CS168: Discussion 4

...

Intro to the Internet

Fall 2022

Announcement:

Starting Tuesday (Sep 20), in person lectures are back !!!

Agenda: Routing!

- Distance-Vector Routing
  - Lecture recap
  - Small modification
  - Slides w/ worksheet
- Link-State Routing

Distance-Vector Routing

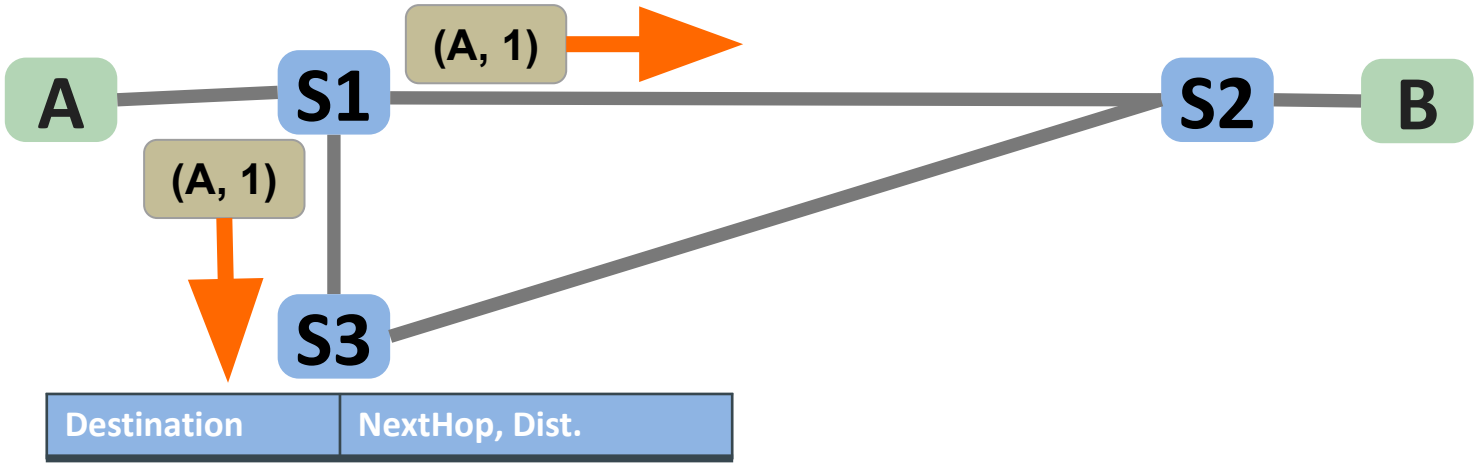
- Each router:
  - Has a map of **destinations** to **next hop/distance**
  - Connected hosts are **statically programmed** into the table

Destination	NextHop, Dist.
X	Direct, 1
Y	S1, 8
Z	S2, 10

D-V Review

Destination	NextHop, Dist.
A	Direct, 1

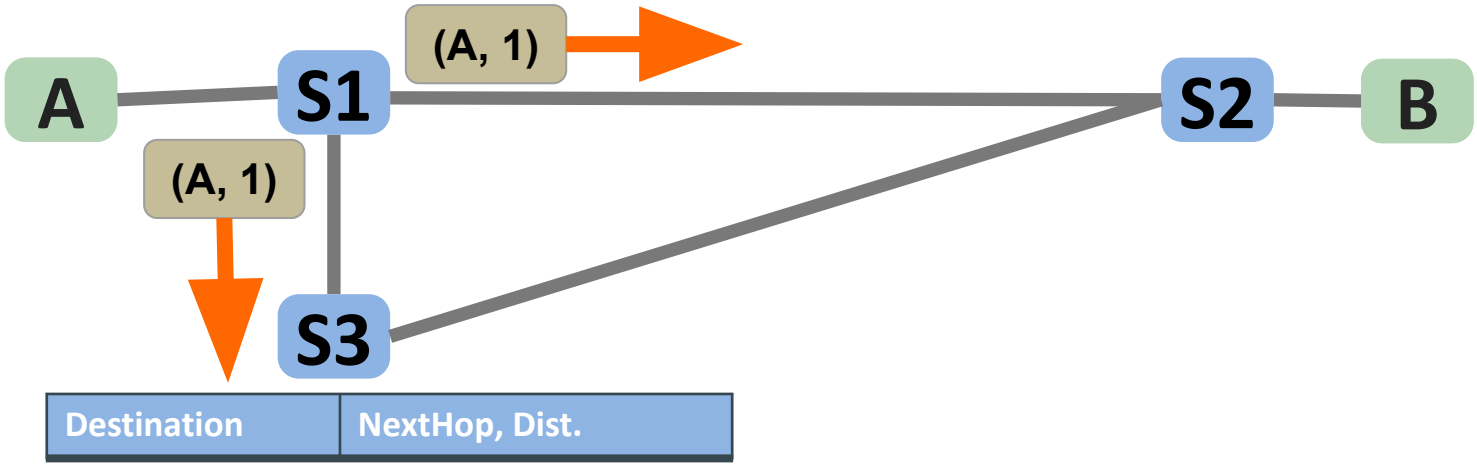
Destination	NextHop, Dist.
B	Direct, 1



D-V Review

Destination	NextHop, Dist.
A	Direct, 1

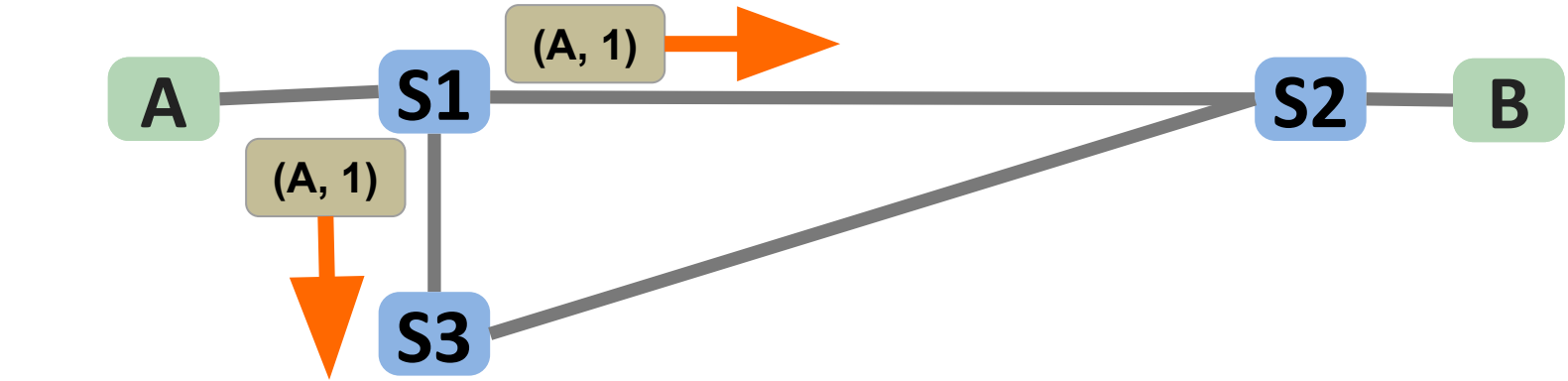
Destination	NextHop, Dist.
B	Direct, 1
A	S1, 2



D-V Review

Destination	NextHop, Dist.
A	Direct, 1

Destination	NextHop, Dist.
B	Direct, 1
A	S1, 2

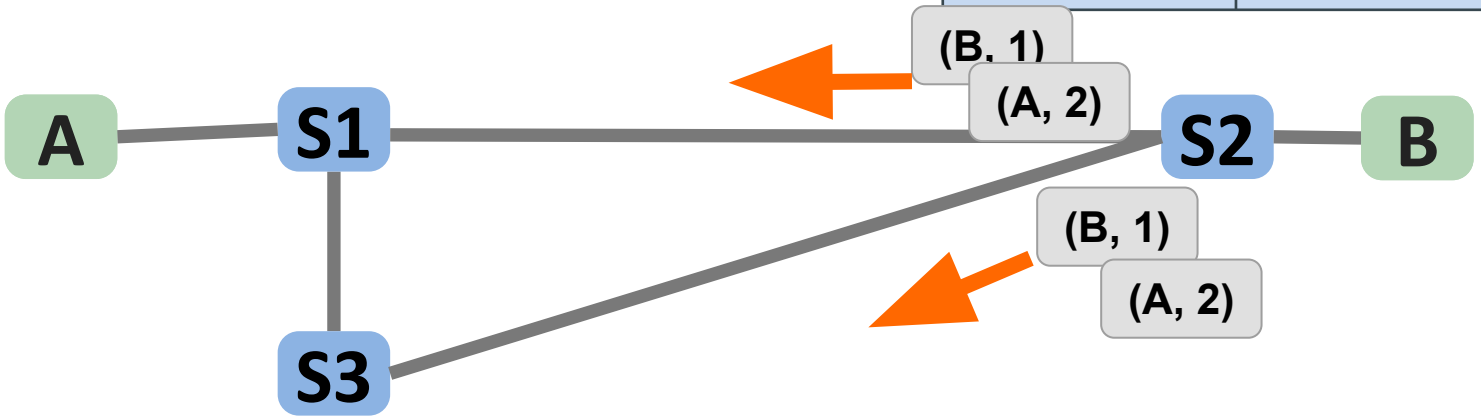


Destination	NextHop, Dist.
A	S1, 2

D-V Review

Destination	NextHop, Dist.
A	Direct, 1

Destination	NextHop, Dist.
B	Direct, 1
A	S1, 2

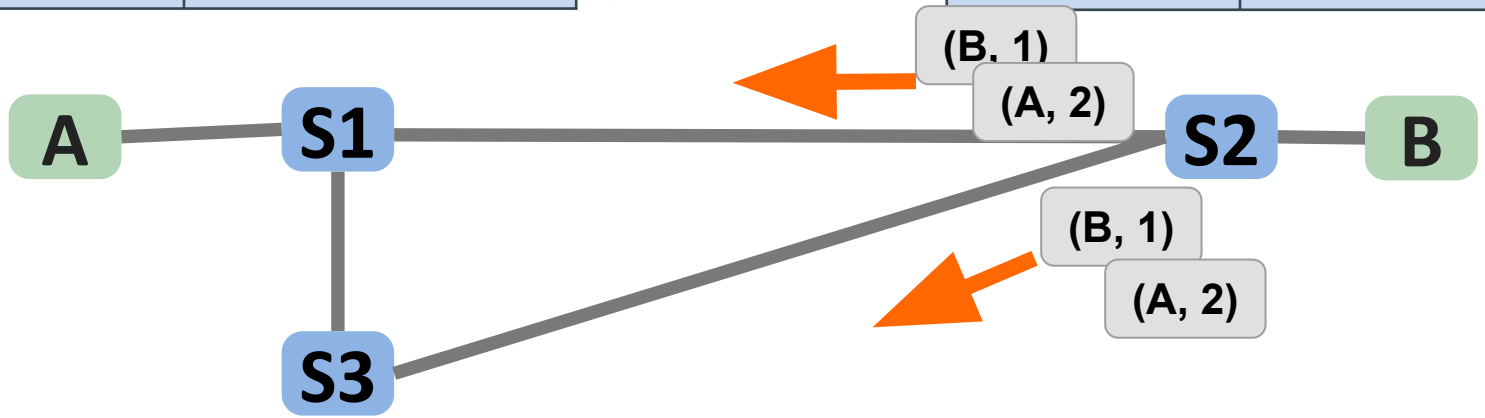


Destination	NextHop, Dist.
A	S1, 2

D-V Review

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 2

Destination	NextHop, Dist.
B	Direct, 1
A	S1, 2

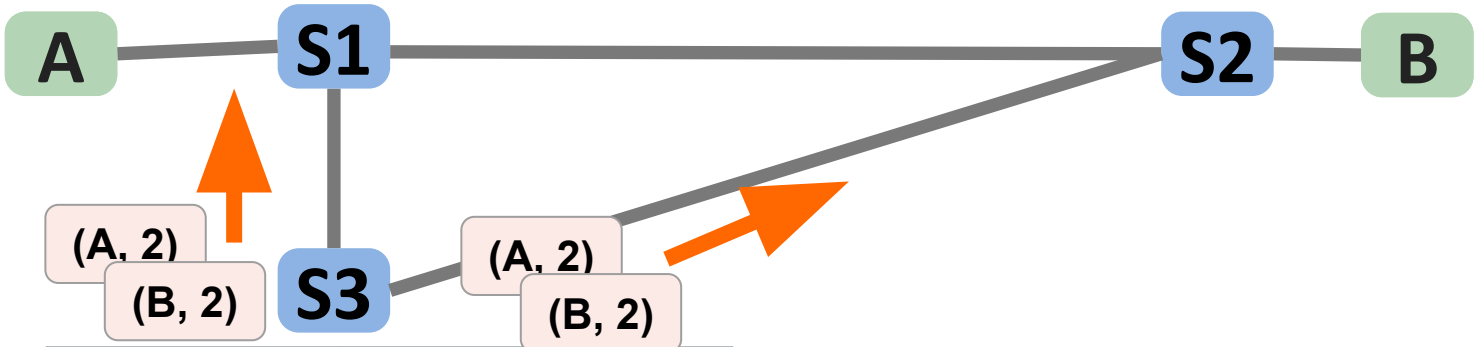


Destination	NextHop, Dist.
A	S1, 2
B	S2, 2

D-V Review

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 2

Destination	NextHop, Dist.
B	Direct, 1
A	S1, 2

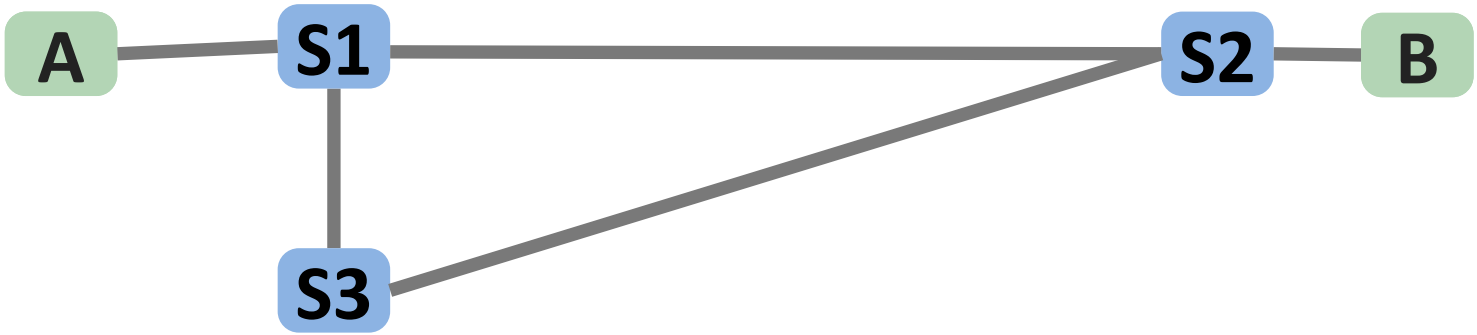


Destination	NextHop, Dist.
A	S1, 2
B	S2, 2

D-V Review (Converged)

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 2

Destination	NextHop, Dist.
B	Direct, 1
A	S1, 2



Destination	NextHop, Dist.
A	S1, 2
B	S2, 2

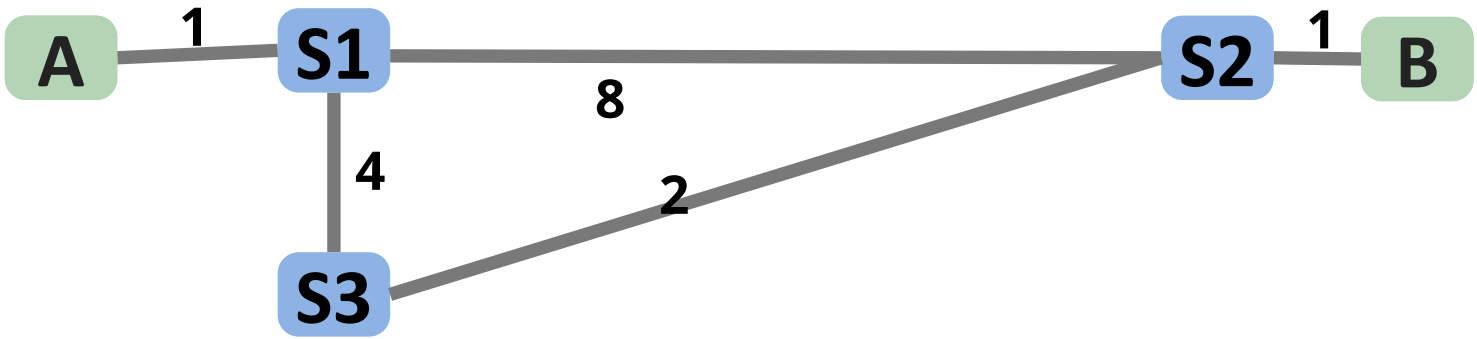
Distances

- What assumptions are we making on each update?
  - S3 gets (A, 1)
  - But stores “A, S1, 2”
  - 2 = 1+1 hop
- Can relax this to be *arbitrary “distance”* values per link
  - S3 gets (A, [advertised cost])
  - But stores “A, S1, [advertised cost] + [link cost]”
    - Assuming min(current cost, proposed cost) is the proposed

D-V Review

Destination	NextHop, Dist.
A	Direct, 1

Destination	NextHop, Dist.
B	Direct, 1

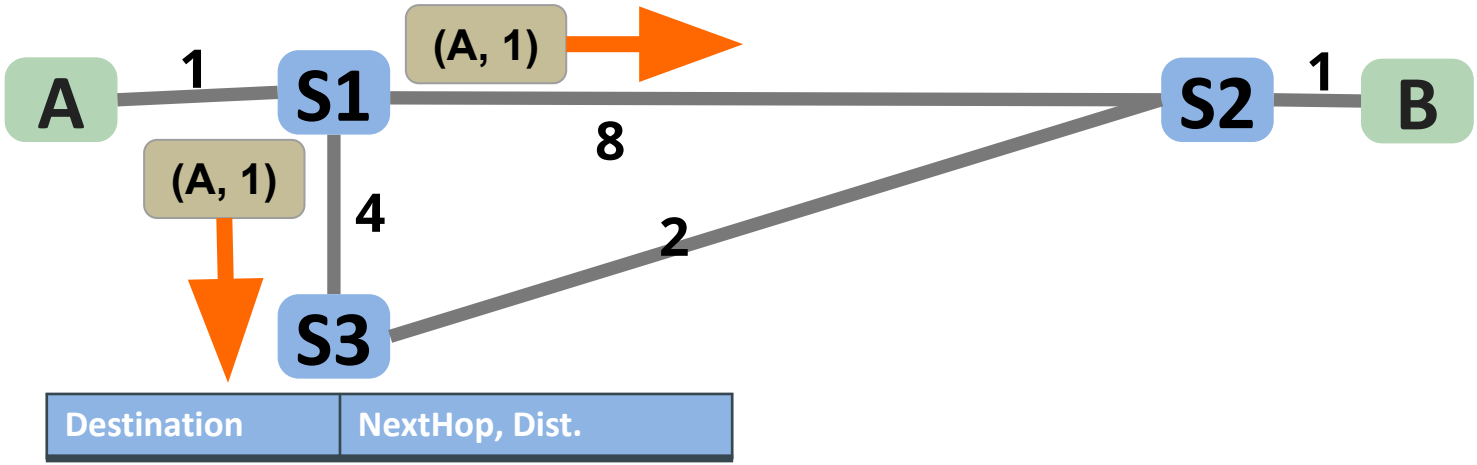


Destination	NextHop, Dist.

D-V Review

Destination	NextHop, Dist.
A	Direct, 1

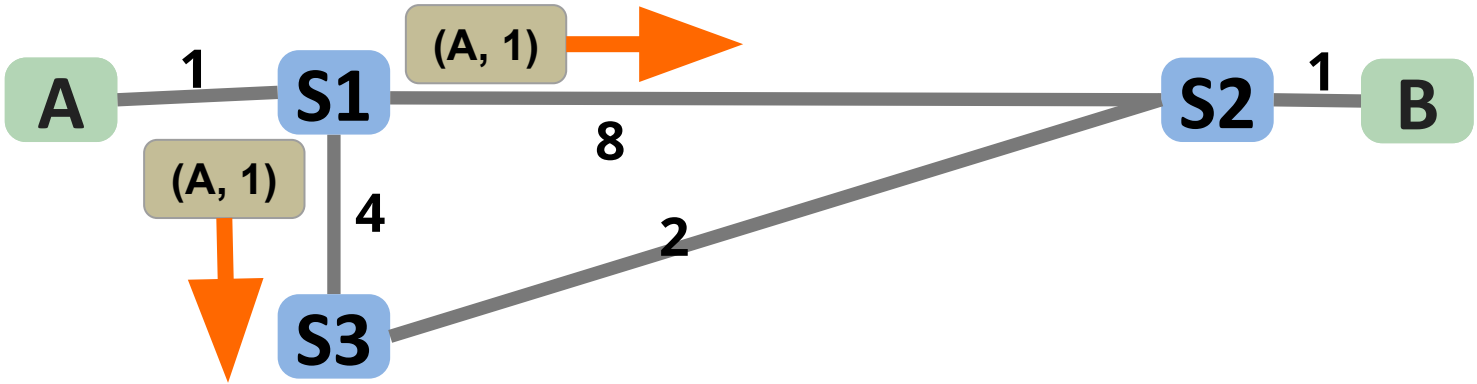
Destination	NextHop, Dist.
B	Direct, 1



D-V Review

Destination	NextHop, Dist.
A	Direct, 1

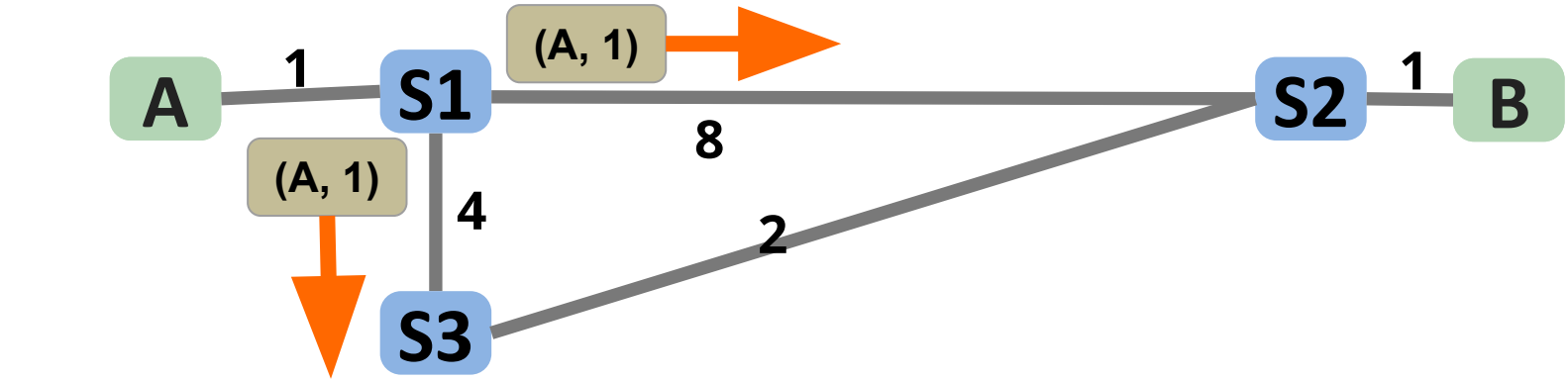
Destination	NextHop, Dist.
B	Direct, 1
A	S1, 9



D-V Review

Destination	NextHop, Dist.
A	Direct, 1

Destination	NextHop, Dist.
B	Direct, 1
A	S1, 9

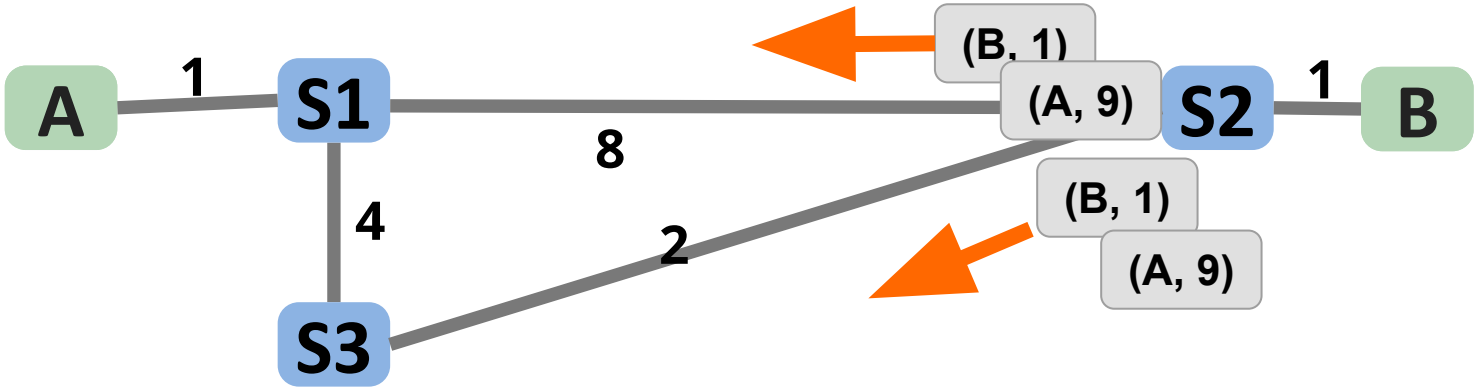


Destination	NextHop, Dist.
A	S1, 5

D-V Review

Destination	NextHop, Dist.
A	Direct, 1

Destination	NextHop, Dist.
B	Direct, 1
A	S1, 9

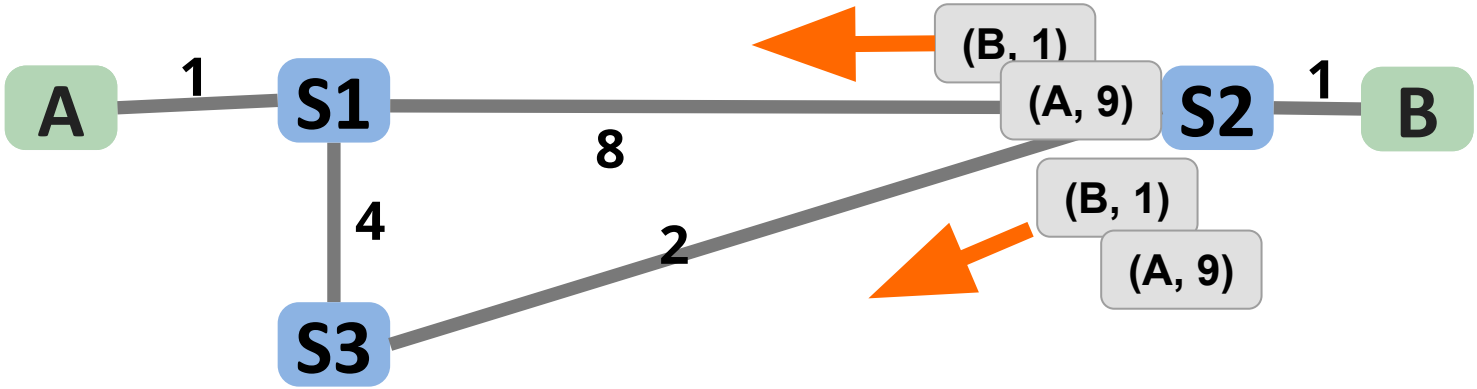


Destination	NextHop, Dist.
A	S1, 5

D-V Review

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 9

Destination	NextHop, Dist.
B	Direct, 1
A	S1, 9

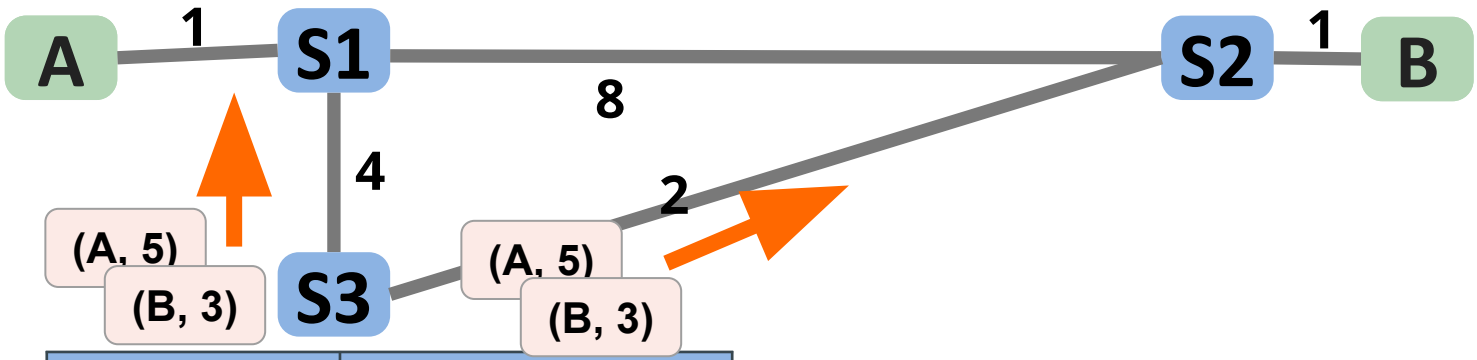


Destination	NextHop, Dist.
A	S1, 5
B	S2, 3

D-V Review

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 9

Destination	NextHop, Dist.
B	Direct, 1
A	S1, 9

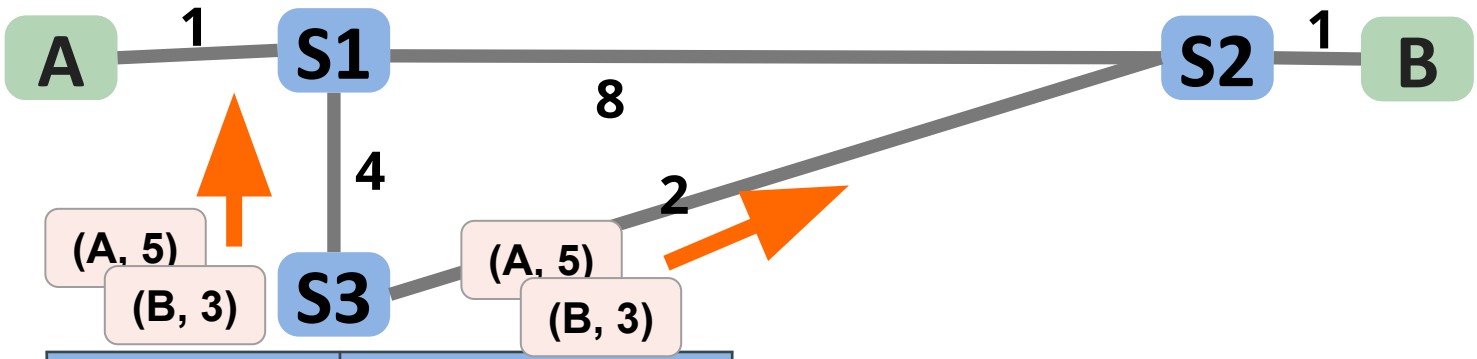


Destination	NextHop, Dist.
A	S1, 5
B	S2, 3

D-V Review

Destination	NextHop, Dist.
A	Direct, 1
B	S3, 7

Destination	NextHop, Dist.
B	Direct, 1
A	S3, 7

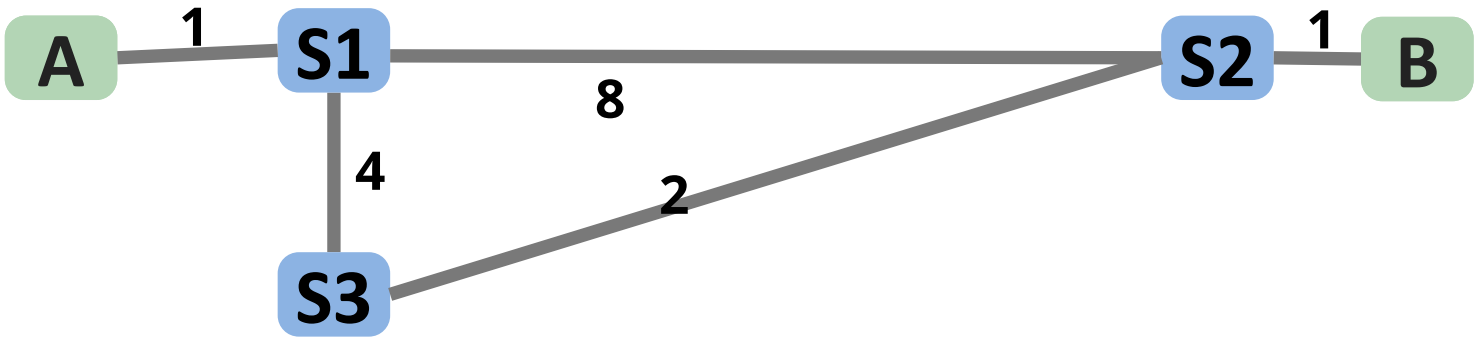


Destination	NextHop, Dist.
A	S1, 5
B	S2, 3

D-V Review (Converged)

Destination	NextHop, Dist.
A	Direct, 1
B	S3, 7

Destination	NextHop, Dist.
B	Direct, 1
A	S3, 7



Destination	NextHop, Dist.
A	S1, 5
B	S2, 3

Distance-Vector Key Points

- Scalable
  - (unlike link-state) routers don’t need global network topology
- Distributed
  - Routers communicate with neighbors to compute routes
- Minimizes Distance
  - Avoids loops and minimizes “distance,” not necessarily physical distance (really can minimize any value: price, latency, etc.)

Two Components of Distance-Vector

- Protocol
  - Each router advertises routes in its table to neighbors
- Algorithm
  - On receiving an advertisement, a router may update its own table

Protocol

Each router advertises its routes to all of its neighbors...

- Periodically
  - Every so many seconds
- Whenever its table changes
  - Due to: new advertisements from neighbors, local link failure, new local link, route timeouts, ...

In *theory*, you only need the first.
In *practice*, the first alone is sufficient to eventually converge, but the second *isn't* if advertisements are dropped. Thus, the second acts like an *optimization*.

Algorithm

Upon receiving advertisement from its neighbor, a router updates its table if one of the following is true:

1. The destination isn't already in its table at all
2. The route in the table is worse than the one advertised
3. The advertiser is the current next hop

```
on new_advertisement A from neighbor N
    if ( A.dst not in table
        OR (A.dist + dist_to_neighbor(N)) < table[A.dst].dist
        OR table[A.dst].next_hop == N ):
      table[A.dst].dist = A.dist + dist_to_neighbor(N)
      table[A.dst].next_hop = N
```

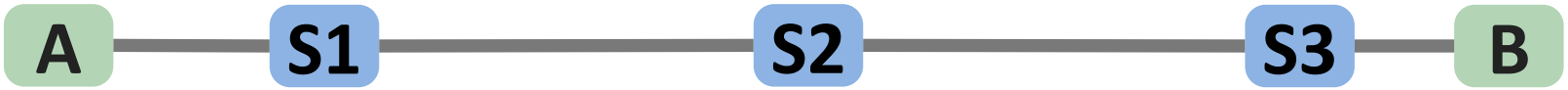
Worksheet

Worksheet Q3 (Split Horizon)

t0 [Fully Converged]

Destination	NextHop, Dist.
A	Direct, 1
?	??/?

Destination	NextHop, Dist.
?	??/?
?	??/?



What is S1's table?

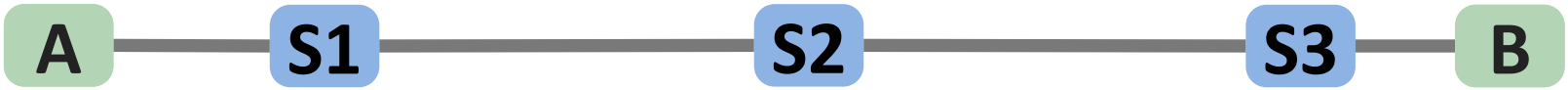
What is S2's table?

Destination	NextHop, Dist.
B	Direct, 1
A	S2, 3

t0 [Fully Converged]

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 3

Destination	NextHop, Dist.
A	S1, 2
B	S3, 2

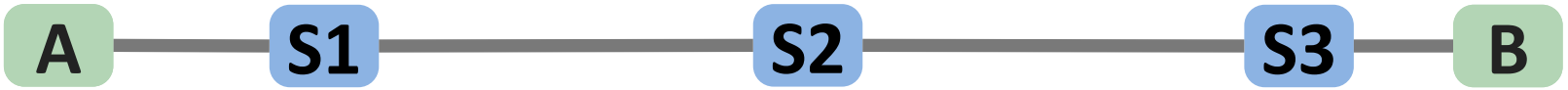


Destination	NextHop, Dist.
B	Direct, 1
A	S2, 3

t0 <now< t1

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 3

Destination	NextHop, Dist.
A	S1, 2
B	S3, 2



(?,?)

(?,?)

Destination	NextHop, Dist.
B	Direct, 1
A	S2, 3

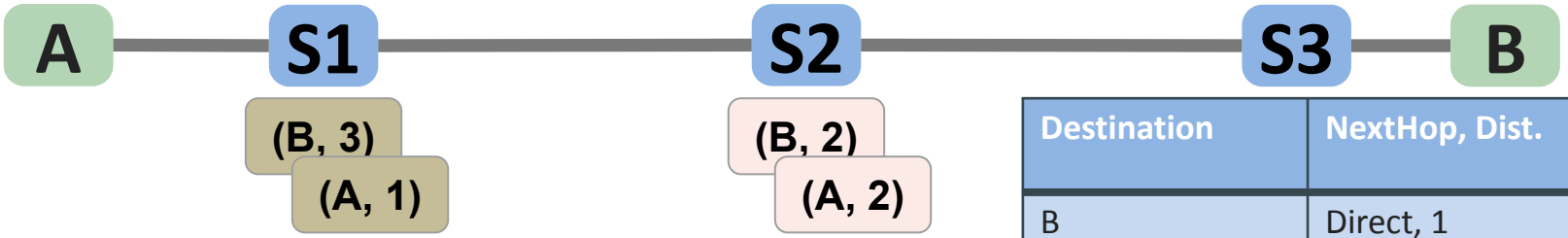
What does S1 send to S2?

What does S2 send to S1?

t0 < t1

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 3

Destination	NextHop, Dist.
A	S1, 2
B	S3, 2

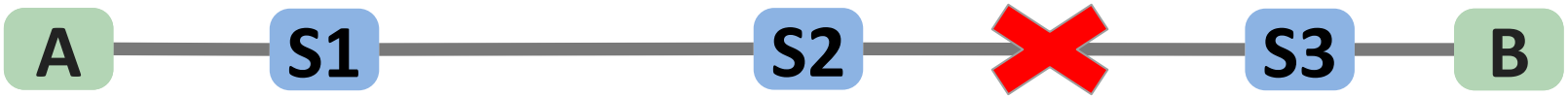


Destination	NextHop, Dist.
B	Direct, 1
A	S2, 3

t1 [S2--S3 link goes down]

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 3

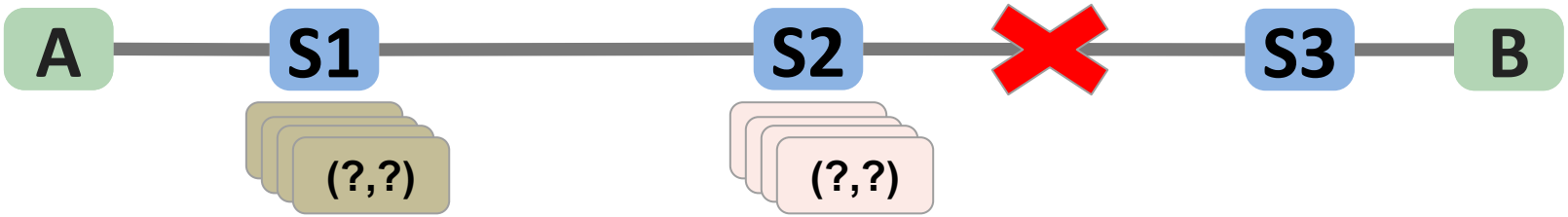
Destination	NextHop, Dist.
A	S1, 2
B	S3, 2



t1<now<t2

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 3

Destination	NextHop, Dist.
A	S1, 2
B	S3, 2



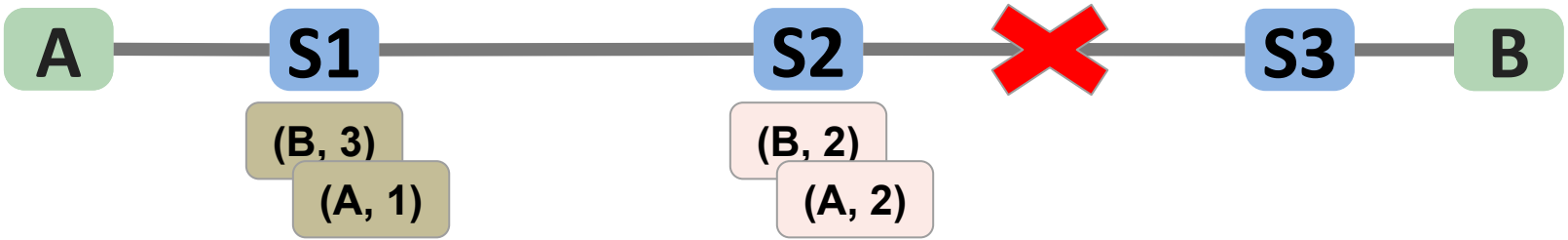
What does S1 send to S2?

What does S2 send to S1?

t1<now<t2

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 3

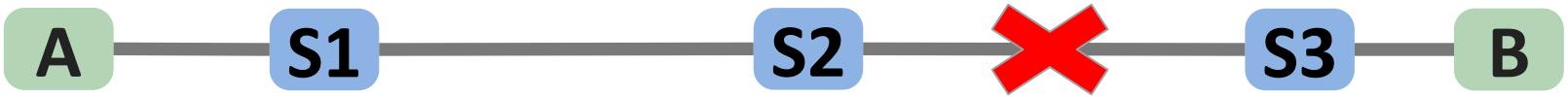
Destination	NextHop, Dist.
A	S1, 2
B	S3, 2



t2 [S2's route to B via S3 expires]

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 3

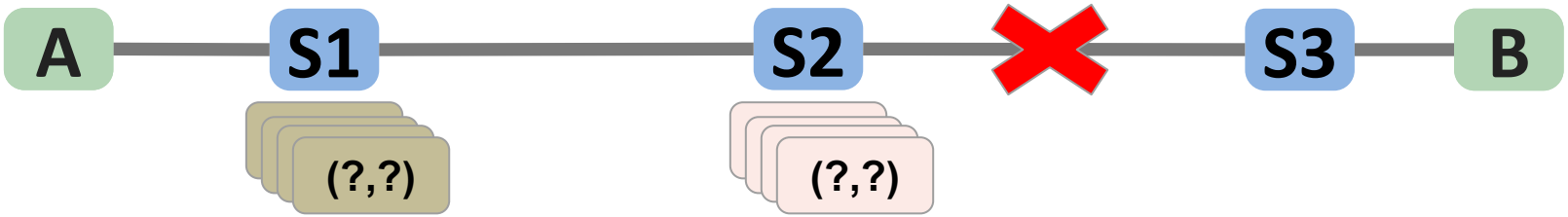
Destination	NextHop, Dist.
A	S1, 2
-	--



t2<now<t3

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 3

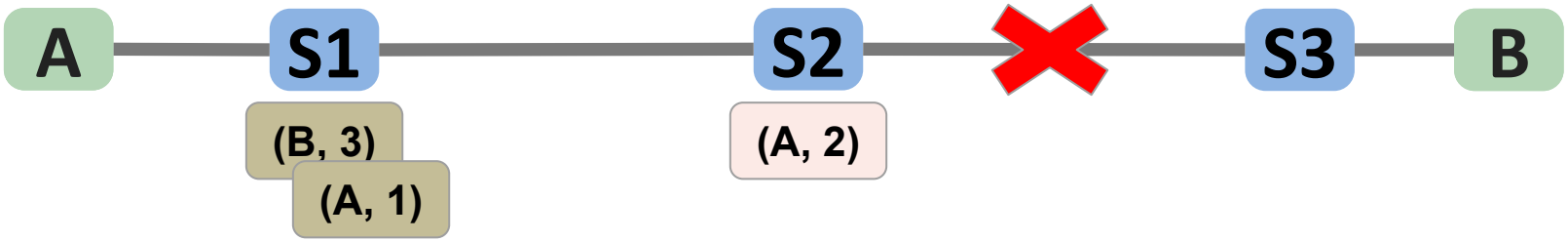
Destination	NextHop, Dist.
A	S1, 2
-	--



t2<now<t3

Destination	NextHop, Dist.
A	Direct, 1
?	??/?

Destination	NextHop, Dist.
?	??/?
?	??/?



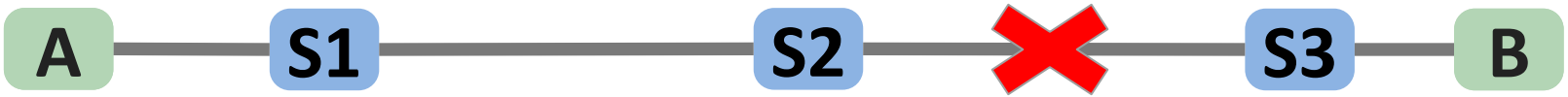
What does S1's table look like? (after receiving S2's message)

What does S1's table look like? (after receiving S1's messages)

t2<now<t3

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 3

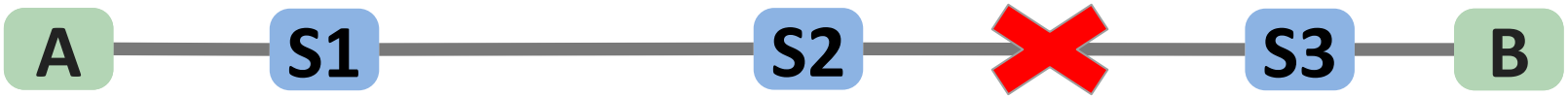
Destination	NextHop, Dist.
A	S1, 2
B	S1, 4



t3 [S2's route to B via S3 expires]

Destination	NextHop, Dist.
A	Direct, 1
-	--

Destination	NextHop, Dist.
A	S1, 2
B	S1, 4

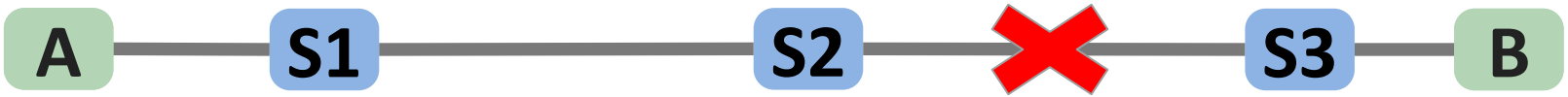


Why does that route expire?

t3 [S2's route to B via S3 expires]

Destination	NextHop, Dist.
A	Direct, 1
-	--

Destination	NextHop, Dist.
A	S1, 2
B	S1, 4

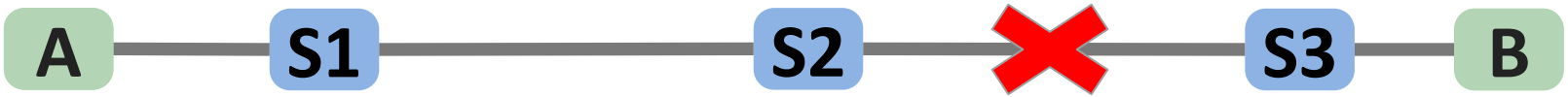


Because it hasn't been updated recently

t3<now<t4

Destination	NextHop, Dist.
A	Direct, 1
?	??/?

Destination	NextHop, Dist.
?	??/?
?	??/?



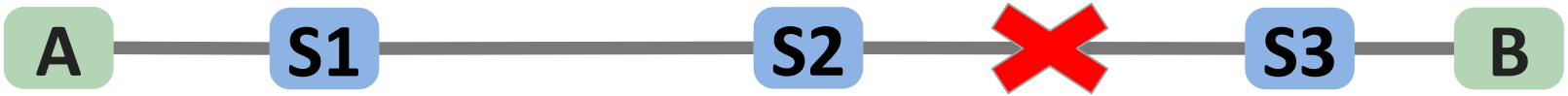
After S1 receives S2's message, what will S1's table look like?

After S2 receives S1's message, what will S2's table look like?

t3<now<t4

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 5

Destination	NextHop, Dist.
A	S1, 2
B	S2, 4



Because it hasn't been updated recently

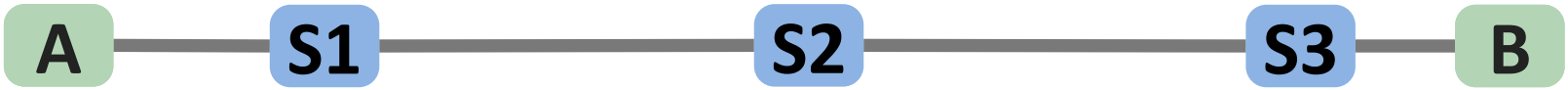
How can we fix this?

- Split Horizon!
  - If you route to destination D through neighbor N:
    - Don’t send updates about D to N

t0 [Fully Converged]

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 3

Destination	NextHop, Dist.
A	S1, 2
B	S3, 2

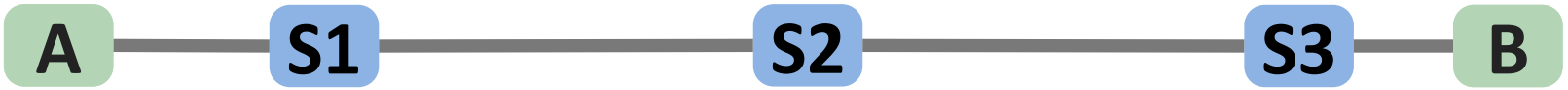


Destination	NextHop, Dist.
B	Direct, 1
A	S2, 3

t0 <now< t1

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 3

Destination	NextHop, Dist.
A	S1, 2
B	S3, 2



(?,?)

(?,?)

Destination	NextHop, Dist.
B	Direct, 1
A	S2, 3

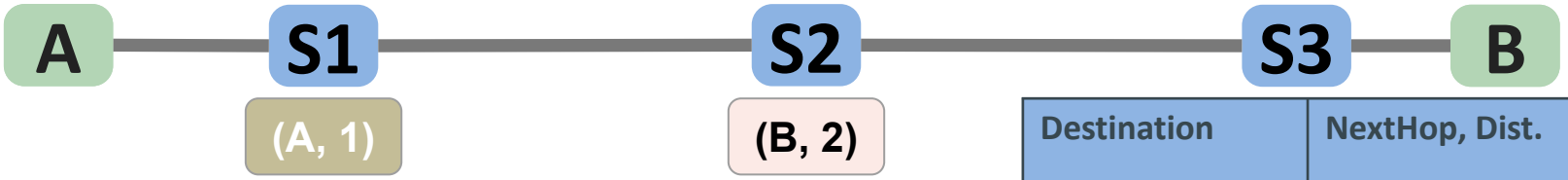
What does S1 send to S2?

What does S2 send to S1?

t0 < t1

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 3

Destination	NextHop, Dist.
A	S1, 2
B	S3, 2



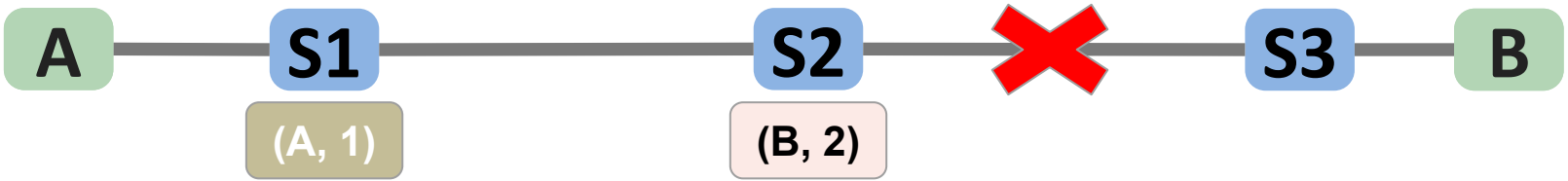
Destination	NextHop, Dist.
B	Direct, 1
A	S2, 3

This is different!

t1 [S2--S3 link goes down]

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 3

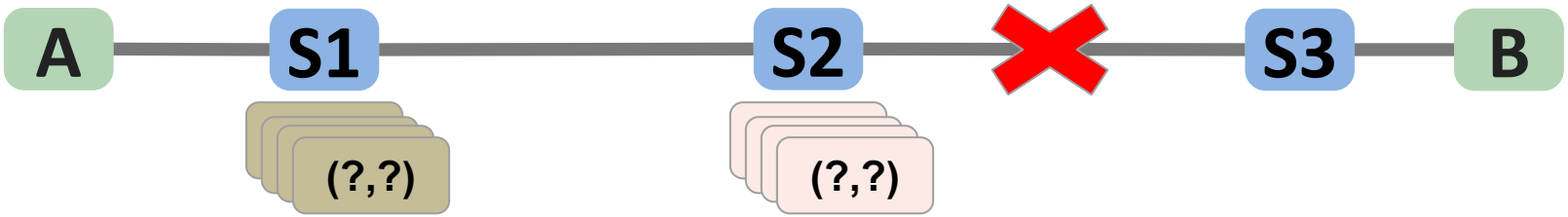
Destination	NextHop, Dist.
A	S1, 2
B	S3, 2



t1<now<t2

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 3

Destination	NextHop, Dist.
A	S1, 2
B	S3, 2



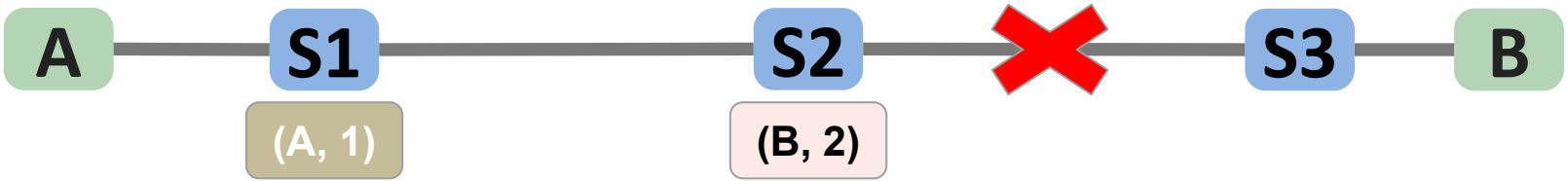
What does S1 send to S2?

What does S2 send to S1?

t1<now<t2

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 3

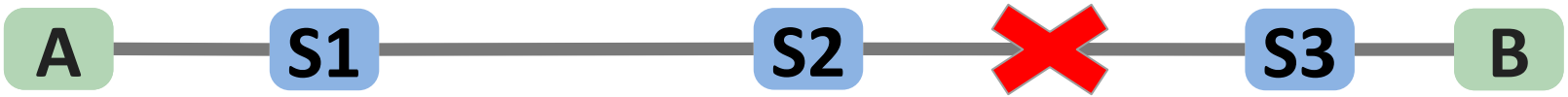
Destination	NextHop, Dist.
A	S1, 2
B	S3, 2



t2 [S2's route to B via S3 expires]

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 3

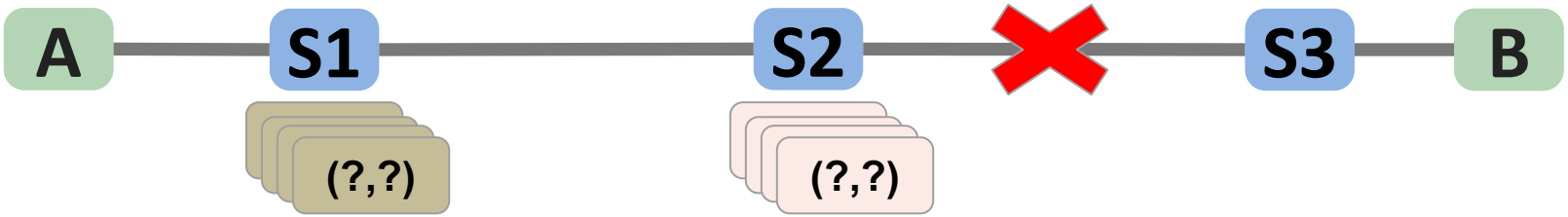
Destination	NextHop, Dist.
A	S1, 2
-	--



t2<now<t3

Destination	NextHop, Dist.
A	Direct, 1
B	S2, 3

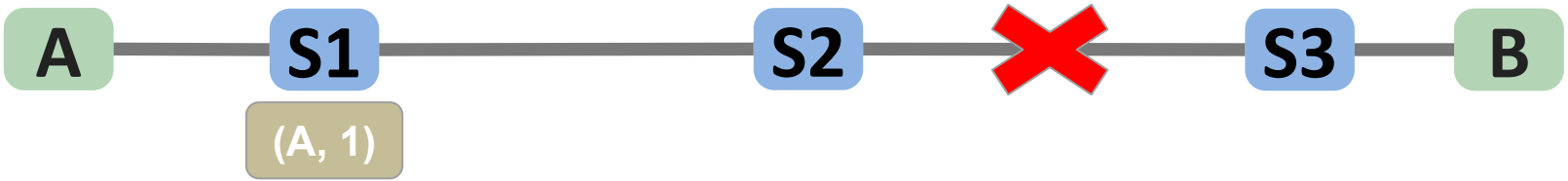
Destination	NextHop, Dist.
A	S1, 2
-	--



t2<now<t3

Destination	NextHop, Dist.
A	Direct, 1
?	??/?

Destination	NextHop, Dist.
?	??/?
?	??/?



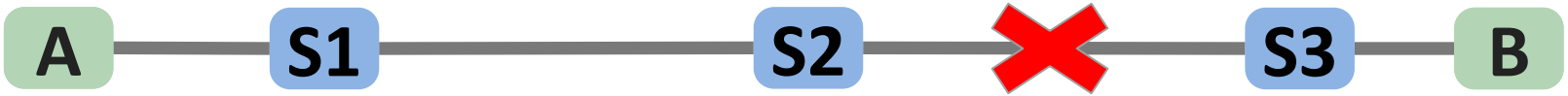
What does S1's table look like? (after receiving S2's message)

What does S1's table look like? (after receiving S1's messages)

t2<now<t3

Destination	NextHop, Dist.
A	Direct, 1
-	--

Destination	NextHop, Dist.
A	S1, 2
-	--



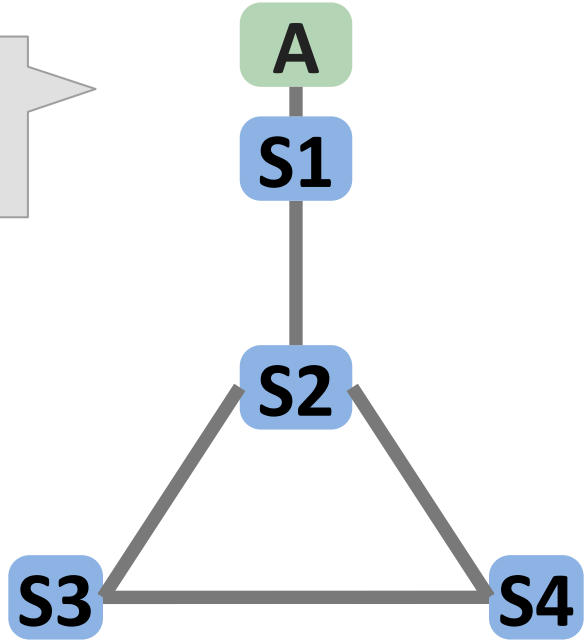
No problems here!

Does this a fix everything?

- Nope,
  - Count-To-Infinity can still happen with Split Horizon

t0 [Fully Converged]

What messages are sent?



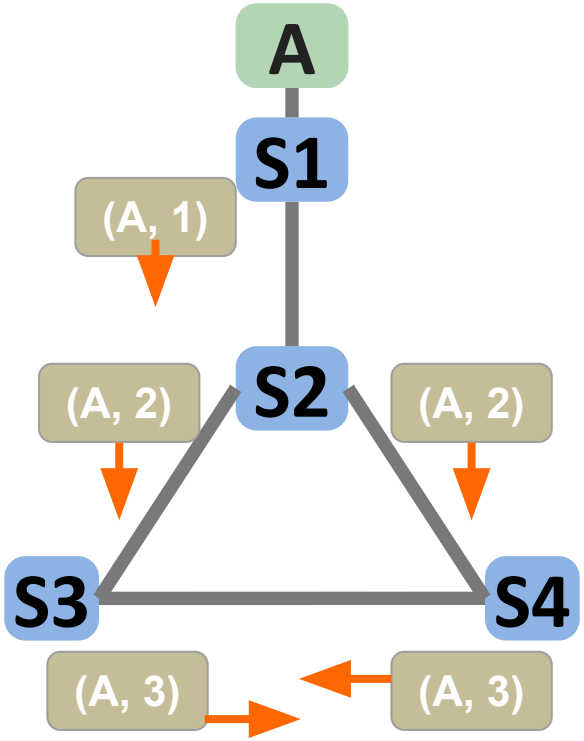
Destination	NextHop, Dist.
A	Direct, 1

Destination	NextHop, Dist.
A	S1, 2

Destination	NextHop, Dist.
A	S2, 3

Destination	NextHop, Dist.
A	S2, 3

t0 < now < t1



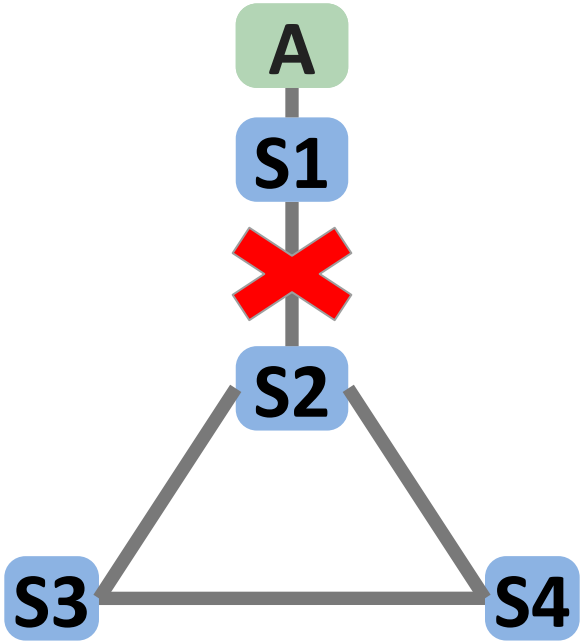
Destination	NextHop, Dist.
A	Direct, 1

Destination	NextHop, Dist.
A	S1, 2

Destination	NextHop, Dist.
A	S2, 3

Destination	NextHop, Dist.
A	S2, 3

t1 [S1--S2 link goes down]

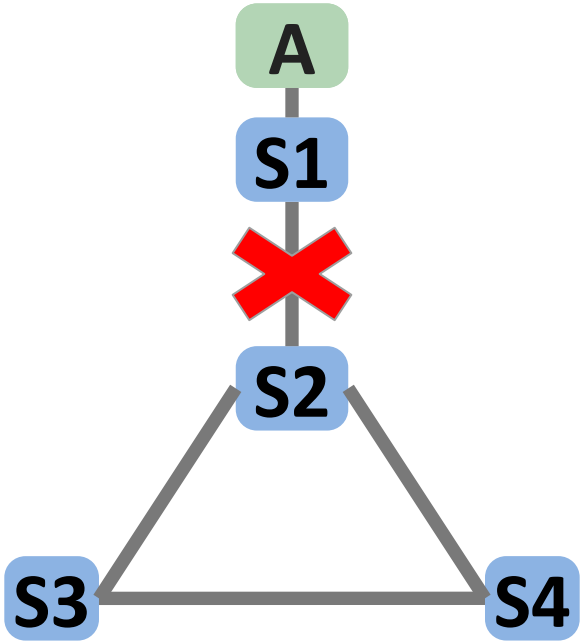


Destination	NextHop, Dist.
A	S1, 2

Destination	NextHop, Dist.
A	S2, 3

Destination	NextHop, Dist.
A	S2, 3

t2 [S2's & S3's routes expire]



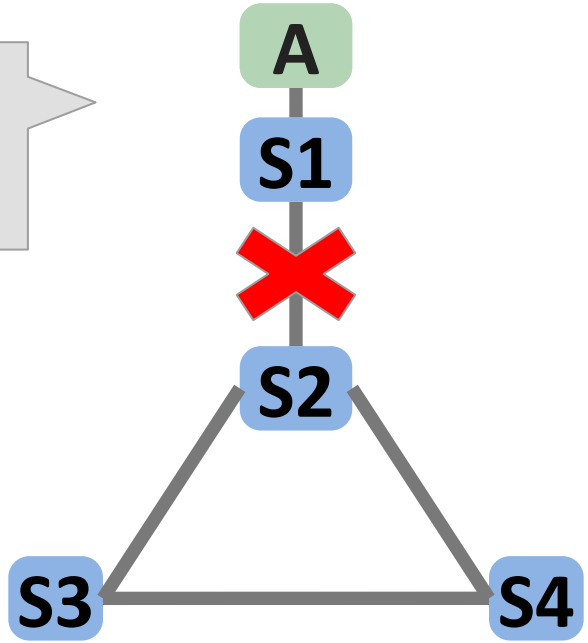
Destination	NextHop, Dist.
-	--

Destination	NextHop, Dist.
-	--

Destination	NextHop, Dist.
A	S2, 3

t3

What messages are sent?



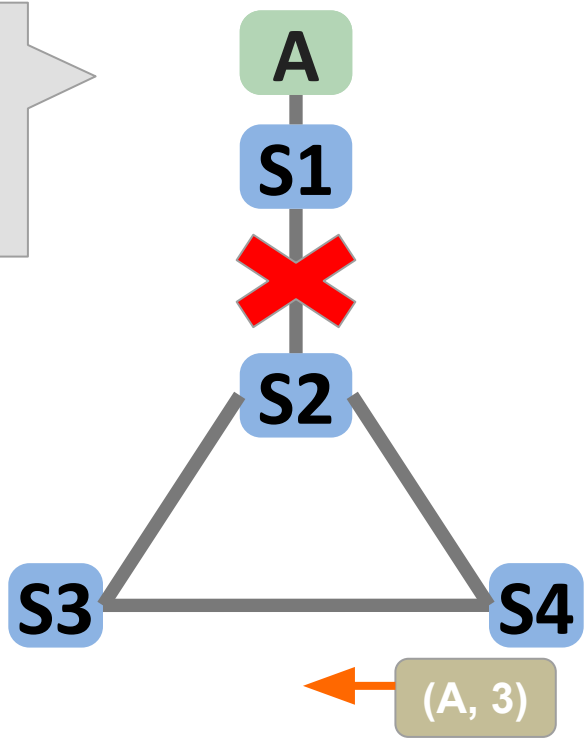
Destination	NextHop, Dist.
-	--

Destination	NextHop, Dist.
-	--

Destination	NextHop, Dist.
A	S2, 3

t3

How do forwarding tables change?



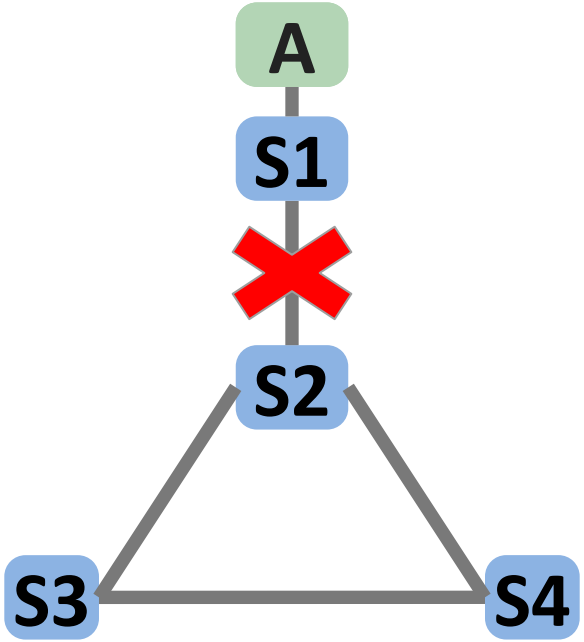
Split Horizon prevents a message to S2

Destination	NextHop, Dist.
-	--

Destination	NextHop, Dist.
-	--

Destination	NextHop, Dist.
A	S2, 3

t3



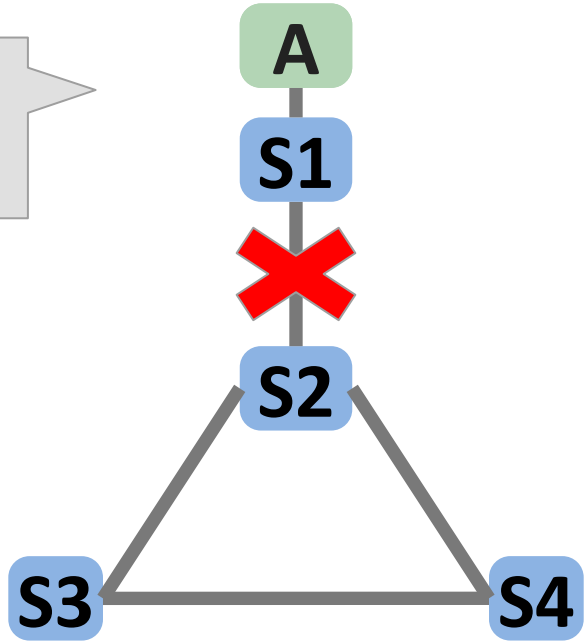
Destination	NextHop, Dist.
-	--

Destination	NextHop, Dist.
A	S4, 4

Destination	NextHop, Dist.
A	S2, 3

t4

What messages are sent?



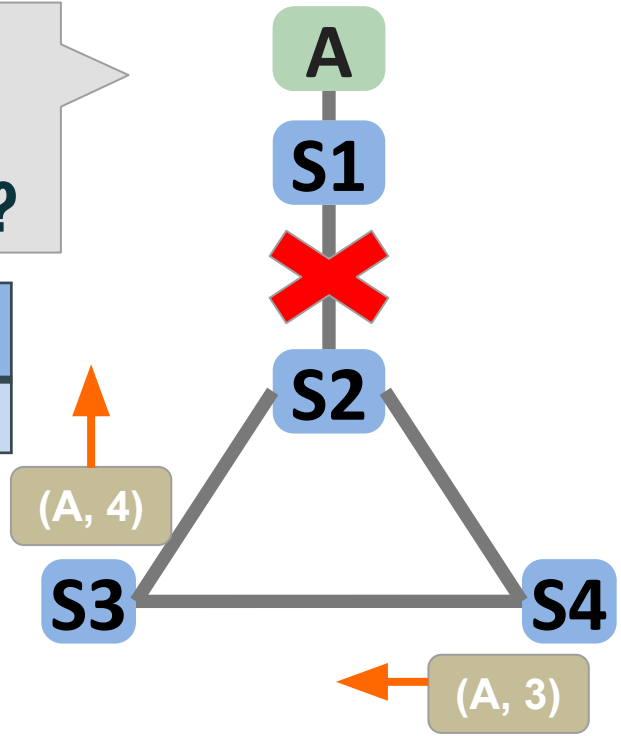
Destination	NextHop, Dist.
-	--

Destination	NextHop, Dist.
A	S4, 4

Destination	NextHop, Dist.
A	S2, 3

t4

How do forwarding tables change?

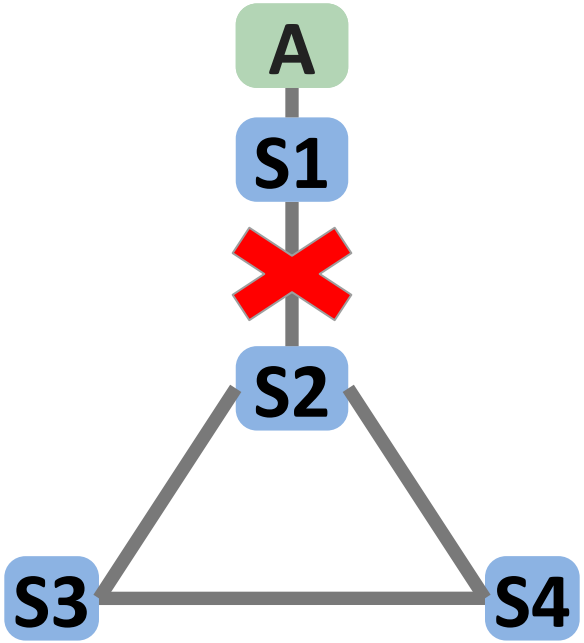


Destination	NextHop, Dist.
-	--

Destination	NextHop, Dist.
A	S4, 4

Destination	NextHop, Dist.
A	S2, 3

t4



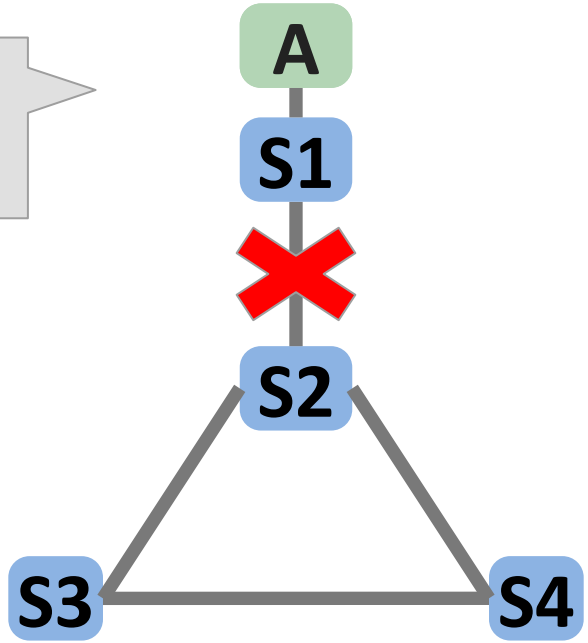
Destination	NextHop, Dist.
A	S3, 5

Destination	NextHop, Dist.
A	S4, 4

Destination	NextHop, Dist.
A	S2, 3

t5

What messages are sent?



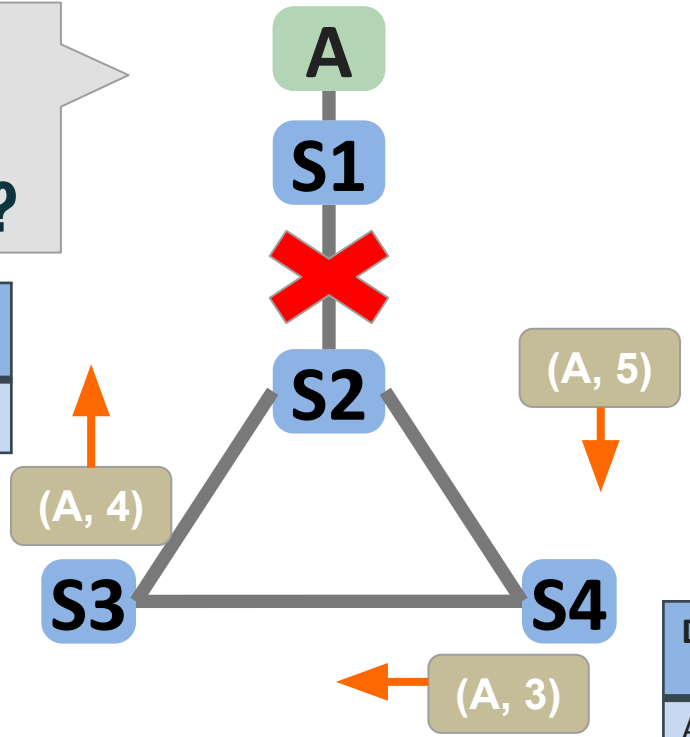
Destination	NextHop, Dist.
A	S3, 5

Destination	NextHop, Dist.
A	S4, 4

Destination	NextHop, Dist.
A	S2, 3

t5

How do forwarding tables change?

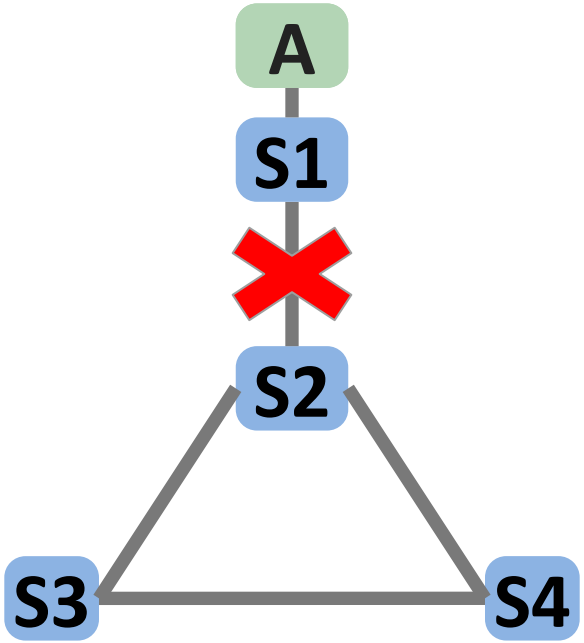


Destination	NextHop, Dist.
A	S3, 5

Destination	NextHop, Dist.
A	S4, 4

Destination	NextHop, Dist.
A	S2, 3

t5



S4's NextHop updated its distance, so S4 must update its distance as well

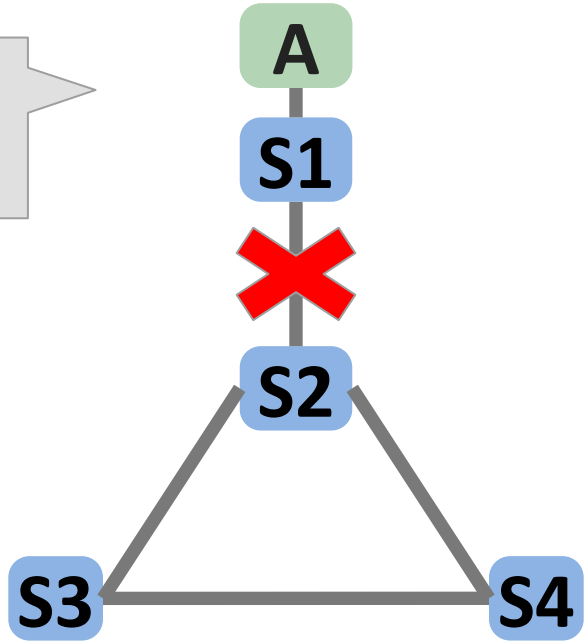
Destination	NextHop, Dist.
A	S3, 5

Destination	NextHop, Dist.
A	S4, 4

Destination	NextHop, Dist.
A	S2, 6

t6

What messages are sent?



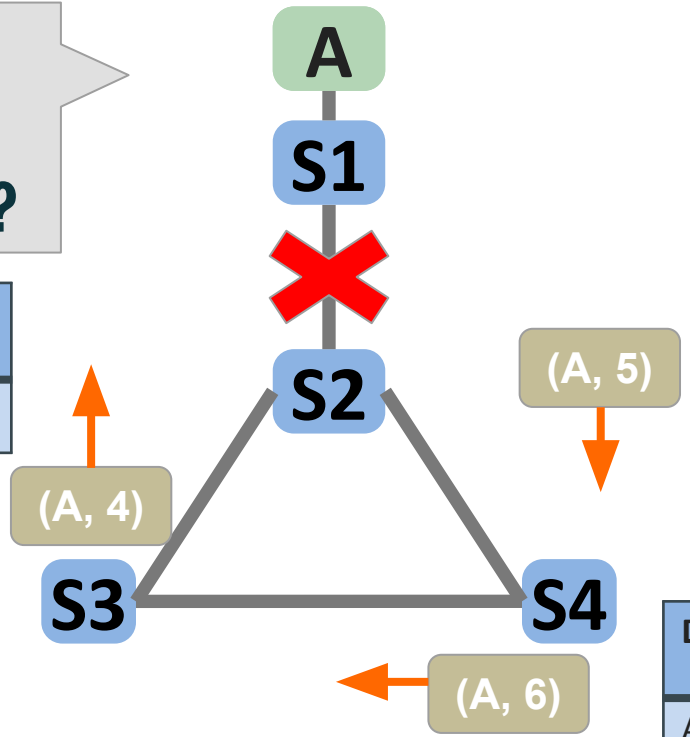
Destination	NextHop, Dist.
A	S3, 5

Destination	NextHop, Dist.
A	S4, 4

Destination	NextHop, Dist.
A	S2, 6

t6

How do forwarding tables change?

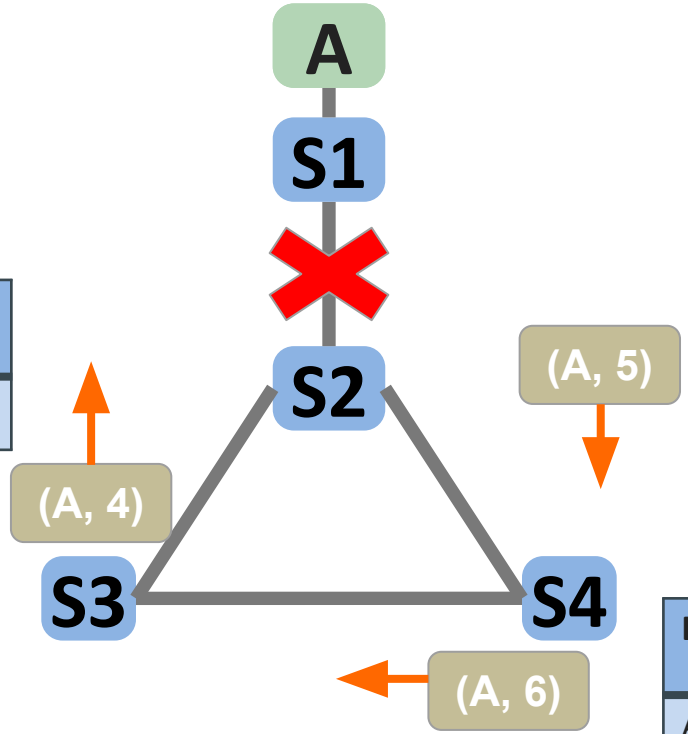


Destination	NextHop, Dist.
A	S3, 5

Destination	NextHop, Dist.
A	S4, 4

Destination	NextHop, Dist.
A	S2, 6

t6



This seems like the start of something bad

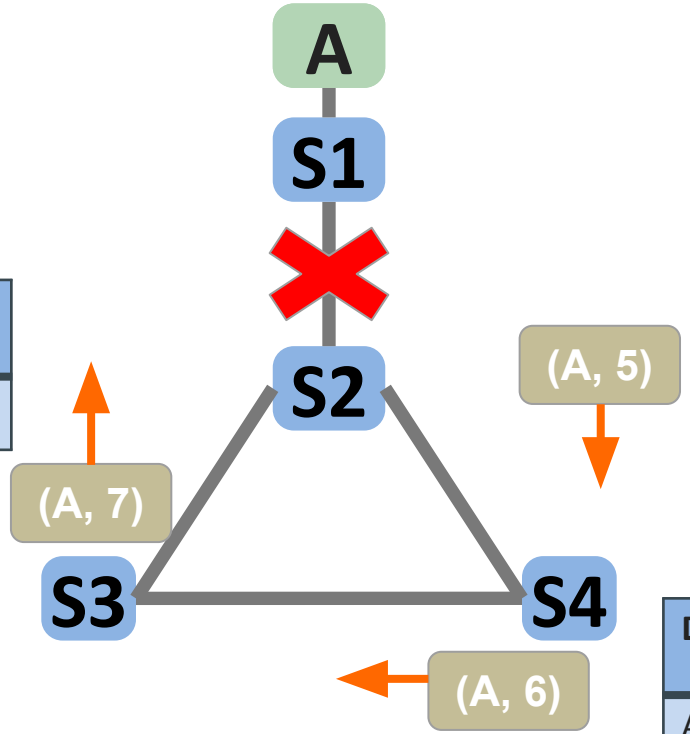
Destination	NextHop, Dist.
A	S3, 5

Destination	NextHop, Dist.
A	S4, 7

Destination	NextHop, Dist.
A	S2, 6

t7

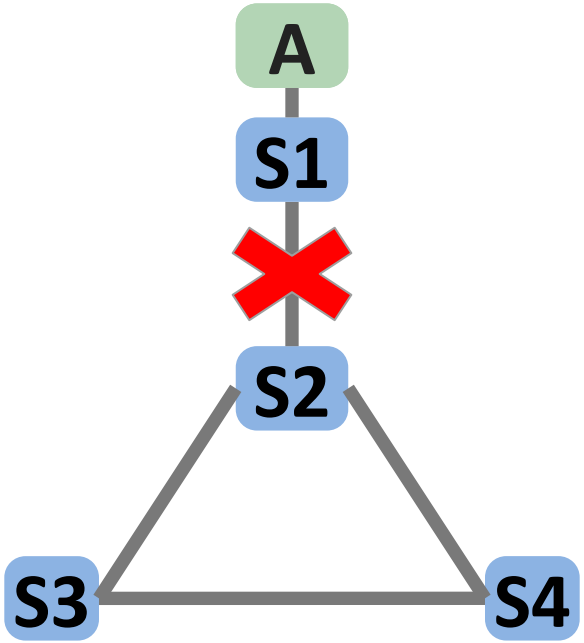
Destination	NextHop, Dist.
A	S3, 5



Destination	NextHop, Dist.
A	S4, 7

Destination	NextHop, Dist.
A	S2, 6

t7

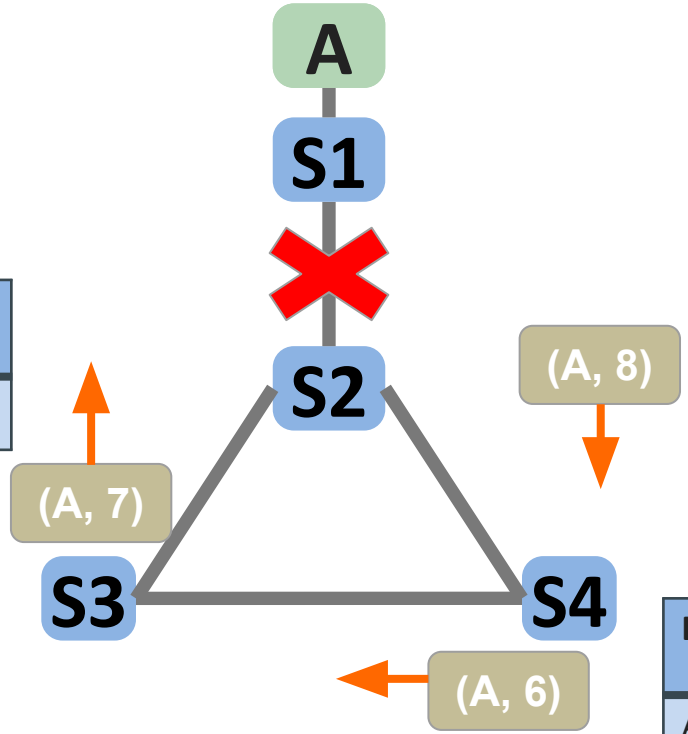


Destination	NextHop, Dist.
A	S3, 8

Destination	NextHop, Dist.
A	S4, 7

Destination	NextHop, Dist.
A	S2, 6

t8

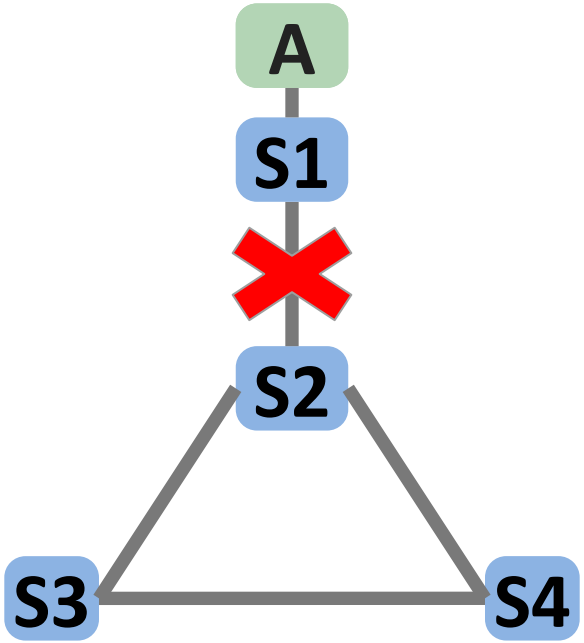


Destination	NextHop, Dist.
A	S3, 8

Destination	NextHop, Dist.
A	S4, 7

Destination	NextHop, Dist.
A	S2, 6

t8



Destination	NextHop, Dist.
A	S3, 8

Destination	NextHop, Dist.
A	S4, 7

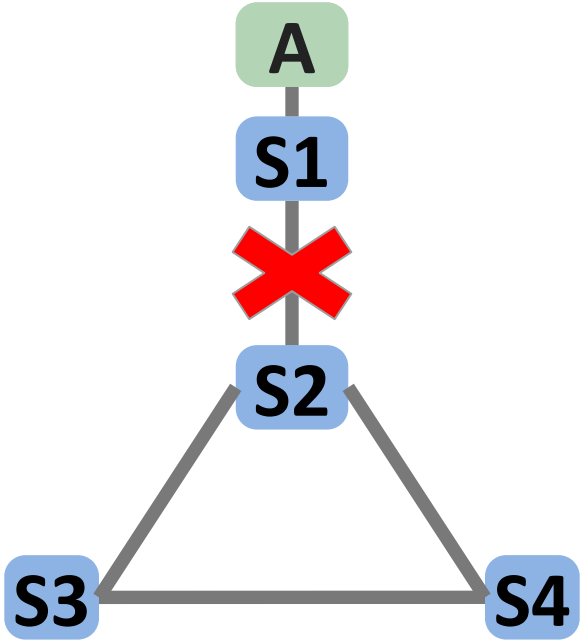
Destination	NextHop, Dist.
A	S2, 9

Oh wait…
We've been here before

t5

Back at t5

Destination	NextHop, Dist.
A	S3, 5



Same next-hop
Smaller distance
Same problem

Destination	NextHop, Dist.
A	S4, 4

Destination	NextHop, Dist.
A	S2, 6

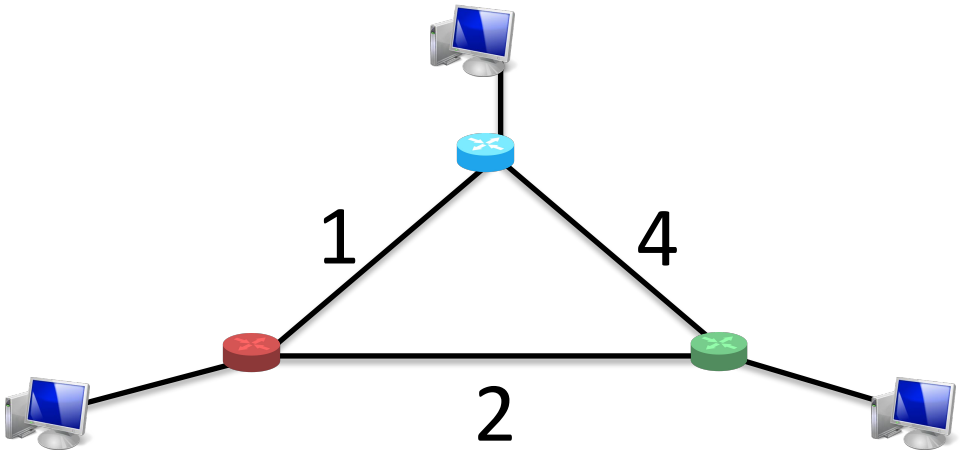
Poison Reverse vs. Route Poisoning

- Poison Reverse
  - Instead of not advertising a route back to its next hop…
  - .. advertise ∞ to its next hop!
  - May mean you advertise different things to different neighbors!
- Route Poisoning
  - Instead of removing a route (e.g., due to a timeout)...
  - .. change its distance to ∞ and continue advertising!
  - Same information sent to every neighbor
- In both cases, instead of omitting a “bad” route, you explicitly advertise it as bad

Link State Routing

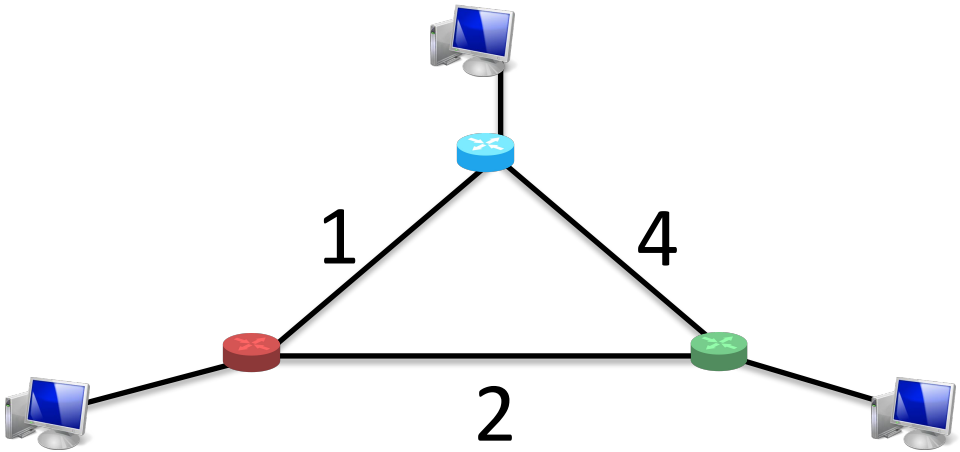
Each router knows its own local “link state”:

- State of each link to its neighbor (up/down)
- Associated costs



Link State Routing

1. Router floods its link state to all other routers.
2. Each router learns global network topology
3. Then, computes shortest path themselves!
    - With Dijkstra's, etc



Link State Routing

1. Router floods its link state to all other routers.
2. Each router learns global network topology
3. Then, computes shortest path themselves!
    - With Dijkstra's, etc

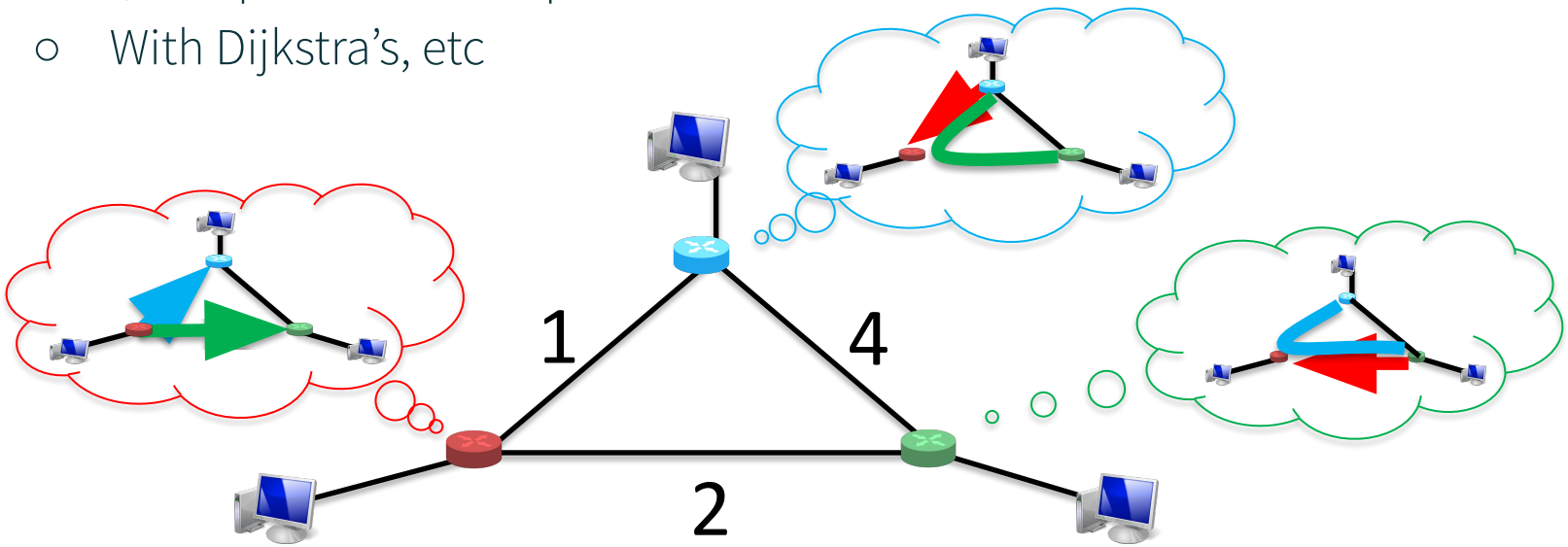


Diagram for Q1