Express Backbone – scaling challenges

Moving Fast with Facebook’s Long-Haul Network

Henry Kwok
Software Engineer, Facebook

Mikhail Vasilyev
Production Engineer, Facebook

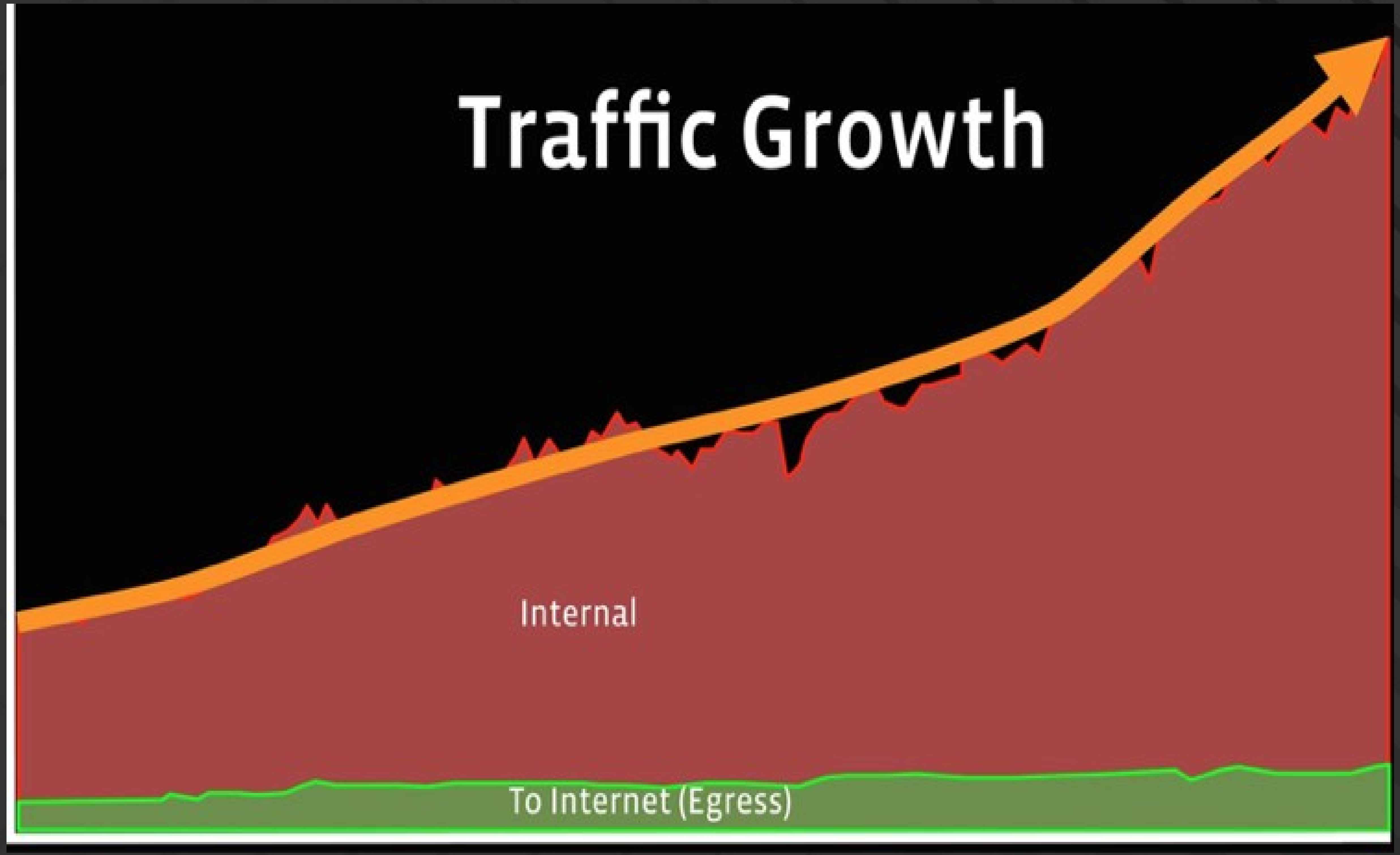
What is Express Backbone?

- Facebook’s own SDN backbone

- **Motivations**
- Network Design
- Traffic Engineering
- Lessons Learned

Traffic Growth

- Machine-to-machine traffic has been growing rapidly
- Fueled by videos and data analytics.
- Vertical scaling cannot meet future demands

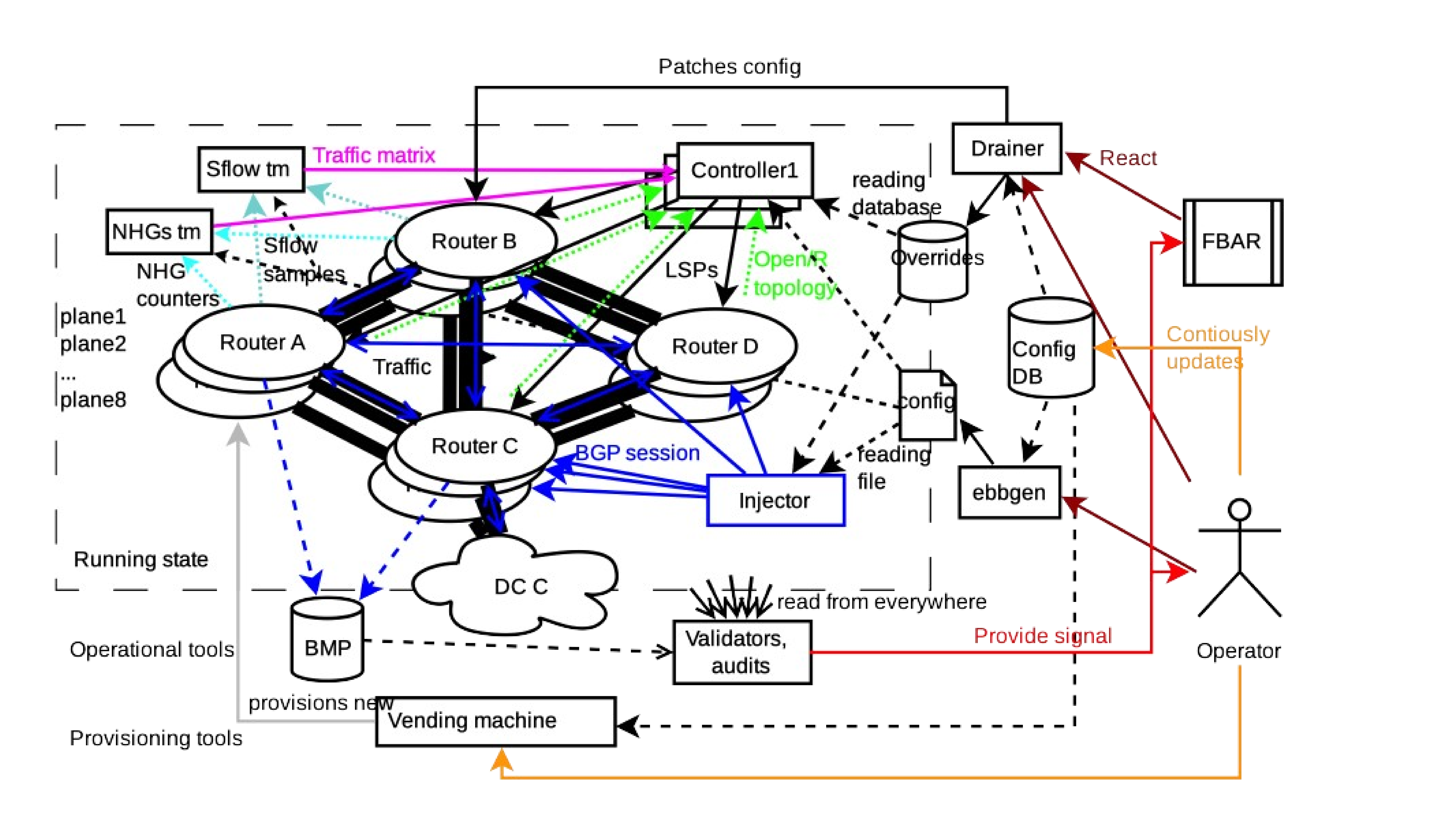


Flexibility

- More choices than RSVP-TE
- Ability to experiment and iterate
- Moving fast

- Motivations
- **Network Design**
- Traffic Engineering
- Lessons Learned

Network Design - overview

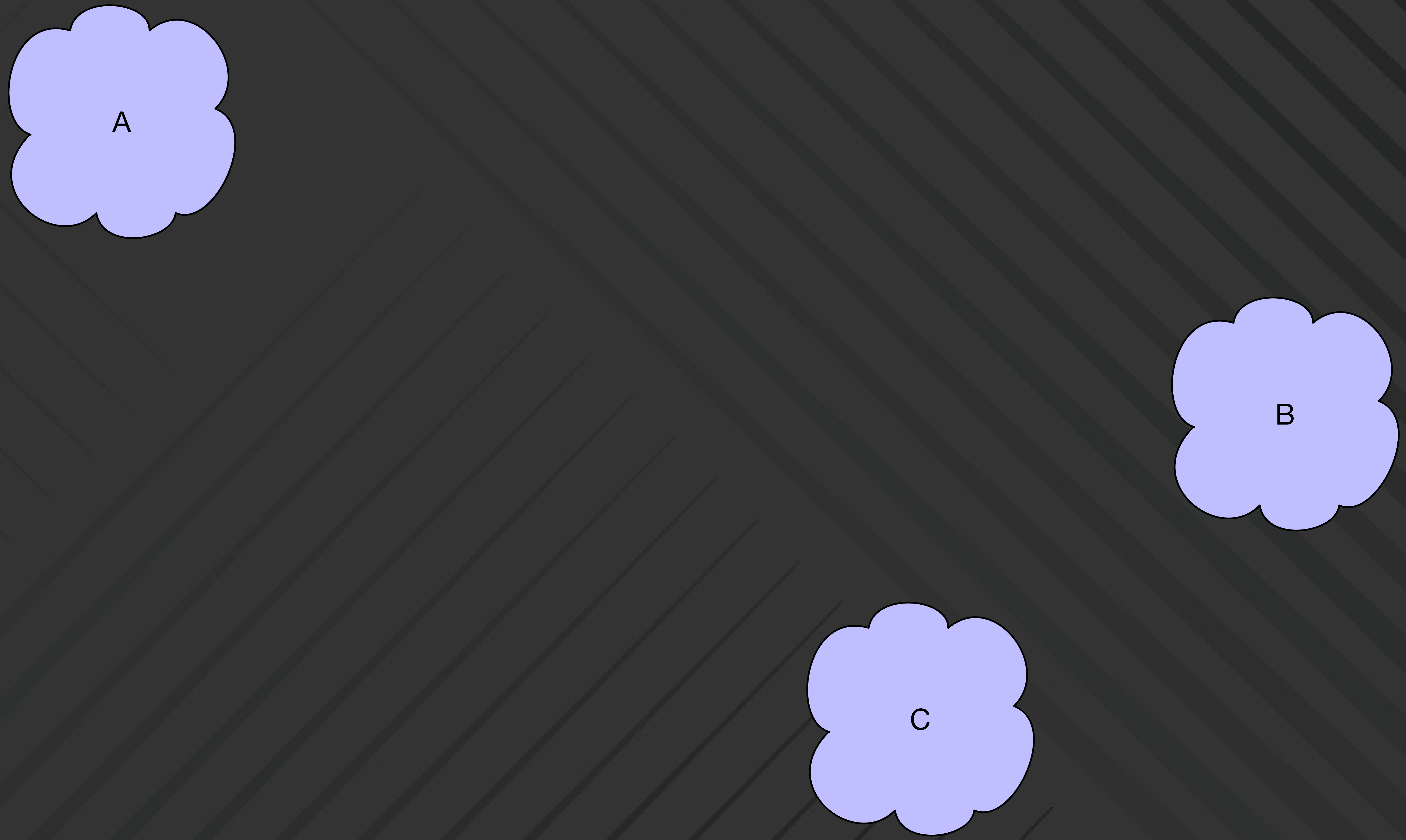


Network Design - overview

Scary?

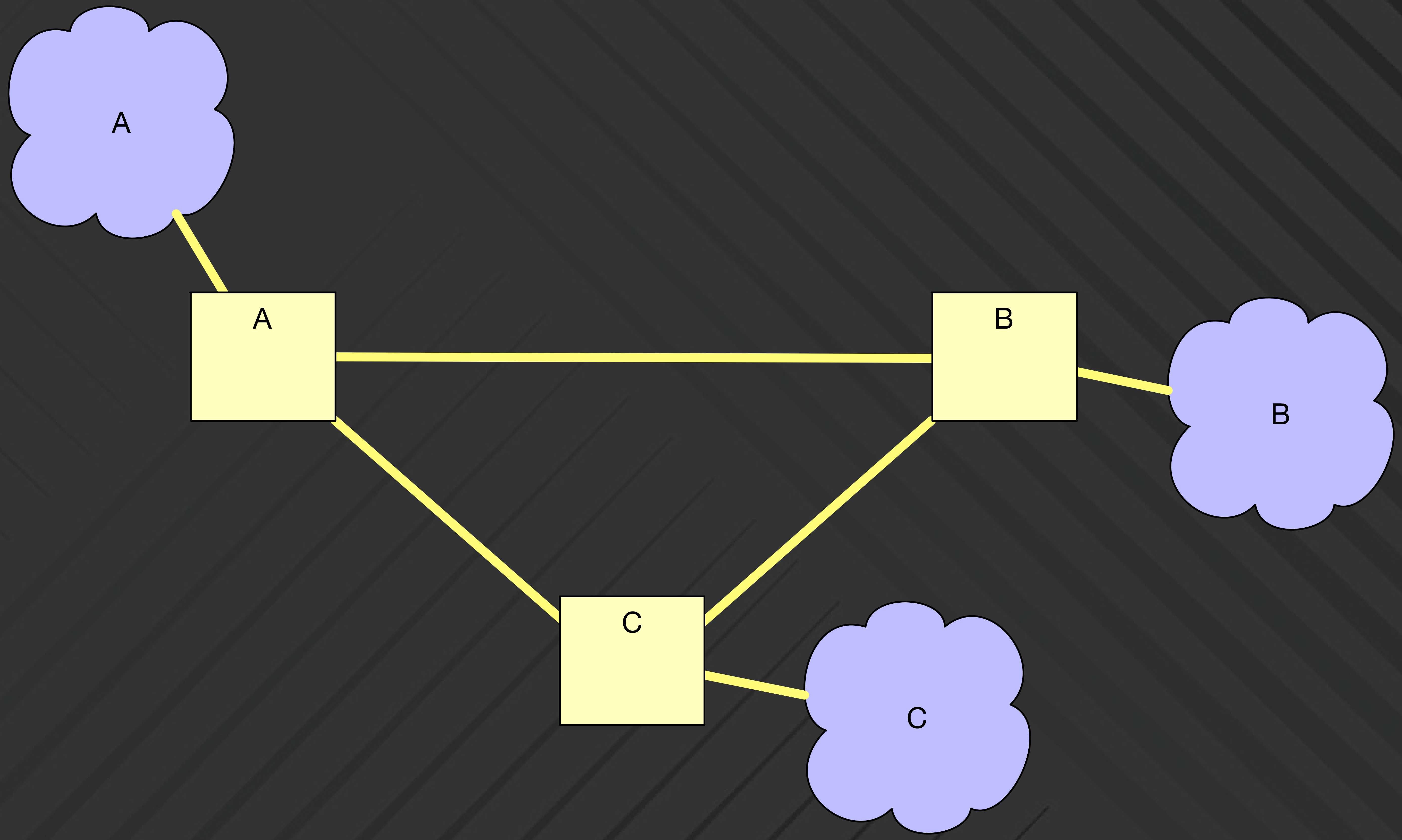
Let’s do it step-by-step

Network Design



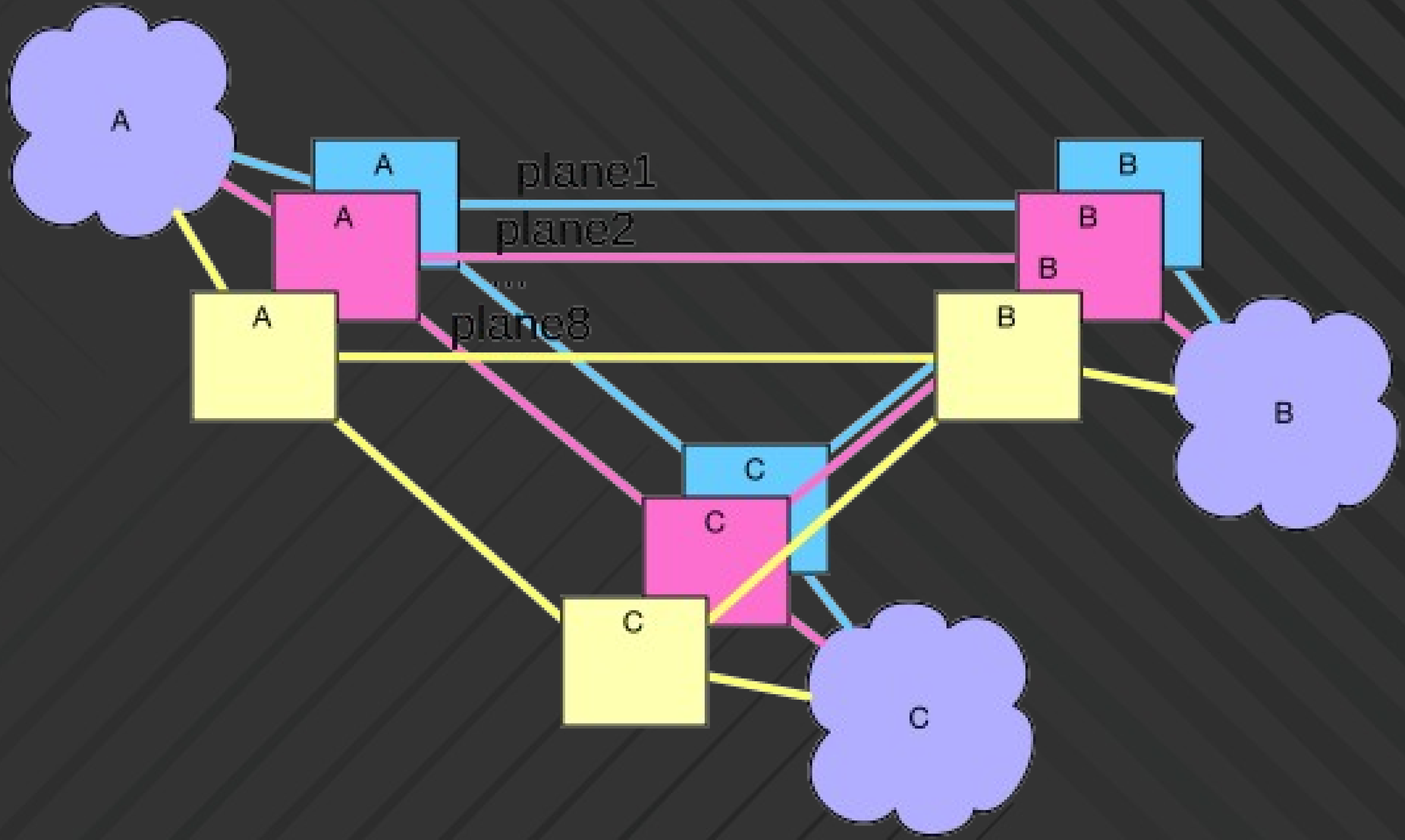
Network Design

- **Commodity switches**



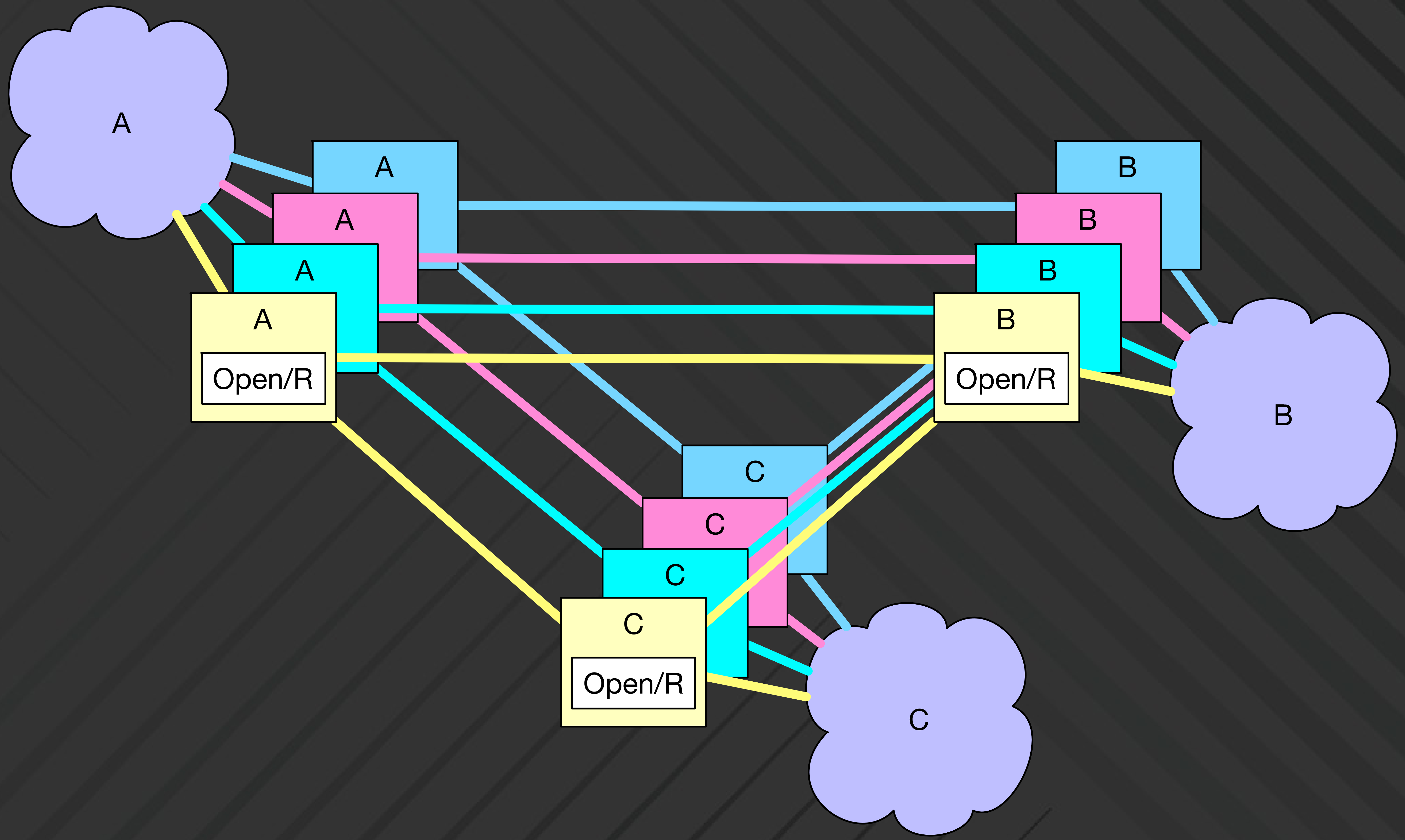
Network Design

- Commodity switches
- **8 parallel forwarding planes**



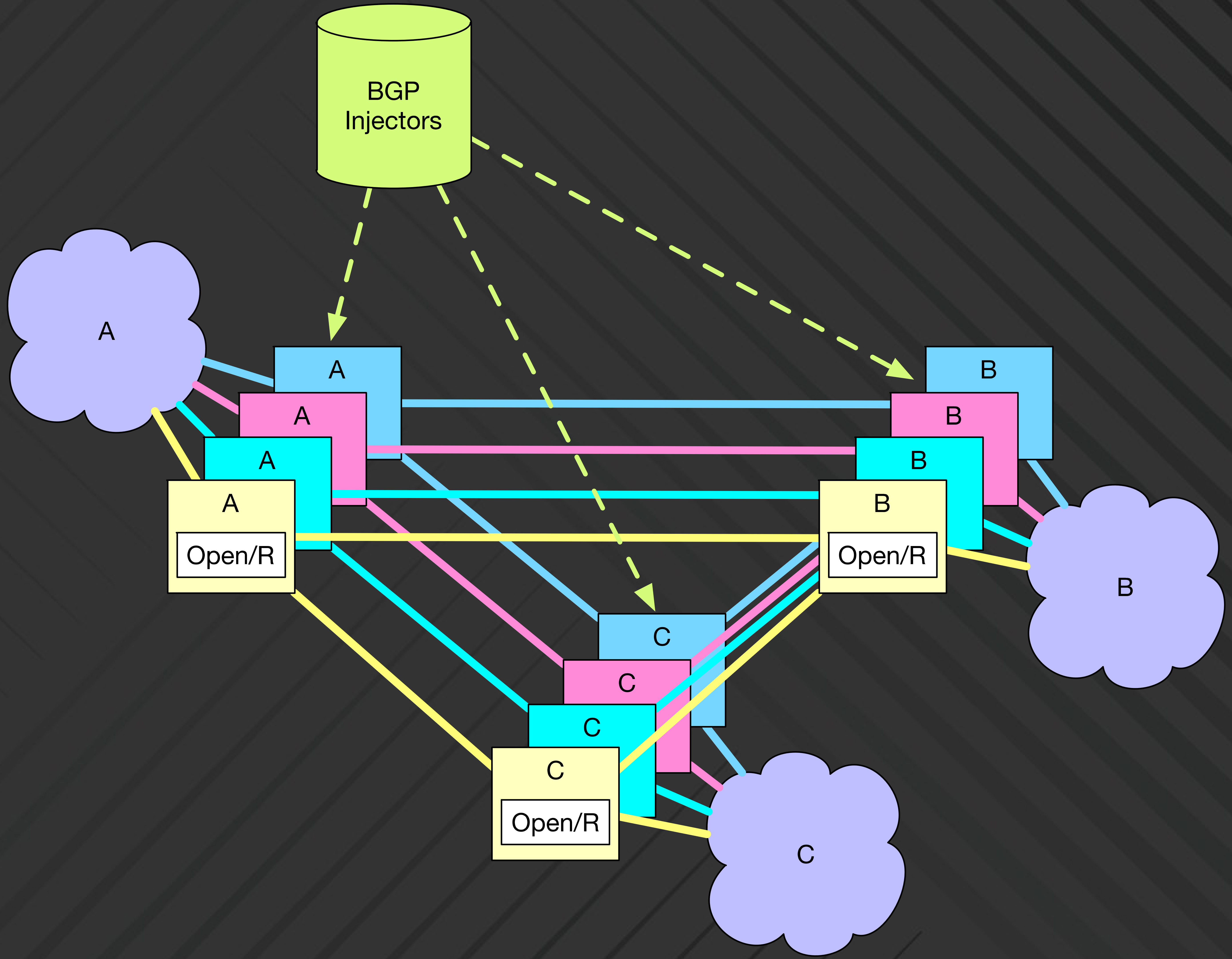
Network Design

- Commodity switches
- 8 parallel forwarding planes
- **Open/R**



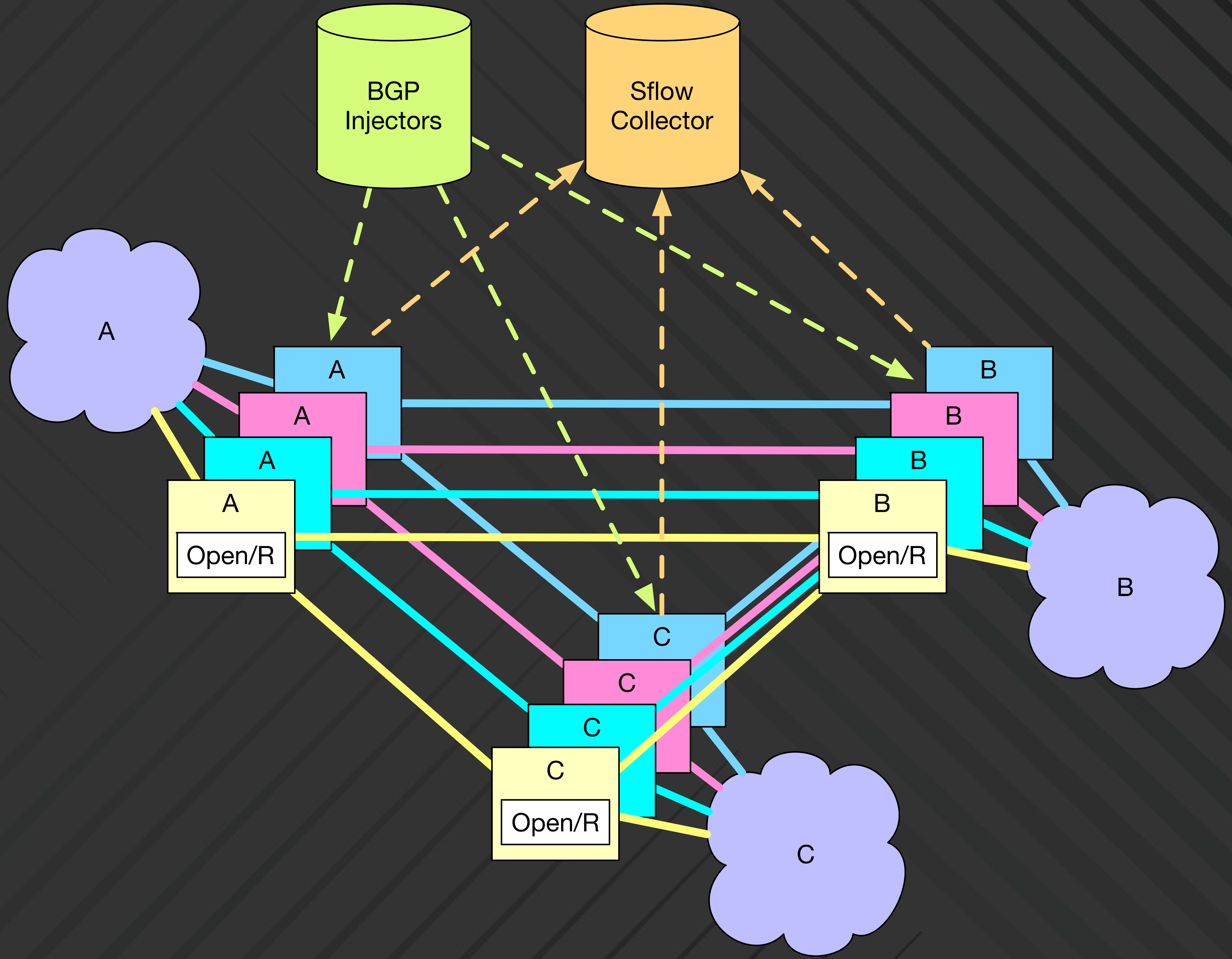
Network Design

- Commodity switches
- 8 parallel forwarding planes
- Open/R
- **BGP injection**



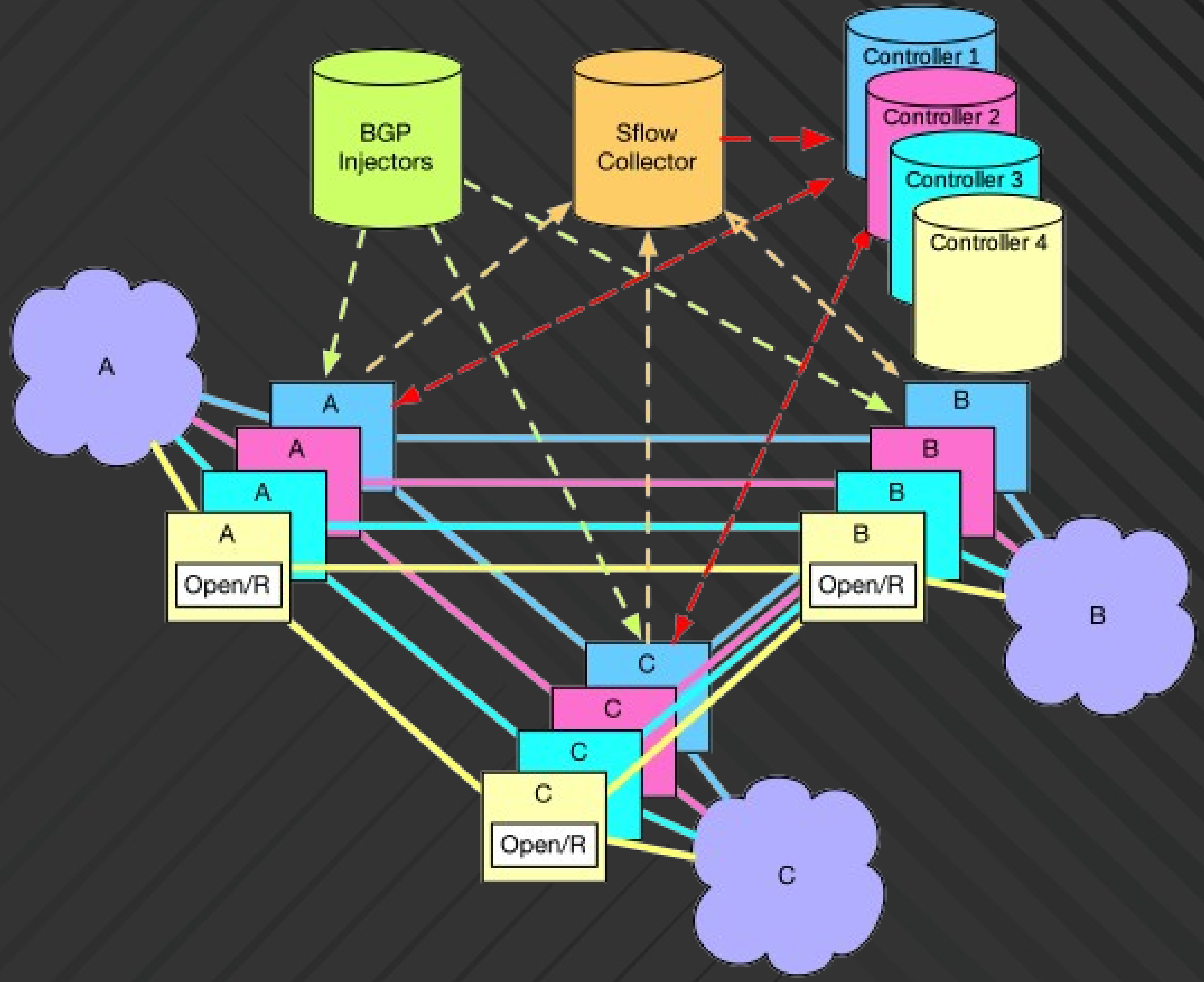
Network Design

- Commodity switches
- 8 parallel forwarding planes
- Open/R
- BGP injection
- **Sflow collector**



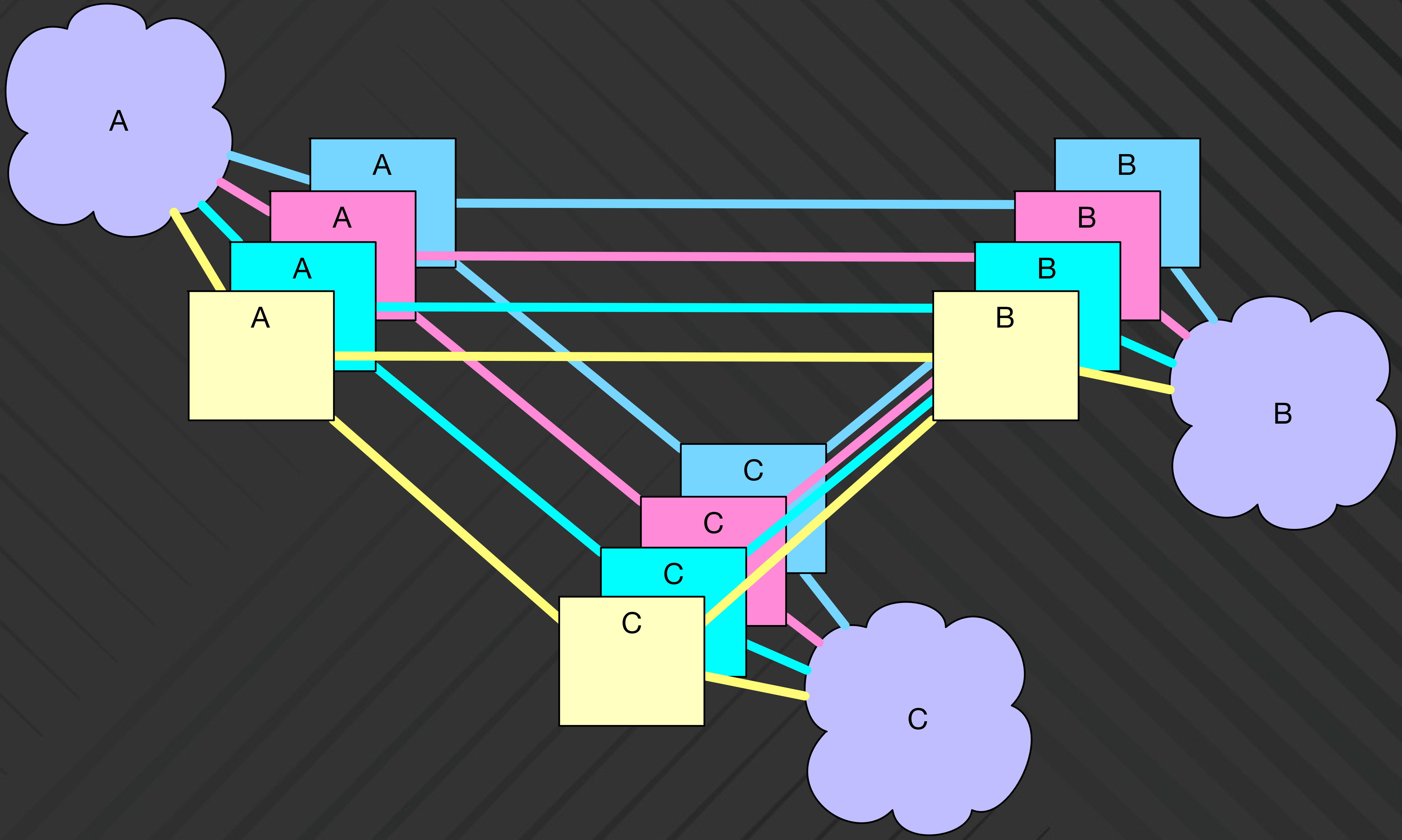
Network Design

- Commodity switches
- Four parallel forwarding planes
- Open/R
- BGP injection
- Sflow collector
- **Traffic-engineering controller**



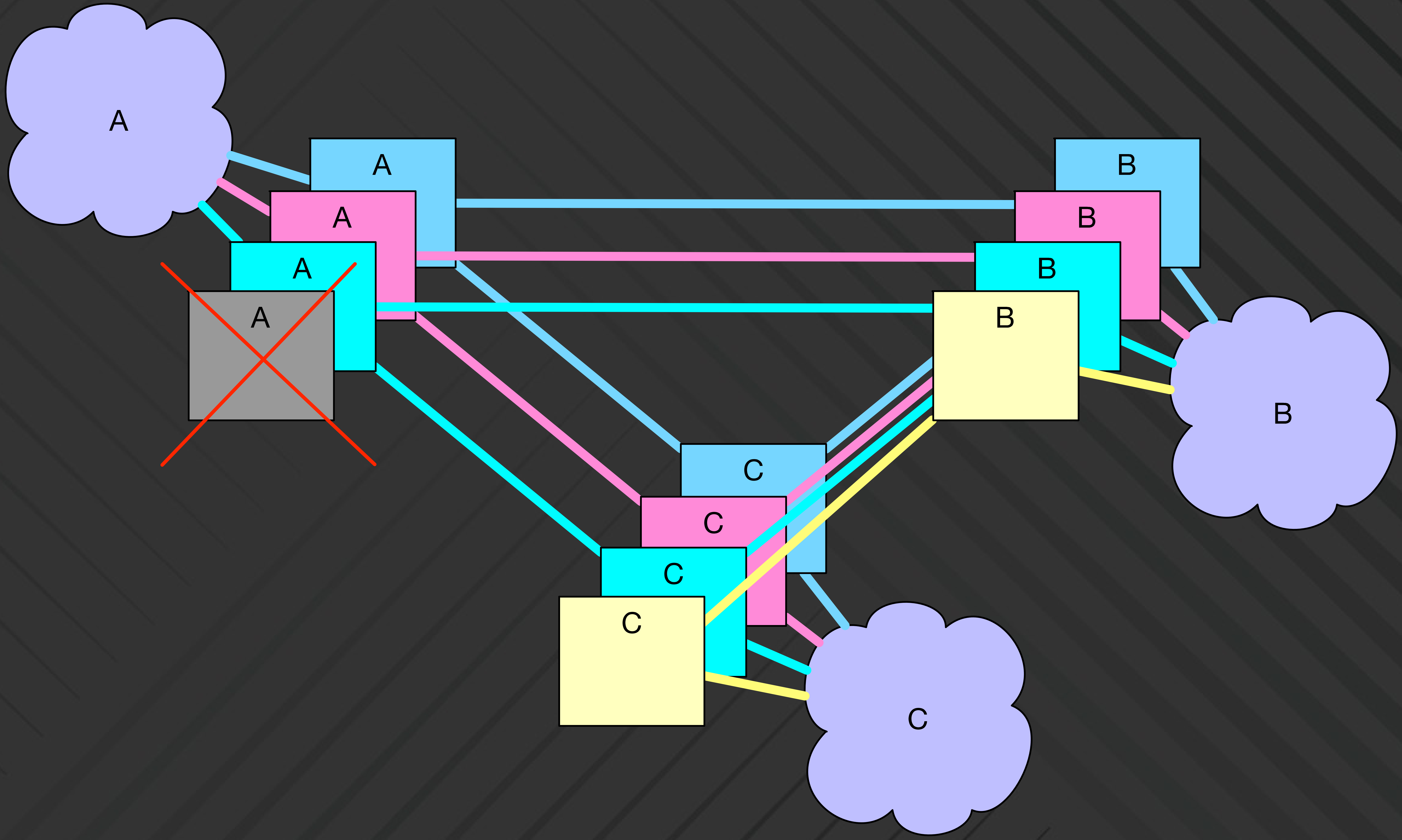
Parallel Forwarding Planes

- **Eight independent and identical forwarding planes**



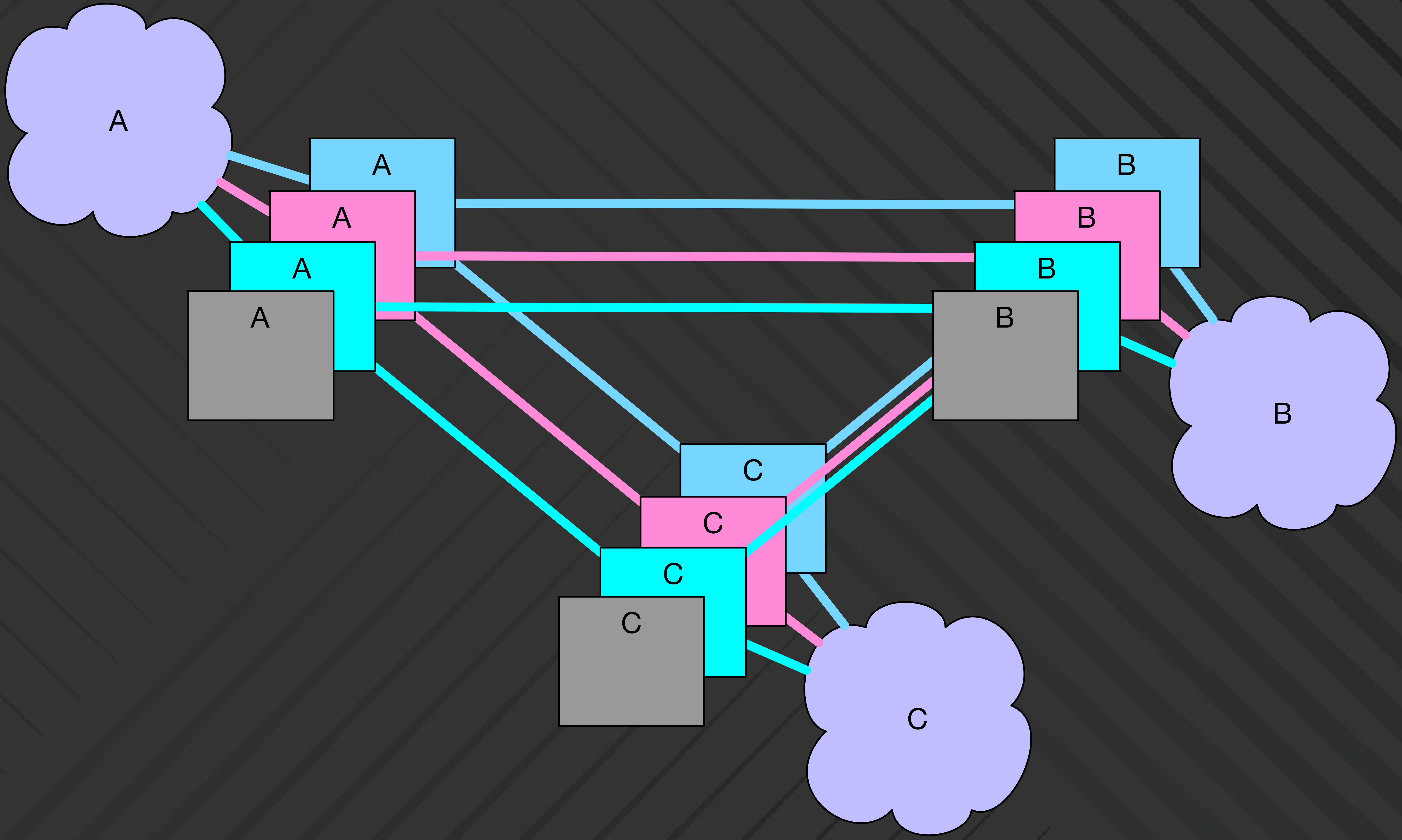
Parallel Forwarding Planes

- Four independent and identical forwarding planes
- **8-way active-active redundancy**



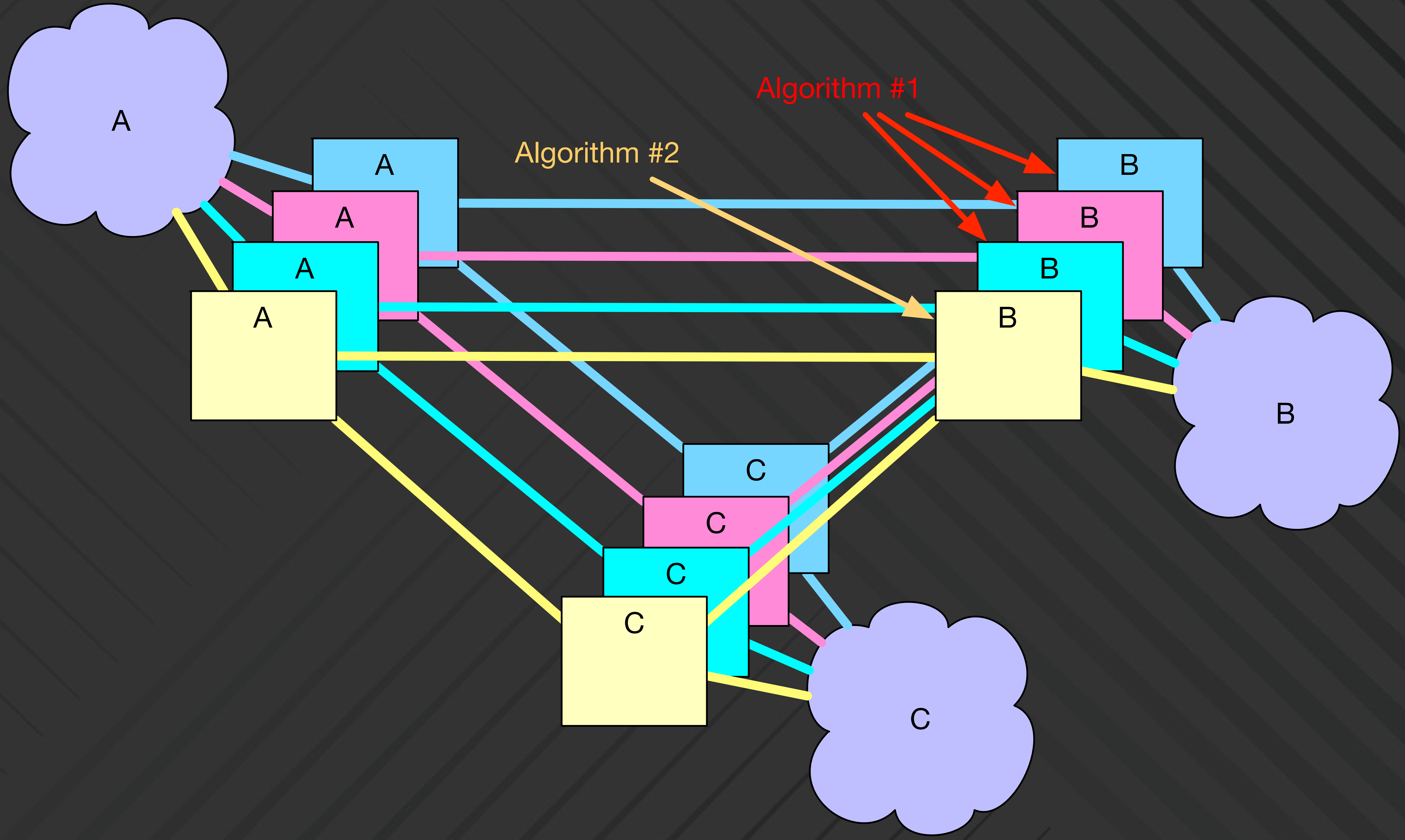
Parallel Forwarding Planes

- Four independent and identical forwarding planes
- 8-way active-active redundancy
- **Incremental changes and canary**



Parallel Forwarding Planes

- Four independent and identical forwarding planes
- 8-way active-active redundancy
- Incremental changes and canary
- **A/B testing**

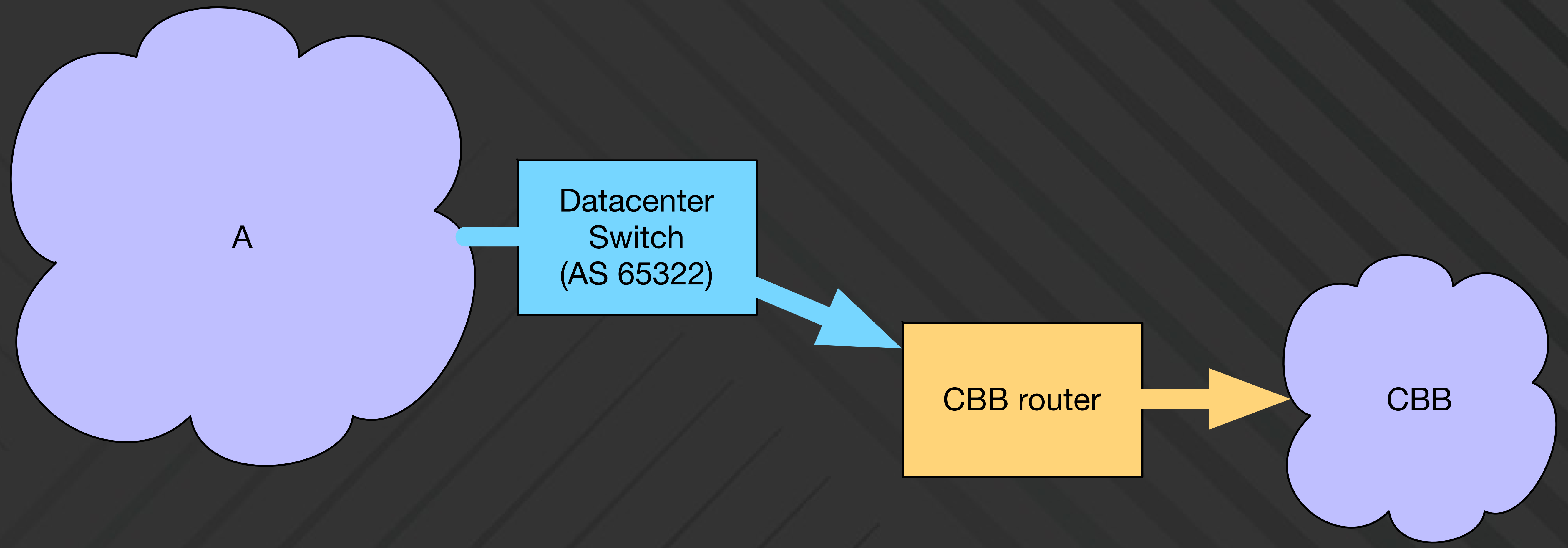


Topology Discovery – Open/R

- Facebook’s routing protocol
- Extensible (e.g. key-value store)
- In-house software → Faster development
- Agent in EBB routers
- Used for LSP failover
- **EBB is the first production network where Open/R is the sole IGP**

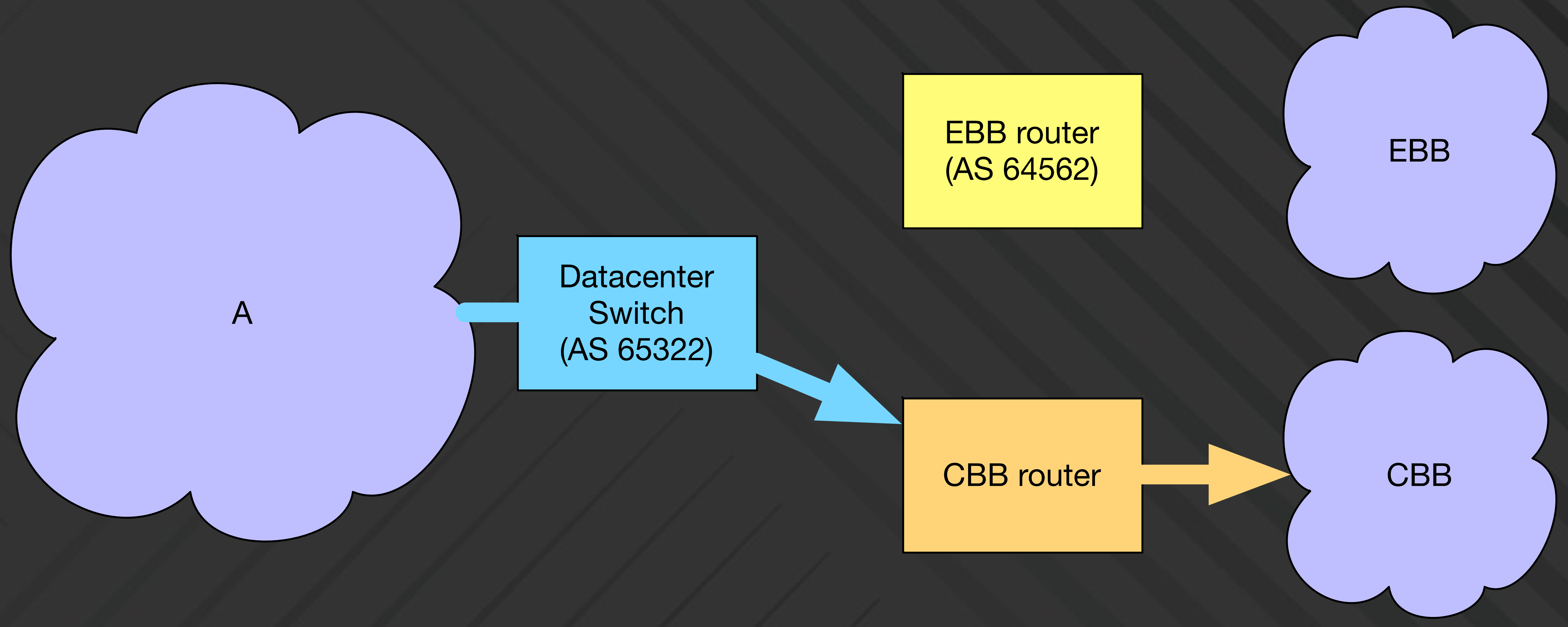
Traffic On-Boarding

- **Incremental on-boarding**



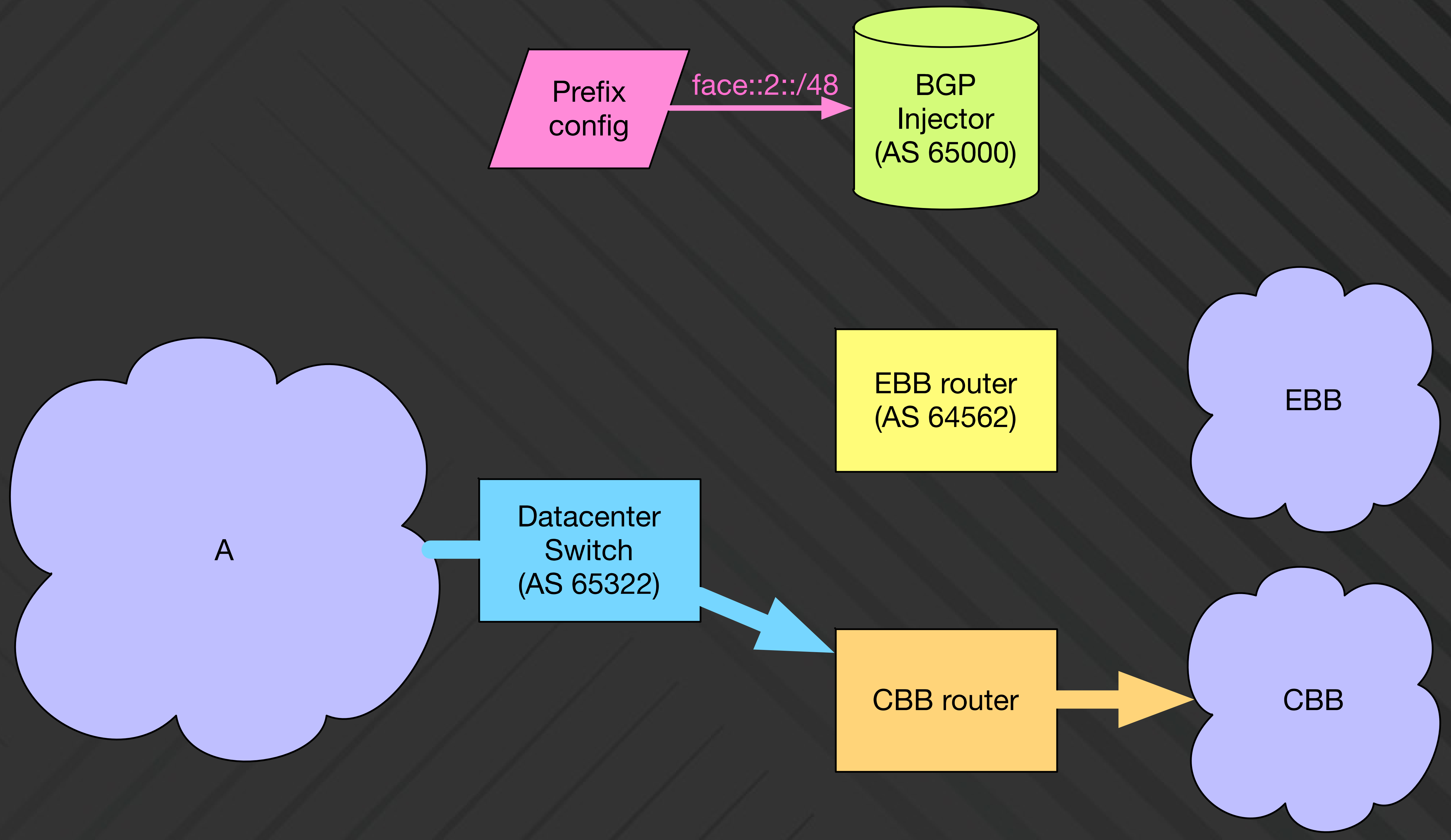
Traffic On-Boarding

- **Incremental on-boarding**



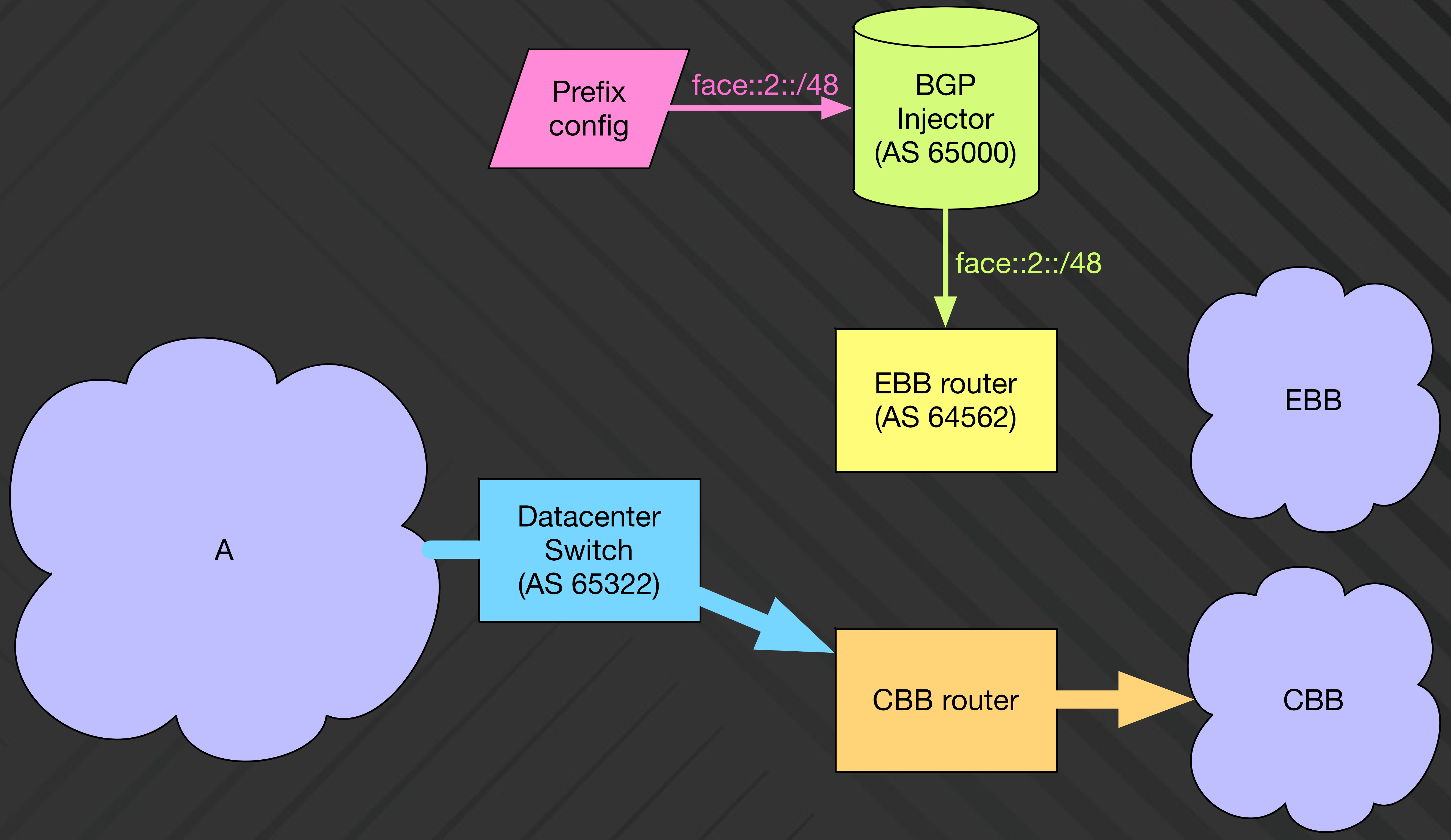
Traffic On-Boarding

- Incremental on-boarding
- **Destination prefix config**



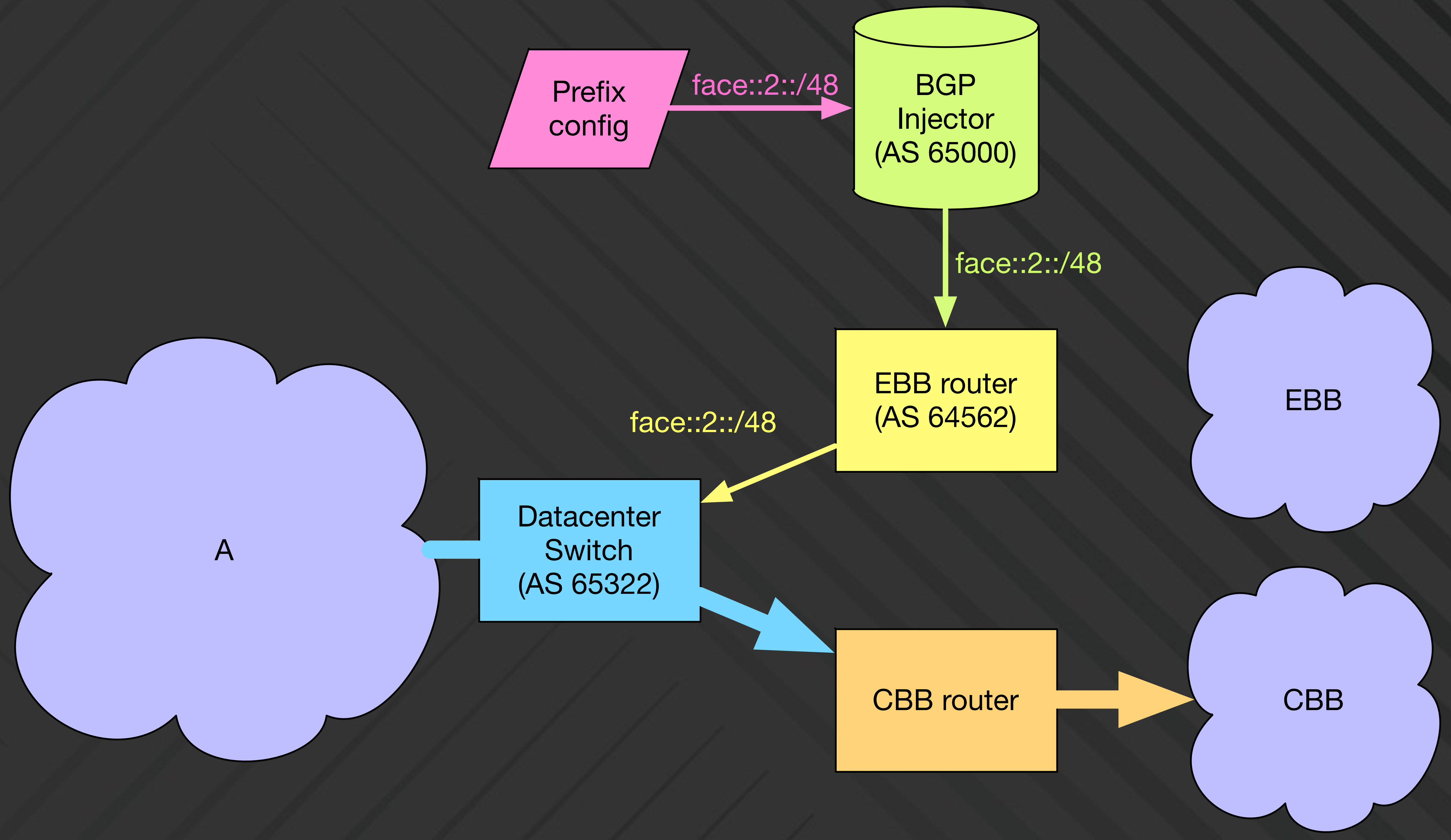
Traffic On-Boarding

- Incremental on-boarding
- Destination prefix config
- **Inject prefixes to EBB routers**



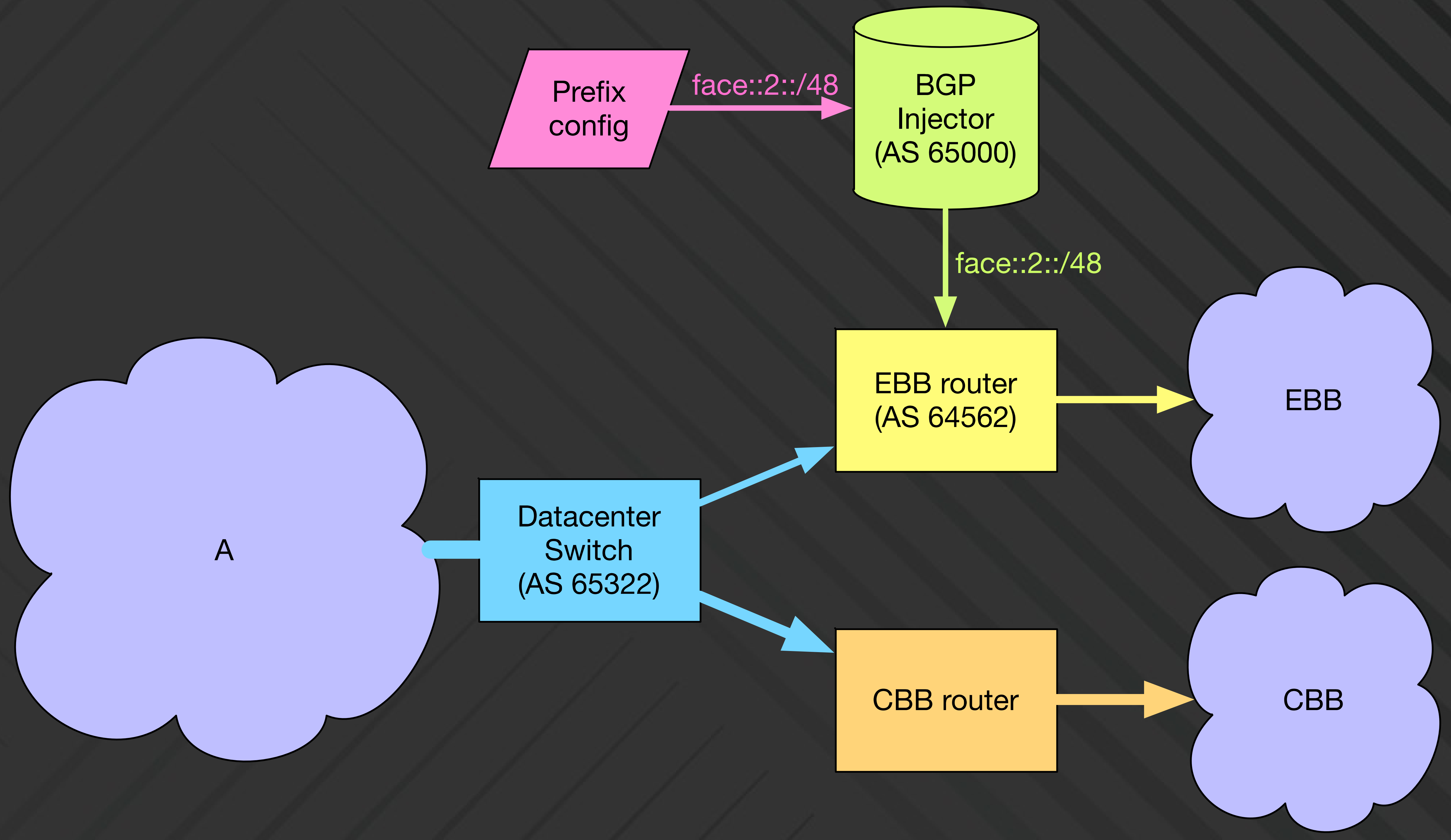
Traffic On-Boarding

- Incremental on-boarding
- Destination prefix config
- **Inject prefixes to EBB routers**



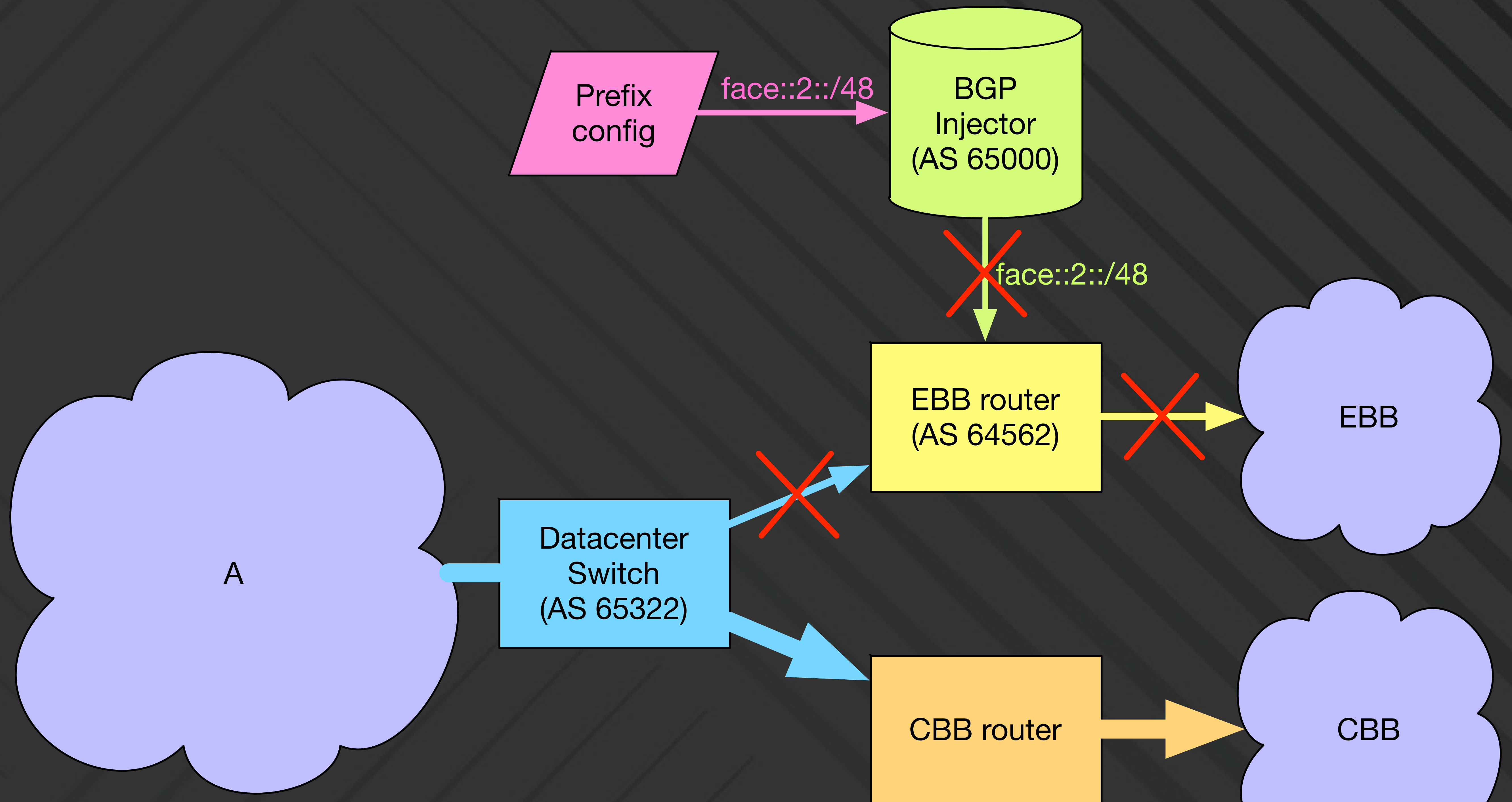
Traffic On-Boarding

- Incremental on-boarding
- Destination prefix config
- **Inject prefixes to EBB routers**



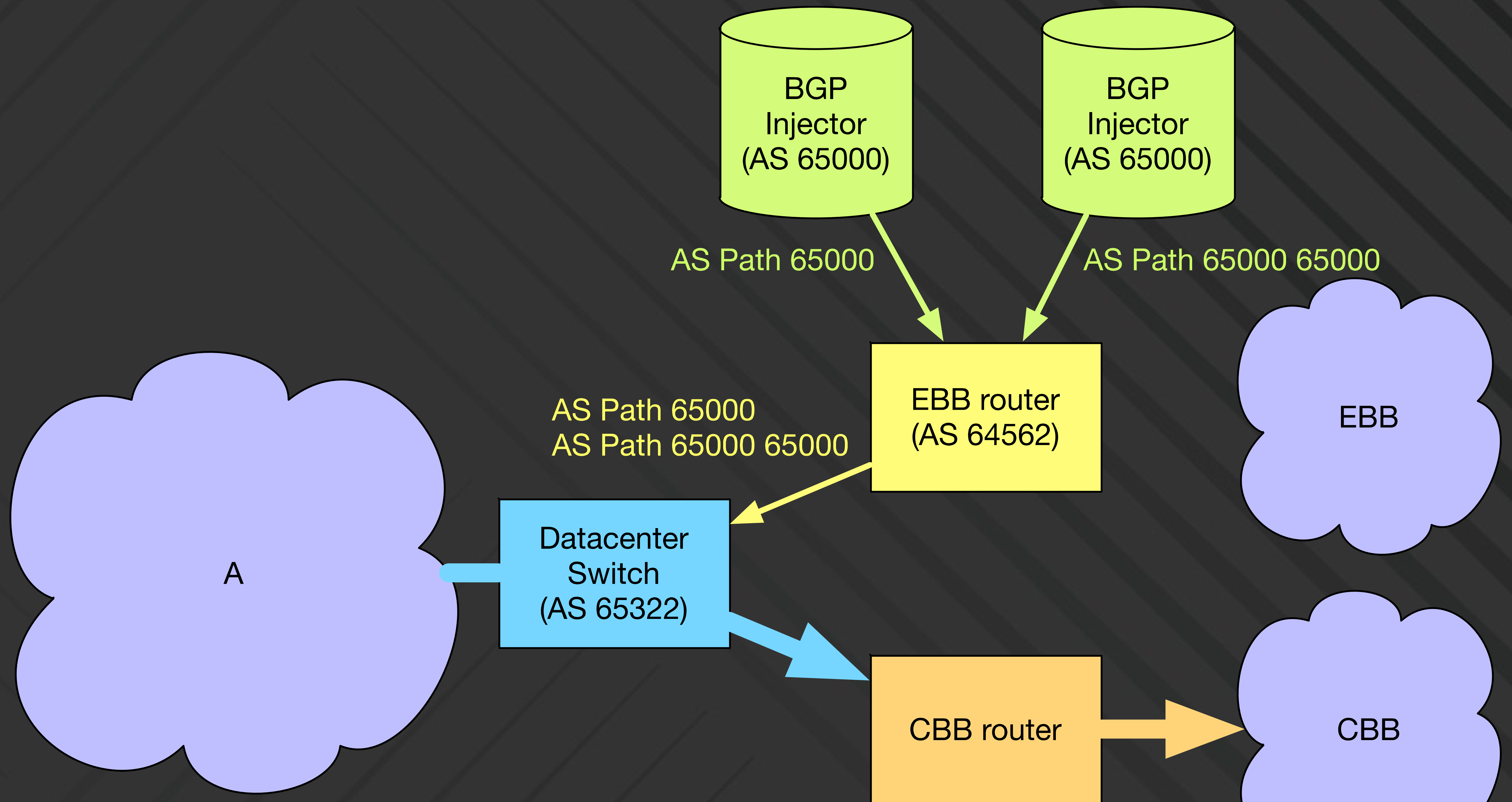
Traffic On-Boarding

- Incremental on-boarding
- Destination prefix config
- Inject prefixes to EBB routers
- **Fall back by withdrawing prefixes**



Traffic On-Boarding

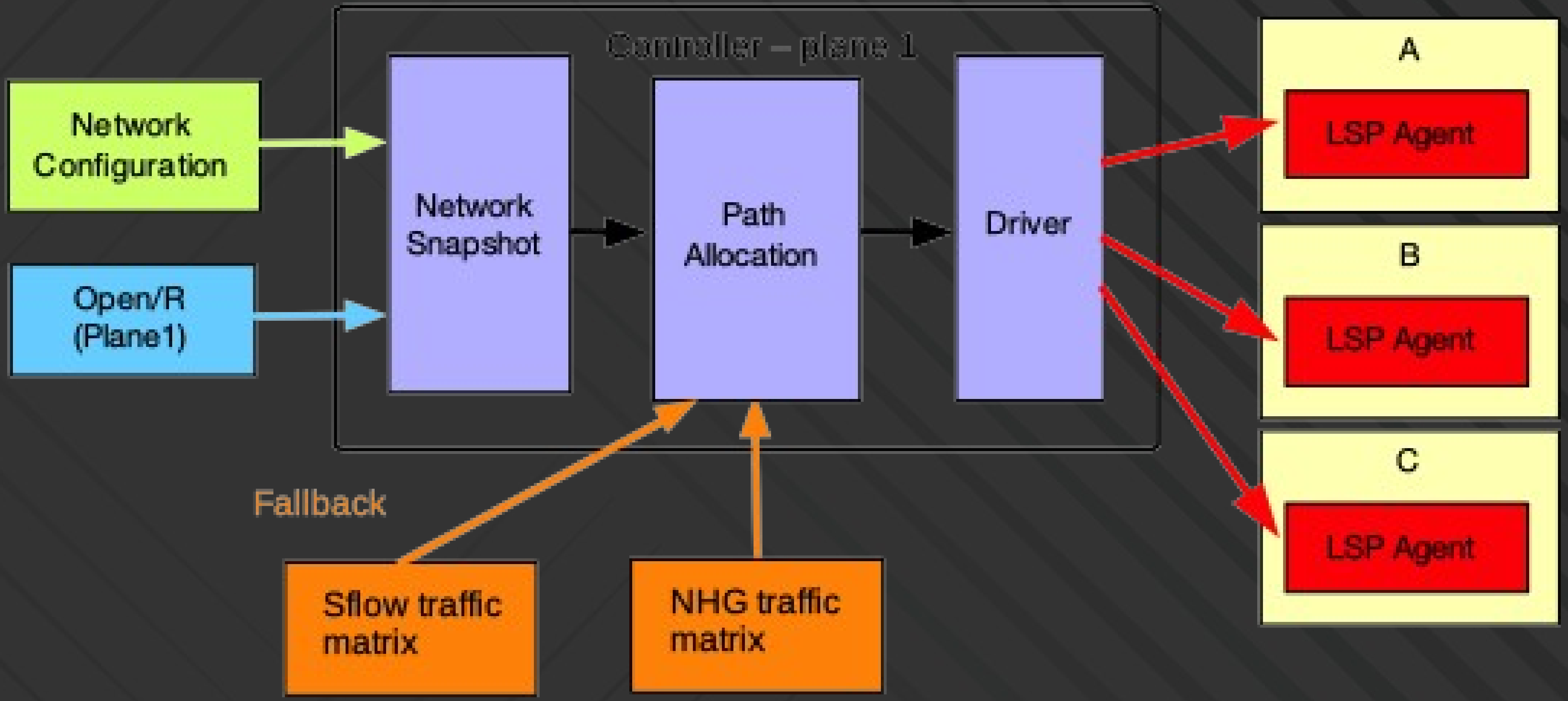
- Incremental on-boarding
- Destination prefix config
- Inject prefixes to EBB routers
- Fall back by withdrawing prefixes
- **Redundancy thru AS path prepend**



- Motivations
- Network Design
- **Traffic Engineering**
- Lessons Learned

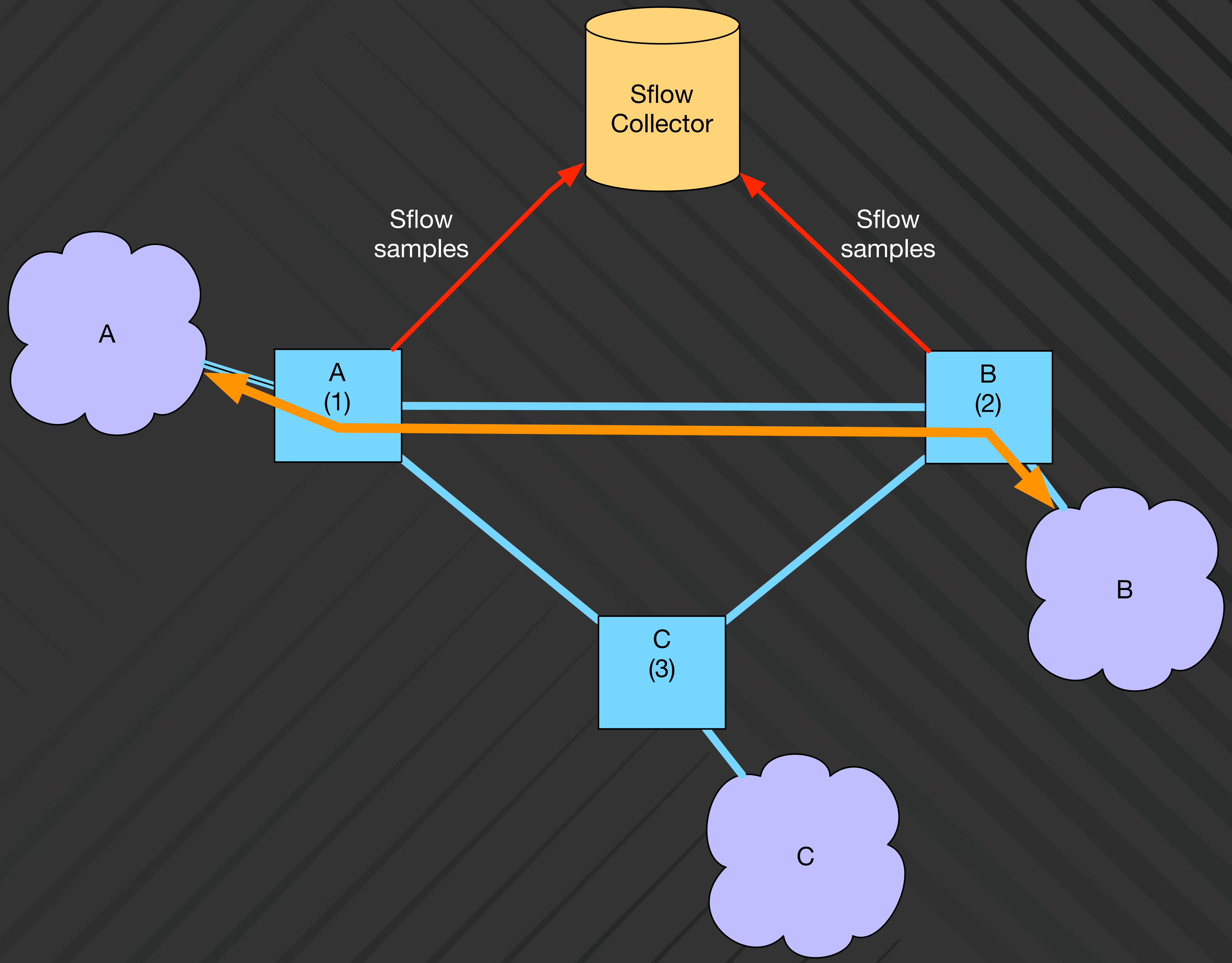
Traffic Engineering

- Network Snapshot
- Traffic matrix
- Path allocation
- **Driver**



Traffic Estimation

- **Collect sFlow samples from all routers**



Traffic Estimation

- **Collect sFlow samples from all routers**

face:1::1	face:2::6	12	1500B
face:2::11	face:1::2	28	1496B
face:2::1	face:1::6	28	128B
face:1::2	face:2::6	12	500B
face:2::5	face:1::1	12	1500B

Traffic Estimation

- Collect sFlow samples from all routers
- **Classify IP addresses to sites**

A	B	12	1500B
B	A	28	1496B
B	A	28	128B
A	B	12	500B
B	A	12	1500B

Traffic Estimation

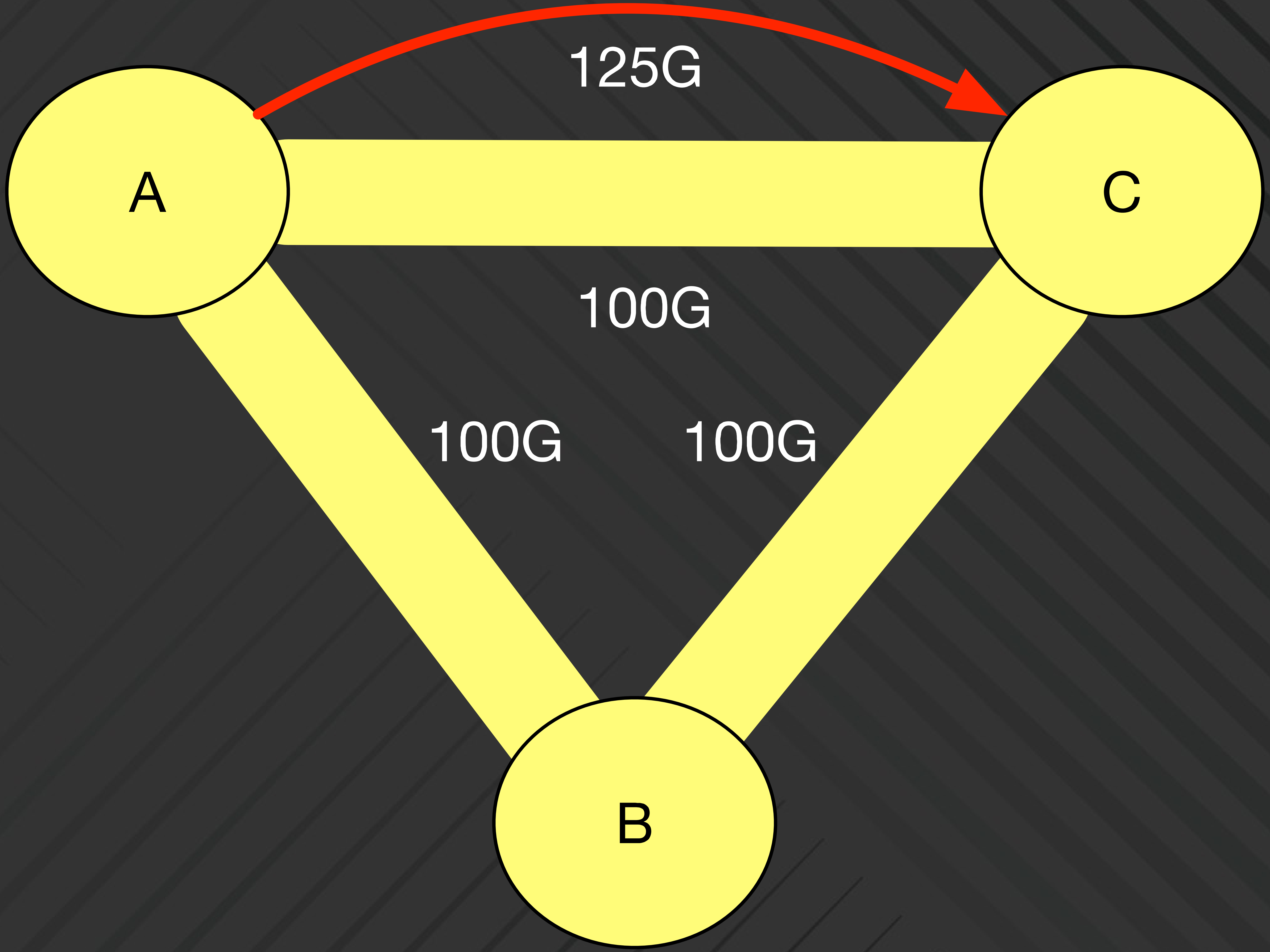
- Collect sFlow samples from all routers
- Classify IP addresses to sites
- **Aggregate samples to estimate # bytes per site pair / DSCP**

A	B	12	2.7 Mbps
B	A	28	2.2 Mbps
B	A	12	2.0 Mbps

Traffic Estimation – NHGs counters

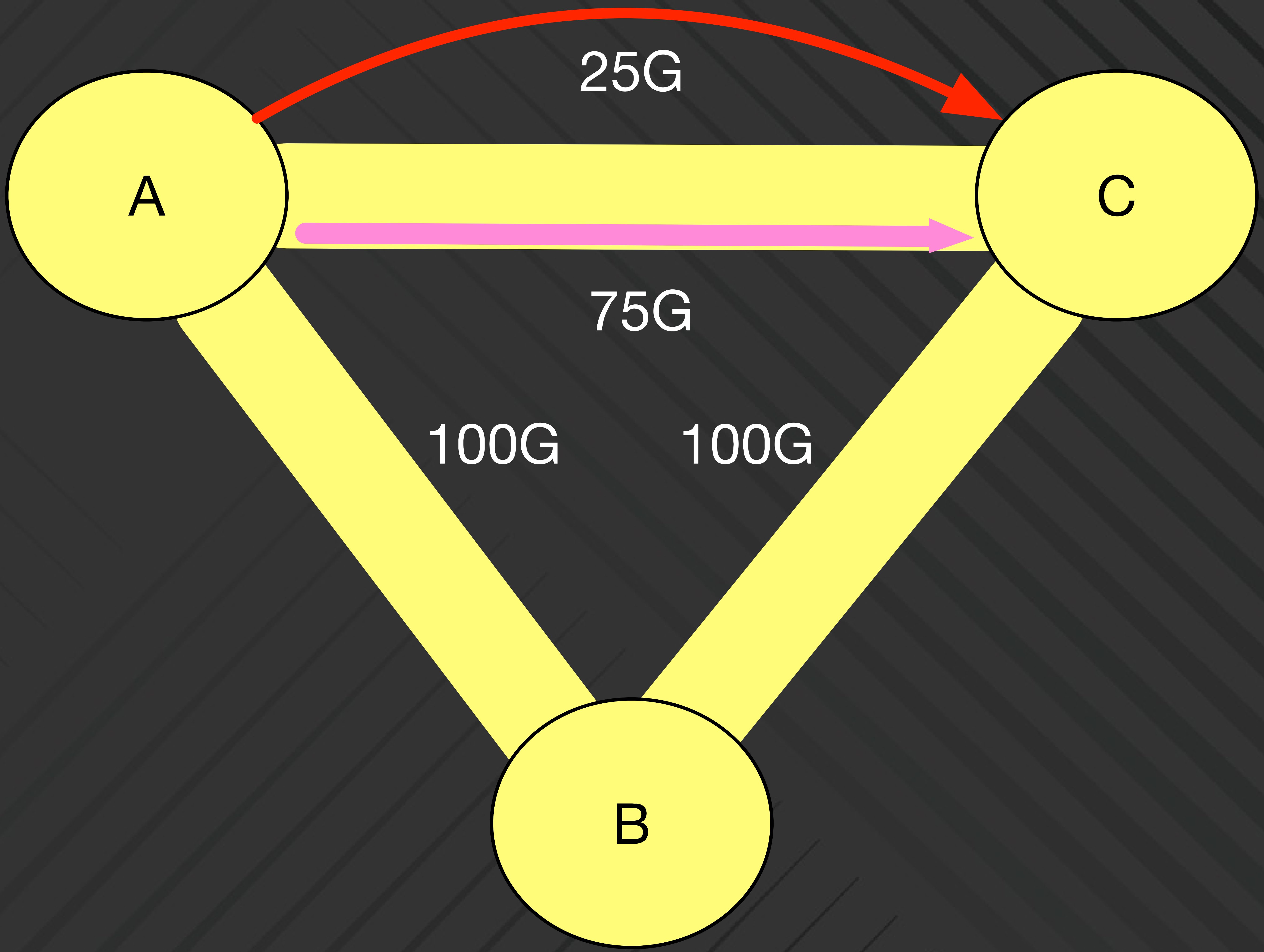
- Connect to LSP agents, running on routers.
- **Translate next-hop groups to site pairs**

Path Allocation



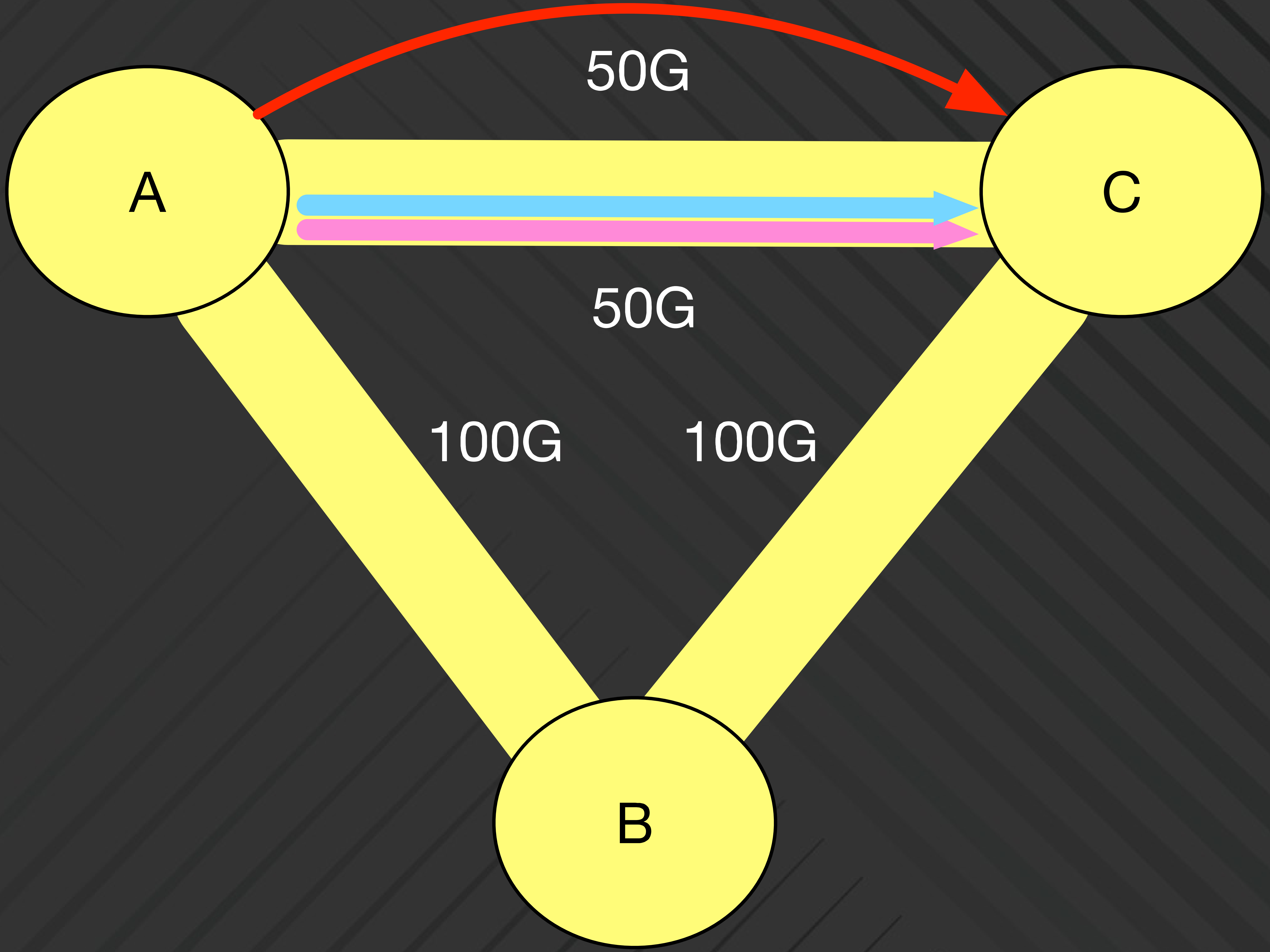
Path Allocation

- **Capacity constrained primary path**



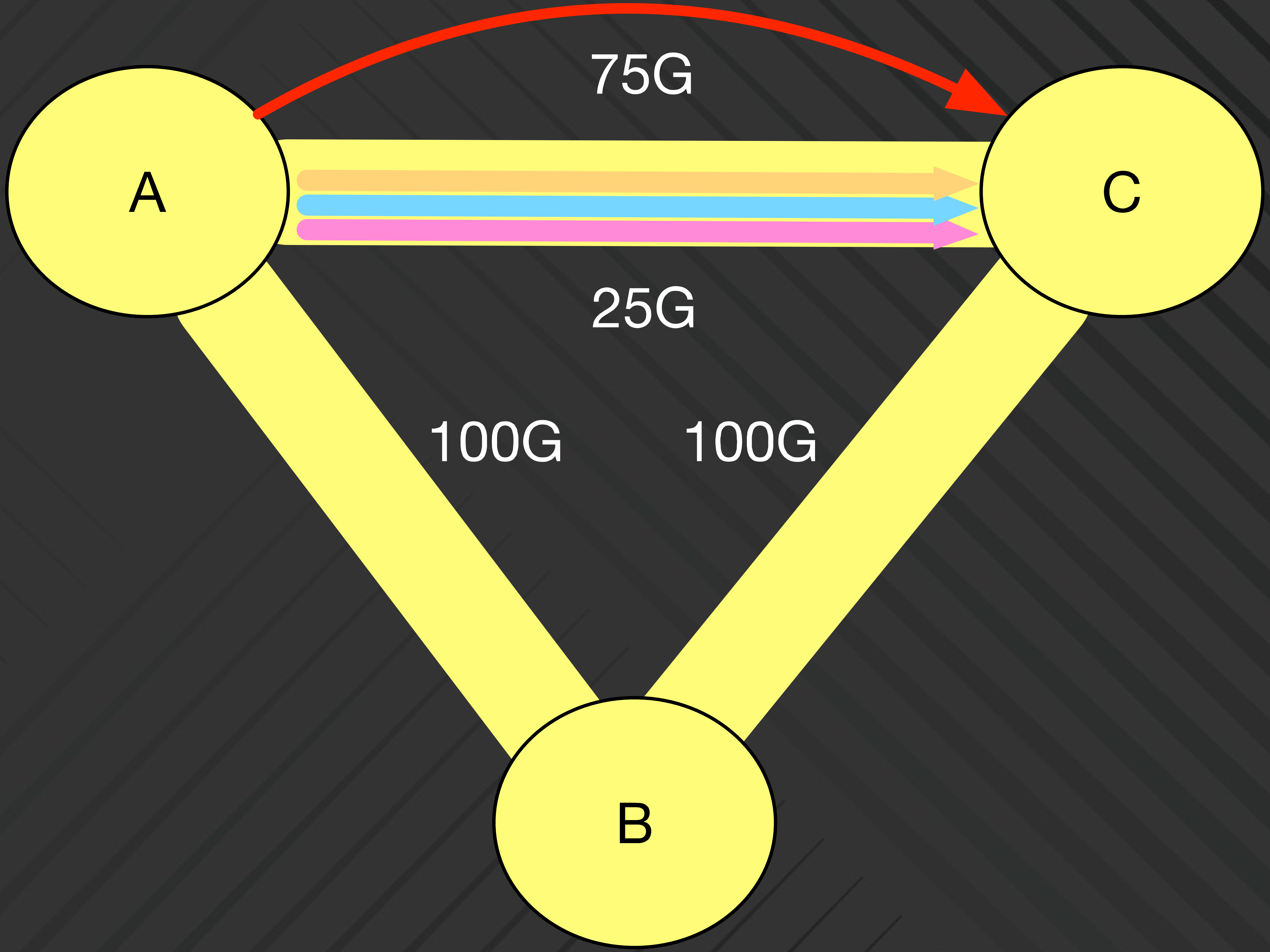
Path Allocation

- **Capacity constrained primary path**



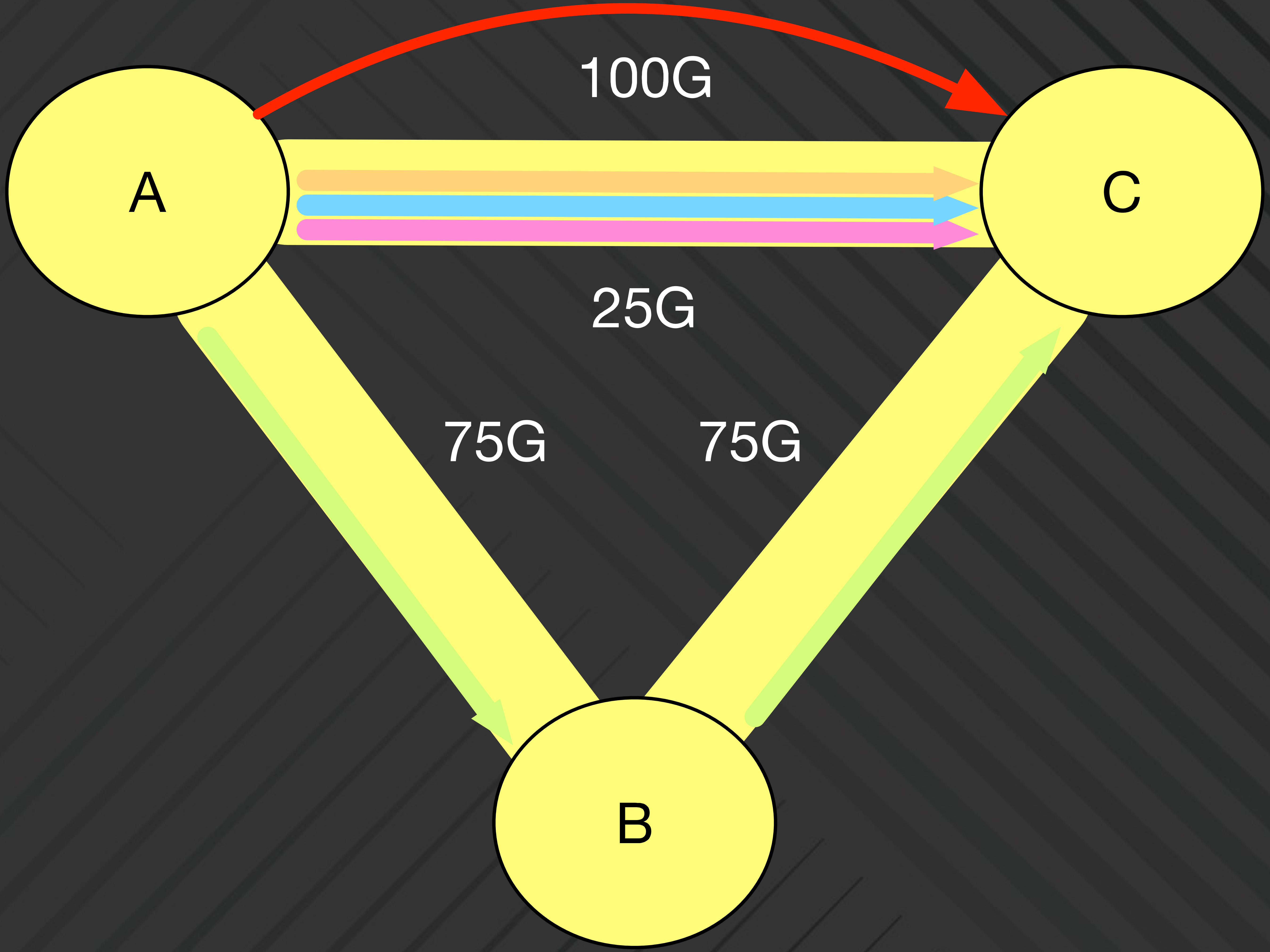
Path Allocation

- **Capacity constrained primary path**



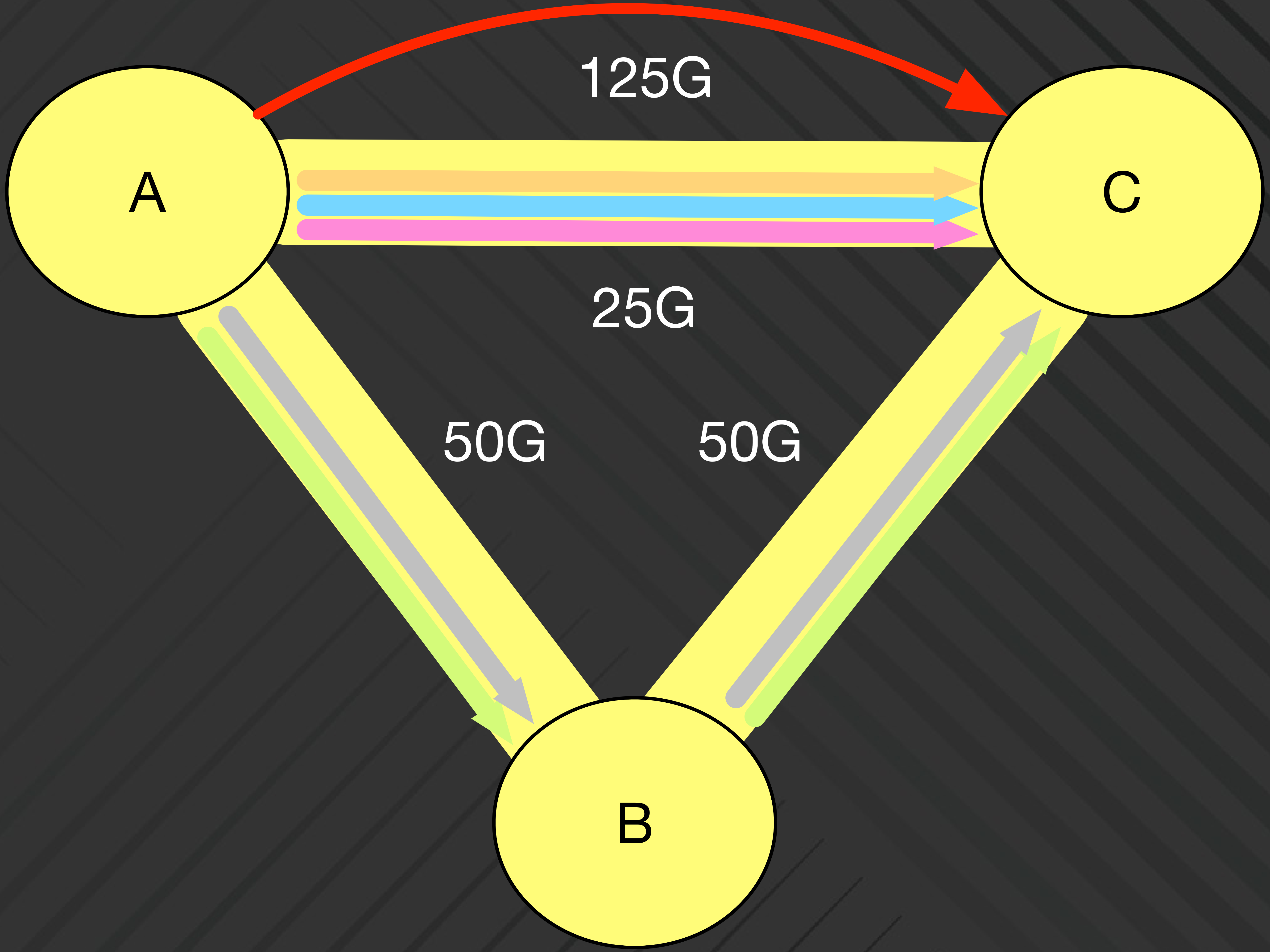
Path Allocation

- **Capacity constrained primary path**



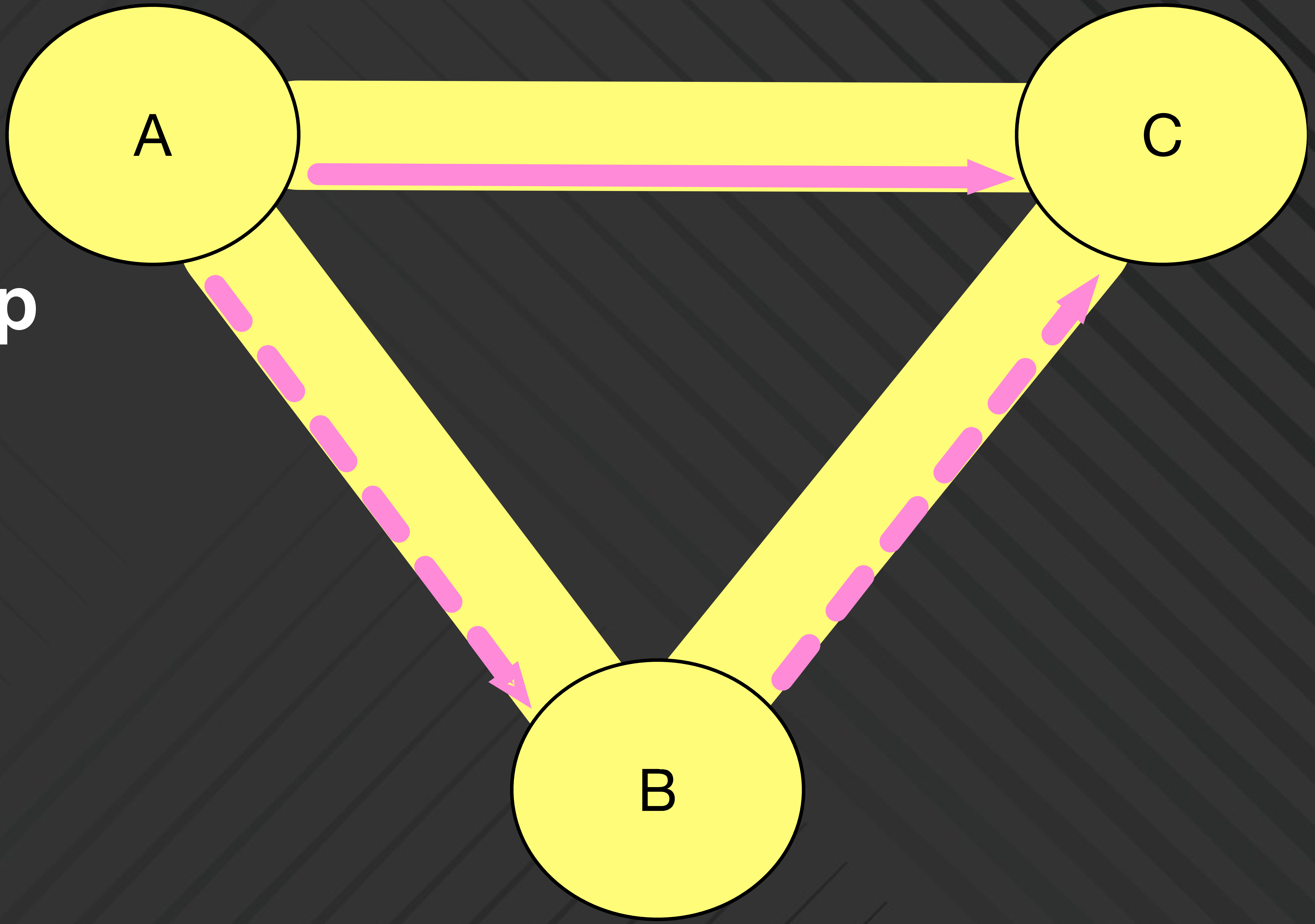
Path Allocation

- **Capacity constrained primary path**



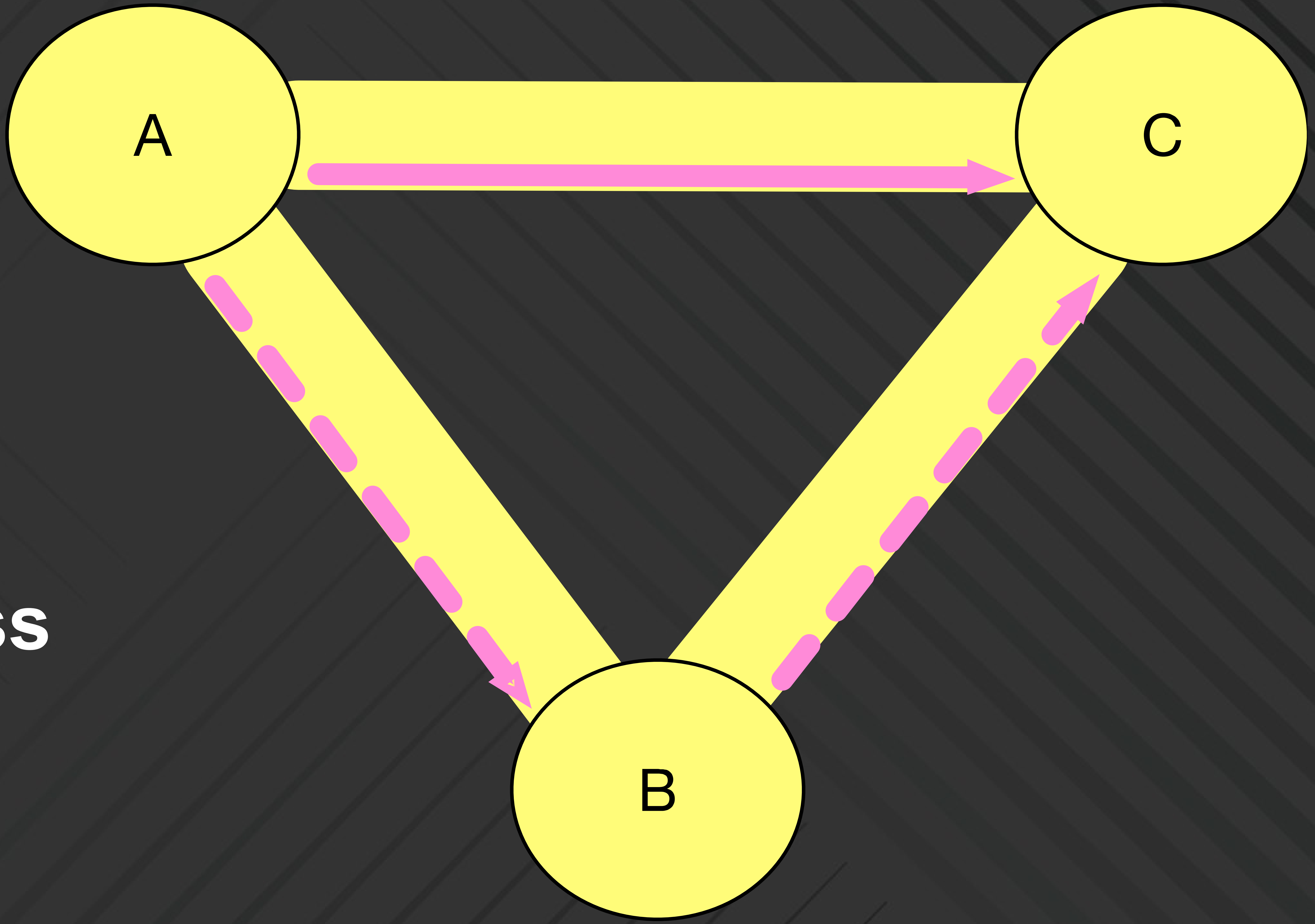
Path Allocation

- Capacity constrained primary path
- **Maximal diverse backup path**



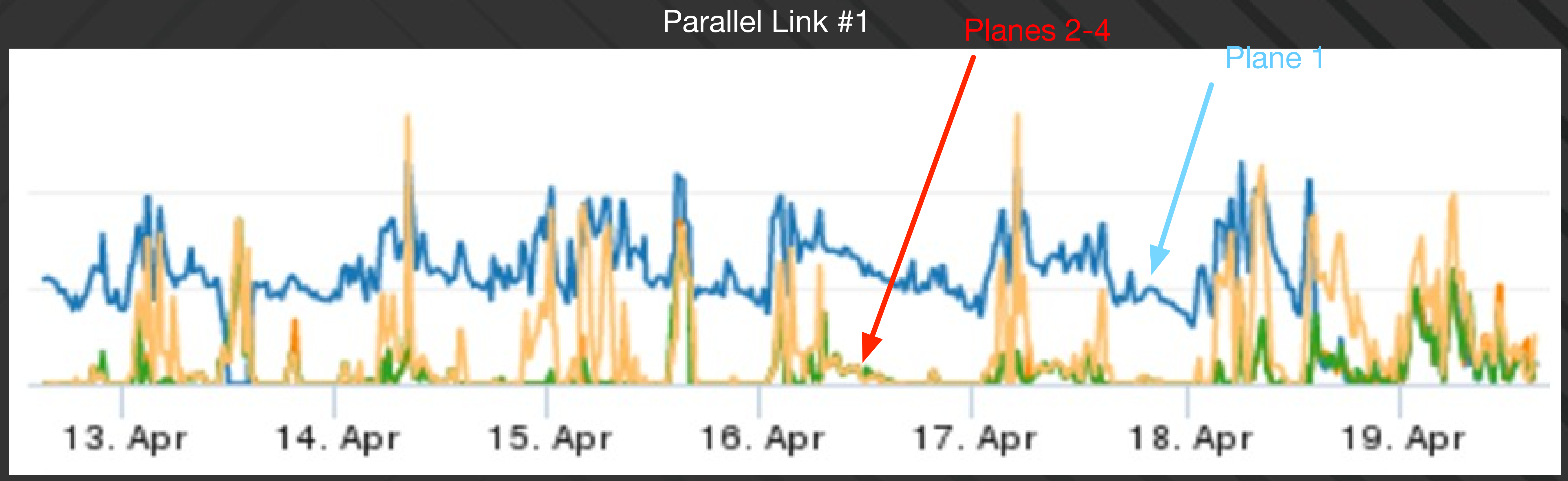
Path Allocation

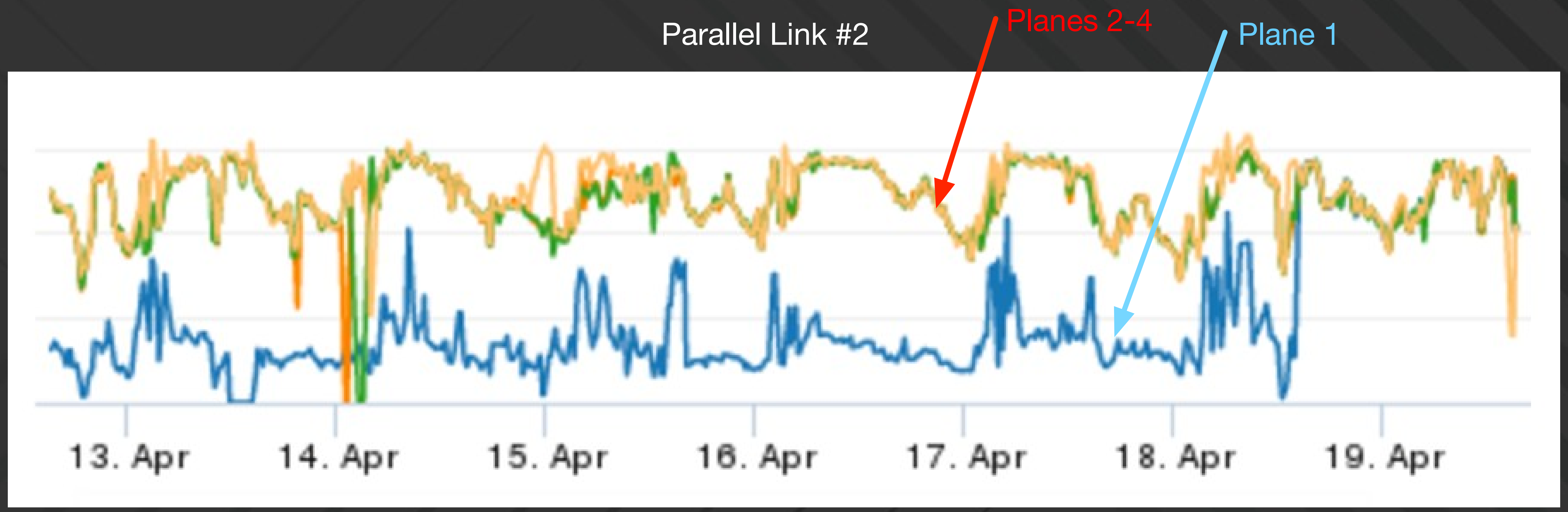
- Capacity constrained primary path
- Maximal diverse backup path
- **One LSP mesh per DSCP-based traffic class**



Path Allocation - MCF

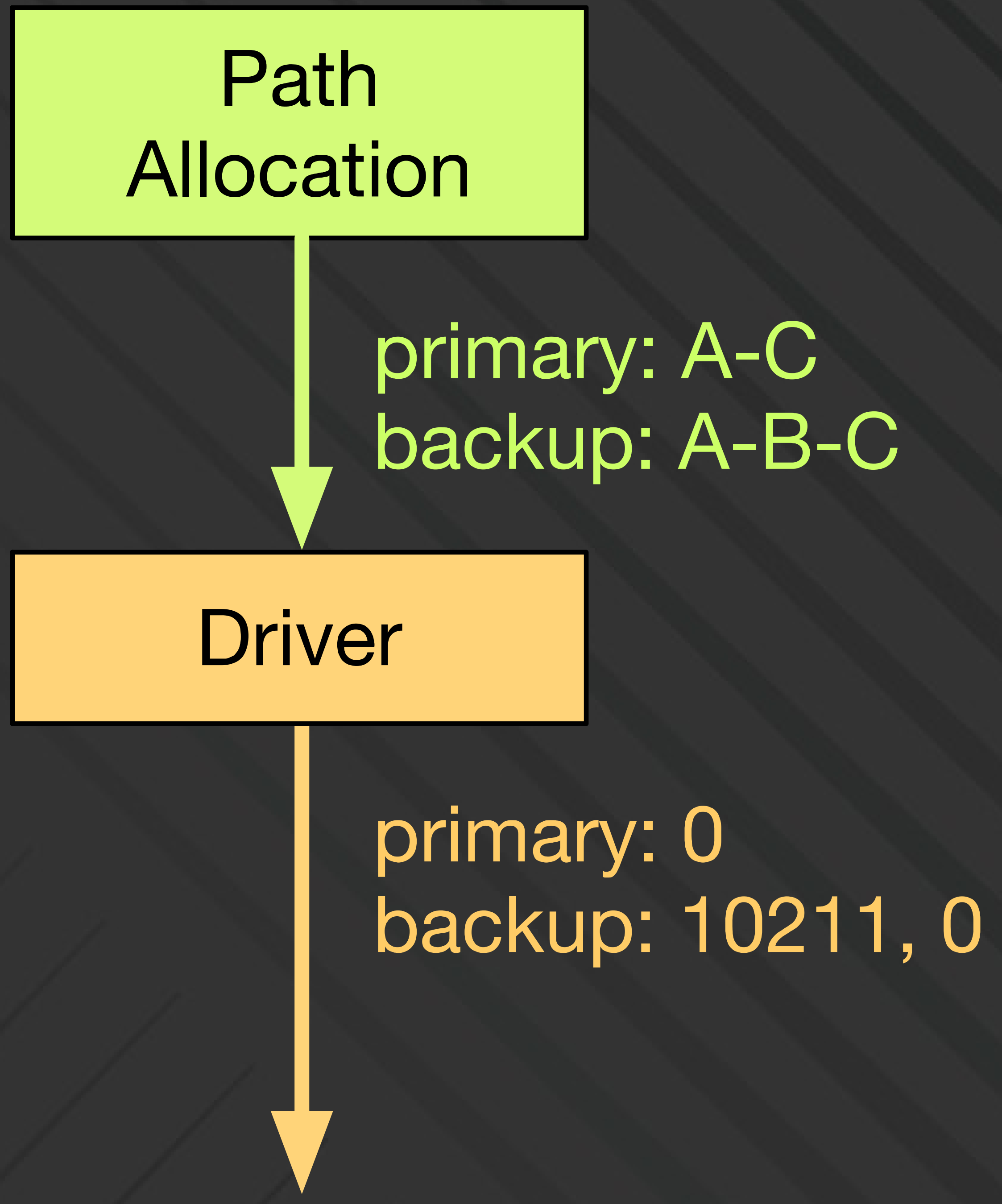
- Other algorithms exist
- Multi-commodity flow
  - Maximize headroom
- Experiment
  - Utilization of two parallel links
  - Spread traffic across them


Parallel Link #1


Parallel Link #2

Driver

- **Segment routing**



Path Allocation

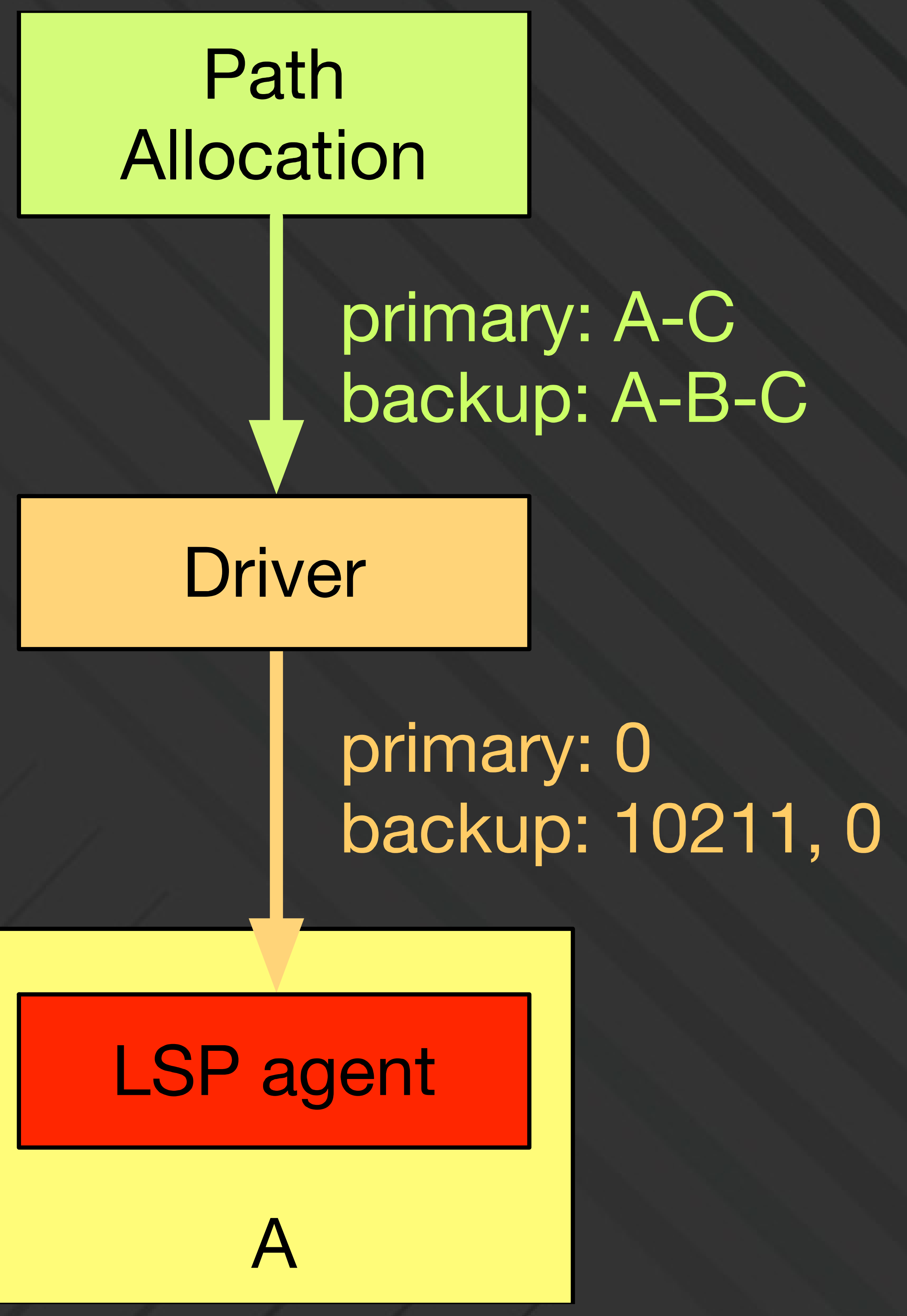
primary: A-C
backup: A-B-C

Driver

primary: 0
backup: 10211, 0

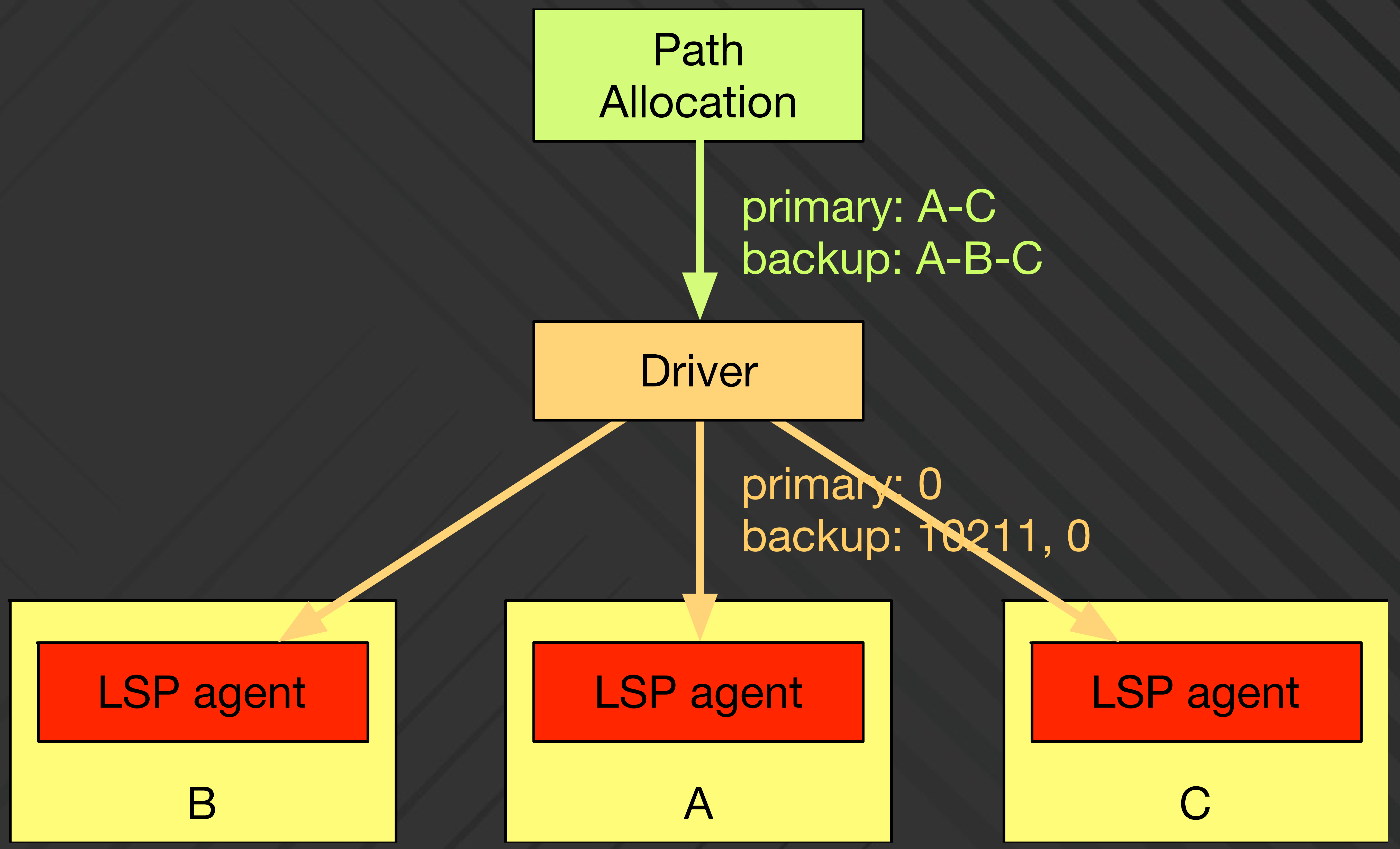
Driver

- Segment routing
- **LSP agent programs LSPs**



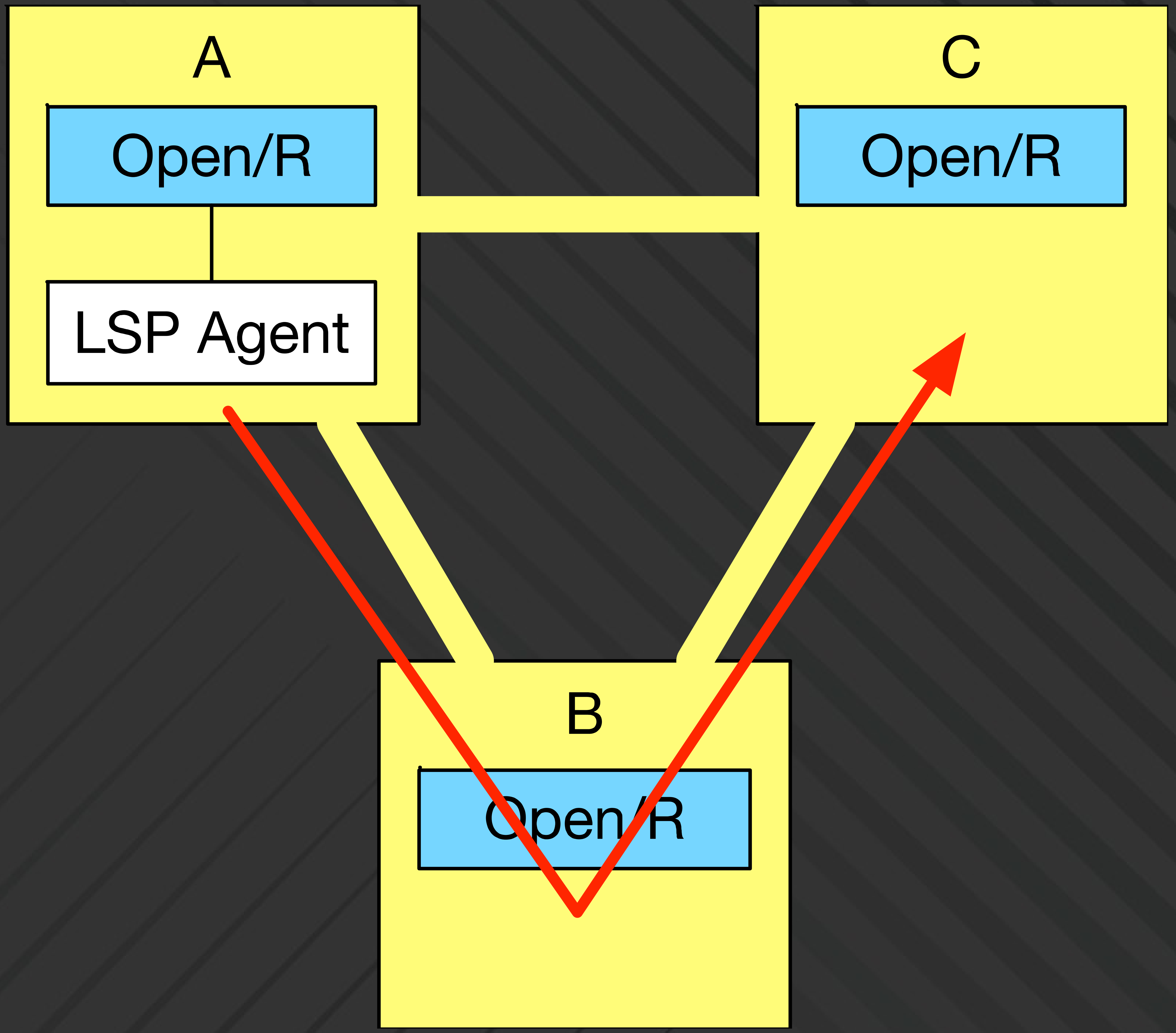
Driver

- Segment routing
- LSP agent programs LSPs
- **No inter-device signaling**



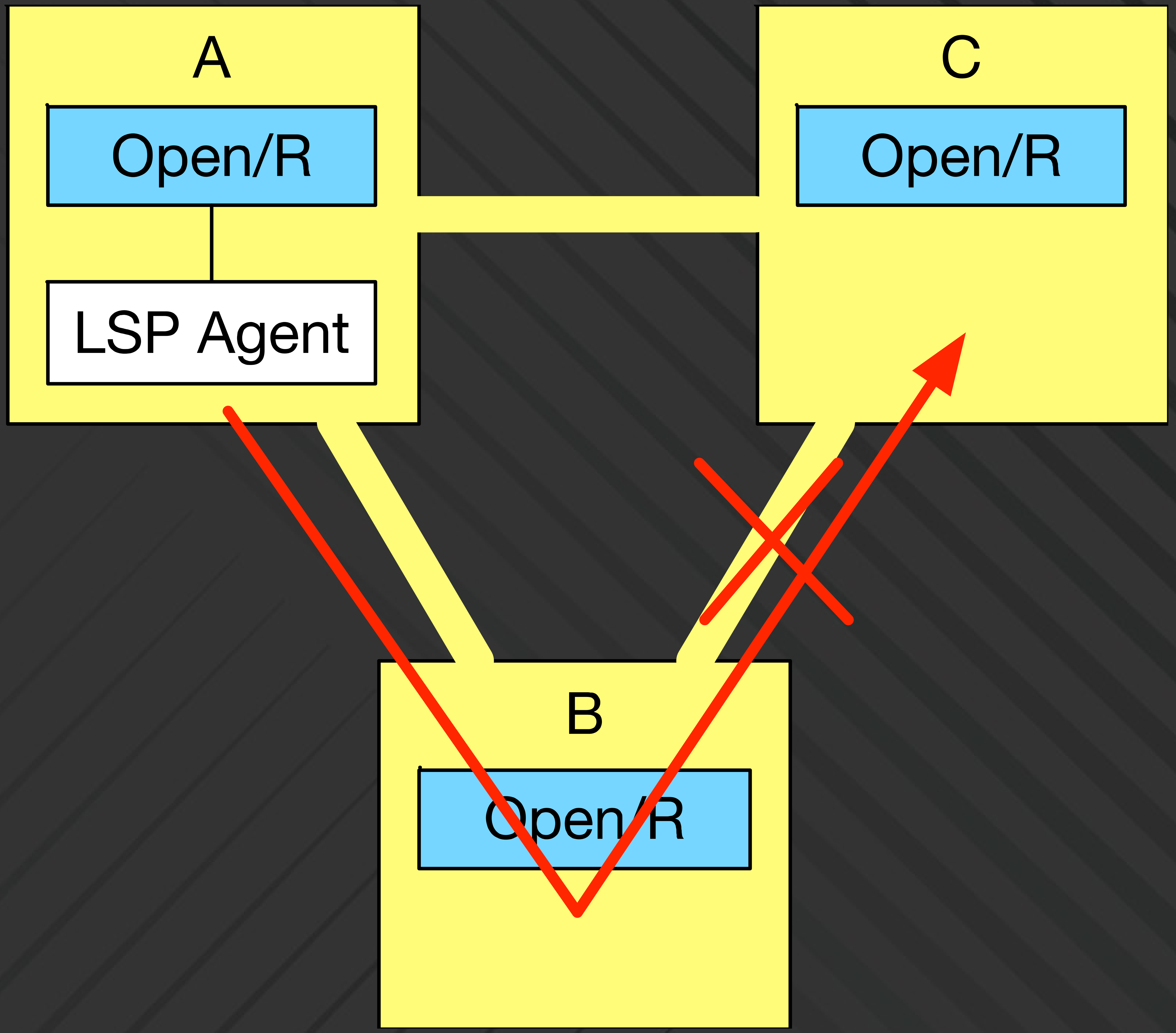
Driver

- Segment routing
- LSP agent programs LSPs
- No inter-device signaling
- Failover
  - **LSP agent reacts to topology changes**



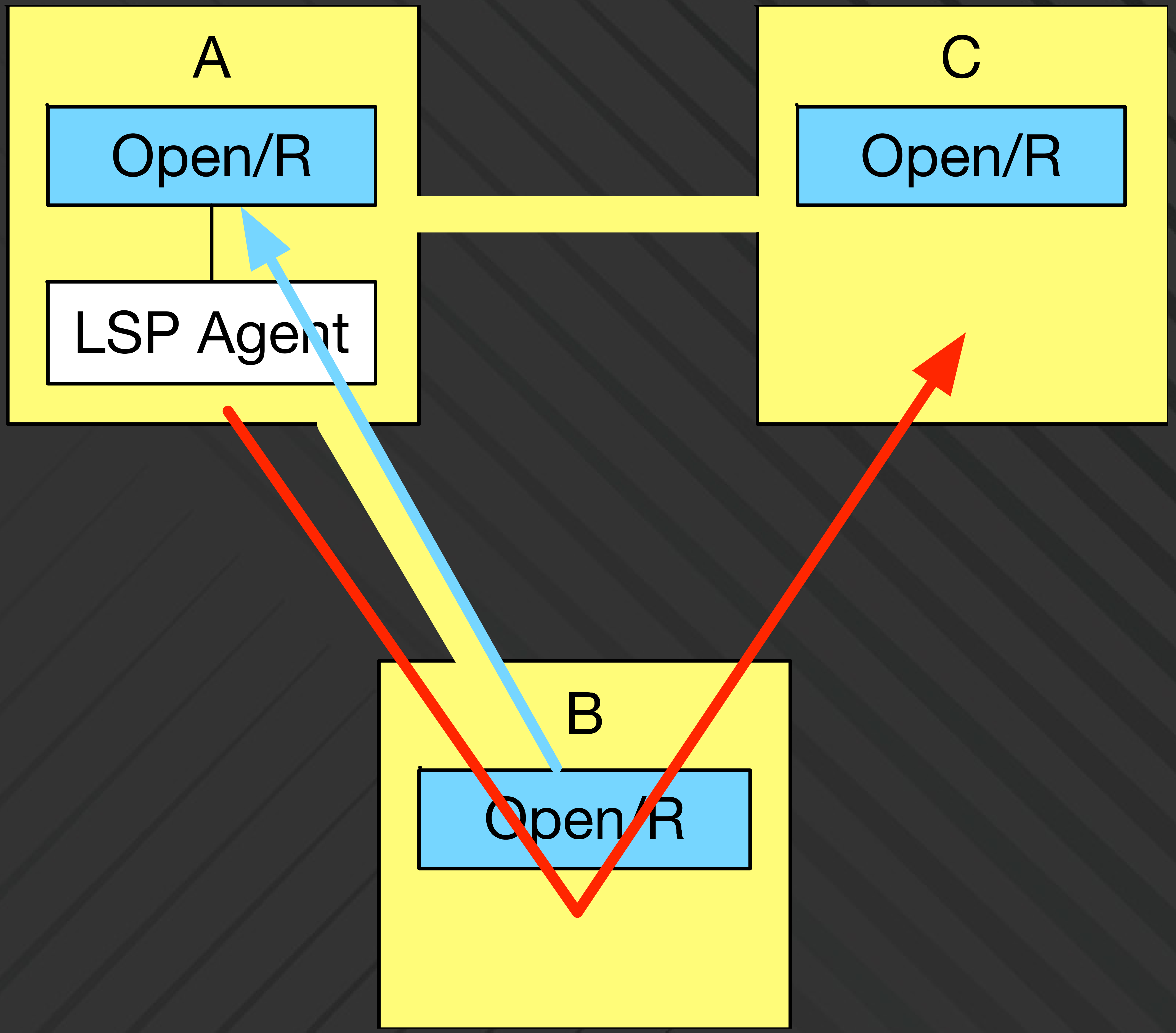
Driver

- Segment routing
- LSP agent programs LSPs
- No inter-device signaling
- Failover
  - LSP agent reacts to topology changes
  - **Use backup path if primary is down**



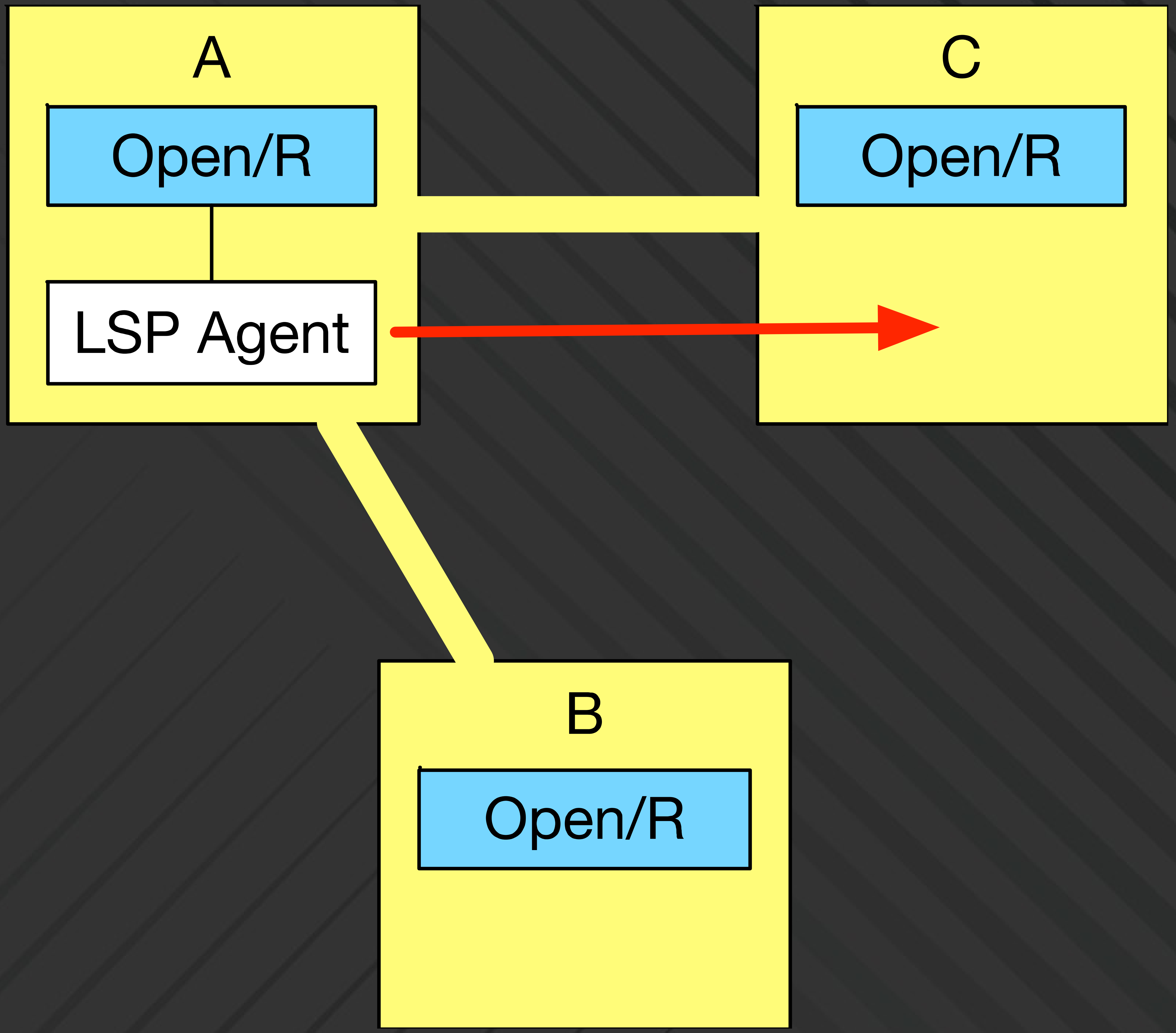
Driver

- Segment routing
- LSP agent programs LSPs
- No inter-device signaling
- Failover
  - LSP agent reacts to topology changes
  - **Use backup path if primary is down**



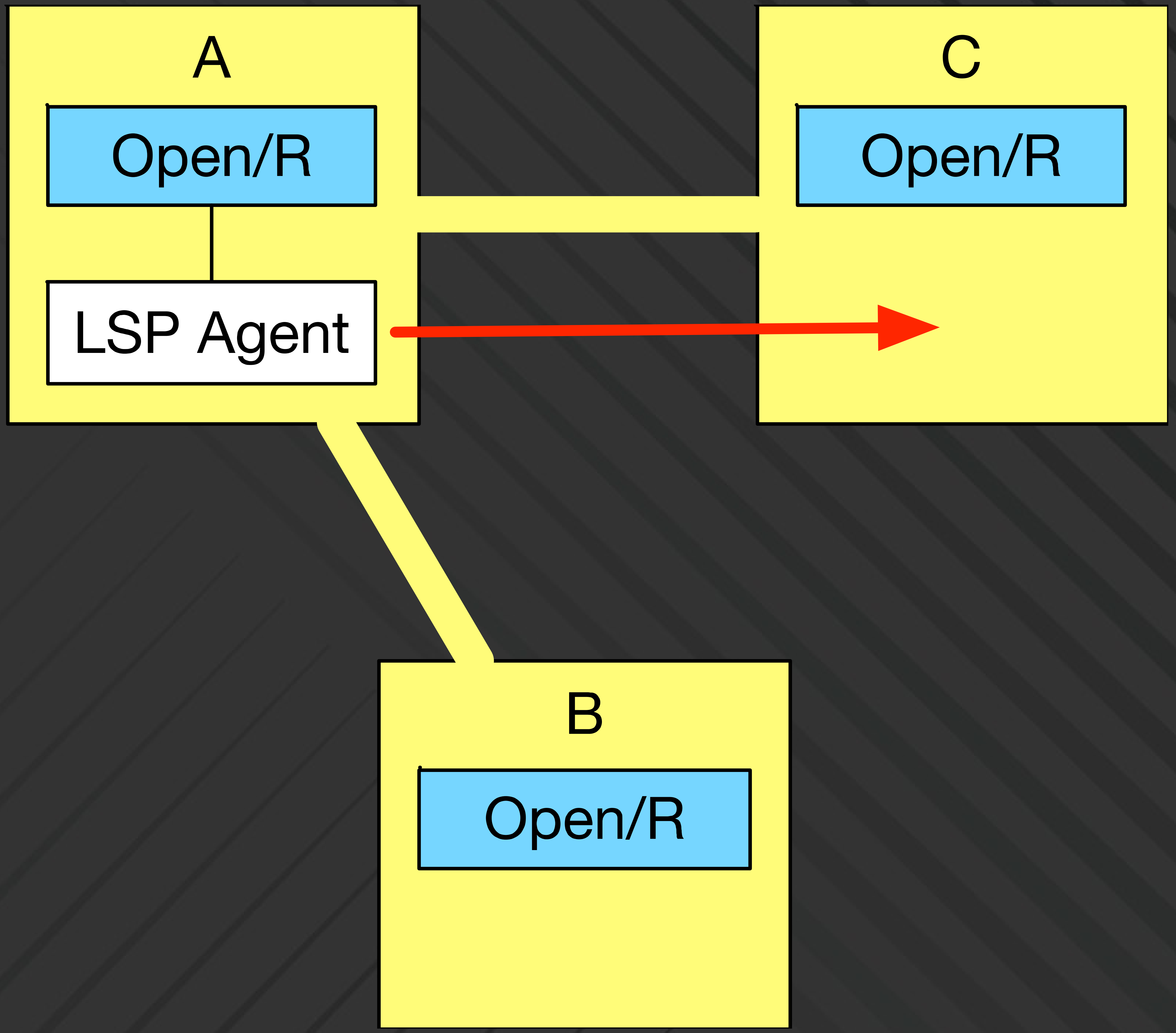
Driver

- Segment routing
- LSP agent programs LSPs
- No inter-device signaling
- Failover
  - LSP agent reacts to topology changes
  - **Use backup path if primary is down**



Driver

- Segment routing
- LSP agent programs LSPs
- No inter-device signaling
- Failover
  - LSP agent reacts to topology changes
  - Use backup path if primary is down
  - **Remove LSP if backup is down**



QOS differences

- Platinum, Gold
  - Try to avoid loss as possible
  - Tiny amount of traffic eligible
- Silver, Bronze
  - We don’t care about drops here, try our best to reduce probability

Controller

- Flexible
  - Can create our own traffic engineering mechanism
  - Support different algorithms per plane per traffic class
  - Driver can be customized per plane
- Minimal Platform Dependency – Avoid platform specific features

- Motivations
- Network Design
- Traffic Engineering
- **Lessons Learned**

Lessons Learned – Software Management

- Lots of software components
- Manual upgrade → labor intensive and error prone
- A single rebuild operation
- Automated rebuild for an entire plane

Lessons Learned - Debugging

- Manual debugging → nearly impossible
  - Frequent changes from controller
  - Large # of software-generated objects
- Automation
  - Validations between controller and routers
  - Fault detection by Netnorad
  - Fault isolation by MPLS trace route

Wins - Operations

- Easy rollout of new software
- Drain is fast. No LSP re-optimization necessary

Wins - Performance

- Reacting to topology changes (drain, fiber cuts, etc.) is typically sub-seconds
- Real-time visibility of LSP path hops
- Correlation of LSPs and link utilization

Wins - Flexibility

- Ability to experiment on new TE algorithms
- Multiple planes allows A/B testing
- Moving fast

A Fun Journey

- EBB went from a concept to reality
- Learn a lot on operating a SDN
- Expanding to new sites
- Turning up new capacities

Questions?

fb.com/mickvav

m.me/mickvav

mickvav@fb.com