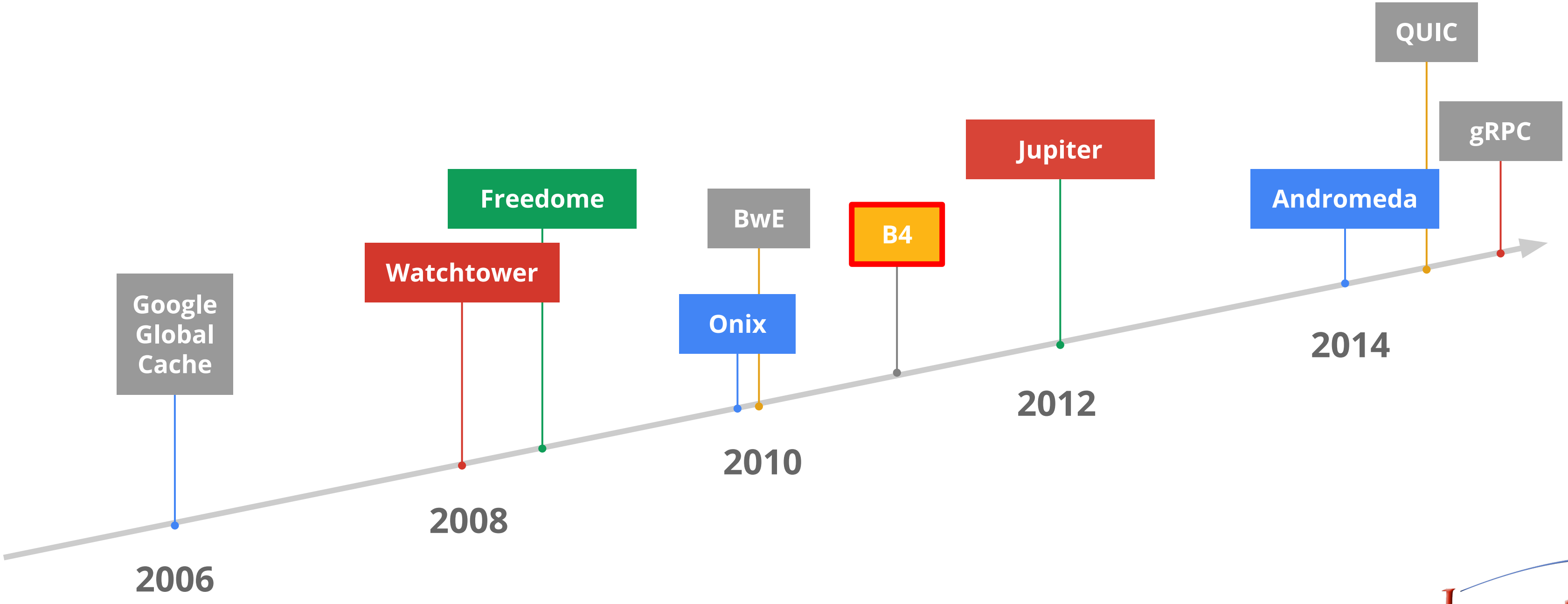
Lessons Learned from B4, Google's SDN WAN

Subhasree Mandal

July 9, 2015

Google Innovations in Networking



More Than the Sum of Parts

Google Networking works together as an integrated whole

- **B4: WAN interconnect**
- GGC: edge presence
- Jupiter: building scale datacenter network
- Freedome: campus-level interconnect
- Andromeda: isolated, high-performance slices of the physical network

Publications in **INFOCOM 2012, SIGCOMM 2013, SIGCOMM 2014, CoNEXT 2014, EuroSys 2014, SIGCOMM 2015**

Motivation for SDN B4

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

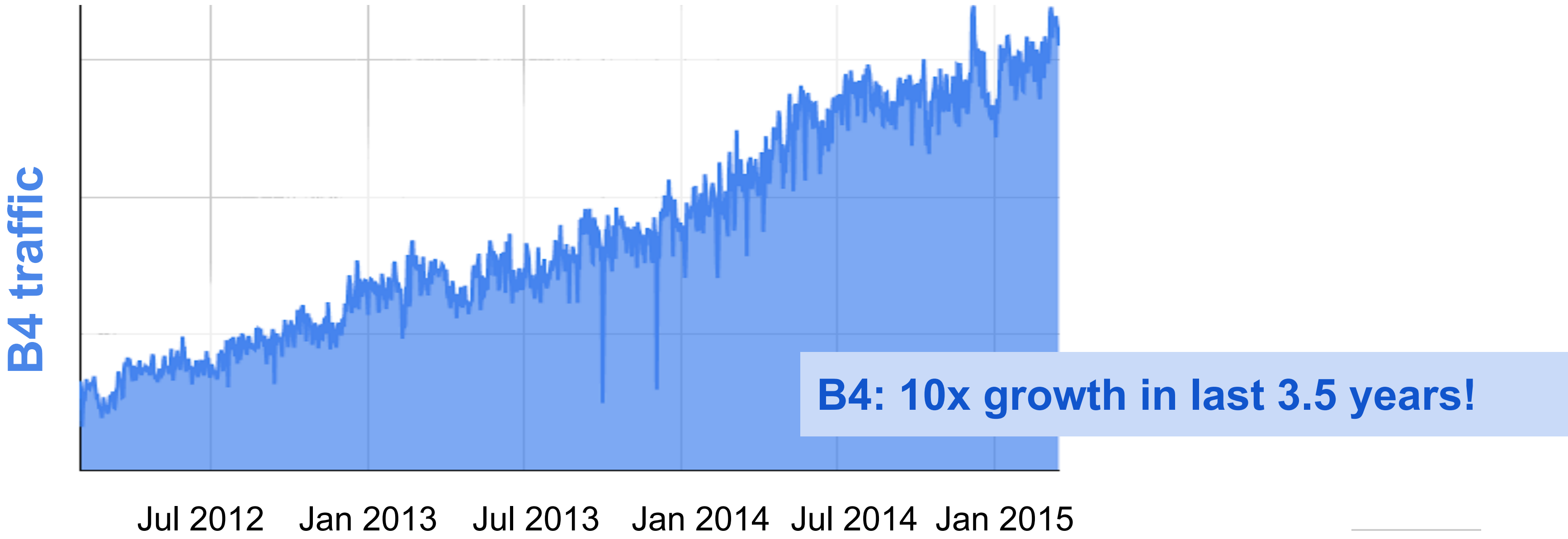
Need a network to connect these data centers to one another

- 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

Google

- 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

Google

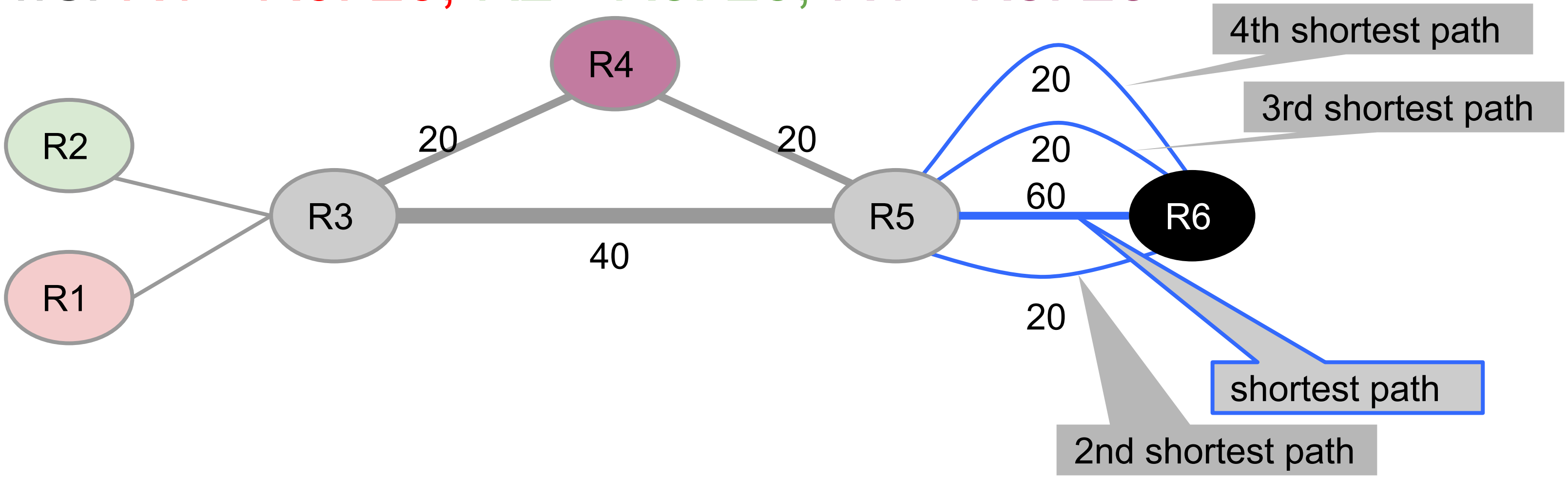
- Background for Traffic Engineering (TE)
- B4-SDN/TE Architecture with OpenFlow protocol
- Benefits of B4-SDN/TE
- Lessons learnt on SDN in three key areas

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

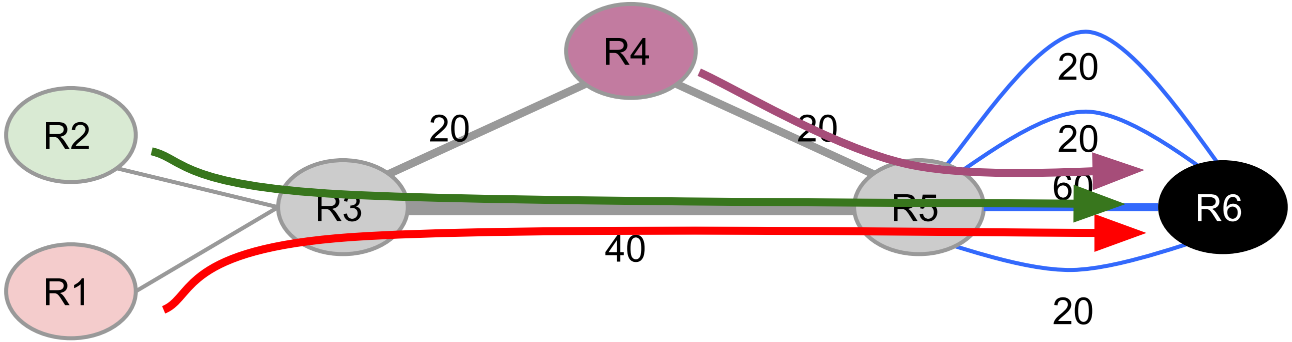
Convergence After Failure

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



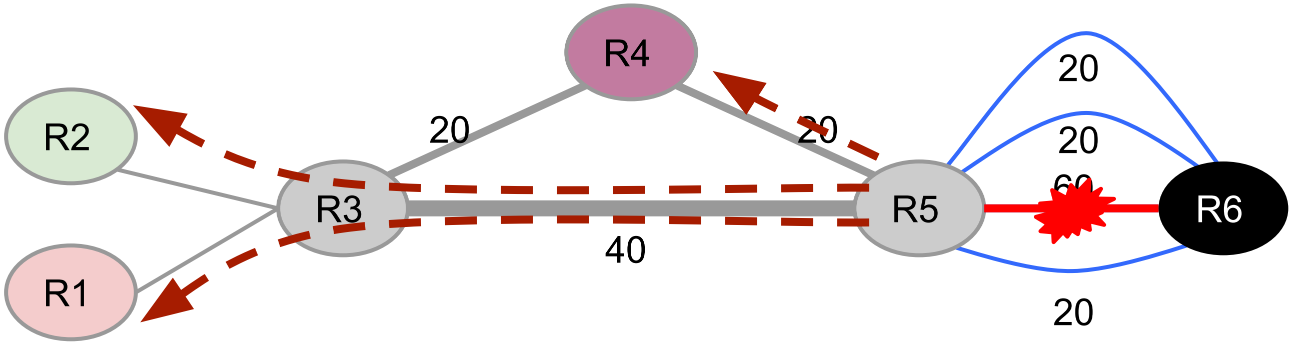
Convergence After Failure

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



Convergence After Failure

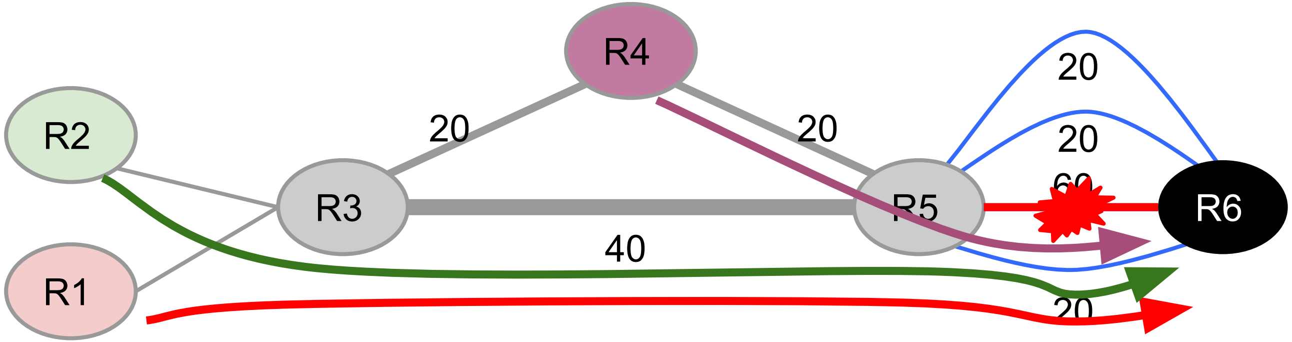
- Flows: R1->R6: 20; R2->R6: 20; R4->R6: 20



- 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

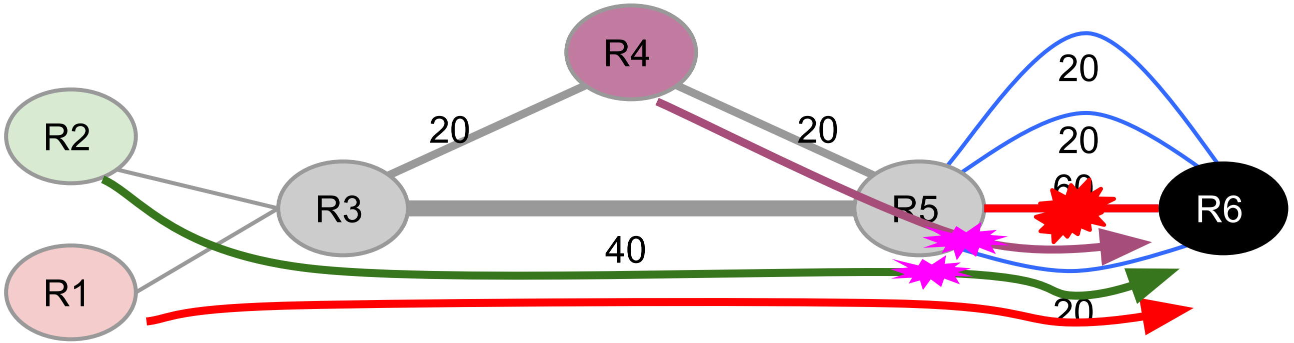


No Traffic Engineering

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

Convergence After Failure

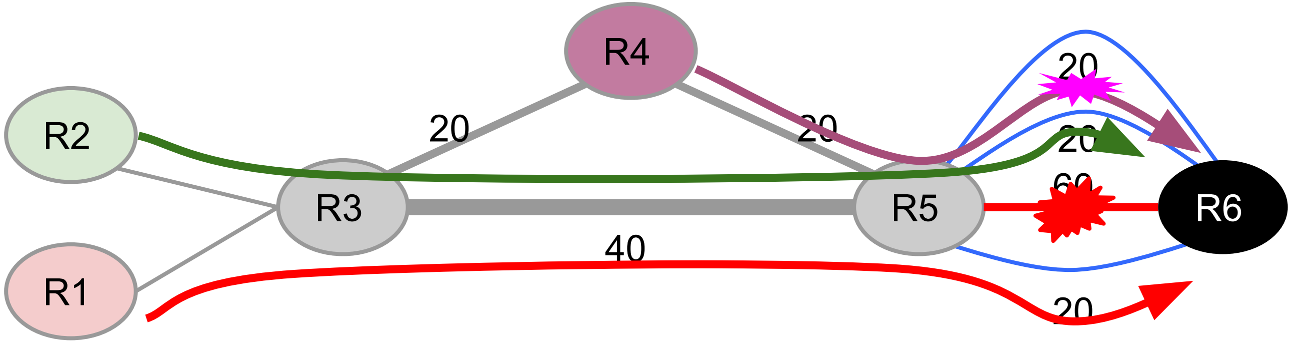
- Flows: R1->R6: 20; R2->R6: 20; R4->R6: 20



- 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

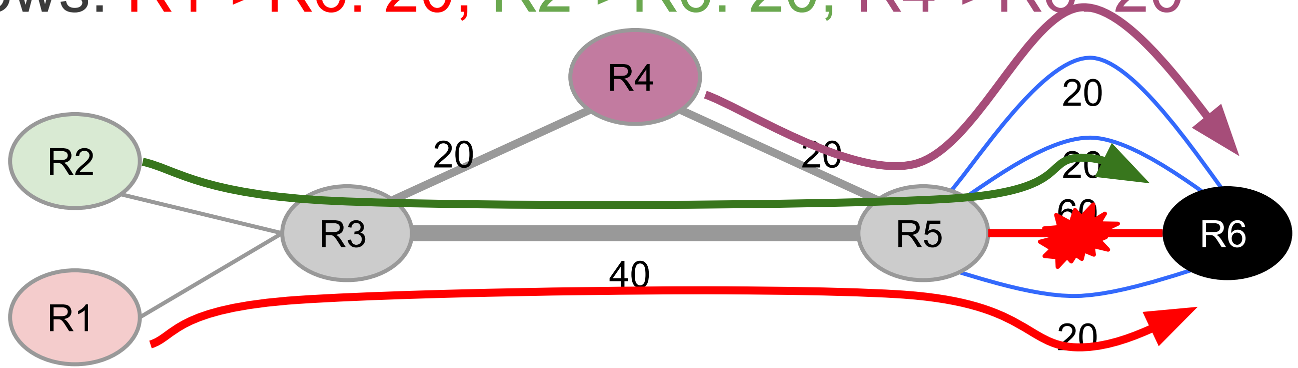
- Flows: R1->R6: 20; R2->R6: 20; R4->R6: 20



- 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

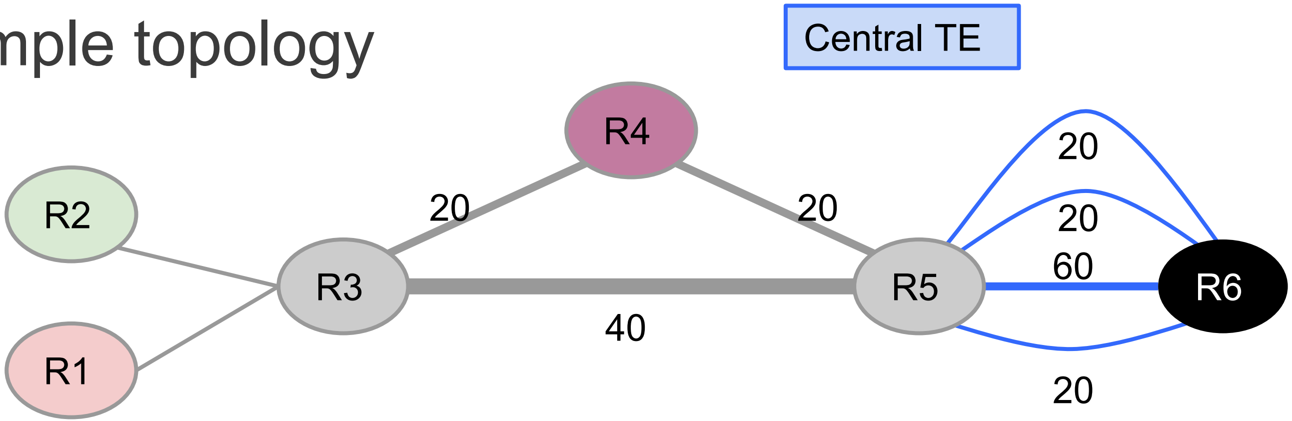


Distributed Traffic Engineering Protocols

- 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

- Simple topology



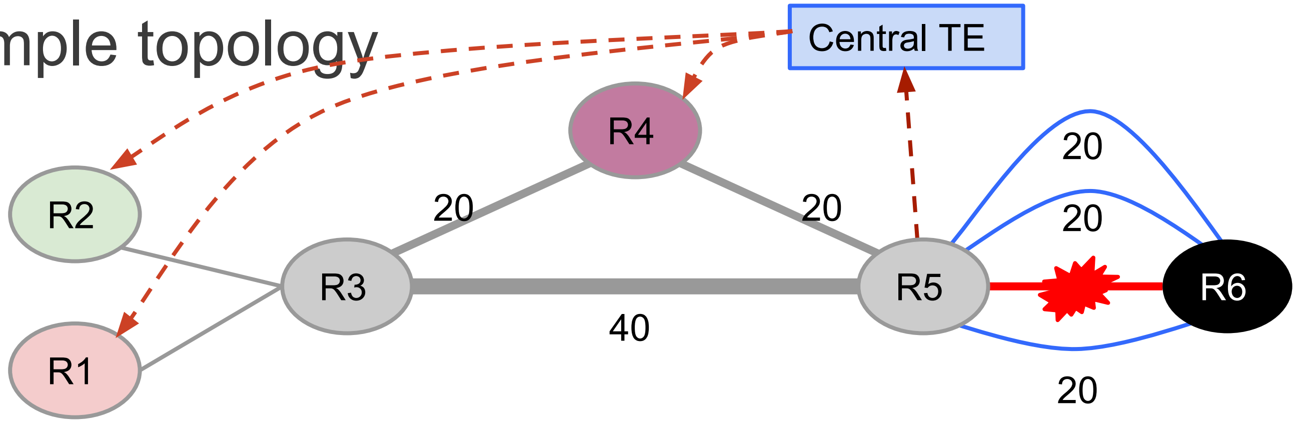
Centralized Traffic Engineering Protocols

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

Centralized Traffic Engineering

Google

- Simple topology



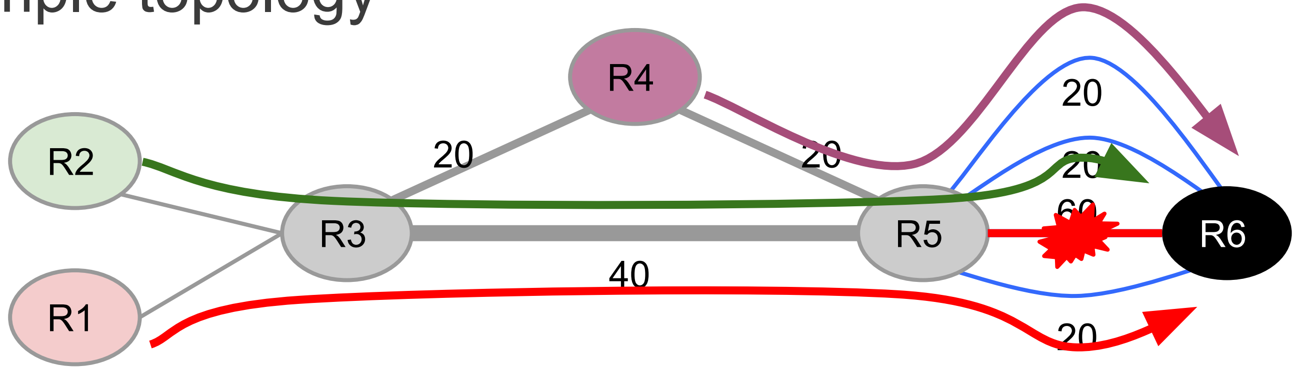
Centralized Traffic Engineering Protocols

- 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

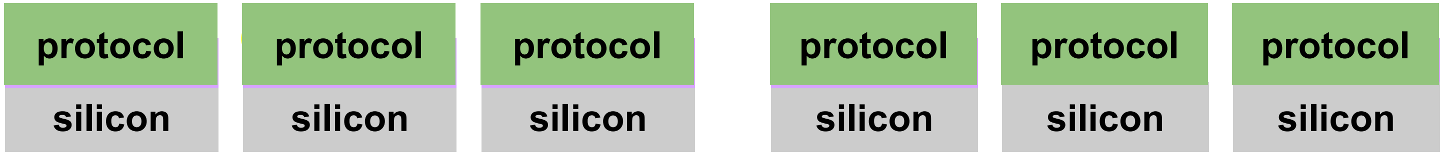
- Flows:
  - R1->R6: 20; R2->R6: 20; R4->R6: 20
- R5-R6 link fails
  - R5 informs TE, which programs routers in one shot
  - Leads to faster realization of target optimum

Advantages of Centralized TE

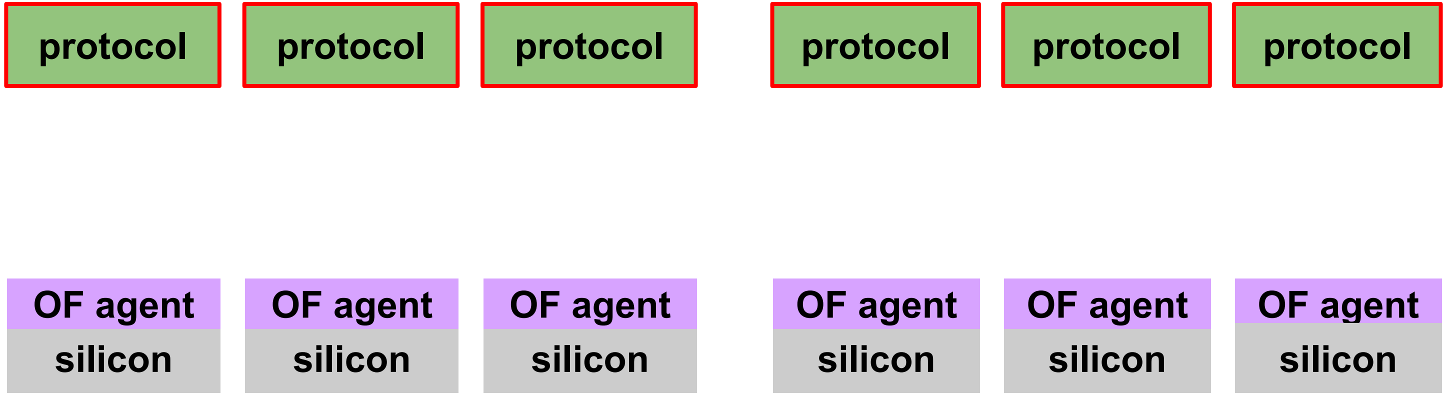
- Better network utilization with global picture
- Converges faster to target optimum on failure
- Allows more control and specifying intent
  - Deterministic behavior simplifies planning vs. overprovisioning for worst case variability
- Can mirror production event streams for testing
  - Supports innovation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

B4 Architecture

B4 Site: SDN Architecture

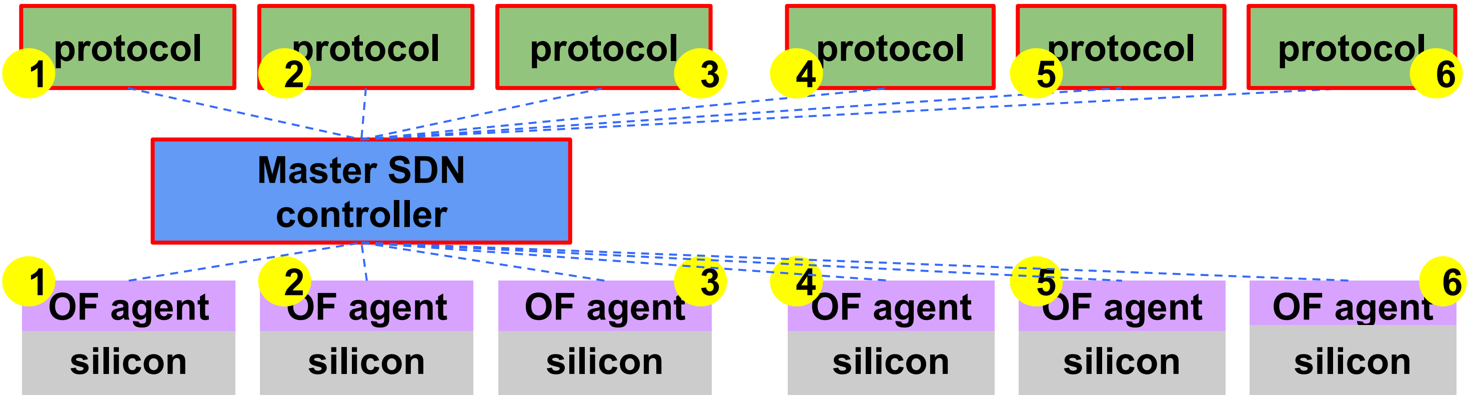


B4 Site: SDN Architecture



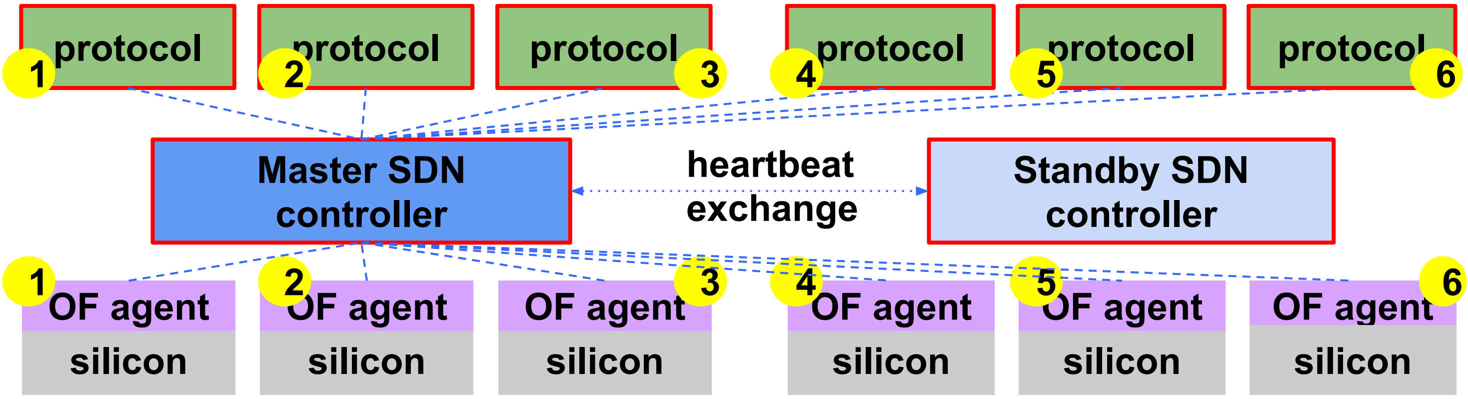
B4 Site: SDN Architecture

Traditional WAN integrated with SDN: still speaking ISIS/BGP



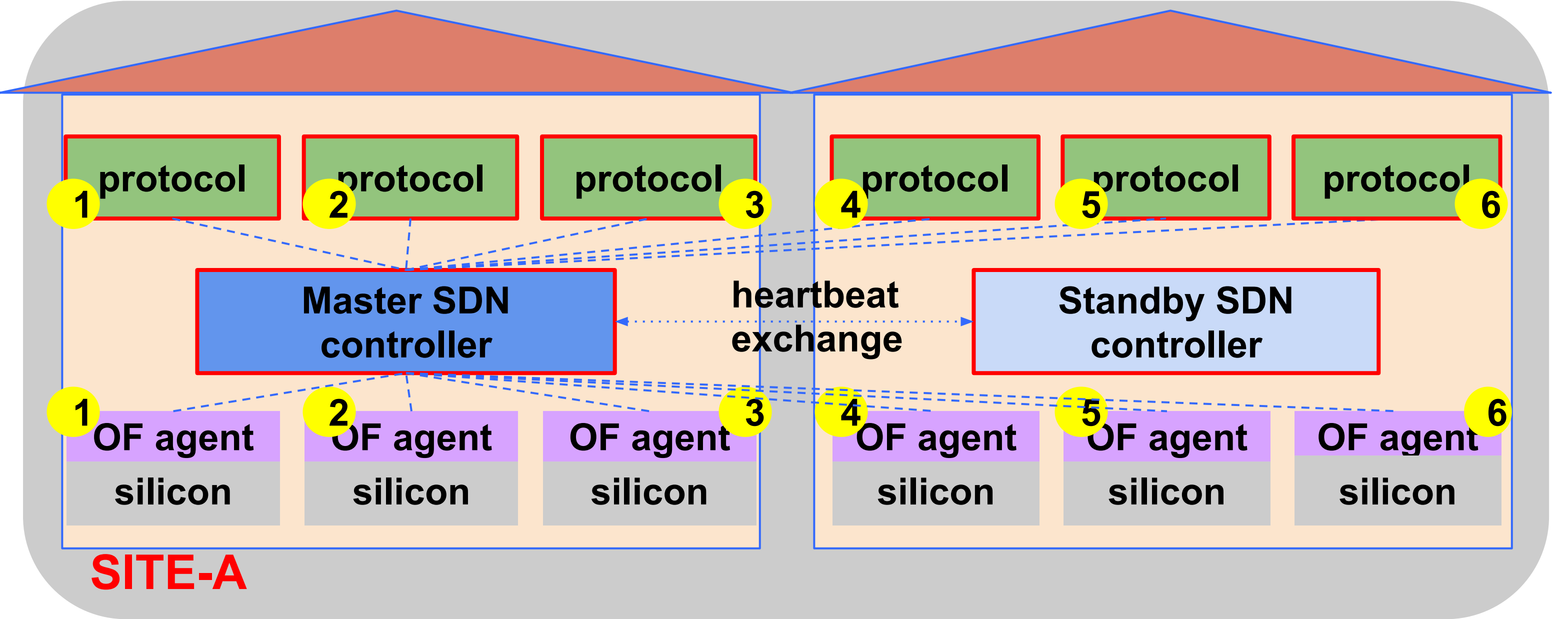
B4 Site: SDN Architecture

Traditional WAN integrated with SDN: still speaking ISIS/BGP



B4 Site: SDN Architecture

Traditional WAN integrated with SDN: still speaking ISIS/BGP

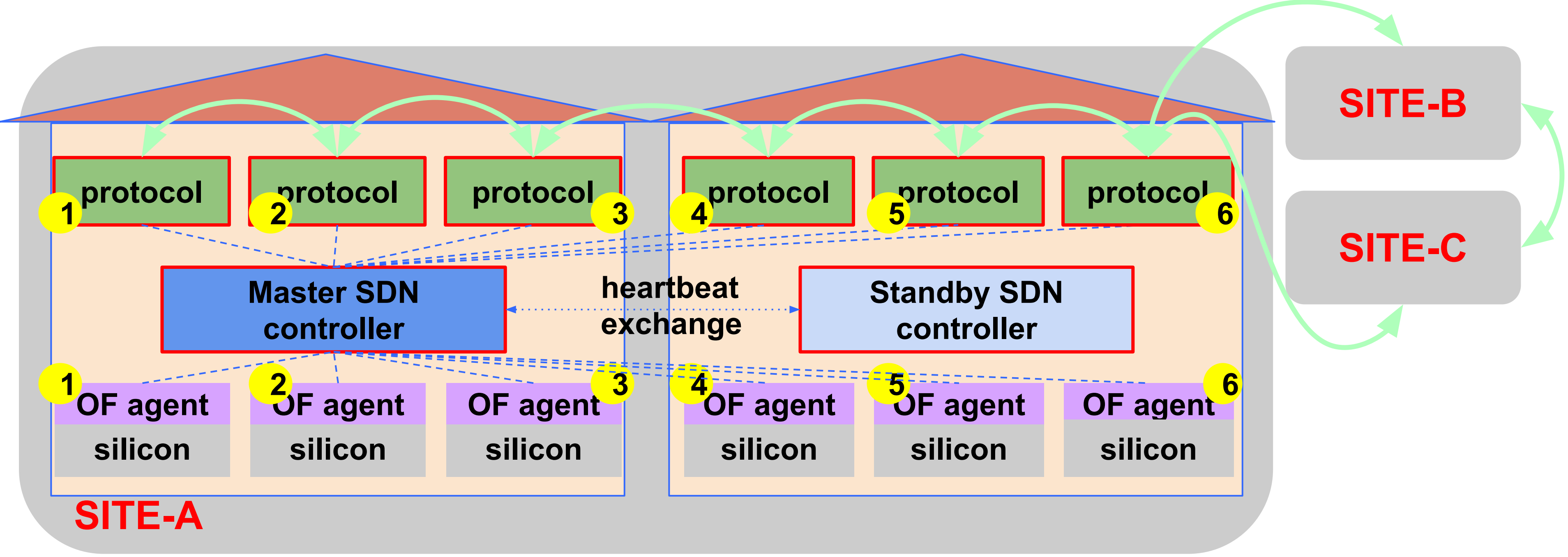


Unit of management is a site = fabric

B4 Site: SDN Architecture

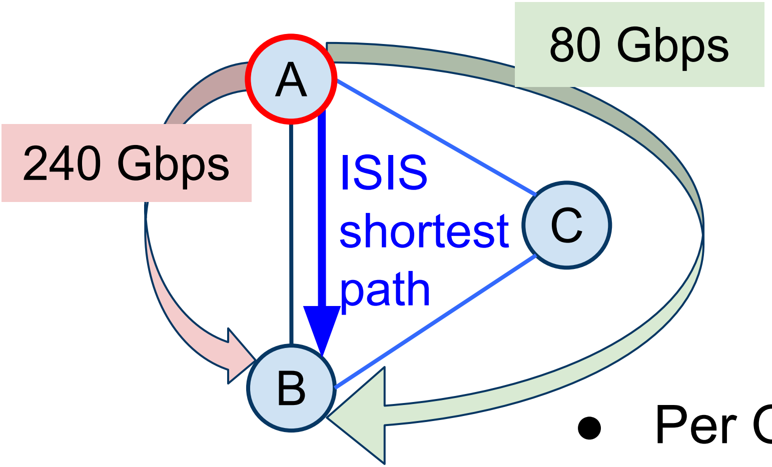
Google

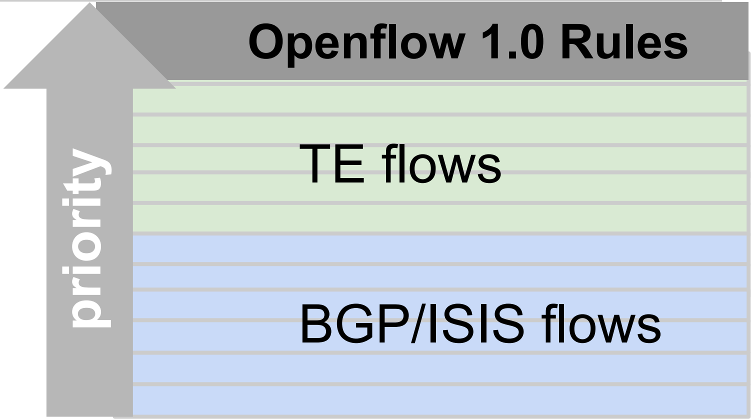
Traditional WAN integrated with SDN: still speaking ISIS/BGP



Unit of management is a site = fabric

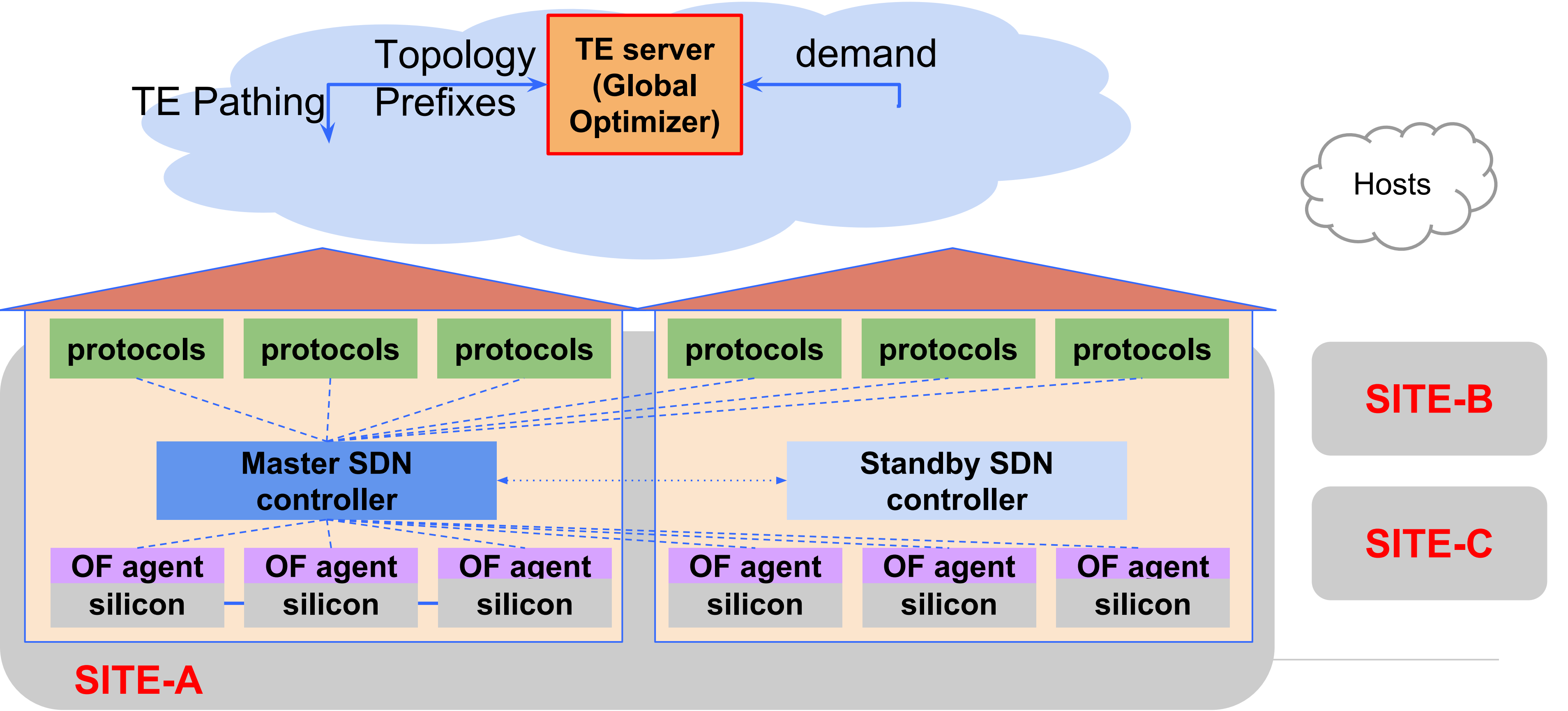
Traffic Engineering Overlay



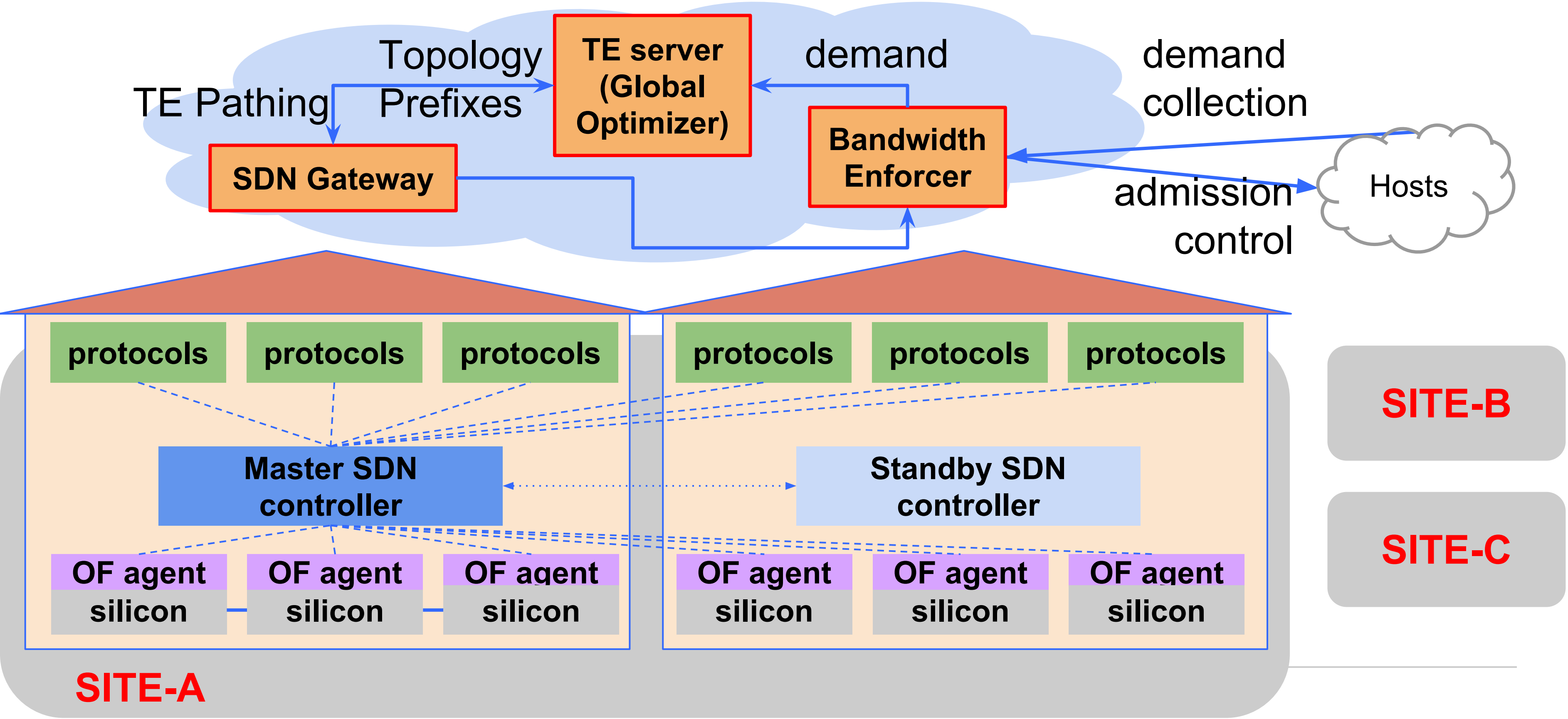


- Per QoS Traffic Engineering (TE)
  - Demand based use of longer paths
  - Max-min fair bandwidth allocation
  - Per app loss/latency/throughput consideration
- TE paths are overlaid on ISIS/BGP routes
  - Higher priority flow rules for TE

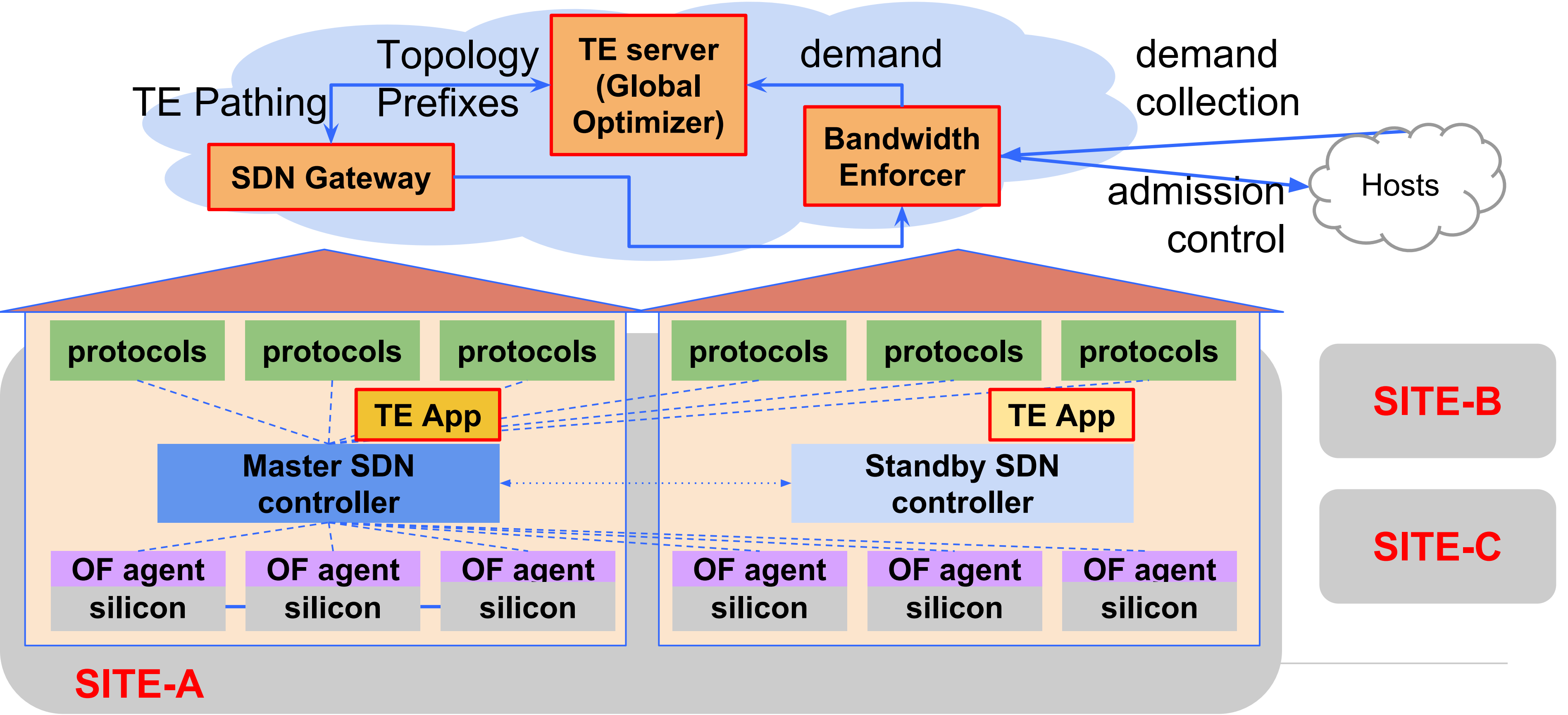
Control Plane Architecture



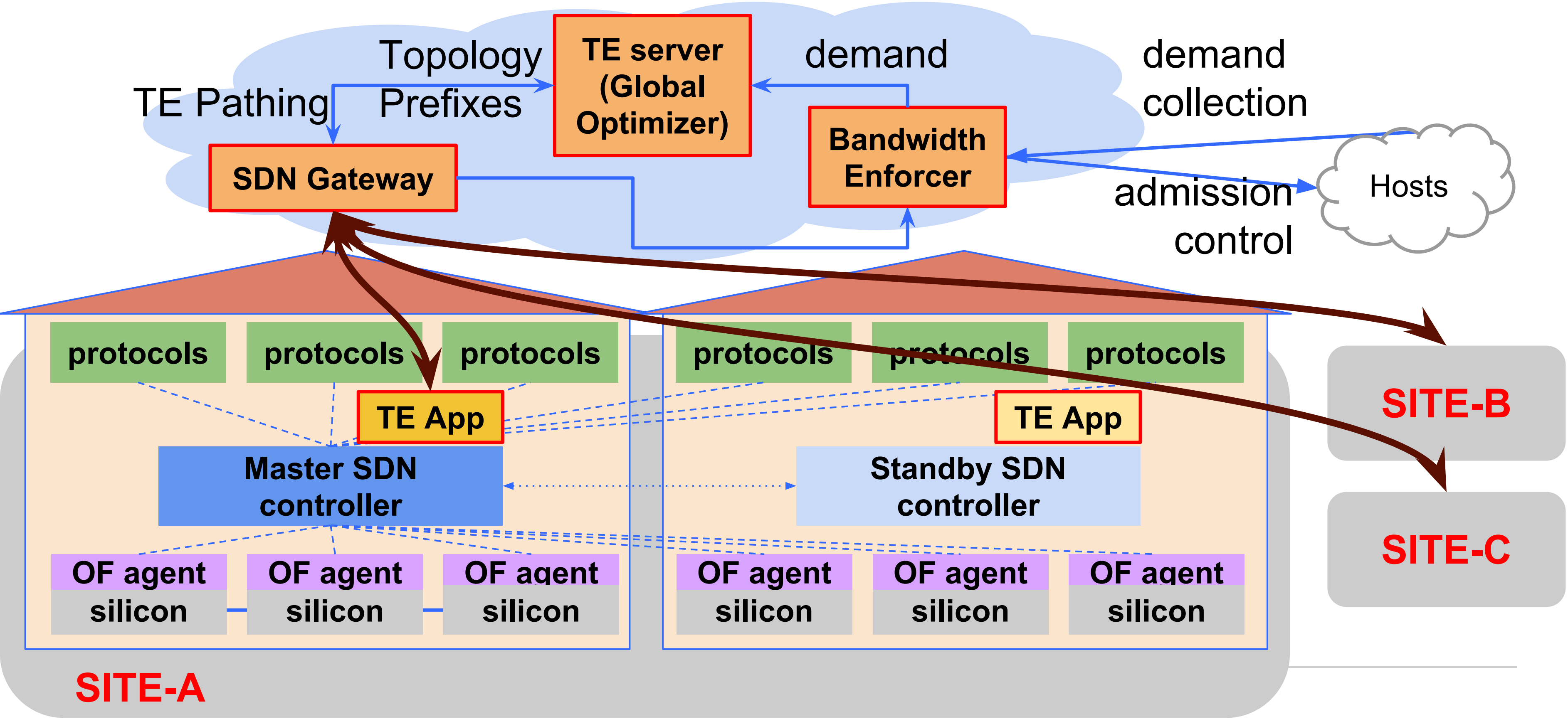
Control Plane Architecture



Control Plane Architecture

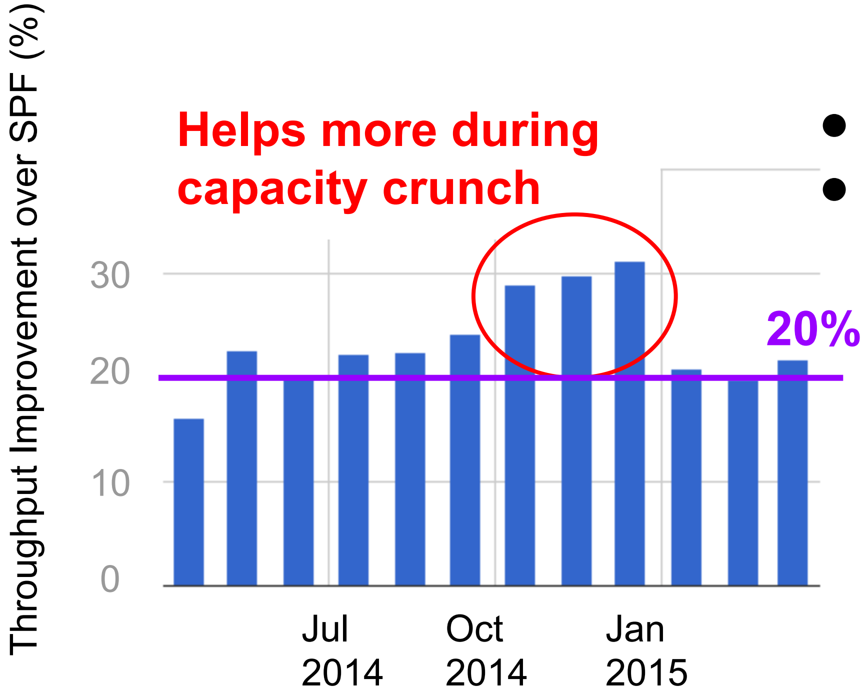


Control Plane Architecture



Benefits of SDN B4 with Centralized Traffic Engineering

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

Lesson on Performance

Controller to Switch Messaging

Initial simple-minded assumptions

- OpenFlow protocol:
  - Flow and control packet (ISIS/BGP/ARP/...) requests sent from controller to OF agent (OFA) sequentially
- OF agent (OFA) can process them in order
- System is always in consistent state

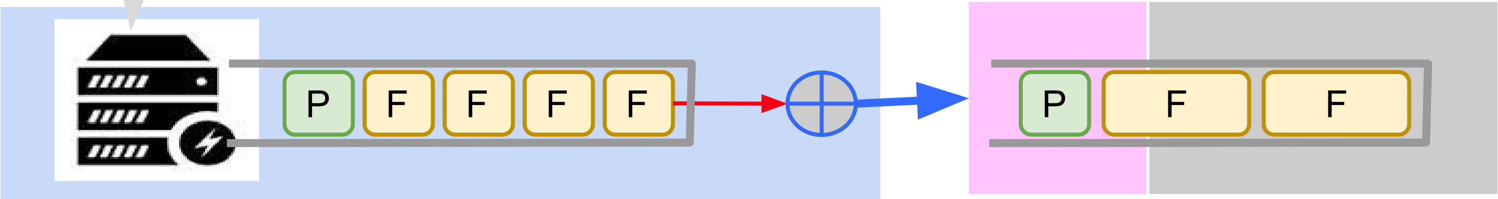
But ….

Messages Backlogged and Delayed!

Google

Fast server

Queue build-up on controller and switch due to slow switch CPU



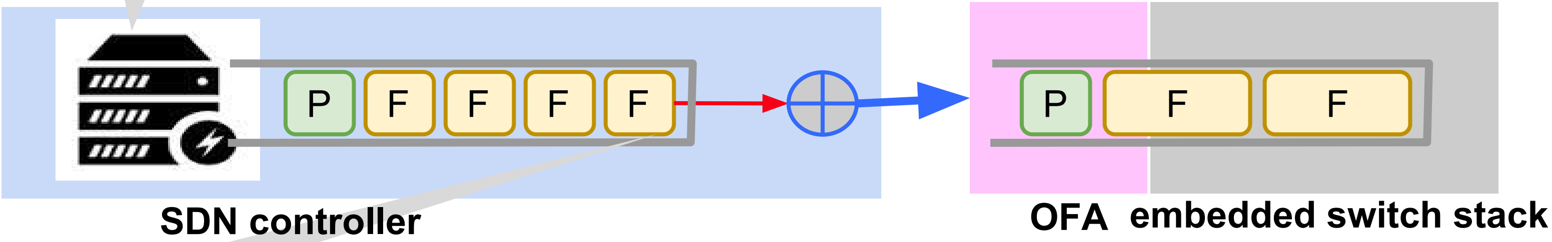
SDN controller

OFA embedded switch stack

Messages Backlogged and Delayed!

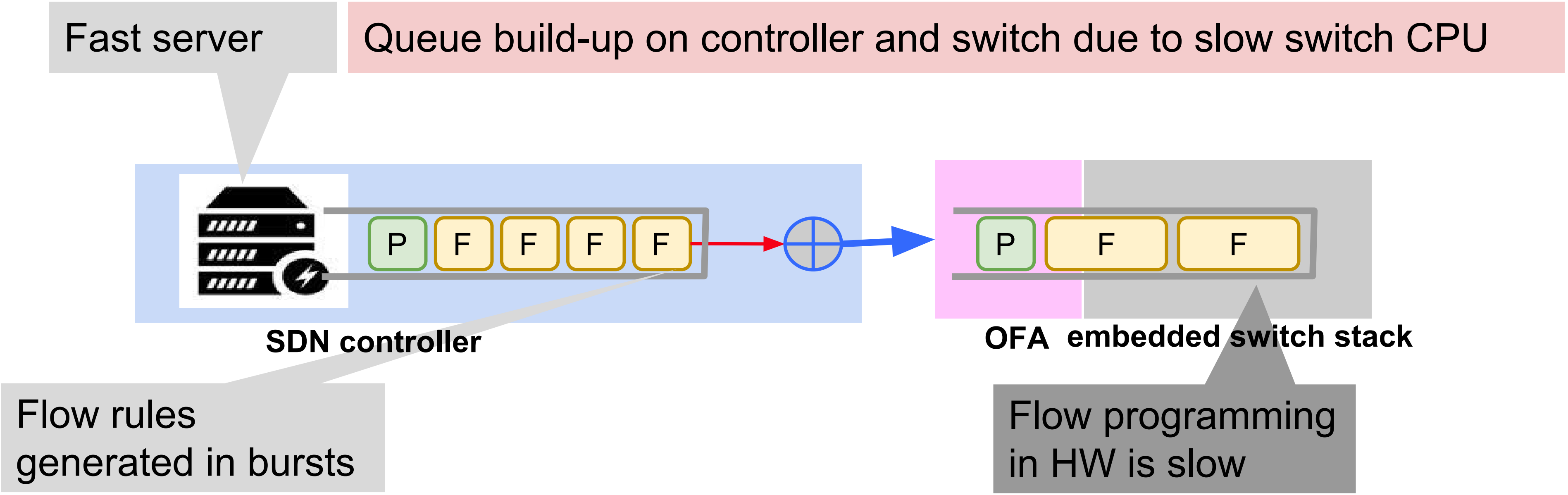
Fast server

Queue build-up on controller and switch due to slow switch CPU



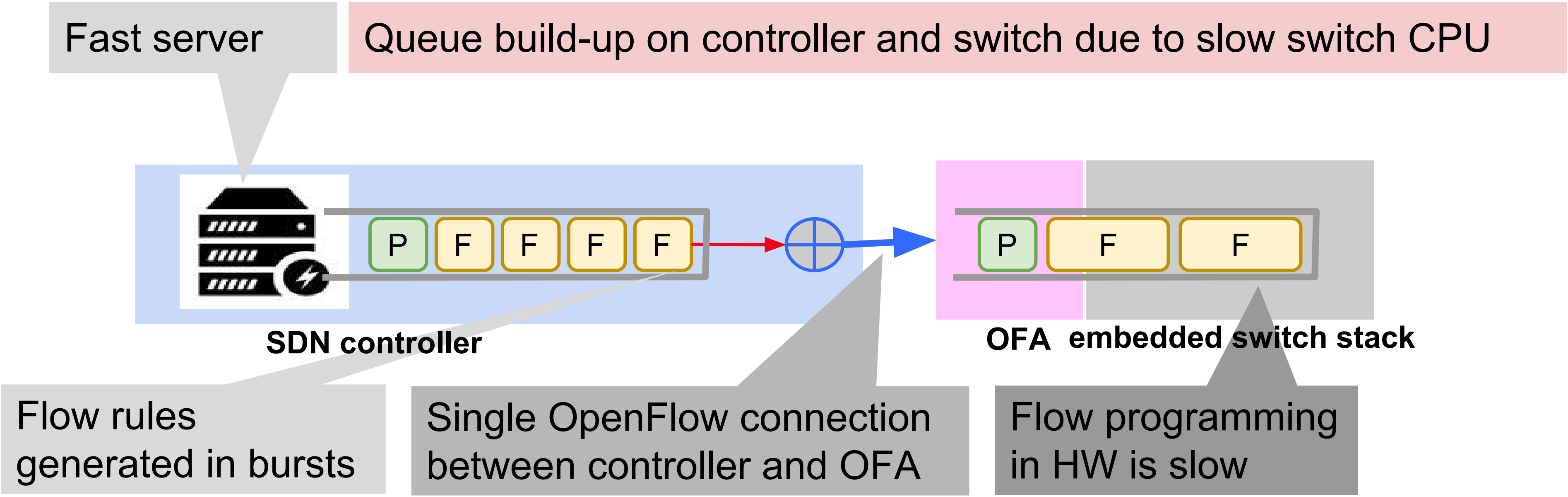
Flow rules
generated in bursts

Messages Backlogged and Delayed!



Messages Backlogged and Delayed!

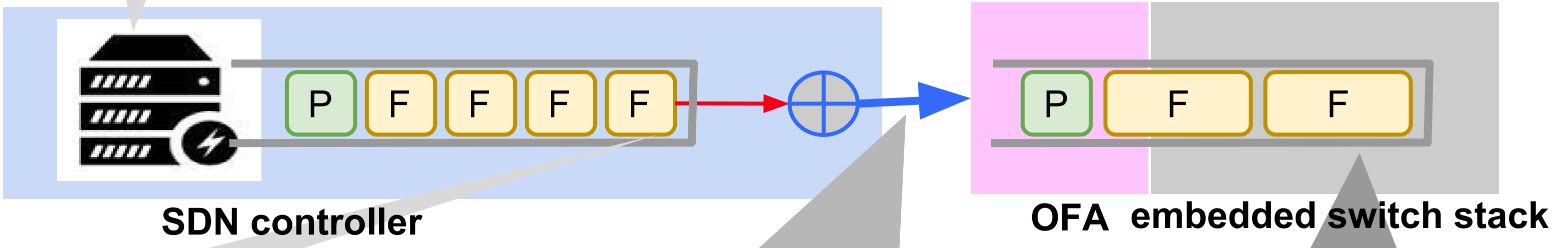
Google



Messages Backlogged and Delayed!

Fast server

Queue build-up on controller and switch due to slow switch CPU



Flow rules generated in bursts

Single OpenFlow connection between controller and OFA

Flow programming in HW is slow

Flow rules cause HOL blocking for packets

packets delayed

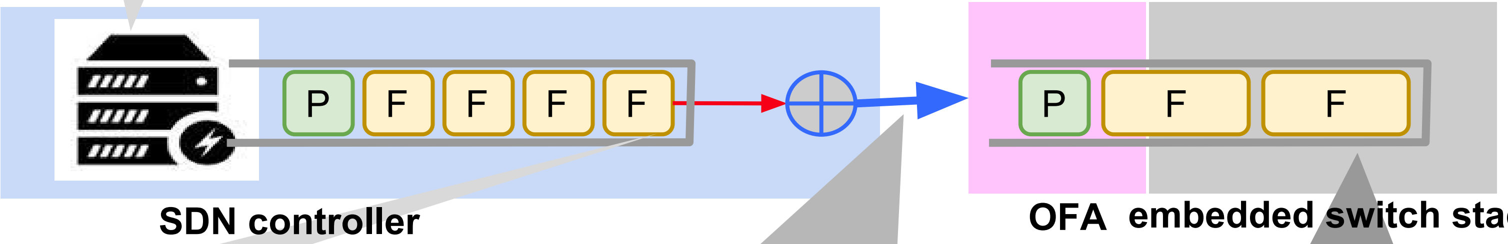
protocols timeout
reconvergence produces more flow rules

Messages Backlogged and Delayed!

Google

Fast server

Queue build-up on controller and switch due to slow switch CPU



Flow rules
generated in bursts

Single OpenFlow connection
between controller and OFA

Flow programming
in HW is slow

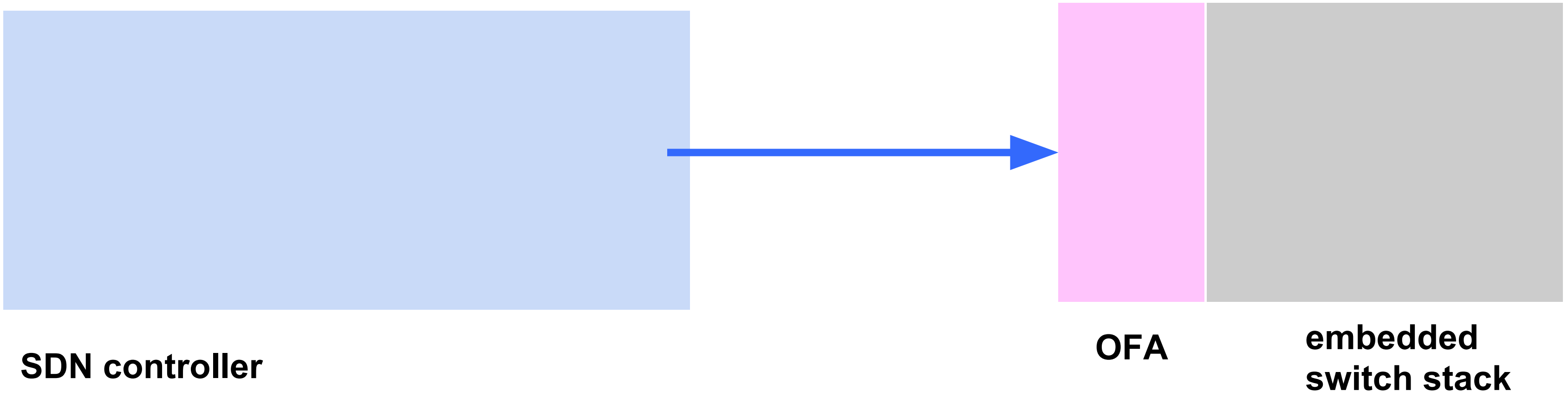
Flow rules cause HOL
blocking for packets

packets
delayed

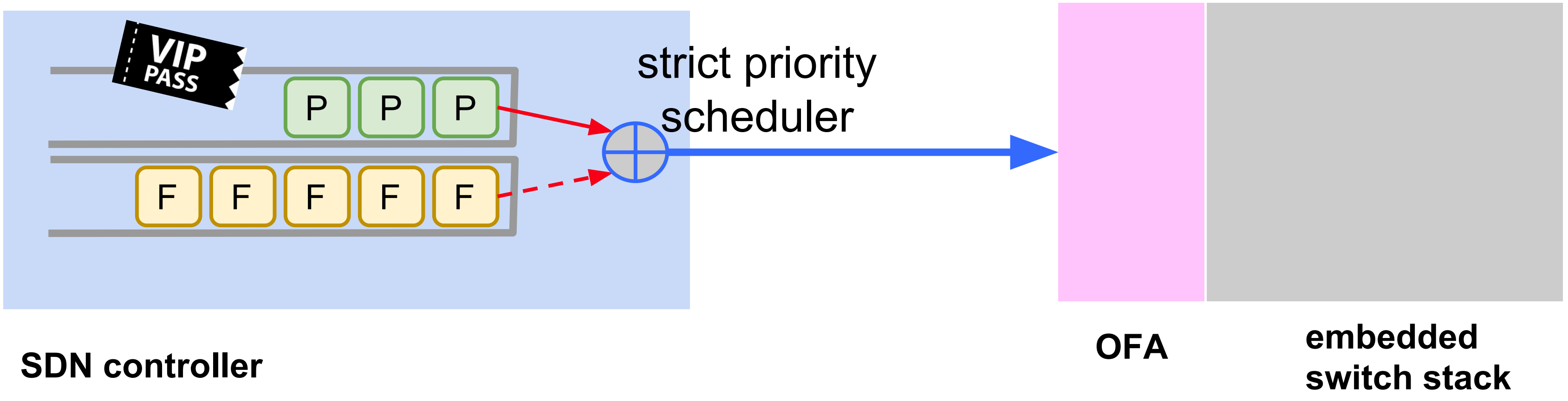
protocols timeout
reconvergence produces more flow rules

Vicious Cycle of Protocol Instability!!!

Lesson: Mitigation with Flow Control

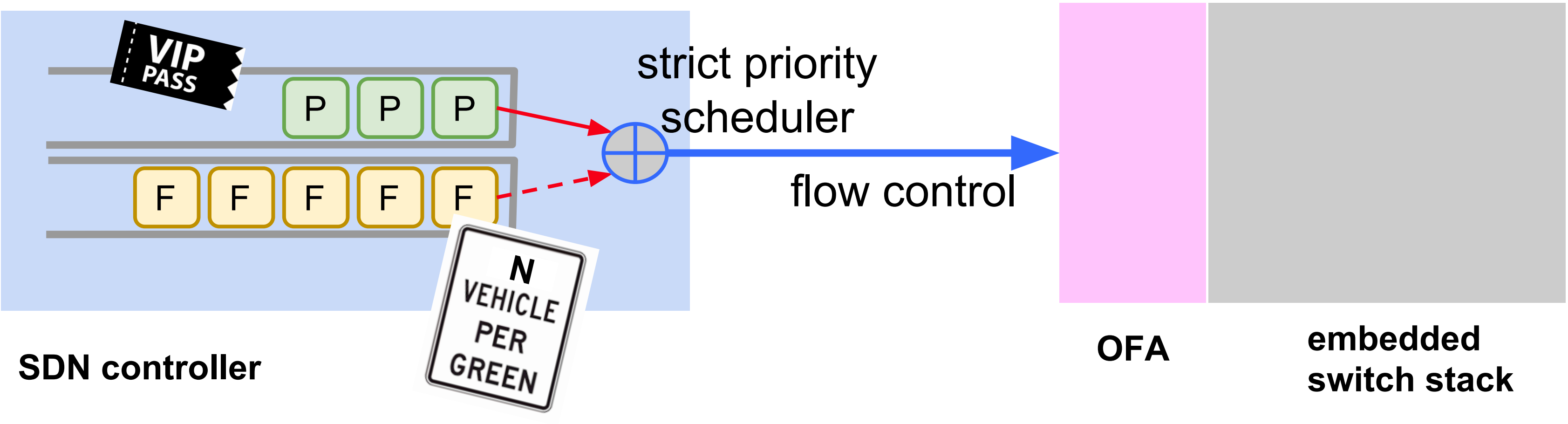


Lesson: Mitigation with Flow Control



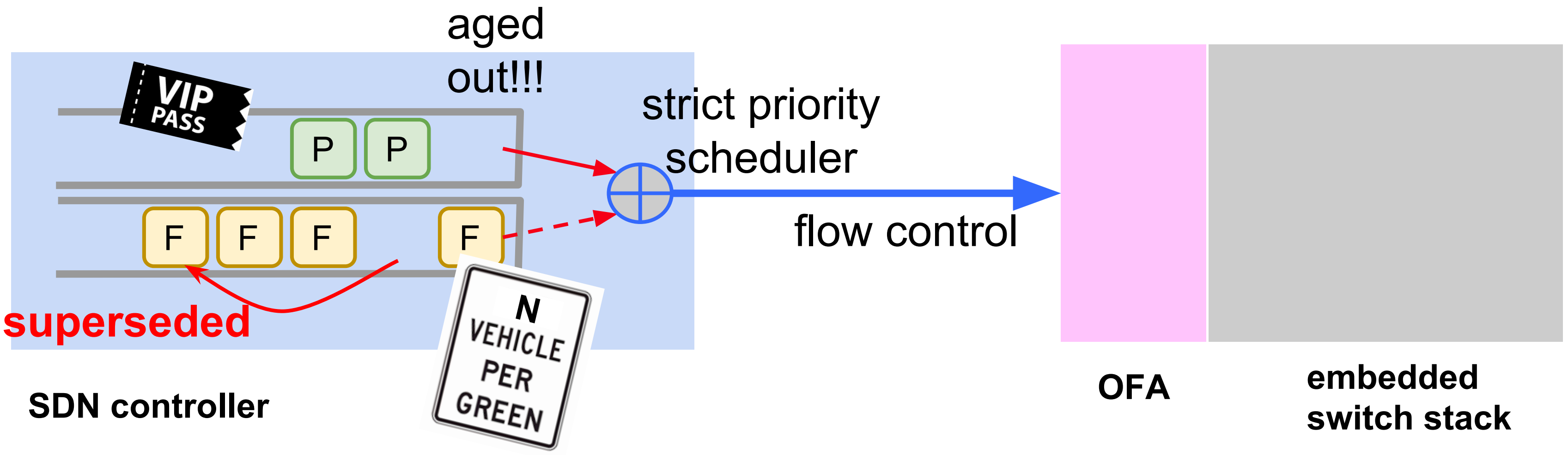
- Separate queue for packet IO and flow request
- Strict priority for packet IO over flow programming

Lesson: Mitigation with Flow Control



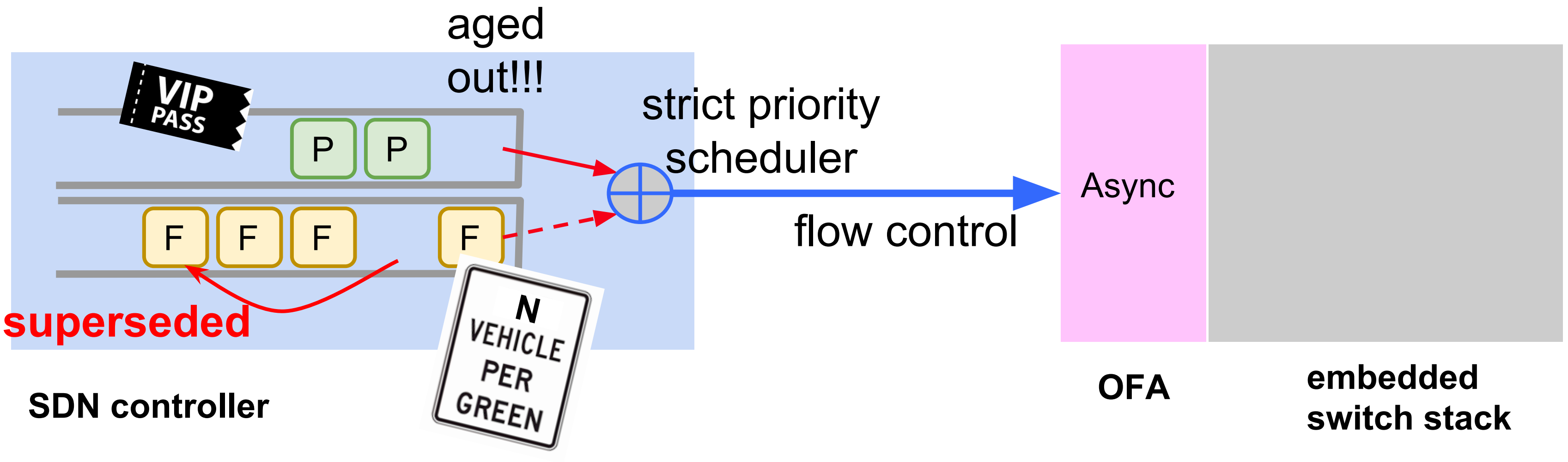
- Separate queue for packet IO and flow request
- Strict priority for packet IO over flow programming
- Limit queue depth in OFA: token based flow control

Lesson: Mitigation with Flow Control



- Separate queue for packet IO and flow request
- Strict priority for packet IO over flow programming
- Limit queue depth in OFA: token based flow control
- Systematics queue drop discipline

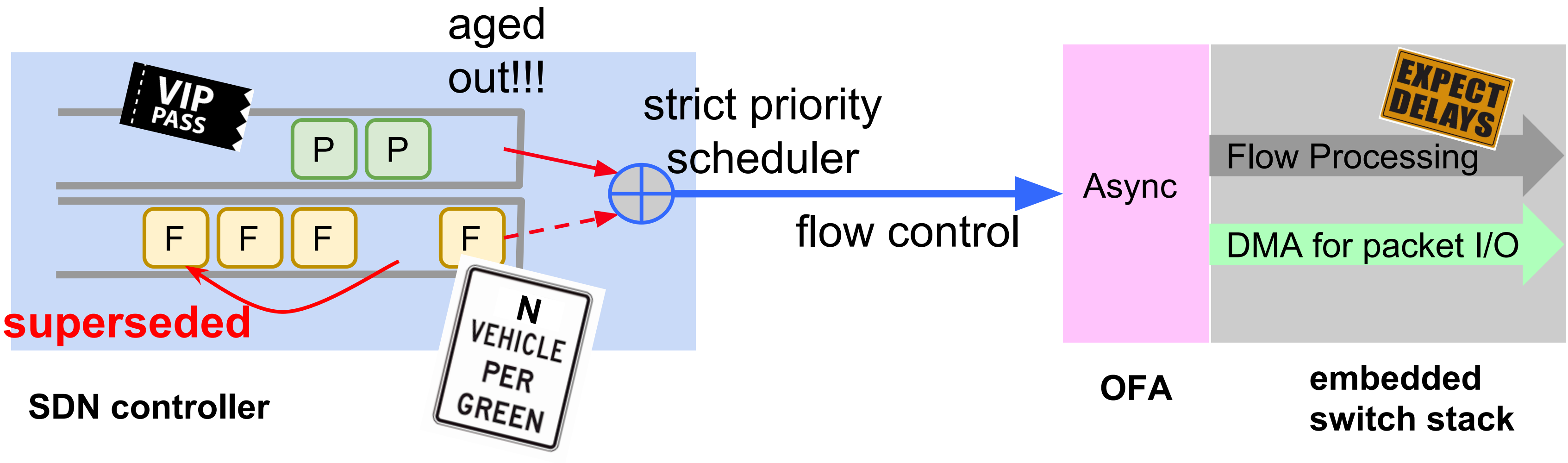
Lesson: Mitigation with Flow Control



- Separate queue for packet IO and flow request
- Strict priority for packet IO over flow programming
- Limit queue depth in OFA: token based flow control
- Systematics queue drop discipline

- Asynchronous OFA

Lesson: Mitigation with Flow Control



- Separate queue for packet IO and flow request
- Strict priority for packet IO over flow programming
- Limit queue depth in OFA: token based flow control
- Systematics queue drop discipline

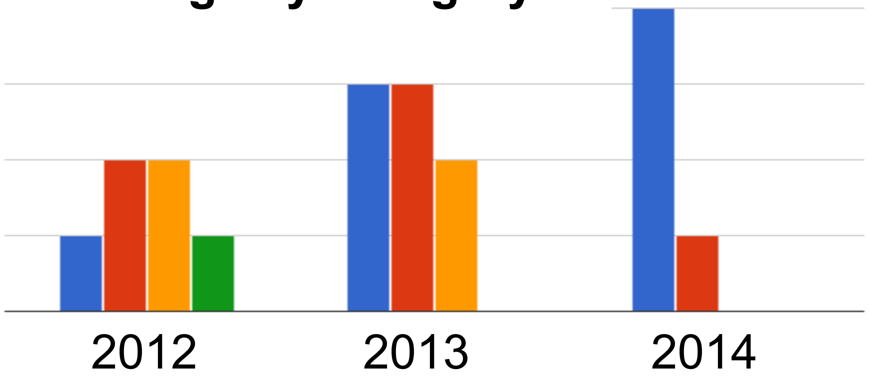
- Asynchronous OFA
- Packet IO out of flow processing pipeline

Lesson on Availability

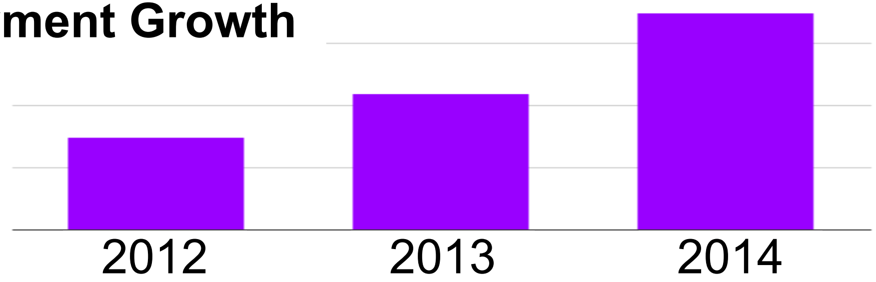
Outages!!!

Google

Postmortem Bugs by Category



Deployment Growth

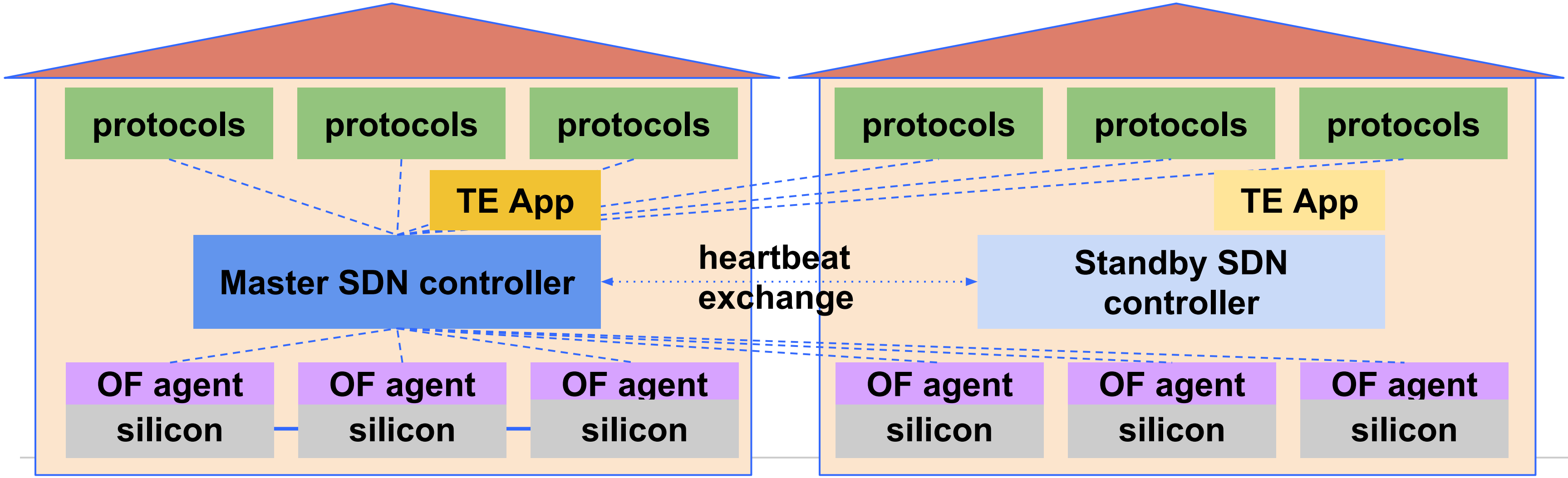


Control Plane Connectivity: Mastership

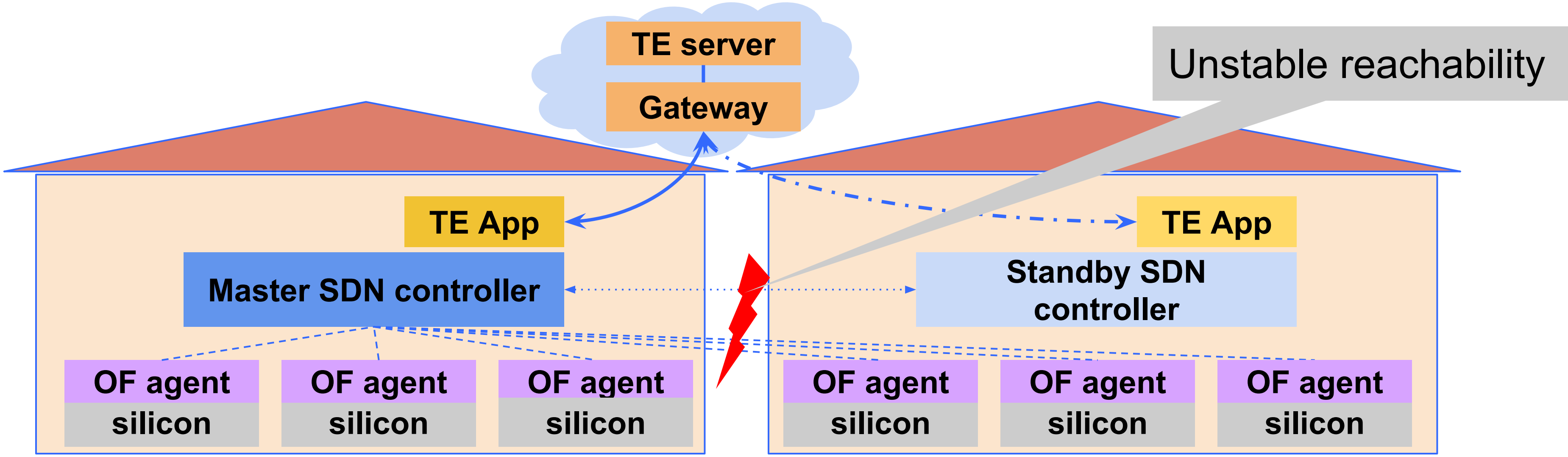
Google

Initial naive design:

- Symmetry between buildings
- Each building can run independently, even if the other one is down
- N+1 controller redundancy sufficient for upgrades, failures etc.



Control Network: Unstable Mastership



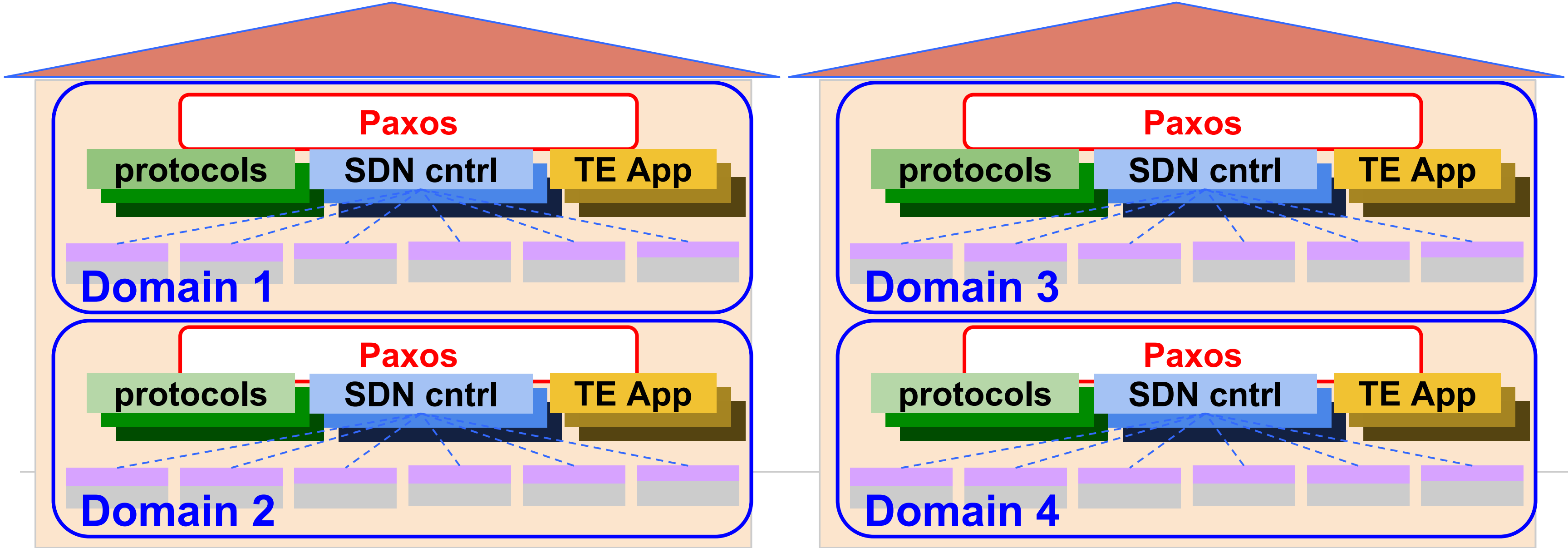
- Both controllers declare mastership:
  - Gateway and OFAs can observe mastership flapping frequently
  - Declared master has partial reachability to switches
- Reported topology changes, pathing changes, flow programming fails

Non-transitive reachability => Packets dropped!!

Lesson: Robust Control Reachability

Google

- Multiple independent domains per site: connected only through dataplane
  - Each domain is unit for safe modular upgrade and maintenance
- Paxos: quorum-based robust master election *within* each domain
- Also removes single point of failure in each site



Lessons on Scaling

Flat Topology Scales Poorly

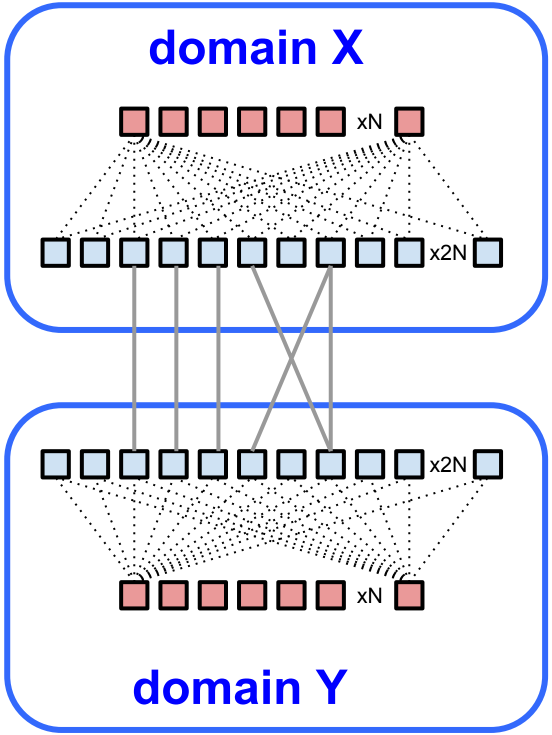
Google

- As B4 grows: more sites deployed
- As compute per site grows:
  - More capacity required per site

Larger switches OR *more switches*

- Larger switches: loss of large capacity on switch failure
- More switches: more nodes and links to manage
  - ISIS and TE will hit scaling issues, converge too slowly...!!!

Lesson: Hierarchical Topology

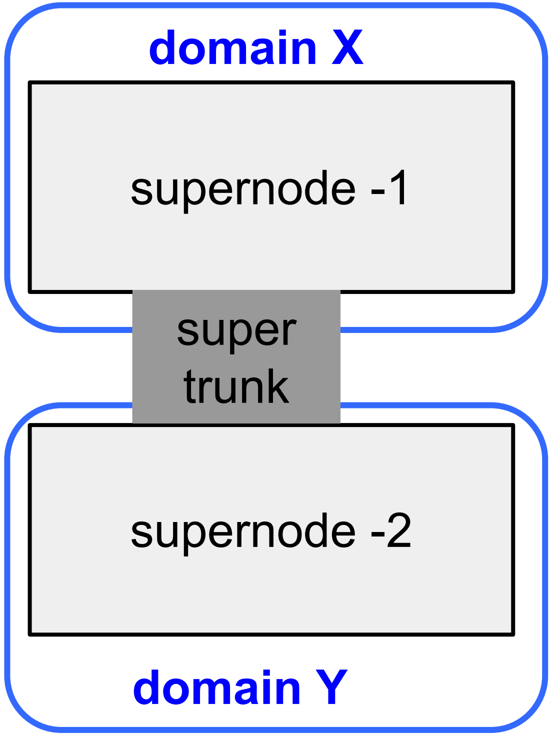


physical topology: domain controller view

Best of both worlds with SDN

- Topology abstractions by domain controllers
  - Supernode: tightly connected nodes/switches
  - Supertrunks: links between super nodes
- Domain controllers compute
  - intra-domain routing
  - impairment due to internal failure

Lesson: Hierarchical Topology



abstract topology: global controller view

Best of both worlds with SDN

- Topology in terms of supertrunk capacity
- TE and ISIS/BGP work on supernodes

Reduces global controller-visible topology complexity by over 100x

Conclusions

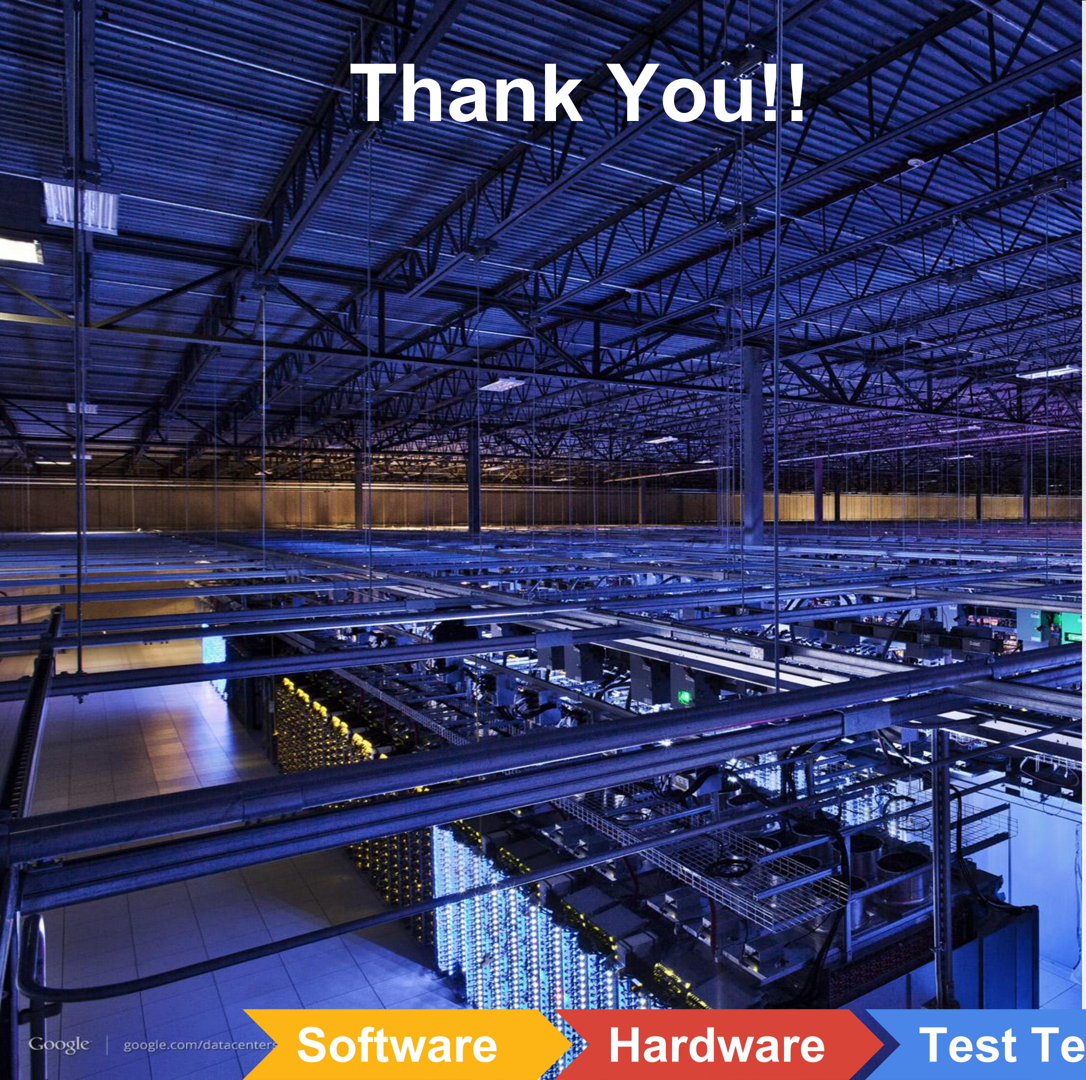
Google

- 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

References

- Upward Max Min Fairness: **INFOCOM 2012**
- B4: Experience with a Globally-Deployed Software Defined WAN: **SIGCOMM 2013**
- Bandwidth Enforcer: Flexible Hierarchical Bandwidth Allocation for WAN Distributed Computing: **SIGCOMM 2015**

Thank You!!



Google Platforms Networking

Hiring

- Interns
- Full time engineers

Locations worldwide:

- Mountain View
- New York
- Sydney

Inspiration and creativity to build Google’s infrastructure:

- Scale that gives the edge
- Research turned into real life production solution

planet

Software | Hardware | Test Technology