Survivable, Real Time Network Services

Defense Advanced Research Projects Agency
Contract F30602-98-1-0225, DARPA Order G409

Quarterly Progress Report
1 January 2000 - 31 March 2000

David L. Mills
Electrical Engineering Department
University of Delaware

1. Introduction

This report covers the work done in support of the DARPA Information Technology Office program in computer networking. Contributors to this effort include Prof. David L. Mills and graduate students Qoing Li and Tamal Basu. The project continues previous research in network time synchronization technology jointly funded by DARPA and US Navy. The technology makes use of the Network Time Protocol (NTP), widely used in the Internet, together with engineered modifications designed to improve accuracy in high speed networks. Specific applications benefiting from this research include multicast topologies, multimedia, real-time conferencing, cryptographic systems, and management of distributed, real-time systems.

This quarterly report is submitted in traditional report form on paper. As the transition to web-based information dissemination of research results continues, almost all status information and progress reporting is now on the web, either on pages belonging to the principal investigator or to his students. Accordingly, this and future progress reports will contain primarily schedule and milestone data; current status and research results are reported on web pages at www.eecis.udel.edu/~mills in the form of papers, technical reports and specific briefings.

2. Autokey

The main focus of work during the last quarter has been the implementation, test and deployment of the Autokey public-key cryptographic authentication scheme in NTP Version 4. The Autokey protocol design is summarized on the web status page and briefing slides at www.eecis.udel.edu/~mills/autokey.htm. These documents have been updated to the current state of the security model and authentication scheme. As of late March, the Autokey design has been implemented, tested and deployed at selected sites in CAIRN. There were several unanticipated problems in this deployment and much was learned.

There are many unobtrusive cryptographic values instantiated by the protocol that are updated when the RSA key pair, Diffie-Hellman parameters and related private values are updated in the normal course of operation. Originally, we thought these values would be updated at relatively infrequent intervals, like once every three months. In that case, it would not be intrusive to restart the NTP daemons in order to recover the latest public values. This turned out to be a significant security vulnerability, especially at or about the time the values were updated.

Accordingly, the protocol was redesigned so that public value refreshment does not require daemon restart and so that refreshment does not have to be synchronous throughout the community

of users. The scheme requires every public value used in the protocol to be signed and have an authentic timestamp, not just the keys and parameters. In order to deflect clogging attacks, the protocol was redesigned with the first line of defense a timestamp check. In case of replay attack, old signatures are discarded before the expensive public-key algorithms are invoked. The design remains potentially vulnerable to a clogging attack involving bogus values with timestamps in the future.

The requirement that every cryptographic value carry a timestamped signature extends to all data values, including the public key, Diffie-Hellman parameters, host name and the files they occupy. The files already have the creation timestamp embedded in the file, but it is necessary that the timestamp be embedded in the file name itself in the form of a file name extension. This allows new file generations to coexist in the various filesystems along with ones currently in use. A server rolling a new RSA key pair, for example, simply restarts the daemon with the new files and all clients will eventually notice this and load the new files without stopping the daemon.

Extensive testing with contrived scenarios pointed out possible obscure vulnerabilities where an intruder could force a server or client into an unstable state and cause the client to time out or loop or otherwise destabilize the Autokey or NTP protocol. In all cases found so far, the only completely robust behavior requires the client to disregard errors or attempted intrusions and in the worst case cause the client to time out and restart the cryptographic association. In the redesign, the timeout period is the NTP reachability period, which is eight times the poll interval. In practice, this should represent acceptable behavior, since the clock discipline algorithm is designed to survive that without significant performance degradation.

It is clear that public value dissemination in a large network is going to be a real problem. The present scheme which requires sending all generated values to a central site and then broadcasting to all clients is not acceptable for other than testing and experiment. This points up the need for centralized certificate storage and retrieval using directory services such as DNS. However, there still remains a bit of work to provide certificate retrieval as an automatic feature in NTP. Specifically, the current scheme using a forked process to access DNS services must be upgraded to a fully asynchronous process. Solutions to this requirement exist; however, it is not clear how generic they will be over the suite of 24 known ports of the software distribution. Fortunately, the volunteer corps of NTP engineers seems to have the porting issues well in hand.

2.1 Future Plans

From the above discussion, it is clear some development in protocol design and implementation is necessary to avoid the vulnerabilities mentioned. In truth these are minor modifications of existing code and should be completed in a month or two. Meanwhile, we are beginning work on a comprehensive report describing the Autokey protocol rationale, design and implementation. It will include a vulnerability analysis and threat assessment. We expect to complete the report by the end of May.

We have obtained a version of the Secure DNS software and brought it up on a FreeBSD development machine here. We plan to use it in testing the asynchronous resolver and certificate retrieval code.

3. Network Simulator

The progress of our project in simulating very large networks has been reported in previous quarterly reports and in web pages and briefing slides. As mentioned previously, we believe the simulator can be an important tool for studying the dynamics of very large networks with the order of 10,000 nodes. However, one of the problems identified in the original project goals was the need to simulate real internets where the structures tend to clusters with fractal-like topologies. Graduate student Tamil Basu is developing a program that can generate random topologies with a fractal character that may more closely emulate the real Internet of today.

Mr. Basu started with the work done by recent graduate Robert Redwinski, who developed a topology generator and discrete event simulator capable of supporting 10,000 nodes with an average connectivity of at least three. Preliminary experiments have been carried out on smaller networks in the order of 3,500 nodes with representative routing protocols used in the Internet of today. The results show the protocols operate as expected in response to induced trauma such as link and node failures. However, the elusive behavior we are looking for and that could explain past observations of anomalistic behavior in the ARPANET and NSFNET have not yet been found.

The simulator is comprised of three components:

1. Random Topology Generator (RTG), which generates random candidate networks according to the Waxman model.

2. One or more routing algorithms, implemented as a collection of finite state automata which communicate over the network.

3. Simulation engine, implemented as a conventional discrete event simulator with several features elaborated below.

The routing algorithms, which are at the heart of the simulator, have been chosen to be the most generic as possible, yet to expose inherent design limitations. Keeping this consideration in mind, the algorithms that have been implemented include variants of Bellman Ford (BF) and the Distance Vector Multicast Routing Protocol (DVMRP). While these algorithms have well known design deficiencies, they were chosen to establish the boundaries of network stability and are expected to be replaced with more subtle versions of BF and Dijkstra/SPF.

For BF the topology generator provides the user with several command line options such as split-horizon in order to more realistically emulate algorithms now in use. The simulation engine itself is a conventional discrete event simulator including a global event queue and event service modules. The design is heavily influenced by the need for very large numbers of nodes and links. One of its key features is a scheme which maintains only one global network state matrix, together with relatively small local event queues which incorporate local deviations, such as changes in the global matrix while a message is in transit.

These three components are currently functional and in use for studying the response of the routing algorithms to induced transients and probabilistic failure scenarios. However, we believe the Waxman model, which produces two-dimensional homogeneous network topologies, does not accurately model the Internet, which has a distinctly nonhomogeneous structure with LANs connected to MANs connected to WANs.

3.1 Approach

In order to model some kind of clustering or fractal topology, the RTG must be overhauled. The RTG is currently an offline process which computes the topology as a database which is then input to the simulation engine. This saves a great deal of time and effort during the simulation, which is of great importance considering the size of the networks, as well as the processing time and memory involved in generating and simulating them.

The RTG currently creates topologies based on the Waxman model, which sites nodes at random coordinates of a two-dimensional space and interconnects them randomly with given average connectivity. This is the principal difference between our model and conventional ones based on OpNET and NS. It directs the output of the topology generation to an ASCII file which is then used by the simulator engine.

The case to be made here is that, in spite of the fact that a random generator is used to develop the technology, it is still unable to mirror the real life scenario where networks are built upon each other. Instead it provides a planar network, albeit random, a structure which is rarely found in real life networks. To solve the issue it is our proposal to implement the RTG in a somewhat different manner, i.e., to harness the inherent advantages of the Waxman model, but at the same time to implement the overall topology using the properties of fractals which is a very good model for the manner in which real life networks exist nowadays.

For this purpose there are two candidate approach strategies:

1. Use the RTG to generate several individual networks of appropriate sizes, stack them one on the other, then recursively generate random interconnects as if a lower network was one node and the interconnect endpoints are chosen randomly.

2. Use the RTG to generate a network of appropriate size, then select random outbound links to several networks of appropriate sizes, but with different topologies due to the nature of the generation process, and then use them to populate the inner areas of circles of increasing diameter stacked upon each other.

A number of issues must be resolved in either approach. One of the most important is the process for selecting the endpoints of the links connecting two networks. Another is how the nodes ar apportioned among the various networks at each level.

A main focus of the proof of concept validation experiments is how the simulator reacts to network partitioning where one or more large networks is disconnected from the remaining population for a time, then reconnected again. From past experience, we know this is the acid test of a routing algorithm, since large quantities of routing information must be shuttled over a potentially large fraction of the network routing database.

3.2 Future Plans

The extensions to support a fractal-style RTG require significant changes and upgrades not only to the RTG but also to affiliated support modules. We plan to complete the analysis, design and implementation of the revised RTG by the end of May and to continue validation and experiment with it over the summer.

4. Infrastructure

Problems continue in the deployment of a truly robust NTP subnet for CAIRN. One is the requirement for upgrade to FreeBSD 3.4 in all routers, in order to support a common suite of cryptographic schemes and radio interfaces. At this time, only isipc6 and udelpc have been upgraded.

A second problem has to do with the serial port hardware and software which supports the GPS radios at SAIC, ISI-W, SAIC and UCL. Modern UART chips include a FIFO with up to 16 stages of delay. The FIFO must be disabled in order to capture precision timestamps from the radio. A related problem common to all Unix systems known to this investigator is a software FIFO implemented in the driver, which also must be disabled. Each driver and each system seems to have a different way to do this, generally requiring a patch or kernel rebuild.

A third problem is apparently peculiar only to dartpc. When a serial port is opened and the first character received, the machine hangs up in such a way that a power-cycle reset is the only way to continue operation. Although this was first observed in an older FreeBSD version and we thought it would go away in the newer FreeBSD 3.4, apparently it has not. Until this problem is resolved, the UDel time is available only from other stratum-1 time servers pogo.udel.edu and rackety.udel.edu.

An obvious and attractive alternative to the serial port problems is to use the IRIG signal generated by the GPS radios and connect to an audio card on the router. The NTP radio clock driver suite includes one that uses classic DSP algorithms to extract timestamps from an IRIG-B signal with accuracies in the low tens of microseconds. Used with our nanokernel modifications, which are now in stock FreeBSD from 3.4, the routers could keep the clock to at least this order of accuracy. However, a generic solution to the FreeBSD audio card interface to the NTP daemon is so far elusive.

We are the happy recipients of a new TrueTime NTS-200 NTP time server with embedded GPS receiver. This is a new product from TrueTime and is so far as we know the first commercial implementation of NTPVersion 4. The implementors used an interesting strategy with an embedded Unix kernel and file system preserved on Flash memory. The approach allows the use of existing protocol modules derived from the NTP software distribution, which should greatly facilitate new feature integration, in particular the public-key cryptography support.

4.1 Future Plans

The upgrade of the CAIRN routers to FreeBSD is going slowly, but we expect to have at least a cluster of routers near the CAIRN epicenter upgraded and in service with Autokey by the end of May. The situation at UCL has not been encouraging. The GPS radio there has been out of service since the last round of testing. One problem is that the radio itself and the router are separated by a considerable distance, which considerably complicates connection of the PPS signal from the radio to the router. It is not clear what can be done about this.

5. Publications

All publications, including journal articles, symposium papers, technical reports and memoranda are now on the web at www.eecis.udel.edu/~mills. Links to the several publication lists are available on that page, as well as links to all project descriptions, status reports and briefings. All pub-

lications are available in PostScript and PDF formats. Briefings are available in HTML, PostScript, PDF and PowerPoint. The project descriptions are cross-indexed so that the various interrelationships are clearly evident. Also included are the documentation pages for various public software distributions. Links to other related projects at Delaware and elsewhere are also included on the various pages. Hopefully, the organization of these pages, which amount to a total of about 300 megabytes of information pages and reference documents, will allow quick access to the latest results and project status in a timely way.

Following is a retrospective list of papers and reports supported wholly or in part on this project and the immediately preceeding project "Scalable, High Speed, Internet Time Synchronization," DARPA Order D012. The complete text of all papers and reports, as well as project briefings, status reports and supporting materials is at www.eecis.udel.edu/~mills.

5.1 Papers

All documents listed below are available at www.eecis.udel.edu/~mills in PostScript and PDF formats. Comprehensive briefing slides are also available in PowerPoint, HTML, PostScript and PDF formats.

1. Mills, D.L. Cryptographic authentication for real-time network protocols. In: *AMS DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 45* (1999), 135-144.

2. Mills, D.L. Adaptive hybrid clock discipline algorithm for the Network Time Protocol. *IEEE/ACM Trans. Networking 6, 5* (October 1998), 505-514.

3. Li, Qiong, and D.L. Mills. On the long-range dependence of packet round-trip delays in Internet. *Proc. IEEE International Conference on Communications* (Atlanta GA, June 1998), 1185-1191.

4. Mills, D.L., A. Thyagarajan and B.C. Huffman. Internet timekeeping around the globe. *Proc. Precision Time and Time Interval (PTTI) Applications and Planning Meeting* (Long Beach CA, December 1997), 365-371.

5. Mills, D.L. Authentication scheme for distributed, ubiquitous, real-time protocols. *Proc. Advanced Telecommunications/Information Distribution Research Program (ATIRP) Conference* (College Park MD, January 1997), 293-298.

6. Mills, D.L. The network computer as precision timekeeper. *Proc. Precision Time and Time Interval (PTTI) Applications and Planning Meeting* (Reston VA, December 1996), 96-108.

7. Mills, D.L. Improved algorithms for synchronizing computer network clocks. *IEEE/ACM Trans. Networks 3, 3* (June 1995), 245-254.

5.2 Technical Reports

8. Sethi, A.S., H. Gao, and D.L. Mills. Management of the Network Time Protocol (NTP) with SNMP. Computer and Information Sciences Report 98-09, University of Delaware, November 1997, 32 pp.

9. Mills, D.L. A precision radio clock for WWV transmissions. Electrical Engineering Report 97-8-1, University of Delaware, August 1997, 25 pp.

10. Mills, D.L. Clock discipline algorithms for the Network Time Protocol Version 4. Electrical Engineering Report 97-3-3, University of Delaware, March 1997, 35 pp.

11. Mills, D.L. Proposed authentication enhancements for the Network Time Protocol version 4. Electrical Engineering Report 96-10-3, University of Delaware, October 1996, 36 pp.

12. Mills, D.L. Simple network time protocol (SNTP) version 4 for IPv4, IPv6 and OSI. Network Working Group Report RFC-2030, University of Delaware, October 1996, 18 pp.

13. Mills, D.L. Simple Network Time Protocol (SNTP). Network Working Group Report RFC-1769, University of Delaware, March 1995, 14 pp.

14. Mills, D.L. Simple Network Time Protocol (SNTP). Network Working Group Report RFC-1769, University of Delaware, March 1995, 14 pp.

5.3 Internet Drafts

15. Mills, D. L., T.S. Glassey and M.E. McNeill. Authentication Scheme Extensions to NTP. Internet Draft draft-mills-ntp-auth-coexist-01.txt, IETF, September, 1998.

16. Mogul, J., D.L. Mills, J. Brittenson, J. Stone and U. Windl. Pulse-per-Second API for UNIX-like Operating Systems, Version 1.0. Internet Draft draft-mogul-pps-api-05.txt, IETF, August 1999, 25 pp.