BOND: Unifying Mobile Networks with Named Data

Michael Meisel

Ph.D. Dissertation Defense
March 16, 2011

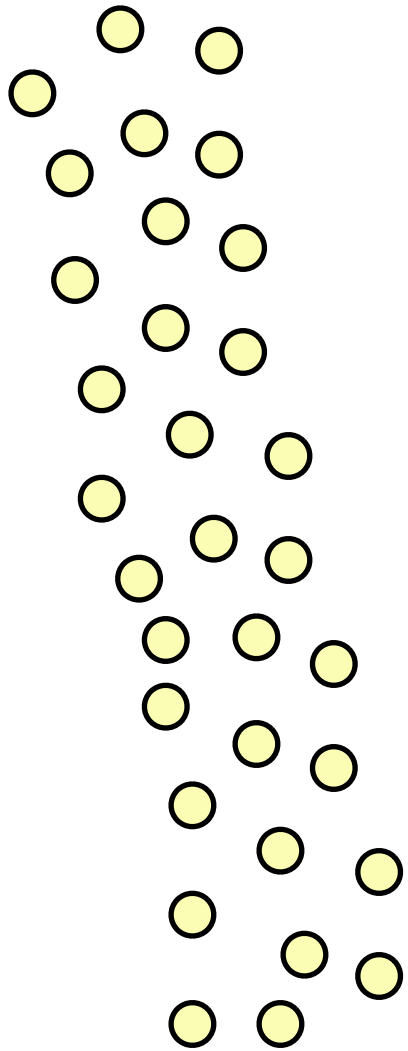
Freeform Wireless Networks

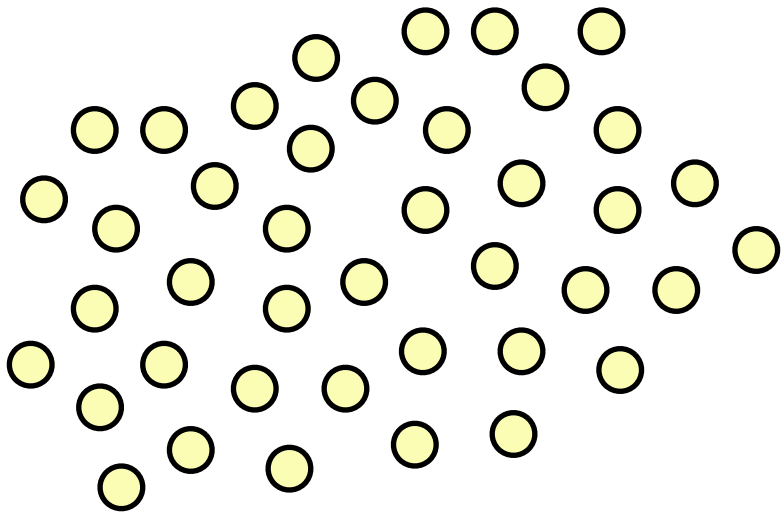
- Multi-hop
- Unpredictable mobility
- Can be connected or disconnected
- Examples: MANETs, VANETs, disruption-tolerant networks, combinations thereof

Goals

- Follow the Named Data Networking (NDN) philosophy
  - Delivery based on “what”, not “where”
- Get rid of holdovers from the wired domain
- One architecture that will work on *any* freeform network
  - Stop treating connected and disconnected networks separately

Connected or Disconnected?





Current Approaches to Freeform Networks

The Wired Approach

1. Each node is assigned an IP address
2. Applications communicate using destination IPs
3. The routing protocol finds a single best path from source to destination
4. At each hop along the path, the sender determines which single node (based on step 3) is allowed to forward the data

Designed > 30 years ago for stationary networks

Issues with The Wired Approach

- Applications care about data, not location
- In freeform networks:
  - IP addresses lose topological meaning, aggregatability
  - Finding, maintaining hop-by-hop paths is expensive
  - Pre-determined paths don’t take advantage of the broadcast nature of wireless

Alternative: Opportunistic Routing

- Goal: improve throughput in stationary mesh networks with lossy links
- Basic approach:
  - Track the quality of every link in the network
  - Senders broadcast, receivers make forwarding decisions based on the quality of their path to the source
- Examples: ExOR [1], MORE [2]

[1] S. Biswas and R. Morris. ExOR: opportunistic multi-hop routing for wireless networks. ACM SIGCOMM Computer Communication Review, 35(4):144, 2005.
[2] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading structure for randomness in wireless opportunistic routing. In SIGCOMM '07, pages 169–180. ACM, 2007.

Alternative: Opportunistic Routing

- Improvements:
  - Takes advantage of the broadcast nature of wireless
- Shortcomings:
  - Cannot handle mobility
  - Still dependent on IP addressing, location-based delivery

Disconnected Network Routing

- Goal: in a disconnected network with unpredictable mobility, increase the probability that data reaches its destination
- Basic approach:
  - Replicate each data packet across many nodes
  - Do not try to figure out where the destination node is
- Examples: Epidemic routing [3], Spray and Wait [4]

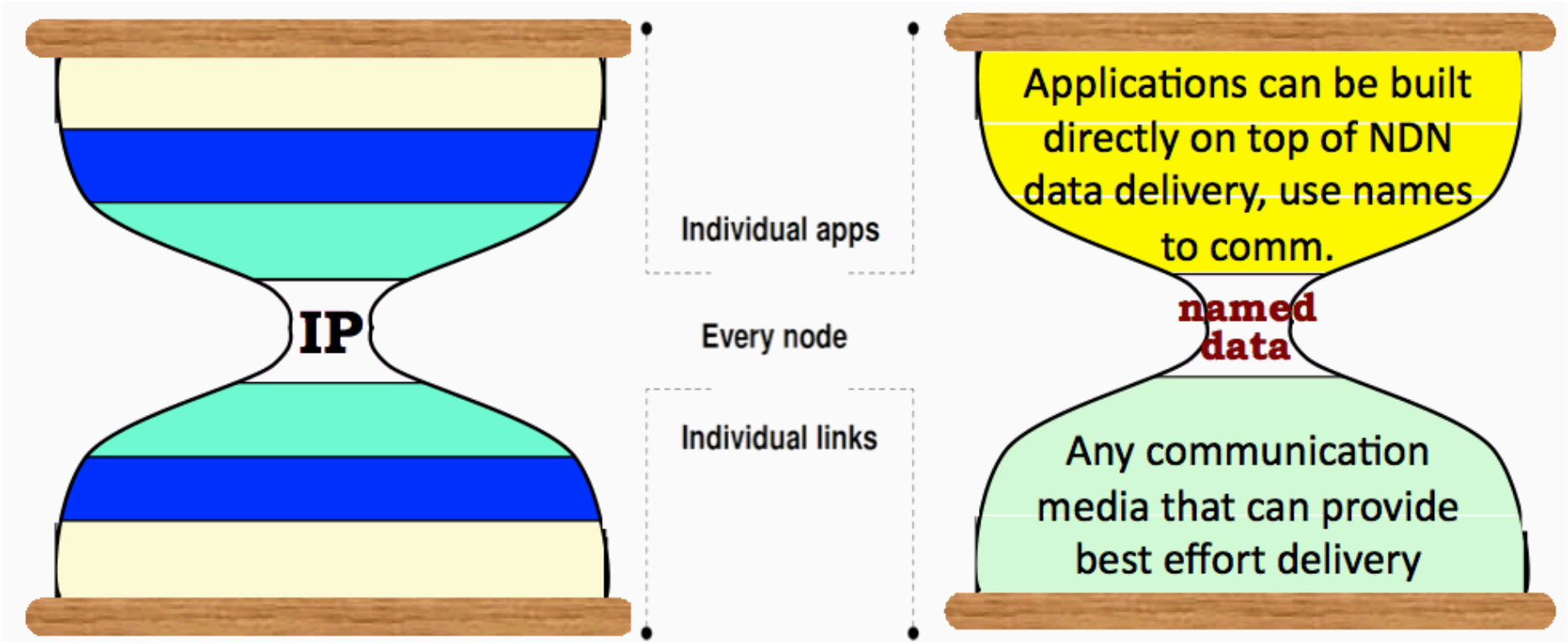
[3] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. Technical Report CS-2000-06, Duke University, 2000.
[4] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In WDTN: SIGCOMM Workshop on Delay-Tolerant Networking, 2005.

Disconnected Network Routing

- Improvements:
  - No pre-determined paths
- Shortcomings:
  - Inefficient for connected networks (or network portions)
  - Still dependent on location-based delivery

Named Data Networking (NDN)

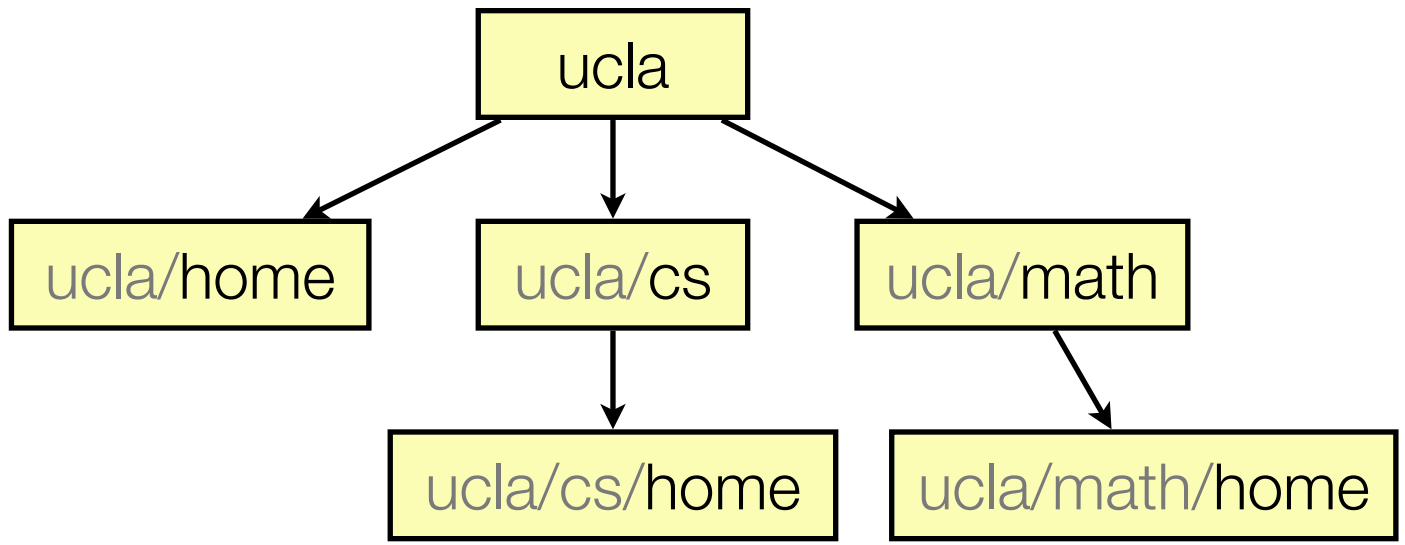
NDN Architecture



- Routing/forwarding is based on data names instead of node addresses

NDN Names

- Data names in NDN are hierarchical



- NDN routing/forwarding can use name prefixes like IP routing uses IP address prefixes

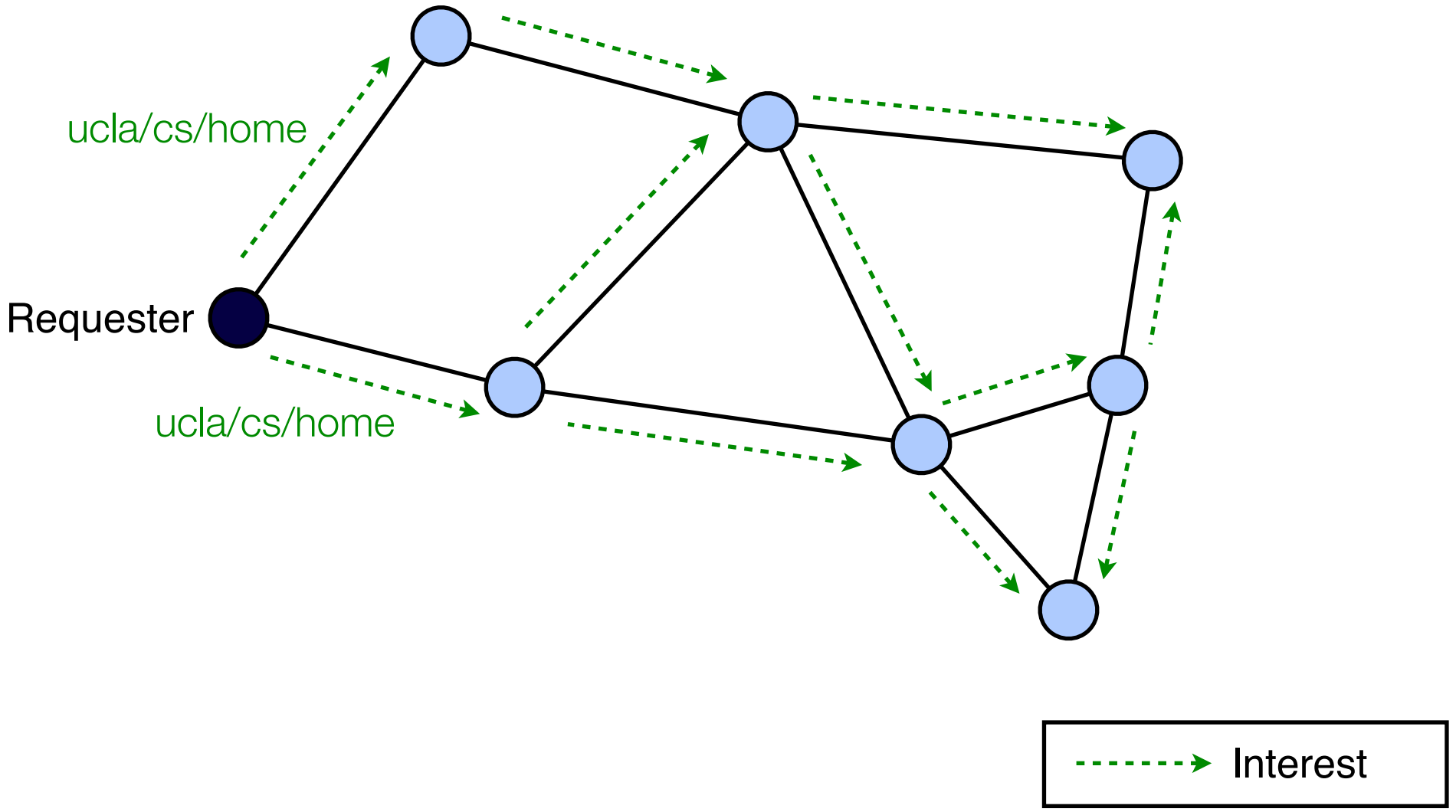
NDN Communication

- ***(Optional Step 0:** Use a routing protocol to announce names or name prefixes)*

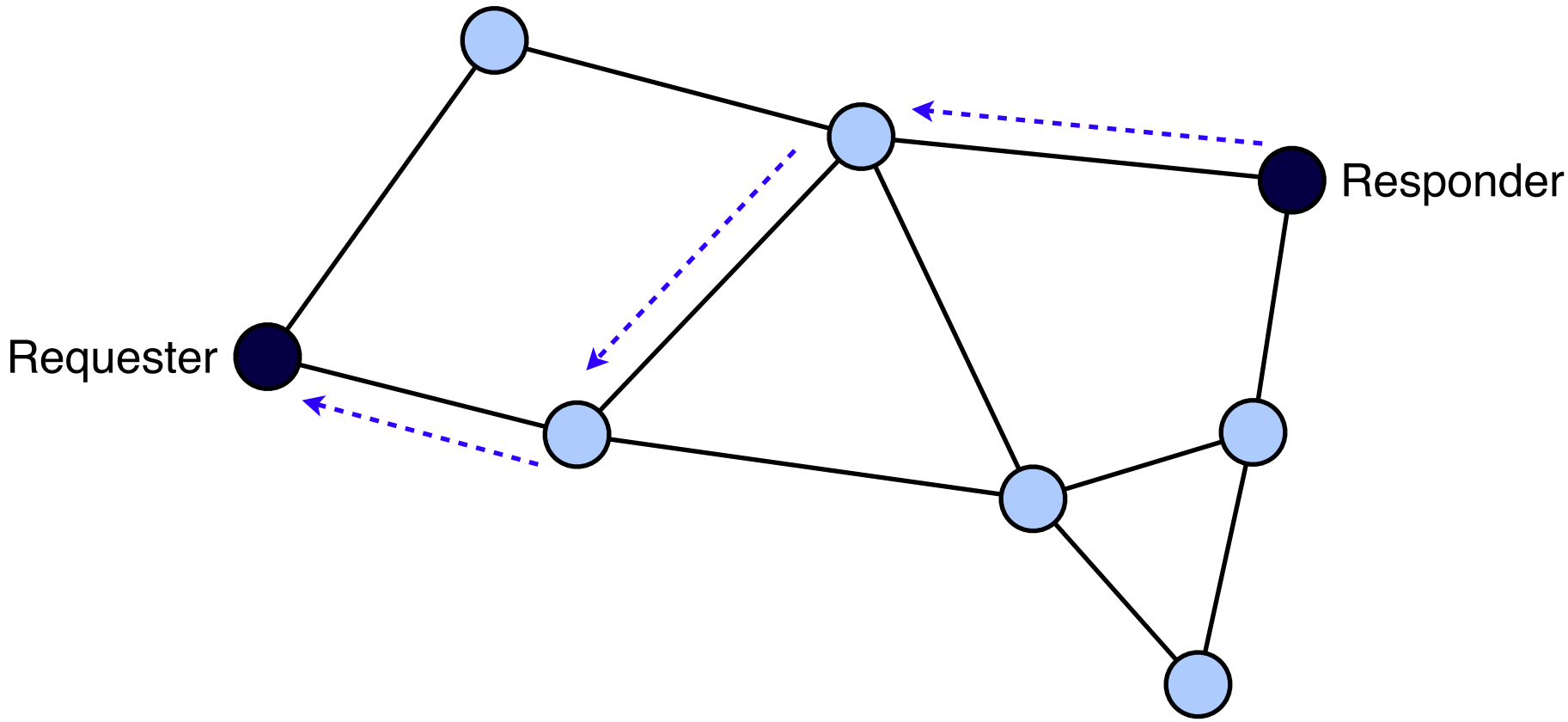
- **Step 1:** An application sends an *Interest* packet containing a request for data by name. It can be flooded or routed.

- **Step 2:** Any node that has the data can send a *Data* packet back towards the source of the Interest. Intermediate nodes cache the data.

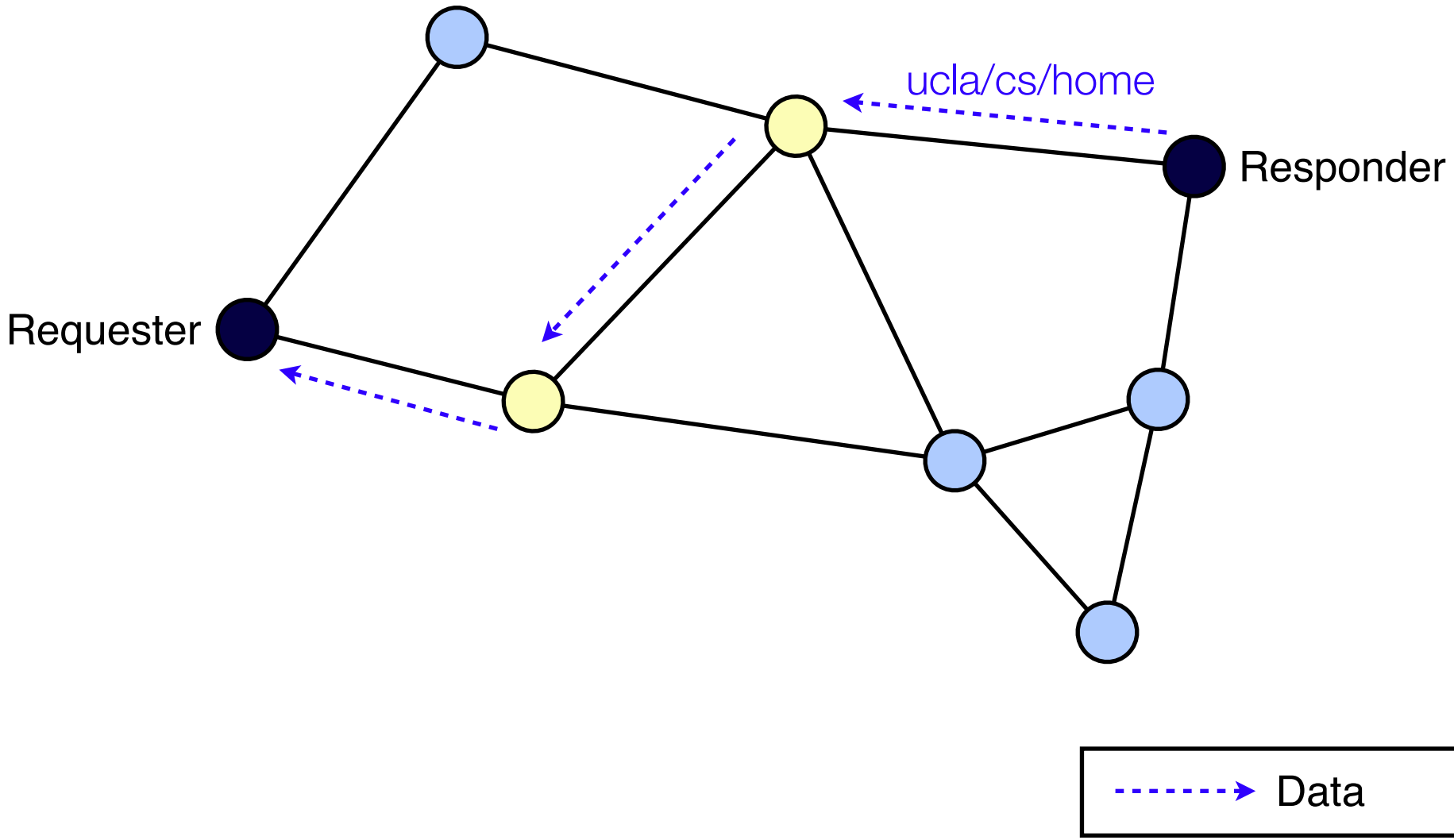
- Future Interests for the same name can be serviced by caches



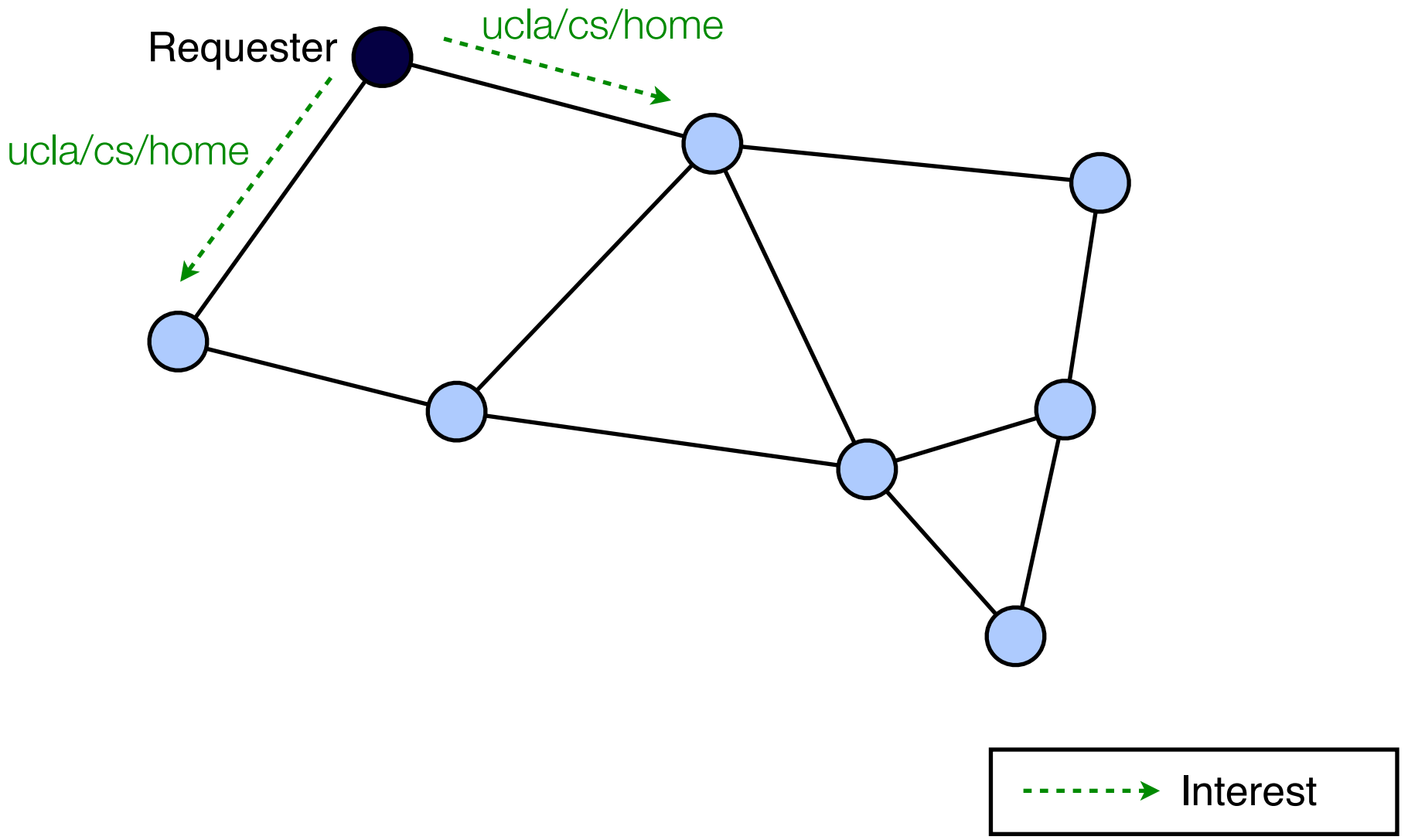
Assume we flood Interests.



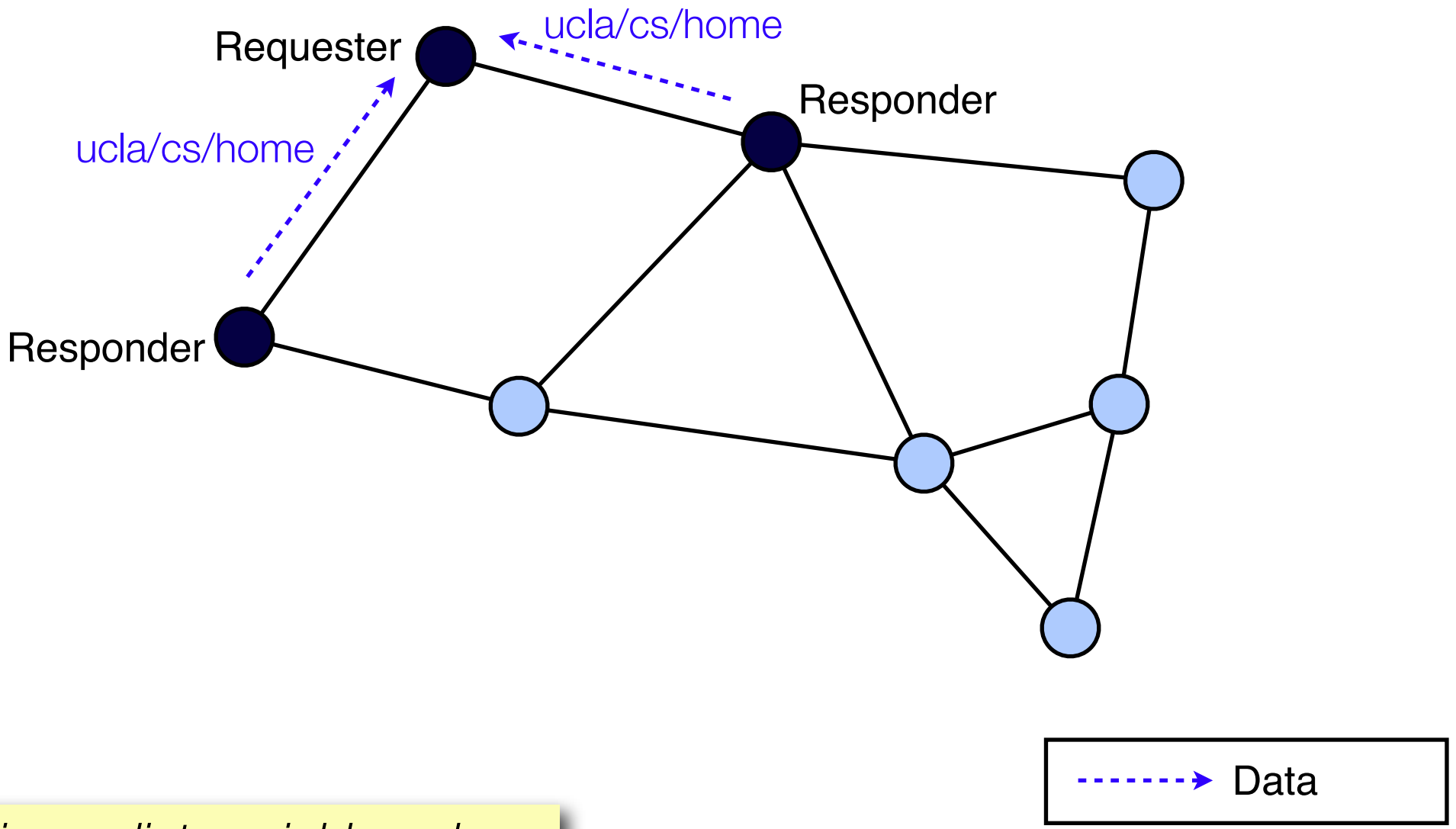
By forwarding the Interests, the intermediate nodes have established a path from any potential responder to the requester.



The nodes that forward the data also cache it.



Suppose another node requests the same data name.



Its immediate neighbors have cached the appropriate data, so they can respond.

NDN's Potential for Freeform Networks

- Applications could communicate based on data names only, retrieving the closest available copy of any data
- Unlike IP addresses, data names don't lose any meaning in the face of mobility
- All nodes could cache data they receive, respond to requests for that data

The BOND Protocol

BOND: Broadcast-Only Named Data

- Applies NDN concepts to freeform networks
- Completely does away with the IP approach:
  - Name-based forwarding in place of IP addresses
  - No control packets or pre-determined paths
  - Broadcast-only transmission at the MAC layer -- forwarding decisions made by the receiver
- Supports both connected and disconnected networks efficiently

Overview

- BOND in connected networks
- BOND in disconnected networks
- Unifying connected and disconnected networks

BOND Names

- BOND exclusively uses names for forwarding
- Two types of names:
  - **data name:** hierarchical data name for a single piece of data
  - **node name:** names a particular node, used in getting data back to the requester
- Data names can be prefix-matched

BOND Communication: Requests

- **Request** packets are sent by a node to solicit a **response** packet from the network, which will contain the requested data
- At first, requests are flooded (like flooded NDN Interests)
- All nodes learn the location of the requester's node name
- Requests contain the desired data name
- Any number of responders may respond with a data packet

BOND Communication: Responses

- Responses take the best *available* path to the requester *on the fly*
- Responders can advertise a name prefix in the response
- Nodes overhearing the response:
  - Cache the data
  - Learn a location of the data name and name prefix
- Future requests for data in the same prefix aren't flooded -- they take the best available path to a responder

Broadcast-Only Forwarding

- Forwarding decisions must be made by the receiver
- **Step 1:** Determine if I can make forward progress. If so:
- **Step 2:** Listen for some time to see if another node closer to the intended destination forwards the packet. If not:
- **Step 3:** Forward the packet

Follow-up Questions

- How does a receiver know how close it is to the destination name?

- How long should a receiver listen, waiting for someone else to forward?

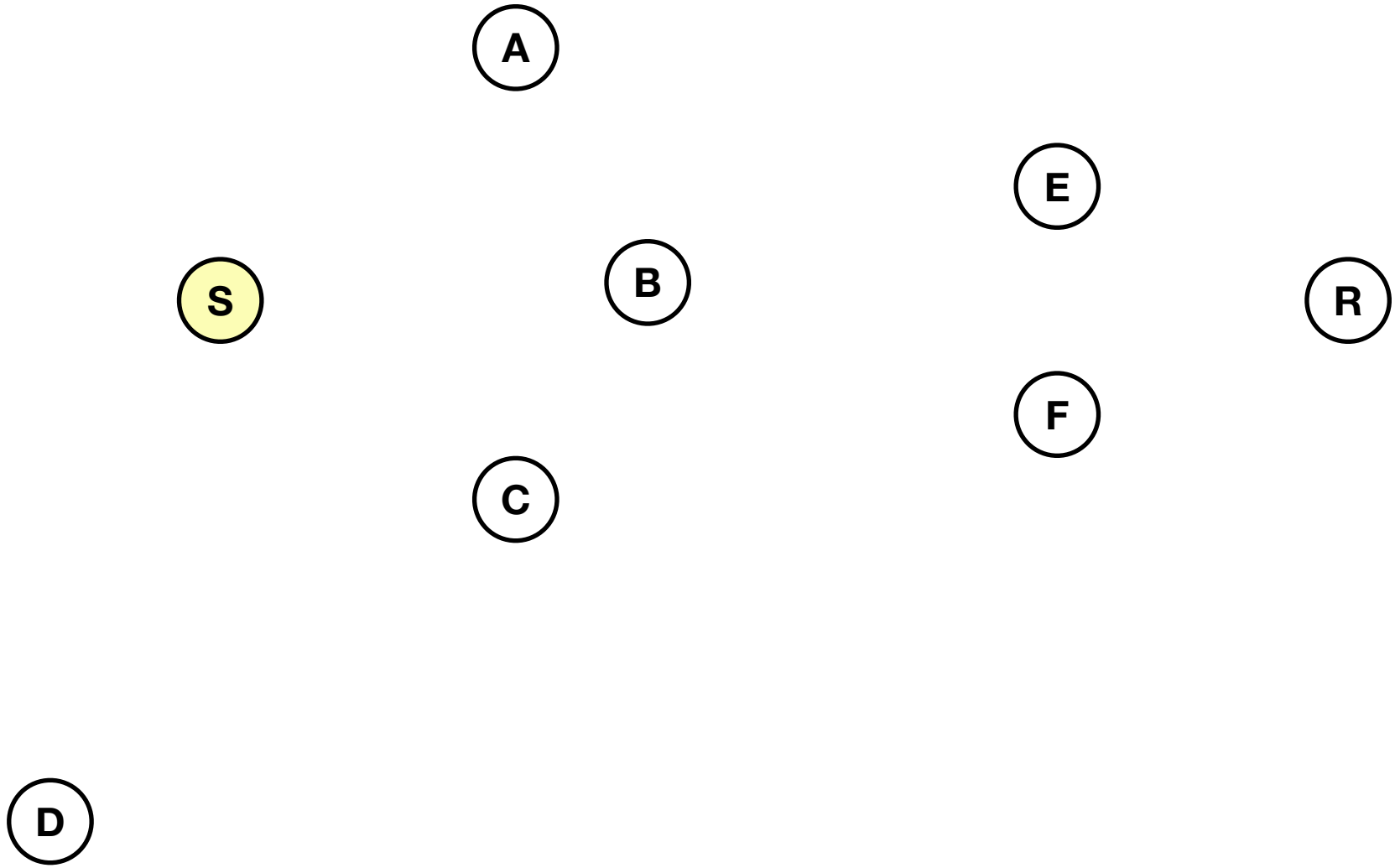
Distances

- The network shares a single **distance metric**
  - (Could be: hop count, receive power, geo distance...)
- In every packet, senders broadcast their distance to the source and destination names
- Nodes remember their distance to active names for some time
- **Only nodes with a smaller distance to the destination name are *eligible forwarders***

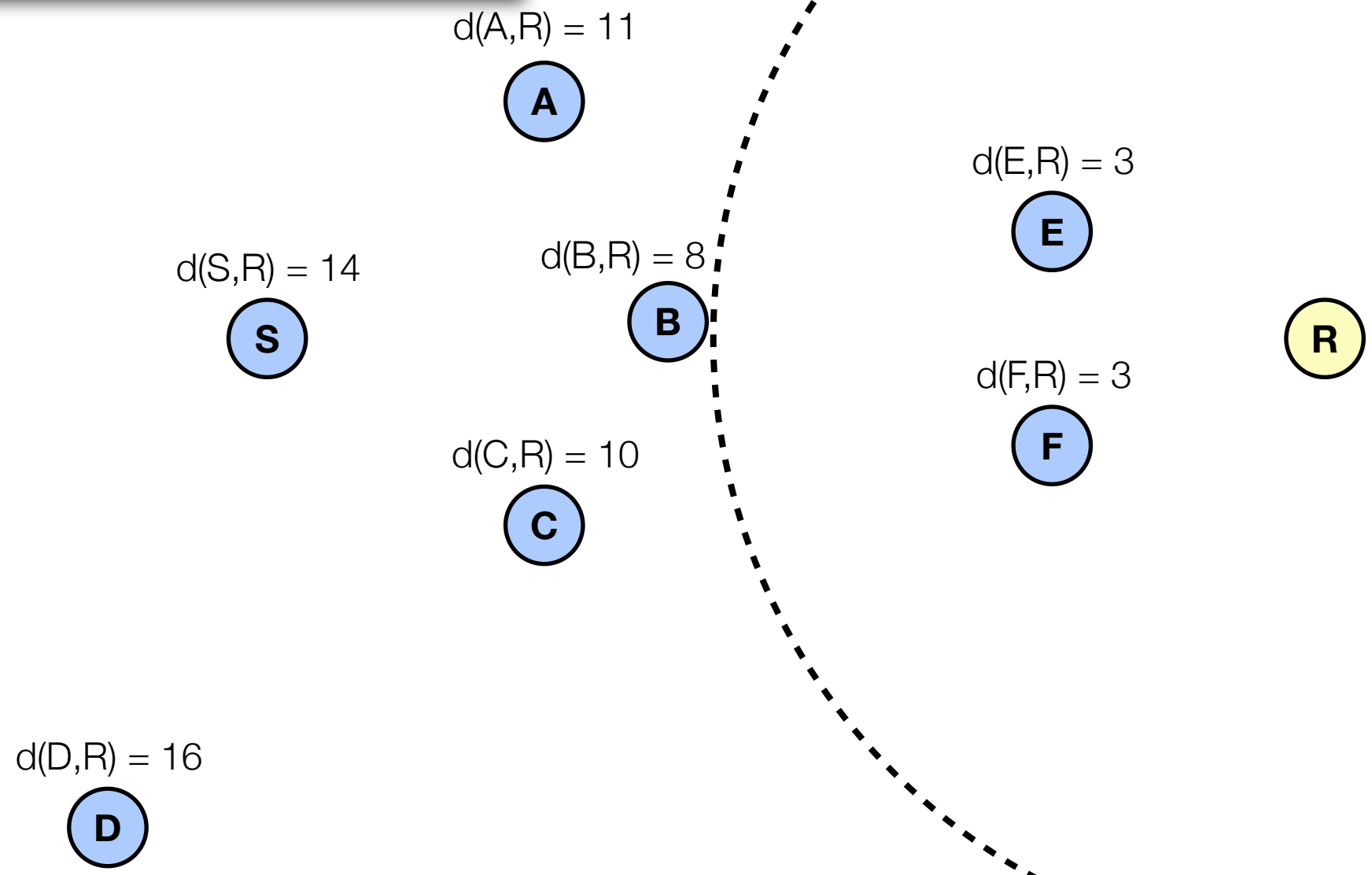
Listening Periods

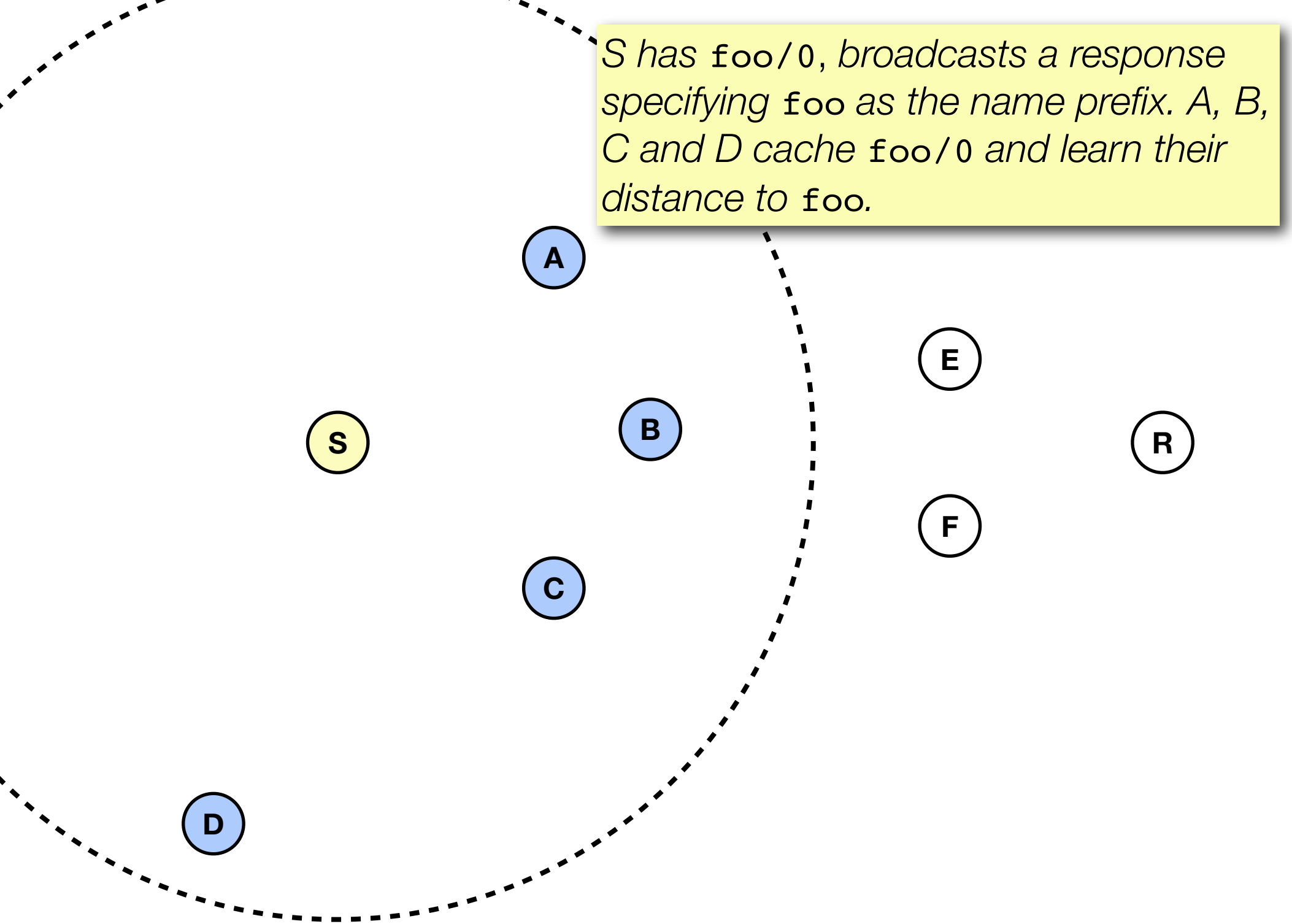
- Eligible forwarders choose their listening period based on the network's **delay metric**
  - Tells the node how long to wait before forwarding
- Only forward if another node does not forward before the listening period is over

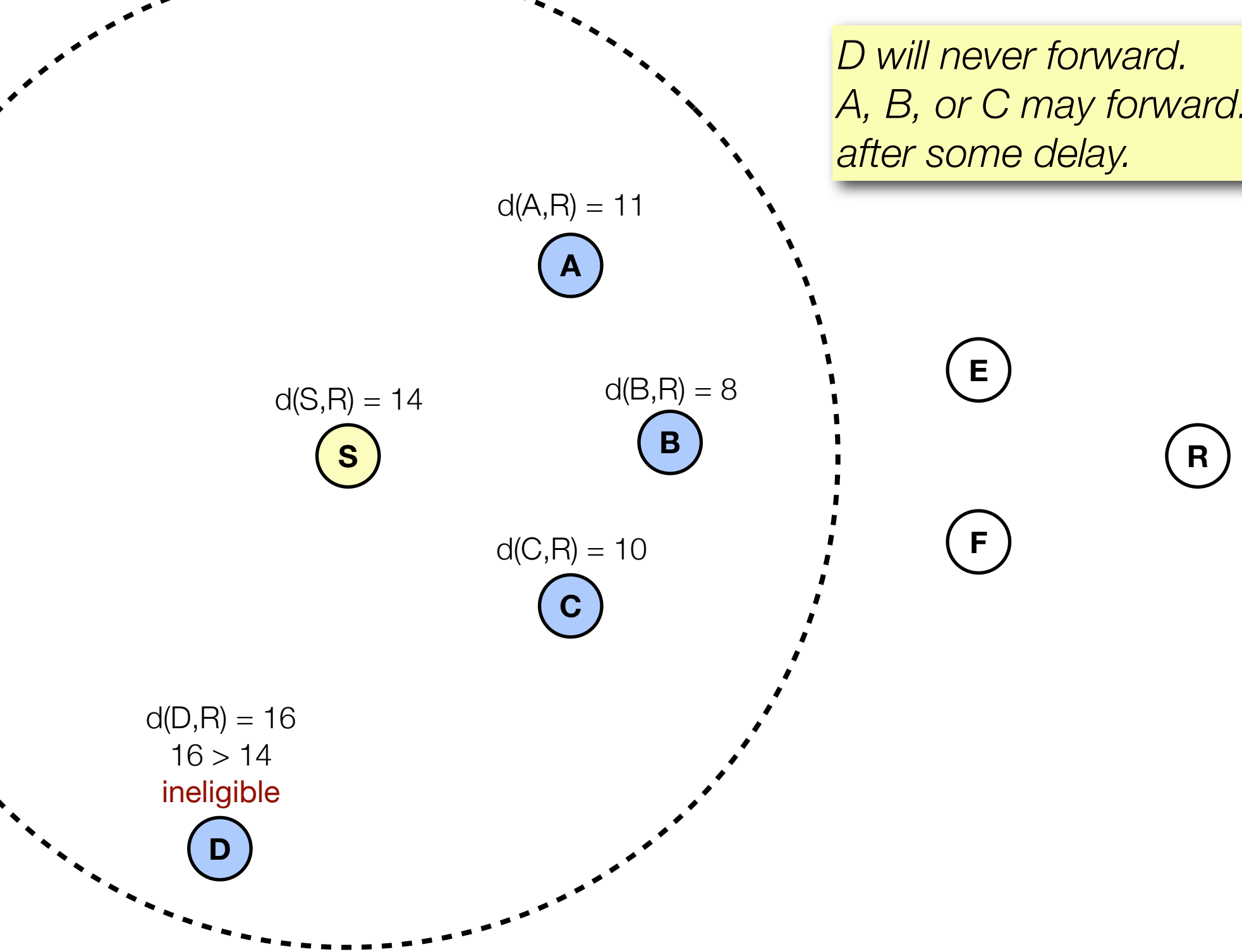
Suppose S has all the data in the prefix `foo`. No other node has this data.



R broadcasts a request for `foo/0`*, all nodes forward, record their distance from R*

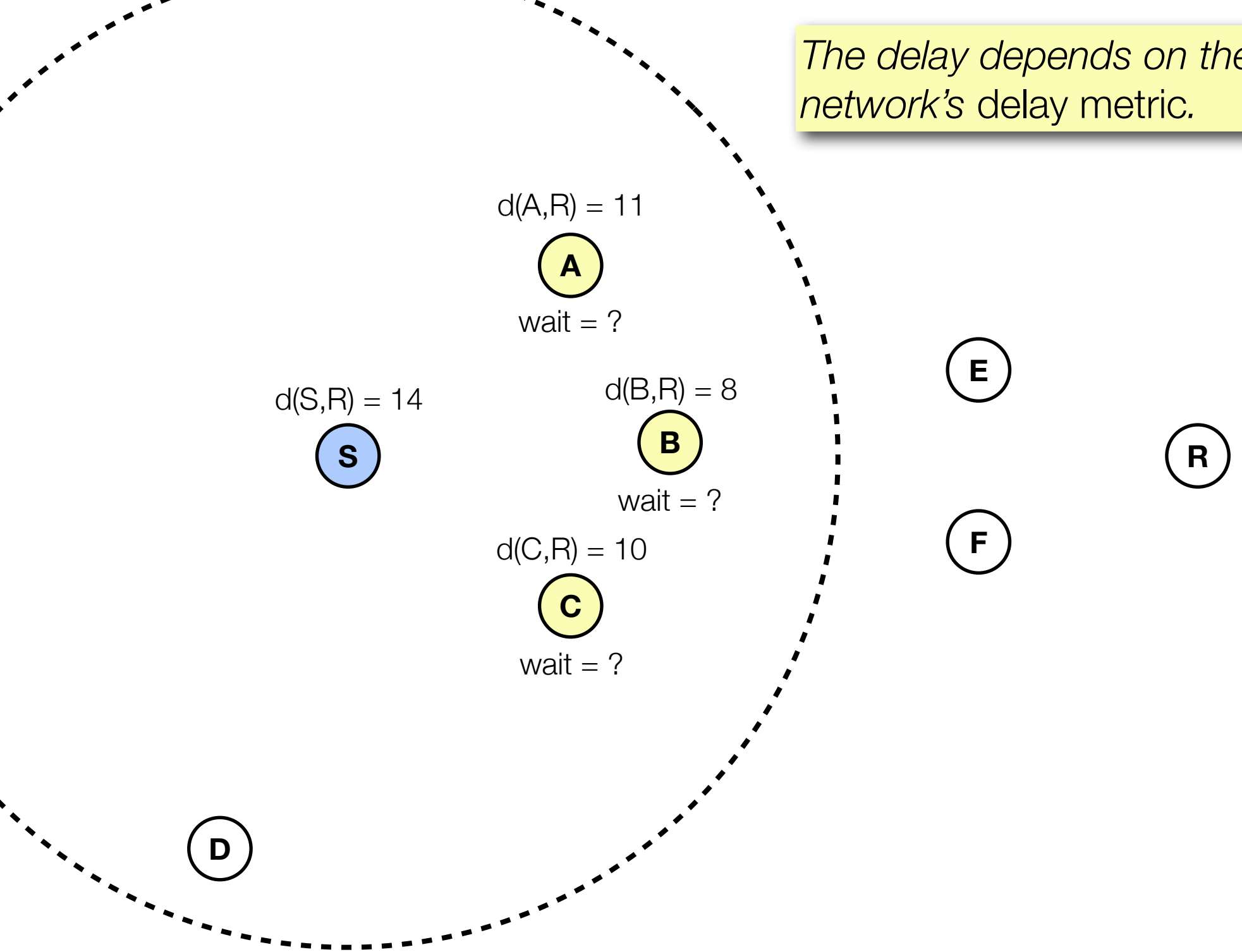



S has foo/0, broadcasts a response specifying foo as the name prefix. A, B, C and D cache foo/0 and learn their distance to foo.
A
B
C
D
E
F
R
S

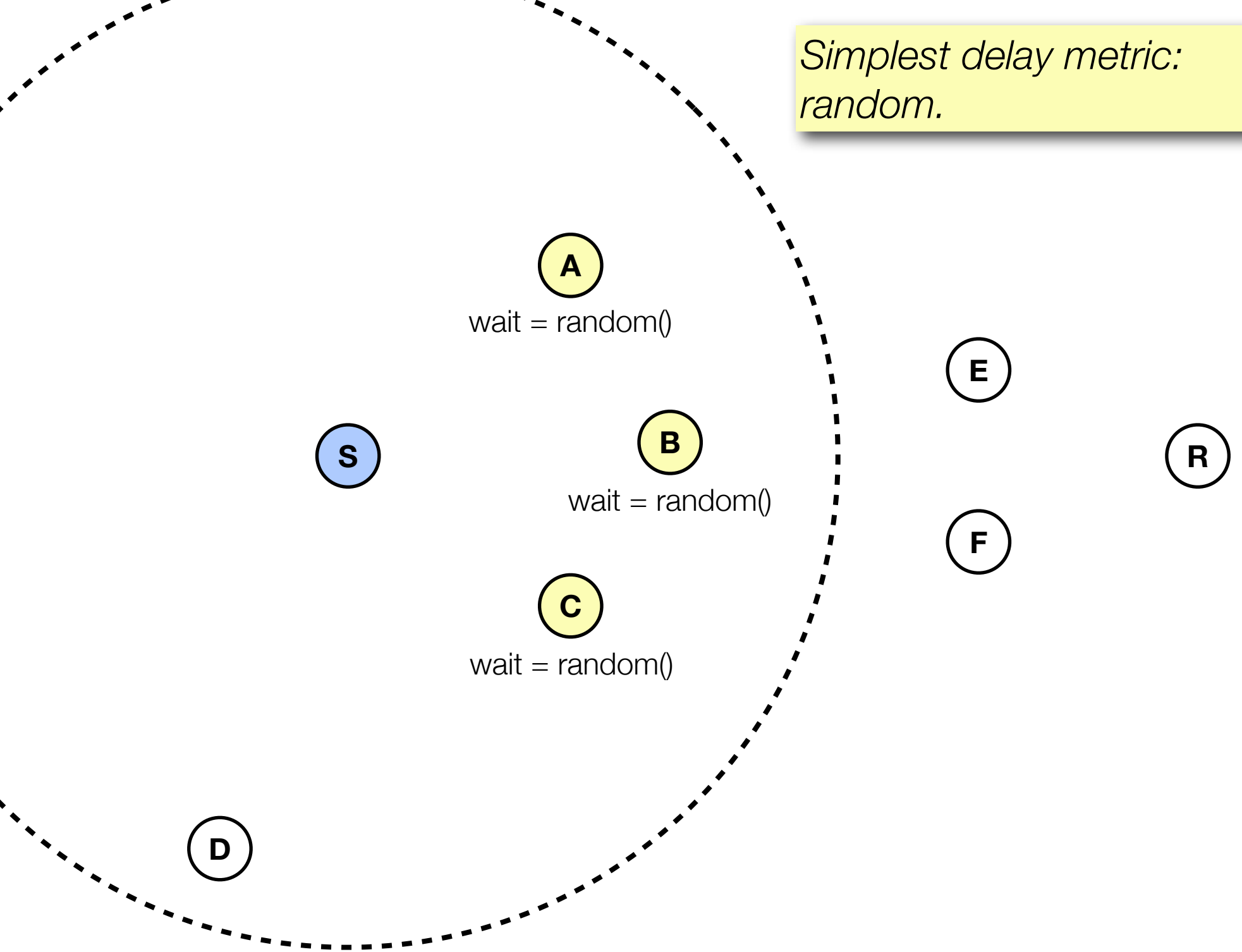


D will never forward.
A, B, or C may forward...
after some delay.

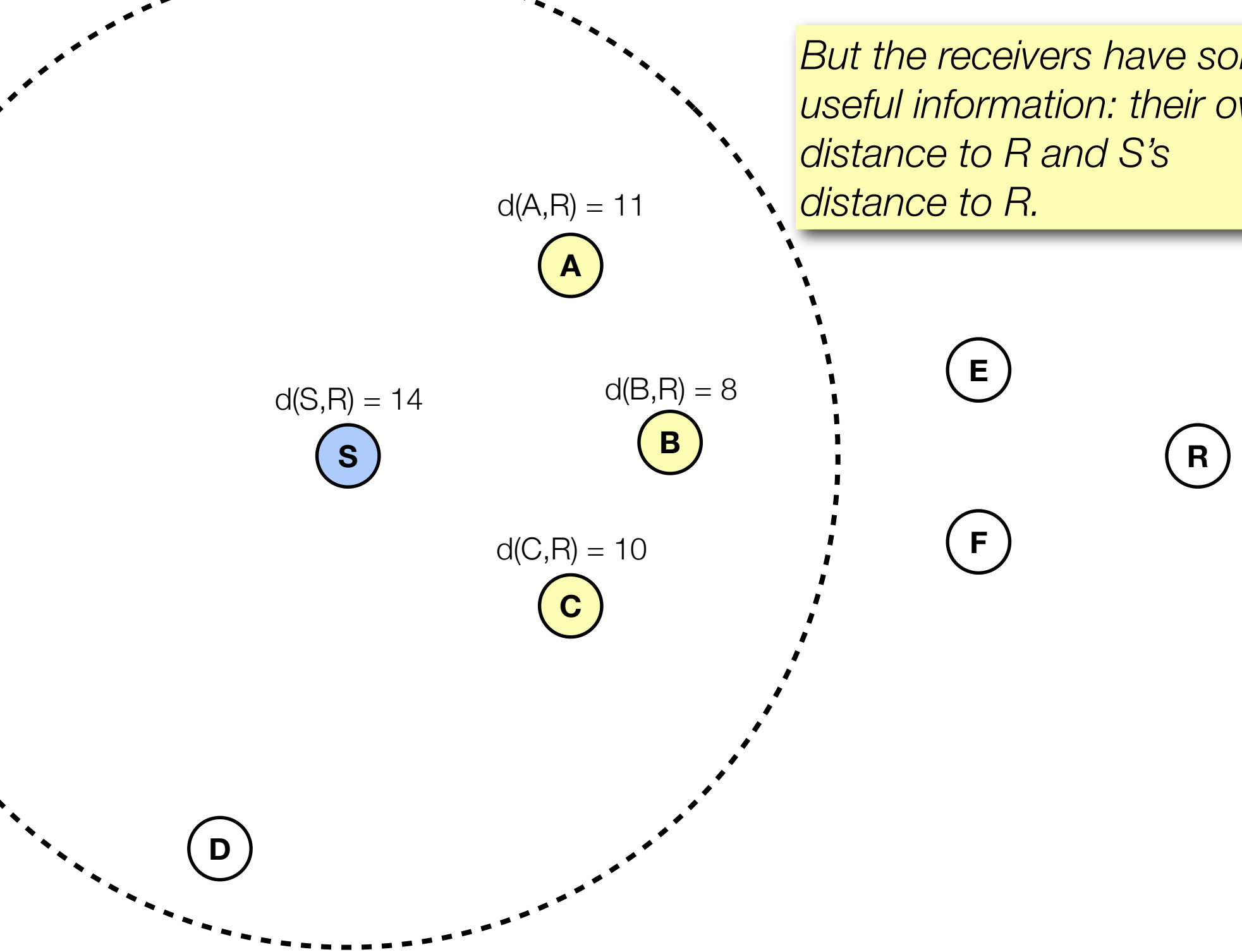
The delay depends on the network's delay metric.

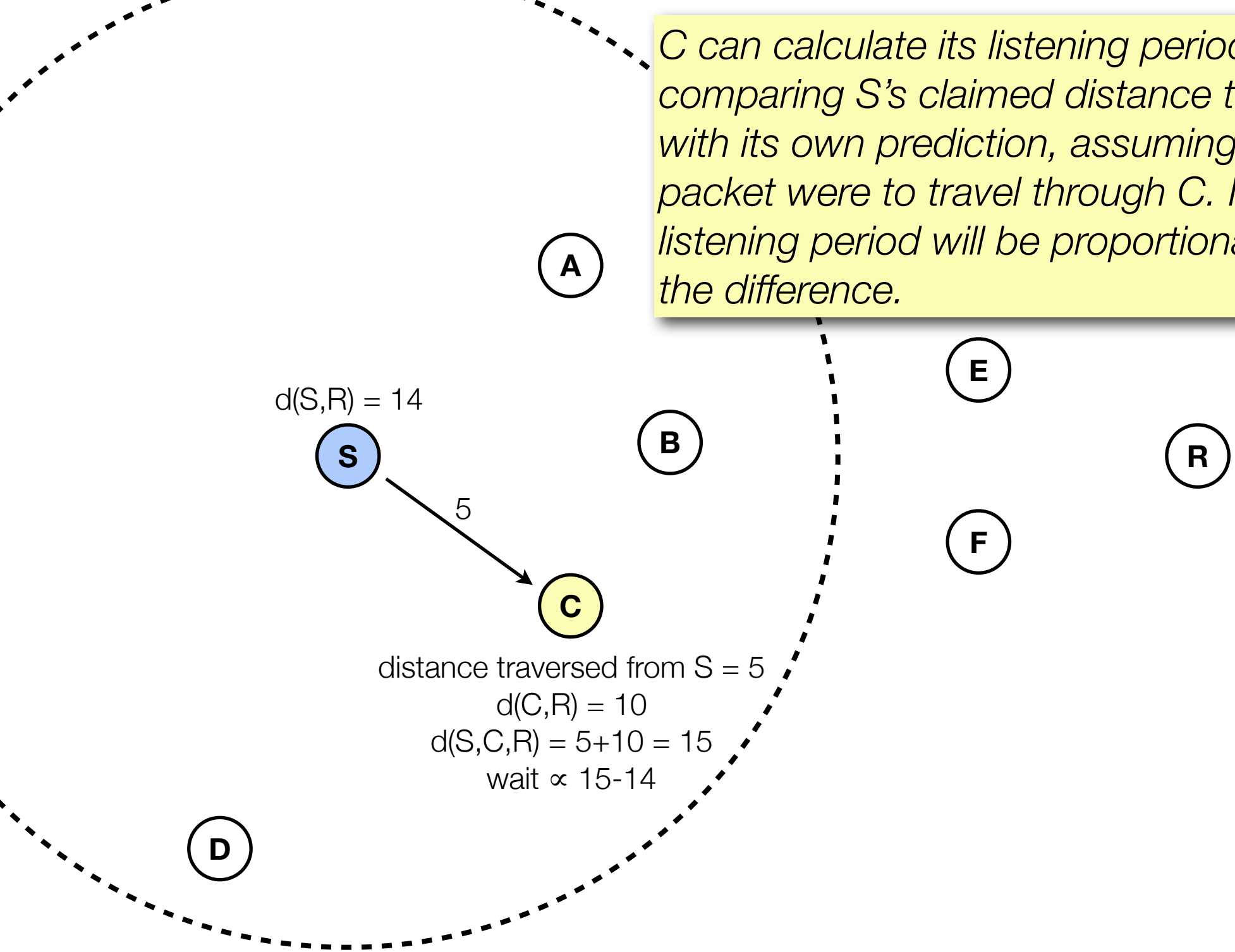


Simplest delay metric: random.

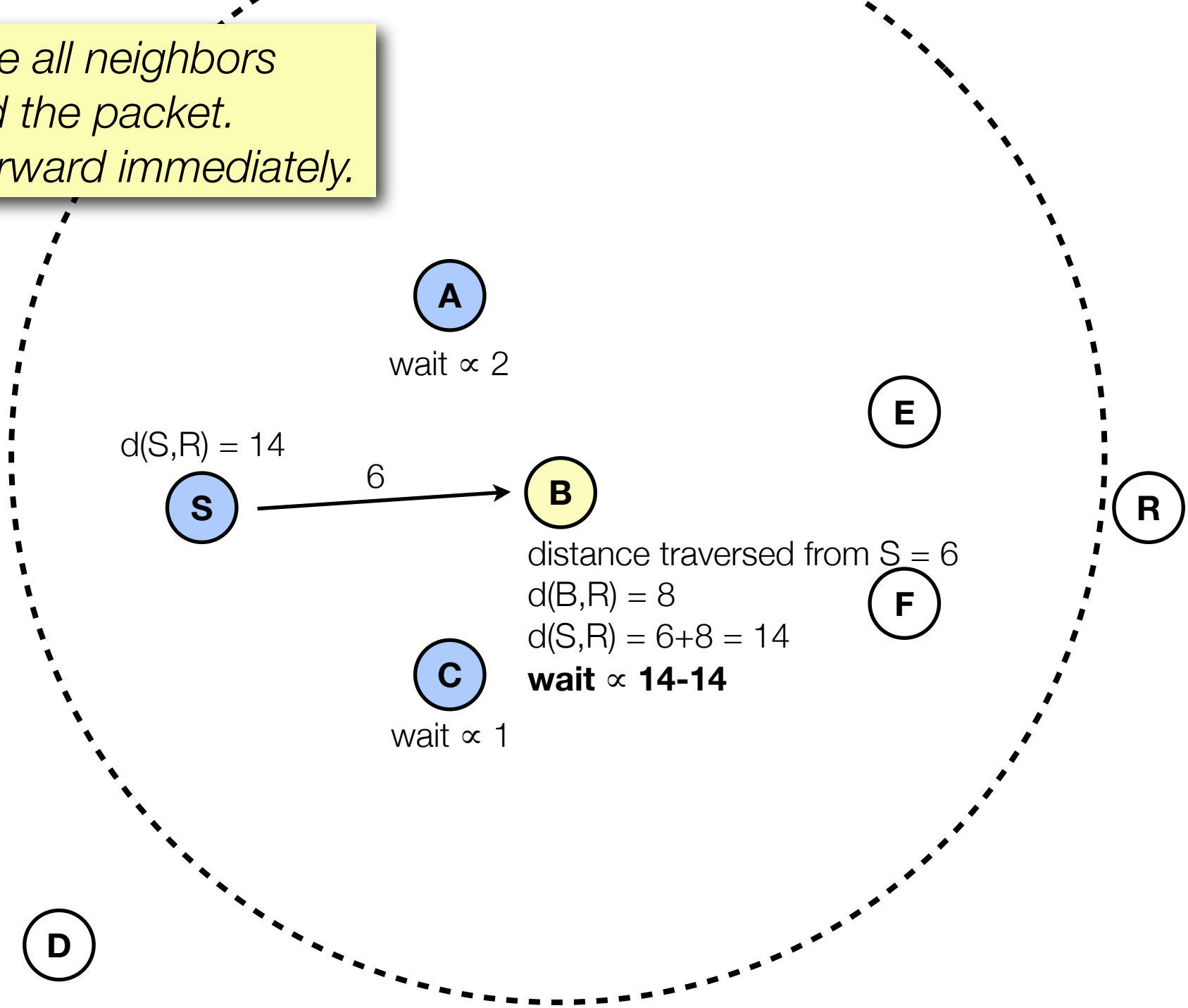


But the receivers have some useful information: their own distance to R and S's distance to R.

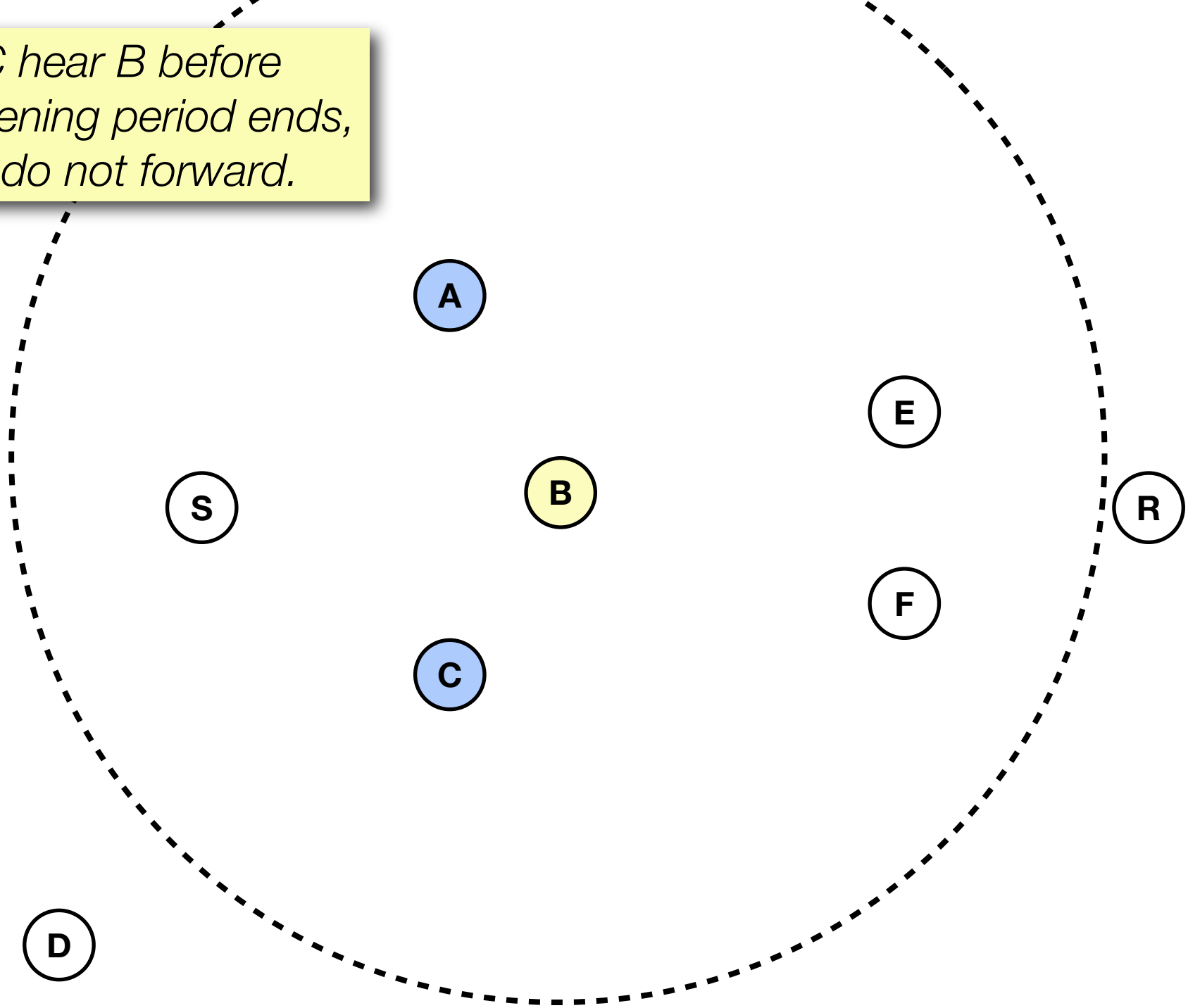



C can calculate its listening period by comparing S's claimed distance to R with its own prediction, assuming the packet were to travel through C. Its listening period will be proportional to the difference.
A
E
d(S,R) = 14
B
S
R
5
F
C
distance traversed from S = 5
d(C,R) = 10
d(S,C,R) = 5+10 = 15
wait ∝ 15-14
D

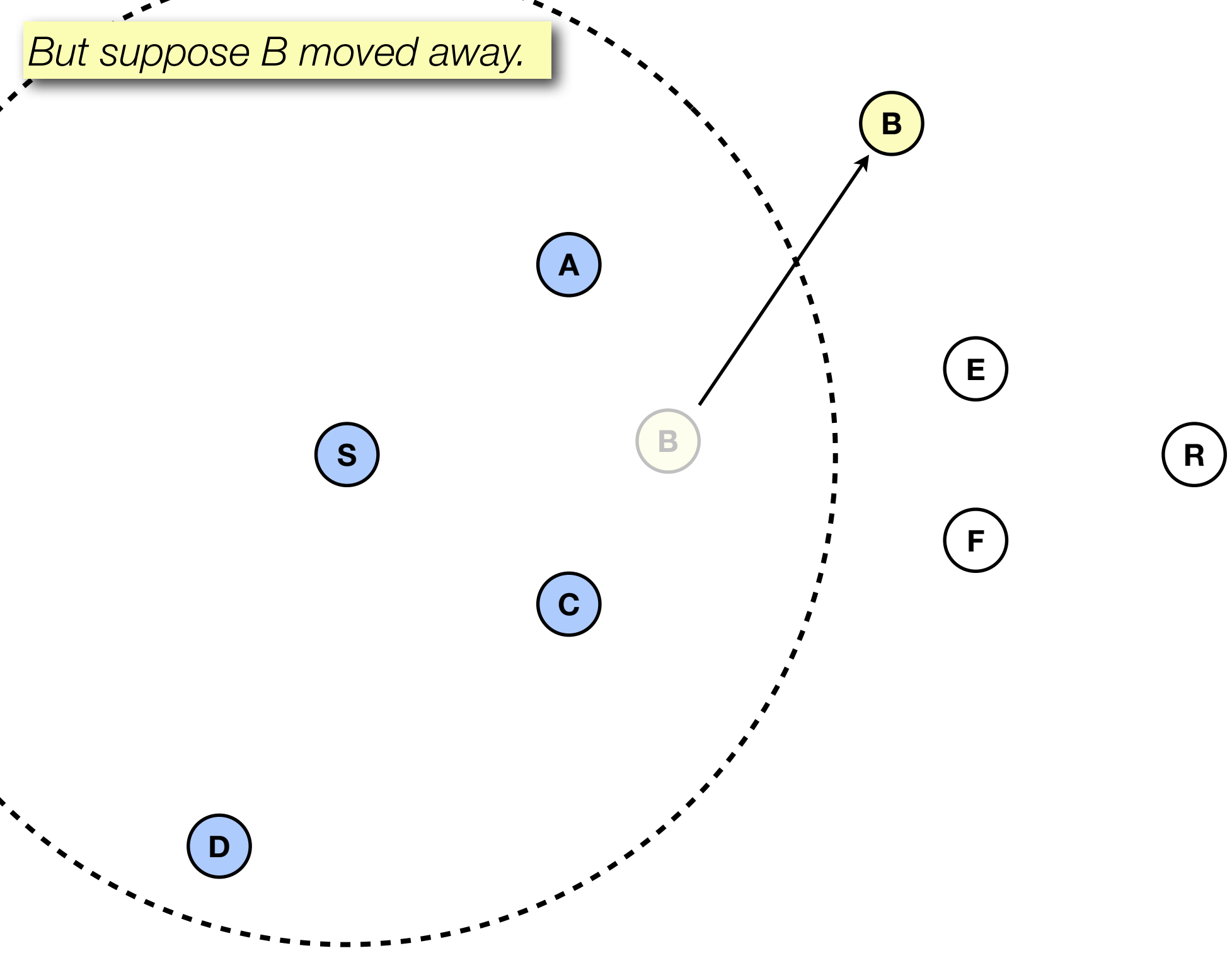
Suppose all neighbors received the packet.
B will forward immediately.



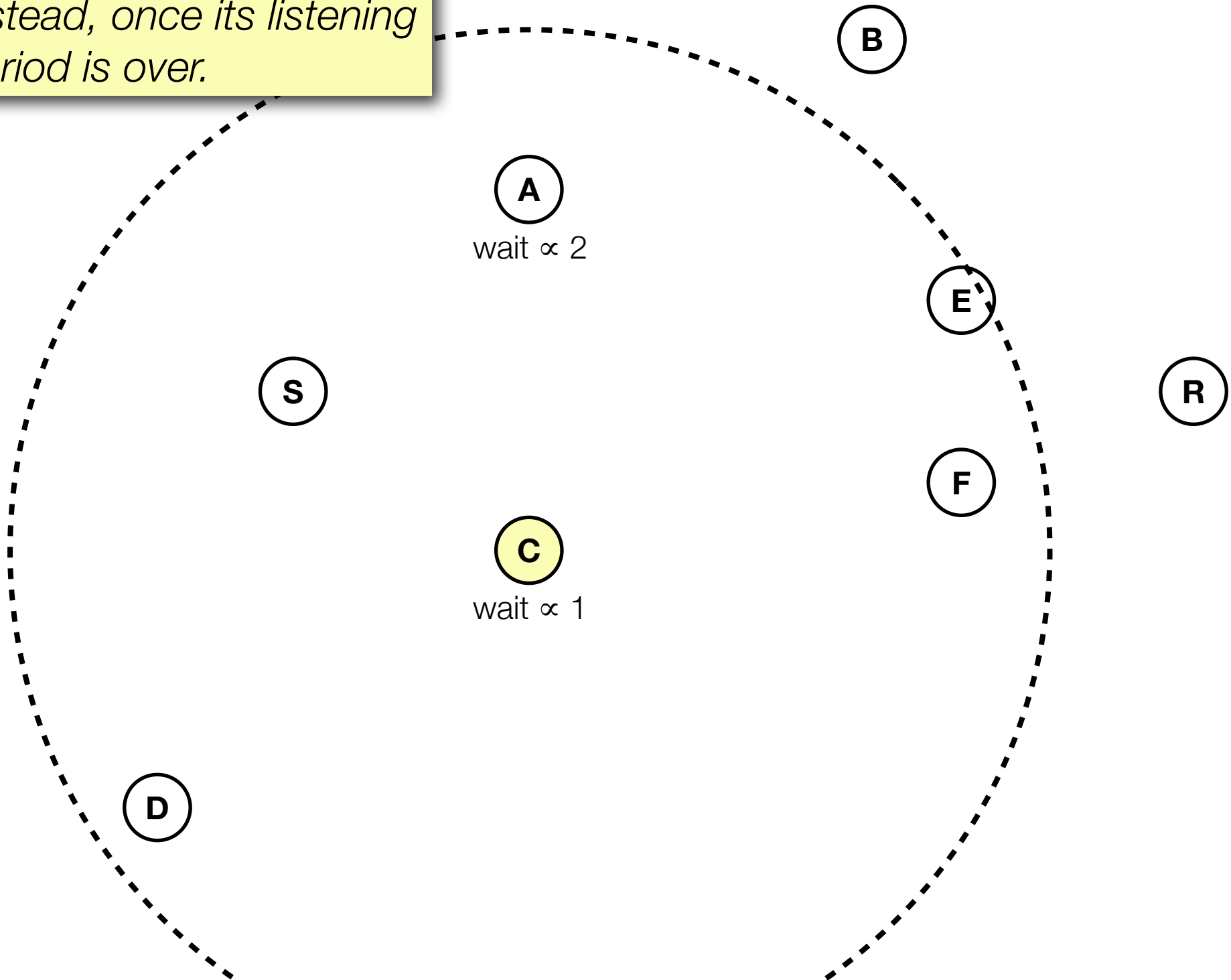
A and C hear B before their listening period ends, so they do not forward.



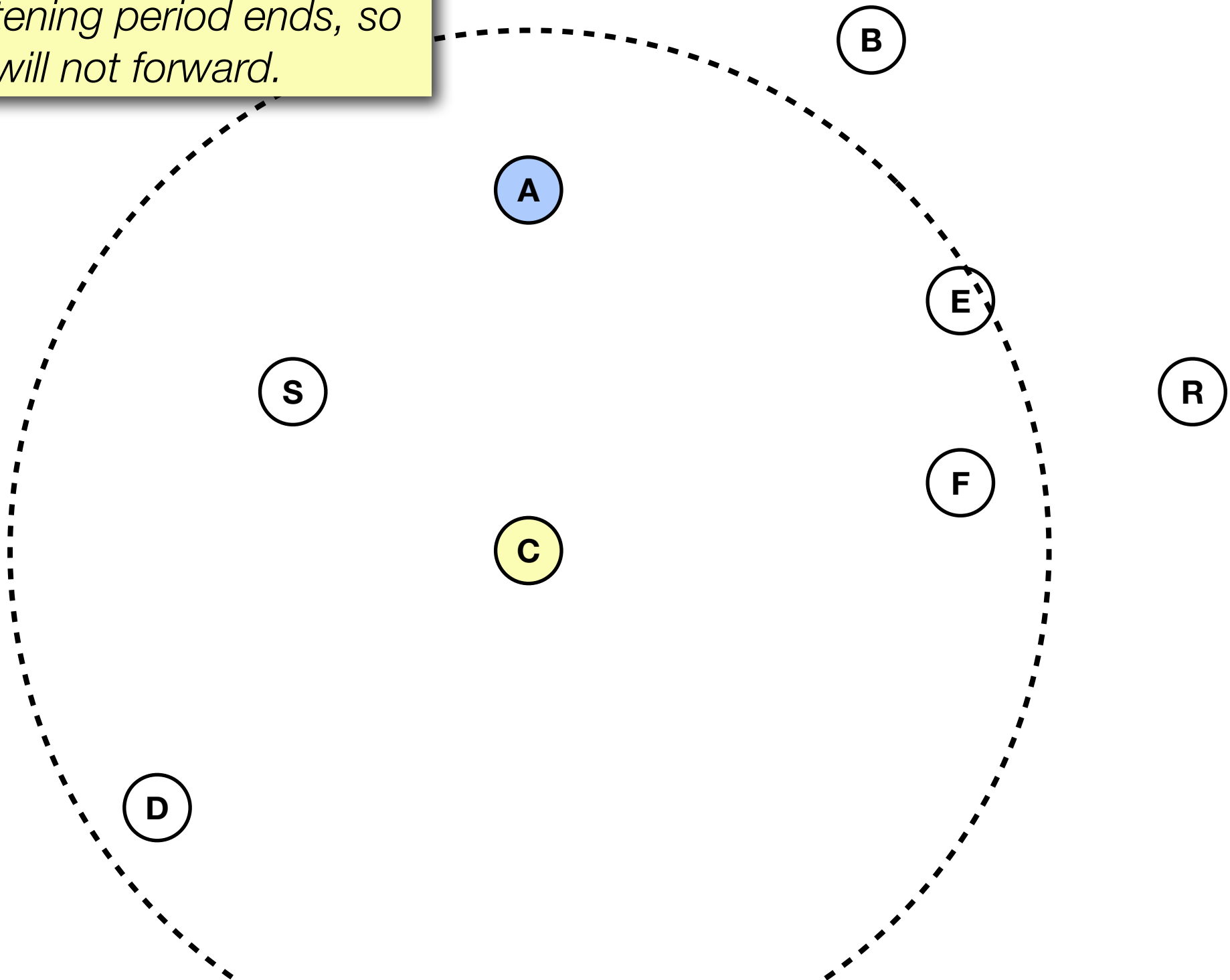
But suppose B moved away.



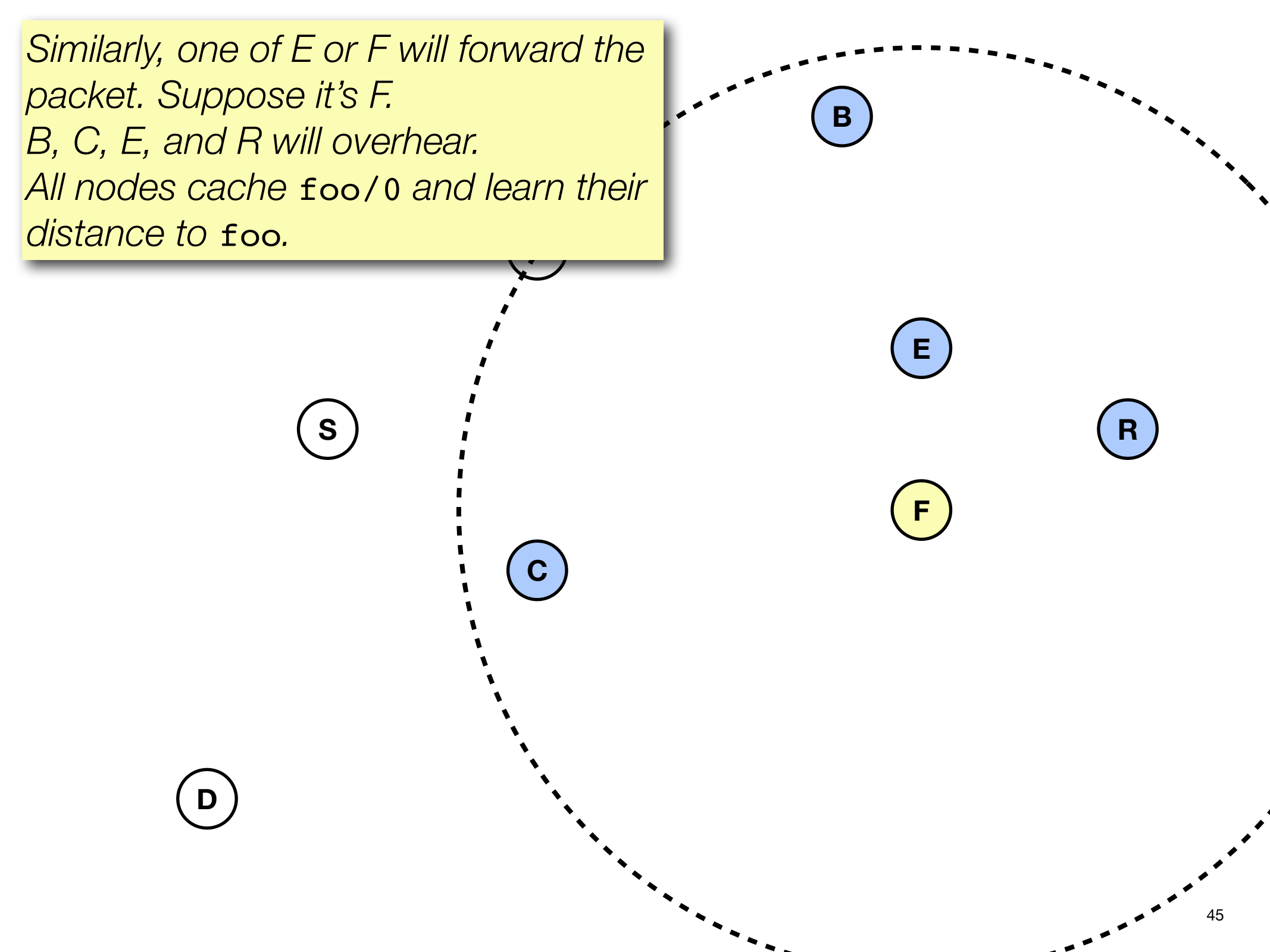
C will forward the packet instead, once its listening period is over.



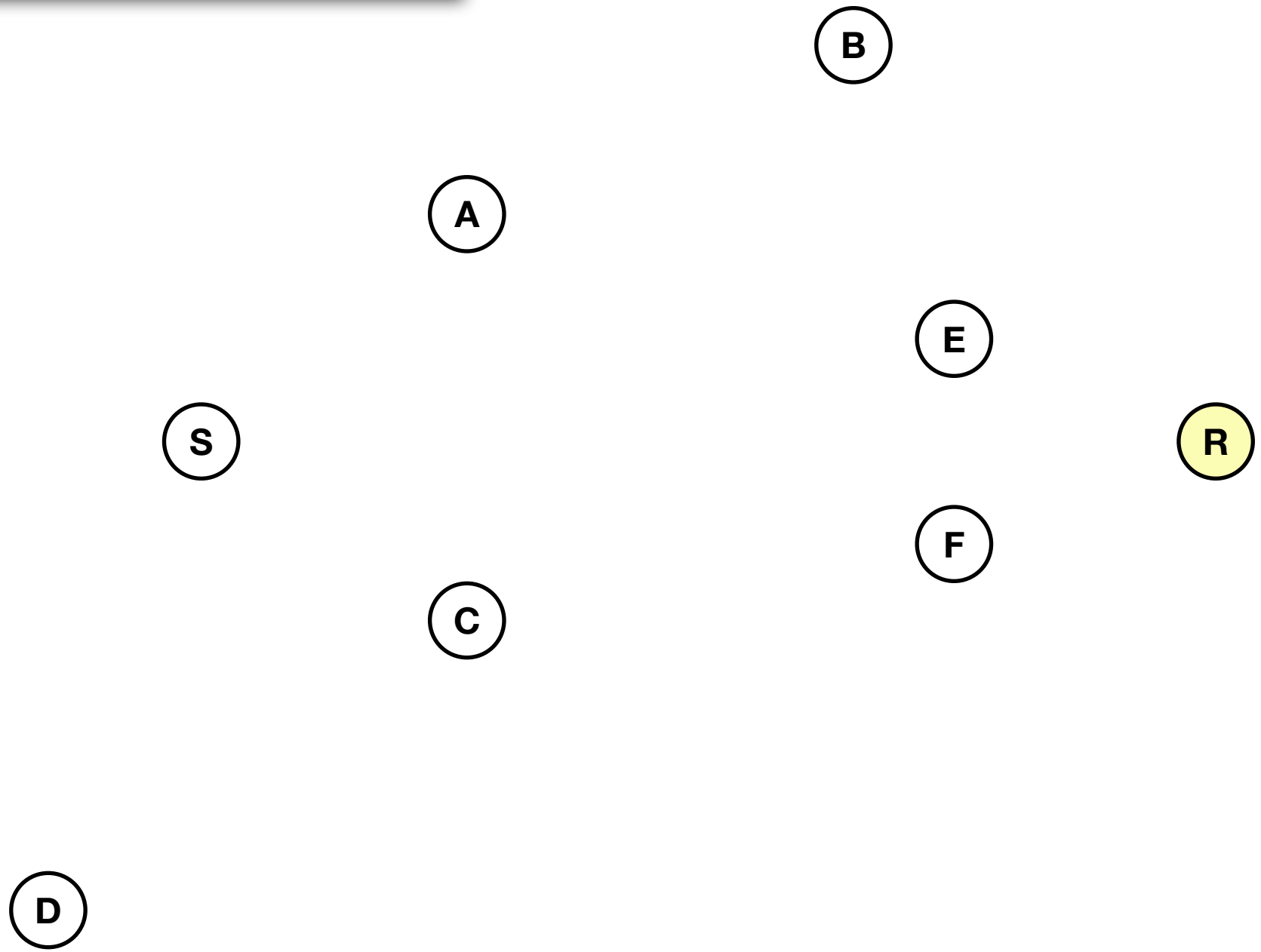
A will hear C before its listening period ends, so A will not forward.



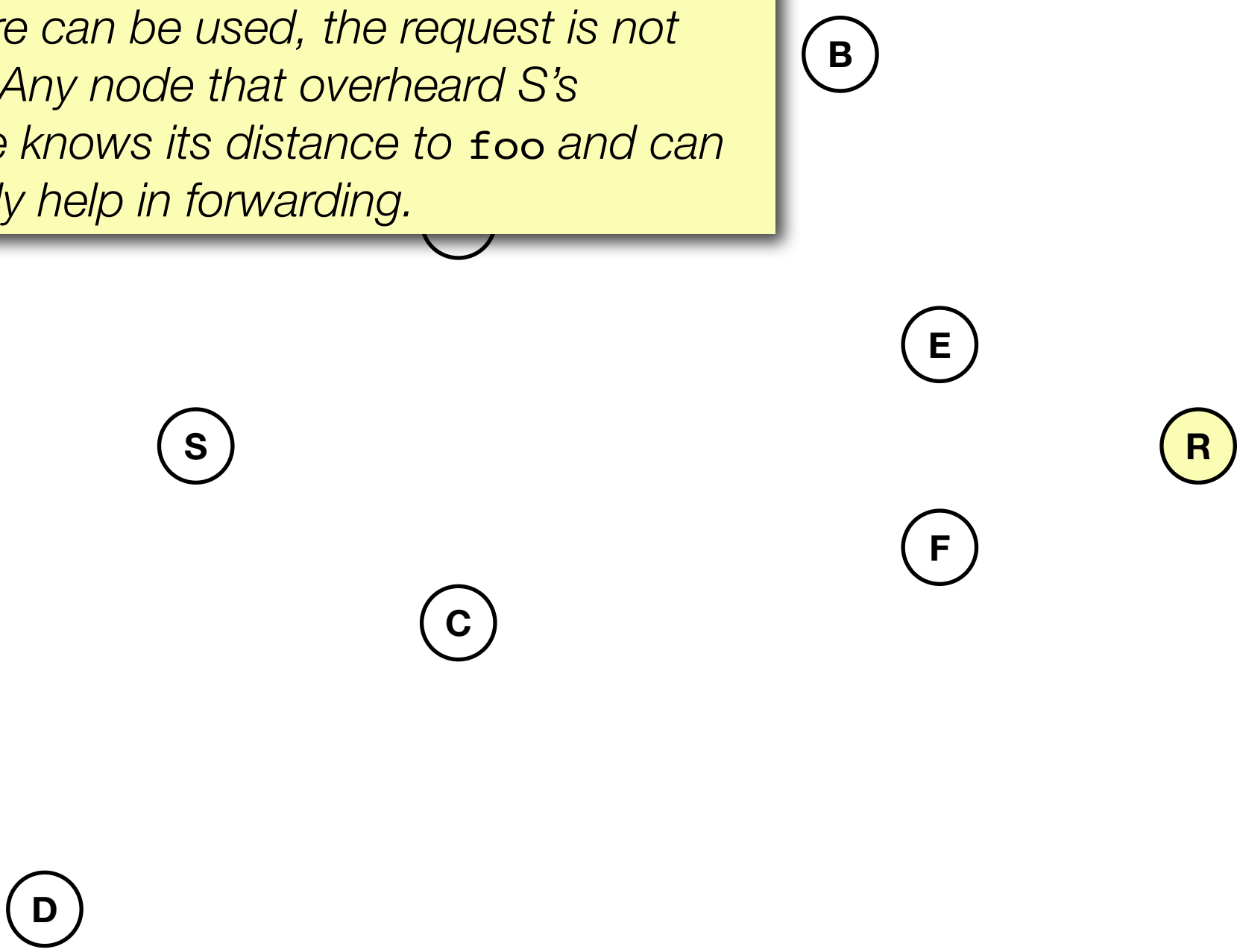
Similarly, one of E or F will forward the packet. Suppose it's F.
B, C, E, and R will overhear.
All nodes cache `foo/0` *and learn their distance to* `foo`.



The response has reached R.



When R requests `foo/1`*, the same forwarding procedure can be used, the request is not flooded. Any node that overheard S's response knows its distance to* `foo` *and can potentially help in forwarding.*



On Reliability

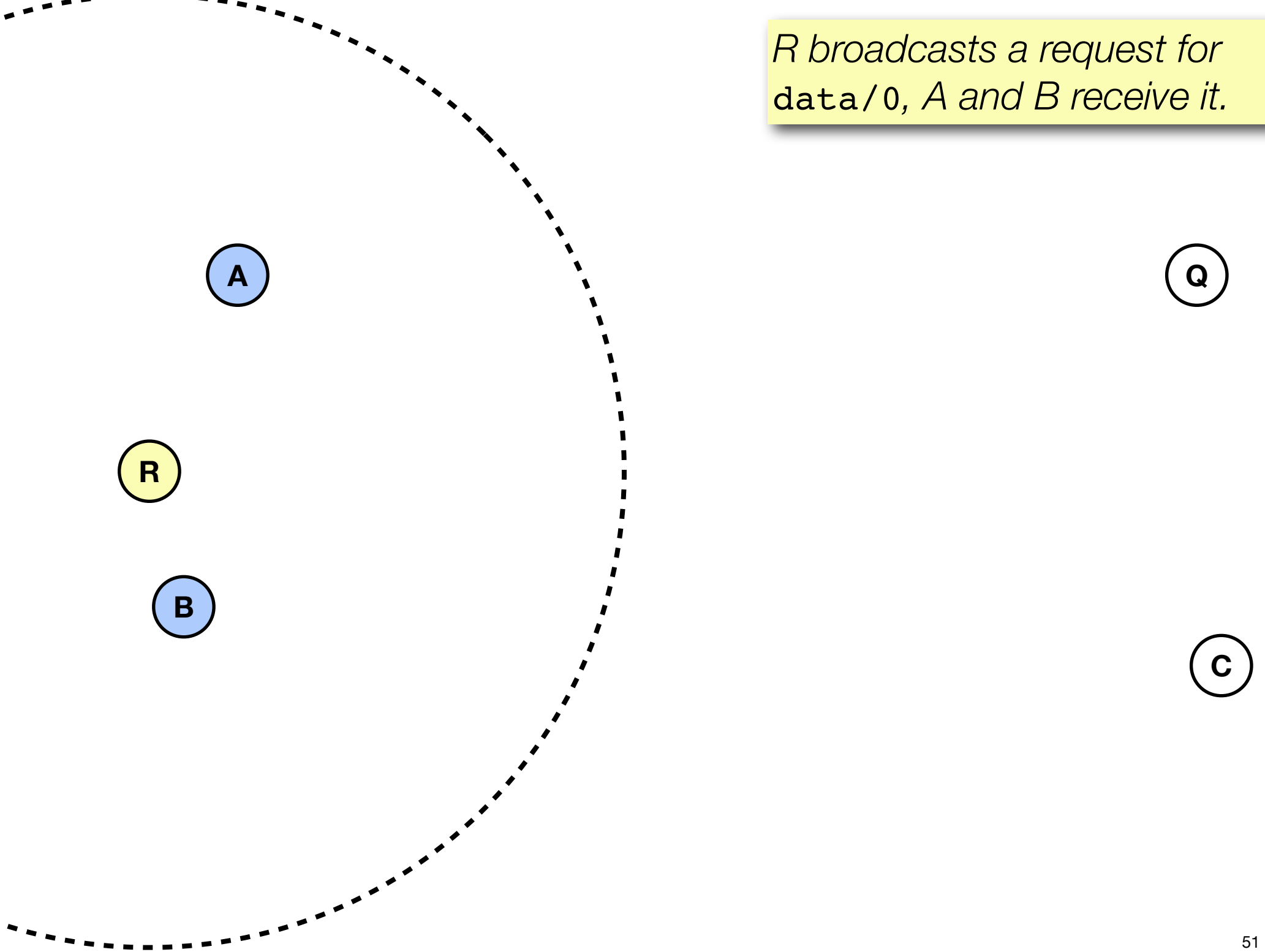
- BOND provides no guarantees
- The requester should re-request any missing data
- Re-requests may not need to travel very far, the data could be cached nearby as a result of the previous request

Disconnected Networks

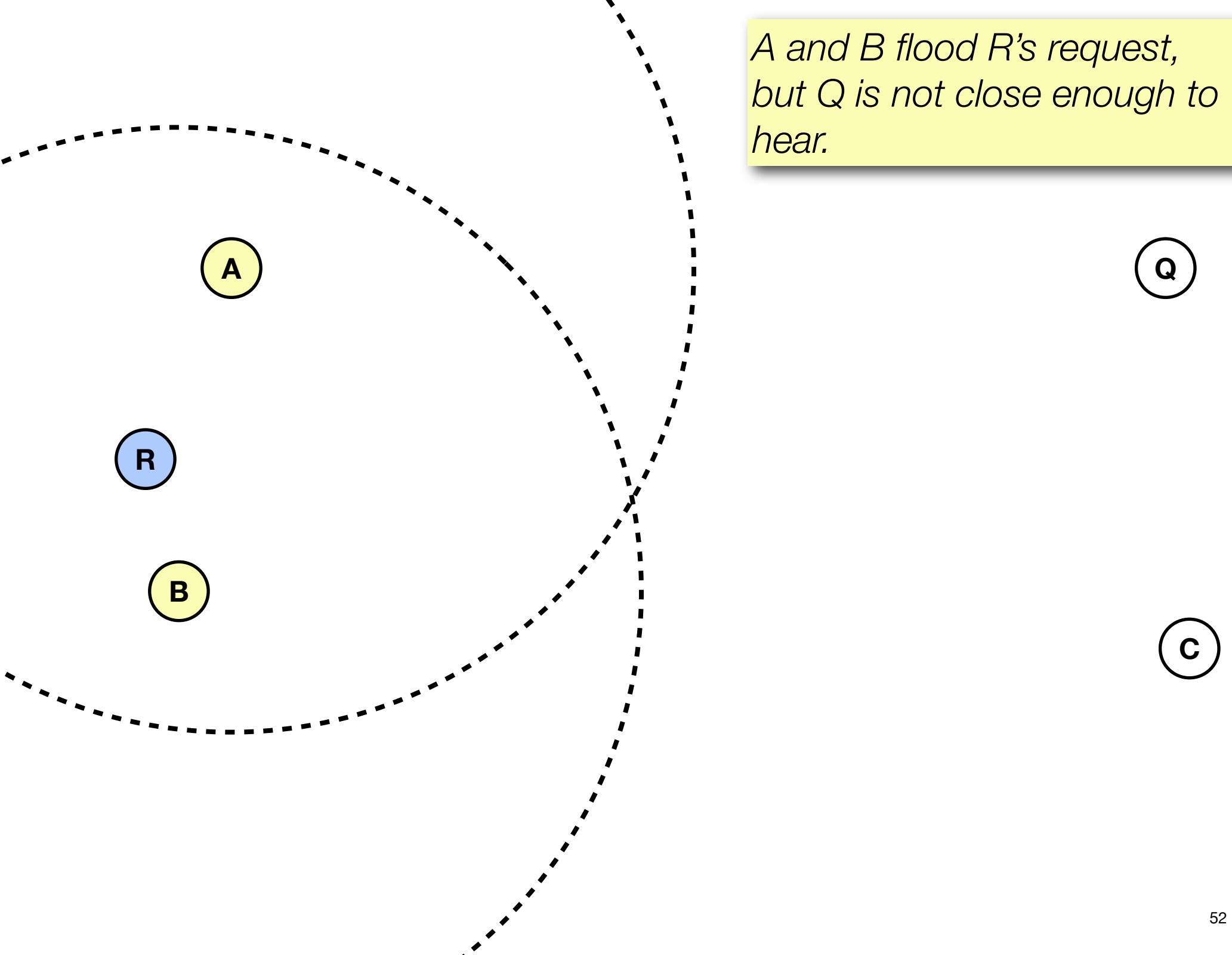
- Caching ferries data between disconnected portions of the network
- But requests need to be ferried, too
- **Request replay:** intermediate nodes retransmit request at regular intervals until either:
  - They hear a response to the request, or
  - They reach a fixed maximum number of replays

Suppose Q is the only node that has `data/0`.

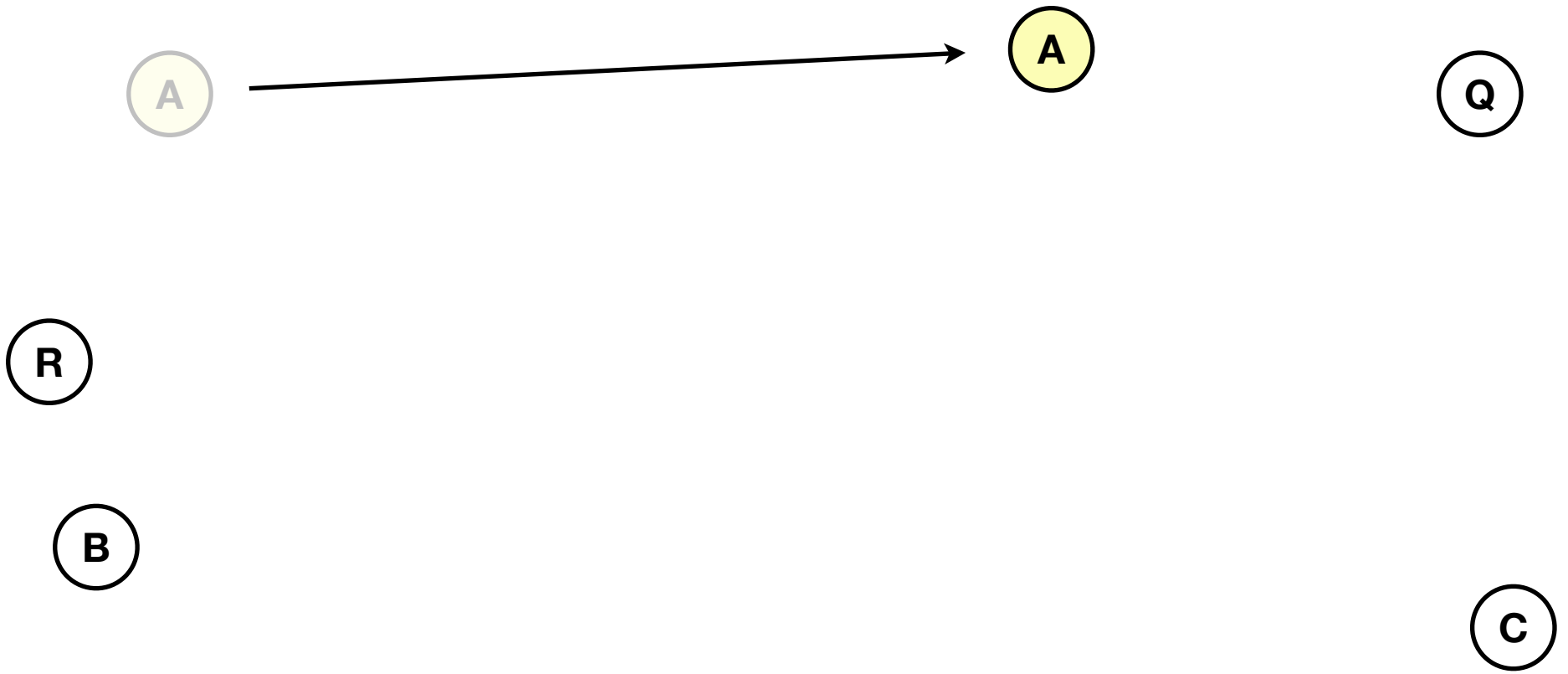
R broadcasts a request for `data/0`*, A and B receive it.*



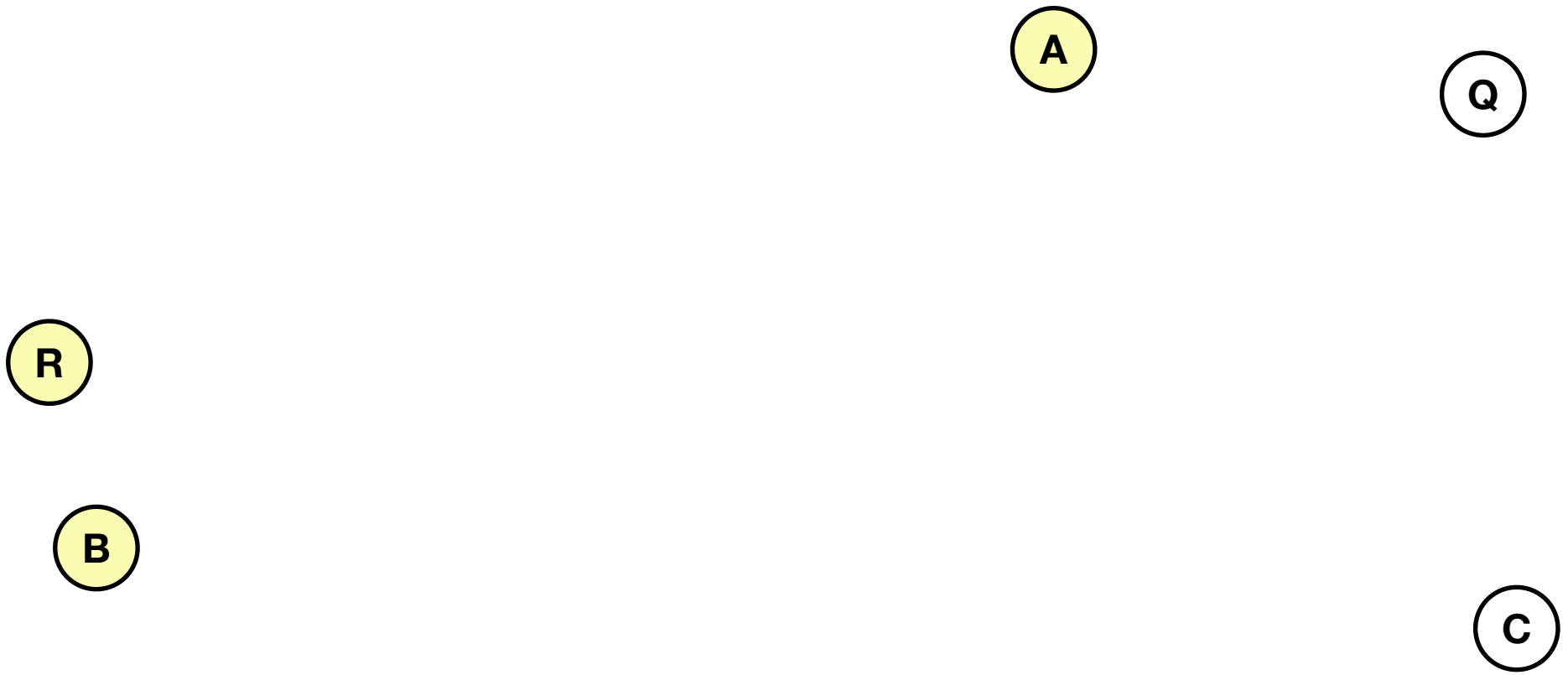
A and B flood R's request, but Q is not close enough to hear.



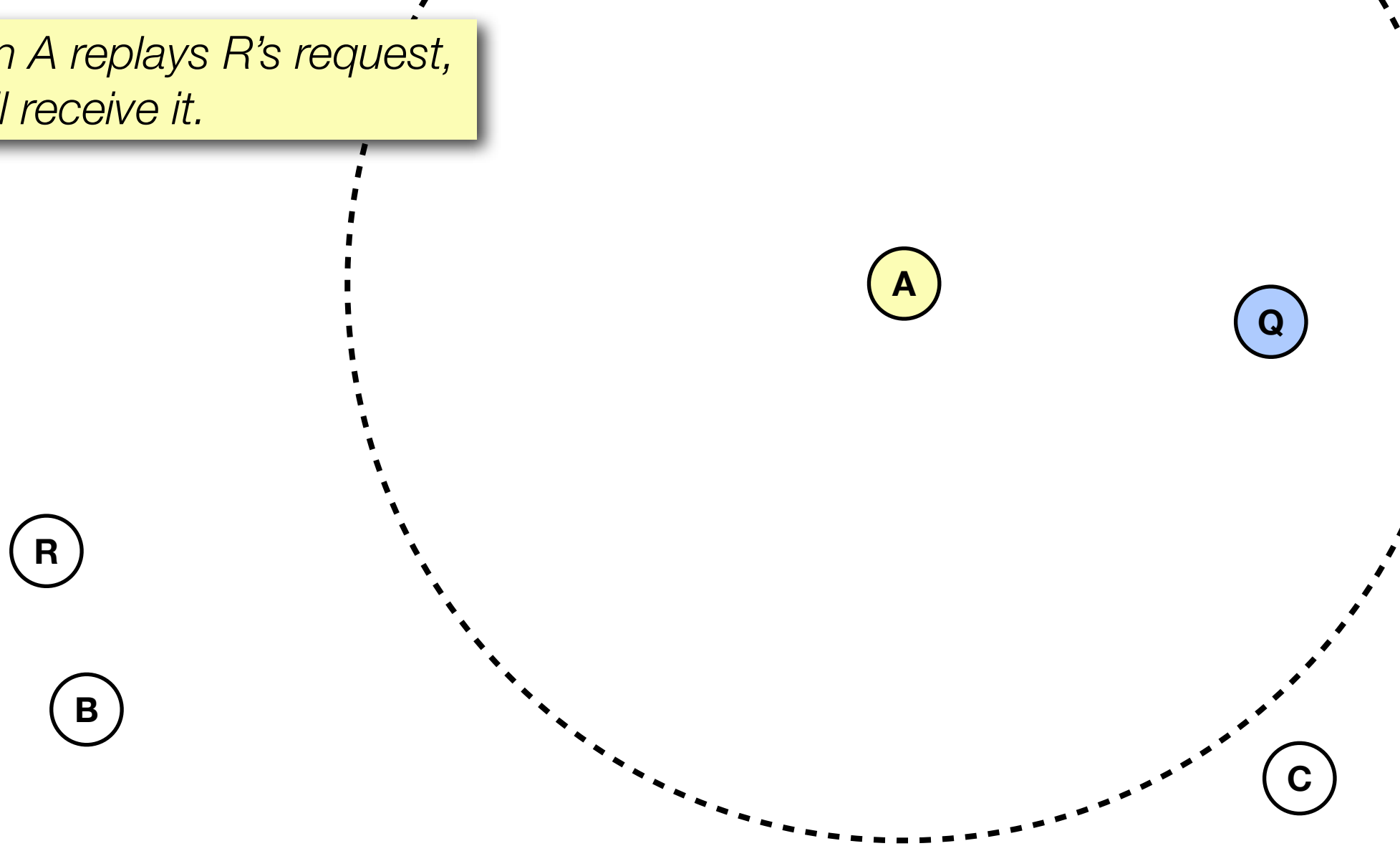
Suppose A moves towards Q.



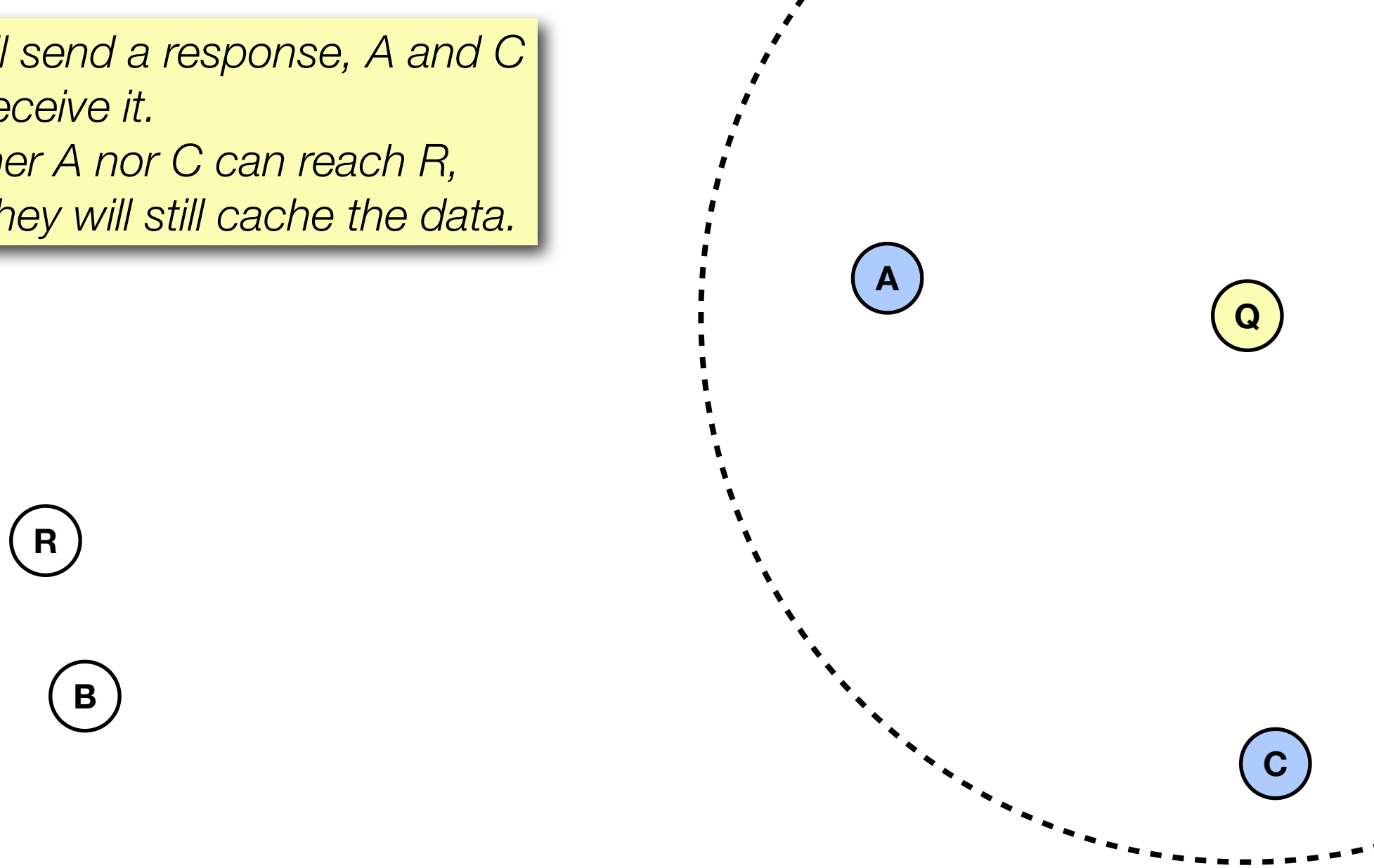
With request replay,
A, B, and R will all retransmit R's request at regular intervals.



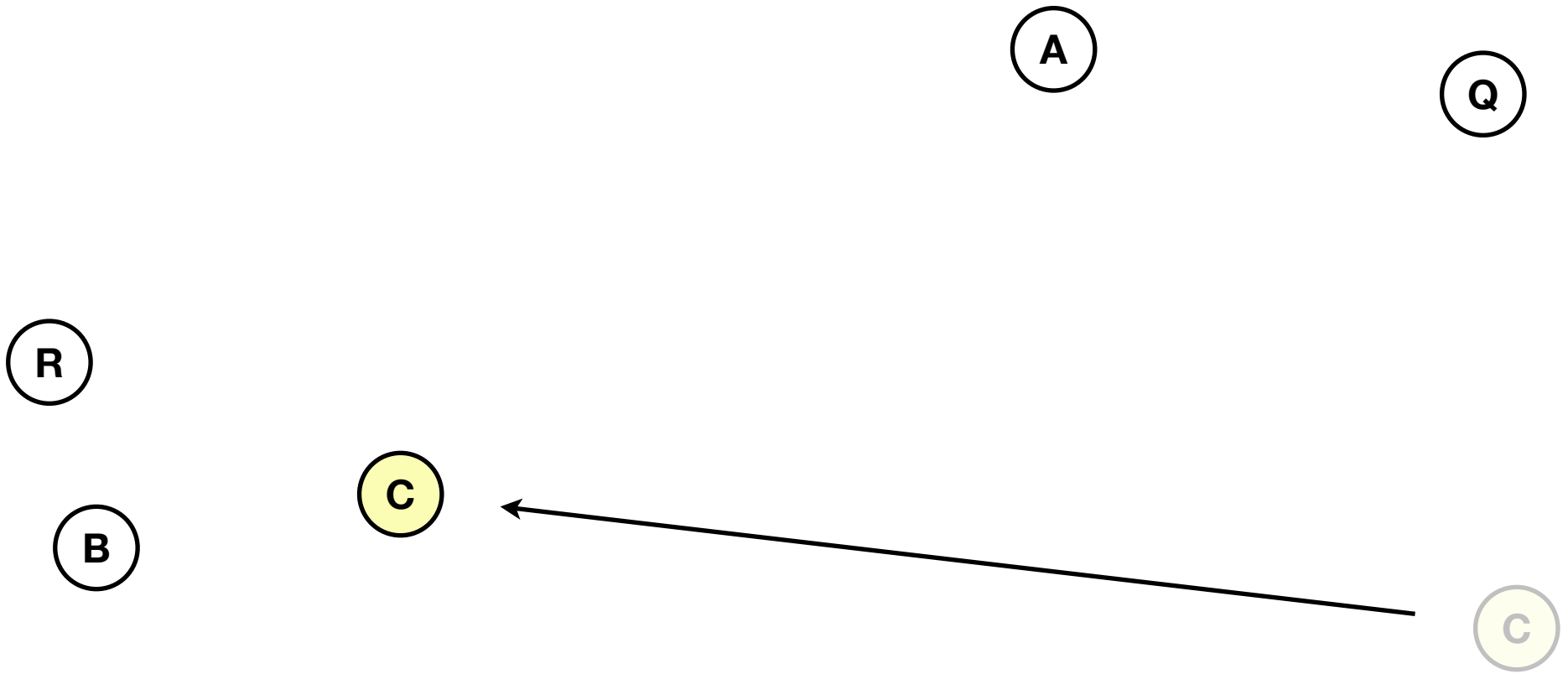
When A replays R's request, Q will receive it.



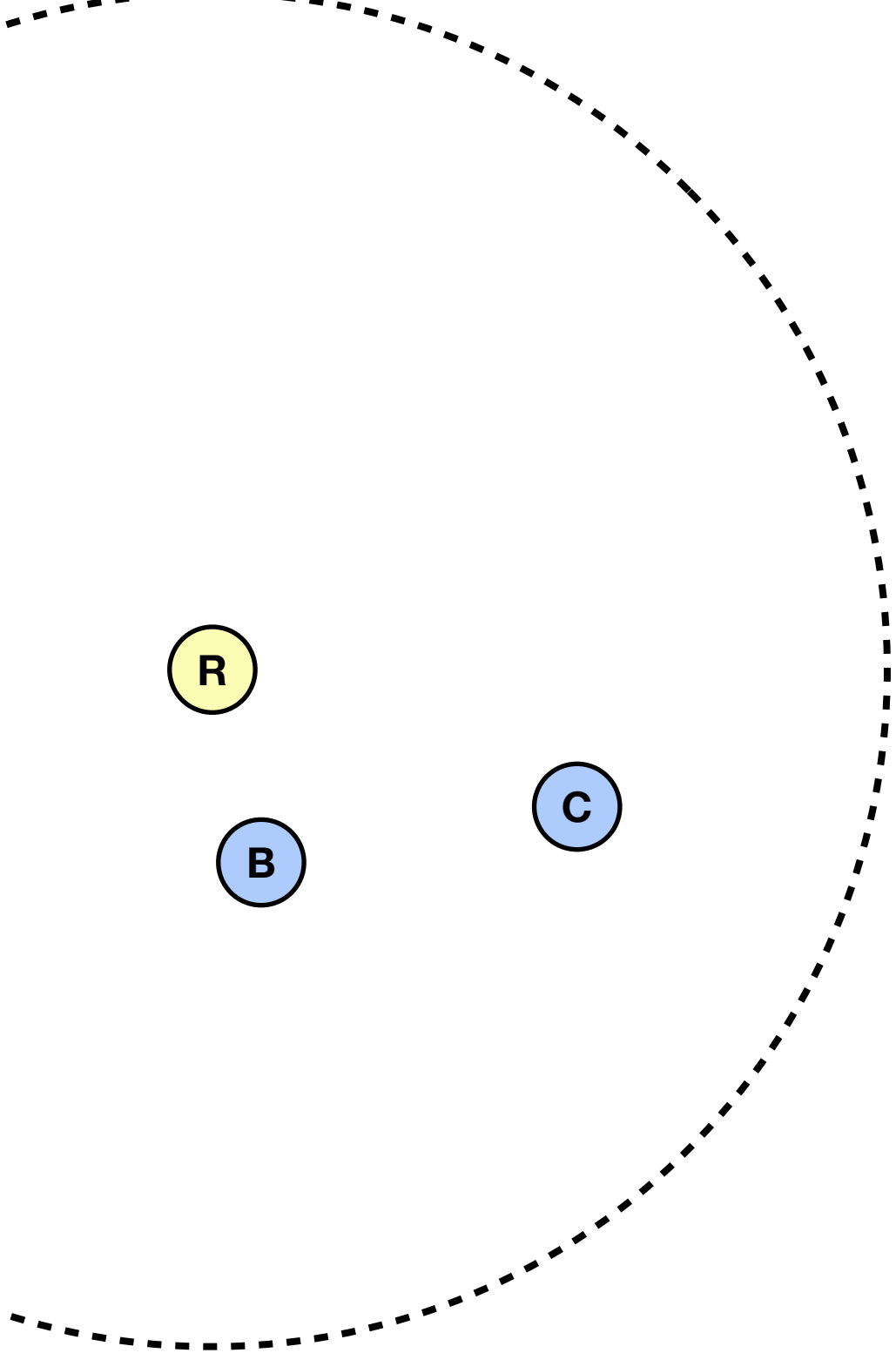
Q will send a response, A and C will receive it.
Neither A nor C can reach R, but they will still cache the data.



Suppose C moves near R.



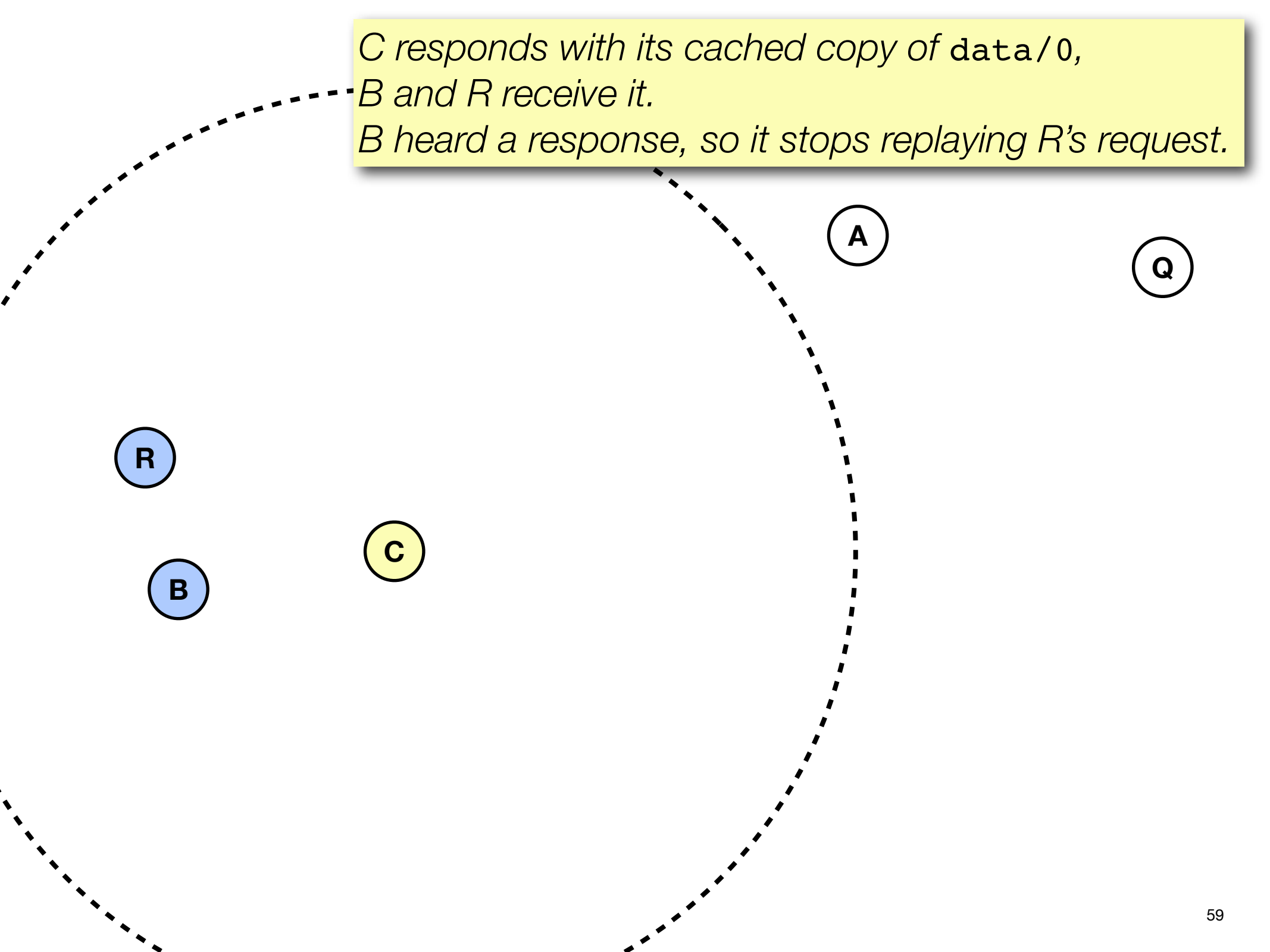
R replays its own request, C receives it.



A

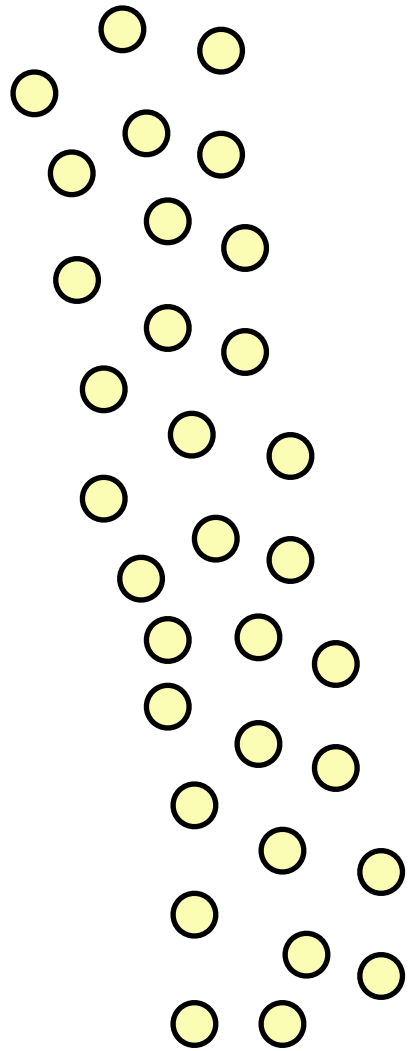
Q

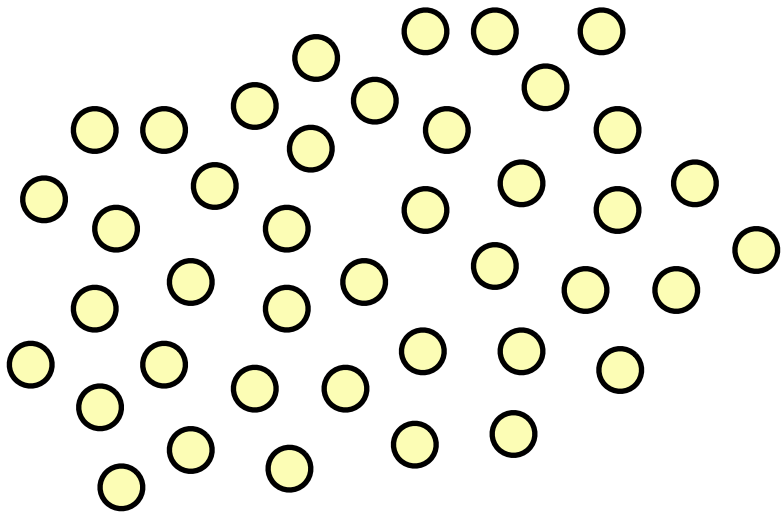
C responds with its cached copy of `data/0`*,*
B and R receive it.
B heard a response, so it stops replaying R's request.



Unifying Connected and Disconnected Networks

Disconnected, but Locally Connected





Limiting Replays

- Request replays are needed to support disconnected networks, but:
    - Request replays are an unnecessary expense within locally connected areas
- **Solution:** Replays should only be used when they are needed by a particular requester

Determining When Replays Are Needed

- In the locally connected network, all responders should hear flooded requests
- No response to repeated flooded requests?
  - The data is not available locally
  - The requester should enable replays for its next request

Avoiding False Positives

- **Problem:** lack of response could also be due to heavy congestion
- Request replay means even more traffic -- bad idea!
- **Solution:** check the portion of the time that the channel is busy (**busy ratio**)
  - High busy ratio: loss could be due to congestion, don't replay
  - Low busy ratio: loss not due to congestion, replay if no response

BOND Evaluation

General Simulation Setup

- QualNet simulator
- Random waypoint model uses steady-state initialization [5]
- To compare traditional routing to BOND's name-based system:
  - Client-server pairs for other protocols, client sends requests at regular intervals, server responds to requests
  - Imitate the above in BOND by never requesting the same data name twice; only one potential responder for each requester

[5] W. Navidi and T. Camp. Stationary distributions for the random waypoint mobility model. Mobile Computing, IEEE Transactions on, 3(1):99 – 108, Jan 2004.

Evaluation Metrics

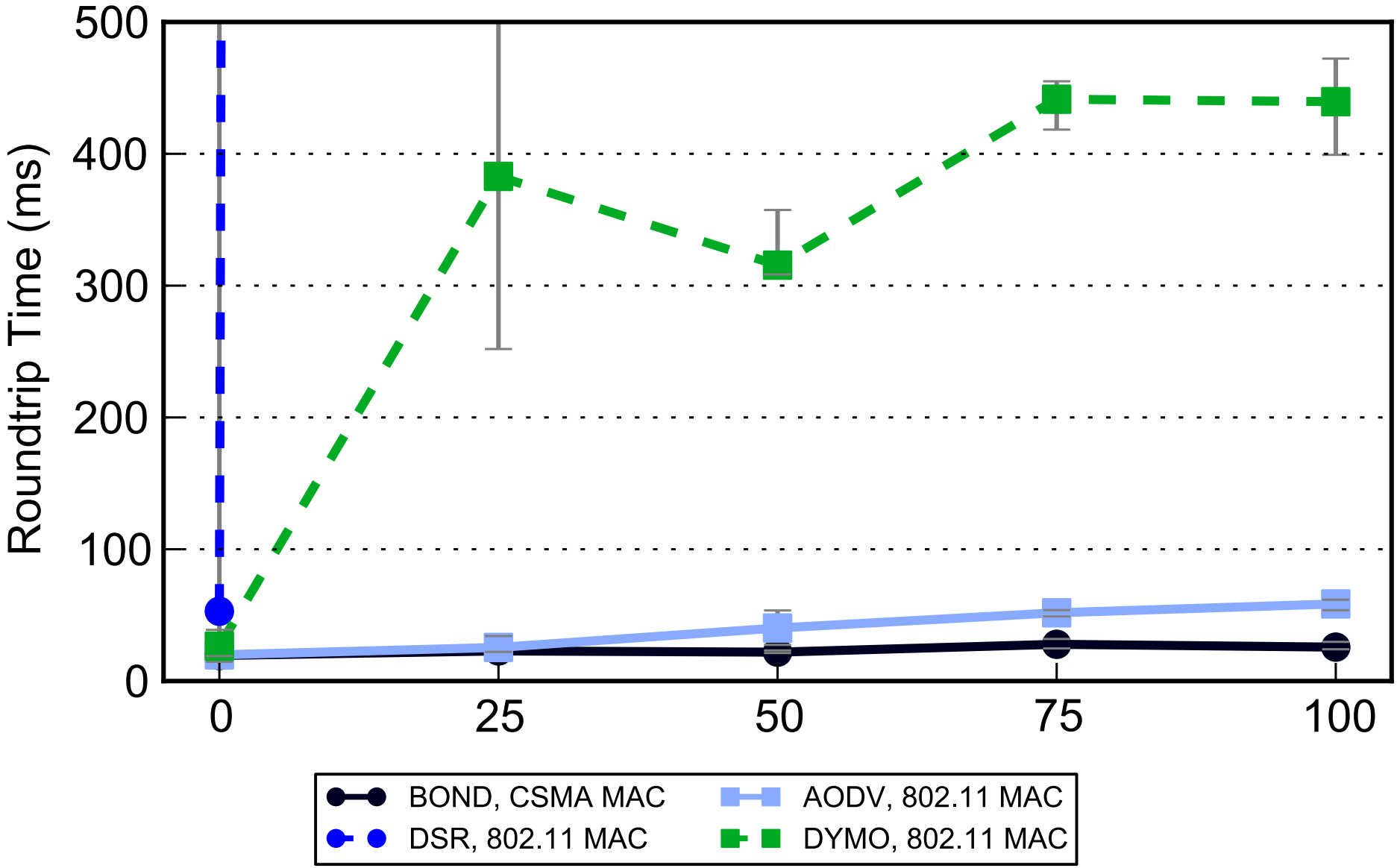
- **Roundtrip time**: Time from request sent to response received
- **Response ratio**: Non-duplicate responses received over requests sent
- **Overhead**: Total bytes transmitted by *all* nodes over non-duplicate response bytes received
- **Path length**: Average length of a *roundtrip* for delivered data (request + response)

BOND Evaluation: Connected Networks

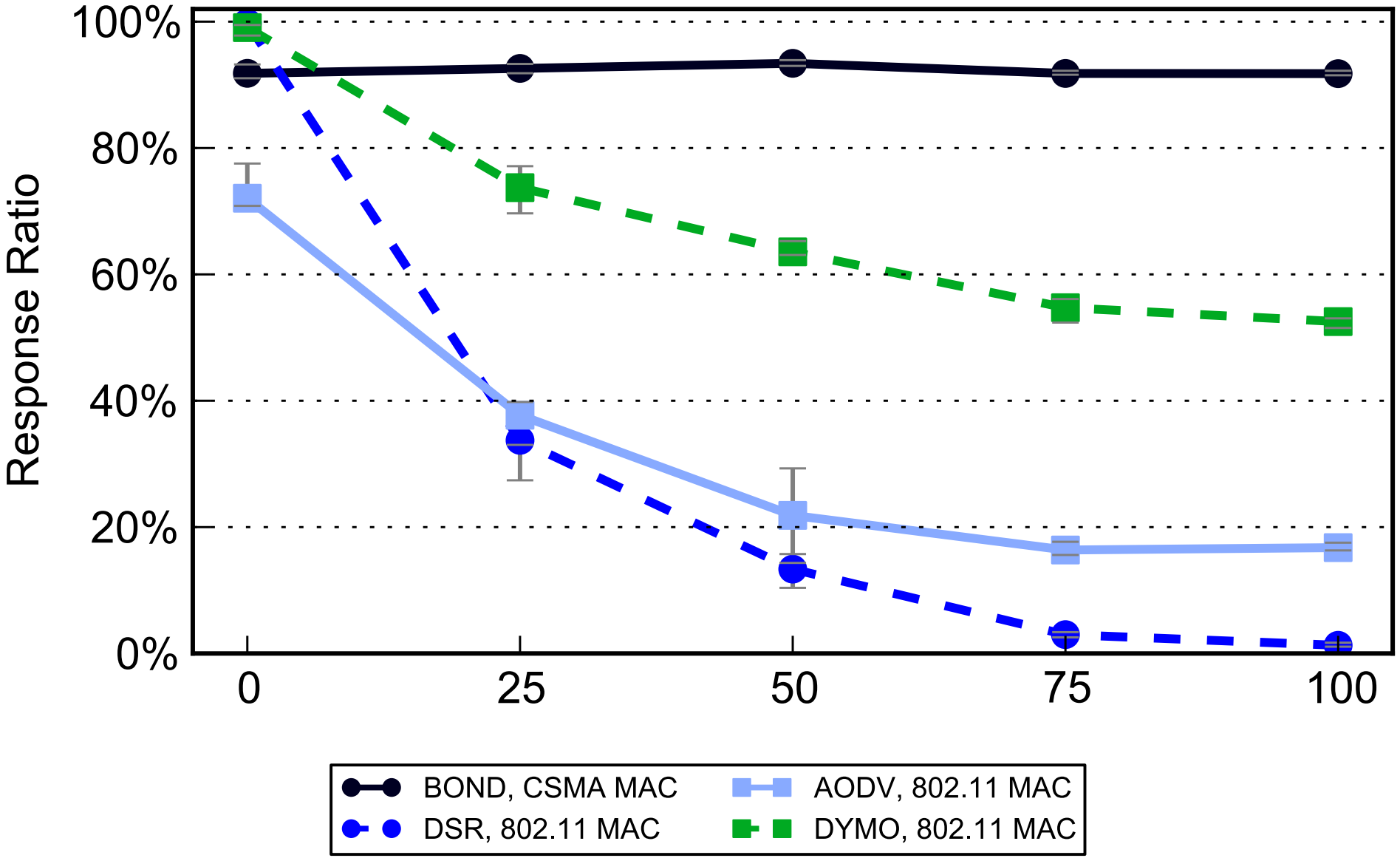
Percent of Nodes Mobile (Setup)

- 100 nodes, 1500 by 1500 meter area, 20 minute duration
- Mobile nodes: Random waypoint mobility, 30 meters per second, no pause time
- Other nodes: stationary
- 8 requester-responder pairs (no overlapping names)
- Requesters send a request every 100 ms
- What happens as the number of mobile nodes increases?

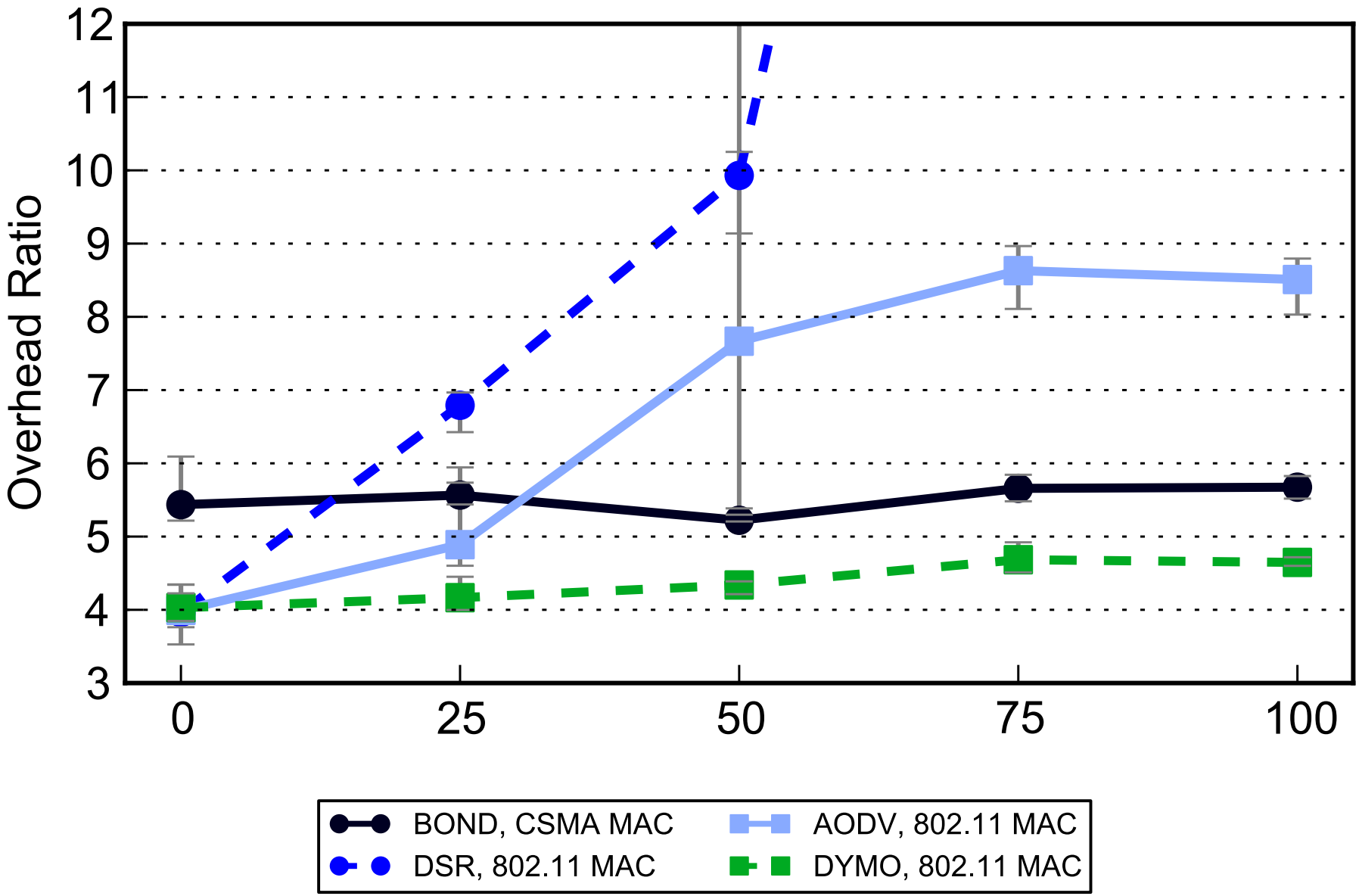
Percent of Nodes Mobile (Latency)



Percent of Nodes Mobile (Response %)



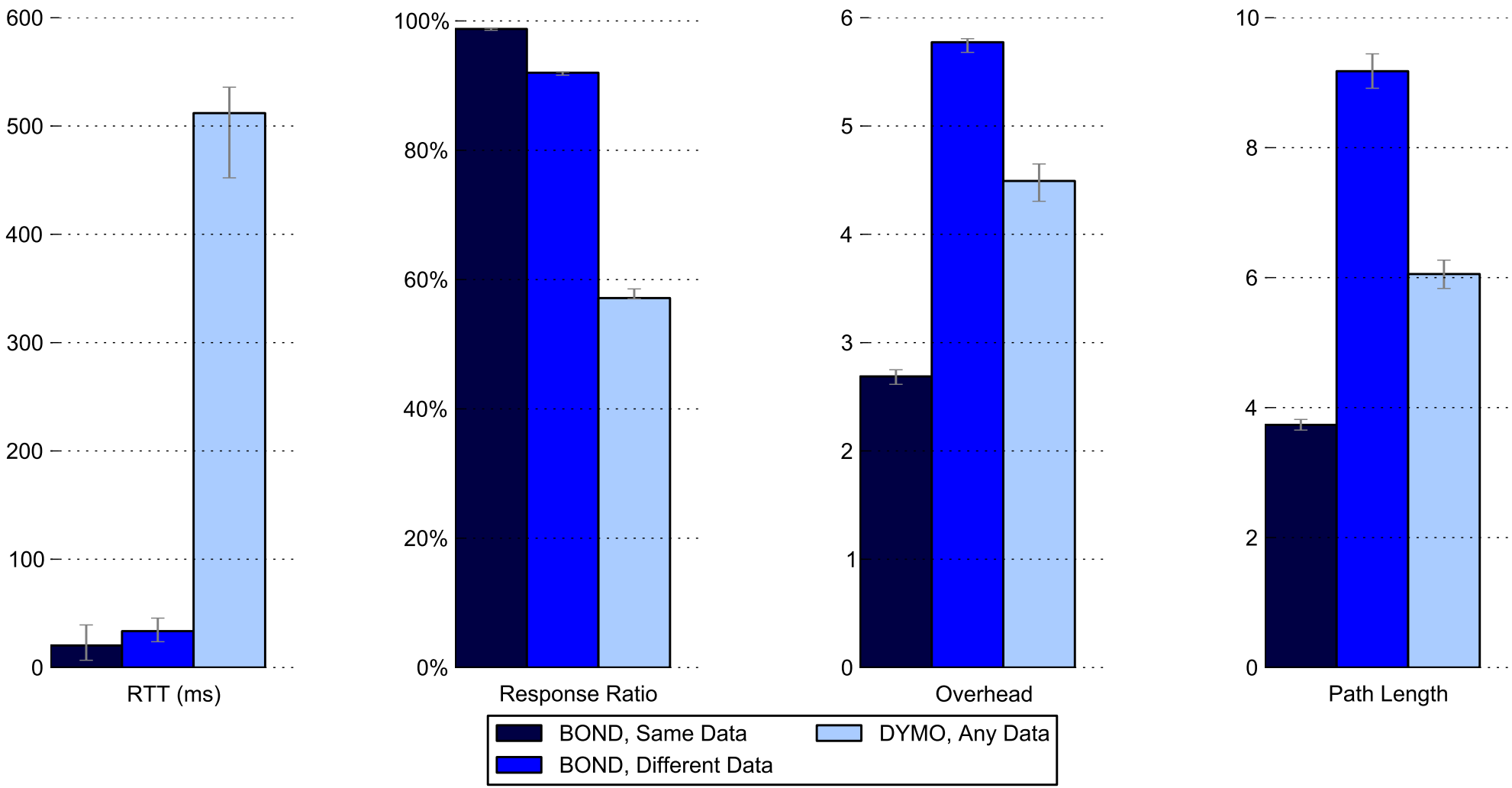
Percent of Nodes Mobile (Overhead)



Caching Named Data (Setup)

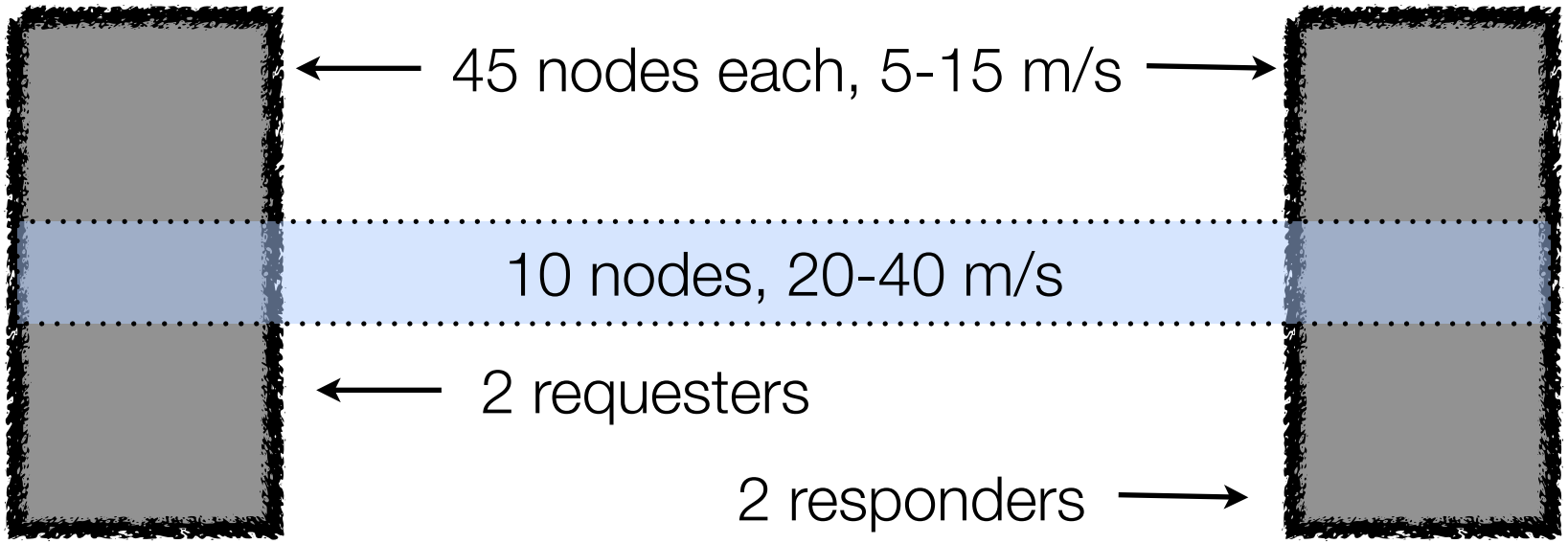
- 100 nodes, 1500 by 1500 meter area, 20 minute duration
- All nodes travel between 10 and 30 meters per second, random waypoint mobility, no pause time
- 8 requesters, 8 potential responders
- What if the requesters all request the same vs. a different sequence of data names?

Caching Named Data



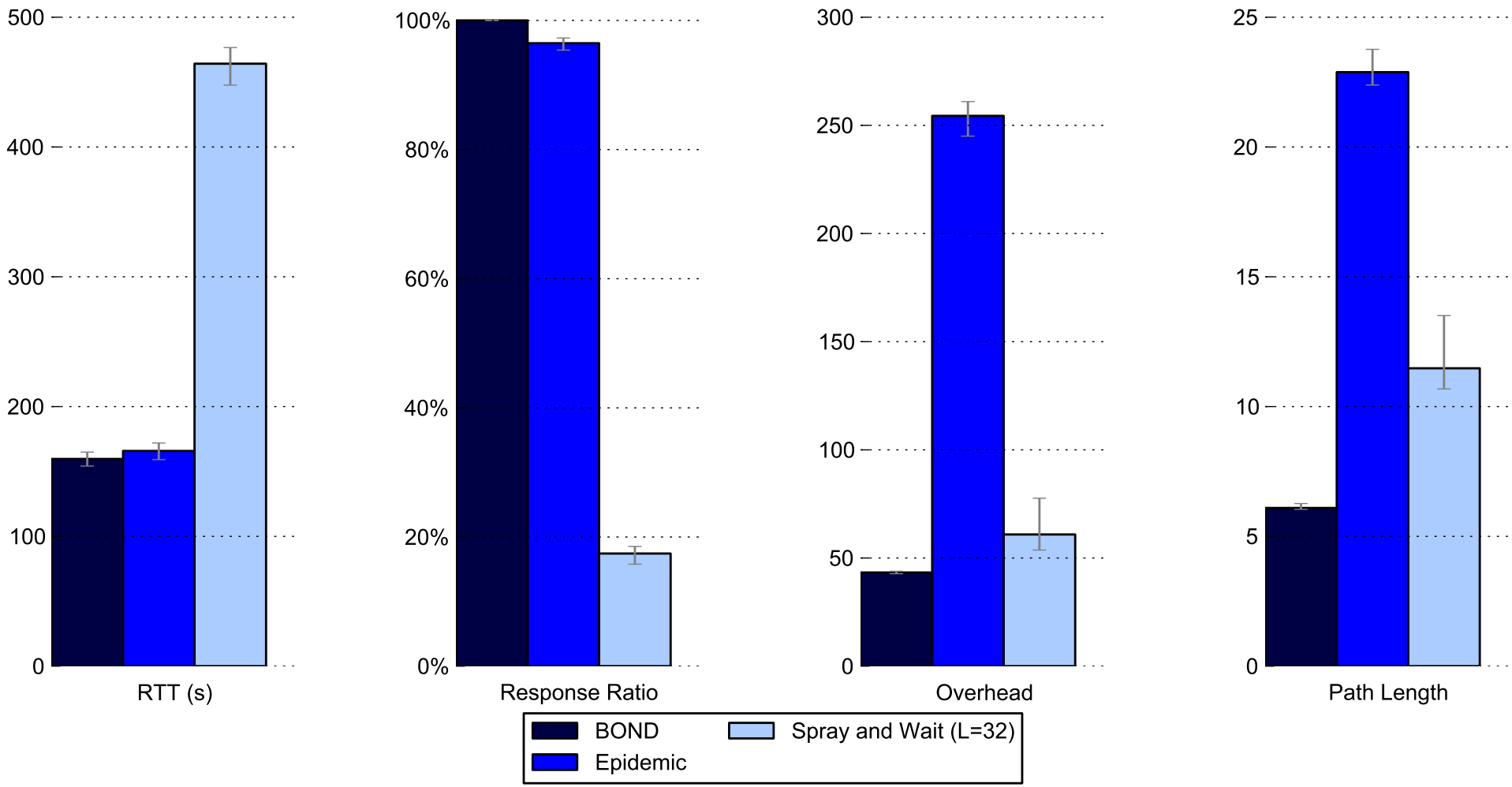
BOND Evaluation: Disconnected Networks

Bridge (Setup)



- Nodes do not change areas, random waypoint within each area
- Comparison protocols: Epidemic Routing, Spray and Wait

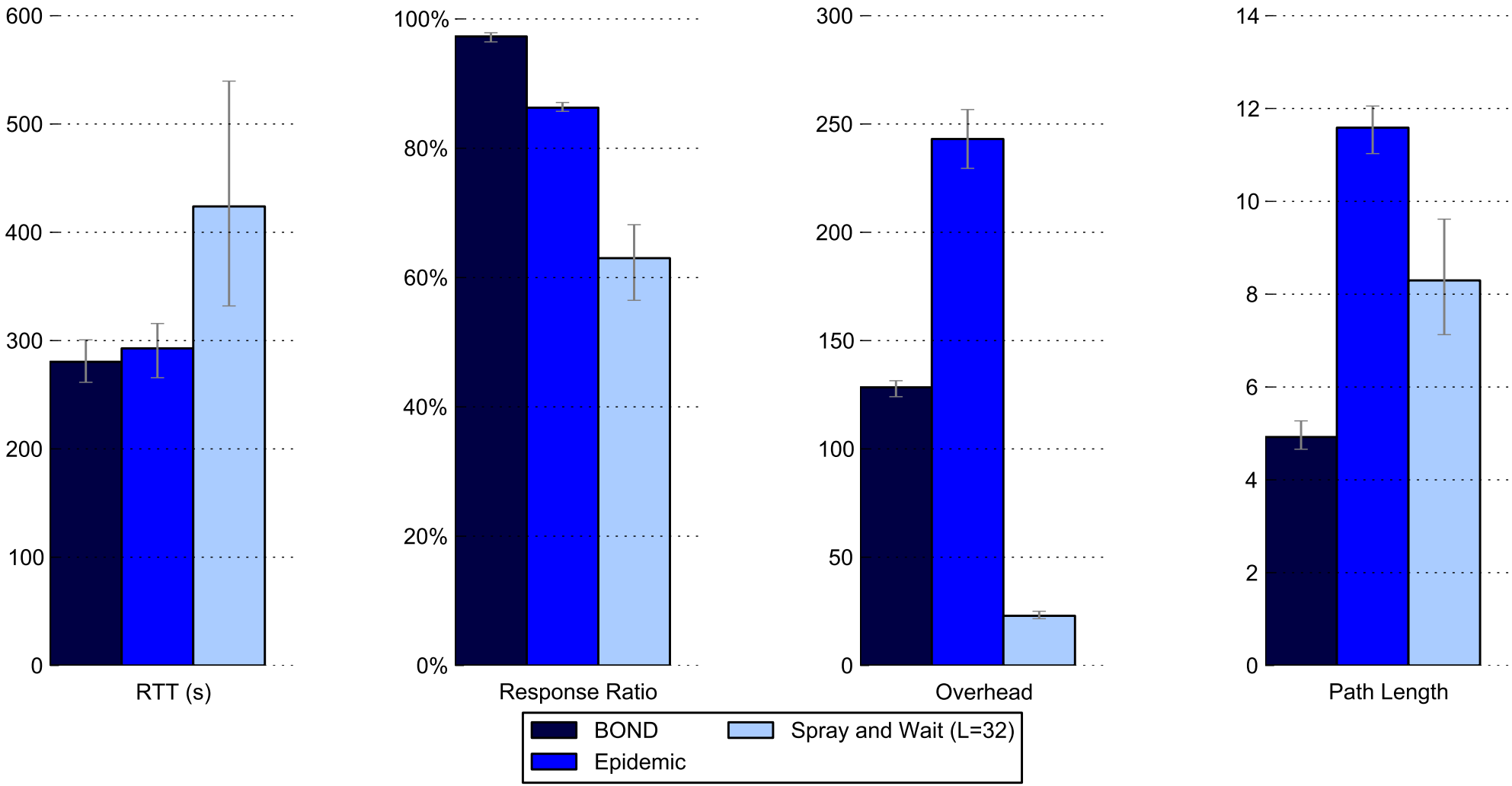
Bridge (Results)



Manhattan (Setup)

- Manhattan mobility model
- 100 nodes, 3000 by 3000 meter area, 20 blocks wide
- 30 min duration
- Nodes travel 10-20 meters per second

Manhattan (Results)



Suggestions for Future Work

- Reducing overhead when request replays are used
- Security
- Data name assignment

Concluding Thoughts

- With the growing number of portable wireless devices, freeform networks could ease strain on infrastructure, but the industry has not made use of them
- BOND removes many of the barriers to deploying multi-hop wireless networks in practice:
  - Devices do not need to be assigned IP addresses
  - The network can be connected or disconnected

Publications and Talks

Dan Jen and Michael Meisel. “APT: A Practical Transit-Mapping Service: Overview and Comparisons.” At *70th Internet Engineering Task Force Meeting, Routing Research Group*. Vancouver, British Columbia, Canada: December 7, 2007.

Dan Jen, Michael Meisel, He Yan, Dan Massey, Lan Wang, Beichuan Zhang, and Lixia Zhang. “Towards a New Internet Routing Architecture: Arguments for Separating Edges from Transit Core.” In *Proceedings of the Seventh ACM Workshop on Hot Topics in Networks (HotNets-VII)*. October 2008.

Dan Jen and Michael Meisel. “APT: An Architecture for Practical Transit Core Separation.” At *Internet Multi-Resolution Analysis Workshop III: Beyond Internet MRA: Network of Networks*. Institute for Pure & Applied Mathematics, Los Angeles, California: November 7, 2008.

Michael Meisel, Vasileios Pappas, and Lixia Zhang. “A Taxonomy of Biologically Inspired Research in Computer Networking.” *Computer Networks*, 54(6): 901 -- 916. April 2010.

Michael Meisel, Vasileios Pappas, and Lixia Zhang. “Ad Hoc Networking via Named Data.” In *Proceedings of the Fifth ACM Workshop on Mobility in the Evolving Internet Architecture (MobiArch)*. September 2010.