# Advances in Computer Network Timekeeping

National Science Foundation

NSF Division of Network and Communications Research and Infrastructure
Grant NCR-93-01002 (June 1993 - June 1996

## Annual Progress Report

For the period 1 June 1994 - 31 May 1995

David L. Mills
Electrical Engineering Department
University of Delaware
8 July 1995

### 1. Introduction

This annual report covers work done in support of our program of computer network timekeeping technology and is designed to complement related work funded by ARPA and the U.S. Navy. Contributors to this effort include Prof. David L. Mills and graduate students Kenneth Monington and Ajit Thyagarajan. Collaborators in this project include Prof. Judah Levine, of the University of Colorado and NIST Boulder, as well as many keepers of network clocks around the world. Specific applications benefiting from this research include multicast services such as multimedia services, real-time conferencing and management of distributed, real-time experiments.

Specific products of this research include the Network Time Protocol (NTP) architecture and protocol, which has been refined through four versions and several years of development and is now widely deployed in the global Internet. The algorithms used in NTP have been specifically engineered for good performance in networks ranging from 9600-bps dial-up links to high speed networks such as SONET and ATM. The present software implementation of NTP is a joint product of over two dozen contributors in the U.S., Europe, Japan and Australia. It has been ported to over 40 server and workstation systems using almost every architecture and operating system on the market today.

We have developed engineered algorithms for NTP which can achieve typical accuracies of a few tens of milliseconds in the sometimes congested global Internet of today and less than a millisecond with fast LANs and modern RISC-based workstations without modification. The algorithms exploit redundant servers and diverse network paths to achieve a high degree of reliability, even in cases where some servers may provide incorrect information or some network resources may fail.

We have worked with workstation vendors to provide precision timekeeping functions with widely used Unix kernels. To this end, Sun, Digital and Hewlett Packard have donated workstations and kernel sources and in return have been given specific kernel software enhancements which they distribute to their customers who need precision time. We have also developed kernel enhancements which utilize signals provided by precision radio and satellite timing receivers in order to discipline the workstation clock to these sources. Using the enhanced kernels and these signals, we have demonstrated that reliable synchronization can be improved to some tens of microseconds in modern RISC-based workstations [MIL94a].

The project also involves the design and construction of precision timekeeping software and hardware that provide improved accuracy and stability consistent with the capabilities of the time dissemination systems operated by various government agencies, including NIST, USNO and various foreign equivalents. Software drivers for the NTP implementation have been constructed for over 25 radio, satellite and telephone receivers using every national standard time dissemination service known to exist in the world. We have developed a relatively inexpensive precision timing receiver based on the LORAN-C radionavigation system, which provides submicrosecond timing accuracies. We have also developed a prototype precision oscillator and interface for a commonly available workstation.

Finally, the project involves the conduct of experiments designed to evaluate the success of the research and assist technology transfer to computer manufacturers and network providers. We have assisted personnel at USNO and NIST in the establishment of primary time servers synchronized to U.S. national time standards. We have worked with ANS personnel in the design and configuration of NTP time services for the former NSFnet backbone network. We have participated in experiments involving the ARPA DARTnet network testbed and real-time multimedia conferencing, as well as experiments in international multicast time synchronization delivery.

## 2. Accomplishments in the Last Year

### 2.1. Improved Algorithms for NTP

As a result of collaboration with Judah Levine of the University of Colorado and NIST Boulder, we have overhauled several of the algorithms used in the NTP implementation, including the basic algorithm that disciplines the local system clock time and frequency [MIL94c], [MIL95b]. Specifically, the original algorithms were designed to operate with update intervals from one to about 17 minutes, depending on measured clock stability and network congestion state. The design is based on a feedback loop in which the clock oscillator time and frequency is continuously adjusted to track a reference signal determined from update messages exchanged between the local host and its peers.

The original design functions as a phase-lock loop (PLL), in which the reference signal is calculated from the time error determined from the updates; however, this was found not to work well with progressively longer update intervals much over 17 minutes. However, there are some applications where update intervals need to be much longer, such as with the NIST and USNO telephone time and German PTB telephone time services.

As the result of a series of analysis and simulation exercises [MIL94a], we concluded that the PLL design is not appropriate for update intervals much larger than 17 minutes and that a design suggested by Judah Levine and called a frequency-lock loop (FLL) is more appropriate in this regime. The new design is a hybrid PLL/FLL one in which the PLL is used at and below 17-minute update interval and the FLL used above that.

The new algorithm was tested over a period of several months, during which machines were rebooted, telephones reconfigured and the NIST telephone service itself disrupted (which was first detected by NTP, in fact). The results show that the system clock can be reliably synchronized to within a few tens of milliseconds with update intervals as long as several hours, and most of the time to within a few milliseconds. The NTP implementation was updated to include this feature, which is now incorporated in the public distribution.

## 2.2. Scalable Architectures and Protocols for Large Scale Network Services

There are inherent problems in achieving reliable and stable time synchronization in very large synchronization subnets of several hundred thousand hosts or more. There are several issues which have to be taken into account in designing such a system, including network failures, availability of servers, and suitable authentication mechanisms. Building such a network of time servers and clients has so far required a good deal of configuration engineering and not a little black art. In a system that is continuously growing and changing, devising an accurate and stable configuration, while ensuring the reliability of the system as a whole, is becoming an increasingly arduous task.

This very problem has long been studied in the engineering of digital telephone networks consisting of primary, secondary, and higher strata of synchronization sources. The problem here is to determine the locations of suitable primary sources for synchronization, so as to minimize errors and avoid common points of failure. Traditional methods using integer-programming techniques have been shown to break down with just a few hundred nodes in the network. A number of heuristics have been proposed in the literature, such as Lagrangian Relaxation, Sub-gradient Optimization, and branch-and-bound algorithms. Most of them however are not scalable to the degree that we have formulated in the problem. We are currently experimenting with a modified Add/Drop algorithm which seems to have the desired scalability features and expect to obtain the results of our experiments shortly.

The problem we have addressed is how a network of q million or more nodes distributed on a global scale can organize itself according to a common metric and set of constraints. For example, the existing NTP subnet is believed to have at least 100,000 servers and clients and may soon have over a million. The subnet now has multiple subtrees, each routed on one of about 100 primary source of time synchronized by radio, satellite or telephone. Each of these primary servers provides synchronization in some cases to over 500 secondary clients at various institutions, which in turn provide tertiary synchronization to several hundred more client workstations. A quality metric, called the synchronization distance, is developed by NTP as the result of individual measurements with participating peers. The goal of the algorithms is to select peers at each level of the subnet tree so as to realize the minimum distance subnet subject to constraints, such as the maximum number of clients for each server.

We are also studying distributed approaches for the ubiquitous and autonomous configuration of time servers and clients in very large networks. The overall problem can be decomposed into several sub-problems, each of which is a challenging one by itself. One such issue that we are concentrating on has to do with the autonomous configuration of a set of time servers and clients into an optimal self-organizing hierarchy. From a theoretical standpoint, this belongs to a class of NP-Complete problems (the dominating set) for which there exist no good approximations. A modified branch-and-bound algorithm seems to hold good promise here. We are currently studying these issues using simulation techniques and expect to implement them shortly as part of a NTP implementation overhaul.

Graduate student Ajit Thyagarajan is working in this area, in particular, on techniques based on graph theory and operations research as applied to the formation and configuration of decentralized, widely distributed, autonomously configured, network services such as NTP, DNS and multicast services. Currently, this involves multicast technology and its application to timekeeping systems, among others. In collaboration with Steve Deering of Xerox PARC, he designed and implemented an enhancement to the popular Distance Vector Multicast Routing Protocol (DVMRP) which avoids

traffic over Internet links to networks where there are no listeners [THY95a]. It operates by pruning the multicast flooding tree using information provided by the leaf routers. Additional work with undergraduate student Bradley Cain has resulted in a proposed new version of the Internet Group Management Protocol (IGMP) which utilizes this enhancement [CAI95].

### 2.3. Intelligent Hardware Clock for the SBus

Graduate students Tyrone Thompson and Don Nelson are completing Masters theses on the design and implementation of precision clock hardware for the SBus. This work involves circuit schematic design, circuit board layout and Xilinx gate array programming. A prototype implementation of a 64-bit clock in Unix timeval format has been completed and tested. It provides precision time and frequency steering by the kernel, and can be read directly in user space as an atomic operation by system and application programs.

However, the function of this hardware as a precision clock is only one facet of the intended design. The board includes two large Xilinx gate arrays, two large FIFOs, a pair of TAXI chips, and photonic transceivers. In the target application, the board is to function as a host interface for our Highball network funded by ARPA. Tyrone Thompson plans to implement the additional functionality in conjunction with a Masters thesis project. As part of this effort, he is test-driving and evaluating a new comprehensive computer-aided design package from Cadence.

### 2.4. Precision Time Synchronization for Symmetric Multiprocessor Systems

An interesting issue is how to synchronize time in a computer system involving multiple, partially autonomous processors. The symmetric multiprocessor systems (SMP) now becoming available involve a number from two to 16 of fast RISC chips such as the SuperSPARC and Digital Alpha working in a shared memory model. In SMP systems, each processor has the capability to implement an independent system clock, but these clocks may not all run at exactly the same frequency. In current kernel implementations, one of these processors is selected/elected to maintain the system clock and the others processor-specific clocks are used only as interval timers for threads assigned those processors.

A drawback to this approach is that the granularity of the system clock is limited to the interval between hardware clock interrupts, commonly called the tick, which ranges form one to ten milliseconds in current workstations. However, the clock hardware associated with each CPU can achieve far better granularities, to the order of a few nanoseconds with the Alpha chip, for example. Accordingly, a means is needed to coordinate, perhaps herd is a better word, the clocks of each processor. In principle, this is the same problem addressed by NTP and, indeed, one might conjecture a workable solution is to run NTP on each processor separately. However, a full implementation of NTP would be impractical in the kernel.

We have developed modifications for the Digital OSF/1 kernel that support the SMP synchronization function in a fashion that might be described as stripped-down NTP [MIL94d]. These enhancements are based on the precision time kernel modifications previously developed [MIL94b] and have been tested in a number of single processor systems for compatibility. They have been incorporated in the current release of the precision time kernel enhancements; however, so far they have not been activated. In the near future, we expect to test and evaluate these enhancements using a two-processor SPARC20 donated by Sun Microsystems and with access to a two-processor Alpha on the Digital corporate network.

### 2.5. Other Activities

Work on the Highball network architecture, protocols and implementation is winding down; however, some publications remain in the pipeline [BON94], [THY95c]. As the result of questions from companies with products based on NTP (Cisco, Bancomm, Hewlett Packard, among others), RFC-1361 was revised and clarified as RFC-1769 [MIL95a].

Graduate student Ken Monington completed his dissertation and defended it in late May. He is now polishing up the final draft and expects to complete it by the end of August. His research is in the area of reliable time synchronization for distributed networks in which participating clocks may not be reliable or consistent. He has developed algorithms which exploit consistency, redundancy and diversity to detect and discard faulty clocks. His plans for next year include a postdoctoral appointment with Prof. Judah Levine.

Undergraduate student Brian Huffman is working on systems programming chores with our donated HP 9000/735 workstation. His primary assignment is to develop a search strategy and shell scripts in order to survey and map the existing NTP subnet.

## 3. Education and Human Resources Development

As consistent with the teaching mission of this University, we continue to train graduate and undergraduate students in computer networking technology in general and the arcane mysteries of network time synchronization in particular. Besides our funded mission in computer network clock synchronization technology, several other students are pursuing individual agendas relating in one form or another to this mission.

Beginning graduate student Marie Conte submitted her undergraduate honors thesis on wireless networks in partial fulfillment of the requirements for the E.E. Degree with Distinction. She is continuing these studies during the Summer in our laboratory with the assignment of writing a paper for publication based on her work. She will be attending the University of Illinois in the fall and hopes to pursue a career in wireless networking.

Undergraduate honors student Tyler Saunders studied during the Spring semester on technology suitable for voice encoding and transmission. We expect undergraduate exchange student Jeffrey Lo, from Imperial College, London, to arrive later in the Summer. He will be marinating in our laboratory until Fall.

## 4. Plans for the Next Year

Our plans for next year follow largely on the progress we have made so far. We believe the most important contributions will be in the area of scalable networks; in particular, development of technology that supports ubiquitous, large scale, autonomously configured network services such as time synchronization and name resolution. We plan to study technology used in other areas, such as warehouse distribution algorithms, and adapt the technology to packet network architectures and protocols. We plan to participate in and contribute to ongoing multicast technology and demonstration programs, with special attention to applications such as these.

We plan also to continue the development of NTP Version 4 protocol and its reference implementation, adapted from the current Version 3 implementation. This is to serve as first a development and proof-of-concept tool and later refined to an operational state and deployed throughout the Internet. We expect to make use of the technology developed in Ken Monington's dissertation as

adapted to NTP. We plan also to continue development of the SMP code and to continue testing and integration of the Highball interface for the SBus.

Coincident with the analytical and development work, we plan to test ideas generated by this work using simulation techniques. We have two network simulators in use now, one a commercial package, the other developed at the MIT Laboratory for Computer Science. We are now evaluating the MARS routing algorithm simulator developed at the University of Maryland for and may make use of it as well.

## 5. Publications

### 5.1. Publications (with abstracts) in 1994

[MIL94a] Mills, D.L. Precision synchronization of computer network clocks. *ACM Computer Communication Review 24, 2* (April 1994). 16 pp.

This paper builds on previous work involving the Network Time Protocol, which is used to synchronize computer clocks in the Internet. It describes a series of incremental improvements in system hardware and software which result in significantly better accuracy and stability, especially in primary time servers directly synchronized to radio or satellite time services. These improvements include novel interfacing techniques and operating system features. The goal in this effort is to improve the synchronization accuracy for fast computers and networks from the tens of milliseconds regime of the present technology to the submillisecond regime of the future.

In order to assess how well these improvements work, a series of experiments is described in which the error contributions of various modern Unix system hardware and software components are calibrated. These experiments define the accuracy and stability expectations of the computer clock and establish its design parameters with respect to time and frequency error tolerances. The paper concludes that submillisecond accuracies are indeed practical, but that further improvements will be possible only through the use of temperature-compensated local clock oscillators.

[MIL94b]Mills, D.L. A kernel model for precision timekeeping. ARPA Network Working Group Report RFC-1589, University of Delaware, March 1994. 31 pp.

This memorandum describes an engineering model which implements a precision time-of-day function for a generic operating system. The model is based on the principles of disciplined oscillators and phase-lock loops (PLL) often found in the engineering literature. It has been implemented in the Unix kernel for several workstations, including those made by Sun Microsystems and Digital Equipment. The model changes the way the system clock is adjusted in time and frequency, as well as provides mechanisms to discipline its frequency to an external precision timing source. The model incorporates a generic system-call interface for use with the Network Time Protocol (NTP) or similar time synchronization protocol. The NTP Version 3 daemon xntpd operates with this model to provide synchronization limited in principle only by the accuracy and stability of the external timing source.

[MIL94c] Mills, D.L. Improved algorithms for synchronizing computer network clocks. *Proc. ACM SIGCOMM 94 Symposium* (London UK, September 1994), 317-327.

A later revision of this paper appears in *IEEE/ACM Trans. Networking*. See [MIL95b].

[MIL94d] Mills, D.L. Unix kernel modifications for precision time synchronization. Electrical Engineering Department Report 94-10-1, University of Delaware, October 1994, 24 pp. [Major revision and update of RFC-1589]

This report describes a model and programming interface for generic operating system software that manages the system clock and timer functions. The model provides improved accuracy and stability for most computers using the Network Time Protocol (NTP) or similar time synchronization protocol. This report describes the design principles and implementations of the model, while related technical reports discuss the design approach, engineering analysis and performance evaluation of the model as implemented in Unix kernels for modern workstations. The NTP Version 3 daemon *xntpd* operates with these implementations to provide improved accuracy and stability, together with diminished overhead in the operating system and network. In addition, the model supports the use of external timing sources, such as precision pulse-per-second (PPS) signals and the industry standard IRIG timing signals. The NTP daemon automatically detects the presence of the new features and utilizes them when available.

There are three prototype implementations of the model presented in this report, one each for the Sun Microsystems SPARCstation with the SunOS 4.1.x kernel, Digital Equipment DECstation 5000 with the Ultrix 4.x kernel, Digital Equipment 3000 AXP Alpha with the OSF/1 V1.x kernel and Hewlett Packard 9000 with the HP-UX 9.0x kernel. In addition, for the DECstation 5000/240 and 3000 AXP Alpha machines, a special feature provides improved precision to 1 μ s (stock Sun kernels already do provide this precision). Other than improving the system clock accuracy, stability and precision, these implementations do not change the operation of existing Unix system calls which manage the system clock, such as gettimeofday(), settimeofday() and adjtime(); however, if the new features are in use, the operations of gettimeofday() and adjtime() can be controlled instead by new system calls ntp_gettime() and ntp_adjtime() as described below.

A detailed description of the variables and algorithms that operate upon them is given in the hope that similar functionality can be incorporated in Unix kernels for other machines. The algorithms involve only minor changes to the system clock and interval timer routines and include interfaces for application programs to learn the system clock status and certain statistics of the time synchronization process. Detailed installation instructions are given in a specific README files included in the kernel distributions.

In this report, NTP Version 3 and the Unix implementation xntp3 are used as an example application of the new system calls for use by a synchronization daemon. In principle, these system calls can be used by other protocols and implementations as well. Even in cases where the local time is maintained by periodic exchanges of messages at relatively long intervals, such as using the NIST Automated Computer Time Service [LEV89], the ability to precisely adjust the system clock frequency simplifies the synchronization procedures and allows the telephone call frequency to be considerably reduced.

[MIL94e] Mills, D.L., and A. Thyagarajan. Network time protocol version 4 proposed changes. Electrical Engineering Department Report 94-10-2, University of Delaware, October 1994, 52 pp.

This report describes several changes and additions proposed for the Network Time Protocol Version 4. Previous versions of this protocol have been widely deployed in the Internet and a considerable body of experience has accumulated in their use. The proposed new version contains

several improvements to existing algorithms, together with new features designed to support multicast modes of operation, fully distributed peer subnets and automated NTP peer discovery and subnet configuration.

This report describes work in progress and is produced in order to stimulate discussion and refinement of the various algorithms. Some of the algorithms, in particular the work on refining the NTP local clock model, has already been implemented and now in test. Other work, including multicast support, is in the prototype phase, but needs further development and consensus on key management and distribution. Other work is still in the formative stage, including the fully distributed modes of operation and peer discovery and subnet configuration.

It is anticipated that further development in these areas will continue while discussion and consensus is reached on the final form of the protocol and its algorithms. When complete, the final design will be submitted for approval by the Internet standards apparatus.

### 5.2. Publications (with abstracts) in 1995

[MIL95a] Mills, D.L. Simple Network Time Protocol (SNTP). DARPA Network Working Group Report RFC-1769, University of Delaware, March 1995, 14 pp. [Major revision and update of RFC-1361]

This memorandum describes the Simple Network Time Protocol (SNTP), which is an adaptation of the Network Time Protocol (NTP) used to synchronize computer clocks in the Internet. SNTP can be used when the ultimate performance of the full NTP implementation described in RFC-1305 is not needed or justified. It can operate in both unicast modes (point to point) and broadcast modes (point to multipoint). It can also operate in IP multicast mode where this service is available. SNTP involves no change to the current or previous NTP specification versions or known implementations, but rather a clarification of certain design features of NTP which allow operation in a simple, stateless remote-procedure call (RPC) mode with accuracy and reliability expectations similar to the UDP/TIME protocol described in RFC-868.

This memorandum obsoletes RFC-1361 of the same title. Its purpose is to explain the protocol model for operation in broadcast mode, to provide additional clarification in some places and to correct a few typographical errors. A working knowledge of the NTP Version 3 specification RFC-1305 is not required for an implementation of SNTP. Distribution of this memorandum is unlimited.

[MIL95b] Mills, D.L. Improved algorithms for synchronizing computer network clocks. *IEEE/ACM Trans. Network*ing (June, 1995). [revision of ACM SIGCOMM 94 paper of the same name]

The Network Time Protocol (NTP) is widely deployed in the Internet to synchronize computer clocks to each other and to international standards via telephone modem, radio and satellite. The protocols and algorithms have evolved over more than a decade to produce the present NTP Version 3 specification and implementations. Most of the estimated deployment of 100,000 NTP servers and clients enjoy synchronization to within a few tens of milliseconds in the Internet of today.

This paper describes specific improvements developed for NTP Version 3 which have resulted in increased accuracy, stability and reliability in both local-area and wide-area networks. These include engineered refinements of several algorithms used to measure time differences between a local clock and a number of peer clocks in the network, as well as to select the best subset from among an

ensemble of peer clocks and combine their differences to produce a local clock accuracy better than any in the ensemble.

This paper also describes engineered refinements of the algorithms used to adjust the time and frequency of the local clock, which functions as a disciplined oscillator. The refinements provide automatic adjustment of algorithm parameters in response to prevailing network conditions, in order to minimize network traffic between clients and busy servers while maintaining the best accuracy. Finally, this paper describes certain enhancements to the Unix operating system kernel software in order to realize submillisecond accuracies with fast workstations and networks.

### 5.3. Work in Progress

[BON94] Boncelet, C.G., Jr, and D. Mills. Labeling Algorithms for Just-in-Time scheduling in TDMA networks. Submitted to: *IEEE/ACM Transactions on Networking*, 1994.

We describe scheduling algorithms for high speed, TDMA communication networks. The problem is to find a route and determine switch schedules so that a burst of information can traverse the network from source to destination. We show that this problem is difficult, i.e., that several simple variations are NP-complete. We present an algorithm that finds schedules, and within a simplifying assumption, is guaranteed to find a route if one exists. Simulations indicate that this algorithm runs quickly in practice and overall scheduling efficiencies of 80-90% can be achieved. We present optimizations for sending messages from one source to many destinations, e.g., multicasting. We also describe a variation in which the problem is to find a repeating schedule and we show how to adapt the basic algorithm for this variation. This new algorithm is no longer guaranteed to find a solution, but is adequate in practice.

[CAI95] Cain, B., A. Thyagarajan and S. Deering. IGMP Version 3. Proposal presented at the 32nd IETF, April 1995.

Abstract not available

[THY95a] Thyagarajan, A., S. Casner and S. Deering. Making the MBone real. *Proc. INET 95 Symposium* (June 1995) (to appear).

Abstract not available

[THY95b] Thyagarajan, A., and S. Deering. Hierarchical DVMRP for the MBone. *Proc. SIGCOMM 95 Symposium* (August 1995) (to appear).

Abstract not available

[THY95c] Thyagarajan, A., and D. Mills. Early commit: a distributed reservation algorithm for TDMA networks. (to be submitted to INFOCON 95).

Abstract not available