IOWA

CS3640

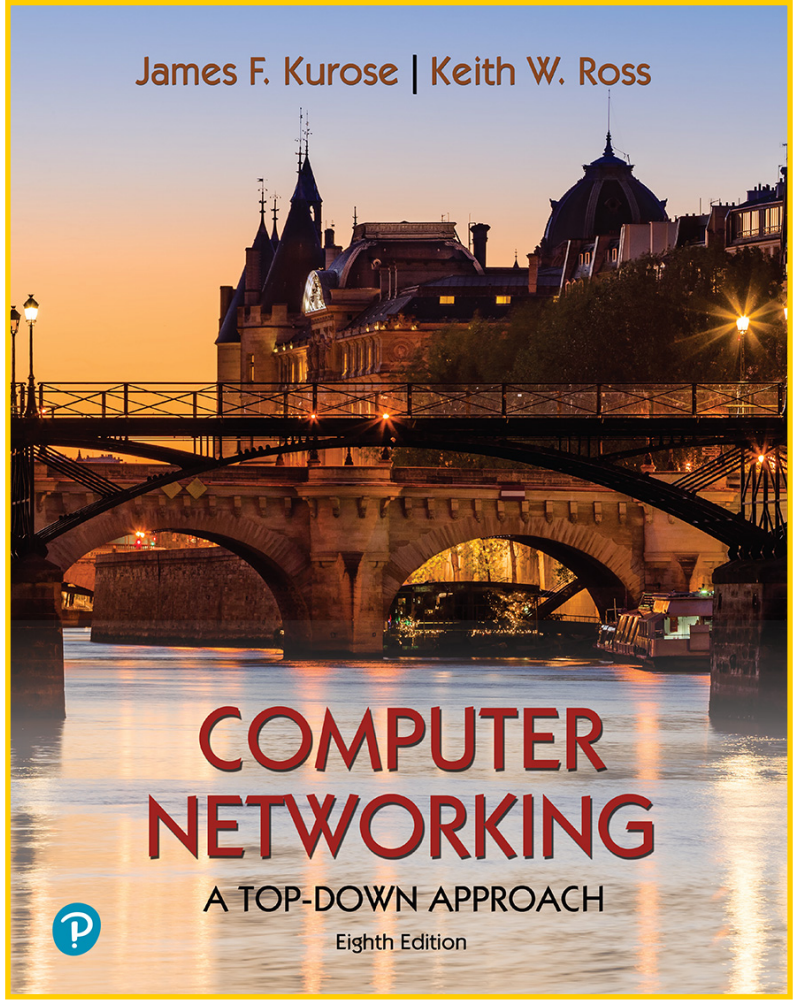
Network Layer (3): The Internet Protocol

Prof. Supreeth Shastri
Computer Science
The University of Iowa

Lecture goals

a two-part discussion on the Internet Protocol, its functionalities, shortcomings, and real-life solutions

- *IPv4 format and addressing*
- *Address management via DHCP*
- *NAT and Middleboxes*
- *IPv6*



Chapters 4.3, 4.5

IOWA

ICANN allocated the last chunk of IPv4 addresses in 2011

Then, how do new hosts obtain and manage their IP addresses?

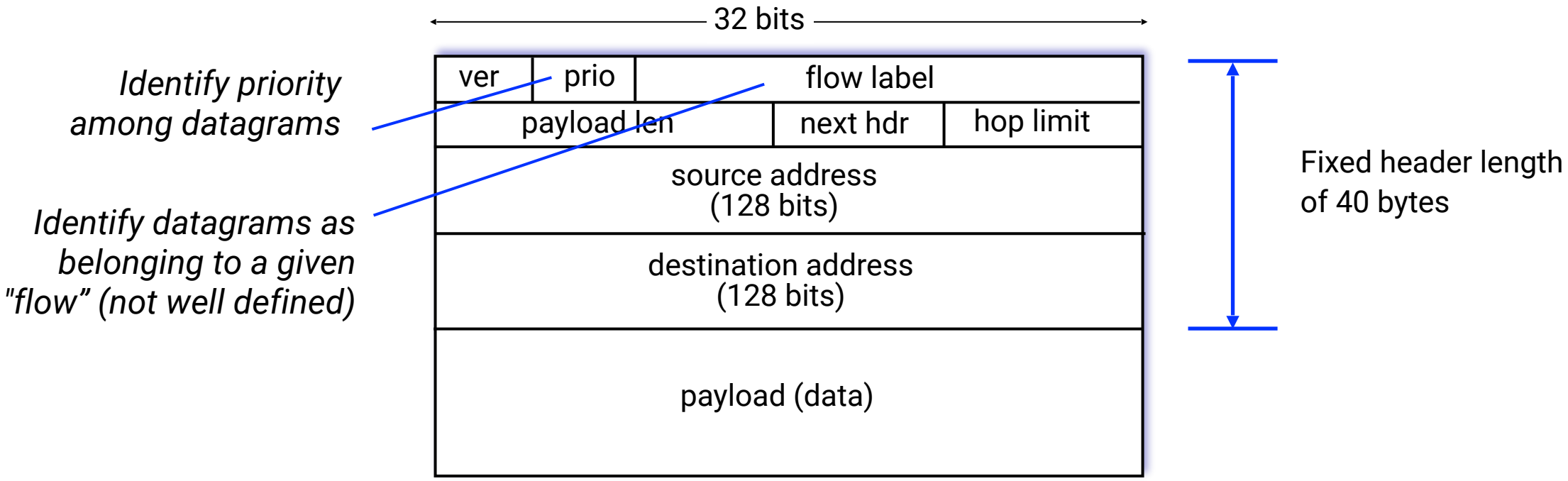
1. Create a new version of the Internet Protocol w/ larger range of addresses

2. Figure out a way to reuse the existing 32-bit address space

IPv6

(or what the Internet visionaries proposed)

IPv6 Datagram Format



Original Motivation

Increase available address space from 2^{32} (4 billion) to 2^{128} (340 trillion trillion trillion)

Additional Motivation

no checksum, no options, no fragmentation or reassembly ⇒ faster packet processing at routers

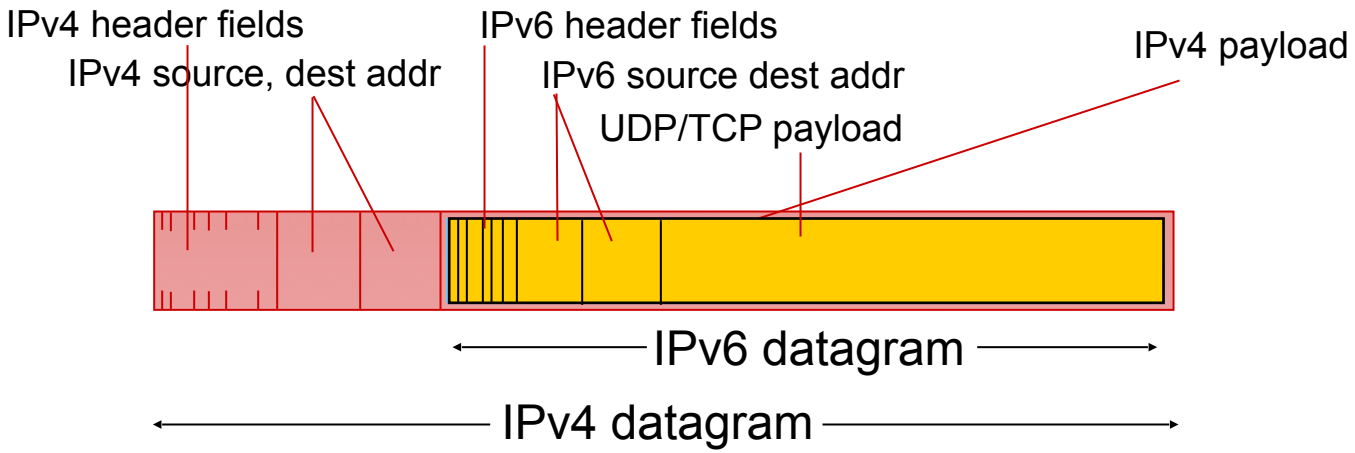
Transition from IPv4 to IPv6

Not all routers can or will upgrade simultaneously

- The Internet had only one “flag day”: 1/1/1983 when all ARPANET hosts switched from NCP to TCP/IP
- So, how will the Internet operate with mixed IPv4 and IPv6 routers?

Tunneling

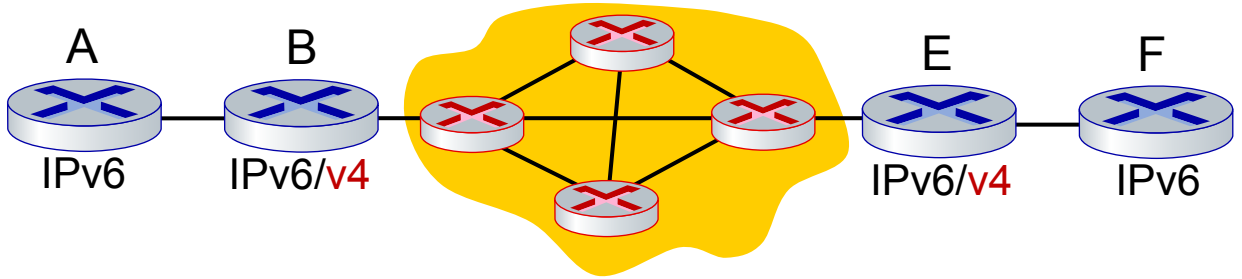
- **Key idea**: carry IPv6 datagram as payload in IPv4 datagram among IPv4 routers
- The concept is used extensively in other contexts such as 3G/4G/5G networks



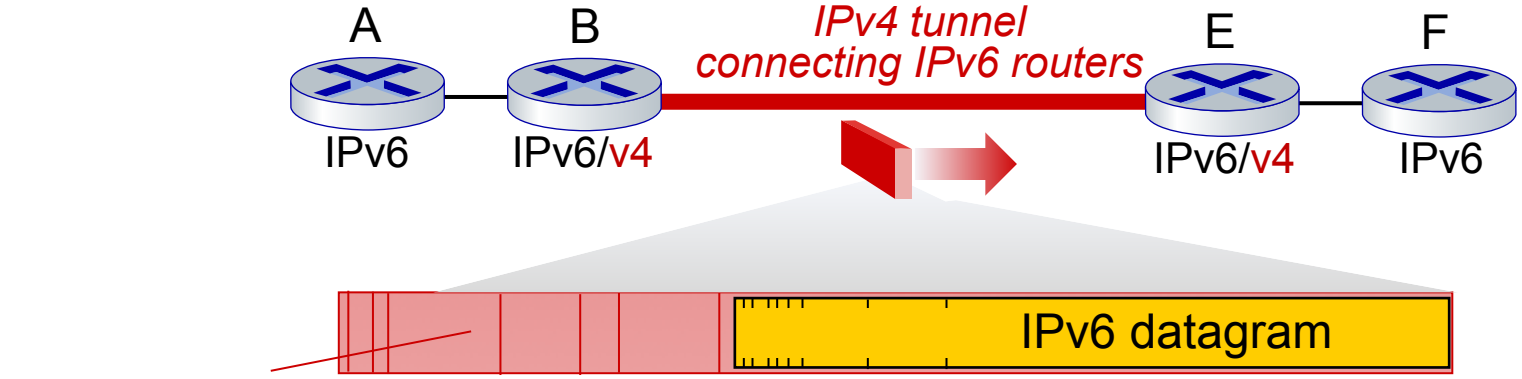
Tunneling and Encapsulation

The usual: IP datagram sent as payload in link-layer frame

Ethernet connecting two IPv6 routers

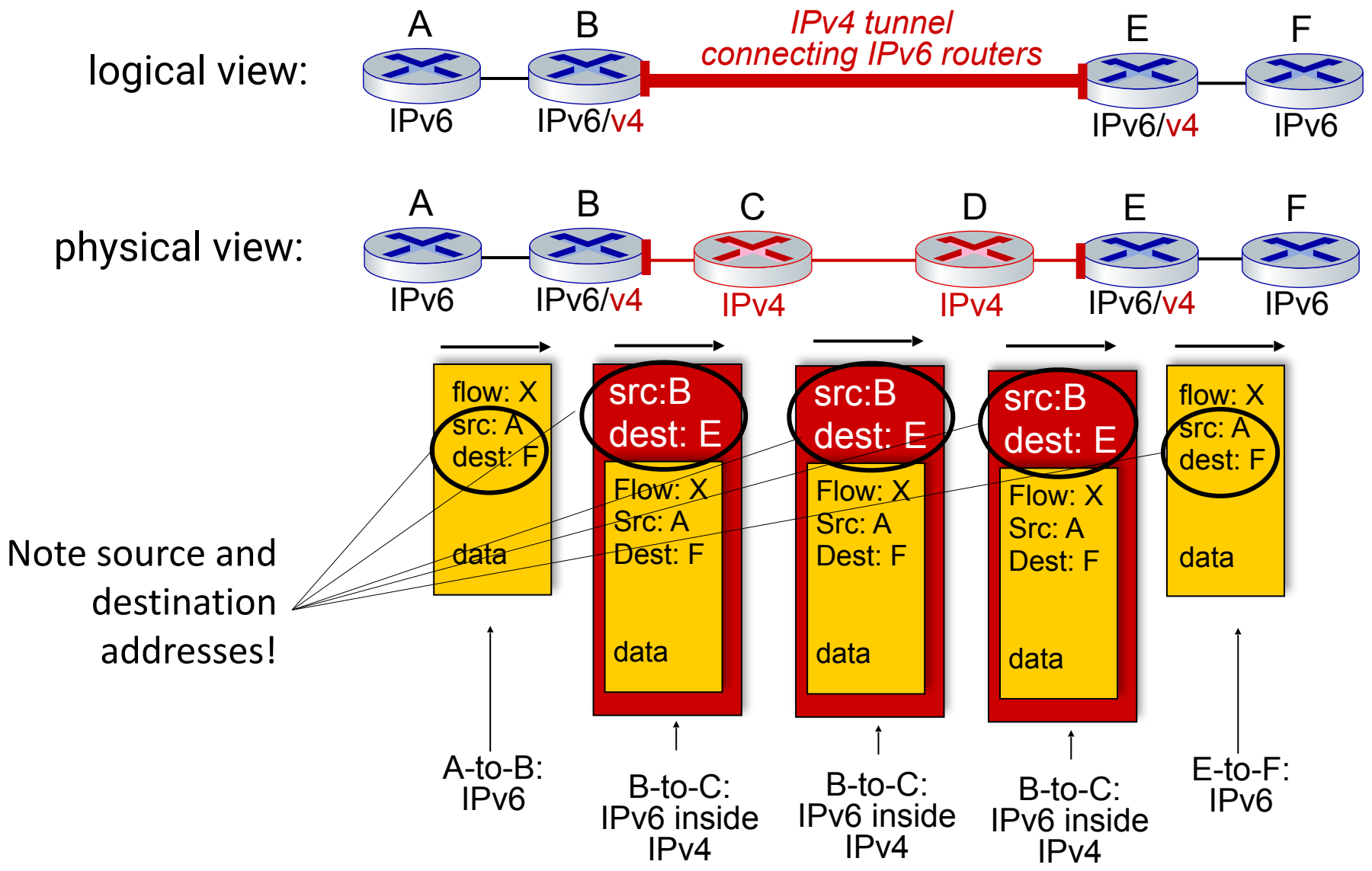


IPv4 network connecting two IPv6 routers



tunneling: IPv6 datagram as payload in a IPv4 datagram

Tunneling


logical view:
A
B
IPv4 tunnel
connecting IPv6 routers
E
F
IPv6
IPv6/v4
IPv6/v4
IPv6
physical view:
A
B
C
D
E
F
IPv6
IPv6/v4
IPv4
IPv4
IPv6/v4
IPv6
Note source and destination addresses!
flow: X
src: A
dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
flow: X
src: A
dest: F
data
A-to-B:
IPv6
B-to-C:
IPv6 inside
IPv4
B-to-C:
IPv6 inside
IPv4
B-to-C:
IPv6 inside
IPv4
E-to-F:
IPv6

IPv6 Slow Adoption

30%

client access to Google search are via IPv6

33%

of all US government domains are IPv6 capable

25

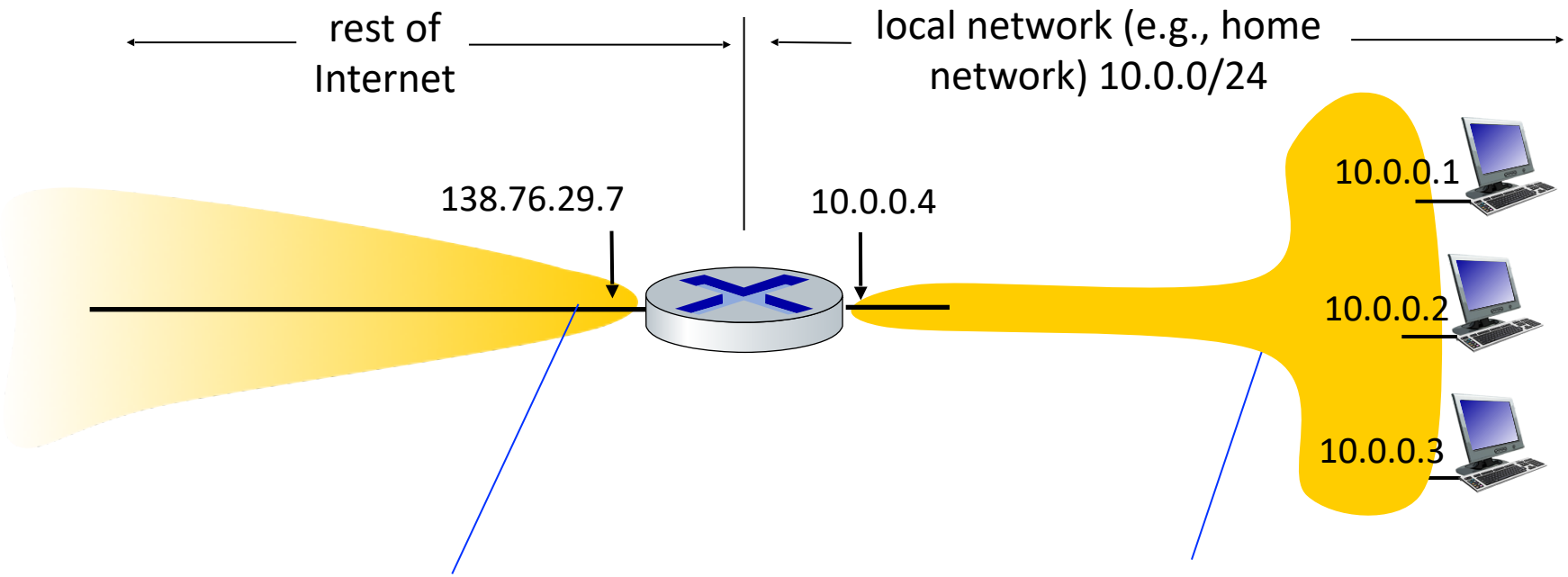
years since IPv6 was standardized

NAT

(or how folks actually solved the problem in the real world)

Network Address Translation (NAT)

All devices in local network share just one IPv4 address as far as outside world is concerned



All datagrams **leaving** local network have the NAT IP address (138.76.29.7) as their source, but have different source port numbers

datagrams with destination within this network have 10.0.0/24 address for source and destination (as usual)

Network Address Translation (NAT)

All devices in local network can have addresses from the "private" IP address space (10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16) that can only be used in local network

How is this useful?

- just one IP address needed from provider ISP for all devices
- can change addresses of host in local network without notifying outside world
- can change ISP without changing addresses of devices in local network
- [bonus] **security**: devices internal to the local network are neither directly addressable nor visible to the outside world

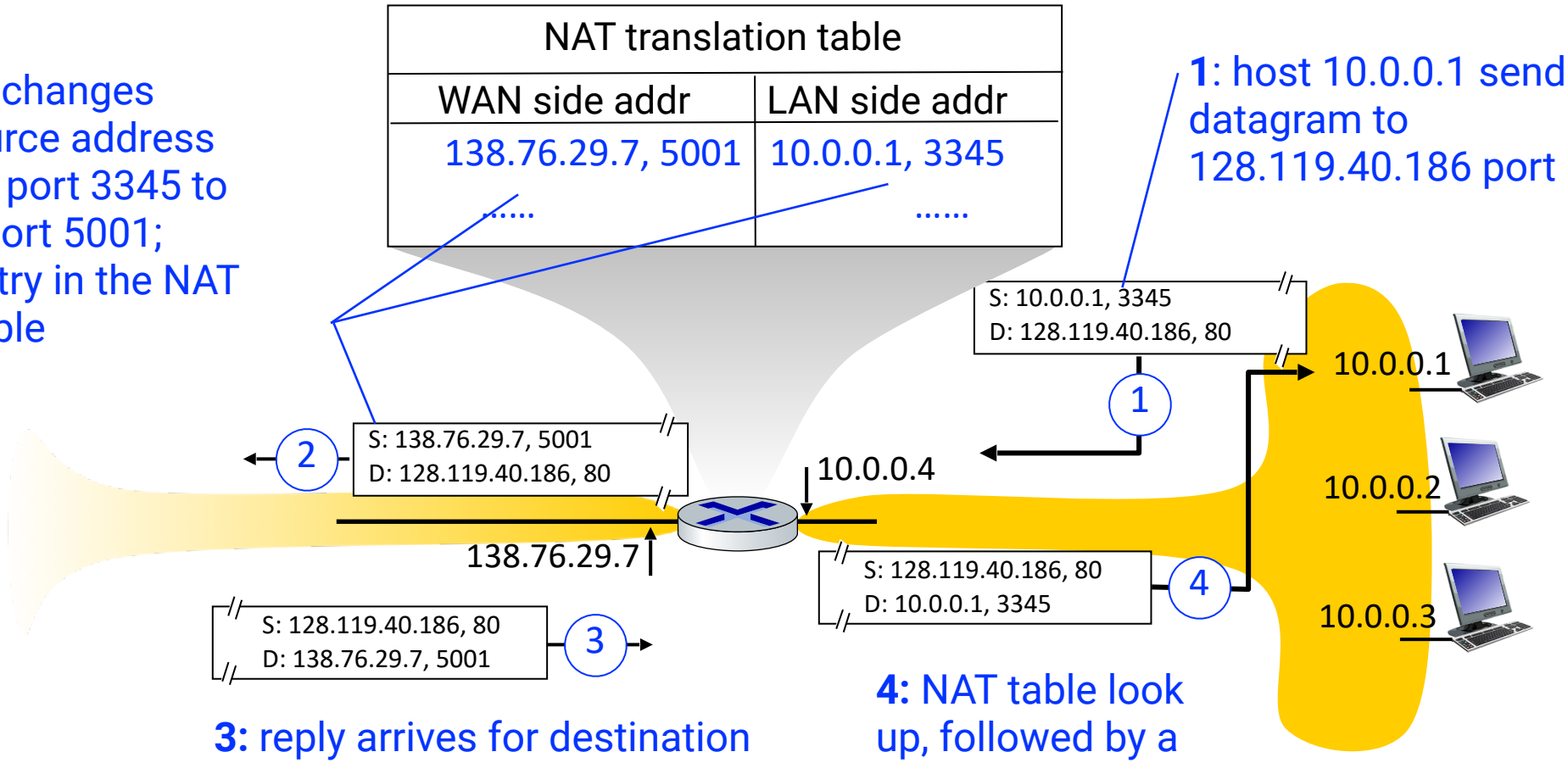
Implementation of NAT

A NAT router must (transparently) perform the following:

1. **For all outgoing datagrams**: replace (source IP address, port #) to (NAT IP address, new port #). Remote clients/servers will perceive (NAT IP address, new port #) as the end host they are communicating with, and will address their packets to that.
2. **Maintain a NAT translation table**: record all mappings from (source IP address, port #) to (NAT IP address, new port #) in a look up table.
3. **For all incoming datagrams**: replace (NAT IP address, new port #) in destination field of every incoming datagram with the corresponding (source IP address, port #) stored in NAT table.

Implementation of NAT

2: NAT router changes datagram source address from 10.0.0.1 port 3345 to 138.76.29.7 port 5001; creates an entry in the NAT translation table



1: host 10.0.0.1 sends datagram to 128.119.40.186 port 80

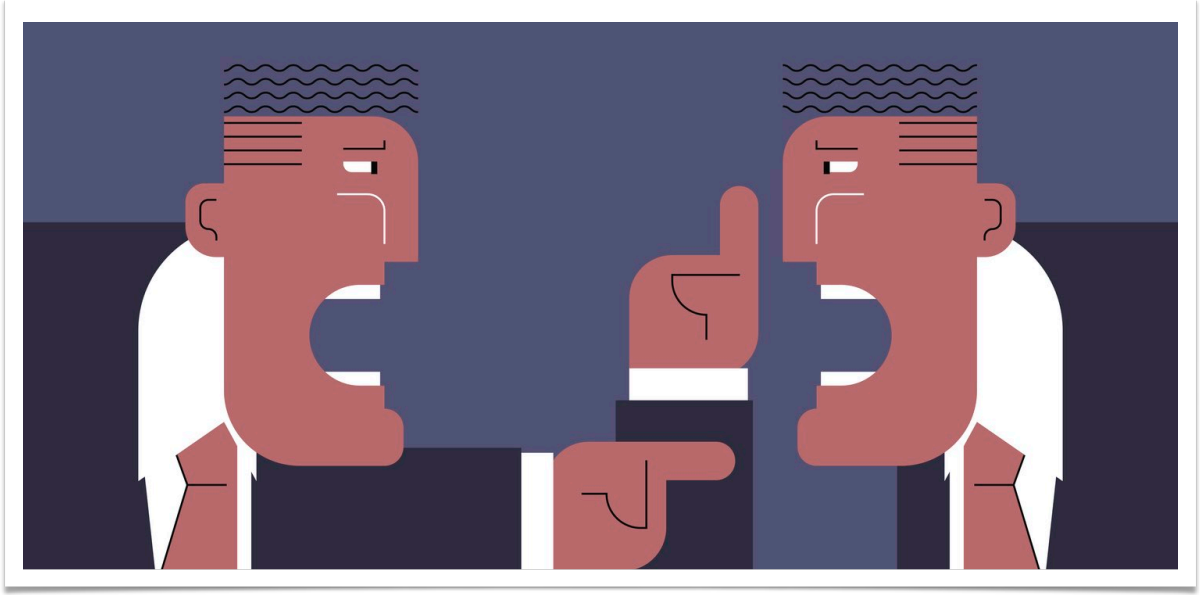
3: reply arrives for destination address: 138.76.29.7 port 5001

4: NAT table look up, followed by a source IP and port number rewrite

Since early days till now

NAT has been

CONTROVERSIAL



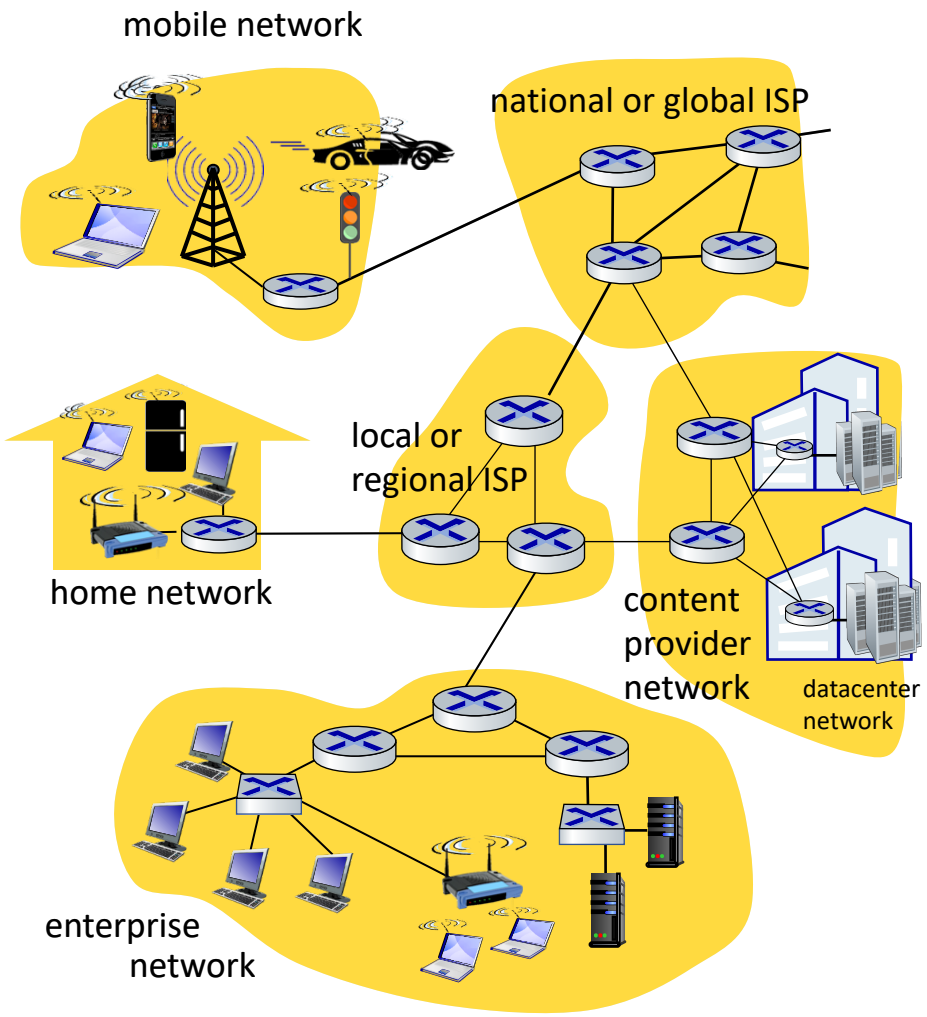
- routers “should” only process up to layer 3
- address “shortage” should be solved by IPv6
- violates end-to-end argument (port # manipulation by network-layer device)
- NAT traversal: what if client wants to connect to server behind NAT?

Middleboxes

(or why stop at NAT when one can rock the boat harder!)

Middlebox (RFC 3234)

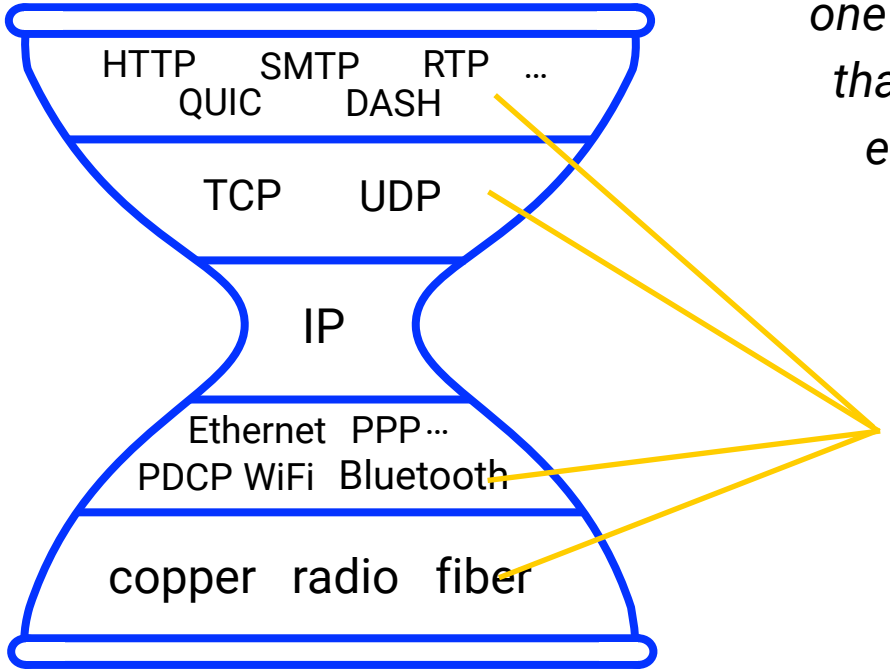
any intermediary box performing functions apart from normal, standard functions of an IP router on the data path between a source host and destination host



Middleboxes are everywhere!

- **NAT**: home, mobile, enterprise networks
- **Firewalls** and **Intrusion detection**: enterprise networks
- **Load balancers**: service providers, mobile networks
- **Network Function Virtualization (NFV)**

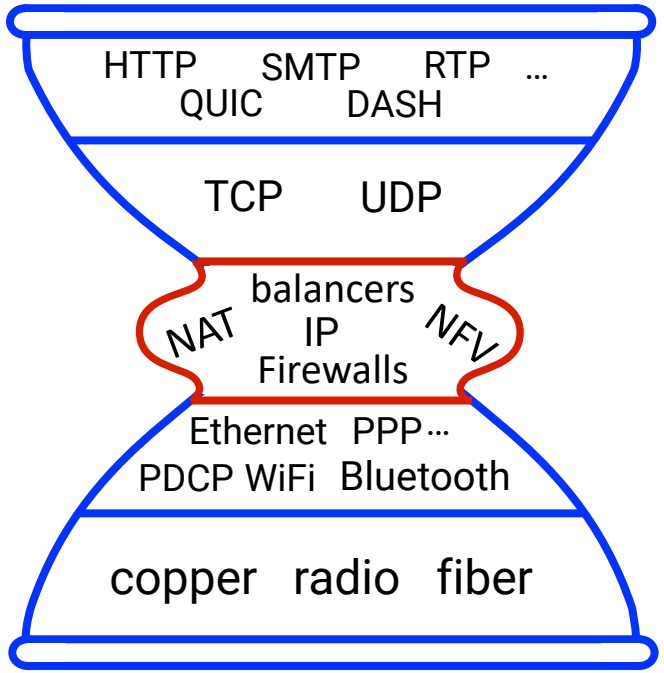
The IP hourglass: *An Organizing Principle for Internet Protocols*



Internet's "thin waist"

*one core network layer protocol that **must** be implemented by every (billions of) Internet-connected device*

***allows many* protocols** in physical, link, transport, and application layers



As the Internet enters its "**middle age**", its waist has expanded!

Spot Quiz (ICON)