REUNITE: A Recursive Unicast Approach to Multicast

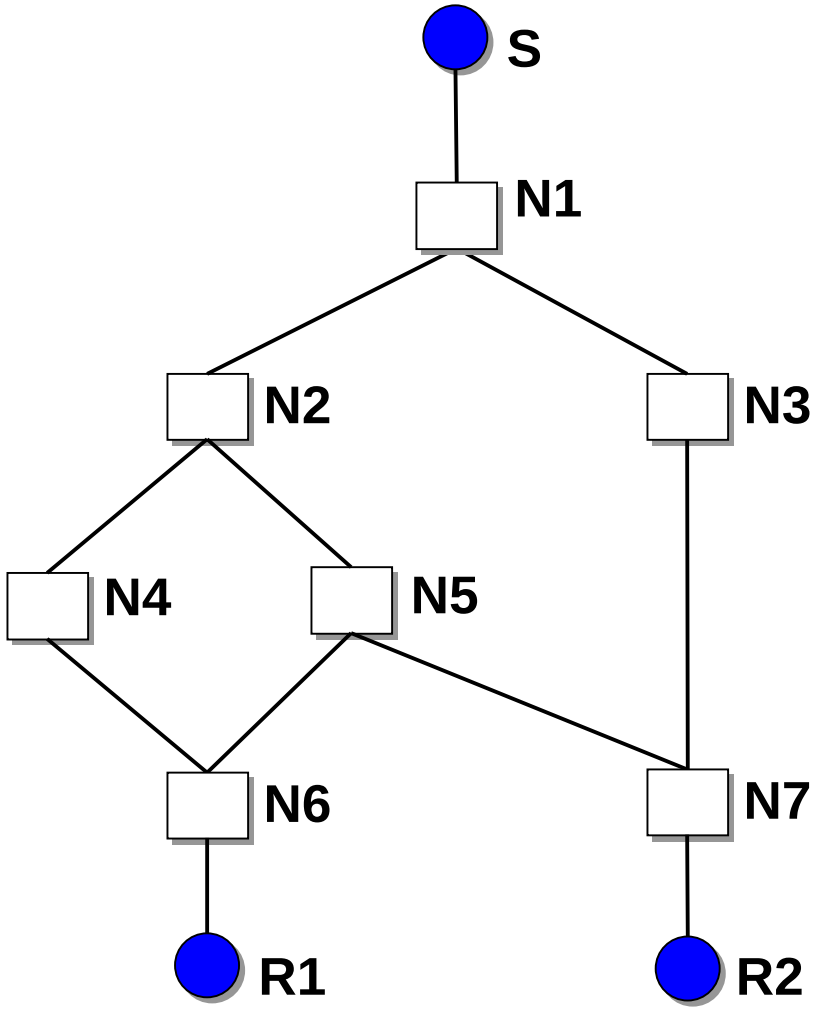
Ion Stoica, T.S. Eugene Ng, Hui Zhang
Carnegie Mellon University
{istoica,eugeneng,hzhang}@cs.cmu.edu

IP Multicast

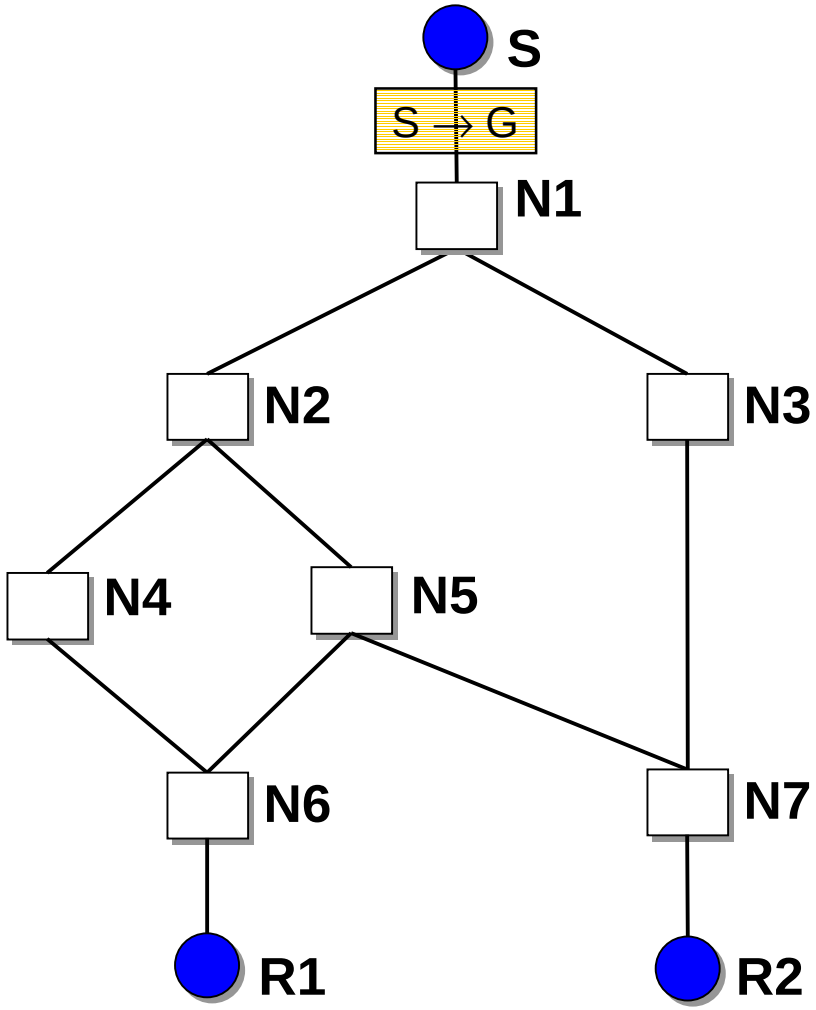
- Open system model
  - anyone can send to a group
  - anyone can listen to a group
  - dynamic join and leave
- Group membership management performed by the network

- Uses **logical class D** IP addresses
  - group identification
  - packet forwarding

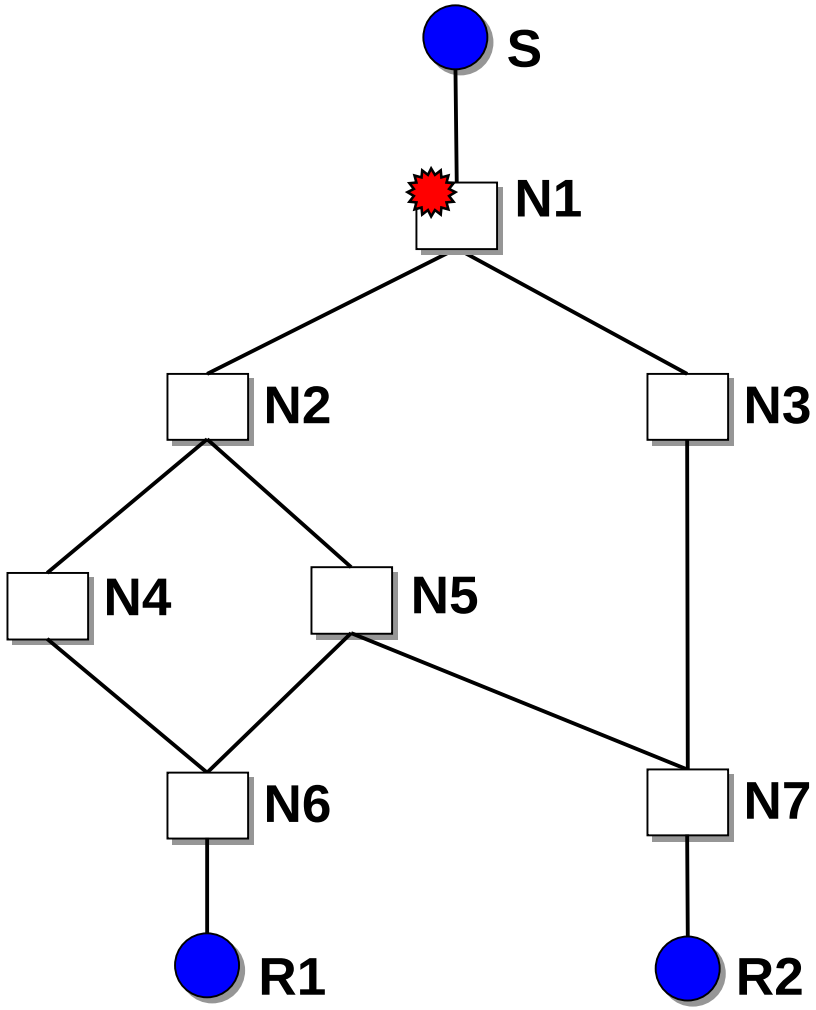
IP Multicast Packet Forwarding



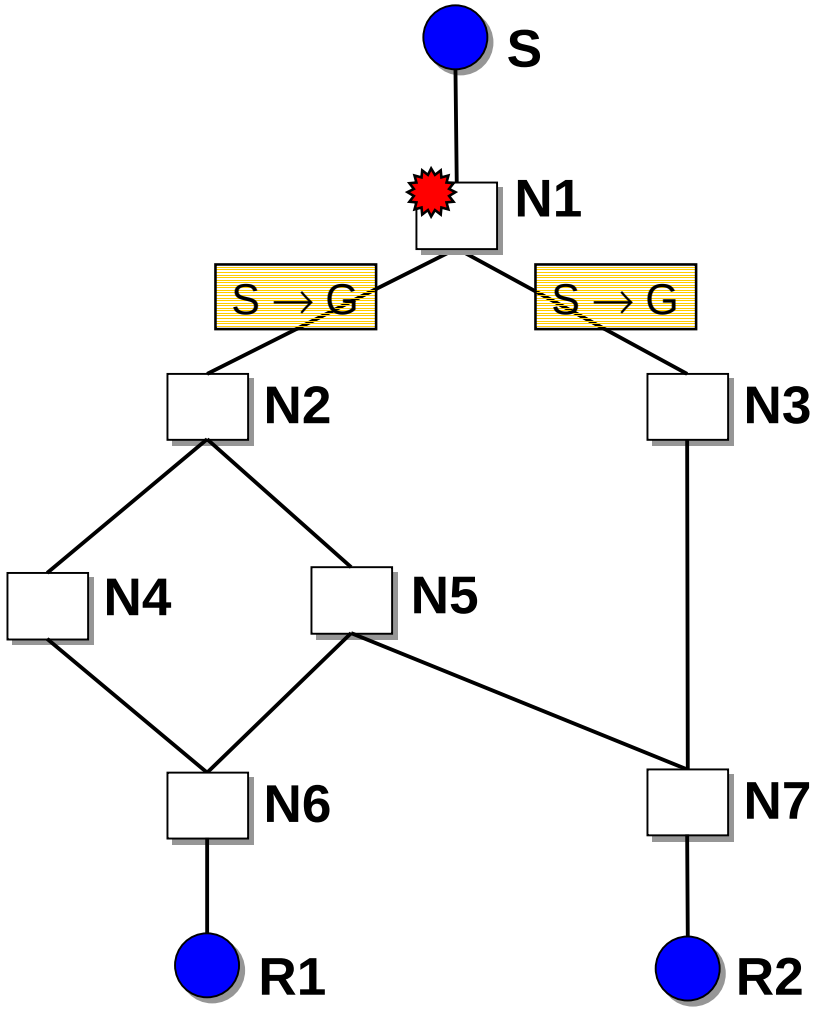
IP Multicast Packet Forwarding



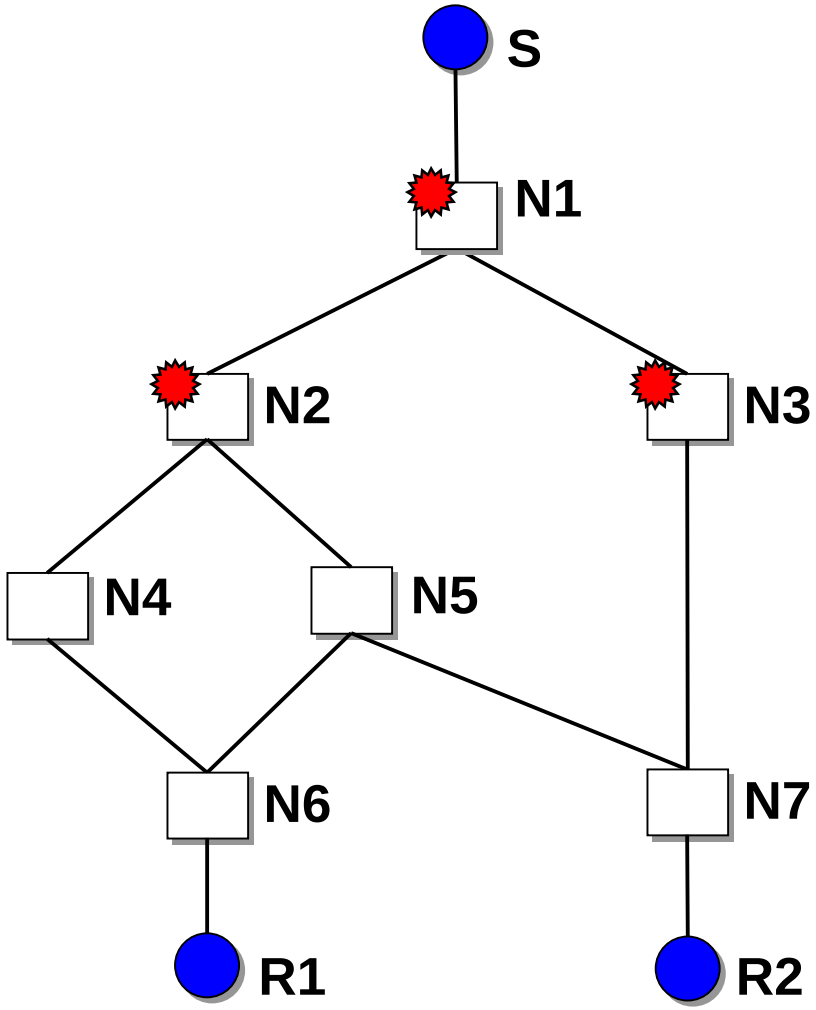
IP Multicast Packet Forwarding



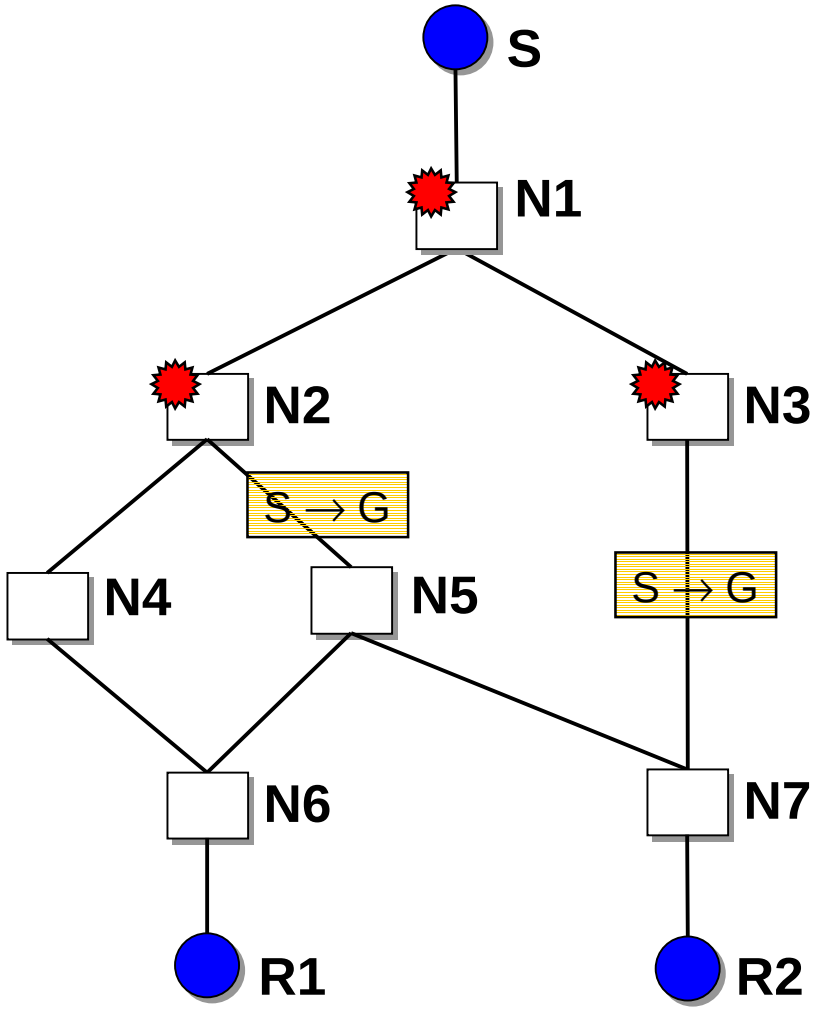
IP Multicast Packet Forwarding



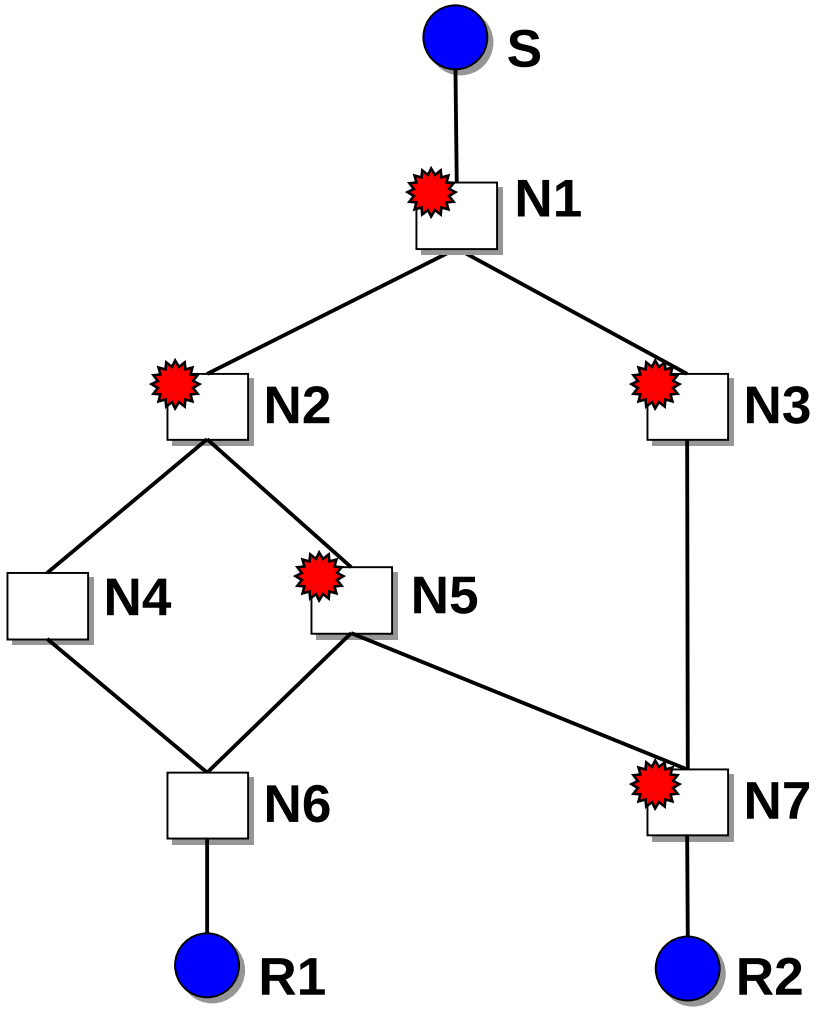
IP Multicast Packet Forwarding



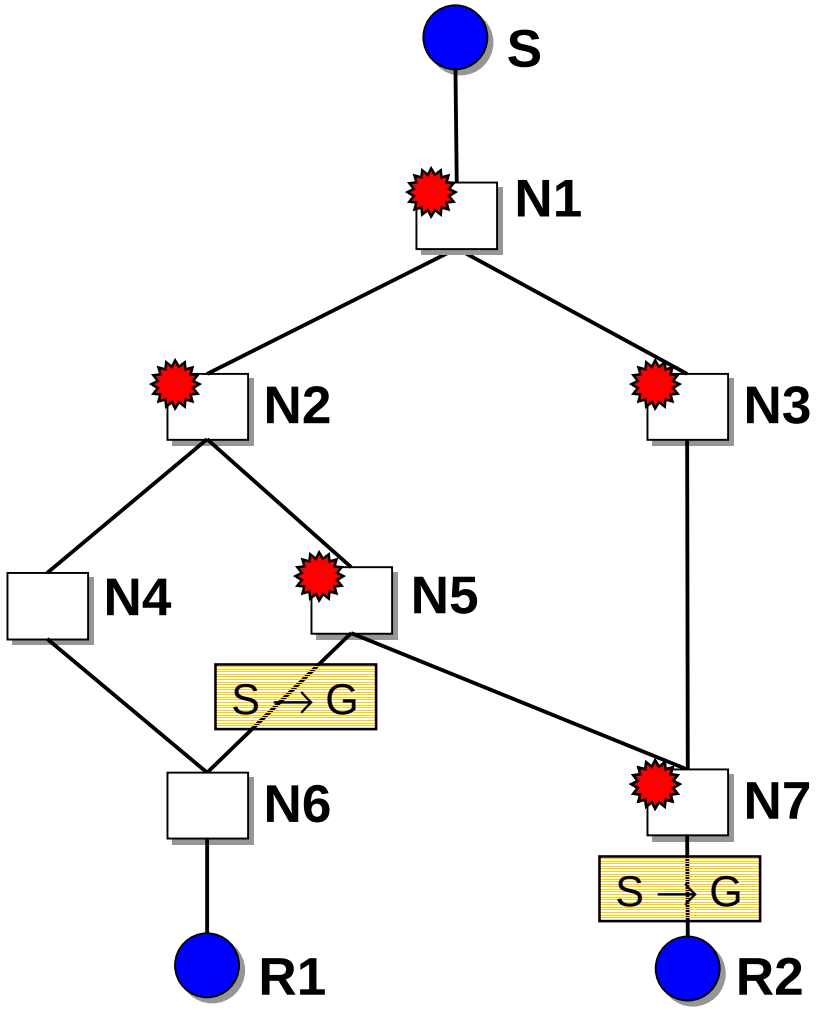
IP Multicast Packet Forwarding



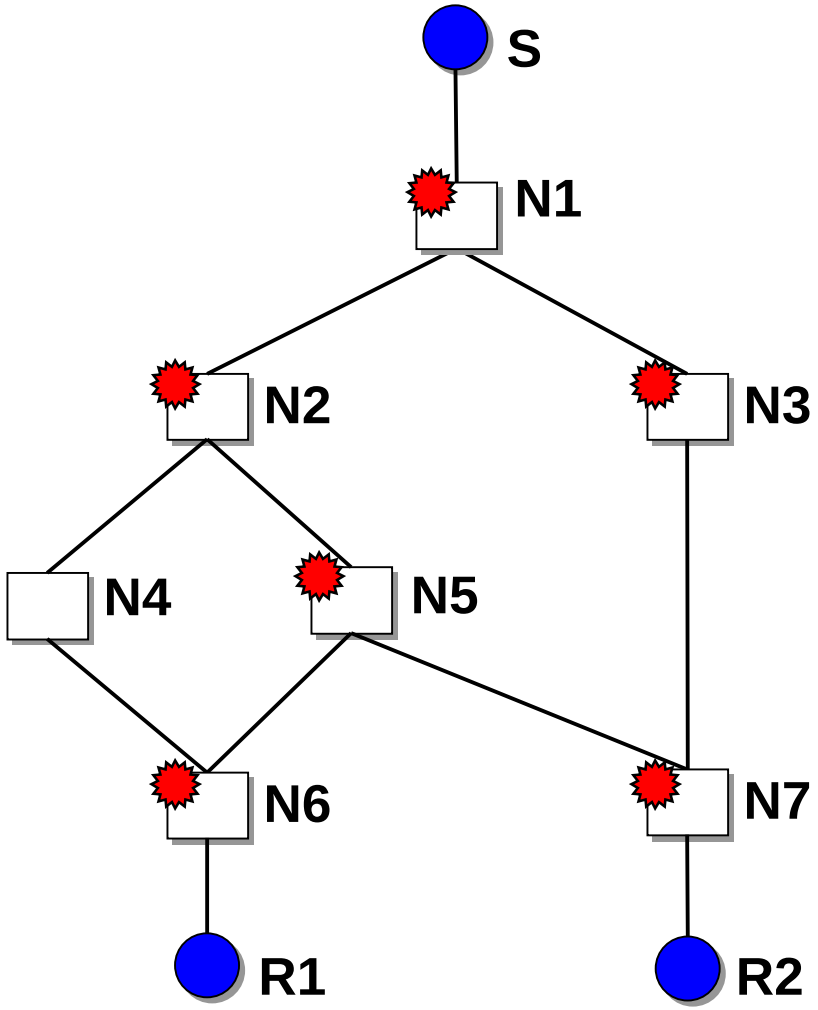
IP Multicast Packet Forwarding



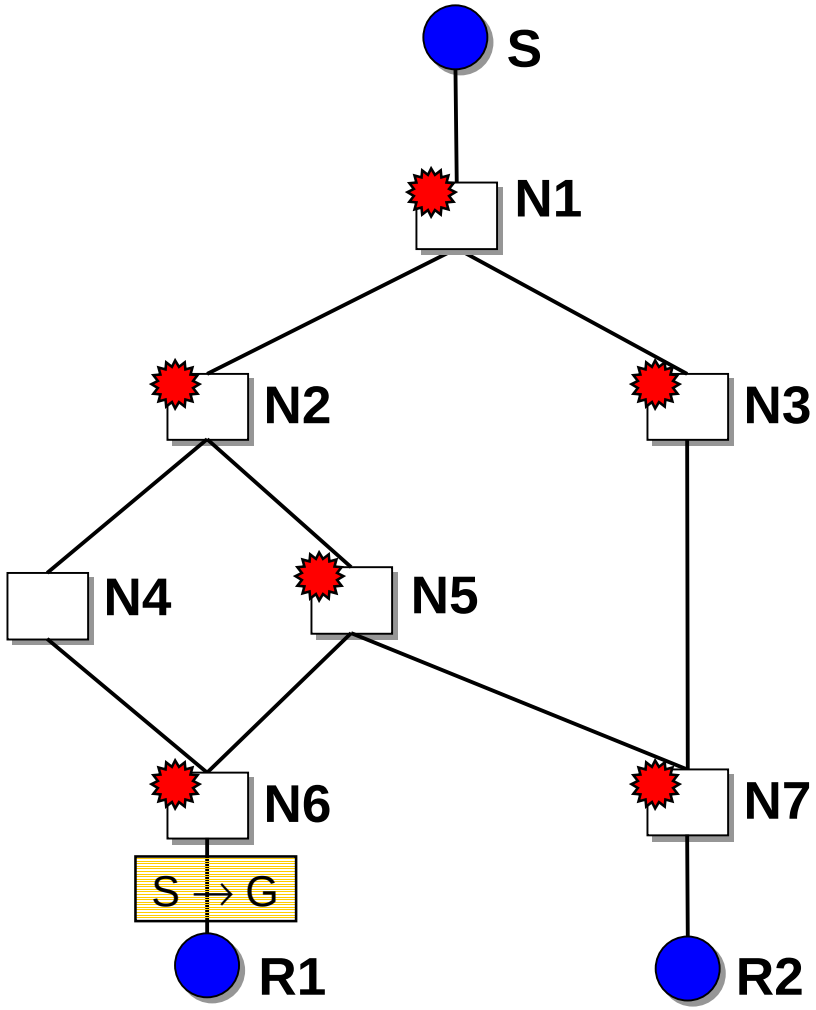
IP Multicast Packet Forwarding



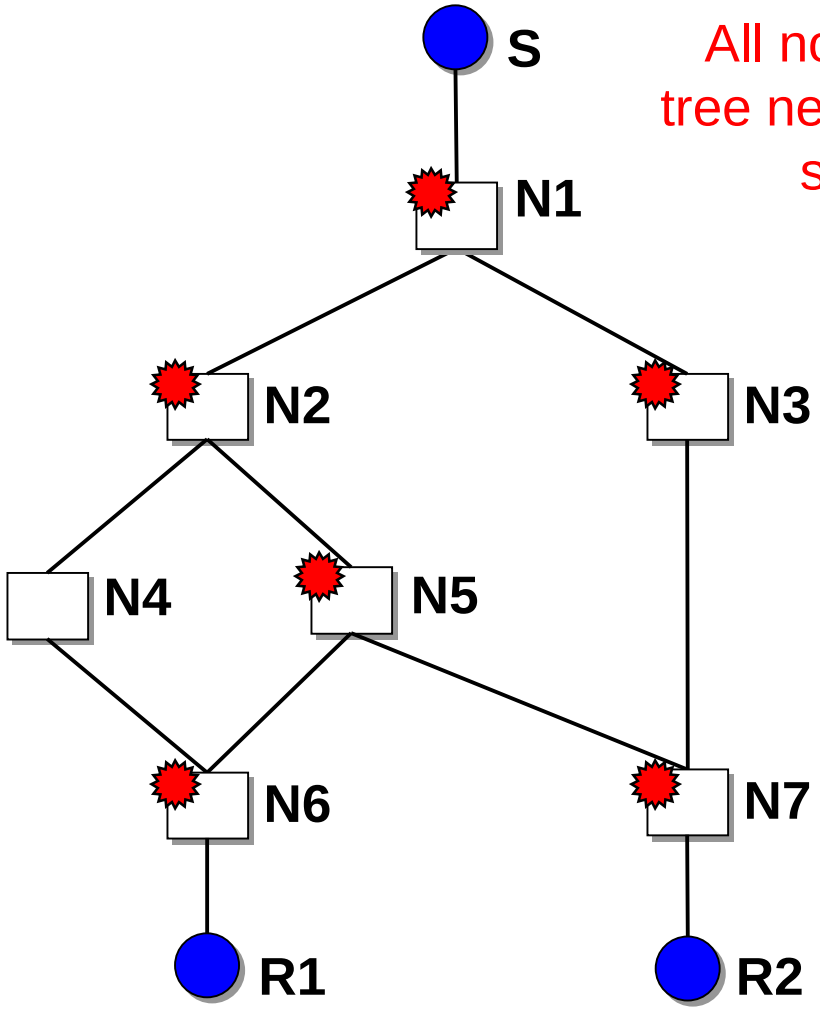
IP Multicast Packet Forwarding



IP Multicast Packet Forwarding



IP Multicast Packet Forwarding



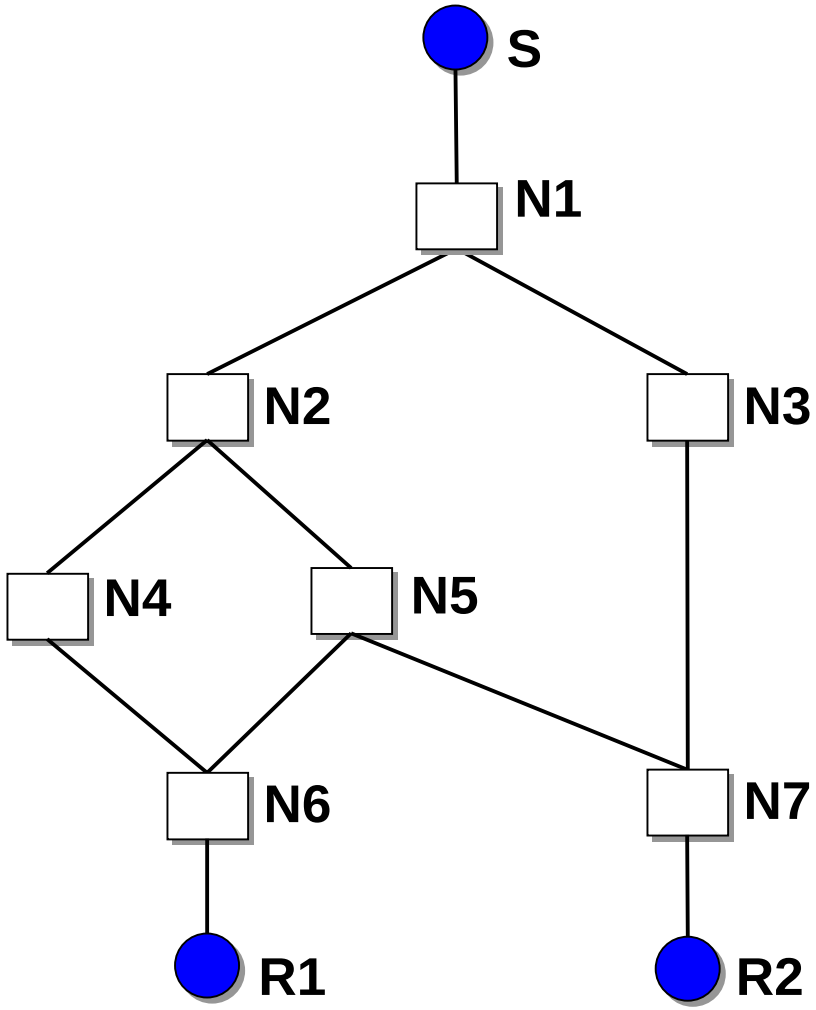
All nodes in the multicast tree need to have forwarding state for group G

IP Multicast vs REUNITE

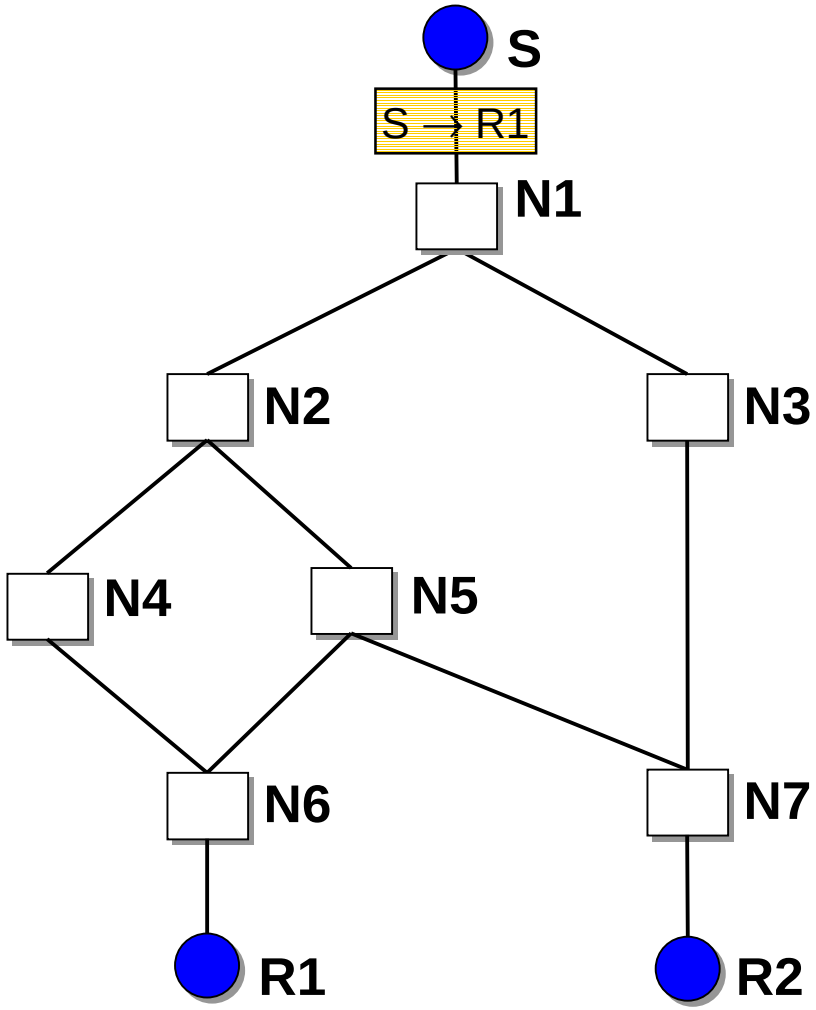
- IP multicast uses **logical class D** IP addresses
  - group identification
  - packet forwarding

- REUNITE uses **topological unicast** IP addresses
  - group identification
  - packet forwarding

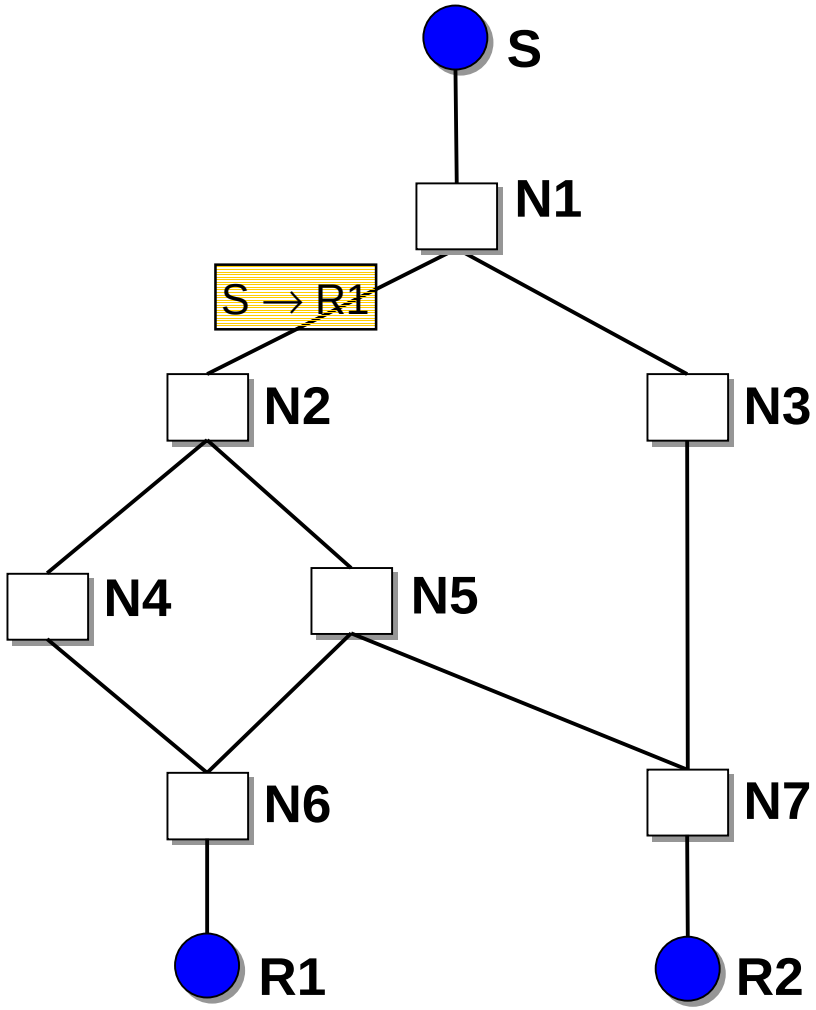
REUNITE Recursive Unicast Packet Forwarding



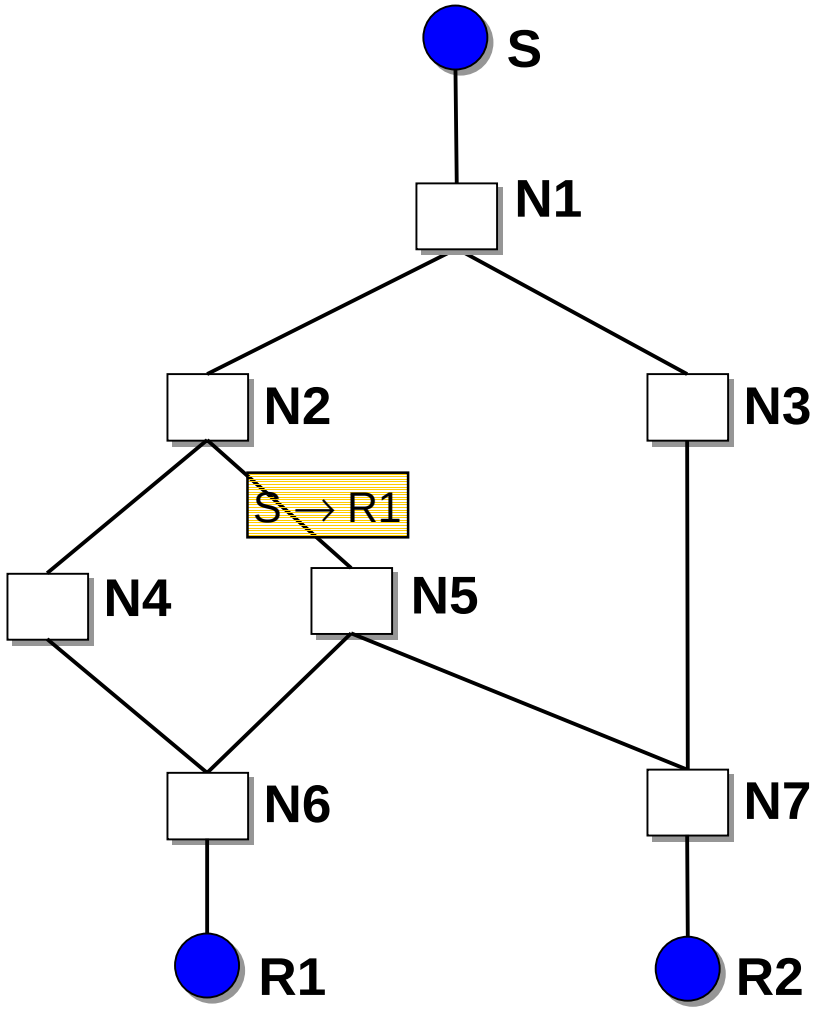
REUNITE Recursive Unicast Packet Forwarding



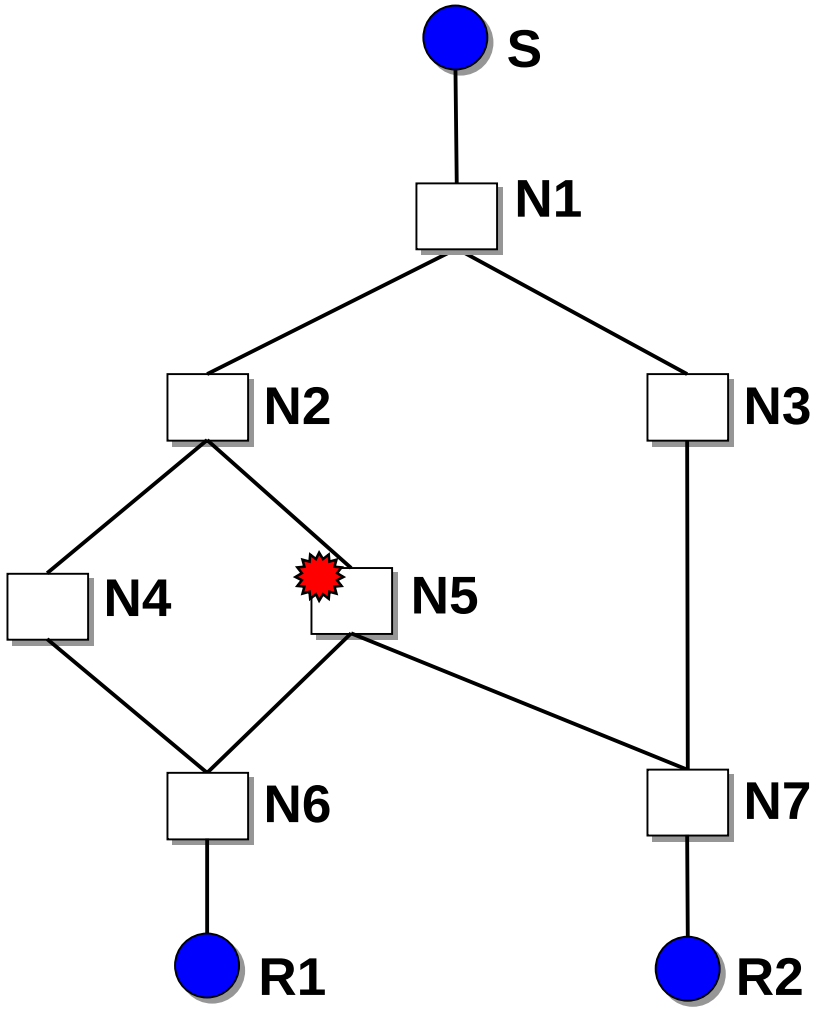
REUNITE Recursive Unicast Packet Forwarding



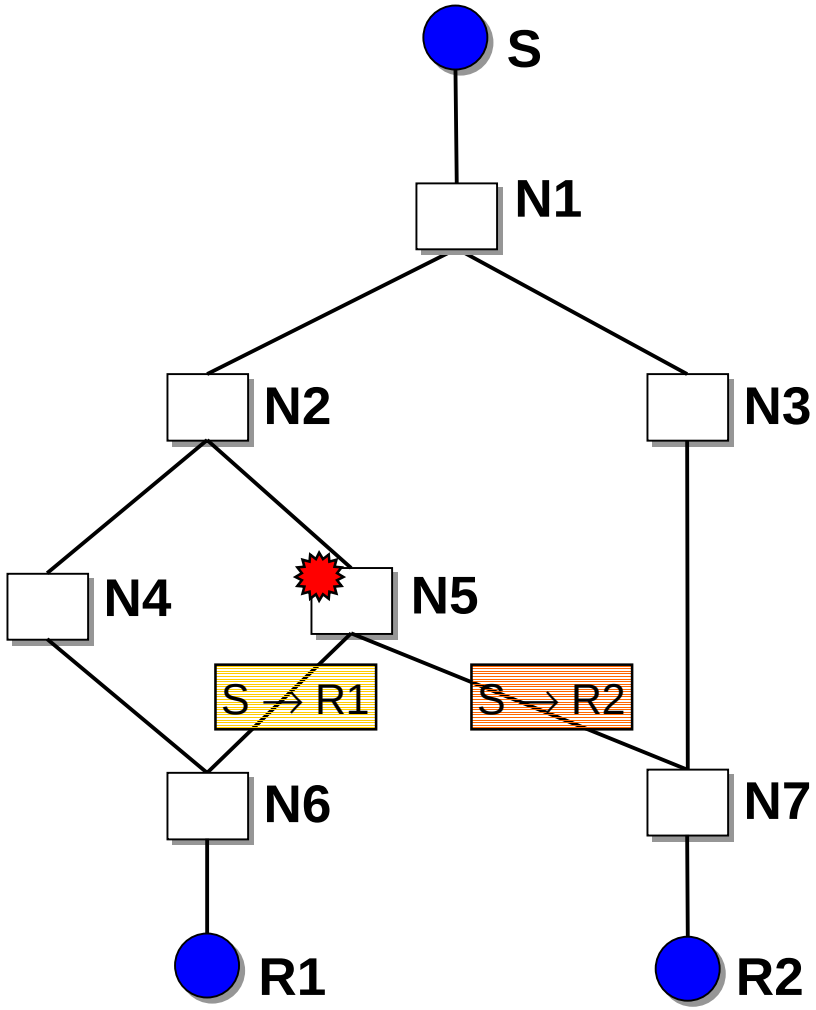
REUNITE Recursive Unicast Packet Forwarding



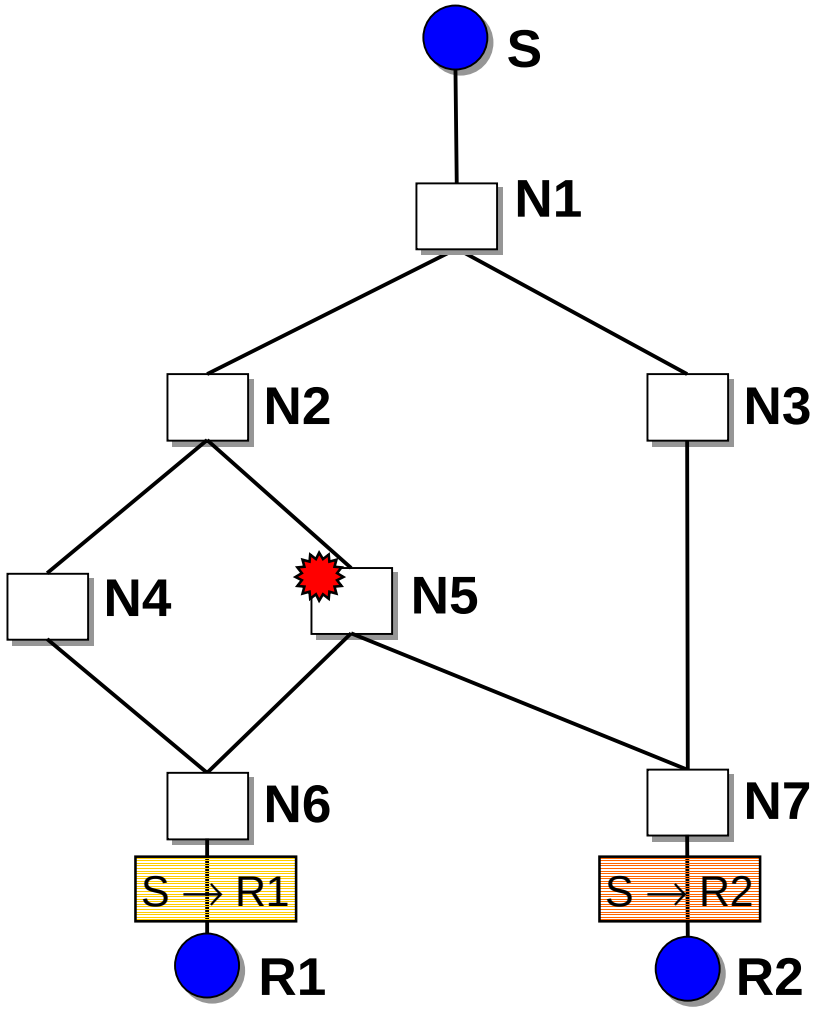
REUNITE Recursive Unicast Packet Forwarding



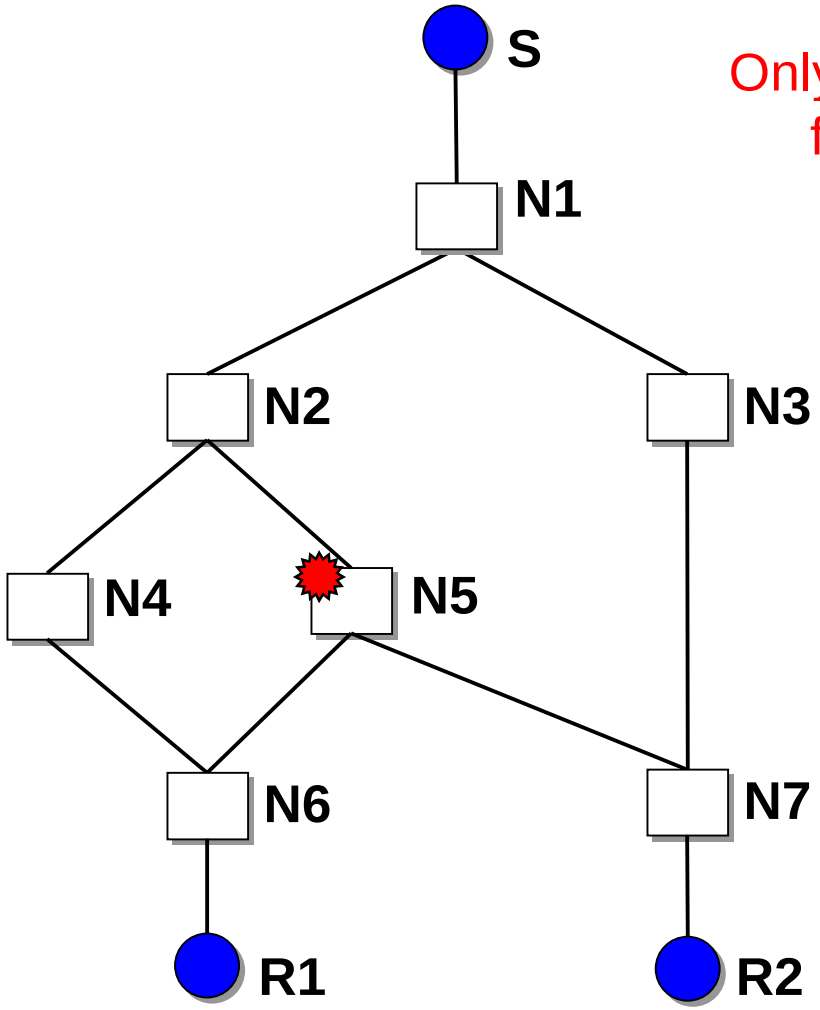
REUNITE Recursive Unicast Packet Forwarding



REUNITE Recursive Unicast Packet Forwarding

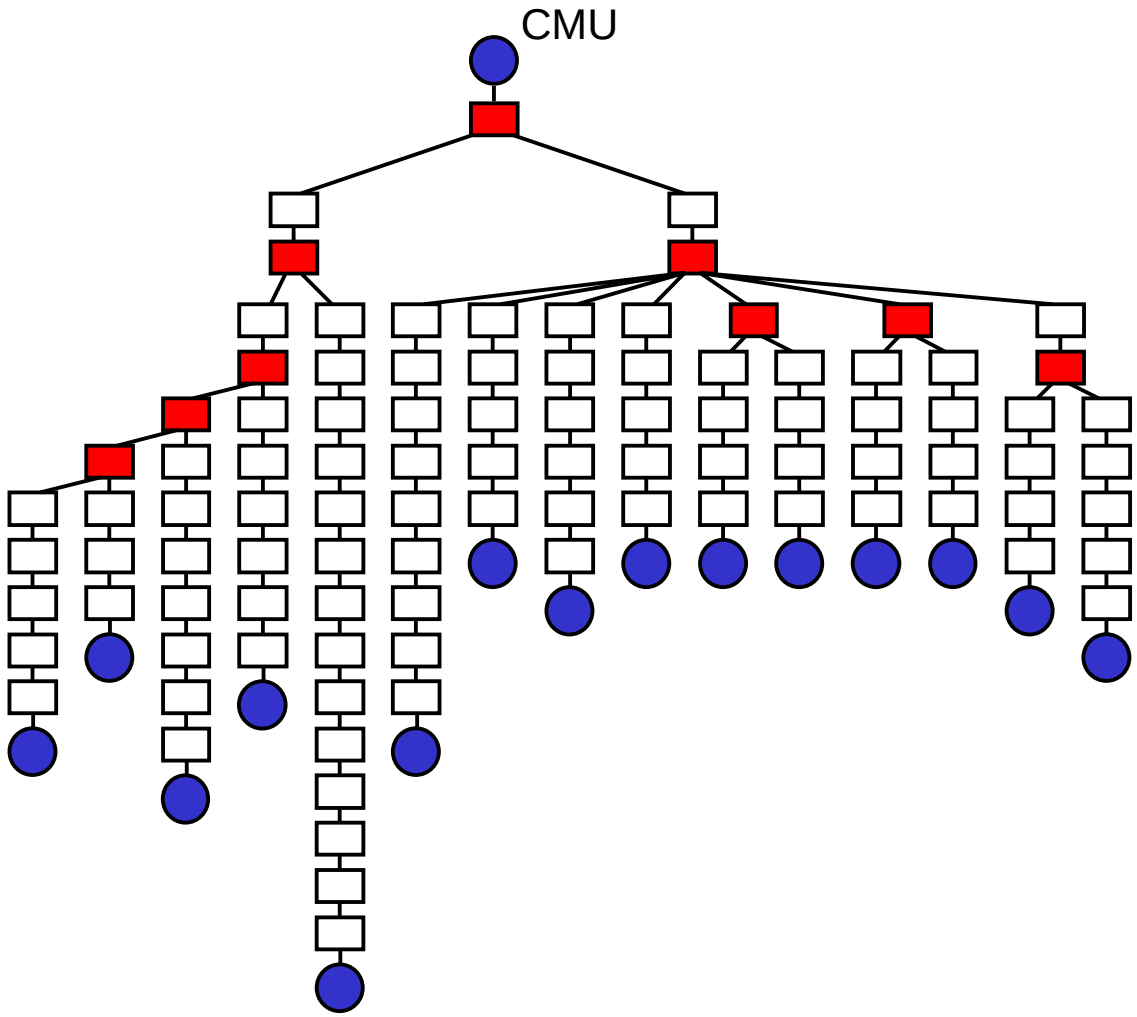


REUNITE Recursive Unicast Packet Forwarding



Only N5 needs to have forwarding state for the group

Why Is This Important?



REUNITE Highlights

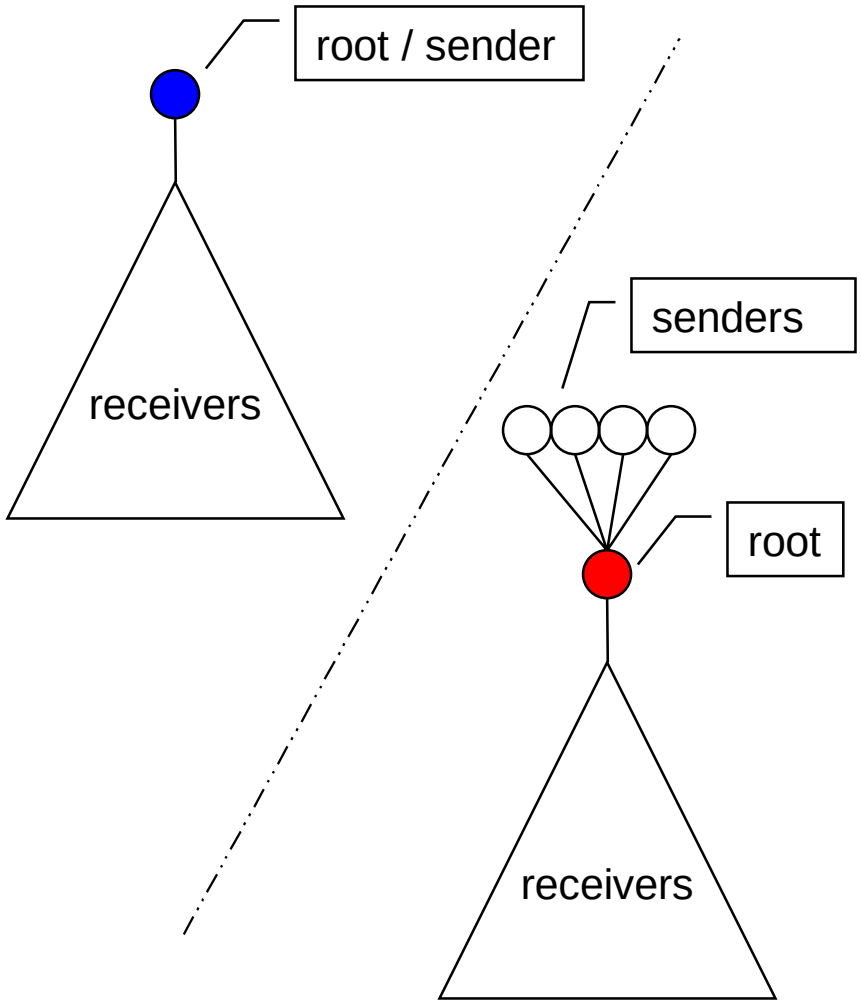
- Improves scalability by removing unneeded multicast forwarding state
- Native support for incremental deployment
- Load balancing and graceful degradation
- No multicast address allocation problem
- Support for access control

Outline

- Group addressing
- Data Path
  - Multicast Forwarding Table (MFT) and packet forwarding
- Control Path
  - Multicast Control Table (MCT) and tree maintenance protocol
- Advantages
- Related Work
- Conclusions

Group Addressing

- A group is identified by the **root**'s IP address and a **root** port number
  - <**root_IP**, **root_port**>
- Root is a general multicast data source
- 1. The root can be the sender of a group
- 2. The root can be a special node that relay packets from sender(s) of a group



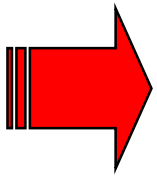
Data Path

Multicast Forwarding Table (MFT)

<root_IP, root_port>	rcvr_IP	rcvr_IP$_1$, rcvr_IP$_2$, ...

Group ID / Source | Destination | Receivers list

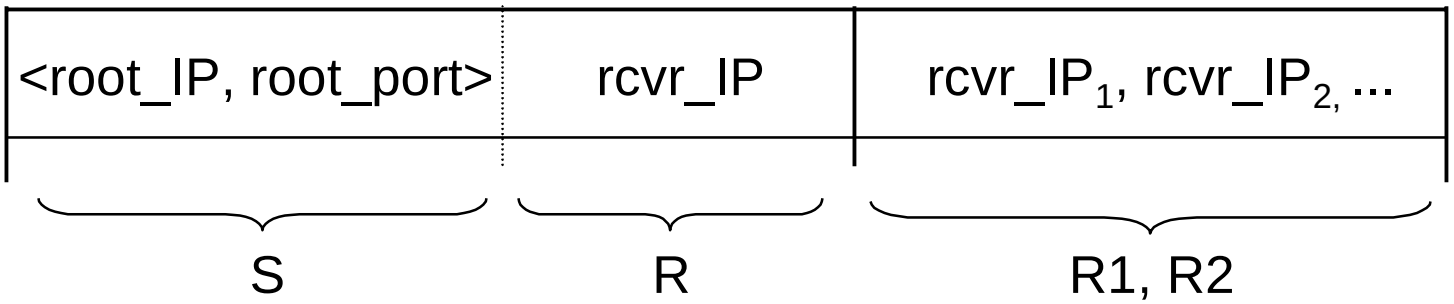
src IP = root_IP
dst IP = rcvr_IP
src port = root_port

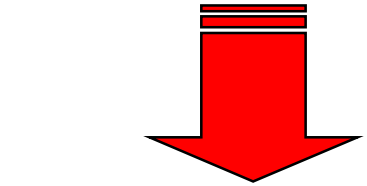


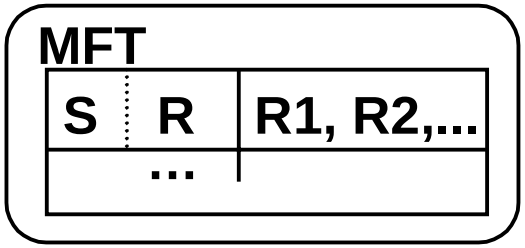
src IP = root_IP
dst IP = rcvr_IP$_1$
src port = root_port

src IP = root_IP
dst IP = rcvr_IP$_2$
src port = root_port

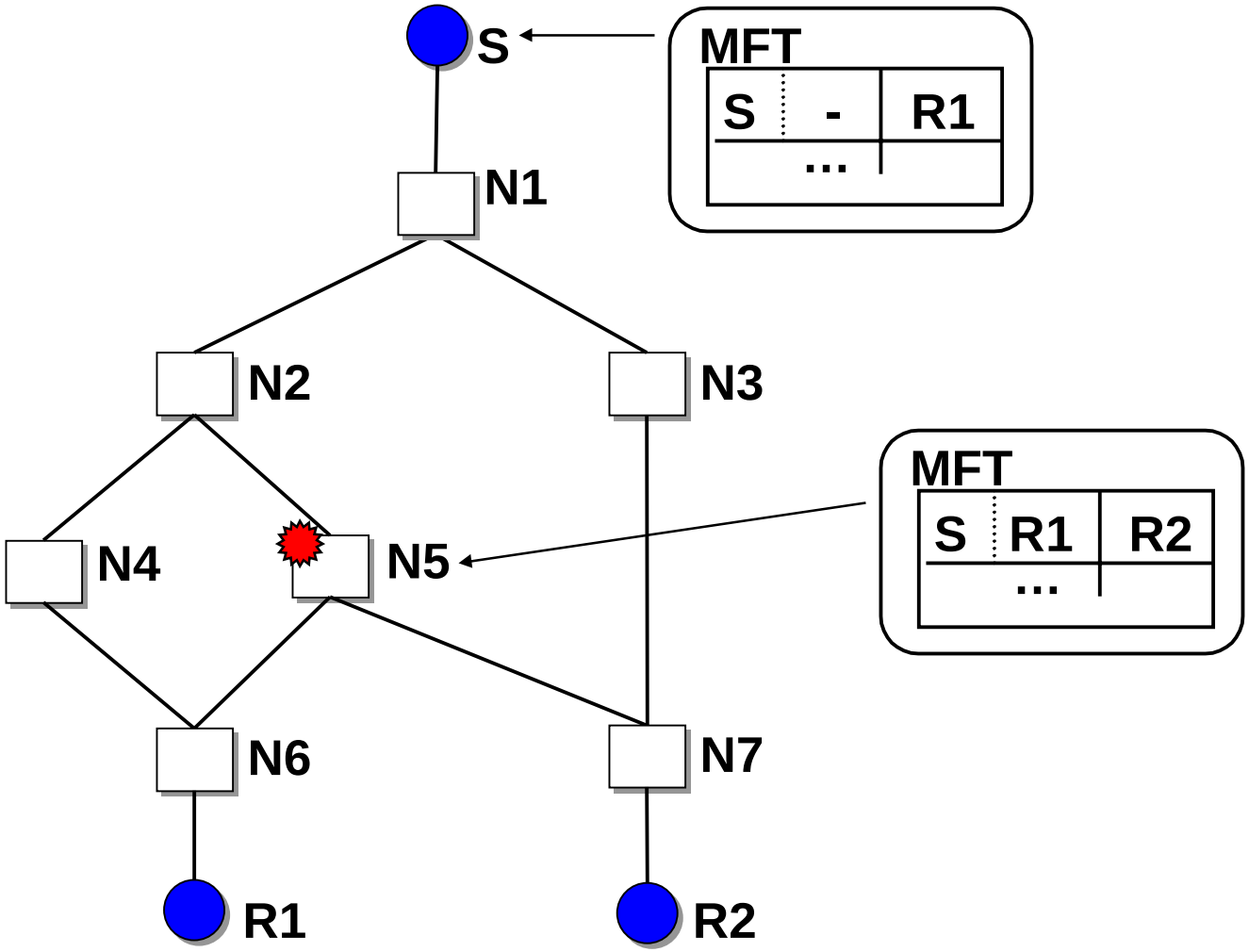
MFT Abbreviation



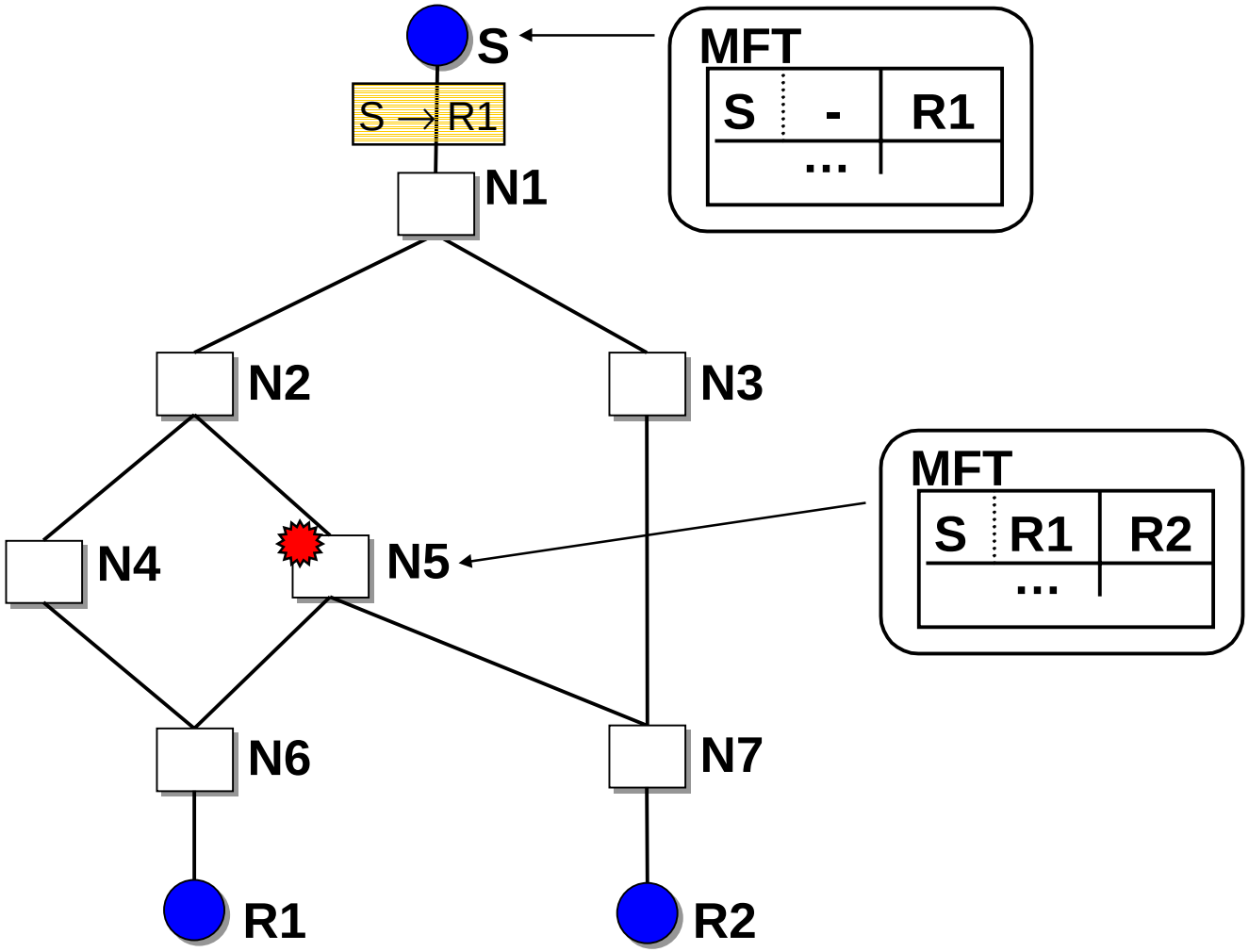




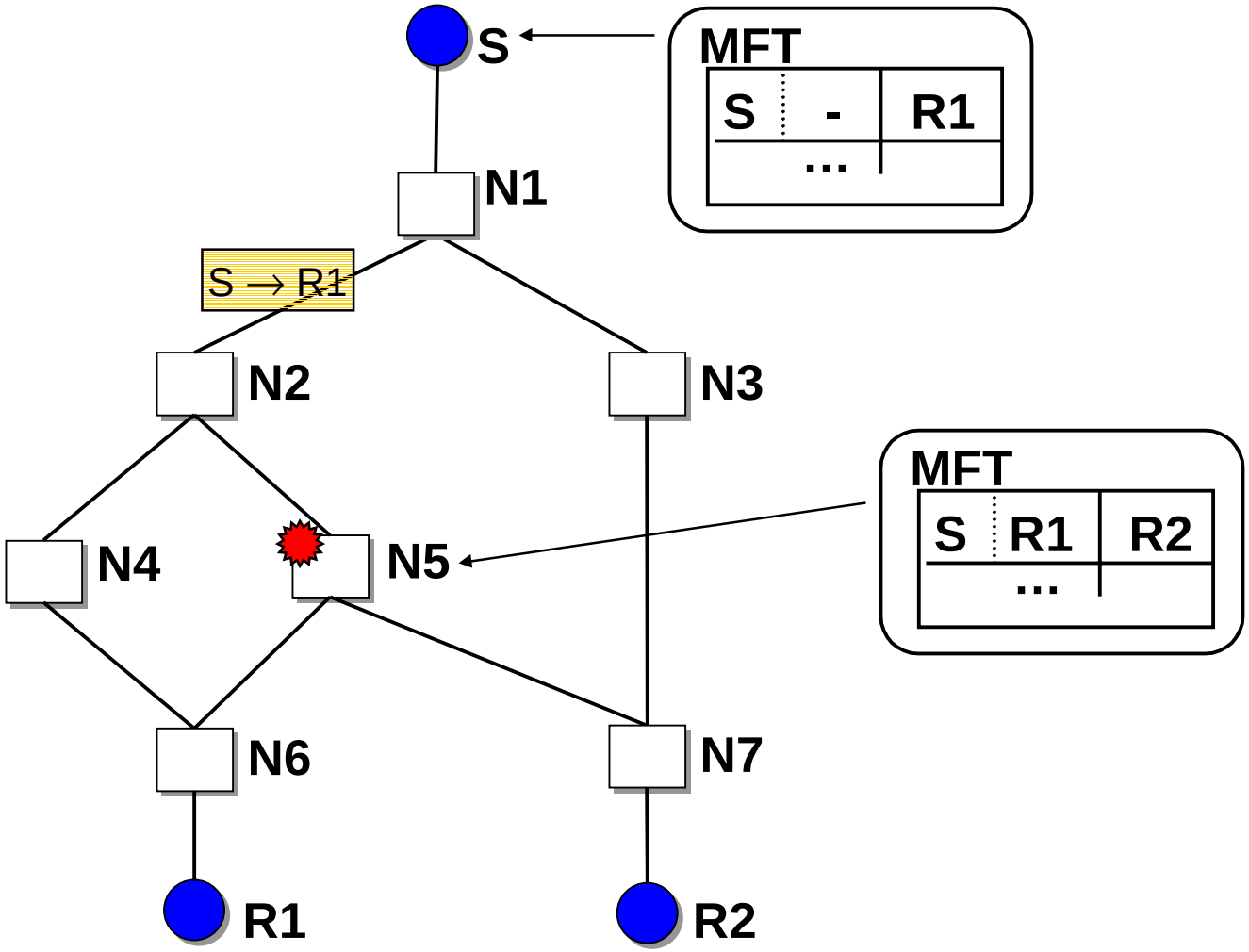
Using MFT in Packet Forwarding



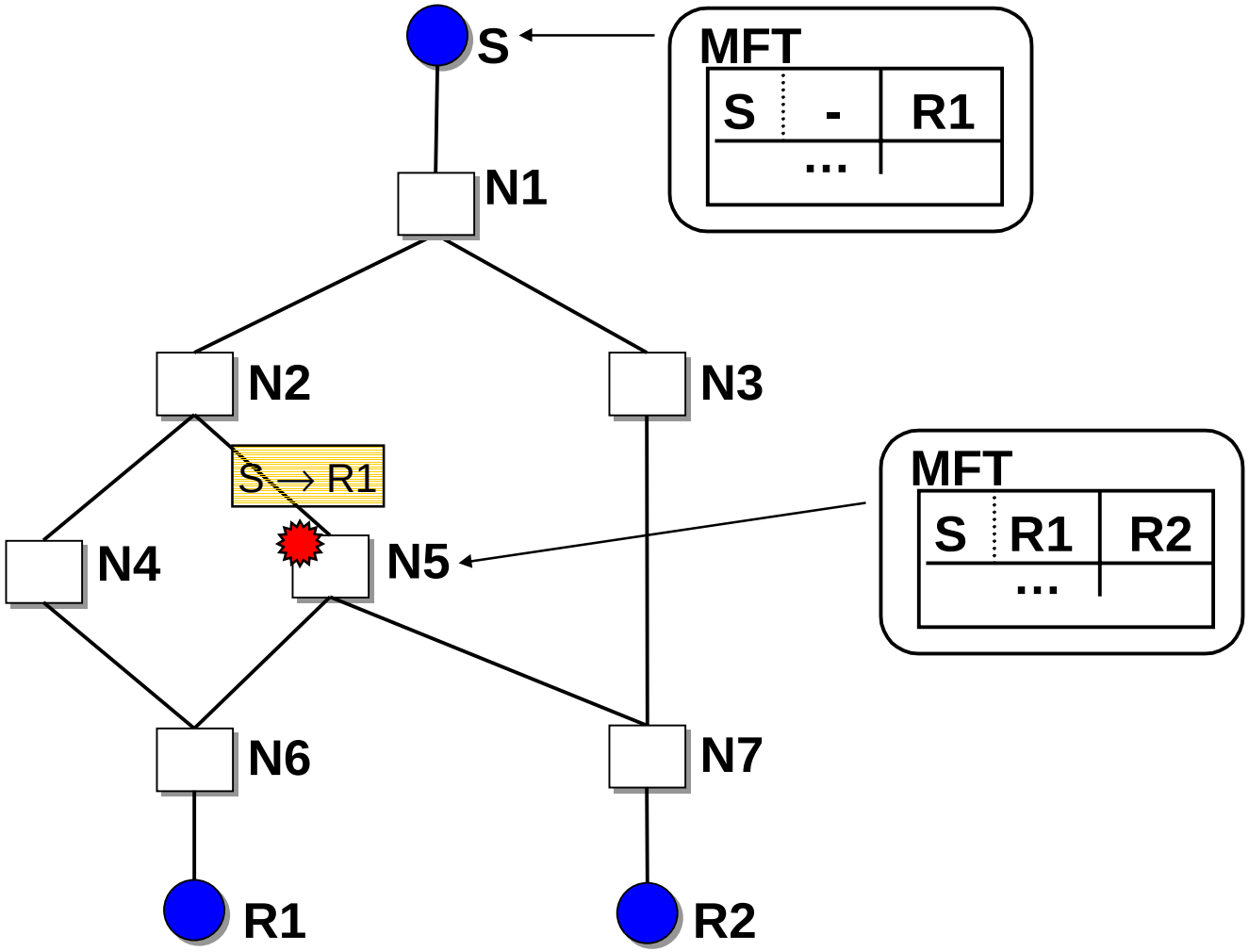
Using MFT in Packet Forwarding



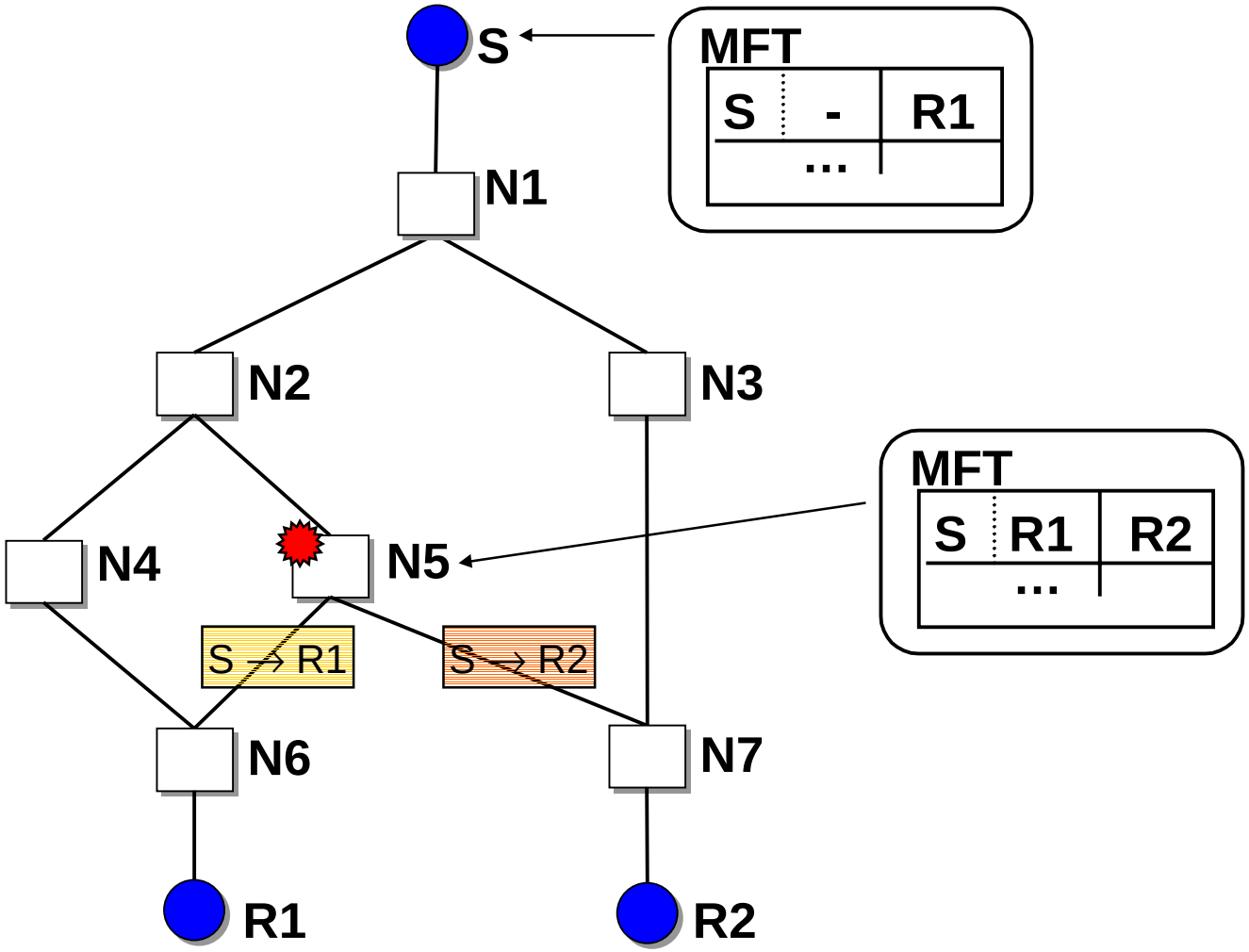
Using MFT in Packet Forwarding



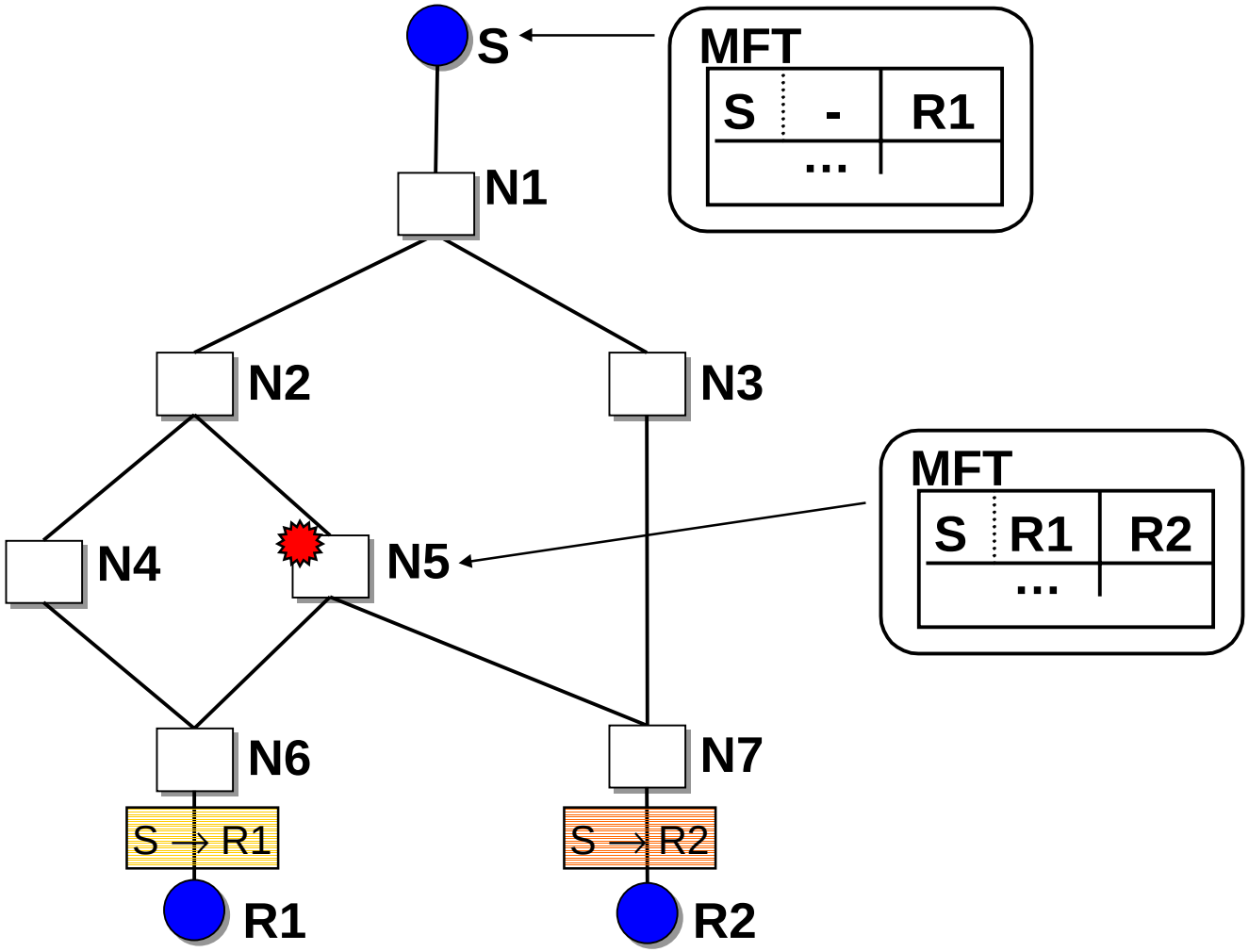
Using MFT in Packet Forwarding



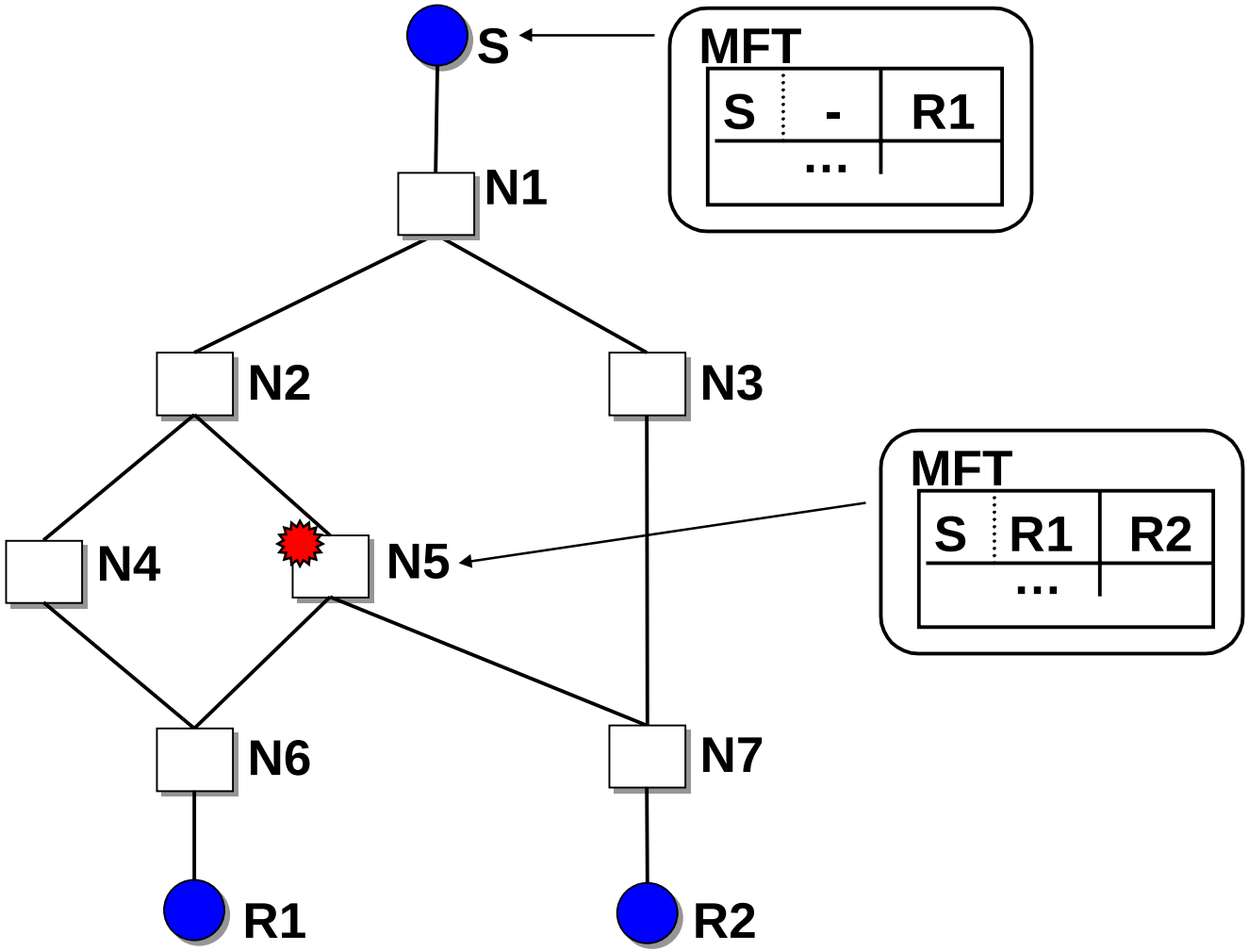
Using MFT in Packet Forwarding



Using MFT in Packet Forwarding

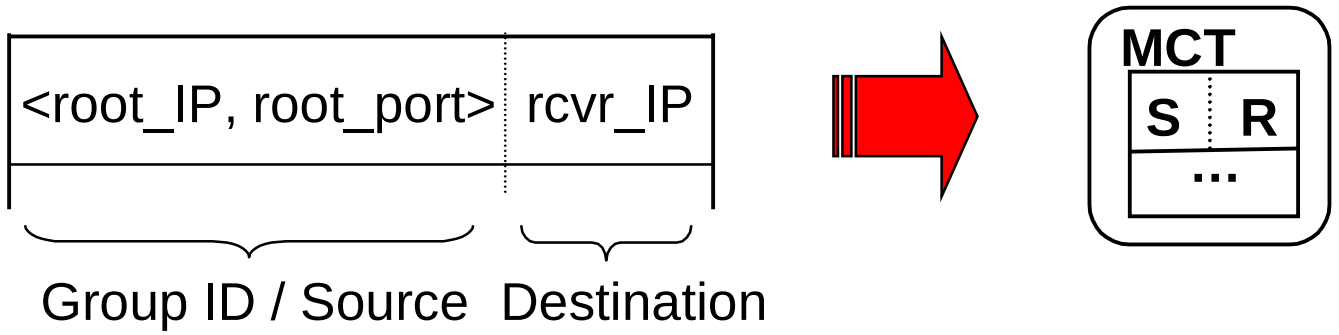


Using MFT in Packet Forwarding



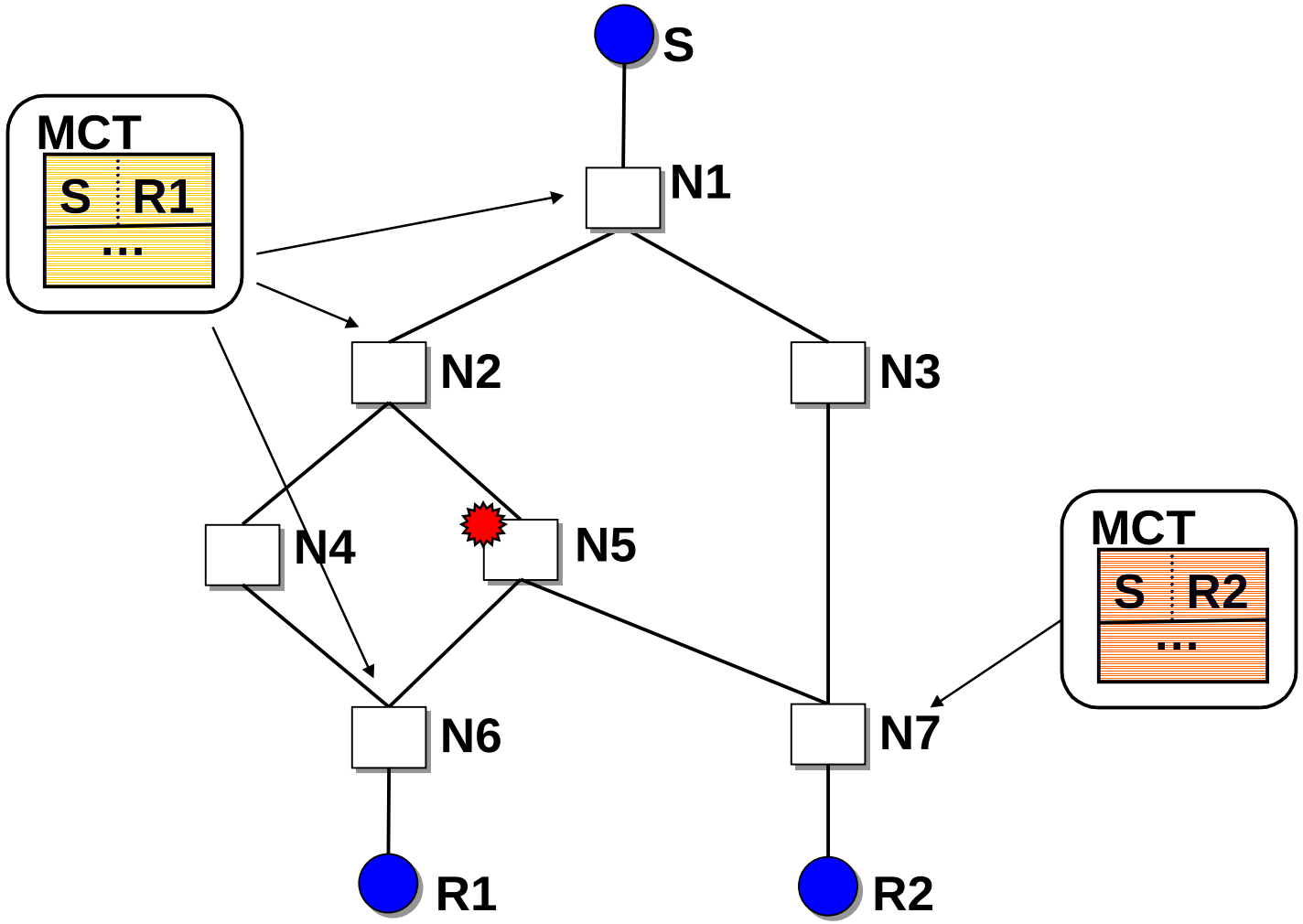
Control Path

Multicast Control Table (MCT)

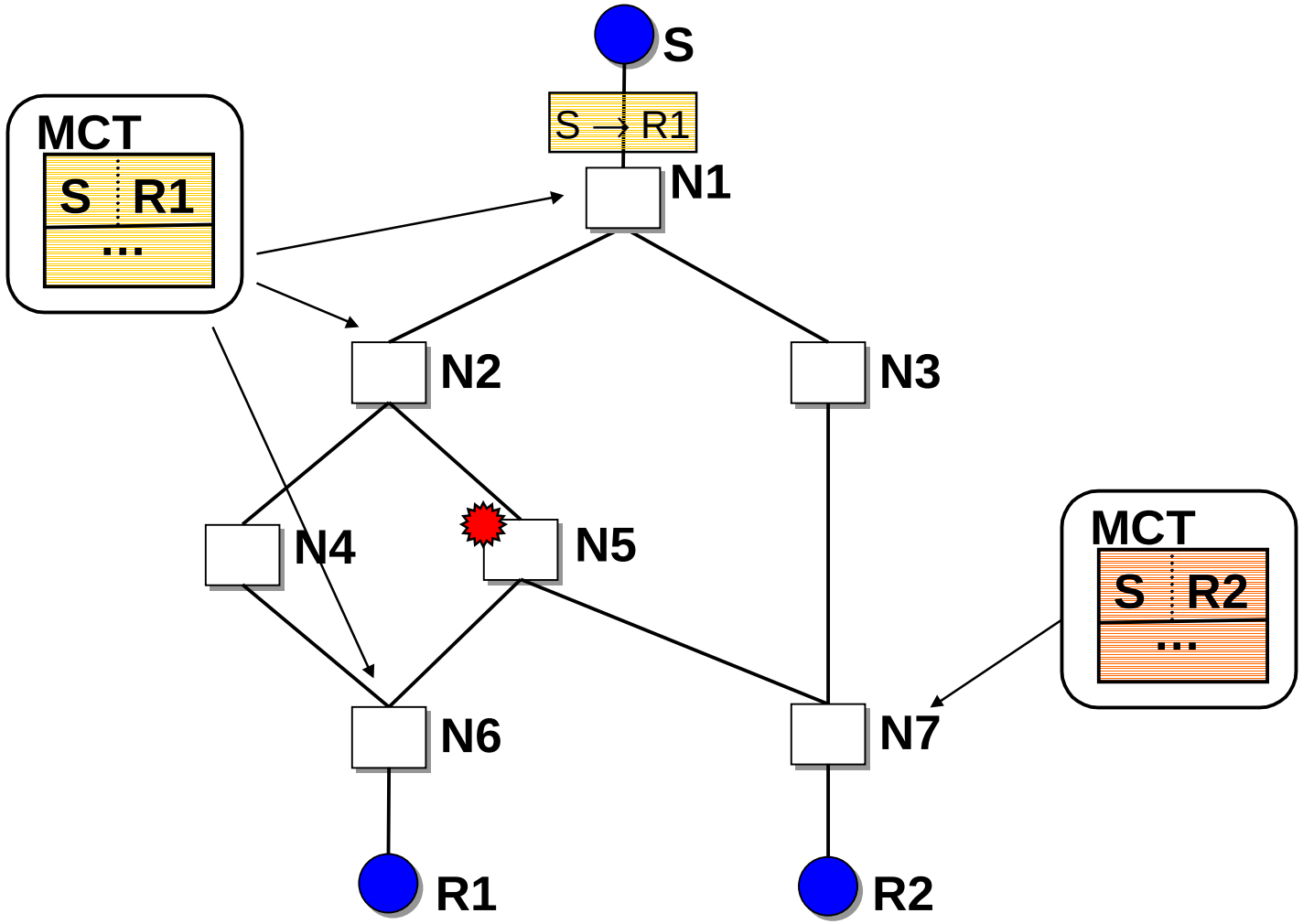


- A group has a MCT entry at node N if
  - N is on the multicast tree
  - N is non-branching

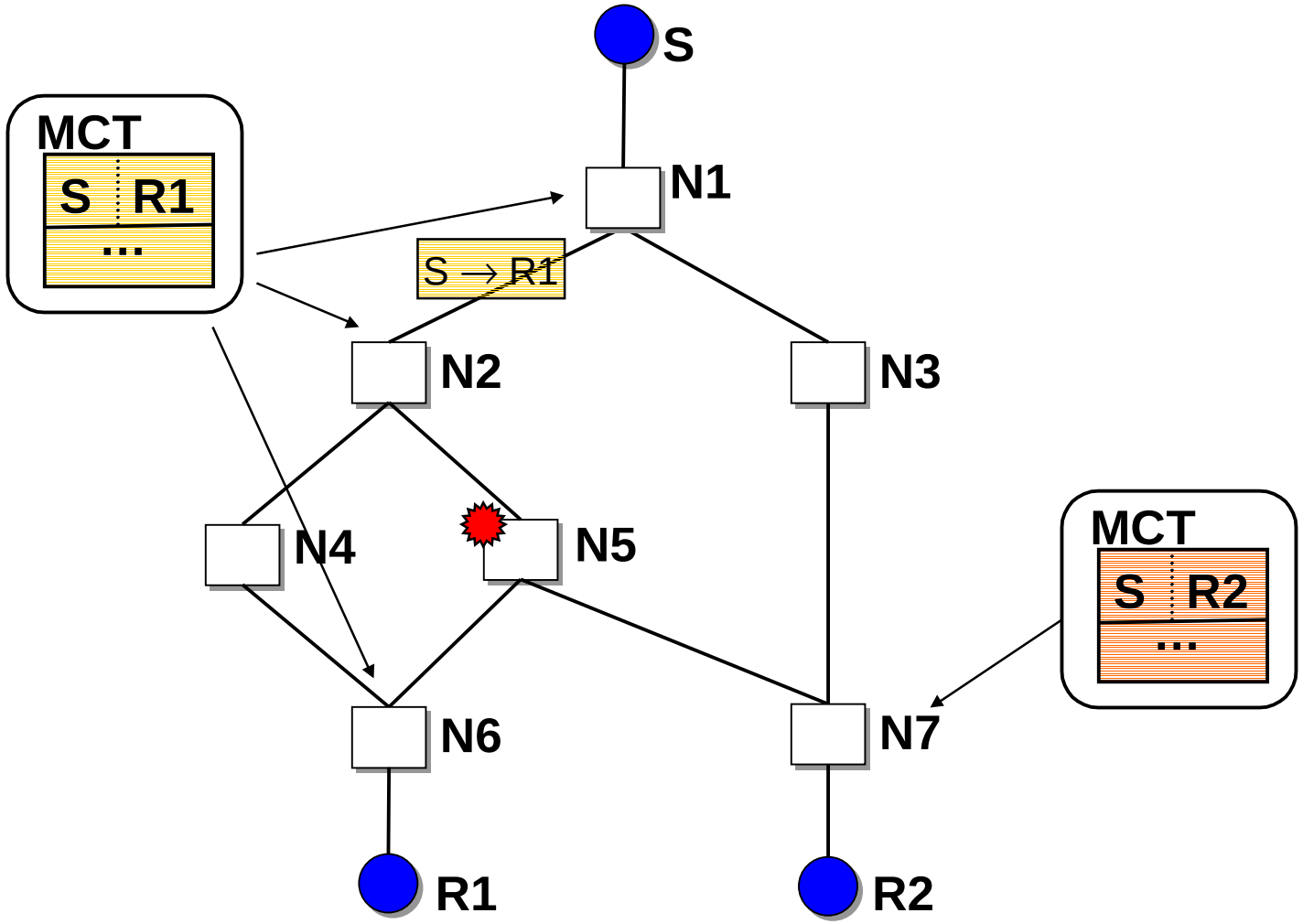
MCTs at Non-Branching Nodes



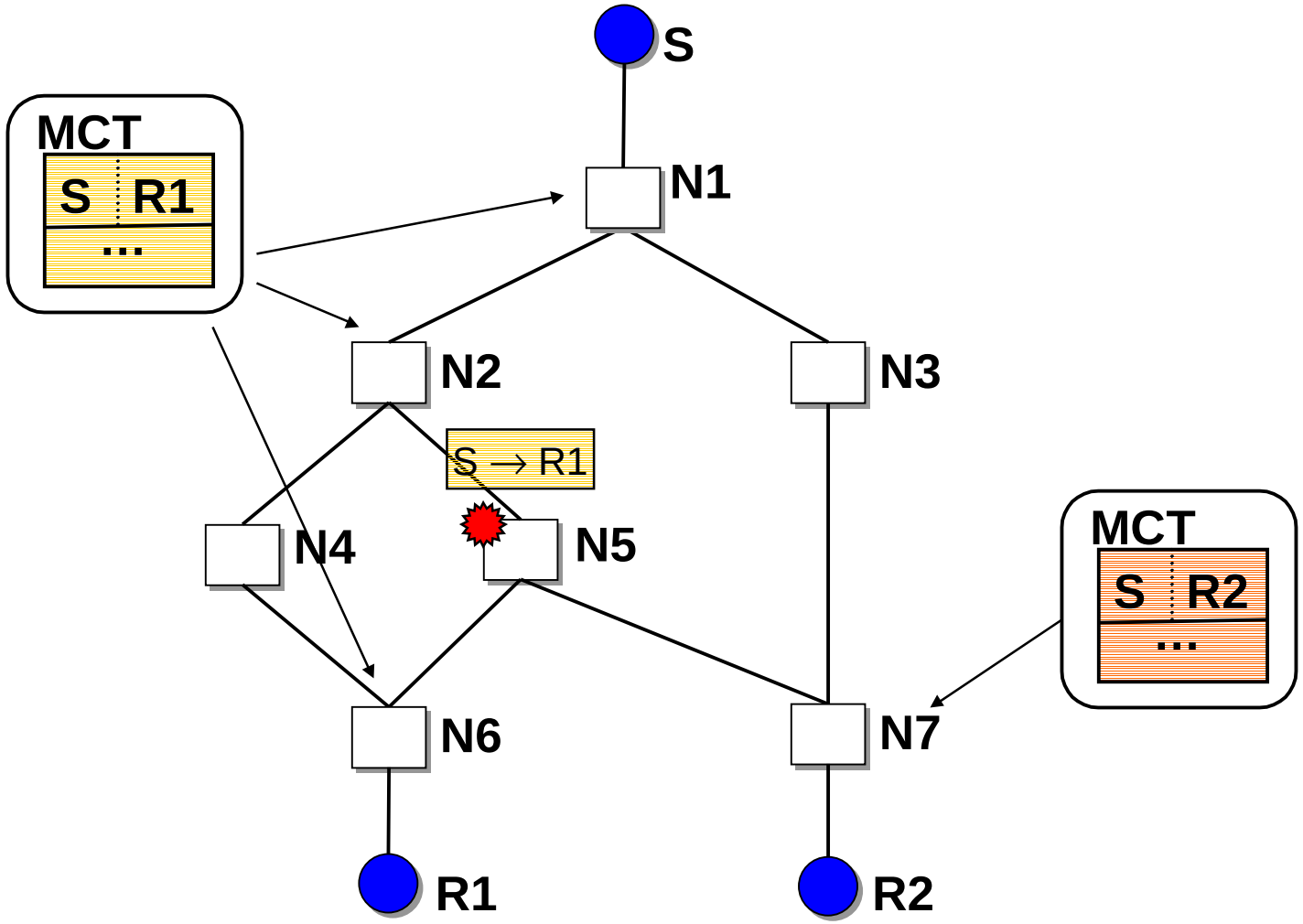
MCTs at Non-Branching Nodes



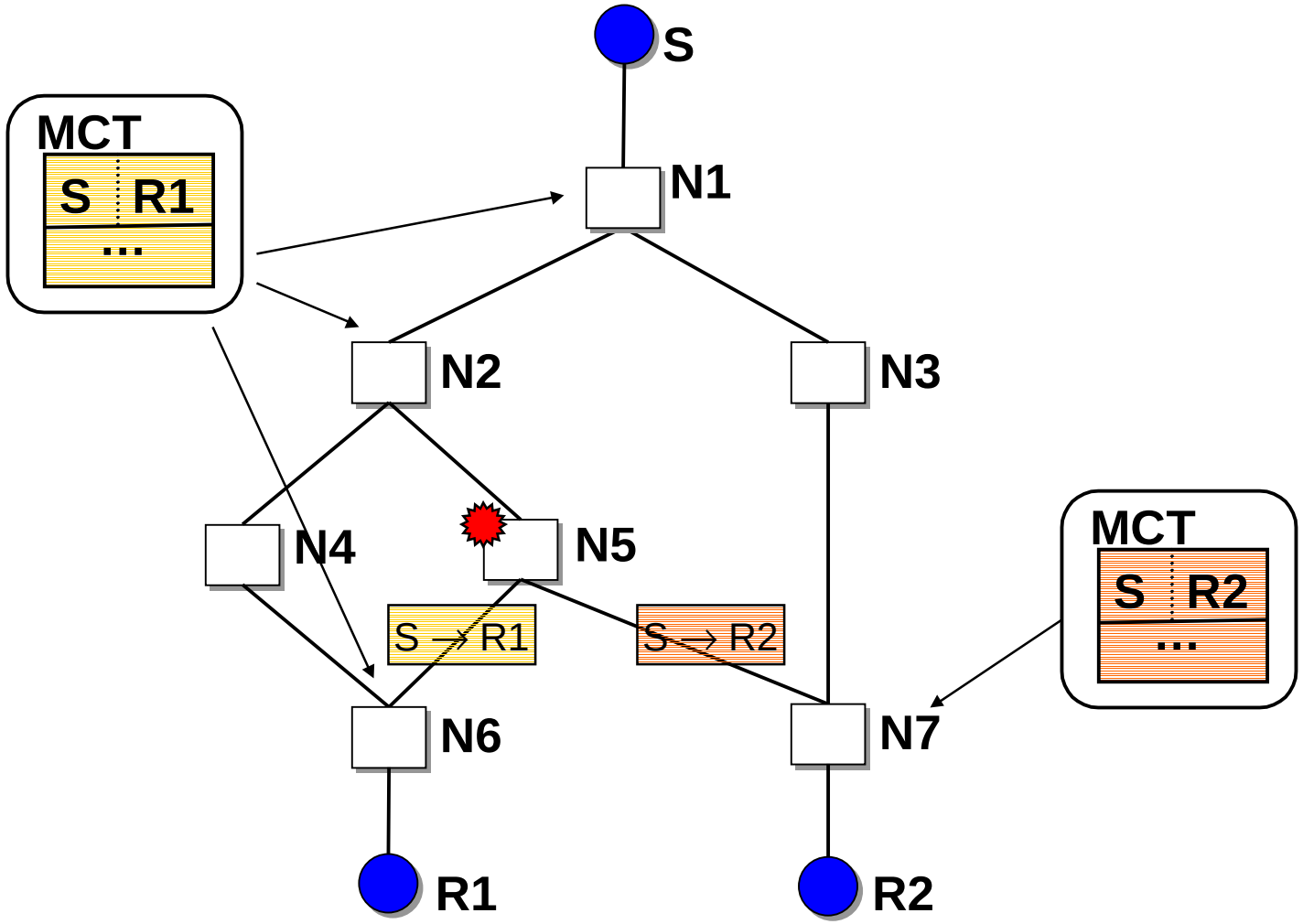
MCTs at Non-Branching Nodes



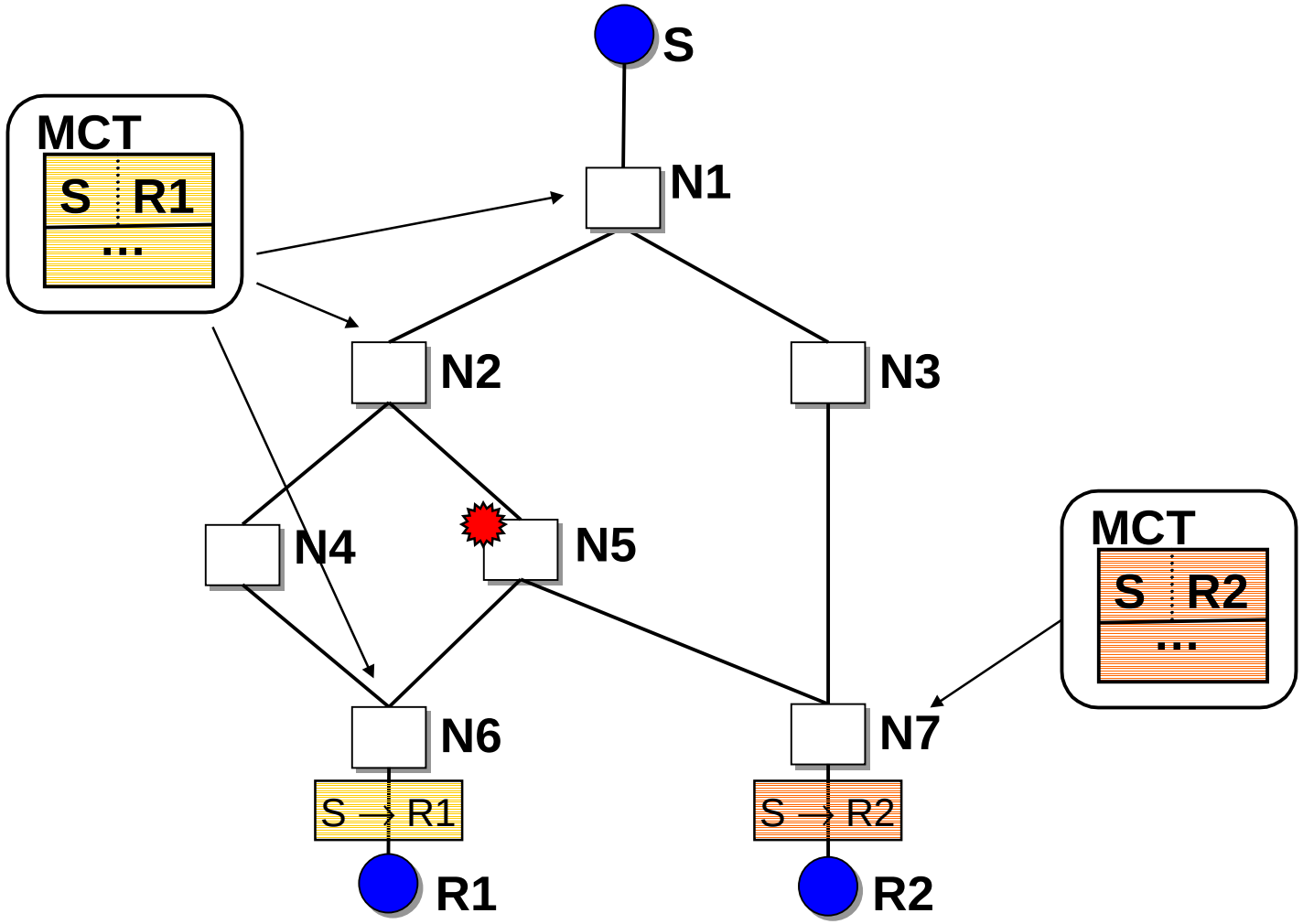
MCTs at Non-Branching Nodes



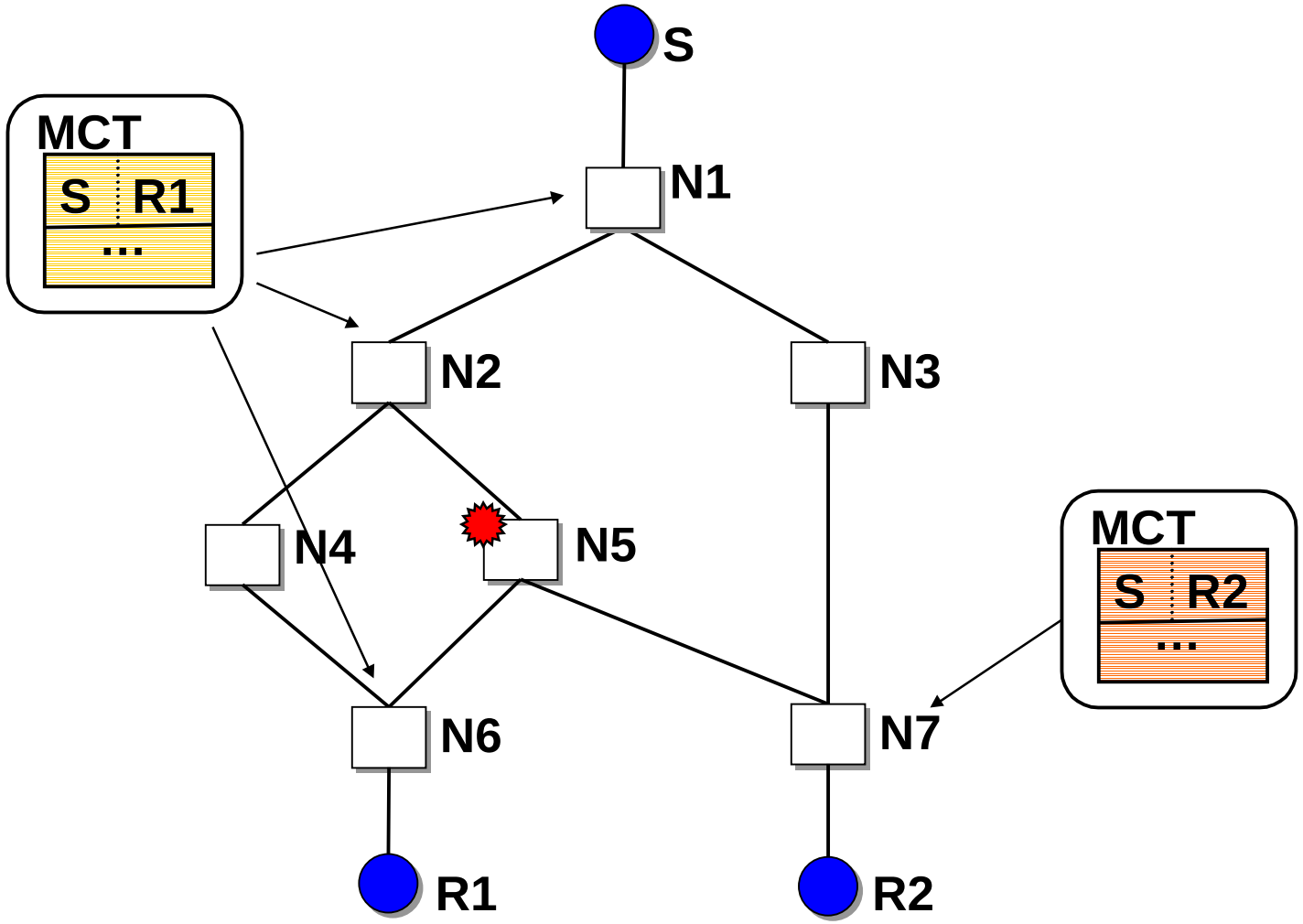
MCTs at Non-Branching Nodes



MCTs at Non-Branching Nodes



MCTs at Non-Branching Nodes



Purpose of MCT

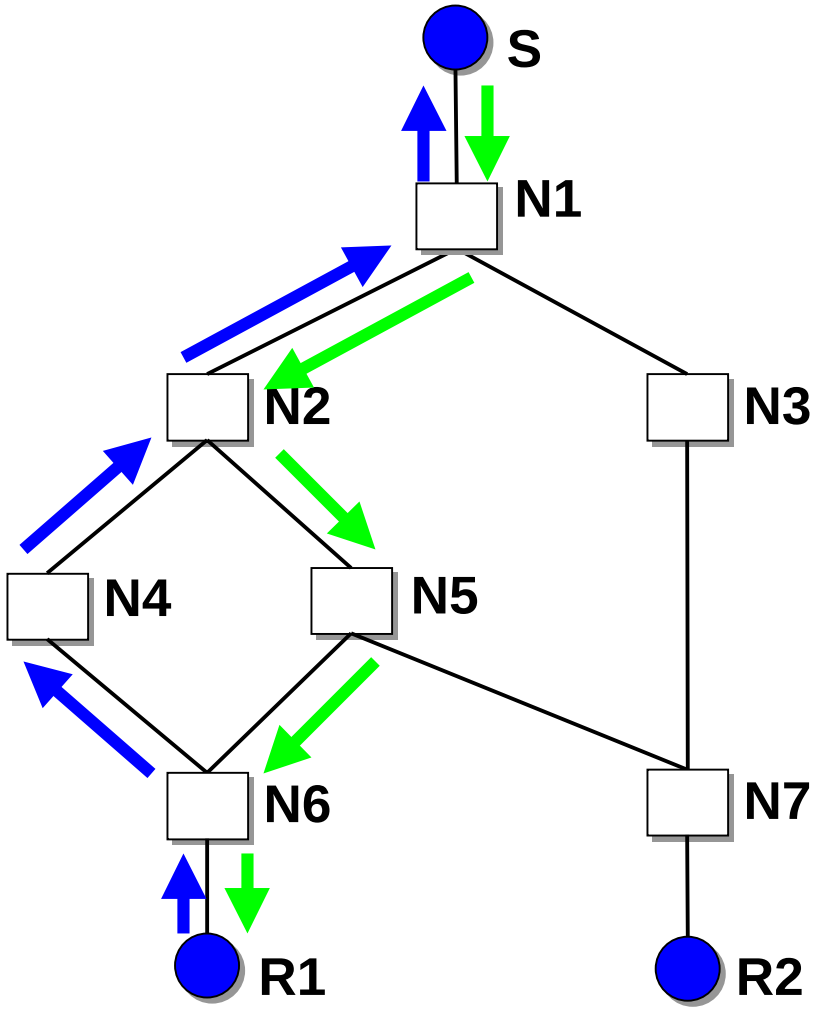
- To mark non-branching nodes on the group tree
- To facilitate group joins
- MCT state is **control path state**

- REUNITE II
  - eliminates the need for MCTs
  - more message types
  - more complex protocol
  - discussed in our technical report

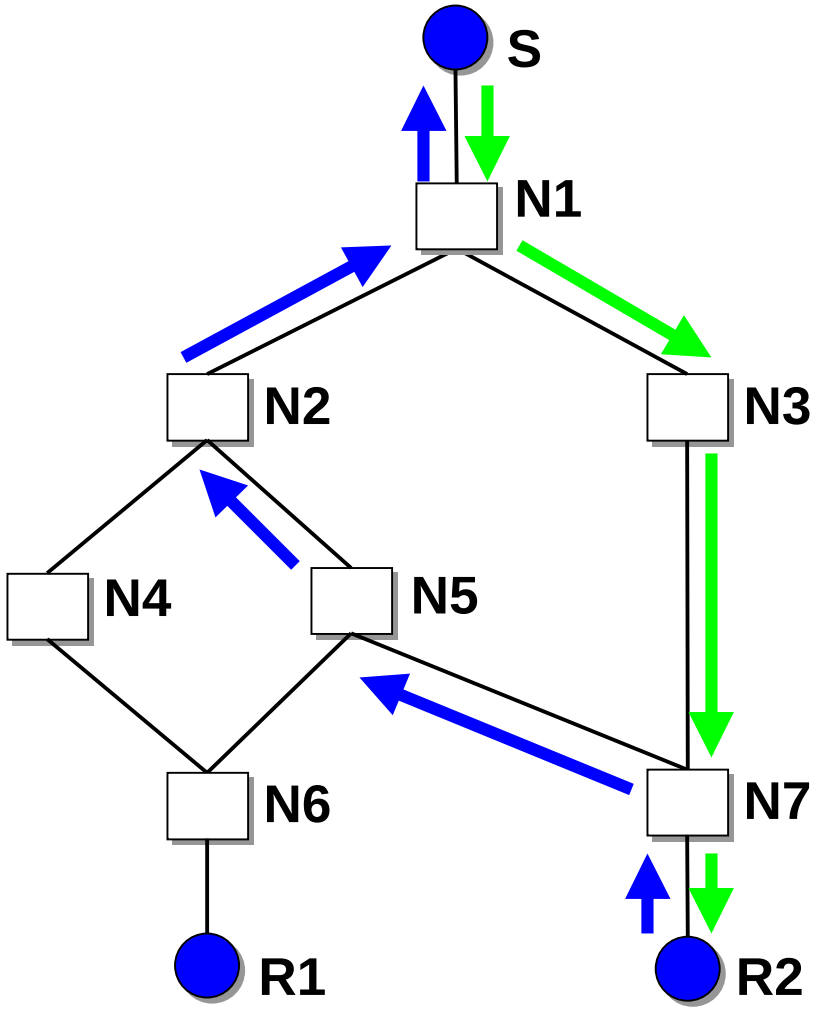
Tree Maintenance Protocol Messages

- **JOIN**: periodically sent by receivers to the root to create/refresh MFT state
- **TREE**: periodically sent by root to receivers to create/refresh MCT and MFT state

Topology Note

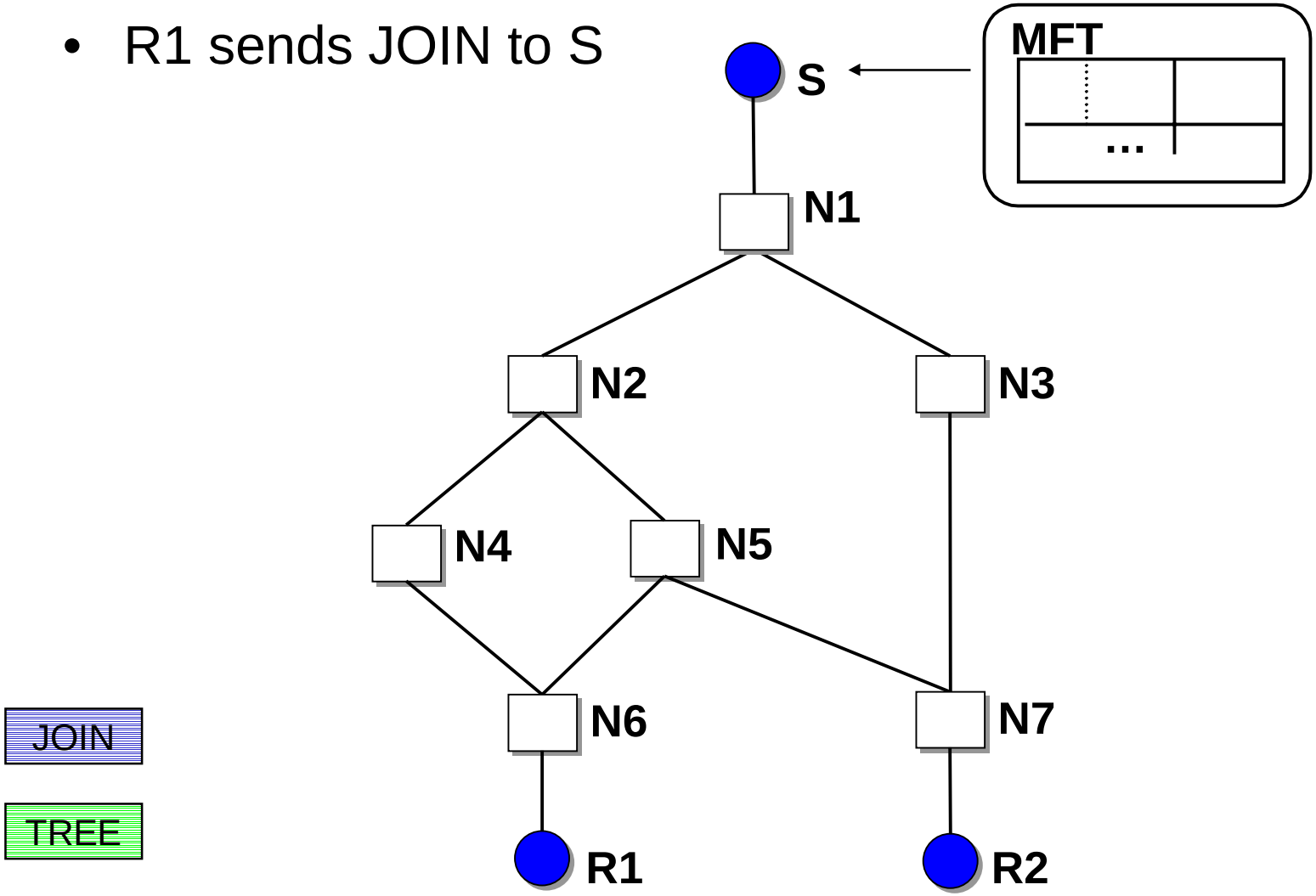


Topology Note



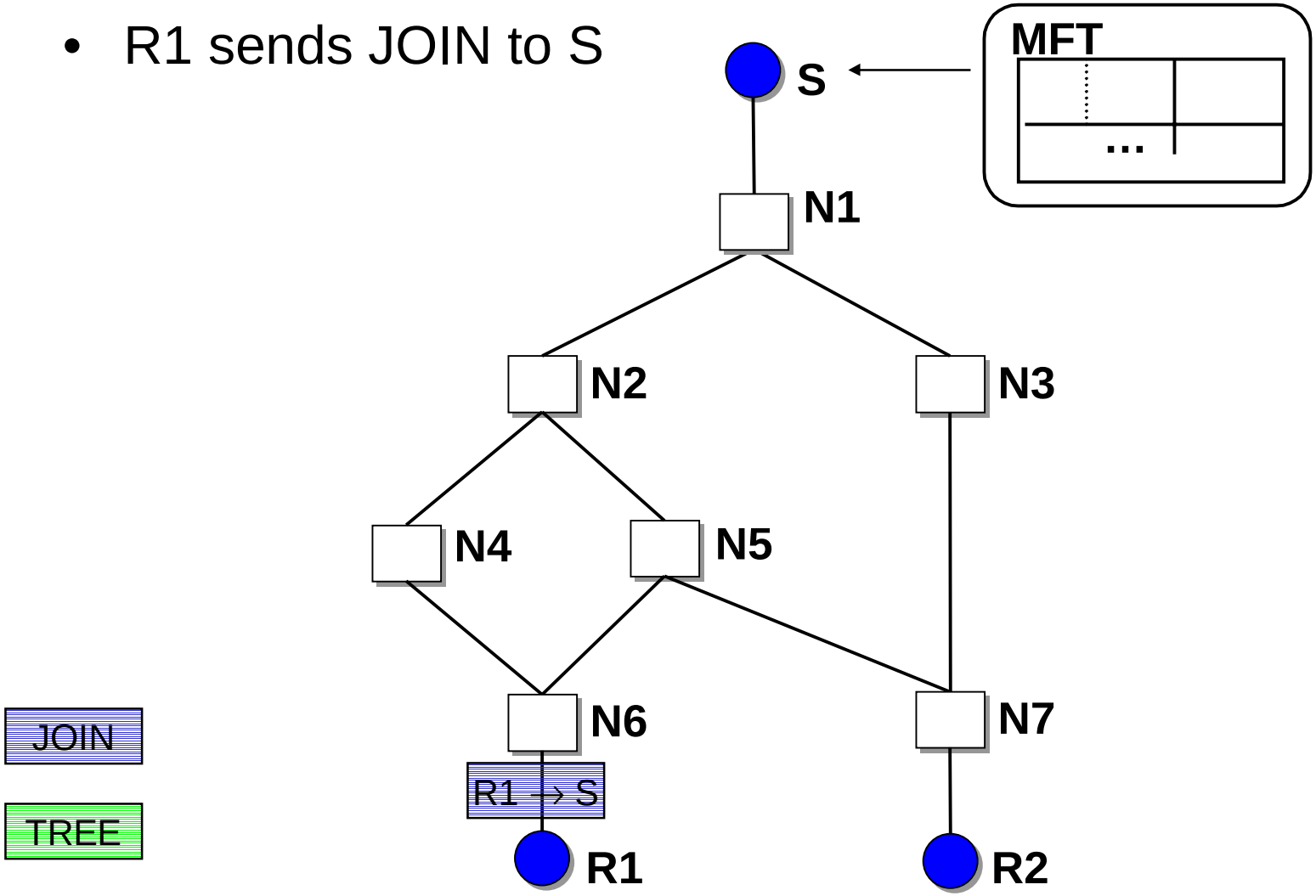
Joining a Group

- R1 sends JOIN to S



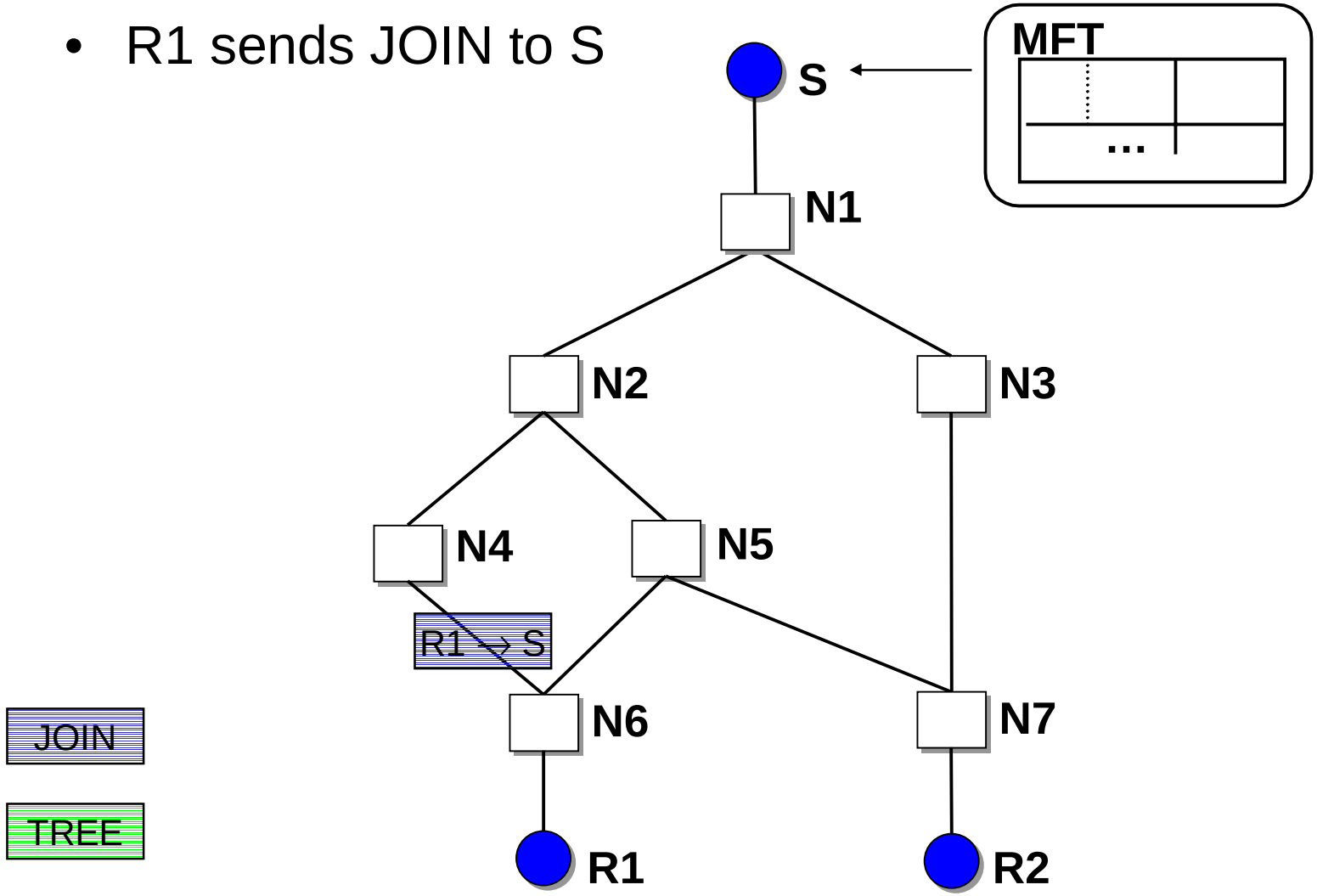
Joining a Group

- R1 sends JOIN to S



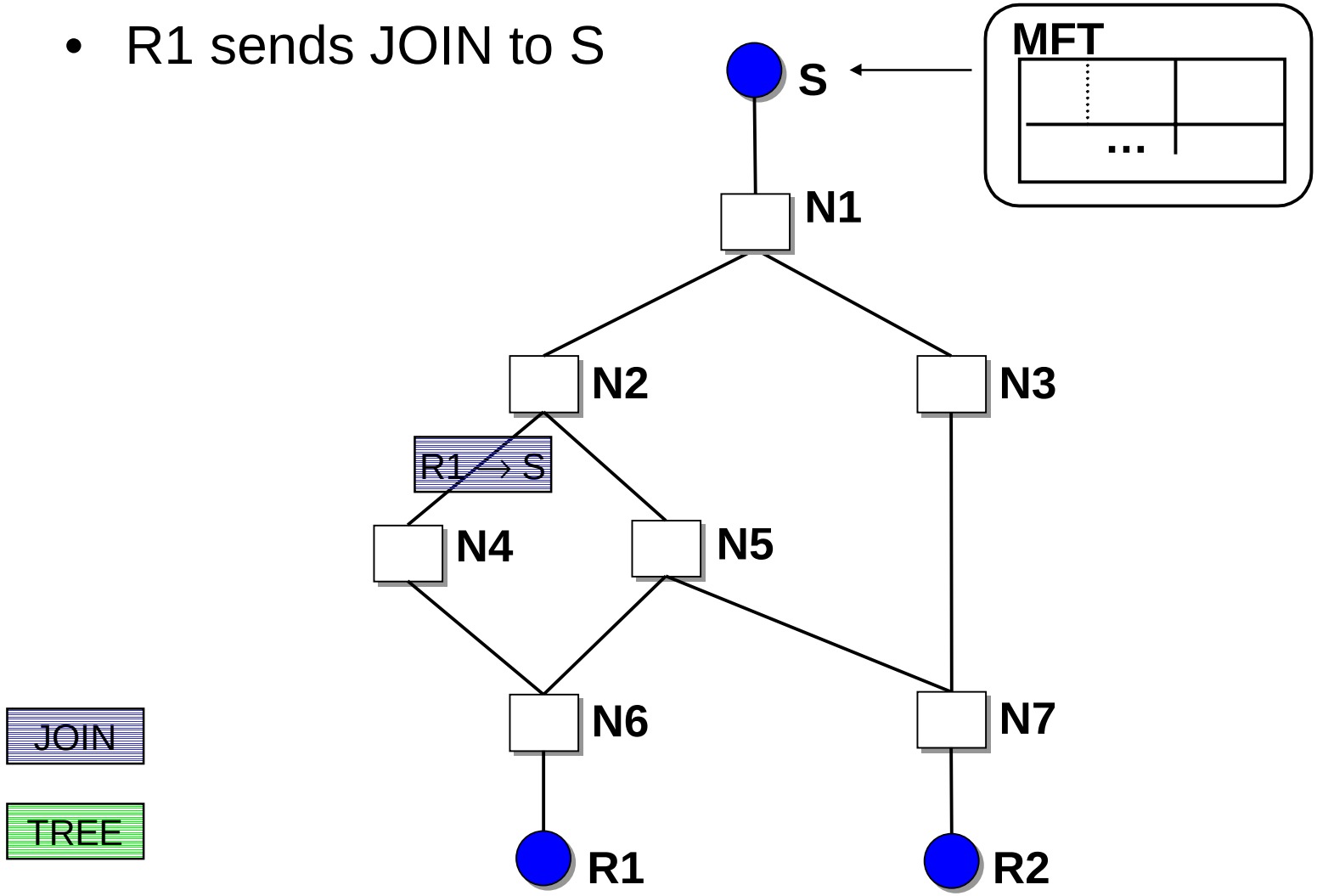
Joining a Group

- R1 sends JOIN to S



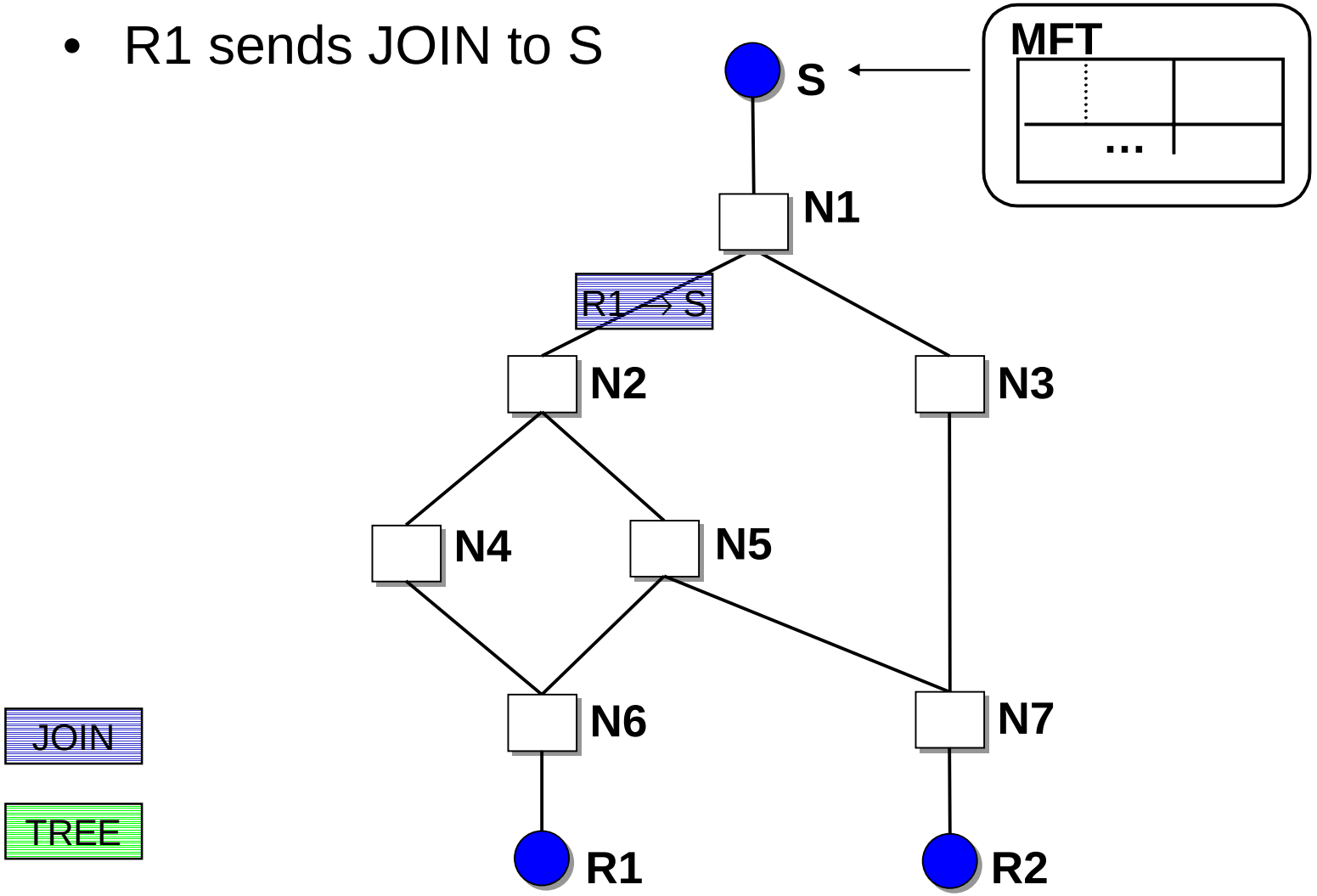
Joining a Group

- R1 sends JOIN to S



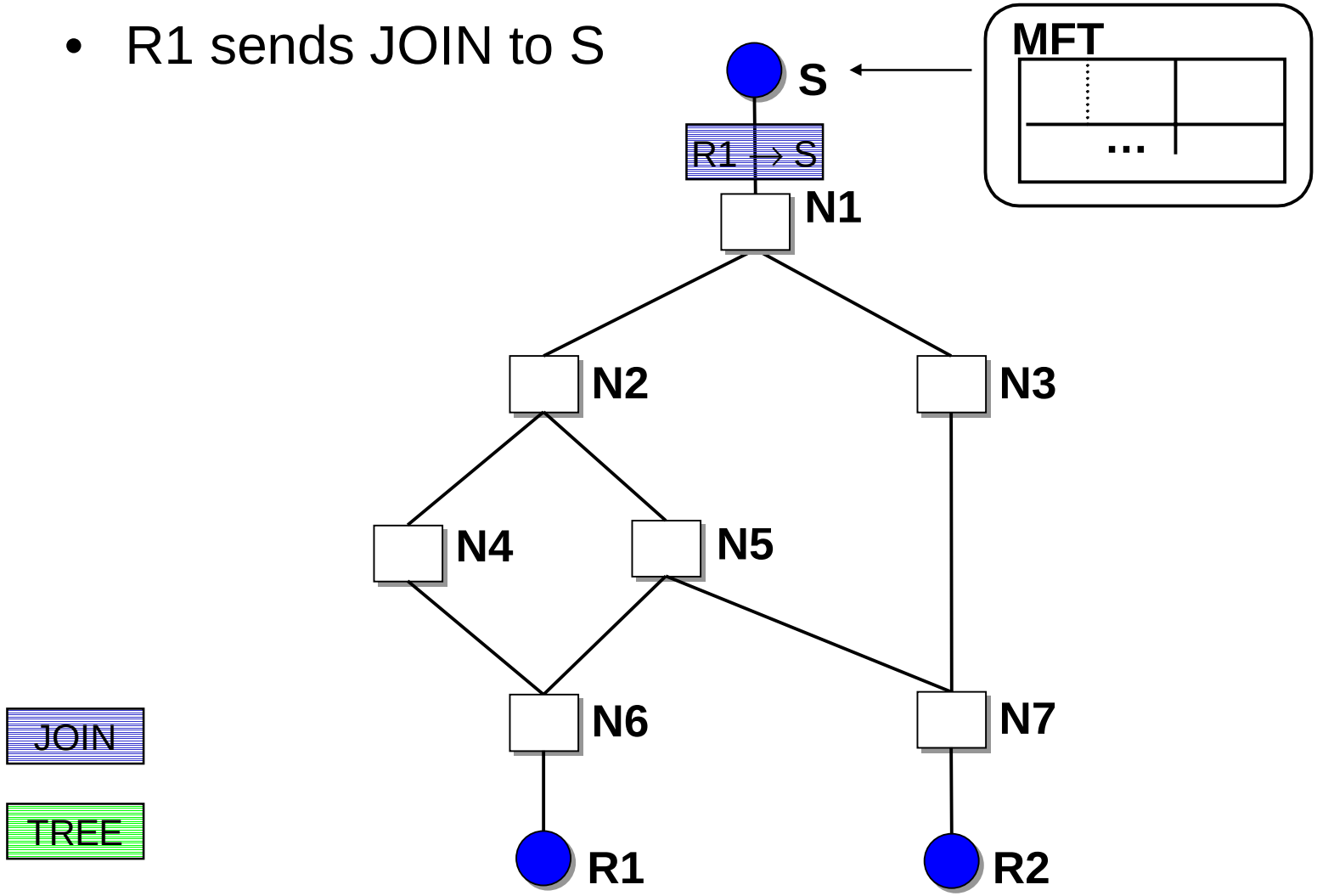
Joining a Group

- R1 sends JOIN to S



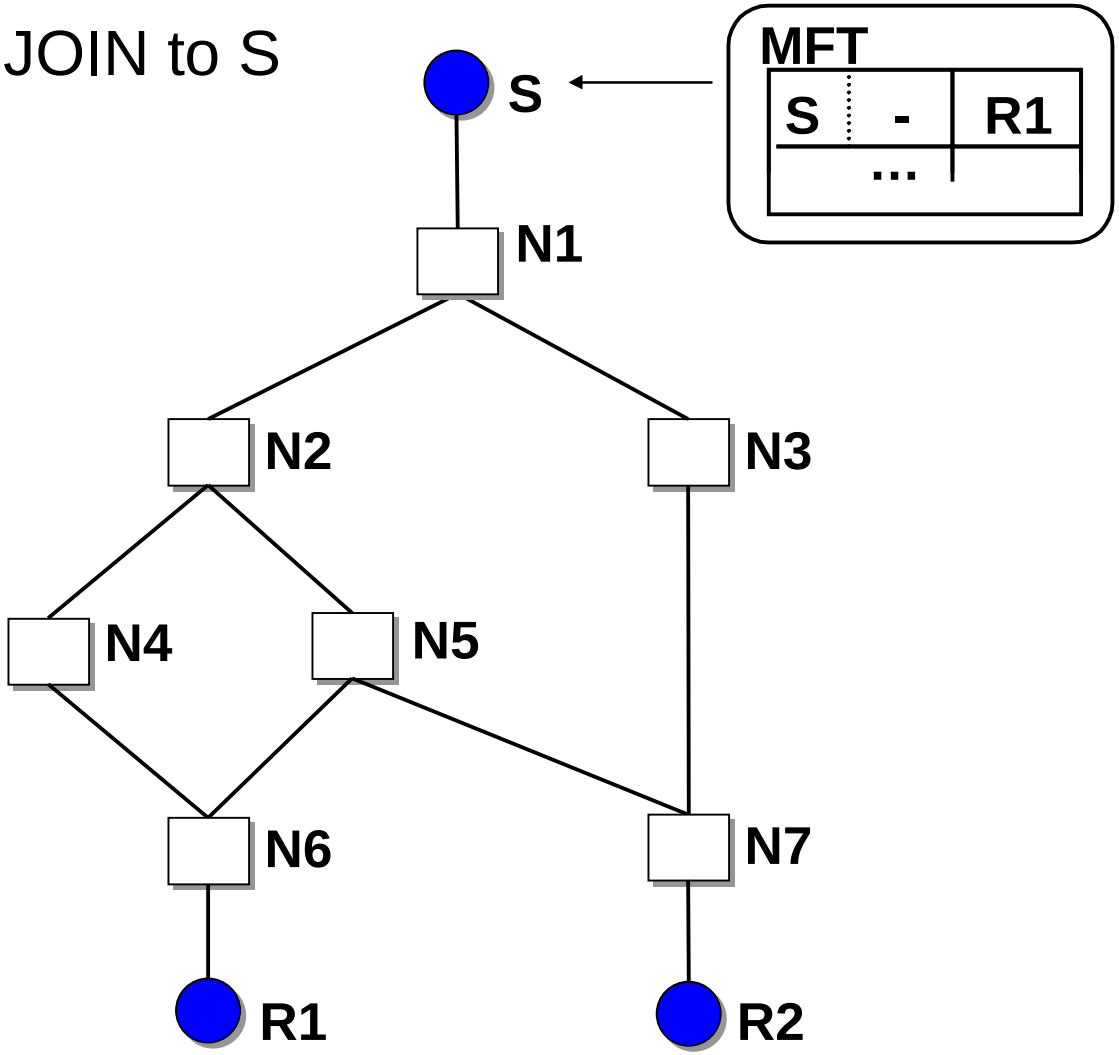
Joining a Group

- R1 sends JOIN to S



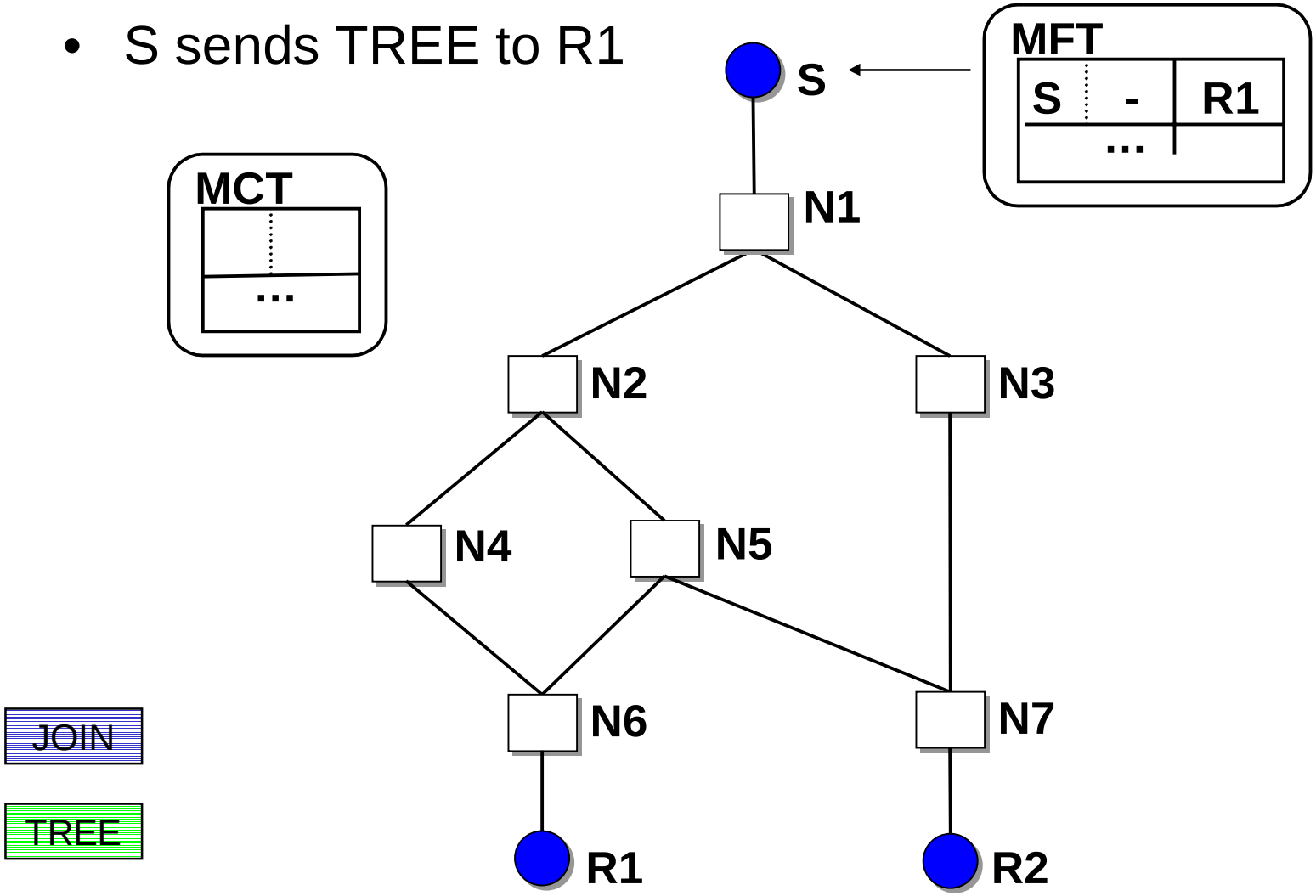
Joining a Group

- R1 sends JOIN to S



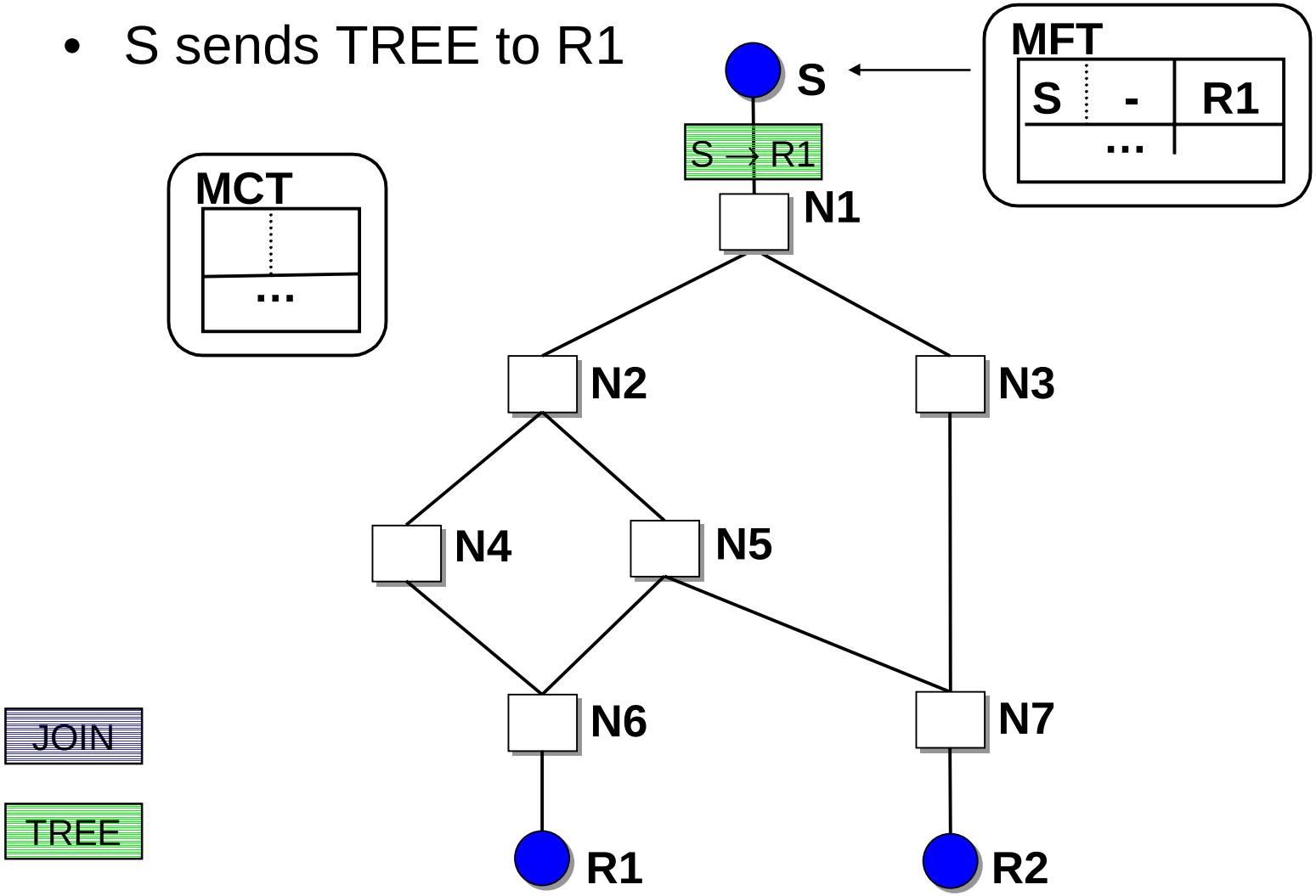
Joining a Group

- S sends TREE to R1



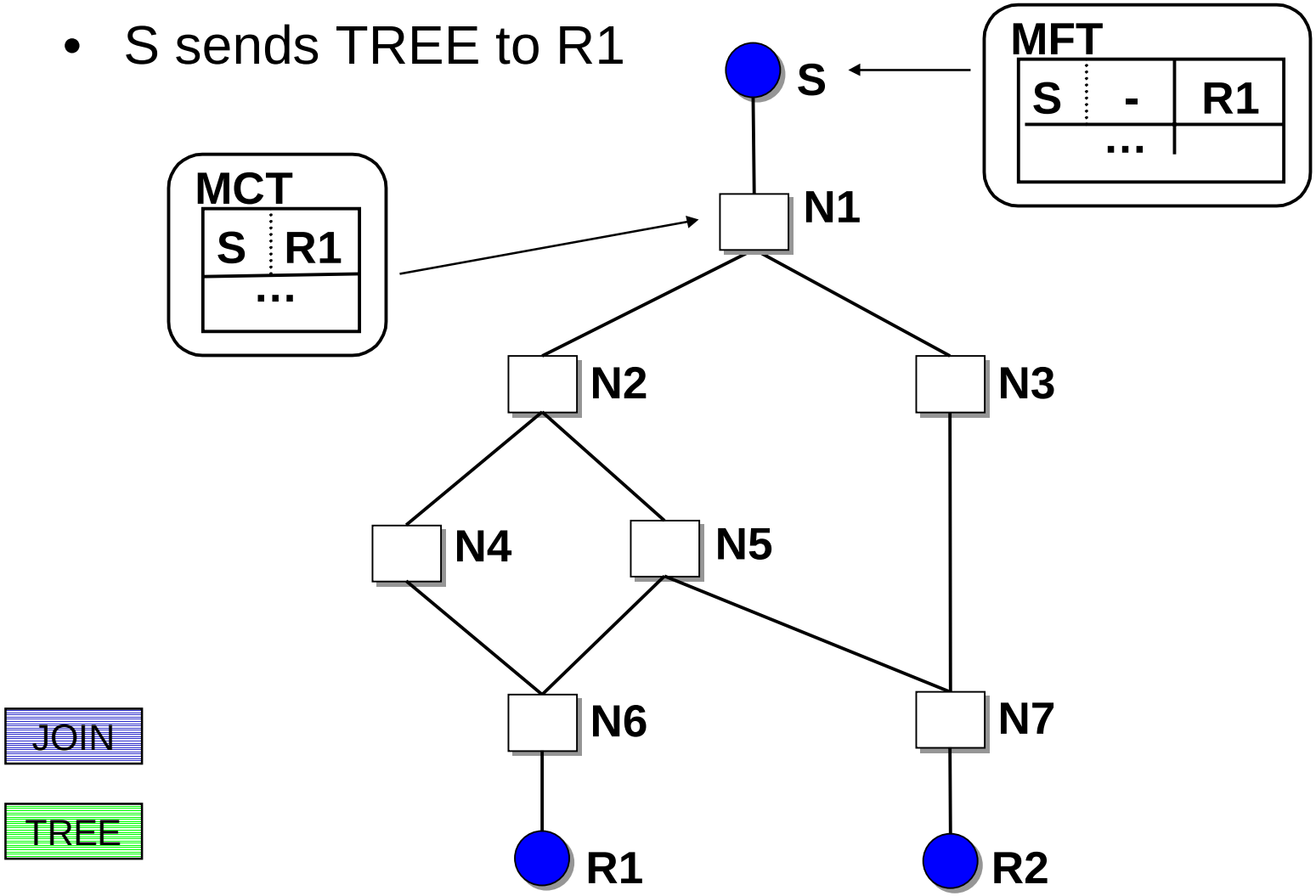
Joining a Group

- S sends TREE to R1



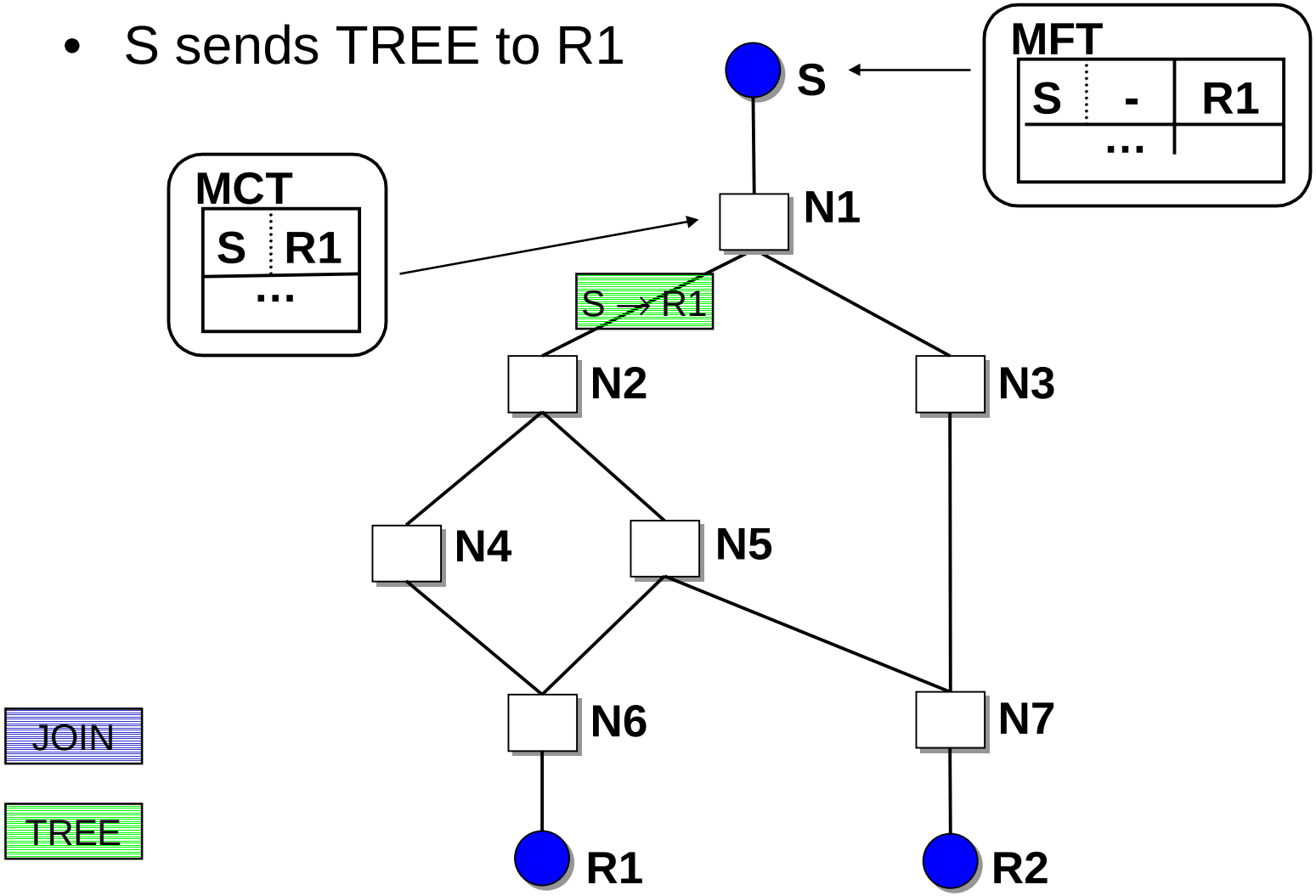
Joining a Group

- S sends TREE to R1



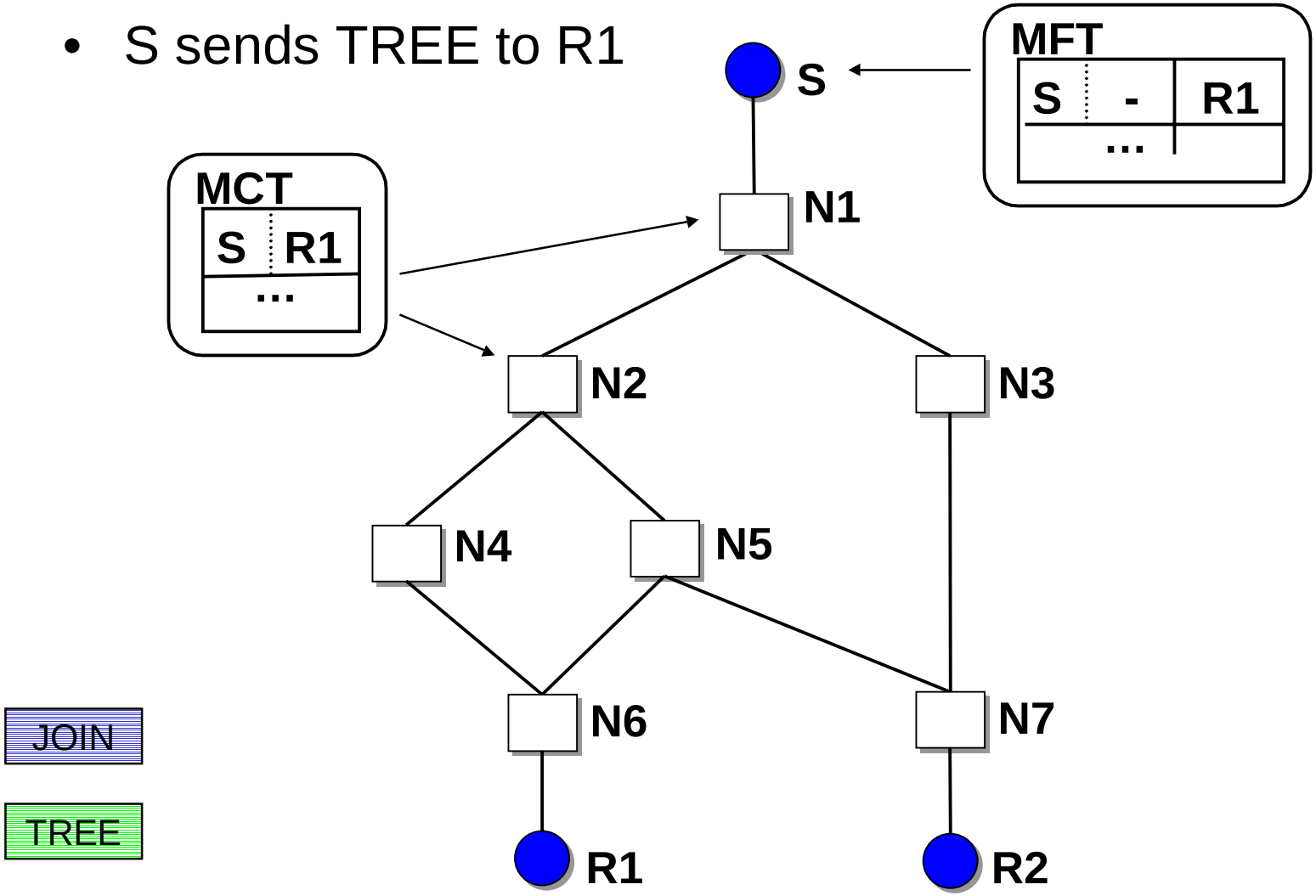
Joining a Group

- S sends TREE to R1



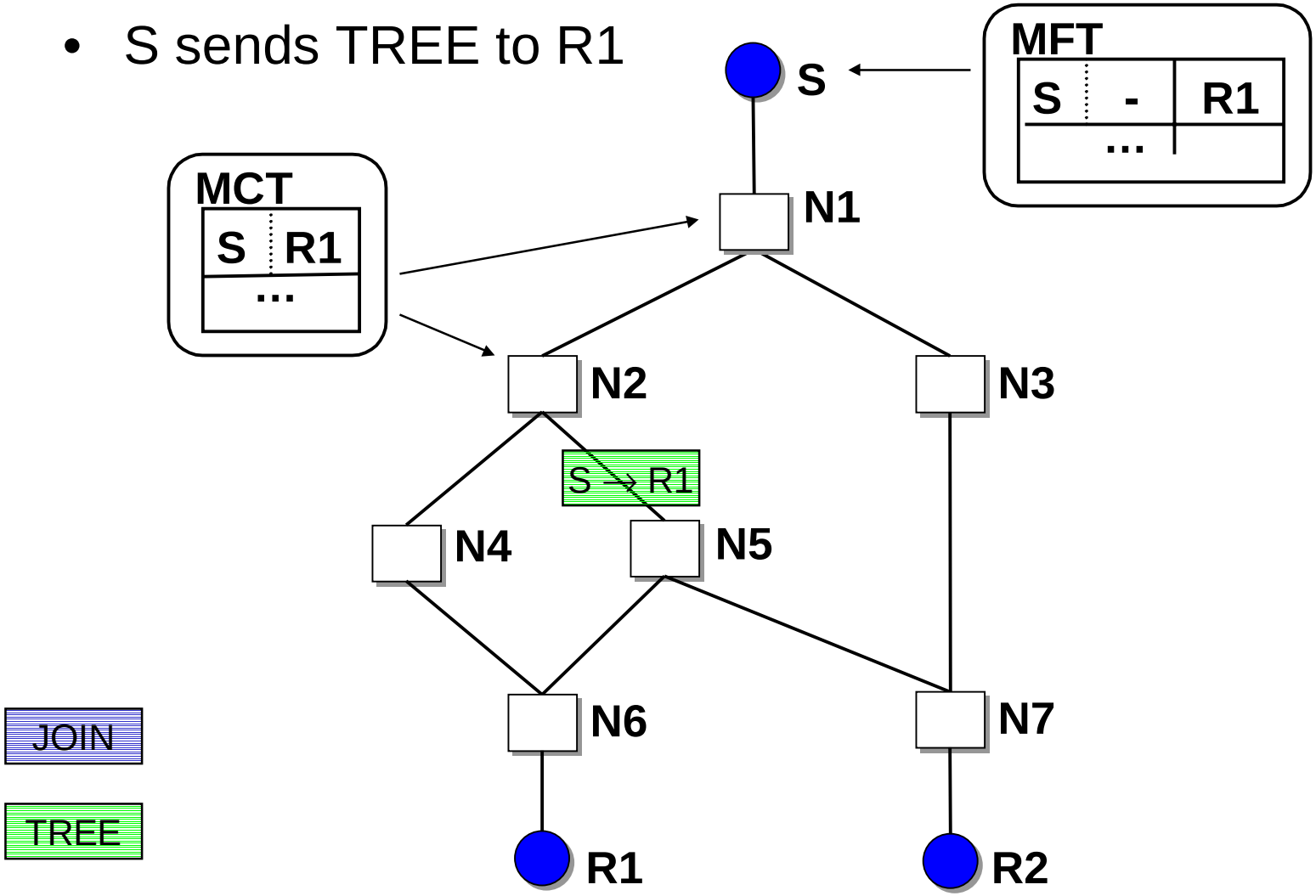
Joining a Group

- S sends TREE to R1



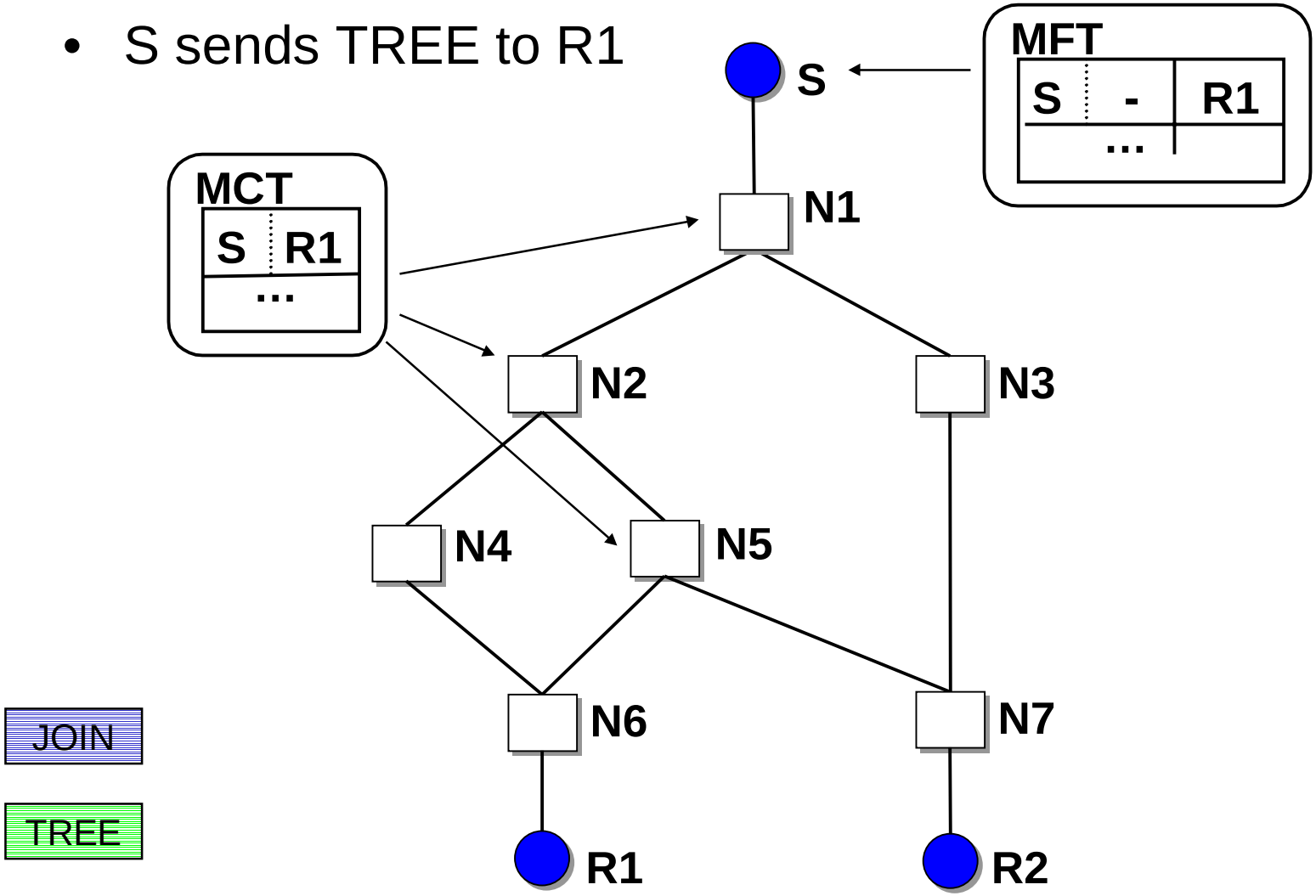
Joining a Group

- S sends TREE to R1



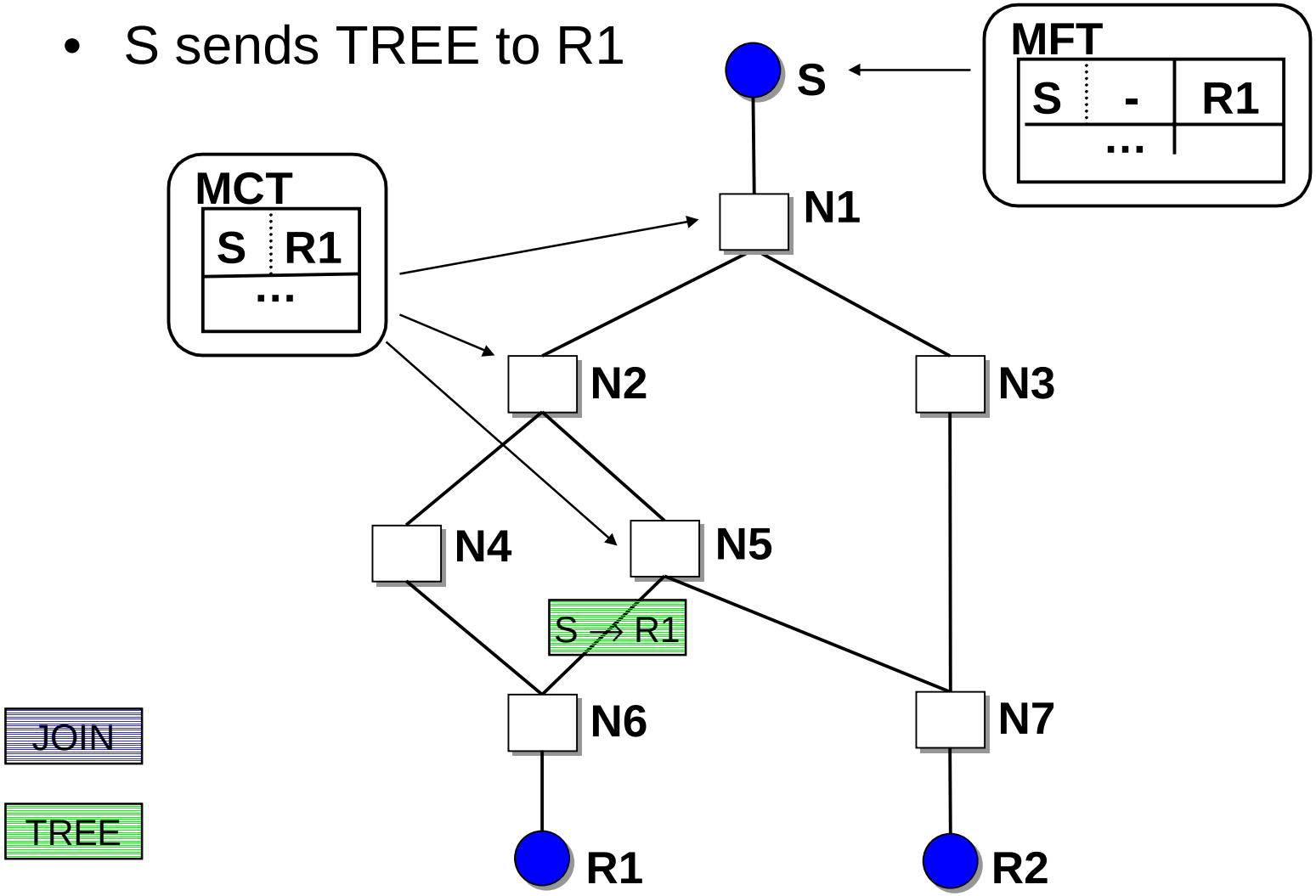
Joining a Group

- S sends TREE to R1



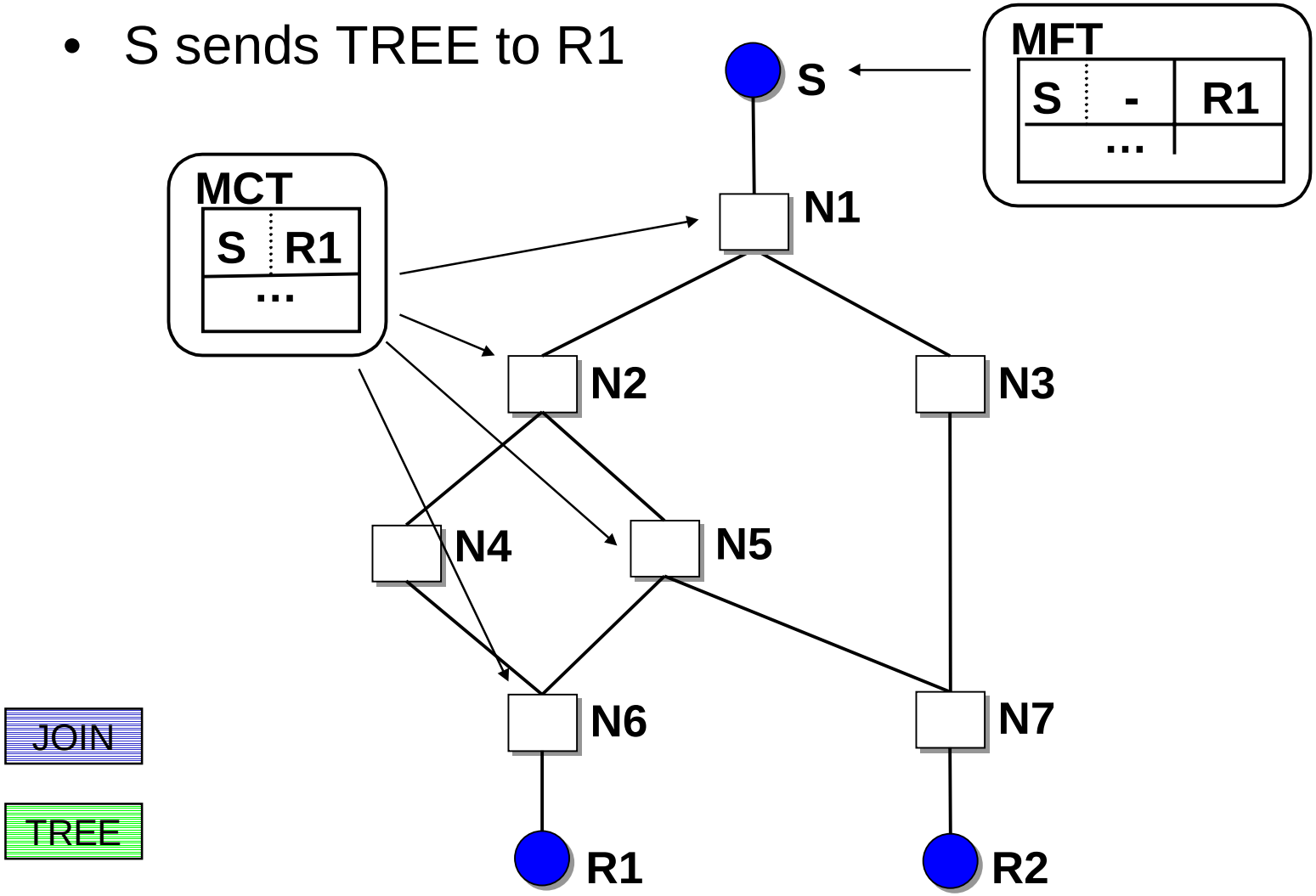
Joining a Group

- S sends TREE to R1



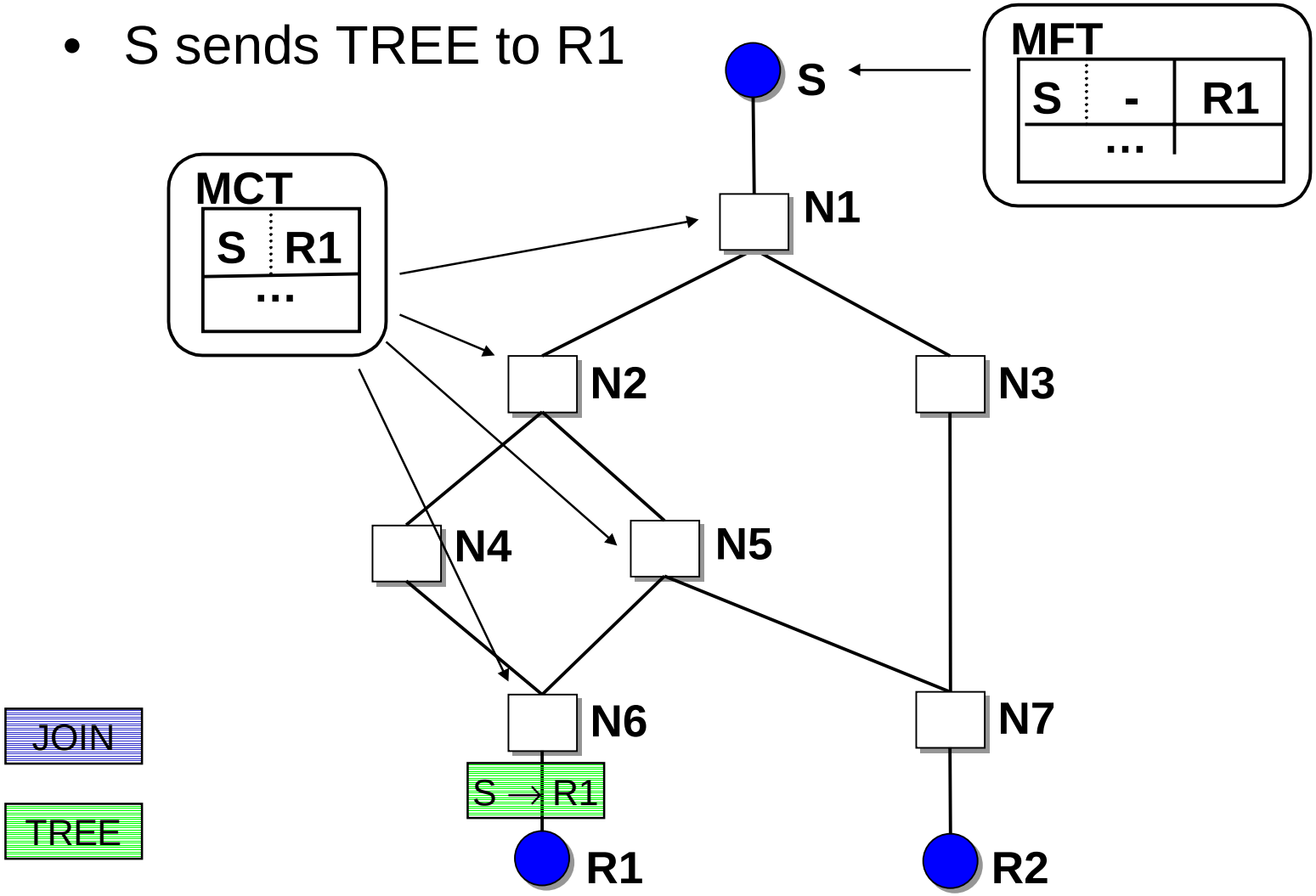
Joining a Group

- S sends TREE to R1



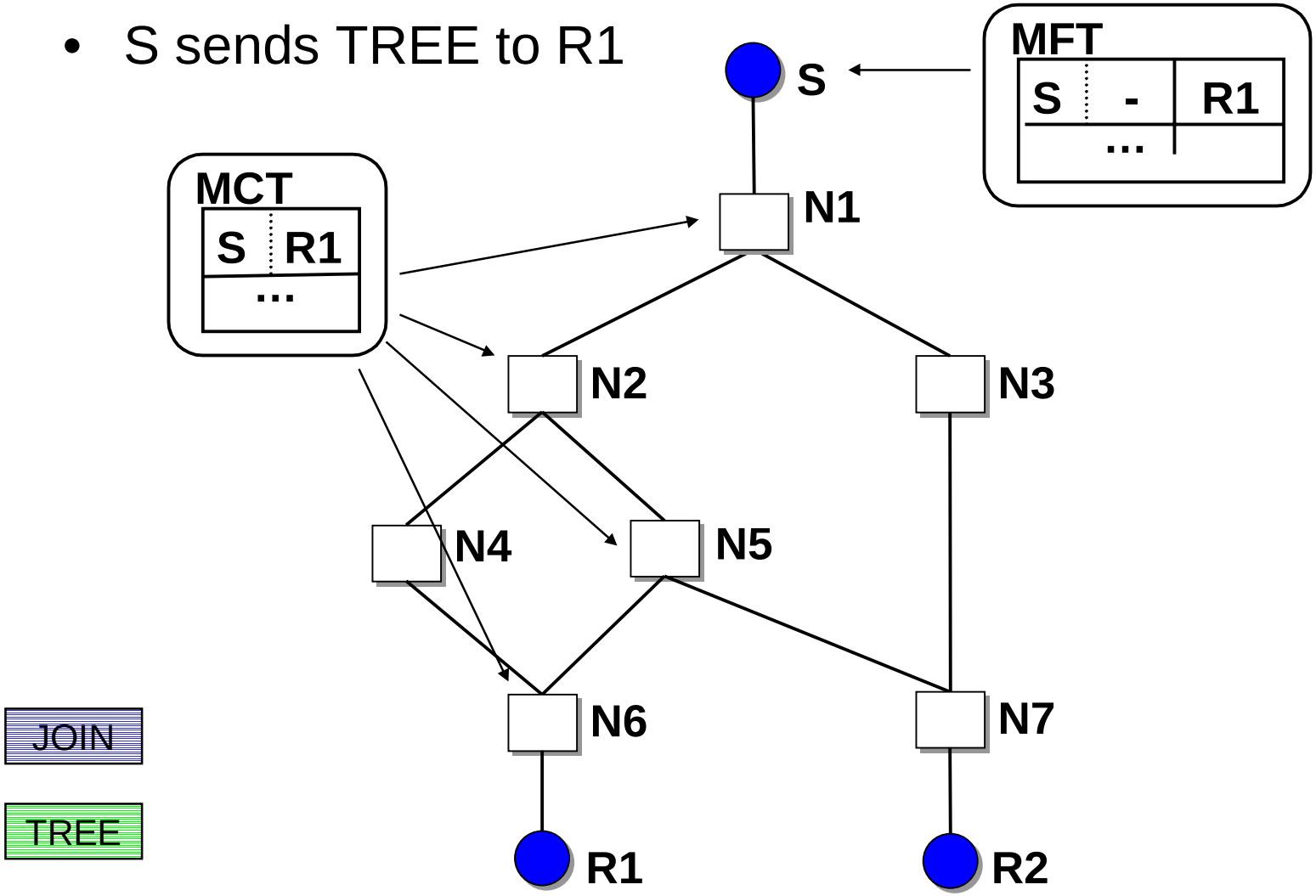
Joining a Group

- S sends TREE to R1



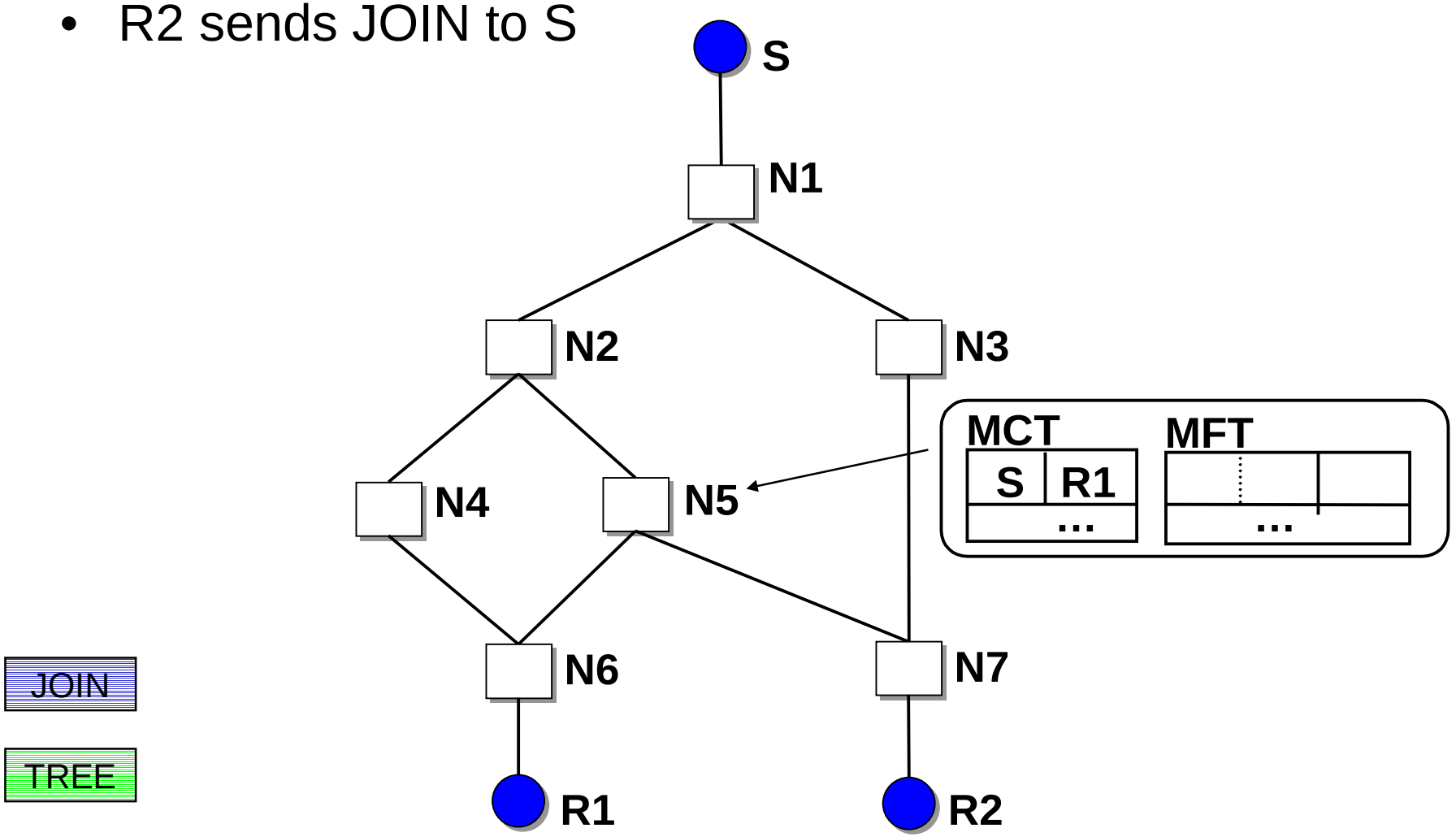
Joining a Group

- S sends TREE to R1



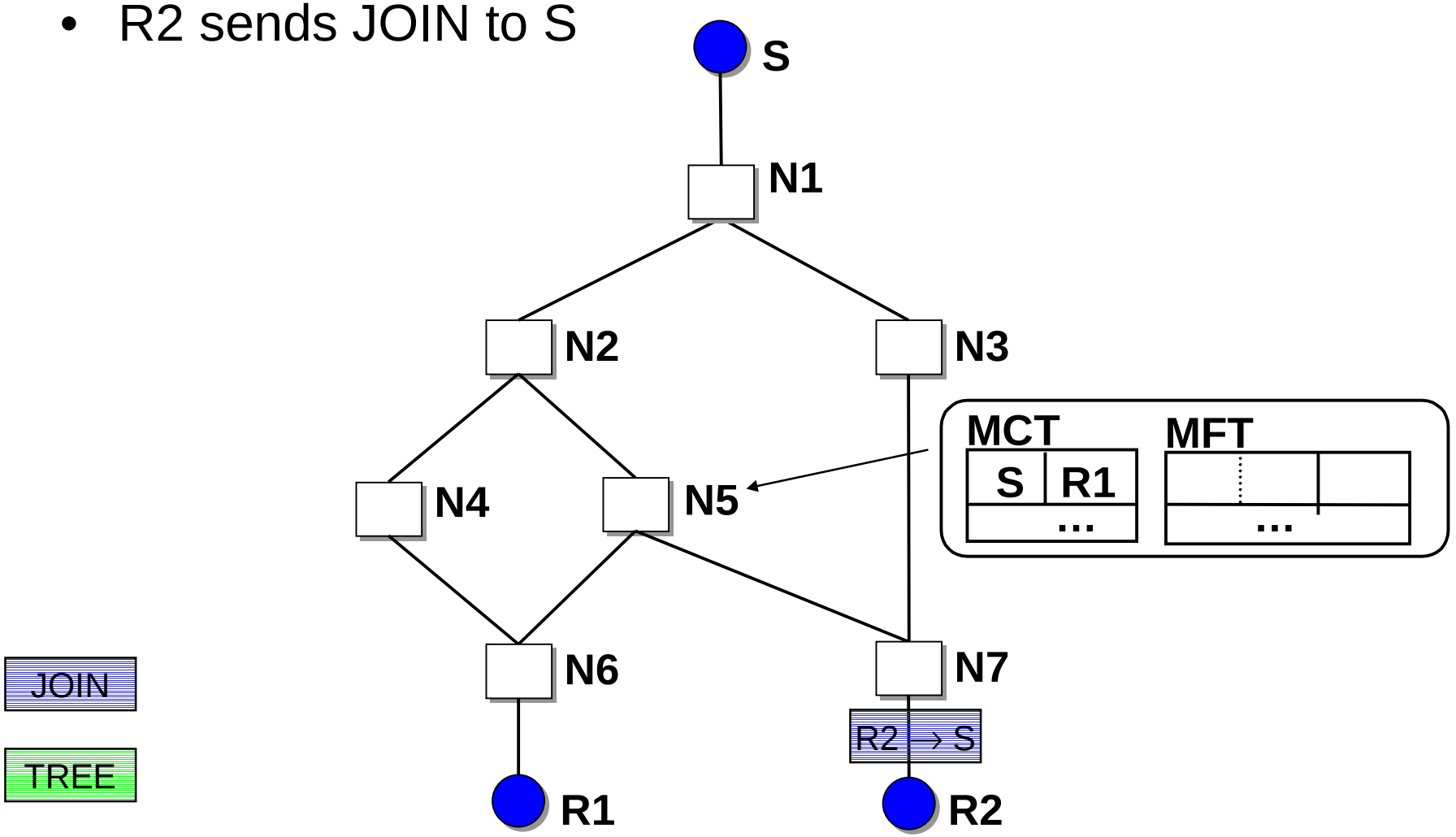
Joining a Group

- R2 sends JOIN to S



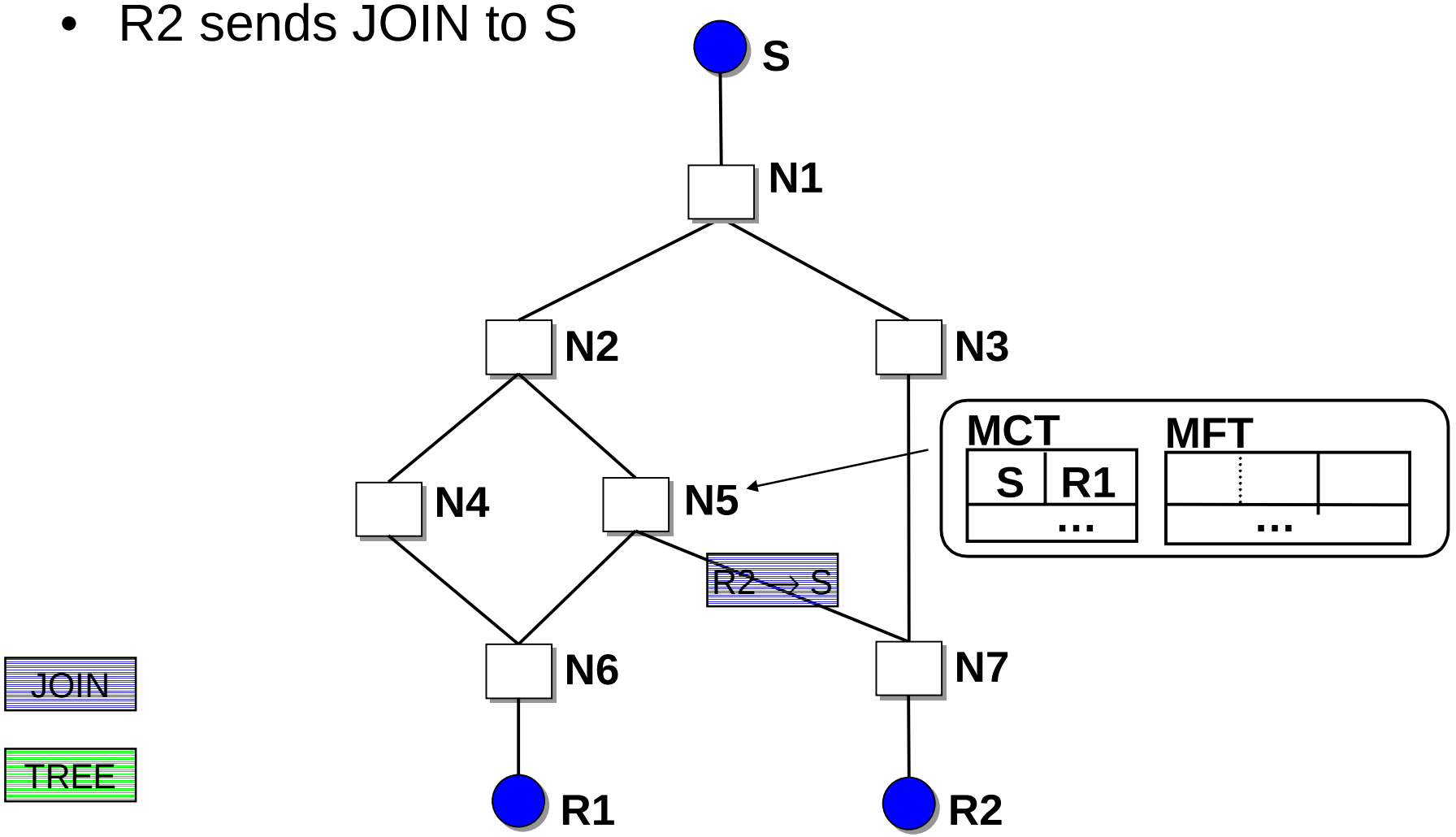
Joining a Group

- R2 sends JOIN to S



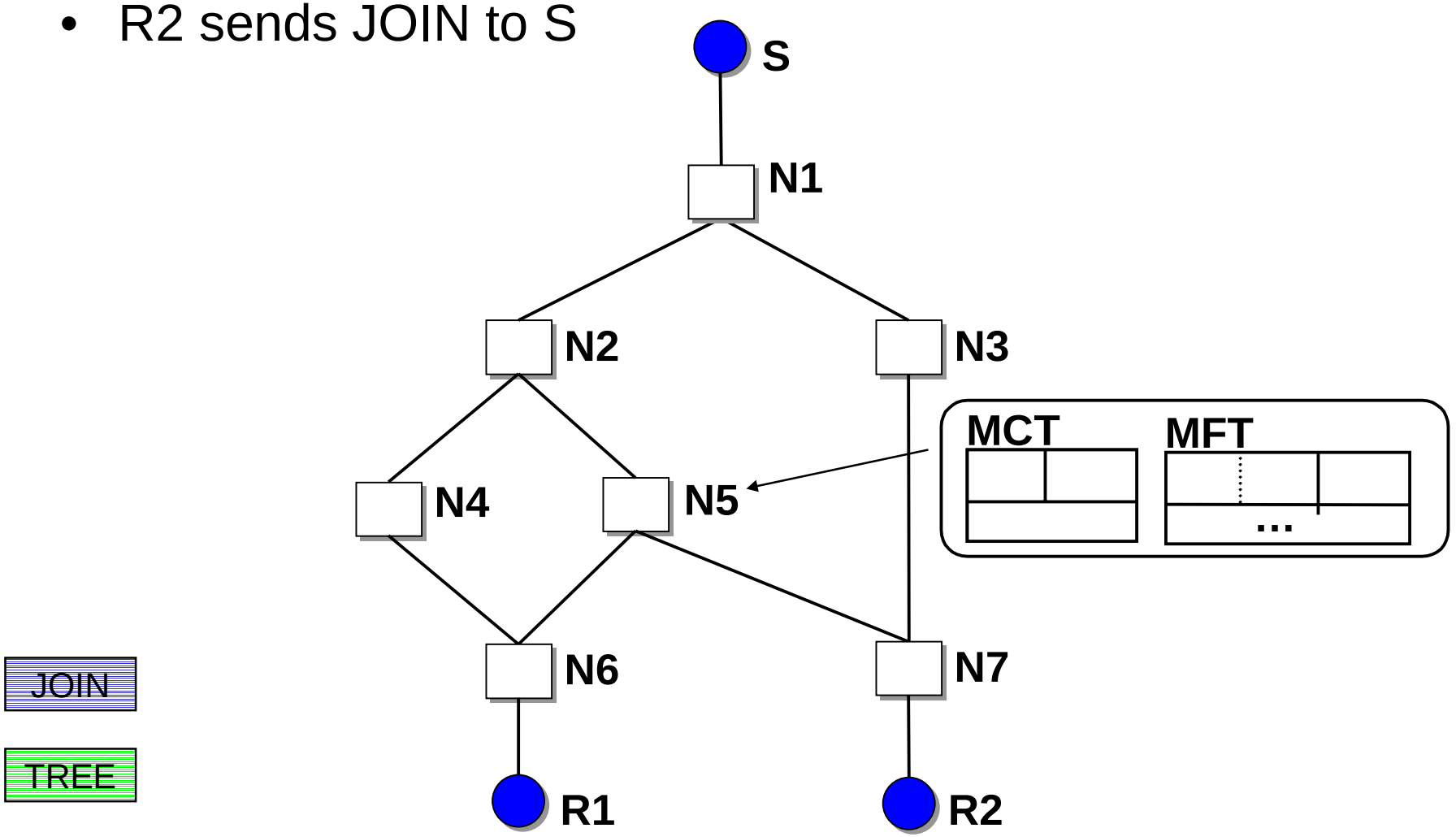
Joining a Group

- R2 sends JOIN to S



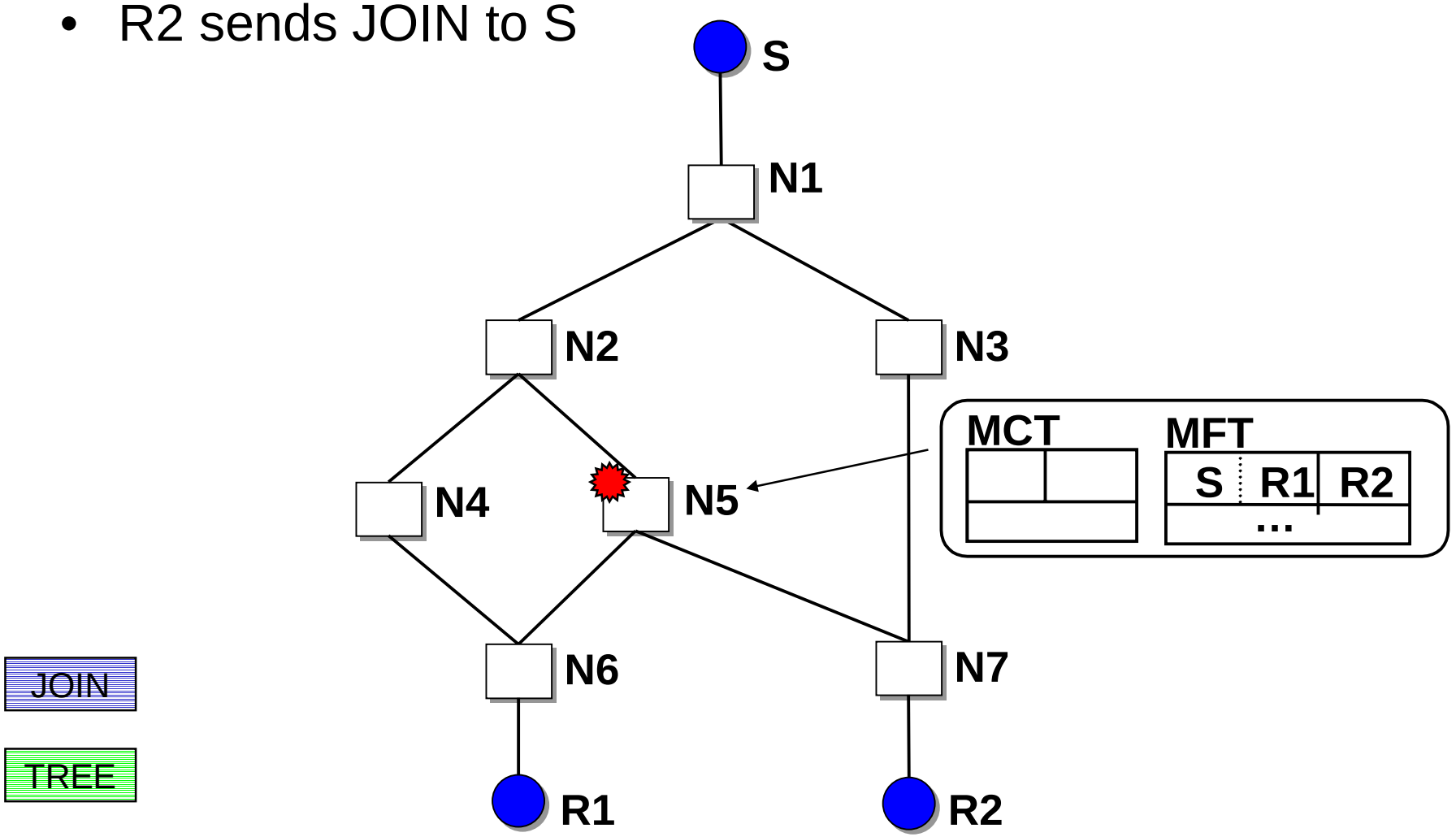
Joining a Group

- R2 sends JOIN to S



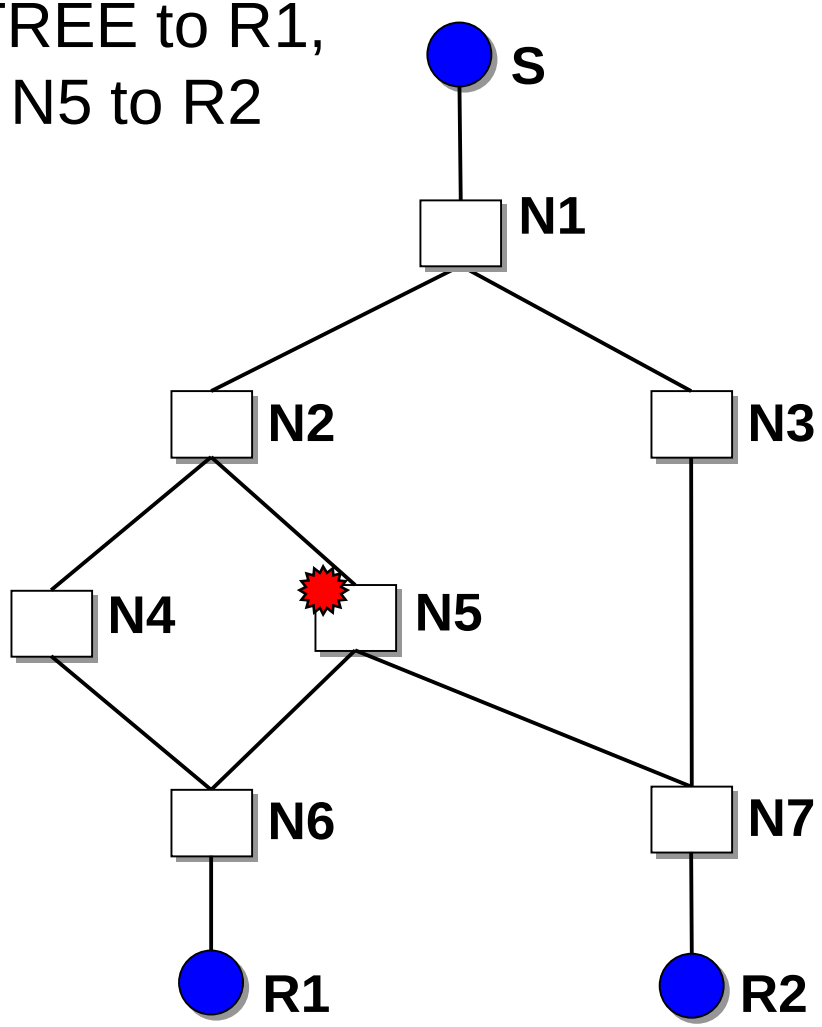
Joining a Group

- R2 sends JOIN to S



Joining a Group

- S sends TREE to R1, copied by N5 to R2

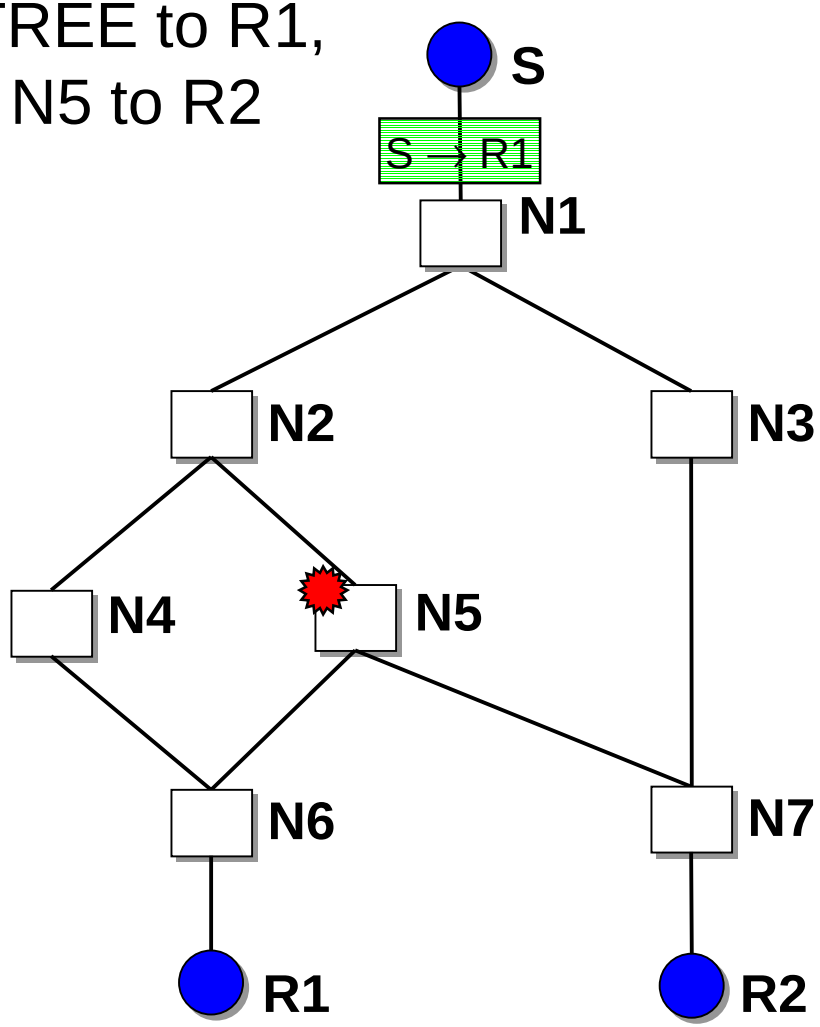


JOIN

TREE

Joining a Group

- S sends TREE to R1, copied by N5 to R2

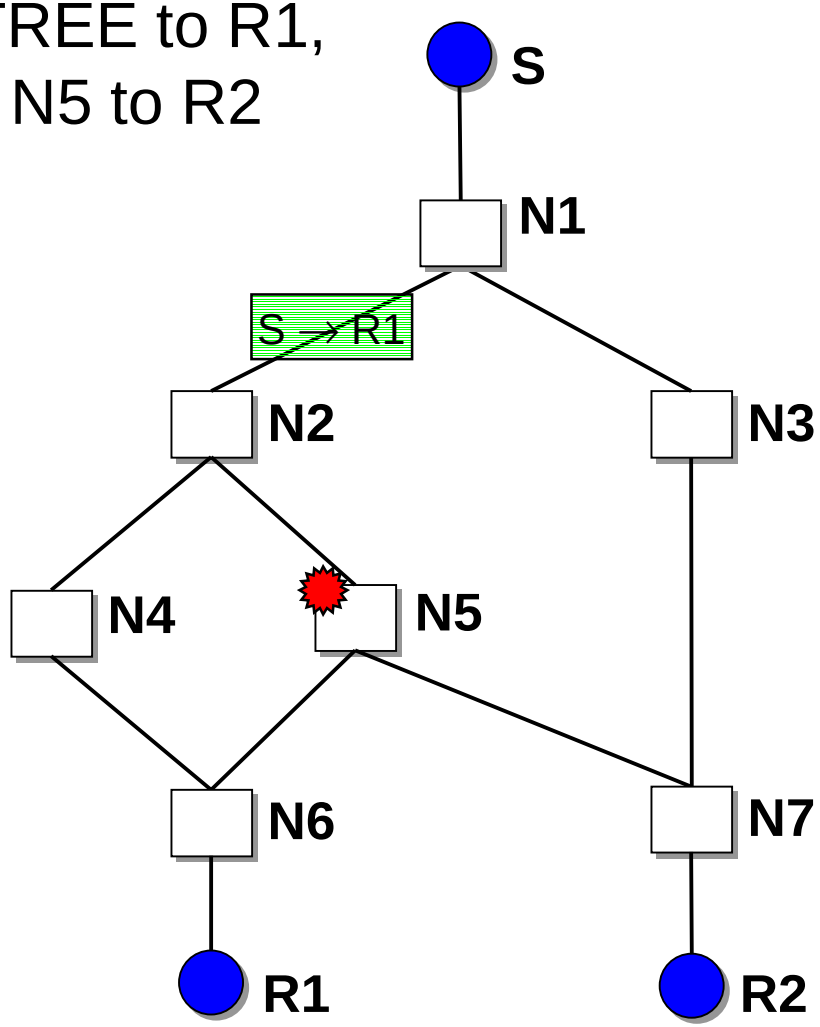


JOIN

TREE

Joining a Group

- S sends TREE to R1, copied by N5 to R2

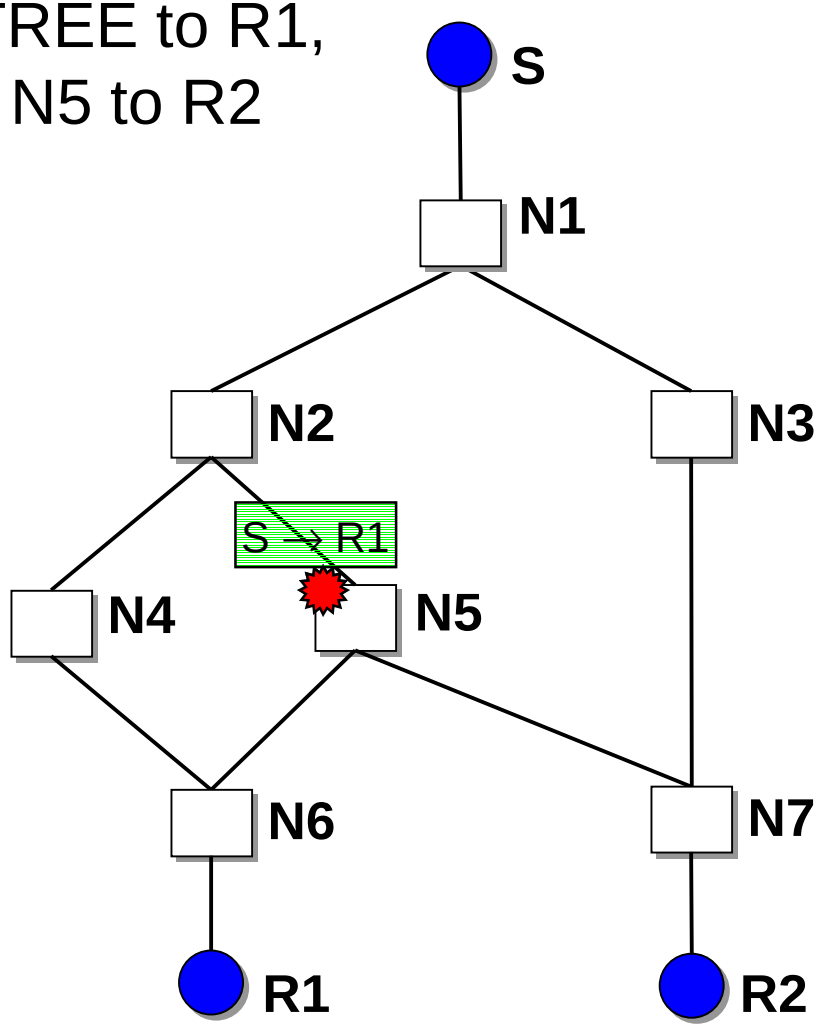


JOIN

TREE

Joining a Group

- S sends TREE to R1, copied by N5 to R2

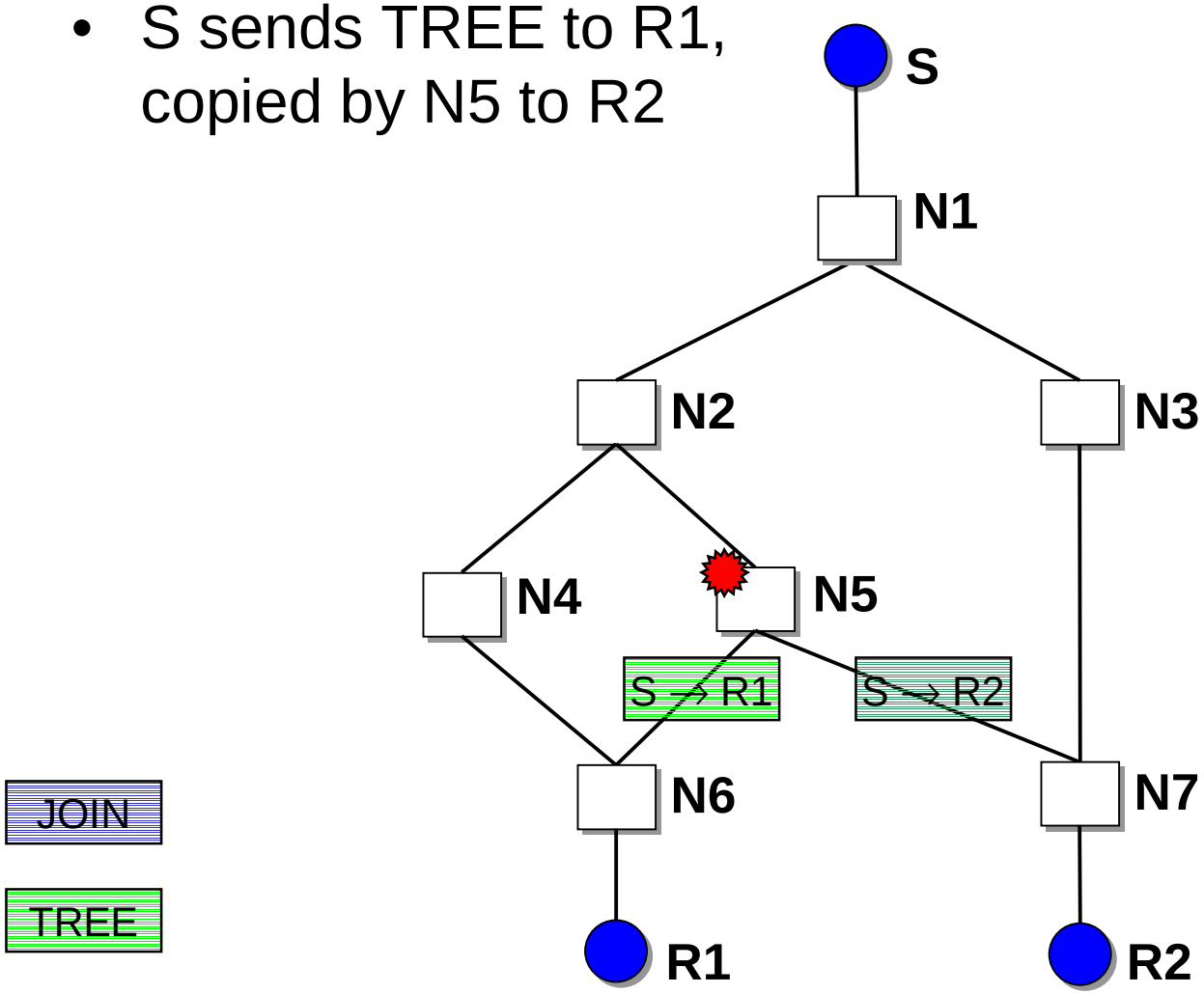


JOIN

TREE

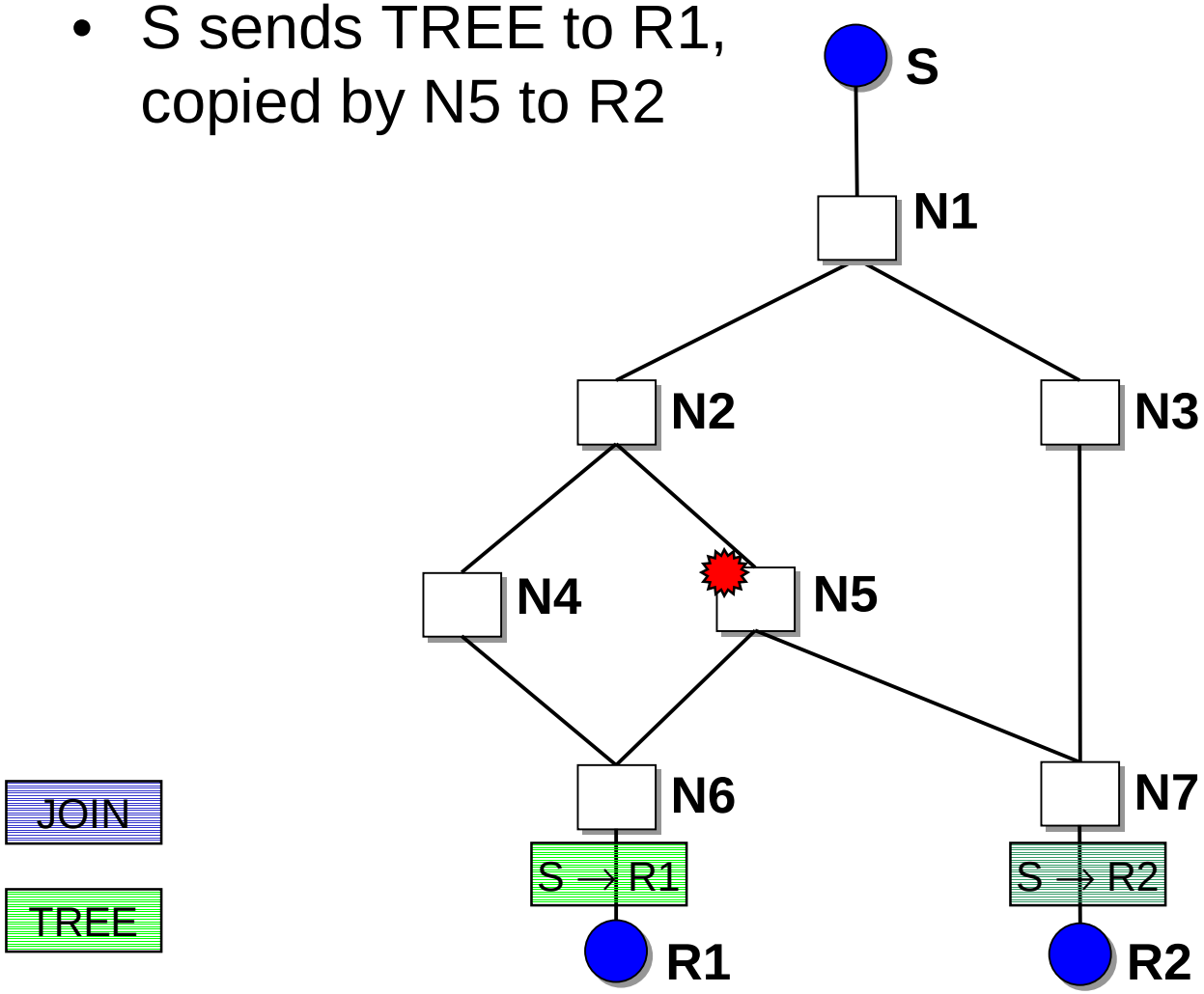
Joining a Group

- S sends TREE to R1, copied by N5 to R2



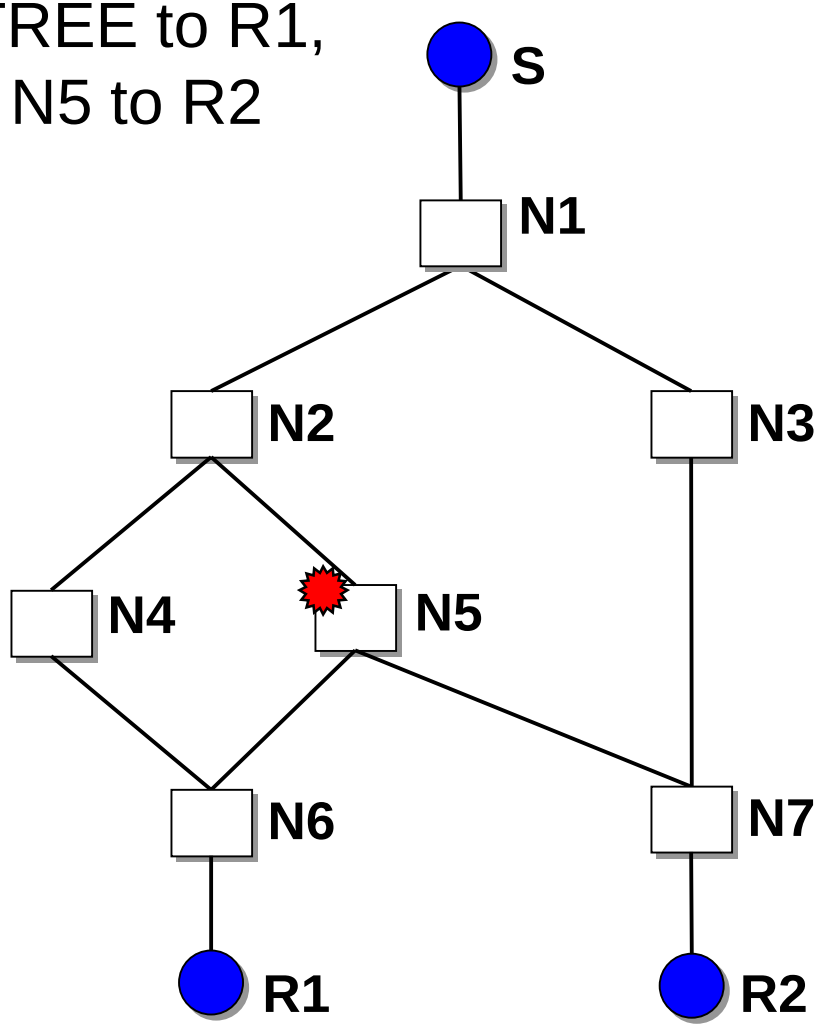
Joining a Group

- S sends TREE to R1, copied by N5 to R2



Joining a Group

- S sends TREE to R1, copied by N5 to R2

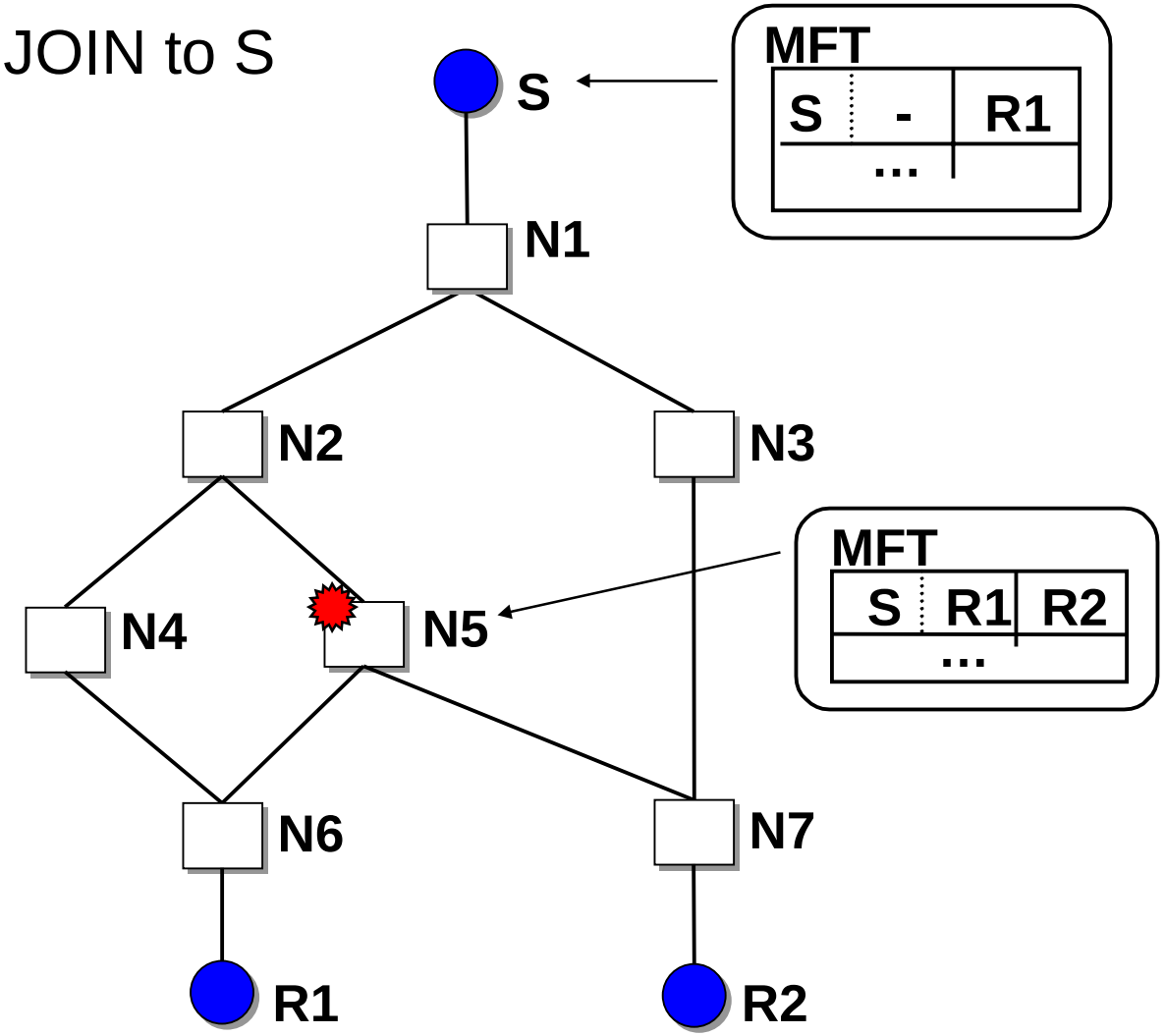


JOIN

TREE

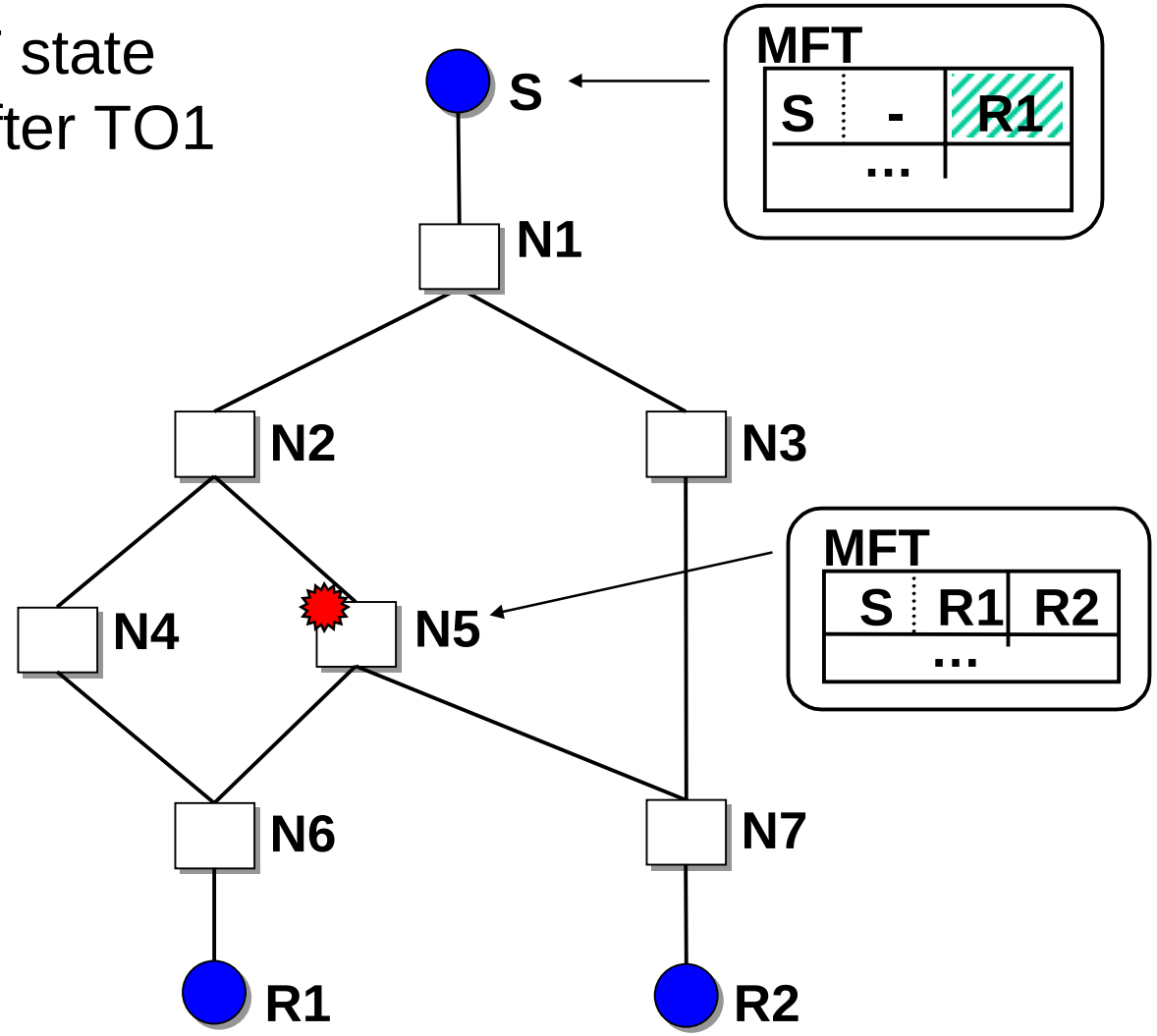
Leaving a Group

- R1 stops JOIN to S



Leaving a Group

- R1's MFT state expires after TO1

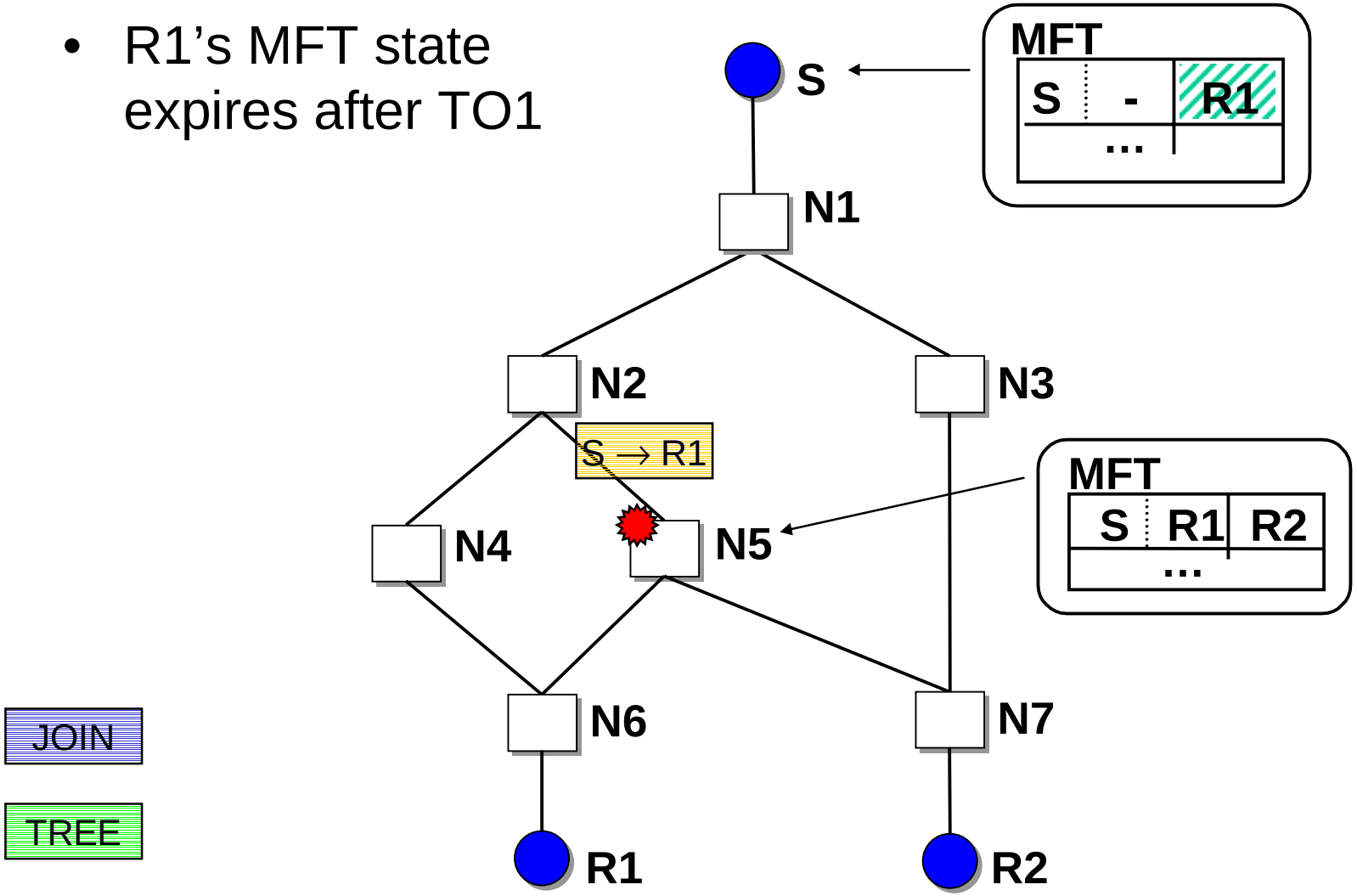


JOIN

TREE

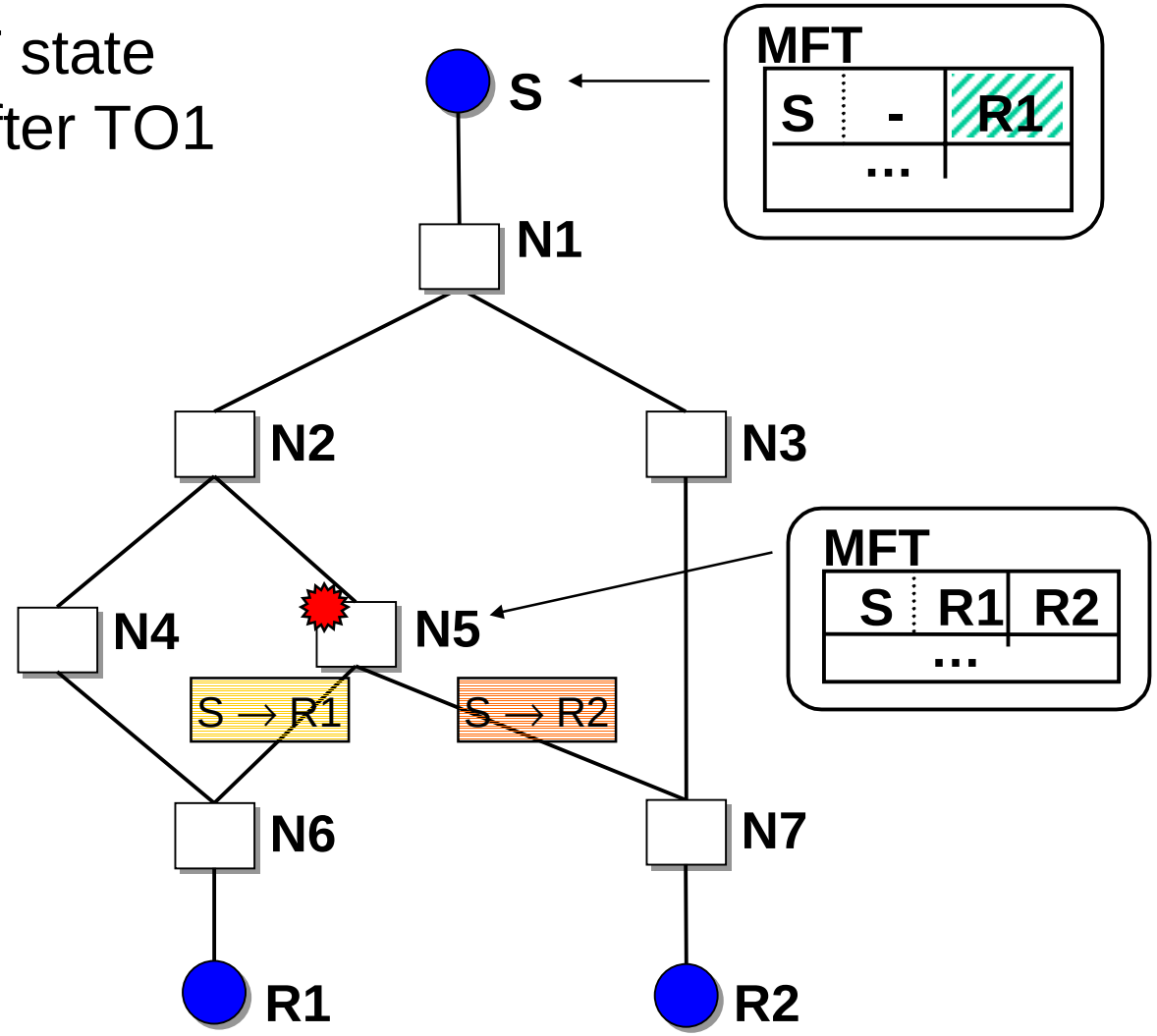
Leaving a Group

- R1's MFT state expires after TO1



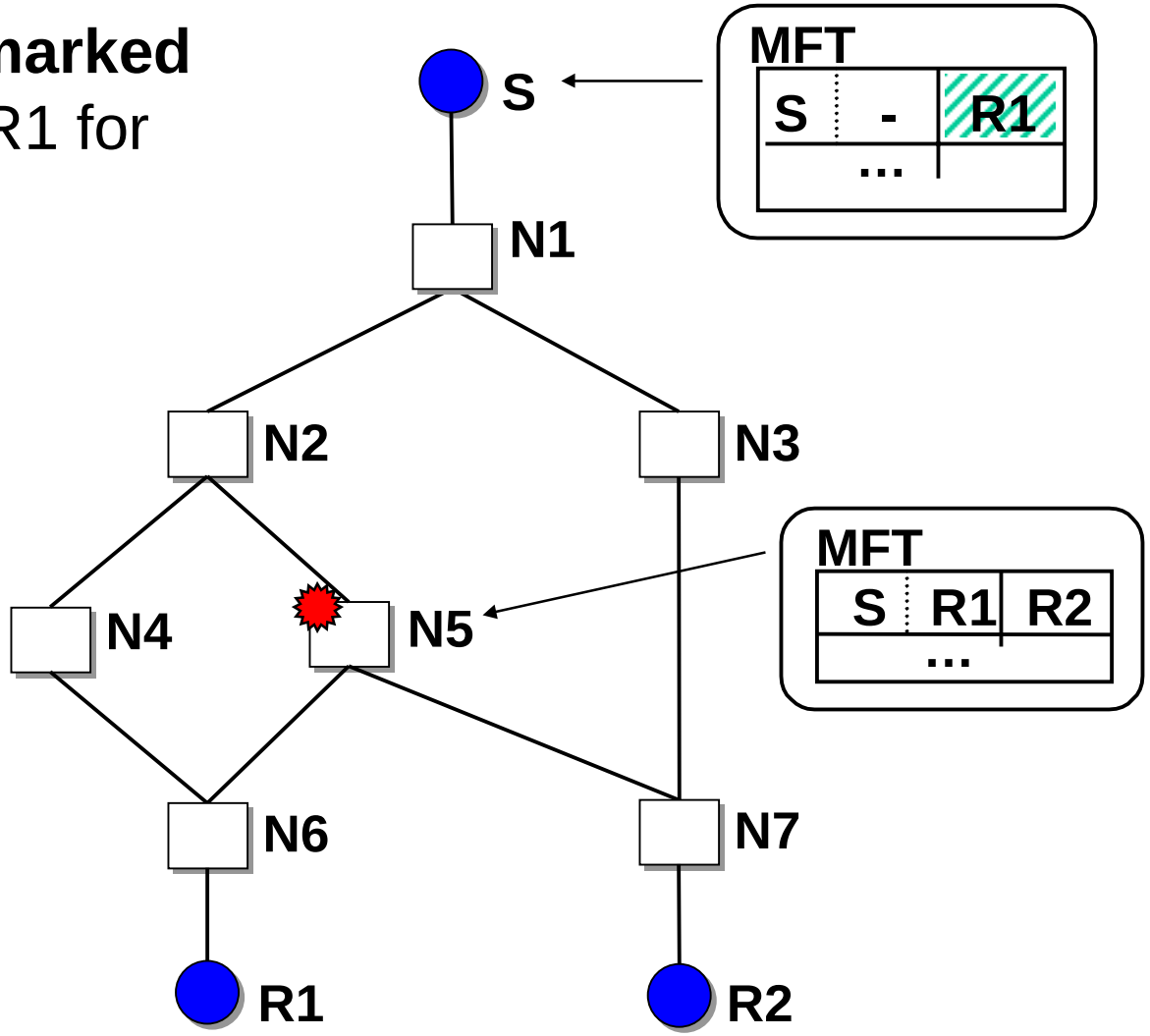
Leaving a Group

- R1's MFT state expires after TO1



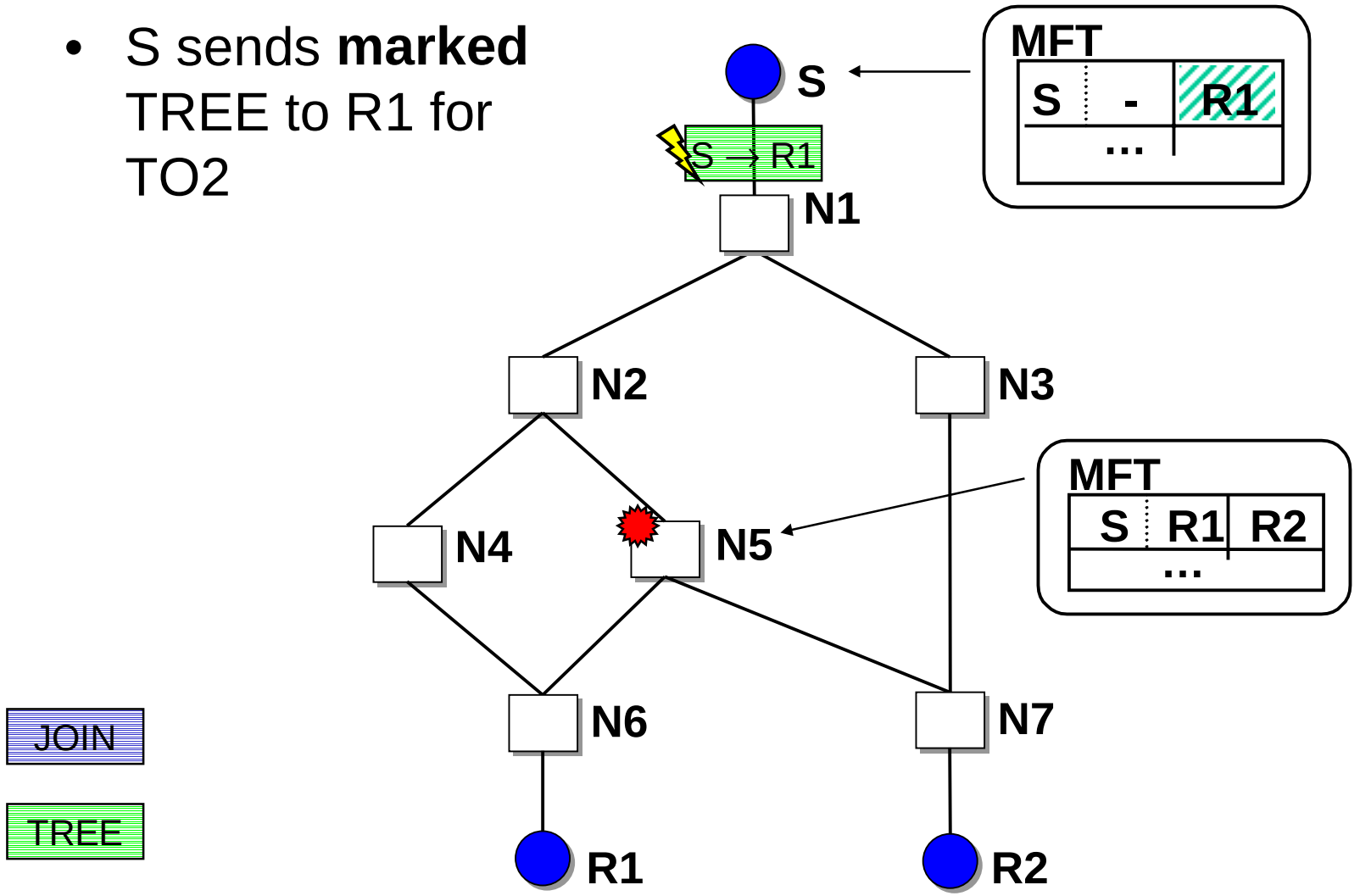
Leaving a Group

- S sends **marked** TREE to R1 for TO2



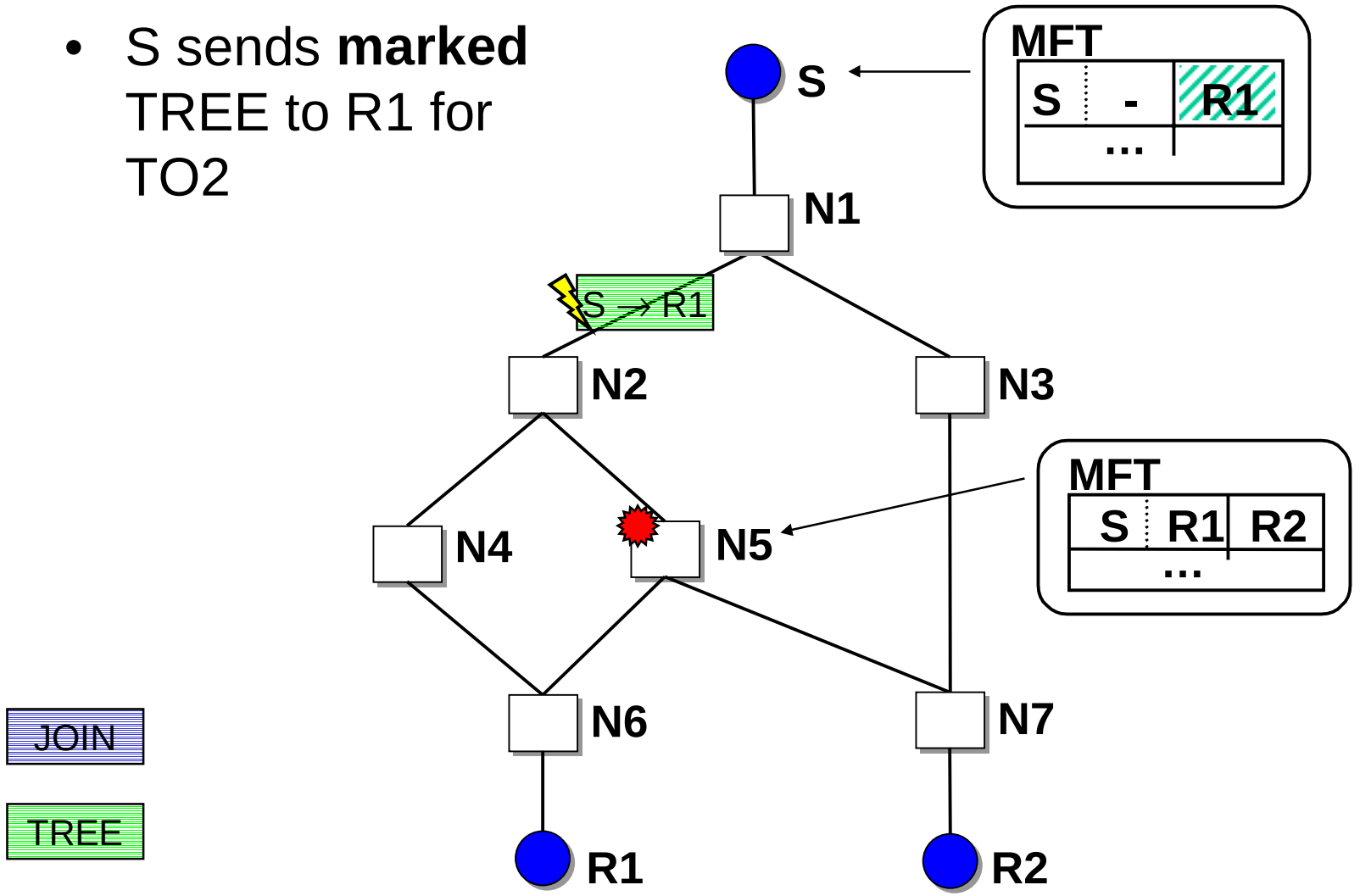
Leaving a Group

- S sends **marked** TREE to R1 for TO2



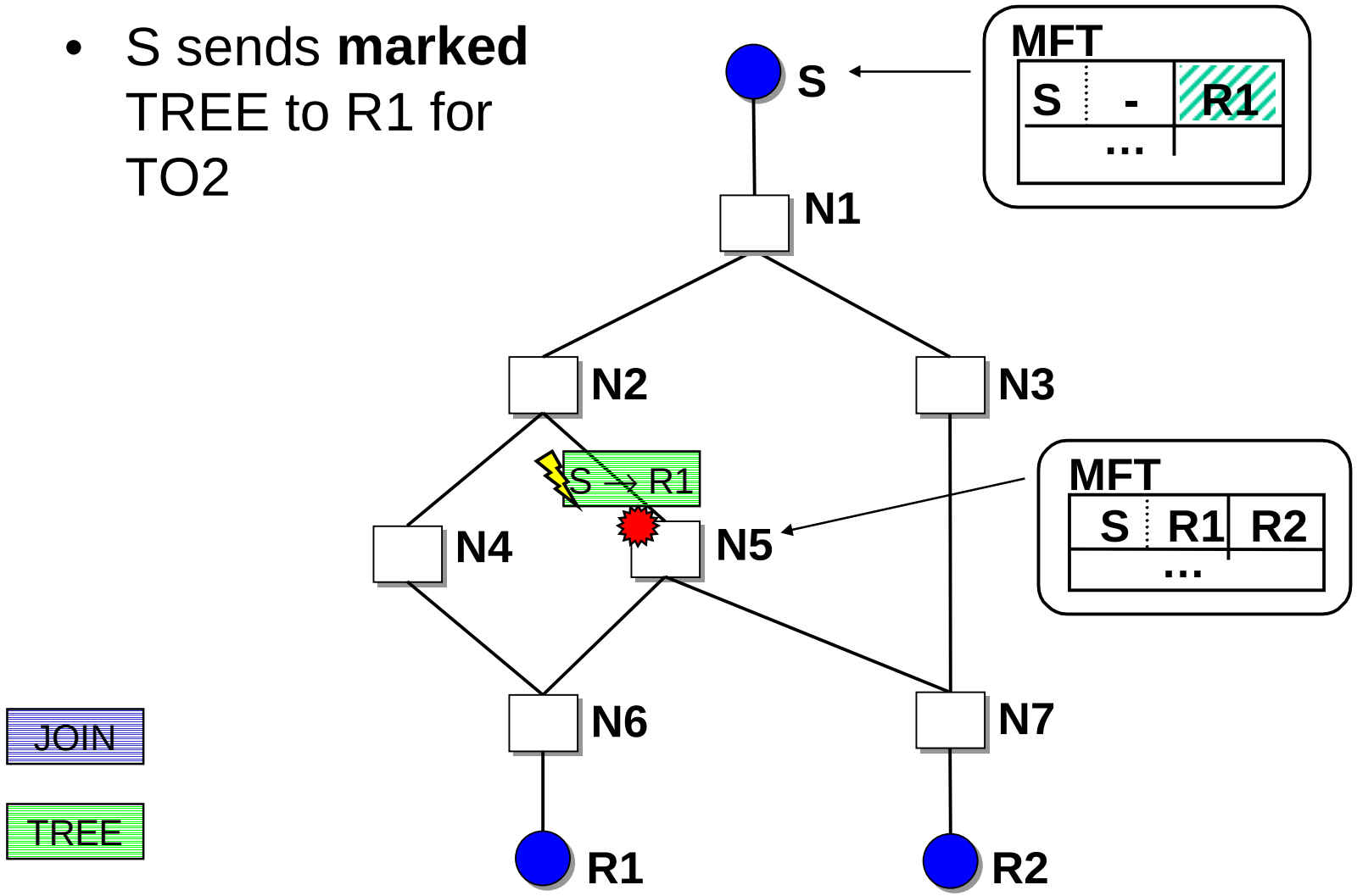
Leaving a Group

- S sends **marked** TREE to R1 for TO2



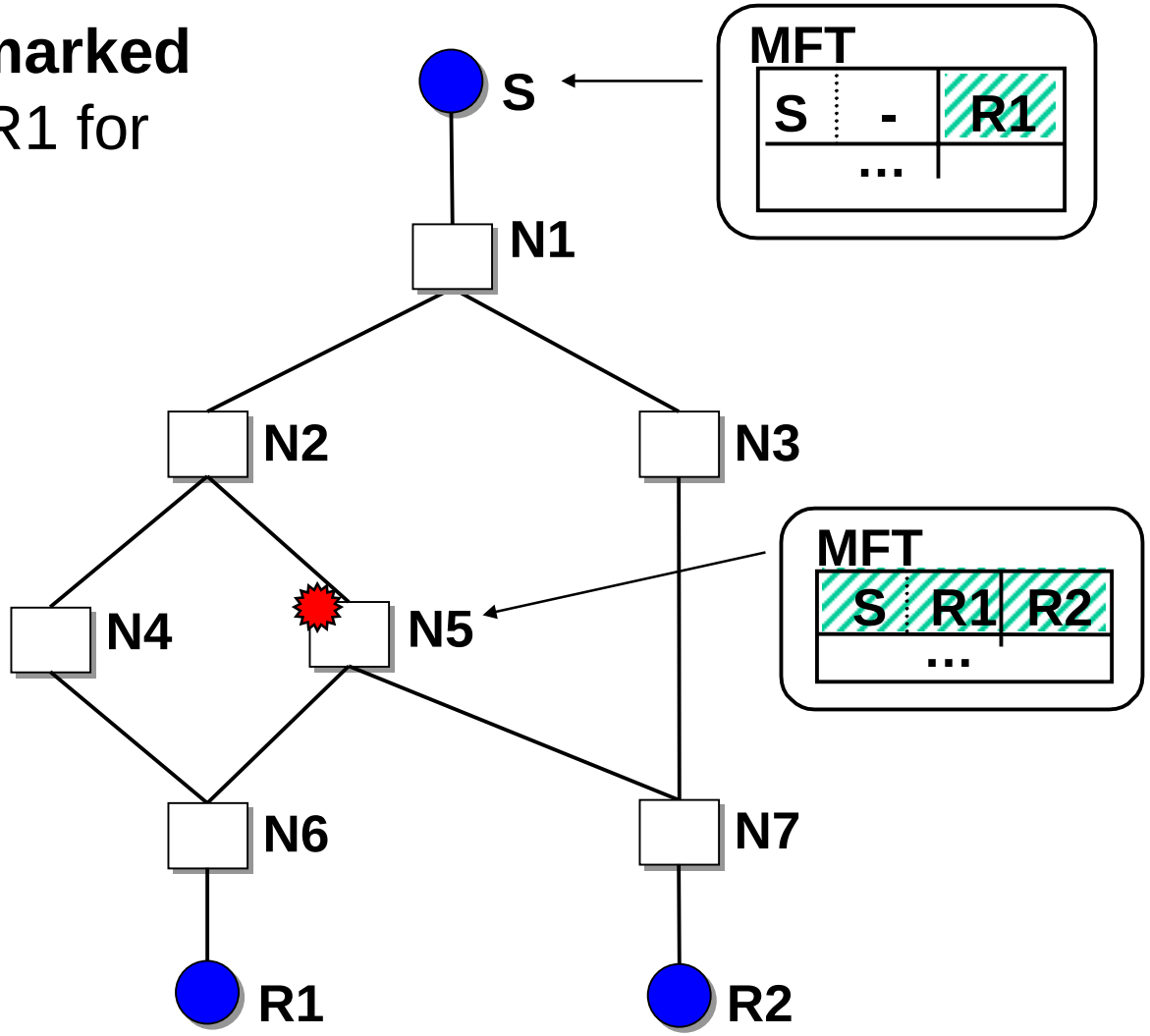
Leaving a Group

- S sends **marked** TREE to R1 for TO2



Leaving a Group

- S sends **marked** TREE to R1 for TO2

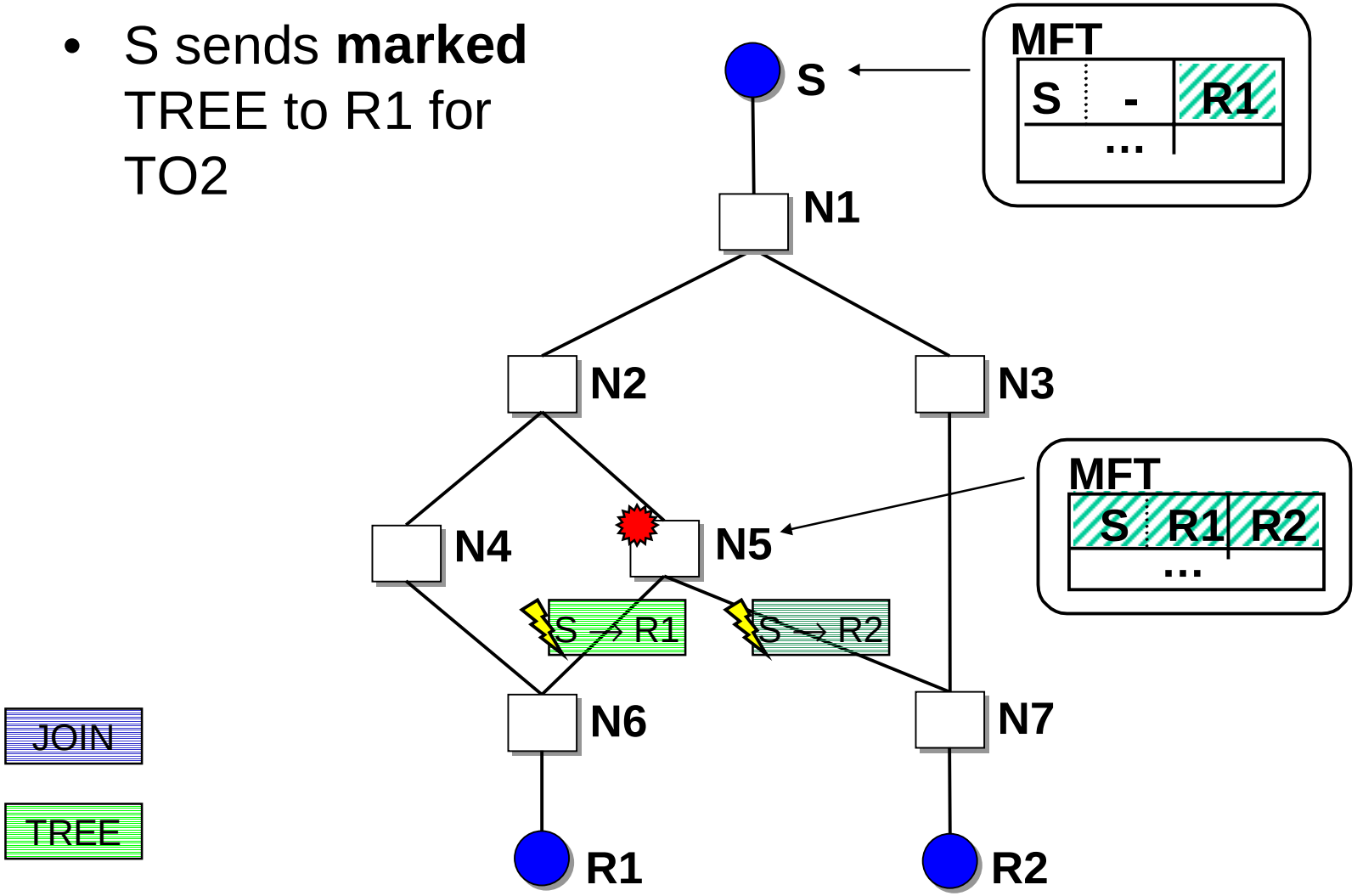


JOIN

TREE

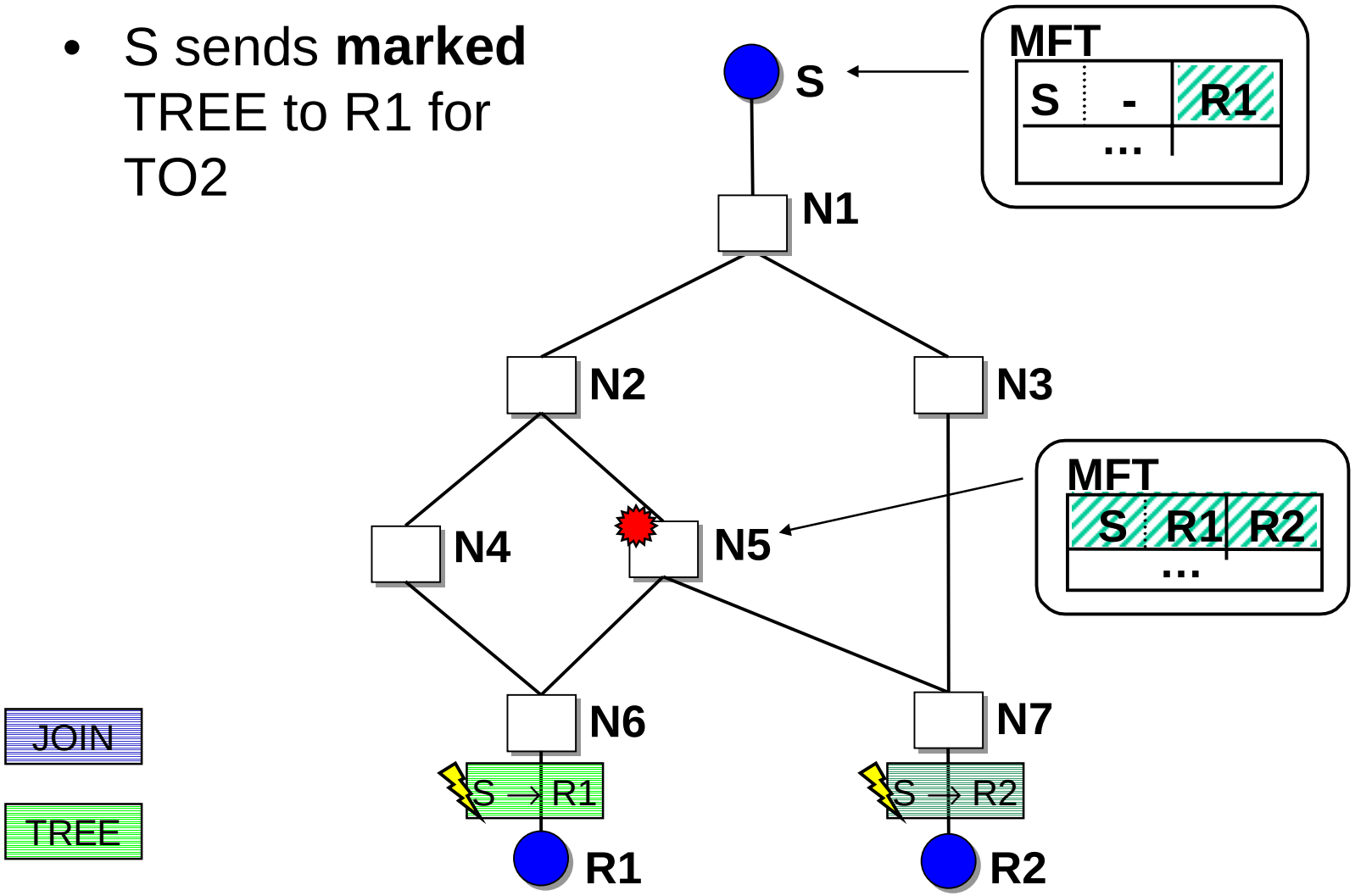
Leaving a Group

- S sends **marked** TREE to R1 for TO2



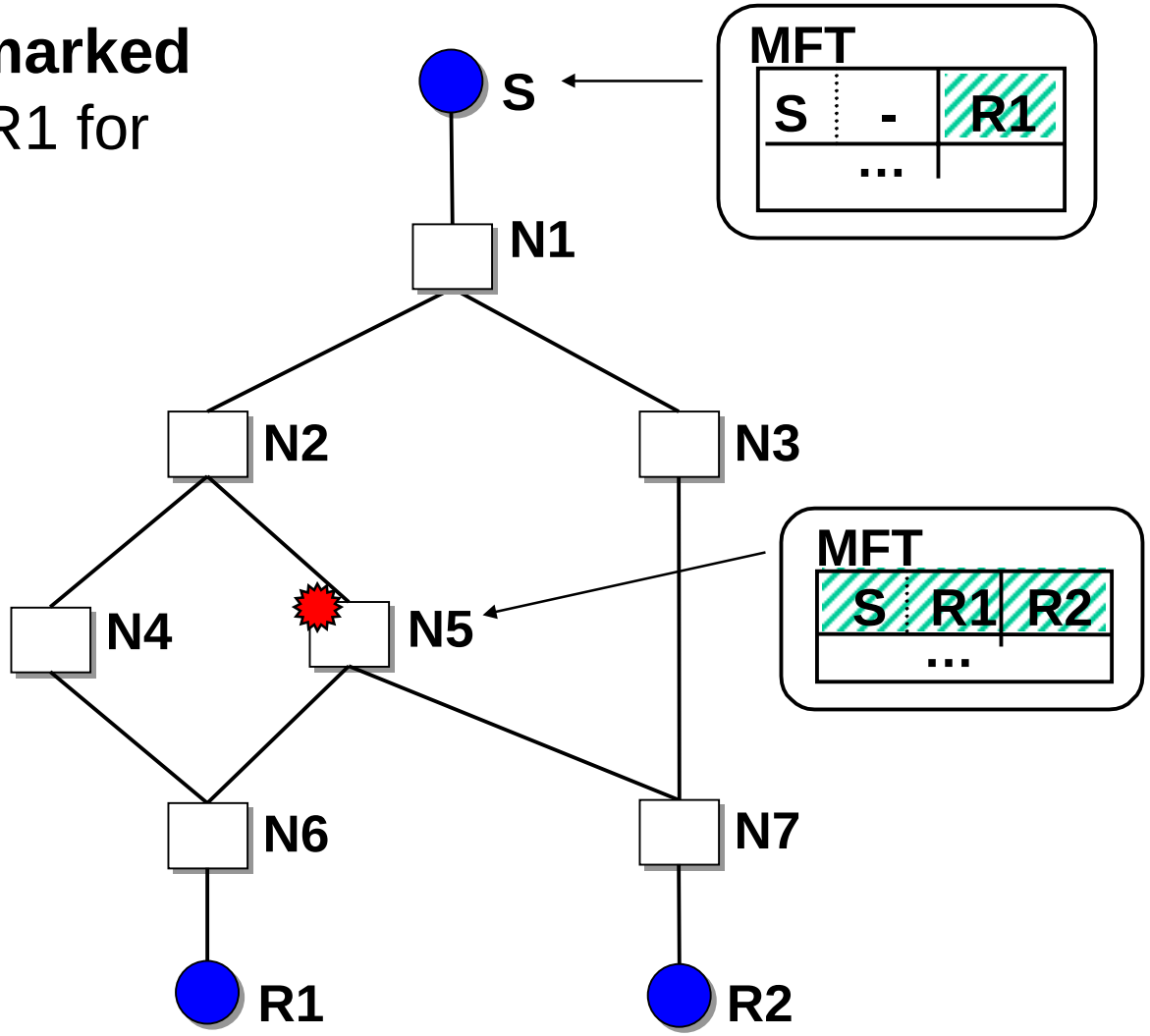
Leaving a Group

- S sends **marked** TREE to R1 for TO2



Leaving a Group

- S sends **marked** TREE to R1 for TO2

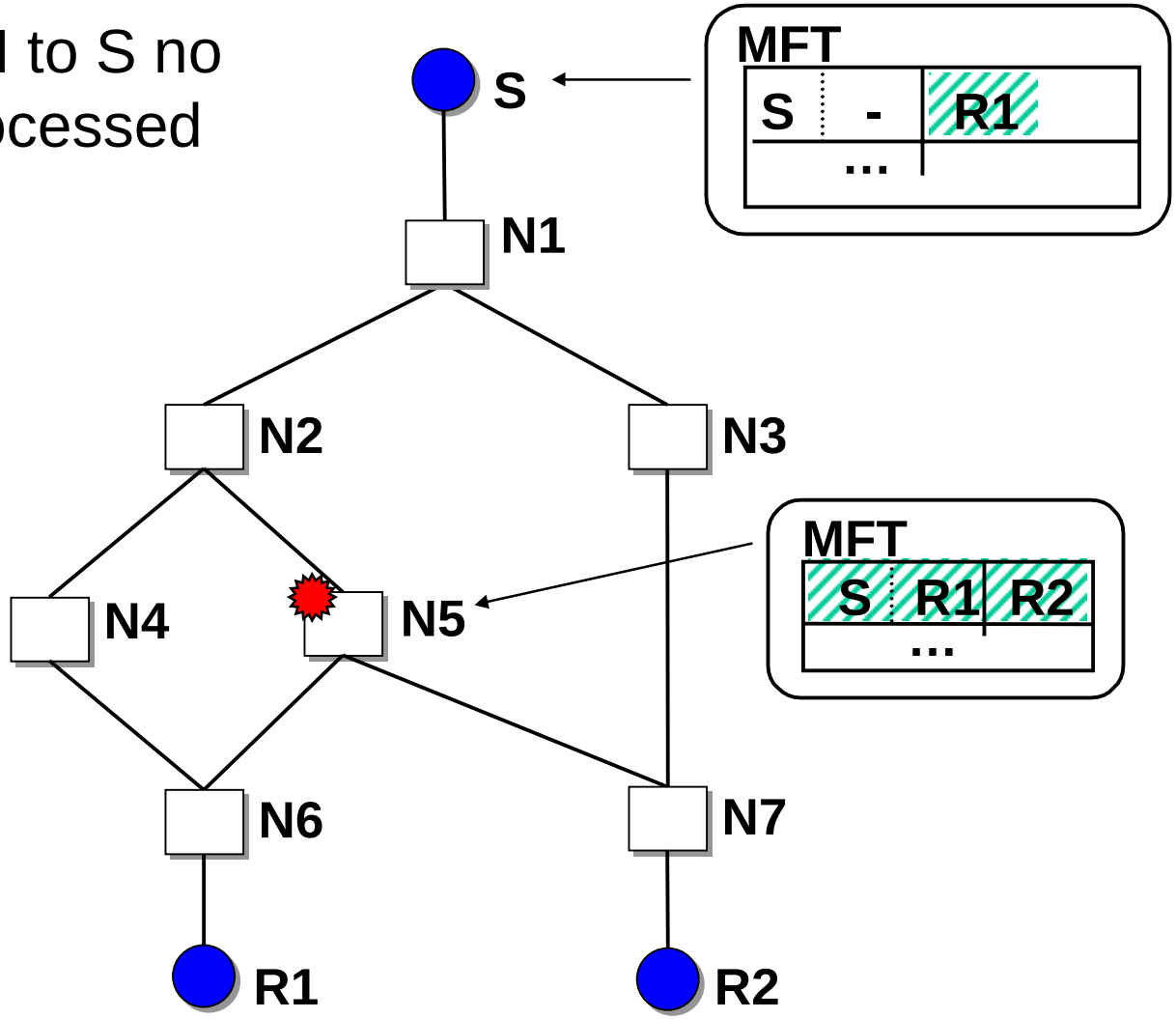


JOIN

TREE

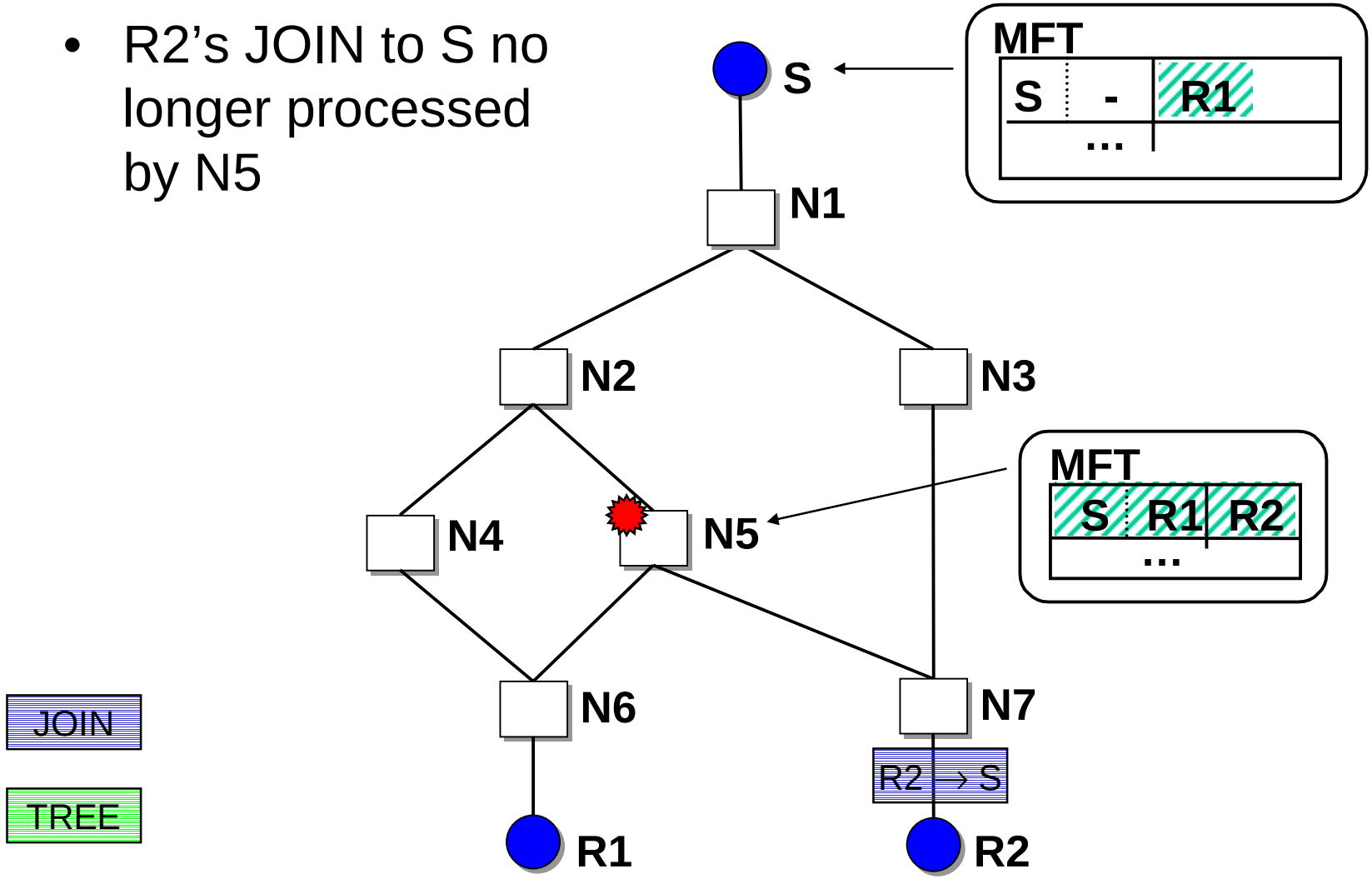
Leaving a Group

- R2's JOIN to S no longer processed by N5



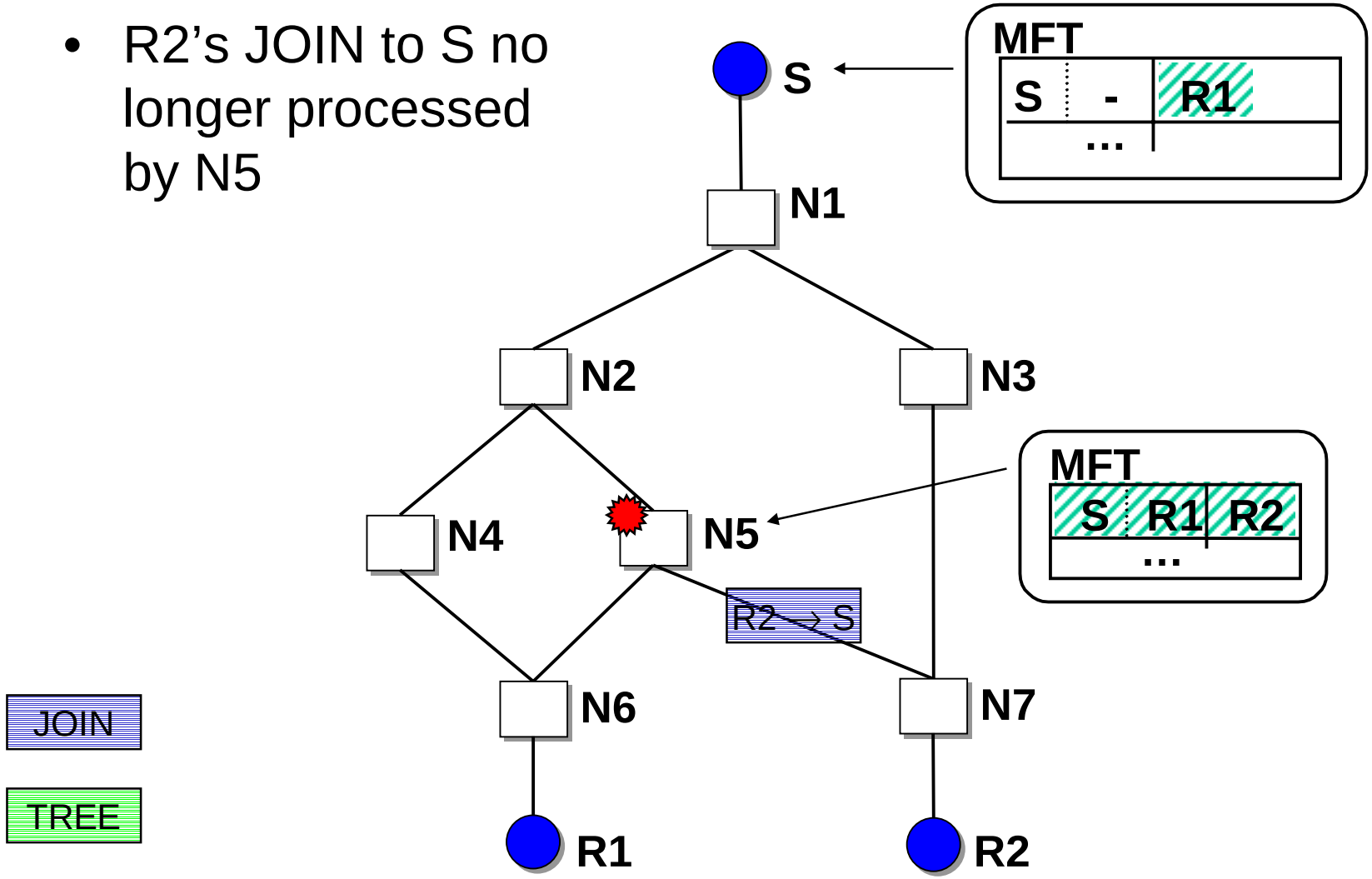
Leaving a Group

- R2's JOIN to S no longer processed by N5



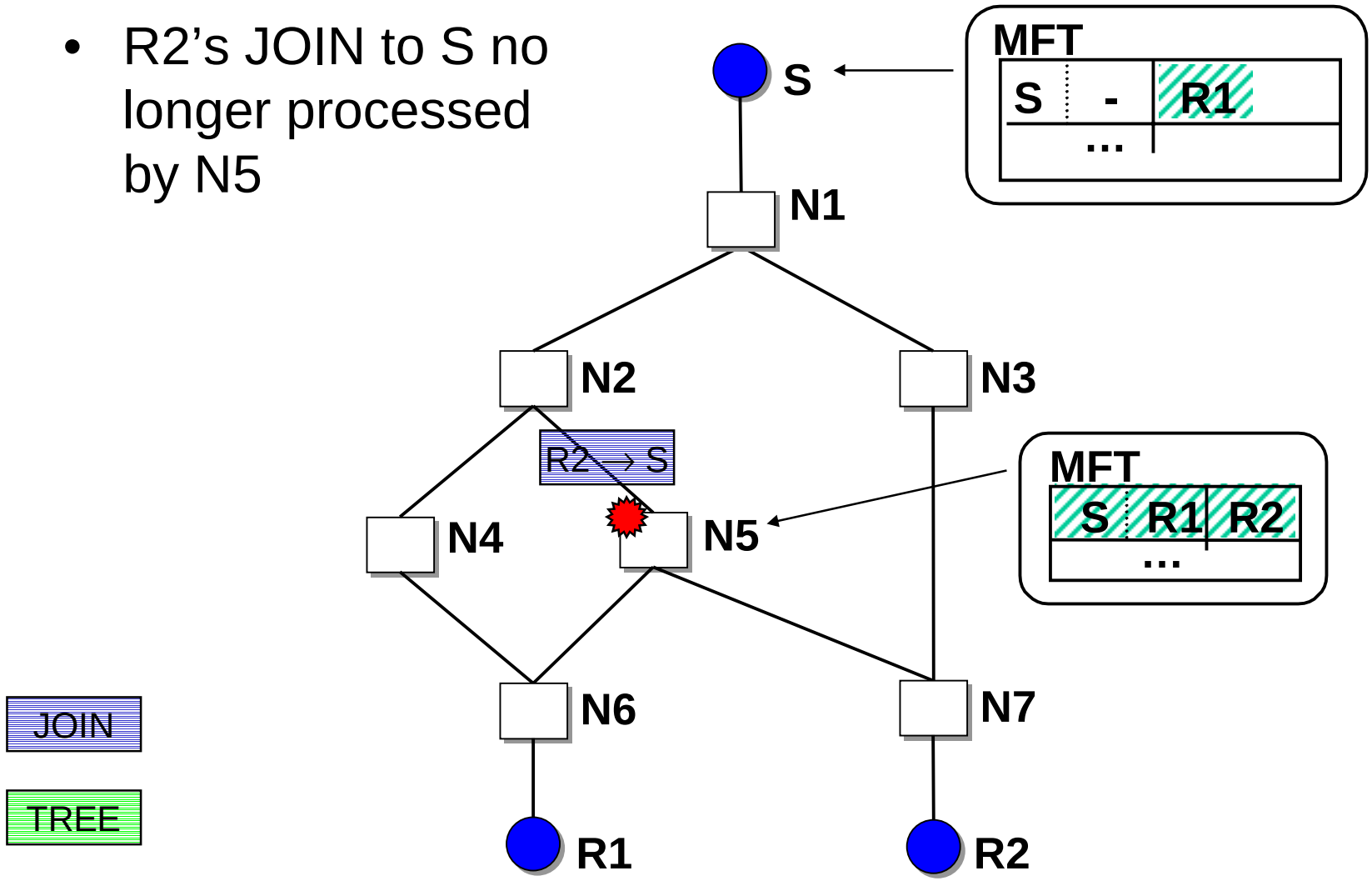
Leaving a Group

- R2's JOIN to S no longer processed by N5



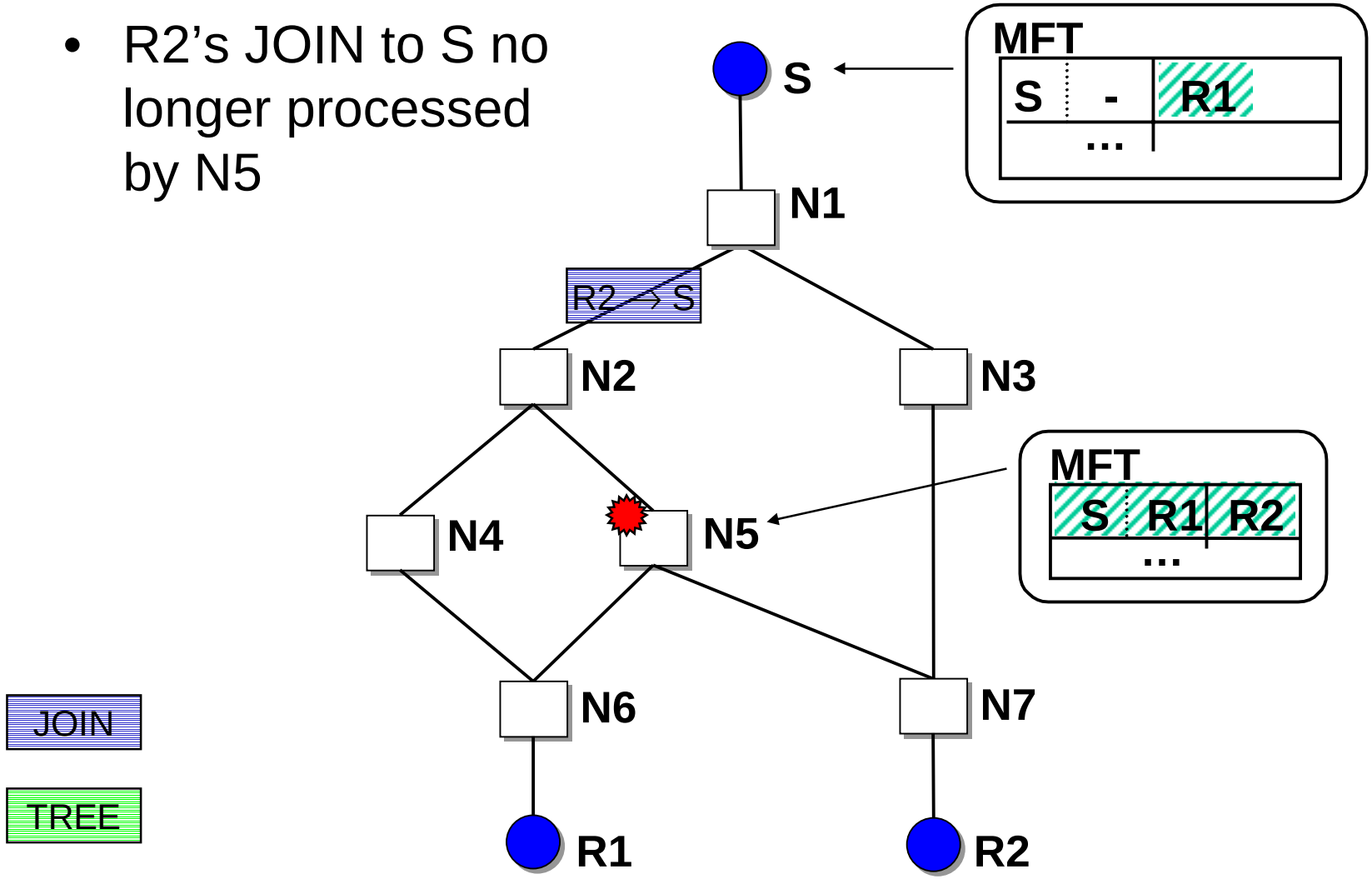
Leaving a Group

- R2's JOIN to S no longer processed by N5



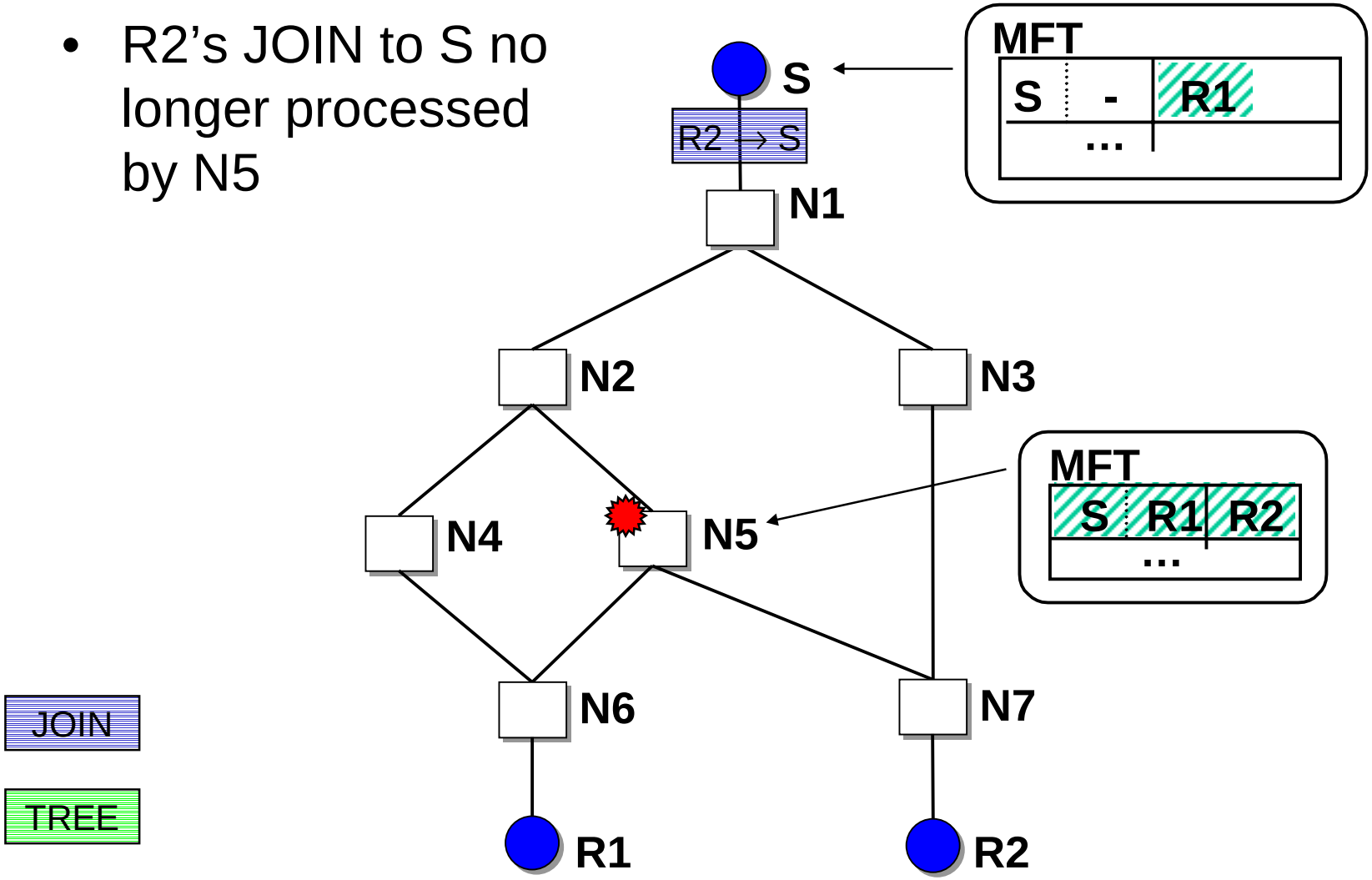
Leaving a Group

- R2's JOIN to S no longer processed by N5



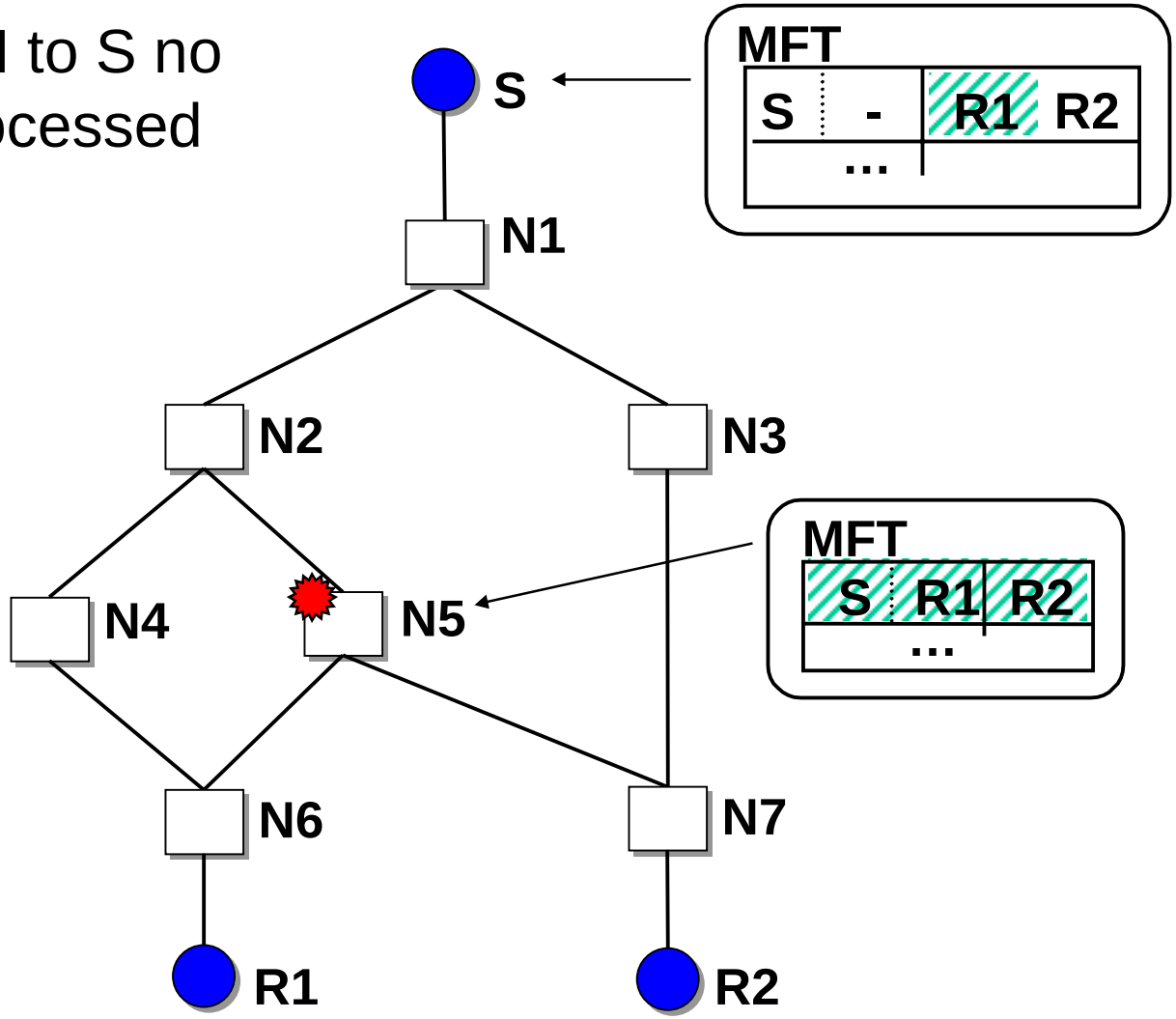
Leaving a Group

- R2's JOIN to S no longer processed by N5



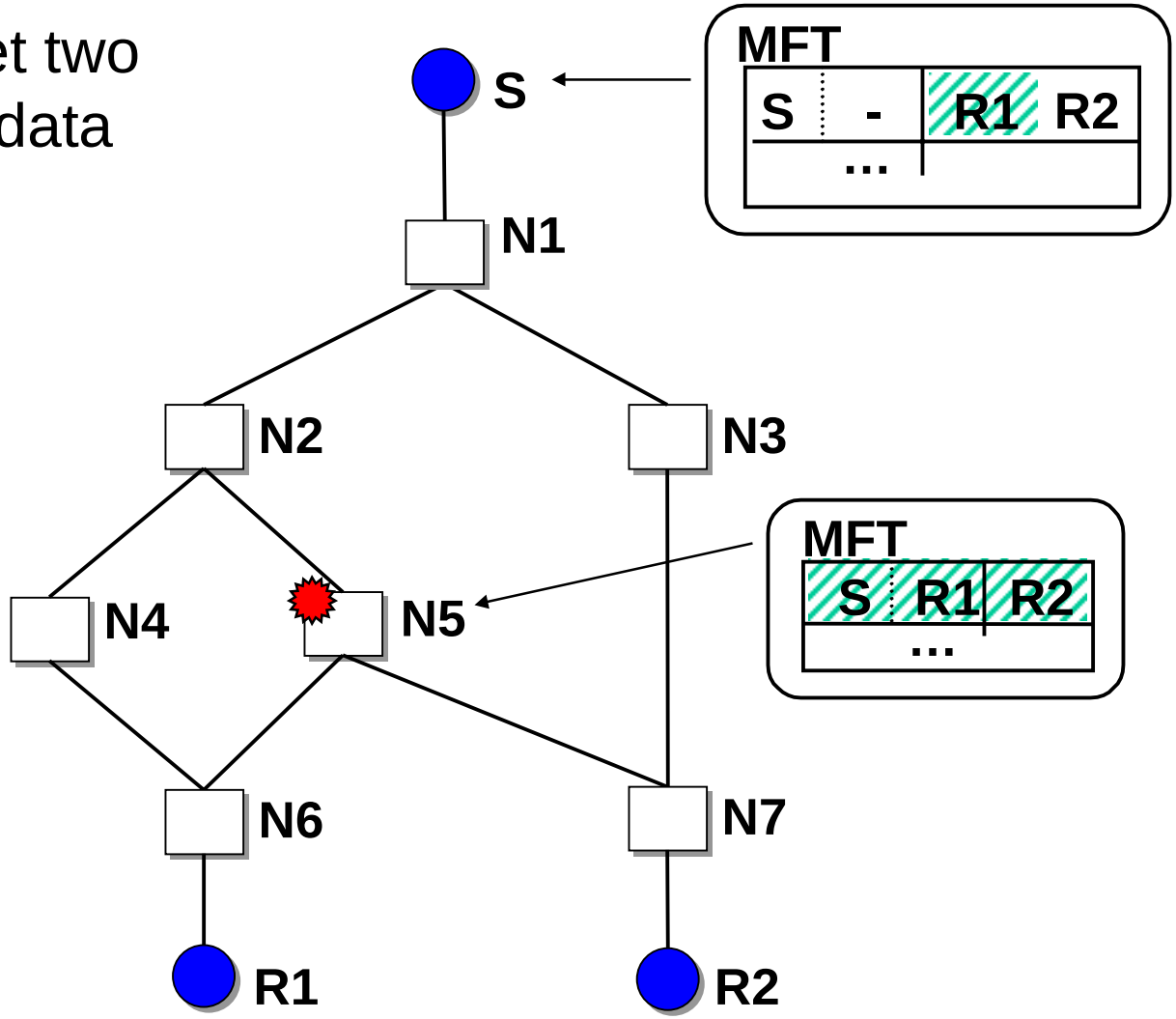
Leaving a Group

- R2's JOIN to S no longer processed by N5



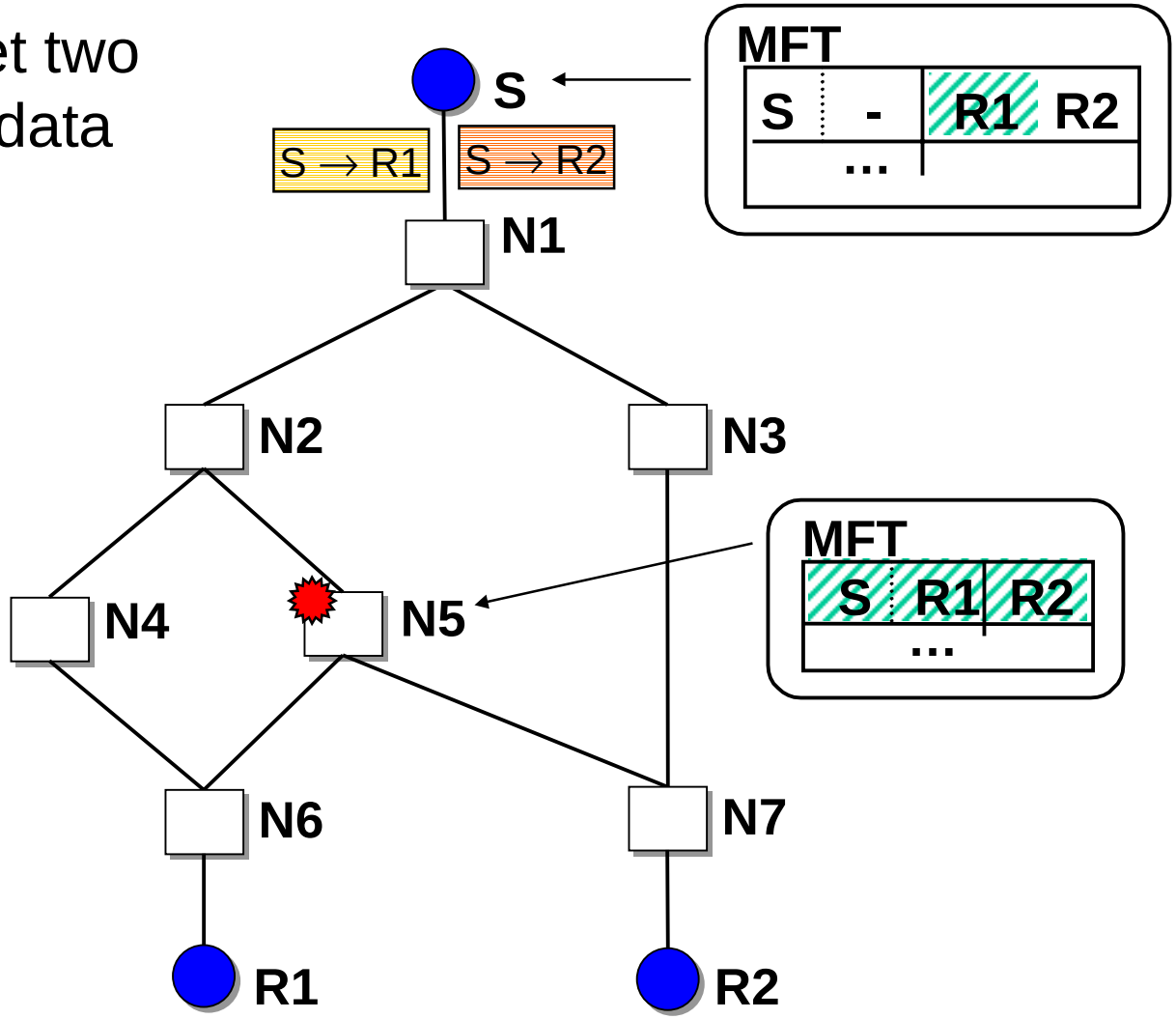
Leaving a Group

- R2 can get two copies of data packets



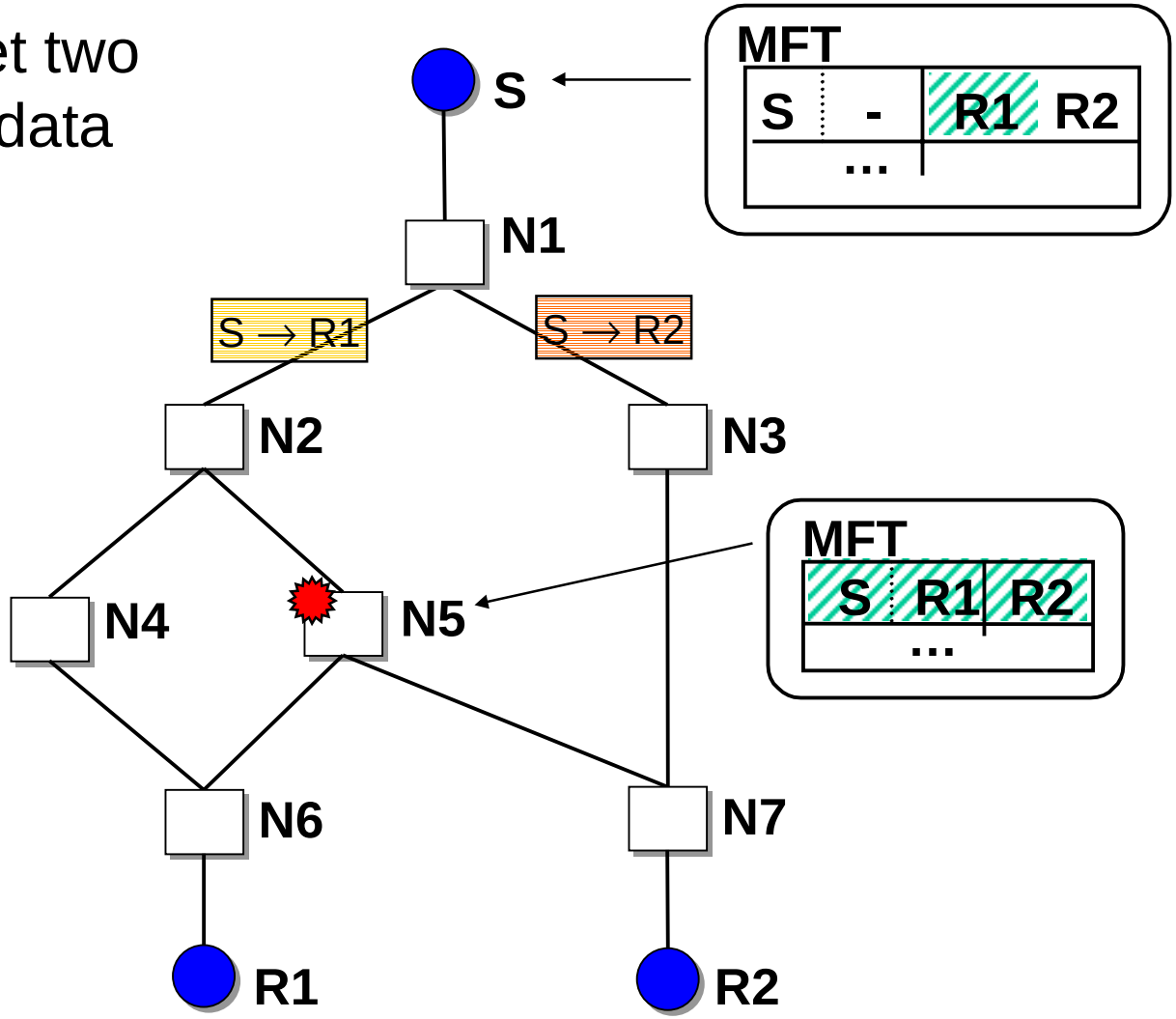
Leaving a Group

- R2 can get two copies of data packets



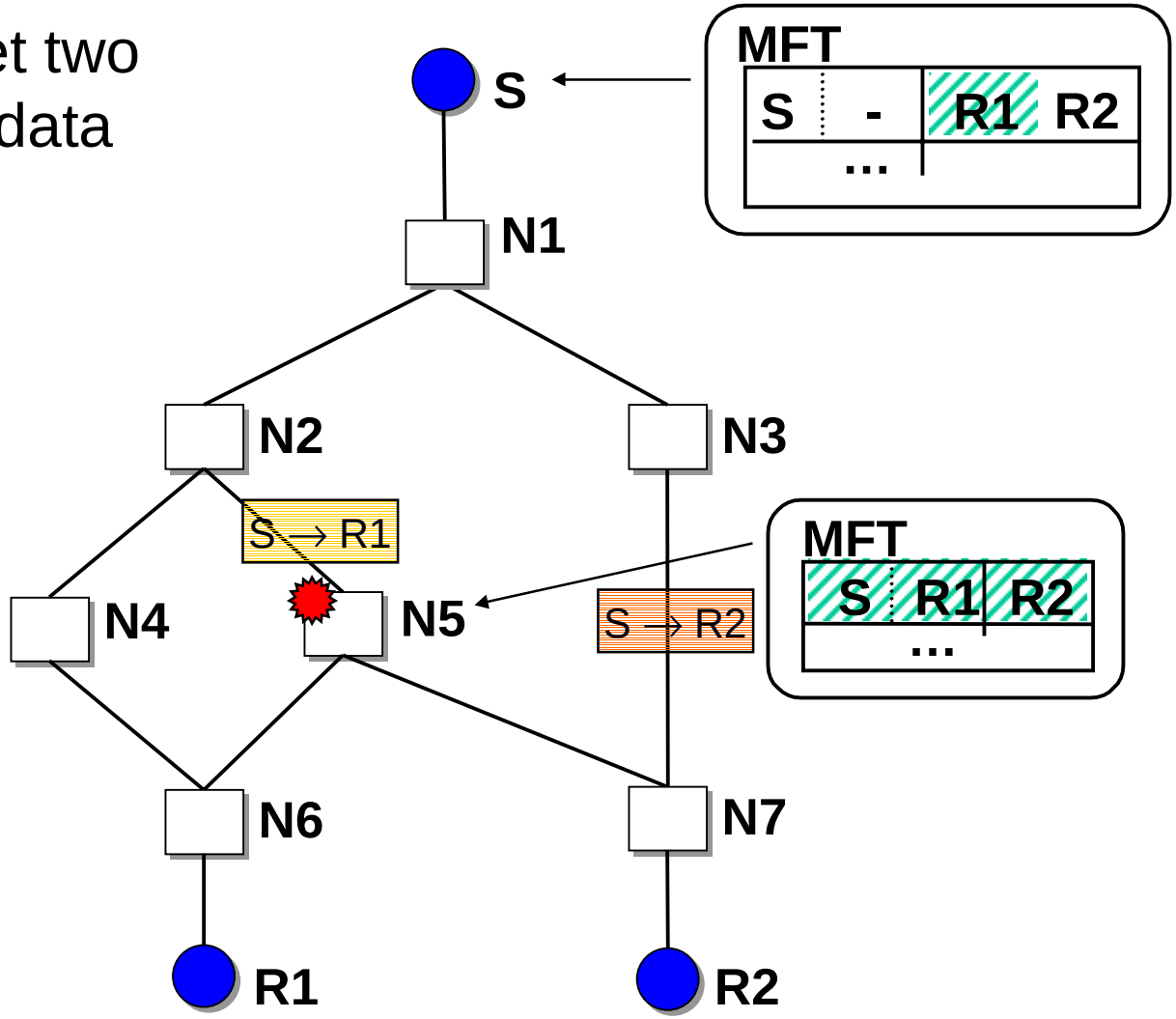
Leaving a Group

- R2 can get two copies of data packets



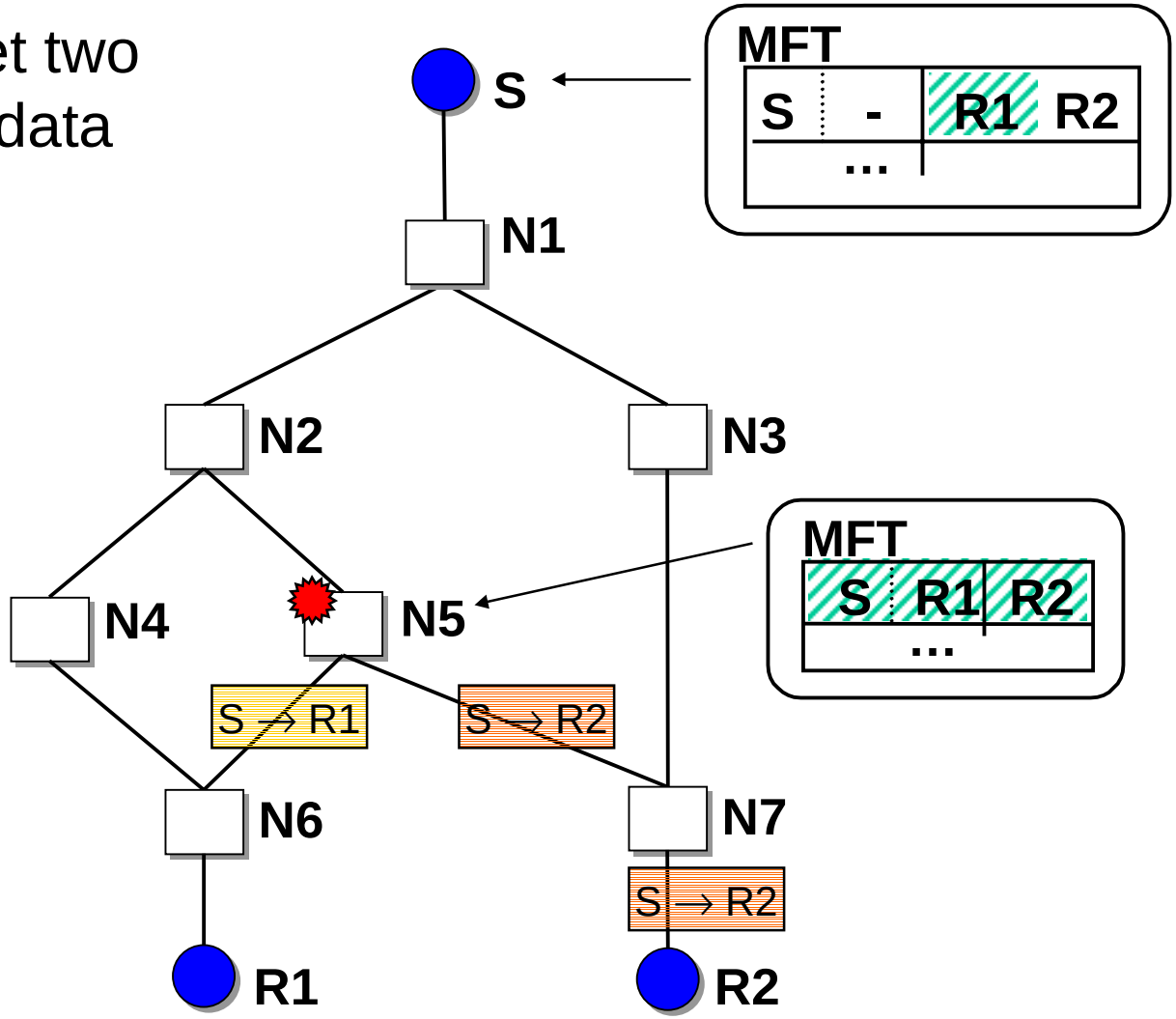
Leaving a Group

- R2 can get two copies of data packets



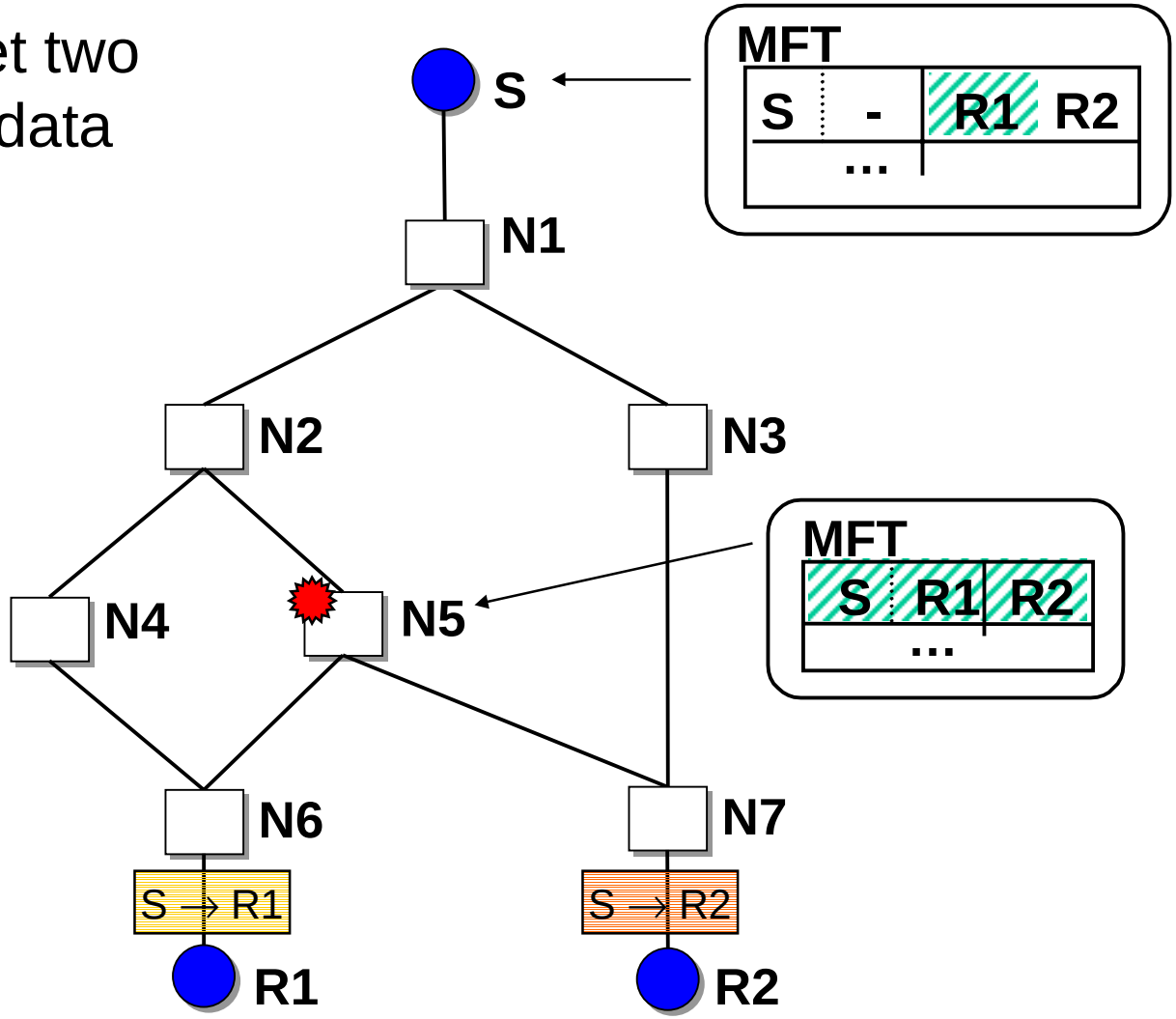
Leaving a Group

- R2 can get two copies of data packets



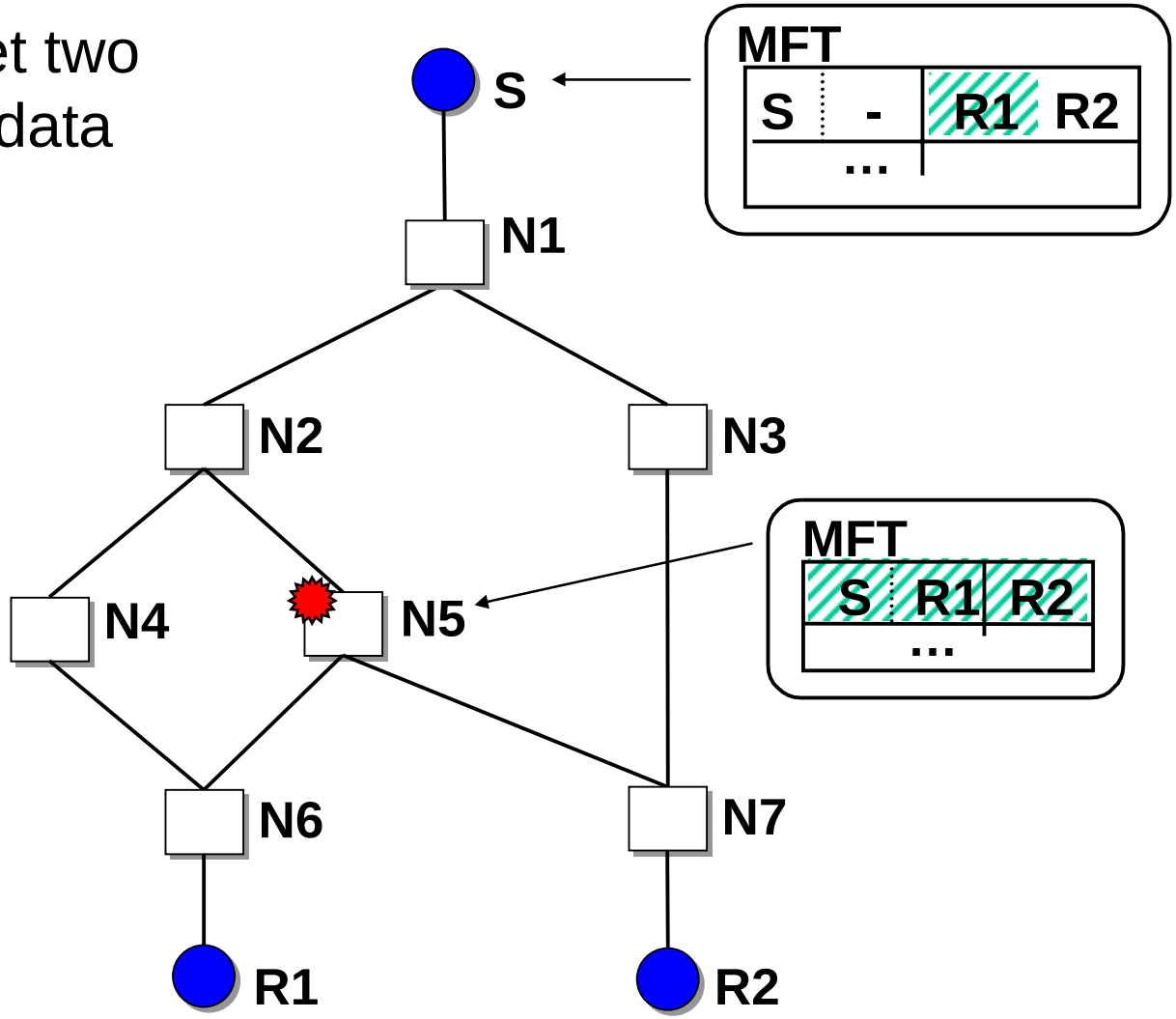
Leaving a Group

- R2 can get two copies of data packets



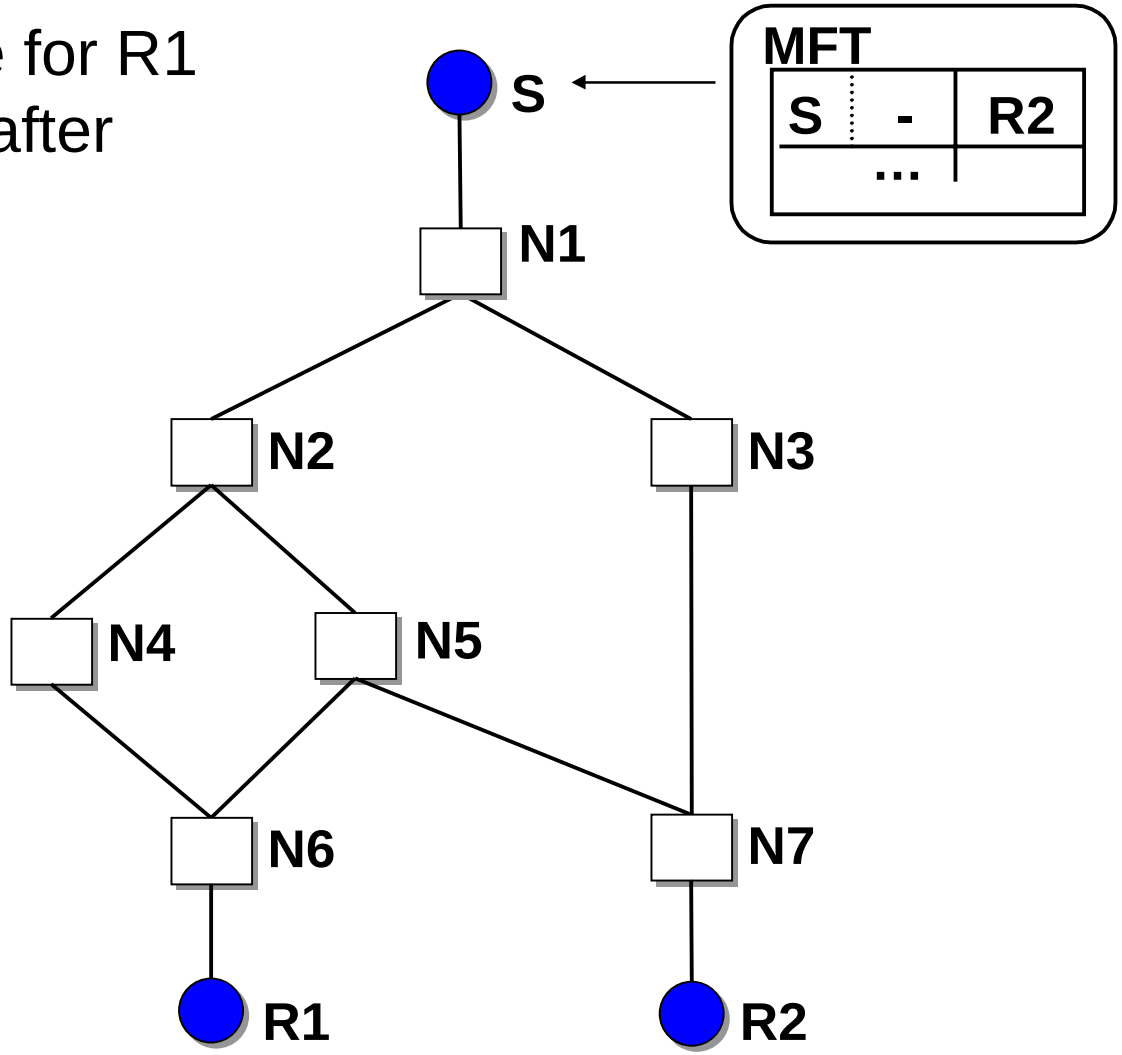
Leaving a Group

- R2 can get two copies of data packets



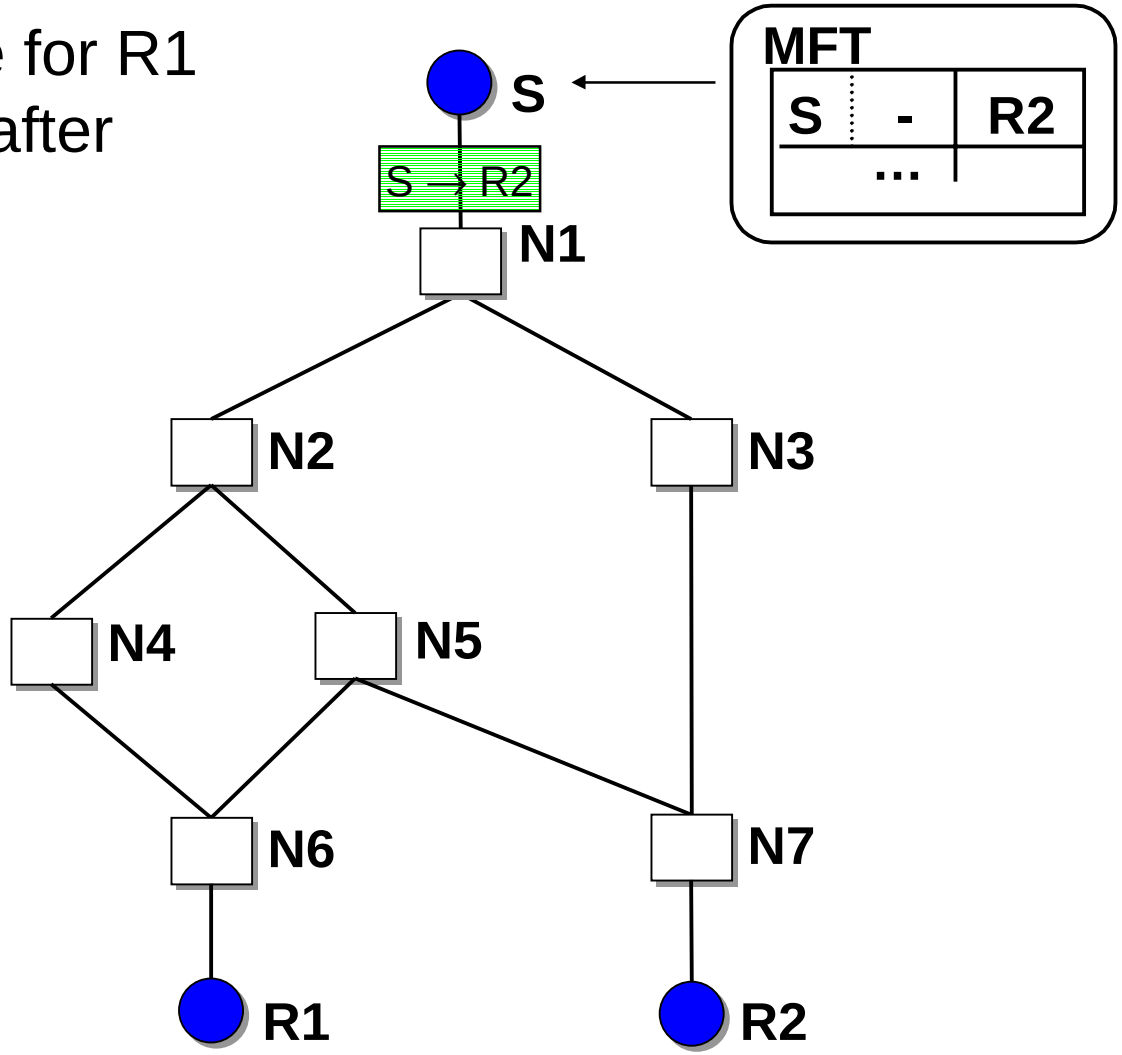
Leaving a Group

- MFT state for R1 removed after TO2



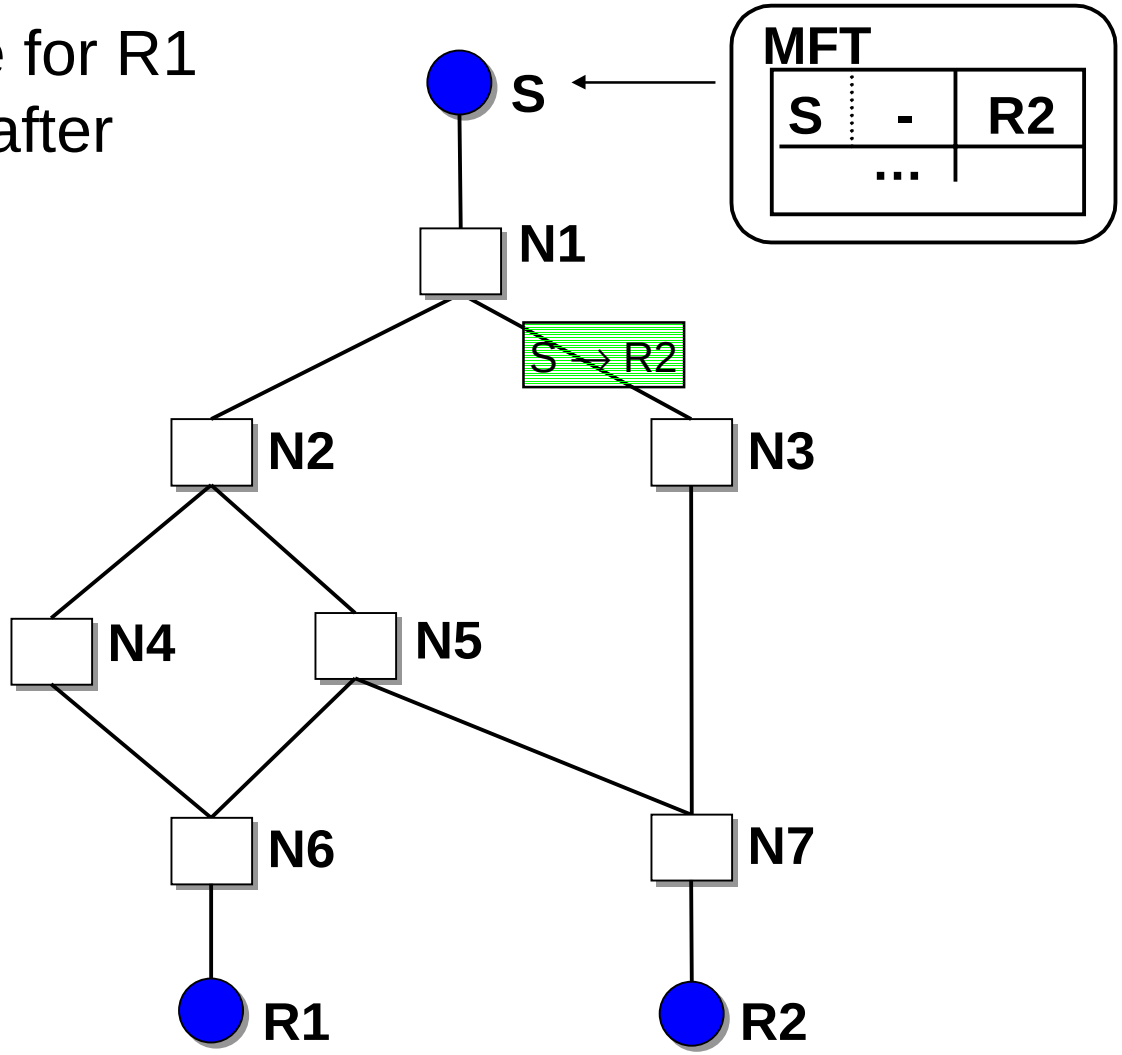
Leaving a Group

- MFT state for R1 removed after TO2



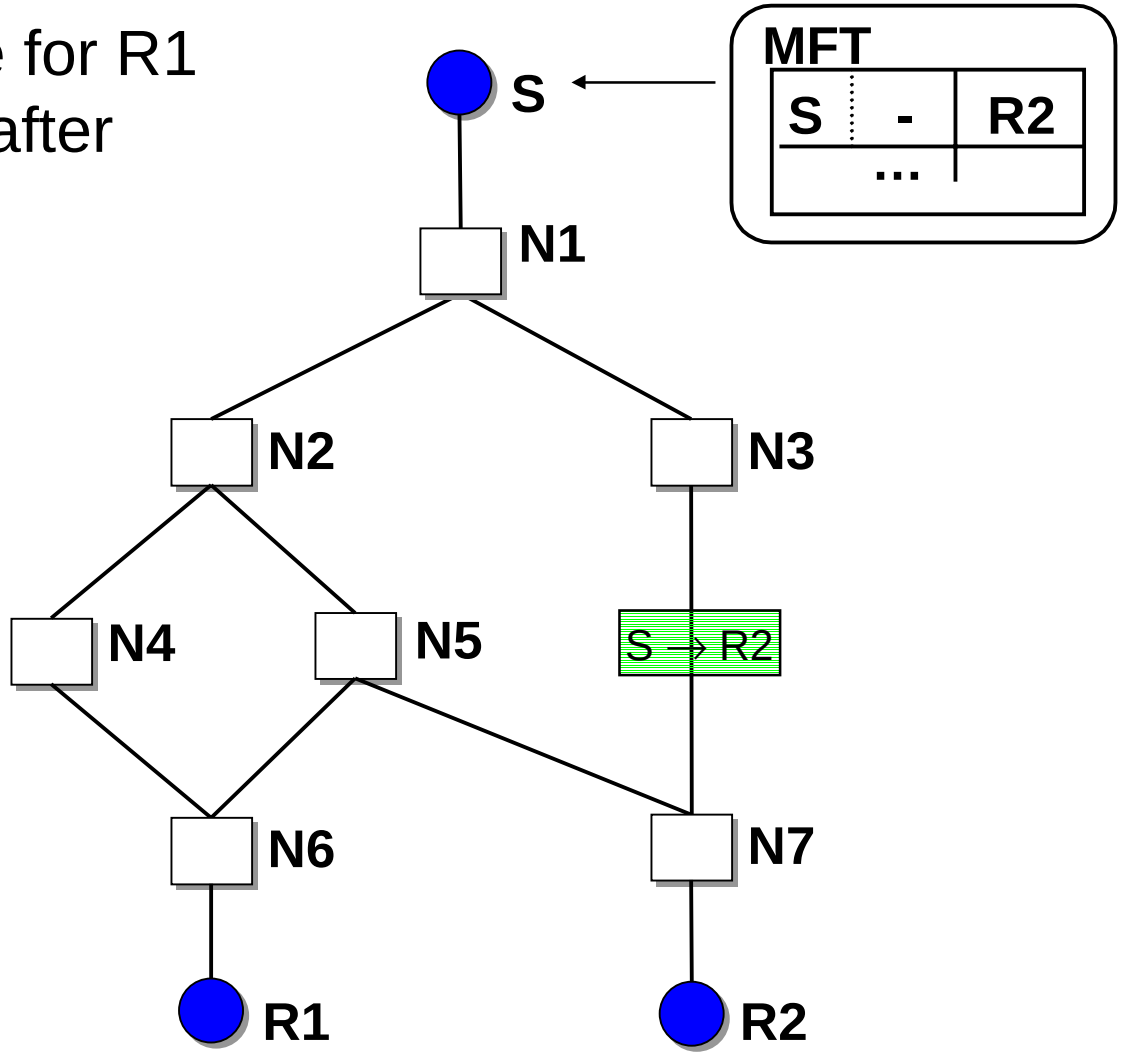
Leaving a Group

- MFT state for R1 removed after TO2



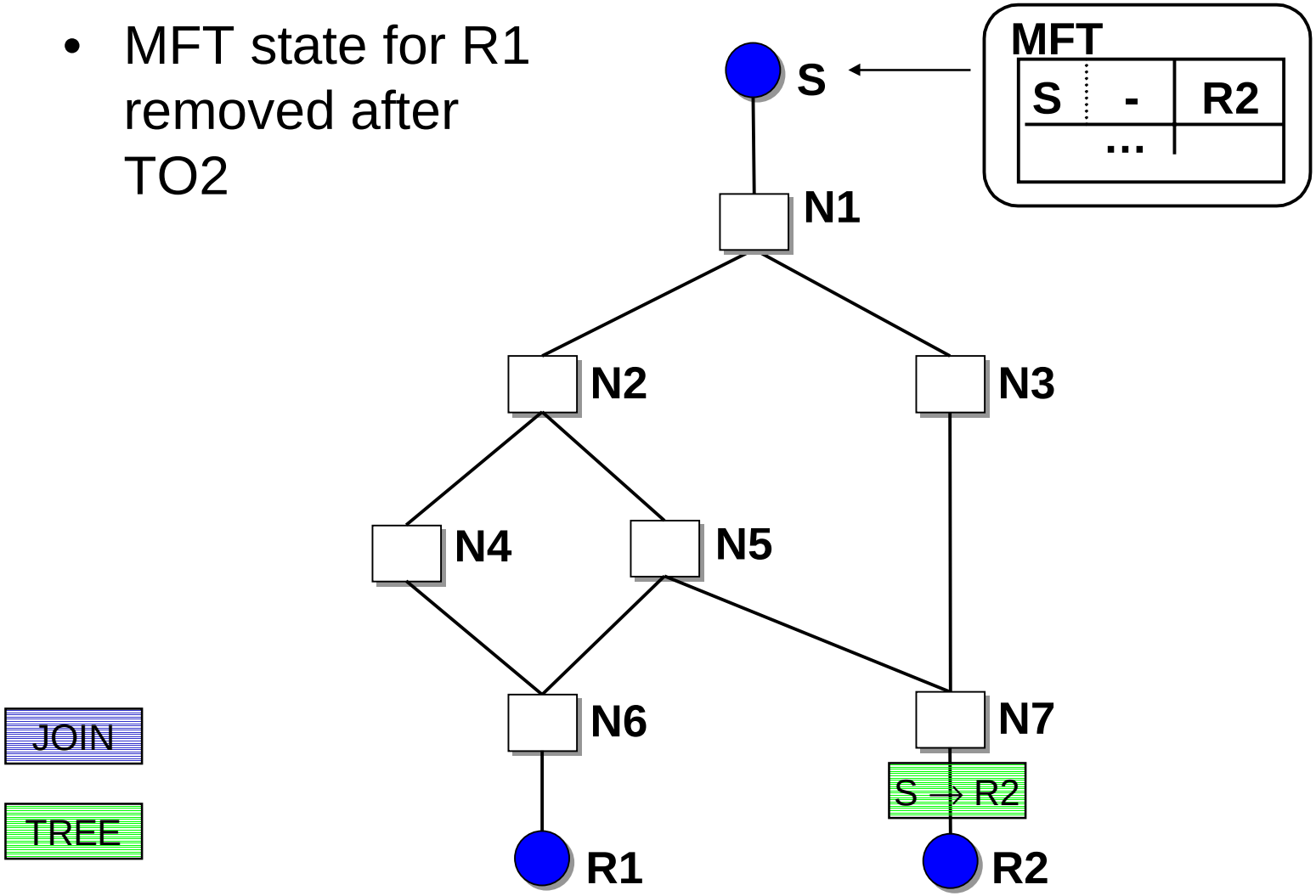
Leaving a Group

- MFT state for R1 removed after TO2



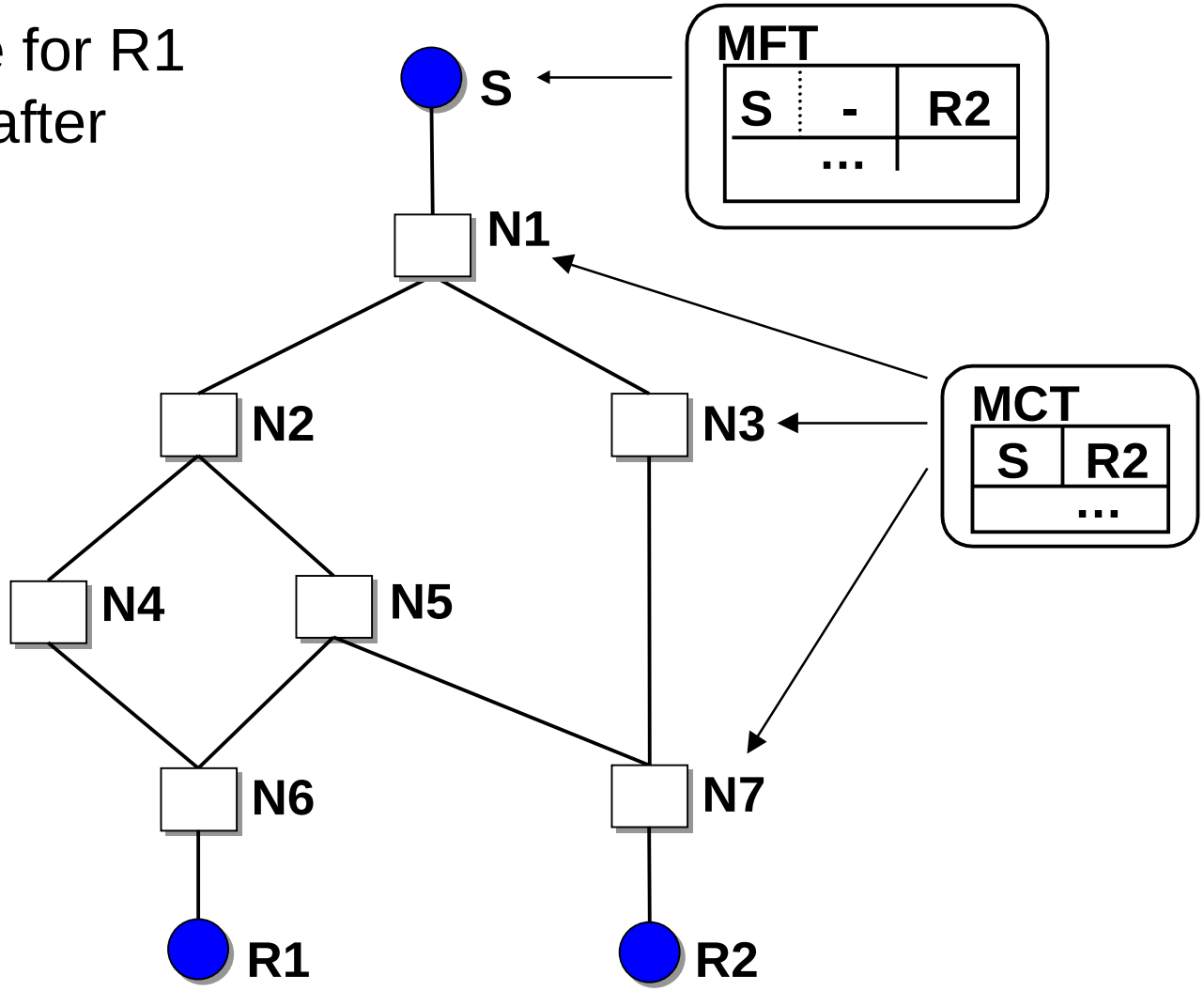
Leaving a Group

- MFT state for R1 removed after TO2



Leaving a Group

- MFT state for R1 removed after TO2



JOIN

TREE

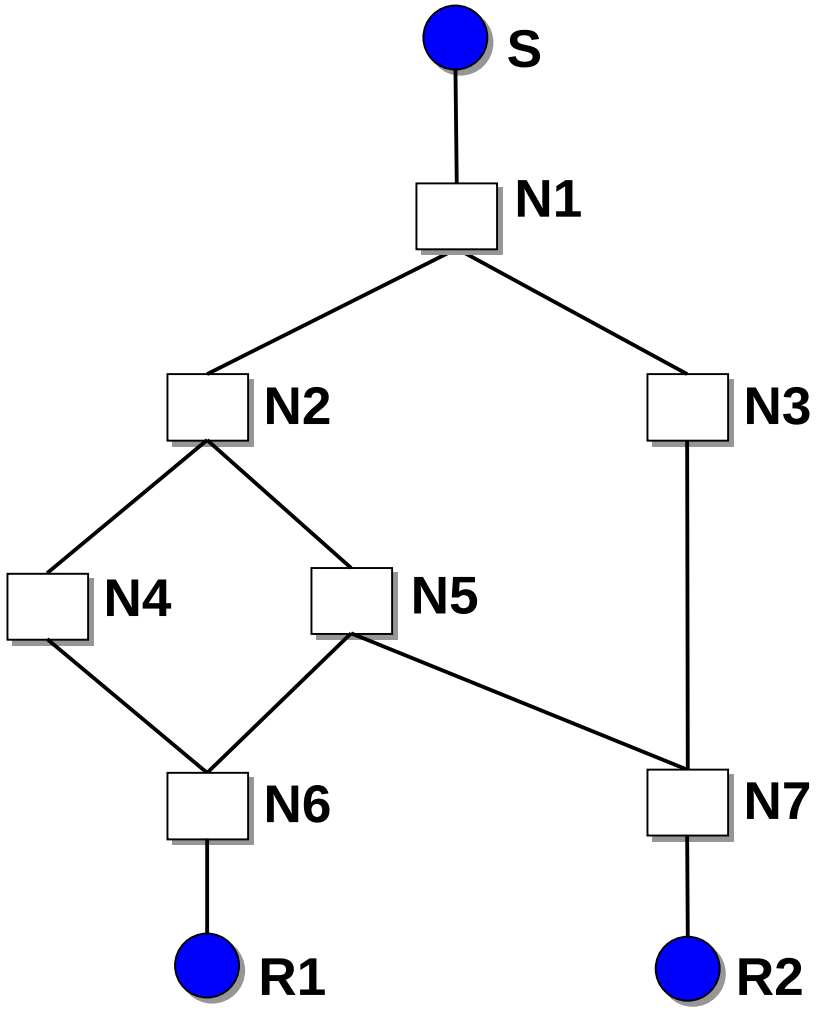
Advantages

- Only branching routers maintain forwarding state
- Scalability is enhanced

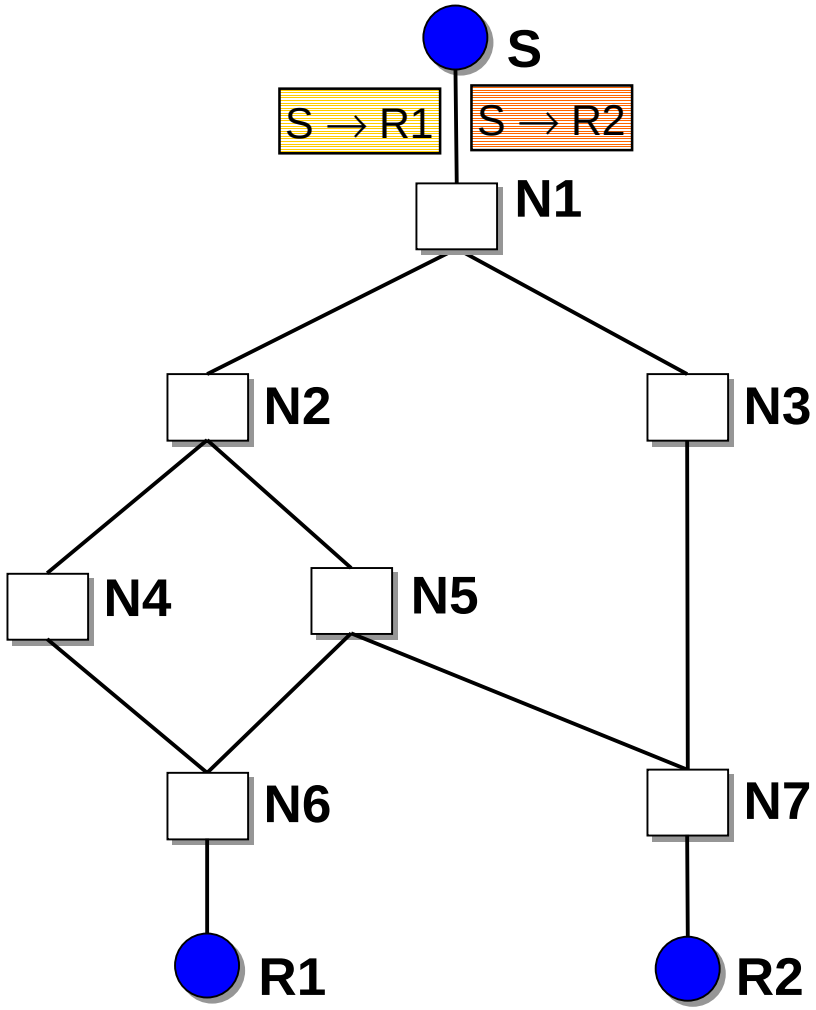
Advantages

- Native support for incremental deployment
  - IP multicast needs IP encapsulation or tunneling
- Load balancing and graceful degradation

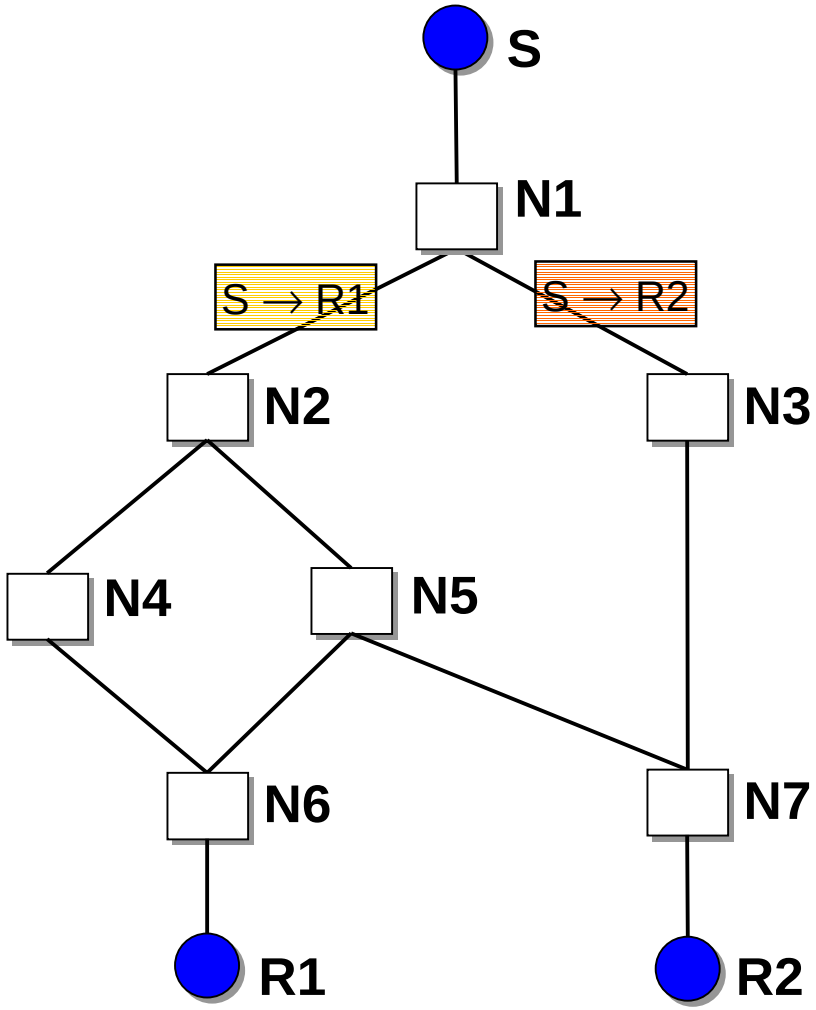
Incremental Deployment



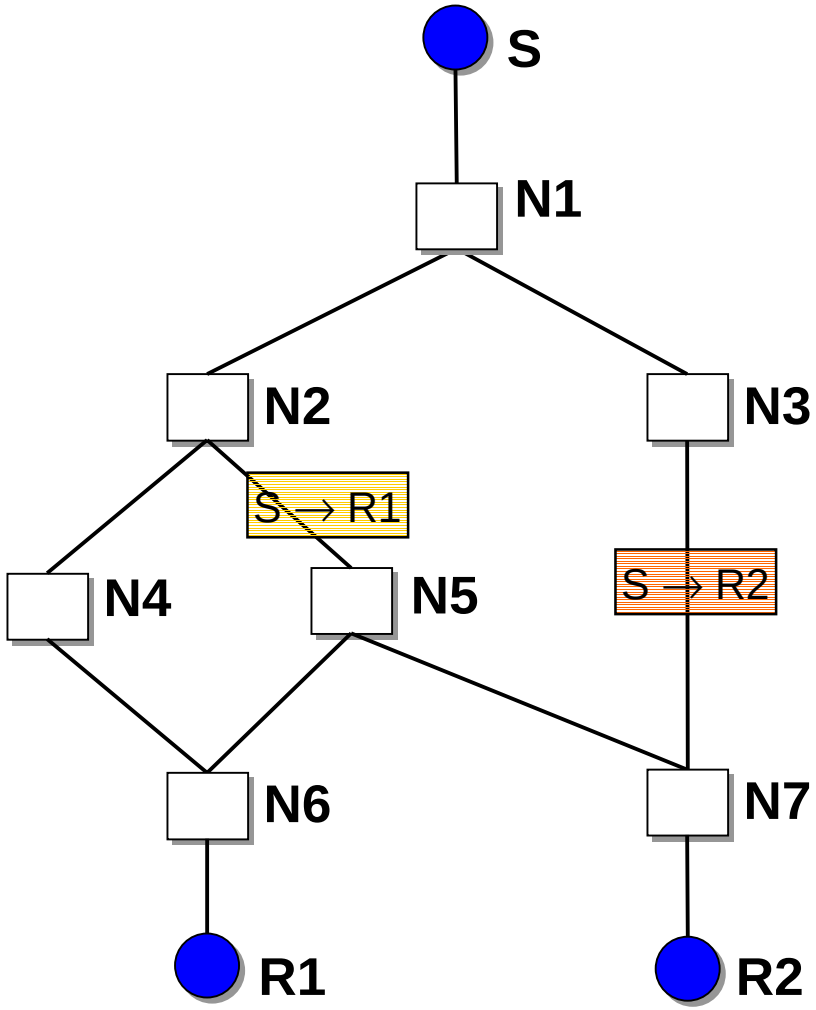
Incremental Deployment



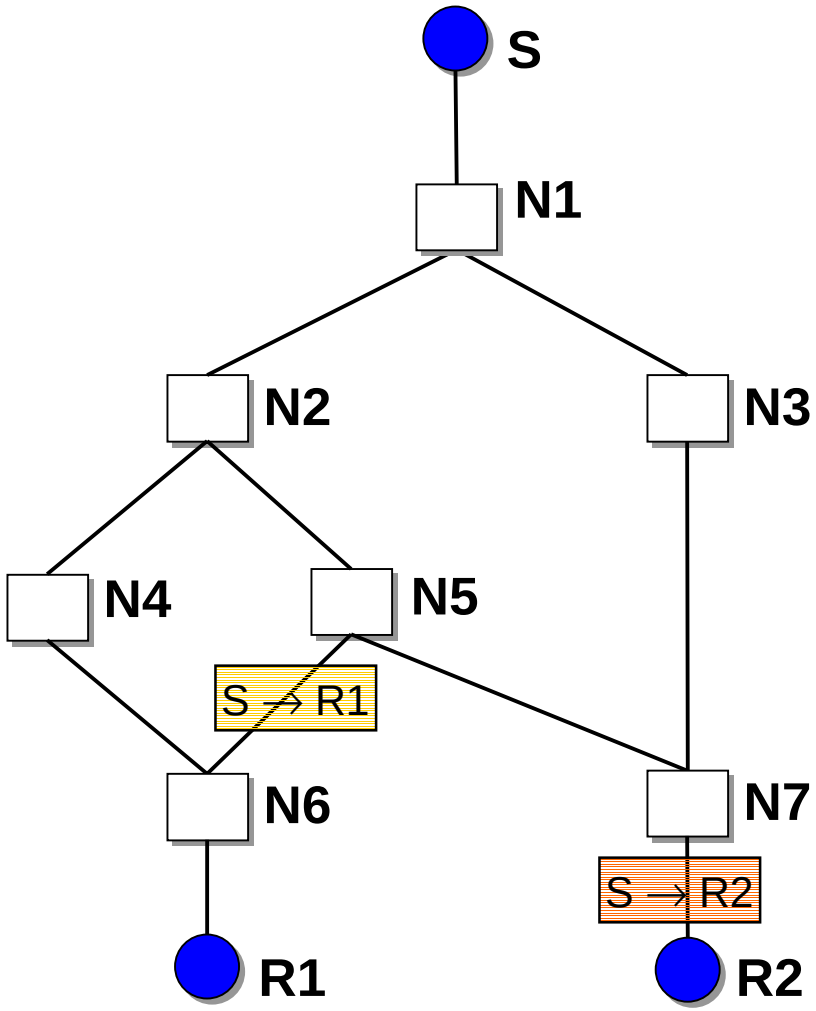
Incremental Deployment



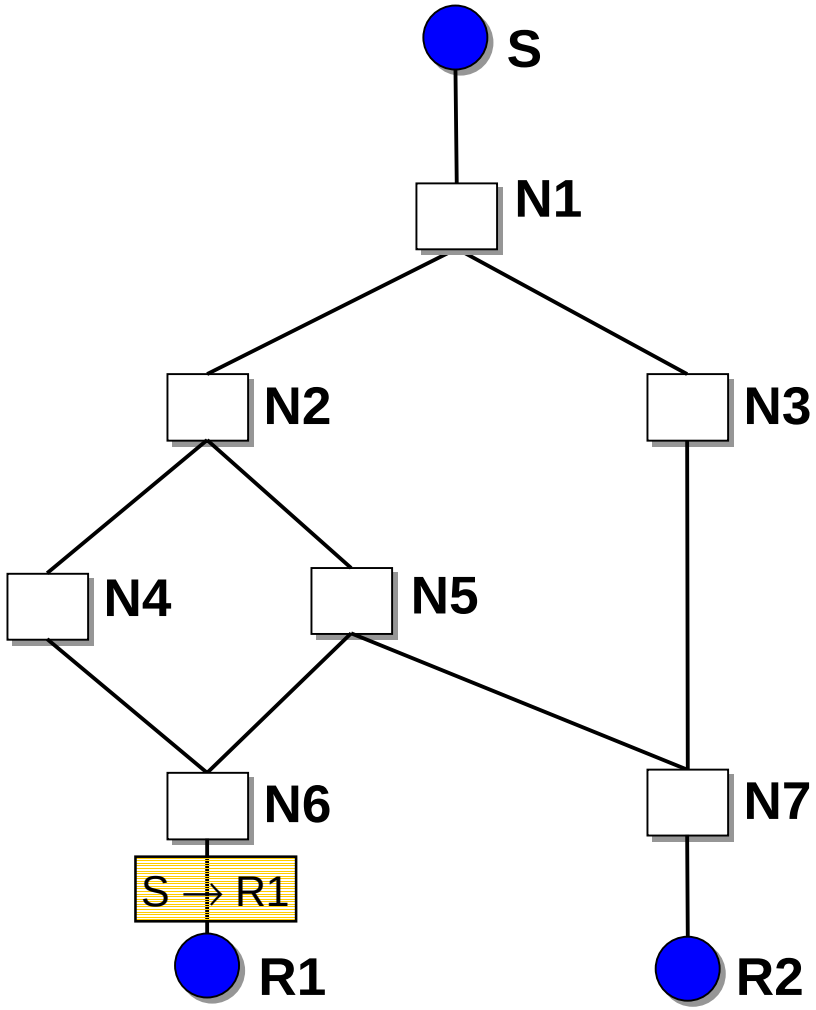
Incremental Deployment



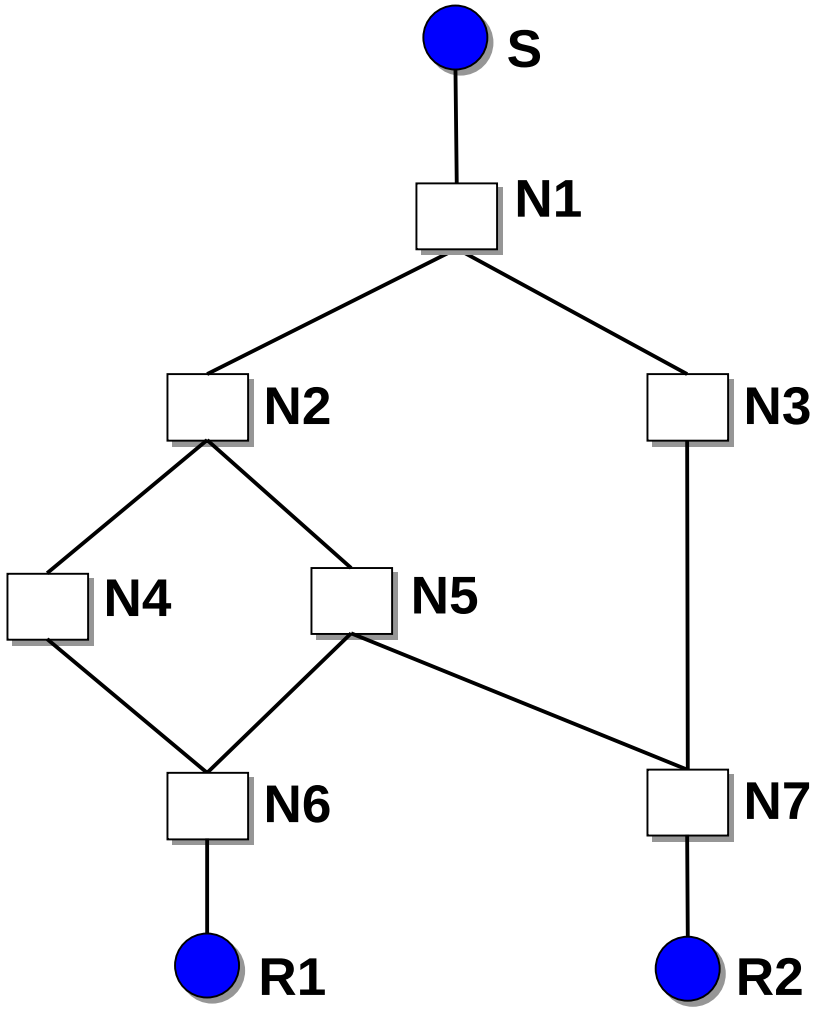
Incremental Deployment



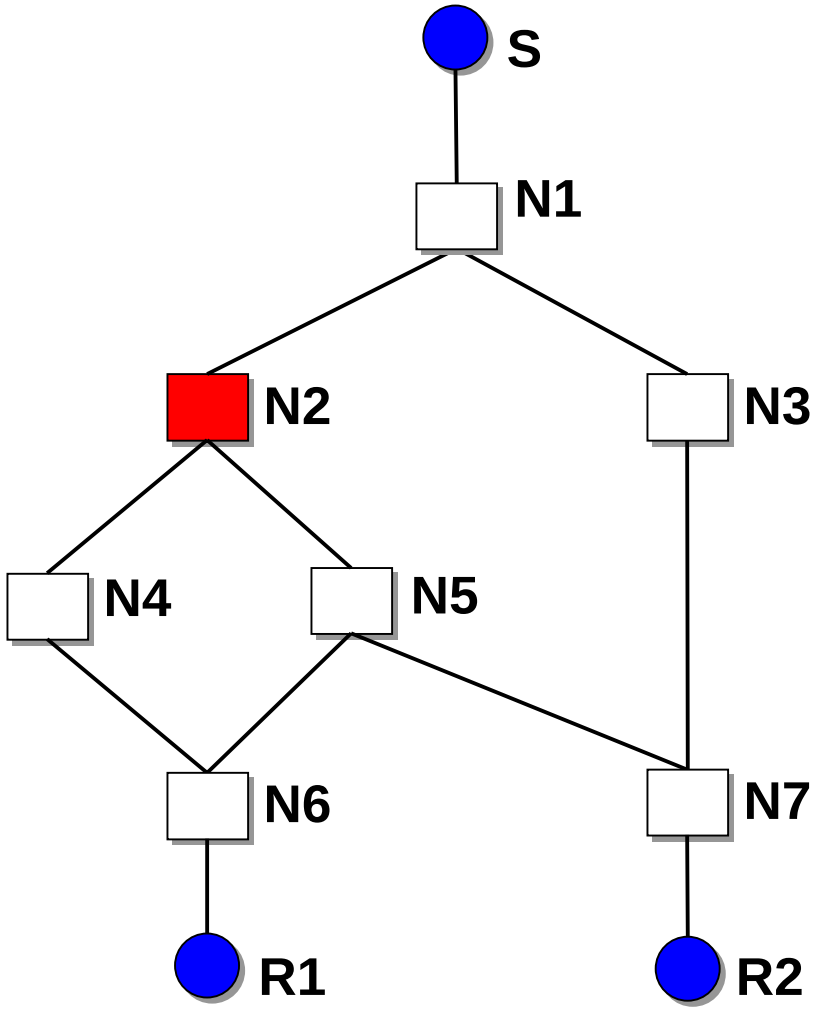
Incremental Deployment



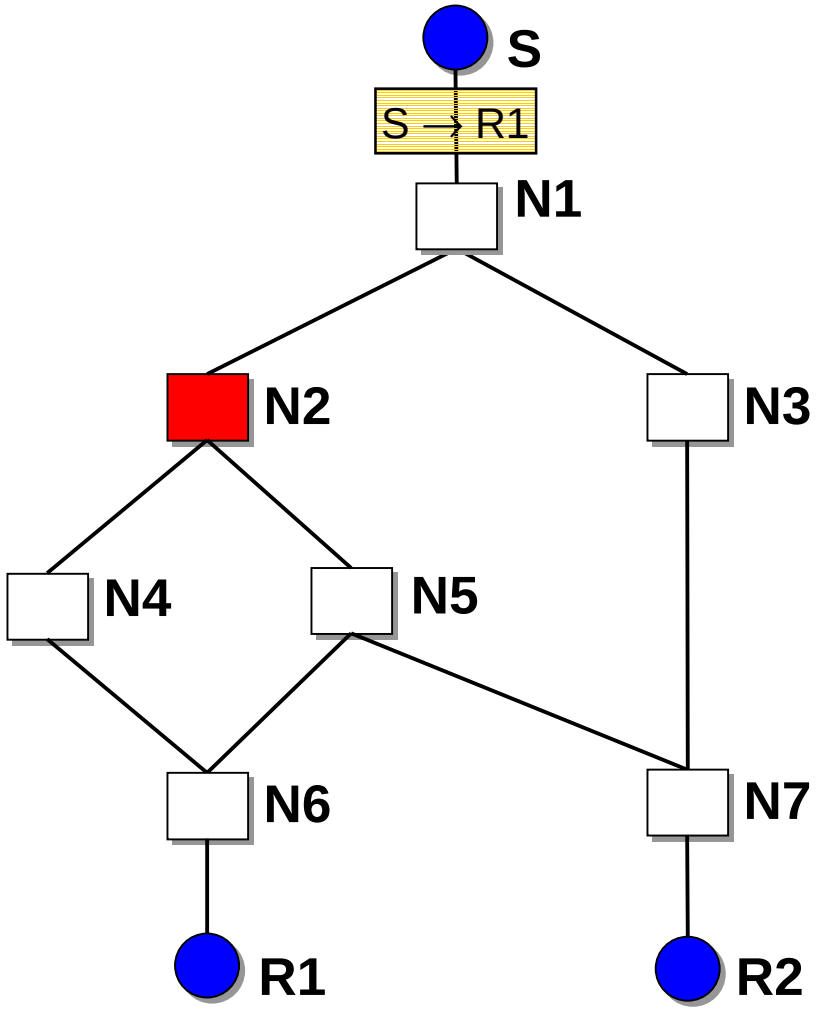
Incremental Deployment



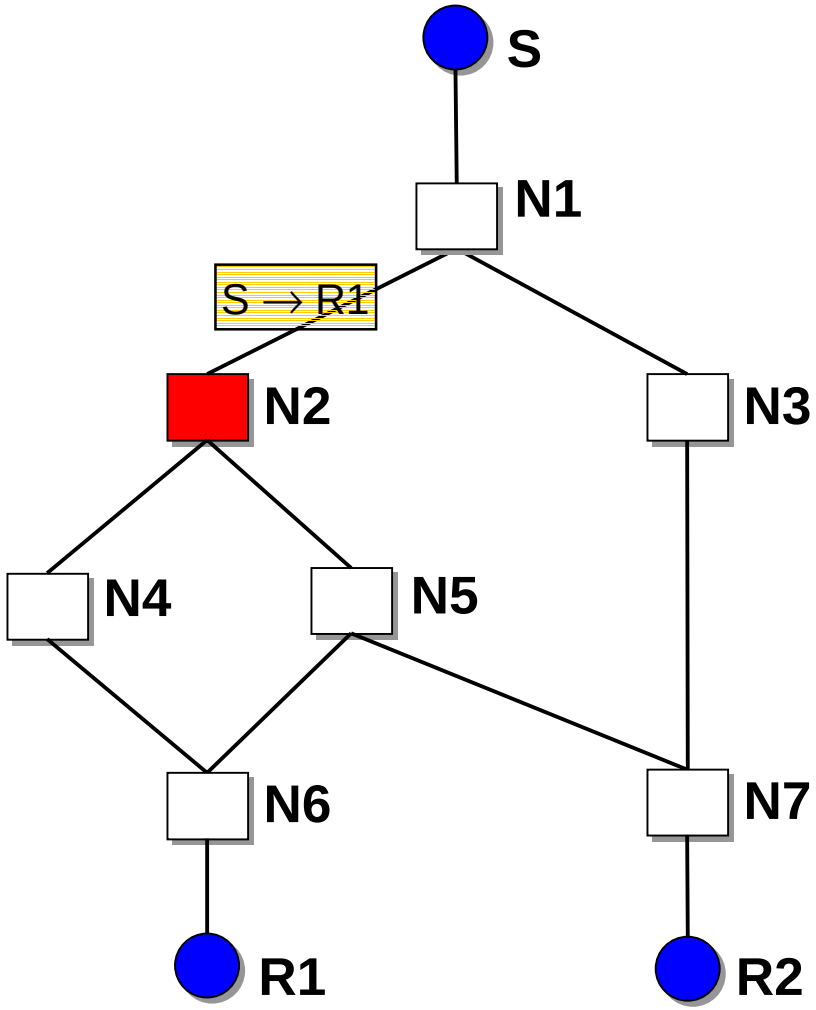
Incremental Deployment



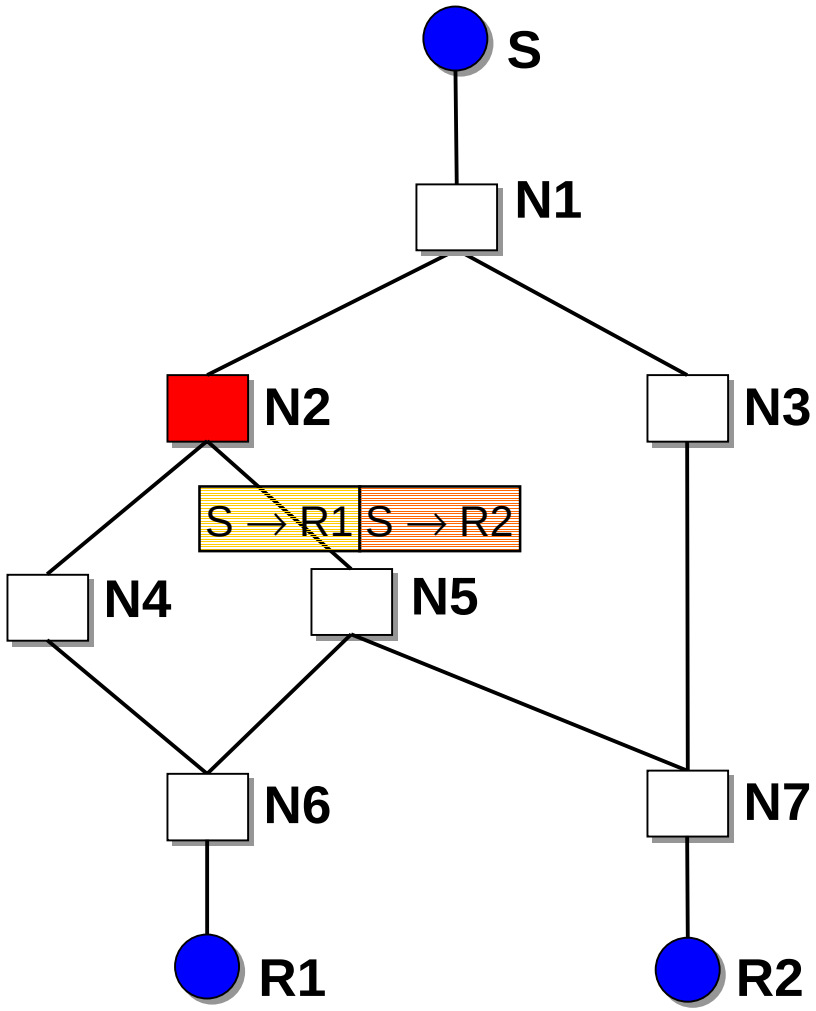
Incremental Deployment



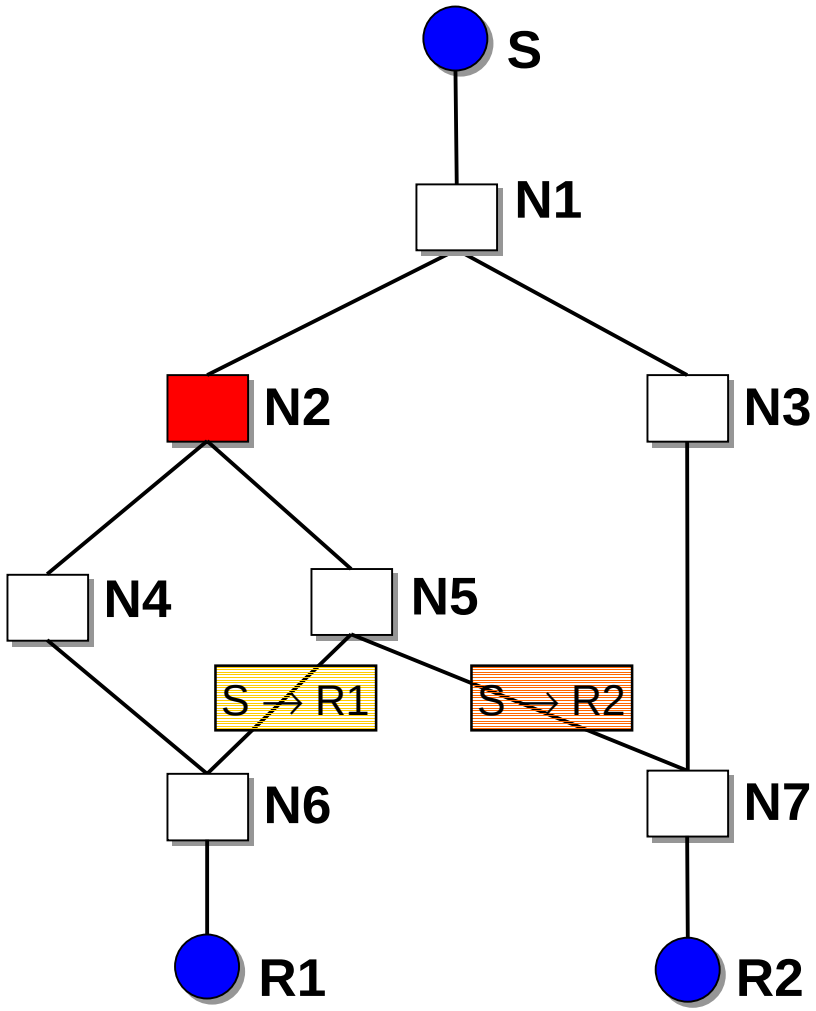
Incremental Deployment



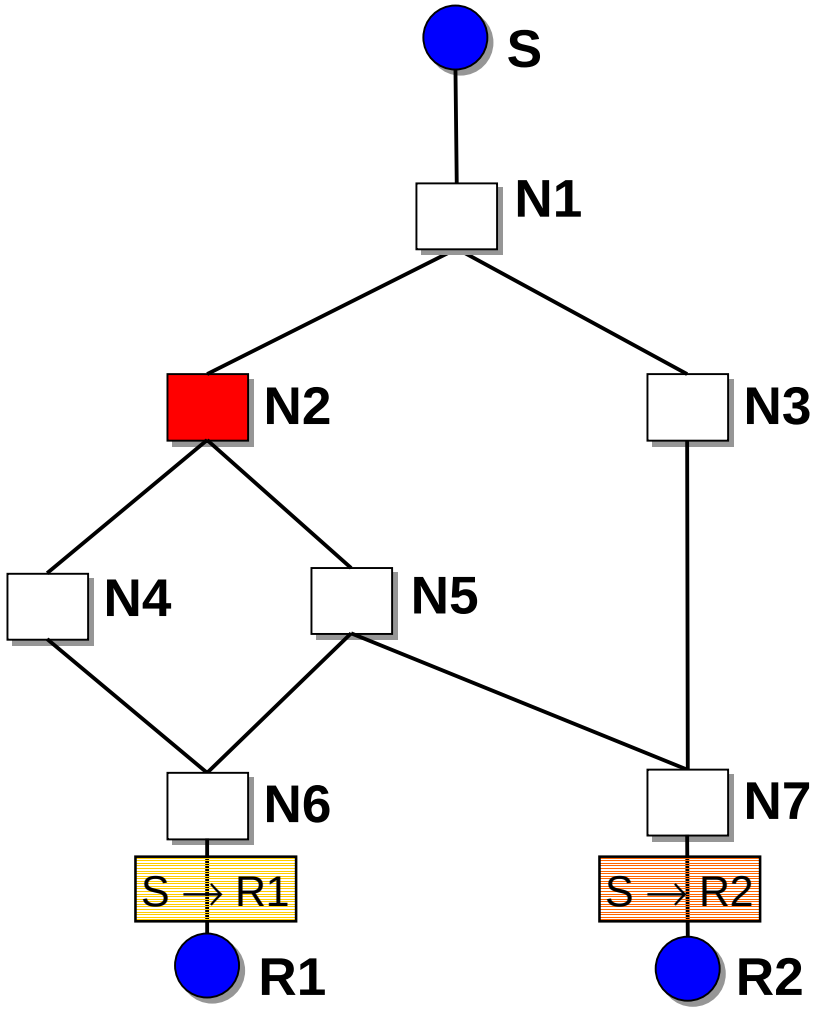
Incremental Deployment



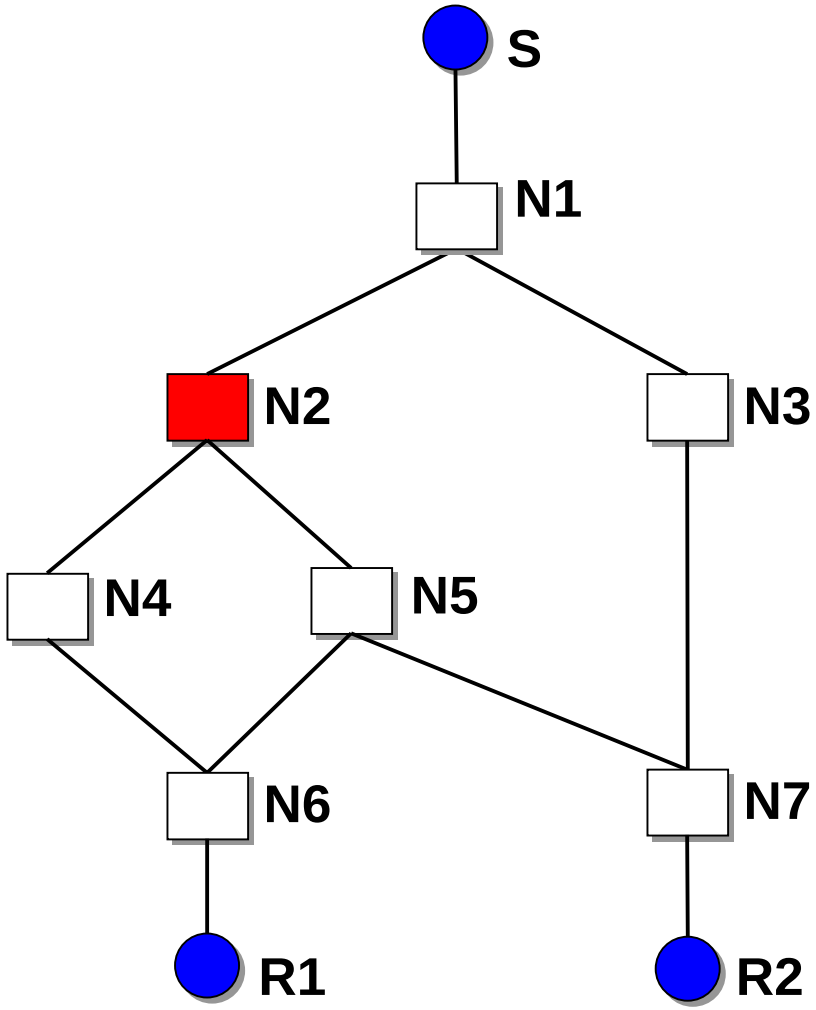
Incremental Deployment



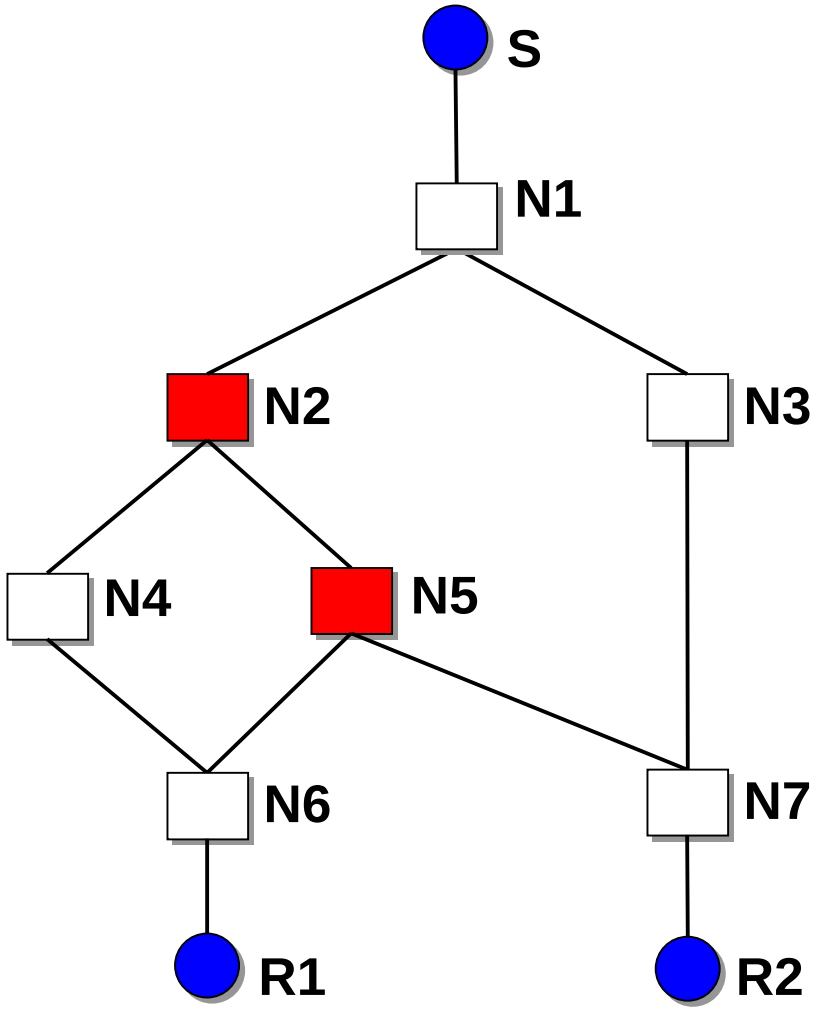
Incremental Deployment



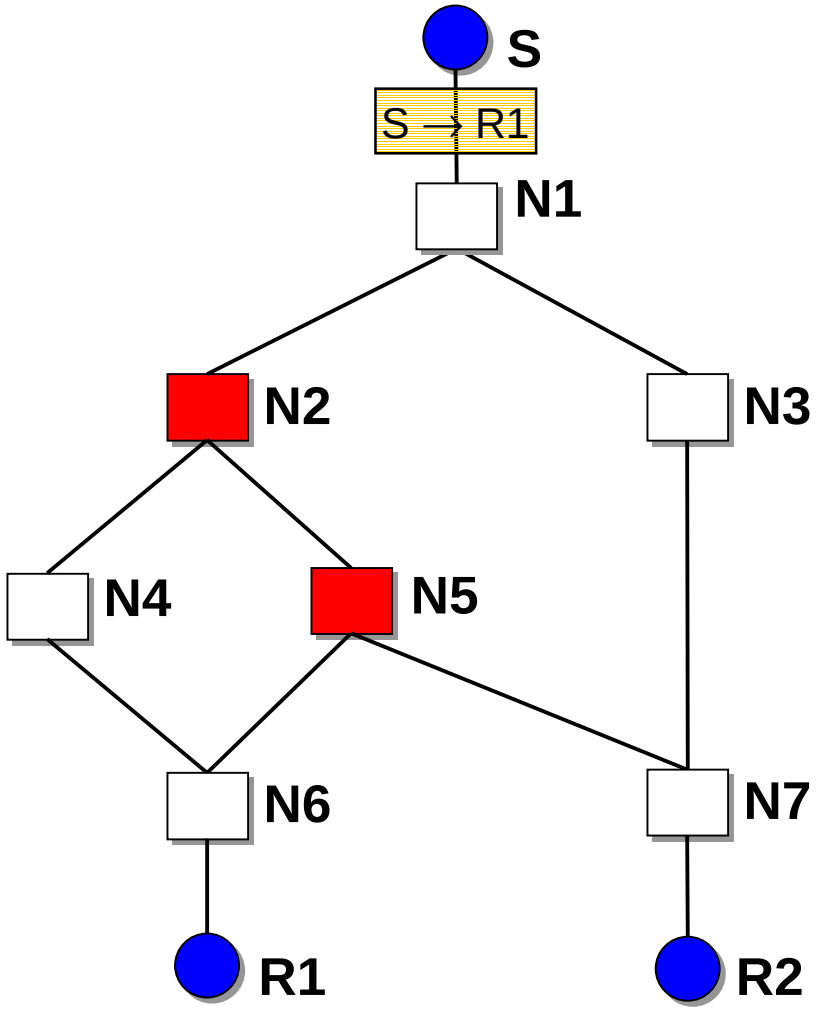
Incremental Deployment



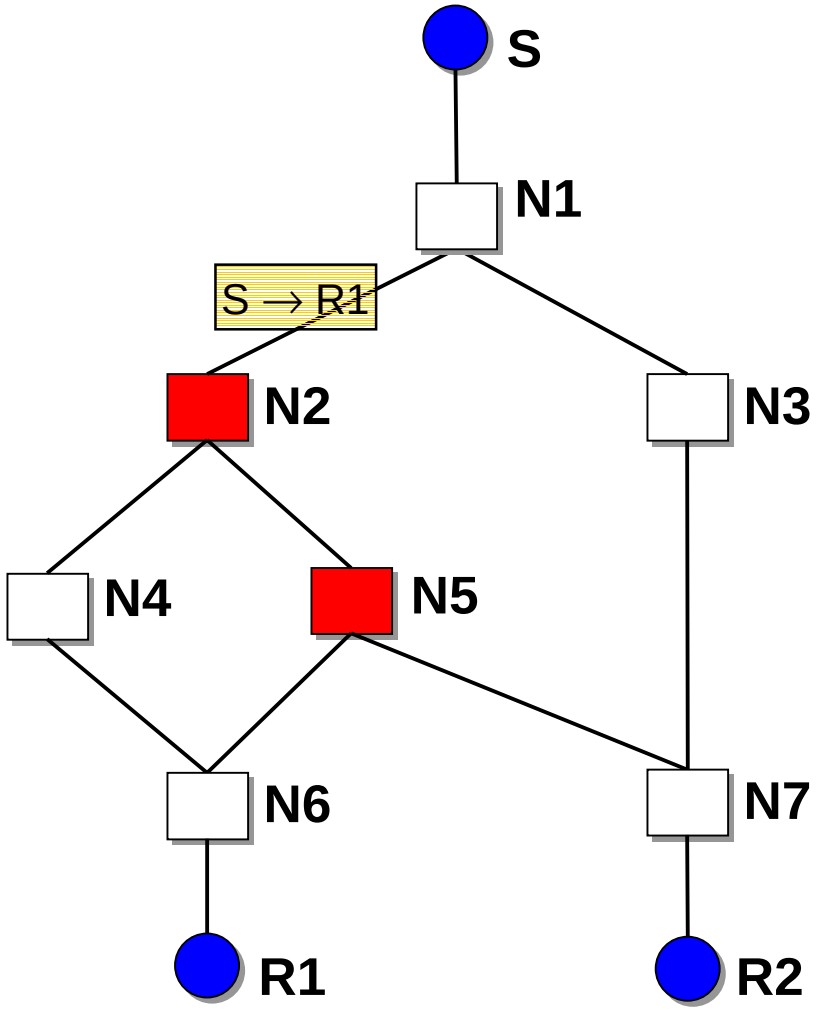
Incremental Deployment



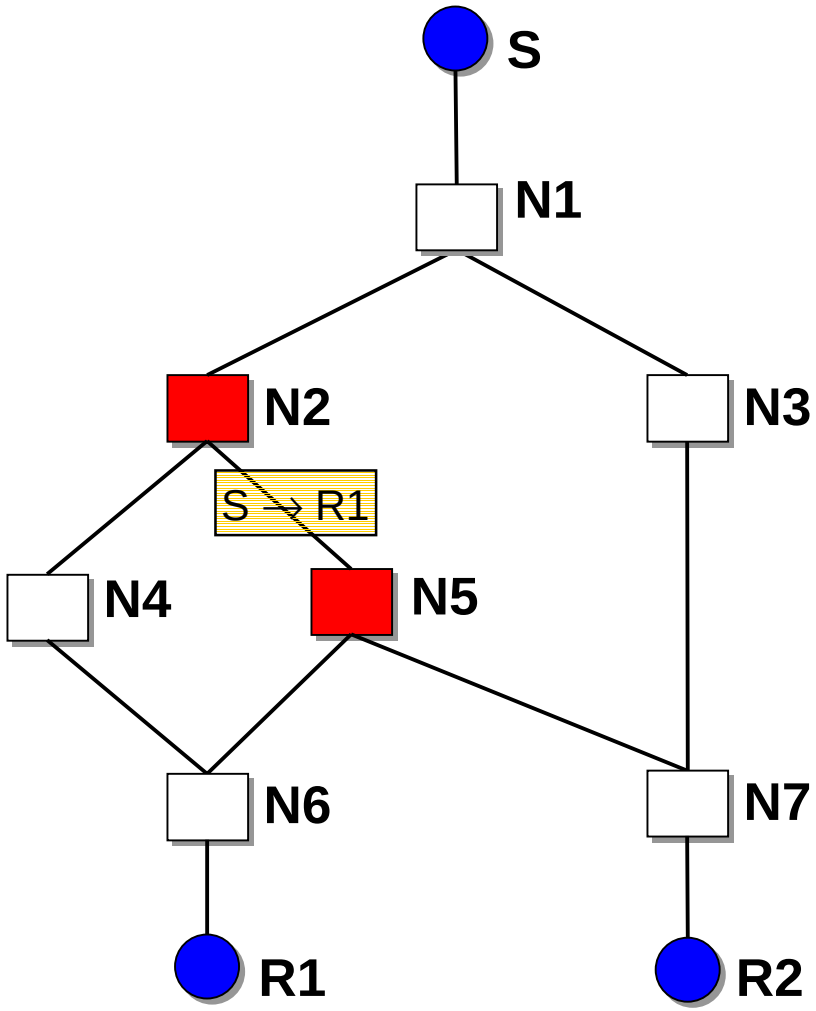
Incremental Deployment



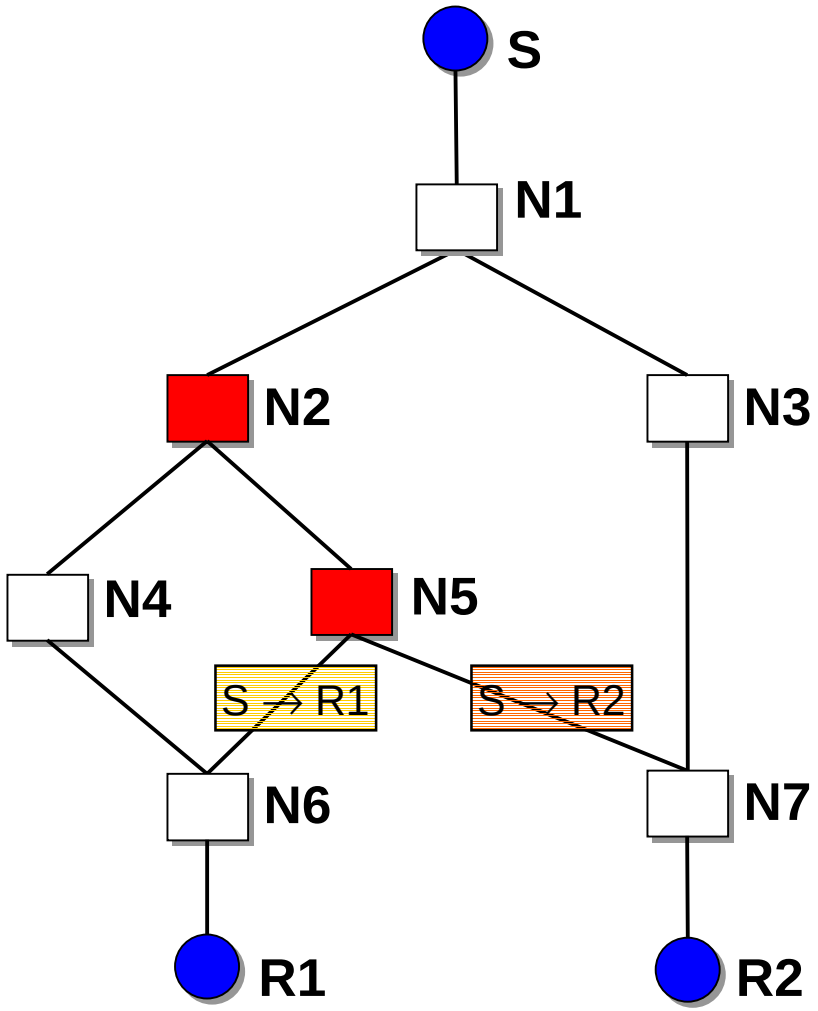
Incremental Deployment



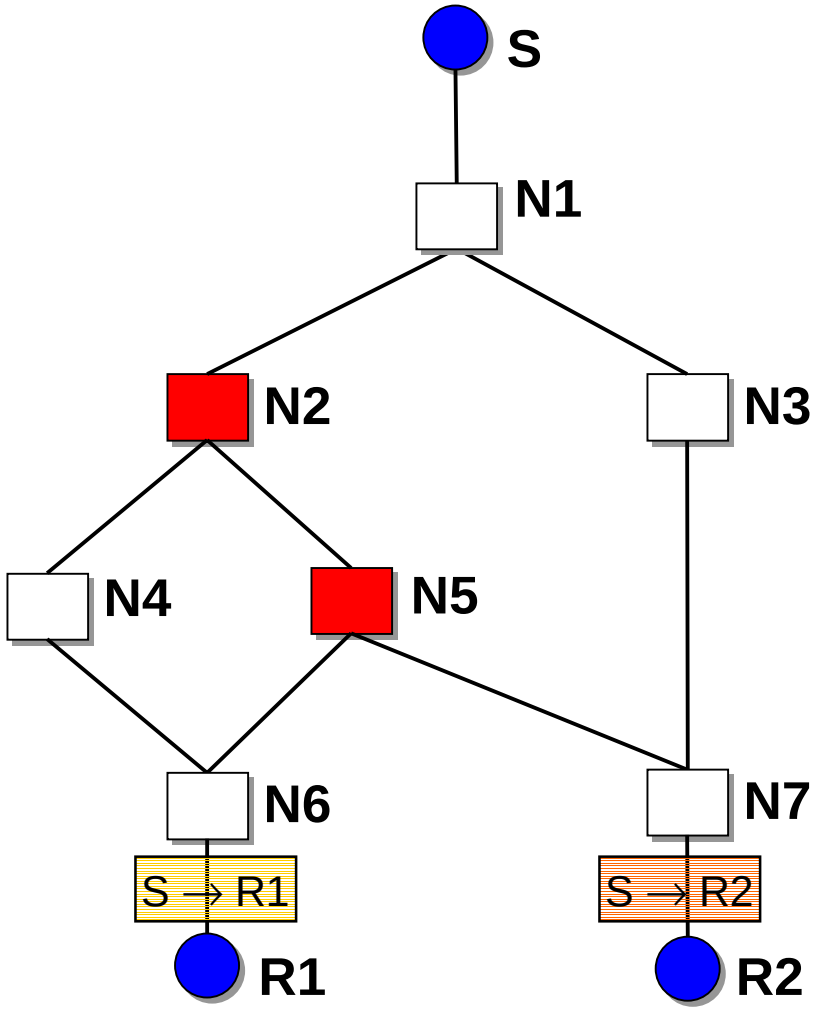
Incremental Deployment



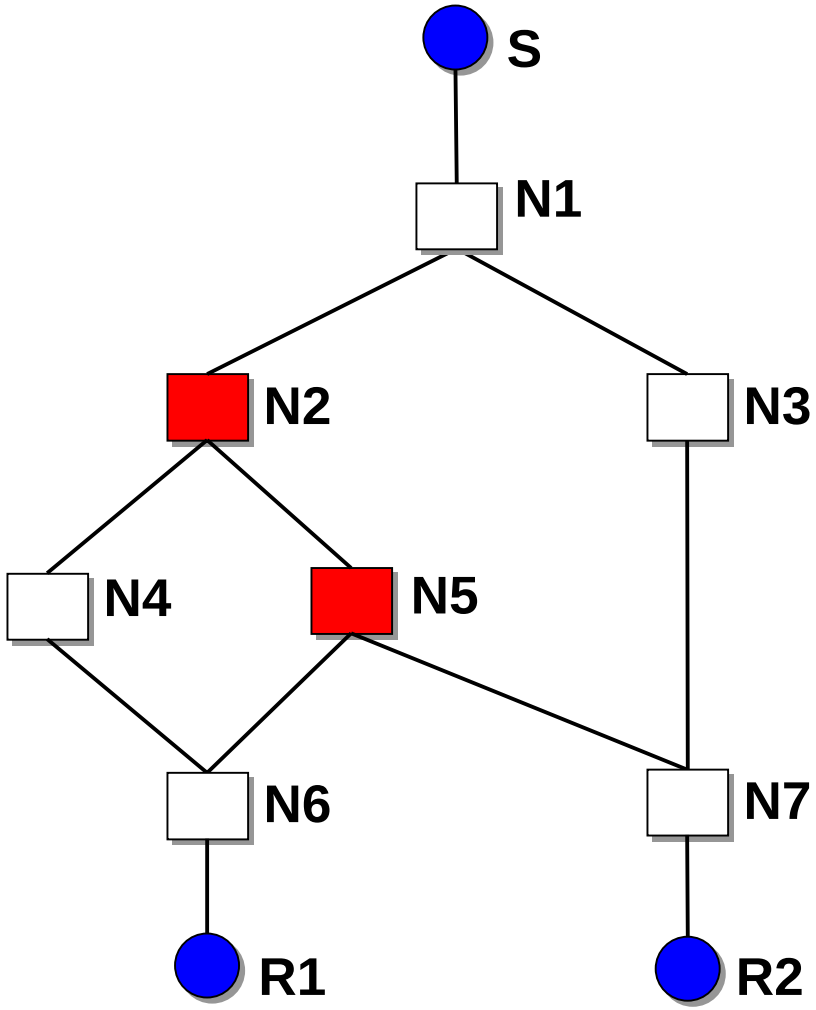
Incremental Deployment



Incremental Deployment

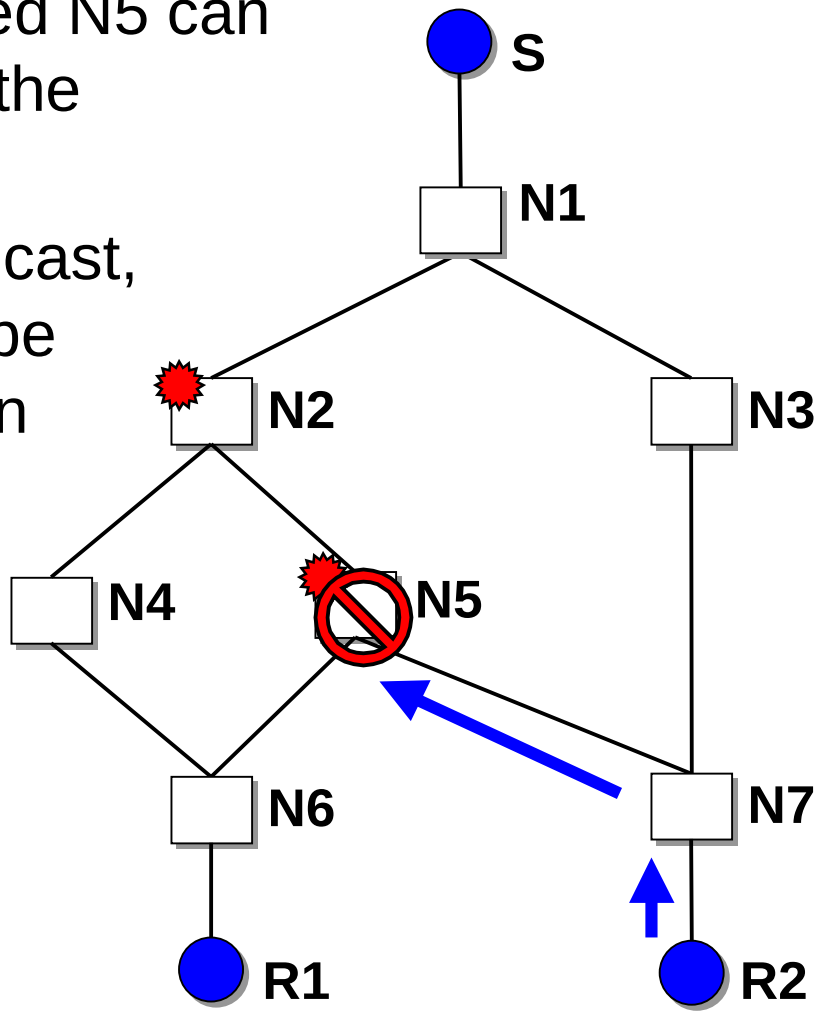


Incremental Deployment



Load Balancing & Graceful Degradation

- Overloaded N5 can let N2 do the branching
- In IP multicast, R2 won't be able to join



Advantages

- No address allocation problem
- Sender access control can be implemented at the root
  - root can authenticate senders
  - less vulnerable to denial of service attack

Related Work

- Dynamic tunneling [Tian & Neufeld, 1998]
- Simple [Perlman et al, 1999] &
  Express [Holbrook & Cheriton, 1999]
  - augment a multicast address with a unicast address
  - eliminate global address allocation problem
  - better access control

Conclusions

- Use *unicast* addresses for group identification and packet forwarding
- Use *recursive unicast* to implement multicast service
- REUNITE has many unique advantages
  - improved scalability
  - incrementally deployable
  - load balancing & graceful degradation
- Technical Report (with REUNITE II) available at

http://www.cs.cmu.edu/~eugeneng/research/reunite/

Conclusions

- Use *unicast* addresses for group identification and packet forwarding
- Use *recursive unicast* to implement multicast service
- REUNITE has many unique advantages
  - improved scalability
  - incrementally deployable
  - load balancing & graceful degradation
- Technical Report (with REUNITE II) available at

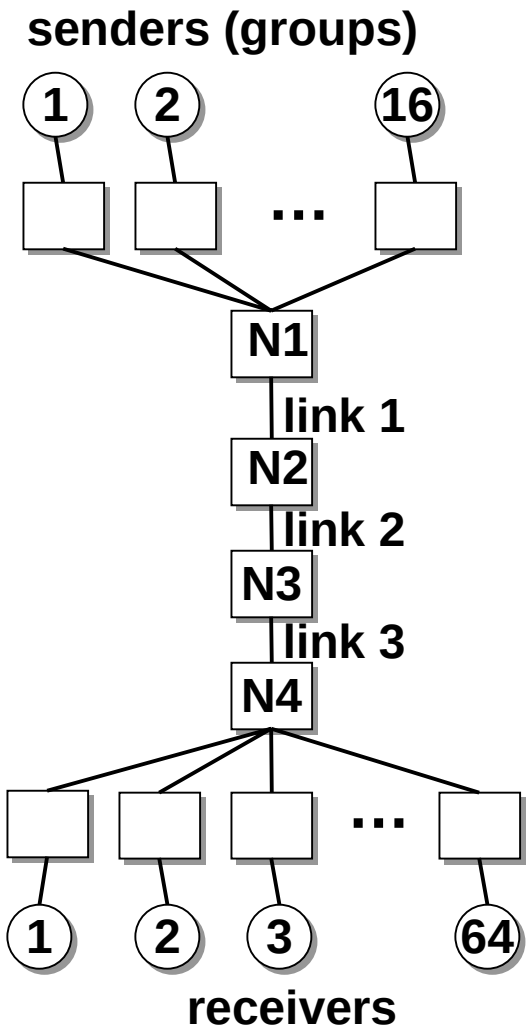
http://www.cs.cmu.edu/~eugeneng/research/reunite/

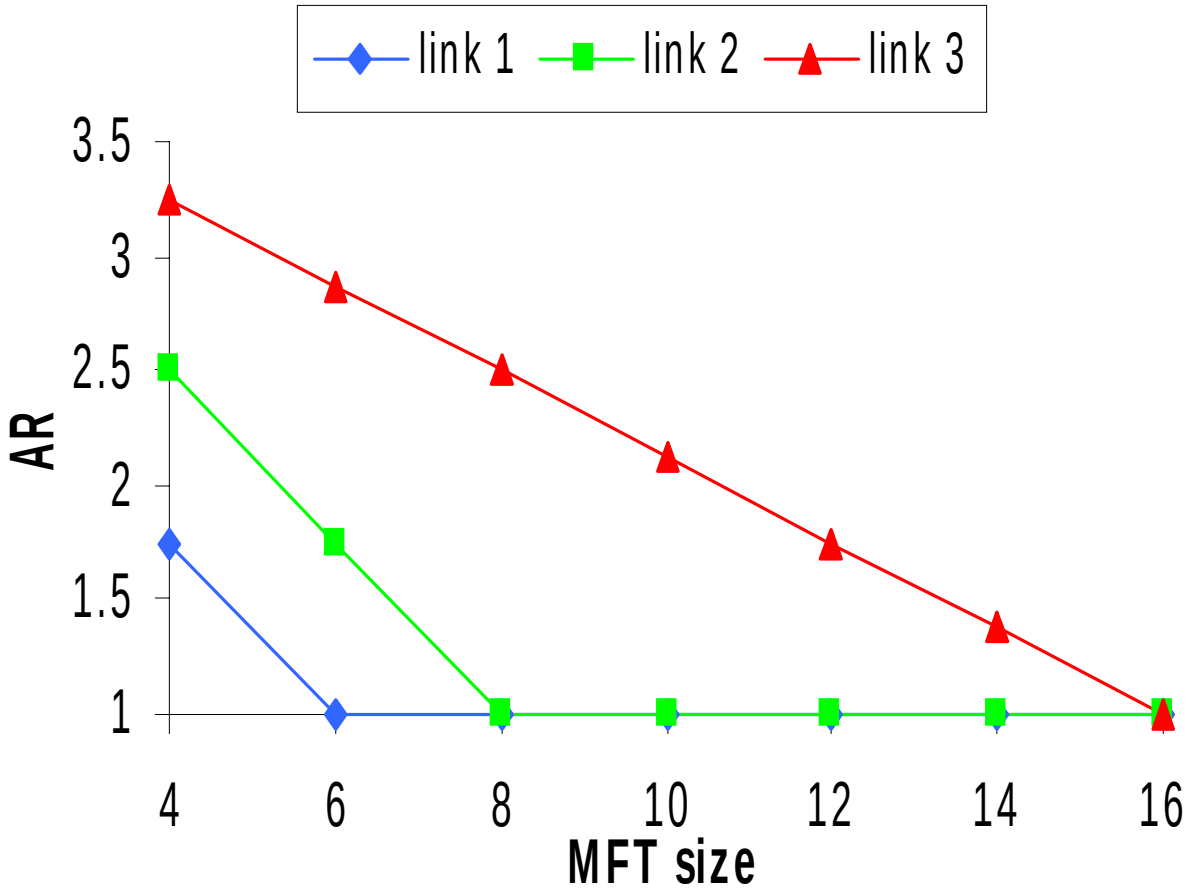
REcursive UNIcast TreE

Simulations

- Implemented in ns-2
- Total simulation time: 60 seconds
  - all receivers join in the first 10 seconds
  - results reported for the last 50 seconds
  - TO1 = TO2 = 5 seconds
  - JOIN refresh period = 2.5 seconds
  - TREE piggybacked onto data packets
- Evaluation metric
  - Average Redundancy (AR): ratio between
    - total number of data packets transmitted on a link
    - total number of *unique* data packets transmitted on that link
  - Maximum Redundancy (MR): maximum number of copies of any data packet that traverse a link

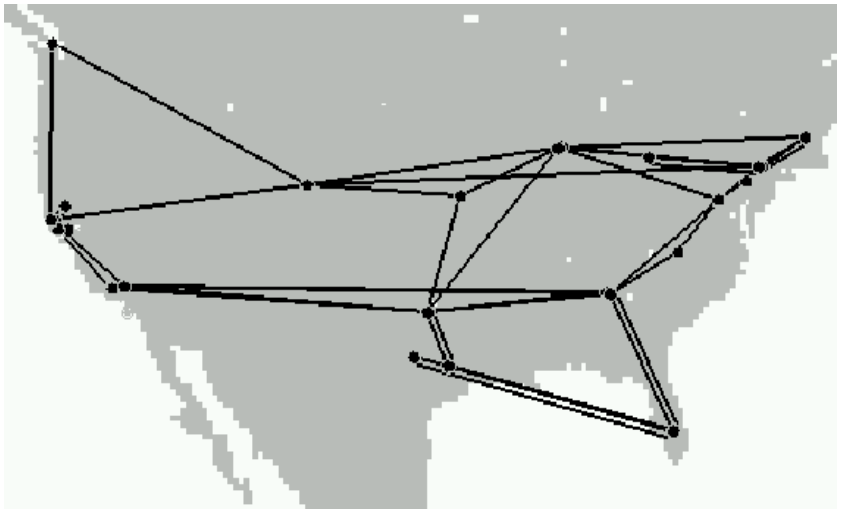
Load Balancing and Graceful Degradation

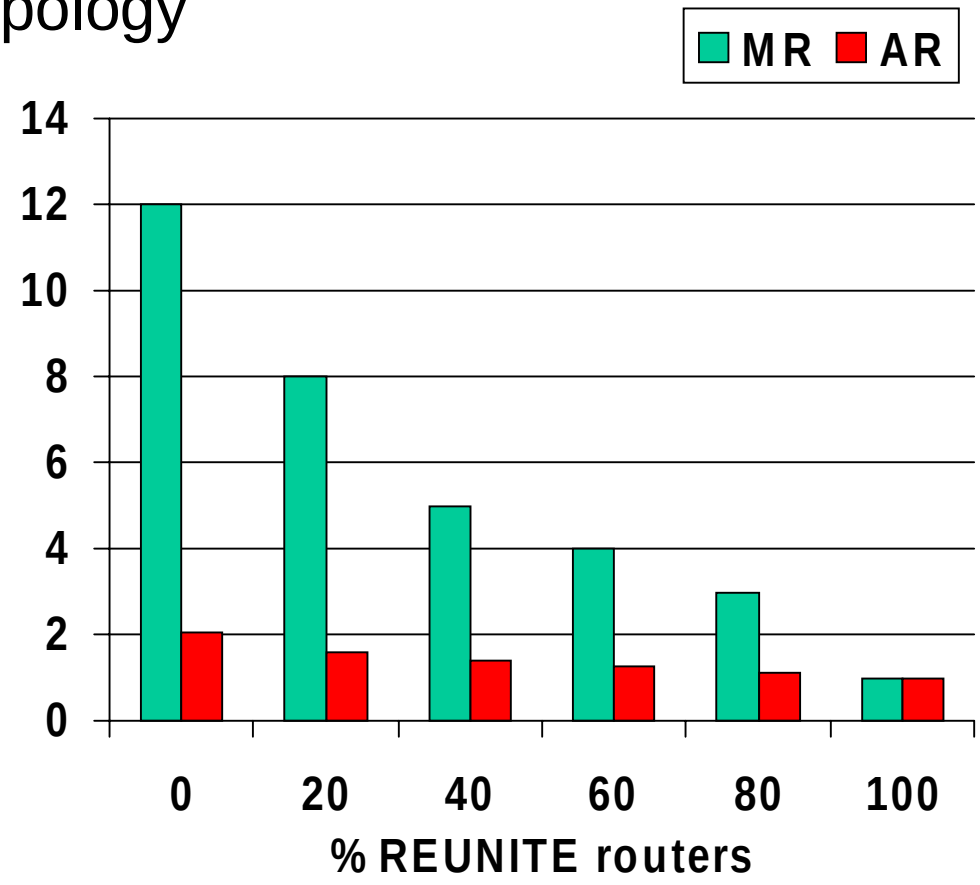




Incremental Deployment

- 8 groups and 64 receivers randomly placed in the 1998 MCI backbone topology





Performance Under Dynamic Joins and Leaves

- 8 groups and 64 receivers placed in 1998 MCI backbone topology
- All routers are REUNITE aware
- Receiver joins modeled by an on-off process
  - OFF period exponentially distributed with mean of 5 seconds
  - ON period exponentially distributed with mean of 25 seconds
- Results
  - AR <= 1.06
  - MR <= 3