Long-Range Short-Burst Mobile Mesh Networking: Architecture and Evaluation

Ram Ramanathan, Christophe Servaes, Warren Ramanathan
Engineering Department
goTenna Inc
Brooklyn, New York, USA
{ram|christophe|warren}@gotenna.com

Ayush Dusia, Adarshpal S. Sethi
Dept. of Computer and Information Sciences
University of Delaware
Newark, Delaware, USA
{adusia|sethi}@udel.edu

Abstract—Decentralized off-grid short-burst communications for public safety and other applications require mobile wireless networks that can be inexpensively and instantly deployed to cover a large area. Maximizing coverage often necessitates trading bit-rate for range. The resulting capacity is insufficient to support existing mesh networking protocols due to their excessive control overhead, and motivates a fresh approach. We present the networking architecture of goTenna—a long-range lightweight device for mobile mesh networking. Our protocol uses a novel zero-control-packet approach for broadcasting and unicasting that builds state by observing packet header information. We describe its experimental evaluation using ns3, and on a goTenna testbed, concluding with some research challenges.

I. MOTIVATION

There is a need for an infrastructure-free communications system for short-burst data communications such as collaborative mapping in the context of public safety, military, and recreation, and for sensing in the context of Internet-of-Things (IoT). Such a system should be decentralized (for robustness), inexpensive, lightweight (for portability), have a very long battery life, allow mobility (of first-responders), and network a significant area with a limited number of devices.

The last point is crucial. A number of mesh and ad hoc networking products and prototypes based on WiFi and similar technologies have failed to address the problem because their short range results in inadequate connectivity or excessive deployment cost which renders them impractical and susceptible to the “zero-start” problem [1]. Unfortunately, with all else equal, any practical range-increasing technique results in a reduction in bit-rate. As a case in point, LoRa [2], a fast-growing standard for long-range communication, has bit-rates of 0.25-50 kbps. While this may be sufficient for short-burst data if we consider just the data, the problem is that networking protocols do not scale at these low bit-rates.

The literature is replete with thousands of MANET broadcast and unicast routing protocols, and several standards exist, such as AODV [3]. However, most if not all such protocols use dedicated control packets such as Hellos, Link-State Update, Route Request/Response etc. The ultra-low capacities characteristic of long-range short-burst communications cannot support this control packet overhead even for modestly-sized networks (further discussed in section II), motivating a dramatic reduction if not elimination of control packet overhead.

II. ARCHITECTURE AND PROTOCOLS

Figure 1 shows the various components comprising the goTenna system. goTenna hardware comes in two main versions: the Mesh, and the Pro, details in [4]. Both run the same Aspen Grove™ mesh protocol stack, explained below. Above this stack, the applications run on a smartphone (Android or iOS), that pairs to the device via Bluetooth. Example applications include texting, emergency beaconing and collaborative mapping. The goTenna device can also be used as a standalone relay, and new applications can be written on top of the device via an open SDK. All application packets are encrypted end-to-end.

The Aspen Grove mesh suite in the Pro consists of a CSMA-based MAC protocol that we call “CSMA lite” as it is a simplified, yet effective version that does not use any control packets, including ACKs. The transport layer provides end-to-end reliability and delivery notification (for unicast). The network layer consists of two protocols: ECHO for efficient zero-control-packet network-wide broadcasting; and VINE for efficient zero-control-packet unicasting. Both utilize header fields of packets to build, respectively, broadcasting and unicasting state. Except for one field, namely the previous sender, the header fields are commonplace in most headers.

1 Approximately 1 mile outdoors depending on terrain, with a record of over 40 miles (free space)
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 determines if it should be critical, that is, relay future packets, or not. A node marks itself critical if and only if the specific packet that it had 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. Details can be found in [5].

VINE uses three fields in the header, namely the sender, the previous sender, and a hop count to build gradient state to 1-hop neighbors, 2-hop neighbors, and origin of the packet, respectively. Over time as traffic flows, an increasingly rich sink tree toward each node is created, resembling the growth of a “vine” in a “grove”. Packets are forwarded along non-increasing gradients until the destination is reached. If there is no fresh-enough gradient state, the packet is broadcast. VINE provides per-hop reliability via implicit acknowledgements, that is, retransmissions based on overheard forwarded packets, and delivery notification via end-to-end acknowledgments.

III. EXPERIMENTAL EVALUATION

Figure 2 shows the results of ns3 simulations comparing ECHO with Flooding and Multi-Point Relay [6] for Broadcasting (2(a)) and VINE with AODV [3] for Unicasting (2(b)). ECHO significantly outperforms both Flooding and MPR, delivering about 26% more packets at 50 nodes. In fact, MPR performs worse than flooding due to its frequent neighborhood transmission. VINE’s outperformance of AODV is even more significant - at 50 nodes nearly 5x more packets are delivered by VINE. The relative gains hold across various combinations of density, mobility and data rate (not reported here).

We have constructed a testbed of 12 goTennas running ECHO and VINE software with attenuators between devices to control the path loss as shown in Figure 3 (top left). By adjusting the attenuator value using an automated script running on a Raspberry PI, many different topologies can be created. Further, any mobility model (e.g. random waypoint) can be emulated by changing the attenuation at each time snapshot in concordance with the model. For the testbed results reported here, the topology was progressively changed from a maximally dense “parking lot” topology to a sparse topology, and back, shown in Figure 3. 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.

A sample summary of testbed results is given in Table I. As can be seen, ECHO provides 55% longer average battery lifetime than Flooding, and delivers more packets than Flooding. Many such experiments have been conducted, including with VINE (not reported due to space constraints).

TABLE I
AGGREGATE TESTBED RESULTS (SEE FIGURE 3 AND TEXT)

<table>
<thead>
<tr>
<th></th>
<th>Delivery</th>
<th>Avg Battery</th>
<th>Wrt Battery</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECHO</td>
<td>89.1%</td>
<td>7.75 hrs</td>
<td>6.5 hrs</td>
</tr>
<tr>
<td>FLDG</td>
<td>84.9%</td>
<td>5.04 hrs</td>
<td>4.0 hrs</td>
</tr>
</tbody>
</table>

IV. DEPLOYMENT AND CHALLENGES

The goTenna Pro has been deployed in fighting forest fires, in the aftermath of hurricanes, and in the military. Several tens of thousands of goTenna Mesh have been used for outdoor recreation, and several thousands are being deployed in small businesses throughout New York city as part of a FEMA grant for hurricane preparedness. Yet, several research challenges remain within the general theme of ultra-low-capacity mesh networking, including further improving ECHO and VINE performance, developing an efficient MAC protocol, optimizing location updating, using multiple radio interfaces, increasing bit-rate, and scaling to large sizes.

REFERENCES