nodes and no mobility (0 m/s), results show an average of 894 stable nodes. This means that this quantity of nodes were at least ST time without changing cluster. Also, for a network size of 1000 nodes with a maximum speed of 20 m/s, a total of 804 nodes were considered stable.

5 Conclusion

In this paper, the SALSA scheme was proposed, aiming to improve the performance of large MANETs. The proposed clustering scheme employs a full distributed approach, in contrast to most well known clustering schemes. Additionally, a new cluster balanced mechanism and a best clustering metric are employed by SALSA, providing a reduced control overhead.

Evaluation results shown that SALSA is capable of outperforming NSLOC in the majority of the scenarios and completely from network sizes of 400 nodes. The most noticeable difference is in the traffic overhead, which is significantly lower (reduced around 45% for all evaluated network sizes) than NSLOC scheme.

Acknowledgments

This work was supported by FCT project MORFEU (PTDC / EEA-CRO / 108348 / 2008). The authors would like to thank the OPNET University Program for the licenses provided for the OPNET ModelerWireless Suite R.

References

1. L. Conceição, D. Palma, and M. Curado, "A novel stable and low-maintenance clustering scheme," in *Proceedings of the 2010 ACM Symposium on Applied Computing*, ser. SAC '10. New York, NY, USA: ACM, 2010, pp. 699–705. [Online]. Available: http://doi.acm.org/10.1145/1774088.1774232

2. J. Yu and P. Chong, “A survey of clustering schemes for mobile ad hoc networks,” *Communications Surveys & Tutorials, IEEE*, vol. 7, no. 1, pp. 32–48, Qtr. 2005.
3. F. Tolba, D. Magoni, and P. Lorenz, “A stable clustering algorithm for highly mobile ad hoc networks,” *Systems and Networks Communications, 2007. ICSNC 2007. Second International Conference on*, pp. 11–11, Aug. 2007.
4. J. Tenhunen, V. Typpo, and M. Jurvansuu, “Stability-based multi-hop clustering protocol,” *Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on*, vol. 2, pp. 958–962 Vol. 2, Sept. 2005.
5. G. Angione, P. Bellavista, A. Corradi, and E. Magistretti, “A k-hop clustering protocol for dense mobile ad-hoc networks,” *Distributed Computing Systems Workshops, International Conference on*, vol. 0, p. 10, 2006.
6. W. Choi and M. Woo, “A distributed weighted clustering algorithm for mobile ad hoc networks,” *Telecommunications, 2006. AICT-ICIW '06. International Conference on Internet and Web Applications and Services/Advanced International Conference on*, pp. 73–73, Feb. 2006.
7. R. Zoican, “An enhanced performance clustering algorithm for manet,” in *MELECON 2010 - 2010 15th IEEE Mediterranean Electrotechnical Conference*, 2010, pp. 1269 –1272.
8. K. T. Mai and H. Choo, “Connectivity-based clustering scheme for mobile ad hoc networks,” *Research, Innovation and Vision for the Future, 2008. RIVF 2008. IEEE International Conference on*, pp. 191–197, July 2008.
9. D. Wei, H. Chan, E. Chuwa, and B. Majugo, “Mobility-sensitive clustering algorithm to balance power consumption for mobile ad hoc networks,” *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, pp. 1645–1648, Sept. 2007.
10. Z. Qiang, Z. Ying, and G. Zheng-hu, “A trust-related and energy-concerned distributed manet clustering design,” in *Intelligent System and Knowledge Engineering, 2008. ISKE 2008. 3rd International Conference on*, vol. 1, 2008, pp. 146 –151.
11. C. Huang, Y. Zhang, X. Jia, W. Shi, Y. Cheng, and H. Zhou, “An on-demand clustering mechanism for hierarchical routing protocol in ad hoc networks,” *Wireless Communications, Networking and Mobile Computing, 2006. WiCOM 2006.International Conference on*, pp. 1–6, Sept. 2006.
12. C.-H. Hsu and K.-T. Feng, “On-demand routing-based clustering protocol for mobile ad hoc networks,” *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, pp. 1–5, Sept. 2007.
13. A. Dana, A. Yadegari, A. Salahi, S. Faramehr, and H. Khosravi, “A new scheme for on-demand group mobility clustering in mobile ad hoc networks,” *Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on*, vol. 2, pp. 1370–1375, Feb. 2008.
14. A. Hussein, S. Yousef, S. Al-Khayatt, and O. Arabeyyat, “An efficient weighted distributed clustering algorithm for mobile ad hoc networks,” in *Computer Engineering and Systems (ICCES), 2010 International Conference on*, 30 2010.
15. OPNET, “Opnet simulator,” 1986, http://www.opnet.com/. [Online]. Available: http://www.opnet.com/
16. B. Divecha, A. Abraham, C. Grosan, and S. Sanyal, “Impact of node mobility on manet routing protocols models,” 2007.