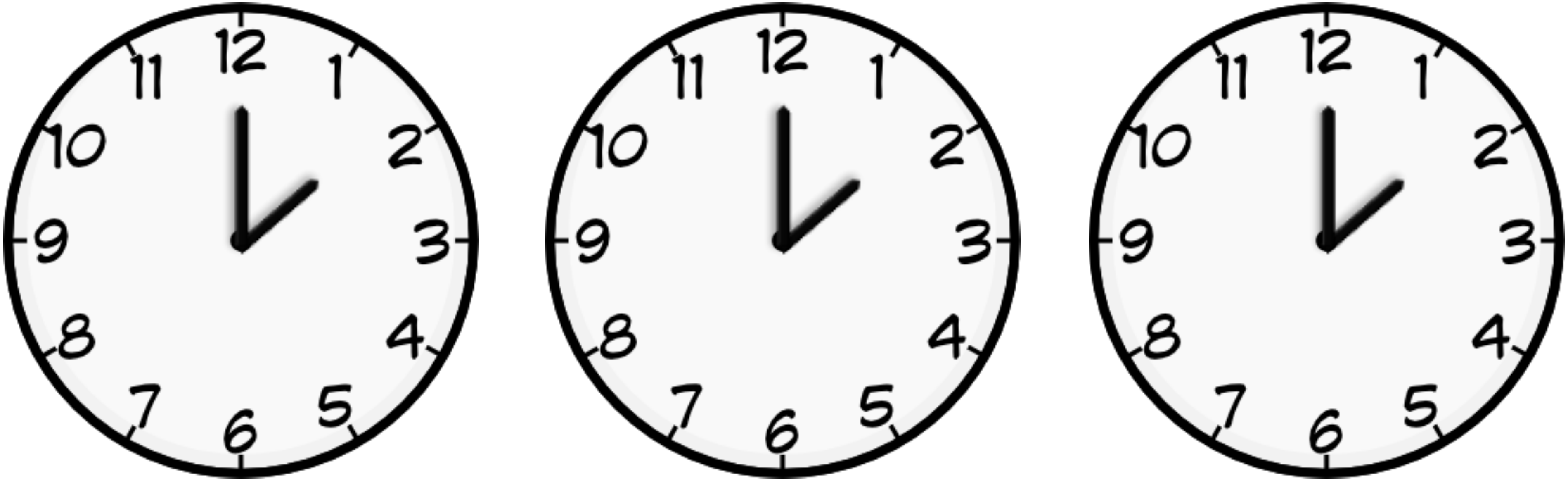
Globally Synchronized time via Datacenter Networks

Ki Suh Lee
Cornell University

Joint work with Han Wang, Vishal Shrivastav and Hakim Weatherspoon

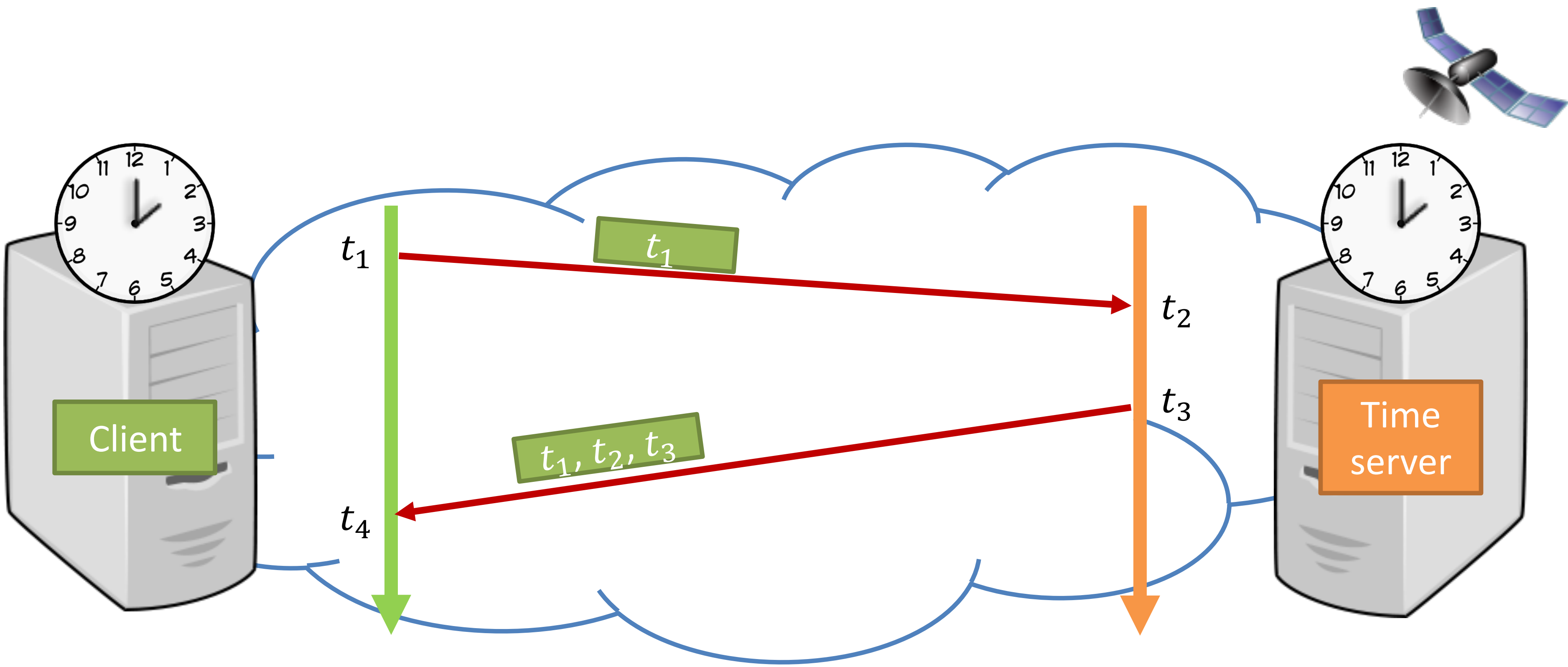
Synchronized Clocks

- Fundamental for network and distributed systems
  - OWD, Monitoring, Coordination, Snapshots, Updates, …
- Goal: *Minimized and bounded precision with scalability*
  - Minimized and bounded precision: *hundreds of nanoseconds*
  - Scalability: *Entire datacenter*



Clock Synchronization Protocol

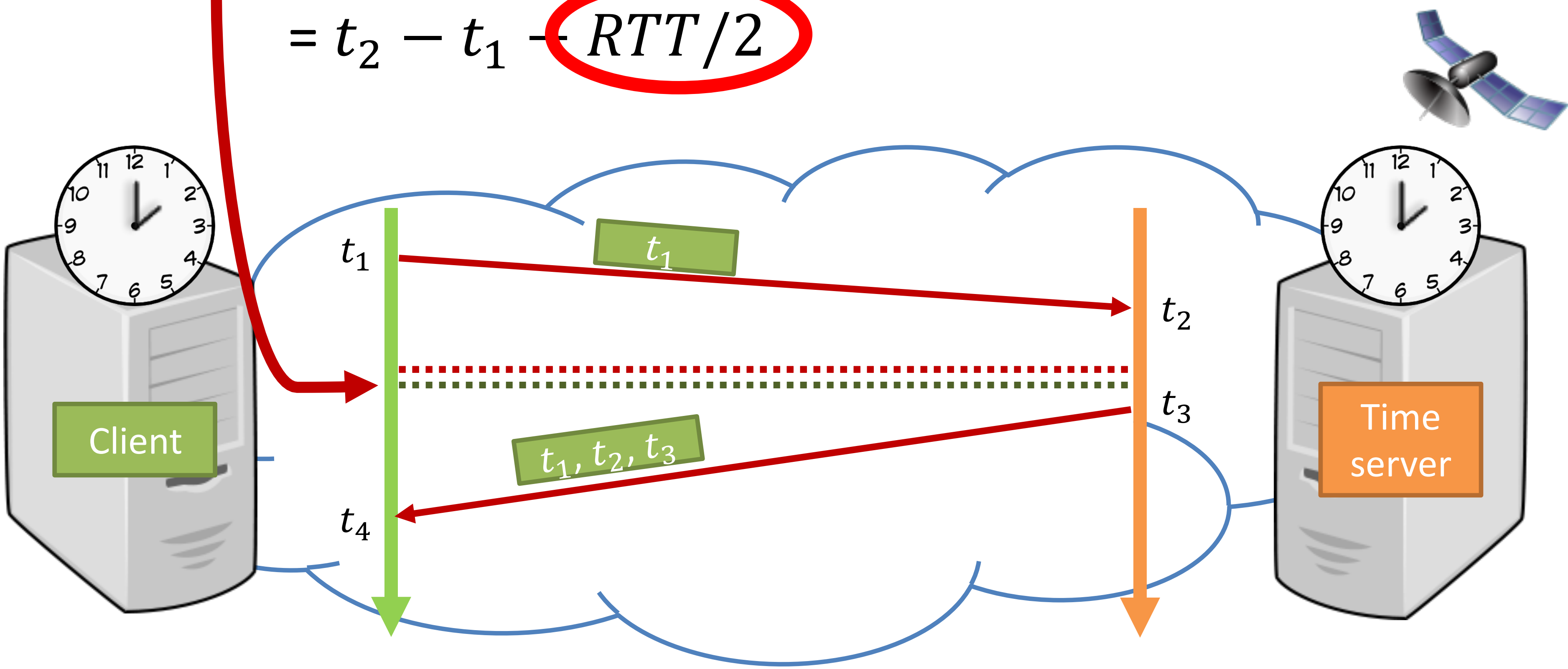
- Offset: Time difference between two clocks
- Precision: The worst case of offset



Clock Synchronization Protocol

- RTT = $(t_4 - t_1) - (t_3 - t_2)$
- Offset =

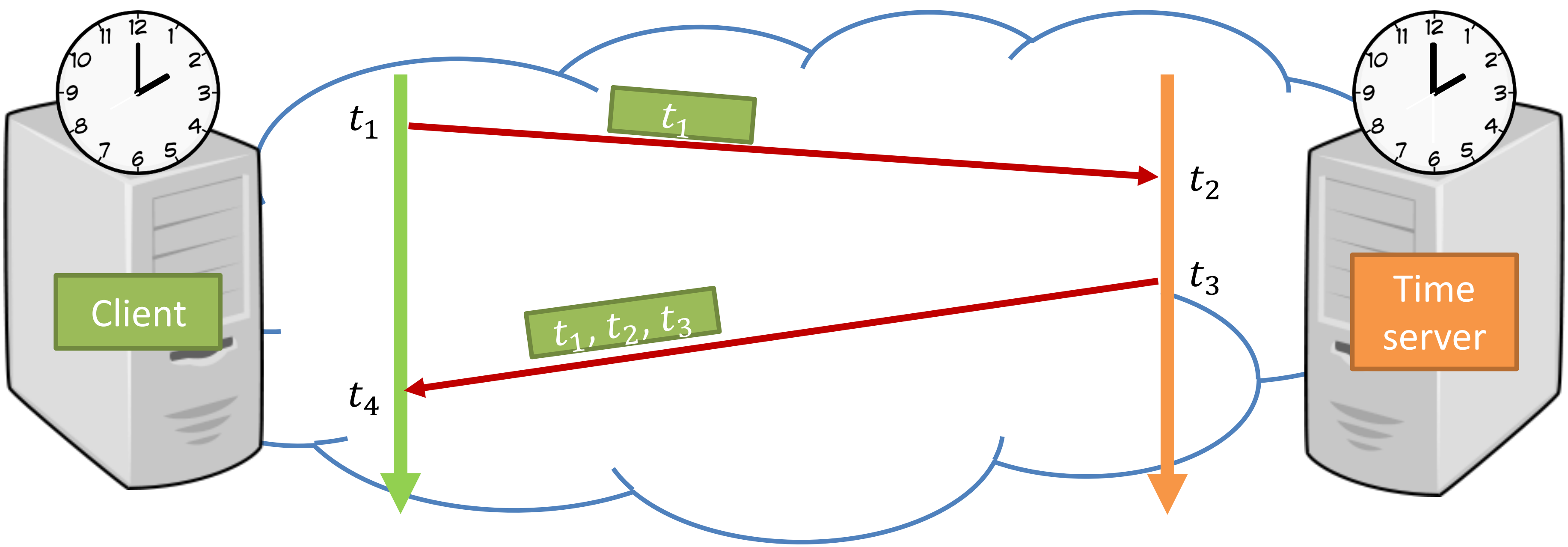
= $t_2 - t_1 - RTT/2$



Current time protocols
do NOT provide ***bounded precision,***
due to ***uncertainty*** in measured RTT!

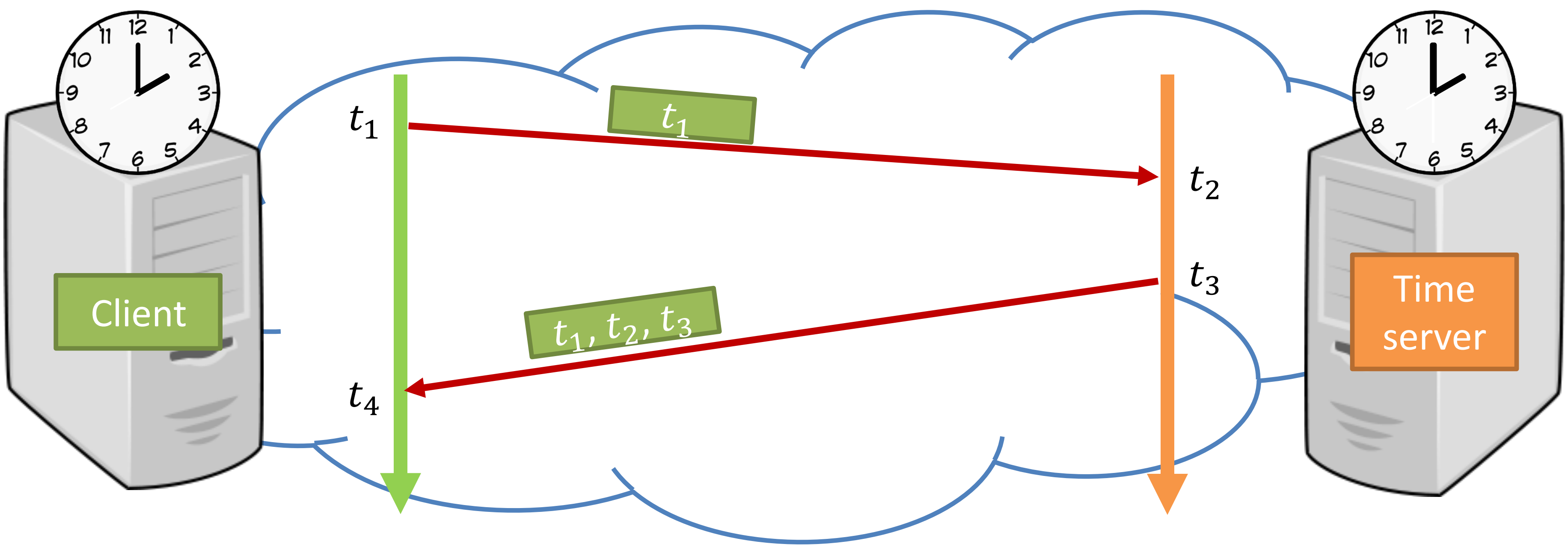
Challenge: RTT is not accurate

- Errors from
  - Oscillator skew
  - Inaccurate Timestamping
  - Network Stack
  - Network Jitter



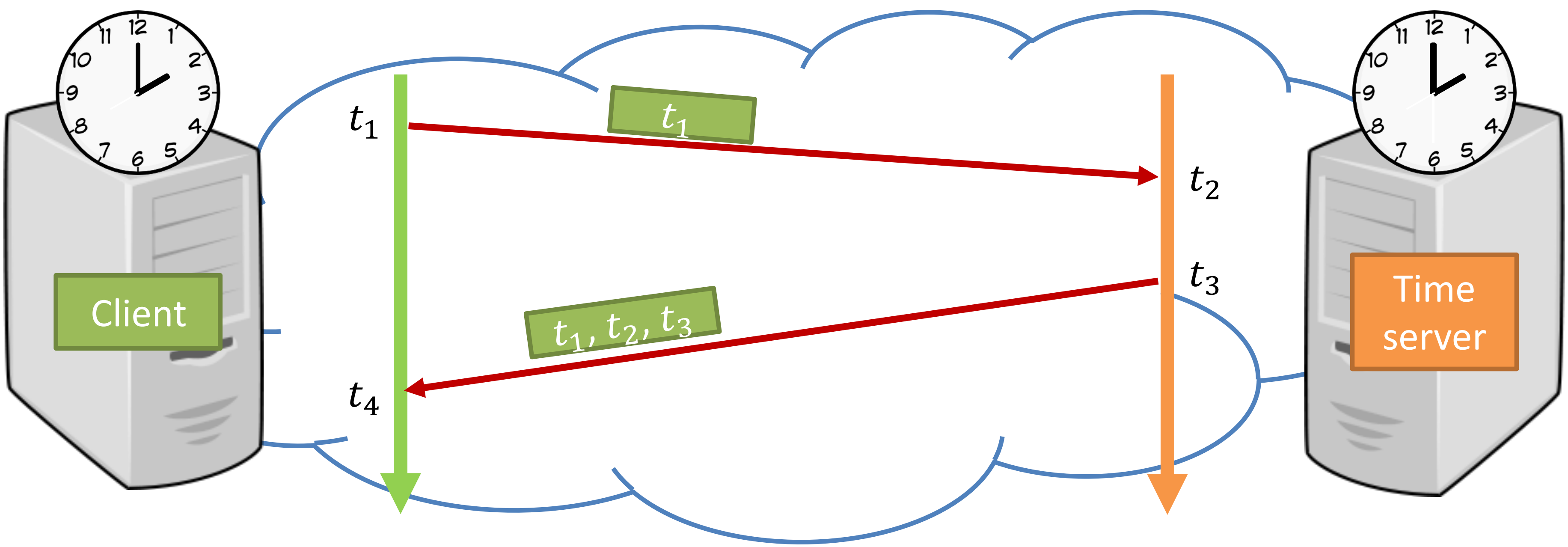
Challenge: RTT is not accurate

- Errors from
  - Oscillator skew
  - Inaccurate Timestamping
  - Network Stack
  - Network Jitter
- PTP
  - Hardware timestamping
  - PTP-enabled switches
  - Filtering / Smoothing



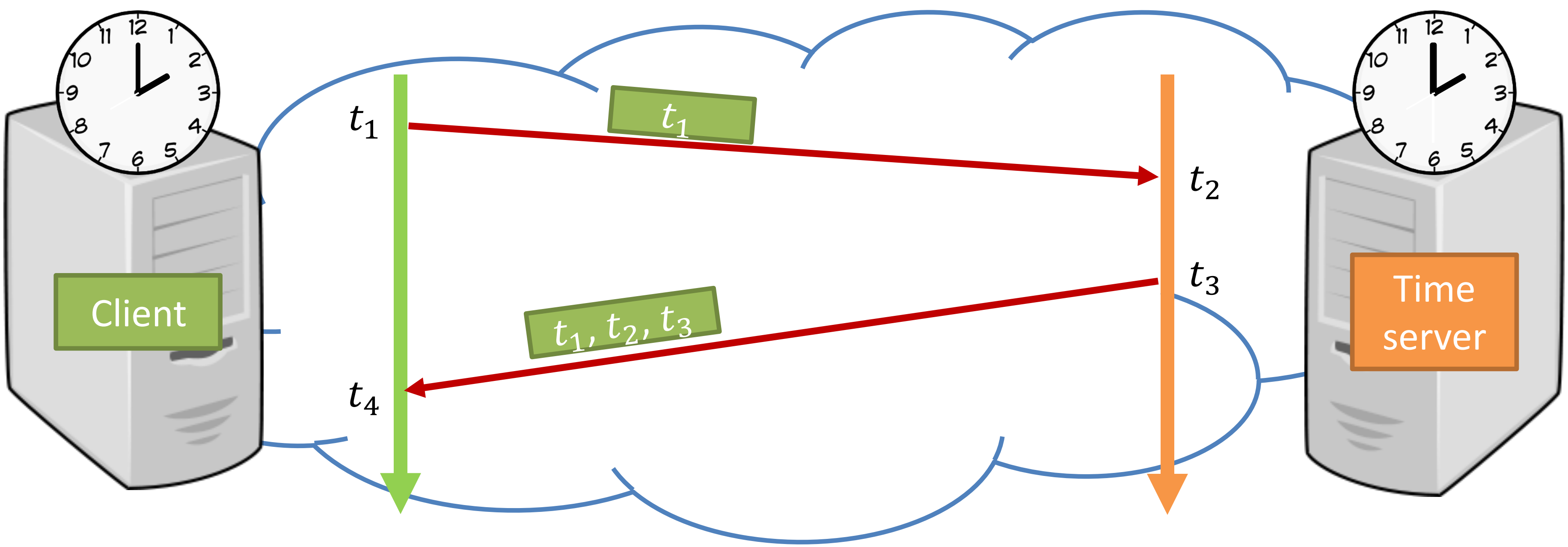
Challenge: Scalability

- Re-synchronization period vs. Network overhead
- Limited number of clients



Synchronization Protocols

	Precision	Scalability	Overhead	Extra Hardware
NTP	us	Good	Moderate	None
PTP	sub-us	Good	Moderate	PTP-enabled devices
GPS	ns	Bad	None	Timing signal receivers, cables



Solution: Use ***the PHY*** to synchronize clocks

Application

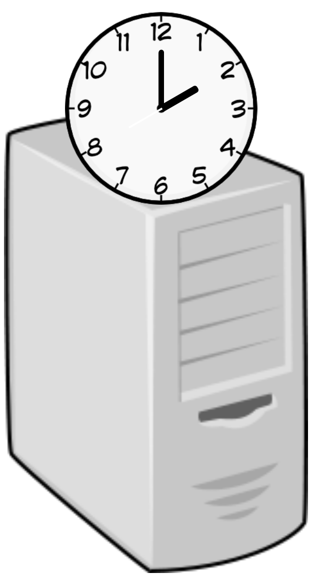
Transport

Network

Data Link

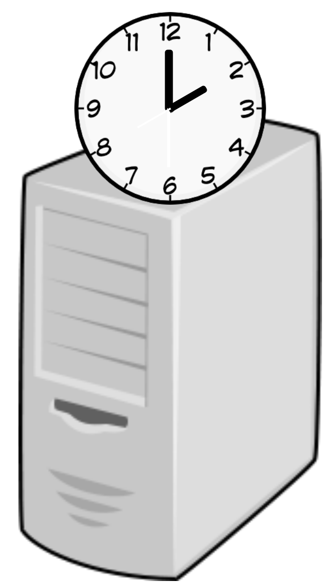
Physical

- Protocol in the PHY
  - Each physically link is already synchronized!
  - No protocol stack overhead
  - No network overhead
  - Scalable: peer-to-peer and decentralized



t_b

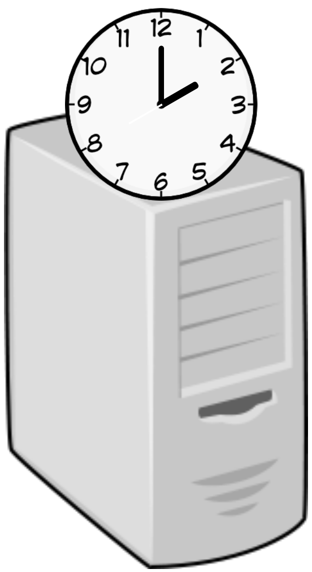
t_c

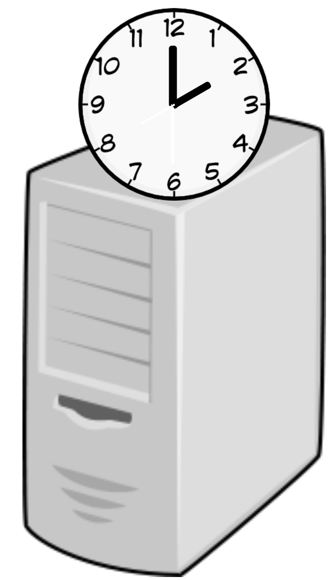


DTP: Datacenter Time Protocol

- Application
- Transport
- Network
- Data Link
- Physical

- ***Highly Scalable with <u>bounded</u> precision!***
  - ~25ns (4 clock ticks) between peers
  - ~150ns for a datacenter with six hops
  - No Network Traffic
  - *Internal* Clock Synchronization
- End-to-End: ~200ns precision!

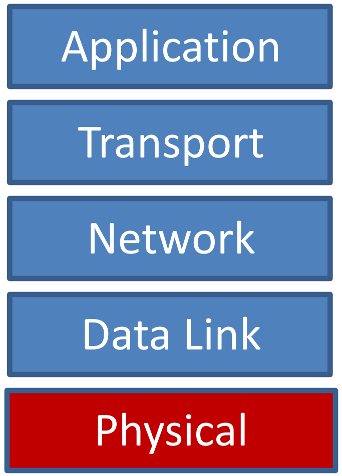




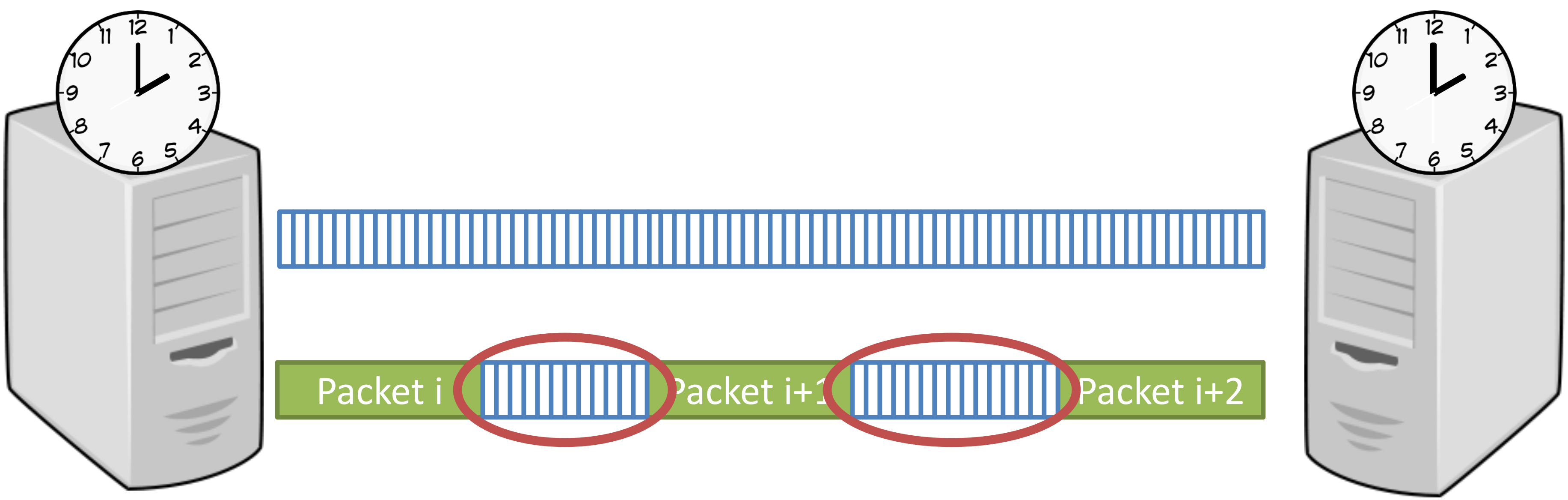
Outline

- Introduction
- Design
- Evaluation
- Discussion
- Conclusion

DTP: Datacenter Time Protocol



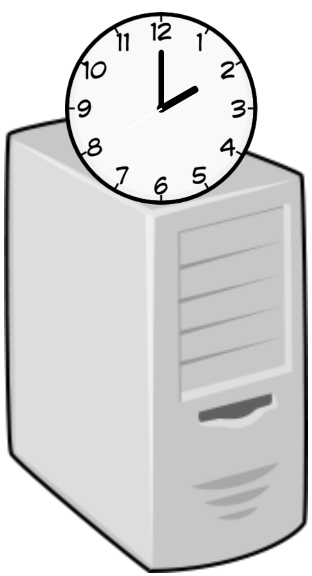
- 10G Background
  - Continuous /I/s when there is no packet
  - At least 12 /I/s between two Ethernet frames

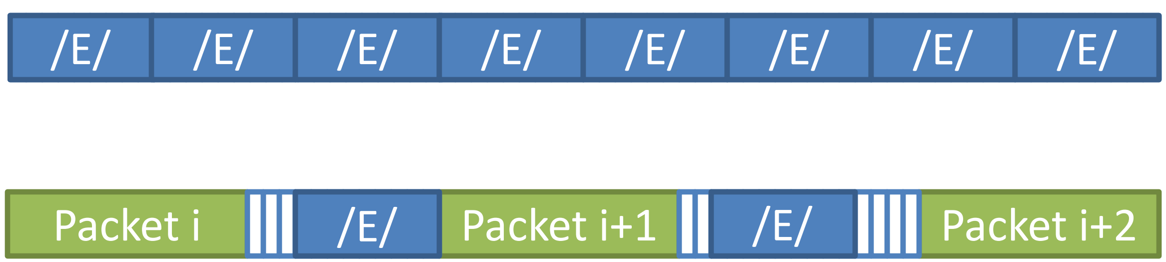


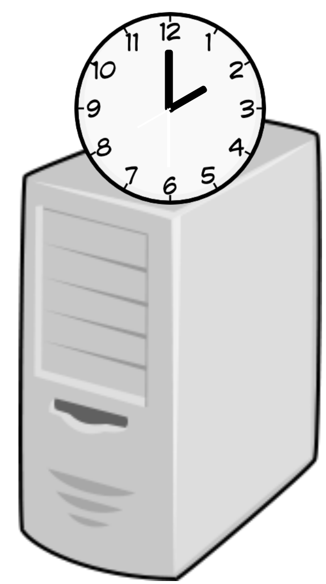
DTP: Datacenter Time Protocol

- Application
- Transport
- Network
- Data Link
- **Physical**

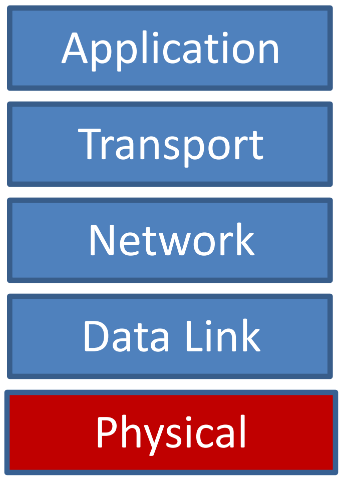
- 10G Background
  - Continuous /I/s when there is no packet
  - At least 12 /I/s between two Ethernet frames
  - 1 Control block (/E/, 66bit) = 8 /I/s
  - At least 1 /E/ between any two frames
  - The PHY is run by 156.25MHz
    - Period is 6.4ns



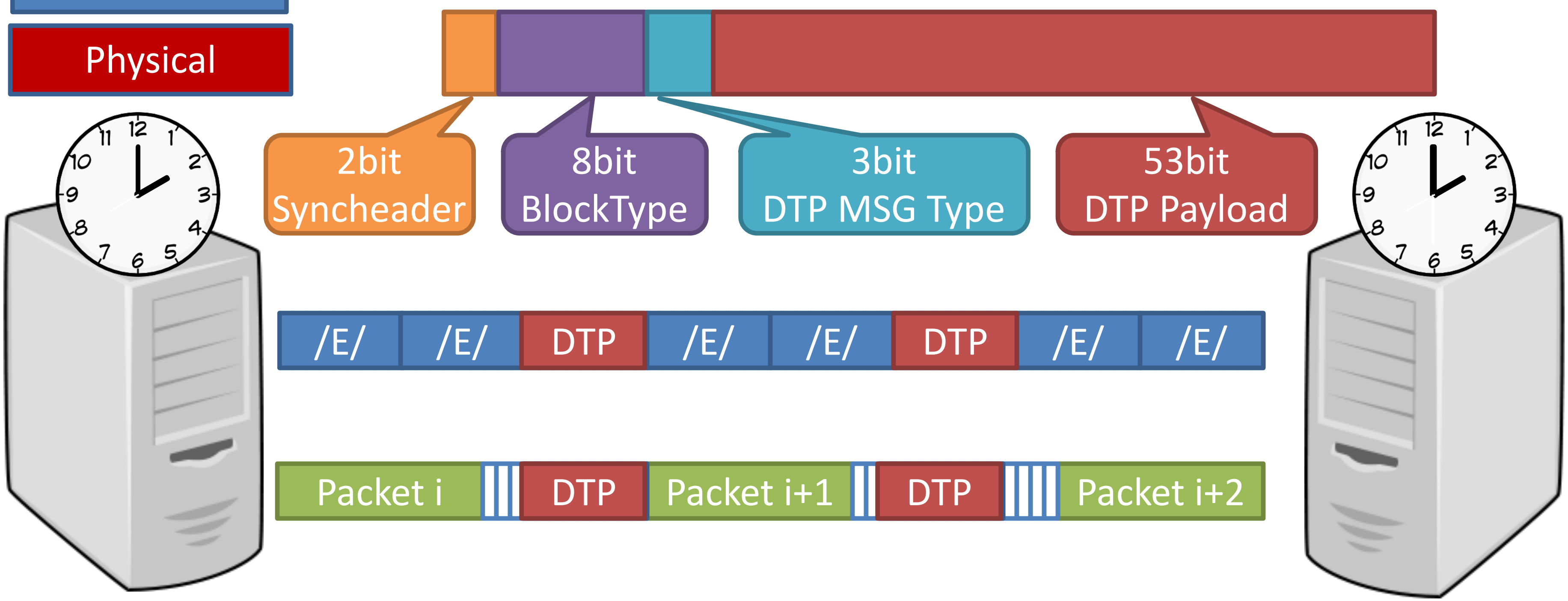




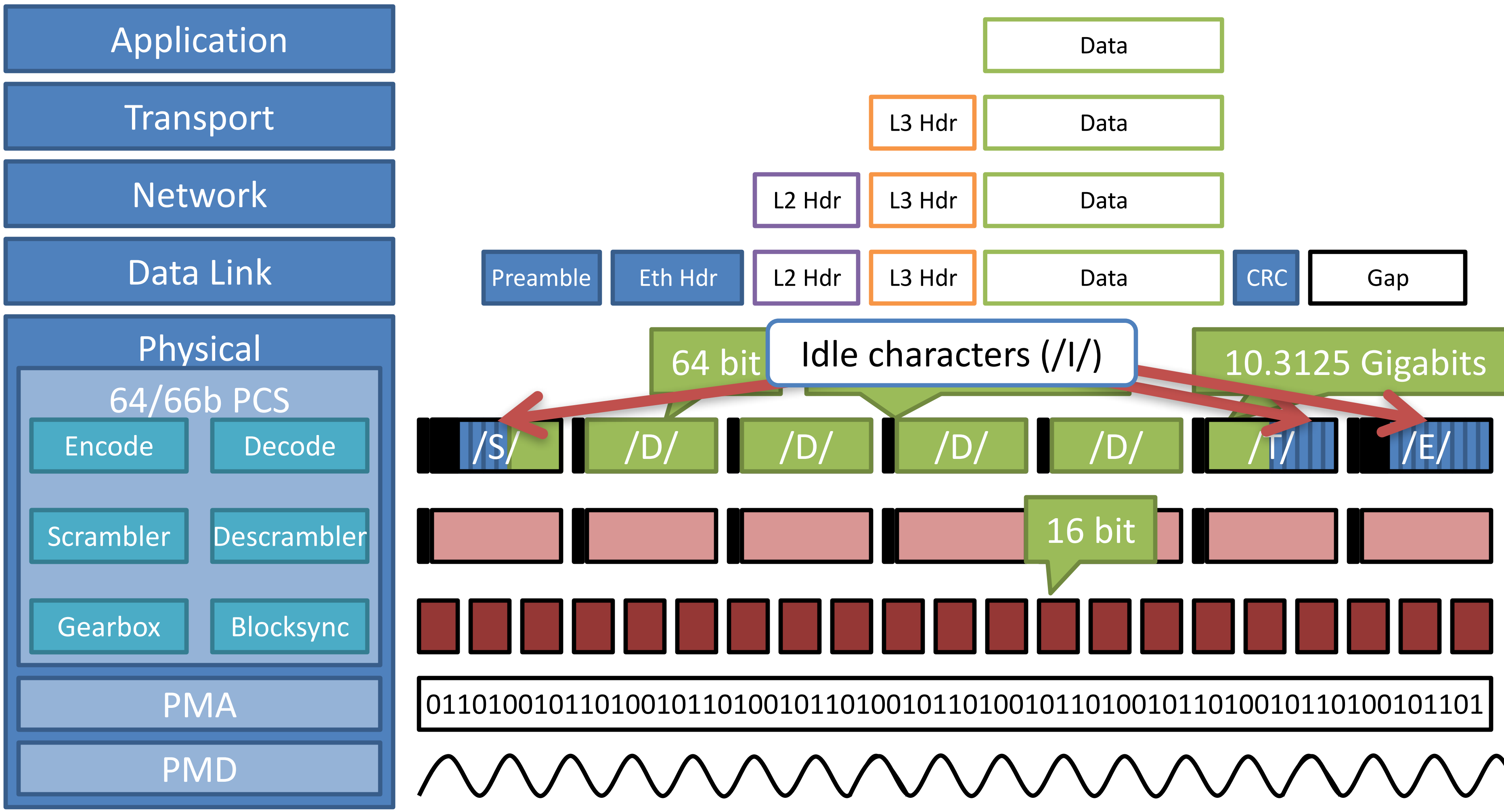
DTP: Datacenter Time Protocol



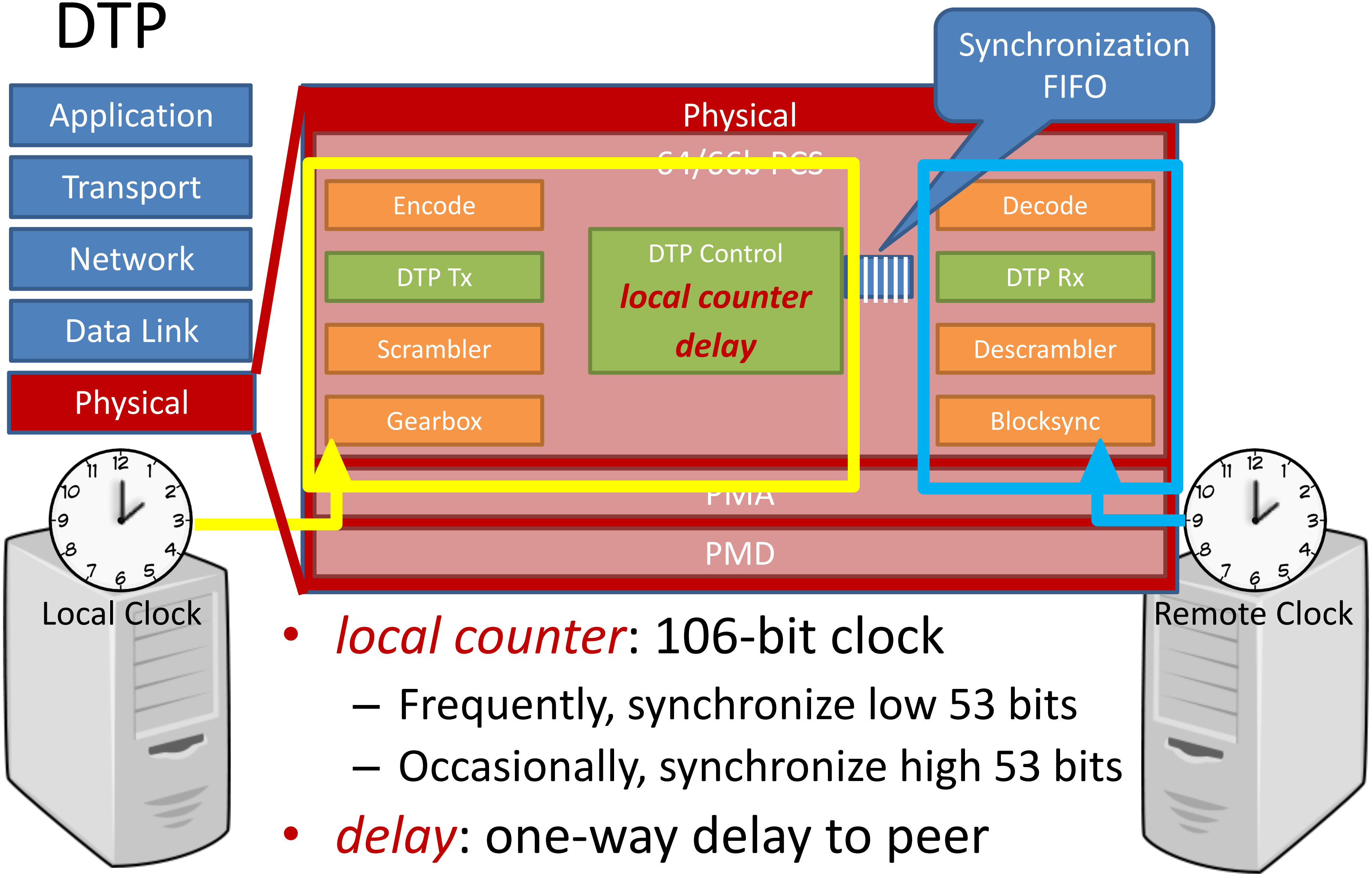
- DTP overwrites /E/ to send protocol messages
  - Frequent messaging
  - No overhead to Ethernet (L2)



10GbE Network Stack



DTP

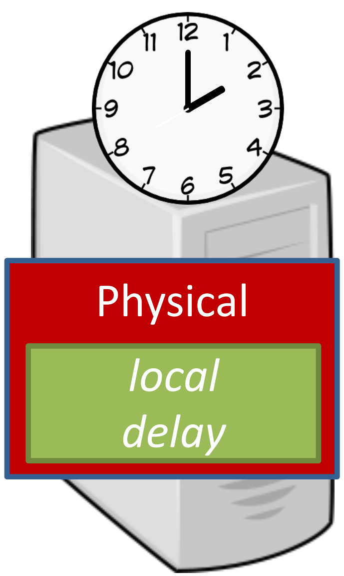


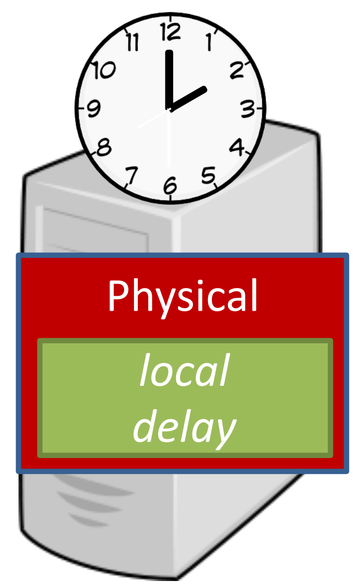
- *local counter*: 106-bit clock
  - Frequently, synchronize low 53 bits
  - Occasionally, synchronize high 53 bits
- *delay*: one-way delay to peer

DTP

Application
Transport
Network
Data Link
Physical

- Runs in two phases between two peers
  - Init Phase: Measuring OWD
  - Beacon Phase: Re-Synchronization

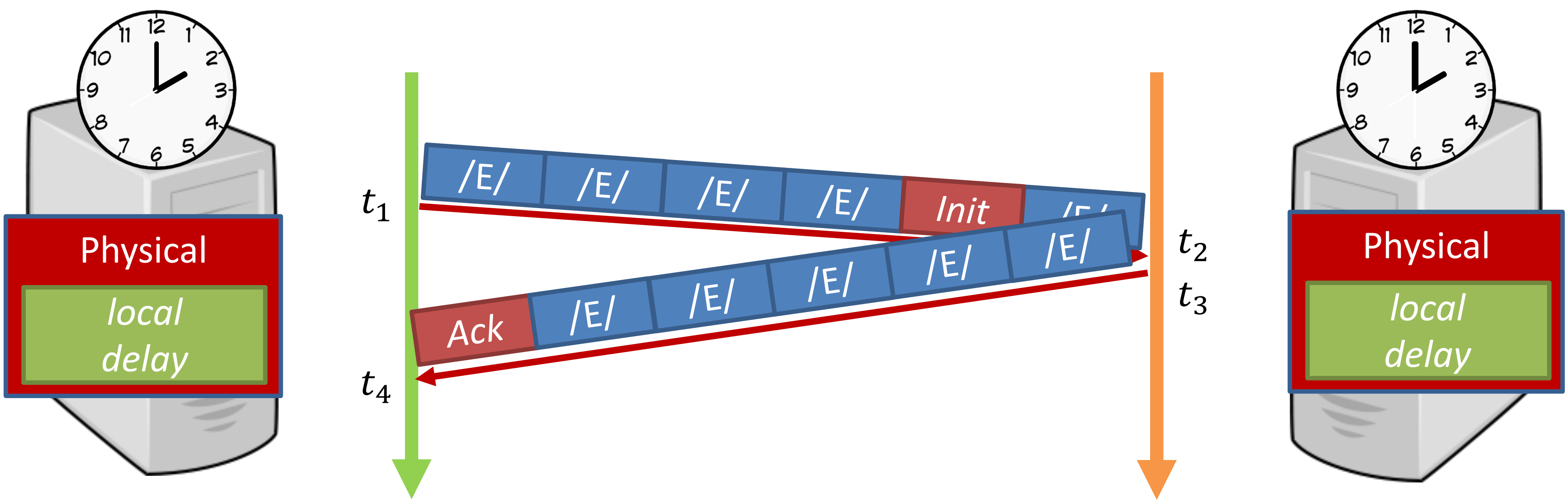




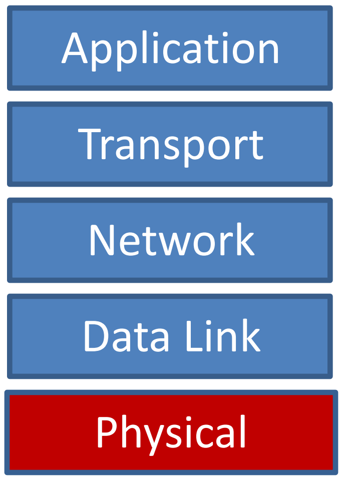
DTP: Init Phase

- Application
- Transport
- Network
- Data Link
- Physical

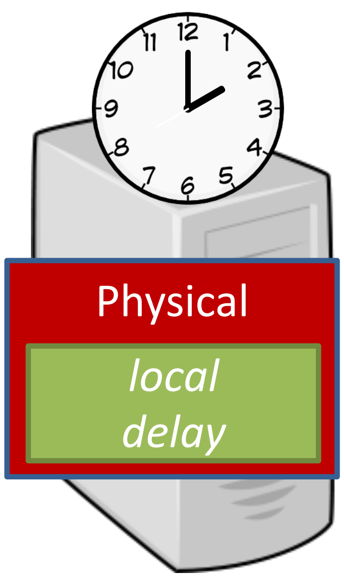
- $delay = (t_4 - t_1 - \alpha)/2$
  - α=3: Ensure *delay* is always less than actual delay
- Introduce 2 clock tick errors
  - Due to oscillator skew, timing and Sync FIFO

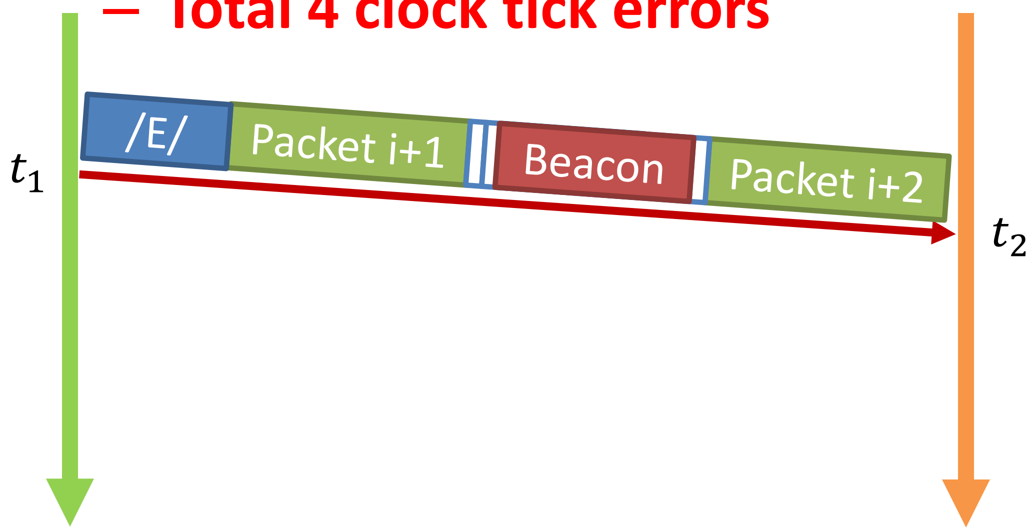


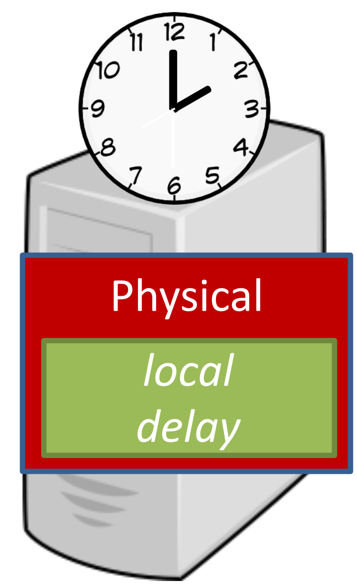
DTP: Beacon Phase



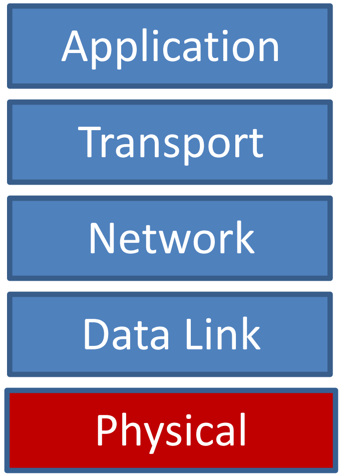
- *local* = max (*local*, *remote*+*delay*)
- Frequent messages
  - Every 1.2 us (200 clock ticks) with MTU packets
  - Every 7.2 us (1200 clock ticks) with Jumbo packets
- Introduces 2 clock tick errors
  - **Total 4 clock tick errors**



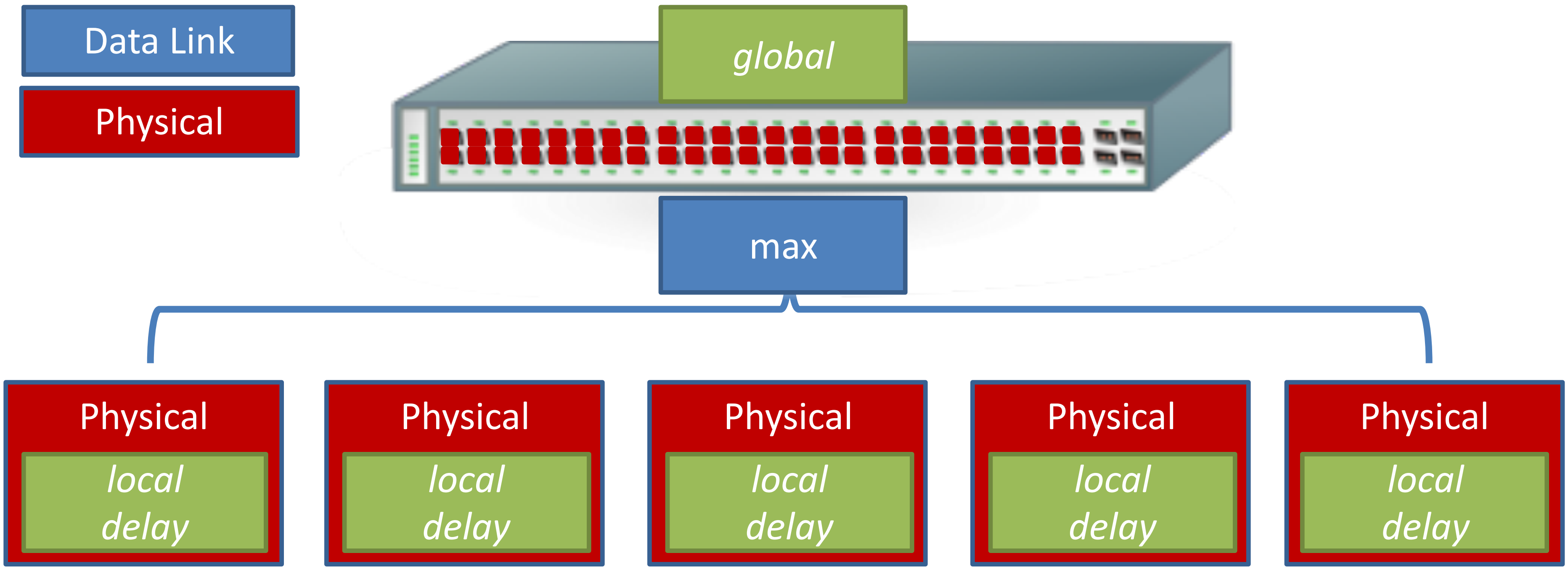




DTP Switch



- *global* = max(*local* counters)
- Propagates *global* via Beacon messages



DTP Daemon

- End-to-End precision
- Access the DTP counter via PCIe
- Estimate DTP time using invariant TSC counter

DTP Property

- Bounded Precision in hardware
  - Bounded by 4T (=25.6ns, T=oscillator tick is 6.4ns)
  - Network precision bounded by 4*TD*
    - D is network diameter in hops
- Requires NIC and switch modifications
  - PTP also requires PTP-enabled devices

DTP vs PTP

	PTP	DTP
Oscillator Skew		
Timestamping	HW - timestamping	PHY timestamping
Network Stack	Not involved	Not involved
Network Jitter	Transparent Clock Boundary Clock	No jitter
Precision	Unbounded Tens to Hundreds ns (When Idle)	Bounded

DTP: Topics discussed in paper

- Handling failure
- Different standards: 1GbE, 25GbE, 40GbE, 100GbE, etc
- External synchronization (i.e. synchronizing to true time)
- Incremental deployment

Handling failure

- Bit Errors
  - Ignores Bit errors in MSBs
  - Appends checksum for low LSBs

- Faulty Devices
  - When too many jumps outside the bound

Different Standards

Data Rate	Encoding	Data Width	Frequency	Period	Δ
1 GbE	8b/10b	8bit	125MHz	8ns	25
10 GbE	64b/66b	32bit	156.25MHz	6.4ns	20
40 GbE	64b/66b	64bit	625MHz	1.6ns	5
100 GbE	64b/66b	64bit	1562.5MHz	0.64ns	2

External Synchronization

- A *master* server
  - Connected to a reference time
  - Broadcasts the mapping between *DTP* and wall time
- Client servers
  - Interpolates time using *DTP* counters

Incremental Deployment

- Updates per rack
  - DTP-enabled switch
  - DTP-enabled NICs
  - One server acting as a *master* for wall time

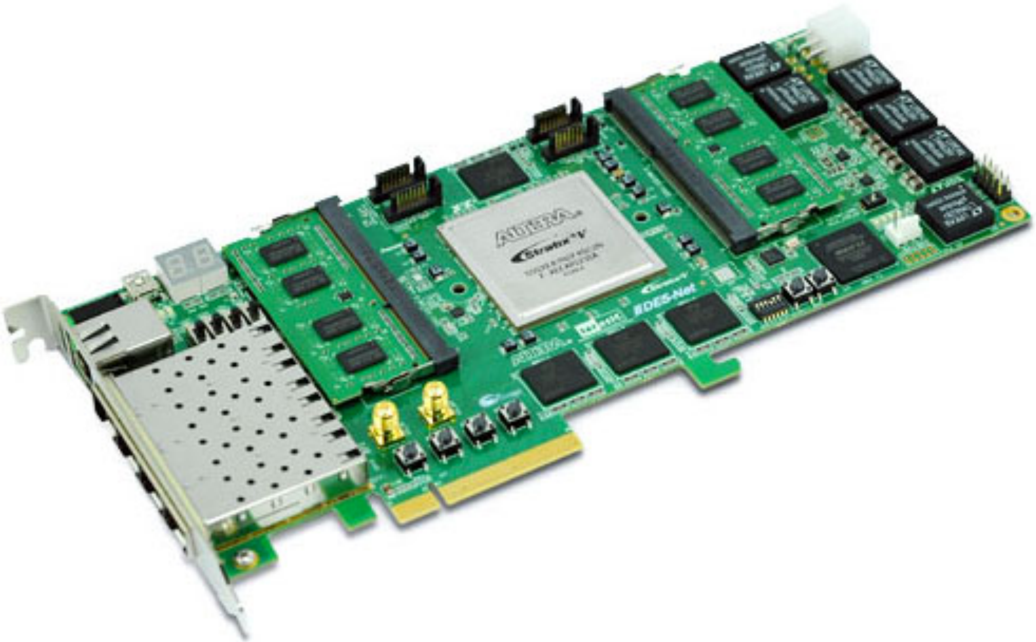
- Synchronizing Racks
  - DTP-enabled switch
  - DTP *beacon-join* message for synchronizing DTP counters
  - Select a new *master*

Outline

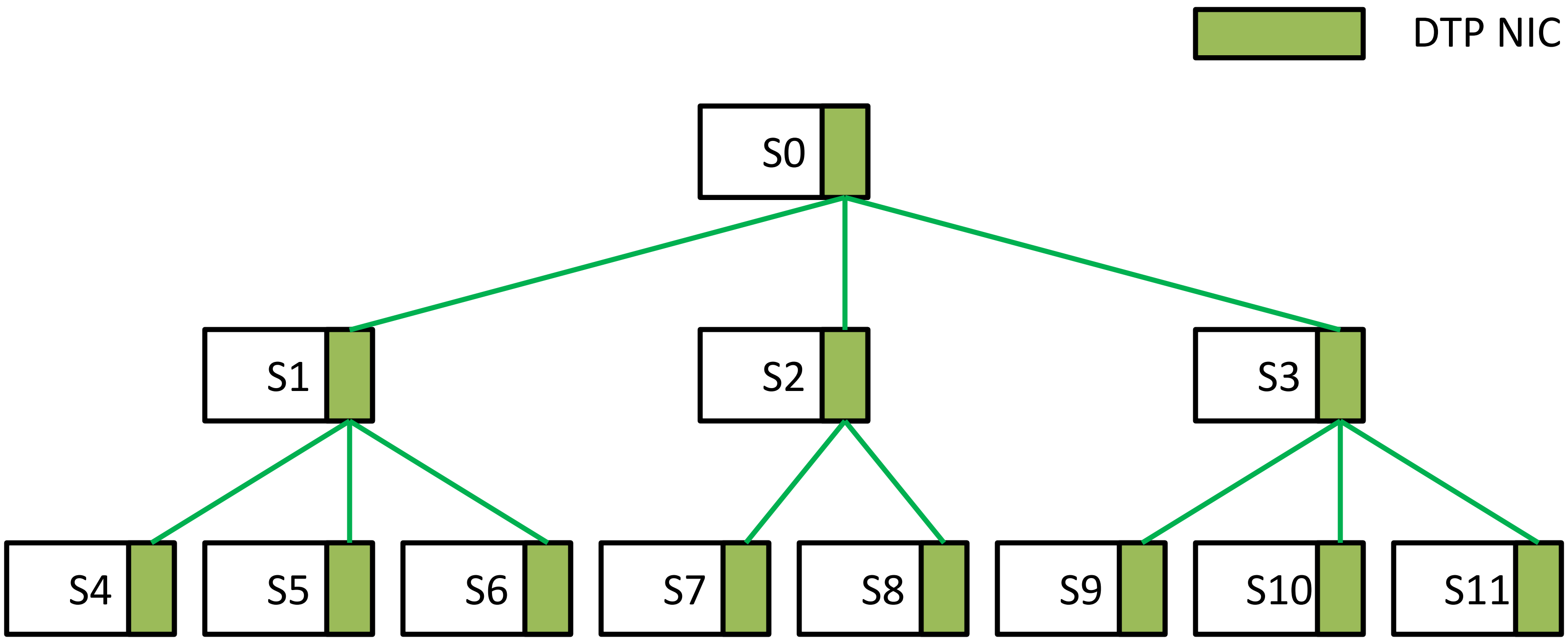
- Introduction
- Design
- Evaluation
- Discussion
- Conclusion

Evaluation

- DTP Prototype
  - Terasic DE5 board with Altera Stratix V
  - Using Bluespec and Connectal framework



Evaluation: DTP Topology

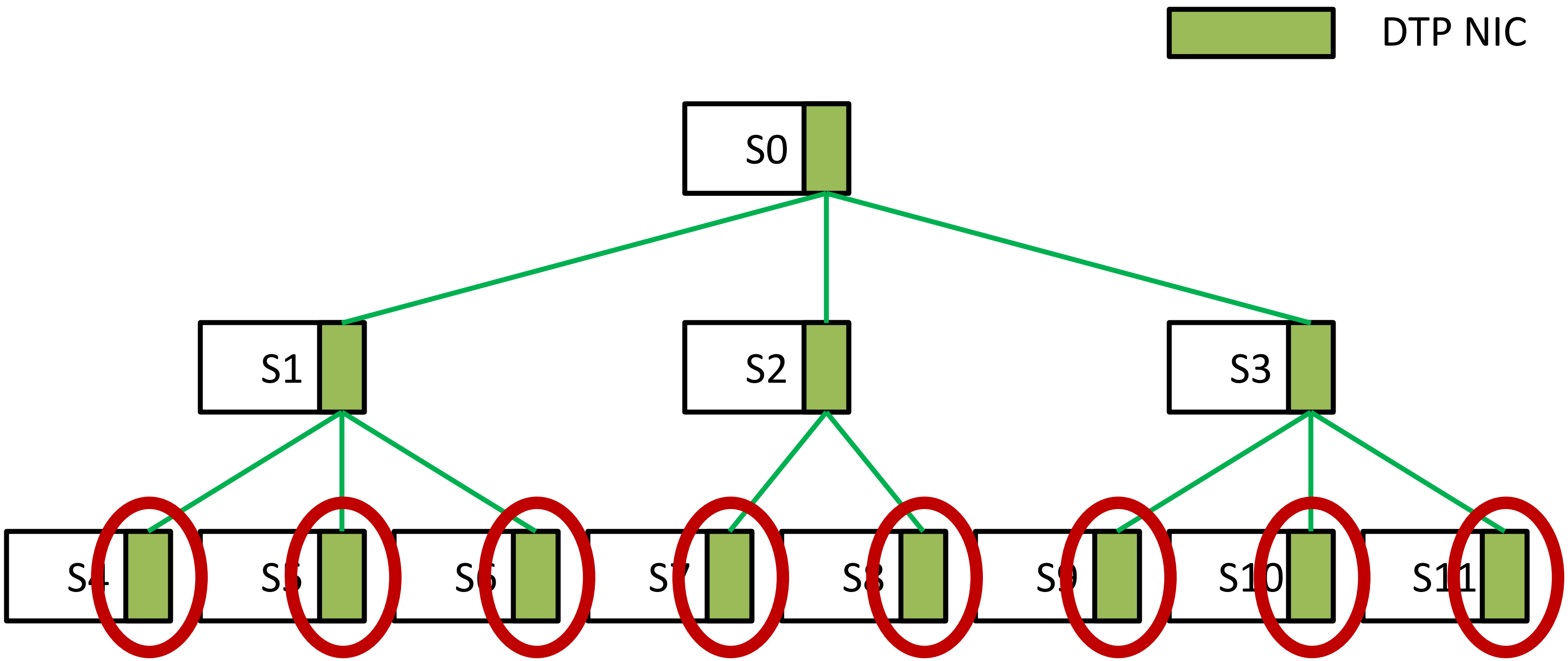


Measured offsets between peers

Evaluation: Logger

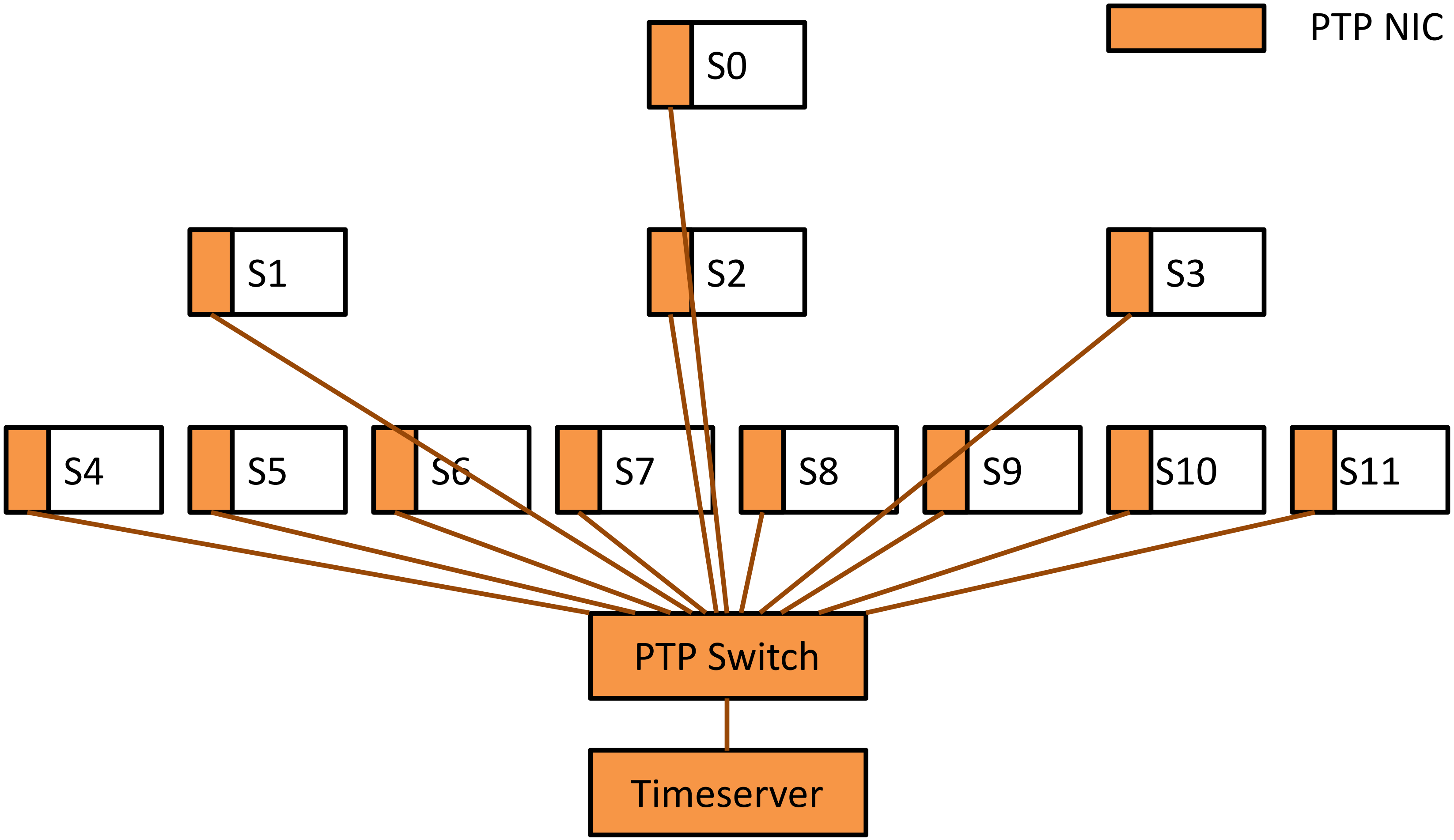
- Offset between peers: $t_3 - t_2 - \mathrm{OWD}$
- Offset between SW and HW: $t_2 - t_1$

Evaluation: DTP Topology

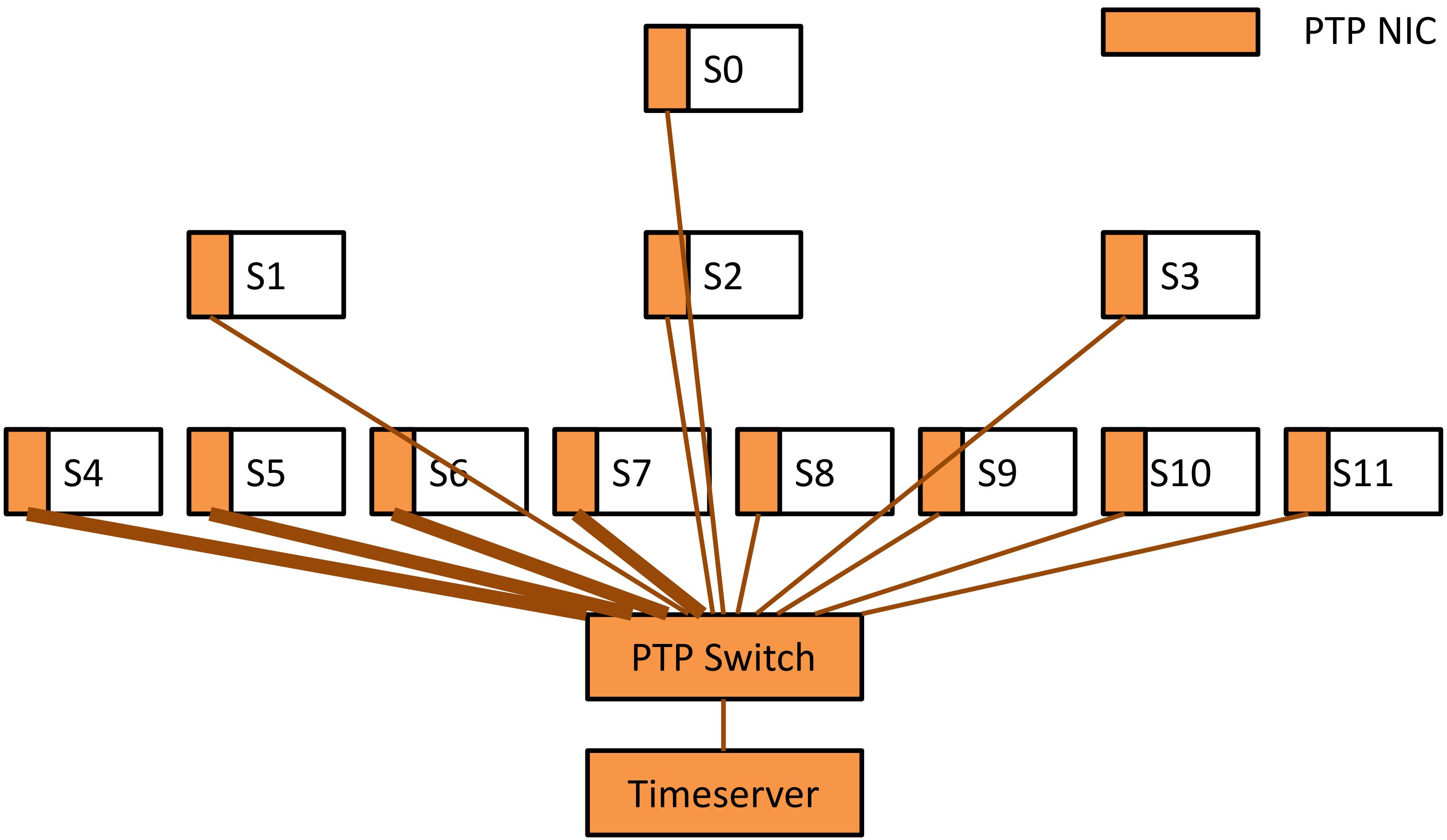


$$Offset = dtp_{SW} - dtp_{HW}$$

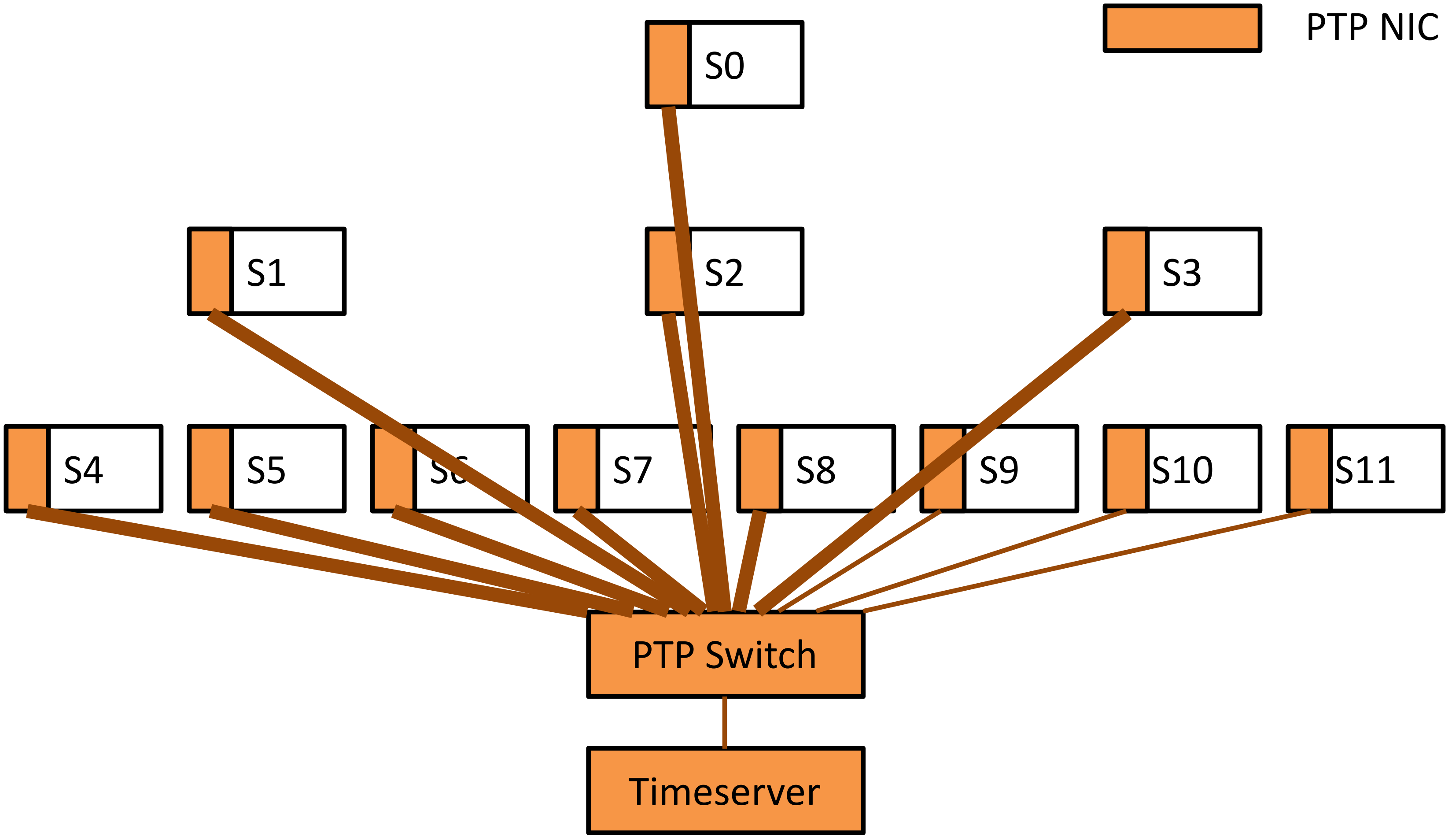
Evaluation: PTP Topology



Evaluation: PTP Topology

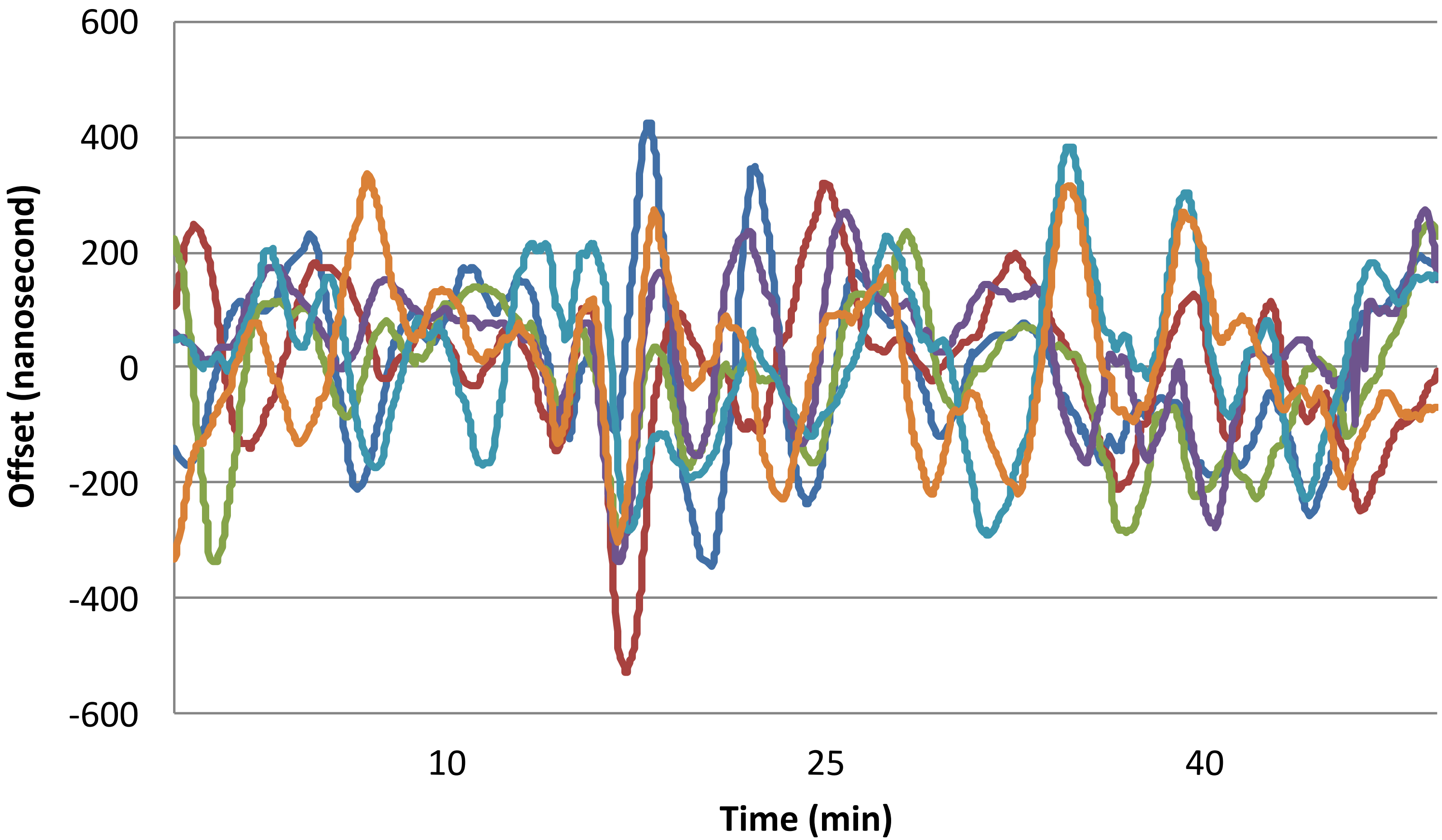


Evaluation: PTP Topology



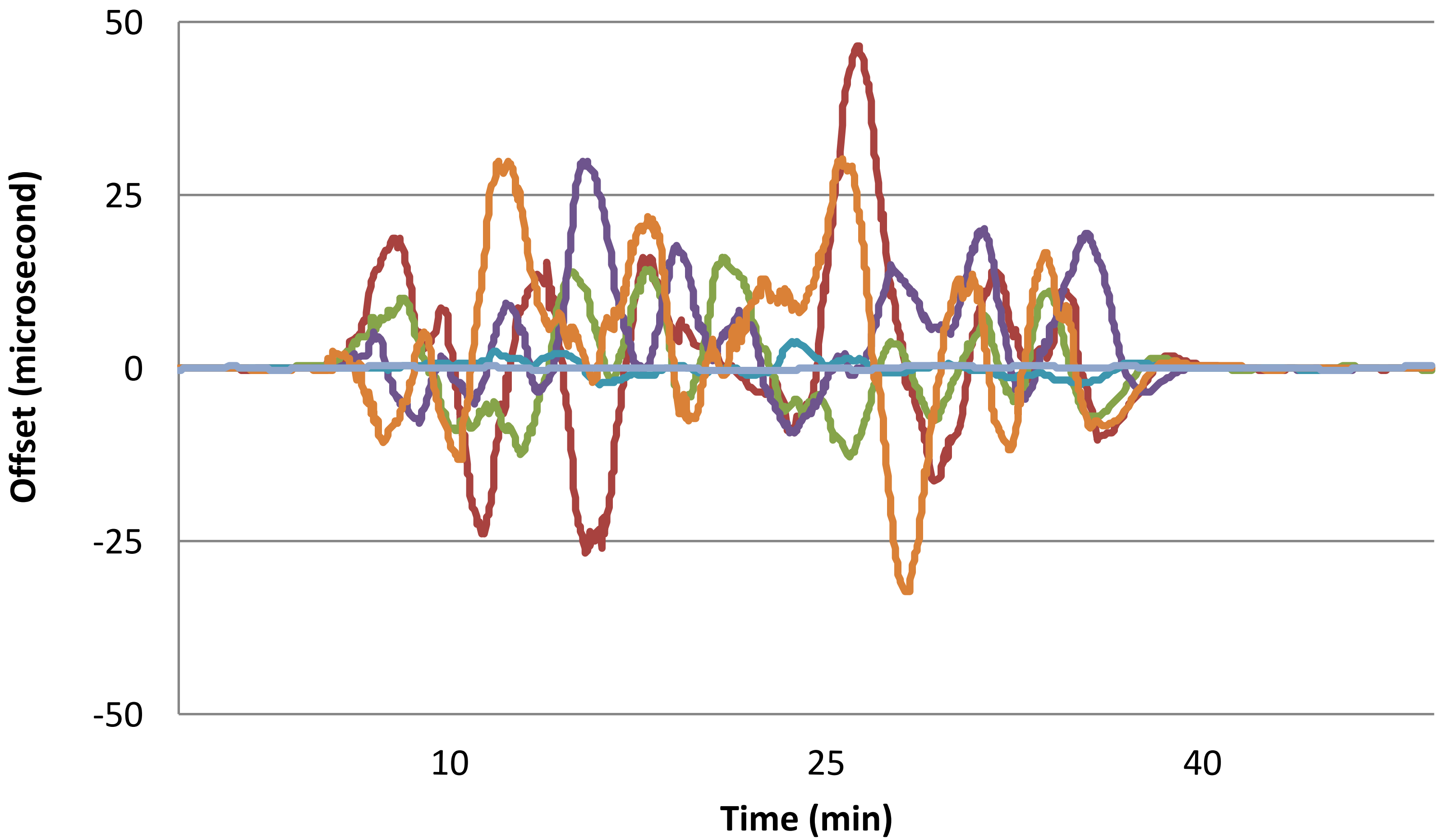
PTP: Idle Network (No traffic)

- *Tens to hundreds of nanosecond precision*



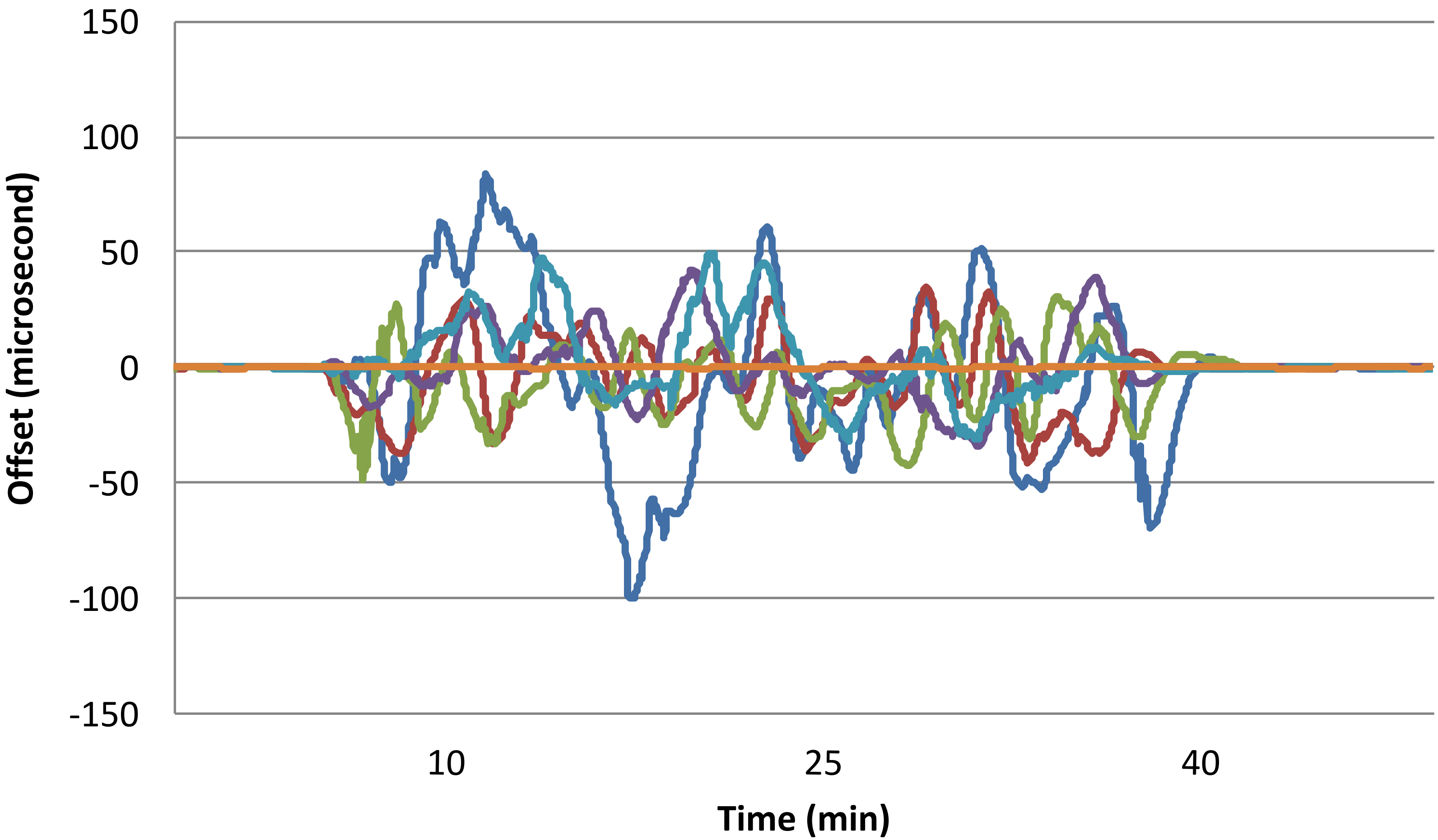
PTP: Medium Loaded (4 Gbps)

- *Tens of microseconds precision*



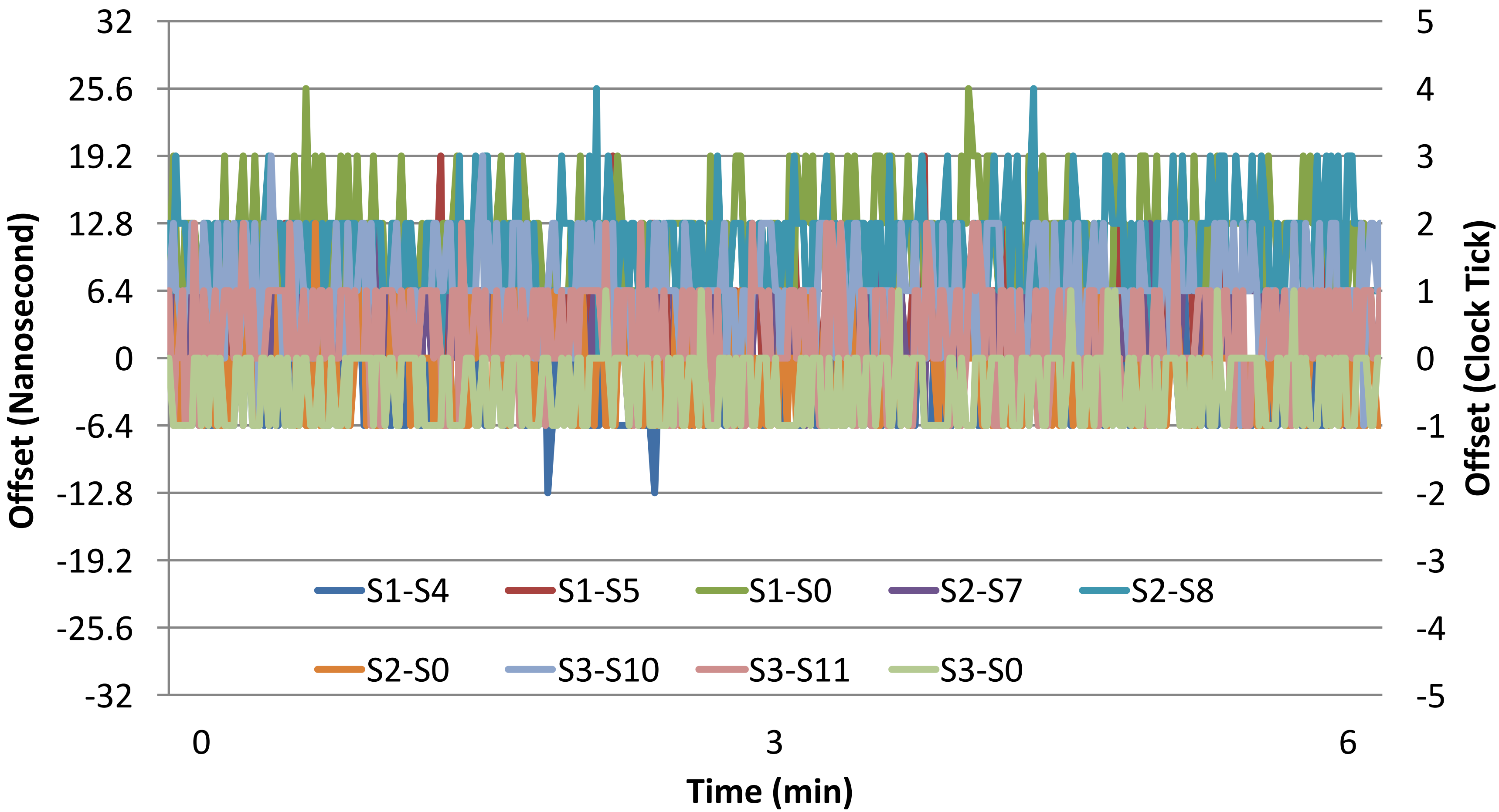
PTP: Heavily Loaded (9 Gbps)

- *Tens to hundreds of microsecond precision*



DTP: Heavily Loaded

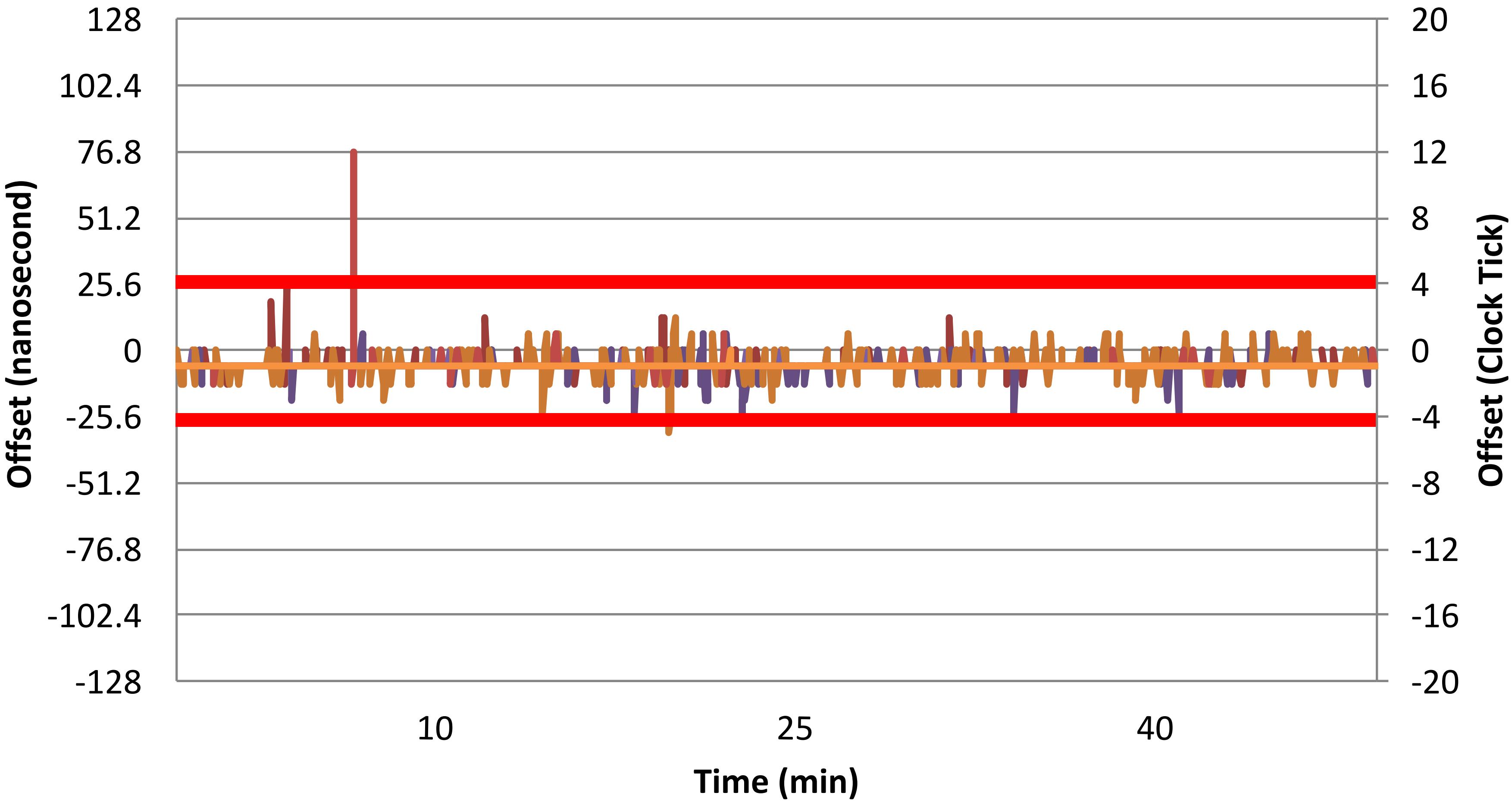
- *Always within 25.6ns (4 clock ticks) between peers*



DTP Daemon

DTP Daemon (after smoothing)

- *Usually can access the counter with 25.6 ns precision*

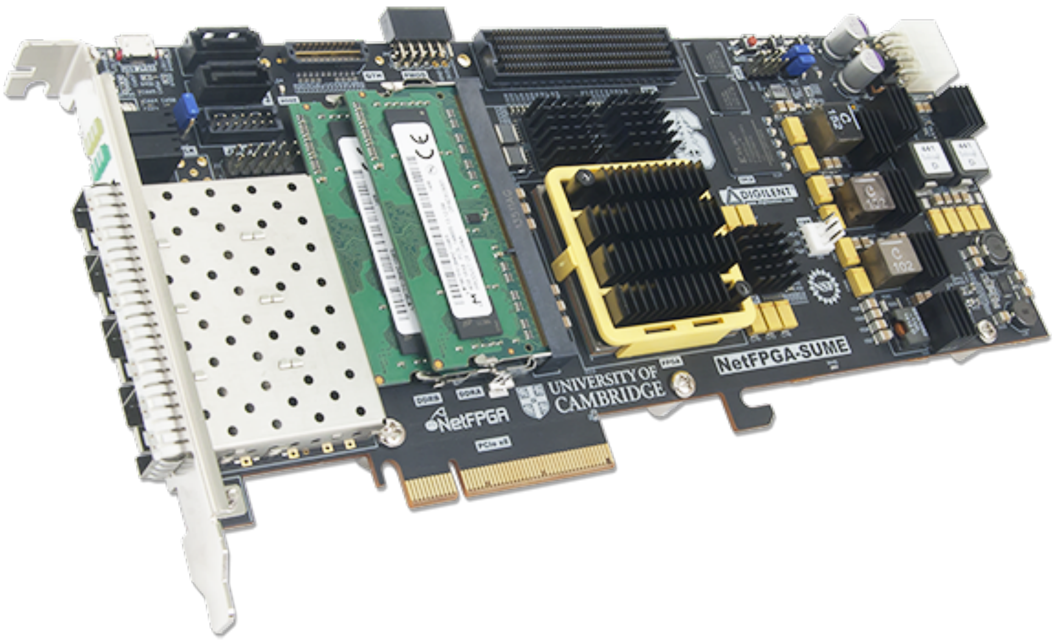


Outline

- Introduction
- Design
- Evaluation
- Discussion
- Conclusion

Next Steps

- Integration with OSNT (Open Source Network Tester)
  - NetFPGA SUME Board with Xilinx Virtex-7



Some Related Work

- Synchronous Ethernet (SyncE)
  - Synchronize the ***frequency*** of clocks
  - DTP, PTP synchronizes the ***time*** of clocks
- WhiteRabbit: PTP + SyncE
  - Sub-nanosecond precision
  - 1GbE only yet
- Commercial PTP + SyncE
  - Tens to hundreds of nanoseconds

Conclusion

- DTP provides *bounded precision* and *scalability*
  - Bounded precision: 4 clock ticks (25.6ns) between peers
  - Scalability: 153.6ns for a datacenter with six hops
  - Free: No Network Traffic
  - Applications: Usually within 25.6ns (without bounds)
  - End-to-End: 153.6 + 25.6 * 2 = 200ns!

Questions?

- http://github.com/hanw/sonic-lite
- http://sonic.cs.cornell.edu
- Email: kslee@cs.cornell.edu

- Come to Poster session tomorrow!