UNIVERSITY OF WATERLOO

CS 456/656
Computer Networks

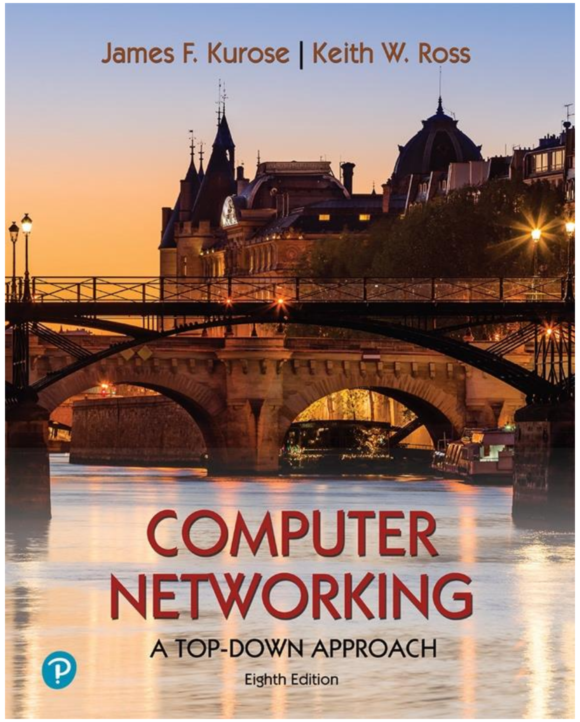
Lecture 9: Network Layer – Part 1

Mina Tahmasbi Arashloo and Uzma Maroof

Fall 2025

A note on the slides

Adapted from the slides that accompany this book.



Computer Networking: A Top-Down Approach
8th edition
Jim Kurose, Keith Ross
Pearson, 2020

Network layer: roadmap

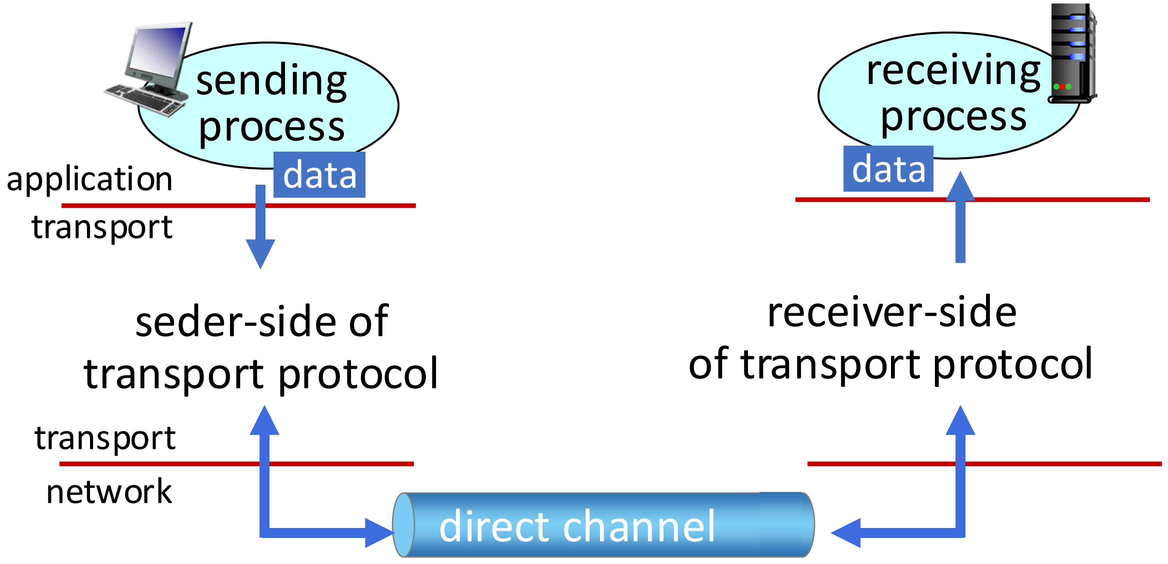
- Network layer overview
- Routing algorithms
- Network layer in the Internet

Network layer: roadmap

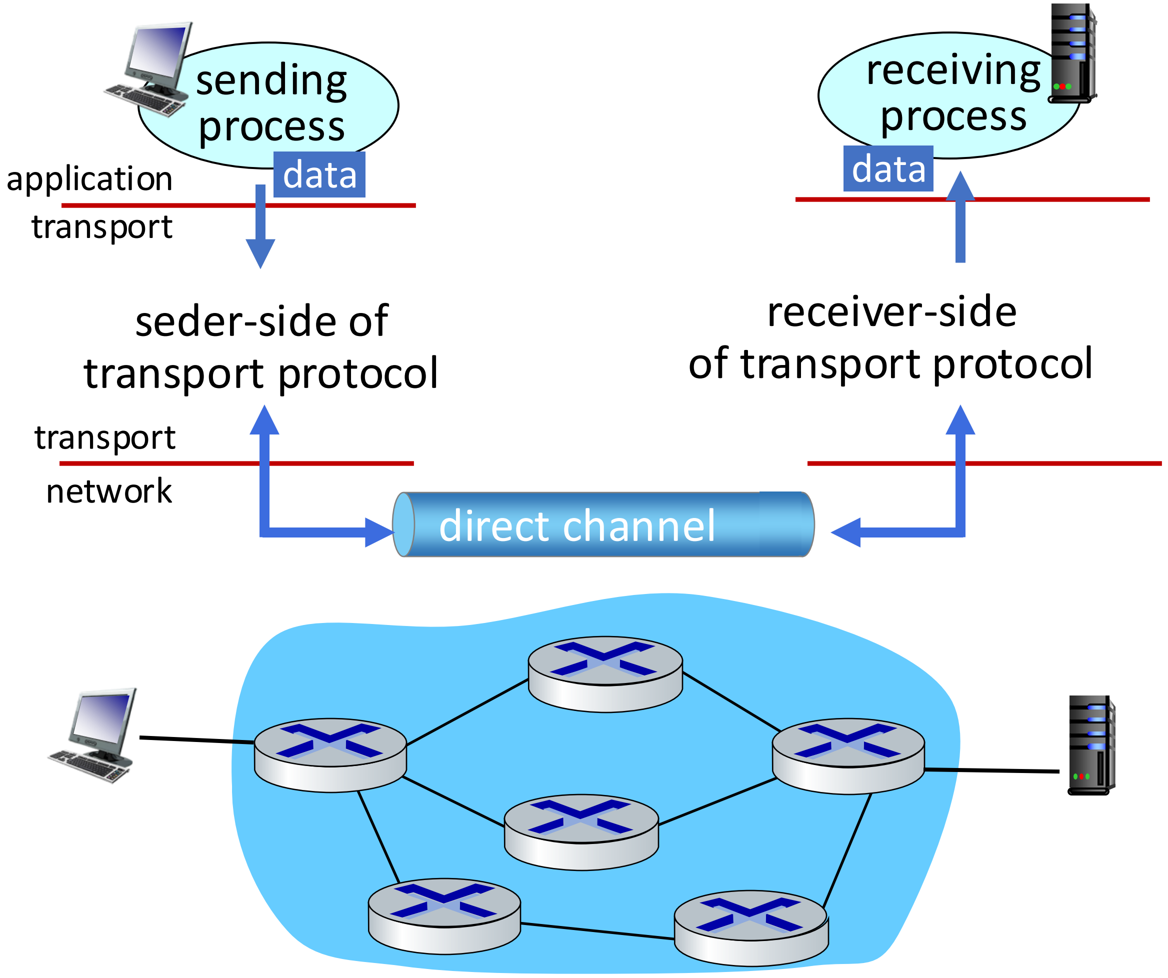
- Network layer overview
  - Network-layer service
  - Addressing, routing, and forwarding
- Routing algorithms
- Network layer in the Internet

Network-layer services

- The transport layer does not concern itself with how data gets from the sending host to the receiving host.
- It assumes there is a direct channel between the two.
- But in reality, there is rarely a direct channel between every pair of hosts in a network…
  - Q: why?

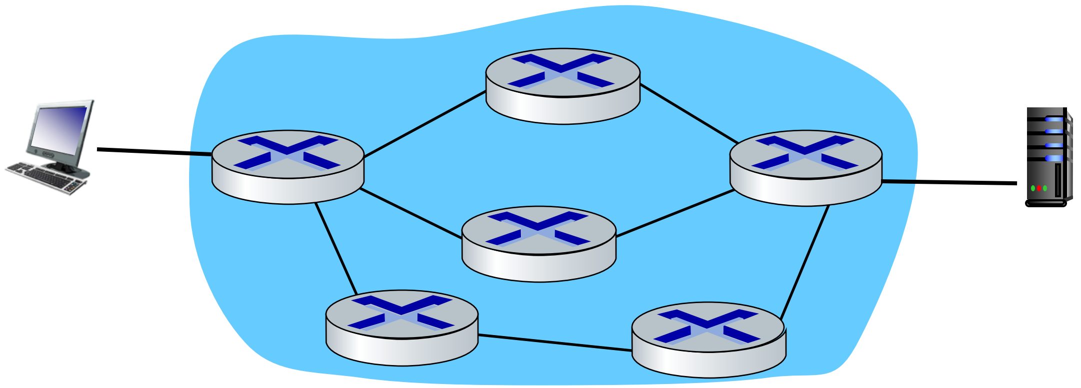


Network-layer services



Network-layer services

- Provide the “illusion” of a direct channel between the sending and receiving host to the transport layer over a shared network.
- On the end hosts:
  - Sender encapsulates transport segments into datagrams and passes them to the link layer.
  - Receiver extracts segments from datagrams and delivers them to the transport layer.
- Through the network:
  - All routers run network layer protocols and transmit datagrams from source to destination



Network-layer channel guarantees

Q: What kinds of guarantees can this "channel" provide?

example for *individual* datagrams:

- guaranteed delivery
- guaranteed delivery with less than 40 msec delay

example for a *flow* of datagrams:

(e.g., all datagrams belonging to the same transport-layer connection)

- in-order datagram delivery
- guaranteed minimum bandwidth to flow
- restrictions on changes in inter-packet spacing

Example of channel guarantees

Network Architecture	Network-Layer Service Model	Quality of Service (QoS) Guarantees ?			
		Bandwidth	No loss	Order	Timing
Internet	best effort	none	no	no	no

Internet "best effort" service model

No guarantees on*:*

i. successful datagram delivery to destination
ii. timing or order of delivery
iii. bandwidth available to end-end flow

Example of channel guarantees

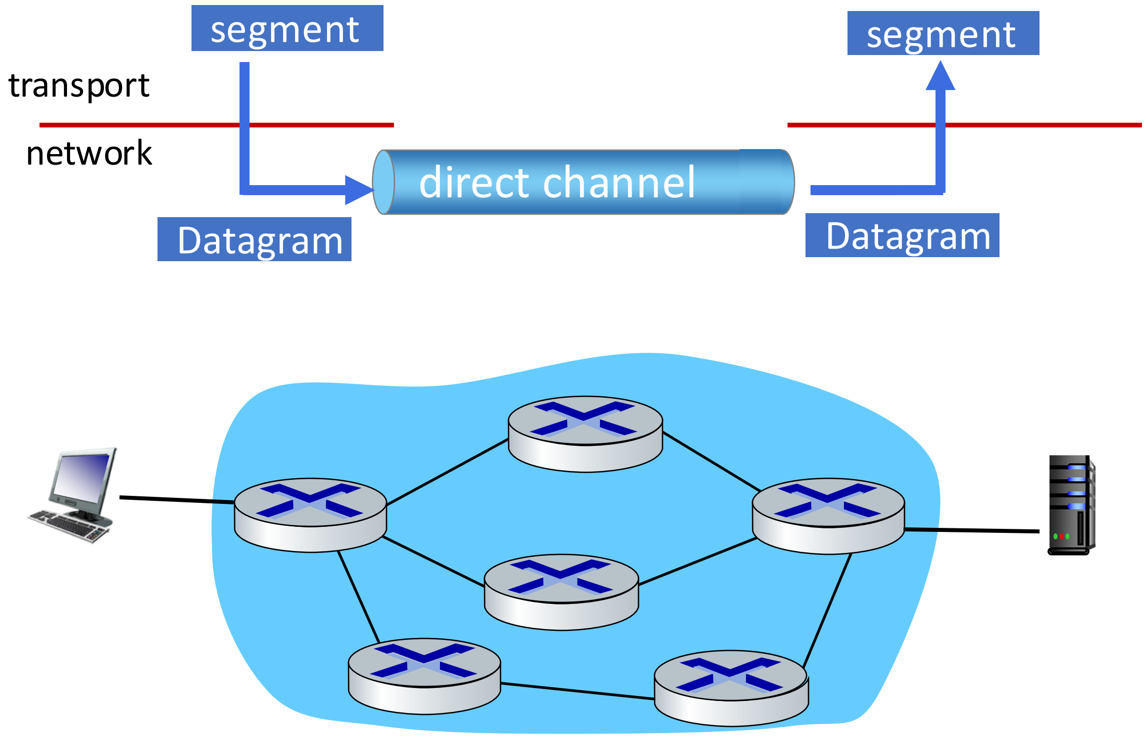
Network Architecture	Network-Layer Service Model	Quality of Service (QoS) Guarantees ?			
		Bandwidth	No loss	Order	Timing
Internet	best effort	none	no	no	no
ATM	Constant Bit Rate	Constant rate	yes	yes	yes
ATM	Available Bit Rate	Guaranteed min	no	yes	no
Internet	Intserv Guaranteed (RFC 1633)	yes	yes	yes	yes
Internet	Diffserv (RFC 2475)	possible	possibly	possibly	no

Network layer: roadmap

- Network layer overview
  - <u>Network-layer service</u>
  - <u>Addressing, routing, and forwarding</u>
- Routing algorithms
- Network layer in the Internet

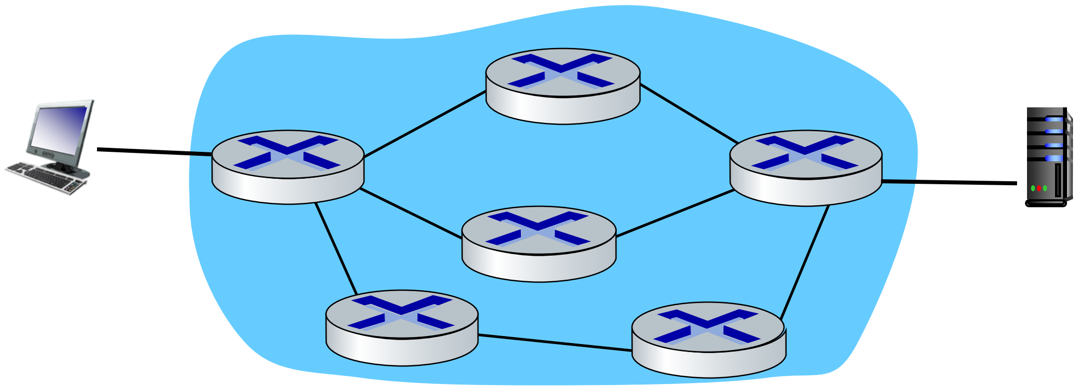
Network-layer services

- Provide the “illusion” of a direct channel between the sending and receiving host to the transport layer over a shared network.
- How?



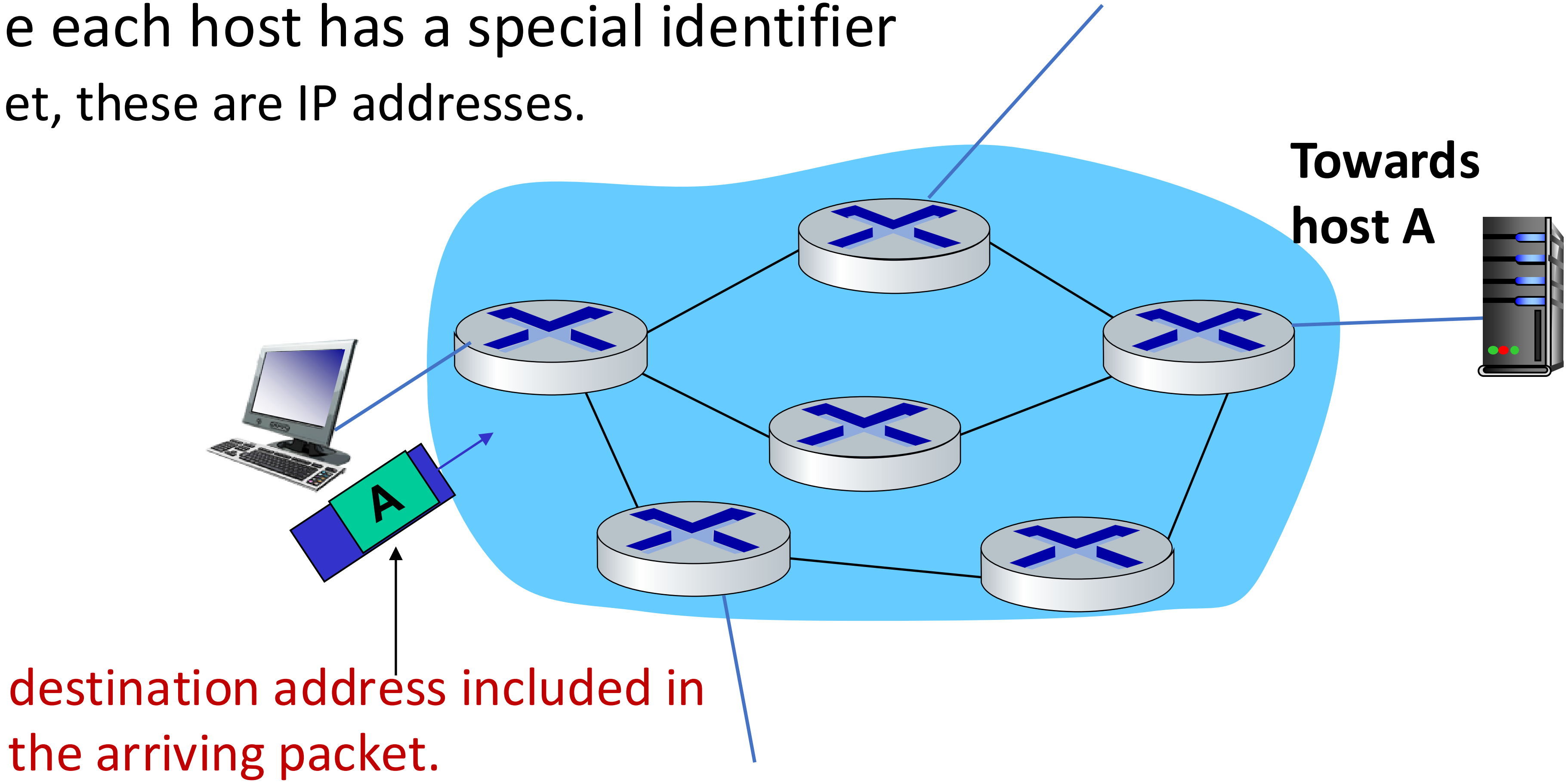
How does the network-layer "implement" a direct channel between a sender and receiver?

- Addressing:
  - Each host needs an *identifier* so the network can distinguish between traffic from/to different hosts.
- Routing:
  - Determine the "best" path from the source to the destination host
- Forwarding:
  - Move packet along the path, i.e., on each router, move packets from the input link to the appropriate output link



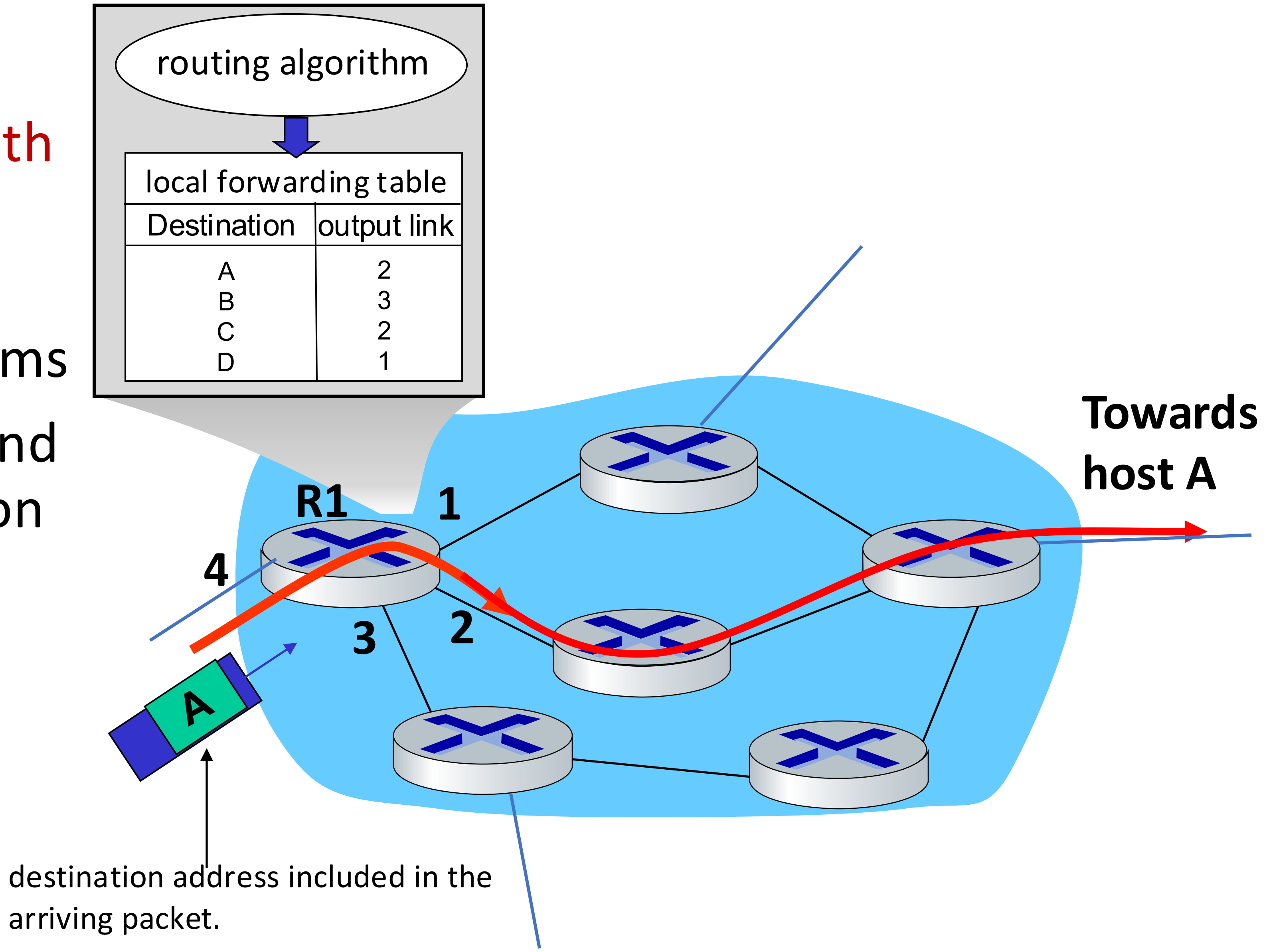
Addressing

- We will discuss addressing over the Internet in depth later.
- For now, assume each host has a special identifier
  - Over the Internet, these are IP addresses.



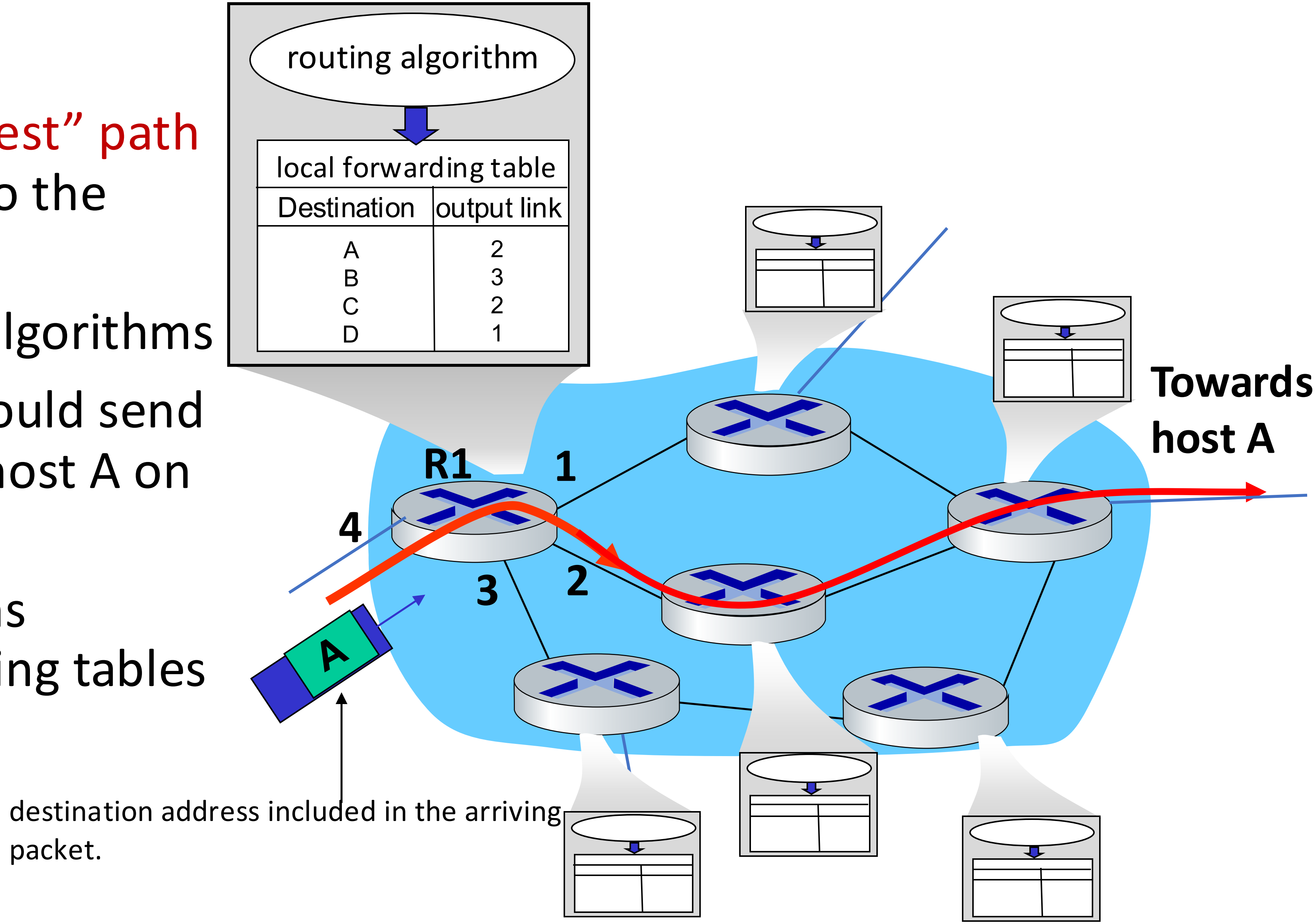
Routing

- Determine the "best" path from the source to the destination host
- Done by routing algorithms
- E.g., router R1 should send packets going to host A on its link #2.



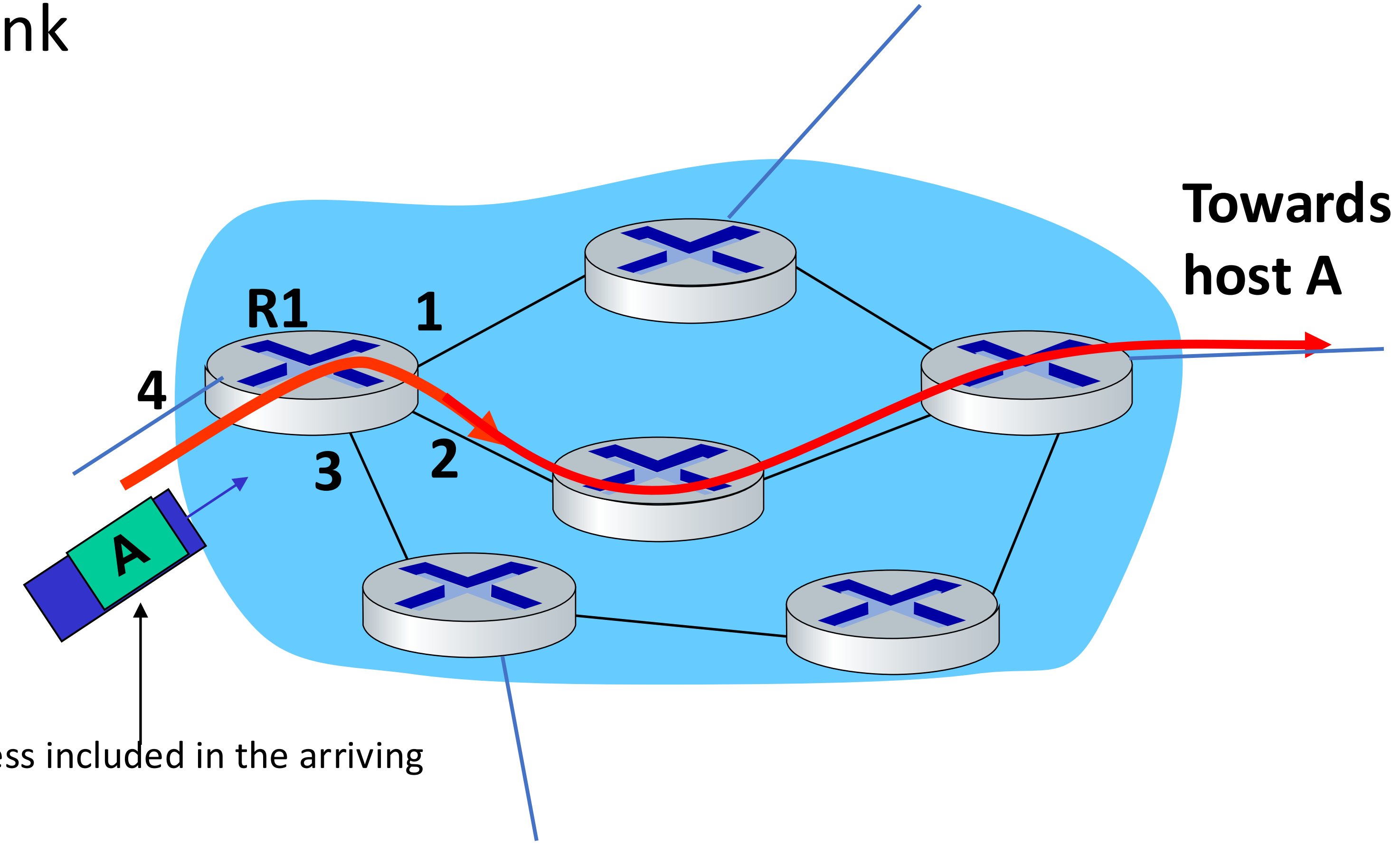
Routing

- Determine the "best" path from the source to the destination host
- Done by routing algorithms
- E.g., router R1 should send packets going to host A on its link #2.
- Routing algorithms populate forwarding tables



Forwarding

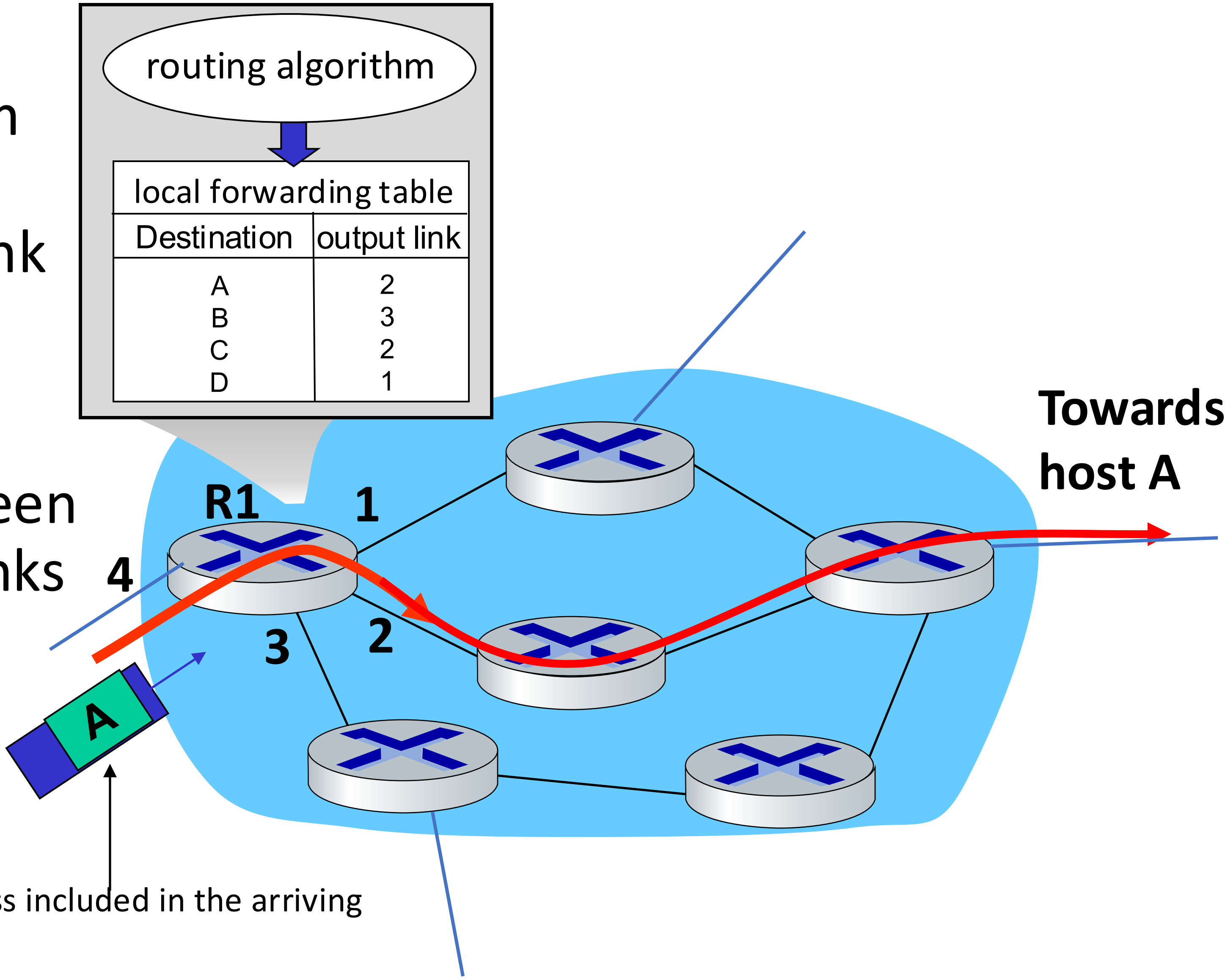
- Moving arriving packets from router's input link to appropriate router output link



destination address included in the arriving packet.

Forwarding

- Moving arriving packets from router's input link to appropriate router output link
- Based on information in the forwarding table
- Happens concurrently between multiple input and output links at very high speed!
  - Millions of packets per second



Routing vs. Forwarding

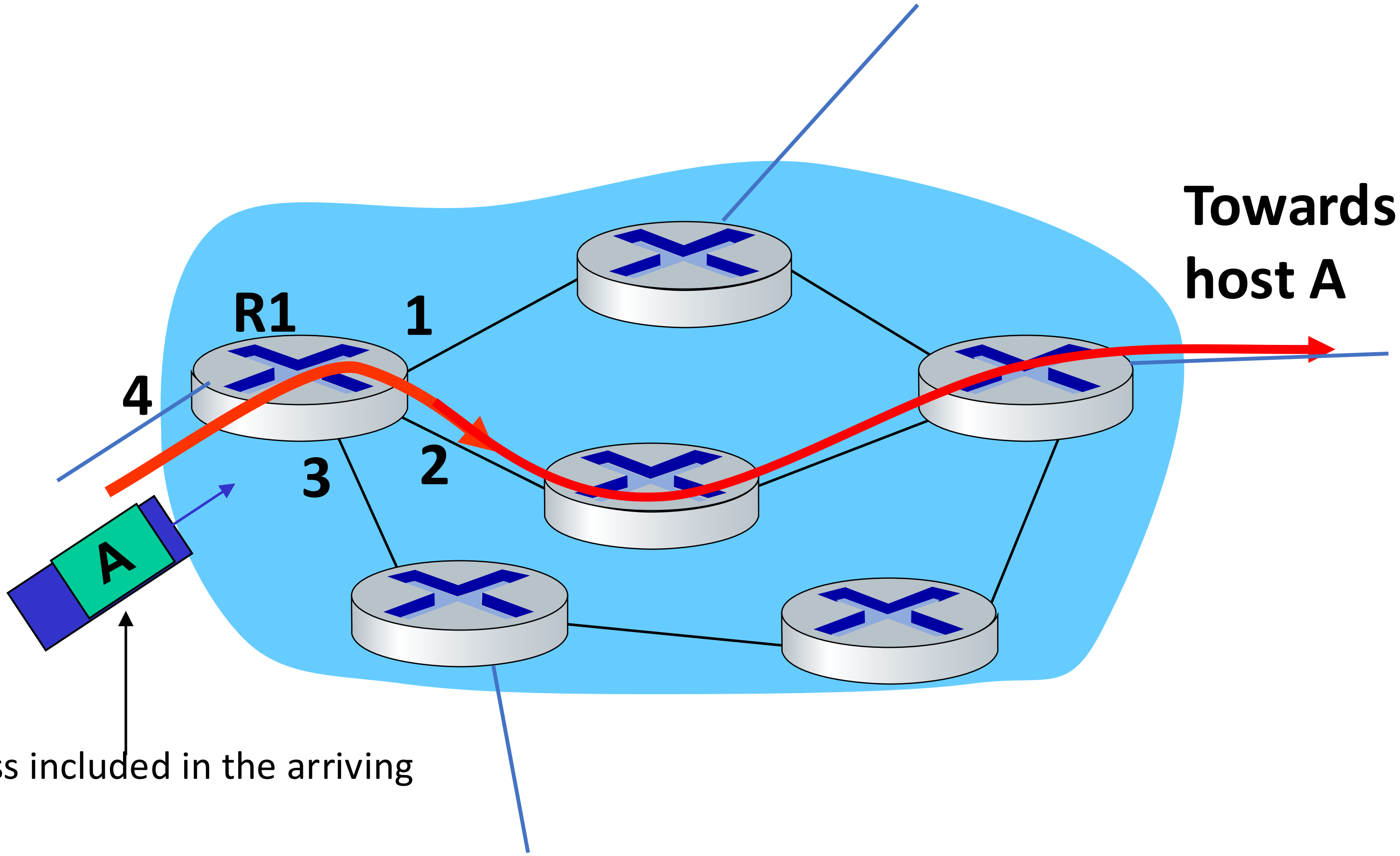
Routing: Global action

- Determining paths to get packets to their destination
- Populate forwarding tables of routers/switches

Forwarding:

- Moving arriving packets from router's input link to appropriate router output link

Local action



destination address included in the arriving packet.

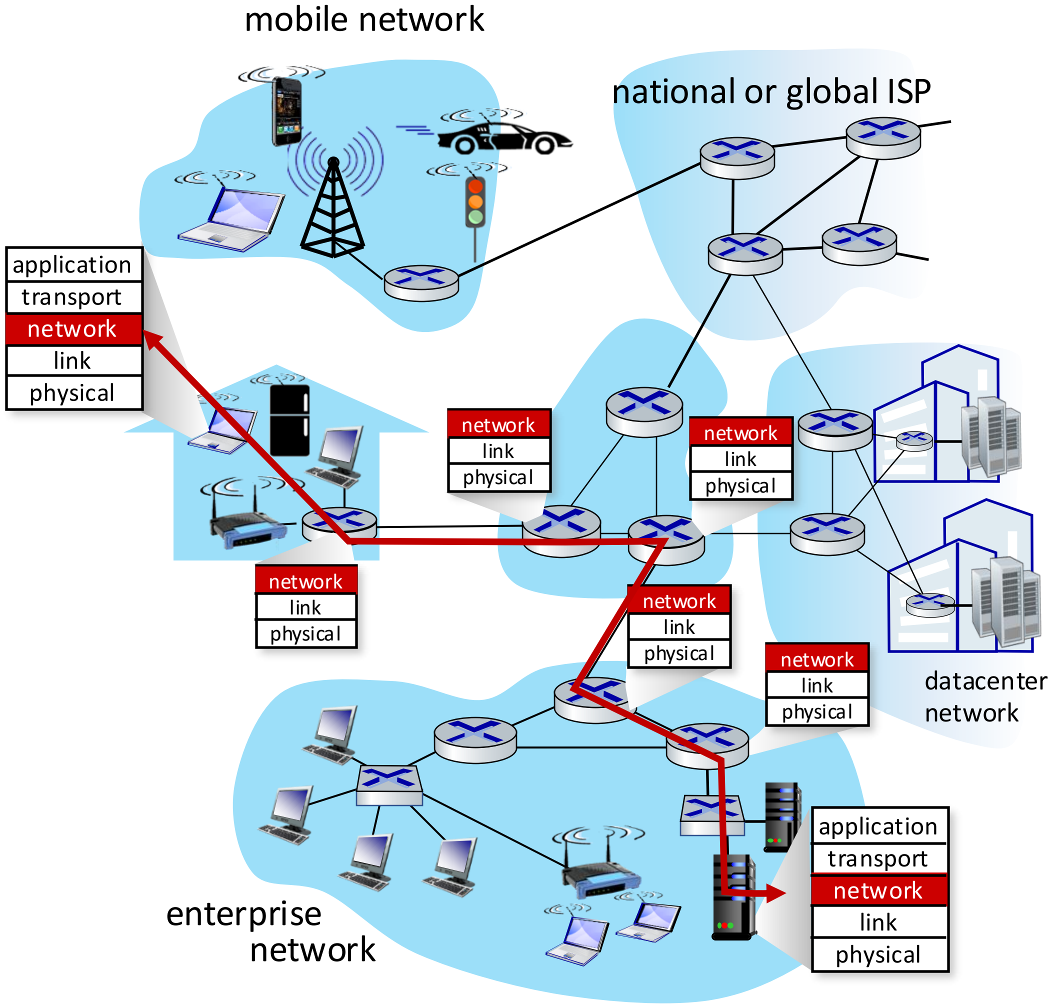
Network layer: roadmap

- Network layer overview
- Routing algorithms
  - Link state
  - Distance vector
- Network layer in the Internet

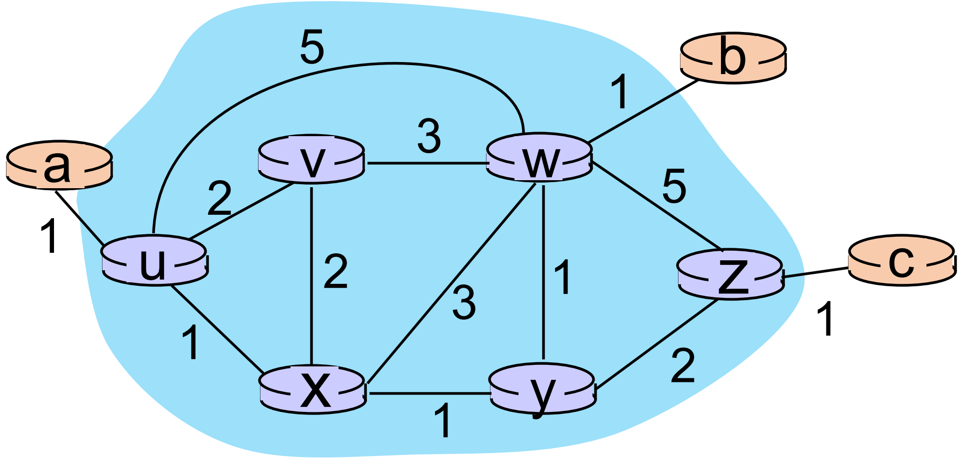
Routing protocols

Routing protocol goal: determine "good" paths (equivalently, routes), from sending hosts to receiving host, through network of routers

- path: sequence of routers packets traverse from given initial source host to final destination host
- "good": least "cost", "fastest", "least congested"
- routing: a "top-10" networking challenge!



Graph abstraction: link costs



$c_{i,j}$: cost of *direct* link connecting i and j

e.g., $c_{w,z} = 5$, $c_{u,z} = \infty$

cost defined by network operator: could always be 1, or inversely related to bandwidth, or related to congestion

graph: $G = (N,E)$

N: set of routers or "end points" connected to a router

$\{ u, v, w, x, y, z, a, b, c \}$

E: set of links =

$\{ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z), \mathrm{(u,a), (w,b), (z,c)} \}$

Routing algorithms approaches

- link state: all routers have complete topology and link cost info
- Distance vector:
  - Routers only know about the state of their attached neighbors
  - Gets updated through an iterative process of computation and exchanging info with neighbors

Network layer: roadmap

- Network layer overview
- Routing algorithms
  - Link state
  - Distance vector
- Network layer in the Internet

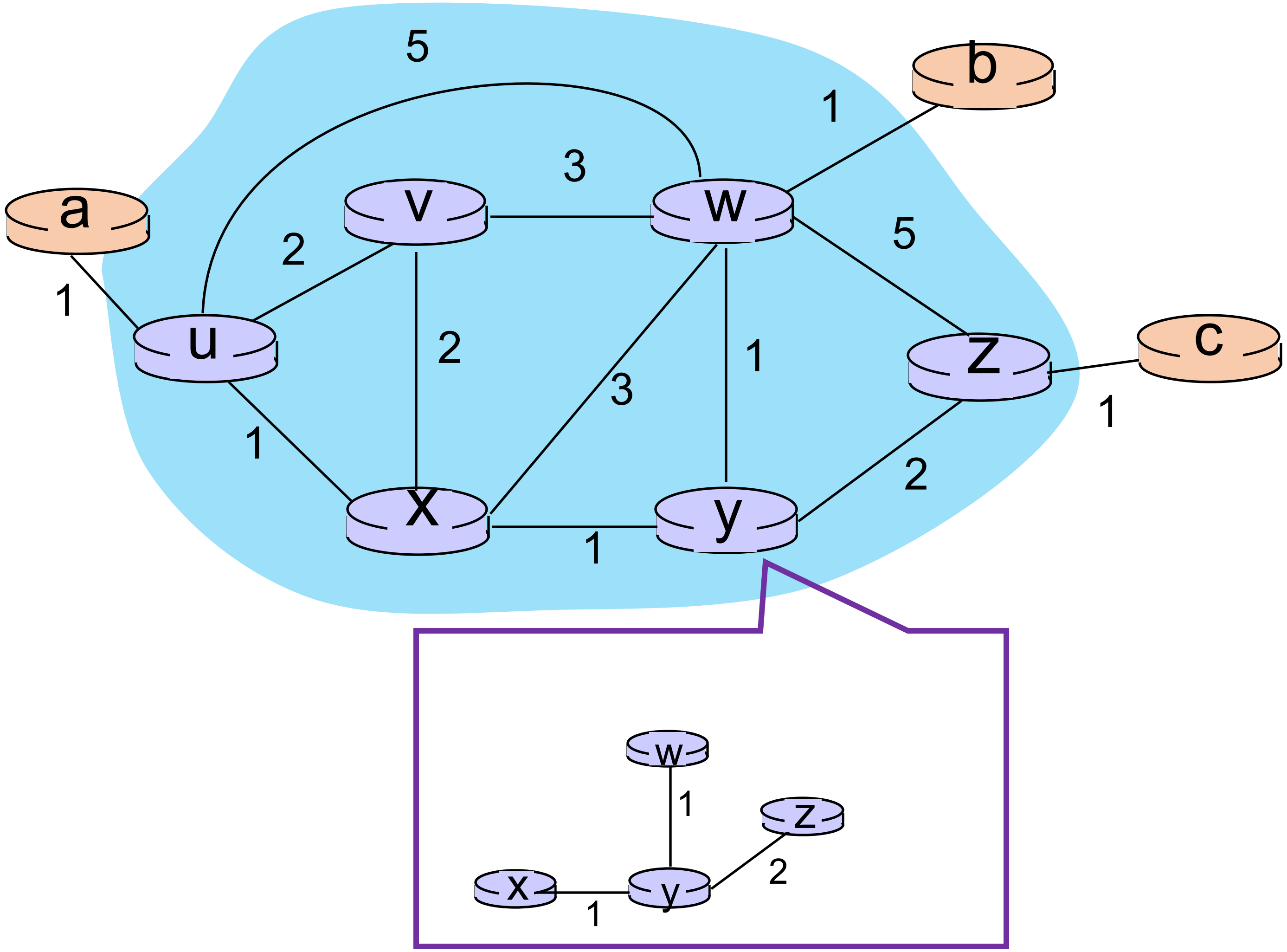
Link-state routing algorithms

- Two components
  - Information propagation
  - Path computation
- Information propagation
  - Each node lets the other nodes know about what it is *directly* connected to and at what cost
  - This way, each node can build a picture of what the network graph looks like
- Path computation
  - Each node can use its local view of the complete network graph to compute *the least-cost path* to each destination

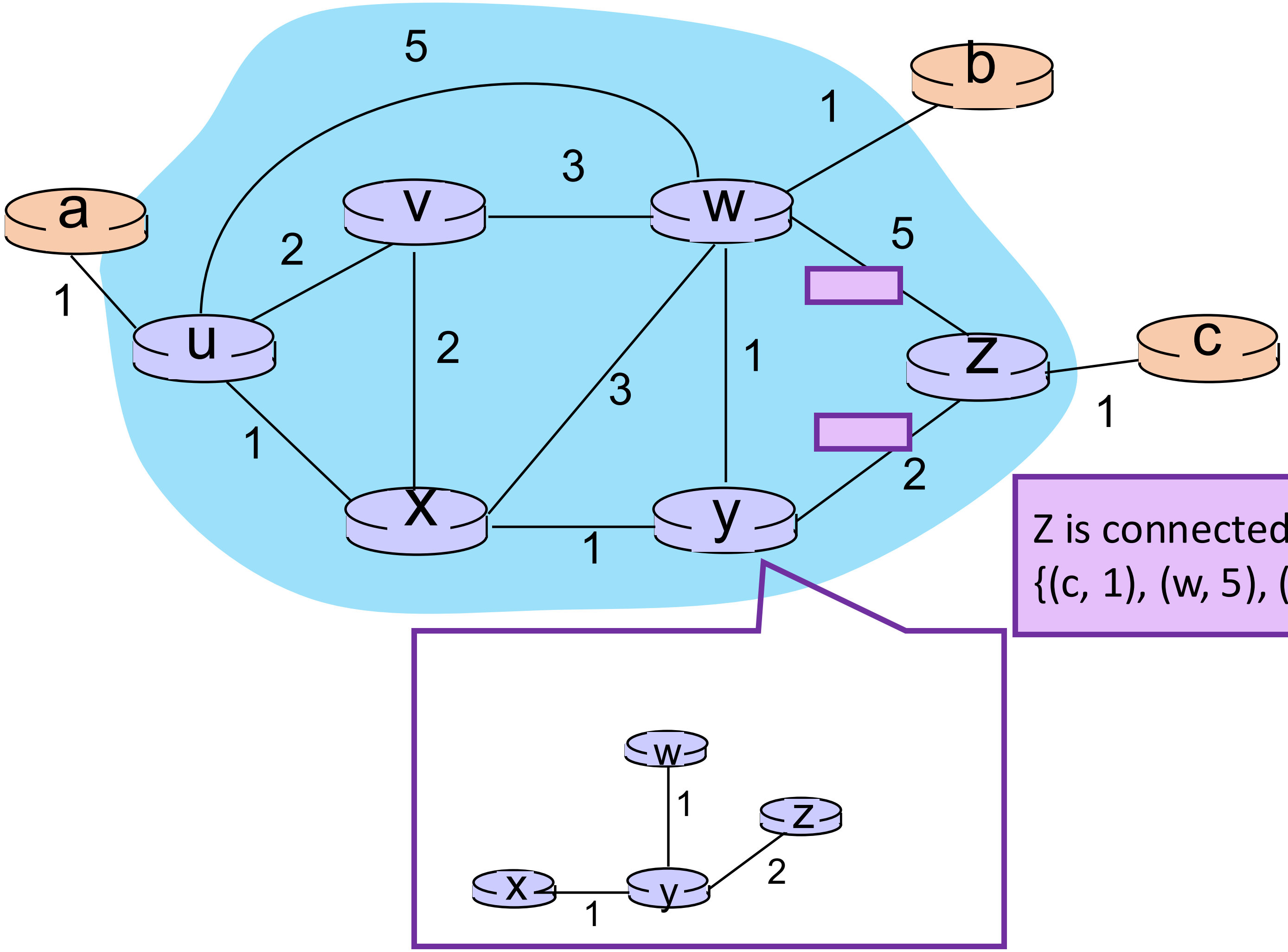
Link-state routing – Information Propagation

- Each router periodically generates a message including
  - It's "ID" (<u>not</u> the same as network-layer identifies such as IP addresses)
  - All the nodes (routers and sets of end hosts) it is directly connected to
  - The "cost" of the links connecting it to its direct neighbors
- Information is propagated to all router
  - The message is sent to the neighbors
  - The neighbors will send it to their neighbors
  - …
- Eventually, all routers will have the same info
  - Topology, link costs, which routers are connected to which end hosts

Link-state routing – Information Propagation

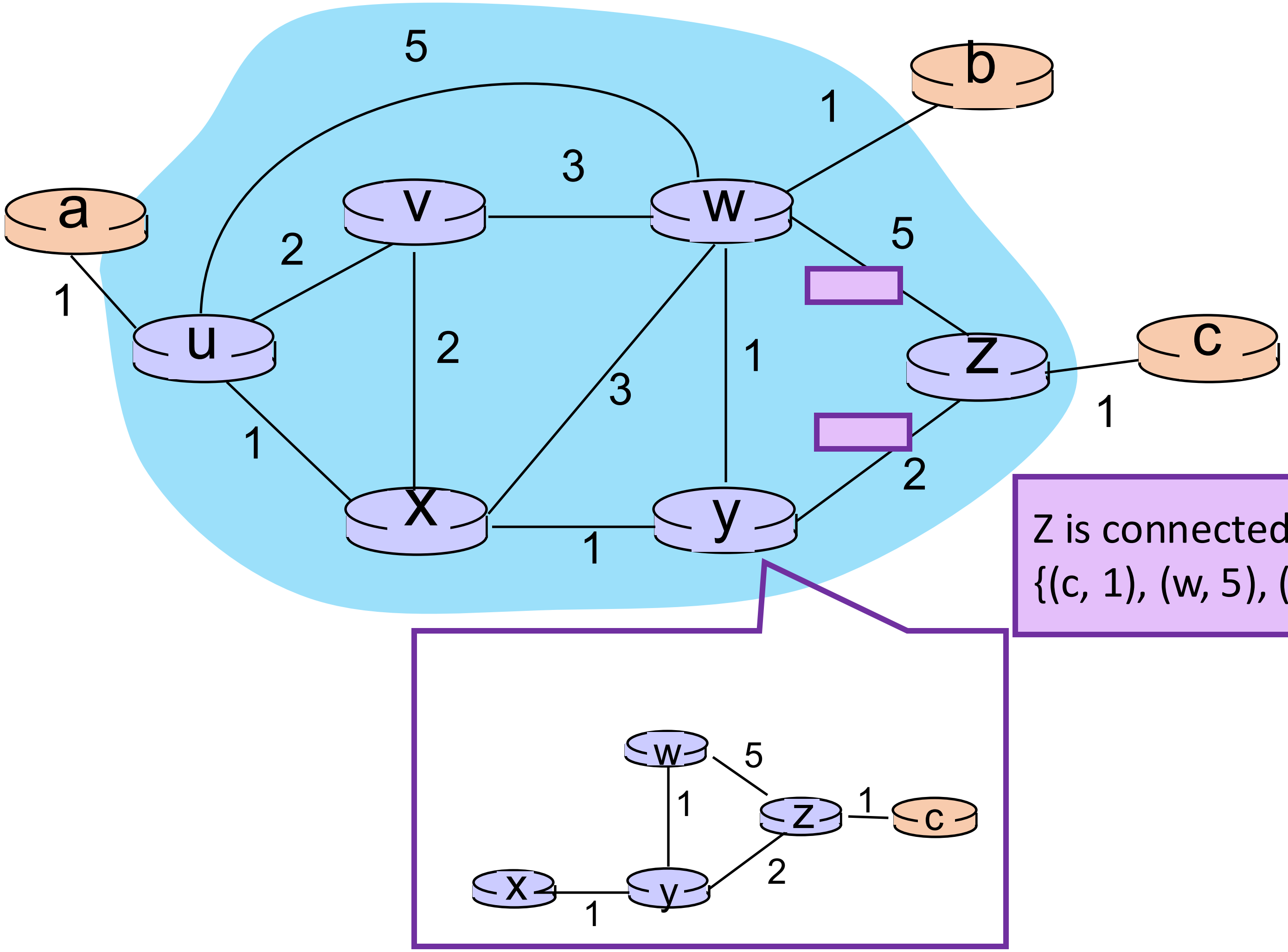


Link-state routing – Information Propagation



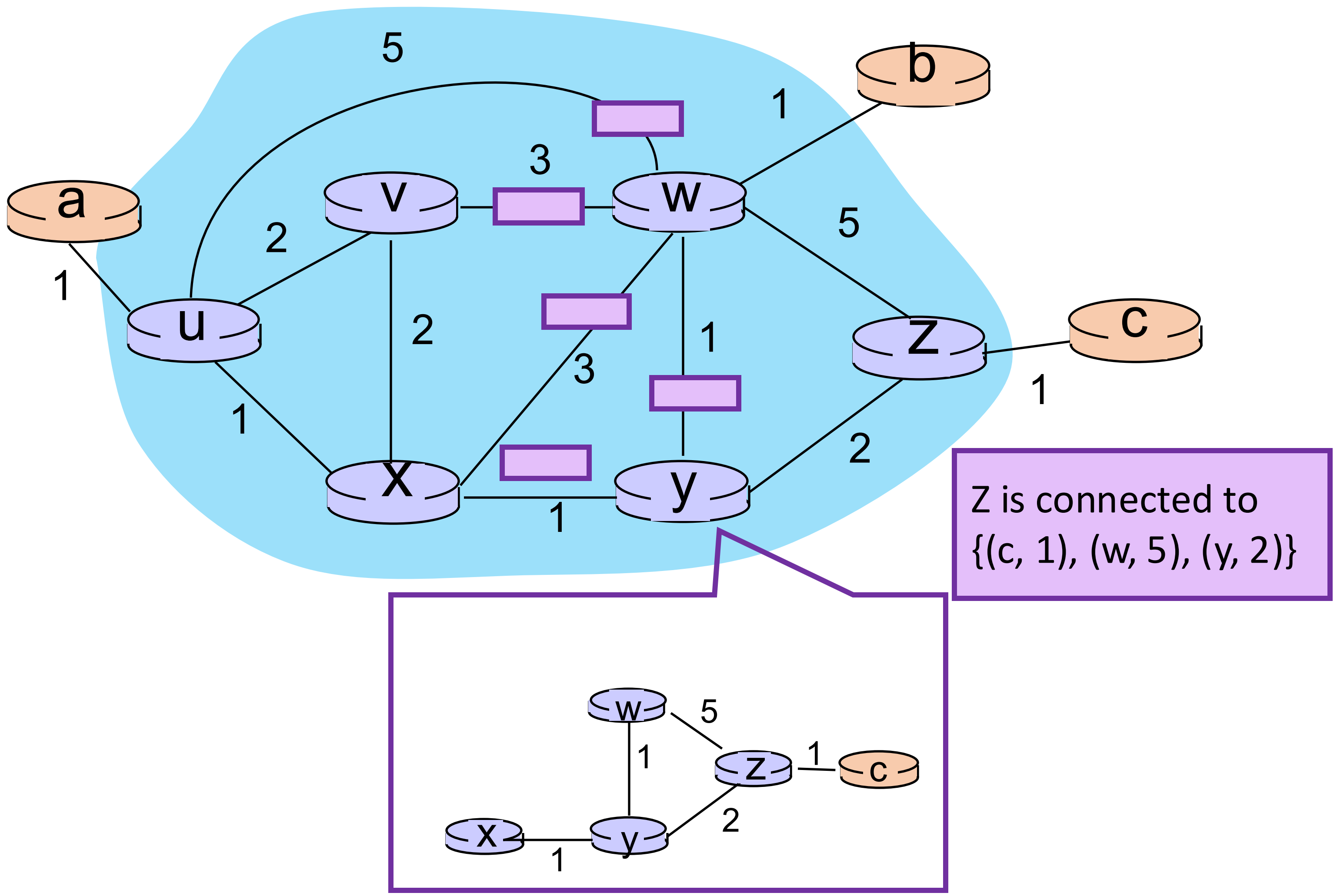
Z is connected to
{(c, 1), (w, 5), (y, 2)}

Link-state routing – Information Propagation



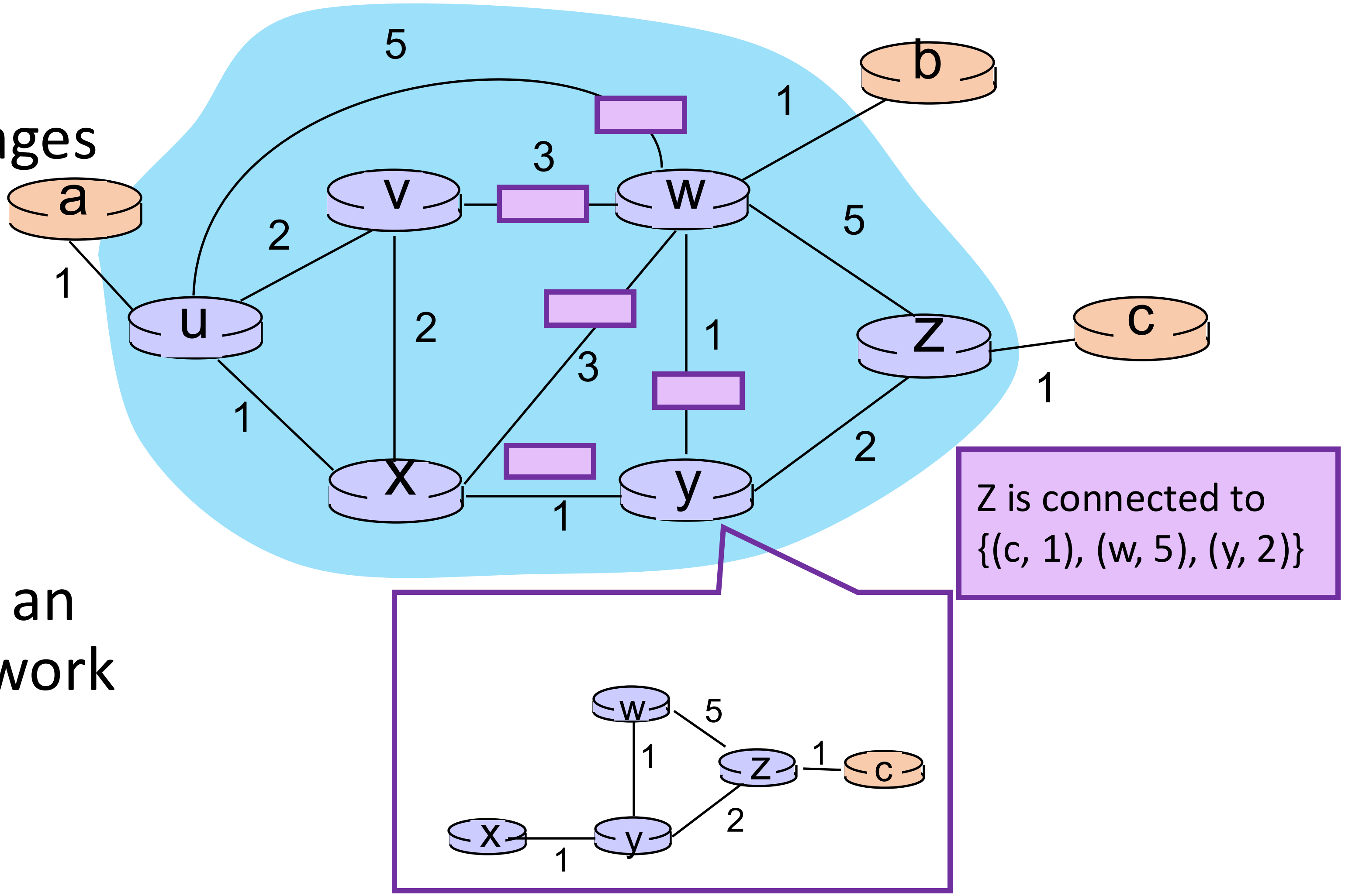
Z is connected to
{(c, 1), (w, 5), (y, 2)}

Link-state routing – Information Propagation



Link-state routing – Information Propagation

- This continuously happens across all nodes…
- Each node generates messages
  - Both periodically and when something changes
- Each node forwards the updates it receives to its neighbors
- So, each node finally builds an accurate picture of the network



Link-state routing – Path computation

- Each router computes least cost paths from itself to all other nodes
  - So, it will know the least-cost path to reach each end host
  - When it gets a datagram destined to a certain end host, it knows which link it should *forward* the datagram to.
- Since every router will (eventually) have the same local version of the topology, the path they pick independently will (eventually) be consistent with each other.
- How does the router compute the least-cost path?
  - Well-known graph algorithms
  - Classic approach, Dijkstra's algorithm (covered in your algorithms course)

Computing least-cost paths: Dijkstra's algorithm

- Runs on each router
- Computes least-cost paths from that router to all other nodes
- iterative: after k iterations, know least cost path to k destinations
- It iterates until it knows least-cost paths to all destinations.

notation

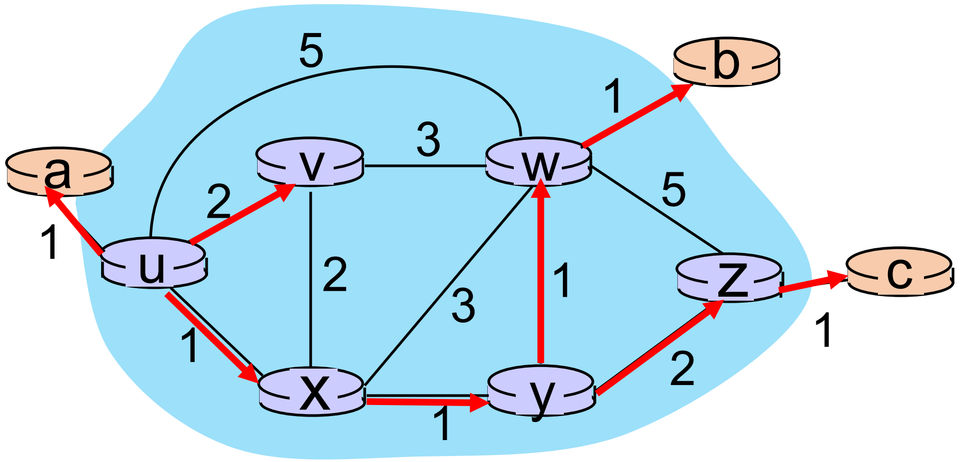
- $c_{x,y}$: direct link cost from node x to y; $= \infty$ if not direct neighbors
- $D(v)$: *current* estimate of cost of least-cost-path from source to destination v
- N': set of nodes whose least-cost-path *definitively* known

Computing least-cost paths: Dijkstra's algorithm

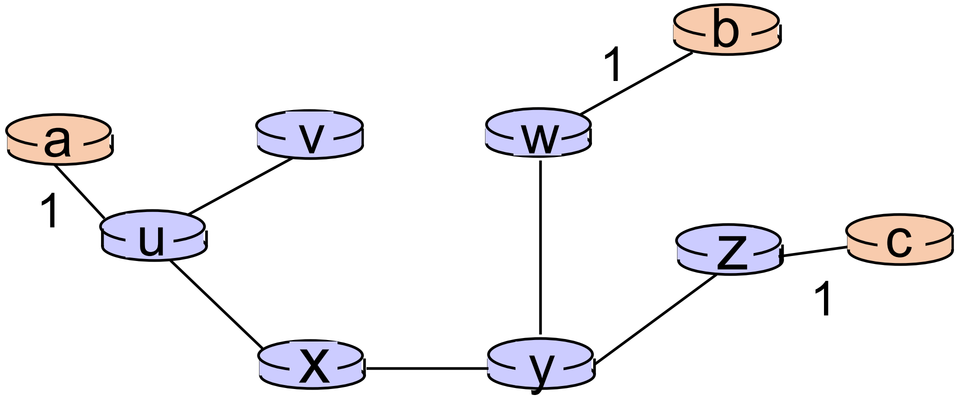
Initialization:
  $N' = \{u\}$ /* <u>router u computes least cost path from itself to all other nodes</u> */
  for all nodes v
    if v adjacent to u /* u initially knows direct-path-cost only to direct neighbors */
      then $D(v) = c_{u,v}$ /* but may not be *minimum* cost! */
    else $D(v) = \infty$

Loop
  find a not in N' such that $D(a)$ is a minimum
  add a to N'
  update $D(b)$ for all b adjacent to a and not in N' :
    $D(b) = \min\left(D(b),\ D(a) + c_{a,b} \right)$
  /* least-path-cost to b either stays the same or is updated to
  the known least-cost-path to a plus direct-cost from a to b */
until all nodes in N'

Computing least-cost paths



resulting least-cost-path tree from u:



resulting forwarding table in u:

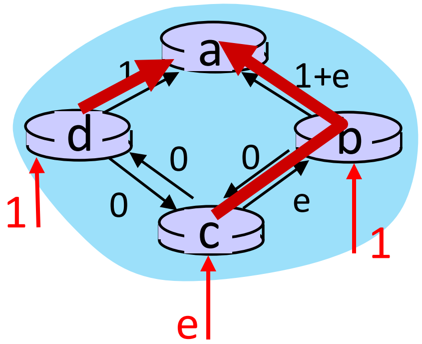
destination	outgoing link
a	(u,a)
b	(u,x)
c	(u,x)
…	…

route from *u* to *a* directly

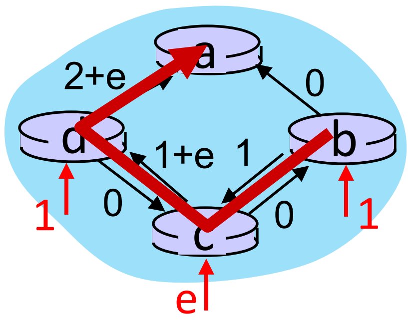
route from u to all other destinations via *x*

Routing based on least-cost: oscillations possible

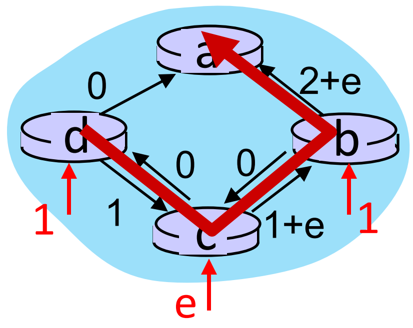
- when link costs depend on traffic volume, route oscillations possible
- sample scenario:
  - routing to destination a, traffic entering at d, c, b with rates 1, e (<1), 1
  - link costs are directional, and volume-dependent



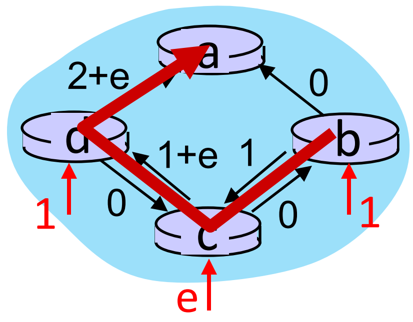
initially



given these costs,
find new routing….
resulting in new costs



given these costs,
find new routing….
resulting in new costs



given these costs,
find new routing….
resulting in new costs

What you need to know about Link State Routing algorithms

- How they work, i.e.,
  - How routers disseminate information about routes
  - How each router builds a full picture of the network
  - How they compute paths to destinations