Ultra Low Latency

Calin Poenaru
Cisco Systems Romania

October, 2012

Agenda

- Common understanding of "Ultra Low Latency"
- Usual ways to measure latency on the switches
- Design critical choices and important considerations to have when building a ULL solution

2

Latency Considerations

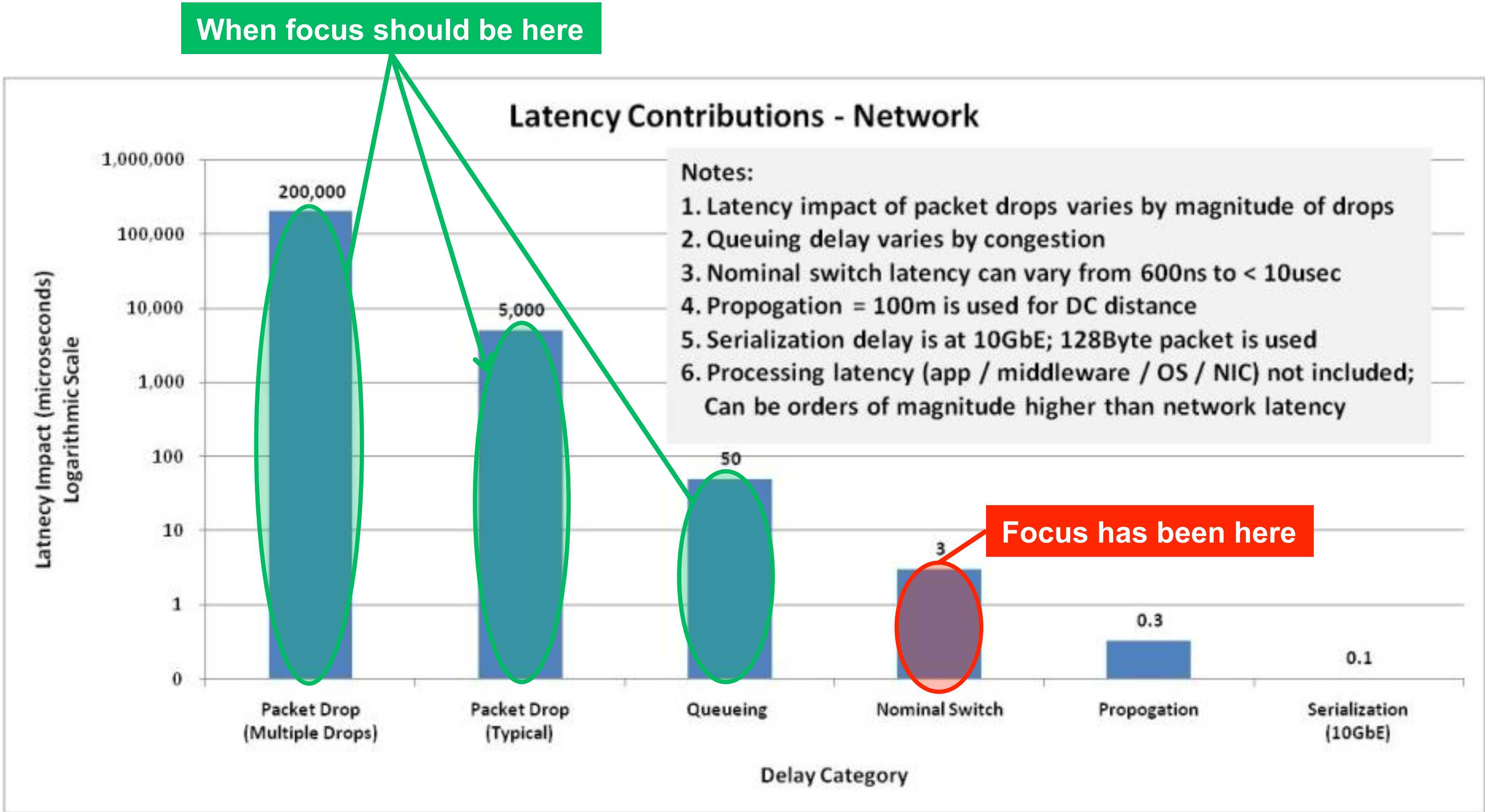
Latency

What is latency?

- Definition of latency: delay introduced in the communication between the time sender initiates it and the receiver receives and processes the information.
- Example: Voice Over IP, Radar, Satellite Communication, Real time application
- Different requirements / different user experience
  - Example of market data: user experience vs. machine trading
  - Examples of industry: Telecommunication vs. Financial

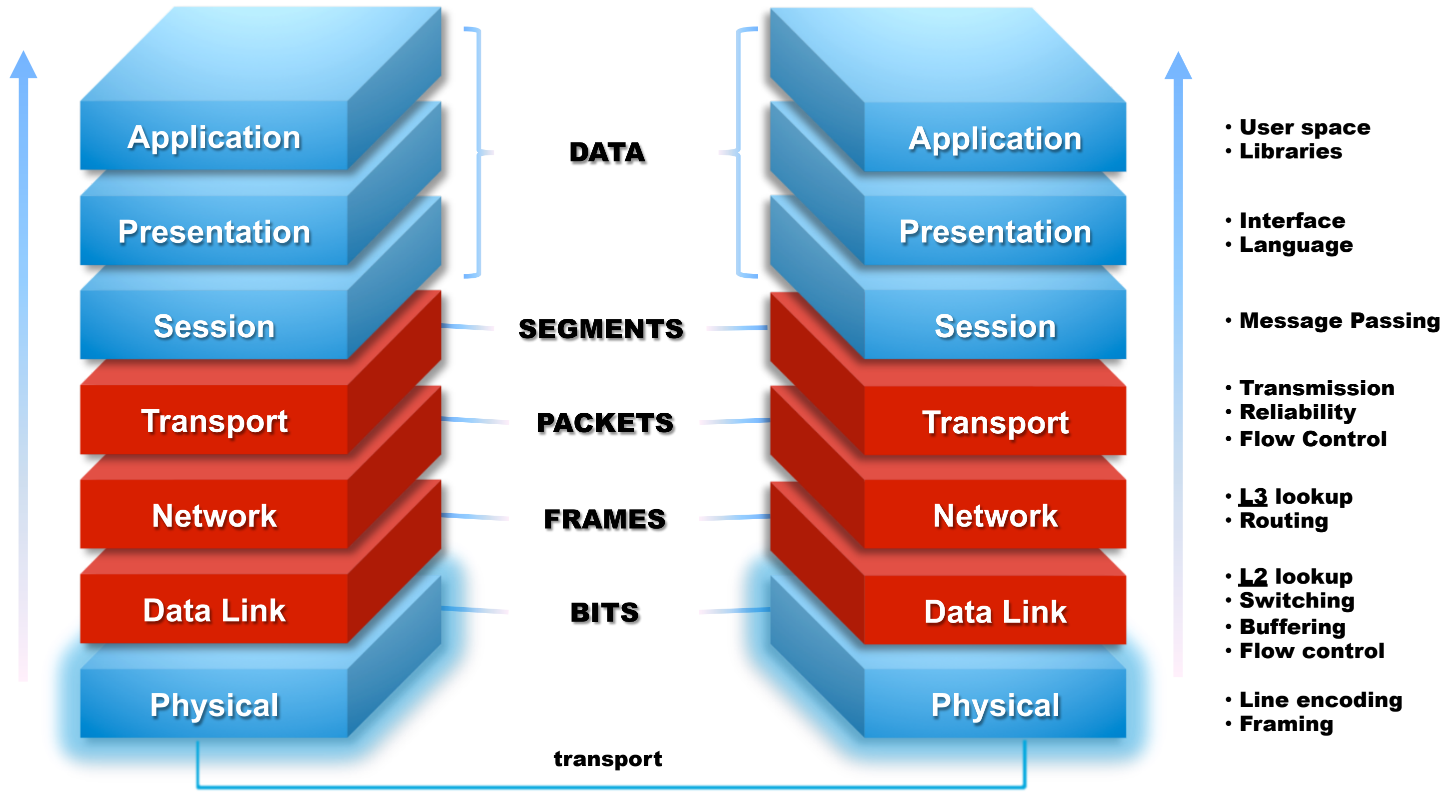
Consensus that performance without loss during stable and peek times is the ultimate goal

Network Latency Contributions by Category
(Y-Axis in Logarithmic Scale)



Latency

Which Latency where? Evolution of the look at the full stack

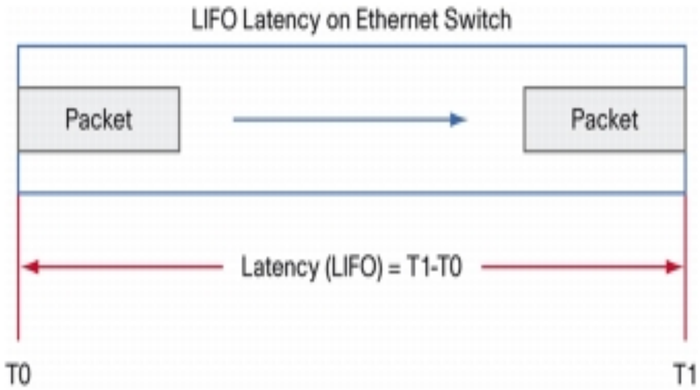


Latency

How to measure the latency in the Network?

- From RFC 1242: for store and forward devices:

The time interval starting when the last bit of the input frame reaches the input port and ending when the first bit of the output frame is seen on the output port.



LIFO: Last In First Out

Source: RFC 1242

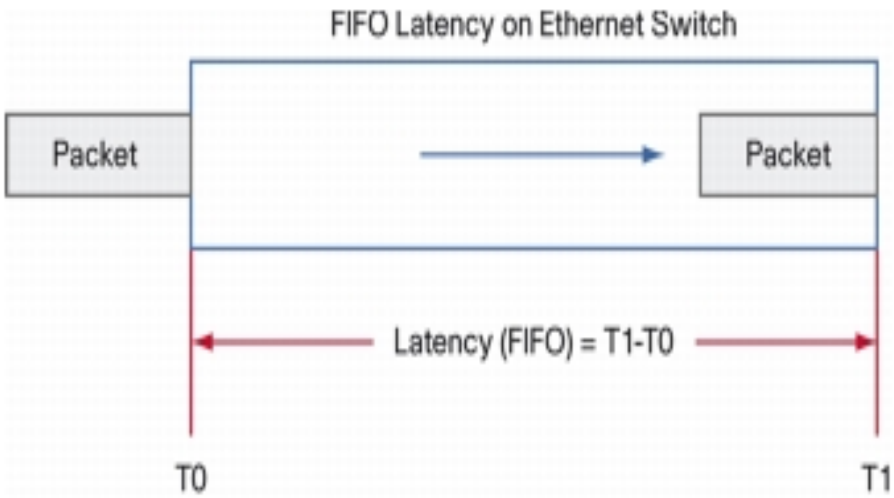
Latency

How to measure the latency in the Network?

- From RFC 1242: for bit forwarding devices *(cut-through devices)*:

The time interval starting when the end of the first bit of the input frame reaches the input port and ending when the start of the first bit of the output frame is seen on the output port

FIFO: First In First Out

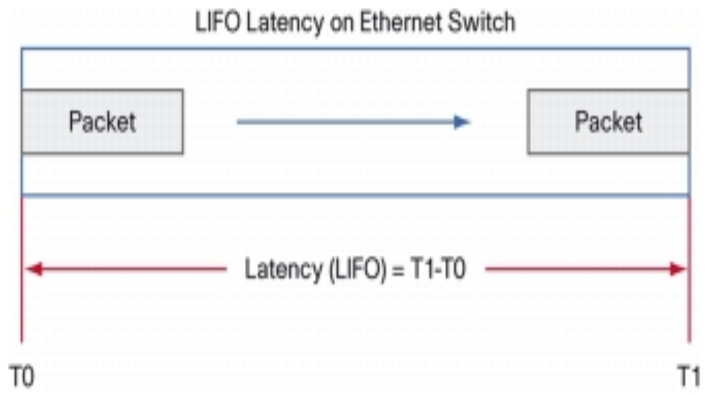


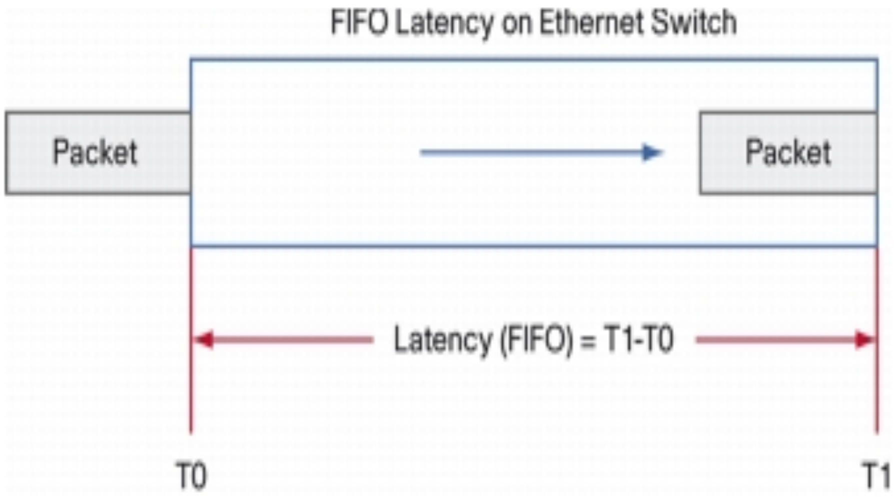
Source: RFC 1242

Latency

How to measure the latency in the Network?

- Measurement method: LIFO or FIFO?
- LIFO = FIFO - (Packet size in bits/ Speed)
- Cable length: identical cable type and length
- Identical amount of ports to test
- Identical testing equipment:
  - Chassis
  - Testing cards
  - Software Revision
- Typical Latency tests:RFC 2544, 2889, 3918

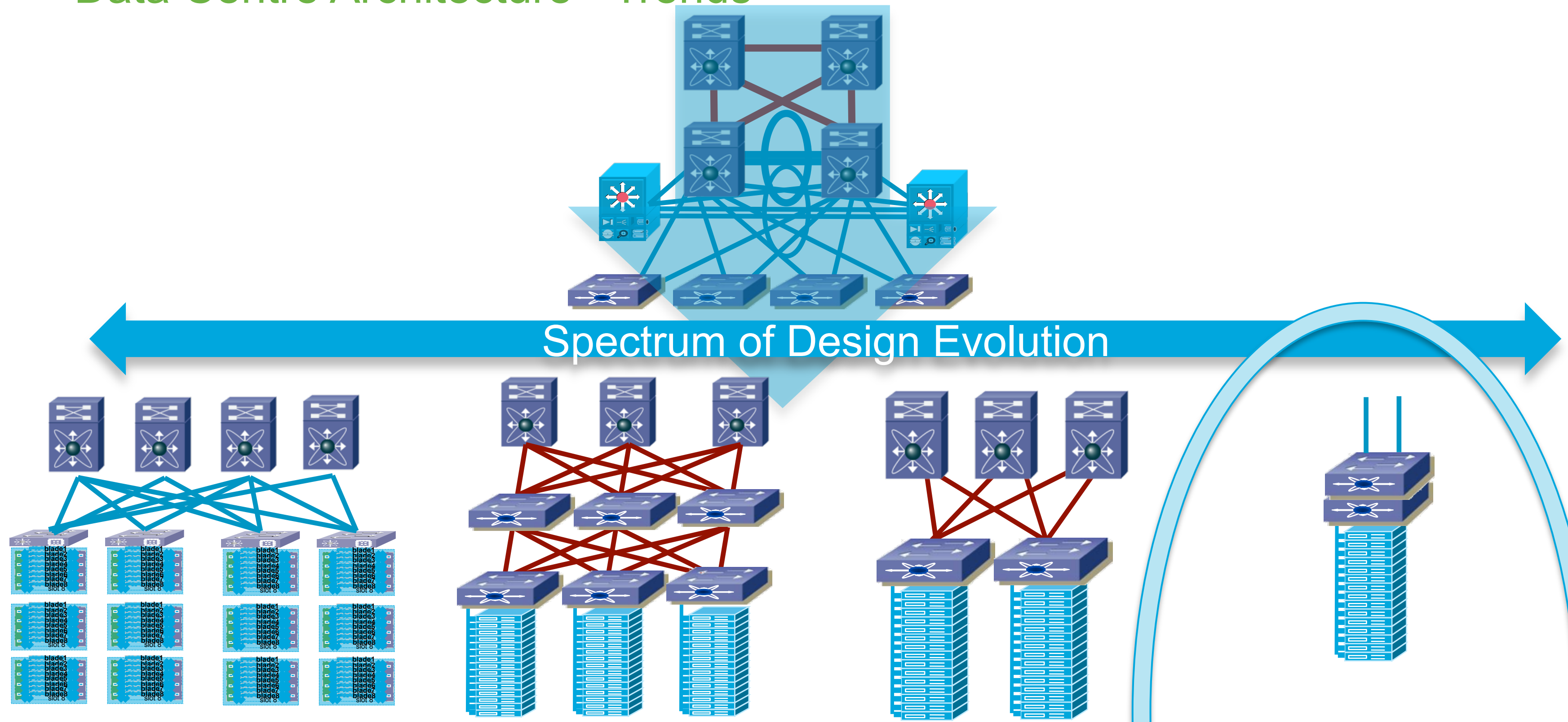




Design Considerations

Design Considerations

Data Centre Architecture - Trends



Virtualized Data Center

- SP and Enterprise
- Hypervisor Virtualization
- Shared infrastructure Heterogeneous
- 1G Edge moving to 10G
- Nexus 1000v, 2000, 5500, 7000 & UCS

Warehouse Scale

- Layer 3 Edge (iBGP, ISIS)
- 1000's of racks
- Homogeneous Environment
- No Hypervisor virtualization
- 1G edge moving to 10G
- Nexus 2000, 3000, 5500, 7000 & UCS

HPC/GRID

- Layer 3 & Layer 2
- No Virtualization
- iWARP & RoCE
- Nexus 2000, 3000, 5500, 7000 & UCS
- 10G moving to 40G

Ultra Low Latency

- High Frequency Trading
- Layer 3 & Multicast
- No Virtualization
- Limited Physical Scale
- Nexus 3000 & UCS
- 10G edge moving to 40G

Design Considerations

Design Consideration #1 : Speed

- Baud rate
- The driver is not bandwidth in ULL

1 G

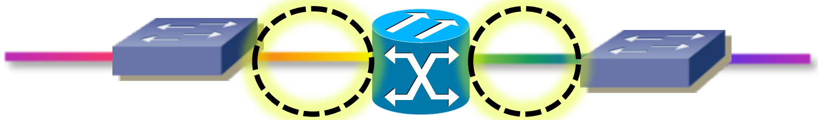
10 G

40 G

The Faster speed should be the faster communication, is that all?

Design Considerations

Design Consideration #1 : Speed



10, 40, 100 GE options to reduce serialization delay

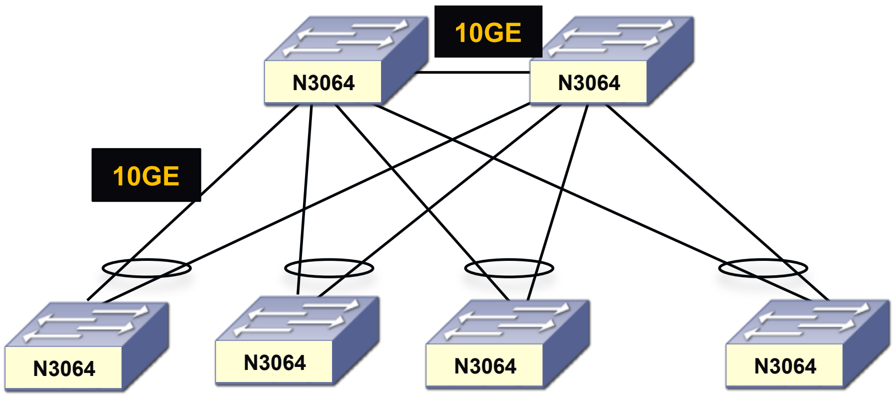
	1 GE	10 GE	40 GE	100 GE
64 byte	0.512 us	0.051us	0.013us	0.005us
128 bytes	1.024 us	0.102us	0.026us	0.010us
256 bytes	2.048 us	0.205us	0.051us	0.021us
512 bytes	4.096 us	0.410us	0.102us	0.041us

- Serialization Delay reduced with higher speeds
- The less speed mismatch the better performance

Design Considerations

Design Consideration #1 : Speed - Performance and Serialization delay

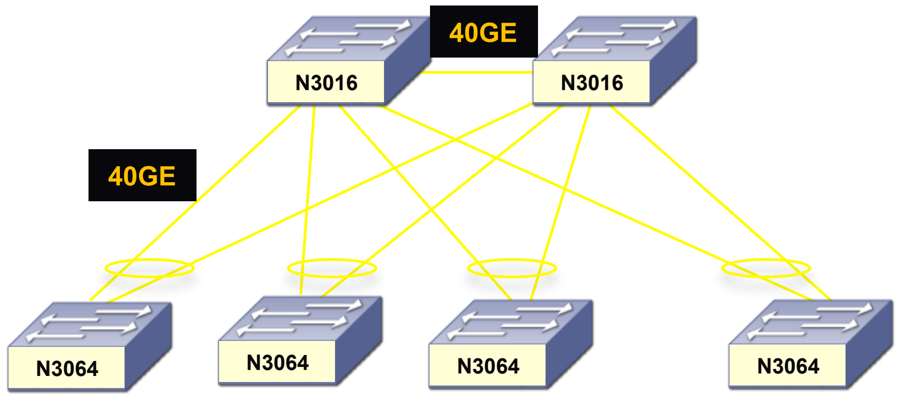
10GE Interconnects



10GE Hosts

128 bytes = 2.802us
258 bytes = 2.907us
512 bytes = 3.531us

40GE Interconnects

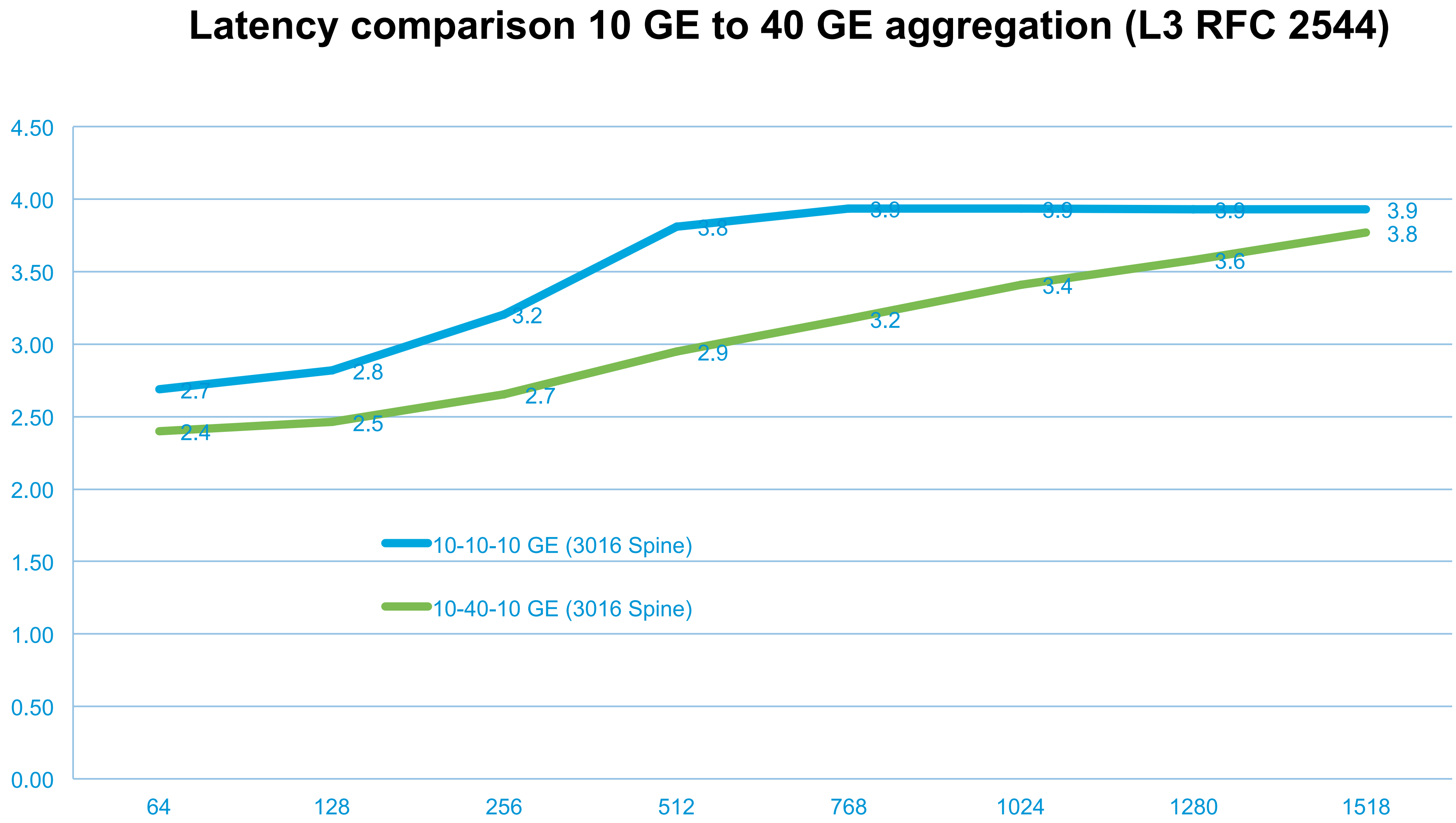


10GE Hosts

128 bytes = 2.45us
258 bytes = 2.65us
512 bytes = 2.951us

128 bytes = 0.352us
258 bytes = 0.257us
512 bytes = 0.580us

Comparison between 10GE and 40 GE aggregation – Layer 3



Ex: Up to 860 nanoseconds faster with 40 GE interconnects to an aggregation N3016

Design Considerations

Design Consideration #2 – Congestion– What is the traffic type?

- Small Flows/Messaging

(Heart-beats, Keep-alive, delay sensitive application messaging)

- Small – Medium Incast

(Hadoop Shuffle, Scatter-Gather, Distributed Storage)

- Large Flows

(HDFS Insert, File Copy)

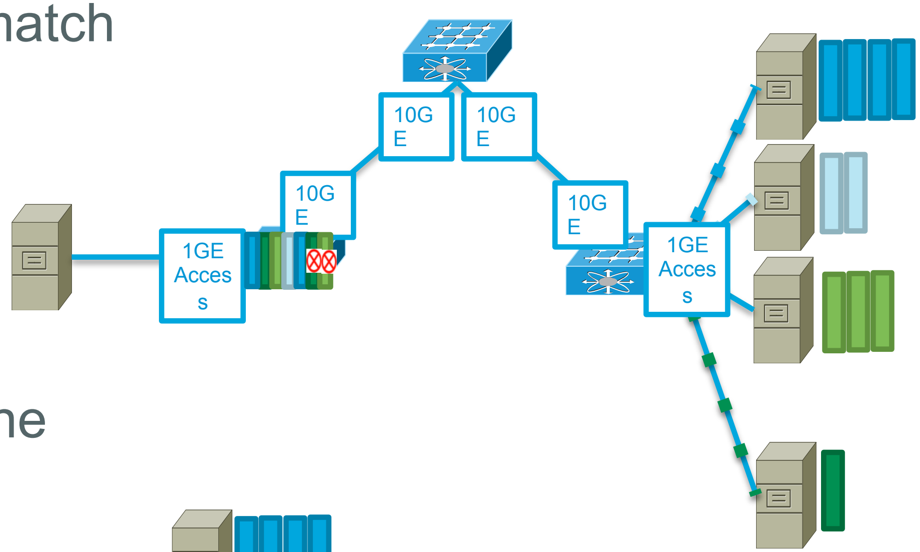
- Large Incast

(Hadoop Replication, Distributed Storage)

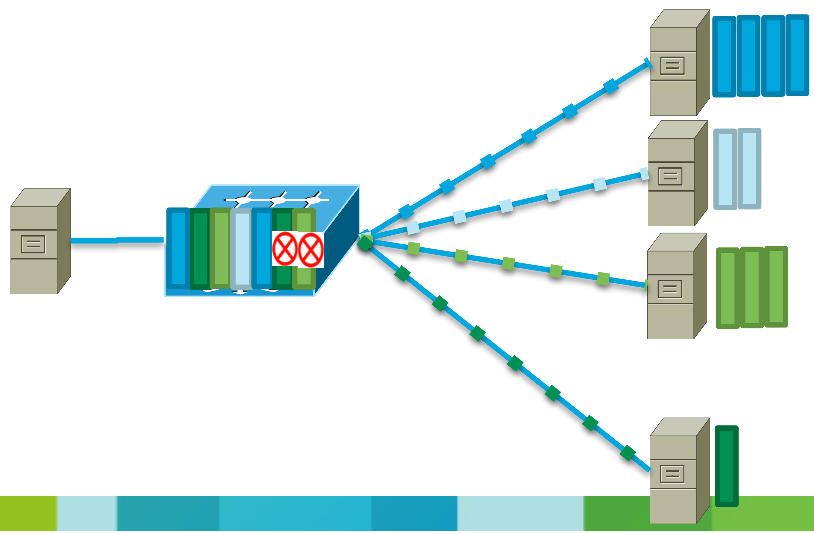
Design Considerations

Design Consideration #2 – Congestion – When are buffers needed?

- Uplink Speed Mismatch

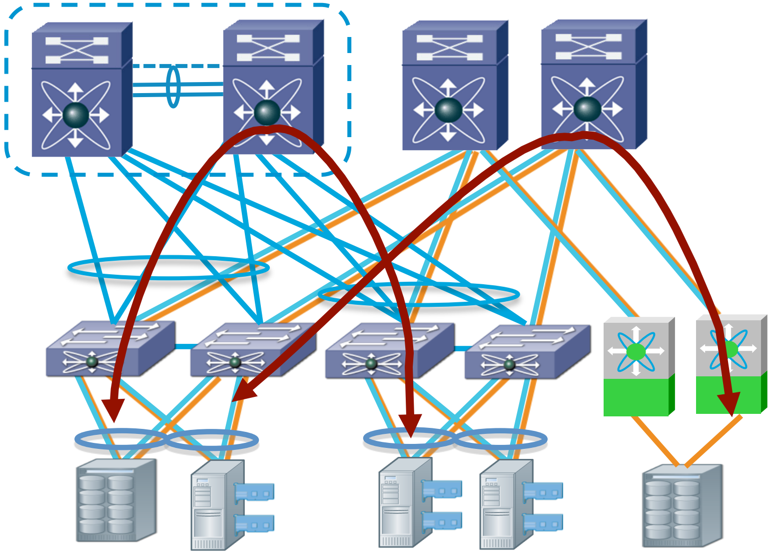


- Incast / Many to One conversations



Design Considerations

Design Consideration #2 – Congestion – When are buffers needed?

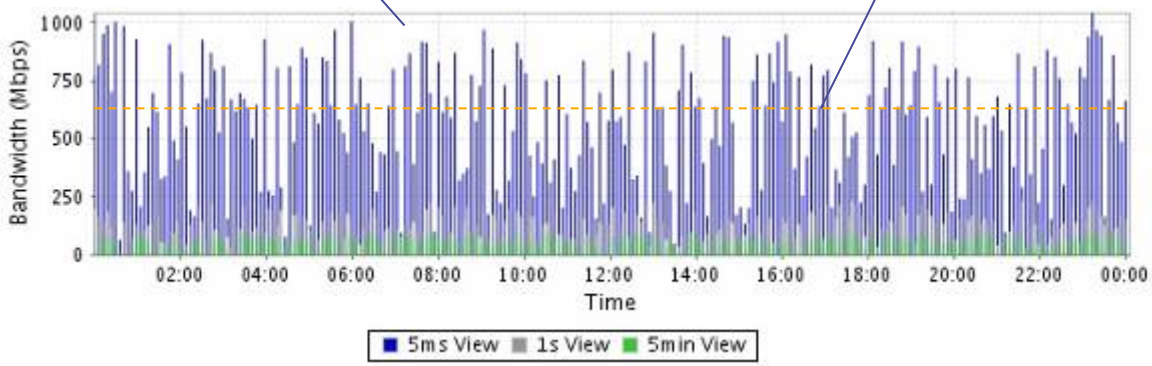


- A balanced fabric is a function of maximal throughput 'and' minimal loss => "Goodput"
- Application-level throughput (goodput): Given by the total bytes received from all senders divided by the finishing time of the last sender.

Source : "*Understanding TCP Incast Throughput Collapse in Datacenter Networks*", Y. Chen, R Griffith, WREN '09

5 millisecond view

Congestion Threshold exceeded

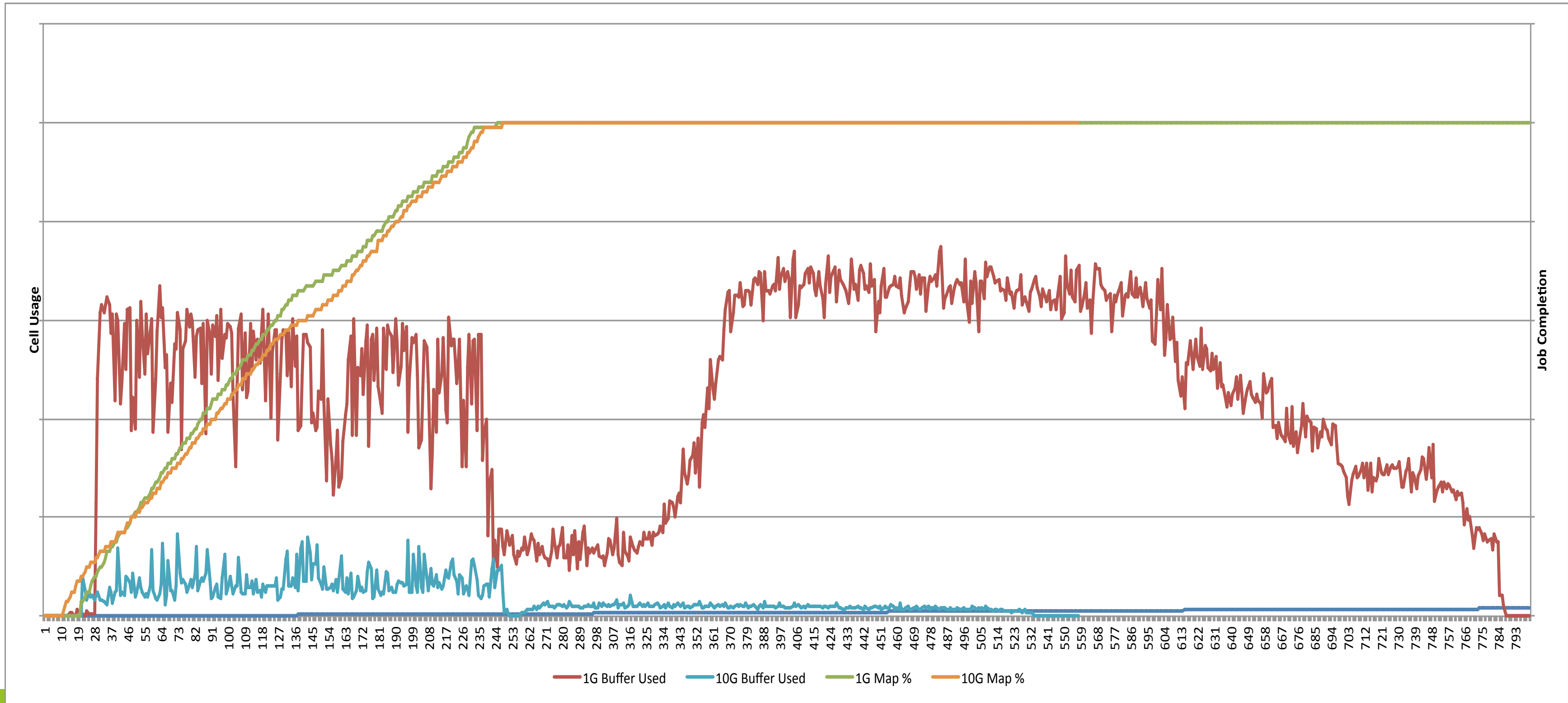


Data Center Design Goal: Optimizing the balance of end to end fabric latency with the ability to absorb traffic peaks and prevent any associated traffic loss

Buffering and link speeds

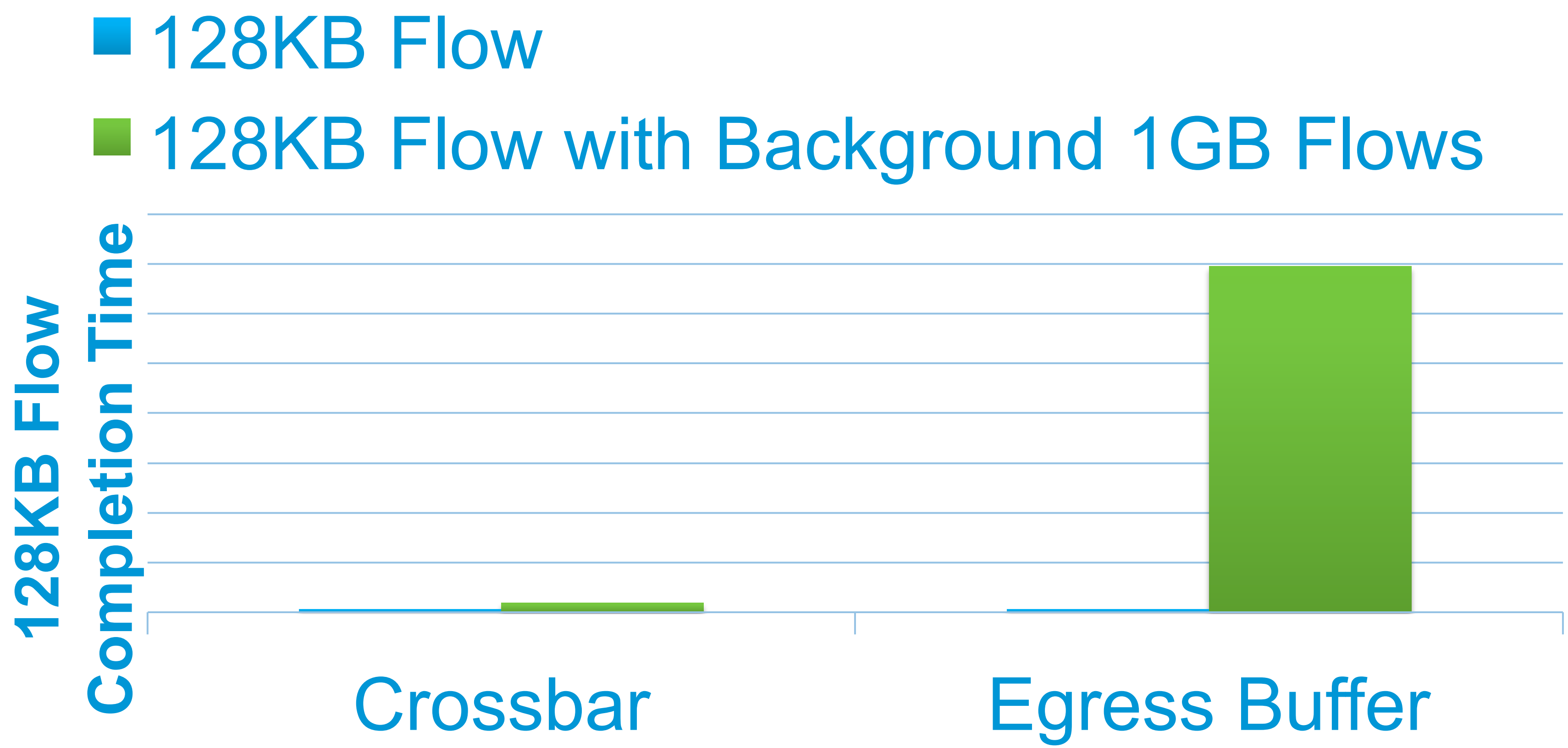
Incast

- Moving from 1GE to 10GE actually lowers the buffer requirement on the switching layer
- By moving to 10GE, the data node has a larger input to receive data lessening the need for buffers on the network as the total aggregate speed or amount of data does not increase substantially
- Current system limits are primarily I/O and Compute capabilities (Disk I/O bound)



Design Considerations

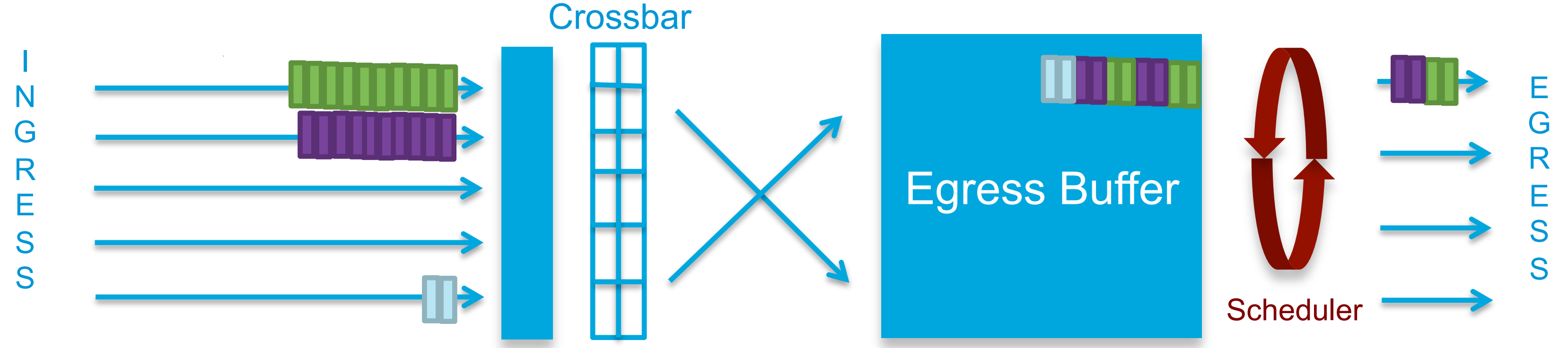
Design Consideration #2 – Buffer Amount – Small Flow Buffer Delay

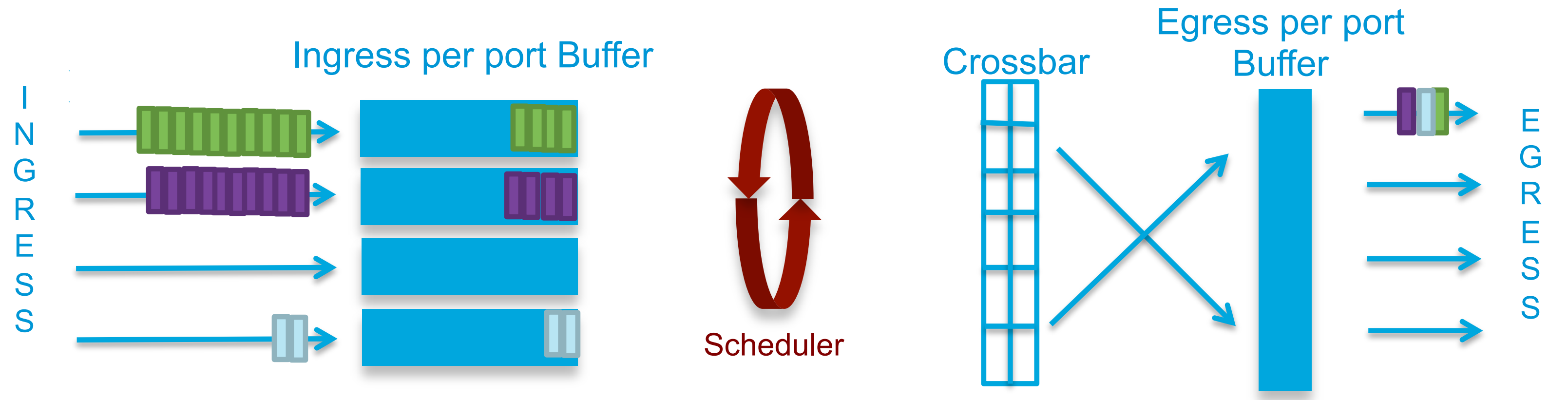


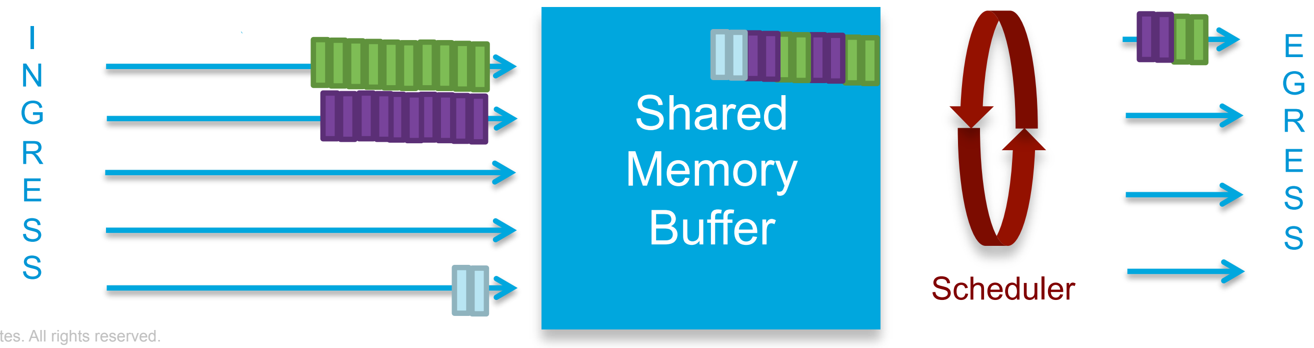
Buffer Architecture Choice Matters

Design Considerations

Design Consideration #2 – Buffer Amount – The Switch Architecture

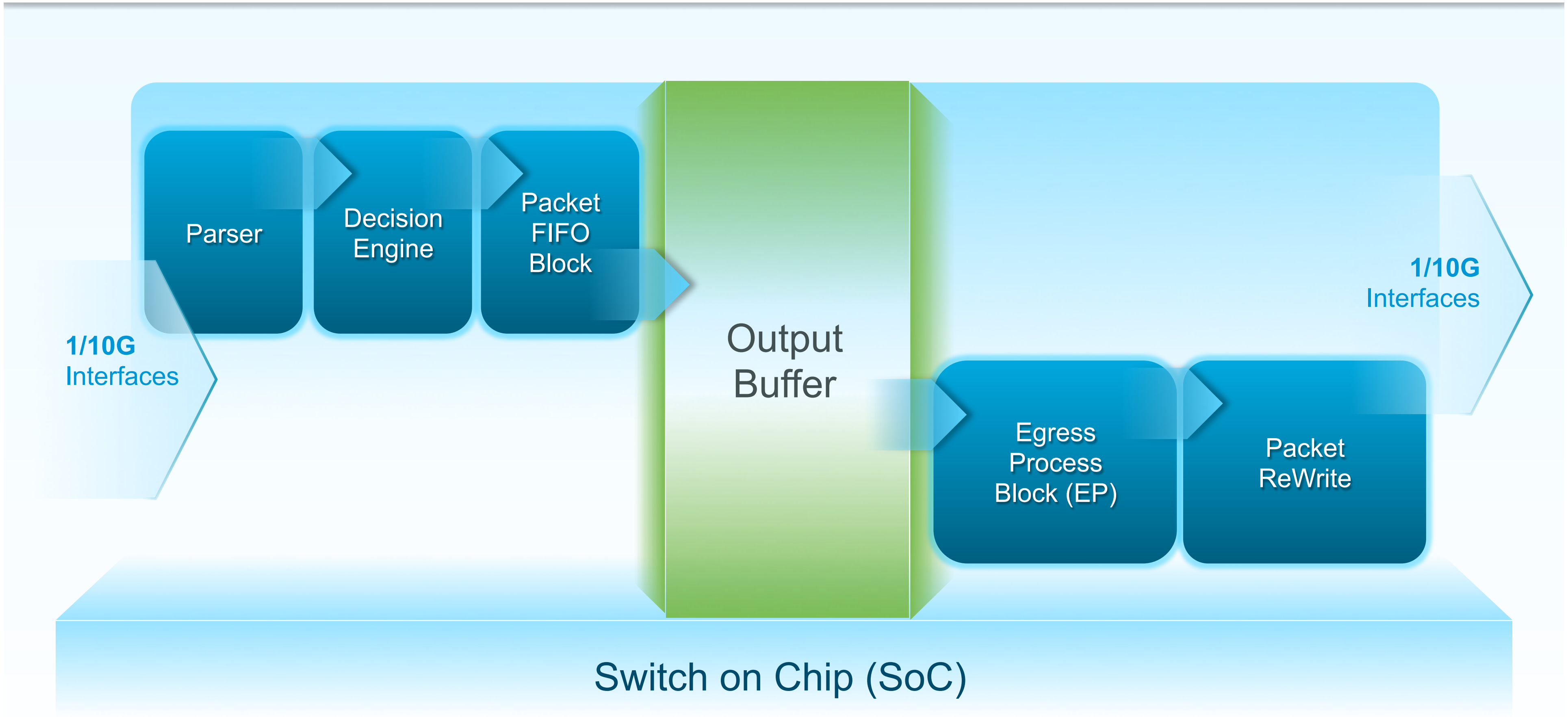






Design Considerations

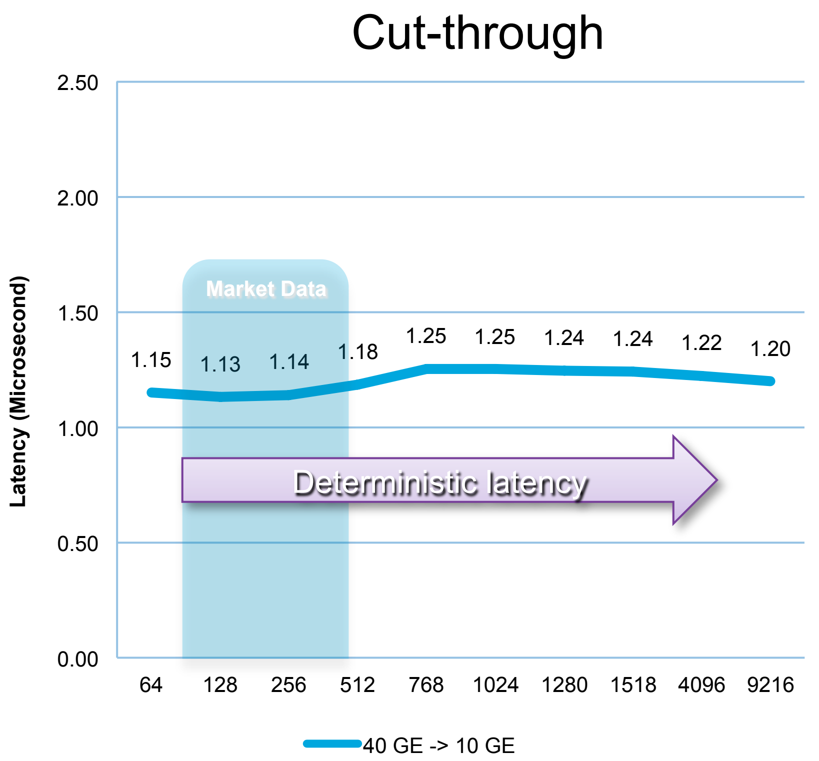
Design Consideration #2 – Buffer Amount – The Switch Architecture

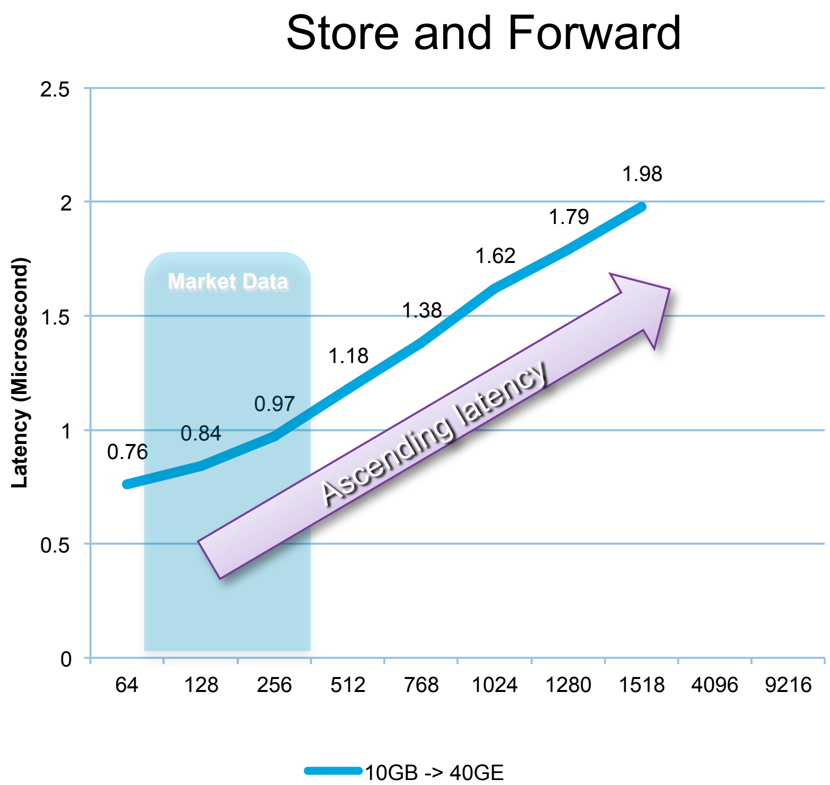


Design Considerations

Design Consideration #3 – Switching mode

- Deterministic vs. Lowest debate





RFC 2544 measuring Latency for different switch architectures

Design Considerations

Design Consideration #4 : Physical Media Type – 10G and 40G

- What media to choose, optical or copper?

Propagation Delay Pd

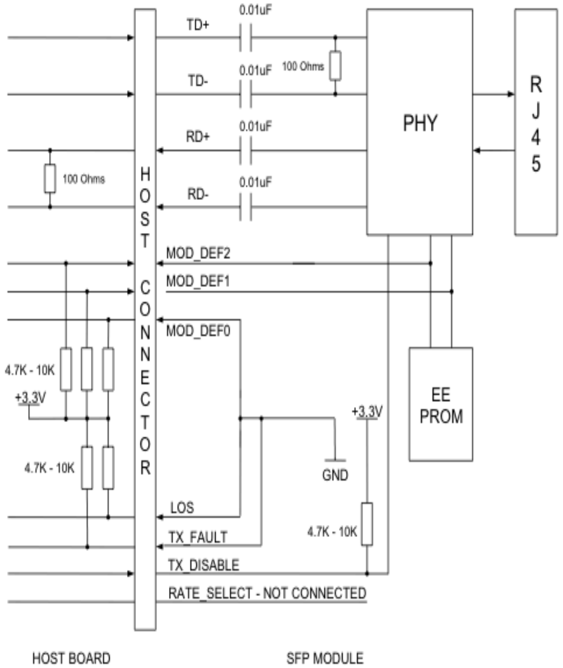
Delay for 1m	Fiber	CX-1	RJ-45
Pd (ns)	5ns	4.3ns	5ns

- Propagation delay Pd= distance / speed
- Electromagnetic speed: s=200 000 km/s
- Light Speed: 300 000 km/s, fiber glass refraction 1.5

Design Considerations

Design Consideration #4 : Physical Media Type – 10G and 40G

- Copper UTP/RJ-45: +**1usec** in the conversion Base SX <-> Base T



Best Practice: Optical for 1GE, passive CX-1 or optical for 10/40GE

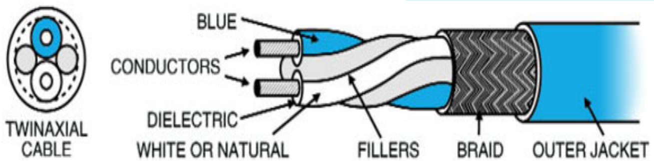
25

Design Considerations

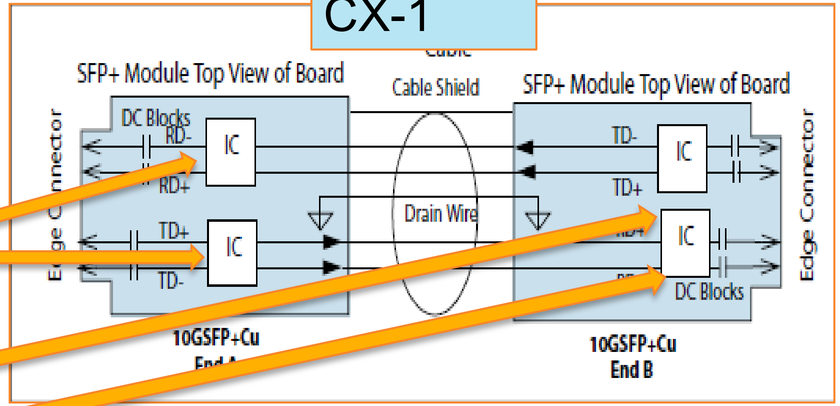
Design Consideration #4 : Physical Media Type

- Active or Passive Twinax?

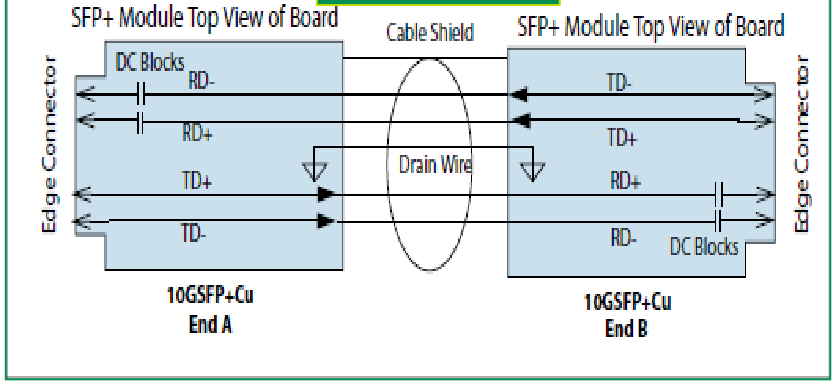
Active process electrical signaling, needs power to drive the IC.
Behave as an optical SFP transceiver



Active CX-1



Passive CX-1



- Passive CX-1 is **0.3ns** latency
- Active CX-1 adds **1ns** latency

Design Considerations

Design Consideration #5 – Feature Set

- CDP / LLDP
- STP
- Layer 3
- Multicast
- Multiple Destination SPAN / Span ACL
- NAT
- Monitoring

Design Considerations

Consideration #5 – Feature Set - Using Python to Buffer Monitoring

- Runs directly from NX-OS CLI
- Pass Arguments
- Displays the scripts on CLI

Interactive Python Shell

```
switch# python
Python 2.7.2 (default, Jan 11 2012, 17:25:37)
[GCC 3.4.3 (MontaVista 3.4.3-25.0.143.0800417 2008-02-22)] on
linux2
Type "help", "copyright", "credits" or "license" for more information.
Loaded cisco NxOS lib!
>>>
```

```
switch# python script.py arg1 arg2
['/bootflash/test.py', 'arg1', 'arg2']
```

Run Python Script

```
switch# python bootflash:showBuffer.py
Mon Jan 30 19:26:36 UTC 2012


|---------------------------------------------------------------------------------------|

                Total Instant Usage            0
                Remaining Instant Usage        46080
                Max Cell Usage                 0
                Switch Cell Count              46080
|---------------------------------------------------------------------------------------|

#
```

Code for the Python script showBuffer.py is in the next slide ….

```
switch# show file bootflash:script.py
```

Design Considerations

Consideration #5 – Feature Set - Using Python to Buffer Monitoring

showBuffer.py

```
#!/usr/bin/python

# Import the CLI class
from cisco import *

# Create Object objCli
objCli = CLI('show hardware internal buffer info pkt-stats brief', False)

# Display raw output:
print objCli.get_raw_output()
```

- Import Cisco library to use the CLI class in your script
- CLI class provides optimized access to the NxOS CLI
- Second parameter to CLI specifies whether output should be printed. Default(True) is set to print.
- Provides raw (unchanged) output as printed by the NxOS CLI

CLI: get_raw_output()

```
|------------------------------------------------------------------------------------|

                        Total Instant Usage             0
                        Remaining Instant Usage     46080
                        Max Cell Usage                  0
                        Switch Cell Count           46080
|-----------------------------------------------------------------------------------|
```

Design Considerations

- Consideration #5 – Feature Set - Using Python to Enhance Monitoring

Hadoop Job Status

```
12/03/27 08:02:23 INFO mapred.JobClient:  map 69% reduce 0%
12/03/27 08:02:24 INFO mapred.JobClient:  map 77% reduce 0%
12/03/27 08:02:25 INFO mapred.JobClient:  map 87% reduce 0%
12/03/27 08:02:26 INFO mapred.JobClient:  map 96% reduce 9%
12/03/27 08:02:27 INFO mapred.JobClient:  map 98% reduce 10%
12/03/27 08:02:28 INFO mapred.JobClient:  map 100% reduce 10%
12/03/27 08:02:29 INFO mapred.JobClient:  map 100% reduce 27%
12/03/27 08:02:30 INFO mapred.JobClient:  map 100% reduce 29%
12/03/27 08:02:32 INFO mapred.JobClient:  map 100% reduce 32%
12/03/27 08:02:35 INFO mapred.JobClient:  map 100% reduce 84%
```

Hadoop job status output while running a 1GB TeraSort using 8 nodes

Buffer Usage

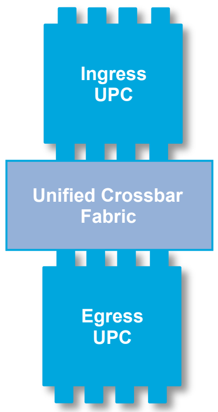
```
2012/03/27 08:02:23     0      *
2012/03/27 08:02:24     3810   -----*
2012/03/27 08:02:25     1127   -*
2012/03/27 08:02:26     0      *
2012/03/27 08:02:27     0      *
2012/03/27 08:02:28     0      *
2012/03/27 08:02:29     0      *
2012/03/27 08:02:30     0      *
2012/03/27 08:02:31     0      *
2012/03/27 08:02:32     4921   -------*
2012/03/27 08:02:33     4299   ------*
2012/03/27 08:02:34     6929   ----------*
2012/03/27 08:02:35     0      *
```

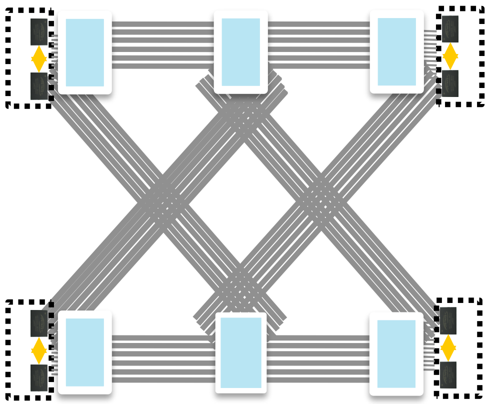
Buffer usage statistics from the switch while running Hadoop TeraSort

Design Considerations

Design Consideration #6 – Simplicity of the Network design – All in one Approach

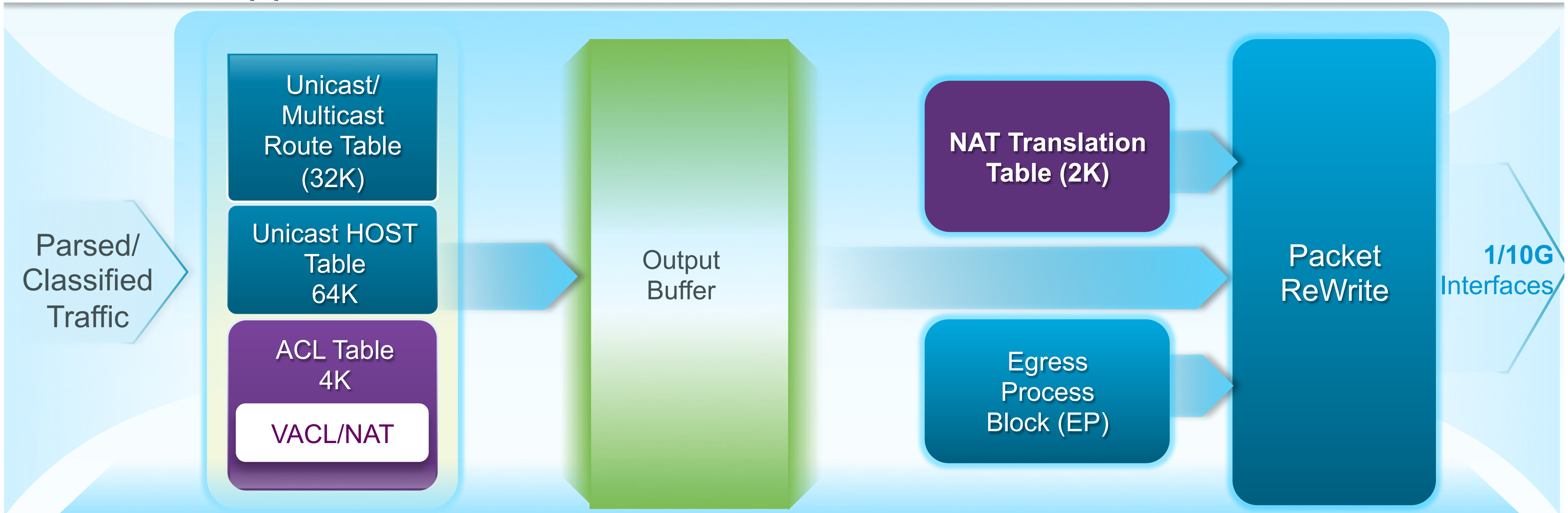
- What is the total port count needed?
- What is the end design / scale targeted?
- Is communication needed between servers, inside / outside POD
- Uniform speed or higher uplink with cut-through switching?
- What is the feature-set required?





Design Considerations

- Design Consideration #6 – Simplicity of the Network design – All in one Approach



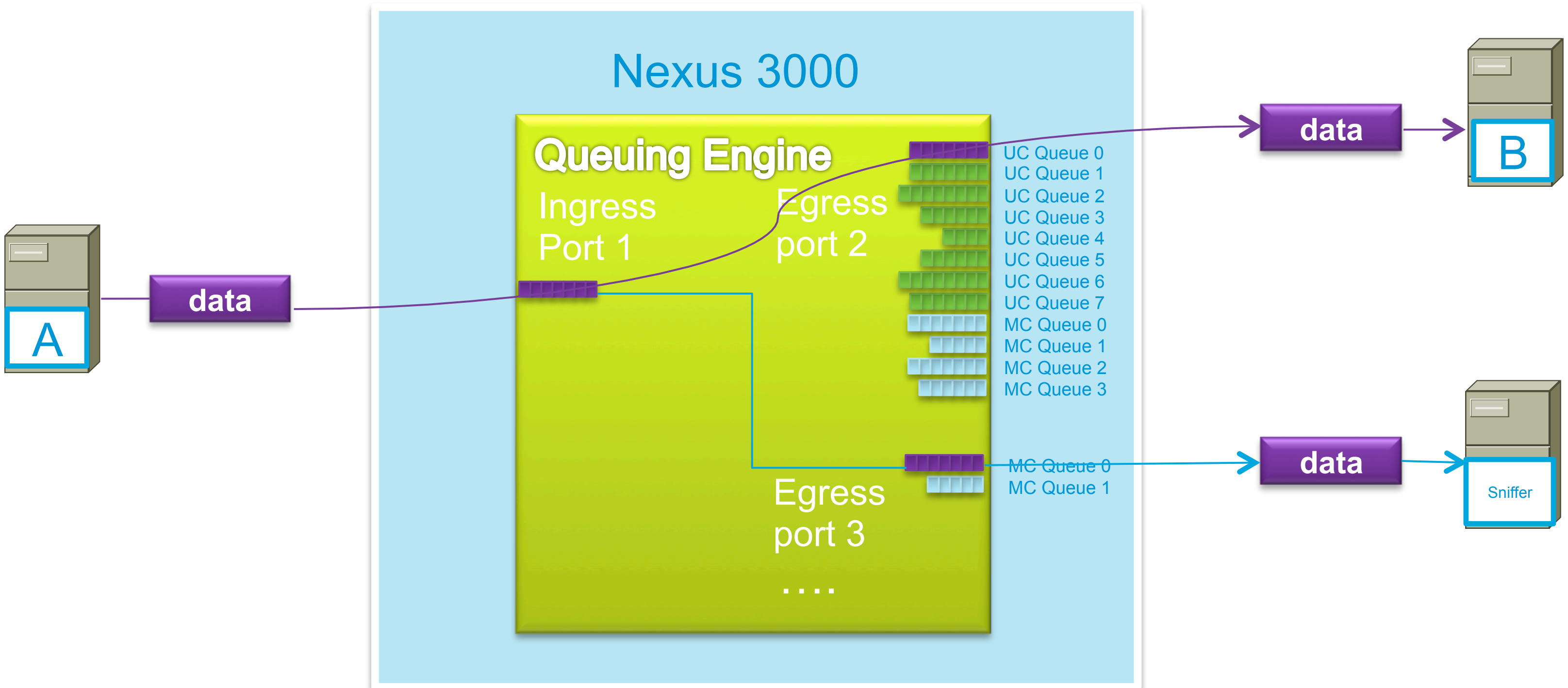
NAT/PAT Classification and Translation

- NAT uses VACL space for classifying and identifying the traffic for NAT translation based on ingress interface
- NAT translation table would provide actual translation info for packet ReWrite block for packet modification before sending the packet out of NAT interface
- For Static NAT, ACL and Translation Table are updated as soon as the NAT static config is added
- For dynamic NAT*, first packet is punted to CPU after ACL classifies it to be NAT flow and then software updates the translation table based on the flow info

* Post FCS

Design Considerations

Design Consideration #6 – Feature set - Linerate SPAN

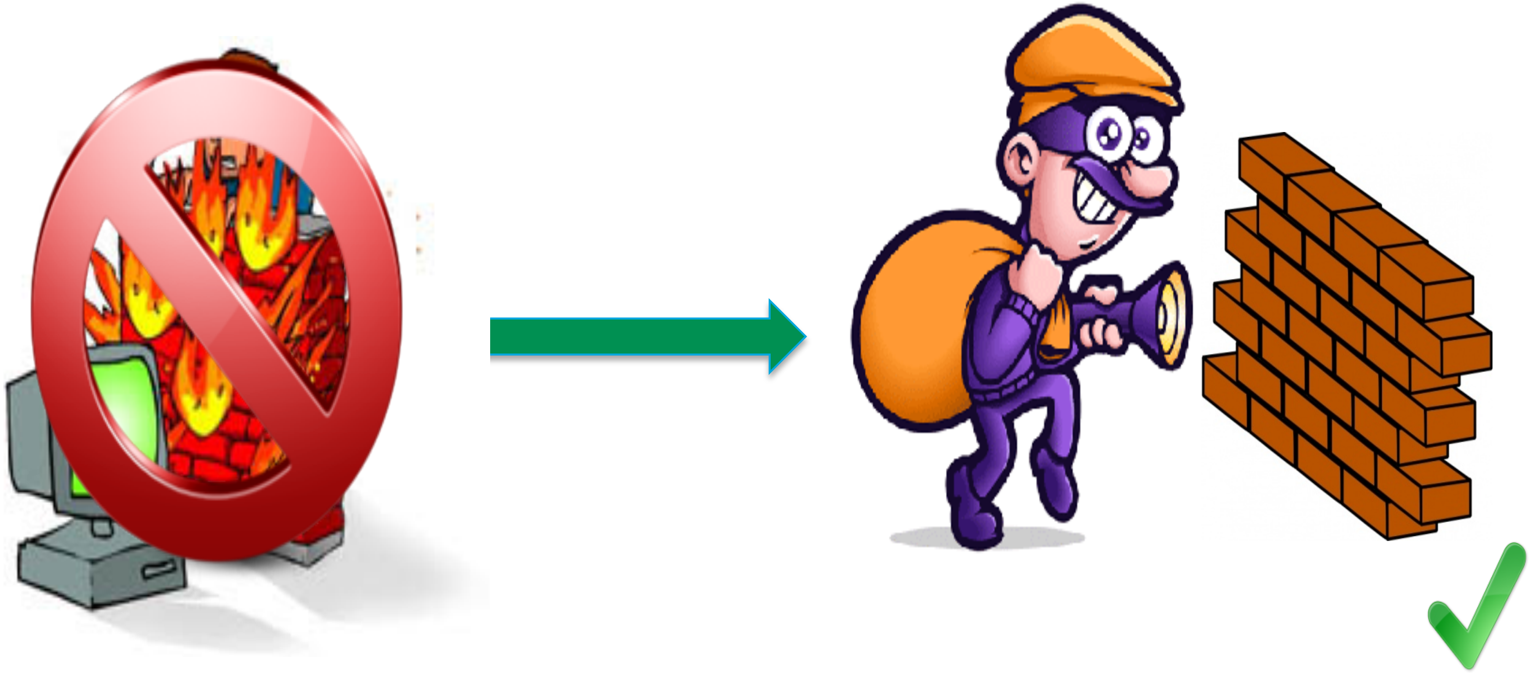


- Traffic to be replicated is marked in the ingress flow
- The replication occurs in the queuing engine and the mirrored traffic is placed in one of two multicast queues

Design Considerations

Design Consideration #7 – Security

- Use Hardware features: ACLs, PVLANs…
- Use OS level security when possible



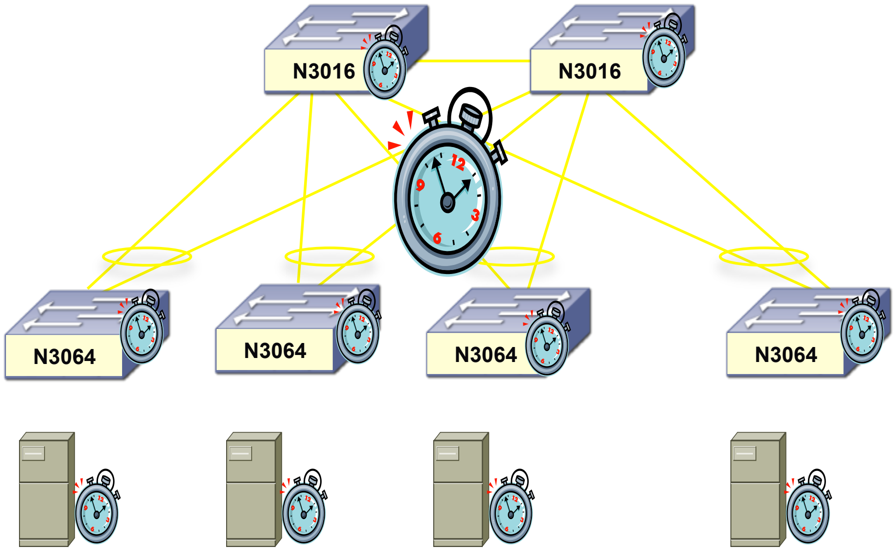
Design Considerations

Design Consideration #8 – Application Precision

- Precision Time Protocol: IEEE 1588v2
- Nanosecond Precision

Telecommunications


Financial trading

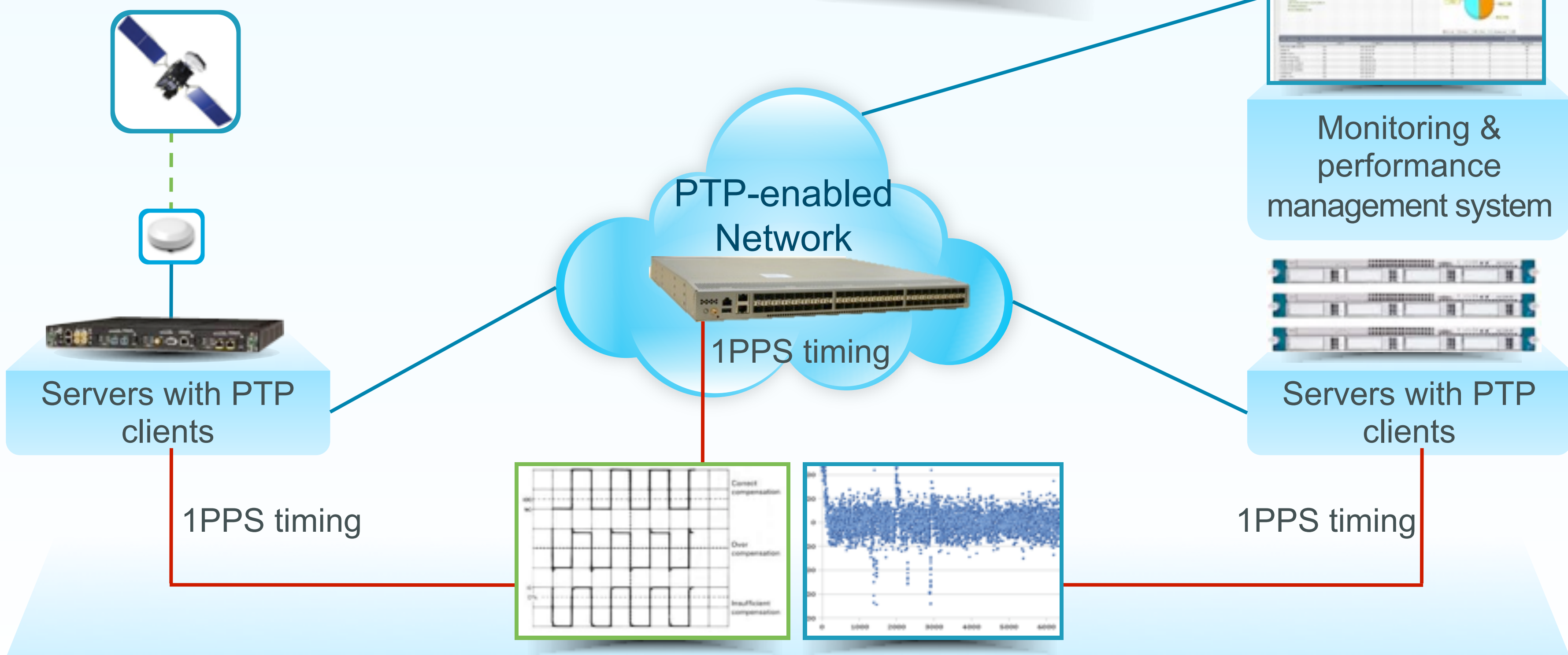


Design Considerations

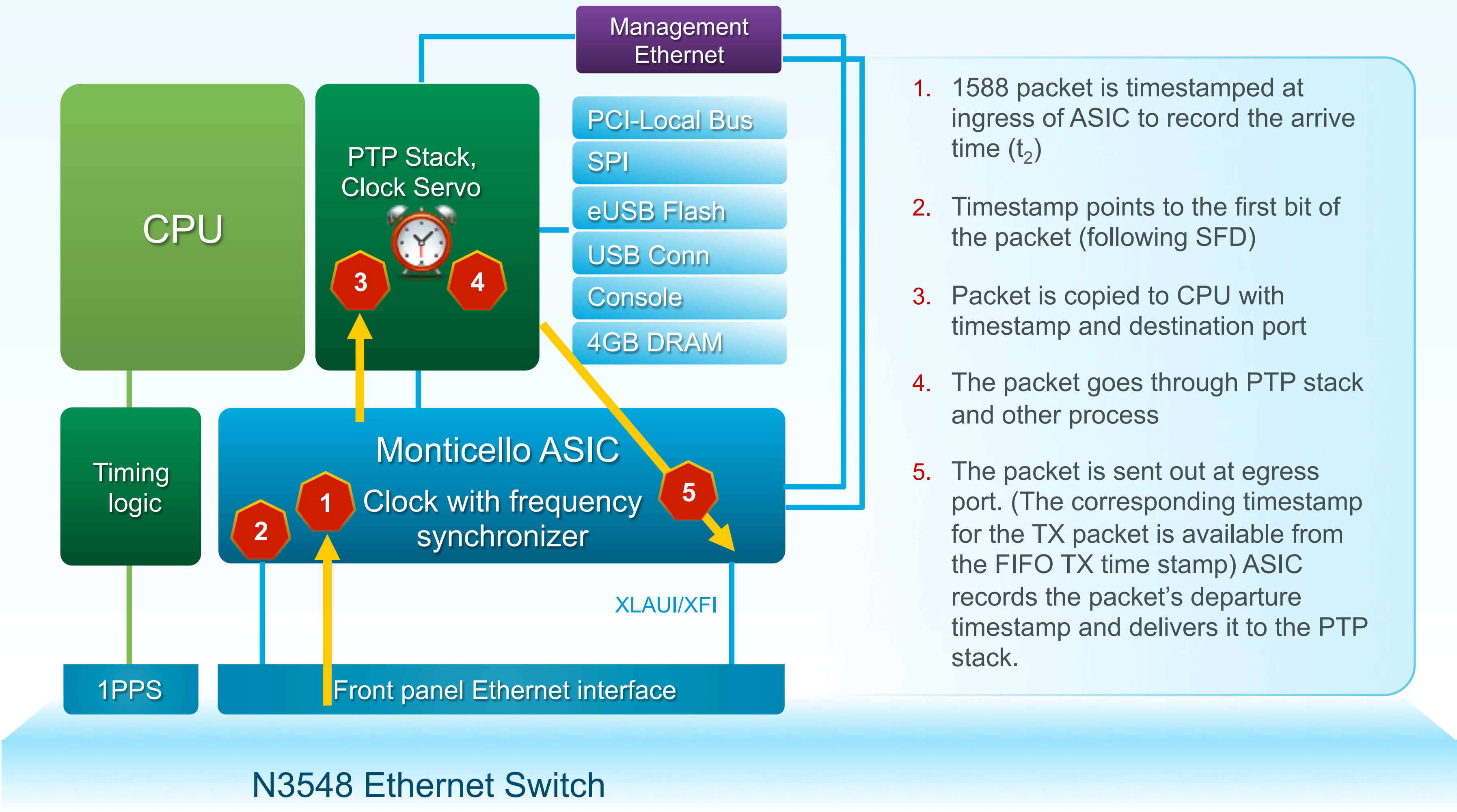
- Design Consideration #8 – Application Precision

Applications @ Switch

- Verify accuracy with 1PPS output
- PONG for Hop-by-Hop Latency Measurements
- Integration with ERSPAN for Accurate Timestamp of Monitored Traffic



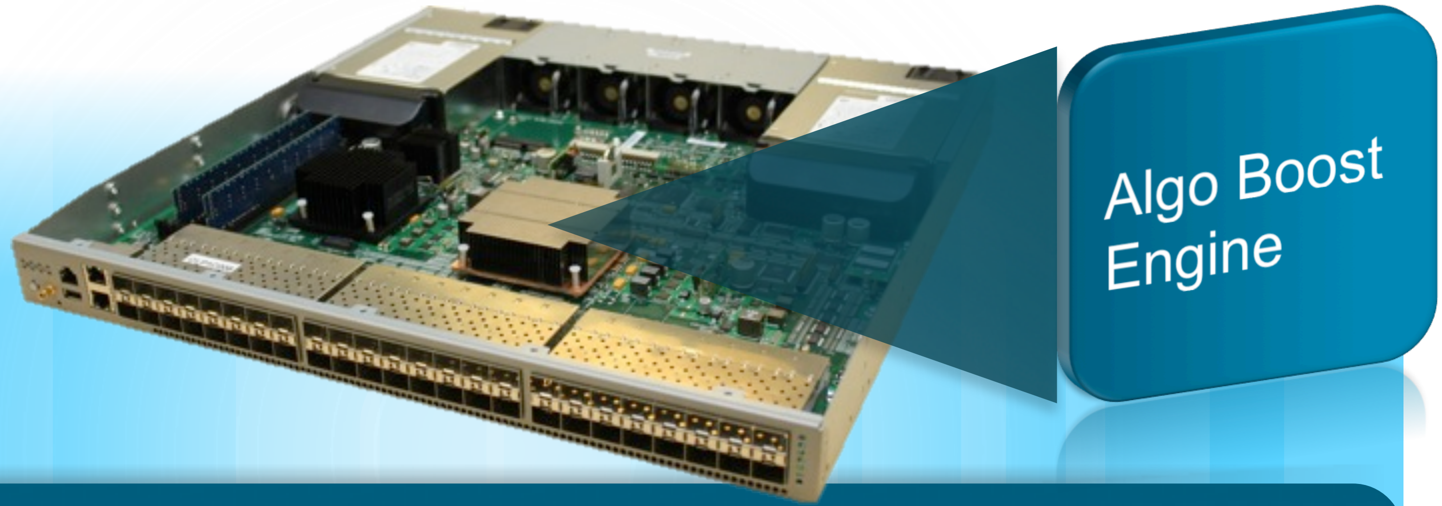
IEEE 1588 Implementation



1. 1588 packet is timestamped at ingress of ASIC to record the arrive time (t_2)
2. Timestamp points to the first bit of the packet (following SFD)
3. Packet is copied to CPU with timestamp and destination port
4. The packet goes through PTP stack and other process
5. The packet is sent out at egress port. (The corresponding timestamp for the TX packet is available from the FIFO TX time stamp) ASIC records the packet's departure timestamp and delivers it to the PTP stack.

New Landscape of ULL

It’s All about ASICS



Typical Specifications

- 48x SFP+ – 100M / 1G / 10G / 40G
- Line rate L2/L3, Unicast & Multicast
- 18MB Packet Buffer
- 32K IPv4 Route, 64K Host, 8K MC
- 4K Flexible ACL / QoS
- Data Center TCP

Algorithm Boost Features

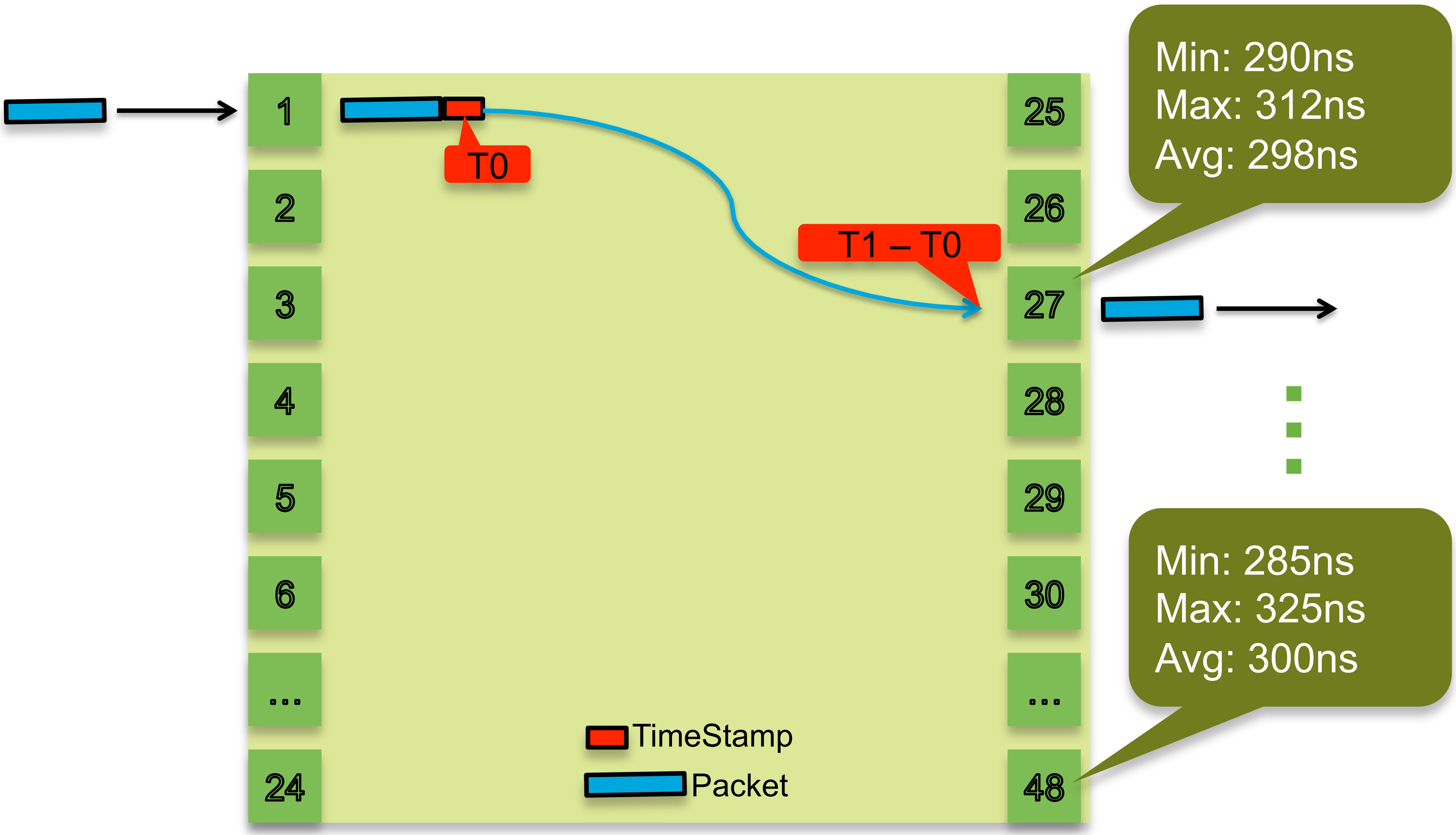
- Ultra Low Latency – <300ns
- Active Latency/Buffer Monitoring
- NAT @ Ultra Low Latency
- Intelligent Traffic Mirroring
- IEEE-1588 PTP w/Pulse Per Second

Latency

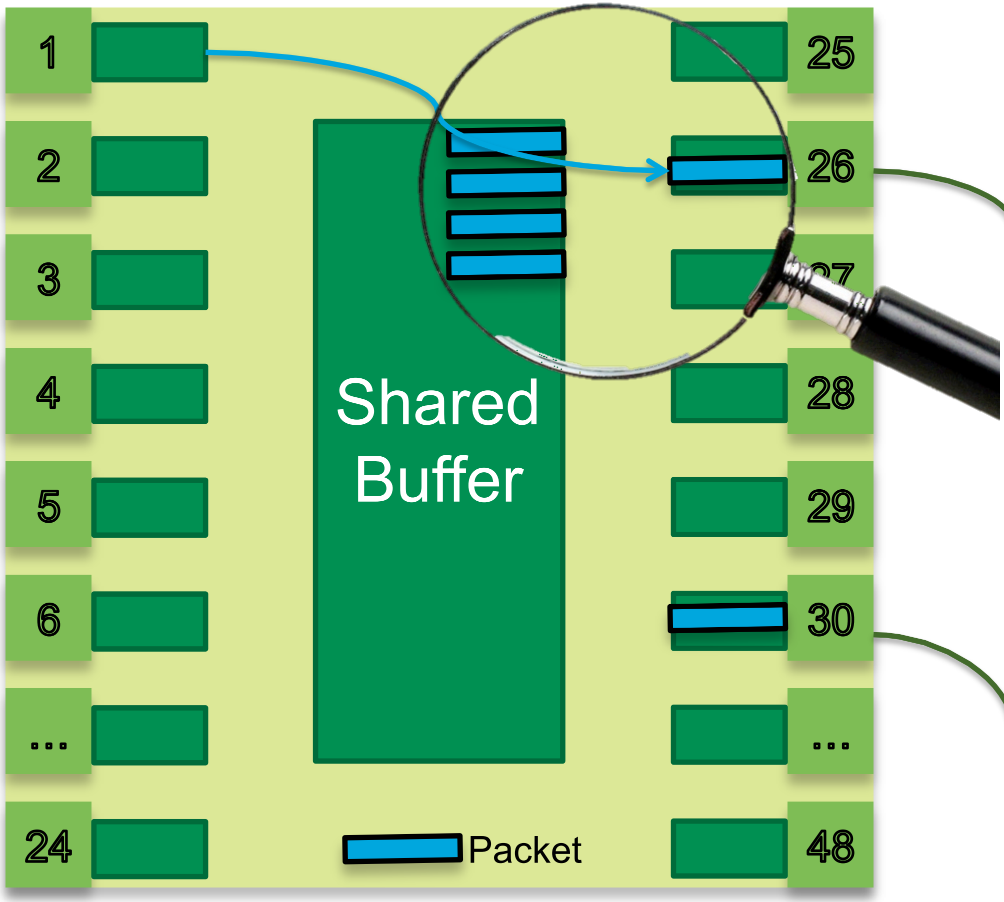
< 300 ns

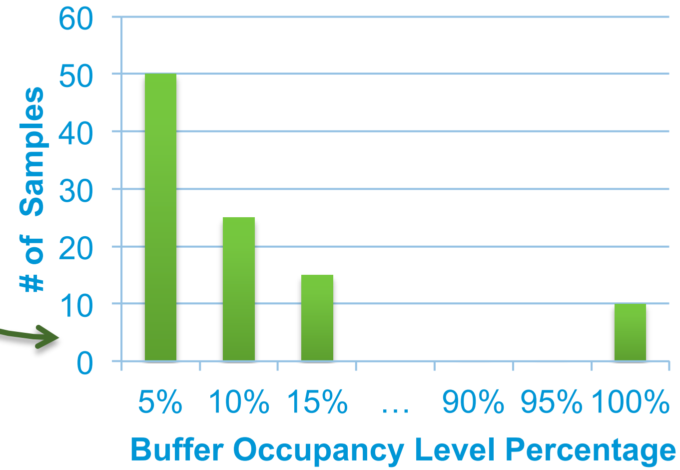
- L2 & L3, Unicast & Multicast
- Independent of packet size
- Independent of features enabled (NAT, ACL)
- Full mesh, 100% linerate

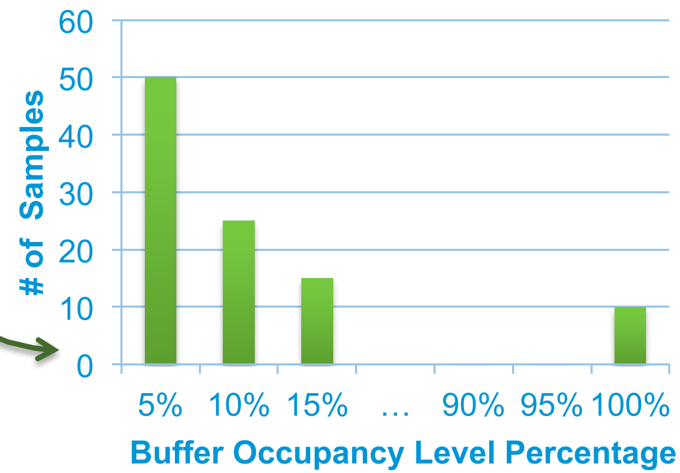
Active Latency Monitoring



Active Buffer Monitoring







Conclusion

- Ethernet is now suitable for ULL applications:
    - Delays as low as 250ns
    - Same delay for L2 and L3
    - No latency penalty when activate smart features (NAT/ACL)
    - Ultralow jitter (8ns worst case)
- SoC design combined with advanced features (buffers, monitoring, etc.) allowed high performance, ultra-low latency switching
- Pick carefully the features that you need for your network (availability, buffering, 10GE vs 1GE, Latency) in order to reach the required performance

Thank you.