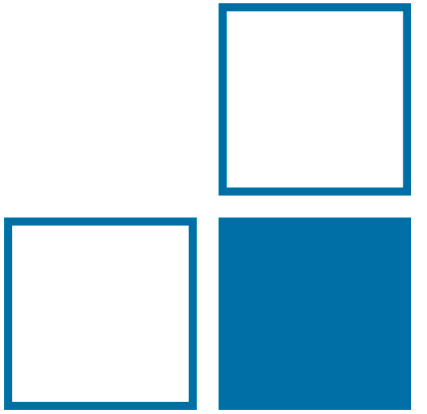
Physikalisch-Technische Bundesanstalt
Braunschweig and Berlin
National Metrology Institute

Time and Frequency Dissemination In Packet–Switched Networks

Dr. Dieter Sibold



Braunschweig

1	Mechanics and Acoustics	2	Electricity
3	Chemical Physics and Explosion Protection	4	Optics
5	Precision Engineering	6	Ionizing Radiation
Q	Scientific-technical Cross-sectional Tasks	Z	Admini-strative Services

Berlin

7 Temperature and Synchrotron Radiation

8 Medical Physics and Metrological Information Technology

AGENDA

- Part 1
  - Introduction
  - Time Synchronization Protocols in Networks
  - PTB's Time Server
  - The Network Time Protocol
  - Leap Second
  - Reference Clocks
  - Comparison with PTP
- Part 2
  - Operational Aspects
  - Security
  - Current Developments

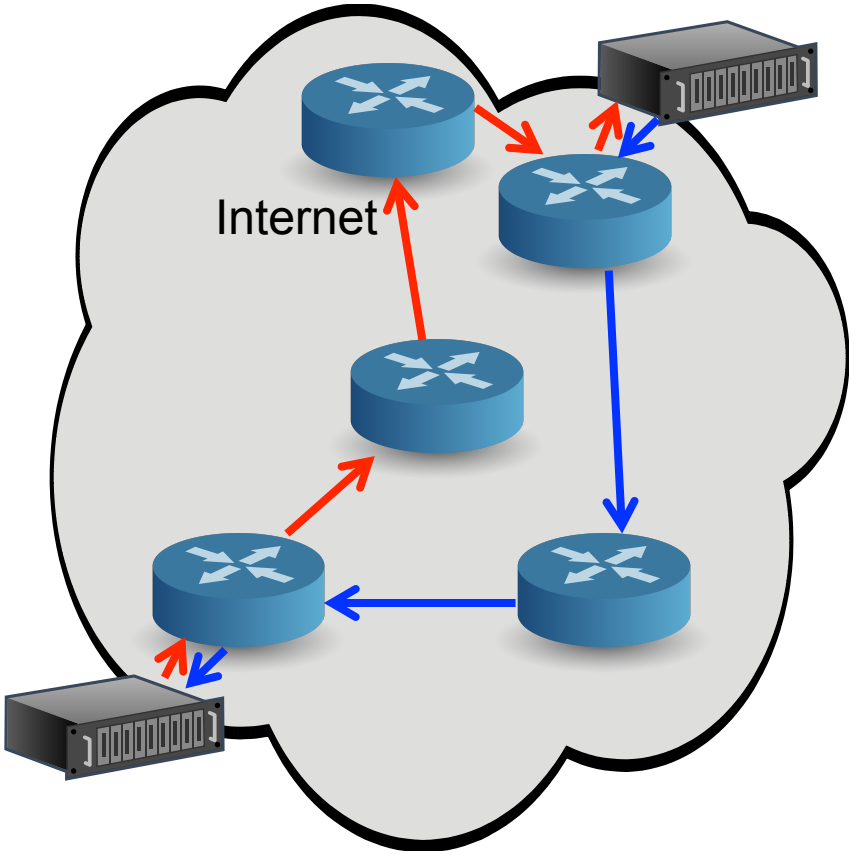
Part 1

Introduction

- Time synchronization of networked devices in packet-switched (computer) networks is necessary in order to:
  - Ensure correctness of processes in distributed and networked systems, e.g.:
    - Manufacturing
    - Stock exchange
    - Billing systems
  - Allow traceability to legal time if required for compliance or legal reasons

Challenges of time synchronization in Packet-switched networks

- No inherent mechanism for time of frequency synchronization (dissemination)
- The path for incoming and outgoing packets may differ
  - Different network bandwidth
  - Different number of hops
  - Each hop adds latency due to queuing
  - Asymmetric path delays (unknown)
- Routing of packets may change from time to time
  - Results in altering path delays for different packets



Major Time Synchronization Protocols

	Synchronous Ethernet (SyncE)	Precision Time Protocol (PTP)	Network Time Protocol (NTP)
Standardization Body	ITU-T	IEEE	IETF
Standard	G.8261, G.8262, G.8264	IEEE 1588	Version 3: RFC 1305 Version 4: RFC 5905
Synchronization of:	Frequency	Time, Frequency	Time
Scope	Syntonization of network devices	Time sync in closed networks	Time sync in closed networks and Internet
Transport	Physics (Layer 1)	Ethernet (L2), IP/UDP (L3)	IP/UDP (L3)
Accuracy	± 4.6 ppm	0.1 µs – 1 ms	10 µs – 100 ms

PTB's Time Servers

- First accessible time servers since April 1999
- Hardware: HP Industrial Workstations with VME-Bus and VME bus based time code generator (TrueTime)
  - `ptbtime1.ptb.de, ptbtime2.ptb.de`

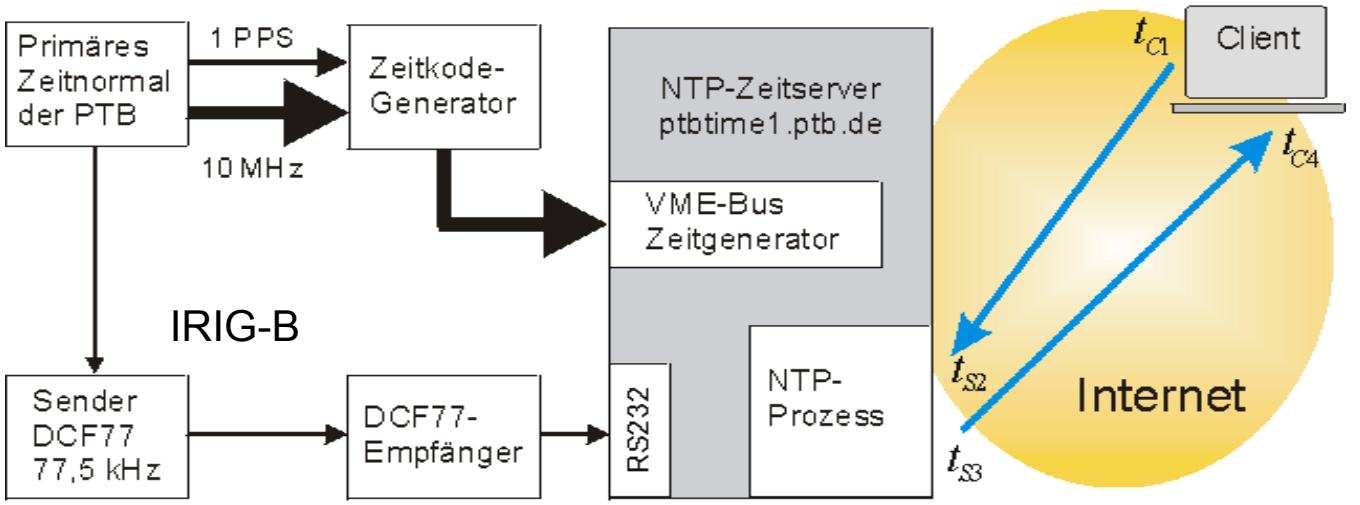
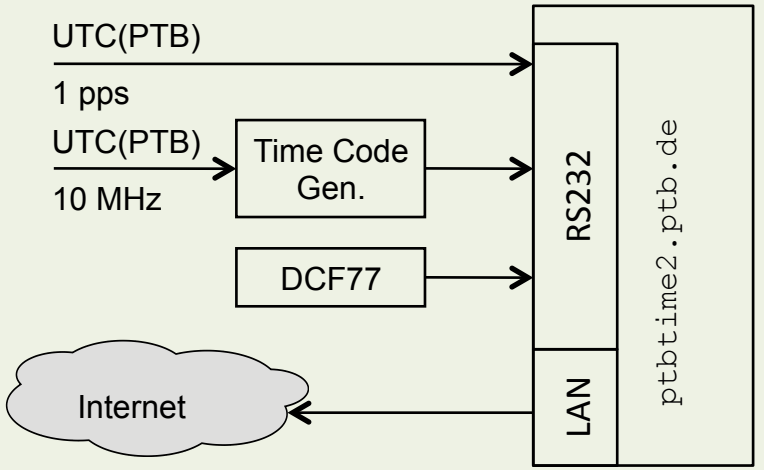


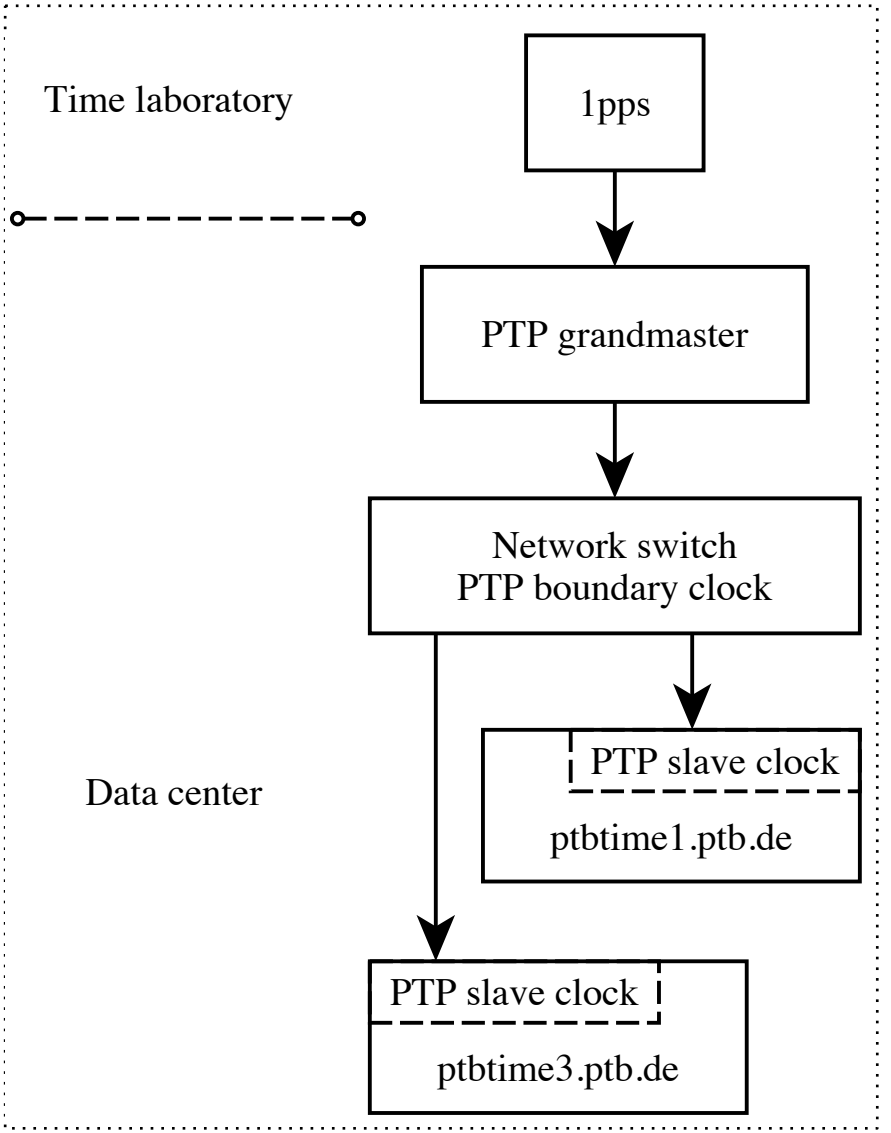
Abb.1

PTB's Time Servers - Today

- Mixture of commodity servers and NTP appliances

  `ptbtime1.ptb.de,`
  `ptbtime2.ptb.de,`
  `ptbtime3.ptb.de`





Network Time Protocol

- Objectives
- History
- NTP's hierarchy and communication modes
- Time calculations
- NTP's header
- NTP's time scale
- Selection and combining algorithm
- System clock adjustment

The Network Time Protocol

Objectives:

- Time synchronization of computer systems in IP networks with respect to UTC with following properties
- Reliability due to:
  - Multi-server approach
  - Selection algorithm to detect false tickers
- Accuracy due to :
  - Computation of network delay
  - Cluster algorithm to find the most accurate set of truechimers
  - Disciplining of time and frequency of the system's clock
- Scalability due to:
  - Hierarchical architecture

The Network Time Protocol

Objectives (continued):

- Strictly monotonically increasing time of the system clock (no backwards time step!)
- Time synchronization is achieved by adjustment of the system clock's frequency $R(t)$

$$T(t) = T(t_0) + R(t - t_0) + D(t - t_0)^2 + x(t)$$

  - Note:
    - The aging term D is neglected by NTP

The Network Time Protocol

Year	Standard	NTP Version	
1985	RFC 958	0	NTP header definition; definition of clock offset and network delay calculation
1988	RFC 1059	1	Complete specification of NTP; first versions of clock filter, selection, and clock discipline algorithms
1989	RFC 1119	2	Addition of NTP control messages and symmetric authentication
1992	RFC 1305	3	Improvements of selection algorithm; introduction of the broadcast mode
2010	RFC 5905	4	Addition of NTP extension fields for additional features; broadcast mode include calibration; IPv6 support
2010	RFC 5906		Autokey: Cryptographic authentication scheme which makes use of NTP's extension fields

The Network Time Protocol

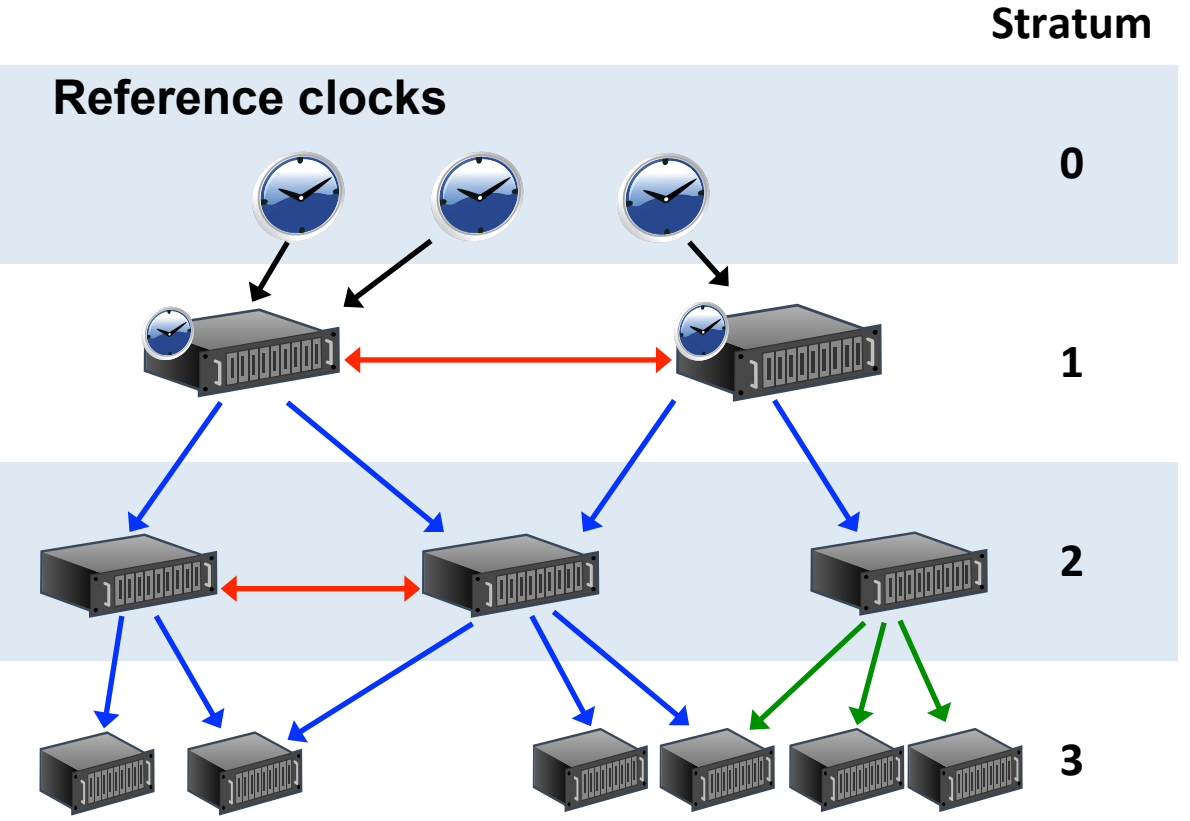
NTP's hierarchy and modes of association

Stratum level i

$0 < i \leq 16$

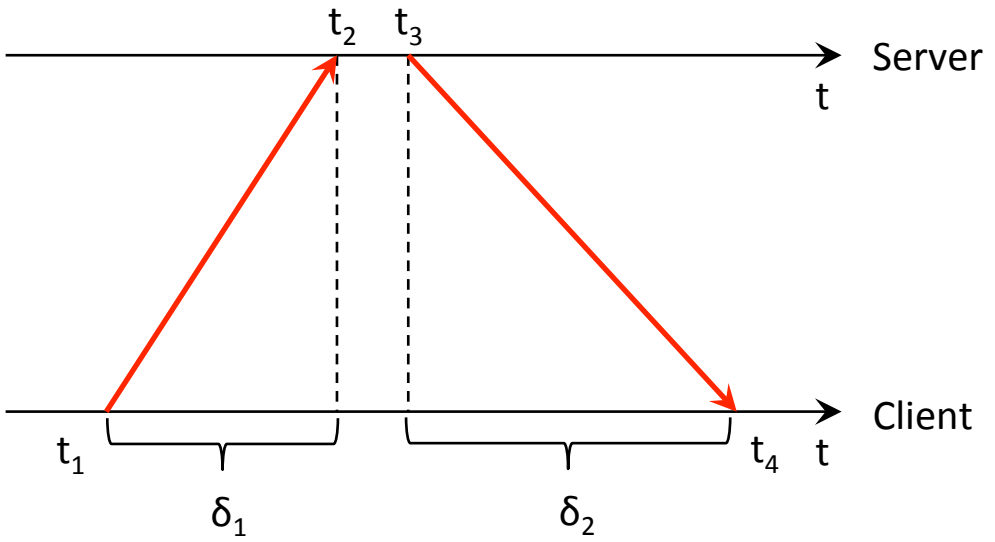
Associations

- Symmetric mode
- Client-Server mode
- Broadcast/Multicast mode



The Network Time Protocol

Delay and offset calculations



Network delay δ is:	$\delta = \delta_1+\delta_2 = (t_4-t_1)-(t_3-t_2)$
Asymmetry factor ξ defined by:	$\delta_1 = \xi\,\delta$ and $\delta_2 = (1-\xi)\,\delta$, with $0 < \xi < 1$
Time offset:	$\theta = ½\,((t_2-t_1)-(t_3-t_4)) + (\xi-½)\,\delta$

Assumption:

Symmetric case:

$\xi = ½$

$\theta = ½\,((t_2-t_1)-(t_3-t_4))$

Analysis of errors

- Error for peer/server association
- Maximum error in the calculation of θ.
  - True offset θ_0
  - Measured offset θ, $T = T_4 - T_1$

$$|\theta_0| \leq |\theta| + \frac{\delta}{2} + \epsilon \,, \text{with } \epsilon = 2(\rho + \phi T)$$

Network delay

Time resolution

Frequency tolerance

Standard implementation: Φ = 15 ppm

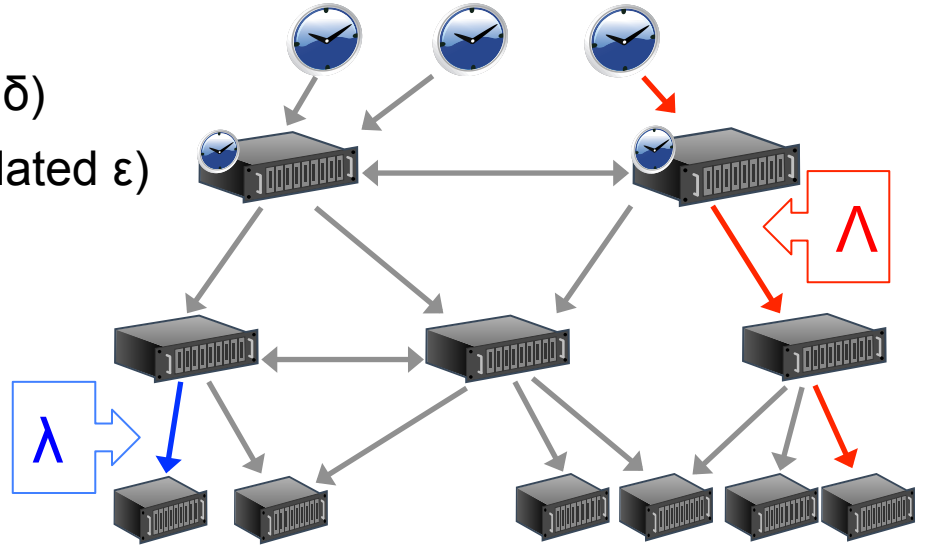
- Dispersion: ε
- Synchronization distance: $\lambda = \frac{\delta}{2} + \epsilon$

Analysis of errors

- NTP tree
  - Accumulation of the synchronization distance λ is the “root distance”

$$\Lambda = \frac{\Delta}{2} + E$$

    - Root delay Δ (accumulated δ)
    - Root dispersion E (accumulated ε)
  - For more details see [Mil2006]



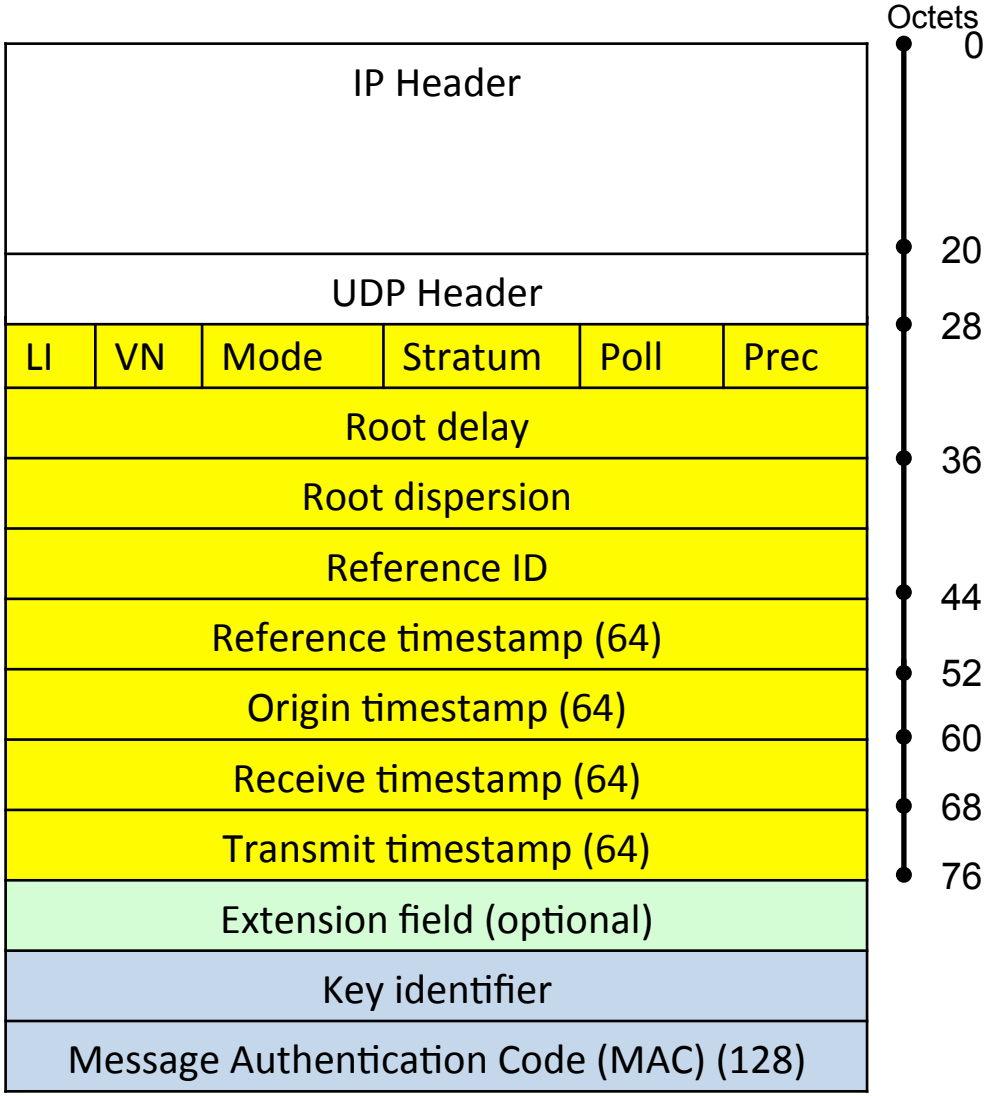
The Network Time Protocol: Packet Header

NTP basic packet header

NTP's extension fields.
Currently only "autokey"

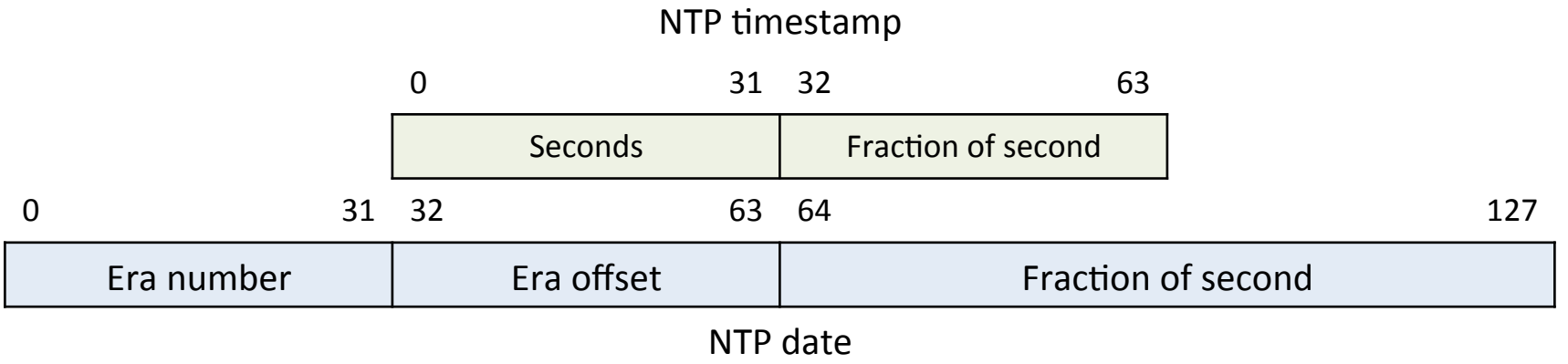
Cryptographic integrity protection
Careful:
- Standard defines MD5 based hash which is deprecated
- Reference implementation supports also SHA1 if the NTPD is compiled against OpenSSL

LI	Leaps second indicator
VN	Version number
Mode	Mode of association
Stratum	0 - 15
Poll	Poll interval $\log_2(\tau)$
Prec	Precision ($\log_2$)



NTP's timestamp and timescale

- NTP timestamps
  - Short format: 64 bit
    - Bit: 0-31: Represents approx. 136 years
    - Bit: 32-63: Fraction of a second with the granularity of 232 ps
  - Long format: 128 bit
    - Bit: 0-31: Era counter increments after
    - Bit: 32-63: Second in the era
    - Bit: 64-127: Fraction of a second with granularity of 5×10^{-20}



The Network Time Protocol

NTP's timestamp and timescale (continued)

- Based on UTC
- Start at the 1. January 1900
- Leap seconds require an adjustment of the counting

Year	M	D	JDN	NTP Date	Era	Timestamp	
-4712	1	1	0	-208.657.814.400	-49	1.795.583.104	First day Julian Era
1	1	1	1.721.426	-59.926.608.000	-14	202.934.144	First day Common Era
1582	10	15	2.299.161	-10.010.304.000	-3	2.874.597.888	First day Gregorian Era
1900	1	1	2.415.021	0	0	0	First day NTP Era 0
1970	1	1	2.440.588	2.208.988.800	0	2.208.988.800	First day Unix Era
1972	1	1	2.441.318	2.272.060.800	0	2.272.060.800	First day UTC
2000	1	1	2.451.545	3.155.673.600	0	3.155.673.600	First day 21st century
2036	2	7	2.464.731	4.294.944.000	0	4.294.944.000	Last day NTP Era 0
2036	2	8	2.464.732	4.295.030.400	1	63.104	First day NTP Era 1
3000	1	1	2.816.788	34.712.668.800	8	352.930.432	

From: [Mil2012]

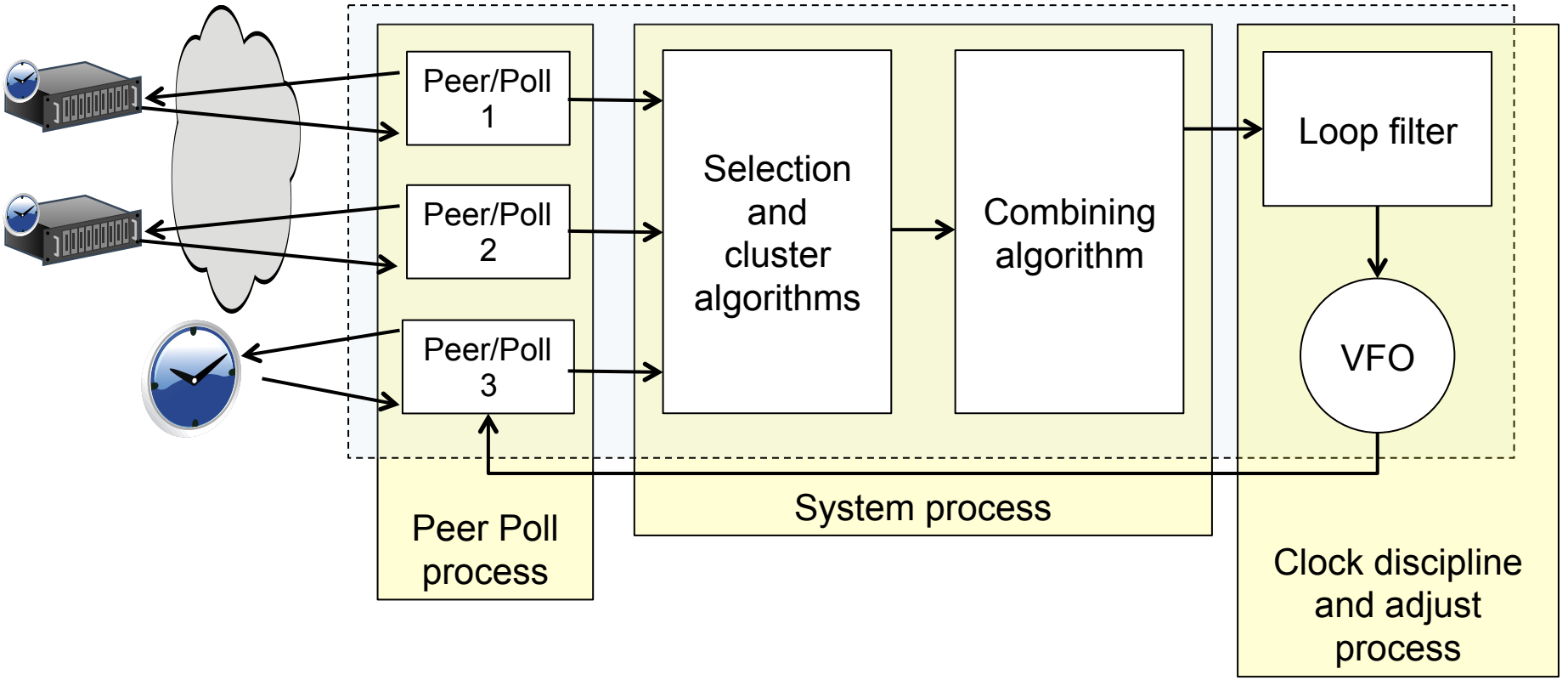
The Network Time Protocol

NTP processes

- Peer poll process
- Selection algorithm
- Cluster and combining algorithm
- Clock discipline and adjust algorithm

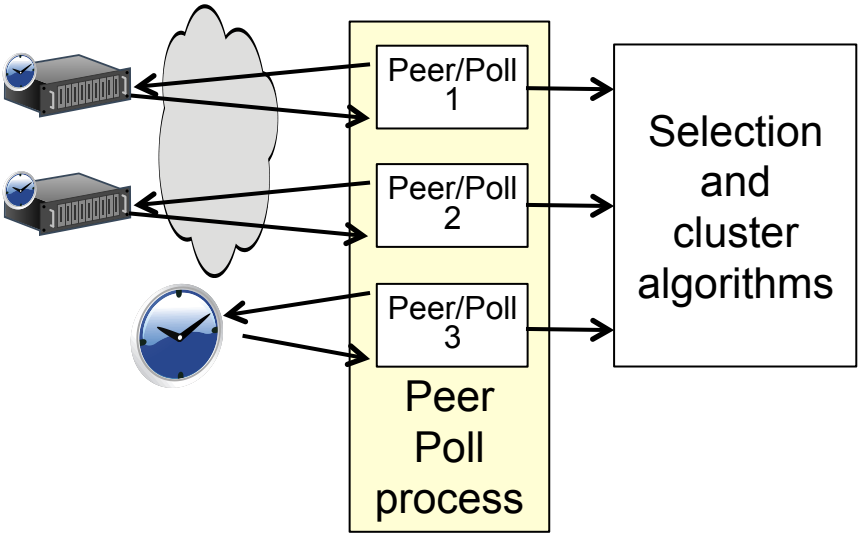
The Network Time Protocol

- **NTP Processes – Overview**



The Network Time Protocol

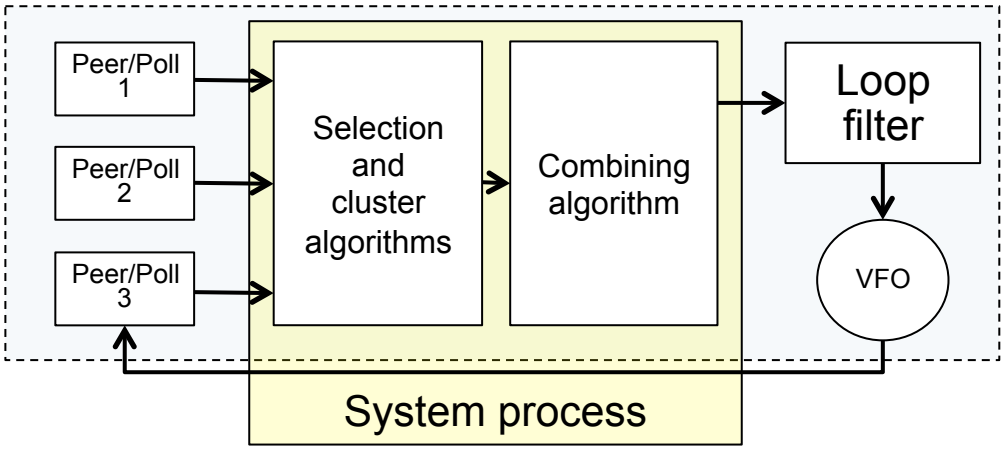
- **Peer and poll processes**

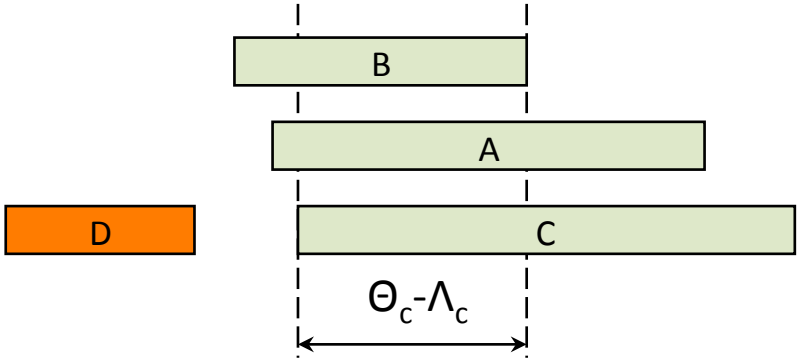


For each association:

- Poll each server at a rate of 2^{τ}, with $4 \leq \tau \leq 17$
- Poll interval is adjusted by the loop filter automatically
- The peer process listen for incoming packets
- Upon receipt the time offset θ and the delay δ is calculated for each association
- Peer process stores the last 8 measured pairs (θ_j, δ_j)
- The pair with the smallest δ_j is used to represent the best estimator for θ
- Computes statistics used by the system processes
  - Jitter φ (RMS of θ_j)
  - Dispersion ε

Selection and cluster algorithm

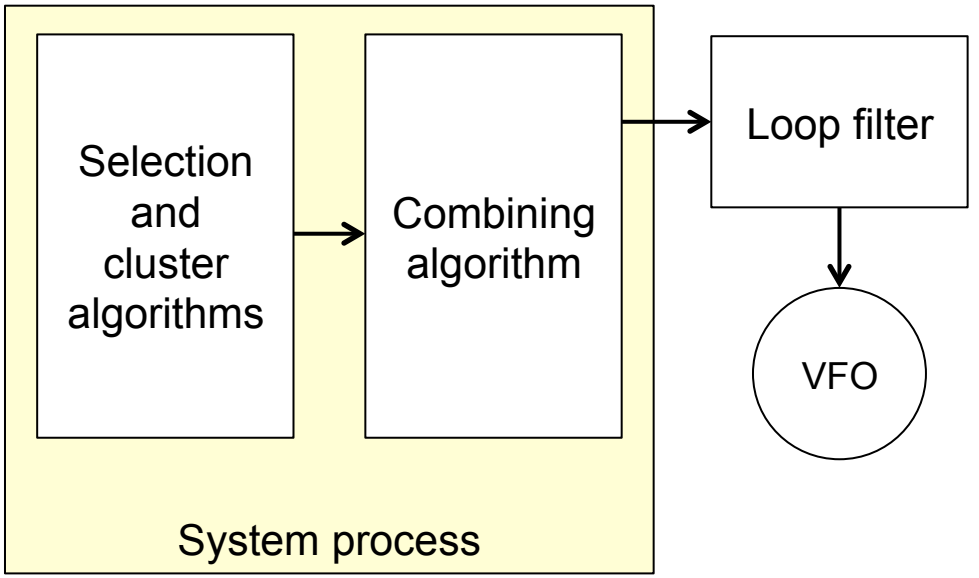




Objectives

- Detect “falsetickers” (D) and reject them
- Find a clique of “truechimers” (A, B, C)
- From the list of truechimers extract the n_{min} peers which provide the best accuracy
- A *preferred* peer always survives the cluster algorithm

- **Combining algorithm**



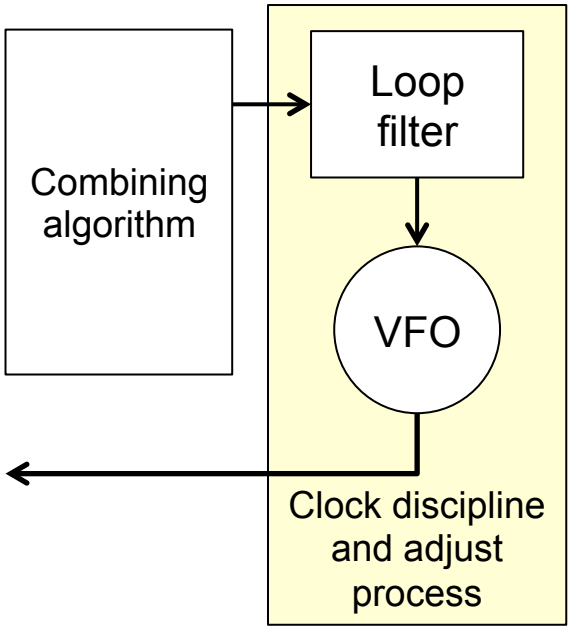
Objectives

- Combine time offset measurement from the survivors of the cluster algorithm

$$\Theta = a \sum_i \frac{\theta_i}{\Lambda_i}$$

- Pass this information to the clock discipline process
- Note: If a *preferred* peer is used
  - In this case the combining algorithm is not used
  - Instead the preferred peers offset is used to discipline the system clock

- **Clock discipline and adjustment process**



Objectives

- Pass information to the loop filter (acts as low pass filter)
- Adjust frequency of system time
- NTP applies a combined PLL / FLL algorithm to control the frequency of the system clock
- Calculate poll update interval for each association

Leap Second

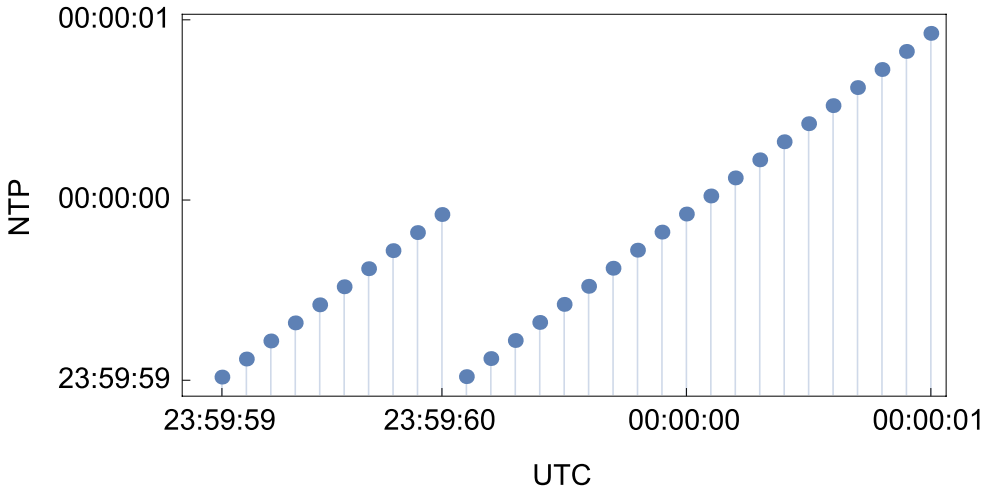
- NTP announces an impending leap second via the leap indicator bit in NTP's packet header
- It processes the leap indicator bit at the end of the last minute of the current day

```
0      | no warning
1      | last minute of the day has 61 seconds
2      | last minute of the day has 59 seconds
3      | unknown (clock unsynchronized)
```

- Time of announcement depends on reference time source
  - GPS: several month before leap event
  - DCF77: one hour before leap event
- Recommend: Usage of NTP leap second file from NIST
  - Consistent announcement of the leap second one day prior of its occurrence

The Mess with the Leap Second

- The leap second is processed by the operation system
- The operating steps back one second at the begin of the leap second (in most cases)
- “Smeared leap second” (– adding a second unnoticed!)
  - Approach introduced by Google (penultimate leap second event)
  - The server does not announce the leap second. Instead the equivalent of one second will is added in small increments during the last day before the leap second by the NTP server.
  - This year Meinberg provided an similar approach upon request



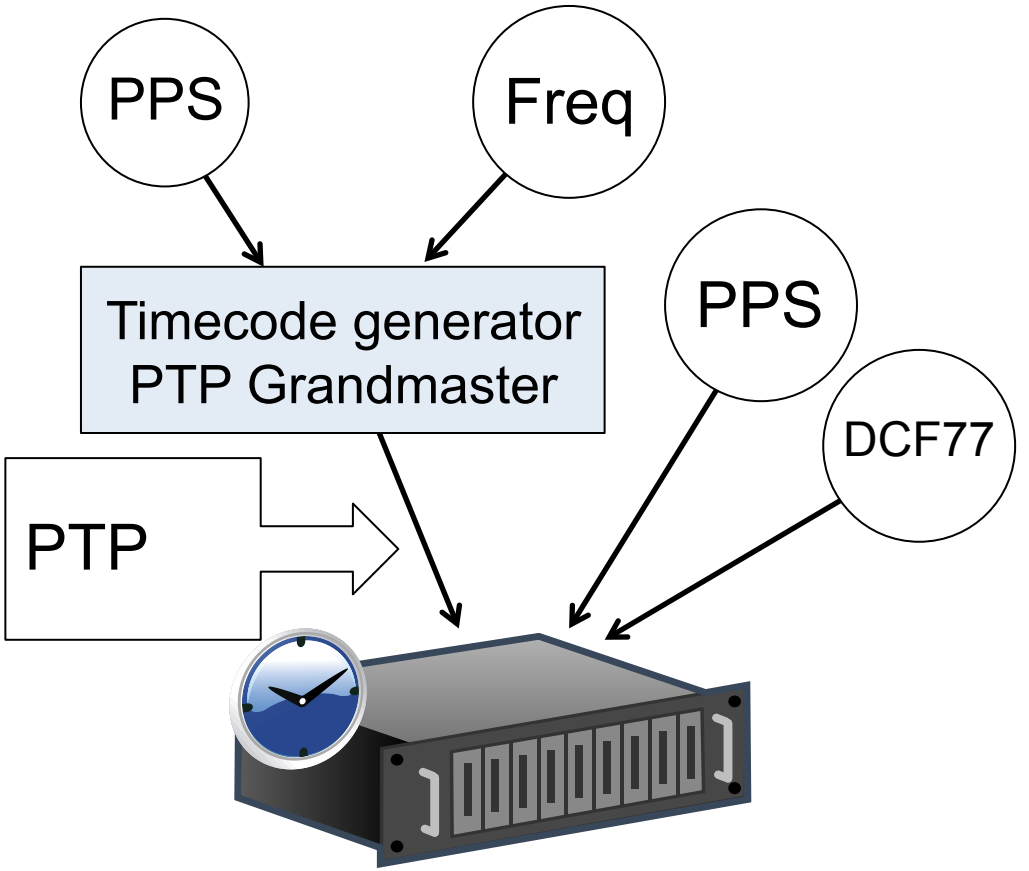
Kernel behavior during leap second event 2015-06-30

```
2015-06-30 23:59:59.228109151, leap 1,
2015-06-30 23:59:59.478348691, leap 1,
2015-06-30 23:59:59.728589861, leap 1,
2015-06-30 23:59:59.978826052, leap 1,
2015-06-30 23:59:59.229081948, leap 1,
2015-06-30 23:59:59.479319607, leap 1,
2015-06-30 23:59:59.729566864, leap 1,
2015-06-30 23:59:59.979825521, leap 1,
2015-07-01 00:00:00.230121614, leap 1,
2015-07-01 00:00:00.480385028, leap 0,
```

Reference Clocks

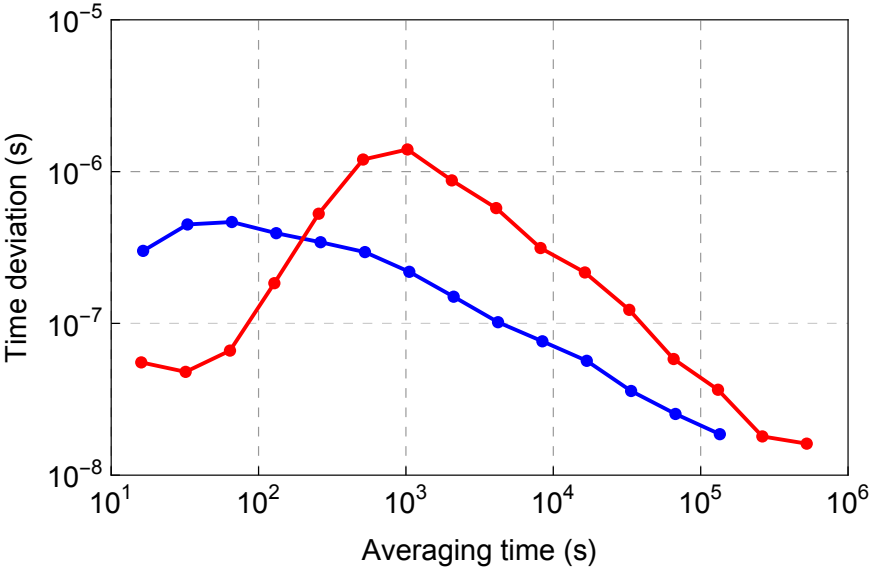
References used by PTB

- Time code generator
  - Synchronized via
    - GPS
    - DCF77
    - Frequency, PPS
- Time code
  - transmitted via
    - PTP
    - IRIG-B
  - Input via
    - RS232
    - PCI-card
- PPS
  - Input via RS232
- Legal requirements may restrict the choice of the time reference



Performance

- Best performance via PPS signal (RS232)
- However:
  - PCI-based time code generator synchronized to a PTP grandmaster via PTP protocol also shows sufficient performance



- TDEV for old (blue) and new (red) NTP servers
- Time references:
  - 1pps (blue)
  - PTP (red)
- Dataset
  - MJD 56658 to MJD 56689 (blue)
  - MJD 56901 to MJD 56931 (red)

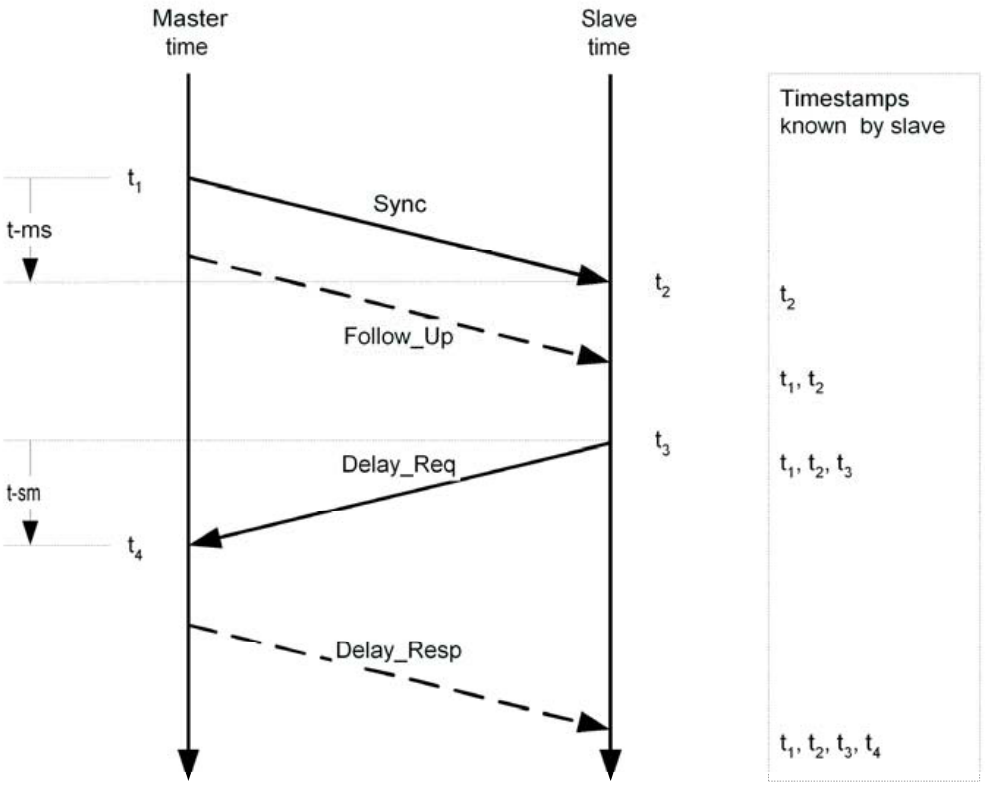
Comparison with PTP

	NTP	PTP
Model	Client-Server	Master-Slave
Network layer	Layer 3 (IP/UDP)	• Layer 2 • Layer 3 (IP/UDP)
Communication mode	• Typically unicast mode • Broadcast/Multicast possible	Multicast based
Networks	Usable for all kind of networks	• Designed for Local Area Networks (LAN) • Also applied in Wide Area Networks. Require control of the network
Hardware dependence	No	All hardware devices between master and slave must support the PTP in order to achieved the desired accuracy.
Clock types	NTP specification describes the server and client state machine	PTP knows the concept of • Grandmaster clock • Boundary clock • Transparent clock • Ordinary clock
Usage	Millions of NTP clients and server in the internet. E.g. • All operating systems • Switches • Routers	• Currently implemented more in the industrial area • Switches (telecom, industrial) • Sensors Because of required HW support more expensive

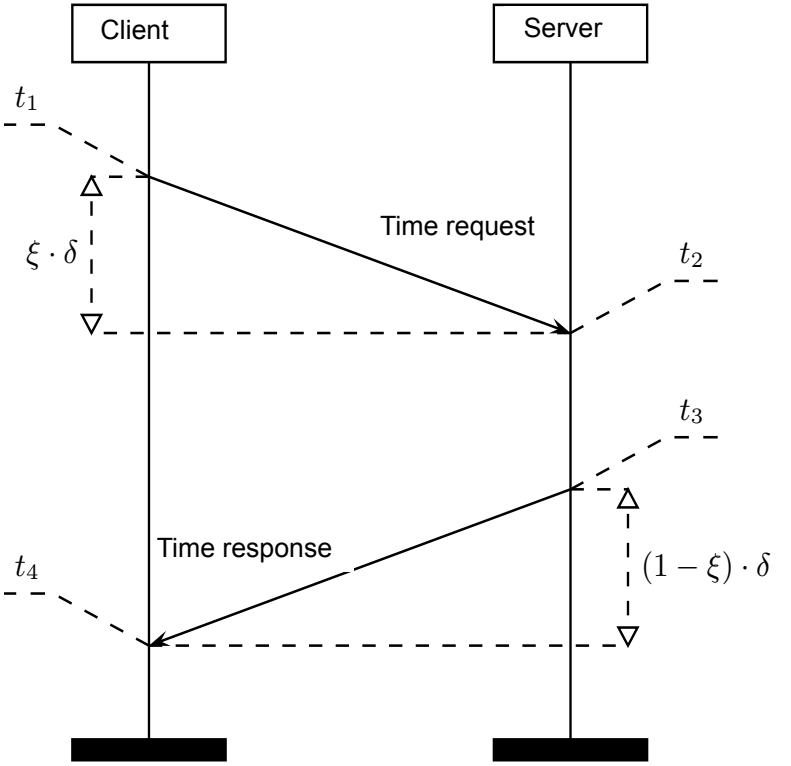
Comparison with PTP

Message exchange

PTP



NTP



Part 2

Operational Aspects

- Configuration
- Monitoring

Operational Aspects - Configuration

- NTP appliances:
  - Configurable via a web based GUI
- “Home made” NTP server
  - Configuration via text files
  - For most Linux distributions:
    - Main configuration file: **`/etc/ntp.conf`**
    - Authentication key file: **`/etc/ntp.keys`**
    - Start/stop script: **`/etc/rc.d/ntp`**

Operational Aspects - Configuration

- For stratum 1 server
  - Use “prefer” option if more than one reference clock (e. g. PTP and DCF77) is configured. E.g.

    `server` *`IP-address`* `minpoll 4 maxpoll 4 prefer`

- „peer“ statements
  - Mutual sync between more than one server (same stratum level)
  - Establish mutual time in the case all reference clocks are not available
  - Use authentication for peering

Operational Aspects - Configuration

- Access control
  - Use “restrict” statements in order to protect for
    - DoS attacks
    - To act as a DoS amplifier
    - To restrict retrievable information
  - Allow public only time requests
    - `restrict -4 default kod nomodify notrap nopeer noquery # IPv4`
    - `restrict -6 default kod nomodify notrap nopeer noquery # IPv6`
  - More access rights for chosen IP addresses

```
1. restrict -4 192.192.100.100 255.255.255.255
2. restrict -4 192.192.100.0 255.255.255.0 nomodify
```

    - Example 1) enables the host 192.192.100.100 to request information from the NTP server and to change the configuration of the NTP process
    - Example 2) enables hosts with an IP address from the class C net 192.192.100.0 to request information from the NTP server

Operational Aspects - Configuration

- Authentication
  - Don’t use autokey (it does not work and it has vulnerabilities)
  - Pre-shared symmetric authentication scheme
    - Requires a secure key exchange with the client
    - Trusted keys has to be configured; e.g.:
      - In `/etc/ntp.conf` add `trustedkey 1 2 15 ( 1001 ... 1008 )`
      - The keys 1, 2, 15 and the keys in the range from (1001 … 1008) are trusted
    - NTPD natively supports MD5 hash mechanisms (MD5 is deprecated)
    - NTPD supports SHA1 hashes if build against the OpenSSL library
- Use NIST’s leap second file

```
leapfile /etc/ntp.leapseconds
```

Operational Aspects - Monitoring

- Statistic records
  - Enable statistics
    - “loopstats” statistics of the selected time source
    - “peerstats” statistics of each configured time source.
  - If these files shall be used for stability analysis use
    - **`minpoll=maxpoll`** (in order to get equidistant time series data)
    - According to RFC 5905: 4 ≤ **`minpoll`** ≤ **`maxpoll`** ≤ 17 (16 sec – 36 h)

MJD	time past midnight	clock offset	frequency offset	RMS jitter	RMS frequency jitter	clock discipline loop time constant
57343	3.000	-0.000003875	9.748	0.000002442	0.001388	3
57343	11.000	0.000000864	9.749	0.000002833	0.001307	3

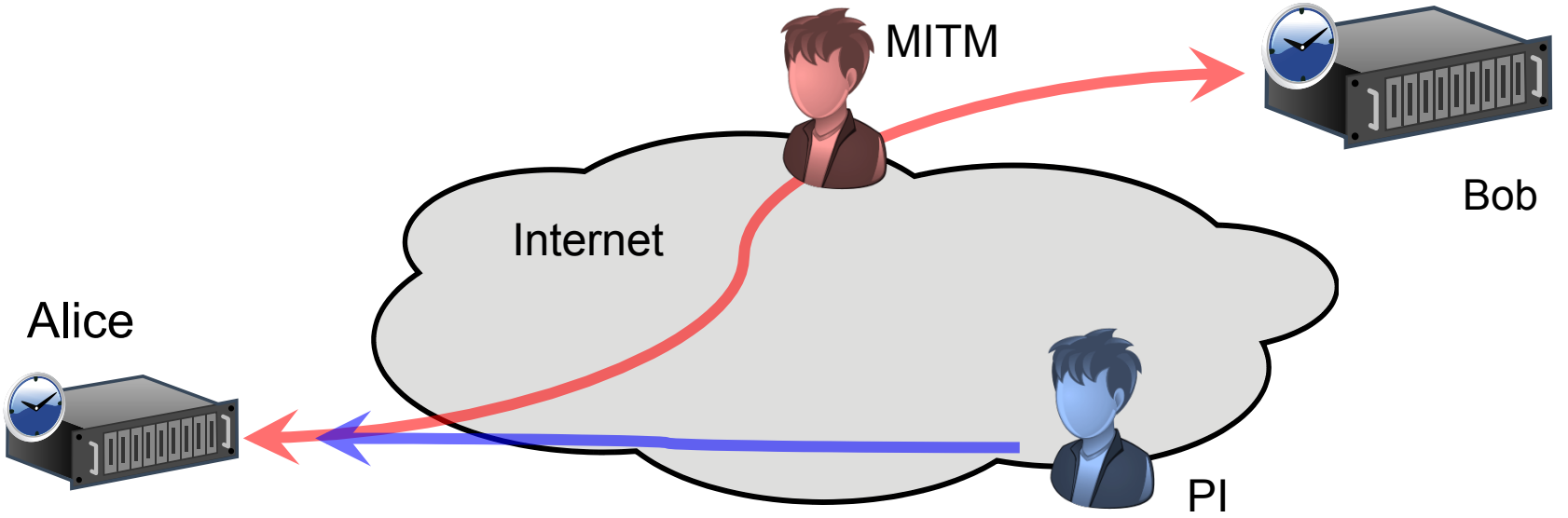
Example of loopstats file

PTB Security

- Threats
- Most prominent threats
- Mitigation measures

Security - Threats

- **IETF's TICTOC WG: RFC 7384 [Miz2014]**
  - Analysis of security threats for time synchronization protocols (NTP and PTP)
  - Defined a set of security requirements for NTP and PTP



Man-in-the-Middle (MITM)
Packet Injector (PI)

Security – Threats: RFC [Miz2014]

Threat	By what means	Who
False Time	Changing the timestamps in the NTP header	MITM
	To masquerade as a legitimate server and sending false time	MITM, PI
	Rogue master attack. Causing the client to believe the attacker is a legitimate time server	MITM, PI
	Delay attack. Impacts the asymmetric factor ξ and therefore the calculation of the time offset	MITM
	Master time source attack (e.g. GPS spoofing). Providing wrong time information to the time server	MITM, PI
Degradation of synchronization	Interception and removal of time synchronization packets.	MITM
Interruption of time service	Denial-of-Service (DoS) attacks including Crypto-DoS Crypto-DoS	MITM, PI

Security – Threats: Mitigation

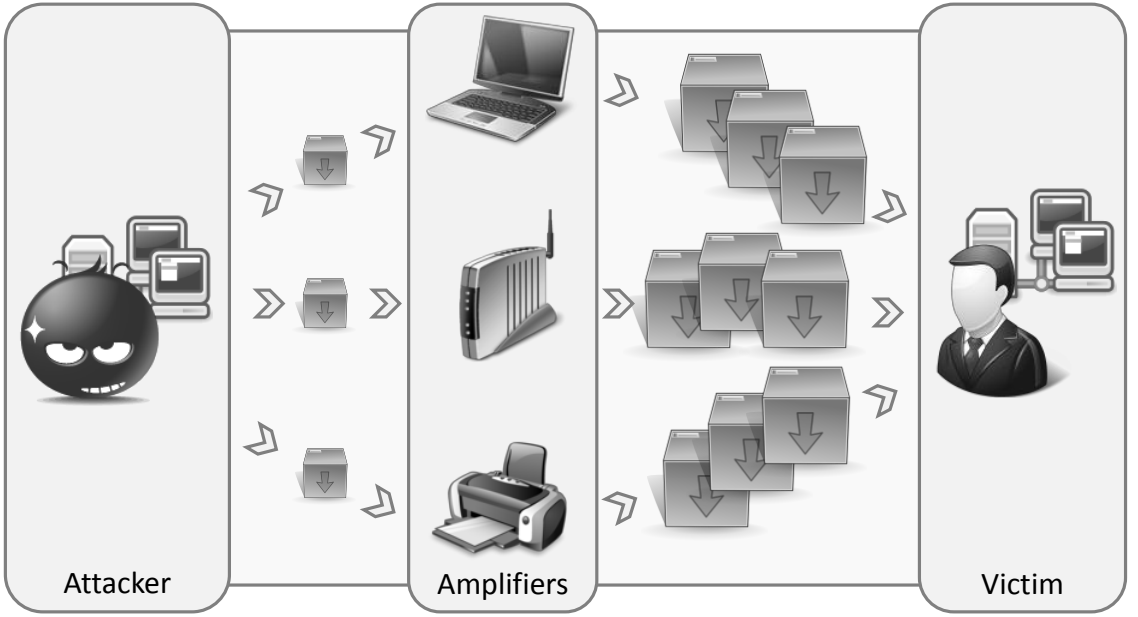
Threat	By what means	Crypto means for mitigation
False Time	Changing the timestamps in the NTP header	Integrity protection: signature or MAC
	To masquerade as a legitimate server and sending false time	authentication
	Rogue master attack. Causing the client to believe the attacker is a legitimate time server	authentication
	Delay attack. Impacts the asymmetric factor ξ and therefore the calculation of the time offset	
	Master time source attack (e.g. GPS spoofing). Providing wrong time information to the time server	
Degradation of synchronization	Interception and removal of time synchronization packets.	
Interruption of time service	Denial-of-Service (DoS) attacks including Crypto-DoS	• In general difficult to protect from • Authentication of clients

Security – NTP's Most Prominent Threats

- DoS: Distributed Denial-of-Service (DDoS) attack with amplification:
  - Czyz, J. *et al.* Taming the 800 Pound Gorilla. 435-448, doi: 10.1145/2663716.2663717 (2014)
  - Rossow, C. in *NDSS Symposium 2014* (Internet Society, San Diego, California, 2014).
- Vulnerabilities in the NTP reference code base
  - E.g.: end of 2014 (vulnerabilities in the crypto-code)
- "*Kiss-of-Death packet*" attack
  - Malhotra, A., Cohen, I. E., Brakke, E., & Goldberg, S. (2015, 2015-01-21). Attacking the Network Time Protocol. Retrieved from http://www.cs.bu.edu/~goldbe/papers/NTPattack.pdf

Security – Threats: DDoS

Distributed Denial-of-Service (DDoS) attack with NTP amplification



From [Ros2014]

- Distributed: Adversary utilizes the resources of many relay stations
- Amplification: The relay station's response is larger than the request
- Amplification factor: BAF
  - BAF=(UDP payload from attacker to amplifier)/(UDP payload from amplifier to victim)

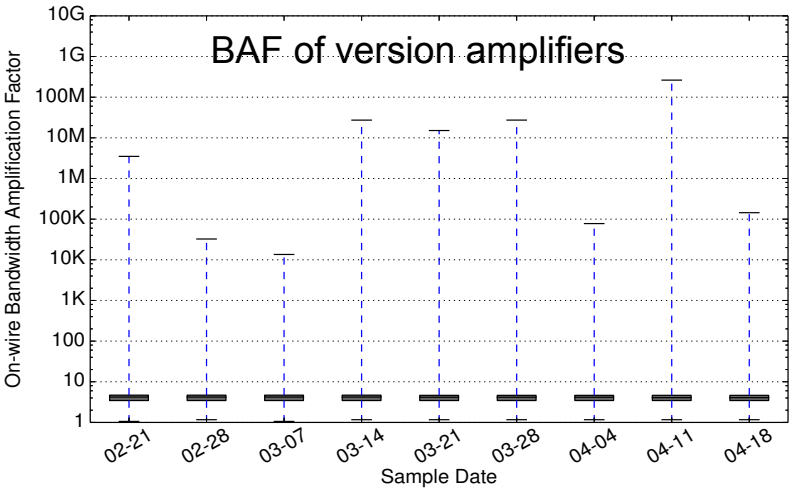
Security – Threats: Denial-of-Service

Precondition

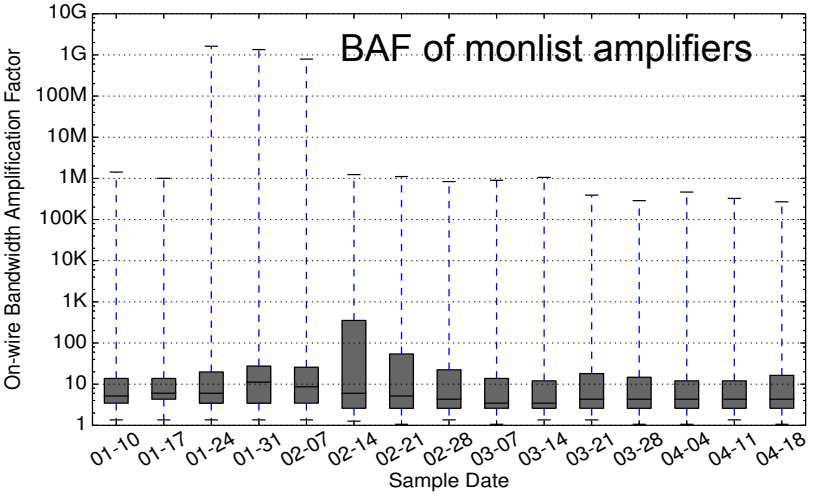
- Amplifier is a public accessible NTP server
- Amplifier runs an NTP daemon that responses to mode-7- or mode-6 packets (e.g. version < 4.2.7, until 1. Q. 2014)
- Attacker applies mode-7 or mode-6 requests (these are no time requests)

Mode	Command	Amplification factor
		Global internet [Czyz et al.] Median BAF
6	`ntpdc -c monlist`	4 – 10, spike in Febr. 50 - 500
7	`ntpdc -c rv`	5

From: [Czy2014]



2014



2014

Security – Mitigation Measures

- **NTP: RFC 1305:**
  - Symmetric pre-shared key approach
  - It works but it doesn‘t scale
  - Not usable for national wide time dissemination
- **NTP: RFC 5906: (Autokey):**
  - Integrity protection based on MAC and
  - authentication based on asymmetric cryptography
  - Several vulnerabilities (S. Röttger, 2011)
- **Ongoing work in the IETF: „Network Time Security“**
  - Security measures to cryptographically protect time synchronization packets
  - Focus on NTP
  - Ability for other time synchronization protocols (e.g. PTP)
- **PTP: Annex K**
  - Is not implemented

Security – Mitigation Measures

- Have a current version of NTPD running!
  - E.g. from yesterday:
    - Mehrere Schwachstellen in der Implementierung des NTP-Daemons vor Version 4.2.8p4 können von einem entfernten, nicht authentisierten Angreifer zur Ausführung beliebigen Programmcodes, der Manipulation von Dateien, der Durchführung von Denial-of-Service-Angriffen und der Umgehung von Sicherheitsvorkehrungen ausgenutzt werden, wodurch die Manipulation der bereitgestellten Zeitangabe erreicht werden kann. Die Schwachstellen sind in der NTP Version 4.2.8p4 behoben worden.
  - Some vulnerabilities in the implementation of NTPD prior to 4.2.8p4 (which is from Nov 2015). Can be used to run arbitary code. Fixed in 4.2.8p4.

Current Developments

- NTS
- BCP
- PTP

Current Developments - NTS

Network Time Security (NTS) provides:

- Authenticity of time servers
- Ability to authenticate time clients to the server
- Ability to perform authorization checks for time clients and servers
- Integrity of synchronization data packets
- Conformity with TICTOC's Security Requirements (RFC 7384)
- Support for NTP
- Ability for other time synchronization protocols, e. g. PTP

Current Developments - NTS

Design criterions

- Usage of symmetric cryptography for time message exchange packets in order to avoid latencies due to cryptographic operations
- Integrity and authenticity of time exchange packets protected by a Message Authentication Code (MAC) produced by a HMAC algorithm
- Unicast associations:
  - Key for MAC generation is unique between client and server
- Broadcast/Manycast associations
  - Usage of *Timed Efficient Stream Loss-tolerant Authentication* (TESLA, RFC 4082) approach for strong server authentication

Current Developments - NTS

Authentication

- Client authenticates its server once, initially (e.g. certificate-based authentication)

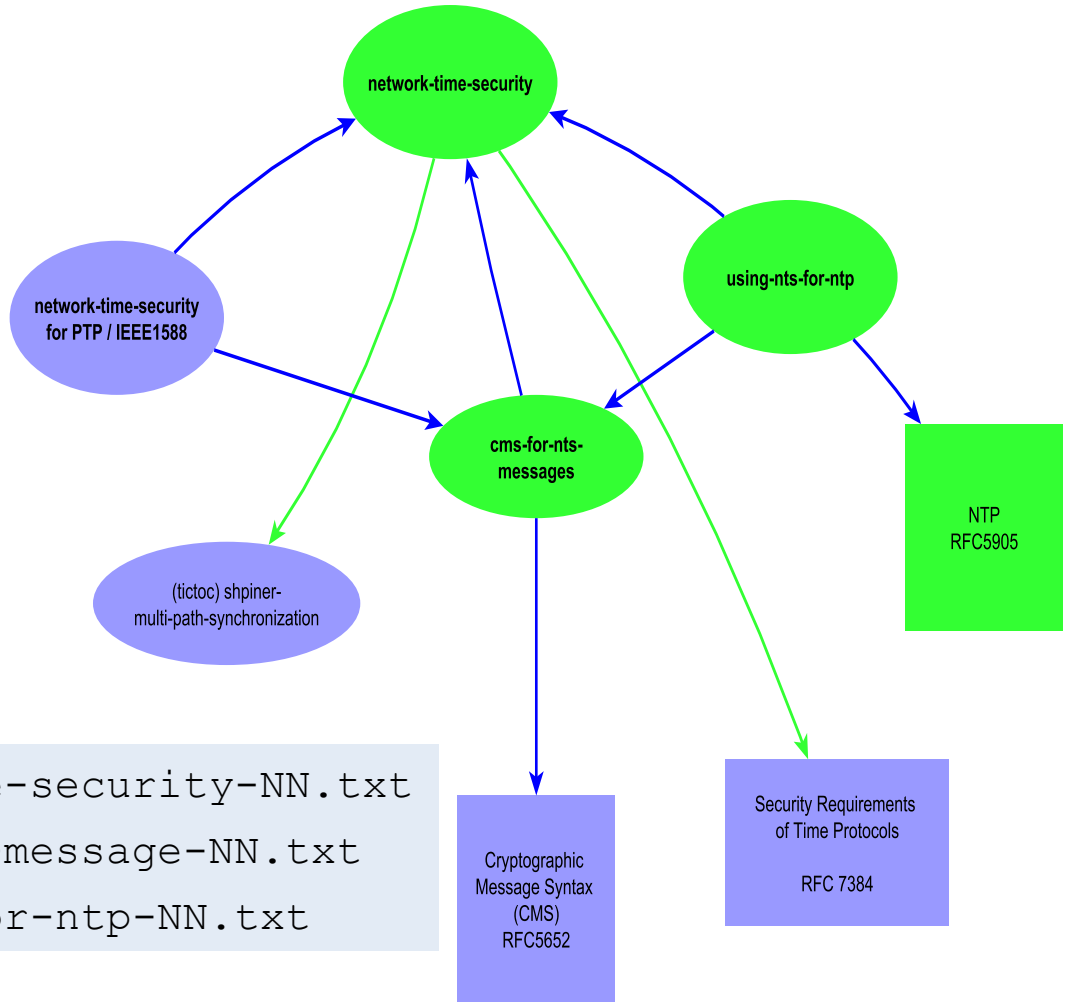
Integrity Protection via MAC

- Unicast mode (client-server): associations are supposed to be stateless on the server side
  - Shared secret („cookie“) for each client-server association.
    - Can be re-generated by the server at any time
    - Is transmitted once, initially, from server to client (transmission secured through asymmetric cryptography)
    - Is then used as key for MAC generation

Current Developments - NTS

- Broadcast mode: uses TESLA protocol (RFC 4082 of IETF)
  - Requires client and server to have at least roughly synchronized clocks initially (ensured via unicast mode)
  - Each packet‘s integrity secured by a MAC, generated with key from one-way-chain
  - Keys are used exactly once
  - Each one-way key is bound to a specific time interval
  - Disclosure follows pre-disclosed schedule
  - Verification of received packets is performed *a posteriori*

Current Developments – NTS Documents



Standardized within the IETF

- Three Internet-Drafts:

- `draft-ietf-ntp-network-time-security-NN.txt`
- `draft-ietf-ntp-cms-for-nts-message-NN.txt`
- `draft-ietf-ntp-using-nts-for-ntp-NN.txt`

http://datatracker.ietf.org/wg/ntp/documents/

Current Developments - BCP

Best Current Practice

- Is intended to be a RFC (Standard track)
- Motivated by the DDoS attack 2014
- Shall provide guidance regarding
  - Configuration of NTP servers
  - Operation of NTP servers
  - Guidelines for manufacturer of NTP appliances
- Draft version can be found under:

  `http://datatracker.ietf.org/doc/draft-reilly-ntp-bcp/`

Current Developments - PTP

IEEE P1588 Working Group

- Revision of IEEE 1588:2008 Standard [IEE2008]
- Subcommittees for:
  - High Accuracy
    - Goal: sub-ns synchronization
    - Based on White Rabbit project
  - Security
    - Specify a native security protocol that provides
      - authentication of the master clock
      - Integrity protection of the time synchronization packets
    - Provide mappings and guidelines to utilize secured tunnel protocols (IPsec, MACsec)

Bibliography

[Czy2014] Czyz, J. *et al.* Taming the 800 Pound Gorilla 435-448, doi: 10.1145/2663716.2663717 (2014).

[IEE2008] IEEE. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. In *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002).* 2008, p. c1-269.

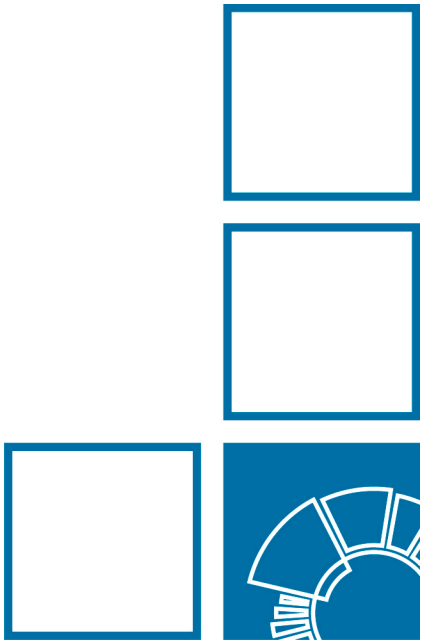
[Mil2006] MILLS, D. *Computer network time synchronization: the Network Time Protocol*. Edtion ed.: CRC Press, 2006. 304 p. ISBN 9780849358050.

[Miz2014] Mizrahi, T. (2014). Security Requirements of Time Protocols in Packet Switched Networks (RFC 7384). doi:10.17487/rfc7384

[Ros2014] Rossow, C. in *NDSS Symposium 2014* (Internet Society, San Diego, California, 2014).

[MBG2015] Malhotra, A., Cohen, I. E., Brakke, E., & Goldberg, S. (2015, 2015-01-21). Attacking the Network Time Protocol. Retrieved from http://www.cs.bu.edu/~goldbe/papers/NTPattack.pdf

[Mil2012] Mills, D. (2012, 2012-05-12). The NTP Era and Era Numbering. Retrieved from https://www.eecis.udel.edu/~mills/y2k.html



Physikalisch-Technische Bundesanstalt
Braunschweig and Berlin
Bundesallee 100
38116 Braunschweig

Max Mustermann
Telefon: 0531 592-####
E-Mail: max.mustermann@ptb.de
www.ptb.de

Stand: 10/13