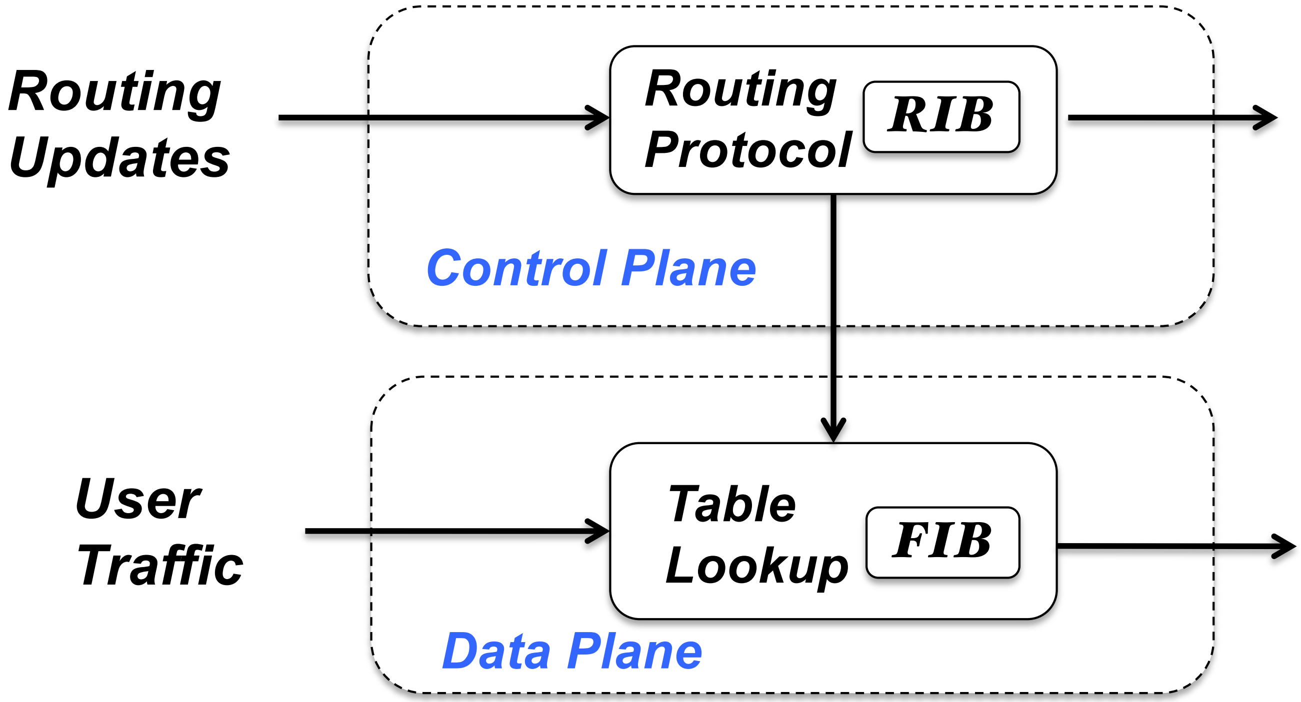
CSC 525: Computer Networks

Software Defined Network (SDN)

- A framework that provides new abstractions of network operations and services.
- It started from early 2000's at Stanford
  - Actually your class projects contained the same idea, and were initially developed by the same people.
- It has now gathered lots of interests from both academia and industry
  - Implementation in hardware and software
  - Products are shipping, startups are growing, and with real-world deployment.

Data Plane and Control Plane



- Data plane: process and forward user traffic.
- Control plane: run routing protocols and other control protocols, compute routing table.

Today's networks

- Network devices are specialized appliances
  - both hardware (data-plane) and software (control plane, management, services) in one box.
  - Command-line interface (CLI)
  - Closed source
- Network management, services, and control are implemented as a distributed system.
  - E.g., routing protocol running on each individual routers
- Traffic demands, desired features, and number of devices are all growing fast.

The Problems

- Very hard to add new features, even small, local changes.
  - Need to go through IETF standardization and vendor implementation and product release.
  - Customization is limited to whatever CLI provides.
- Very hard to configure, manage and reason about network behaviors.
  - Due to its distributed nature and large scale.

Separate control and data planes

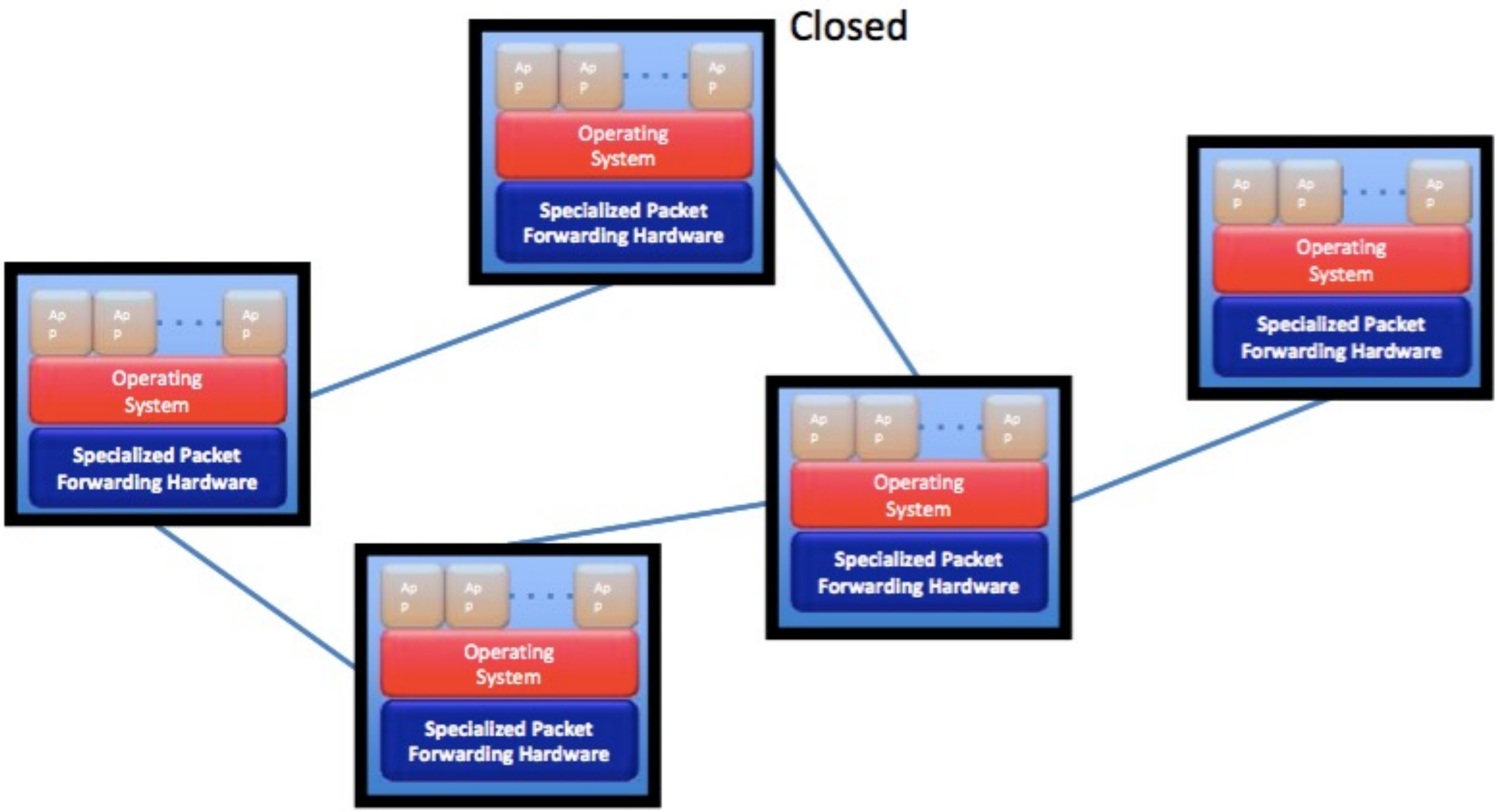
- Answer to the first problem: make network devices programmable by physically separating control and data planes
  - A network device only has the data plane and supports a set of APIs for control plane.
  - All the network features (e.g., routing) are done in software separated from the physical devices, and use the APIs to communicate with the devices.
  - Service providers can implement their own control logics, new features, etc. in software, and use the APIs to export the results to the devices.

Centralized control

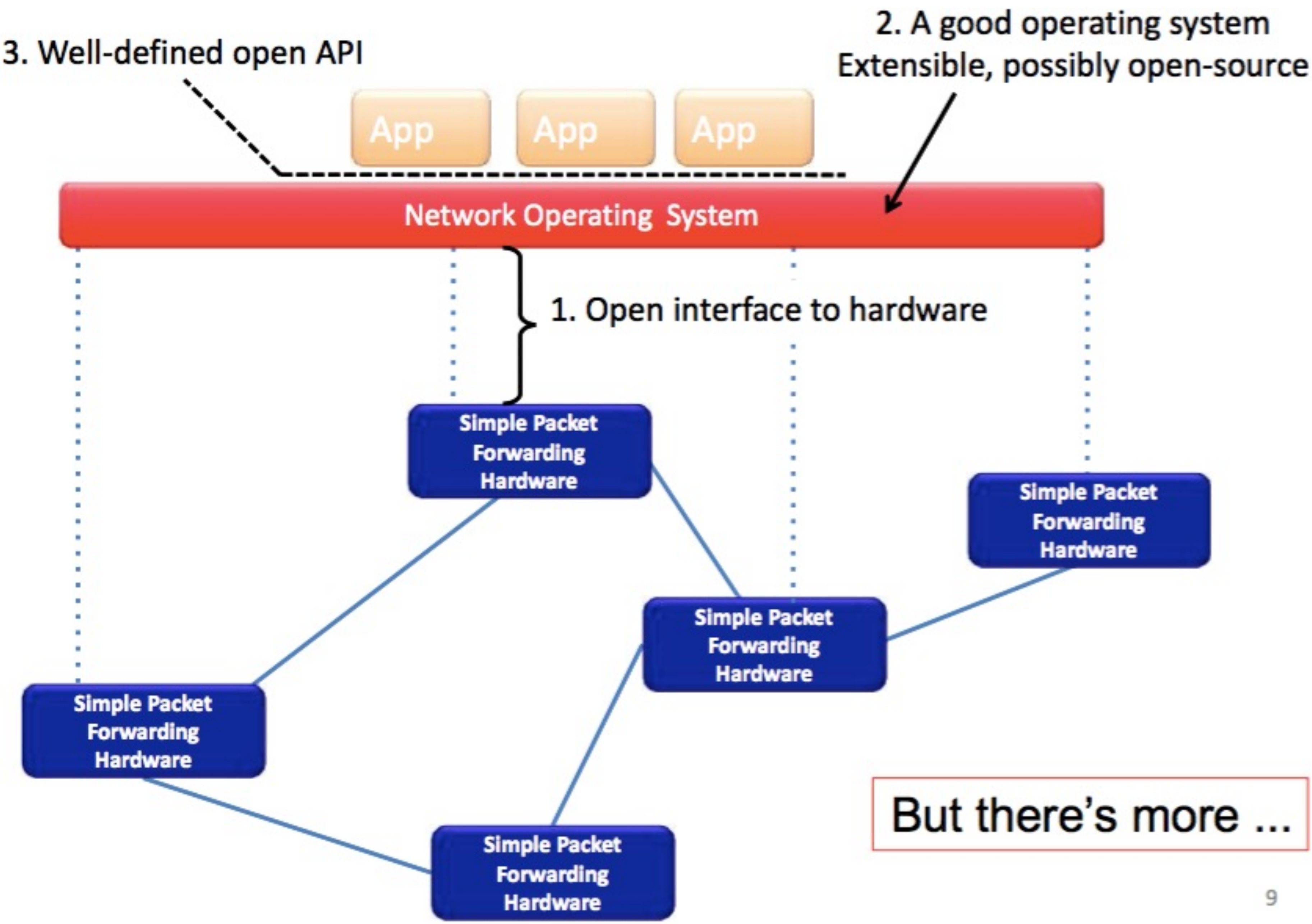
- Answer to the second problem: implement control logics over a global view of the network.
  - Use some mechanisms to collect information from all devices to form a complete view of the network.
  - Control algorithms are implemented over this complete view.
    - Much easier than using distributed protocols between devices.
  - The results are disseminated to devices.

Today's Network

- Closed Boxes, Fully Distributed Protocols

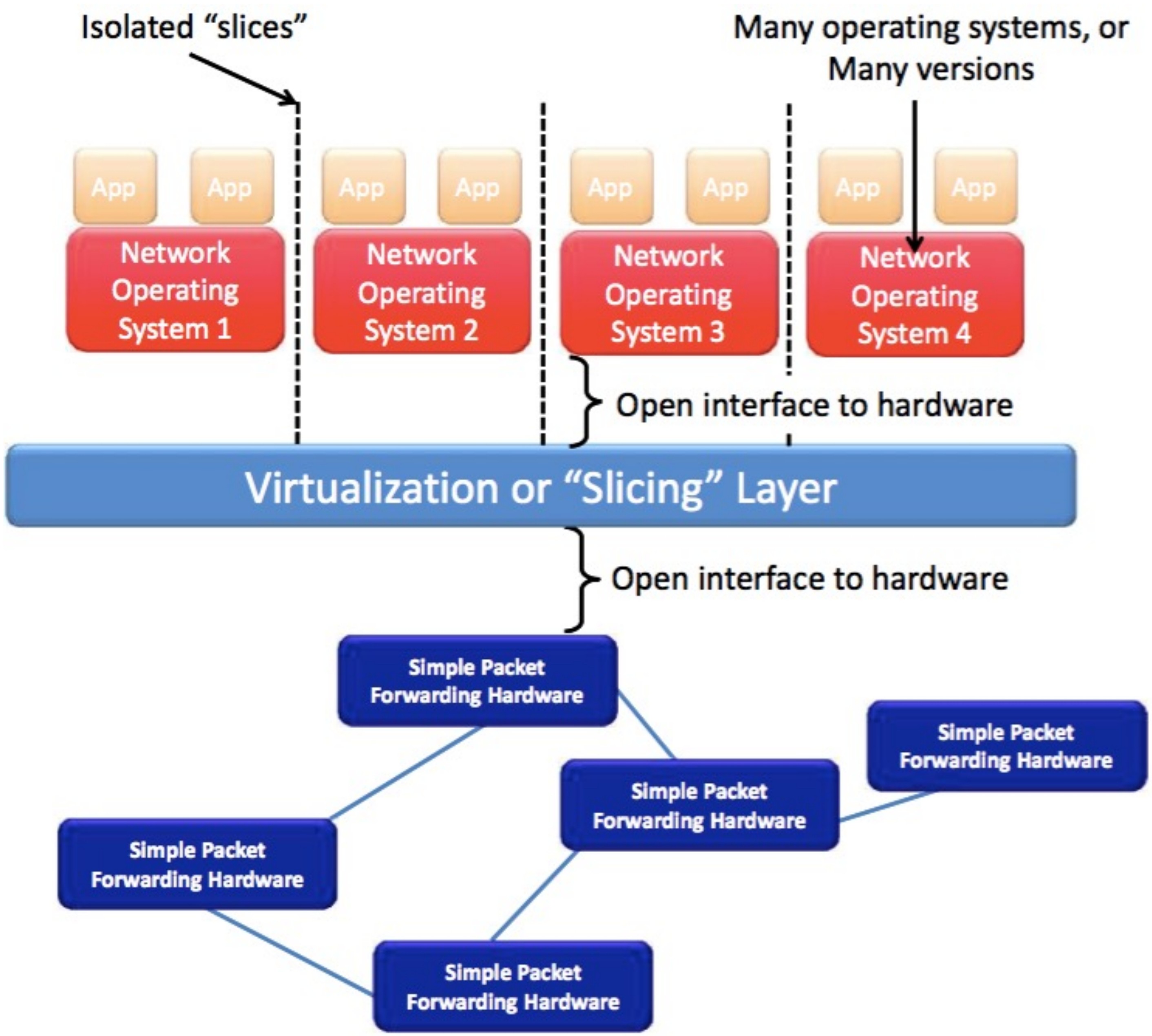


Software Defined Network



But there's more ...

Virtualized SDN



OpenFlow

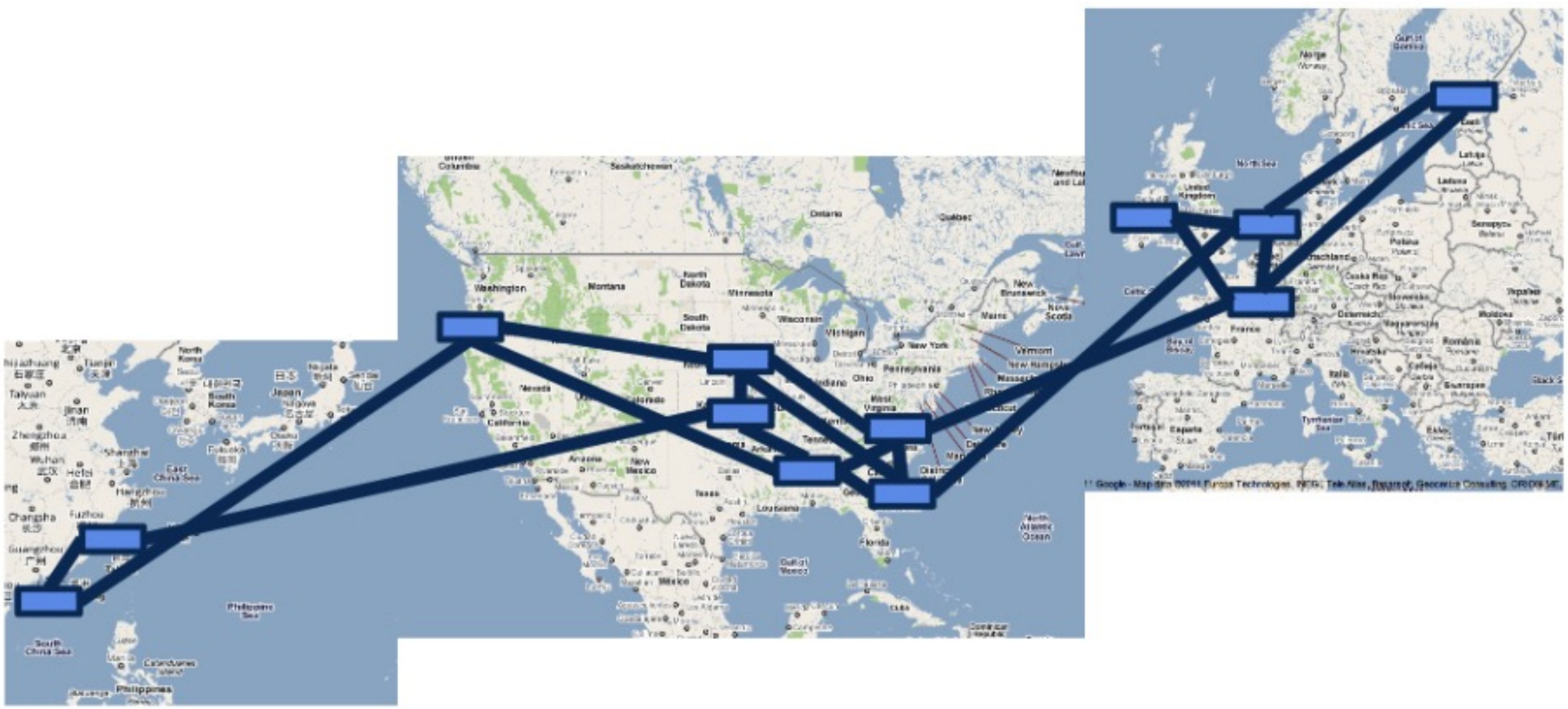
- One way to realize the SDN vision.
  - Define the interface, the so-called ”southbound” interface, between Network OS and forwarding hardware.
  - The central OpenFlow Controller (OFC)
  - Each forwarding element runs an OpenFlow Agent (OFA), which implements the southbound interface.
  - Define the basic operation.
  - Available in hardware and software.

OpenFlow Operations

- Each device maintains one or multiple "flow tables"
  - Each entry has header fields, counter, and action.
- Upon packet arrival, find a matching entry in the flow table by comparing header fields.
  - If found, take the "action" and update counter.
  - If not, queue the packet, send its header to the Controller, which will do some route computation then send the new result to the device. The device installs new rules in flow table and continue forwarding packets.
  - The Controller and Agents communicate using SSL/TCP.

B4: Google's SDN-based WAN

- An SDN solution to traffic engineering (TE)
  - Demonstrate the feasibility of SDN networks
  - Demonstrate the benefits of centralized TE enabled by SDN.



Motivation for Backend Backbone

Google

WAN Intensive Apps

YouTube Web Search
Google+ Maps AppEngine
Photos and Hangouts
Android/Chrome Updates

Data centers deployed across the world

- Serve content with geographic locality
- Replicate content for fault tolerance

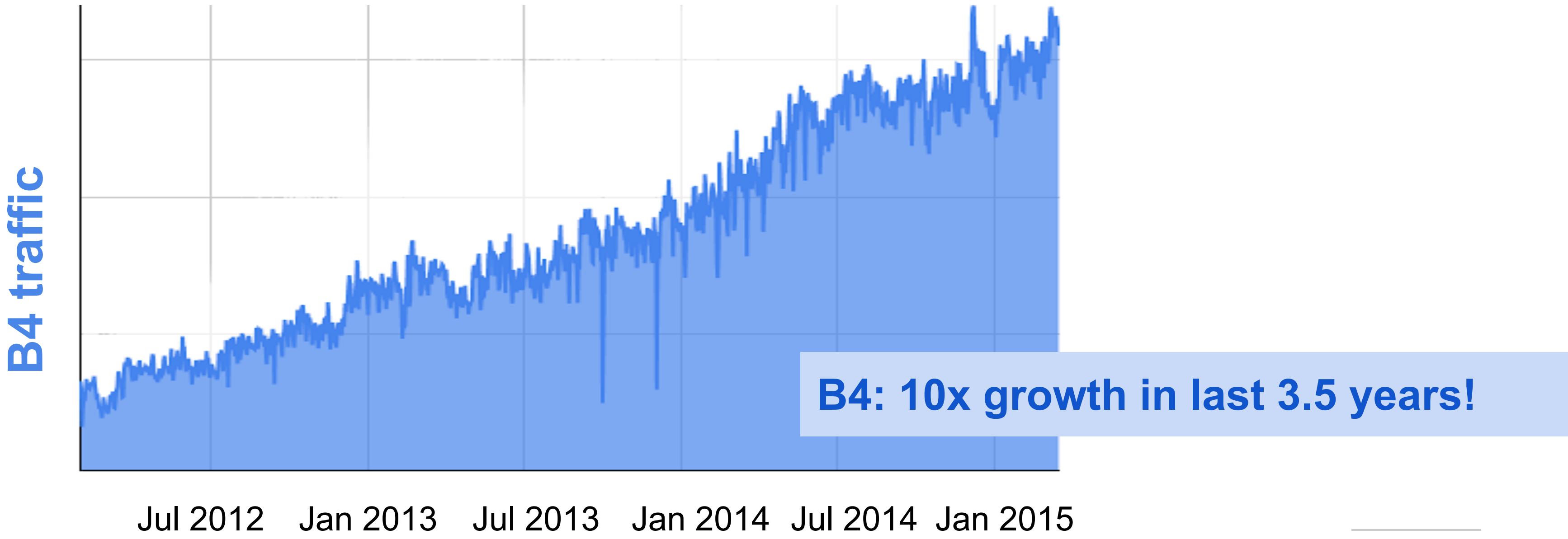
Need a network to connect these data centers to one another

- Not on the public Internet
- Cost effective network for high volume traffic
- Application specific variable in SLO
- Bursty/bulk traffic (not smooth/diurnal)

Two Backbones

Two separate backbones:

- B2: Carries Internet facing traffic → Growing faster than the Internet
- B4: Inter-datacenter traffic → More traffic than B2, growing faster than B2



Growth vs Cost

Does cost per bit/sec go down with additional scale?

- Consider analogies with compute or storage

Networking *cost/bit doesn't naturally decrease with size*

- Quadratic complexity in pairwise interactions and broadcast overhead of all-to-all communication requires more expensive equipment
- Manual management and configuration of individual elements
- Complexity of automated configuration to deal with non-standard vendor configuration APIs

SDN to Solve It

- Faster innovation: separate smarts out of embedded devices
  - Leverage powerful compute in Google servers
  - Faster feature roll-outs on controllers
  - Less frequent switch firmware upgrade
  - Easier hardware upgrade/replacement
- Efficient network management
  - Manage fabric, rather than collection of devices
- Cost effective: opportunity for centralized Traffic Engineering (TE)
  - Higher overall throughput, via better utilization of deployed hardware
    - Need not overprovision
  - Leverage multi-objective multi-commodity flow optimization algorithms
    - More optimal throughput and faster convergence ….

Topics for Today

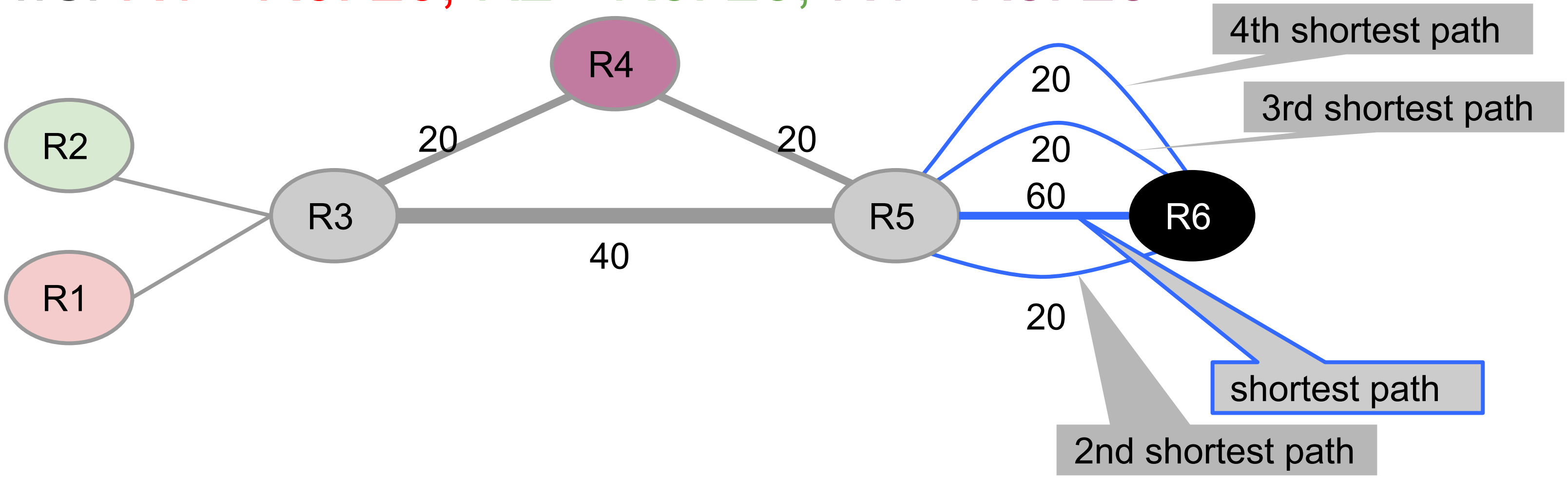
- Background for Traffic Engineering (TE)
- B4-SDN/TE Architecture with OpenFlow protocol
- Benefits of B4-SDN/TE
- Lessons learnt on SDN in three key areas

Performance	Availability	Scale
Fast producer/slow consumer: flow control to the rescue	Robust control plane connectivity and stable mastership is critical	SDN is natural fit for abstraction and hierarchy

Background for Centralized Traffic Engineering

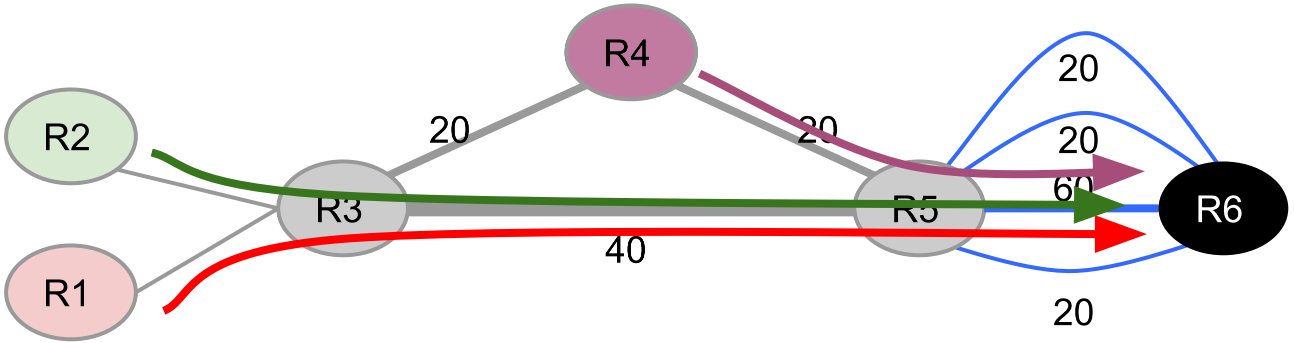
Convergence After Failure

- Flows: R1->R6: 20; R2->R6: 20; R4->R6: 20



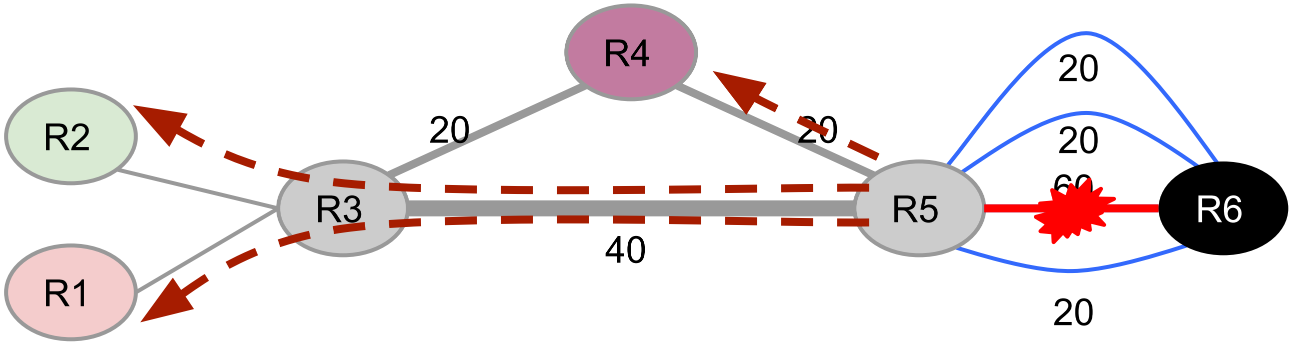
Convergence After Failure

- Flows: R1->R6: 20; R2->R6: 20; R4->R6: 20



Convergence After Failure

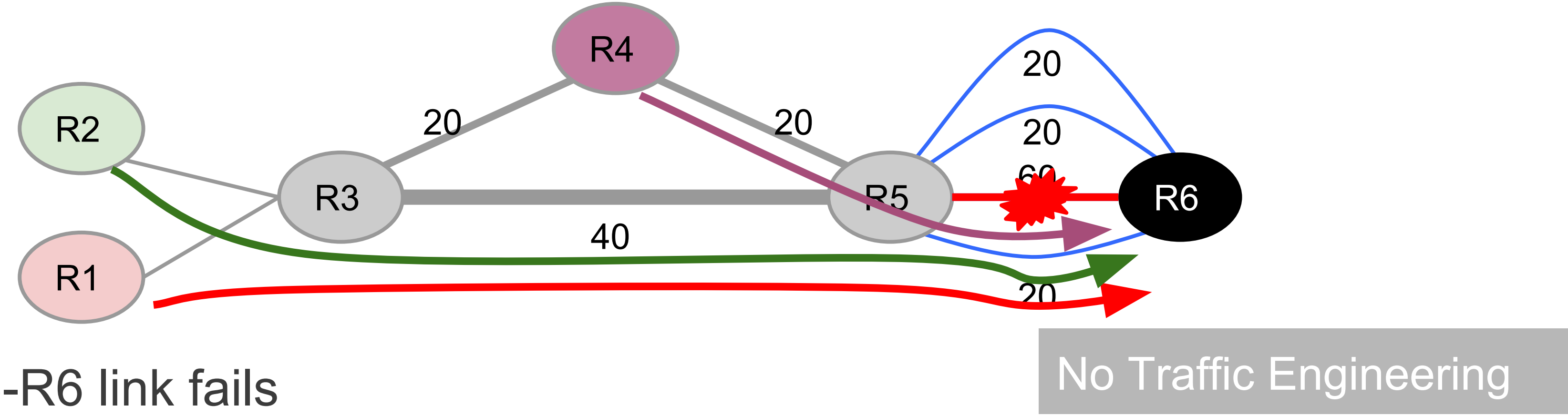
- Flows: R1->R6: 20; R2->R6: 20; R4->R6: 20



- R5-R6 link fails
  - R1, R2, R4 *autonomously* find next best path

Convergence After Failure

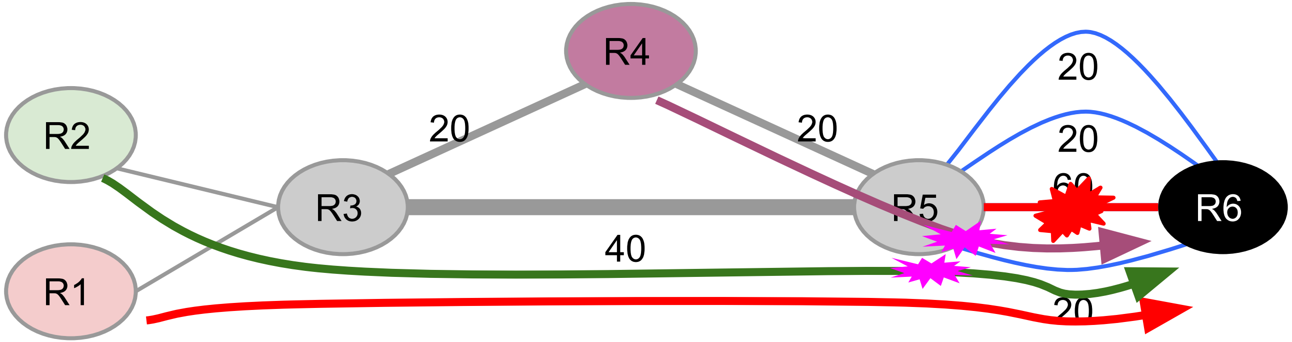
- Flows: R1->R6: 20; R2->R6: 20; R4->R6: 20



- R5-R6 link fails
  - R1, R2, R4 *autonomously* try for next best path
  - R1, R2, R4 push **20** altogether

Convergence After Failure

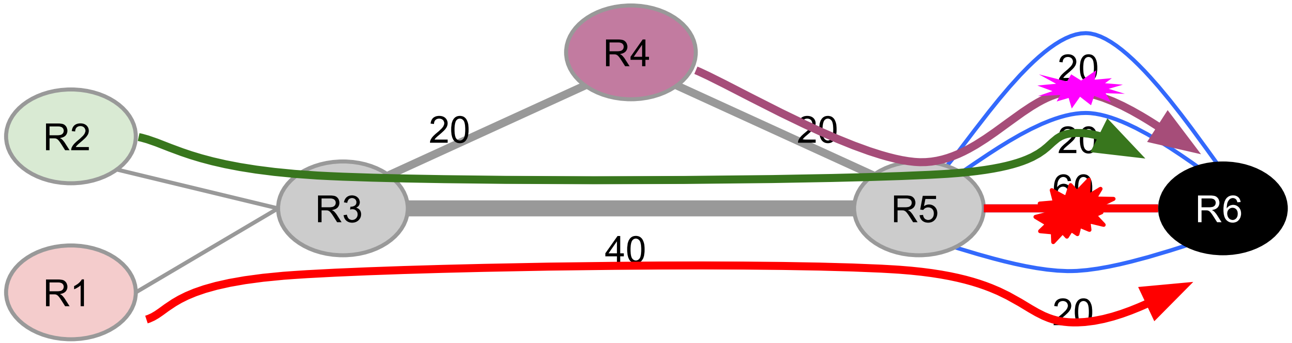
- Flows: R1->R6: 20; R2->R6: 20; R4->R6: 20



- R5-R6 link fails
  - R1, R2, R4 *autonomously* try for next best path
  - R1 wins, R2, R4 retry for next best path

Convergence After Failure

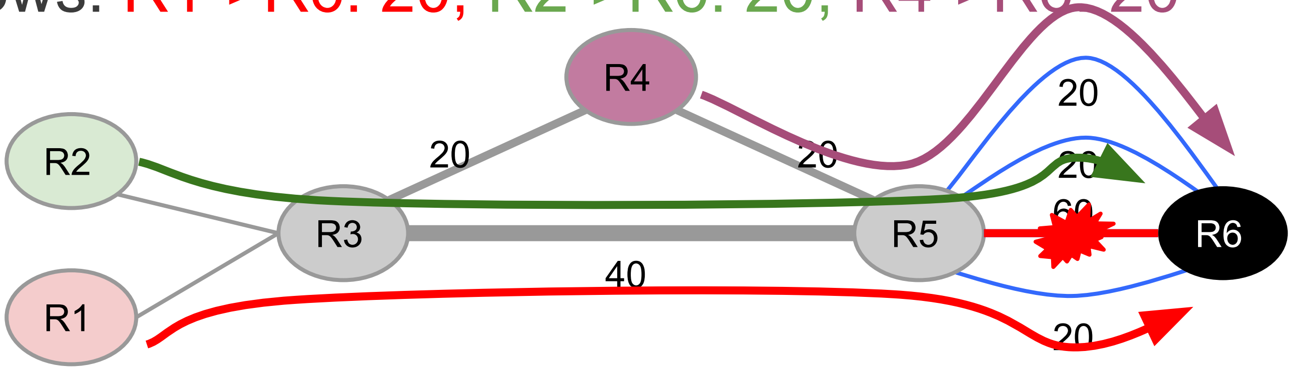
- Flows: R1->R6: 20; R2->R6: 20; R4->R6: 20



- R5-R6 link fails
    - R1, R2, R4 *autonomously* try for next best path
    - R1 wins, R2, R4 retry for next best path
    - R2 wins this round, R4 retries again

Convergence After Failure

- Flows: R1->R6: 20; R2->R6: 20; R4->R6: 20



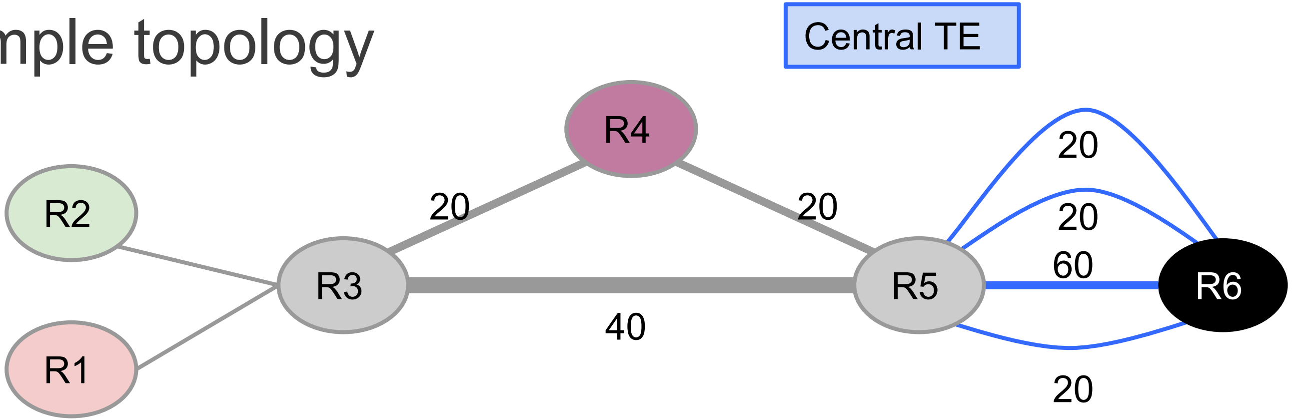
Distributed Traffic Engineering Protocols

- R5-R6 link fails
  - R1, R2, R4 *autonomously* try for next best path
  - R1 wins, R2, R4 retry for next best path
  - R2 wins this round, R4 retries again
  - R4 finally gets third best path!

Centralized Traffic Engineering

Google

- Simple topology

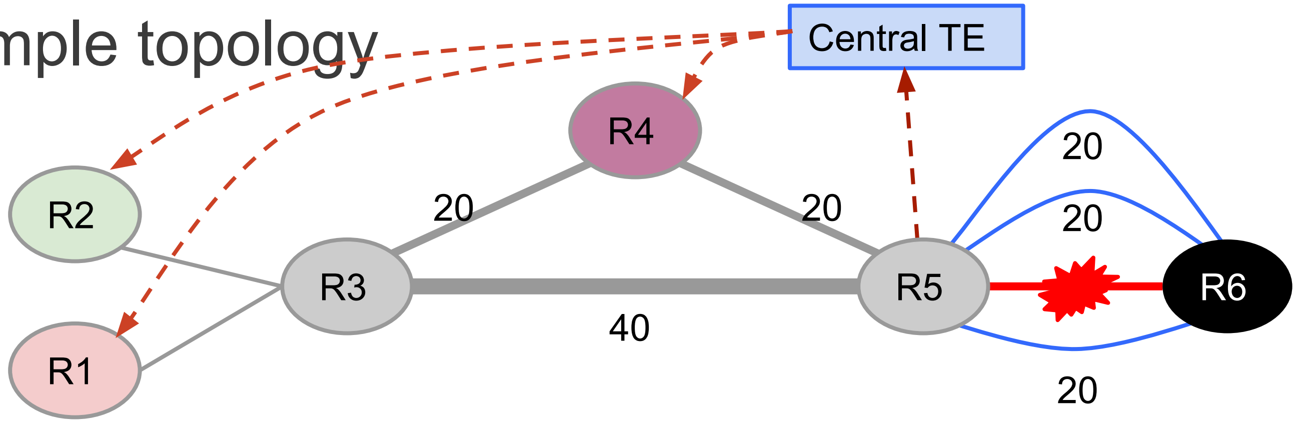


Centralized Traffic Engineering Protocols

- Flows:
  - R1->R6: 20; R2->R6: 20; R4->R6: 20

Centralized Traffic Engineering

- Simple topology



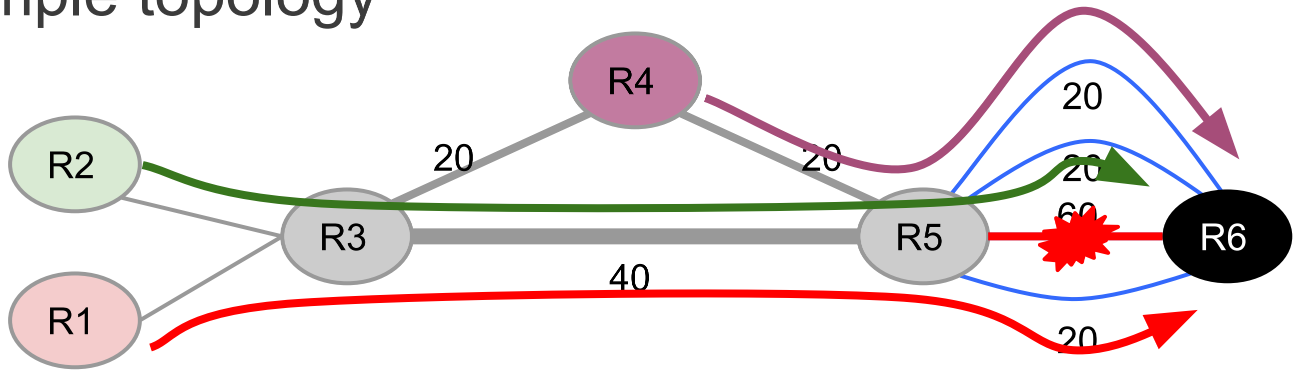
Centralized Traffic Engineering Protocols

- Flows:
  - R1->R6: 20; R2->R6: 20; R4->R6: 20
- R5-R6 fails
  - R5 informs TE, which programs routers in one shot

Centralized Traffic Engineering

Google

- Simple topology



Centralized Traffic Engineering Protocols

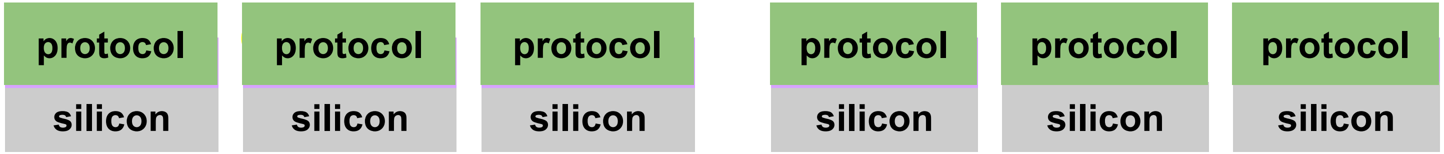
- Flows:
  - R1->R6: 20; R2->R6: 20; R4->R6: 20
- R5-R6 link fails
  - R5 informs TE, which programs routers in one shot
  - Leads to faster realization of target optimum

Advantages of Centralized TE

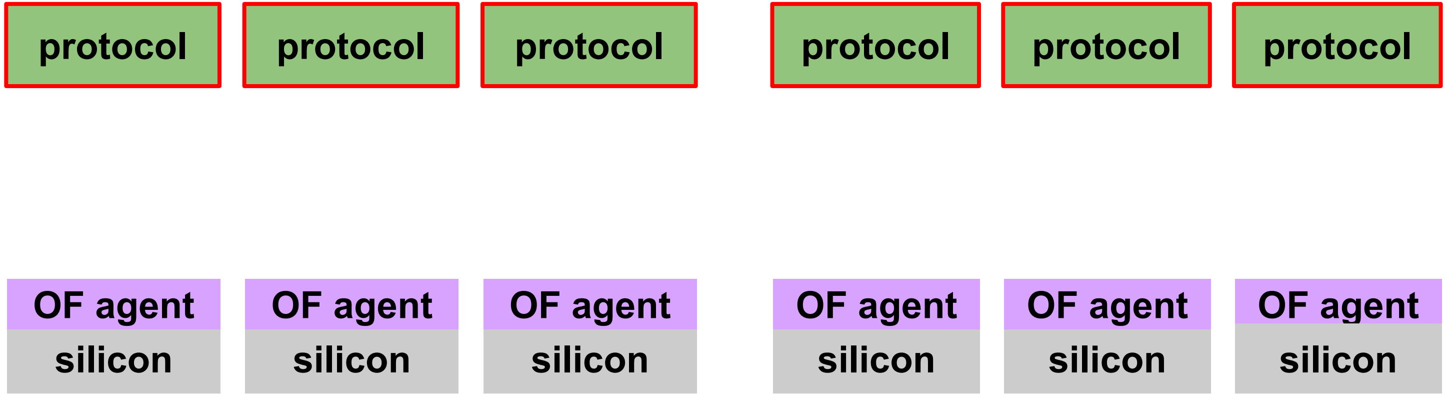
- Better network utilization with global picture
- Converges faster to target optimum on failure
- Allows more control and specifying intent
  - Deterministic behavior simplifies planning vs. overprovisioning for worst case variability
- Can mirror production event streams for testing
  - Supports innovation and robust SW development
- Controller uses modern server hardware
  - 50x (!) better performance

B4 Architecture

B4 Site: SDN Architecture

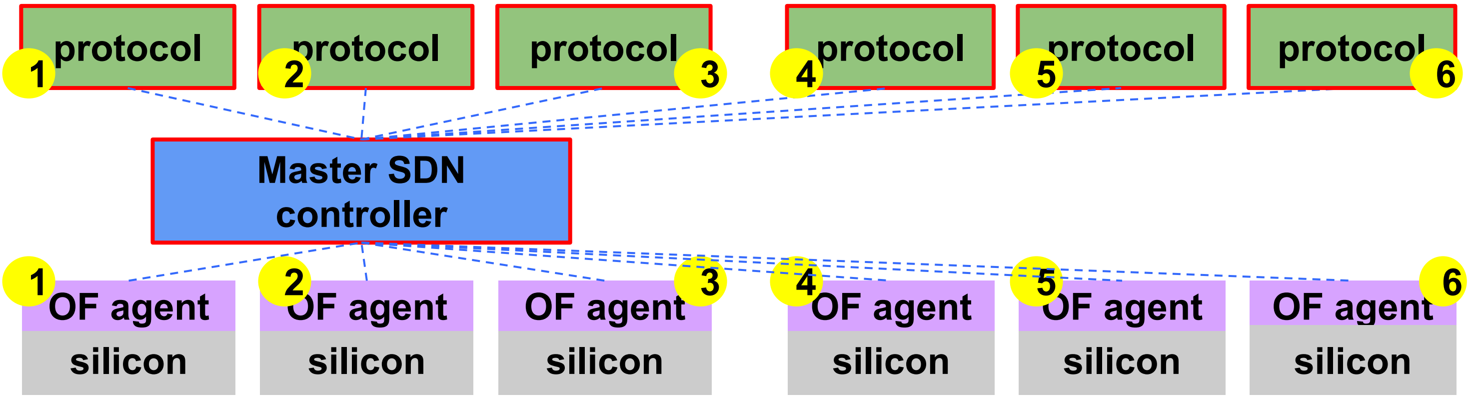


B4 Site: SDN Architecture



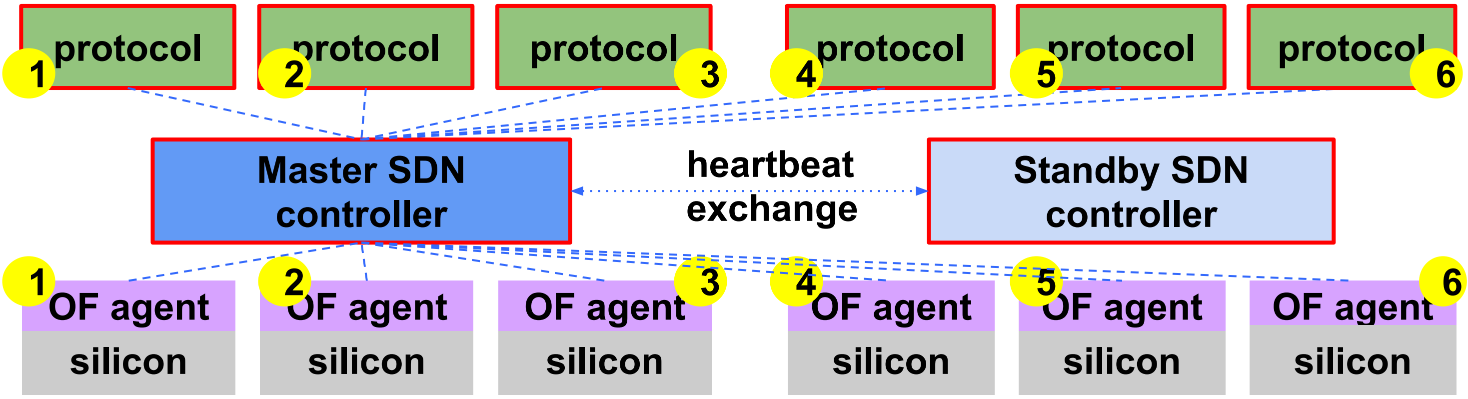
B4 Site: SDN Architecture

Traditional WAN integrated with SDN: still speaking ISIS/BGP



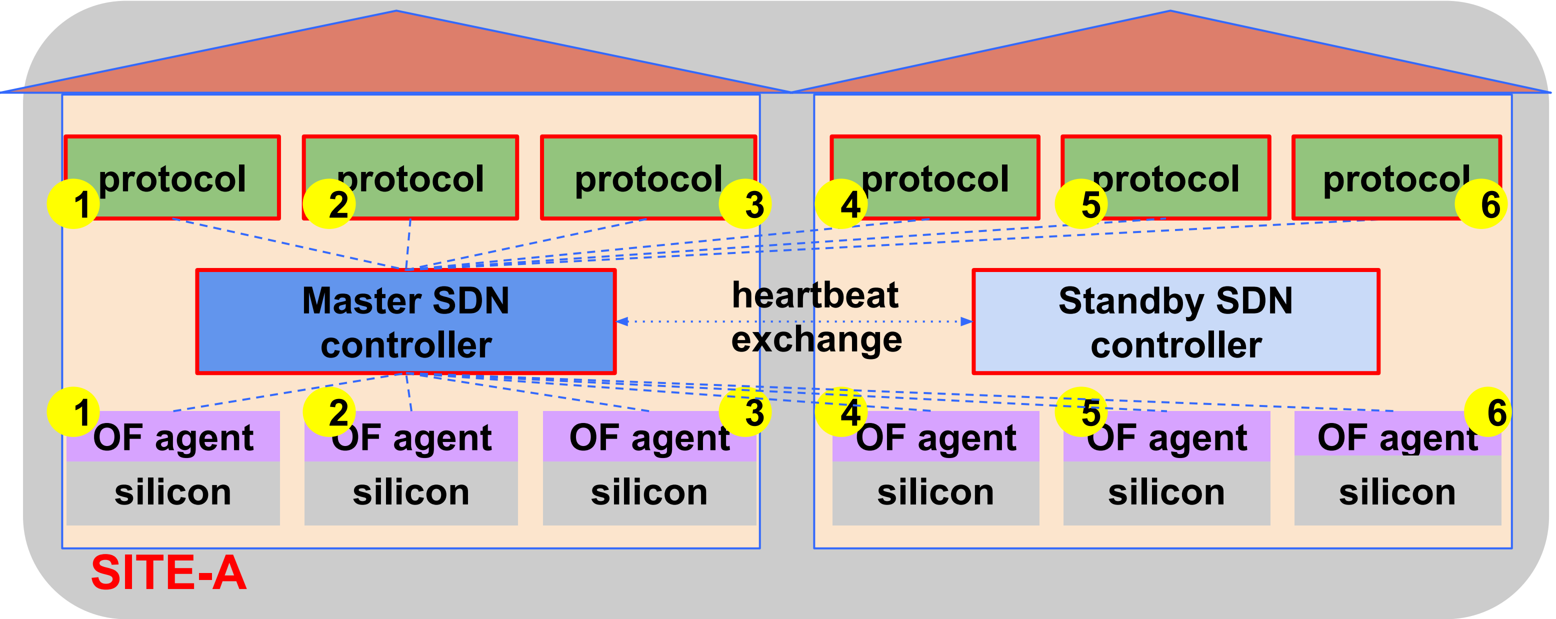
B4 Site: SDN Architecture

Traditional WAN integrated with SDN: still speaking ISIS/BGP



B4 Site: SDN Architecture

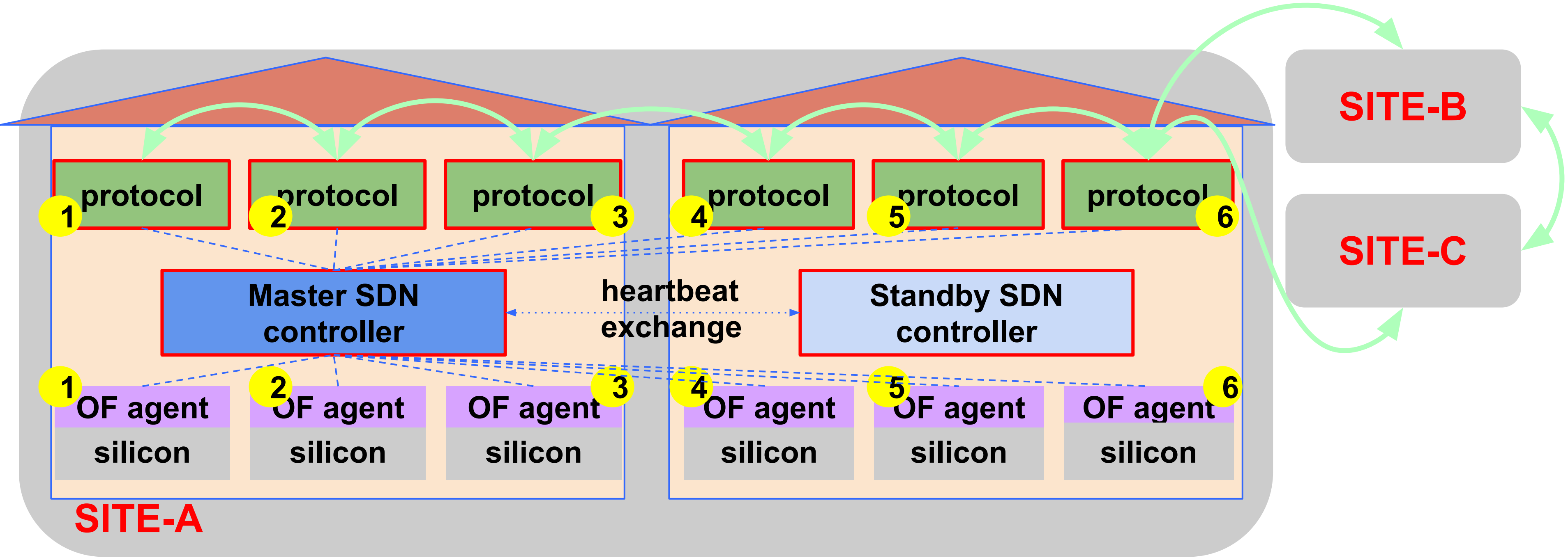
Traditional WAN integrated with SDN: still speaking ISIS/BGP



Unit of management is a site = fabric

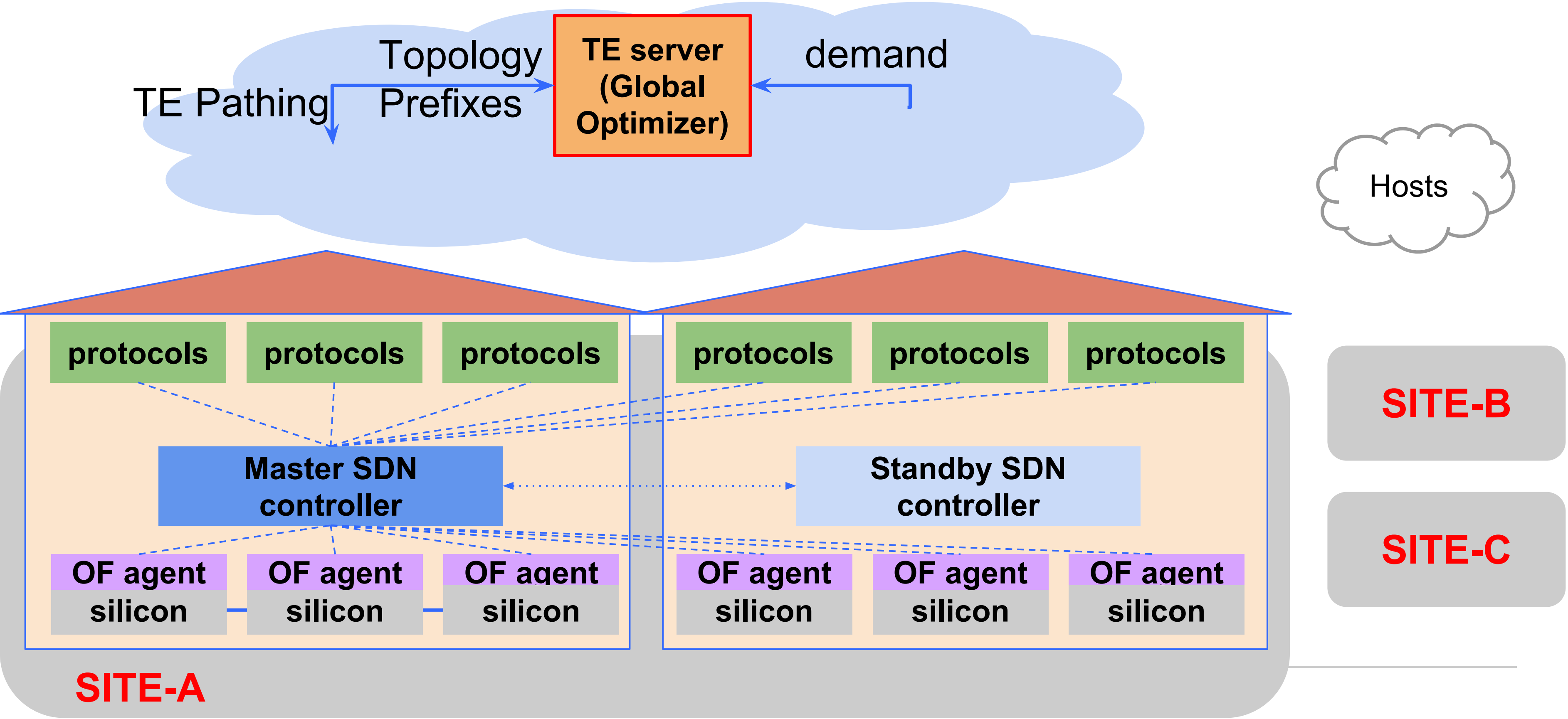
B4 Site: SDN Architecture

Traditional WAN integrated with SDN: still speaking ISIS/BGP

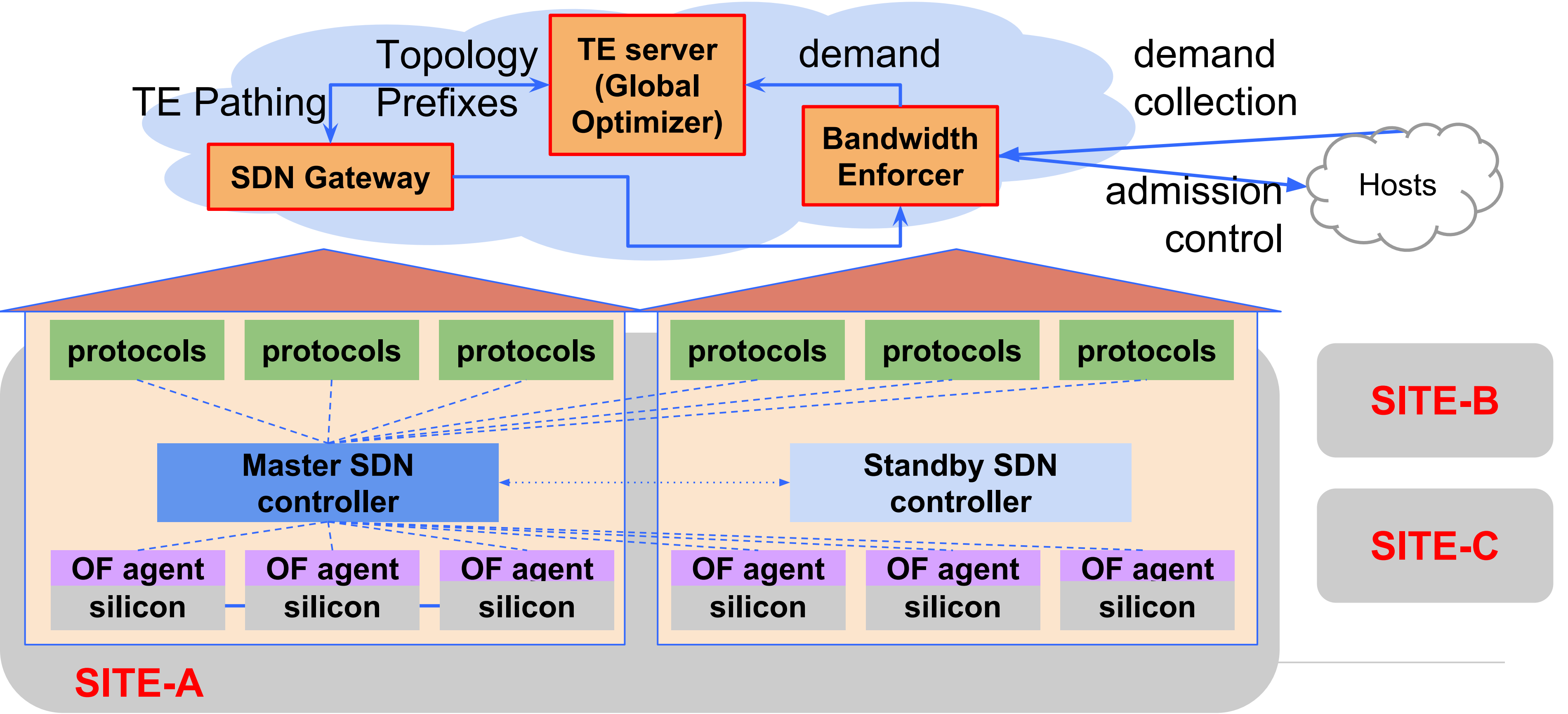


Unit of management is a site = fabric

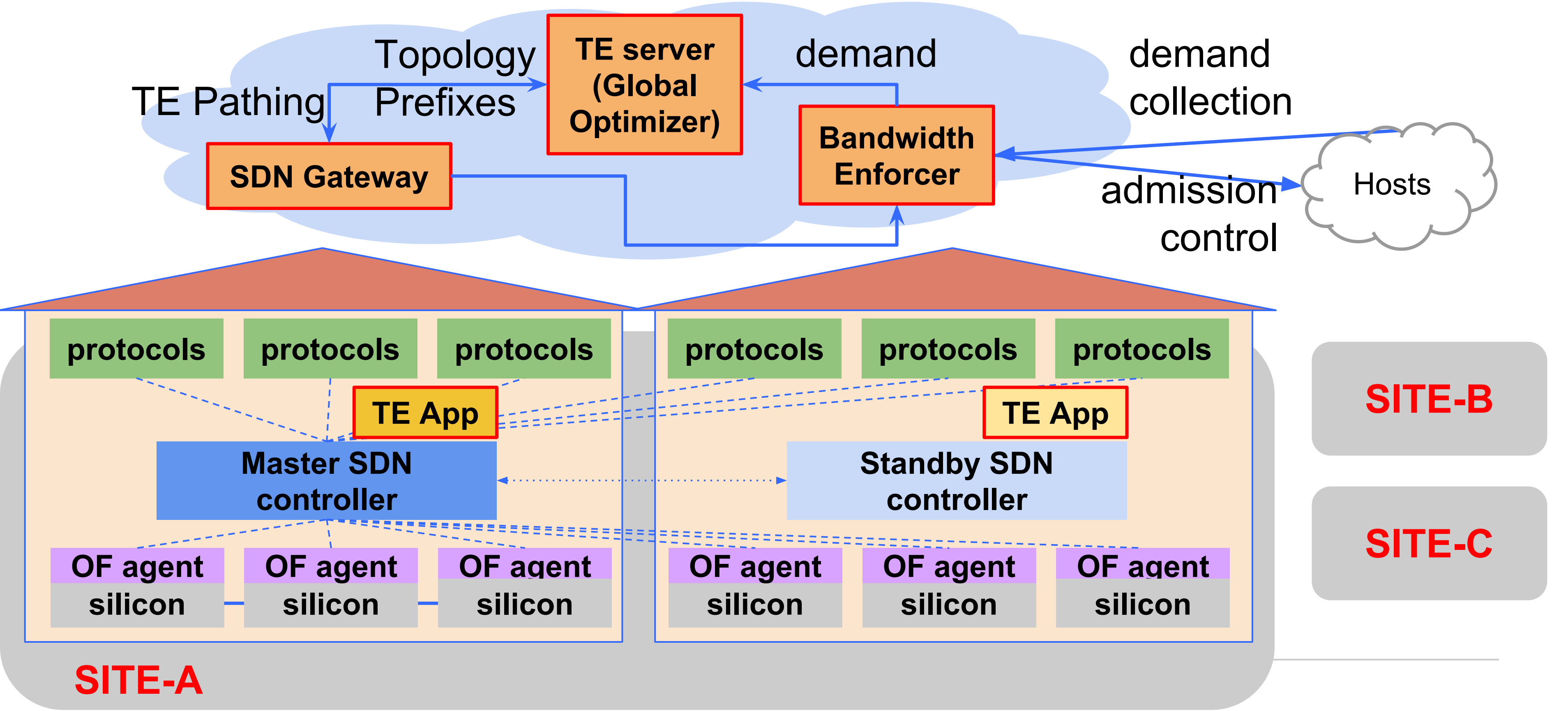
Control Plane Architecture



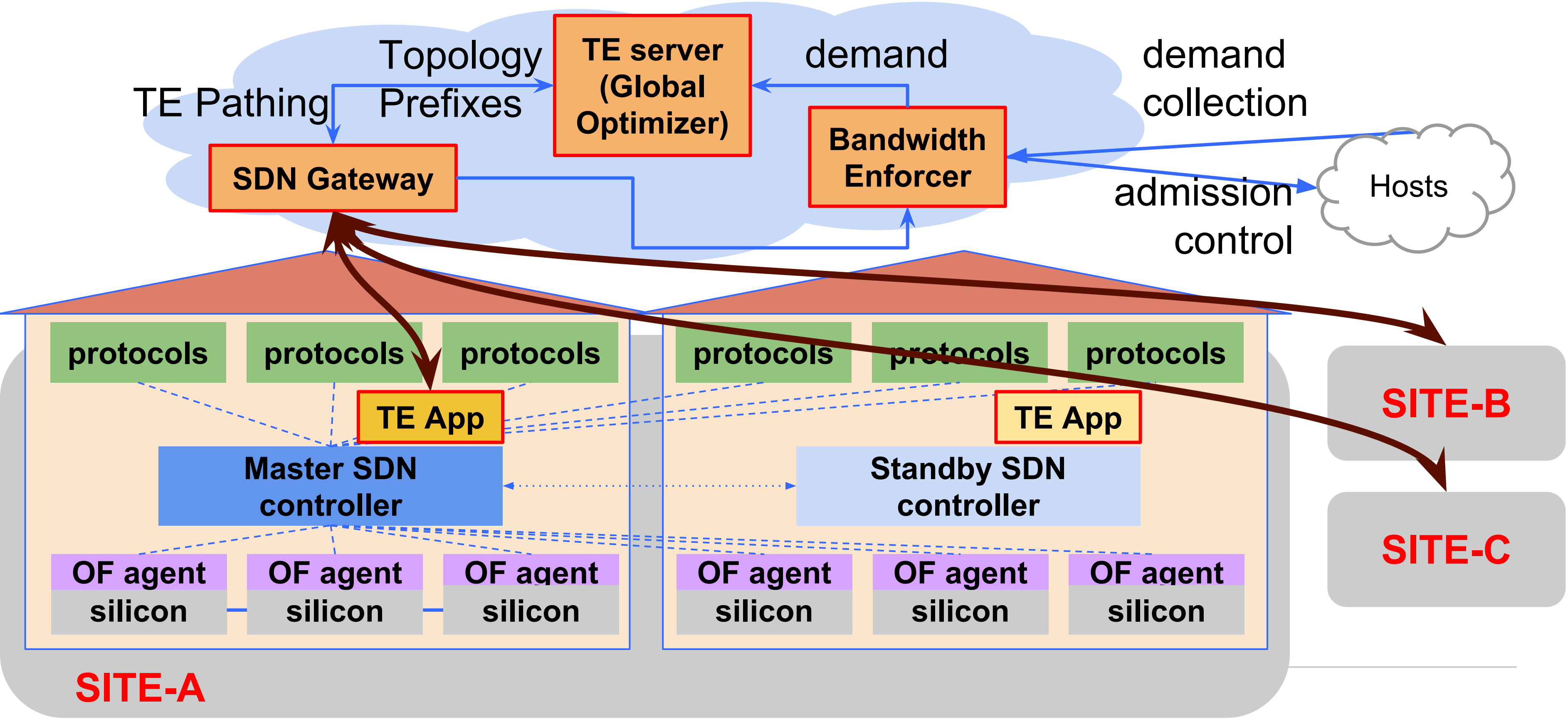
Control Plane Architecture



Control Plane Architecture

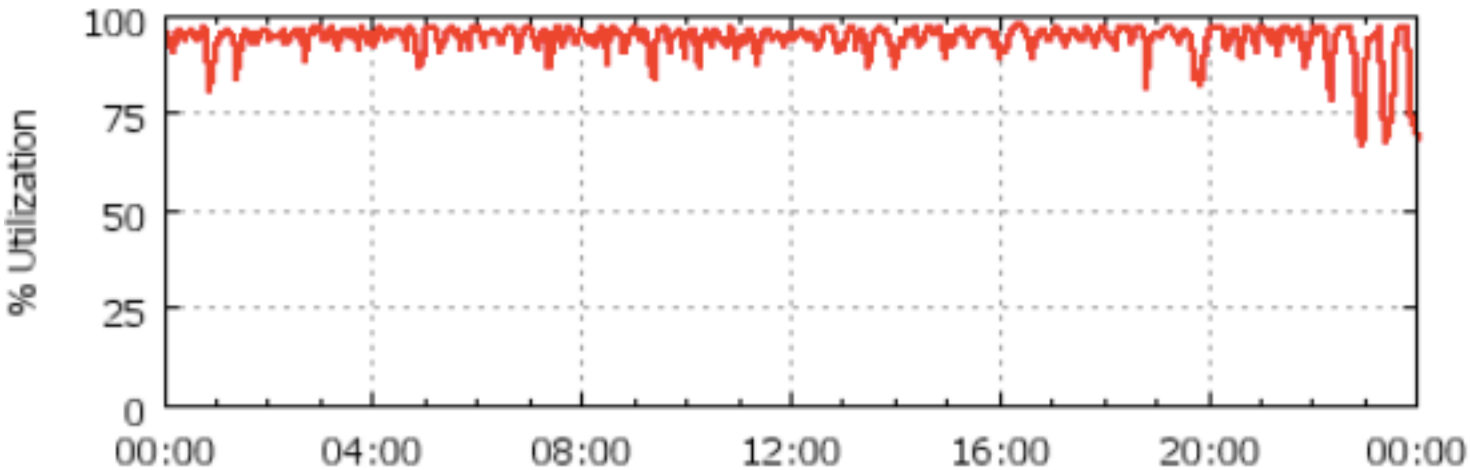


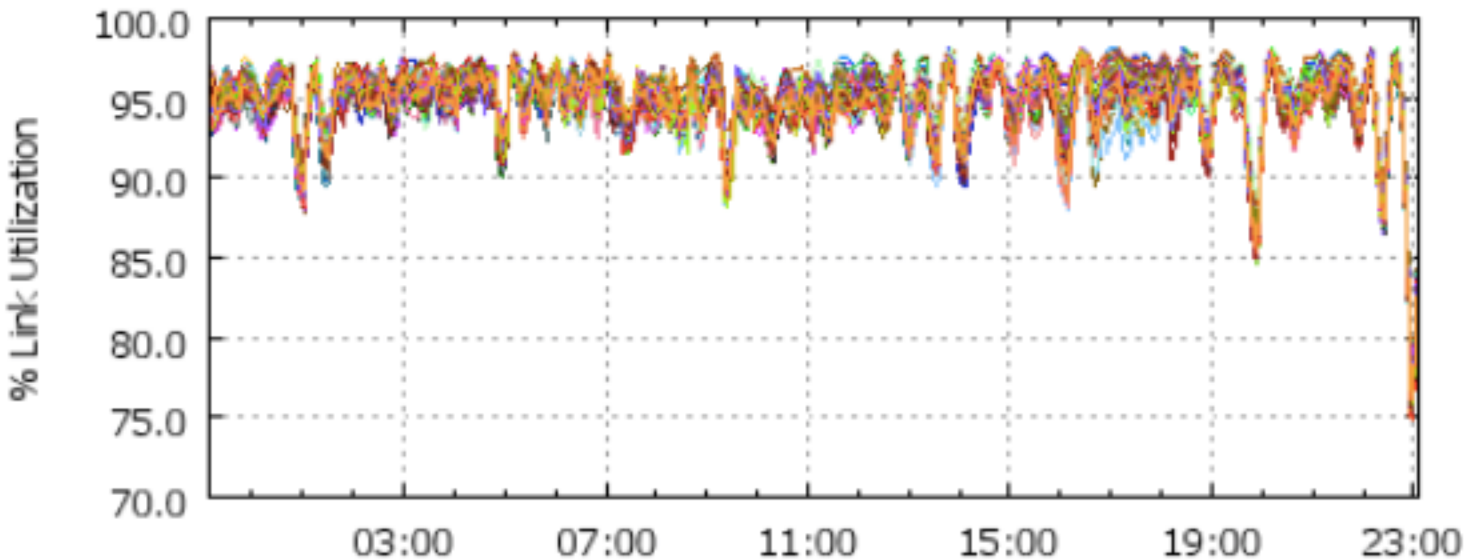
Control Plane Architecture



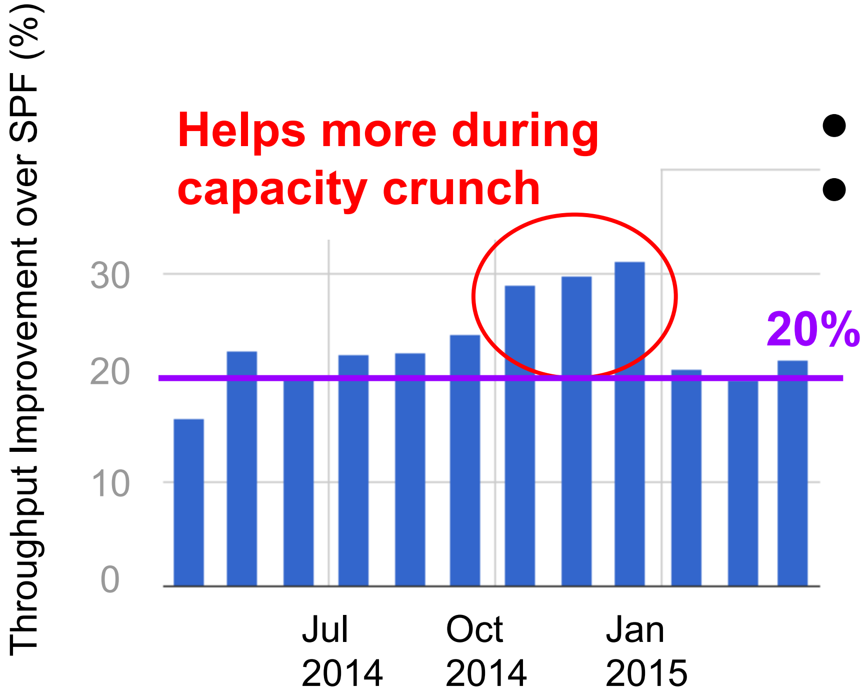
Benefits of SDN B4 with Centralized Traffic Engineering

High and Balanced Utilization





Benefits of TE Over Shortest Path



- ~20% increase in throughput over SPF
- Larger benefits during capacity crunch

Lowers the requirement for bandwidth provisioning

Other Benefits

Software and hardware feature roll outs decoupled

- Software timescale feature roll out
  - Hitless SW upgrades and new features
    - No packet loss and no capacity degradation
    - Most feature releases do not touch the switch
- Slower HW upgrades
  - 3 generations of HW under same SDN architecture

Conclusions

- SDN is beneficial in real-world
  - Centralized TE delivered upto 30% additional throughput!
  - Decoupled software and hardware rollout
- Lessons to work in practice
  - System performance: Flow control between components
  - Availability: Robust reachability for master election
  - Scale: Hierarchical topology abstraction