# Routing (part 4)

Lecture 26

http://www.cs.rutgers.edu/~sn624/352-F24

Srinivas Narayana

RUTGERS
UNIVERSITY | NEW BRUNSWICK

# The network layer enables reachability.
# We’ll see protocols that solve subproblems.

How does an endpoint get an address?

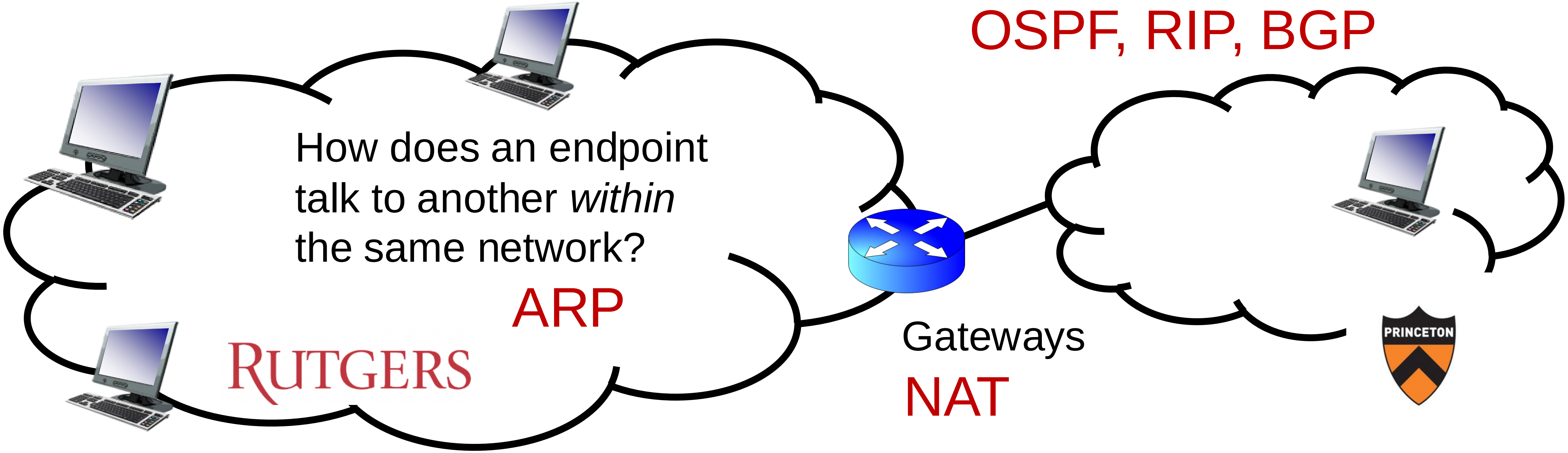
DHCP

Debugging?

ICMP

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

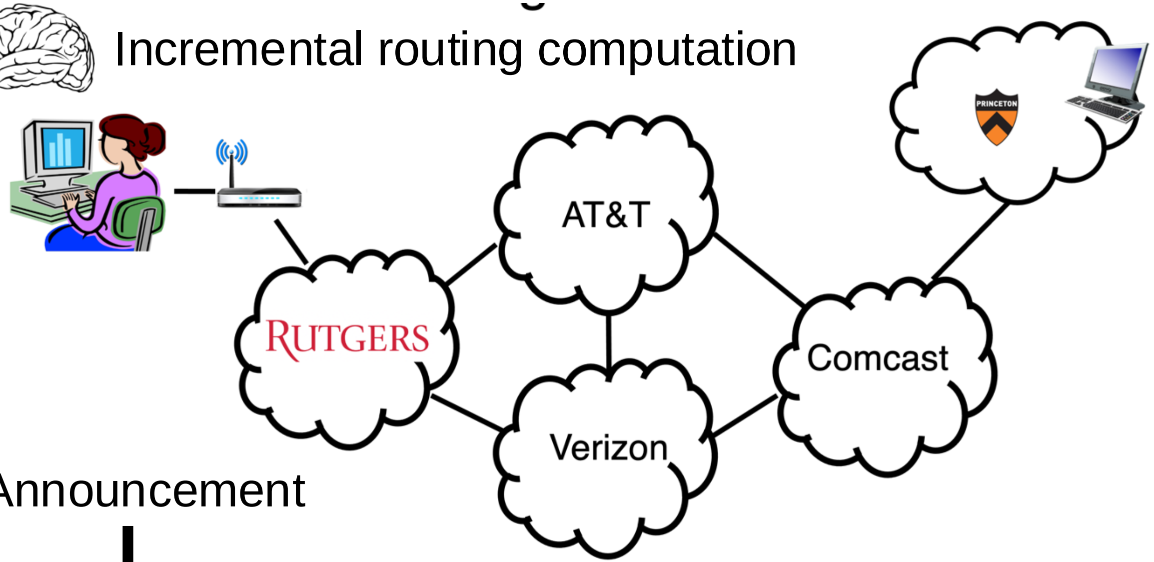
Routing protocols
OSPF, RIP, BGP



# Internet is large and federated

Abstractions of topology for protocol messages

Incremental routing computation




Routing protocols

- Link state protocols
- Distance vector protocols
- Path vector protocols

Routing announcement = destination prefix + attributes

AS-level path

Next hop

…

Announcement

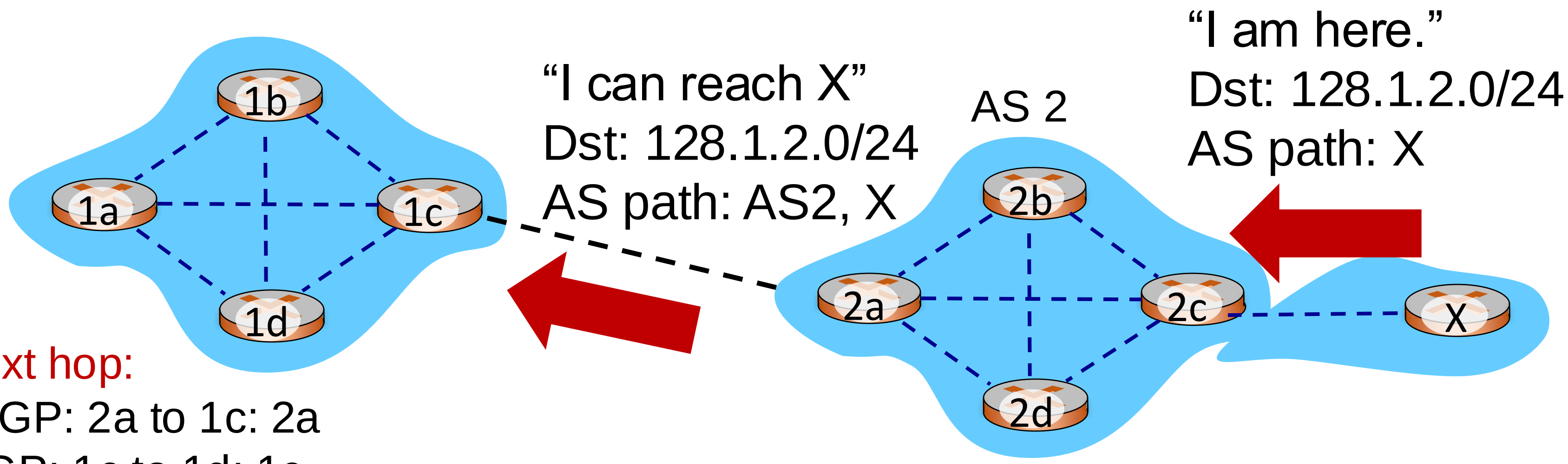
Import policy

Route selection

Export policy

Control plane

Data plane

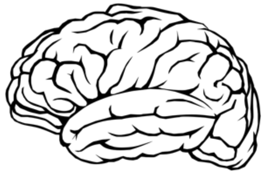


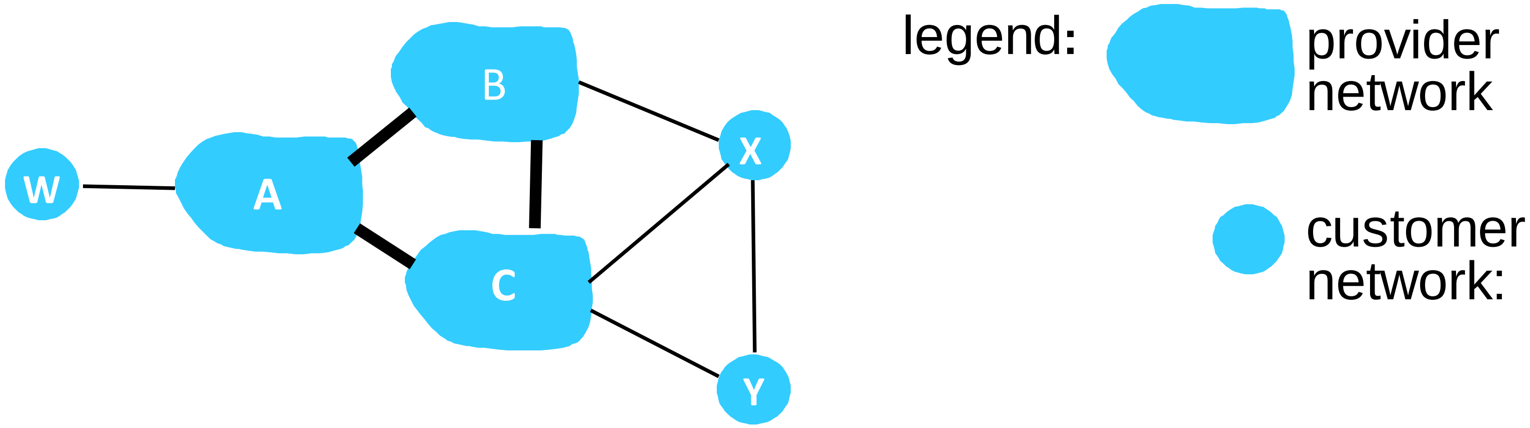

Next hop:

eBGP: 2a to 1c: 2a

iBGP: 1c to 1d: 1c

# BGP Import Policy

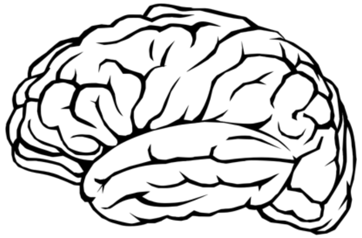





Suppose an ISP wants to minimize costs by avoiding routing through its providers when possible.

- Suppose C announces path Cy to x
- Further, y announces a direct path (“y”) to x
- Then x may choose not to import the path Cy to y since it has a peer path (“y”) towards y

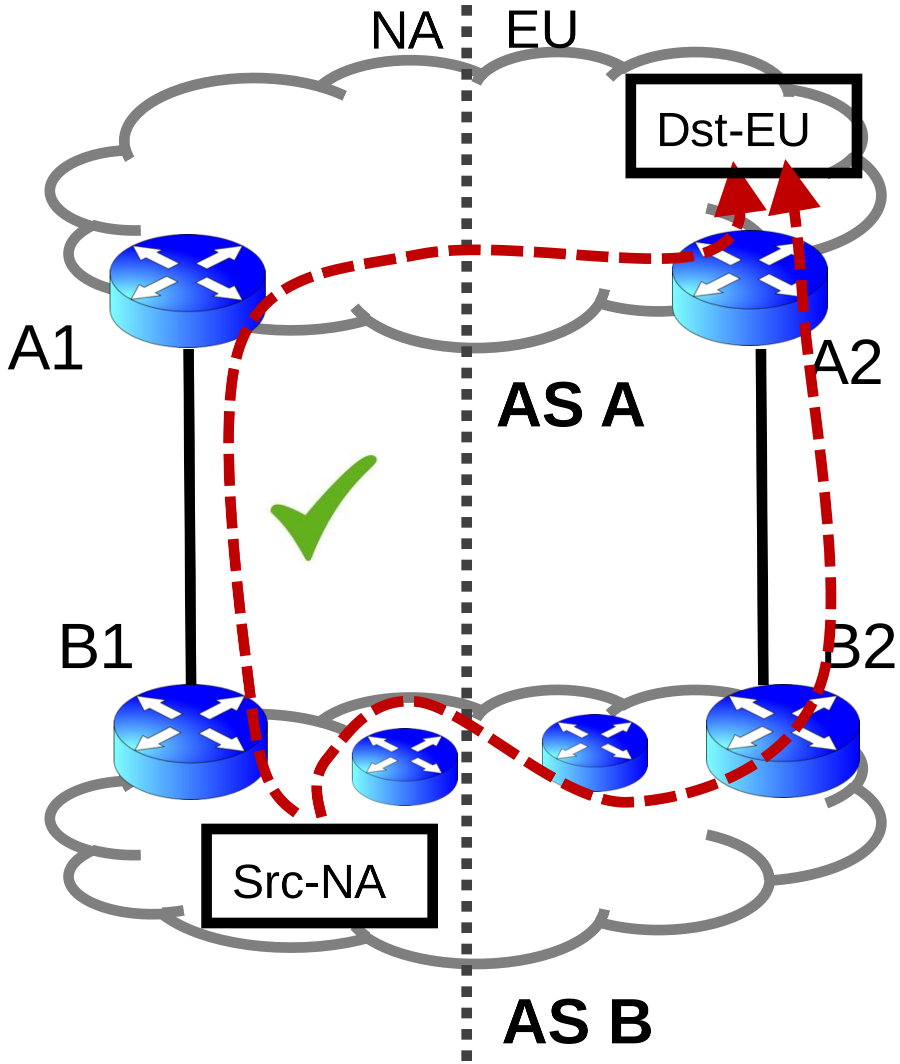
# Q2. BGP Route Selection



- When a router imports more than one route to a destination IP prefix, it selects route based on:
  1. local preference value attribute (import policy decision -- set by network admin)
  2. shortest AS-PATH
  3. closest NEXT-HOP router
  4. Several additional criteria: You can read up on the full, complex, list of criteria, e.g., at https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/13753-25.html

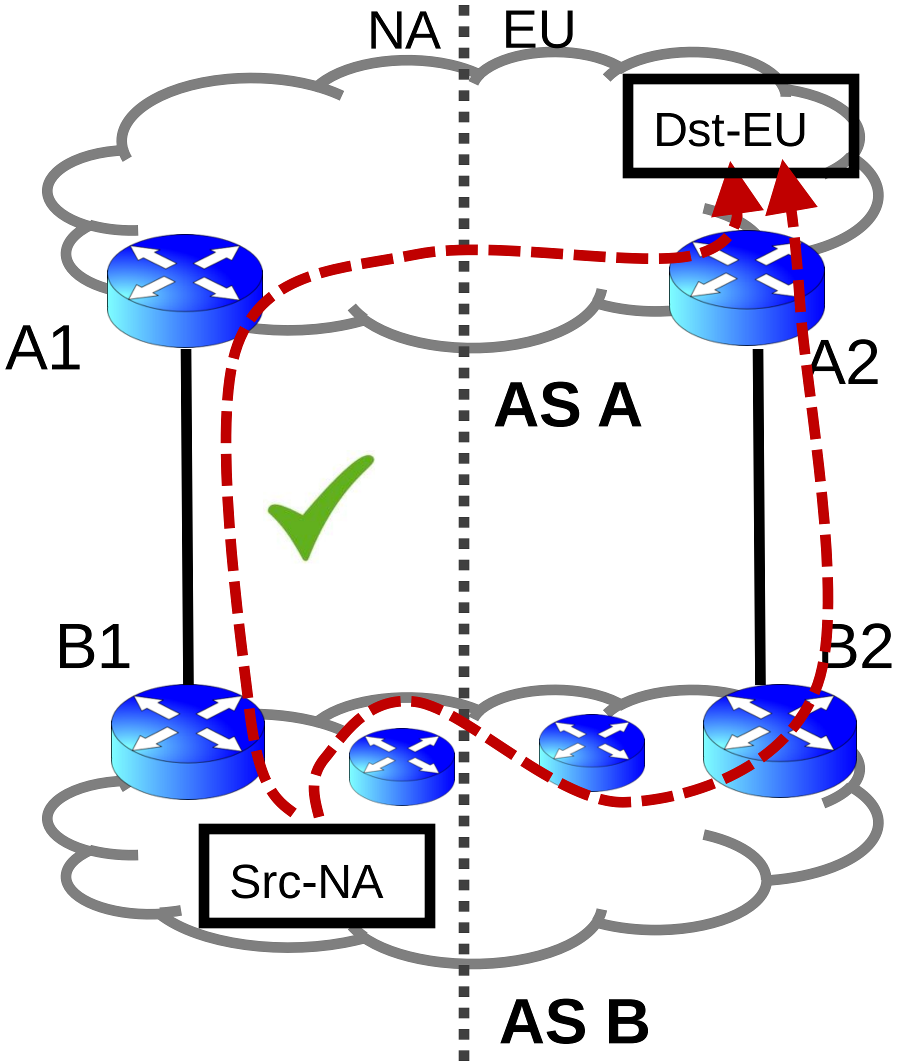
# Example of route selection

- Suppose AS A and B are connected to each other both in North America (NA) and in Europe (EU)
- A source in NA wants to reach a destination in EU
- There are two paths available
  - *Assume* same local preference
  - Same AS path length
- Closest next hop-router: choose path via B1 rather than B2

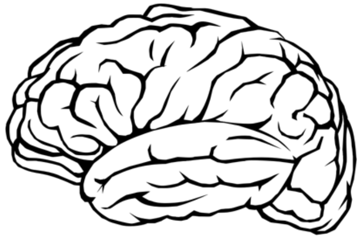


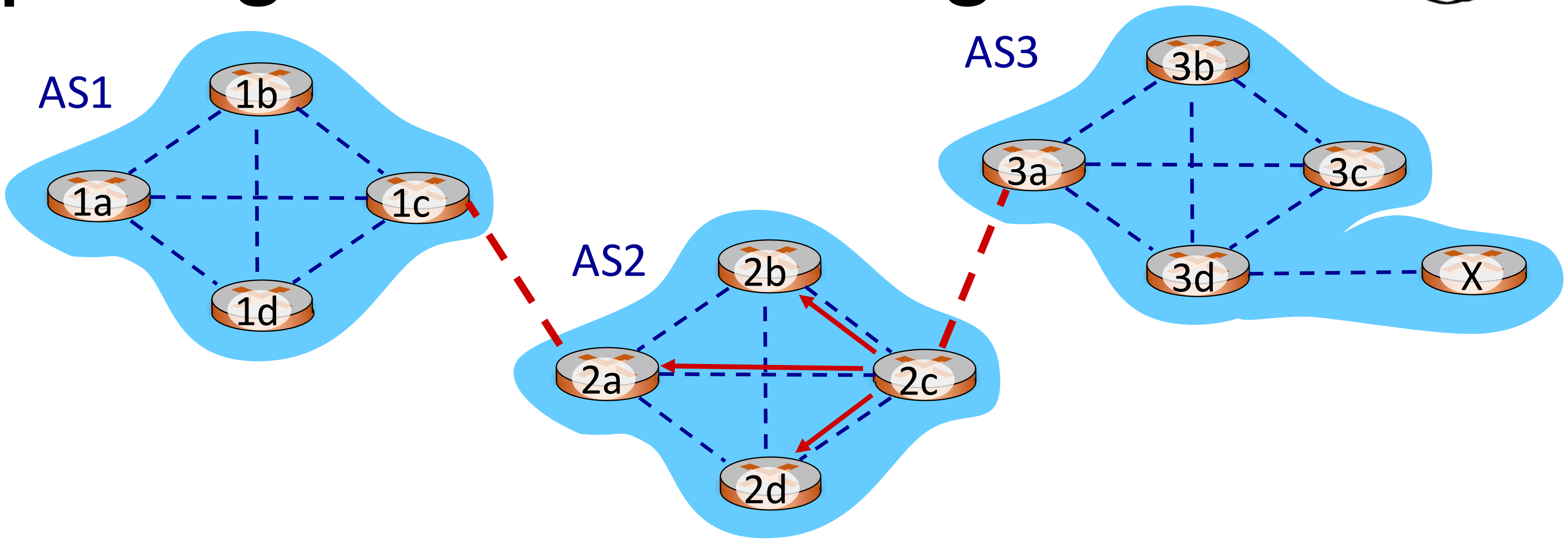
# Example of route selection

- Choosing closest next-hop results in early exit routing
  - Try to exit the local AS as early as possible
  - Also called hot potato routing
- Reduce resource use within local AS
  - potentially at the expense of another AS



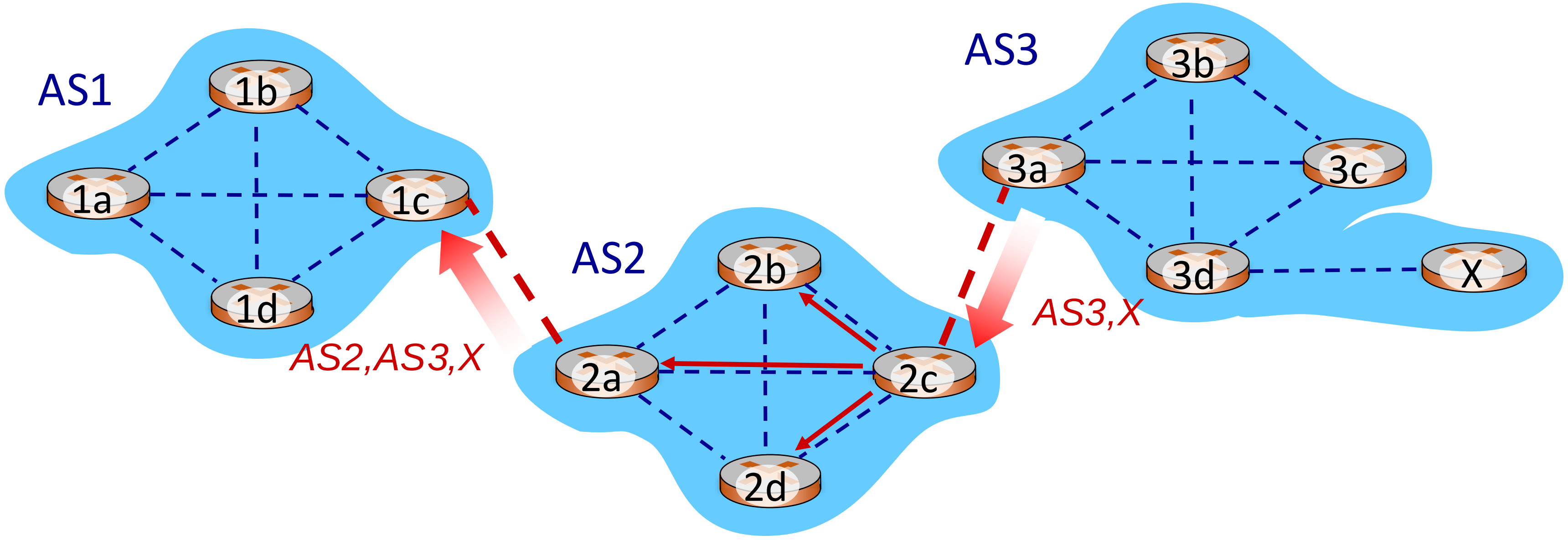
# Computing the forwarding table





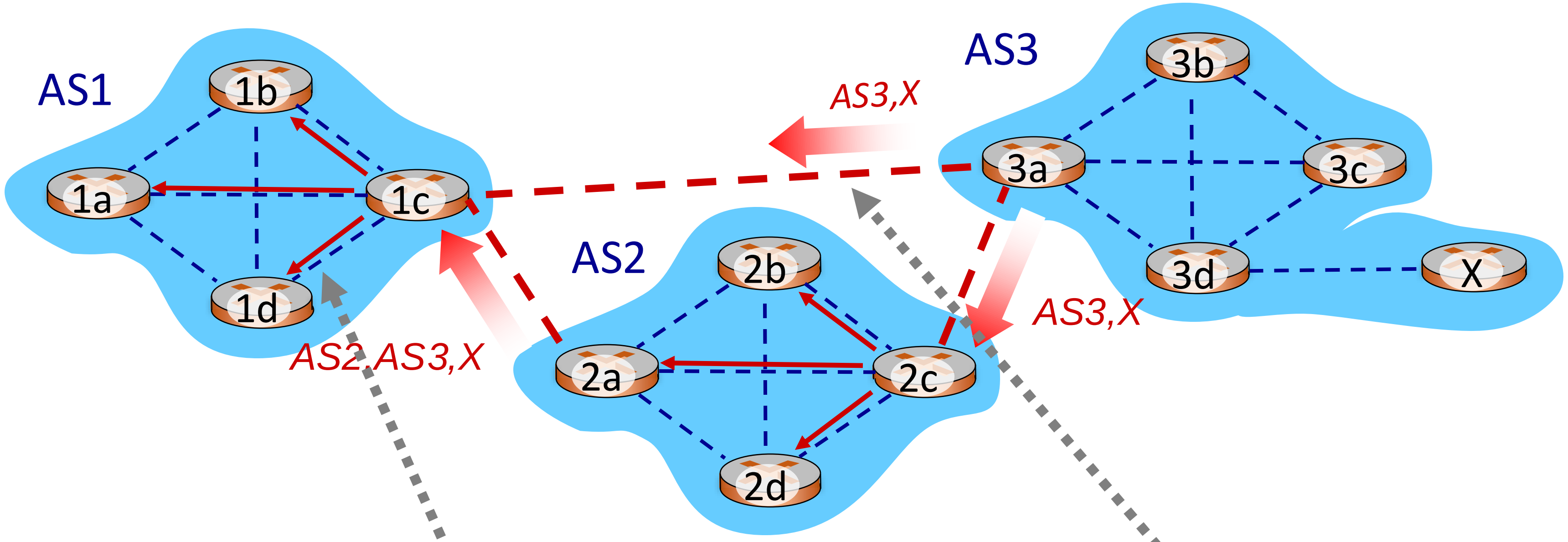

- Suppose a router in AS1 wants to forward a packet destined to external prefix X.
- How is the forwarding table entry for X at 1d computed?
- How is the forwarding table entry for X at 1c computed?

# eBGP and iBGP announcements




- AS2 router 2c receives path announcement AS3,X (via eBGP) from AS3 router 3a
- Based on AS2 import policy, AS2 router 2c imports and selects path AS3,X, propagates (via iBGP) to all AS2 routers
- Based on AS2 export policy, AS2 router 2a announces (via eBGP) path AS2, AS3, X to AS1 router 1c

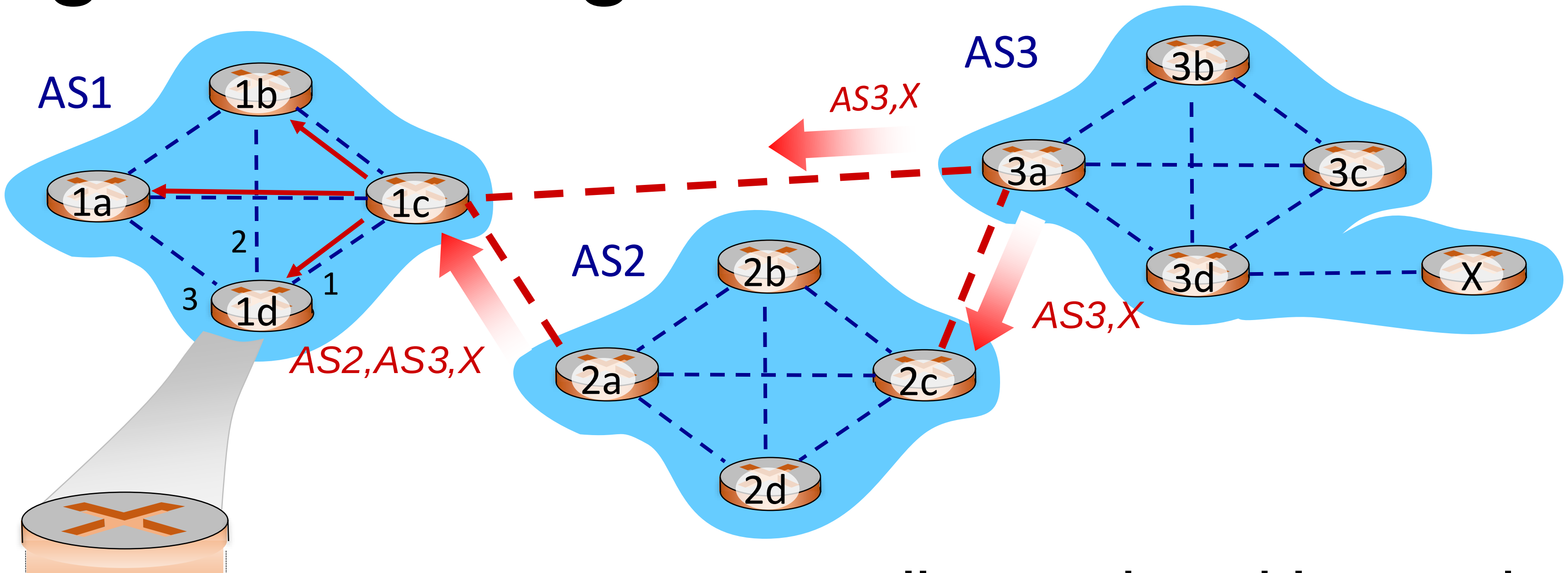
# eBGP and iBGP announcements




A given router may learn about multiple paths to destination:

- AS1 gateway router 1c learns path AS2,AS3,X from 2a (next hop 2a)
- AS1 gateway router 1c learns path AS3,X from 3a (next hop 3a)
- Through BGP route selection process, AS1 gateway router 1c chooses path AS3,X, and announces path within AS1 via iBGP (next hop 1c)

# Setting forwarding table entries




| dest | Next-Hop |
|---|---|
| … | … |
| X | 1c |
| … | … |

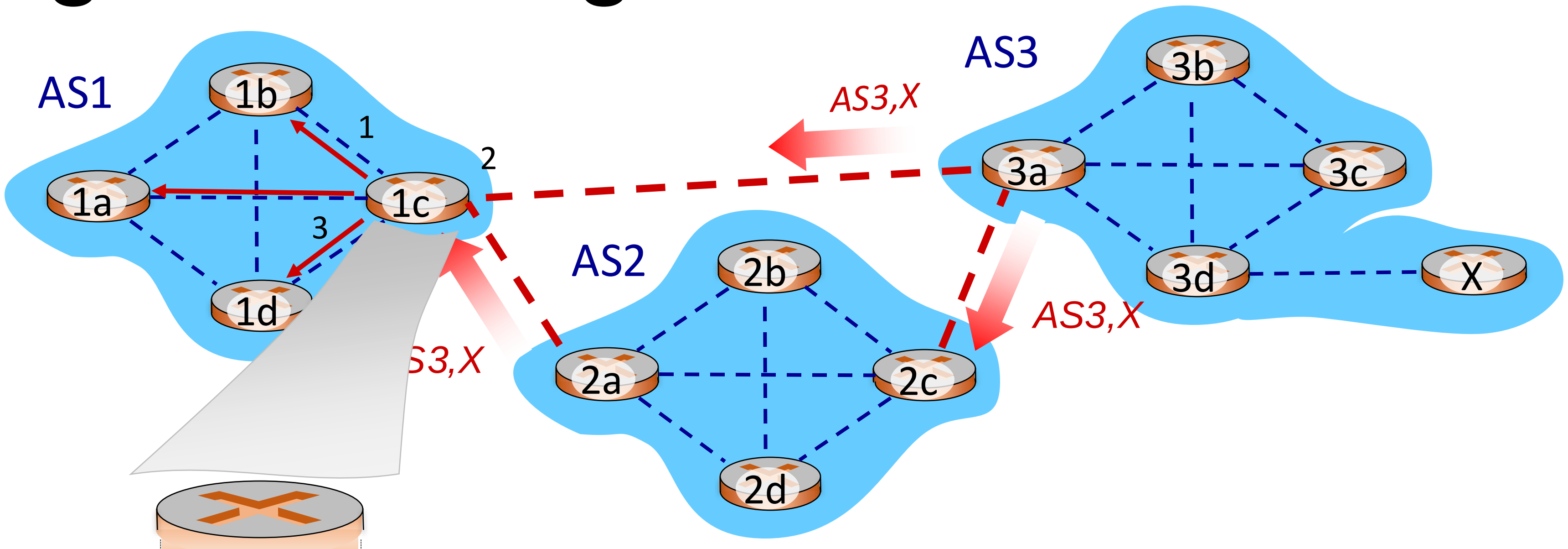
| NH | interface |
|---|---|
| … | … |
| 1c | 1 |
| … | … |

| IP Dst | Out port |
|---|---|
| X | 1 |

Forwarding table

- recall: 1a, 1b, 1d learn about dest X via iBGP from next-hop 1c: “path to X goes through 1c”
- 1d: intra-domain routing: to get to 1c, forward over outgoing local interface 1

# Setting forwarding table entries




| dest | Next-Hop |
|---|---|
| … | … |
| X | 3a |
| … | … |

| NH | interface |
|---|---|
| … | … |
| 3a | 2 |
| … | … |

| IP Dst | Out port |
|---|---|
| X | 2 |

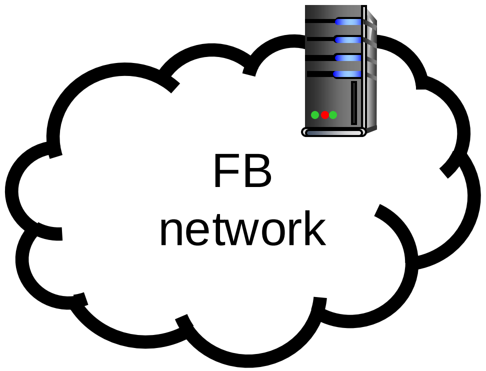
Forwarding table

- recall: 1c learns about dest X via eBGP from next-hop 3a: “path to X goes through 3a”
- 1c: to get to link-local neighbor 3a, forward out interface 2

# Summary: Inter-domain routing

- Federation and scale introduce new requirements for routing on the Internet
- BGP is *the* protocol that handles Internet routing
- Path vector: exchange paths to a destination with attributes
- Policy-based import of routes, route selection, and export

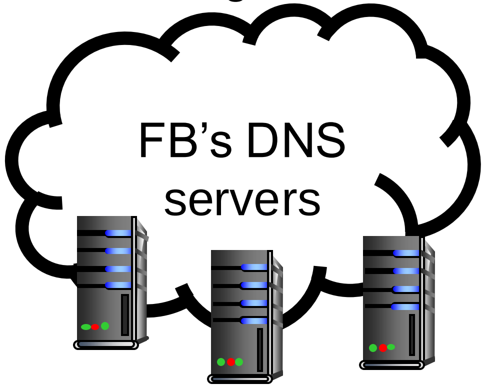
# BGP's impact: October '21 FB++ outage

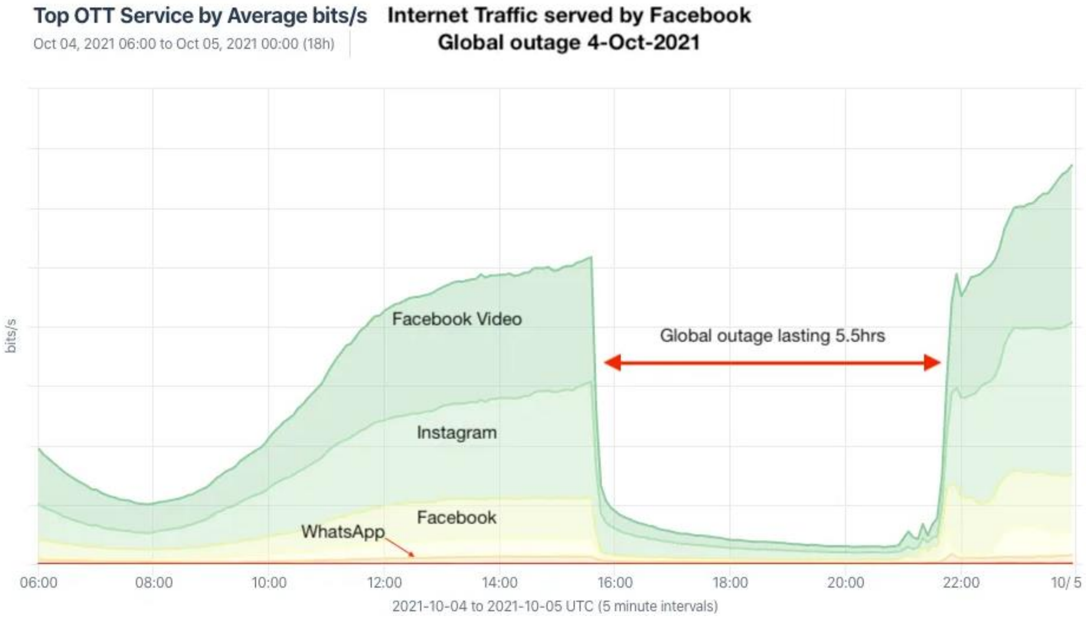



BGP route
withdrawal:
"I can't reach FB
anymore"

Rest of the
Internet

BGP route
withdrawal: don't
use me to get to FB




No remote access (no more reachability due to BGP withdrawal of DC and DNS servers)

Restricted physical access (prox can't verify, can't access prox server)

https://engineering.fb.com/2021/10/05/networking-traffic/outage-details/

By Doug Madory - https://www.kentik.com/blog/facebooks-historic-outage-explained/, CC BY 4.0, https://commons.wikimedia.org/w/index.php?curid=110816752

# Network Address Translation (NAT)

# Background: The Internet's growing pains

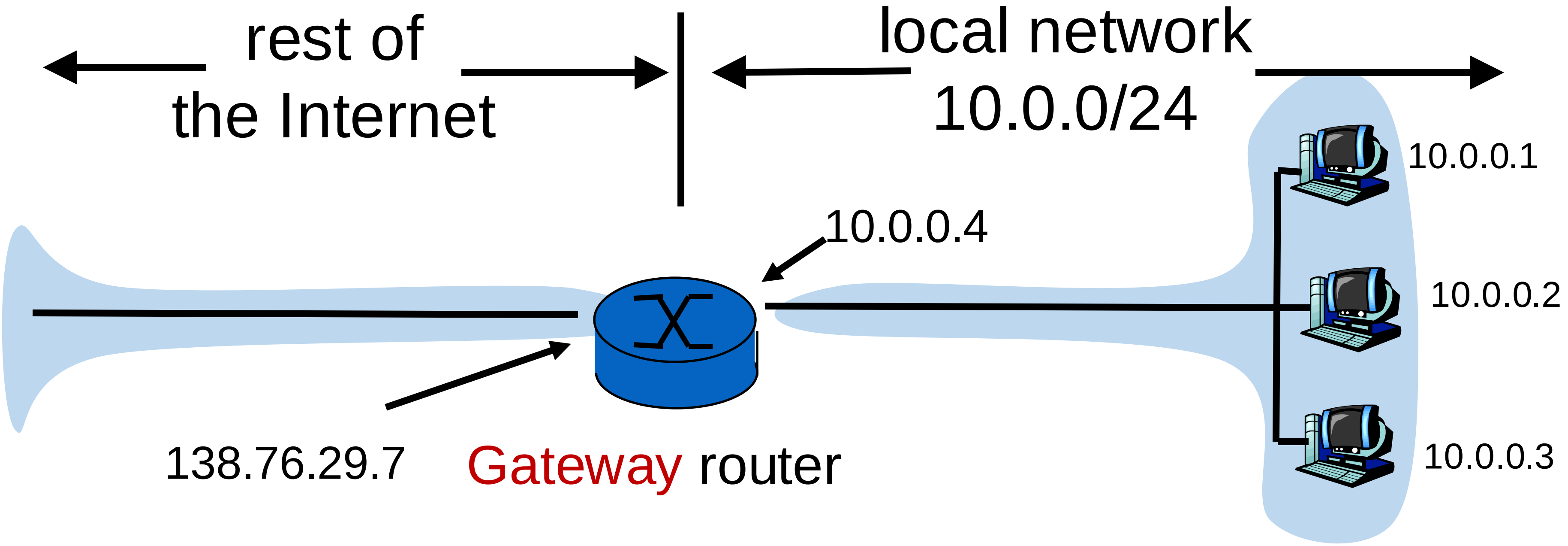
- Networks had incompatible addressing
  - IPv4 versus other network-layer protocols (X.25)
- Entire networks were changing their Internet Service Providers
  - ISPs don't want to route directly to internal endpoints
- IPv4 address exhaustion
  - Insufficient large IP blocks even for large networks
  - 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

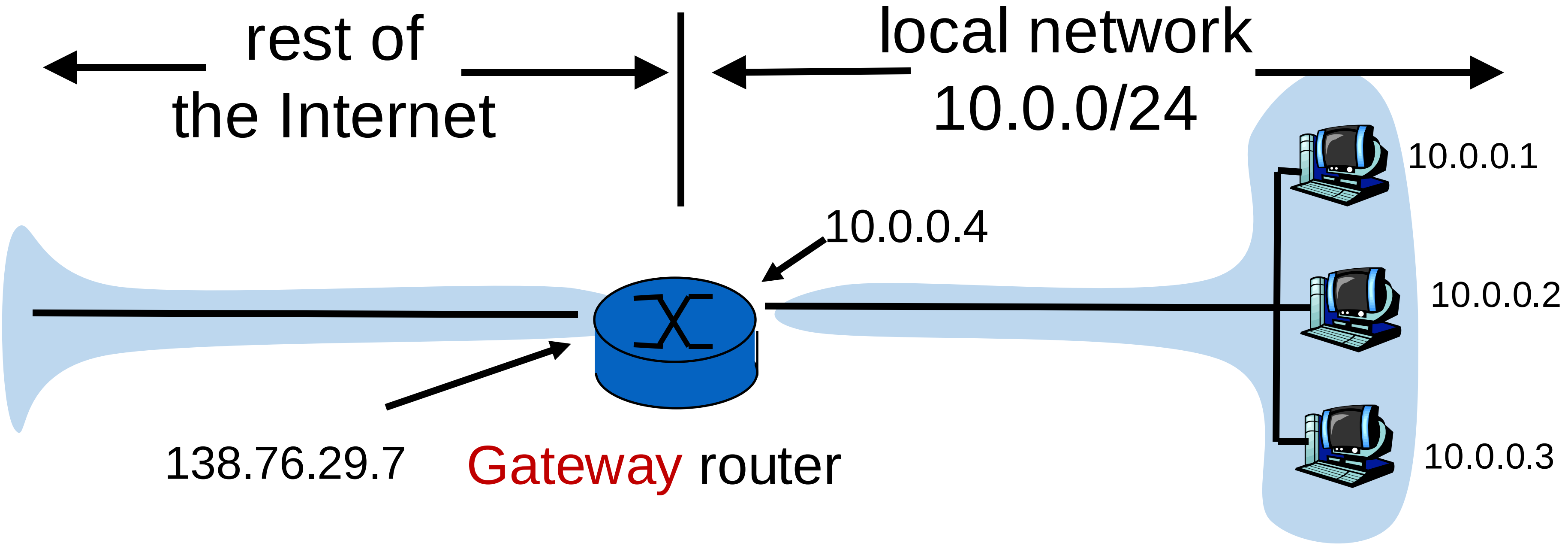
- When a router modifies fields in an IP packet to:
- Enable communication across networks with different (network-layer) addressing formats and address ranges
- Allow a network to change its connectivity to the Internet en masse by modifying the source IP to a (publicly-visible) gateway IP address
- 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)



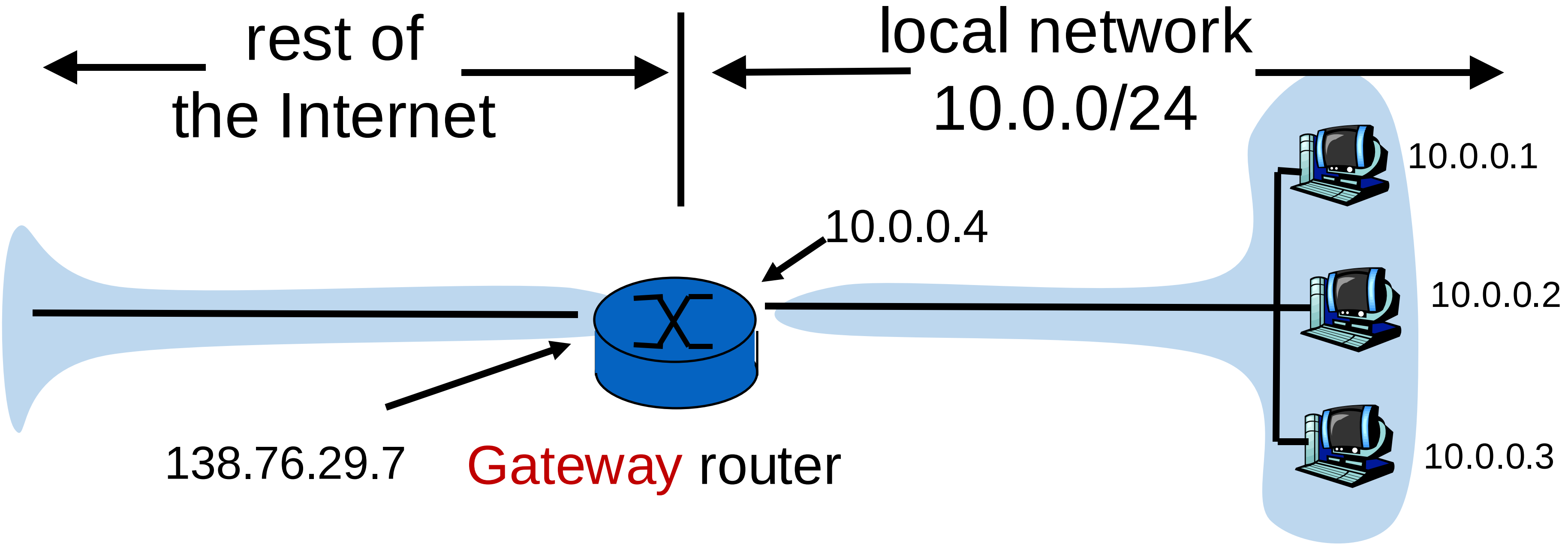

- The gateway's IP, 138.76.29.7 is publicly visible
- The local endpoint IP addresses in 10.0.0/24 are private
- All datagrams leaving local network have the same source IP as the gateway

# Typical NAT setup (NAPT)



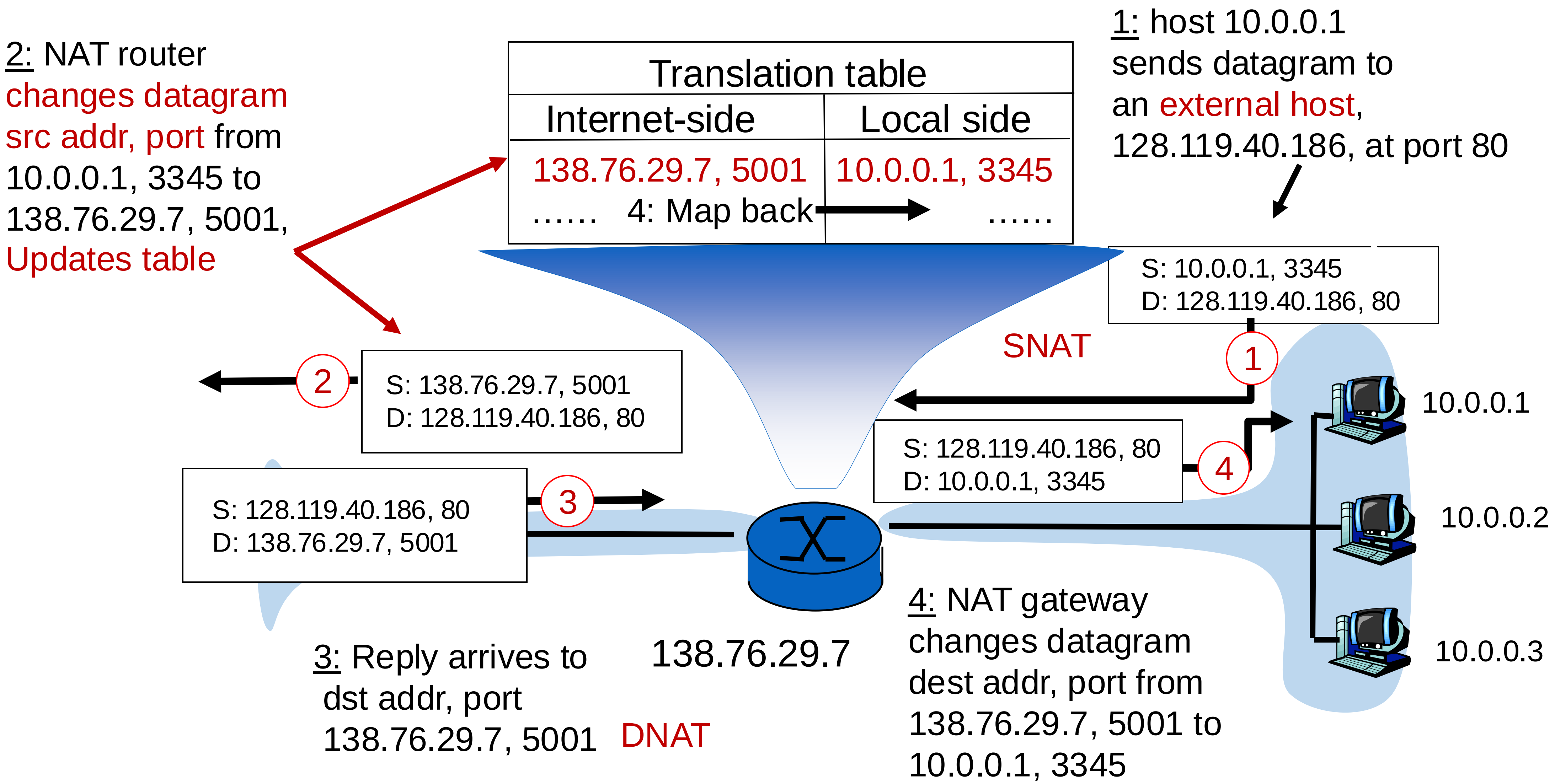

That is, for the rest of the Internet, the gateway masquerades as a single endpoint representing (hiding) all the private endpoints. The entire network just needs one (or a few) public IP addresses.

# 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

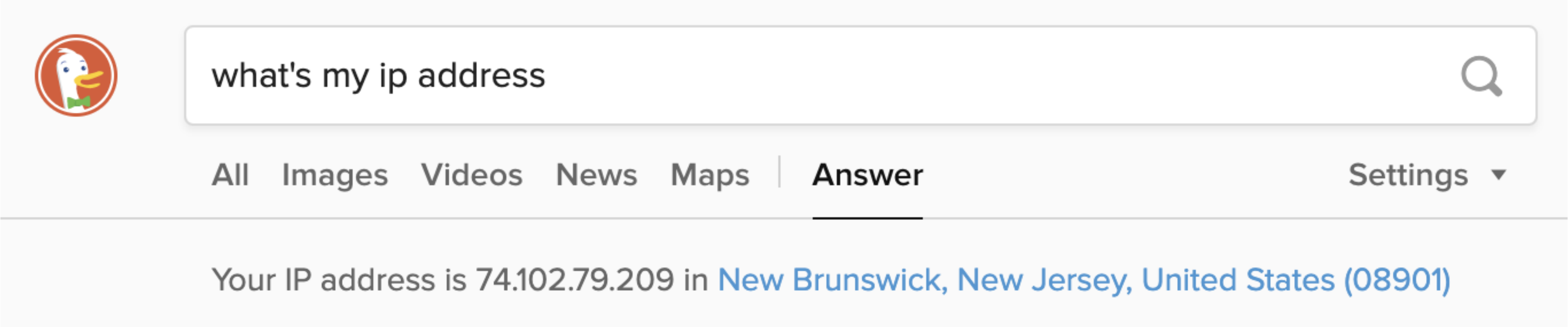
# If you’re home, you’re likely behind NAT

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

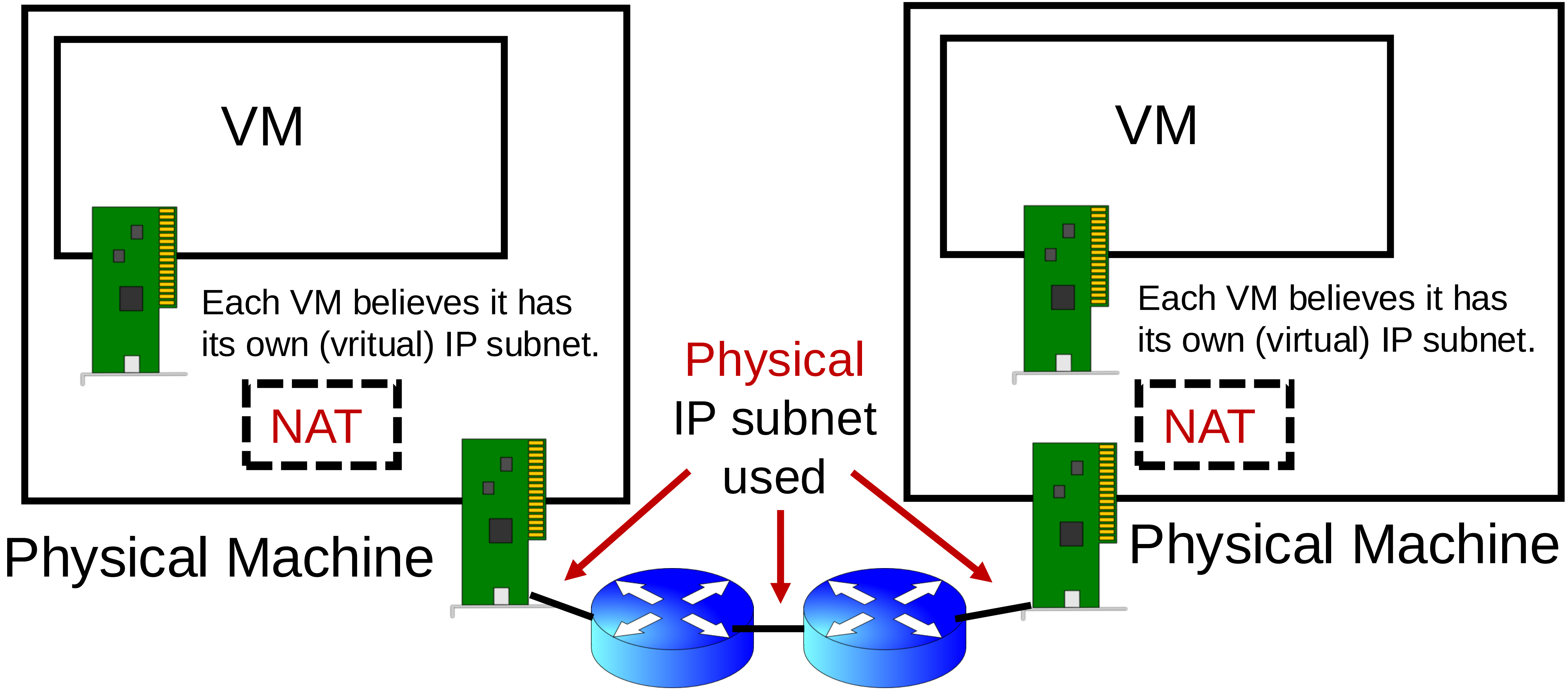
- You can check this by comparing your local address (visible from `ifconfig`) and your externally-visible IP address (e.g., type “what’s my IP address?” on your browser search bar)

# 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
- 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

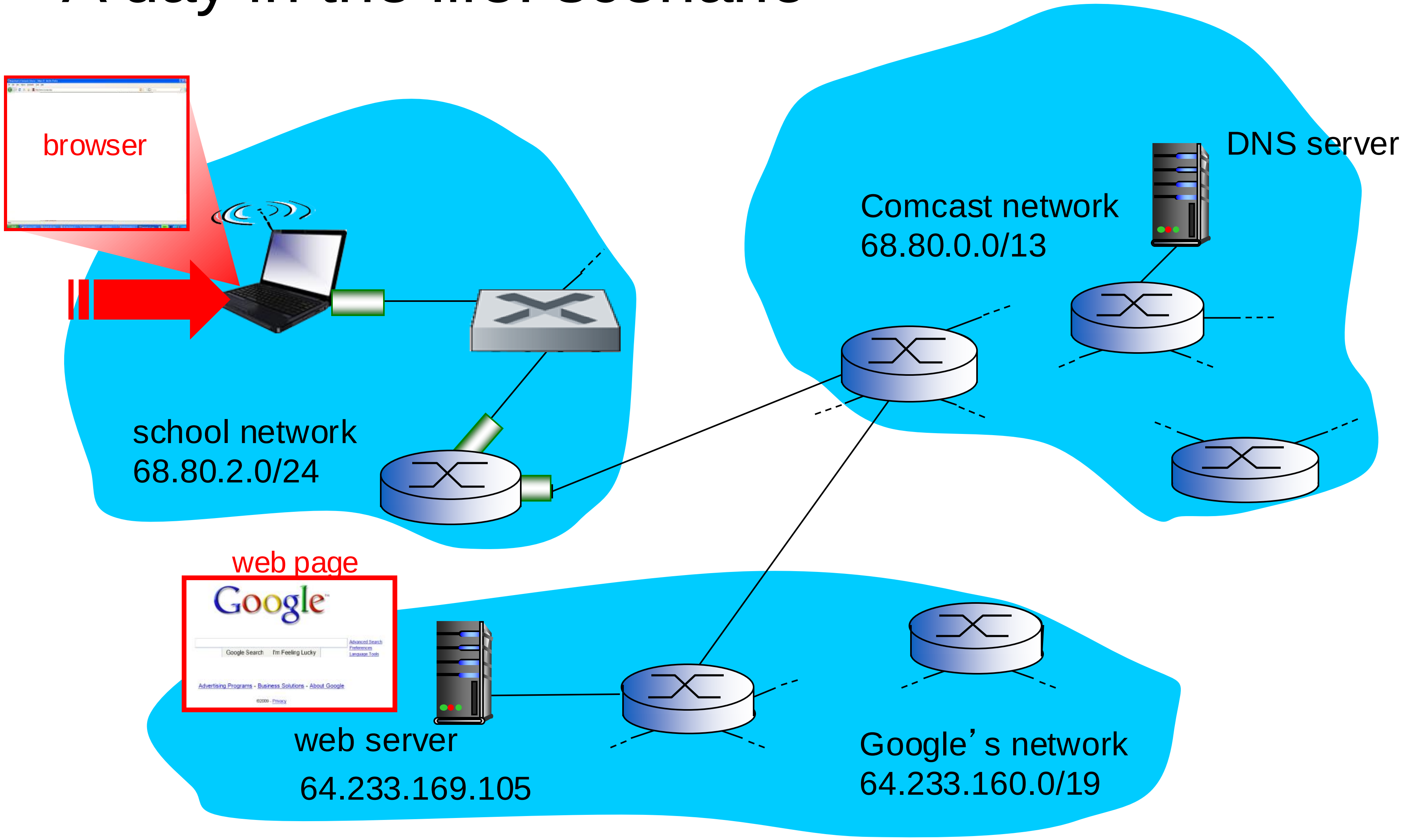
# Synthesis of protocols

# Synthesis: a day in the life of a web request

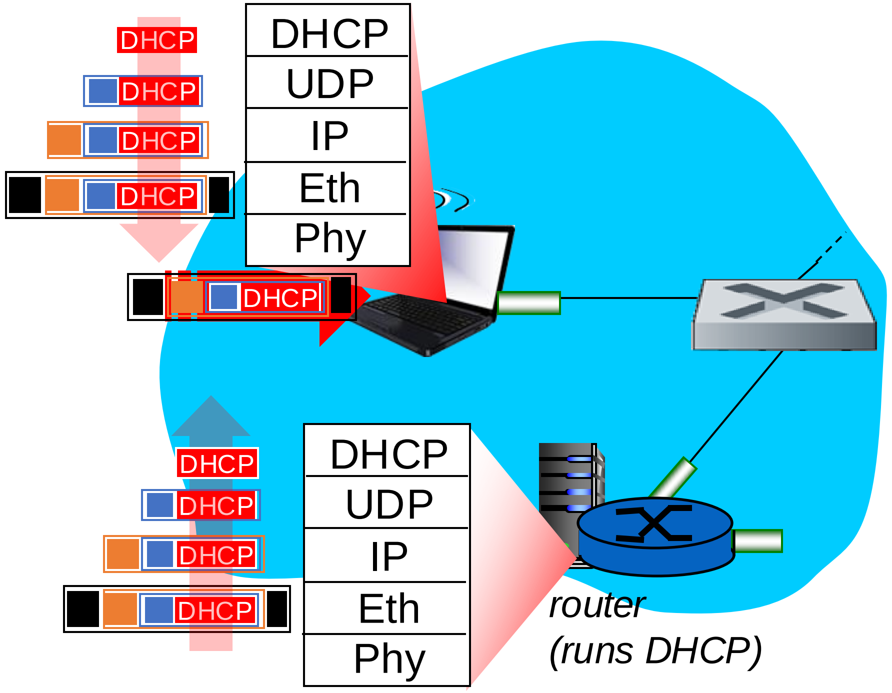
- Goal: identify, review, understand protocols (at all layers) involved in seemingly simple scenario: requesting www page

- Scenario: student attaches laptop to campus network, requests/receives www.google.com

# A day in the life: scenario



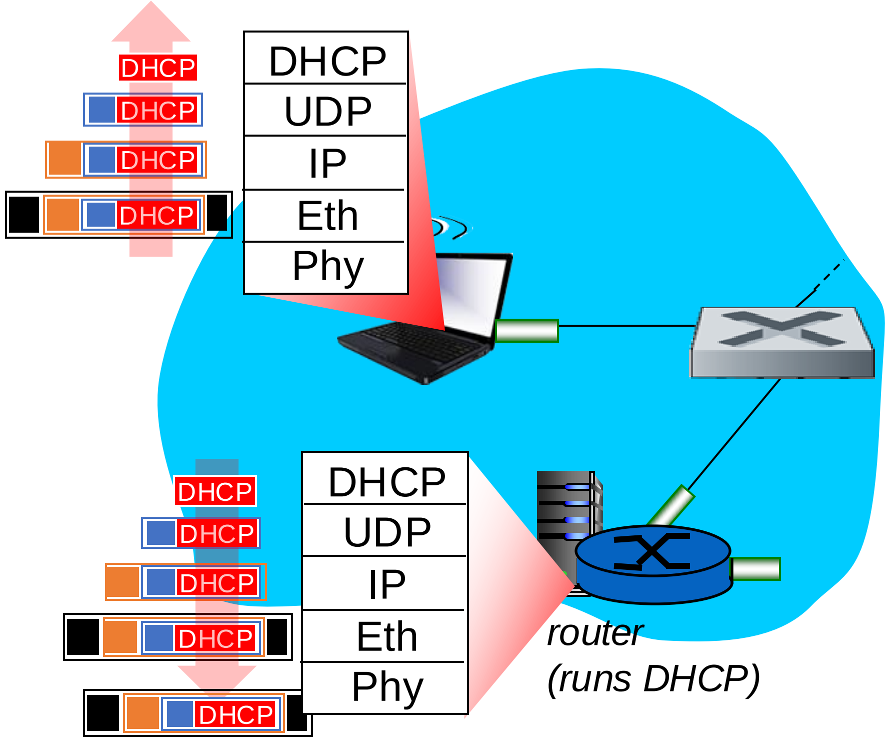
# A day in the life… connecting to the Internet




- connecting laptop needs to get its own IP address, addr of first-hop router, addr of DNS server: use *DHCP*

- DHCP request encapsulated in UDP, encapsulated in IP, encapsulated in link layer Ethernet
- Packet broadcast (dest: FFFFFFFFFFFF) on the local network, received at a router running DHCP server
- Ethernet decapsulated to IP decapsulated to UDP decapsulated to DHCP

# A day in the life… connecting to the Internet

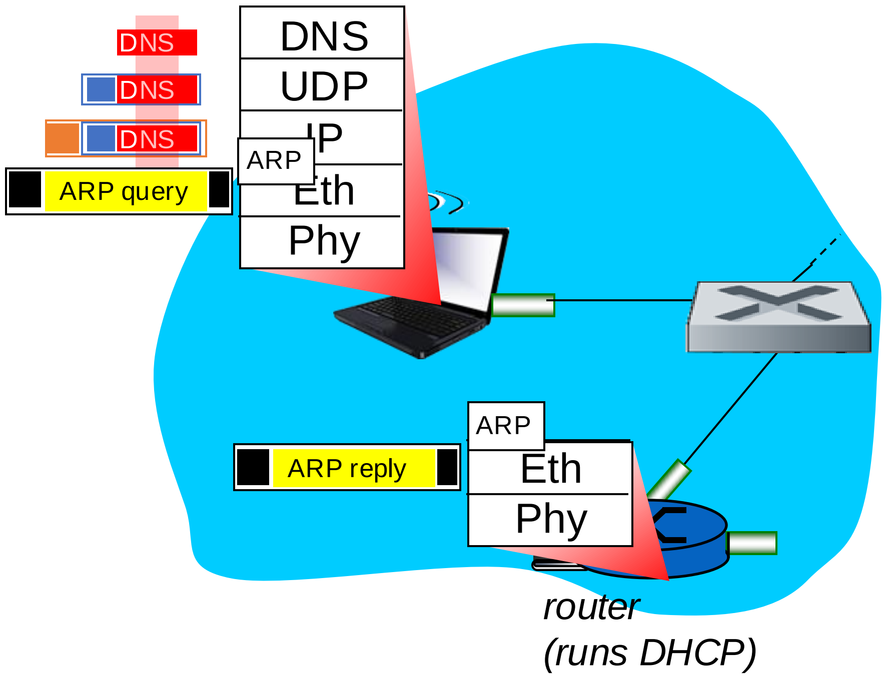



- DHCP server formulates *DHCP ACK* containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server

- DHCP client receives DHCP ACK reply

Client now has IP address, knows name & addr of DNS server, IP address of its first-hop router

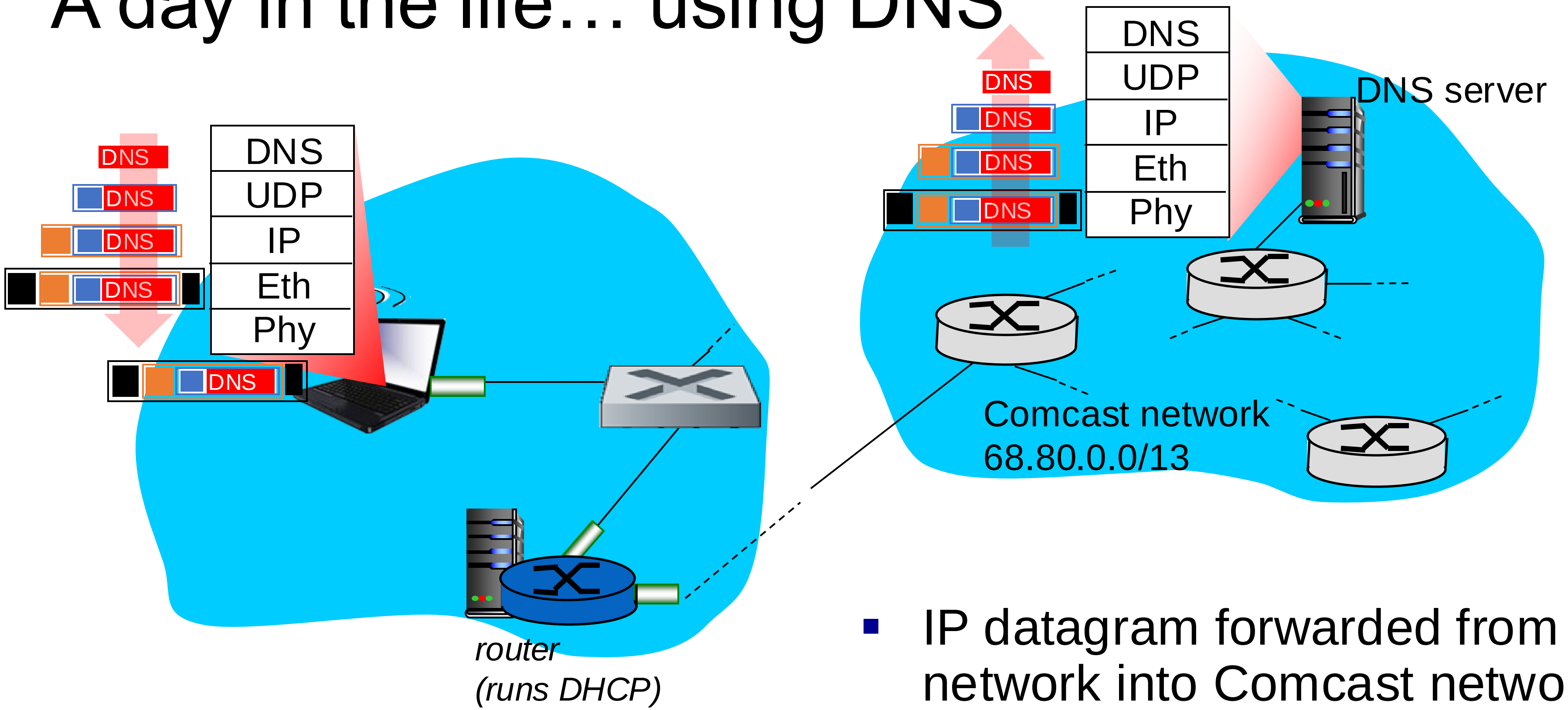
# A day in the life… ARP (before DNS, before HTTP)




- before sending *HTTP* request, need IP address of www.google.com: *DNS*

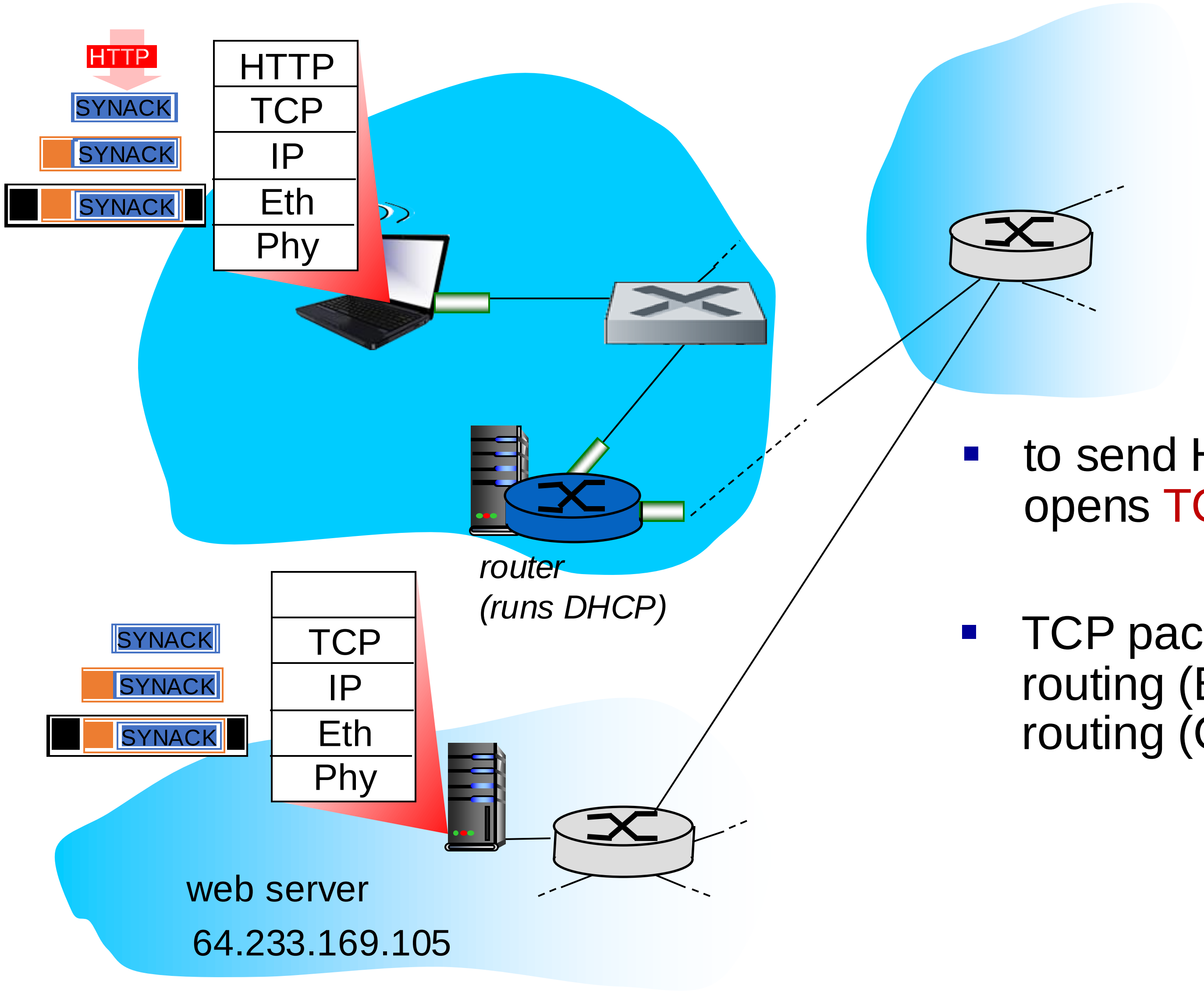
- DNS query created, encapsulated in UDP, encapsulated in IP, encapsulated in Eth. To send frame to router, need MAC address of router interface: ARP
- ARP query broadcast, received by router, which replies with ARP reply giving MAC address of router interface
- client now knows MAC address of gateway router, so can now send packet containing DNS query

# A day in the life… using DNS



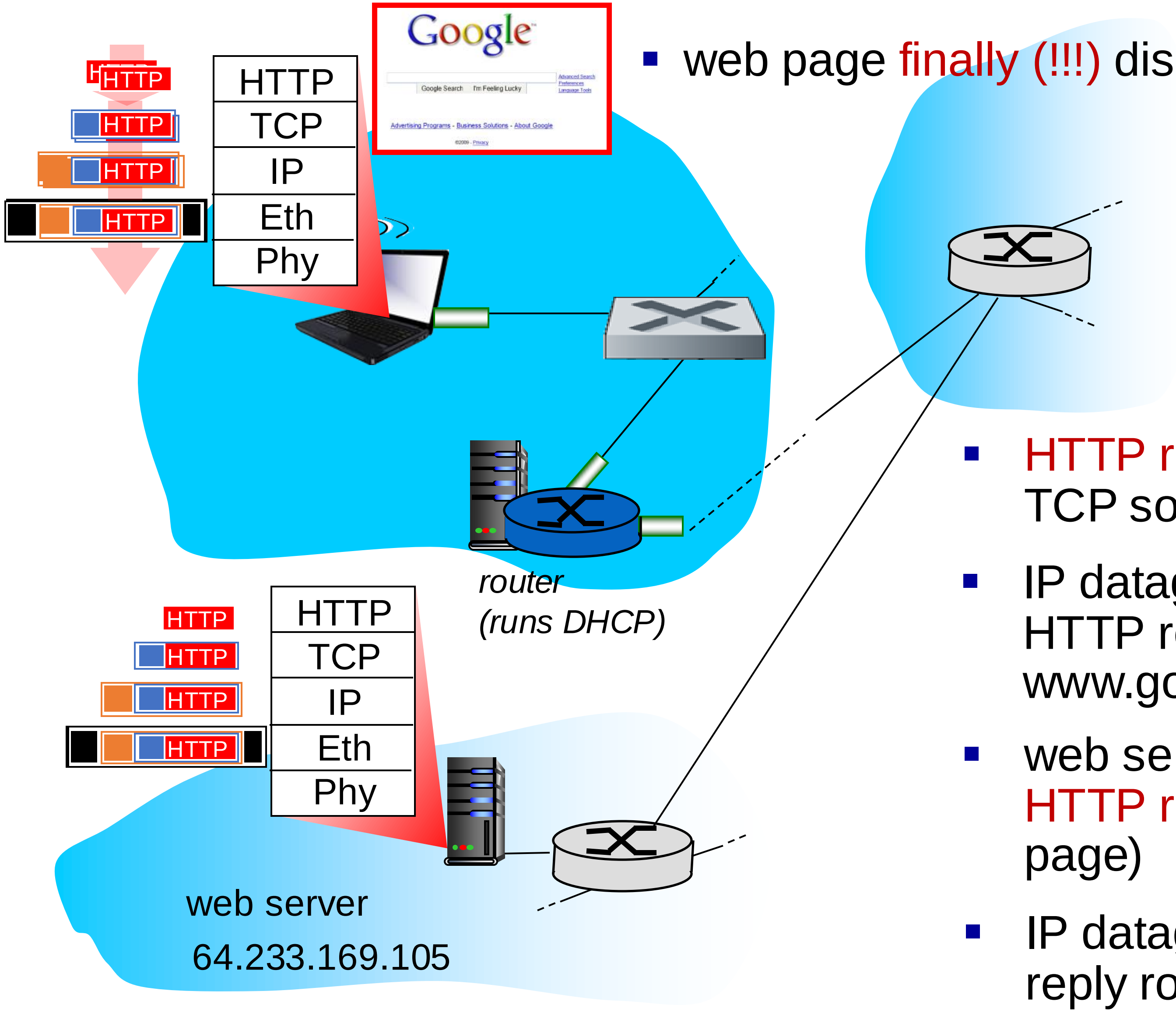

- IP datagram containing DNS query from client to gateway router
- IP datagram forwarded from campus network into Comcast network, routed (tables created by EIGRP, OSPF, and/or BGP routing protocols) to DNS server
- decapsulated to DNS server
- DNS server replies to client with IP address of www.google.com

# A day in the life…TCP connection carrying HTTP




- to send HTTP request, client first opens TCP socket to web server
- TCP packet routed using inter-domain routing (BGP) and intra-domain routing (OSPF, EIGRP) to web server

# A day in the life… HTTP request/reply




- web page finally (!!!) displayed
- HTTP request sent into TCP socket
- IP datagram containing HTTP request routed to www.google.com
- web server responds with HTTP reply (containing web page)
- IP datagram containing HTTP reply routed back to client

# Internet Technology

RUTGERS
UNIVERSITY | NEW BRUNSWICK

# Outro

- Computer networks are a stack of layers
  - Built for modularity
  - Each layer does one set of functions very well
  - Each layer depends on the layers beneath it

- Many general and useful principles
  - Applicable to real life (e.g., feedback control)
  - Applicable to computer system design (e.g., indirection & hierarchy)

# Outro: Now what?

- Go about life as usual
  - One difference: enhanced abilities to work with Internet-based tech
- Apply your new skills to solve a problem you care about
  - Tons of free and open-source software and platforms. Opportunities
- Deepen your understanding of the Internet and its tech
  - CS 553 Internet services (Spring 2025)
- Push the boundaries of Internet tech
  - Talk to me if you're interested in research