SDN WAN Applications

Sangeetha Abdu Jyothi

CS 538 February 26, 2018

Initial "Killer apps"

Cloud virtualization

- Create separate virtual networks for tenants
- Allow flexible placement and movement of VMs

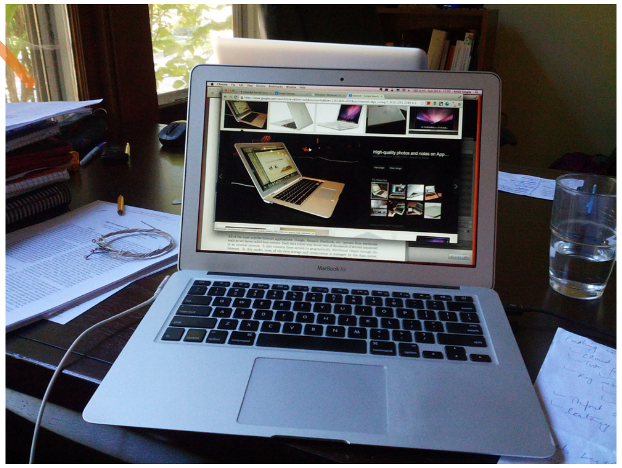
WAN traffic engineering

- Drive utilization to near 100% when possible
- Protect critical traffic from congestion

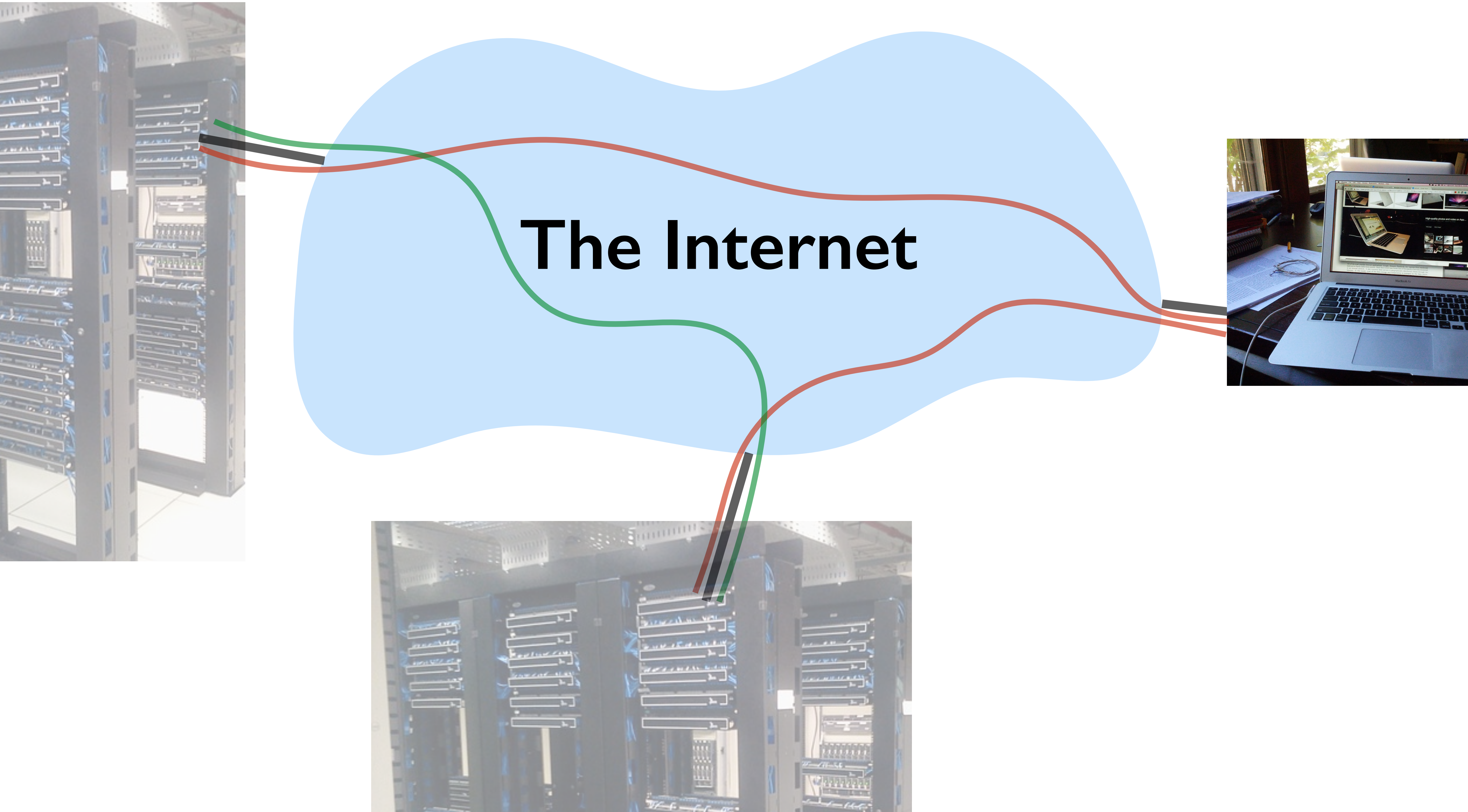
Key characteristics of the above

- Special-purpose deployments with less diverse hardware
- Existing solutions aren't just annoying, they don't work!

How large online services work



How large online services work



Why multiple data centers?

Data availability

Load balancing

Latency

Local data laws

Hybrid public-private operation

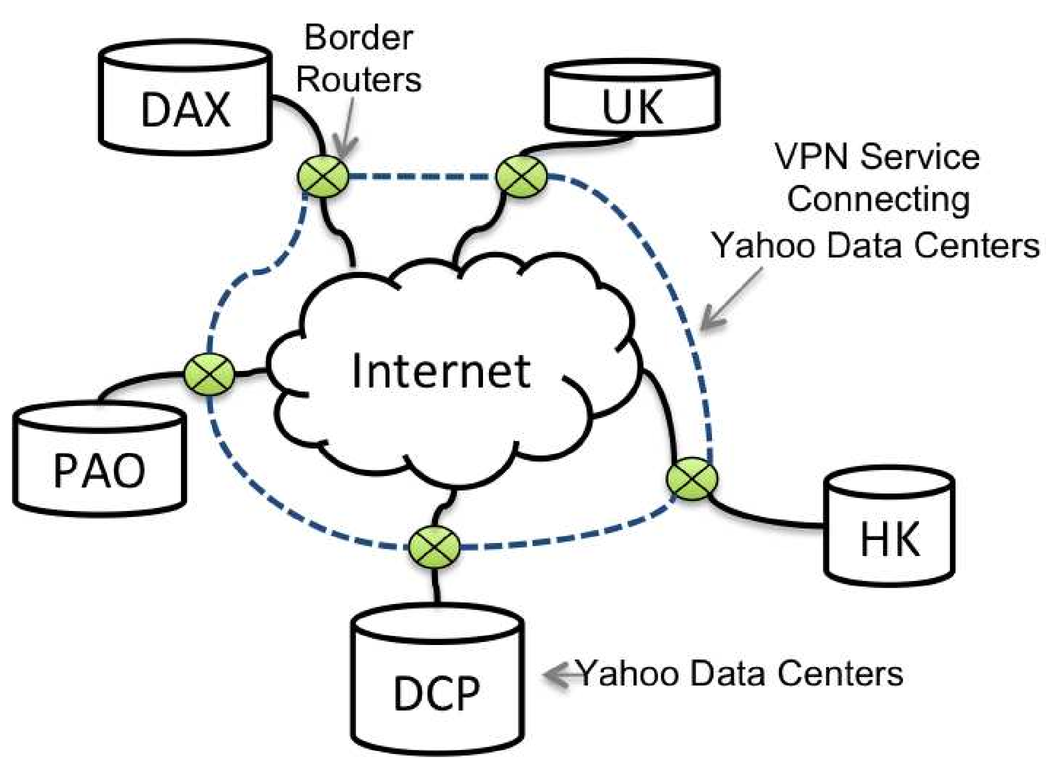
Inter-data center traffic is significant

IEEE INFOCOM, 2011

A First Look at Inter-Data Center Traffic Characteristics via Yahoo! Datasets

Yingying Chen[1], Sourabh Jain[1], Vijay Kumar Adhikari[1], Zhi-Li Zhang[1], and Kuai Xu[2]

[1]University of Minnesota-Twin Cities
[2]Arizona State University



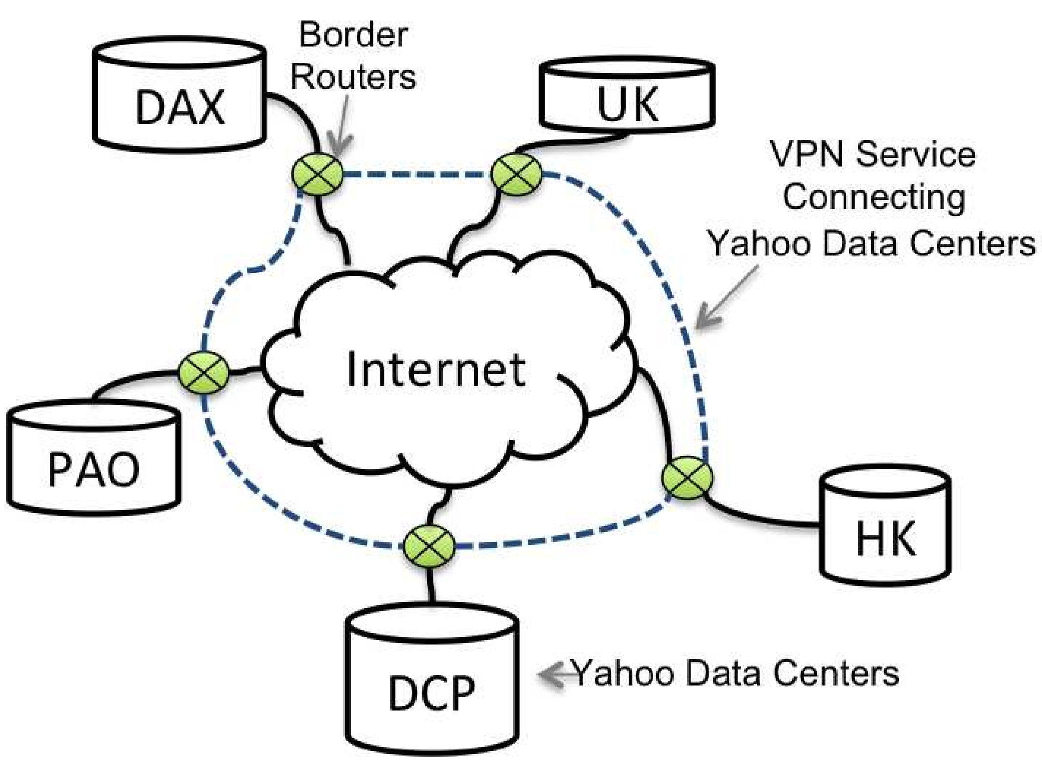
Inter-data center traffic is significant

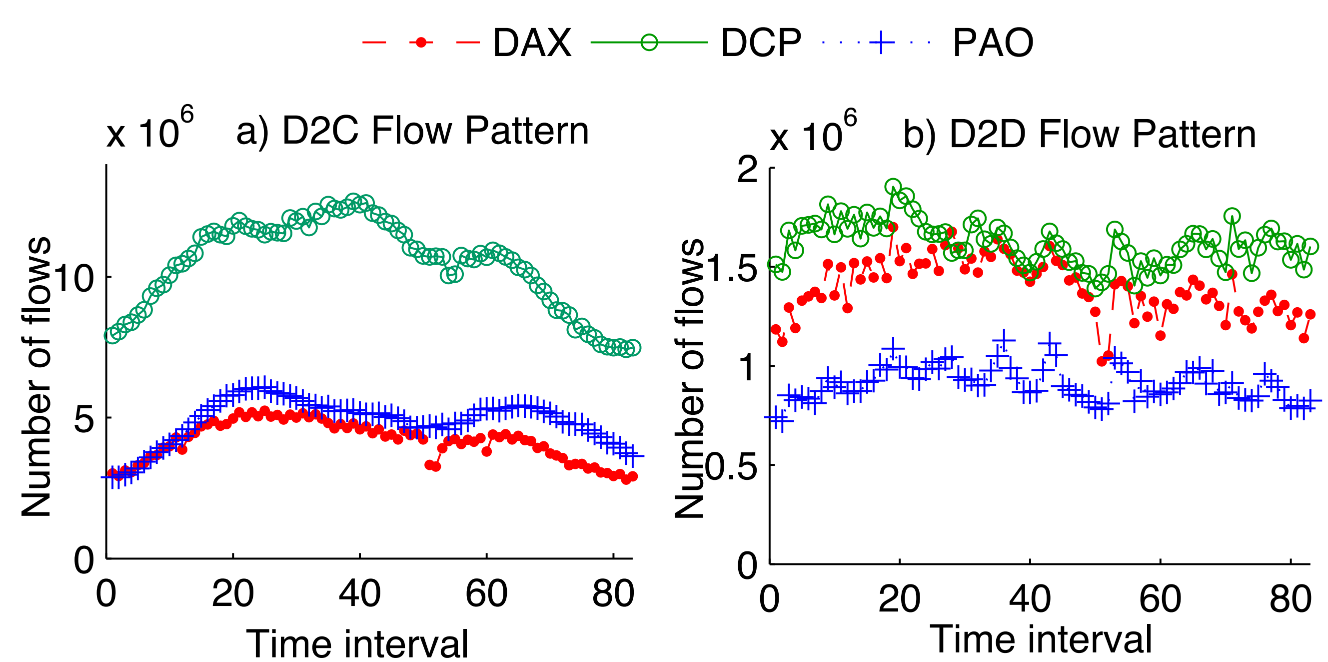
IEEE INFOCOM, 2011

A First Look at Inter-Data Center Traffic Characteristics via Yahoo! Datasets

Yingying Chen[1], Sourabh Jain[1], Vijay Kumar Adhikari[1], Zhi-Li Zhang[1], and Kuai Xu[2]

[1]University of Minnesota-Twin Cities
[2]Arizona State University





Inter-data center traffic is significant


Back-Office Web Traffic on The Internet

Enric Pujol
TU Berlin
enric@inet.tu-berlin.de

Philipp Richter
TU Berlin
prichter@inet.tu-berlin.de

Balakrishnan Chandrasekaran
Duke University
balac@cs.duke.edu

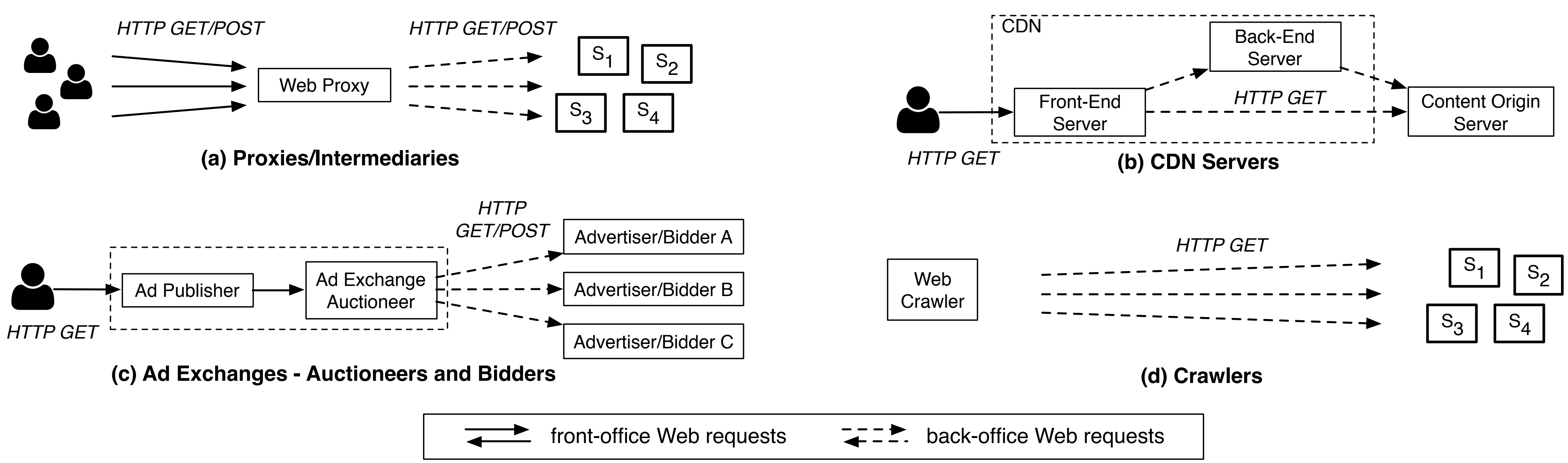
Georgios Smaragdakis
MIT / TU Berlin / Akamai
gsmaragd@csail.mit.edu

Anja Feldmann
TU Berlin
anja@inet.tu-berlin.de

Bruce Maggs
Duke / Akamai
bmm@cs.duke.edu

Keung-Chi Ng
Akamai
kng@akamai.com

[IMC 2014]



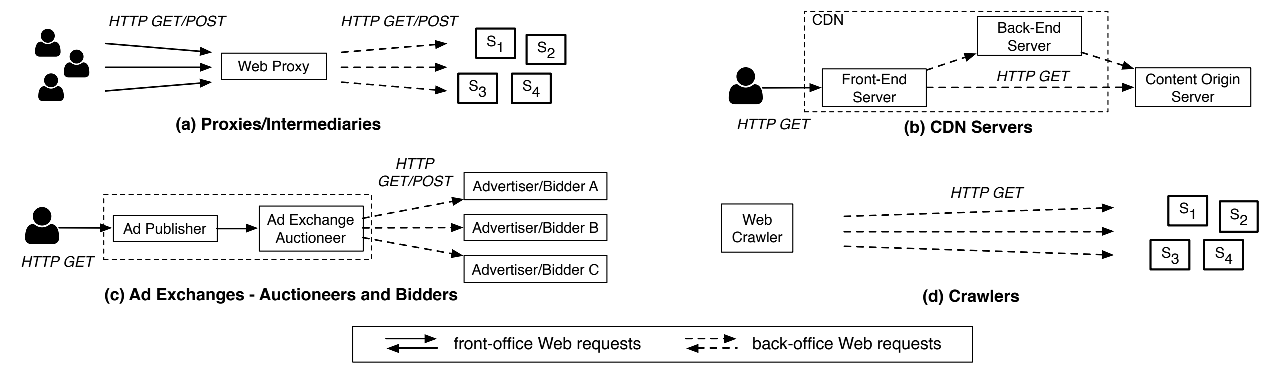
"Back office" web traffic:
server-to-server rather than
directly communicating with user

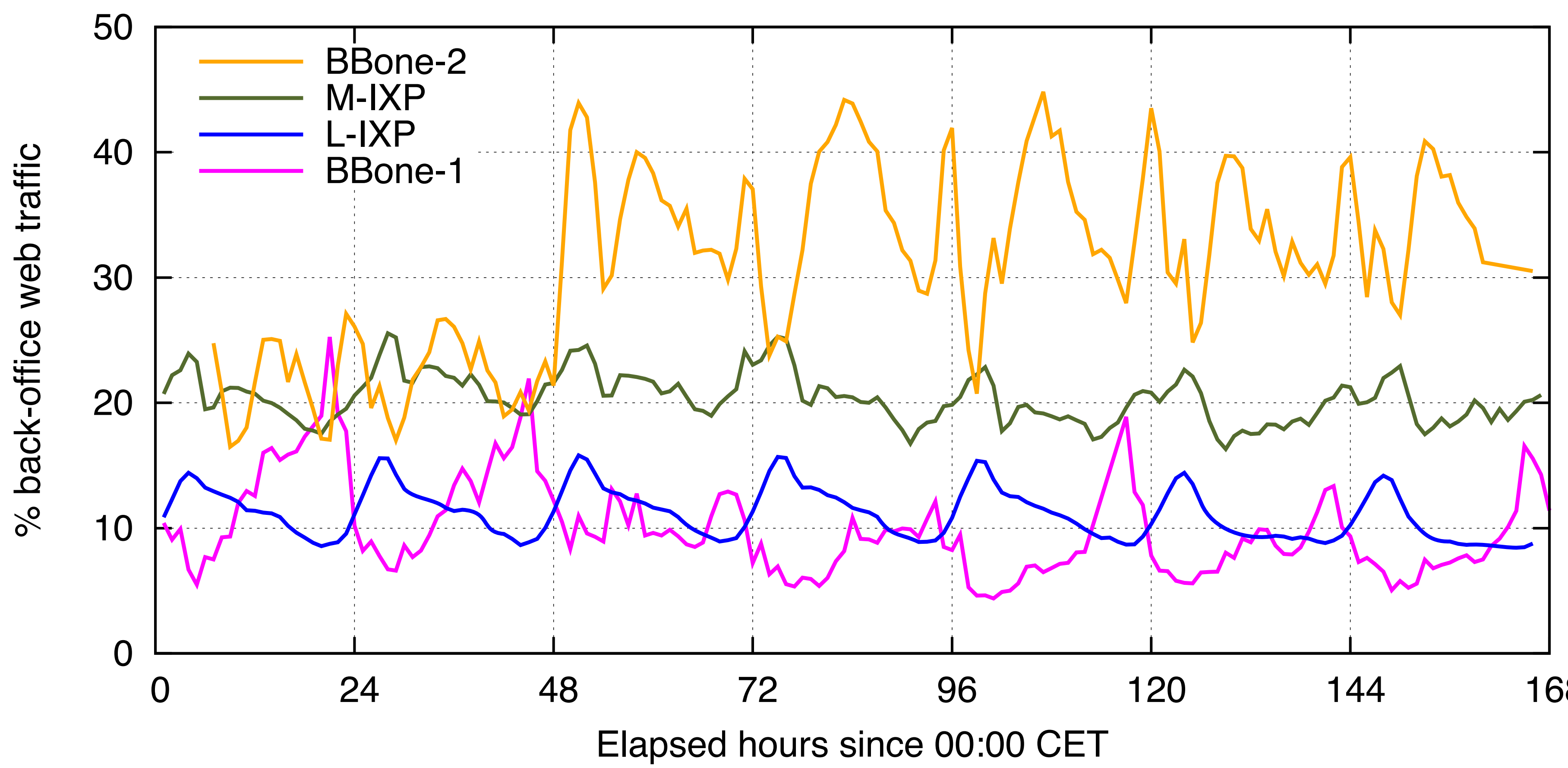
Inter-data center traffic is significant

Back-Office Web Traffic on The Internet

Enric Pujol, TU Berlin, enric@inet.tu-berlin.de
Philipp Richter, TU Berlin, prichter@inet.tu-berlin.de
Balakrishnan Chandrasekaran, Duke University, balac@cs.duke.edu
Georgios Smaragdakis, MIT / TU Berlin / Akamai, gsmaragd@csail.mit.edu
Anja Feldmann, TU Berlin, anja@inet.tu-berlin.de
Bruce Maggs, Duke / Akamai, bmm@cs.duke.edu
Keung-Chi Ng, Akamai, kng@akamai.com

[IMC 2014]





% of web traffic that is “back office” in 4 ISP, IXP data sets

Google's WAN (2011)

"B4: Experience with a Globally-Deployed Software Defined WAN"
Jain et al., ACM SIGCOMM 2013

"B4 has been in deployment for three years, now carries more traffic than Google's public facing WAN, and has a higher growth rate."

Recall: Classic TE solutions

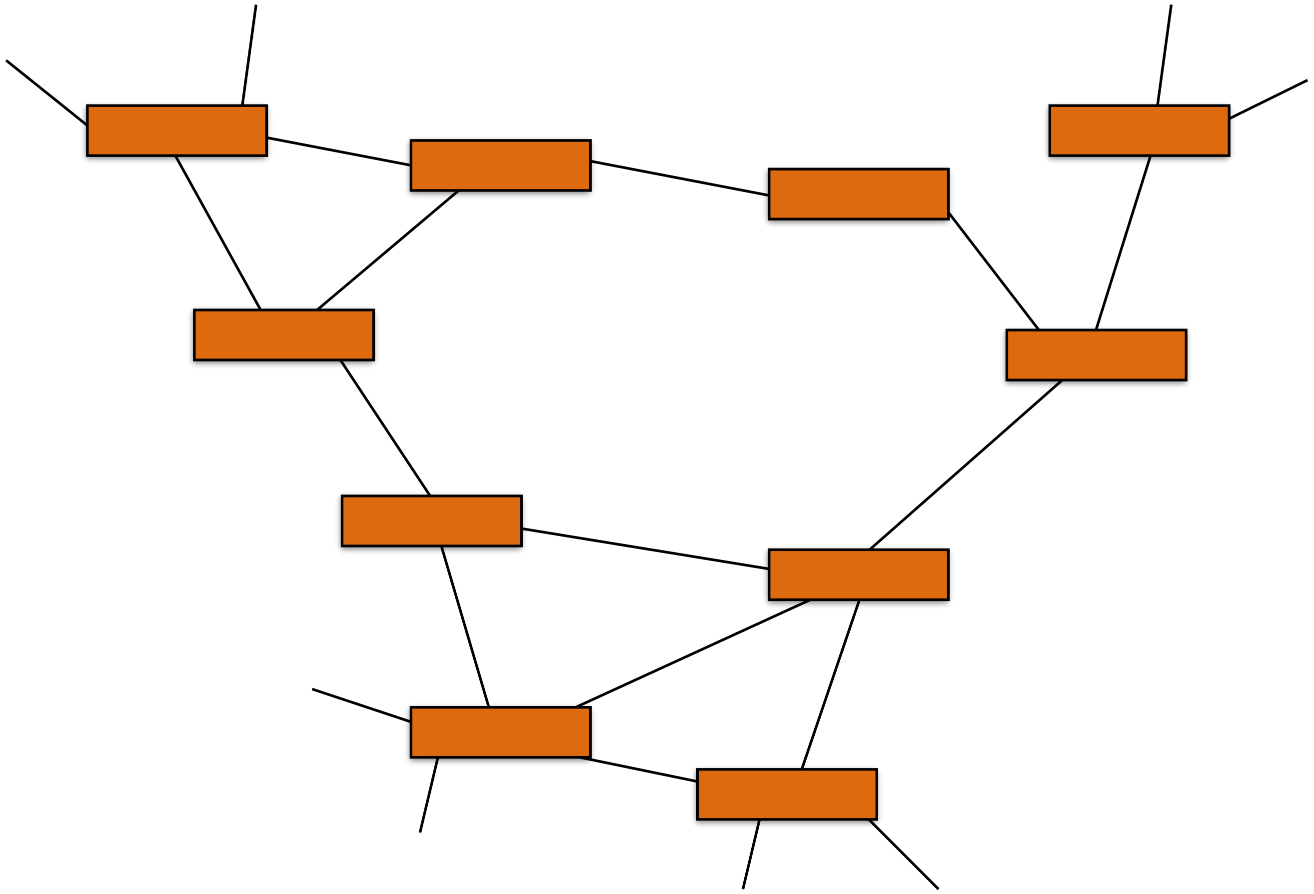
Approach 1: Optimize OSPF weights

- e.g. OSPF-TE
- Need to propagate everywhere: can't change often
- Artificial constraints make it difficult to optimize
  - Same weights apply to all traffic
  - So all traffic at one ingress follows same paths

Approach 2: Allocate traffic to explicit MPLS paths

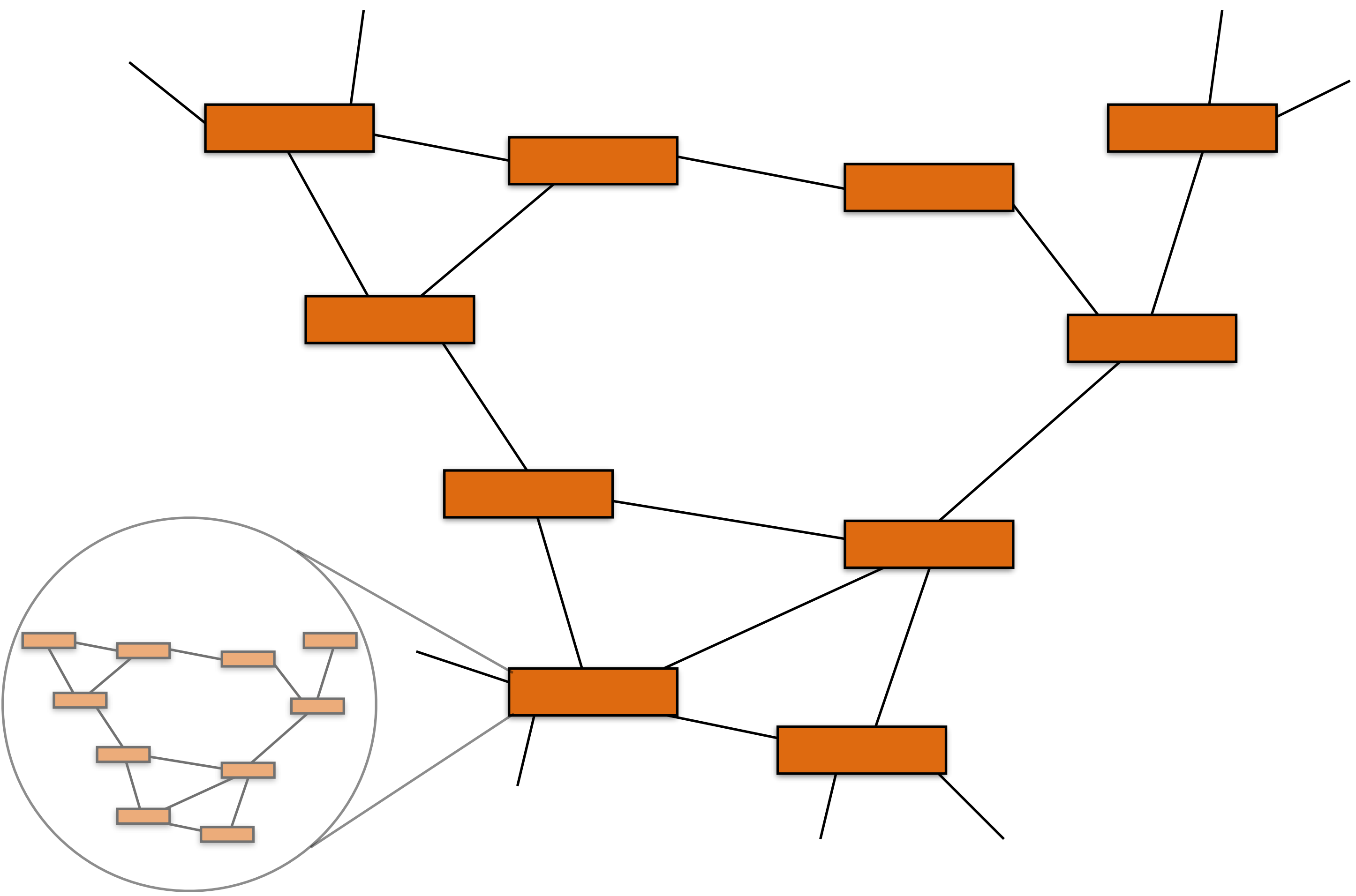
- Control protocol like RSVP-TE reserves capacity and constructs MPLS tunnels at each router along path
- Tradeoff: improves path choice but also state in routers
  - Not all possible paths will be available

Recall: MPLS with RSVP-TE



[Slide Credit: Ankit Singla]

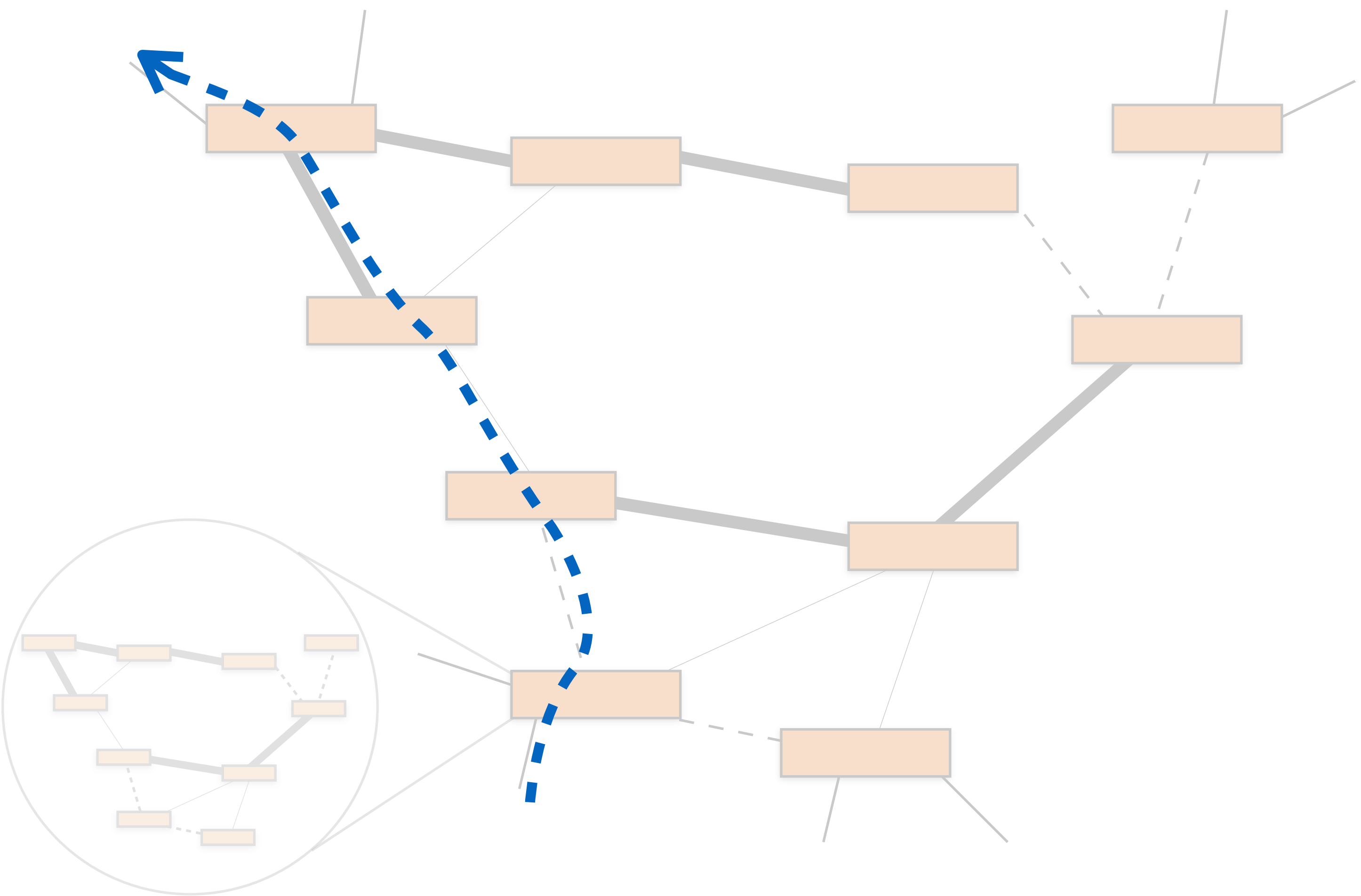
Recall: MPLS with RSVP-TE



1 Link-state protocol (OSPF / IS-IS)

[Slide Credit: Ankit Singla]

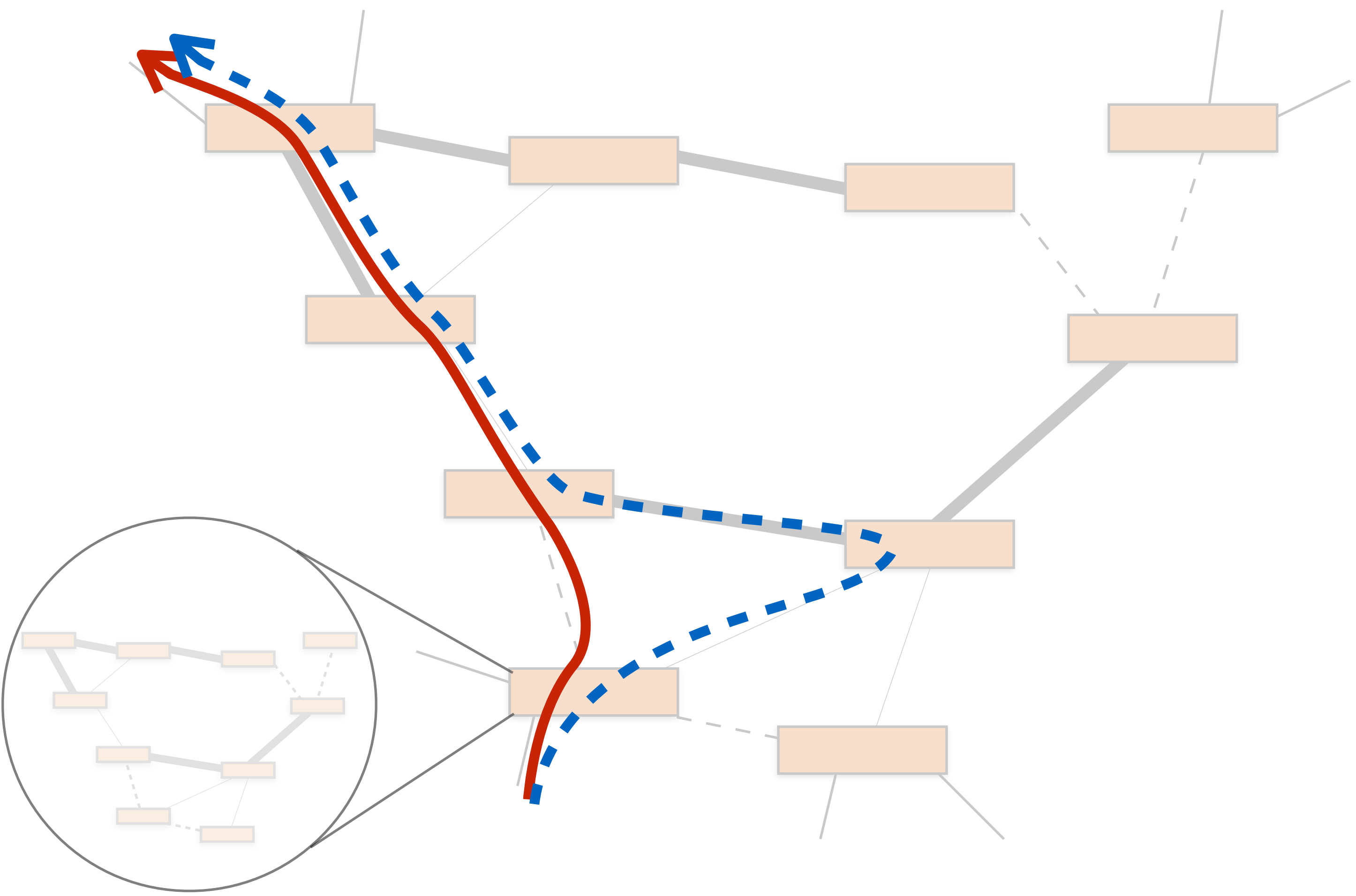
Recall: MPLS with RSVP-TE



1. Link-state protocol (OSPF / IS-IS)
2. Also flood available bandwidth info
3. Fulfill tunnel provisioning requests

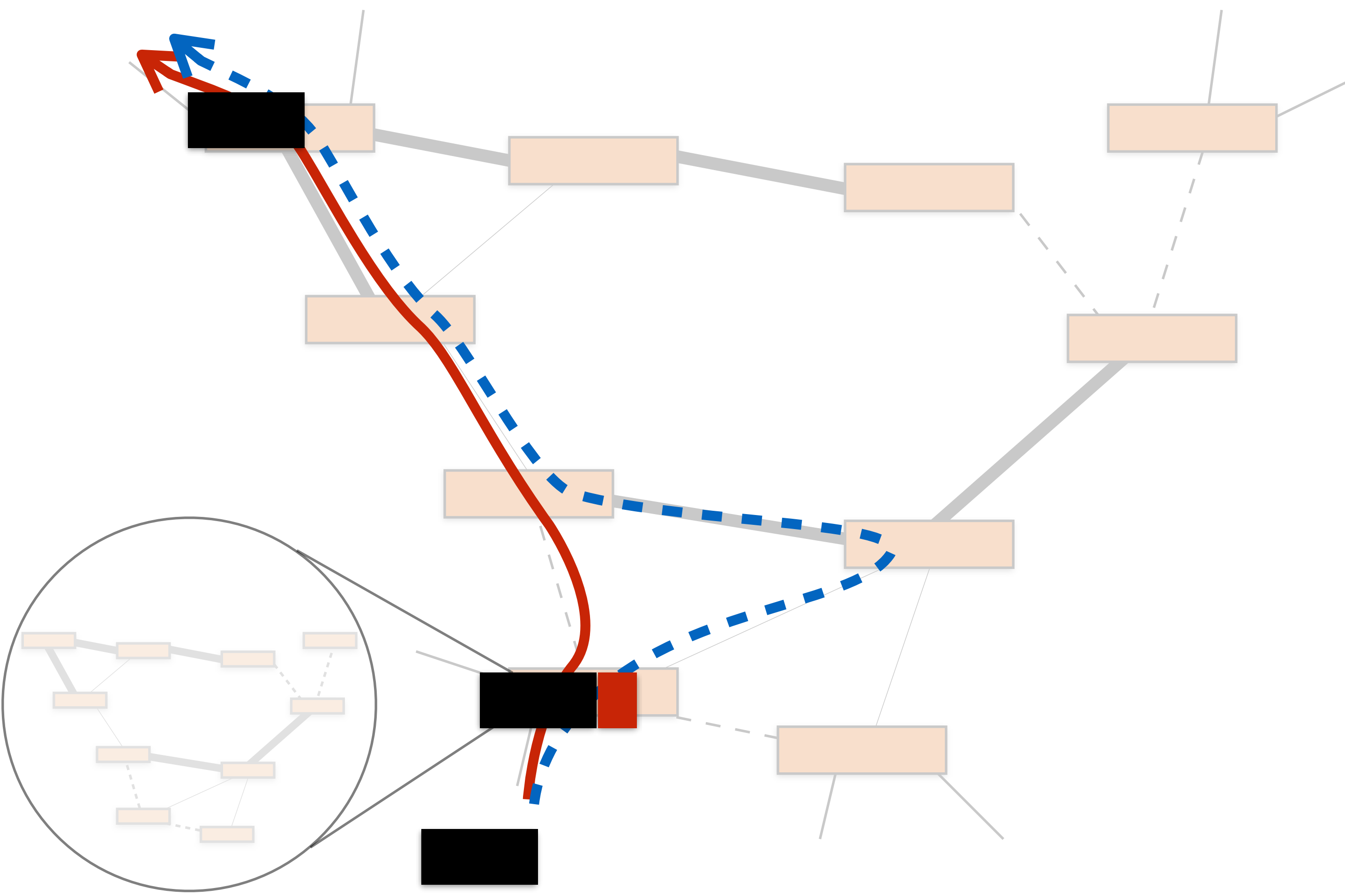
[Slide Credit: Ankit Singla]

Recall: MPLS with RSVP-TE



1. Link-state protocol (OSPF / IS-IS)
2. Also flood available bandwidth info
3. Fulfill tunnel provisioning requests
4. Update network state, flood info

Recall: MPLS with RSVP-TE



1. Link-state protocol (OSPF / IS-IS)
2. Also flood available bandwidth info
3. Fulfill tunnel provisioning requests
4. Update network state, flood info

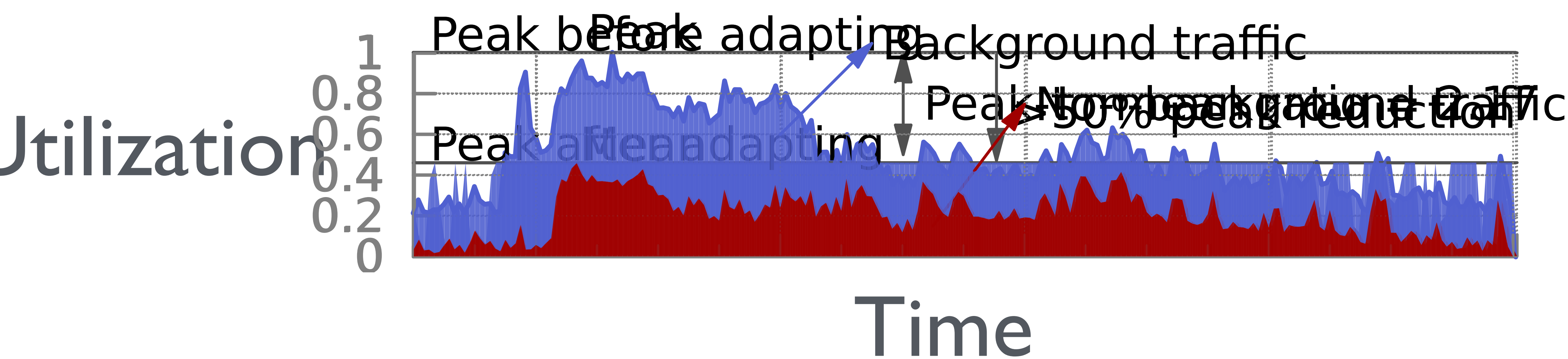
[Slide Credit: Ankit Singla]

Problem 1: inefficiency

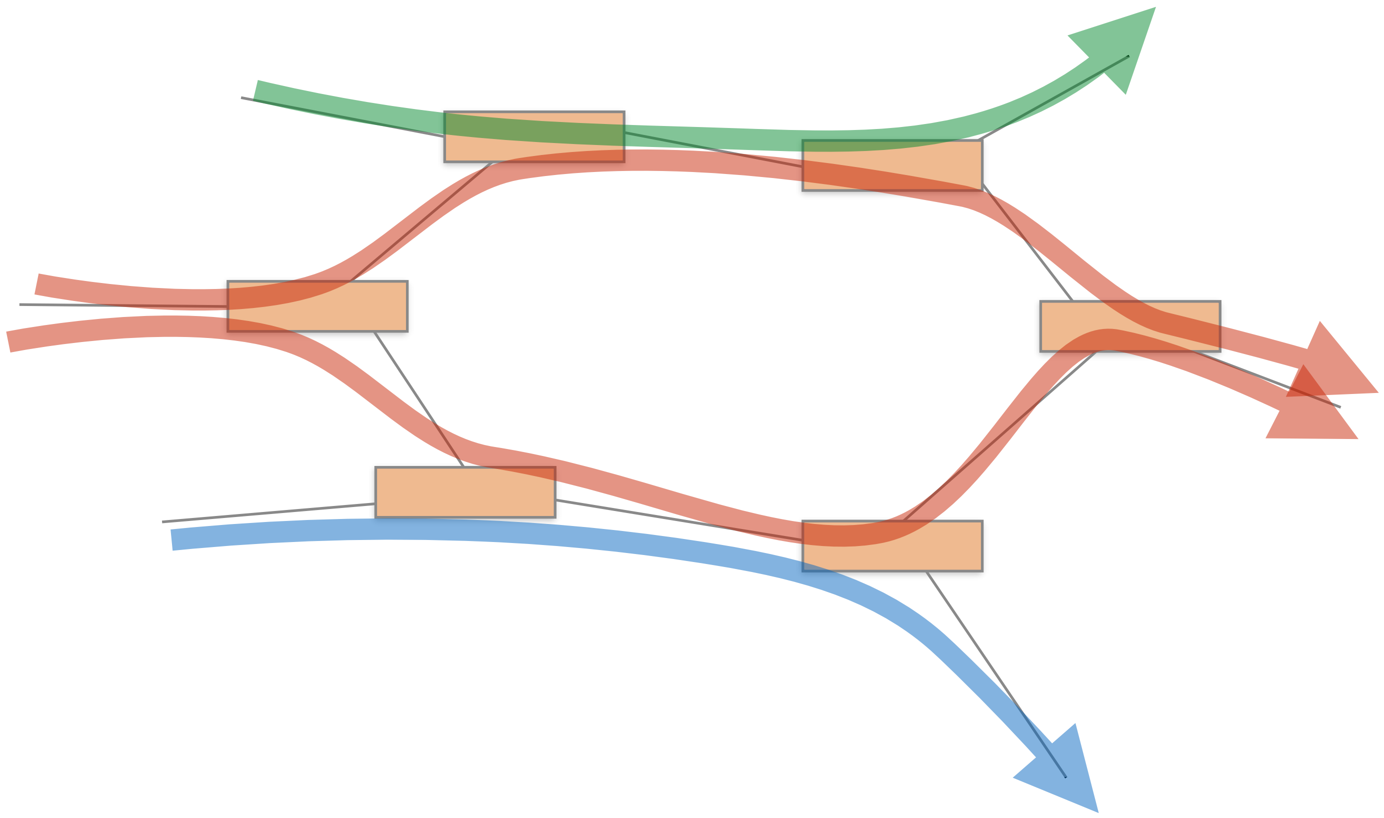
ACM SIGCOMM, 2013

Achieving High Utilization with Software-Driven WAN

Chi-Yao Hong (UIUC) Srikanth Kandula Ratul Mahajan Ming Zhang
Vijay Gill Mohan Nanduri Roger Wattenhofer (ETH)
Microsoft



Problem 2: inflexible sharing



2x the bandwidth!

B4 key design decisions

[Jain et al., SIGCOMM 2013]

Separate hardware from software

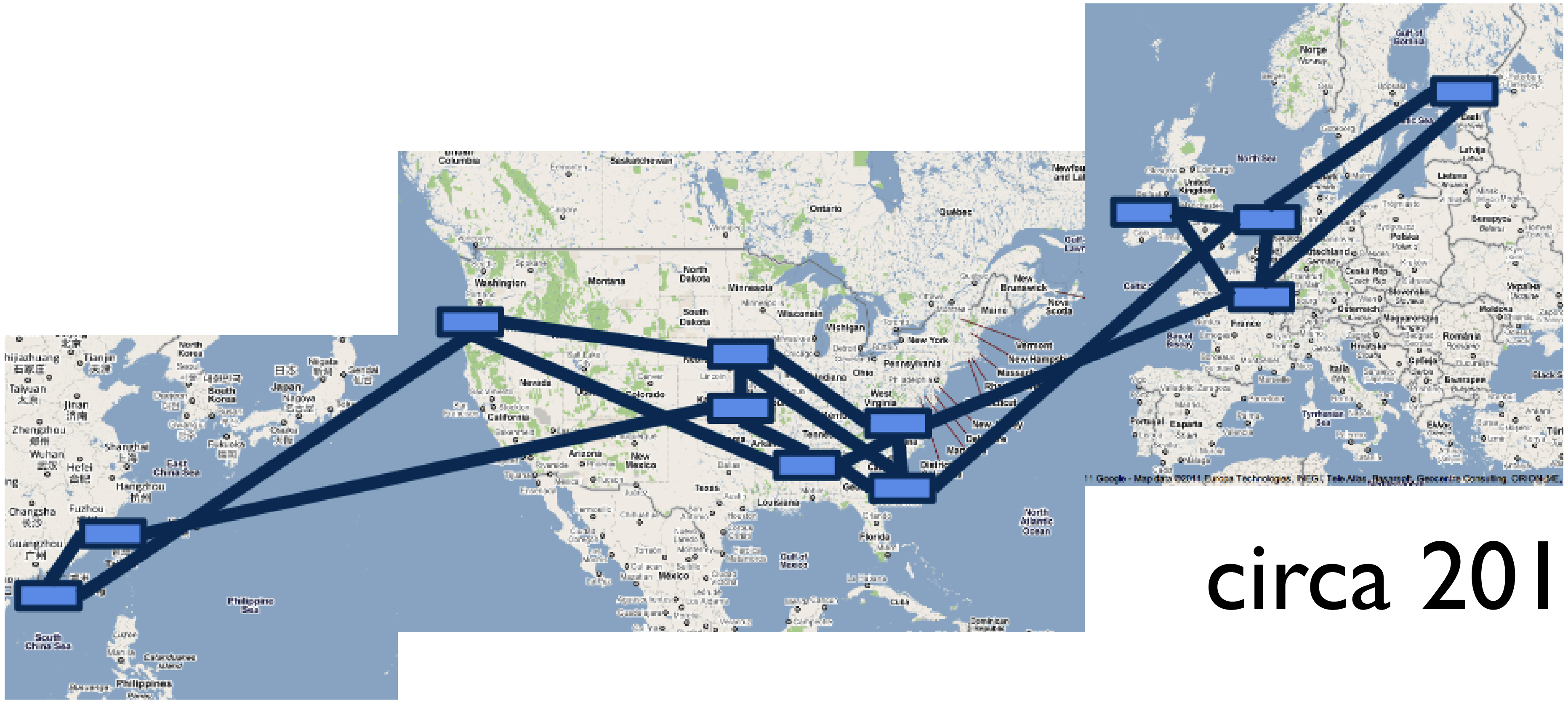
B4 routers custom-built from merchant silicon

Drive links to 100% utilization

Centralized traffic engineering

Google's B4

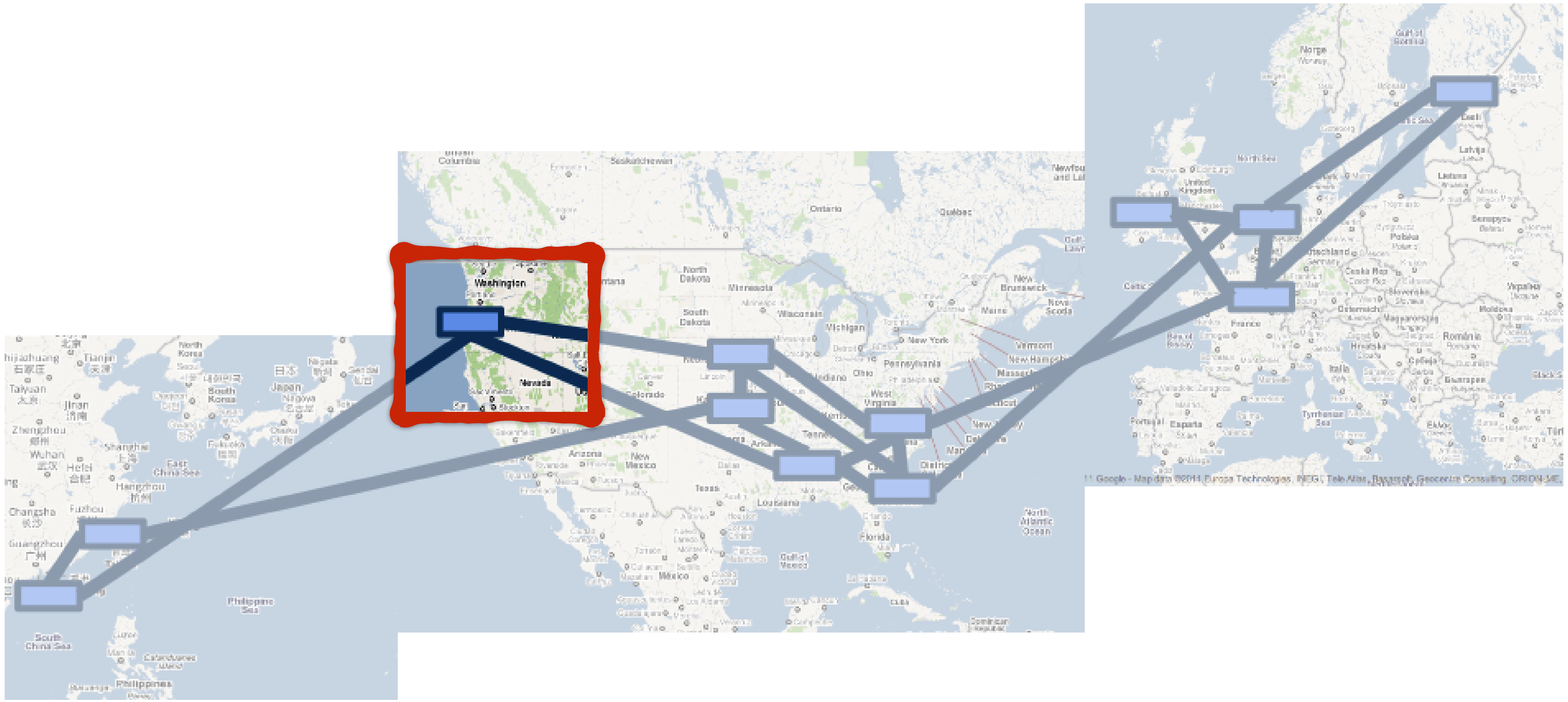
"B4: Experience with a Globally-Deployed Software Defined WAN"
Jain et al., ACM SIGCOMM 2013



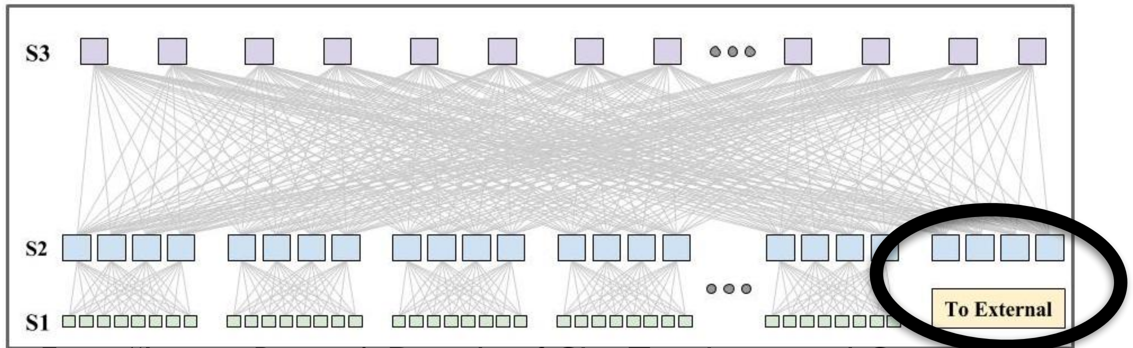
circa 2011

Google's B4

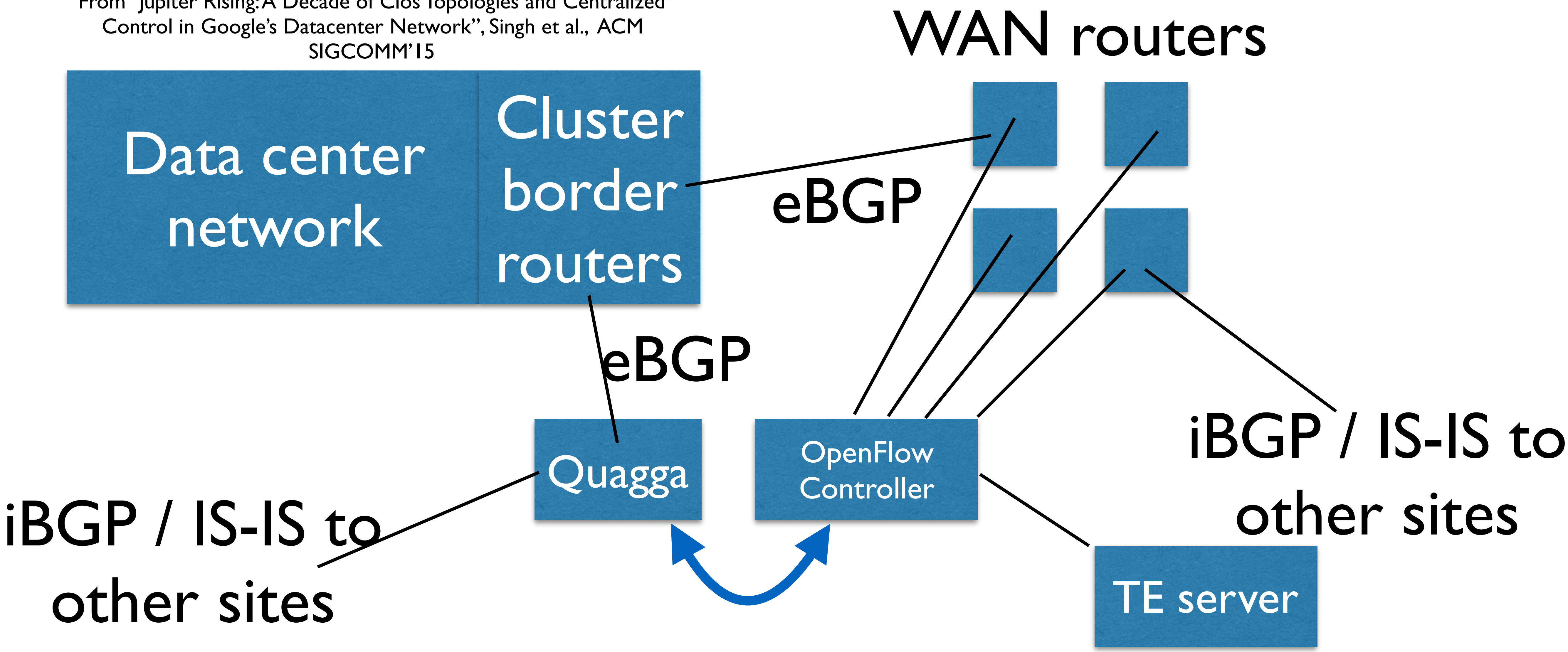
"B4: Experience with a Globally-Deployed Software Defined WAN"
Jain et al., ACM SIGCOMM 2013



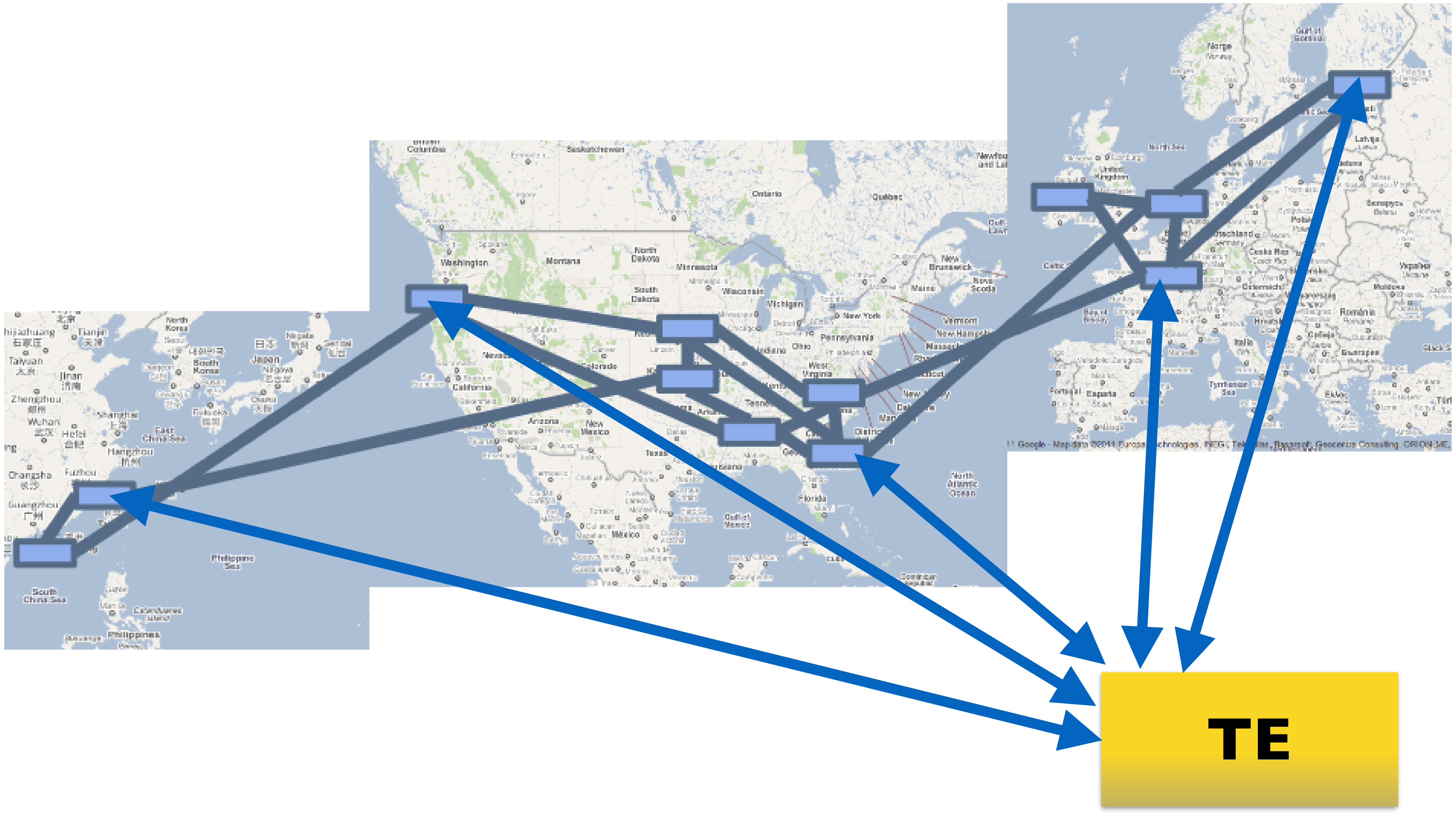
Google's B4: view at one site



From "Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network", Singh et al., ACM SIGCOMM'15



Google's B4: Traffic engineering



Vs. Semi-Distributed TE

What aspects of B4 would have been difficult with MPLS-based TE such as TeXCP?

What aspects of B4 are similar to TeXCP?

Small group discussion

1 How does B4 scale?

- Subsecond centralized scheduling of more traffic than Google's public WAN serves!

2 What does B4 assume about network's traffic?

- In what environments would these assumptions be violated?
- In what other environments would they be valid?

How does B4 scale?

How does B4 scale?

Hierarchy

- Not a simple controller-to-switch design!

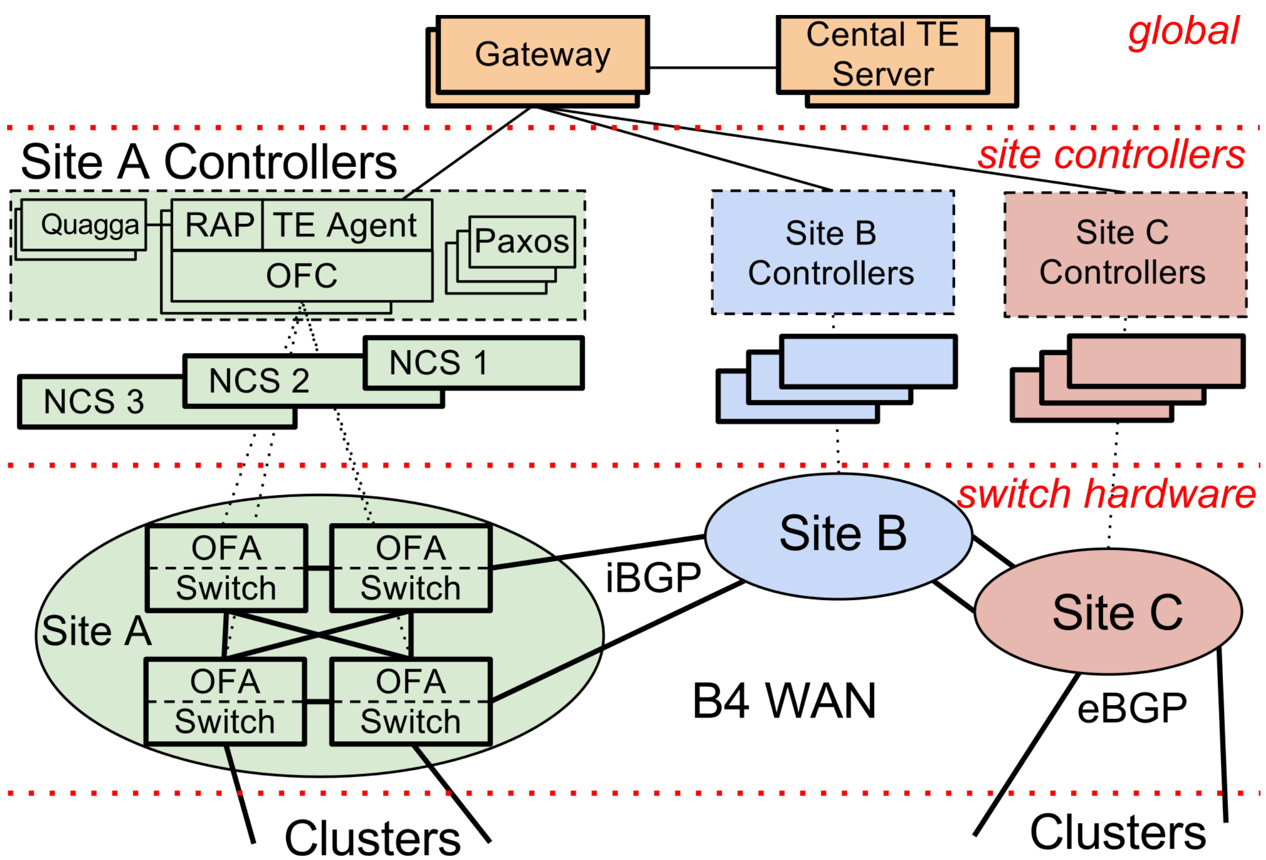


Figure 2: B4 architecture overview.

How does B4 scale?

Hierarchy

- Not a simple controller-to-switch design!

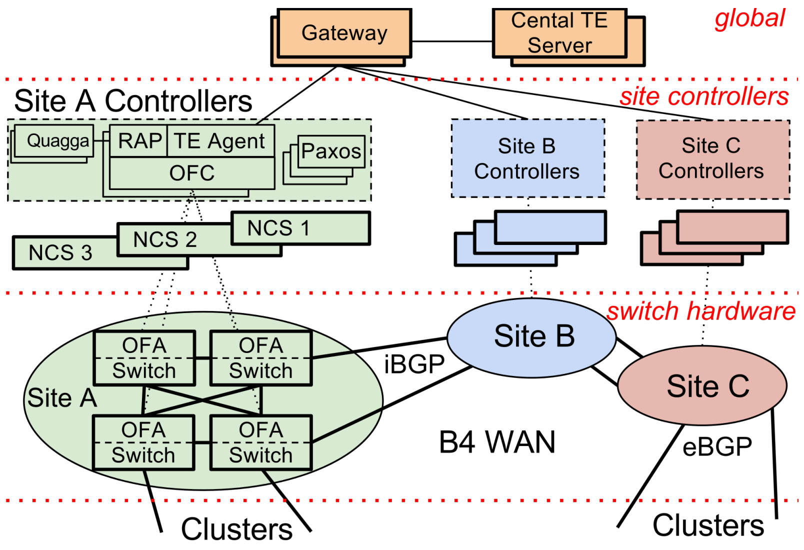


Figure 2: B4 architecture overview.

How does B4 scale?

Hierarchy

- Not a simple controller-to-switch design!

Aggregation

- Node = site (data center)
- Link = 100s of links
- Flow group = {src, dst, QoS} tuple

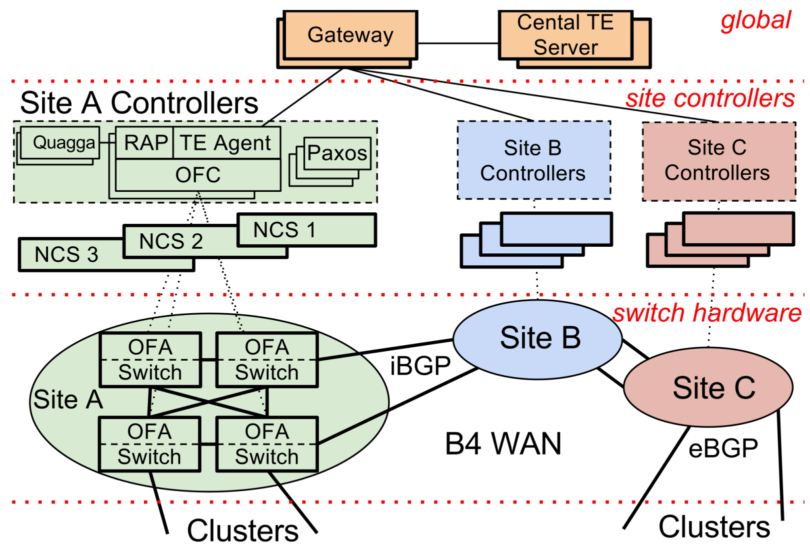


Figure 2: B4 architecture overview.

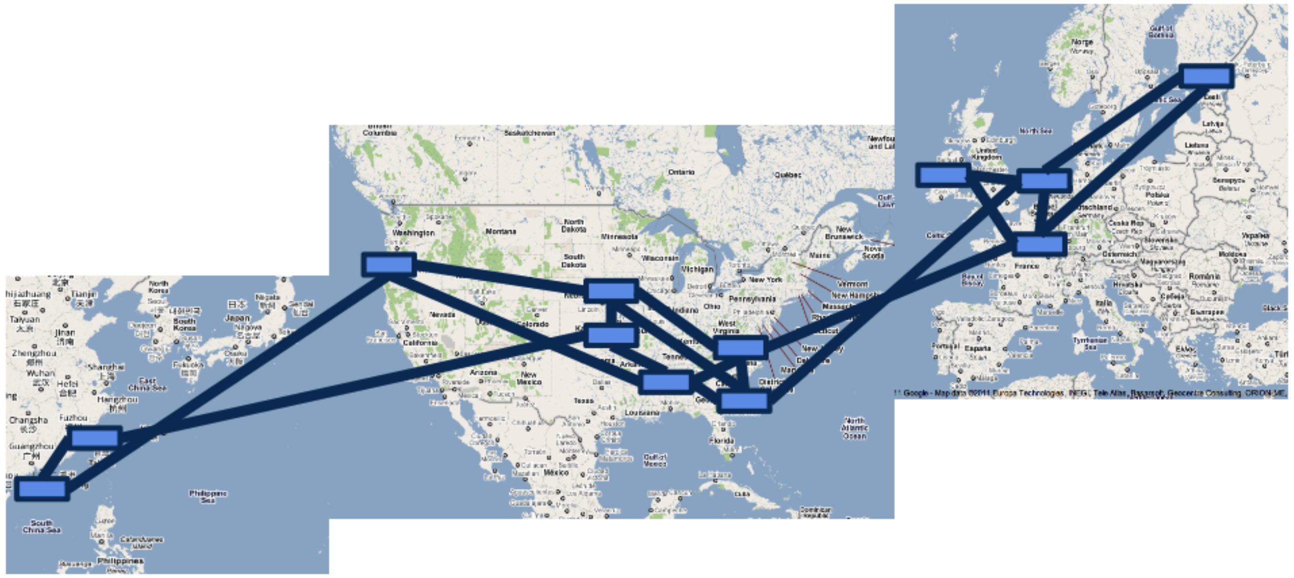


Figure 1: B4 worldwide deployment (2011).

How does B4 scale?

Hierarchy

- Not a simple controller-to-switch design!

Aggregation

- Node = site (data center)
- Link = 100s of links
- Flow group = {src, dst, QoS} tuple

Algorithms

- Greedy heuristic approximation algorithm

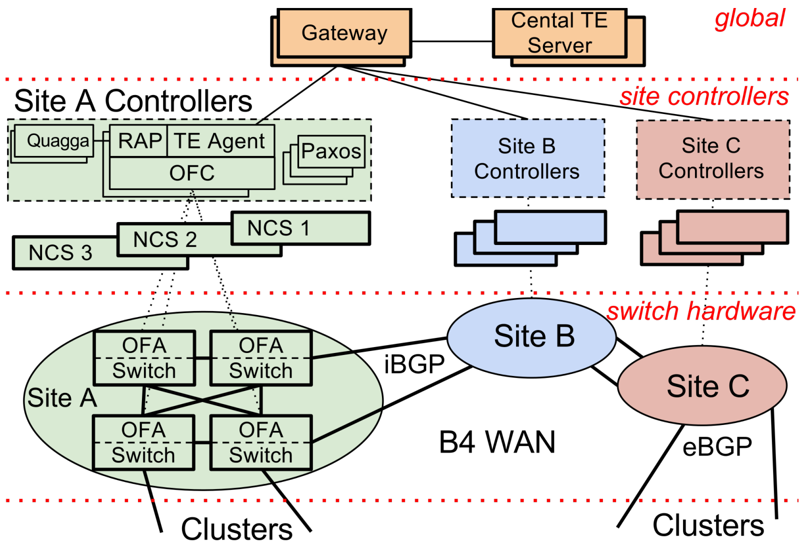


Figure 2: B4 architecture overview.

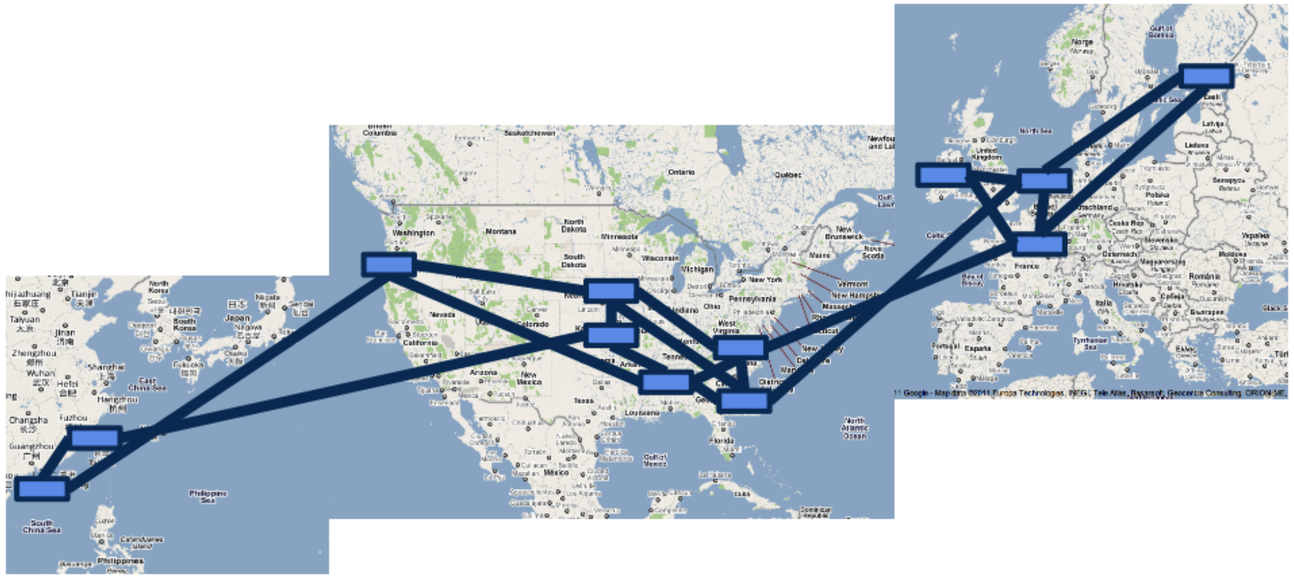
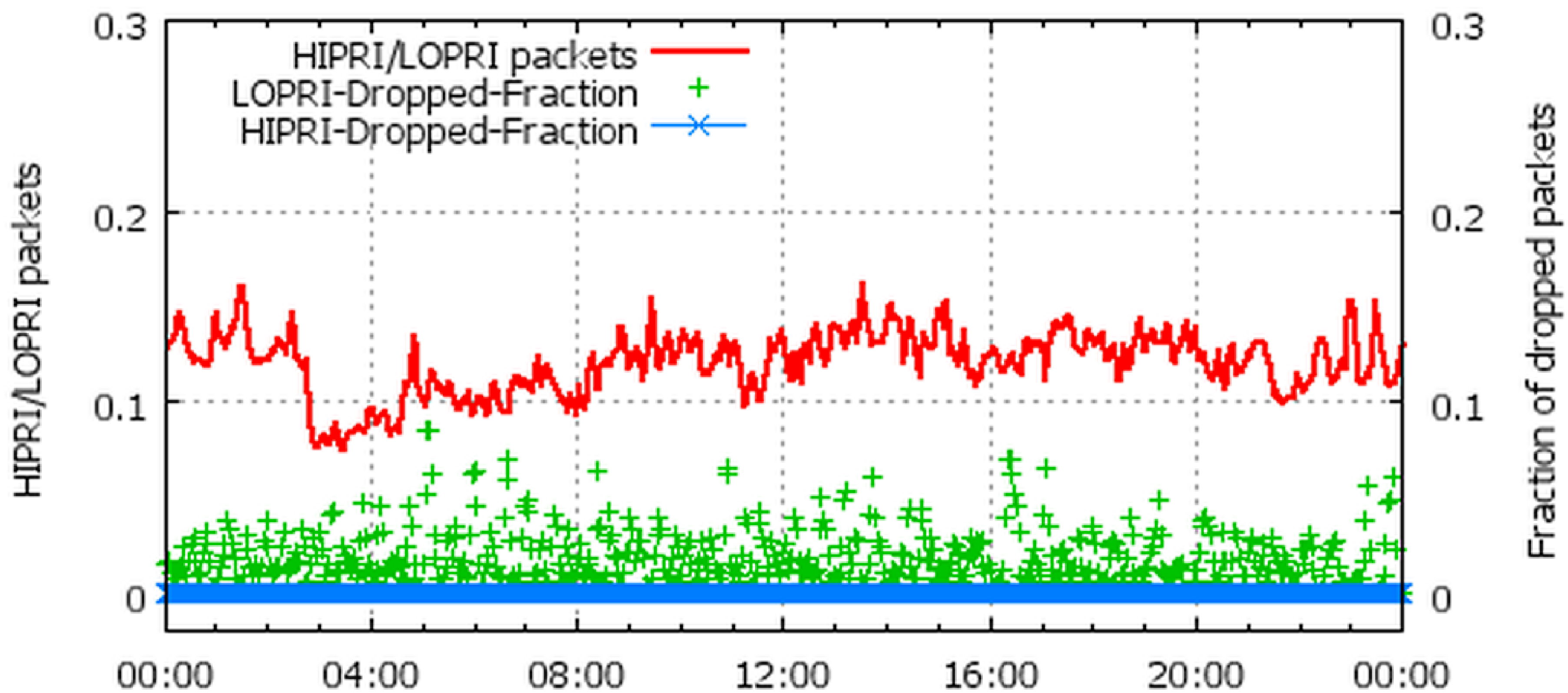


Figure 1: B4 worldwide deployment (2011).

What assumptions about traffic?

Design makes what assumption about traffic to approach 100% utilization on some links?

- High priority traffic is in the minority
- Elastic traffic is the majority (backups, offline data analytics, ...)



What assumptions about traffic?

Design makes what assumption about traffic to approach 100% utilization on some links?

- High priority traffic is in the minority
- Elastic traffic is the majority (backups, offline data analytics, ...)

When would that assumption be violated?

- Google's user-facing wide area network

Microsoft's SWAN

ACM SIGCOMM, 2013

Achieving High Utilization with Software-Driven WAN

Chi-Yao Hong (UIUC) Srikanth Kandula Ratul Mahajan Ming Zhang
Vijay Gill Mohan Nanduri Roger Wattenhofer (ETH)
Microsoft