Network Working Group
Request for Comments: 1129

D. L. Mills
University of Delaware
October 1989

Internet Time Synchronization: the Network Time Protocol

Abstract

This memo describes the Network Time Protocol (NTP) designed to distribute time information in a large, diverse internet system operating at speeds from mundane to lightwave. It uses a returnable-time architecture in which a distributed subnet of time servers operating in a self-organizing, hierarchical, master-slave configuration synchronizes local clocks within the subnet and to national time standards via wire or radio. The servers can also redistribute time information within a network via local routing algorithms and time daemons.

The architectures, algorithms and protocols which have evolved to NTP over several years of implementation and refinement are described in this paper. The synchronization subnet which has been in regular operation in the Internet for the last several years is described along with performance data which shows that timekeeping accuracy throughout most portions of the Internet can be ordinarily maintained to within a few tens of milliseconds, even in cases of failure or disruption of clocks, time servers or networks.

This memo describes the Network Time Protocol, which is specified as an Internet Standard in RFC-1119. Distribution of this memo is unlimited.

Keywords: network clock synchronization, standard time distribution, fault-tolerant architecture, maximum-likelihood estimation, disciplined oscillator, Internet protocol.

Table of Contents

1. Introduction 1
1.1. Performance Requirements 2
1.2. Discussion of Approaches 3
2. Time Standards and Distribution 4
3. Network Time Protocol 5
3.1. Implementation Model 7
3.1.1. Modes of Operation 7
3.1.2. Data Formats 8
3.1.3. State Variables 9
3.2. Procedures 10
3.3. Robustness Issues 11
4. Sample Processing and Selection Operations 13
4.1. Data Filtering 13
4.2. Peer Selection 16
5. Local Clock Design 18
6. NTP in the Internet System 21
6.1. Time Servers 21
6.2. Synchronization Subnet 21
6.3. Performance Analysis 22
7. Future Directions 24
8. References 25

List of Figures

Figure 1. Implementation Model 6
Figure 2. NTP Packet Header 9
Figure 3. Calculating Delay and Offset 11
Figure 4. Offset vs Delay 14
Figure 5. Phase-Lock Loop Model 18
Figure 6. Raw Offsets 22
Figure 7. Filtered Offsets 23
Figure 8. Error Distribution 23

List of Tables

Table 1. Outlyer Selection Procedure 17

1. Introduction

How do hosts and gateways in a large, dispersed networking community know what time it is? How accurate are their clocks? In a recent survey involving 94,260 hosts and gateways of the Internet system [MIL89b], 20,758 provided local time using three time-transfer protocols. About half of the replies had errors greater than two minutes, while ten percent had errors greater than four hours. A few had errors over two weeks. Most local clocks are set by eyeball-and-wristwatch to within a minute or two and rarely checked after that. Many of these are maintained by some sort of battery-backed clock/calender device using a room-temperature crystal oscillator that may drift as much as a second per day and can go for weeks between manual corrections. For many applications, especially distributed internet applications, much greater accuracy and reliability is required.

Accurate, reliable time is necessary for financial and legal transactions, stocks and bonds trading, transportation reservation, dispatch and control, and many other applications involving widely distributed resources. These applications often require multiple, concurrent, real-time access to databases distributed over an internet of many networks. Internet time synchronization is necessary for reliable caching, archiving and transaction sequencing. Even the management functions of the internet and its component networks need accurate time in order to manage control activation, monitoring coordination and significant event timestamping. Coordinated network time is extremely useful when searching for elusive bugs and transients in day-to-day operations.

This paper presents an overview of the architecture, protocol and algorithms of the Network Time Protocol (NTP) used in the Internet system to synchronize clocks and coordinate time distribution. The Internet consists of over 100,000 hosts on about 800 packet-switching networks interconnected by a similar number of gateways. In this paper the capitalized *Internet* refers to this particular system, while the uncapitalized *internet* refers to any generic system of multiple networks interconnected by gateways. While the backbone networks and gateways are carefully engineered for good service, operating speeds and service reliability vary considerably throughout the Internet. This places severe demands on NTP, which must deliver accurate and reliable time in spite of component failures, service disruptions and possibly mis-engineered implementations.

In the remainder of this introductory Section 1, issues in the requirements, approaches and comparisons with previous work are discussed. The architecture of the NTP synchronization subnet, including the primary time references and distribution mechanisms, is described in Section 2. An overview of the NTP protocol is given in Section 3 and further intricacies in a formal specification and implementation guide published elsewhere [MIL89a]. Section 4 describes the algorithms used to improve the accuracy of delay and offset measurements made over statistically noisy internet paths and to select the best clock from among a set of mutually suspicious clocks. Section 5 describes a local clock design based on a first-order, adaptive-parameter phase-lock loop and capable of accuracies to the order of a millisecond, even after extended periods when all synchronization paths to primary reference sources have been lost. The international NTP synchronization subnet of time servers and clients now operating on the Internet is described and its performance assessed in Section 6. Further details on measured performance in the Internet can be found in [MIL89b]. Section 7 discusses further development and issues for future research.

1.1. Performance Requirements

In this paper to synchronize frequency means to adjust the clocks in the network to run at the same frequency, while to synchronize time means to set the clocks so that all agree upon a particular epoch with respect to Coordinated Universal Time (UTC), as provided by national standards, and to synchronize clocks means to synchronize them in both frequency and time. A clock synchronization subnet operates by measuring clock offsets between the various time servers in the subnet and so is vulnerable to statistical delay variations on the various transmission paths. In the Internet the paths involved can have wide variations in delay and reliability. The routing algorithms can select landline or satellite paths, public network or dedicated links or even suspend service without prior notice.

It should be noted that stable frequency synchronization in large subnets requires finely tuned tracking loops and multiple phase comparisons over relatively long periods of time, while reliable time synchronization requires carefully engineered selection algorithms and the use of redundant resources and diverse transmission paths. For instance, while only a few comparisons are usually adequate to determine local time in the Internet to within a few tens of milliseconds, dozens of measurements over many days are required to reliably resolve oscillator drift and stabilize frequency to a few milliseconds per day [MIL89b].

Thus, the performance requirements of an internet-based time synchronization system are particularly demanding. A basic set of requirements must include the following:

1. The primary time reference source(s) must be synchronized to national standards by wire, radio or portable clock. The system of time servers and clients must deliver continuous local time based on UTC, even when leap seconds are inserted in the UTC timescale.

2. The time servers must provide accurate and precise time, even with relatively large stochastic delays on the transmission paths. This requires careful design of the data smoothing and deglitching algorithms, as well as an extremely stable local clock oscillator and synchronization mechanism.

3. The synchronization subnet must be reliable and survivable, even under unstable conditions and where connectivity may be lost for periods up to days. This requires redundant time servers and diverse transmission paths, as well as a dynamically reconfigurable subnet architecture.

4. The synchronization protocol must operate continuously and provide update information at rates sufficient to compensate for the expected wander of the room-temperature crystal oscillators used in ordinary computer systems. It must operate efficiently with large numbers of time servers and clients in continuous-polled and procedure-call modes and in multicast and point-to-point configurations.

5. The system must operate in existing internets including a spectrum of machines ranging from personal workstations to supercomputers, but make minimal demands on the operating system and supporting services. Time-server software and especially client software must be easily installed and configured.

1.2. Discussion of Approaches

There are many ways that hosts distributed throughout a large geographic area can synchronize clocks to UTC. In North America the U.S. and Canada operate broadcast radio services with a UTC timecode modulation which can be decoded by suitable receivers [NBS79]. One approach to time synchronization is to provide timecode receivers at every site where required. However, these receivers are specialized, moderately expensive and subject to occasional gross errors due to propagation and equipment failures.

The U.S. National Institute of Standards and Technology (NIST) (formerly National Bureau of Standards), recently announced a computer time service available to the general public by means of a standard telephone modem [NBS88]. The service is intended for use by personal workstations to set clock-calenders, for example, but would not be suitable for a large population of clients calling on a frequent, regular basis without further redistribution.

In principle, it is possible to use special network facilities designed for time synchronization, such as a dedicated FM or TV subcarrier or timecode rebroadcast by a cable system. For many years AT&T has synchronized digital switching equipment to the Basic Synchronization Reference Frequency (BSRF), which consists of a master oscillator synchronized to UTC and a network of dedicated 2048-kHz links embedded in the transmission plant. AT&T and other carriers are planning to use the Global Positioning System and the LORAN-C radionavigation system to synchronize switches in various areas of the country to UTC. However, neither of these methods would be economically viable for widespread deployment in a large, diverse internet system.

Various mechanisms have been used in the Internet protocol suite to record and transmit the time at which an event takes place, including the Daytime protocol [POS83a], Time protocol [POS83b], ICMP Timestamp message [DAR81a] and IP Timestamp option [SU81]. In the Hellospeak routing protocol [MIL83b] one or more processes synchronize to an external reference source, such as a timecode receiver or time daemon, and the corrections are distributed via routing updates and a minimum-delay spanning tree rooted on these processes. The Unix 4.3bsd time daemon *timed* [GUS85a] uses an elected master host [GUS85b] to measure offsets of a number of slave hosts and send periodic corrections to them.

Experimental results on measured times and delays in the Internet are discussed in [COL88], [MIL83a] and [MIL85b]. Other synchronization algorithms are discussed in [GUS84], [HAL84], [LAM78], [LAM85], [LUN84], [MAR85], [MIL85a], [MIL85b], [MIL89a], [RIC88], [SCH86], [SRI87] and [TRI86] and protocols based on them in [MIL81], [MIL85c], [MIL89a] and [TRI86]. NTP uses techniques evolved from them and both linear systems and Byzantine agreement methodologies. Linear methods for digital telephone network synchronization are summarized in [LIN80], while Byzantine methods for clock synchronization are summarized in [LAM85].

In an internet system involving many networks and gateways a useful approach is to equip a few strategically located time-server hosts (or gateways) with timecode receivers and coordinate time distribution using a suitable internet protocol. However, the success of this approach requires very accurate and reliable mechanisms to process and distribute the time information, since timecode

receivers, time servers and the internet itself cannot be considered wholly reliable. While reliable clock synchronization has been studied using agreement algorithms [LAM85], [SRI87], in practice it is not possible to distinguish the *truechimer* clocks, which maintain timekeeping accuracy to a previously published (and trusted) standard, from the *falseticker* clocks, which do not, on other than a statistical basis. In addition, the algorithms and protocols discussed in the literature do not necessarily produce the most accurate time on a statistical basis and may produce unacceptable network overheads and instabilities in a large, diverse internet system.

The above approach was used in the design of the NTP synchronization mechanisms, which were evolved as the result of numerous experiments, analyses and stepwise refinements over an eight-year period. It became evident that accurate and reliable internet time synchronization can be achieved only through a integrated approach to system design including the primary reference sources,