NTPv4 Specification Update

David L. Mills
University of Delaware
http://www.eecis.udel.edu/~mills
mailto:mills@udel.edu

From NBS Special Publication 432 (1979 edition, now out of print)

Why are we roosting here?

- The Network Time Protocol (NTP) has evolved from humble beginnings over two decades and five versions to the NTP Version 4 of today.
- NTP of one version or another is now deployed in millions of clients in just about every computer that can be connected to the Internet.
- Thousands of NTP public primary (stratum 1) servers are scattered all over the globe, some operated by the national standards laboratories of countries in all continents, including Antarctica and soon on Mars.
- The NTP current standards landscape includes NTPv3, documented in RFC-1305 and SNTPv4, documented in RFC-2030. Neither of these is at full standards status.
- There is critical need to update RFC-1305 to reflect the current NTPv4 architecture, protocol and algorithms.
- There is critical need to update RFC-2030 to reflect current best practices and avoid flooding attacks as sustained by U Wisconsin, NIST and USNO.

Agenda for a flock of birds

- We need to separate the specifcation issue from the reference implementation, although both have evolved together.
- The NTPv4 packet header is identical to the NTPv3 packet header with the following exceptions.
  - The reference identifier field has been changed to support IP addresses longer than 32 bits and in certain cases to show an error message.
  - The header syntax now includes one or more optional extension fields used by the new public key authentication scheme.
- A number of protocol improvements have been made for enhanced security, provisions for IPv6 and algorithm refinement. These will be described later.
- The NTPv4 reference implementation (ntpd) has evolved considerably since the NTPv3 implementation (xntpd), both to correct errors, enhance performance and support new functionality. While not in an of itself the subject of specification, it is an important component in the specification refinement and validation process.

Protocol and algorithm refinements

- Reference identifier semantics have been changed to support addresses longer than 32 bits, but without change in functionality.
  - With addresses longer than 32 bits, a MD5 hash of the address is used instead of the IP address itself.
  - At stratum 15 and above the field may contain a four-octet information or error message.
- The clock discipline algorithm has been redesigned for fast initial response to large frequency errors and for improved stability with long poll intevals.
- The clock filter and selection algorithms have been redesigned to improve performance, especially with very fast processors and networks.
- New burst modes have been added to speed initial frequency adaptation and reduce jitter.
- Repetitive timer operations have been randomized to avoid bunching.

Reverence implementation refinements

- All computations except raw timestamp differences use floating-double arithmetic. This resolves, for now, the “34-year” rollover problem.
- Certain ambiguities in the NTPv3 clock filter algorithm and timestamp calculations have been resolved.
- The annoying and misleading “virtual time” used in NTPv3 has been removed. This avoids cases where the system time appeares to be correct, but the actual time could be substantially different.

Other new features

- The suite of reference clock drivers has been expanded to include virtually all radio, satellite and modem services available anywhere.
  - A set of audio drivers has been added to support IRIG signals and shortwave time signals from US and Canadian radio stations.
- The kiss-o'death (KoD) packet is used to provide useful diagnostic information to clients, as well as an access control mechanism to suppress traffic incompatible with the server security model.
- A call-gap mechanism is provided to detect and suppress flooding attacks from ill-conceived client implementations.
  - It uses an LRU stack with probabilistic preemption.
  - Upon detection of a flood, a (rate controlled) KoD packet is returned.
  - Compliant implementations will cease operation if a KoD packet is received.
  - Call gap is now in use at UDel, NIST and USNO with varying degrees of success.

Enhanced system clock resolution

- The new Nanokernel kernel modifications provide nanosecond system clock resolution. It replaces the original Microkernel, which provides resolution limited to one microsecond.
  - This is useful, since modern workstations and PCs can cycle through the kernel and return the current time in less than one microsecond.
- The Nanokernel modifications are now available in FreeBSD and Linux kernels. They provide enhanced resolution and support for the pulse-per-second (PPS) signal available with some reference clocks.
  - The NTPv4 operating system interface has been rebuilt to support the Nanosecond kernel and PPS signal.
- The PPS signal driver has been upgraded to support the PPS API interface available on most operating systems, including FreeBSD, Linux, Solaris, SunOS and Alpha.
  - With this driver and a PPS signal from a good GPS receiver, NTPv4 precision is routinely better than one microsecond.

Cryptographic authentication

- Traditional symmetric key cryptography continues to be supported.
  - Only the MD5 message digest alrogithm is supported; the DES-CBC algorithm is toast, mainly due to pesky Government export rules.
  - MD5 is now available in several commercial products and in use (for fee) by public servers operated by national governments (not US).
- A new security model and authentication protocol based on public key cryptography is now available.
  - The Autokey security model and authentication protocol is specially designed for public time servers with large client populations.
  - Identity keys for authentic security compartments can now be retrieved using a secure web at ISC.
  - Autokey has been in regular operation at Udel, USNO and ISC, but not yet widely deployed.
  - A comprehensive specification document is available in PDF at http://www.eecis.udel.edu/~mills/database/reports/stime/stime.pdf.

New public key cryptographic authentication (Autokey)

- The Autokey security model and authentication protocol is designed to authenticate servers to clients.
  - Autokey is based on public key cryptography augmented with zero-knowledge identity proofs.
  - The security model provides multiple overlapping security compartments.
  - The implementation uses the OpenSSL cryptographic library and is conmpatible with the current PKI infrastructure.
  - The algorithms are specially designed to minimize resources with large client populations and to avoid flooding and middleman attacks. After an initial exchange, protected packets carry no additional overhead.
  - Retrieving and refreshing cryptographic media is completely automatic and requires no operator intervention.
  - Initial setup is simple using the provided key/certificate generator program. Certificates are compatible with PKI and industry standards.
  - Autokey can also be used to retrieve the leap-second table where available.

New autonomous configuration (Autoconfigure)

- A new configuration scheme called Autoconfigure has been designed, implemented and tested.
  - It uses an expanding ring search with the usual metric, timeout and refresh mechanisms.
  - It does not use the traditional anycast paradigm, which trolls for a single server in the nearby neighborhood. It uses the manycast paradigm where clients troll for a plurality of servers, then trim the respondents using the NTP mitigation algorithms until the best three are left
  - This works well for relatively small networks with a modest population of servers and in cases where extreme reliability is required
  - The present design produces mostly flat hierarchies; further development is needed for deeper forests and where servers engage in a whispering campaign.
  - The scheme can be adapted to the new pool.ntp.org scheme, where multiple servers are randomized in DNS replies.

New NTP simulator

- The NTPv4 software distribution includes a simulation environment useful for testing and evaluation.
- The simulation environment provides the same operating system infrastructure as traditional Unix operating systems.
- Network and operating system latencies are simulated using synthetic but realistic noise generators or with data files collected during regular operation.
- The daemon operates in the same way and with all algorithms in vivo and in vitro with the simulated operating system interface.
- The simulator has been highly useful for test and evaluation in scenarious where in vivo testing is difficult or impossible.

Currently available literature

- An extensive suite of slide presentations is available on the NTP project page http://www.eecis.udel.edu/~mills/ntp.html.
  - It includes overview, architecture, protocol and algorithms.
  - The algorithms are documented by a set of flow charts.
  - Several white papers and executive summaries explore issues such as the NTP timescale and era numbering, 2036 rollover and 34-year ambiguity.
  - Extended discussion of the Autokey and Autoconfigure schemes.
- The SNTP Internet Draft draft-mills-sntp-v4-00.txt has been on the RFC Editor's queue for over a year.
- The NTP Autokey security model and protocol has been documented as an Internet Draft suitable for standards track. It is in PDF with many necessary equations and not suitable for Postel ASCII formating.
- The NTPv4 reference implementation and documentation is available at www.ntp.org and has been widely deployed, but not yet adopted by all operating system providers.

Bird migration

- Every effort should be ignited to have the SNTP draft, now stalled on the RFC Editor’s queue, advanced at least to information status. This is to help avoid the stupid implementations now flooding NIST and USNO. See ftp://ftp.rfc-editor.org/in-notes/internet-drafts/draft-mills-sntp-v4-00.txt.
- A protocol document specifing only the raw protocol state machine and transition functions should be created and advanced along the standards track. In truth, this would be the SNTP document as amended and enhanced with the protocol features documented herein.
- A new document describing the algorithms necessary for a fully compliant NTP server should be prepared. This would use a flowchart oriented approach rather than the code segments used in RFC-1305.
- The security scheme is best described in a separate document. A starting point is the existing PDF document at http://www.eecis.udel.edu/~mills/database/reports/stime/stime.pdf.

Enhancements to include or publish as informational RFCs

- Cal-gap for flooding defense
- Autoconfigure and Manycast schemes
- Pool.ntp.org autoconfigure scheme
- Reference clock interface and PPS support
- Who knows what else…

Further information

- NTP home page http://www.ntp.org
  - Current NTP Version 4 software and documentation
  - FAQ and links to other sources and interesting places
- David L. Mills home page http://www.eecis.udel.edu/~mills
  - Papers, reports and memoranda in PostScript and PDF formats
  - Briefings in HTML, PostScript, PowerPoint and PDF formats
  - Collaboration resources hardware, software and documentation
  - Songs, photo galleries and after-dinner speech scripts
- Udel FTP server: ftp://ftp.udel.edu/pub/ntp
  - Current NTP Version software, documentation and support
  - Collaboration resources and junkbox
- Related projects http://www.eecis.udel.edu/~mills/status.htm
  - Current research project descriptions and briefings