

# Providing time & frequency at scale

Michael “MC” Cardell Widerkrantz

2020-03-12



# Michael “MC” Cardell Widerkrantz



Research & Development at Netnod.

Contact: [mc@netnod.se](mailto:mc@netnod.se)

<https://www.netnod.se/>

Personal: <https://hack.org/mc/>

# Netnod Internet Exchange

- Internet infrastructure organisation founded in 1997.
- 30 employees.
- 100% owned by a foundation.

# Netnod Internet Exchange

- Largest Internet Exchange Point operator in the Nordics (>160 connected networks in Stockholm).
- Manages the i.root-servers.net DNS server carrying the DNS root zone.
- DNS anycast services to ccTLDs, enterprises and partners.
- Time and frequency services.

# Netnod's Internet Exchange

- Internet Exchange with  $\approx 200$  ASNs.
- WDM transport throughout the Nordics between IX points.
- Metro access between datacenters.

# Netnod's Internet Exchange Presence



# Time & frequency @ Netnod I

- From 1993 Peter Löthberg provided NTP to the public from an atomic clock.
- Since at least 2003 Netnod has provided NTP services.
- Under contract with Swedish Post and Telecom Authority (PTS).

# Time & frequency @ Netnod II

- 4 sites. Working on #5.
- Cesium clocks, 2x at each site.
- Local time scales traceable to UTC (SP) within  $\pm 250$  ns.
- NTP servers in hardware!
- PTP on request.
- Operational in this setup since 2015.
- <https://www.netnod.se/ntp/>







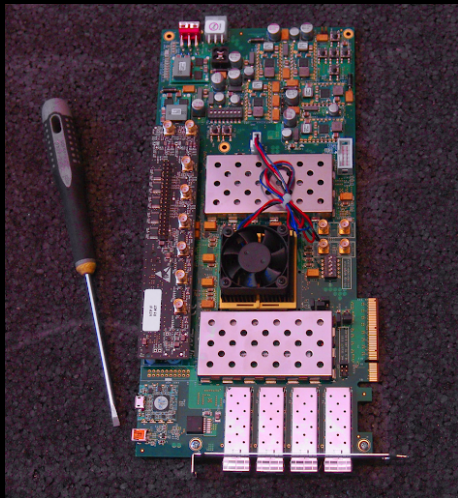
CBL: 222, Delay: 6,21 ns

CBL: 208, Delay: 6,17 ns

CBL: 221, Delay: 6,21 ns



# Xilinx VC709



# NTP Server

- Full wirespeed (4 × 10 GigE) NTP at once.
- DoS protection.
- Clock inputs (PPS + 10 MHz) × 2, connected to both local timescales.
- Automatic failover to second clock if primary fails.

# Verilog NTP server

[https://github.com/Netnod/FPGA\\_NTP\\_SERVER](https://github.com/Netnod/FPGA_NTP_SERVER)

# NTP server addresses

Anycast address: `ntp.se`

Malmö:

- `mmo1.ntp.se`

- `mmo2.ntp.se`

Göteborg:

- `gbg1.ntp.se`

- `gbg2.ntp.se`

Stockholm:

- `sth1.ntp.se`

- `sth2.ntp.se`

Sundsvall:

- `svl1.ntp.se`

- `svl2.ntp.se`

# PTP

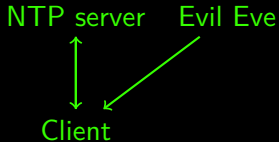
- PTP available on request.
- Delivered over dedicated fibre.



# NTP protocol

- Oldest protocol still in use on the Internet?
- Does anyone remember PDP-11 Fuzzballs?
- Insecure.

# Spoofing



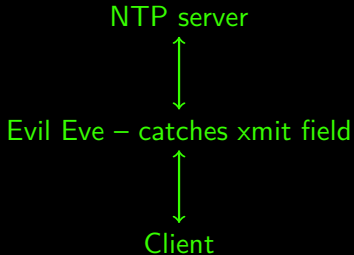
- UDP-based.
- No source identification other than IP address.
- If you can spoof the source, you can lie about time!

# Spoof protection

- NTP `xmit` field — timestamp of sending program.
- On a client, this was originally used for sending what the client thought the time was.
- On modern clients, a spoof protection. 8 random bytes.
- The server simply copies the `xmit` field to the origin time (`org`) field.
- Client verifies that received `org` = sent `xmit`.

# Eve – the woman in the middle

Still open for (wo)man in the middle attacks.



## But NTP supports signatures!

- Symmetric keys only.
- Similar to TSIG in DNS.
- Needs a different key per consumer.
- Can't publish keys – defeats purpose.
- We support symmetric NTP signatures in the FPGA server for specific customers.

# Network Time Security (NTS)

- Network Time Security (NTS) in IETF NTP working group.
- Netnod has been a large part of the standards work.
- <https://datatracker.ietf.org/doc/draft-ietf-ntp-using-nts-for-ntp/>
- “WG state: Submitted to IESG for Publication”

# NTS Protocol

- Really two protocols: NTS-KE & NTP
- NTS-KE: Key exchange over TLS 1.3.
- NTP extended with Extension Fields for NTS specific data, signatures, et c...

# Netnod's NTS server

Proof-of-concept server running at:  
`nts.ntp.se:4443`



# NTS implementations

Netnod is supporting three implementations:

- A Go client with NTP and NTS-KE packages.
- A Python client and server with NTP and NTS-KE modules.
- A Verilog implementation of the NTP extensions.

There are a few other implementations...

# Go implementation

- Implemented by myself and friends on some (remote) IETF hackathons.
- Client: <https://gitlab.com/hacklunch/ntsclient/>
- NTS-KE protocol: <https://gitlab.com/hacklunch/ntske>
- NTP with NTS extension fields:  
<https://gitlab.com/hacklunch/ntp>
- Netnod sponsored with pizza and beer.

# Python PoC server and client

- Consultant Christer “wingel” Weinigel (<http://weinigel.se/>).
- <https://github.com/Netnod/nts-poc-python>

# Verilog NTS

- Consultants Joachim Strömbergson and Peter Magnusson from Assured (<https://assured/>).
- Code not yet available.

# NTS-KE – the key exchange

- Establish NTP server to talk to.
- Establish algorithm.
- Establish session keys: C2S & S2C
- Get initial session cookies.

# NTP Extension Fields for client

- Cookie.
- Placeholder asking for more cookies.
- Signature with C2S key.

# NTP Extension Fields for server

- Signature with S2C key.
- Maybe new cookie for client's later use (if client asked for it).
- Fields can be encrypted optionally.

# Cookies

- Keeps most of the server's state in cookies.
- Exact content of cookies are known only to server.
- Session keys are in cookies, everything the server needs to verify client and sign the NTP packet.



# IETF NTP working group

<https://datatracker.ietf.org/wg/ntp/about/>

# Michael “MC” Cardell Widerkrantz

Contact: [mc@netnod.se](mailto:mc@netnod.se)

<https://www.netnod.se/>