k-hop Backbone Formation in Ad Hoc Networks

Paolo Bellavista and Eugenio Magistretti
Dipartimento di Elettronica, Informatica e Sistemistica
University of Bologna
Bologna, Italy 40136
{pbellavista, emagistretti}@deis.unibo.it

Abstract*—Several recent research activities have started to recognize the relevant role of k-hop clustering in Mobile Ad hoc NETworks (MANET) to effectively support many relevant tasks, e.g., packet routing and information dissemination at the network and application layer, respectively. k-hop clustering determination and maintenance is especially crucial to achieve good scalability in dense MANET scenarios, i.e., geographical areas with relatively high and almost constant density of mobile devices communicating in ad-hoc mode (such as in airport terminals, shopping malls, and university campuses), which are becoming of growing industrial relevance. The paper specifically addresses a primary aspect not yet widely investigated in the literature about k-hop clustering: how to exploit the k-hop clustering process also to effectively build an optimal backbone connecting all clusterheads identified by the process. We propose an original k-hop backbone formation protocol that, under the dense MANET assumption, outperforms other solutions in the literature especially in terms of imposed overhead, by exploiting highly localized intra-cluster interactions and by avoiding any kind of multi-hop broadcasts.

I. Introduction

Envisioned by futurists a few years ago, today the deployment scenario of pervasive and ubiquitous mobile computing devices, present anywhere and anytime, is almost reality. Therefore, there is an emerging and growing need for novel solutions to support highly decentralized execution environments where there are no constraints on terminal mobility and distributed applications are the result of impromptu collaborations among a high number of wireless peers. In that context, several research efforts have addressed the goal of facilitating the development and supporting the runtime execution of services over Mobile Ad-hoc NETworks (MANET) [1]. In particular, a specific kind of MANET scenario is becoming of increasing industrial and commercial relevance, called dense MANET in the following [2]. On the one hand, a dense MANET includes a large number of wireless devices located in a relatively small area at the same time, e.g., as it will probably happen in the near future in shopping malls, airport waiting rooms, and university campuses. On the other hand, it has a node density (the average number of wireless nodes at single-hop distance from any dense MANET participant) that is both spatially uniform and almost invariant during long time intervals. The dense MANET assumption has already demonstrated to simplify the design of protocols and application-level supports in these challenging environments, such as in the application domain of information replication/retrieval [2].

However, the huge number of nodes involved in dense MANET exacerbate the challenging issue of scalability: designing highly scalable solutions for packet routing and information dissemination with minimum overhead is still a very open challenge in the field. In any type of MANET, either dense or not, the crucial technical solution to ensure scalability is the effective determination of clusters, i.e., groups of wireless nodes around one node (clusterhead) with special role (typically of intra-cluster coordination), and of gateways among different clusters, i.e., nodes located at the edge of at least two different clusters. Most research work on MANET clustering has focused on single-hop clustering, i.e., grouping determination where nodes are at most 1-hop distant from clusterheads. However, due to the limited area covered by 1-hop clusters, these clustering solutions relevantly suffer from node mobility. That motivates the increasing interest in k-hop clustering to both improve scalability (lower number of clusters to cover the same MANET deployment area) and to reduce the effect of clustering degradation due to mobility (mobility has a lower impact on cluster node membership as k increases) [3], [4], [5]. For instance, a few papers have already shown that k-hop clustering can help to achieve scalable solutions for several network management tasks: for packet routing, clusters represent a natural organization to localize routing information maintenance [6], while for Bluetooth scatternet formation clustering permits to minimize the number of composing piconets [3].

Given its relevance for scalability, the paper focuses on k-hop clustering and specifically addresses a primary aspect not yet widely investigated in the literature. We propose an original solution, taking advantage of the k-hop clustering process, to effectively build an optimal backbone connecting all the clusterheads chosen by a k-hop clustering solution. Let us note that this problem of backbone formation requires identifying the best and minimum set of intermediate nodes (gateways in k-hop clustering environments) to interconnect clusterheads, and not only to identify the best paths between any pair of clusterheads. The proposed k-hop backbone formation protocol is strongly based on localized intra-cluster interactions and avoids the need for multi-hop broadcasts.

The primary guideline is to promote the selection of nodes located at the borders of adjacent clusters and the choice of shared paths among clusterheads while maintaining the backbone connected, as better detailed in Section III. Under the dense MANET assumption, our proposal has demonstrated to outperform the few solutions already known in the literature especially in terms of imposed communication overhead.

In addition to the original proposal for k-hop backbone formation, the paper reports first simulation results about the viability and efficiency of the proposed solution. In fact, differently from most of the few proposals in the literature, we have implemented our protocol on top of ns-2 and extensively simulated it in wide-scale deployment environments. The preliminary results obtained are encouraging and provide useful insights for the general comprehension of the possible tradeoffs behind k-hop clustering, which still represents a frontier and largely unexplored research area. Finally, the reported results also provide a relevant feedback for the proposed solution and are guiding our on-going work on protocol refinement.

The remainder of the paper is organized as follows. Section II is devoted to provide the needed background about k-hop backbone formation and, most important, to provide an exact overview of the state-of-the-art in the field. Section III focuses on the description of our original proposal, and Section IV reports most significant experimental results, by detailing the simulated wide-scale execution environment. Conclusive remarks and directions of future research work end the paper.

II. Related Work

Some different formulations of the clustering problem have been proposed in the literature. Generally, clustering protocols aim at determining one manager entity for each cluster, i.e., the clusterhead [7]. Most work addresses the creation of clusters where all nodes are 1-hop distant from clusterheads (for a remarkable survey, see [8]). Some recent proposals, instead, try to determine k-hop clusters, i.e., to split and structure the MANET so that any node is within k hops from at least one clusterhead [3], [4], [5]. Finally, some solutions aim at forming disjoint clusters, i.e., to strictly partition nodes in disjoint groups [9], while others admit overlapping by introducing gateways belonging to different clusters at the same time [10].

Specifically focusing on k-hop clustering, a few solutions already exist. [3] proposes two novel protocols to identify k-hop clusters with at least $k+1$ hops between any clusterhead pair. These protocols are based on three stages: (i) every node becomes aware of its k-hop neighborhood and of *values* associated to k-hop neighbors: values are locally determined, e.g., depending on node ID (k-lowestID) or number of k-hop neighbors (k-CONID); (ii) one node floods a clustering request; (iii) according to the values received from k-hop neighborhood, every node decides whether to join a cluster or to create its own, and broadcasts its decision. Let us observe that the third stage is essential to completely cover the MANET: in fact, the decision of a node is not only influenced by the values of its k-hop neighbors, but also by their choice of being clusterhead or not [3].

In our previous contribution, we addressed similar constraints and proposed original heuristics for cluster minimization based on the idea of placing clusterheads at optimal distances (nearly $g = \lceil(\sqrt{3}k)\rceil$) via g-hop flooding [11]. Different clustering assumptions are in [4], where two clusterheads may be 1-hop distant and clusters are not necessarily contiguous, i.e., cluster members can be connected to their clusterhead via nodes belonging to different clusters. [4] is mostly relevant because it provides an interesting proof of the NP-completeness of the cluster minimization problem. Finally, [10] addresses the formation of clusters with no clusterheads, where any pair of nodes belonging to the same cluster is at most at k-hop distance, but this work only presents the implementation of solutions for $k = 1$. In the following, we overview related work in the specific area primarily addressed by the paper and still largely unexplored, i.e., k-hop clustering backbone formation.

A. Backbone Formation

A number of 1-hop backbone formation protocols have been proposed (see the survey in [8]), essentially based on the notion of Connected Dominating Set. Given a network graph G=(V,E) [8], a Connected Dominating Set is defined as a connected subgraph of G consisting of a set $S \subseteq V$ of nodes, so that every node in V either belongs to S or is one-hop distant from (at least) one node belonging to S.

To the best of our knowledge, due to the novelty and complexity of the issue, there is only one protocol for k-hop clustering backbone formation, AC-LMST [5]. AC-LMST has the primary goal of minimizing the number of backbone nodes by working on top of k-lowestID. First, each clusterhead collects information about all neighboring clusterheads; then, it tries to select and minimize gateways. During the first stage, every clusterhead aims to reveal its presence to the clusterheads responsible of adjacent clusters (defined as clusters with at least one node sharing a link with a node in the other cluster). Then, it broadcasts a message within $2k+1$ hops, which is the maximum possible distance between two adjacent clusterheads: that broadcast is also exploited to build shortest paths between clusterheads. Then, AC-LMST builds a Local Minimum Spanning Tree (LMST) connecting adjacent clusterheads, by using a metric based on number of hops; that is obtained by sharing distance information (i.e., shortest paths and weights) with adjacent clusterheads. Finally, every clusterhead is able to locally compute the LMST covering itself and its neighbors. Section III presents additional insights about this protocol, which will be thoroughly compared with our original proposal.

III. Localized Backbone Formation in k-hop Clustering

Highly populated and large MANET naturally suffer from scalability issues [12]. Clustering could be seen as the application of the well-known “divide et impera” principle to promote network organization by privileging local interactions only. However, MANET nodes need to communicate also with remote parties and that calls for global overlay structures superimposed to the cluster organization. Backbones enable effective remote (inter-cluster) communications by spanning the whole MANET via virtual connections joining adjacent clusters: messages between nodes belonging to different clusters can be conveyed on the backbone segment connecting their clusters. Therefore, any cluster includes a few special nodes that also belong to the backbone, termed *gateways* in the following; the number of gateways is very small if compared with the total number of MANET nodes in the cluster. Gateways are responsible for inter-cluster communications involving all nodes in their cluster and for forwarding backbone traffic. Backbone exploitation significantly increases network scalability by involving only the limited set of gateway nodes in several communication operations, thus permitting to avoid network flooding (in particular during topology discovery) [13].

In principle, a backbone can be created on top of 1-hop as well as k-hop clusters. In practice, some non-negligible differences in backbone formation exist, stemming from the difference in direct node visibility (or at least direct clusterhead visibility) between the two situations. In the 1-hop clustering case, backbones overlap with Connected Dominating Sets [8], i.e., any network node belongs to the backbone or is one hop away from a backbone node (see Section II). With regards to k-hop clusters, backbone segments span multiple hops within the same cluster and nodes are not always directly in the reach of backbone participants. On the one hand, this complicates backbone formation, since completely local protocols [8] cannot be applied any longer. On the other hand, it represents a significant opportunity to further improve network scalability, by requiring only a few nodes actively operating within any large k-hop cluster. Finally, the design of backbone formation solutions depends on the fact that they can exploit or not already determined clusterheads (see also the discussion in the previous section)[10]. Here, we focus on solutions exploiting elected clusterheads within each cluster, because clusterhead determination is crucial to facilitate backbone formation. Algorithms for the latter situation (known in the literature only for 1-hop clusters, even if formulated in the general framework of k-hop clusters) are provided in [10].

Broadly speaking, k-hop backbones can offer an effective support for information diffusion. A number of hybrid routing protocols can be devised, where proactive tables are maintained within each cluster, while reactive techniques apply to inter-cluster communications by exploiting gateway relays [13]. Active research in the field does not limit to packet routing but also includes techniques for resource dissemination in dense MANET [14]. In addition, backbones are particularly important in energy-poor scenarios, e.g., typical wireless sensor network applications, where only gateways must support communications while other nodes can temporarily turn off their wireless interfaces. We observe that this is exceptionally crucial if the sensing range of each device largely exceeds its communication range [15].

Notwithstanding the above advantages, due to the novelty of the research area, k-hop backbone solutions have not been sufficiently investigated so far. To the best of our knowledge, the only protocol proposed is AC-LMST [5]. AC-LMST is an effective solution but only focuses on clusterheads, without exploiting the potential benefit coming from the knowledge of cluster border nodes, which may act as suitable intermediaries between adjacent clusters (see the following section). In the following, we will explicitly distinguish between gateways, which are nodes belonging to the backbone, and *border nodes*, which belong to more than one cluster. In addition, AC-LMST requires a local topology learning phase (i.e., discovery of shortest paths toward adjacent clusterheads) that uses $2k+1$-wide broadcasts from any clusterhead. This imposes a communication overhead strongly dependent on node density, thus not fitting well the addressed deployment scenarios of dense MANET. We rapidly note that in these cases it is hard to apply the few broadcast overhead reduction schemes known in the literature [16] because k-hop clustering calls for determining shortest paths: any communication canceled by broadcast optimizations potentially increments the hop-distance between clusterhead and members, thus implicitly leading to construction of smaller and erroneous clusters.

Secondly, local spanning trees calculated among clusterheads do not necessarily identify optimal solutions, involving the smallest number of gateways. In fact, the choice of shared paths is often preferable to that of shortest ones. Let us illustrate the concept by means of the example in Figure 1 and consider nodes 1, 2, and 3. Shortest paths from 1 to 2 and from 1 to 3 include 4 and 5 gateways each. However, it can be easily identified a backbone solution with only 7 gateways (the circled nodes). Similar cases often occur in dense areas, with uniformly deployed nodes. Incidentally, this consideration also proves that the global minimum spanning tree method [5] does not provide an optimal solution. For the sake of briefness, we do not include here the ILP0-1 formulation, i.e., following the Integer Linear Programming formalism with variables allowed to assume only binary values, of this NP-complete problem. We observe that this formalization contributes to the clear definition of the problem, rather than to the design of feasible solution algorithms; interested readers can refer to a companion technical report associated to this paper [17]. Finally, we rapidly remark an extremely positive characteristic of AC-LMST: the exchange of $2k+1$ clusterhead information automatically rules out short loops (with length ≤ 4), which particularly hinder gateway optimization.

A. Border Nodes-Initiated Backbone Formation (BNI-BF)

Moving from the consideration that the clustering process leaves useful information on border nodes about adjacent clusters, we have designed and evaluated the original Border Nodes-Initiated Backbone Formation (BNI-BF). BNI-BF creates a connected backbone upon the cluster structure, joining clusterheads of adjacent clusters by exclusively leveraging highly localized *intra*-cluster interactions and avoiding multi-hop broadcasts. BNI-BF proposes an effective border selection strategy, allowing to determine shared paths between clusterheads (see Figure 1), while keeping the backbone connected. BNI-BF can be combined with any k-hop clustering strategy, e.g., k-lowest ID [3], k-CONID [3], Circle [11] respecting the following constraints: (i) all nodes are within k hops from at

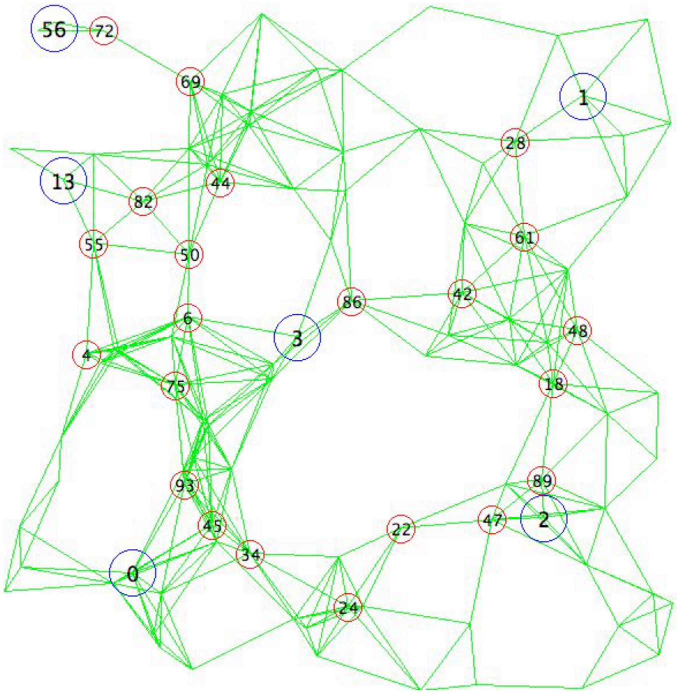

Fig. 1. Example of gateway selection using BNI-BF (100 nodes, $k = 3$). Larger circles represent clusterheads, smaller circles gateways, intersections nodes, and edges links.

least one clusterhead, and (ii) any clusterhead pair is separated from at least $k+1$ hops. Moreover, BNI-BF is particularly effective when combined with clustering algorithms that satisfy the following condition:

Condition 1: Let us assume that the topology is connected, and that clusters $1, 2...n$ are formed. Then, it is not possible to partition clusters in two or more sets $X_k = x_1, x_2, ..., x_s$, $k = 1, 2...m$ (with $x_1, ..., x_s$ clusters and $m < n$) where one cluster set X_i has no shared border nodes with any other set X_j. Formally, $\neg\exists i : \; BN(X_i, X_j) = \emptyset \;\; \forall j$ where $BN(i, j)$ is the set of border nodes shared between cluster set X_i and cluster set X_j.

This condition could be easily enforced also by slightly extending algorithms that do not satisfy it (e.g., k-lowestID and k-CONID [3]). First, we argue that if two or more cluster set partitions exist (and the network is connected), then there are at least two neighbor nodes belonging to disjoint cluster sets. We observe that this condition is a local translation of the general above defined Condition 1, being necessary but not sufficient for partitioning. Second, it is possible (and easy) to prove that only nodes k-hop distant from all their clusterheads can be on the boundary of a cluster partition (see the companion technical report). Then, we can amend these situations with single-hop broadcast exchanges from these crucial nodes, which are a really small node subset. In case one of these nodes realizes that its cluster set is completely disjoint from the cluster set of one of its neighbors, it records the entire set (and respective clusterheads) to a *SpecialClusters* list.

After the clustering process, every node knows its distance from the clusterheads of clusters it belongs to. Generally speaking, it also knows the $k+1$ distant clusterheads in case it is one of the special nodes above: in this stage, BNI-BF treats SpecialClusters exactly as normal clusters. BNI-BF operates by selecting borders placed in the intersection of the largest number of clusters as gateways between adjacent clusters. That establishes a sort of ranking among borders. Borders can autonomously communicate the list of the clusters they belong to (including SpecialClusters), to all clusterheads of these clusters. Thus, every clusterhead obtains a clear picture of the adjacent topology and can ideally select the best path toward adjacent clusterheads.

However, if all borders communicated their cluster list to neighbor clusterheads, the algorithm would impose a high communication overhead. Thus, we propose a heuristic that, even if unable to reach the optimal solution, can dramatically limit overhead. BNI-BF aims at letting best borders (i.e., at the intersection of the highest number of clusters) propagate their information before lower-ranked borders. We assume that the execution of BNI-BF on borders (or at least on nodes placed in the neighborhoods of adjacent clusterheads) can be loosely scheduled following different priority classes, based on the ranking choice. We implement it by properly setting local timers at the cluster formation stage. We rapidly note that even if this loose synchronization is not respected, the resulting backbone is still connected. Best borders forward their information toward clusterheads of their clusters with a very simple algorithm:

- Initially all borders are eligible to assume gateway role.
- Suppose node B is one of the best borders, and it is eligible to assume the gateway role. B locally (i.e., to single-hop neighbors) broadcasts the list of adjacent clusterheads and respective distances.
- All borders receiving this message compare clusterhead list against their own. If their clusterhead list is a subset of the received list, they dismiss their gateway eligibility.
- All nodes with a smaller hop distance from at least one of the clusterheads than the related value contained in the list are eligible to forward the message. We observe that, for any clusterhead included in the list, there must be at least one of B's neighbors improving that distance (it straightforwardly follows from the definition of k-hop clusters, see the companion technical report).
- These nodes enter a random backoff, after which they forward the message, by updating the clusterhead distances in the clusterhead list according to the following rule: for each clusterhead, every forwarder includes the minimum between the distance contained in the received message and its own.
- The last operation is defused if the node receives, during the random backoff, any message from its neighbors with all distance values better or equal to its own.

These operations are autonomously repeated by any single node until clusterheads are reached. Then, clusterheads collect messages and locally rank borders according to the number of connecting clusters; ties are broken by choosing lowest ID borders. Clusterheads determine a subset of borders allowing to reach all adjacent clusterheads. This is done according to the following greedy procedure:

- every time a border is selected, the list of all the connected clusterheads is included in a temporary list;
- another border is selected only if it permits to reach a clusterhead not included in the temporary list.

Even if this procedure is provably not optimal (i.e., does not determine the subset of borders with the smallest cardinality neither leads to a gateway set with minimum cardinality), nonetheless it permits to obtain a connected backbone provided that the underlying topology is connected. Finally, clusterheads deliver a confirmation message to all selected borders by following unicast paths. These paths can be locally established during the first stage of BNI-BF by storing routing information on each node forwarding borders messages. All nodes receiving the confirmation message become backbone gateways.

IV. Experimental Results

Differently from most related work in the field, which mainly concentrates on proposing first high-level guidelines of solutions for backbone formation by assuming ideal communication medium, we have extensively evaluated BNI-BF via more realistic ns2-based simulations [18]. In the following, we will show the two primary results obtained: (i) the size of the backbone built by BNI-BF is very similar to the AC-LMST one [5] even if our protocol exploits only local communications; (ii) BNI-BF relevantly outperforms AC-LMST in terms of imposed overhead, which is almost independent of node number for the proposed solution, thus well fitting dense MANET deployment scenarios.

We consider N stationary nodes deployed on a square region of size $S \times S$. We modeled communications as follows: IEEE 802.11 MAC protocol, 2.4 GHz transmission band, 11Mbps bandwidth, nominal radio range equal to 250m, and Two-ray Ground propagation model [19]. First, we deployed an increasing number of nodes, from $N = 50$ to $N = 200$, in an enlarging geographical region in order to maintain node density constant and equal to 6 (as in Figure 5 in [5]). All reported results are average values over 50 simulation runs. Figure 2 shows the total number of nodes belonging to the BNI-BF backbone. Obviously, the number of backbone nodes increases as N increases: however, it shows not to depend on N but on the deployment area (square size S, which increases with N in our experiments). Interestingly, the backbone size decreases as k increases, by confirming the effectiveness of k-hop clustering in dense MANET. A comparison with [5] shows that backbone size results are very similar. Let us notice that this holds even if BNI-BF does not exploit any specific strategy for loop avoidance: as future work, we are planning to integrate lightweight solutions removing loops of size ≤ 4 by exploiting topology data exchanged between adjacent clusterheads; with that enhancement we expect BNI-BF to outperform [5] also from the point of view of backbone size.

In addition, we have evaluated the message overhead required to build the backbone in BNI-BF. For the sake of clarity, the reported results include only the messages exchanged after cluster formation. In these runs, we deployed an increasing number of nodes (from $N = 50$ to $N = 200$) in a square of fixed size $S = 1500m$ (node density is not constant). As we expected, Figure 3 shows that the overhead is largely independent of the number of deployed nodes, thus being a crucial element for scalability in dense MANET. Values for $N = 50$ are significantly lower because the topology is disconnected for $S = 1500m$. Let us finally stress that the figure reports the gross total number of messages exchanged in the network: this is largely lower than N and the average communication overhead per node significantly decreases while growing N.

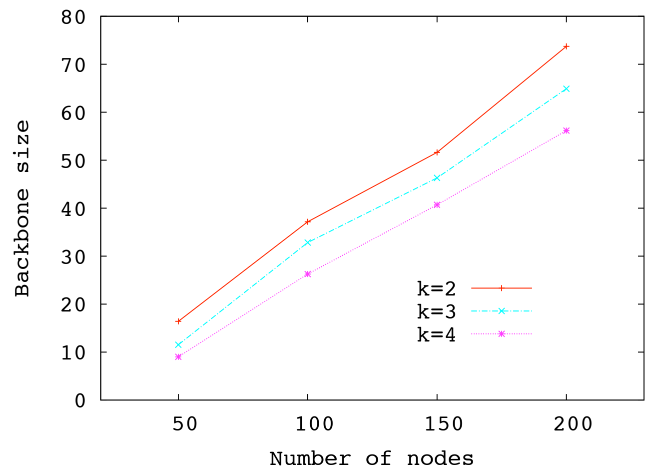

Fig. 2. BNI-BF backbone size

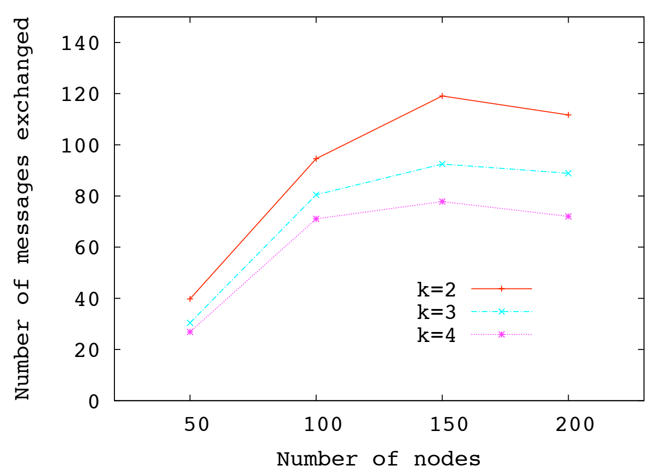

Fig. 3. BNI-BF message overhead

V. Conclusions

The challenging issues of scalable packet routing and information dissemination in dense MANET call for original solutions based on k-hop clustering. In particular, an open challenge is the determination of optimal backbones connecting clusterheads. Our research work has demonstrated that the assumptions of high node population and uniform node density both permit the design of novel backbone formation protocols, based on k-hop clustering awareness and with limited overhead.

The first encouraging simulation results reported in the paper are stimulating additional research activities. In particular, we are exploring lightweight decentralized algorithms to avoid the inclusion of short loops during backbone formation; that could show a relevant positive impact on both backbone size and message overhead.

ACKNOWLEDGEMENT

We are pleased to thank Davide Tartaglione and Massimo Barbati for their collaboration in the earlier stages of this research. This work is partially supported by the Italian MIUR PRIN MOMA and the CNR IS-MANET projects.

REFERENCES

[1] I. Chlamtac, M. Conti, and J.-N. Liu. Mobile ad hoc networking: Imperatives and challenges. *Elsevier Journal on Ad Hoc Networks*, 1(3), Jul. 2003.

[2] P. Bellavista, A. Corradi, and E. Magistretti. Lightweight autonomic dissemination of entertainment services in wide-scale wireless environments. *IEEE Communications Magazine*, 43(6), Jun. 2005.

[3] F. G. Nocetti, J. Solano-Gonzales, and I. Stojmenovic. Connectivity-based k-hop clustering in wireless networks. *Kluwer Telecommunications Systems*, 22(4), Apr. 2003.

[4] A. D. Amis, R. Prakash, T. H.-P. Vuong, and D. T. Huynh. Max-min d-cluster formation in wireless ad hoc networks. In *IEEE INFOCOM*, Mar. 1999.

[5] S. Yang, J. Wu, and J. Cao. Connected k-hop clustering in ad hoc networks. In *IEEE ICPP*, Jun. 2005.

[6] Y. T. Mingliang Jiang and J. Li. Cluster-based routing protocol. IETF Internet draft, Jul. 1999.

[7] A. Ephremides, J. E. Wieselthier, and D. J. Baker. A design concept for reliable mobile radio networks with frequency hopping signaling. *Proceedings of the IEEE*, 75(1), Jan. 1987.

[8] S. Basagni, M. Mastrogiovanni, A. Panconesi, and C. Petrioli. Localized protocols for ad hoc clustering and backbone formation: A performance comparison. *IEEE Transactions on Parallel and Distributed Systems*, 17(4), Apr. 2006.

[9] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 14(7), Jul. 1997.

[10] P. Krishna, N. Vaidya, M. Chatterjee, and D. K. Pradhan. A cluster-based approach for routing in dynamic networks. *AMC Computer and Communication Review*, 49(2), Feb. 1997.

[11] G. Angione, P. Bellavista, A. Corradi, and E. Magistretti. A k-hop clustering protocol for dense mobile ad-hoc networks. In *IEEE IWDDS Workshop*, Jul. 2006.

[12] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. In *ACM MOBICOM*, Seattle, WA, USA, Aug. 1999.

[13] D. Kim, S. Ha, and Y. Choi. K-hop cluster-based dynamic source routing in wireless ad-hoc packet radio network. In *IEEE VTC*, May 1998.

[14] P. Bellavista, A. Corradi, and E. Magistretti. Comparing and evaluating lightweight solutions for replica dissemination and retrieval in dense manets. In *10th IEEE ISCC*, July 2005.

[15] H. Zhang and J. Hou. Maintaining sensing coverage and connectivity in large sensor networks. *Wireless Ad Hoc and Sensor Networks: An International Journal*, 1(1-2), Jan. 2005.

[16] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol. In *IEEE INMIC*, Dec. 2001.

[17] P. Bellavista and E. Magistretti. k-hop Backbone Formation and Cluster Maintenance in Ad Hoc Networks. Technical report, DEIS LIA, University of Bologna, Apr. 2007.

[18] ns-2 (The Network Simulator). http://www.isi.edu/nsnam/ns/.

[19] TS Rappaport. *Wireless Communications: Principles and Practice*. IEEE Press Piscataway, NJ, USA, 1996.