Examples of Financial Sector Network requirements

Low-latency links / Monitoring / PTPv2 clocksync / BGP as IGP

Pedro V. Estrela, PhD
Sr. Performance Engineer
30-May-2018

About the presenter

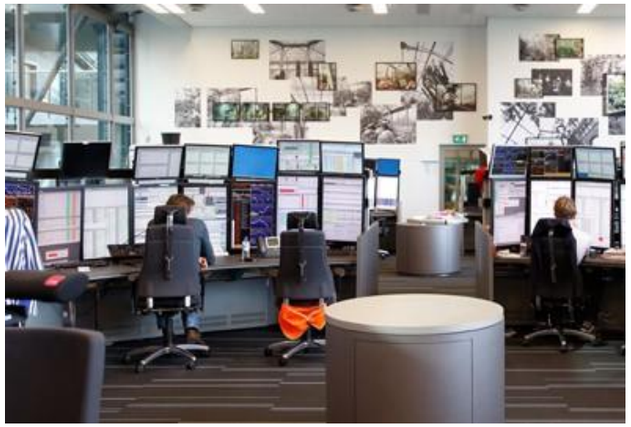
- PhD in mobile networks (2007)
  - NS2 simulations
  - Transparent mobility
  - Fast mobile-IP handovers

- Performance engineer (2008 ->)
  - Think of the engineer that tunes Formula1 cars
  - Latency optimization and analysis
  - Prototypes and reverse engineering

About IMC financial markets

- Think of the currency house at the airport, but for:
  - Many products: Options / Futures / Bonds / ETFs
  - Fully automated operations
  - Fast worldwide network

- Everybody technical
  - Trading team = Determine Prices
  - Technology team = Adjust orders Quickly



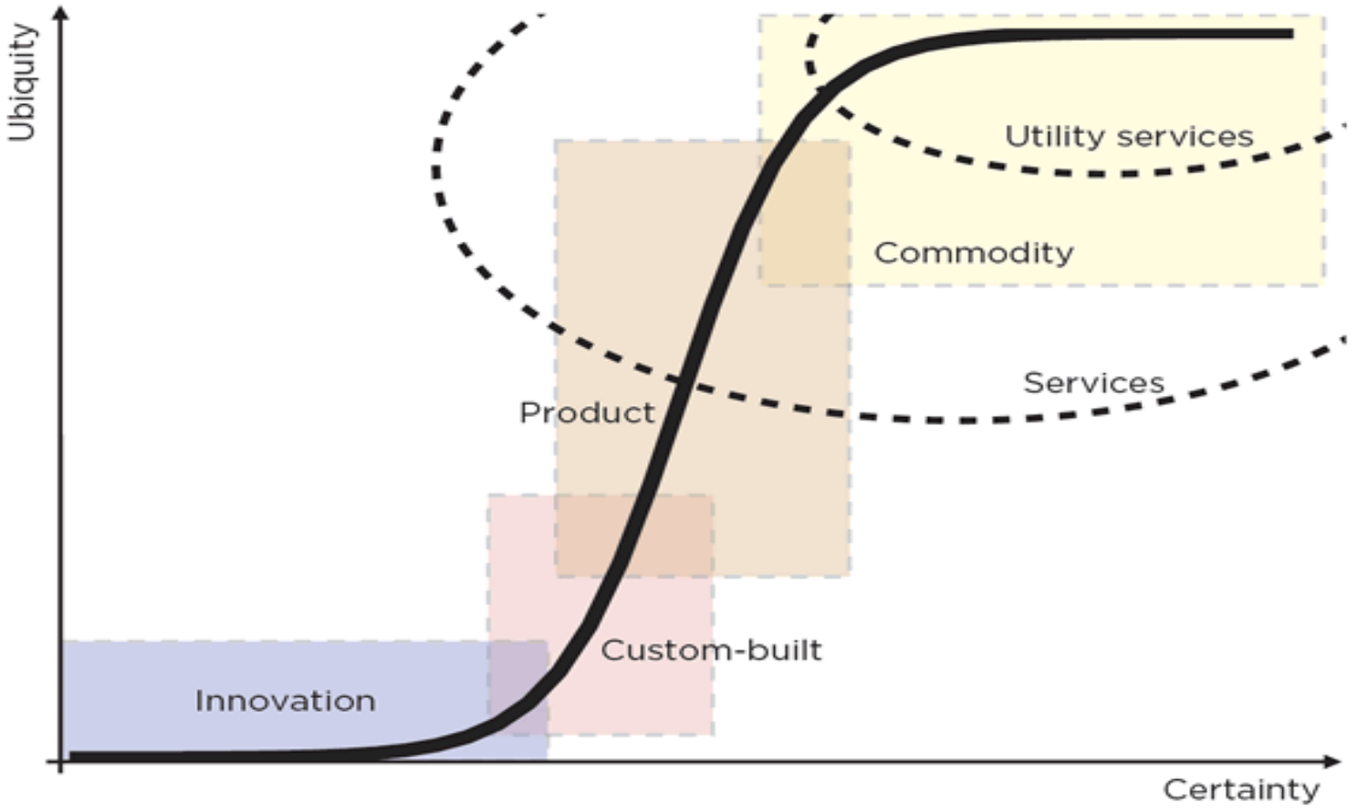
WHAT WE DO IN LESS THAN 2 MIN.

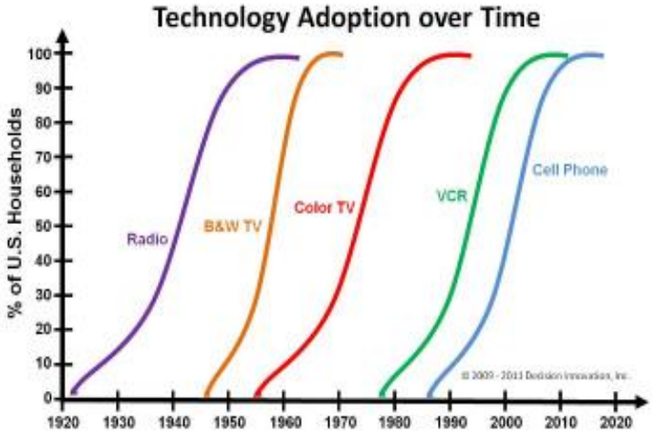
http://www.imc.com/eu/about-us#what-we-do
https://www.youtube.com/watch?v=WFsvY_YRhvg

Exchange price-time priority

Source: http://www.tbo.com/news/business/tampa-area-shopppers-get-jump-on-black-friday-deals-20141127/

Relative latency -> Innovation

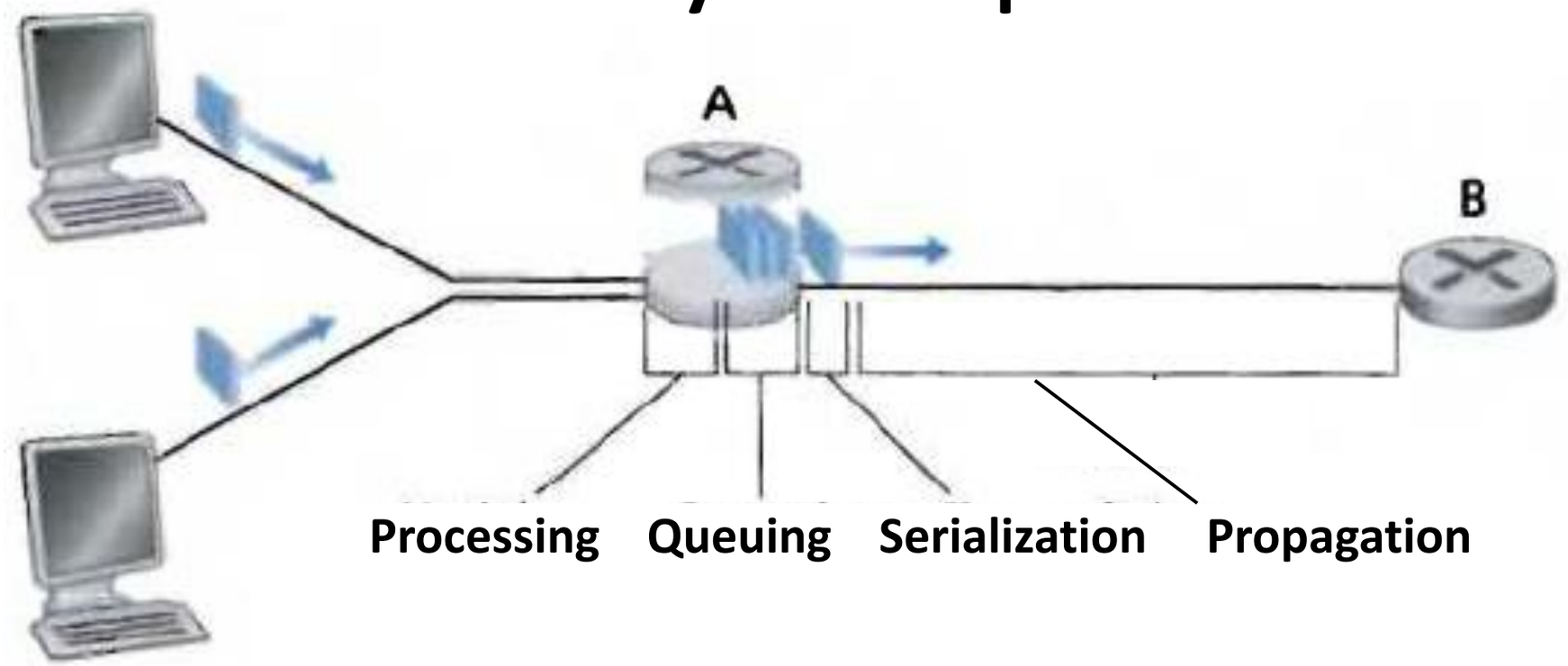




Source: *https://www.slideshare.net/CloudCamp/evolution-of-business-activities*
Source: *http://mapthefuture.mikemace.com/2013/02/excerpts-from-map-future.html*

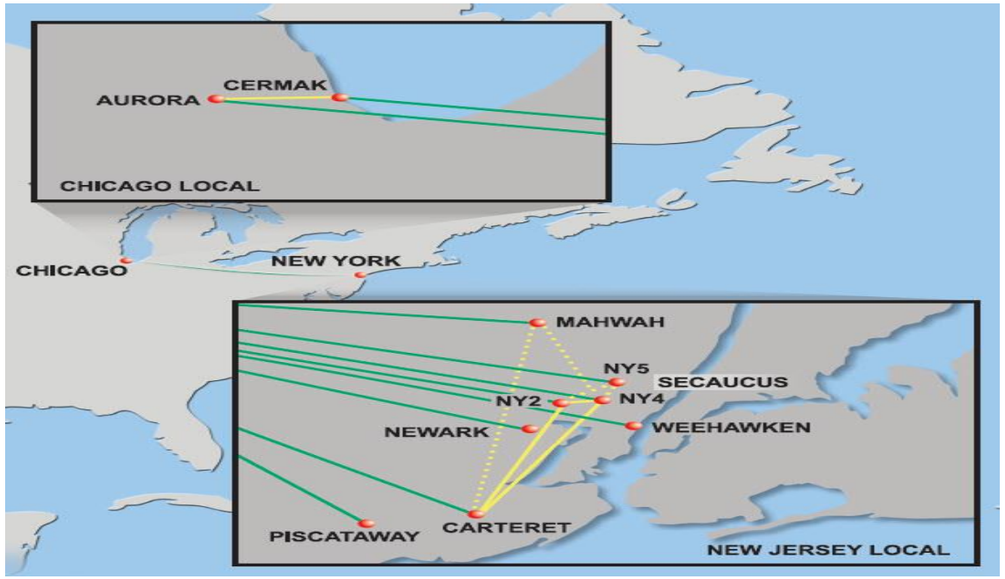
Low-Latency Links

Latency components

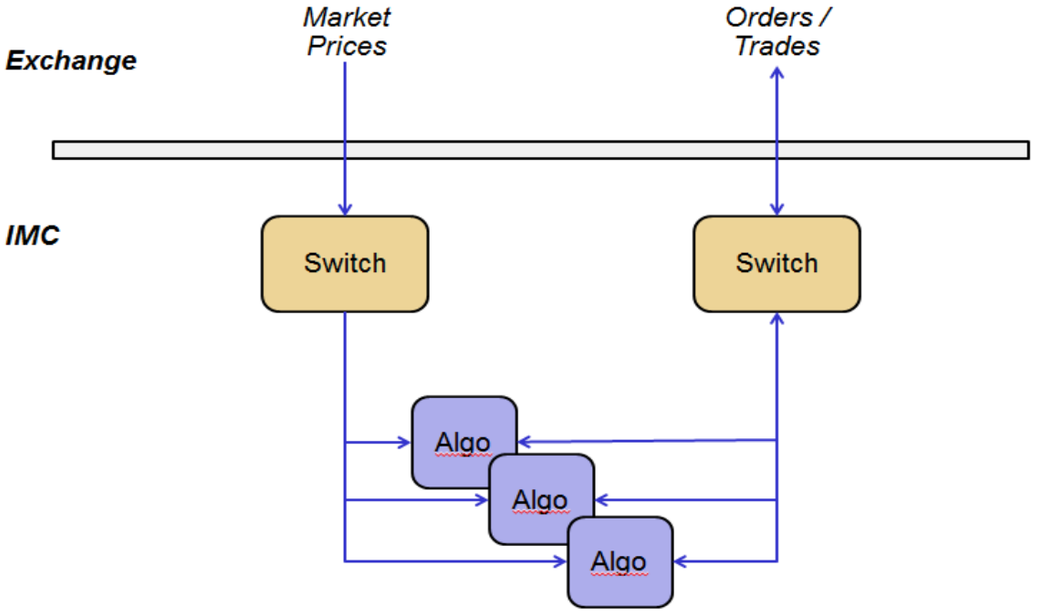


Trend is clearly: Faster / Raw Hardware / More expensive

Wide Area



Local Area



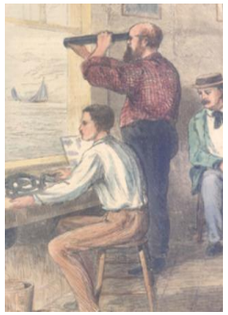
WAN examples

Historical fact: Every new wide area technology was first used for trading

Year	Technology	Who
1815	**Pigeons**	*Rothchild knew that Napoleon lost the war first*
1836	**Telescopes**	*Shore agents knew if cargo was spoilt on Boats*
1897	**Telegraph**	*Bookies sent Horse race results to outside the tracks*
2010	**Fiber**	*Spread networks drills mountains on NY-Chicago*
2012	**Microwave**	*McKay jumps the very same mountains using Radio*
2015	**Fiber**	*Hibernia builds new straighter Atlantic cable*
…	**…**	**…**

Hours / Days

Milli Seconds



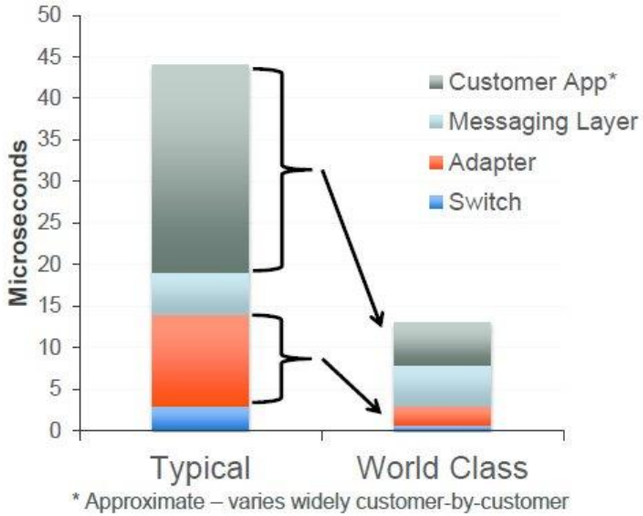
Source: http://www.forbes.com/forbes/2010/0927/outfront-netscape-jim-barksdale-daniel-spivey-wall-street-speed-war.html
Source: https://www.moaf.org/publications-collections/financial-history-magazine/111/_res/id=Attachments/index=0/Plundered_by_Harpies.pdf

LAN examples

Year	Device	Latency
2008	Cisco 4900	*2600ns OWD*
2011	Cisco 3064	*1000ns OWD*
2011	Arista 7124	*500ns OWD*
2013	Cisco 3548	*200ns OWD*
…	**…**	**…**

Micro Seconds

Nano Seconds

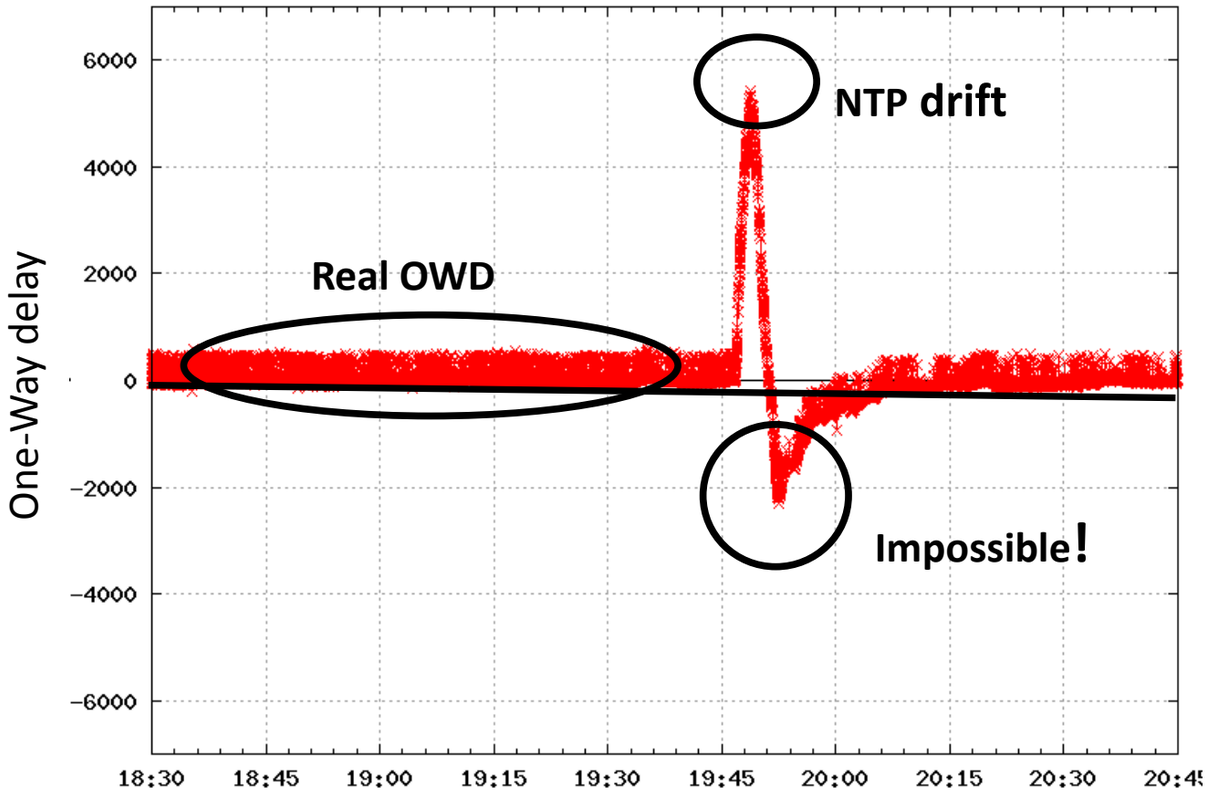


(Circa 2012)

Source: https://www.cisco.com/c/dam/en/us/products/collateral/switches/catalyst-4900-series-switches/press_coverage.pdf
Source : https://www.arista.com/en/company/news/press-release/352-pr-20110314-01
Source : https://newsroom.cisco.com/press-release-content?type=webcontent&articleId=1028561
Source: https://newsroom.cisco.com/press-release-content?articleId=362594
Source: https://www.theregister.co.uk/2012/02/08/solarflare_application_onload_engine/

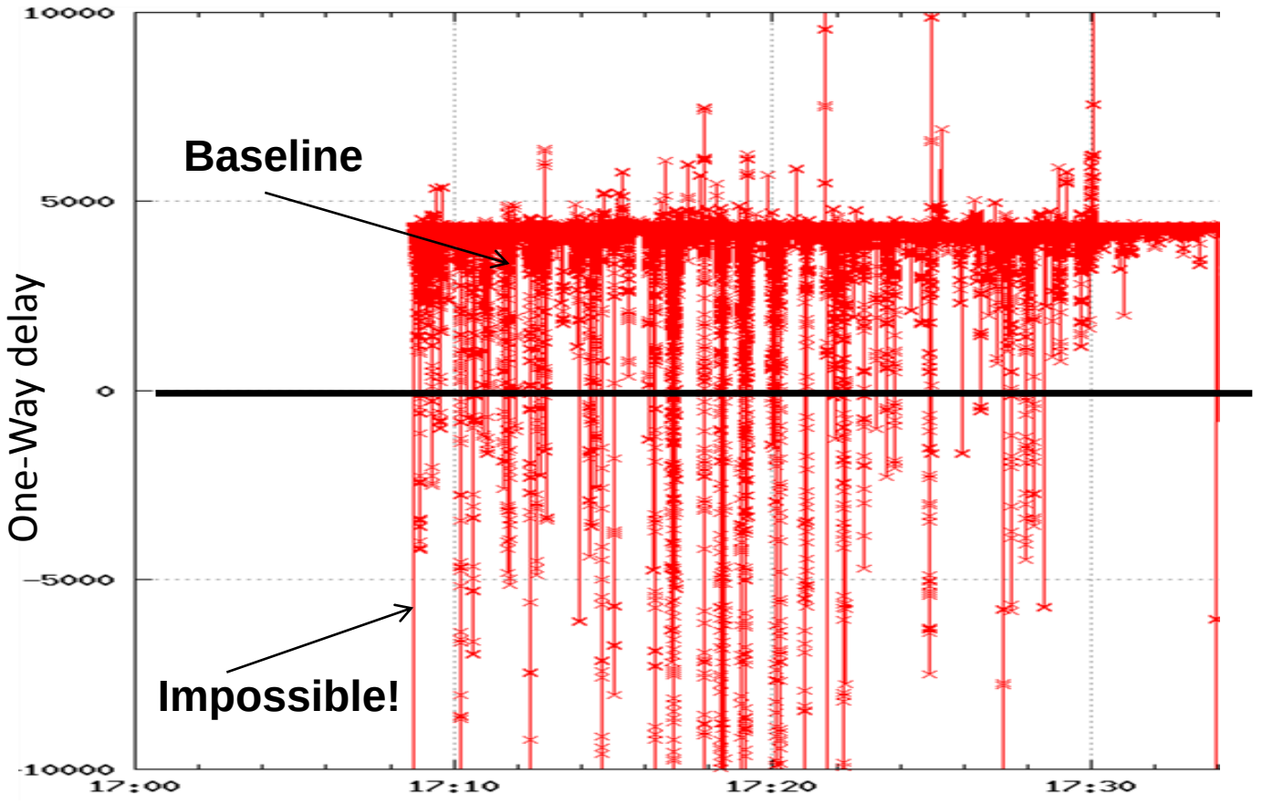
Network Monitoring

NTP clocksync



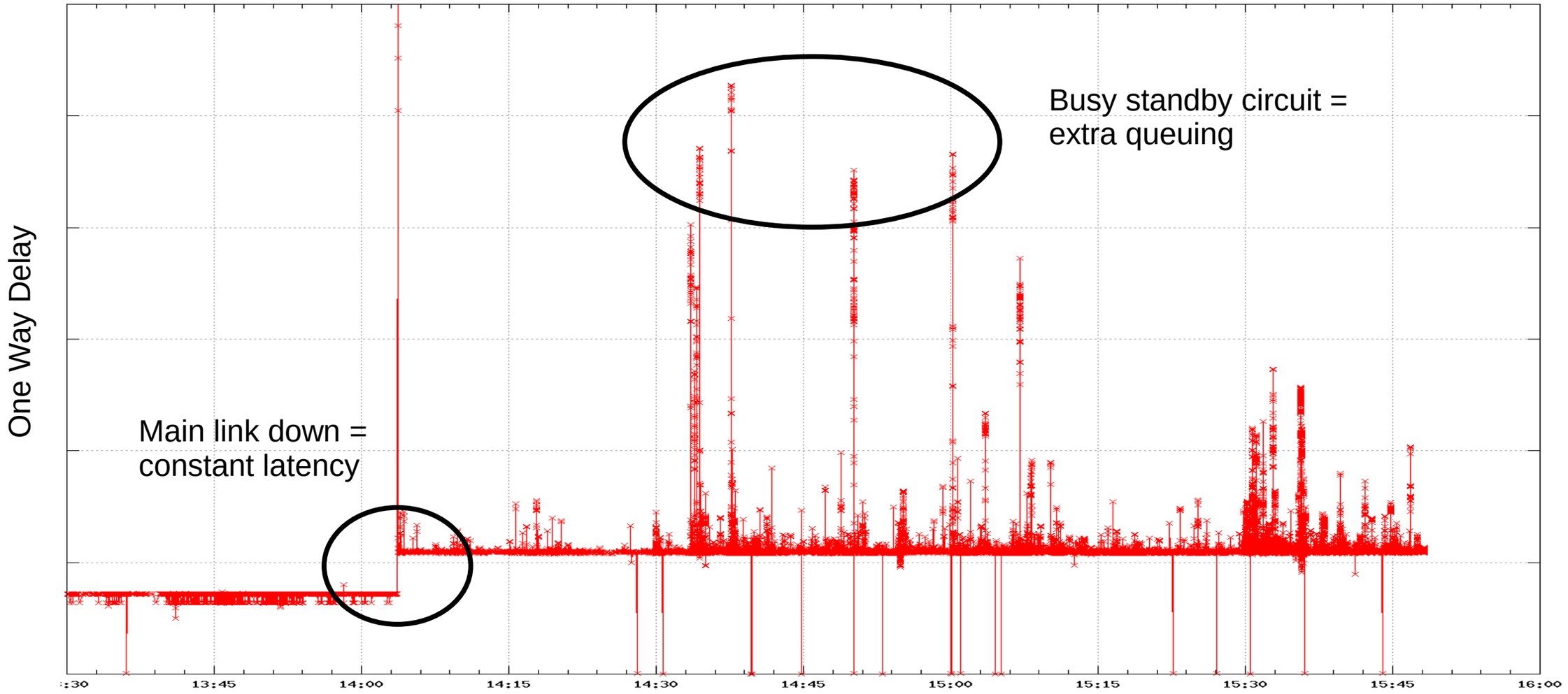
(Circa 2010)

SW timestamps



(Circa 2010)

Network convergence impact

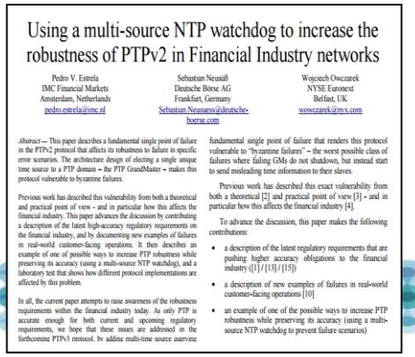


(Circa 2010)

Increasing robustness of PTPv2 Financial networks

25 September
ISPCS 2014
Austin, Texas

(2014 paper)



Using a multi-source NTP watchdog to increase the robustness of PTPv2 in Financial Industry networks

Pedro V. Estrela
IMC Financial Markets
Amsterdam, Netherlands
pedro.estrela@imc.nl

Sebastian Neusüß
Deutsche Börse AG
Frankfurt, Germany
Sebastian.Neusuess@deutsche-boerse.com

Wojciech Owczarek
NYSE Euronext
Belfast, UK.
wowczarek@nyx.com

Abstract — This paper describes a fundamental single point of failure in the PTPv2 protocol that affects its robustness to failure in specific error scenarios. The architecture design of electing a single unique time source to a PTP domain – the PTP GrandMaster – makes this protocol vulnerable to byzantine failures.

Previous work has described this vulnerability from both a theoretical and practical point of view - and in particular how this affects the financial industry. This paper advances the discussion by contributing a description of the latest high-accuracy regulatory requirements on the financial industry, and by documenting new examples of failures in real-world customer-facing operations. It then describes an example of one of possible ways to increase PTP robustness while preserving its accuracy (using a multi-source NTP watchdog), and a laboratory test that shows how different protocol implementations are affected by this problem.

In all, the current paper attempts to raise awareness of the robustness requirements within the financial industry today. As only PTP is accurate enough for both current and upcoming regulatory requirements, we hope that these issues are addressed in the forthcoming PTPv3 protocol. Its addme multi-time source ouervive fundamental single point of failure that renders this protocol vulnerable to "byzantine failures" – the worst possible class of failures where failing GMs do not shutdown, but instead start to send misleading time information to their slaves.

Previous work has described this exact vulnerability from both a theoretical [2] and practical point of view [3] - and in particular how this affects the financial industry [4].

To advance the discussion, this paper makes the following contributions:

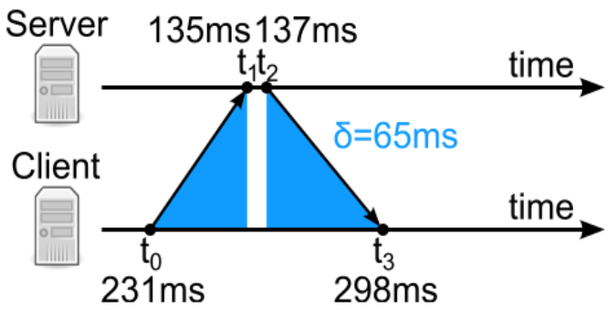
- a description of the latest regulatory requirements that are pushing higher accuracy obligations to the financial industry ([1] [13] [15])
- a description of new examples of failures in real-world customer-facing operations [10]
- an example of one of the possible ways to increase PTP robustness while preserving its accuracy (using a multi-source NTP watchdog to prevent failure scenarios)

(best paper award)

Network time synchronization

IETF NTPv4:

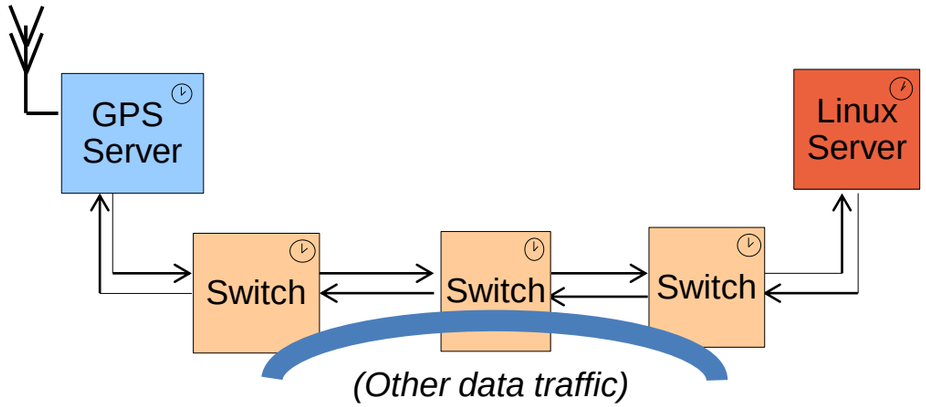
- Mature
- milli-seconds accuracy
- Multiple time sources



https://en.wikipedia.org/wiki/Network_Time_Protocol

IEEE PTPv2:

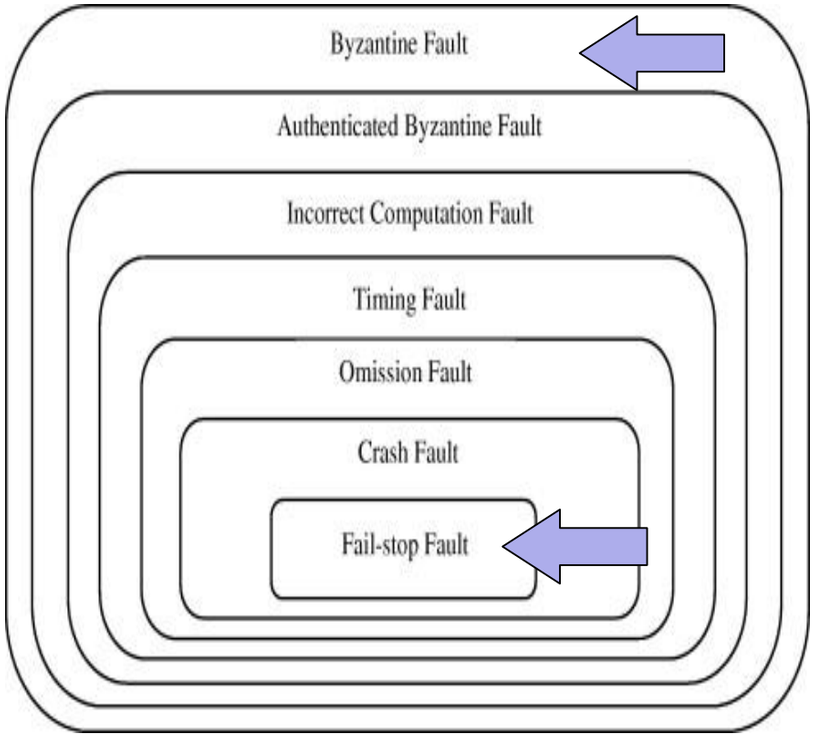
- Fresh
- nano-seconds accuracy
- Single time source



https://en.wikipedia.org/wiki/Precision_Time_Protocol

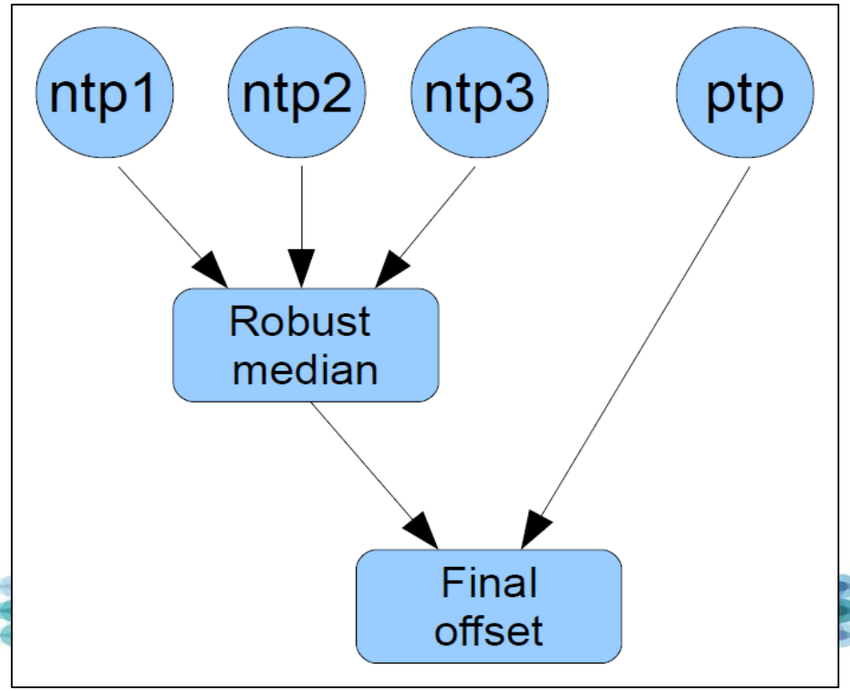
PTPv2 byzantine failures

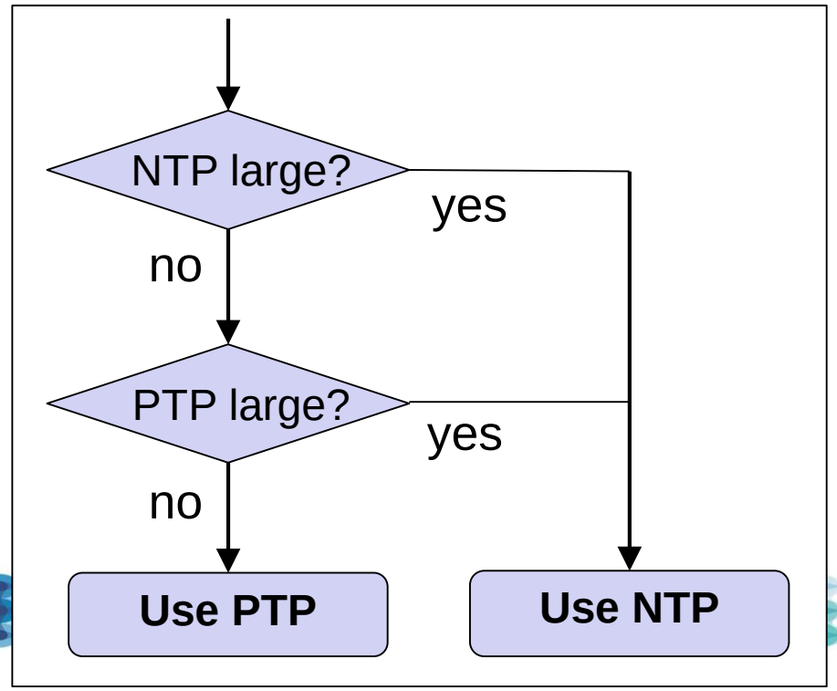
- Deutche Borse, August 2013
  - Active GM sent *bad* time (leap seconds = 0)
  - Backup GMs remained passive
  - Slaves jumped 35s => Trading halted
- IMC, July 2011
  - Same issue: Single source
- ESMA, Jan 2018
  - Regulator requires traceable 100us error



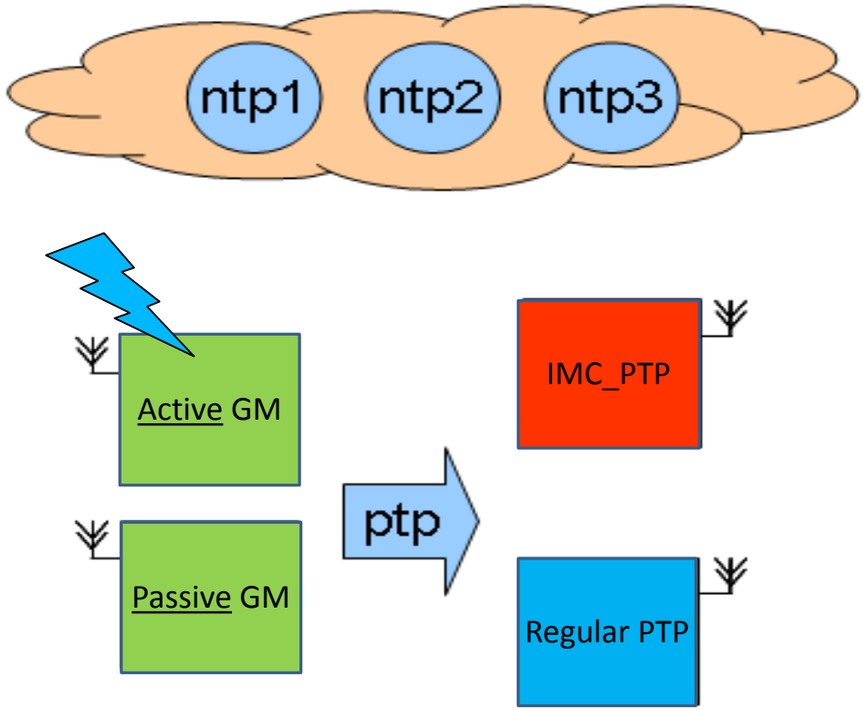
Run PTP + an NTP watchdog

- 3x NTP servers queried in parallel to PTP
- Median overrides PTP offset
  - -0.02 ms
  - +0.01 ms
  - +35000 ms
- PTP only touches the clock if allowed

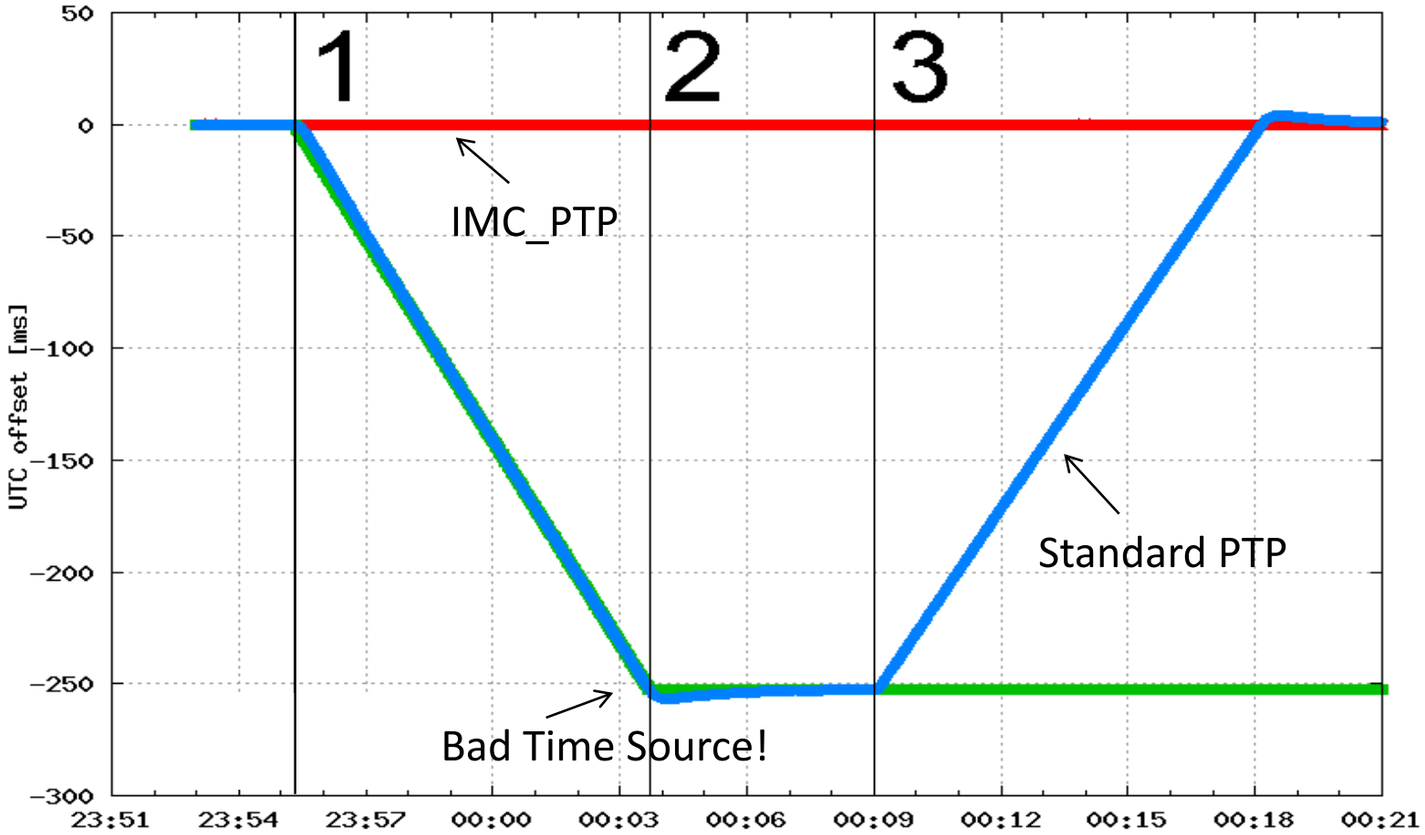




Testbed

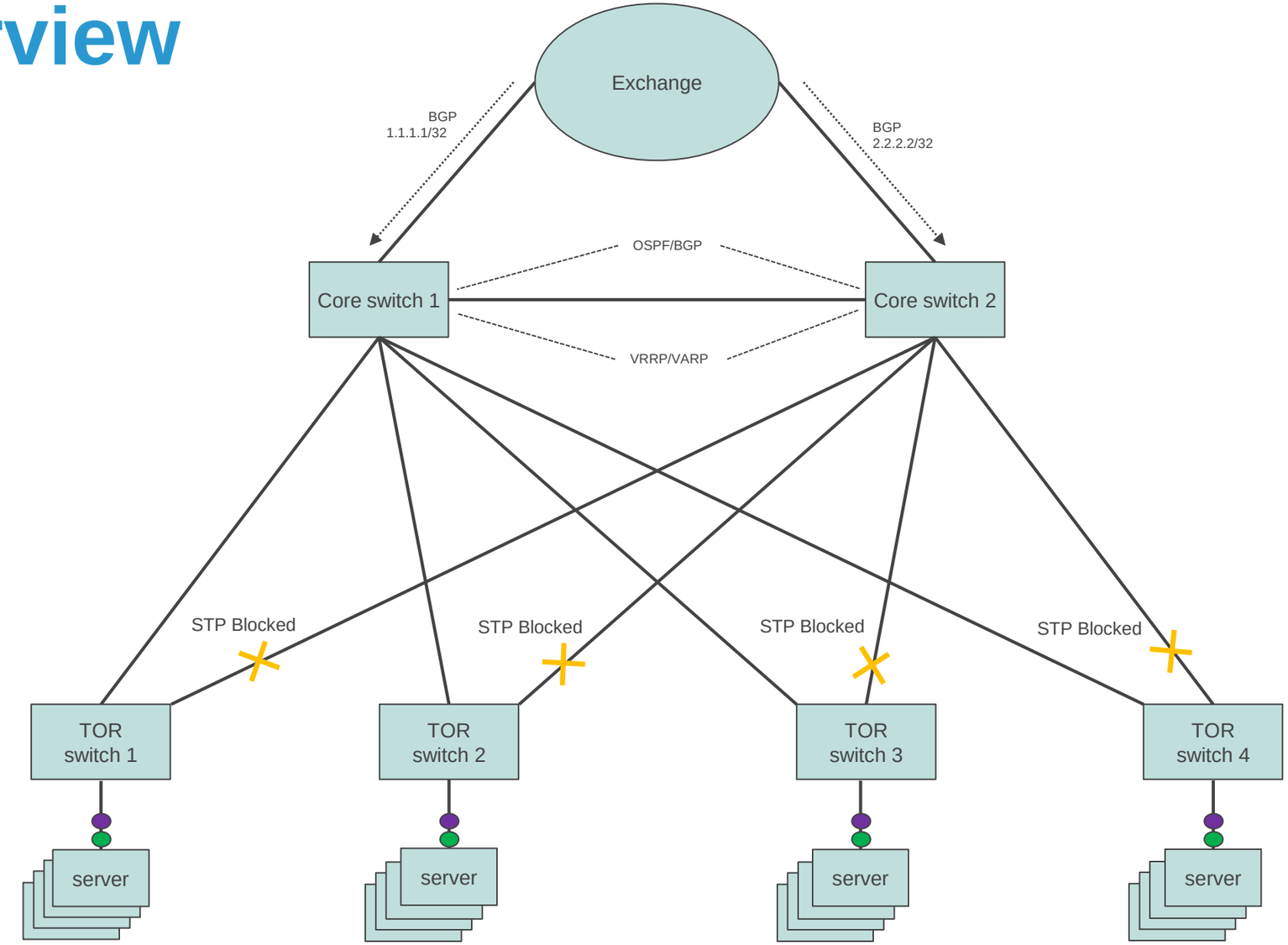


Clock error results



Using BGP as an IGP

DC overview


Exchange
BGP
1.1.1.1/32
BGP
2.2.2.2/32
OSPF/BGP
Core switch 1
Core switch 2
VRRP/VARP
STP Blocked
STP Blocked
STP Blocked
STP Blocked
TOR
switch 1
TOR
switch 2
TOR
switch 3
TOR
switch 4
server
server
server
server

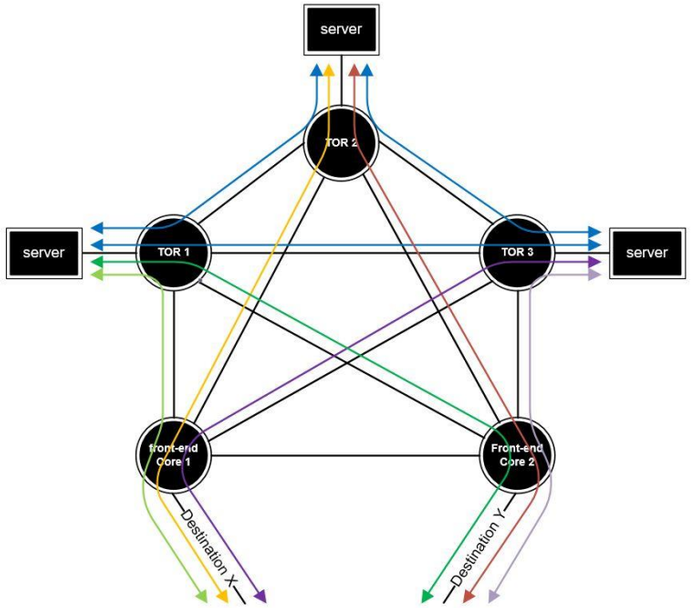
Issue description

- L3 IGP routing limitations
  - Traffic engineering is hardly possible
  - Valuable bandwidth not used
  - Multiples protocols cause complexity and risk
    - OSPF, BGP, EIGRP, static routing

- L2 switching limitations
  - Spanning-tree blocked ports
  - MLAG / TRILL = no control on flows, proprietary, special HW requirements
  - The risk of using ancient layer-2 technology is a risk by itself

Solution: BGP mimicked as an IGP

- Inspired by RFC 7938 (facebook / microsoft / arista)
  - Main difference: no symmetrical infrastructures, especially WAN
  - 1x switch = 1x confederation-AS BGP (best of iBGP+eBGP worlds)
- Per-hop routing decisions
  - Using additive latency-based metrics (MED), instead of AS-PATH
- Advanced traffic engineering
  - Based on prefix communities. Flexible from whole DCs to /32 hosts
  - Separate high/low-latencies; Separate elephants/mice
- Fast Failover (<100ms)
  - Timer tuning
  - Bidirectional forwarding detection



Thanks!

- Network requirements
  - LAN + WAN Low-latency
  - Scalability & Automation
  - Visibility & Clock Sync

- More questions?
  - IMC: *https://www.imc.com/eu/careers*
  - Scientific papers: *https://www.researchgate.net/project/PTP-Clock-Synchronization*