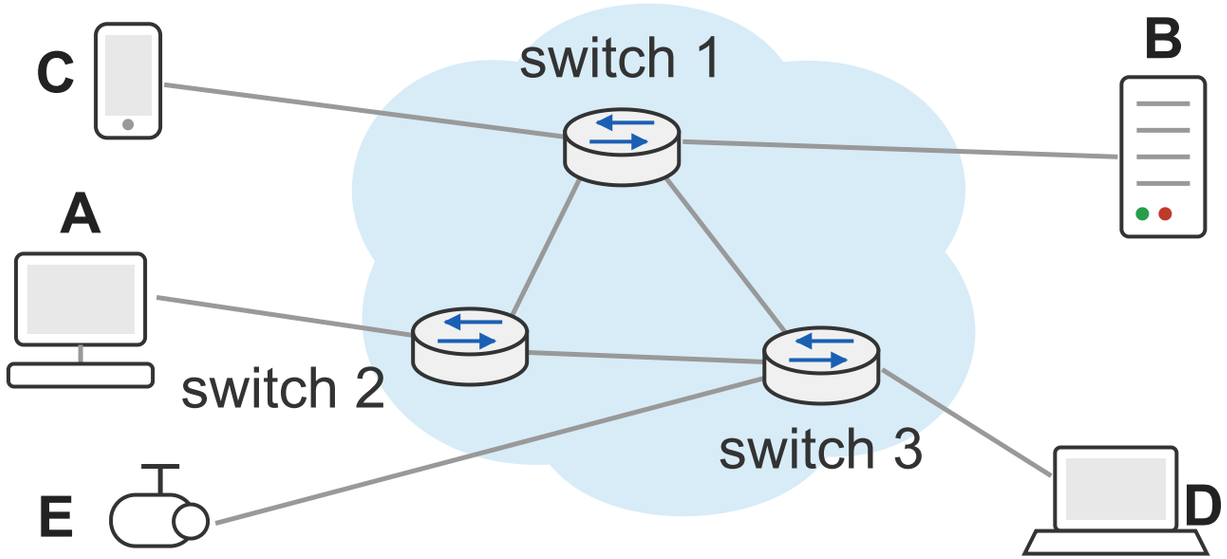
UNIVERSITY OF WATERLOO

Finding paths - Part I

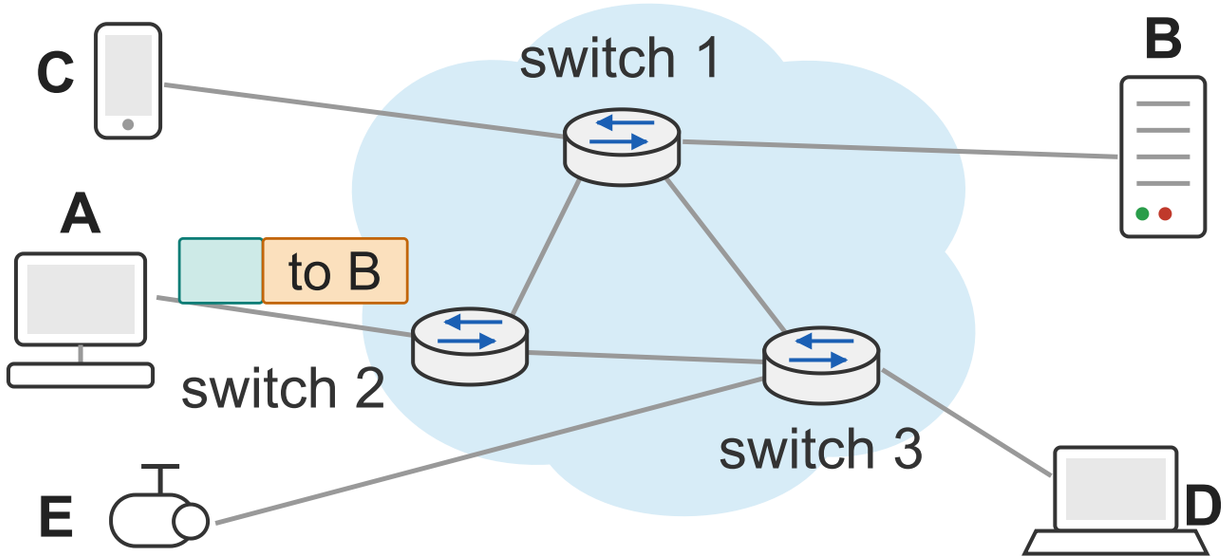
CS 456/656, Fall 2026

Mina Tahmasbi Arashloo

How does the network get a packet from computer A to computer B?

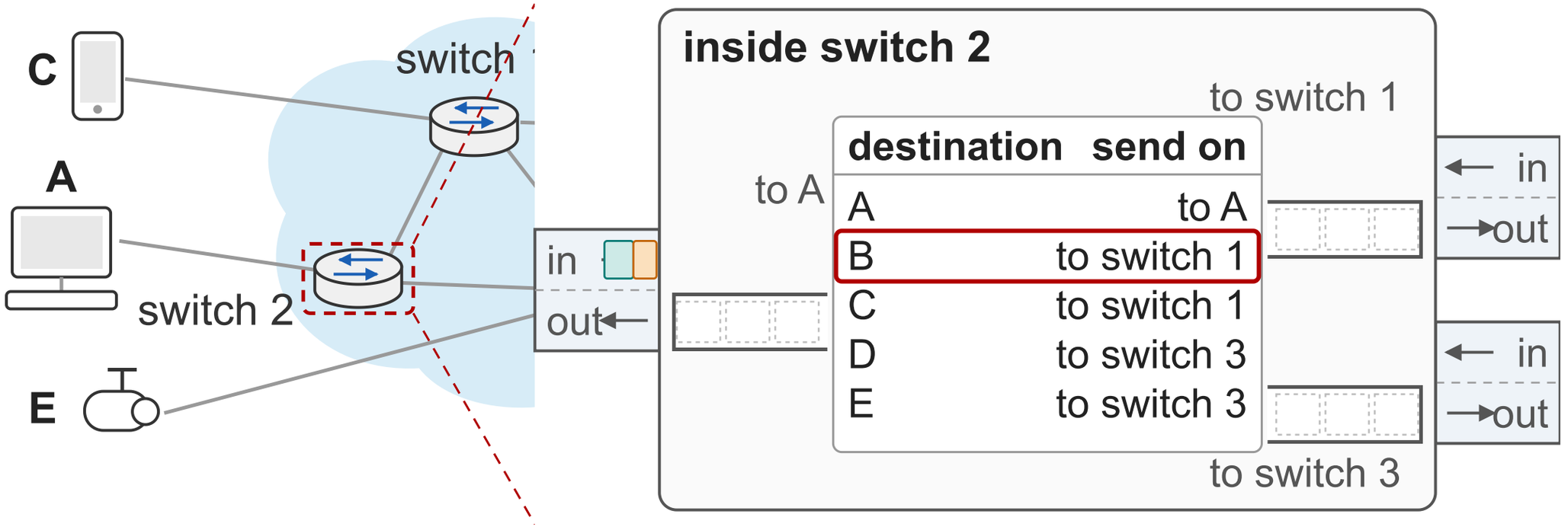


How does the network get a packet from computer A to computer B?



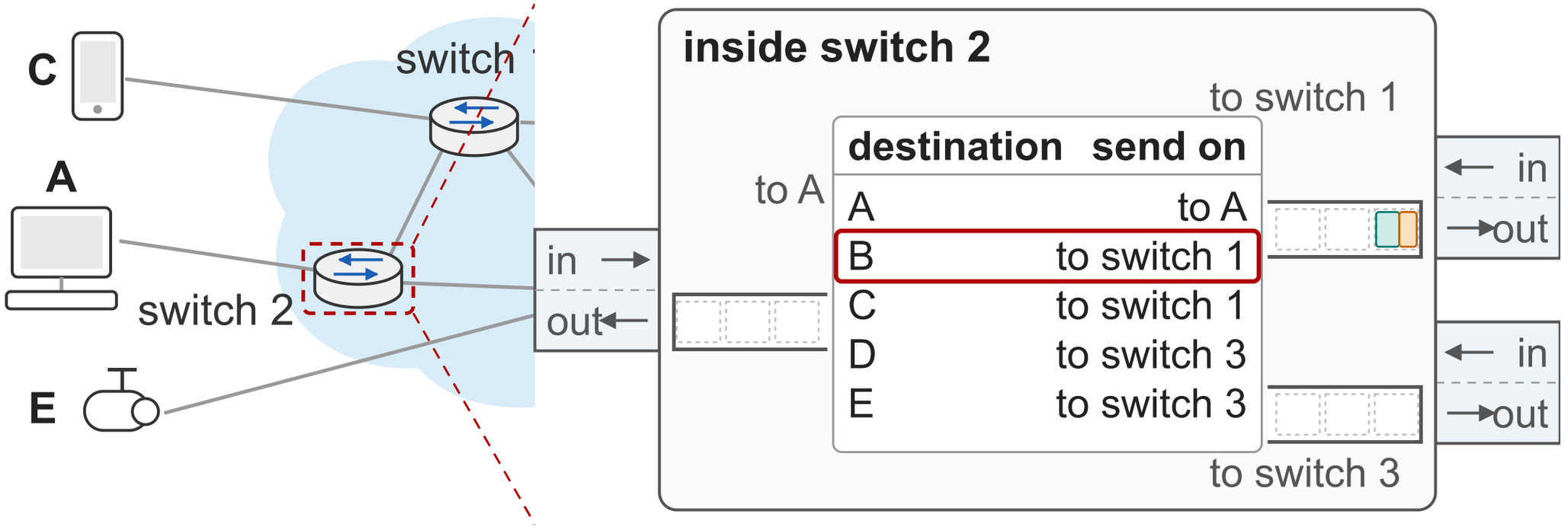
A sends a packet to B over its link to switch 2

How does the network get a packet from computer A to computer B?



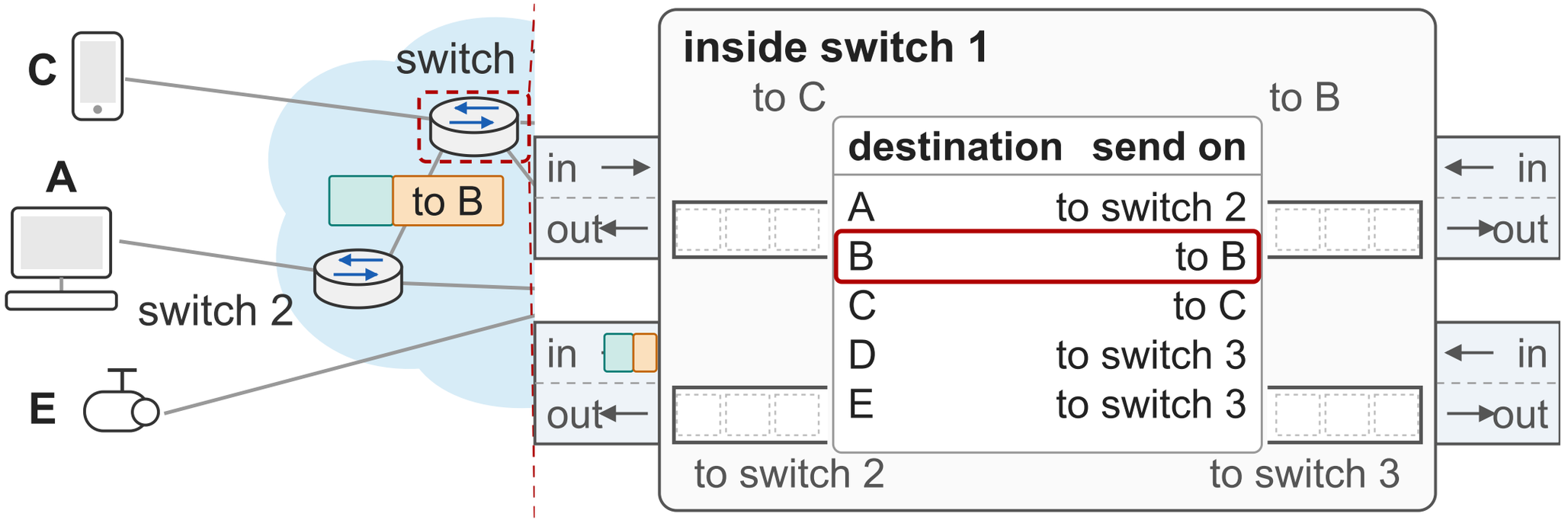
Switch 2 looks up B in its forwarding table.
Sends the packet on the link to switch 1

How does the network get a packet from computer A to computer B?



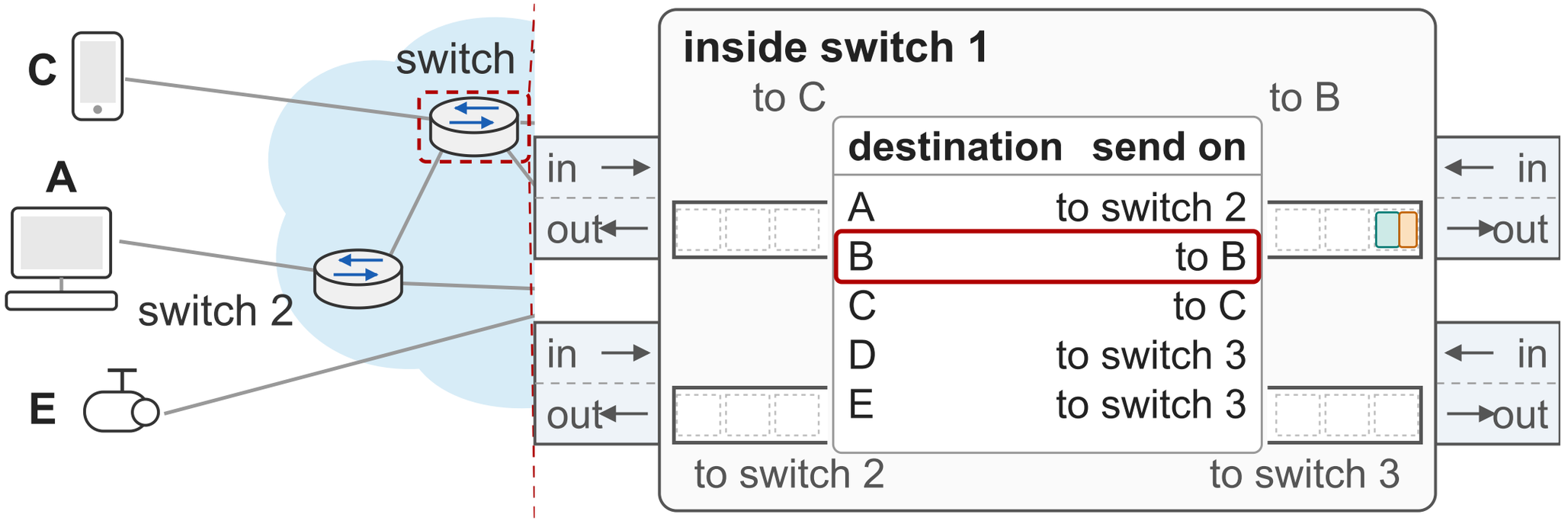
The packet is buffered in that link's queue
to wait for its turn to be transmitted

How does the network get a packet from computer A to computer B?



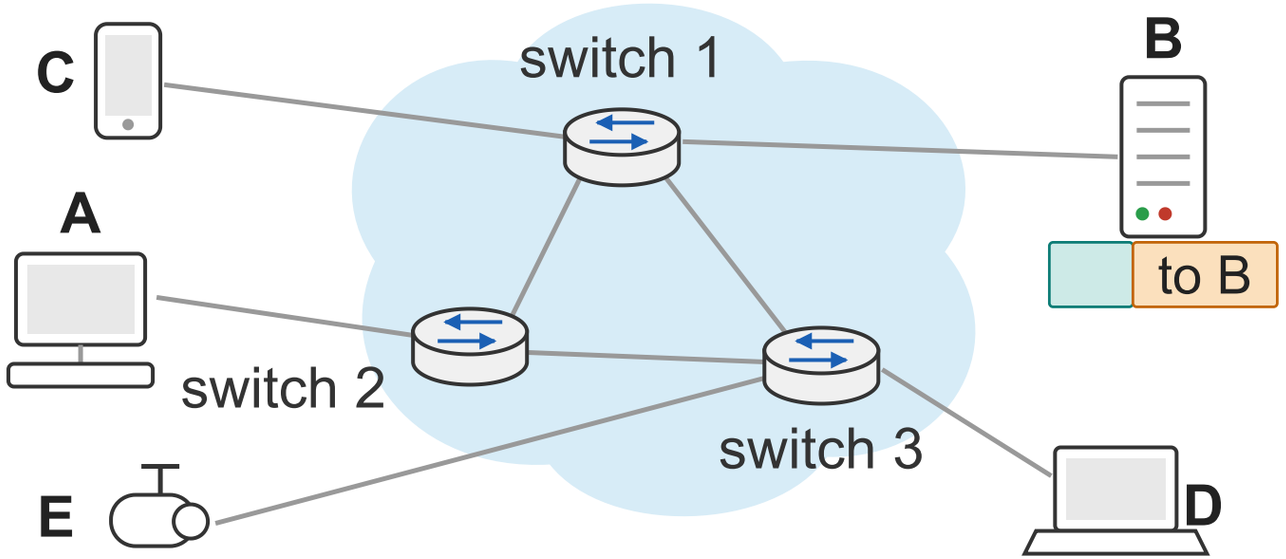
Switch 1 looks up B in its forwarding table.
Sends the packet on the link to B

How does the network get a packet from computer A to computer B?



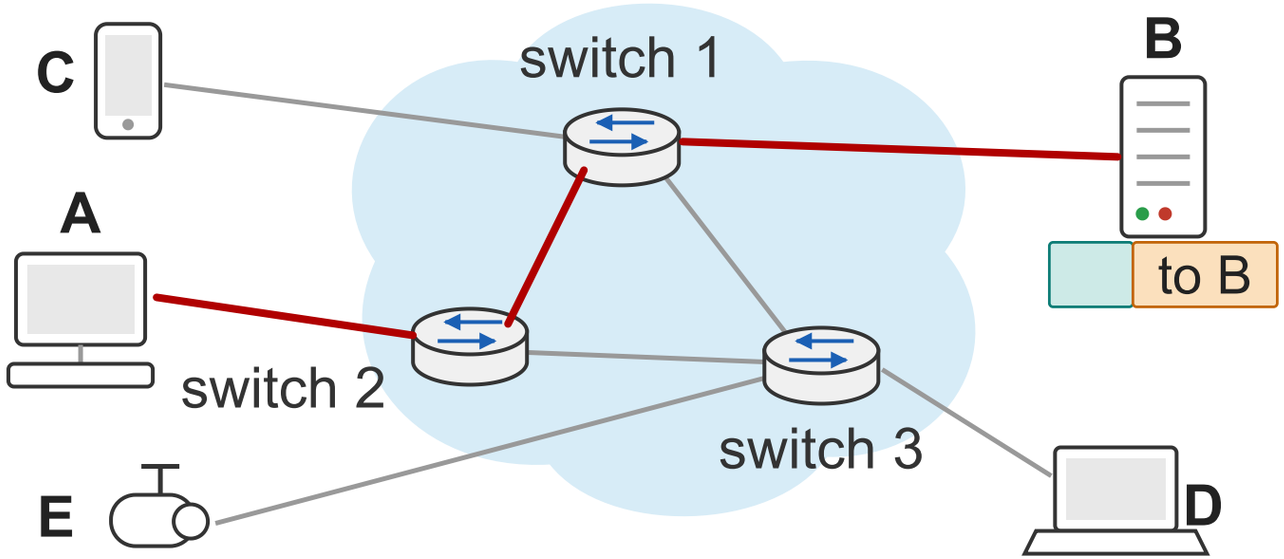
The packet is buffered in that link's queue
to wait for its turn to be transmitted

How does the network get a packet from computer A to computer B?



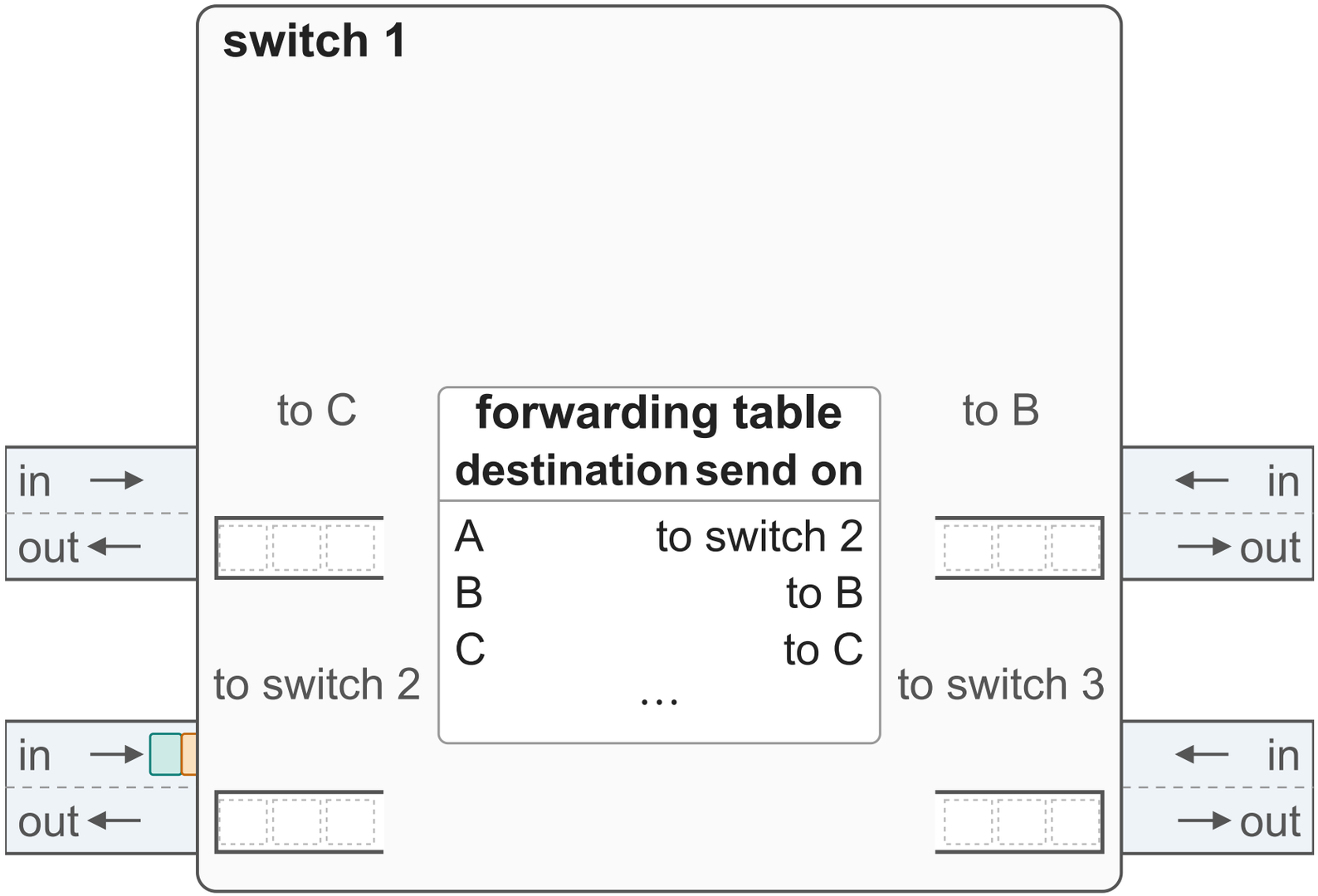
B receives the packet

How does the network get a packet from computer A to computer B?

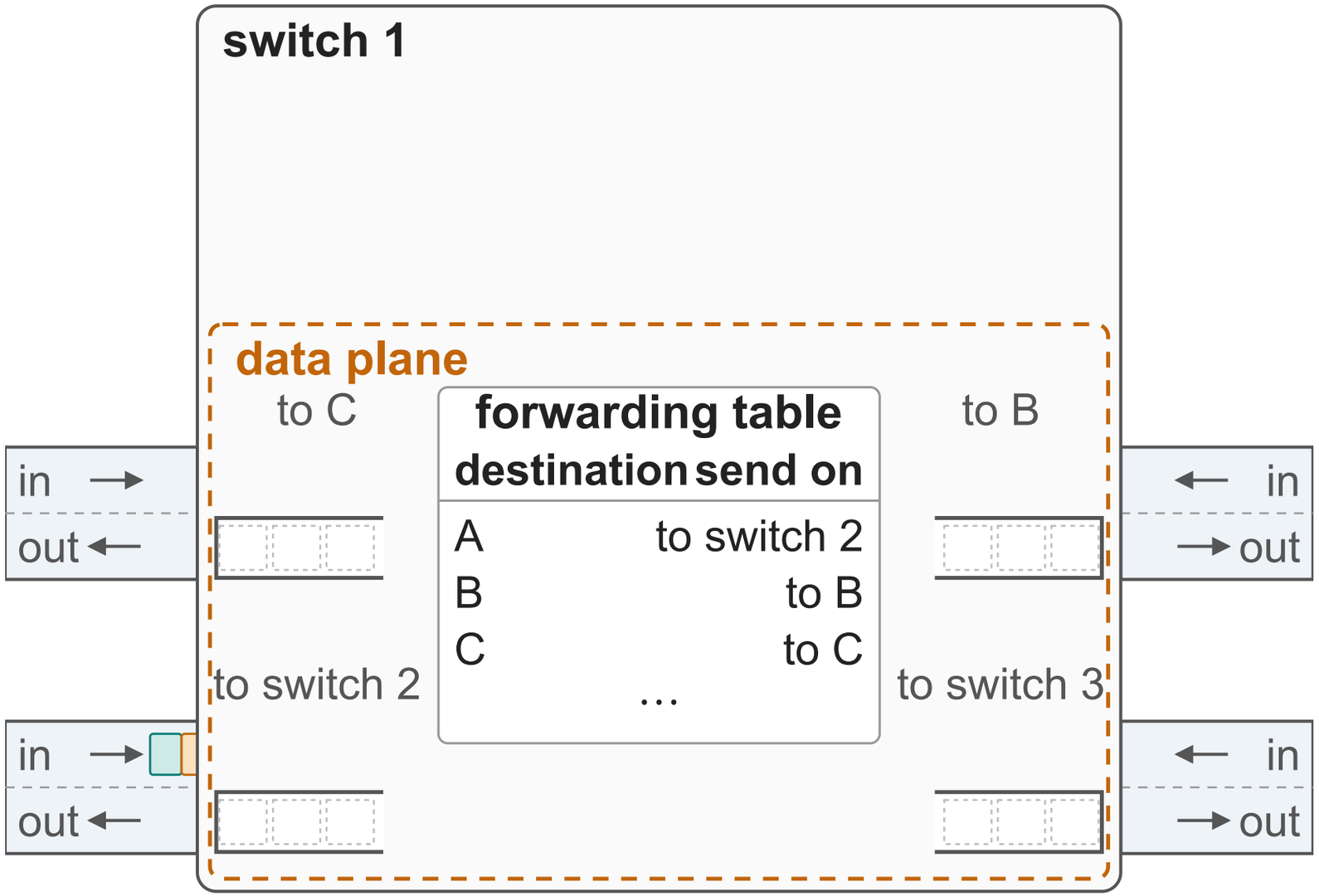


How did the switches know how to forward the packet along the path A → switch 2 → switch 1 → B?

Data plane and control plane



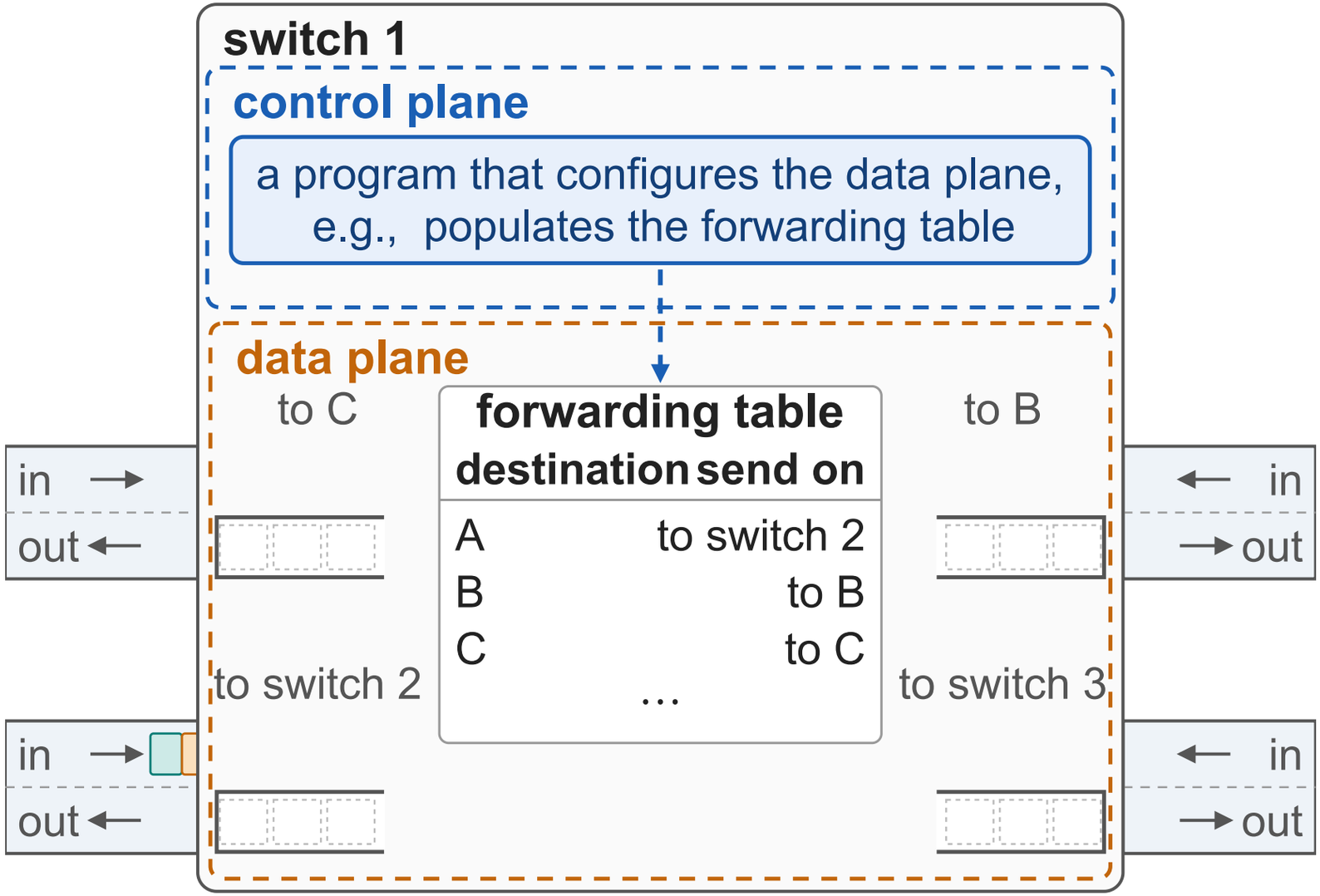
Data plane and control plane



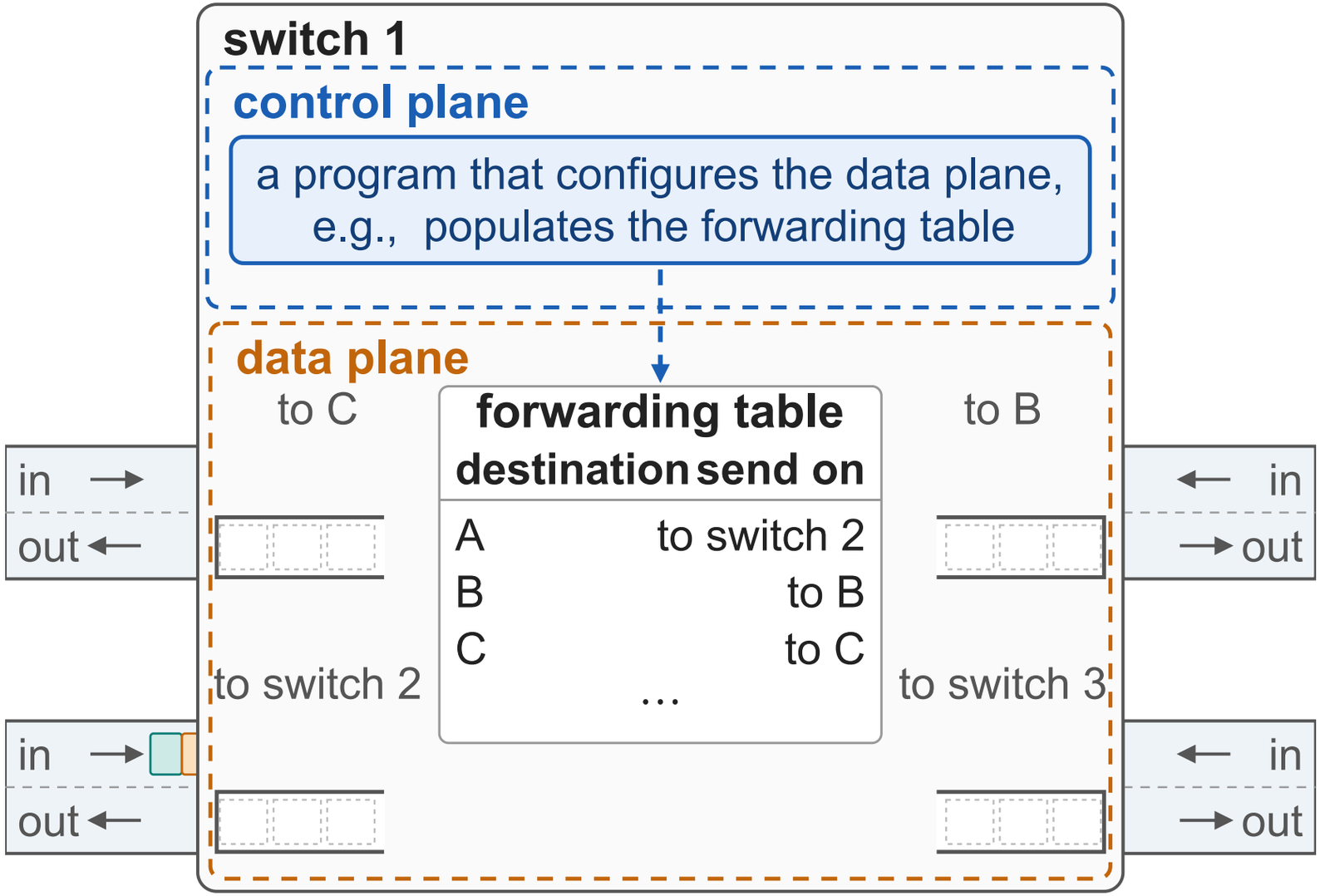
Data plane

- Handles each packet
- One main job: forwarding
  - look up the next hop
  - buffer the packet
  - send it out of the link to next hop
- Has to be fast:
  - On a 100 Gbps link, output a 1000 B packet every 80 ns

Data plane and control plane

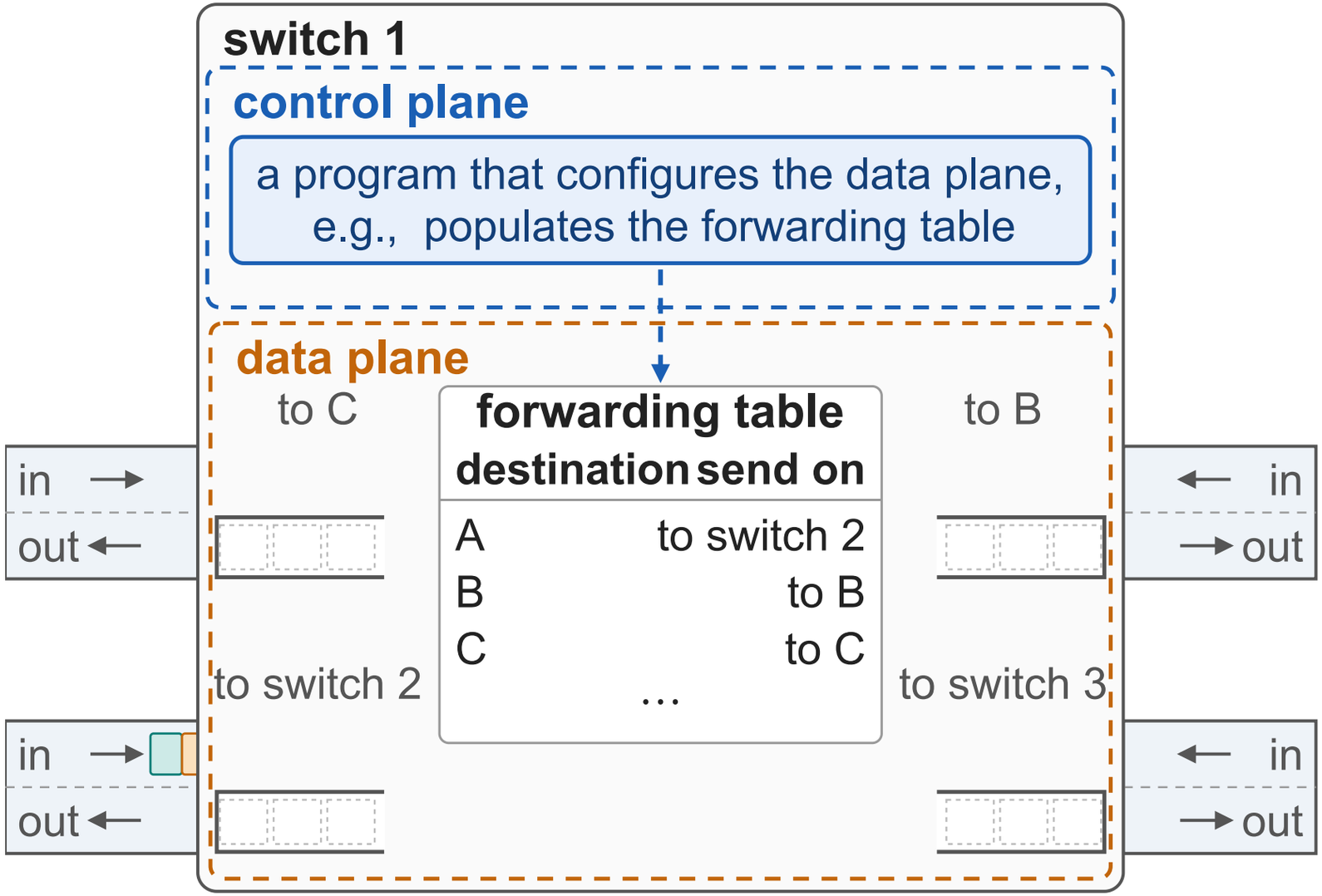


Data plane and control plane

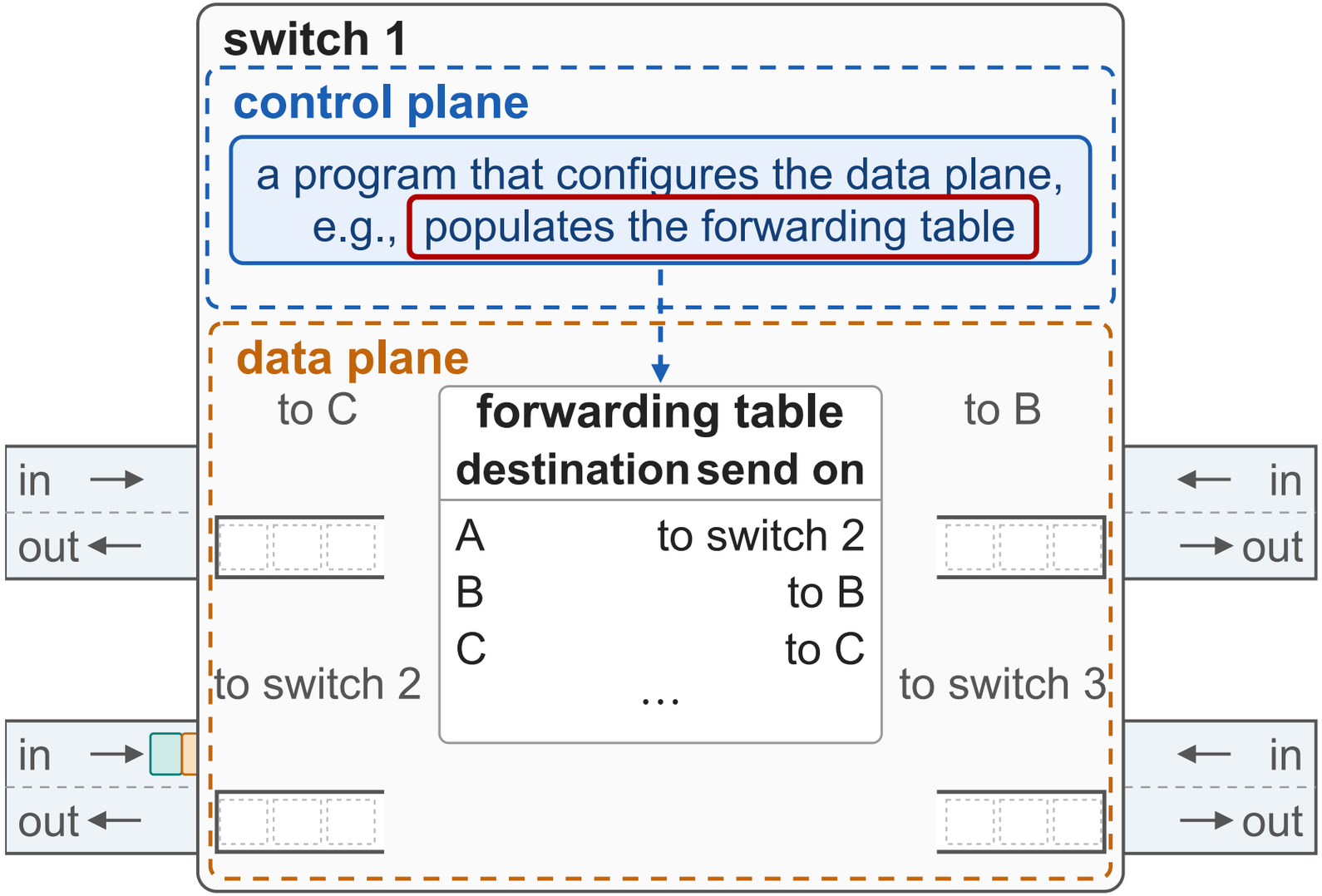


This is a recurring pattern in systems: a fast data path and a slower control path that configures it or handles exceptions.

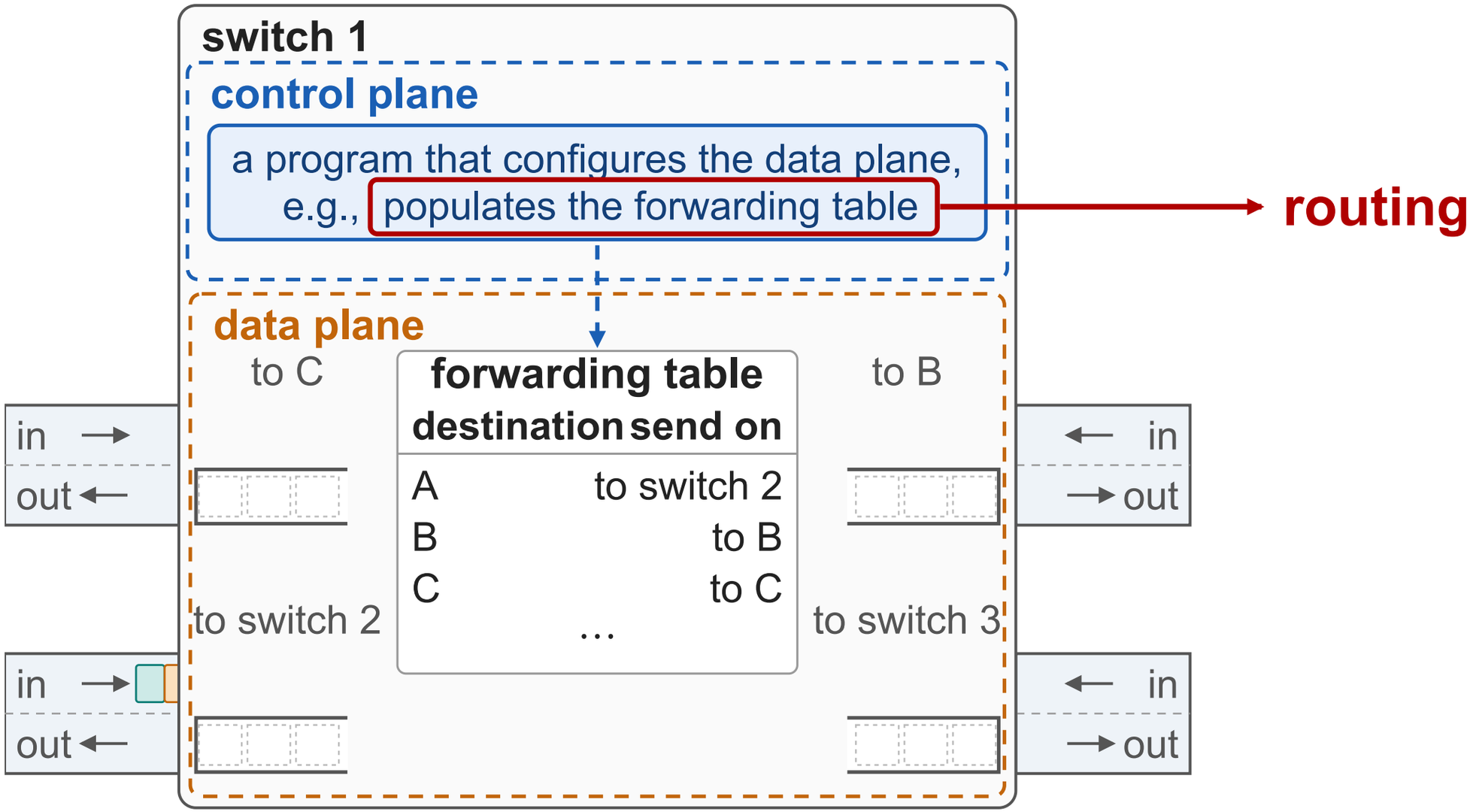
Data plane and control plane



Data plane and control plane



Data plane and control plane



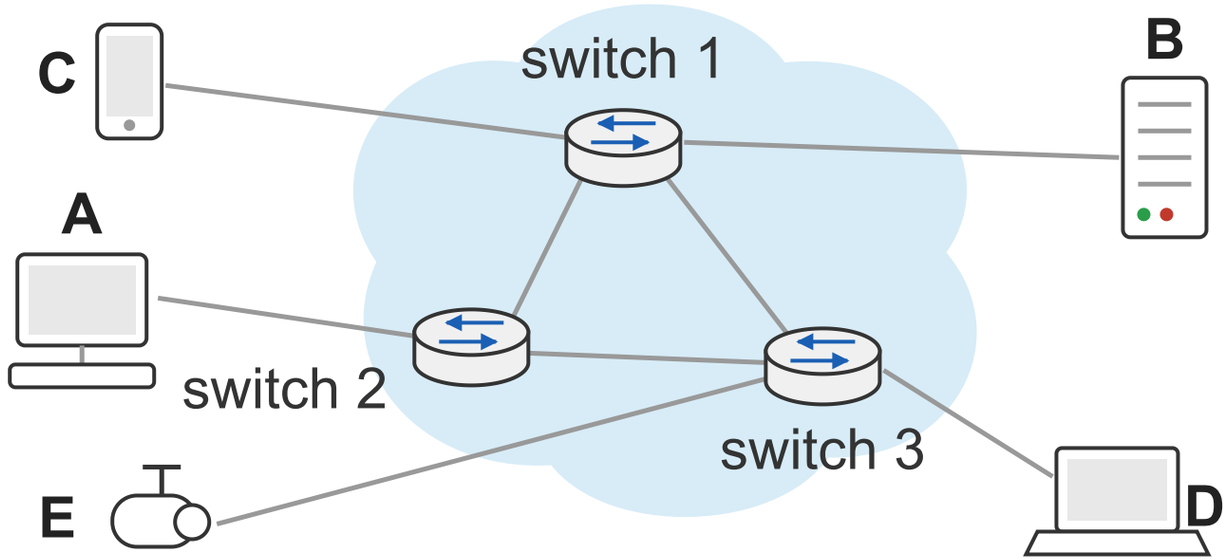
Routing

- One of the main jobs of the control plane is **routing**
- Routing ensures the forwarding table in each switch is populated in a way such that:
  - Every endpoint is *reachable* from every other endpoint
  - over “best” path

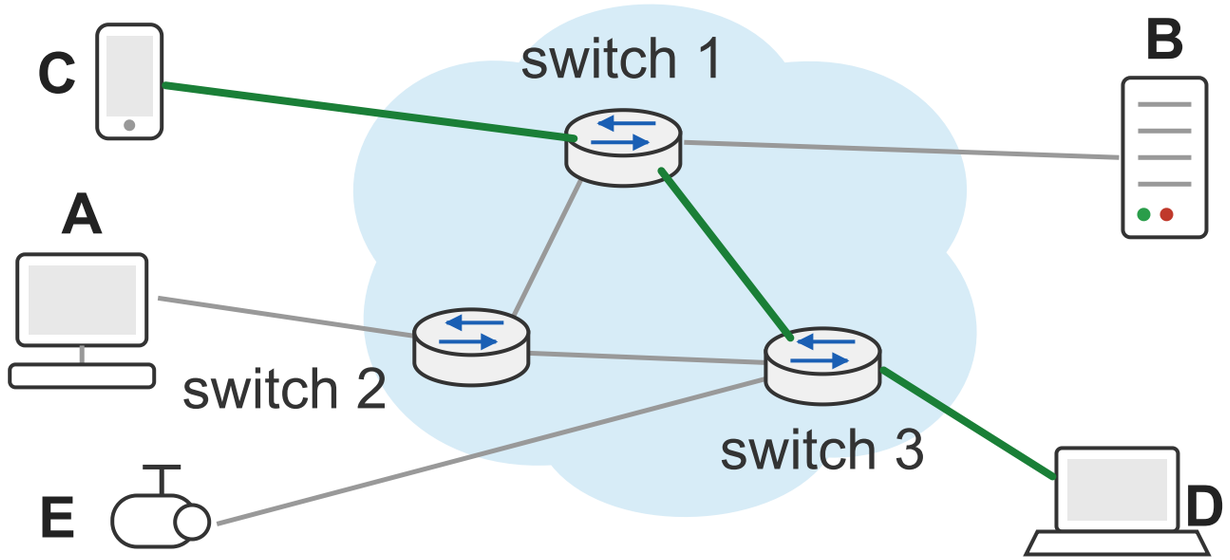
Routing

- One of the main jobs of the control plane is **routing**
- Routing ensures the forwarding table in each switch is populated in a way such that:
  - Every endpoint is *reachable* from every other endpoint
  - over "best" path **?**

Fewest hops?

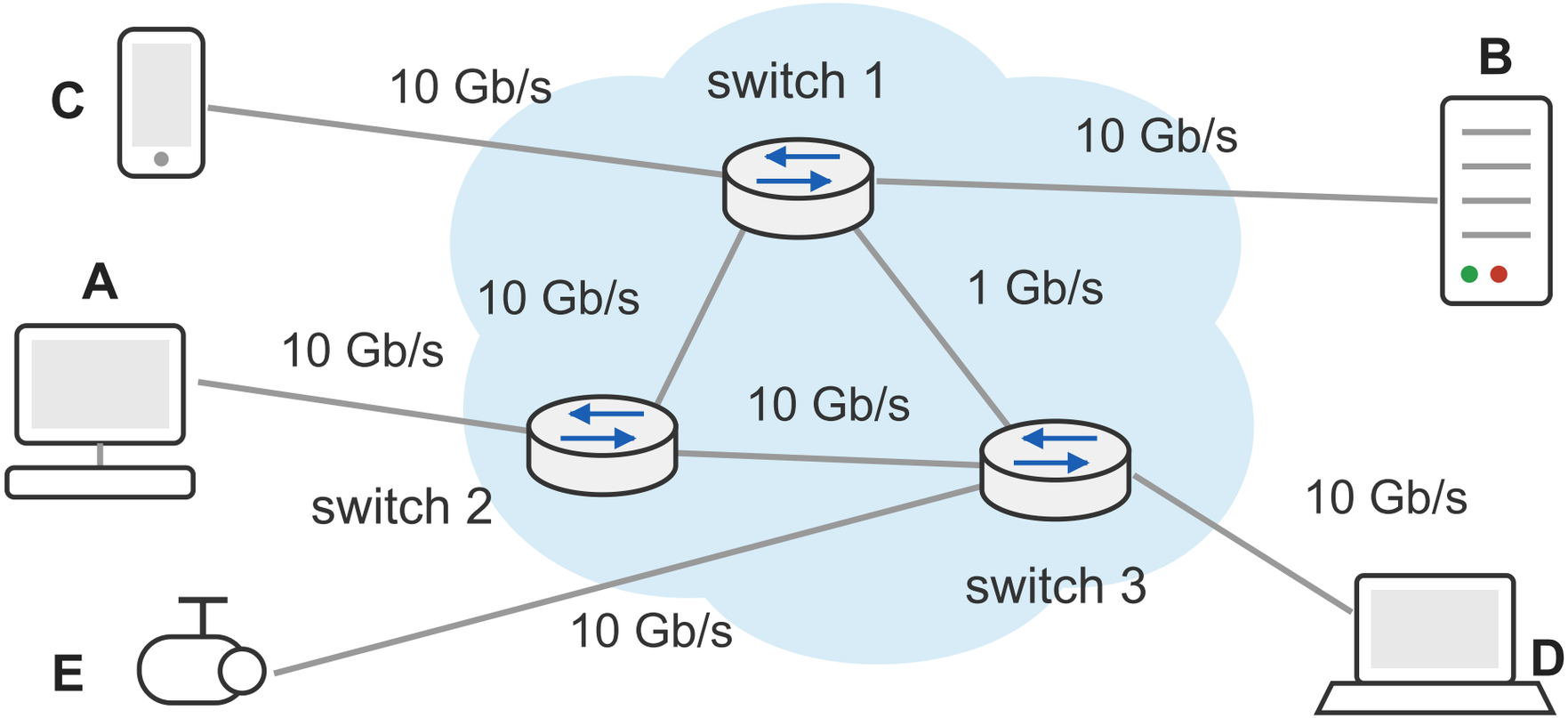


Fewest hops?

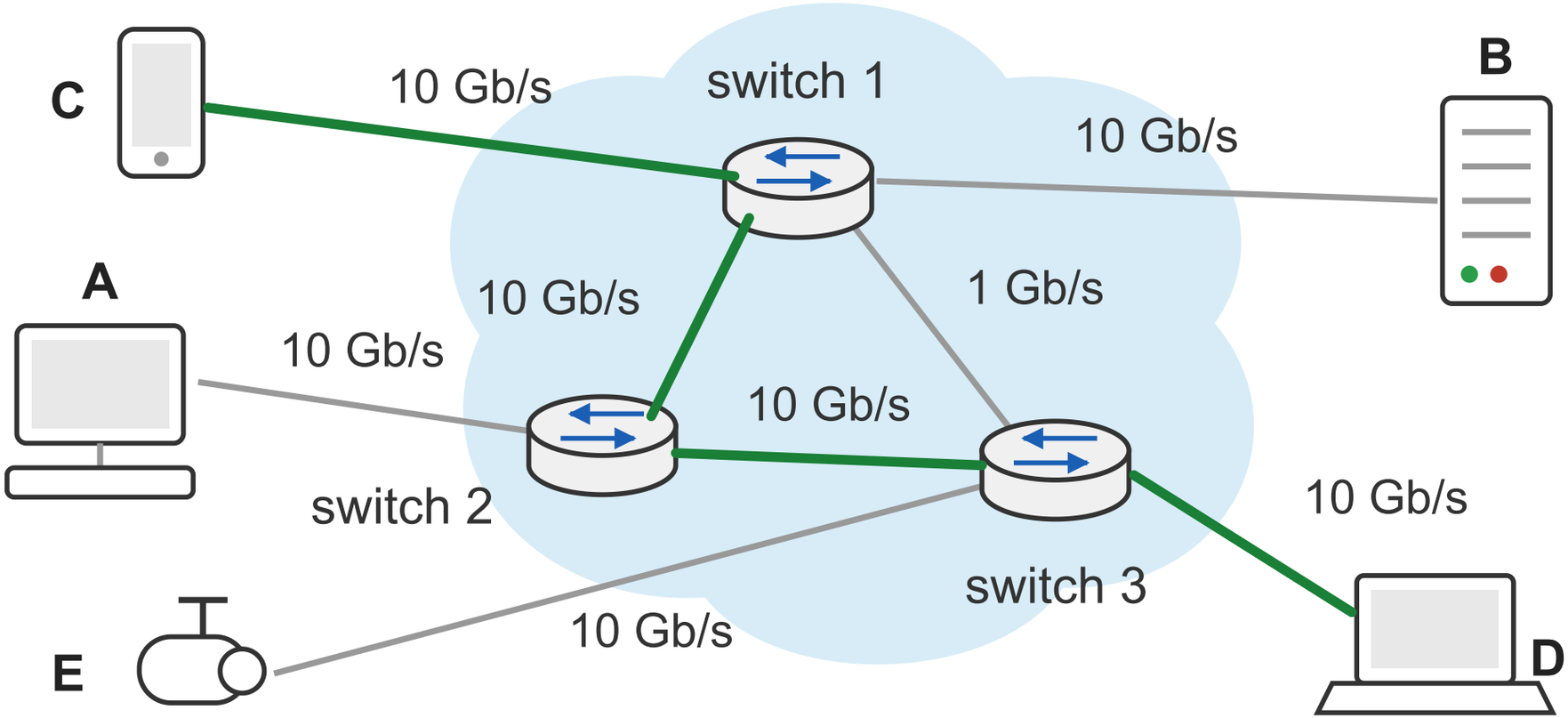


When would you consider this the “best path”?

High throughput?



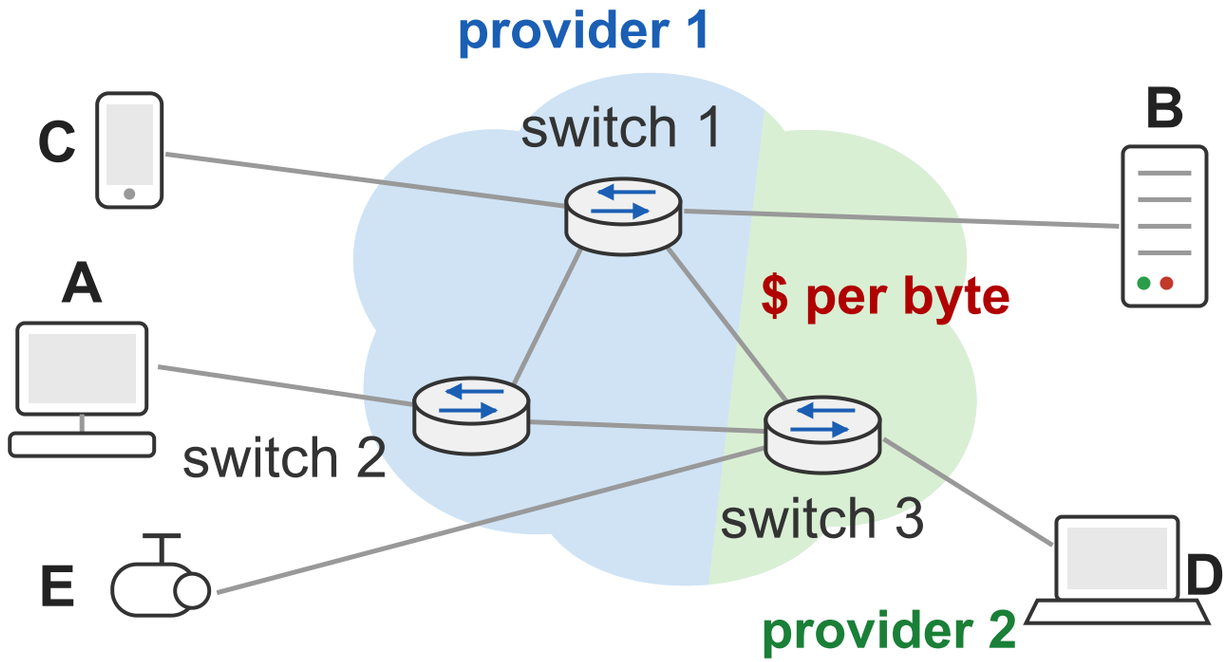
High throughput?



When would you consider this the “best path”?

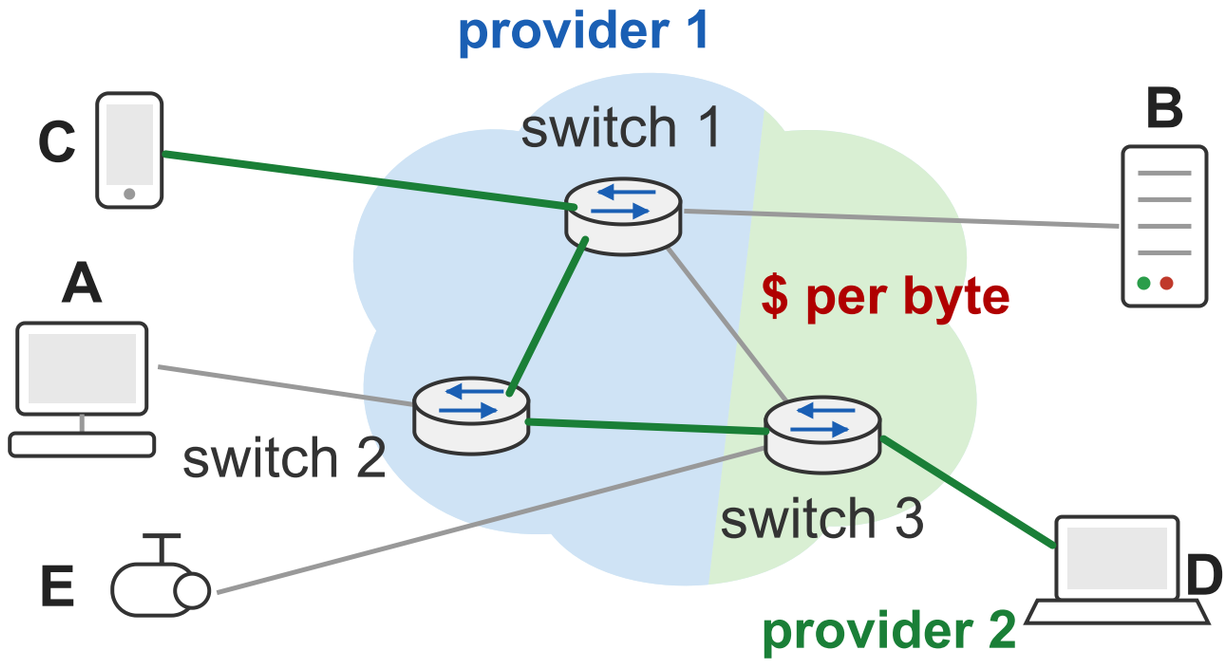
The cheapest?

Two providers. Provider 1 pays provider 2 a flat rate, except on the link from switch 1 to switch 3, billed per byte.



The cheapest?

Two providers. Provider 1 pays provider 2 a flat rate, except on the link from switch 1 to switch 3, billed per byte.



Routing

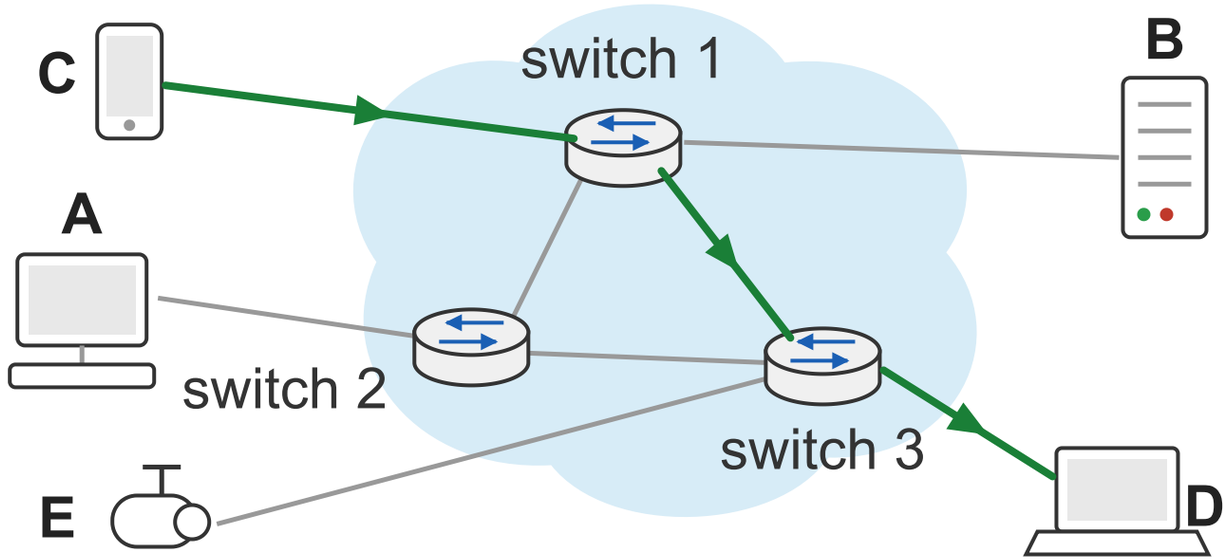
- One of the main jobs of the control plane is **routing**
- Routing ensures the forwarding table in each switch is populated in a way such that:
  - Every endpoint is *reachable* from every other endpoint
  - over "best" path

Routing

- One of the main jobs of the control plane is **routing**
- Routing ensures the forwarding table in each switch is populated in a way such that:
  - Every endpoint is *reachable* from every other endpoint
  - over "best" path

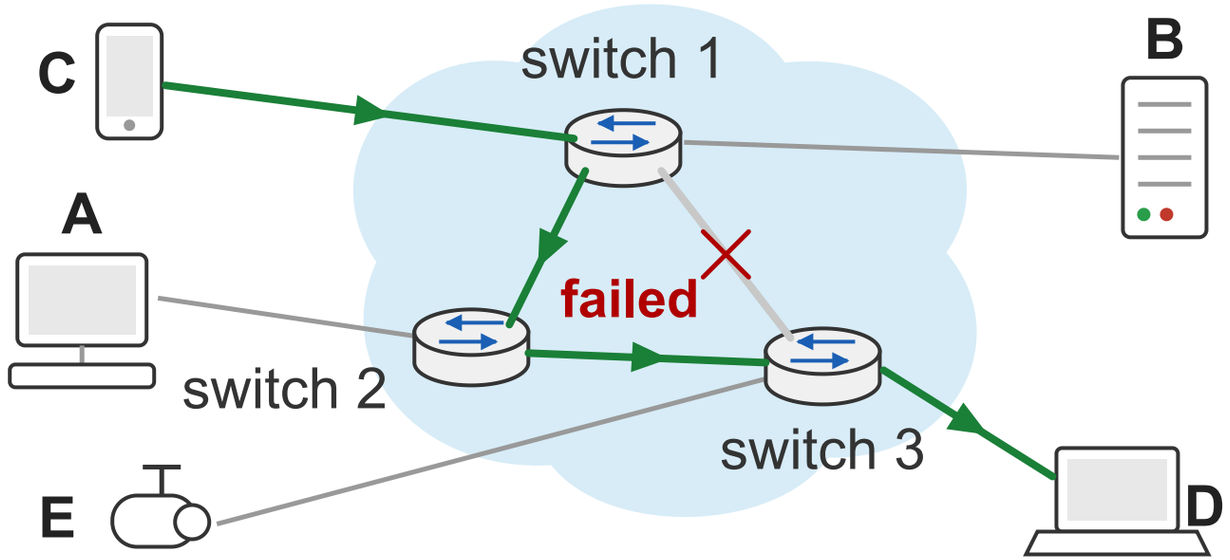
There is no one definition of "best" path.

Routing is not a one-time decision



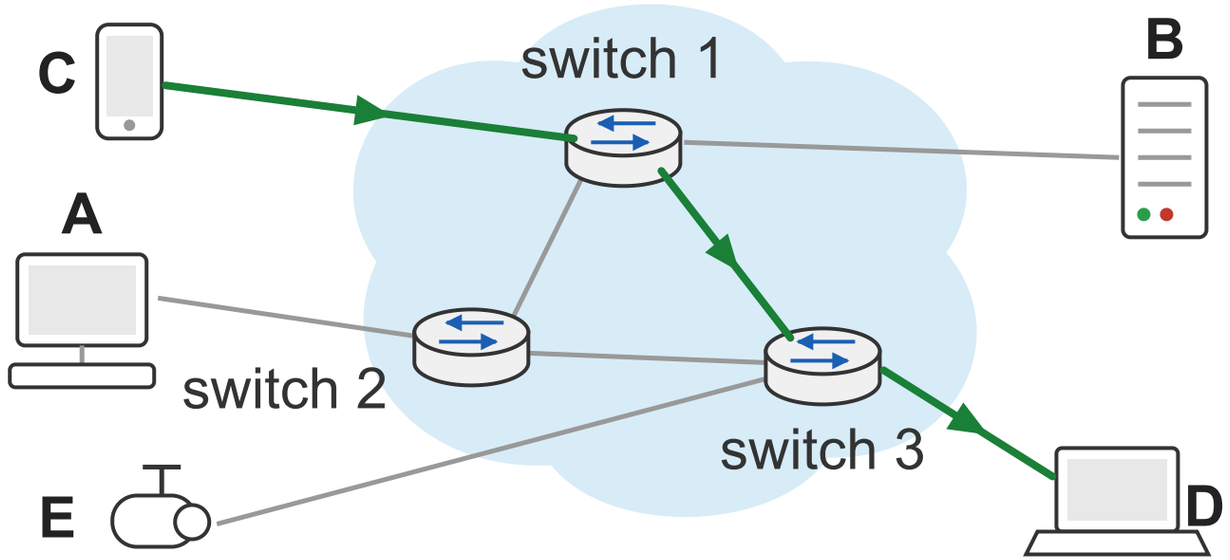
Links can fail

Routing is not a one-time decision



Links can fail

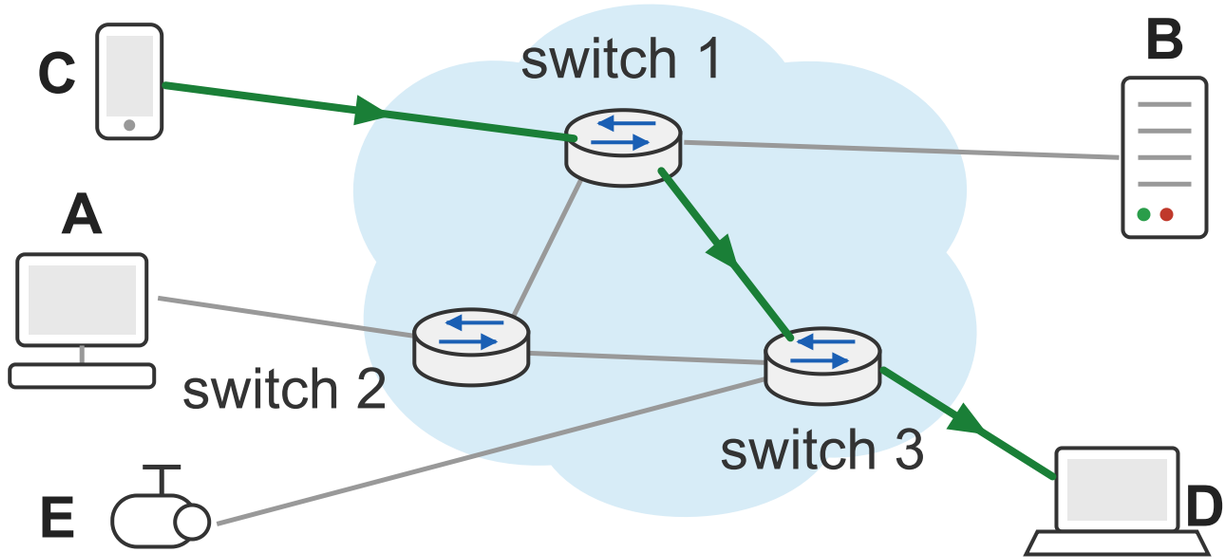
Routing is not a one-time decision



Links can fail

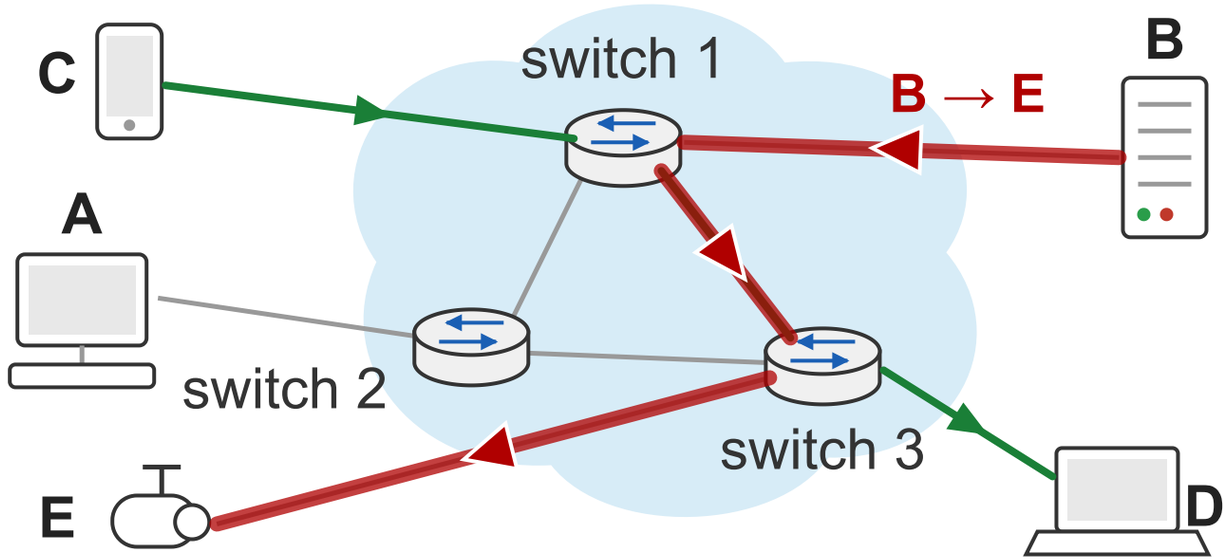
and come back

Routing is not a one-time decision



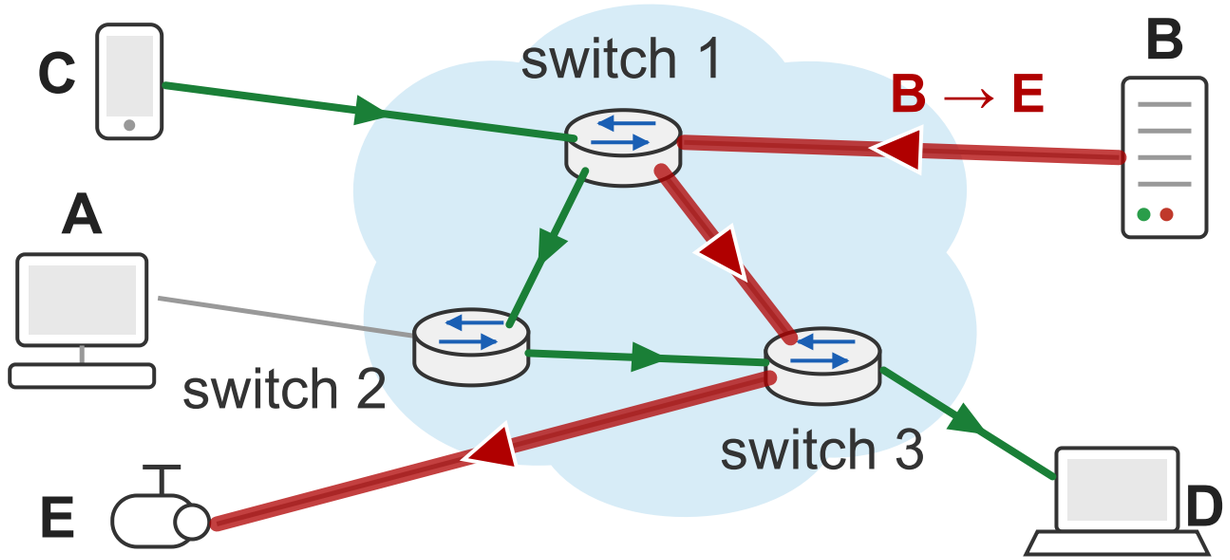
Traffic demand can change

Routing is not a one-time decision



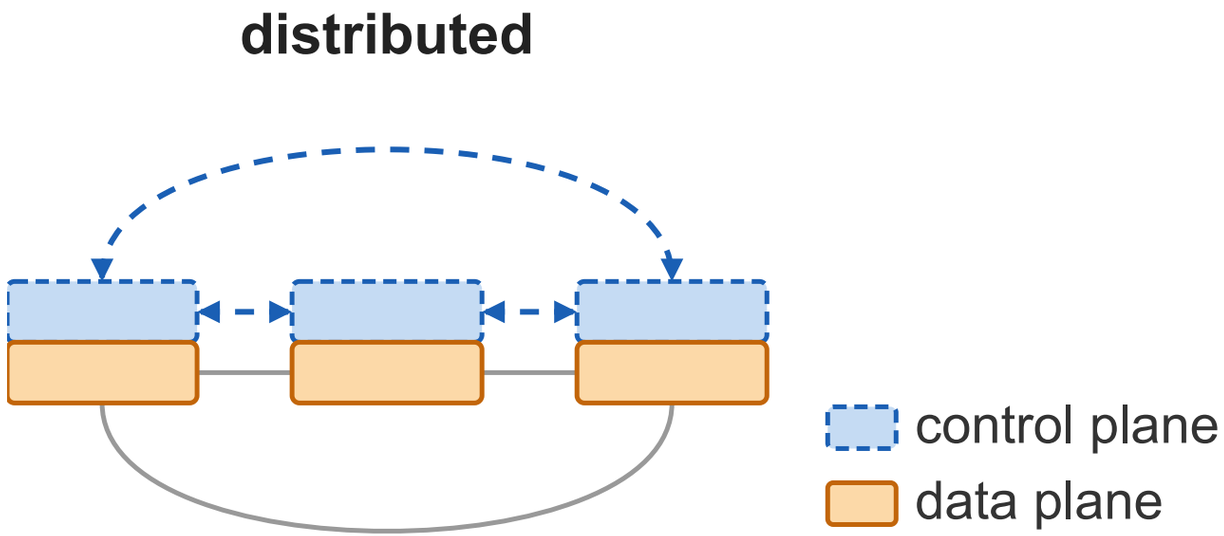
Traffic demand can change

Routing is not a one-time decision



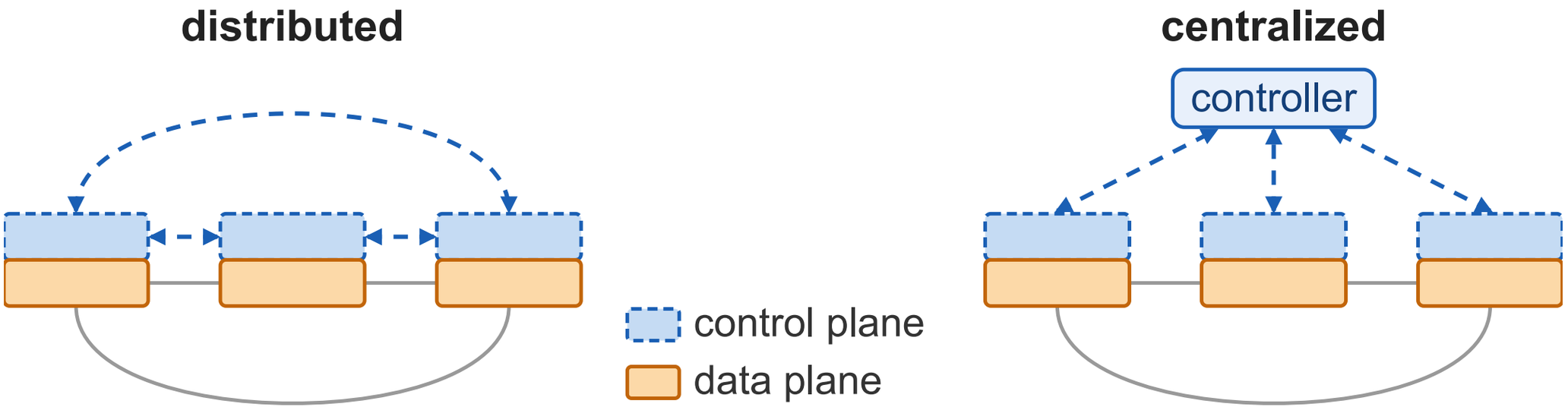
Traffic demand can change

Who decides the paths?



Switches communicate with each other to find the best paths in a distributed manner.

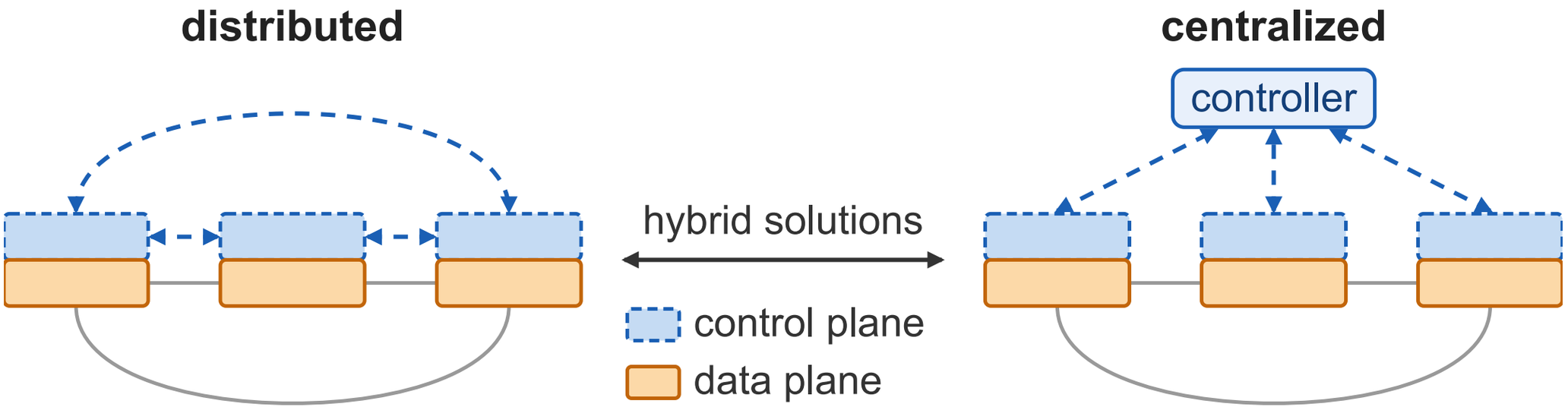
Who decides the paths?



Switches communicate with each other to find the best paths in a distributed manner.

Switches provide information to a centralized controller, who decides the paths globally, and communicates it back to the switches.

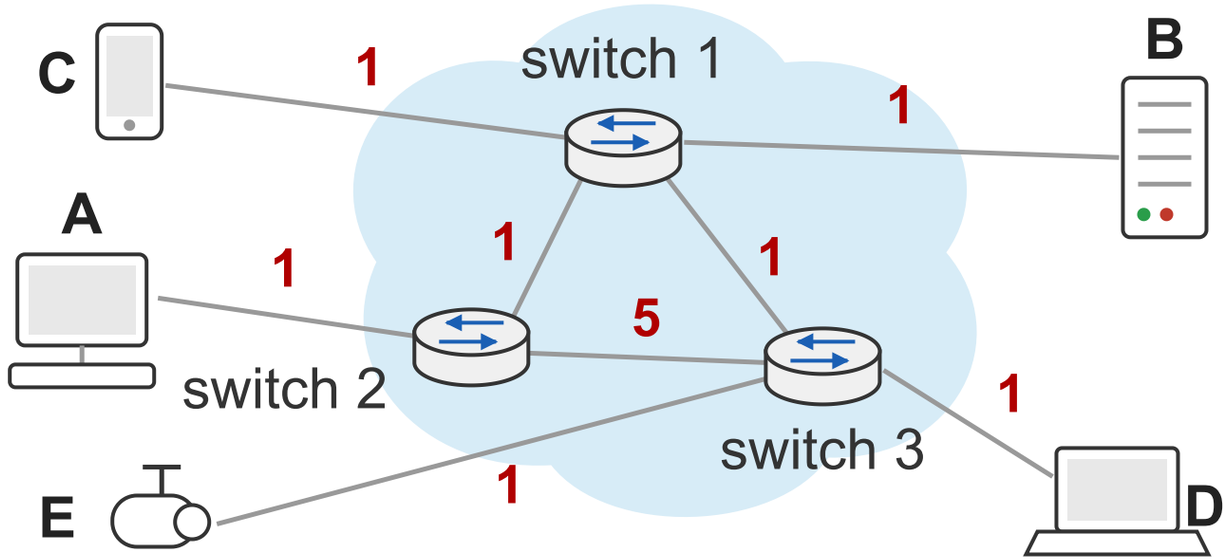
Who decides the paths?



Switches communicate with each other to find the best paths in a distributed manner.

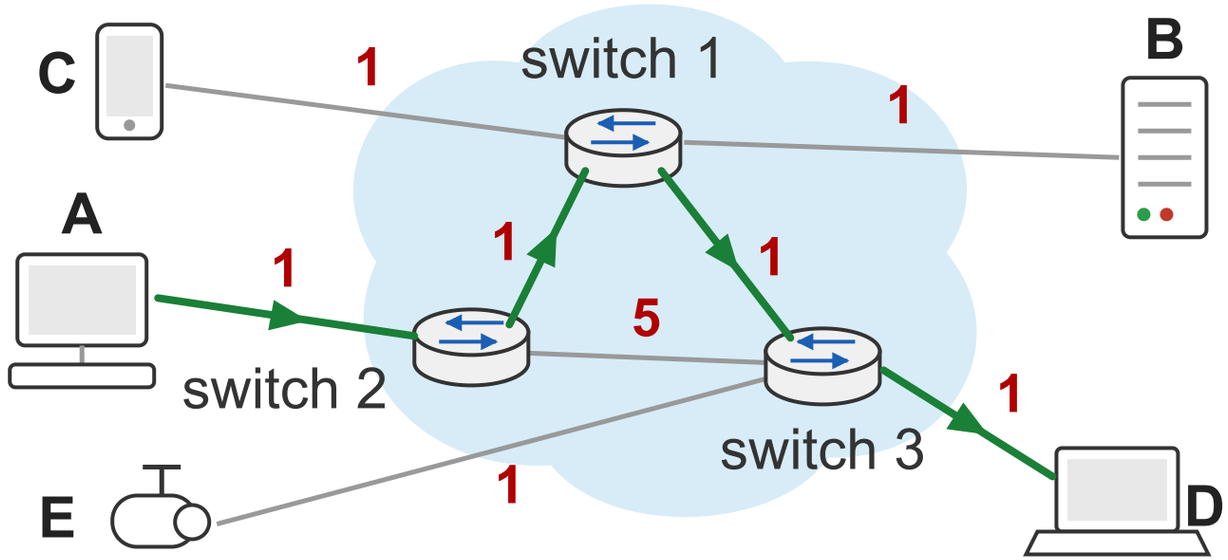
Switches provide information to a centralized controller, who decides the paths globally, and communicates it back to the switches.

A common routing strategy: least-cost paths



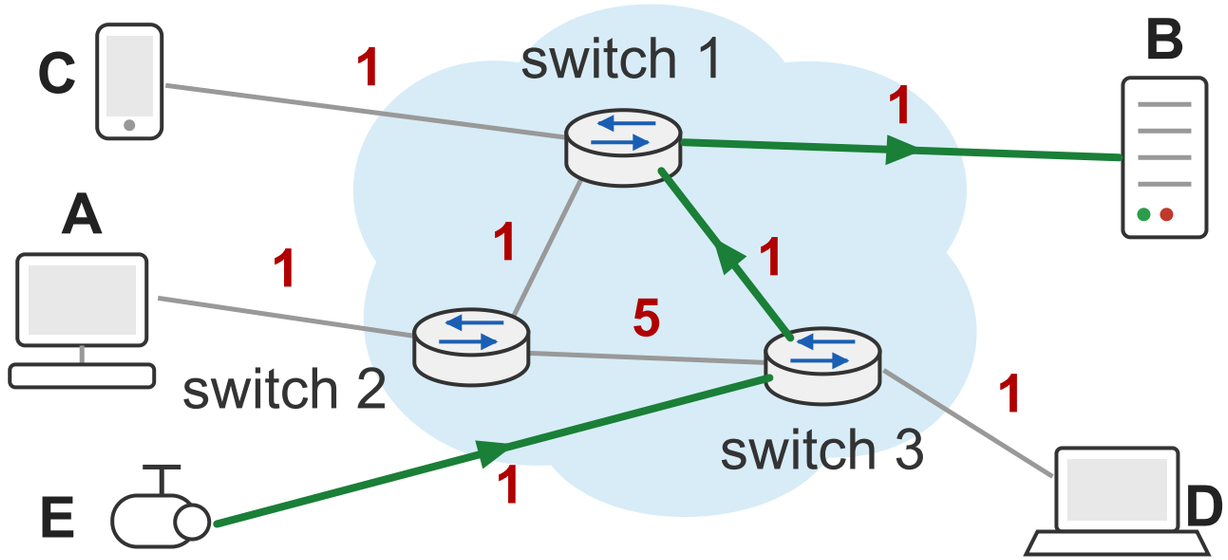
- Every link is given a cost (e.g., by the operator and/or measured by the switches)
- Best path: the one with the least total cost (sum of its links' costs)
- The routing algorithm (distributed or centralized) finds the least-cost paths and populates each forwarding table (destination → next hop)

A common routing strategy: least-cost paths



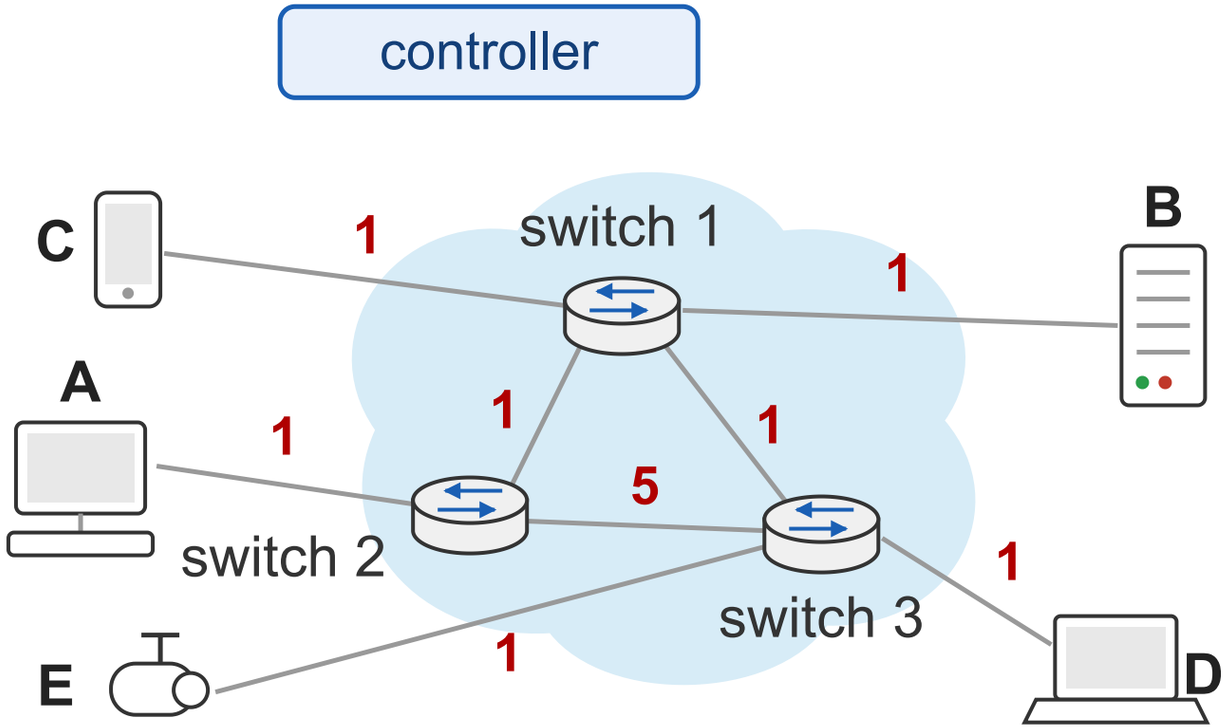
- Every link is given a cost (e.g., by the operator and/or measured by the switches)
- Best path: the one with the least total cost (sum of its links' costs)
- The routing algorithm (distributed or centralized) finds the least-cost paths and populates each forwarding table (destination → next hop)

A common routing strategy: least-cost paths

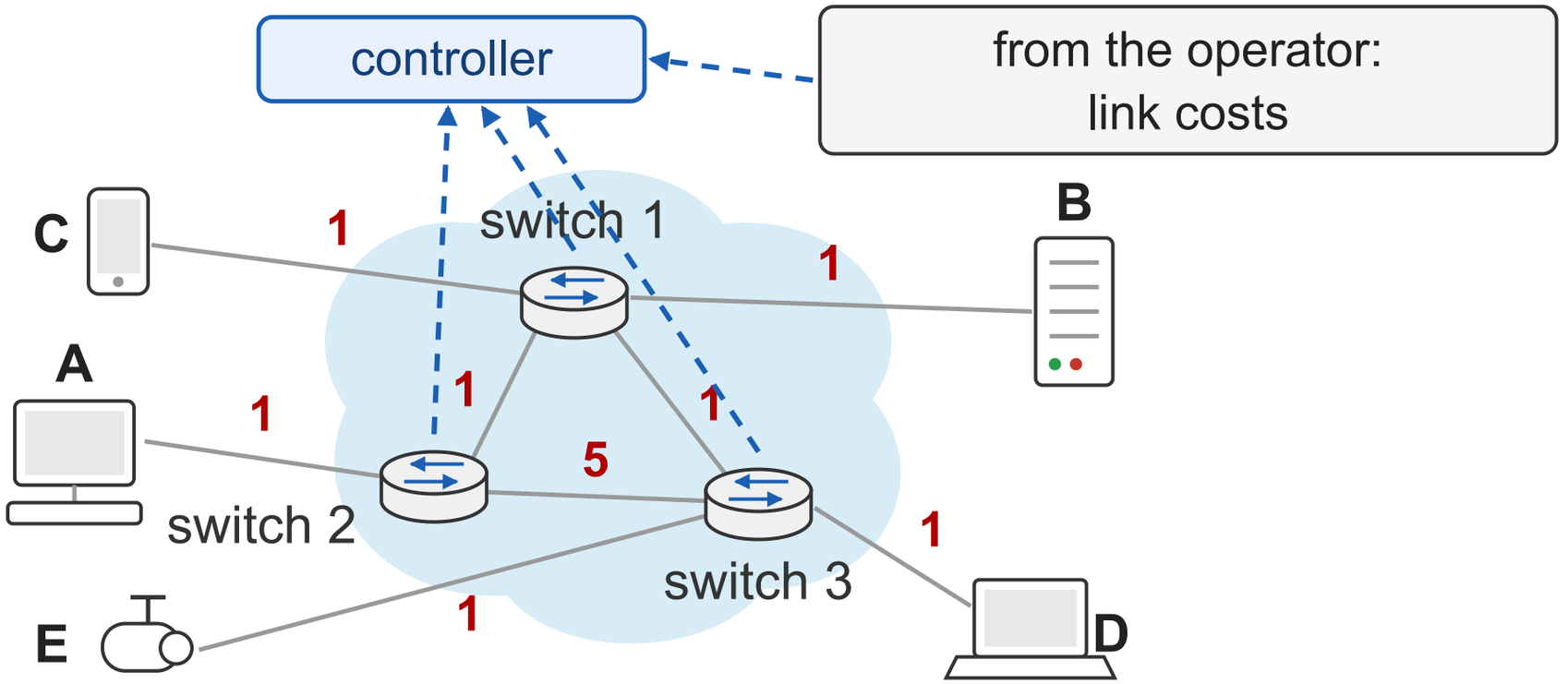


- Every link is given a cost (e.g., by the operator and/or measured by the switches)
- Best path: the one with the least total cost (sum of its links' costs)
- The routing algorithm (distributed or centralized) finds the least-cost paths and populates each forwarding table (destination → next hop)

How would this be implemented in a centralized control-plane?

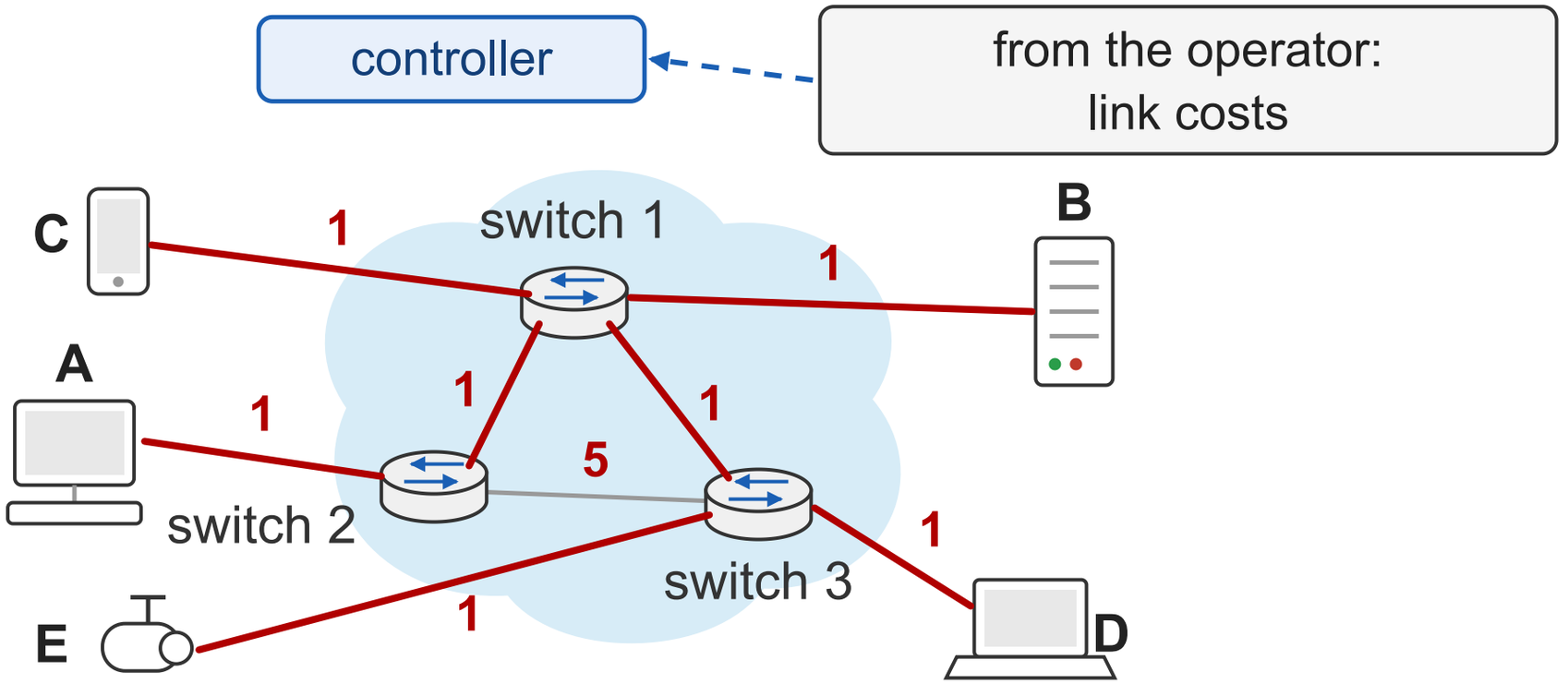


How would this be implemented in a centralized control-plane?



The switches report link status (e.g., which links are up, measurements that can determine a link's cost)

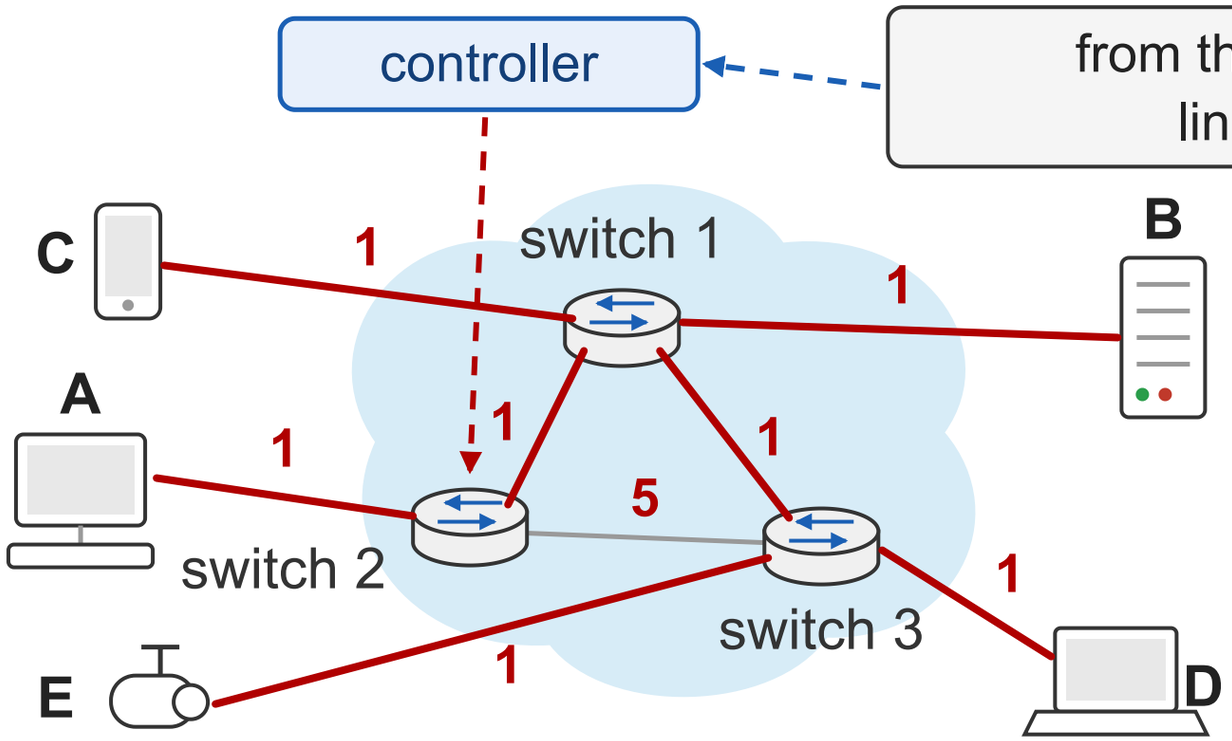
How would this be implemented in a centralized control-plane?



The controller:

- finds the least-cost paths to each destination (e.g., using graph algorithms)

How would this be implemented in a centralized control-plane?



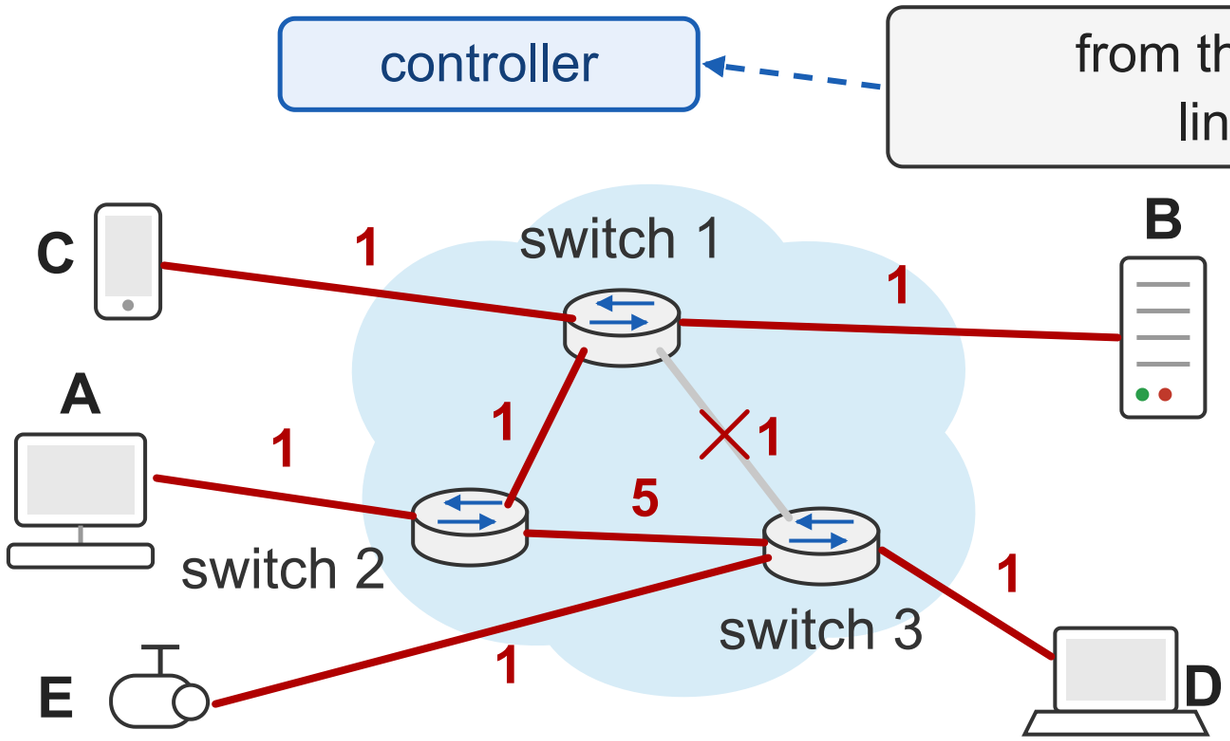
switch 2's table

destination	next hop
A	A
B	switch 1
C	switch 1
D	switch 1
E	switch 1

The controller:

- computes the forwarding table rules
- communicates the rules to the switches

How would this be implemented in a centralized control-plane?



switch 2's table

destination	next hop
A	A
B	switch 1
C	switch 1
D	**switch 3**
E	**switch 3**

When the controller learns about a failure,
it modifies forwarding rules accordingly.

The costs decide what “best” means

By picking what a link’s cost stands for, the operator can implement different policies:

The costs decide what “best” means

By picking what a link’s cost stands for, the operator can implement different policies:

- Fewest hops: every link costs 1

The costs decide what “best” means

By picking what a link’s cost stands for, the operator can implement different policies:

- Fewest hops: every link costs 1
- Cheapest: cost = what the operator pays to send over the link

The costs decide what “best” means

By picking what a link’s cost stands for, the operator can implement different policies:

- Fewest hops: every link costs 1
- Cheapest: cost = what the operator pays to send over the link
- Maintenance: raise a link’s cost so traffic moves off it before it is taken down

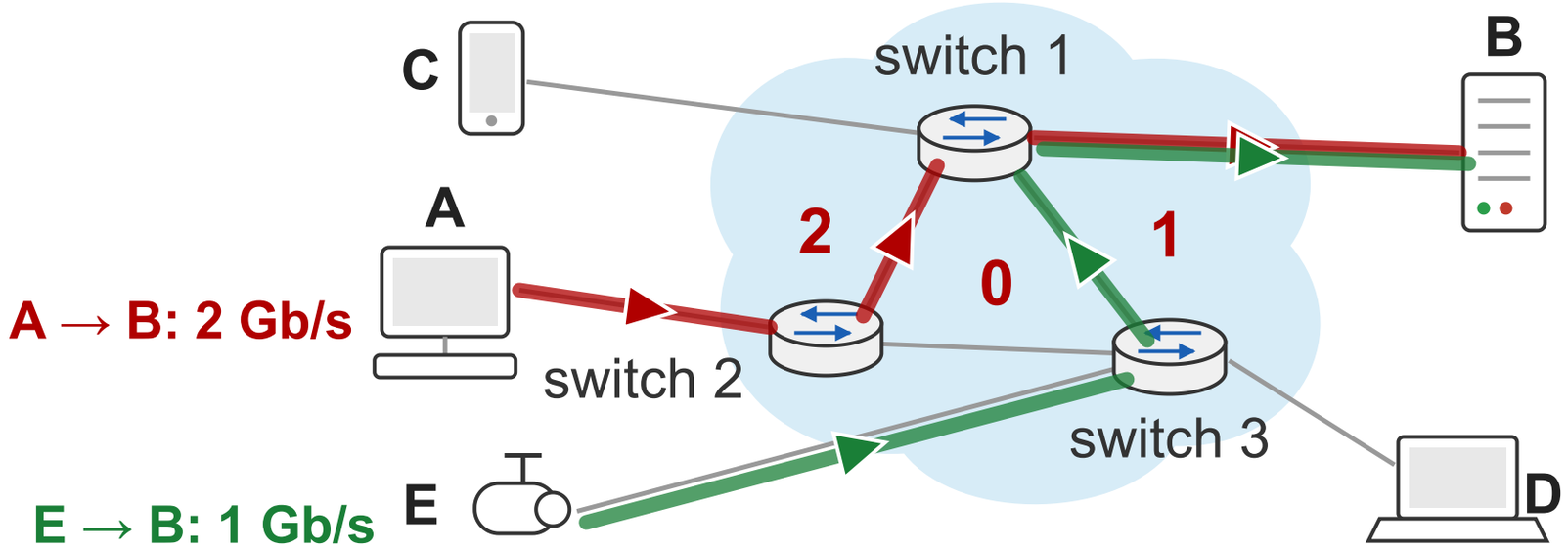
The costs decide what “best” means

By picking what a link’s cost stands for, the operator can implement different policies:

- Fewest hops: every link costs 1
- Cheapest: cost = what the operator pays to send over the link
- Maintenance: raise a link’s cost so traffic moves off it before it is taken down
- …

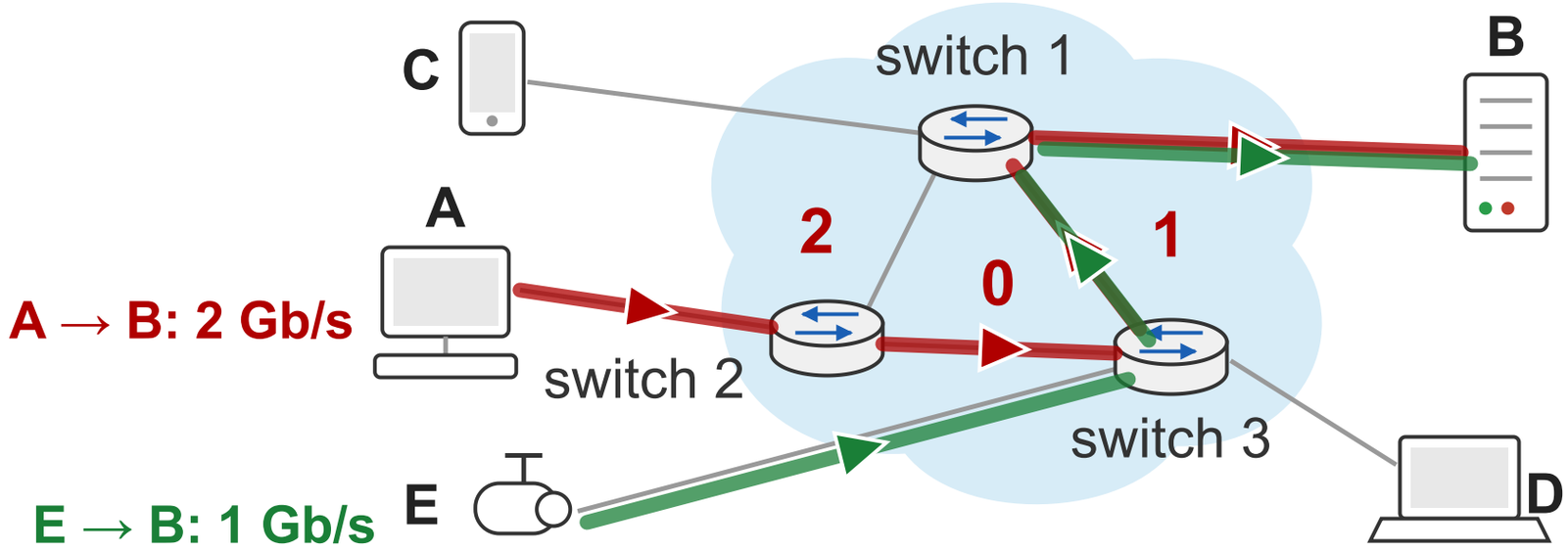
Careful with the costs

What if the cost of a link is its traffic load (i.e., the traffic crossing that link)?



Careful with the costs

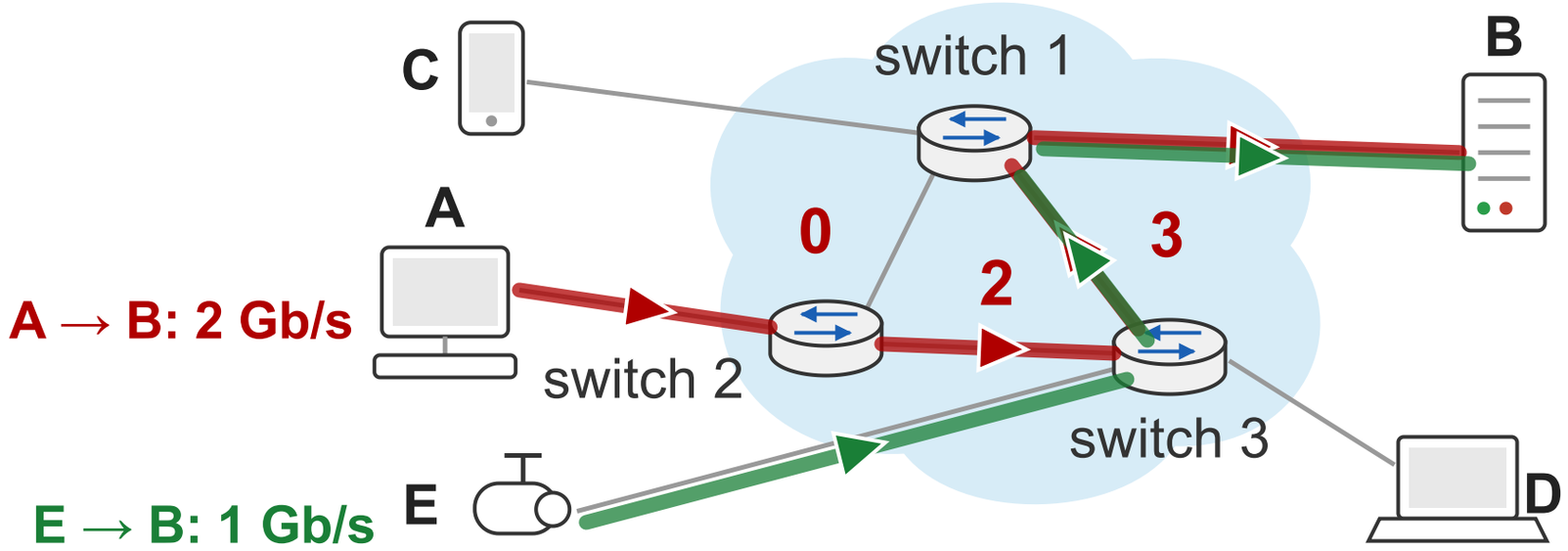
What if the cost of a link is its traffic load (i.e., the traffic crossing that link)?



A → B traffic moves from the direct path
to the path through switch 3

Careful with the costs

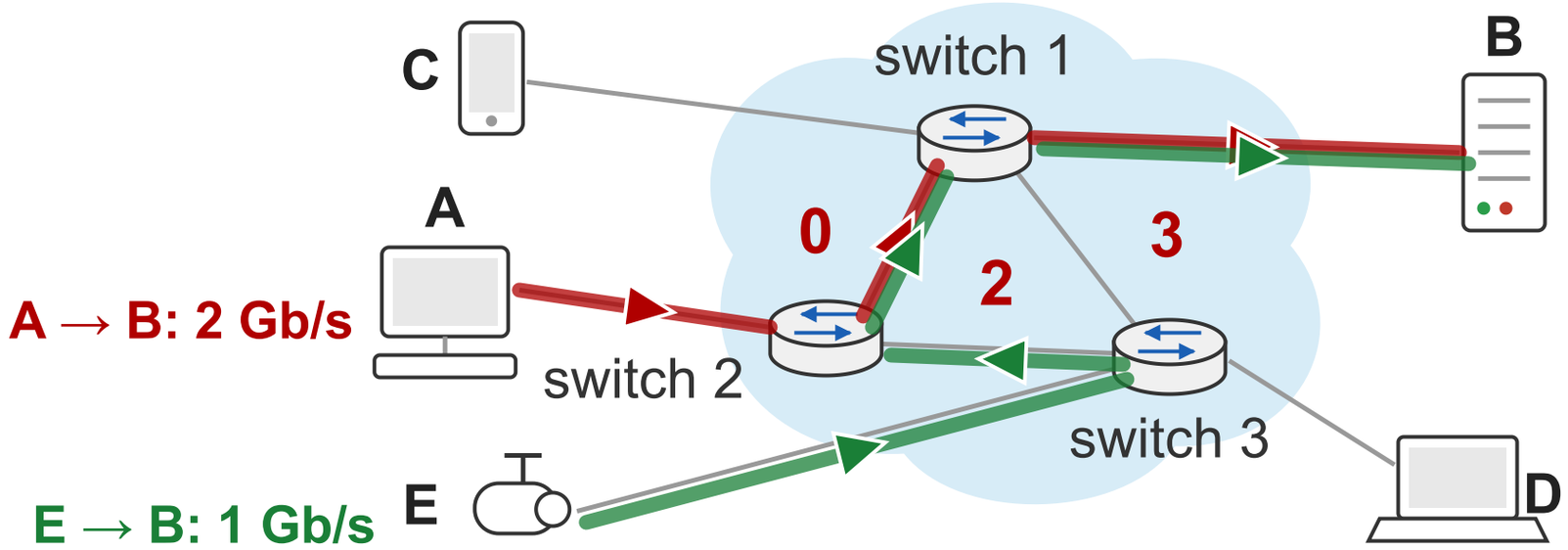
What if the cost of a link is its traffic load (i.e., the traffic crossing that link)?



The link costs are measured again

Careful with the costs

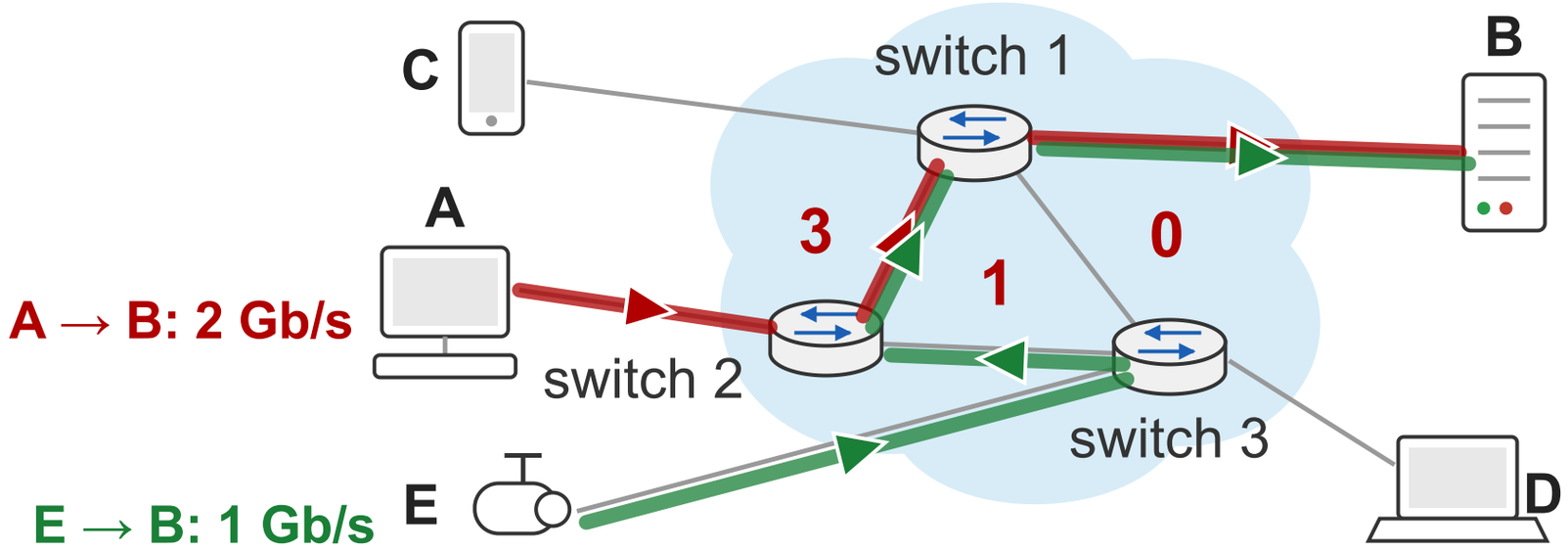
What if the cost of a link is its traffic load (i.e., the traffic crossing that link)?



A → B traffic moves back to the direct path;
E → B traffic moves to the path through switch 2

Careful with the costs

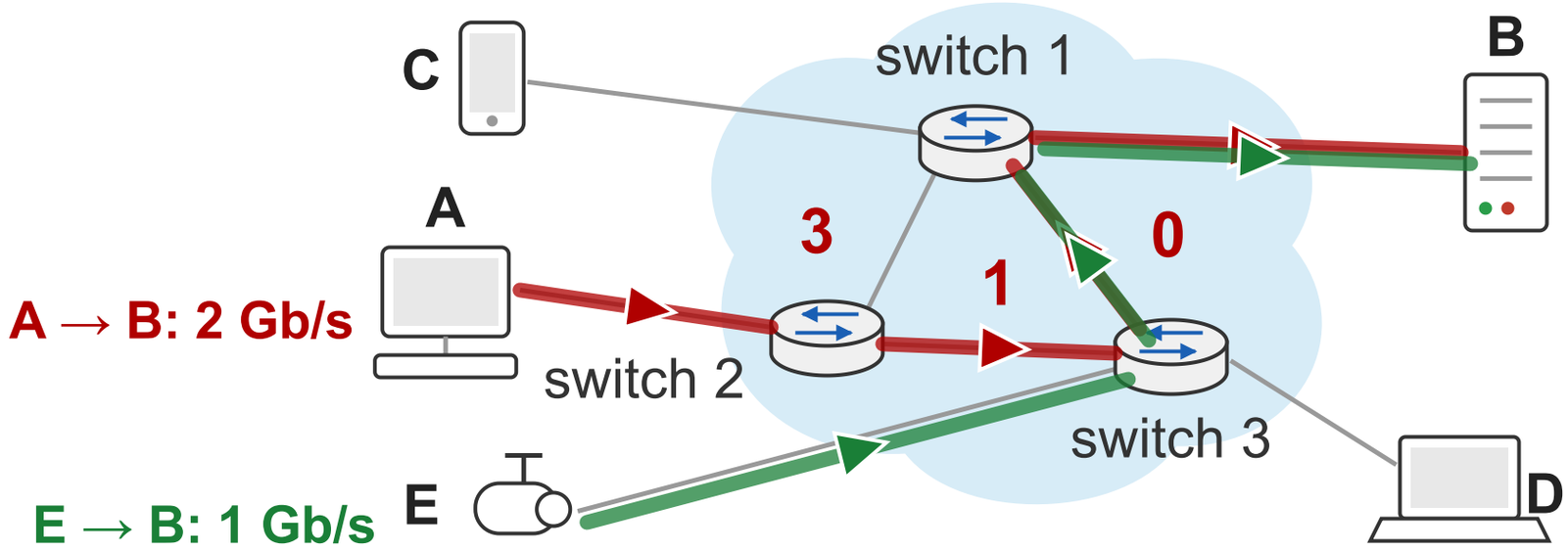
What if the cost of a link is its traffic load (i.e., the traffic crossing that link)?



The link costs are measured again

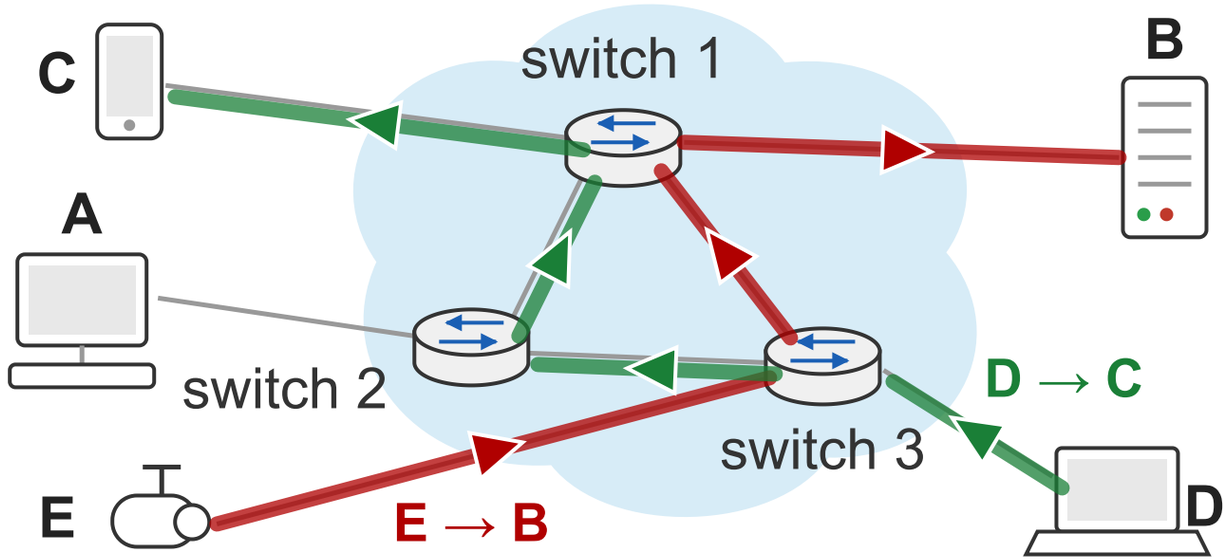
Careful with the costs

What if the cost of a link is its traffic load (i.e., the traffic crossing that link)?



A → B traffic moves through switch 3 again, E → B back to the direct path: the paths keep flipping

Least-cost paths are not always enough

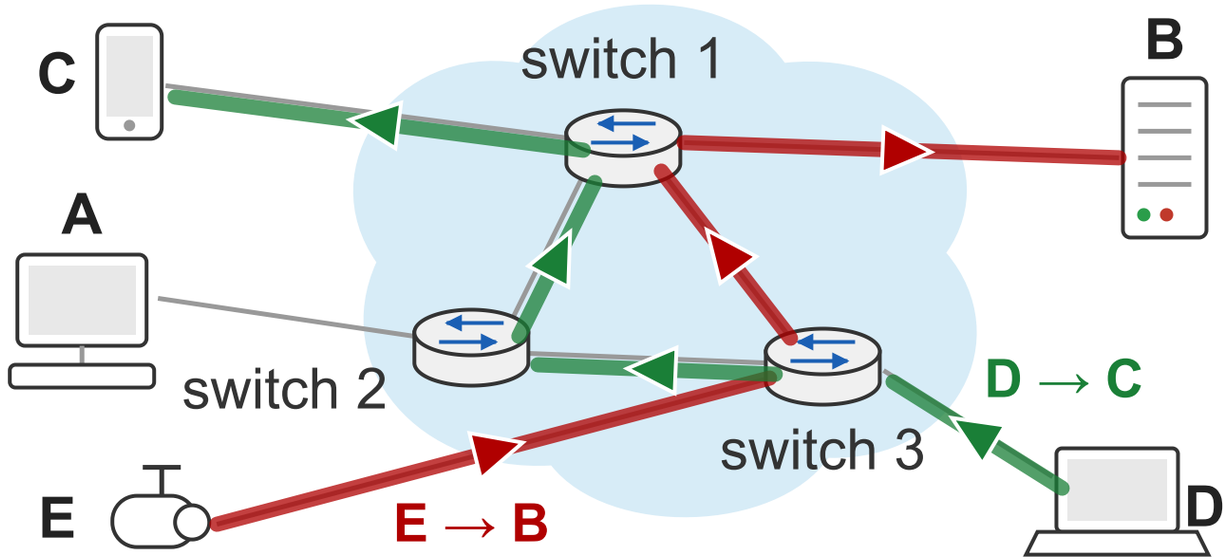


- Policy: keep the big flows apart
  - Two long-running high-throughput traffic flows: E → B and D → C
  - We don’t want their paths to overlap

Keep the big flows apart with costs? (in-class)

(in-class discussion)

Pick costs for the links so that least-cost paths keep the two flows apart.



Reset

How would this be implemented in a centralized control-plane?

