CS 352
Network: ICMP, NAT, Routing

Lecture 22

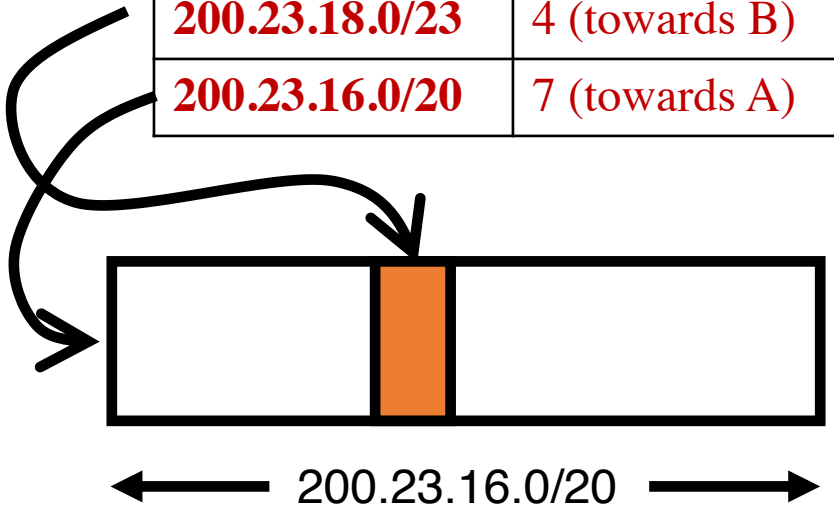
http://www.cs.rutgers.edu/~sn624/352-F22

Srinivas Narayana

RUTGERS
UNIVERSITY | NEW BRUNSWICK

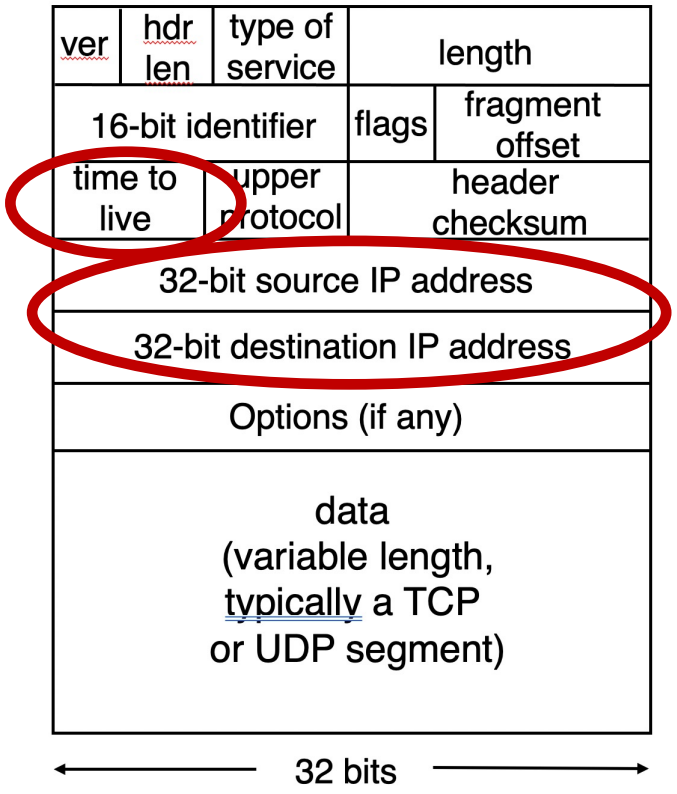
Review

Dst IP Prefix	Output port
65.0.0.0/8	3
128.9.0.0/16	1
200.23.18.0/23	4 (towards B)
200.23.16.0/20	7 (towards A)



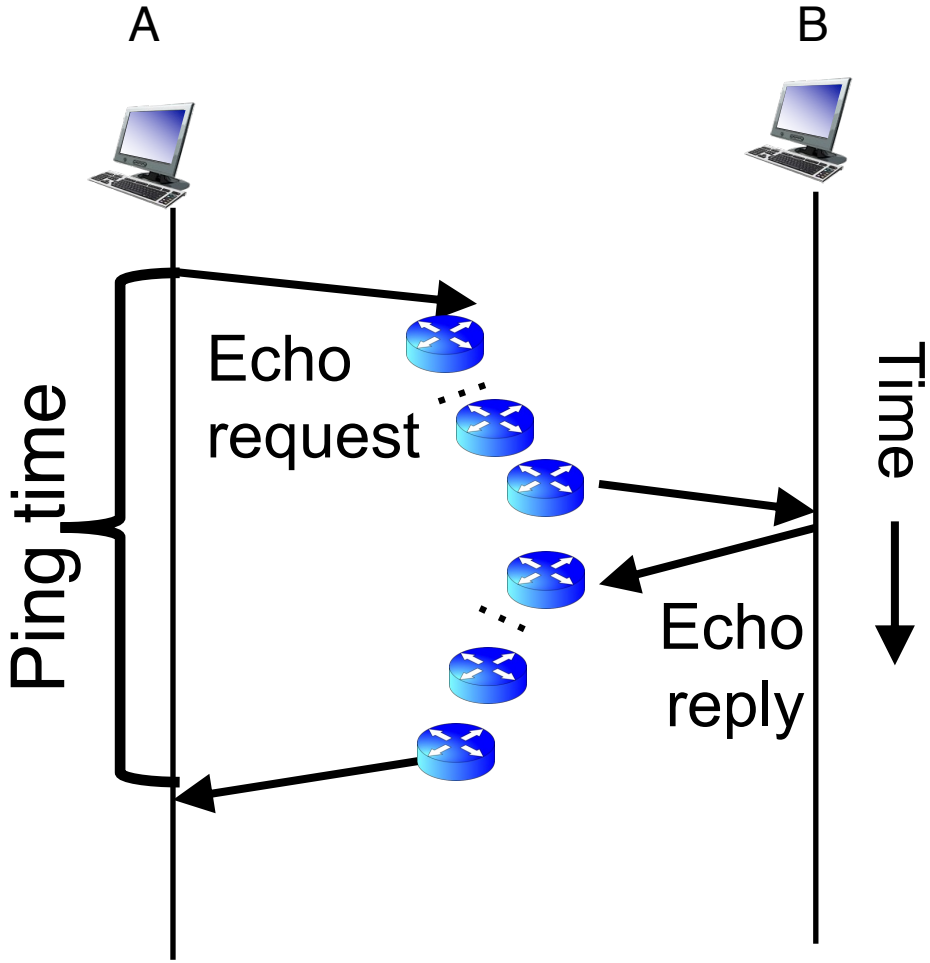
Longest Prefix Matching

IPv4 datagram



ICMP echo request and reply.

`ping`



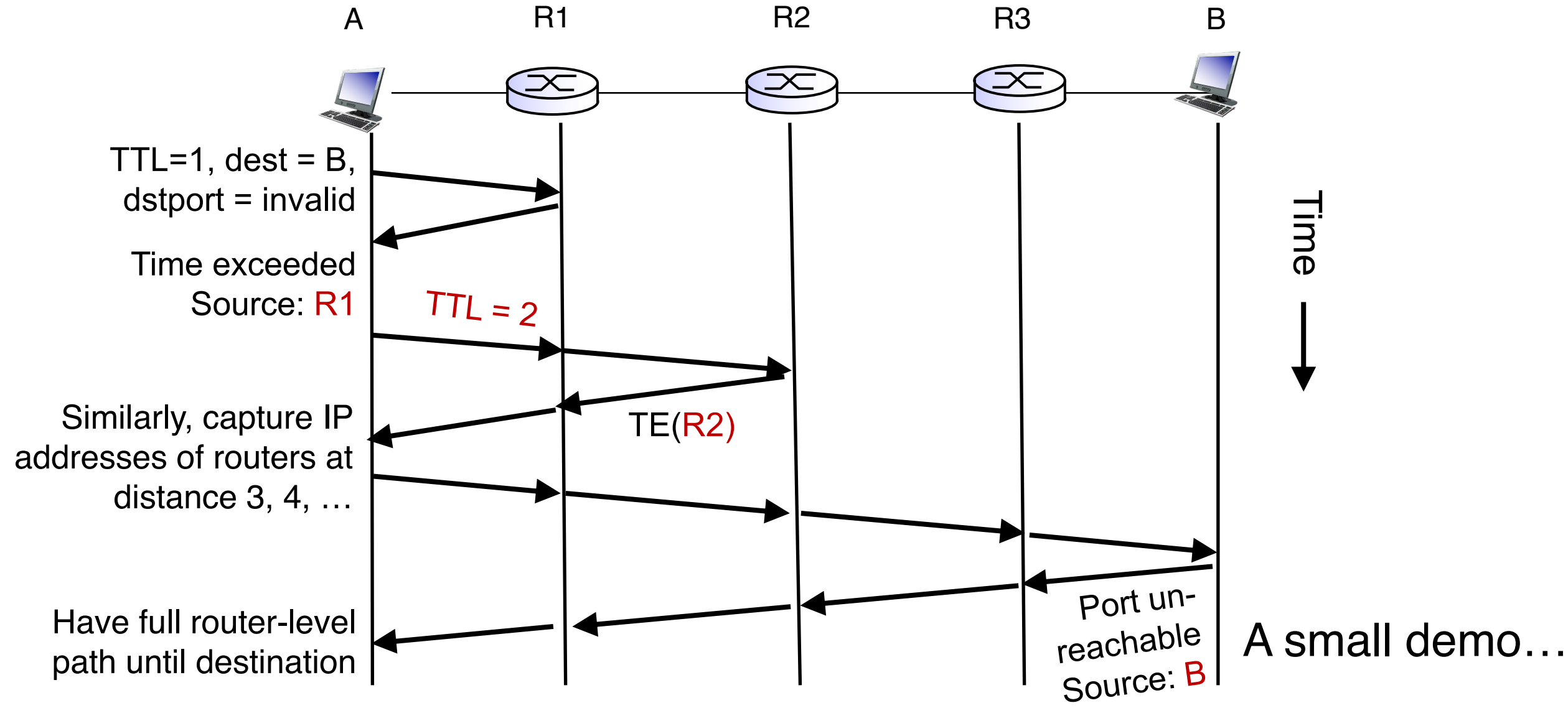
Traceroute

- A tool that can record the router-level path taken by packets
- A clever use of the IP time-to-live (TTL) field
- In general, when a router receives an IP packet, it decrements the TTL field on the packet
  - A failsafe mechanism to ensure packets don't keep taking up network resources for too long
- If a router receives a packet with TTL=0, it sends an ICMP time exceeded message (type=11, code=0) to the source endpoint

Traceroute

- Traceroute sends multiple packets to a destination endpoint
- But it progressively increases the TTL on those packets: 1, 2, ...
- Every time a time exceeded message is received, record the router's IP address
- Process repeated until the destination endpoint is reached
- If the packet reaches the destination endpoint (i.e.: TTL is high enough), then the endpoint sends a port unreachable message (type=3, code=3)

Traceroute



A small demo…

Summary of ICMP

- A protocol for network diagnostics and troubleshooting
- Two useful tools: ping and traceroute
- Ping: test connectivity to a machine totally outside your control
  - Use ICMP echo request and reply
- Traceroute: determine router-level path to a remote endpoint
  - A smart use of the TTL field in the IP header

Network Address Translation (NAT)

Background: The Internet's growing pains

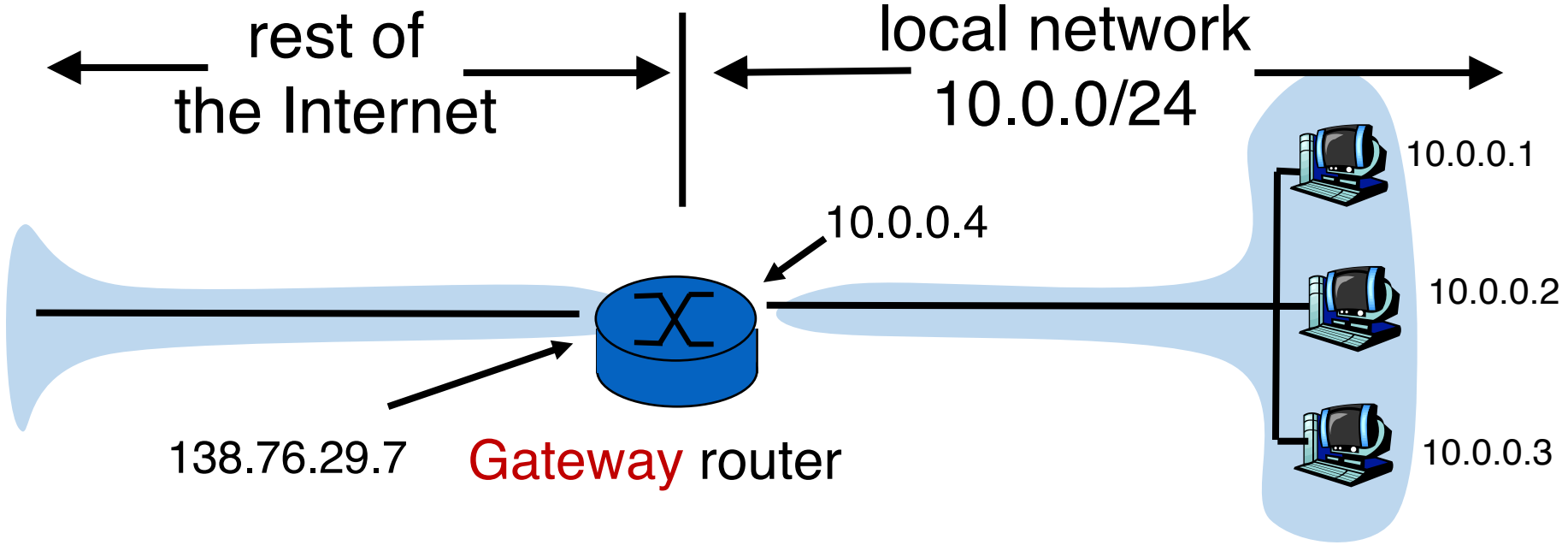
- Networks had incompatible addressing
  - IPv4 versus other network-layer protocols (X.25)
  - Routable address ranges different across networks
- Entire networks were changing their Internet Service Providers
  - ISPs don't want to route directly to internal endpoints
- IPv4 address exhaustion
  - Insufficient large IP blocks even for large networks
  - Rutgers (AS46) has > 130,000 publicly routable IP addresses
  - IIT Madras (a well-known public university in India, AS141340) has 512

(Source: ipinfo.io)

Network Address Translation

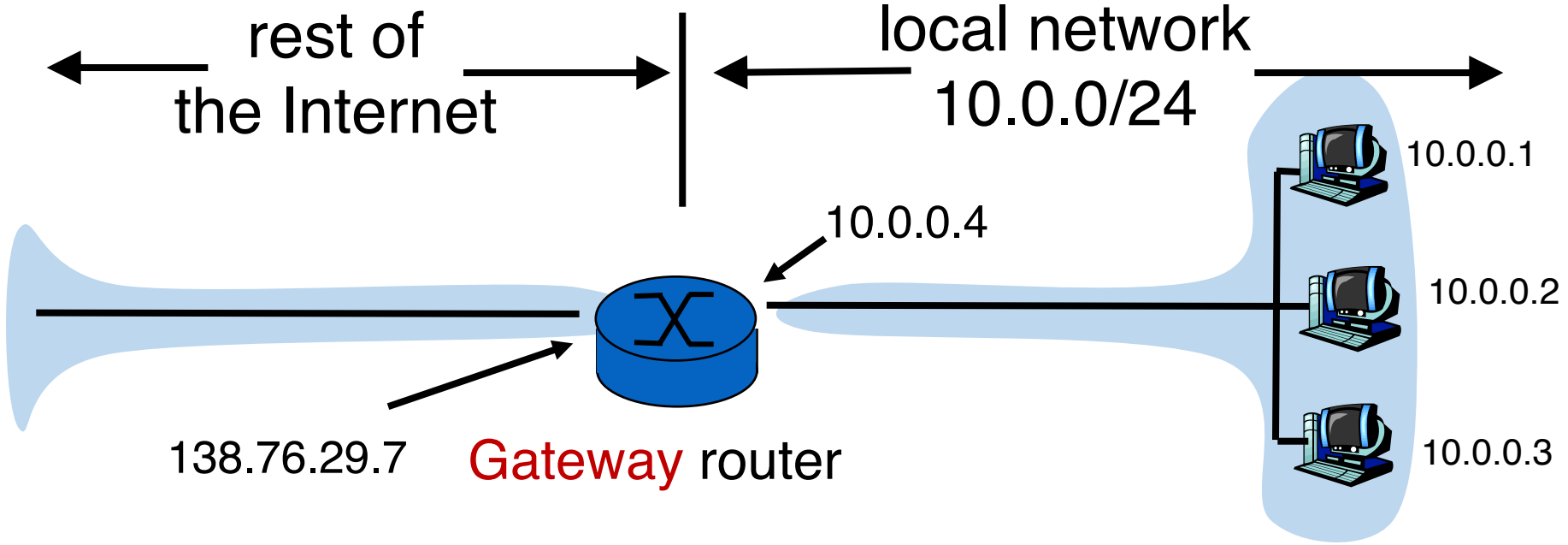
- When a router modifies fields in an IP packet to:
- Enable communication across networks with different (network-layer) addressing formats and address ranges
- Allow a network to change its connectivity to the Internet en masse by modifying the source IP to a (publicly-visible) gateway IP address
- Masquerade as an entire network of endpoints using (say) one publicly visible IP address
  - Effect: use fewer IP addresses for more endpoints!
- We'll see a standard design: "Network address and port translation" (NAPT, RFC 2663)

Typical NAT setup (NAPT)



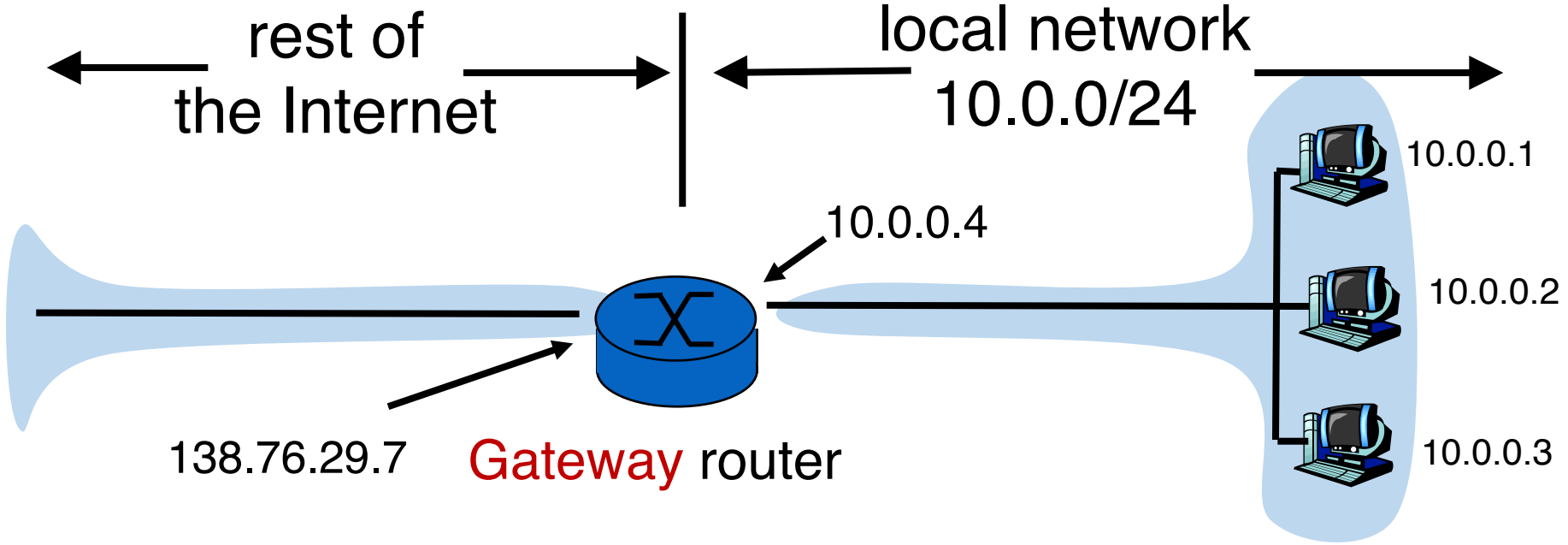
- The gateway's IP, 138.76.29.7 is publicly visible
- The local endpoint IP addresses in 10.0.0/24 are private
- All datagrams leaving local network have the same source IP as the gateway

Typical NAT setup (NAPT)



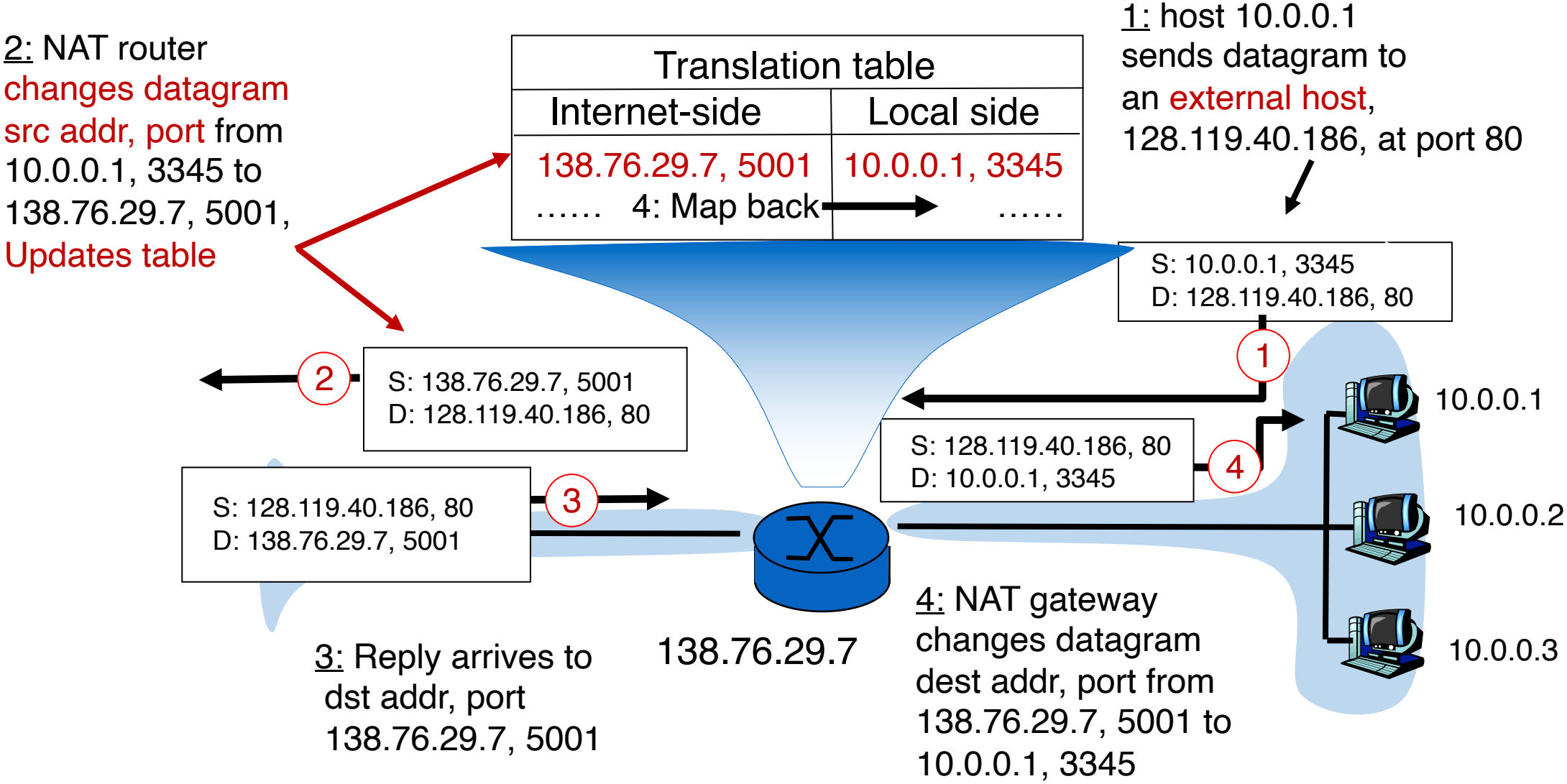
That is, for the rest of the Internet, the gateway masquerades as a single endpoint representing (hiding) all the private endpoints. The entire network just needs one (or a few) public IP addresses.

Typical NAT setup (NAPT)



The NAT gateway router accomplishes this by using a different transport port for each distinct (transport-level) conversation between the local network and the Internet.

Typical NAT setup (NAPT)



Features of IP-masquerading NAT

- Use one or a few public IPs: You don’t need a lot of addresses from your ISP
- Change addresses of devices inside the local network freely, without notifying the rest of the Internet
- Change the public IP address freely independent of network-local endpoints
- Devices inside the local network are not publicly visible, routable, or accessible
- Most IP masquerading NATs block incoming connections originating from the Internet
  - Only way to communicate is if the internal host initiates the conversation

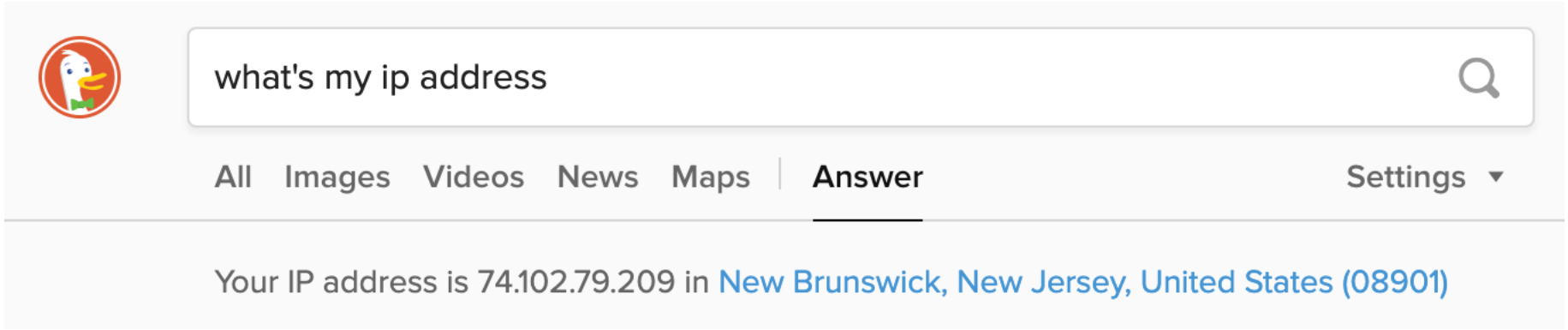
If you're home, you're likely behind NAT

- Most access routers (e.g., your home WiFi router) implement network address translation

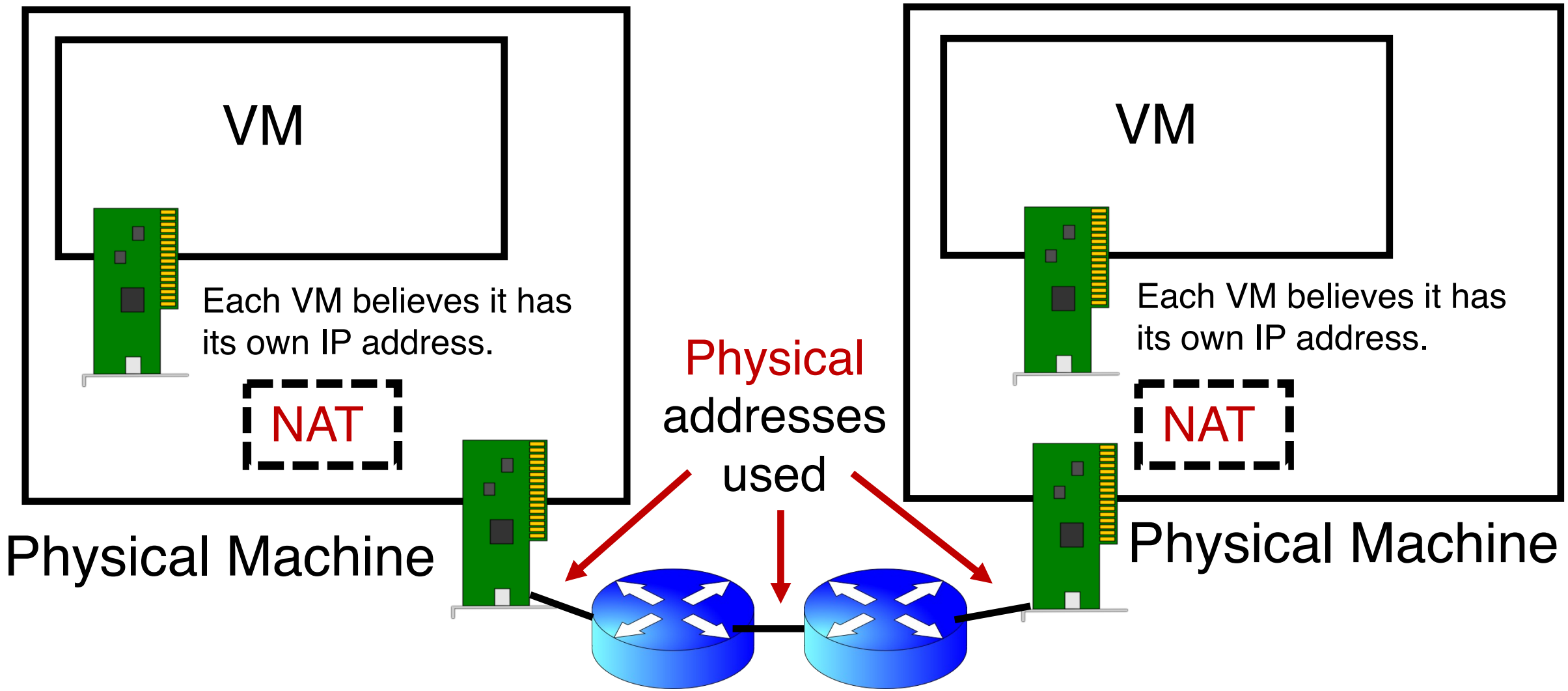
- You can check this by comparing your local address (visible from `ifconfig`) and your externally-visible IP address (e.g., type "what's my IP address?" on your browser search bar)

If you're home, you're likely behind NAT

```
[flow:352-S20]$  ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether f0:18:98:1c:fc:36
        inet6 fe80::1036:7dea:82ee:e868%en0 prefixlen 64 secured scopeid 0xa
        inet 192.168.1.151 netmask 0xffffff00 broadcast 192.168.1.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
[flow:352-S20]$
```



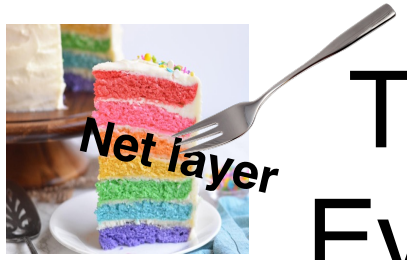
On public cloud, you're behind NAT



Limitations of IP-masquerading NATs

- Connection limit due to 16-bit port-number field
  - ~64K total simultaneous connections with a single public IP address
- NAT can be controversial
  - "Routers should only manipulate headers up to the network layer, not modify headers at the transport layer!"
- Application developers must take NAT into account
  - e.g., peer-to-peer applications like Skype
- Internet "purists": instead, solve address shortage with IPv6
  - 32-bit IP addresses are just not enough
  - Esp. with more devices (your watch, your fridge, …) coming online

Routing Protocols



The network layer is all about reachability. Every protocol below solves a sub-problem.

How does an endpoint get an address?

DHCP

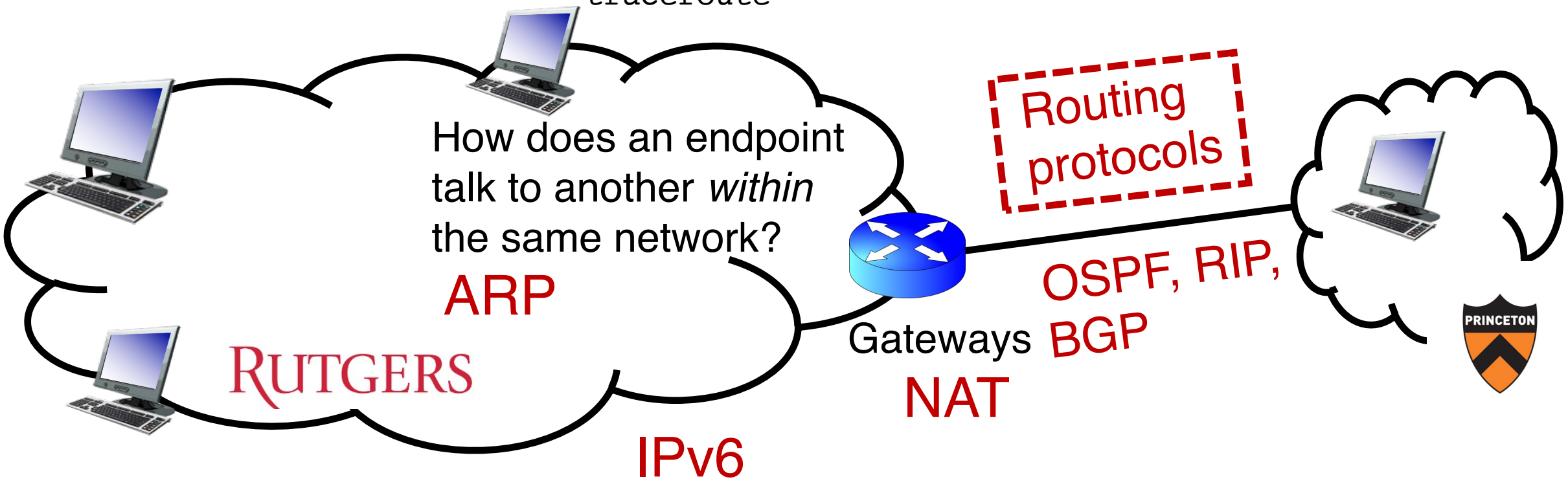
Debugging

ICMP

`ping`

`traceroute`

How does an endpoint talk to another *outside* its network?



Review: Key network-layer functions

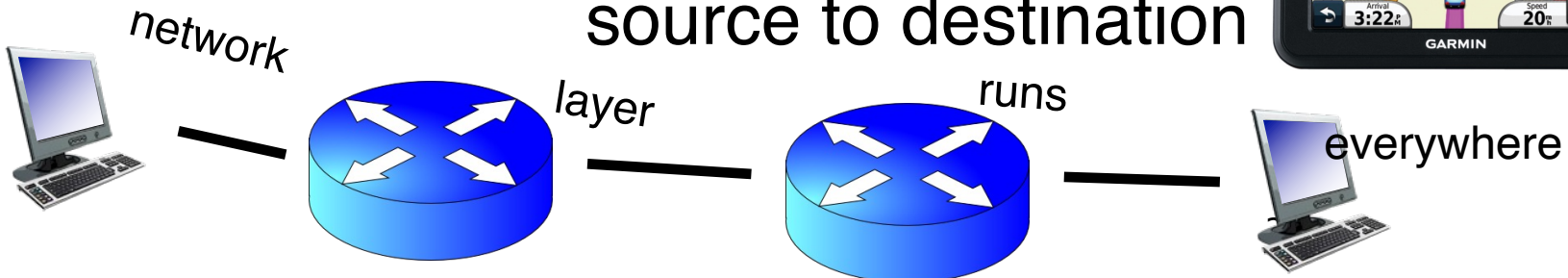
- Forwarding (data plane): move packets from router's input to appropriate router output

- Routing (control plane): determine route taken by packets from source to destination

Analogy: taking a road trip

- Forwarding: process of getting through single interchange

- Routing: process of planning trip from source to destination



Routing is a fundamental problem in networking.

How would one design a “Google Maps” to navigate the Internet?

Goals of Routing Protocols #1

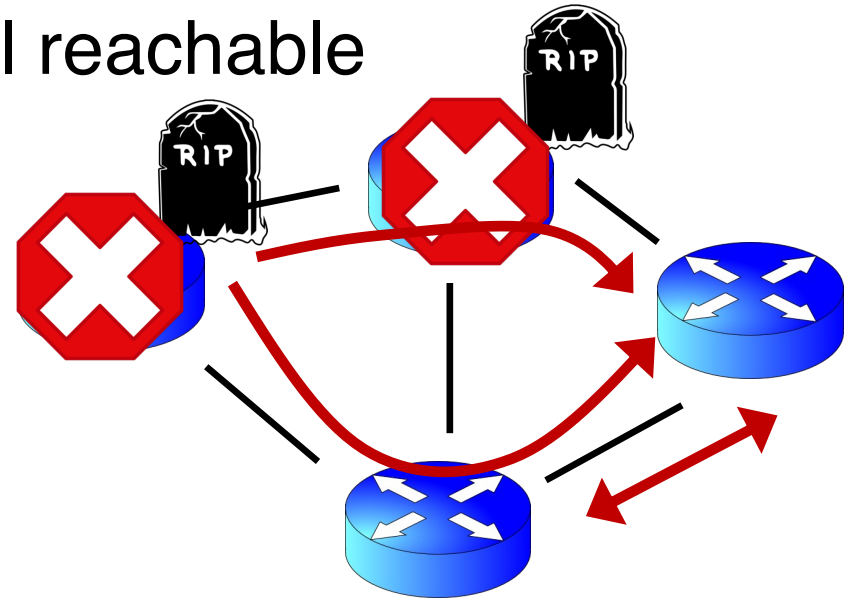
- Determine good paths from source to destination

- "Good" = least cost
  - Least propagation delay
  - Least cost per unit bandwidth (e.g., $ per Gbit/s)
  - Least congested (workload-driven)

- "Path" = a sequence of router ports (links)

Goals of Routing Protocols #2

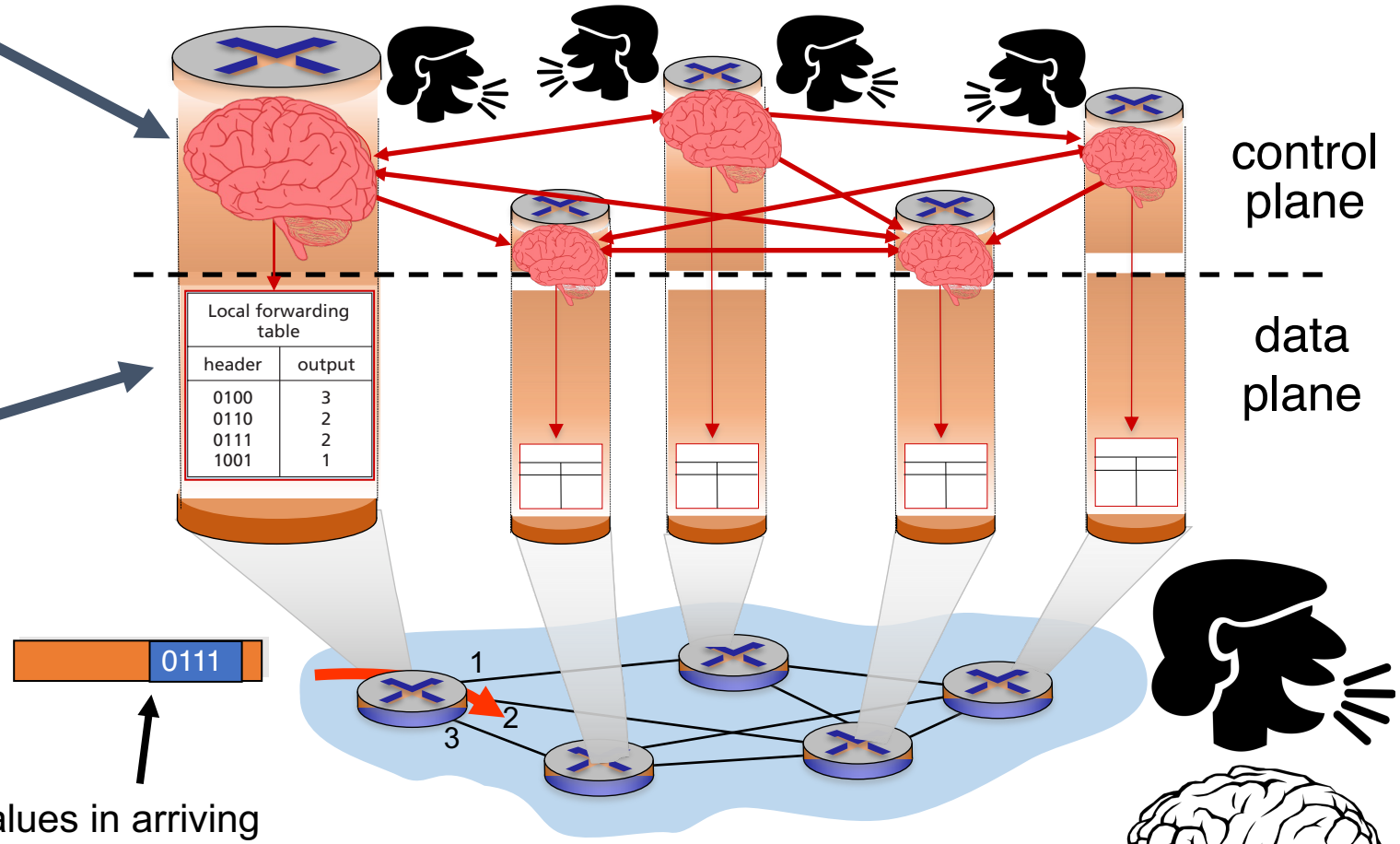
- Make networks resilient to failures
- Routers & links can fail without taking down the entire network
- Entire subsets can be unreachable; rest still reachable
- Hence, the protocol must be distributed



Per-router control plane

Distributed control plane: Components in every router interact with other components to produce a routing outcome.

Data plane per-packet processing, moving packet from input port to output port



Routing protocol

Q1. What info exchanged?

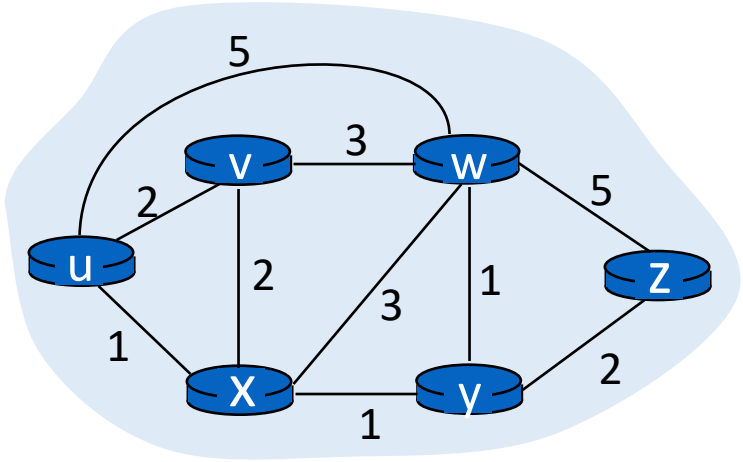
Q2. What computation?

The graph abstraction

- Routing algorithms work over an abstract representation of a network: the graph abstraction

Ex: Rutgers campus

u: Computer Science
v: School of Engineering
…



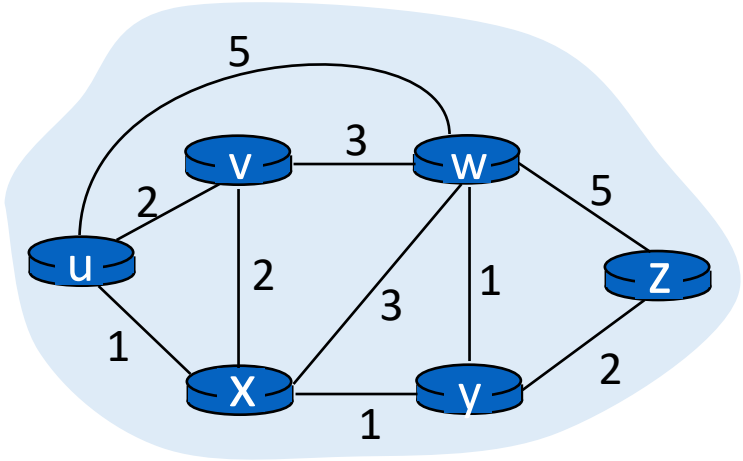
- Each router is a node in a graph
- Each link is an edge in the graph
- Edges have weights (also called link metrics). Set by netadmin

The graph abstraction

- Routing algorithms work over an abstract representation of a network: the graph abstraction

Ex: Rutgers campus

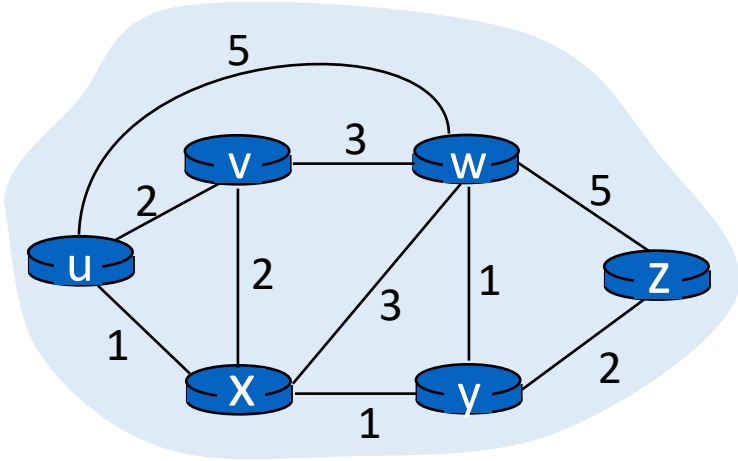
u: Computer Science
v: School of Engineering
…



- G = (N, E)
- N = {u, v, w, x, y, z}
- E = { (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

The graph abstraction

- Cost of an edge: c(x, y)
  - Examples: c(u, v) = 2, c(u, w) = 5
- Cost of a path = sum of edge costs
  - c(path x → w → y → z) = 3 + 1 + 2 = 6



- Outcome of routing: each node should determine the least cost path to every other node
- Q1: What information should nodes exchange with each other to enable this computation?
- Q2: What algorithm should each node run to compute the least cost path to every node?

Coming up next

Link state protocols

Each router has complete information of the graph

Messages exchanged by flooding all over the network

Communication expensive, but complete

Distance vector protocols

Each router only maintains distances & next hop to others

Messages are exchanged over each link and stay within the link

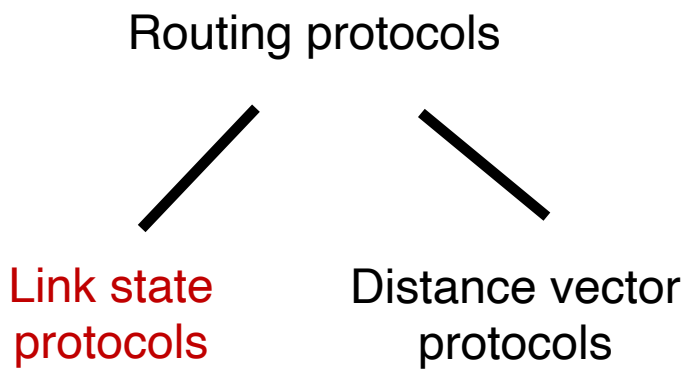
Communication cheap, but incomplete

Link State Protocols

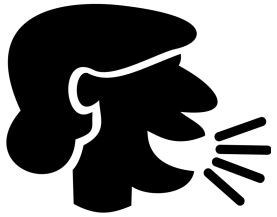
Link state protocol

- Each router knows the state of all the links and routers in the network

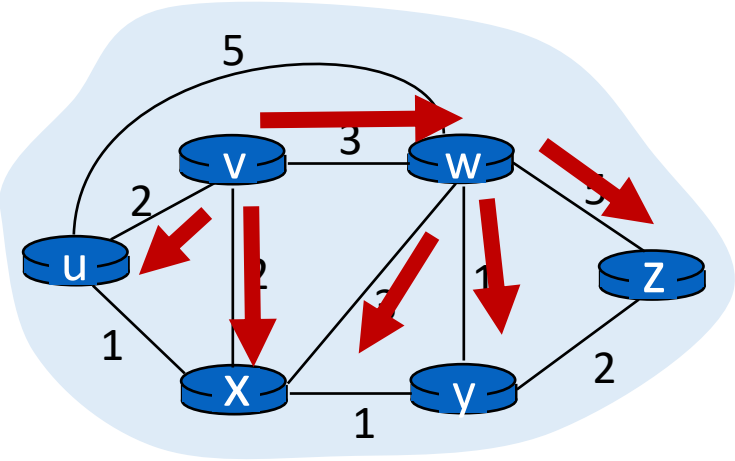
- Every router performs an independent computation on globally shared knowledge of network's complete graph representation



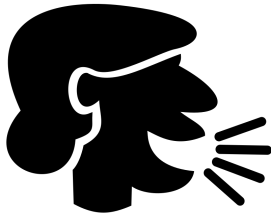
Q1: Information exchange



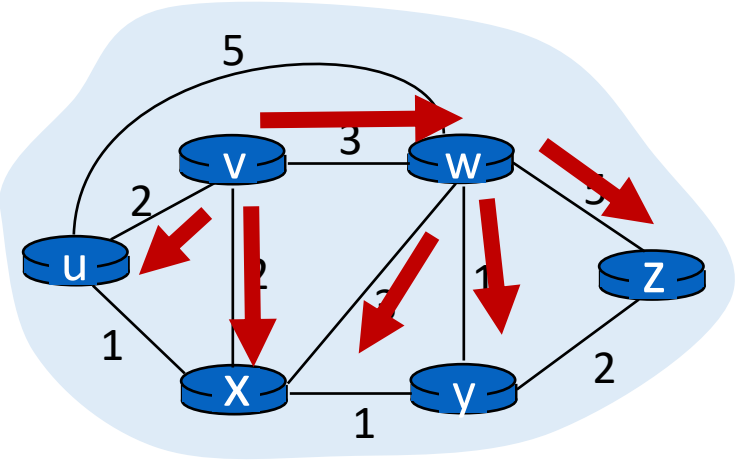
- Link state flooding: the process by which neighborhood information of each network router is transmitted to all other routers
- Each router sends a link state advertisement (LSA) to each of its neighbors
- LSA contains the router ID, the IP prefix owned by the router, the router's neighbors, and link cost to those neighbors
- Upon receiving an LSA, a router forwards it to each of its neighbors: flooding



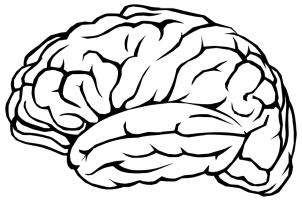
Q1: Information exchange



- Eventually, the entire network receives LSAs originated by each router
- LSAs put into a link state database
- LSAs occur periodically and whenever the graph changes
  - Example: if a link fails
  - Example: if a new link or router is added
- The routing algorithm running at each router can use the entire network's graph to compute least cost paths



Q2: The algorithm



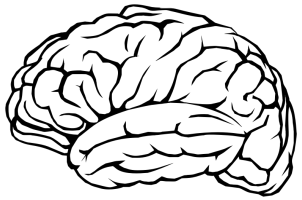
Dijkstra's algorithm

- Given a network graph, the algorithm computes the least cost paths from one node (source) to all other nodes
- This can then be used to compute the forwarding table at that node
- Iterative algorithm: maintain estimates of least costs to reach every other node. After k iterations, each node definitively knows the least cost path to k destinations

Notation:

- c(x,y): link cost from node x to y; = ∞ if not direct neighbors
- D(v): current estimate of cost of path from source to destination v
- p(v): (predecessor node) the last node before v on the path from source to v
- N': set of nodes whose least cost path is definitively known

Dijsktra's Algorithm



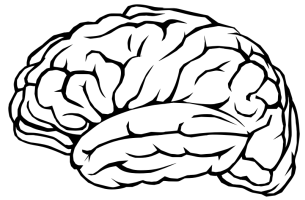
```
1  Initialization:
2    N' = {u}
3    for all nodes v
4      if v adjacent to u
5          then D(v) = c(u,v)
6      else D(v) = ∞
7
8  Loop
9     find w not in N' such that D(w) is a minimum
10    add w to N'
11    update D(v) for all v adjacent to w and not in N' :
12       D(v) = min( D(v), D(w) + c(w,v) )
13    /* new cost to v is either old cost to v or known
14     shortest path cost to w plus cost from w to v */
15 until all nodes in N'
```

Initial estimates of distances are just the link costs of neighbors.

Least cost node among all estimates. This cost cannot decrease further.

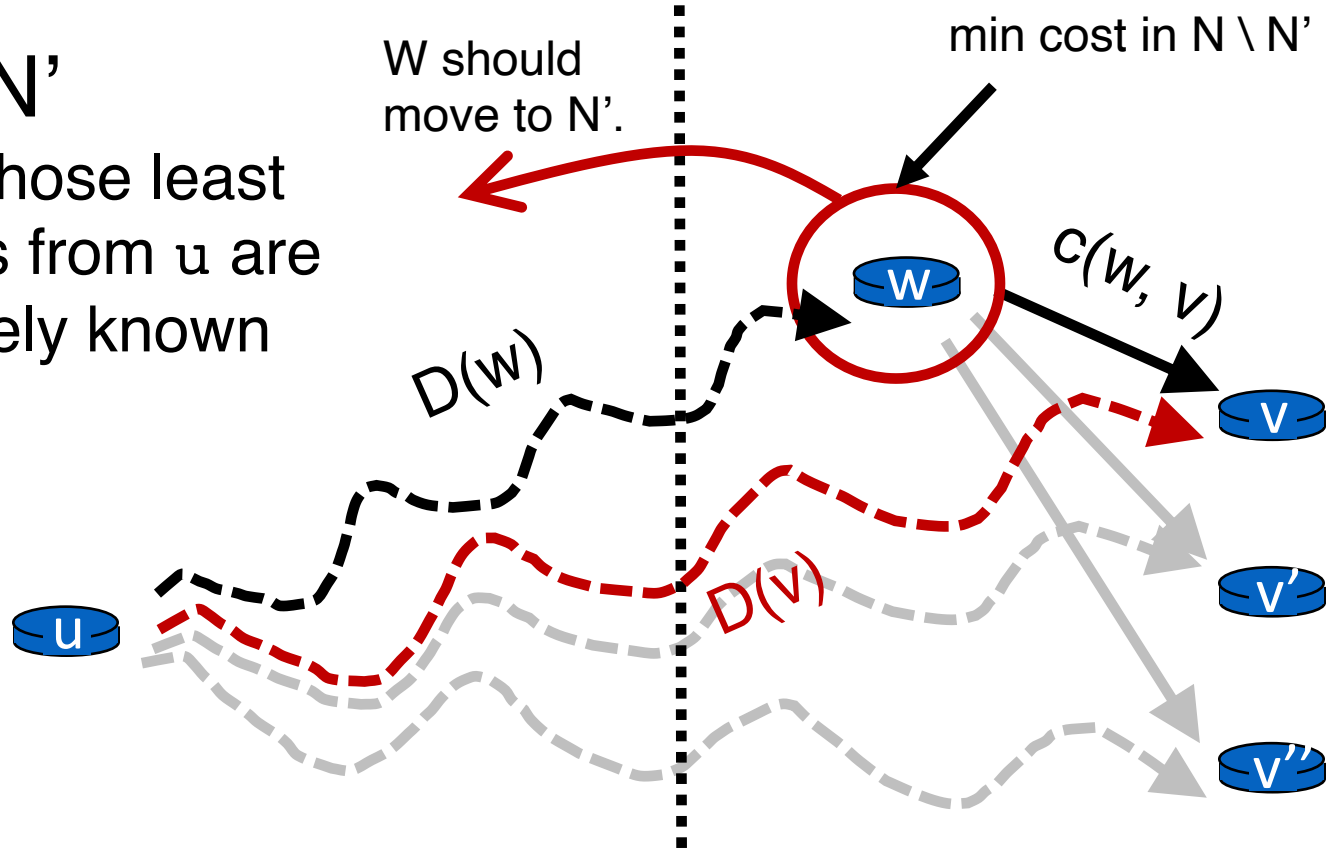
Relaxation

Visualization



N’

nodes whose least cost paths from u are definitively known



N \ N’

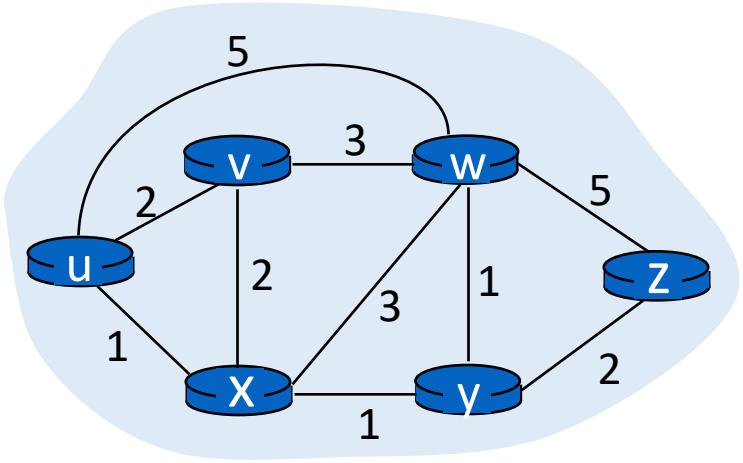
Nodes with estimated least path costs, not definitively known to be smallest possible

Relaxation: for each v in N \ N’, is the cost of the path via w smaller than known least cost path to v?
If so, update D(v)
Predecessor of v is w.

Cost of path via w: D(w) + c(w,v)
Cost of known best path: D(v)

Dijkstra's algorithm: example

Step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



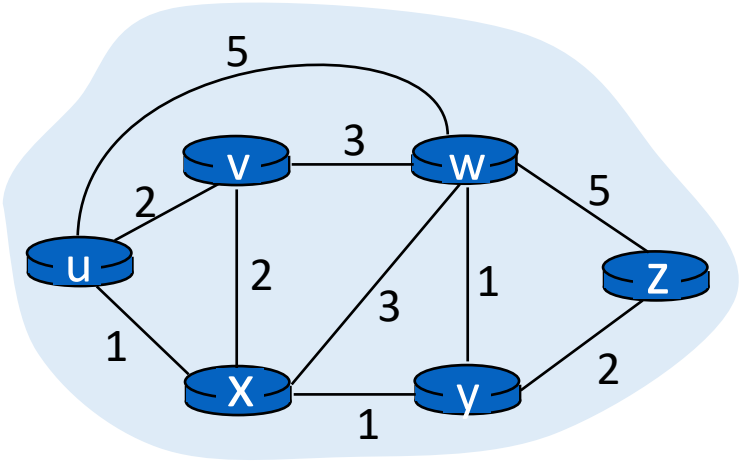
Constructing the forwarding table

- To find the router port to use for a given destination (router), find the predecessor of the node iteratively until reaching an immediate neighbor of the source `u`

- The port connecting `u` to this neighbor is the output port for this destination

Constructing the forwarding table

- Suppose we want forwarding entry for z.



D(v),p(v) D(w),p(w) D(x),p(x) D(y),p(y) D(z),p(z)

2,u 3,y 1,u 2,x 4,y

z: p(z) = y

y: p(y) = x

x: p(x) = u

x is an immediate neighbor of u

Forwarding table at u:

destination	link
z	(u,x)

Summary of link state protocols

- Each router announces link state to the entire network using flooding
- Each node independently computes least cost paths to every other node using the full network graph
- Dijkstra's algorithm can efficiently compute these best paths
  - Easy to populate the forwarding table from predecessor information computed during the algorithm