Survivable, Real Time Network Services

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

Quarterly Progress Report
1 October 1999 - 31 December 1999

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. NTP Version 4

Work continued on the NTP Version 4 reference implementation and distribution for Unix, VMS and Windows. We have incorporated numerous patches required by the over two dozen ports of the code to various Unix architectures and operating systems. Following is a partial list of improvements during the reporting period.

1. The autoconfigure system has been further enhance to automatically detect and build the RSA rsaref20 package of cryptographic routines. The current design allows this package or offshore equivalent to be incorporated in the NTP daemon without change. The only significant problem we have detected is that the DES routine does not work correctly on a little-endian machine.

2. Two new radio clock drivers were designed and implemented, both using the audio codec of a Sun workstation and a conventional shortwave receiver. One of these is for the transmissions from NIST time/frequency stations WWV in Colorado and WWVH in Hawaii. The other is for the transmissions from the Canadian time/frequency station CHU. Both make extensive

use of digital signal processing technology and principles of optimum receiver design. The intent of these projects is to demonstrate that a high performance primary NTP server can be built using only an inexpensive shortwave receiver and a low end workstation.

3. Extensive preparations for the century rollover were complete well before the actual event. This investigator activated a fleet of time tellers and watchers with almost every national time service in the world. The recorders were running at the instant of roll and mostly business as usual. The interesting exception was the Automatec Computer Time Service operated by NIST. ACTS uses a bank of telephone modems to provide a computer readable timecode on request. The service is used by the stock exchanges to timestamp transactions, for instance. The problem was the ACTS firmware, which incorrectly handled the roll and affected the network of NIST time servers on the Internet. Alert timekeeper Judah Levine at NIST quickly discovered the problem and shut down the NTP servers until the problem was fixed.

4. NIST has entered into an agreement with Certified Time, Inc., in which NTP time servers operated by NIST will be equipped with cesium oscillators and means to provide a cryptographically authenticated timestamping service using NTP as the delivery vehicle. The necessary cryptographic data will be incorporated in the NTP extension fields which have been designed for this service.

5. A junkbox PC was refurbished and sprinkled with FreeBSD 3.4 in order to explore and shake down bugs in the pending upgrade of the CAIRN router software to that version. In addition, this machine has proved highly useful in exposing bugs peculiar to the little-endian Intel architecture.

3. NTP Autokey

Much of the effort during the reporting period was involved with the autokey facility, which provides a cryptographically secure authentication function for the various protocol modes supported by NTP. As this facility is nearing maturity, a more detailed description of its operation is in order.

Authentication support allows the NTP client to verify that the server is in fact known and trusted and not an intruder intending accidentally or on purpose to masquerade as that server. The NTPv3 specification RFC-1305 defines a scheme which provides cryptographic authentication of received NTP packets. Originally, this was done using the Data Encryption Standard (DES) algorithm operating in Cipher Block Chaining (CBC) mode, commonly called DES-CBC. Subsequently, this scheme was augmented by the RSA Message Digest 5 (MD5) algorithm using a private key, commonly called keyed-MD5. Either algorithm computes a message digest, or one-way hash, which can be used to verify the server has the correct private key and key identifier. NTPv4 retains this scheme and, in addition, provides a new *autokey* scheme based on reverse hashing and public key cryptography. Authentication can be configured separately for each association.

The authentication options specify the suite of keys, select the key for each configured association and manage the configuration operations, as described below. The flag which controls these functions can be set or reset by configuration commands and also by remote configuration commands sent by a control program running in another machine. If the flag is enabled, persistent peer associations and remote configuration commands are effective only if cryptographically authenticated. If this flag is disabled, these operations are effective even if not cryptographic authenticated. It should be understood that operating in the latter mode invites a significant vulnerability where a rogue hacker can seriously disrupt client timekeeping.

The flag affects all authentication procedures described below; however, it operates differently if cryptographic support is compiled in the distribution. If this support is available and the flag is enabled, then persistent associations are mobilized and remote configuration commands are effective only if successfully authenticated. If the support is unavailable and the flag is enabled, then it is not possible under any conditions to mobilize persistent associations or respond to remote configuration commands. The flag normally defaults to enable if cryptographic support is available and to disable otherwise.

With the above vulnerabilities in mind, it is desirable to set the flag in all cases. One aspect which is often confusing is the name resolution process which maps server names in the configuration file to IP addresses. In order to protect against bogus name server messages, this process is authenticated using an internally generated key which is normally invisible to the user. However, if cryptographic support is unavailable and the flag is enabled, the name resolution process will fail. This can be avoided either by specifying IP addresses instead of host names, which is generally inadvisable, or by leaving the flag disabled and enabling it once the name resolution process is complete.

3.1 Private Key Scheme

The original RFC-1305 specification allows any one of possibly 65,536 keys, each distinguished a 32-bit key identifier, to authenticate an association. The servers involved must agree on the key and key identifier to authenticate their messages. Keys and related information are specified in a key file which must be distributed and stored using secure procedures beyond the scope of the NTP protocol itself. Besides the keys used for ordinary NTP associations, additional ones can be used as passwords for the utility programs.

When the NTP daemon is first started, it reads the key file and installs the keys in the key cache. However, the keys must be activated before they can be used with a configuration command or command from a remote control program. This allows, for instance, the installation of possibly several batches of keys and then activating or inactivating each batch remotely using the control program. This also provides a revocation capability that can be used if a key becomes compromised. A special command selects the key used as the password for the utility programs.

3.2 Autokey Schemes

The original NTPv3 authentication scheme described in RFC-1305 continues to be supported. In NTPv4 the autokey is available, with or without public-key cryptography. It operates much like the S-KEY scheme, in that a session key list is constructed and the entries used in reverse order. The scheme is specifically designed for multicast modes, where clients normally do not send messages to the server. In these modes, the server uses the scheme to generate a list of key identifiers from a random initial value by repeated hashing of a session key. Each entry on the list includes the source and destination addresses, a public value and a private value known only to the list generator. The list is used in reverse order to generate a unique session key for each message sent. The client regenerates the session key and verifies that the hash matches the previous session key.

For use in multicast modes the private value of the session key is set to zero, so any receiver can authenticate the message and follow the hash sequence to the initial value. Therefore, each message contains the public values binding the session key to the initial value, but these values need to be verified only when the server generates a new key list or more than four server messages have been lost. This is done using a command/response protocol described below to retrieve the initial sequence number, key and public-key signature.

A variant of the scheme is used for client/server and symmetric-peer modes. In these modes the client generates a key list as in the multicast modes, but includes a private value or cookie when generating the hash sequence. The cookie is generated as the hash of a session key derived from the client and server addreses and a random value known only to the server and never divulged. A client learns the cookie value and public-key signature using the command/response protocol. While it is possible for an intruder to intercept the cookie and generate spurious messages, these messages cannot be used to spoof other than the legitimate server. Even in this case, the hash sequence makes it highly unlikely that a replay or spoof would be acceptable to the client.

When used with symmetric-peer modes, each peer operates as in client/server mode but independently of the other, except that the private value used to generate the key list and authenticate the source is developed using the Diffie-Hellman key agreement algorithm augmented by public-key signatures. In addition and since persistent associations are mobilized for symmetric modes, the integrity of the key list is verified by matching the hash of the current key identifier to the previous key identifier.

3.3 Extension Fields

The autokey schemes require no change to the NTP packet header format or message authentication code (MAC); however, one or more optional extensions fields can be inserted between the header and the MAC. The extension fields are used by the command/response protocol to exchange cryptographic values, including the initial sequence number and key for multicast modes and the cookie used in client/server and symmetric modes. Since packets containing extension fields must be authenticated without requiring private values or key agreement algorithms, the session keys are regenerated with a private value of zero.

The extension field formats are defined in Internet Draft draft-NTP-auth-coexist-00.txt. The MAC itself is constructed in the same way as NTPv3, but using the original NTP header and the extensions field padded to a 64-bit boundary. Each new public value is encrypted by the host private value. It is the intent of the design, not yet finalized, that the public value, encrypted public value, public key and certificate be embedded in the extension fields where the client can decrypt as needed. However, the relatively expensive encryption and decryption operations are necessary only when the public value is changed.

Note that both the original NTPv3 authentication scheme and the new NTPv4 autokey scheme operate separately for each configured association, so there may be several session key lists operating independently at the same time. Since all keys, including session keys, occupy the same key cache, provisions have been made to avoid collisions, where some random roll happens to collide with another already generated. Since something like four billion different session key identifiers are available, the chances are small that this might happen. If it happens during generation, the generator terminates the current session key list. By the time the next list is generated, the collided key will probably have been expired or revoked.

While permanent keys have lifetimes that expire only when manually revoked, random session keys have a lifetime specified at the time of generation. When generating a key list for an association, the lifetime of each key is set to expire one poll interval later than it is scheduled to be used. The maximum lifetime of any key in the list is specified by the autokey command. Lifetime enforcement is a backup to the normal procedure that revokes the last-used key at the time the next key on the key list is used.

3.4 Public-Key Cryptography

The autokey schemes are vulnerable to a man-in-the-middle attack, where an intruder can intercept messages and prevent direct communication between the server and client. While considered relatively unlikely in a properly constructed network, this hazard cannot be neglected. These attacks can be deflected using public-key cryptography and RSA-based digital signatures. Provisions for public-key cryptography have been implemented in the NTP daemon using the RSA Laboratories rsaref20 software package, which is available from many sources in the US and Europe. When the NTP distribution is compiled with this package several new commands and command options are available.

The primary function of public-key cryptography is to sign and verify the cryptographic values exchanged between a sender and receiver, including the initial sequence number and key in multicast modes and the cookie in client/server and symmetric modes. The sender signs these values using its private key, which is generated by the sender and never divulged. The receiver verifies the signature using the public key of the sender as provided by insecure means.

It the present stage of implementation, public-key cryptography is available only in client/server and symmetric active modes. In these modes the public key is installed when the association is configured at startup. The multicast and symmetric passive modes require a mechanism to obtain the public key on-the-fly when a new client shows up. This is to be provided in future using a reverse-DNS lookup using the IP address of the sender.

Public key management is implemented in much the same way as that used by the ssh facility. A public/private key pair is generated by the special program, which also generates private MD5 keys and Diffie-Hellman parameters. This program generates four files: ntp.keys containing the DES/MD5 private keys, ntpkey containing the RSA private key, and ntpkey_*host* containing the RSA public key, where host is the DNS name of the local machine. The fourth file contains the Diffie-Hellman prime modulus and generator. All four files contain randomly generated data seeded by the system clock.

4. CAIRN Router Timekeeping

Three Spectracom GPS receivers have been installed at Science Applications International Corporation (SAIC), Information Science Institute (ISI-West) and University College London (UCL). These have been connected to the CAIRN routers at these locations which run the latest version of NTP. Surprisingly, the performance of these routers as NTP primary servers is so far unacceptable. Rather than achieving accuracies generally better than 100 μs as expected, random and systematic errors were as high as 10 ms in some cases.

Among the problems found are some familiar ones. The radio connection uses a conventional serial port operating at 9600 bps. Ordinarily, the hardware jitter with such a connection is less than one bit period, or about 100 μs. However, recent implementations of the 16450 UART chip include a FIFO of up to 16 stages, in order to reduce the interrupt load on the system. In the worst case, this could result in a character delay of about 15 ms, clearly unacceptable for good timekeeping. Recent versions of FreeBSD include provisions to disable the FIFO and this has been done on some, but not all the routers.

Another problem is in the software driver for the UART chip. Again in order to minimize the interrupt load, the driver includes a software FIFO that allows the character stream to be batched for deliver to upper level software. Unfortunately, none of the extant operating systems of today, including FreeBSD, contain provisions to disable the software FIFO. This investigator has had to inspect and modify the UART driver in each and every operating system and upgrade used in our laboratory for the last 14 years.

Still another problem is apparent in at least some of the CAIRN routers at the three sites and the University of Delaware. The master clock ensemble at our laboratory consists of dual redundant Spectracom GPS receivers backed up by dual redundant Spectracom WWVB receivers and would ordinarily be considered an extremely solid time source. Any of these radios can be connected using either the ASCII timecode and a serial port and or the audio IRIG signal and a sound card in the router. Considering the serial port problems described above, it would be highly desirable to use the sound card. However, after extensive investigation and experiment, it has not been possible to do that in the FreeBSD 2.2.5 version now running in the routers.

To compound our troubles, it has not been possible to use the serial port on the UDel router, as the first character sent to the router causes the hardware to freeze and requires a complete hardware reset and system reboot to get it back on the air. This problem has not occurred at the other three sites, even though the NTP configuration, kernel configuration and hardware are all identical. Further resolution of all problems has been put on hold until the FreeBSD version 3.4 has been installed in all routers.

5. Miscellany

The University of Delaware Abilene connection has been turned up giving the campus access via a 155-Mbps connection to Qwest. Subsequently, the 10-Mbps fiber link between the DCnet research machines and the Abilene router was upgraded to 100 Mbps and a new high speed router donated by Torrent was installed. A series of tests confirmed that the diameter of the pipes between the DCnet machines and the vBNS router which serves ISI-W is at least 40 Mbps and limited only by the crude measurement software used. A more detailed characterization will be done as time permits.

As reported earlier, one of our three cesium oscillators has failed the beam tube, which is at the heart of the instrument. This particular instrument was donated by the U.S. Naval Observatory and has already had one tube replacement. In order to reduce the cost of replacement, a used tube was provided. A new tube has a lifetime of about ten years; apparently, the used tube was near end life and failed after only about one year of operation. The instrument was sent for repair and provided with a used tube after a search to find the best of a lot of used tubes available at the repair facility.

6. Personnel

Tamal Basu joined the project in September 1999. He is currently funded half time by the DARPA project and half time by another NSF project. He is to continue the work initiated by Mr. Redwinski; in particular, extending the analytical model and routing protocol test suite. Qiong Li is finishing up his dissertation and expects to graduate in the Spring semester of 2000, our first Millennium Baby.

Ajit Thyagarajan continues work to complete his dissertation. His topic is the analytical and experimental study of autoconfigure algorithms suitable for deployment in a survivable internet. He is finishing up extensions of the current centralized algorithms to operate in a distributed context. He expects to complete all requirements by the end of the Fall semester 1999.

7. Meetings

This investigator was invited to present a briefing on NTP at a symposium at the University of Pusan in Korea at no cost to the Government. Unfortunately, he contracted a serious throat infection and fever prior to the scheduled visit and had to cancel at the last minute. Complications from the infection required total vocal abstinence for most of the next two months. The visit has so far not been rescheduled.

8. Plans for Next Quarter

High on the list of activities for the next quarter is finishing up the autokey code, in particular overhauling the DNS resolver code to provide a DNS name and possibly certificates when provided the IP address of a multicast or manycase server. Additional work needs to be done to verify that the protocols work in all combinations of modes and persistent/ephemeral associations. This is a significant task, since the combinations of modes, various failure abd recovery scenarios is quite complicated.

The current effort to complete the autokey implementation is preliminary to the partially implemented autoconfigure facility, which is the immediate goal of the current research program. The autokey experience was much more intricate than first anticipated; however, the experience has provided new understanding of the autoconfigure issues and allow more rapid progress in the final design and implementation phase of the project.

9. 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 publications 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.

9.1 Papers

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.

9.2 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.

9.3 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.