UC Santa Cruz

UC Santa Cruz Previously Published Works

Title

A Cross-Layer Framework to Support Real-Time and Elastic traffic in MANETs

Permalink

<https://escholarship.org/uc/item/40g0z8tx>

Author

Garcia-Luna-Aceves, J.J.

Publication Date

2011-03-28

Peer reviewed

A cross-layer framework to support real-time and elastic traffic in MANETs

J.J. Garcia-Luna-Aceves

Computer Engineering Department
University of California, Santa Cruz

Rolando Menchaca-Mendez

Computer Research Center
Mexican National Polytechnic Institute

Abstract—A Framework for Integrated Routing, Scheduling and Traffic Management (FIRST) in Ad Hoc Networks is introduced. Unicast and multicast routes are established in coordination with the scheduling of transmissions and bandwidth reservations in a way that bandwidth and delay guarantees can be enforced on a per-hop and end-to-end basis. Results from detailed simulation experiments show that, compared to a protocol stack consisting of 802.11 DCF for channel access, AODV or OLSR for unicast routing, and ODMRP for multicast routing, FIRST attains similar or better performance for elastic traffic, and up to two orders of magnitude improvement in end-to-end delays, with twice the amount of data delivery for real-time traffic.

I. INTRODUCTION

The price, performance, and form factors of processors, radios and storage elements are such that wireless ad hoc networks have become a reality. In theory, they constitute an ideal vehicle to support disaster-relief and battlefield operations, emergency search and rescue missions, and many other distributed applications on the move. However, in practice, these applications cannot be supported effectively in these networks today. To a large extent, this is due to the fact that the protocol architectures used in wireless ad hoc networks, and mobile ad hoc networks (MANET) in particular, are derivatives of the protocol-stack architectures developed for wired networks and the Internet. The stark differences between wireless and wired networks call for a cross-layer approach to managing network resources in support of information dissemination. Section II provides a small sample of the considerable prior work focusing on cross-layer approaches to routing and transmission scheduling. The solutions that have been proposed in the past are either based on centralized algorithms requiring too much information at each node, or do not integrate routing and scheduling with the establishment of bandwidth reservations and traffic management. Furthermore, prior solutions do not address the integration of multicast routing with transmission scheduling.

Section III presents an overview of FIRST (Framework for Integrated Routing, Scheduling and Traffic Management). Section IV describes the channel structure assumed in FIRST and the priority-based queuing system to handle signaling traffic, elastic data flows and real-time flows. Sections V and VI describe the neighbor protocol and distributed transmission scheduling used in FIRST, which provides delay guarantees. This scheduling is coupled with a transaction-oriented end-to-end reservation scheme for time slots, which is driven

by the establishment of routes to provide end-to-end delay guarantees. Section VII presents the interest-driven approach used in FIRST for unicast and multicast routing. Nodes maintain flow-ordered routing meshes consisting of multiple paths from sources to destinations over which relays are capable of establishing channel reservations that meet the end-to-end requirements of the flows being routed.

Section VIII describes the results of detailed simulation experiments used to study the performance of FIRST and compare it with the performance of a traditional MANET protocol stack, which consists of the IEEE 802.11 DCF for channel access working independently of the routing protocols used for unicasting (AODV, OLSR) and multicasting (ODMRP).

II. RELATED WORK

Due to space limitations we present only a very small representative sample of prior cross-layering approaches attempting to make routing and channel access more efficient in ad hoc networks. Chen and Heinzelman [1] provide a comprehensive survey on routing protocols that provide some sort of support for QoS in MANETs, and Melodia et. al. [2] present a survey of cross-layer protocols for wireless sensor networks.

MACA/PR (multiple access collision avoidance with piggyback reservations) [3] was one of the first approaches attempting to integrate channel access, routing and traffic management. MACA/PR, which supports only unicast traffic, extends the IEEE 802.11 DCF to incorporate a bandwidth reservation mechanism and includes a modified version of DSDV [4] that keeps track of the bandwidth of the shortest paths to each destination and the maximum bandwidth available over all possible paths. The first data packet of a real-time flow makes reservations along the path for subsequent packets in the connection. One-hop scheduling information is piggybacked in data packets and ACKs which reserve time-synchronized windows at specified time intervals. Reservations are made taking into account only two-hop neighborhood information and without coordination with the routing protocol.

In the context of multicast communication, most of the work has focused on static networks (i.e., [5] [6]). In these works, the authors formulate the joint multicast routing and power control problem [6] or the network planning problem [5] as a cross-layer optimization problem. However, no proposals have been made on the integration of scheduling and routing for many-to-many communication.

III. FIRST

FIRST assumes that nodes share a single wireless channel organized into time frames consisting of a fixed number of time slots, and that nodes can transmit as many packets as the length of the time slot allows. The objective in FIRST is to orchestrate the scheduling, routing and traffic management functions of a MANET in a way that sources and destinations of flows perceive the network as a virtual link dedicated to the dissemination of those flows.

Nodes use a priority-based queueing system composed of queues for signaling traffic, elastic (non real-time) traffic and real-time (RT) traffic. A RT queue is created for every new RT flow traversing a node, and it is associated with the time slot(s) reserved for the flow. This queueing system exerts traffic management to expedite the exchange of control packets needed for channel access, routing, and reservations; avoid multiple access interference of RT flows so that they can meet their end-to-end deadlines; and attain high throughput and delivery rates for elastic data flows.

Nodes use a neighbor protocol to exchange neighborhood information regarding the identifiers of neighbor nodes and the reservations they have made, and this information is used for scheduling and reservations.

Accessing the time slots of each frame is based on a combination of distributed elections of available time slots and reservations of time slots. For those time slots that have not been reserved, nodes use a distributed election algorithm based on hashing functions of node identifiers; however, in contrast to prior similar schemes, these elections are such that nodes have channel-access delay guarantees. To support RT data flows, reservations are made on time slots selected by the routing algorithm so that the channel access times of all nodes in the paths from sources to destinations of a RT flow are ordered in time and meet the end-to-end delay requirement of the flow.

The routing algorithm in FIRST establishes and maintains loop-free *flow-ordered routing meshes* from sources to unicast or multicast destinations. A flow-ordered routing mesh is a connected component of the network formed with nodes that have the resources needed to establish a channel access schedule that provides end-to-end bandwidth and delay guarantees to data flows. These routing meshes also have the further advantage of equipping FIRST with a fast and efficient way of repairing routes because they contain extra paths that are ready to be used in case of a link break. This helps diminishing the impact of node mobility over the quality of service perceived by RT flows.

IV. CHANNEL STRUCTURE AND TRAFFIC MANAGEMENT

Nodes share the same frequency band, and we assume that clock synchronization among all the nodes in the network is achieved through a multihop time synchronization scheme such as the one implemented in Soft-TDMAC [7] which is a TDMA based MAC protocol that runs over commodity 802.11 hardware. Nodes access the common channel assuming that it

is organized using a time-division multiple access structure, which we call FIRST frame. Each FIRST frame is composed of N time slots (from slot 0 to slot $N-1$). We use the position of a slot within the FIRST frame as the identifier of the slot. A FIRST frame does not have any particular structure and any time slot can be used to transmit a sequence of packets (signaling or data). There is only one special-propose time slot used to admit new nodes to the network. These admission slots occur every A time slots, with $A \gg N$, and are used by nodes to transmit their first hello packet on a contention basis. When a node is allowed to transmit over a time slot, it fits as many packets as possible in it, and selects those packets from its local transmission queues using a strict priority scheduler. All queues are FIFO and are served using a priority-based algorithm.

Reservation packets have the highest priority (p_{Rsv}), the next priority is given to network-layer signaling packets (p_{ctr}), and data packets waiting in data queues have the lowest priority. Data queues can be either elastic or RT, and RT queues are assigned higher priority (p_{RT}) than the priority given to elastic queues ($p_{elastic}$). The priority of a RT queue created for flow f is increased from p_{RT} to p_{RT+} if the current time slot t was reserved on behalf of flow f . Hello packets are transmitted with the lowest priority ($p_{Hello-} < p_{elastic}$) if more than $hello_period/2$ seconds but less than $hello_period$ seconds have elapsed since the last time a hello packet was transmitted, because there is no need for the information yet. However, if more than $hello_period$ seconds have elapsed, then the priority of the hello packet is set to $p_{Hello+} > p_{ctr}$.

To summarize, during a time slot allocated to a node, the relationships among traffic priorities is:

$$p_{Hello-} < p_{elastic} < p_{RT} < p_{RT+} < p_{ctr} < p_{Hello+} < p_{Rsv}$$

V. NEIGHBOR PROTOCOL

Reservations and transmission scheduling in FIRST use distributed algorithms that require each node to know the nodes within its two-hop neighborhood. We denote $N(u) = \{v : (u, v) \in E\}$ as the one-hop neighborhood of u and $N(N(u))$ as the two-hop neighborhood of u .

To gather two-hop neighborhood information, each node transmits a *hello* message periodically each $hello_period$ seconds that contains a list of tuples for the node itself and for each of its one-hop neighbors. Each tuple is composed of a node identifier, a list of the identifiers of the node's reserved slots and the length of the list of reserved slots. Nodes store the last hello message received by each one-hop neighbor in their *neighbor lists*. If no hello message is received from a given neighbor in three consecutive hello periods, then the neighbor is deleted from the neighbor list.

VI. TRANSMISSION SCHEDULING

Channel access algorithm used in FIRST consists of three simple ways to determine which node should transmit in a time slot. On every slot t , node u with identifier id^u first checks if it is the owner of the slot (i.e., if $(t + id^u) \bmod N = 0$) and if so, u can access the channel. If node u does not own the slot,

it checks if the owner is present in its two-hop neighborhood (i.e., if there is $v \in N(N(u))$ such that $(t + id^v) \bmod N = 0$). If this is the case, then node u listens to the channel.

If there is a collision of the mod operation, the time slot is not considered as owned by any of the contending nodes and a hash-based election is held among the nodes participating in the collision. If the owner of the time slot is not present in the two-hop neighborhood, node u checks if it has a reservation on the slot $(t \bmod N)$. If the node has, it can access the channel. Otherwise, node u checks if there is $v \in N(N(u))$ such that v has reserved the slot $(t \bmod N)$. If that node exists, node u listens to the channel.

If none of the two previous conditions are met, node u employs a hash-based election scheme [8] to select the node that can access the network. Node u computes the priority of each node v in its two-hop neighborhood as $p_v^t = \text{Hash}(id^v \oplus t) \oplus id^v$ where $\oplus$ is the concatenation operator and **Hash** is a hash function with good uniformity properties. Node u can access the channel if $p_u^t > p_v^t$ for any node v in its two-hop neighborhood. Otherwise, node u listens to the channel.

The rationale for this approach is to attain the simplicity of prior schemes based on hash-based elections (e.g., NAMA [8]), while alleviating the “coupon collector’s” problem [9] that arises in probabilistic hash-based elections.

A. Channel Reservations

When a node becomes part of a persistent RT data flow, it uses the reservation protocol to reserve future slots to be used on behalf of that particular RT flow. Unlike many prior channel access schemes, the selection of a particular slot $r \in [0, N - 1]$ in the FIRST frame is coordinated by the routing layer so that the channel access schedules of the relays of the flow are *flow ordered*. The relays of a flow are *flow ordered* if every single one of them can access the channel in a time ordered sequence of slots. To achieve this behavior, and as it is shown in Eq. 1, a node x that is relaying data packets towards destination D employs a random number $slot_D \in [0, N - 1]$ generated and disseminated by D as a reference to compute the interval of flow ordered slot identifiers corresponding to its current distance to D (d_D^x).

$$[(slot_D - d_D^x \Delta) \bmod N, ((slot_D - d_D^x \Delta) \bmod N + \Delta) \bmod N] \quad (1)$$

By reserving slots from these intervals, any two consecutive nodes n_i and n_{i+1} (with distances to destination D : $d_D^{n_i} + 1 = d_D^{n_{i+1}}$) that lay in a successor path $p = \{s, n_1, n_2, \dots, D\}$ established by the routing algorithm will have the right to access the channel within a maximum time of $2\Delta\tau$ seconds. Where Δ is the size in number of slots of the interval defined by Eq. 1 and τ is the length in seconds of each time slot.

In addition to the reservation, a node that becomes part of a RT flow has to create a new RT queue which is associated with the flow and with the set of slots reserved on behalf of that flow. During a reserved time slot, the priority of this particular queue is raised to p_{RT+} , which is higher than the priority of

the remaining data queues (RT or elastic). Hence, no packet from a different flow is transmitted during these slots unless the queue under consideration is empty.

As it is described in Section VII, the successor paths employed to transport RT flows are established using nodes that have reported to have available slots located at the adequate positions in the FIRST frame. This way, FIRST increments the probability that these nodes successfully reserve the adequate slots in the FIRST frame.

The process of reserving a time slot begins by finding a suitable candidate slot in the interval defined by Eq. 1. A suitable candidate slot for node x is a slot not currently owned or reserved by any node in its two-hop neighborhood, including x itself, and that is not in the process of being reserved. Once a candidate time slot $slot_c$ is identified, a node with identifier id^x transmits a Reservation Request $RsR^x = (id^x, slot_c)$ to its neighbors and waits for $N\tau$ seconds to collect the replies from them. If all the nodes $v \in N(x)$ reply with a Reservation Granted $RsG = (id^x, slot_c, id^v)$ granting the reservation, and neither $RsR = (id^y, slot_c)$ such that $id^y > id^x$, nor $RsG = (id^y, slot_c, id^w)$ such that $id^y > id^x$ are received; then node x considers the slot $slot_c$ as reserved by itself.

Upon reception of a $RsR = (id^x, slot_c)$, a node with identifier id^u replies with a $RsG = (id^x, slot_c, id^u)$ granting the reservation if none of the following conditions are true: the owner of slot $slot_c$ is present in the one-hop neighborhood of node u ($\nexists v \in N(u) : (slot_c + id^v) \bmod N = 0$) or a node in u ’s one-hop neighborhood holds a reservation over $slot_c$ or there is a concurrent request owned by a node with identifier id^y such that $id^y > id^x$. Otherwise, node u replies with a Reservation Denied $RsD = (id^x, slot_c, id^u)$.

Nodes refresh their reservations by means of hello messages for as long as they are part of the corresponding flows.

VII. LOOP-FREE INTEREST-DRIVEN ROUTING AND END-TO-END SCHEDULING

A. Meshes

Routing in FIRST is based on *destination meshes* and *routing meshes*. A destination D is a connected *destination mesh* containing one or more nodes. In the case of a unicast data flow, D is a singleton that contains a node with identifier D , whereas in the case of a multicast data flow, D contains the members of a 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 routing mesh is used to forward data packets of a flow from sources to destinations (unicast or multicast). It is composed of feasible paths from sources to destinations with the restriction that every single node in those paths has to be *flow ordered*. As discussed in Section VI-A, a node is flow ordered for a given RT flow if the node can reserve a time slot (or set of time slots) at the appropriate position in the FIRST frame as required by the end-to-end schedule of the flow.

Meshes are activated and deactivated by the presence or absence of data packets and are initiated and maintained by the

unicast destination or by a dynamically elected representative of the multicast destination. The first source that becomes active for a given destination sends its first data packet piggybacked in a Mesh Request (*MR*) packet that is flooded up to a horizon threshold. If the interest expressed by the source spans more than the single data packet, the intended receiver(s) of a *MR* will start the process of establishing and maintaining its routing mesh.

B. Information Stored and Exchanged

FIRST uses mesh announcements (MA) to establish and maintain routing meshes and destination meshes, to coordinate end-to-end schedules for RT flows and, in the case of a multicast group, to elect the core of the group. As long as there are active sources, the elected core or unicast destination periodically transmits MAs that contain monotonically increasing sequence numbers.

A *mesh announcement* (MA_D^{*B}) transmitted by node B for destination D is a eight-tuple of the form: $(id^{*B}, core_D^{*B}, sn_D^{*B}, d_D^{*B}, mm_D^{*B}, next_D^{*B}, slot_D^{*B}, cascade_D^{*B})$ 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. 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. $next_D^{*B}$ is the identifier of the preferred next hop of node B towards the core of D . We assume that the core of a unicast destination is always the destination itself. $slot_D$ is a randomly selected time slot identifier that is used in Eq. 1 as a reference to pick out the slots that will be reserved by the relays of a data flow. Lastly, $cascade_D^{*B}$ is a flag that indicates whether B has available slots in the interval defined by Eq. 1 of Section VI-A (i.e., if B is flow-ordered for destination D).

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 received from each of its neighbors regarding that destination. In our notation, and to differentiate from an announcement that has just been received, a MA received from neighbor B that is already stored in L_D is denoted as MA_D^B (with the $*$ dropped). The MAs 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:

$$MA_D^B \prec MA_D^A \Leftrightarrow (sn_D^B < sn_D^A) \vee (sn_D^B = sn_D^A \wedge d_D^B > d_D^A) \vee (sn_D^B = sn_D^A \wedge d_D^B = d_D^A \wedge id^B < id^A) \quad (2)$$

In addition to L_D , a node x also keeps track of the core of the destination ($core_D^x$), 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.

C. Processing Mesh Announcements

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.

x 's feasible distance to the core of D (fd_D^x) is a non-increasing function over time 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. Please note that the elements of a feasible set are not necessarily flow ordered.

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

The sequence number stored at node x for the core of destination D (sn_D^x) is a strictly increasing function over time that can only be reset by a change of core (Eq. 4). The max function of Eq. 4 is the max function defined over the natural numbers.

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

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. 2. By definition, the core of the group has a 0 distance to itself and its feasible distance is always 0. In this paper, we assume that link costs (lc) are always equal to 1.

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

The address of the next hop to the core of D ($next_D^x$) is also computed using the relation $\prec$ defined in Eq. 2, the current values of the feasible distance and sequence number, and the flow-ordered flag:

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

where $CF_D^x = \{i : i \in L_D \wedge fd_D^x = d_D^i \wedge sn_D^i = sn_D^x \wedge cascade_D^i = true\}$ is the set of x 's *flow-ordered* feasible neighbors for destination D . When routing elastic data packets, CF_D^x is relaxed to $F_D^x = \{i : i \in L_D \wedge fd_D^x = d_D^i \wedge sn_D^i = sn_D^x\}$, which is the set of x 's feasible neighbors for D .

Lastly, if a node x receives a MA advertising a core with a larger identifier then L_D is set to $\{MA_D^{*B}\}$, $core_D^x$ is set to $core_D^{*B}$. 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). mm_D^x equals nil in the case of a unicast destination. A node x is a mesh member 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.

D. Transmission of Mesh Announcements

Nodes transmit MAs to inform other nodes about updates in their routing state. Therefore, whenever the core of a destination generates a new MA with a larger sequence number, the latter is disseminated along the network advertising the new sequence number (Eq. 4) and establishing next hop pointers towards the core ($core_D$) (Eq. 6). The mesh composed of these next hop pointers from a 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 .

E. Packet Forwarding

When a source has data to send, it checks whether it has received, within the last three MA periods, a MA advertising the intended destination. If not, it broadcasts a MR. If so, the sender simply broadcast the data packet.

Upon reception of a data packet, nodes check for a hit in their data packet cache. 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. If the node is part of the destination, it also passes the data packet to upper layers. Eq. 8 is used by node x to decide if it has to forward a data packet with destination D received from neighbor y :

$$mm_D^x = RM \vee mm_D^x = MM \vee \exists y \in L_D : d_D^y > d_D^x \wedge next_D^y \leq id^x \quad (8)$$

Eq. 8 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. Elastic data packets follow directed meshes composed of a single path and RT data packets are routed using directed meshes composed of multiple flow-ordered paths. Please note, that in the case of a unicast destination, the first two terms of Eq. 8 are always false; hence, nodes only forward data packets if they are part of a selected source-destination shortest path.

VIII. PERFORMANCE RESULTS

We present simulation results comparing FIRST with ODMRP [10] for the case of multicast traffic, as well as AODV [11] and OLSR [12] for the case of unicast traffic.

In our experiments, ODMRP, AODV and OLSR run on top of IEEE 802.11 DCF [13] and all the protocols use a 802.11b physical layer. We selected these protocols because they have become *de facto* baselines for performance comparisons. Even though they were not designed for RT traffic, they are a good reference that allows us to highlight the performance gains of our approach.

We use packet delivery ratio, generalized group delivery ratio and end-to-end delay as our performance metrics. The generalized group delivery ratio is a multicast-specific metric in which a data packet is considered as delivered, only if it is received by at least a given proportion of the multicast group members. For this paper we set a threshold of 80%. We employ a combination of random waypoint and group mobility models as our mobility model.

We used the discrete event simulator Qualnet 3.9 [14] that provides well tuned versions of ODMRP, AODV, OLSR and IEEE 802.11 DCF ("WiFi" for short). Each simulation was run for ten different seed values. The multicast protocols use the same period of three seconds to refresh their routing structures (join query period for ODMRP and announcement periods for FIRST). FIRST's frame size is set to 200 slots and each slot has a duration of 0.5 ms. The value of Δ (see Section VI-A) is set to 21. For ODMRP, the forwarding group timeout was set to three times the value of the join query period, as advised by its designers. Table. I lists the details of the simulation environment. Due to space limitations, we do not show results for networks subject only to unicast or multicast traffic. However, those results follow the same pattern we discuss for the case of networks subject to unicast and multicast traffic.

TABLE I
SIMULATION ENVIRONMENT

Total nodes	100	Node placement	Random
Simulation area	$1800 \times 1800m^2$	Simulation time	150s
Phy. Model	802.11b	Tx. power	15dbm
		Tx. Rate	11000000bps
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

In these experiments the number of multicast groups is increased from 1 to 6 with 3 concurrent active sources per multicast group. One out of these three sources is defined as real-time (RT) while the remaining two are elastic. The selection of the RT source is random. Multicast sources are also group members. In addition to the multicast flows, we have five concurrent CBR flows among nodes that are randomly selected from nodes that are not part of any multicast group. Group members follow the group mobility model with group regions of $900 \times 900m^2$. CBR and MCBR sources transmit 10 packets of 200 bytes per second. The five unicast flows are defined as RT.

Fig. 1(a) shows the packet delivery ratio attained by OLSR and AODV when they are running in parallel with ODMRP

Fig. 1. Performance with increasing number of active groups, 3 MCBR sources per group and five unicast flows.

as well as the delivery ratio attained by FIRST. In the figure, the graphs labeled as “FIRST RT” present the average results taken over the RT flows only, the graphs labeled as “FIRST elastic” present the average results taken over the elastic flows only, and the graphs labeled “FIRST elastic+RT” present the average results taken over the totality of the flows. From the figure, we notice that FIRST multicast outperforms ODMRP even for the case of the elastic flows. The figure also shows that FIRST RT scales quite well as the number of multicast groups increases. Regarding unicast routing, we observe that AODV performs similar to FIRST when the network has one and up to four multicast groups. However, as the number of multicast groups increases and the network is more heavily loaded, FIRST unicast clearly outperforms both OLSR and AODV by delivering more than twice as many packets as OLSR, and around 25% more packets than AODV.

The group delivery ratio is shown at Fig. 1(b). In terms of this metric, FIRST performs similar to ODMRP for elastic traffic. However, the group delivery ratio attained by FIRST is 10% higher than that of ODMRP for RT traffic. Fig. 1(c) shows the end-to-end delay attained by the different protocols. The figure shows that, as the number of sources increases, the average delay of FIRST unicast is an order of magnitude smaller than that of AODV, which is an order of magnitude smaller than the one attained by OLSR. The delay of FIRST multicast for elastic data packets is higher than the one in ODMRP. The latter is caused by an increase in the number of high-priority control packets which is a function of the number of flows.

IX. CONCLUSIONS

We introduced FIRST, a cross-layer protocol framework for wireless ad hoc networks that integrates interest-driven routing with priority-based queueing for traffic management, end-to-end bandwidth reservations controlled by the routing, and distributed transmission scheduling. All these components work together to provide end-to-end delay and bandwidth guarantees to RT unicast and multicast data flows in MANETs. Our simulation results show that FIRST scales better than the traditional protocol stack which consists of the IEEE 802.11 DCF working independently of the routing protocols.

ACKNOWLEDGMENTS

This work was sponsored in part by the U.S. Army Research Office (ARO) under grant W911NF-05-1-0246, by the Baskin Chair of Computer Engineering and by the Mexican National Polytechnic Institute under grant SIP-20100417.

REFERENCES

- [1] L. Chen and W. B. Heinzelman, “A survey of routing protocols that support qos in mobile ad hoc networks,” *IEEE Network*, vol. 21, no. 6, pp. 30–38, November-December 2007.
- [2] T. Melodia, M. C. Vuran, and D. Pompili, “The state of the art in cross-layer design for wireless sensor networks,” *Lecture Notes in Computer Science*, vol. 3883/2006, pp. 78–92, 2006.
- [3] C. R. Lin and M. Gerla, “Asynchronous multimedia multihop wireless networks,” in *Proc. of INFOCOM '97*, vol. 1, Apr 1997, pp. 118–125 vol.1.
- [4] C. E. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers,” in *Proc. of ACM SIGCOMM '94*, 1994, pp. 234–244.
- [5] Y. Wu, P. A. Chou, Q. Zhang, K. Jain, W. Zhu, and S.-Y. Kung, “Network planning in wireless ad hoc networks: a cross-layer approach,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 136–150, Jan. 2005.
- [6] J. Yuan, Z. Li, W. Yu, and B. Li, “A cross-layer optimization framework for multihop multicast in wireless mesh networks,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 11, pp. 2092–2103, Nov. 2006.
- [7] P. Djukic and P. Mohapatra, “Soft-tdmac: Software tdma-based mac over commodity 802.11 hardware,” in *Proc. of INFOCOM 2009*, 2009.
- [8] L. Bao and J. J. Garcia-Luna-Aceves, “A new approach to channel access scheduling for ad hoc networks,” in *Proc. of MobiCom '01*. ACM, 2001, pp. 210–221.
- [9] W. Feller, *An introduction to probability theory and its applications*, 2nd ed. John Wiley, 1957, vol. 1.
- [10] 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. WCNC., 1999, pp. 1298–1302 vol.3.
- [11] 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., Feb 1999, pp. 90–100.
- [12] 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.
- [13] “IEEE Std 802.11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications,” pp. i-445, Nov 1997.
- [14] “Qualnet 3.9, Scalable Network Technologies: <http://www.scalablenetworks.com>.”