CS 168 Summer 2025

Multicast and Collectives

Discussion 12

1 True or False

1.1 IP multicast is widely deployed across different domains for global Internet communication.

1.2 In the DVMRP protocol, the multicast forwarding table requires one entry per source, per multicast group.

1.3 Core-Based Trees (CBT) guarantee that packets are forwarded along the least-cost paths to all group members.

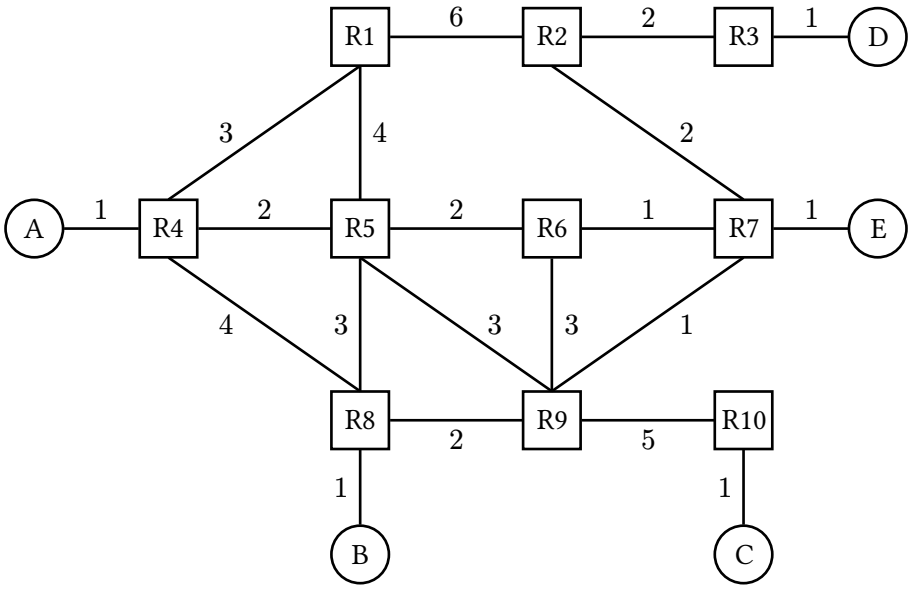
1.4 Overlay multicast requires routers to implement specialized multicast protocols, unlike IP multicast.

1.5 The AllReduce collective operation can be implemented efficiently using a ring topology, where each node sends and receives data to/from its neighbors.

1.6 The stretch factor in overlay multicast measures the ratio of the overlay path cost to the underlay path cost, with lower stretch indicating better performance.

1.7 Collective operations in AI training, such as Broadcast and Reduce, are unrelated and cannot be combined to form other operations like AllReduce.

2 Multicast I



Consider running DVMRP on this topology. Hosts A, B, C are members of group G1.

2.1 D sends a multicast packet to G1. What path does the packet take to reach B?

2.2 Does the multicast path from D to B change if host E joins the group? If so, what is the new path?

The remaining subparts are independent of earlier subparts.

Now, consider running CBT on this topology. R2 is the core, and group G1 initially starts out with no group members.

Note: The join message is unicast toward the root, but if an intermediate router is already on the tree, it does not need to forward the join message any further. (This is because all subsequent routers toward the root will also be on the tree.)

2.3 C joins group G1. What path does the join message take?

2.4 Next, B joins group G1. What path does the join message take?

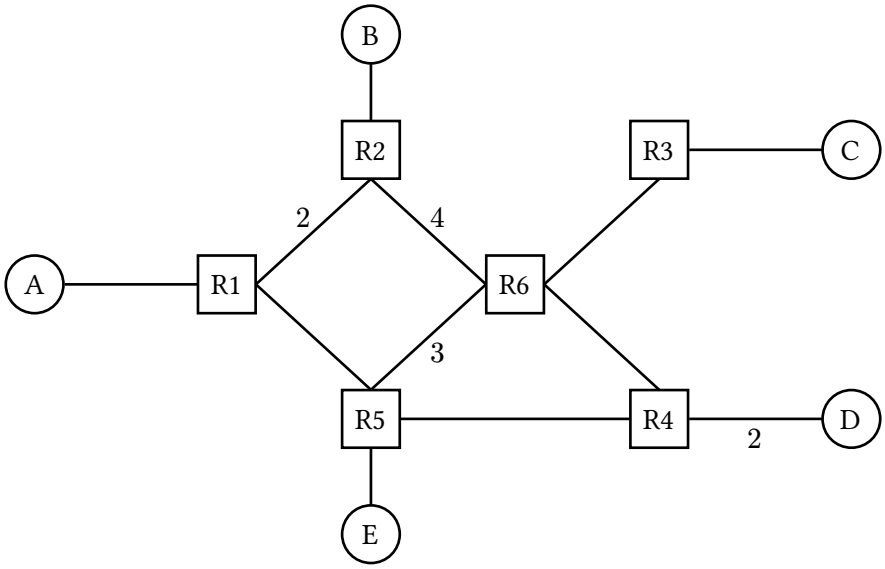
2.5 Next, A joins group G1. What path does the join message take?

2.6 B sends a multicast packet to group G1. What paths does the packet take?

2.7 Suppose that R3 was the core instead. B sends a multicast packet to group G1. What paths does the packet take?

3 Multicast II

Consider the network topology below. Link costs represent packet travel times, in seconds. Ignore packet processing or queuing delays when computing packet travel times.



Consider running DVMRP on this topology. Each subpart continues on from previous subparts.

At the start, A and E are the only members of group G1, and no other groups exist.

3.1 DVMRP runs for some time and reaches a steady-state. At steady-state, all prunable branches of multicast trees are pruned.

A sends a multicast packet to G1. What links are used to send this packet?

3.2 Now, C decides to join group G1. To do this, C runs a host-to-router protocol (e.g. IGMP).

Immediately after the host-to-router protocol reaches steady-state, which routers know that C has joined G1?

3.3 DVMRP runs for some time and again reaches steady-state. At steady-state, all prunable branches of multicast trees are pruned.

Now, E sends a multicast packet to group G1. What links are used to send this packet?

3.4 At this steady-state, is it possible for B to send a message to group G1?

3.5 Now, B sends a multicast to group G1.

What route does the packet take from B to E?

3.6 When B sends a multicast packet to group G1, how long does it take for the packet to reach all G1 members?

3.7 Now, suppose that hosts A, B, D join a new group, G2. (Remember that A, C, E are the members of G1.)

DVMRP runs for some time and again reaches steady-state. At steady-state, all prunable branches of multicast trees are pruned.

Fill out the multicast forwarding table at router R6.

From:	**To:**	**Children:**

3.8 Now, these multicast packets get sent:

- A sends a packet to G1.
- A sends a packet to G2.
- B sends a packet to G2.
- D sends a packet to G1.

Who is the last host to receive any of the four packets?

3.9 Who is the first host to receive any of the four packets?

Now, consider running CBT instead on the same topology. The remainder of this question is independent of all earlier subparts. Each subpart continues on from previous subparts.

At the start, G1 has no members. R1 is selected as the core for group G1.

3.10 C sends a Join message for group G1. What routers forward this request?

3.11 Now, D sends a Join message for group G1. What routers forward this request?

Note: The join message is unicast toward the root, but if an intermediate router is already on the tree, it does not need to forward the join message any further. (This is because all subsequent routers toward the root will also be on the tree.)

3.12 Now, B sends a multicast packet to G1. How long does the packet take to reach all group members?

3.13 Now, C sends a multicast packet to G1. How long does the packet take to reach all group members?

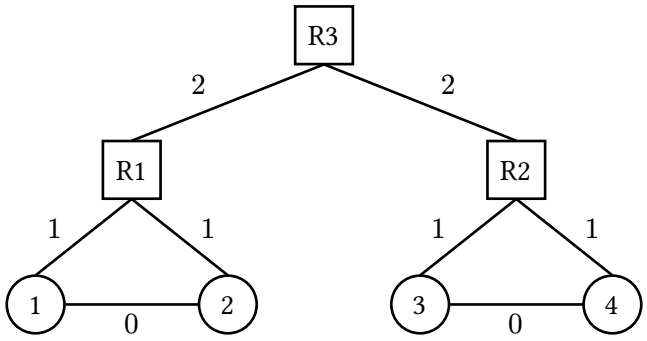
3.14 Now, the A—R1 link goes down. Who can still send messages to G1?

3.15 Which G1 members will still receive messages sent to G1?

4 Collectives

Consider the **underlay** topology below.

- Link costs represent packet travel time, in seconds.
- Ignore any processing and queuing delays.
- Assume links have very high bandwidth, i.e. packet travel time is the only source of delay.



4.1 In this topology, some links are labeled with cost 0. Why might this be a reasonable model for an AI training datacenter network?

We want to perform an AllReduce operation on nodes 1, 2, 3, 4. Each node has a 4-Gbit vector, and each vector is split into four 1-Gbit vector elements.

4.2 Consider using a **full mesh** overlay topology to run the AllReduce operation.

What does the **overlay** topology look like? Draw virtual links in the diagram below. Then, label each virtual link with its packet travel time (according to the underlay network).

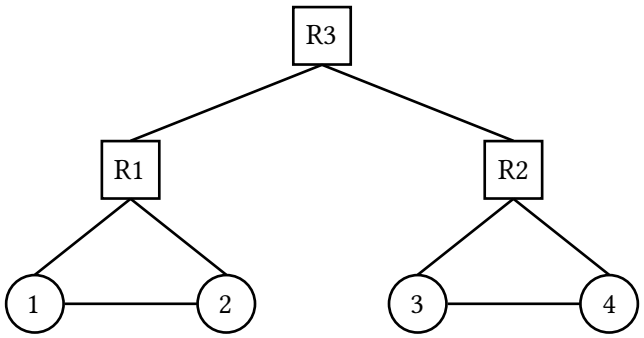
4.3 To execute AllReduce with the full-mesh overlay, how much data is sent by each node? How much data is sent in total?

4.4 How many seconds are needed to execute AllReduce with the full-mesh overlay?

4.5 To execute AllReduce with the full-mesh overlay, how much data is sent across each link (in either direction)?

Label each link with the total amount of data sent across that link (in either direction).

Assume that all data is unicast.



4.6 Next, consider using a **ring** overlay topology to run the AllReduce operation. You can assume that we ignore ring optimization, so each node sends its entire vector to the next, instead of one element at a time.

What does the **overlay** topology look like? Draw virtual links in the diagram below. Then, label each virtual link with its packet travel time (according to the underlay network).

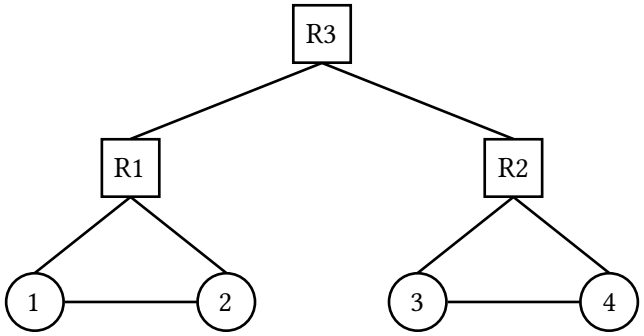
For subparts 4.7 and 4.8: Assume Node 4 is the first to send data, and the ring works clockwise. This means that Node 4 sends its data to Node 3, who calculates the sum of its vector with Node 4's vector, and passes it on, so on and so forth.

4.7 How many seconds are needed to execute AllReduce with the ring-based overlay? Assume that we start at Node 4, and work clockwise.

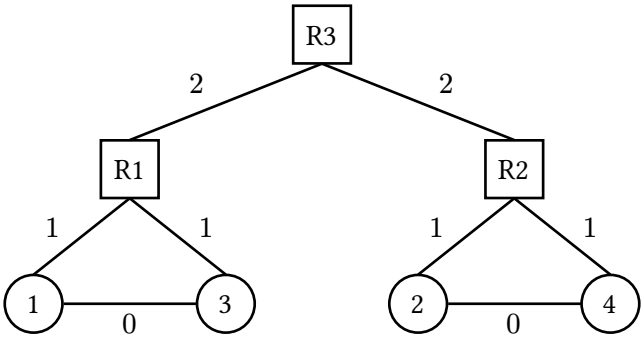
4.8 To execute AllReduce with the ring-based overlay, how much data is sent across each link (in either direction)?

Label each link with the total amount of data sent across that link (in either direction).

Assume that all data is unicast.



Next, suppose we relabel the nodes in the **underlay** topology. (Recall that the operator can decide what numbers to assign to each node, and the operator's choice of numbering can affect the performance of the resulting collective operation.)



4.9 Consider using a **ring** overlay topology again to run AllReduce, but with the re-numbered nodes shown above. Again, you can assume that we ignore ring optimization, so each node sends its entire vector to the next, instead of one element at a time.

What does the **overlay** topology look like? Draw virtual links in the diagram below. Then, label each virtual link with its packet travel time (according to the underlay network).

For subparts 4.10 and 4.11: Assume Node 4 is the first to send data, and the ring works clockwise. This means that Node 4 sends its data to Node 3, who calculates the sum of its vector with Node 4′s vector, and passes it on, so on and so forth.

4.10 How many seconds are needed to execute AllReduce with the ring-based overlay (and the re-numbered nodes)?

4.11 To execute AllReduce with the ring-based overlay, how much data is sent across each link (in either direction)?

Label each link with the total amount of data sent across that link (in either direction).

Assume that all data is unicast.