DASH, Deferred Aggregated routing for Scalable ad-Hoc networks

David Palma
Department of Informatics Engineering
Centre for Informatics and Systems
University of Coimbra (CISUC)
Coimbra, Portugal
palma@dei.uc.pt

Marilia Curado
Department of Informatics Engineering
Centre for Informatics and Systems
University of Coimbra (CISUC)
Coimbra, Portugal
marilia@dei.uc.pt

Abstract—**A new Deferred Routing approach for future wireless networks, focused on large scale mobile and dynamic networks, is described. A hierarchy is defined in order to maintain different routing granularity levels to each network cluster. This allows an efficient management and dissemination of the necessary routing information, as well as resilience in mobility scenarios, for the usage of Deferred Routing with context aggregation. A complexity analysis shows an overall more scalable protocol when compared to other well known routing protocols and simulation results confirm this performance increase. The impact is very significant for larger scenarios, and represents an improvement for future networks, such as ad-hoc networks.**

I. INTRODUCTION

During the past years, technological advances promoted a massive dissemination of new wireless capable devices with greater processing power, higher memory and increased autonomy. All these aspects cleared the way for the creation of new applications suitable for every day use - file transferring, sensoring, messaging, multimedia, and gaming, among others. In a near future each person is expected to be surrounded by hundreds or even thousands of these devices [1], motivating the development of networks capable of connecting them whilst supporting the application's requirements. There are several occasions in which these networks may be appropriate, such as conferences, music concerts or football games, allowing people to interact fully using their devices, thus supporting the concept of social context-awareness [2]. However, managing such a network where there is no available infra-structure to support it, or where the existing networks are not capable of dealing with a large number of users, is still a challenge.

Ad-hoc networks have increasingly shown their importance in the dissemination of dynamic wireless communication systems, standing out for being available anywhere, without requiring any existing infra-structure, and for typically being self-organised, self-administrated and self-maintained. Said characteristics make these networks perfect candidates for supporting the above mentioned scenarios, dealing with the dynamic network connections created by people movement and other interactions.

Despite existing works on this topic - such as the Optimized Link-state Routing Protocol (OLSR) [3] which provides an optimisation for the typical link-state routing, and the Dynamic MANET On-demand Routing Protocol (DYMO) [4] which, on its hand, offers an on-demand routing approach - maintaining routing performance for large scale networks is still an issue. Taking this problem into account, several works propose different schemes involving techniques such as dynamic addressing which keeps network nodes organized in a well defined topology (Dynamic Address Routing for Scalable Ad-hoc and Mesh Networks (DART) [5]), geographic partitioning to easily create stable clusters [6] and typical clustering solutions such as [7].

Even though the mentioned works represent an important contribution for scalable routing, these techniques depend on complex approaches for ensuring correct address assignments and stable clusters or on specific hardware (such as Global Positioning Systems) which may reduce network lifetime or may not be feasible in certain scenarios. Considering the creation of an Ad-hoc network, due to the interaction of people, efficient routing could be easily achieved by using the information obtained from its context. In [8], the sense of community is used to create a logical topology which enhances routing. However, it is still dependent on a typical on-demand routing scheme which is not suitable for scalability purposes.

In section II, a routing proposal which is able to handle context for scalability purposes is specified. A new concept named Deferred Routing is presented, where clusters are aggregated and mapped into a virtual hierarchical tree with different levels of granularity. This will allow routing decisions to be postponed between clusters with different granularity levels until the final destination is reached. Such an architecture, with virtually aggregated clusters, make routing tables resilent to mobility and other disruptive phenomena, while reducing their size, thus maintaining routing at a scalable level. A performance evaluation and complexity analysis is presented in section III, followed by the conclusions in section IV.

II. DASH, PROPOSAL SPECIFICATION

The main purpose of routing is to correctly choose the next hop for a packet to be forwarded to, such that it is able to reach

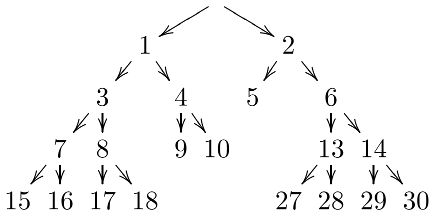

Fig. 1. Example of the Hierarchical Tree Representation

its destination. Typical solutions are focused on determining the optimal route to be followed by a packet, however the cost of maintaining an updated routing table of the entire network, as well as of retrieving an on-demand path, may be too high - particularly when large scale and very dynamic networks are concerned.

Considering routing scalability, a common solution is to assign nodes to clusters, restricting routing operations within each cluster and having special routing procedures for inter-cluster communication. However, this approach typically requires that clusters constantly exchange topology information between themselves, which is still costly. Moreover, most cluster based routing approaches completely disregard clusters' context, becoming highly vulnerable to mobility phenomena and loss of connections, disrupting the routing and cluster performance, due to frequent cluster reassignments and consequent new route calculations.

Other approaches simplify the message update exchange by using a hierarchy maintained by special nodes which are able to connect directly with other clusters, performing as dedicated routers, similar to the typical Internet architecture [9], at the cost of using specific hardware which may not be available in normal situations.

By using the clusters' context or similar information that can be retrieved in the nodes, these can be efficiently aggregated in a way that mobility and other interactions will not affect the created structure. This structure could be organised in a binary tree fashion, in which each leaf represents a cluster and each root corresponds to two or more aggregated clusters (figure 1), allowing different views of the network's structure depending on the cluster's identification (ID). The creation and management of this tree will be detailed in the following subsection.

Furthermore, an approach capable of handling such a structure for scalable routing purposes is presented in subsection II-B, using a Deferred Routing scheme where the available information about the desired destination becomes more accurate when reaching clusters in the same hierarchy branch, and thus with similar context.

A. Hierarchical Tree Management

The creation of a Hierarchical Tree and the assignment of each leaf to a cluster is a necessary process that allows the proposed routing scheme, the Deferred Aggregated routing for Scalable ad-Hoc networks DASH, to be efficient. This process could take into account any type of information that can be used by any clustering scheme. However, the possible clustering schemes that can be used for this task are out of the scope of the presented work. For illustrative purposes, the "proximity" of nodes context-wise will be used for the creation of the Hierarchical Tree, even though other information could be used, since this proposal is generic enough to support any clustering scheme. By keeping nodes grouped according to the available information, the clusters are created. Each cluster will then represent a context and should be "close" (i.e. hierarchically close) to other clusters with similar properties. The definition of the clusters proximity will depend on the used information and will be represented by the hierarchical tree siblings.

A Hierarchical Tree is depicted in figure 1, where each number is used as a representation of a cluster's context, the Context ID, hence forth known as CID. The leaves with the CIDs 5, 9, 10, 15-18 and 27-30, represent and are assigned to real clusters, while the remaining CIDs represent views of aggregated clusters, mapping these clusters into a broader one that includes them. As an example of this, CID 7 describes the cluster of both CIDs 15 and 16, while CID 1 corresponds to a generic view of the context of all the cluster siblings. In a real scenario, CID 1 could represent the cluster of a Computer Science Department where, 3 and 4 are used to separate people attending lectures from people studying, which then have more specific contexts that could indicate a specific room or area, and give insights about future interactions.

Knowing that clusters are formed by nodes in similar contexts, good connectivity is expected to exist inside each cluster, and nodes will likely remain in "nearby clusters", meaning that if any transition between clusters should occur, a node will still be close to its original cluster. Taking into account the above description of the Context Tree, each pair of siblings is expected to have physical connectivity between each other.

In order to support a number of c clusters as leaves, the tree must have at least l levels. Knowing that the clusters' context is organized in a binary tree, this value is defined by $l = \lceil \log_2 c \rceil$, keeping a CID for each branch of the tree. These IDs can be determined and assigned by any process, which should be responsible for managing cluster information, determining the cluster of each group of nodes, using $CID \times 2 + 1$ and $CID \times 2 + 2$ for assigning the left and right ID in each branch division, respectively, and using equation $\left\lceil \frac{CID-2}{2} \right\rceil$ for the joining ID.

B. DASH Routing Scheme

The DASH routing scheme uses a link-state paradigm for the maintenance of routes inside each cluster. However, in order to perform in an entirely scalable fashion, routing information regarding other clusters is restrained according to context proximity, being more precise for closer contexts than for distant ones. This is achieved by keeping only aggregated views of the network, for instance, referring to the presented hierarchical tree in figure 1, a node in the cluster identified as CID 15, will only keep connectivity information to reach its “brother” (i.e. the cluster with CID 16). For all the other clusters, it will simply maintain aggregated views according to their hierarchical distance, such that 17 and 18 are only seen as a cluster with the CID 8, while 9 and 10 are aggregated as having CID 4. The remaining clusters, being even more distant context-wise, will be grouped in a broader view with CID 2.

Besides keeping less information about the network topology, the usage of aggregated views allows a powerful routing scheme resilient to node mobility. In fact, the impact of a node changing its cluster is only perceived in a small area, assuming that such a transition rarely occurs between clusters which are very distant in the hierarchy. This approach avoids the typical disruptive table update per node in the network whenever a node changes its cluster. In fact, assuming that a node’s cluster may change from CID 29 to 30, the only two clusters affected will be the ones mapped by CID 29 and 30, since all the remaining clusters will have an entry for that node, using their own network’s perspective. This perspective will be represented into aggregated views which will have CIDs 14, 6 or 2, depending on cluster “proximity”.

Having aggregated views of the network represents a key point on the efficiency of this approach, however it raises a new challenge for inter-cluster routing. While inside the cluster a typical link-state routing protocol keeps an updated routing table, paths throughout neighbour clusters are not known. Even though typical hierarchical schemes rely on special nodes in each cluster responsible for this task, by maintaining routes between all the existing clusters, this may not be feasible in highly dynamic networks. In order to tackle this problem, the previously presented concept, Deferred Routing, is introduced by the DASH protocol.

The Deferred Routing approach consists in using the presented Hierarchical Tree, for forwarding packets between different clusters until the desired cluster destination is reached, and where, being in the desired cluster, intra-cluster routing will take care of determining the best path to the destination node. The first step of this process consists in determining the destinations’ node CID, by either looking it up in a table which can either be maintained by the routing protocol or by another application such as a cluster management service, responsible for mapping each node’s unique identifier to a specific Cluster/Context ID. Being aware that the retrieved CID may consist of an aggregated view, the nodes responsible for forwarding a data packet choose the most suitable node inside their own cluster, capable of reaching that CID, unaware of the remaining path to be taken by that packet.

The concept of Deferred Routing may appear somewhat similar to the Fisheye Routing approaches such as [10], in the sense that routing information has different levels of accuracy. However, in this work the routing accuracy or frequency is not on a node basis and does not depend on the hop-distance to another node, it depends on the hierarchical “distance” of a cluster to another, which is related with context aware parameters.

The most important aspects required to ensure performance and scalability have already been mentioned, however, another relevant aspect is the specification of the exchanged routing messages. This proposal assumes that the used routing protocol can be modified in order to be cluster aware, exchanging routing messages within its cluster. In addition to the protocol’s own data, these messages should transmit information about the cluster’s connectivity to other clusters. This cluster connectivity data will be generated by nodes capable of communicating with other clusters (hereby known as gateway nodes), which by overhearing other cluster’s information insert in their own routing information an entry with themselves as gateways to another cluster, and the number of cluster hops necessary to reach it. This allows each cluster to maintain a cluster connectivity table to be used for packet forwarding.

Whenever a node receives routing information about any cluster to which it does not belong to, it will retain all the relevant data about the network according to its own position in the network hierarchy. This means that the node will, for instance, map the received CID information to its own view of such ID. Another important aspect is the process of mapping a node to its Context ID, which allows every node to know where to forward a packet. For the sake of simplicity, it will be assumed that the routing protocol is responsible for this task. Since a node’s cluster is expected to be fairly stable, not changing when link connectivity is lost or when another node joins/leaves the cluster, this process should be reactive instead of proactive. This will allow the consumption of few resources and will ensure that only hierarchically “close” clusters are forwarded a new tuple Identifier-Context ID whenever there is an update, as it is most likely that the Identifier is aggregated in higher views for more “distant” clusters, not changing anything at all.

III. PERFORMANCE ANALYSIS

In order to evaluate DASH, an overhead and complexity analysis has been performed. Simulation results are also presented, comparing the performance of the well known OLSR protocol with this proposal. However, since any routing protocol may be used with this new routing scheme, the theoretic analysis will only take into account the overhead introduced by information specifically required by the proposal, similarly to [11].

A. Routing Overhead

Considering a network of n nodes, and assuming a cluster organisation where there is a maximum number of k_{max} nodes per cluster, the routing overhead incurred by each node is represented by equations 1 and 2. The former equation represents the overhead required for each pair Identifier-Context ID mapping ($IC_{mapping}$). This tuple exists for each node known in the network, and must be shared among all the existing nodes, in such a way that only the Identifiers from a node’s cluster will not be sent by that node, thus $n - k_{max}$ entries. Even though this is being considered as routing overhead, this task could be performed by another process.

The latter equation represents Cluster Connectivity (CC) known by each node, assuming that there is no more than GW active Gateways for each cluster to be connected to. This number corresponds to the hierarchy level of the cluster's Context ID, l, since only aggregated views are kept for the routing procedure. It is important to note that the connectivity information is sent periodically (with an interval of $interval$), throughout time ($time$), contrasting with the Identifier-Context ID tuple information which is only re-sent whenever a node changes its cluster, interfering with the current view of a neighbour cluster. In that case only the new Identifier-Context ID pair needs to be propagated, avoiding unnecessary overhead of resending the entire table.

$$IC_{mapping} = (n - k_{max}) \times Identifier_{size} \times CID_{size} \quad (1)$$

$$\begin{aligned} CC = GW \times l \times Identifier_{size} \times \\ Hops\_Count_{size} \times \tfrac{time}{interval} \end{aligned} \quad (2)$$

Despite having to include this additional information in the routing messages, the suggested approach still needs less resources for routing purposes than other protocols. In fact, common link-state protocols would include in each of their routing messages, exchanged by each node, the entire topology information, thus having severe scalability problems. Even if a typical cluster based solution is considered, the information exchanged by each cluster-head will always have to include topology information about all the existing clusters. Therefore, the only improvement from these solutions is that such information is limited to special nodes, which will have an increased overhead. Overall the proposed routing approach is much lighter than existing solutions.

B. Storage and Communication Complexity

Table I presents communication and storage complexity for some relevant protocols for scalability in MANETs, assuming the worst case scenario, where m represents the number of zones defined in Zone-Based Hierarchical Link State routing protocol (ZHLS) [6], and n the total number of nodes.

For DASH, since each node will know where, in the hierarchy, another node is, the storage complexity will be, in the worst case scenario, $O(n)$. However, it is important to notice that the usage of aggregated views could contribute to a significant reduction of the storage complexity by grouping nodes' Identifiers in each view, as there is always a small number (l) of existing views. Assuming that every Identifier associated to a view can be grouped, a storage complexity of $O(l)$ would be achieved.

Considering that each cluster has k_{max} nodes, the number of clusters c, in a network of n nodes, is represented by equation 3.

$$c = \left\lceil \frac{n}{k_{max}} \right\rceil \quad (3)$$

TABLE I
COMPLEXITY COMPARISON

Protocols	Communication	Storage
Link-state	$O(n^2)$	$O(n)$
DASH	$O(k_{max}^2)$	$O(n)$
ZHLS	$O(\frac{n}{m}^2 + nm)$	$O(\frac{n}{m})$
DART	$O(\log_2 n)$	$O(n)$

Regarding the communication complexity of DASH, and knowing the complexity of a typical link-state protocol, a constant value is expected to be achieved (as shown in equation 4), since the link-state protocol is restricted inside each cluster. This result can still be further improved by using a technique such as the Multipoint Relay (MPRs) nodes in OLSR.

$$O\left(\frac{n}{c}^2\right) \Leftrightarrow O\left(\frac{n}{\frac{n}{k_{max}}}^2\right) \Leftrightarrow O(k_{max}^2) \quad (4)$$

C. Simulation Analysis

In order to evaluate DASH's performance, several simulations were conducted using the OPNET Modeler Wireless Suite® [12]. Five different scenarios were simulated, differing each one on the percentage of mobile nodes (0%, 25%, 50%, 75% and 100%). Each scenario consisted of 196 nodes deployed in a 7 by 7 node grid, at a distance of 70 meters from each other (vertically and horizontally), forming a cluster. All the scenarios were run 30 times with different seed values, for a total time of 900 seconds. The nodes followed the default random waypoint model, which is usually considered the standard mobility model for MANETs, with a pause time uniformly distributed between 60 and 120 seconds, and a maximum speed of $4km/h$, corresponding to typical pedestrian walk [13]. The physical layer of the wireless nodes follows the IEEE 802.11g ($54Mbit/s$) and a theoretical maximum range of $100m$ [14].

Node movement starts after the first 10 seconds of simulation and continues until the end. Regarding the existing traffic flows, there are 28 nodes which generate constant size packets of $4KB$ at a rate of 2 to 6 packets per second, representing instant messaging packets or file transfers [15]. All the destinations are randomly chosen and, since packet transmission only starts after 100 seconds of simulation, all the mobile nodes are already randomly distributed through the scenario. Each cluster represents a context, and nodes' mobility is restricted within it. In order to provide a fair comparison, the nodes in the OLSR scenarios will also have their mobility equally limited.

1) Simulation Results: All the presented graphs show a 95% confidence interval for the results. This was obtained from the central limit theorem which states that, regardless of a random variable's actual distribution, as the number of samples (i.e. runs) grows large, the random variable has a distribution that approaches that of a normal random variable with mean m, the same mean as the random variable itself.

a) Average Routing Traffic Sent: When considering routing scalability, the amount of routing traffic generated by

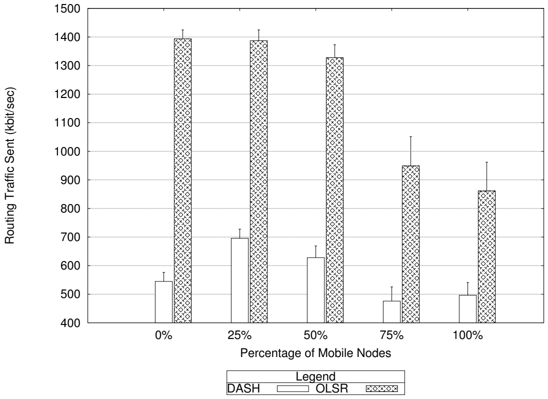

Fig. 2. Average Routing Traffic Sent

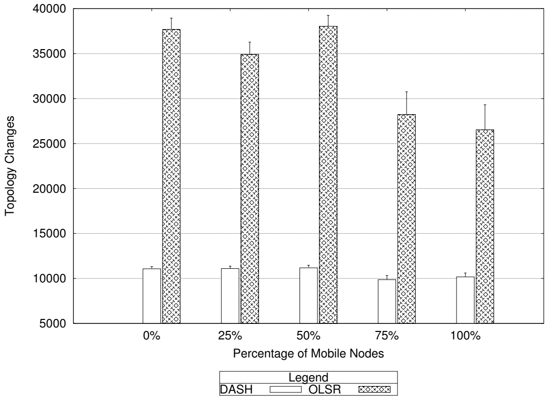

Fig. 3. Average Number of Topology Changes

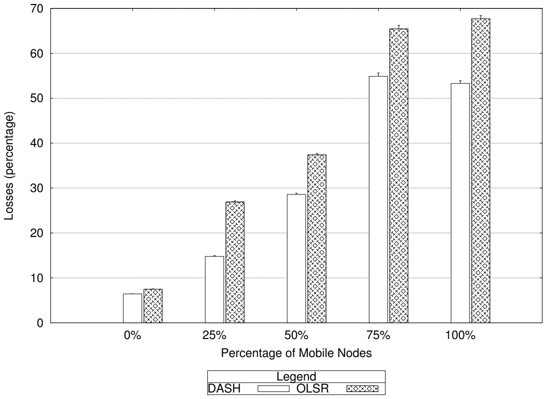

Fig. 4. Average Losses

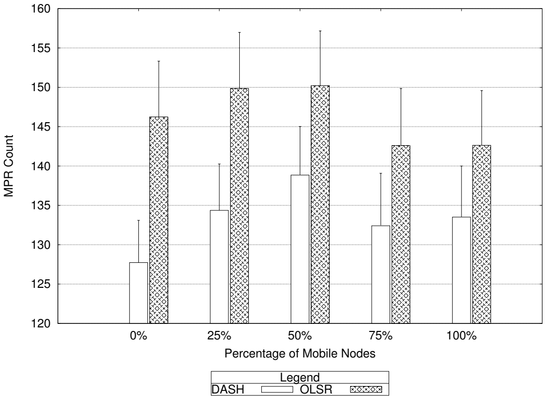

Fig. 5. Average Number of MPR Nodes

a protocol reflects its capacity to deal with different sized networks, in a small or large scale perspective. Figure 2 presents the volume of routing information generated by the analysed protocols, showing that the OLSR protocol creates up to 60% more Routing Traffic in a static network, needing much more resources in all simulated scenarios.

b) Average Number of Topology Changes: A routing protocol's stability depends largely on how the networks' topology is handled and on how many changes occur during time. These changes require that new routing messages are exchanged and result on routing table recalculations, which may create additional overheads and re-routing processes. The registered number of topology changes in the performed simulations is depicted in figure 3, showing that DASH is significantly more stable than the OLSR protocol, keeping the number of topology changes very low and almost the same for different mobility scenarios. On the other hand, the OLSR protocol reveals that has issues in keeping its topology stable, being subject to many topology changes, even in a static scenario.

c) Average Losses: The percentage of lost traffic, for all the existing traffic flows, is depicted in figure 4. The performance of both routing approaches is noticeably better in scenarios with reduced mobility - such that less than 10% of traffic is lost when there is no mobility and more than 50% when all nodes are mobile. Even though the proposed routing approach has only a slightly better performance in the static scenario than OLSR, this performance increases for scenarios with mobility, being close to 50% received traffic in a fully mobile scenario against about 30% received traffic where OLSR is used.

d) Average MPR Count: One important characteristic of the OLSR protocol is the usage of Multi-Point Relay (MPR) nodes as a mean to enhance the expensive topology management procedure of a link-state routing protocol. Typically, a lower number of MPRs represents a lower overhead, as it allows the maintenance of topology awareness with less sent and forwarded Topology Control (TC) messages. Since that in DASH there is no need for MPR nodes between different clusters and that the network is more stable, the number of MPR nodes is smaller when compared with OLSR (figure 5).

e) Average Delay: Shown in figure 6, the results obtained for traffic delay indicate that the DASH routing approach has a very similar performance when compared with OLSR. The average delay decreases as the percentage of mobile nodes rises. This is due to the amount of traffic received in higher mobility scenarios - in which distant path flows, that usually

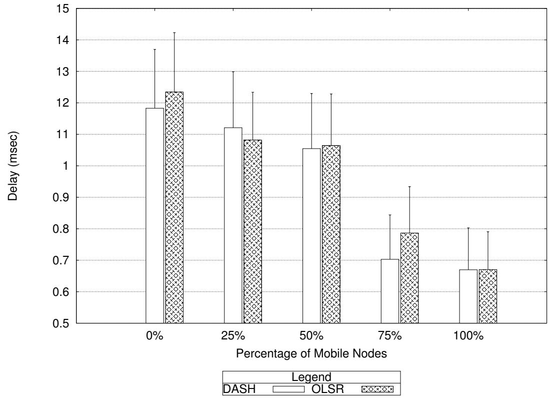

Fig. 6. Average End-to-End Delay

have a higher delay, do not reach their destination, as depicted by the higher loss rate in these scenarios.

IV. CONCLUSION

Motivated by the increasing number of devices capable of creating a network, a new highly scalable routing approach, DASH, has been presented. This new scheme proposes a different perspective on how to maintain routing information, by creating a well defined hierarchy tree, which is able to take into account context or other information. Such strategy allows an efficient exchange of routing messages without disruption when nodes leave or enter the network, or even when nodes have a high mobility profile.

The great performance results obtained by DASH are mainly achieved through the maintenance of aggregated routing information, which depends on a node's view of the network, according to its hierarchical position. This efficient management of routing information ensures that node changes only have a local disruptive impact and do not propagate unnecessarily through the entire network.

In order to maintain intra-cluster routing, any routing protocol can be used, it is only required that a small amount of information is added to the used routing messages. This additional information represents a small overhead when compared to other cluster-based solutions, which typically maintain inter-cluster topology information. In fact, in this proposal, inter-cluster routing is achieved by using an innovative Deferred Routing technique, which even though may lead to sub-optimal paths regarding the number of hops, it still reveals better performance than the OLSR protocol.

Future work shall involve the comparison of this proposal with other hierarchical and cluster-based protocols, analysing both routing overhead and traffic performance. Further optimisations could be achieved if a Context ID-Identifier aggregation scheme was used, avoiding one entry per node, thus representing a significant reduction in storage complexity, since several Identifiers are constantly mapped to the same view.

ACKNOWLEDGEMENT

This work was supported by the Portuguese National Foundation for Science and Technology (FCT) through a PhD Scholarship (SFRH / BD / 43790 / 2008) and by the National Project MORFEU (PTDC / EEA-CRO / 108348 / 2008). The authors would like to thank the OPNET University Program for the licenses provided for the OPNET Modeler Wireless Suite®.

REFERENCES

[1] A. Cimmino and P. Donadio, "Overall requirements for global information multimedia communication village 10th strategic workshop," *Wireless Personal Communications*, vol. 49, no. 3, pp. 311–319, May 2009. [Online]. Available: http://dx.doi.org/10.1007/s11277-009-9686-3

[2] H. Rahnama, A. Sadeghian, and A. Madni, "Social context awareness in ad hoc system of systems," in *2007 IEEE International Conference on System of Systems Engineering*, San Antonio, TX, USA, 2007, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/Xplore/login.jsp?url=http.ieee.orgecision=-203

[3] T. Clausen and P. Jacquet, "Optimized link state routing protocol (olsr)," Published Online, Internet Engineering Task Force, RFC 3626, October 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3626.txt

[4] Chakeres and C. Perkins, "Dynamic manet on-demand (dymo) routing," Published Online, Internet Engineering Task Force, Work in Progress, Internet-Draft, March 2009. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-manet-dymo-17.txt

[5] J. Eriksson, M. Faloutsos, and S. V. Krishnamurthy, "Dart: Dynamic address routing for scalable ad hoc and mesh networks," *Networking, IEEE/ACM Transactions on*, vol. 15, pp. 119–132, 2007.

[6] T. Hamma, T. Katoh, B. B. Bista, and T. Takata, "An efficient zhls routing protocol for mobile ad hoc networks," in *DEXA '06: Proceedings of the 17th International Conference on Database and Expert Systems Applications*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 66–70.

[7] L. Canourgues, J. Lephay, L. Soyer, and A. Beylot, "A scalable adaptation of the OLSR protocol for large clustered mobile ad hoc networks," *Advances in Ad Hoc Networking*, 2008.

[8] H. Kim and M. Yoo, "A scalable ad hoc routing protocol based on logical topology for ubiquitous community network," in *Advanced Communication Technology, The 9th International Conference on*, vol. 2, 2007, pp. 1306–1310.

[9] L. Villasenor-Gonzalez, Y. Ge, and L. Lament, "HOLSR: a hierarchical proactive routing mechanism for mobile ad hoc networks," *Communications Magazine, IEEE*, vol. 43, no. 7, pp. 118–125, 2005.

[10] C.-C. Yang and L.-P. Tseng, "Fisheye zone routing protocol: A multi-level zone routing protocol for mobile ad hoc networks," *Computer Communications*, vol. 30, no. 2, pp. 261 – 268, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/B6TYP-4KW5T63-1/2/f56b93284a1d863aa376df61ac023b0c

[11] D. Palma and M. Curado, "Nodrop, nature optimized deferred routing protocol," in *INFOCOM Workshops 2009, IEEE*, 19-25 2009, pp. 1 –2.

[12] "Opnet simulator," http://www.opnet.com/. [Online]. Available: http://www.opnet.com/

[13] 3GPP, "Spatial channel model for multiple input multiple output (mimo) simulations. technical specification group radio access network." 3rd Generation Partnership Project, Tech. Rep. 7, 2008.

[14] S. Ortiz, "IEEE 802.11n: The road ahead," *IEEE Computer*, vol. 42, no. 7, pp. 13–15, 2009.

[15] Z. Xiao, L. Guo, and J. Tracey, "Understanding instant messaging traffic characteristics," in *Distributed Computing Systems, 2007. ICDCS '07. 27th International Conference on*, 2007, p. 51.