Suitability of Network Time Protocol (NTP) for Time Dissemination

PTTI 2020 (revised)

Steven Sommars
Wheaton, Illinois.
stevesommarsntp@gmail.com

This paper was originally published in *Proceedings of the 51st Annual Precise Time and Time Interval Systems and Applications Meeting*, Monterey, California, January 2020. This March 17, 2020 update contains several corrections.

BIOGRAPHY

Steven Sommars was formerly a Consulting Member of Technical Staff at Alcatel-Lucent concentrating on wireless packet data for over 20 years. He has a PhD in physics from Stony Brook University and is a Senior Member of IEEE.

ABSTRACT

The Network Time Protocol, NTP [1], is commonly used for IP network time transfer. Reliance on NTP can carry risks. Servers supporting NTP may be operated with little or no administration. Even well-maintained servers may occasionally supply incorrect network time. NTP packets may be treated by IP networks as an ongoing network attack leading to network drops or delays.

INTRODUCTION

NTP uses connectionless UDP messages to transfer time between computers. When operating in client-server mode an NTP client transmits requests (mode 3 messages) and receives NTP responses (mode 4 messages) from the NTP server. In addition to commercial NTP servers there are several popular open source software implementations including *ntpd*, *chrony* and *ntpsec*. Time transfer accuracy is limited to half the round-trip time, typically on the order of milliseconds. Issues falling into two categories, structural issues and IP network issues are considered in this paper.

PART 1. NTP STRUCTURAL ISSUES

As discussed in [2] NTP servers may deliver erroneous time stamps. Figure 1 shows where the T1, T2, T3 and T4 timestamps are defined.

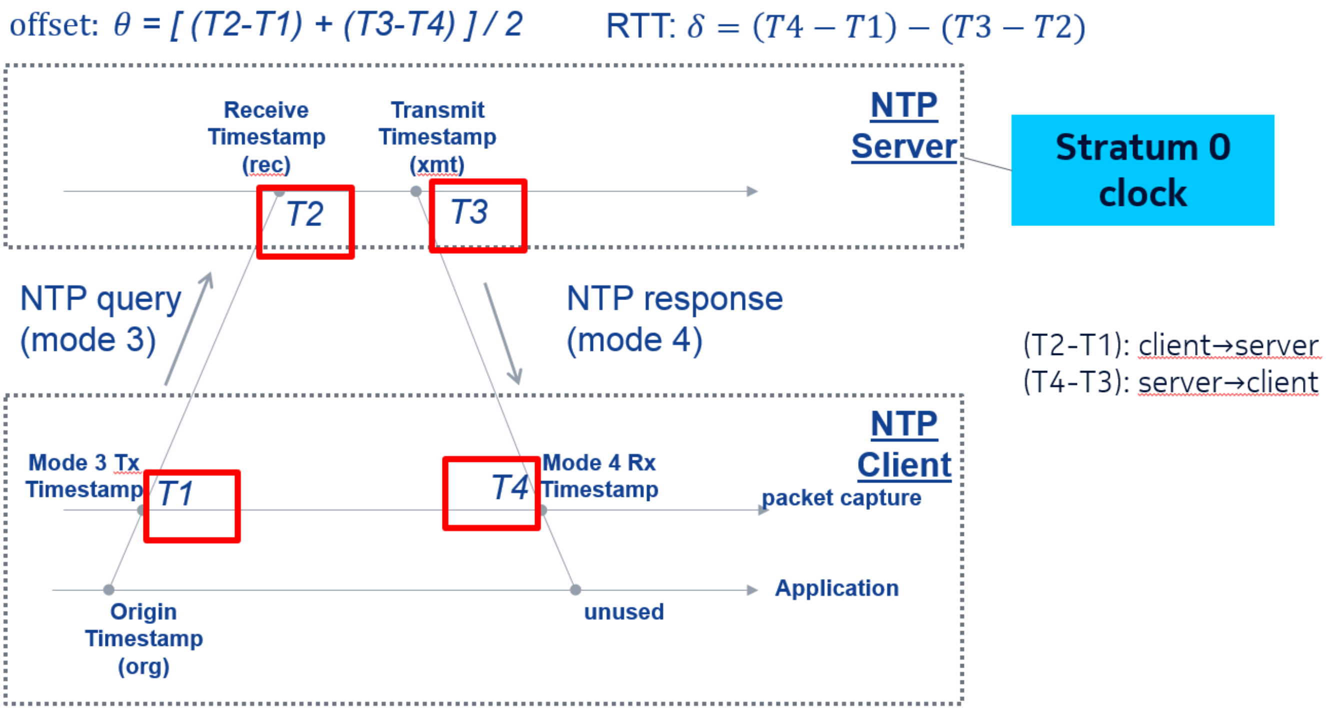

offset: $\theta = [\,(T2-T1) + (T3-T4)\,]\,/\,2$ RTT: $\delta = (T4 - T1) - (T3 - T2)$

Figure 1 NTP client-server T1, T2, T3, T4. Exchange.

In late 2019 an NTP server in Slovenia started returning inaccurate T2 time stamps (red below).

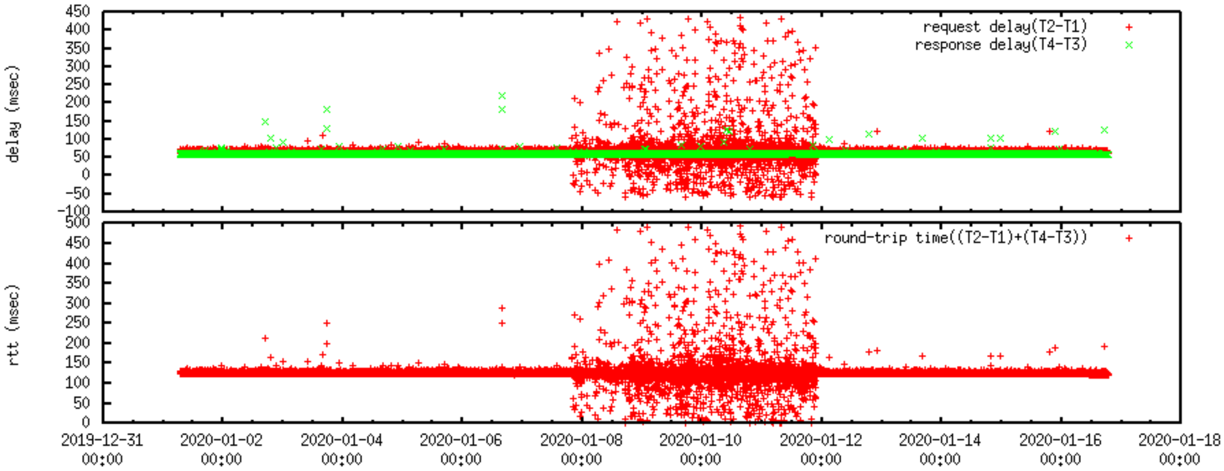


Figure 2 Stale T2 time stamps

Some early NTP server software exhibits a *stale T2* behavior where the T2 timestamp predates the NTP request's arrival. If the server also has high CPU/memory load the NTP request processing may be delayed leading to a late T2 timestamp. The server operator rebooted the server which cleared the problem. Unfortunately the condition recurred within a week. Software updates could have improved the performance; however the manufacturer had dropped support of this unit.

A university in Indiana NTP server had a repetitive one second offset every hour between *XX*:59:01 and *XX*:59:02. An incompatibility between an add-on board and older NTP firmware was solved by a firmware update.

At different times during December 2019/January 2020 four stratum 1 GPS/NTP servers exhibited GPS rollover, leading to time stamp errors of 1024 weeks. Software updates were unavailable.

These NTP server problems are merely recent examples, see [2] for others. Where possible the author notifies the server administrator via email error details. Often the discussion is cordial and sometimes corrections can be found. In reporting errors some troubling patterns have emerged.

- *Zombie servers*. Some NTP servers are zombies, no administrator can be found. Large time errors may persist for days and months.
- *Error denial*. Administrators report “no errors in our logs”, or “do not see in our lab”
- *Disclaiming responsibility*. A “no accuracy guarantees” policies is sometimes voiced.
- *Unsupported equipment*. Manufacturers of commercial NTP servers may supply software updates as needed. Business considerations dictate that support of older equipment be dropped. The NTP server operator may not be aware of support expiration. Even if the server is still under support the latest software may not have been installed.
- *Allowable errors*. Some operators feel that if a plausible trigger for an error can be found, that error is somehow excusable.
- *Limited monitoring*. Until an email report arrives an NTP server operator is often unaware that a problem occurred.
- *Undisciplined backup clocks*. When clock sources attributable to UTC(k) sources are unavailable, an operator may configure an NTP server to use a local, undisciplined (freerunning) clock. Time stamp errors typically grow with time.

Many well-administered NTP servers have operated for years without visible errors, of course. It is difficult for NTP clients to identify them, however.

A robust NTP client should treat its time sources with some skepticism; that is part of defensive programming. Discovering that servers treat accuracy as less than essential continues to be disturbing.

PART 2. NTP AND IP NETWORKS

Early NTP software included many local and remote diagnostic features. The remote queries are *in-band* using the NTP UDP port 123. One of these was the *monlist* query. According to documentation monlist is used to:

> “Obtain and print traffic counts collected and maintained by the monitor facility”

When used remotely a one-IP-packet status request can result in a large response.

```
remote address        port local address      count m ver code avgint  lstint
===============================================================================
27.124.10.238          80 60.248.100.202     33805 7 2      0      3  138976
27.124.10.189          80 60.248.100.202     51527 7 2      0      4  155908
23.224.77.3            80 60.248.100.202     15399 7 2      0    192   10484
etc., 600 counts returned.
```

A monlist quest of a few bytes may trigger a response of over 40KB. Since the request is unauthenticated the IP source is subject to spoofing. Some older NTP software installations are susceptible to use of monlist in a *distributed denial of service* attack (DDoS) with *amplification*. NTP spoofing was a significant problem in the mid 2010’s. Many NTP software developers and users believe the problem no longer exists. Unfortunately the aftereffects still exist.

In mid-October 2019 the round trip delay between suburban Chicago and Maryland suddenly increased, as shown in Figure 3

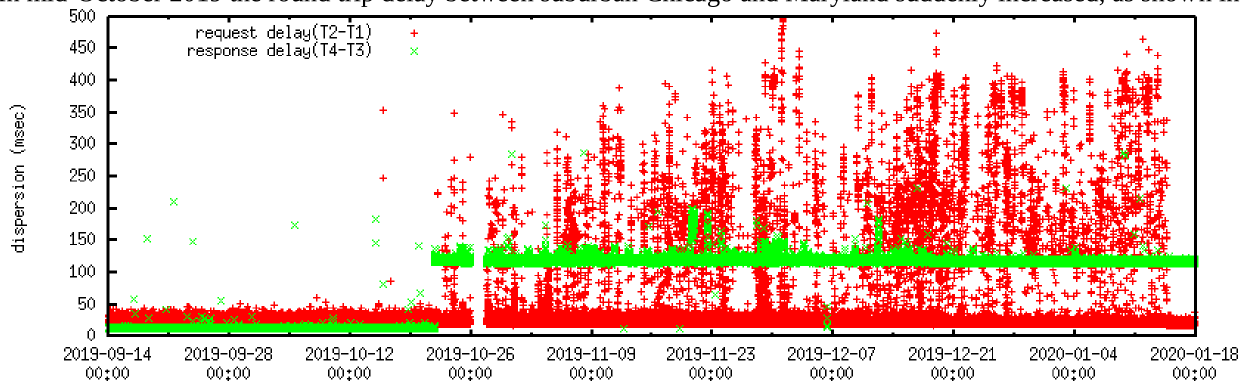


Figure 3 Chicago (Comcast) - Gaithersburg, Maryland (NIST) NTP one-way delays

In addition to increased delay the loss rate between the same client and all NIST IPv4 servers increased.

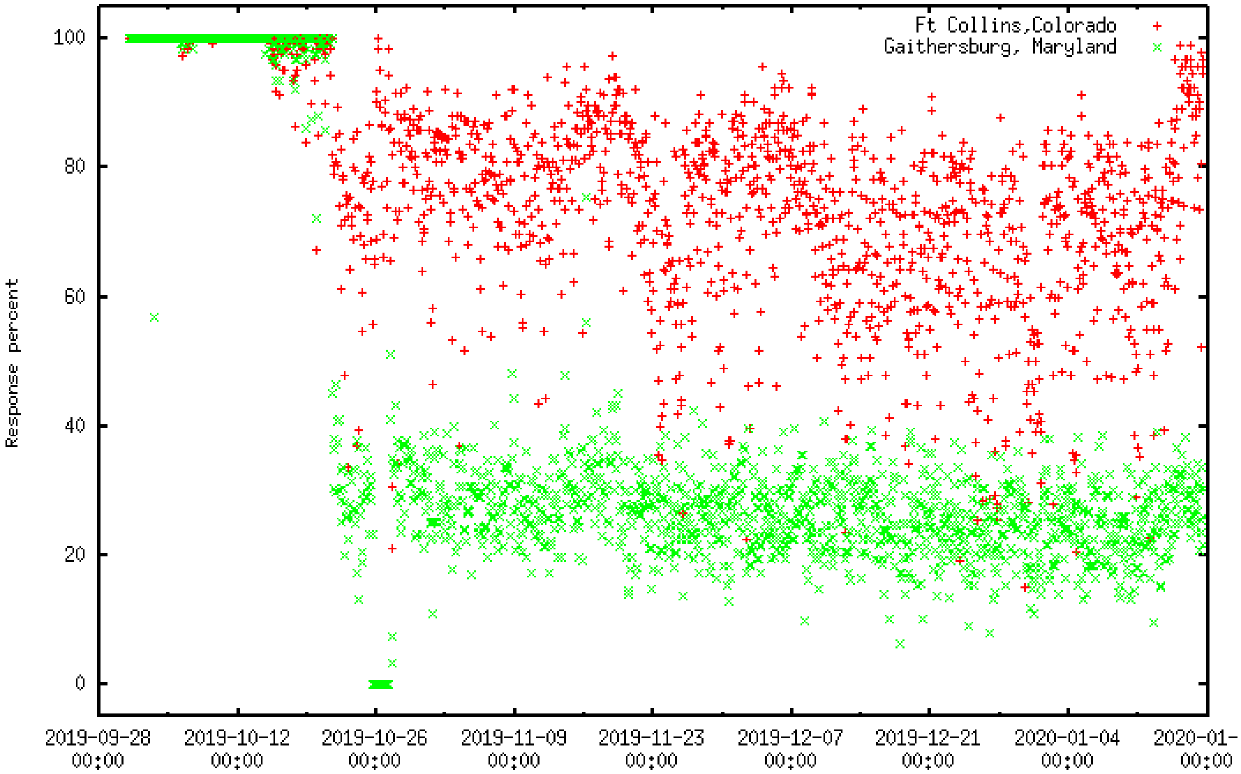


Figure 4 NTP response probability. Chicago (Comcast) and NIST sites.

Figure 4 shows NTP response rates dropping to 20-40%. Simultaneous tests over this same period from a nearby client served by the ISP AT&T had near 100% response rate and no noticeable delay changes. Also the NIST IPv6 servers were unaffected.

In addition to NTP (UDP port 123), the NIST servers support an older protocol, TIME (RFC868, UDP port 37).

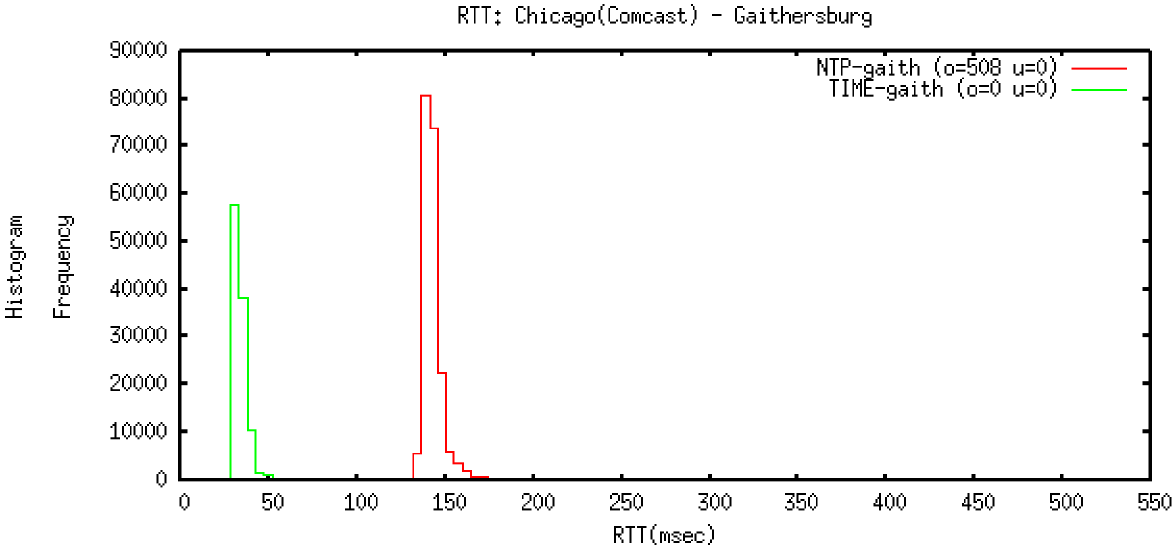


Figure 5 IPv4 NTP and TIME round-trip-times

Figure 5 compares the IPv4 round-trip times for the two protocols between Chicago (Comcast) and Gaithersburg, Maryland (NIST). The RTT is significantly higher for NTP even though the same IP source, destination and network paths are used. No RTT difference was seen for IPv6.

Discussions with NIST staff revealed that they and some customers were seeing similar behavior, see Figure 6.

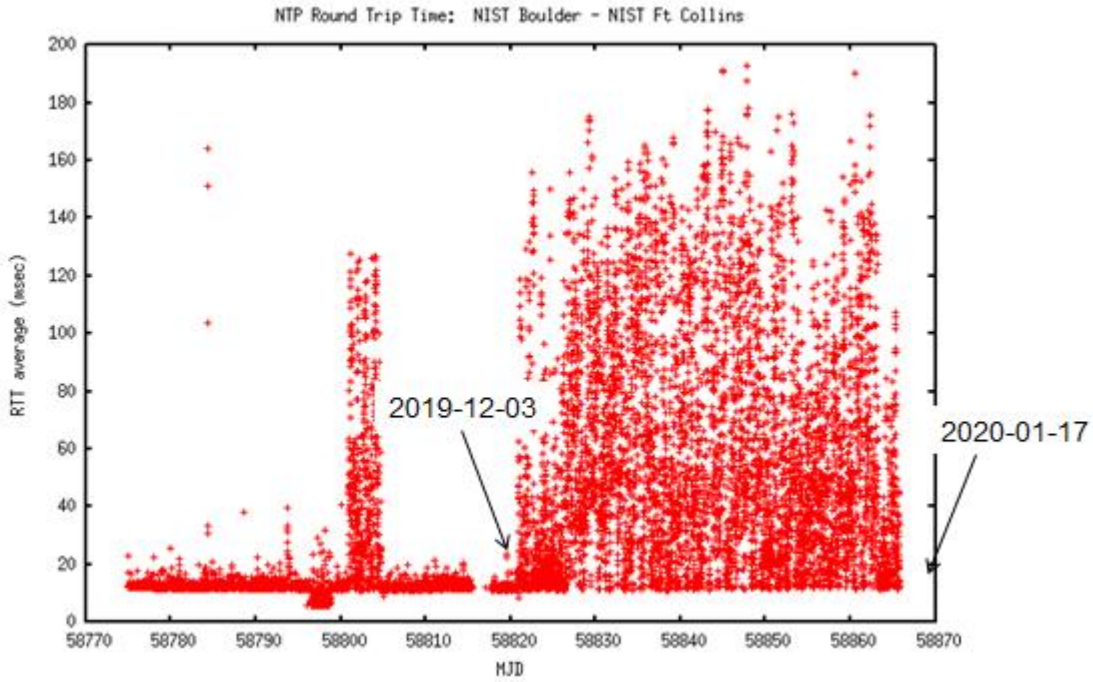


Figure 6 NTP RTT between two NIST sites (data courtesy Mike Lombardi)

In Figure 7 shows that large NTP round-trip times have been seen for several years.

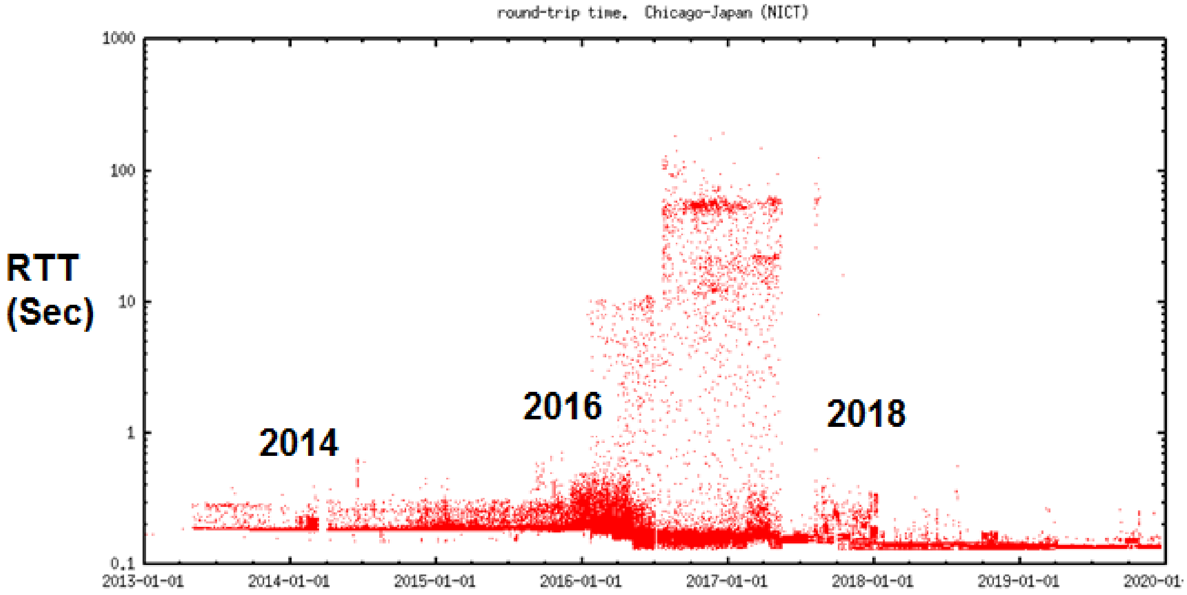


Figure 7 NTP round trip time, Chicago - Japan (NICT)

ICMP echo requests (ping) between Chicago and Japan showed no high delays.

We conclude that NTP traffic on some paths can experience intentional interference in the form of delays and drops. Using IP probes with varying time-to-live values it may be possible to locate where the impairment occurs. The network operator CenturyLink was responsible for the impairment towards the NIST sites. The author's attempts to contact CenturyLink have been unsuccessful. Drops were also seen for NTP servers located in Europe where Telia, Zayo and other network operators seem to be responsible.

Searching through the NANOG email archives at https://mailman.nanog.org/pipermail/nanog/ shows many discussions of the NTP monlist problem and potential remediation techniques. Rate limiting NTP traffic was discussed extensively in 2014. Adding delay to NTP traffic won't lessen the impact of DDoS attacks, so why is it being done? One PTTI participant suggested that the delay may have been an unintentional byproduct of network middleboxes.

Network Time Security (NTS), https://tools.ietf.org/html/draft-ietf-ntp-network-time-security-15, enables NTP server authentication. Extension fields of sizes 200+ bytes are appended to normal NTP packets so that authentication cookies can be exchanged. See Figure 8

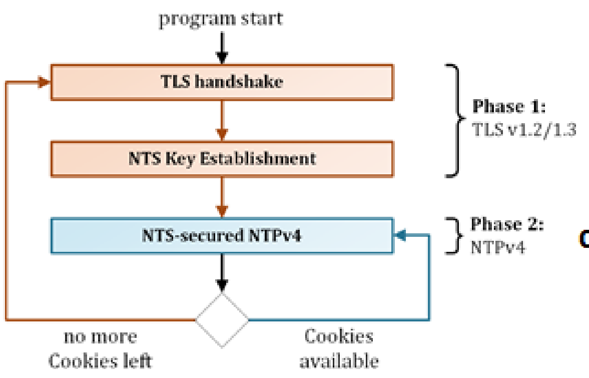


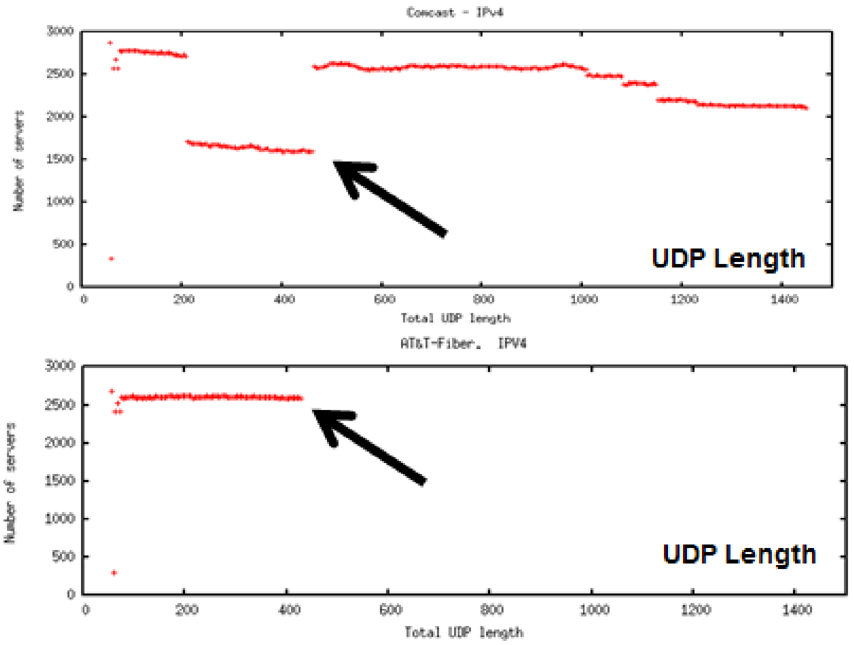
Diagram: Martin Langer

connection oriented. TCP (port 123?)

connectionless. UDP port 123

Figure 8 Network Time Security (NTS)

Another countermeasure to the NTP monlist attacks is based on the size of the NTP packet and interferes with NTS traffic.



ISP=Comcast.
"Hole" 212-460 bytes

Similar seen in
Europe, Japan

ISP=ATT.
Cutoff@ 428 bytes

Figure 9 IPv4 NTP size restrictions

Figure 9 shows the results of sending NTP packets with varying payload sizes to ~2500 NTP servers world-wide. The y-axis shows how many NTP servers respond to each payload size. When the ISP is Comcast a "hole" can be seen for payload sizes between 212 and 460 bytes for about 800 servers. Similar holes are seen when scanning is done from Asia and Europe. The AT&T ISP applied a hard cutoff at payload sizes of 428 bytes. In the NANOG archives and other sources one sees discussions about NTP packet size restrictions being used to counteract the monlist DDoS attack. The NTS authors confirm that size-based NTP blockage is seen and that it interferes with the security-enabled protocol.

Current NTP software corrected the vulnerability years ago. One wonders, "If the NTP problems are solved, why haven't the restrictions been removed?" In part the reason is the NTP problems have not been solved, only lessened. Some network operators relate that their regular IPv4 scans show over 700,000 IPv4 public NTP servers are still vulnerable to the monlist attack. I scanned these servers from multiple sites and found that less than 1% could be used for NTP amplification attacks. The discrepancy remains unexplained.

My discussion with network operators is at an impasse. Few respond to my queries; I'm not a paying customer. The ones with which I've had brief discussions feel that the protections now in place: NTP rate limits and NTP size restrictions have largely solved the problem. There is no incentive to investigate alternative means to restrict monlist traffic. To quote from a recent presentation[3]

> NANOG: "NTT has deployed rate limiters on all external facing interfaces"

NTP is listed as one of the exploitable UDP ports.

Why isn't IPv6 NTP affected? One likely reason are that IPv6-capable NTP servers is on average much newer than IPv4-only gear. Also fewer IPv6-capable NTP servers are advertised publically than IPv4-only. We should note that the previous NANOG presentation explicitly lists IPv6 NTP as an exploitable port, so it is possible that rate limits may affect IPv6 in the future.

CONCLUSION

The free and largely anonymous time service provided by public NTP may be acceptable for casual usage even over a potentially hostile Internet. The consequences of incorrect time are often insignificant.

Clients requiring reliable accurate time should use NTP cautiously. NTP servers can distribute incorrect time, especially when they are operated by unknown parties. Even the most prestigious national measurement institutions may sometimes see NTP server failures resulting in incorrect time.

Protecting NTP messages from in-flight manipulation continues to be a concern. Network Time Security (NTS) is a potential solution. While interference from NTP packet size filters seems likely, moving the protocol to an alternate UDP port might be a solution.

It isn't obvious how to work with network operators to remove or improve the NTP filters. Avoiding intentional drops from overactive NTP rate-limiting filters could also be solved by not using UDP 123 and moving to an alternate port. Abandoning NTP altogether and creating a new time transfer protocol should be considered.

REFERENCES

[1] D. Mills and etal, "RFC 5905. Network Time Protocol Version 4: Protocol and Algorithms Specification," June 2010. [Online]. Available: https://www.ietf.org/rfc/rfc5905.txt.

[2] Sommars, Steven E., "Challenges in Time Transfer using the Network Time Protocol (NTP)," *Proceedings of the 48th Annual Precise Time and Time Interval Systems and Applications Meeting*, Monterey, California, January **2017**, pp. 271-290. [Online]. Available http://leapsecond.com/ntp/NTP_Paper_Sommars_PTTI2017.pdf

[3] Bjarnason, Steinthor , “Withstanding the Infinite: DDoS Defense in the Terabit Era”. NANOG 74 – October 2018. Available online:
https://pc.nanog.org/static/published/meetings/NANOG74/1789/20181001_Bjarnason_Withstanding_The_Infinite__v1.pdf