

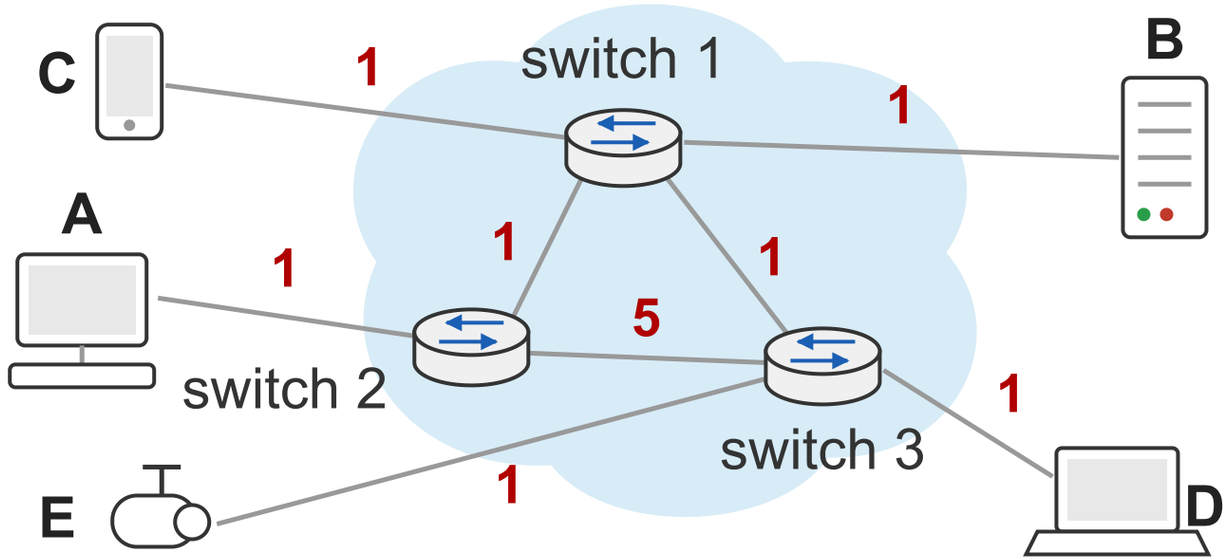
UNIVERSITY OF WATERLOO

Finding paths - Part II

CS 456/656, Fall 2026

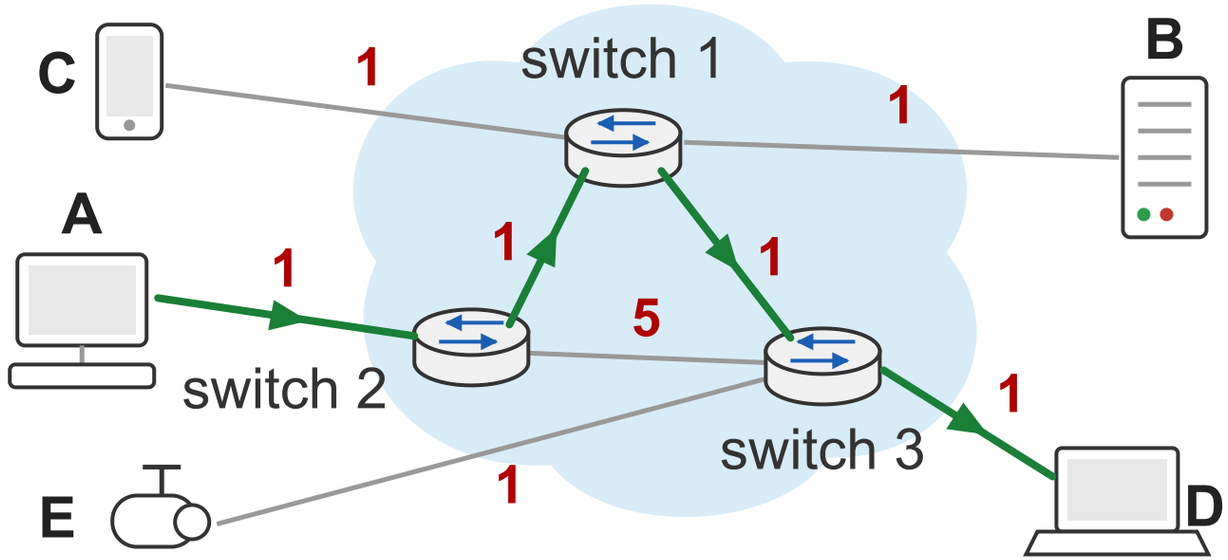
Mina Tahmasbi Arashloo

Recap: least-cost paths



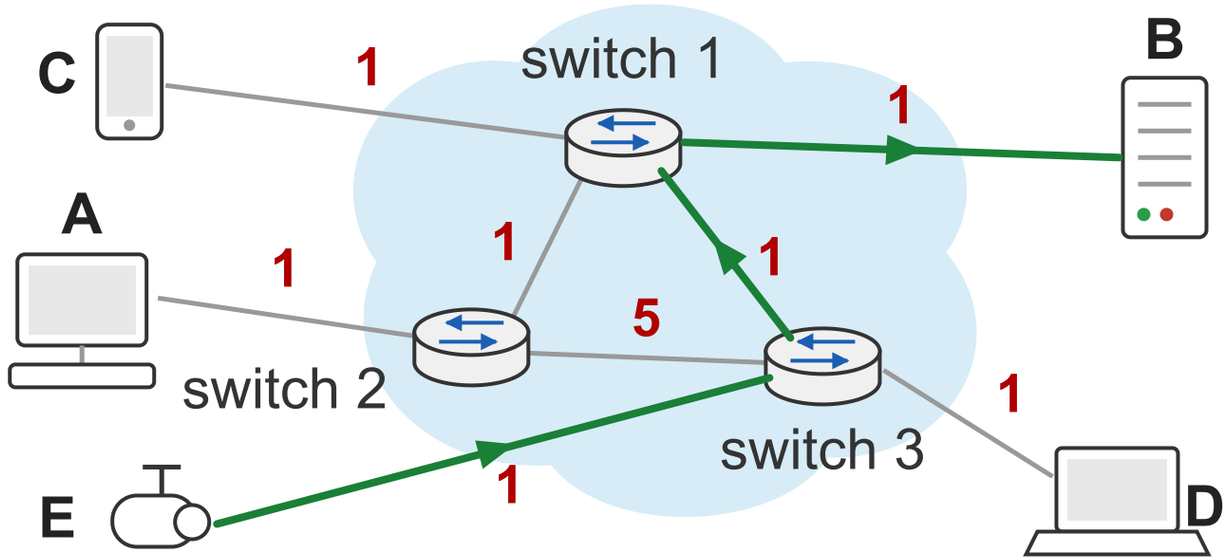
- Every link is given a cost (e.g., by the operator and/or measured by the switches)
- Best path: the one with the least total cost (sum of its links' costs)
- The routing algorithm (distributed or centralized) finds the least-cost paths and populates each forwarding table (destination → next hop)

Recap: 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)

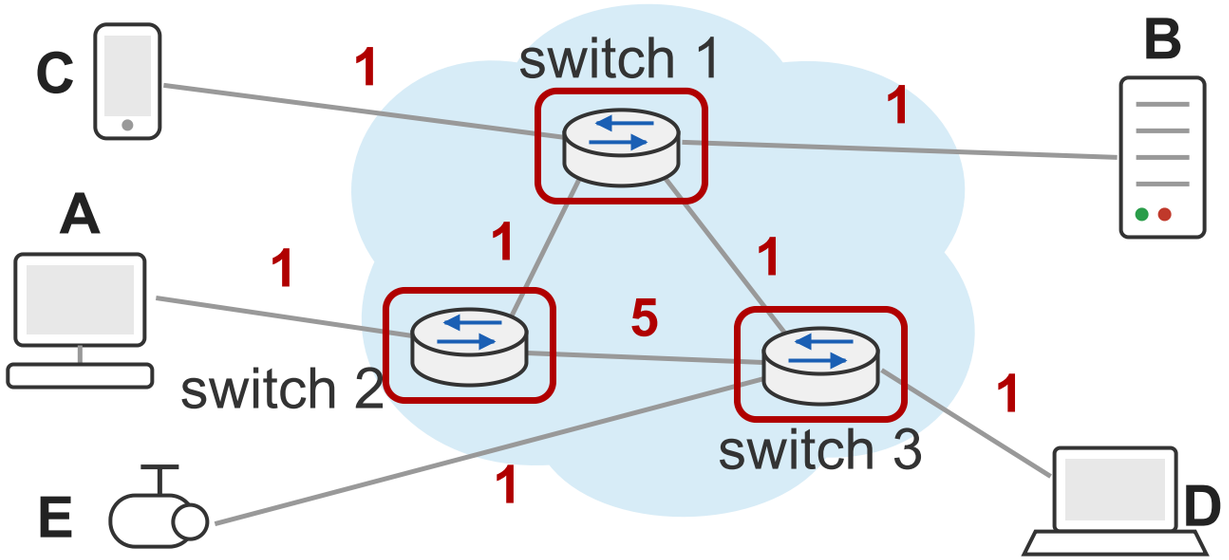
Recap: least-cost paths



- 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)

Implementing least-cost routing in a distributed control plane

- Suppose there is no controller.
- You can write one program that runs, independently, on the control plane of each switch (in red).
- The program on one switch can send and receive messages to and from the program on the neighbouring switches.
- Goal: each switch should learn the next hop on the least-cost path to every destination.



The control plane program (in-class)

```
void on_start() {
    // TODO
}

// runs when msg arrives on link k
void on_message(k, msg) {
    // TODO
}
```

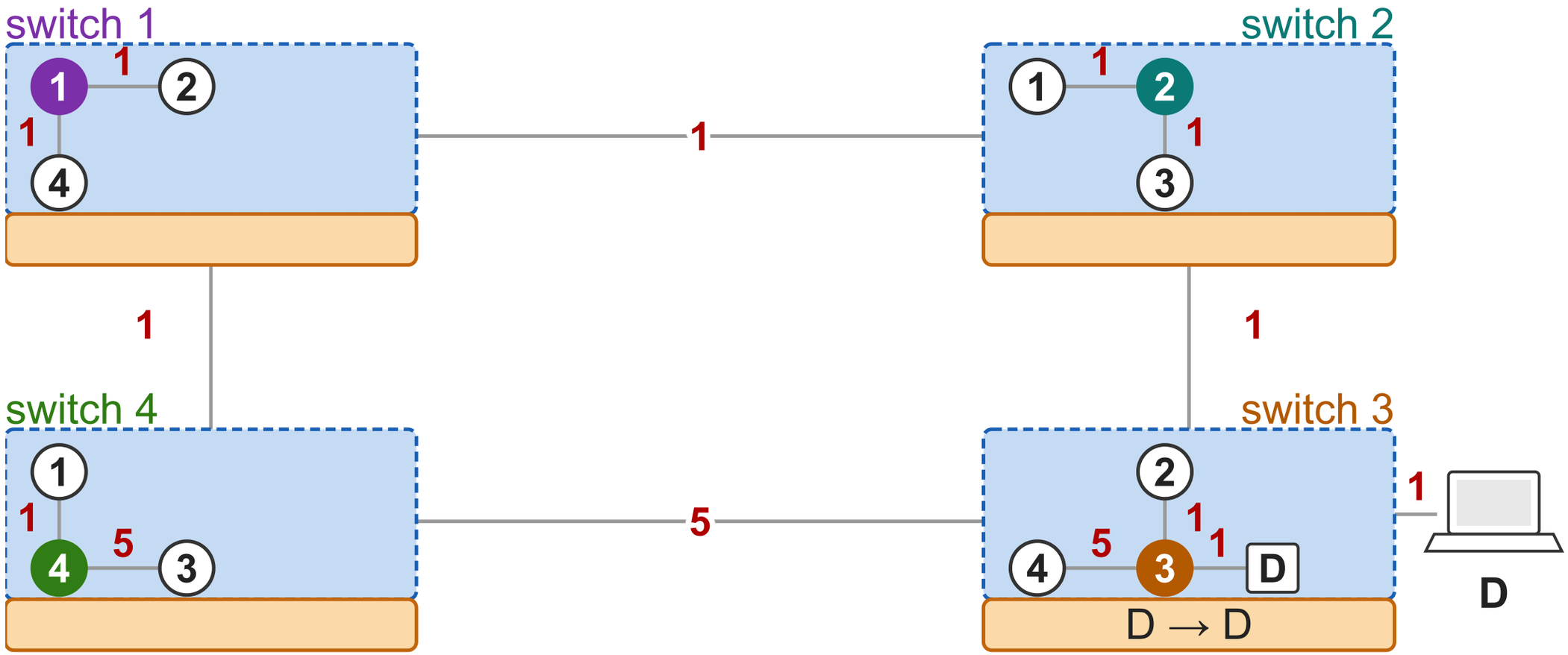
nothing saved yet | load | save | download as .qmd | Reset

Assume no link failure, no change to costs, and no messages are lost.

- `switch_id`: this switch's id, e.g. `1`
- `links[k]`: link k's `.neighbour` (a switch or a host) and `.cost`; k = 0, 1, 2, ...
- `send_msg(k, msg)`: sends `msg` over link k
- `fw_table[dest] = k`: packets for `dest` go out on link k

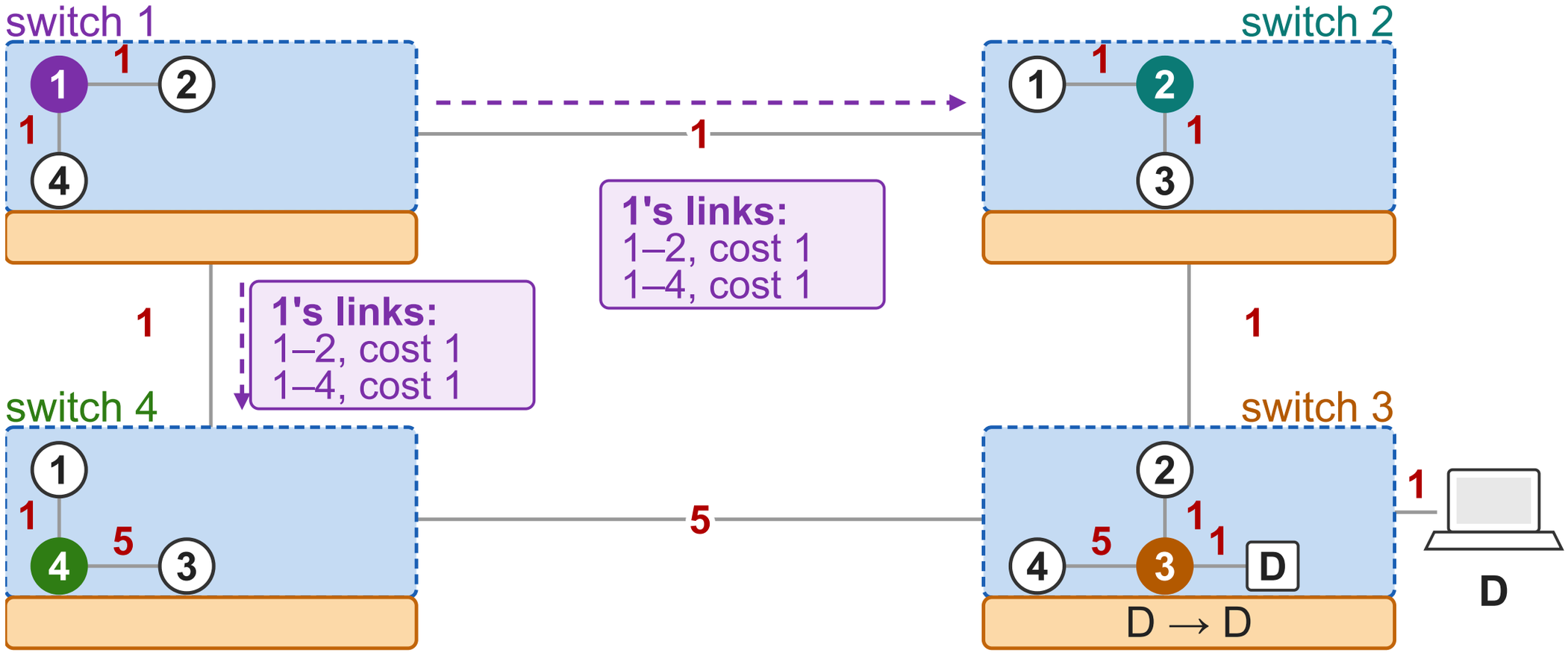
Every switch in the network learns about the whole topology

Each switch knows only its own links



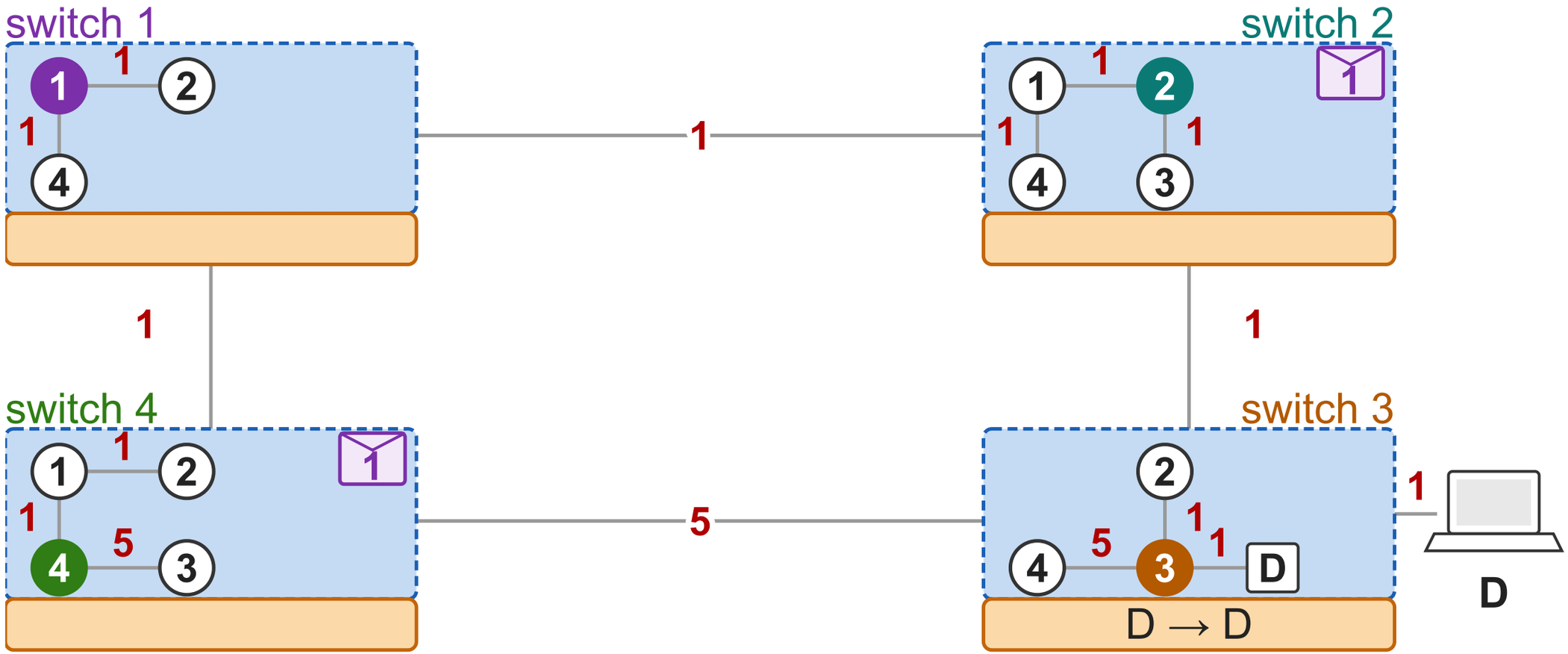
Every switch in the network learns about the whole topology

Each switch sends information about its links to its neighbors

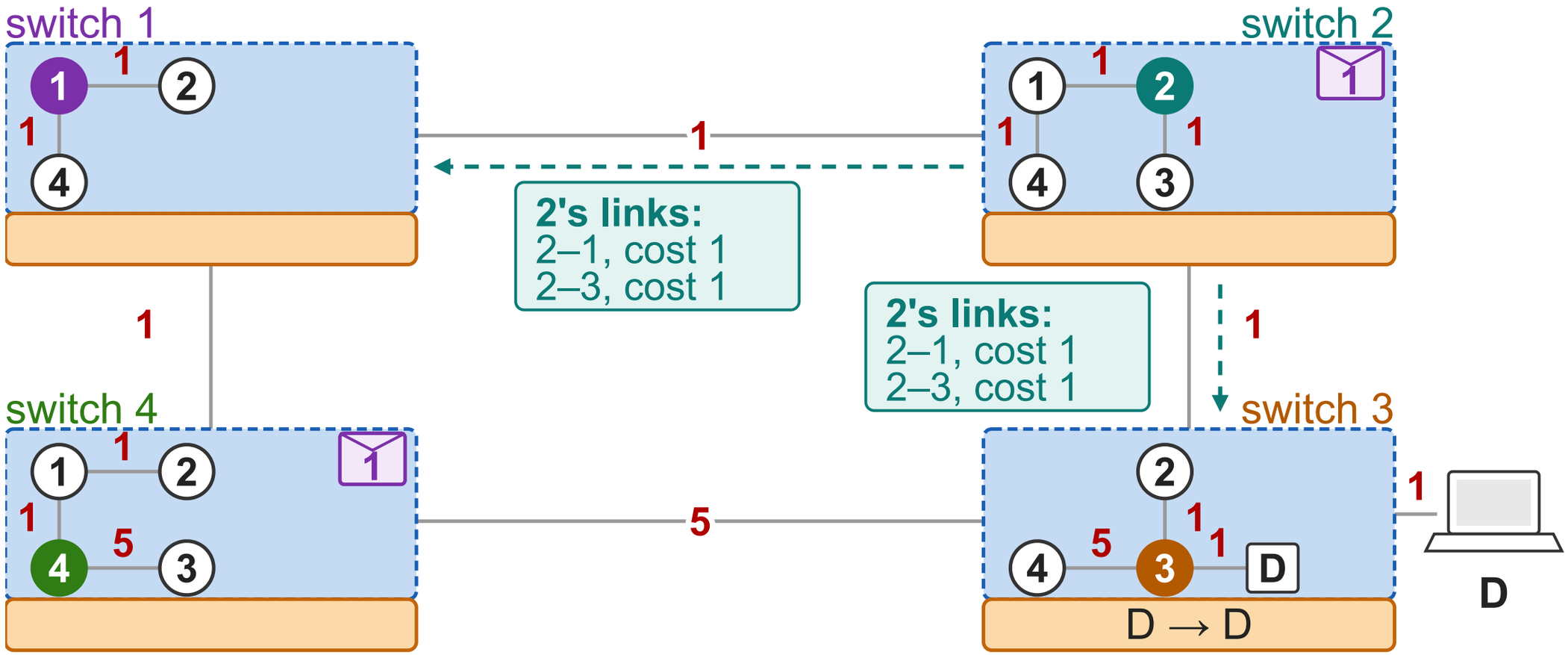


Every switch in the network learns about the whole topology

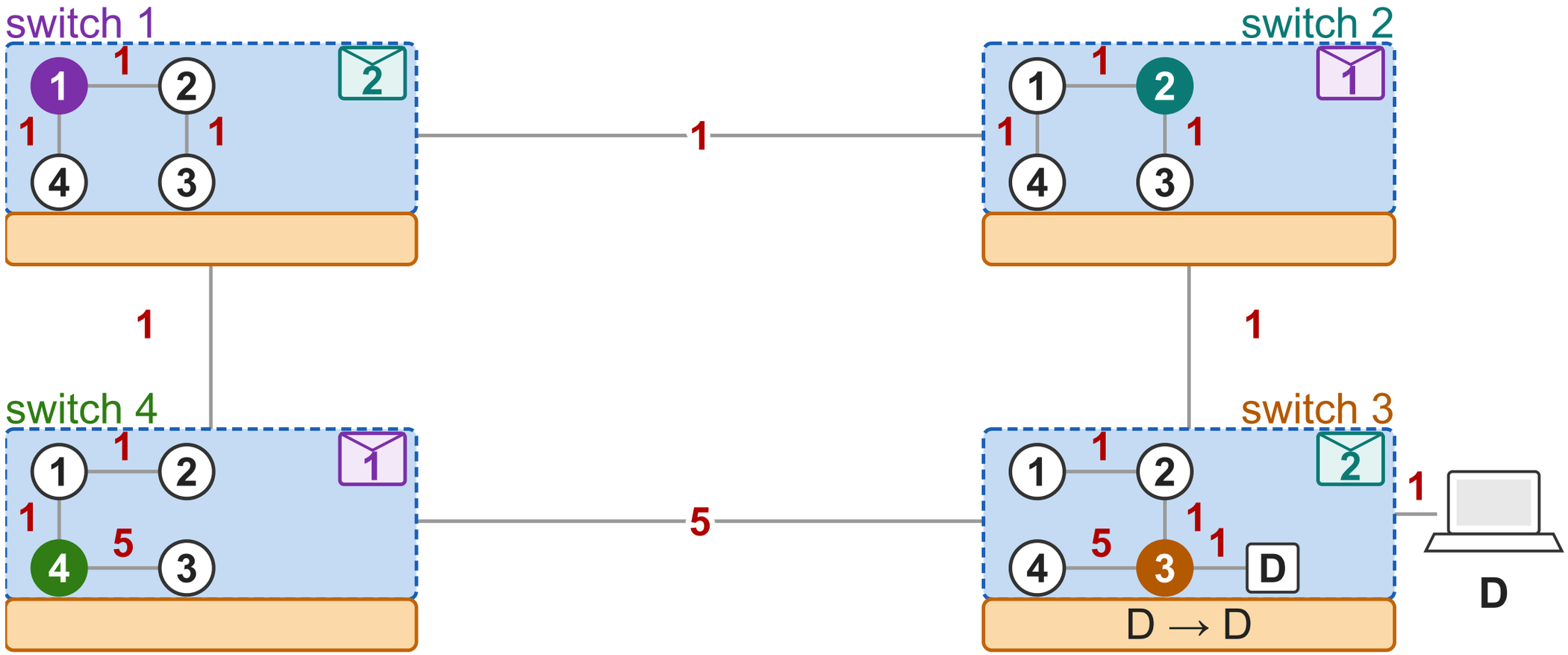
Each switch updates its graph based on the received messages from neighbors



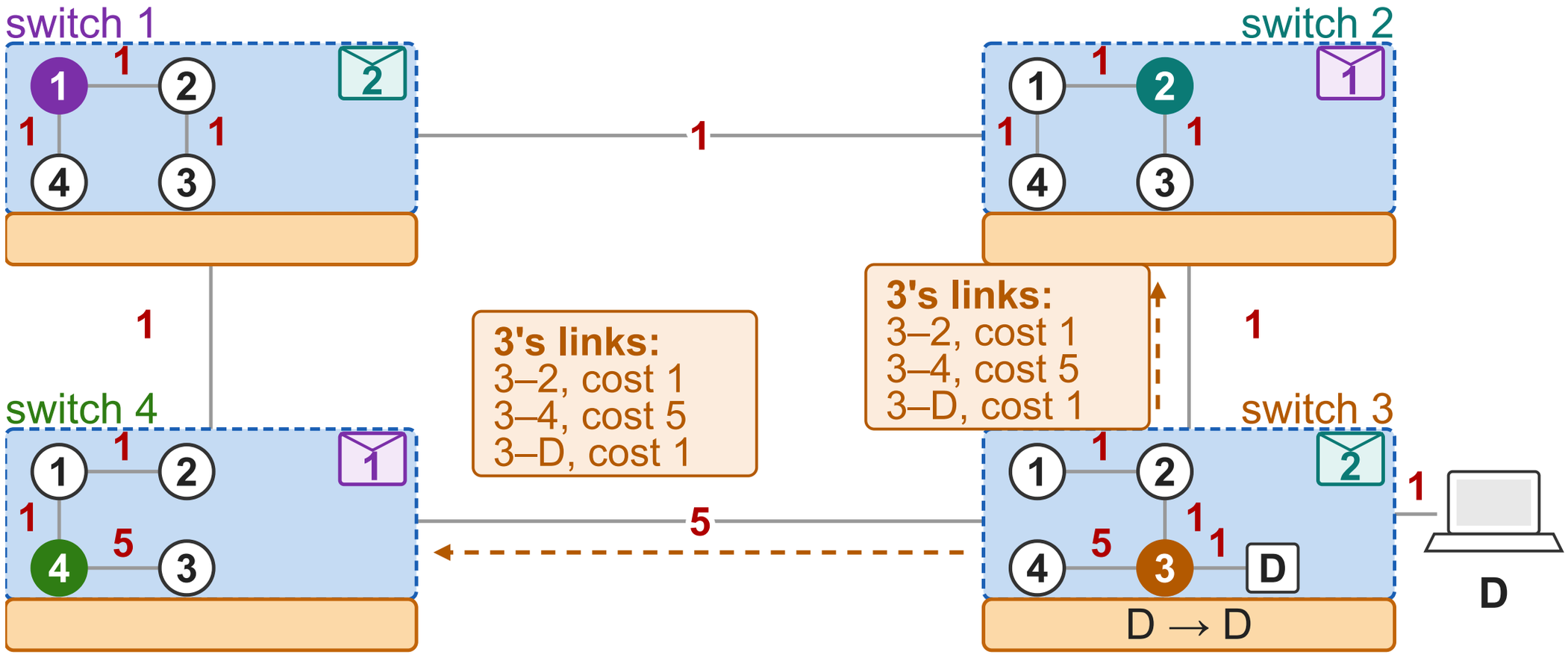
Every switch in the network learns about the whole topology



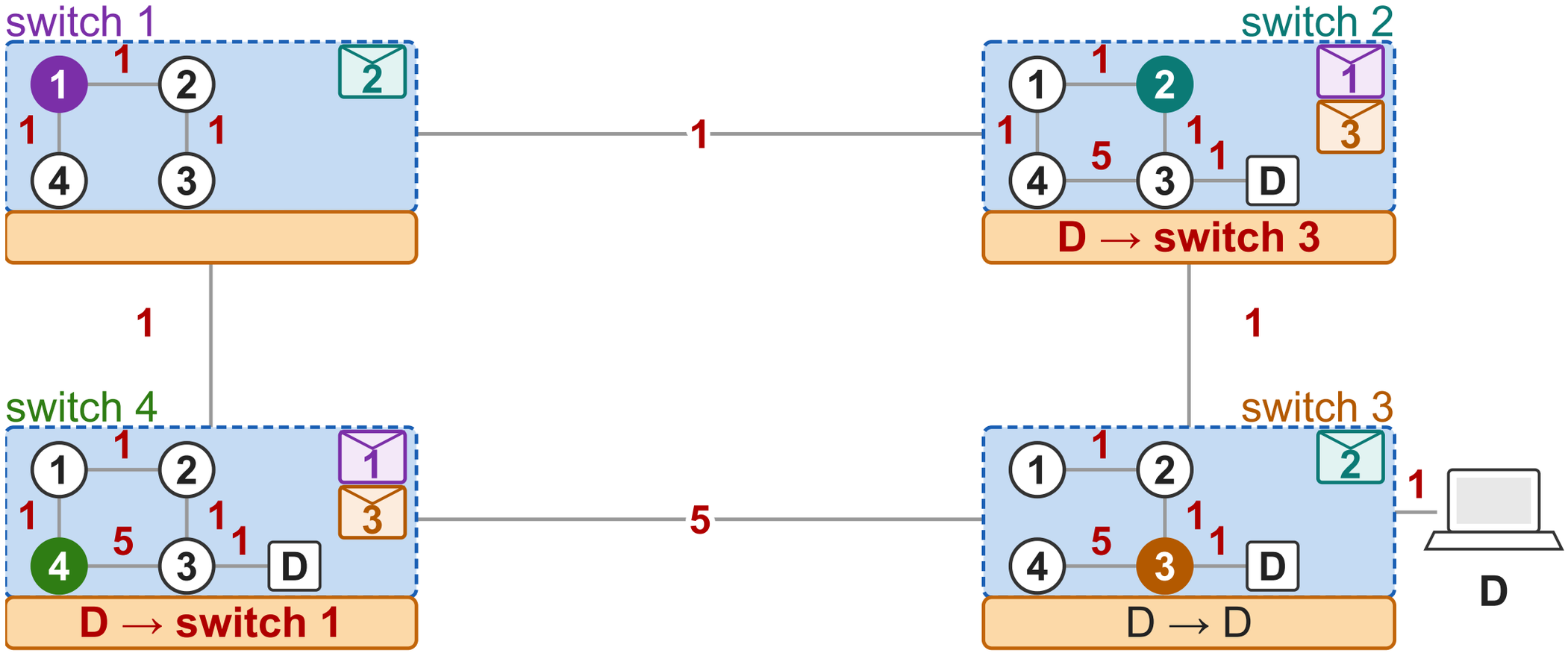
Every switch in the network learns about the whole topology



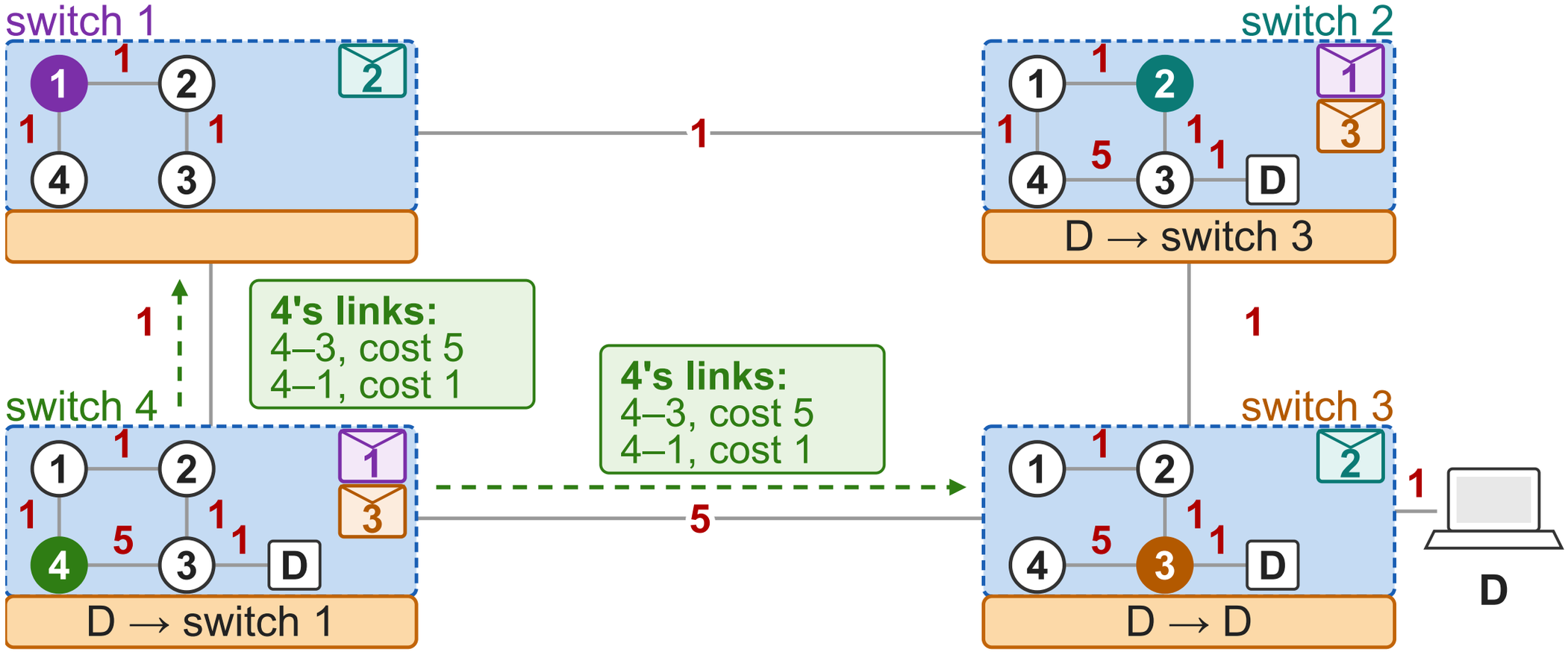
Every switch in the network learns about the whole topology



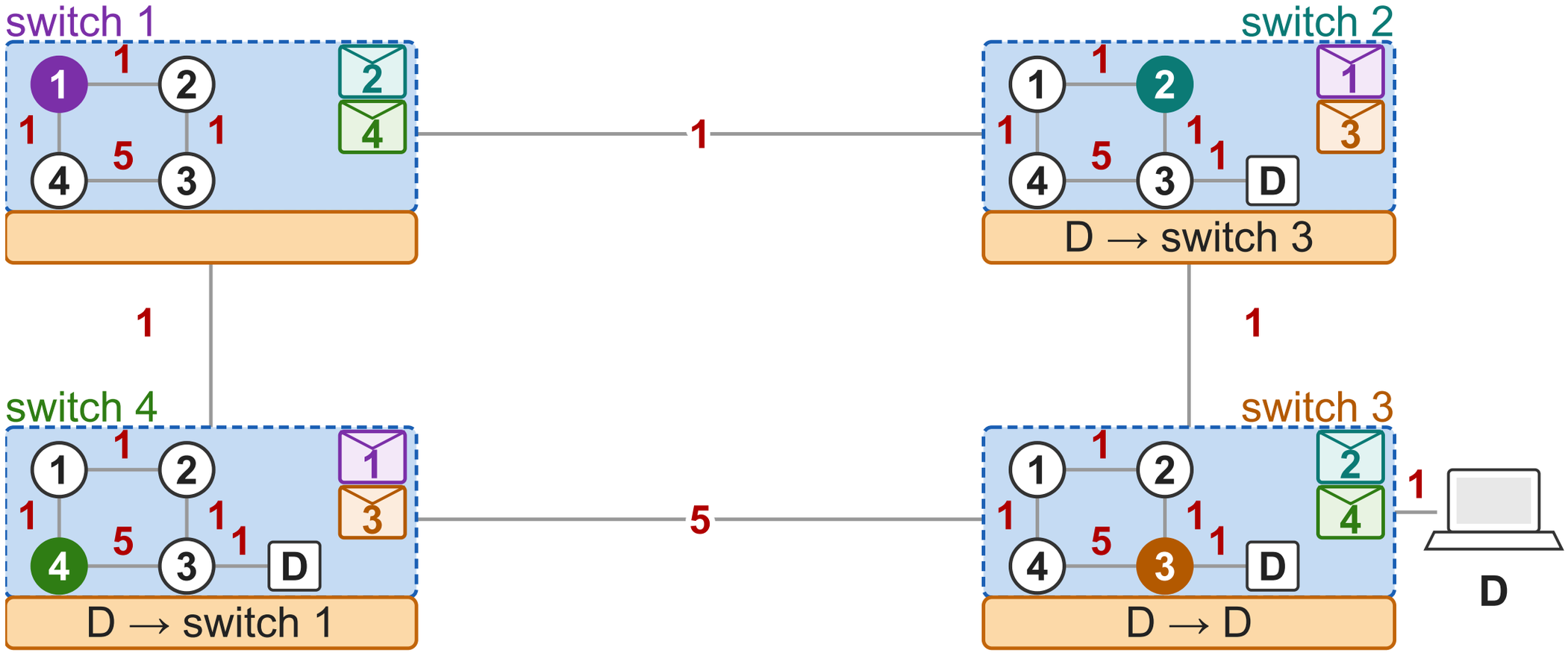
Every switch in the network learns about the whole topology



Every switch in the network learns about the whole topology

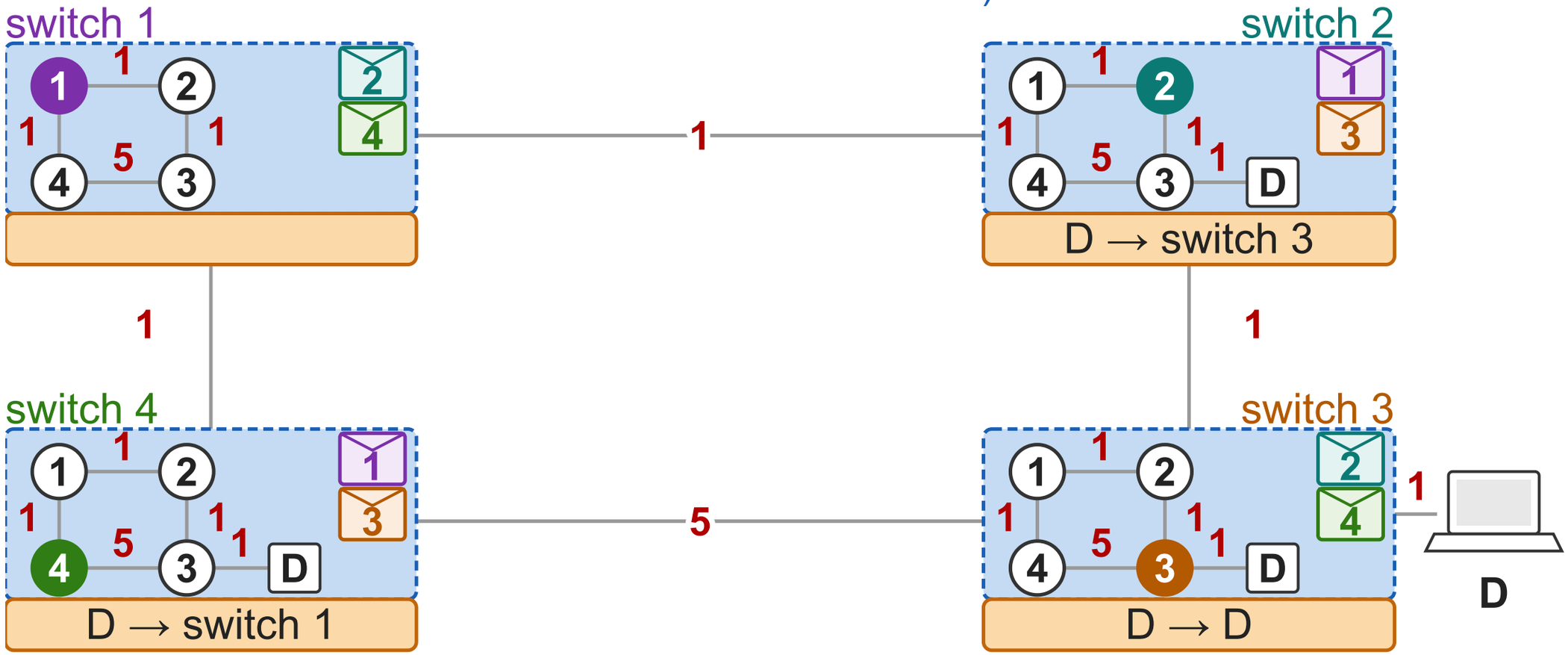


Every switch in the network learns about the whole topology



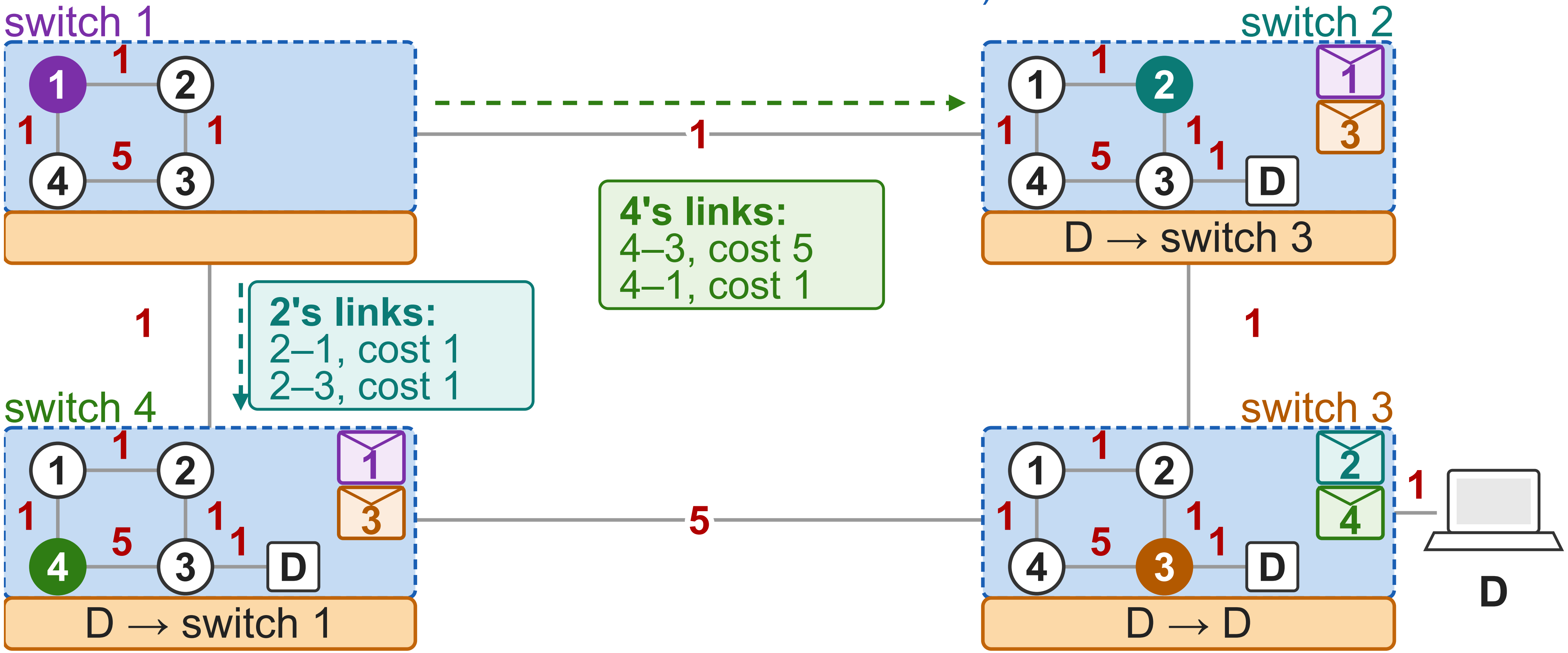
Every switch in the network learns about the whole topology

Each switch forwards the messages it got to its other neighbors (recipients drop duplicated messages and won't forward them)



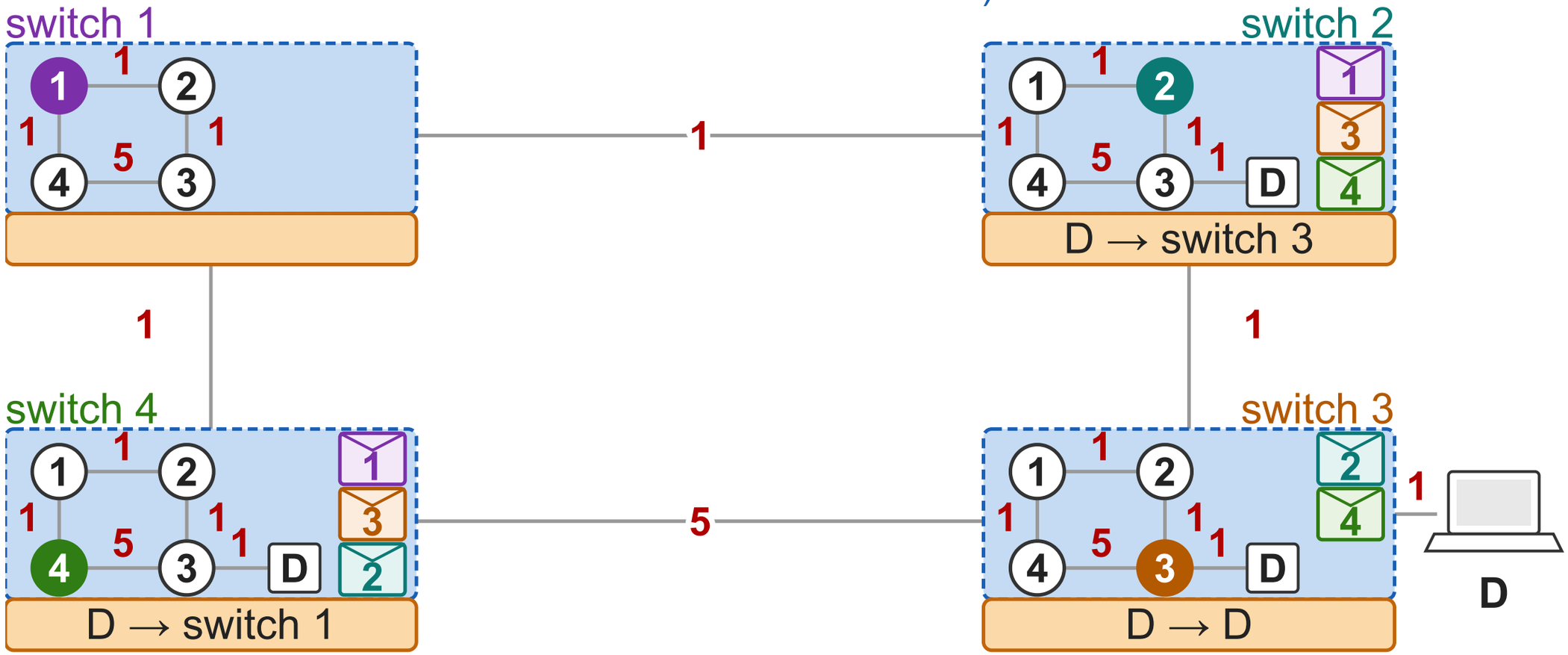
Every switch in the network learns about the whole topology

Each switch forwards the messages it got to its other neighbors (recipients drop duplicated messages and won't forward them)



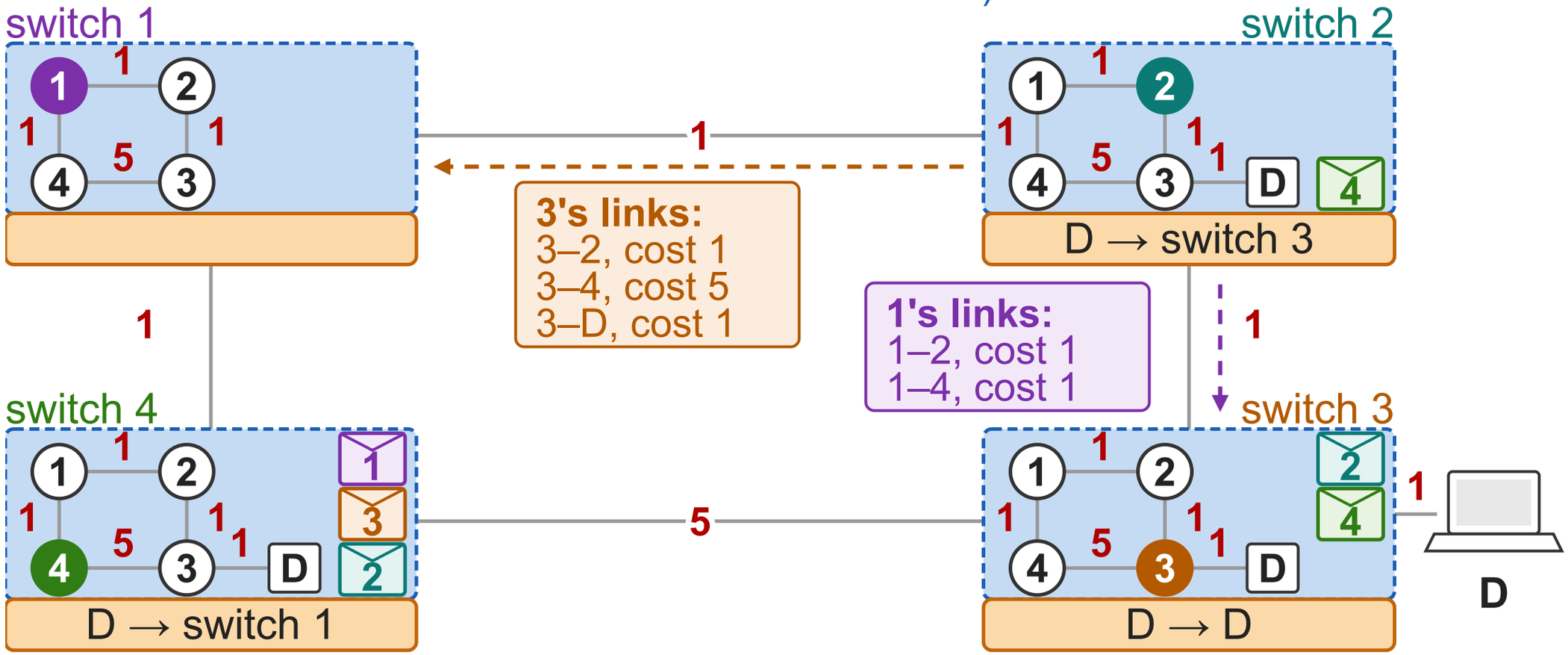
Every switch in the network learns about the whole topology

Each switch forwards the messages it got to its other neighbors (recipients drop duplicated messages and won't forward them)



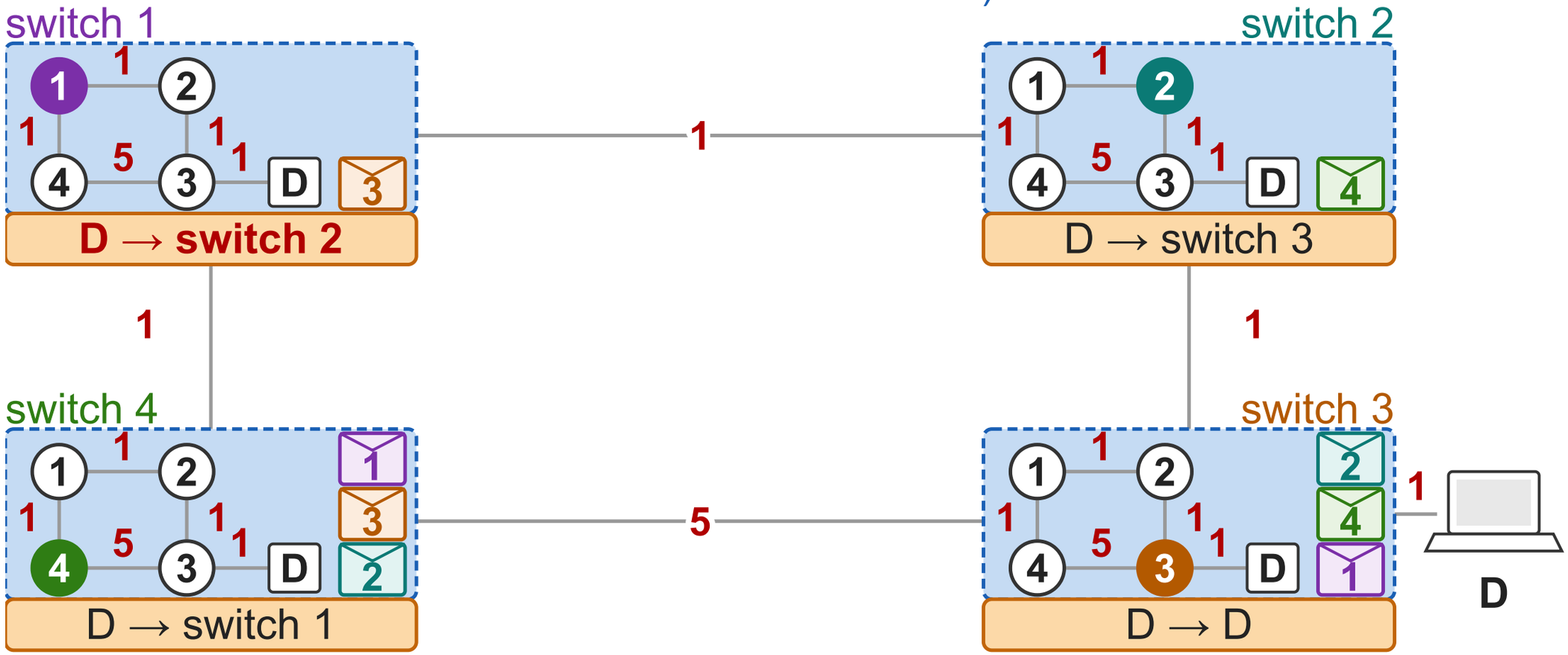
Every switch in the network learns about the whole topology

Each switch forwards the messages it got to its other neighbors (recipients drop duplicated messages and won't forward them)



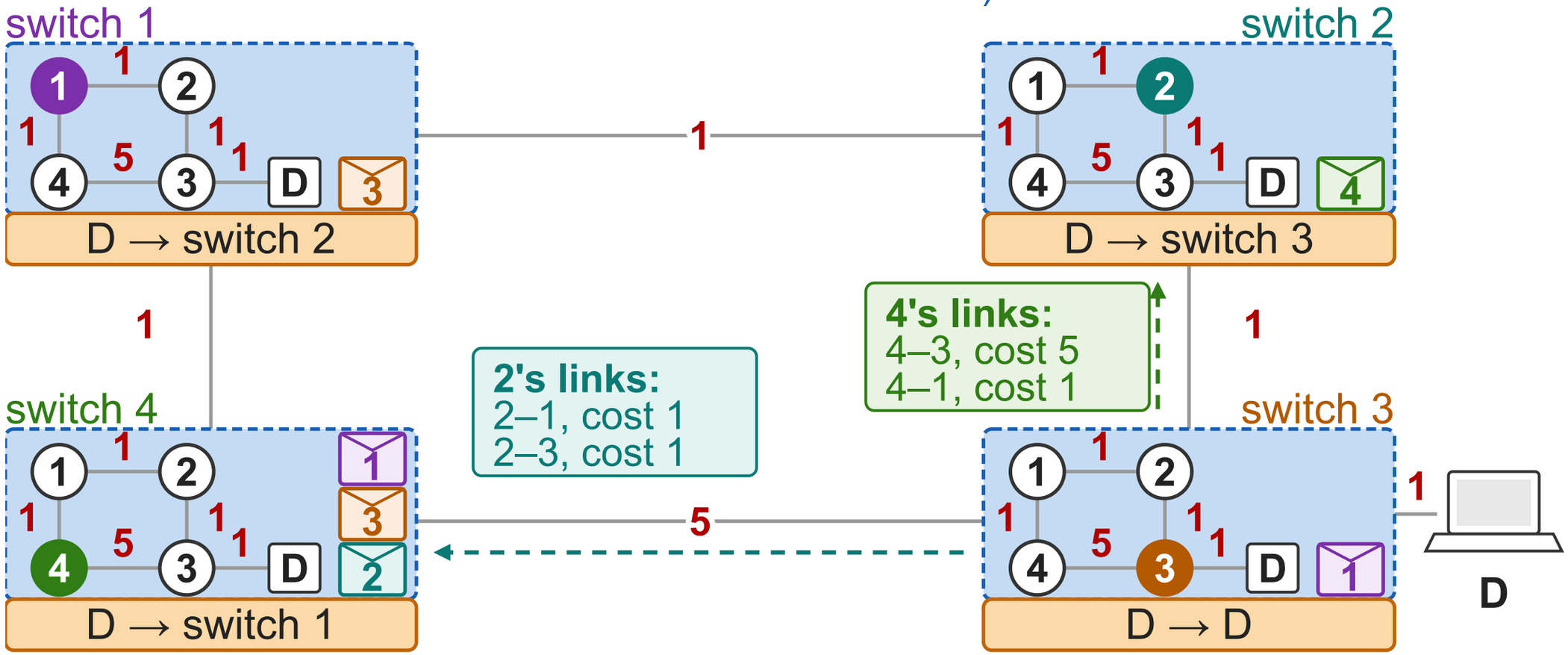
Every switch in the network learns about the whole topology

Each switch forwards the messages it got to its other neighbors (recipients drop duplicated messages and won't forward them)



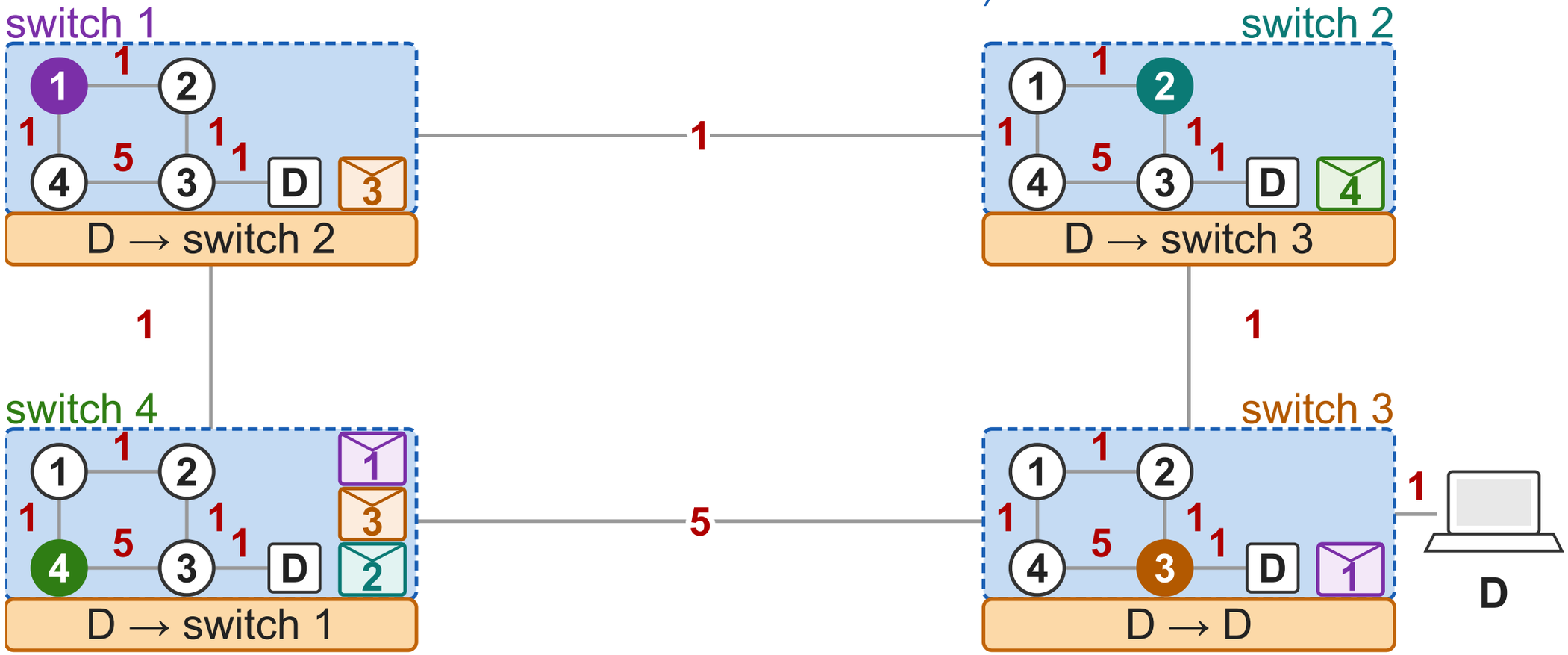
Every switch in the network learns about the whole topology

Each switch forwards the messages it got to its other neighbors (recipients drop duplicated messages and won't forward them)



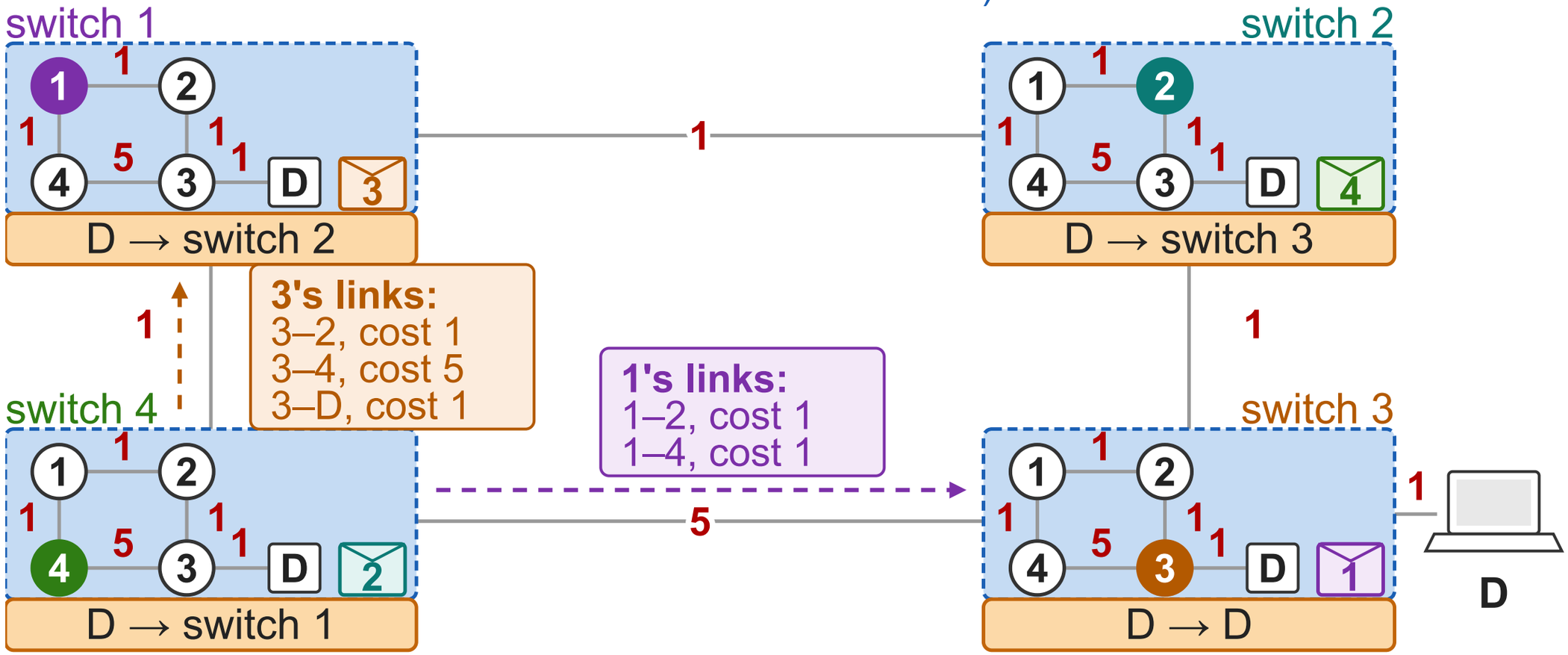
Every switch in the network learns about the whole topology

Each switch forwards the messages it got to its other neighbors (recipients drop duplicated messages and won't forward them)



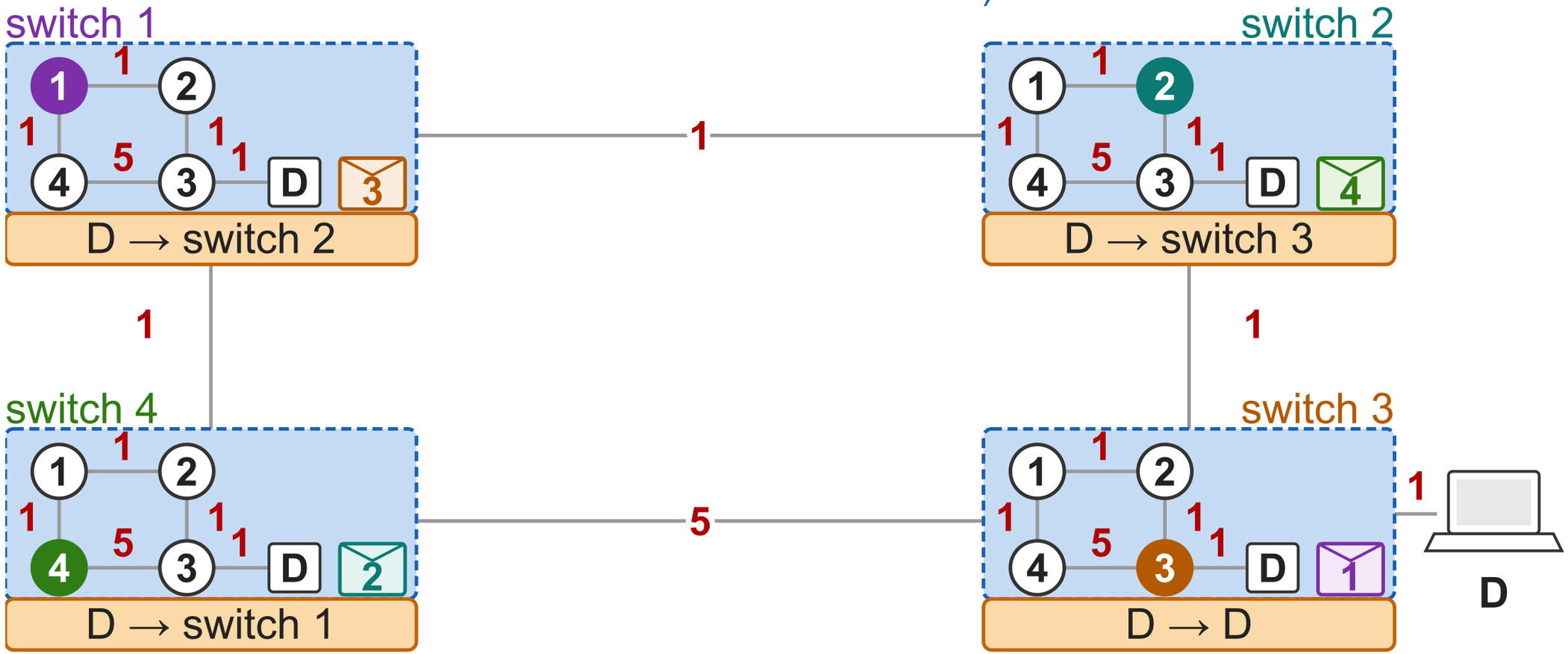
Every switch in the network learns about the whole topology

Each switch forwards the messages it got to its other neighbors (recipients drop duplicated messages and won't forward them)



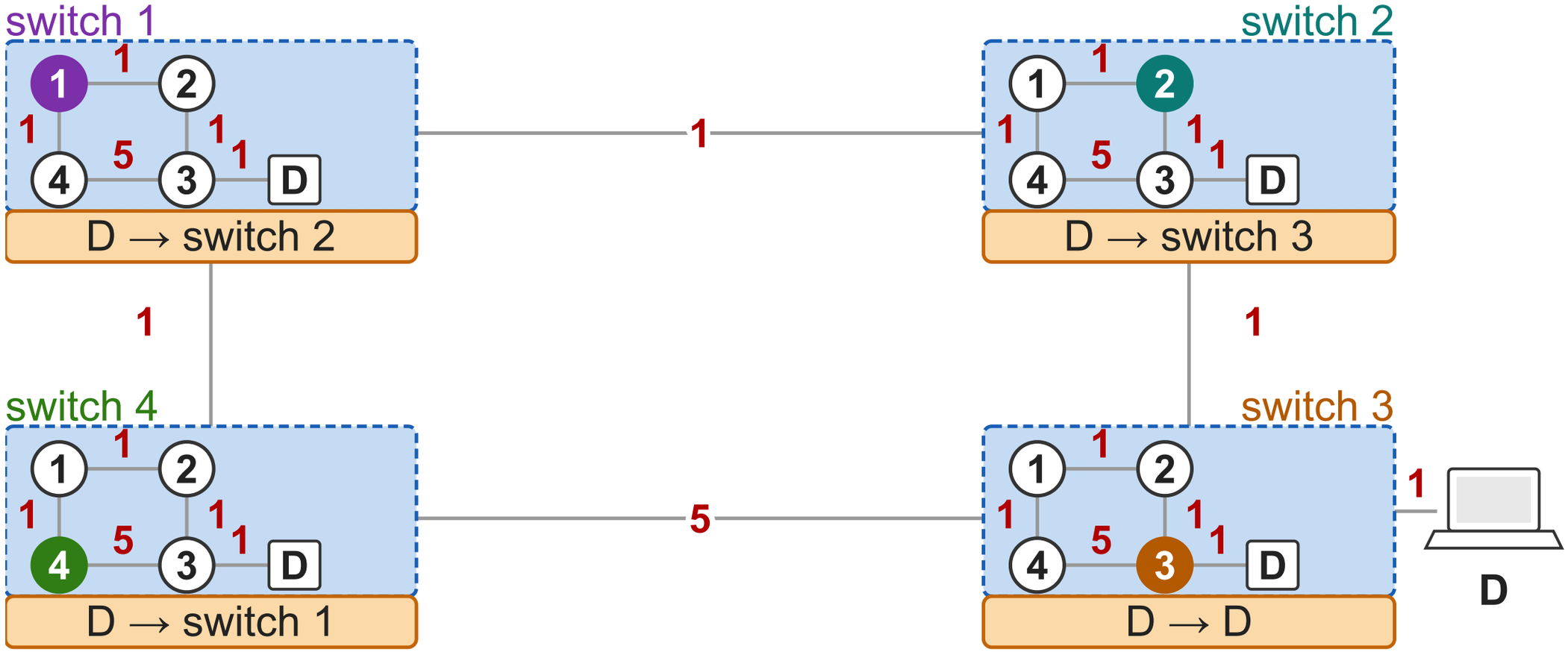
Every switch in the network learns about the whole topology

Each switch forwards the messages it got to its other neighbors (recipients drop duplicated messages and won't forward them)



Every switch in the network learns about the whole topology

Every switch knows the entire topology



Link state routing

This is called link state routing. It has two main stages:

Link state routing

This is called link state routing. It has two main stages:

- Information propagation:

Link state routing

This is called link state routing. It has two main stages:

- Information propagation:
  - Each switch sends a message with its ID, the switches and hosts it is directly connected to, and the cost of those links

Link state routing

This is called link state routing. It has two main stages:

- Information propagation:
  - Each switch sends a message with its ID, the switches and hosts it is directly connected to, and the cost of those links
  - Its neighbours forward it to their neighbours, and so on; messages go out periodically and when something changes

Link state routing

This is called link state routing. It has two main stages:

- Information propagation:
  - Each switch sends a message with its ID, the switches and hosts it is directly connected to, and the cost of those links
  - Its neighbours forward it to their neighbours, and so on; messages go out periodically and when something changes
  - Eventually, every switch has the same view of the network: its topology and link costs

Link state routing

This is called link state routing. It has two main stages:

- Information propagation:
  - Each switch sends a message with its ID, the switches and hosts it is directly connected to, and the cost of those links
  - Its neighbours forward it to their neighbours, and so on; messages go out periodically and when something changes
  - Eventually, every switch has the same view of the network: its topology and link costs
- Path computation:

Link state routing

This is called link state routing. It has two main stages:

- Information propagation:
  - Each switch sends a message with its ID, the switches and hosts it is directly connected to, and the cost of those links
  - Its neighbours forward it to their neighbours, and so on; messages go out periodically and when something changes
  - Eventually, every switch has the same view of the network: its topology and link costs
- Path computation:
  - Each switch computes the least-cost paths on its own copy of the graph (e.g., with Dijkstra's algorithm)

Link state routing

This is called link state routing. It has two main stages:

- Information propagation:
  - Each switch sends a message with its ID, the switches and hosts it is directly connected to, and the cost of those links
  - Its neighbours forward it to their neighbours, and so on; messages go out periodically and when something changes
  - Eventually, every switch has the same view of the network: its topology and link costs
- Path computation:
  - Each switch computes the least-cost paths on its own copy of the graph (e.g., with Dijkstra's algorithm)
  - Every switch has the same graph, so the paths they pick are consistent with each other

Link state routing

This is called link state routing. It has two main stages:

- Information propagation:
  - Each switch sends a message with its ID, the switches and hosts it is directly connected to, and the cost of those links
  - Its neighbours forward it to their neighbours, and so on; messages go out periodically and when something changes
  - Eventually, every switch has the same view of the network: its topology and link costs
- Path computation:
  - Each switch computes the least-cost paths on its own copy of the graph (e.g., with Dijkstra's algorithm)
  - Every switch has the same graph, so the paths they pick are consistent with each other
- In reality, switches are not in lock step. Each forwards a message when it arrives, and recomputes its paths when its graph changes

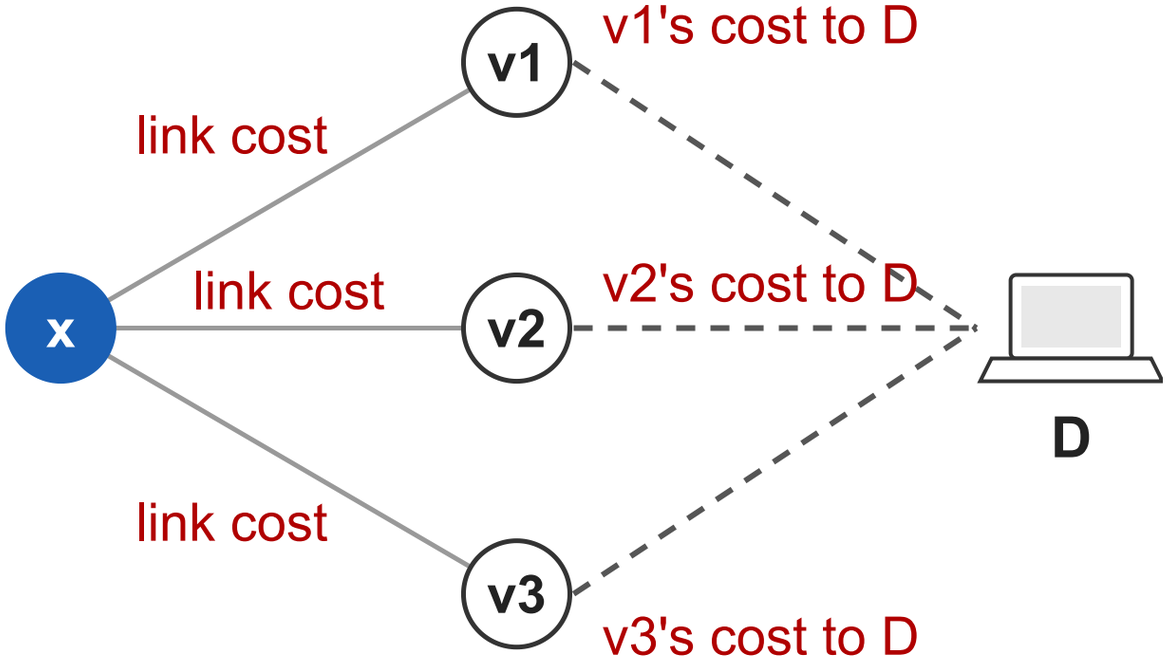
Link state routing

This is called link state routing. It has two main stages:

- Information propagation:
  - Each switch sends a message with its ID, the switches and hosts it is directly connected to, and the cost of those links
  - Its neighbours forward it to their neighbours, and so on; messages go out periodically and when something changes
  - Eventually, every switch has the same view of the network: its topology and link costs
- Path computation:
  - Each switch computes the least-cost paths on its own copy of the graph (e.g., with Dijkstra's algorithm)
  - Every switch has the same graph, so the paths they pick are consistent with each other
- In reality, switches are not in lock step. Each forwards a message when it arrives, and recomputes its paths when its graph changes
- Example protocols: OSPF and IS-IS

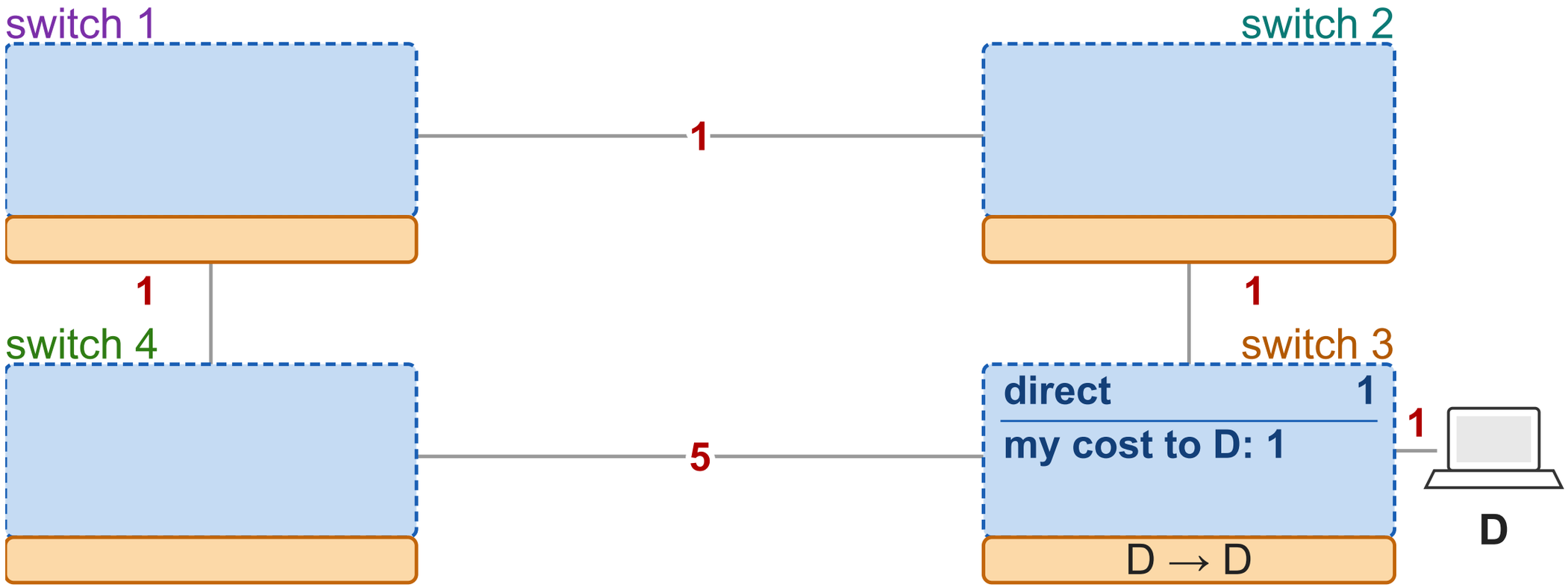
Do we need every switch to know the whole topology?

- The least-cost path from switch x to D starts with a link to one of x's neighbors
- So x only needs to know, for each neighbor v: the cost of its link to v, and the cost of v's path to D
- x's cost to D = the minimum, over its neighbours v, of (link cost to v + v's cost to D)



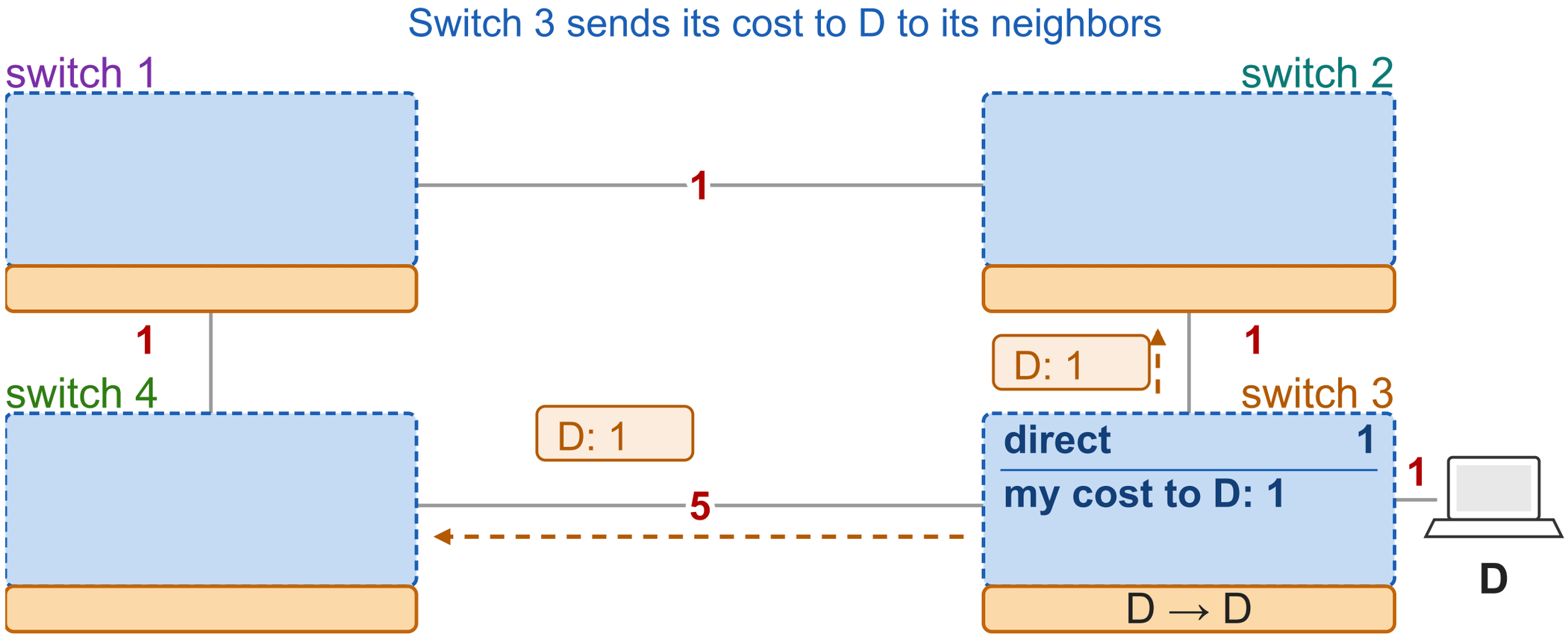
Each switch only knows the paths reachable through its neighbors

Shown: only the information related to the costs to D. Messages and switch state include similar information about all destinations.



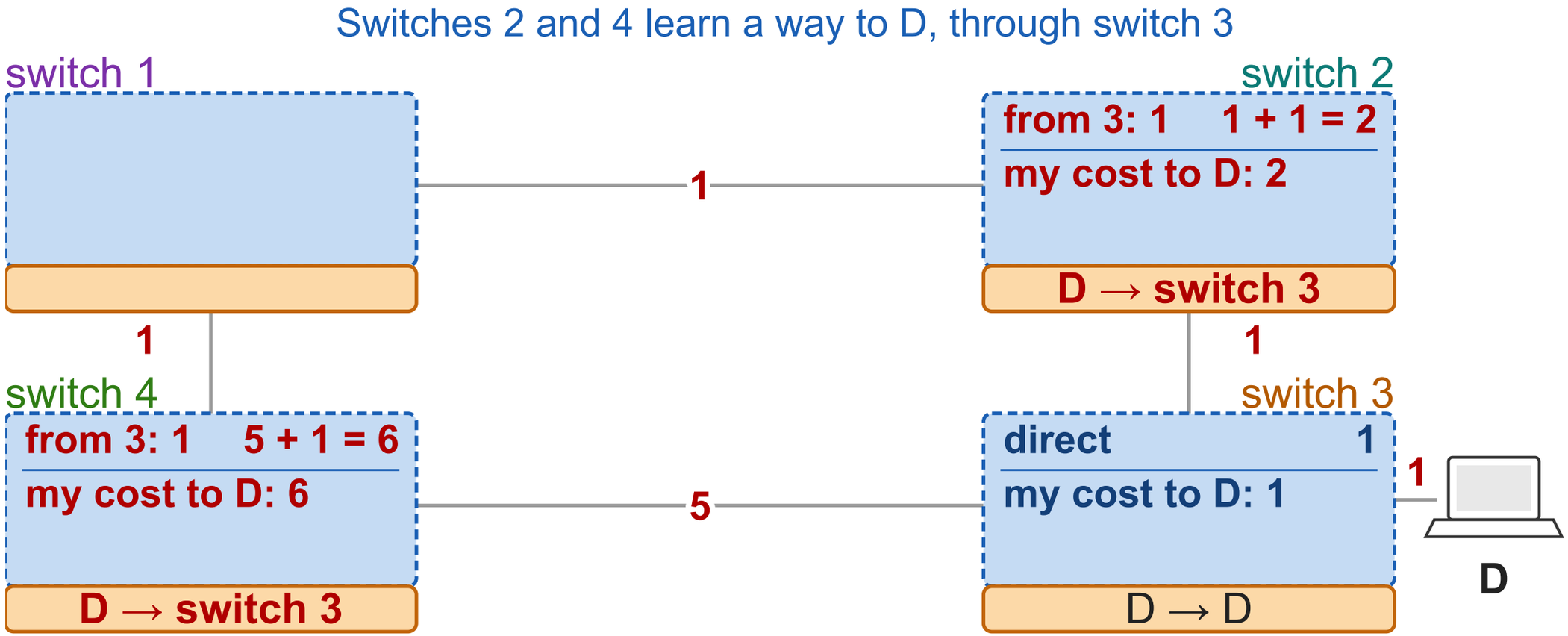
Each switch only knows the paths reachable through its neighbors

Shown: only the information related to the costs to D. Messages and switch state include similar information about all destinations.



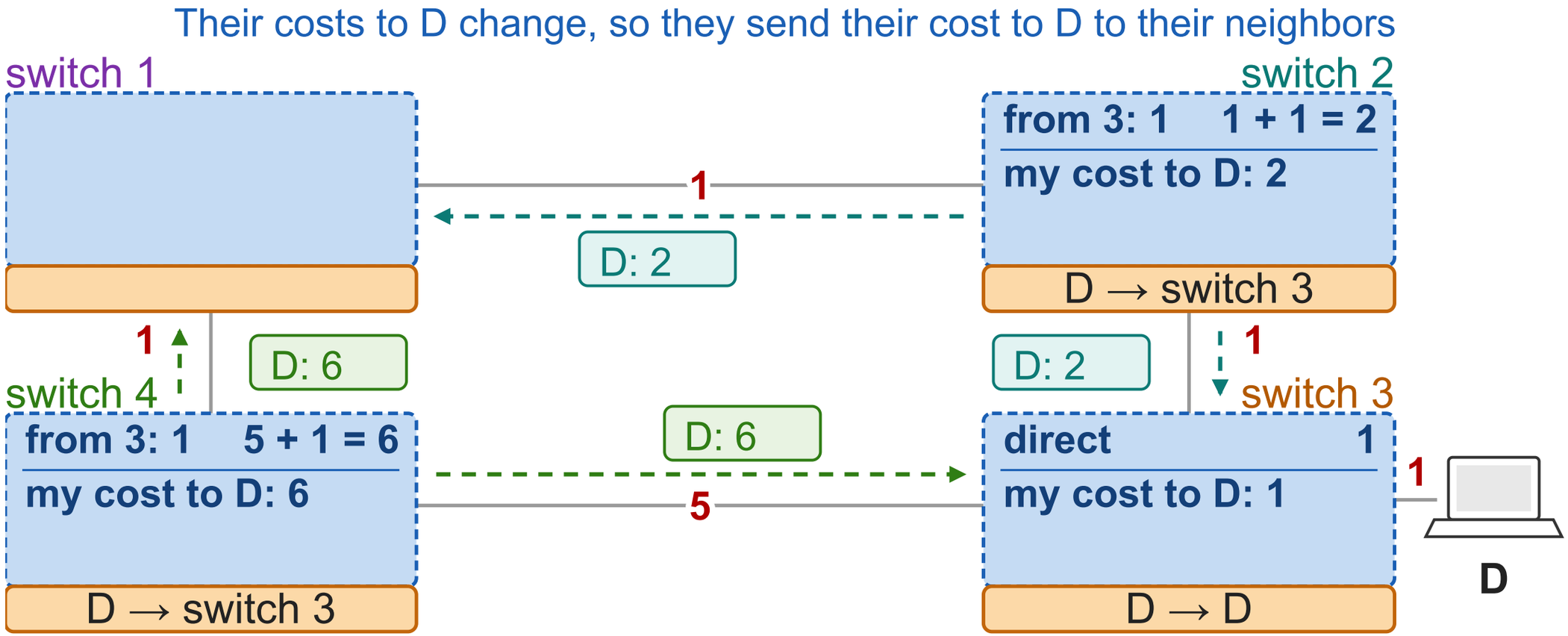
Each switch only knows the paths reachable through its neighbors

Shown: only the information related to the costs to D. Messages and switch state include similar information about all destinations.



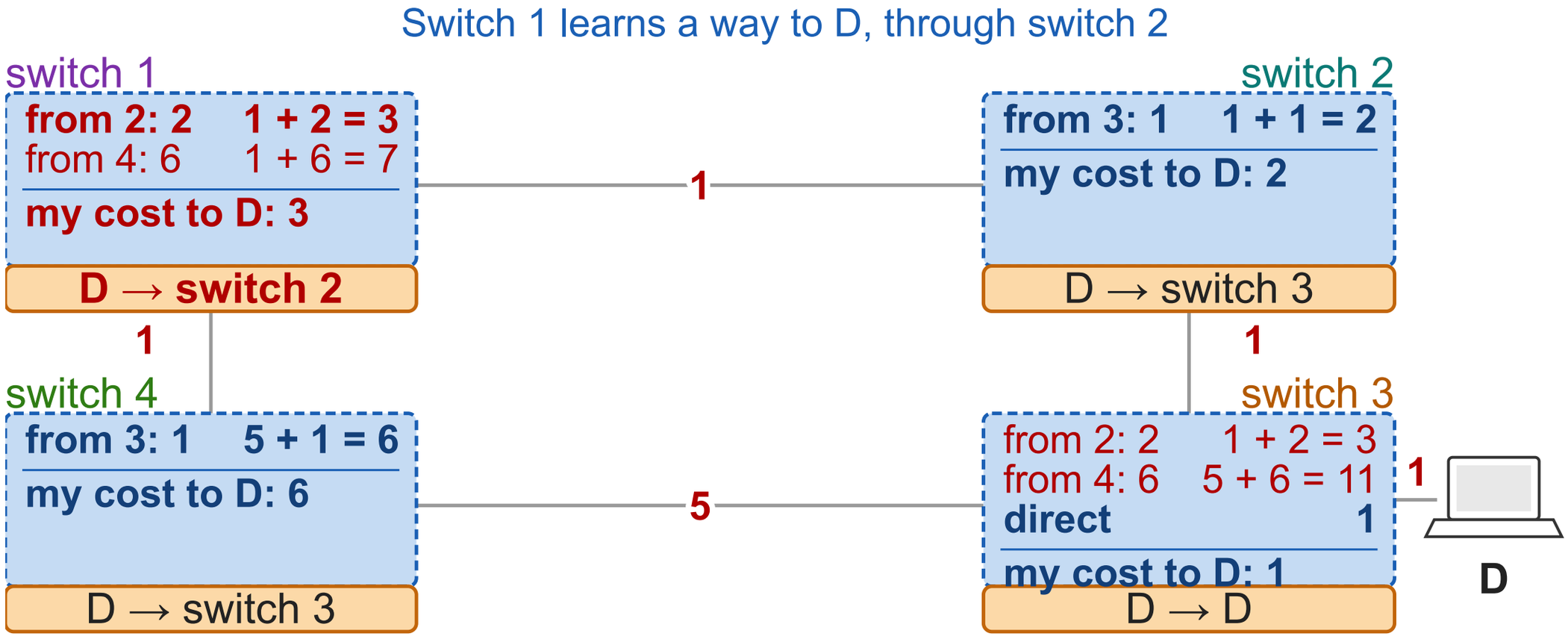
Each switch only knows the paths reachable through its neighbors

Shown: only the information related to the costs to D. Messages and switch state include similar information about all destinations.



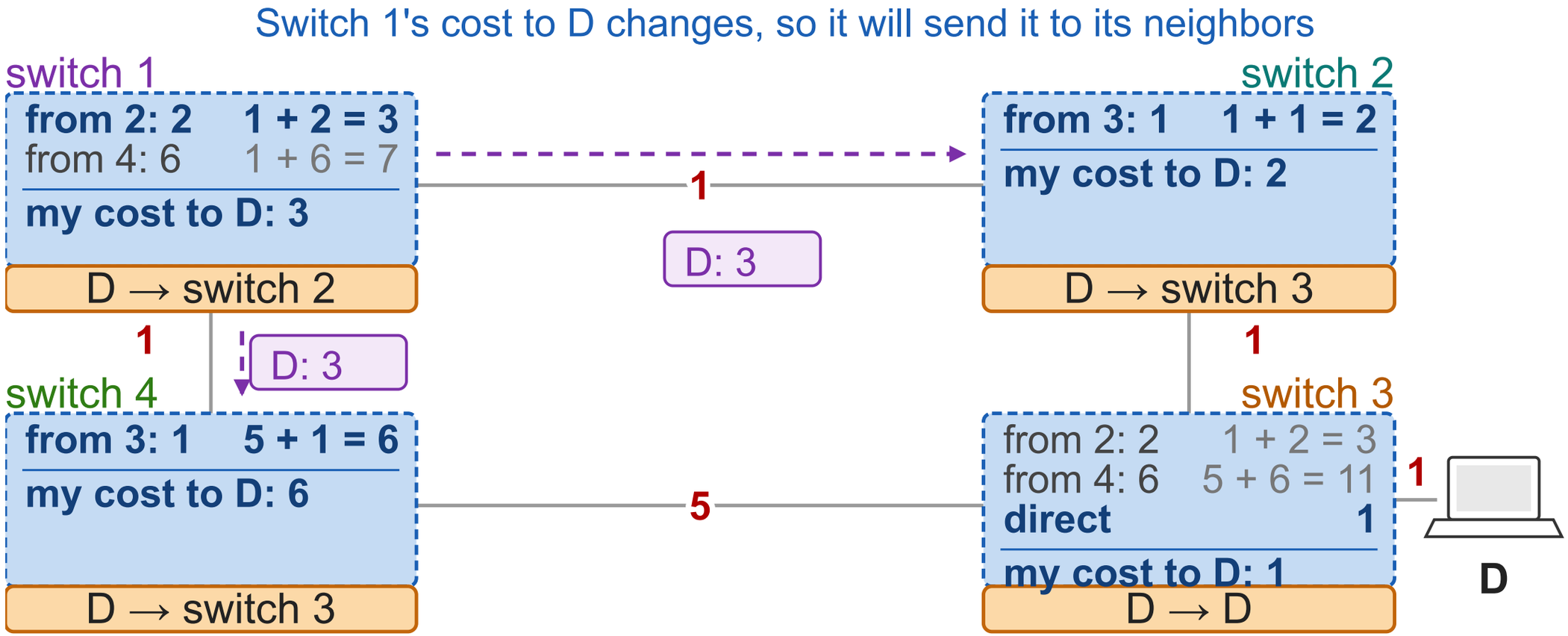
Each switch only knows the paths reachable through its neighbors

Shown: only the information related to the costs to D. Messages and switch state include similar information about all destinations.



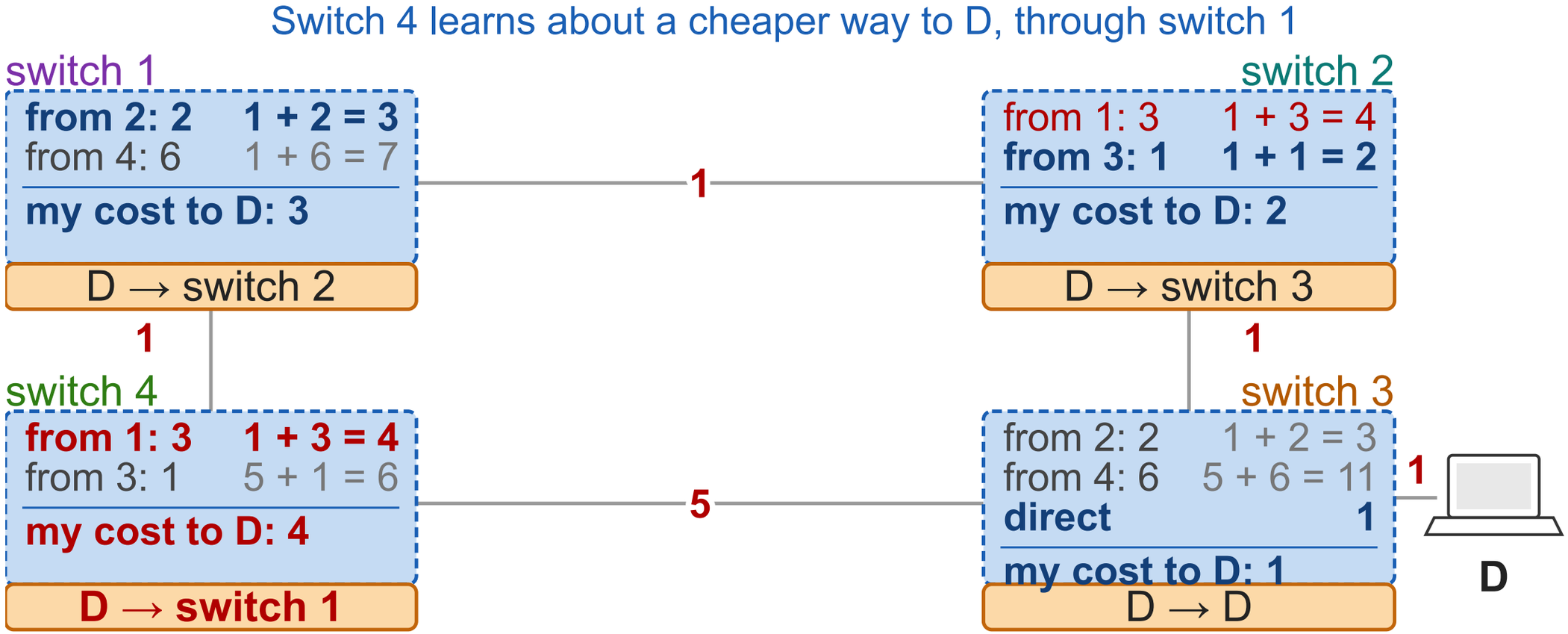
Each switch only knows the paths reachable through its neighbors

Shown: only the information related to the costs to D. Messages and switch state include similar information about all destinations.



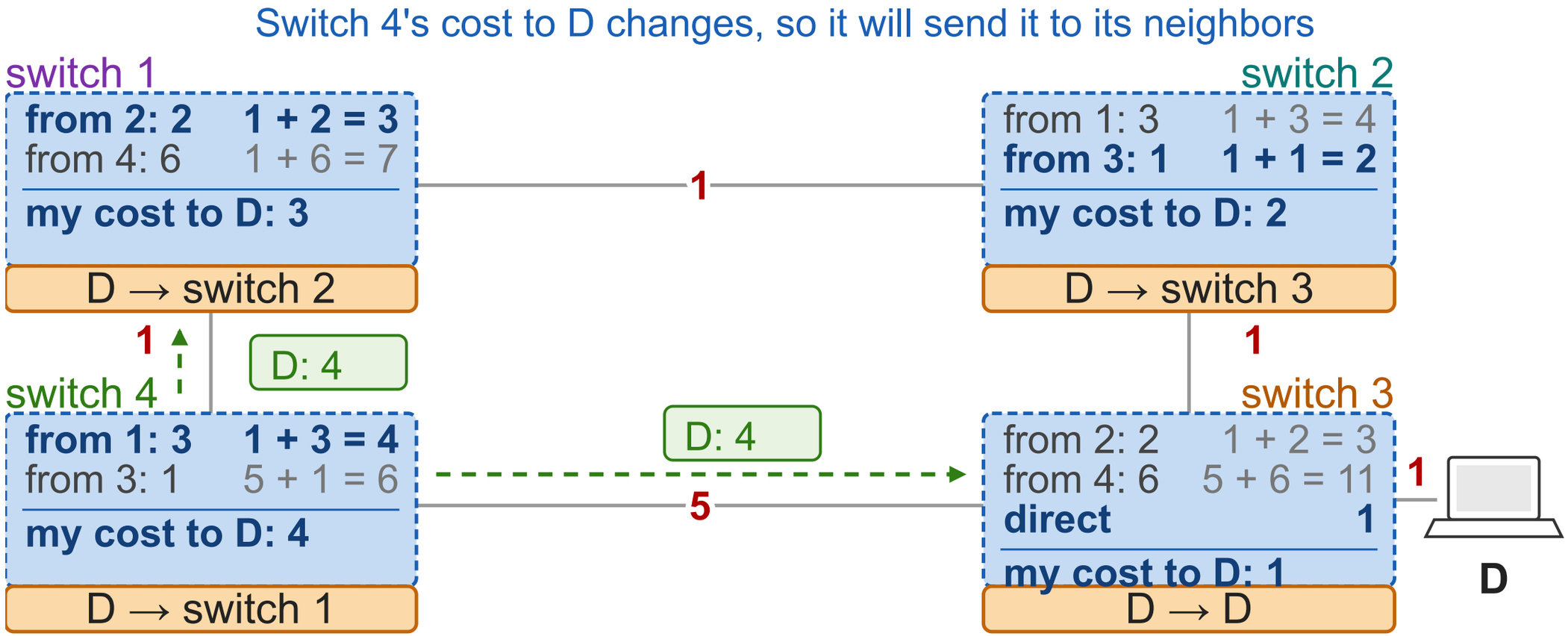
Each switch only knows the paths reachable through its neighbors

Shown: only the information related to the costs to D. Messages and switch state include similar information about all destinations.



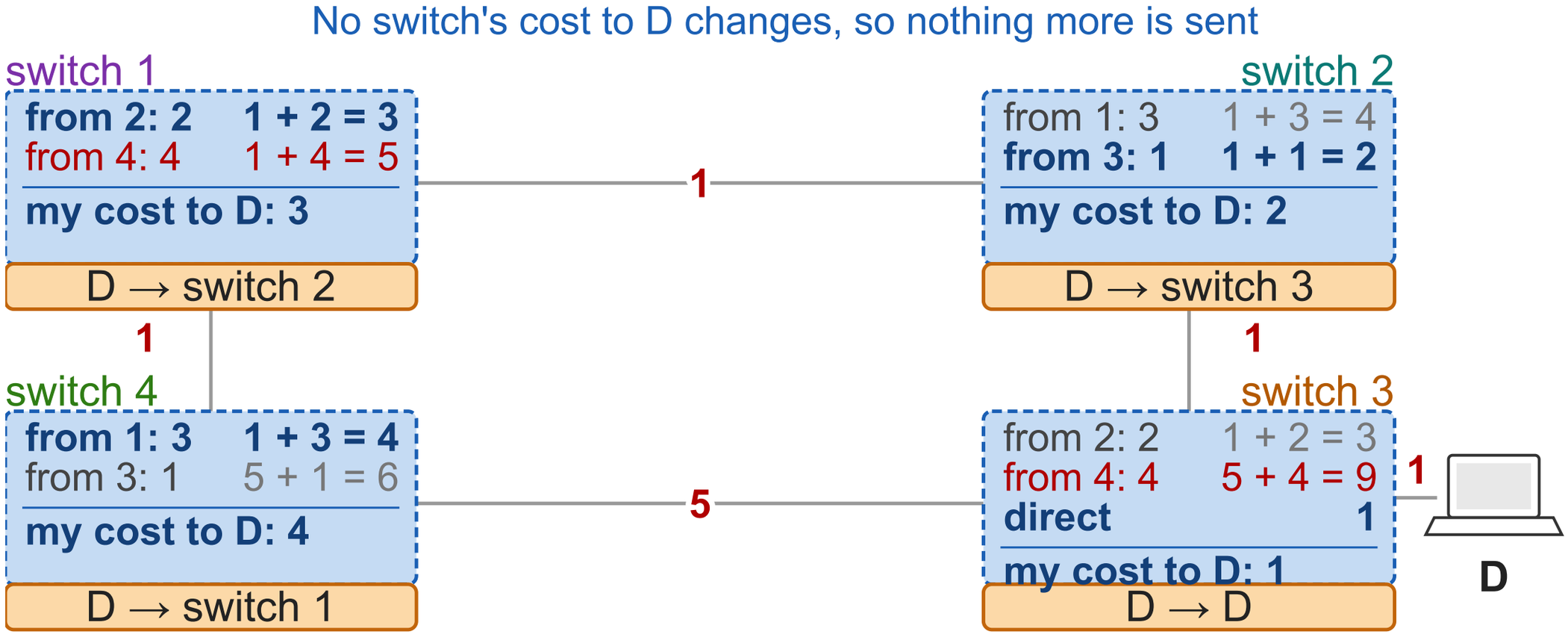
Each switch only knows the paths reachable through its neighbors

Shown: only the information related to the costs to D. Messages and switch state include similar information about all destinations.



Each switch only knows the paths reachable through its neighbors

Shown: only the information related to the costs to D. Messages and switch state include similar information about all destinations.



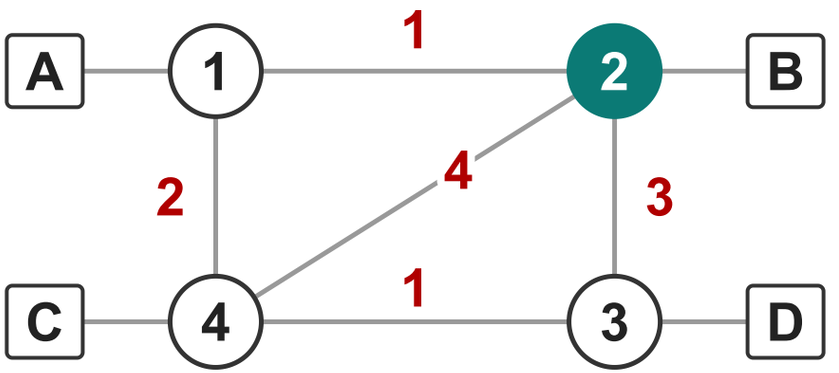
Distance vector routing

This is called distance vector routing.

Distance vector routing

This is called distance vector routing.

- A switch's distance vector contains its cost to every destination



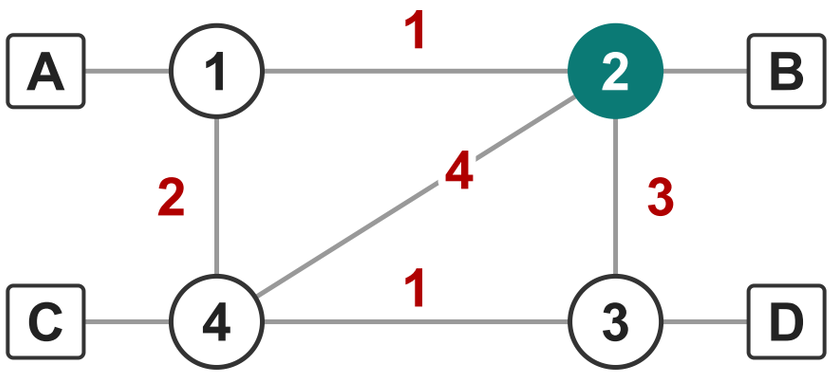
Switch 2's distance vector (host links cost 1):

destination	A	B	C	D
cost	2	1	4	4

Distance vector routing

This is called distance vector routing.

- A switch's distance vector contains its cost to every destination
- The distance vector is included in the message it sends to its neighbors



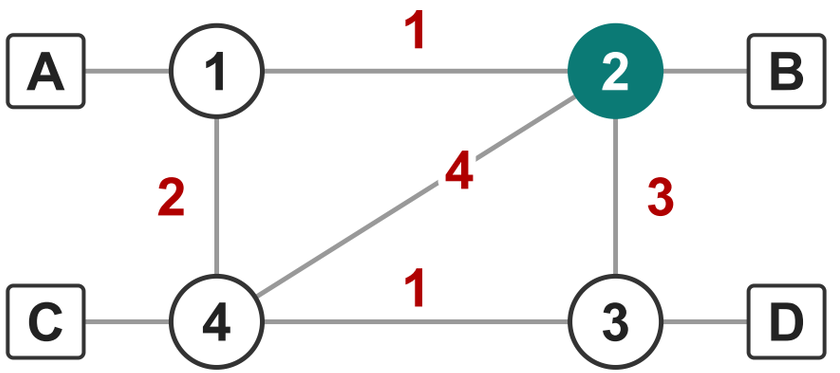
Switch 2's distance vector (host links cost 1):

destination	A	B	C	D
cost	2	1	4	4

Distance vector routing

This is called distance vector routing.

- A switch's distance vector contains its cost to every destination
- The distance vector is included in the message it sends to its neighbors
- Each switch x keeps the cost of each of its links, and the last distance vector each neighbor sent it



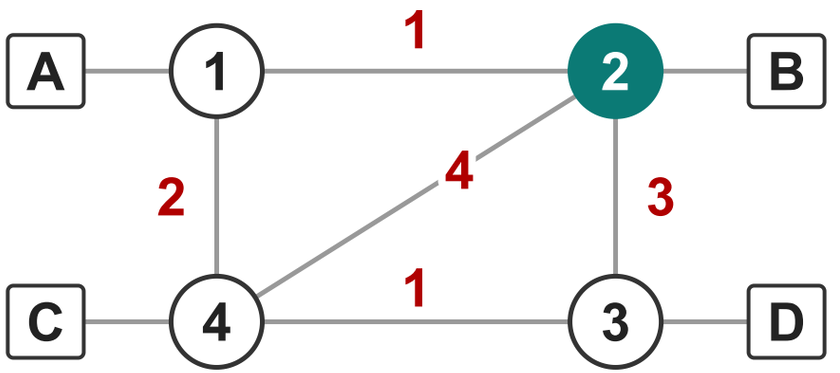
Switch 2's distance vector (host links cost 1):

destination	A	B	C	D
cost	2	1	4	4

Distance vector routing

This is called distance vector routing.

- A switch's distance vector contains its cost to every destination
- The distance vector is included in the message it sends to its neighbors
- Each switch x keeps the cost of each of its links, and the last distance vector each neighbor sent it
- x's cost to each destination y = the minimum, over its neighbors v, of (link cost to v + v's cost to y, the Bellman-Ford equation)



Switch 2's distance vector (host links cost 1):

destination	A	B	C	D
cost	2	1	4	4

Distance vector routing (continued)

Distance vector routing (continued)

- When a switch's distance vector changes, it sends the updated one to its neighbors

Distance vector routing (continued)

- When a switch's distance vector changes, it sends the updated one to its neighbors
- If there are no changes, it will not send any messages to its neighbors

Distance vector routing (continued)

- When a switch's distance vector changes, it sends the updated one to its neighbors
- If there are no changes, it will not send any messages to its neighbors
  - Note the difference from link state

Distance vector routing (continued)

- When a switch's distance vector changes, it sends the updated one to its neighbors
- If there are no changes, it will not send any messages to its neighbors
  - Note the difference from link state
- In reality, switches are not in lock step

Distance vector routing (continued)

- When a switch's distance vector changes, it sends the updated one to its neighbors
- If there are no changes, it will not send any messages to its neighbors
  - Note the difference from link state
- In reality, switches are not in lock step
  - Each recomputes its distance vector when a message arrives or one of its links changes

Distance vector routing (continued)

- When a switch's distance vector changes, it sends the updated one to its neighbors
- If there are no changes, it will not send any messages to its neighbors
  - Note the difference from link state
- In reality, switches are not in lock step
  - Each recomputes its distance vector when a message arrives or one of its links changes
- Example protocol: RIP

Reacting to link changes (in-class) (in-class discussion)

- How would you handle a change to link availability or cost in a link state routing algorithm?

Reset

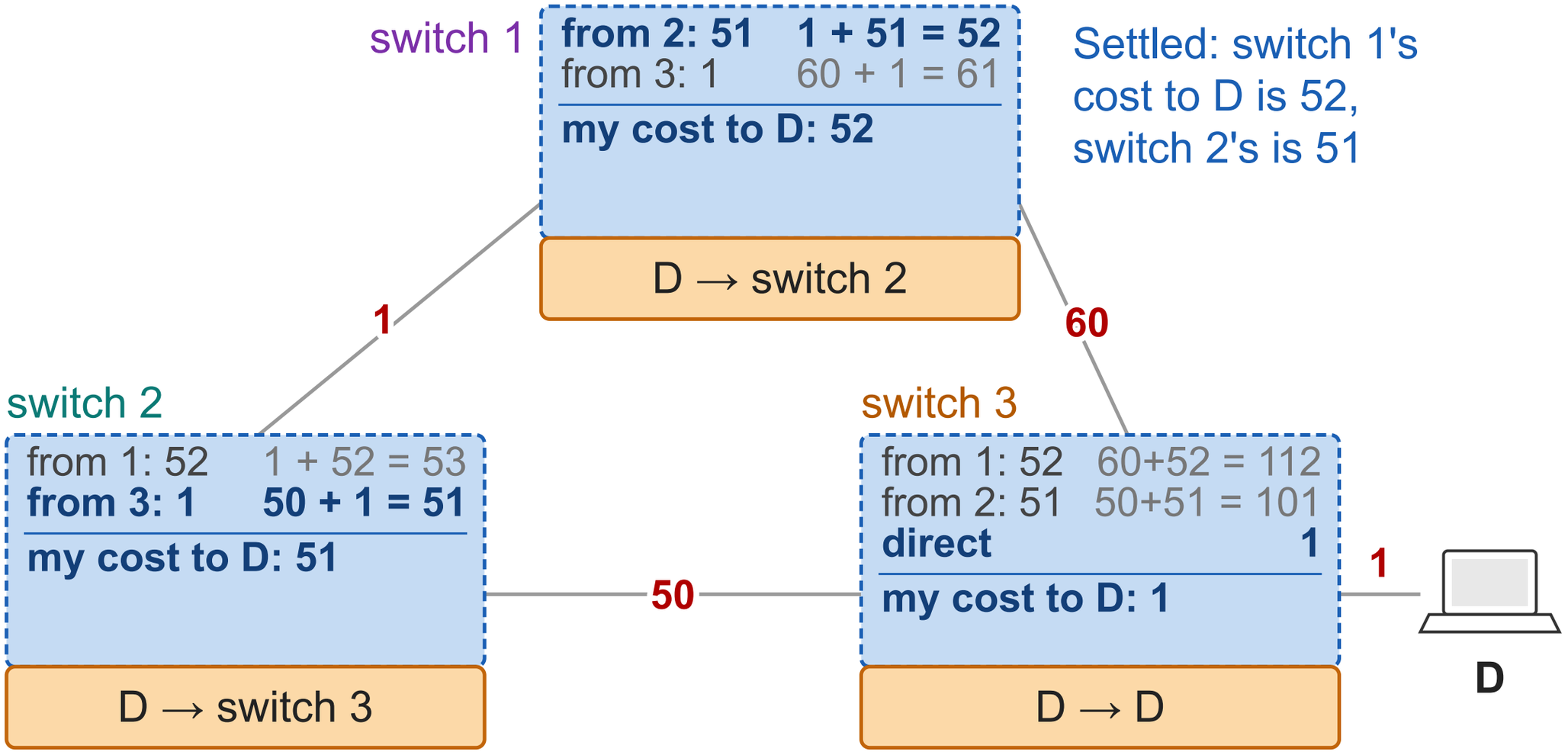
Reacting to link changes (in-class) (in-class discussion)

- How would you handle a change to link availability or cost in a link state routing algorithm?

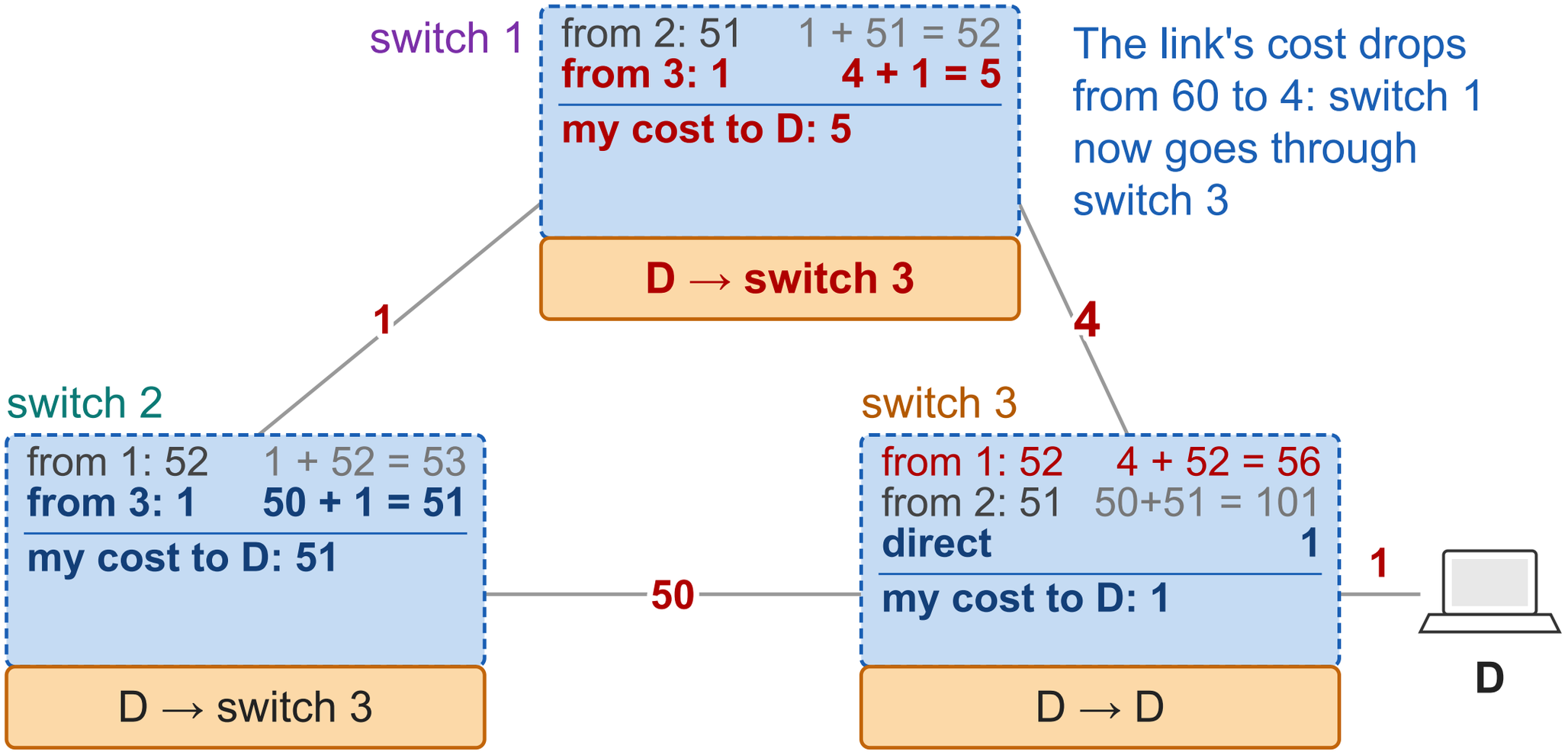
Reset

- What about distance vector?

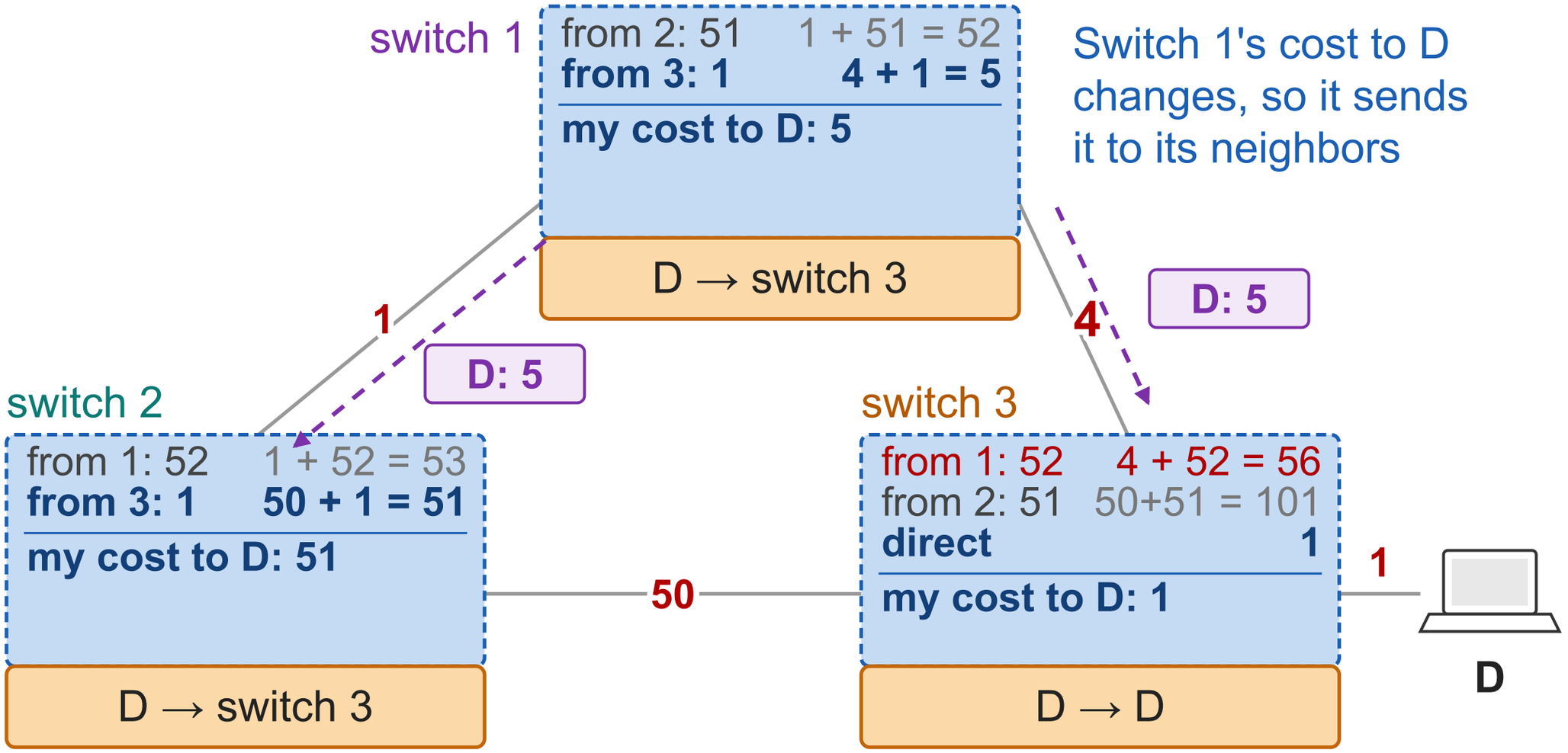
Distance vector: Link cost decreases



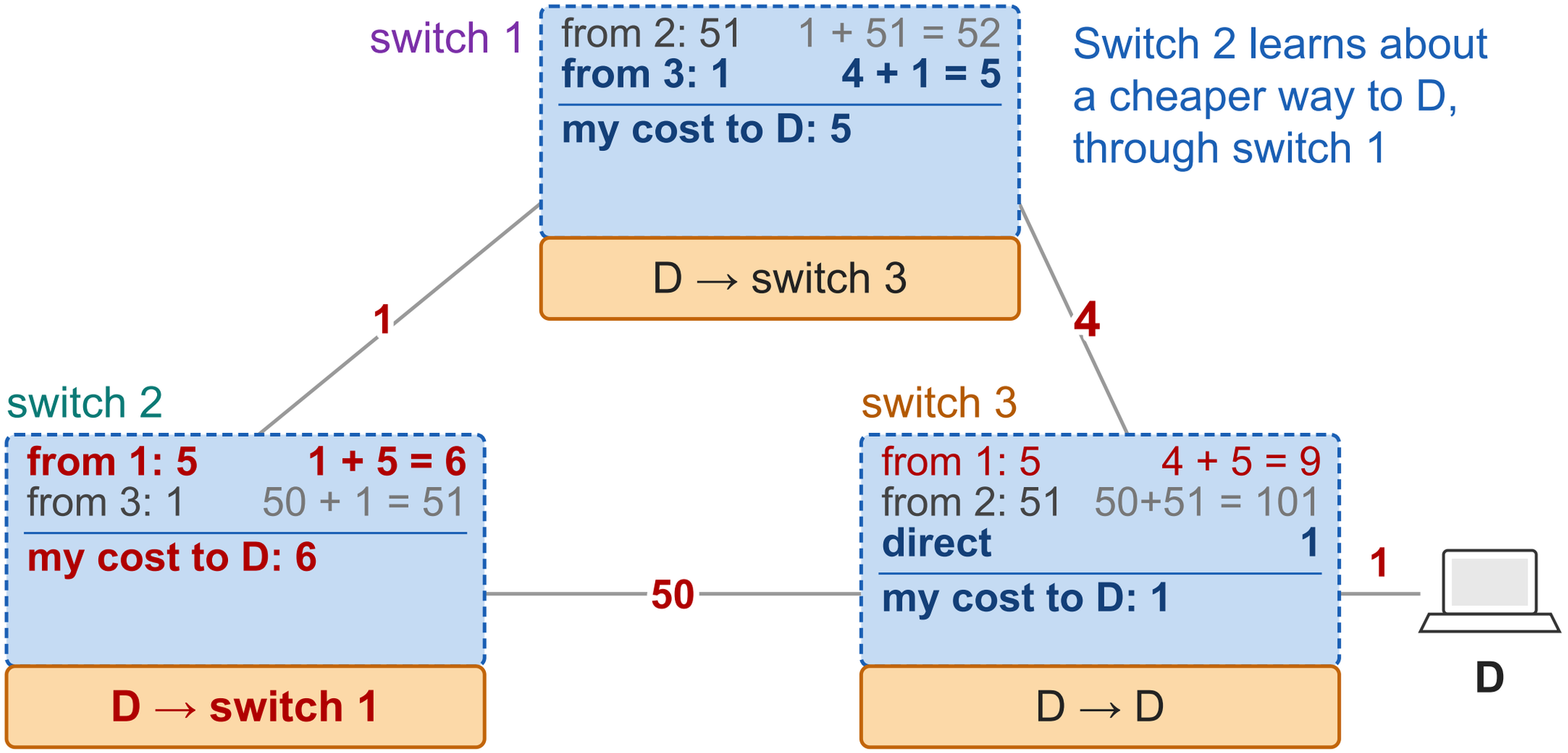
Distance vector: Link cost decreases



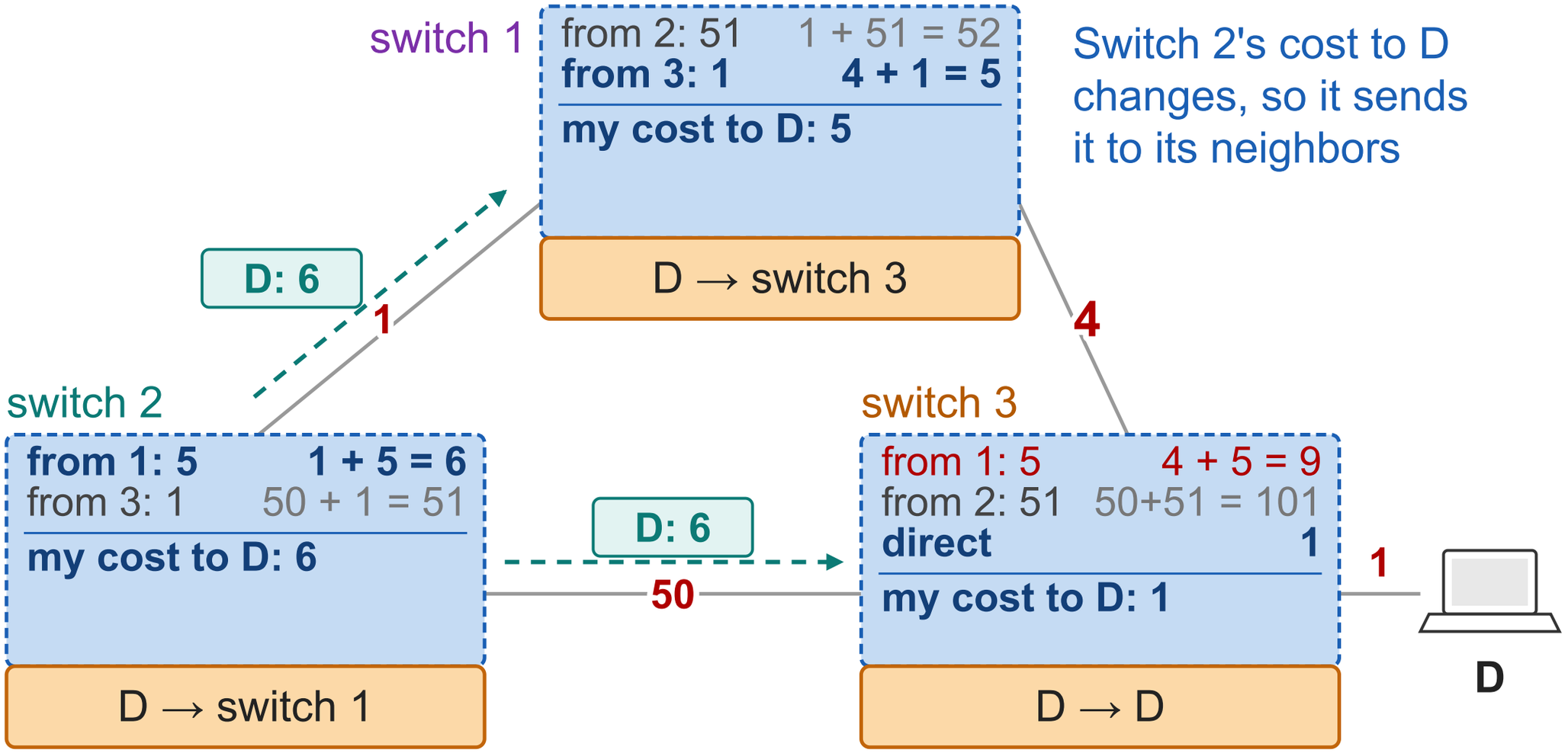
Distance vector: Link cost decreases



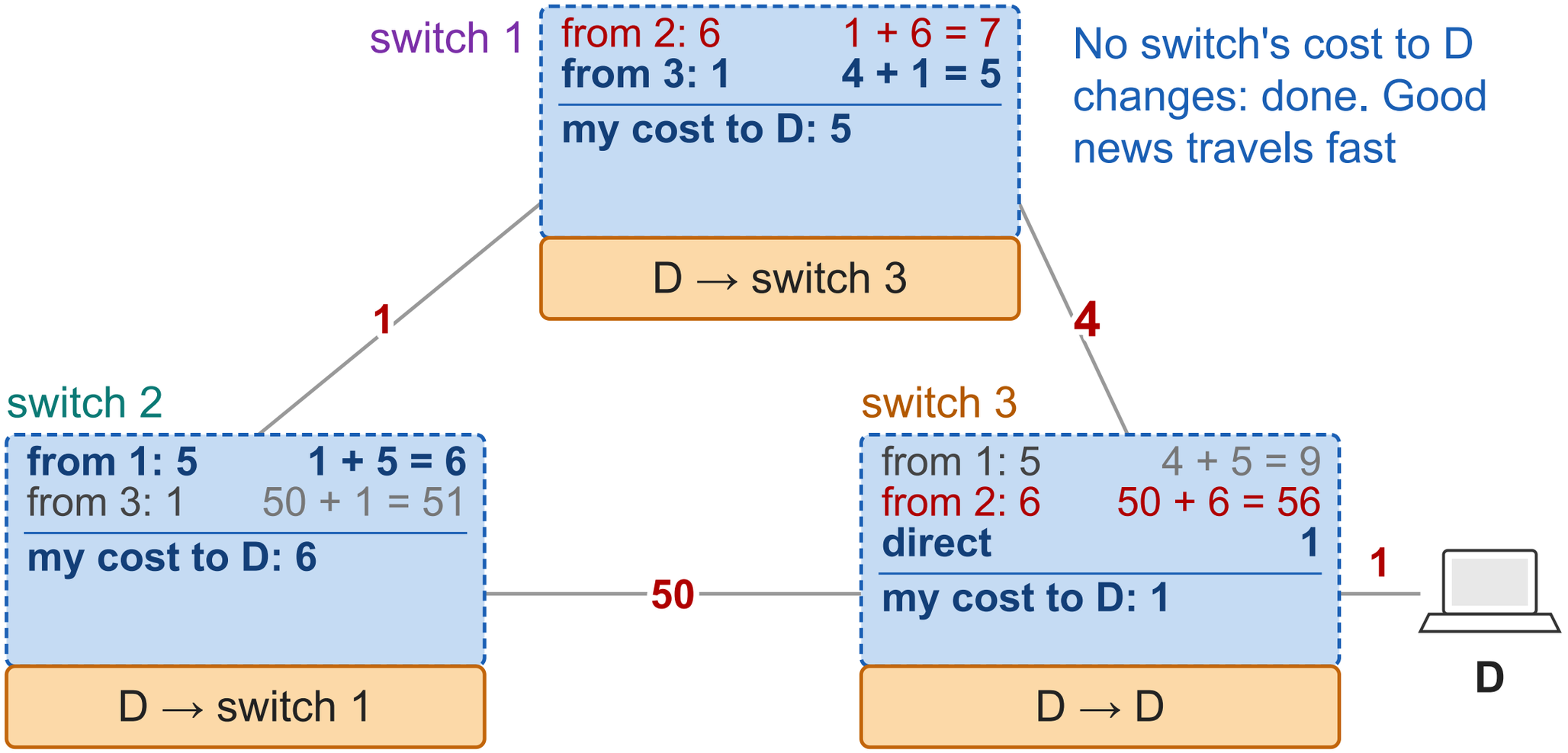
Distance vector: Link cost decreases



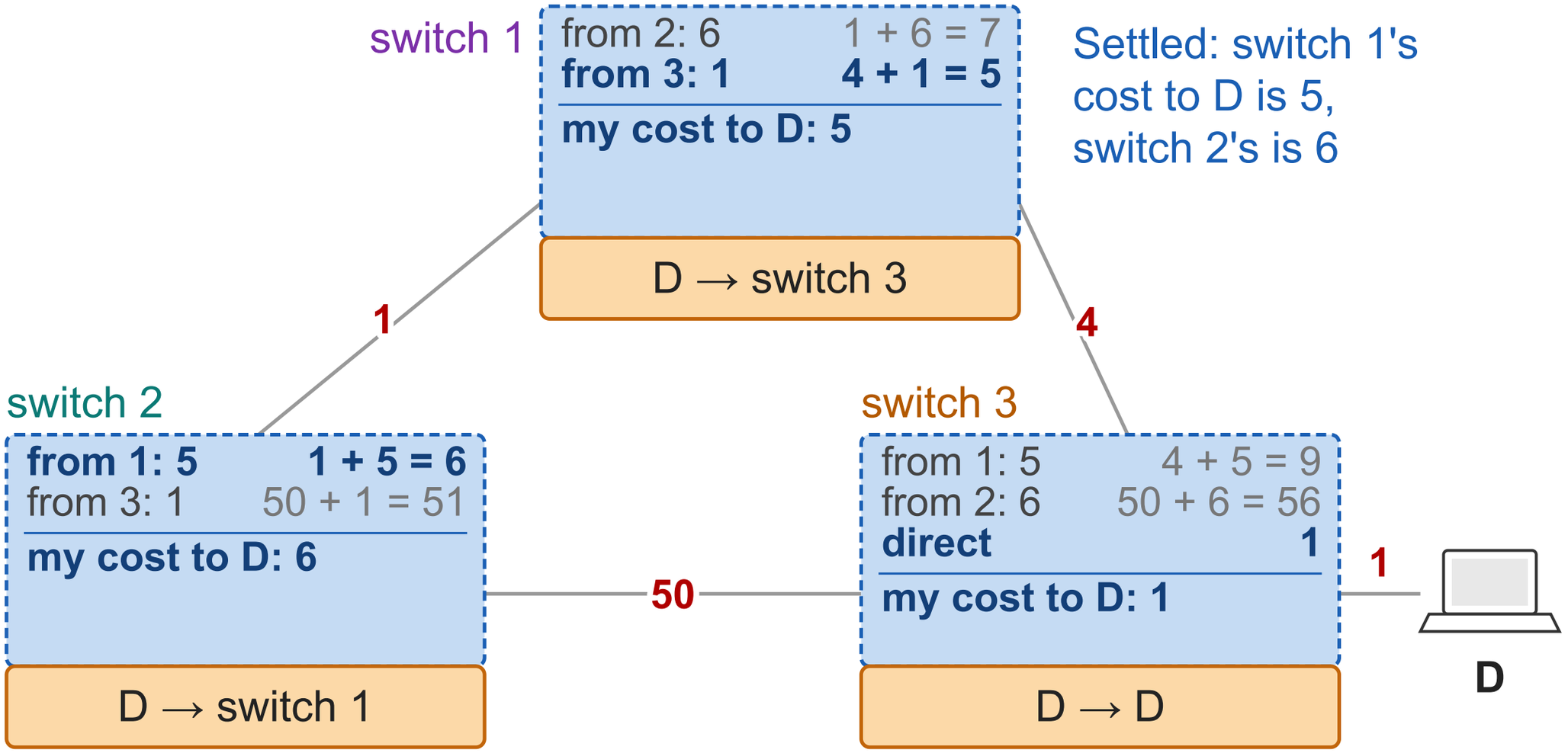
Distance vector: Link cost decreases



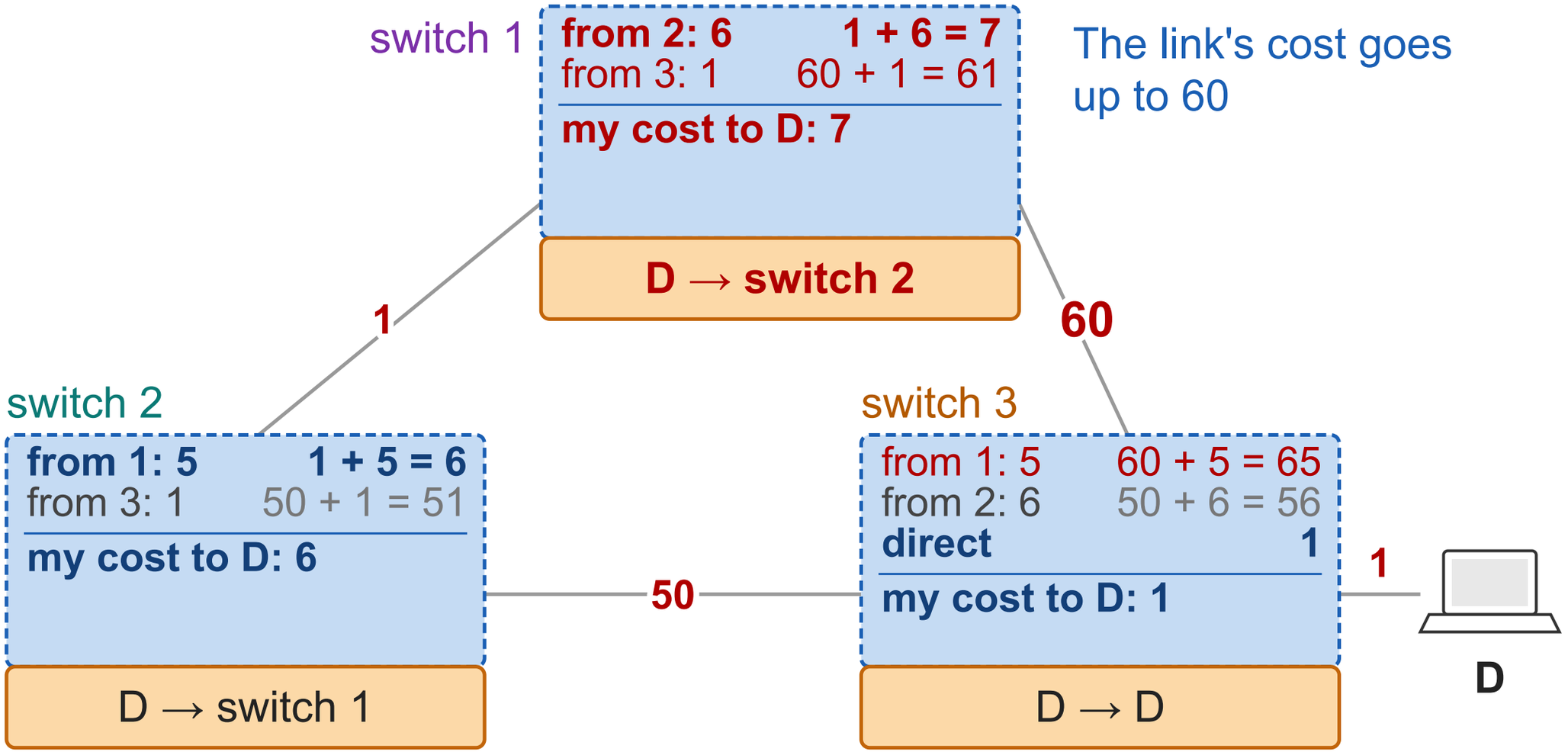
Distance vector: Link cost decreases



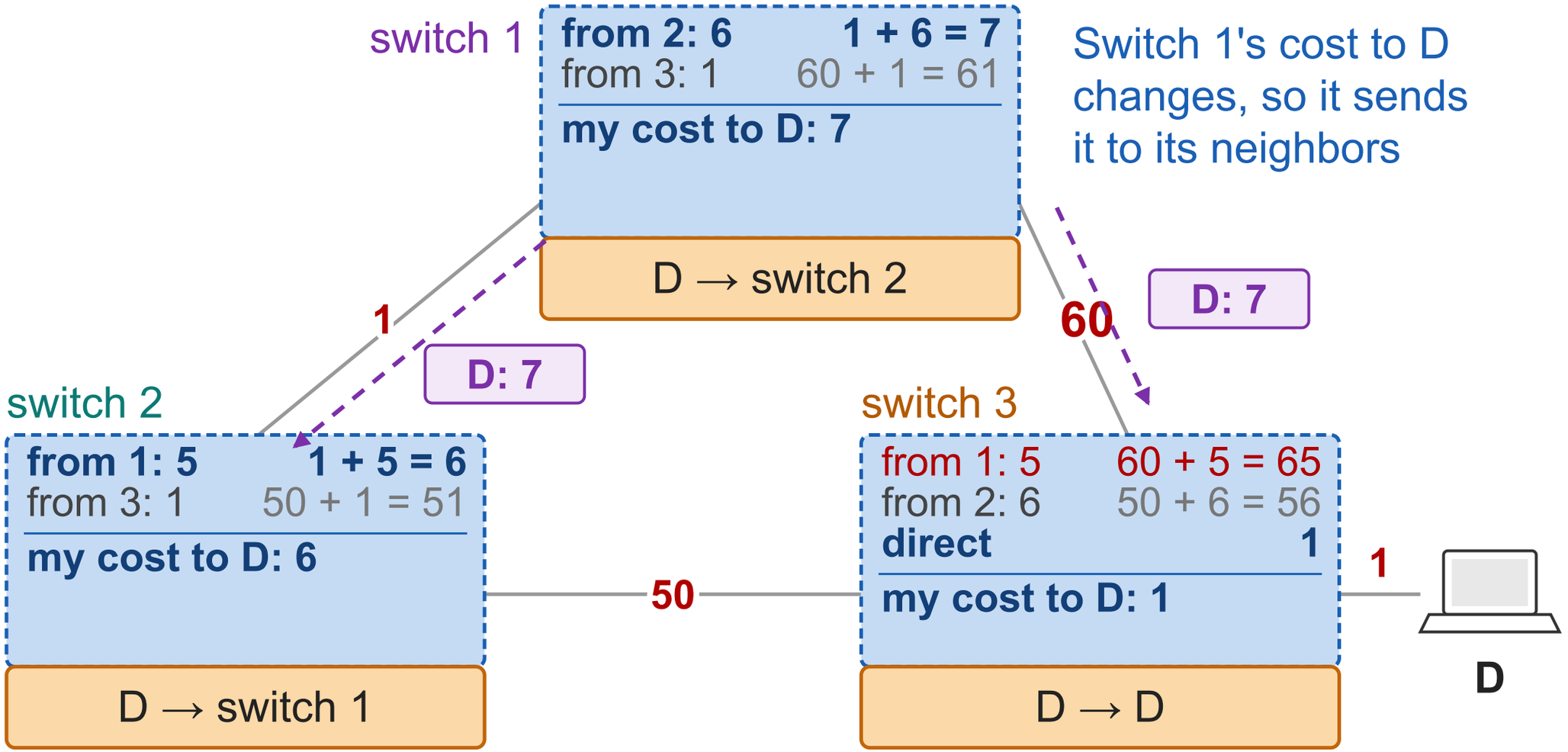
Distance vector: Link cost increases



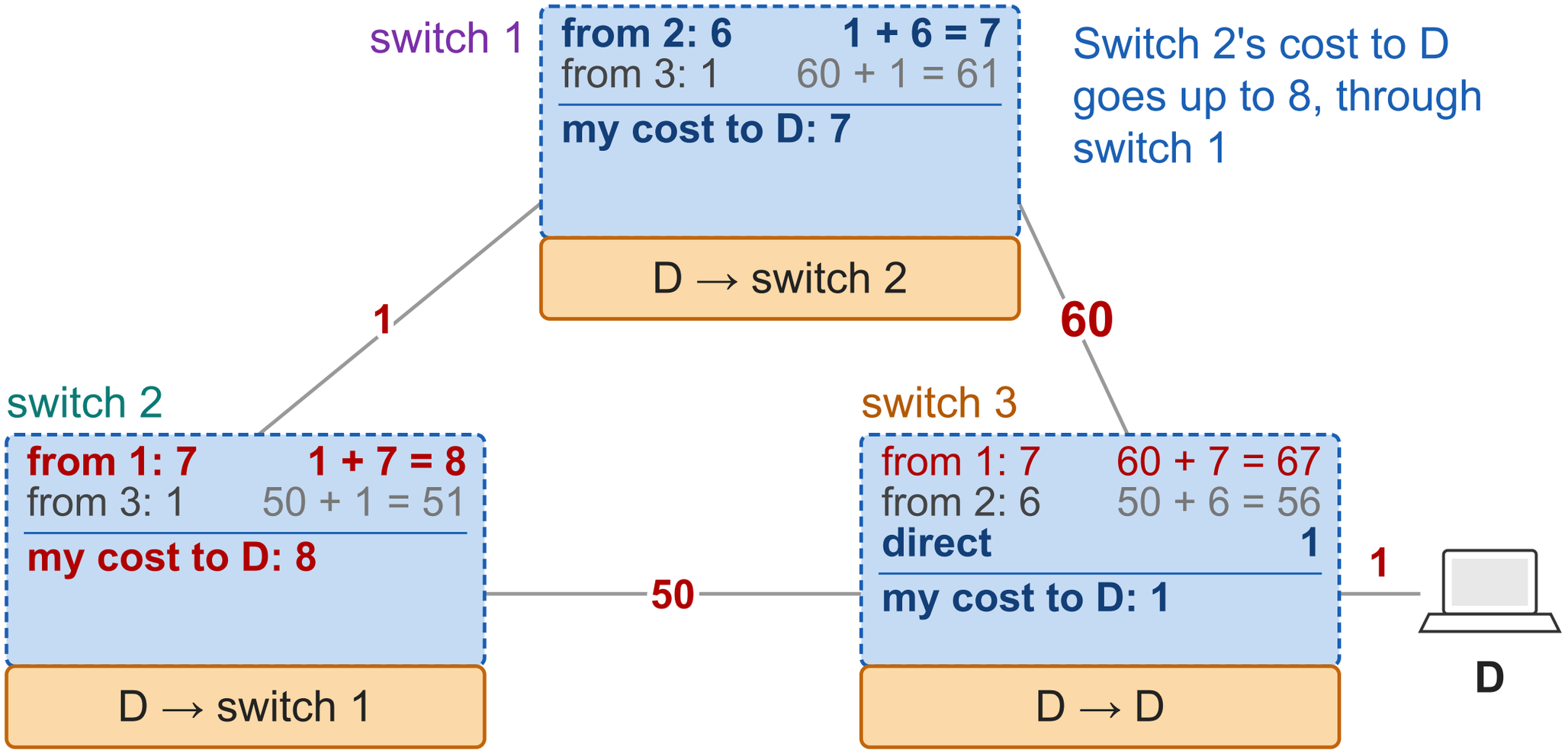
Distance vector: Link cost increases



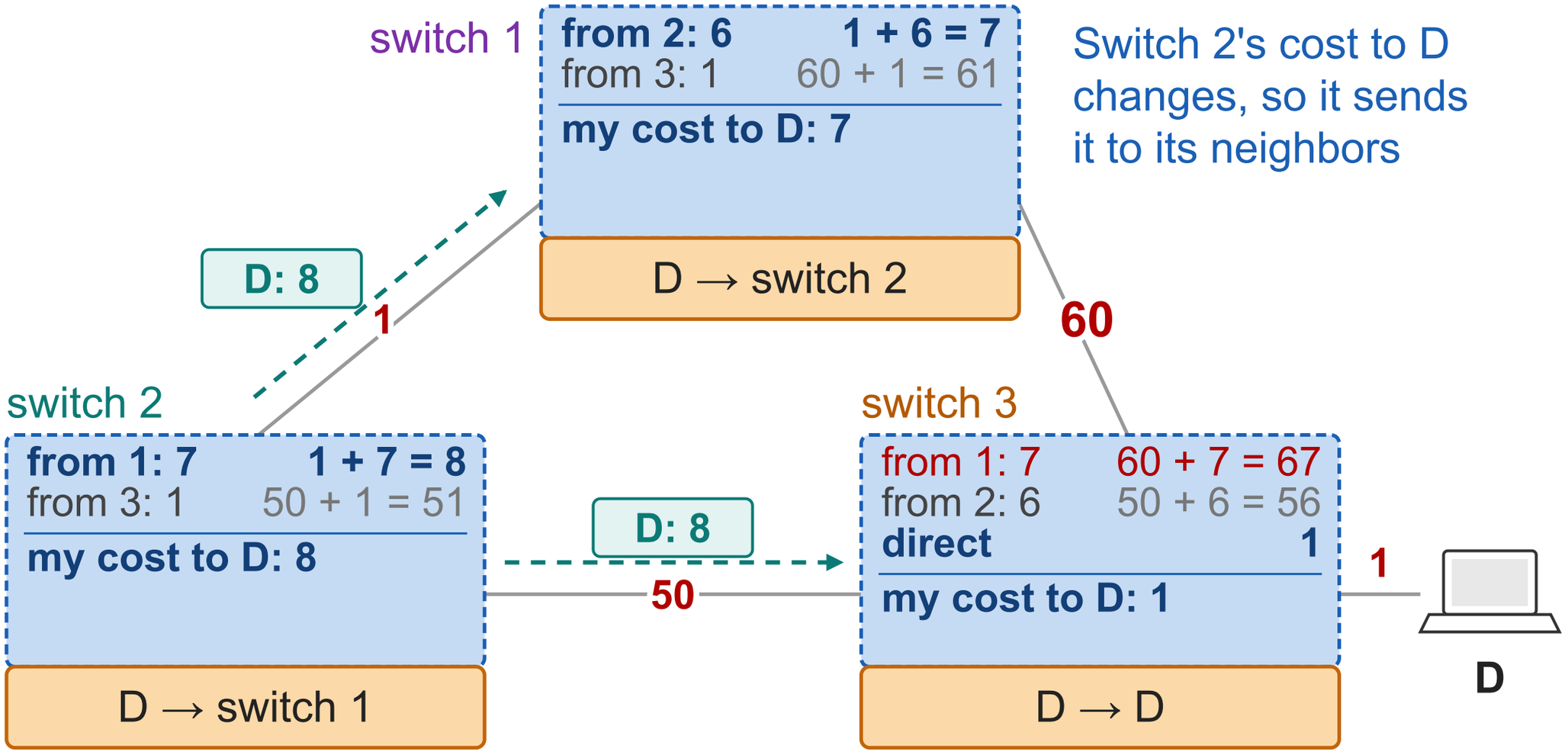
Distance vector: Link cost increases



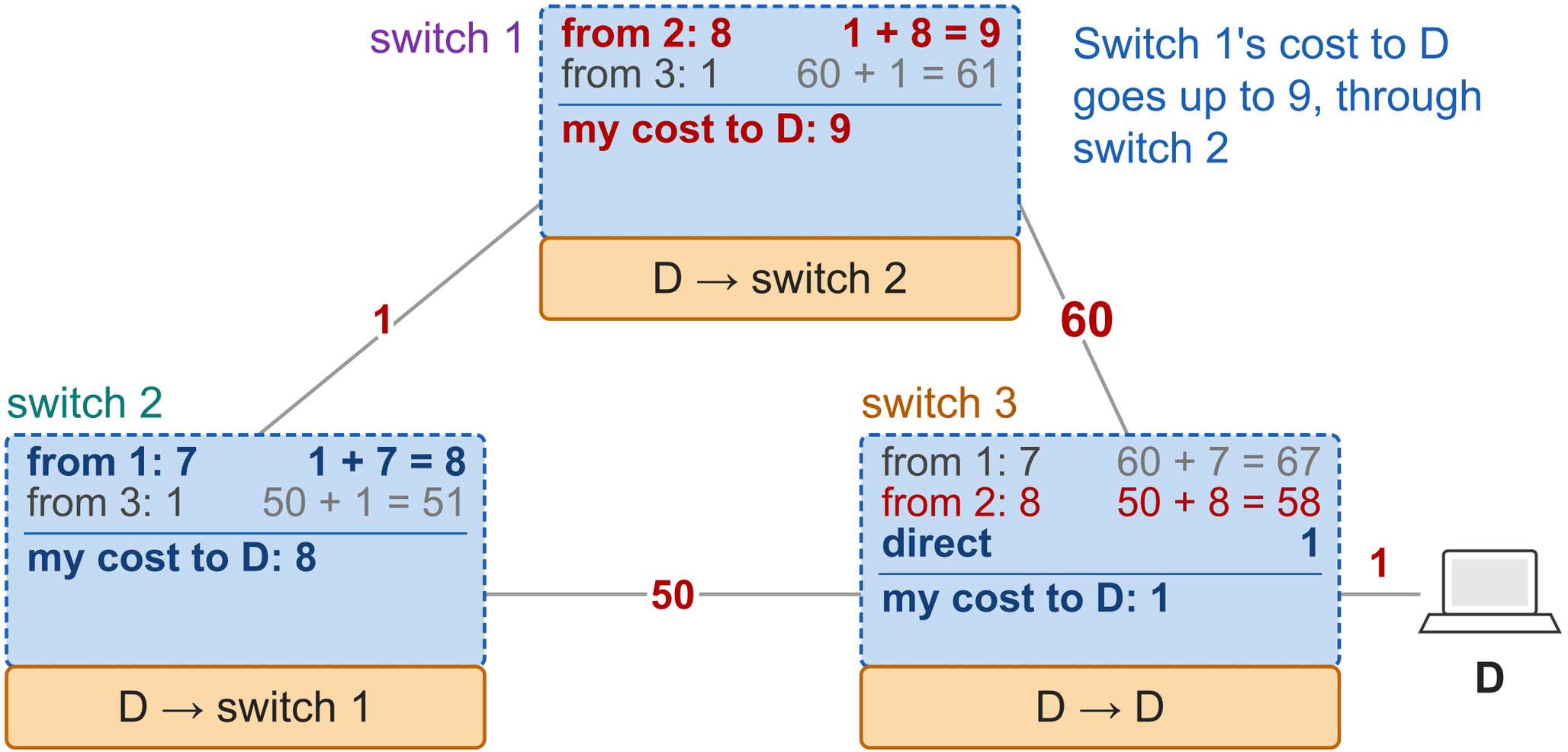
Distance vector: Link cost increases



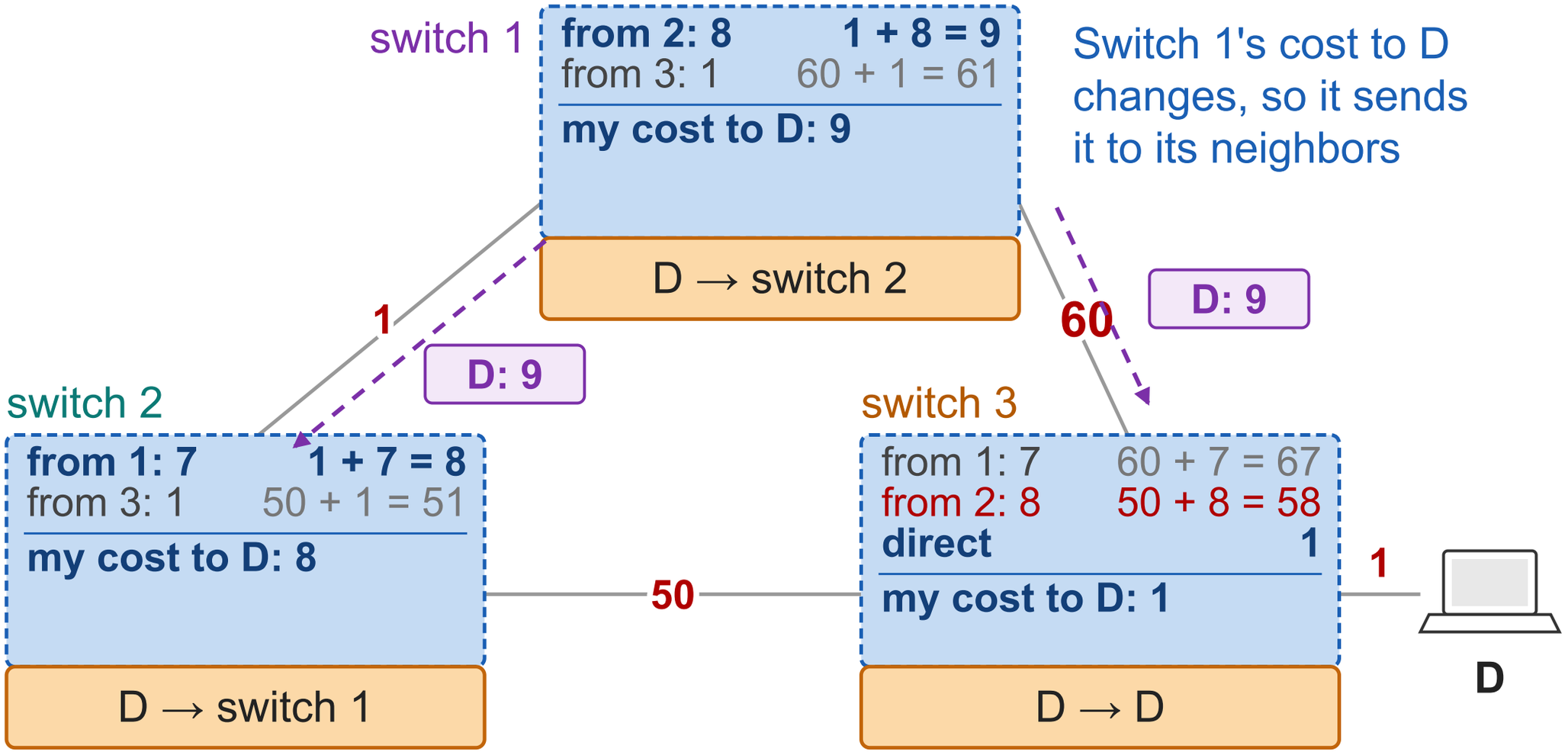
Distance vector: Link cost increases



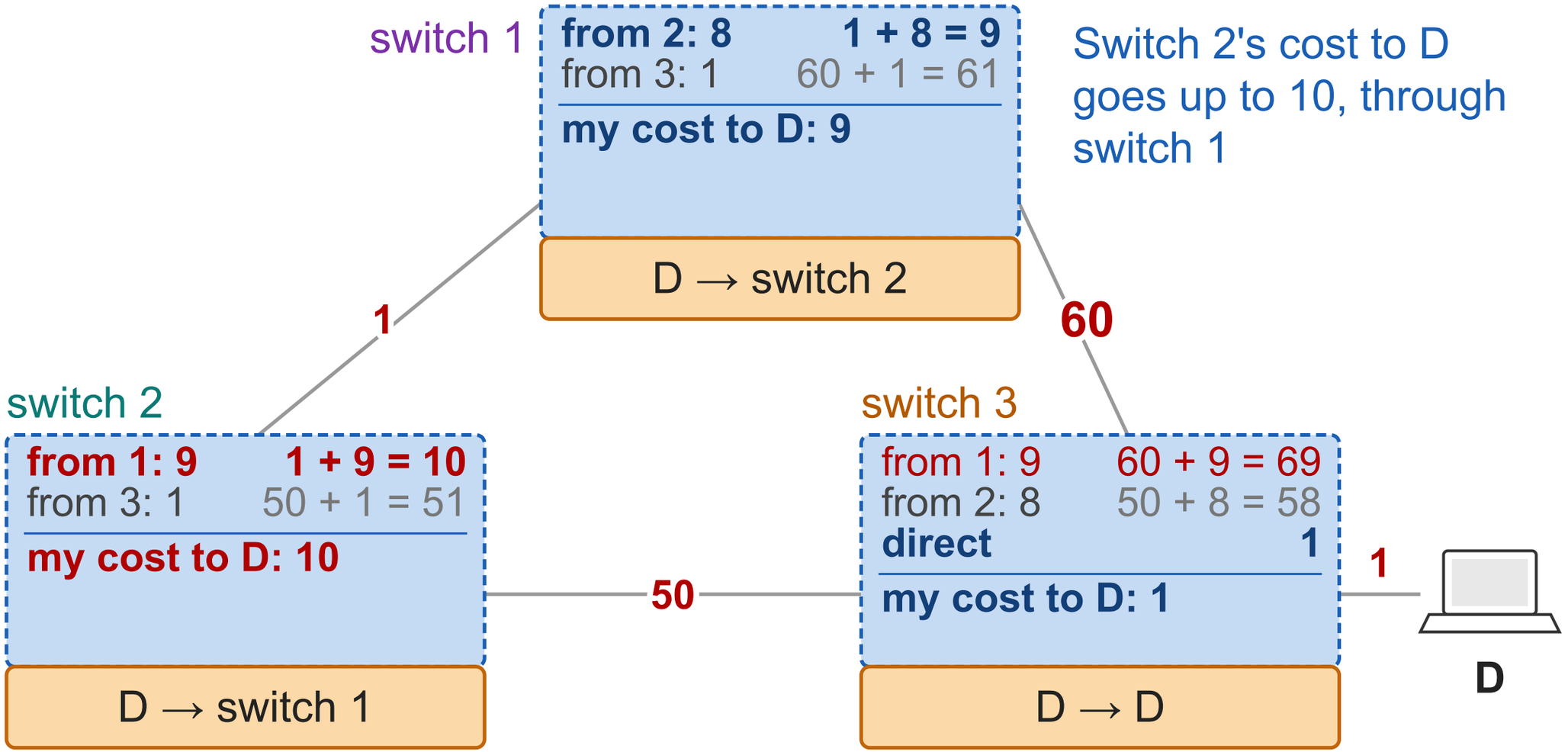
Distance vector: Link cost increases



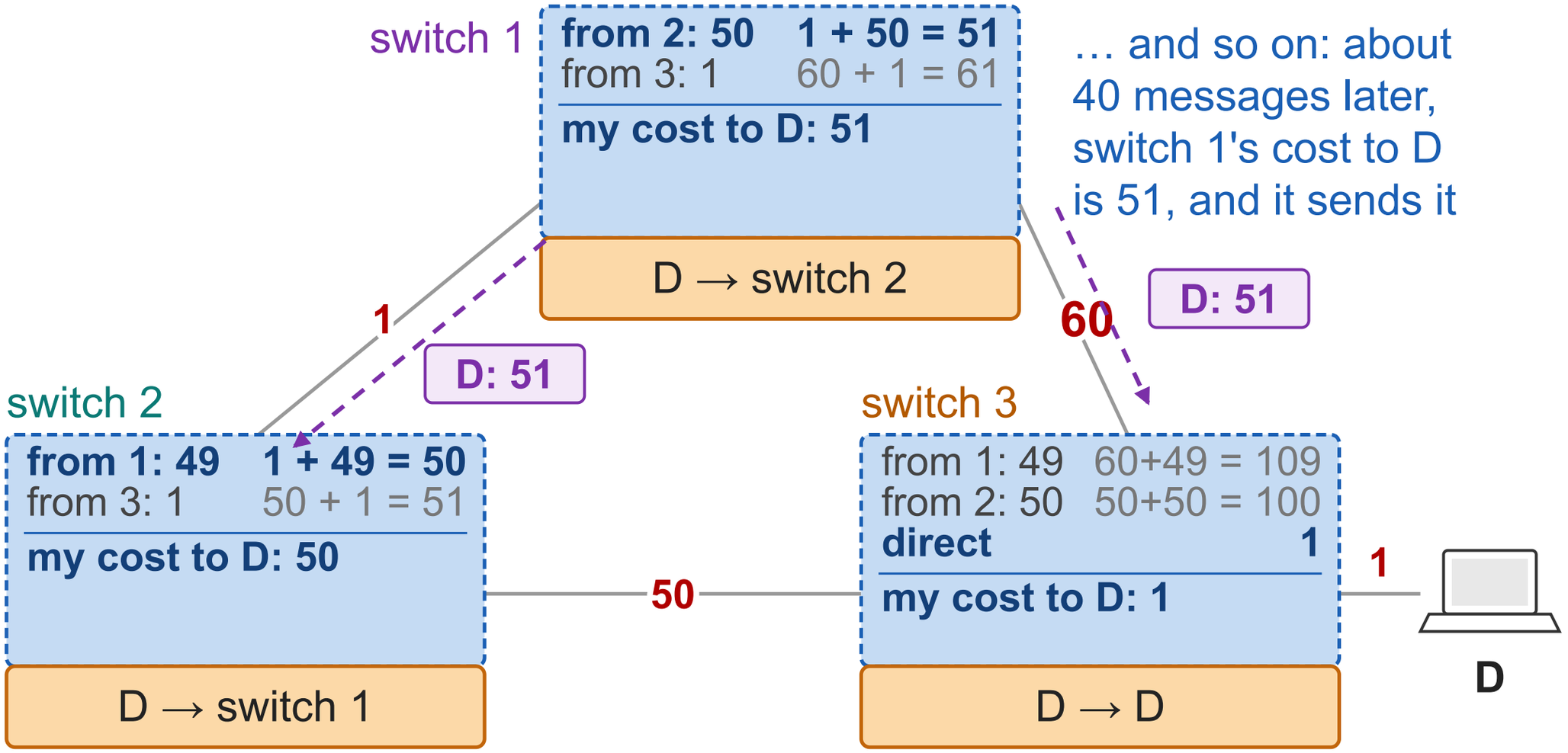
Distance vector: Link cost increases



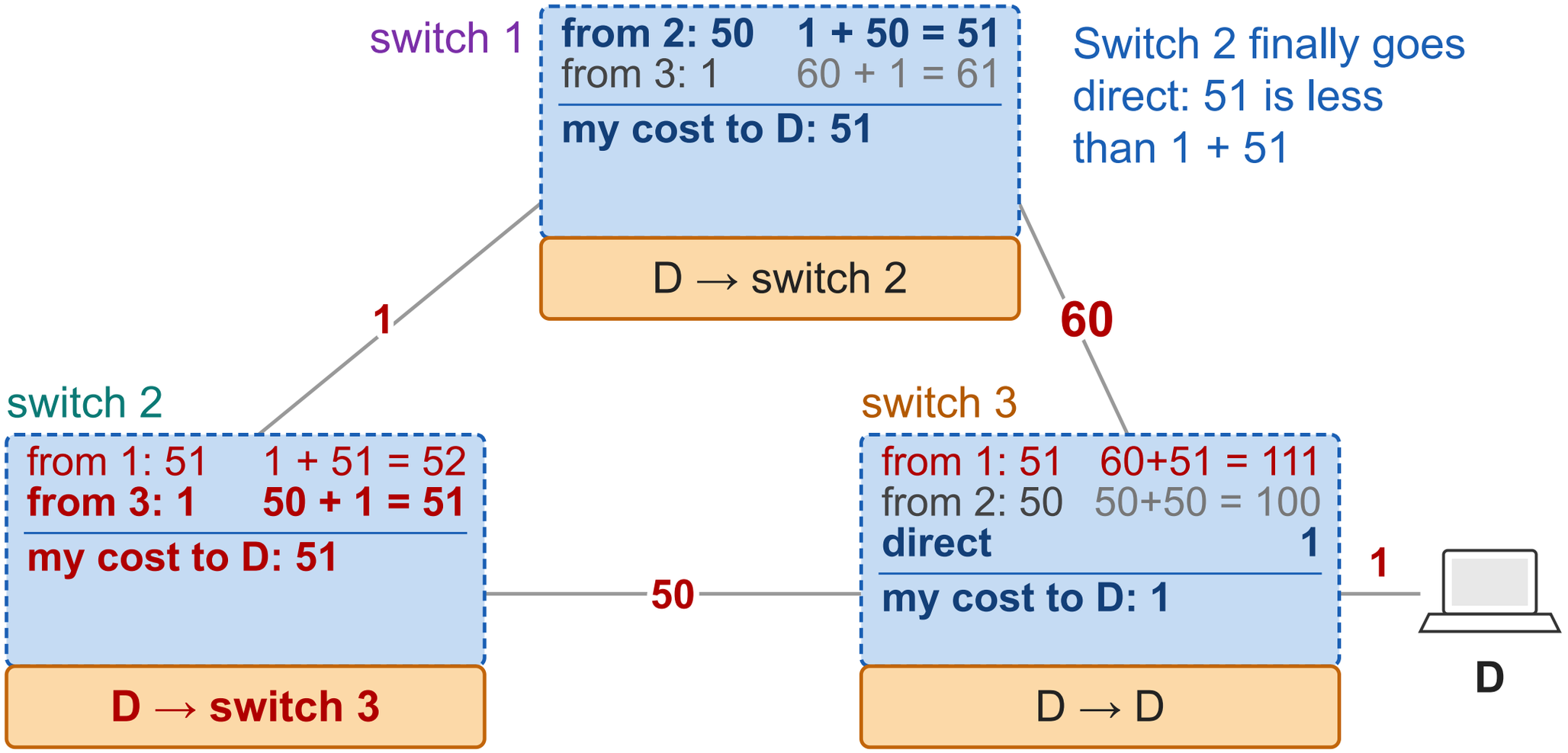
Distance vector: Link cost increases



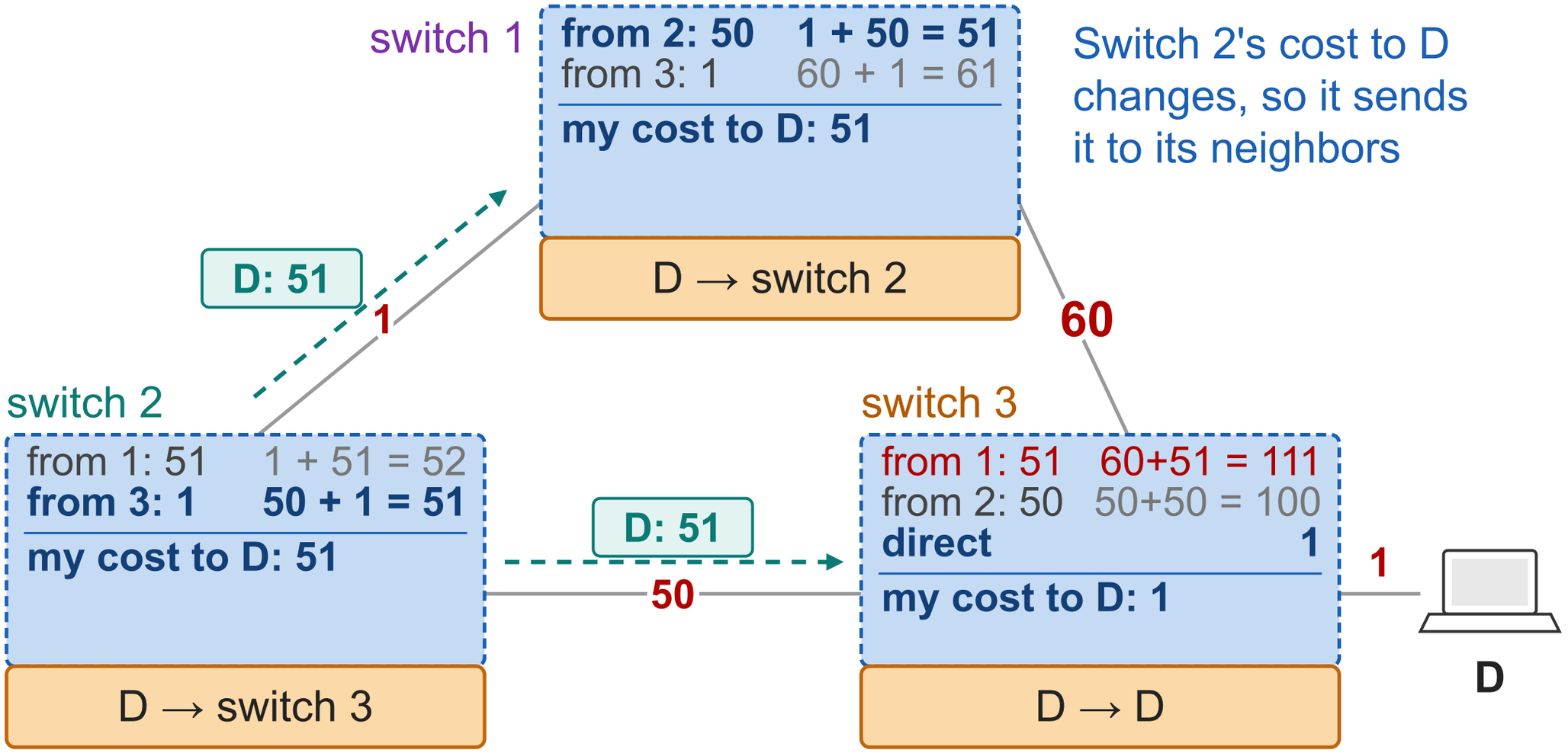
Distance vector: Link cost increases



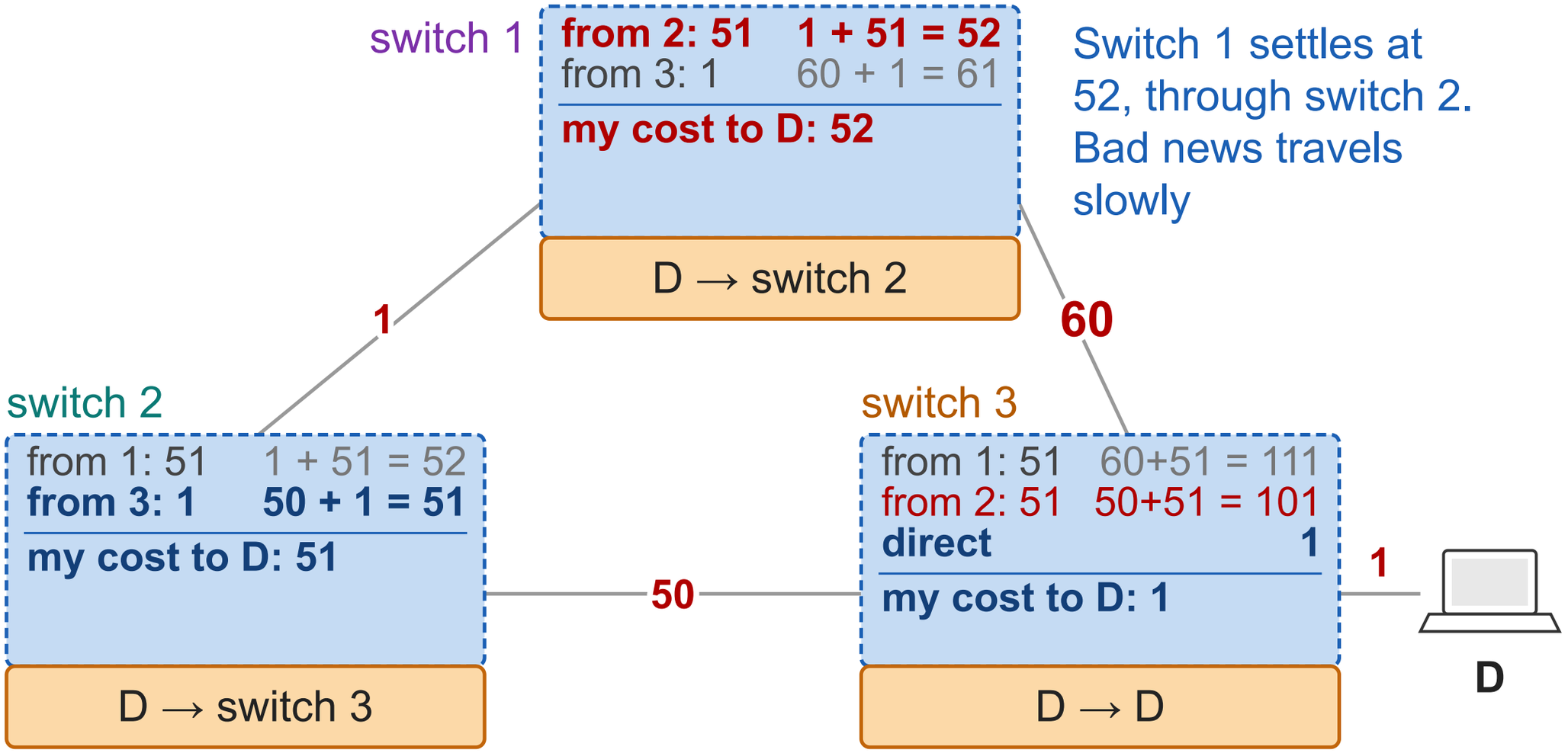
Distance vector: Link cost increases



Distance vector: Link cost increases



Distance vector: Link cost increases



What went wrong? (in-class) (in-class discussion)

- What is the problem? What happens to packets for D meanwhile?
- How would you fix it?



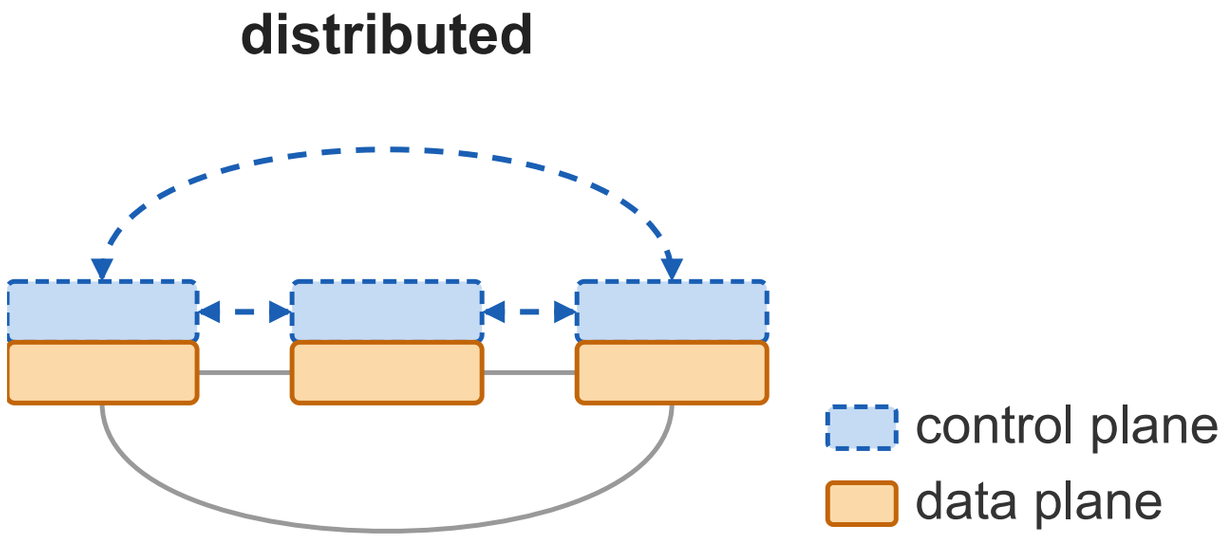
Reset

Comparing routing algorithms (in-class)

Metric	Routing algorithm 1	Routing algorithm 2

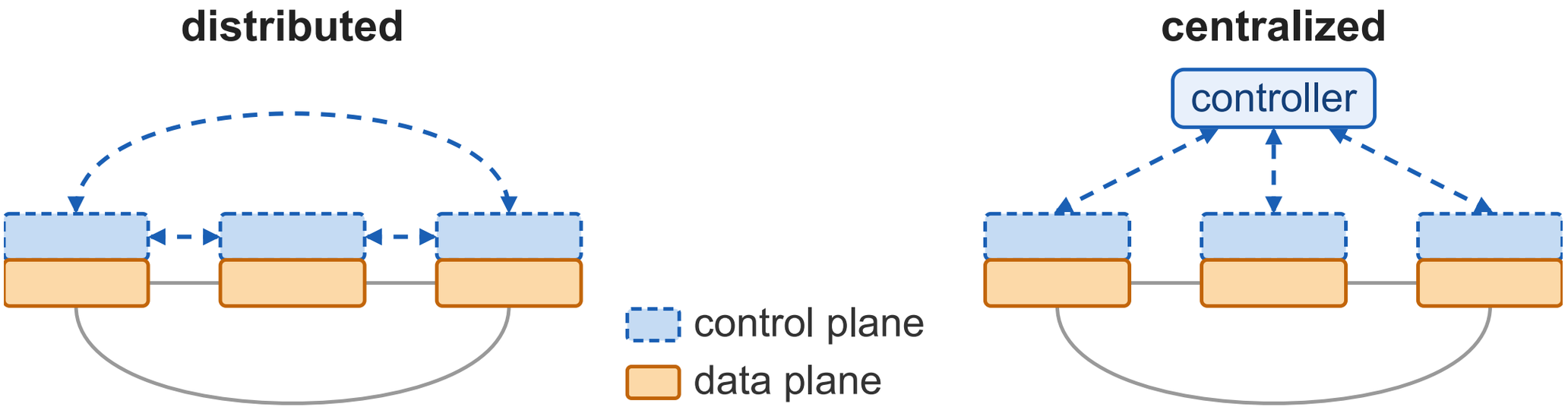
Reset

Who decides the paths?



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

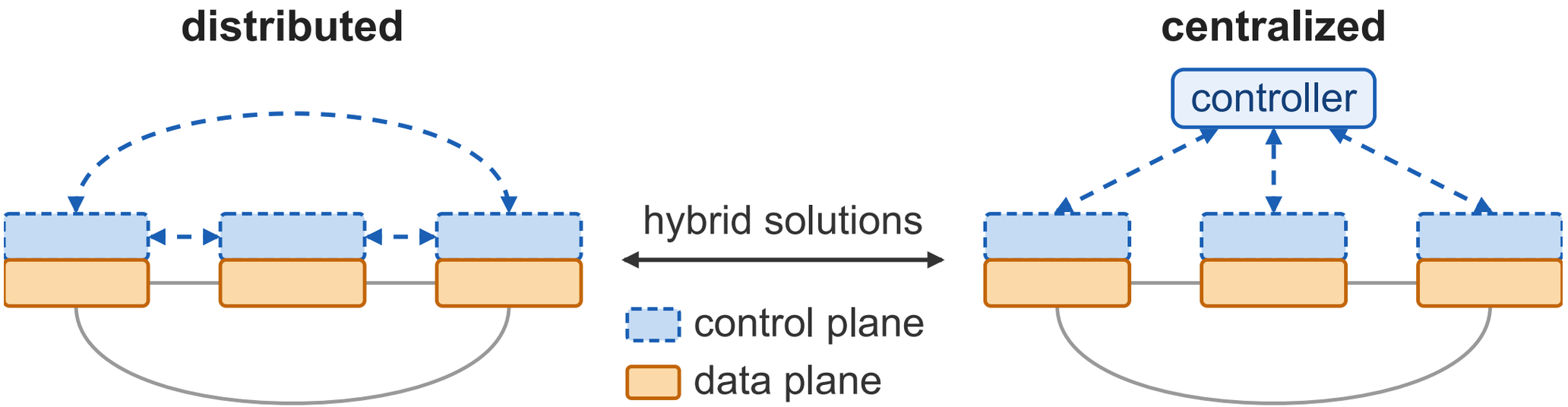
Who decides the paths?



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.

Centralized vs distributed (in-class) (in-class discussion)

- What are the pros and cons of a centralized and a distributed control plane?

Reset