Symbiosis in Scale Out Networking and Data Management

Amin Vahdat
Google/UC San Diego

vahdat@google.com

Overview

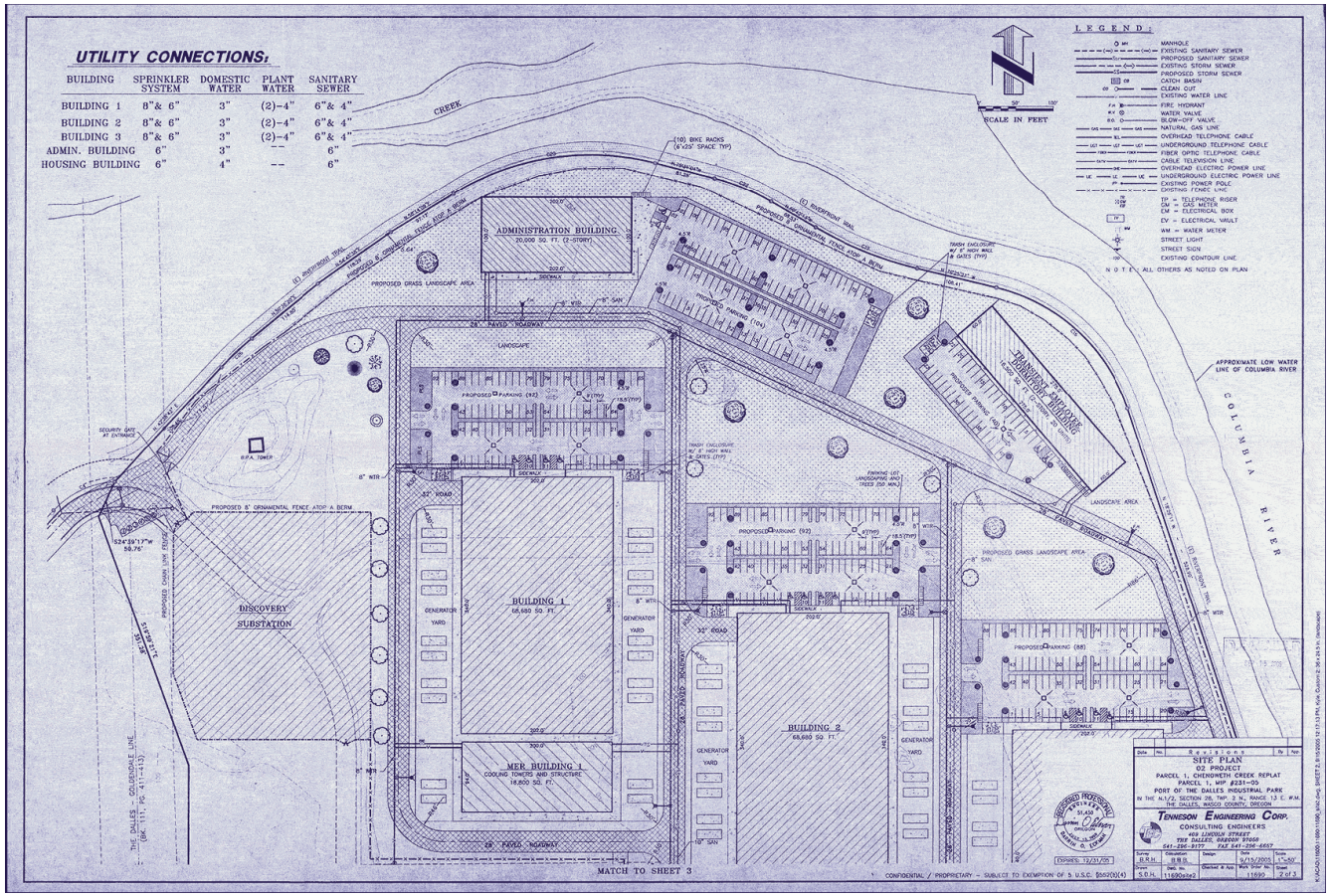
- Large-scale data processing needs scale out networking
  - Unlocking the potential of modern server hardware for at scale problems requires orders of magnitude improvement in network performance
- Scale out networking requires large-scale data management
  - Experience with Google's SDN WAN suggests that logically centralized state management critical for cost-effective deployment and management
  - Still in the stone ages in dynamically managing state and getting updates to the right places in the network

Overview

- Large-scale data processing needs scale out networking
- **WARNING: Networking is about to reinvent many aspects of centrally managed, replicated state with variety of consistency requirements in a distributed environment**
  - Experience with Google's SDN WAN suggests that logically centralized state management critical for cost-effective deployment and management
  - Still in the stone ages in dynamically managing state and getting updates to the right places in the network

Vignette 1: Large-Scale Data Processing Needs Scale Out Networking

Motivation



Blueprints for 200k sq. ft. Data Center in OR

San Antonio Data Center

Chicago Data Center

Dublin Data Center

All Filled with Commodity Computation and Storage

Network Design Goals

- Scalable interconnection bandwidth
  - Full bisection bandwidth between all pairs of hosts
    - Aggregate bandwidth = # hosts × host NIC capacity
- Economies of scale
  - Price/port constant with number of hosts
  - Must leverage commodity merchant silicon
- Anything anywhere
  - Don’t let the network limit benefits of virtualization
- Management
  - Modular design
  - Avoid actively managing 100’s-1000’s network elements

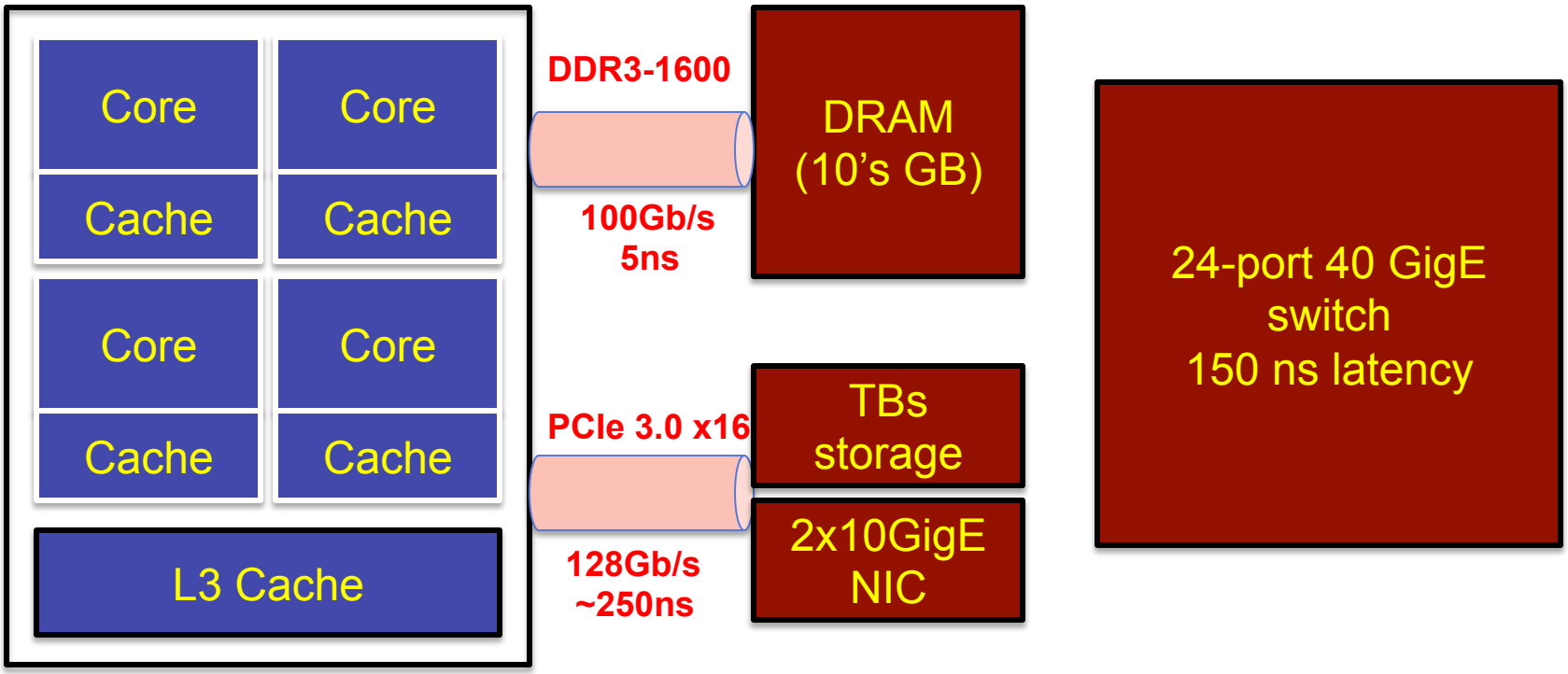
Scale Out Networking

- Advances toward *scale out* computing and storage
  - Aggregate computing and storage grows linearly with the number of commodity processors and disks
  - Small matter of software to enable functionality
  - Alternative is *scale up* where weaker processors and smaller disks are *replaced* with more powerful parts
- Today, no technology for scale out networking
  - Modules to expand number of ports or aggr BW
  - No management of individual switches, VLANs, subnets

The Future Internet

- Applications and data will be partitioned and replicated across multiple data centers
  - 99% of compute, storage, communication will be inside the data center
  - Data Center Bandwidth Exceeds that of the Access
- Data sizes will continue to explode
  - From click streams, to scientific data, to user audio, photo, and video collections
- Individual user requests and queries will run in parallel on thousands of machines
- Back end analytics and data processing will dominate

Emerging Rack Architecture



- Can we leverage emerging *merchant switch* and newly proposed *optical transceivers and switches* to treat entire data center as single logical computer

Amdahl's (Lesser Known) Law

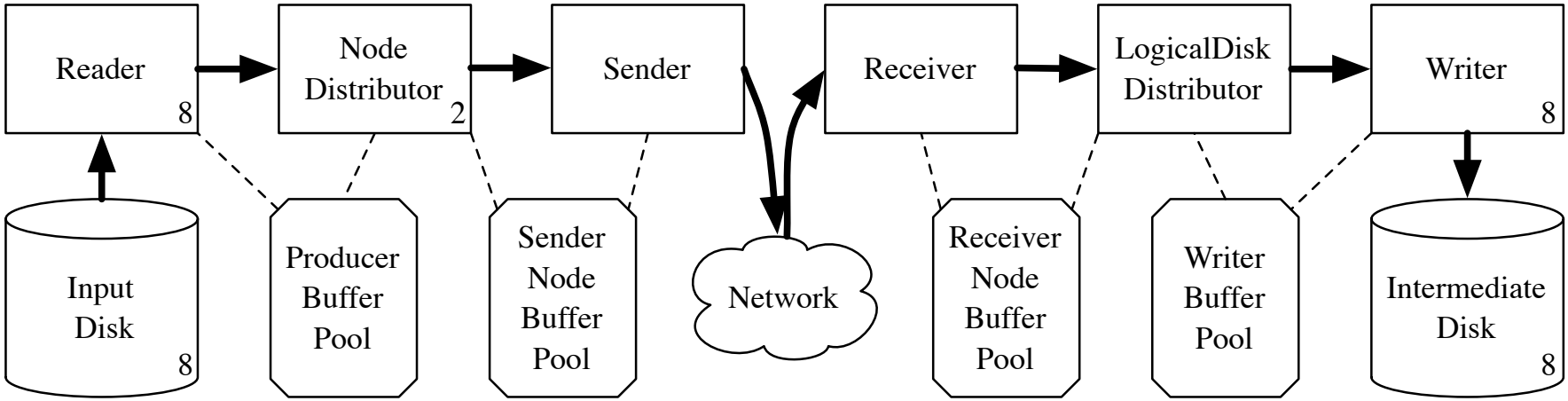
- Balanced Systems for parallel computing
- For every 1Mhz of processing power must have
  - 1MB of memory
  - 1 Mbit/sec I/O
  - In the late 1960's
- Fast forward to 2012
  - 4x2.5Ghz processors, 8 cores
  - 30-60Ghz of processing power (not that simple!)
  - 24-64GB memory
  - But 1Gb/sec of network bandwidth??
- Deliver 40 Gb/s bandwidth to 100k servers?
  - 4 Pb/sec of bandwidth required *today*

Sort as Instance of Balanced Systems

- Hypothesis: significant efficiency lost in systems that bottleneck on one resource
- Sort as example
- Gray Sort 2009 record
  - 100 TB in 173 minutes on 3452 servers
  - ~22.3 Mb/s/server
- Out of core sort: 2 reads and 2 writes required
- What would it take to sort 3.2 Gb/s/server?
  - 4x100 MB/sec/node with 16 500 GB-disks/server
  - 100 TB in 83 minutes on 50 server?

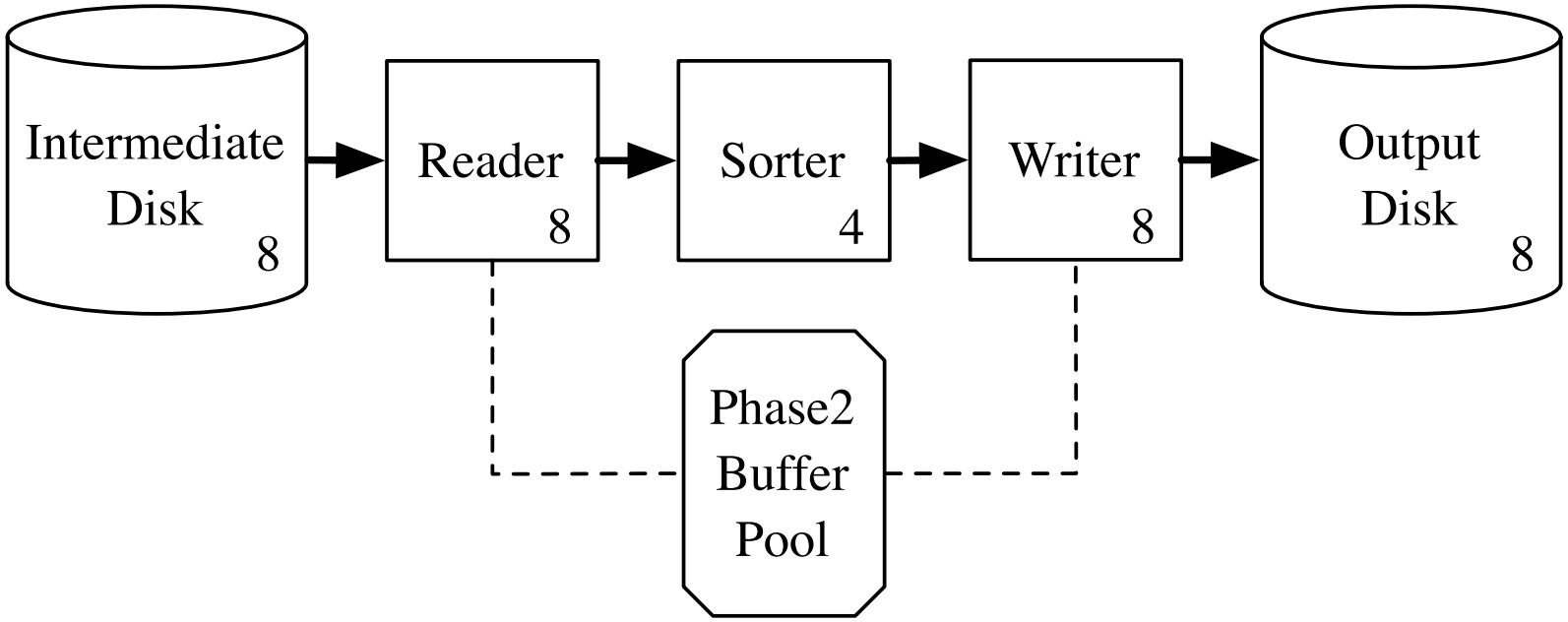
TritonSort Phase 1

Map and Shuffle



TritonSort Phase 2

Reduce



Reverse Engineering the Pipeline

- Goal: minimize number of logical disks
  - Phase 2: read, sort, write (repeat)
  - One sorter/core
  - Need 24 buffers (3/core)
  - ~20GB/server: 830MB/logical disk
  - 2TB/830MB/logical disk ➔ ~2400 logical disks
- Long pole in phase 1: LogicalDiskDistributor buffering sufficient data for streaming write
  - ~18GB/2400 logical disks = 7.5MB buffer
  - ~15% seek penalty

Balanced Systems Really Do Matter

- Balancing network and I/O results in huge efficiency improvements
  - How much is a factor of 100 improvement worth in terms of cost?
  - "TritonSort: A Balanced Large-scale Sorting System," Rasmussen, et al., NSDI 2011.

System	Duration	Aggr. Rate	Servers	Rate/server
Yahoo (100TB)	**173 min**	**9.6 GB/s**	**3452**	**2.8 MB/s**
TritonSort (100TB)	**107 min**	**15.6 GB/s**	**52**	**300 MB/s**

TritonSort Results

- **http://www.sortbenchmark.org**
- Hardware
  - HP DL-380 2U servers, 8-2.5 Ghz cores, 24 GB RAM, 16x500-GB disks, 2x10 Gb/s Myricom NICs
  - 52-port Cisco Nexus 5020 switch
- Results 2010
  - GraySort: 100 TB in 123 mins/48 nodes, 2.3 Gb/s/server
  - MinuteSort: 1014 GB in 59 secs/52 nodes, 2.6 Gb/s/server
- Results 2011
  - GraySort: 100 TB in 107 mins/52 nodes, 2.4 Gb/s/server
  - MinuteSort: 1353 GB in 1 min/52 nodes, 3.5 Gb/s/server
  - JouleSort: 9700 records/Joule

Generalizing TritonSort – Themis-MR

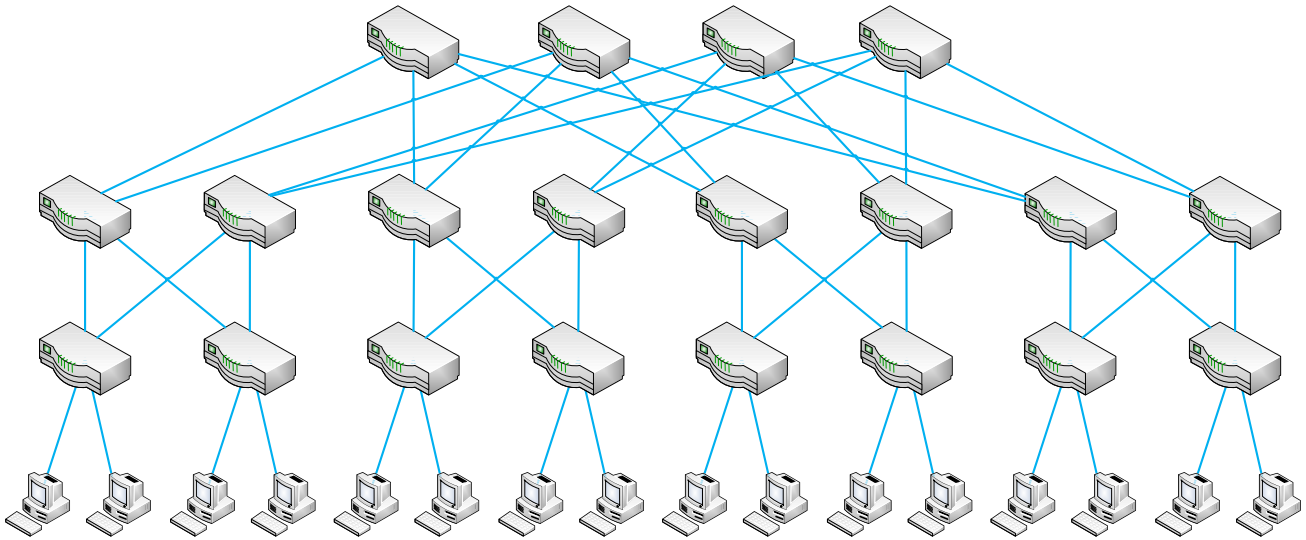
- TritonSort's very constrained
  - 100B records, even key distribution
- Generalize with same performance?
  - MapReduce natural choice: map → **sort** → reduce
- **Skew:**
  - Partition, compute, record size, …
  - Memory management now hard
- Task-level to job-level fault tolerance for performance
  - Long tail of small- to medium-sized jobs on <= 1PB of data

Current Status

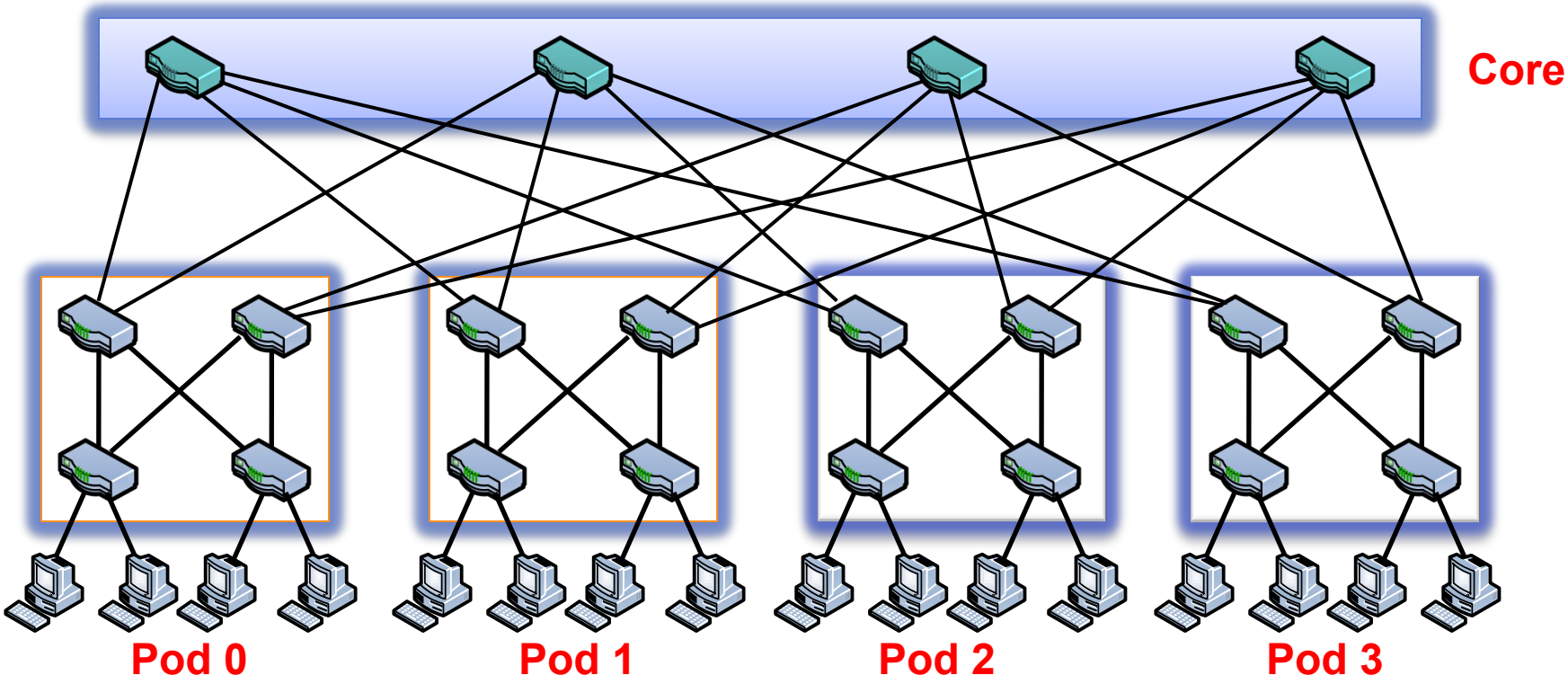
- Themis-MR outperforms Hadoop 1.0 by ~8x on 28 node, 14TB GraySort
  - 30 minutes vs. 4 hours
- Implementations of CloudBurst, PageRank, Word Count being evaluated
- Alpha version won 2011 Daytona GraySort
  - Beat previous record holder by 26%, 1/70 nodes

Driver: Nonblocking Multistage Datacenter Topologies

M. Al-Fares, A. Loukissas, A. Vahdat. A Scalable, Commodity Data Center Network Architecture. In SIGCOMM ’08.

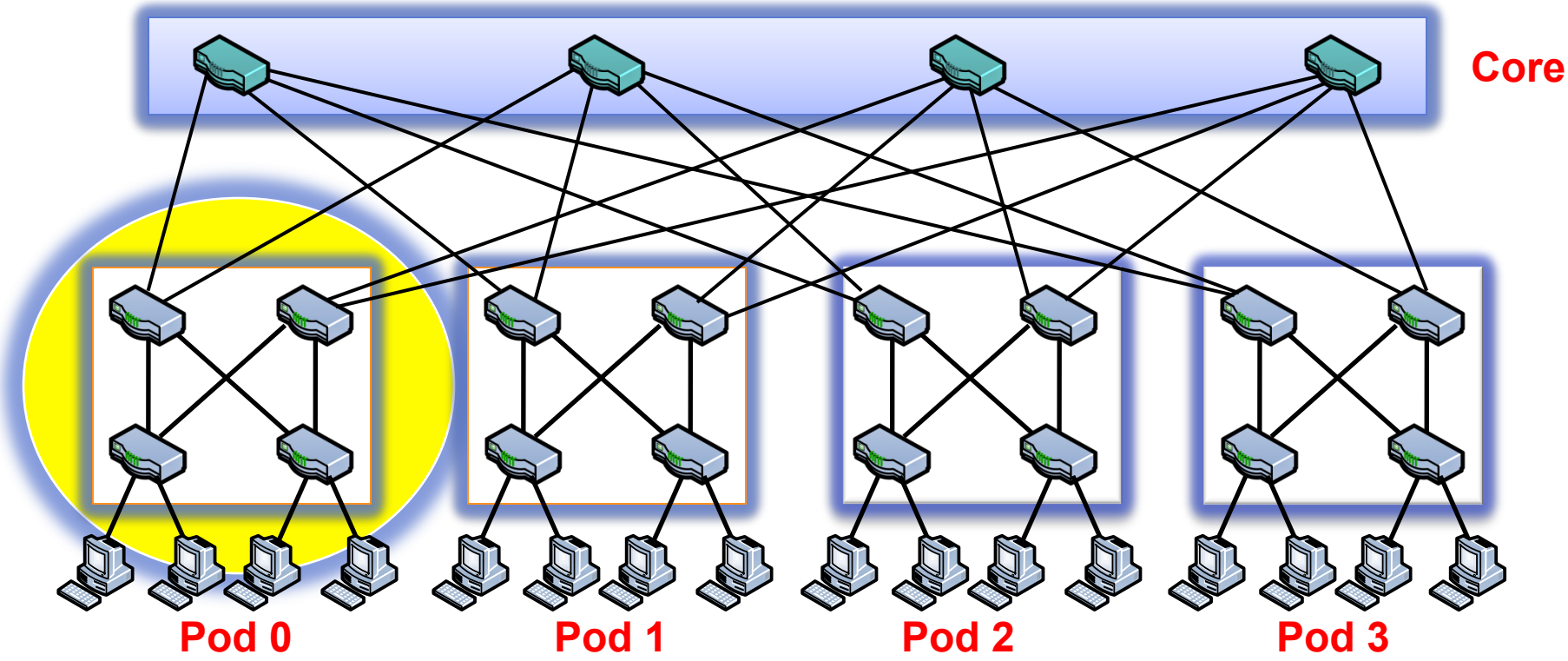


Scalability Using Identical Network Elements



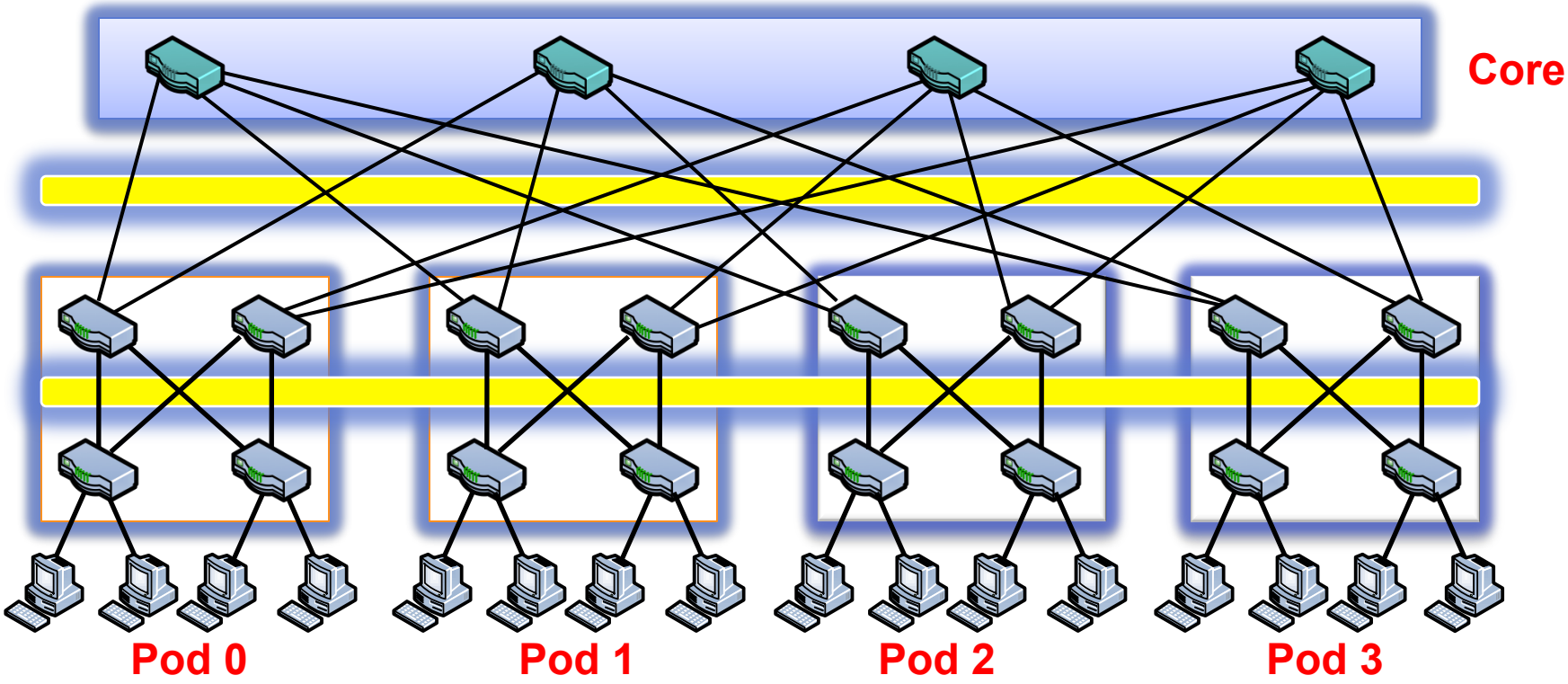
- Fat tree built from 4-port switches

Scalability Using Identical Network Elements



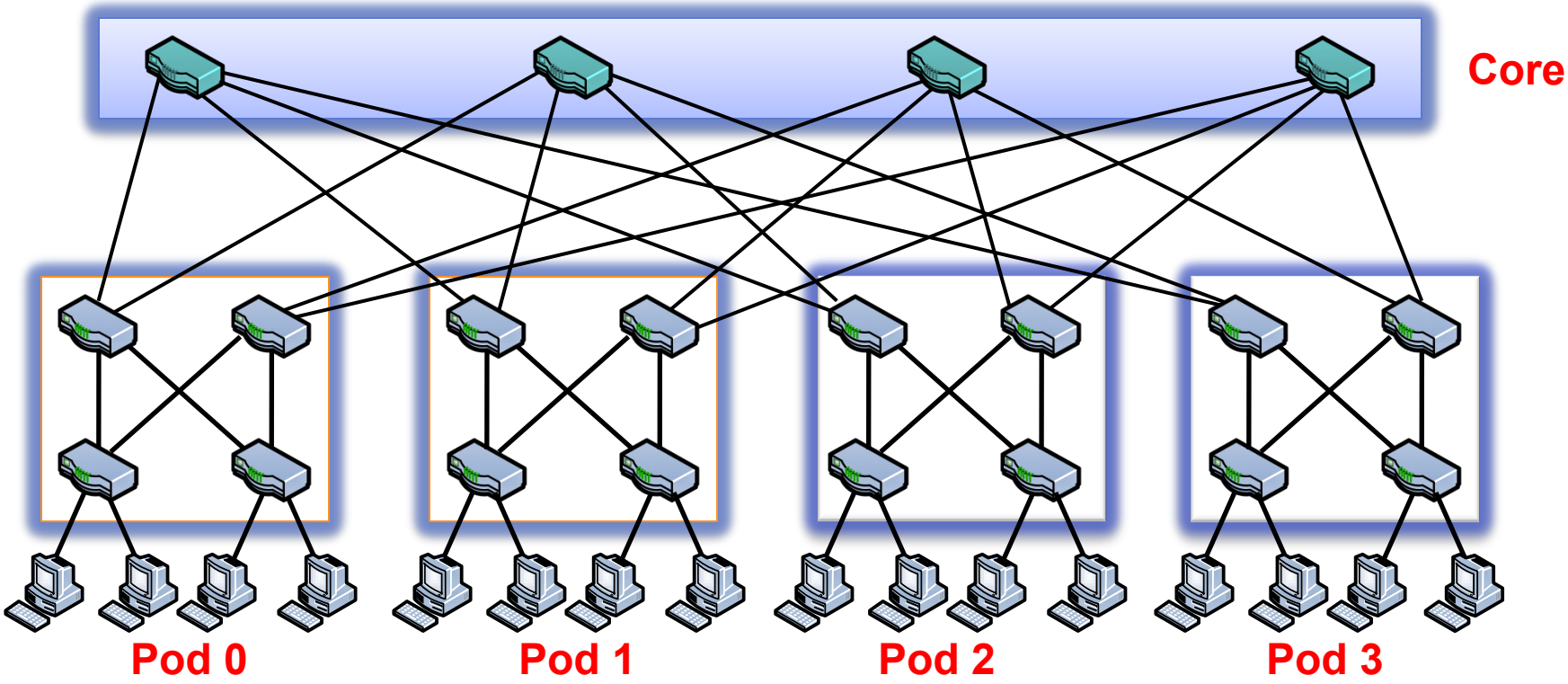
- Support 16 hosts organized into 4 pods
  - Each pod is a 2-ary 2-tree
  - Full bandwidth among pod-connected hosts

Scalability Using Identical Network Elements



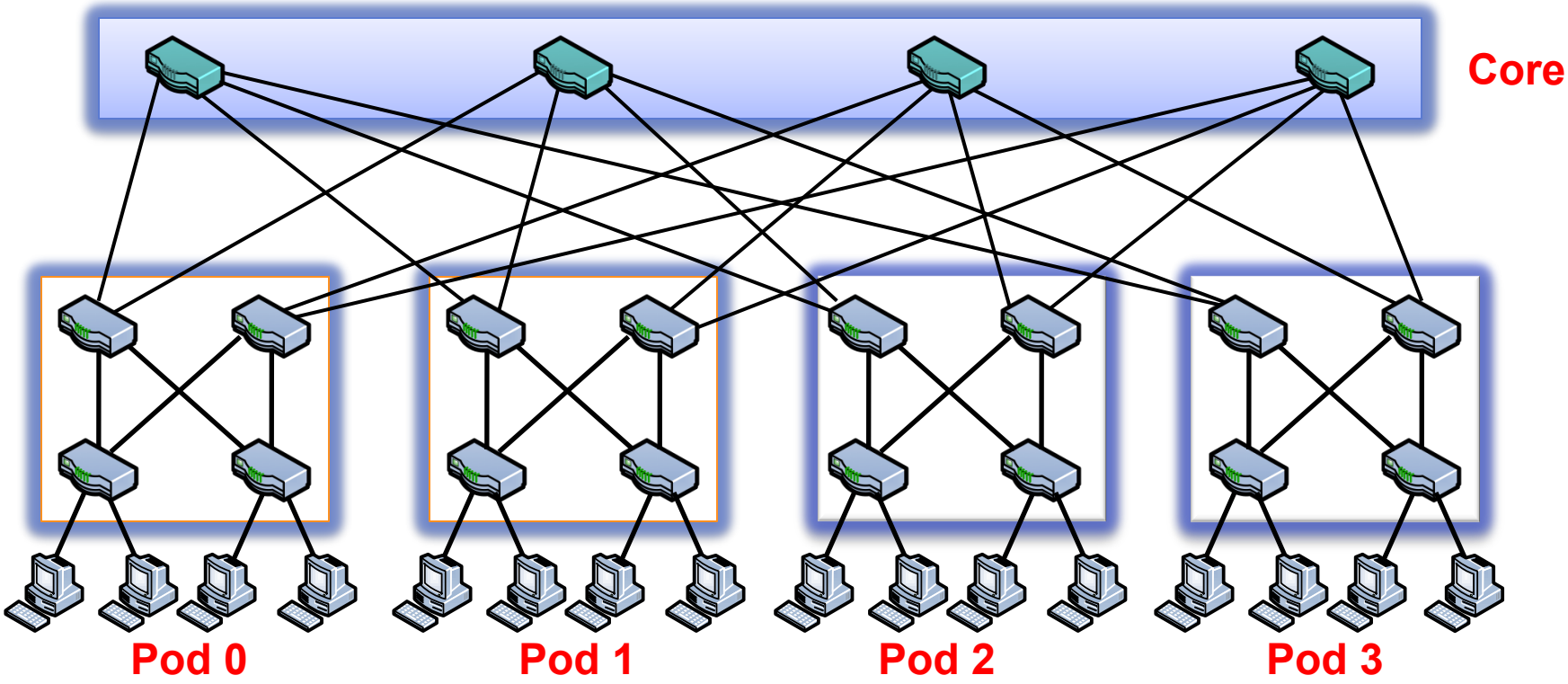
- Full bisection bandwidth at each level of fat tree
  - Rearrangeably Nonblocking
  - Entire fat-tree is a 2-ary 3-tree

Scalability Using Identical Network Elements



- $(5k^2/4)$ k-port switches support $k^3/4$ hosts
  - 48-port switches: 27,648 hosts using 2,880 switches
- Critically, approach scales to 10 GigE at the edge

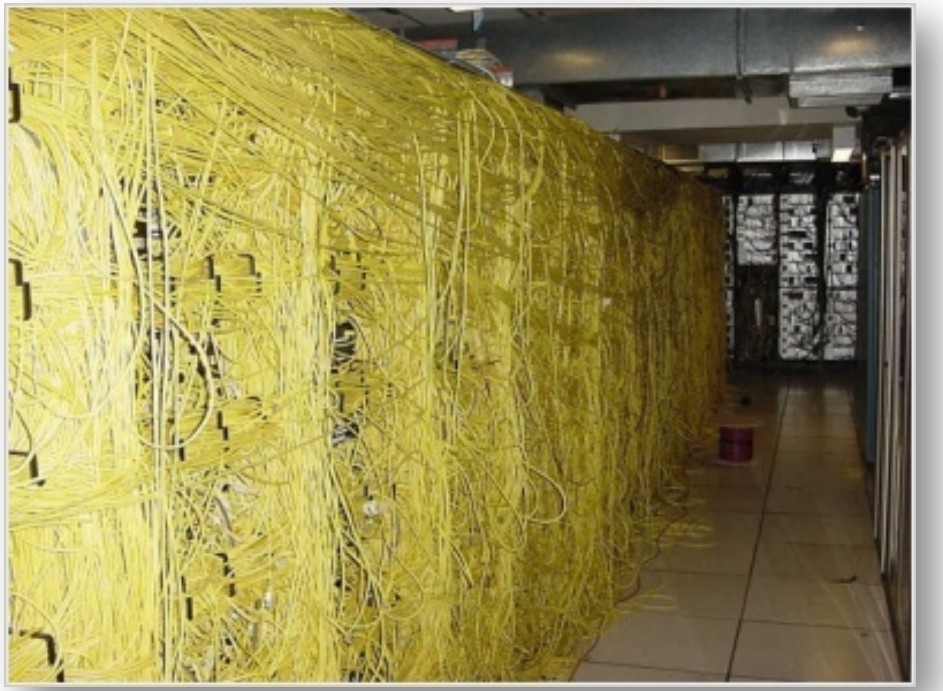
Scalability Using Identical Network Elements



- Regular structure simplifies design of network protocols
- Opportunities: performance, cost, energy, fault tolerance, incremental scalability, etc.

Problem - 10 Tons of Cabling

- 55,296 Cat-6 cables
- 1,128 separate cable bundles
- If optics used for transport, transceivers are ~80% of cost of interconnect



Our Work

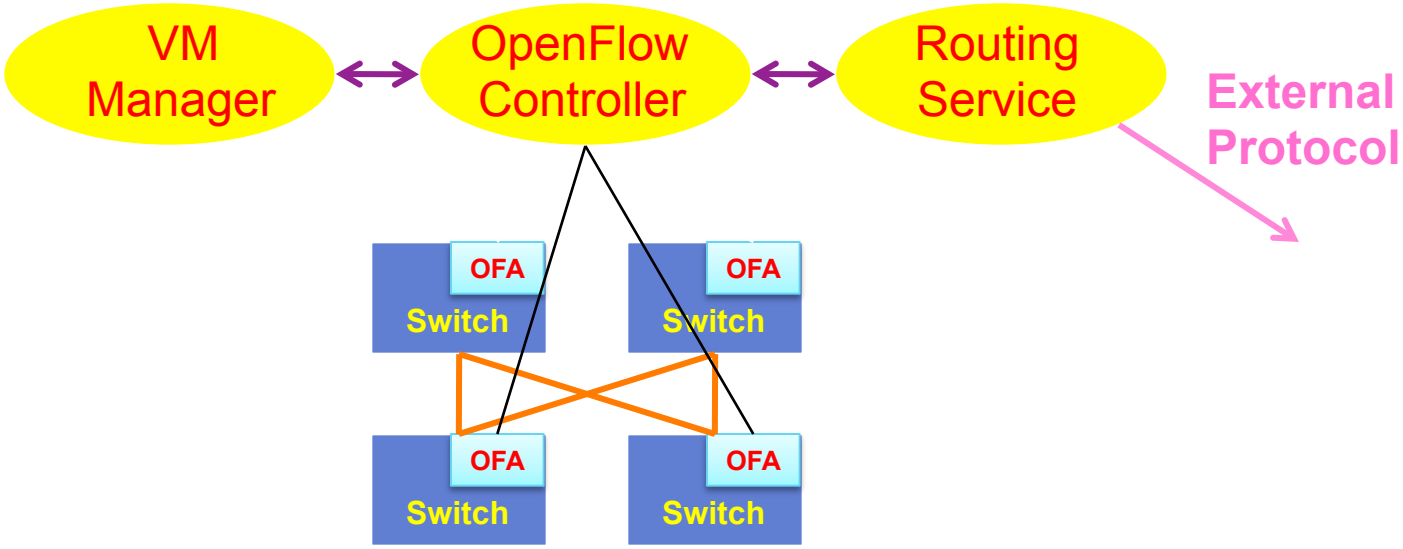
- Switch Architecture [SIGCOMM 08]
- Cabling, Merchant Silicon [Hot Interconnects 09]
- Virtualization, Layer 2, Management [SIGCOMM 09,SOCC11a]
- Routing/Forwarding [NSDI 10]
- Hybrid Optical/Electrical Switch [SIGCOMM 10,SOCC11b]
- Applications [NSDI11, FAST12]
- Low latency communication [NSDI12,ongoing]
- Transport Layer [EuroSys12,ongoing]
- Wireless augment [SIGCOMM12]

Vignette 2: Software Defined Networking Needs Data Management

Network Protocols Past and Future

- Historically, goal of network protocols is to eliminate centralization
  - Every network element should act autonomously, using local information to effect global targets for fault tolerance, performance, policy, security
  - The Internet probably would not have happened without such decentralized control
- Recent trends toward *Software Defined Networking*
  - Deeper understanding of building scalable, fault tolerant logically centralized services
  - Majority of network elements and bandwidth in data centers under the control of a single entity
  - Requirements for virtualization and global policy

Software Defined Networking (SDN)



- Separate control plane from data plane
- Open Network Foundation and OpenFlow protocol leading the charge to enabling SDN
  - OFC → OFA API?

SDN Challenges

- Control plane replication, fault tolerance, scale
- No fate sharing between control and data plane
- Configuration management
  - When new router comes online or topology changes, how to push information to the right places?
- Network management: adaptively drill down to retrieve appropriate network state
- Virtualization and multiple control planes
- All challenges of large-scale distributed databases

- **State of the art: CSV files and perl scripts**

Google’s Software Defined WAN Architecture

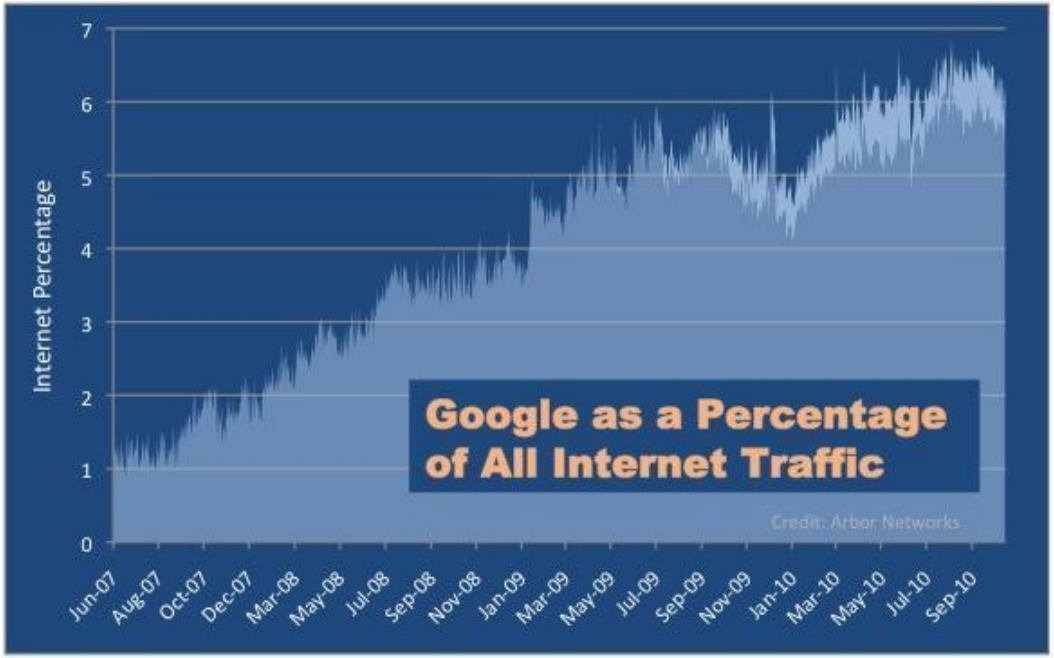
ATLAS 2010 Traffic report

Posted on Monday, October 25th, 2010 | Bookmark on del.icio.us

Google Sets New Internet Traffic Record

by Craig Labovitz

This month, Google broke an equally impressive Internet traffic record — gaining more than 1% of all Internet traffic share since January. If Google were an ISP, as of this month it would rank as the **second largest** carrier on the planet.
Only one global tier1 provider still carries more traffic than Google (and this ISP also provides a large portion of Google's transit).



Cloud Computing Requires Massive Wide-Area Bandwidth

- Low latency access from global audience and highest levels of availability
  - Vast majority of data migrating to cloud
  - Data must be replicated at multiple sites
- WAN unit costs decreasing rapidly
  - But not quickly enough to keep up with even faster increase in WAN bandwidth demand

WAN Cost Components

- Hardware
  - Routers
  - Transport gear
  - Fiber
- Overprovisioning
  - Shortest path routing
  - Slow convergence time
  - Maintain SLAs despite failures
  - No traffic differentiation
- Operational expenses/human costs
  - Box-centric versus fabric-centric views

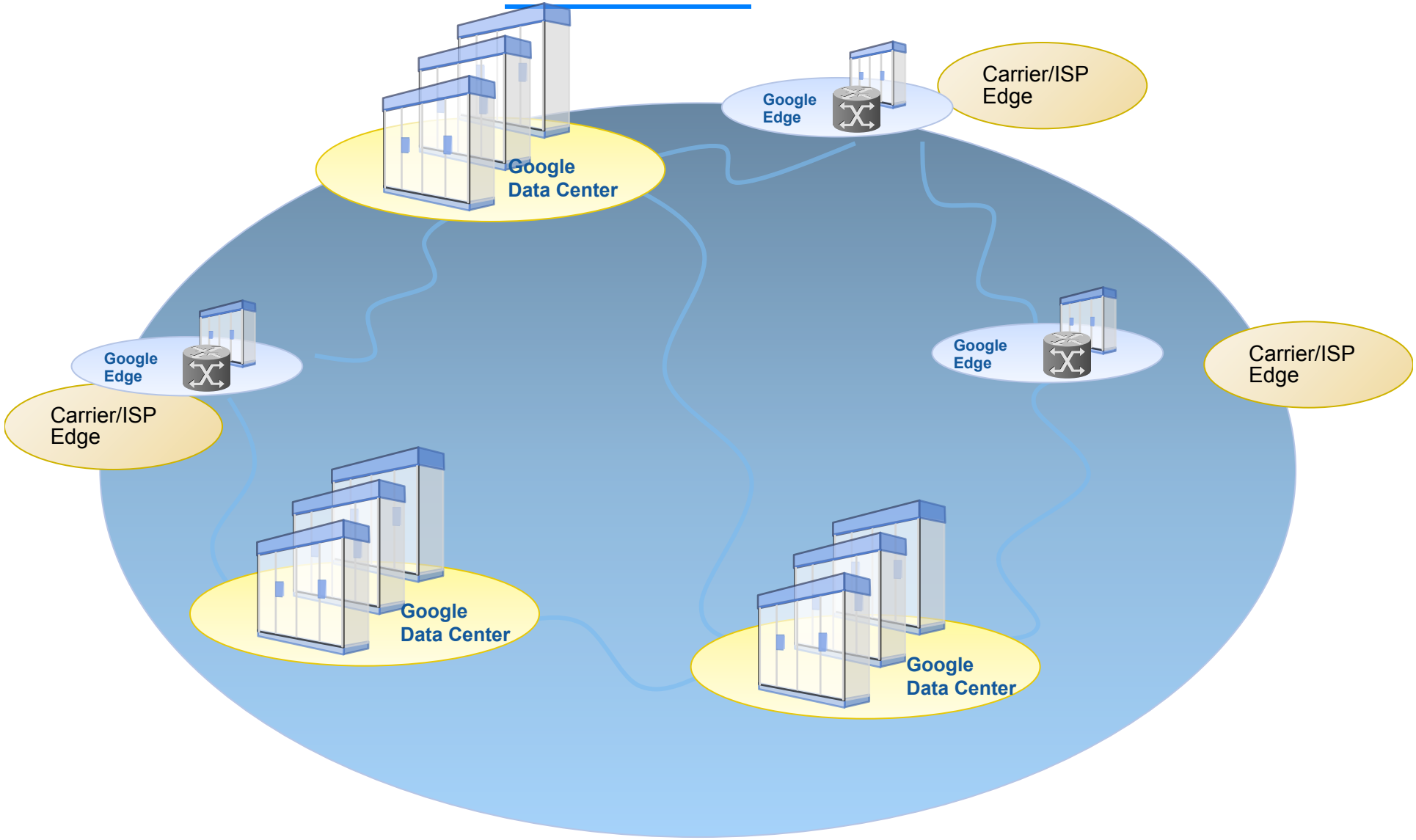
Why Software Defined WAN

Google

- Separate hardware from software
  - Choose hardware based on necessary features
  - Choose software based on protocol requirements
- Logically centralized network control
- Automation: Separate monitoring, management, and operation from individual boxes
- *Flexibility and Innovation*

Result: A WAN that is more efficient, higher performance, more fault tolerant, and cheaper

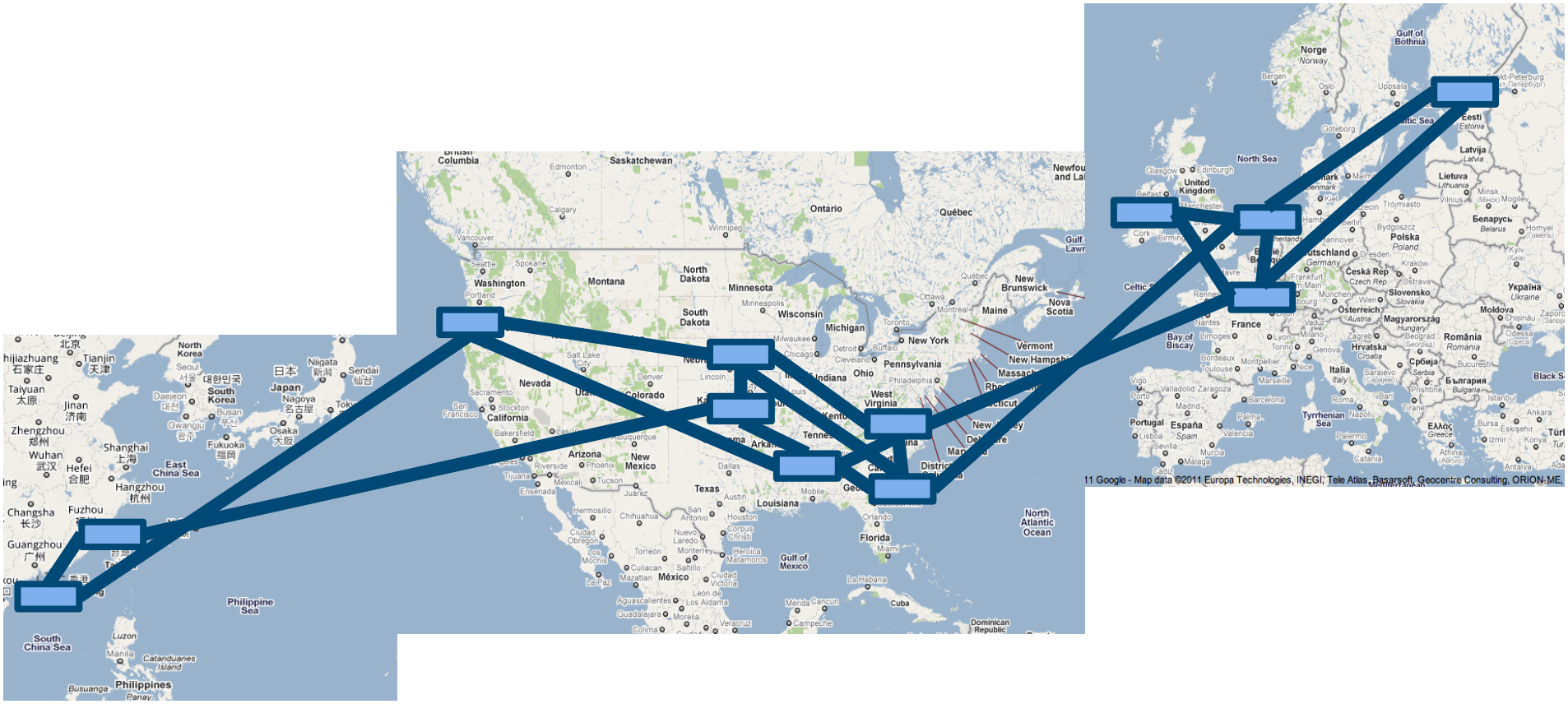
A Warehouse-Scale-Computer (WSC) Network



Google's WAN

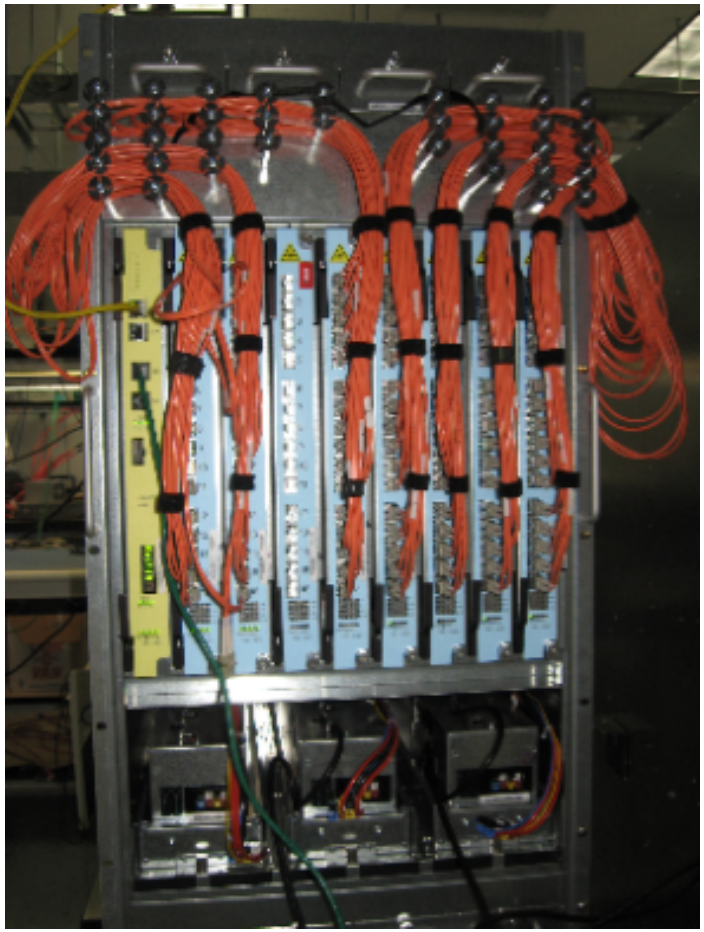
- Two backbones
  - I-Scale: Internet facing (user traffic)
  - G-Scale: Datacenter traffic (internal)
- Widely varying requirements: loss sensitivity, topology, availability, etc.
- Widely varying traffic characteristics: smooth/diurnal vs. bursty/bulk

Google's Software Defined WAN



G-Scale Network Hardware

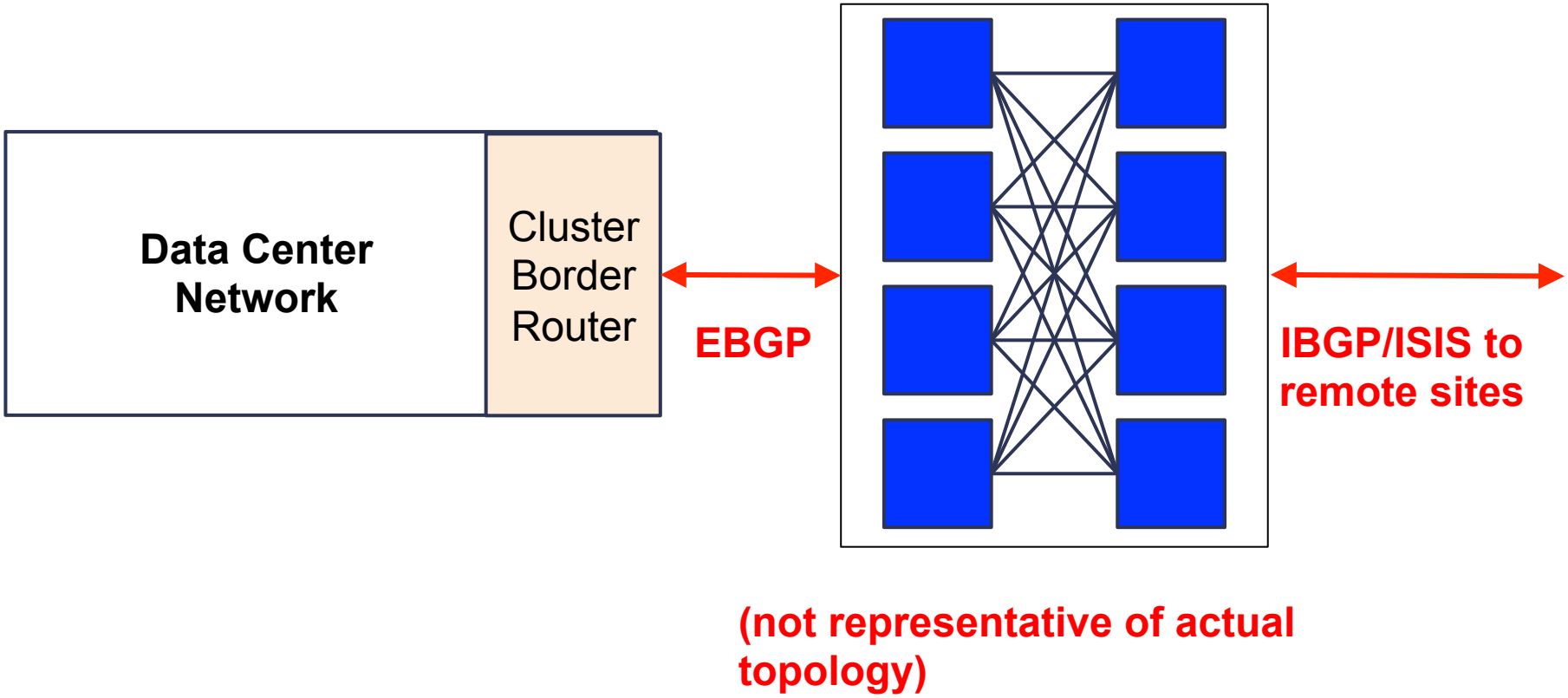
- Built from merchant silicon
  - 100s of ports of nonblocking 10GE
- OpenFlow support
- Open source routing stacks for BGP, ISIS
- Does not have all features
  - No support for AppleTalk...
- Multiple chassis per site
  - Fault tolerance
  - Scale to multiple Tbps



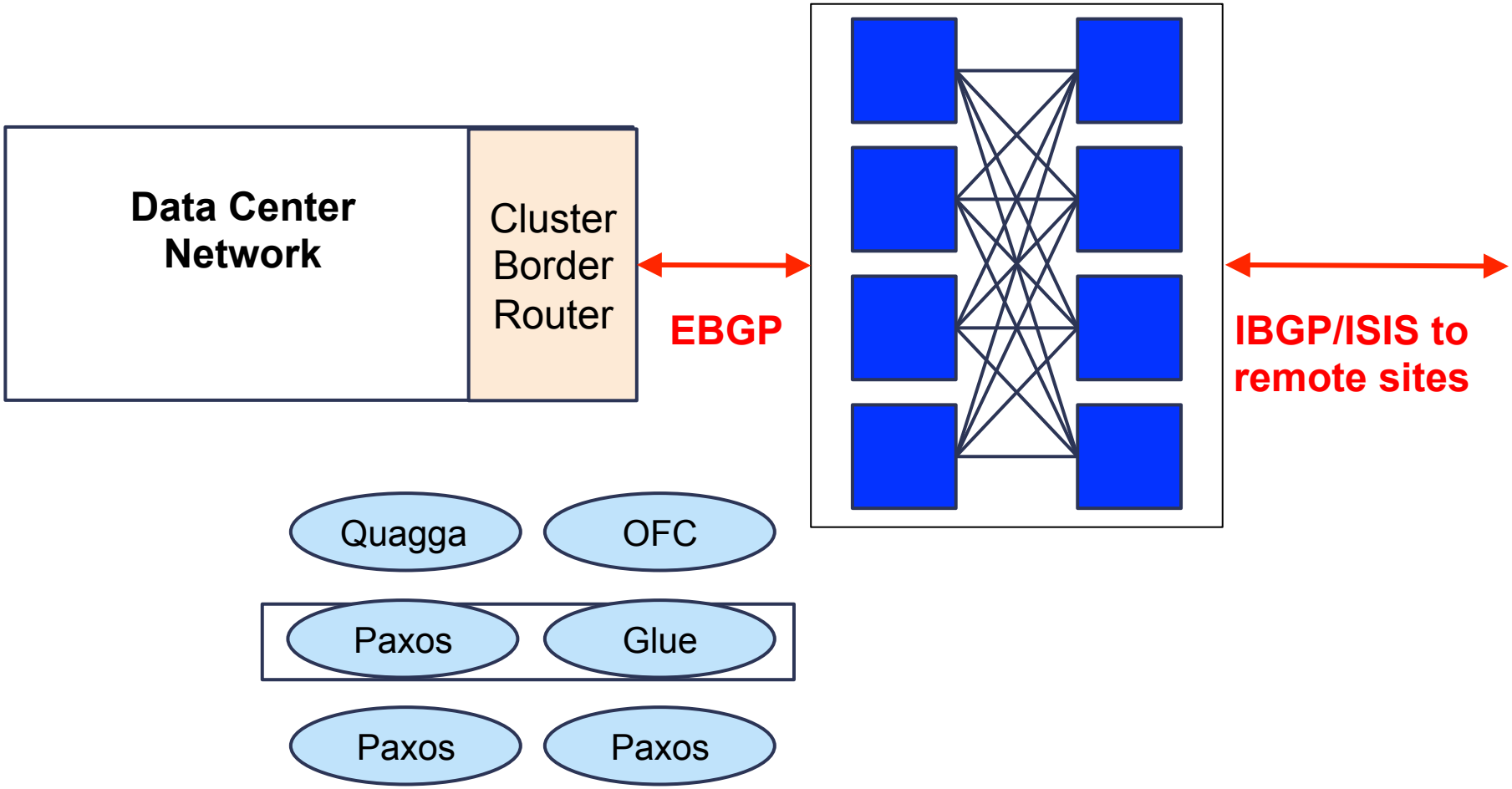
G-Scale WAN Deployment

- Multiple switch chassis in each domain
  - Custom hardware running Linux
- Quagga BGP stack, ISIS/IBGP for internal connectivity

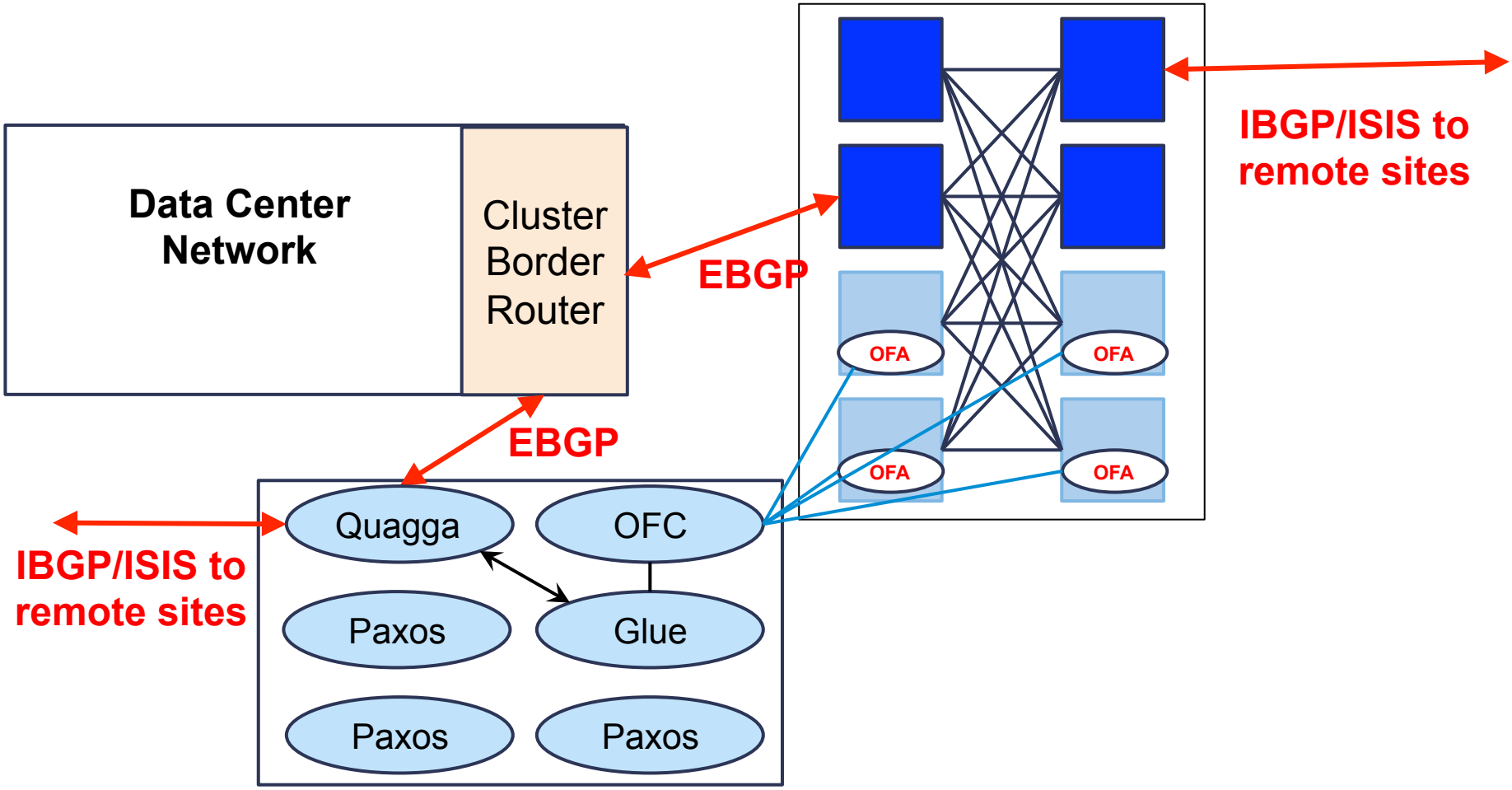
Mixed SDN Deployment



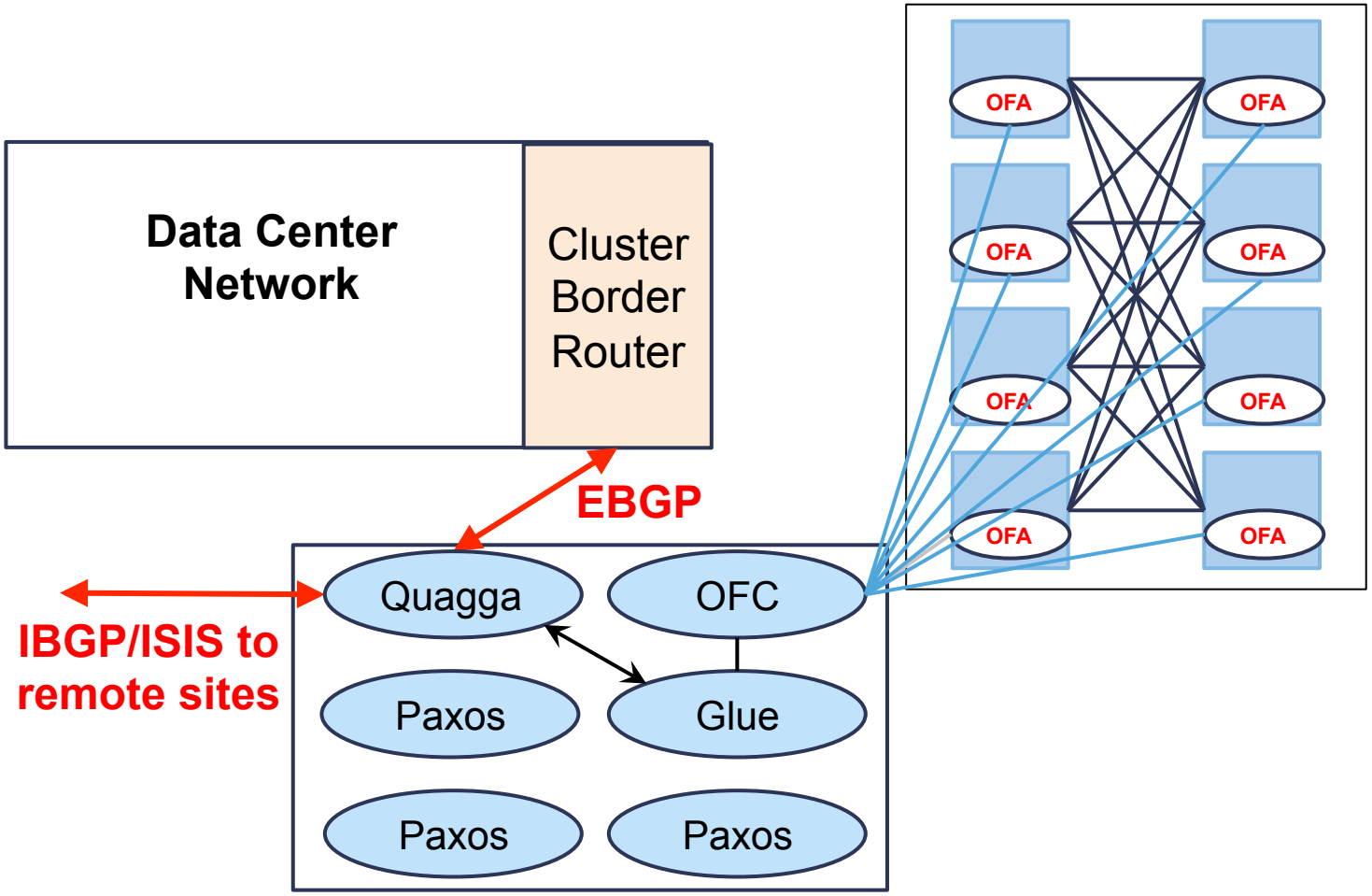
Mixed SDN Deployment



Mixed SDN Deployment

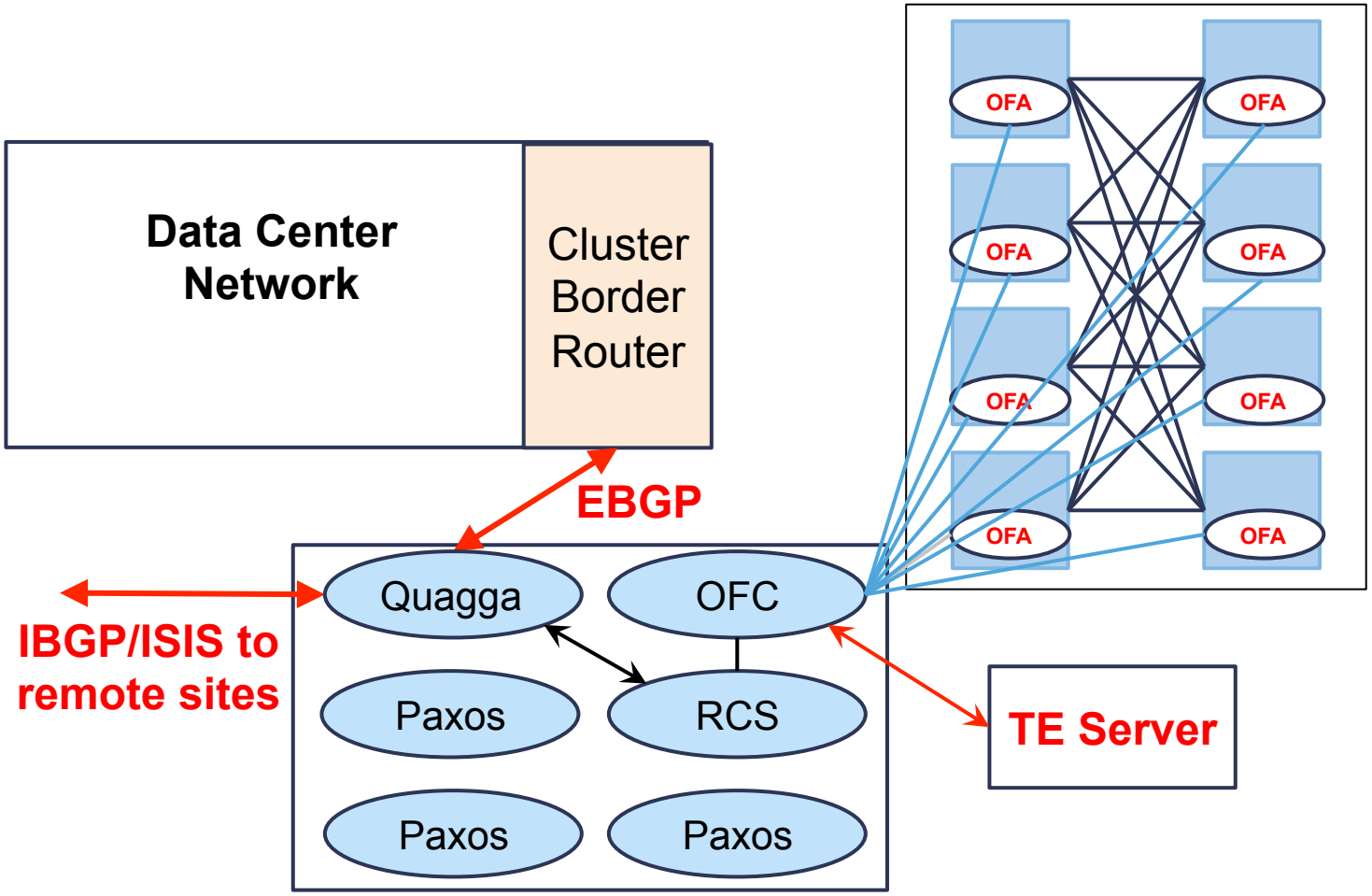


Mixed SDN Deployment



- SDN site delivers full interoperability with legacy sites

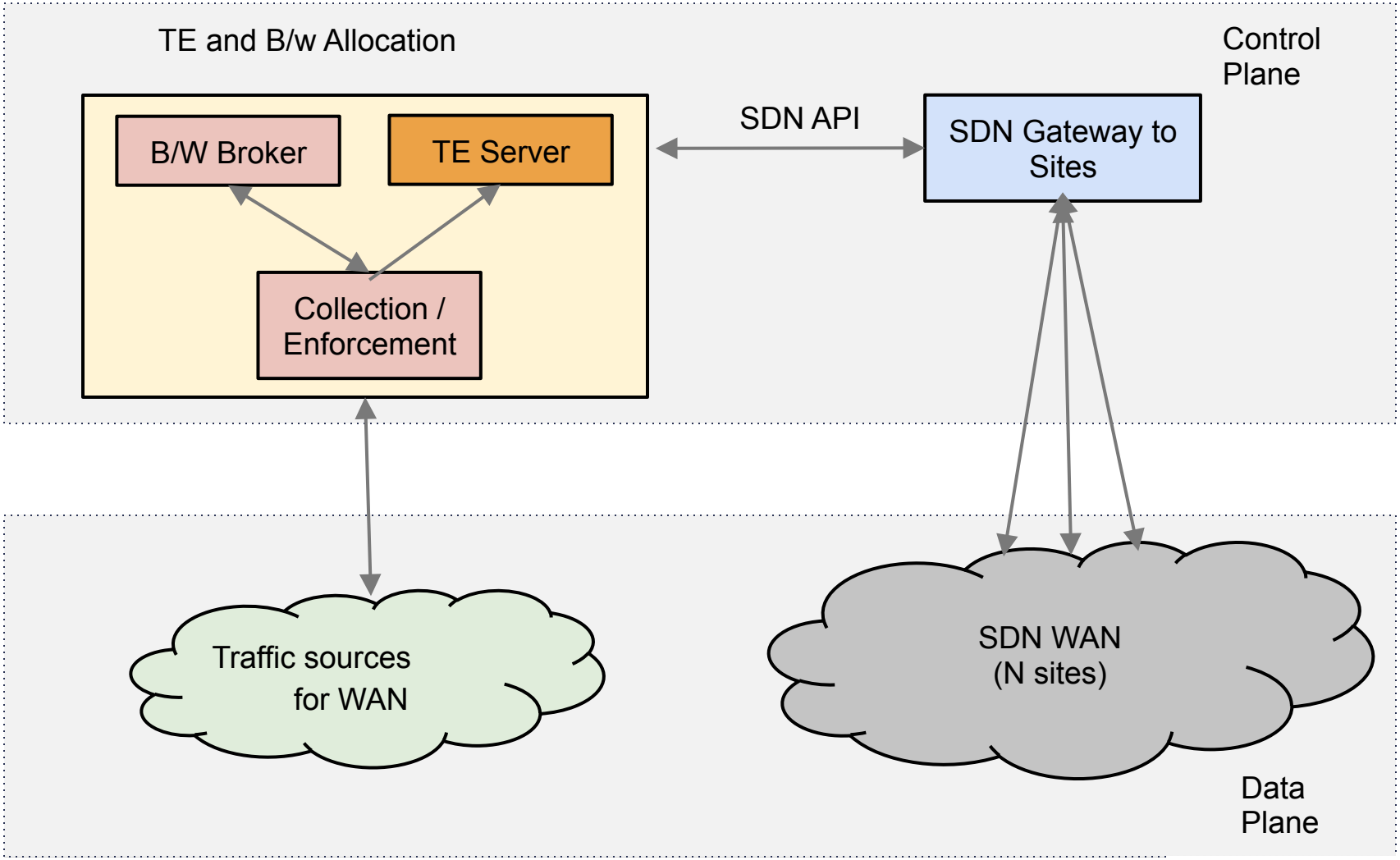
Mixed SDN Deployment



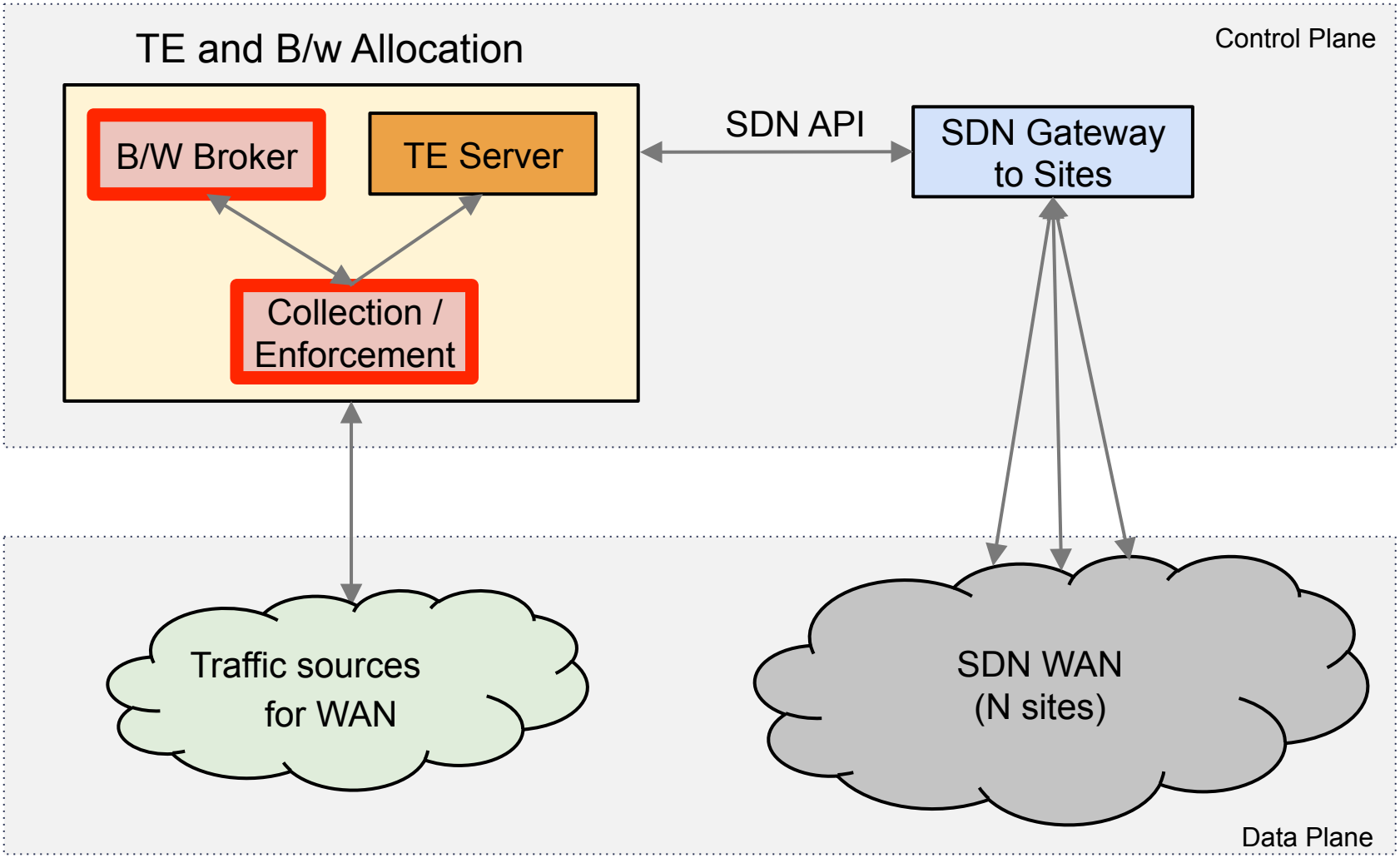
- Ready to introduce new functionality, e.g., TE

Bandwidth Broker and Traffic Engineering

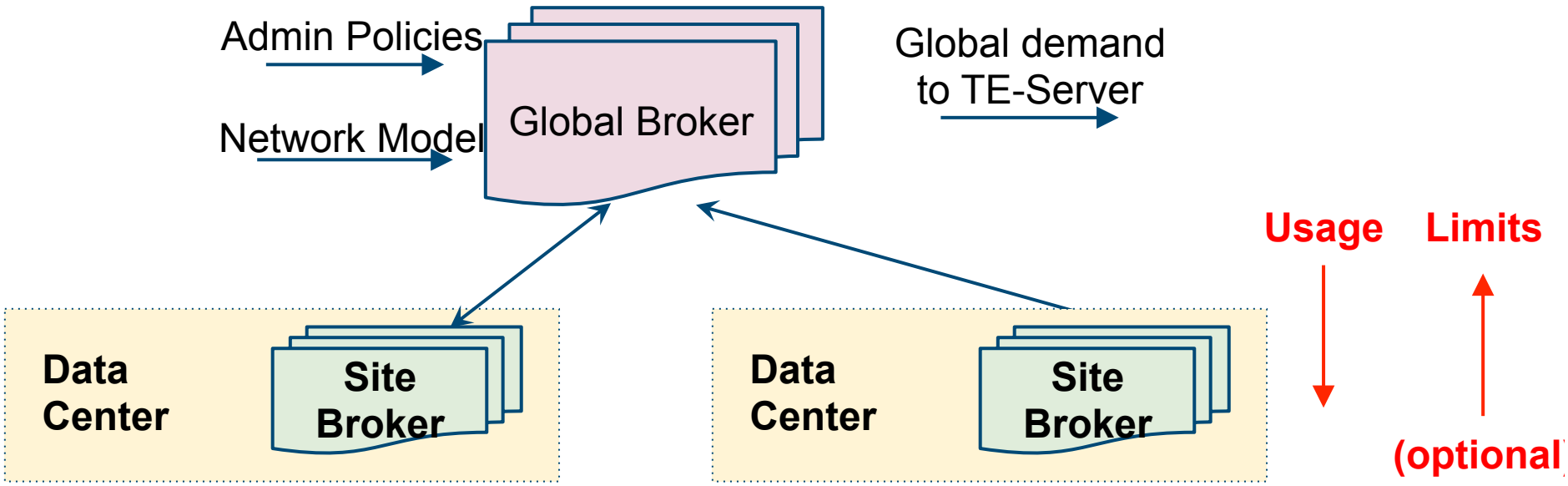
High Level Architecture



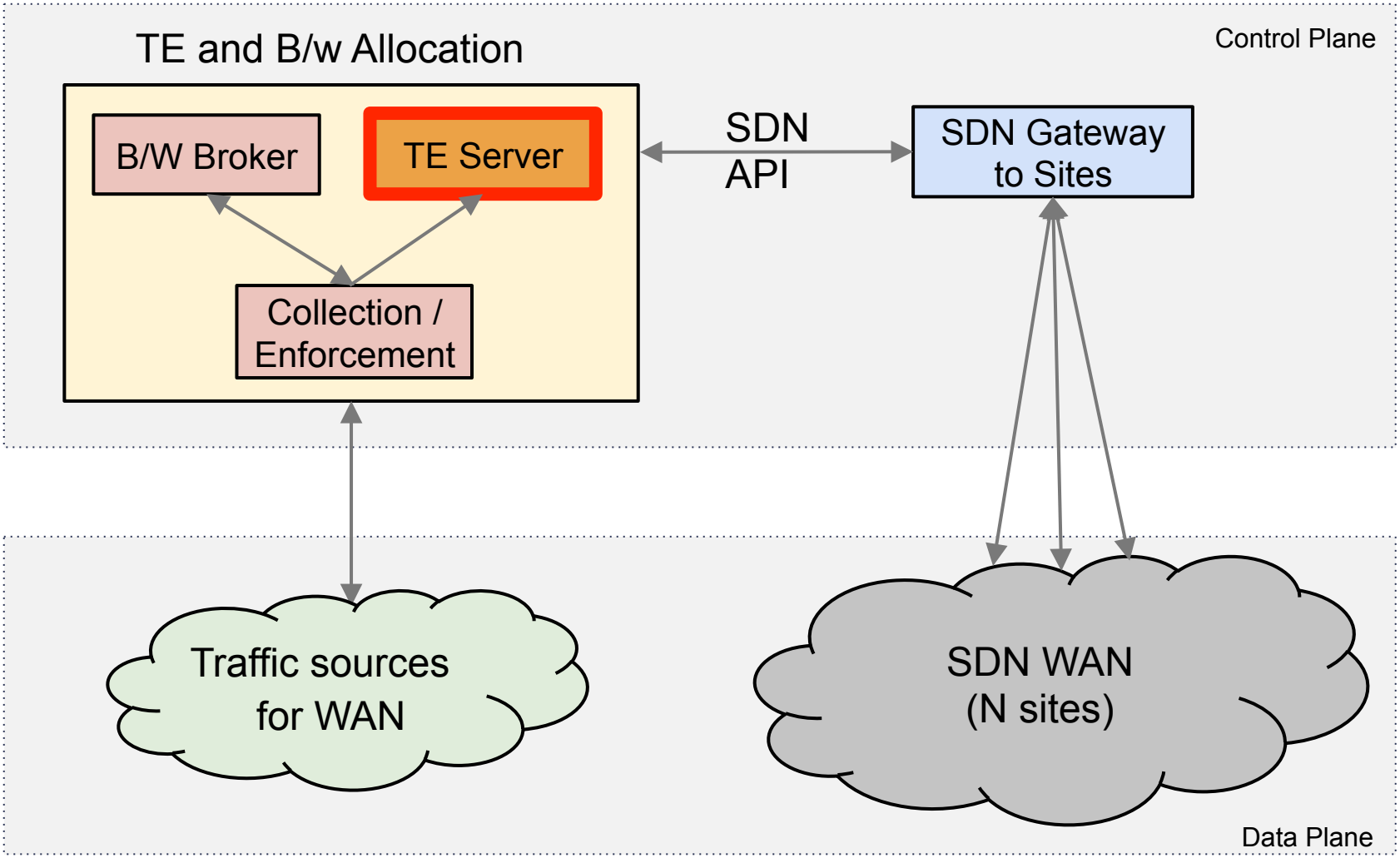
High Level Architecture



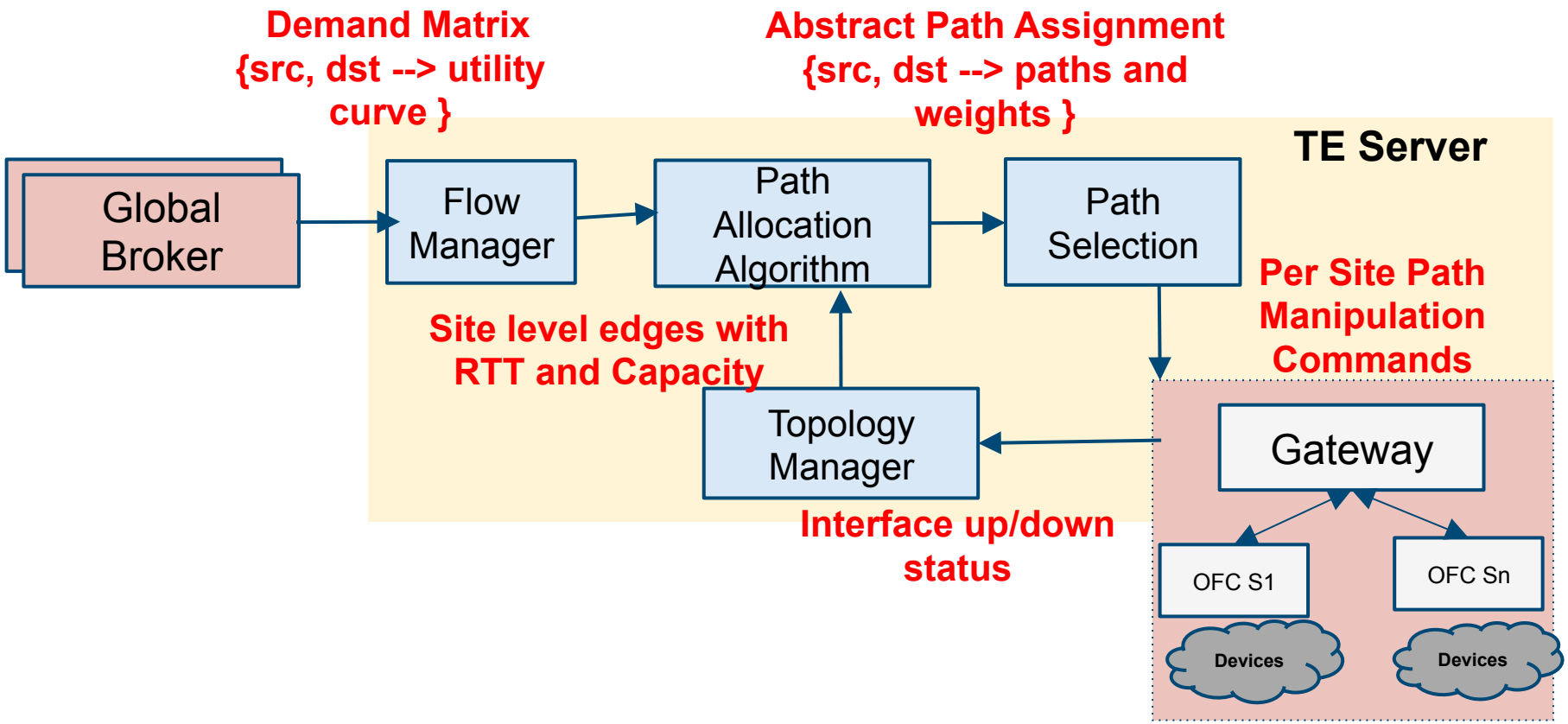
Bandwidth Broker Architecture



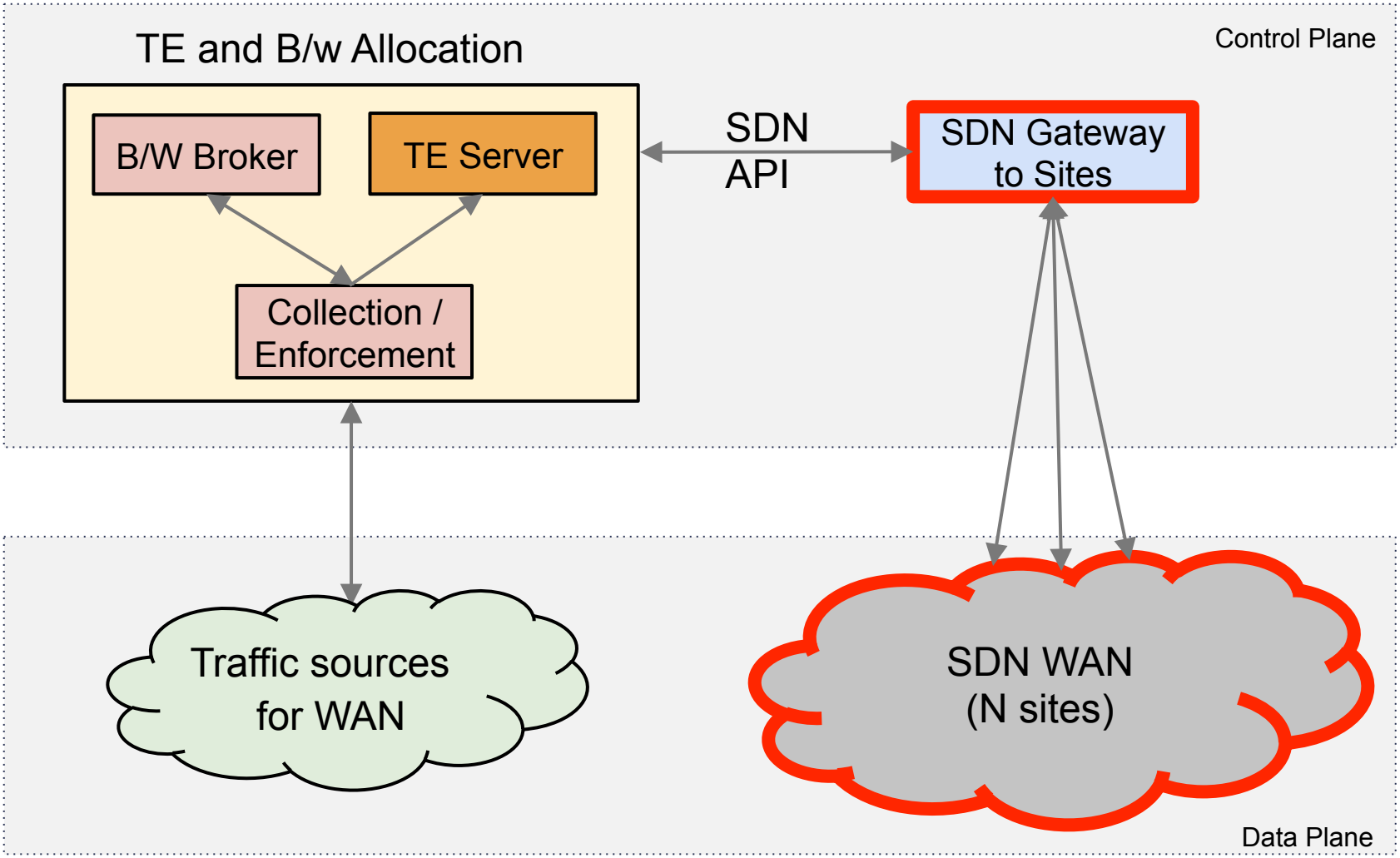
High Level Architecture



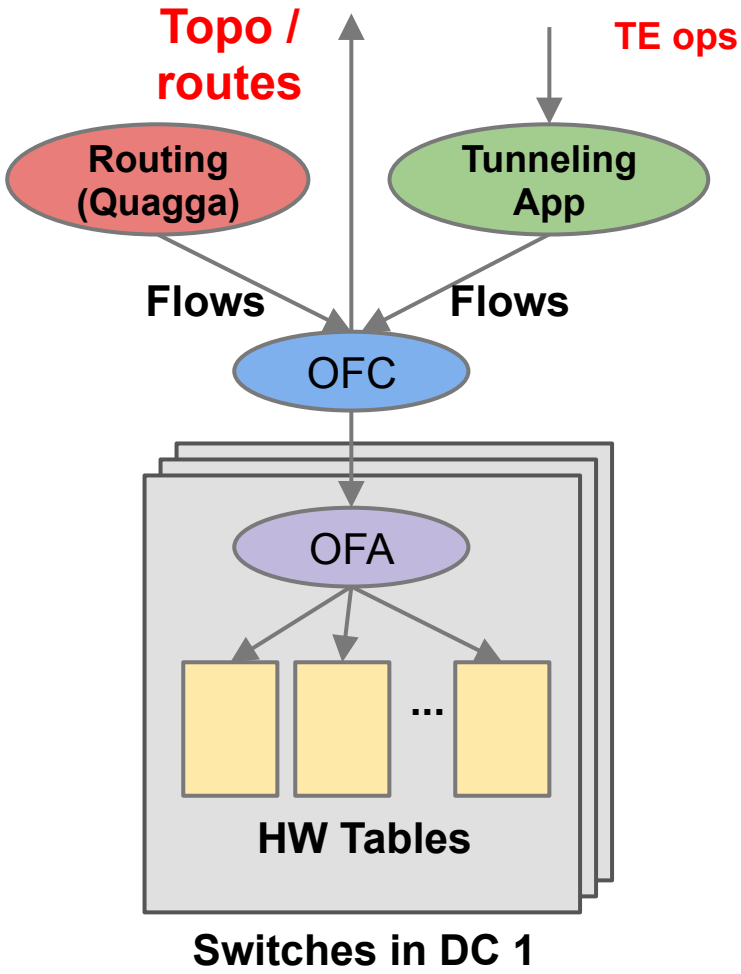
TE Server Architecture



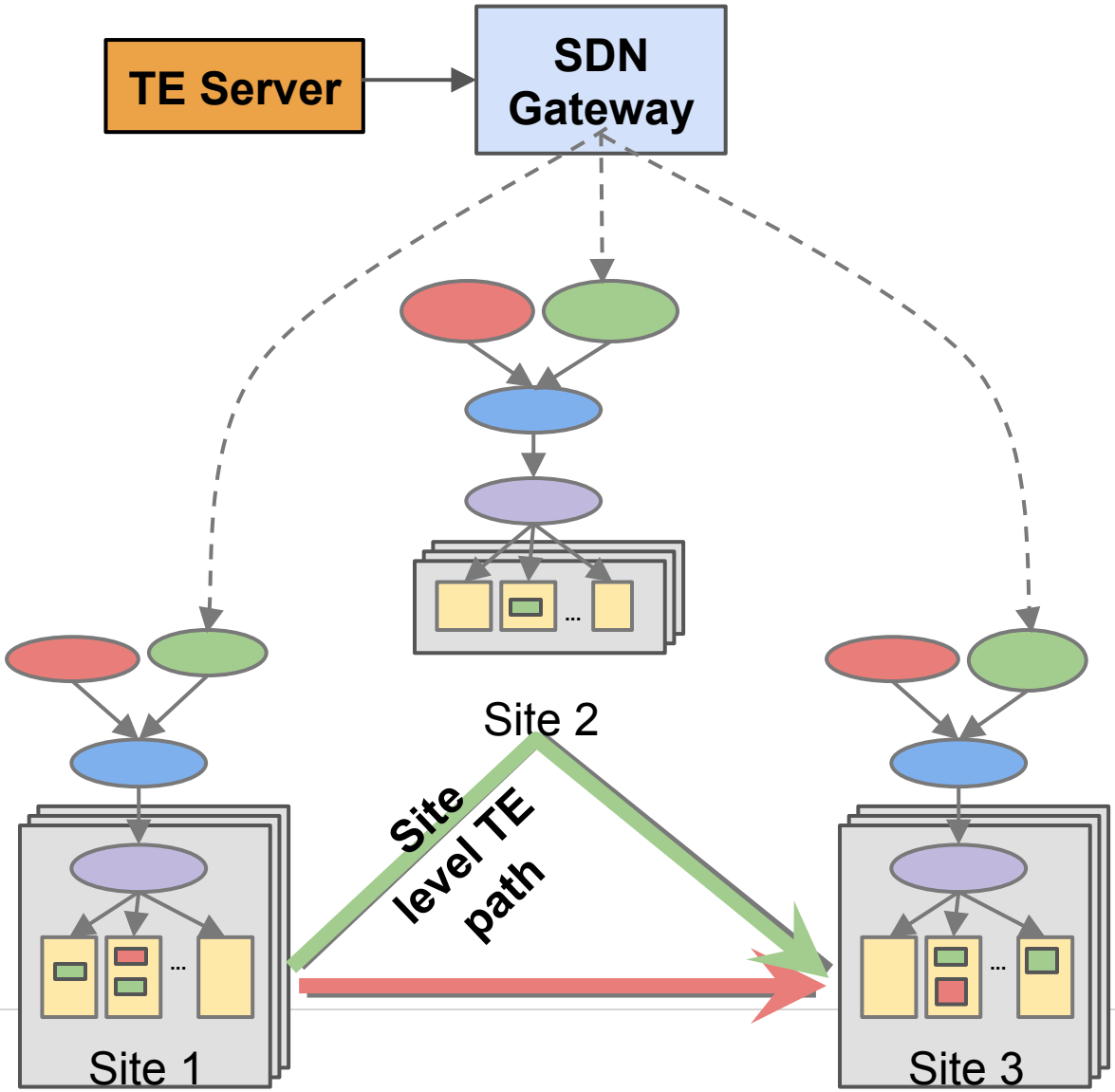
High Level Architecture



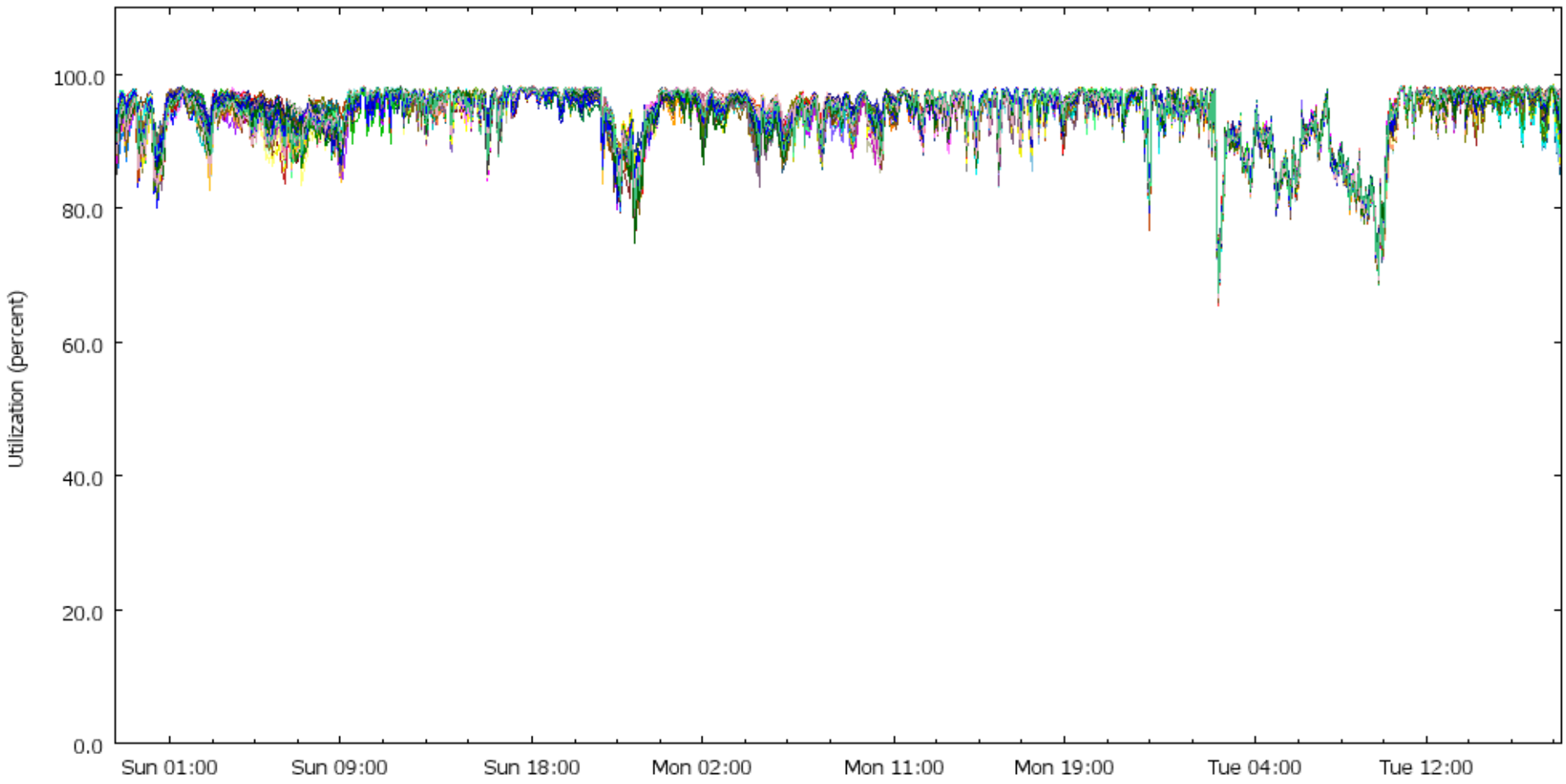
Controller Architecture



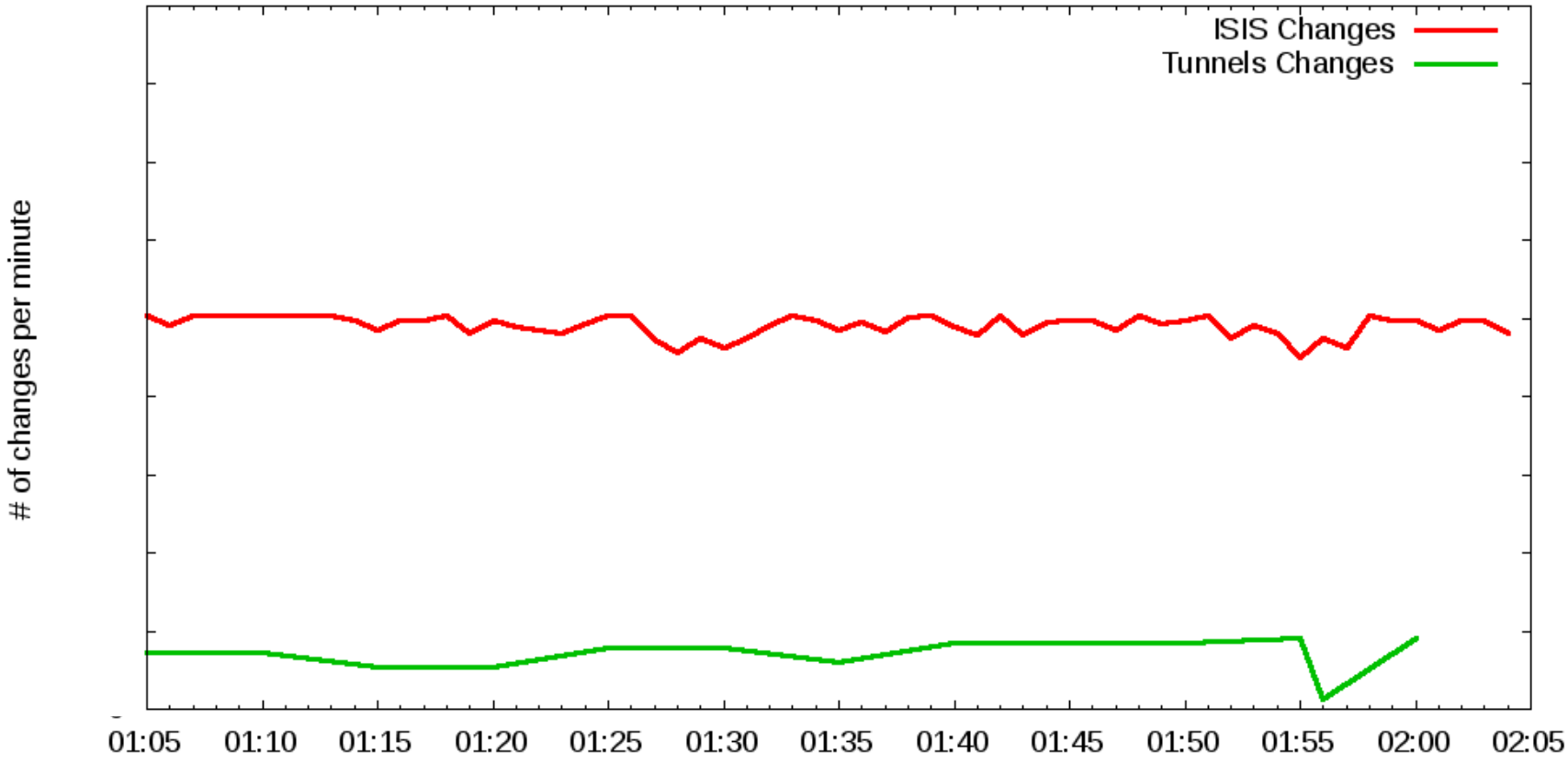
Controller Architecture



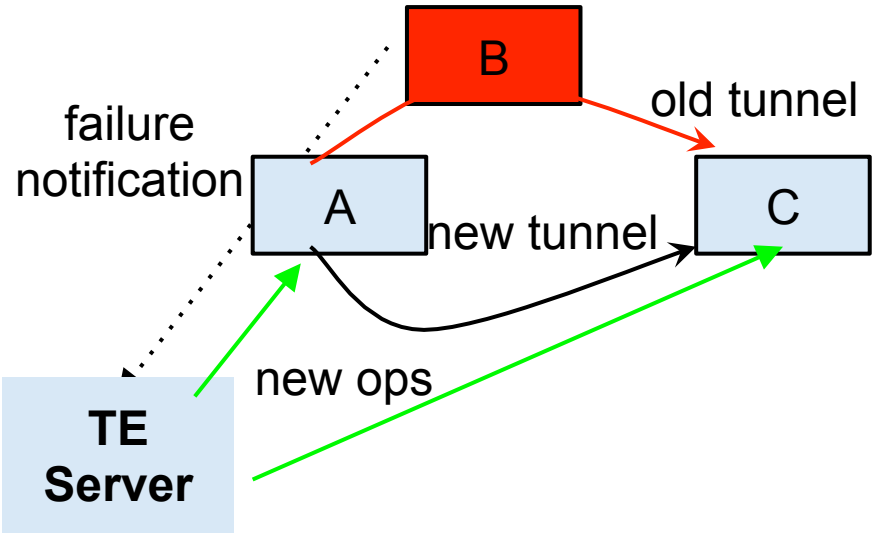
Sample Utilization



Benefits of Aggregation



Convergence under Failures

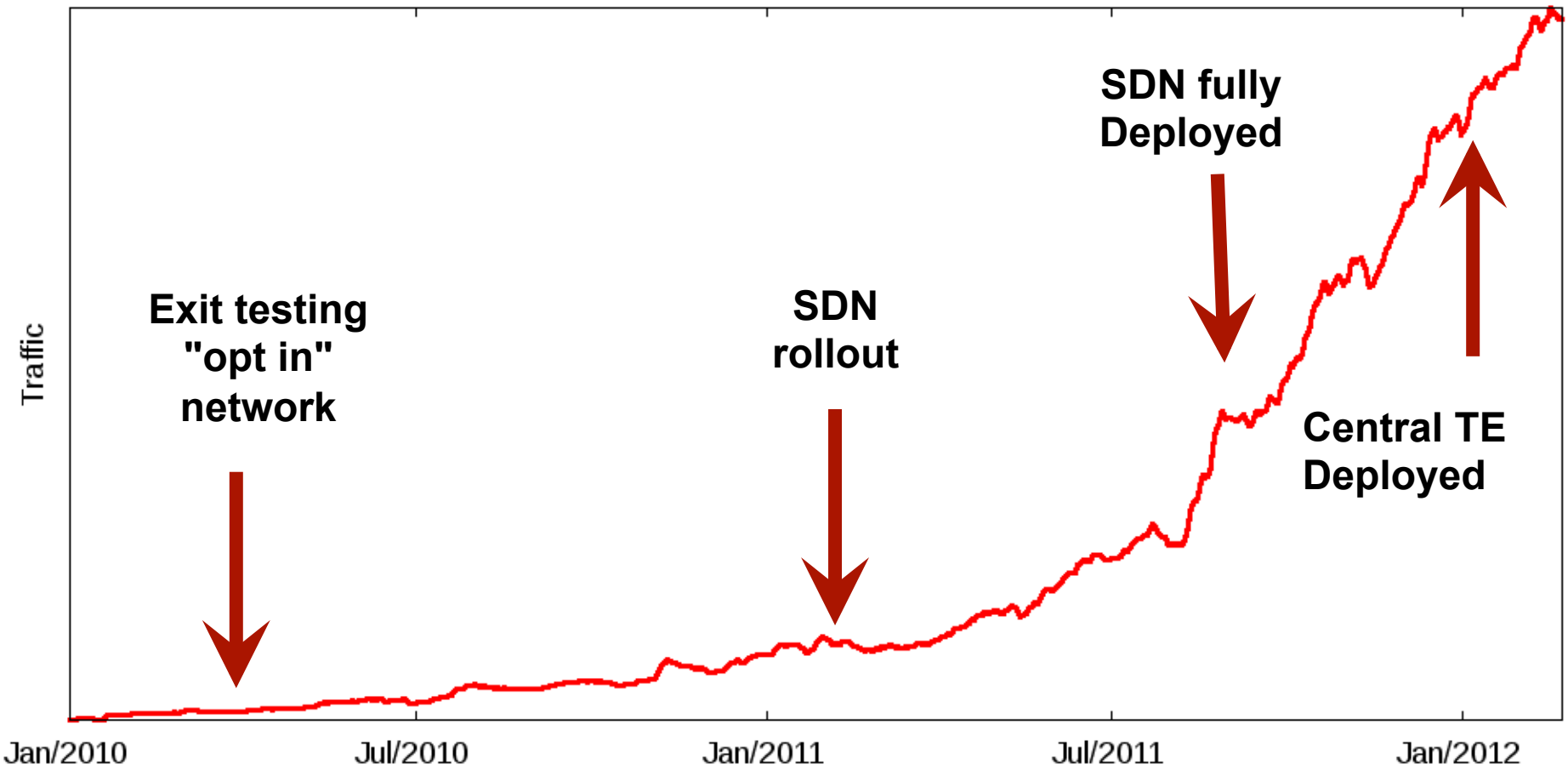


no-TE: traffic drop ~ 9 sec
with-TE: traffic drop ~ 1 sec

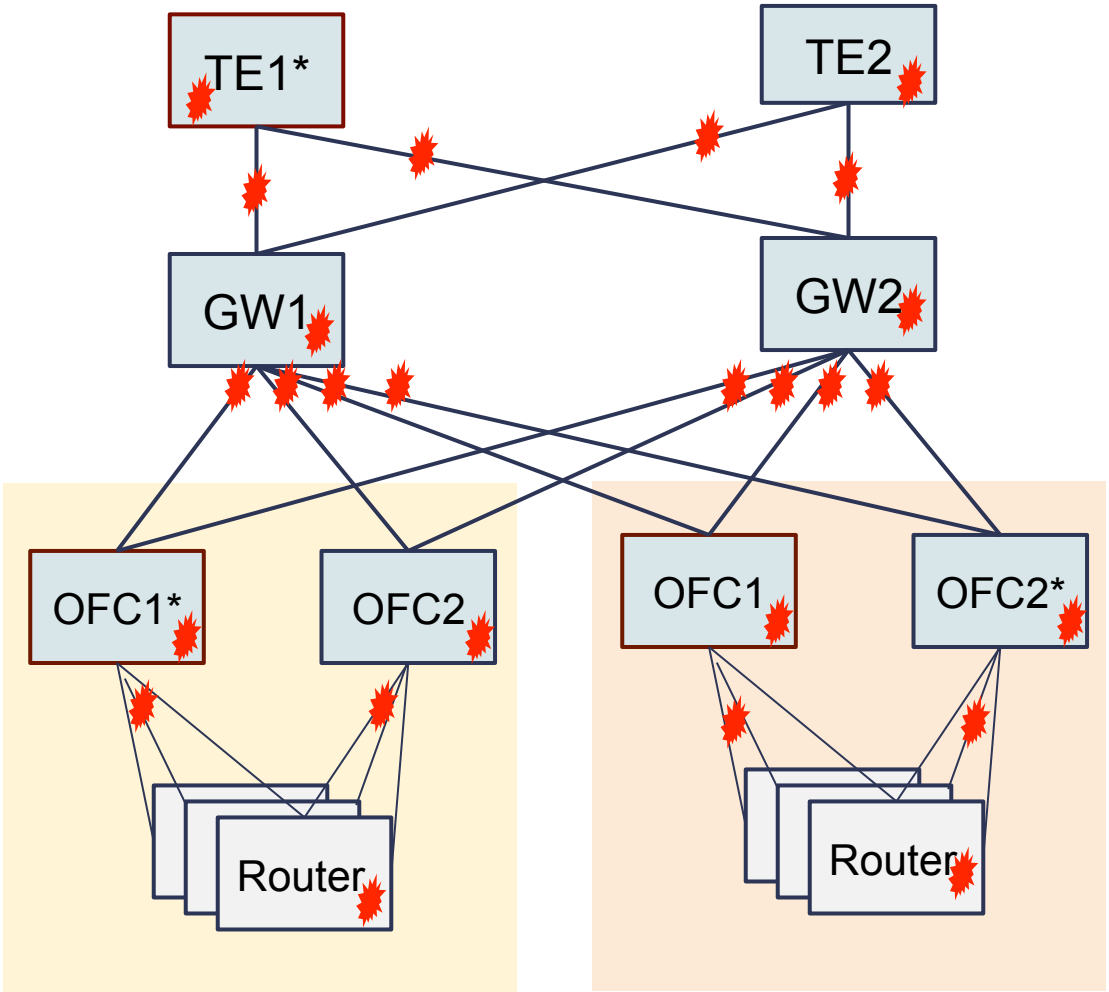
Without TE: Failure detection and convergence is slower:

- Delay 'inside' TE << timers for detecting and communicating failures (in ISIS)
- Fast failover may be milliseconds, but not guaranteed to be either accurate or "good"

G-Scale WAN History



Range of Failure Scenarios



Potential failure condition

* indicates mastership

Trust but Verify: Consistency Checks

Google

TE View	OFC View	Is Valid	Comment
Clean	Clean	yes	Normal operation.
Clean	Dirty	no	OFC remains dirty forever
Clean	Missing	no	OFC will forever miss entry
Dirty	Dirty	yes	Both think Op failed
Dirty	Clean	yes	Op succeeded but response not yet received by TE
Dirty	Missing	yes	Op issued but not received by OFC
Missing	Clean	no	OFC has extra entry, and will remain like that
Missing	Dirty	no	(same as above)

Implications for ISPs

- Dramatically reduce the cost of WAN deployment
  - Cheaper per bps in both CapEx and OpEx
  - Less overprovisioning for same SLAs
- Differentiator for end customers
  - Less cost for same BW or more BW for same cost
- Possible to deploy incrementally in pre-existing network
  - Deployment experience with Google's global SDN production WAN suggests SDN is real and it works
  - But it’s just the beginning

Conclusions

- Large-scale data processing needs scale out networking
  - Unlocking the potential of modern server hardware for at scale problems requires orders of magnitude improvement in network performance
- Scale out networking requires large-scale data management
  - Experience with Google's SDN WAN suggests that logically centralized state management critical for cost-effective deployment and management
  - Still in the stone ages in dynamically managing state and getting updates to the right places in the network

Thank you!