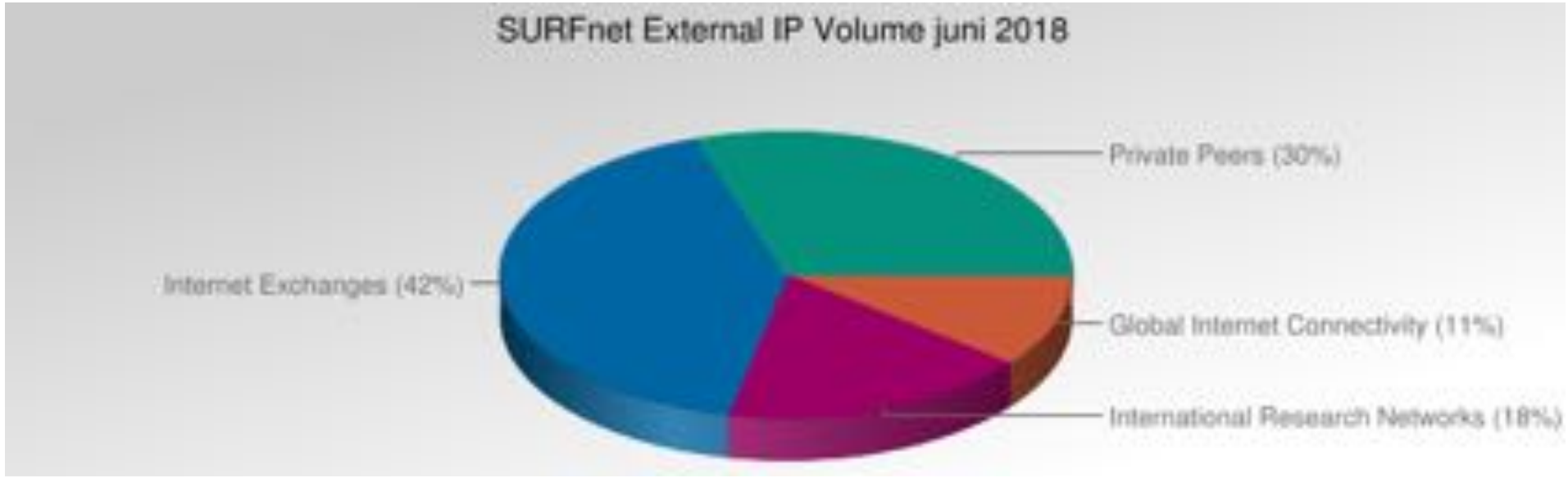
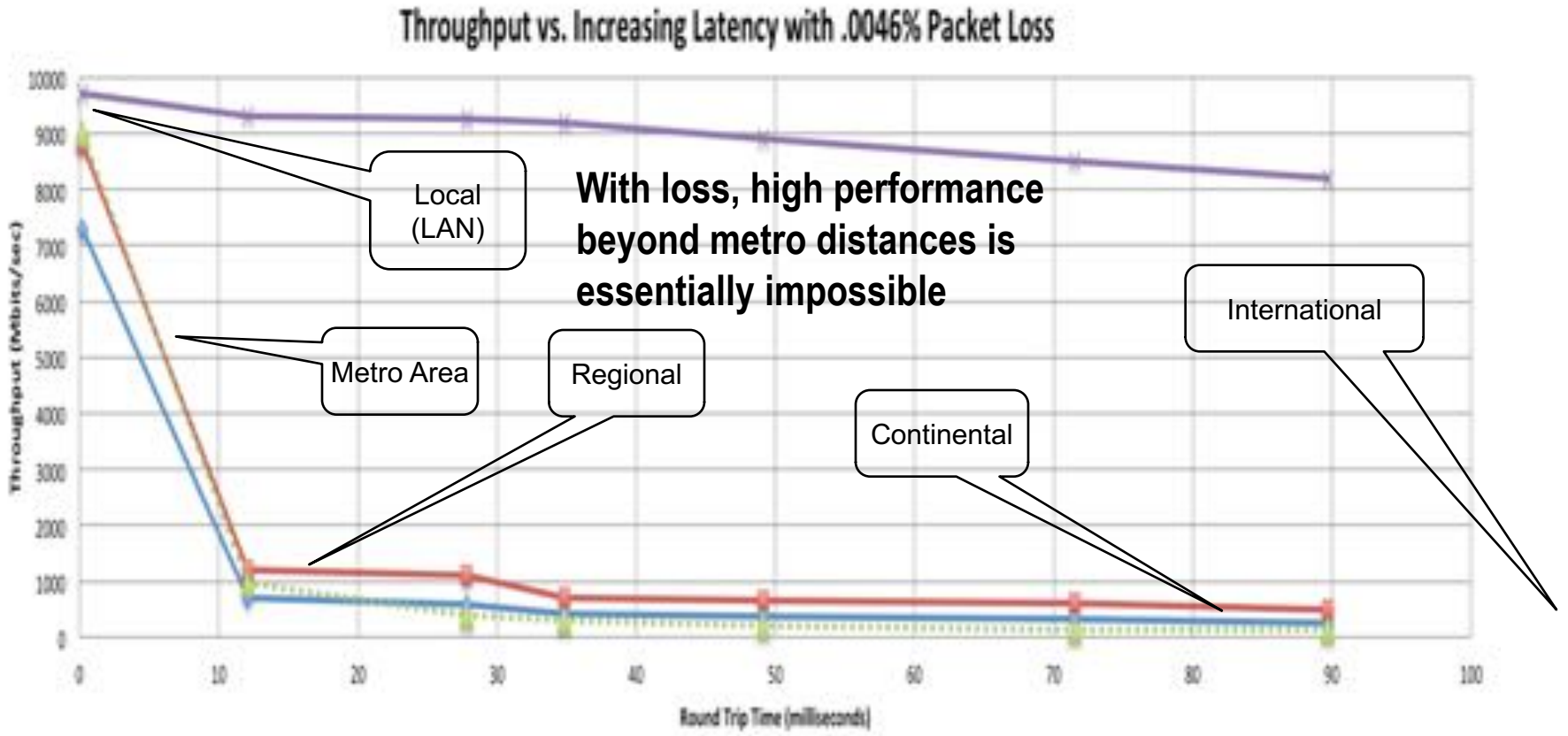
Traffic categories



International Data Transfers with TCP for Example Radio Astronomy Data



What about UDP?

- Interesting discussions around UDP
  - Google's QUIC and IETF version of QUIC
  - >8% of traffic (depending on the point of analysis)
  - Interesting APNIC blog: https://blog.apnic.net/2018/05/15/how-much-of-the-internet-is-using-quic/
  - More interesting research done by: *Jan Rüth, PhD student at the Chair for Communication and Distributed Systems at RWTH Aachen University in Germany.*
- Other commercial UDP based applications are available
  - I.e., Aspera (IBM)
- Other open source UDP tools…
  - UDT

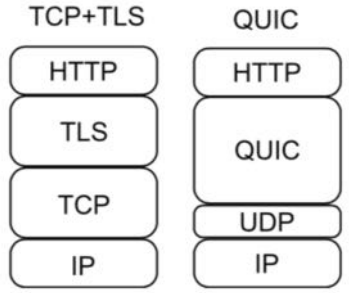


Figure 1. Comparison of TLS and QUIC

https://www.ietfjournal.org/quic-performance-and-security-at-the-transport-layer/

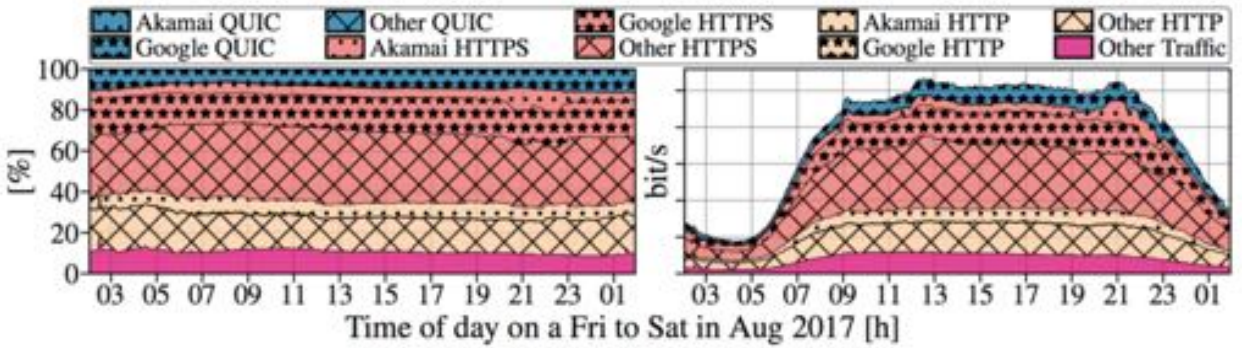
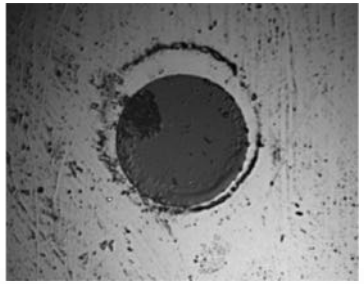
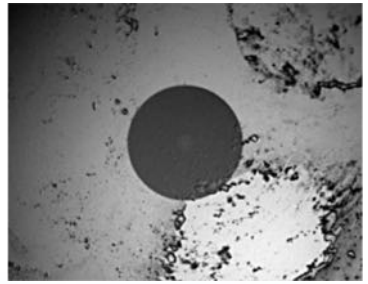


Figure 4: Relative QUIC shares (left) and absolute traffic (right) in the mobile network of a major European Tier-1 network. QUIC shares (blue) in contrast to HTTP (yellow) and HTTPS (red). Note: the ISP requested the actual traffic volume not be disclosed. From: https://blog.apnic.net/2018/05/15/how-much-of-the-internet-is-using-quic/

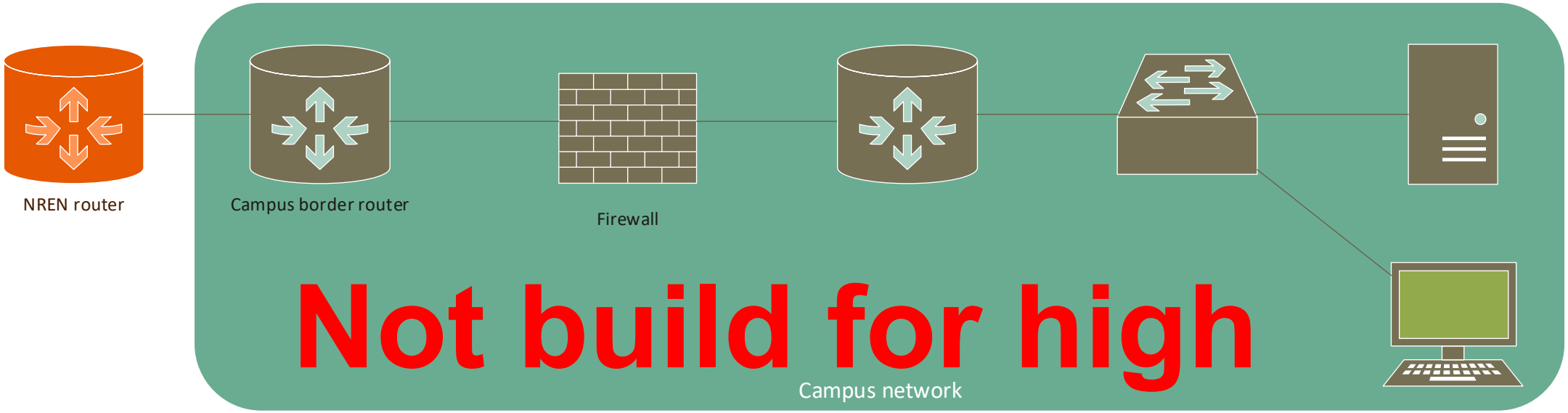
Network issues – Dirt

- Dirty fibers
  - GE cleaning was optional
  - 10GE fiber cleaning is strongly advised
  - 100GE fiber cleaning necessity
  - Actually just always clean them, get a fiber clearer
  - Optic interface cleaning with a fiber cleaning pen
  - SUNET has a great blog https://www.sunet.se/blogg/long-read-cleanliness-is-a-virtue/

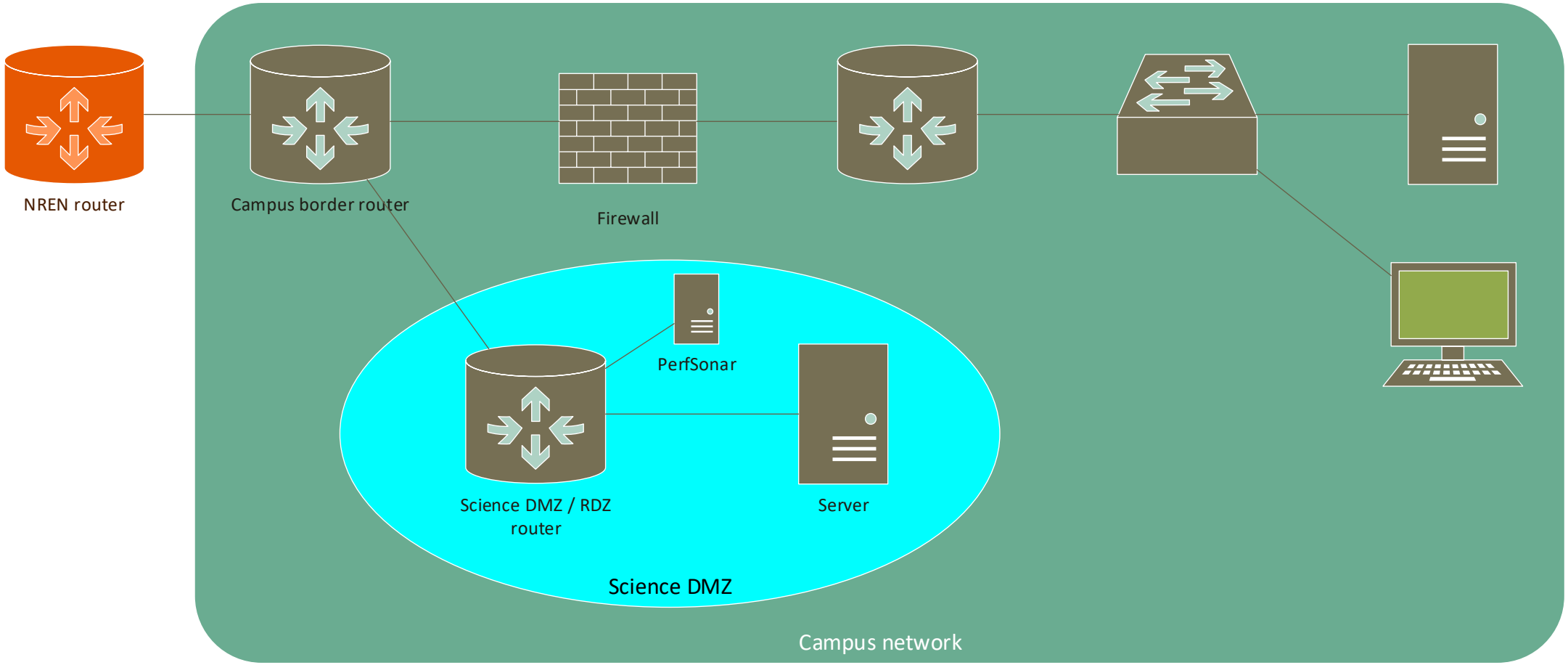


Typical campus network (very simple)



Not build for high performance

Network architecture solution…



Data security while in transit

Your data while being transported are just Ethernet frames

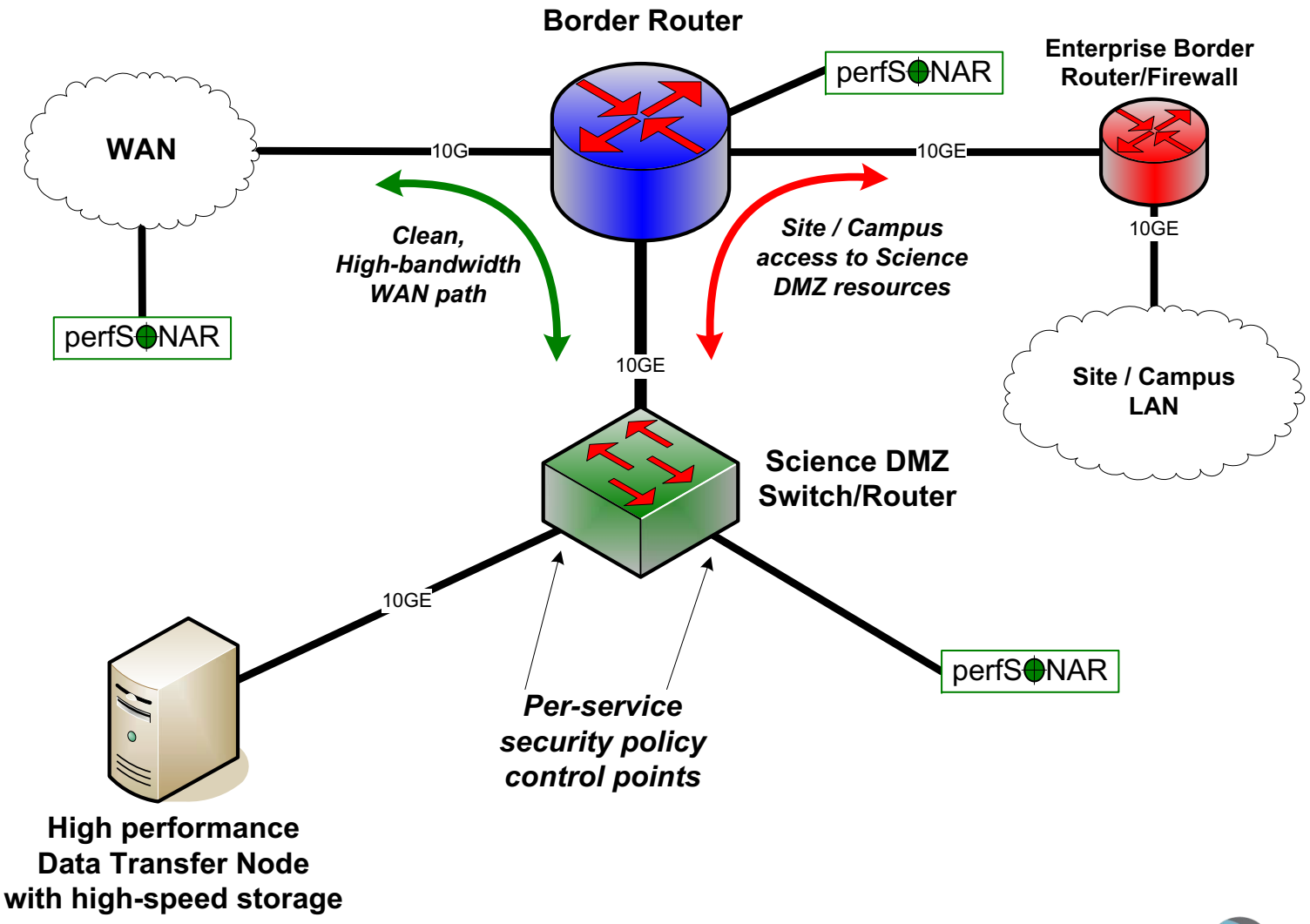
- On SURFnet DWDM backbone:
  - Colored WDM wave
  - Possibly 10G signals muxed into ONT4 container

Fiber tapping is a serious risk

- Only need 1% of optical light, which is below variations caused by temperature/pressure/other
- Don’t need to interrupt link to setup
- (nearly) impossible to detect

Consider your data to be at risk while in transit ➔ protect it (encryption)

Science DMZ Design Pattern (Abstract)

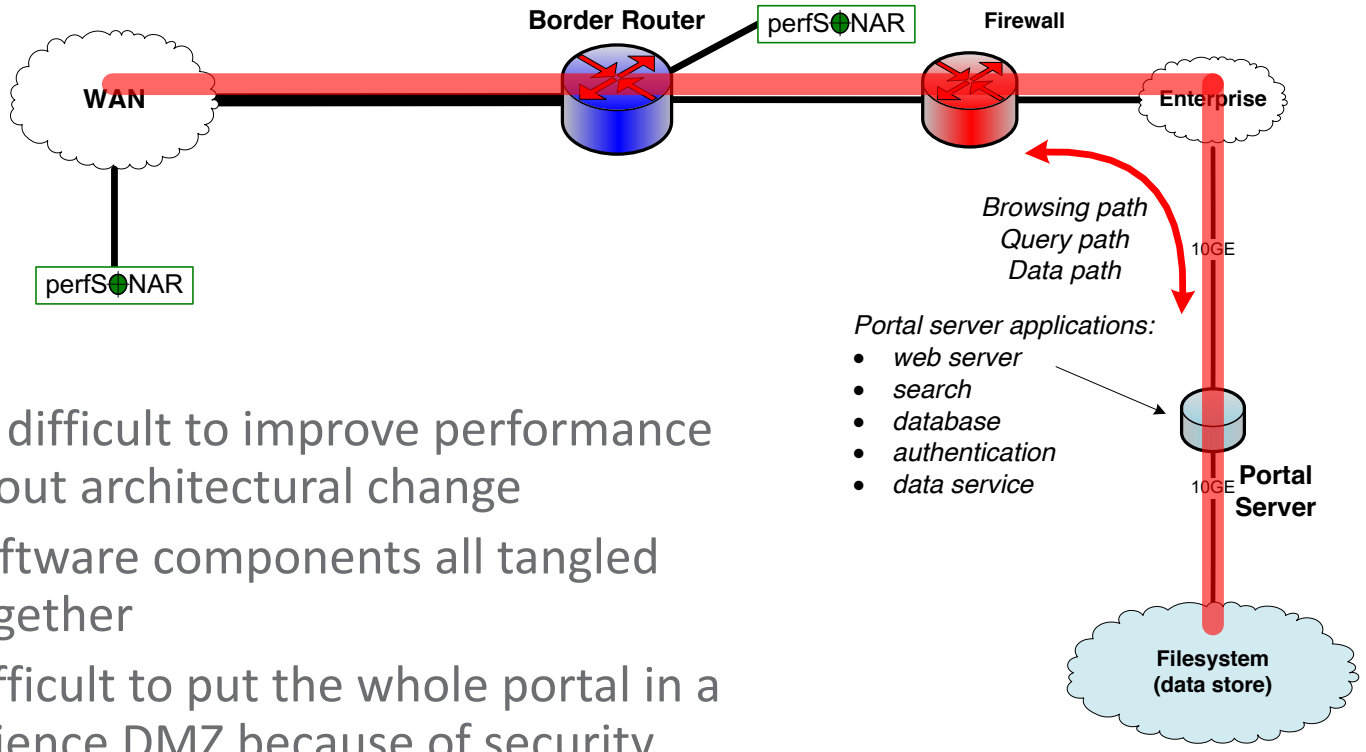


Intrusion detection

- Two major components in most R&E entities:
  - Bro:
    - Packet processing engine and event handler
    - Works as an IDS, but different from signature-based IDSes
    - Highly extensible policy language
    - Can basically be taught to handle many kinds of events, not just security events
  - Signature-based IDS:
    - Snort
    - Suricata
- Yes, you can run both on your campus and in your Science DMZ!

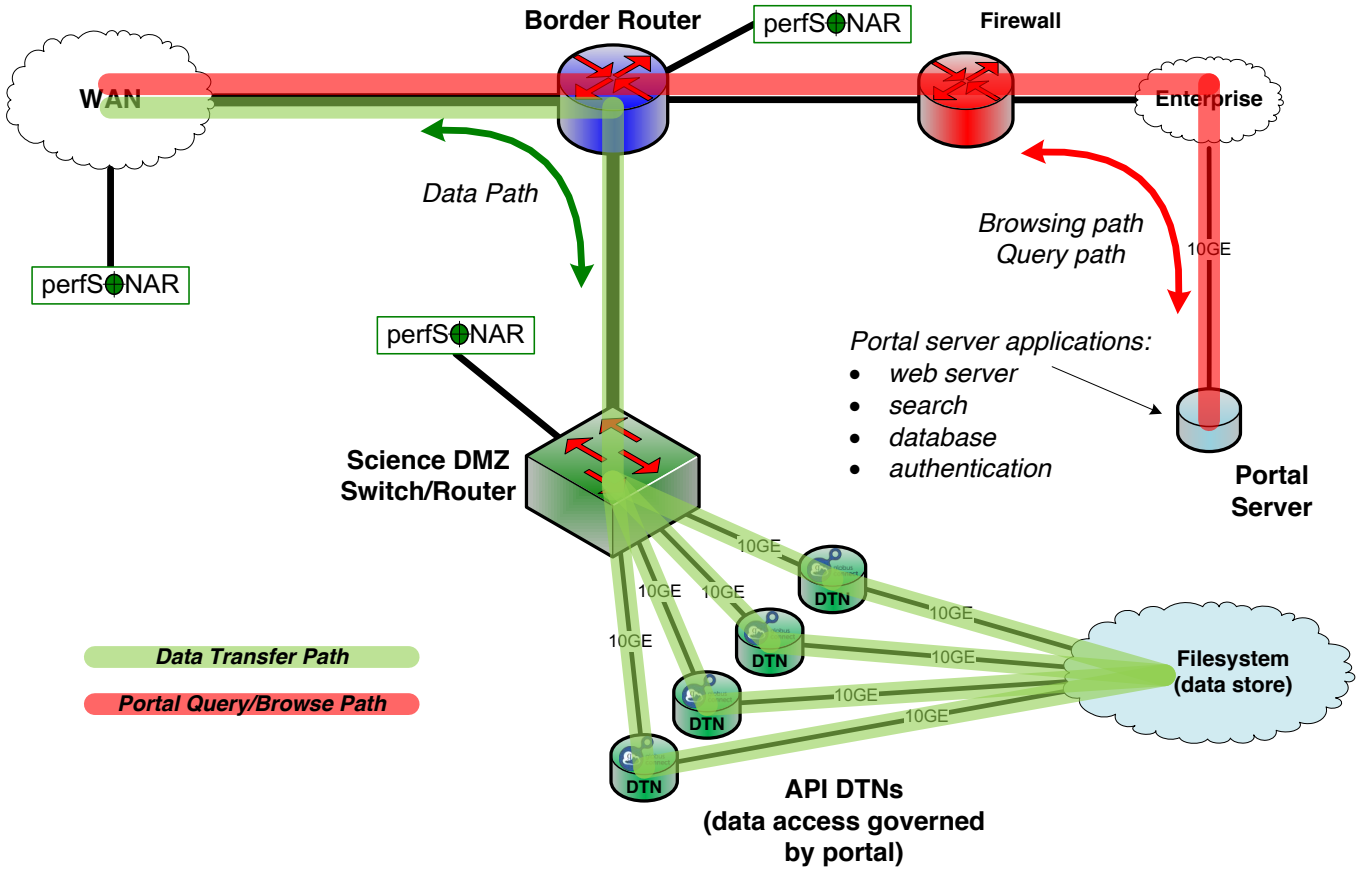
ESnet

Legacy Portal Design



- Very difficult to improve performance without architectural change
  - Software components all tangled together
  - Difficult to put the whole portal in a Science DMZ because of security
  - Even if you could put it in a DMZ, many components aren't scalable
- What does architectural change mean?

Next-Generation Portal Leverages Science DMZ

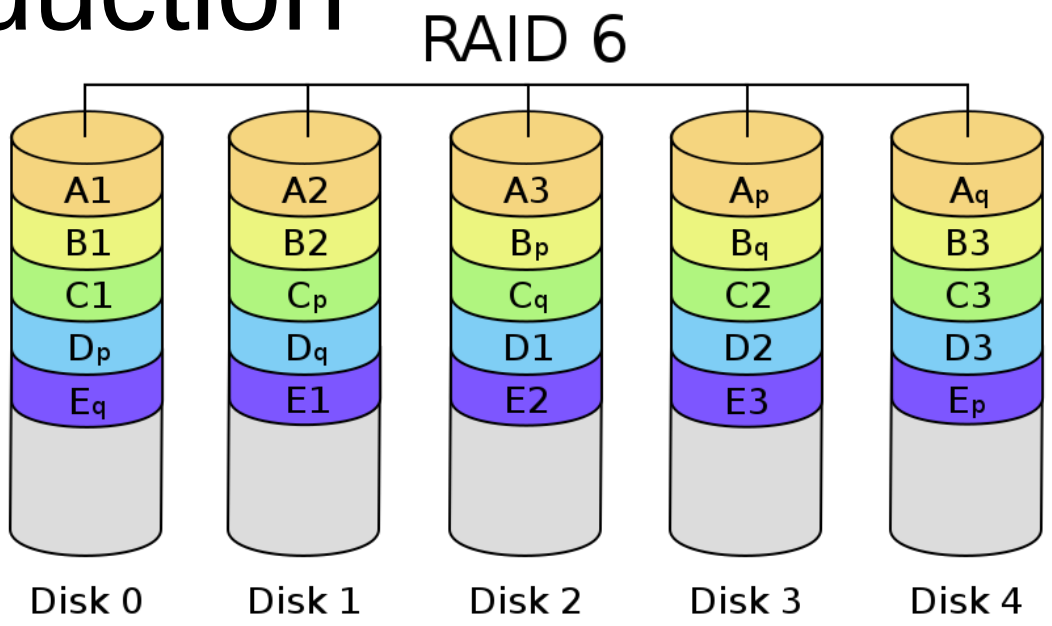


https://peerj.com/articles/cs-144/

ESnet

Introduction

- But then disks got bigger and bigger and rebuilding took longer and longer
- Chances increased of a second disk failure before the end of a rebuild
- No problem->RAID6
- Store two extra parity bits
- Now two disks can fail
- In case of disk failure, use parity bits on other disks to reconstruct the contents of the failed disk → rebuilding



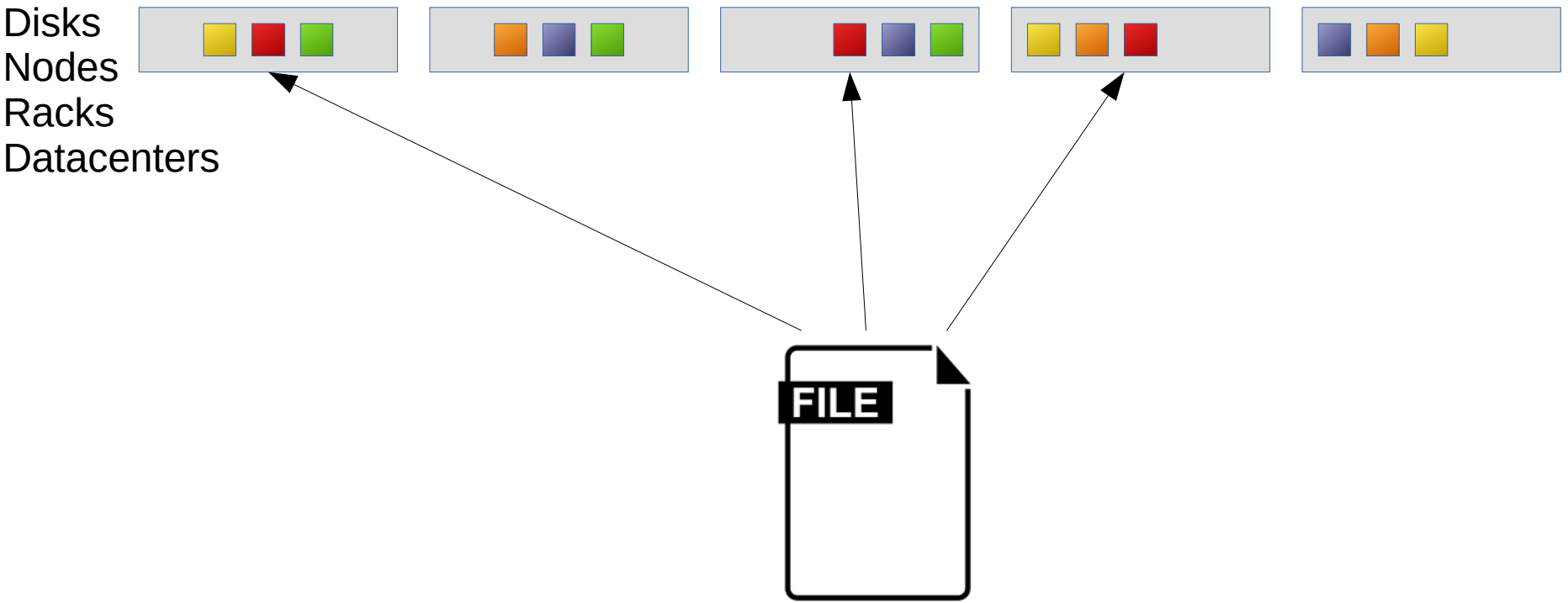
Introduction

- But the disks are still getting bigger and bigger and rebuilding still takes longer and longer
- So even rebuilding for RAID6 is going to take too long
- What do we do now?

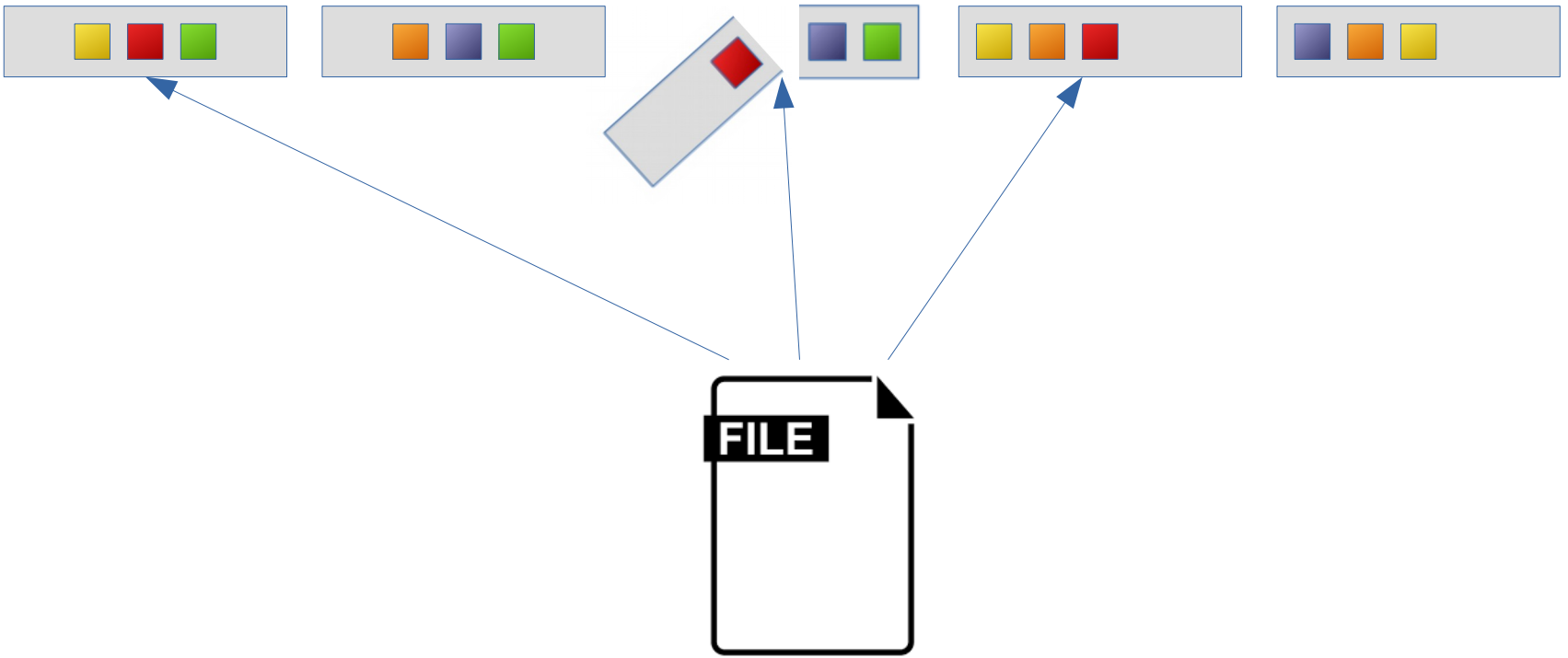
Data durability

- Redundancy not in single RAID set but distributed over nodes in a cluster

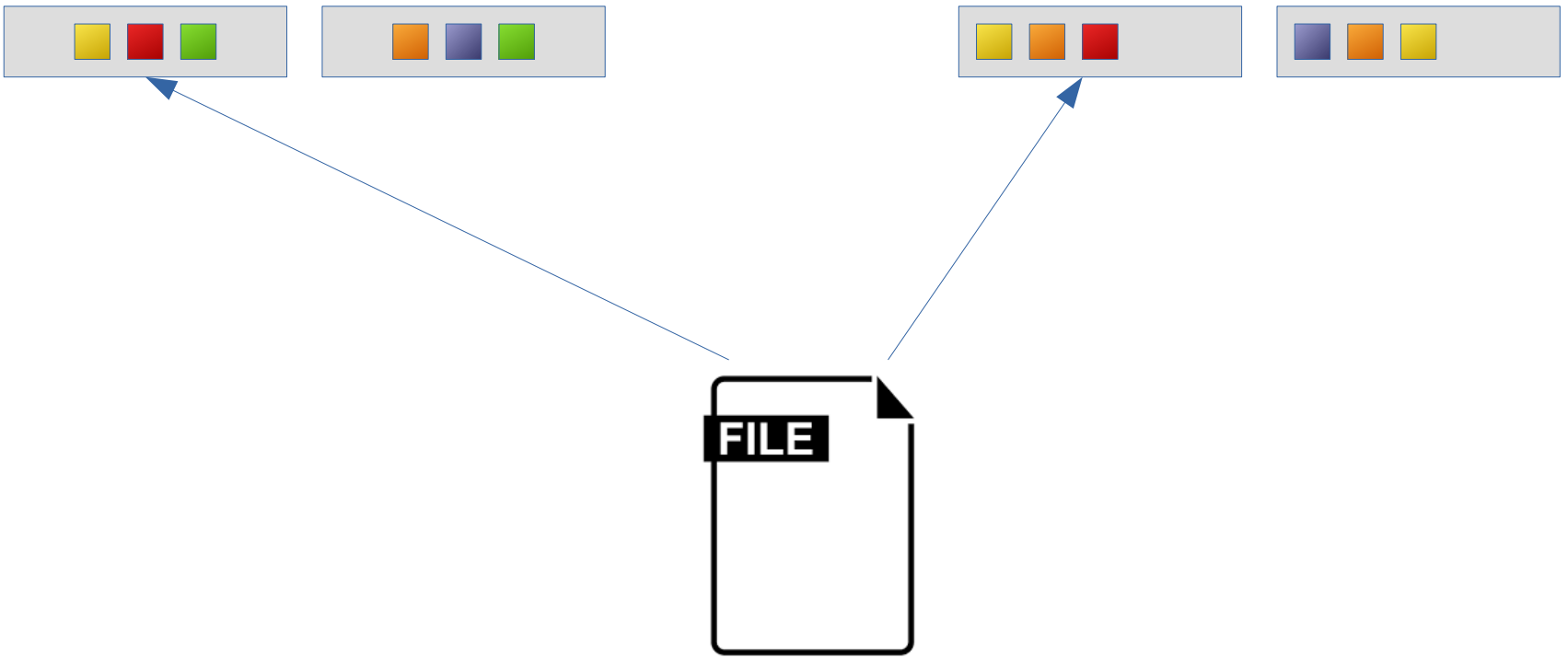
Data durability



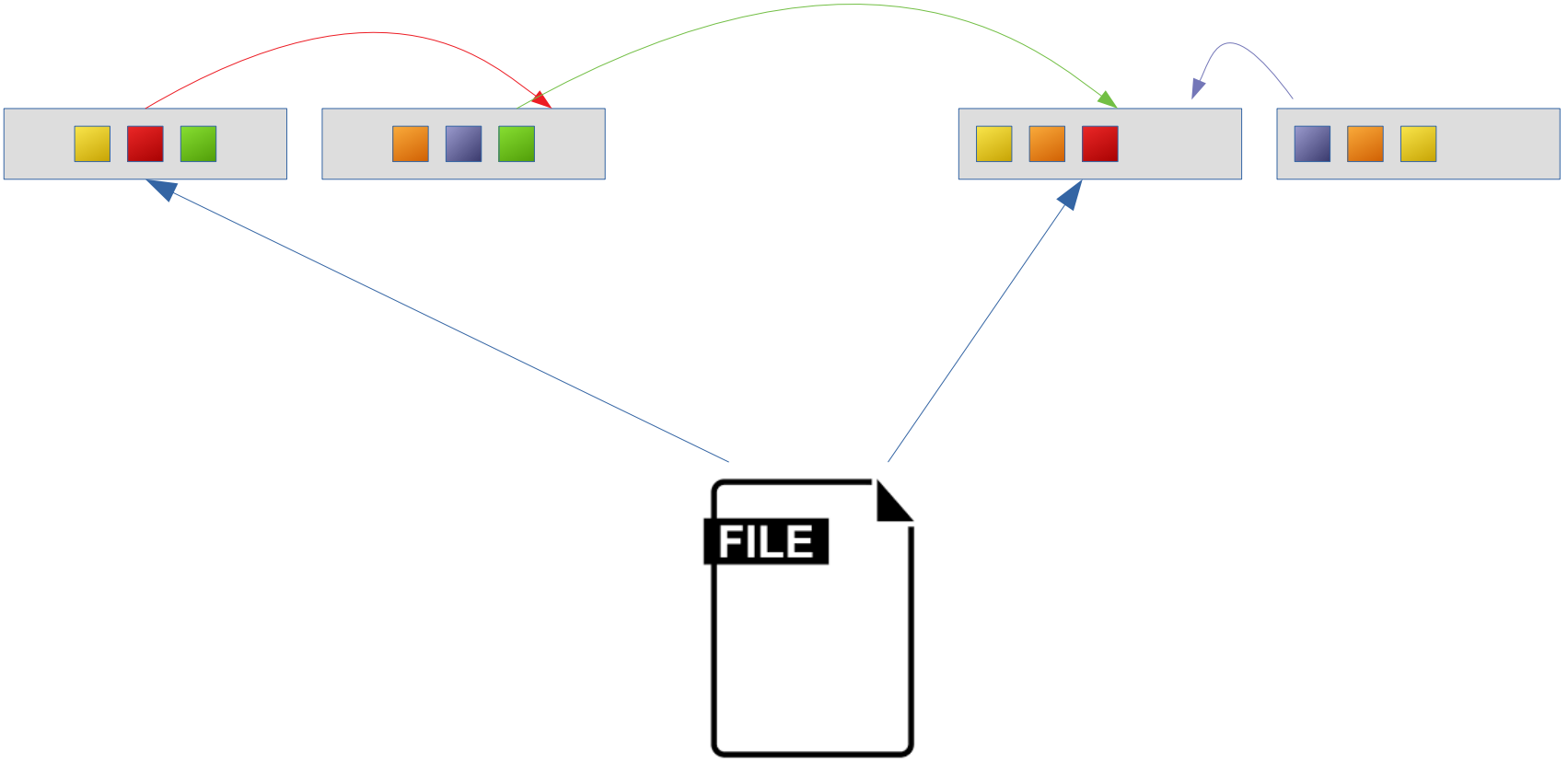
Data durability



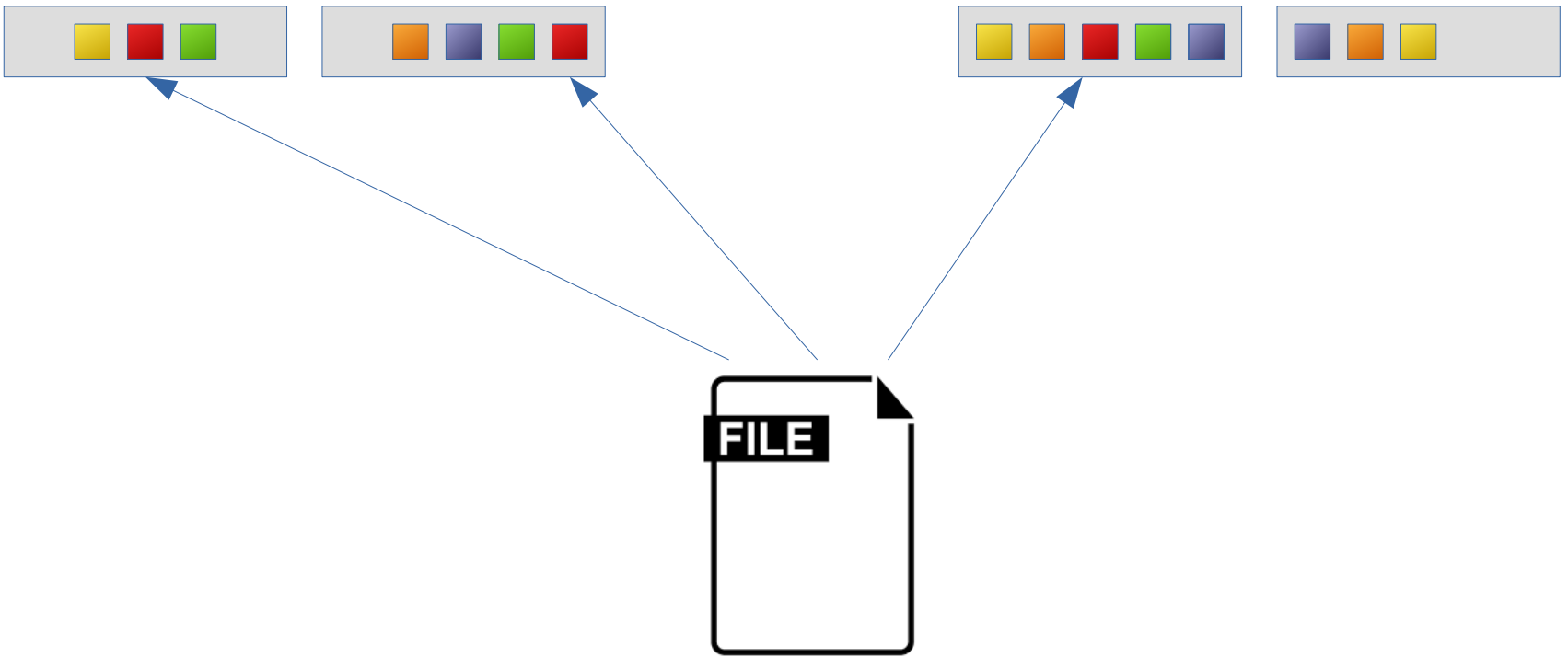
Data durability



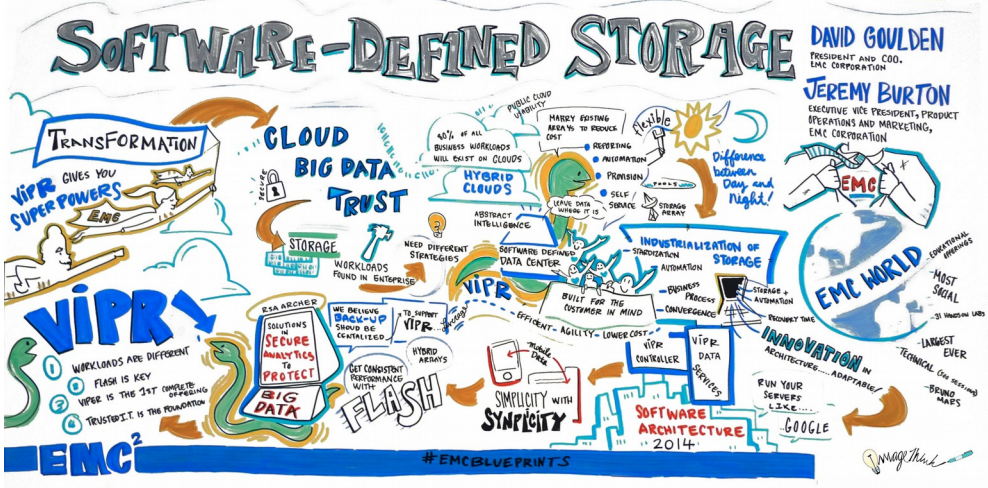
Data durability



Data durability



Data durability

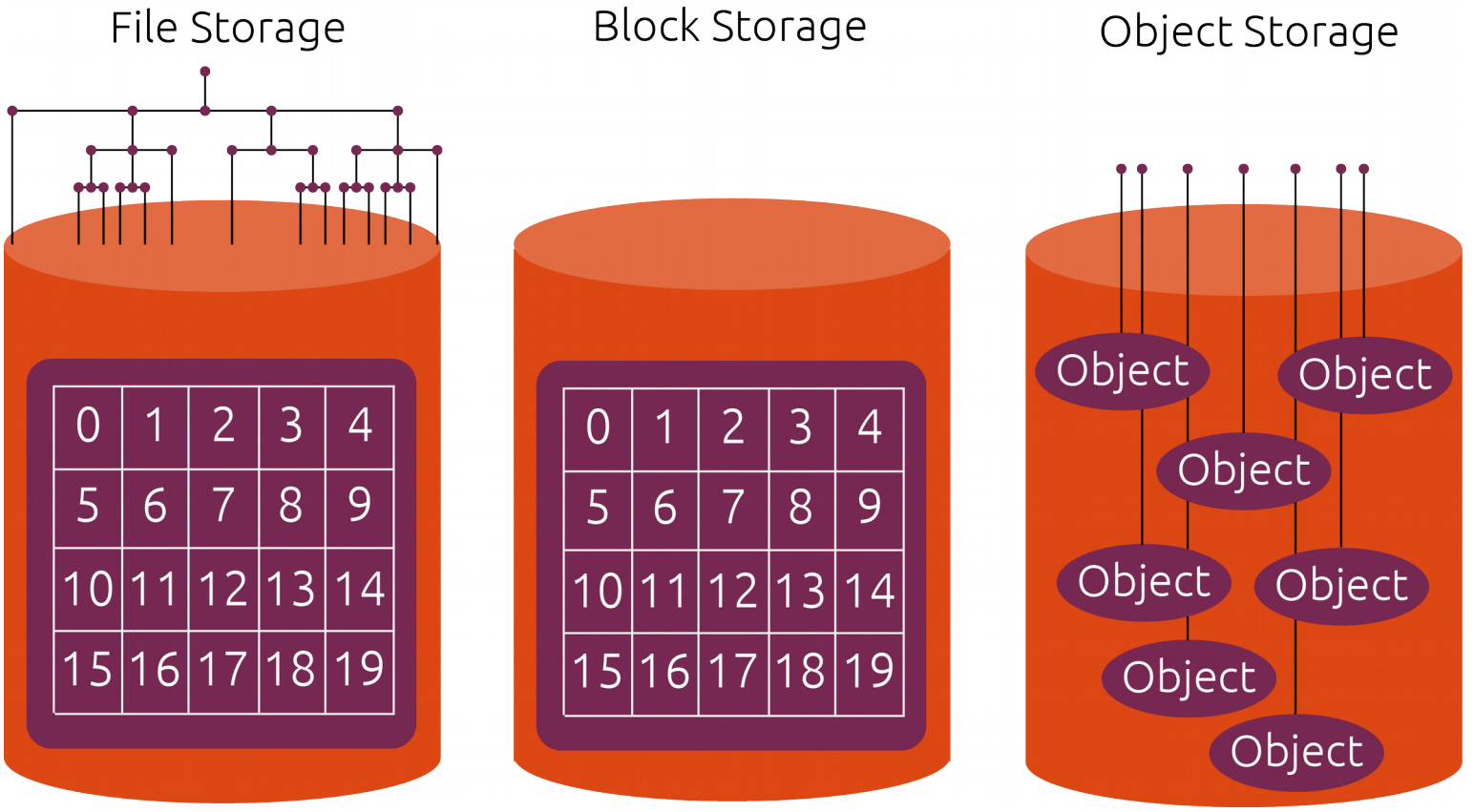


- Software Defined Storage
  - Auditting processes
  - Failures are handled by the storage system itself. No manual intervention required. Saves a lot of effort :)
  - No individual disk or node is responsible for the durability of the data
- CEPH, SWIFT, IBM Spectra Scale (declustered array), Netapp StorageGrid, Huawei OceanStor 9000, FUJITSU Storage ETERNUS CD10000 S2,…..
- Currently testing dCache with CEPH storage backend

CAP theorem

- CAP theorem (Eric Brewer)
  - Consistency
    (all nodes see the same data at the same time)
  - Availability
    (every request receives a response about whether it succeeded or failed)
  - Partition Tolerance
    (the system continues to operate despite arbitrary partitioning due to network failures)
- You can get only 2 out of 3
- SWIFT drops consistency to get availability, partition tolerance
- CEPH drops availability to get consistency and partition tolerance

Types of storage



SWIFT & CEPH

- SWIFT is only an object store and nothing else
- CEPH can be an object store (RGW), file-based storage (CEPHFS) and block storage (RBD)
- Both run on commodity hardware
- Both are Software Defined Storages
- Both have no SPOFs
- Both are self-healing

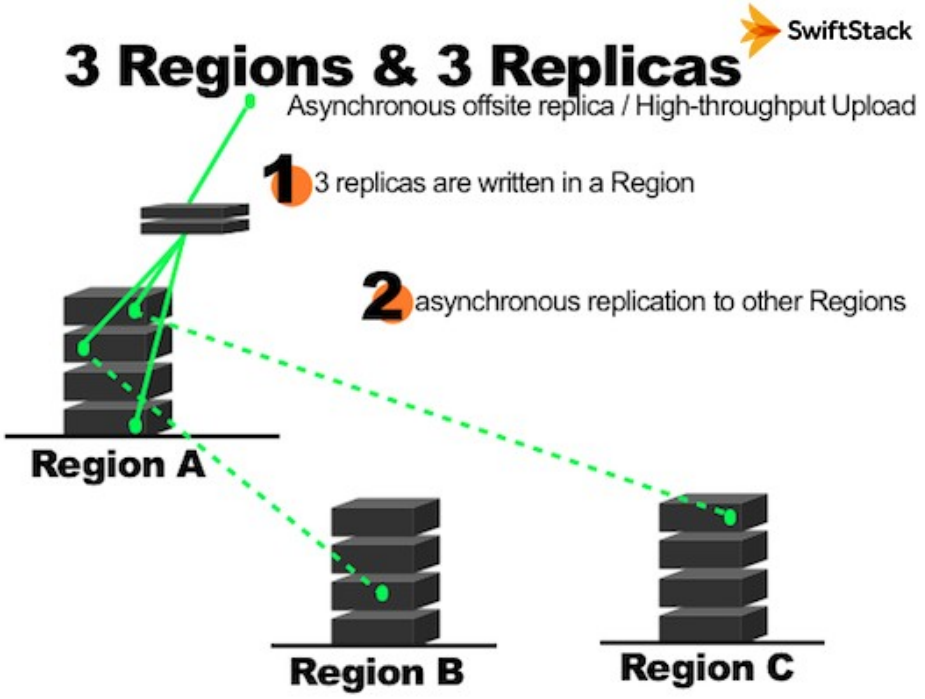
SWIFT

- CRUD – Create Read Update Delete
- Objects are accessible through an API (via URLs and https)
- Object locations as URLs for scalability of storage system
  - https://proxy.swift.surfsara.nl/v1/KEY_05b2aafab5a745eab2726d88649d95fe/mycontainer/myobject

SWIFT

- Unstructured data
  - Text, video, scientific data backups,websites,.....
- Highly available
- Eventual consistent
  - No transactional data
- Speaks its own SWIFT protocol and S3
- Massive scalability,
  SWIFT scales to the

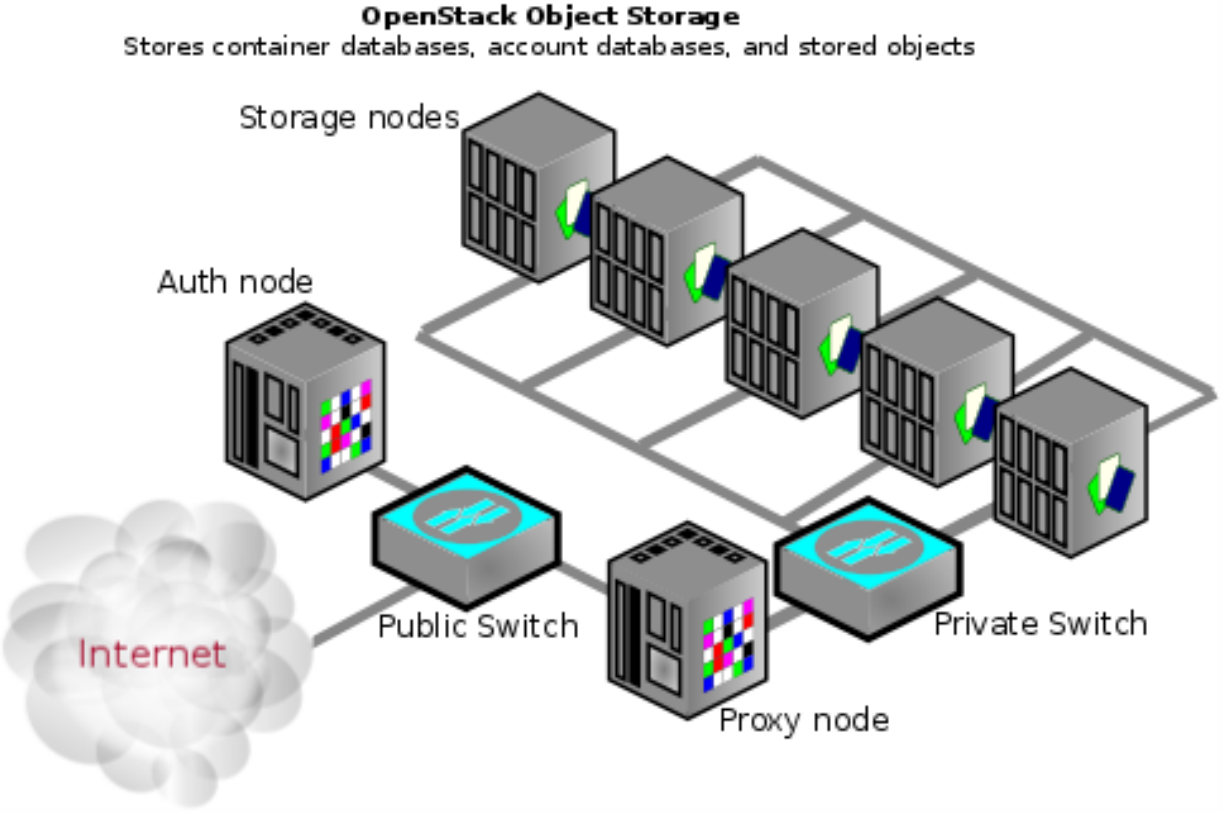
SWIFT


SwiftStack
3 Regions & 3 Replicas
Asynchronous offsite replica / High-throughput Upload
1
3 replicas are written in a Region
2
asynchronous replication to other Regions
Region A
Region B
Region C

SWIFT

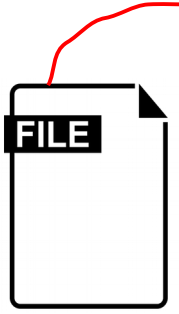
- Single name space
- Geographically distributed

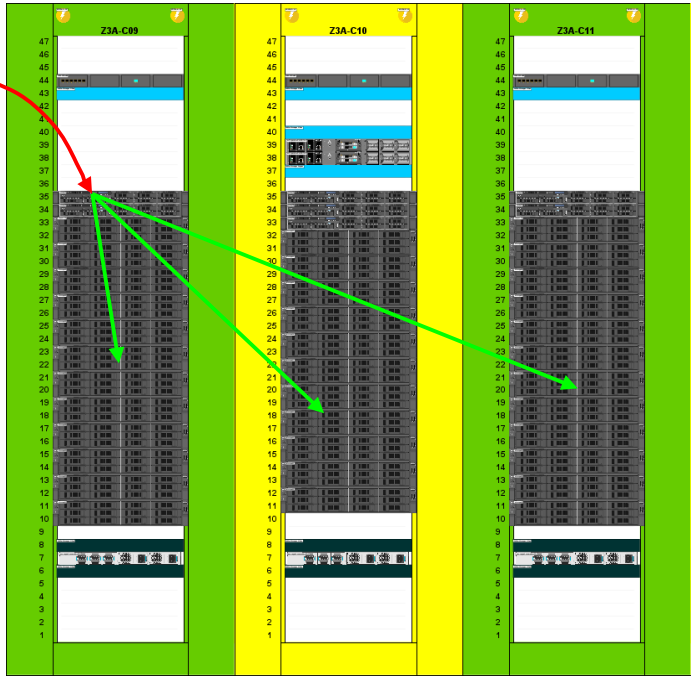
SWIFT


OpenStack Object Storage
Stores container databases, account databases, and stored objects
Storage nodes
Auth node
Internet
Public Switch
Private Switch
Proxy node

SWIFT

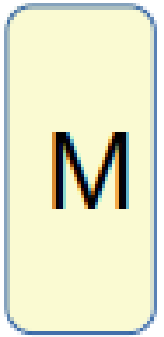
- Storage policies
  - 3 replica’s →
  - Also Erasure coding like 8+4
  - SSDs/HDDs
  - Geographic location



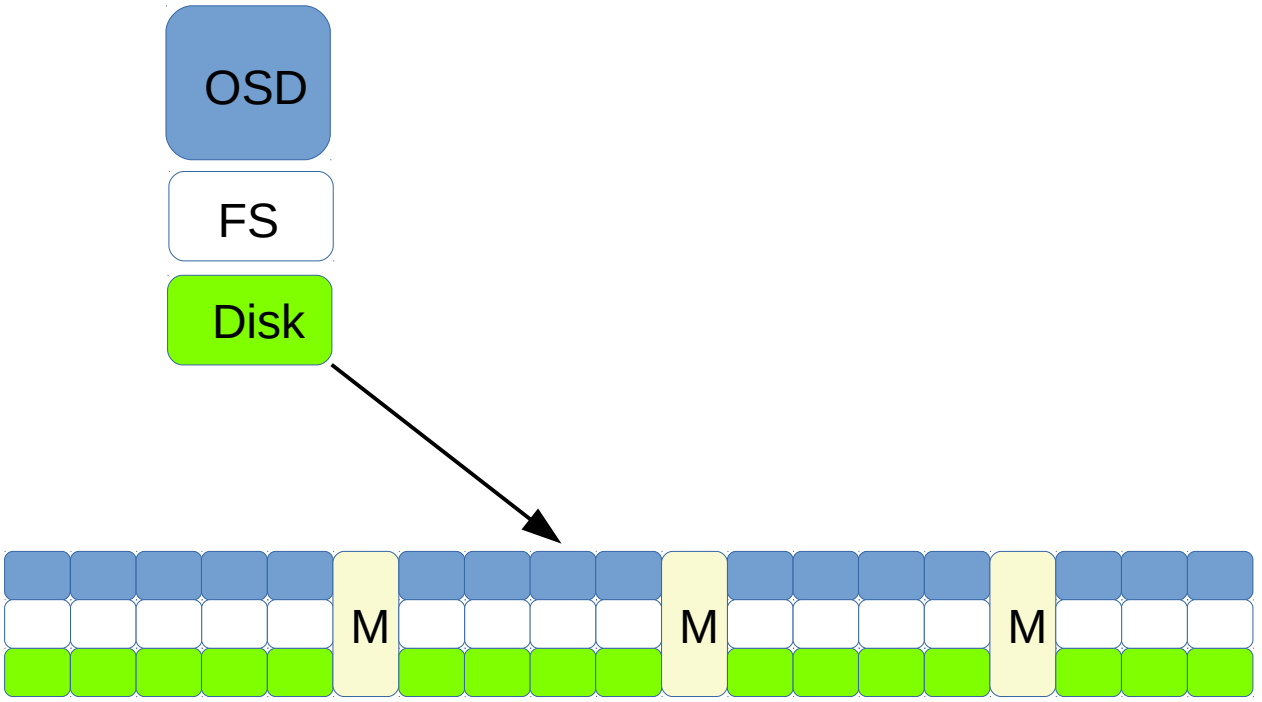


CEPH

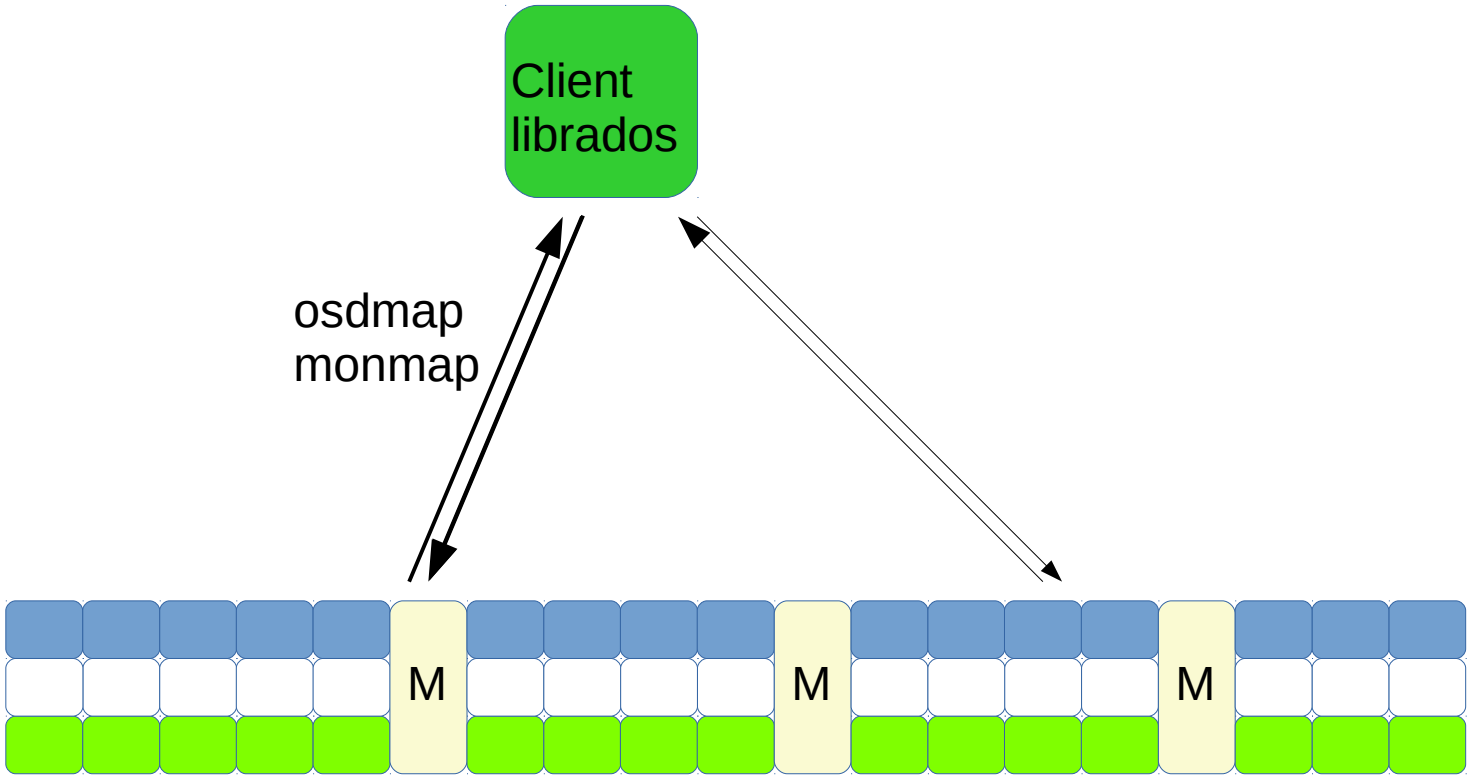
- CEPH components
  - Monitor/Manager
    - Management
    - Statistics
    - Consensus distributed decicion making
    - Cluster membership and state
    - Odd number
  - OSD
    - 1 per disk
    - Serves objects to clients
    - Replication and recovery



CEPH

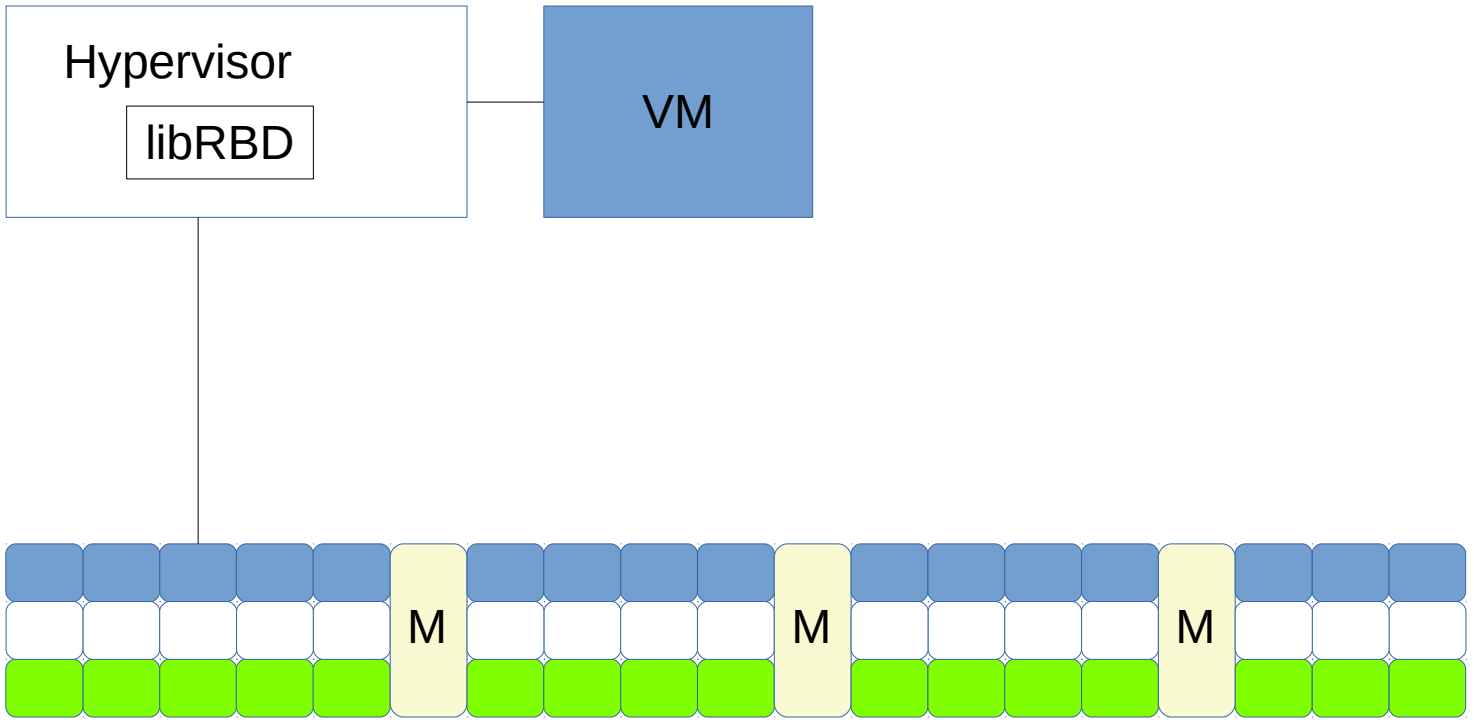


CEPH



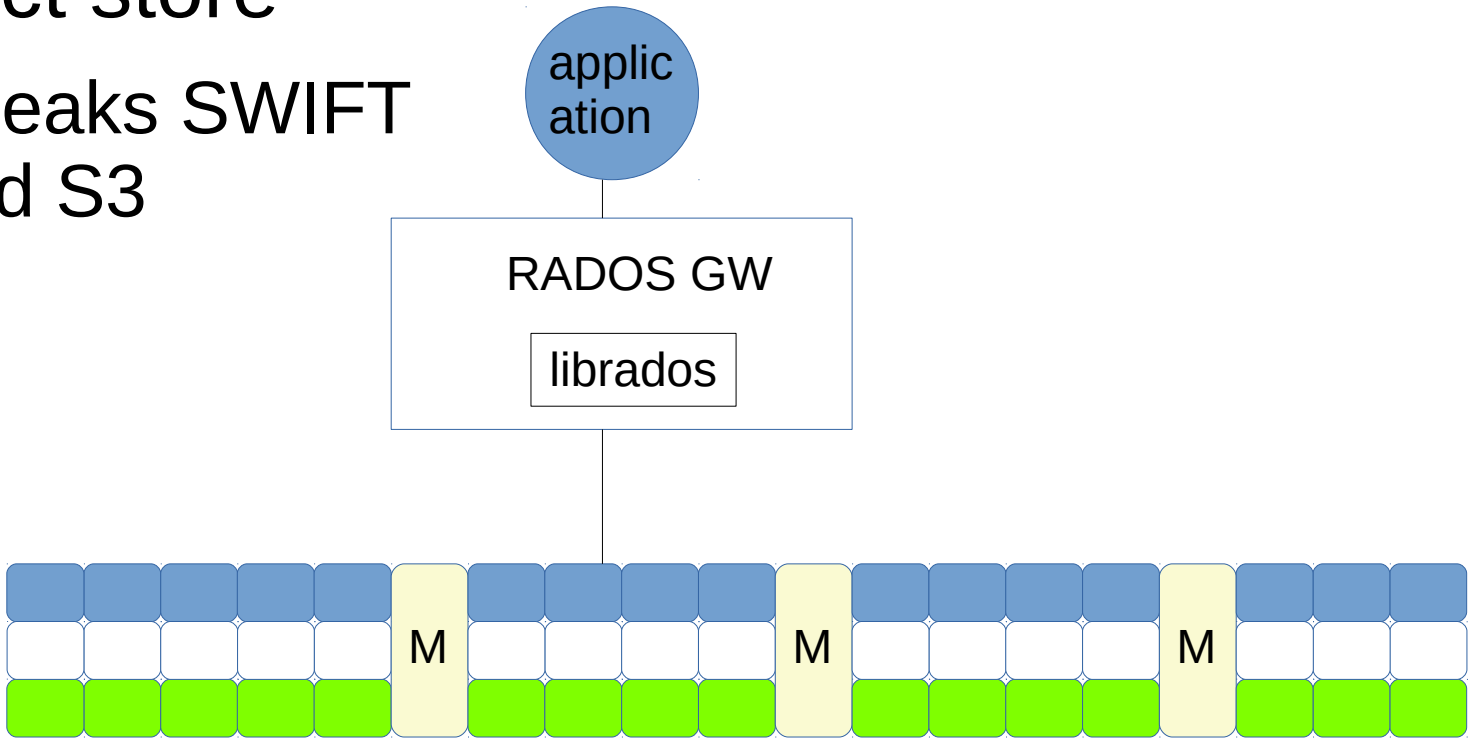
CEPH

- Block device



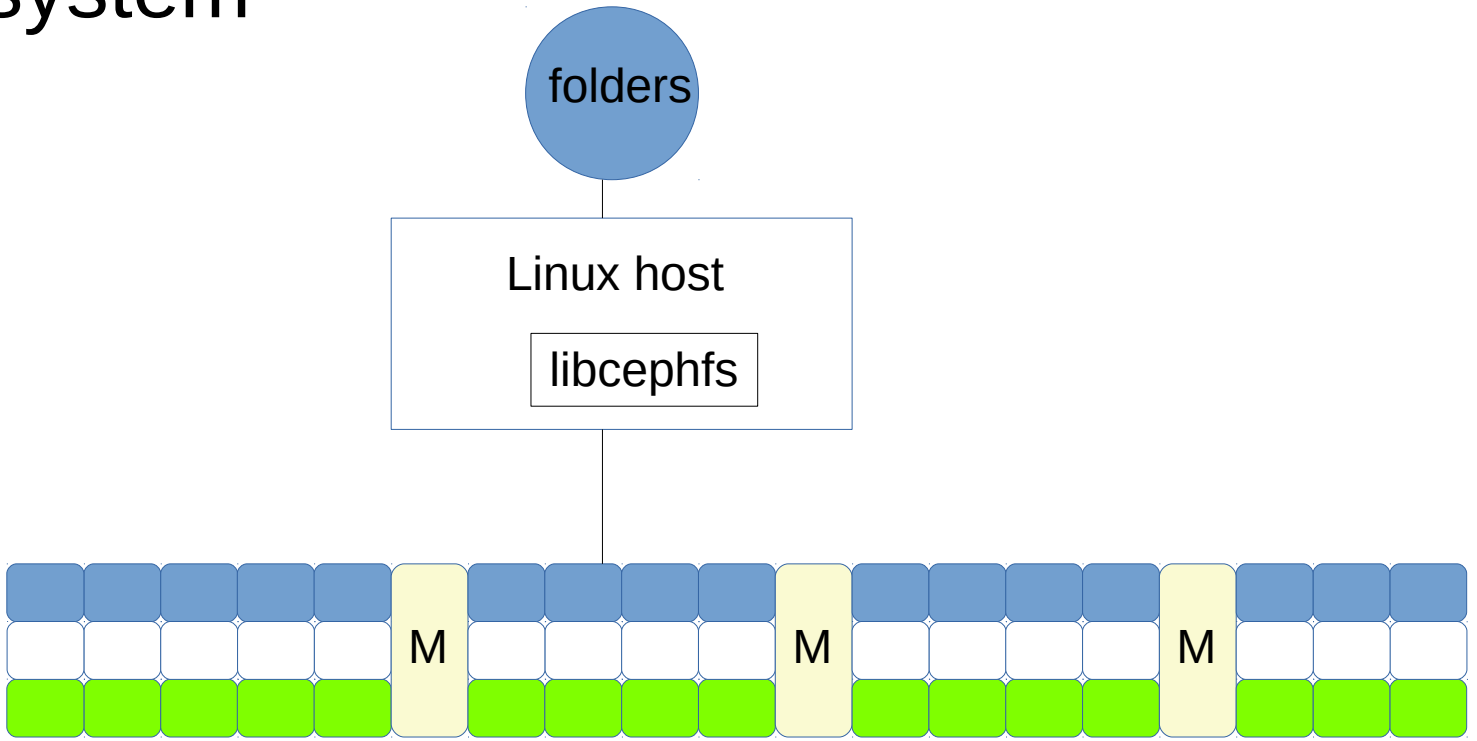
CEPH

- Object store
  - Speaks SWIFT and S3



CEPH

- File system



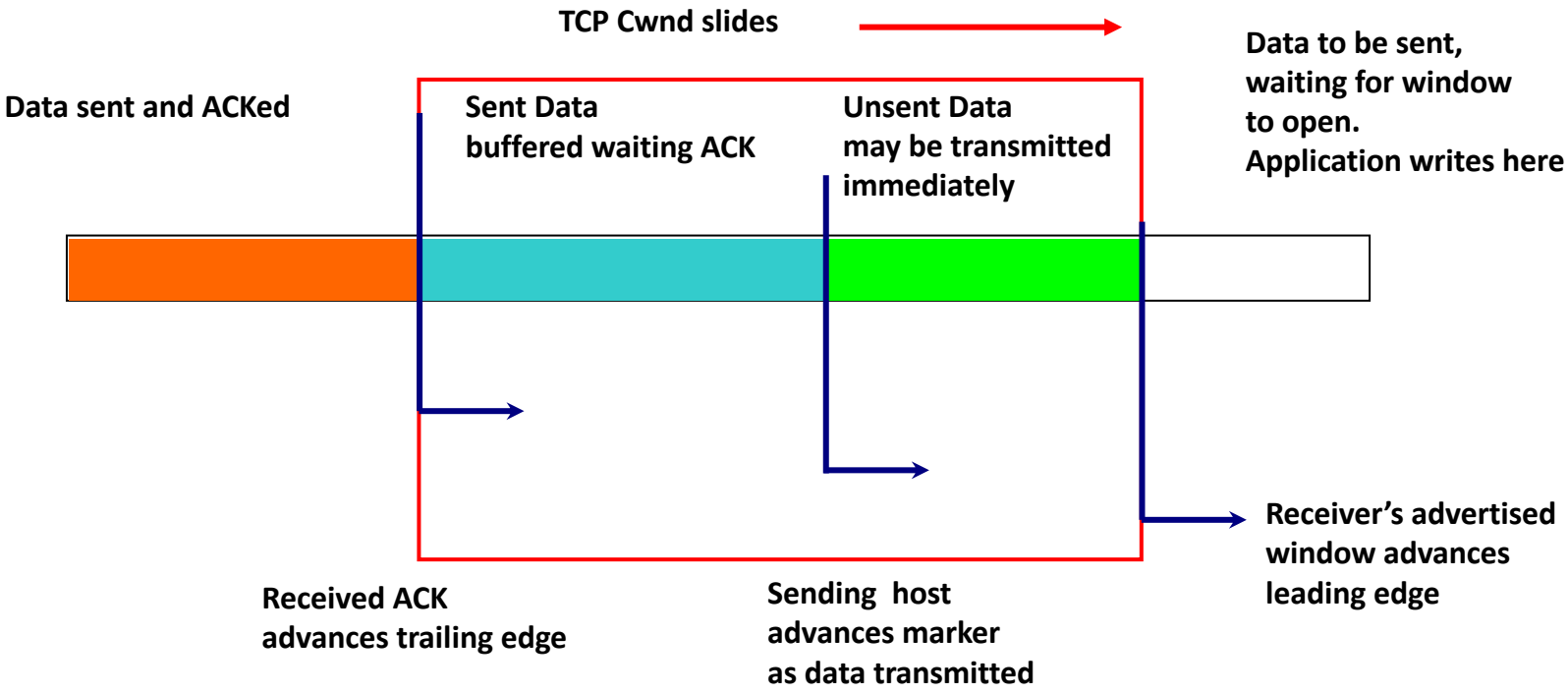
TCP Protocol

TCP/IP

- Most date transfers use TCP/IP
- TCP is a connection-oriented, reliable transport protocol
  - The data is presented to the remote user bit-wise correct
  - Positive acknowledgement (ACK) of each received segment (flow control)
    - Sender keeps record of each segment sent
    - Sender awaits an ACK – "I am ready to receive byte 2048 and beyond"
    - Sender starts timer when it sends segment – so can re-transmit
- Other TCP goals:
  - Prevent network overload (slow start) and "meltdown" (congestion avoidance)
  - Use the capacity efficiently
  - Share the available capacity fairly amongst the users
- TCP has worked well from ~1kbit/s to 100 Gbit/s BUT …
  - Packet loss taken is as indication of congestion causing TCP to back off
- This is a problem for high bandwidth long distance networks
- AND You need to tune TCP

TCP Flow Control: Sender – Congestion Window

- TCP uses a congestion window, cwnd, a sliding window to control the data flow
  - Byte count giving highest byte that can be sent with out without an ACK
  - Transmit buffer size and Advertised Receive buffer size important.
  - ACK gives next sequence no to receive AND
    The available space in the receive buffer.
  - Timer kept for each packet

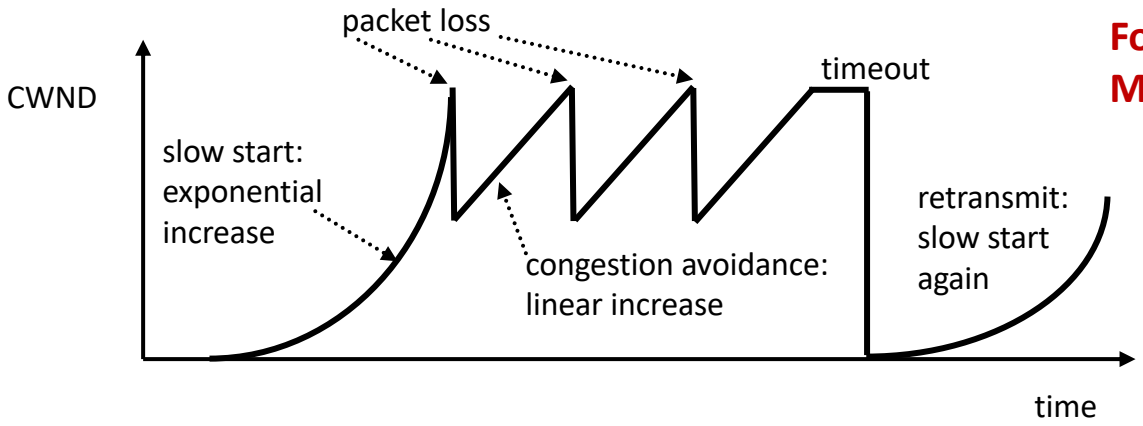


TCP Slowstart

- Probe the network - get a rough estimate of the optimal congestion window size
- The larger the window size, the higher the throughput
  - **Window size = Throughput * Round-trip Time [BDP in TCP tuning]**
- exponentially increase the congestion window size until a packet is lost
  - cwnd initially 1 MTU then increased by 1 MTU for each ACK received
    - Send 1st packet get 1 ACK increase cwnd to 2
    - Send 2 packets get 2 ACKs inc cwnd to 4
    - Time to reach cwnd size W = RTT*$\log_2$(W)
  - Rate doubles each RTT

Note on TCP tuning:
For 10 Gbit/s with 32 ms rtt need 40 MByte TCP buffer

For 10 Gbit/s trans-Atlantic need 190 MByte TCP buffer

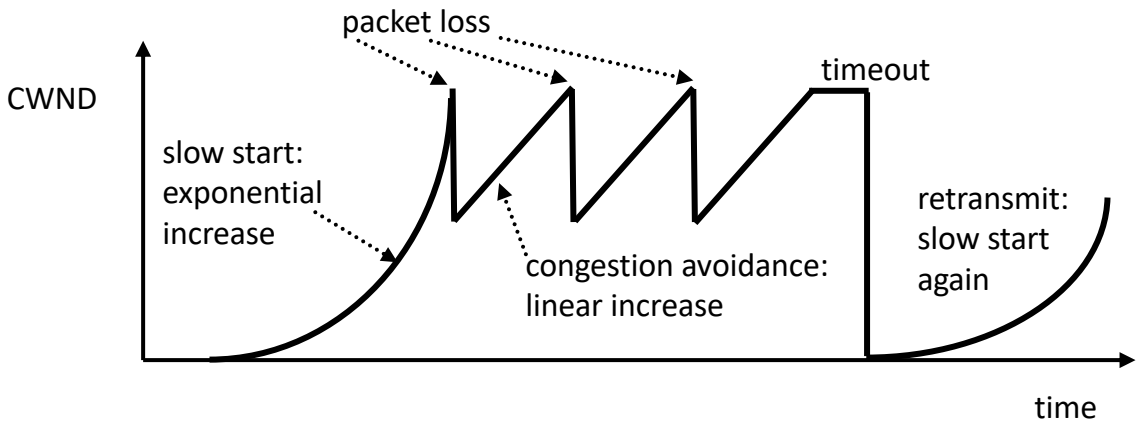


TCP AIMD Congestion Avoidance

- **additive increase:** starting from the rough estimate, linearly increase the congestion window size to probe for additional available bandwidth
  - cwnd increased by 1 /MTU for each ACK – linear increase in rate

  cwnd -> cwnd + a / *cwnd* ***- Additive Increase, a=1***
- TCP takes packet loss as indication of congestion !
- **multiplicative decrease:** cut the congestion window size aggressively if a packet is lost
  - Standard TCP reduces cwnd by 0.5

  cwnd -> cwnd – b (*cwnd*) ***- Multiplicative Decrease, b= ½***
  - Slow start to Congestion avoidance transition determined by ssthresh
- **Packet loss is a killer**

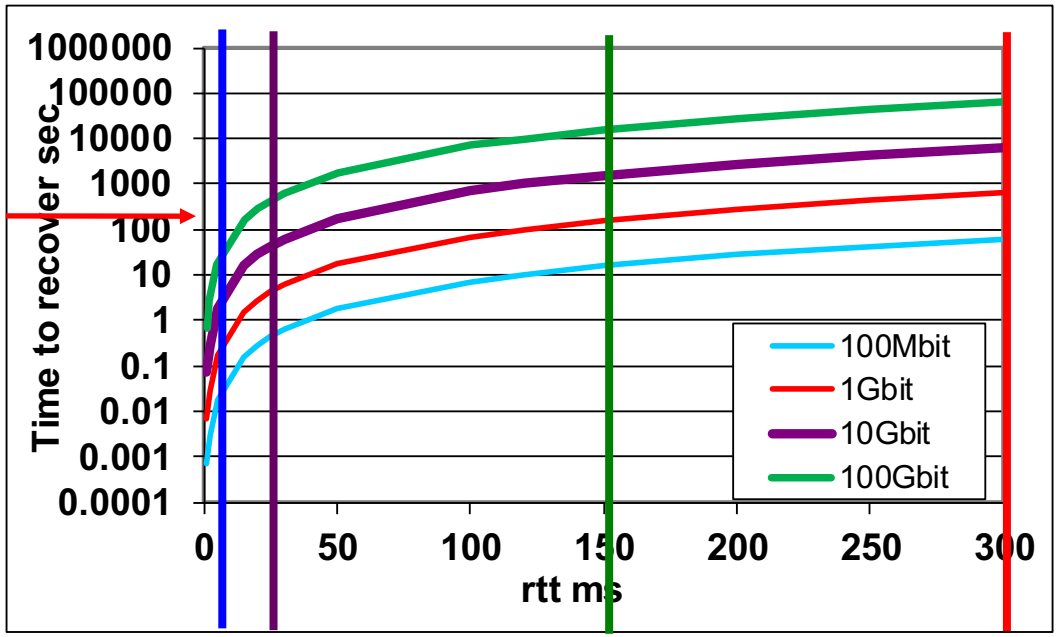


TCP (Reno) – Recovery Time

- The time for TCP to recover its throughput from 1 lost 9000 byte packet given by:

$$\rho = \frac{C * RTT^2}{2 * MSS}$$

2 min



- For 10 Gbit/s

UK 6 ms Europe 25 ms USA 150 ms Aus 300ms
2.5 s 43 s 26 min 104 min

Avoid Packet Loss

Tuning a DTN

Network Tuning for 100 Gigabit Ethernet

- **Hyper threading**
  - Turn off in the BIOS
- **Wait states**
  - Disable / minimise use of c-states. Use the BIOS and at boot time
- **Power saving Core Frequency**
  - Set governor “performance”
  - Set cpufreq to maximum
  - Depends on scaling_driver:

```
Read the current settings
$cat
/sys/devices/system/cpu/cpu*/cpufreq/cpuinfo_cur_freq
$cat
/sys/devices/system/cpu/cpu*/cpufreq/scaling_governor
Set
$echo “performance” >
/sys/devices/system/cpu/cpu*/cpufreq/scaling_governor
```

acpi-cpufreq allows setting cpuinfo_cur_freq to max
intel_pstate does not but seems fast anyway

Network Tuning for 100 Gigabit Ethernet

- **NUMA**
  - Check which cores are on which CPU socket & PCIe layout

```
$numactl -H
$cat /sys/devices/system/node/node*/cpulist
$lspci -tv
$cat /sys/class/net/*/device/uevent
```

  - Check which CPU cores are attached to the NIC.

```
$ls /sys/class/net/
$cat /sys/class/net/enp131s0f1/device/local_cpulist
```

- **IRQs**
  - Turn off the irqbalance service
    - prevents balancer from changing the affinity scheme.

```
#systemctl stop irqbalance.service
#systemctl disable irqbalance.service
```

  - Set affinity of the NIC IRQs to use CPU cores on the node with PCIe to NIC
    - 1 per CPU.
    - For UDP seems best NOT to use the CPU cores used by the apps.

```
#cat /proc/irq/<irq>/smp_affinity
#echo 400 > /proc/irq/183/smp_affinity
#/usr/sbin/show_irq_affinity_cpulist.sh enp131s0f0
#/usr/sbin/set_irq_affinity_cpulist.sh 8-11 enp131s0f0
```

Network Tuning for 100 Gigabit Ethernet

- **Interface parameters**
  - Ensure interrupt coalescence is ON – 3 µs, 8 µs, 80 µs, more ?
  - Ensure Rx & Tx checksum offload is ON
  - Ensure tcp-segmentation-offload is ON
  - Set the Tx Rx ring buffer size

```
#ethtool -C <i/f> rx-usecs 8 or 80
#ethtool -K <i/f> rx on tx on
#ethtool -K <i/f> tso on
#ethtool -G <i/f> rx 8192
#ethtool -G <i/f> tx 8192
```

- **MTU**
  - Set IP MTU 9000 Bytes

```
Best set in files eg ifcfg_ethx
mtu=9000
```

- **Firewall**
  - Check it is on and allows the correct ports

```
# systemctl status firewalld.service
```

- **Routing**
  - Check you are using the NIC you expect

```
$ route -en
Files /etc/sysconfig/network-scripts/route-<NIC>
```

Network Tuning for 100 Gigabit Ethernet

- **Queues**
  - Set txqueuelen
    – transmit Q (I used 1000 but 10,000 recommended)
  - Set netdev_max_backlog – say 250000
    – Q between interface and IP stack
- **Kernel parameters**

```
Best in file /etc/sysctl.conf
```

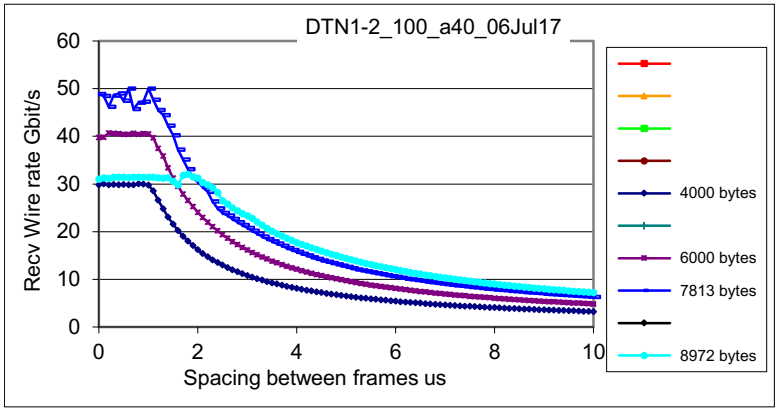
  - net.core.rmem_max net.core.wmem_max
  - net.ipv4.tcp_rmem net.ipv4.tcp_wmem (min / default / max)
  - net.ipv4.tcp_mtu_probing (jumbo frames)
  - net.ipv4.tcp_congestion_control (htcp, cubic)
  - net.ipv4.tcp_mem (set the max to cover rmem/wmem max)
- **Set the affinity of the applications**
  - Using the correct core has a big effect.
- **Better to choose fewer high speed cores**
  - AENEAS Deliverable 4.1 https://drive.google.com/file/d/1-IQ0psShLcJPgKIZTxlR1rVkogAQTGMo/view
  - http://www.mellanox.com/related-docs/prod_software/Performance_Tuning_Guide_for_Mellanox_Network_Adapters.pdf
  - Esnet FasterData https://fasterdata.es.net/network-tuning/

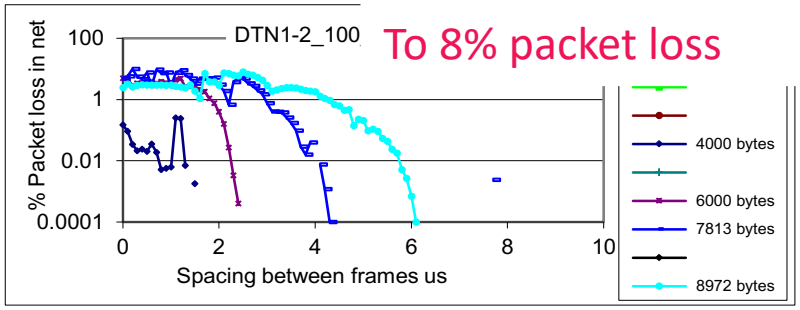
Some Effects of Tuning

udpmon: Size of Rx Ring Buffer

- ConnectX-5, set affinity of udpmon to core 6.
- Use ethtool -S <enp131s0f0> look at rx_out_of_buffer

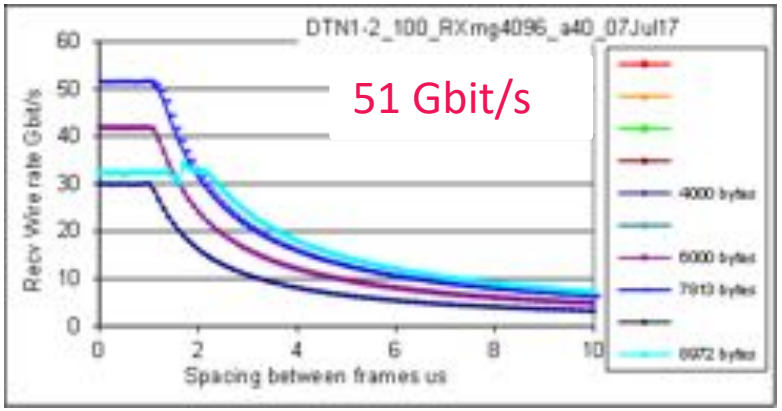
RX ring 1024





To 8% packet loss

RX ring ≥4096

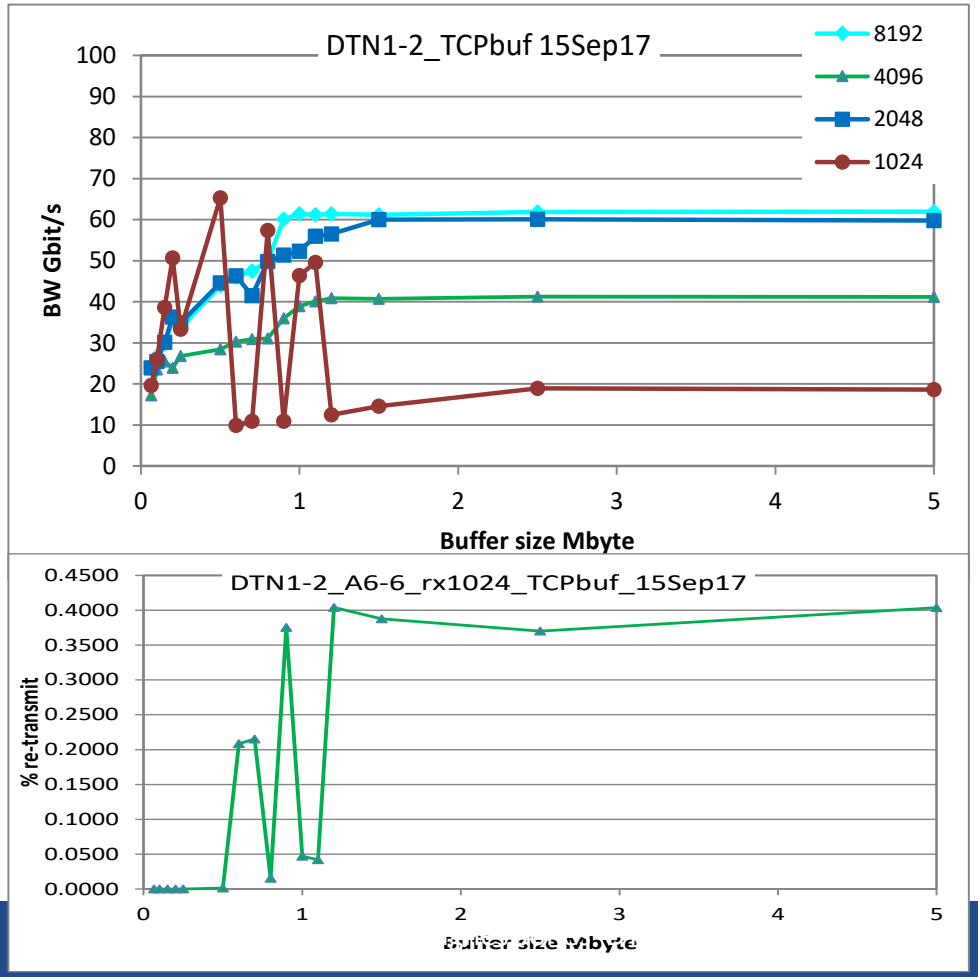


51 Gbit/s

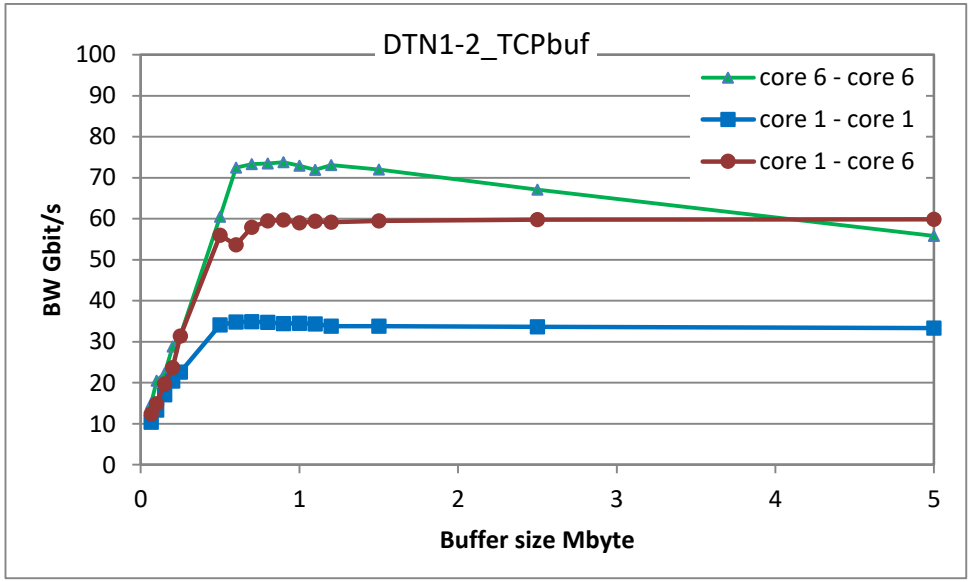
No packet loss

TCP Throughput iperf2 effect of Rx Ring Buffer

- ConnectX-5, iperf Core6 – core6
- Correlation of low throughput and re-transmits for Rx ring 1024



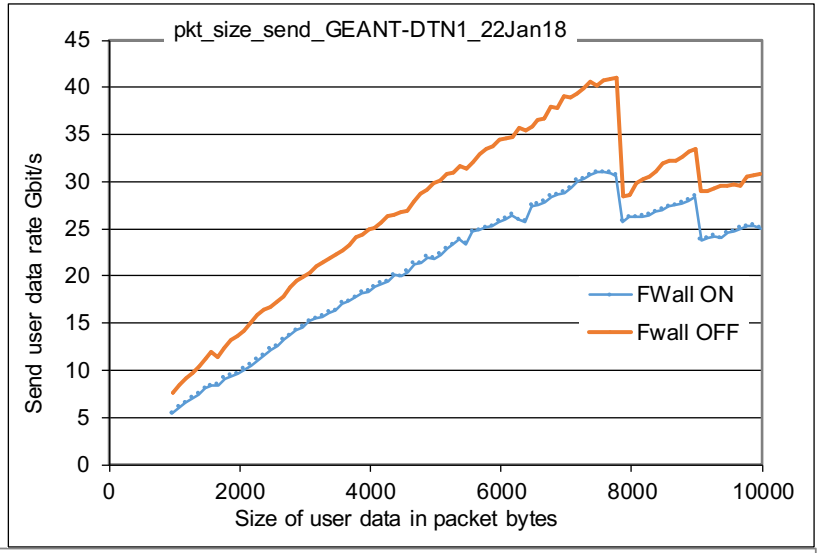
iperf3: TCP Throughput
Using different CPU cores and Nodes



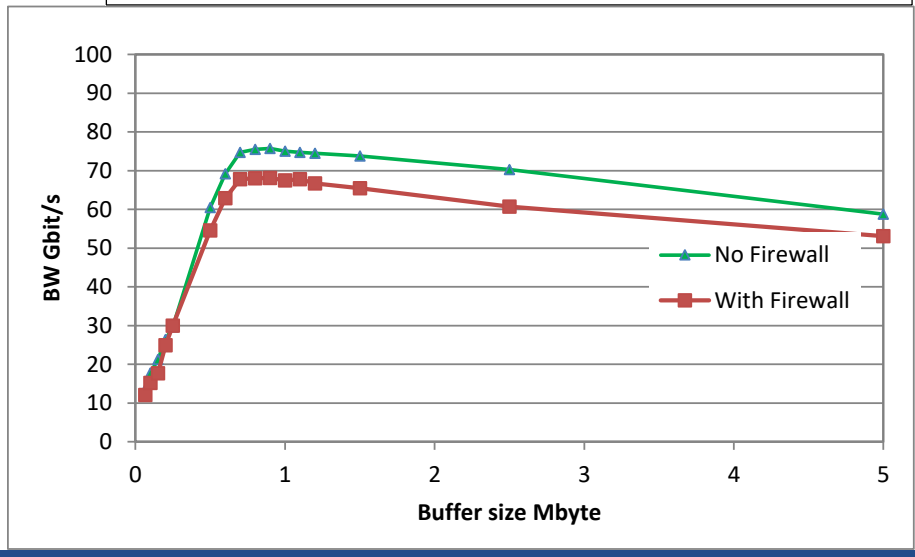
- Firewalls OFF, TCP offload on, TCP cubic stack
- RTT 0.4 ms.
- Delay Bandwidth Product 0.5 MB.
- Rises smoothly to the plateau at 0.5 MBytes.
- Throughput:
  - 75 Gbit/s Both send & receive on node 1
  - 60 Gbit/s Send on node 0 receive on node 1
  - 35 Gbit/s Both send & receive on node 0
- Very few TCP re-transmitted segments observed

The effect of Firewalls

- Run udpmon_send on core 6
- Move IRQs from core 6.
- ConnectX-5 NICs Rx ring buffer 4096
- Send rate vs packet size
- Effect of firewall ~10 Gbit/s reduction



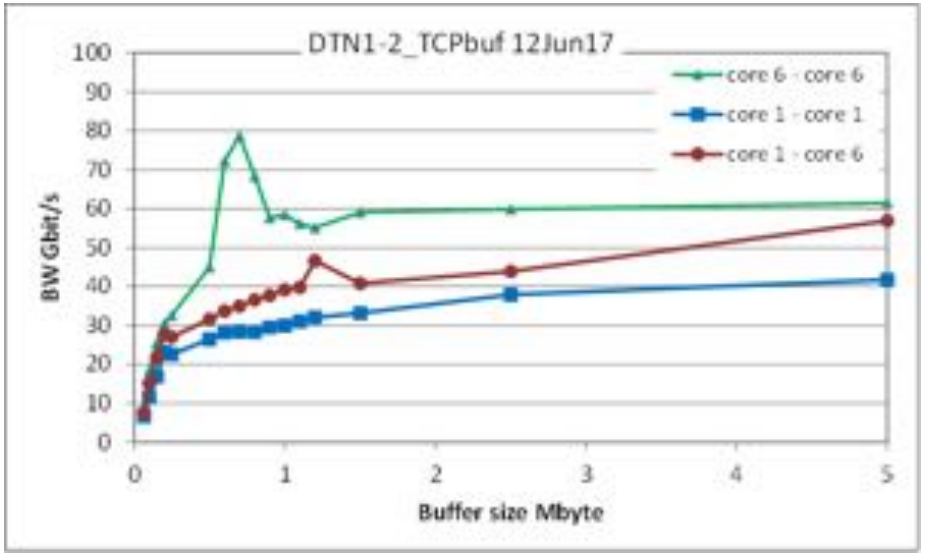
- Run iperf3 on core 6,
  TCP offload on, TCP cubic stack
- RTT 0.4 ms. DBP 0.5 MBytes.
- Rises smoothly plateau at 0.5 Mbytes
- Achievable throughput falls by 7.3 Gbit/s
- No TCP re-transmitted segments



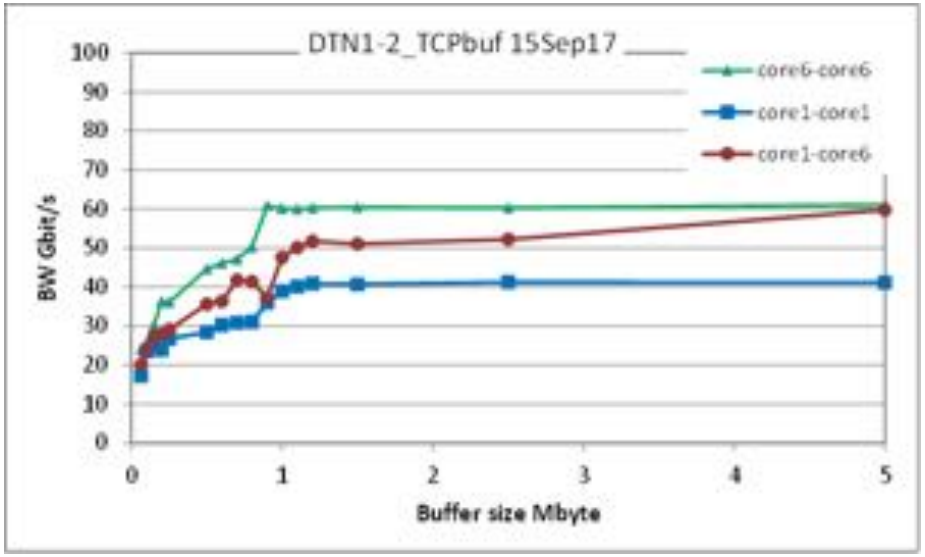
TCP Test Program: Throughput iperf2 & iperf3

- ConnectX-5, NIC rx buffer 4096,
- Iperf core6 – core6
- While transmitting at 80 Gbit/s the CPU was 98% in kernel mode.

iperf3



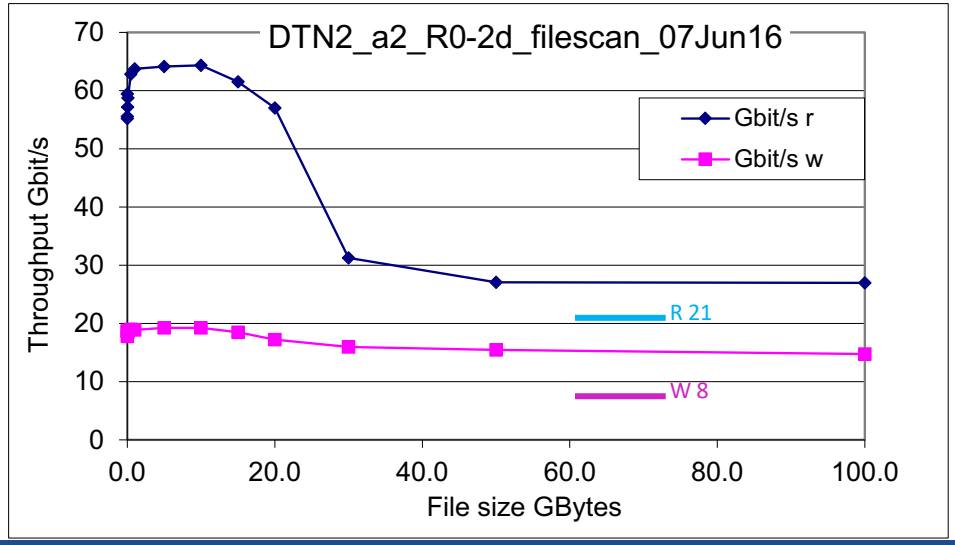
iperf2

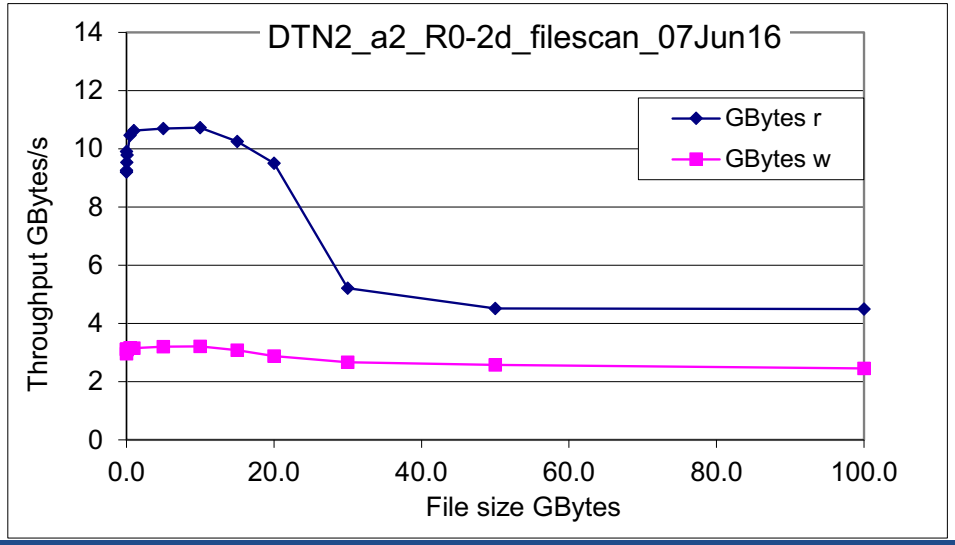


Performance of NVMe disks on the GÉANT DTN

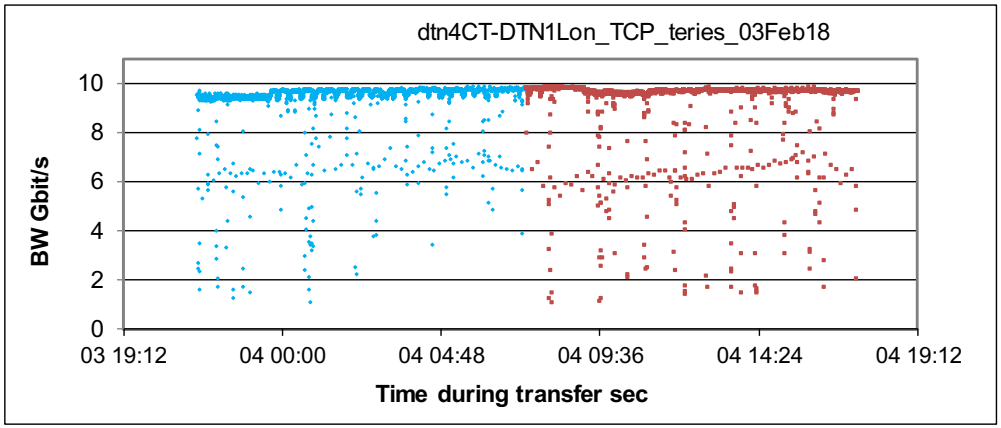
- IRQs distributed over all cores on both nodes
- Run disk_test on core 2 Node 0
- Measure sequential read and write disk-memory rates as function file size
- 2 disks in RAID0 xfs file system

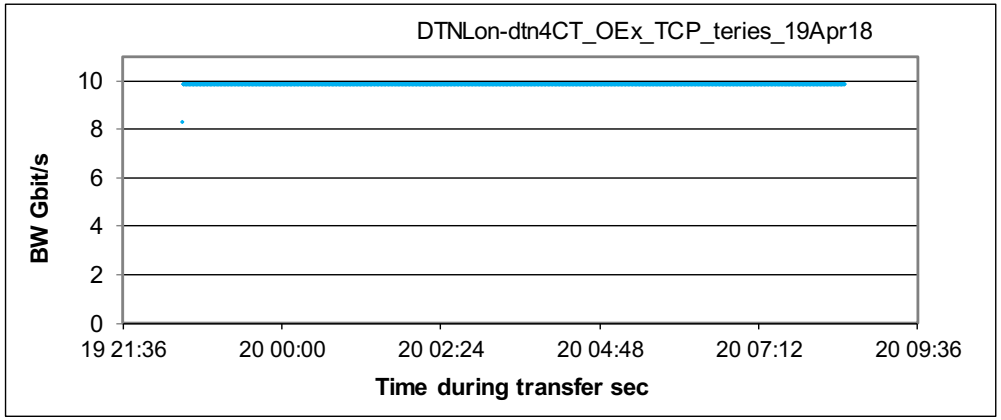
	Read Gbit/s	Write Gbit/s	
1 Disk	~6.2	12.5	yes read < write !
RAID0 2disks	27	15.5	





10 Gigabit TCP: SANReN Cape Town to GÉANT London



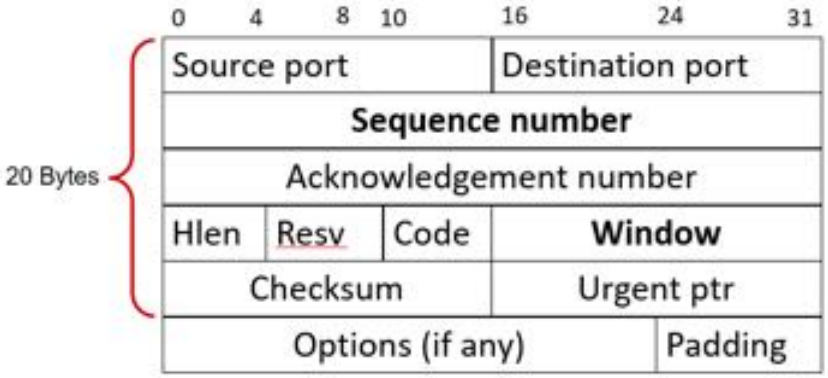


- **Production routed IP path**
- Achievable TCP throughput over 20 Hrs
- Peak 9.5 Gbit/s
- RTT 142 ms
- BDP 178 MBytes for 10 Gig

- **Direct link Open exchanges**
- Achievable TCP throughput over 10 Hrs
- Peak 9.9 Gbit/s
- No TCP re-transmits
- Representative of SKA path on WACS cable

The TCP Protocol Limit

TCP Header



2^{32} → 4096 MB

2^{16} → 64 kB

- To fix the Window size there is the Window Scale factor negotiated at the SYN exchange. RFC 7323 (obsoletes 1323)
- Max value 14 → max Window ($2^{16} + 2^{14}$) → 1024 MB
- Window size < Sequence number
  - Deal with sequence number wrapping – every 0.33s
  - Allow to tell if a segment is old or new

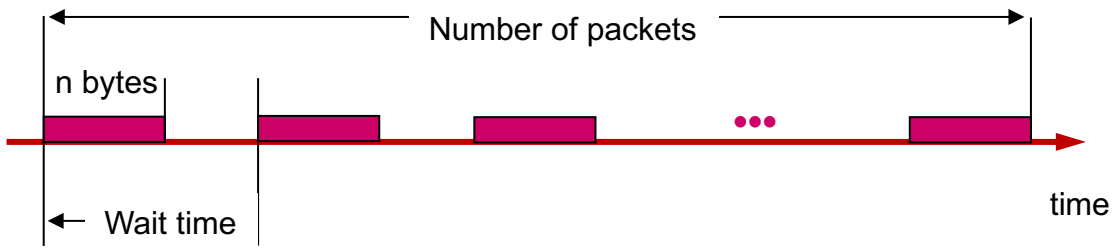
High Performance Data Transfers
What is important?

- The data moving application and protocols
  - Data movement – file transfer / “record access” / data flow topology
  - The use of TCP or UDP – staged transfers or real time flows
- Host performance
  - Hardware / VM configuration
  - Tuning the network stack and kernel parameters
  - Locking the application to a CPU – setting affinity
  - Interrupt handling and load balancing
- Check the performance of the network elements:
  end-host – work group – campus – access links - backbones
  - No traffic bottlenecks
  - No Packet loss
  - Available bandwidth meets requirements
  - Stability
- Don’t forget the Disk sub-system performance

Measuring Network Performance

- Common tools to measure along the path used to send the data
  - ping <host>
  - traceroute <host> both directions to check the path
  - iperf, iperf3, udpmon
- Network Characteristics to observe:
  - Utilisation of the links – Cacti, MRTG, Nagios, …
  - End-to-end routes
  - TCP & UDP achievable throughput
  - Packet loss
  - Latency
  - Packet jitter
  - Light levels
  - Network availability
  - Network stability

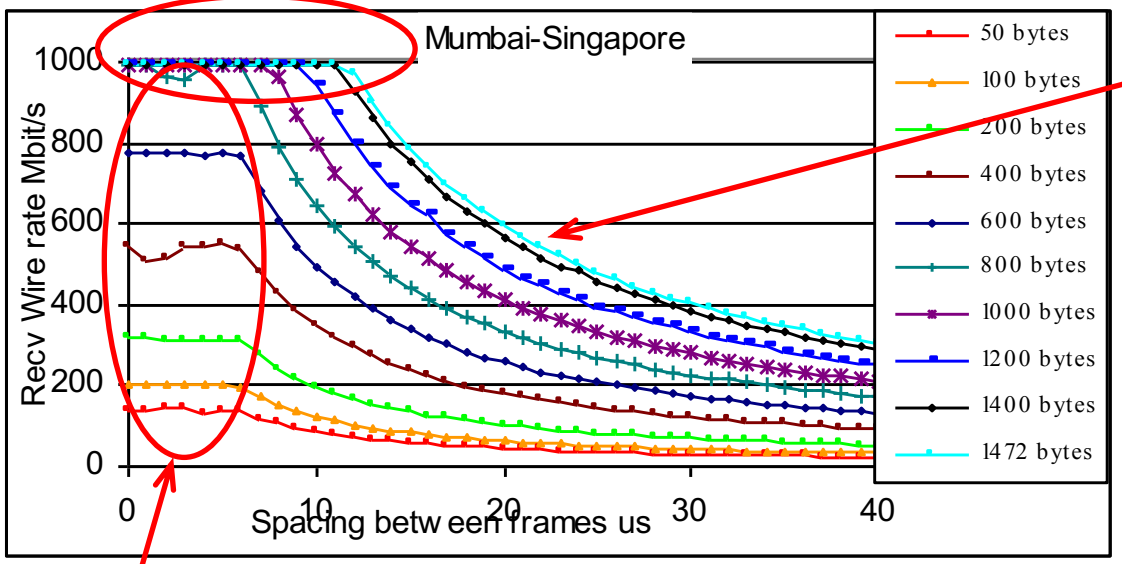
udpmon



- Programs work in client-server pairs (with set affinity) to:
  - Send a controlled stream of UDP frames spaced at regular intervals with 64 bit sequence numbers & send time stamp.
  - Can vairy frame size and frame transmit spacing.
  - Count the packets received and check the sequence & timing of the packets.
  - Identify if packets lost in the end host or network.
  - CPU load on end hosts
- Allows measurement of:
  - Achievable UDP bandwidth,
  - Packet loss, packet ordering, packet jitter histogram inter-packet arrival times
  - Relative 1-way delay, Packet dynamics & packet loss patterns.
  - Quality of the connection path and its stability.

End Hosts: UDP achievable throughput Ideal shape

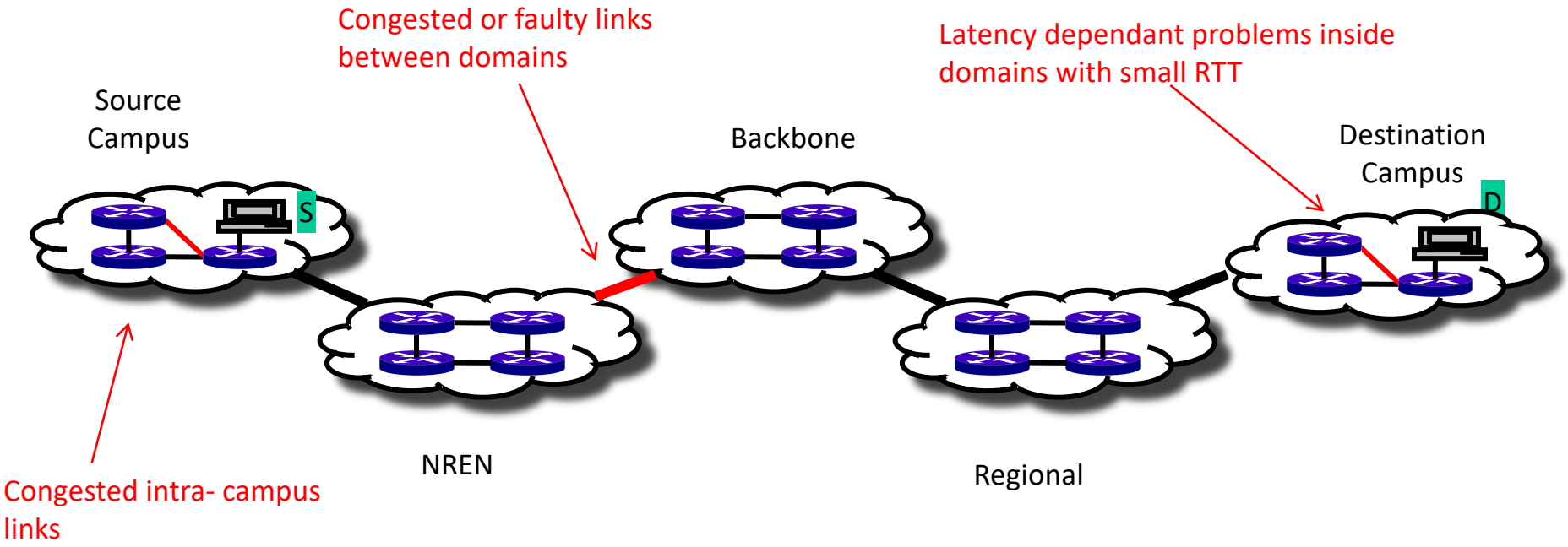
Flat portions
Limited by capacity of link
Available BW on a loaded link



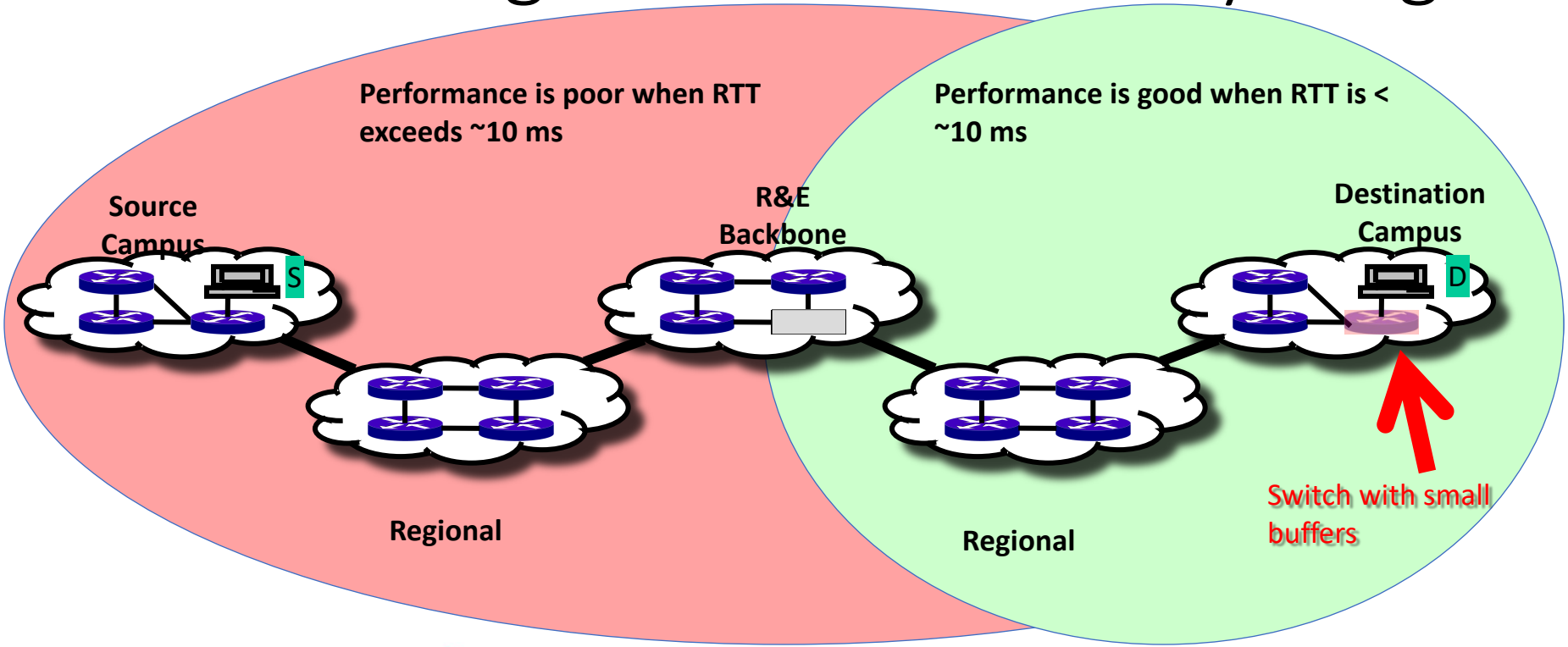
Shape follows 1/t
Packet spacing most important.

Cannot send packets back-2-back
End host: NIC setup time on PCI / context switches

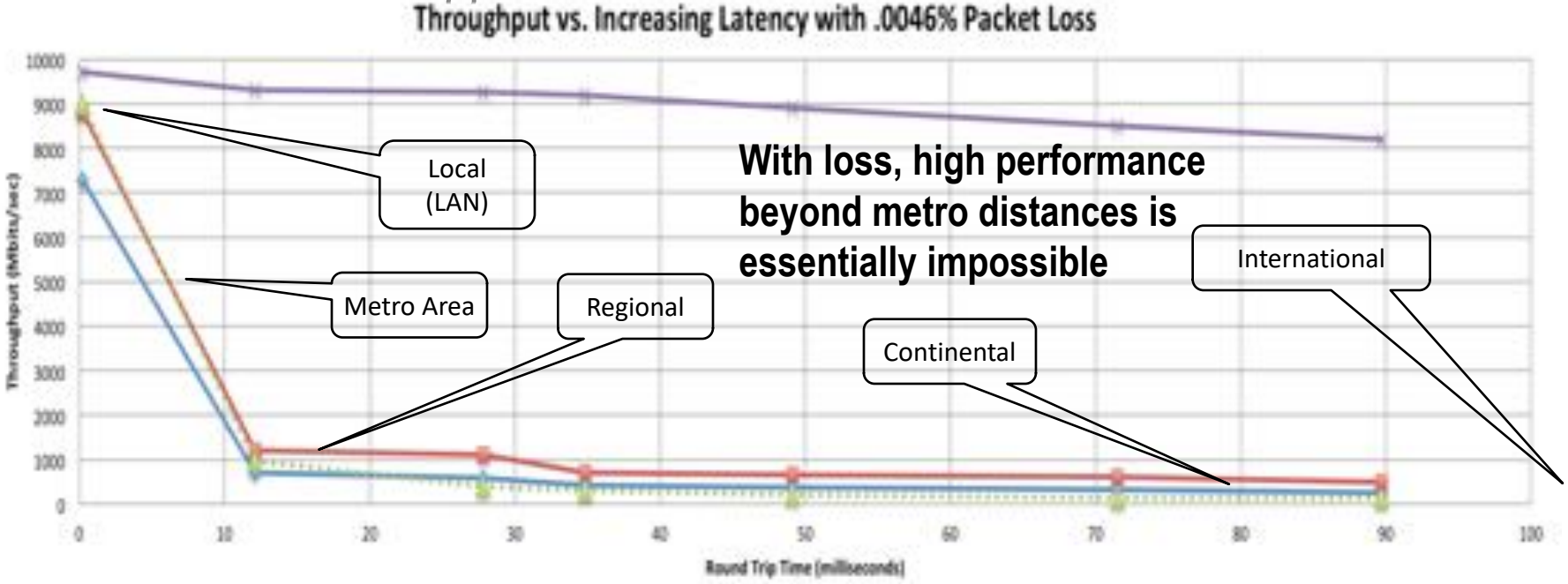
Where Are The Problems?



Local Testing Will Not Find Everything



Soft Failures Cause Packet Loss and Degraded TCP Performance



Soft Network Failures

- Soft failures are where basic connectivity functions, but high performance is not possible.
- TCP was intentionally designed to hide all transmission errors from the user:
  - "As long as the TCPs continue to function properly and the internet system does not become completely partitioned, no transmission errors will affect the users." (From IEN 129, RFC 716)
- Some soft failures only affect high bandwidth long RTT flows.
- Hard failures are easy to detect & fix
  - soft failures can lie hidden for years!
- One network problem can often mask others

Problem Statement: Hard vs. Soft Failures

perfSONAR

- "Hard failures" are the kind of problems every organization understands
  - Fiber cut
  - Power failure takes down routers
  - Hardware ceases to function
- Classic monitoring systems are good at alerting hard failures
  - i.e., NOC sees something turn red on their screen
  - Engineers paged by monitoring systems
- "Soft failures" are different and often go undetected
  - Basic connectivity (ping, traceroute, web pages, email) works
  - Performance is just poor
- How much should we care about soft failures?

ESnet ENERGY SCIENCES NETWORK GÉANT INDIANA UNIVERSITY INTERNET2 UNIVERSITY OF MICHIGAN

Causes of Packet Loss

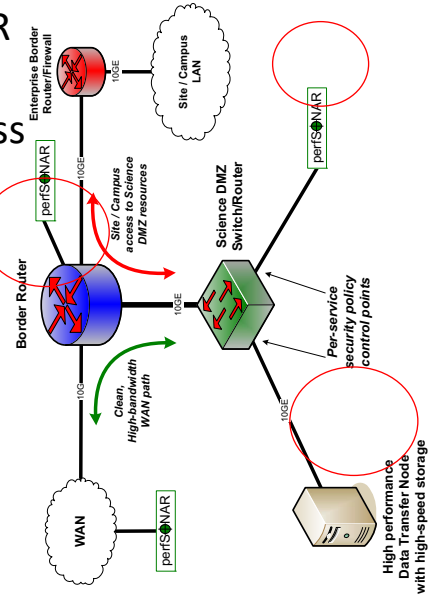
- Network Congestion
  - Easy to confirm via SNMP, easy to fix with $$
  - This is not a ‘soft failure’, but just a network capacity issue
  - Often people assume congestion is the issue when it fact it is not.
- Under-buffered switch dropping packets
  - Hard to confirm
- Under-powered firewall dropping packets
  - Hard to confirm
- Dirty fibers or connectors, failing optics/light levels
  - Sometimes easy to confirm by looking at error counters in the routers
- Overloaded or slow receive host dropping packets
  - Easy to confirm by looking at CPU load on the host

Network Monitoring

- All networks do some form monitoring.
  - Addresses needs of local staff for understanding state of the network
    - Would this information be useful to external users?
    - Can these tools function on a multi-domain basis?
- Beyond passive methods, there are active tools.
  - E.g. often we want a 'throughput' number. Can we automate that idea?
  - Wouldn't it be nice to get some sort of plot of performance over the course of a day? Week? Year? Multiple endpoints?
- perfSONAR = Measurement Middleware

perfSONAR

- All the previous Science DMZ network diagrams have little perfSONAR boxes everywhere
  - The reason for this is that consistent behavior requires correctness
  - Correctness requires the ability to find and fix problems
    - ***You can't fix what you can't find***
    - ***You can't find what you can't see***
    - ***perfSONAR lets you see***
- Especially important when deploying high performance services
  - If there is a problem with the infrastructure, need to fix it
  - If the problem is not with your stuff, need to prove it
    - Many players in an end to end path
    - Ability to show correct behavior aids in problem localization



What is perfSONAR?

- perfSONAR is a tool to:
  - Set network performance expectations
  - Find network problems (“soft failures”)
  - Help fix these problems
  - All in multi-domain environments
- These problems are all harder when multiple networks are involved
- perfSONAR is provides a standard way to publish active and passive monitoring data
  - This data is interesting to network researchers as well as network operators

ESnet ENERGY SCIENCES NETWORK GÉANT INDIANA UNIVERSITY INTERNET2 UNIVERSITY OF MICHIGAN

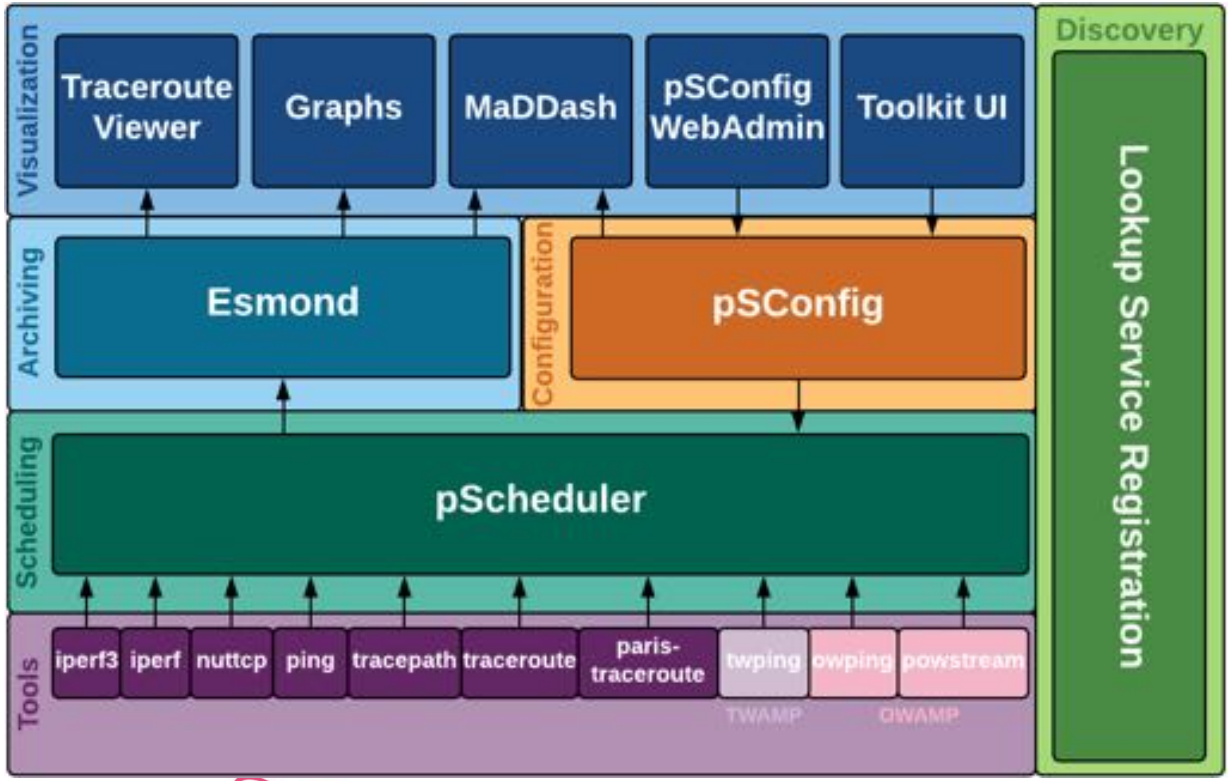
Simulating Performance

- It's infeasible to perform at-scale data movement all the time – as we see in other forms of science, we need to rely on simulations
- Network performance comes down to a couple of key metrics:
  - Throughput (e.g. "how much can I get out of the network")
  - Latency (time it takes to get to/from a destination)
  - Packet loss/duplication/ordering (for some sampling of packets, do they all make it to the other side without serious abnormalities occurring?)
  - Network utilization (the opposite of "throughput" for a moment in time)
- We can get many of these from a selection of measurement tools – enter the perfSONAR Toolkit

perfSONAR Toolkit

- The “perfSONAR Toolkit” is an open source implementation and packaging of the perfSONAR measurement infrastructure and protocols
  - http://docs.perfsonar.net/install_getting.html
- All components are available as RPMs, DEBs, and bundled as CentOS 7, Debian 8,9 or Ubuntu 14,16,18 -based packages (as for perfSONAR v. 4.1.2)
  - perfSONAR tools are much more accurate if run on a dedicated perfSONAR host
- Very easy to install and configure
  - Usually takes less than 30 minutes

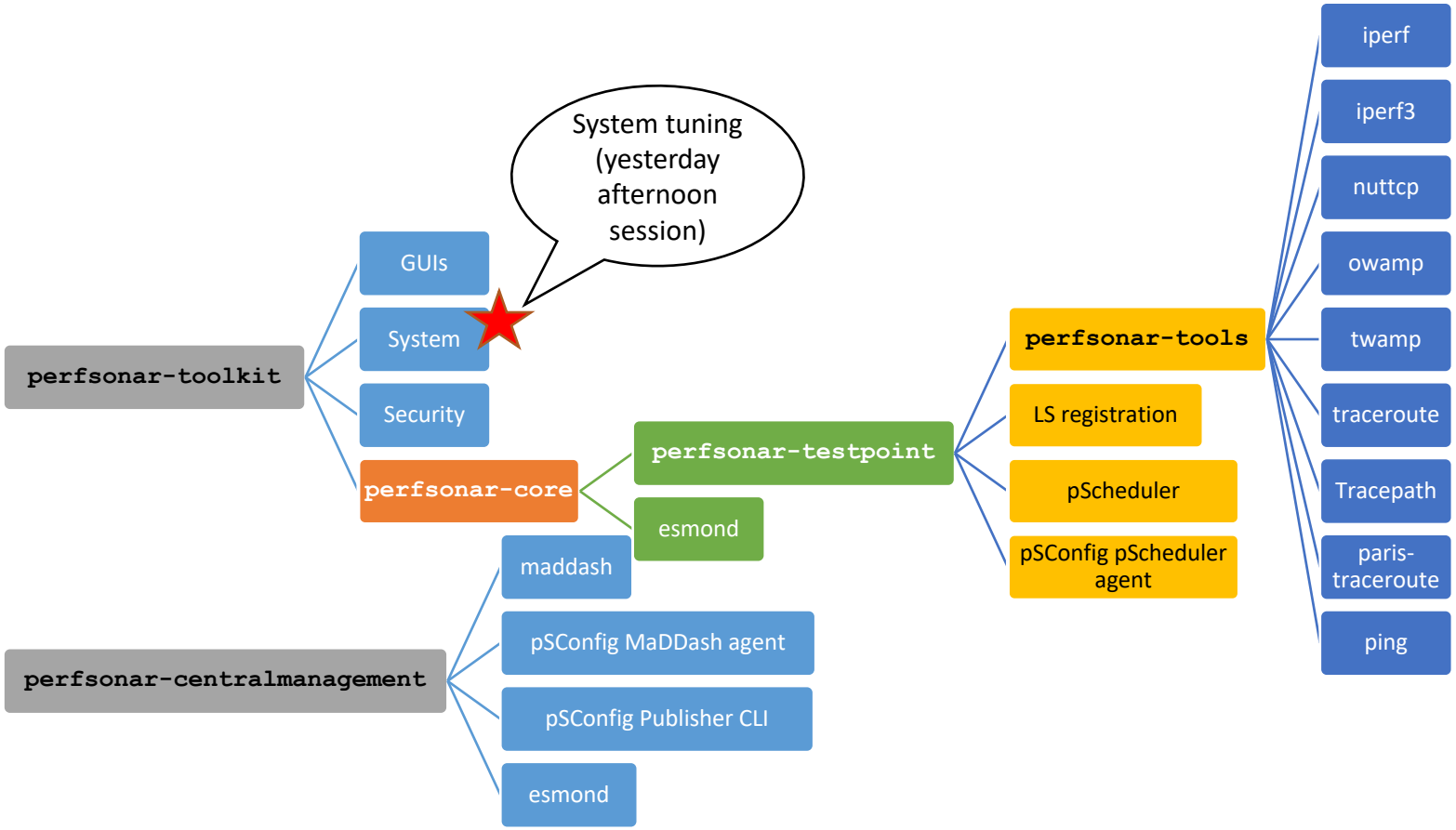
Architecture



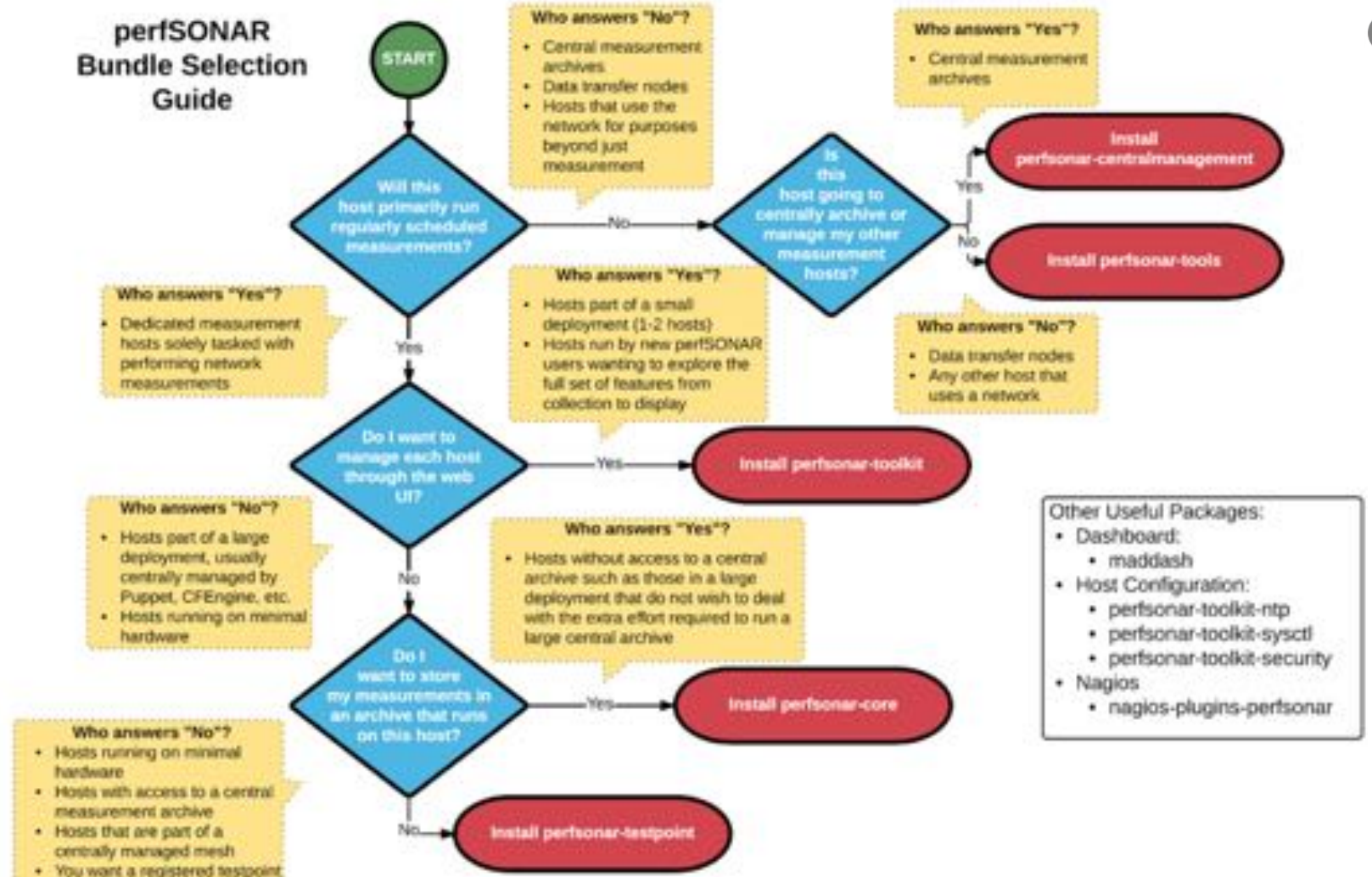
Install Options: Classic or Advanced

- CentOS 7 ISO image
  - Full toolkit install
  - Easy, all contained
- Want more control? Bundle of packages
  - perfsonar-tools
  - perfsonar-testpoint
  - perfsonar-core
  - perfsonar-toolkit
  - perfsonar-centralmanagement
  - + optional packages

Package bundles structure



perfSONAR Bundle Selection Guide



perfSONAR Host Hardware

ASTRON perfSONAR training

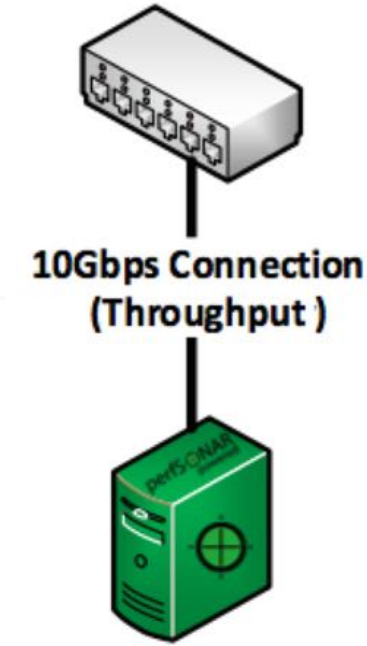
Antoine Delvaux, PSNC, antoine.delvaux@man.poznan.pl
Szymon Trocha, PSNC, szymon.trocha@man.poznan.pl
24-26 September 2018

Use Cases

- There are several deployment strategies for perfSONAR Hardware:
  - Bandwidth Only Testing
  - Latency Only Testing
  - Combined
    - Individual NIC for Bandwidth and Latency Testing
    - Shared NIC

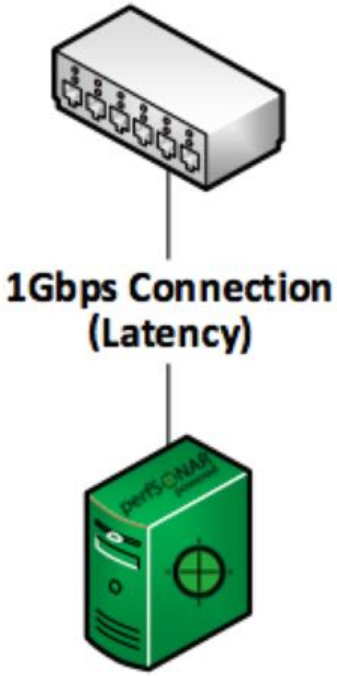
Bandwidth Use Case

- The bandwidth host is designed to saturate a network to gain a measure of achievable throughout (e.g. how much information can be sent, given current end-to-end conditions)
- Can test using TCP (will back off) or UDP (won't back off) – the end result is still the same
- Connectivity can be any size – typically you will want a host that matches the bottleneck of your network



Latency Use Case

- Tests are lightweight (e.g. smaller packets, less of them)
- Designed to measure things like jitter (variation in arrival times of data), packet loss due to congestion, and the time it takes to travel from source to destination
- Connection can be smaller – typically 100Mb or 1Gb connections will do fine. 10Gbps latency testing is not really necessary



Why Separate These?

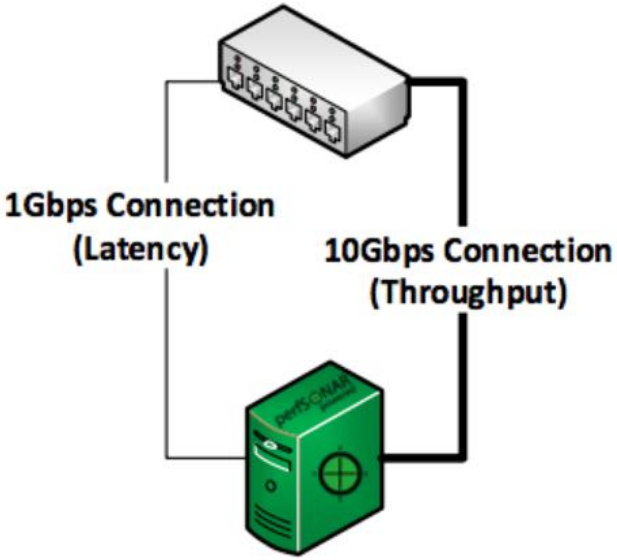
- Bandwidth testing is ‘heavy’ in that it is designed to fill the network as quickly as possible
  - E.g. the memory on the host, the queues on the NIC, the LAN, the WAN, etc.
  - Most throughput tests will cause loss, even if its temporal
- Latency testing is ‘light’ in that it wants to know if there is something that is perturbing the network
  - Congestion from other sources, a failing interface, etc.

Why Separate These?

- Because of the conflicting use case – running these at the same time is problematic
  - A heavy bandwidth test could cause loss in the latency testing.
  - This makes it challenging to figure out ‘where’ the loss is coming from; the host or the network
- If operating two machines isn’t possible, it is desirable to run these on a single host. There are to ways to do this:
  - Dual NICs
  - Single NIC, with isolated testing

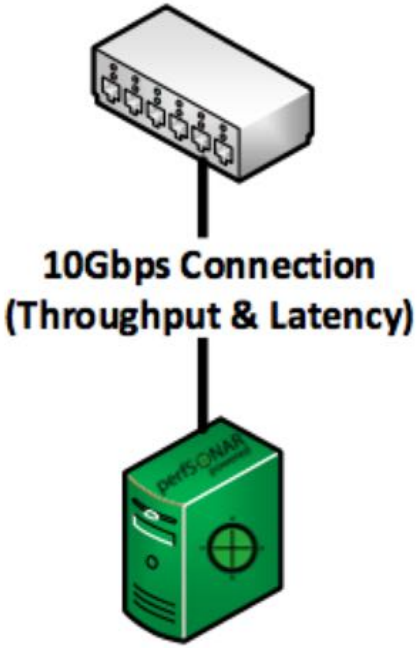
Dual NIC Testing Use Case

- Newer releases of the perfSONAR software facilitate the use of two interfaces
- Host-level routing manages the test traffic to each of the interfaces
- Bottlenecks are still possible:
  - If the host has a single CPU managing both sets of test traffic
  - If there is a memory bottleneck
  - If the NICs do not have an offload engine, they both will need to rely on the CPU to manage traffic flow internally



Single NIC/Dual Testing Use Case

- If the host has a single NIC, tests can be configured to share access
- The previous bottlenecks surrounding the NIC, CPU, and Memory are not as impactful (e.g. they will still be a problem, but impact both sets of tests equally, and one at at time)



Hardware Selection

- Selecting hardware to do the job of measurement is not impossible
- Optimize for the use case of “memory to memory” testing, e.g. we don’t care about the disk subsystem
- Things that matter
  - CPU speed/number
  - Motherboard architecture
  - Memory availability
  - Peripheral interconnection
  - NIC card design + driver support

CPU/Motherboard/Memory

perfSONAR

- Motherboard/CPU
  - Intel Sandy Bridge or Ivy Bridge CPU architecture
    - Ivy Bridge is about 20% faster in practice
    - High clock rate better than high core count for measurement
    - Faster QPIC for communication between processors
  - Multi-processor is waste given that cores are more and more common
  - Motherboard/system possibilities:
    - SuperMicro motherboard X9DR3-F
    - Sample Dell Server (Poweredge r320-r720)
    - Sample HP Server (ProLiant DL380p gen8 High Performance model)
- Memory speed – faster is better
  - We recommend at least 8GB of RAM for a test node (minimum to support the operating system and tools). More is better – especially for testing over larger distances and to multiple sites.

ESnet ENERGY SCIENCES NETWORK GÉANT INDIANA UNIVERSITY INTERNET2 UNIVERSITY OF MICHIGAN

NIC

- There is a difference between 1G and 10G (or larger) testing
- As network speeds increase (e.g. requiring more packets to pass through interfaces per second) problems that are very nuanced become easier to see
  - Failing equipment with small (< .01%) packet loss
  - CRC errors
  - Microbursts of congestion
- Consider these options when choosing a NIC speed

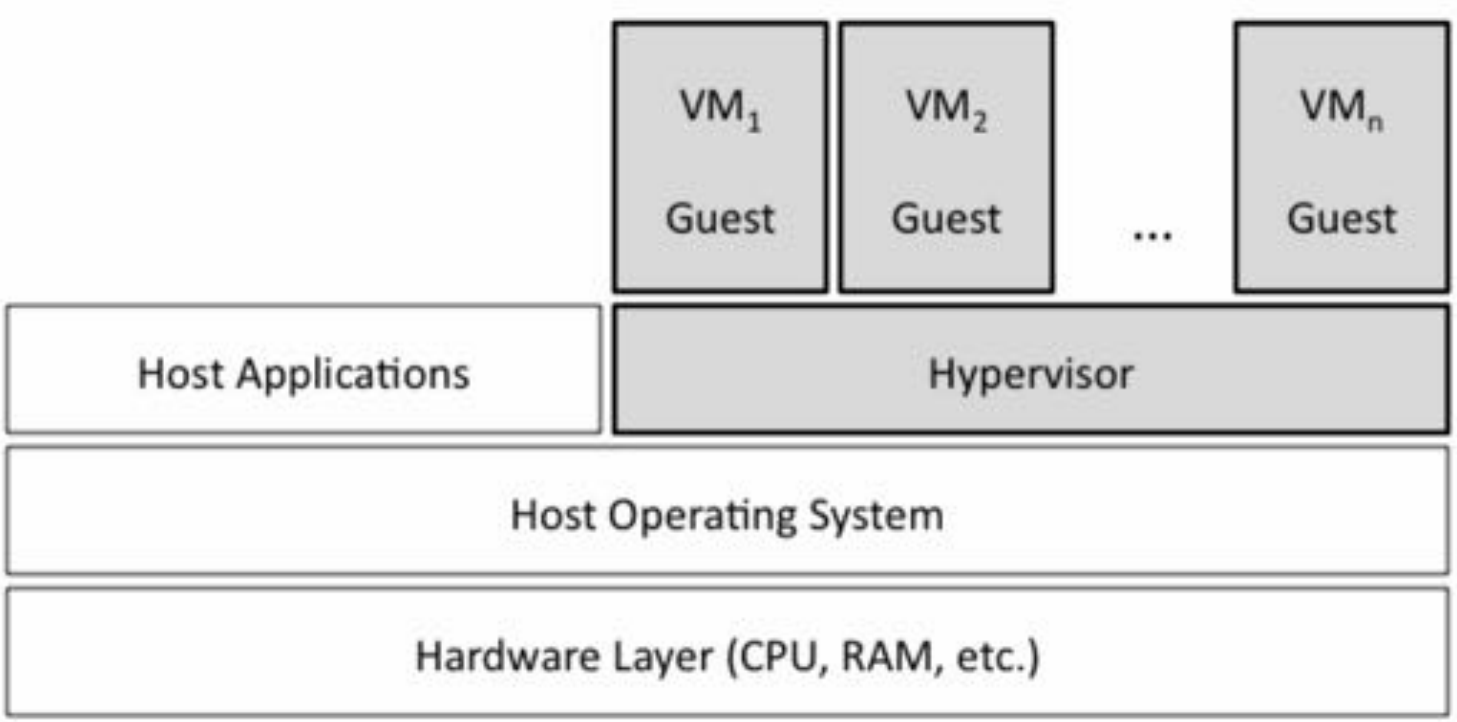
Small nodes

- Low-cost hardware (~200 Euro)
- Known limits
- Can become example measurement experimentation and training platform about network measurement

Hardware Suggestions

- The target is continually being moved and updated recommendations can be found at: http://docs.perfsonar.net/install_hardware.html
- Additionally talk with others: perfsonar-user@internet2.edu
- Check out existing deployment examples: http://docs.perfsonar.net/deployment_examples.html

Virtualization Introduction



What Time is it?

- Known complication: the ability to keep accurate time.
- perfSONAR uses NTP (network time protocol) which is designed to keep time monotonically increasing
  - Slows a fast clock, skips ahead a slow clock. Never 'reverses' time
- VM environments rely on the hypervisor to tell them what time is – this means time could skip forwards, or backwards.
  - IF NTP sees this, it turns off – this is normally catastrophic for measurement purposes (when do I start? When do I end?)
- Picture on right – jitter observed after a hypervisor adjusted the clock.

Functionality Comparison

perfSONAR

- Pros:
  - Ability to have many ecosystems (Windows, FreeBSD, Linux, etc.) invoked through a standard management layer
  - Utilize resources ‘horizontally’ on the machine. E.g. most times a server sits idle if it has no task. By stacking multiple guest machines onto a single host, the probability of the resource being better utilized increases
  - Ad-hoc, short time testing
- Cons:
  - Limit is reached when machines require resources beyond what is available. Can ‘plan’ for this and design the system so its underutilized, or overprovision in the hopes that there will be no conflicts
  - Because this is a shared resource, it won’t do one job very well.

ESnet ENERGY SCIENCES NETWORK GÉANT INDIANA UNIVERSITY INTERNET2 UNIVERSITY OF MICHIGAN

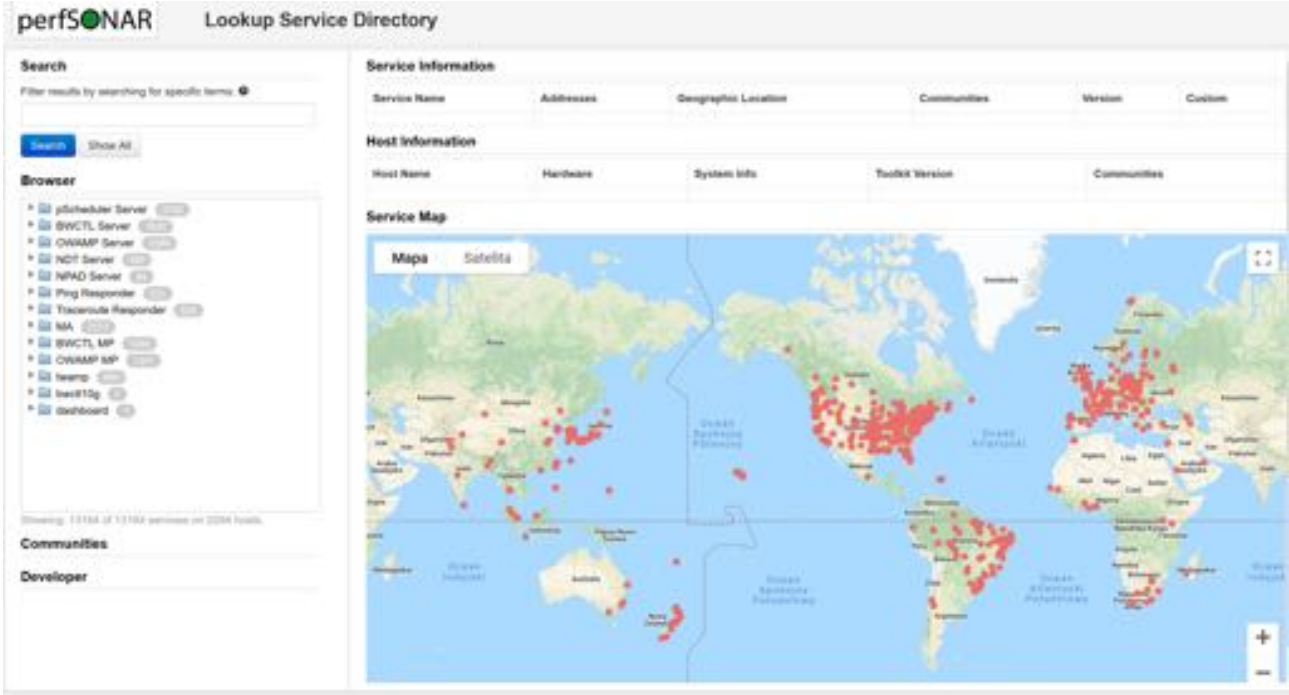
Consolation Prize

- Virtualization can be useful:
  - Experimental setup
  - Testing virtual environments (e.g. cloud providers)
  - Non-latency/bandwidth sensitive testing (passive monitoring, etc.)
  - Smaller performance expectation versus the network
    - E.g. if you are supporting testing for 100s of 100MB connected laptops, a 1G or 10G server in a virtual machine is far greater than the bottleneck of performance

Node discovery

- Once you complete the administrative info – your host will attempt to register with the “Lookup Service”
- This is a global directory that makes it easier to find perfSONAR nodes
- If your host has the admin info present, and isn’t a private IP, it will do this automatically

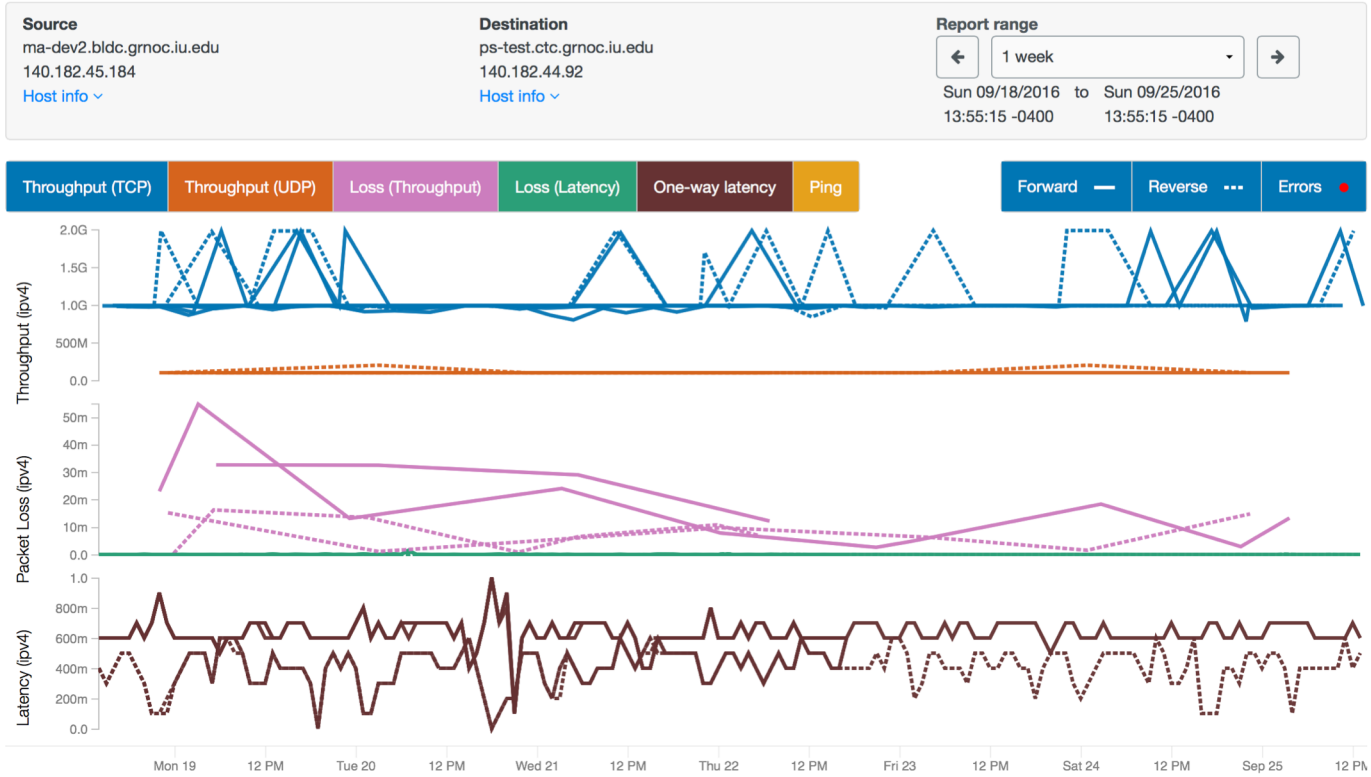
Finding other instances



http://stats.es.net/ServicesDirectory/

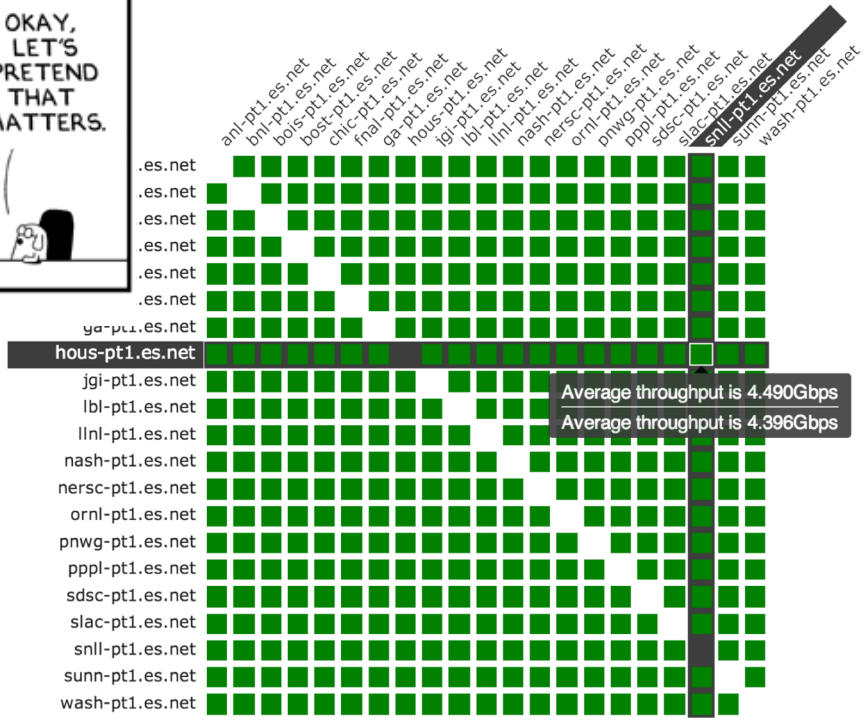
Performance History

perfSONAR



ESnet ENERGY SCIENCES NETWORK GÉANT INDIANA UNIVERSITY INTERNET2 UNIVERSITY OF MICHIGAN

MaDDash: http://ps-dashboard.es.net

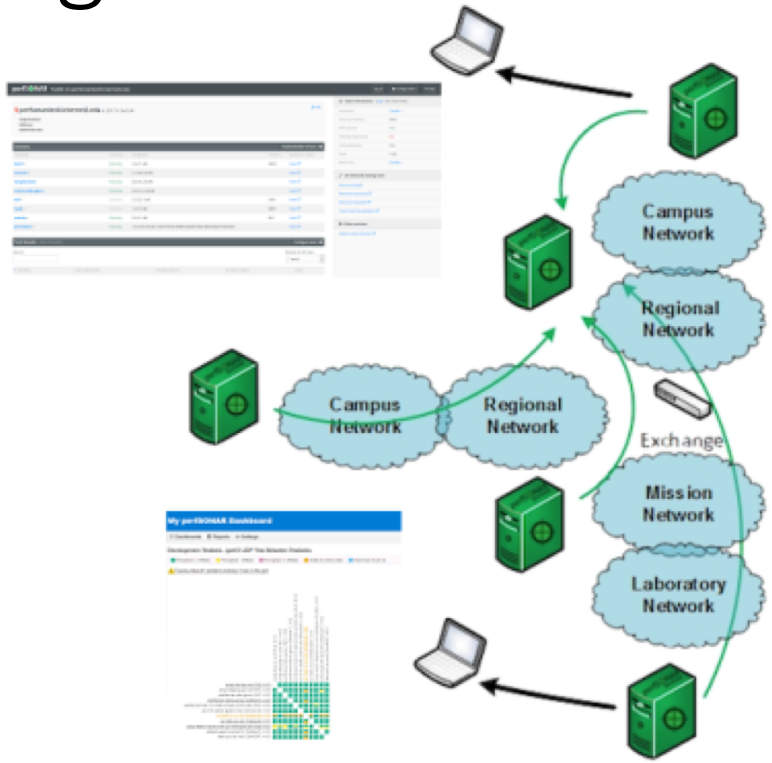


Regular Testing

- There are a couple of ways to do this:
  - Beacon: Let others test to you (e.g. no regular configuration is needed)
  - Island: Pick some hosts to test to – you store the data locally. No coordination with others is needed
  - Mesh: full coordination between you and others (e.g. consume a testing configuration that includes tests to everyone, and incorporate into a visualization)

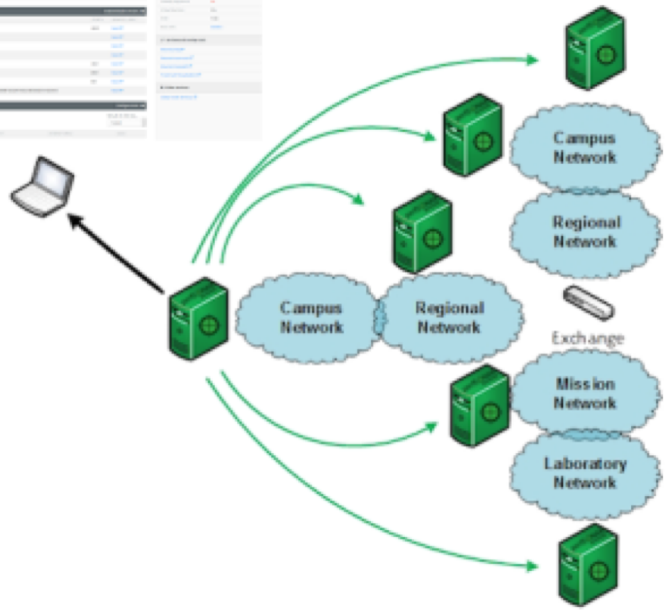
Regular Testing - Beacon

- The beacon setup is typically employed by a network provider (regional, backbone, exchange point)
  - A service to the users (allows people to test into the network)
  - If no regular tests are scheduled, minimum requirements for local storage.
  - Makes the most sense to enable all services (bandwidth and latency)



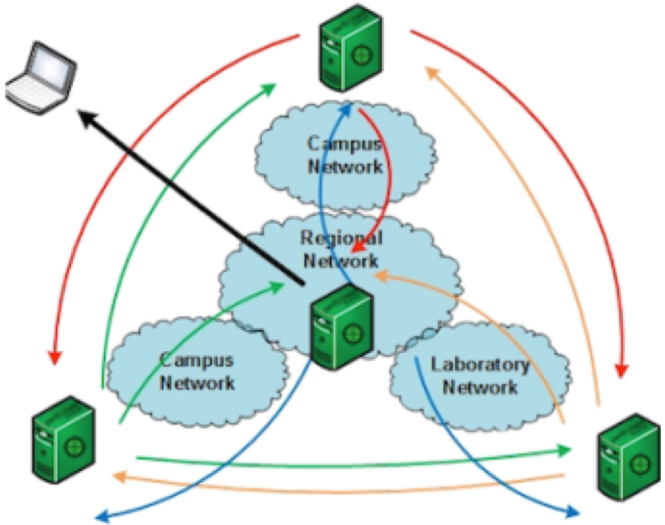
Regular Testing - Island

- The island setup allows a site to test against any number of the 1200+ perfSONAR nodes around the world, and store the data locally.
  - No coordination required with other sites
  - Allows a view of near horizon testing (e.g. short latency – campus, regional) and far horizon (backbone network, remote collaborators).
  - OWAMP is particularly useful for determining packet loss in the previous cases.
  - Throughput will not be as valuable when the latency is small



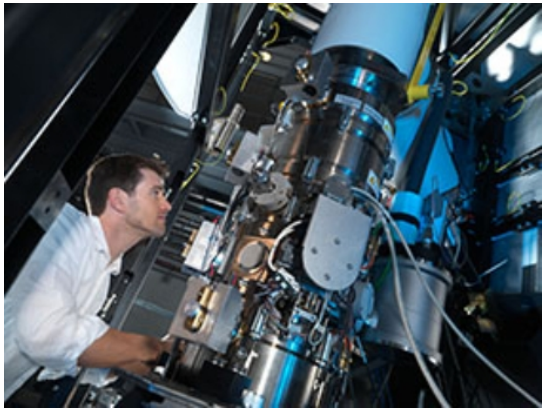
Regular Testing - Mesh

- A full mesh requires more coordination:
  - A full mesh means all hosts involved are running the same test configuration
  - A partial mesh could mean only a small number of related hosts are running a testing configuration
- In either case – bandwidth and latency will be valuable test cases



Growth of data-intensive research

- Growing interest in moving large volumes of research data
  - Captured or generated data to remote computing facility
  - Remote visualisation
  - Data replication / distributed storage / backups
  - To / from cloud
- Data set volumes are increasing
  - 100 TB is no longer 'very large'
  - But moving 100 TB takes 10 Gbit/s of throughput for 24 hours
- Examples:
  - Astrophysics, genomics, environmental sciences, …
  - The new Titan Krios cryo-EM/ET microscope at Diamond
  - The Square Kilometer Array (SKA) project



www.diamond.ac.uk

www.skatelescope.org

Case study – University of Southampton

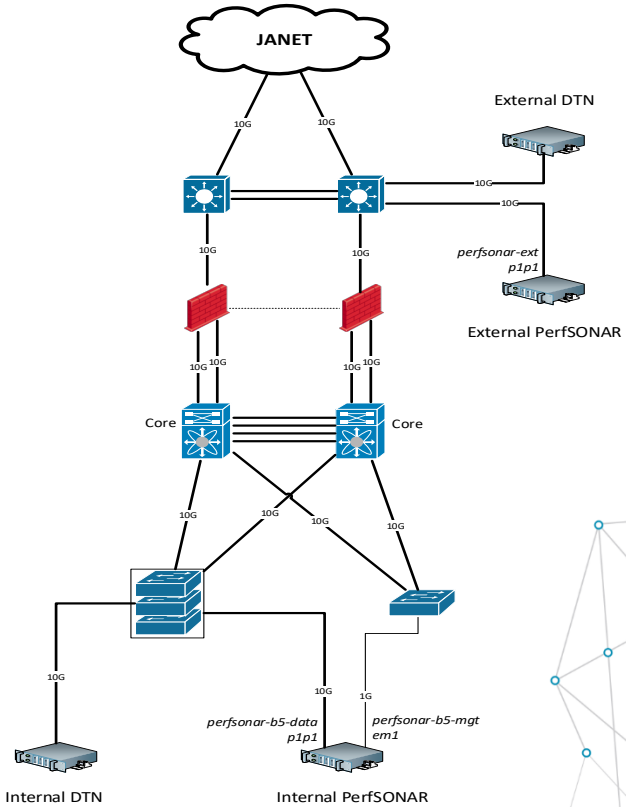
- An example of data still being moved by physical media
  - Southampton μ-VIS X-Ray Imaging Centre
  - Has local facilities, but takes samples to Diamond Light Source ~6 times a year
  - Might gather 10-40 TB of experimental result data per visit
  - One data set typically a ~50 GB file, plus up to 5,000 8-25 MB files
  - Tried using network and *rsync*; obtained ~30 MB/s (240 Mbit/s)
  - Would take 4 days to copy 10 TB home over Janet, best case

- We ought to be able to do better…
  - Diamond end has already deployed Science DMZ
  - Southampton has a 10 Gbit/s campus link to Janet
  - A target of 2 Gbit/s would allow ~1 TB per hour

www.diamond.ac.uk

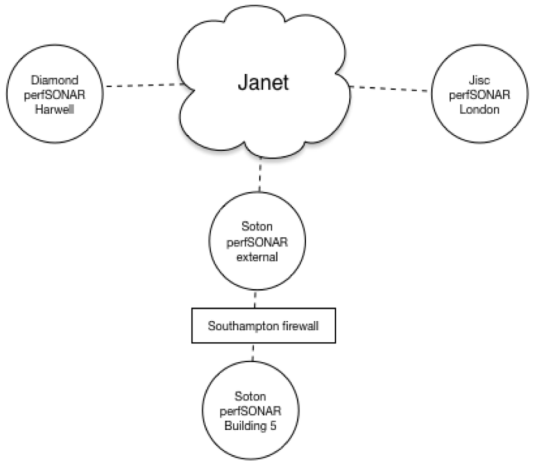
Working with computing service & researchers

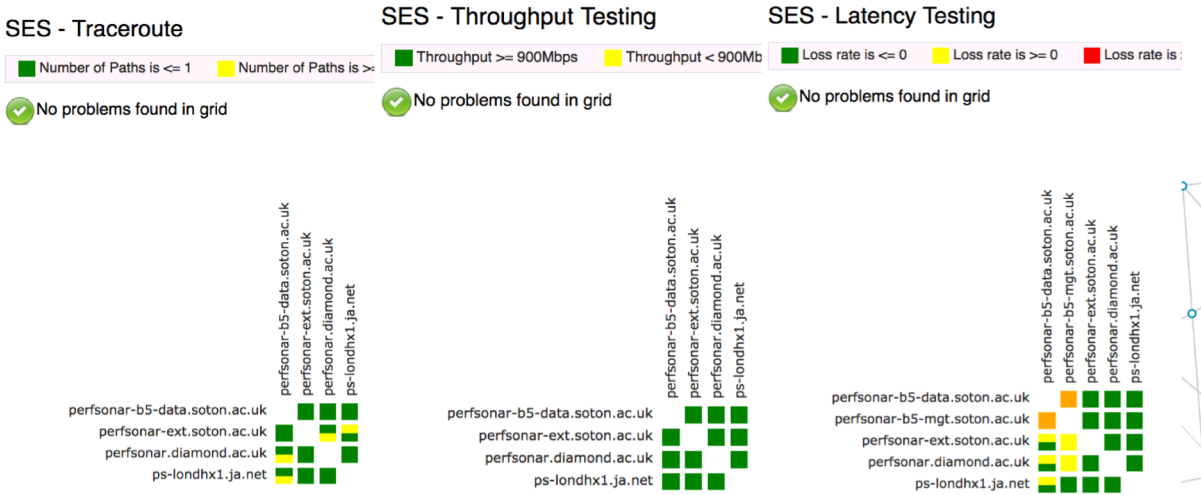
- Met with Diamond and Soton IT & research staff
- Agreed a phased plan of action:
  - Change to using Globus software tools
  - Deploy perfSONAR to measure network characteristics
  - Engineer 10 Gbit/s link to research file store, internal to campus firewall
  - Pilot a 10 Gbit/s DTN at the campus edge
- Outcome:
  - External data transfers achieving 2-4 Gbit/s
  - Potential to transfer their most recent 12 TB data set in 6-12 hours (overnight)



perfSONAR network measurements

- Jisc has deployed two perfSONAR servers – one at a London PoP, one at our Slough DC
- We set up a perfSONAR mesh for the Southampton case study (on a Jisc VM mesh server)
- Used measurement points at Diamond, Janet (London), and two at Southampton
- See - http://ps-dash.dev.ja.net/maddash-webui/index.cgi?dashboard=SES



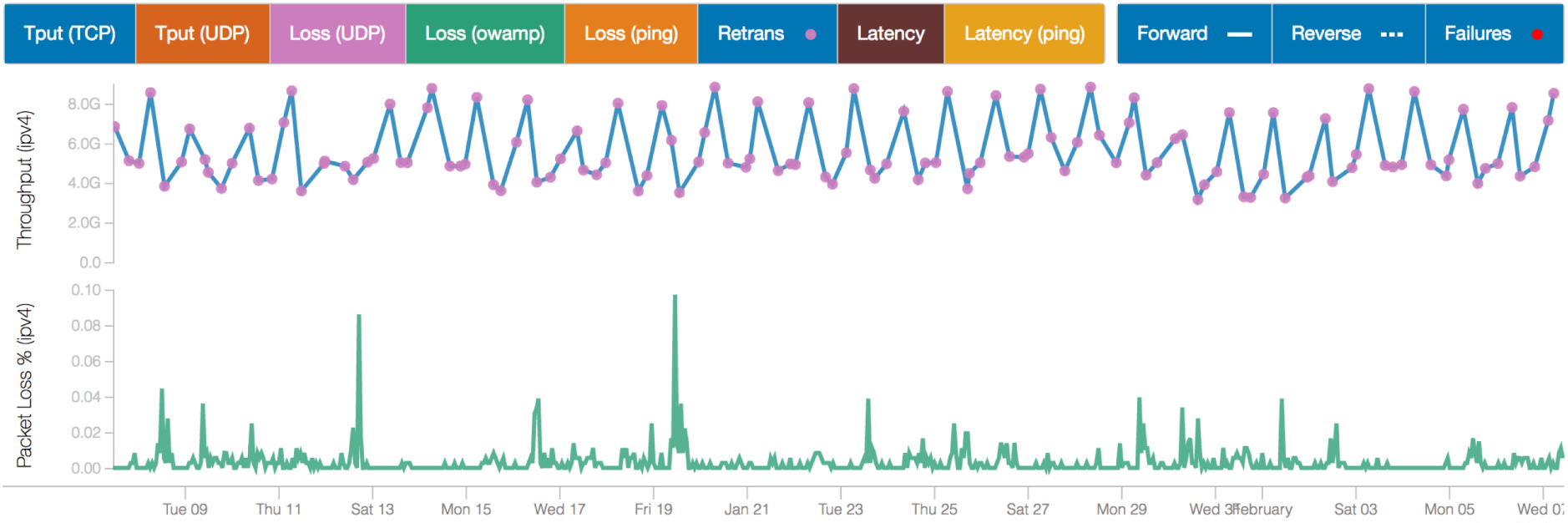


Janet London pS node to internal pS node

Source
ps-londhx1.ja.net
194.83.97.209,2001:630:3c:f800:0:0:0:209
Host info

Destination
perfsonar-b5-data.soton.ac.uk
152.78.176.16
Host info

Report range
1 month
Sun 01/07/2018 10:57:37 (GMT+0) to Wed 02/07/2018 10:57:37 (GMT+0)



Janet London pS node to external pS node

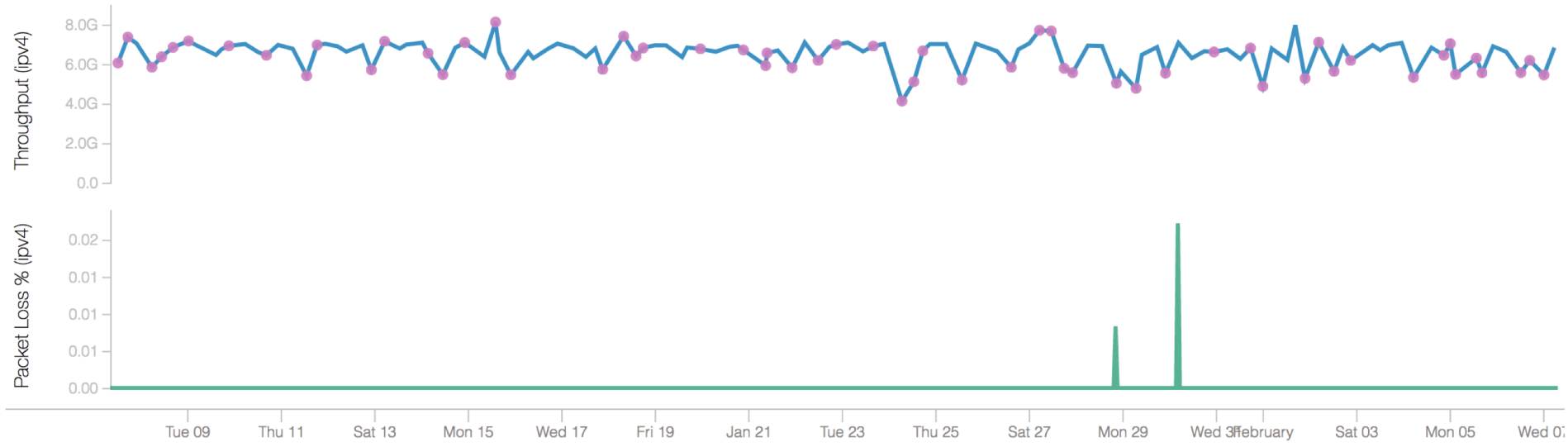
Source
ps-londhx1.ja.net
194.83.97.209,2001:630:3c:f800:0:0:0:209
Host info

Destination
perfsonar-ext.soton.ac.uk
152.78.1.2
Host info

Report range
1 month
Sun 01/07/2018 10:59:12 (GMT+0) to Wed 02/07/2018 10:59:12 (GMT+0)

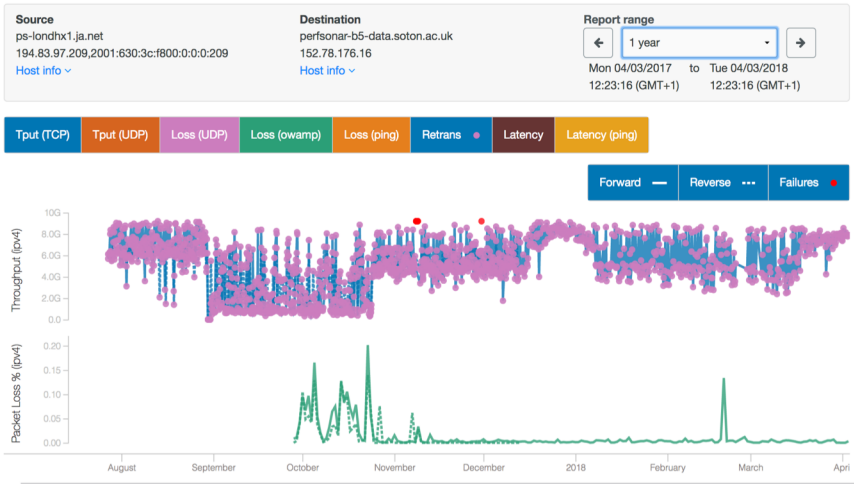
Tput (TCP) | Tput (UDP) | Loss (UDP) | Loss (owamp) | Loss (ping) | Retrans | Latency | Latency (ping)

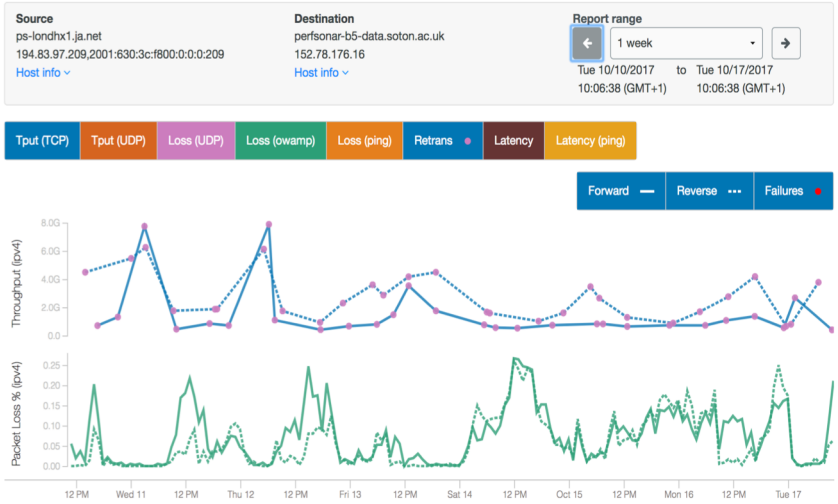
Forward | Reverse | Failures



Aside: perfSONAR highlighting a firewall issue

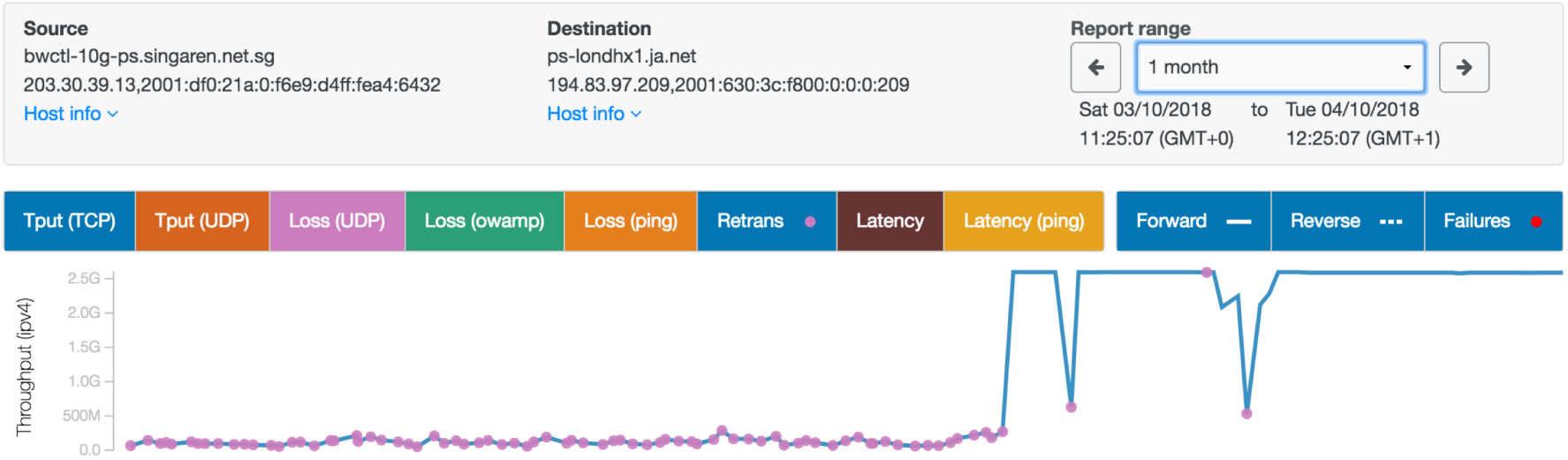
- Slight persistent packet loss after a routine update of the Southampton firewall
- Resulting throughput issues not reported by users, or observable with Jisc Netsight view
- But highlighted by perfSONAR; clear drop in throughput, with higher loss (up to 0.3 %)
- Also gives interesting insight into traffic characteristics over a year-long period





perfSONAR: measuring Imperial – SingAREN

- Working with Imperial College and SingAREN
- New genomics project, needs to send/receive up to 200 TB of data between sites
- perfSONAR highlighted one-way issue on Singapore -> Janet path; faulty hardware
- Resolved with TEINCC/CERNET, now get 2.5 Gbit/s single stream, both ways



perfSONAR Risk

- Since perfSONAR hosts are usually fast, well connected hosts, the main risk is that someone will get on and use the host for a DDOS attack
  - If this happens, WE ALL SUFFER!
  - perfSONAR nodes will get taken down, making the perfSONAR ecosystem less useful
- Data on the host is not particularly valuable.

Conclusions

- A perfSONAR server should requires the same amount of “care and feeding” as any server
  - Yum auto-updates help a lot, but need to make sure they are set them up correctly
  - General server best practices are sufficient
  - Use external monitoring when you can to watch for bad behaviors
- Security is only as advanced as you are willing to make it.
  - Use of external tools, or the audits that you perform, can be a strong defense.
  - If no effort is put in, be prepared to treat the machine as disposable (e.g. do you want ‘pets’ or do you want ‘cattle’)
    - In the disposable case – you certainly don’t want to integrate the machine into your environment very tightly
- There is no magic pill in this space
  - If someone wants to get in, odds are they have a lot more resources than you do to make it so
  - perfSONAR nodes are public and have been compromised before
- Spend some time talking to the right people at your campus about expectations and realities, and then make a plan.