CS4450, CS5456
Introduction to Computer Networks

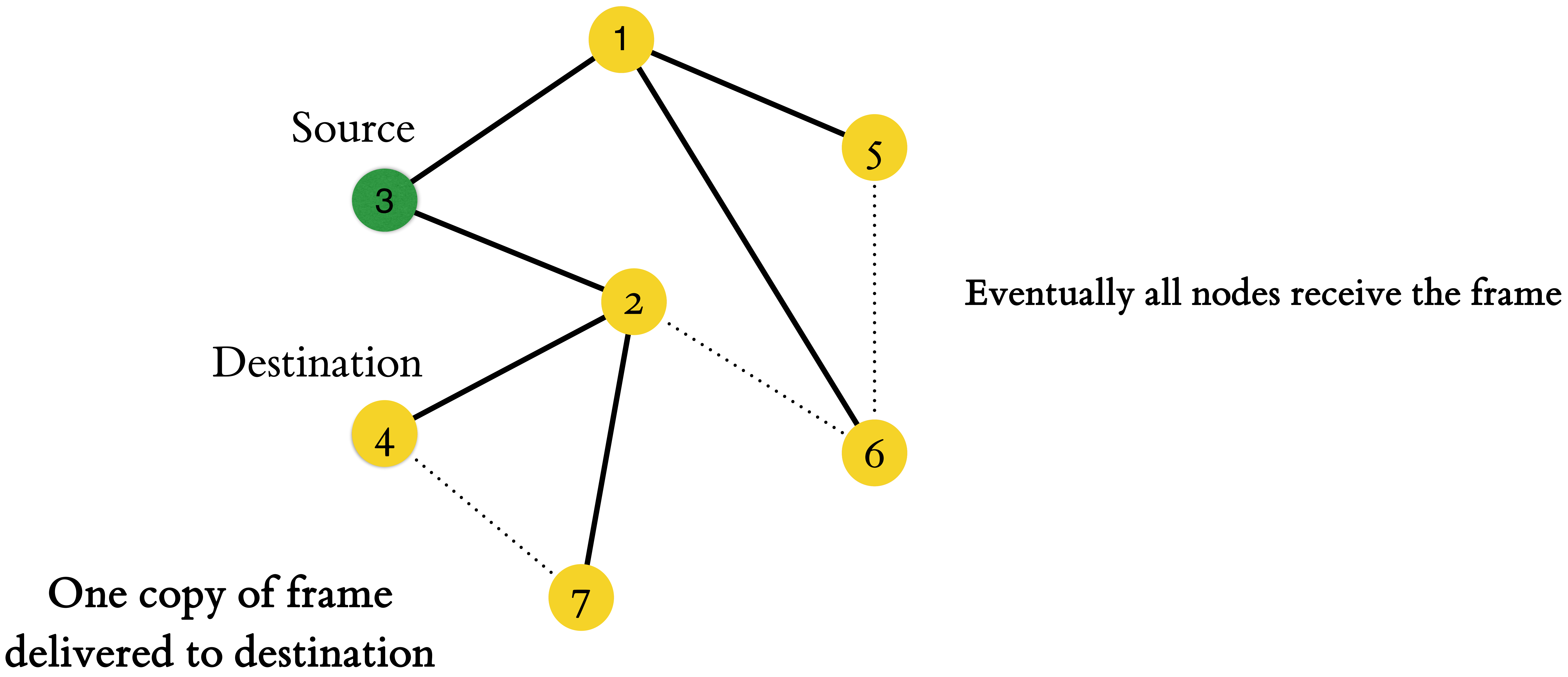
Lecture 10

Rachee Singh

Routing on a spanning tree using flooding

1. Step 1: Ignore all links not belonging to the spanning tree
2. Step 2: Source node sends or “floods” frames out to every neighbor on the spanning tree
3. Step 3: Send incoming frame out to all links other than the one that sent them

Flooding example



Spanning Tree

Routing on a spanning tree using flooding

1. Easy routing algorithms for trees
   1. Simply flood packets to all neighbors
2. Good properties:
   1. No loops
   2. No complex state to be stored on each node

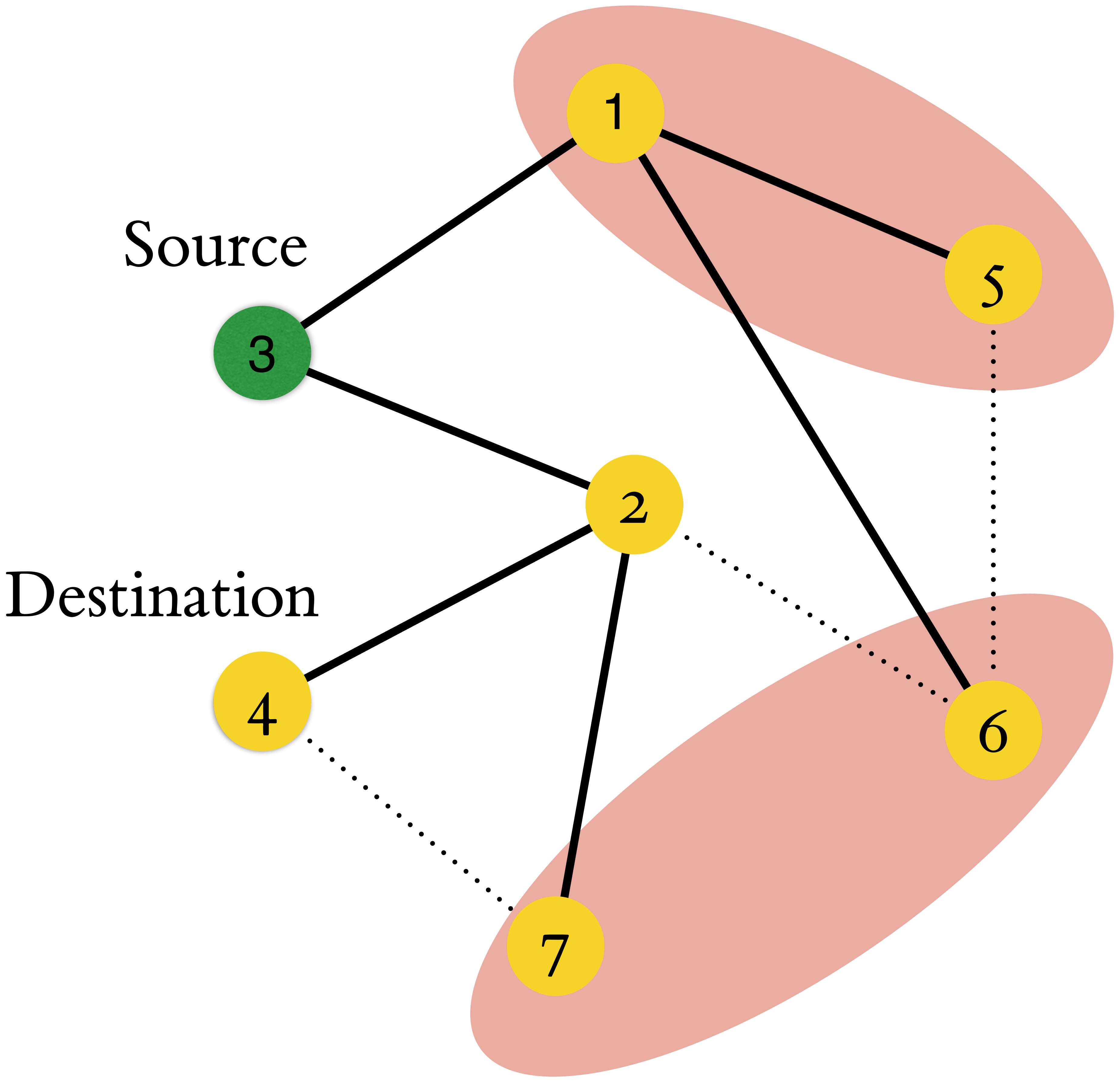
What have we learnt so far?

1. Terminology
    1. End systems (or hosts), switches, routers, links
    2. Network performance metrics (latency, bandwidth, BDP, loss)
2. Design choices
    1. Packets vs. circuits
    2. Best effort vs. reliable (or guaranteed) delivery
    3. Layering
    4. End-to-end principle ("smart" end hosts but "dumb" network)
3. Physical layer
    1. Modulate bits onto signals (modulation formats, signal quality)
    2. Shanon and Hartley's law to channel capacity
4. Data link layer
    1. Broadcast ethernet (legacy) with CSMA/CD
    2. Switched ethernet
    3. Broadcast storms and spanning tree protocol

What now?

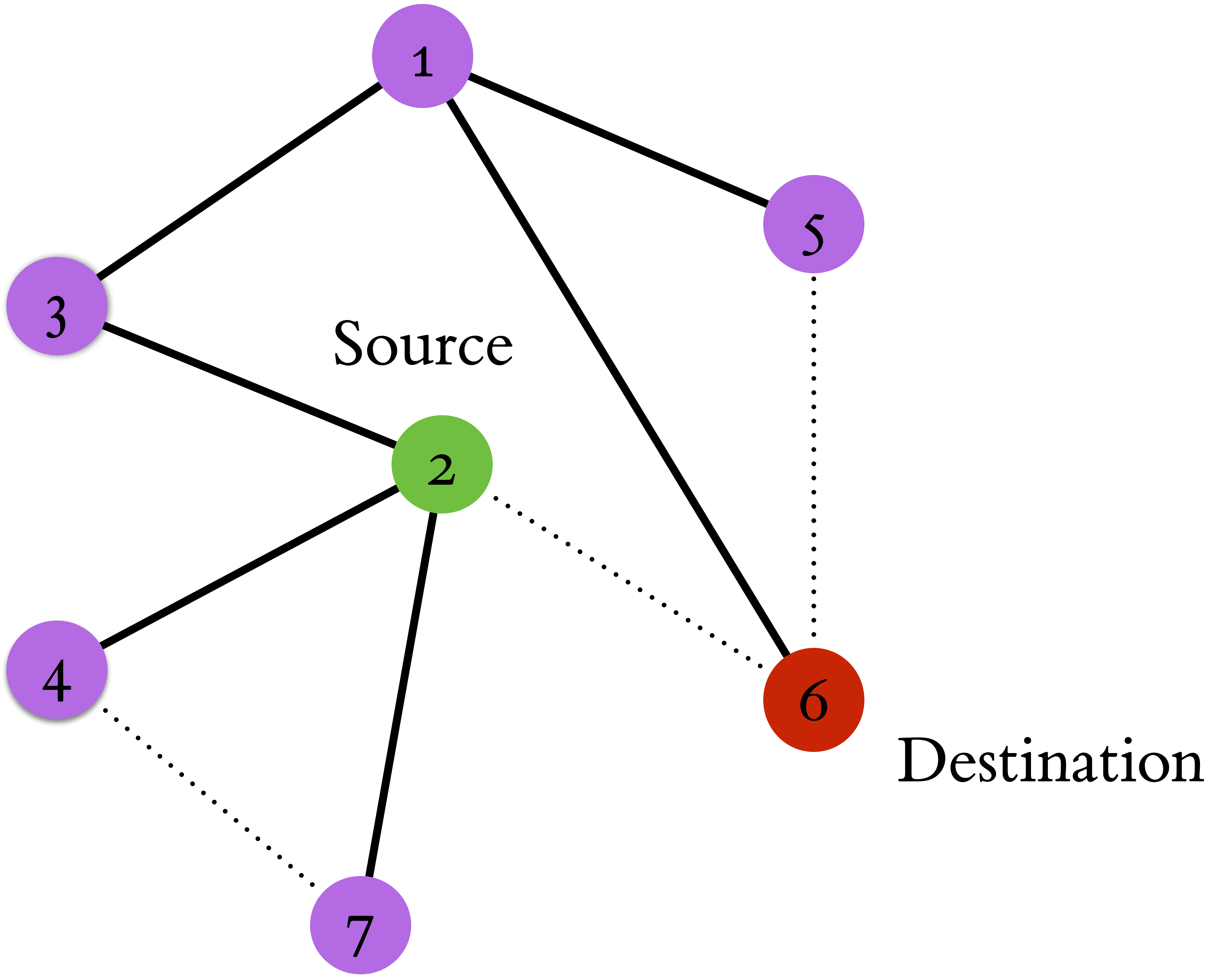
1. The STP allows hosts connected to Ethernet to communicate
2. Do we need anything else for hosts to communicate on networks?
3. If you have multiple Ethernet networks
   1. Why not use spanning tree for all these networks?
   2. In short, given we have the data link layer (L2), why do we need the network layer (L3)?

Issues with routing using flooding



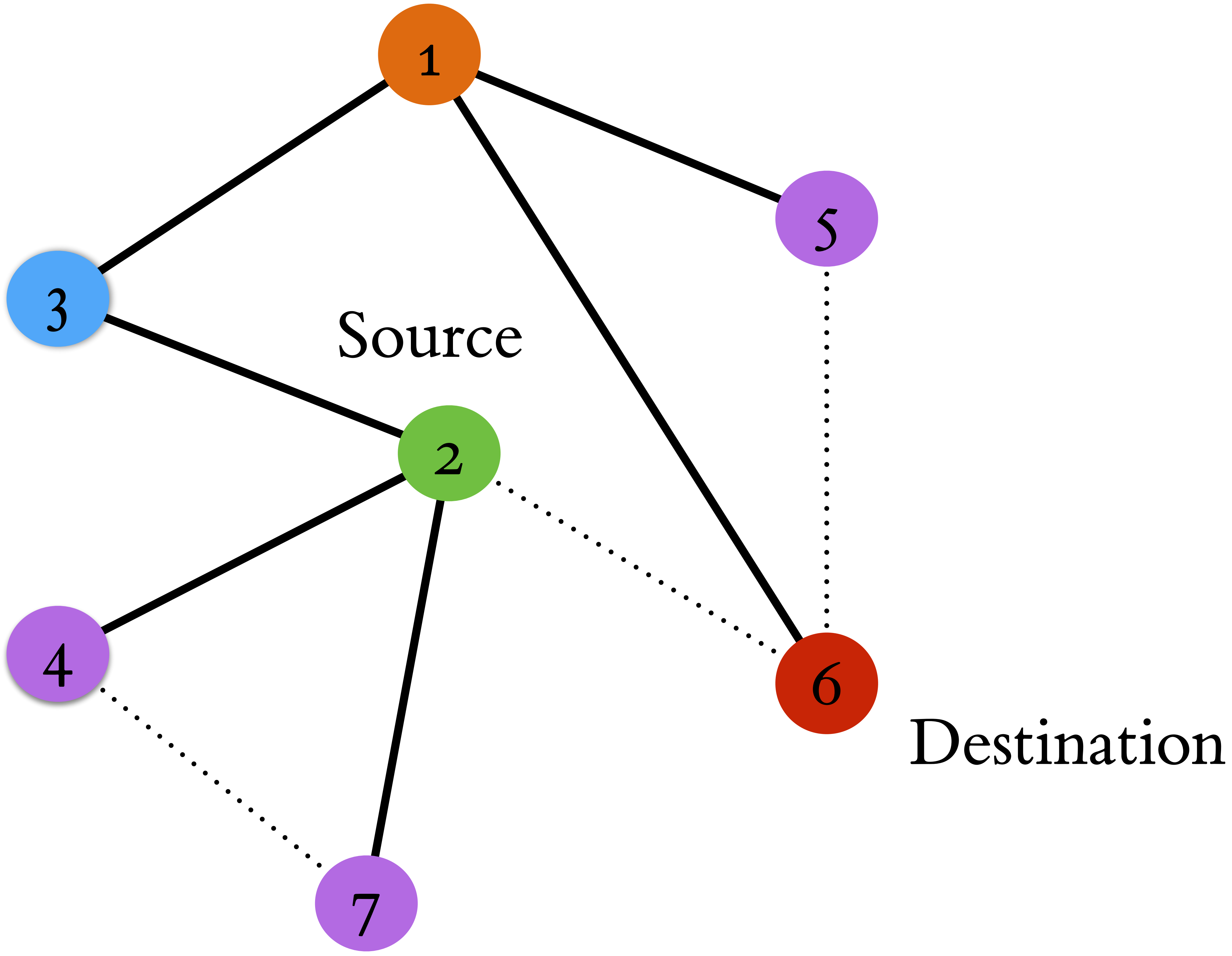
Issue 1: Each host has to do unnecessary packet processing!
(to decide whether the packet is destined to the host)

Issues with routing using flooding



Issue 2: Higher latency!
(The packets unnecessarily traverse much longer paths)

Issues with routing using flooding



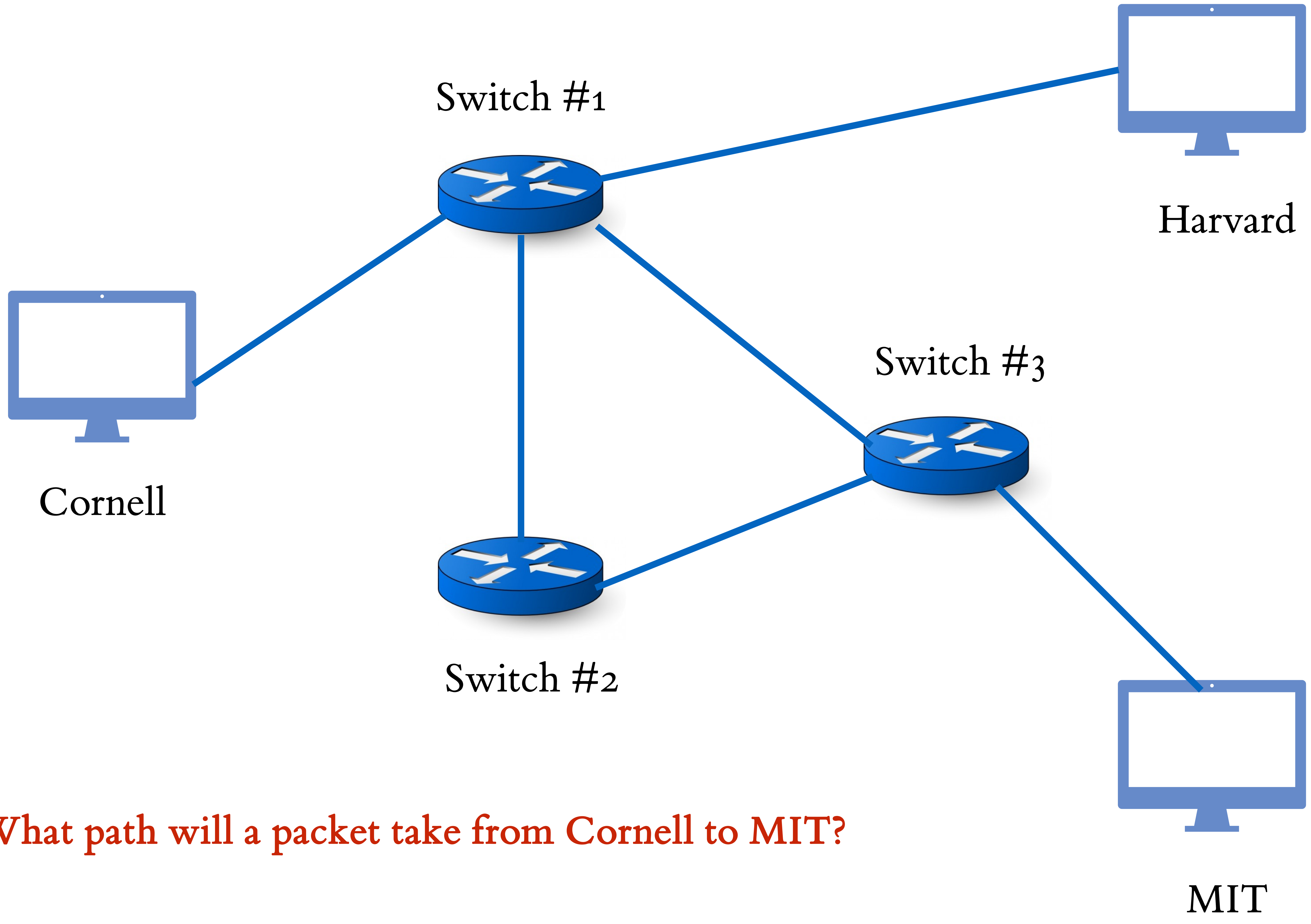
Issue 3: Lower bandwidth availability!

(Frames between 2-6 and 3-1 unnecessarily have to share bandwidth)

Why do we need a network layer?

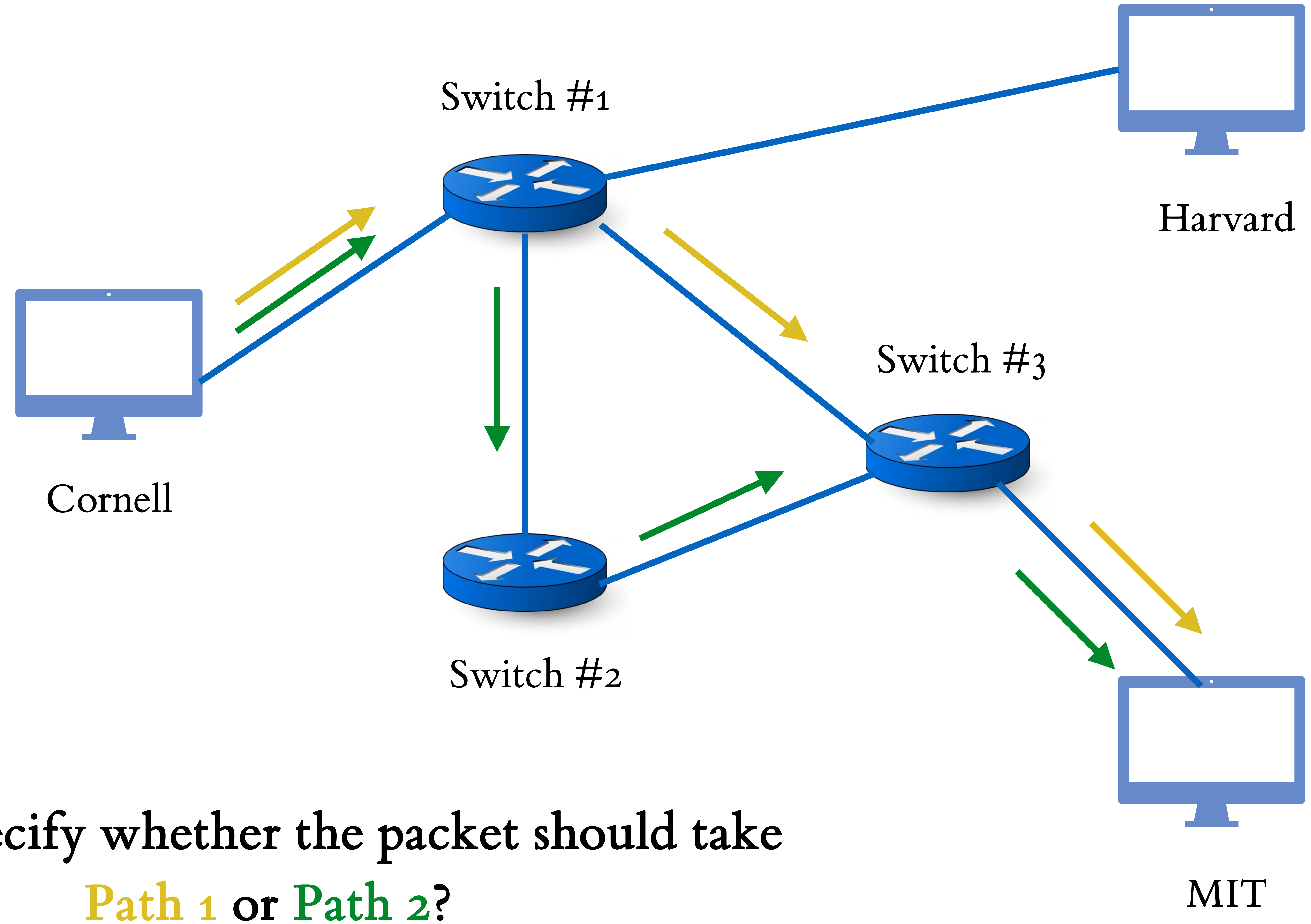
1. Network layer performs *routing* of packets
    1. Alleviates issues with flooding
    2. It uses routing tables
    3. At Network layer (L3), packets are the unit of communication
    4. .. like frames at L2

Routing packets using routing tables



What path will a packet take from Cornell to MIT?

Routing packets using routing tables



How to specify whether the packet should take
Path 1 or Path 2?

Network Layer (L3)

Network Layer Topics

1. Addressing
2. Forwarding
3. Routing

Network Layer Addresses

1. Network layer addresses are commonly called “IP addresses”
2. Hierarchical (unlike flat addresses)
3. We will discuss them later

Forwarding

1. The process by which a *local router* determines
    1. Output link (or next-hop) for each packet
2. How does the router do this?
    1. Read destination address from packet's L3 header
    2. Search own routing table for the destination address

Routing

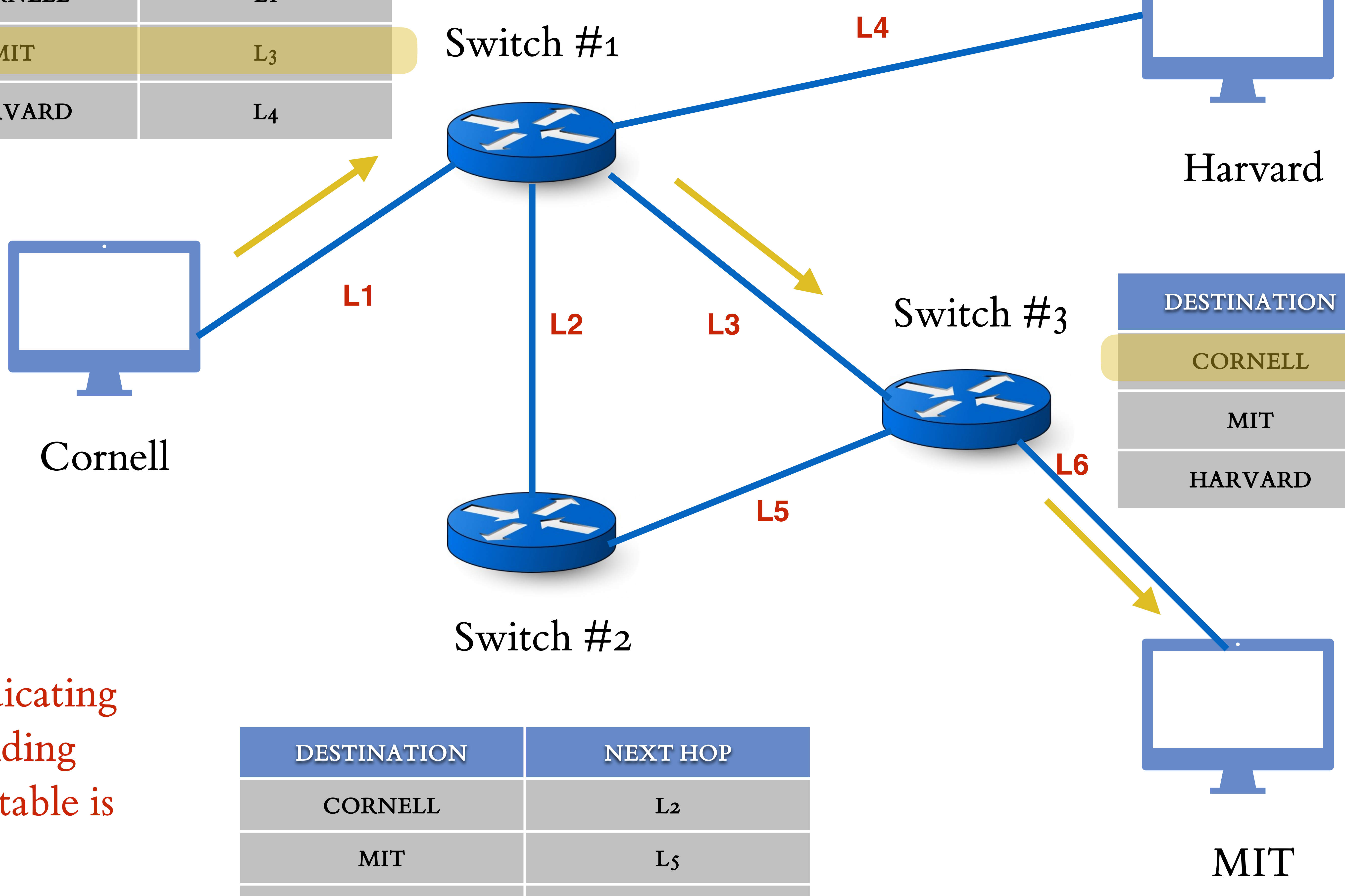
1. *Network-wide* process that determines
    1. Content of routing tables
2. Routing determines the end-to-end path for each destination
3. How do routers do this?
    1. This is the focus of this and the next few lectures

Routing vs. Forwarding

1. Forwarding is a *"data plane"* function
    1. Directing one packet at a time
    2. Happens locally on every router
2. Routing is a *"control plane"* function
    1. Computing the routing tables that guide packets
    2. Jointly computed by routers using distributed algorithms
3. Forwarding and routing happen at *very different timescales*

Routing packets using routing tables

DESTINATION	NEXT HOP
CORNELL	L1
MIT	L3
HARVARD	L4



DESTINATION	NEXT HOP
CORNELL	L5
MIT	L6
HARVARD	L3

Each switch stores a table indicating the next hop for corresponding destination of a packet. This table is called a routing table

DESTINATION	NEXT HOP
CORNELL	L2
MIT	L5
HARVARD	L5

Goals of routing

1. Goal 1: *valid* routing in the network
    1. Routing finds a path to a given destination
    2. How to know if the state of routers' routing tables is *valid*?
2. Goal 2: *efficient* routing in the network

Validity of global routing state

1. *Local routing state* is the forwarding table in a single router
    1. By itself, the state of single router can not be evaluated
    2. It must be evaluated in terms of the global context
2. *Global routing state* is the collection of forwarding tables of all routers
    1. Global state determines the path packets take

What is “valid” routing state?

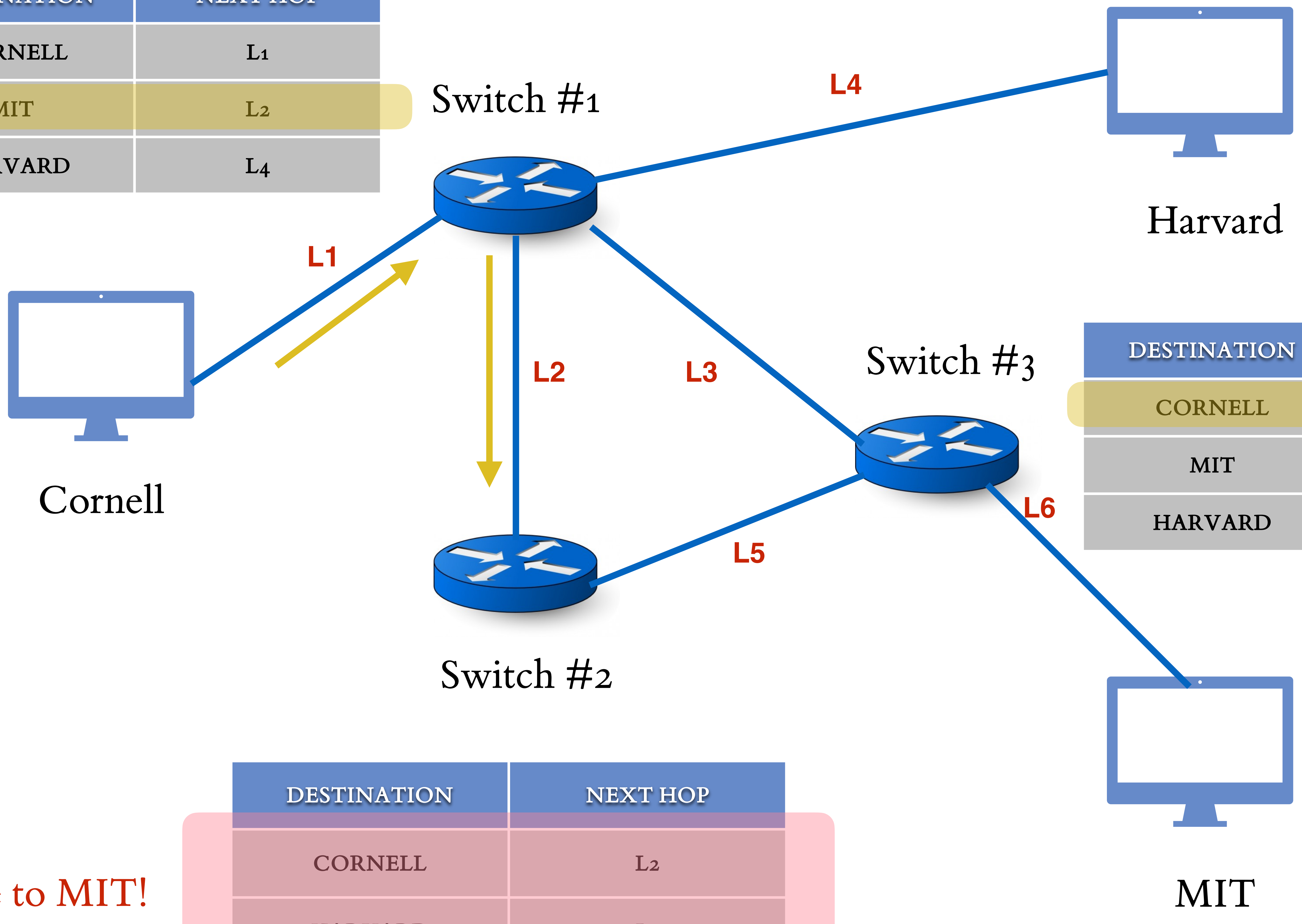
1. Global state is *valid*:
    1. if it produces forwarding decisions that always deliver packets to their destinations
2. Goal of routing protocols: compute valid routing state
3. But, how do we know if the routing state if valid?
    1. Need correctness conditions for routing

Necessary and sufficient conditions for valid routing state

1. Global routing state is valid *if and only if*:
   1. There are *no dead ends* (other than destinations)
   2. There are *no loops*
2. A dead end is when there is no outgoing link or next-hop
   1. Packet arrives at a router
   2. But the forwarding decision does not find a next-hop
3. A loop is when a packet cycles around the same set of nodes forever

Dead ends in routing state

DESTINATION	NEXT HOP
CORNELL	L1
MIT	L2
HARVARD	L4



DESTINATION	NEXT HOP
CORNELL	L5
MIT	L6
HARVARD	L3

Dead end! No route to MIT!

DESTINATION	NEXT HOP
CORNELL	L2
HARVARD	L5

Necessary (“only if”) condition for validity

1. If you run into a dead end before reaching the destination
   1. Will never reach destination
2. If you run into a loop
   1. Will never reach the destination

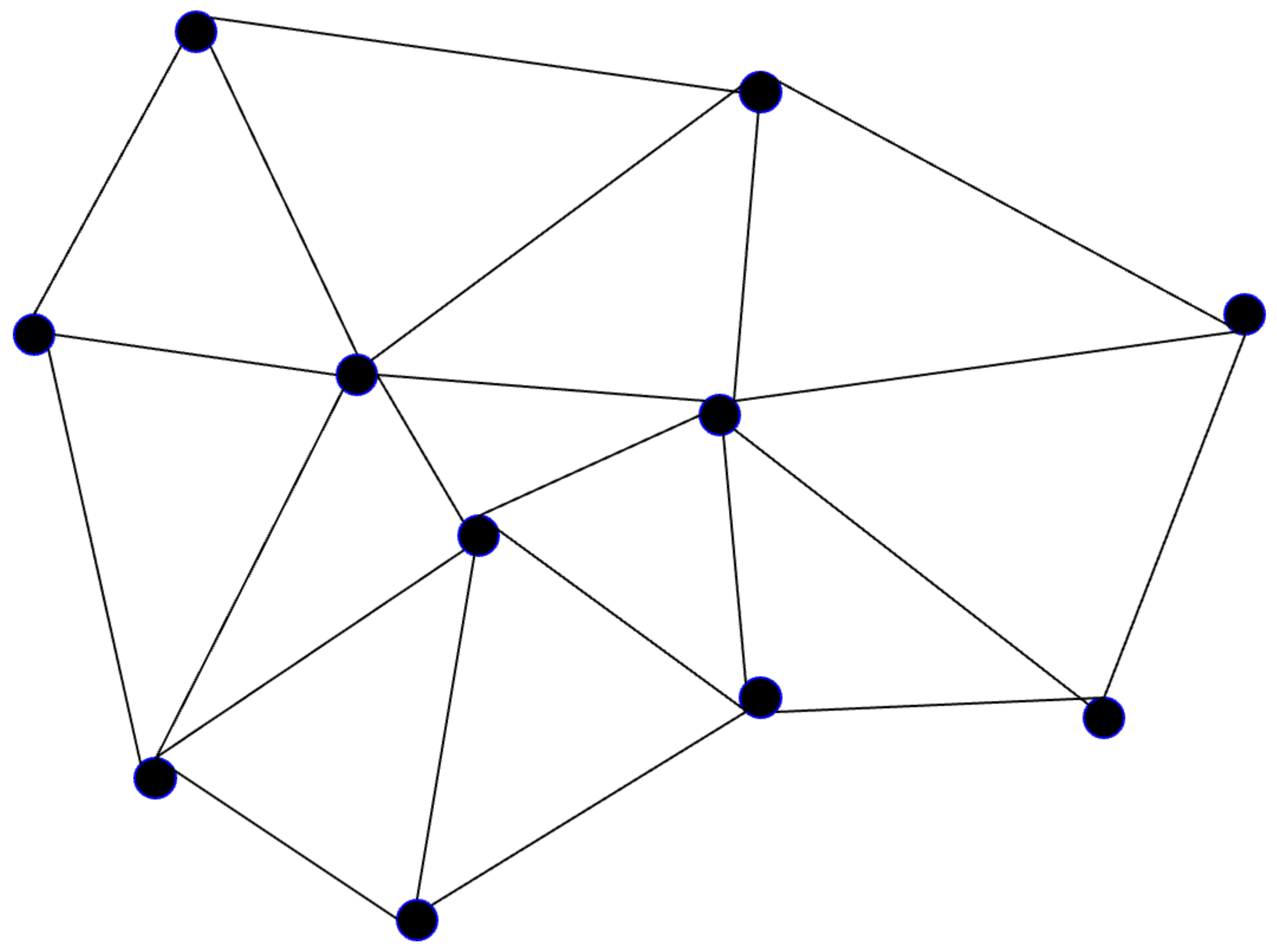
Sufficient (“if”) condition for validity

1. Assume no dead ends and no loops
2. Packet must keep wandering the network (without repeating)
3. Only finite number of links to visit in the network
    1. It cannot keep wandering forever (since no loops)
    2. Must eventually reach the destination

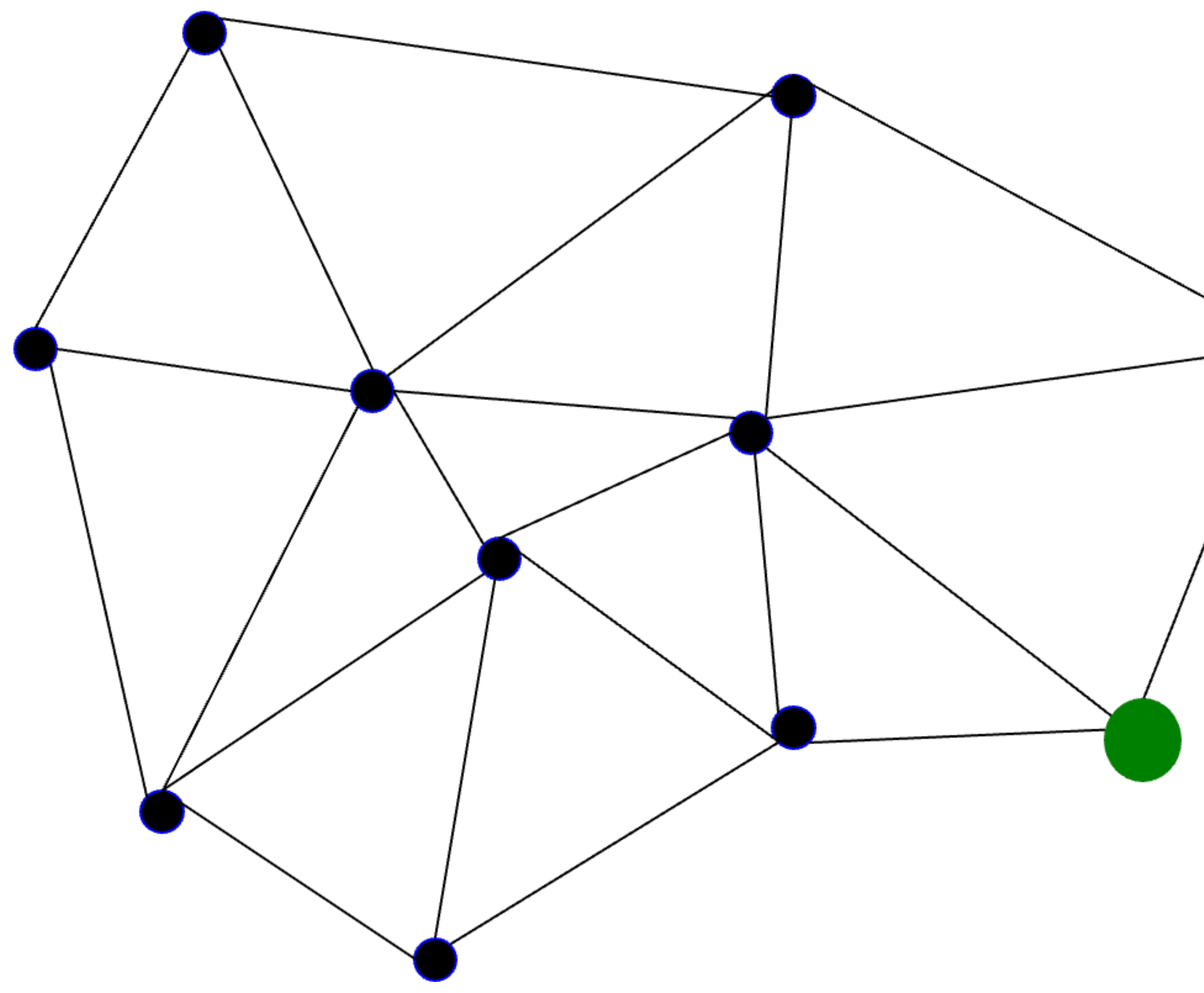
Check validity of routing state

1. Focus on a single destination
    1. Ignore all other routing state
2. Mark outgoing link or next-hop with an arrow
    1. Only one at each router
3. Eliminate all links with no arrows
4. Look at what is left

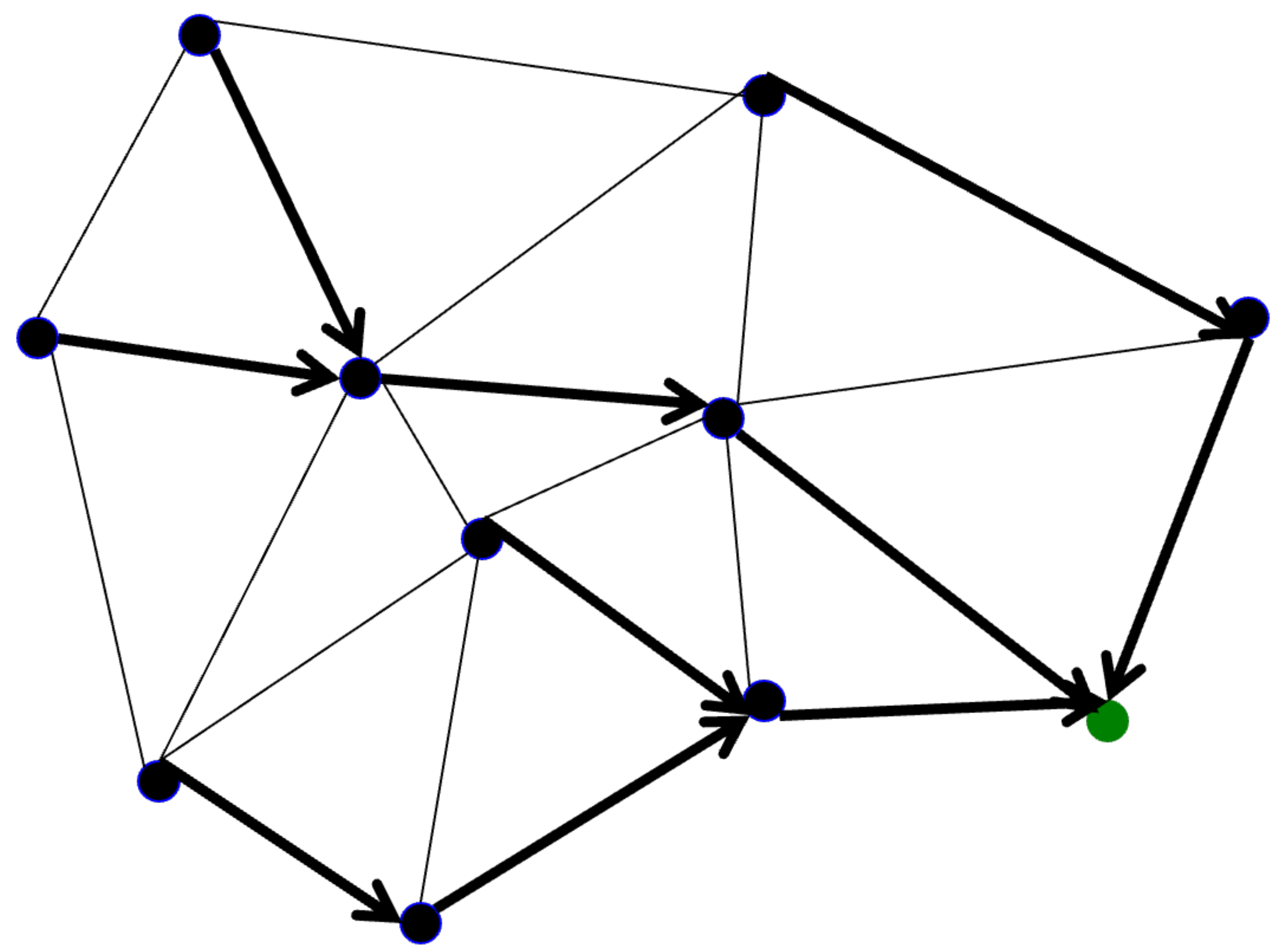
Check validity of routing state: example



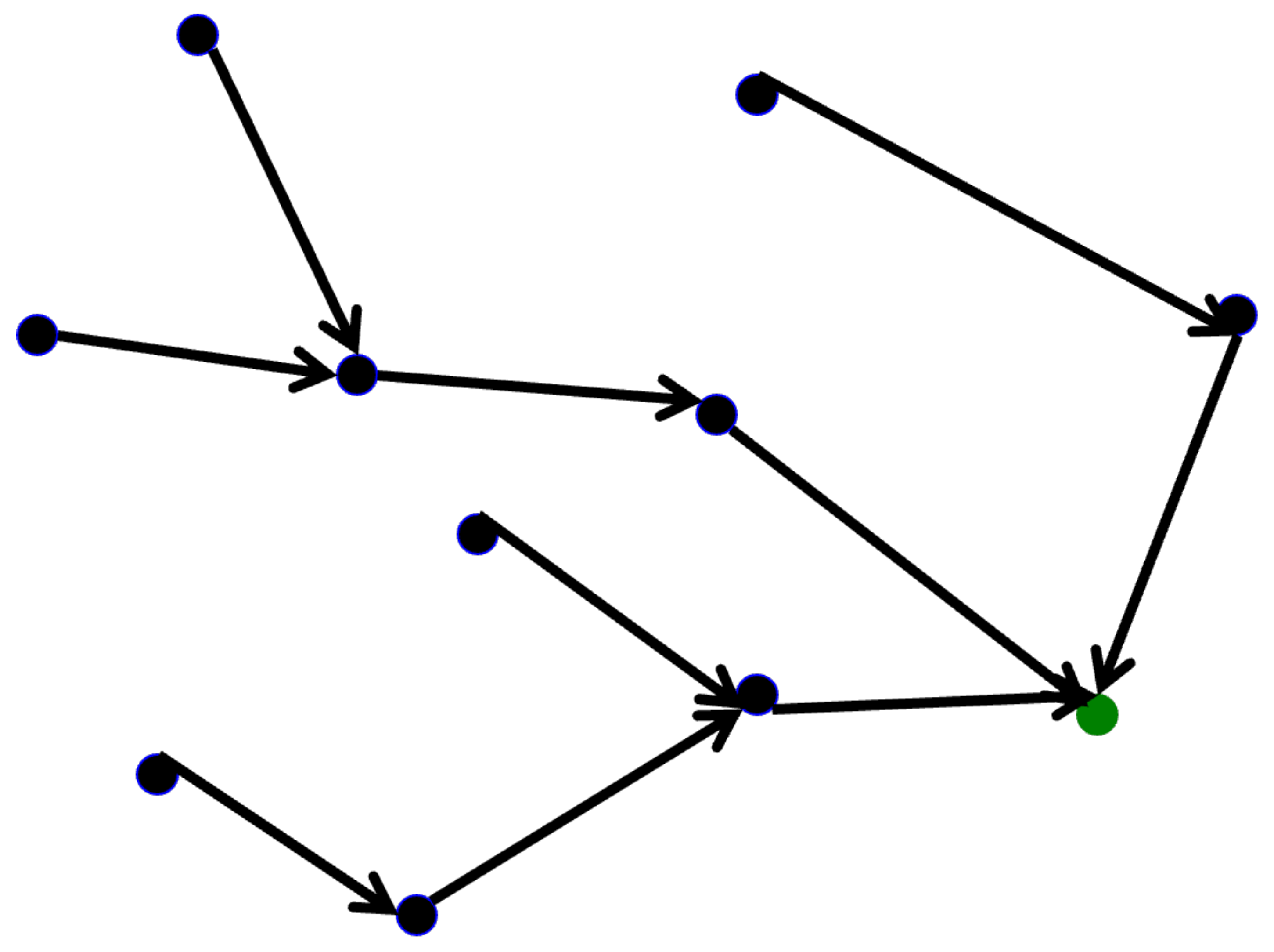
Example: pick a destination



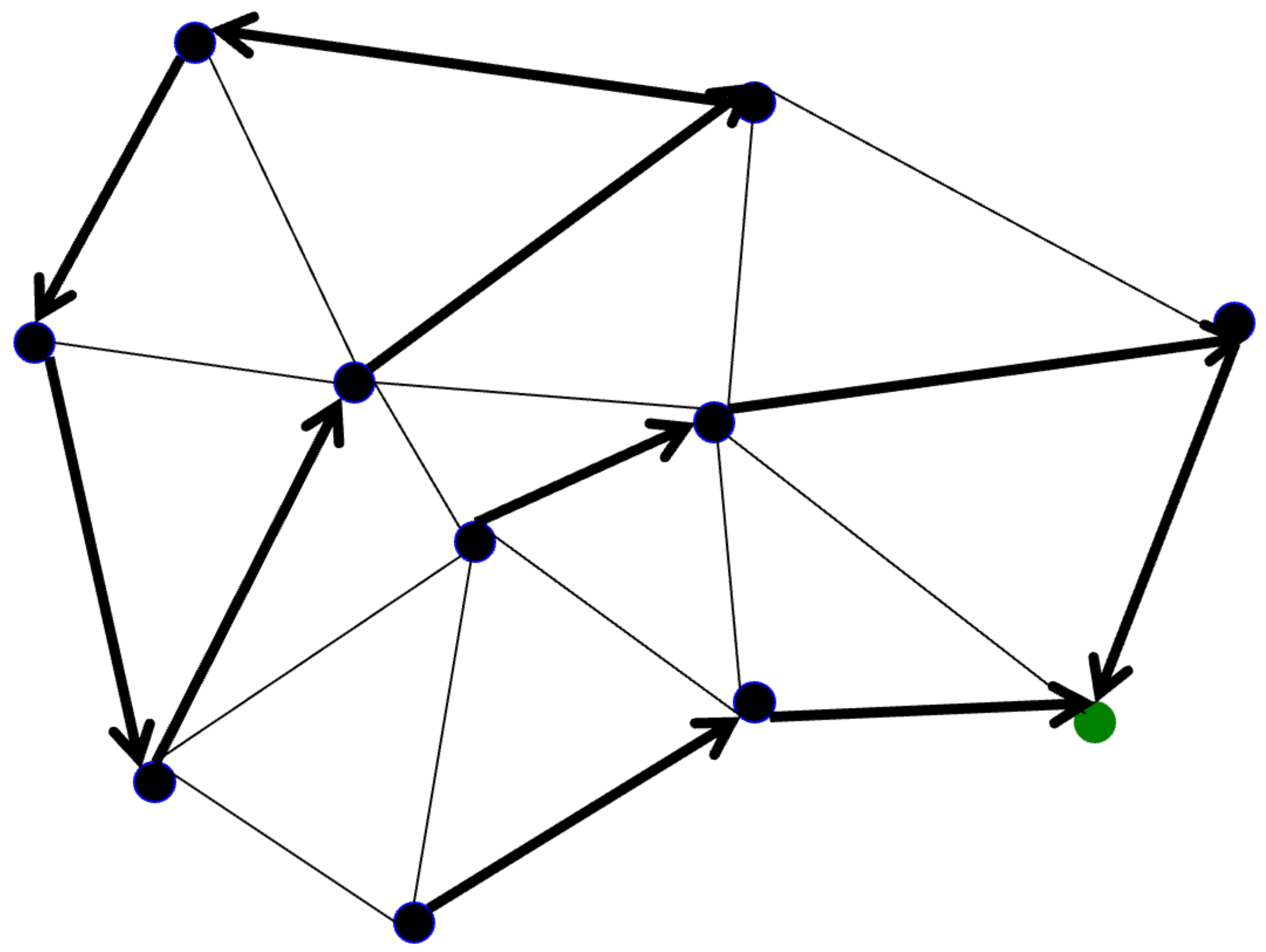
Example: put arrows on outgoing links to destination



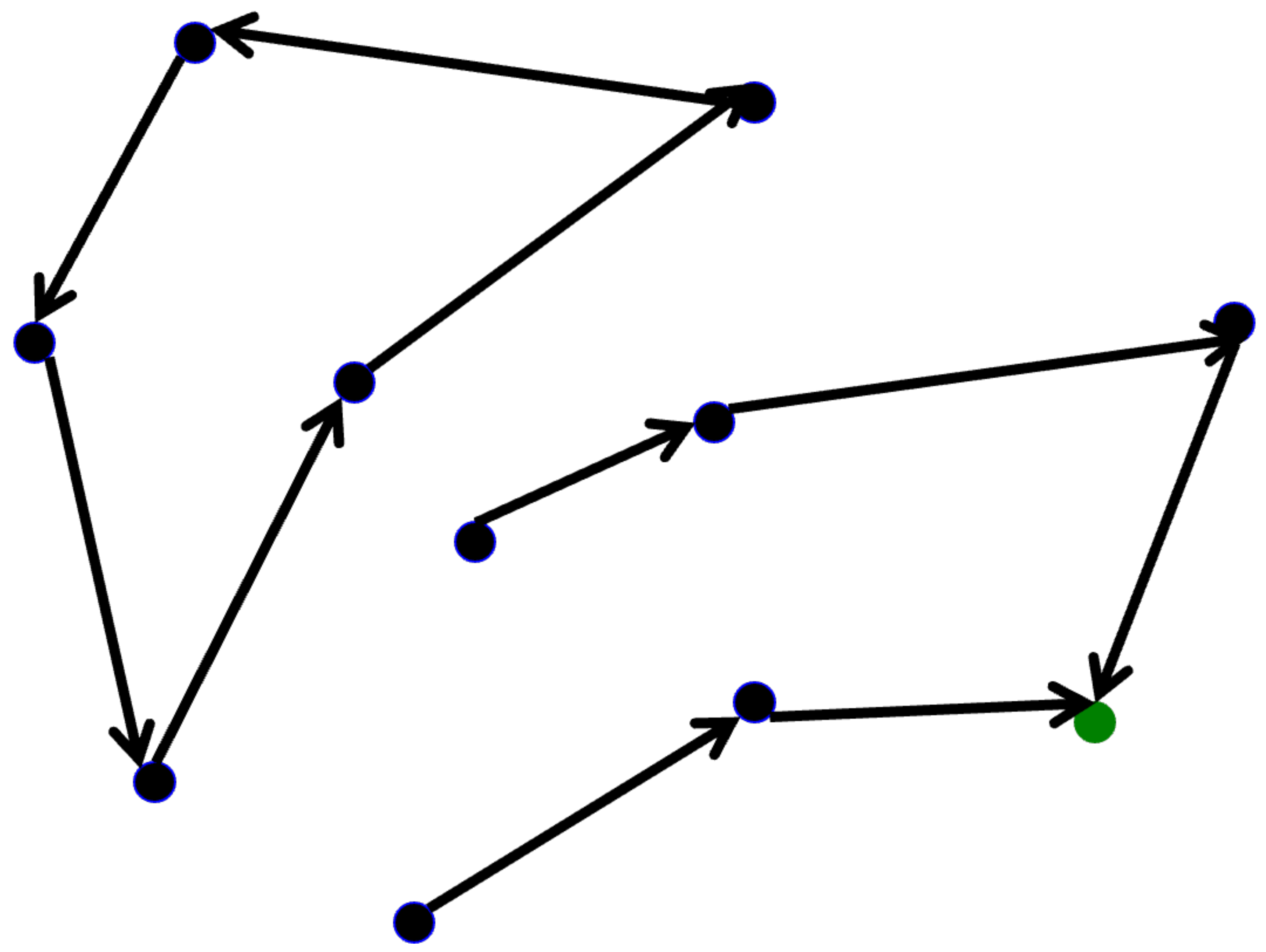
Example: remove unused links



Example: remove unused links



Example: remove unused links



Easy to check validity

1. Dead ends are obvious
    1. Nodes without outgoing links
2. Loops are obvious
    1. Disconnected from rest of the graph

Goals of routing

1. Goal 1: *valid* routing in the network
    1. How to know if the state of routers' routing tables is *valid*?
2. Goal 2: *efficient* routing in the network
    1. Finding a *least cost path* to a given destination

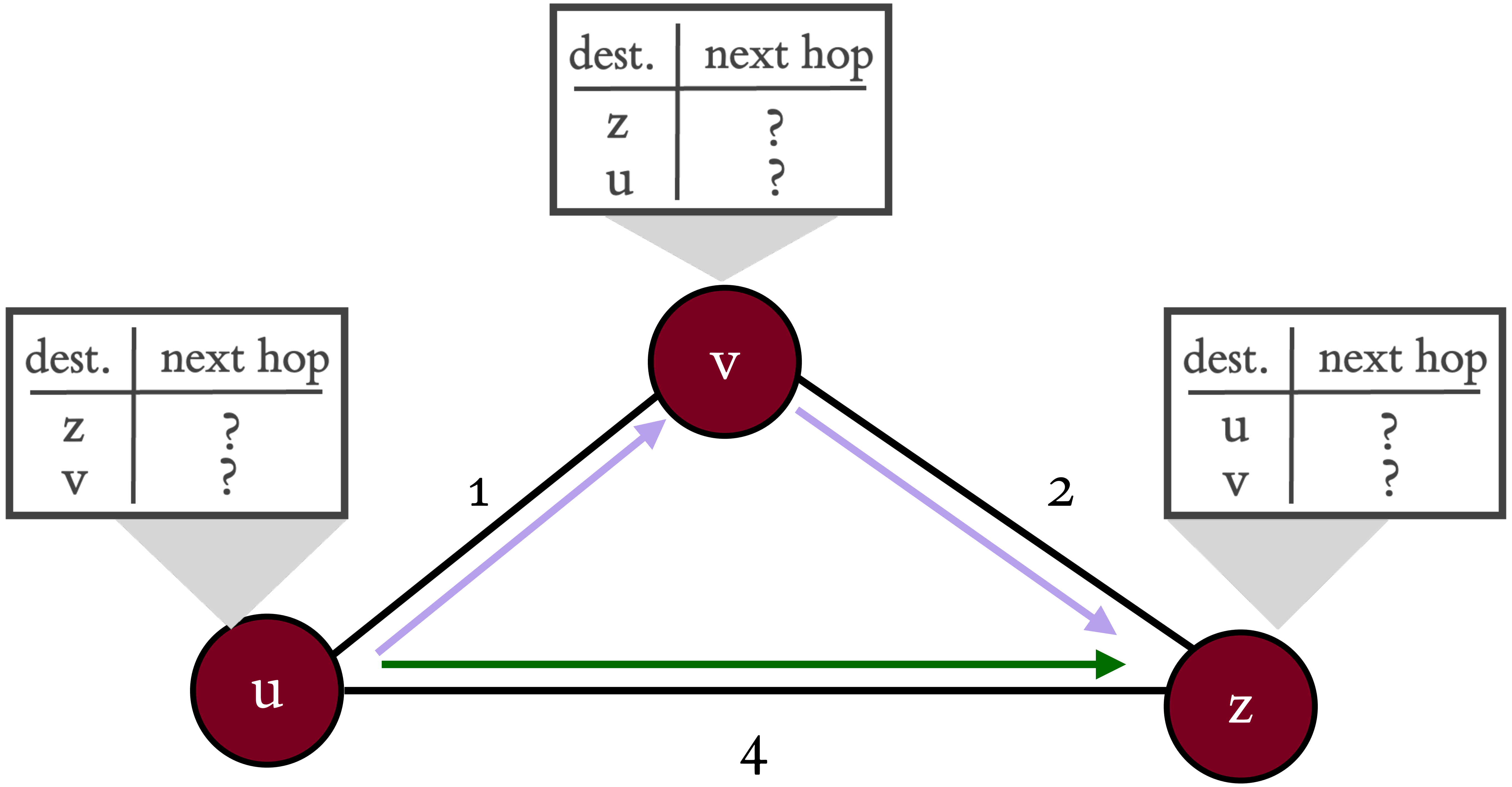
"Cost" in routing

1. Least cost routing tries to find paths with minimum X
2. What can X be?
   1. Latency
   2. Number of hops in the path
   3. Weight
   4. Failure probability
   5. …
3. Assume each link has some cost
4. We want to minimize the cost of paths
   1. Cost of a path = sum of the costs of links on the path

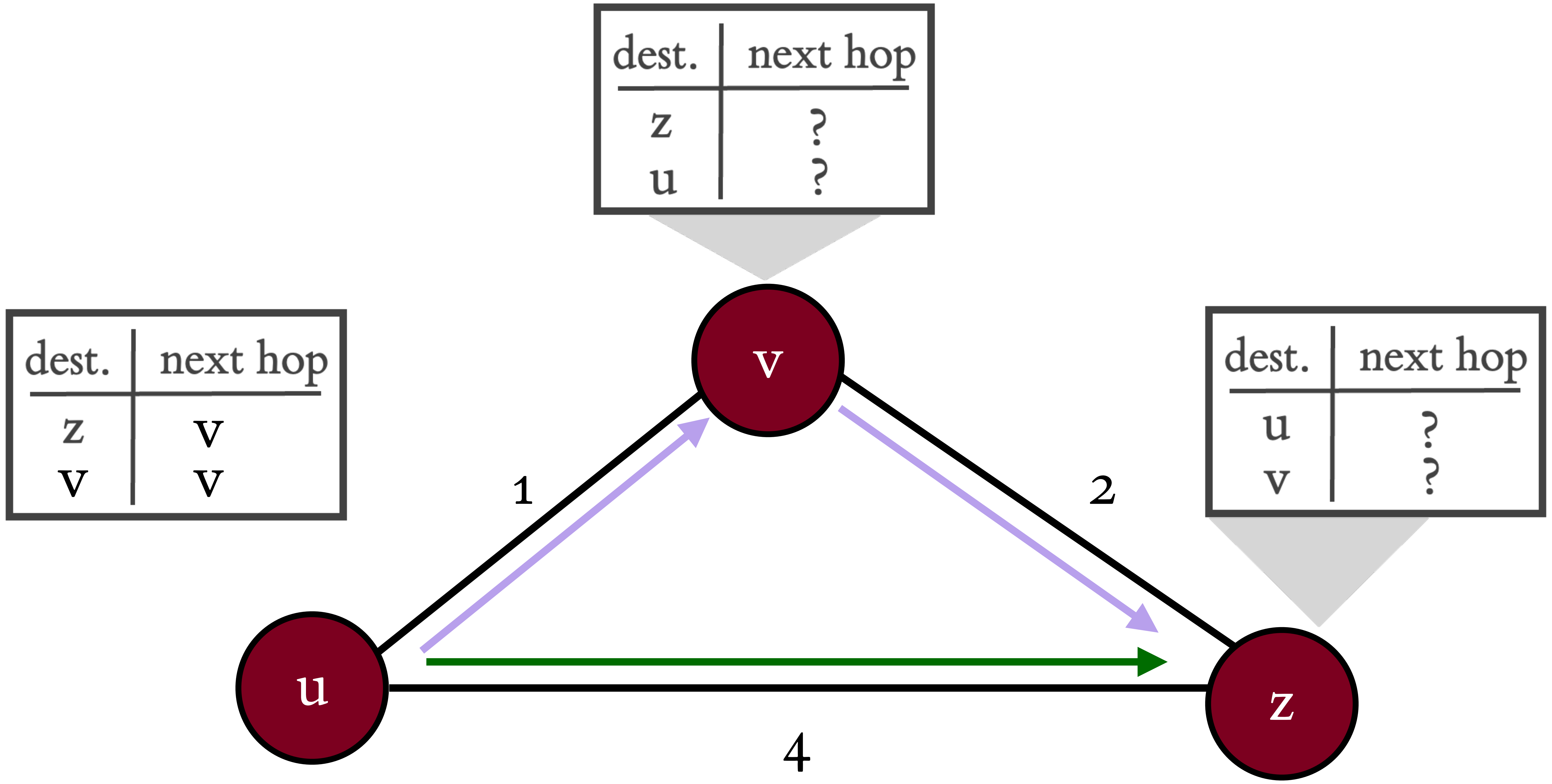
Least cost path routing

1. Approach 1: Link state routing
2. Approach 2: Distance vector routing

Least cost path routing



Least cost path routing



Least cost u—> z path: u—>v —>z

Least cost u—>v path: u->v

Least cost path routing

1. Given: router graph and link cost
2. Goal: find least cost paths
    1. From each source router
    2. To each destination router
3. How do you find least cost paths from a source to ALL destinations?
    1. Dijkstra's algorithm

Least cost routes

1. Least cost routes automatically avoid loops
    1. No sensible cost metric is minimized by traversing loops
    2. Least cost routes end up forming **spanning trees** to the destination

Link state routing: protocol vs. algorithm

1. Link state routing protocol **creates a global view** of the network
    1. Where to create the global view?
    2. How to create the global view?
    3. When to run route computation?
2. Algorithm **finds shortest paths** on the global network view
    1. Create shortest paths using standard algorithms