Networking: Data Transfer Nodes and Tuning

Richard Hughes-Jones
GÉANT

The research leading to these results has received funding from the
European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 731122 (GN4-2).

Richard.Hughes-Jones@geant.org

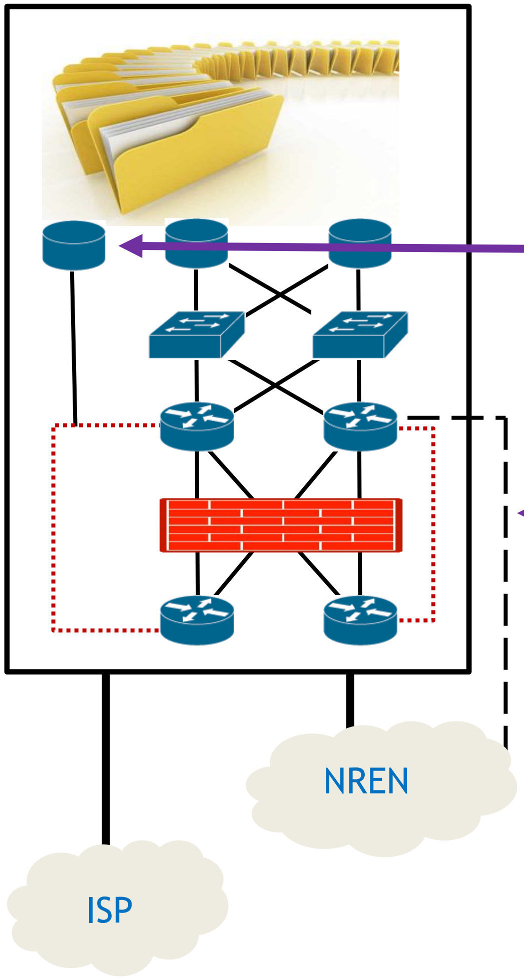
Agenda

- Setting the Scene for Moving Data
  - A Generic Site
  - What is a DTN
- The TCP Protocol
- Tuning a DTN
- Some Effects of Tuning
- Data Transfer Tools
  - WLCG, AENEAS and SKA – moving on from GridFTP

- What performance do we get in Real Life?
- Troubleshooting

Setting the Scene for Moving Data

A Generic Site



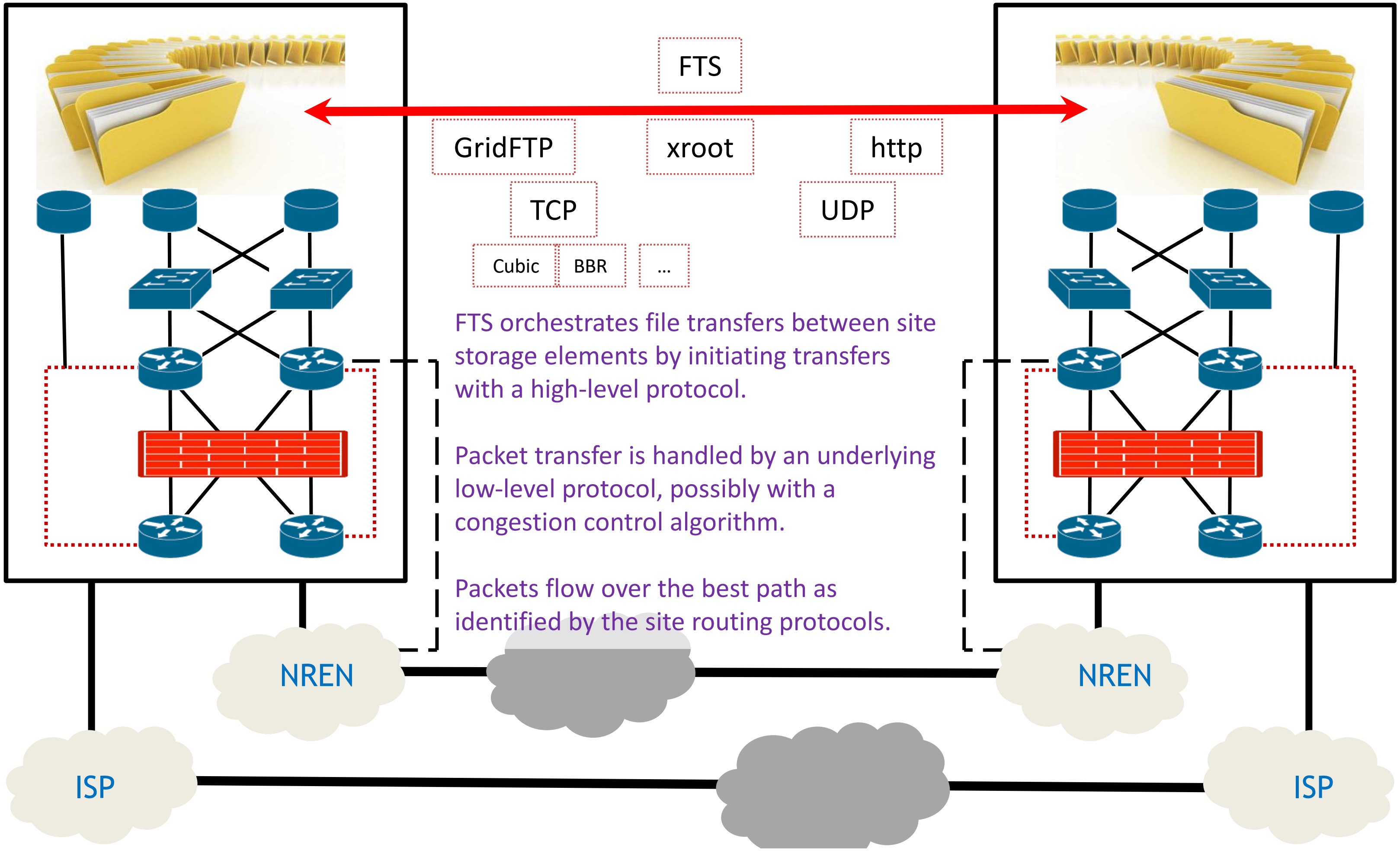
Files are hosted by storage servers, some or all of which may be in a "Science DMZ" with privileged access to the external network.

Where there is a research network link, there may be a privileged path.

The site has one or more connections to the outside world via commercial or research network providers.

Thanks to Tony Cass, CERN

Overall Picture



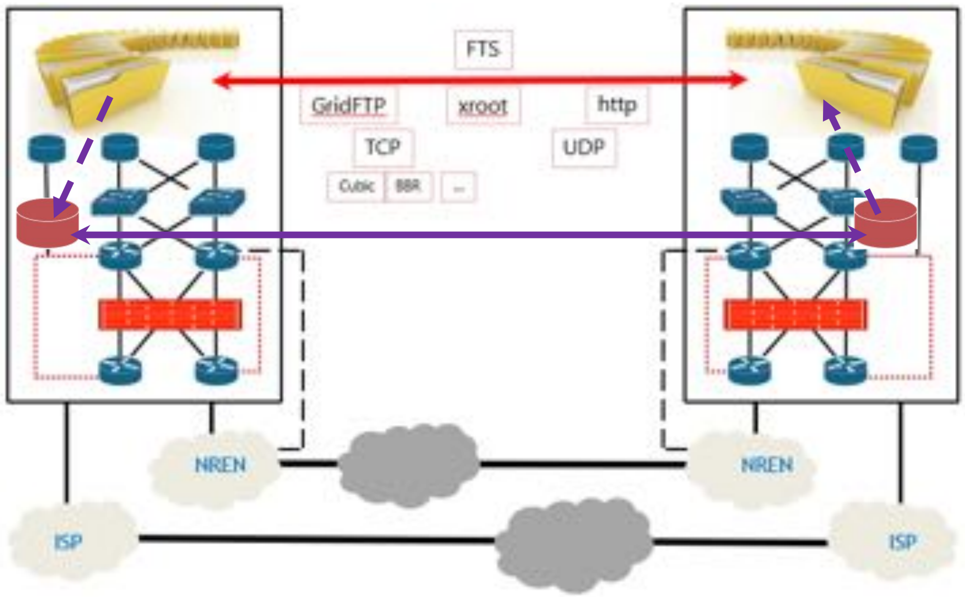
FTS orchestrates file transfers between site storage elements by initiating transfers with a high-level protocol.

Packet transfer is handled by an underlying low-level protocol, possibly with a congestion control algorithm.

Packets flow over the best path as identified by the site routing protocols.

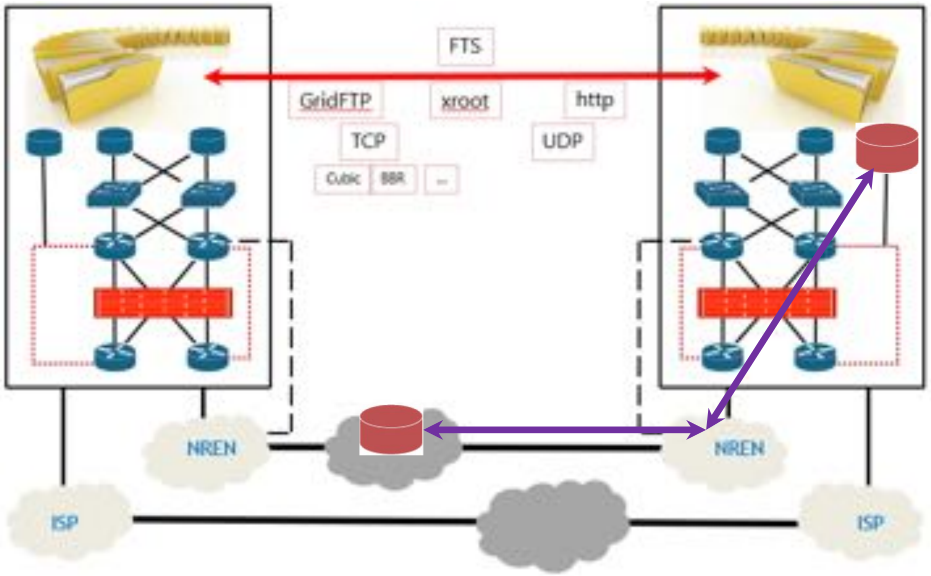
Application – OS – Disks – Node – NIC – Campus – NREN – GEANT – Intercontinental

DTN — Data Transfer Node (ESNET)



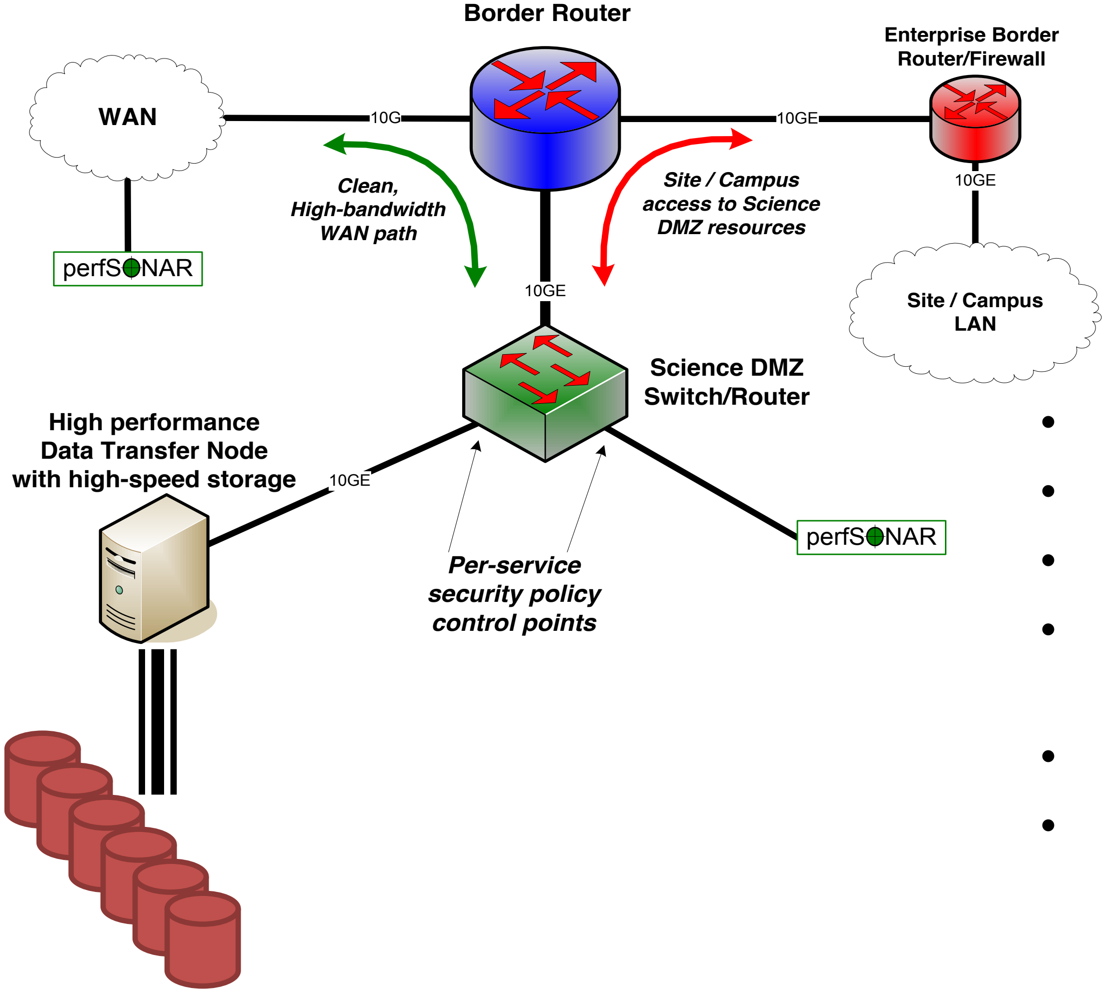
- For ESNET, a Data Transfer Node is a "purpose built [server] dedicated to the function of wide area data transfer"
  - https://fasterdata.es.net/science-dmz/DTN/
- For most WLCG sites, storage nodes will meet the ESNET DTN definition so having a dedicated DTN will be an overhead as data must be copied to/from this node before the wide area transfer.
  - Having a DTN does, though, offer an advantage if point-to-point circuits are used as there is a clearly defined end-point within the site.
  - And a DTN is appropriate, of course, in other contexts, for example an HPC installation where large scale external transfers are not a major activity.

DTN — Data Transfer Node (GÉANT)



- For GÉANT, a DTN is a server in their network.
- The aim is to have a node that can be used both to set a performance baseline against which transfers between site storage nodes can be compared and to allow users to test file transfer application behaviour.
  - Perfsonar is for regular and long-term monitoring, taking into account intra-site networking issues.
  - A GÉANT DTN is more for punctual tests and to enable tests with larger data volumes.

A Typical Science DMZ



- Use a port on Border Router
- Campus firewall remains the same.
- Default deny
- Security policy exceptions only allow traffic from partners.
- Many different modern versions.
- perfSONAR at border & close to the data transfer node

Eli Dart ESnet

TCP Protocol

TCP/IP

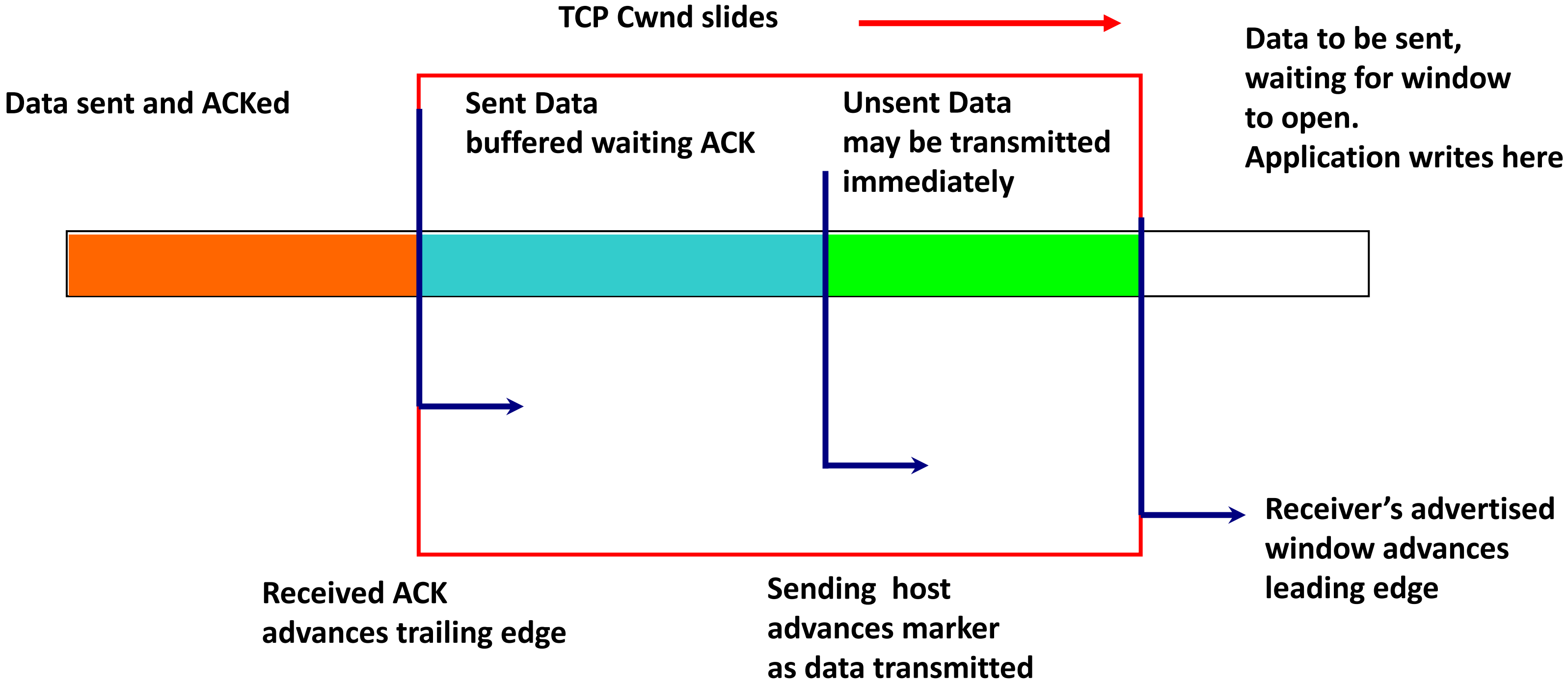
- Most date transfers use TCP/IP
- TCP is a connection-oriented, reliable transport protocol
  - The data is presented to the remote user bit-wise correct
  - Positive acknowledgement (ACK) of each received segment (flow control)
    - Sender keeps record of each segment sent
    - Sender awaits an ACK – "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

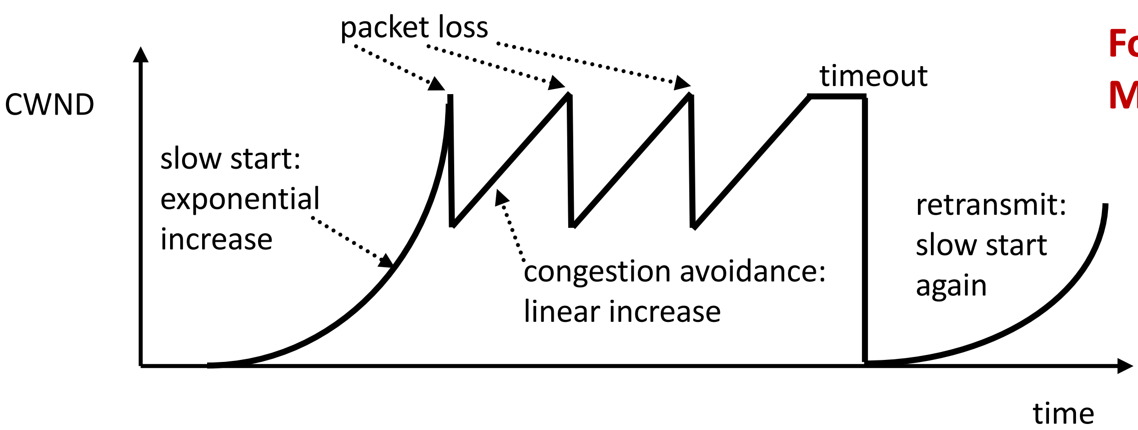


TCP Slowstart

- 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

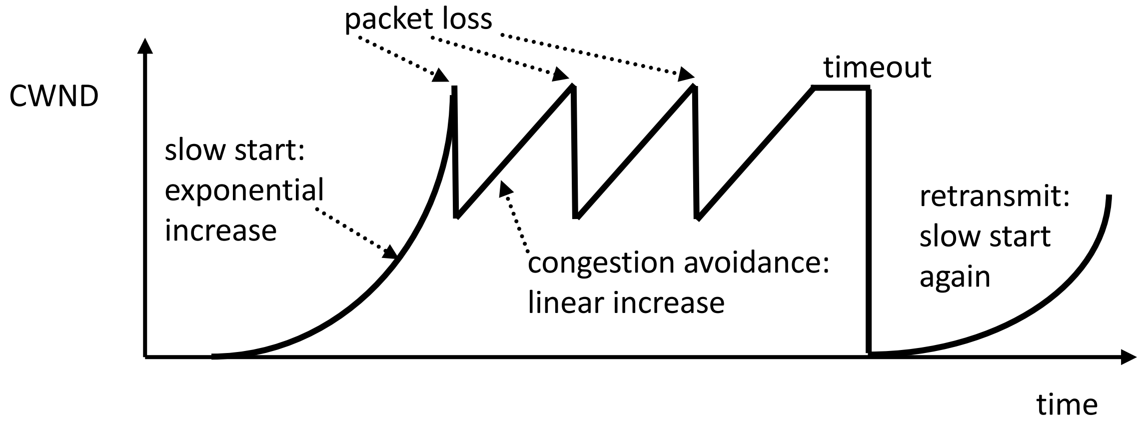


TCP AIMD Congestion Avoidance

- **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**

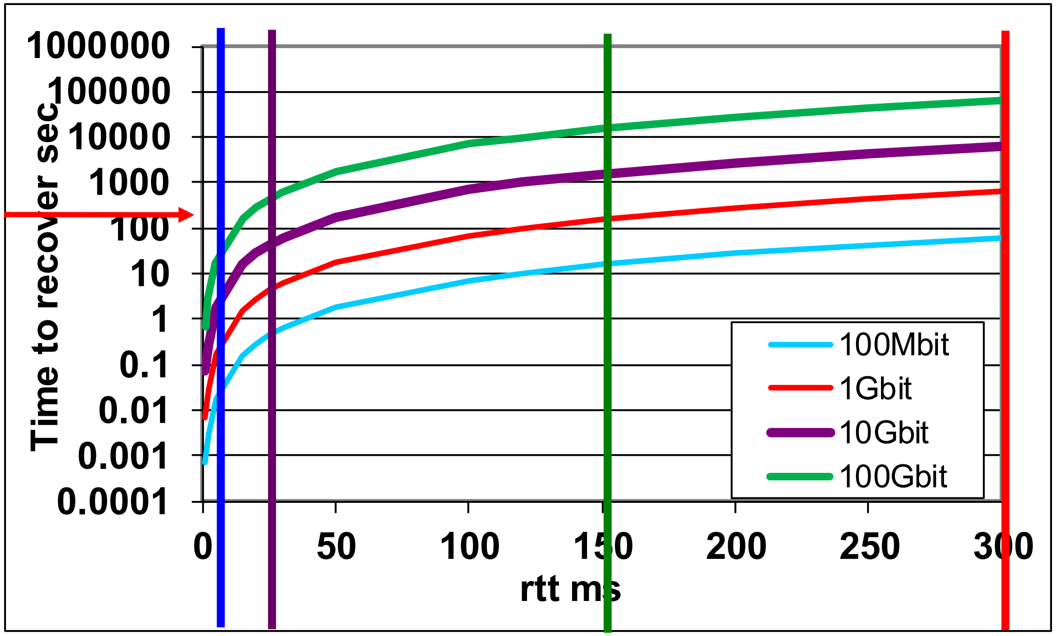


TCP (Reno) – Recovery Time

- 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



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

- For 10 Gbit/s

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
```

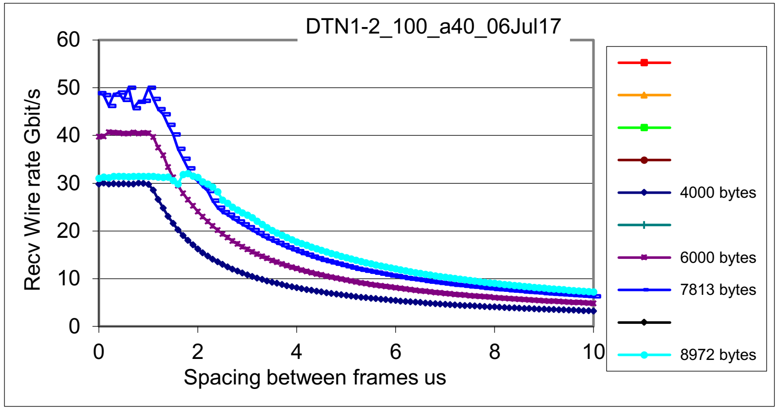
  - net.core.rmem_max net.core.wmem_max
  - net.ipv4.tcp_rmem net.ipv4.tcp_wmem (min / default / max)
  - net.ipv4.tcp_mtu_probing (jumbo frames)
  - net.ipv4.tcp_congestion_control (htcp, cubic)
  - net.ipv4.tcp_mem (set the max to cover rmem/wmem max)
- **Set the affinity of the applications**
  - Using 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

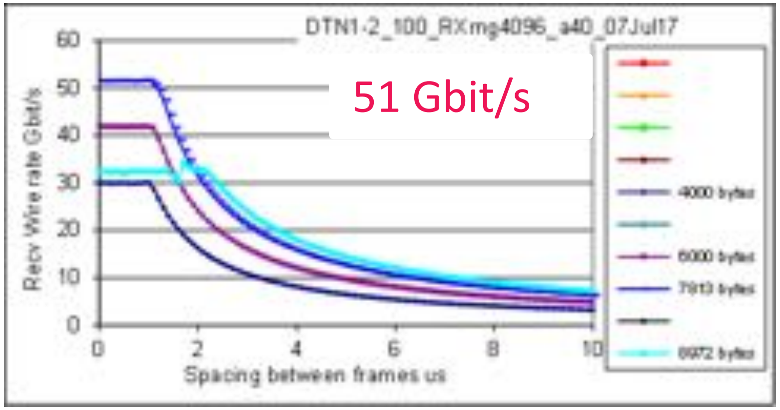
udpmon: Size of Rx Ring Buffer

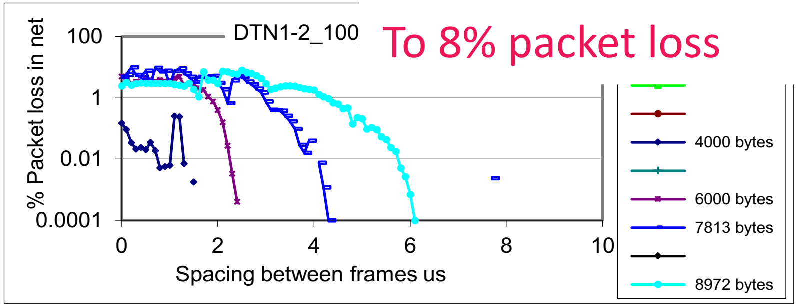
- ConnectX-5, set affinity of udpmon to core 6.
- Use ethtool -S <enp131s0f0> look at rx_out_of_buffer

RX ring 1024



RX ring ≥4096



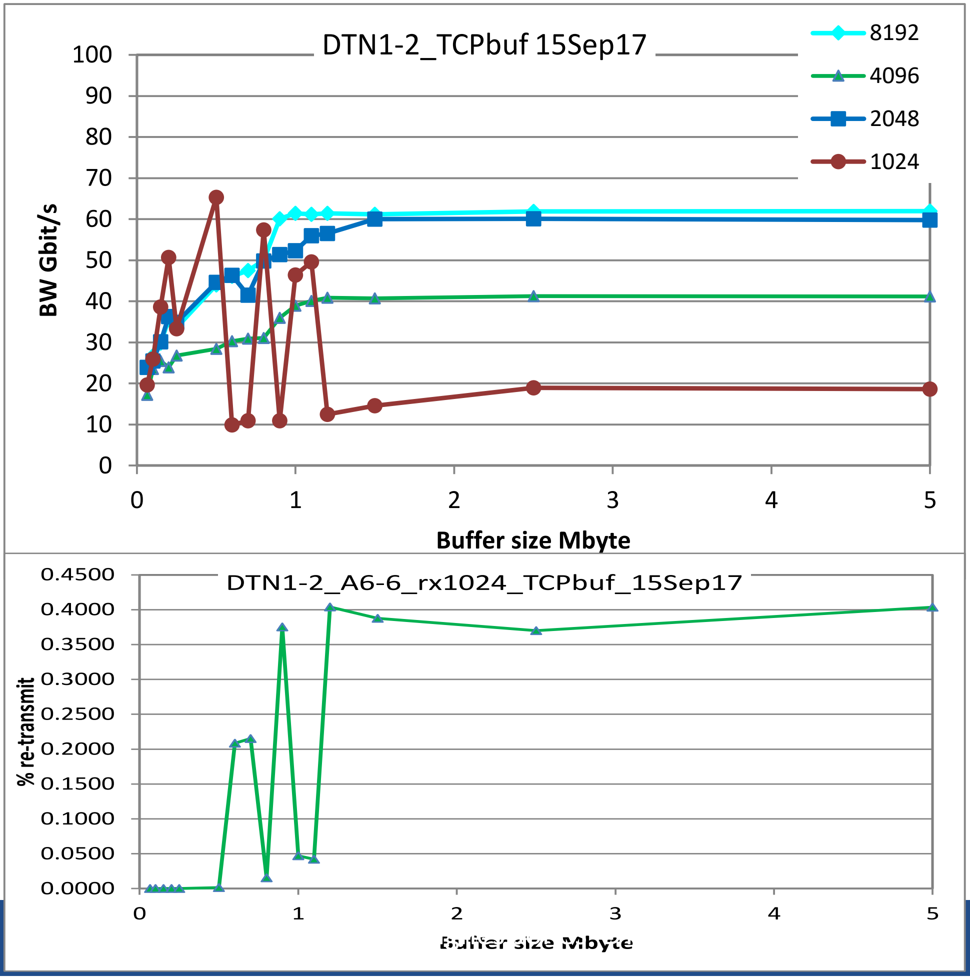


To 8% packet loss

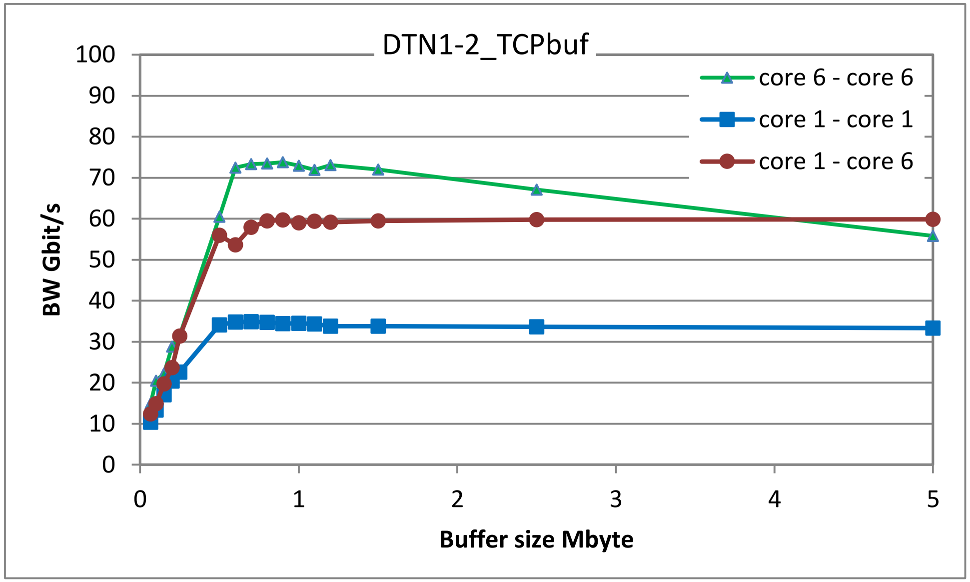
No packet loss

TCP Throughput iperf2 effect of Rx Ring Buffer

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



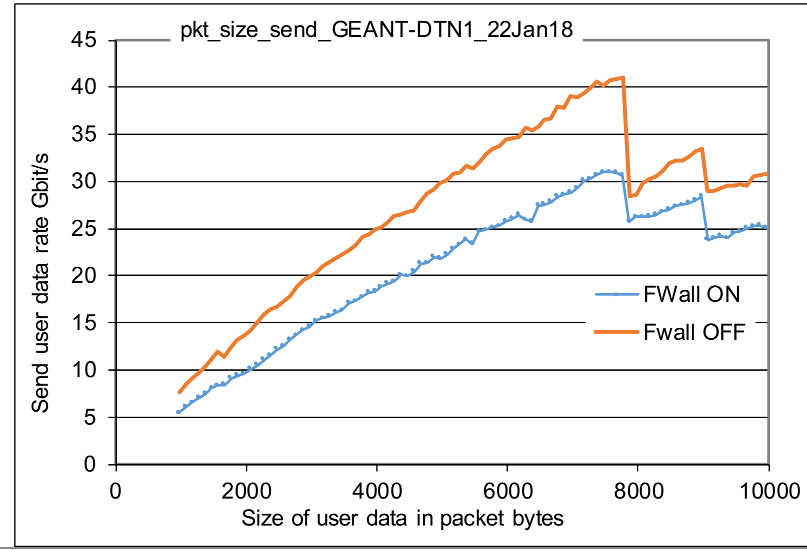
iperf3: TCP Throughput
Using different CPU cores and Nodes



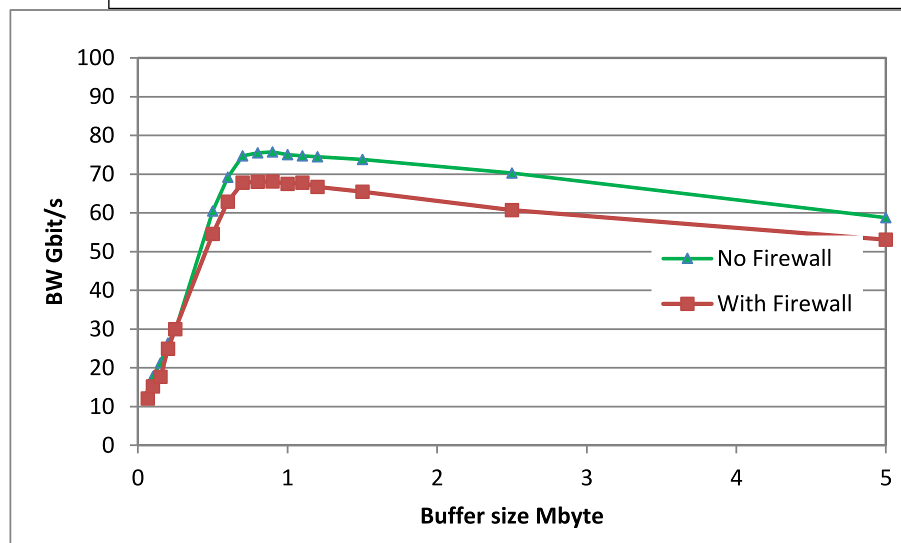
- Firewalls OFF, TCP offload on,
  TCP cubic stack
- RTT 0.4 ms.
- Delay Bandwidth Product 0.5 MB.
- Rises smoothly to the plateau
  at 0.5 MBytes.
- Throughput:
  - 75 Gbit/s Both send & receive on node 1
  - 60 Gbit/s Send on node 0 receive on node 1
  - 35 Gbit/s Both send & receive on node 0
- Very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



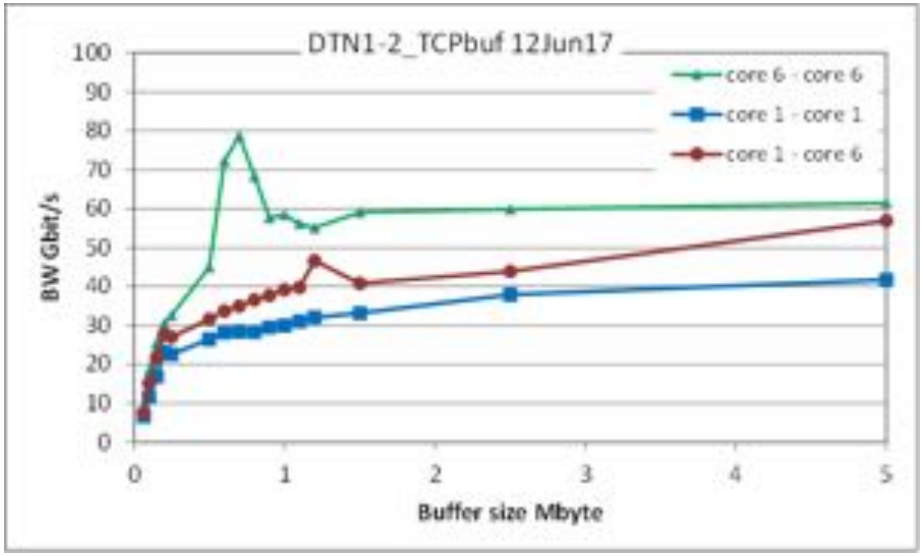
- Run iperf3 on core 6,
  TCP offload on, TCP cubic stack
- RTT 0.4 ms. DBP 0.5 MBytes.
- Rises smoothly plateau at 0.5 Mbytes
- Achievable throughput falls by 7.3 Gbit/s
- No TCP re-transmitted segments



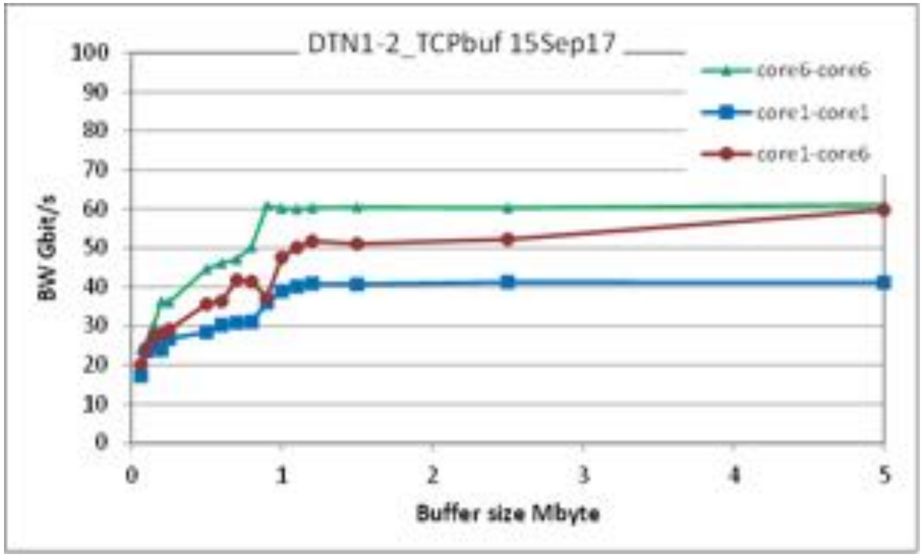
TCP Test Program: Throughput iperf2 & iperf3

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

iperf3



iperf2



Data Transfer Tools
Moving on from GridFTP

Thanks to Maria Girone, CERN

WLCG & Astronomy: data moving applications & tools

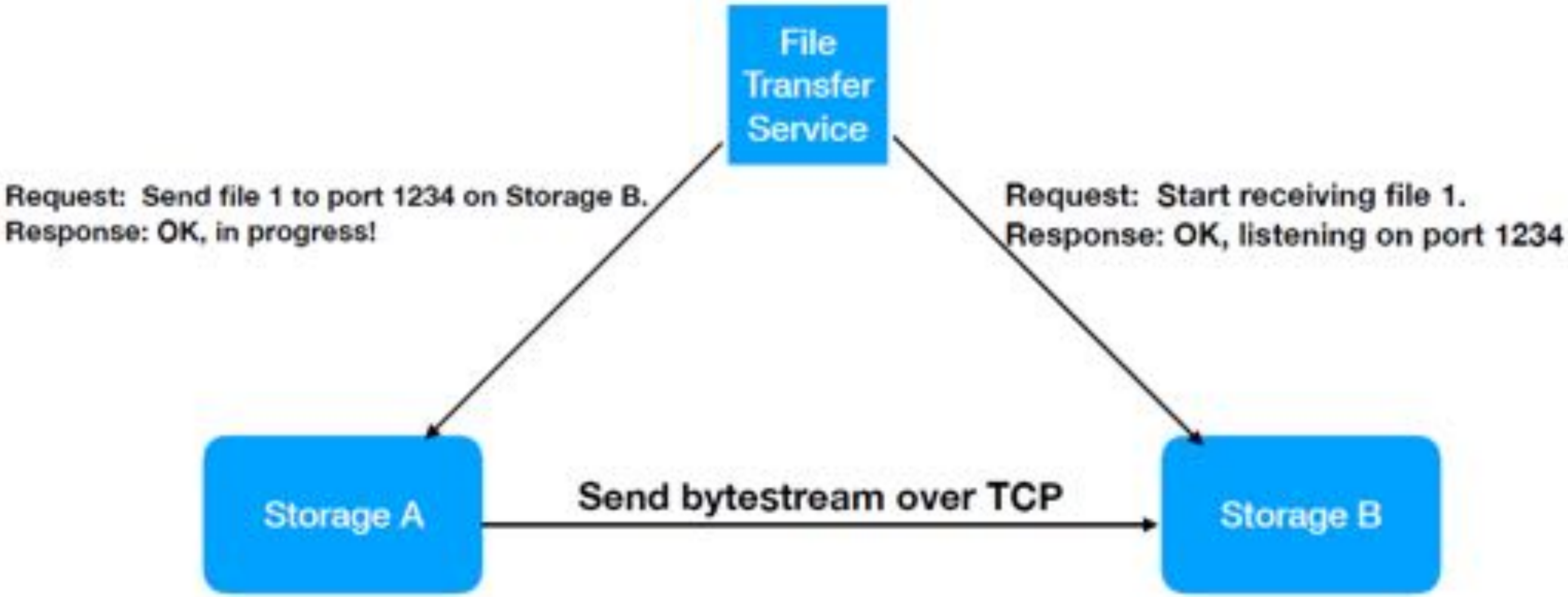
Service	ALICE	ATLAS	CMS	LHCb	CTA	LOFAR
Workflow manager	Alien	PanDA		DIRAC WMS	DIRAC WMS	Self made (genericpipeline)
Data Manager			PhEDEx	DIRAC DMS	DIRAC DMS	Own made (ltacp)
Catalogue Technology	MySQL	Central Oracle	Central Oracle	DIRAC File Catalogue / Oracle	DIRAC File Catalogue / Oracle	Oracle DB (via astrowise)
Information system	Alien	AGIS	SiteDB	DIRAC CS	DIRAC CS	Brains?
File transfer tool	Xrootd	FTS/ SRM	FTS/ SRM FDT	FTS/ SRM GridFTP / WebDAV	FTS/ SRM GridFTP / WebDAV	SRM / globus-url-copy
Local file access	Xrootd	Misc	Misc	Xrootd	Xrootd	Gridftp
Copy to disk	Misc	Misc	Misc	SRM GridFTP / xrootd	SRM GridFTP / xrootd	Per site: gridftp/SRM
Served remotely	Xrootd	Xrootd	Xrootd	Xrootd	Xrootd	
Storage Federation		Xrootd	Xrootd	Xrootd / WebDAV	Xrootd / WebDAV	dCache

A (New) WLCG Transfer Ecosystem

- Mid 2017 it was announced that Globus Toolkit support would end.
- Simulated work on a replacing components
  - **Grid Security Infrastructure (GSI):** An authentication and authorization infrastructure based around concepts of identity and X509 proxies.
  - **GridFTP:** A FTP-like transfer protocol that build on top of GSI, supports third-party transfers, and multi-TCP-stream transfers.
- Propose to use HTTPS with WebDAV extensions
  - e.g. `COPY` to allow Third Party Copy; AAI
- Must link into the WLCG distributed computing File Transfer System
- Timescales CERN DOMA project with involvement from AENEAS and SKA:
  - **Phase 1 31 Dec 2018:** Survey replacement protocols, at least one production site enable a non-GridFTP third-party-copy.
  - **Phase 2 30 June: 2019:** All sites providing >3PB of storage to WLCG experiments required to have one non-GridFTP endpoint in production.
  - **Phase 3 31 December 2019:** All sites providing storage to WLCG experiments must provide a non-GridFTP endpoint.

Thanks to Brian Bockelman, Alessandra Forti , Andy Hanushevsky Mario Lassnig CHEP 2018

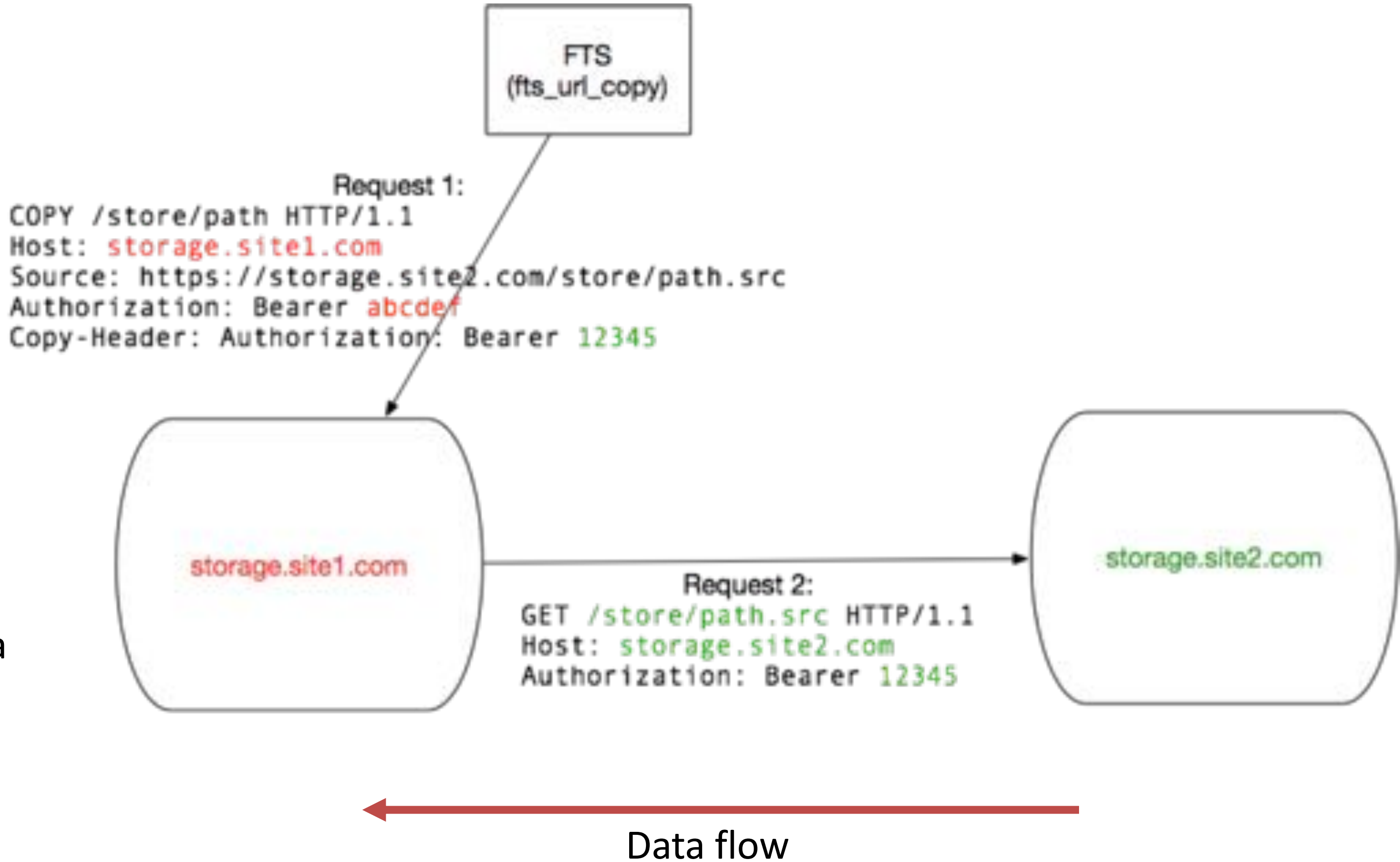
GridFTP Today



- FTS must be authorized to talk to both endpoints.
- Endpoints support the same protocol (GridFTP).
- State Machine & Queueing is in FTS layer.

HTTP Protocol Recap

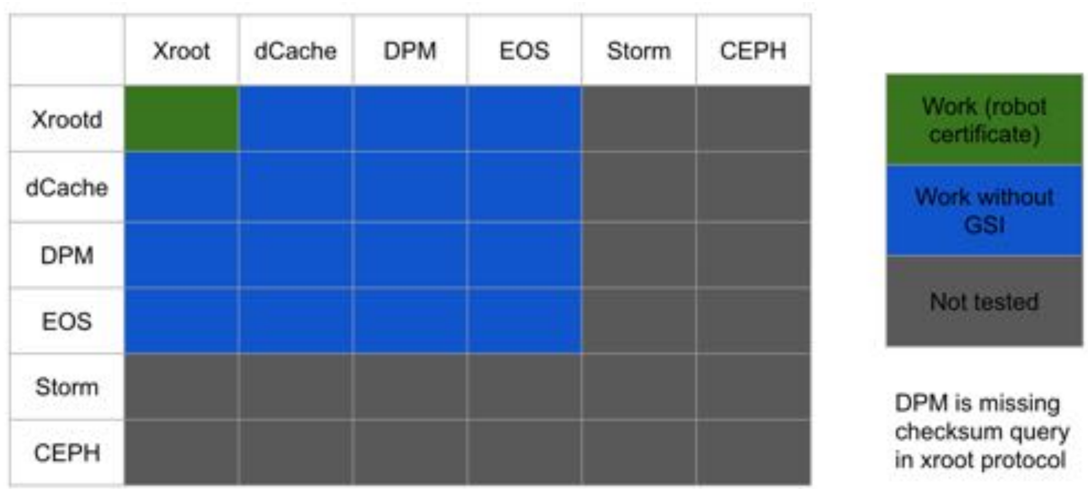
- The WebDAV "COPY" verb is used to orchestrate transfers.
- The "active" side performs a GET / PUT against a remote endpoint.
  - COPY command includes URL for passive side
  - Passive side sees pure HTTPS.
  - Active side can use a bearer token or the third party (FTS) can delegate an X509 proxy.



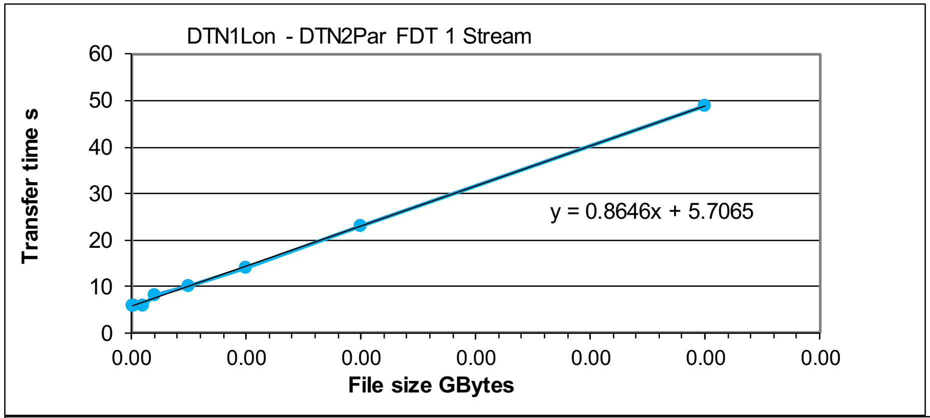
HTTP Connectivity

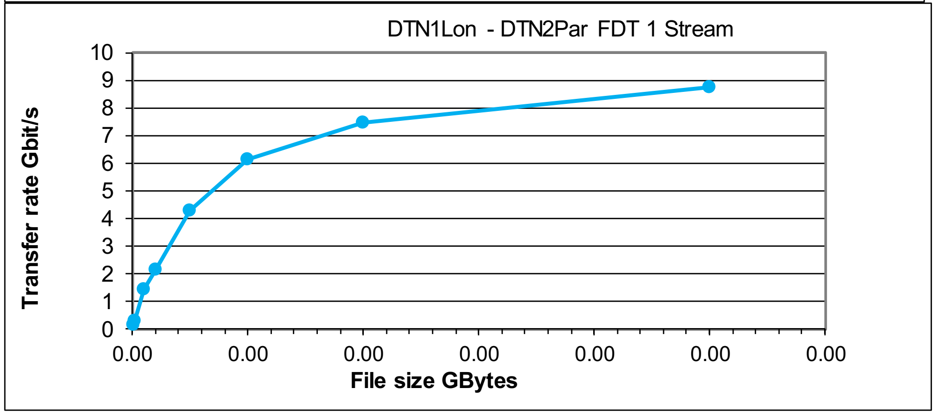
- Xrootd (XrdHttp) now speaks both xrootd and WebDAV/HTTPS protocols.

- Storage Layers:
  - **dCache**: largely working and interoperable.
  - **DPM**: largely working and interoperable.
  - **Xrootd**: Works in active mode (with tokens) or passive mode; does not support X509 delegation.
  - **EOS**: Works only in “passive mode”, no support for token-based transfers. Some DNS issues with test endpoint.



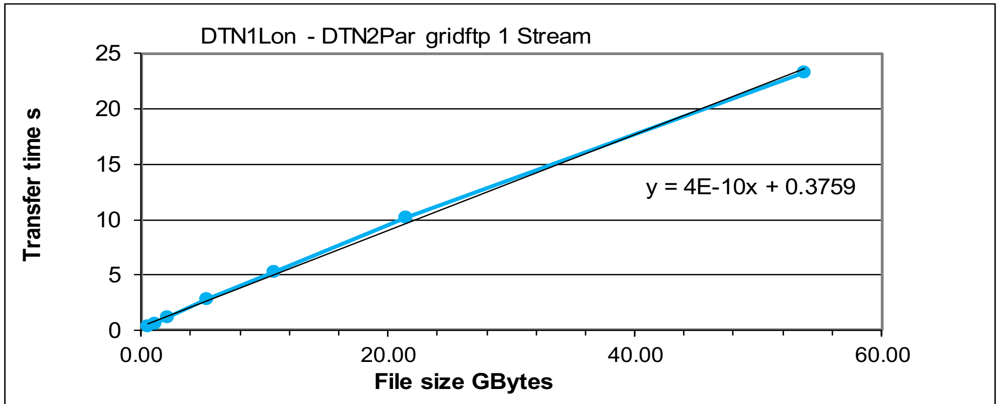
FDT tests between London and Paris
Disk-to-disk over 10Gb/s link

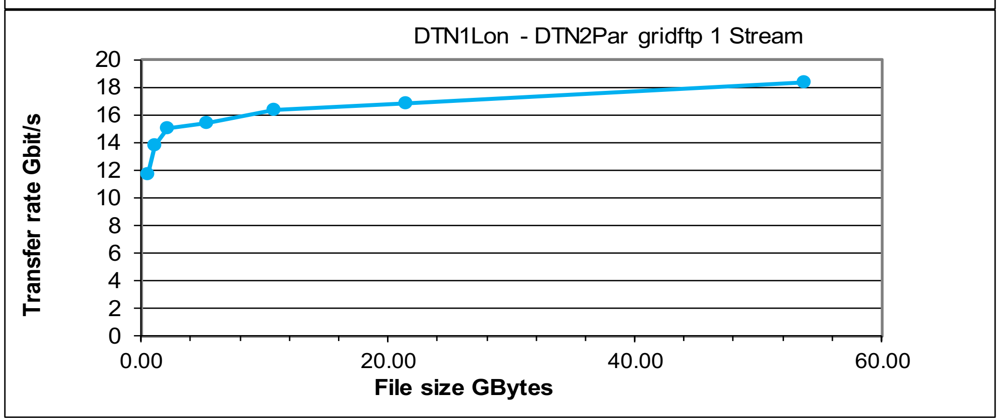




- Files on NVMe disks
- Time a linear increase with File size from 5 to 50 Gbytes
- 5.7s overhead time
- Transfer rate up to 9 Gbit/s
-

GridFTP tests between London and Paris
Disk-to-disk over 100Gb/s link



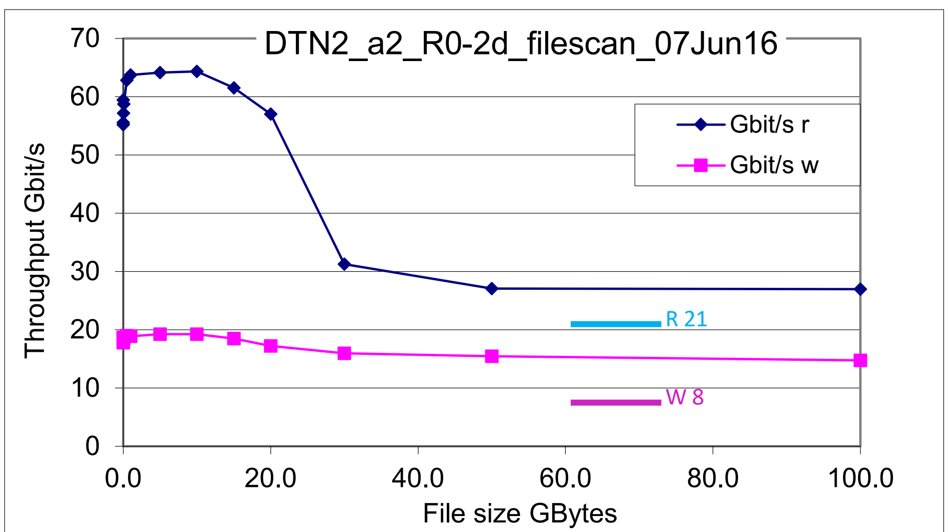


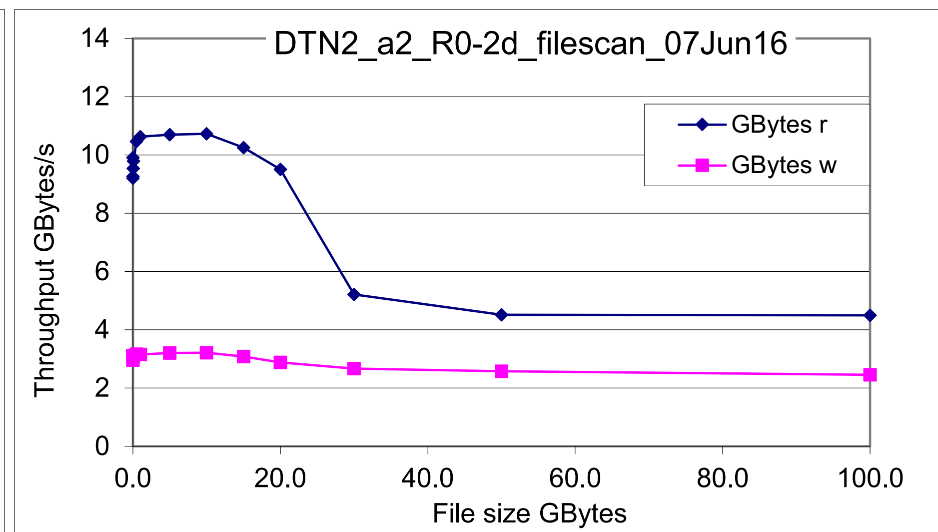
- Same files as in FDT tests
- Time a linear increase with File size from 5 to 50 Gbytes
- Small overhead time
- Transfer rate 18 Gbit/s
- Consistent with disk-memory rate for 1 MVMe disk and 2 disks.

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

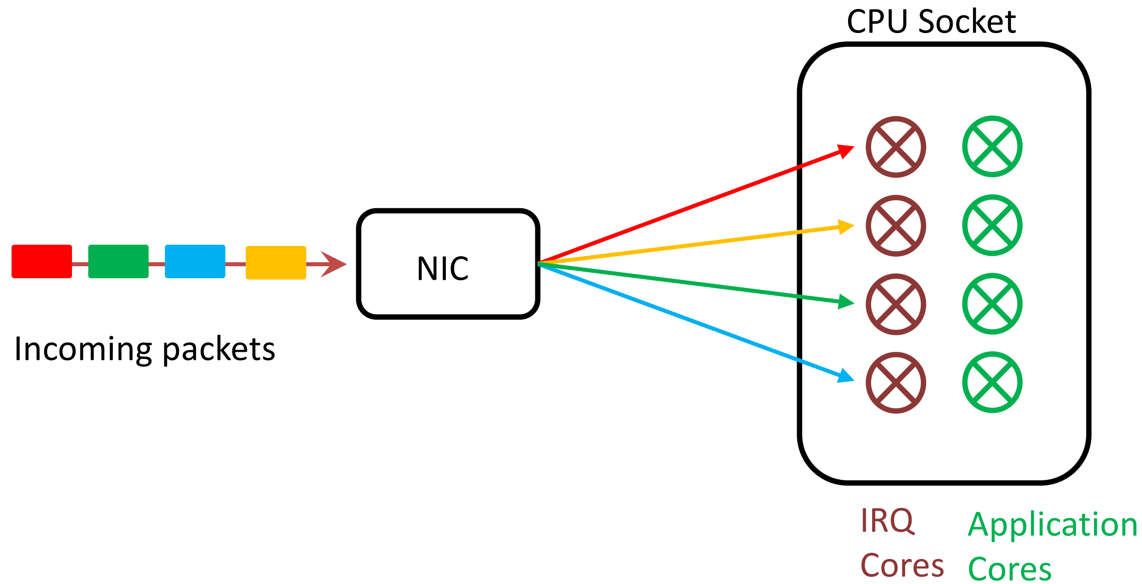
	Read Gbit/s	Write Gbit/s	
1 Disk	~6.2	12.5	yes read < write !
RAID0 2disks	27	15.5	





Using aRFS on the NIC

- accelerated Receive Flow Steering
- NIC directs packets & IRQs for that NIC receive ring to a specified cores
  - Define flow steering rules with ethtool
  - Set the affinity of the IRQ to a specified core
  - Set the affinity of the application to a separate core

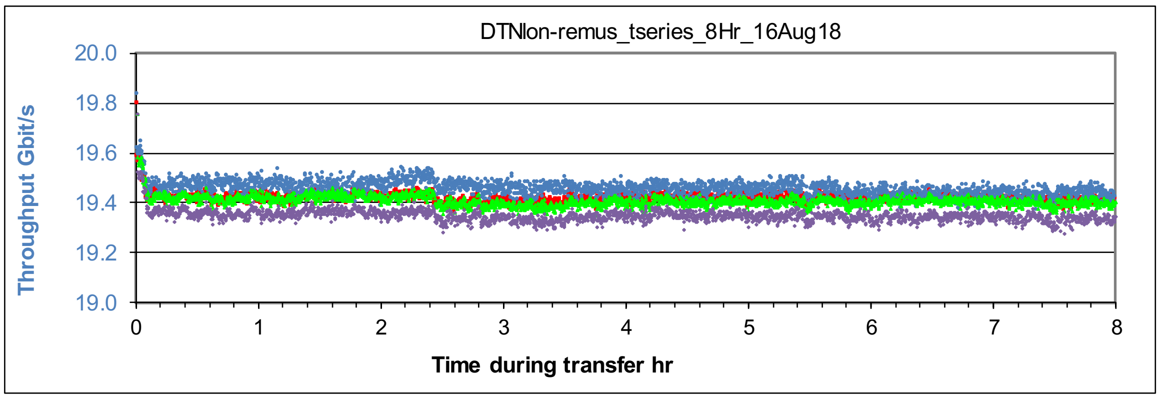



```
# ethtool -U eth2 flow-type udp4 dst-port 14233 loc 1 action 10
# ethtool -U eth2 flow-type udp4 dst-port 14234 loc 1 action 11
# ethtool -U eth2 flow-type udp4 dst-port 14235 loc 1 action 12
# ethtool -U eth2 flow-type udp4 dst-port 14236 loc 1 action 13
```

Advanced European Network of E-infrastructures for Astronomy with the SKA AENEAS - 731016

Use Case: Four 20 Gigabit flows

- No Flow Steering – not very good.
  - 0 to 25% packet loss on the flows.
- Configure receive host NIC accelerated Receive Flow Steering
- Four simultaneous 20 Gbit/s flows between London and JBO for 8 hours
  - A few of the 10s sample periods showed some packet loss, overall $4*10^{-7}$ %.



- CPU Cores:
  - 2 cores for IRQ
  - 4 cores for Application
- 100% CPU load for App cores

Linux 4.4.6-300.fc23.x86_64 (DTNlon) 15/08/18 _x86_64_ (12 CPU)

18:55:45	CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%gnice	%idle
18:55:46	all	9.11	0.00	26.96	0.00	0.00	0.27	0.00	0.00	0.00	63.66
18:55:46	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
18:55:46	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
18:55:46	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
18:55:46	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
18:55:46	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
18:55:46	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
18:55:46	6	27.27	0.00	72.73	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18:55:46	7	23.00	0.00	77.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18:55:46	8	26.00	0.00	74.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18:55:46	9	24.75	0.00	75.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18:55:46	10	0.00	0.00	0.00	0.00	0.00	3.45	0.00	0.00	0.00	96.55
18:55:46	11	0.00	0.00	0.00	0.00	0.00	1.96	0.00	0.00	0.00	98.04

Questions ?

Thanks to Richard Hughes-Jones

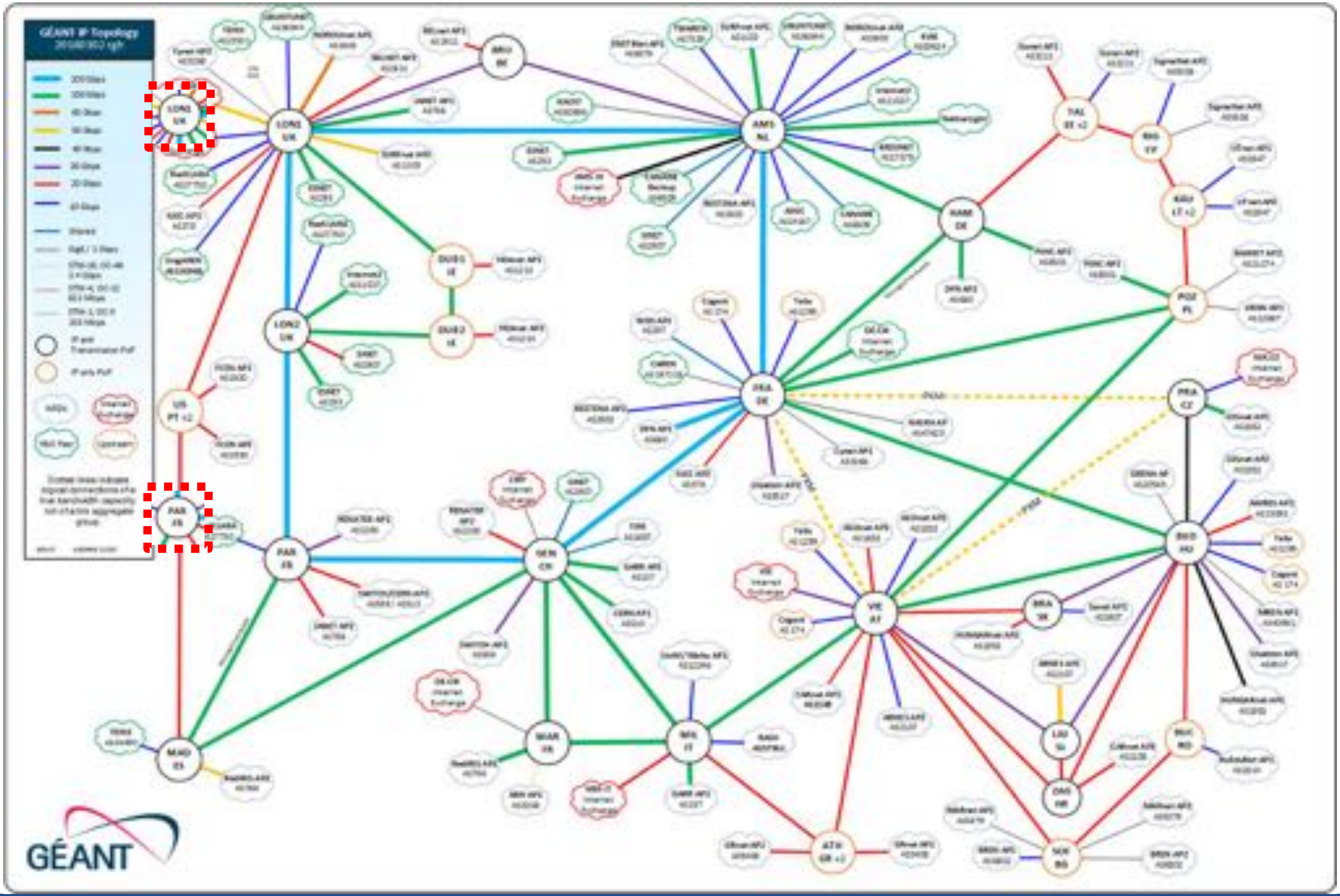
The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 731122 (GN4-2).

Richard.Hughes-Jones@geant.org

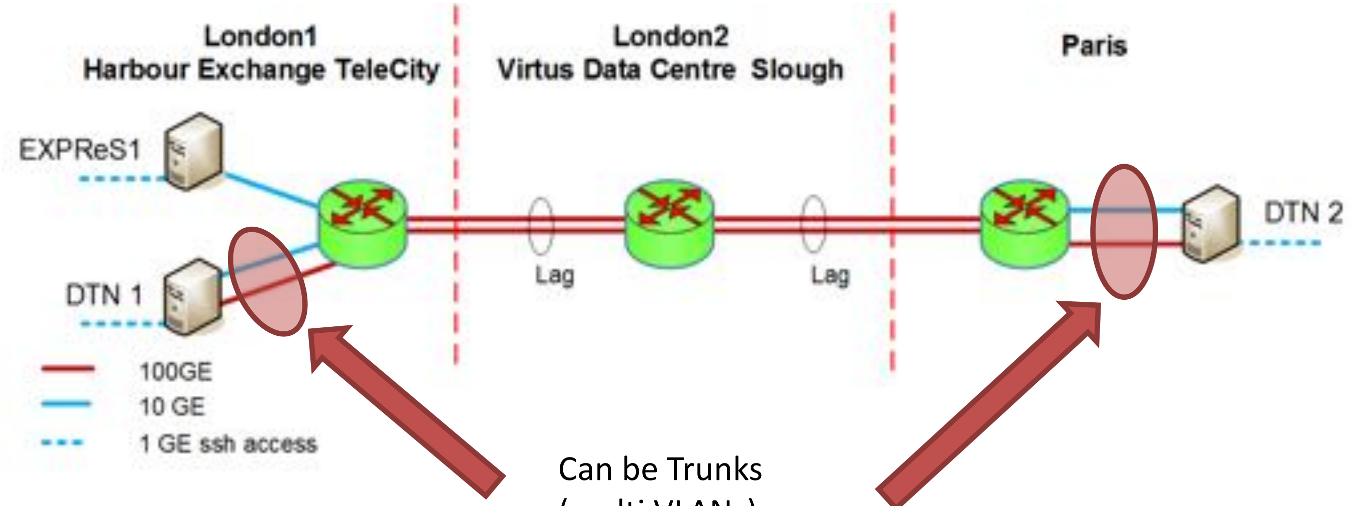
Networking:
Applications and Troubleshooting

Richard Hughes-Jones
GÉANT

What do we get in Real Life?


GÉANT

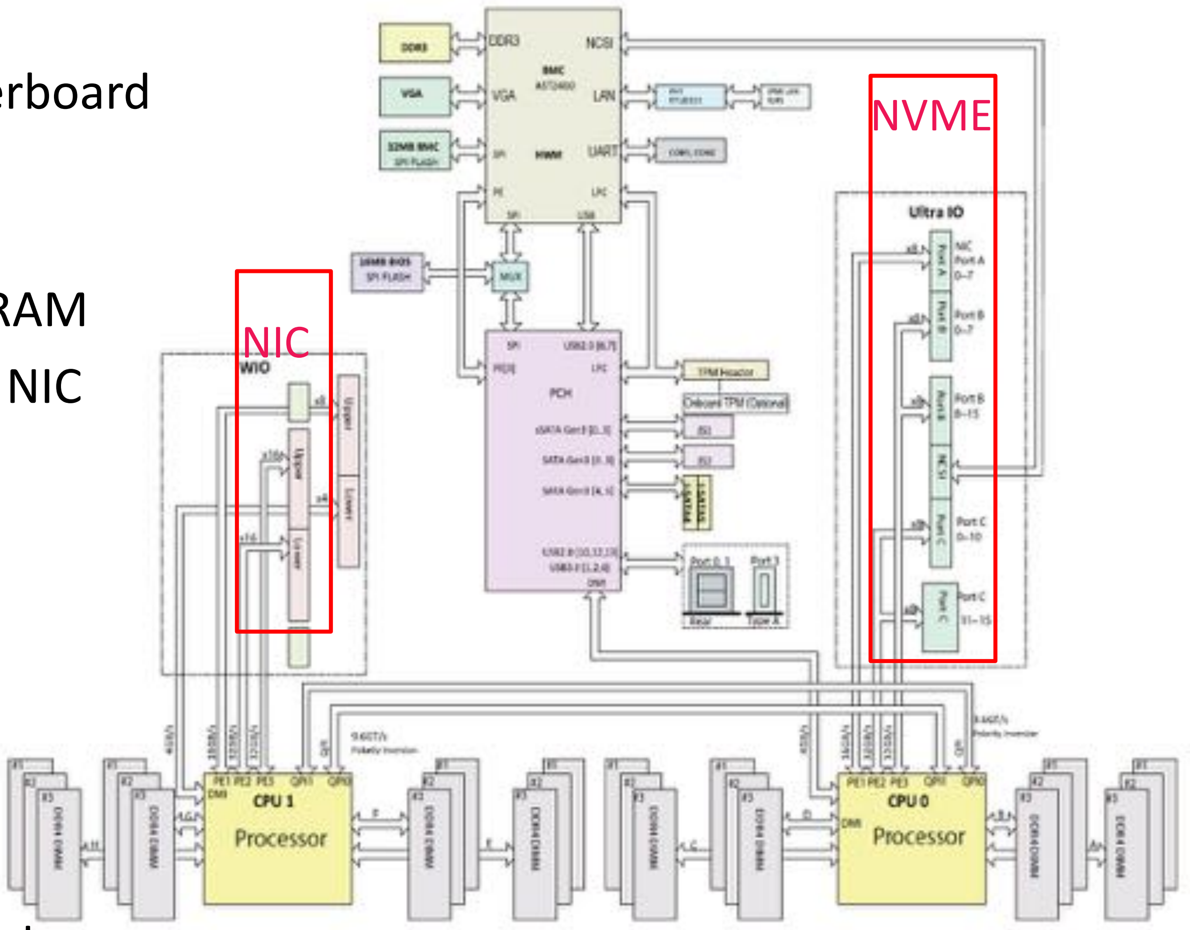
Network Topology Connecting the DTNs



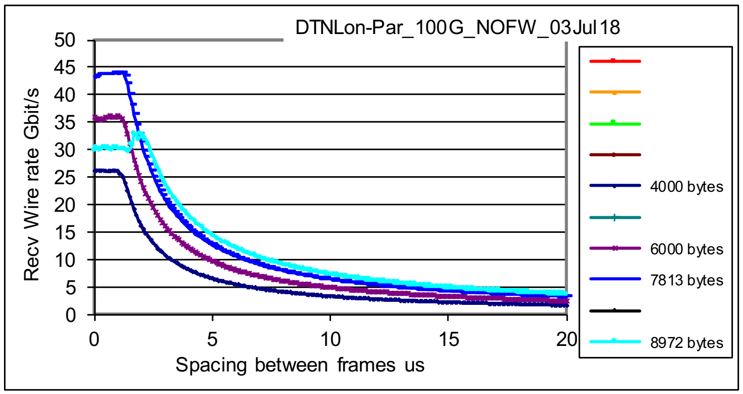
Can be Trunks
(multi VLANs)
e.g. Join LHCONE

The GÉANT DTN Hardware

- A lot of help from Boston Labs (London UK)
- Mellanox (UK & Israel)
- Supermicro X10DRT-i+ motherboard
- Two 6 core 3.40GHz
  Xeon E5-2643 v3 processors
- 128 GB DDR4 2133MHz ECC RAM
- Mellanox ConnectX-5 100 GE NIC
  - 16 lane PCI-e
  - As many interrupts as cores
  - Driver
    MLNX_EN 4.4-1.0.1.0
- 6 Intel DC3700 400 GB NVME
  - 8 lane PCI-e
- Fedora 28 with the
  4.13.9-300.fc27.x86_64 kernel

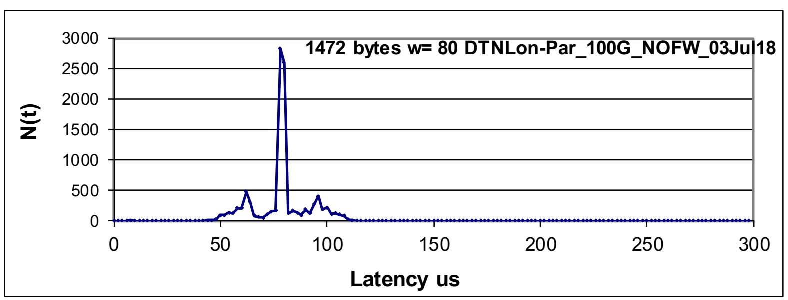


UDP Performance over GÉANT: Throughput & Jitter



Achievable UDP Throughput

- London to Paris over GÉANT
- No Firewall
- Core 6 is on the socket with the PCIe to the NIC
- ConnectX-5 NIC
- Rx ring buffer 8192
- Throughput 43 Gbit/s for 7813 Byte packet

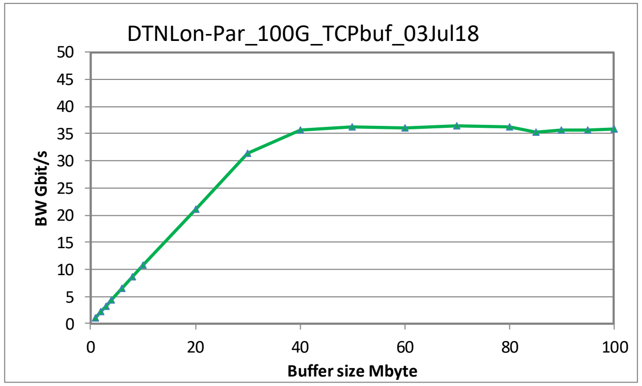


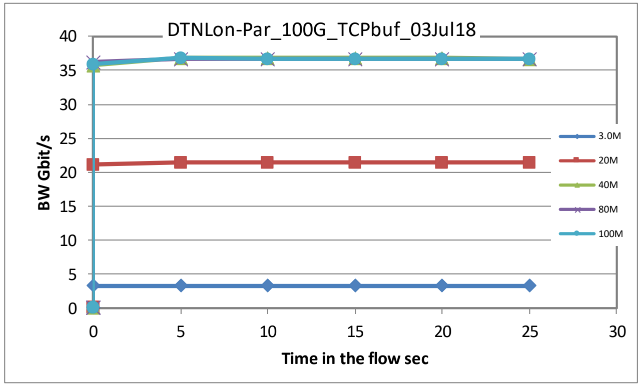
Inter- packet arrival times

- Jitter 4 µs FWHM
- Some side lobes at ± 16 µs due to cross traffic
- Good network stability.

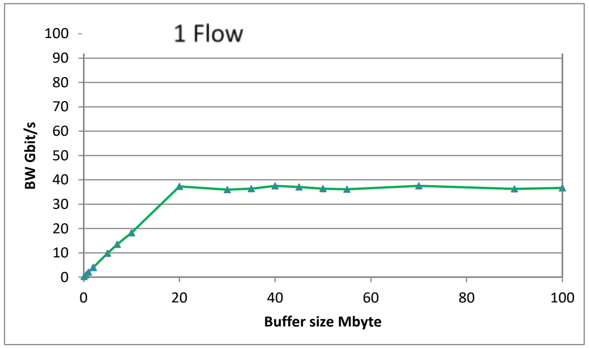
100 Gigabit TCP Performance GÉANT London to Paris

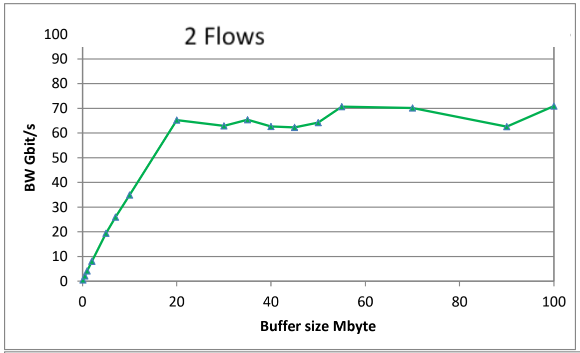
- Route: London-London2-Paris
- TCP offload on, TCP cubic stack
- Firewalls ON
- RTT 7.5 ms.
- Delay Bandwidth Product 93.8 MB for a 100 Gbit/s flow.
- One TCP flow rises smoothly to the 36 Gbit/s plateau at window of ~35 MBytes. (Includes Slowstart)
- Rate after slowstart 37.1 Gbit/s
  - Plateau from 5s onwards
- NO TCP re-transmitted segments
- Achievable throughput limited by CPU not DBP
  - Active core 100 % in kernel mode TCP buffer ≥ 40 MB
  - Lab tests got ~60 Gbit/s
  - FireWalls OFF improves by ~ 4 Gbit/s

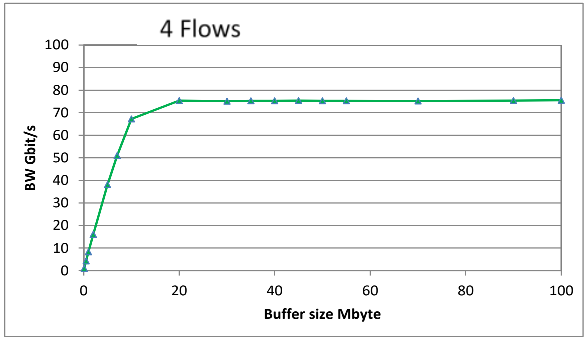


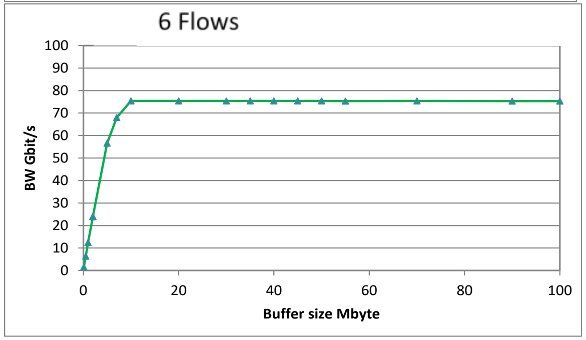


TCP Performance Multiple Flows London – Paris with iperf



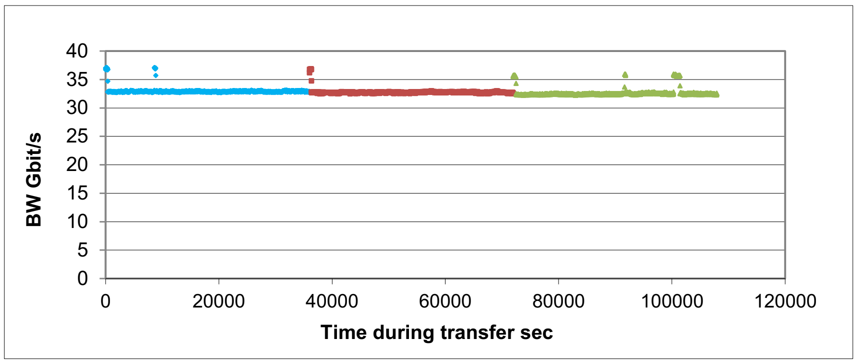


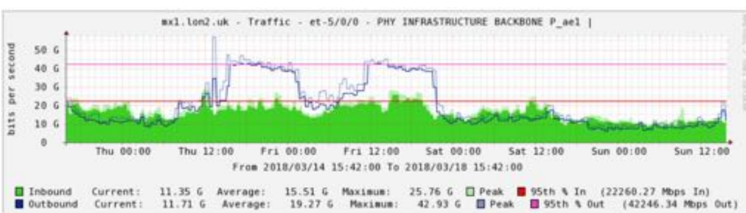




- Firewalls ON
- Each flow on a different core
- 2 flows reach 70 Gbit/s
  3 flows a stable 75 Gbit/s
- ≥ 3 flows 0.02 to 0.04 %
  TCP re-transmissions
- CPU usage important
  some cores ~80% kernel

TCP Performance London – Paris
32 Gbit/s Single Flow Over GÉANT



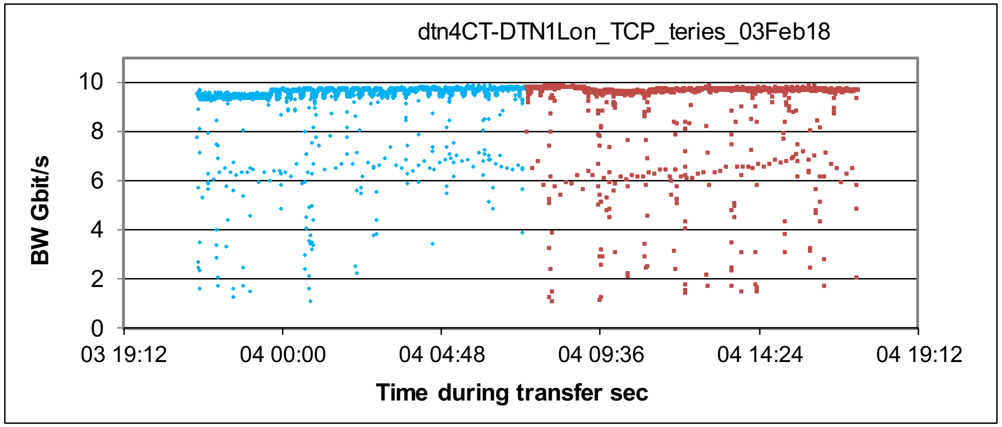


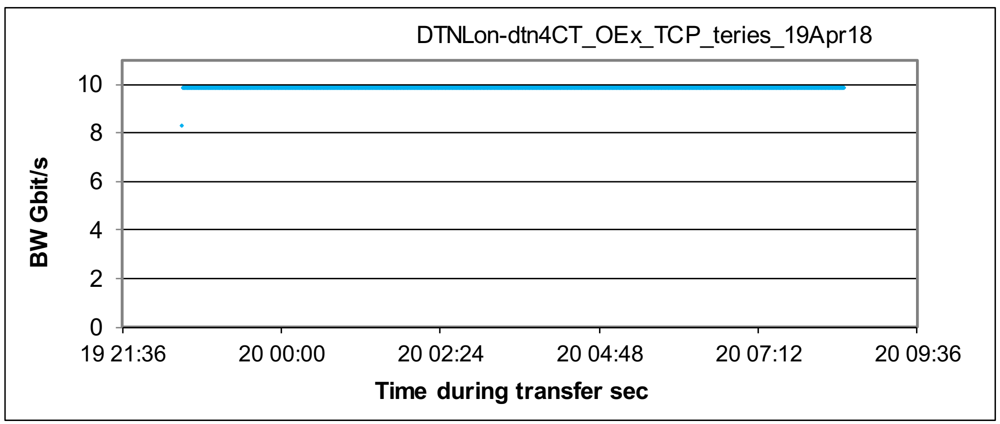
- RTT 7.5 ms
- TCP buffer size 40 MBytes
- TCP throughput over 30 Hrs
- 32.5 Gbit/s
- No TCP segment re-transmissions
- Very stable

The R&E Network Path used in the Tests

- With AARNet & SANReN tested inter-continental performance
  - Network tests – protocols & long haul effects – 10 & 100 Gigabit
  - Sustained data transfers

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

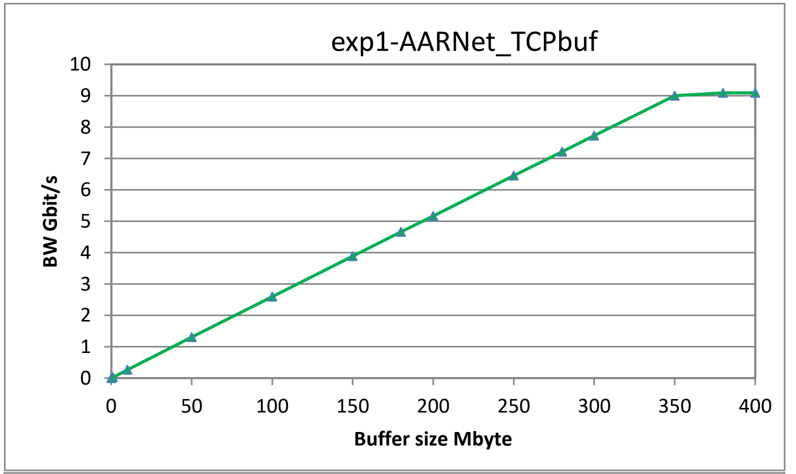
- **Direct link Open exchanges**
- Achievable TCP throughput over 10 Hrs
- Peak 9.9 Gbit/s
- No TCP re-transmits
- Representative of SKA path on WACS cable

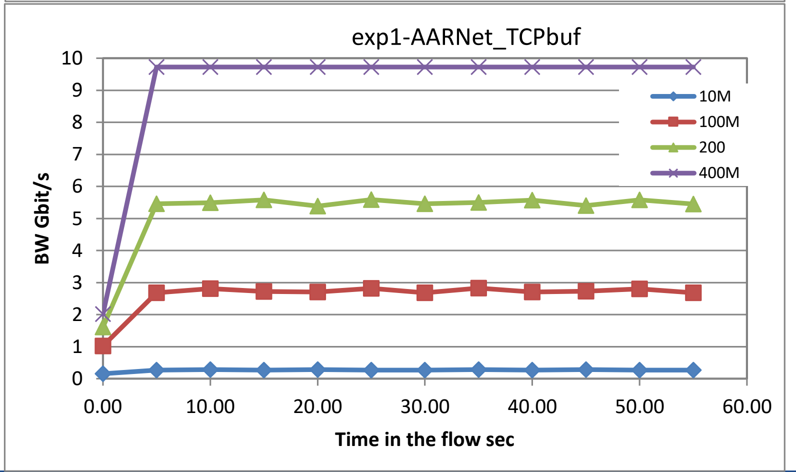
10 Gigabit TCP
GÉANT London to AARNet Canberra

- Route using ANA300 & AARNet 100Gig:
  London-Washington-Los Angeles-Sydney-Canberra
- TCP offload on, TCP cubic stack
- RTT 304 ms.
- Delay Bandwidth Product 280 MB.

- One TCP flow rises smoothly to the plateau at 350 MBytes.
- Throughput:
  - Average including slow start 9.09 Gbit/s
  - Plateau from 5s onwards 9.73 Gbit/s.

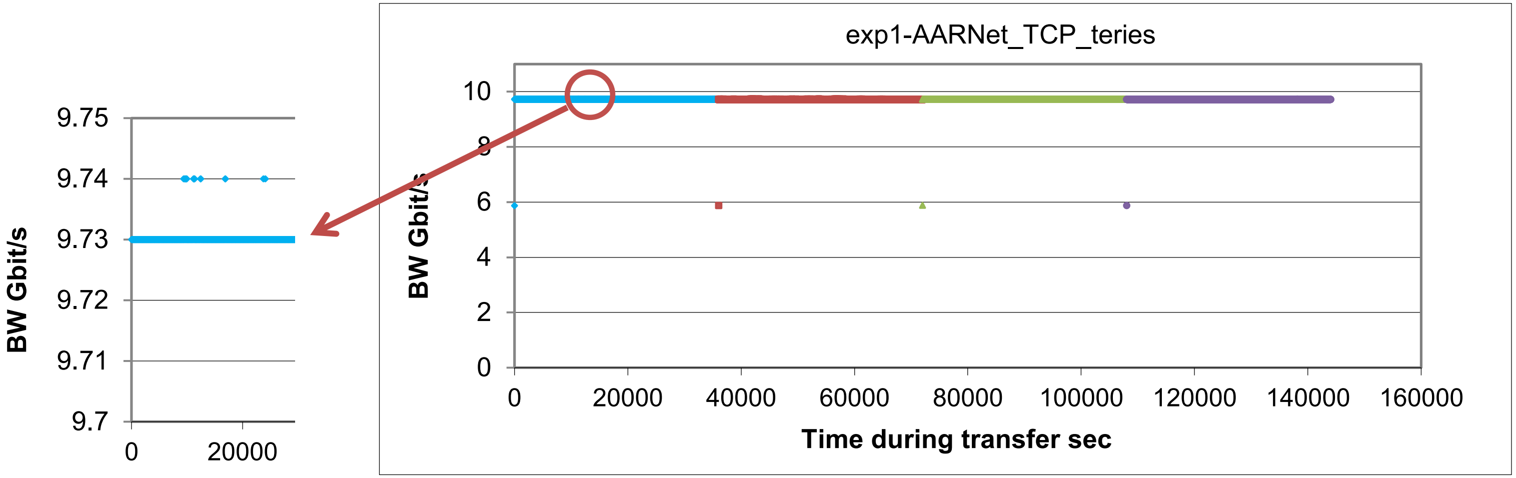
- NO TCP re-transmitted segments





Time Series of Achievable TCP Throughput GÉANT London to AARNet Canberra

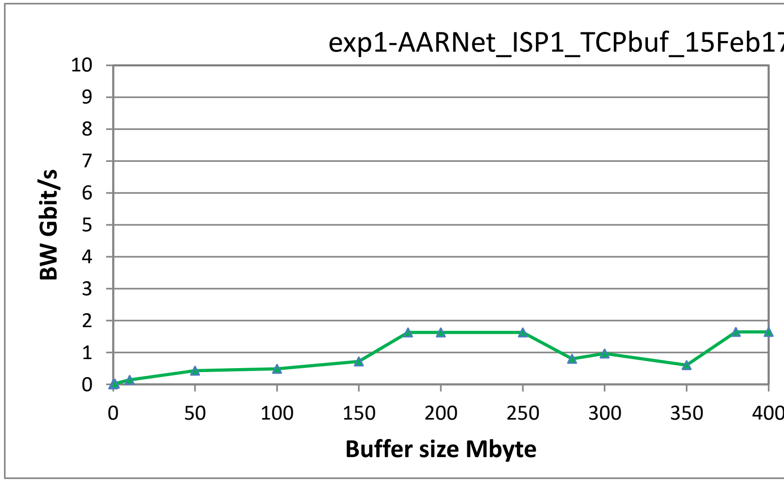
- Route using ANA300 & AARNet 100Gig:
  London-Washington-Los Angeles-Sydney-Canberra
- RTT 304 ms. Delay Bandwidth Product 280 MB.
- Achievable throughput recorded every 10s for 40 hours.
- A constant rate of 9.73 Gbit/s was achieved.
- Slow start points clearly visible at 6 Gbit/s at the start of every period.
- No TCP segment re-transmissions were observed during these tests.

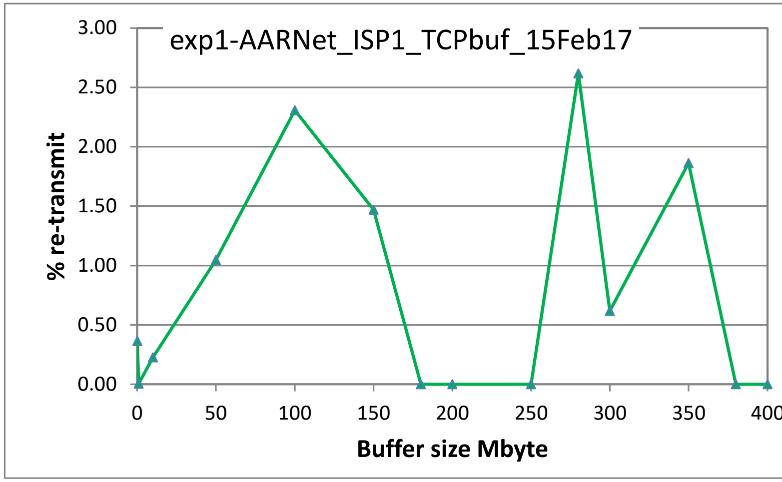


GÉANT London to AARNet Canberra using the ISP 1

- Route using ISP 1:
  London-(ISP 1 TransAtlantic & US LA)-AARNet-Canberra
- 10 GE NIC
- TCP offload on, TCP cubic stack
- RTT 304 ms.
- Delay Bandwidth Product 280 MB.

- One TCP flow reaches 1.6 Gbit/s.
- Considerable & variable number of re-transmitted segments to 2.5%
- Possible rate-limiting and DoS detection by the ISP

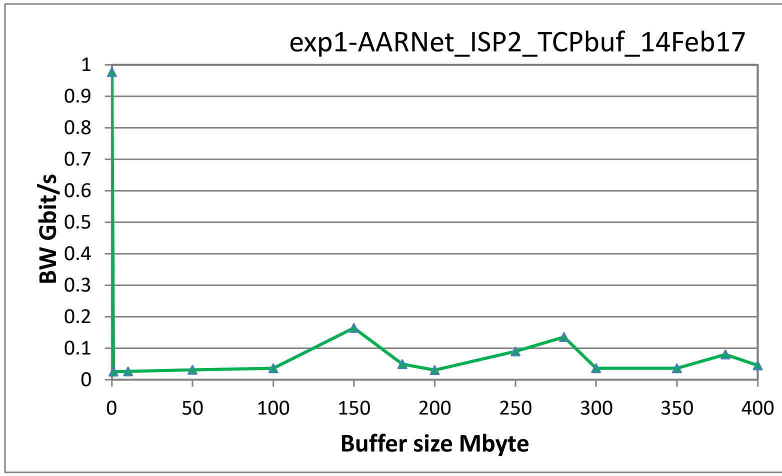


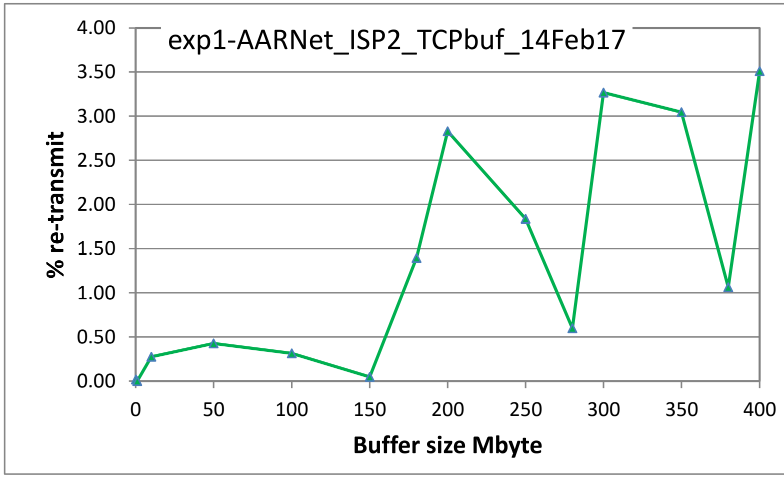


GÉANT London to AARNet Canberra using the ISP 2

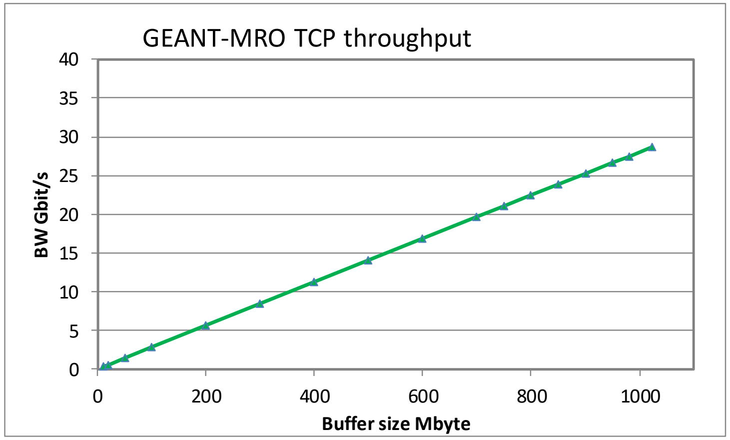
- Route using ISP 2:
  London-(ISP 2 TransAtlantic & US LA)-AARNet-Canberra
- 10 GE NIC
- TCP offload on, TCP cubic stack
- RTT 304 ms.
- Delay Bandwidth Product 280 MB.

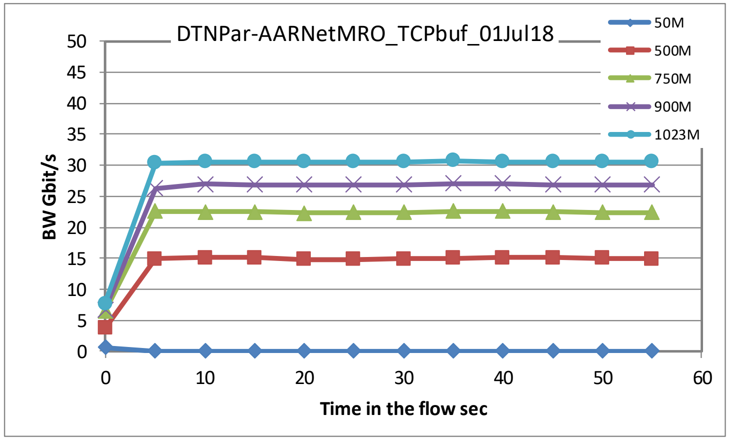
- One TCP flow nearly gets to 1 Gbit/s
- But the other flows just make 160 Mbit/s
- Considerable & variable number of re-transmitted segments
  3.2 – 3.5%
- Some tests ended early!
- Possible rate-limiting and DoS detection by the ISP





100 Gigabit between GÉANT Paris and AARNet MRO

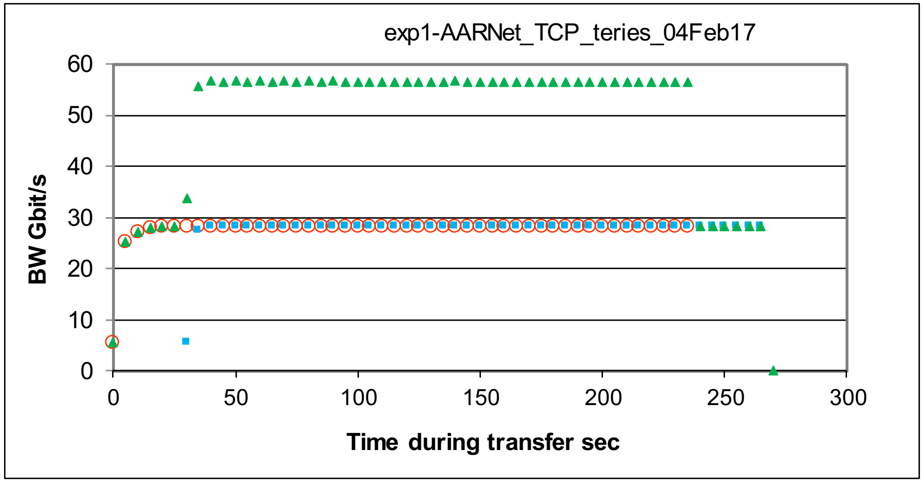




- Route GÉANT, ANA300, Internet2, & AARNet:
  Paris-New York-Seattle-LosAngeles-Sydney-Perth-MRO
- TCP offload on, TCP cubic stack
- Fedora 26 kernel 4.11.0-0.rc3.git0.2.fc26.x86_64
- RTT 279 ms.
- Delay Bandwidth Product 3.78 GB for 100 Gigabit
- One TCP flow rises smoothly to 28.7 Gbit/s
  at 1023 MBytes including slowstart.
- No TCP re-transmitted segments
- Rate after slowstart 30.6 Gbit/s

- Reach the limit of TCP protocol
  Max TCP window is 1 Gbyte
- Rate for RTT 279 ms and TCP window 1023 MB
  30.7 Gbit/s

100 Gigabit: Multiple flows between GÉANT and AARNet



- Route GÉANT, ANA300, Internet2, & AARNet:
  Paris-New York-Seattle-LosAngeles-Sydney-Canberra
- RTT 303 ms.
- TCP window 1023 MB.

- Two 4 minute TCP flows
- Second flow started 30s after the first

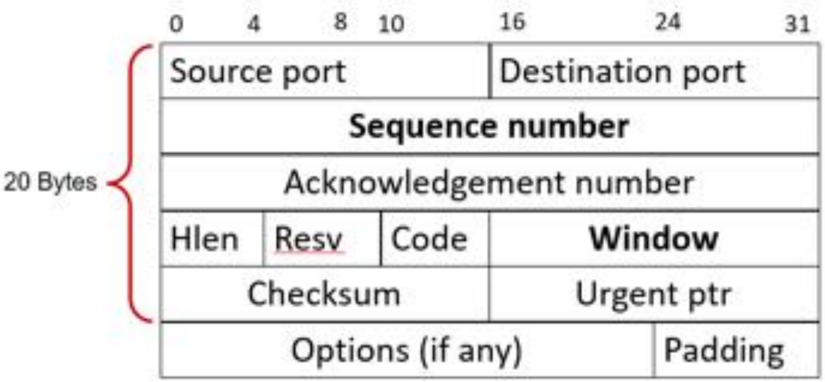
- Each flow stable at 28.3 Gbit/s
- Total transfer rate 56.6 Gbit/s
- 1.55 Tbytes data sent in 4.5mins.

- No TCP segments re-transmitted.
- Takes a substantial fraction of NREN BackBone

Demonstrates: large volume long distance transfers are possible.

The TCP Protocol Limit

TCP Header

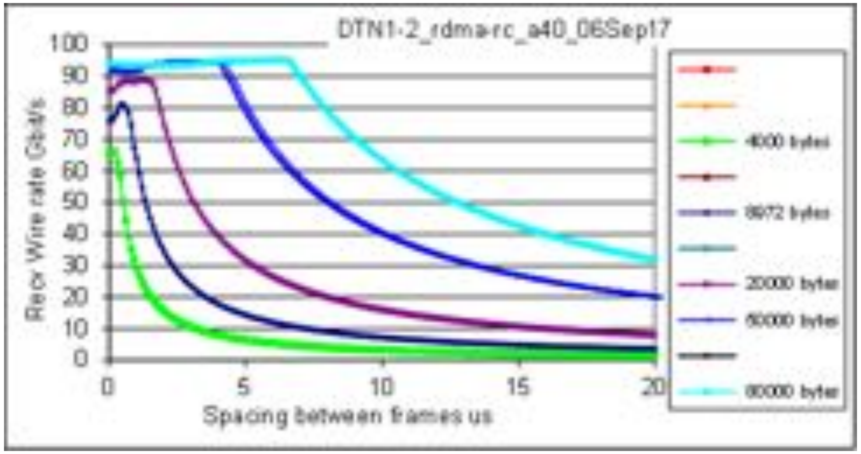


2^{32} → 4096 MB

2^{16} → 64 kB

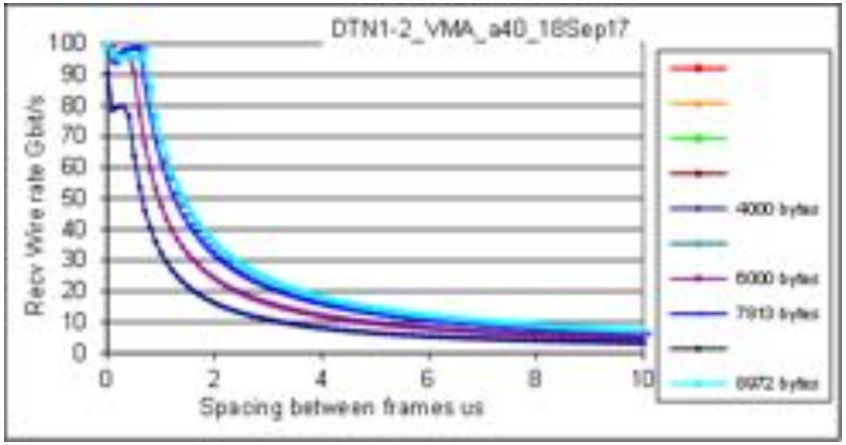
- To fix the Window size there is the Window Scale factor negotiated at the SYN exchange. RFC 7323 (obsoletes 1323)
- Max value 14 → max Window ($2^{16+}2^{14}$) → 1024 MB
- Window size < Sequence number
  - Deal with sequence number wrapping – every 0.33s
  - Allow to tell if a segment is old or new

RDMA RC and Kernel Bypass library libvma



RDMA

- Max packet size 4096 Bytes,
- Every message is acknowledged
- The CPU was 90% in user mode.
- App design needs to take care of ring buffers



libvma

- Over 95 Gigabits UDP
- The CPU mainly in user mode.
- Standard application
- Poor TCP performance

Investigation of other low level protocols

Troubleshooting Measurements and Tools

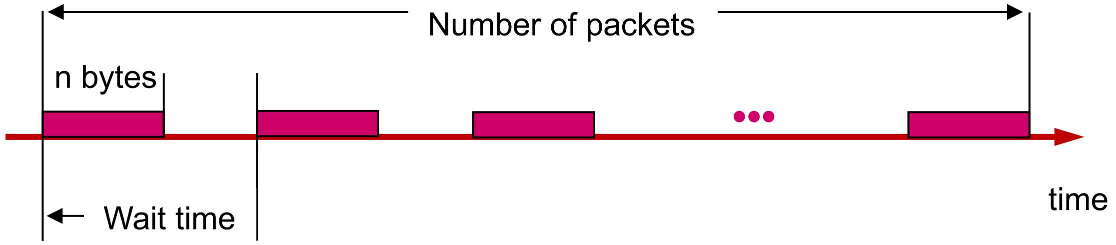
High Performance Data Transfers
What is important?

- 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

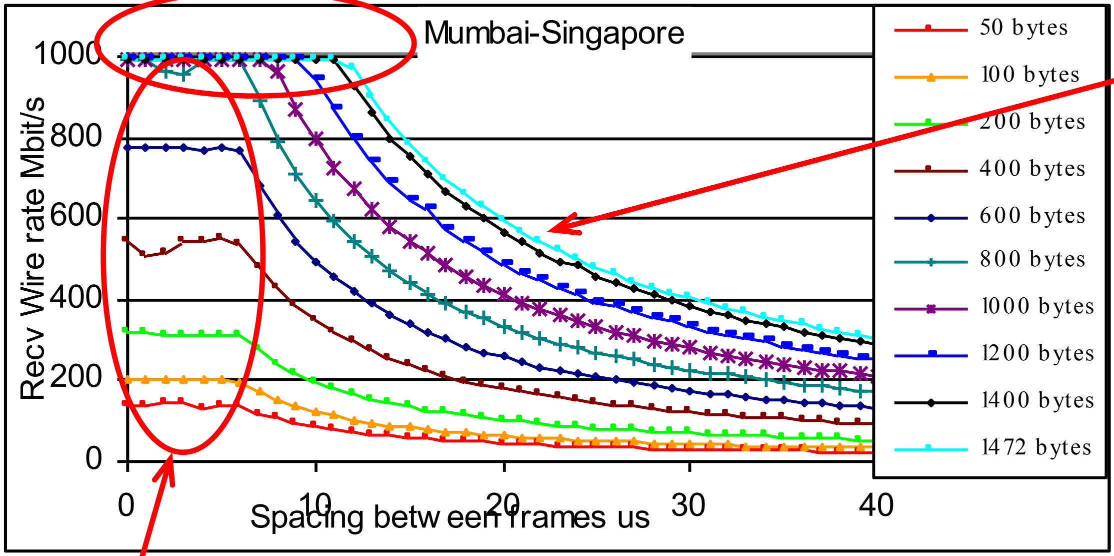
udpmon



- 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

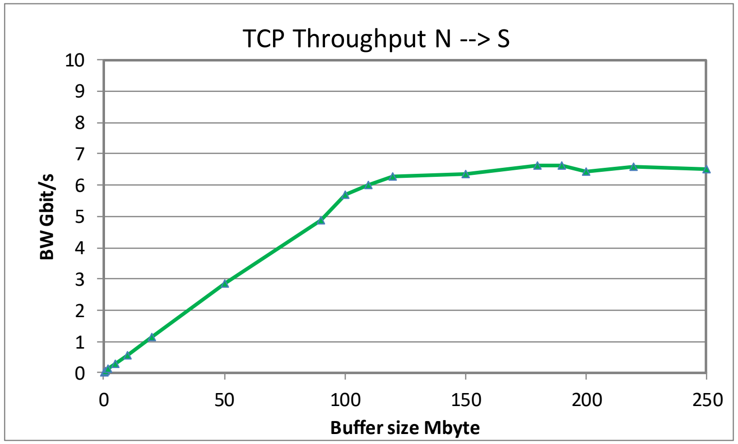


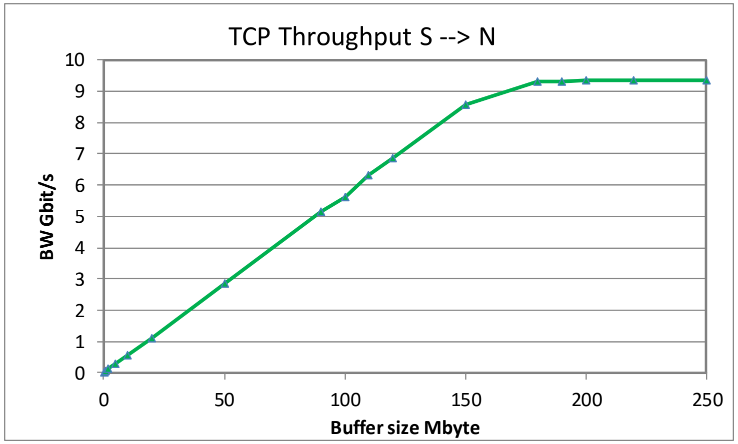
Shape follows 1/t
Packet spacing most important.

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

Troubleshooting
Some Examples

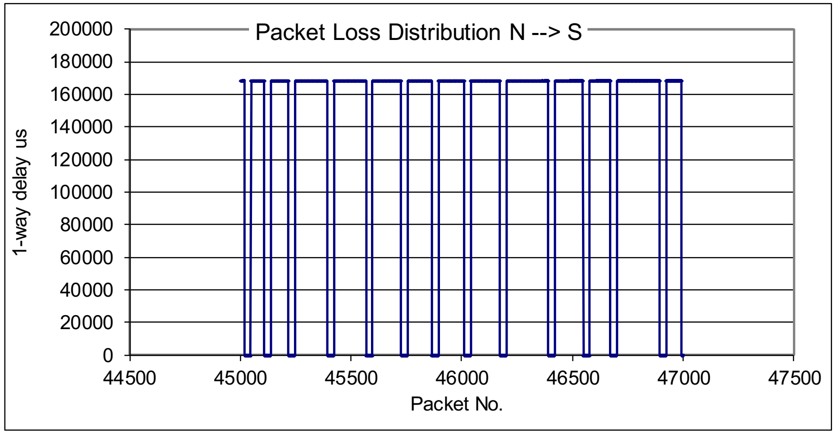
Asymmetrical Performance - Rate Policing





Re-transmits: 0.01% @ large buffers

Almost no re-transmits

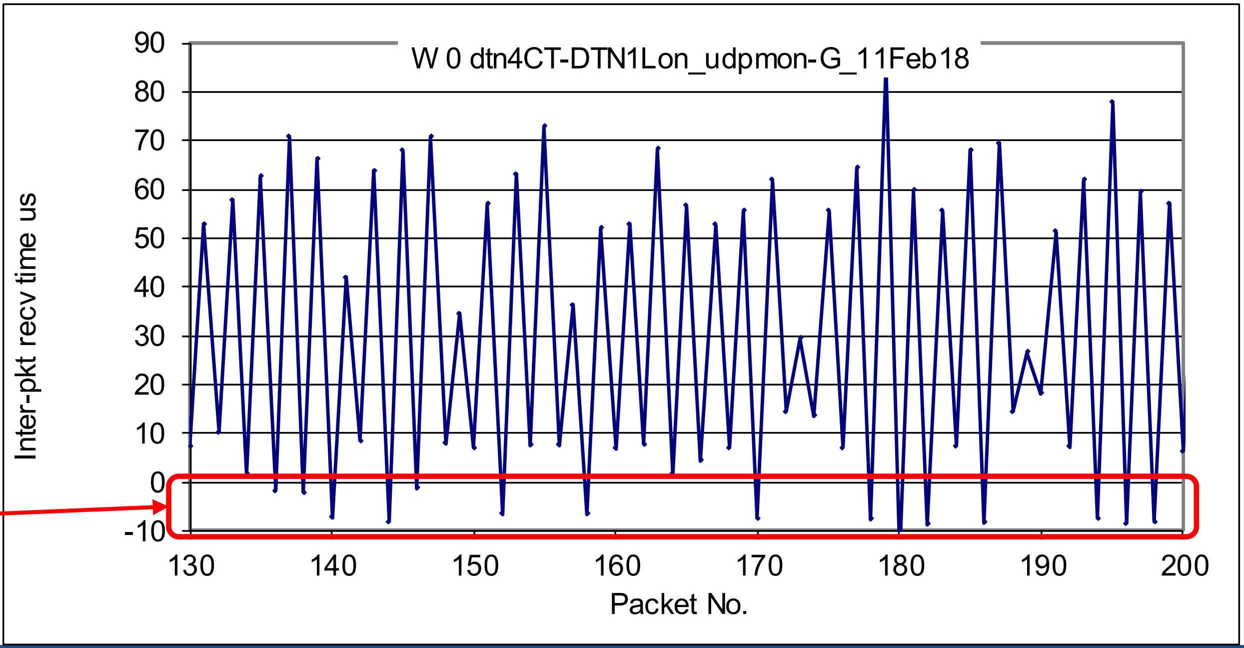


- UDP test
- Indication of rate control.
- Jumbo packets sent at 9.5 Gbit/s
- Recv at lower rate.
- Packets lost after 45,000 sent, ~340 ms
- Regular losses of ~25 frames.

Packets Out of Order

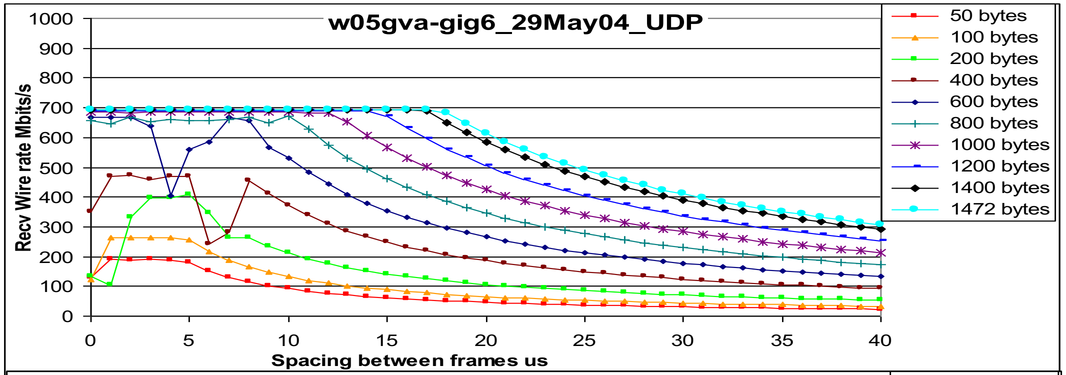
- Many packets out of order but none lost
- Examine with UDP stream
- Jumbo packets sent at 9.5 Gbit/s
- Negative inter-packet arrival times indicate packet arrived earlier than previous packet.
- Due to 10 GE bonding (lag) set-up.

Negative time - packet arrived earlier than previous

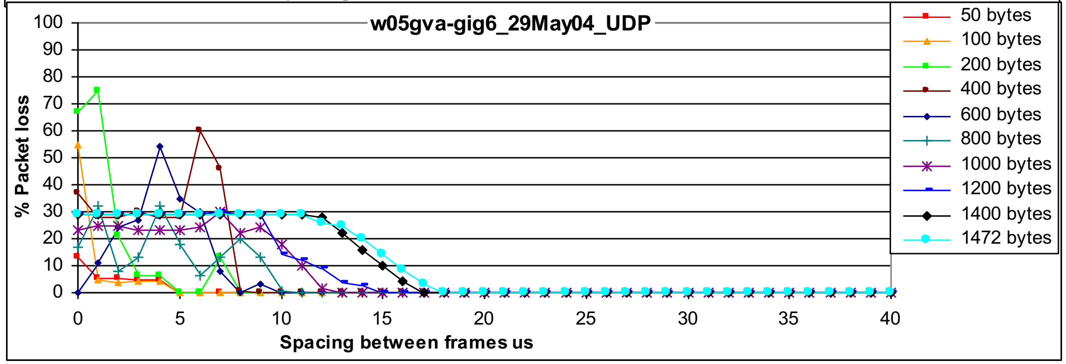


Network switch limits behaviour

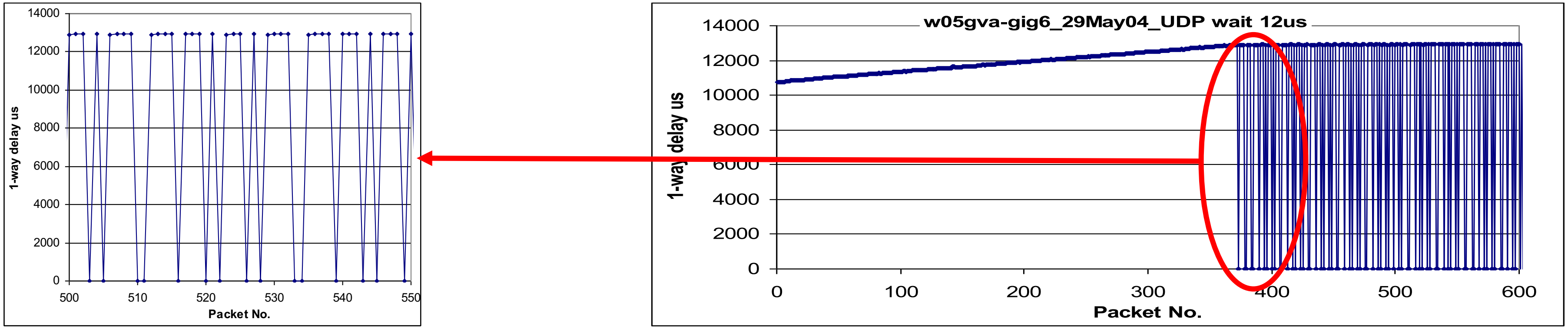
- End2end packets from udpmon
- Only 700 Mbit/s throughput



- Lots of packet loss

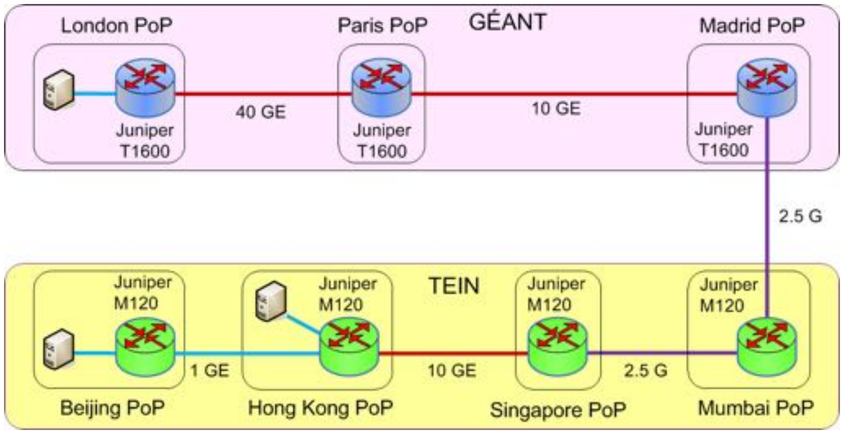


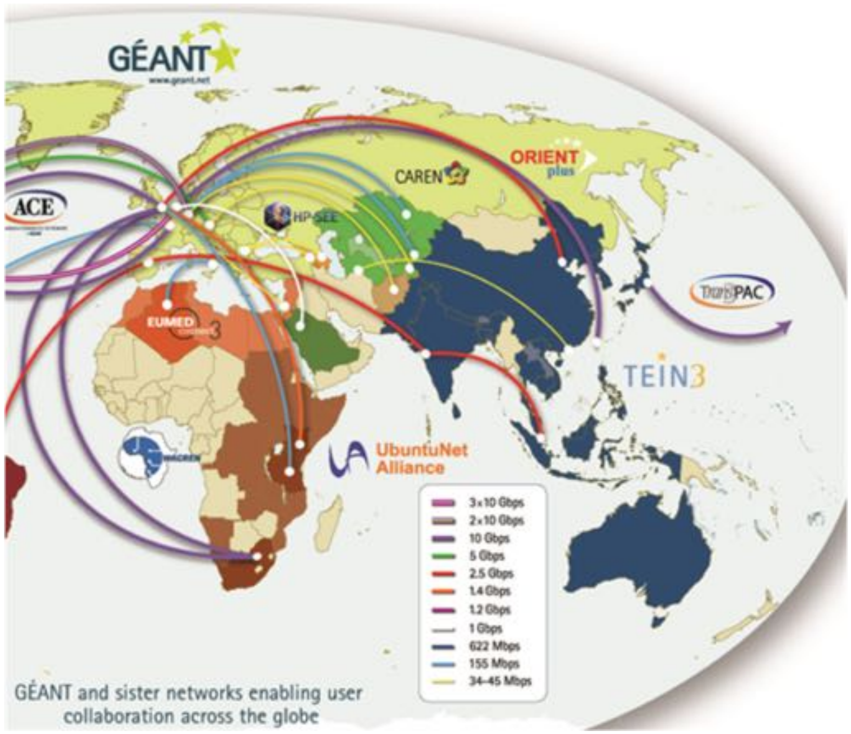
- 1-way delay & Packet loss distribution shows throughput limited



Stability of data transmission Europe – Asia-Pacific

- Stability tests made between London and Hong Kong.

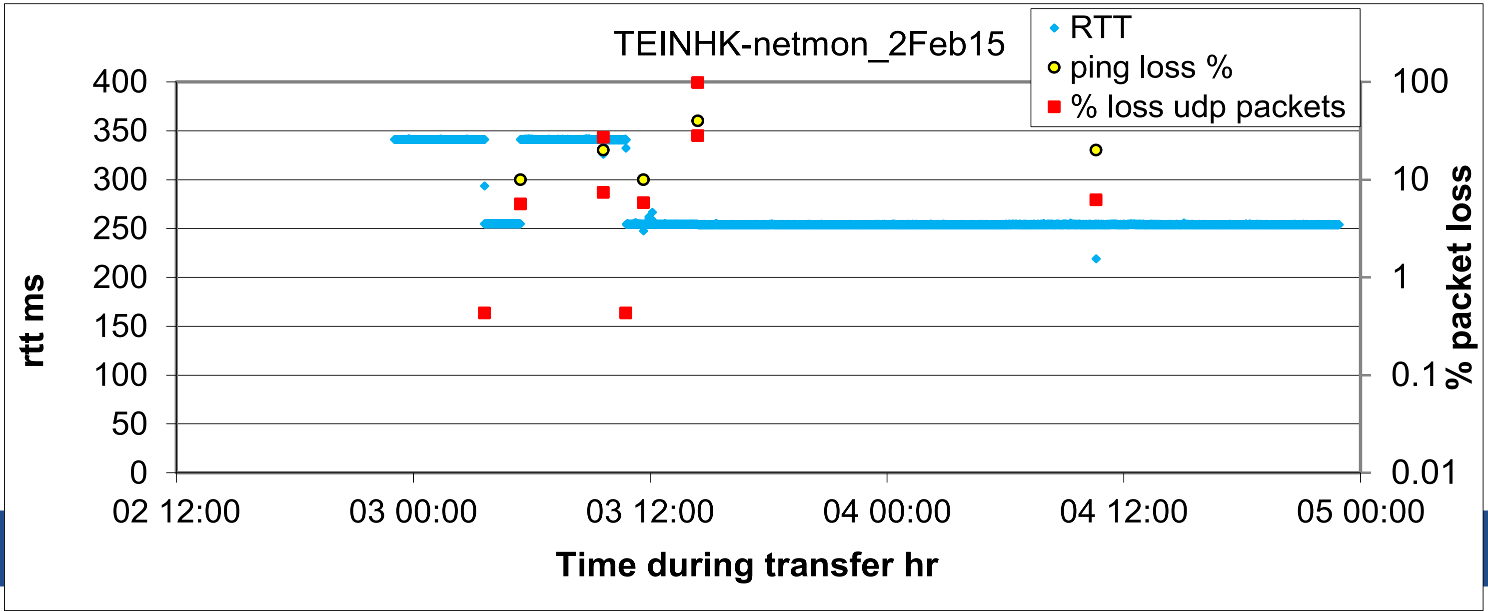




- Simultaneous time series tests
- Measure:
  - RTT & packet loss with ping
  - UDP throughput & packet loss udpmon

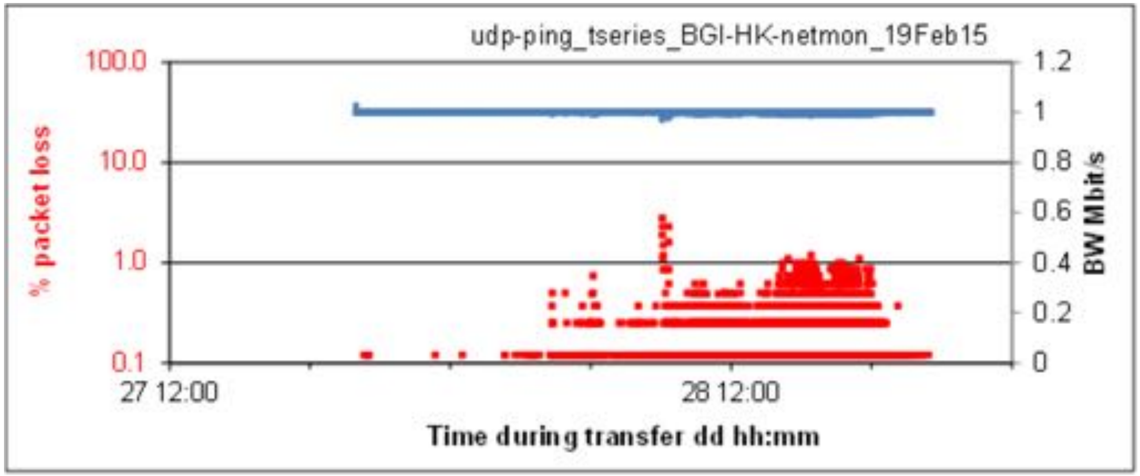
Testing link stability & data transmission Europe – Asia-Pacific Academic Path

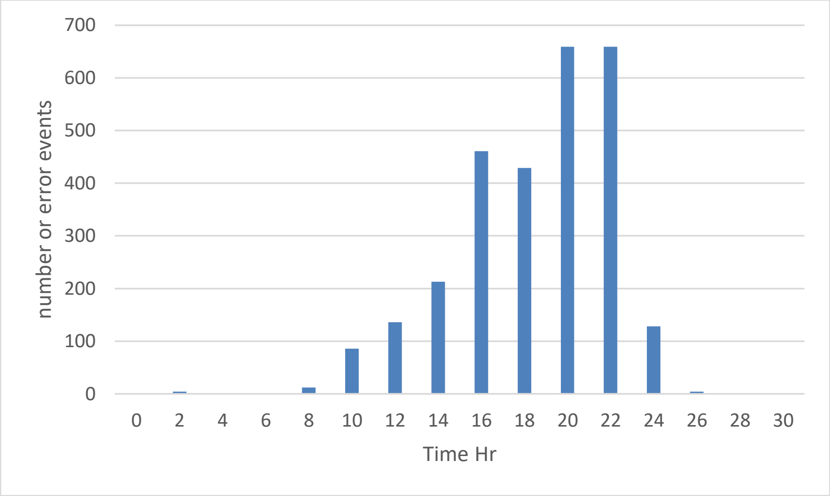
- Test made for 2 days 100 Mbit/s flow
- Many distinct periods of different round trip times
- Step changes to the RTT of between 220 µs and 87 ms.
- Several consecutive packets lost
- Correlation between step change in RTT and packet loss.
- micro-breaks caused by variation in the path at the SDH or optical layers of the links.
- TCP connections were not dropped.



Network Stability:
Hong Kong – London commercial route

- UDP 1 Mbit flow made for about a day.
- Packet loss events from udpmon occur at specific times.

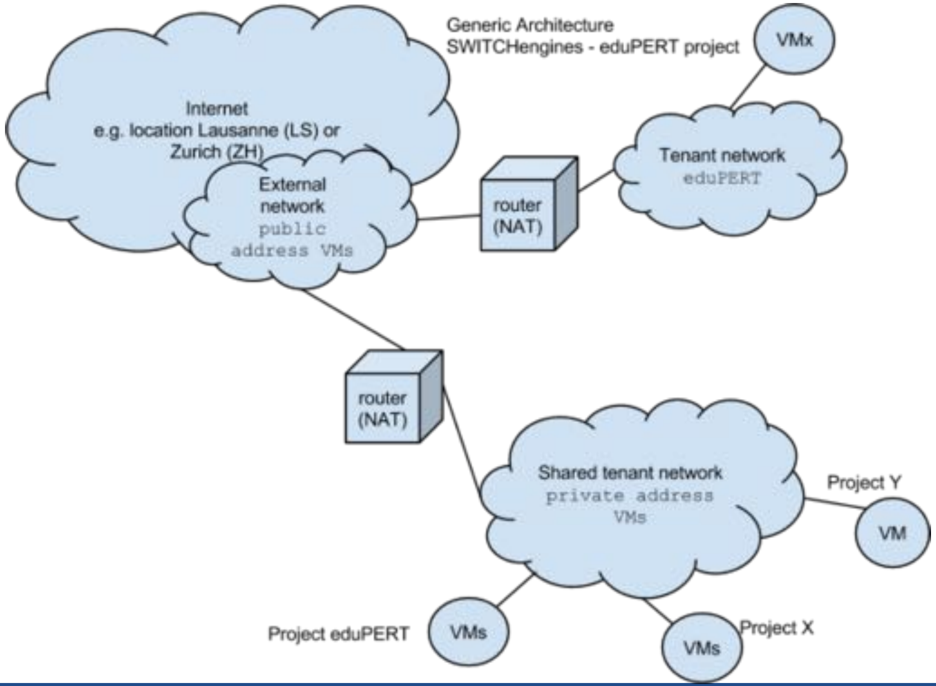




- Number of loss events per 2 Hr. period through the flow.

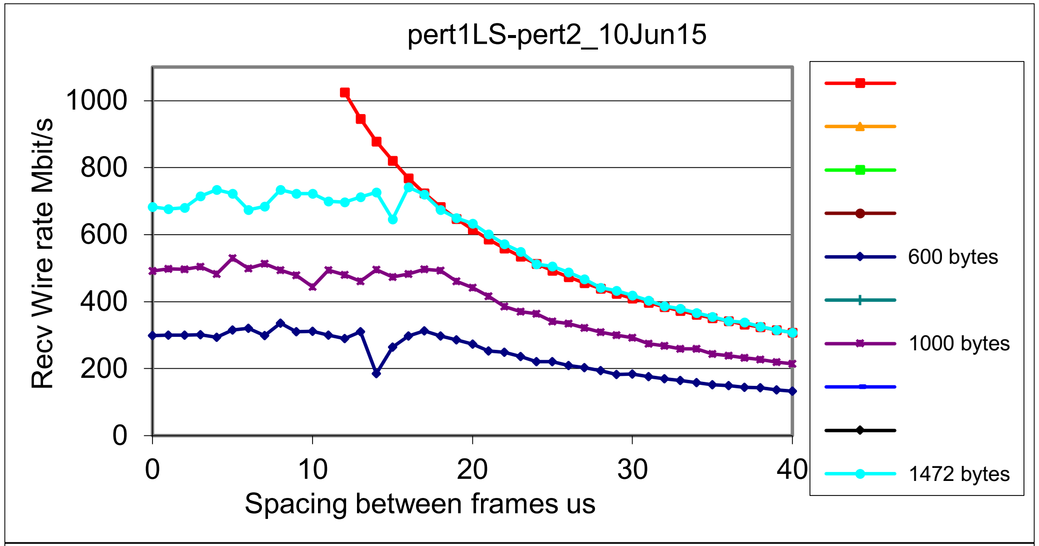
Testing VMs

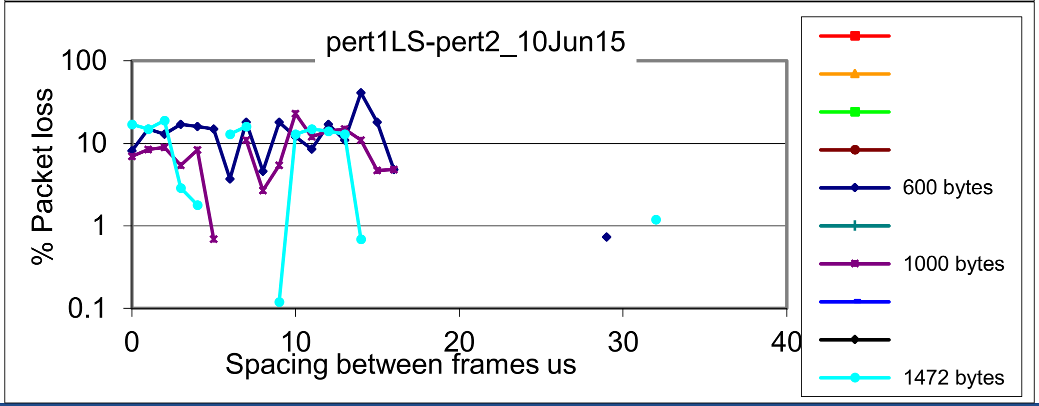
- Cloud environment put together by SWITCH to help eduPERT training
  - Based on OpenStack running ubuntu
  - 10 Gigabit Ethernet Hardware with private IP addresses
  - NAT devices enable controlled mapping of ports to VMs
  - 20 VMs available
- Tests made between 2 VMs in the same cloud.



udpmon on a VM cloud
Achievable Throughput & Packet loss

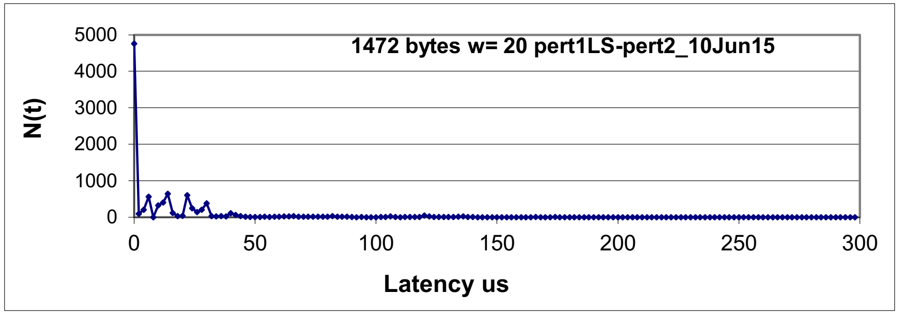
- Max throughput 700 Mbit/s
- Significant 10-20% packet loss packet spacing < 17 µs

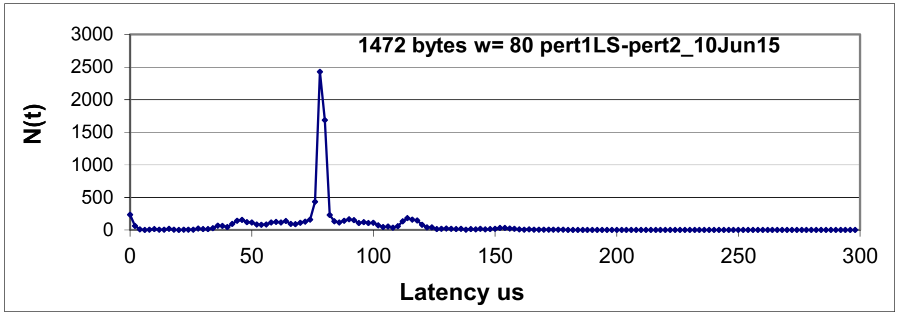


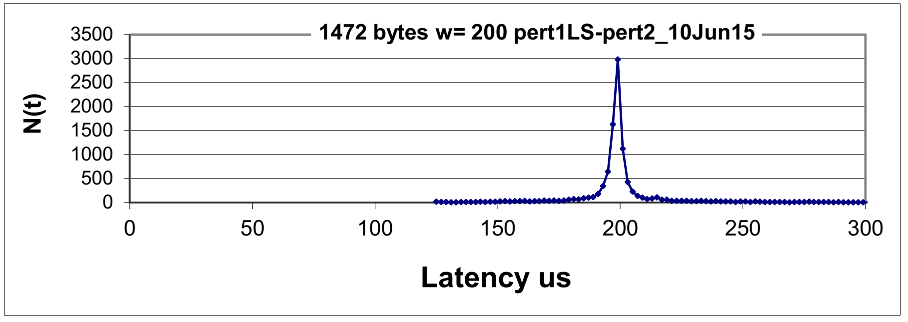


udpmon on a VM cloud
Packet Jitter

- UDP packets sent with spacing of 20, 80 and 200 µs.
- Interrupt coalescence was on for both VMs, confirmed by the peak at 2 µs.
- A FWHM of ~4 µs suggests the network was not heavily overloaded, but there is evidence of cross traffic.

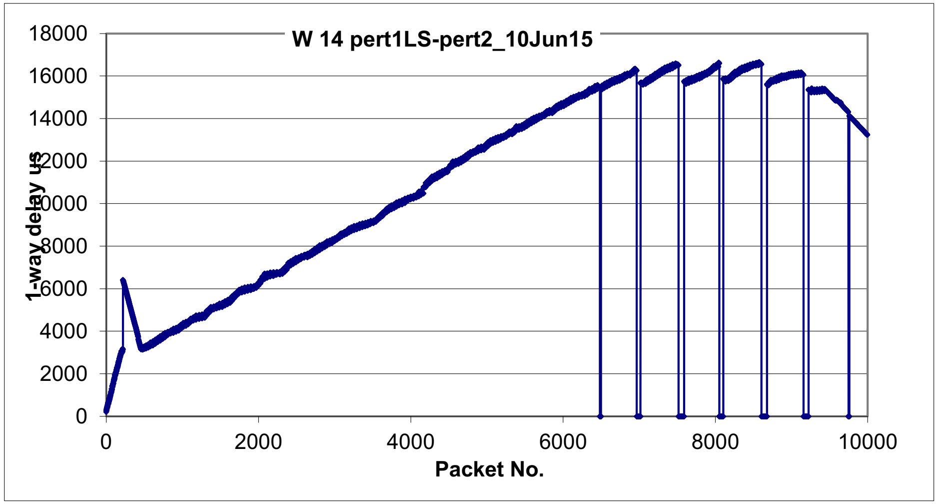


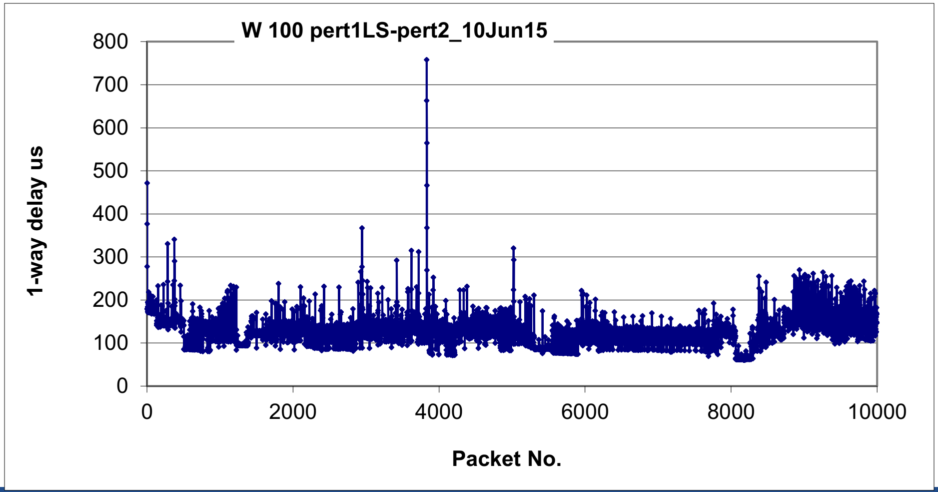




udpmon on a VM cloud
1-way delay

- Plot shows a classic bottleneck
  - Latency increasing
  - Regular packet loss
- 14 µs corresponds to a rate of ~880 Mbit/s.
- Good performance at 100 µs ~125 Mbit/s.

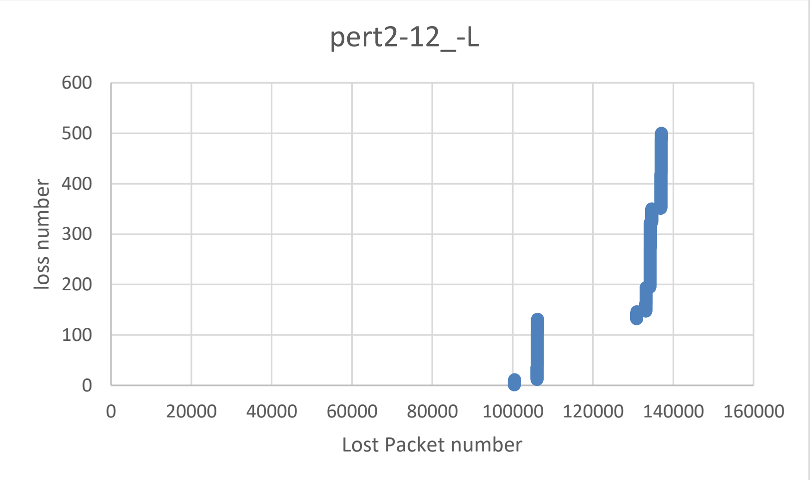


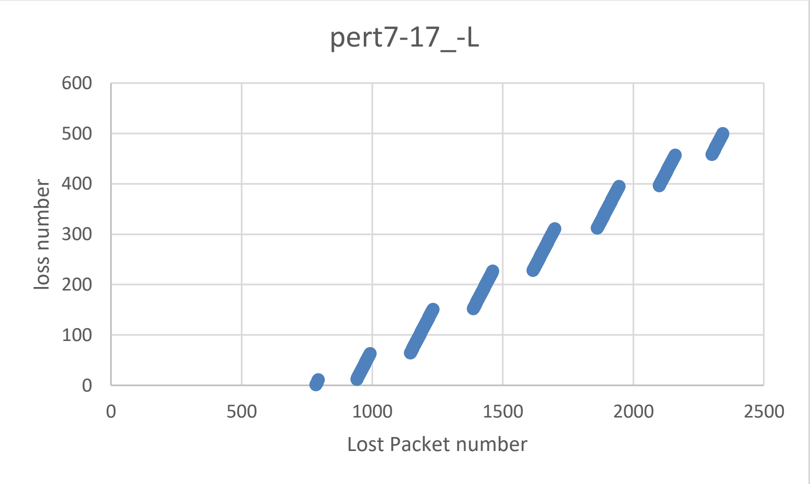


udpmon on a VM cloud packet loss distributions

- Tests made at about 490 Mbit/s
- 8 pairs of VMs used
- 1M packet sent
- Different types of packet loss distributions.

- Seems like you need care using clouds for data transfers.





Questions ?

Advanced European Network of E-infrastructures for Astronomy with the SKA AENEAS - 731016

Thanks to Richard Hughes-Jones

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 731122 (GN4-2).

www.geant.org Richard.Hughes-Jones@geant.org

NVMe Disks

- Non Volatile Memory express
  – a scalable host controller interface.
- Designed for SSD attached to PICe PCIe cards or 2.5” drives.
- Block IO based – lockless block layer
- Shorter data path bypasses costly AHCI / SCSI layers
- Latency & CPU cycles reduced > 50%
  - SCSI 6 µs 19,500 cycles
  - NVMe 2.8 µs 9,100 cycles
- Parallelism - per CPU HW queues: