Survivable, Real Time Network Services

Defense Advanced Research Projects Agency
Contract F30602-98-1-0225, DARPA Order G409/J175

Quarterly Progress Report
1 April 2002 - 30 June 2002

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, graduate student Harish Nair and undergraduate research student John Conner. MS student Tamal Basu has graduated and taken a position at Acorn Networks. Phd student Qiong Li has graduated and taken a position with Phillips Research.

The project continues research in network time synchronization technology funded by DARPA, US Army ARL and NASA/JPL. 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 and sensor networks.

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.

In brief summary, we have passed the following milestones during the quarter. These include

1. We have completed extensions to the Network Time Protocol for IPv6 support. While we have tested IPv6 functionality with all protocol modes and security features described in this report, there are still some fine tuning for the utility programs and compatibility issues.

2. We have completed an implementation of the Autokey Version 2 security protocol for NTP and submitted an Internet Draft with the revised protocol specification. The implementatino includes cryptographic challenge/response identity schemes and certificate trails.

3. We have improved the behavior of the Manycast optimized configuration scheme for NTP, including a hysteresis scheme to avoid clockhopping and means to manage stratum management.

4. We have implemented a proof-of-concept discrete event simulator for NTP in the Interplanetary Internet and tested it using planetary ephemerides.

Occasionally for record, these paper reports will include a more extensive discussion drawing on the work reported on the web. This report includes information about all projects contributing to the combined effort, with the funding source for each one specifically stated.

2. Autonomous Authentication

The missions considered in this project involve autonomous sensors that might be deployed from a reconnaissance vehicle over a battlefield or from a space probe over a planetary surface. Once deployed, the sensor network must operate autonomously using an ad-hoc wireless infrastructure as sensors are deployed or destroyed or the network is damaged or compromised and then repaired. In the traditional fog of war scenario, sensors may be able to communicate directly only with nearby neighbors and in particular may be able to assess trust only intermittently and not always directly from a trusted source.

The goal of this project is to develop and test security protocols which resist accidental or malicious attacks on the servers and clients in a sensor network. Clients must determine that received messages are authentic; that is, were actually sent by the intended server and not manufactured or modified by an intruder. In addition, they must verify the authenticity of any server using only public information and without requiring external management intervention.

A sensor network is protected by a set of cryptographic values, some of which are instantiated before deployment and some of which are generated when needed after deployment. Probably the most important value is the group key which must be instantiated in each sensor before deployment. A sensor proves to another sensor that it is a legitimate group member if it can prove it knows this value. In addition to the group key, every sensor has a host key used to sign messages and certificates and one or more certificates signed by the host key. While the group key must persist for the lifetime of the group, or at least for the lifetime of the mission, the host key and certificates can be regenerated from time to time.

In our model a subset of sensors is endowed by some means as trusted primary sensors, either directly by command or indirectly by election in case the network becomes fragmented. The remaining secondary sensors must authenticate from the primary sensors, directly or indirectly, using only cryptographic values already instantiated. In other words, sensors can rely on no help other than already available from other sensors via the security protocol.

2.1 Brief Description of Work and Results

Our approach involves a cryptographically sound and efficient methodology for use in autonomous sensor networks, as well as other ubiquitous, distributed services deployed in the Internet. As demonstrated in the reports and briefings cited on the web page, there is a place for Public-Key Infrastructure (PKI) schemes, but none of these schemes alone satisfies the requirements of a real-time sensor network security model. The Photuris and ISAKMP schemes proposed by the IETF require per-association state variables, which contradicts the principles of the remote procedure call (RPC) paradigm in which servers keep no state for a possibly large population of clients. An evaluation of the PKI model and algorithms as implemented in the OpenSSL cryptographic library leads to the conclusion that any scheme requiring every real-time message to carry a PKI digital signature could be vulnerable to a clogging attack.

We have used the Network Time Protocol (NTP) software and the widely distributed NTP synchronization subnet in the Internet as a testbed for distributed protocol development and testing. Not only does the deployment, configuration and management of the NTP subnet have features in common with sensor applications, but a synchronization service itself must be an intrinsic feature of a sensor network infrastructure.

While NTP Version 3 contains provisions to authenticate individual servers using symmetric key cryptography, it contains no means for secure keys distribution. Public key cryptography provides for public key certificates that bind the server identification credentials to the associated keys. Using PKI key agreements and digital signatures with large client populations can cause significant performance degradations, especially where precise timing and low power is required. In addition, there are problems unique to NTP in the interaction between the authentication and synchronization functions, since reliable key management requires reliable lifetime control and good timekeeping, while secure timekeeping requires reliable key management.

A revised security model and authentication scheme called Autokey was proposed in earlier reports and papers cited on the web page. It has been evolved and refined over time and implemented in NTP Version 4. It is based on a combination of PKI, IPSEC and a pseudo-random sequence generated by repeated hashes of a cryptographic value involving both public and private components. This scheme has been deployed and evaluated in a local environment and in the CAIRN research testbed. An executive summary of the design and implementation is included here. The Authentication Options program documentation page provides additional details. A technical report and Internet Draft cited below defines the protocol and data structures in detail.

The Autokey scheme is now in its third generation after the original described in the technical report and Version 1 described in previous Internet Drafts. While designated Version 2, a definitive protocol specification document has not yet been completed. The major changes include upgrading the cryptographic library to OpenSSL, which provides a selection of message digest and signature encryption routines, as well as support for X.509 certificates. The protocol has been simplified and made more rugged and stable in the event of network or server disruptions. A major deficiency in previous versions where the cookie was transmitted in clear has been fixed by encrypting it using public key cryptography. In addition, the latest version has a selection of identity schemes based on cryptographic challenge/response algorithms.

2.2 Autokey

The Autokey scheme is based on the PKI algorithms of the OpenSSL library, which includes an assortment of message digest, digital signature and encryption schemes. As in NTPv3, NTPv4 supports symmetric key cryptography using keyed MD5 message digests to detect message modification and sequence numbers (actually timestamps) to avoid replay. In addition, NTPv4 supports timestamped digital signatures and X.509 certificates to verify the source as per common industry practices. What makes the Autokey scheme special is the way in which these algorithms are used to deflect intruder attacks while maintaining the integrity and accuracy of the time synchronization function. The detailed design is complicated by the need to provisionally authenticate under conditions when reliable time values have not yet been acquired. Only when the server identities have been confirmed, signatures verified and accurate time values obtained does the Autokey protocol declare success.

The NTP message format has been augmented to include one or more extension fields between the original NTP header and the message authenticator code (MAC). The Autokey protocol exchanges cryptographic values in a manner designed to resist clogging and replay attacks. It uses timestamped digital signatures to sign a session key and then a pseudo-random sequence to bind each session key to the preceding one and eventually to the signature. In this way the expensive signature computations are greatly reduced and removed from the critical code path for constructing accurate time values.

Each session key is hashed from the IPv4 or IPv6 source and destination addresses and key identifier, which are public values, and a cookie which can be a public value or hashed from a private value depending on the mode. The pseudo-random sequence is generated by repeated hashes of these values and saved in a key list. The server uses the key list in reverse order, so as a practical matter the next session key cannot be predicted from the previous one, but the client can verify it using the same hash as the server.

There are three Autokey protocol variants in NTP, one for client/server mode, another for broadcast/multicast mode and a third for symmetric active/passive mode. The Association Management program documentation page provides additional details. For instance, in client/server mode the server keeps no state for each client, but uses a fast algorithm and a private value to regenerate the cookie upon arrival of a client message. A client sends its designated public key to the server, which generates the cookie and sends it to the client encrypted with this key. The client decrypts the cookie using its private key and generates the key list. Session keys from this list are used to generate message authentication codes (MAC) which are checked by the server for the request and by the client for the response. Operational details of this and the remaining modes are given in the Internet Draft referenced elsewhere on this page.

2.3 Identity Schemes

The timestamped digital signature scheme provides secure server authentication, but it does not provide protection against masquerade, unless the server identity is verified by other means. The PKI security model assumes each client is able to verify the certificate trail to a trusted certificate authority (TA), where each descendent client must prove identity to the immediately ascendent server by one or more means, such as a credit card number or PIN. While Autokey supports this model by default, in a hierarchical ad-hoc sensor network, especially with server discovery schemes like Manycast, hicking the certificate trail and proving identity at each stratum level is not practical.

Our model is that every member of a closed group, such as might be operated by a timestamping service, be in possession of a secret group key. This could take the form of a private certificate or one or another identification schemes described in the literature. These schemes are at the heart of the security model preventing masquerade and middleman attacks. While the scheme described in RFC-2875 is based on a ubiquitous Diffie-Hellman infrastructure, it is expensive to generate and use when compared to others in the literature. There are four schemes now implemented in Autokey to prove identity: one using private certificates (PC), a second using trusted certificates (TC), a third using a modified Schnorr algorithm (IFF aka Identify Friendly or Foe) and the fourth using a modified Guillou-Quisquater (GQ) algorithm.

The PC scheme is very simple and involves the use of a private certificate as group key. A certificate is designated private by a X509 Version 3 extension field when generated by utility routines in the NTP software distribution. The certificate is distributed to all other group members by secret means and is never revealed to others. A client is marked as trusted immediately upon reception of the first response and authentic when the first signature is verified. This scheme is cryptographically strong as long as the private certificate is protected; however, it can be very awkward to refresh the keys or certificate, since new values must be securely distributed to a possibly large population and activated simultaneously.

The TC scheme involves a conventional certificate trail and a sequence of certificates, each signed by an issuer one stratum level lower than the client and terminating on a self signed trusted certificate, as described in RFC-2510. A certificate is designated trusted by a X509 Version 3 extension field when generated by utility routines in the NTP software distribution. A client at each stratum *n* obtains all certificates leading to the trusted certificate from its server, then requests the server to sign the client certificate. Subsequently, all of these certificates can be provided to dependent clients on demand. In this scheme keys and certificates can be refreshed at any time, but a masquerade vulnerability remains unless a request to sign a client certificate is validated by some means such as reverse-DNS lookup.

The two remaining schemes involve a cryptographically strong challenge-response exchange. Both schemes start when the client sends a nonce to the server, which then rolls its own nonce, performs a mathematical operation and sends the results along with a message digest to the client. The client performs another mathematical operation and verifies the results match the message digest. Of the two schemes, IFF is used when the certificate is generated by a third party, such as a commercial service and in general has the same refreshment and distribution problems as PC. On the other hand, when certificates are generated by routines in the NTP distribution, GQ is the obvious choice. In this scheme the secret group key is not used directly, but hidden in a mathematical expression involving a public/private key pair which can be refreshed at any time. The public member is conveyed in the certificate by a X509 Version 3 extension field which changes for each regeneration of key pair and certificate.

2.4 Key Management

The cryptographic data used by Autokey are generated by a utility program designed for this purpose. This program generates several files, containing MD5 symmetric keys, RSA and DSA public keys, identity group keys and self signed X.509 Version 3 certificates. The certificate format and contents conform to RFC-3280, although with some liberty in the interpretation of extension fields. During generation, a private/public key pair is chosen along with a compatible message digest algorithm. During operation, a client can obtain this and any other certificate held by the server. The client can also request a server acting as a certificate authority to sign and return a certificate.

The lifetimes of all cryptographic values are carefully managed and frequently refreshed. While public keys and certificates have lifetimes that expire only when manually revoked, random session keys have a lifetime specified at the time of generation. Ordinarily, key lists are regenerated about once per hour and other public and private values are refreshed about once per day. Appropriate scripts running from a Unix cron job about once per month can automatically refresh public/private key pairs and certificates without operator intervention. The protocol design is

specially tailored to make a smooth transition when these values are refreshed and to avoid vulnerabilities due to clogging and replay attacks.

2.5 Leapseconds Table

The National Institute of Science and Technology (NIST) archives an ASCII file containing the epoch for all historic and pending occasions of leap second insertion since 1972. While not strictly a security function, the Autokey scheme provides means to securely retrieve the leapseconds table from a server or peer. At present, the only function provided is to fetch the leapseconds table via the network; the daemon itself makes no use of the values. The latest version of the nanokernel software for SunOS, Alpha, FreeBSD and Linux cited below retrieves the latest TAI offset via NTP and provides this on request to client applications.

2.6 Present Status

Autokey version 2 has been implemented in a wide range of machine architectures and operating systems using both IPv4 and IPv6 address families. It has been tested under actual and simulated attack and recovery scenarios. The current public software distribution for NTPv4 includes Autokey and also a prototype version of the Manycast autonomous configuration scheme described on the companion Autonomous Configuration page. The distribution is available for download at www.ntp.org.

All four identity schemes described above have been implemented and tested. At present, the means to activate which one is used in practice lies in the parameters and keys selected during the key generation process. There remains some testing to explore modes of interoperation when different schemes are used by different clients and servers in the same NTP subnet.

2.7 Future Plans

The Autokey technology research and development process is basically mature, although refinements may be expected as the proof of concept phase continues with prototype testing in the NTP environment. We believe the technology is ready to exploit in other critical environments such as real sensor networks and critical mission command and control systems. However, what needs to be done first is to advance the standards track process.

The Internet draft on the Autokey protocol specification has been under major revision. The latest draft has been submitted to the STIME task force for review. A high priority is to complete revisions of the draft to include the latest development of the identity schemes. We expect it will be issued for last call by the IETF and launched on the standards track.

3. Autonomous Configuration

The missions considered in this project funded by DARPA also involve autonomous sensors that might be deployed from a helicopter over a battlefield or from a space probe over a planetary surface. As in Autokey, once deployed, the sensor network must operate autonomously using an ad-hoc wireless infrastructure as sensors are deployed or destroyed or the network is damaged or compromised and then repaired. Appropriate algorithms and protocols must be developed to facil-

itate automatic, quasi-optimal configuration of sensor applications in response to network damage and repair and without requiring external management intervention.

Our approach involves IP multicasting and distributed, goal-oriented algorithms that survey the current service topology to discover, authenticate and configure a quasi-optimal forest of spanning trees rooted on the primary service providers. We have developed heuristic algorithms that attempt to minimize a distance metric corresponding to the most accurate time, subject to constraints designed to protect the server and network resources. These algorithms are designed to work in real time with minimal impact on other services that might share the sensor platform.

We have used the Network Time Protocol (NTP) software and the widely distributed NTP synchronization subnet in the Internet as a testbed for distributed protocol development and testing. The ad-hoc deployment, configuration and management of the NTP subnet have features in common with sensor applications, both because of a mostly casual management style, but also because the Internet is inherently unreliable with unpredictable failure and recovery scenarios.

A new paradigm called Manycast has been developed for the automatic detection of NTP servers in the IP multicast near-neighborhood. It can use either symmetric key cryptography or the Autokey protocol described on the Autonomous Authentication page as an integral component in order to cryptograhically verify server credentials. An executive summary of the paradigm and implementation is included below. The Association Management and Automatic NTP Server Configuration Options program documentation pages provide additional details.

3.1 NTP Multicast/Manycast Support

NTP broadcast mode does not extend beyond the local subnet. Where service is intended beyond the local subnet, IP multicasting can be used if supported by the network infrastructure. Various means have been developed using these methods to discover services and resolve addresses, for example. Manycasting is an automatic dynamic discovery and configuration paradigm. It is distinct from anycasting, where a single service provider is selected from a number that may respond to a multicast invitation. Manycasting is designed for highly robust services where multiply redundant respondents are continuously evaluated and quasi-optimal subsets mitigated using engineered algorithms. In general, NTP requires a plurality of servers in order for the mitigation algorithms, which are based on Byzantine agreement methods, to reliably cast out intruders and provide the best timekeeping performance.

The NTP Manycast scheme uses an expanding-ring search with pruning and variable poll rate in order to minimize network overhead. Servers and clients can use multiple group addresses with different scoping rules to segregate traffic. A client trolls the nearby network neighborhood looking for available manycast servers, authenticates them using either symmetric key or public key cryptography and then evaluates their time values with respect to other servers that might be already present or lurking in the vicinity. The intended result is that each manycast client mobilizes client associations with the “best” three nearest available manycast servers, yet automatically reconfigures to sustain this number should one or another degrade, fail or become compromised.

A manycast client begins sending multicast messages to the group address at a moderate rate and minimum time-to-live (TTL), depending on how many servers have already been found. A Many-

cast server in scope and equal or lower stratum replies with its address, which causes the client to mobilize an ordinary unicast association, which then continues as if originally configured. If Autokey is enabled on the client, the NTP and Autokey protocols, run concurrently until the server credentials are verified. When the protocols stabilize, the NTP mitigation algorithms discard all but the best three associations. If necessary, the client increases the TTL in steps until three acceptable associations are found, after which it polls at a lower sustaining rate to discover new servers that might come up.

3.2 Present Status and Future Plans

Manycast mode has been implemented and tested and now in regular operation in our local networks and in the CAIRN research testbed. This support has been incorporated in the public NTP software distribution at www.ntp.org and available for download. The configuration files in all manycast servers and clients are identical and do not require per-host configuration. Initial experience in scenarios involving two dozen primary and secondary servers confirm that the manycast algorithms do in fact reconfigure properly in response to server failures and the NTP subnet reconfigures automatically while preserving good NTP performance.

In the current design, manycast servers respond to client messages at equal and higher strata and clients accept server messages at equal or lower strata. In order to reduce the implosion hazard in very large networks, the mitigation algorithms artificially increase the distance of servers with strata below a configured floor and above a configured ceiling. This scheme is only a preliminary cut at what should be a more sophisticated strategy. This problem was anticipated by Ajit Thyagarajan in his (unfinished) dissertation, but the schemes proposed therein are not practical in their present form. Future work is to evolve these schemes and how they could be implemented in the present framework.

It is likely that some sort of whisper campaign protocol will be necessary in order to balance the load among the available servers. Load leveling can be implemented using an extension field which carries, for example, a list of the current servers mobilized by the Manycast client. A MRU list of recently heard servers is already available in the NTP reference implementation and used for access controls. Manycast servers can use the MRU list and combined extension field lists to compute a decision threshold for each server. Each server compares a random roll to its threshold to determine whether to respond to a client request. Details remain to be worked out.

4. Multivariate Trust Models

The missions considered in this project funded by Army Research Laboratory under a Cooperative Technology Agreement involve autonomous sensors that might be deployed from a helicopter over a battlefield or from a space probe over a planetary surface. While the Autokey protocol requires cryptographic values that are either trusted or untrusted, the goal of this project is to determine from various security related variables maintained in a possibly damaged distributed database the level of trust that can be determined from all the information available. For this purpose the security variables can have either discrete or continuous values reflecting the degree of confidence in some related cryptographic function such as certificate trail authentication or identity verification.

4.1 Background

The conventional approach to computer network security involves security protocols based on public key cryptography, digital signatures and challenge/response identity algorithms. Computers are typically interconnected in a lattice where each computer can communicate with a small number of neighbors and those neighbors with their neighbors and so on. Before a session and possibly during one, the security protocol instantiates cryptographic values such as certificates and parameters, then proceeds through protocol exchanges with each neighbor to verify authenticity and identity.

Network security depends on private and public cryptographic values, some of which are instantiated before deployment and some of which are generated and verified when needed after deployment. The purpose of the security protocol is to distribute and maintain public values such as certificates, confirm authenticity and verify identity credentials. In addition, the protocol must determine when a server has been compromised or its host key, certificate or group has been stolen or presented as bogus. However, when implemented in a sensor network, these protocols and algorithms must not consume excessive overhead in power and bandwith.

The Autokey protocol was originally intended as an optional extension to the Network Time Protocol (NTP), but is directly applicable to other distributed network services and sensor networks. We chose to use NTP as a prototype testbed because it is widely available in the Internet and an extensive body of software is already available and suitable as a test vehicle.

Autokey operates using a number of programmed exchanges which establish operational parameters, hike certificate trails, confirm identity and convey initialization parameters. Once the exchanges are complete, the protocol uses a one-way pseudo-random sequence to bind individual packets to the instantiated parameters. However, in its original form, the Autokey design does not tolerate scenarios where some values may be missing or when the network has become seriously fragmented. The intent of the designs studied in this project is to soften the model where trust and/or identity is not absolute, but calibrated by a metric using evidence accumulated from neighbors and the security environment. The intent is to provide a systematic framework within which degraded or incomplete security functions can be combined and augmented to produce a level of trust appropriate to the application.

4.2 Approach

In conventional security models, trust is inferred by certificate trails, assuming all authorities along the trail are trusted and do not sign certificates from individuals who have not proved identity. The methodology used in PEM and reported in recent papers relies on recommenders as somewhat less trusted than root certificate authorities. The recommenders can themselves have recommenders, each of which adds an increment of trust. Presumably, the weight of multiple recommenders establishes the level of trust sufficient to convince a server to sign a certificate, for example. This model has found application not only to PEM but given mathematical rigor in recent papers.

The problem in sensor networks is that the notion of recommender is hard to pin down, as the trust attributed to the traditional recommender is ultimately derived from sources outside the formal security model. In other words the notion of recommender cannot be identified or quantified in

advance; it must be evolved from the other sensors and available battle damage reports that may vary during operation as sensors are compromised and replaced. It is unclear how the notion of degraded cryptographic strength or compromise can be factored into existing security protocols such as Autokey. This is one of the anticipated research directions in this project.

Our approach also involves the use of classic pattern analysis and classification (PAC) methods, which were first developed well over two decades ago. These methods can be used to classify the outcome of an experiment represented by a feature vector where each component of the vector corresponds to a value resulting from the experiment. A set of feature vectors record the outcome for each participant in some defined protocol. In the original PAC model the vectors are members of an n-dimensional vector space, but it is unclear if a security application will find this useful.

A set of decision planes is defined to classify the possible outcomes of the experiment. The planes take the form of intervals in 1-space, areas in 2-space, volumes in 3-space and so on. In 3-space, for example, there may be many volumes defining nonintersecting or intersecting (ambiguous) outcomes either predefined or developed as the result of a training sequence. Once trained, each feature vector (in 3-space) becomes a point and its classification determined by which volume to resides in.

A similar approach can be used for cryptographic applications, although the decision planes are presumably fixed by design and not probabilistic. The security protocols establish measures of authenticity and identity, while PAC methods combine these measures provided by participating group members to establish a trust relation between each member and each other member of the group.

4.3 Current Status

Certified digital signature schemes provide secure server authentication, but do not protect against masquerade, unless the server identity is verified by other means. Our model is that every member of a closed group, such as might be operated by a timestamping service, be in possession of a secret group key. This could take the form of a private certificate or one or another identification schemes described in the literature. These schemes are at the heart of the security model preventing masquerade and middleman attacks.

As previously described, we have implemented four identity schemes in Autokey: one using private certificates (PC), a second using trusted certificates (TC), a third using a modified Schnorr algorithm (IFF aka Identify Friendly or Foe) and the fourth using a modified Guillou-Quisquater (GQ) algorithm. While the scheme described in RFC-2875 is based on a ubiquitous Diffie-Hellman infrastructure, it is expensive to generate and use when compared to the IFF and GQ schemes. As argued elsewhere, these schemes have varying degrees of cryptographic strength and their availability and success could form one component of a distributed trust metric.

4.4 Future Plans

Our plan is to extend the capabilities of Autokey (or a derivative protocol) to survive in a damaged or compromised network. This includes, but is not limited to, periodic challenge/response exchanges, possibly augmented by Byzantine agreement principles. These could take the form of a random challenge broadcast where nearby neighbors respond with unique identity codes as in

the GQ identity scheme. For instance, a number of different GQ keys could be hidden in the sensor program that would be exposed should some programmatic tripwire be cut.

Other data useful in constructing trust relations include how many neighbor certificates have been verified, the length and redundancy of the certificate trail and which identity exchanges have survived and what algorithm was used. how many recommendations have been received from other peers, whether the peer time has been synchronized, how many clogging attack packets have been detected in the last minute, and so forth. Some of these components may age in the form of an exponential decay.

We expect to use PAC methods to systematically assemble the results of these exchanges and threat assessments to develop trust relations between each group member and all other members. Some feature vector components may not be ideally suited to the traditional PAC model. For example, the case of geographic position or distance from a neighboring sensor and the uncertainty of the actual value may be a factor. A handy interpretation may be the distance in n-space from the particular feature to a defined outcome vector representing the uncertainty or distrust in the measurement.

5. Timekeeping in the Interplanetary Internet

There is a long tradition in the planetary science community of controlling experiments entirely from Earth, although there is some progress using semi-autonomous vehicles for Mars surface exploration. Most missions require some kind of clock to determine windows of communication opportunities, for example. During periods where communication with Earth is not possible, there is need to synchronize clocks among the orbiters, base stations and exploration vehicles participating in the mission.

The Interplanetary Internet (IPIN) consists of Earth stations, Mars planetary orbiters, base stations and exploration vehicles participating in a space mission. This project explores synchronization issues in the IPIN using suitable modifications to the Network Time Protocol (NTP). NTP is widely used to synchronize computers in the Earth Internet and has been deployed in low-orbit Earth orbiters, but not in interplanetary missions. There are three issues which distinguish the Earth Internet from the IPIN: bodies are moving, connectivity between them is necessarily not continuous and transmission delays can be very long.

With support from Jet Propulsion Laboratories (JPL) and NASA, we have begun a study of timekeeping issues in the Interplanetary Internet, specifically for Mars missions and supporting infrastructure. Currently, the clocks for all space vehicles are coordinated on Earth using a software library called SPICE. With new technology, timekeeping in space would be distributed so that timed experiments could be accurately controlled autonomously and without intervention from Earth. There are three issues which distance the new technology from current practice: bodies are moving, connectivity between them is necessarily not continuous and transmission delays can be very long.

The Mars internet includes three segments. The Earth segment consists of the current Internet, including experiment hosts and NASA Deep Space Network (DSN) earth station gateways in California, Spain and Australia. The Mars segment consists of orbiters, base stations and rovers near and on Mars. The DSN segment consists of the space between DSN gateways, space-

craft transport buses and Mars orbiters. The orbiters function as gateways, experiment data buffers and supporting infrastructure, including time synchronization.

The current NTP technology has no provisions for mobile servers and clients, where range and range rates can vary with time, and only minimal provisions for intermittent connectivity. In the Mars internet, orbiters and surface stations may have only intermittent connectivity, while in the DSN segment real-time connectivity is possible only at scheduled opportunities and then only with very long delays. These considerations are mitigated by the fact that ranges and range rates can be predicted with some accuracy from the known positions of the spacecraft bus, orbiters and surface stations using ephemerides maintained by astronomical means.

The mission of this project is to develop an understanding of the timekeeping issues in the IPIN, including Mars missions and beyond. The technology must be able to Extract ephemerides data from mission housekeeping files both on Earth, the spacecraft bus and Mars orbiters to compute range and range rate for useful transmission opportunities. The goals of the project are to

1. Enhance the NTP protocol to encode and transmit ephemerides and/or orbit element data along with NTP timestamp data.

2. Develop new or modified algorithms using these data to determine accurate time and frequency offsets between moving objects.

3. Demonstrate a proof-of-concept in the form of a space simulator using suitably modified components of the existing NTP software distribution embedded in a special purpose discrete time simulator.

5.1 Approach

Our model is that the timescales for all platforms, be they in space or on the surface, run at a constant rate relative to atomic time (TAI). This creates somewhat of a problem in that for an event like a supernova explosion the flash should be observed by all platforms consistent with the quasi-planar wavefront from the source. This requires all platforms calculate the offset from barycentric time (TDB) relative to the local clock and assumed position and velocity relative to barycentric coordinates, which changes continuously. However, the ephemeris for each platform is presumed known by astronomical observation or other means, so that a common-view technique can be used.

Time can be disciplined on a one-way or two-way basis. In a one-way scheme a server with a presumed correct clock broadcasts its time and position at intervals to be determined. Just before each broadcast, the server determines the TDB time according to its local clock and ephemeris and sets the NTP transmit timestamp to this value. The server includes the barycentric coordinates at the time of transmission in an extension field.

Upon reception, the client determines the TDB time according to its local clock and ephemeris. It then calculates the propagation time between the known point of transmission and apparent point of reception and adds this to the transmit timestamp in the packet. The difference between this time and the receive timestamp represents a local clock error estimate; however, an exact value is apparent only after an interative procedure. First, the client adjusts the receive timestamp by the error estimate, then redetermines position according to the ephemeris. This results in a new error

estimate. The procedure iterates until differences between successive error estimates are acceptably small. The final value of the error estimate drives the NTP clock discipline algorithm and closes the loop.

A two-way scheme is similar to the NTP symmetric mode involving two peers. Each peer sends a packet at intervals to be determined. The packet contains the same information as in the one-way scheme and in addition the originate and receive timestamps as in NTP symmetric mode. In addition, the barycentric coordinates for each of the three timestamps is convened in extension fields. The advantage of this scheme is the same as for NTP symmetric mode. Each peer can independently measure the TDB offset of the other as well as the overall roundtrip delay.

We anticipate a network of NTP peers where each platform is in intermittent contact with one or more other peers and where the NTP subnet is continuously evolving and reconfiguring according to a fixed or ad hoc schedule. Existing features of NTP, especially the Manycast autoconfiguration mode are directly applicable, as well as the Autokey authentication scheme.

It may will happen that residual clock frequency offsets may introduce considerable error if the time between updates is relatively long, as would be expected during communication opportunities between Earth and mission spacecraft. After a few measurements the frequency can be disciplined in the usual way, but this affects the position and velocity vectors and residuals with respect to the ephemeris. What makes frequency-induced errors more nasty is that the frequency may fluctuate due to spacecraft thermal cycles and power management. Assuming primary servers on Earth together with ephemerides of the transmitter location, the above scheme continues to refine the residuals and develop global time. Kalman filters or ARIMA methods might be a good tool to deal with the residuals and steer to the best time.

There is some concern that the expense of these calculations, both in processor cycles and thermal management, may not be justified in all cases. For instance, NTP between Mars orbiters and the surface is no different than NTP between Earth orbiters and the ground. In fact, NTP has flown in space before on an AMSAT satellite where the embedded Intel processor ran the same code as used on Earth. This assumes the satellite doesn’t move very far during the roundtrip propagation time for the NTP message and reply. If finer correction is required, orbital elements could be derived from radio rise and set times and corrections computed on the fly.

There has been some discussion on what the Mars orbiters can do with respect to antenna orientation. There is a limited fuel supply to point the antenna to Earth and it may not be a good idea spending fuel to point it at other orbiters or the ground in order to exchange NTP packets. Also, it is not likely the orbiters can communicate with each other using an omnidirectional antenna and low power, at least most of the time. However, omnidirectional antennas would seem to be the choice when communicating with surface platforms. Assuming the surface platforms can discipline their own local clock to some degree of precision, the surface clock could be used as a flywheel to synchronize orbiter clocks as they pass over the platform.

5.2 Current Status

Harish Nair is poking through the JPL SPICE library and documentation for means to predict position and velocity vectors for moving objects in space and on planetary surfaces. He has used the available algorithms and interfaces to convert TDB time to position, which is the basis for the

time transfer technique described above. We have found example data for the Earth, Moon, other planets and an interplanetary mission. While these data are highly useful in initial testing and evaluation, we will eventually need additional interesting scenarios involving Mars orbiters and surface platforms.

We have adapted the NTP software distribution to function both as a daemon for ordinary time synchronization means and also as a simulator for integration in a space simulator. This capability has been integrated in the public release for others to experiment with. A description and status report is on the NTP Discrete Event Simulator page

Initial experimentation with Earth and Mars ephemerides revealed a nasty surprise. Presumably because of observation error residuals and SPICE interpolation, the apparent timescale has a rather large jitter in the order of a second. This might not surprise hardened space hands, but it sure surprised the simulator. However, all this means is that the clock discipline tuning parameters for near-Earth baselines might be quite different for interplanetary baselines and the accuracy and precisions achievable might not nearly be as good.

5.3 Future Plans

The general plan is to evaluate the above approach in detail, especially the impact of ephemerides jitter and oscillator wander. We can simulate the effects of oscillator wander, but the effects of ephemerides jitter will need to be explored in more detail. Our immediate plans are to run experiments with different clock discipline tuning parameters.

6. Infrastructure and Support

The various algorithms used in NTP have been evaluated using a special purpose simulator called ntpsim. The simulator uses algorithms somewhat simplified from those in the reference implementation, but behave very nearly the same under nominal conditions with typical network delay jitter and oscillator frequency wander. However, the NTP algorithms have grown in complexity over the years and some quirks of the reference implementation have cropped up from time to time. This has been most apparent when operators, fearful of dreaded backward time steps, have insisted that clocks always be slewed, rather than stepped, even if they are initially in error by a considerable amount, like a week.

The ntpsim algorithms include a number of heuristic defenses against low probability events, such as transient spikes, mode changes, “clockhopping”, server restarts and so forth. In principle, it would be possible to incorporate these features in ntpsim; however, this would require a good deal of effort and require verification that the features work the same in both simulation and practice.

The Windows PC used for utility word processing and multimedia presentation authoring is getting very old. A replacement has been purchased and installed in the laboratory. At the moment, there is no real alternative to Windows and Office, and these products cost as much as the hardware.

At the last PI meeting this investigator was the only one in the room not using a laptop and projector. Finally, a laptop with screen big and clear enough for limited vision folks to use was found and purchased. Next PI meeting there won’t be anybody without a laptop.

7. 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 preceding 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.

7.1 Papers and Reports

1. Mills, D.L. Public-Key cryptography for the Network Time Protocol. Internet Draft draft-ietf-stime-ntpauth=02.txt, University of Delaware, July 2002, 45 pp.

2. Li, Q., and D.L. Mills. Jitter-based delay boundary prediction of wide-area networks. *IEEE/ACM Trans. Networking 9, 5* (October 2001), 578-590

3. Levine, J., and D. Mills. Using the Network Time Protocol to transmit International Atomic Time (TAI). *Proc. Precision Time and Time Interval (PTTI) Applications and Planning Meeting* (Reston VA, November 2000), 431-439.

4. Mills, D.L., and P.-H. Kamp. The nanokernel. *Proc. Precision Time and Time Interval (PTTI) Applications and Planning Meeting* (Reston VA, November 2000), 423-430.

5. Mills, D.L. A brief history of NTP time: confessions of an Internet timekeeper. Submitted for publication; please do not cite or redistribute.

6. Li, Q., and D.L. Mills. Investigating the scaling behavior, crossover and anti-persistence of internet packet delay dynamics. *Proc. 1999 IEEE GLOBECOM 99 Symposium* (Rio de Janerio, Brazil, December 1999).

7. 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.

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

9. Mills, D.L., T.S. Glassey, and M.E. McNeil. Authentication scheme extensions to NTP. Internet Draft draft-mills-ntp-auth-coexist-01.txt, University of Delaware and Glassey-McNeil Technologies, September 1998, 9 pp.

10. 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.

11. 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.