Seamless BGP Migration with Router Grafting

Eric Keller, Jennifer Rexford
Princeton University

Kobus van der Merwe
AT&T Research

NSDI 2010

Dealing with Change

- Networks need to be highly reliable
  - To avoid service disruptions
- Operators need to deal with change
  - Install, maintain, upgrade, or decommission equipment
  - Deploy new services
  - Manage resource usage (CPU, bandwidth)
- But… change causes disruption
  - Forcing a tradeoff

Why is Change so Hard?

- Root cause is the monolithic view of a router (Hardware, software, and links as one entity)

Why is Change so Hard?

- Root cause is the monolithic view of a router (Hardware, software, and links as one entity)

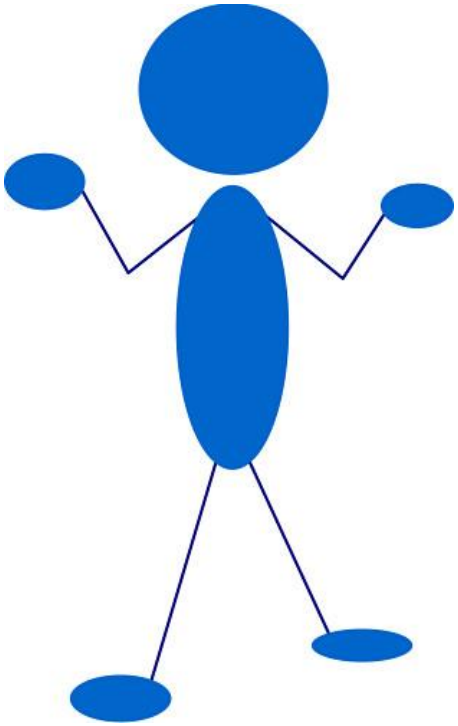
Revisit the design to make dealing with change easier

Our Approach: Grafting

- In nature: take from one, merge into another
  - Plants, skin, tissue

- Router Grafting
  - To break the monolithic view
  - Focus on moving link (and corresponding BGP session)

Why Move Links?

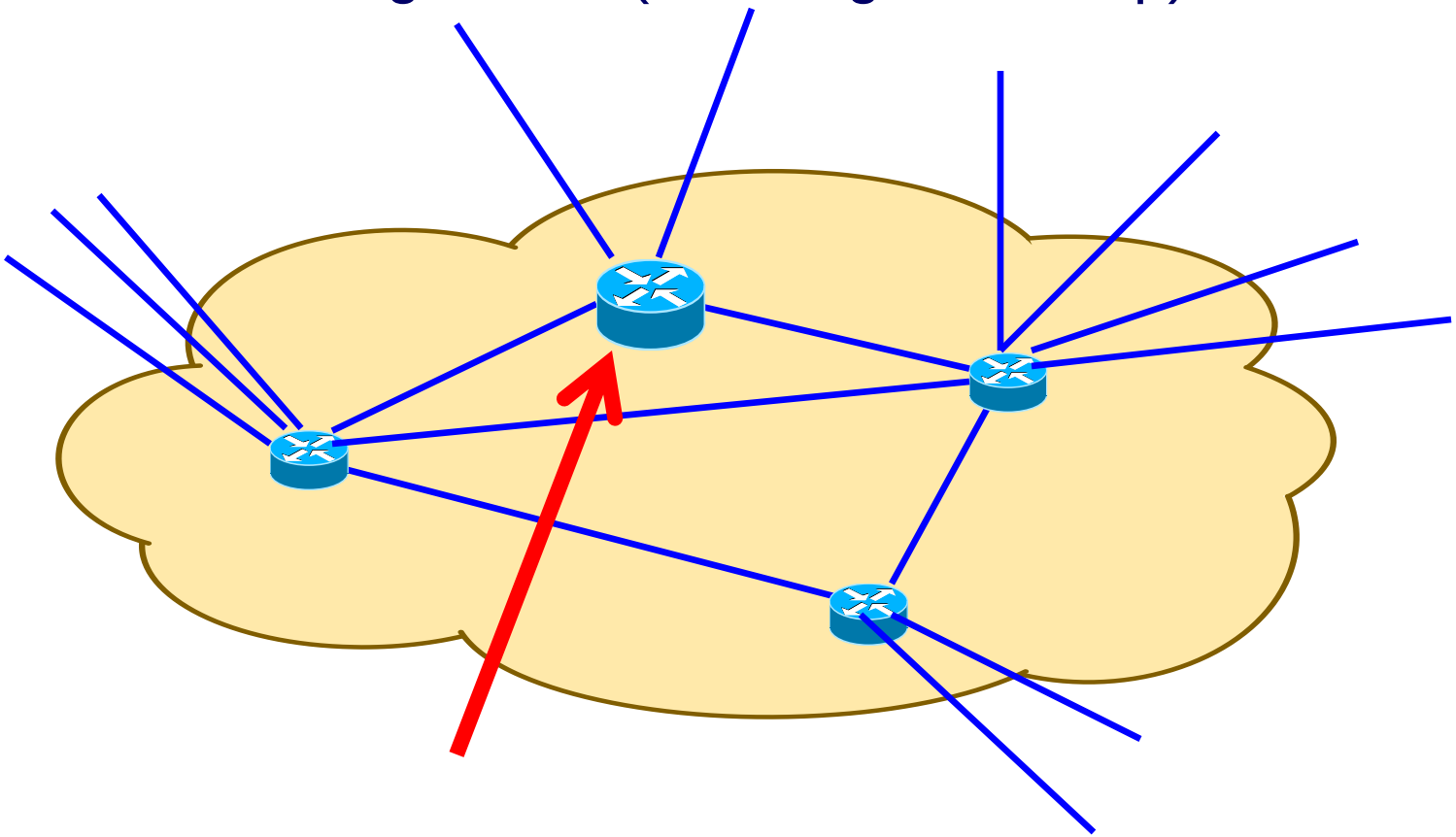


Planned Maintenance

- Shut down router to…
  - Replace power supply
  - Upgrade to new model
  - Contract network
- Add router to…
  - Expand network

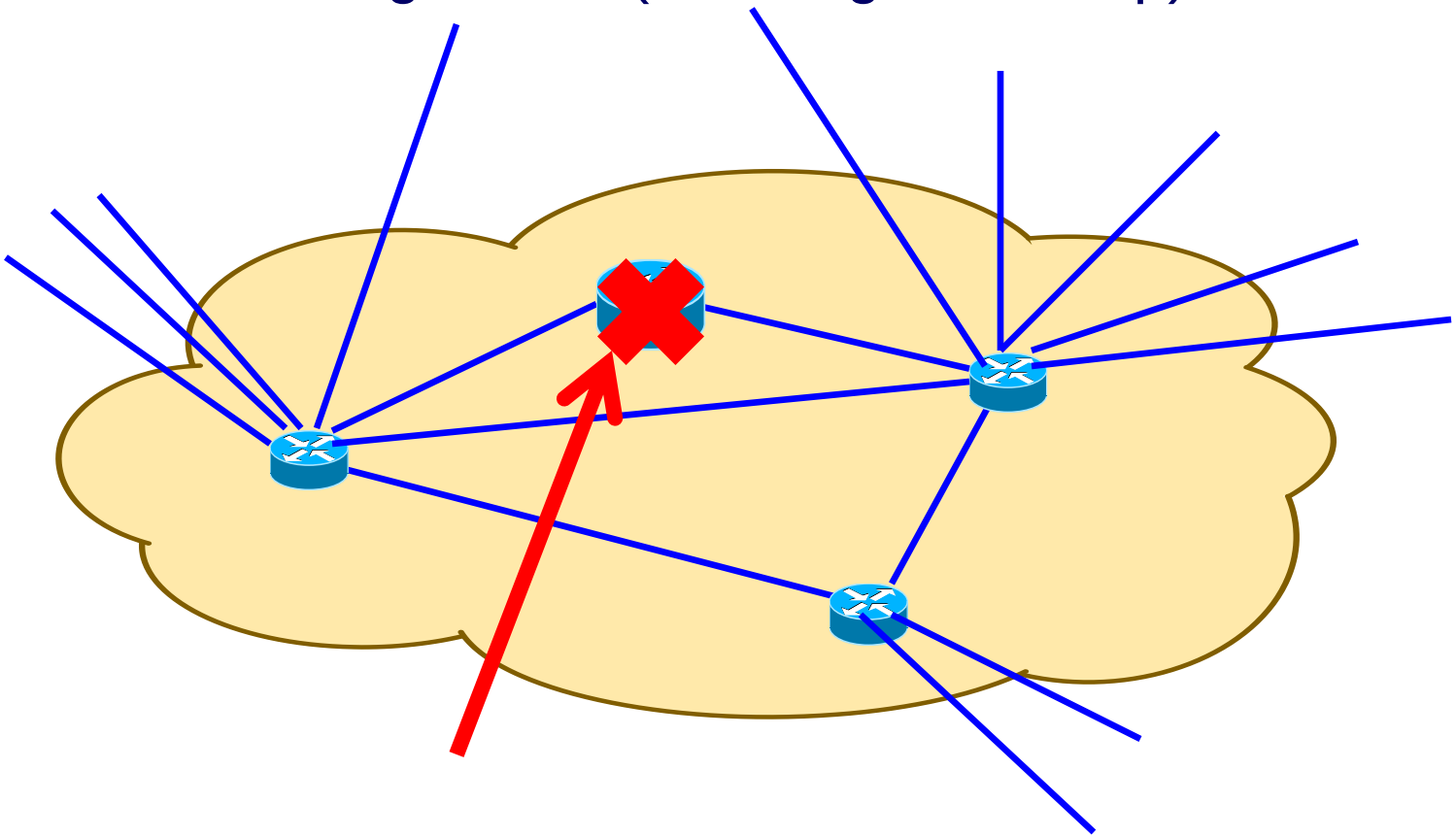
Planned Maintenance

- Could migrate links to other routers
  - Away from router being shutdown, or
  - To router being added (or brought back up)



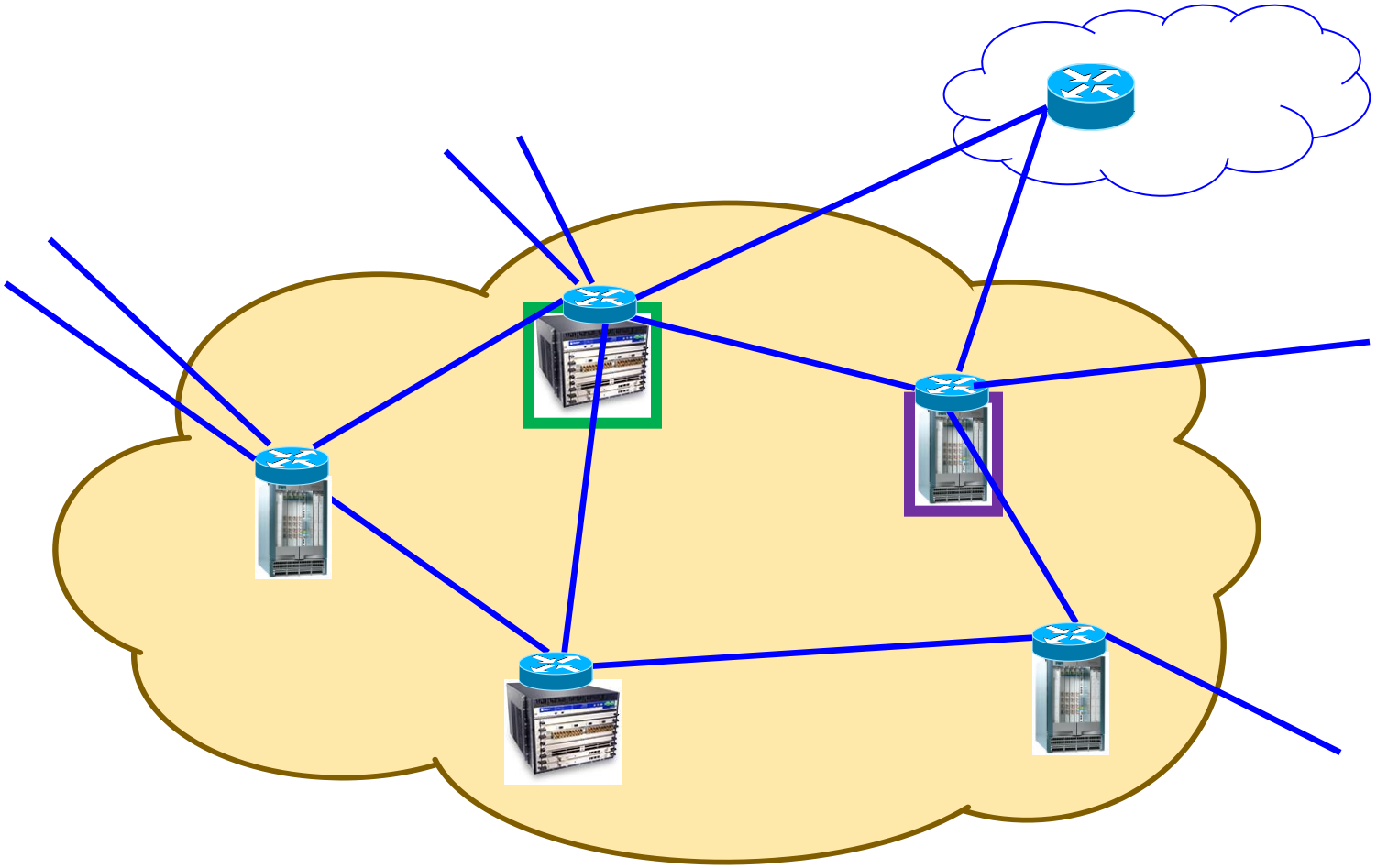
Planned Maintenance

- Could migrate links to other routers
  - Away from router being shutdown, or
  - To router being added (or brought back up)



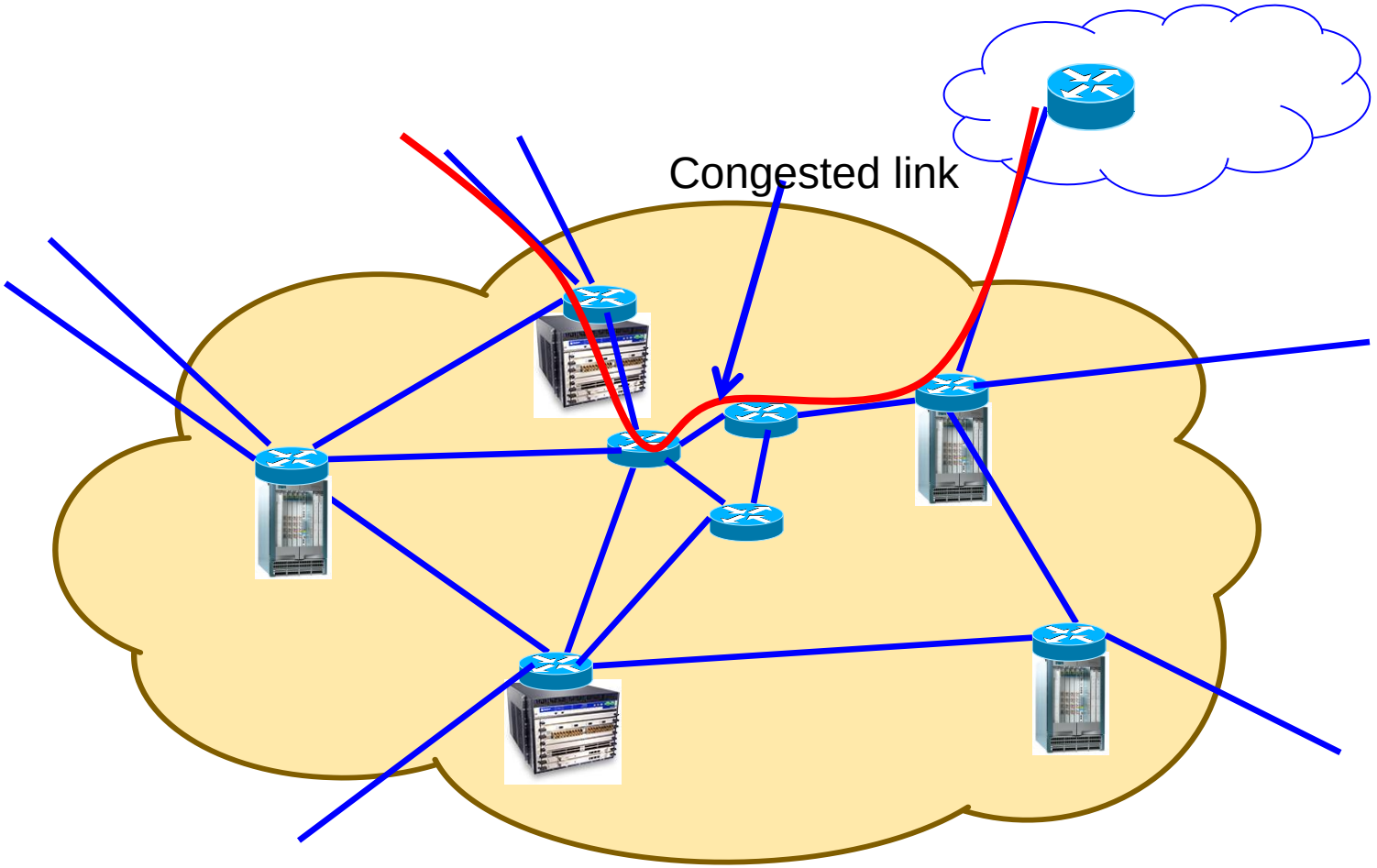
Customer Requests a Feature

Network has mixture of routers from different vendors
* Rehome customer to router with needed feature



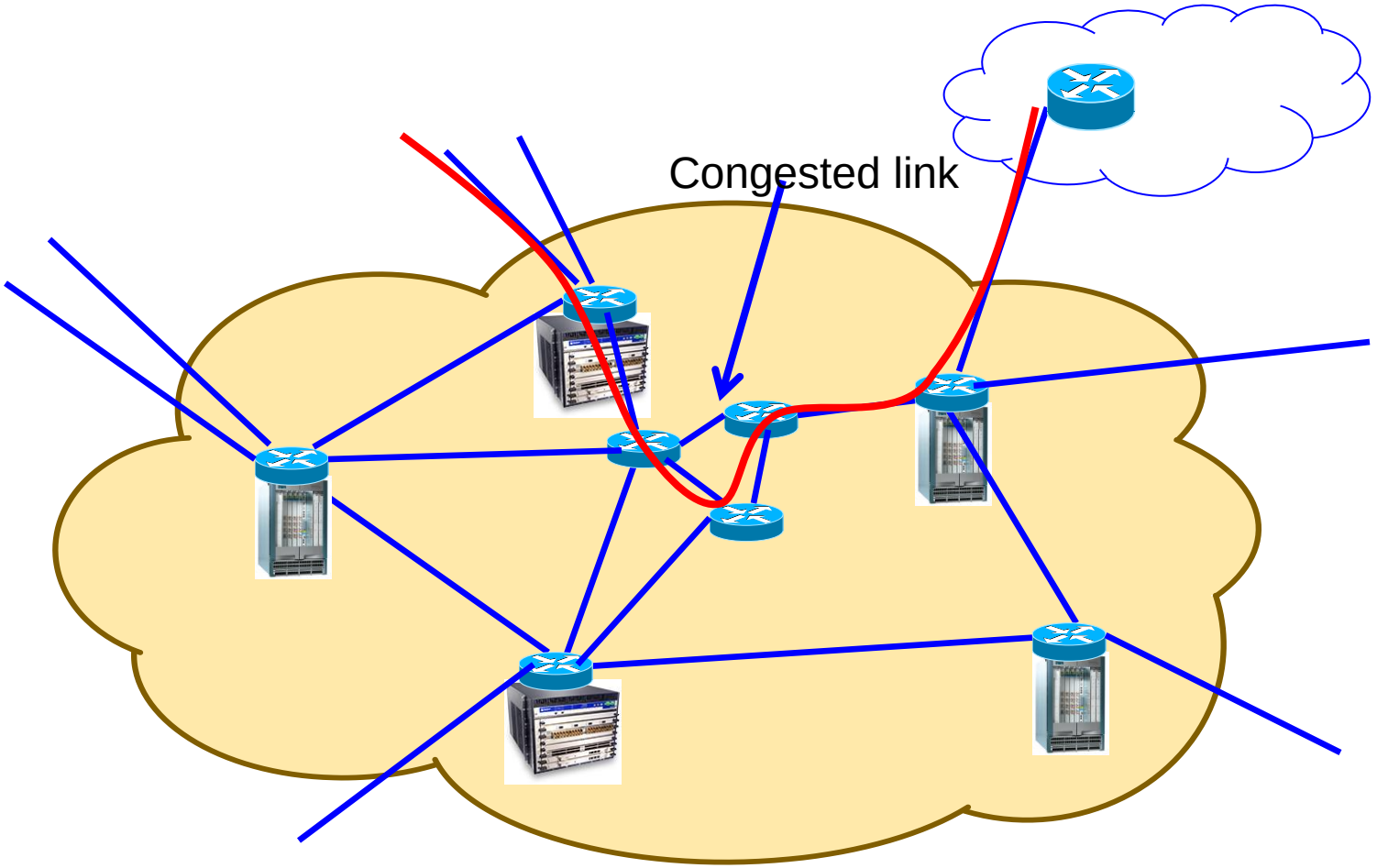
Traffic Management

Typical traffic engineering:
* adjust routing protocol parameters based on traffic



Traffic Management

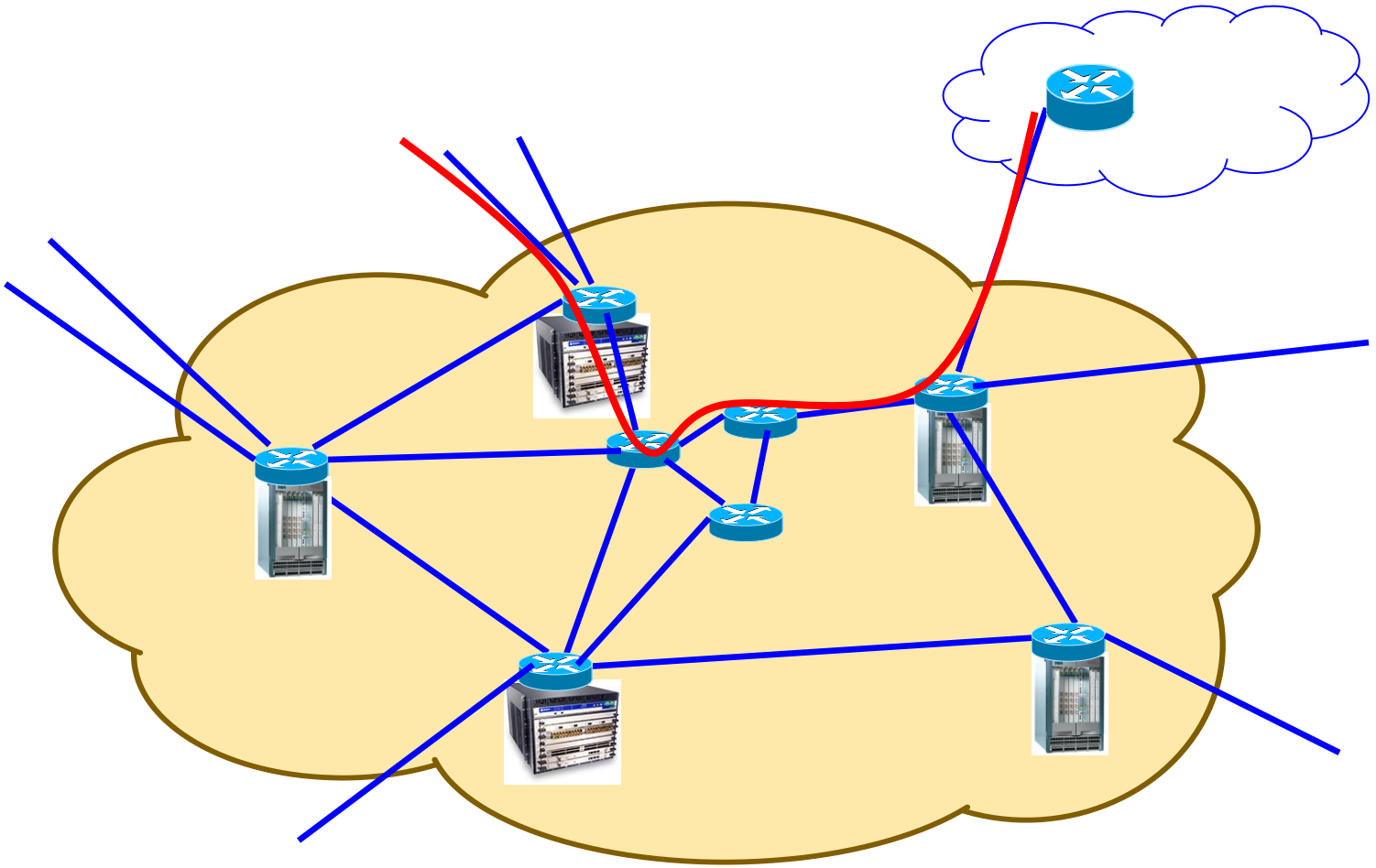
Typical traffic engineering:
* adjust routing protocol parameters based on traffic



Traffic Management

Instead…

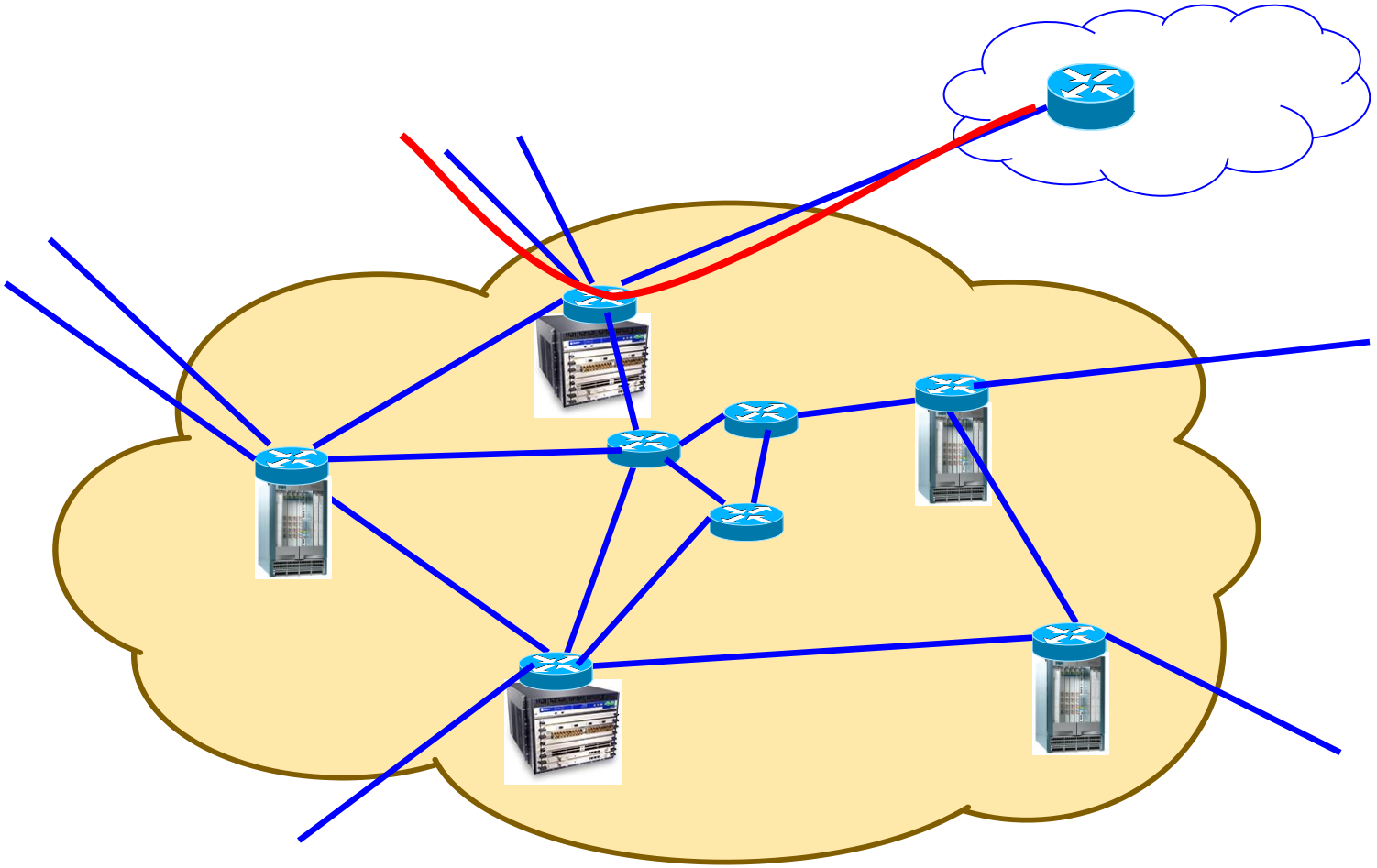
* Rehome customer to change traffic matrix



Traffic Management

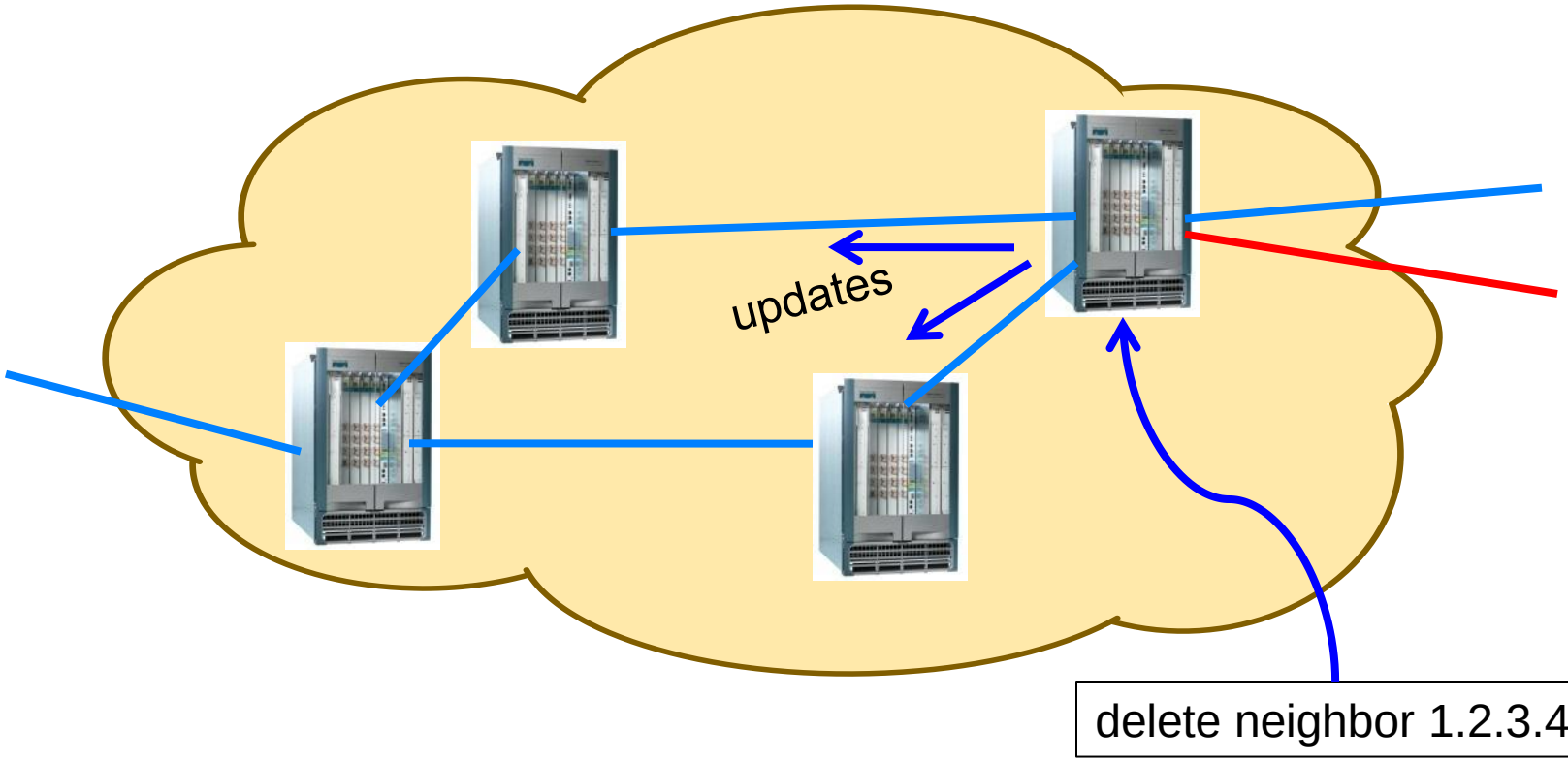
Instead…

* Rehome customer to change traffic matrix



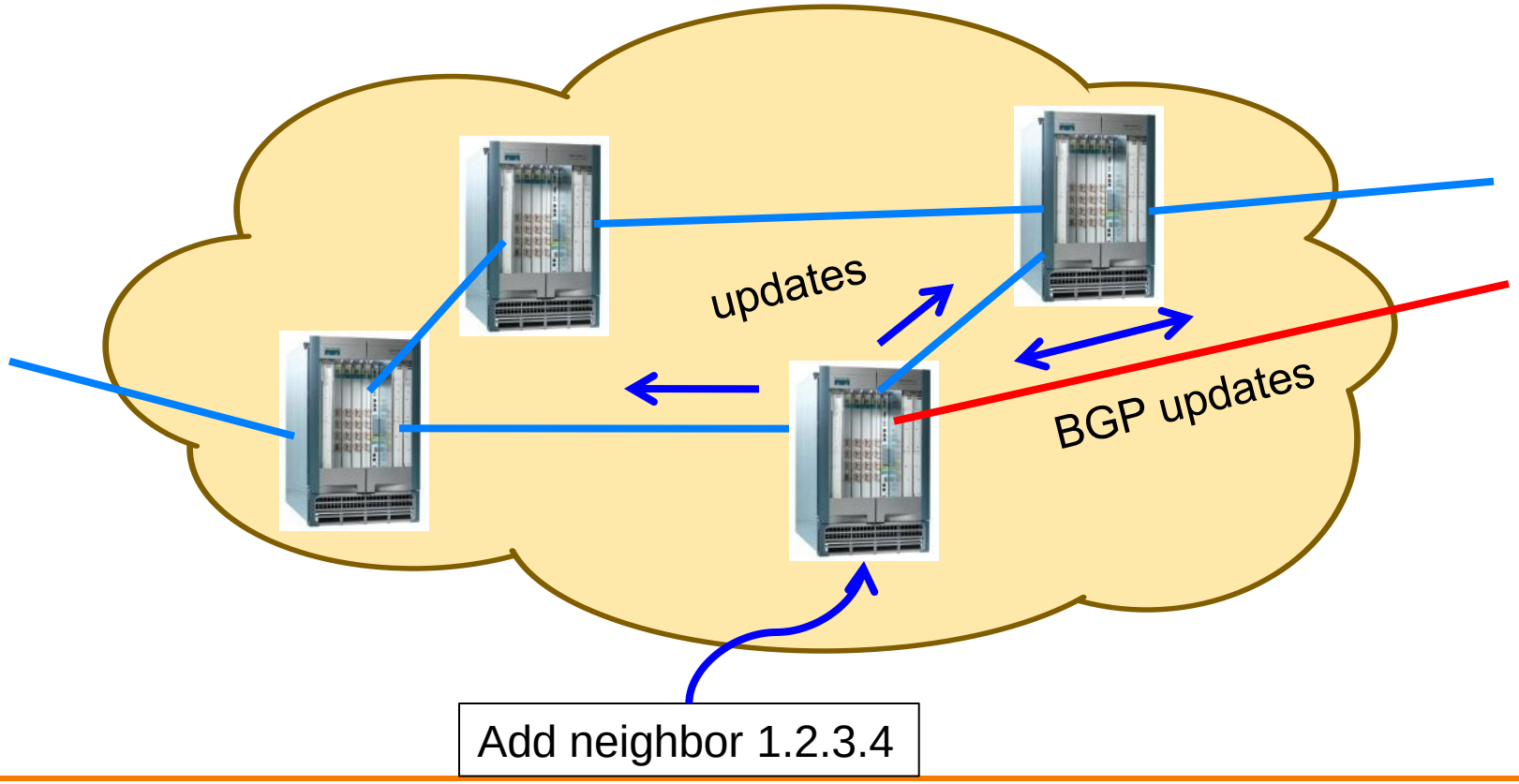
Understanding the Disruption (today)

1) Reconfigure old router, remove old link
2) Add new link link, configure new router
3)



Understanding the Disruption (today)

1) Reconfigure old router, remove old link
2) Add new link link, configure new router
3) Establish new BGP session (exchange routes)

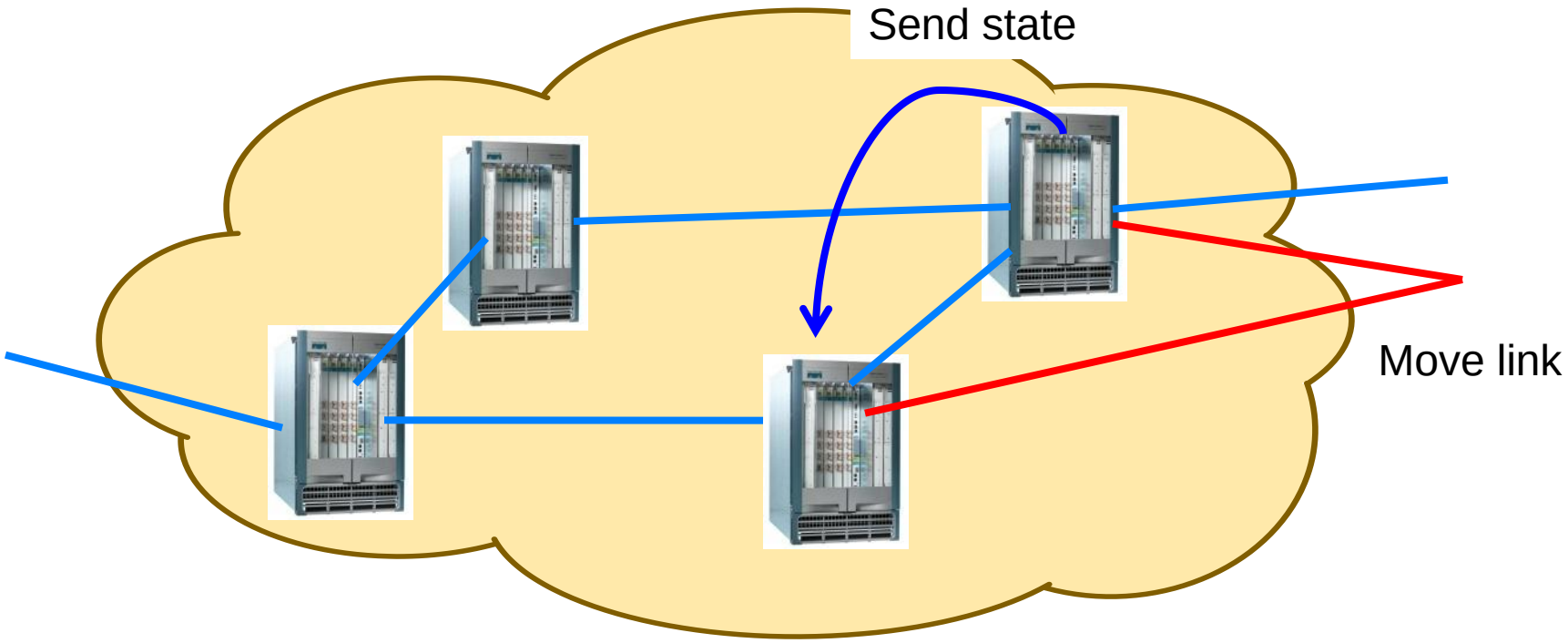


Understanding the Disruption (today)

1) Reconfigure old router, remove old link
2) Add new link link, configure new router
3) Establish new BGP session (exchange routes)

Downtime (Minutes)

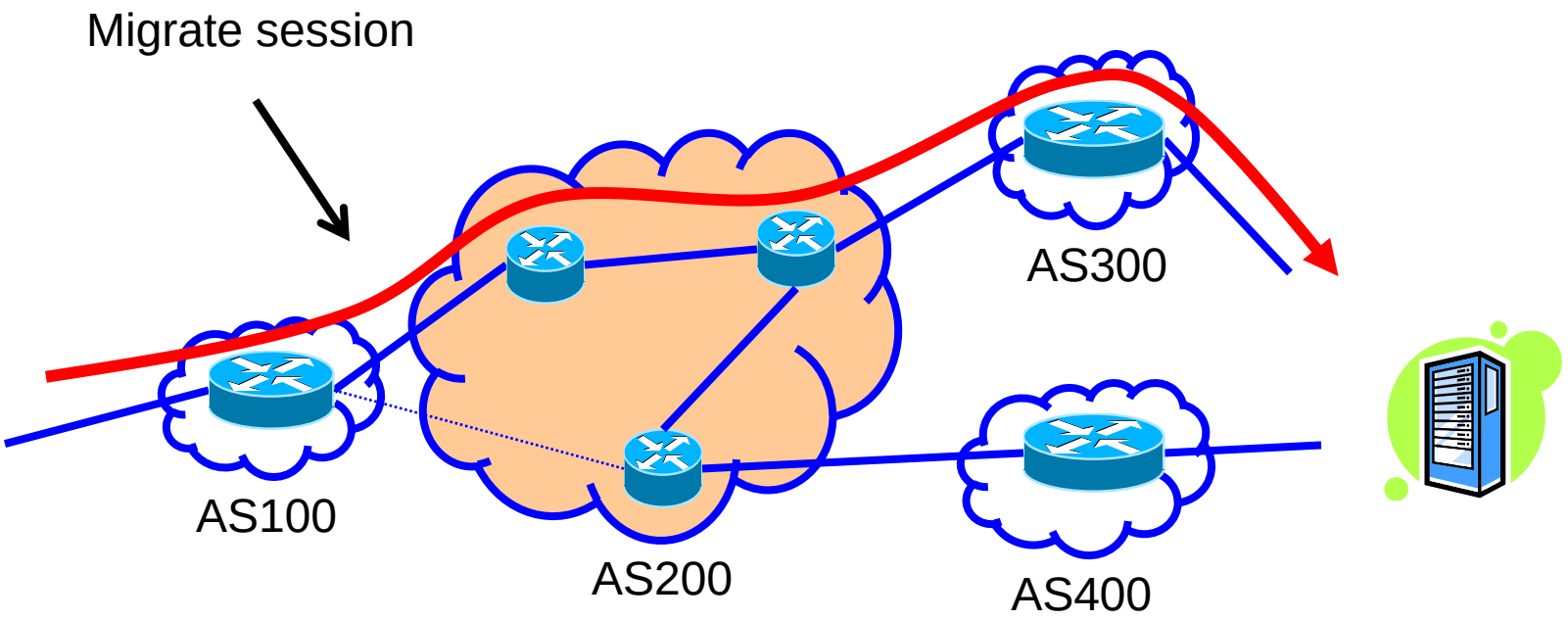
Router Grafting: Breaking up the router



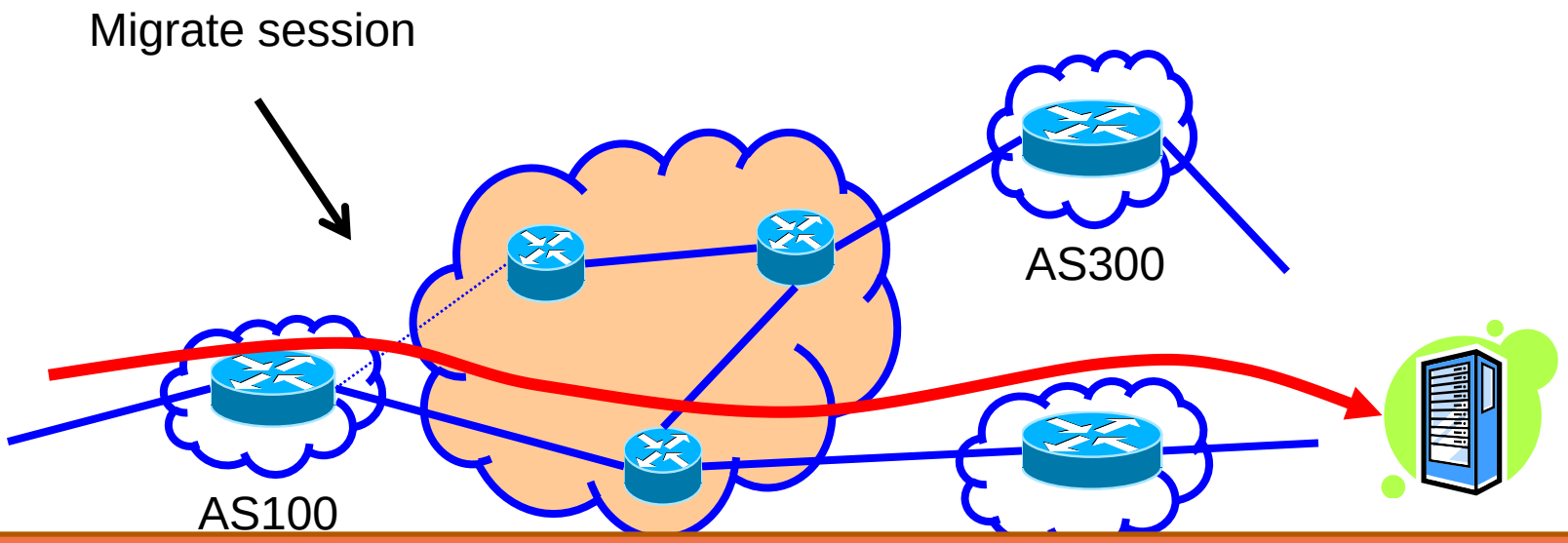
Router Grafting: Breaking up the router

Router Grafting enables this breaking apart a router (splitting/merging).

Not Just State Transfer



Not Just State Transfer



The topology changes

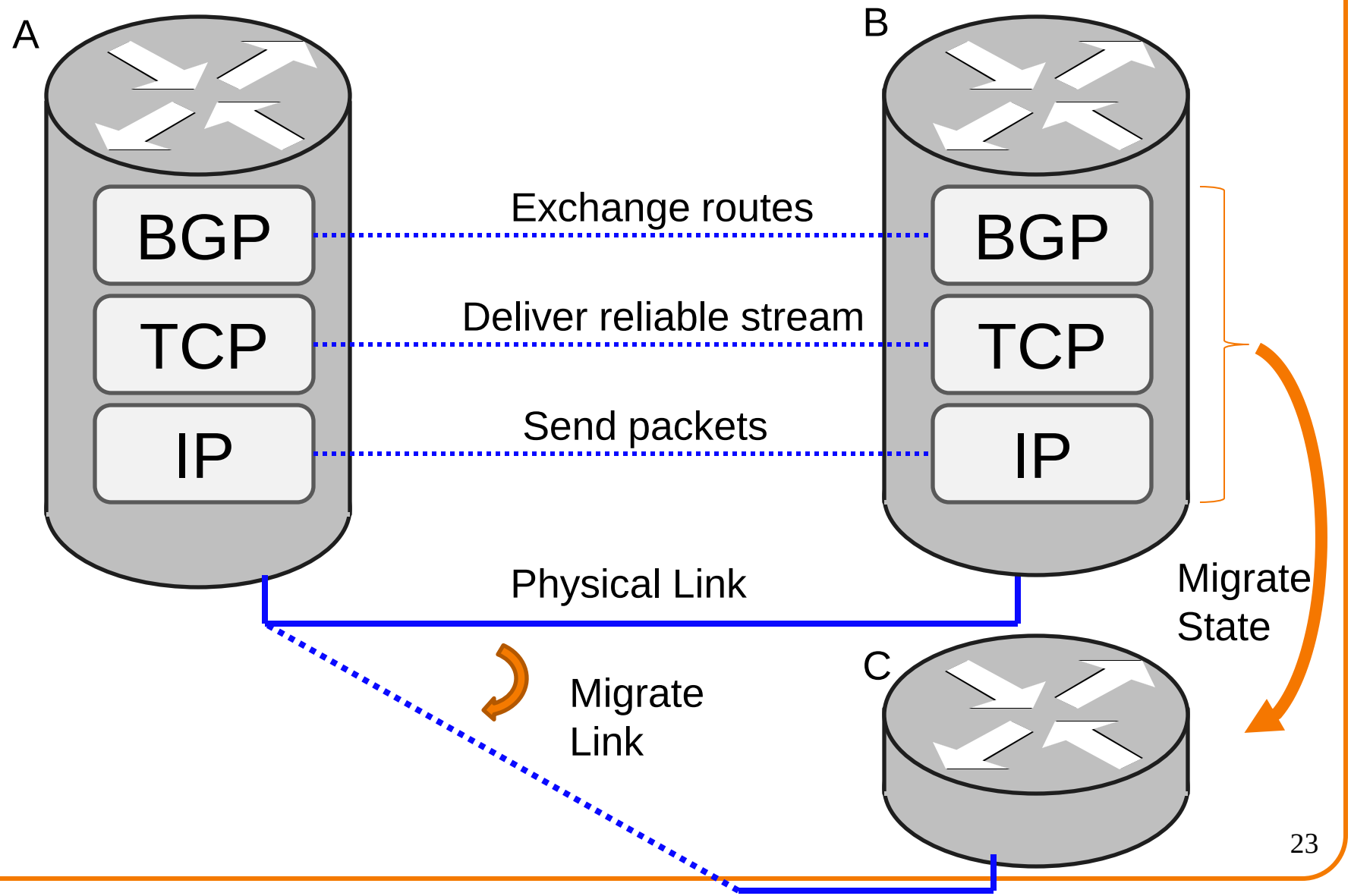
(Need to re-run decision processes)

Goals

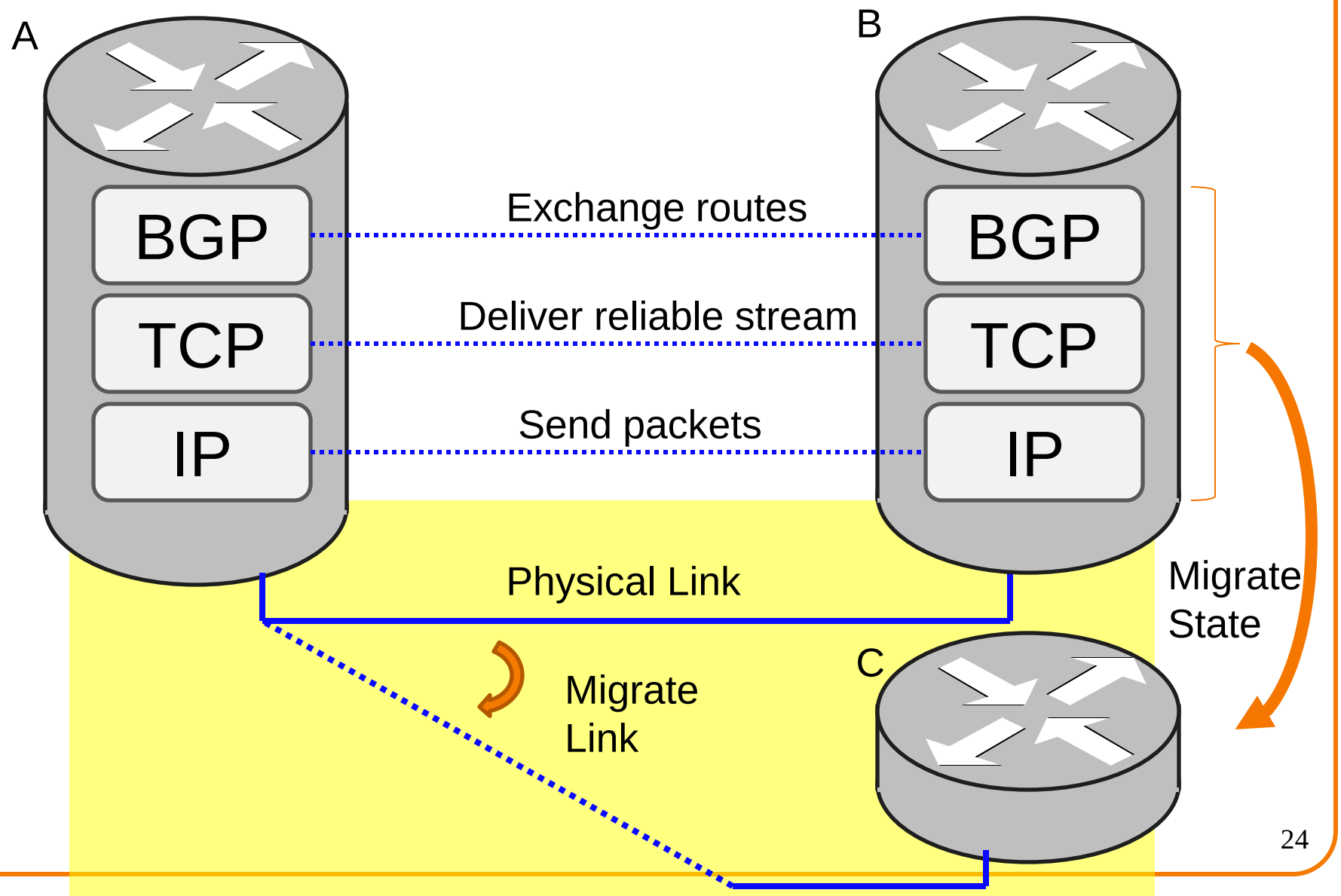
- Routing and forwarding should not be disrupted
  - Data packets are not dropped
  - Routing protocol adjacencies do not go down
  - All route announcements are received

- Change should be transparent
  - Neighboring routers/operators should not be involved
  - Redesign the routers not the protocols

Challenge: Protocol Layers

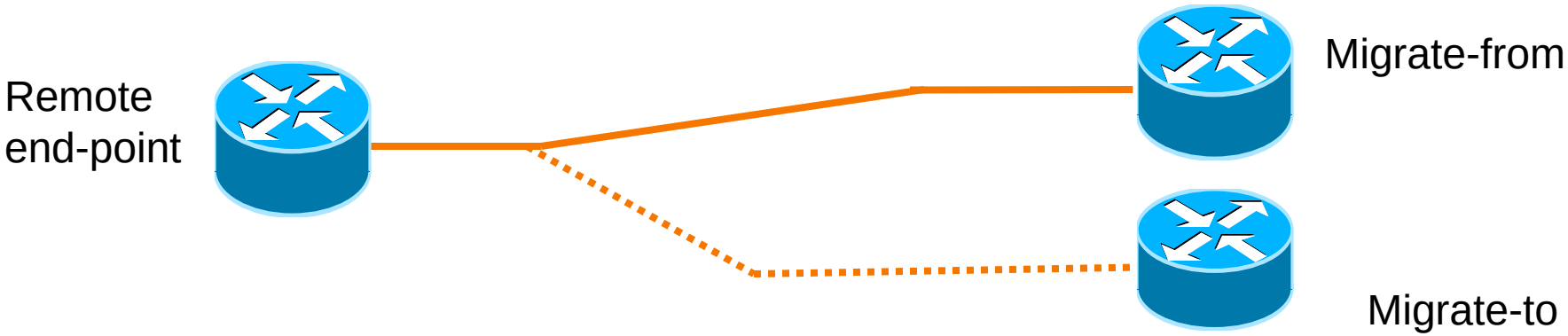


Physical Link



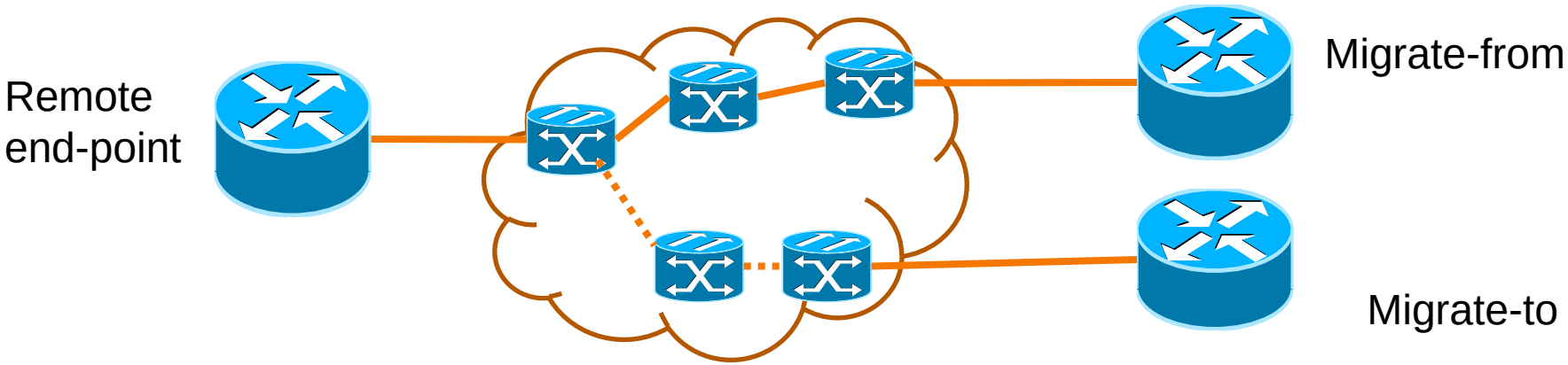
Physical Link

- Unplugging cable would be disruptive

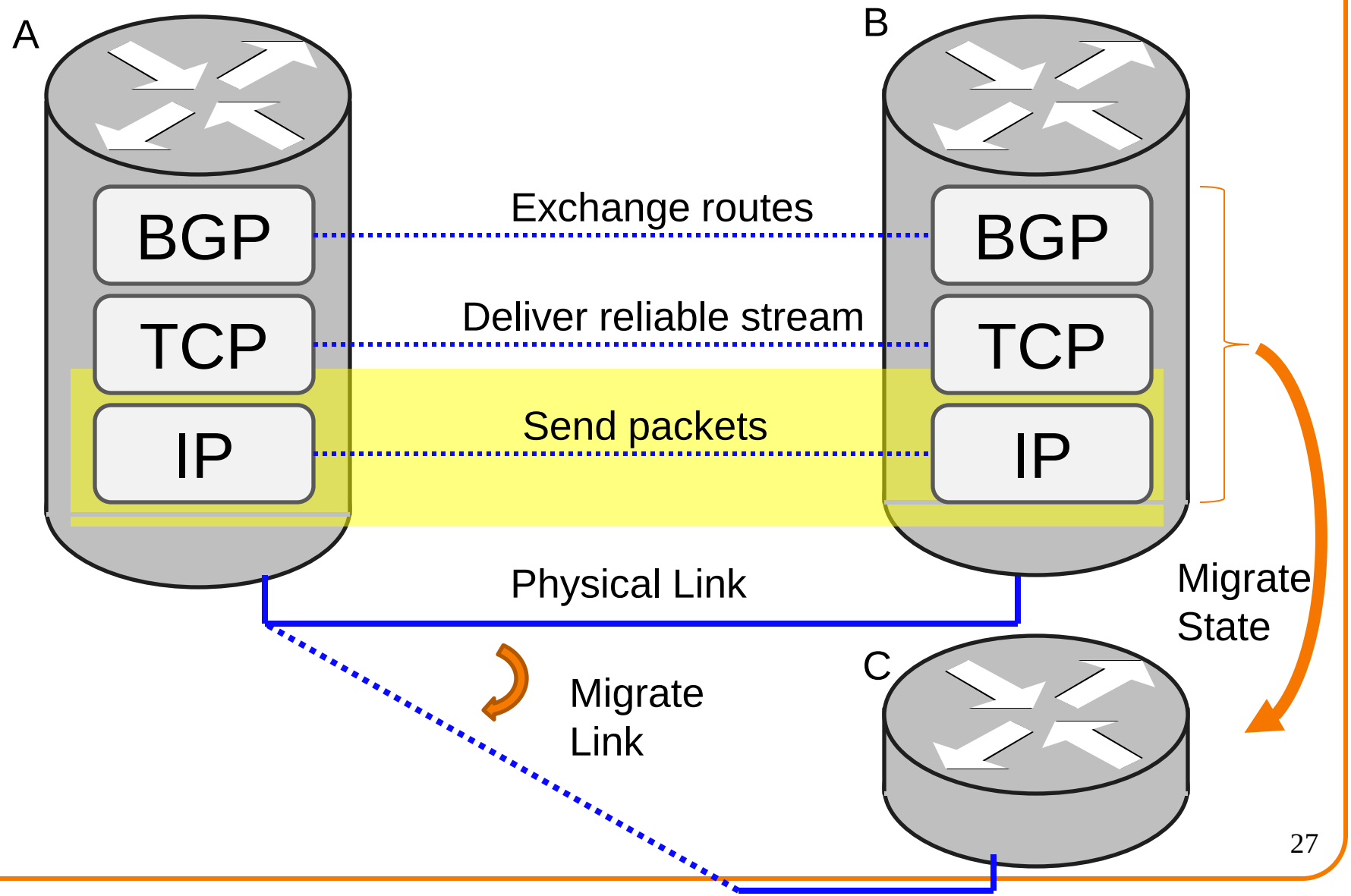


Physical Link

- Unplugging cable would be disruptive
- Links are not physical wires
  - Switchover in nanoseconds

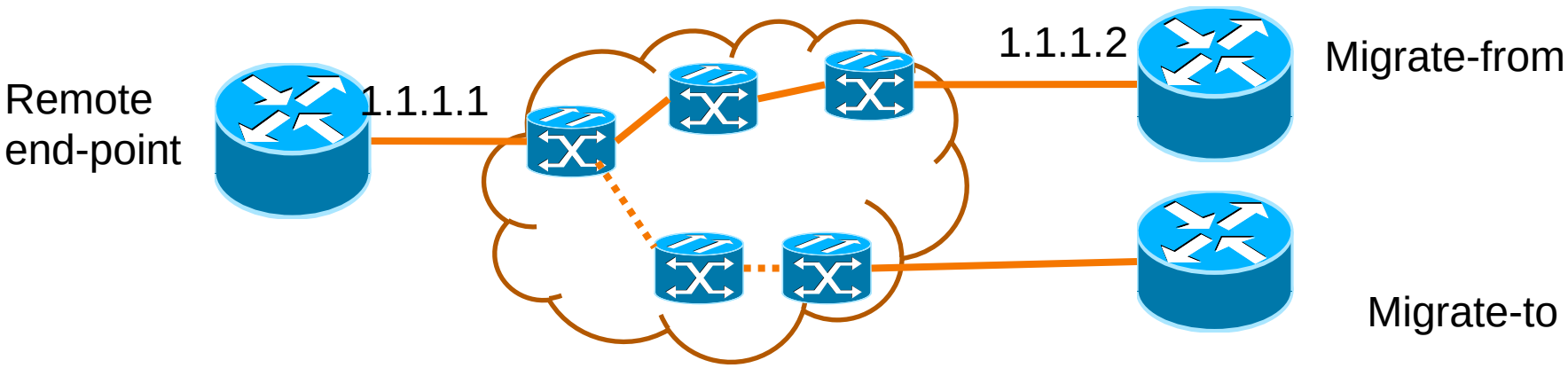


IP


A
B
BGP
TCP
IP
Exchange routes
Deliver reliable stream
Send packets
BGP
TCP
IP
Physical Link
Migrate
State
C
Migrate
Link

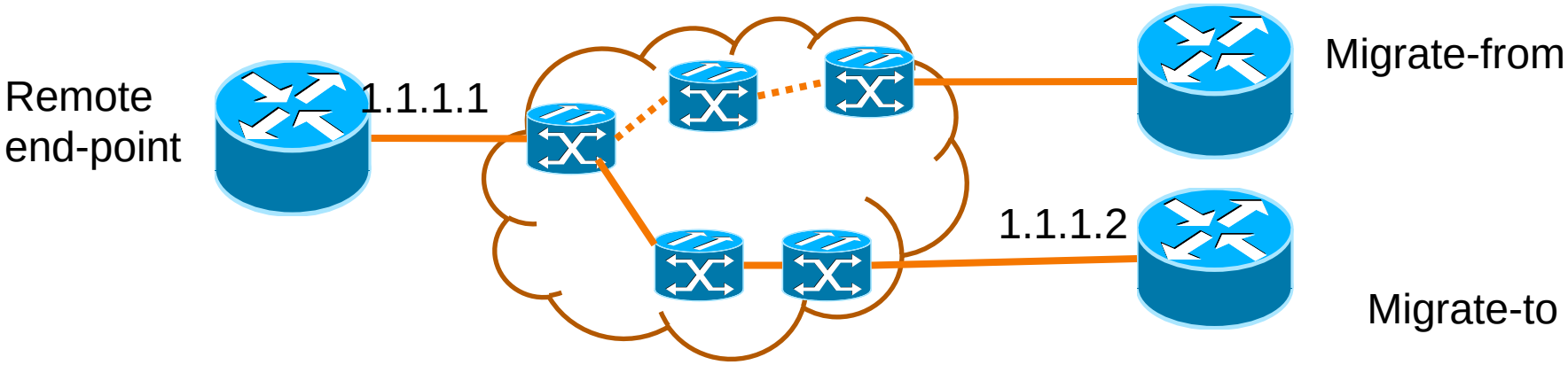
Changing IP Address

- IP address is an identifier in BGP
- Changing it would require neighbor to reconfigure
  - Not transparent
  - Also has impact on TCP (later)

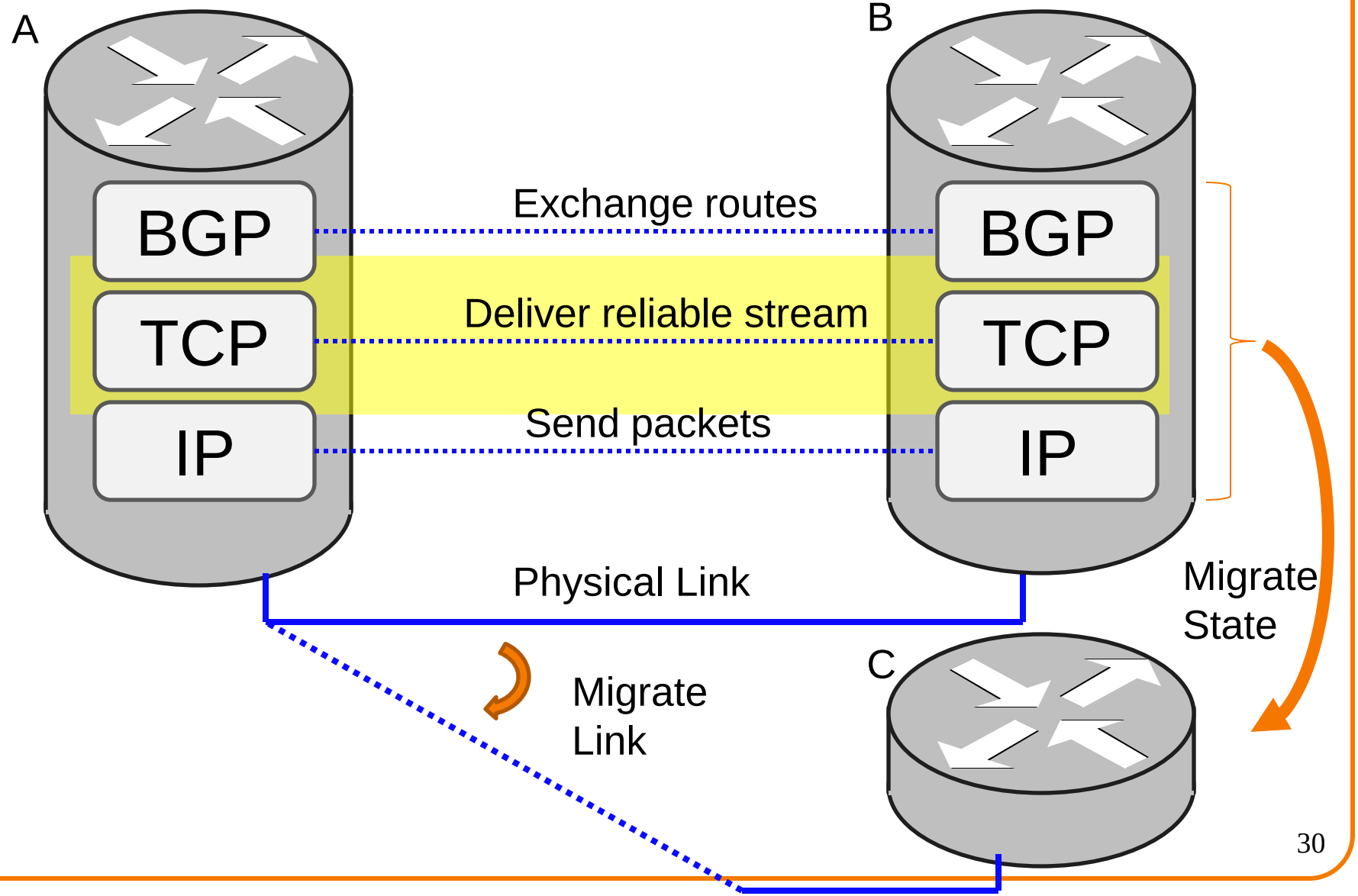


Re-assign IP Address

- IP address not used for global reachability
  - Can move with BGP session
  - Neighbor doesn't have to reconfigure



TCP

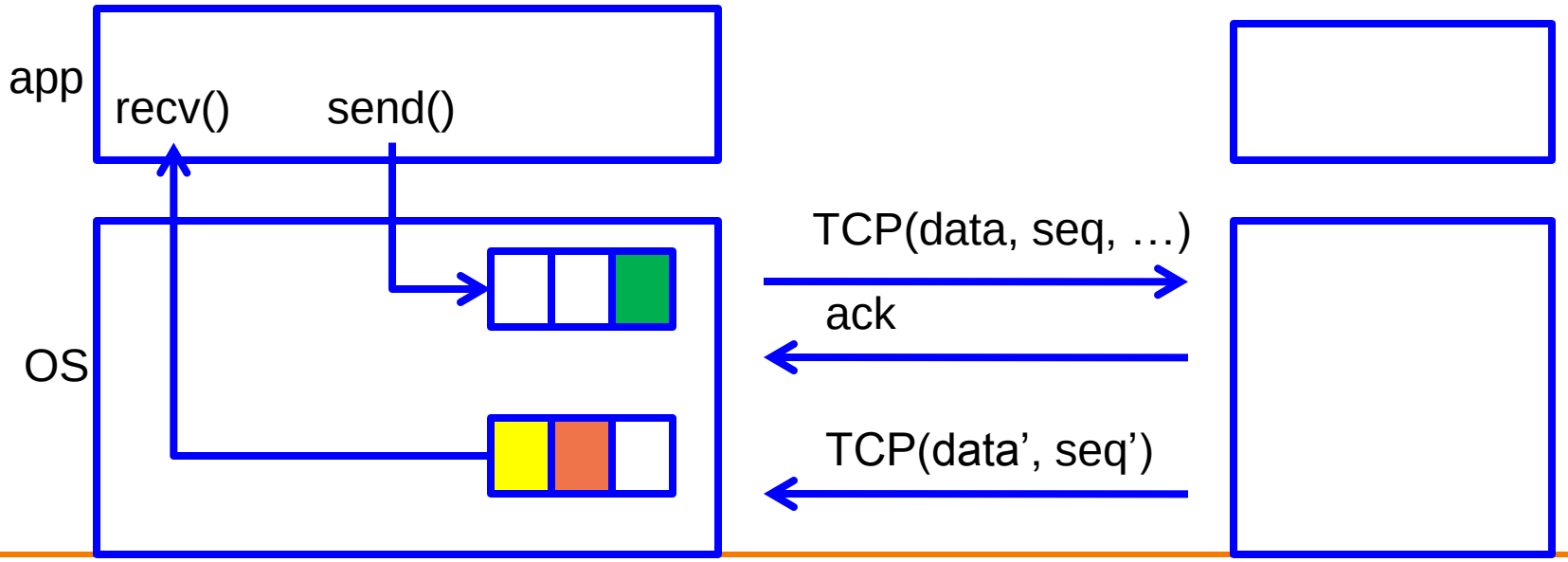

A
B
BGP
TCP
IP
Exchange routes
Deliver reliable stream
Send packets
BGP
TCP
IP
Physical Link
Migrate State
C
Migrate Link

Dealing with TCP

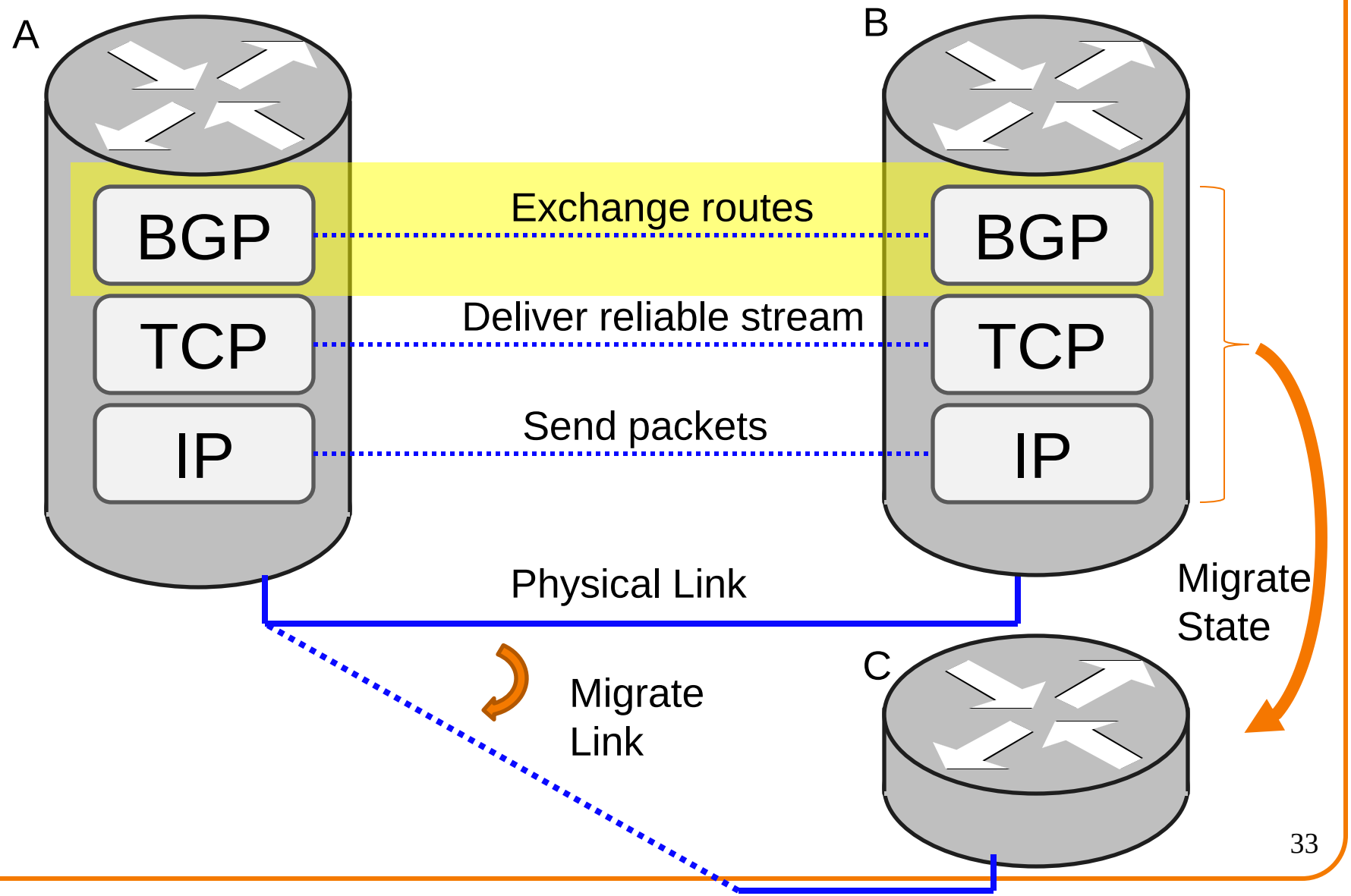
- TCP sessions are long running in BGP
  - Killing it implicitly signals the router is down
- BGP and TCP extensions as a workaround (not supported on all routers)

Migrating TCP Transparently

- Capitalize on IP address not changing
  - To keep it completely transparent
- Transfer the TCP session state
  - Sequence numbers
  - Packet input/output queue (packets not read/ack'd)



BGP

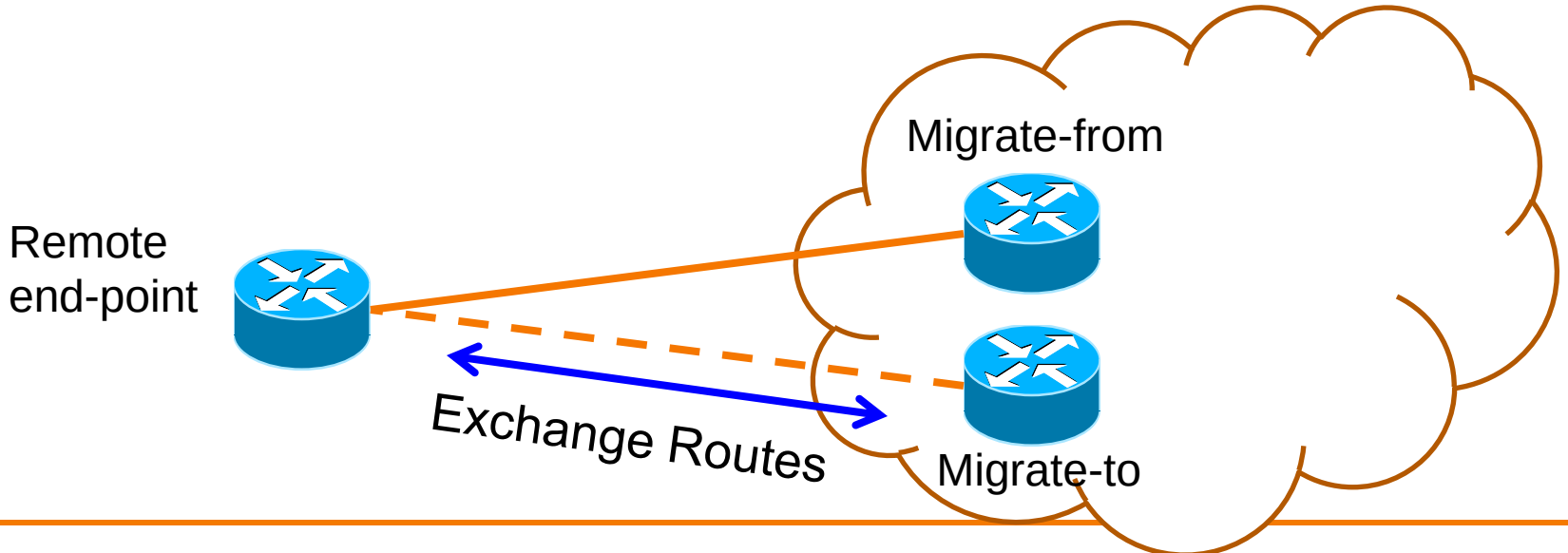

A
B
BGP
TCP
IP
Exchange routes
Deliver reliable stream
Send packets
BGP
TCP
IP
Physical Link
Migrate
State
C
Migrate
Link

BGP: What (not) to Migrate

- Requirements
  - Want data packets to be delivered
  - Want routing adjacencies to remain up
- Need
  - Configuration
  - Routing information
- Do not need (but can have)
  - State machine
  - Statistics
  - Timers
- Keeps code modifications to a minimum

Routing Information

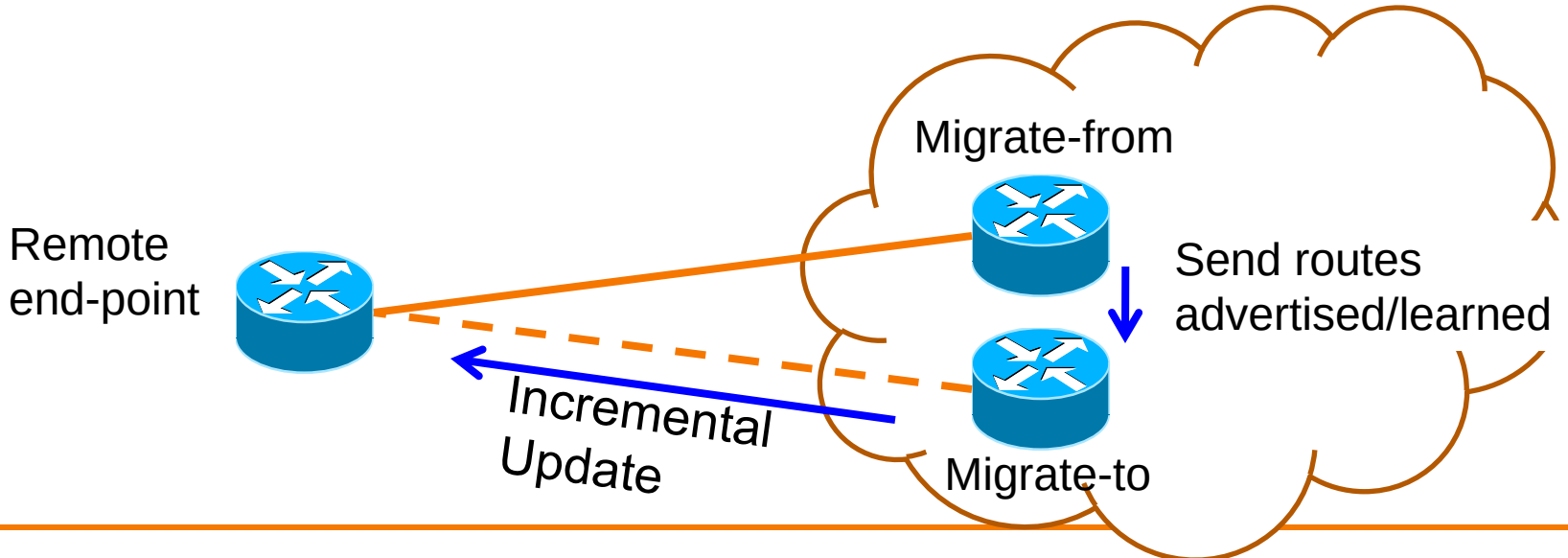
- Could involve remote end-point
  - Similar exchange as with a new BGP session
  - Migrate-to router sends entire state to remote end-point
  - Ask remote-end point to re-send all routes it advertised
- Disruptive
  - Makes remote end-point do significant work



Routing Information (optimization)

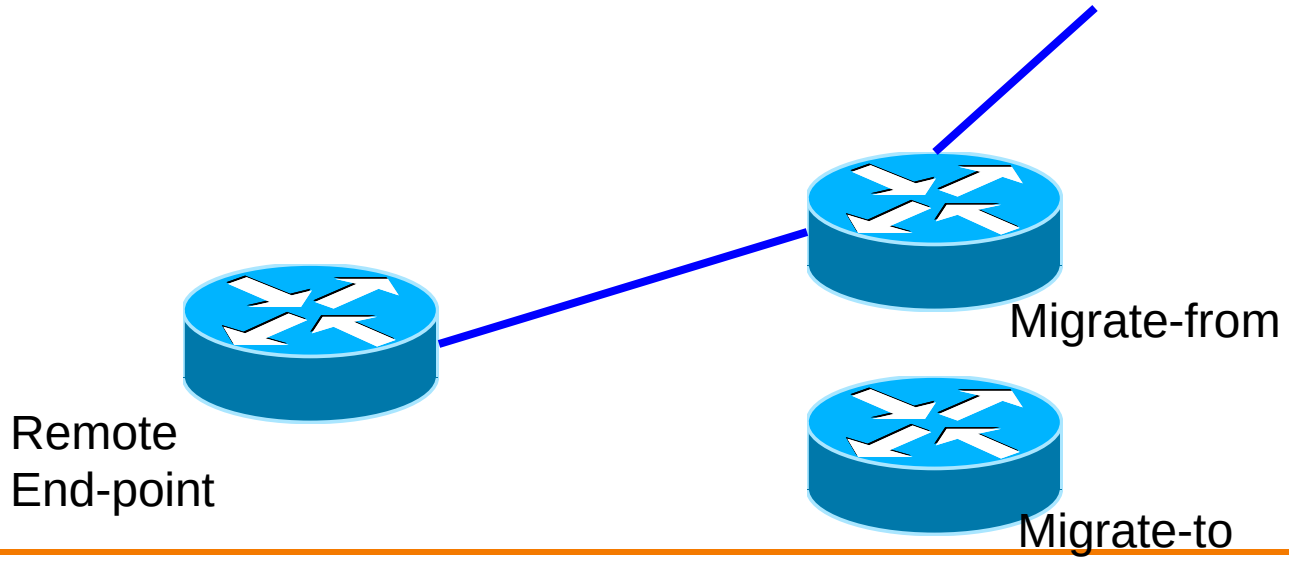
Migrate-from router send the migrate-to router:

- The routes it learned
  - Instead of making remote end-point re-announce
- The routes it advertised
  - So able to send just an incremental update



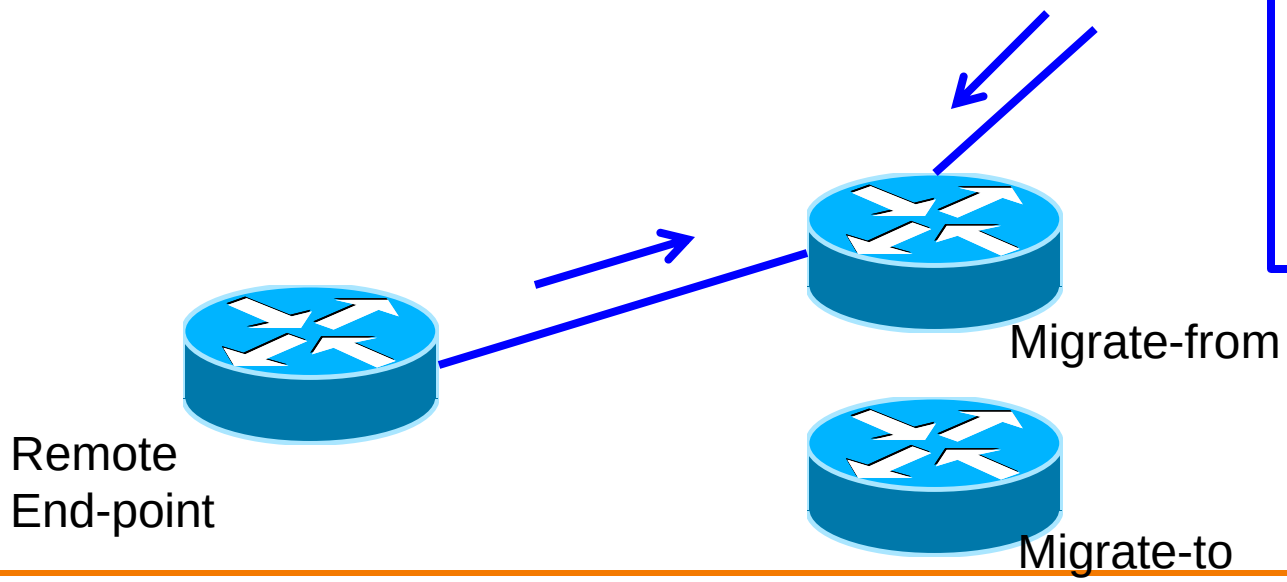
Migration in The Background

- Migration takes a while
  - A lot of routing state to transfer
  - A lot of processing is needed
- Routing changes can happen at any time
- Disruptive if not done in the background



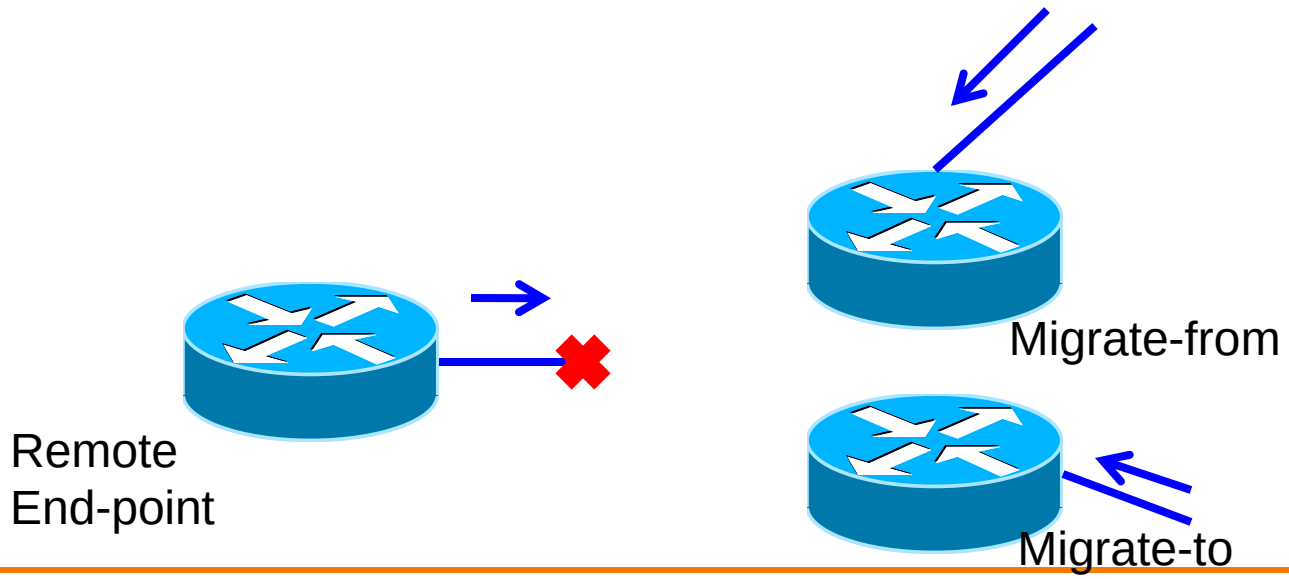
While exporting routing state

BGP is incremental, append update



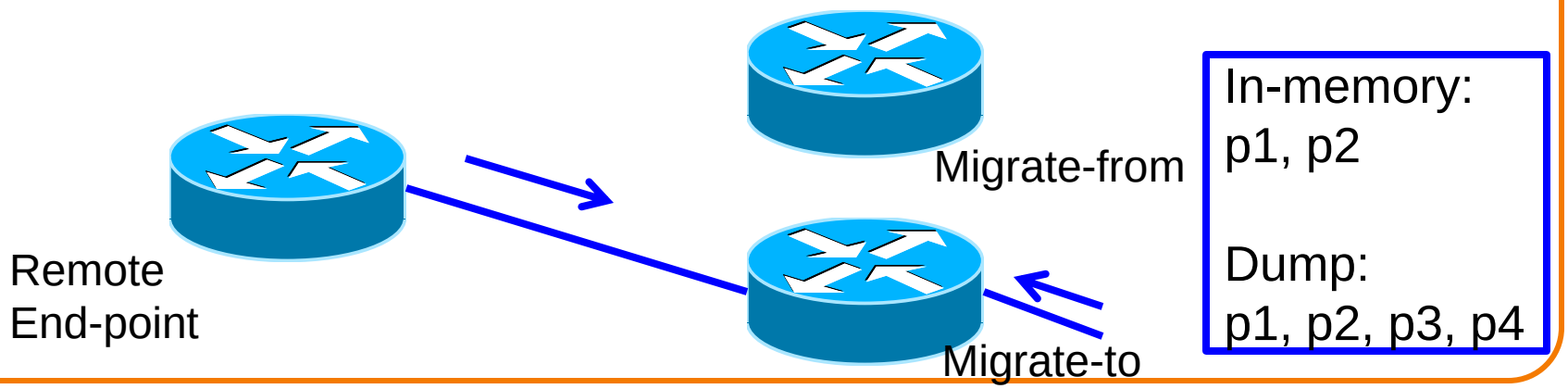
While moving TCP session and link

TCP will retransmit



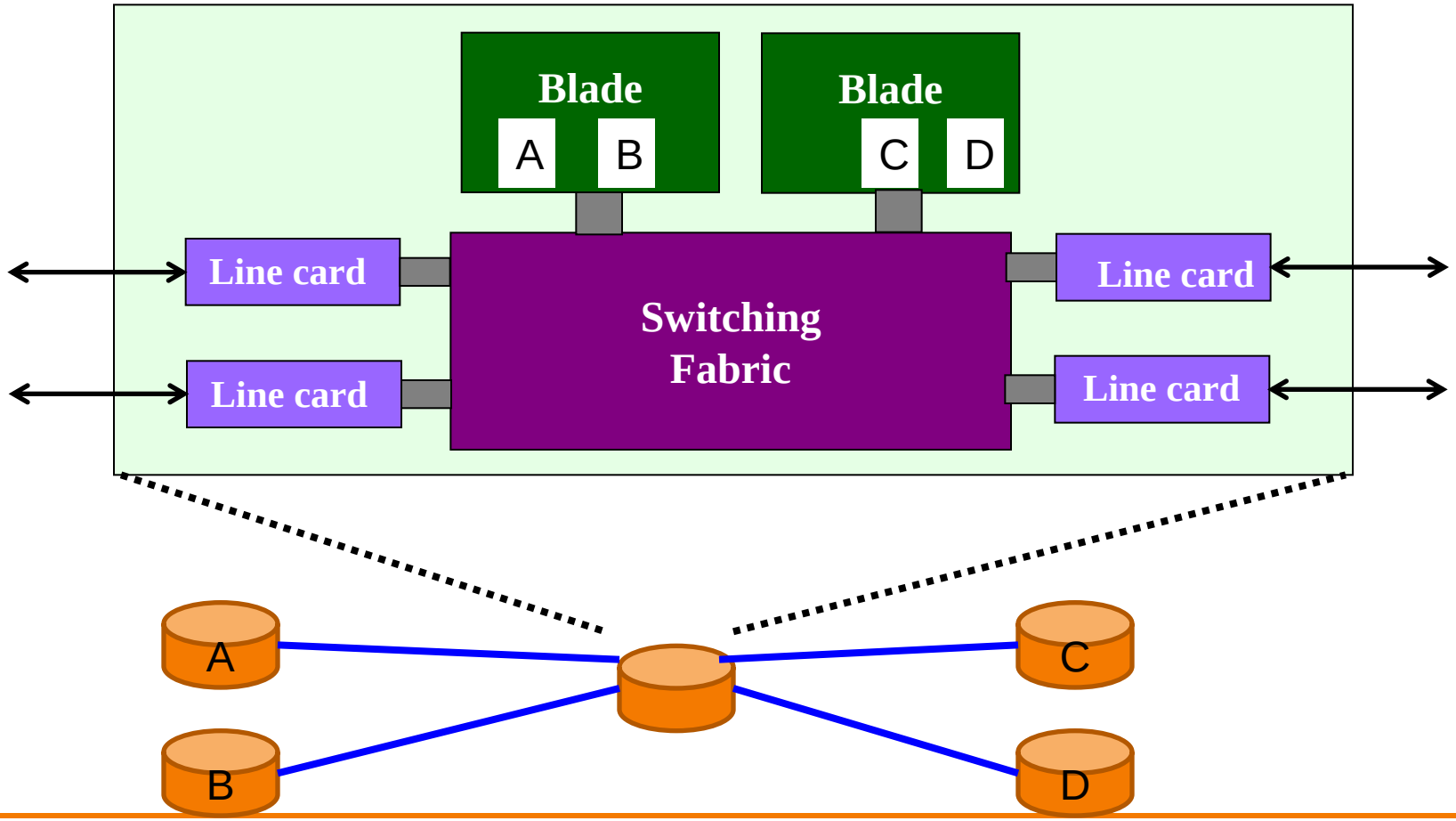
While importing routing state

BGP is incremental, ignore dump file



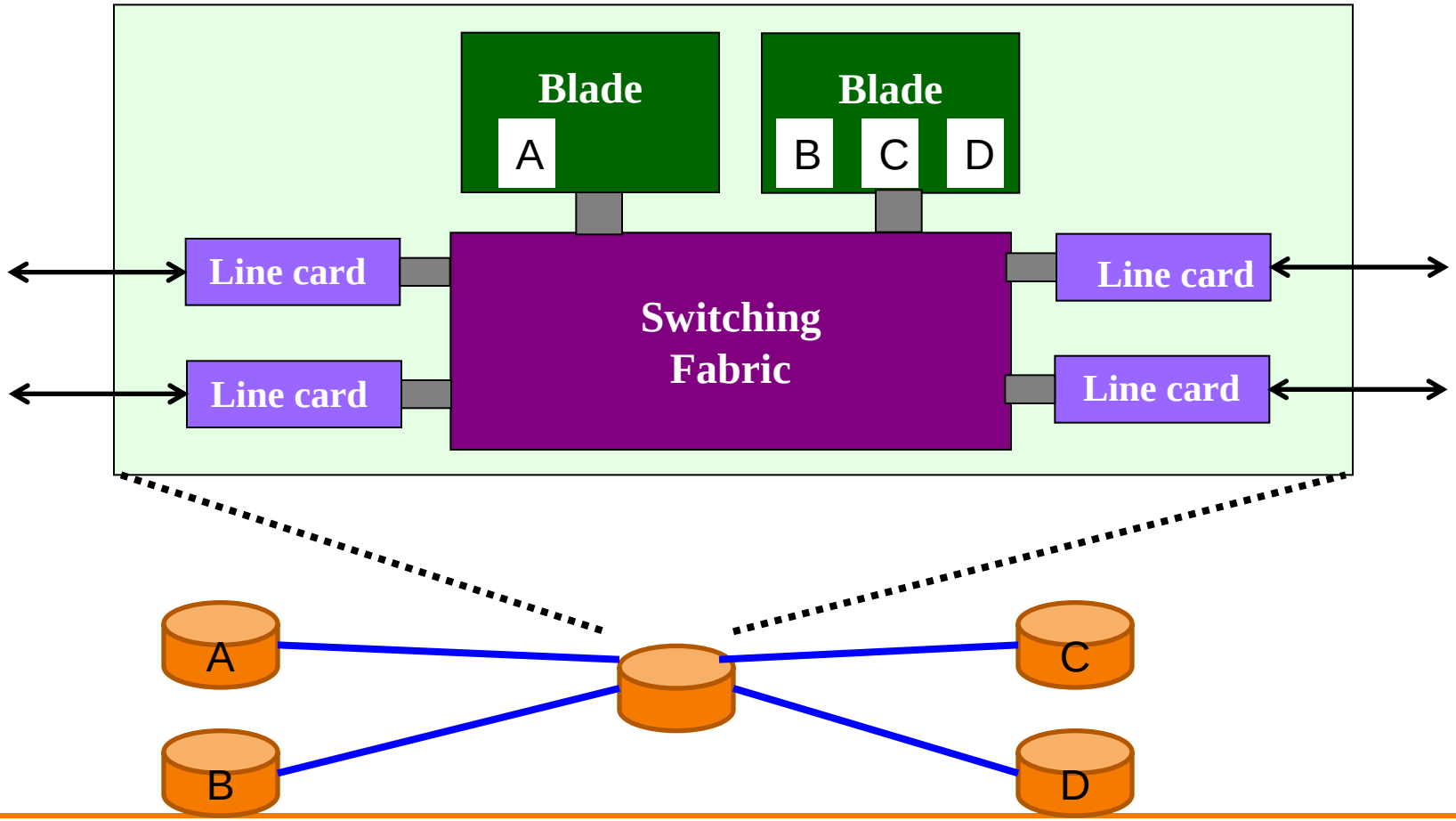
Special Case: Cluster Router

- Don't need to re-run decision processes
- Links 'migrated' internally



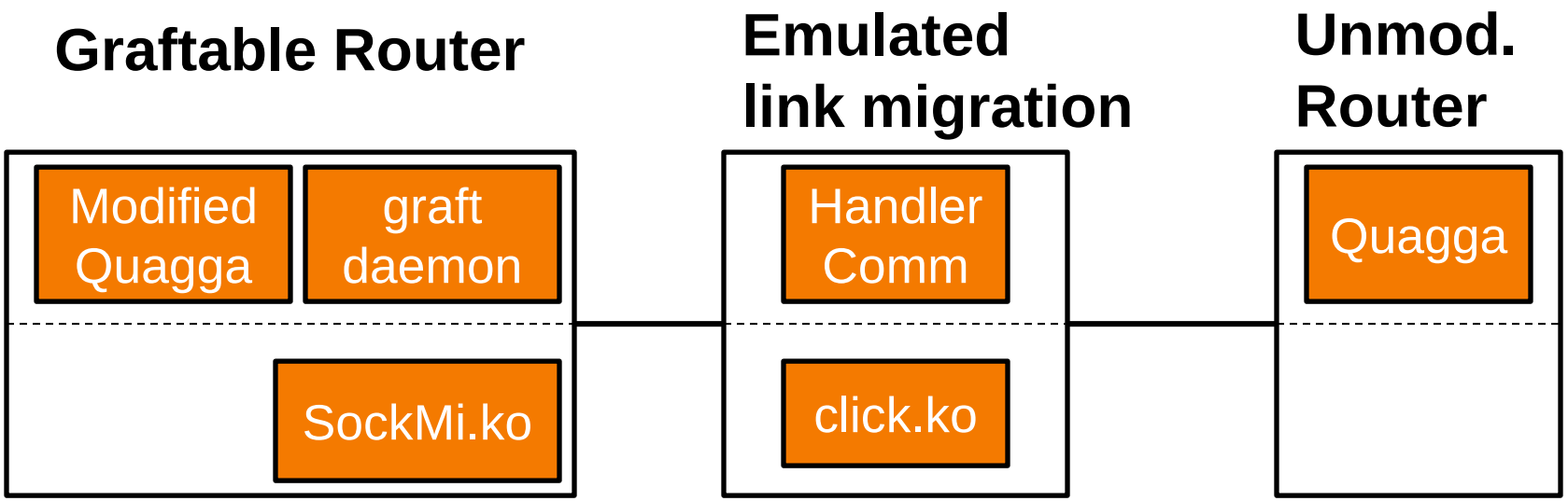
Special Case: Cluster Router

- Don't need to re-run decision processes
- Links 'migrated' internally



Prototype

- Added grafting into Quagga
  - Import/export routes, new ‘inactive’ state
  - Routing data and decision process well separated
- Graft daemon to control process
- SockMi for TCP migration



Evaluation

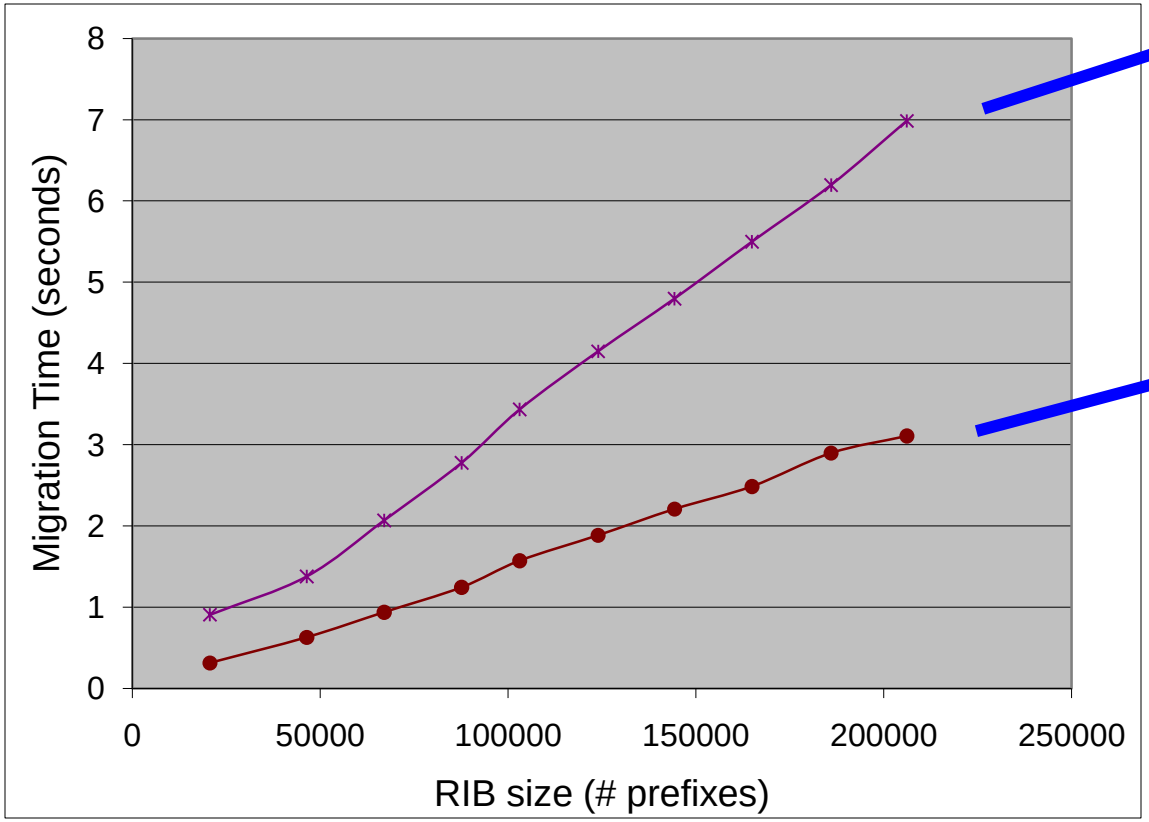
- Impact on migrating routers
- Disruption to network operation
- Overhead on rest of the network

Evaluation

- Impact on migrating routers
- Disruption to network operation
- Overhead on rest of the network

Impact on Migrating Routers

- How long migration takes
  - Includes export, transmit, import, lookup, decision
  - CPU Utilization roughly 25%



Between Routers
0.9s (20k)
6.9s (200k)

Between Blades
0.3s (20k)
3.1s (200k)

Disruption to Network Operation

- Data traffic affected by not having a link
  - nanoseconds
- Routing protocols affected by unresponsiveness
  - Set old router to “inactive”, migrate link, migrate TCP, set new router to “active”
  - milliseconds

Conclusions and Future Work

- Enables moving a single link/session with…
  - Minimal code change
  - No impact on data traffic
  - No visible impact on routing protocol adjacencies
  - Minimal overhead on rest of network

- Future work
  - Explore applications
  - Generalize grafting
    (multiple sessions, different protocols, other resources)

Questions?

Contact info:

ekeller@princeton.edu

http://www.princeton.edu/~ekeller