# NTP Server Clock Adjustment with Chrony

**Chapter** · January 2021

DOI: 10.1007/978-981-15-6198-6_16



**3 authors:**

Amina Elbatoul Dinar
Université Abdelhamid Ibn Badis Mostaganem
**22** PUBLICATIONS **42** CITATIONS
SEE PROFILE

Boualem Merabet
University Centre of Naama
**79** PUBLICATIONS **385** CITATIONS
SEE PROFILE

Samir Ghouali
University Mustapha Stambouli of Mascara
**81** PUBLICATIONS **204** CITATIONS
SEE PROFILE

# NTP Server Clock Adjustment with Chrony


**Amina Elbatoul Dinar, Boualem Merabet, and Samir Ghouali**

**Abstract** As of now, all servers have an equipment or programming clock to which reference is made to time stamp records, exchanges, messages, and so forth. This clock, albeit structured around a quartz oscillator, floats like any customary watch, which implies this common watch cannot a match to such created machines that are networked and share common resources like file systems. For example, UNIX is a development tool which makes command, based on its work on comparing file modification dates. Similarly, the correlation of log messages from several systems becomes very difficult if they are not at the same time. In this article, we will concentrate on this topic by designing a server utilizing the NTP convention since the primary "focus" of the NTP usage is UNIX frameworks, and to be more explicit, we will see the management of the NTP server with the Chrony tool.

**Keywords** Network time protocol · Servers synchronization · Chrony · Kali linux

## 1 Introduction

On servers, numerous procedures use time [1–4], some record the hour of a client's association in a log, others the hour of a request for an online deals framework for instance. Time exactness turns out to be especially basic when a few machines cooperate; they need a period estimation to synchronize their activities.

A. E. Dinar (✉) · B. Merabet · S. Ghouali
Faculty of Sciences and Technology, Mustapha Stambouli University, Mascara, Algeria
e-mail: amina.dinar@univ-mascara.dz

B. Merabet
e-mail: boualem19985@yahoo.fr

S. Ghouali
e-mail: s.ghouali@univ-mascara.dz

A. E. Dinar
LSTE Laboratory, University Mustapha Stambouli of Mascara, Mascara, Algeria

S. Ghouali
STIC Laboratory, Faculty of Engineering, University of Tlemcen, Tlemcen, Algeria



J. K. Mandal et al. (eds.), *Applications of Internet of Things*, Lecture Notes in Networks and Systems 137, https://doi.org/10.1007/978-981-15-6198-6_16

Companies in the transport sector also have a major interest in supporting their computers systems and networks with servers using NTP and PTP protocols, particularly to ensure more efficient use of their GPS. For an aircraft, flying at an average speed of nearly 1000 km/h, a one-second delay represents a position error of more than 250 m. An hourly reliability including all parameters (zones leap years… etc.) becomes essential.

The high time resolution obtained allows computer and/or robotic accuracy at a scale exceeding one millisecond and thus allows for greater efficiency and production speed, thanks to the coordination of the machines. In this way, the sequencing of these exercises therefore gains in robotization, and the groups working with these machines are then increasingly effective.

For health care facilities, a time synchronization system is particularly important to: ensure proper planning of medical teams; proper administration of medication at the right time and in the right order of prescription; ensure the smooth running of surgical procedures. Datacenters need a time domain in the millisecond range for platform virtualization. The chronology of events also allows errors to be traced on the same millisecond scale: Traceability ensures a backup, or automatic backup, at night requiring an accuracy of about ten seconds. This increases the reliability of daily backups; the time server allows protecting against time deviations caused by an electrical frequency that is not stable enough which varies permanently around 50 Hz in Europe, and the synchronization provided by the server in NTP allows reliable and robust clustering [5–7].

This paper is composed as follows: Sect. 2 presents distinctive synchronization convention framework systems. We focused our study on NTP configurations and variety of reference clocks and sources. Section 3 consists to administrate NTP by Chrony tool to which its task are compared with those of NTP on Sect. 4. Section 5 presents security NTP mechanism and attack detection to ensure it legitimacy and trustworthiness. Finally, Sect. 6 concludes the paper with future directives.

## 2 How to Ensure the Synchronization of Networked Equipment?

### *2.1 Time Protocol*

It is the subject of RFC868; relying on UDP or TCP, it can be summarized as the servers sending a packet containing the time in seconds elapsed since January 1, 1900, at 0H. Time protocol was used by the UNIX timed daemon but its low resolution and the lack of specification of transit time compensation mechanisms led to the study of a more sophisticated protocol [8].

## 2.2 Simple Network Time Protocol (SNTP)

SNTP (SNTPv4) is proposed for essential servers furnished with a solitary reference clock, just as for customers with a solitary upstream server and no reliant customers. The completely created NTPv4 usage is expected for optional servers with various upstream servers and numerous downstream servers or customers. Other than these contemplations, NTP and SNTP servers and customers are totally interoperable and can be intermixed in NTP subnets. A SNTP essential server executing the on-wire convention has no upstream servers with the exception of a solitary reference clock. On a basic level, it is undefined from a NTP essential server that has the alleviation calculations and accordingly fit for moderating between various references tickers [9].

## 2.3 Network Time Protocol (NTP)

It is the subject of RFC1305 and is in its third version. Much more elaborate than time protocol, it allows the creation of networks of NTP entities with multiple redundancies in order to ensure the permanent and reliable synchronization of the machines concerned. The main contribution to the work on NTP is that of D. L. Mills from the University of Delaware [3]. Filtering and selection algorithms and implementation models are defined in NTP. They allow NTP clients to determine the best source of synchronization, eliminate suspicious sources, and correct network transit times at any time. Regarding its implementation, one of the main characteristics of an NTP network is its pyramidal structure [10]. Time references synchronize NTP servers that are directly connected to them. These constitute "stratum" 1, and they will each synchronize several dozen other servers that will constitute "stratum" 2 and so on up to the terminal clients. This principle makes it possible to distribute the load of the servers well while maintaining a "distance" to the relatively small reference sources [11, 12].

NTP is therefore a protocol that allows synchronizing the time of different systems through an IP network. Clients synchronize their clocks with servers. These servers synchronize themselves with other servers and so on. This network is organized in layers called stratums [5, 13].

The network time protocol (NTP): Presented in 1985 as RFC 958 by D. L. Mills and modified in 2010 in form NTPv4 as RFC 5905, the network time protocol is a long standing and wide-spread convention for appropriating time data. NTP utilizes the association—less UDP convention by means of port 123. Its engineering works with a progressively layered correspondence model. Getting time data from stratum 0 sources stratum 1 servers convey an opportunity to layers beneath, etc. With each layer, the stratum number increases, and the feasible precision diminishes. Other than other correspondence models, the unicast mode (customer to server, server to customer) is the most predominant usual way of doing things.

## *2.4 NTP Configuration*

This segment (Section) gives best practices to NTP arrangement and activity. Application of these accepted procedures that are explicit to the network time foundation implementation.

### 2.4.1 Staying up with the Latest

There are numerous renditions of the NTP convention being used, and various usages on a wide range of stages. The practices right now intended to apply by and large to any execution of RFC5905. NTP clients should choose a usage that is effectively kept up. Clients should stay up with the latest on any known assaults on their chose execution and send refreshes containing security fixes when pragmatic.

### 2.4.2 Utilize Enough Time Sources

A NTP execution that is consistent with [RFC5905] takes the accessible wellsprings of time and presents this planning information to advanced crossing point, grouping, and joining calculations to get the best gauge of the right time.

### 2.4.3 Utilize an Assorted Variety of Reference Clocks

When utilizing servers with appended equipment reference timekeepers, it is proposed that various kinds of reference tickers be utilized. Having sources with autonomous executions implies that any one issue is more averse to cause assistance interference [8].

An NTP server can operate in the following modes:

- Simple server mode: it only responds to requests from its clients.
- Active symmetric mode: it asks to be synchronized by other servers and announces to them that it can also synchronize them.
- Passive symmetric mode: same thing but on the initiative of other servers.
- Broadcast mode: intended for local networks, it is limited to the distribution of time information to customers who may be either passive or discover the servers with which they will synchronize.
- Client mode: sends requests to one or more servers.

To synchronize our clocks with our computer network, the most secure and dependable strategy is to have a committed NTP or SNTP server. The architecture in NTPv4 allows a 10× greater time accuracy than the old NTPv3 protocol [14, 15]. The proximity (of the server to the network) provides a minimum latency between the server and our clocks, computers and other equipment.

The implementation of the NTP protocol as well as various drivers used for the connection of time references permit implementing both a simple terminal client and a primary server. The purely NTP part runs on a large number of operating systems: Solaris 2, HP/UX 9.x, SunOS 4.x, OSF/1, IRIX 4.x, Ultrix 4.3, AIX 3.2, A/UX, BSD, Kali Linux. Achieving good accuracy depends on how well the messages are identified at:

**The application level** UNIX is not a real-time system, and it is the least efficient solution but the easiest to implement.

**The level of the kernel software queues** much more precise solution but requires intervention in the kernel.

In our study, our operating system is Linux (Kali), we configure the time of our machine and set the system time with timedatectl, and this command will display the time information of our system:

```
root@amina - kali: - # timedatectl
Local time: mar. 2019 - 08 - 20 23:33:36 CET
Universal time: mar. 2019 - 08 - 20 22:33:36 UTC
RTC time: mar. 2019 - 08 - 20 23:33:36
Time zone: Africa/Algiers (CET, +0100)
System clock synchronized: yes
NTP service: inactive
RTC in local TZ: yes
Use RTC in UTC by calling ‘timsdatectl set - local - rtc 0'.
```

If the clock is not automatically synchronized online, the server time can be configured using set-time: `#sudo timedatectl set - time`.

We list the different time zones by list-timezones: `#timedatectl list - timezones |grep Algeria`.

The time zone is configured using set-timezone: `# sudo timedatectl set - timezoneAfrica/Algeria`.

One of the largest clusters of public NTP servers is called `pool.ntp.org.`. This one is configured by default in most Linux distributions.

Under the latest versions of Linux, the system clock is automatically synchronized in a network. This synchronization is managed by the systemd-timesyncd.service service. More information about this service can be accessed by the command: `# Systemctl status systemd - timesyncd`.

```
$ timedatectl
```

It is therefore possible to synchronize the clock of all the servers on your network by synchronizing each of them with the global NTP network, but as soon as the network grows, it becomes advantageous to have your own NTP server.

There are several other NTP concepts: stepping, slewing, insane time, drift, and jitter.

| remote | refid | st | t | when | poll | reach | delay | offset | jitter |
|---|---|---|---|---|---|---|---|---|---|
| 0.kali.pool.n . | POOL. | 16 | p | - | 64 | 0 | 0.000 | 0.000 | 0.000 |
| 1.kali.pool.n . | POOL. | 16 | p | - | 64 | 0 | 0.000 | 0.000 | 0.000 |
| 2.kali.pool.n . | POOL. | 16 | p | - | 64 | 0 | 0.000 | 0.000 | 0.000 |
| 3.kali.pool.n . | POOL. | 16 | p | - | 64 | 0 | 0.000 | 0.000 | 0.000 |
| ntp.kali.com . | POOL. | 16 | p | - | 64 | 0 | 0.000 | 0.000 | 0.000 |
| -102.130.49.223 | 85.199.214.98 | 2 | u | 792 | 1024 | 337 | 242.941 | -22.544 | 4.332 |
| -ns.bitco.co.za | 41.78.128.17 | 3 | u | 759 | 1024 | 377 | 242.113 | -19.735 | 4.662 |
| -160.119.238.133 | 196.21.187.2 | 2 | u | 786 | 1024 | 377 | 260.651 | -21.193 | 8.842 |
| +chilipepper.can | 145.238.203.14 | 2 | u | 779 | 1024 | 377 | 91.593 | -24.107 | 2.746 |
| *pugot.canonical | 17.253.108.253 | 2 | u | 928 | 1024 | 377 | 85.897 | -30.369 | 7.252 |
| -dbn-ntp.mweb.com | 194.58.204.148 | 2 | u | 791 | 1024 | 377 | 287.697 | -20.849 | 4.199 |
| +golem.canonical | 145.238.203.14 | 2 | u | 311 | 1024 | 377 | 93.789 | -32.492 | 6.703 |

- Remote: specifies the hostname address of time provider that is we are getting time from we have,
- Refid: indicates the type of time reference source that we are connecting to,
- st: specifies the stratum of that time provider,
- When: specifies the number of seconds since the last time poll occurred,
- Poll: indicates the number or seconds between tow time polls,
- Reach: is the key to knowing that NTP is working properly because it is a circular bit buffer, it show us the statue of the last eight NTP messages (377 is the eight octal bit). Each NTP missed packet response is tracked over in the next eight NTP update intervals reach field,
- Offset: specifies the time difference between the local system and the time on the time provider which is in milliseconds [16].

#ntptrace

Localhost: stratum 3, offset −0.046775, synch distance 0.152070

We use also the Ntptrace command to monitor time synchronization, he specifies the time provider's stratum which lists also the time offset between the local system and the time provider.

Indeed, having your own NTP server allows you to: improve synchronization between network servers, reduce traffic due to time synchronizations on the Internet connection, keep servers synchronized even in the event of an Internet outage, and avoid unnecessary strain on the global NTP network.

## 3 NTP Server with Chrony

Kali Linux uses Chrony software as the default NTP server, and this program is installed by the command: #sudo apt-get install chrony.

Then, we configure Chrony by editing the file/etc./chrony/chrony.conf. In this configuration file, there is a certain amount of information, such as: The line beginning with pool indicates the address of the NTP servers (or groups of servers more precisely) to be used and the maximum number of resources to be used. A priori, we can continue to use the default selection.

Drift file indicates the file to use to record the time drift of the server from the pool. It allows you to resynchronize the clock faster. Chrony does not allow customers to synchronize with this time service. The clients' network must be authorized by allowing directive by editing the following line at the end of the file: The address of our network, for example allow 192.168.0/24.

We can launch Chrony and activate it when the server starts: #sudo systemctl enablechrony and # sudo systemctl start chrony.

Chrony listens on UDP port 123 (default port for the NTP service).

Make sure that this port on the firewall was opened, so that clients can synchronize.

As Chrony is now in charge of synchronizing our system clock, we disable systemd-timesyncd by: $ sudo timedatectl set-ntp false.

Chrony provides a command line interface to query and manage Chrony: chronyc. We can therefore display the servers with which we are synchronized by the command:

$ chronyc sources.

The server that starts with ^* is the current time source. Those starting with ^+ are used to calculate an average time, and those starting with ^− are not currently used.

## 4 NTP Chrony Comparison Tasks

NTP underpins the auto key convention to validate servers with open key cryptography. Note that the convention has been demonstrated to be unreliable, and it will be presumably supplanted with a usage of the network time security (NTS) particular, NTP has been ported to even more working frameworks, he incorporates an enormous number of drivers for different equipment reference timekeepers, chrony requires different projects (for example gpsd or ntp-refclock) to give reference time by means of the SHM or SOCK interface, and he can perform helpfully in a situation where access to time reference is irregular. NTP needs normal surveying of the reference to function better, and he can as a rule synchronize the clock quicker and with more time precision. It rapidly adjusts to unexpected changes clock (for example because of changes in the temperature of the precious stone oscillator), and he can perform well not withstanding when the system is clogged for longer timeframes.

Chrony bolsters equipment time stamping on Linux, which permits very exact synchronization on neighborhood systems, and he offers help to work out the addition or misfortune pace of the continuous clock, for example, the clock that keeps up when the PC is killed. It can utilize this information when the framework boots to set the framework time from a redressed adaptation of the ongoing clock. These continuous clock offices are just accessible on Linux, up until now [14].

## 5 NTP Security Mechanisms

In the standard arrangement, NTP groups are exchanged unprotected among client and server. A foe that can turn into a man-in-the-middle is subsequently ready to drop, replay, or change the substance of the NTP parcel, which prompts debasement of the time synchronization or the transmission of bogus time data. A hazard assessment for time synchronization is given in [RFC7384]. NTP gives two inner security systems to ensure legitimacy and trustworthiness of the NTP parcels. The two measures ensure the NTP parcel by methods for a message authentication code (MAC). Neither of them scrambles the NTP's payload, since this payload data is not viewed as secret. Detection of attacks [17] through monitoring administrators should screen their NTP instances to identify assaults. Many known assaults on NTP have specific marks. Ordinary attacks marks include:

1. Zero root parcels: a bundle with a source timestamp set to zero.
2. A bundle with an invalid cryptographic MAC.

The perception of numerous such bundles could show that the customer is enduring an onslaught [18].

## 6 Conclusion, Perspectives and Some Advices

In this article, we concentrate on this subject by designing a server utilizing the NTP convention since the primary focus of the NTP execution is UNIX frameworks, to be progressively unequivocal; we see the administration of the NTP server with the Chrony tool. On a local network, the use of broadcast mode makes it possible to simplify the configuration of clients. Distribute the load well by setting up as many layers as necessary, in particular so as not to overload the public reference servers. Soon, we will use versions of xntpd, only the redacted versions of the DES are exportable from the US, and they carry the word export in their name and are sometimes several numbers late compared to the current version. As xntpd continues to evolve rapidly, our research has led us to study in the near future, how to use UNIX implementation to secure the NTP server by Chrony.

## References


1. Dinar, A.E., Ghouali., S., Merabet, B., Feham, M.: Packet synchronization in an network time protocol server and ASTM Elycsys packets during detection for cancer with optical DNA Biochip. In: International Congress on Health Sciences and Medical Technologies, Tlemcen, Algeria, 5–7 December (2019)
2. Zhao, K.J., Zhang, A.I., Mning, D.Y.: Implementation of network time server system based on NTP. Electronic Test **7**, 13–16 (2008)
3. Li, X.Z.H.: Research on the Network Time Synchronization System Based on IEEE1588. National Time Service Center, Chinese Academy of Sciences (2011)
4. Novick, A., Lombardi, M.: A comparison of NTP servers connected to the same reference clock and the same network. In: Proceedings of the 2017 Precise Time and Time Interval Systems and Applications Meeting, Monterey, California, pp. 264–270, 30 January–2 February (2017)
5. Warrington, R.B., Fisk, P.T.H., Wouters, M.J., Lawn, M.A., Thorn, J.S., Quigg, S., Gajaweera, A., Park, S.J.: Time and Frequency Activities at the National Measurement Institute, Australia. Frequency Control Symposium and Exposition. In: Proceedings of the IEEE International, pp. 231–234 (2005)
6. Mills, D.L.: Internet Time Synchronization: The Network Time Protocol. IEEE Trans. Commun. **39**(10) (1991)
7. IEEE Std 1588-2008: IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. IEEE1588-2008 Standard (2008)
8. https://www.thegeekdiary.com/what-is-the-refid-in-ntpq-p-output/. Last accessed 12 Aug 2019
9. Langer, M., Behn, T., Bermbach, R.: Securing Unprotected NTP Implementations Using an NTS Daemon. In: IEEE International Symposium on Precision Clock Synchronization for Measurement Control and Communication (ISPCS) (2019). https://doi.org/10.1109/ispcs.2019.8886645
10. Lombardi, M., Levine, J., Lopez, J., Jimenez, F., Bernard, J., Gertsvolf, M., et al.: International Comparisons of Network Time protocol Servers. In: Proceedings of the 2014 Precise Time and Time Interval Systems and Applications Meeting, Boston, Massachusetts, pp. 57–66, 1–4 December (2014)
11. Sommars, S.E.: Challenges in Time Transfer Using the Network Time Protocol (NTP). In: Proceedings of the 48th Annual Precise Time and Time Interval Systems and Applications Meeting, California, pp. 271–290, January (2017)
12. Vijayalayan, K., Veitch, D.: Rot at the roots examining public timing infrastructure. In: Proceedings of the 35th Annual IEEE International Conference on Computer Communications, San Francisco, California, pp. 1–9, April (2016)
13. Matsakis, D.: Time and Frequency Activities at the U.S. Naval Observatory. Frequency Control Symposium and Exposition. In: Proceedings of the 2005 IEEE International, pp. 271–224 (2005)
14. Mills, D.L.: RFC1305 - NTPv3. http://rfc-editor.org/. Last accessed 25 Oct 2018
15. Mills, D.L.: RFC4330 - SNTPv4. http://rfc-editor.org/. Last accessed 25 Oct 2018
16. https://chrony.tuxfamily.org. Last accessed 24 Sept 2018
17. Bennabti, S., Dinar, A.E., Merzougui, R., Merabet, B., Ghouali, S.: Risk cryptography planning in telecommunications systems 'CRYP-TS': attack strategy & ethical hacking. In: Conference on Electrical Engineering CEE, Ecole Militaire Polytechnique, Algiers (2019)
18. Hoffmann, M., Toorop, W.: NTP Working Group A. Malhotra Internet-Draft Boston University Intended Status: Informational K. Teichel Expires: 9 January 2020 PTB. https://datatracker.ietf.org/meeting/105/agenda/ntp-drafts.pdf. Last accessed 02 July 2019