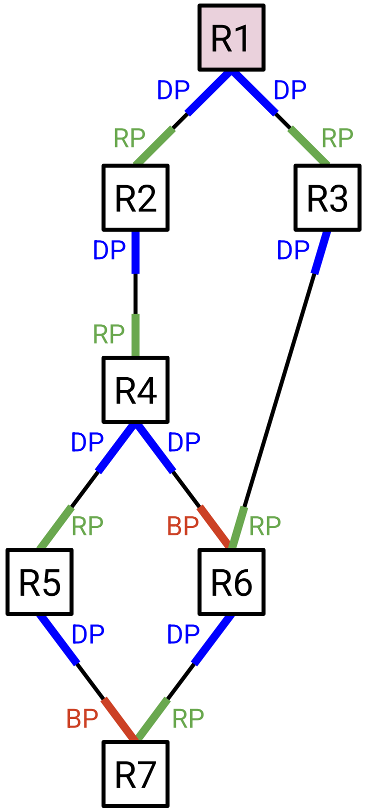
Lecture 18 (End-to-End 1)

Ethernet, STP



CS 168, Spring 2025 @ UC Berkeley

Slides credit: Sylvia Ratnasamy, Rob Shakir, Peyrin Kao, Murphy McCauley

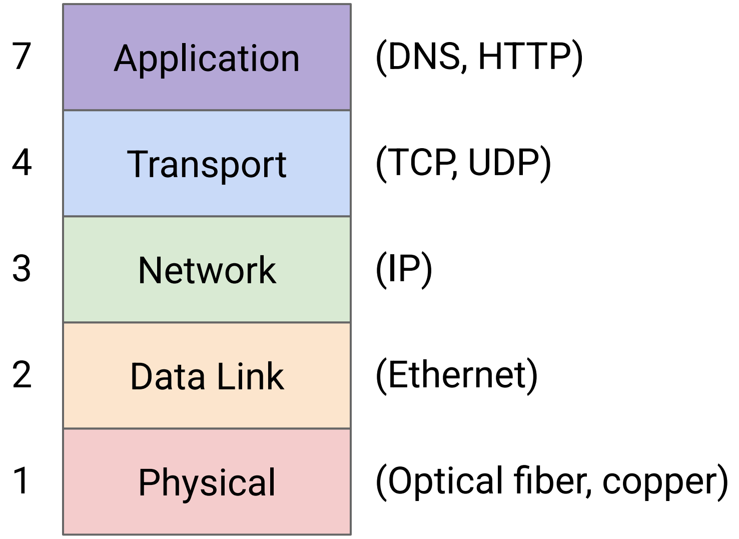
Connecting Local Hosts

Lecture 18, CS 168, Spring 2025

Link Layer

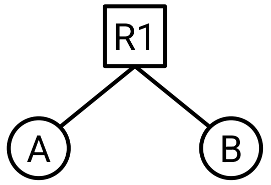
Today: A closer look at Layer 2.

- Ethernet: How do we send data inside a local network?
- ARP: How do we connect Layer 2 and Layer 3?
- DHCP: What happens when you join the network for the first time?

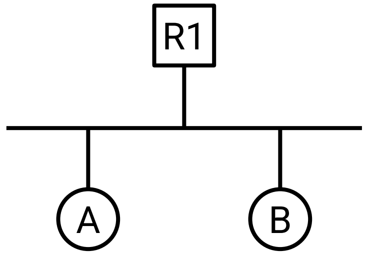


Connecting Local Hosts

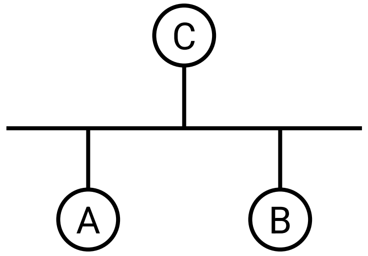
So far, we've assumed that every link connects exactly two machines:



In reality, a single wire can connect multiple computers:

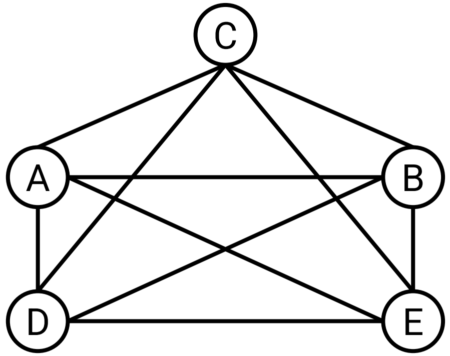


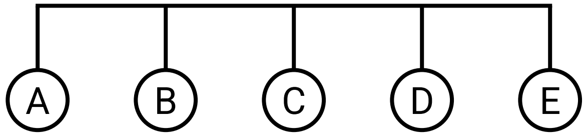
From the wire's perspective, the router is just a machine like any other:



Connecting Local Hosts

How could we connect hosts in a local network?





Mesh: Link between every pair of machines.

- For every new host, we have to add new links to every other host.
- Need a lot of (physical) ports per host.

Bus: A single wire for all machines.

- Introduces a **shared media**.

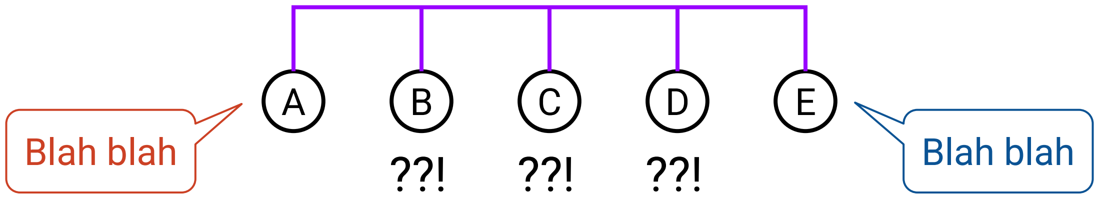
Shared Media

Shared media: Many machines using the same wire.

- If multiple machines transmit at the same time, signals will *interfere* or *collide*.
- Analogy: People talking simultaneously on a group call.

Note: Shared media is not necessarily a wire.

- Could be light signals on a shared optical fiber.
- Or radio waves on a shared wireless link.



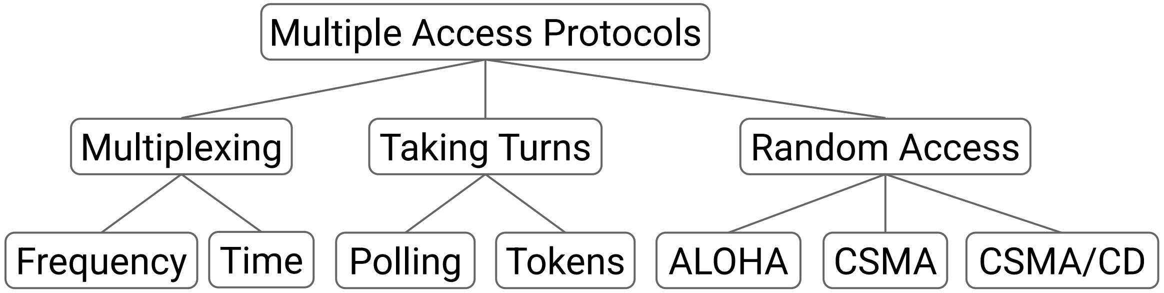
Multiple Access Protocols

Lecture 18, CS 168, Spring 2025

Multiple Access Protocols

A **multiple access protocol** allocates the shared media to everyone wanting to use it.

- 3 types of approaches.
- Several protocols of each type.



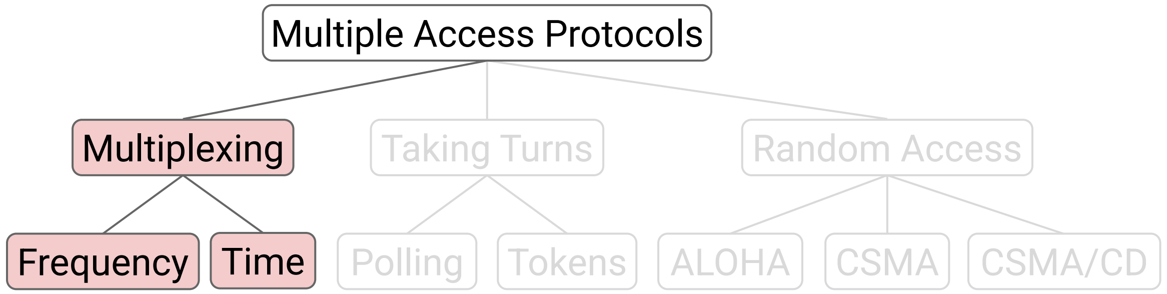
Multiple Access Protocols (1/3) – Multiplexing

Idea: Allocate a fixed slice of resources to each node.

- **Frequency-based** multiplexing: Divide medium into frequency channels.
- **Time-based** multiplexing: Give each node some fixed time slots.

Problem: Can be wasteful.

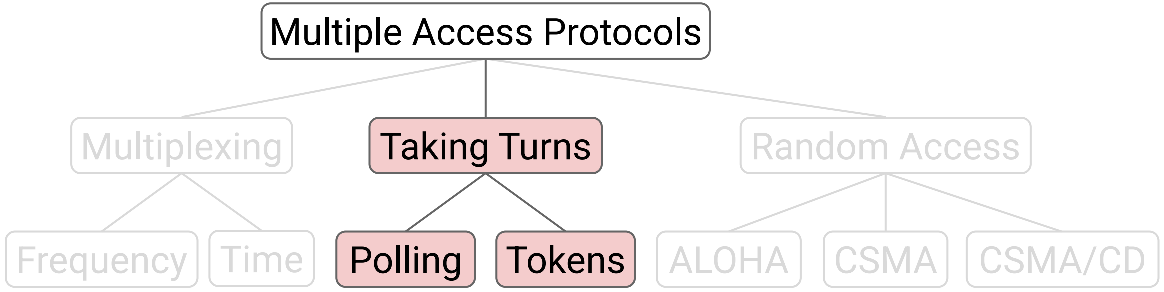
- There's only so much frequency/time available to allocate.
- If a node has nothing to say, some frequency/time goes unused.



Multiple Access Protocols (2/3) – Taking Turns

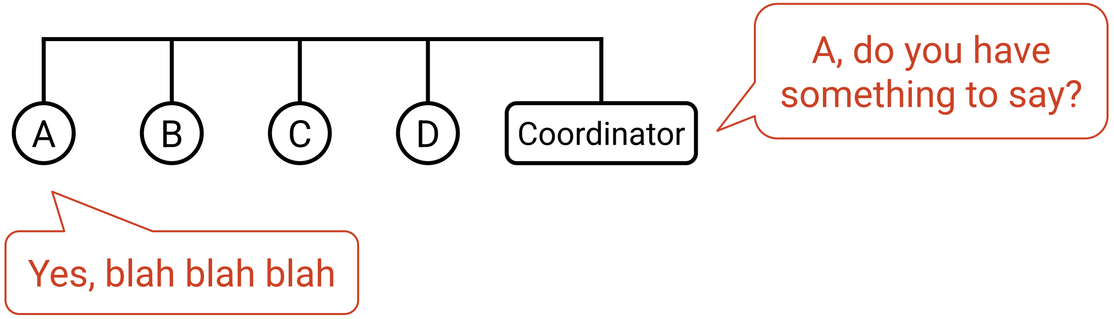
Idea: Nodes take turns speaking.

- **Polling protocols**: A coordinator decides when each node can speak.
  - Example: Bluetooth.
- **Token passing**: Pass a virtual token around. Only the node with the token can speak.
  - Example: IBM Token Ring, FDDI.

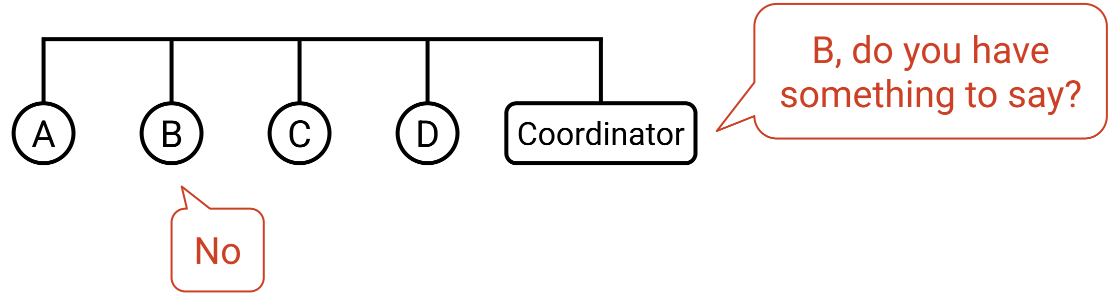


Multiple Access Protocols (2/3) – Taking Turns

Polling protocols: A coordinator decides when each node can speak.



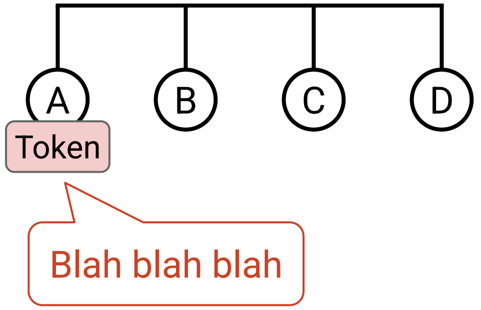
A can speak for as long as it needs.



B has nothing to say. Coordinator can immediately move on to C.

Multiple Access Protocols (2/3) – Taking Turns

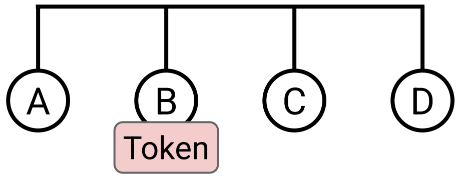
Token passing: Pass a virtual token around. Only the node with the token can speak.



A holds the token.

A can speak for as long as it needs.

When A is done, it passes the token to B.



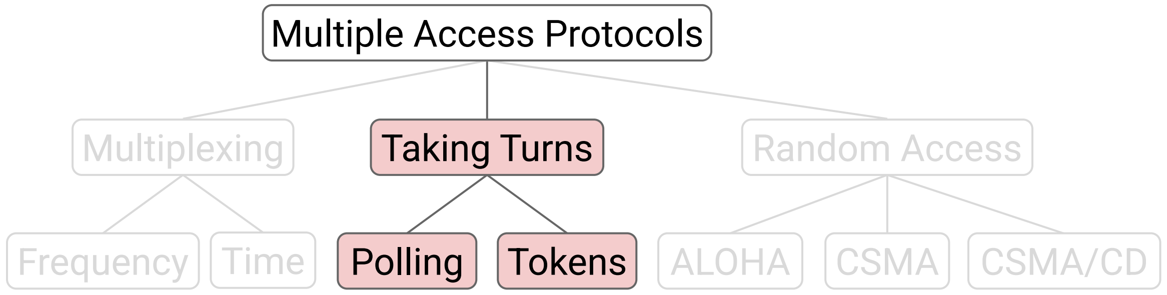
B has nothing to say.

It can immediately pass the token to C.

Multiple Access Protocols (2/3) – Taking Turns

Idea: Nodes take turns speaking.

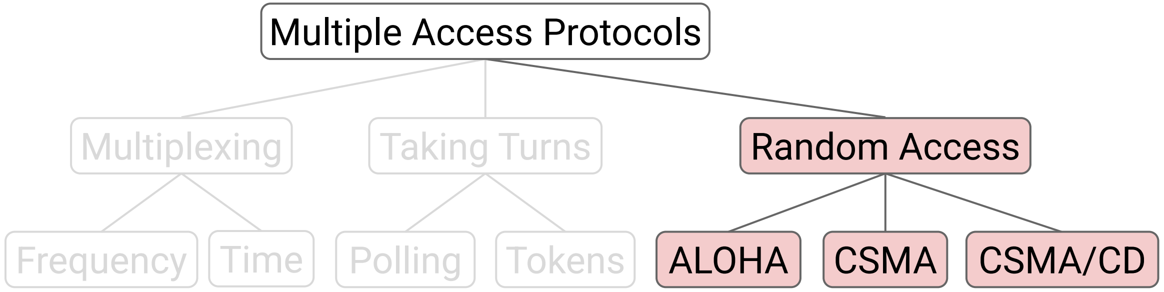
- Benefit: No more time wasted on idling.
  - If someone has nothing to say, immediately move on.
- Problem: Complexity.
  - Need to implement inter-node communication.
  - How do we elect the central coordinator?
  - What if two nodes both think they have the token?



Multiple Access Protocols (3/3) – Random Access

Idea: Nodes talk whenever they have something to say.

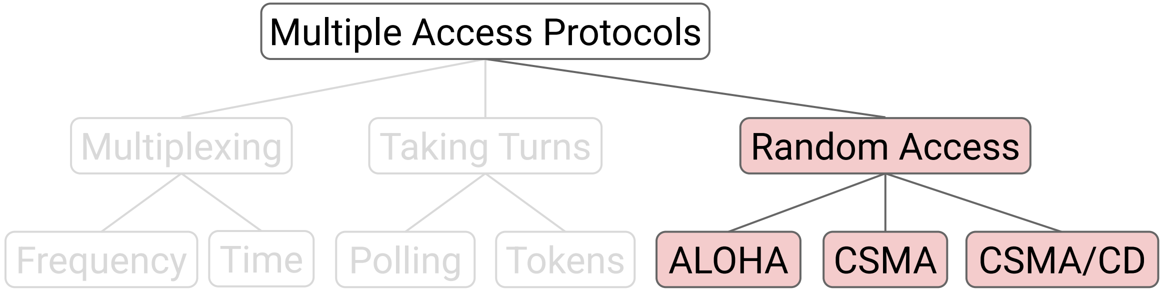
- Deal with collisions when they occur.
- Benefit: Simplicity. No coordinators or tokens.



Multiple Access Protocols (3/3) – Random Access – ALOHA

ALOHA random access scheme: "rude" version.

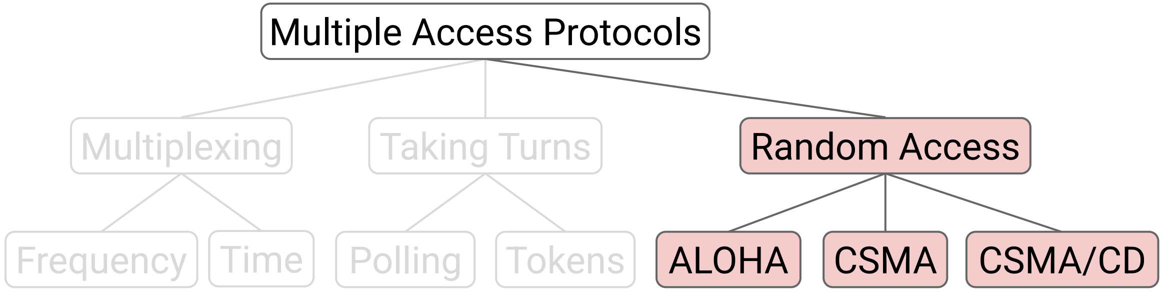
- If you have a packet, just send it.
  - Recipient replies with an ack.
- If two nodes send simultaneously, collision corrupts the packets.
  - No ack!
- If you don't get an ack: Wait some random amount of time, then resend.
  - Randomness helps avoid another collision.



Multiple Access Protocols (3/3) – Random Access – CSMA

CSMA (Carrier Sense Multiple Access): "polite" version.

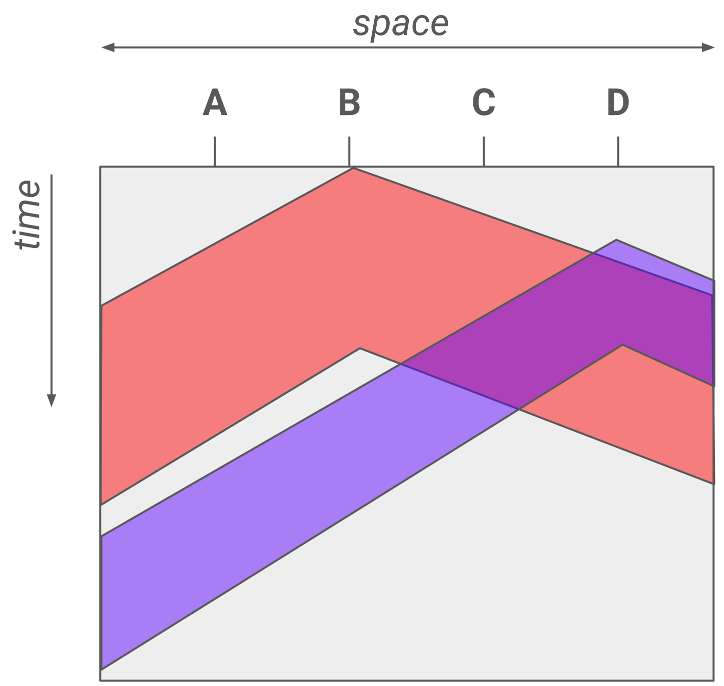
- First, listen to see if anyone is sending.
- Only start sending when it's quiet.



Multiple Access Protocols (3/3) – Random Access – CSMA and Propagation Delay

CSMA does not necessarily avoid collisions, because of **propagation delay**.

- t=0: B starts sending.
    - Signal takes time to reach A, C, D.
- t=2: D wants to send.
    - Signal hasn't reached D yet!
    - D thinks it's quiet and starts sending.
- Result: Collision!



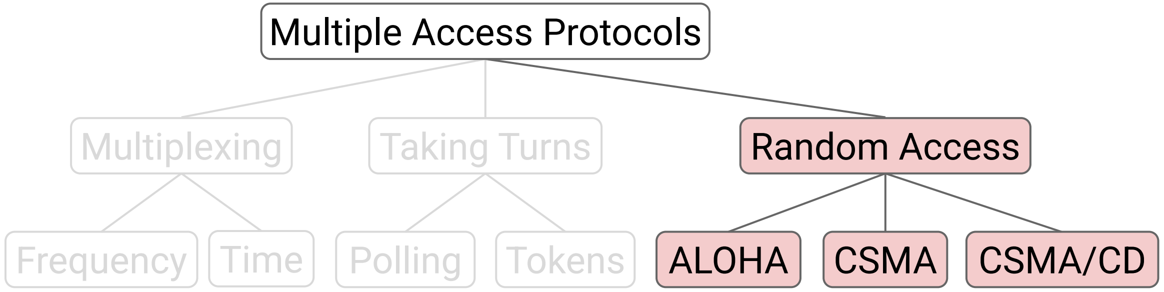
Multiple Access Protocols (3/3) – Random Access – CSMA/CD

CSMA/CD (Carrier Sense Multiple Access with Collision Detection):

- Listen before sending, but also *while* you're sending.
- If you hear someone else sending, stop! Collision detected.

CSMA/CD uses **binary exponential backoff**:

- After every collision, wait up to twice as long before resending.
  - After first collision: Wait between 0–4 seconds.
  - If resend collides again: Wait between 0–8 seconds.
- Resends fast when possible, slowing down when necessary (e.g. many senders).



Brief History of Multiple Access Protocols

Why was that protocol called ALOHA, anyway?

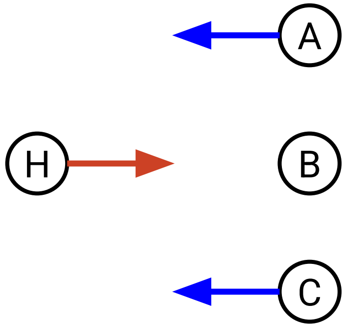
- Norman Abramson had a problem at the University of Hawaii in 1968.
- How do we allow users on other islands to access the central hub computer?

Solution: ALOHANet *(Additive Links Online Hawaii Area)*. Hugely influential.

- Shared wireless medium between all the computers.

ALOHANet used two frequencies:

- Hub transmits on its own frequency.
    - One sender = no collisions.
    - All remote nodes listen to this frequency.
- All remote sites transmit on the other frequency.
    - May collide. Use ALOHA to deal with collisions.
    - Only the hub listens to this frequency.



Sending Ethernet Packets

Lecture 18, CS 168, Spring 2025

Ethernet as LAN Network Protocol

Local Area Networks (LANs) are generally Ethernet.

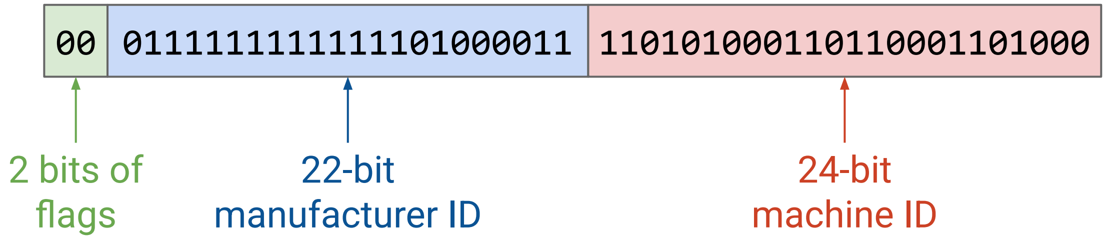
Machines in the same LAN can exchange messages directly at Layer 2!

- No need for IPs, routers, forwarding, etc.
- Analogy: If we're in the same room, we can talk without using the postal system.

Ethernet Addressing

At layer 2, each machine has a **MAC address** (*Media Access Control*).

- Can be used even if you don't have an IP address.
- 48 bits, written in hex, e.g. `f8:ff:c2:2b:36:16`.
- Stored permanently on the machine ("burned in").
  - Often can be overridden by software.
- Allocated according to organization, e.g. manufacturer of the machine.
- Globally unique. You might plug your computer in anywhere.



Beware of endianness.

Types of LAN Communication

Ethernet supports three types of communication:

- **Unicast**: Send a packet to a single recipient.
- **Broadcast**: Send a packet to everyone on the local network.
- **Multicast**: Send a packet to everyone in a specific group.
    - Machines in the local network can join groups.

Types of LAN Communication (1/3) – Unicast

Unicast: Send a packet to a single recipient.

- Destination = the recipient's MAC address.

Recall: On a shared medium, everybody gets the signal.

- When you get a packet, check the destination to see if it's meant for you.
- If not, ignore the packet.

Types of LAN Communication (2/3) – Broadcast

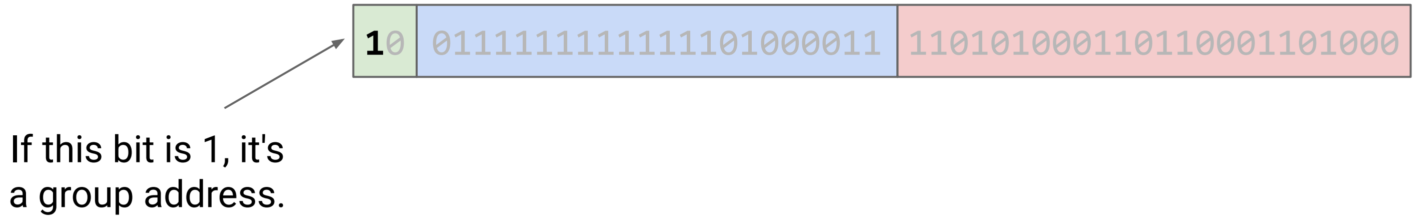
Broadcast: Send a packet to everyone on the local network.

- Packet already reaches everybody on the shared medium.
    - We just need to make sure everyone knows it's meant for them.
- Destination = the broadcast address, `FF:FF:FF:FF:FF:FF`.

Types of LAN Communication (3/3) – Multicast

Multicast: Send a packet to everyone in a specific group.

- Again, everyone gets the packet – need to ensure the group knows it's for them.
- Each group has a *group address*.
  - Individual computer's address: First bit 0.
  - Group address: First bit 1.
- Broadcast is just a special case of multicast, where everyone is in a group.



Beware of endianness.

Types of LAN Communication (3/3) – Multicast

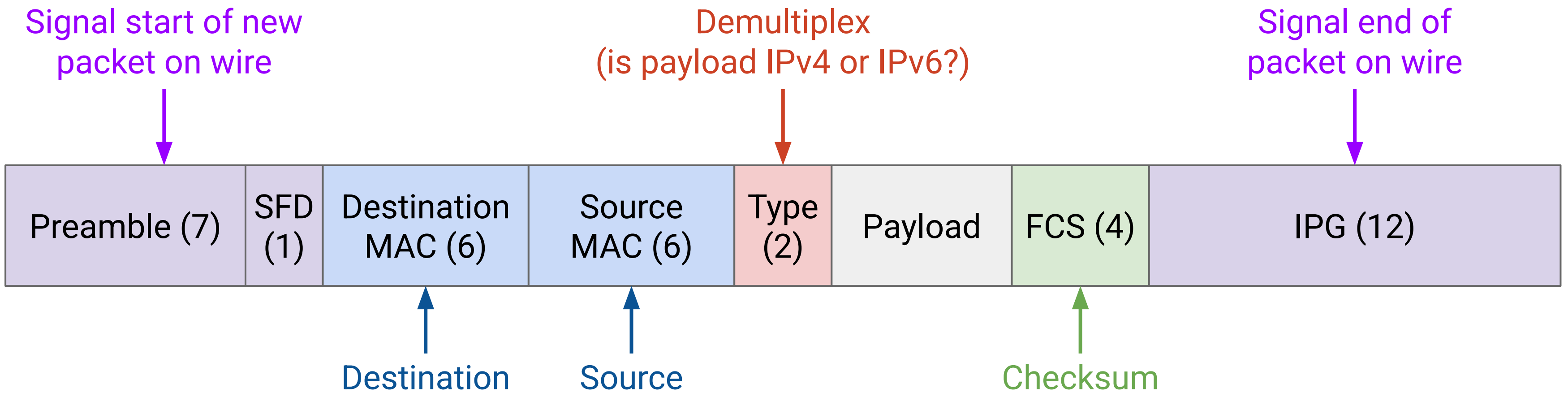
Example of using multicast: Bonjour/mDNS.

- Invented by Apple.
- All Apple products join a multicast group.
- Apple devices can multicast to the group.
  - iPhone says: "Hey, local Apple products, can anyone play music?"
  - Apple TV says: "Hey, local Apple products, I'm an Apple TV."
- Actually uses DNS advertisements that are sent to multicast addresses.
  - Uses specific DNS record types (e.g. SRV) to advertise capabilities.

Ethernet Packet Structure

A data packet in Ethernet is often called a *frame*.

- Many fields (destination, source, type, checksum) similar to IP header.
- Need additional fields to separate packets on the wire.



Layer 2 Networks

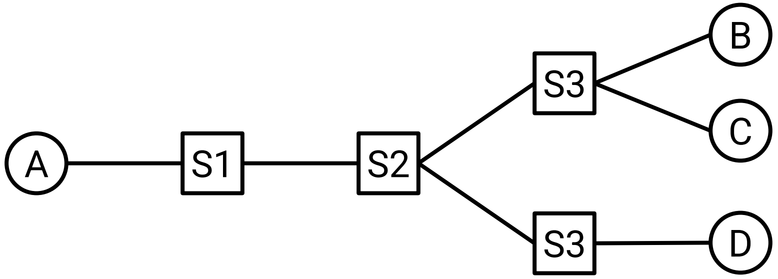
Lecture 18, CS 168, Spring 2025

Layer 2 Networks

We could use more than one wire in a local network.

Switches need to forward packets toward their destination.

- If a switch receives a broadcast packet: Send it to every neighbor.
- Multicast is more complicated. Need to know who's in the group.



Layer 2 Networks

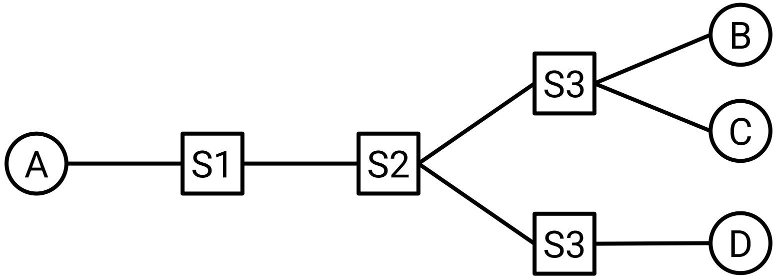
Routing protocols from Layer 3 can also be used at Layer 2.

- Destinations are MAC addresses, instead of IP addresses.

Problem: MAC addresses can't be aggregated.

- Allocated by manufacturer, not geographically.
- This is why Layer 2 can't scale to the Internet.

S2's Table	
Destination	Next Hop
A	R1
B	R3
C	R3
D	R4



Naive Approach: Flooding

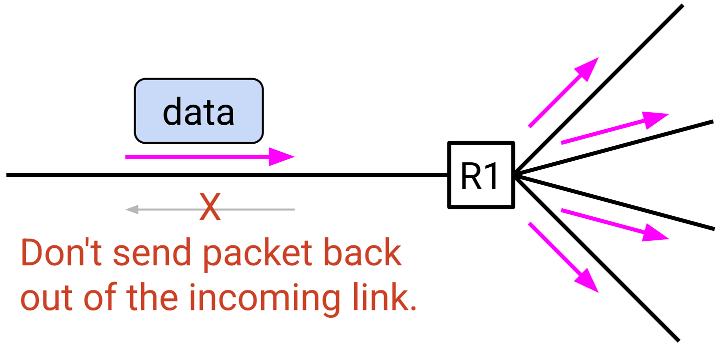
Lecture 18, CS 168, Spring 2025

Naive Approach: Flooding

When you receive a packet, send it out of all ports, except the incoming port.

Will this work?

Yes. The packet will reach everyone, including the destination.

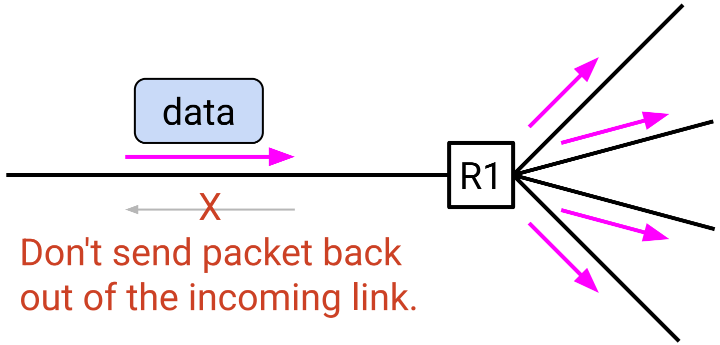


Naive Approach: Flooding

Problems with flooding? Wastes bandwidth in two ways:

1. Packet is sent to other hosts (not the destination).
    - Solution: Use learning switches to build forwarding tables.
2. Loops create broadcast storms.
    - Solution: Use STP to remove loops.

Fixing these 2 problems will result in a good Layer 2 routing protocol.



Learning Switches

Lecture 18, CS 168, Spring 2025

Learning Switches: Motivation

Problems with flooding? Wastes bandwidth in two ways:

1. Packet is sent to other hosts (not the destination).
   - Solution: Use learning switches to build forwarding tables.
2. Loops create broadcast storms.
   - Solution: Use STP to remove loops.

Fixing these 2 problems will result in a good Layer 2 routing protocol.

We could run a routing protocol to fix this, but we'll instead use learning switches, which are simpler.

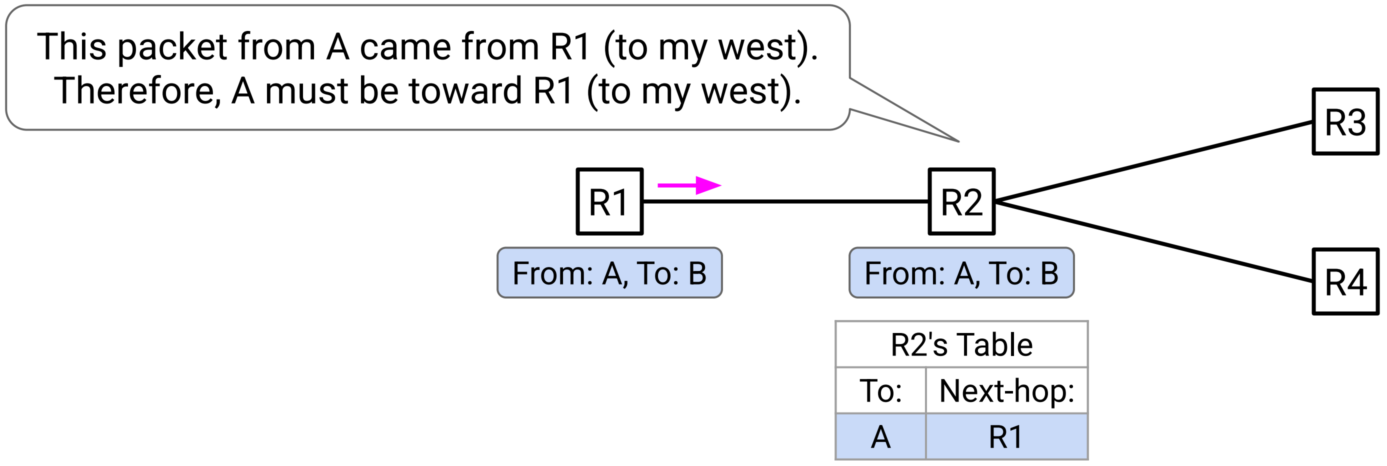
Learning Switches: Routing Process

Key idea: When you receive a data packet, you get a clue about where the **sender** is.

- If a packet from host A comes from the west...
- ...then host A must be to my west!

Use this idea to add a table entry for the **sender**.

- As more packets get sent, more table entries get added.
- Note: The packet only helps you learn about the sender, not the receiver.

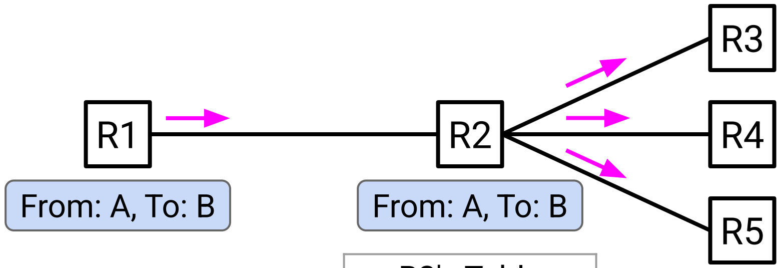

This packet from A came from R1 (to my west).
Therefore, A must be toward R1 (to my west).
R1
R2
R3
R4
From: A, To: B
From: A, To: B

R2's Table	
To:	Next-hop:
A	R1

Learning Switches: Forwarding Process

Forwarding is **destination-based**: Use the destination to decide the next-hop.

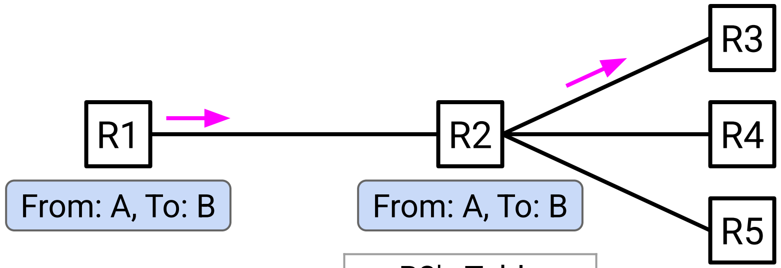
- If the destination exists in the table: Forward to corresponding next-hop.
- If the destination is not in the table: Flood out of all ports (except incoming port).



R2's Table	
To:	Next-hop:

Case 1: No entry for B (destination) in table.

Flood the packet to all ports
(except incoming port).



R2's Table	
To:	Next-hop:
B	R3

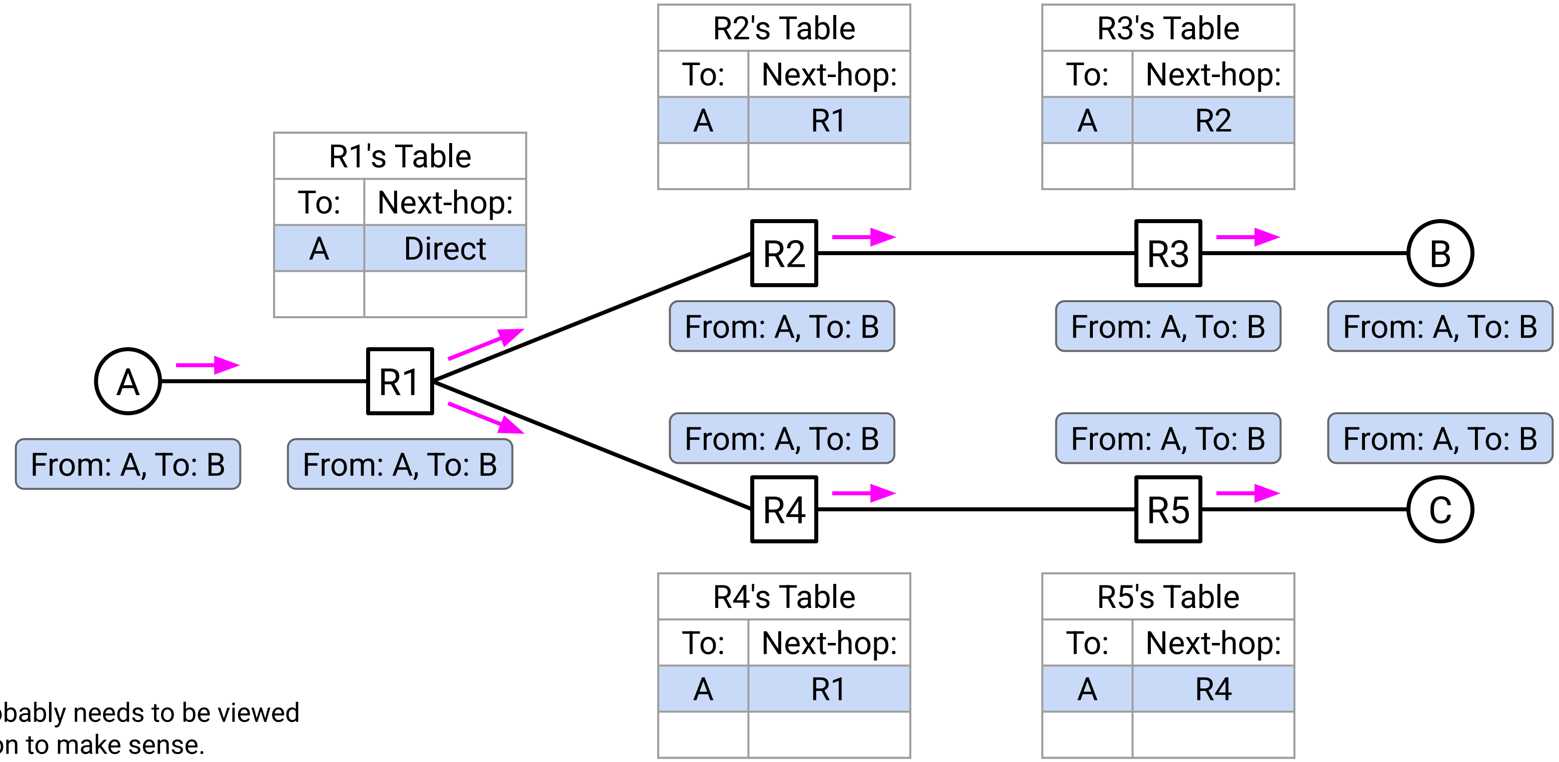
Case 2: Entry for B (destination) is in table.

Use table entry to forward to next-hop.

Learning Switches: Example

All tables start empty. Nobody has an entry for B, so the packet gets flooded.

Along the way, everyone gets a clue about A's location, and adds a table entry for A.



R1's Table	
To:	Next-hop:
A	Direct

R2's Table	
To:	Next-hop:
A	R1

R3's Table	
To:	Next-hop:
A	R2

R4's Table	
To:	Next-hop:
A	R1

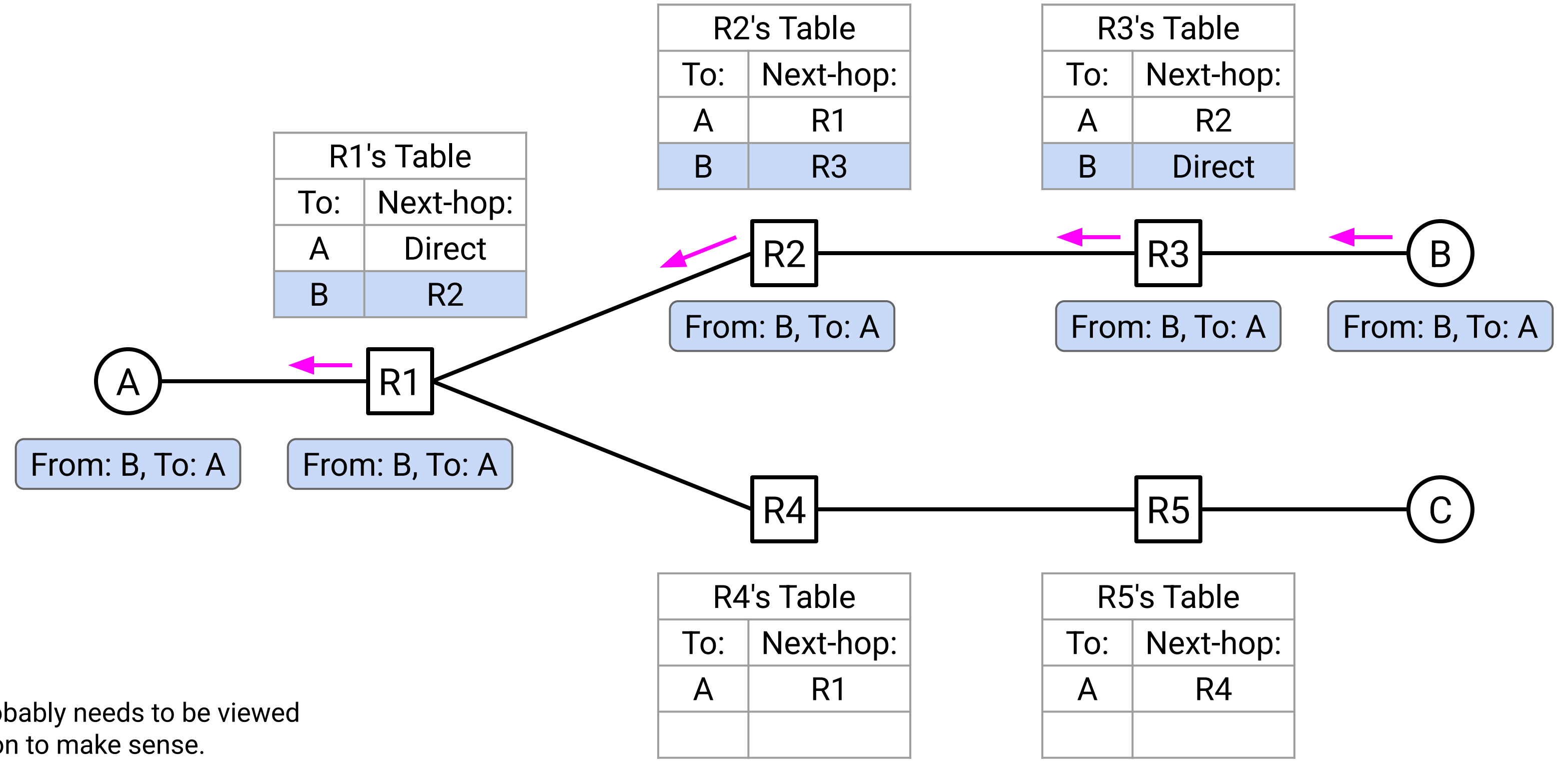
R5's Table	
To:	Next-hop:
A	R4

This slide probably needs to be viewed with animation to make sense.

Learning Switches: Example

This packet can get directly forwarded to A (no flooding) using the forwarding tables.

Routers along the B-to-A path get a clue about B's location, and add a table entry for B.



R1's Table	
To:	Next-hop:
A	Direct
B	R2

R2's Table	
To:	Next-hop:
A	R1
B	R3

R3's Table	
To:	Next-hop:
A	R2
B	Direct

R4's Table	
To:	Next-hop:
A	R1

R5's Table	
To:	Next-hop:
A	R4

This slide probably needs to be viewed with animation to make sense.

Learning Switches: Expiring Routes

Each table entry is associated with a TTL. If the TTL reaches 0, the entry is deleted.

- This allows invalid routes (e.g. link goes down) to eventually expire.

Spanning Tree Protocol (STP)

Lecture 18, CS 168, Spring 2025

Spanning Tree Protocol: Motivation

Problems with flooding? Wastes bandwidth in two ways:

1. Packet is sent to other hosts (not the destination).
    - Solution: Use learning switches to build forwarding tables.
2. Loops create broadcast storms.
    - Solution: Use STP to remove loops.

Fixing these 2 problems will result in a good Layer 2 routing protocol.

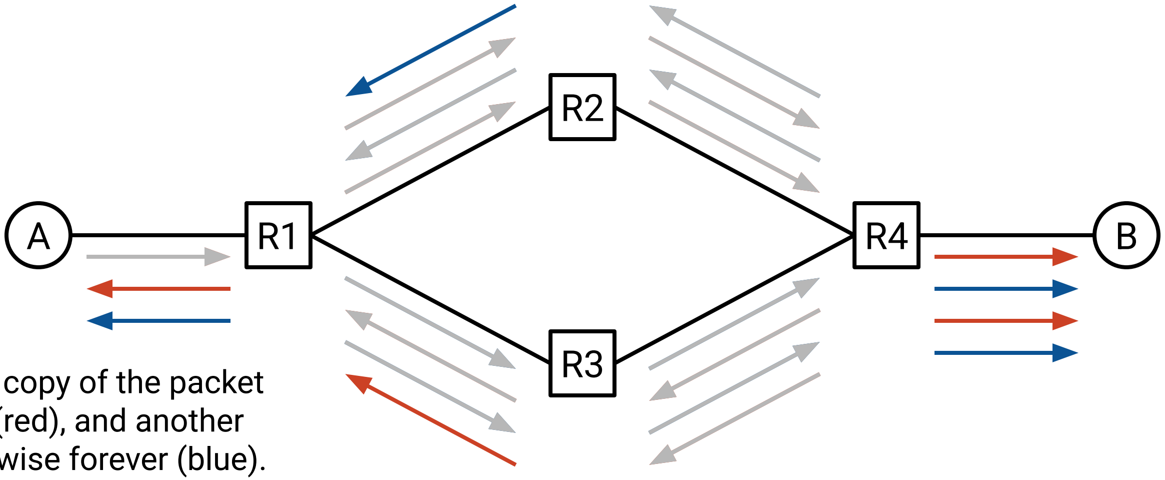
We'll now build a protocol for removing loops.

After the loops are removed, we can use learning switches to build forwarding tables and forward packets.

Spanning Tree Protocol: Motivation

Loops can cause **broadcast storms**: Packets get forwarded around a loop forever.

- Forwarding tables don't help. If the tables are empty, everyone floods the packet.
- Learning switches don't help. We can install entries for the sender, but we still don't know where the receiver is, so the packet still gets flooded forever.
- Not flooding out of the incoming port doesn't help.
  Packets could travel in loops of length 3 or more. (See example below.)



Color is just for clarity. A copy of the packet loops clockwise forever (red), and another copy loops counterclockwise forever (blue).

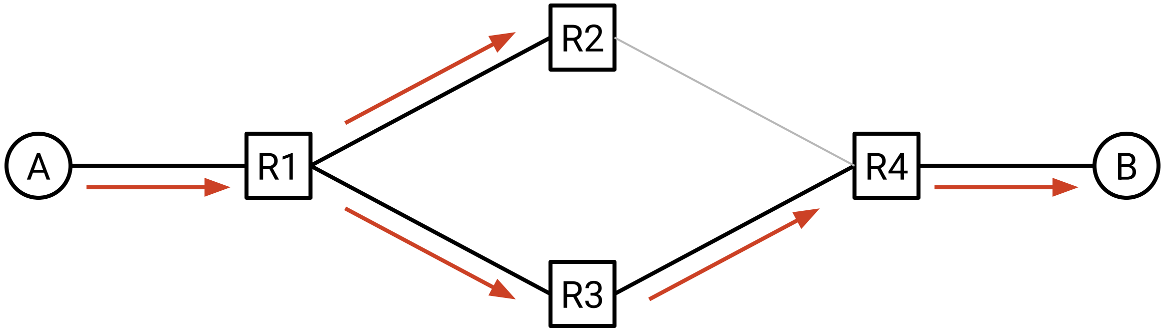
Spanning Tree Protocol: Motivation

Our goal: "Disable" links to remove all the loops.

- We'll now design the **Spanning Tree Protocol (STP)** to achieve this.
- After the loops are removed, we can use learning switches to route packets.

Some other possible solutions, besides STP:

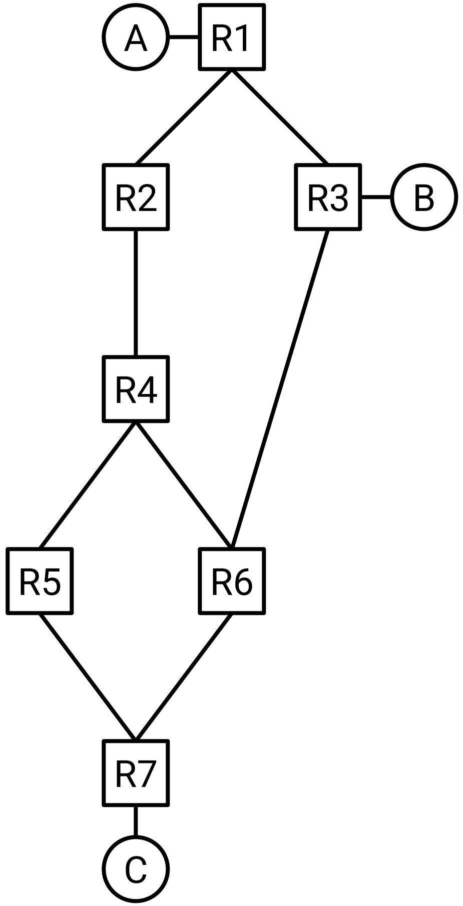
- Drop packets when TTL expires: Won't work, because Ethernet header doesn't have a TTL field.
- Drop packets we've seen before: Won't work, because Ethernet header doesn't have an ID field for identifying duplicates.



STP Ignores Hosts

Hosts don't participate in STP.

Routers will do the job of disabling links to remove loops.



Step 1: Electing a Root

Step 1: Elect one of the routers to be a **root switch**.

- Each router is assigned an ID.
- The root switch is the router with the **lowest ID**.

Each router's ID consists of two values, put together:

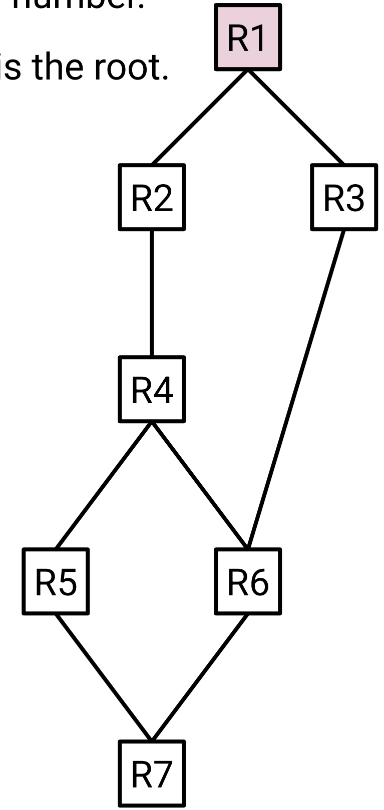
- **Priority**: Manually set by operator.
- **MAC address** of router.
  Used as tiebreaker if two routers have the same priority.

How do we pick a root switch?

- **Manually**: Operator sets a low priority on the chosen root.
- **Default**: All routers have the same default priority value.
  If operator doesn't override the default, the router with the lowest MAC address is the root.

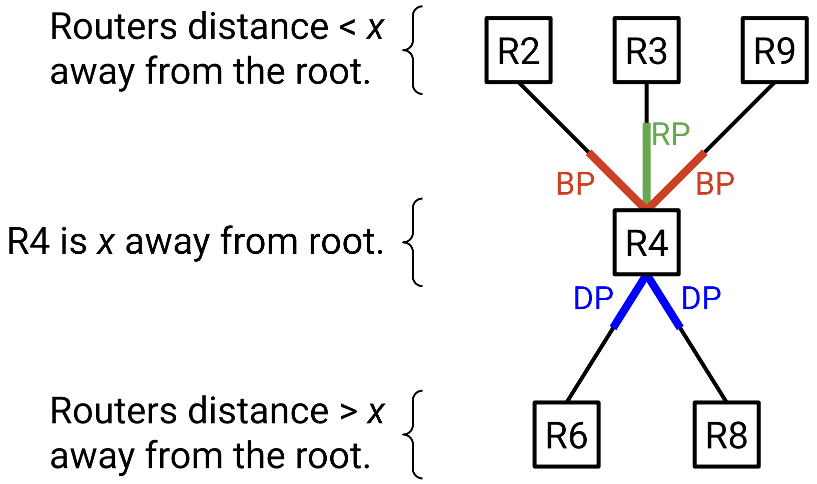
We'll assume the ID is the router number.

Thus, R1 is the root.



Step 2: Labeling Ports

Step 2: Each router labels each of its ports as one of three types:


Routers distance < *x* away from the root.
R4 is *x* away from root.
Routers distance > *x* away from the root.
R2
R3
R9
RP
BP
BP
R4
DP
DP
R6
R8

Out of the ports leading to routers *closer* to the root:

- The **Root Port** is the one along the least-cost path to the root.
- **Blocked Ports** are all the other ports (not along least-cost path).

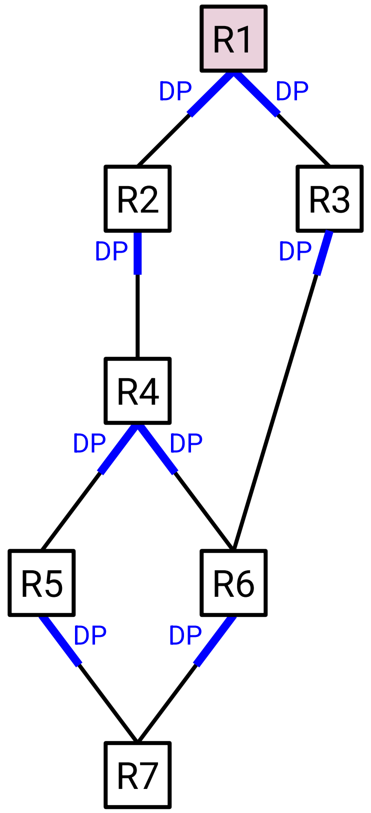
Designated Ports lead to routers *further* from the root.

Port Labels: Designated Ports

Designated Ports lead to routers *further* from the root.

Out of the ports leading to routers *closer* to the root:

- The **Root Port** is the one along the least-cost path to the root.
- **Blocked Ports** are all the other ports (not along least-cost path).

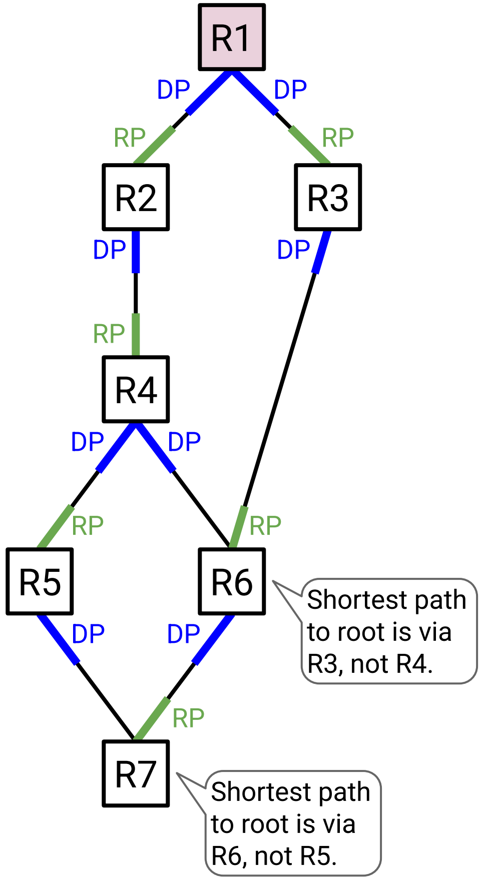


Port Labels: Root Ports

Designated Ports lead to routers *further* from the root.

Out of the ports leading to routers *closer* to the root:

- The **Root Port** is the one along the least-cost path to the root.
- **Blocked Ports** are all the other ports (not along least-cost path).

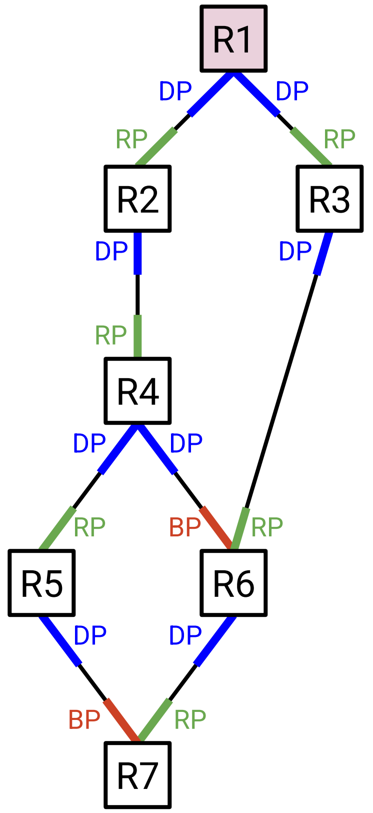

R1
DP
DP
RP
RP
R2
R3
DP
DP
RP
R4
DP
DP
RP
RP
R5
R6
Shortest path to root is via R3, not R4.
DP
DP
RP
R7
Shortest path to root is via R6, not R5.

Port Labels: Blocked Ports

Designated Ports lead to routers *further* from the root.

Out of the ports leading to routers *closer* to the root:

- The **Root Port** is the one along the least-cost path to the root.
- **Blocked Ports** are all the other ports (not along least-cost path).



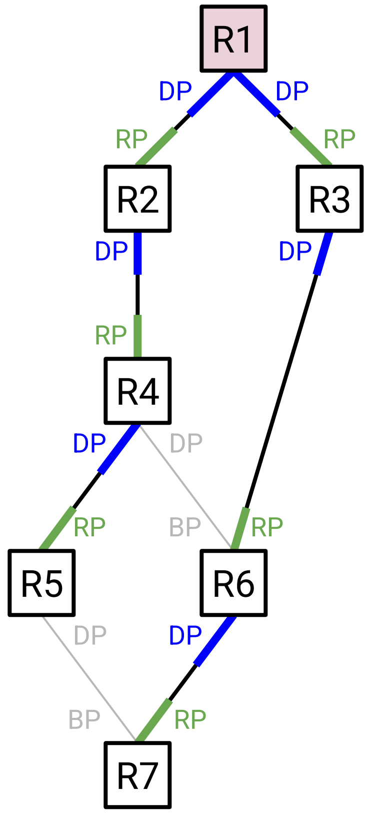
Step 3: Disabling Links

Each router can now disable its **blocked ports**.

- To disable a port: Do not send/receive data packets with that port.

Why does this work?

- **Root port** is your best path toward root.
- **Blocked ports** are worse, redundant paths toward root. Disable these ports.
- **Designated ports** lead away from the root. Other further-away routers might use this port (along their shortest paths to root), so leave these enabled for further-away routers to use.



Step 3: Disabling Links

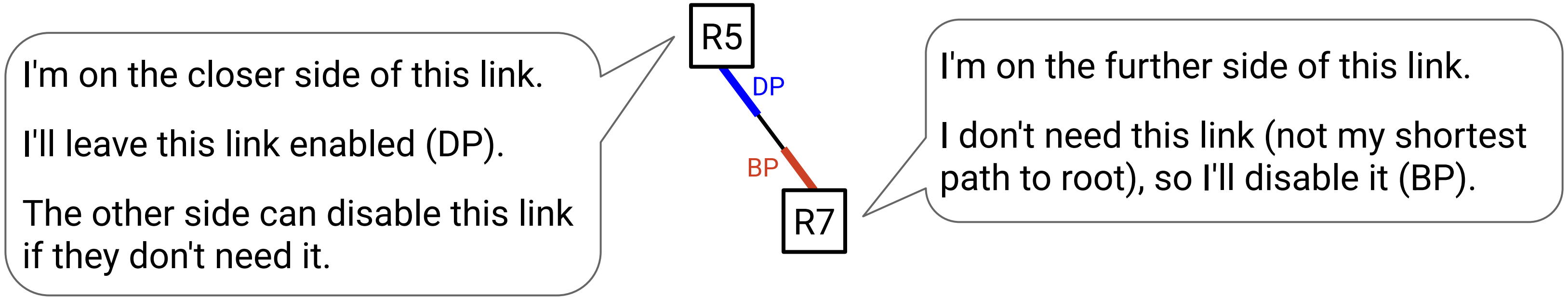
Each link is disabled by only one side.

The further side thinks:

- Is this link along my shortest path to root?
- If yes, enable it (**RP**). If no, disable it (**BP**).

The closer side thinks:

- The further side might need this link (it might be along their shortest path to root). So I'll leave it enabled just in case (**DP**).



I'm on the closer side of this link.

I'll leave this link enabled (DP).

The other side can disable this link if they don't need it.

I'm on the further side of this link.

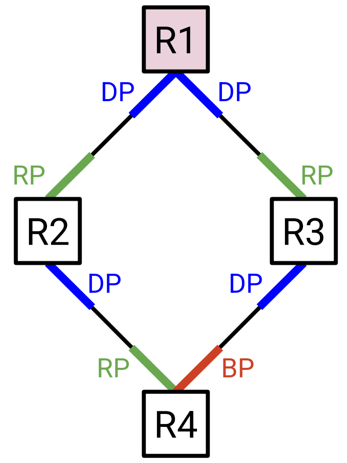
I don't need this link (not my shortest path to root), so I'll disable it (BP).

Breaking Ties

What if there are two shortest paths to the root?

Use router ID to tiebreak.

- Lower ID = shortest path to root (**RP**).
- Higher ID = not shortest path to root (**BP**).



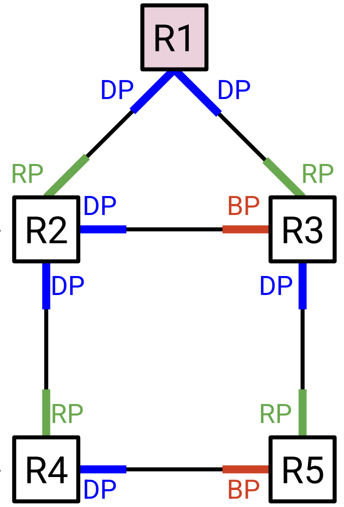
R2 and R3 both yield shortest paths to root.

Tiebreak: Choose R2 (lower ID).

Breaking Ties

If the link leads to a router equally-far from the root as you:

- If their ID is higher, they are "further" (**DP**).
- If their ID is lower, they are "closer" (**RP** or **BP**).



I'm 1 away from root.
R3 is also 2 away from root, but its ID is higher, so R3 is "further" (DP).

I'm 1 away from root.
R2 is also 1 away from root, but its ID is lower, so R2 is "closer" (RP/BP).

I'm 2 away from root.
R5 is also 2 away from root, but its ID is higher, so R5 is "further" (DP).

I'm 2 away from root.
R4 is also 2 away from root, but its ID is lower, so R4 is "closer" (RP/BP).

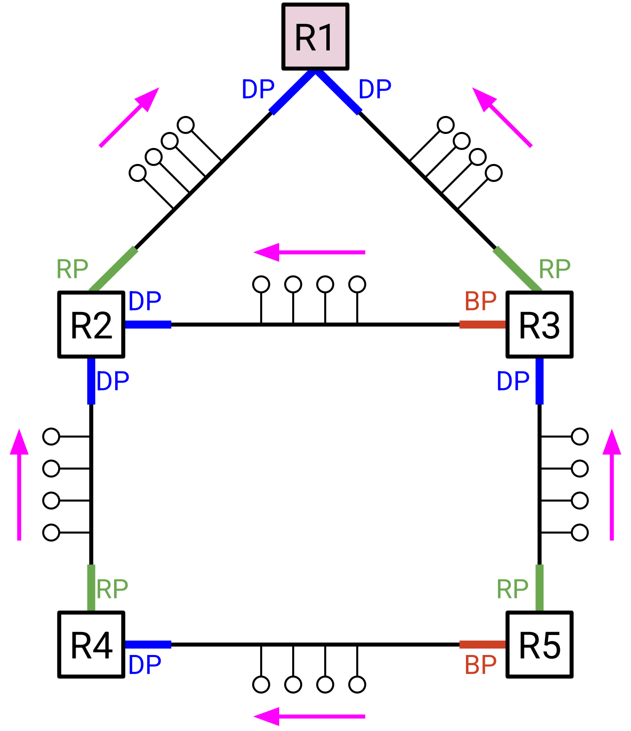
STP With Shared Medium Links

What does "Designated" mean, anyway?

- Remember: A single link could have multiple hosts/routers connected to it (shared medium).
- When hosts on that link send a packet, the packet is received on the **DP**, but not on the **BP**.

This approach helps us avoid loops.

- If the packet was received on both ports, it could take multiple paths to the destination, creating a loop.



Implementing STP: BPDU Exchanges

Lecture 18, CS 168, Spring 2025

Implementing STP with BPDU Exchanges

STP requires global knowledge of the network:

- Who is the root (lowest ID)?
- Does this link lead closer to the root, or further from the root?

To learn this information, routers exchange control-plane messages.

- They have a fancy name: **Bridge Protocol Data Units (BPDUs)**.
- Similar to advertisements in distance-vector.

BPDU Advertisement Rules

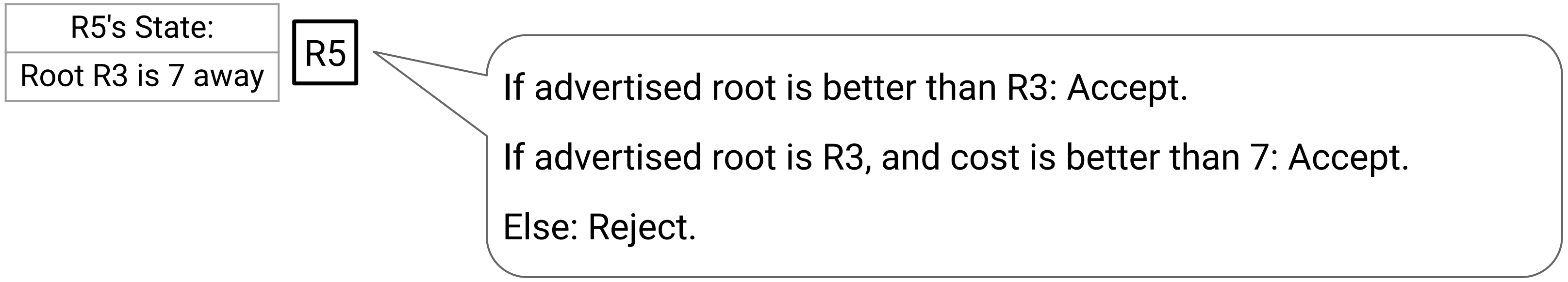
Each router keeps track of two things:

- Who is the best root I know about? *(Recall: "better" = "lower ID".)*
- How far away is this best root?

When you send a BPDU to your neighbors, include these two things.

When you hear a BPDU from your neighbor, accept it and update your state if:

- The advertised root has a lower ID. *(Note: Accept the better root, even if cost is worse.)*
- The advertised root is the same, but the cost to root is better.



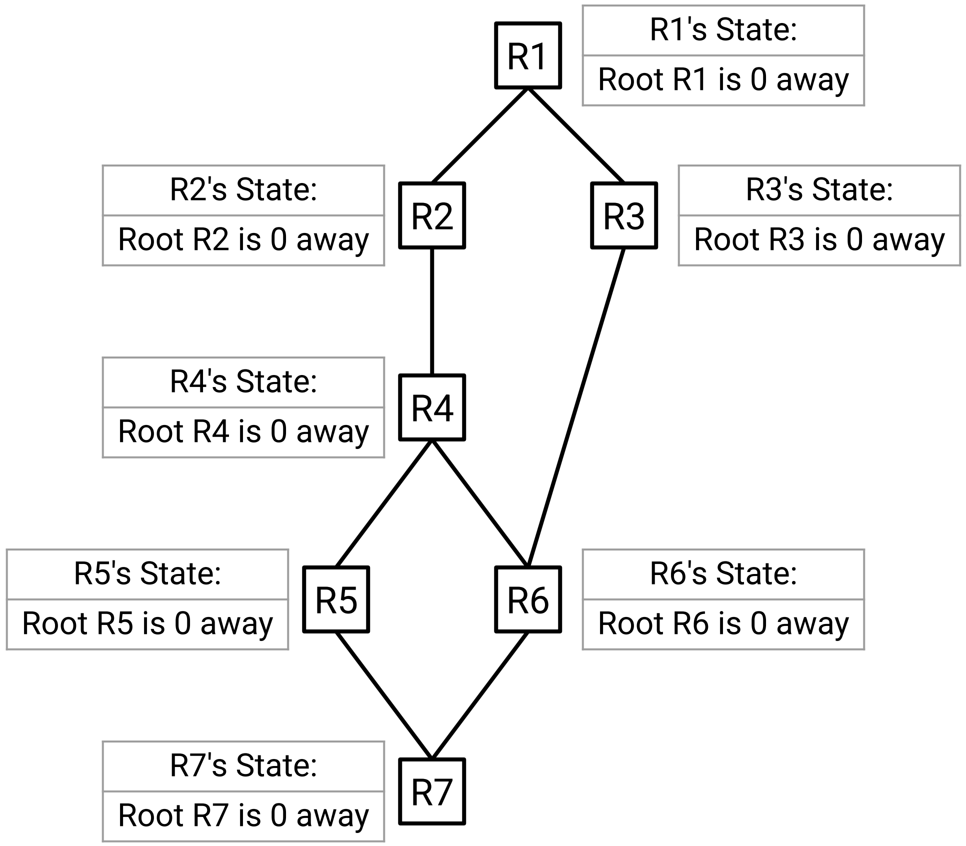
If advertised root is better than R3: Accept.

If advertised root is R3, and cost is better than 7: Accept.

Else: Reject.

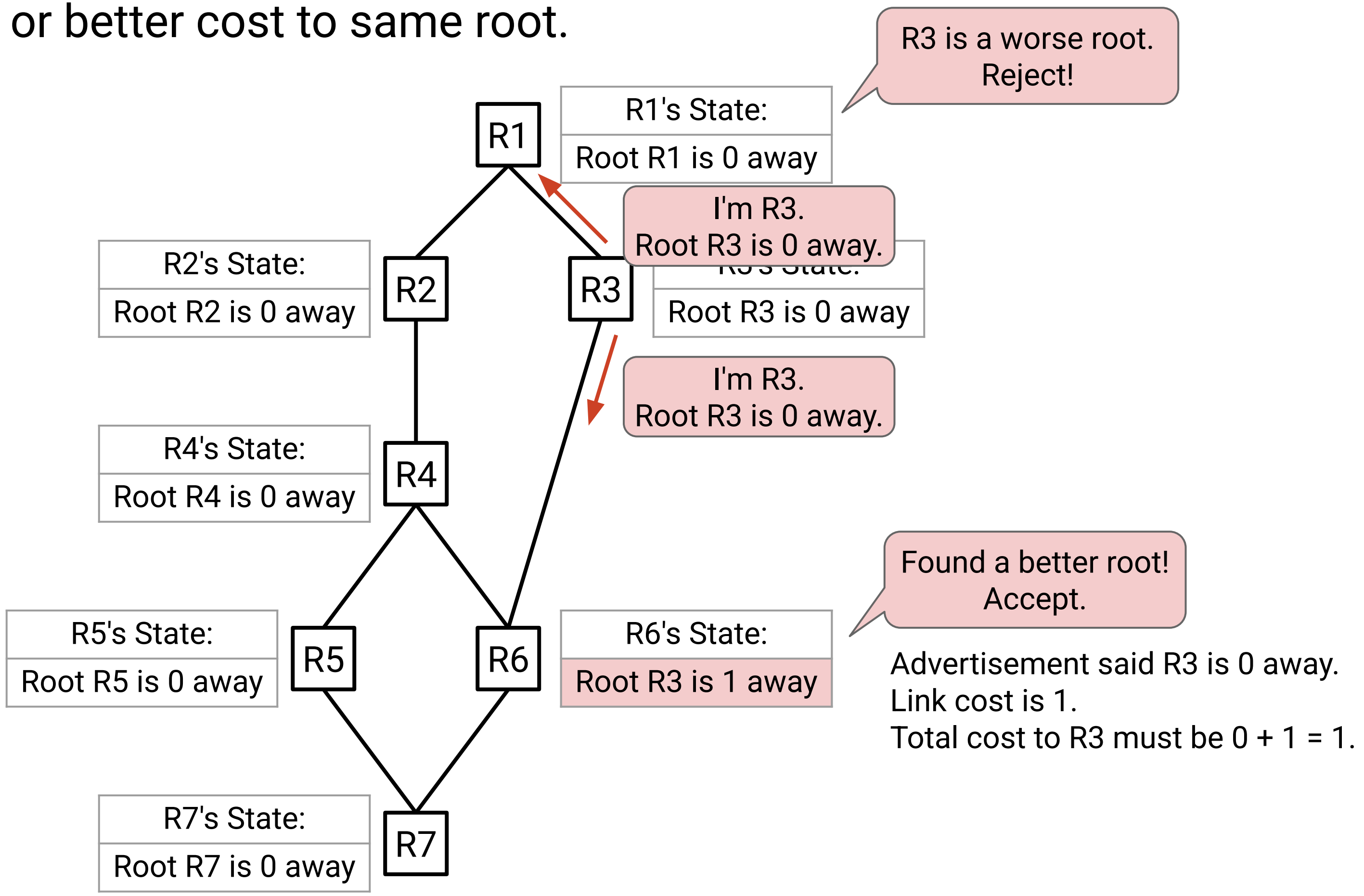
BPDU Exchange Example

Initially, every router only knows about itself, so it thinks the root is itself.



BPDU Exchange Example

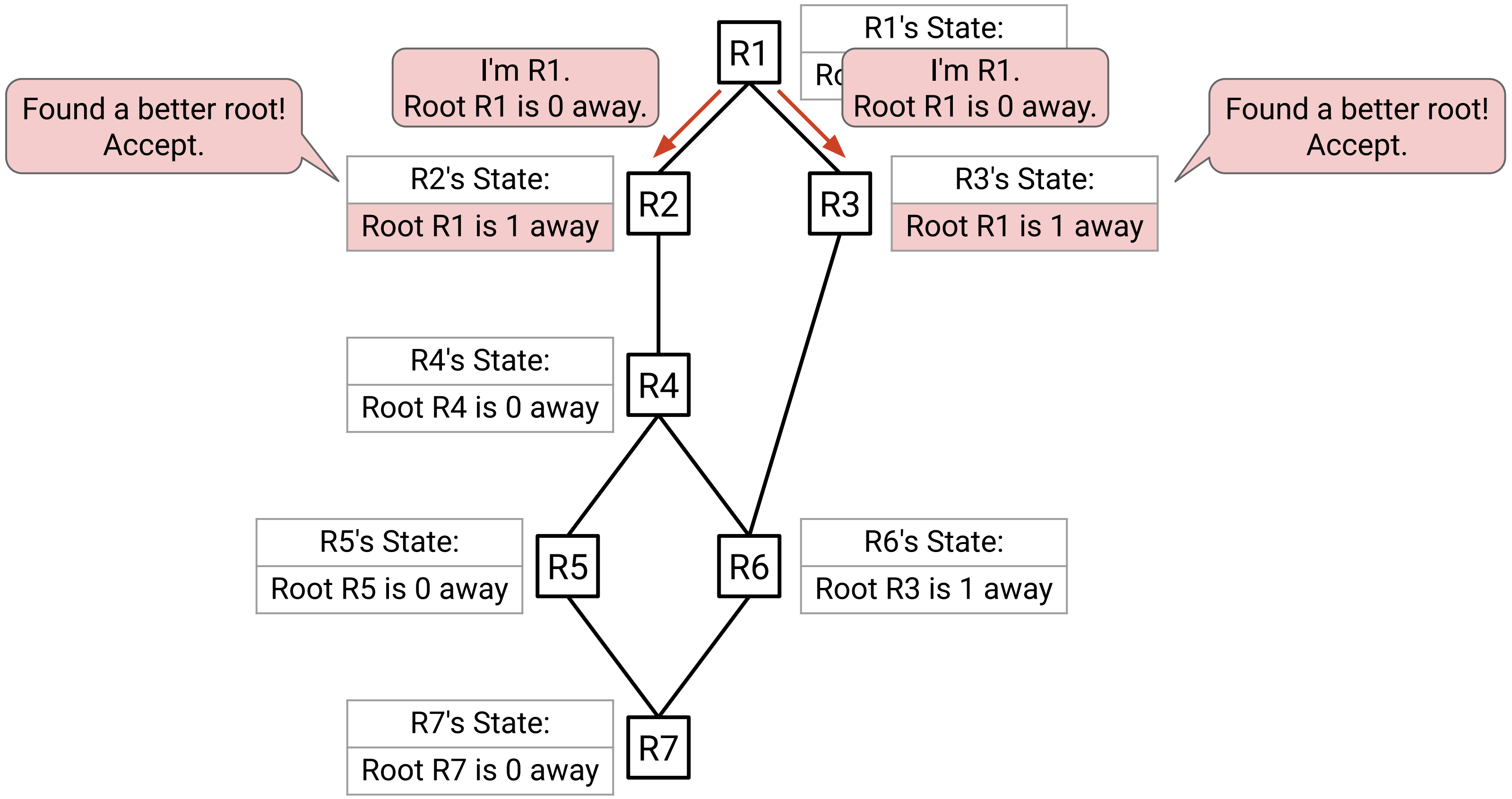
Accept if better root, or better cost to same root.



Advertisement said R3 is 0 away.
Link cost is 1.
Total cost to R3 must be 0 + 1 = 1.

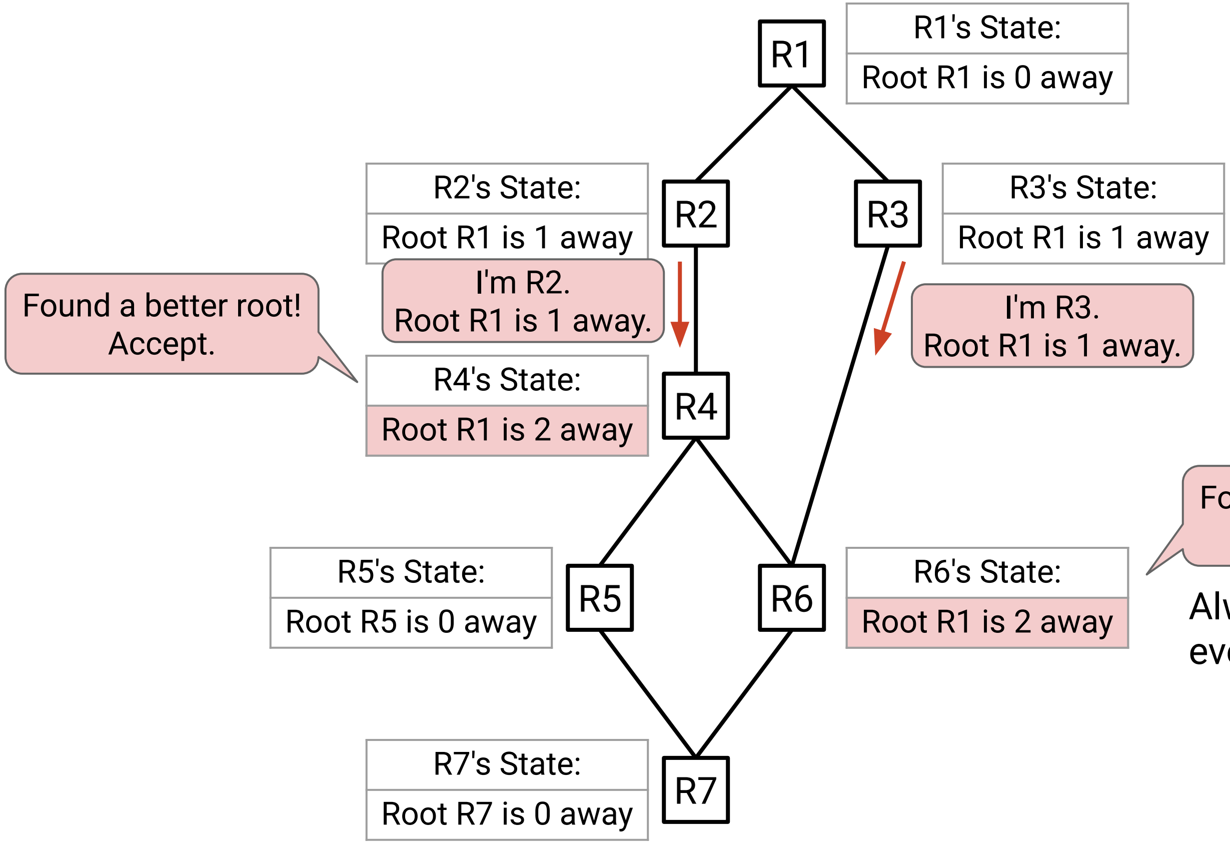
BPDU Exchange Example

Accept if better root, or better cost to same root.



BPDU Exchange Example

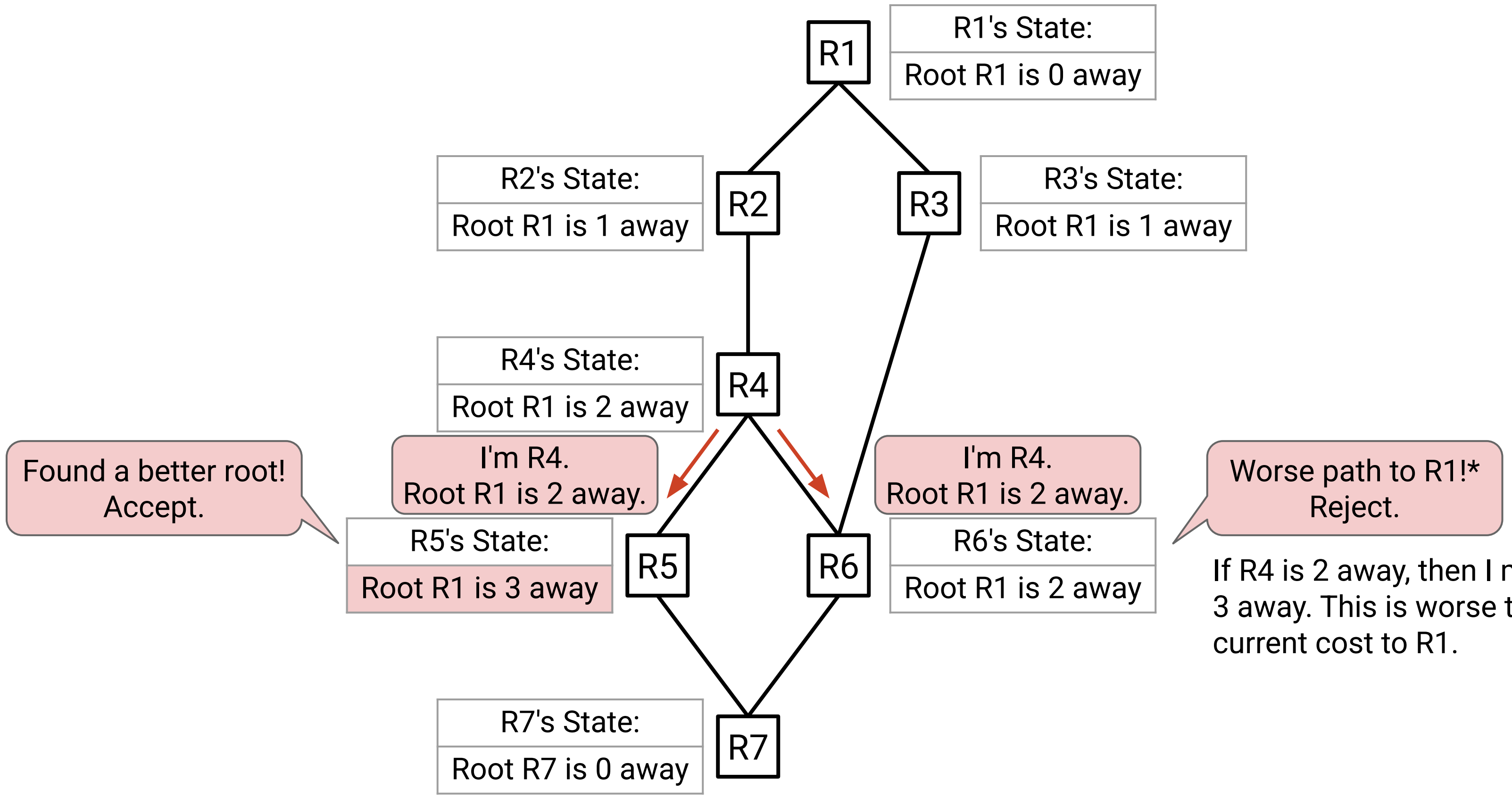
Accept if better root, or better cost to same root.



Always accept the better root, even if the cost is worse.

BPDU Exchange Example

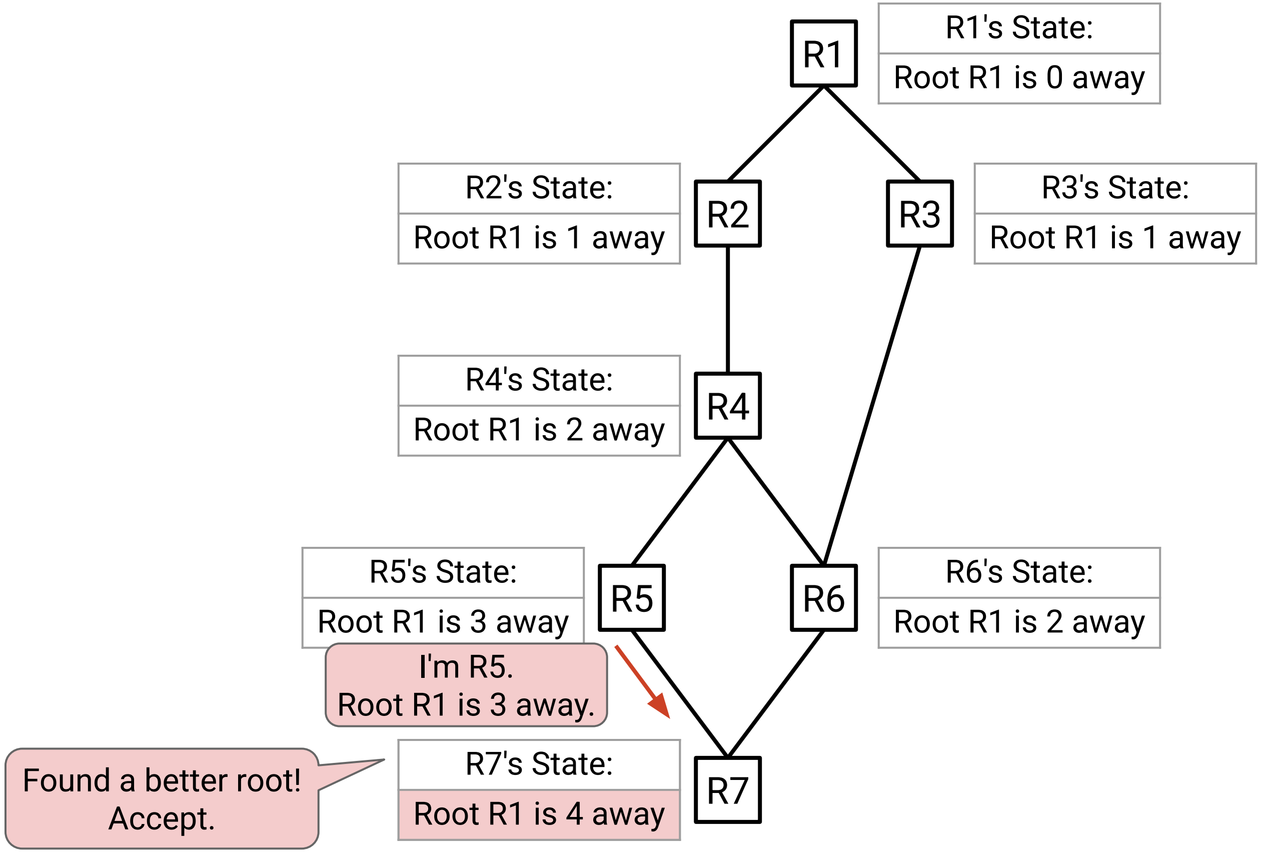
Accept if better root, or better cost to same root.



If R4 is 2 away, then I must be 3 away. This is worse than my current cost to R1.

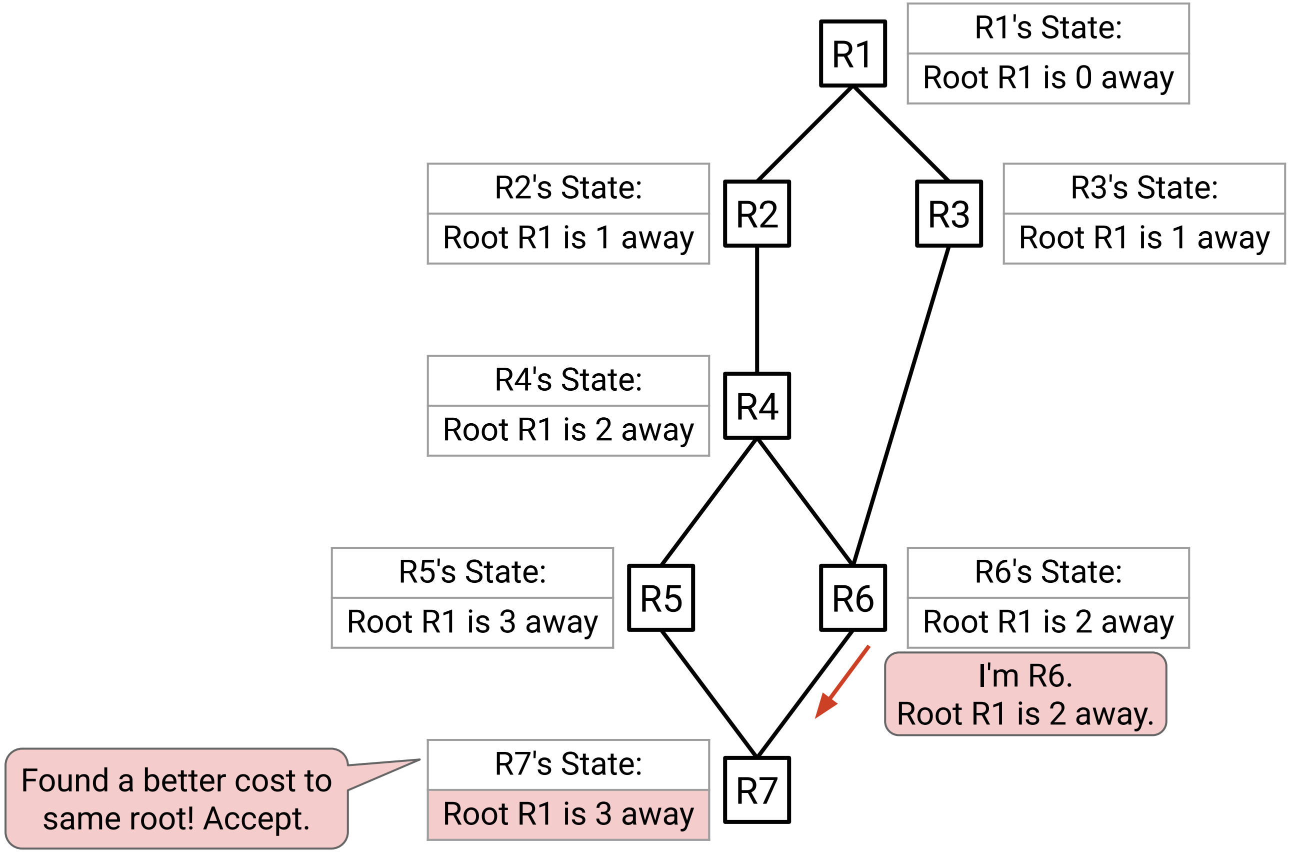
BPDU Exchange Example

Accept if better root, or better cost to same root.



BPDU Exchange Example

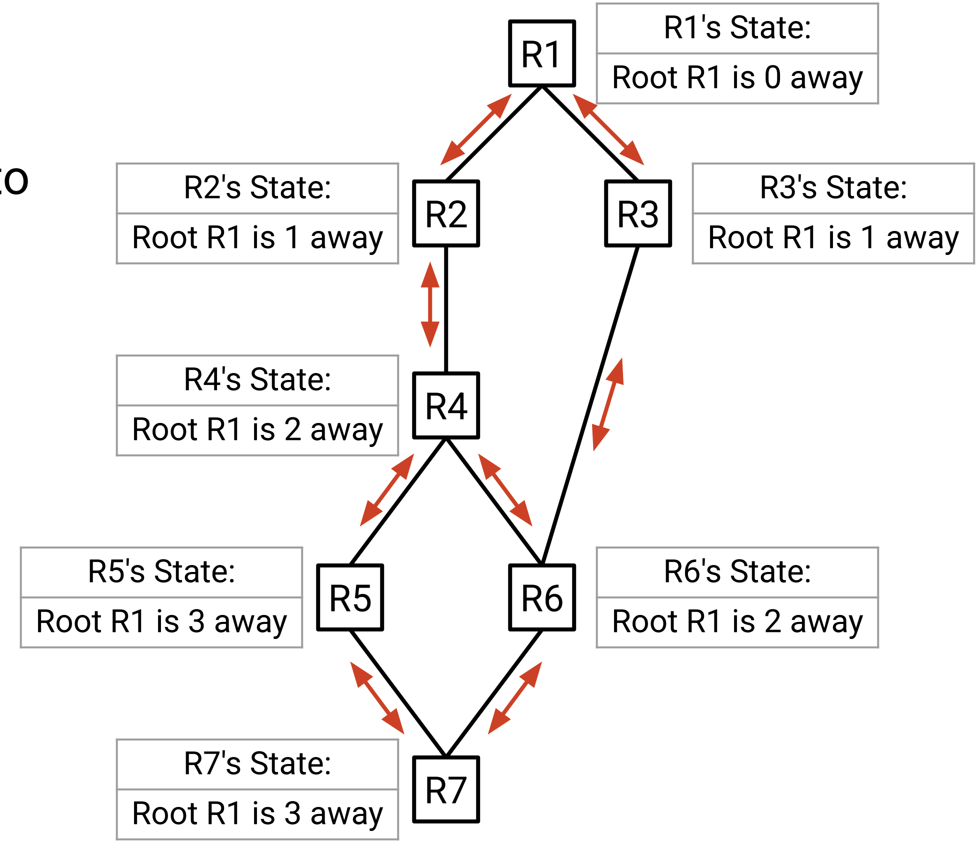
Accept if better root, or better cost to same root.



BPDU Exchange Example

After convergence, periodic advertisements allow you to learn your neighbors' cost-to-root values.

Use your neighbors' cost-to-root values to assign DP, RP, BP.



Why BPDU Exchanges Work

Why does this work?

- The true root has the lowest ID, so advertisements with the true root will always be accepted over worse roots.
- Other exchanges might happen (e.g. "I thought root is R7, now I think it's R3"), but eventually, the true root (e.g. R1) will propagate to everybody.

Spanning Tree Protocol: Recap

Designated Ports lead to routers *further* from the root.

Out of the ports leading to routers *closer* to the root:

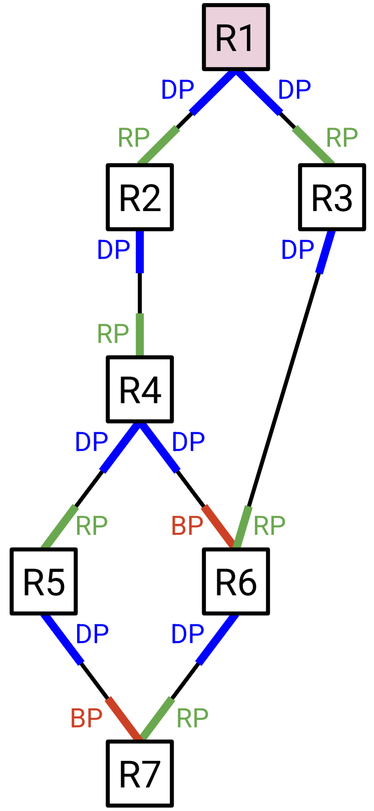
- The **Root Port** is the one along the least-cost path to the root.
- **Blocked Ports** are all the other ports (not along least-cost path). Disabling BPs helps us remove loops.

Routers exchange information in BPDUs (e.g. "Root R1 is 3 away") to help routers agree on the root and label ports as DP/RP/BP.

When you hear a BPDU from your neighbor, accept it and update your state if:

- The advertised root has a lower ID.
  (Note: Accept the better root, even if cost is worse.)
- The advertised root is the same, but the cost to root is better.

Router with lowest ID is elected as the root.


R1
DP
DP
RP
RP
R2
R3
DP
DP
RP
R4
DP
DP
RP
BP
RP
R5
R6
DP
DP
BP
RP
R7