STEP-1: TCP Basics & Science DMZ
May 23, 2024

David Wheeler, NCSA / University of Illinois

OpenCI Award: OCI 21-38296

Outline

- Science DMZ
- TCP
- Science DMZ Variations
- Buffer Investigation

Science DMZ

Science DMZ Purpose

The Science DMZ has two main purposes:

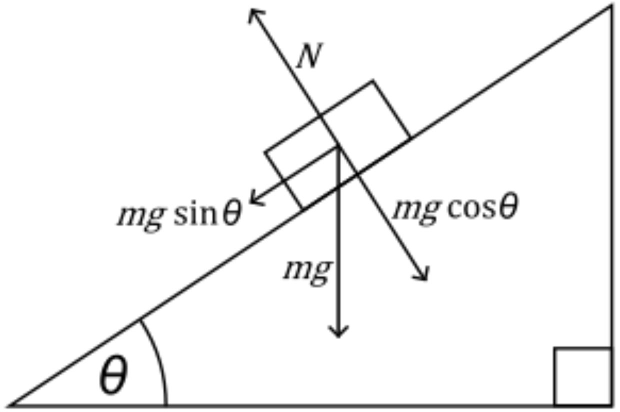
- A security architecture which allows for better segmentation of risks, and more granular application of controls to those segmented risks. In the case of the Science DMZ, the goal is to limit the segmented risk profile to that of a dedicated data transfer host.
- An attempt to streamline tuning and troubleshooting by removing degrees-of-freedom (from both a reliability and performance perspective) from the active data transfer path.

The Science DMZ in One Slide

Consists of **four key components**, all required:

- ***"Friction free" network path***
    - Highly capable network devices (wire-speed, deep queues)
    - Virtual circuit connectivity option
    - Security policy and enforcement specific to science workflows
    - Located at or near site perimeter if possible
- ***Dedicated, high-performance Data Transfer Nodes (DTNs)***
    - Hardware, operating system, libraries all optimized for transfer
    - Includes optimized data transfer tools such as Globus and GridFTP
- ***Performance measurement/test node***
    - perfSONAR
- ***Once it's up, users often need training – Engagement is key***

© 2023 Globus



© 2023 Wikipedia

perfSONAR

Science DMZ Background

- The data mobility performance requirements for data intensive science are beyond what can typically be achieved using traditional methods
  - Default host configurations (TCP, filesystems, NICs)
  - Converged network architectures designed for commodity traffic
  - Conventional security tools and policies
  - Legacy data transfer tools (e.g. SCP)
  - Wait-for-trouble-ticket operational models for network performance

Science DMZ Background

- The Science DMZ model describes a performance-based approach
  - Dedicated infrastructure for wide-area data transfer
    - Well-configured data transfer hosts with modern tools
    - Capable network devices
    - High-performance data path which does not traverse commodity LAN
  - Proactive operational models that enable performance
    - Well-deployed test and measurement tools (perfSONAR)
    - Periodic testing to locate issues instead of waiting for users to complain
  - Security posture well-matched to high-performance science applications

TCP

TCP – Ubiquitous and Fragile

- Networks provide connectivity between hosts
  - From an application's perspective, the interface to "the other end" is a socket
  - Communication is between applications – mostly over TCP
- TCP – the fragile underpinning
  - TCP is (for very good reasons) timid – packet loss is interpreted as congestion
  - Packet loss in conjunction with latency is a performance killer
  - Like it or not, TCP is used for the vast majority of data transfer

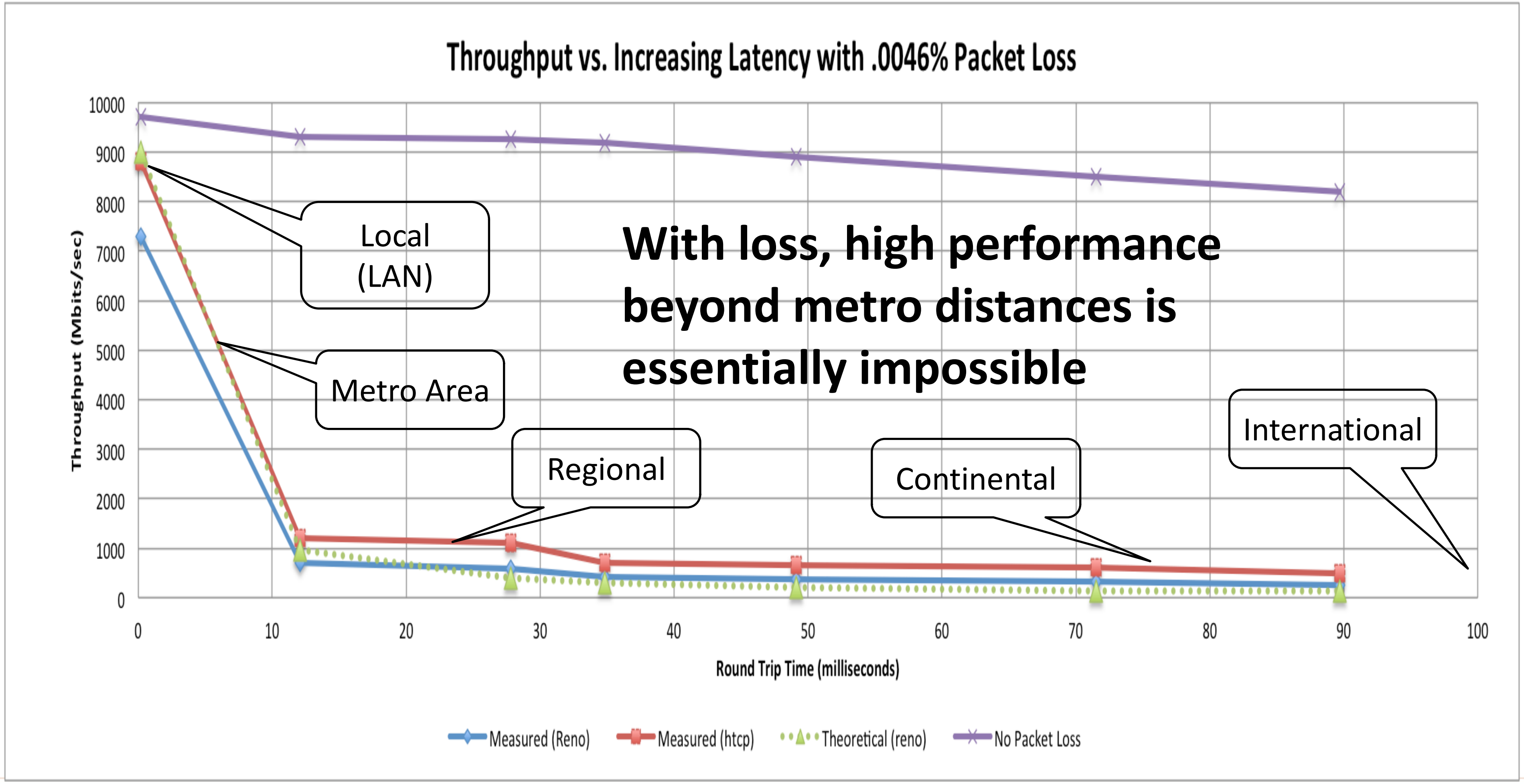
Packet and Data Loss

- Isn't TCP reliable?
- We are going to talk about this a lot
  - The data isn't lost forever, it's dropped somewhere on the path
  - Usually by a device without enough buffer space to accept it, or by a device that concludes the data is corrupted, so it won't send it
- Once it's dropped, we have a way of knowing it's been dropped
  - TCP is reliable, each end is keeping track of what was sent, and what was received
  - If something goes missing, it's resent
  - Resending is what takes the time and causes the slowdown

Packet and Data Loss

- TCP reliably and transparently recovers from packet loss by retransmitting lost packets
  - This is how it provides a reliable data transfer service to the applications which use it: e.g., Web, Email, GridFTP, perfSONAR, etc
  - The reliability mechanisms dramatically **reduce performance** when they are exercised
- We want to eliminate the causes of packet loss – so that we don't need to test out the (slow) way that TCP can recover
- **What is the impact of that recovery**?

A little packet loss makes a large difference in TCP performance



How Do We Accommodate TCP?

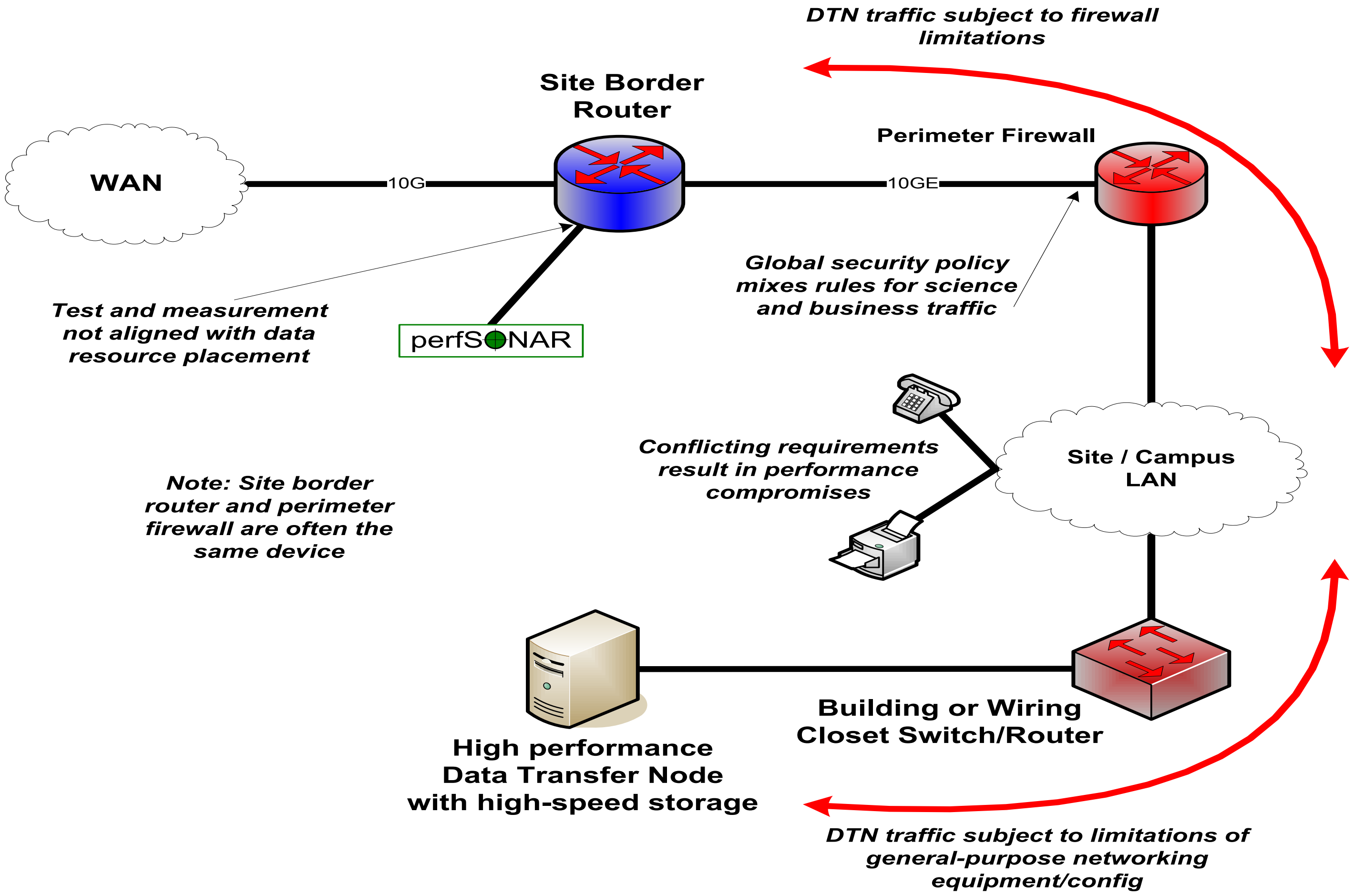
- High-performance wide area TCP flows must get loss-free service
  - Sufficient bandwidth to avoid congestion
  - Deep enough buffers in routers and switches to handle bursts
    - Especially true for long-distance flows due to packet behavior
    - No, this isn’t buffer bloat
- Equally important – the infrastructure must be verifiable so that clean service can be provided
  - Stuff breaks
    - Hardware, software, optics, bugs, etc.
    - How do we deal with it in a production environment?
  - Must be able to prove a network device or path is functioning correctly
    - Regular active tests should be run - perfSONAR
  - Small footprint is a huge win
    - Fewer the number of devices means it is easier to locate the source of packet loss

Science DMZ Variations

Science DMZ Has Many Forms

- There are a lot of ways to combine these features – it all depends on what you need to do
  - Small installation for a project or two
  - Facility inside a larger institution
  - Institutional capability serving multiple departments/divisions
  - Science capability that consumes a majority of the infrastructure
- Some of these are straightforward, others are less obvious
- Key point of concentration: eliminate sources of packet loss / packet friction

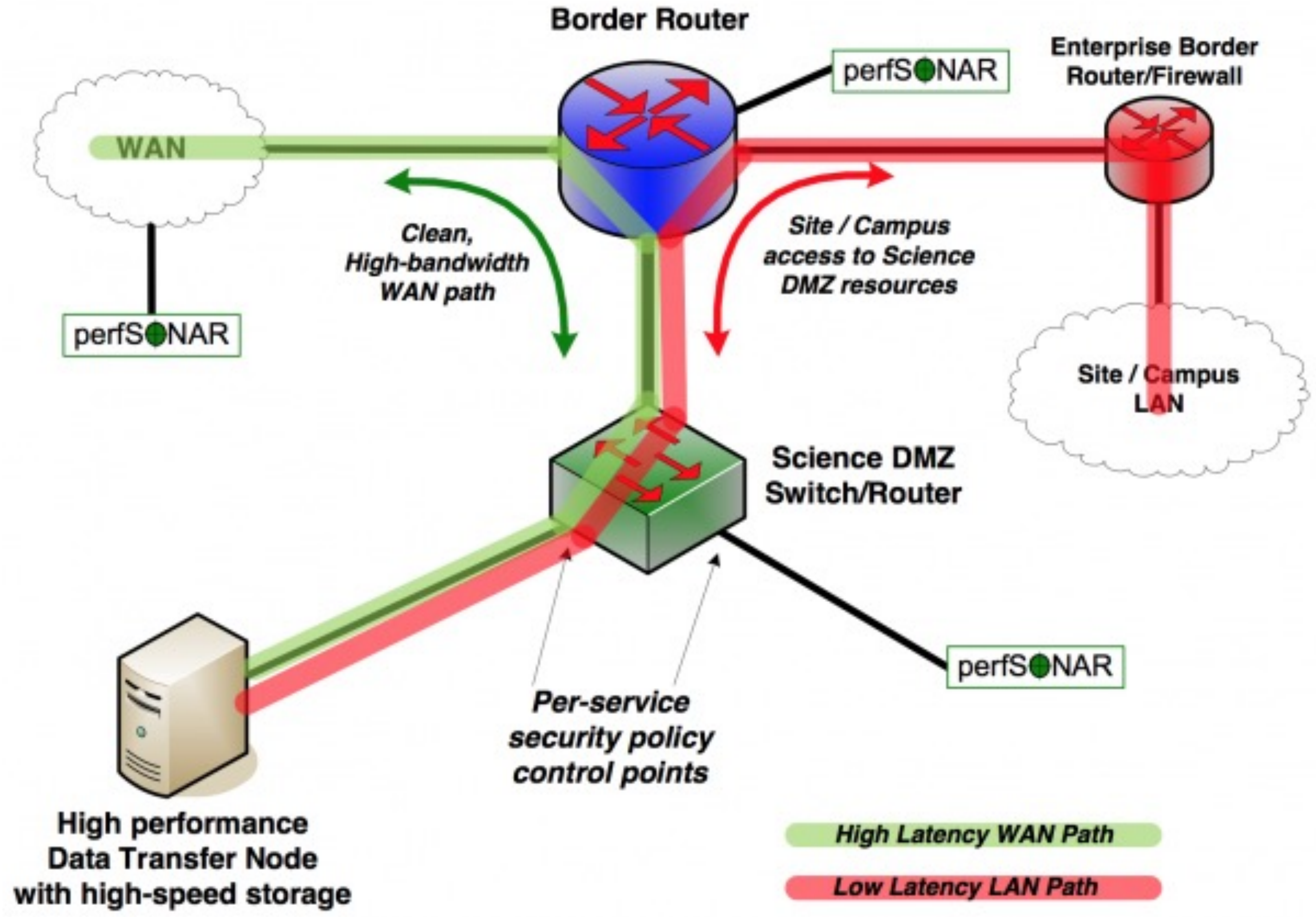
Legacy Method: Ad Hoc DTN Deployment



Legacy Method: Ad Hoc DTN Deployment

- This is often what gets tried first
- Data transfer node deployed where the owner has space
  - This is often the easiest thing to do at the time
  - Straightforward to turn on, hard to achieve performance
- If lucky, perfSONAR is at the border
  - This is a good start
  - Need a second one next to the DTN
- Entire LAN path needs to be sized for data flows
- Entire LAN path is part of any troubleshooting exercise
- This usually fails to provide the necessary performance.

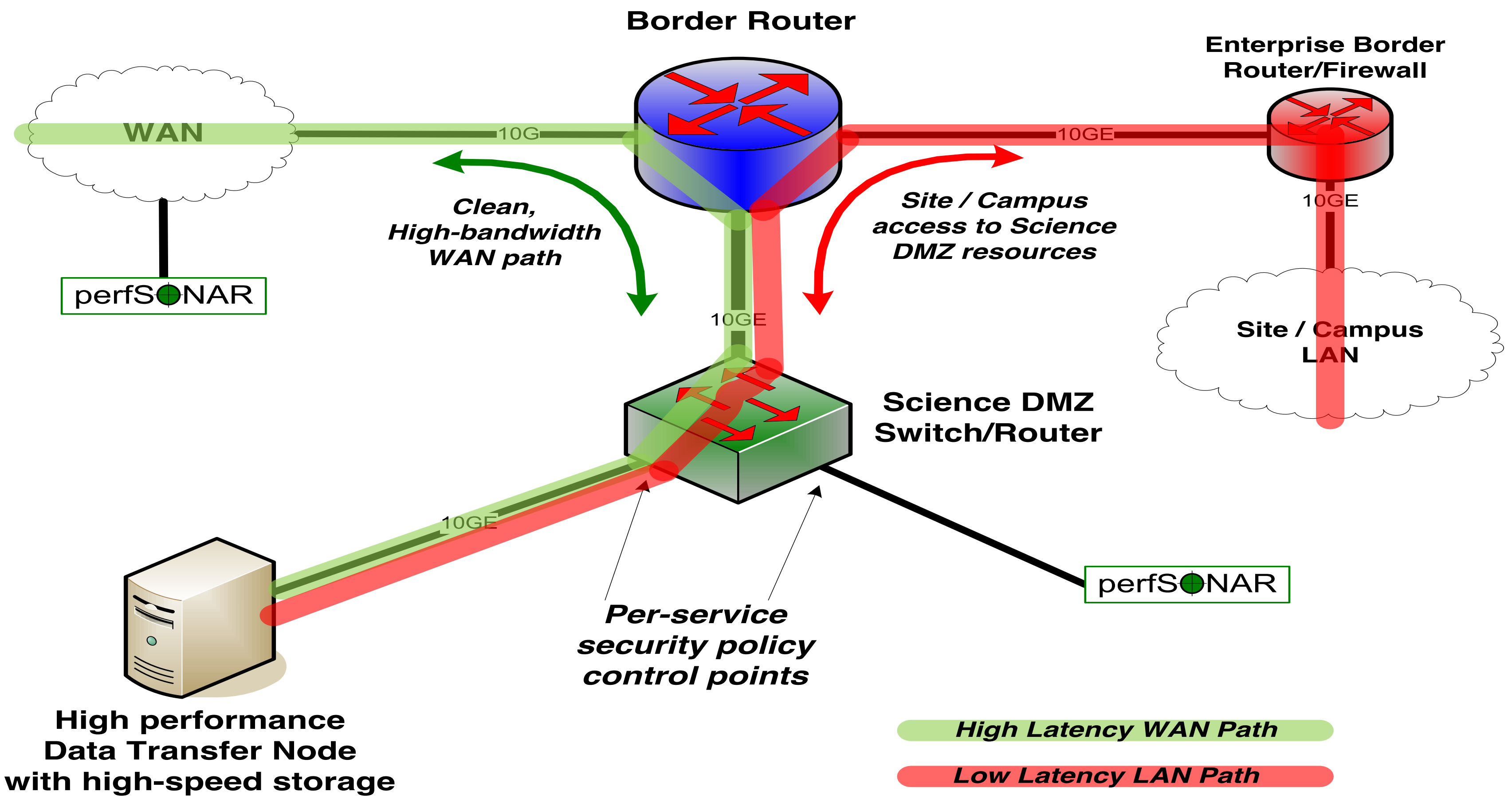
Better approach: simple Science DMZ



Small-scale Science DMZ Deployment

- Add-on to existing network infrastructure
  - All that is required is a port on the border router
  - Small footprint, pre-production commitment
- Easy to experiment with components and technologies
  - DTN prototyping
  - perfSONAR testing
- Limited scope makes security policy exceptions easy
  - Only allow traffic from partners
  - Add-on to production infrastructure – lower risk

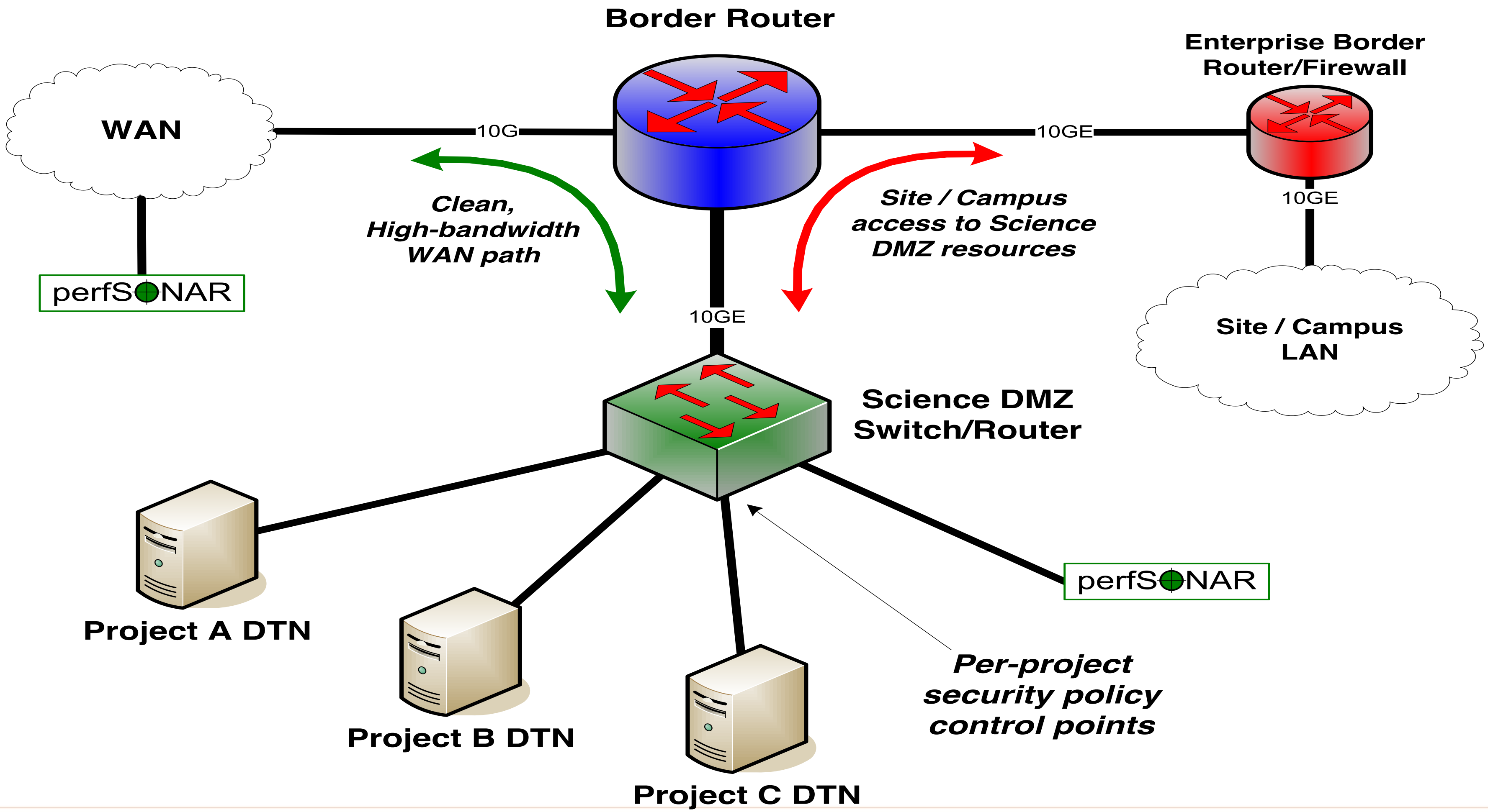
Prototype Science DMZ Data Path



Support For Multiple Projects

- Science DMZ architecture allows multiple projects to put DTNs in place
  - Modular architecture
  - Centralized location for data servers
- This may or may not work well depending on institutional politics
  - Issues such as physical security can make this a non-starter
  - On the other hand, some shops already have service models in place
- This can provide a cost savings – it depends
  - Central support for data servers vs. carrying data flows
  - How far do the data flows have to go?

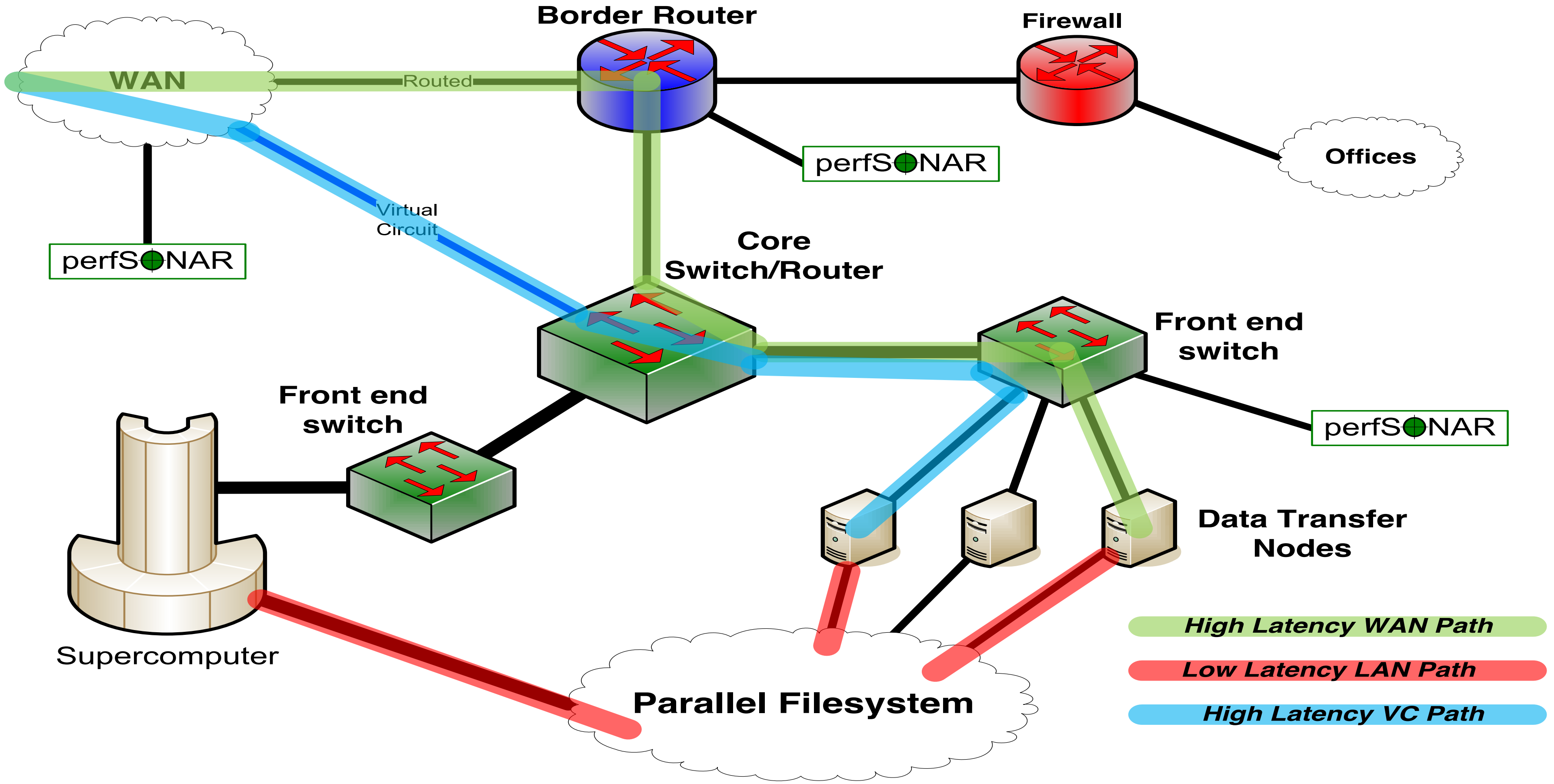
Multiple Project Support



HPC Center Deployment

- High-performance networking is assumed in this environment
  - Data flows between systems, between systems and storage, wide area, etc.
  - Global filesystem often ties resources together
    - Portions of this may not run over Ethernet (e.g. IB)
    - Implications for Data Transfer Nodes
- "Science DMZ" may not look like a discrete entity here
  - By the time you get through interconnecting all the resources, you end up with most of the network in the Science DMZ
  - This is as it should be – the point is appropriate deployment of tools, configuration, policy control, etc.
- Office networks can look like an afterthought, but they aren't
  - Deployed with appropriate security controls
  - Office infrastructure need not be sized for science traffic

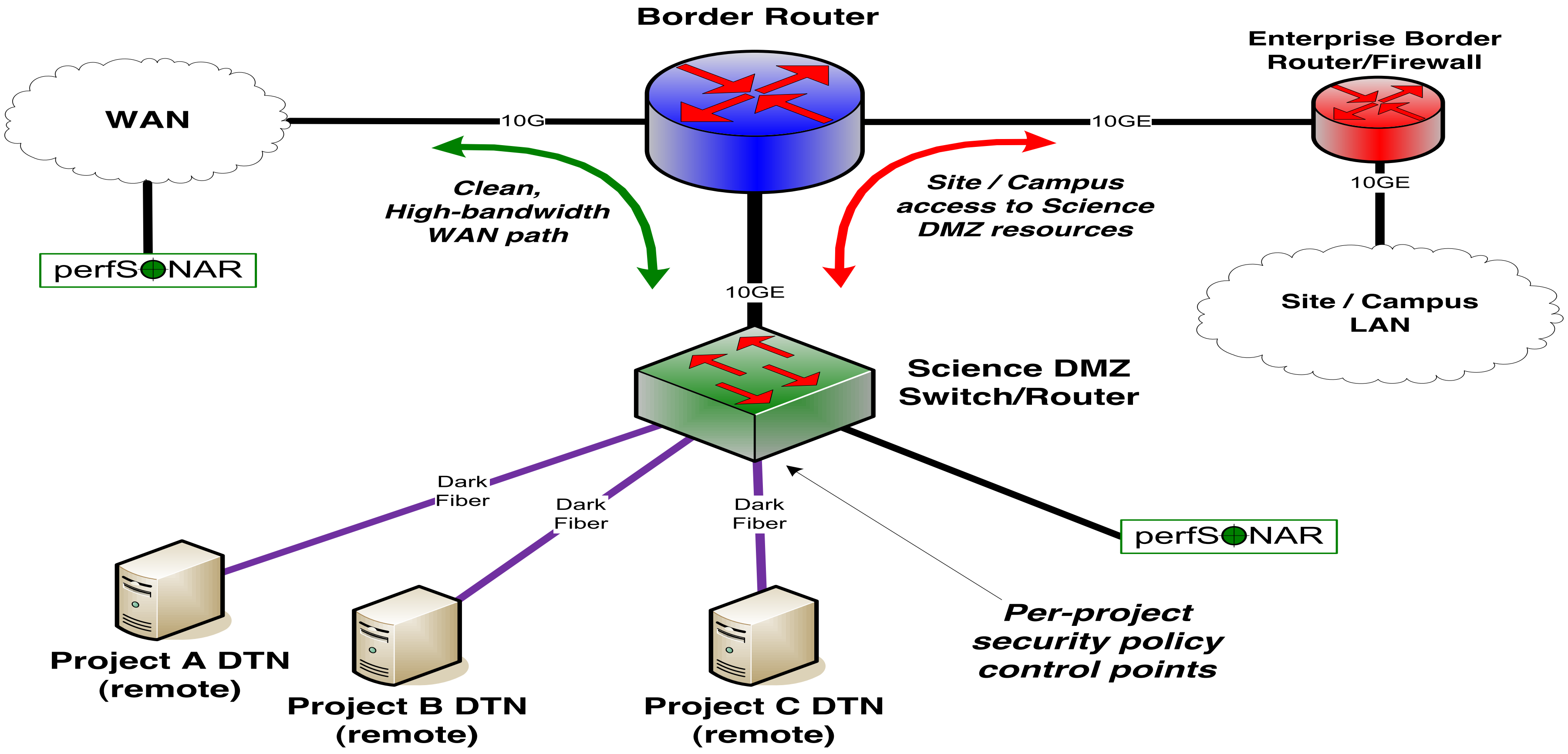
HPC Center Data Path



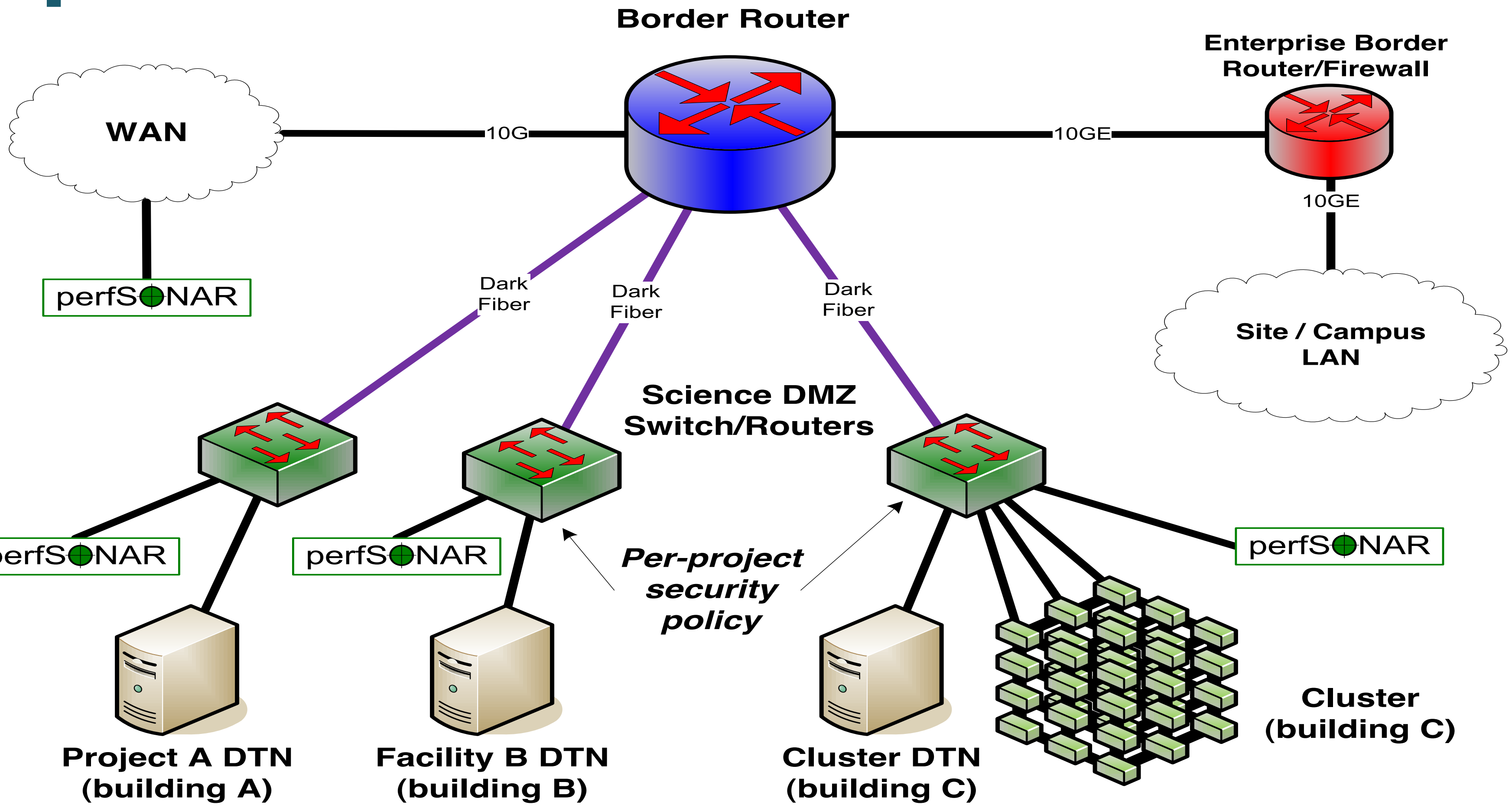
Distributed Science DMZ

- Fiber-rich environment enables distributed Science DMZ
  - No need to accommodate all equipment in one location
  - Allows the deployment of institutional science service
- WAN services arrive at the site in the normal way
- Dark fiber distributes connectivity to Science DMZ services throughout the site
  - Departments with their own networking groups can manage their own local Science DMZ infrastructure
  - Facilities or buildings can be served without building up the business network to support those flows
- Security is potentially more complex
  - Remote infrastructure must be monitored
  - Solutions depend on relationships with security groups

Distributed Science DMZ – Dark Fiber



Multiple Science DMZs – Dark Fiber

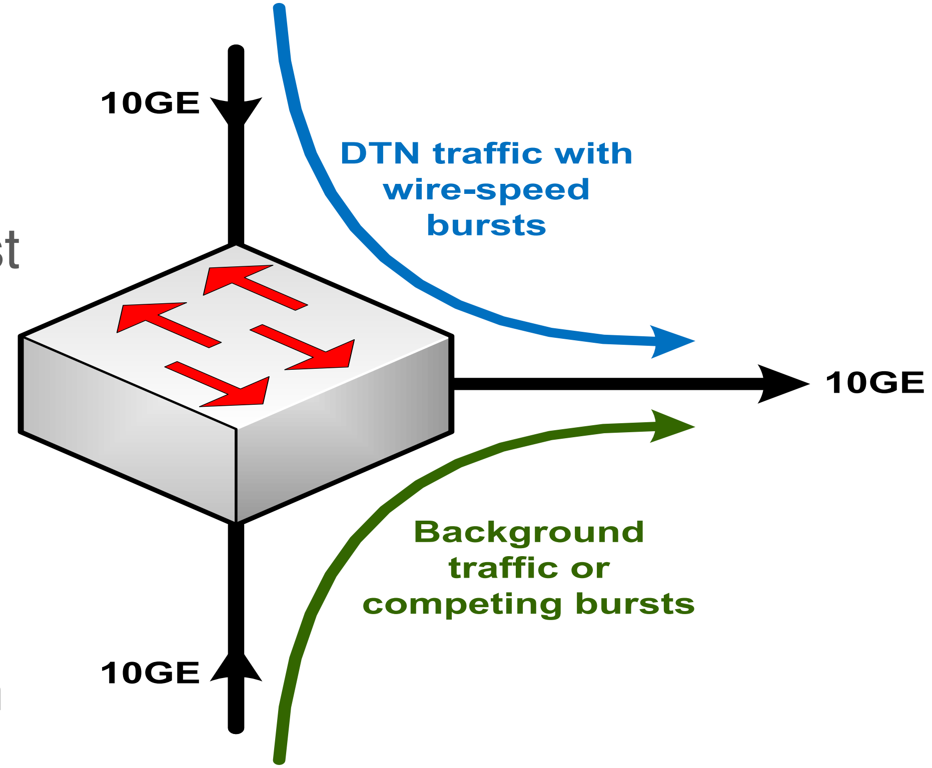


Commonality between these designs

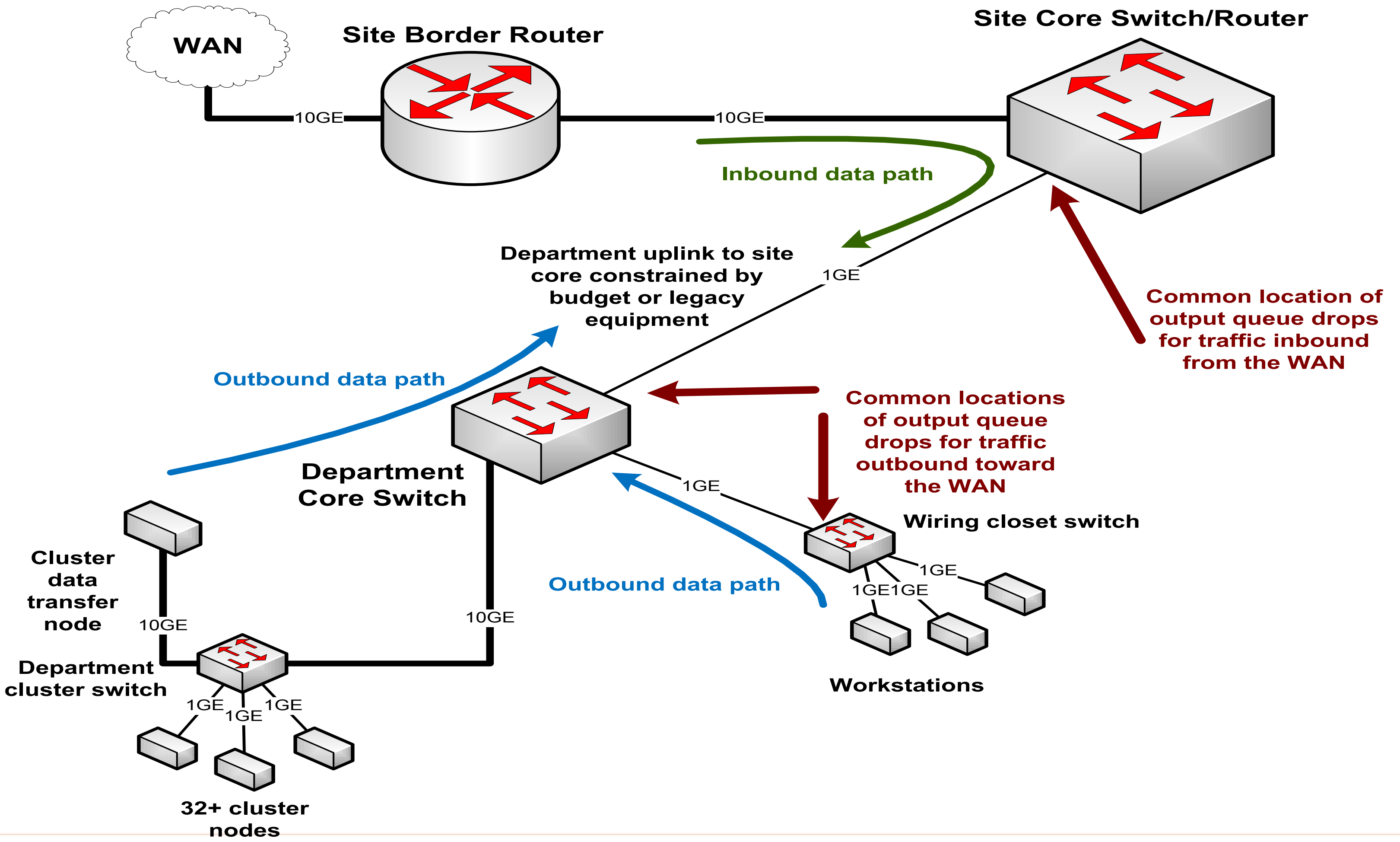
- Two common threads exist in all these examples
- Accommodation of TCP
  - Wide area portion of data transfers traverses purpose-built path
  - High performance devices that don't drop packets
- Ability to test and verify
  - When problems arise (and they always will), they can be solved if the infrastructure is built correctly
  - Small device count makes it easier to find issues
  - Multiple test and measurement hosts provide multiple views of the data path
    - perfSONAR nodes at the site and in the WAN
    - perfSONAR nodes at the remote site

Common Circumstance: Multiple Ingress Data Flows, Common Egress

- Hosts will typically send packets at the speed of their interface (1G, 10G, 100G, etc.)
  - Instantaneous rate, not average rate
  - If TCP has window available and data to send, host sends until there is either no data or no window
- Hosts moving big data (e.g. DTNs) can send large bursts of back-to-back packets
  - This is true even if the average rate as measured over seconds is slower (e.g. 4Gbps)
  - On microsecond time scales, there is often congestion
  - Router or switch must queue packets or drop them



Output Queue Drops – Common Locations



Cyberinfrastructure Support

- Lots of community resources
  - Ask folks who have already done it
  - Ask the Science DMZ mailing list: sciencedmz@es.net
- Vendors can be very helpful – just ask the right questions
  - Request an eval box (or preferably two)
  - Ask for config examples to implement a particular feature
    - E.g. "Please give me the QoS config for the following:"
      - 1 queue for network control (highest priority) – 5% of interface buffer memory
      - 1 queue configured for tail-drop (lower priority) – 95% of interface buffer memory
      - With that config, how many milliseconds of buffer are in the tail-drop queue when measured at interface wire speed?

Important concepts to be aware of

- Deep interface queues (e.g. ***buffer***)
  - Output queue or VOQ – doesn't matter
  - What TCP sees is what matters – fan-in is *not* your friend
  - No, this isn't buffer bloat
- Good counters
  - We like the ability to reliably count *every* packet associated with a particular flow, address pair, etc.
    - Very helpful for debugging packet loss
    - Must not affect performance (just count it, don't punt it)
    - sflow support if possible
  - If the box is going to drop a packet, it should increment a counter somewhere indicating that it dropped the packet
    - Magic vendor permissions and hidden commands should not be necessary
    - Some boxes just lie – run away!
- Single-flow performance should be wire-speed

Buffer Investigation

Key takeaways

- Under-buffered network devices are the ***single greatest threat*** to data intensive use of the network. You can make hosts, operating systems, and application choices perform better for ‘free’ by tuning, but it will cost $$$ to fix a junky switch or router
- You will be steered toward non-optimal choices when you talk with the vendor community because they don’t understand the fundamental math that we are explaining here
- A 1U/2U data center/rack/LAN network device should never be in the path of your data-intensive network use case.
- Non-passive (e.g., stateful) security devices are the same for buffering, and are actually worse due to the processing overhead.
- Anytime you jump around the OSI stack – you add friction (e.g., routing when you don’t need to, application layer inspection, etc.)

The all-important Buffer (No Cut Through)

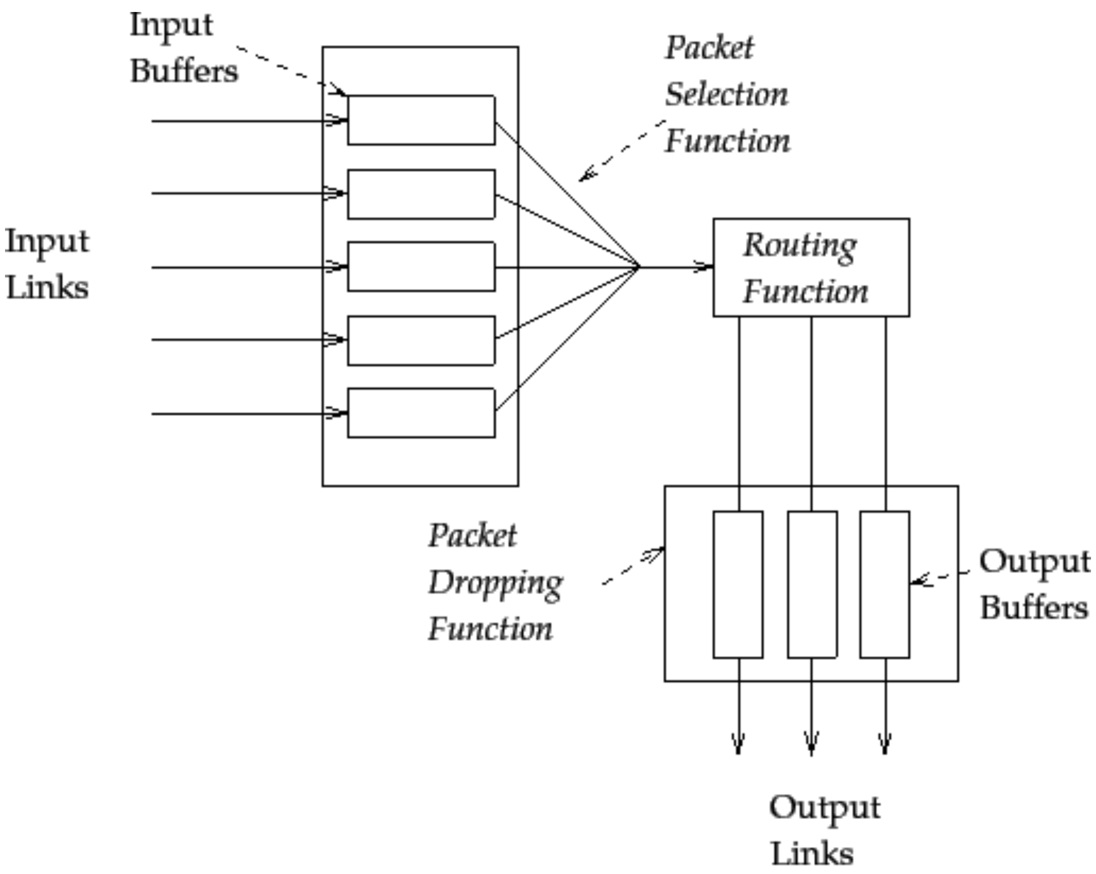


Figure 1: Basic Router Architecture

The all-important Buffer (No Cut Through)

- Data arrives from multiple sources

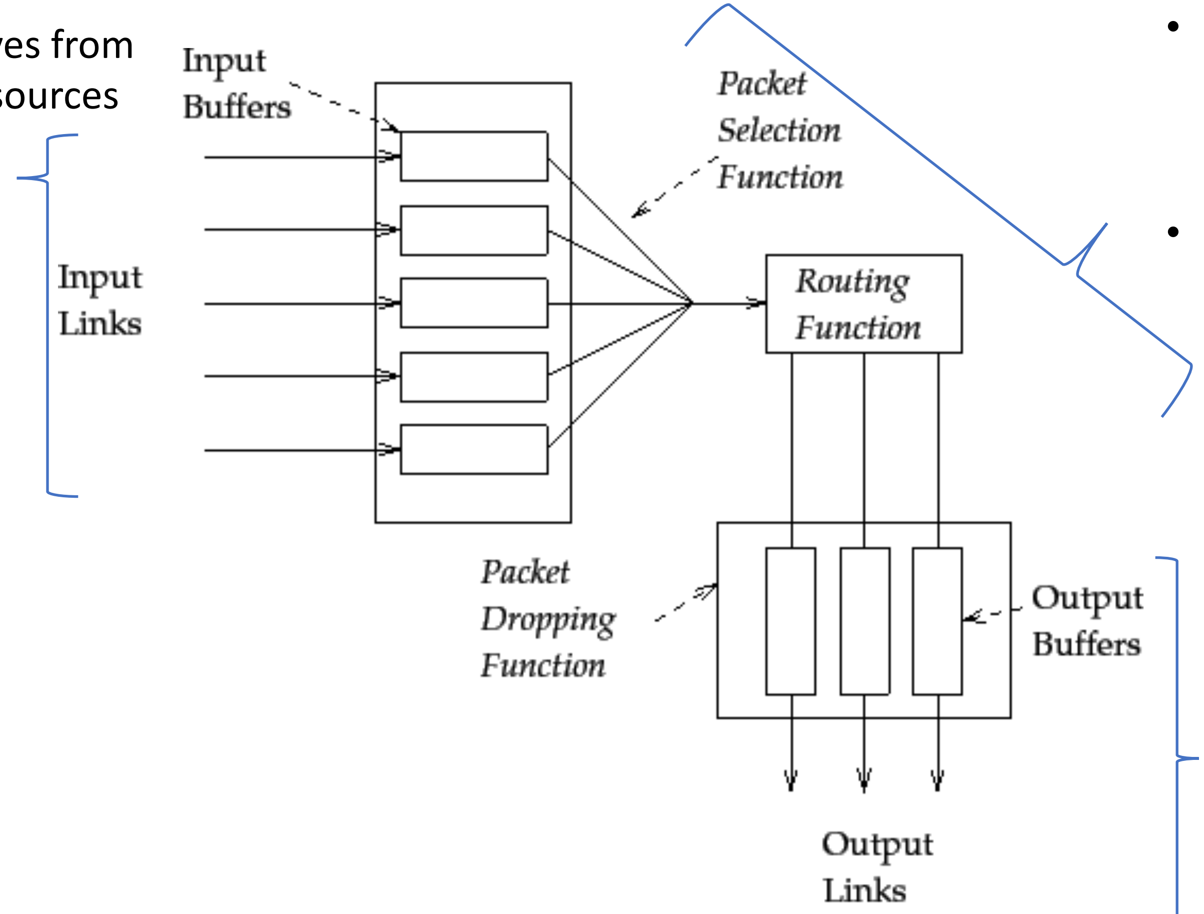
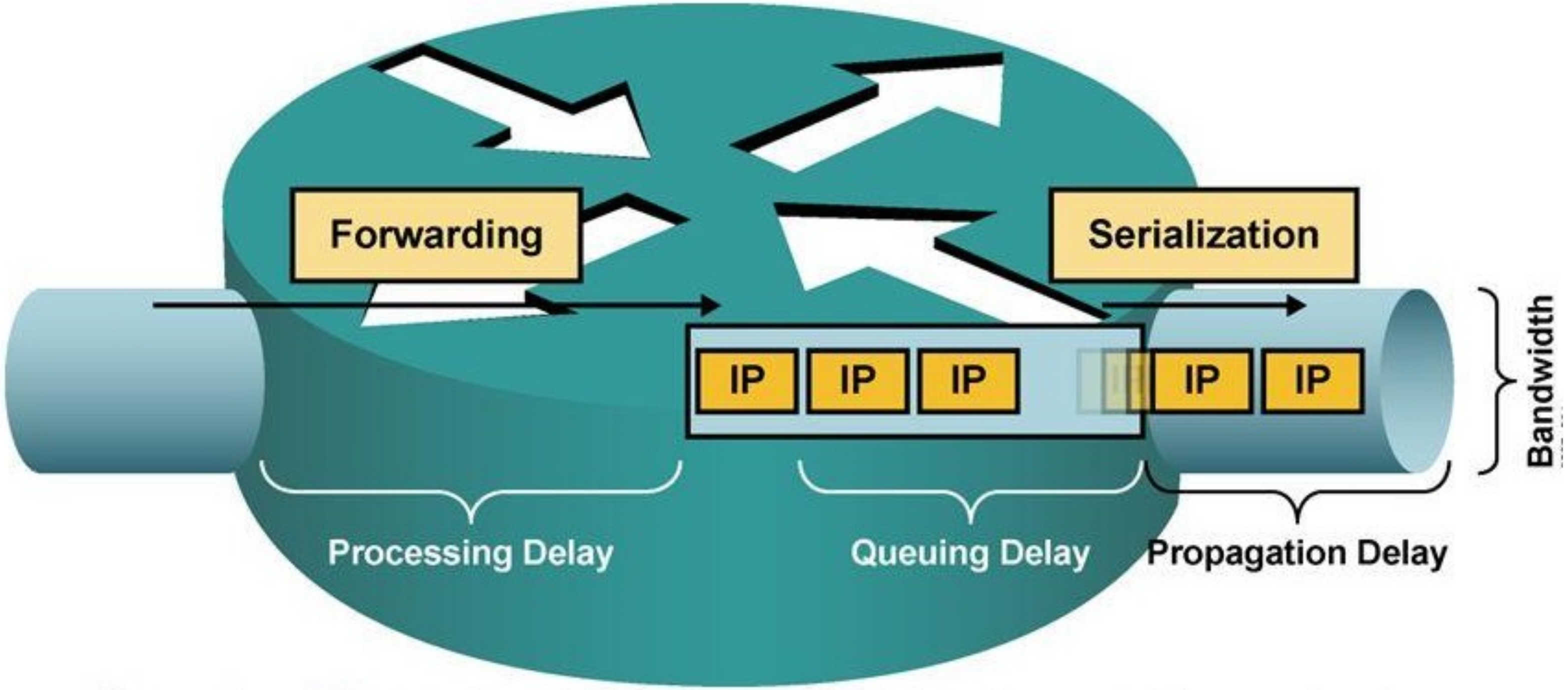


Figure 1: Basic Router Architecture

- Buffers have a finite amount of memory
  - Some have this per interface
  - Others may have access to a shared memory region with other interfaces
- The processing engine will:
  - Extract each packet/frame from the queues
  - Pull off header information to see where the destination should be
  - Move the packet/frame to the correct output queue

- Additional delay is possible as the queues physically write the packet to the transport medium (e.g., optical interface, copper interface)

The all-important Buffer (No Cut Through)



- Processing delay: The time it takes for a router to take the packet from an input interface, examine it, and put it into the output queue of the output interface.
- Queuing delay: The time a packet resides in the output queue of a router.
- Serialization delay: The time it takes to place the "bits on the wire."
- Propagation delay: The time it takes for the packet to cross the link from one end to the other.

The all-important Buffer (No Cut Through)

- ***The Bandwidth Delay Product***
  - The amount of “in flight” data for a TCP connection (BDP = bandwidth * round trip time)
- Example: 10Gb/s cross country, ~100ms
  - 10,000,000,000 b/s * .1 s = 1,000,000,000 bits
  - 1,000,000,000 / 8 = 125,000,000 bytes
  - 125,000,000 bytes / (1024*1024) ~ ***125MB***
- Ignore the math aspect: it’s making sure there is memory to catch and send packets
  - As the speed increases, there are more packets.
  - If there is not memory, we drop them, and that makes TCP sad.

The all-important Buffer (No Cut Through)

- Buffering isn't as important on the LAN (this is why you are normally pressured to buy 'cut through' devices)
  - Change the math to make the Latency 1ms = ***1.25MB***
  - 'Cut through' and low latency switches are designed for the data center, and can handle typical data center loads that don't require buffering (e.g. same to same speeds, destinations within the broadcast domain)
- Buffering *MATTERS* for WAN Transfers
  - Placing something with inadequate buffering in the path reduces the buffer for the entire path. E.g. if you have an expectation of 10Gbps over 100ms – don't place a 12MB buffer anywhere in there – your reality is now ~10x less than it was before (e.g. 10Gbps @ 10ms, or 1Gbps @ 100ms)
- Ignore the math aspect, it's really just about making sure there is memory to catch packets. As the speed increases, there are more packets. If there is not memory, we drop them, and that makes TCP sad.
  - Memory on hosts, and network gear are equally important

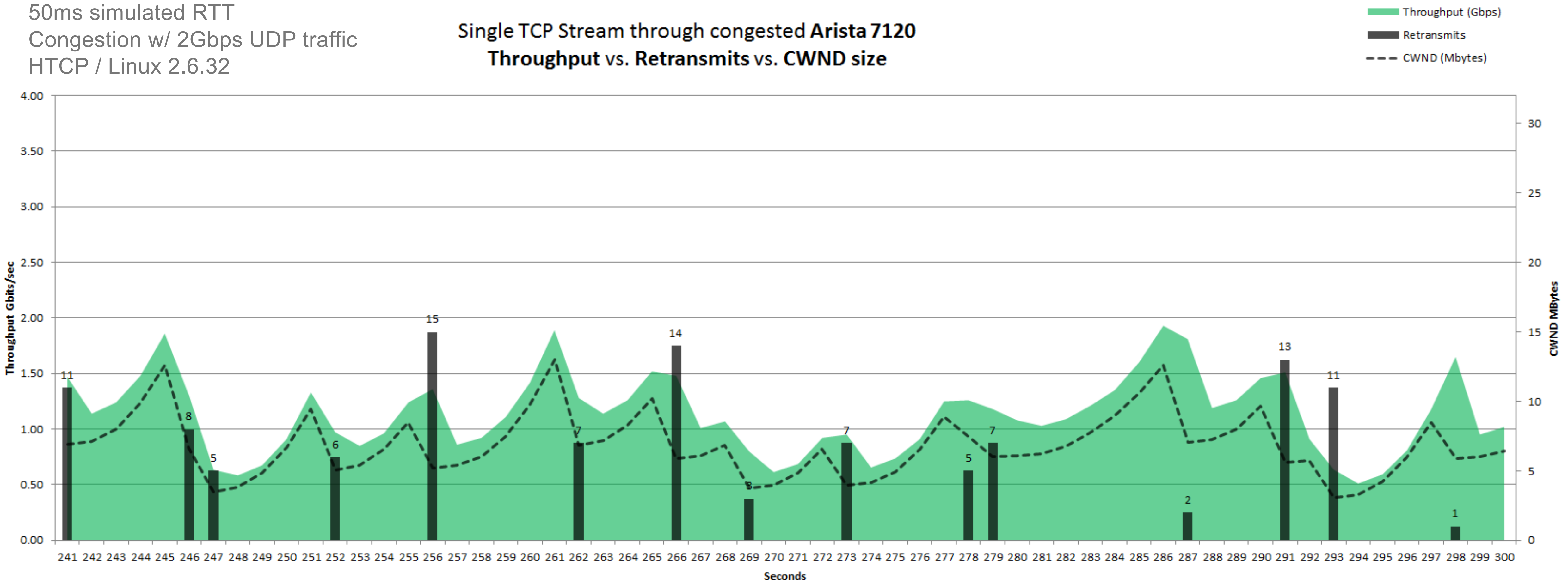
The all-important Buffer (No Cut Through)

- What does this “look” like to a data transfer? Consider the test of iperf below
  - See TCP ‘ramp up’ and slowly increase the window
  - When something in the path has no more space for packets – a drop occurs. TCP will eventually react to the lost packet, and ‘back off’
  - In the example, this first occurs when we reach a buffer of around 6-8MB. Then after backoff the window is halved a couple of times
  - This happens again later – at a slightly higher buffer limit. This could be because there was cross traffic the first time, etc.

```
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[ 14]   0.00-1.00   sec   524 KBytes  4.29 Mbits/sec    0    157 KBytes
[ 14]   1.00-2.00   sec  3.31 MBytes  27.8 Mbits/sec    0    979 KBytes
[ 14]   2.00-3.00   sec  17.7 MBytes   148 Mbits/sec    0   5.36 MBytes
[ 14]   3.00-4.00   sec  18.8 MBytes   157 Mbits/sec  214   1.77 MBytes
[ 14]   4.00-5.00   sec  11.2 MBytes  94.4 Mbits/sec    0   1.88 MBytes
[ 14]   5.00-6.00   sec  10.0 MBytes  83.9 Mbits/sec    0   2.39 MBytes
[ 14]   6.00-7.00   sec  16.2 MBytes   136 Mbits/sec    0   3.63 MBytes
[ 14]   7.00-8.00   sec  23.8 MBytes   199 Mbits/sec    0   5.50 MBytes
[ 14]   8.00-9.00   sec  38.8 MBytes   325 Mbits/sec    0   8.23 MBytes
[ 14]   9.00-10.00  sec  57.5 MBytes   482 Mbits/sec    0   11.8 MBytes
[ 14]  10.00-11.00  sec  81.2 MBytes   682 Mbits/sec    0   16.2 MBytes
[ 14]  11.00-12.00  sec  50.0 MBytes   419 Mbits/sec   35   3.93 MBytes
[ 14]  12.00-13.00  sec  15.0 MBytes   126 Mbits/sec    0   2.20 MBytes
[ 14]  13.00-14.00  sec  11.2 MBytes  94.4 Mbits/sec    0   2.53 MBytes
[ 14]  14.00-15.00  sec  13.8 MBytes   115 Mbits/sec    1   1.50 MBytes
[ 14]  15.00-16.00  sec  6.25 MBytes  52.4 Mbits/sec    5    813 KBytes
[ 14]  16.00-17.00  sec  5.00 MBytes  41.9 Mbits/sec    0    909 KBytes
[ 14]  17.00-18.00  sec  5.00 MBytes  41.9 Mbits/sec    0   1.37 MBytes
[ 14]  18.00-19.00  sec  10.0 MBytes  83.9 Mbits/sec    0   2.43 MBytes
[ 14]  19.00-20.00  sec  17.5 MBytes   147 Mbits/sec    0   4.22 MBytes
```

TCP Congestion Control

50ms simulated RTT
Congestion w/ 2Gbps UDP traffic
HTCP / Linux 2.6.32



Decoding Specifications

- "*The buffering behaviors of the switches and their operating system, such as behavior under memory stress, are typically proprietary information and not well documented*" http://www.measurementlab.net/blog/traffic-microbursts-and-their-effect-on-internet-measurement/

- "Even if you know ***how much*** packet buffer is in the switch, assumptions on ***how it is deployed*** that are not backed up by testing can lead to unhappiness. What we like to say is that is ***the job of the network engineers to move bottlenecks around***."
  - *Jim Warner* *http://people.ucsc.edu/~warner/buffer.html*

Decoding Specifications

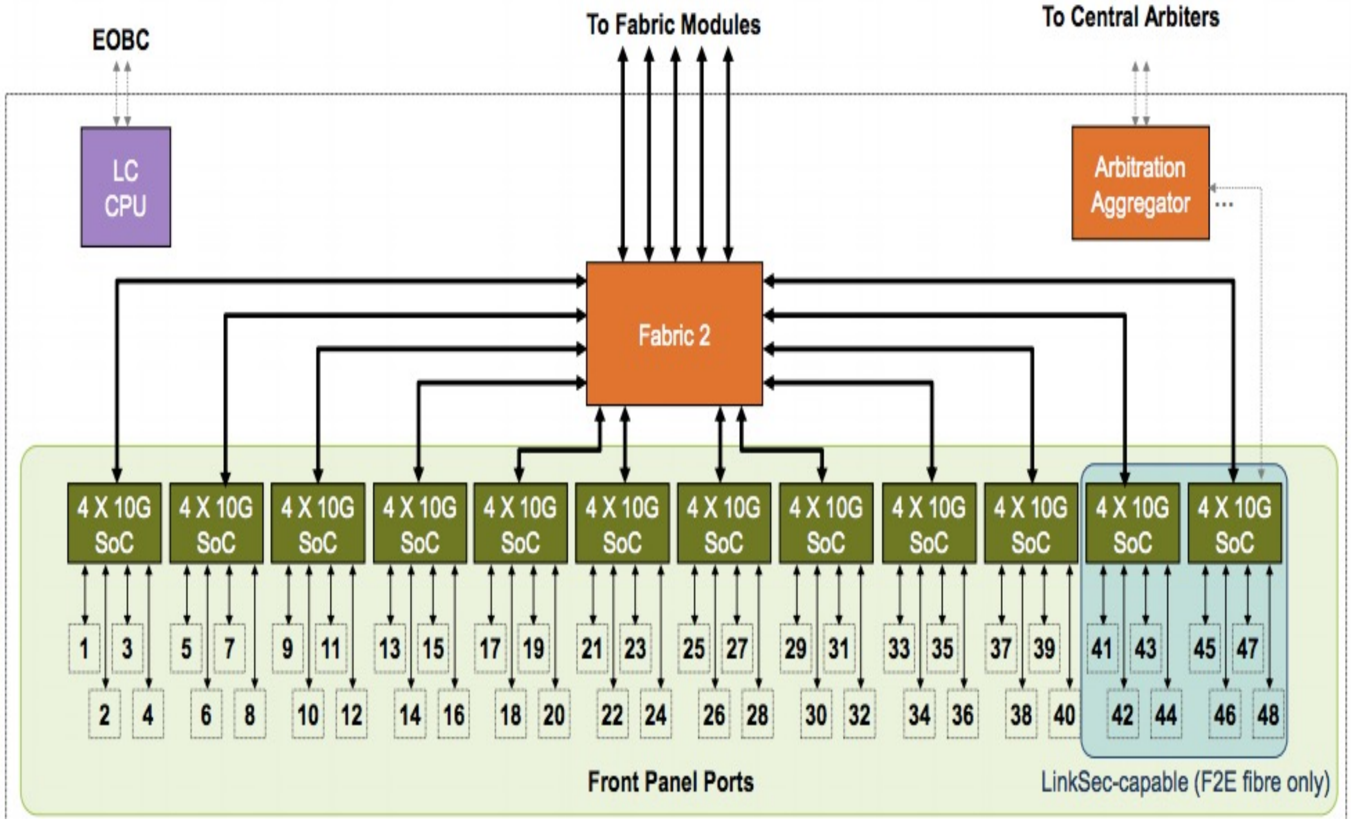
- So lets say the spec sheet says this:

VOQ buffer	72 MB per module

- What does ‘module’ mean?
  - Typically this means the amount of memory for the entire switch (if it’s a single unit) or a blade (if the chassis supports more than one.
  - BUT … this memory can be allocated in a number of different ways:
    - ***Shared between all ports***
    - ***Dedicated (smaller) amounts per-port***
    - ***Shared between ASICS, which control a bank of ports***

Decoding Specifications

- Consider this architecture
  - 48 Ports
    - 12 ASICS
    - 4 Ports per ASIC
  - ***72MB*** total
    - ***6MB per ASIC***
    - If all ports are in use – expect that each port has access to ***1.5MB***. If only one is in use, it can use 6MB
  - Additional memory is often available in a 'burst buffer' in the fabric



Decoding Specifications

- Recall: https://www.switch.ch/network/tools/tcp_throughput/
- What does 6MB get you?
  - 1Gbps @ <= 48ms (e.g., ½ needed for coast-to-coast)
  - 10Gbps @ <= 4.8ms (e.g., metro area)
- What does 1.5MB get you?
  - 1Gbps @ <= 12ms (e.g., regional area)
  - 10Gbps @ <= 1.2ms (e.g., data center [or more accurately, rack or row)]
- In either case – remember this assumes you are the only thing using that memory … congestion is a more likely reality

Buffer Takeaways

- Try before you buy
  - Request a demo unit (or two)
  - Learn all the ins and outs
- Develop tests for worst case scenario
  - Plug in all the ports, and create traffic with a hardware tester (IXIA, SPIRENT) or a perfSONAR resource
  - Cross traffic within the switch
  - Testing to far away resources (latency is your friend and enemy)
- If you can’t get single stream TCP to work well, buffers are often the core of the problem
- It’s truly worth spending the extra $ on buffer!!

Summary

- There is no single “correct” way build a Science DMZ
  - These are design patterns, not rules
- It depends on things like:
  - site requirements and policies
  - existing resources
  - availability of dark fiber
  - budget
- The main point is to reduce the opportunities for packet loss, and be able to find loss if it’s present

Resources

https://fasterdata.es.net/science-dmz/ - ESnet resource page on Science DMZ

https://linuxclustersinstitute.org/ - LCI offers advanced technical training for those interested in deploying high-performance computing clusters through its workshops.

https://fasterdata.es.net - An Expert Guide for End-to-End Performance Tuning, Tools and Techniques

Questions?