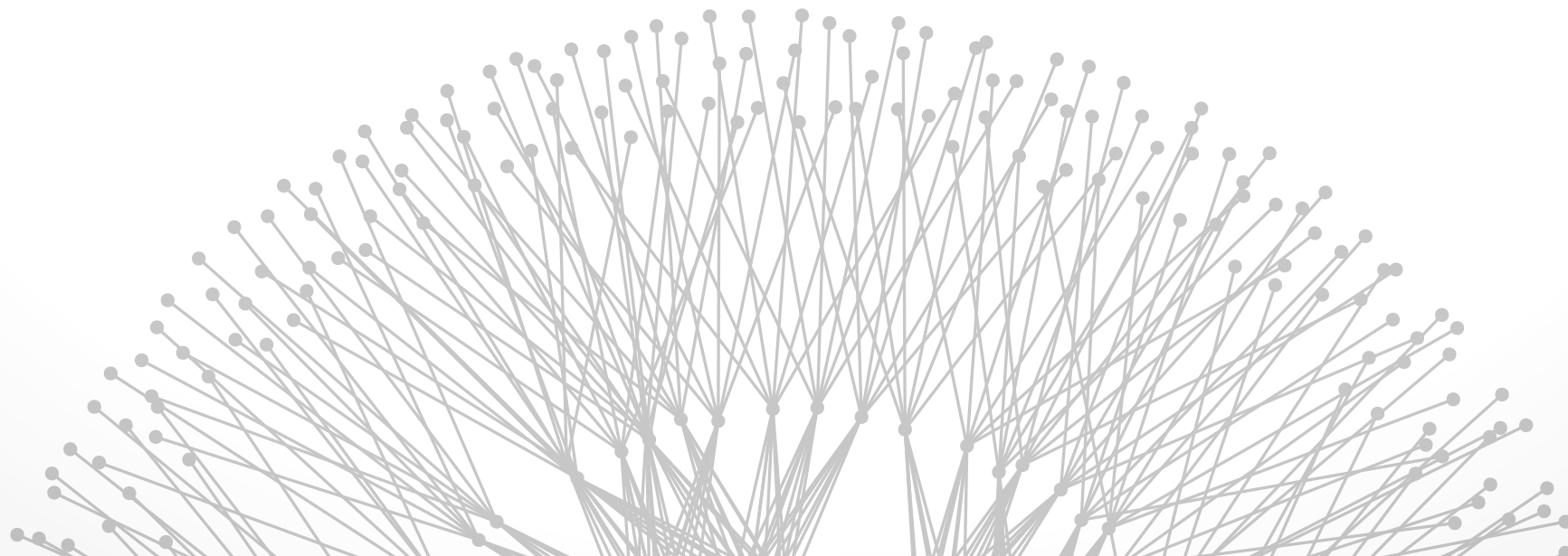


# SDN Applications

Brighten Godfrey and Chi-Yao Hong  
CS 538 October 10 2013



# 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!*



## Environments

- Mobility
- Control of wireless infrastructure
- Performance optimization
- ...

## Infrastructure / “northbound APIs”

- Programming languages
- APIs to assist policy-compliant updates
- Verification
- ...

# 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

# 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?



## 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

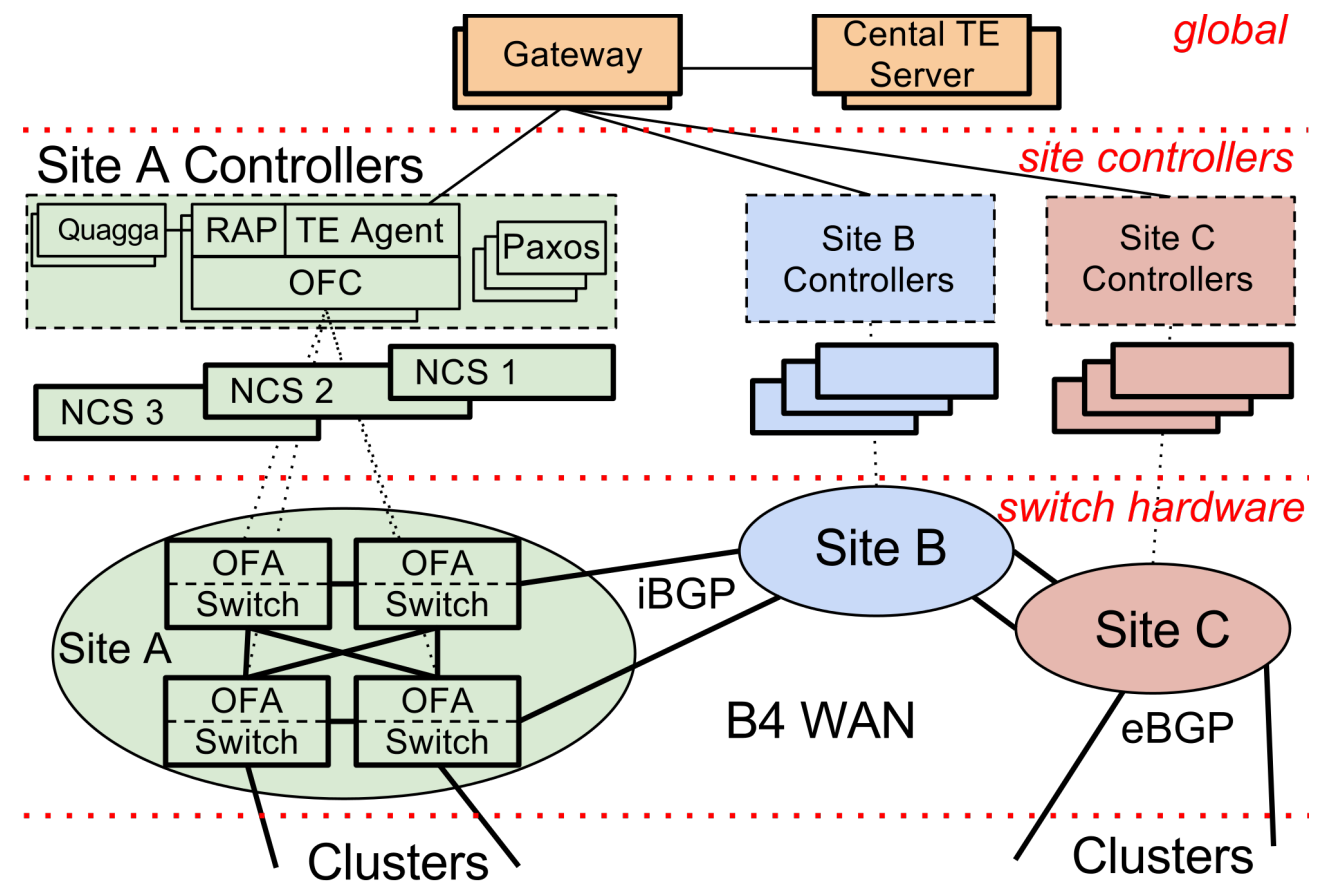


Figure 2: B4 architecture overview.



# How does B4 scale?



## Hierarchy

- Not a simple controller-to-switch design

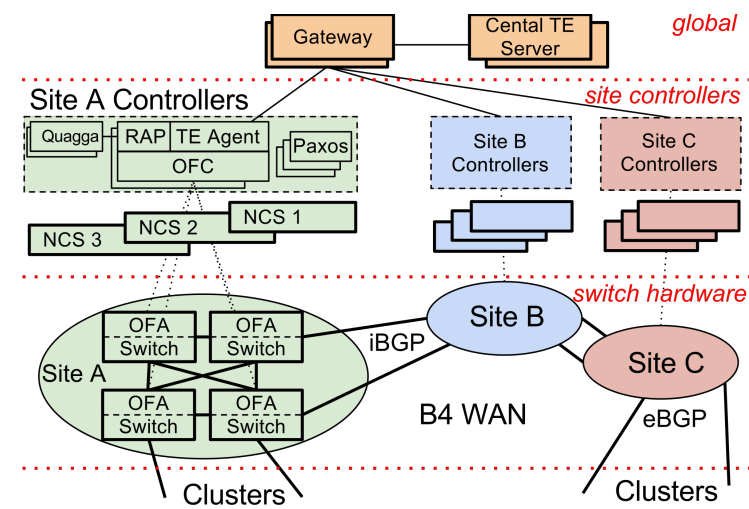


Figure 2: B4 architecture overview.

# 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

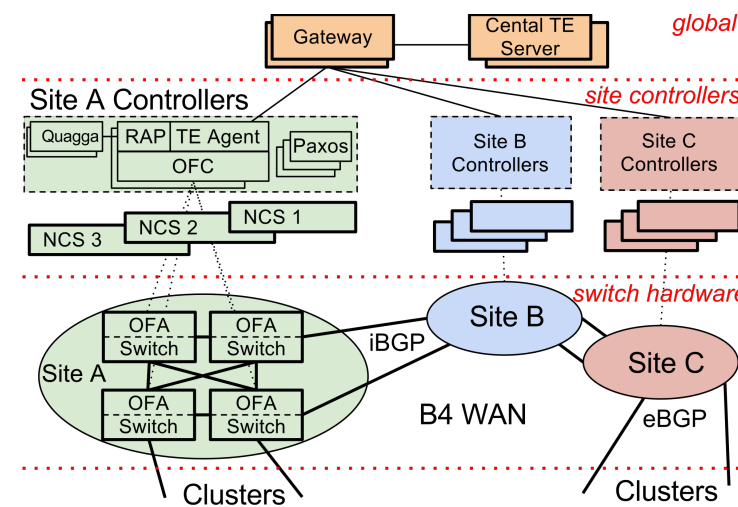


Figure 2: B4 architecture overview.

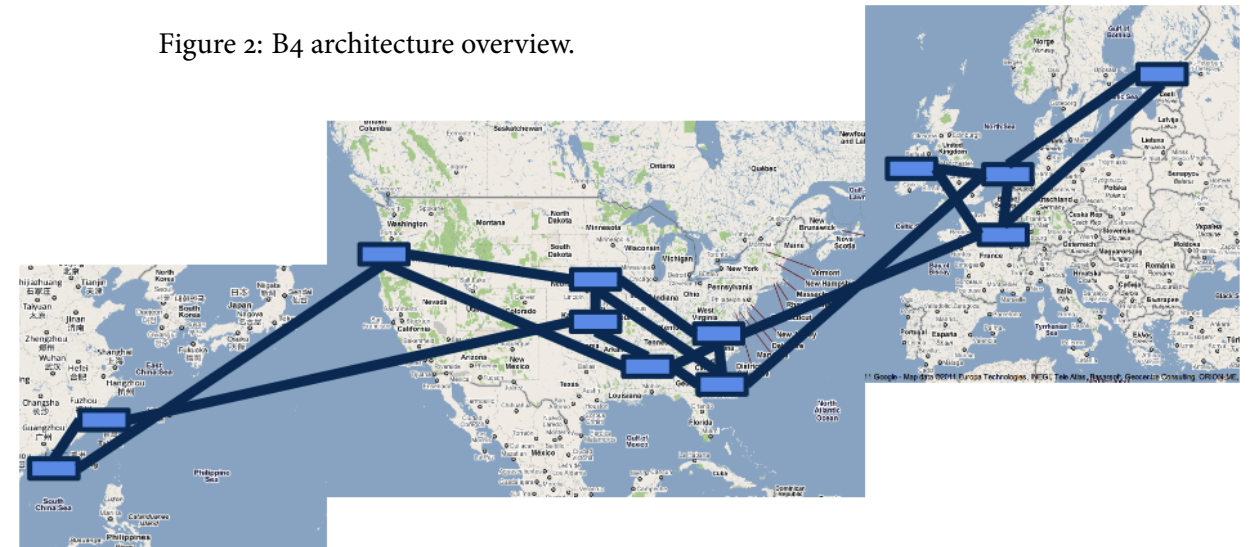


Figure 1: B4 worldwide deployment (2011).

# 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

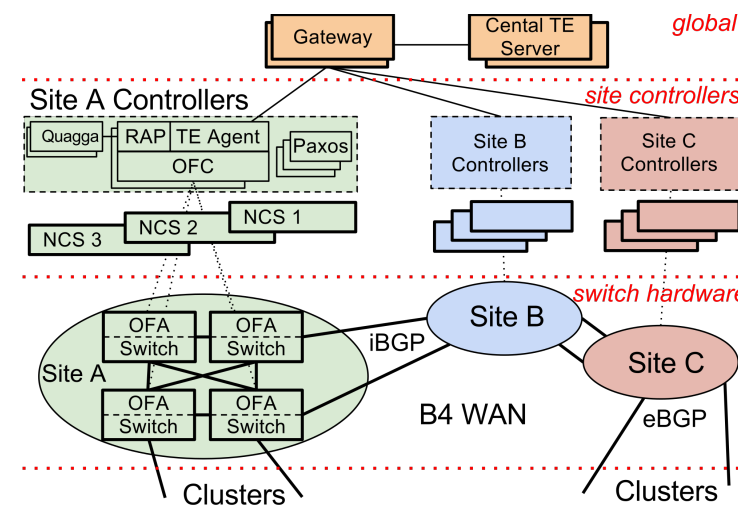


Figure 2: B4 architecture overview.

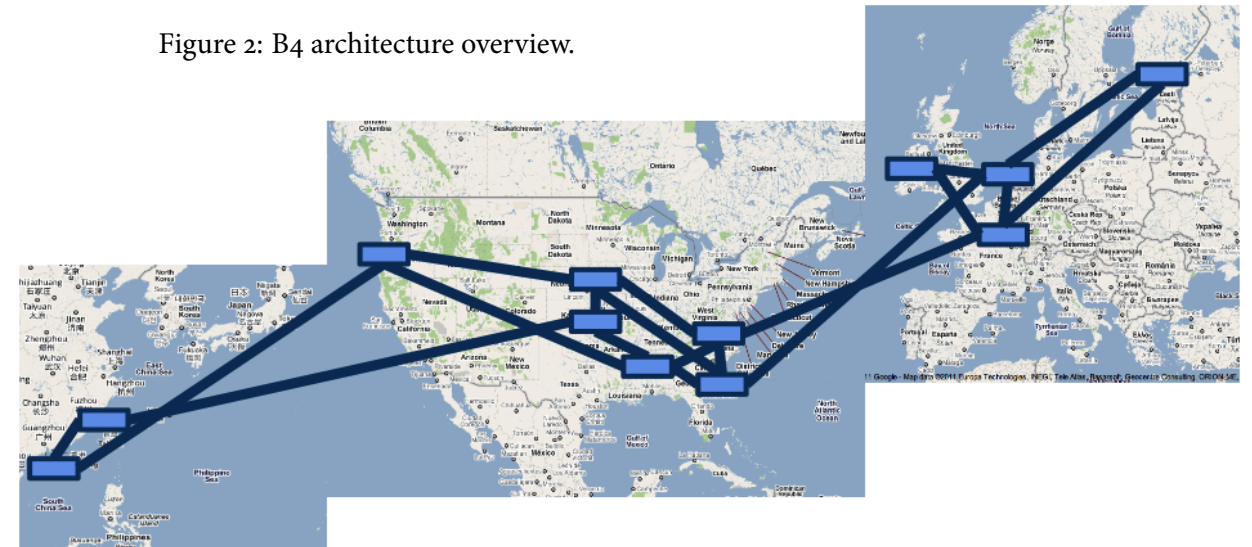


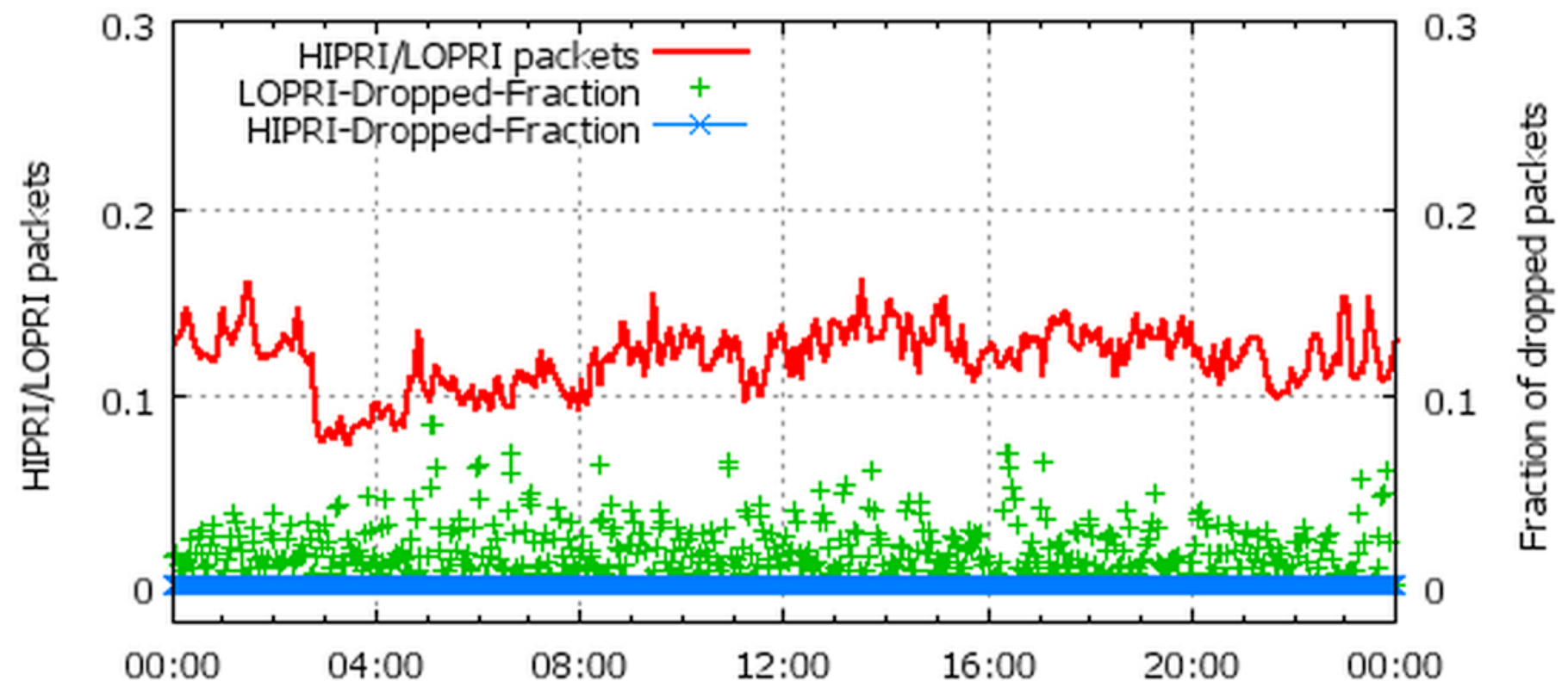
Figure 1: B4 worldwide deployment (2011).

# 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