Instituto Superior
de Engenharia
Politécnico de Coimbra

NTP

Network Services 1

Tomás Faria Ramos
2022140175

2023/2024

Contents

List of Figures

Chapter 1

Introduction to NTP

WARNING: This document is up for revision. It's currently found incomplete

The **Network Time Protocol (NTP)** is a networking protocol for clock synchronization between computer systems over packet-switched, variable-latency data networks.

NTP is intended to synchronize all participating computers to within a few milliseconds of **UTC**.

It uses the **intersection algorithm** to select accurate time servers and is designed to mitigate the effects of variable network latency.

The protocol is usually described in terms of a **client-server model**, but can as easily be used in **peer-to-peer** relationships where both peers consider the other to be a potential time source.

Implementations send and receive timestamps using **UDP** on **port 123**. They can also use **broadcasting** or **multicasting**, where clients passively listen to time updates after an initial round-trip calibrating exchange.

NTP supplies a warning of any impending leap second adjustments, but no information about local time zones or daylight saving time.

The current protocol is version 4 **(NTPv4)** which is backwards compatible with version 3.

Chapter 2

Clock Strata

NTP uses a hierarchical, semi-layered system of time sources. Each level of this hierarchy is termed a **stratum** and is assigned a number starting with zero for the **reference clock** at the top. A server synchronized with a **stratum** *n* server runs at **stratum** *n + 1*.

The number represents the distance from the **reference clock** and is used to prevent cyclical dependencies in the hierarchy.

2.0.1 Strata

Stratum 0

These are high-precision timekeeping devices such as **atomic clocks**, **GNSS** (including **GPS**) or other **radio clocks**.

They generate a very accurate pulse per second signal that triggers an interrupt and timestamp on a connected computer. These devices are also known as **reference clocks**.

NTP servers cannot advertise themselves as **stratum** 0. A **stratum** field set to 0 in a **NTP** packet indicates an **unspecified stratum**.

Stratum 1

These are computers whose **system time** is synchronized to within a few microseconds of their attached **stratum** 0 devices. **Stratum** 1 servers may

peer with other **stratum** 1 servers for **sanity check** and **backup**.

These are also referred to as **primary time servers**.

Stratum 2

There are computers that are synchronized over a network to **stratum** 1 servers. Often a **stratum** 2 computer queries several **stratum** 1 servers.

Stratum 2 computers may also peer with other **stratum** 2 computers to provide more stable and robust time for all devices in the peer group.

Stratum 3

There are computers that are synchronized to **stratum** 2 servers. They employ the same algorithms for peering and data sampling as **stratum** 2 and can themselves act as servers for **stratum** 4 computers, and so on.

Other strata

The upper limit for **stratum** is 15; **stratum** 16 is used to indicate that a device is unsynchronized.

Chapter 3

NTP Message Format

NTP uses two types of messages: **clock synchronization** and **NTP control messages**. All **NTP messages** mentioned in this document refer to **NTP clock synchronization messages**. **NTP control messages** are used in environments where network management is needed.

A clock synchronization message is encapsulated in a UDP message. The main fields are:

- LI (Leap Indicator): A 2-bit leap indicator. When set to 11, it warns of an alarm condition (clock unsynchronized); when set to any other value, it is not to be processed by **NTP**;
- VN (Version Number): A 3-bit version number that indicates the version of **NTP**. The latest version is version 4;
- Mode: A 3-bit code that indicates the work mode of **NTP**. This field can be set to these values:
  - 0: reserved;
  - 1: symmetric active;
  - 2: symmetric passive;
  - 3: client;
  - 4: server;

  - 5: broadcast or multicast;
  - 6: **NTP** control message;
  - 7: reserved for private use.

- Stratum: An 8-bit integer that indicates the **stratum** level of the local clock;
- Poll: An 8-bit signed integer that indicates the maximum interval between successive messages, which is called the poll interval;
- Precision: An 8-bit signed integer that indicates the precision of the local clock;
- Root Delay: **Round-trip delay** to the primary reference source;
- Root Dispersion: The maximum error of the local clock relative to the primary reference source;
- Reference Identifier: Identifier of the particular reference source;
- Reference Timestamp: The local time at which the local clock was last set or corrected;
- Originate Timestamp: The local time at which the request departed from the client for the service host;
- Receive Timestamp: The local time at which the request arrived at the service host;
- Transmit Timestamp: The local time at which the reply departed from the service host for the client;
- Authenticator: Authentication information.

Chapter 4

Clock Synchronization Algorithm

A typical **NTP client** regularly polls one or more **NTP servers**. The client must compute its **time offset** and **round-trip delay**.

Time offset θ is the difference in absolute time between the two clocks. It is defined by

$$\theta = \frac{(t_1 - t_0) + (t_2 - t_3)}{2}$$

and the **round-trip delay** δ by

$$\delta = (t_3 - t_0) - (t_2 - t_1)$$

where:

- t_0 is the client's timestamp of the request packet transmission;
- t_1 is the server's timestamp of the request packet reception;
- t_2 is the server's timestamp of the response packet transmission;
- t_3 is the client's timestamp of the response packet reception.

Chapter 5

Software Implementations

5.0.1 Windows Time

All **Microsoft Windows** versions since **Windows 2000** include **Windows Time Service (W32Time)**, which has the ability to synchronize the computer clock to an **NTP server**.

W32Time was originally implemented for the purpose of the **Kerberos version 5** authentication protocol, which required to be within 5 minutes of the correct value to prevent **replay attacks**. The network time server in **Windows 2000 Server** (and **Windows XP)** does not implement **NTP** disciplined synchronization, only locally disciplined synchronization with **NTP/SNTP** correction.

5.0.2 OpenNTPD

OpenNTPD is a **NTPv3/SNTPv4** implementation with a focus on security and encompassing a privilege separated design. Whilst it is aimed closely at the simpler generic needs of **OpenBSD** users, it also includes some protocol security improvements while still being compatible with existing **NTP servers**.

The simpler code base sacrifices accuracy, deemed unnecessary in this use case. A portable version is available in Linux package repositories.

Chapter 6

Security Concerns

Because adjusting system time is generally a privileged operation, part or all of **NTP** code has to be run with some privileges in order to support its core functionality.

A **stack buffer overflow** exploit was discovered and patched in 2014. Apple was concerned enough about this vulnerability that it used its auto-update capability for the first time. On systems using the reference implementation, which is running with root user's credential, this could allow unlimited access. Some other implementations, such as OpenNTPD, have smaller code base and adopted other mitigation measures like privilege separation, are not subject to this flaw.

NTP servers can be susceptible to man-in-the-middle attacks unless packets are **cryptographically signed for authentication**. The computational overhead involved can make this impractical on busy servers, particularly during DoS attacks. NTP message spoofing from a man-in-the-middle attack can be used to alter clocks on client computers and allow a number of attacks based on bypassing of **cryptographic key expiration**.

NTP has been used in DDoS attacks. A small query is sent to an **NTP server** with the return IP address spoofed to be the target address. Similar to the DNS amplification attack, the server responds with a much larger reply that allows an attacker to substantially increase the amount of data being sent to the target.

6.1 Secure Extensions

NTP itself includes support for authenticating servers to clients. **NTPv3** supports a **symmetric key mode**, which is not useful against MITM. The public key system known as "autokey" in **NTPv4** adapted from IPSec offers useful authentication, but is not practical for a busy server. Autokey was also later found to suffer from several design flaws, with no correction published, save for a change in the message authentication code.

Network Time Security (NTS) is a secure version of **NTPv4** with **TLS**. The main improvement over previous attempts is that a separate "key establishment" server handles the heavy **asymmetric cryptography**, which needs to be done only once. If the server goes down, previous users would still be able to fetch time without fear of MITM. **NTS** is currently supported by several time servers.

Microsoft also has an approach to authenticate **NTPv3**/**SNTPv4** packets using a **Windows domain identity**, known as **MS-SNTP**.