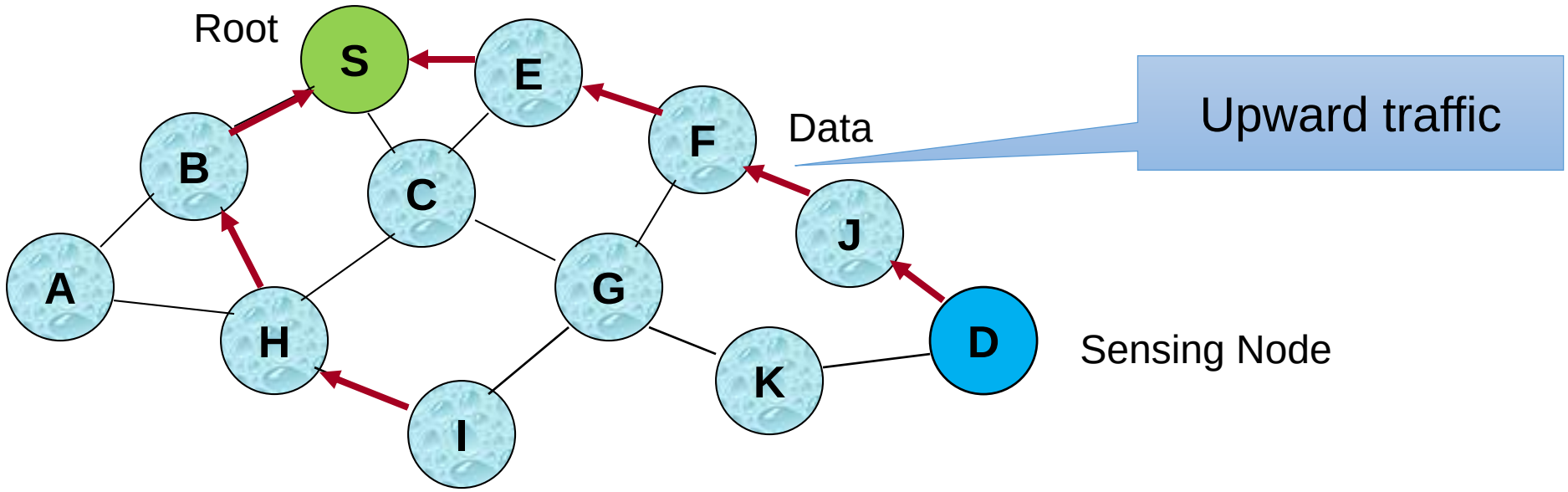
Source Routing for Downward Data Traffic

WSN Lab final project, 2016-2017

Upward traffic



Many-to-one, up the tree

Source Routing – overview

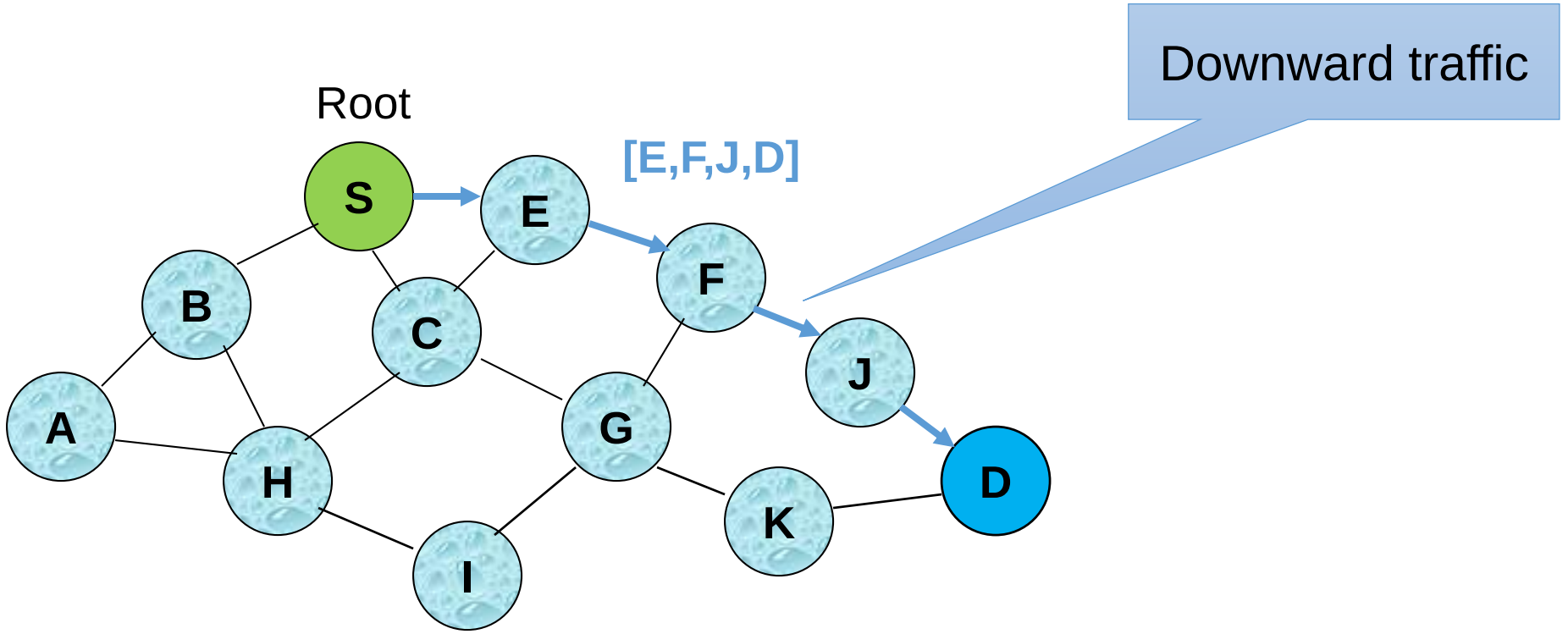
- Source specifies entire route: places complete path to destination in the message header:
- Computed by the root node only

R1,R2,R3,**D**	Payload

- Intermediate nodes just forward to specific next hop:

R1 would look at path in header and forward to **R2**, and so on until the packet reaches the destination **D**

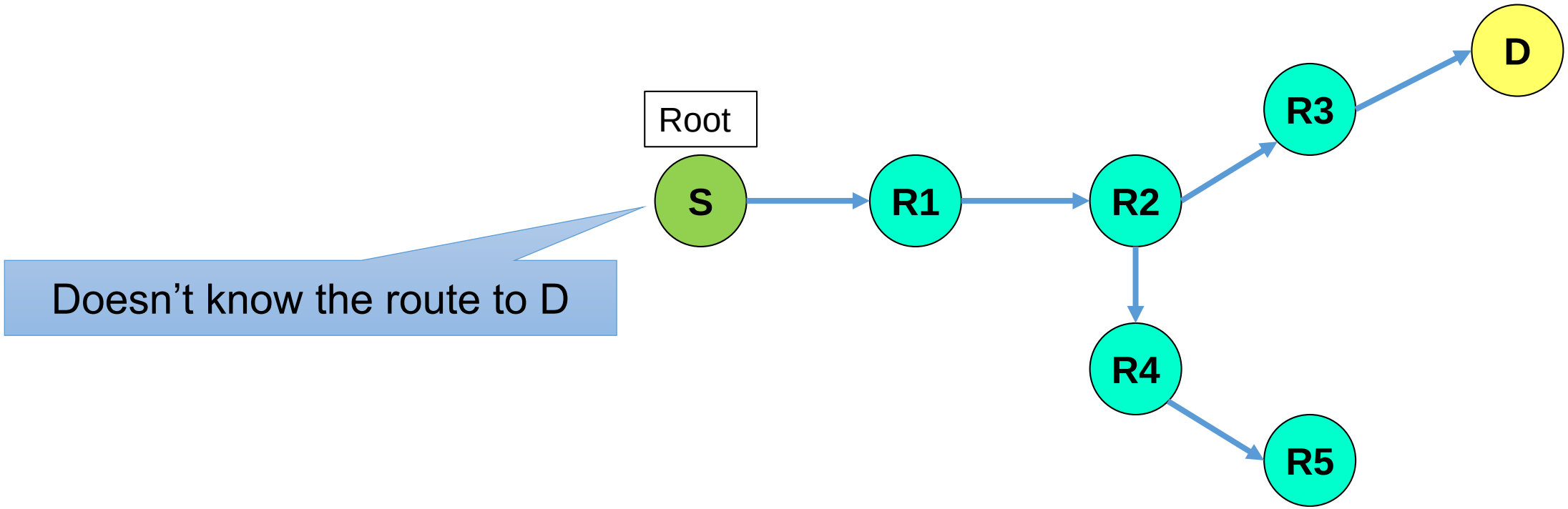
Source Routing – overview



Sink (the tree root) can now also send packets to the nodes

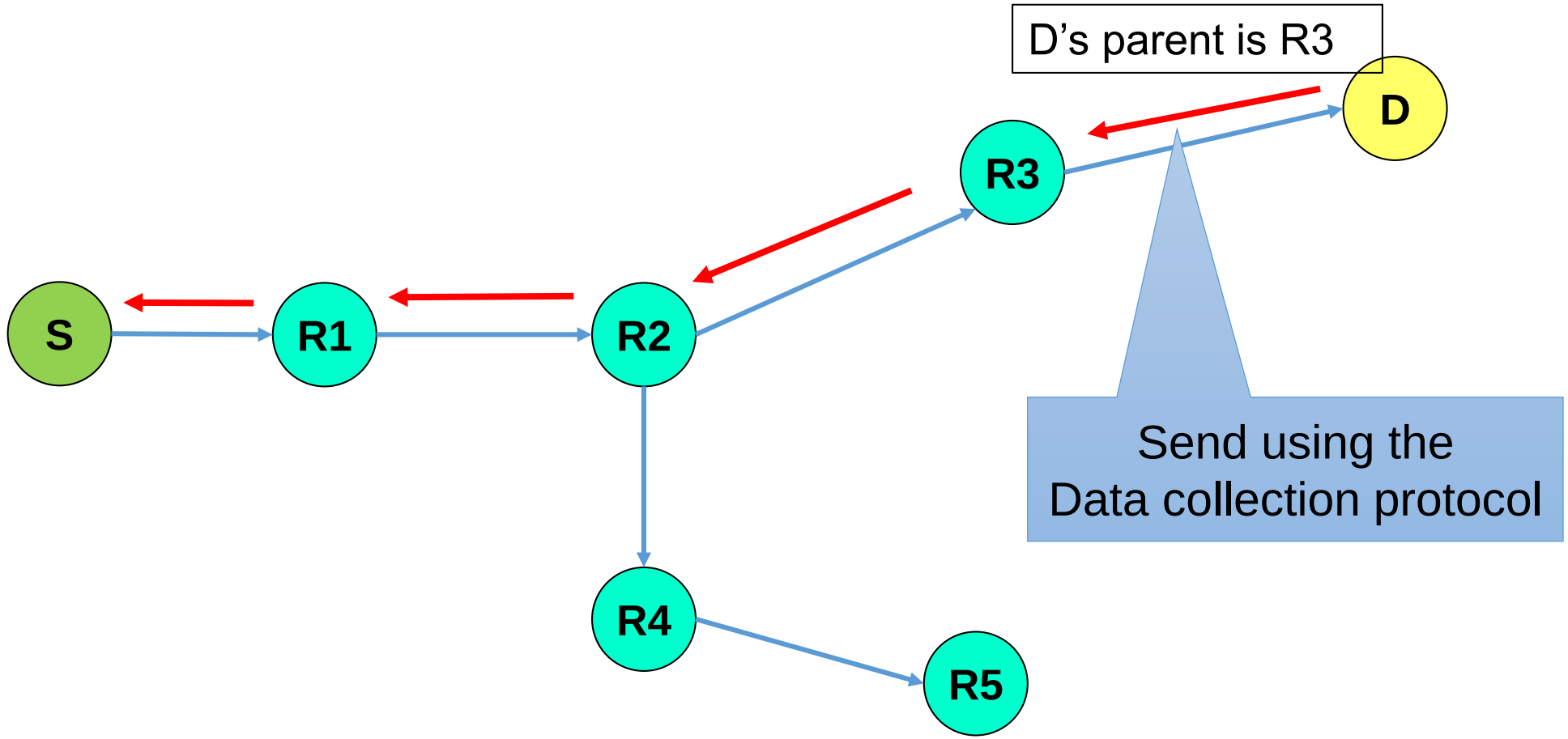
To begin with:

- Objective: Node S wants to send data to node D



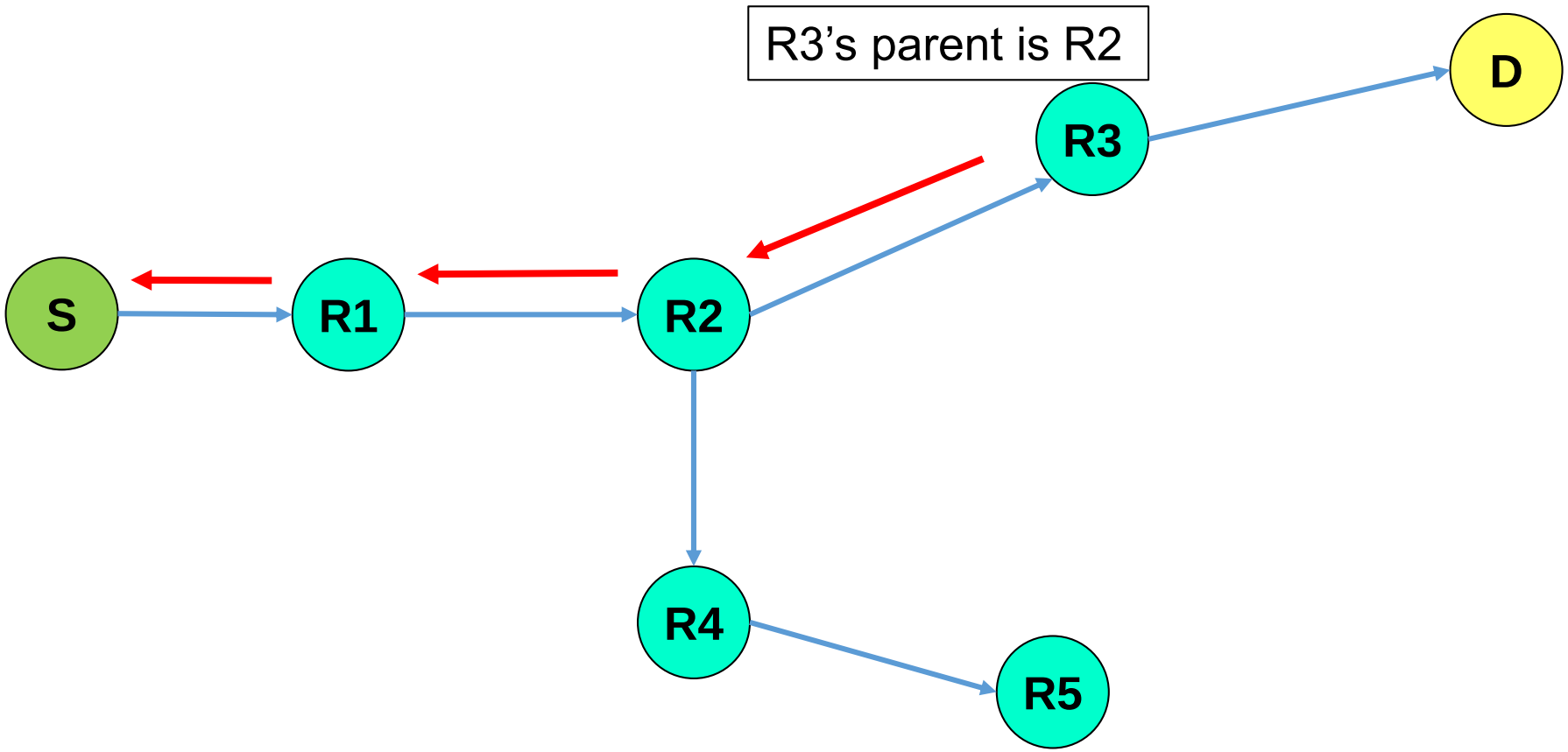
Phase 1: Collecting the routing information

Child	Parent
D	R3



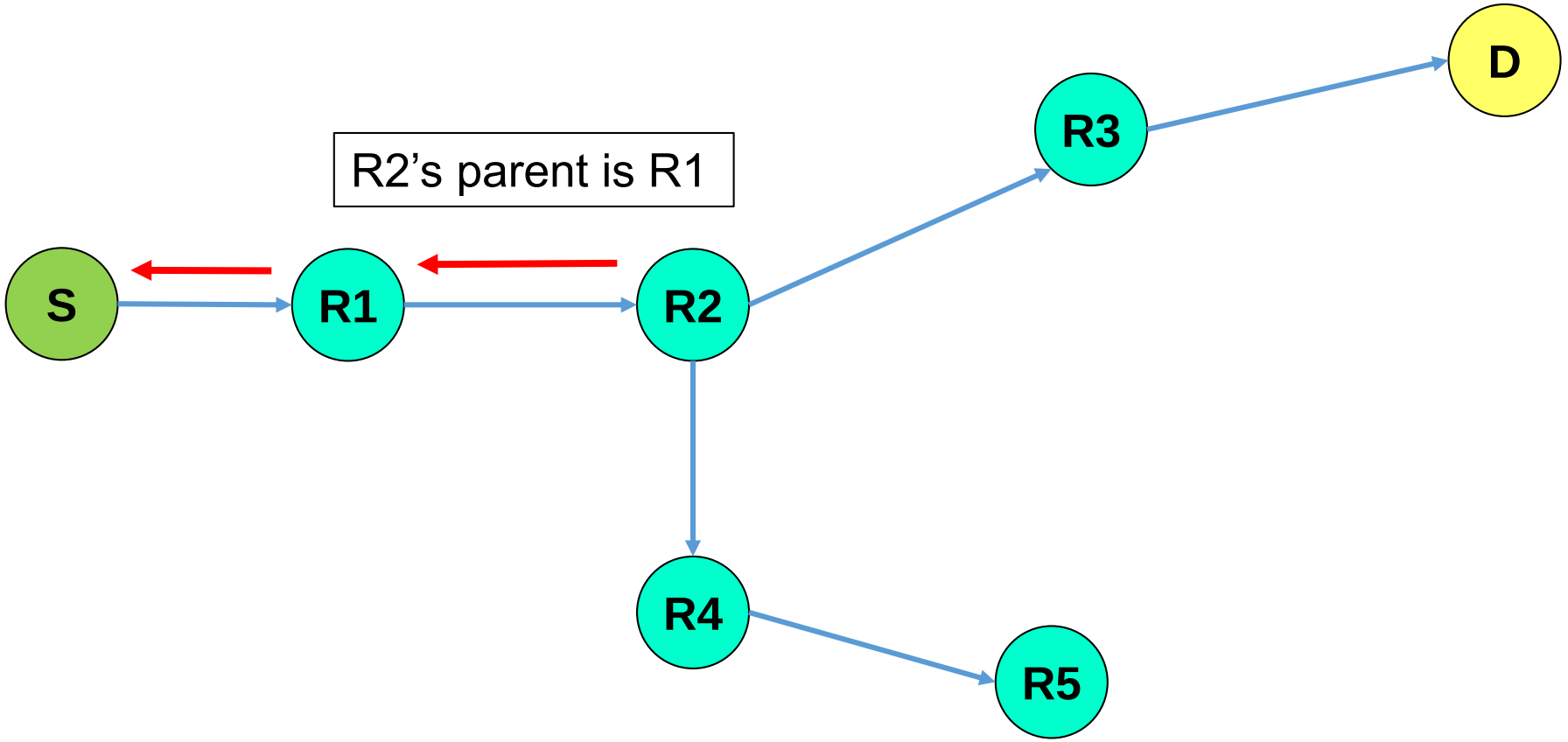
Phase 1: Collecting the routing information

Child	Parent
R3	R2
D	R3



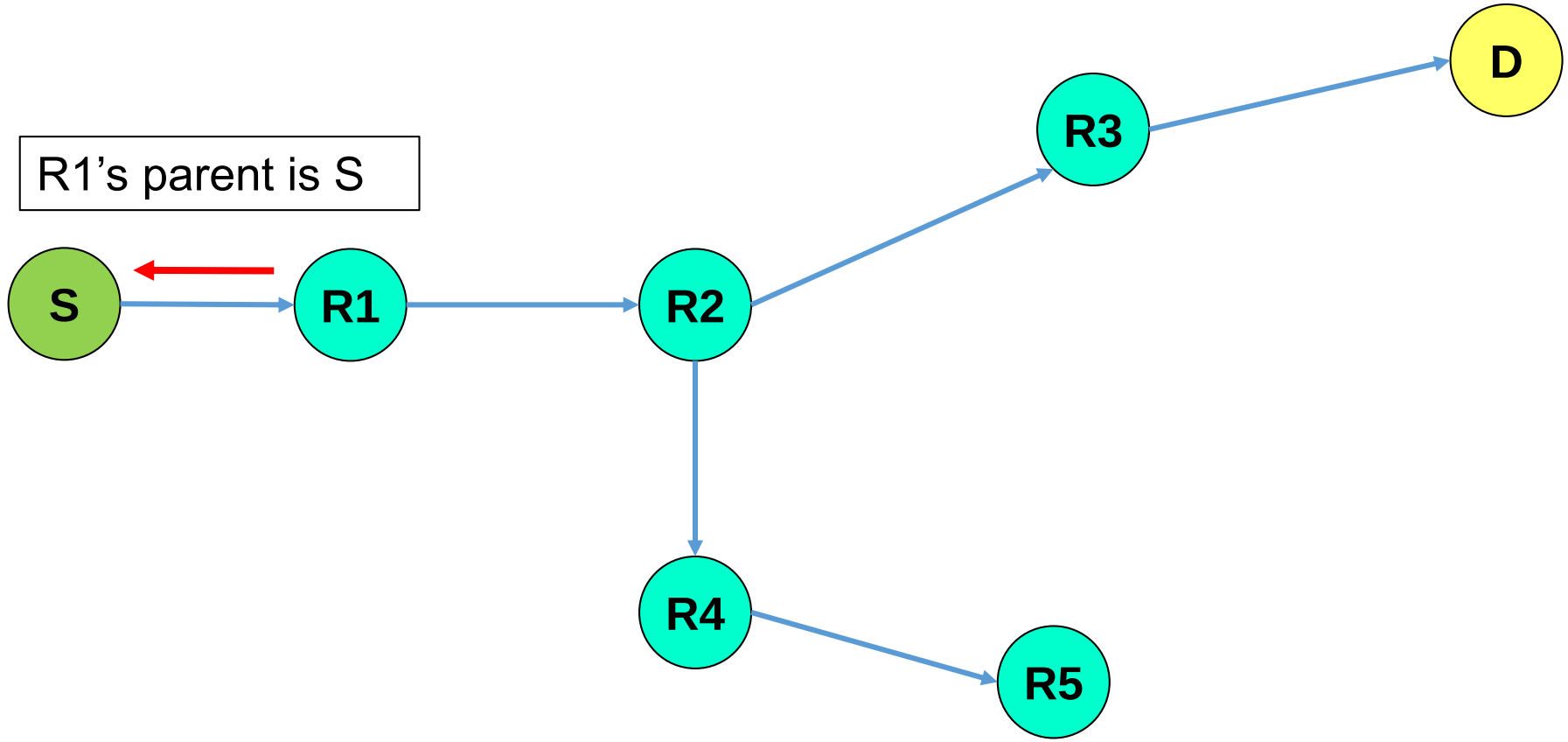
Phase 1: Collecting the routing information

Child	Parent
R2	R1
R3	R2
D	R3



Phase 1: Collecting the routing information

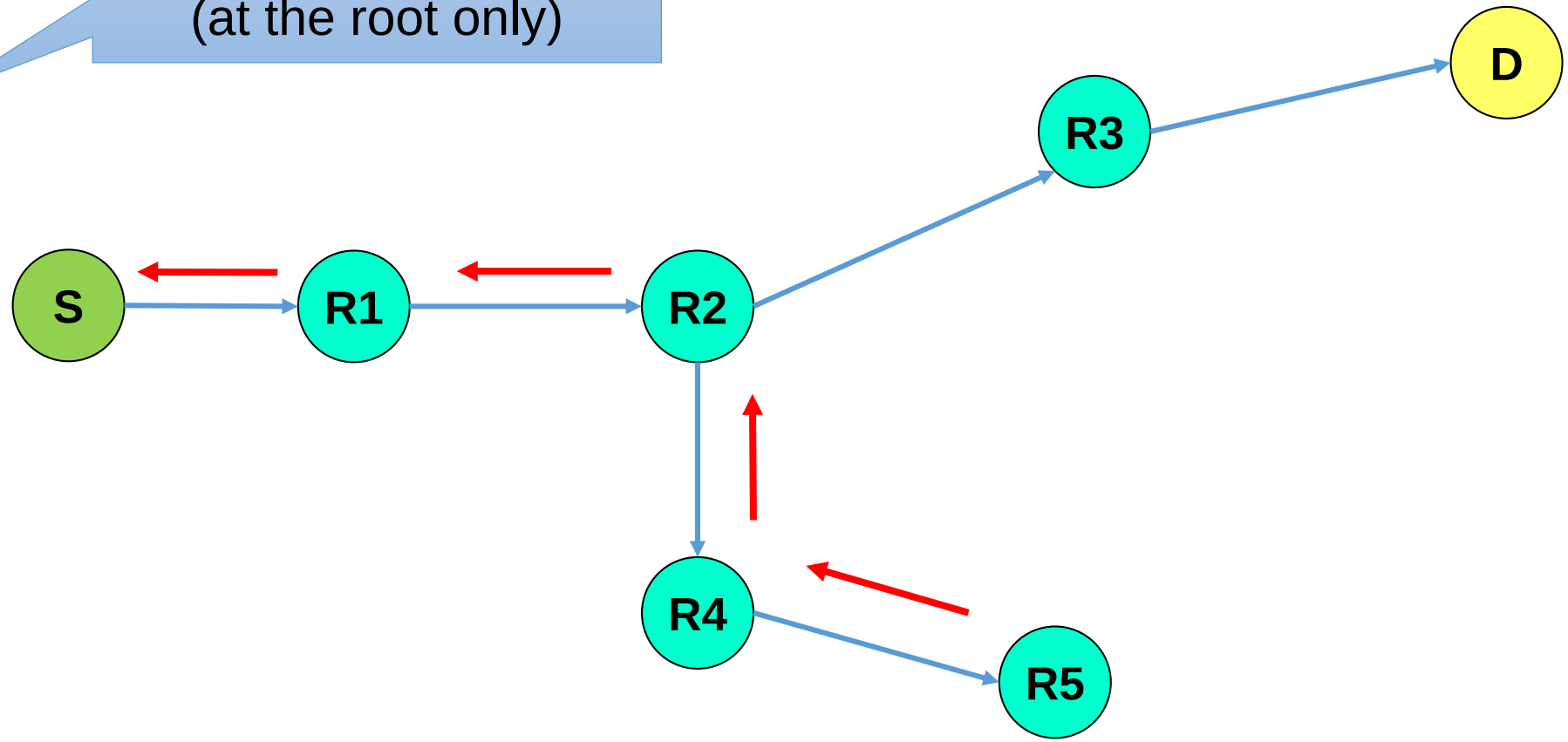
Child	Parent
R1	S
R2	R1
R3	R2
D	R3



Phase 1: Collecting the routing information

Source routing table
(at the root only)

Child	Parent
R1	S
R2	R1
R3	R2
D	R3
R5	R4
R4	R2



When Sending Packets

- When node **S** sends a data packet to **D**, it checks its route cache. If route exists, then entire route is included in the packet header
  - Hence the name source routing

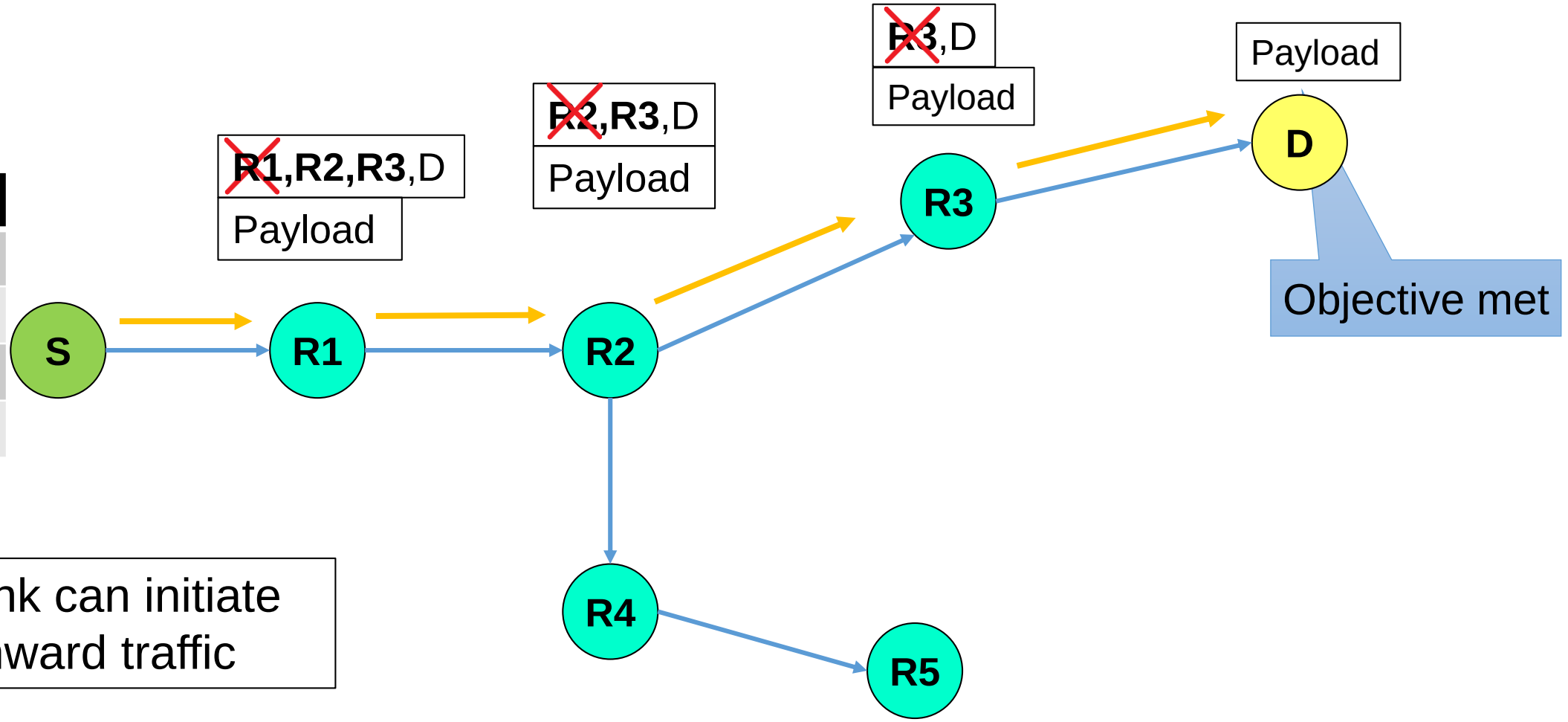
Child	Parent
R1	S
R2	R1
R3	R2
D	R3
R5	R4
R4	R2

Algorithm:

1. Assign N:=D
2. Search for node N in the table to find N's parent P
3. If N is not found or a loop is detected, drop the packet
4. If P ==root, transmit the packet to next-hop node N
5. Else add N to the source routing list of the packet, assign N:=P, go to step 2

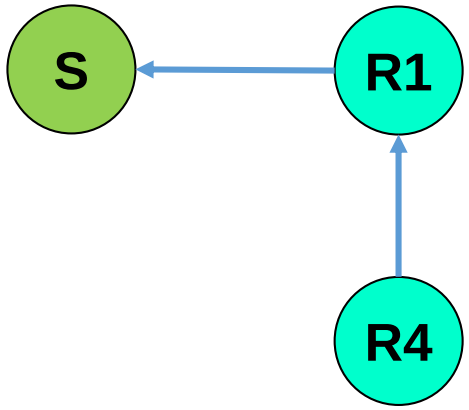
Source Routing - Phase 2: Data Delivery

Child	Parent
R1	S
R2	R1
R3	R2
D	R3



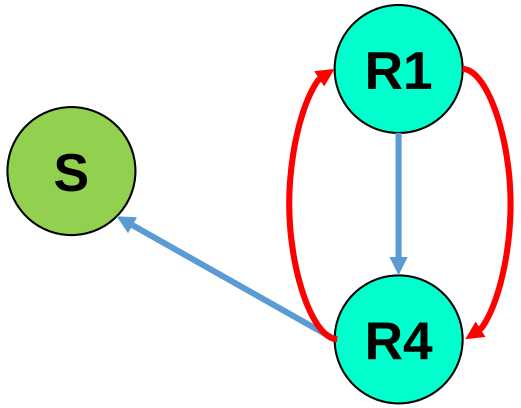
Only sink can initiate downward traffic

Source Routing – Loop detection



Child	Parent
R1	S
R4	R1

If the message "**R4's parent is S**" is lost, we have a loop:



Child	Parent
R1	R4
R4	S

Intended record

Child	Parent
R1	R4
R4	R1

Stale record

Program Structure

- Your program should have:
  - Many-to-one data collection interface (you already have)
  - One-to-many data delivery interface

  - The one-to-many interface should provide **send** command and **receive** event
    - The sender (root) specifies the node to which it wants to send the data

      OneToMany.send(**destination ID**, message)

  - The application module that uses these interfaces