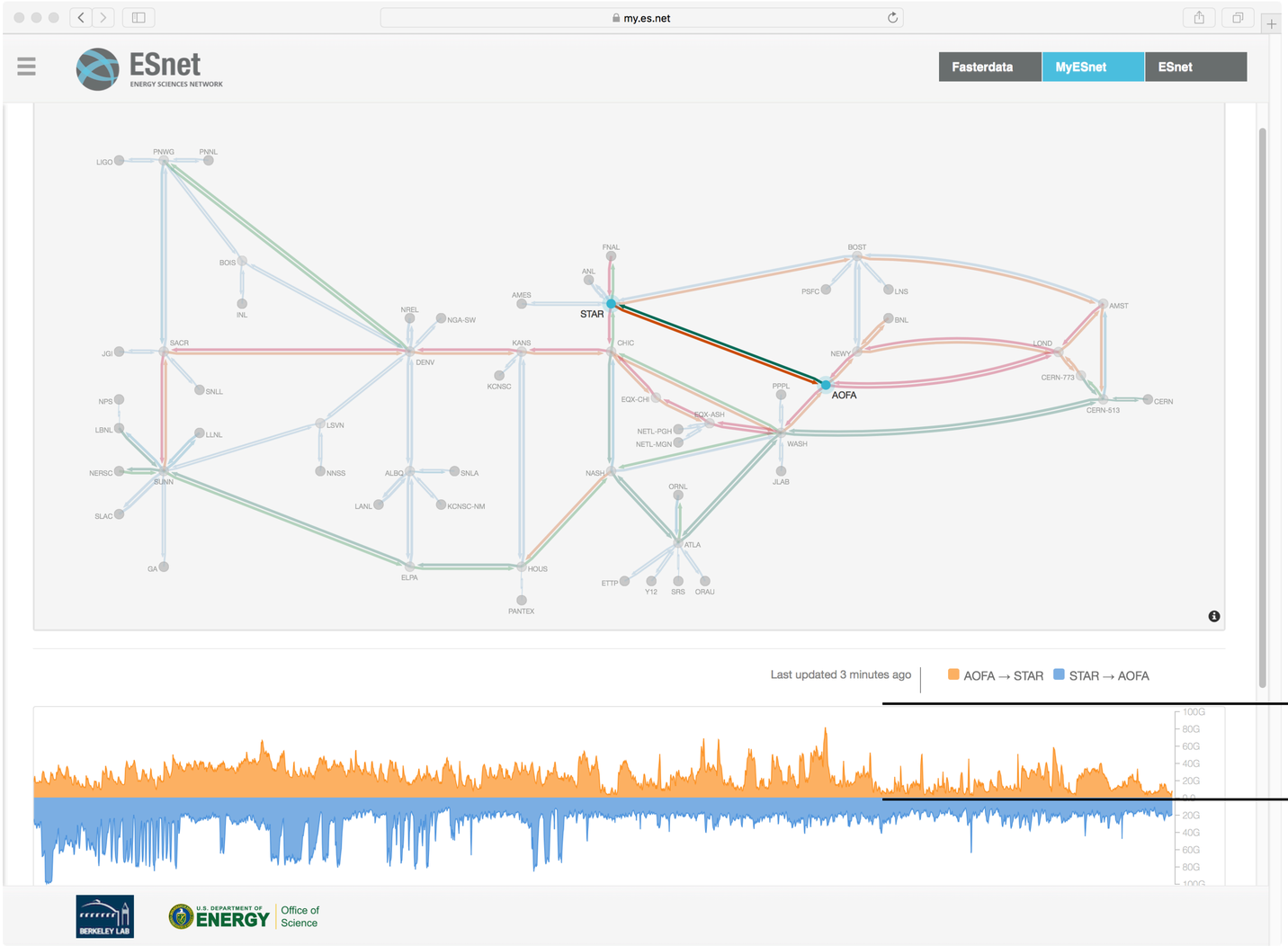
Questioning TCP

Yatish Kumar

ESnet

Why is our link utilization low ?

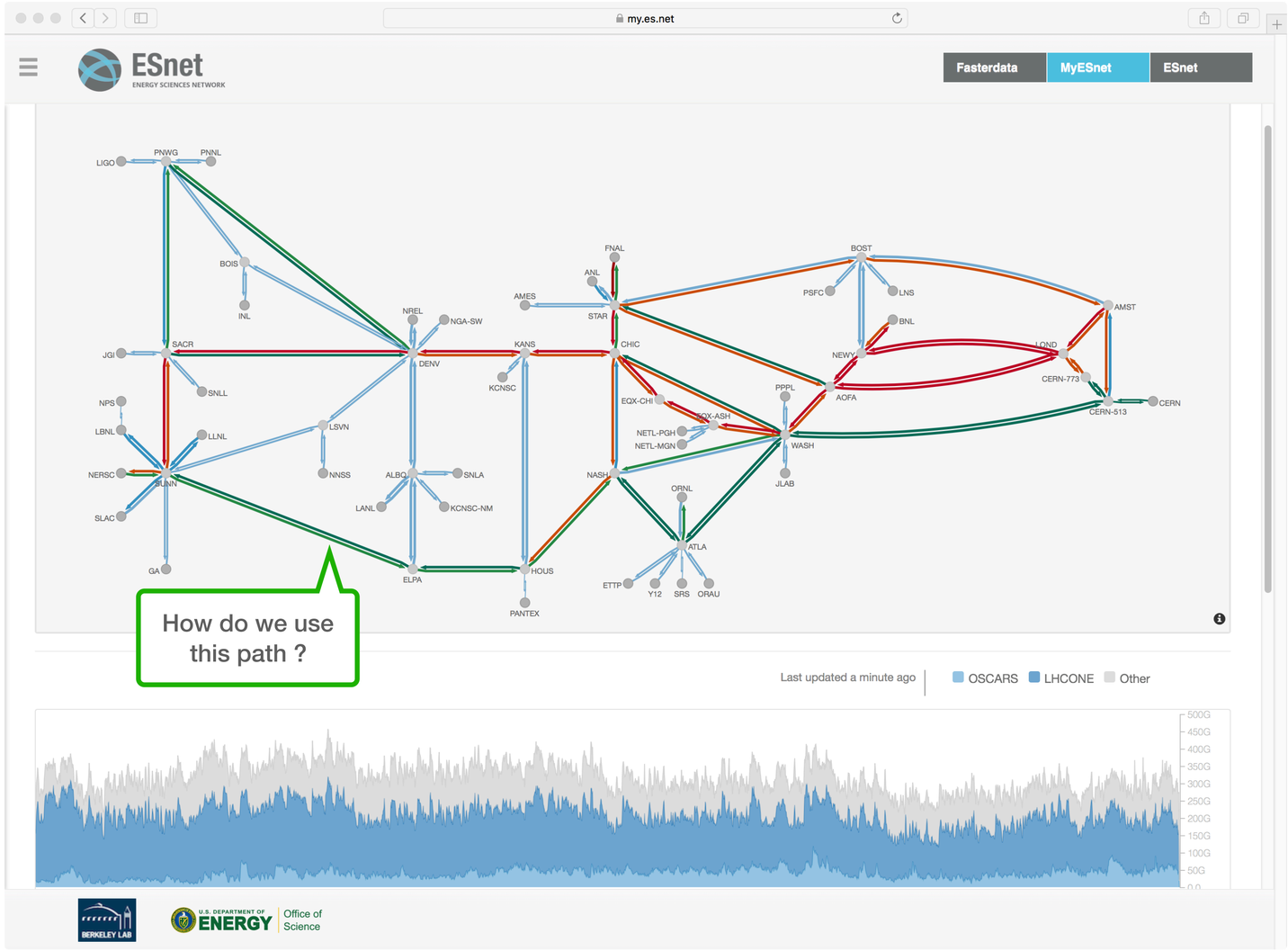


If large science data flows persist for many hours, why don't they operate 2x faster by filling our links up to 100% ?

One of our busy links, yet it is operating at far less than 100G

ESnet

How do we use our whole network ?



Without resorting to manual intervention or building custom tunnels how do we get IP to use every path, not just the Dijkstra shortest path ?

ESnet

Questioning TCP

- Reliable delivery
  - Unburden applications from having to correct for network packet loss
- Network congestion protection
  - Protect the network when lots of people want to use it
- IP based resiliency
  - Reroute at L3
- Zero explicit signalling
  - Use packet loss as a clue to avoid network congestion

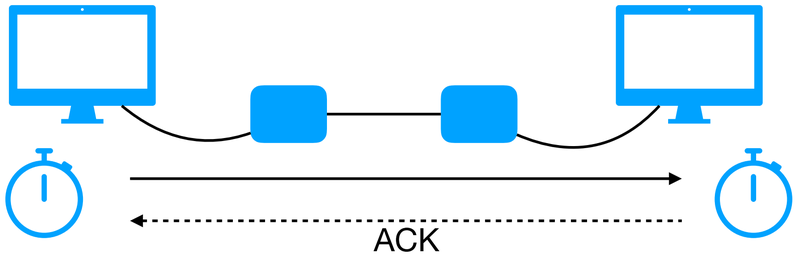
TCP connects these four needs but they are independent of each other.

ESnet

Are we stuck in the 80's ?

- Reliable delivery
  - Operating systems were the cool new thing. VAX, Unix, DOS, CPM, IBM-46xx etc..
  - Having the OS take care of reliable delivery was awesome. But why L4 ?
    - We could make the session layer (L5) or the presentation layer (L6) reliable.

- Reliable delivery creates the following problems:
  - In order packet delivery, or massive reassembly buffers
  - Some form of loss detection
    - TCP maintained a timeout for cases of packet loss, but had a much better host to host mechanism based on sequence numbers and ACKs.
    - Routers "elegantly" exploited the timeout mechanism to signal congestion by silently discarding packets. Either the sender or the receiver would wait for a missing packet, and would eventually declare a retransmission.
    - This lead to the need for in order packet delivery so that we could quickly declare a lost packet.
    - And …. In order flows are the basis for the dreaded elephant flow



ESnet

A loss free TCP network is an oxymoron

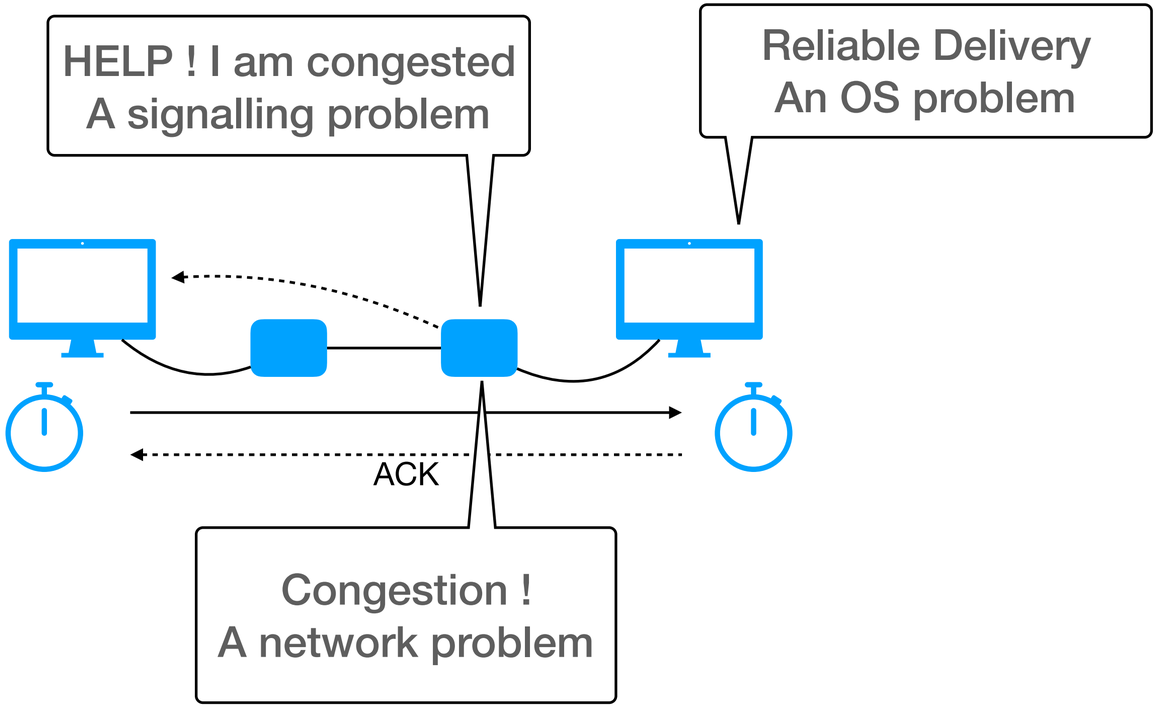
If packet loss is how the network signals congestion, then we will most certainly have packet loss.

Unless TCP underperforms and is unable to fill the network due to window size limits

1. Detecting buffer fill via. Latency avoids the poor practice of packet discards as a signaling mechanism.

2. If the network stops discarding packets in order to “send a message” reliable delivery can move up to L5 or L6 with retransmissions based on real packet loss. BER / Less frequent congestion

3. Out of order delivery is less time critical, and can be sorted out over an entire dataset, not a transmission window.
    1. Leading to the extinction of elephants
    2. Allowing mice to flood every link on our network



ESnet

Evolving past steampunk protocols

1. Do reliable delivery over a dataset
2. Deal with network congestion without discarding
3. Exploit UDP over IP to fill the network with mice

L5 or L6
UDP
IP

To achieve this we need simultaneous
- Change in the OS / Network protocol stack
- Change in routers for congestion protection
- Change in applications to adapt to the new OS presentation layer

For every device on the internet. Simultaneously.

ESnet

Ok. Ok.. Domain Specific Evolution

Just deal with data intensive R&E networking

Leverage GridFTP / XrootD and Science DMZ as a place to start.
- We only monkey with OS layer changes on a few DTNs

Introduce some form of DSCP or ECN like markings using SDN in our network
- or embedded In-band Telemetry / OWAMP measurements to detect congestion
- failing that, we do nothing and rely on the DTNs to detect latency changes (BBR style)

Leverage segment routing / PCE to move our UDP packets across multiple paths
- or simply provision multiple L2 circuits between sites and hand them to the OS
- failing that, we do nothing and let the joy of having no elephant flows sink in for a while

Which boils down to us needing :
A high energy physicist
A few DTNs
A networking protocol megalomanic



ESnet

Backup Slides

TCP loss vs. rate chart

ESnet

L7 solution to the routing problem

ESnet