

Network Time Protocol (Version 3) Specification, Implementation and Analysis

Abstract

This document describes the Network Time Protocol (NTP), specifies its formal structure and summarizes information useful for its implementation. NTP provides the mechanisms to synchronize time and coordinate time distribution in a large, diverse internet operating at rates from mundane to lightwave. It uses a returnable-time design 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 reference time via local routing algorithms and time daemons.

Status of this Memo

This RFC specifies an IAB standards track protocol for the Internet community and requests discussion and suggestions for improvements. Please refer to the current edition of the “IAB Official Protocol Standards” for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Keywords: network clock synchronization, standard time distribution, fault-tolerant architecture, maximum-likelihood estimation, disciplined oscillator, internet protocol, high-speed networks, formal specification.

Preface

This document describes Version 3 of the Network Time Protocol (NTP). It supersedes Version 2 of the protocol described in RFC-1119 dated September 1989. However, it neither changes the protocol in any significant way nor obsoletes previous versions or existing implementations. The main motivation for the new version is to refine the analysis and implementation models for new applications at much higher network speeds to the gigabit-per-second regime and to provide for the enhanced stability, accuracy and precision required at such speeds. In particular, the sources of time and frequency errors have been rigorously examined and error bounds established in order to improve performance, provide a model for correctness assertions and indicate timekeeping quality to the user. The revision also incorporates two new optional features, (1) an algorithm to combine the offsets of a number of peer time servers in order to enhance accuracy and (2) improved local-clock algorithms which allow the poll intervals on all synchronization paths to be substantially increased in order to reduce network overhead. An overview of the changes, which are described in detail in Appendix D, follows:

1. In Version 3 The local-clock algorithm has been overhauled to improve stability and accuracy. Appendix G presents a detailed mathematical model and design example which has been refined with the aid of feedback-control analysis and extensive simulation using data collected over ordinary Internet paths. Section 5 of RFC-1119 on the NTP local clock has been completely rewritten to describe the new algorithm. Since the new algorithm can result in message rates far below the old ones, it is highly recommended that they be used in new implementations. Note that use of the new algorithm does not affect interoperability with previous versions or existing implementations.
2. In Version 3 a new algorithm to combine the offsets of a number of peer time servers is presented in Appendix F. This algorithm is modelled on those used by national standards laboratories to combine the weighted offsets from a number of standard clocks to construct a synthetic laboratory timescale more accurate than that of any clock separately. It can be used in an NTP implementation to improve accuracy and stability and reduce errors due to asymmetric paths in the Internet. The new algorithm has been simulated using data collected over ordinary Internet paths and, along with the new local-clock algorithm, implemented and tested in the Fuzzball time servers now running in the Internet. Note that use of the new algorithm does not affect interoperability with previous versions or existing implementations.
3. Several inconsistencies and minor errors in previous versions have been corrected in Version 3. The description of the procedures has been rewritten in pseudo-code augmented by English commentary for clarity and to avoid ambiguity. Appendix I has been added to illustrate C-language implementations of the various filtering and selection algorithms suggested for NTP. Additional information is included in Section 5 and in Appendix E, which includes the tutorial material formerly included in Section 2 of RFC-1119, as well as much new material clarifying the interpretation of timescales and leap seconds.
4. Minor changes have been made in the Version-3 local-clock algorithms to avoid problems observed when leap seconds are introduced in the UTC timescale and also to support an auxiliary

precision oscillator, such as a cesium clock or timing receiver, as a precision timebase. In addition, changes were made to some procedures described in Section 3 and in the clock-filter and clock-selection procedures described in Section 4. While these changes were made to correct minor bugs found as the result of experience and are recommended for new implementations, they do not affect interoperability with previous versions or existing implementations in other than minor ways (at least until the next leap second).

5. In Version 3 changes were made to the way delay, offset and dispersion are defined, calculated and processed in order to reliably bound the errors inherent in the time-transfer procedures. In particular, the error accumulations were moved from the delay computation to the dispersion computation and both included in the clock filter and selection procedures. The clock-selection procedure was modified to remove the first of the two sorting/discarding steps and replace with an algorithm first proposed by Marzullo and later incorporated in the Digital Time Service. These changes do not significantly affect the ordinary operation of or compatibility with various versions of NTP, but they do provide the basis for formal statements of correctness as described in Appendix H.

Table of Contents

1.	Introduction	1
1.1.	Related Technology	2
2.	System Architecture	4
2.1.	Implementation Model	6
2.2.	Network Configurations	7
3.	Network Time Protocol	8
3.1.	Data Formats	8
3.2.	State Variables and Parameters	9
3.2.1.	Common Variables	9
3.2.2.	System Variables	12
3.2.3.	Peer Variables	12
3.2.4.	Packet Variables	14
3.2.5.	Clock-Filter Variables	14
3.2.6.	Authentication Variables	15
3.2.7.	Parameters	15
3.3.	Modes of Operation	17
3.4.	Event Processing	19
3.4.1.	Notation Conventions	19
3.4.2.	Transmit Procedure	20
3.4.3.	Receive Procedure	22
3.4.4.	Packet Procedure	24
3.4.5.	Clock-Update Procedure	27
3.4.6.	Primary-Clock Procedure	28
3.4.7.	Initialization Procedures	28
3.4.7.1.	Initialization Procedure	29
3.4.7.2.	Initialization-Instantiation Procedure	29
3.4.7.3.	Receive-Instantiation Procedure	30
3.4.7.4.	Primary Clock-Instantiation Procedure	31
3.4.8.	Clear Procedure	31
3.4.9.	Poll-Update Procedure	32
3.5.	Synchronization Distance Procedure	32
3.6.	Access Control Issues	33
4.	Filtering and Selection Algorithms	34
4.1.	Clock-Filter Procedure	35
4.2.	Clock-Selection Procedure	36
4.2.1.	Intersection Algorithm	36
5.	Local Clocks	40
5.1.	Fuzzball Implementation	41
5.2.	Gradual Phase Adjustments	42
5.3.	Step Phase Adjustments	43
5.4.	Implementation Issues	44

6.	Acknowledgments	45
7.	References	46
A.	Appendix A. NTP Data Format - Version 3	50
B.	Appendix B. NTP Control Messages	53
B.1.	NTP Control Message Format	54
B.2.	Status Words	56
B.2.1.	System Status Word	56
B.2.2.	Peer Status Word	57
B.2.3.	Clock Status Word	58
B.2.4.	Error Status Word	58
B.3.	Commands	59
C.	Appendix C. Authentication Issues	61
C.1.	NTP Authentication Mechanism	62
C.2.	NTP Authentication Procedures	63
C.2.1.	Encrypt Procedure	63
C.2.2.	Clustering Algorithm	38
C.2.2.	Decrypt Procedure	64
C.2.3.	Control-Message Procedures	65
D.	Appendix D. Differences from Previous Versions.	66
E.	Appendix E. The NTP Timescale and its Chronometry	70
E.1.	Introduction	70
E.2.	Primary Frequency and Time Standards	70
E.3.	Time and Frequency Dissemination	72
E.4.	Calendar Systems	74
E.5.	The Modified Julian Day System	75
E.6.	Determination of Frequency	76
E.7.	Determination of Time and Leap Seconds	76
E.8.	The NTP Timescale and Reckoning with UTC	78
F.	Appendix F. The NTP Clock-Combining Algorithm	80
F.1.	Introduction	80
F.2.	Determining Time and Frequency	80
F.3.	Clock Modelling	81
F.4.	Development of a Composite Timescale	81
F.5.	Application to NTP	84
F.6.	Clock-Combining Procedure	84
G.	Appendix G. Computer Clock Modelling and Analysis	86
G.1.	Computer Clock Models	86
G.1.1.	The Fuzzball Clock Model	88
G.1.2.	The Unix Clock Model	89
G.2.	Mathematical Model of the NTP Logical Clock	91
G.3.	Parameter Management	93
G.4.	Adjusting VCO Gain (α)	94

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.