Advances in Computer Network Timekeeping

NSF Division of Network and Computer Reserach Grant NCR-93-01002
Final Report

Scalable, High Speed, Internet Time Synchronization

Defense Advanced Research Projects Agency Contract DABT 63-95-C-0046
Quarterly Progress Report for the period ending 28 May 1997

David L. Mills
Electrical and Computer Engineering Department
University of Delaware
14 July 1997

1. Introduction

This is the final project report for the NSF Division of Network and Communications Research and Infrastructure Grant NCR-93-01002, funded from June 1993. It is also a quartely project report for DARPA Information Technology Office Contract DABT 63-95-C-0046 The principal goal in the combined research program is the continued improvement in analysis, design, implementation, deployment and management of the distributed time synchronization function in large internets. Components of this work are supported by NSF and also by contracts and grants from DARPA ITO, U.S. Naval Surface Weapons Center and U.S. Army Research Laboratories. Additional support has been contributed in the form of special purpose timing receivers and cesium oscillators by U.S. Coast Guard and U.S. Naval Observatory, general purpose workstations by Hewlett Packard, Sun Microsystems and Digital Equipment, and other equipment grants from Arbiter, Cisco, Delmarva Power, Austron and Bancomm Divisions of Datum, and TrueTime.

As the components of this work interlock with each other in many and intricate ways, it is difficult and probably misleading to identify each work item as associated with a particular sponsor. Accordingly, the intended function of this report is a final report on the current NSF funding period, as well as an intermediate quarterly project report for DARPA.

Those contributing effort to the combined effort include:

Prof. David L. Mills, Principal Investigator
Graduate Students (at various times): Stephen Bijansky, Bradley Cain, Brian Huffman, Erik Perkins, Ajit Thyagarajan
Undergraduate Students (at various times): Stephen Bijansky, Marie Conte, Brian Huffman, Douglas Miller

Since the beginning of the current funding period in June 1993, there has been considerable progress in a number of areas, including the continued development and implementation of the Network Time Protocol (NTP) Version 4 and its subset the Simple NTP (SNTP), analysis and modelling of computer clocks, experimental studies of timekeeping functions in the Internet, development of teaching laboratory projects and development of new courses at both the graduate and undergraduate level.

1.1 Network Time Protocol Version 4

Since the first version of NTP came into service almost twenty years ago, there have been a continuing series of improvements in accuracy, reliability, security and interoperability. As suites of improvements have matured, new versions of the protocol specification and reference implementation have been released, culminating in the current NTP Version 3, which has been in widespread use for the last several years. There have been numerous incremental upgrades since Version 3 came into use, all of which are backwards compatible. Eventually, these along with others proposed in [10], will become the basis of NTP Version 4. During the current funding period, the Version 4 design has been upgraded in four important areas: autonomous configuration, security model and authentication scheme, clock discipline algorithms and multicasting technology. These are reviewed in following subsections.

1.2 Autonomous Configuration

By far the most often asked question by a newcomer to NTP is how to configure the clients and servers of the time synchronization subnet. Engineering accurate and reliable subnets is an intricate business and includes considerations which are not immediately obvious, such as finding a good set of redundant servers, avoiding common points of failure and dealing with sometimes cranky Unix kernels and computer clocks with large inherent frequency errors. The issues become even more complex when broadcasting and multicasting are used to synchronize large populations of clients over possible global Internet spans. Some idea of the complexity is evident by the facts there are well over 100,000 servers and clients in the public Internet, plus many thousands more behind corporate and government firewalls, all depending on about 230 primary time servers; that is, those synchronized directly to national time standards by radio, satellite or modem.

As in many other areas in the NTP design, the best solution to these problems is for the architecture and protocol model to deal with them automatically and autonomously. In the most attractive scenario, the shrinkwrapped software would be installed and configured automatically on each different machine architecture and operating system, then search its environment for suitable servers and maintain operation automatically, including searching for additional servers as failures become known.

In fact, as the result of volunteer contributions to the programming effort, automatic configuration for various architectures and operating systems is a near reality, at least to the extent of the capabilities of the GNU autoconfigure system, as described later in this report. While it is possible to simplify the procedure by artful DNS naming conventions, completely, manual selection of remote servers and operating modes, continues to be the standard practice.

A model capable of completely autonomous network configuration has been designed as part of Ajit Thyagarajan’s dissertation. The model uses an expanding-ring multicast search to discover nearby servers, then configures multiple-stratum client-server tree using an add/drop greedy heuristic designed to minimize the NTP synchronization distance (roughly equivalent to expected accuracy) under degree and distance constraints. The design spreads the load equally among the set of available servers, while at the same time insuring a high level of accuracy, redundancy and diversity. Mr. Thyagarajan’s dissertation is almost complete, needing only completion of the proof-of-concept implementation, which is built upon the existing NTP Version 3 software base.

1.3 Security Model and Authentication Scheme

During the design of the Kerberos security system, the designers at MIT realized the importance of reliable timekeeping in the face of possible hostile attack. They used NTP to synchronize the computer clocks and enforce lifetimes on the various cryptographic media used in the system. Even without cryptographic means, the NTP architecture and protocol are inherently resistant to most kinds of attacks; however, there are some vulnerabilities that can be exploited by a determined intruder.

Accordingly, and specifically for Kerberos, the NTP architecture and protocol include means for cryptographic authentication of servers and clients. This is done with cryptographic checksums using either the Data Encryption Standard (DES) operated in Cipher-Block Chaining (CBC) mode, or the MD5 hash function defined by RSA. Both of these use secret keys upon which the security of the method depends. However, these methods become exceedingly awkward when large populations of servers and clients are involved, since a secret key must be used for each client-server association and there are well over a quarter million such associations in the Internet now. A much better approach would have each client authenticate each server and synthesize keys automatically without relying on secret key files, as is now the standard practice.

The application clearly calls for public-key cryptography; but, as described in [3] and [5], all known methods are too slow, even on fast, modern workstations, when clock accuracies better than a millisecond are required. Accordingly, a search was made of existing methods described in the literature or proposed by IETF task forces or proposed by others known personally to this investigator (Steve Kent foremost among them). The result is three different schemes, each applicable to one of the several association modes used by NTP. All three schemes use the present DES-CBC or MD5 cryptographic checksums for backwards compatibility, but differ in the manner in which the keys are developed.

One scheme is designed for symmetric modes, in which two or more redundant servers operate peer paths with each other and with primary servers. Ordinarily, each server operates at stratum 2 (primary servers operate at stratum 1). If for some reason one or more servers lose connectivity to their configured primary servers, then these servers automatically synchronize to the remaining stratum-2 servers and continue operation at stratum 3. When connectivity is restored, the subnet reconfigures automatically as originally intended.

Another scheme is designed for busy servers with hundreds, possibly thousands, of clients. In such cases, the server operates in a stateless mode and retains no persistent state between client requests, so it is not possible to maintain individual keys for each association, nor to use any of the proposed IETF authentication schemes such as Photuris. Therefore, in this scheme the private key is recomputed for each request message upon arrival, but using a fast hash algorithm based on a server private value, which is never disclosed, plus public values in the request itself.

The third scheme is designed for multicast servers and clients, in which the server sends periodic messages, but ordinarily receives no requests at all. In this scheme, the server periodically generates a list of keys that are used only once. Each key is a hash of the previous key and includes a private server value which is never disclosed. The server then uses the list in reverse order, as in the S-KEY scheme. Clients verify that the hash of the current key equals the key last used in order to verify the identity of the sender. The scheme provides for the occasional loss of a message and for the infrequent use of public-key cryptography to verify the key list actually was generated by the intended server.

The revised security model and authentication scheme are described in [5]. This report includes an exhaustive security analysis of the NTP architecture and protocol, including a vulnerability assessment of the new scheme and the hardware and software environment in which NTP runs.

1.4 Clock Discipline Algorithms

A project that has been ongoing since first tick of the NTP clock is improving the accuracy and stability of computer timekeeping to the maximum extent possible with current computer hardware and software. For instance, the clock reading precision has been improved from a few milliseconds on a Sun SPARC running SunOS 4.1 a decade ago to less than a microsecond on a Sun UltraSPARC running Solaris 2.5 today. In practice, each significant improvement in reading precision invites a re-examination of the NTP algorithms, since residual inaccuracies, formerly ignored in the interest of simplicity, become significant. The most recent re-examination [9] describes several incremental improvements which result in reliable time synchronization to the order less than a millisecond on LANs and lightly loaded T1 networks such as DARTnet.

During the current funding period, a number of advances were made in the analysis, modelling and implementation of various timekeeping systems. One of the most important of these is a set of modifications to the Unix kernel which provides very fine control of the computer clock time and frequency. The modifications described in [11][12][13] include an interface for time and frequency adjustments provided by a time synchronization protocol such as NTP, as well as precision frequency discipline provided by a pulse-per-second (PPS) signal produced by devices such as radio and satellite timing receivers and cesium clocks.

An active agenda has been pursued to persuade the various workstation makers to incorporate these modifications in their current products. To date, Digital has done this with Digital Unix and Sun Microsystems has promised this with the next version of Solaris. In addition, public versions of Linux and FreeBSD have these modifications in stock kernels. So far, while the modifications have been ported to HP-UX, Hewlett-Packard has no known plans to do this. Meanwhile, another active agenda has been pursued to provide an industry standard API for the PPS signal using one of the modem control leads of a serial port. So far, Digital, Sun and the FreeBSD implementers have agreed to use the standard API, but the actual implementation in stock kernels is so far incomplete.

A most important factor in achieving the highest degree of accuracy is exhaustive analysis and modelling of the computer clock oscillator. A large body of work exists in the analysis and modelling of precision quartz oscillators used in frequency and time standards, but only one paper (by Judah Levine of NIST) other than the papers and reports of this investigator, has considered the common computer oscillator, which in no way can be considered a precision device. The papers and reports published by this investigator [4] reveal subtle differences in the stability characteristics of precision quartz oscillators and the uncompensated quartz oscillators used in typical workstations and servers.

The recent analysis reported in [4] and [8] is based on a statistic used widely in the precision time and time interval community called Allan variance, but is in general little known outside that community. The project considered the same network paths as described in a network experiment described later in this report. They included primary time servers in the U.S., Europe, Asia and South America, and network delays that can be described only in terms suitable for a pinball machine.

From these data and the measurement of Allan variance, certain subtle but highly effective modifications in the current clock discipline algorithm were suggested. This led to the development of a NTP simulator which faithfully mimics the operation of each of the significant NTP algorithms and can be driven with data collected from real NTP associations, as described in later sections of this report, or driven by synthetic data. The modifications were implemented in the simulator and tested with real and synthetic data, as described in [2]. Our near term plans include implementing these improvements in the current NTP software. The simulator itself has been made available to interested collaborators for further investigation.

1.5 Multicasting Technology

Intrinsic to the NTP Version 4 design is the use of multicasting, not only for the time synchronization function, but also for clients to discover servers capable of providing accurate and reliable time synchronization. NTP has been in the broadcast/multicast business for a long time; in fact, the first Internet multicast group address, 224.0.1.1, was assigned to NTP. Many subtle features and subtle problems have been discovered in the multicast paradigm, many of them first stumbled upon with NTP multicast servers and clients. At present, there are over a dozen NTP multicast servers operating in Europe and the U.S. providing synchronization to an unknown number of clients.

Graduate students Ajit Thyagarjan and Bradley Cain have been instrumental in advancing the multicasting agenda, both within the IETF and is various systems developed by them. Mr. Thyagarajan spent two summers as an intern at Xerox PARC learning the trade from Steve Deering and associates. He is the principal architect of the hierarachical model proposed for the MBONE [14][15]. He and Mr. Cain are among the principal architects of the IGMP protocol, in particular IGMP Version 3 [1], which bind multicast host receivers to local routers. This work has been reported at IETF meetings and, in the case of Mr. Cain, a Masters Thesis.

During the current funding period, the NTP software for Unix and Windows has been enhanced with several features which exploit multicast technology, as described in [2]. Perhaps the most intricate has to do with simultaneous multicast and unicast operations where multicast service is used as backups for the more accurate unicast service. This involves the use of a crafted mitigation scheme which ranks local radio clocks, modem services, unicast servers and multicast servers by a set of criteria designed to select the most accurate servers available, while providing orderly fallback should one or more of these servers fail.

Another feature designed, implemented and documented [10], is a scheme where a multicast client polls a multicast server in client-server mode in order to calibrate the multicast routing tree propagation delay and, after accurate calibration has been determined, revert to ordinary listen-only multicast mode. As the multicast tree propagation delay can be quite different than the unicast routing paths, especially with CBT and PIM sparse-mode routing algorithms, this scheme can provide accuracy consistent with client/server modes.

Multicasting technology is also the topic of ongoing collaborations with others in the community, including University College London and Science Applications International, as well as the topic of a recent proposal submitted on behalf of this investigator and collaborators in our department.

2. Infrastructure

A good deal of time on the part of this investigator and students has been devoted to development and maintenance of personal and laboratory infrastructure, such as web pages, operating system and network upgrades, teleconferencing subsystems and various collaborative software distributions. There is now an extensive set of web pages for the NTP site www.eecis.udel.edu/~ntp, which is currently maintained by volunteers from Hewlett-Packard, plus personal home pages for this investigator www.eecis.udel.edu/~mills, each of his students, and the Internetworking Research laboratory www.eecis.udel.edu/~millslab.

This work is taken seriously - all publications and software distributions of any form are available from the web pages, as well as specially crafted briefings for each of the current active projects and many of those completed in the last several years. Published papers, reports and memoranda are available in PostScript format with tables, formulas and graphics intact as in the published version. The briefings are in three formats: PostScript, Microsoft PowerPoint and HTML. Most of the figures and graphs shown in the papers and reports are included in the briefings along with explanatory text.

One happy consequence of the toil required to construct the web pages is a reduction in the number of messages received about NTP and the effort to answer newbie questions. The sender is simply directed to the web pages and we both get on with normal life. This is not a trivial consequence, since it reduced the volume from some 50 messages per day to only a couple.

Another infrastructure project is maintenance of the NTP software itself. Previously, much of the effort required was solving some problem in porting the distribution to another machine architecture or operating system. Since operating system upgrades for most workstations occur once or twice a year, this effort was not trivial. Fortunately, veteran Unix programmer Harlan Stenn volunteered to upgrade the distribution to use the GNU autoconfigure system, which vastly simplifies the porting and upgrade problems. Even with Harlan’s help, maintaining the distribution, which has the clawmarks of over two dozen implementers, remains a significant part of our effort.

Still another infrastructure project is the maintenance of our DARTnet connection, which provides access to the current DARTnet experiment community as well as indirect access to the new CAIRN experiment community. The access is funded by DARPA to support ongoing research projects at no cost to NSF.

3. SNMP project

Recently, a project was started to develop a MIB for NTP. This is an issue which has been dormant for some years. Comprehensive provisions for remote monitoring and control of NTP servers and clients were included in the original NTP Version 2 software, which appeared well before SNMP became available in any form. There was and is considerable advocacy within IETF to provide SNMP support in NTP. However, this involves considerable effort for several reasons, including the fact that the single software distribution is self-configured for over two dozen machine architectures and operating systems, involves hundreds of variables in the case of busy servers, and no clear way is available to extract NTP daemon variables using the SNMP support provided with current workstations.

Prof. Adarsh Sethi of the Computer and Information Sciences Department volunteered his time and that of his graduate student to specify a MIB and implement prototype software to provide SNMP support usable by available management programs. At this time, a preliminary MIB specification has been completed and extensions to available SNMP agent software is under way. Because of the novel way the system works, further description may be of interest.

The system is designed to appear as a SNMP proxy agent to a manager program running in another computer. However, in order to avoid a considerable rewrite of the existing NTP monitoring code, the proxy agent communicates with the NTP daemon using the native NTP monitoring protocol. The proxy agent thus translates SNMP requests into NTP requests and sends them to the NTP daemon, which in principal may run in some other computer. The proxy agent then translates NTP monitoring data received from the NTP daemon to SNMP format and forwards it to the manager program.

In the final version to be developed, the proxy agent software must be embedded in the native SNMP support for each operating system. The preliminary version uses available SNMP agent software for this purpose. Since it is not practical to modify software distributed with current workstations, and the IETF has not yet come to agreement on how to interface variables resident outside the kernel, full deployment of the NTP MIB within the standard SNMP must await resolution of these issues.

4. SNTP project

The NTP software implementation for Unix and Windows, while highly evolved to enhance accuracy and reliability, is a highly complex and relatively large body of software. The algorithms are designed to cope with very large network jitter, untrustworthy servers and many different operating systems. However, there are cases where the size and complexity of the full NTP implementation could probably be described as overkill. There are two applications which could benefit from a stripped-down version of the software without significantly degrading performance.

One of these applications is where the NTP protocol support is embedded in a dedicated time server which is itself directly synchronized to an integrated radio or satellite receiver. Devices including a GPS receiver, Ethernet interface and NTP support are now manufactured by Bancomm, TrueTime, Spectracom and others. Since the server is never synchronized to any source but the onboard receiver, there is no need for the complex suite of algorithms used with the full implementation.

The other application involves private workstations where the client has access to server(s) “close” on the same network and where the client never acts as a server for other clients. Typical examples of such systems include the ubiquitous PC running Windows 95 and with no time-critical applications onboard.

In both of these applications, the complete suite of algorithms developed for the full NTP implementation are not justified. However, when designing a product to operated seamlessly in the current Internet with some mix of full and subset implementations, it is important to provide a clearly stated standard. After careful consideration of the issues involved, a formal subset of the standard specification RFC-1305 was developed and now called the Simple Network Time Protocol or SNTP. The specification has been amended twice, both to improve clarity, correct minor errors and provide support for IP Version 6 and OSI. The final draft of the specification has been published as an RFC [6].

There are now several shareware implementations of SNTP for various PCs, including Windows 95 and NT clients, Macintosh and others. While SNTP could in principle be used for Windows NT servers, the full NTP implementation is more appropriate for these servers and is now available.

5. Internet Experiments

The Internet experiment program, which has been ongoing in one form or another since 1979, continues to develop as interesting new phenomena are detected. There were four interesting experiments conducted during the funding period, one to determine the extent of deployment and expected accuracy of NTP servers and clients in the Internet of today, another to calibrate the delays between primary servers located in far away places, a third to explore the phenomenon of long-range dependency in the Internet and a fourth to explore the use of NTP as a global Internet monitoring tool.

5.1 Global Timekeeping

In September 1996 at the invitation of timekeepers in various countries of the world, a network time synchronization experiment was initiated to monitor the time differences between primary time servers in the U.S. and those in other countries in Europe, Asia and South America. The primary goal in the experiment is to assess how reliable and with what accuracy time synchronization could be maintained via intercontinental Internet paths, which are presently overloaded and approaching serious congestion levels. A secondary goal is to collect network performance data over relatively long periods, so that possibly interesting long-range dependency phenomena could be revealed.

The experiment involves primary time server *pogo* in our laboratory and almost two dozen other primary time servers, including some at institutions charged with maintaining national time standards and others used as controls. All servers involved in the experiment are synchronized by radio, satellite or modem to national time standards with accuracies in general less than a millisecond. In the experiment, outbound and return one-way delays are measured between *pogo* and each of the other time servers at intervals of about one minute. The measurements are made using the standard monitoring capabilities of the current NTP software distribution for Unix and Windows and saved in a database. After nine months of continuous operation, the database amounts to well over a gigabyte.

The results of a preliminary analysis of the database are reported in [4], from which a few surprises emerge. One of them is that the differences between the outbound and reciprocal propagation delays on intercontinental circuits are surprisingly small, while the differences on domestic paths can be quite large. Since these differences directly affect the time accuracy with secondary time servers (those without an independent source of synchronization), it is important to identify the cause. At the present level of development, the international infrastructure is thinly deployed, with in most cases only a single international path available. However, as is now well known, those occasions where the outbound and reciprocal path are quite different are legion in the domestic Internet.

In fact, by using extensions to the algorithms developed for NTP, it is possible to measure the propagation delay differences with a high degree of accuracy and separate it from the queueing delays in the routers along the path. From this analysis, it is clear that the propagation delay and queueing delay contribute about equal proportions to the total delay on the intercontinental paths. In the final analysis, timekeeping over international paths is rather better than expected, considering the sometimes severe congestion and large delay variations involved.

On some paths, specifically to the Italian national primary time server, exceptionally large delay variations occur over the hours of the day and days of the week. This is certainly not unexpected, but the degree of variation is surprising and suggests that Italians leave their computers at work and don't surf the web in the evening to the degree that Americans do.

5.2 Long Range Dependency

An interesting facet of computer traffic modelling has emerged in recent research projects called long-range dependency or LRD. Briefly, the tail of the traffic distribution does not decay exponentially as expected with current models in the literature. Instead, the behavior appears fractal in nature, where the statistical characteristics observed over one time scale is similar to that observed over another quite different time scale. The literature reports this phenomenon over time scales from milliseconds to thirty minutes or so on a LAN. We were interested if this behavior extended to longer intervals in the global Internet.

Graduate student Qiong Li has accumulated convincing evidence to suggest that LRD behavior extends to a period of several months at least. This conclusion is based on an analysis of the database collected during the global timekeeping experiment described previously. The phenomenon is quite surprising, since mechanisms that might have effects on the delay models over time scales of months are not immediately apparent.

Upon further analysis, we believe the explanation lies not in the network itself, but in the traffic volume generated by the computers that use the network. To test this conclusion, we have analyzed the interarrival statistics for new connections to our busy web server for the last year and confirmed these statistics also show LRD, once the affects of hourly and daily variations have been carefully removed. Finally, we believe at least some of the LRD is due to additional provisioning of existing Internet paths and turning up new networks around the globe. Carried to a natural conclusion, LRD analysis may prove a valuable tool for projecting future traffic and provisioning needs.

Mr. Li has written a draft paper about these observations and preliminary conclusions. After internal review and markup, it will be submitted for publication.

5.3 Survey of Internet Time Servers

On occasion during the evolution of NTP over the years, informal surveys have been carried out in order to determine the extent of deployment and in general how well the NTP synchronization subnet is working. Starting with several hundred servers in the latter half of the 1980s, the population has grown to an estimated total of well over 100,000 servers on the public Internet, plus thousands more on private and corporate internets. These populations were estimated by extrapolating known populations of NTP servers and clients relative to the total populations of corporate and university networks, so can be considered only approximate.

In an effort to update the database, as well as survey how well the global synchronization subnet is working, a massive survey has been conducted using monitoring tools included in the NTP software distribution for Unix and Windows, as well as the remote monitoring features of the NTP daemon itself. These facilities are designed to reveal quite intimate behavior of the daemon and its algorithms, including its own system state variables and each of its peer state variables.

In the survey, a number of schemes were used to discover the primary time servers; that is, those synchronized to independent sources of time, such as a GPS receiver. Over 230 of these were found, about half of which are registered in the list of public primary time servers maintained for public access. The state variables for these primary servers were then retrieved and added to the database. Using a special monitoring feature, for each of the primary servers the Internet addresses for each of its recent clients were then retrieved and these clients surveyed in the same way. Continuing in this way, the state variables for 34,000 servers and clients were added to the database. For these, over 182,000 server-client association pairs were found and the time and frequency offsets determined between them.

A preliminary analysis of these data reveals a number of findings useful for the assessment of service quality, such as the time and frequency error distributions, as well as a number of cases where the protocol was misconfigured or operating improperly. A summary of the results has been posted on the web for public access and will be the subject of a paper for early publication.

5.4 The ‘Bot

Experience with NTP over the years, including the day to day maintenance of a number of servers and clients and the results of surveys such as the above, have demonstrated the value of NTP as a diagnostic tool. So far as can be determined, NTP is the longest running, continuously running, diagnostic protocol ever operated on a computer network. Since the first version began operation in the late 1970s, instances of the protocol have been operating and collecting intricate timekeeping data from which not only the clocks have been disciplined, but subtle problems have been detected, such as intermittent memory problems, input/output hardware failures, network routing flaps, unexpected network route selection and many other phenomena, simply by carefully watching the clock offset and path delay variations. In fact, by calibrating the computer clock frequency with temperature, NTP has been used as a sensitive room temperature thermometer and air conditioning system monitor.

Graduate student Bradley Cain conceived the idea of using the NTP monitoring programs and survey scripts as a toolkit for remote network diagnostics. Using a crafted set of shell scripts, he built a proof-of-concept system called the ‘Bot (short for robot), that is in effect an expert system which generates continuous, low rate queries to selected remote time servers and clients. The data retrieved represent paths from each client to each of its own servers. By carefully selecting the remote clients and servers, it is possible to find paths spanning interesting places in the Internet, such as provider interchange points, overseas links and so forth. The paths are polled looking for anomalistic behavior, such as delay surges and dropouts. If the behavior exceeds some tripwire, depending on the nature of the monitored data, additional tools are launched, including the ubiquitous traceroute, in order to sound the path from the monitoring host to each end of the path and discover things like routing flaps, link failures and congested links.

The most attractive feature about this experiment is that it is ubiquitous and does not rely on sometimes jealously guarded performance data or access to SNMP monitoring facilities. The pings are low level and in general lost in the network noise and in most cases can be piggybacked on real NTP synchronization paths. The experiment produces regular reports, runs continuously and autonomously, and requires no human intervention other than to fix broken things that are detected by the ‘Bot.

Results from initial experience with the ‘Bot are encouraging. In preliminary experiments, several instances of unexpected and intermittent link congestion were found, including some in other countries. In addition, there is great promise for the development of new network troubleshooting tools and expert systems as crafted by new generations of graduate students. This is expected to be an ongoing project, with results reported on the web and in papers and reports as experience develops.

6. Collaborations

There is a long history of collaborations between this investigator and others involved with ongoing DARPA and NSF programs. Especially useful have been joint projects with government and industry, including the DARPA DARTnet and CAIRN projects, and support for ongoing projects in time synchronization at NIST, USNO, and their counterparts in other countries. This investigator and Judah Levine of NIST have supported each other in the design and implementation of computer clock discipline algorithms and in the deployment of NIST primary time servers in several places in the US. This investigator and Richard Schmidt of USNO have collaborated in the deployment of USNO primary time servers and in the provision of cesium oscillators in our laboratory.

A joint proposal with professors Khokhar and Gao of our department has been submitted to NSF. This project is designed to advance the state of the art in internet multicasting algorithms, as well as support the DARPA CAIRN proposal, which itself is designed as a collaboration between University College London, Science Applications International and our laboratory. The infrastructure to support this effort is funded by DARPA.

7. Teaching and Course Development

Four new courses were developed and taught during the funding period. Two of these cover the areas of data communications and computer networks, one taught at the senior undergraduate level and the other at the graduate level. A descirption and syllabus for all courses is on the Web. The undergraduate course is an elective in the senior year of our new Computer Engineering Program curriculum. The graduate course is a core course for the PhD qualifier examinations. These courses have been evolved over thirty years of teaching topics in data communications and networks at undergraduate and graduate levels and in continuing education seminars. In passing, it is interesting to note how dramatically the core topics have evolved and changed in response to government deregulation, proliferation of computers and the coming of the Internet and the Web.

The other two courses cover the area of cryptographic algorithms and their application to computer network security and are intended as a two-semester sequence at the graduate level. The first course lays the mathematical foundations, including the theory of finite groups and the algorithms used for computation with these groups, including those for solving modular equations and generating large primes. The course continues with a study of the current public-key and private-key cryptosystems and related algorithms.

The second of the two-course sequence covers principles of secure computer and computer network design, including threats and countermeasures, cryptosystems now deployed on the Internet and issues of government concern and regulation. Both courses are taught now as seminars at the advanced graduate level for students in The Electrical Engineering and Computer and Information Sciences departments.

7.1 Laboratory Demonstrations

Over the years, a number of experimental devices have been designed and built to evaluate different methods of synchronizing computers to national means of time and frequency dissemination. These include an audio device driver that can decode standard Inter-Range Instrumentation Group (IRIG) signals to synchronize a workstation using the native audio codec, as well as an inexpensive LORAN-C receiver and software for a PC, and a DSP program that implements a digital modem for shortwave radio data transmission [7]. While useful for their intended purpose to synchronize computer clocks, the primary motivation for these projects was as demonstration and laboratory projects for graduate and undergraduate instruction. The projects are carefully documented and designed to demonstrate modern communications theory reduced to practice in the form of modular hardware and software components.

The newest addition to this kit is a DSP program that uses an inexpensive shortwave receiver and DSP chip to demodulate and decode signals transmitted by NIST stations WWV and WWVH. The gadget delivers an ASCII timecode string that can be used by a PC or workstation to set the computer clock with accuracy in the order of a few milliseconds. While presently implemented as a special purpose DSP program in assembler code, the next step anticipated is an implementation in portable C that can use a PC sound card or workstation native audio codec as the radio interface. The design, including source code and documentation, has been made available to a group of experimenters for evaluation and possible further development. This project has only recently been completed and remains to be reported in the literature.

7.2 Graduate and Undergraduate Students

Graduate student Ken Monington has completed his dissertation on network clock synchronization and taken a postdoctoral appointment at NIST with Judah Levine. Ajit Thyagarajan is in the final stages of his dissertation on autonomous configuration of distributed services. Bradley Cain has completed his masters thesis on multicasting protocols and joined Bay Networks. First-year graduate student Qiong Li has begun study in the area of long-range dependency leading to a paper to be submitted for publication.

Undergraduate students Stephen Bijansky and Marie Conte completed their undergraduate theses and were awarded a Degree with Distinction, which is the Engineering School equivalent to the traditional honors degree.

8. Miscellany

Attendance at a number of symposia and conferences has been supported during the funding period, including trips on behalf of SIGCOM program committee meetings, SIGCOM paper presentations, and others. Graduate students Ajit Thyagarajan and Bradley Cain attended selected IETF meetings to present position papers and draft reports.

The German federal government sponsors a series of computer science retreats at Dagstuhl in southern Germany. A weeklong retreat was held In March 1996 on the special topic of computer time synchronization. This investigator was invited to present an in-depth seminar on the architecture and models of NTP. The meeting presented the opportunity to exchange views with other investigators, especially those involved in theoretical areas.

This investigator spent four months of sabbatical leave at University College London in the fall of 1995. Besides exchanging very interesting and useful views and participating in class instruction, the occasion provided an ideal opportunity for an in-depth review of the mathematical theory of finite groups and the application to cryptographic algorithms.

In recognition of the joint interest of electrical engineers and computer scientists in the areas studied by this investigator and the courses taught, he has been appointed to the faculty of the Computer and Information Sciences Department. In addition, his existing appointment in the Electrical Engineering Department has been changed with the department official name, which is now the Electrical and Computer Engineering Department. The full official title may be too long for a business card.

9. Publications

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

2. 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.

3. 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.

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

5. 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.

6. 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. Major revision and update of RFC-1769 and RFC-1769.

7. Mills, D.L. An optimal linear receiver and codec for a class of radiotelegraph signals. Electrical Engineering Department Report 95-8-1, University of Delaware, August 1995, 91 pp.

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

9. Mills, D.L. Precision synchronization of computer network clocks. *ACM Computer Communication Review 24, 2* (April 1994), 28-43.

10. 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, 32 pp.

11. Mills, D.L. Unix kernel modifications for precision time synchronization. Electrical Engineering Department Report 94-10-1, University of Delaware, October 1994, 24 pp.

12. Mills, D.L. Unix kernel modifications for precision time synchronization. Network Working Group Report RFC-1589, University of Delaware, March 1994. 31 pp.

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

14. Thyagarajan, A., and S. Deering. Hierarchical DVMRP for the MBone. *Proc. SIGCOMM 95 Symposium* (August 1995).

15. Thyagarajan, A., S. Casner and S. Deering. Making the MBone real. *Proc. INET 95 Symposium* (June 1995).