A Diagnostic Tool for Ad-Hoc and Delay-Tolerant Networks

Edoardo Biagioni esb@hawaii.edu University of Hawai'i at Mānoa

***Abstract*—Ad-Hoc and Delay-Tolerant Networks (AHDTNs) can be very useful in environments where more traditional networking technologies fail. Determining the practical effectiveness of AHDTNs can however be challenging. We review design considerations and practical experience with a novel mechanism for monitoring and analyzing the performance of AHDTNs. This new mechanism, AllNet Trace, somewhat resembles a combination of `ping` and `traceroute`, and can be used for the same purposes as either `ping`, with or without IP record route, or `traceroute`. AllNet Trace can be used to obtain information about networks where packets may be forwarded to next-hop destinations more than once, or may be arbitrarily delayed.**

We describe the design, implementation, and performance of AllNet Trace within the context of the AllNet protocol, which is designed to securely deliver interpersonal data over both AHDTNs and the Internet.

***Index Terms*—Ad-Hoc Networks, Delay Tolerant Networks, Network Monitoring, Network Analysis**

I. Introduction

Connecting mobile devices directly to one another without infrastructure, has been both a subject of research and a focus of practical development. The latter has been done by companies [1], [2], [3] and NGOs [4], [5] rather than researchers. The middle ground, between research networks and practical networks, has not been widely explored.

AllNet [6], [7], [8], [9], [10] is intended to be useful on a variety of devices including wireless or wired, and to develop new research that takes into account this practical experience.

As a result of the research and development to date, AllNet currently supports interpersonal communication of text messages up to about 500 characters. Depending on network availability, messages are delivered over the Internet, over ad-hoc and delay-tolerant wireless communications, or both.

Supporting delay-tolerant networking (DTN) requires that individual devices cache others' messages and forward them as possible, usually in an ad-hoc fashion. Devices that connect to the Internet intermittently can retrieve cached messages from other AllNet devices that have persistent Internet connections, an interaction that resembles an email client retrieving messages from an email server. Devices connected to the Internet automatically self-organize into a Distributed Hash Table (DHT) [11], [12], [13], [14]. The DHT makes it easy for intermittently connected devices to locate and retrieve their messages, and avoids the need for centralized servers.

When a device is connected to the Internet, it forwards each message to a small number of DHT nodes. When the device can communicate with other devices over Ad-Hoc networking, they forward messages to each other in a manner resembling Epidemic routing [16], so messages are forwarded to any device that hasn't received them yet.

With these different mechanisms, it is important to be able to test how well the network is working. One way of testing is to set up accounts for fictitious users – as a fully distributed system, AllNet accounts are created by exchanging keys among two devices, so creating special testing accounts is free and straightforward. While this end-to-end testing is useful, it does not reveal the path that messages take across the network.

Further, the implementation of AllNet is divided into a networking kernel called the AllNet daemon and one or more user interface clients. All clients running on a single host communicate through the same daemon. The daemon is independent of the clients, and could be included as part of any standard operating system. Just as `ping` elicits a reply from the OS kernel, AllNet trace requests elicit replies from the AllNet daemon.

AllNet Trace is a new mechanism that works well in Ad-Hoc and Delay Tolerant Networks and combines many of the benefits of `ping` and `traceroute`. The contributions of this paper include the description of AllNet Trace and practical experience using AllNet Trace.

II. AllNet Trace

AllNet Trace is a novel mechanism combining many of the features of `ping` and `traceroute` [21], together with ideas drawn from the seldom-used IP record route option [18].

Specifically, a trace is a distinct packet sent by any AllNet device. As in the IPv4 record route option, each participating AllNet device adds its own ID to the trace packet before forwarding it, and also sends a trace reply packet back to the sender. This resembles a combination of `ping`, `traceroute`, and the IP record route option. As in `ping`, responses allow determination of liveness and round-trip latency. As in `traceroute`[21], devices that forward the transmission may also reply, letting the sender reconstruct the path(s) taken by the original transmission. As in the IP record route option, each outgoing packet may record a trace of participating nodes that forward it, and each reply packet returns this trace to the original sender.

Unlike these three mechanisms, AllNet Trace combines these features seamlessly. AllNet Trace packets include a bit to request responses from forwarding devices, which if set allows Trace to behave more like `traceroute`, and if clear, makes Trace behave more like `ping`. When forwarding devices do send responses, each response is sent directly to the original sender. Another bit specifies whether responses should include the IDs of forwarding devices, as in `ping` with IP record route. While IP record route may only record up to

9 forwarding devices, AllNet Trace messages may include over 40 entries in a 1500-byte message.

The result of a hypothetical trace is shown in Figure 1. As is typical, the first reply comes from the local system, 0 hops away.

A. Operation of the AllNet Trace Command

Every AllNet device self-selects a 128-bit trace address that, if selected at random, will generally be unique. Uniqueness is not required, and IDs may be set manually, or devices may be set to use a different ID for each trace. Only some of the bits of this ID are sent in each reply – the present AllNet implementation by default sends 16 meaningful bits.

Trace reply messages contain a list of these device trace addresses, shown as two-byte hex numbers in Figure 1.

```
device_A: src/allnet/v3/bin/trace
trace to matching destination:
 1:    0.005891s rtt,  0 89.51/16
trace to matching destination:
       0.005891s rtt,  0 89.51/16
 1:    0.075356s rtt,  1 f6.5d/16
trace to matching destination:
       0.005891s rtt,  0 89.51/16
       0.075356s rtt,  1 f6.5d/16
 1:    0.599550s rtt,  2 ce.76/16
sent 1 packet, received 3
```

Fig. 1. The output of the trace command in a hypothetical network. The trace ID 89.51 is used by the local system.

Trace requests contain a per-packet trace ID, selected at random independently of the per-device trace address. Trace IDs serve to associate trace replies with trace requests. The output of the trace command replaces trace IDs with numerical sequence numbers, shown as `1:` in Fig. 1.

Trace requests also have a flag indicating whether replies are desired from intermediate devices, and zero or more trace entries recording earlier devices that forwarded this request. And like every AllNet packet, trace message contain a hop count and a hop limit and source and destination addresses. If the bits specified in a destination address match the first few bits of the trace ID, the trace request is addressed to this device. Trace requests sent with 0 bits of destination address are intended for any destination.

With `-m`, AllNet Trace requests that only devices matching the destination address reply, and that (unless `-i` is specified) intermediate devices add their information before forwarding requests. The resulting information is similar to that of `traceroute`, although the method by which the information is collected is different. `Traceroute` sends packets with increasing Hop Limit (also known as Time To Live or TTL), and records the response when packets are dropped. This method works well when each packet is forwarded only once by each router, and when routers return an indication that a packet has been dropped. Neither of these is true for AllNet – packets may be dropped or forwarded once or multiple times, with no feedback returned to the sender.

Running a successful `traceroute` results in transmission of six packets per hop. Running AllNet trace results in transmission of at most one packet per forwarding device. This number scales with the size of the network, but since any other AllNet packet takes priority over Trace packets, AllNet Trace cannot be used for Denial of Service (DoS) attacks.

Since AllNet Trace collects information within the trace packet itself, responses return a reliable record of devices that the outgoing packet has visited. The IPv4 Record Route header option [18] collects similar information and has been an inspiration for AllNet Trace, but is limited in size (at most 9 hops), and is only infrequently used for network diagnostics (IPv6 Record Route is even more limited, and can only be used with a loose source route).

To function more like `ping`, the `-i` switch requests that no intermediate nodes respond, so that all replies are from destination nodes. With 0 bits of destination address specified, this will elicit a reply from all the nodes in the network, whereas with a specific destination only the selected node(s) will respond. Examples are shown in Figure 2.

```
device_A: src/allnet/v3/bin/trace -i ce
 1:   0.138126s rtt, 2 ce.76/16
sent 1 packet, received 1
device_A: src/allnet/v3/bin/trace -i
 1:   0.001769s rtt, 0 89.51/16
 1:   0.025070s rtt, 1 f6.5d/16
 1:   0.106987s rtt, 2 ce.76/16
sent 1 packet, received 3
```

Fig. 2. Trace commands requesting replies only from the final destination(s). In the second command, the unspecified final destination address matches every possible AllNet address.

A trace request with no bits of destination address is useful when networks are small and frequently changing, as in the case of many AHDTNs.

AllNet Trace by default stops collecting data after 5 seconds, but can be told to wait arbitrarily long before giving up. The GUI version of the trace tool, shown in Figure 3, collects data continuously.

Figure 3 is a real trace, and shows actual Trace IDs. By convention, devices that are permanently connected to the Internet and intended to seed the DHT have been assigned special IDs where only the first few bits are non-zero. For these specific hosts, Trace IDs are set manually.

This trace shows that DHT "seed" devices `00`, `80`, `c0` are all active – DHT seed device `40` was down at the time of this trace. Non-seed devices also join the DHT whenever they are on the Internet. In this trace, all the devices other than the last are on the Internet in some way or other. As long as Internet-connected devices are reachable from the outside, i.e. as long as incoming connections and messages are not blocked by a firewall, such devices can be full-fledged members of the DHT.

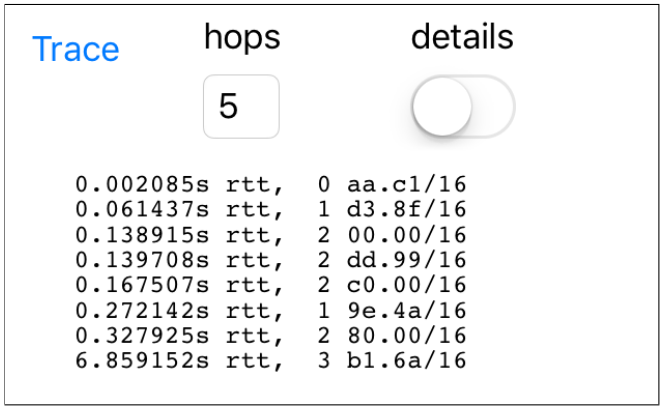

Fig. 3. A real trace in the AllNet xchat GUI.

According to the Trace in Figure 3, two devices are directly connected to this device (which has Trace ID `aa.c1`). Since the Internet counts as a single hop, directly connected devices may be physically very distant. Four devices are each two hops away, and the last device is three hops away.

B. The AllNet Trace Protocol

The AllNet protocol includes a packet type for management of the AllNet network. This is somewhat similar to the function of ICMP in the Internet, and similar systems for other networks.

Currently AllNet has five major groups of management packets: **beacons** used to manage peer-to-peer links, **peer** and **DHT** packets used to connect with other devices across the Internet, **data requests** used to request data cached in other devices, **keep-alive** packets for miscellaneous uses, and **trace** packets.

Trace packets are of two types, trace requests and trace replies. Both carry zero or more trace entries. As a trace request is forwarded and if `intermediate_replies` is set, the number of trace entries increases as new entries are added to record the path of the packet through the network. In contrast, the size of a trace reply remains the same when it is forwarded.

Each trace entry records up to 64 bits (8 bytes) of address and the number of address bits that are valid. In all the examples above, addresses have 16 valid bits, and 16 bits is used by default by the current implementation, but the protocol supports any number between 0 and 64 inclusive.

Each trace entry also records the number of hops in the AllNet header of the trace request that was received, and the local time (with an estimate of the accuracy of the local clock) at which the trace reply was forwarded. To the extent that clocks of different devices are accurately synchronized, the local time of forwarding can be used to estimate forwarding delays.

Trace requests and replies are forwarded like any other AllNet packet, except with low priority. The low priority means that Trace packets never interfere with regular traffic. Since data packets have higher priority than trace traffic, data packets should be able to reach any device that trace can reach.

III. Evaluation of AllNet Trace

For small AllNet networks, sending AllNet Trace requests to a 0-bit destination address is a good way to discover the entire network, as shown in the second part of Figure 2. This is useful in highly dynamic networks, and the trace can be repeated as often as desired, e.g. with the Unix `watch` command.

For larger or less dynamic networks, a single trace can reveal the path to a destination. Just like `traceroute`, network administrators may use the result of a trace to improve the performance of the network.

Since AllNet Trace packets are self-contained and carry all information about the trace, a trace packet received with considerable delay is still useful and may provide valuable information, especially if intermediate devices have added their own information.

Unlike `ping` and more similar to `traceroute`, AllNet Trace packets require additional processing compared to normal data packets. Because of this, and because trace packets are sent with the lowest possible priority, any latency measurement is likely to be an overestimate of the latency that would be seen by data packets and their acks.

The local response gives an idea of the overhead of replying to traces. Table I shows minimum, average, and maximum response times in a trial of 100 trace requests addressed to each machine's own Trace ID. The two machines both run a recent version of Ubuntu Linux. Based on information in `/proc/cpuinfo`, the slower machine has a 32-bit 800MHz Celeron CPU with no onboard cache, the faster machine has dual 64-bit Pentium 3GHz processors with 3MB caches.

MHz	bits	CPU	Linux	min	avg	max	avg ping
800	32	Celeron	4.4	9.3	19.8	38.2	0.13
3,000	64	Pentium	4.10	0.4	1.0	13.3	0.03

TABLE I
COMPARISON OF TRACE OVERHEAD ON A SLOWER AND A FASTER MACHINE RUNNING `bin/trace -i -r 100 -t 1` *local address*. TIMES ARE IN MILLISECONDS (MS).

Table I also shows the average time to `ping` localhost on each machine. Clearly `ping` is much faster, perhaps in part due to being built-in to the Linux kernel.

A trace was also done to a device across the Internet, but only one AllNet hop away. The traces averaged 142ms, with a minimum of 104ms and a maximum of 674ms, compared to a ping to the same machine measuring between 100ms and 103ms. These numbers show that while at its best the performance of AllNet is close to the performance of `ping`, AllNet has much more variability. This variability can be explained in part by the implementation of AllNet as a collection of local processes communicating via sockets.

Taking a broader view, even with this variability, out of 100 packets sent, AllNet has delivered each one within less than a second. Since AllNet is designed to deliver interpersonal

communications, such delays are acceptable in an early version of AllNet such as this.

IV. Related Work

People have diagnosed networks ever since networks were first built, leading to a variety of mechanisms at all levels of the OSI stack [20]. The AllNet Trace mechanism described here belongs on the Network layer of the OSI stack, the same as all IP mechanisms including `ping` and `traceroute`,

The unicast version of AllNet Trace, if one existed, would somewhat resemble the modified `traceroute` proposed in 1993 [21] and obsoleted in 2012 [22], with the difference that AllNet Trace explicitly specifies, in each packet, whether it seeks responses from intermediate devices. As mentioned above, this is also similar to the IP record route option [18]. The 1993 proposal also carries additional link information whose usefulness in a dynamic mobile network is not clear.

The differences between AllNet trace and the regular `traceroute` were described in Section II-A. Under highly dynamic conditions `traceroute` reports inconsistent information reflecting the differences in the paths taken by successive probe packets, whereas by consolidating all information in a single packet, AllNet Trace can give a consistent view of the path taken by a trace request. If no intermediate responses are sought, AllNet Trace behaves more like `ping`.

AllNet forwarding on AHDTNs is inspired by Epidemic Routing [16], with packets prioritized according to local rules. Similar to AllNet, Vahdat and Becker note the necessity of "placing an upper bound on message hop count and per-node buffer space (the amount of memory devoted to carrying other hosts messages)", and that it is "desirable to have multiple copies of a message in transit simultaneously."

More recently, and focusing entirely on DTNs, Grasic and Lindgren [23] review other studies and proposed algorithms, attempting to compare and summarize different research contributions with differing goals, including "delivery ratio, average delay or overhead ratio". Unlike such studies, AllNet Trace is a hands-on tool for diagnosing actual live networks. Rather than attempting to satisfy important but abstract network goals, AllNet Trace is likely to be used once a problem is detected or suspected. For example, AllNet Trace can show that a device that was previously reachable no longer is, or that an intermediate device that is needed for end-to-end communication is not responding to trace messages.

V. Conclusions and Acknowledgements

This paper describes a new diagnostic tool, AllNet Trace, that is proving useful in a practical network designed to provide connectivity among mobile devices leveraging Ad-Hoc Networking, Delay-Tolerant Networking, and, when Internet access is available, Distributed Hash Tables.

Network diagnostic tools are invaluable when building and evaluating real networks, and as such deserve careful study. This paper analyzes one such tool in the context in which it is used. In particular, AllNet Trace is well suited to the broadcast nature of AllNet, and provides useful information about packet forwarding in Ad-Hoc and Delay Tolerant Networks as well as across the Internet.

The principles of AllNet Trace should be broadly applicable to networks with high delays and networks where intermediate devices may forward packets more than once.

The author gratefully acknowledges contributions by Henry Eck, Marifel Barbasa, Andreas Brauchli, Tiago Couto, Caterina Desiato, Henry Eck, W. Wesley Peterson, and others.

References

[1] Tom Simonite, "The Latest Chat App for iPhone Needs No Internet Connection", Technology Review, March 28, 2014.
[2] Nick Statt, "GoTenna creates a cell network out of thin air anywhere on Earth", CNET, July 17, 2014.
[3] Matthew Humphries, "FabFi: an open source wireless network for $60 per node", geek.com, June 27 2011.
[4] Klint Finley, "Out in the Open: Take Back Your Privacy With This Open Source WhatsApp", wired.com, May 19 2014.
[5] Tom Simonite, "Build Your Own Internet with Mobile Mesh Networking", Technology Review, July 9, 2013.
[6] C. Desiato and E. Biagioni, "Sharing Networking Resources to Create a Pervasive Infrastructure", Ninth International Conference on Technology, Knowledge, and Society, 13-14 January 2013, Vancouver, Canada.
[7] E. Biagioni, "A Ubiquitous, Infrastructure-Free Network for Interpersonal Communications", 4th International Conference on Ubiquitous and Future Networks (ICUFN 2012), July 4-6, 2012, Phuket, Thailand.
[8] E. Biagioni, "Ubiquitous Interpersonal Communication over Ad-Hoc Networks and the Internet", at the 47th HICSS (Hawaii International Conference on Systems Sciences), in January 2014.
[9] E. Biagioni, "Distributed Anonymous Computation of Social Distance", 13th Annual IEEE Consumer Communications and Networking Conference, 9-12 January 2016, Las Vegas.
[10] "AllNet, A technology for ubiquitous interpersonal communication" https://sourceforge.net/projects/allnet/?source=directory
[11] Ratnasamy, Francis, Handley, Karp and Shenker, "A scalable content-addressable network", ACM SIGCOMM 2001.
[12] Stoica, Morris, Karger, Kaashoek, and Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications", ACM SIGCOMM 2001.
[13] Rowstron and Druschel, "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems", 18th Int'l Conf. on Distributed Systems Platforms, 2001.
[14] Zhao, Huang, Stribling, Rhea, Joseph, Kubiatowicz, "Tapestry: A Resilient Global-Scale Overlay for Service Deployment", IEEE JSAC, vol. 22, 2004.
[15] Jonathan Warren, "Bitmessage: A Peer-to-Peer Message Authentication and Delivery System" Nov. 2012, `https://bitmessage.org/bitmessage.pdf`
[16] A. Vahdat and D. Becker. "Epidemic routing for partially connected ad hoc networks". Technical Report CS-200006, Duke University, April 2000.
[17] E. Biagioni, "Mobility and Address Freedom in AllNet", June 6, 2014. http://hdl.handle.net/10125/34243
[18] Information Sciences Institute, University of Southern California. "Internet Protocol", IETF RFC 791, September 1981. The Record Route option is described on pages 20-21.
[19] E. Biagioni, "AllNet: using Social Connections to Inform Traffic Prioritization and Resource Allocation". 2012, `http://alnt.org/social-distance.pdf`
[20] International Standard ISO/IEC 7498-1, Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model, second edition 1994, corrected and reprinted 1996-06-15.
[21] G. Malkin, "Traceroute Using an IP Option", RFC 1393, January 1993. The more commonly-used `traceroute` is described in Section 1.
[22] C. Pignataro and F. Gont, "Formally Deprecating Some IPv4 Options", RFC 6814, November 2012.
[23] S. Grasic, A. Lindgren, "An analysis of evaluation practices for DTN routing protocols", CHANTS 12, 7th ACM international workshop on Challenged networks, pages 57-64, Istanbul, Turkey, 2012.