Mobile Ad Hoc Backbones for Multi-Radio Networks

Stefano Basagni
Dept. of Electrical and Computer Engineering
Northeastern University
Boston, MA
Email: basagni@ece.neu.edu

Maurizio A. Nanni
Dept. of Electrical and Computer Engineering
Northeastern University
Boston, MA
Email: maurinan@gmail.com

***Abstract*—We present a new distributed protocol for setting up and maintaining a backbone for multi-hop mobile networks with multiple radio interfaces. Our solution, termed MM-Backs, is effective in producing backbones with limited size, while being robust and reliable in the face of node mobility. We also describe how MM-Backs can be used to transparently support ad hoc routing, without the need to modify a routing protocol to adapt to multiple radios. We compare our protocol with a solution for mobile backbones running in single-radio networks. MM-Backs provides backbones of reasonable size, with superior connectivity, shorter route lengths, higher resilience and lower maintenance-related overhead. When running AODV over MM-Backs backbones we observe a remarkably higher packet delivery ratio and lower overhead than when running AODV over single-radio backbones or by itself.**

I. INTRODUCTION

In this paper we tackle the problem of defining and testing a new protocol for mobile, multi-hop and multi-radio networks that builds and maintain a *backbone*, i.e., a communication structure formed by selected nodes and by the links among them. The typical *raison d'être* of a backbone in a multi-hop network is that of providing a hierarchy among the network nodes that, by creating a smaller communication structure, enhances the scalability of network protocols and operations. The importance of efficient backbone construction and maintenance in mobile multi-hop radio networks is witnessed by the host of research papers dedicated to this topic. Baker and Ephremides [1] started exploring it in the late seventies, and since then the problem has been revisited for routing [2] and multimedia [3] purposes, among many. More recently, backbone construction has been perfected to efficiently deal with node mobility. This is obtained by making backbone formation and maintenance operations a part of common network primitives, such as basic neighbor discovery and link management, without incurring relevant overhead. Typical examples are provided by ETSA [4], GDMAC [5], TRUNC-K [6] and M-Backs [7]. All these protocols have been defined for mobile, multi-hop *single-radio* networks, where communication happens through a single radio, over a single channel. This paper is concerned with exploring solutions for mobile backbones for *multi-radio* networks. To the best of the authors' knowledge there is only another backbone formation protocol that works on multi-hop networks with two radios. This is a variant of the ETSA protocol proposed by Ju and Rubin for backbone synthesis over mesh networks [8]. The protocol, however, does not use both interfaces for backbone formation: ETSA is applied to the long-range interface to build the mesh backbone, and the second interface is used for providing connectivity to the user of the mesh network.

The protocol we present in this paper, termed *MM-Backs* for Mobile Multi-radio Backbones, bases its operations on two radio interfaces with different transmission radii, r and R, with $r < R$. While exchanging HELLO packets over the topology formed by r, backbone links are formed and maintained over the longer range radio. MM-Backs achieves all properties that are desirable for efficient backbone construction: (a) The protocol does not require nodes to be synchronized; (b) information is carried over packets (the HELLO packets) that are needed anyway for basic network operations; (c) the size of the HELLO packets remains manageably small; (d) the protocol is based on very simple code, and produces and maintains a backbone in constant time, i.e., in a time that does not depend on the number of nodes in the network; (e) nodes in the backbone are the best suited for the job (this may dynamically change in time); (f) the protocol does not require extra hardware (e.g., GPS) and does not constrain the mobility of the nodes.

We also describe how our multi-radio approach to backbone formation can be transparently used to improve the performance of other communication protocols, such as ad hoc routing. We show how MM-Backs can enable multi-radio routing without the routing itself needing to be modified for multi-radio nodes. We finally evaluate the performance of MM-Backs through ns-2/MIRACLE-based simulations. In order to assess the effectiveness of deploying multi-radio networks we compare MM-Backs and M-backs in terms of the characteristics of the backbones they produce (e.g., size, how long they are connected, how resilient and robust they are to link failure, the length of their routes) and also with respect to the support they can provide to an ad hoc, on-demand routing protocol such as AODV [9]. We observe that MM-Backs remarkably outperforms M-Backs (which in turn has been shown to perform better than ETSA, TRUNC-K and GDMAC in [7]), and that, especially when AODV traffic is offered to the network, it is successful in delivering a larger number of

978-1-4673-0437-5/12/$31.00 ©2012 IEEE

data packets even at very high traffic, when AODV/M-Backs performance dives down because of congestion and AODV with no backbone support cannot even find routes, let alone deliver any packet.

II. MM-BACKS: MOBILE MULTI-RADIO BACKBONES

In the description of MM-Backs we use the following notation. Each node is characterized by its unique identifier (ID) u. We indicate with HN (*head node*), AN (*associate node*) and RN (*regular node*) the three possible roles a node can assume. HNs form the backbone, ANs are nodes with backbone capabilities but not in the backbone, and RNs are nodes with no backbone capabilities. We indicate with $\textit{role}(u)$ the current role of node u. We also indicate with $\textit{headID}(u)$ the ID of the HN with which u is currently affiliated. Each node u computes a *weight* $\textit{wt}(u)$, which is a real number indicating its suitability to serve as backbone node (this is only relevant for backbone capable nodes). This weight can change in time, and may force a node to change its role. For instance, the weight of a node can be based on its mobility rate, available energy, computational or storage capacity, congestion level or some other application-dependent measurable metrics. The set $N_\rho(u)$ is defined as the set of node u 1-hop (or direct) neighbors according to ρ, i.e., the nodes with which u can communicate directly. We also define $N_\rho^+(u) = N_\rho(u) \cup \{u\}$. The multi-radio networks we consider here have nodes with two radio interfaces I_r and I_R, with transmission radii r and R, respectively, with $r \leq R$.

HELLO packets have a fundamental role in MM-Backs in that they are not just used to discover new neighbors or to detect whether the link to a neighbor is still available.

Nodes include their ID, weight, role, ID of the HN they affiliate with and the list of the HNs that are one and two hops away with respect to I_r. Through receiving HELLO packets, node u maintains a *neighbor table* containing information on all the nodes currently in $N_r(u)$. If u is a HN it also maintains a *backbone neighbors table* that stores information about all the HNs that are currently in $N_R(u)$.

Each node u starts MM-Backs by simply broadcasting its own ID, weight and role to whomever is currently within its transmission radius r. The initial role of a node is either AN or RN, depending on whether it is backbone capable or not, respectively. Node u then keep executing MM-Backs by periodically running the *UpdateRole* procedure (Algorithm 1).

Algorithm 1 *UpdateRole*()

```
1: $\textit{HeadConversion}()$
2: $\textit{sendHELLO}(I_r)$
```

This procedure starts by calling the *HeadConversion* procedure, by the end of which node u has decided its role (HN, AN or RN). This role is then communicated by transmitting a HELLO packet over I_r (line 2).

The *HeadConversion* procedure is described by Algorithm 2.

Algorithm 2 *HeadConversion*()

```
if ($\textit{role}(u) = $ RN) then
    $\textit{headID}(u) \leftarrow \max_{wt(v)}\{v \in N(u)\}$
else
    if $\exists\ v \in N(u)$ s.t. $\textit{headID}(v) = u$ then
        $\textit{headID}(u) \leftarrow u$
    else
        $\textit{headID}(u) \leftarrow \max_{wt(v)}\{v \in N^+(u)\}$
    if $\textit{headID}(u) = u$ then
        $\textit{role}(u) \leftarrow$ HN
    else
        $\textit{role}(u) \leftarrow$ AN
```

As an example of MM-Backs, consider the 13-node network depicted in Fig. 1. The links induced by I_r are indicated by solid lines. For ease of description we assume that each node has a unique weight whose value ranges in the set $\{1, 2, \ldots, 13\}$, that all nodes are backbone capable and that they start MM-Backs approximately at the same time. Initially all nodes are ANs. After having collected information about the weight of its neighbors over I_r, each node executes *UpdateRole*. This triggers the execution of the *HeadConversion* procedure. Nodes 11, 12 and 13, being the nodes with the highest weight in their neighborhood, become HNs. (They are the shaded nodes in the figure.) Every other node instead selects the nodes with the highest weight in its neighborhood. For instance, node 8 chooses node 13 as the HN with which it affiliates; Node 3 selects node 12; node 9 chooses node 11, and so on. In broadcasting its role, node 8 includes in its HELLO packet that it is one hop away from HNs 11 and 13. Through this information, these HNs are made aware that they are two hops away from each other, and set up a link over I_R. Similar actions are taken by nodes 3, 11 and 12. The backbone is therefore formed (over I_R) by nodes 11, 12 and 13. (The backbone links are depicted as dashed line.)

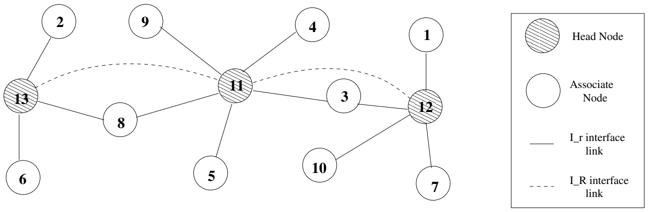


Fig. 1. Example of multi-radio backbone.

III. SUPPORT FOR COMMUNICATION PROTOCOLS

MM-Backs can be used to transparently support communication protocols originally designed for flat ad hoc network topologies, such as AODV, DSR, geo-based and more. More specifically, the backbone module can provide service to routing (or other protocols) by catching its packets, whether control or data ones, and forwarding them using different radios for different tasks. In fact, the routing protocol does not even have to know that the network is in fact a multi-radio network, or that it is hierarchically organized: It is MM-Backs that decides whichever radio to use and for which routing task.

For example, for an on-demand routing protocol such as AODV [9] that uses RREQ and REPLY control packets for route discovery, MM-Backs can support enhanced performance by implementing the route discovery process over the backbone: RREQ packets are broadcast over G_R to reach the destination, and REPLY packets use the backbone back to the source node by unicast. The actual routing of data packet is performed, as usual, on the flat network topology G_r. Since route discovery is performed on the backbone, a route is returned to the source node as a set of HNs. However, data routing is performed over G_r. The conversion between inter-HN links (G_R links) and paths over G_r is performed by MM-Backs that, using its neighbor table, provides the correct next hop in G_r given the next hop HN in the backbone. We notice that while the route of HNs that the packet is to follow (over G_R) is given, the actual path (over G_r) between each pair of HNs is computed on the fly, when the packet arrives at each HN. This makes AODV more resilient to mobility, as we observe clearly in our experiments (Section IV).

Algorithms 3 and 4 describe the procedures executed by MM-Back upon receiving a packet from the upper layer (e.g, routing) or the lower layers (MAC), respectively.

Algorithm 3 *BackboneRecvFromUpperLayer*(*pkt*)

```
1: if isControlPacket(pkt) then
2:     sendDown(pkt, $I_R$)
3: else
4:     pkt.destHN = pkt.next_hop
5:     pkt.next_hop = getLocalNextHop(pkt.destHN)
6:     sendDown(pkt, $I_r$)
```

When a packet *pkt* arrives at the backbone module of node u from the upper layer, u first checks whether *pkt* is a routing control packet. If this is the case, the packet is sent down to interface I_R (Algorithm 3, lines 1–2) for forwarding on the backbone. Otherwise, it is forwarded to the next HN, as indicated by the routing module in the *next_hop* field of the packet. In particular, node u assigns to the *destHN* the next hop as indicated by AODV, and computes the actual next hop in G_r through the procedure *getLocalNextHop* that uses the node neighbor table for this task (Algorithm 3, lines 4–6). The packet is then passed down to I_r.

Algorithm 4 *BackboneRecvFromLowerLayer*(*pkt*)

```
1: if isControlPacket(pkt) then
2:     if (role($u$) = HN $\vee$ pkt.dst $\neq$ bcast_addr $\vee$ pkt.RREQdest = $u$)
       then
3:         sendUp(pkt)
4:     else
5:         discard(pkt)
6: else if (pkt.dst = $u$ $\vee$ pkt.destHN = $u$) then
7:     sendUp(pkt)
8: else
9:     pkt.next_hop = getLocalNextHop(pkt.destHN)
10:    sendDown(pkt, $I_r$)
```

If a packet *pkt* arrives at the backbone module of node u from the lower layer, u checks if the packet is a control packet. If this is the case, the backbone module sends the packet up to the routing module if u is a HN or if the packet has been transmitted in unicast mode or if node u is the destination of the route discovery process; otherwise the packet is discarded (Algorithm 4, lines 1 to 5). For instance, if the routing protocol in use is AODV, the backbone module discards all the RREQ packets if u is not a HN, while it would send up those packet sent by unicast such as REPLY packets even if u is a AN. If the packet *pkt* is a data packet, node u checks whether it is the packet recipient or if it is the destination HN of the forwarding process on G_r. If this is the case, it sends up the packet to the routing module (Algorithm 4, lines 6–7). Otherwise, the packet is forwarded to the next hop HN (Algorithm 4, lines 9–10).

For example, consider the scenario depicted in Fig. 1. If a packet has to be delivered from source node 2 to destination node 1, it will be forwarded as follows. Assuming that node 2 has no route to node 1, AODV creates an RREQ packet that is broadcast over I_R (Algorithm 3, lines 1–2). The RREQ packet is received by the backbone module and forwarded over the backbone (dashed links) from HN 13 to HN 12, while it will be discarded by the ANs that can hear the broadcast. HN 12 is responsible for the packets destined to node 1, since AN 1 is affiliated with it. Therefore, HN 12 responds with a REPLY packet that is forwarded back over I_R. The discovered route is the path $P = (2, 13, 11, 12, 1)$, a G_R path. Once the REPLY packet gets to the node that started the discovery process (AN 2), the data packet can be forwarded towards the destination node. In particular, the packet is forwarded to the next HN 11 through AN 8. HN 13 knows how to forward the packet to the next hop 11 of path P. This information is stored in its neighbor table, which is maintained up to date through HELLO packets, and retrieved through the *getLocalNextHop* procedure. Similarly, once the packet reaches HN 11, it is forwarded towards HN 12 through AN 3. Finally, once the packet reaches HN 12, it is delivered to its final destination AN 1, which is affiliated with HN 12.

IV. PERFORMANCE EVALUATION

In order to assess the effectiveness of building backbones over multi-radio nodes and using them for routing support we compared the performance of MM-Backs with that of its single-radio counterpart, M-Backs [7]. We selected M-Backs because its operation are very similar to those of MM-Backs (which makes the comparison fairer and aimed at showing the advantages of multi-radio), and also because we have shown that M-Backs outperforms previously proposed solutions [7]. Running on single-radio network G_r, M-backs needs to build backbone links between HNs that are at most three hops away by choosing intermediate ANs, called gateways nodes (GNs). Therefore a "virtual" M-Backs link is implemented through at most three physical links on G_r. As a consequence, for GN selection, HELLO packets contain additional information on the actual composition of routes between pairs of HNs [7]. Equally important, we compared MM-Backs and M-Backs with AODV [9] run over the flat network topology, which shows how by using the backbone to spread and gather routing control packets enables routing scalability, allowing packet delivery where AODV by itself would fail. We remark that this is obtained in a totally transparent way: AODV does not need to be modified (we used the same implementation to run on both the flat network and on the backbones).

We implemented MM-Backs, M-Backs and AODV in the VINT Project *ns-2* Simulator and its extension ns-MIRACLE. We consider nodes with two radios, one with high data rate and shorter range and one with lower data rate but longer transmission radius. The high data-rate interface is set to perform as a 802.11g card with transmission radius $r = 150$m and data rate 18Mbs. The second wireless interface simulates a 802.11b card, with a much lower data-rate (1Mbs) and a transmission range of about 450m. Transmissions on the two interfaces occur on orthogonal (i.e., non-interfering) channels. Our experiments concern networks with 200 to 500 nodes initially scattered uniformly and randomly over a square area with side $L = 1000$m. Nodes then move according to the Gauss-Markov mobility model [11]. This model is designed to adapt node movements to different levels of randomness via one tuning parameter $\alpha \in [0, 1]$. Results shown here have been obtained by setting $\alpha = 0.5$. Nodes move at the average speed of 2.5m/s. The *HelloPeriod* is set to 2s. If a node does not receive a HELLO packet from a neighbor for 6s or more their link is considered dead, and that neighbor entry is removed from the node neighbor table. Simulations run for 10000s. Nodes start their operation randomly during the first 50s of simulation. Metrics are measured after the first 100s of simulations. Each point shown is achieved with 95% statistical confidence within 5% precision.

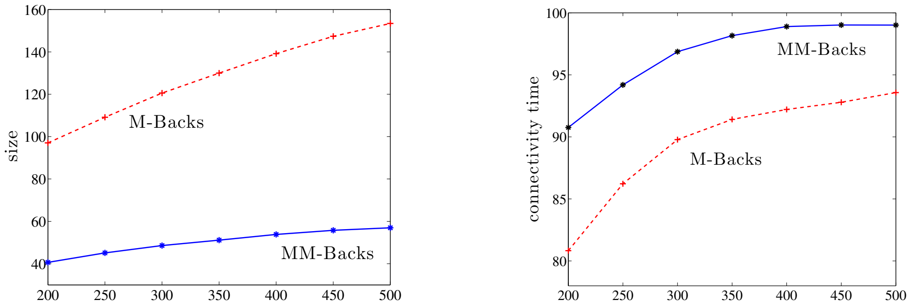
A. Backbone metrics

The following metrics concern key topology characteristics of a multi-hop backbone: *Backbone size*: The number of nodes in the backbone (HNs in MM-Backs, HNs and GNs in M-Backs). *Backbone connectivity*: The percentage of simulation time during which the backbone is connected. *Backbone resilience*: The number of links that should be removed in order to disconnect the backbone. *Route length*: The length (in hops) of the shortest route between any two nodes, passing through the backbone. For this set of metrics measures are collected by taking a network snapshot every second of the simulation time after the first 100s. Results are shown in Fig. 2. Fig. 2(a) depicts the average backbone size as the number of network nodes increases. MM-Backs clearly takes advantage of interface I_R that allows it to build smaller backbones. Their size is quite contained, being always between 40 and 57 nodes, i.e., always below one fifth of the number of nodes in the network. M-Backs needs to select GNs to build a connected backbone: Its size ranges between 97 and 153 nodes. In Fig. 2(b) we show the average backbone connectivity. Connectivity of MM-Backs backbones is maintained in more than 90% of the simulation time, because of the longer lasting I_R links when nodes move. M-Backs is able to maintain the backbone connected only for 82% of the simulation time in sparser networks, and for 94% of the time in networks with $n = 500$. The reasons why MM-Backs backbones are sometimes disconnected (even when the flat network topology is connected) are the time it takes to a node to react to topology changes. In Fig. 2(c) we show the average backbone resilience, i.e., the average number of links that should fail in order to disconnect the backbone. MM-Backs backbone resilience ranges between an average of 170.42 links ($n = 200$) and 450.12 links ($n = 500$). M-Backs, instead, goes from an average of 63.98 backbone links in sparse networks to 291.1 links in the denser ones. Routes are fairly short in MM-Backs. Their average length (in hops) ranges between 1.98 in networks with 200 nodes, and 1.95 in networks with 500, as depicted in Fig. 2(d). Clearly, the shorter radius r used by M-Backs imposes longer routes: Their length averages always above 4 hops, independently of network size.

B. Routing support

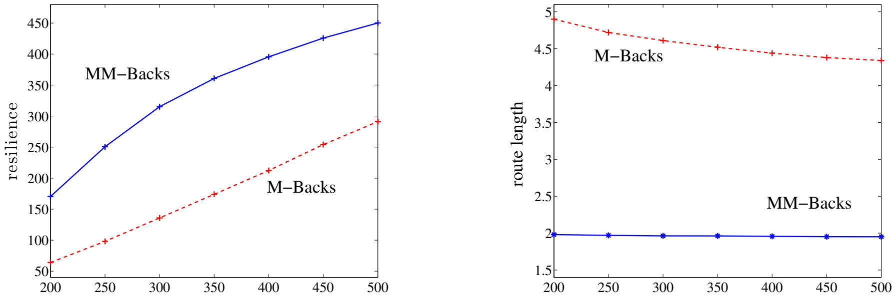
We investigate how MM-Backs and M-Backs support a routing protocol such as AODV by disseminating control packets (RREQ, REPLY, etc.) only through the backbone. Data packets are routed through the flat network topology, using the interface I_r in both protocols. Packets are assigned different priorities: HELLO packets have have the highest, then control packets and then data packets. The queue size has been set to 100 packets. When capacity is reached, packets are dropped according to their priority. Data packet are UDP packets. Their size is 500B. They are generated according to a Poisson arrival process with parameter λ ranging in $\{5, 15, 25, \ldots, 95\}$ (number of packets in the network per second). For each new packet a source and a destination are randomly selected among the nodes. The metrics we investigate are the following: *Packet delivery rate*: The percentage of packets correctly delivered to their intended destination. *End-to-end packet delay*: The average end-to-end packet delay of delivered packets, and *Number of RREQ*: The average number of RREQ packets flooded through the network every second.

In Fig. 3(a) and Fig. 3(b) we show the average packet delivery rate in networks with 300 and 500 nodes, respectively, as the traffic rate λ increases. (Results for networks with 200 and 400 nodes show similar trends.) MM-Backs always succeeds in delivering at least 94% of the data packets to their intended destination, independently of the traffic in the network. M-Backs, on the other hand, delivers about 94% of the data packets in a low traffic scenario. Increasing λ to 95 it successfully drives home only an average of 74% packets. This is mainly due to the fact that through the second radio interface inter-nodal routes are shorter in MM-Backs backbones, and that the forwarding of AODV control packets takes advantage of the higher resilience to mobility shown by MM-Backs (Section IV-A). Furthermore, routes in MM-Backs are defined as sequences of HNs, and forwarding of data packets is based on routes (over I_r) that are computed *while the packet is en route*, always finding the most current way to get to the next HN. In M-Backs, instead, the entire route is defined during the routing discovery phase and it is more likely that mobility invalidates it. In denser networks ($n = 500$) both protocols have to deal with congestion, here defined to happen when the route discovery failure exceed



(a) Size.

(b) Connectivity.

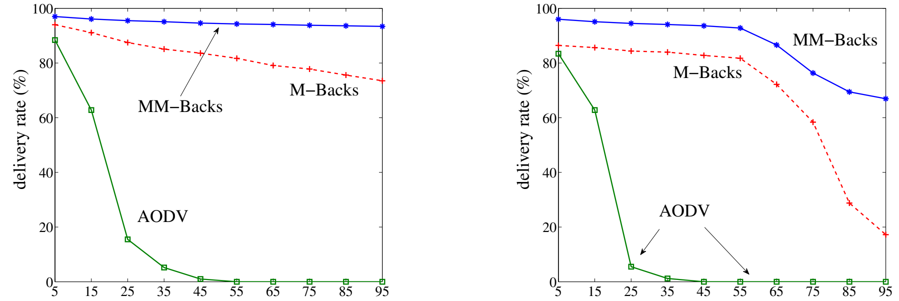


(c) Resilience.

(d) Route length.

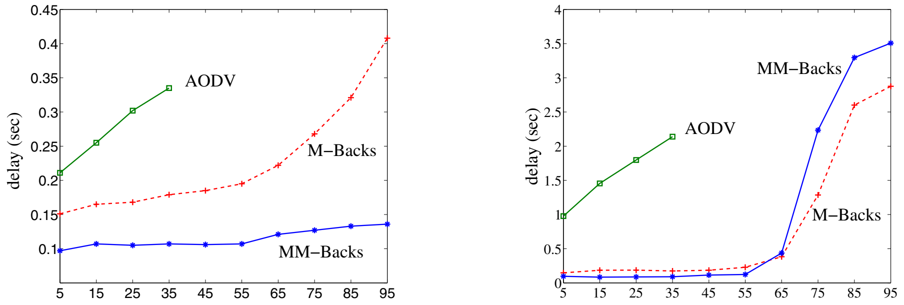
Fig. 2. Backbone metrics vs. increasing network size.

50%. In the considered scenarios congestion occurs when $\lambda > 55$. When traffic is high ($\lambda = 95$) MM-Backs is able to deliver about 67% of the data packets. M-Backs, instead, delivers only 17%. The almost fourfold better performance of MM-Backs is mainly due to the different routes followed by control and data packets. Since MM-Backs dedicates a radio interface (I_R) to route discovery and since its routes are on average at least twice shorter than those in M-Backs, route discovery has higher chances to be successful on MM-Backs backbones. For M-Backs, instead, the route discovery rate decreases considerably and many data packets are dropped at the source node for routing discovery timeout. In normal traffic conditions ($\lambda \leq 55$), M-Backs delivers on average between 82% and 86% of generated packets, while MM-Backs average ranges between 93% and 96%, which is still consistently better. The difference made by using a backbone to discover routes through a backbone is shown by the performance of AODV over the flat network topology. In both scenarios ($n = 300$ and $n = 500$) AODV packet delivery rate is above zero only for $\lambda \leq 35$, and even when the traffic load is fairly low ($\lambda = 15$) it is able to deliver just a bit over 60% of the offered traffic. When $\lambda = 25$ AODV delivery rate is already below 20%. In this scenarios using a backbone, and especially a multi-radio one, literally *enable* routing, i.e., allows AODV to function where it could not. Figures 3(c) and 3(d) show the average end-to-end delay for data packet delivered successfully in networks with 300 and 500 nodes, respectively. In the sparser case, MM-Backs takes an average of 100ms to 130ms in *any* traffic condition to deliver a packet. M-Backs latency averages between 160ms and 200ms in traffic scenarios without congestion ($\lambda \leq 55$), while it increases up to 400ms when $\lambda > 55$. For networks with 500 nodes both protocols have to deal with congestion. In congested scenarios ($\lambda > 55$) MM-Backs delays are slightly higher than those of M-Backs. The reason is to be found in the considerably higher number of packets that MM-Backs is able to deliver. (The average latency is computed over the data packets successfully delivered by MM-Backs, which, as mentioned, in this case are up to four times as many as those delivered by M-Backs.) AODV delays are shown only for traffic rates for which it delivers packets ($\lambda \leq 35$). We observe that in both scenarios the delay reduction provided by MM-Backs is at least 52.3% ($n = 300$, $\lambda = 5$), reaching 95.5% for dense networks and $\lambda = 35$. We also investigate the average number of RREQ packets flooded through the network each second. This metric allows us to determine which backbone formation algorithm is more effective in reducing route discovery overhead. As expected, the amount of control traffic increases with the



(a) Delivered packets, $n = 300$.

(b) Delivered packets, $n = 500$.



(c) End-to-end data packet delay, $n = 300$.

(d) End-to-end data packet delay, $n = 500$.

Fig. 3. Traffic-related metrics in mobile backbones vs. increasing traffic (packets per second).

number of packets injected in the network. When $n = 300$ MM-Backs takes advantage of the shorter routes over I_R and shows a 59% decrease in routing overhead compared to M-Backs. In denser topologies, as before, both protocols face congestion when $\lambda > 55$. However, MM-Backs, given the reduced backbone size and its shorter routes, is able to better deal with congestion, showing a reduction in routing overhead of 61% with respect to that of M-Backs. In both scenarios, the average number of RREQ packets per second of AODV over the flat network topology spins out of control already for small values of λ. For instance, when $\lambda = 25$ and $n = 300$ the number of AODV RREQ packets throughout the network are 305.2% (131.4%) more than those of MM-Backs (M-Backs).

V. CONCLUSIONS

In this paper we proposed MM-Backs, a distributed protocol for building and maintaining a connected backbone for mobile multi-radio ad hoc networks. We demonstrated the effectiveness of our multi-radio approach by comparing, via simulations, the performance of MM-Backs with its single-radio counterpart, M-Backs. MM-backs is also effective in supporting routing (here, AODV) by remarkably reducing routing control overhead and enabling a higher percentage of data packets correctly delivered to their destination.

REFERENCES

[1] D. J. Baker and A. Ephremides, “The architectural organization of a mobile radio network via a distributed algorithm,” *IEEE ToC*, vol. COM-29, no. 11, pp. 1694–1701, Nov. 1981.
[2] R. Sivakumar, P. Sinha, and V. Bharghavan, “CEDAR: A core-extraction distributed ad hoc routing algorithm,” *IEEE JSAC*, vol. 17, no. 8, pp. 1454–1465, August 1999.
[3] C. R. Lin and M. Gerla, “Adaptive clustering for mobile wireless networks,” *IEEE JSAC*, vol. 15, no. 7, pp. 1265–1275, Sept. 1997.
[4] H.-J. Ju and I. Rubin, “Mobile backbone synthesis for ad hoc wireless networks,” *IEEE ToWC*, vol. 6, no. 12, pp. 4285–4298, Dec. 2007.
[5] M. A. Nanni and S. Basagni, “Mobile ad hoc backbones: Formation and maintenance,” in *Proceedings of IEEE RWS 2010*, New Orleans, LA, January 10–14 2010, pp. 613–616.
[6] S. Lee, B. Bhattacharjee, A. Srinivasan, and S. Khuller, “Efficient and resilient backbones for multihop wireless networks,” *IEEE Transactions on Mobile Computing*, vol. 7, no. 11, pp. 1349–1362, 2008.
[7] M. A. Nanni and S. Basagni, “M-Backs: Mobile backbones for multihop wireless networks,” in *Proceedings of IEEE WCNC 2011*, Cancun, Mexico, March 28–31 2011.
[8] H.-J. Hu and I. Rubin, “Backbone topology synthesis for multiradio mesh networks,” *IEEE JSAC*, vol. 24, no. 11, pp. 2116–2126, Nov. 2006.
[9] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, “Ad hoc on-demand distance vector (AODV) routing,” Request for Comments: 3561—MANET Working Group of the IETF, July 2003.
[10] S. Basagni and M. A. Nanni, “Mobile multi-radio ad hoc backbones: Definition, correctness and performance evaluation,” ECE, Northeastern U., Boston, MA, Tech. Rep. 01/2011, April 2011.
[11] F. Bai and A. Helmy, “A survey of mobility models in wireless ad hoc networks,” in *Wireless Ad Hoc and Sensor Networks*. Springer, October 2006, pp. 275–300.