Title
LEMUR: Efficient Multicasting in Ad-hoc Networks Using Label Switching and Unicast Routing

Authors
Cirimelli-Low, Dylan
Garcia-Luna-Aceves, J.J.

Publication Date
2024

Data Availability
The data associated with this publication are within the manuscript.

Peer reviewed

LEMUR: Efficient Multicasting in Ad-hoc Networks Using Label Switching and Unicast Routing

Dylan Cirimelli-Low
Computer Science and Engineering Department
University of California, Santa Cruz
Santa Cruz, California, USA
dcirimel@ucsc.edu

J.J. Garcia-Luna-Aceves
Department of Electrical and Computer Engineering
University of Toronto
Toronto, ON, Canada
jj.garcialunaaceves@utoronto.ca

***Abstract*—To avoid forwarding loops and the transmission of unwanted replicas of multicast data packets, current multicast routing protocols designed for ad-hoc networks require routers to use packet caches listing enough information about multicast data packets that have been forwarded. In addition, existing multicast-routing solutions for ad-hoc networks either require a multicast routing protocol that operates concurrently with a unicast routing protocol, or adding substantial signaling to the baseline unicast routing protocol. We introduce a new approach for multicasting embedded in unicast routing that eliminates the need to use packet caches for multicasting along shared multicast trees by means of label switching, and incurs minimum additional signaling overhead to attain multicast routing.**

***Index Terms*—Routing, multicast, ad-hoc networks**

I. Introduction

Support of multipoint communication (one-to-many and many-to-many) in wireless ad-hoc networks is critically important for many applications. The many applications of wireless ad-hoc networks involving multipoint communication include search and rescue operations, military deployments, classrooms and conventions in which people share information dynamically using their mobile devices, and sensor networks in which sources send data to multiple sites. However, one-to-one communication (i.e., unicasting) is at least as important in the very same networks and is needed even as part of applications that require multipoint communication.

Traditionally, unicasting is supported by means of unicast routing protocols, and multipoint communication is supported using a multicast routing protocol in order to use the limited communication bandwidth efficiently by reducing the number of transmissions needed to reach all the intended receivers of a message.

As our summary in Section II of related work on multicast routing protocols in ad-hoc networks shows, prior multicast-routing approaches must rely on the use of packet caches with information about multicast data packets that have been forwarded recently. Unlike wired networks, in which routers may select the receivers for each packet by forwarding them to specific interfaces, wireless ad-hoc networks use a single broadcast channel shared by all nodes. Hence, the multicast routing protocols used in wired networks cannot be used in wireless ad-hoc networks without the use of additional mechanisms aimed at preventing the transmission of multicast data packet replicas or the transmission of multicast data packets along forwarding loops.

This paper introduces **LEMUR** (for Label-Enabled Multicasting on Unicast Routing), which is a technique based on label switching that enables core-based tree multicast routing in a wireless network without the need for packet caches, cross-layering mechanisms, or listing the routers a data packet has traversed. To the best of our knowledge, LEMUR is the first approach to multicast routing that does not require using packet caches or cross-layering mechanisms to allow multicast routing over broadcast channels.

Section III describes LEMUR. The key idea of LEMUR is to enable many-to-many communication over a bi-directional shared tree by forwarding to *labels* that encode virtual interface numbers, which mimic the functionality of a wired multicast architecture. LEMUR eliminates the need for a multicast routing protocol by constructing a core-based tree from the forwarding information base (FIB) of the loop-free unicast routing protocol running in the network, and then selectively forwarding multicast data packets by replacing the IP destination address with the correct label on a per-hop basis. A label encodes the necessary information so multicast data packets are only forwarded by a node's parent and children on the shared tree, and no looping occurs.

Section IV presents the results of simulation experiments showing that LEMUR induces minimum additional overhead on the unicast routing protocol within which it operates. We used RIPPLE-WiN [5] as the loop-free unicast routing protocol on which LEMUR operates because RIPPLE-WiN has been shown to be much more efficient than other unicast routing protocols for ad-hoc networks in terms of its signaling overhead. Section V provides our conclusions.

II. Related Work

Many multicast routing protocols for wireless ad-hoc networks have been proposed and implemented over the years, and a number of surveys describe the various approaches that have been proposed [7]. We only summarize a few of the approaches in the prior work on multicast routing. Prior multicast routing protocols for wireless ad-hoc networks can be classified into two classes: tree-based protocols and mesh-based protocols.

A tree-based multicast routing protocol establishes and maintains either a shared multicast routing tree or one multicast routing tree per source of a multicast group to deliver data packets from sources to receivers of a group. There are many well-known examples of this type of multicast routing protocol. Multicast AODV [12], which maintains a shared multicast tree for a group and augments the signaling of AODV to find group leaders. Other examples of tree-based approaches include ROMANT [14], which maintains a shared tree per group, and ADMR [6], which maintains a multicast tree per source per group.

A mesh-based multicast routing protocol establishes a mesh that connects all the receivers of a multicast group with a core defined for the multicast group or the sources of the group. The first example of this type of multicast routing protocol was CAMP [10], which uses unicast-routing information to find the pre-assigned core of a multicast group and uses its own signaling to build a mesh between the core and the receivers of the group. ODMRP [8] floods signaling packets originating at each source of a multicast group. Other examples of mesh-based approaches include PUMA [13], which works independently of the unicast routing protocol, similar shared-tree variants [15], as well as variants of ODMRP like DCMP [4] and NSMP [9].

What is important to note from all the prior work on multicast routing for ad-hoc networks is that both mesh-based and tree-based protocols require the use of packet cashes to avoid forwarding loops or duplicates of data packets.

III. LEMUR

The objective of LEMUR is to introduce minimum additional signaling to an existing unicast routing protocol to build a Core-Based Tree (CBT) [1] per multicast group and avoid the use of packet caches for the efficient forwarding of multicast data packets.

A. Signaling

Figures 1 through 6 illustrate the messages exchanged as part of LEMUR. In these figures, grey nodes denote receivers for the multicast group, and v^* denotes that router v is the group core. Solid lines represent the minimum spanning tree used to route packets from the multicast group core to interested nodes, and dash lines represent the remaining topological connections.

In order for a node to receive or transmit data as part of a LEMUR multicast group, it must first join the group by appending itself to the shared multicast tree of the group by sending an interest to be part of the group towards the core of the group. A LEMUR interest states: g, the identifier of the multicast group's core, r, the identifier of the intended receiver of the interest, and $\mathcal{L}_g^n$, the set of labels the node will transmit to for this multicast group. The usage of labels is discussed in detail in Section III-B.

Figures 1 and 2 show the steps taken to construct the CBT from a cold start. Referencing the FIB of the unicast routing protocol, a node n sends an interest message, denoted by $I()$, to the next-hop node towards the core.

If the intended receiver is not a member of the multicast tree for the group specified in the received interest, the receiver itself transmits an interest to its own next hop towards the core, as provided by the unicast FIB.

As shown in Figure 1(b), when node d receives the interest for group a from node e, d responds with a reply message $R(a, e, \mathcal{L}_a^d)$, which acknowledges receipt of e's interest in group a and allows d to advertise its own labels for the multicast group a, $\mathcal{L}_a^d$.

The process is repeated, as shown in Figure 2, until the interest from e reaches the group's core, at which point all nodes on the path from e to a have become members of the multicast group, either as a receiver or forwarder.

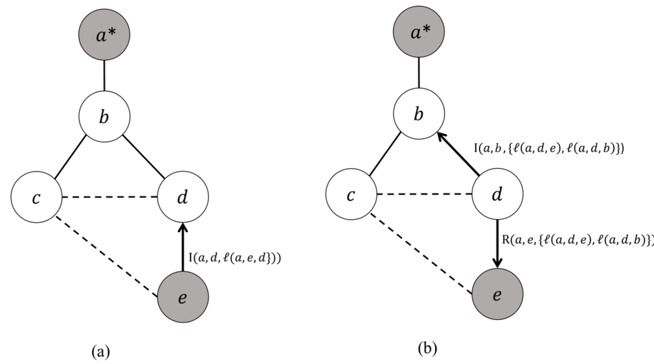


Fig. 1: Propagation of multicast interest

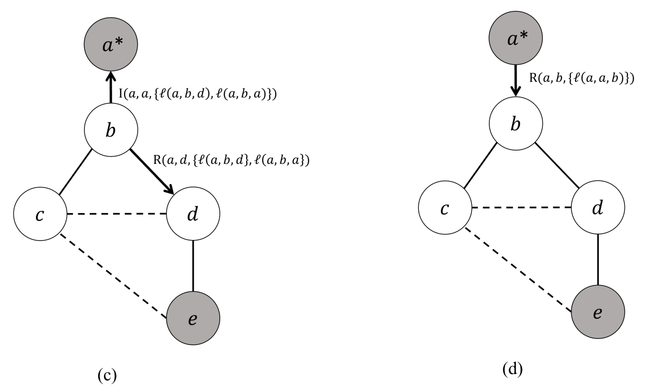


Fig. 2: Propagation of multicast interest (cont.)

An interior tree node that stops receiving interest from a child will take no action but passively remove the child from the multicast group. An interior tree node which have no remaining children will remove itself from the multicast tree by stopping it's transmission of multicast interests. Hence, a node that would like to receive multicast data, or has recently received an interest from at least one child, should periodically refresh its subscription to the multicast group by transmitting an interest to its parent in the shared tree.

Figure 3 demonstrates the steps needed to add a new subscriber to an existing multicast tree. In step (a), node c becomes interested in the multicast group a and transmits

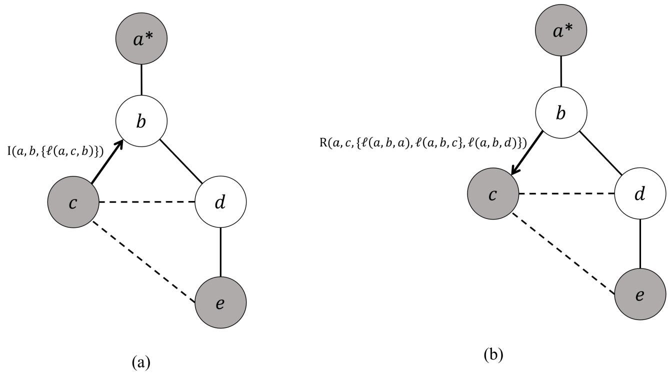


Fig. 3: Appending member to multicast tree

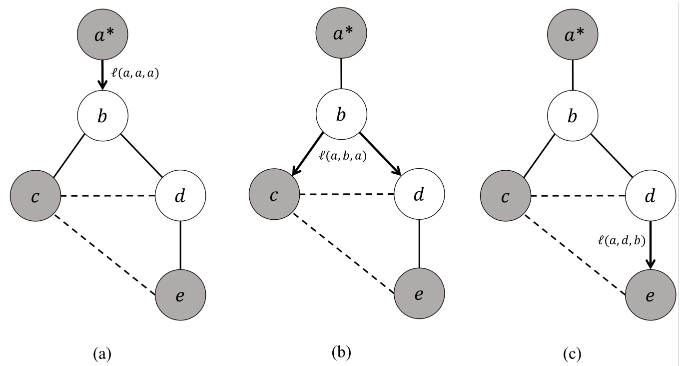


Fig. 4: Dissemination of multicast data from group core

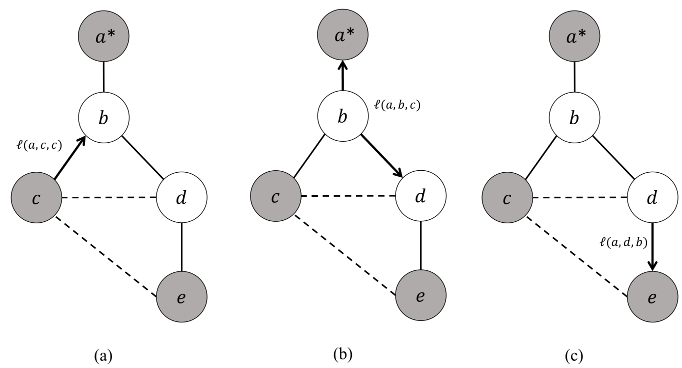


Fig. 5: Dissemination of multicast data from group member

the interest $I(a, b, \mathcal{L}_a^c)$. Because b is already a part of the multicast tree, b responds with the reply $R(a, c, \mathcal{L}_a^b)$ but does not propagate the interest up the tree. Although e also receives the multicast interest from c in step (a), c does not process the interest since the interest states b as the intended recipient.

B. Label switching

To prevent the creation of forwarding loops when transmitting multicast data packets, data should be forwarded only along the shared tree. That is, multicast data originating from a core or leaf node should only be forwarded by the tree child(ren) or parent, respectively. Data originating or forwarded by an interior node should be forwarded by both child(ren) and parent. To prevent ping-pong looping between tree nodes, a node should forward any given packet only once.

In a wired network, where each node or network segment is accessible through a unique interface, forwarding packets in such a way is trivial. As a protocol for wireless networks, LEMUR employs label switching to create virtual interfaces over which multicast data is forwarded.

A label, denoted ℓ, is a 32-bit integer value constructed from a hashing function that takes as input three identifiers: (1) the identifier of the multicast group core; (2) the identifier of the sender; (3) the identifier of the node that the data packet was received from.

For each multicast group g, node n maintains a set of labels, $\mathcal{L}_g^n$, representing virtual interfaces that exclude particular tree neighbors. In interest and reply messages for a particular multicast group, nodes advertise their sets of labels, which are locally stored by the receiver. For example, the label $\ell(g, n, z)$ corresponds to the virtual interface connected to all tree neighbors of n in group g, other than z.

To prevent ping-pong looping, when a node forwards a multicast data packet, it sets the destination IP address of the packet to the label corresponding to the virtual interface that excludes the previous hop.

Figure 4 illustrates how label switching is used to deliver multicast data from the group core to all subscribers.

In step (a) of Figure 4, a is the originator of a data packet, and, hence, has no previous hop, so it transmits with the label $\ell(a, a, a)$, which states that the data is for subscribers of group a and its own identifier is a. The packet is received by b, at which point b checks to see if the label $\ell(a, a, a)$ is known. Because the label is part of $\mathcal{L}_a^a$, which was advertised in a's reply to b, b is able to deduce that the received packet is from a and for group a, so b forwards the packet using the label $\ell(a, b, a)$, b's virtual interface for group a that excludes a as a receiver. In step (b), the packet is received by c and d. Since $\ell(a, b, a) \in \mathcal{L}_a^b$, which is known to c, c locally receives the packet as data from a. In step (c), d forwards the data packet to e using the label $\ell(a, d, b)$. Node b receives the packet but drops it since the virtual interface excludes b. Node c also receives the packet, but because c is not a tree neighbor of d and, therefore, cannot decode the sender's label, the packet is dropped.

Figure 5 illustrates how the same technique can be used to deliver multicast data from a group subscriber to all other subscribers. Interestingly, this example also reveals that shortest path routing of multicast data is not guaranteed for group members other than the core: although c is a direct neighbor of e, the data packet is routed through a common tree ancestor.

C. Handoff Procedure

Figure 6 shows how interest and replies are used to prevent the formation of forwarding loops.

In step (a) of Figure 6, node d moves to be a direct neighbor of a, the group's core; however, according to b, d is still its

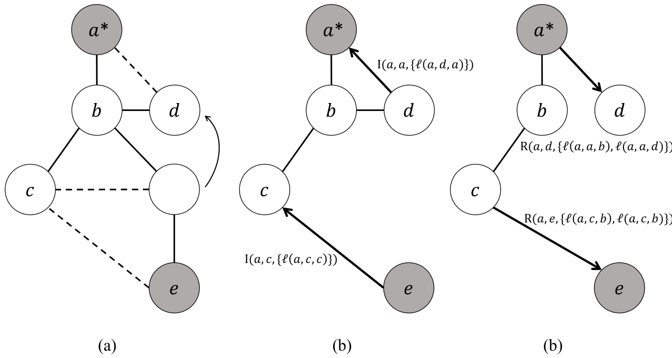


Fig. 6: Handoff procedure as a result of topology change

child. If d were to immediately select a as its new parent, a forwarding loop would form between a, b, and d. To prevent this, before d may forward packets received from a, it must send an interest directly to a, which announces to b that d should be removed as its child. Note that if d's interest to a is not received by b, either due to a collision or if b temporarily leaves b's transmission range, a temporary forwarding loop may form between b, d, and a. When e renews its interest in group a, it will choose c as its new parent since it is c's next hop towards a, as provided by the unicast FIB. In step (c), d and e are acknowledged as children of a and c, which enables them to forward packets received from their parents.

D. Information Maintained

In order to properly decode multicast labels, a node must store, for each group g, for each tree neighbor k, $\mathcal{L}_g^k$. An efficient way to store multicast labels that optimizes label lookup speed is to keep a map $\mathcal{M}$ that maps a given label to a tuple (l_g, l_s, l_e), the group, sender, and excluded identifiers, respectively. When a multicast interest or reply is received by node n from node k, for group g, n stores the following mappings in $\mathcal{M}$: (1) $\ell(g,k,k) \rightarrow (g,k,k)$; (2) $\ell(g,k,n) \rightarrow (g,k,n)$; (3) $\forall \ell(l_g,l_s,l_e) \in \mathcal{L}_g^k : \ell(l_g,l_s,l_e) \rightarrow (g,k,\emptyset)$, where $\emptyset$ indicates that the identifier of the excluded node is unknown. Accordingly, n forwards a packet if and only if $[\ell(l_g,l_s,l_e) \in \mathcal{M}] \wedge [(\ell(l_g,l_s,l_e) \neq \ell(g,k,n)]$. Because interest and reply messages both state the core identifier g and sender identifier s, L_g^s need not advertise $\ell(g,s,s)$, since the receiver can locally generate it.

In addition, a lifetime timer must be kept for each child in the multicast tree. If one lifetime elapses without a child c refreshing interest in the parent, the parent p removes the child from the multicast group and removes $\ell(g,p,c)$ from $\mathcal{L}_g^p$.

IV. Performance Evaluation

To evaluate LEMUR, we compare the performance of LEMUR when used with a multicast group versus a unicast routing protocol and equivalent set of unicast flows.

To ensure a fair comparison and to demonstrate that LEMUR eliminates the need for a multicast routing protocol, LEMUR is integrated with RIPPLE-WiN [5], a state-of-art routing protocol for ad-hoc networks.

To reduce control plane overhead, LEMUR messages are aggregated with the unicast routing messages generated by RIPPLE-WiN. LEMUR consults the FIB of RIPPLE-WiN and uses the "hello" messages exchanged (in the absence of routing updates) as a part of RIPPLE-WiN to ensure fresh neighborhood information.

Experiments are carried out in the ns-3 [11] network simulator and simulate ad-hoc networks using 802.11n Wi-Fi, configured for a data rate of 52 Mbps.

Each trial consists of a randomized network of 20 nodes constructed in the following manner: All nodes are placed randomly within a 100m x 500m bounding box; then, each node that is not within 50m of at least one other node is replaced randomly until the resulting topology is connected. This placement strategy ensures that all traffic flows are deliverable, at least when the network topology is static. In experiments with mobility, nodes randomly walk the bounding box and reflect off the boundaries.

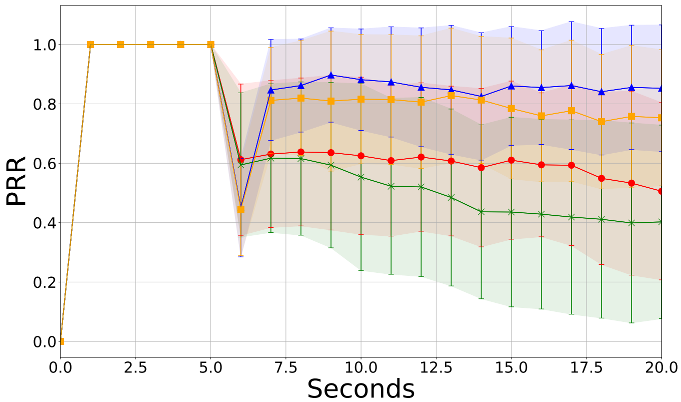
In each trial, a node is randomly selected as the source, and 5 other nodes are selected as sinks. For the multicast experiments, the source node acts as the multicast group core, and the 5 sink nodes begin to transmit multicast interests after 5 seconds, which is a sufficient amount of time to ensure that the RIPPLE-WiN routing tables have converged. For the unicast experiments, unicast UDP flows are created from the source to each sink. Each UDP flow consists of packets containing 1000 bytes of artificial payload, and the arrivals of such packets are drawn from a Poisson distribution with parameter λ. For both the unicast and multicast experiments, data-plane traffic is not generated until 5 seconds into the 20-second trials.

LEMUR is evaluated in both mobility-varying and traffic load-varying experiments, described in sections IV-A and IV-B, respectively. For Figures 7 and 8, each data point represents the mean of 50 trials, and error bars represent ± 1 standard deviation. Large standard deviations in results can generally be attributed to the fact that the depth of the forwarding trees for both unicast and multicast experiments may vary greatly between trials.

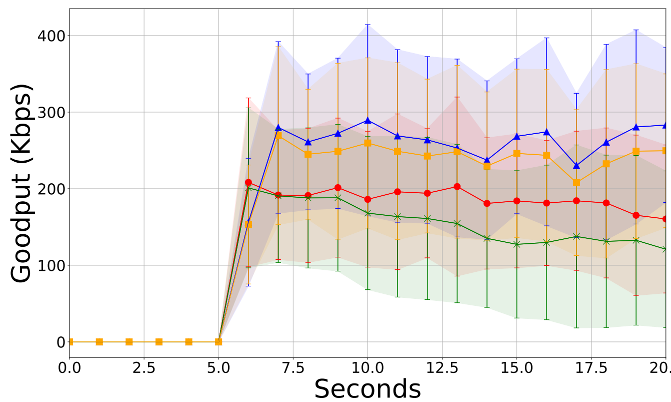
A. Mobility Experiments

The results of the mobility experiments are shown in Figure 7. In these experiments, λ is fixed at a value of 8, and the Wi-Fi nodes move at pedestrian levels of mobility– either 1 m/s or 2 m/s.

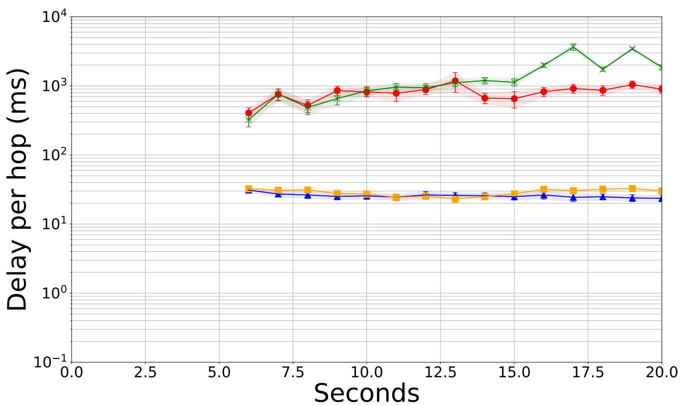
In Figure 7a, the packet reception rate is normalized for the number of intended receivers; in other words, if 3 of the 5 sink nodes receive a multicast data packet, the PRR would consider .6 packets to be received. Figure 7a shows, for both mobility levels, an increase of approximately 25% PRR when LEMUR multicast groups are used. This improvement in PRR can be attributed to the fact that multicast using LEMUR reduces congestion of the data plane, thereby reducing occurrences of multiple access interference. For both unicast and multicast



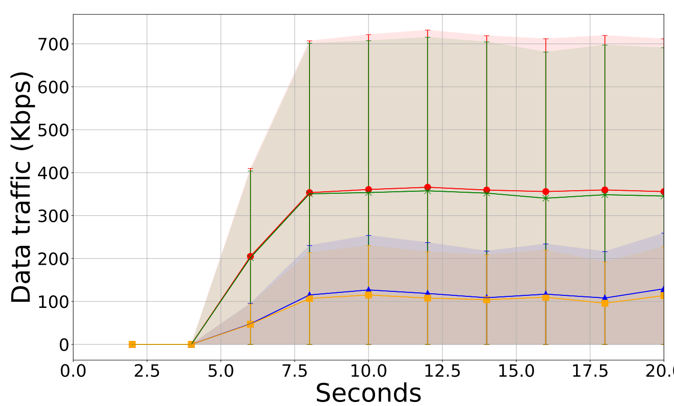
(a) Normalized PRR



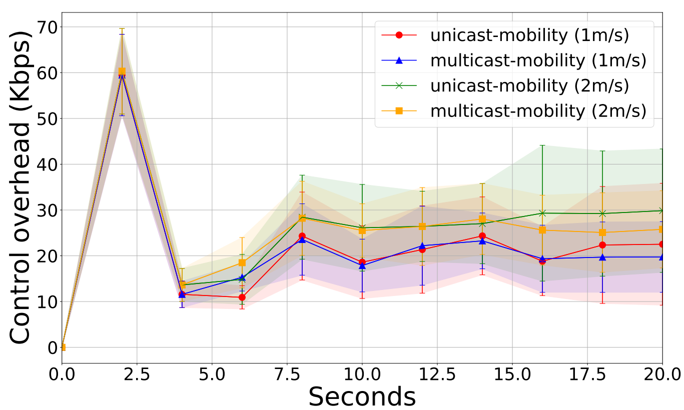
(b) Goodput



(c) Normalized delay



(d) Data plane overhead



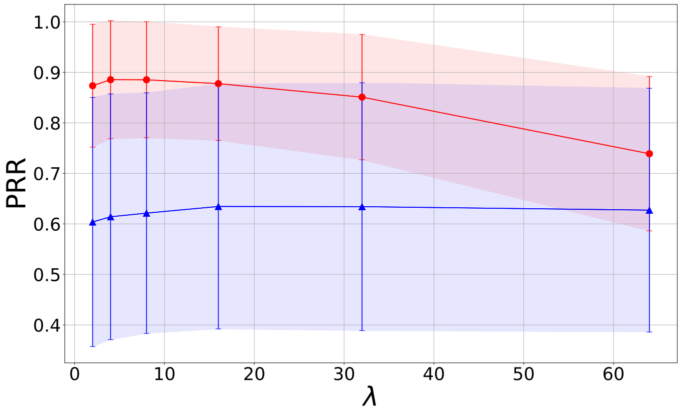
(e) Control plane overhead

Fig. 7: Results for mobility experiment

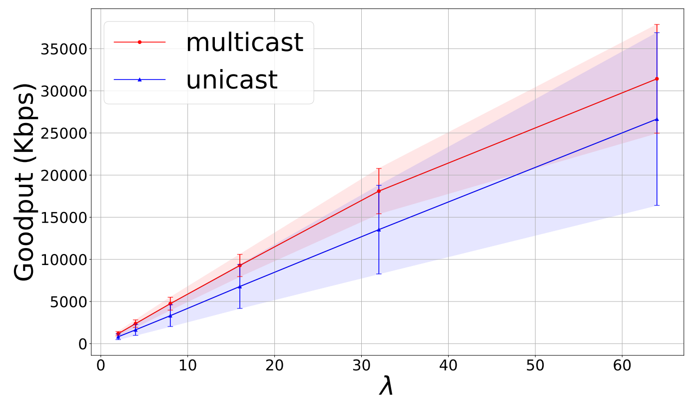
experiments, PRR is greater for 1 m/s than for 2 m/s since at higher levels of mobility, the information in the FIB is more likely to be stale, and a higher level of mobility induced more RIPPLE-WiN routing signaling, thereby increasing the potential for collisions with data packets.

The results of Figure 7b mirror that of Figure 7a and show that more application-layer data is delivered when LEMUR multicast is used. Here, a multicast data packet may be counted towards the goodput multiple times, once per sink that receives the packet.

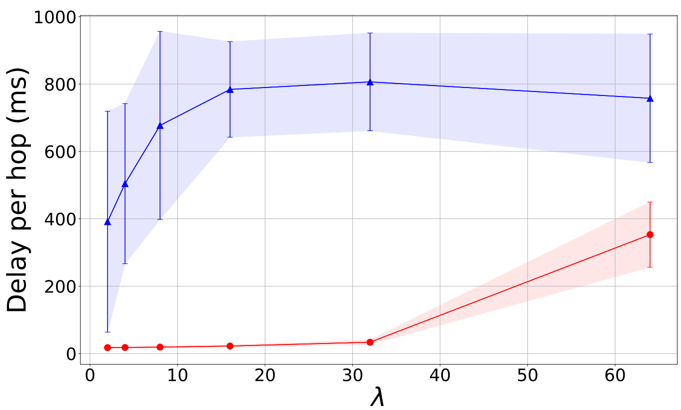
Figure 7c shows that LEMUR multicast reduces delay by two orders of magnitude. Here, delay is measured as the time from when a packet is enqueued in a node's transmission buffer until the packet is successfully delivered to the intended receiver, divided by the number of hops between the originator and intended receiver. Only packets that are successfully delivered are considered in the calculation. Because there is substantially less multiple access interference using multicast, MAC-layer queuing delays are much shorter using LEMUR. It



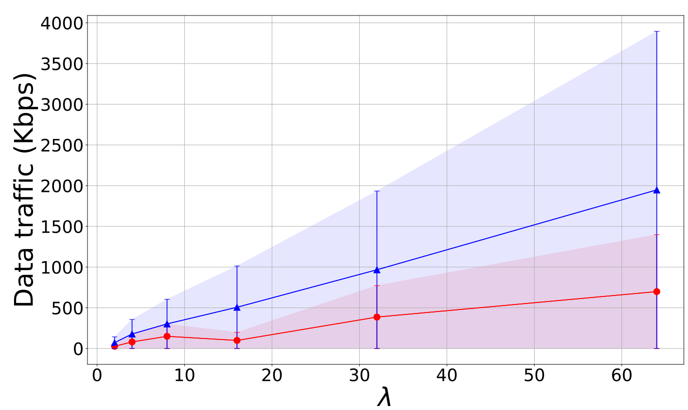
(a) Normalized PRR



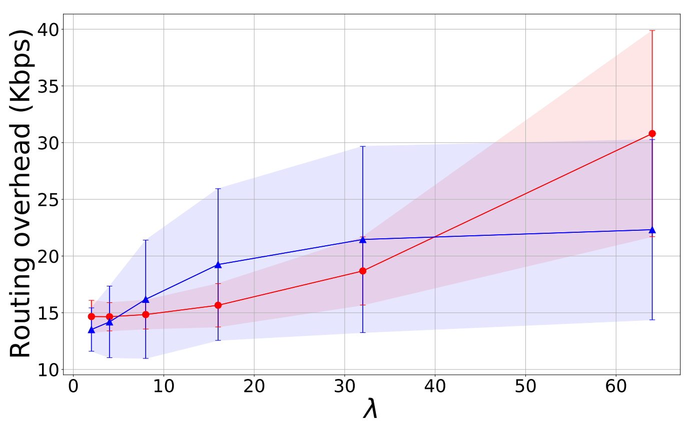
(b) Goodput



(c) Normalized delay



(d) Data plane overhead



(e) Control plane overhead

Fig. 8: Results for variable-load experiment

is important to note that multicast data packets in LEMUR are sent to the broadcast MAC address, which does not solicit an ACK and, therefore, cannot induce a backoff due to a collision at the receiver.

Figure 7d shows the aggregate overhead of data packets being transmitted or forwarded. As expected, the overheads of the unicast curves are significantly greater than those of the multicast curves since 5 unicast packets must be transmitted – one to each sink– to do the same work as one multicast packet from the group core. In theory, the data traffic is independent of the mobility level; however, we observe slightly higher data traffic at 1 m/s because the probability of a packet being dropped by the forwarded node is lower than at 2 m/s.

Figure 7e illustrates that there is virtually no additional control plane overhead introduced when using multicast since LEMUR taps into the unicast routing protocol's FIB and neighbor lifetimes.

B. Variable-Load Experiments

In the experiments shown in Figure 8, the traffic parameter λ is varied from 2 to 64 (where the inter-packet arrival times are drawn from an exponential random variable with mean $1/\lambda$), and the topology is static. For each trial, a data point represents the mean results of a 20-second run.

Figure 8a shows that for all traffic loads, LEMUR multicast has a higher packet delivery rate. At $\lambda = 64$, the PRR begins to decline due to high saturation. In a congested network, the forwarding of a downstream packet may interfere with the transmission of a new packet at an intermediate hop. This problem is exacerbated by the fact that broadcast transmissions do not use the 802.11 distributed coordination function to mitigate multiple access interference due to hidden terminals. Recently, several amendments ([2], [3]) to Wi-Fi have been proposed to improve the reliability of broadcasts by modifying the CTS-to-self and binary exponential backoff functions.

Figure 8b illustrates that at all congestion levels, LEMUR multicast delivers more data packets.

Figure 8c indicates that significant backlogging takes place when $\lambda = 64$. At higher traffic loads, packets experience longer queuing delays since 1) there is, with high probability, a number of arrivals ahead of the packet, and 2) the transmission of broadcast packets will be deferred until the channel is evaluated to be unused, using the Wi-Fi clear channel assessment mechanism.

Figure 8d illustrates that, at all traffic loads, multicast introduces less congestion to the data plane than unicast does.

Figure 8e shows the surprising result that when the network is congested, there is more control plane activity when multicast is used. However, the overall overhead is still lower than that of unicast. This increase in routing overhead is due to the fact that routing updates, which are themselves broadcast, have a higher probability of colliding with multicast data than unicast data, which may invoke the 802.11 DCF to avoid collisions.

V. Conclusions

We introduced LEMUR, a novel label-switching framework that enables multicast routing over a core-based tree in wireless networks. We have shown, by way of ns-3 network simulation, that by reducing the congestion in the data plane, LEMUR simultaneously increases the packet delivery rate while reducing per-hop delay.

As demonstrated in Figure 5, a fundamental flaw of CBT-based forwarding is that data packets must be routed through the core, regardless of the shortest path between group subscribers. One direction for future research is the development of a mesh-based forwarding scheme that allows multicast group members to route packets based on the shortest paths from the source to all other subscribers.

Given that LEMUR is designed to operate in an ad-hoc setting where the multicast group core may leave the network without warning, LEMUR could be made more resilient by adopting an election procedure that hands off the group to a new core without requiring complete reconstruction of the CBT. In the same vein, redundancy by way of multiple cores per group could improve LEMUR's ability to deliver multicast data if the topology becomes partitioned, and, therefore, some nodes would not have a path to the core.

References

[1] T. Ballardie, P. Francis, and J. Crowcroft, "Core Based Trees (CBT)," *Proc. ACM SIGCOMM '93*, Oct. 1993.

[2] C. Chousidis, R. Nilavalan and L. Lipan, "Expanding the use of CTS-to-Self mechanism for reliable broadcasting on IEEE 802.11 networks," *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2014.

[3] C. Chousidis, I. Pisca, I. and Z. Huang, "A Modified IEEE 802.11 MAC for Optimizing Broadcasting in Wireless Audio Networks," *J Netw Syst Manage*, 2020.

[4] S.K. Das, B.S. Manoj, and C.S. Ram Murthy, "A Dynamic Core Based Multicast Routing Protocol for Ad Hoc Wireless Networks," *Proc. ACM MobiHoc 2022*, June 2002.

[5] J.J. Garcia-Luna-Aceves and D. Cirimelli-Low, "Simple and Efficient Loop-Free Multipath Routing in Wireless Networks," *Proc. ACM MSWIM 2023*, Oct. 2023.

[6] J.G. Jetcheva and D.B. Johnson, "Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks," *Proc. ACM MobiHoc 01*, Oct. 2001.

[7] L. Junhai, et al., "A Survey of Multicast Routing Protocols for Mobile Ad-Hoc Networks," *IEEE Communication Surveys & Tutorials*, 2009.

[8] S.J. Lee, M. Gerla, and Chian, "On-demand multicast routing protocol," *Proc. IEEE WCNC 99*, Sept. 1999.

[9] S. Lee and C. Kim, "Neighbor Supporting Ad Hoc Multicast Routing Protocol," *Proc. ACM MobiHoc 2000*, August 2000.

[10] E. L. Madruga and J.J. Garcia-Luna-Aceves, "Scalable Multicasting: The Core-Assisted Mesh Protocol," *ACM Mobile Networks and Applications Journal*, 2001.

[11] NS-3 network simulator. Online: https://www.nsnam.org

[12] E. Royer and C. Perkins, "Multicast operation of the ad hoc on-demand distance vector routing protocol," *Proc. ACM Mobicom 99*, August 1999.

[13] R. Vaishampayan and J.J. Garcia-Luna-Aceves, "Efficient and Robust Multicast Routing in Mobile Ad Hoc Networks," *Proc. IEEE MASS 2004*, Oct. 2004.

[14] R. Vaishampayan and J.J. Garcia-Luna-Aceves, "Robust Tree-based Multicasting in Ad-hoc Networks," *Proc. Workshop on Multihop Wireless Networks, IPCCC 04*, Arizona, April 2004.

[15] R. Vaishampayan, K. Obraczka, and J.J. Garcia-Luna-Aceves, "An Adaptive Redundancy Protocol for Mesh Based Multicasting," *Computer Communications*, March 2007.