ECHO: Efficient Zero-Control Network-Wide Broadcast for Mobile Multi-hop Wireless Networks

Ram Ramanathan, Christophe Servaes, Warren Ramanathan
goTenna Inc.
81 Willoughby Street, Brooklyn, NY 11201
Email: {ram | christophe | warren}@gotenna.com

Abstract—**Network-wide broadcasting is a key requirement in military networks for supporting applications such as situation awareness and emergency messages. Current methods are based on flooding, or use control packets which limits scalability. We present a novel protocol called ECHO that constructs and maintains a broadcast backbone without using any control packets. Instead, using a field in the data packet header, a node listens for an "echo" of the specific packet that it transmitted to determine its membership in the backbone. ECHO is deterministic, source-independent, fully distributed, accommodates mobility, and balances battery consumption across nodes.**

We prove the correctness of ECHO and show that its communication complexity is $O(N)$ lower than that of Flooding and Multi-Point Relay (MPR) in dense networks. Simulations over random mobile networks show that ECHO uses 2.5x-9x fewer transmissions than Flooding to achieve similar delivery ratio. Experiments on a 12-node testbed of goTenna mobile mesh networking devices show that ECHO reduces transmissions by 3x, and increases battery life by >2x over Flooding. ECHO's performance advantages are crucial for scalable broadcast in low-power, low-capacity military multi-hop wireless networks.

I. INTRODUCTION

In a military Multi-hop Wireless Network (MWN), it is often necessary to do a *network-wide broadcast*, that is, send a packet from a given source to all nodes in the network. Examples include Position Location Information (PLI), situation reports, group chats, clock synchronization messages, and routing control messages [1], [2]. The Network-wide Broadcast (NWB) problem is to determine the (minimal) set of nodes that should re-transmit the packet so that it reaches all nodes in the network.

A simple solution to the NWB problem is *flooding*, namely having every node retransmit the message once. However, this results in excessive transmissions and collisions, causing what is commonly known as a *broadcast storm* [3]. This has motivated several efforts toward *efficient* NWB, that is, reducing the total number of transmissions. Most, if not all of these works are either probabilistic (i.e., do not guarantee delivery even in lossless conditions), assume location information, or utilize control packets to collect local or global topology information [4]. Control packets result in overhead that reduces capacity for data, and limits scalability, and in low-capacity systems, may be prohibitively expensive. Further, they cause network instability under lossy conditions, and are a security vulnerability [5].

We present the first deterministic, zero-control-packet, location-unaware protocol for efficient network-wide broadcasting in mobile multi-hop wireless networks. Called ECHO, our protocol uses node identifier information within the data packet header to determine – in a fully distributed and source-independent manner – the set of *critical*[1] nodes whose transmission is sufficient for network-wide broadcast.

ECHO consists of two inter-woven phases. In the Full Flood (FF) phase, which is done infrequently, a data packet is flooded. During this phase, a node marks itself critical if and only if the specific packet that it broadcast was "echoed" by another node, that is, if it is identified as the previous sender in a received packet. All other packets until the next FF constitute the Pruned Flood (PF) phase wherein only the critical nodes re-broadcast the packet. An overwhelming majority of packets are transmitted using PF, resulting in high broadcast efficiency. Unlike prior deterministic protocols (e.g. [6]) ECHO does not use any control packets or explicit topology information.

ECHO utilizes a distributed mechanism for randomly selecting the node that sends the occasional data packet via full flood, thereby randomizing the critical node responsibility amongst network nodes. This balances battery usage over nodes, and increases resiliency to failures. The critical nodes created in an FF phase originated from a particular source are *source independent*, that is, are valid for the origination of packets from any other source as well.

We prove formally that the set of critical nodes that ECHO produces is sufficient for source-independent network-wide broadcast. We also show that for dense networks the asymptotic communication complexity of ECHO is $O(N)$ lower than that of Flooding [7] and Multi-point Relay (MPR) protocol [6]. We have simulated ECHO and Flooding on multi-hop wireless networks of varying size, density and mobility with a PLI-like application model and a dismounted mobility context. Our results show that for a 30 node network, ECHO reduces the total packet transmissions by a factor ranging from 9x (high density) to 2.5x (lower density) while maintaining very nearly the same delivery ratio as Flooding. The number of ECHO transmissions increase much more slowly with increasing size as compared to Flooding, and are largely independent of velocity in the dismounted mobility regime.

---

[1] Also referred to in the literature as *dominating*, *relay* or *rebroadcast* nodes

ECHO has been implemented as part of the goTenna Pro [8] – a small handheld device for the military, first-responders and other professionals. The device pairs with a smartphone and functions as an MWN router, forming a mesh network with other such devices. We have created a testbed of goTenna Pro devices connected by controllable attenuators and examined the performance of ECHO and Flooding over a time-varying sequence of 11 topologies that very roughly captures the link dynamics of troops starting close to each other, then spreading out, and then coming back in. Our testbed results show that, in the scenario examined, ECHO uses roughly 3x less transmissions and increases battery life by 2x while maintaining delivery within about 2% of Flooding.

While ECHO is applicable to all MWNs, it is especially crucial for low-capacity, low-power applications such as short-burst long-range mobile networking where the additional overhead of control packets or flooding is often unaffordable.

II. RELATED WORK

The problem of network-wide broadcasting in multi-hop wireless networks has received much attention. Flooding [7], [9] has been a de facto solution. The problems with Flooding were highlighted in [3]. Soon thereafter, there appeared a number of works on efficient broadcasting [6], [7], [10]. In [11] a comparison and classification of solutions into probability-based, area-based and neighbor-knowledge methods is given.

Probabilistic, location-based and counter-based schemes are described in [3], and probabilistic schemes are also studied in [12], [13], [14]. These reduce the number of transmissions significantly, but are not reliable in that they do not guarantee delivery even in lossless conditions [4]. Reliable (deterministic) schemes were proposed in [6], [7], [10]. The multi-point relaying [6] and the dominant pruning [7] are both based on covering a two-hop neighborhood with a minimal set of one hop neighbors. Multi-point relaying is used as part of the IETF standard OLSR [1]. These schemes are *source dependent*, that is, the relaying nodes change based on source, and therefore may require carrying the relay nodes as part of a message. *Source independent* methods were proposed in [15], [16].

From a graph-theoretic viewpoint, the network-wide broadcasting problem in a centralized setting can be formulated as the *Minimum Connected Dominating Set* problem, which is NP-complete [17], and an approximation algorithm is given in [18].

Existing protocols are either centralized (impractical), probabilistic (unreliable), or obtain topology information via control packets [15] which limits scalability and increases vulnerability by blackhole or spoofing attacks [5]. ECHO is unique in that it is the first fully-distributed protocol that is deterministic, uses no control packets, and is source independent. Further, it randomizes the relay (dominating/critical) nodes to balance battery usage.

III. THE ECHO PROTOCOL

ECHO is a network-layer protocol that efficiently delivers an application-layer message to all reachable nodes in the network. Data packets are prefixed with a header consisting of the following relevant fields. The descriptions are with respect to a reference node R.

- *origin*: The node that originated the packet, its "source".
- *sender*: The node from which R received the packet.
- *previous-sender*: The node from which the *sender* first received the packet (N/A if sender is origin)
- *seq-num*: A sequence number unique at the origin.
- *flood-indicator*: An 1-bit field to indicate if this packet is a Full Flood (FF) or Pruned Flood (PF) – see below.

There are two kinds of network-wide broadcasts in ECHO: *Full Flood (FF)* and *Pruned Flood (PF)*. A data packet is marked either an FF or PF via the *flood-indicator*. A packet marked FF is sent using Flooding, that is, transmitted exactly once by all nodes. During an FF, each node determines if it should mark itself a *critical node* as described below; and during PF, only critical nodes transmit. Thus, the core part of ECHO happens in the FF phase where critical nodes constituting a broadcast backbone are selected distributively. FF packets are sent periodically to reset the critical nodes to account for topology changes. A vast majority of packets are sent in PF mode where only critical nodes transmit.

Upon receiving a data packet marked FF, node R first determines if it is a duplicate by referring to the *seq-num* field, as in Flooding. If it is not, then it re-transmits the packet, but before doing so, sets the *sender* field to R and the *previous-sender* field to the *sender*. If it is a duplicate, then unlike Flooding where the packet is simply discarded, R checks if the *previous-sender* field is its own id, namely R. If it is, we say that "R hears an echo". This would be true if the neighbor first received the packet from R. If R hears an echo of its transmission, then R marks itself "critical". However, if this condition is not met by *any* packet for a configured period, then R marks itself non-critical. The state diagram for this aspect of ECHO is shown in Figure 1.

Intuitively, if the packet was not echoed, it means that the packet was a duplicate for all neighbors, which in turn means all of its neighbors can receive the packet from some other node, and hence the node need not re-transmit subsequent packets, i.e., be critical. A formal proof is given in section IV.

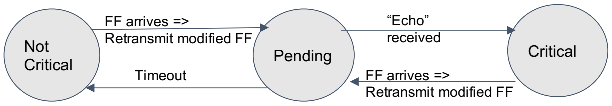


Fig. 1. ECHO critical node computation state diagram. "Echo Received" means receiving a Full Flood with previous-sender marked as own id.

Upon receiving a data packet marked PF, a node checks if its state is critical or pending. If so, it re-transmits the packet, else it does not. We observe that the originator of an FF is always a critical node.

The ECHO algorithm for determining node criticality is given in Algorithm 1. It marks a reference node R either critical or Non-Critical. Only key pieces of the logic are shown. In particular, the randomization of FF generation above

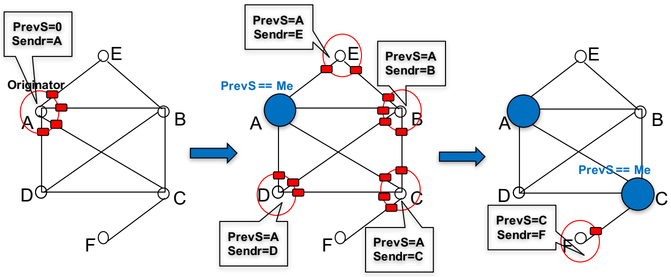


Fig. 2. Example ECHO operation on an FF originating from node A. Since nodes A and C are the only nodes that receive an "echo" (previous-sender equals identifier), they mark themselves critical (big filled circle) and the others mark themselves non-critical. Subsequent packets are only transmitted by A and C irrespective of originator (source independence).

is not included. An example execution of ECHO is illustrated in Figure 2. We formally prove its correctness in section IV.

Algorithm 1 ECHO Algorithm at node R

```
 1: procedure WHENPACKETRECEIVED(pkt)
 2:     if pkt.seqNum has not been received earlier then
 3:         pkt.previous-sender ← pkt.sender
 4:         pkt.sender ← $R$
 5:         transmit pkt
 6:         set echo timer                ▷ Round-trip delay + margin
 7:     else if pkt.previous-sender equals $R$ then
 8:         mark CRITICAL                 ▷ Echo received
 9:         discard pkt
10:     end if
11: end procedure
12: procedure WHENECHOTIMEREXPIRES
13:     mark NON-CRITICAL                 ▷ No echo received
14: end procedure
```

Data packets across all nodes are marked PF by default and FF approximately once every *FF-interval*, which is configured based on expected topology dynamics. This is done in a fully distributed manner as follows. Each node, upon receiving a data packet from the application, checks if it has received an FF within the last *FF-interval*. If not, it marks its packet as FF and all nodes execute the critical node determination as described above. Note that this FF suppresses the generation of FF from other nodes in most cases, although in rare cases multiple FFs are originated simultaneously. This does not affect the correctness of the algorithm, albeit there may be more critical nodes than necessary. On the other hand, this additional redundancy helps in increasing the delivery ratio under lossy conditions. By virtue of the FF origination randomization, critical node selection is also randomized when possible, thereby balancing energy consumption across nodes.

Between two FF's topology changes may cause loss in delivery. However, our experiments over random mobile topologies have shown that the loss is very much tolerable (see sections V,VI).

The Full Flood only happens for one data packet within the FF interval. Thus, if we have 30 nodes, and the FF and origination intervals are 60 and 30 seconds respectively (the simulation parameters in section V), there is only one FF for every 60 PFs. This ratio increases with decreasing origination interval and increasing size.

We note that while the *previous-sender* is an addition to the header, it is not equivalent to an entire control packet since it is a small (2 bytes in our implementation) addition that is constant (independent of size/density). We also note that while it is technically 2-hop-away information, it is *not* tantamount to a 2-hop *topology* because the links amongst the senders and previous senders are not known.

IV. THEORETICAL ANALYSIS

We first prove the correctness of ECHO. That is, given a graph G=(V,E), we show that Algorithm 1 running on each node will, in the FF phase, result in a *Connected Dominating Set (CDS)*: a subset of *dominating* nodes such that every node is either in the CDS or adjacent to a node in the CDS [18]. It is easy to see that a CDS is necessary and sufficient for guaranteed broadcast delivery barring packet losses. Below, we use the terms "critical" and "dominating" interchangeably.

For this section, we make three assumptions: the graph G is connected (A1); messages are ordered on receive, i.e., there is a notion of "first received" message (A2); and the system is lossless (A3). We note that these are for the proof only, our simulations and testbed results (sections V and VI) include packet losses and network partitioning. The discussion below is for a single Full Flood packet – we show that at the end of the flood a CDS is formed.

Definition IV.1. *Given a node u, let $p(u)$ denote the node from which u first received the packet. Let $c(u) = \{v\colon p(v) = u\}$.*

The $p(u)$ and $c(u)$ denote, respectively, the "parent" and set of "children" of u. Note that $c(u)$ can be an empty set, and $p(u)$ is undefined for the originator.

Observation IV.1. *The set of edges $(u, p(u))$ constitute a spanning tree of G rooted at the originator of the flood.*

To see this, note that by Algorithm 1 every node transmits the packet once. From assumptions A1, A3, every node receives the message, and by A2 there is a unique $p(u)$ for every u.

Lemma IV.2. *ECHO marks a node x critical if and only if $|c(x)| > 0$.*

Proof. Consider a node x. If $|c(x)| > 0$, there exists some child y of x. Per definition IV.1, y received the packet first from x. In the execution of ECHO (see Algorithm 1), y will invoke lines 3-5. The transmission will be received by x per assumption A3. In the execution of ECHO at x, lines 7-9 will be executed as x has already received this packets (line 2). Hence, node x is marked critical. For the "only if" case, if x is marked critical, then it must have received a packet from some y with previous-sender equal to itself, implying that y received the packet first from x. This in turn implies, per definition IV.1 that $x = p(y)$, hence $|c(x)| > 0$. □

By Lemma IV.2, all nodes except those without children, namely the "non-leaf" nodes, are critical nodes. Since the originator is reachable from every critical node via a chain of parent nodes, we have

Observation IV.3. *The set of critical nodes induce a connected subgraph.*

Lemma IV.4. *Every node with $|c(x)| = 0$ ("leaf" nodes) is adjacent to a critical node.*

Proof. Let x be a leaf node. Let $y = p(x)$. Clearly, y has a child, i.e., $|c(y)| > 0$. By lemma IV.2, y is a critical node, and since a parent is adjacent to the child, the lemma follows. □

The following theorem combines the above lemmas and proves the correctness of ECHO.

Theorem IV.5. *For any connected network $G = (V, E)$, and a given packet flooded from a node, the critical nodes as per Algorithm 1 form a connected, dominating set.*

Proof. Every node is eventually either critical or non-critical. Per Lemma IV.4, if a node is not critical, it is adjacent to a critical node. Therefore the set of critical nodes is a *dominating* set. Per observation IV.3, the critical nodes are *connected*. Thus, Algorithm 1 results in a connected dominating set. □

Neither Algorithm 1 nor the proof of Theorem IV.5 utilized the origin (source) of the packet, and a PF originated at any node can be delivered with only critical nodes transmitting. *Thus, ECHO is source independent.*

We note that while ECHO ensures that the critical nodes are chosen so that network-wide broadcast reaches all nodes, it does not guarantee optimality. This is not surprising since the Minimum Connected Dominating Set problem is NP-complete even in the centralized setting [18], and therefore polynomial-time optimal algorithms are highly unlikely.

We now consider the communication complexity CX, that is the number of bytes per second transmitted by ECHO, Flooding, and Multi-point Relays [6], a well-known method for reducing broadcast transmissions, and used in (OLSR) [1] protocol. Let N be the number of nodes and B be the data packet size in bytes. We assume *each node* periodically generates broadcast traffic at a rate of R_{gen} packets per second.

For Flooding (FLDG), each node generates $R_{gen}B$ bytes/sec, each of which is transmitted once by every other node. Thus

$$CX_{FLDG} = R_{gen}N^2B \tag{1}$$

We note that CX is not for a single packet but for R_{gen} packets per second per node – hence the communication complexity is $O(N^2)$ and not $O(N)$ even though each packet is transmitted only once by every node.

For ECHO, suppose the Full Flood frequency is $R_{ff} <= R_{gen}$, and let N_c denote the number of critical nodes. Then,

$$CX_{ECHO} = R_{ff}N(B + b) + R_{gen}N(N_c + 1)(B + b) \tag{2}$$

Here, the first term is the complexity of the Full Flood, with b denoting the additional fields required in the ECHO header, namely the sender and previous sender which are not required in pure flooding. We note that this is typically very small, so $b << B$. The second term captures the complexity of Pruned Floods being sent only by the critical nodes and the originator.

From eqs 1 and 2, the ratio of expected transmissions is:

$$\frac{CX_{FLDG}}{CX_{ECHO}} = \frac{N}{\frac{R_{ff}}{R_{gen}}(1 + \frac{b}{B}) + (N_c + 1)(1 + \frac{b}{B})} \tag{3}$$

Thus, the relative gain over Flooding increases with decreasing R_{ff}, and decreasing N_c. The slower the topology change, the smaller we can make R_{ff}, and increase the advantage of ECHO over Flooding.

We now consider Multi-Point Relays. In [19], the authors derive the control overhead for $B = 1$ as $hMN + \tau R_N D_N N$, where h is the Hello message frequency, M is the average number of neighbors, τ is the rate of topology control generation, R_N is the number of relaying nodes, and D_N is the average number of MPR links per node.

MPRs can be used independent of OLSR if no unicasting is needed. Therefore, in fairness to MPR, τ should be taken as zero. On the other hand, unlike equation 2, the equation in [19] excludes the cost of actually transmitting data, which is $R_{gen}NR_N$. Thus,

$$CX_{MPR} = hMN + R_{gen}NR_N \tag{4}$$

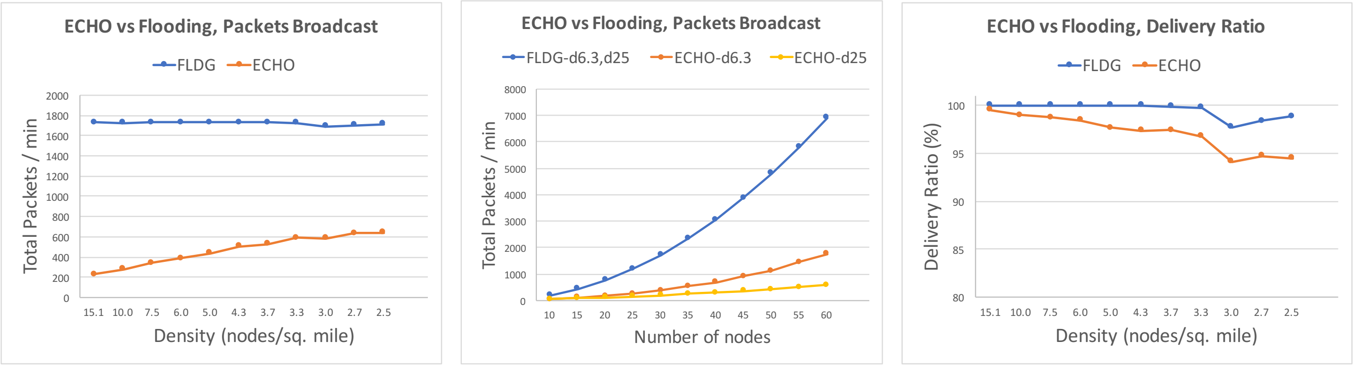
We consider the communication complexity above for two broad classes of networks: *dense* and *sparse*. By dense networks, we mean networks where the node degree $d = O(N)$, i.e., increases as the network grows. In such networks, the critical/relay nodes (N_c for ECHO and R_N for MPR) are nearly constant ($O(1)$) no matter the size of the network as each such node can reach $O(N)$ nodes. However, the average number of neighbors M in Eq. 4 grows as $O(N)$. By sparse networks we mean those where $d = O(1)$, that is, constant as N increases. In such networks, the relay nodes have to grow as $O(N)$ no matter the protocol.

Substituting the values of M, R_N, N_c into the equations 1, 2, 4, and using the fact that R_{gen}, R_{ff} and h are constants, the communication complexities are as given in Table IV.

TABLE I
COMMUNICATION COMPLEXITY

	Degree	Flooding	MPR	ECHO
DENSE	$d = O(N)$	$O(N^2)$	$O(N^2)$	$O(N)$
SPARSE	$d = O(1)$	$O(N^2)$	$O(N^2)$	$O(N^2)$

ECHO's communication complexity is $O(N)$ lower (better) than that MPR (and other topology based protocols) and Flooding in dense networks. Intuitively, the key reason for the $O(N)$ advantage over MPR in dense networks is the control traffic, in particular the additional $O(N)$ cost per node of conveying the 2-hop topology information in MPR. Since dense networks are quite common in military deployments,



(a) Transmissions vs density, N=30

(b) Transmissions vs size, density 6.3 and 25

(c) Delivery ratio vs density, N=30

Fig. 3. Total packets transmitted as a function of varying size, density ((a)-(b)), and delivery ratio vs density (c).

TABLE II
SIMULATION PARAMETERS.

	Parameter	Default	Sweep range
Network	Size	30 nodes	10-60
	Density	6.3 nodes/sq.ml	2.3-25
	Velocity	4 mph	1-12
	Traffic period	30 ± 5 s	N/A
	Transm. Range	1 mile	N/A
	Mobility	Rand. Waypoint	N/A
	Channel Loss	5%	N/A
ECHO	FF period	60 s	N/A
	Echo timeout	200 ms	N/A

the reduction of complexity in dense deployments is a crucial unique advantage of ECHO.

V. SIMULATION RESULTS

We have implemented ECHO within a discrete-event custom simulation model in Python. The reason we used a from-scratch model in Python rather than *ns3* or OPNET is to take advantage of graph analysis and network generation packages (e.g. NetworkX) in the Python ecosystem. The simulation models ECHO and Flooding in full detail on a per-packet basis. It has an abstract model of medium access control, that is, it introduces a random delay and loss (both configurable) for every packet. Traffic models the generation of PLI packets, with period randomly chosen between 25s and 35s. Table II contains the key parameters used in the model.

We use two performance metrics: (1) delivery ratio, which is the ratio between actual receives and expected receives across all nodes; and (2) the number of packets transmitted per minute. Both are standard measures for efficient broadcast evaluation [11], [4]. We compare ECHO with pure flooding. Each point represents an average over 10 randomly seeded simulations, each lasting 30 minutes.

Figure 3(a) shows that ECHO total transmissions are about 2.5x to 9x fewer than Flooding depending on the density. As density decreases, the network is sparser and ECHO needs more critical nodes and hence more transmissions. For Flooding, however, each node transmits every packet once, the number of packets is unaffected by density except at very low densities where the network is sometimes disconnected.

From Figure 3(b) we see that ECHO is far more scalable than Flooding. While both result in more transmissions as size increases, the increase in ECHO is at a much slower pace. The pace of increase for Flooding is the same irrespective of density since all nodes have to transmit once. For ECHO, the increase is slower at density 25/sq. mile than at 6.3/sq mile, in line with theoretical predictions. For 60 nodes at 25 nodes/sq mile, the gain over flooding is as much as 12x!

Figure 3(c) compares the delivery ratios of Flooding and ECHO. Since Flooding has the maximum possible redundancy, it is not surprising that ECHO cannot improve upon it. However, ECHO is always within about 3% of Flooding.

Finally, our experiments with varying velocity show that ECHO and Flooding transmissions and delivery ratio is largely unchanged with increasing velocity in the dismounted regime up to 12 mph – plot not shown due to space constraints.

VI. TESTBED RESULTS

We have implemented ECHO and Flooding in the firmware of the goTenna Pro device [8]. The goTenna Pro is intended for short-burst multi-hop communication over long ranges (measured in miles, depending on the terrain). In order to evaluate the performance, we have constructed a testbed with attenuators between devices to control the path loss between the devices. By adjusting the attenuator value using an automated script running on a Raspberry PI, many different topologies can be created.

For the testbed results reported here, the number of nodes is 12. The topology was progressively changed from a maximally dense "parking lot" topology to a sparse, disconnected topology, as shown in Figure 4. In the figure, each node represents *two devices*, each device connected to both the devices in the adjacent node. Network-wide broadcast packets were sent once every 30 seconds from every node, and each topology was in place for about 5 minutes before automatically moving on to the next in the sequence. At the end of this sequence, the topology was again put through additional changes going from sparser back to densest (not shown due to space constraints).

We have taken detailed measurements of performance on individual nodes' delivery ratio, battery lifetime and total transmissions. Due to space constraints we present only the

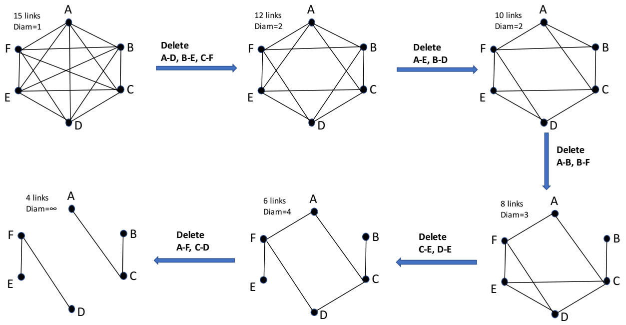


Fig. 4. First part of the topology sequence for testbed performance evaluation. The intent is to very roughly capture the link dynamics of troops starting in a "parking lot" and spreading out, and in the second part (not shown), coming back in. Each node contains two goTenna devices connected to each other and to each of the devices in adjacent nodes for a total of 12 nodes.

aggregate, top-level results in Table III. The second column shows the the *normalized transmissions*, which is the number of transmissions per originated packet. We used normalized transmissions here because Flooding drained the batteries earlier and therefore originated fewer packets. The third and fourth column show, respectively, battery lifetime averaged over all nodes, and the lifetime earliest drained battery.

TABLE III
AGGREGATE TESTBED RESULTS (SEE FIGURE 4 AND TEXT)

	Delivery	Norm. Xmissions	Avg Battery	Wrst Battery
ECHO	95.76%	3.55	8.83 hrs.	7.92 hrs
FLDG	97.86%	10.73	3.84 hrs.	3.42 hrs

The delivery ratio of ECHO is within 2.1% of Flooding, and is above 95%, while transmitting 3x less packets overall, in line with simulation results for 12 nodes. The average and worst case battery life is more than 2x longer than the corresponding values for Flooding.

VII. CONCLUDING REMARKS

Network-wide broadcasting is a key requirement in most military multi-hop wireless networks. Most practically viable protocols utilize topology information to compute relay nodes. This information is typically collected using control packets that can be prohibitively expensive especially in dense, low-capacity networks.

ECHO represents a radical departure from the prevalent thinking of collecting topology information via control packets to compute relay nodes. Rather, by using just two fields in the data packet header itself during occasional flooded packets, it learns source-independent critical nodes without control packets. Eliminating control packets makes the network more scalable, and invulnerable to control attacks. ECHO adapts to mobility, is tolerant of packet loss, and randomizes the set of critical nodes to balance battery consumption.

Simulation and testbed results have shown that ECHO significantly outperforms flooding on number of transmissions (2x-9x) and battery life (2x) while providing 95% or more delivery ratio. ECHO has a lower communication complexity ($O(N)$) in dense networks than both MPR [6] and Flooding ($O(N^2)$), while matching them for sparse networks. These improvements do not leverage any particular aspect of the MAC or RF, nor are dependent on the traffic in any particular way. Thus, ECHO can significantly enhance the scalability and lifetime of any multi-hop wireless network.

ECHO is robust, and scalable, making it a valuable protocol for real-world military multi-hop wireless networks. ECHO is currently being used in professional markets as part of the goTenna Pro [8]. Future work on ECHO includes further reducing the number of critical nodes, better resiliency to losses and node failures, and non-asymptotic scalability analysis.

ACKNOWLEDGMENT

The authors would like to thank Russ Dauer and the goTenna team for their input, encouragement and support.

REFERENCES

[1] T. H. Clausen and P. Jacquet, "Optimized link state routing protocol (olsr)," *RFC 3626; Available: http://www.ietf. org/rfc/rfc3626.txt*, 2003.

[2] C. E. Perkins and E. Royer, "Ad hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, February 1999, pp. 90–100.

[3] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wireless networks*, vol. 8, no. 2-3, pp. 153–167, 2002.

[4] I. Stojmenovic and J. Wu, "Broadcasting and activity scheduling in ad hoc networks," *Mobile Ad Hoc Networking*, pp. 205–229, 2004.

[5] B. Kannhavong, H. Nakayama, Y. Nemoto, N. Kato, and A. Jamalipour, "A survey of routing attacks in mobile ad hoc networks," *IEEE Wireless communications*, vol. 14, no. 5, 2007.

[6] A. Qayyum, L. Viennot, and L. Laouiti, "Multipoint relaying: An efficient technique for flooding in mobile ad-hoc networks," INRIA, Tech. Rep. Tech. Report 3898, 2000.

[7] H. Lim and C. Kim, "Flooding in wireless ad hoc networks," *Computer Communications*, vol. 24, no. 3-4, pp. 353–363, 2001.

[8] "gotenna pro," https://www.gotenna.com/pages/gotenna-pro-homepage, accessed: 2018-09-12.

[9] K. Obraczka, K. Viswanath, and G. Tsudik, "Flooding for reliable multicast in multi-hop ad hoc networks," *Wireless networks*, vol. 7, no. 6, pp. 627–634, 2001.

[10] G. Călinescu, I. I. Măndoiu, P.-J. Wan, and A. Z. Zelikovsky, "Selecting forwarding neighbors in wireless ad hoc networks," *Mobile Networks and Applications*, vol. 9, no. 2, pp. 101–111, 2004.

[11] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *Proc. ACM MOBIHOC*, 2002, pp. 194–205.

[12] D. Reina, S. Toral, P. Johnson, and F. Barrero, "A survey on probabilistic broadcast schemes for wireless ad hoc networks," *Ad Hoc Networks*, vol. 25, pp. 263–292, 2015.

[13] Y. Sasson, D. Cavin, and A. Schiper, "Probabilistic broadcast for flooding in wireless mobile ad hoc networks," in *Wireless Communications and Networking, 2003*, vol. 2. IEEE, 2003, pp. 1124–1130.

[14] V. Manfredi, R. Ramanathan, W. Tetteh, R. Hain, and D. Ryder, "Share: Scalable hybrid adaptive routing for dynamic multi-hop environments," in *Proc. IEEE Conf. on Ubiquitous Intelligence and Computing*, 2017.

[15] I. Stojmenovic, M. Seddigh, and J. Zunic, "Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks," *IEEE Transactions on parallel and distributed systems*, vol. 13, no. 1, pp. 14–25, 2002.

[16] J. Wu and H. Li, "A dominating-set-based routing scheme in ad hoc wireless networks," *Telecommunication Systems*, vol. 18, no. 1-3, pp. 13–36, 2001.

[17] M. R. Garey and D. S. Johnson, *Computers and intractability*. WH Freeman New York, 2002, vol. 29.

[18] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, 1998.

[19] P. Jacquet, A. Laouiti, P. Minet, and L. Viennot, "Performance of multipoint relaying in ad hoc mobile routing protocols," in *International Conference on Research in Networking*. Springer, 2002, pp. 387–398.