UC Santa Cruz

UC Santa Cruz Previously Published Works

Title

Robust and Scalable Integrated Routing in MANETs Using Context-Aware Ordered Meshes

Permalink

<https://escholarship.org/uc/item/0n28p387>

Author

Garcia-Luna-Aceves, J.J.

Publication Date

2010-03-14

Peer reviewed

Robust and Scalable Integrated Routing in MANETs Using Context-Aware Ordered Meshes

Rolando Menchaca-Mendez[§]

[§]Computer Engineering Department
University of California, Santa Cruz
Santa Cruz, CA 95064, USA
Email: menchaca@soe.ucsc.edu

J.J. Garcia-Luna-Aceves^{§,*}

^{*}Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304, USA
Email: jj@soe.ucsc.edu

Abstract—This paper introduces CAROM (Context-Aware Routing over Ordered Meshes), a new context-aware routing framework for multicast and unicast routing in mobile ad hoc networks. We use the concept of region of interest to identify connected components of the network that span sources and destinations to restrict signaling to these regions. Then, context information is used to compute routing meshes composed of shortest-paths located inside of the regions of interest. Experimental results based on extensive simulations show that CAROM attains similar or better data delivery and end-to-end delays than traditional unicast and multicast routing schemes for MANETs (AODV, OLSR, ODMRP), and that CAROM incurs only a fraction of the signaling overhead of traditional routing schemes.

I. INTRODUCTION

The application requirements of mobile ad hoc networks (MANETs) are in stark contrast with the way in which today's MANET routing protocols operate. First, supporting both point-to-point and many-to-many communication in a MANET involves running a unicast and a multicast routing protocol in parallel, which is very inefficient. Second, most of the routing protocols proposed in the past only take into account topological information when establishing their routing structures. The latter is particularly disadvantageous in the context of MANETs, where topologies change not only because of node mobility but because of physical-layer effects (e.g., fading and noise). The creation of a route in a MANET impacts the contention perceived by each node in the route, as well as any other node within interference range of the route. Furthermore, because bandwidth is scarce and links are multiple access, the transmission queues of multiple nodes interact with each other. As our summary of related work in Section II indicates, existing routing protocols for MANETs are such that: (a) they support either unicast routing or multicast routing; (b) the dissemination of signaling traffic in MANETs is not closely linked to the interest that nodes have on destinations, and is structured as either strictly on-demand, strictly proactive, or the use of both types of signaling by dividing the network into zones; and (c) routing decisions are taken either, based solely on topological information, or just considering a small subset of the many factors that impact the performance of a MANET.

This paper introduces CAROM (Context-Aware Routing over Ordered Meshes), a framework for integrated multicast and unicast routing that tackles many important problems of MANET environments in an integrated way. CAROM identifies regions of interest that are composed of nodes laying in shortest paths from sources to destinations as well as nodes that are located one hop away from them. Then, these regions of interest are ordered using hop distances and a cost metric obtained from context information that is collected by each node and disseminated along the regions of interest. The gradients established by this ordering can then be used to reach the destination from any node located inside of the region of interest following minimum cost shortest paths. The cost of using a node to reach a given destination is proportional to its degree of local mobility, its local congestion and to the length of its data queues. This cost is also inversely proportional to the number of feasible parents that the node has to reach the intended destination. Section III describes the algorithms and mechanisms used in CAROM to attain this operation.

Section IV describes the results of simulation experiments used to study the performance of CAROM and compare it with that of relevant multicast and unicast routing protocols for MANETs. Our comparison addresses the performance of the protocols purely for multicast routing, purely for unicast routing, as well as their performance in supporting unicast and multicast routing. We compare CAROM with ODMRP [7] to determine CAROM's effectiveness as just a multicast routing protocol, and consider different numbers of sources, and the use of group and random waypoint mobility models. For the case of unicast traffic, we compare CAROM against OLSR [3] and AODV [10] when increasing the average inter arrival time of unicast flows whose length and inter arrival time are exponentially distributed. We also compare CAROM against the combined use of AODV and ODMRP, and the use of OLSR and ODMRP in a scenario with both unicast and multicast traffic. The results show that CAROM provides substantial performance improvements over the traditional approach to supporting unicast and multicast routing. CAROM attains similar or better delivery ratios and lower delays and communication overhead than the traditional approaches.

II. RELATED WORK

There have been a large number of routing protocols proposed to date for MANETs, and we can only address a very small sample of them in this paper due to space limitations.

Unicast routing protocols for MANETs are typically classified into proactive and on-demand. Proactive routing protocols maintain routing information for all destinations independently of the interest in them, i.e., regardless of the unicast flows in the network. There have been proposals based on distance (e.g., WRP [9]) or link-state information (e.g., OLSR [3]), and many approaches to reduce the amount of overhead incurred in disseminating routing information proactively. The power-aware routing metrics proposed at [13] are intended for proactive protocols. On-demand routing protocols (e.g., AODV [10], DSR [5]) maintain routes for only those destinations for which there is interest, which makes them attractive when not all the destinations are very popular. AODV-MR [14] is an extension to ADOV that augments Join Query packets with an interference-related metric that is used to establish reduced inter-flow and intra-flow interference paths. DLAR [6] is similar to AODV-MR. Join Query packets are piggybacked with information regarding the length of the nodes' data queues. The destination then chooses the least loaded route and sends a Route Reply packet back to the source via the selected route.

Energy Aware Routing [12] is an on-demand routing protocol for sensor networks that computes low energy paths from sources and destinations and then performs load balancing among these paths in a probabilistic fashion. Similar to CAROM, Energy Aware Routing implicitly defines regions of interest. However, to establish its regions of interest, Energy Aware Routing assumes that location information is available to the nodes in the network.

Multicast routing protocols can also be classified on proactive and on-demand or by the type of routing structure they construct and maintain; namely, tree-based and mesh-based protocols. A tree-based multicast routing protocol constructs and maintains either a shared multicast routing tree or multiple multicast trees (one per each sender) to deliver packets from sources to receivers. Several tree-based multicast routing protocols have been reported (e.g., MAODV [11], ADMR [4]). However, in the context of MANETs, establishing and maintaining a tree or a set of trees in the presence of frequent topology changes incurs substantial exchange of control messages. A simpler and more robust approach to maintaining trees is to maintain a routing mesh consisting of a connected sub-graph of the network containing all receivers of a particular group and the relays needed to maintain connectivity. Three basic approaches of mesh-based multicast routing are characterized by the On-Demand Multicast Routing Protocol (ODMRP) [7], the Core Assisted Mesh Protocol (CAMP) [1], and the Protocol for Unified Multicasting through Announcements (PUMA) [15]. Hydra [8] is a sender-initiated multicast protocol that can work on either tree or mesh mode and that has the advantage of defining regions of interest that limit the dissemination

of control information. Despite of using regions of interest, Hydra's core nodes still flood the whole network.

III. CAROM: CONTEXT-AWARE ROUTING OVER ORDERED MESHES

A. Overview

CAROM establishes a region of interest for each active destination. An *active destination* is either a unicast destination with at least one active source or a multicast destination with at least one multicast group member and one active source. In general, we conceptualize a destination D as a connected *destination mesh* containing one or more nodes. In the case of a unicast destination, D is a singleton that contains a node with identifier D , whereas in the case of a multicast destination, D contains the members of the multicast group as well as a set of nodes needed to keep D connected. We refer to this set of nodes used to maintain D connected as the multicast mesh of D (MM_D). A destination's *region of interest* is a connected component of the network composed of all the shortest paths from active sources to the destination plus all those nodes located one hop away from them.

Once a region of interest is established, it is used to restrict the dissemination of control information, so that signaling packets are not disseminated outside of it. Then, the region of interest is ordered using sequence numbers, hop distances to the destination and a cost metric that is a function of the current state of the network. Once the ordering is established, data packets can be routed from sources to destination using meshes composed of minimum-cost shortest paths.

The cost of using a node to reach a given destination is proportional to its degree of local mobility which is a function of how dynamic its one-hop neighborhood has been, its degree of local congestion and to the length of its data queues. This cost is also inversely proportional to the number of feasible parents (i.e., nodes with strictly shorter distance) that the node has to reach the intended destination. Our framework can be easily extended to include more metrics, such as energy, that may be relevant to specific applications.

B. Mesh Activation and Deactivation

In CAROM, meshes are activated based on interest using mesh-activation requests (MR) packets that are disseminated across the network until they reach the destination's region of interest or an intended receiver. MR packets establish an ordering of the nodes in the network that is used to route mesh announcement (MA) packets back to the sources. Upon reception of a MR, a previously inactive receiver starts the routing mesh creation and maintenance process. In the case of a multicast destination, the reception of a MR also triggers a distributed core election that is held among multicast members.

A MR states the type of message, an application-defined horizon threshold that is used to define the scope of the dissemination of the MR, the sender, a Time-to-Live (TTL), the intended unicast destination or multicast group, a multicast-specific flag (*rvsited*) that indicates if the MR has been

relayed by a group member and, a sequence number that is used as identifier for the message.

Upon reception of a MR, a node determines if it is an intended destination of the MR. If it is not, it scans for a hit in a data cache storing tuples of the form $(sender\ id, packet\ id)$. If the $(sender\ id, packet\ id)$ -pair is already in the cache, or if the horizon value is reached, the MR is not forwarded any further. Intermediate nodes also store their distances to the source of the MR which is computed from the value of the TTL field. When the MR reaches an intended destination, the latter must start advertising its presence using mesh announcements (MA). If the destination is a multicast group, the receivers of the group that received a MR with the *visited* flag not set participate in a distributed core election. The details of the core election process are described in Section III-I.

Cores and unicast destinations disseminate MAs with newer sequence numbers every mesh-announcement period (MA-period), unless they stop receiving data packets for two consecutive MA-periods. The soft state regarding the routing structures is timed out and deleted if neither MAs nor data packets are received in three consecutive MA-periods.

C. Mesh Establishment and Maintenance

As we have mentioned, once a destination becomes active by receiving MRs stating persistent interest from at least one source, it starts advertising its existence periodically using mesh announcements (MAs).

A *mesh announcement* (MA_D^{*B}) transmitted by node B for destination D is a seven-tuple of the form:

$$(id^{*B}, core^{*B}, sn_D^{*B}, d_D^{*B}, cost_D^{*B}, mm_D^{*B}, next_D^{*B})$$

where id^{*B} is the identifier of B , $core_D^{*B}$ is either the identifier of the core of the multicast group D known by B , or the identifier of the unicast destination. sn_D^{*B} is the largest sequence number known by B of destination D , d_D^{*B} is the distance of B to the core of D or to the destination itself in the case of a unicast destination. $cost_D^{*B}$ is the cost to reach the destination D through B , mm_D^{*B} is a multicast-specific flag that indicates if B is a mesh member, a multicast group member, both, or a regular node. In the case of a unicast flow, the eight bits that store the membership code are used to indicate the node's longest known distance (ds_D^x) to an active source for the unicast destination. These distances were established by a previous dissemination of a MR and are used to route MAs back to sources that are not already included in the flow's region of interest. $next_D^{*B}$ is the identifier of B 's preferred next hop towards the core of D . Please note, that from our definition of destination we can infer that the core of a unicast destination is always the destination itself (it is the only element of the singleton).

For a given destination D , nodes maintain a neighborhood list L_D which stores an ordered set composed of the MAs that the node has recently received from each of its neighbors regarding that destination. In our notation, and to differentiate from an announcement that has just been received, an an-

nouncement received from neighbor B that is already stored in L_D is denoted as MA_D^B (with the $*$ dropped).

The announcements stored in L_D are also augmented with a time stamp (ts) obtained from the local clock and are ordered using a strict total order relation $\prec$ which is defined as follows:

$$\begin{aligned} MA_D^B \prec MA_D^A \Leftrightarrow & (sn_D^B < sn_D^A) \vee (sn_D^B = sn_D^A \wedge d_D^B > d_D^A) \\ & \vee (sn_D^B = sn_D^A \wedge d_D^B = d_D^A \wedge cost_D^B > cost_D^A) \vee \\ & (sn_D^B = sn_D^A \wedge d_D^B = d_D^A \wedge cost_D^B = cost_D^A \wedge id^B < id^A) \end{aligned} \quad (1)$$

The proof that $\prec$ is antireflexive, asymmetric and transitive is straightforward and comes from the fact that sequence numbers, distances, cost and node identifiers can be seen as natural or real numbers and that node identifiers are unique. The details are omitted for the sake of brevity.

In addition to L_D , a node x also keeps track of the core of the destination ($core_D^x$) which is dynamic in the case of a multicast destination, the largest sequence number known for the destination (sn_D^x), its current distance to the core of the destination (d_D^x), its *feasible distance* to the core of the destination (fd_D^x), its preferred next hop towards the core ($next_D^x$) and its mesh membership status flag mm_D^x that in the case of a unicast destination has no meaning. For unicast destinations nodes also keep track of the longest known distance to any source (ds_D^x). Nodes use the function fc_D (Eq. 8) to query their current cost to reach the destination. The initial value of the routing state is as follows: $L_D \leftarrow \emptyset$, $core_D^x \leftarrow next_D^x \leftarrow nil$, $sn_D^x \leftarrow 0$, $d_D^x \leftarrow fd_D^x \leftarrow \infty$, $ds_D^x \leftarrow \infty$ and $mm_D^x \leftarrow REG$ (regular node).

D. Processing Mesh Announcements

Upon reception of a MA_D^{*B} , node x updates its routing information according to the following formulas.

As it is shown in Eq. 2, node x accepts a MA if it contains a sequence number equal or larger than the current largest sequence number stored at node x , or if it is the first time that a MA is received from B . The MA is dropped otherwise.

$$L_D \leftarrow \begin{cases} L_D \cup \{MA_D^{*B}\} & \text{if } MA_D^B \notin L_D \\ L_D - \{MA_D^B\} \cup \{MA_D^{*B}\} & \text{if } sn_D^x \leq sn_D^{*B} \\ L_D & \text{if } sn_D^x > sn_D^{*B} \end{cases} \quad (2)$$

The feasible distance to the core is a non-increasing function that can only be reset by a change of core or by a new sequence number (Eq. 3). Feasible distances are used to select a *feasible set* of next hops towards the core of the destination.

$$fd_D^x \leftarrow \begin{cases} d_D^{*B} & \text{if } sn_D^{*B} > sn_D^x \\ \min\{fd_D^A, d_D^{*B}\} & \text{if } sn_D^{*B} = sn_D^x \\ fd_D^x & \text{otherwise} \end{cases} \quad (3)$$

The sequence number stored at node x for the core of destination D (sn_D^x) is a strictly increasing function that can only be reset by a change of core (Eq. 4). The max function of Eq. 4 is the traditional max function defined over the natural numbers. CAROM uses sequence numbers consisting of a core-specific time stamp taken from the real-time clock of

the node, and an unsigned monotonically increasing counter. When the counter reaches its maximum value, the node uses a new time stamp in its sequence number and resets the counter to zero. We assume that the real-time clock of a node is not reset on reboot or adjusted for daylight savings.

$$sn_D^x \leftarrow \max\{sn_D^x, sn_D^{*B}\} \quad (4)$$

The distance to the core of destination D of node x (d_D^x) is computed using Eq. 5 and the relation $\prec$ defined in Eq. 1. By definition, the core of the group has a 0 distance to itself and its feasible distance is always 0.

$$d_D^x \leftarrow \begin{cases} d_D^i + 1 : \max_{i \in L_D : sn_D^i = sn_D^x} \{i\} & \text{if such } i \text{ exists} \\ \infty & \text{otherwise} \end{cases} \quad (5)$$

The address of the next hop to the core of D ($next_D^x$) is also computed using the relation $\prec$ defined in Eq. 1, the current values of the feasible distance and sequence number:

$$next_D^x \leftarrow \begin{cases} id^i : \max_{i \in F_D^x} \{i\} & \text{if such } i \text{ exists} \\ nil & \text{otherwise} \end{cases} \quad (6)$$

where $F_D^x = \{i : i \in L_D \wedge fd_D^x = d_D^i \wedge sn_D^i = sn_D^x\}$ is the set of x 's feasible neighbors for destination D . Please note, that Eq. 6 selects from the set of the feasible next hops the one that reports the lowest cost to reach the destination. Lastly, if a node x receives a MA advertising a core with a larger identifier ($core_D^{*B} > core_D^x$) then L_D is set to $\{MA_D^{*B}\}$, $core_D^x$ is set to $core_D^{*B}$ and the other parameters are set as follows: fd_D^x to d_D^{*B} , d_D^x to $d_D^{*B} + 1$, sn_D^x to sn_D^{*B} , and $next_D^x$ to id^{*B} . Otherwise, if $core_D^{*B} < core_D^x$, then the MA is simply discarded.

The mesh membership flag $mm_D^x \in \{RM, MM, RCV, REG, NIL\}$ indicates whether x is a regular node (REG), a group receiver (RCV), a mesh member (MM) or both group receiver and mesh member (RM).

A node x is a mesh member if and only if:

$$\exists y \in L_D : mm_D^y \neq REG \wedge mm_D^y \neq NIL \wedge d_D^y > d_D^x \wedge next_D^y \leq id^x \wedge ts_D^y \geq ct + MA_period \quad (7)$$

where ts_D^y is the time stamp added to y when it was stored in L_D , ct is the current value of x 's clock and MA_period is the value of the MA-period.

E. Transmission of Mesh Announcements

Nodes transmit MAs to inform other nodes about updates in their routing state. These updates can be originated by such internal events as a change in the group membership status (a node joining or leaving a multicast group) that modify the value of mm_D^x , and the generation of a new sequence number in the case of the core; or by the reception of MA_D^{*B} from a neighbor B . Therefore, whenever the core of the destination generates a new MA with a larger sequence number, the latter is disseminated along the network advertising the new sequence number (Eq. 4), computing new cost values and establishing next hop pointers towards the core ($core_D$) (Eq.

6). The mesh composed of these next hop pointers from a given source to the $core_D$ is denominated as that source's *routing mesh*.

In the case of a multicast destination, a MA transmitted by a multicast group member R , forces R 's next hop (n) to update its mesh membership status according to Eq. 7. If this changes the value of mm_D^n , then n will have to transmit a new MA to advertise its new state. This way, nodes that lay in paths $p = R, n, n_1, \dots, n_k, core$ with $next_D^R = n, next_D^n = n_1, \dots, next_D^{n_k} = core_D$ are forced to become multicast mesh members, creating a connected component that contains all the receivers of a multicast group and that we have denominated as the *multicast destination*. The set of nodes $M^D = \{y : mm_D^y = MR \vee mm_D^y = MM\}$ form the *multicast mesh* of the multicast destination D .

To augment the reliability of the routing and multicast meshes, nodes can force more than one next hop (if available) to become mesh members or to route packets to the destination. For instance, to force two next hops to join to the multicast mesh, Eq. 6 has to be modified, so that it returns the second to the maximum $i \in L_G \mid i \in L_D : fd_D^i = d_D^i \wedge sn_D^i = sn_D^x$.

F. Context Awareness

We define the *context* of a node x with respect to destination D as a time varying n-dimensional vector $C_D^x(t)$. One or more components of $C_D^x(t)$ are time varying functions that depend on the local conditions of x . These conditions can be either *internal* (e.g., battery power, length of data queues) or *external* (e.g., local contention, relative mobility). Some components can be destination and topology dependant (e.g., number of feasible next hops to D). A subset of the components of C_D^x can also be constants that represent static properties of the node such as processing power or storage capacity.

For our experiments, we define $C_D^x(t)$ as a 4-dimensional vector $(\rho_n, \mu_n, f_s(\varphi_n), \phi)$ where ρ_n is the current *level of local contention*, μ_n is the current *level of relative mobility*, φ_n is the average queue length, and ϕ is inversely proportional to the cardinality of the set F_D^x of x 's feasible neighbors for destination D as defined in Section III-D

To measure local contention, we use a simple and intuitive metric based on the proportion of time in which the channel is perceived as busy. First, we define *instantaneous local contention* c as the ratio t_b/S_p , where t_b is the amount of time the channel was perceived busy during a sampling period of S_p seconds. Then, we compute an exponential weighted moving average and we get $\rho_n = (1 - \beta)\rho_{n-1} + \beta c$ where ρ_n is the current *level of local contention*, β is a constant used to assign weight to the level of local contention calculated at the previous sampling period (ρ_{n-1}) and, the current instantaneous local contention (c).

The level of relative mobility μ_n is computed in a similar way. Each node keeps track of how its one-hop neighborhood has changed between two consecutive sampling periods, then, nodes compute an exponential weighted moving average. We define instantaneous relative mobility m as $[d/(r + d)]$,

where d is the number of new or missing one-hop neighbors detected during the current sampling period with respect to the neighbors detected in the previous sampling period, and r is the number of neighbors that did not change from the previous sampling period with respect to the current sampling period. The *level of relative mobility* during sampling period n is $\mu_n = (1 - \beta)\mu_{n-1} + \beta m$. Where β is as in the case of the level of local contention.

To compute the average queue length φ_n we use the same approach as the previous two metrics, however, since the values taken by φ_n can be highly variable and unbounded, we use a smoothing function $f_s(\varphi_n)$ defined as $1 - e^{-0.02\varphi_n}$. Lastly, ϕ is computed as $1/|F_D^x|$.

The current value for β is 0.2. However, a sensitivity analysis showed that the performance of CAROM is not very sensitive to this parameter.

Lastly, as it is shown in Eq. 8, fc_D^x is computed as a convex combination of $C_D^x(t)$ plus the minimum cost reported by a feasible next hop to D .

$$fc_D^x = \begin{cases} c_1\rho_n + c_2\mu_n + c_3f_s(\varphi_n) + c_4\phi + cost_D^i & \text{if such } i \text{ exists} \\ \infty & \text{otherwise} \end{cases} \quad (8)$$

where the max function is computed using the relation $\prec$ defined in Eq. 1.

G. Packet Forwarding and Local Repairs

When a source has data to send, it first checks whether it has received a MA advertising the intended destination within the last three MA-periods. If not, it broadcasts a MR as described in Section III-B. Otherwise, the sender forwards the data packet according to its routing table.

Upon reception of a data packet, nodes first check for a hit in their data packet cache which stores the sender's address and sequence number of recently received data packets. If the (*sender's address, sequence number*) pair is already in the cache, the packet is silently dropped. Otherwise, the receiving node inserts the pair in its cache and determines whether it has to relay the data packet or not. The node also passes the packet to the upper layers if it is a receiver for the flow.

The following predicate is used by node x to decide if it has to forward a data packet with destination D received from neighbor y :

$$mm_D^x = RM \vee mm_D^x = MM \vee \exists y \in L_D : d_D^y > d_D^x \wedge next_D^y \leq id^x \quad (9)$$

Eq. 9 states that node x forwards a data packet received from node y if x is part of the multicast mesh or if x was selected by the previous relay (y) as one of its next hops to the core. This way, data packets travel along directed routing meshes until they reach either the first mesh member or the destination itself and then, in the case of a multicast destination, they are flooded along the multicast mesh. In the case of a unicast destination, the first two terms of Eq. 9 are always false; hence, nodes only

forward data packets if they are part of a source-destination shortest path.

Forwarders located in directed routing meshes employ the transmission of data packets by their next hops as implicit ACKs. If a node fails to receive three consecutive implicit ACKs from a neighbor, then it removes that node from the neighborhood list L_D and updates the value of $next_D^x$ using Eq. 6 and the value of d_D^x using Eq. 5.

As discussed in Section III-E, a change in any of these two values forces node x to transmit a new MA to advertise its new routing state and, for instance, force a newly selected next hop node to route data packets towards the destination. If after removing the neighbor from L_D , the value of $next_D^x$ equals *nil*, then the current feasible distance fd_D^x is included in the MA (instead of the new value of the distance d_D^x). MAs with $next_D^x = nil$ and $id^{*x} \neq core_D^{*x}$ are interpreted as a *neighbor request* which inform other nodes that x no longer has a route towards the destination. Upon reception of a neighbor request from x , nodes will eliminate x 's entry from L_D and update their routing state accordingly. Additionally, a node y that complies with the predicate:

$$next_D^y \neq nil \wedge (sn_D^y > sn_D^{*x} \vee sn_D^y = sn_D^{*x} \wedge d_D^y \leq d_D^{*x}) \quad (10)$$

also transmits a MA as a response to the neighbor request. This is because y is in fact a feasible next hop for x and can be used to reach the destination through a loop-free path.

H. Regions of Interest

A *region of interest* of a unicast or multicast flow is a connected component of the network that contains those nodes relevant to the dissemination of information for the flow, namely destinations, senders, relay nodes located in shortest paths connecting sources and destinations as well as nodes located one hop away from them. Because all the nodes in the region of interest are likely to participate in the flow, they take part proactively in the signaling needed to maintain routing information for the flow. By the same token, nodes located outside of the region of interest defined for a destination (unicast or multicast) do not participate in the process of routing data packets for that destination; hence, transmitting, receiving or storing MAs regarding that destination is simply overhead to them.

For the case of unicast flows, and since a unicast destination is a static singleton, nodes outside the region of interest of a unicast destination simply stop the propagation of such MAs. In contrast, a multicast destination is a dynamic set of nodes; furthermore, nodes may send to a multicast group without being part of the group. Accordingly, and to support a receiver-initiated method for multicast receivers to join groups and to let non-group members send data to multicast groups, the mesh of a multicast destination is not confined to its region of interest. Instead, all nodes in the network receive information about the existence of the core for a group that has been activated by MRs, i.e., all nodes receive MAs about active

multicast destinations. MAs are sent within a multicast region of interest with much higher frequency than outside of it. This frequency decreases exponentially with respect to the distance in hops from a node to the boundary of the region of interest. To achieve the latter behavior, nodes located outside of a region of interest of a multicast destination only relay every other MA they receive.

Regions of interest are established with the support of the data packets that flow from sources to destinations. A node considers itself part of the region of interest of destination D if during the last two mesh announcement periods: (1) it has generated, received or relayed a data packet with destination D , or (2) it has received a mesh announcement MA_D^{*B} such that $core_D^{*B} = B$ and $d_D^{*B} = 0$.

Fig. 1 presents an example of a region of interest of a destination D that has two active sources S_1 and S_2 . Nodes labeled m are part of the region of interest because they lay in shortest paths from a sender to the destination and have recently been used as relays. Nodes such as x and y are part of the region of interest because they are located one hop away from an active shortest path and have recently overheard a data packet addressed to D . Nodes like z , which are outside of the region of interest receive MAs every mesh-announcement period (MA-period), but they do not relay them any further if D is a unicast destination or relay every other MA if D is a multicast destination. Hence, in the case of a unicast destination, nodes like a' will not receive MAs and, in the case of a multicast destination, will receive MAs with half of the frequency and relay them with one quarter of the frequency with which they are generated by the core of D .

When a mesh request (MR) for destination D reaches a node x that is part of the region of interest of D , x has to reply with a MA. The state established by the MR (see Section III-B) is then used to route the MA generated by x towards the source that originated the MR. This way, the new source will acquire routes towards the destination and will be included into the region of interest. The state used to route MAs towards the source is short-lived and is deleted within a MA-period.

Fig. 1. Region of interest of destination D .

I. Core Election

Core elections are held only if the MR contains a multicast address. Upon reception of a MR, a group receiver determines

whether it has received a MA from the core of the multicast group within the last two MA-periods. If the node has, no further action in this regard is needed. Otherwise, the receiver checks the value of the *rvsited* flag and if it equals 0, then the receiver considers itself the core of the group and starts transmitting MAs stating itself as the core of the group. If the value of the flag equals 1, then the receiver does not participate as a contender in the core election.

As described in Section III-E, nodes propagate MAs based on the best announcements they receive from their neighbors. A MA with a higher core id is considered better than one with a lower core id. Therefore, if a node receives a MA advertising a core with a larger id than the current core, then the new core is adopted. On the other hand, if a MA advertising a core with a smaller id is received, then nodes check if they have recently broadcasted a MA with the current core, and if so, the MA is simply ignored. Otherwise, nodes send a MA that forces the neighbor with the smaller core to adopt the core with higher id. Eventually, each connected component has only one core.

IV. PERFORMANCE COMPARISON

We present simulation results comparing CAROM against ODMRP for the case of multicast traffic, and against AODV and OLSR for the case of unicast traffic as well as against AODV with ODMRP and OLSR with ODMRP for the case of combined unicast and multicast traffic. We use ODMRP, AODV, and OLSR in our experiments, because they are *de facto* baselines for performance comparisons of multicast and unicast routing protocols. We use packet delivery ratio, generalized group delivery ratio, generalized group end-to-end delay, end-to-end delay, path-length specific end-to-end delay, control overhead, and total overhead as our performance metrics. The generalized group delivery ratio is an extension of the group reliability metric introduced in [16], in which a packet is considered as delivered, if and only if it is received by a given proportion of the multicast group members. As with the previous metric, the generalized group end-to-end delay only considers packets that were received by at least a given proportion of multicast receivers. These two metrics emphasize the importance of group delivery by not considering packets that are received by a small subset of the group members. The control overhead is the average number of control packets generated by the routing protocols and the total overhead is the average number of packets that are actually transmitted by the physical layer. The remaining metrics are self explanatory.

The protocols are tested with IEEE 802.11 DCF as the underlying MAC protocol, and all signaling packets are sent in broadcast mode for the multicast protocols. We use random waypoint and group mobility [2] as our mobility models. The first model allows us to test the protocols on general situations in which each node moves independently, and the latter models situations in which the members of a team tend to move in groups. In the group mobility model, each group decides its group mobility direction and speed randomly. Each node then decides its internal mobility randomly and computes its actual mobility by summing the two mobility vectors [2]. We used the

TABLE I
SIMULATION ENVIRONMENT

Total nodes	100	Node placement	Random
Simulation area	$1700 \times 1700m^2$	Simulation time	150s
MAC Protocol	802.11	Tx. power	15dbm
		Tx. Rate	2000000bps
Data source	MCBR and CBR	Pkts. per src.	1000
Mobility model	Random waypoint	Pause time	10s
		Min.-Max. Vel.	1-10m/s
Mobility model	Group mobility	Node pause time	10s
		Grp. pause time	10s
Grp. Min-Max Vel.	1-10m/s	Node Min-Max Vel.	1-10m/s

discrete event simulator Qualnet [17] version 3.9, that provides a realistic simulation of the physical layer, and well tuned versions of ODMRP, AODV and OLSR. To have meaningful comparisons, all the multicast protocols use the same period of three seconds to refresh their routing structures (join query periods for ODMRP and announcement periods for CAROM). For ODMRP, the forwarding group timeout was set to three times the value of the join query (JQ) period, as advised by its designers. The value of CAROM's horizon threshold was set to the same value as the TTL used in the ODMRP's JQs, which is the worst-case scenario for propagation of MRs in CAROM. Table I lists the details of the simulation environment.

A. Multicast Traffic

We first focus on an experiment in which we have a single multicast group of 20 nodes and the number of concurrent active senders increases. Each sender transmits 1000 packets at a rate of 10 packets of 256 bytes per second. Sources are not group members. Figs. 2(a-e) present the results when nodes move according to the random waypoint (RWP) and group (Grp) mobility models. In the latter model the 20 nodes that belong to the multicast group move around inside of a mobile square region of $900 \times 900m^2$. The remaining 80 nodes (including the sources) move following the random waypoint mobility model. Fig. 2(a) shows the delivery ratio attained by the different protocols as the number of concurrent sources is increased. We observe that CAROM performs similar to or better than ODMRP for both mobility models and in particular for 12 or more sources. Fig. 2(b) presents the generalized group delivery ratio attained by the protocols when the delivery threshold is set to 80%. We notice that, when nodes move according to the group mobility model, CAROM achieves a performance improvement of up to 17%. As it can also be seen in Fig. 2(b), ODMRP does not benefit from the proximity among group members.

Fig. 2(c) shows that, as the number of sources is increased, the control overhead (CO) induced by CAROM remains constant for both mobility models. The latter result was expected because in CAROM a single node (the core) disseminates control information per multicast group whereas in ODMRP all the sources disseminate control information. Moreover, the regions of interest used in CAROM restrict the dissemination of control information to those nodes that are likely to participate in a multicast flow which further reduces the number of

control packets. The same figure also shows that the average total number of packets transmitted by nodes (TO) running CAROM is much less than that of ODMRP. We also observe that both protocols induce less overhead when nodes move according to the group mobility model. The reason is that, in general, routing meshes tend to be smaller when the group members are close by. Figs. 2(d and e) show that CAROM achieves end-to-end delays and group delays that are up to three orders of magnitude smaller than that of ODMRP. The increased delay shown by ODMRP is mainly due to the large amount of high priority control packets that are injected into the network and that tend to stall data packets (Fig. 2(c)). Please note that the delay graphs are plotted using logarithmic scale. These results also allow us to highlight the benefits of using context information to select those routes that present the most favorable conditions in terms of stability, congestion, load and robustness.

B. Unicast Traffic

In this scenario we evaluate the performance of CAROM as a unicast routing protocol. Unicast flows are generated following an exponential distribution with a varying mean interarrival time ($1/\lambda$) and a mean flow duration time of $1/\mu = 100$ seconds. The latter corresponds to 33.3% of the total simulation time which is 300 seconds. CBR flows are established among randomly selected nodes. CBR sources generate data packets of 256 bytes at a rate of 10 packets per second. In these experiments, all nodes move according to the random waypoint mobility.

The results are shown in Figs. 3(a-c)). From Fig. 3(a), we can observe that CAROM scales much better than AODV and OLSR. For $1/\lambda = 2.5s$, CAROM delivers as much as three times the packets delivered by AODV and 17% more packets than OLSR. The reason behind this behavior can be observed in Fig. 3(b) that presents the total and control overhead induced by the protocols. From the figure we notice that the control overhead induced by CAROM remains almost constant across the different values of $1/\lambda$ while that of AODV experiences a steep increase. CAROM is able to keep the control overhead constant due to the restricted flooding enforced by the regions of interest. Additionally, even when regions of interest confine the dissemination of control packets, they still allow many nodes to acquire routes towards the destinations. On the other hand, the continuous arrival of new flows forces AODV to constantly flood the whole network with route request packets. As $1/\lambda$ is decreased, more flows are generated and flooding becomes more costly because it tends to make the network congested. Under high congestion, many control packets are lost due to collisions. The latter is interpreted by AODV as broken links that need to be repaired and hence, nodes react to these packet lost by generating even more route request and route reply packets making the network even more congested. As can be observed in Fig. 3(b), the network reaches a saturation point where a large amount of the packets are dropped at the queues. An evidence of this, is the fact that the number of

Fig. 2. Performance with increasing number of MCBR sources.

control packets generated by the nodes is even larger than the number of packets that are actually transmitted.

The end-to-end delay attained by the protocols is presented in Fig. 3(c). The figure shows that the best performing protocol is AODV but with CAROM attaining delays that are always lower than one second.

C. Combined Multicast and Unicast Traffic

This set of experiments evaluates the performance of the routing protocols in a scenario with combined multicast and unicast traffic. We increase the number of concurrent active multicast groups from 1 to 6. Each multicast group is composed of 15 members and three multicast members of each group are randomly selected to act as MCBR sources. Group members move according to the group mobility model with a group area of $900 \times 900m^2$. This scenario also include 5 CBR flows between nodes that do not belong to a multicast group and that move following the random mobility model. Each CBR and MCBR source sends a total of 1000 data packets of 256 bytes at a rate of 10 packets per second. The results are shown in Figs. 4(a-f). From Fig. 4(a) we can observe that CAROM attains higher delivery ratios than the other protocols for both unicast and multicast traffic. CAROM delivers up to 24% more data packets than AODV and up to 32% more than OLSR, and at the same time, up to 20% more multicast data packets than ODMRP when it is used in conjunction with OLSR and up to 22% when it is coexisting with AODV. CAROM also attains higher generalized group delivery ratios than ODMRP for more than 1 group (Fig. 4(b)) and the lowest delays for both unicast and multicast traffic (Fig. 4(e)). CAROM provides up to three orders of magnitude improvement in group delay over that of ODMRP. All these

while incurring far less control (CO) and total overhead (TO) than the other protocols (Fig. 4(c)). Lastly, Fig. 4(d) shows that CAROM consistently outperforms AODV and OLSR by attaining lower delays for all path lengths.

V. CONCLUSIONS

We introduced CAROM, an integrated routing framework for MANETs that employs regions of interest and end-to-end context information to support multicast and unicast data flows. The context of a node for a given destination is composed of local and end-to-end metrics that are encoded in a single cost metric using a simple convex combination. This information is then disseminated from destinations to sources with the objective of identifying the less congested, more reliable and stable shortest paths that can be used to connect sources to destinations. Regions of interest are further used to restrict the dissemination of context information to those nodes that are likely to participate in the data flow. The results of a series of simulation experiments illustrate that CAROM scales much better than traditional unicast and multicast protocol such as AODV, OLSR and ODMRP. The latter is particularly apparent in scenarios that combine unicast and multicast data flows where CAROM provides up to three orders of magnitude improvement in delay and significantly higher packet delivery while inducing ten times less control overhead.

REFERENCES

- [1] J. J. Garcia-Luna-Aceves and E. L. Madruga. The core-assisted mesh protocol. *IEEE Journal on Selected Areas in Communications*, 17(8):1380–1394, Aug 1999.

Fig. 3. Only Unicast Traffic: Performance with decreasing interarrival time.

Fig. 4. Performance with increasing number of active groups and five unicast flows.

- [2] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang. A group mobility model for ad hoc wireless networks. In *Proc. of ACM/IEEE MSWiM'99*, pages 53–60, 1999.
- [3] P. Jacquet, A. Laouiti, P. Minet, and L. Viennot. Performance analysis of OLSR multipoint relay flooding in two ad hoc wireless network models. In *The second IFIP-TC6 NETWORKING Conference*, may 2002.
- [4] J. G. Jetcheva and D. B. Johnson. Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks. In *MobiHoc '01: Proc. of the 2nd ACM intl. symp. on Mob. ad hoc net. & comp.*, pages 33–44. ACM, 2001.
- [5] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, 353:153–181, 1996.
- [6] S.-J. Lee and M. Gerla. Dynamic load-aware routing in ad hoc networks. In *Proc. of IEEE ICC 2001.*, volume 10, pages 3206–3210 vol.10, 2001.
- [7] S.-J. Lee, M. Gerla, and C.-C. Chiang. On-demand multicast routing protocol. In *Proc. of the IEEE Wireless Comm. and Net. Conf., 1999. WNCN.*, pages 1298–1302 vol.3, 1999.
- [8] R. Menchaca-Mendez and J. J. Garcia-Luna-Aceves. Scalable multicast routing in manets using sender-initiated multicast meshes. In *Proc. of IEEE MASS 2008*, pages 1–12, Oct. 2008.
- [9] S. Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mob. Netw. Appl.*, 1(2):183–197, 1996.
- [10] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *Proc. of the Second IEEE Workshop on Mob. Comp. Syst. and App., 1999. WMCSA '99.*, pages 90–100, Feb 1999.
- [11] E. M. Royer and C. E. Perkins. Multicast operation of the ad-hoc on-demand distance vector routing protocol. In *MobiCom '99: Proc. of the 5th annual ACM/IEEE intl. conf. on Mob. comp. and net.*, pages 207–218. ACM, 1999.
- [12] R. C. Shah and J. M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *Proc. of IEEE WNCN'2002*, volume 1, pages 350–355, Mar 2002.
- [13] S. Singh, M. Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proc. of ACM/IEEE MobiCom '98*, pages 181–190. ACM, 1998.
- [14] A. P. Subramanian, M. M. Buddhikot, and S. Miller. Interference aware routing in multi-radio wireless mesh networks. In *Proc. of IEEE WiMesh 2006.*, pages 55–63, Sept. 2006.
- [15] R. Vaishampayan and J. J. Garcia-Luna-Aceves. Efficient and robust multicast routing in mobile ad hoc networks. In *Proc. of the IEEE Conf. on Mob. Ad-hoc and Sensor Syst., 2004*, pages 304–313, Oct. 2004.
- [16] K. Viswanath, K. Obraczka, and G. Tsudik. Exploring mesh and tree-based multicast routing protocols for manets. *IEEE Transactions on Mobile Computing*, 5(1):28–42, 2006.
- [17] Qualnet 3.9, scalable network technologies: <http://www.scalablenetworks.com>.