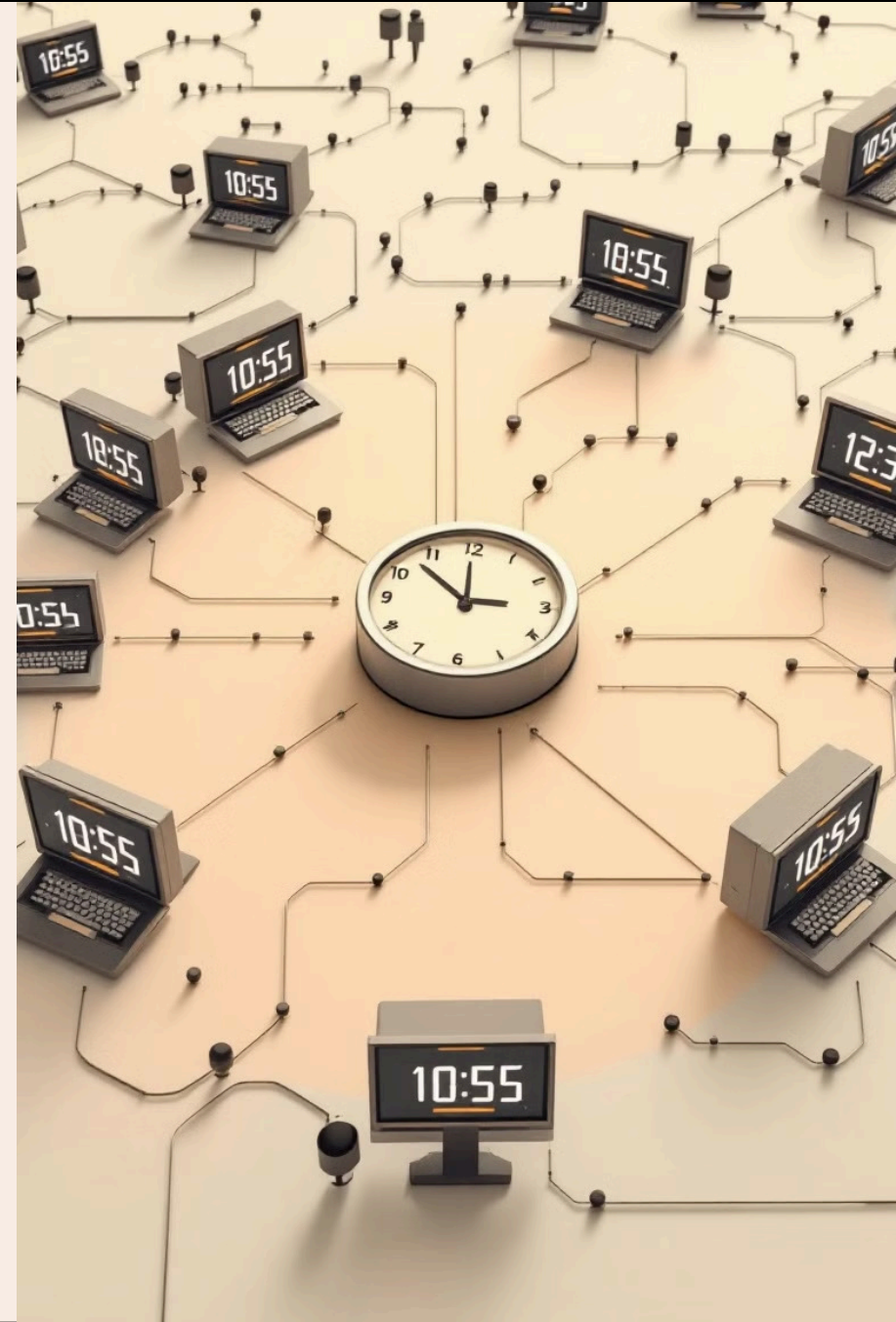


Excuse me, do you have the time ?

Please excuse all the AI images.

This is an "experiemental" set of slides.

This slide deck © 2025 by David Groves is licensed under Creative Commons Attribution-ShareAlike 4.0 International



Why Time Sync Matters

Log Accuracy

Including causality across distributed systems.

Broadcast Systems

SMPTE / ST2059 built on PTP.

Stock Market / HFT

Banking and stock trading systems require precise timing.

Databases: Optimistic Concurrency

Can avoid the need for locking and vastly improve performance.



Goals.

Time synchronization in networks has two distinct goals:

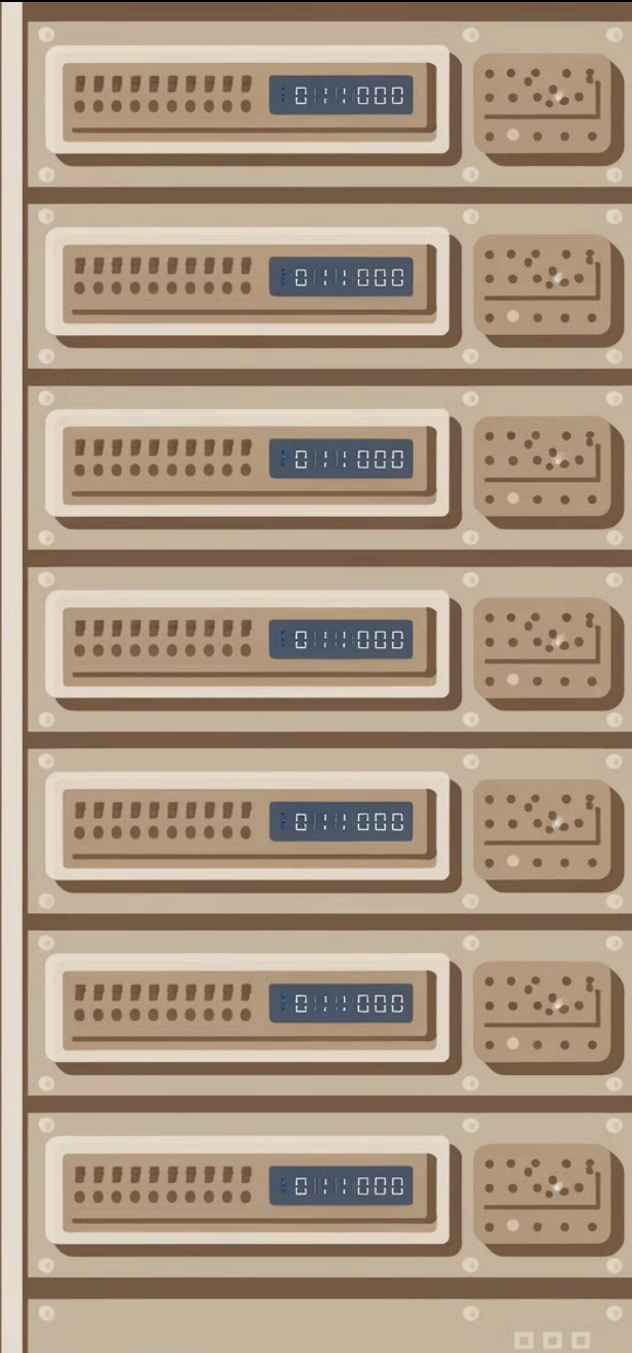
1. Device-to-Device Synchronization

Ensures all devices in a network have the **same time relative to each other**, regardless of whether that time is accurate to the real world.

Ideally you want both!

2. Wall Clock Accuracy

Ensures devices maintain time that is **accurate to the real world standard** (UTC/GMT).



Stepping vs. Slewing the Clock

Stepping the Clock

Instantaneously changes the clock time.

Issues with Stepping:

- Stepping backwards can break causality
 - Effects can come before cause
 - Many applications assume time moves forward monotonically
- Stepping forwards can break durations
 - A customer may get billed for more usage than they actually had
 - You may jump past a timed event and not trigger it

Slewing the Clock

Gradually adjusts the clock time.

Benefits of Slewing:

- Preserves causality of events
- Maintains monotonic time progression

Generally preferred for ongoing time synchronization, but may take too long if clock offset is too high.

Mario Odyssey: Time Manipulation Trick

The Setup

Speedrunners set the Switch clock to 8 minutes before daylight savings time shift occurs.

The Execution

Players spend 8 minutes playing the game, until they get to a part where they throw a seed into a plant pot that normally takes 20 minutes to grow.

The Zone Change

They run around a corner into a building (zone change), then return to find time has jumped forward due to daylight savings.

The Reward

The plant has already sprouted and produces a moon, saving 12 minutes of waiting time!



NTP.

NTP (Network Time Protocol).

Aims for microsecond level precision.

Developed by David Mills in 1985.

Currently at NTPv4, with some NTPv3 legacy.



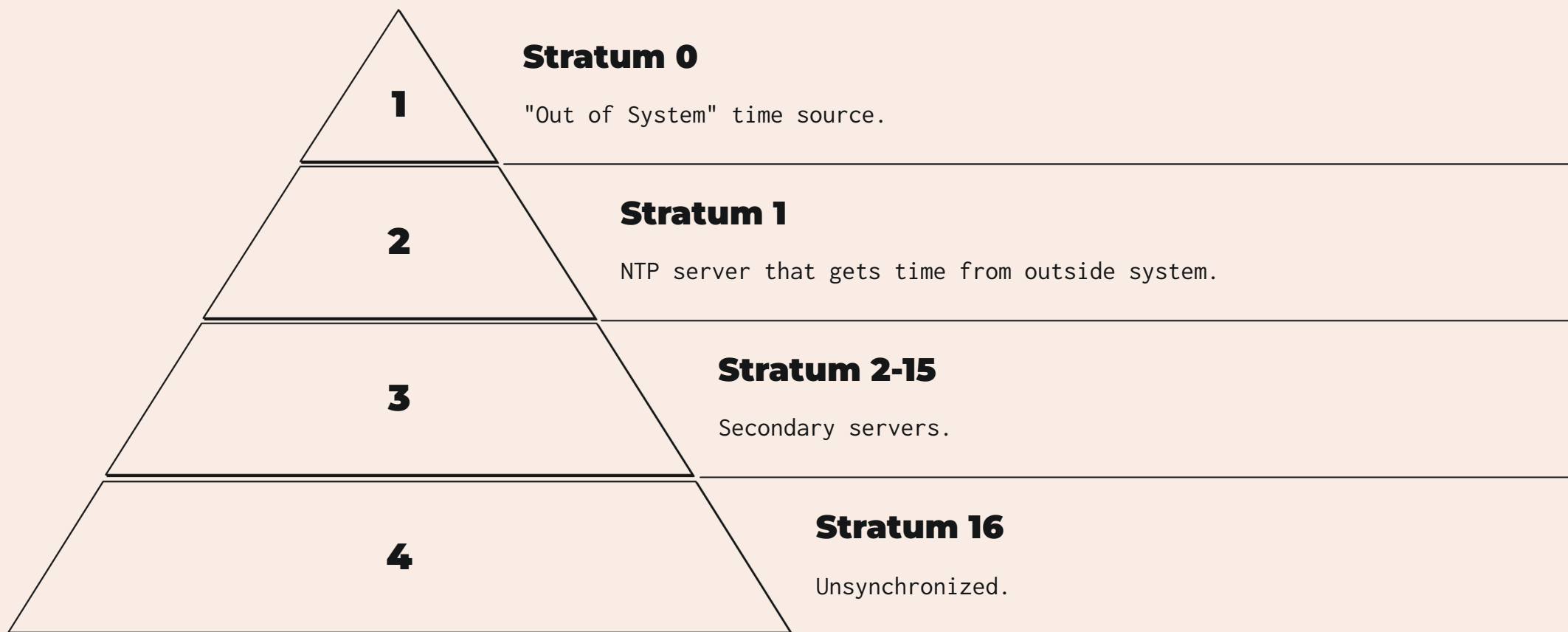
NTP (v4) Packet Structure

Offset	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	LI		VN		Mode		Stratum								Poll								Precision									
4	32	Root Delay																															
8	64	Root Dispersion																															
12	96	Reference ID																															
16	128	Reference Timestamp (64-bits)																															
20	160																																
24	192	Origin Timestamp (64-bits)																															
28	224																																
32	256	Receive Timestamp (64-bits)																															
36	288																																
40	320	Transmit Timestamp (64-bits)																															
44	352																																

NTP Packet Fields

- **2bits: Leap Indicator (LI):** Warns of impending leap second to be inserted/deleted in the last minute of the month.
- **3bits: Version Number (VN):** NTP version number (e.g., 3 for NTPv3, 4 for NTPv4).
- **3bits: Mode:** Indicates the mode (e.g., query/response/peer/broadcast/multicast).
- **8bits: Stratum:** Indicates the clock stratum level.
- **8bits: Poll Interval:** Maximum interval between successive NTP messages.
- **8bits: Precision:** Precision of the local clock.
- **32bits: Root Delay:** Total round-trip delay to the primary reference source.
- **32bits: Root Dispersion:** Maximum error relative to the primary reference source.
- **32bits: Reference ID*:** Identifier of the reference source (complicated !)
- **64bits: Reference Timestamp:** Time at which the system clock was last set or corrected.
- **64bits: Origin Timestamp:** Time at which the client sent the request.
- **64bits: Receive Timestamp:** Time at which the server received the request.
- **64bits: Transmit Timestamp:** Time at which the server sent the response.

NTP Architecture: Stratum Levels



Reference ID's.

Reference IDs have different meanings depending on the Stratum level:

0 **Stratum 0: "Kiss of Death" Packets**

- Kiss of Death.
- The Reference ID contains a 4-letter ASCII code indicating an error, examples :-
 - RATE
 - DENY
 - INIT

1 **Stratum 1: Time Source Identifiers**

When Stratum is set to 1, the Reference ID contains 0-padded ASCII characters identifying the Stratum 0 source:

- GPS: Global Positioning System time
- MSF: UK radio time signal
- WWVB: US NIST radio time signal

2 **Stratum 2-15: Network Identifiers**

- May be an IPv4 address (which fits in 32 bits)
- For IPv6 addresses (which don't fit), common practice is to take an MD5 hash and truncate to the first 4 bytes

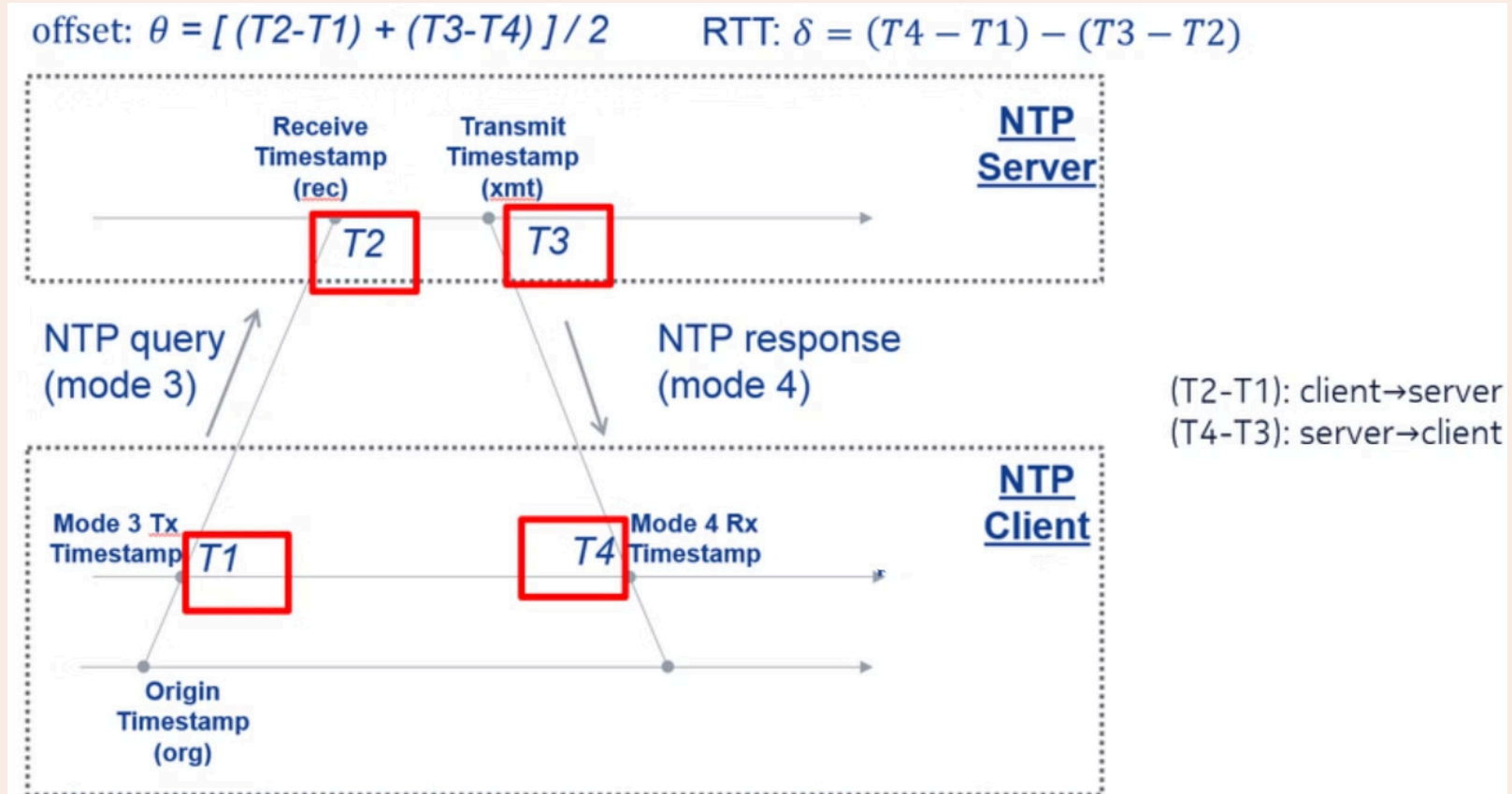
NTP Timestamp Format.

- NTP uses a 64-bit timestamp format to represent the time.
- The timestamp is split into two 32-bit parts:
 - The first 32 bits represent the number of seconds since the NTP epoch (January 1, 1900, at 00:00:00 UTC).
 - The second 32 bits represent the fraction of a second, providing a resolution of about 200 picoseconds.
- We have an NTP rollover problem 2 years ahead of the UNIX rollover problem !
- NTP uses system date to infer what epoch we are in.

Example: NTP Representation of 11 March 2025 @ 12:34:56.12345

Calculation	Value
Seconds since NTP epoch (Jan 1, 1900)	3,954,264,896
Complete NTP timestamp (binary)	11101011100111010101111110000000 (seconds) 00011111100110101101110100110111 (fraction of second)

NTP Timestamp Exchange. Regular Mode.



NTP Algorithms

Clock Select

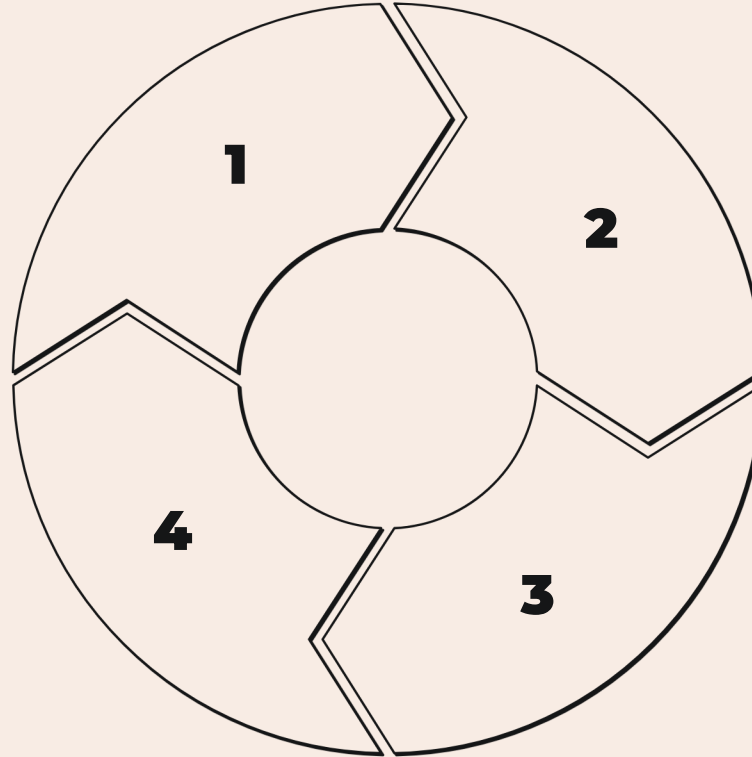
Chooses reliable servers.

Based on latency / jitter /
stratum / precision.

Clock Discipline

Nudge local time.

Adjust oscillator correction.



Filter Outliers.

Hard filter outliers.

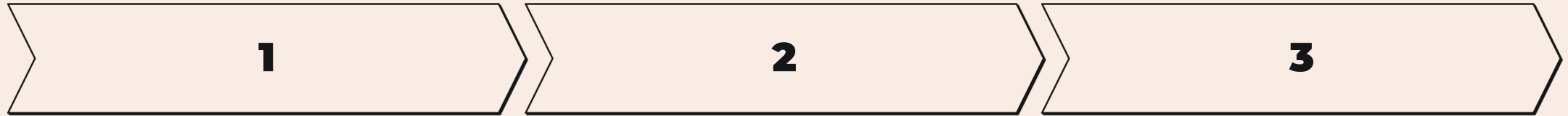
Sample Weighting

Weights sources closer to mean
higher.

NTP TLV Extensions

- **NTP TLV's** (Type-Length-Value) add extensions to the protocol.
- TLV's are added to the end of the standard packet.
- TLV's can send signalling commands like "increase frequency of NTP updates".
- Another important use of TLV extensions is **signing NTP updates**. By adding a TLV field containing a digital signature of the NTP packet, systems can authenticate the update and ensure that it has not been tampered with.
 - **NTP** supports pre-shared-secret signing using SHA-1 or SHA-256.
 - **NTPv4** introduced Autokey. Don't use it, it is broken in multiple ways.
 - **NTS** (Network Time Security) uses an out of band connection secured by TLS to exchange a "session" key to use with NTP.

NTP Sources of Error / Latency



Client

- Userspace program makes packet with timestamp in it.
- Kernel needs to process it.
- Put in NIC TX buffer.
- Be played out NIC.

Network

- Packet spends time in network devices RX and TX buffers.
- Packet spends time in cables.
- Without QoS, these residence times can be variable.

Server

- Packet enters RX buffer on server.
- Kernel needs to give packet to userspace program.
- Kernel needs to schedule userspace program.

- And the same back the other direction.
- This limits NTP's practical accuracy, typically to milliseconds in LAN environments and tens of milliseconds across the internet.

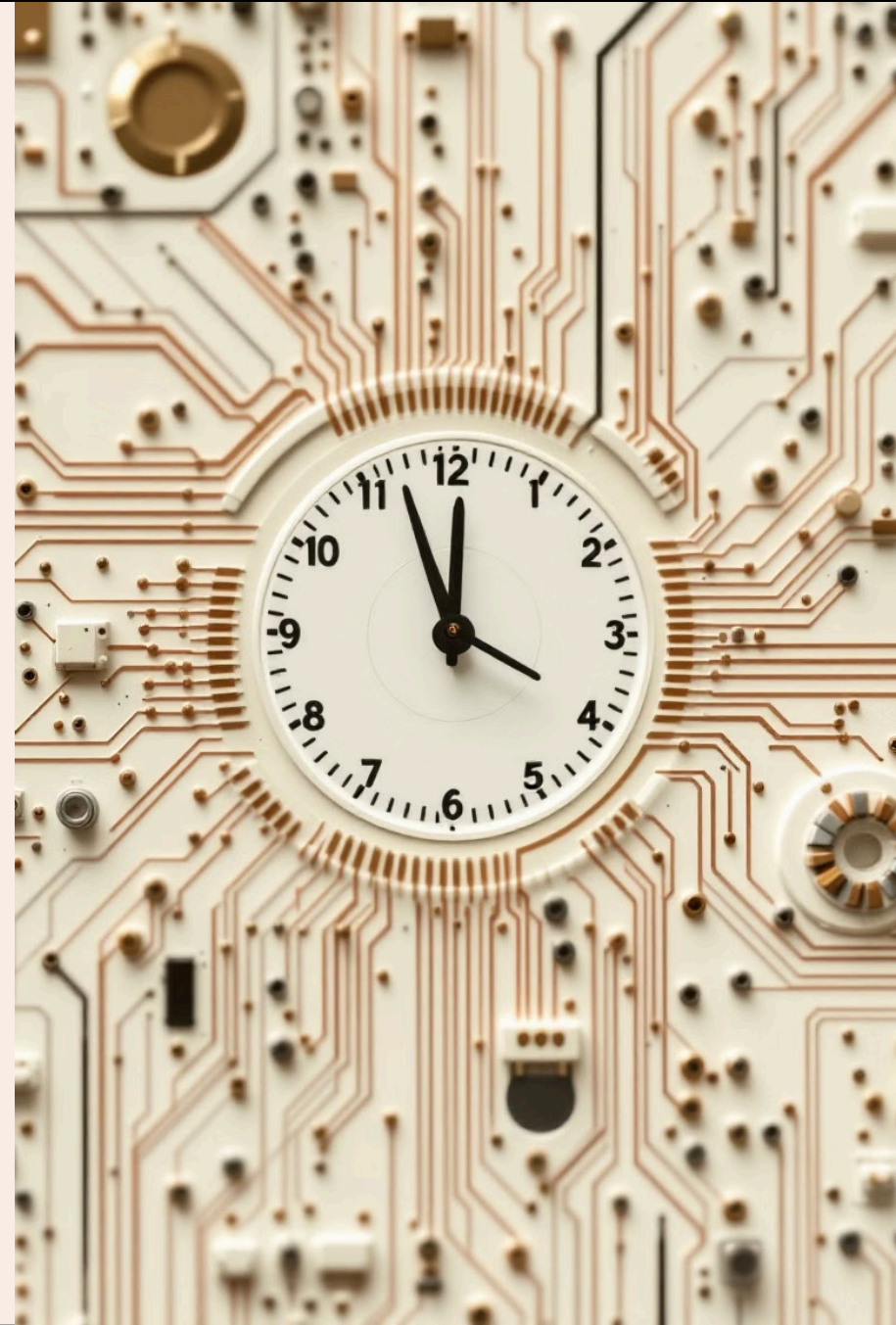
Precision Time Protocol (PTP).

An introduction to PTP, a protocol providing precise time synchronization in local area networks.

IEEE 1588 - 2002: PTPv1 (legacy).

IEEE 1588 - 2008: PTPv2 (common).

IEEE 1588 - 2019: PTPv2.1 (new, backwards compatible).



NTP vs. PTP: Network Participation

NTP (Network Time Protocol)

Passive Network: NTP operates with a passive network. Only the client and server directly participate in the time synchronization process. Intermediate network devices like routers and switches do not actively contribute or adjust timestamps, they are "mere transport".

PTP (Precision Time Protocol)

Active Network: PTP involves active participation from network elements. Routers and switches along the path can adjust timestamps.

This reduces errors in timesync due to the network transport.

PTP vs. NTP: Hardware Requirements

PTP-Specific Hardware Support



Hardware Timestamping

PTP requires specialized Network Interfaces with hardware timestamping capabilities.



Network Infrastructure

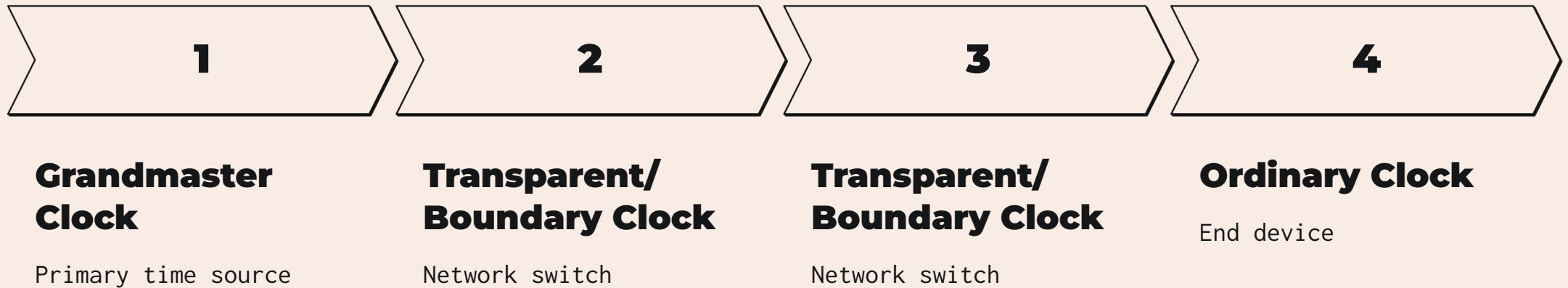
PTP needs PTP-aware switches and routers (boundary clocks or transparent clocks) that actively participate in the synchronization process.

- PTP capable computer NIC's used to be rare and expensive.
 - But now a Raspberry PI 5 is capable.
- Basic Switches/Routers usually still don't have support.

PTP Clock Modes

- **Grandmaster Clock:** The primary time source in a PTP network. It provides the reference time to which all other clocks synchronize.
- **Ordinary Clock:** Either a Master (timeTransmitter / leader) or Slave (timeReceiver / follower).
- **Transparent Clock:** A clock that measures the delay of PTP packets as they pass through it and compensates for this delay in the PTP timestamps. This improves the accuracy of time synchronization.
- **Boundary Clock:** A clock that acts as both a Slave on one side and a Master clock on the other side.

PTP's Example Flow.



PTP's Grandmaster Clock

- The Grandmaster Clock is the primary time source in a PTP network.
- It is elected using the BMCA (Best Master Clock Algorithm).

The BMCA uses several factors to determine the best clock, in this order.

- **Priority1**: User-configurable priority (0-255, lower is better).
- **ClockClass**: Quality of the clock (e.g., 6 for GPS-locked).
- **ClockAccuracy**: Precision level (e.g., 0x21 for within 100 ns).
- **OffsetScaledLogVariance**: Stability of the clock.
- **Priority2**: Secondary user-configurable priority (0-255, lower is better).
- **ClockIdentity**: Tiebreaker (usually based on MAC address or IP address).

PTP Transport Mechanisms: UDP vs. Ethernet

PTP over UDP.

- UDP for message transport.
- Port 319 (timekeeping events).
- Port 320 (non-time critical messages).
- Most messages are Multicast 224.0.1.129 / FF0x::181.
- Peer to Peer messages are Multicast 224.0.0.107 / FF02:6B

(More addresses in multi-domain environments).

PTP over Ethernet.

- Raw Ethernet frames.
- EtherType (0x88F7).
- 01-1B-19-00-00-00 for most messages.
- 01-80-C2-00-00-0E peer to peer messages.

Types of PTP messages.

ANNOUNCE (multicast): Announces a master clock and starts an BMCA election.

SYNC (multicast): The main message type. Contains the timing information.

FOLLOW_UP (multicast): Only used if clock doesn't have hardware timestamping.

DELAY_REQ (unicast): Used in End to End mode. Measures delay from Master to Slave.

DELAY_RESP (unicast): Responses to **Delay_Req**.

PDELAY_REQ (unicast): Used in Peer to Peer mode. Measures delay between devices.

PDELAY_RESP (unicast): Responses to **PDelay_Req**.

MANAGEMENT / SIGNALLING (mixed): Control traffic.

IEEE 1588 - 2008 Timestamp Format.

- Note: IEEE 1588 - 2002 uses an incompatible format.
- Other proprietary formats / quirks exist (Intel 64bit TOD for example).

<u>Bits</u>	<u>Value</u>
0-15 (16 bits)	15.26 femtoseconds (1/65536 of a nanosecond).
16-47 (32 bits)	Nanoseconds.
48-95 (48 bits)	Seconds since 00:00:00 1st Jan 1970 TAI.

- Due to the 48bit counter, PTP won't overflow for for about 9 million years.

PTP: End-to-End vs. Peer-to-Peer

End-to-End Delay

- In End-to-End Delay mode, the master clock sends sync messages directly to the slave clocks.
- When the messages pass through a transparent clock, the devices.
 - Logs when the packet/frame entered the device.
 - Knows when the packet/frame exits the device.
 - Either
 - Modifies the correctionField.
 - Sends a **FOLLOW_UP** correction packet/frame.
- Time spent in the network device is called "Residence Time".

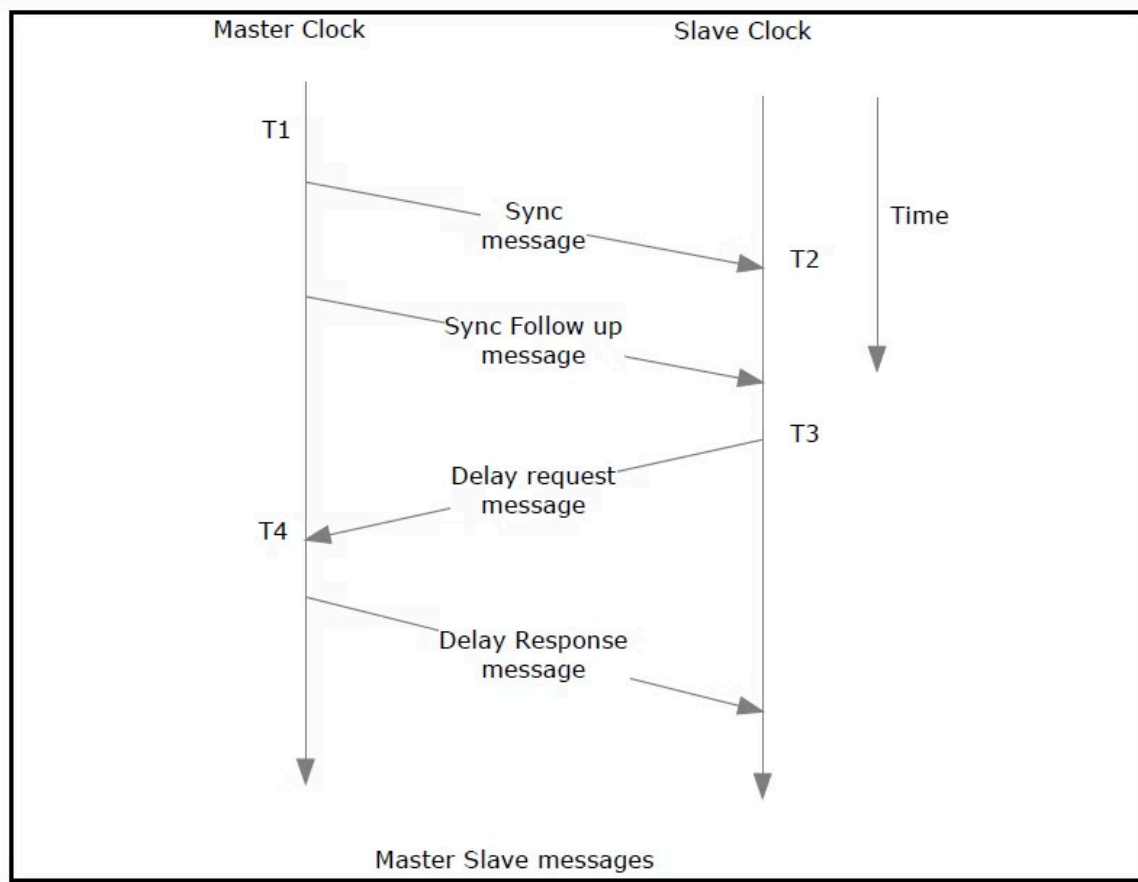
Peer-to-Peer Delay

In Peer-to-Peer Delay mode, each network node also measures link latency.

- Neighbours exchange pDelay messages.
- This lets them measure transmission delay.
- Only works with Point to Point Ethernet.
- When a device receives a packet/frame from a port, it adds the pDelay value to the packet.

1000 nanoseconds ~ 200m of fibre.

End to End Mode.



- If One Step, no Follow Up.
- If Two Step, then T1 time is in FollowUp.
- Calculation is then like it was in NTP.

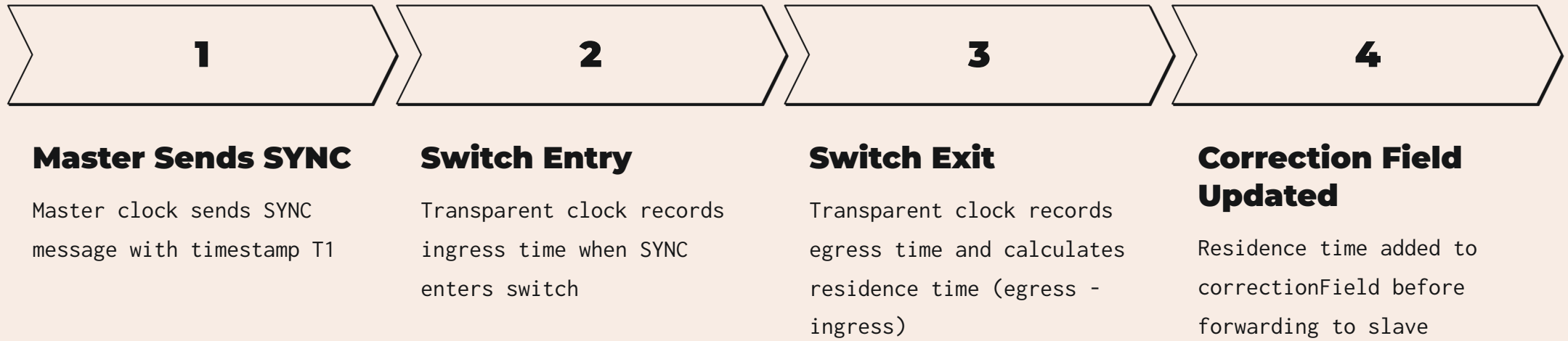
$$\text{Offset} = ((T2 - T1) - (T4 - T3)) / 2$$

This may pass through multiple switches in end to end mode.

This might not look different from NTP !

Peer to Peer Mode.

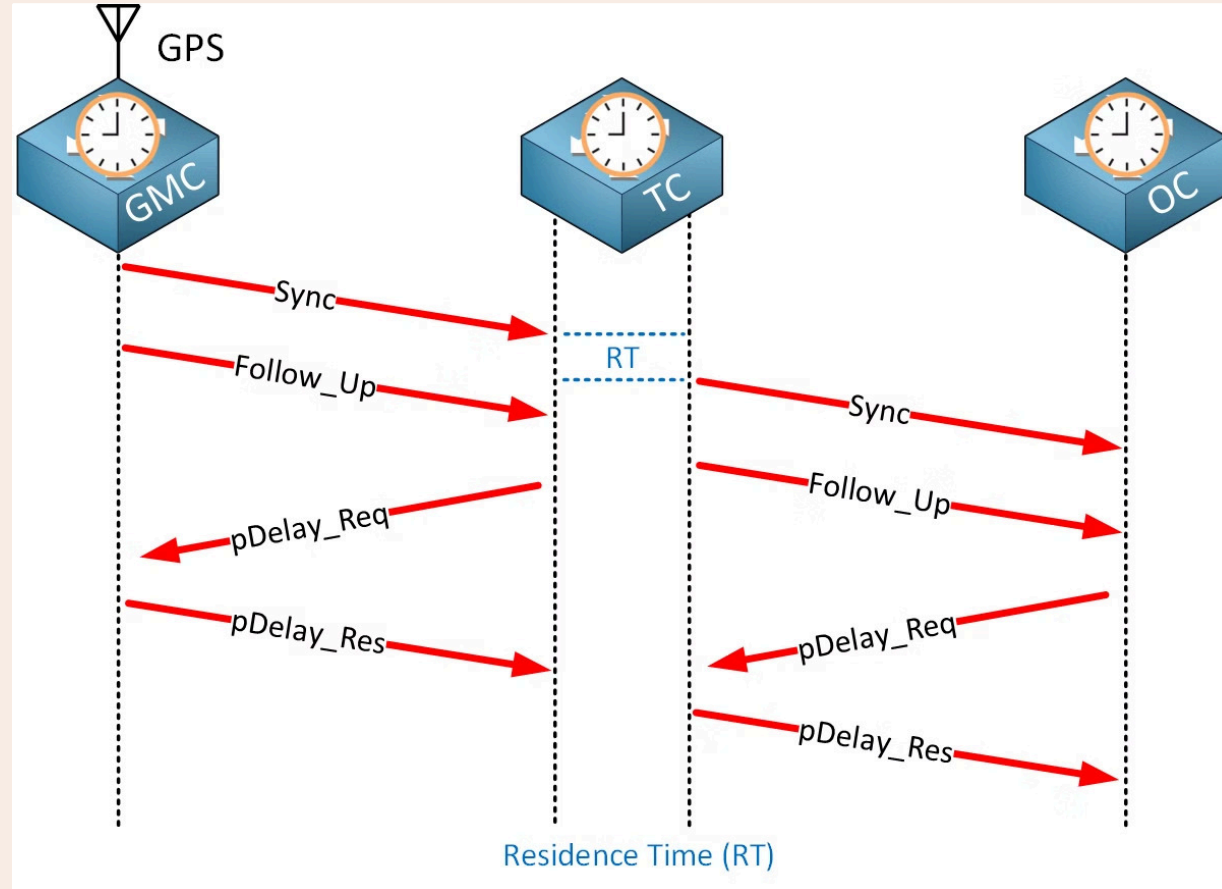
In Peer-to-Peer mode, transparent clocks add residence time to SYNC messages as they pass through:



The transparent clock measures exactly how long the SYNC message spends inside the switch (residence time). This value is then added to the correctionField of the PTP packet, allowing the slave to compensate for this delay and achieve more accurate synchronization.

In one-step clocks, the correctionField is updated directly in the SYNC message. In two-step clocks, the residence time is added to the FOLLOW_UP message's correctionField.

Peer to Peer Mode.



PTP Hardware Requirements

PTP-Aware Server Equipment

Servers require NICs with **hardware timestamping** capabilities to achieve microsecond or nanosecond precision.

Essential components include a **high-precision oscillator on the NIC**. Getting time from the CPU has too high latency/jitter.

NIC PHY chips with timestamp registers that capture packet arrival/departure times at the hardware level for two-step-sync.

NIC PHY chips that can modify packet/frame content for one-step-sync.

PTP-Aware Switch Equipment

A centralised **high precision oscillator** for keeping overall switch time.

A system to sync that oscillator to the **per-port / per-linecard oscillators**.

ASIC's that can **calculate and update residence time** for transparent mode.

ASIC's that can independently **collect peerDelay** for peer to peer mode.

Delay Mitigation: NTP vs. PTP Models

Different time synchronization protocols handle network delays through distinct mechanisms, affecting their accuracy and use cases.

NTP

Mitigates **statistical delays** through:

- Round-trip time calculations.
- Statistical filtering algorithms.
- Averaging multiple samples.

Cannot compensate for asymmetric network paths or variable queuing delays with high precision.

PTP End-to-End

Addresses **residence time delays** through:

- Transparent clock residence time measurements.
- CorrectionField accumulation across network path.

PTP Peer-to-Peer

Also handles **link-specific delays** through:

- Individual link delay measurements
- Neighbor-to-neighbor calibration
- Per-segment delay compensation

More resilient to topology changes and reconfiguration events.

Buy a PTP Grandmaster.



- Microchip TP4100 (top) = \$13,000. Common in Mobile Networks.
- Meridian II = \$4,000. Dual use NTP / PTP master.

Build your own PTP Grandmaster.

1 Use appropriate server.

- Hardware Timestamping NIC.
 - (or PCIe slot to contain one).
- PCIe slot for OCP timecard.

3 Sync all clocks.

Use `phc2sys` to sync.

- Server clock to Timecard.
- NIC clock to Timecard.

(ideally set Timecard + NIC to same PCI-E lane).

2 Antenna

Attach the OCP timecard via a cable to an antenna with good GPS reception.

This likely needs to be on the roof of your building.

4 Configure the PTP daemon

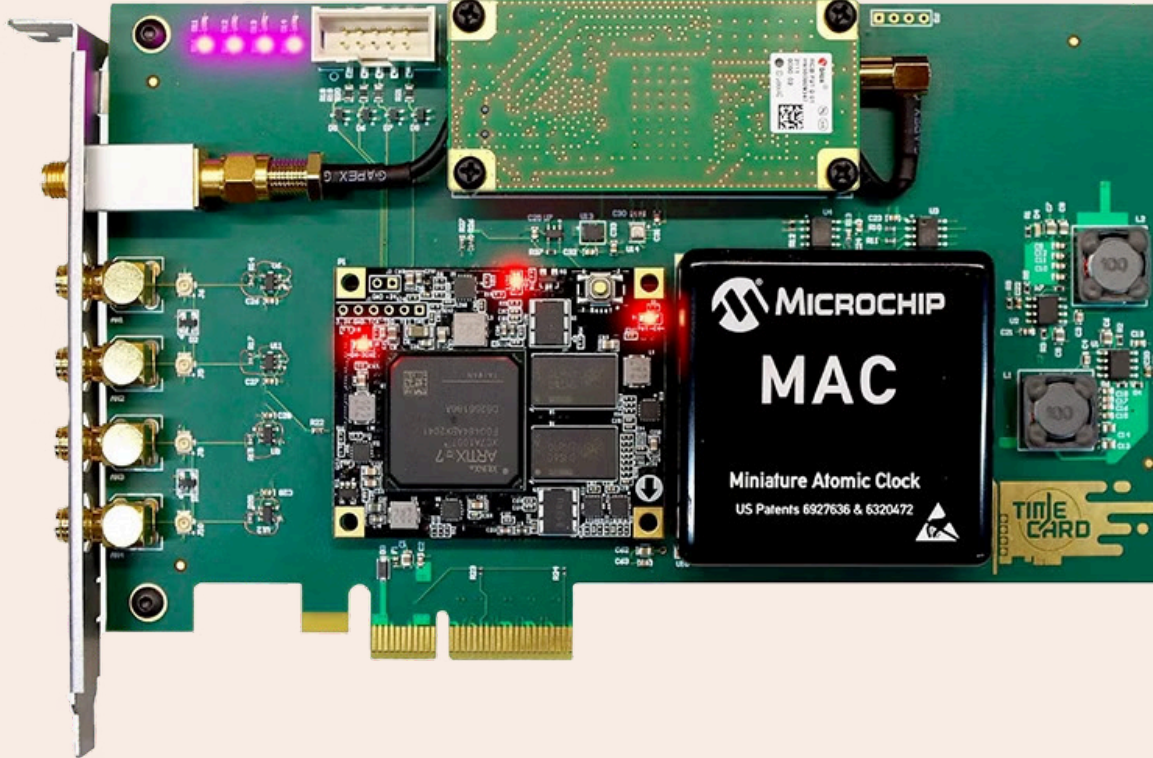
Set up `ptp4l`.

Configure to use hardware timestamping mode.

Configure to use one-pass if NIC supports it.

Otherwise configure `FollowUps`.

OCP Timecard from Timebeat.



- GNSS port (top left).
- 4 x External PPS ports.
- Rubidium Atomic clock on right.
- Measured holdover of MAC around ± 30 nanoseconds per day.

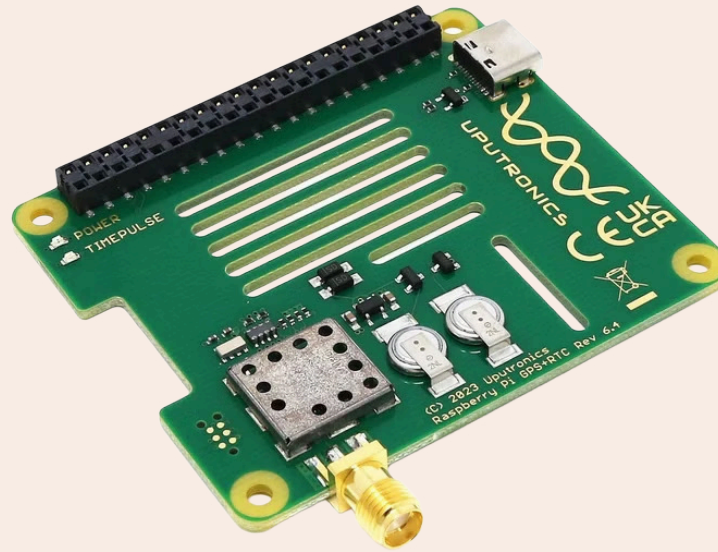
Card is around \$300.

Atomic Clock Module around \$800.

Building a Cheaper One.



Raspberry Pi 5



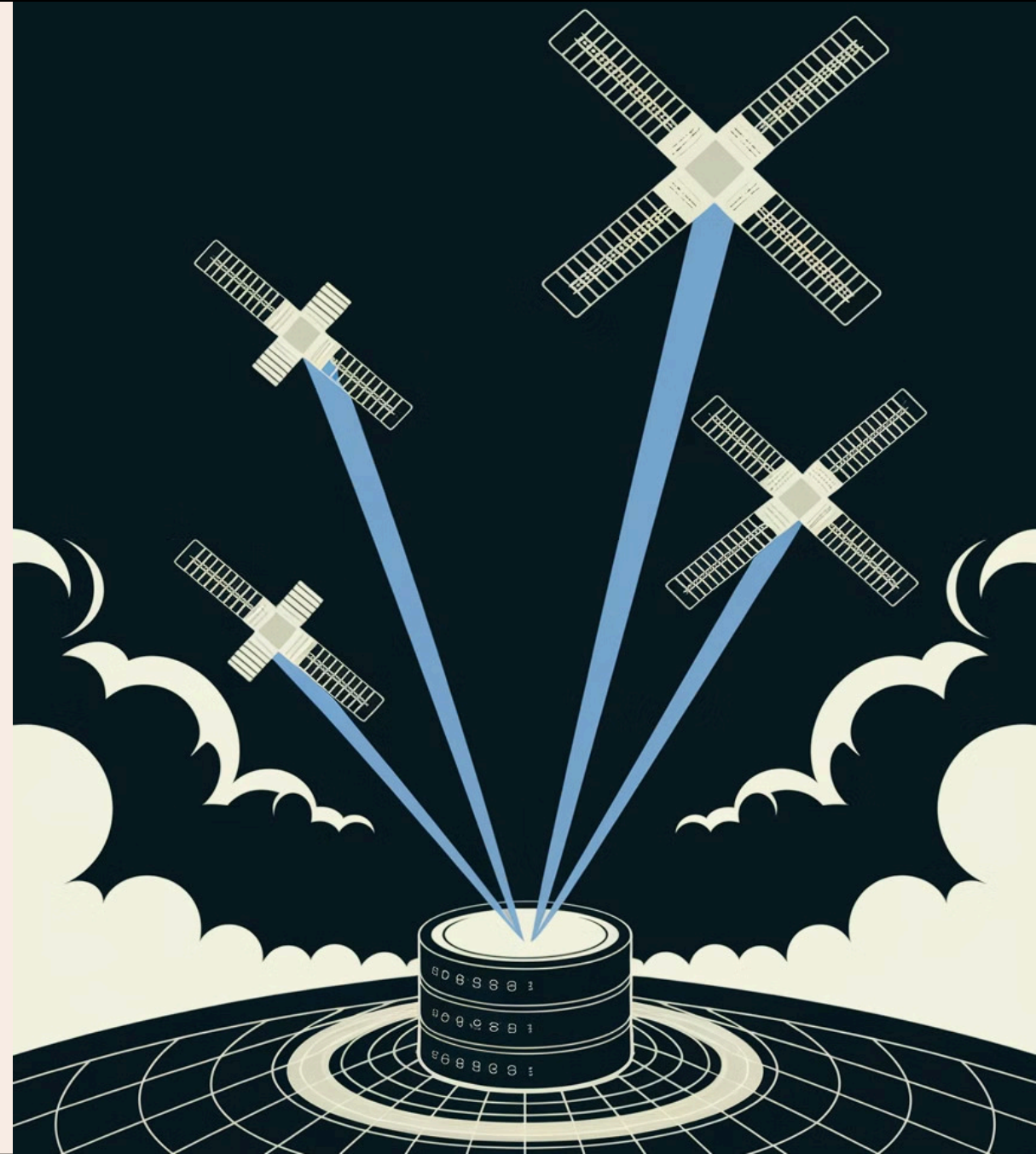
GNSS Pi-Hat

Pi5 8GB = £80.

GNSS Pi-Hat = £48.

No local high
precision
oscillator.

A Real World Example.



A DC Network.

- 2 x PTP SuperMicro servers.
 - With OCP Timecards + Rubidium Atomic Clocks.
 - Intel XL710-BM2 NIC's (40Gbps).
 - Each connected to independant GPS antenna.
- 4 x Spine Switches (Arista 7280 series).
- 16 x Leaf Switches (Arista 72xx series).
- Each leaf had 13 servers connected to it.
 - 208 servers in total.
- All 40Gbps.
- Every switch was a boundary clock.

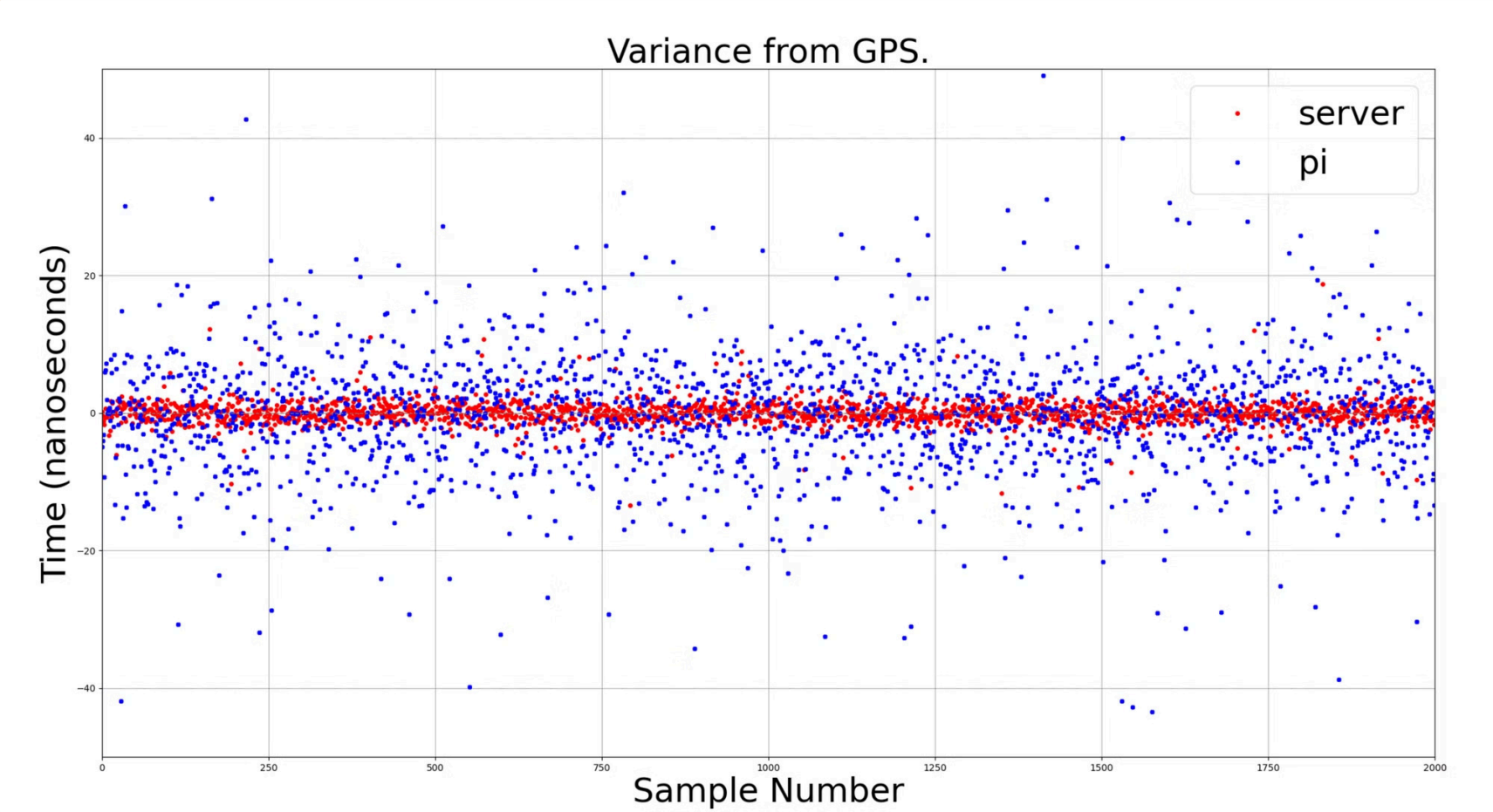


Testing.

- Take a test database server getting its time from PTP.
- Also connect a OCP Timecard + GPS reciever to it.
- Use phc2sys / pmc to compare OCP Timecard clock to system sync'd clock.
- Plots results !



Results.



What should I do ?



What are my requirements ?

- Do I need accurate walltime, or just sync'd devices ?
- What level of precision do I need under normal conditions ?
- What level of precision do I need with a failed external clock ?
- Do I need accuracy on hosts if the network goes down ?
- Do I have some specific requirement (like A/V SMPTE / ST2059 requires UDP transport).

What is my budget / kit ?

- GPS antenna ?
- Master Clock (bought vs built) ?
- Network switches / routers with PTP ?
 - Transparent vs Boundary Clock ?

Precision Comparison.

Synchronization Method	Expected Precision
NTP over Internet	1-10ms.
NTP over LAN (no QoS).	1-3ms.
NTP over LAN (priority-queue).	~1ms.
PTP End-to-End with GPS	100-500ns.
PTP Peer-to-Peer with GPS	5-50ns.

Misc Notes.

- Beware of Hypervisors and Time Sync.
- If you can make all your network devices boundary clocks, it is likely better than having transparent clocks.
- Beware of Leap Seconds (use Google smear method ?).
 - Remember TAI vs UTC offset (leap seconds).
- Use monitoring to check your timesync is working as expected.
- NTP interleaved mode works a bit like FOLLOW_UP messages and can improve NTP accuracy in stable networks.

Questions ?

