Information Network

Lecture 10: Network Layer – Data Plane

Holger Thies

KYOTO UNIVERSITY

京都大学

Chapter 3 outline

3.1 transport-layer services

3.2 multiplexing and demultiplexing

3.3 connectionless transport: UDP

3.4 principles of reliable data transfer

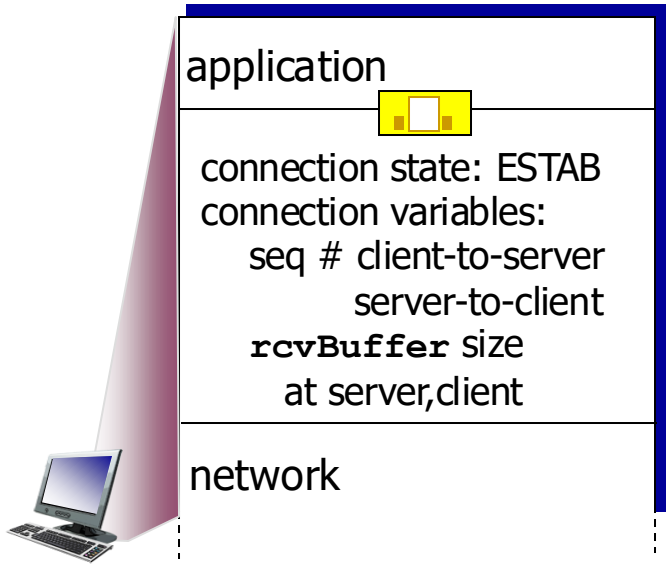
3.5 connection-oriented transport: TCP

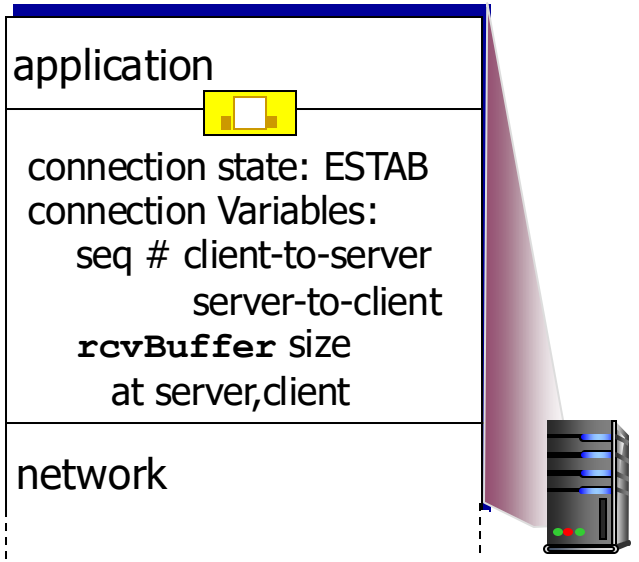
- segment structure
- reliable data transfer
- flow control
- connection management
- congestion control

Connection Management

before exchanging data, sender/receiver “handshake”:

- agree to establish connection (each knowing the other willing to establish connection)
- agree on connection parameters

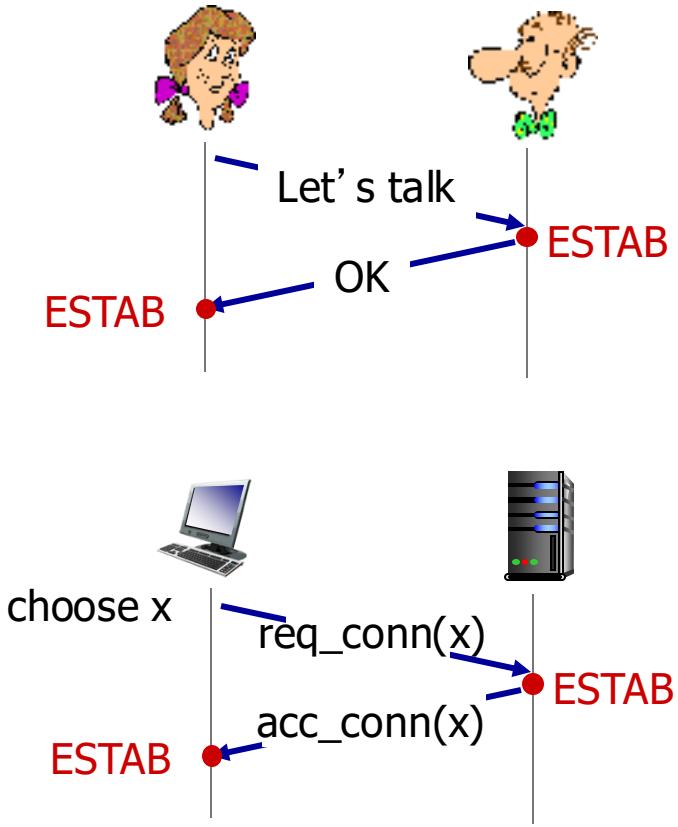




京都大学

Agreeing to establish a connection

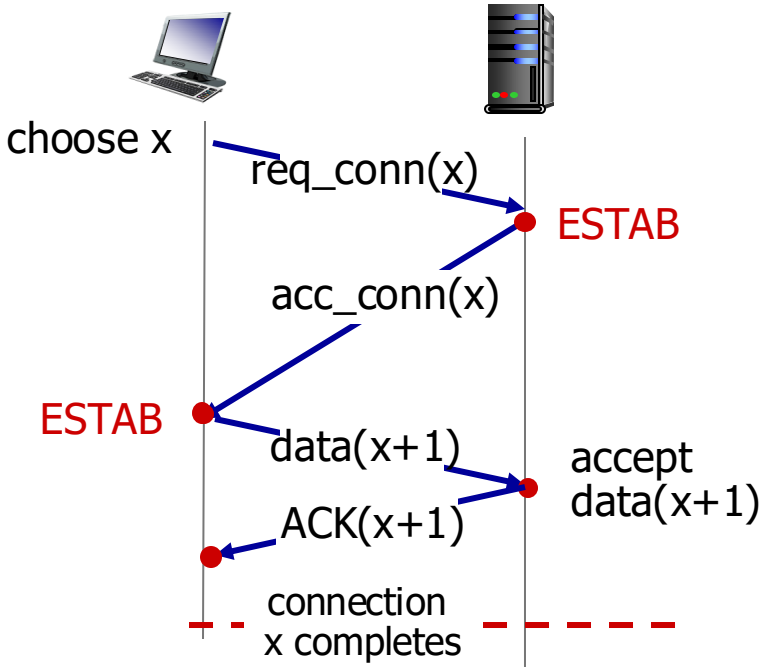
2-way handshake:



Q: will 2-way handshake always work in network?

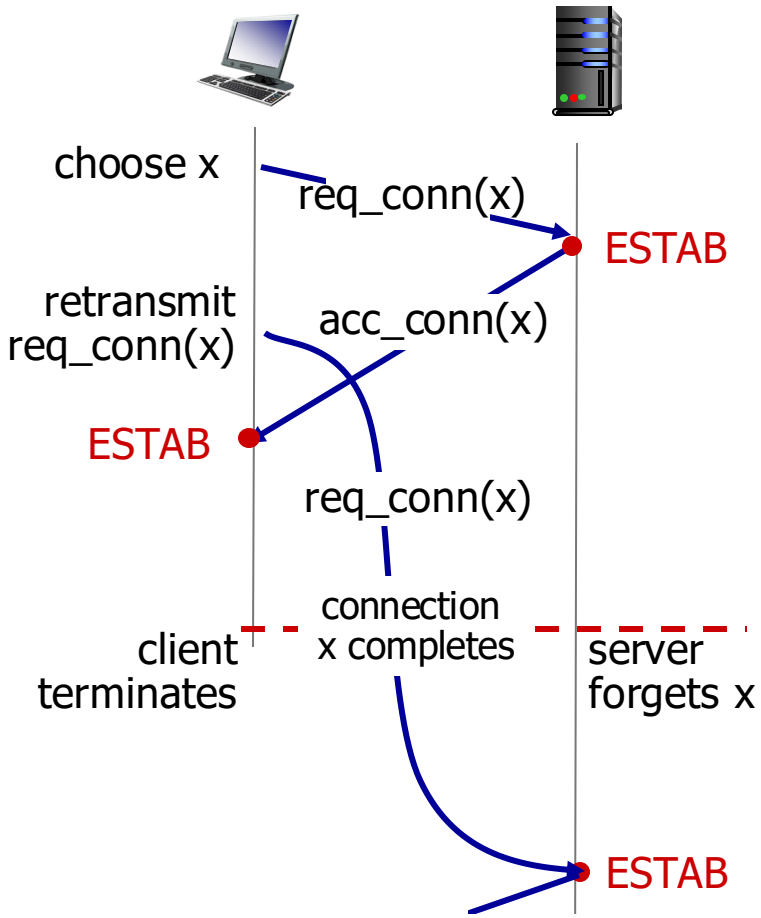
- variable delays
- retransmitted messages (e.g. due to message loss)
- message reordering
- can't "see" other side

2-way handshake scenarios



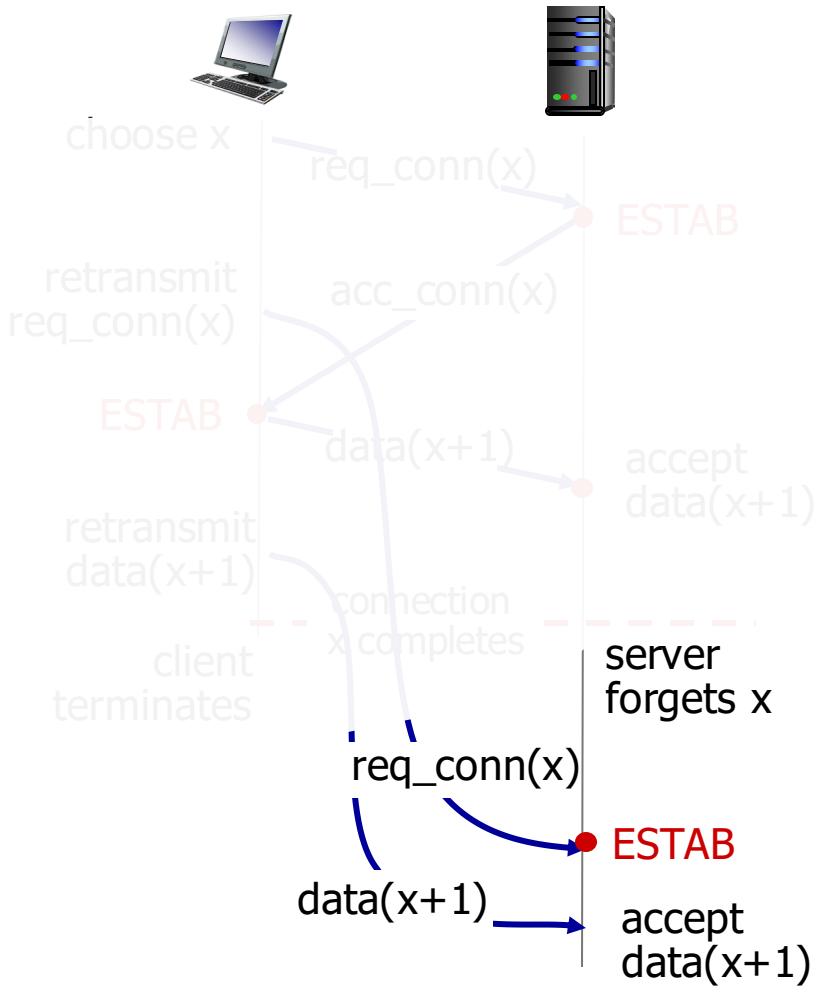
No problem!

2-way handshake scenarios



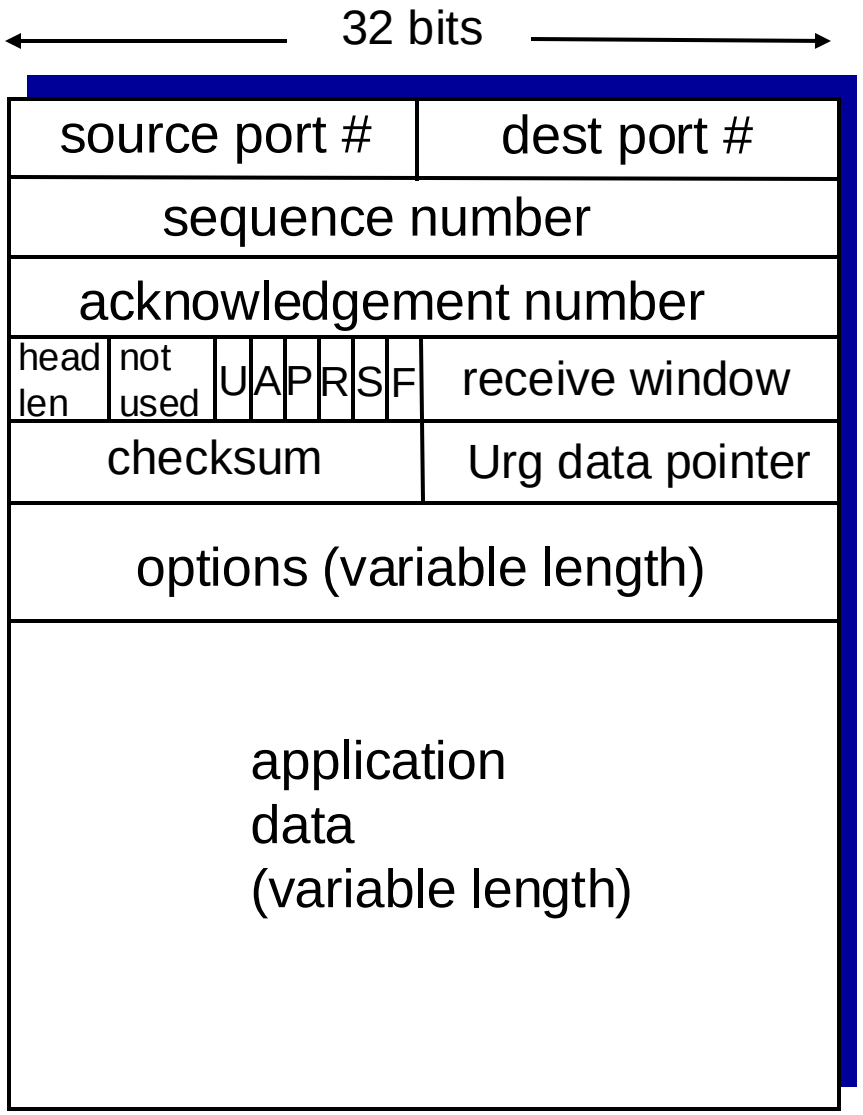
Problem: half open connection! (no client)

2-way handshake scenarios

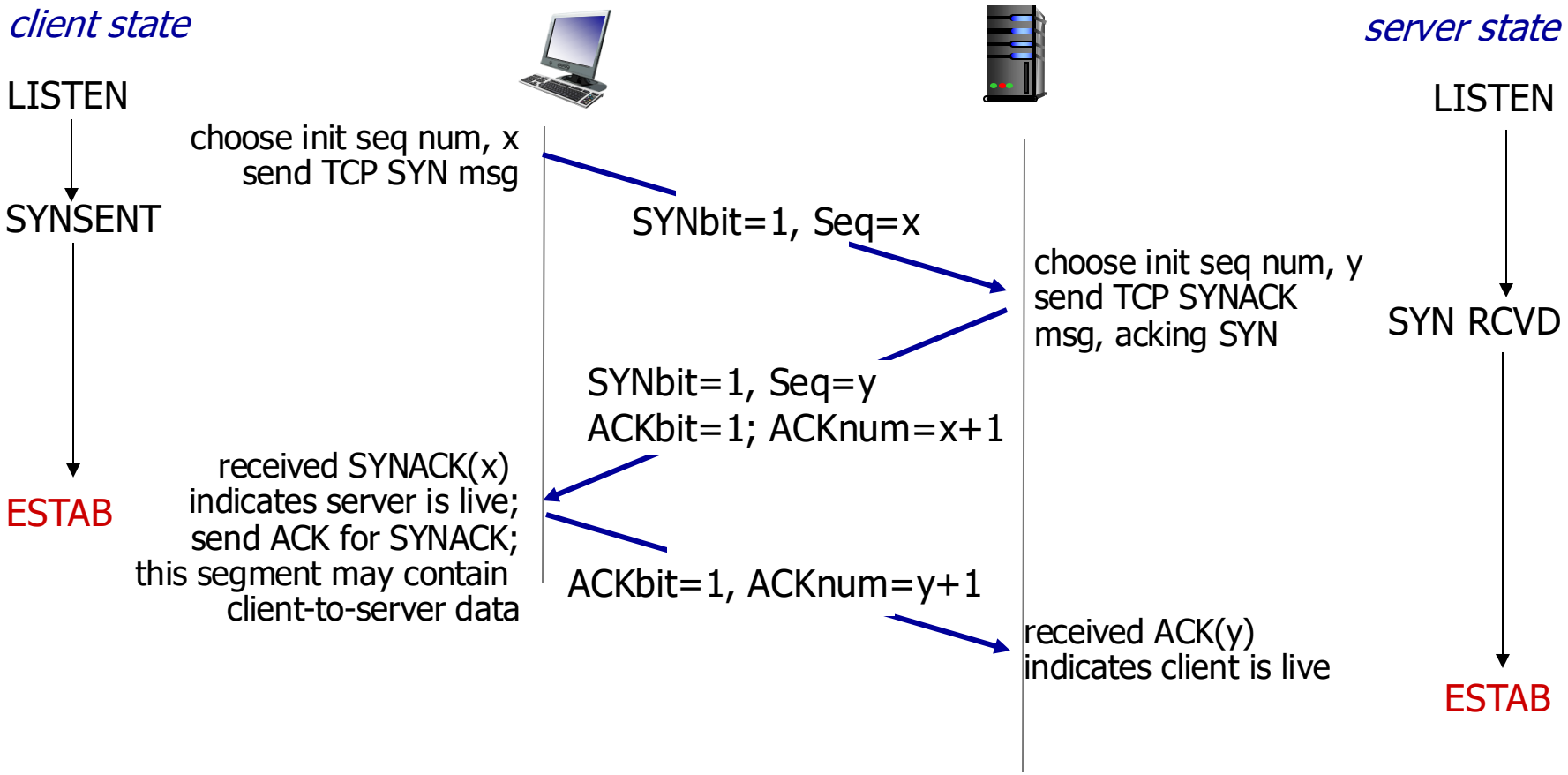


Problem: duplicate data accepted!

TCP segment structure



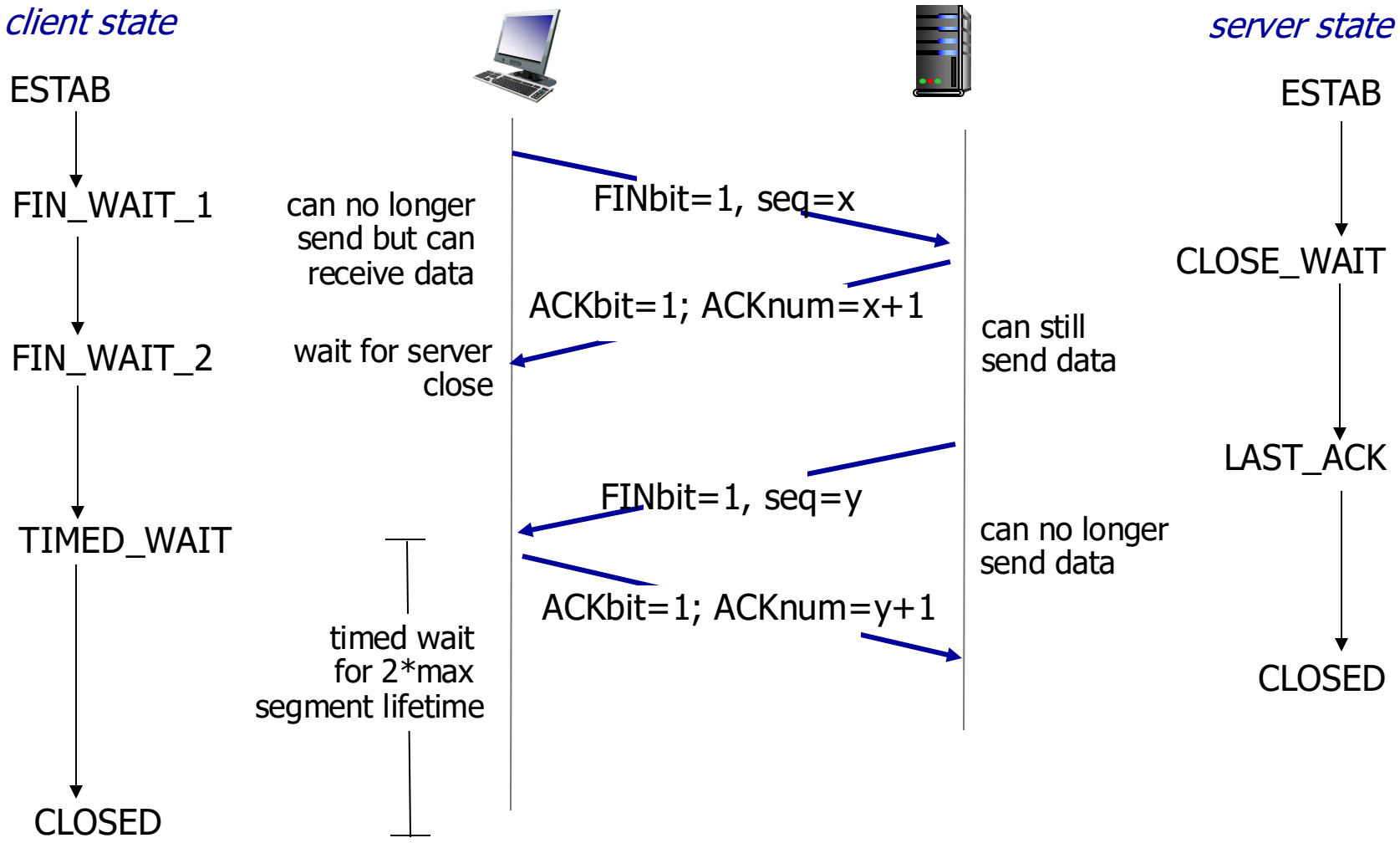
TCP 3-way handshake



TCP: closing a connection

- client, server each close their side of connection
  - send TCP segment with FIN bit = 1
- respond to received FIN with ACK
  - on receiving FIN, ACK can be combined with own FIN

TCP: closing a connection



Transport layer: summary

- principles behind transport layer services:
  - multiplexing, demultiplexing
  - reliable data transfer
  - flow control
  - congestion control
- instantiation, implementation in the Internet
  - UDP
  - TCP

next:

- leaving the network “edge” (application, transport layers)
- into the network “core”

Today's lecture

4.1 Overview of Network layer

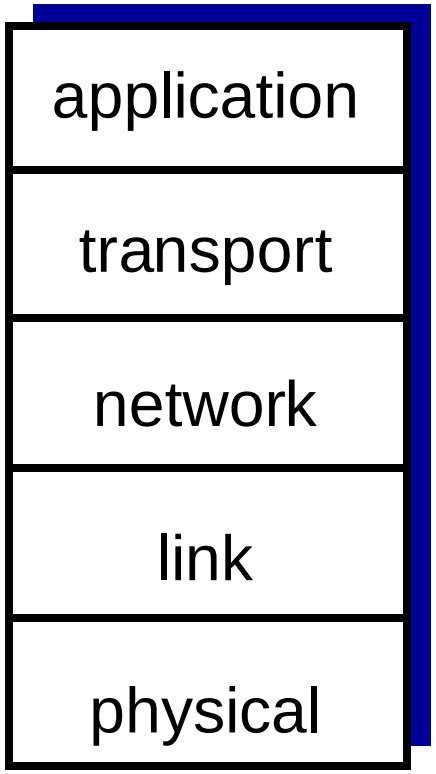
- data plane
- control plane

4.2 What's inside a router

4.3 IP: Internet Protocol

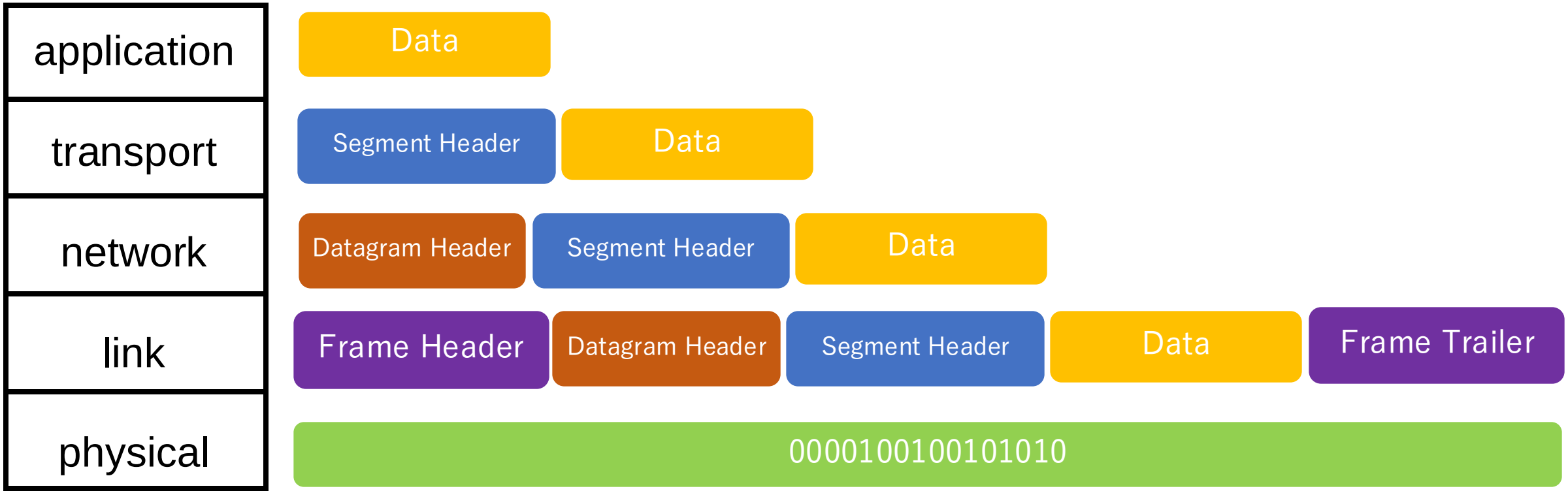
Internet protocol stack

- *application:* supporting network applications
  - FTP, SMTP, HTTP
- *transport:* process-process data transfer
  - TCP, UDP
- *network:* routing of datagrams from source to destination
  - IP, routing protocols
- *link:* data transfer between neighboring network elements
  - Ethernet, 802.111 (WiFi), PPP
- *physical:* bits "on the wire"



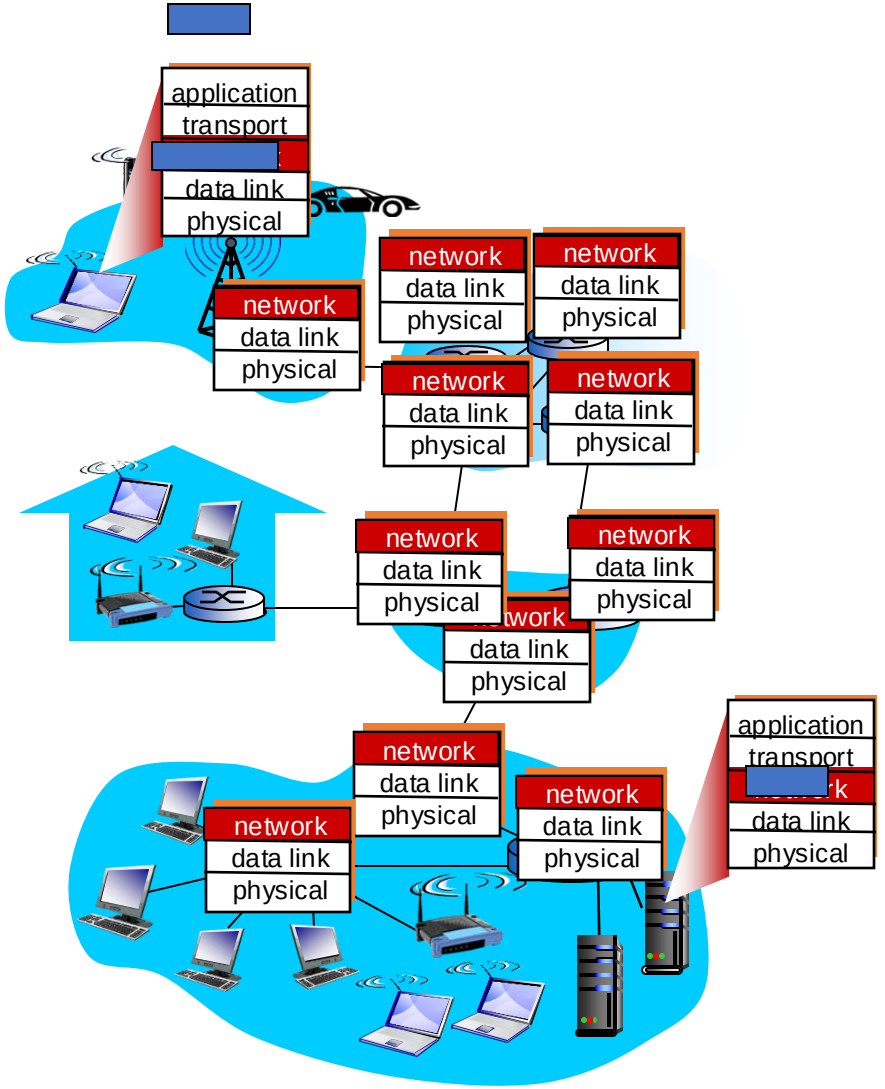
京都大学

Packets, Segments, Datagrams, etc.



Network layer

- transport segment from sending to receiving host
- on sending side encapsulates segments into datagrams
- on receiving side, delivers segments to transport layer
- network layer protocols in *every* host, router
- router examines header fields in all IP datagrams passing through it



京都大学

Two key network-layer functions

network-layer functions:

- *forwarding:* move packets from a router's input link to appropriate router output link
- *routing:* determine route taken by packets from source to destination
  - *routing algorithms*

analogy: taking a trip

- *forwarding:* process of getting through single interchange
- *routing:* process of planning trip from source to destination

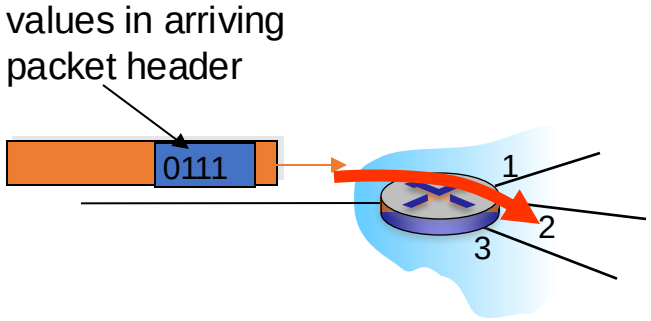
forwarding

routing

Network layer: data plane, control plane

Data plane

- local, per-router function
- determines how datagram arriving on router input port is forwarded to router output port
- forwarding function



Control plane

- network-wide logic
- determines how datagram is routed among routers along end-end path from source host to destination host
- two control-plane approaches:
  - *traditional routing algorithms:* implemented in routers
  - *software-defined networking (SDN)*: implemented in (remote) servers

Network-layer service model

Network Architecture	Service Model	Quality of Service (QoS) Guarantees ?			
		Bandwidth	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

京都大学

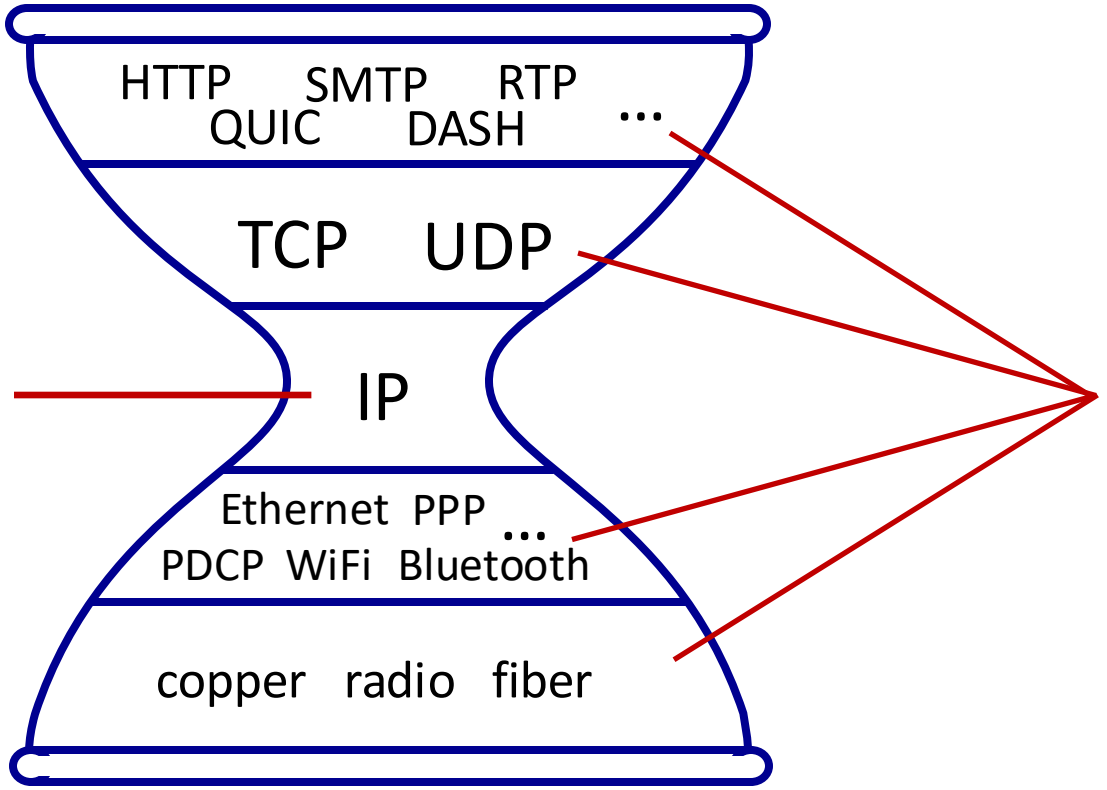
Best effort service

- IP does not provide any delivery guarantees, error correction or assurance of bandwidth, delay, or packet order.
- IP was designed to be simple and lightweight, focusing only on delivering data packets.
  - Reduces processing overhead and makes the system scalable to billions of devices
- IP assumes that applications requiring reliability will manage it themselves using higher-layer protocols like TCP.
  - Keeps the core network efficient while allowing different applications to implement custom reliability features.

The IP hourglass

Internet's "thin waist":

- *one* network layer protocol: IP
- *must* be implemented by every (billions) of Internet-connected devices



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

Today's lecture

4.1 Overview of Network layer

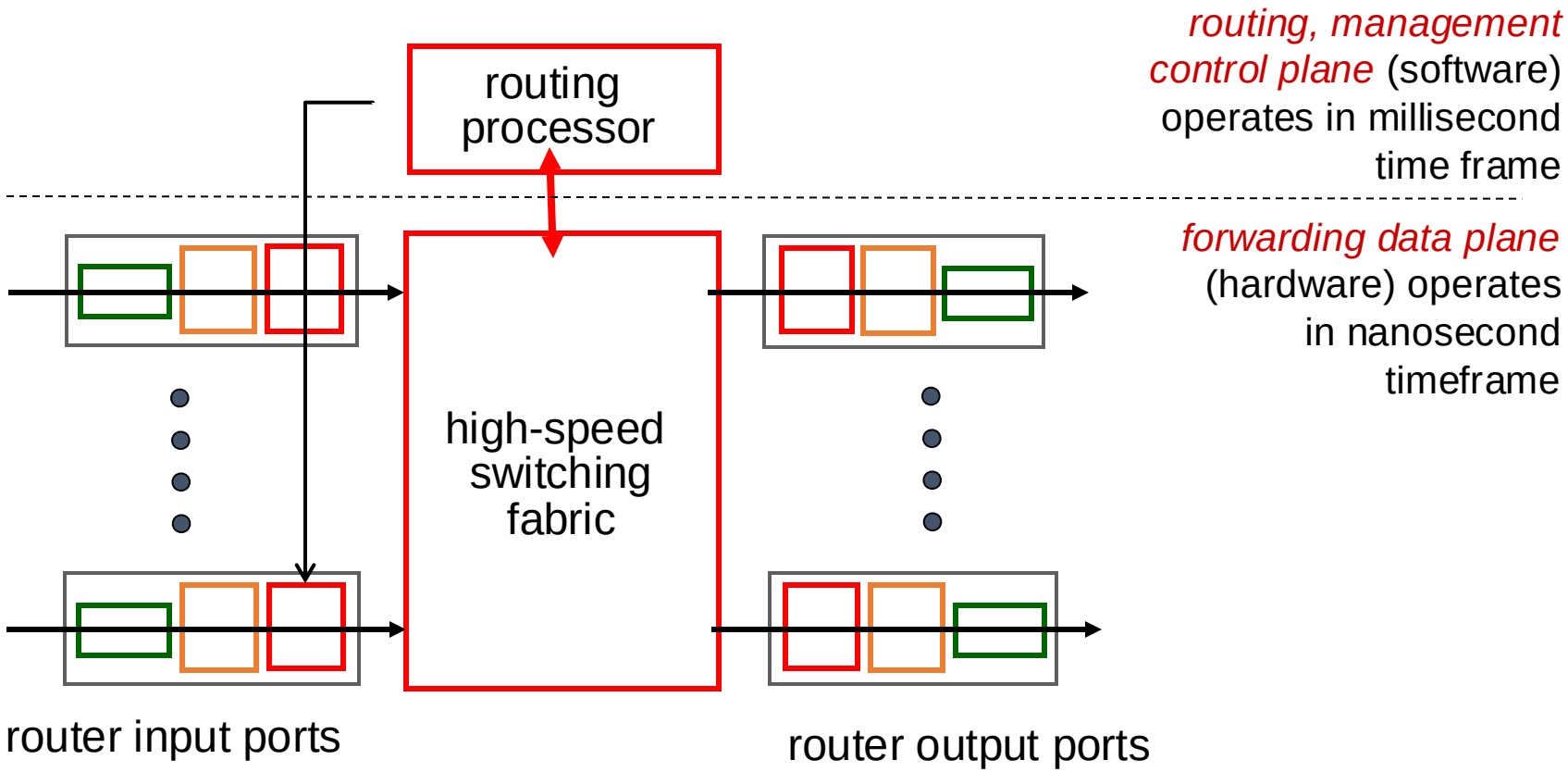
- data plane
- control plane

4.2 What's inside a router

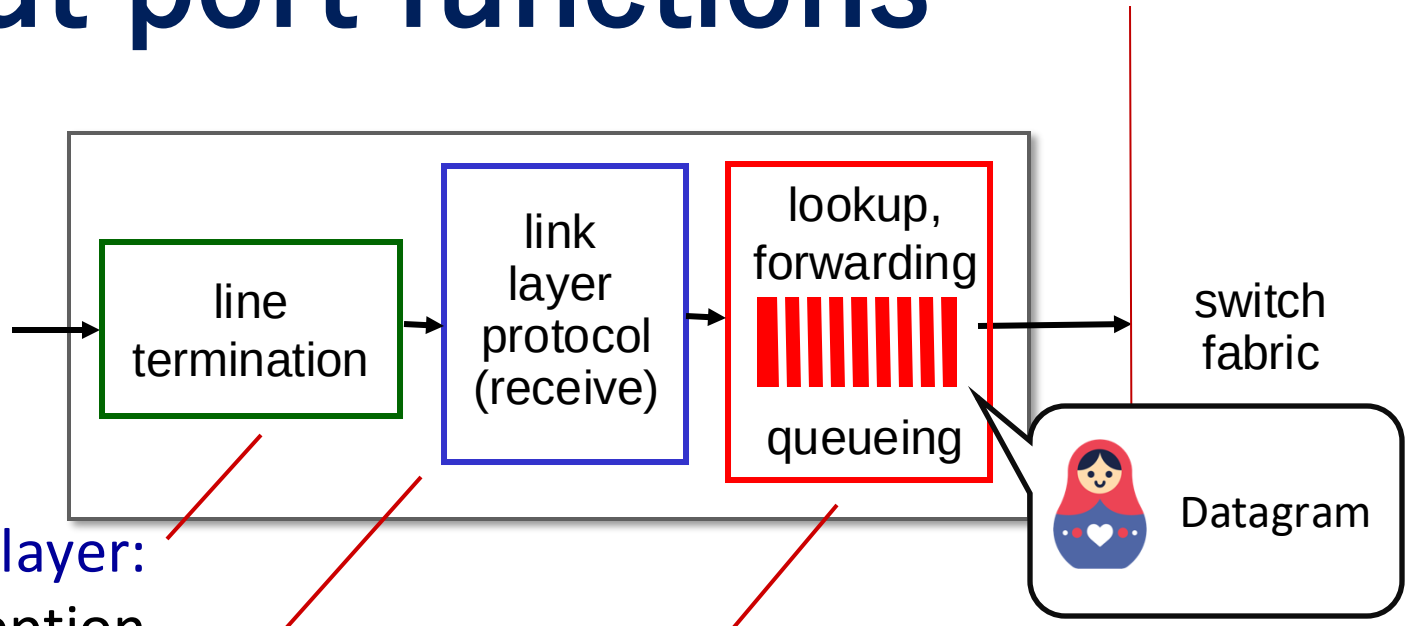
4.3 IP: Internet Protocol

Router architecture overview

- high-level view of generic router architecture:



Input port functions



physical layer:
bit-level reception

link layer:
e.g., Ethernet

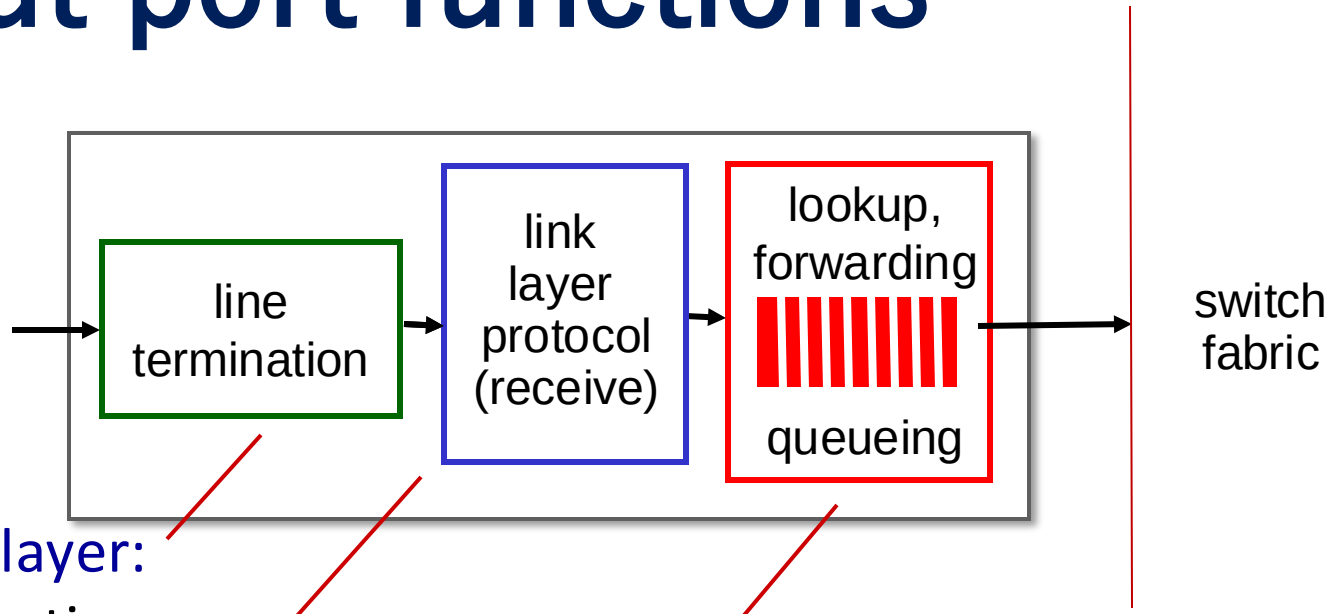
Frame

decentralized switching:

- using header field values, lookup output port using forwarding table in input port memory
- goal: complete input port processing extremely fast
- input port queuing: if datagrams arrive faster than forwarding rate into switch fabric

京都大学

Input port functions



physical layer:
bit-level reception

link layer:
e.g., Ethernet

Switching:

- using header field values, lookup output port by usinga forwarding table in the input port memory
- destination-based forwarding: forward based only on destination IP address (traditional)
- generalized forwarding: forward based on any set of header field values

京都大学

Destination-based forwarding

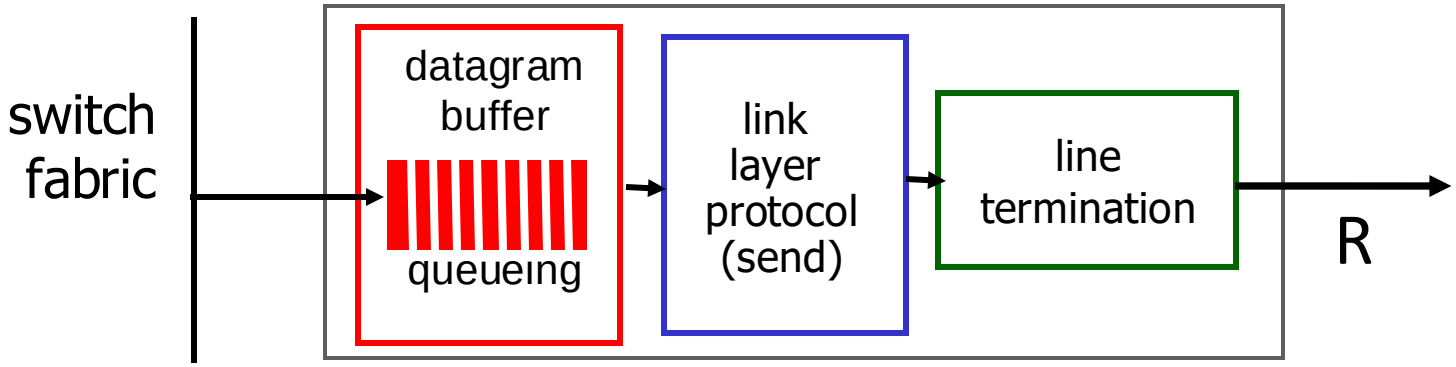
forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Queueing

- Packet queues can form at both input and output ports.
- How large the queues get depends on the traffic load, relative speed of the switching fabric, and the line speed.
- If the queue grows too large, the router’s memory will be exhausted, and packet loss occurs.

Queuing



- *Buffering* required when datagrams arrive from fabric faster than link transmission rate. *Drop policy:* which datagrams to drop if no free buffers?

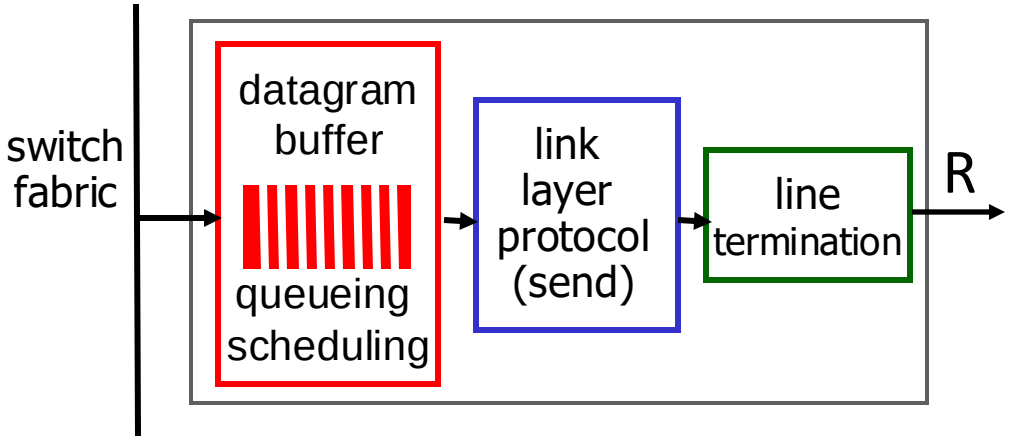
→ Datagrams can be lost due to congestion, lack of buffers

- *Scheduling discipline* chooses among queued datagrams for transmission

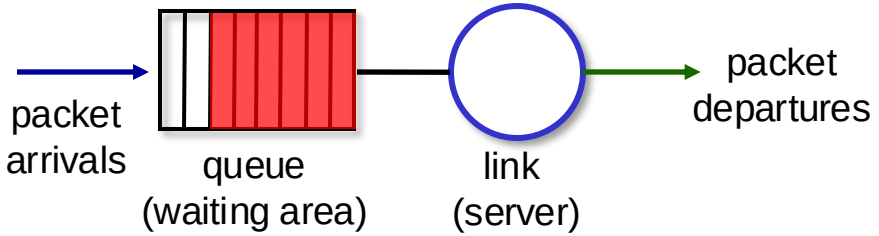
→ Priority scheduling – who gets best performance, how to ensure network neutrality

京都大学

Buffer Management



Abstraction: queue



buffer management:

- drop: which packet to add, drop when buffers are full
  - tail drop: drop arriving packet
  - priority: drop/remove on priority basis

packet scheduling:

deciding which packet to send next on link

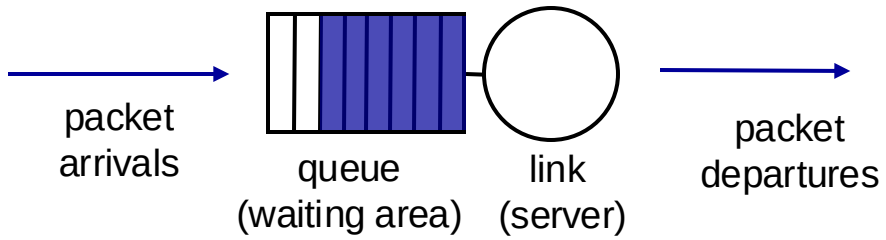
京都大学

Scheduling

- If several packets are queued at the output port, a packet scheduler at the output port must determine which packet to transmit first.
- If a new packet arrives to a full queue, a decision has to be made of which packet to discard.
- Simplest policy: First-in, First-out (FIFO), but not always appropriate.

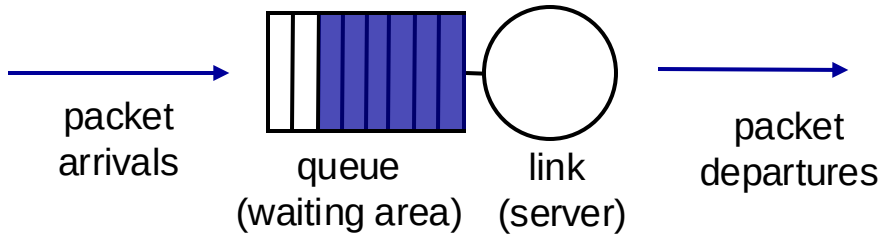
FIFO scheduling

- Packets arriving at the link output queue wait for transmission if the link is currently busy transmitting another packet.
- When a packet is completely transmitted, it is removed from the queue.
- Packets are transmitted in the same order as they arrive.



Scheduling mechanisms

- *scheduling:* choose next packet to send on link
- *FIFO (first in first out) scheduling:* send in order of arrival to queue
  - *discard policy:* if packet arrives to full queue: who to discard?
    - *tail drop:* drop arriving packet
    - *priority:* drop/remove on priority basis
    - *random:* drop/remove randomly



Priority Queuing

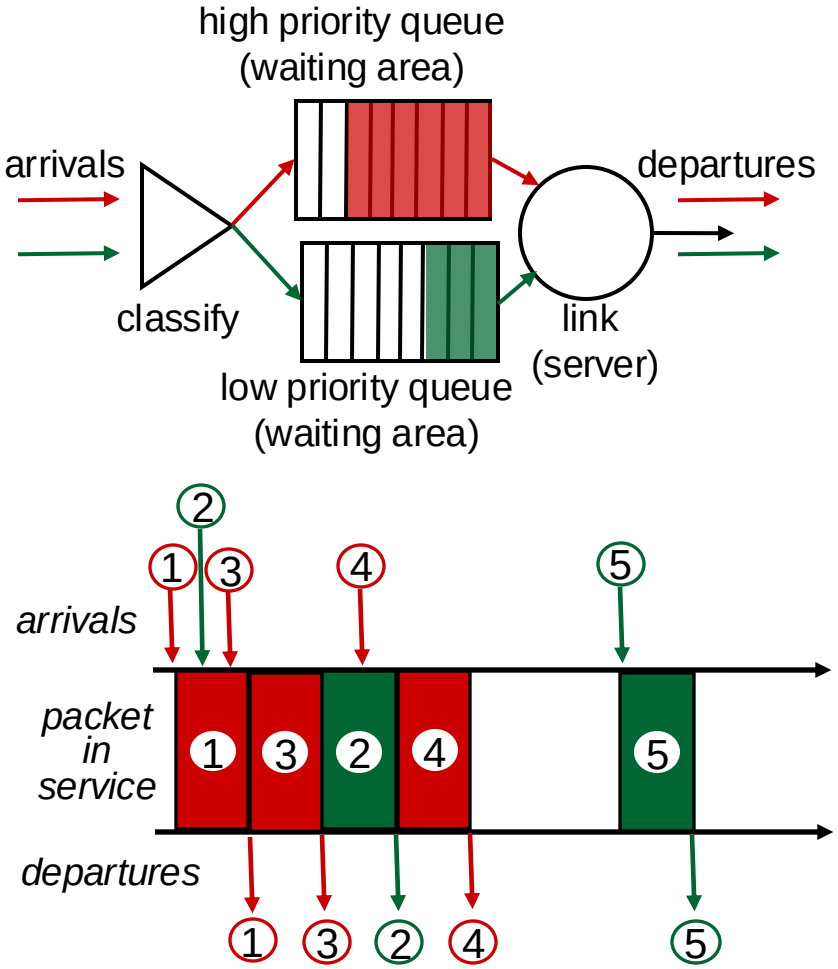
- Packets arriving at the output link are classified into priority classes.
- This can for example be used to prioritize real-time data like voice-over-IP packets over data such as SMTP e-mail packets.
- Each priority class gets their own queue.
- The next packet to transmit will be chosen from the highest priority class that has a nonempty queue.
- Inside the same priority class usually FIFO is used.

Priority queuing

priority scheduling:

send highest priority queued packet

- multiple *classes*, with different priorities
  - class may depend on marking or other header info, e.g. IP source/dest, port numbers, etc.



京都大学

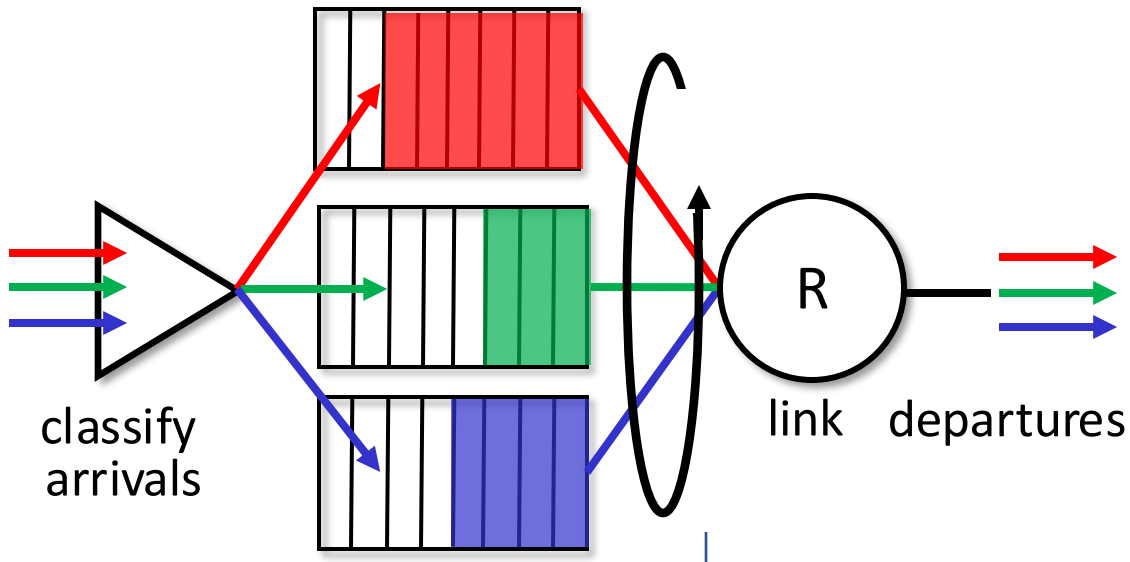
Round Robin

- Packets are sorted into classes as with priority queueing.
- However, instead of having a strict priority, the scheduler alternates between the classes.
- If there is no packet waiting in the queue for a class, immediately go to the next class.

Scheduling policies: round robin

Round Robin (RR) scheduling:

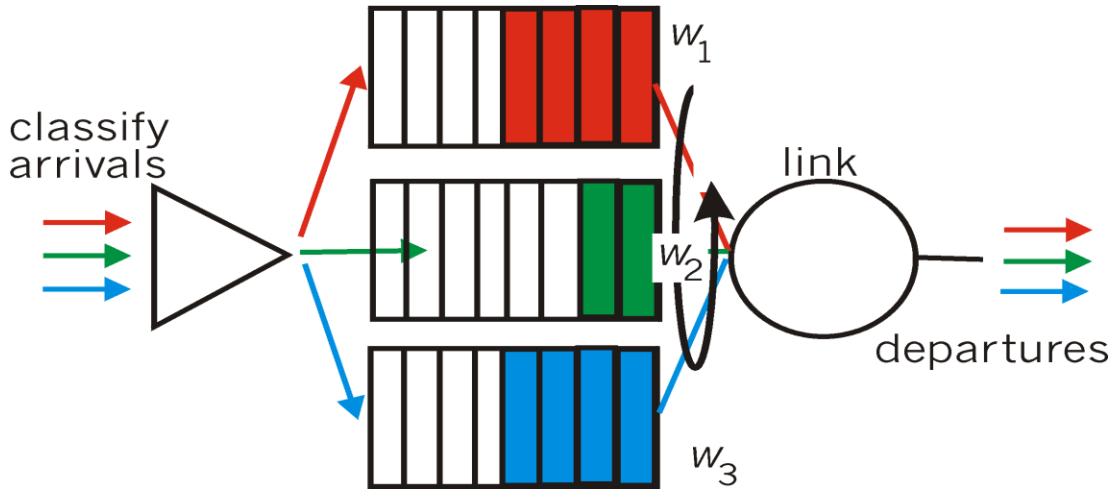
- arriving traffic classified, queued by class
  - any header fields can be used for classification
- server cyclically, repeatedly scans class queues, sending one complete packet from each class (if available) in turn



Scheduling policies: still more

Weighted Fair Queuing (WFQ):

- generalized Round Robin
- each class gets weighted amount of service in each cycle



Outline

4.1 Overview of Network layer

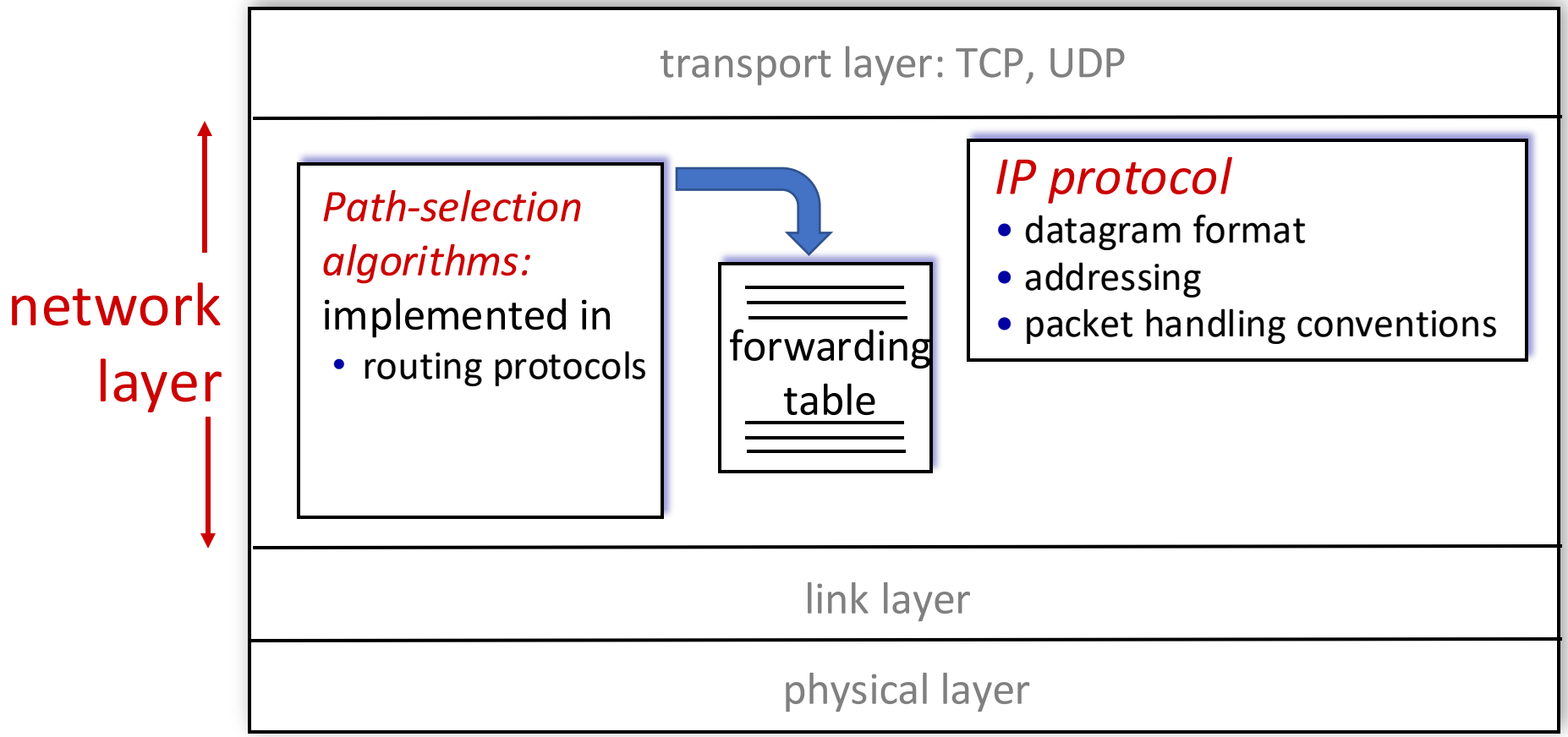
- data plane
- control plane

4.2 What's inside a router

4.3 IP: Internet Protocol

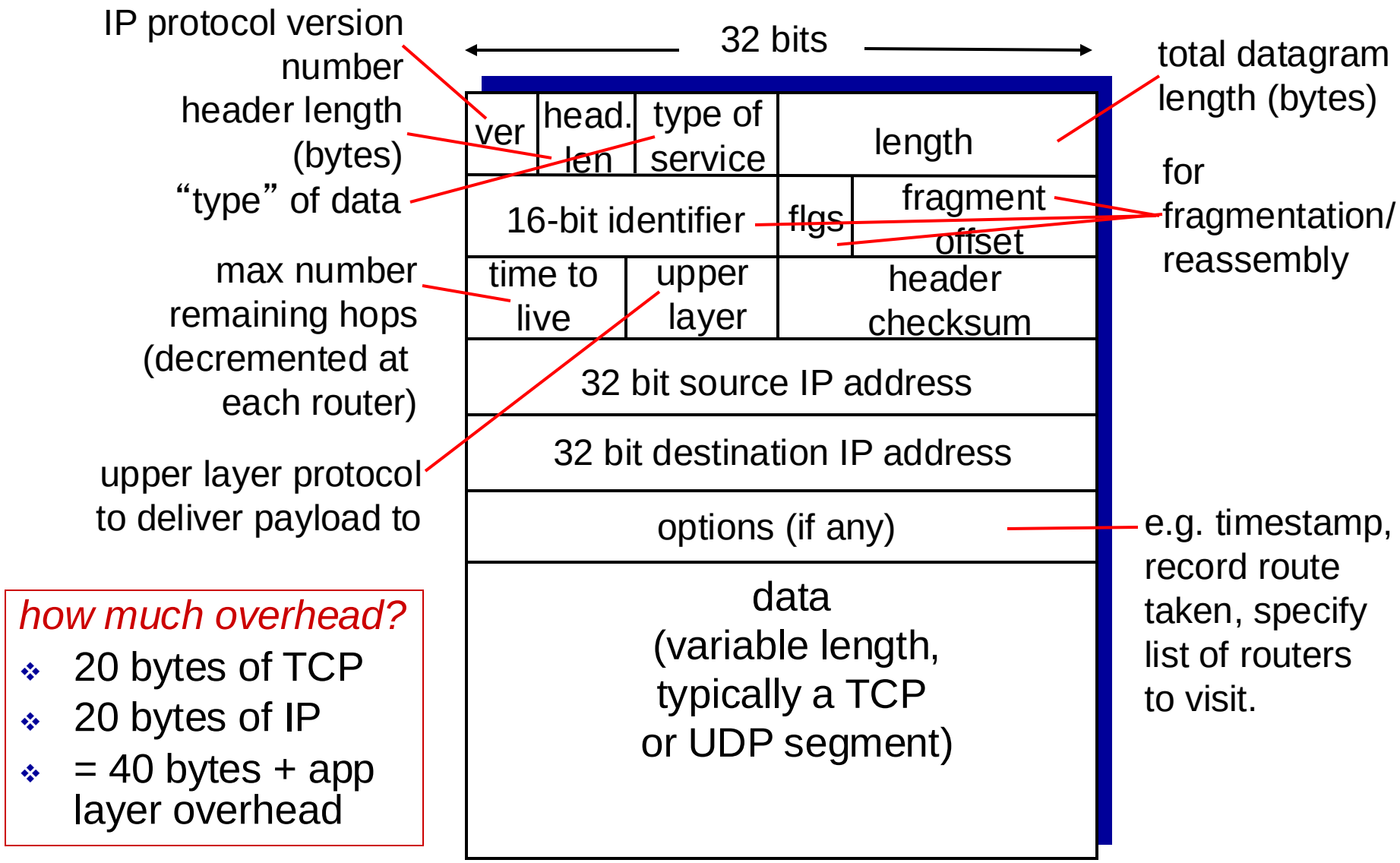
Network Layer: Internet

host, router network layer functions:



京都大学

IP datagram format

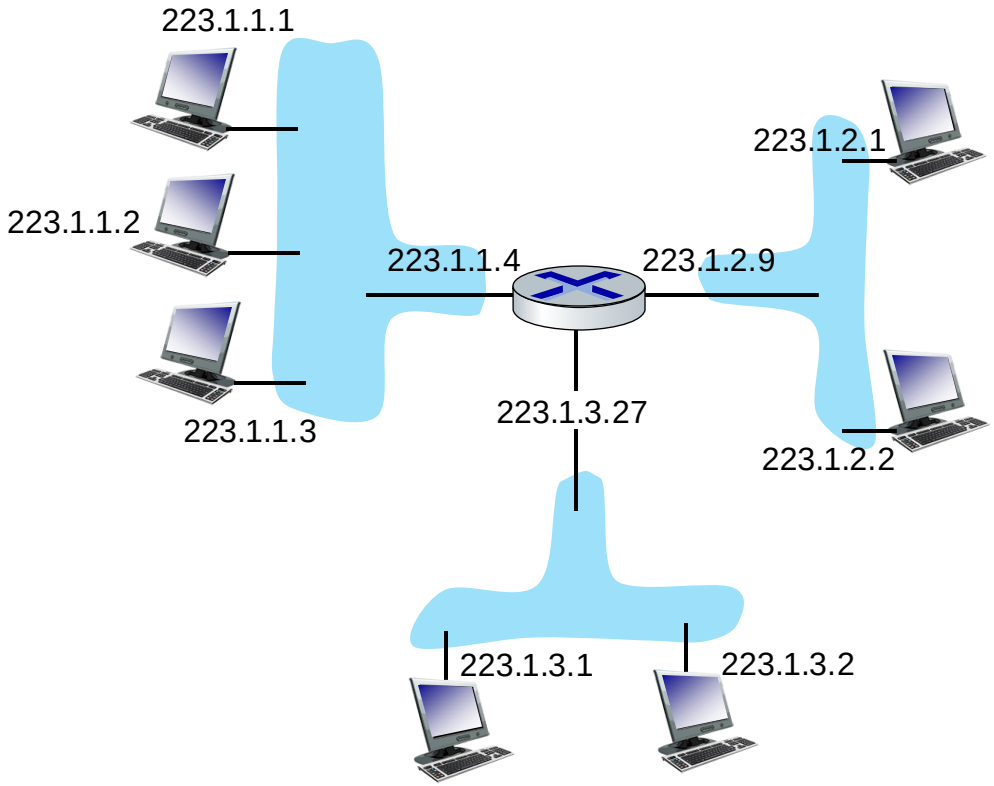


how much overhead?

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

IP addressing: introduction

- IP address: 32-bit identifier associated with each host or router *interface*
- interface: connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one or two interfaces (e.g., wired Ethernet, wireless 802.11)



dotted-decimal IP address notation:

223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

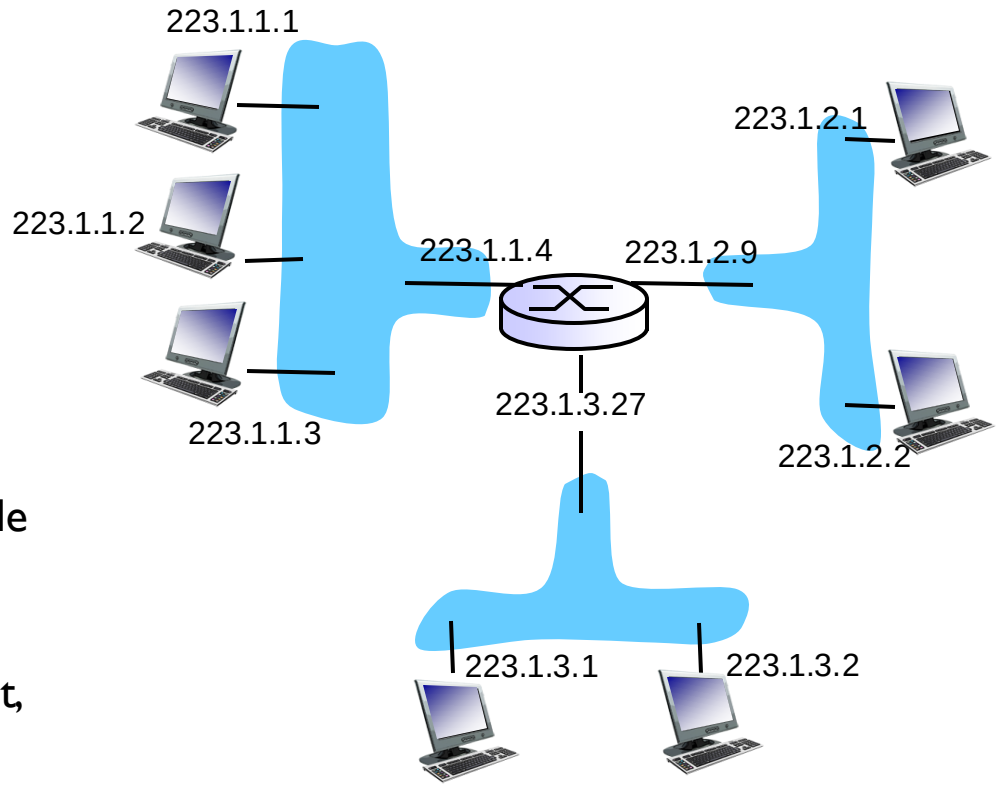
京都大学

IP addressing

- A host typically has only one or two links into the network (Ethernet, wireless, …).
- When IP wants to send a datagram, it does so over one of these links.
- The boundary between host and the physical link is called an interface.
- A router has to receive a datagram and forward it to some other link
- A router usually has several links.
- The boundary between the router and any of its links is also called interface.
- Each interface is required to have its unique IP address.

IP addressing: introduction

- *IP address:* 32-bit identifier for host, router *interface*
- *interface:* connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one or two interfaces (e.g., wired Ethernet, wireless 802.11)
- *IP addresses associated with each interface*



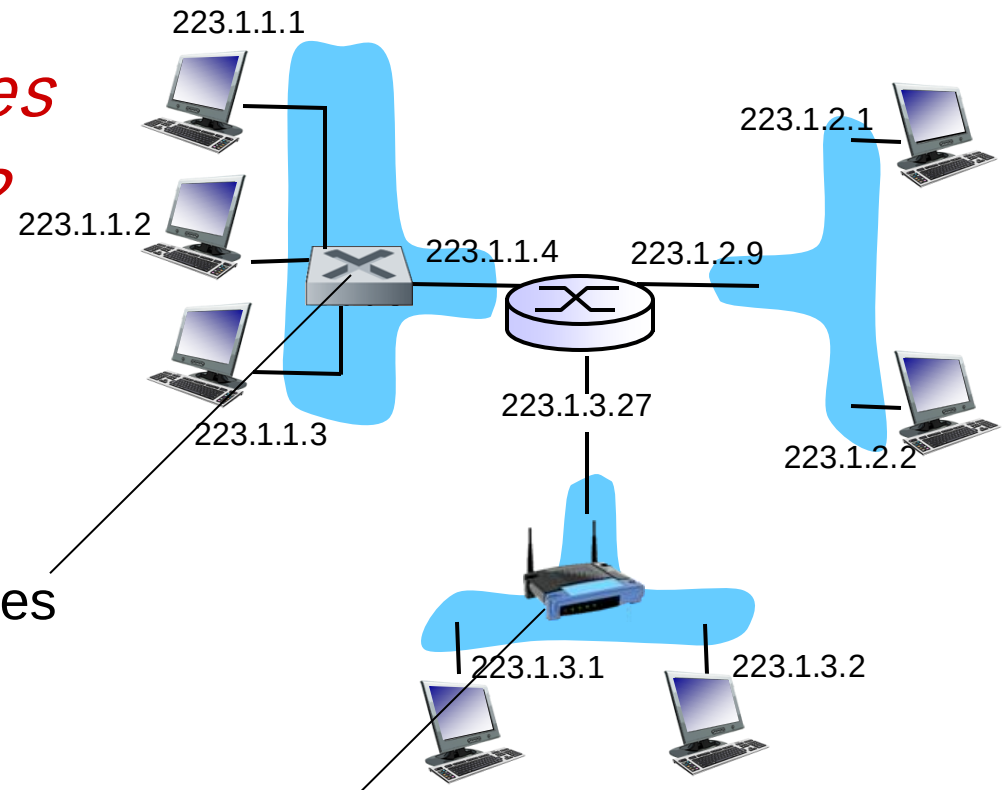
223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

京都大学

IP addressing: introduction

Q: how are interfaces actually connected?



A: wired Ethernet interfaces connected by Ethernet switches

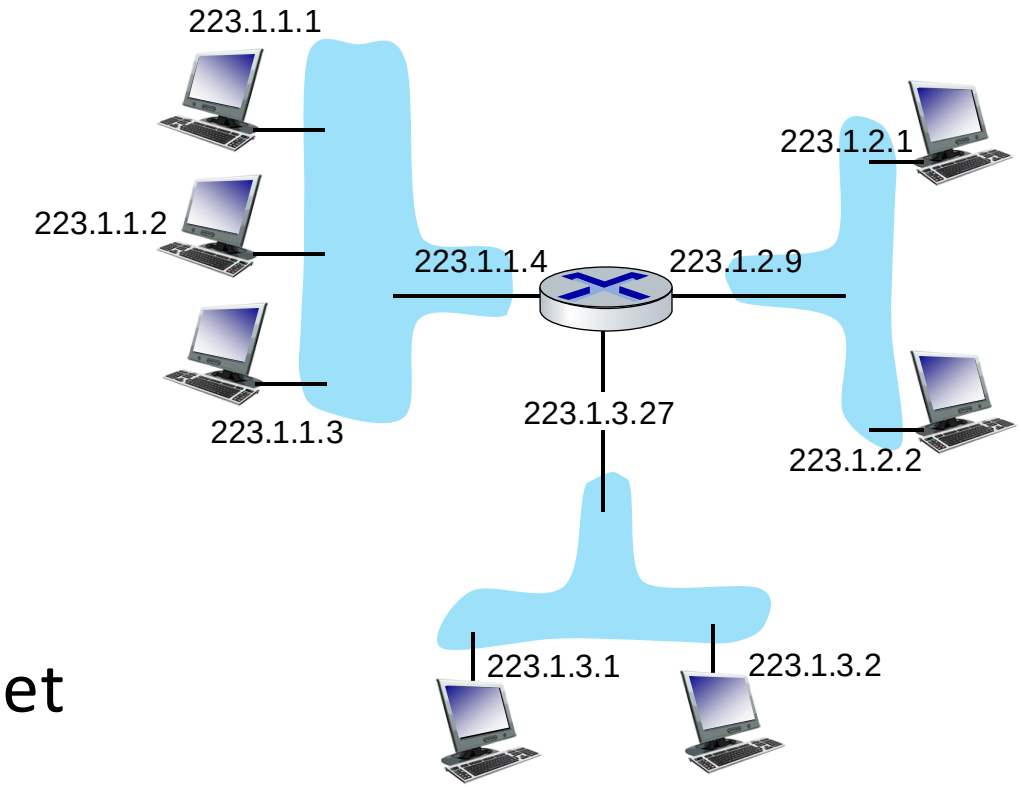
For now: don't need to worry about how one interface is connected to another (with no intervening router)

A: wireless WiFi interfaces connected by WiFi base station

京都大学

Subnets

- *What's a subnet ?*
  - device interfaces that can physically reach each other without passing through an intervening router
- IP addresses have structure:
  - subnet part: devices in same subnet have common high order bits
  - host part: remaining low order bits

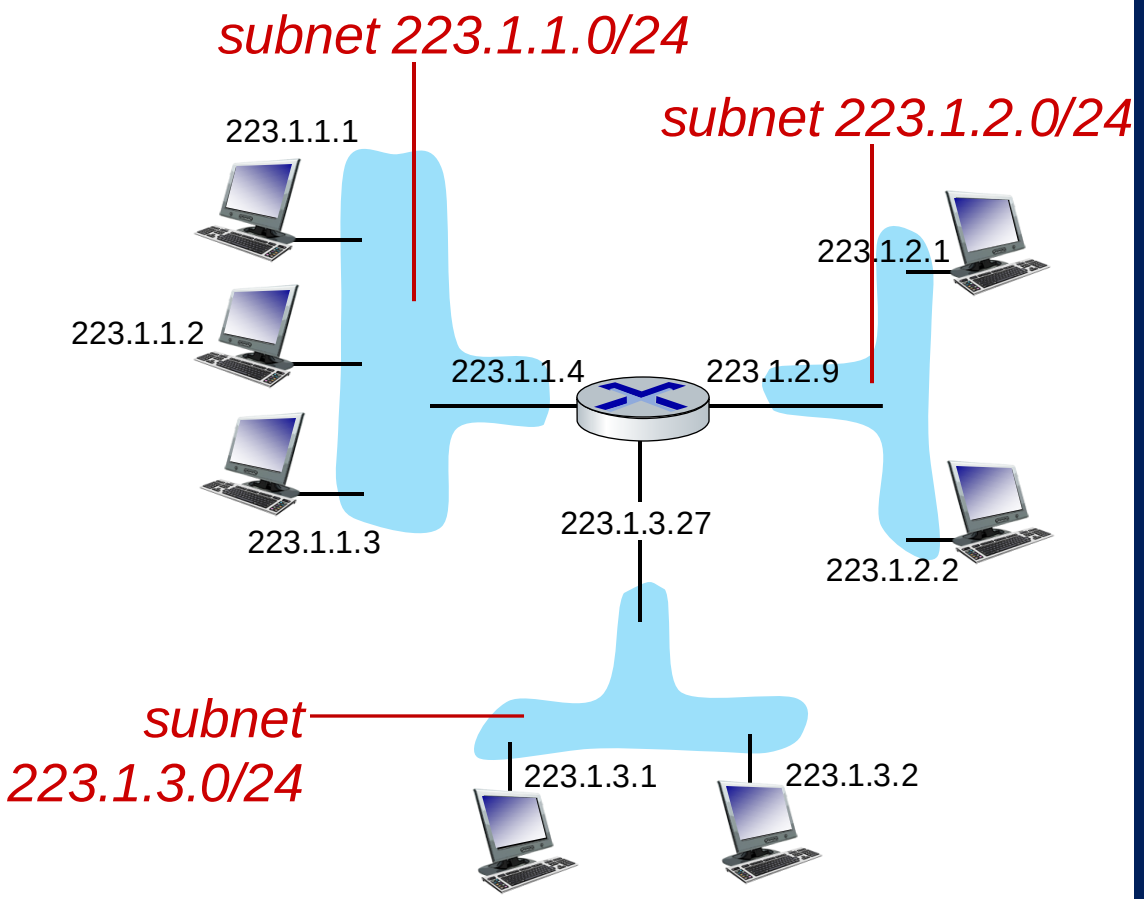


network consisting of 3 subnets

Subnets

Recipe for defining subnets:

- detach each interface from its host or router, creating "islands" of isolated networks
- each isolated network is called a *subnet*

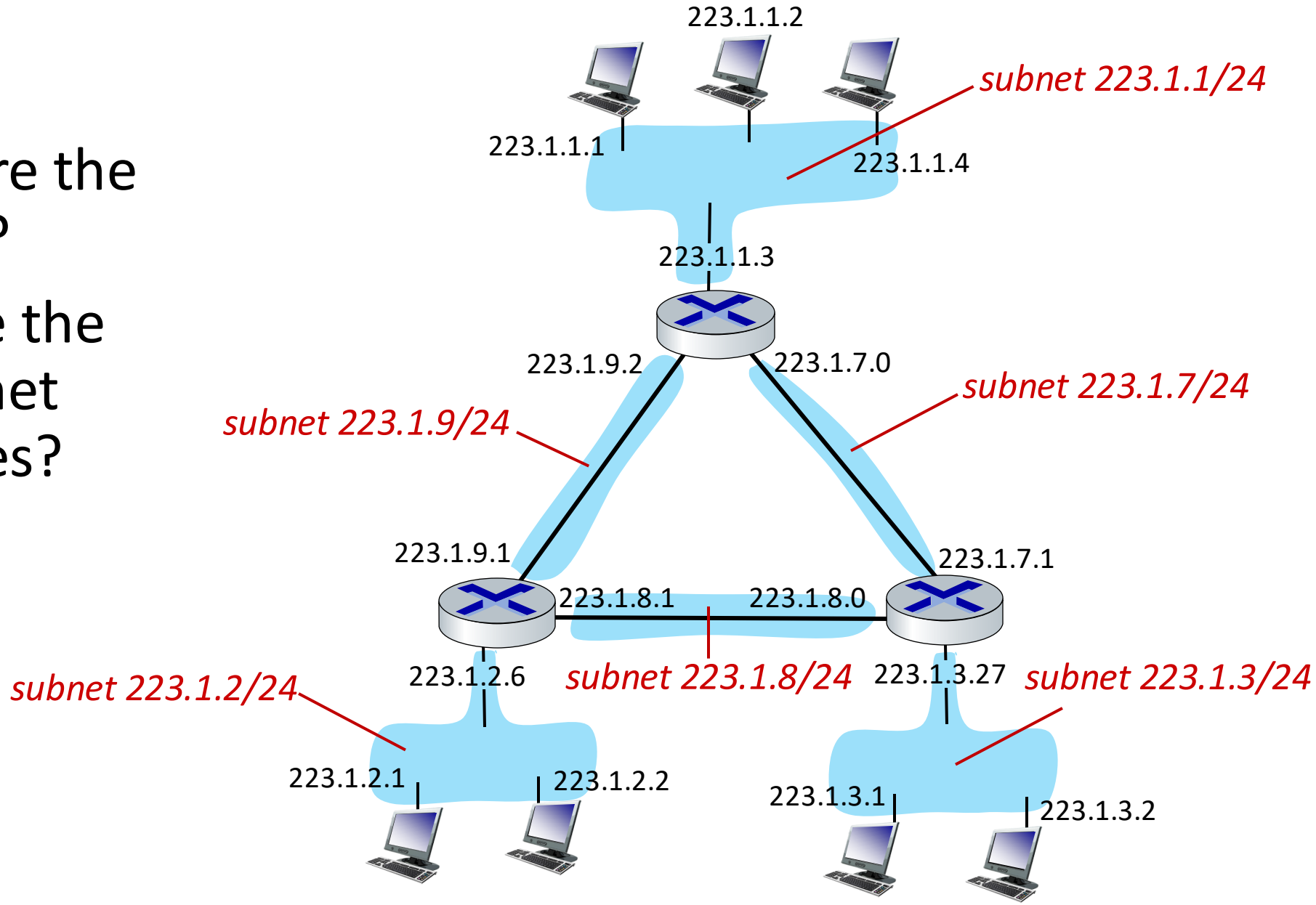


subnet mask: /24
(high-order 24 bits: subnet part of IP address)

京都大学

Subnets

- where are the subnets?
- what are the /24 subnet addresses?

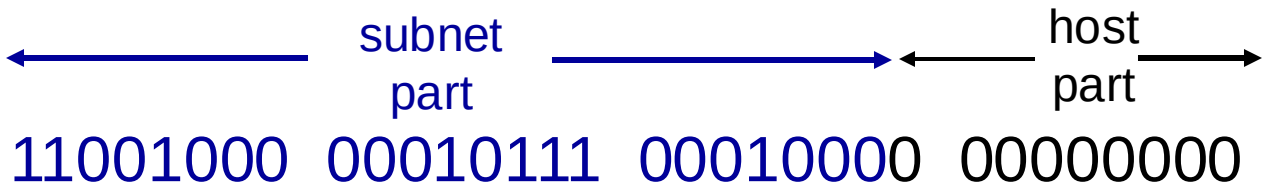


京都大学

IP addressing: CIDR

CIDR: Classless InterDomain Routing (pronounced “cider”)

- subnet portion of address of arbitrary length
- address format: a.b.c.d/x, where x is # bits in subnet portion of address



200.23.16.0/23

京都大学

How to get an IP address

- To obtain a block of IP addresses for use within an organization, the network administrator might first contact its ISP.
- The ISP itself has been allocated a block of IP addresses and can further divide its address block.
- The ISP gets its IP addresses from the Internet Corporation for Assigned Names and Numbers (ICANN) which is responsible for allocating IP addresses and managing DNS root servers.

How to get an IP address

- Once an organization has obtained a block of IP addresses, it can assign individual IP addresses to hosts and router interfaces.
- The address can be manually set up by the network administrator.
- Routers will typically have a fixed IP address that the system administrator configures manually.
- Host addresses are usually not set up manually, but dynamically using the Dynamic Host Configuration Protocol (DHCP).
- DHCP dynamically gets an IP address from the server.

DHCP: Dynamic Host Configuration Protocol

goal: allow host to *dynamically* obtain its IP address from network server when it joins network

- Easy to join and leave a network frequently
- allows reuse of addresses (only hold address while connected/"on")
- support for mobile users who want to join network (more shortly)

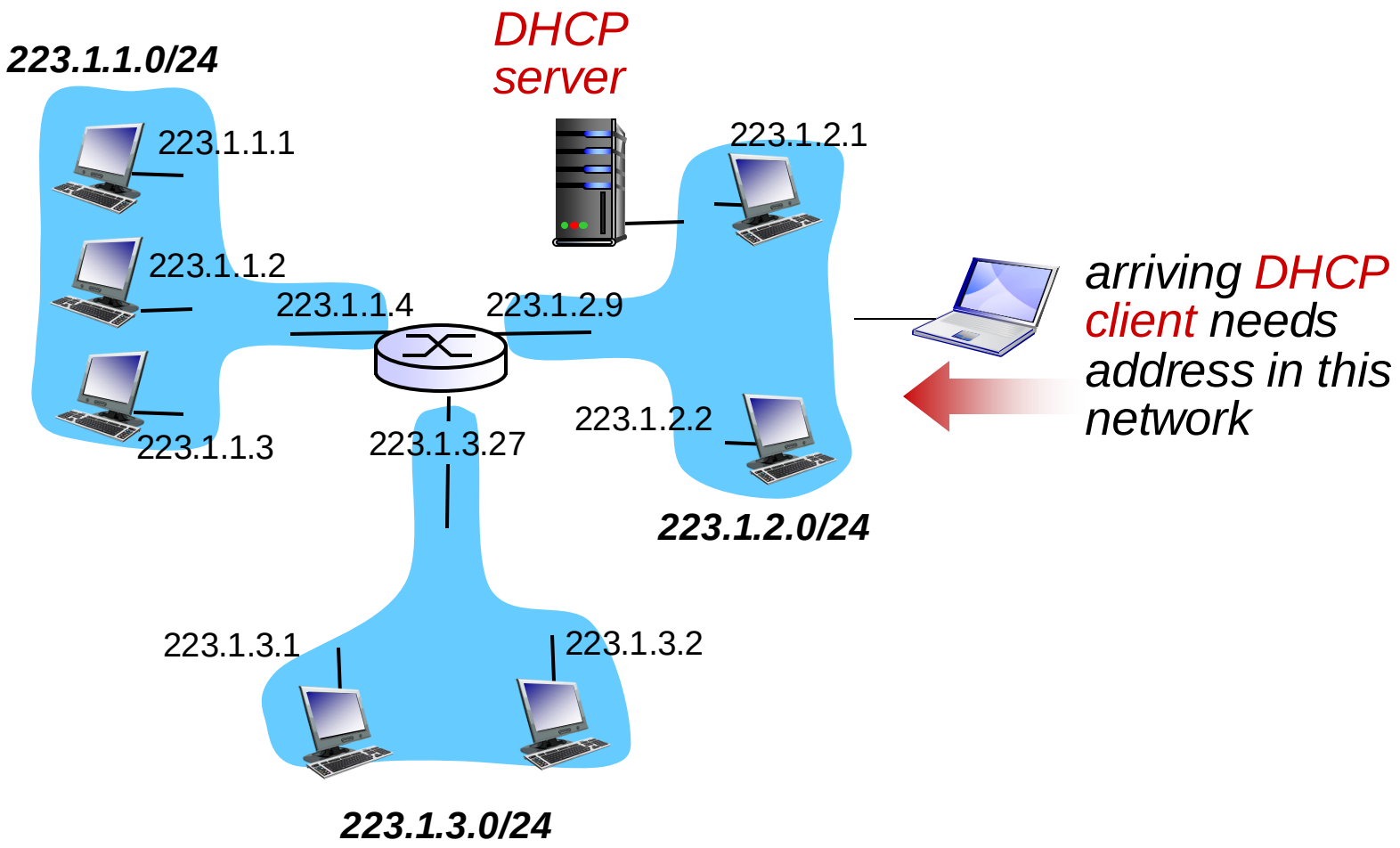
DHCP overview:

- host broadcasts "DHCP discover" msg [optional]
- DHCP server responds with "DHCP offer" msg [optional]
- host requests IP address: "DHCP request" msg
- DHCP server sends address: "DHCP ack" msg

DHCP messages

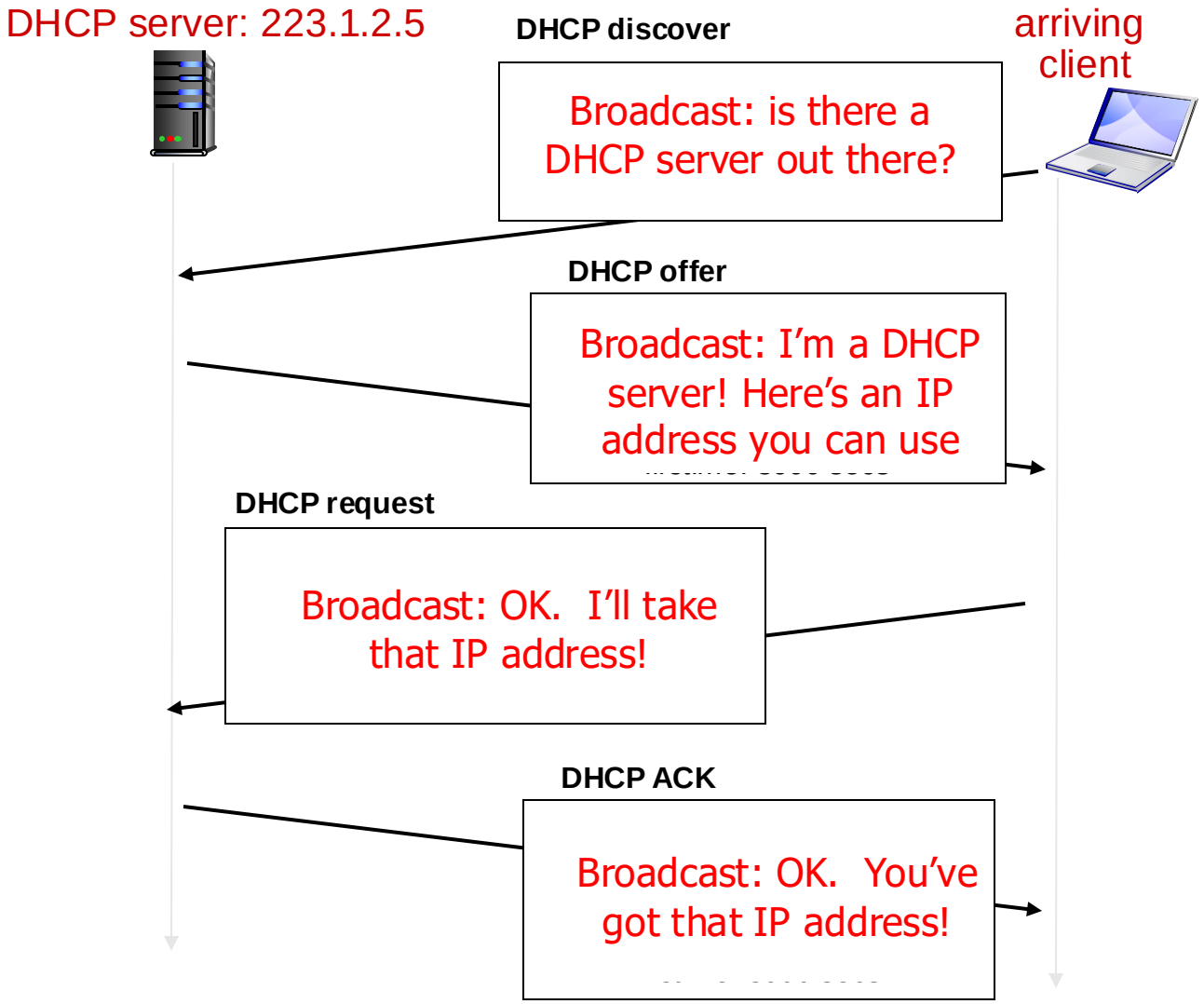
- DHCP server discovery: The client sends an UDP message with destination port 67. This message is broadcasted to all hosts on the subnet by using the broadcasting destination IP address of 255.255.255.255. The sender IP address is set to 0.0.0.0 and the sender port number to 68. The message also contains a transaction ID number to identify the request.
- DHCP server offer: Each DHCP server receiving a discovery message responds with a DHCP offer message that is again broadcasted to all hosts on the subnet. The offer message contains the transaction ID, a proposed IP address for the client, the IP address of the DHCP server and a time how long the IP address will be valid (lifetime).
- DCHP request: The client chooses an IP address among the offers and responds with a DHCP request message. The message contains a copy of the configuration parameters.
- DHCP acknowledgment: The server responds with a DHCP ack message, acknowledging the requested parameters.

DHCP client-server scenario



京都大学

DHCP client-server scenario



京都大学

IP addresses: how to get one?

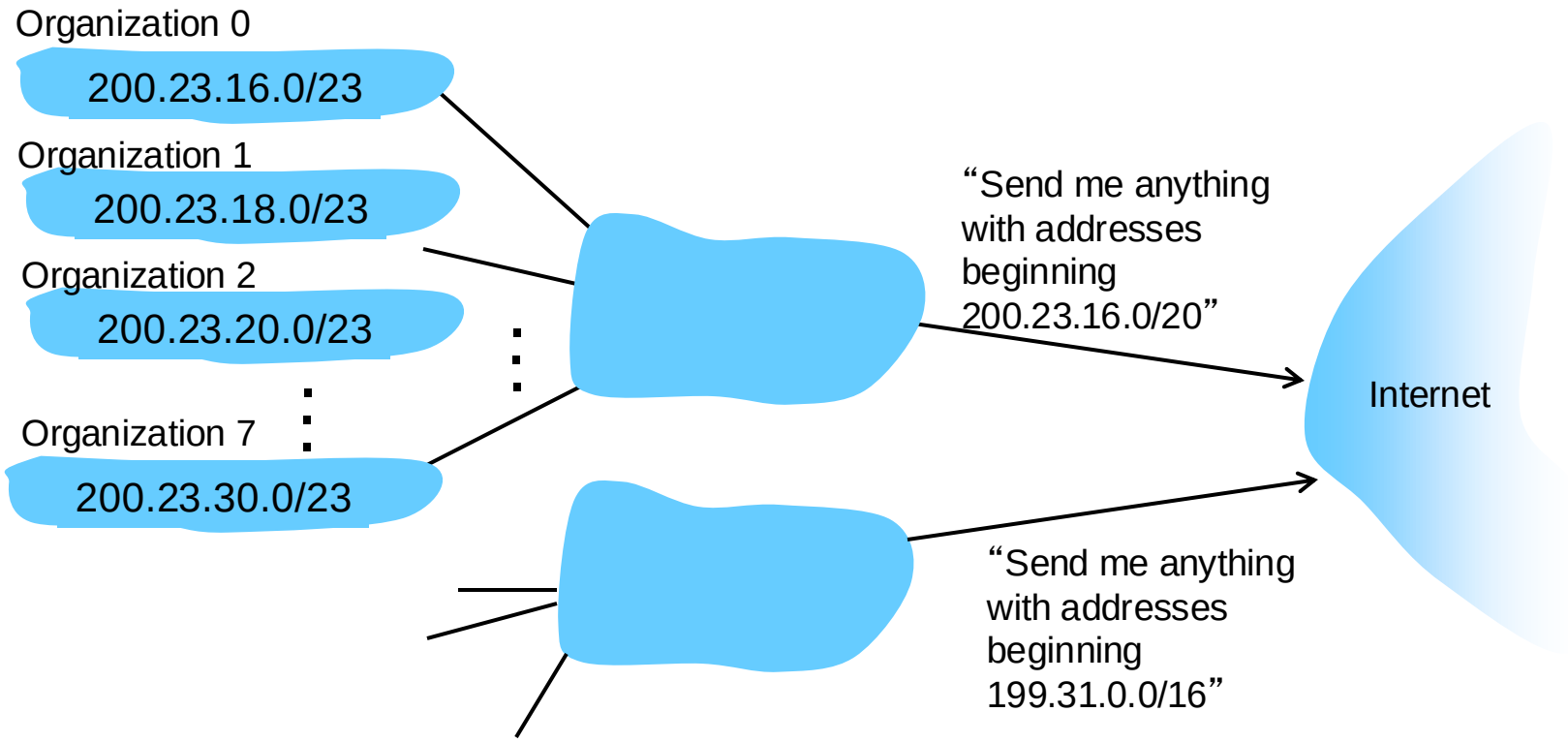
Q: how does *network* get subnet part of IP addr?

A: gets allocated portion of its provider ISP's address space

ISP's block	11001000 00010111 00010000	00000000	200.23.16.0/20
Organization 0	11001000 00010111 00010000	00000000	200.23.16.0/23
Organization 1	11001000 00010111 00010010	00000000	200.23.18.0/23
Organization 2	11001000 00010111 00010100	00000000	200.23.20.0/23
...	.....	....	....
Organization 7	11001000 00010111 00011110	00000000	200.23.30.0/23

Hierarchical addressing: route aggregation

hierarchical addressing allows efficient advertisement of routing information:



京都大学

IP addressing: last words …

Q: how does an ISP get block of addresses?

A: ICANN: Internet Corporation for Assigned Names and Numbers http://www.icann.org/

- allocates IP addresses, through 5 regional registries (RRs) (who may then allocate to local registries)
- manages DNS root zone, including delegation of individual TLD (.com, .edu , …) management

Q: are there enough 32-bit IP addresses?

- ICANN allocated last chunk of IPv4 addresses to RRs in 2011
- NAT (next) helps IPv4 address space exhaustion
- IPv6 has 128-bit address space

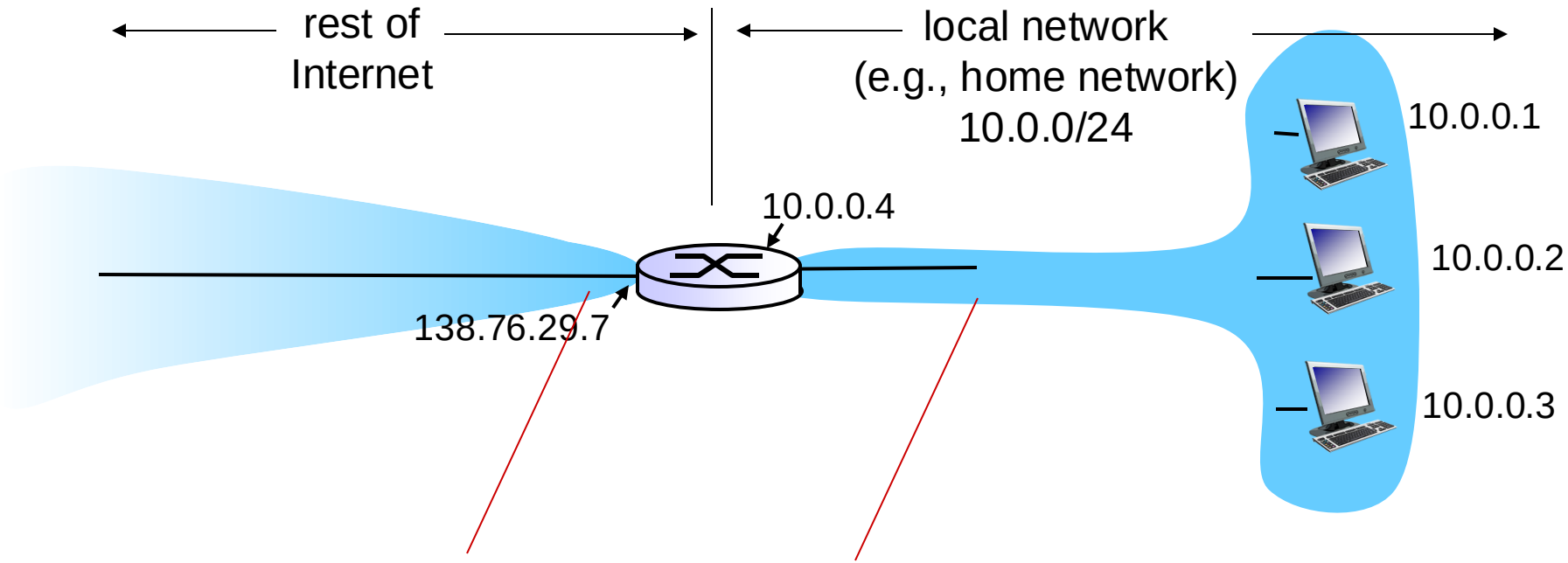
Network Address Translation (NAT)

- Every device needs an IP address.
- Whenever multiple machines need to be connected in a local area network, a range of IP addresses would need to be allocated by the ISP.
- In a typical home or small office network this is not feasible.
- Network Address Translation (NAT) is a simpler approach to address allocation of IP addresses in a small local area network.

Network Address Translation (NAT)

- NAT is implemented in a NAT-enabled router.
- The NAT-enabled router resides in the home network.
- Inside the home network, local IP addresses starting with 10.0.0 are used for local communication within the private network.
- Local IP addresses only have meaning inside the local network and cannot be used outside.
- The NAT-enabled router looks like a single device for the outside world.
- Thus, there is only one global IP address for the whole local network.
- The NAT-enabled router must translate between local and global addresses.
- For this, the NAT-enabled router uses its own port numbers and a NAT translation table mapping its port numbers to local IP addresses and original port numbers.

NAT: network address translation



all datagrams *leaving* local network have *same* single source NAT IP address: 138.76.29.7,different source port numbers

datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

京都大学

NAT: network address translation

motivation: local network uses just one IP address as far as outside world is concerned:

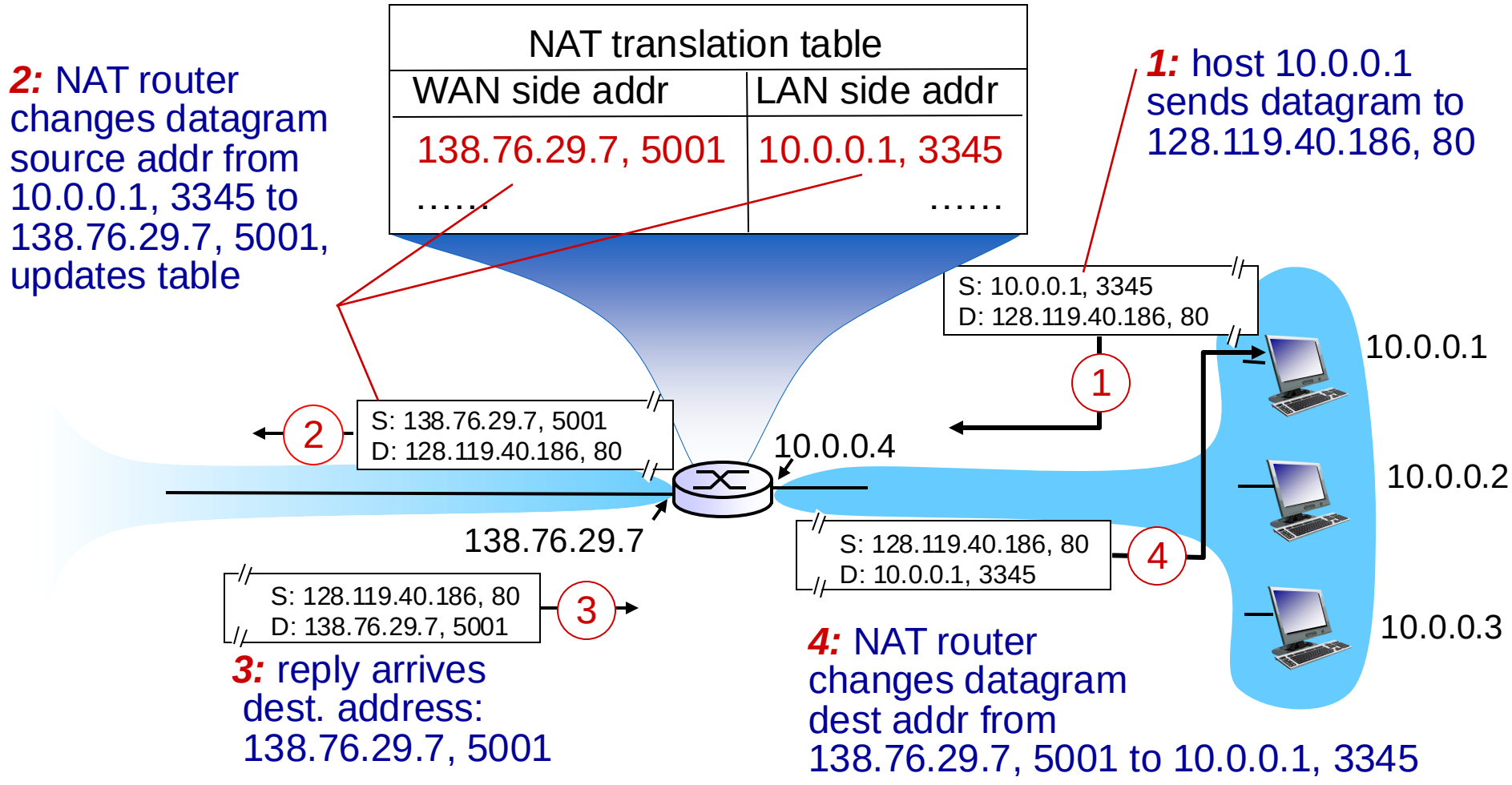
- range of addresses not needed from ISP: just one IP address for all devices
- can change addresses of devices in local network without notifying outside world
- can change ISP without changing addresses of devices in local network
- devices inside local net not explicitly addressable, visible by outside world (a security plus)

NAT: network address translation

implementation: NAT router must:

- *outgoing datagrams: replace* (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
  . . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr
- *remember (in NAT translation table)* every (source IP address, port #) to (NAT IP address, new port #) translation pair
- *incoming datagrams: replace* (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

NAT: network address translation



IPv6: motivation

- *initial motivation:* 32-bit address space would be completely allocated (only around 4.3 billion addresses available).
- IPv6 uses 128 bit addresses
- additional motivation:
  - header format helps speed processing/forwarding

IPv6 datagram format:

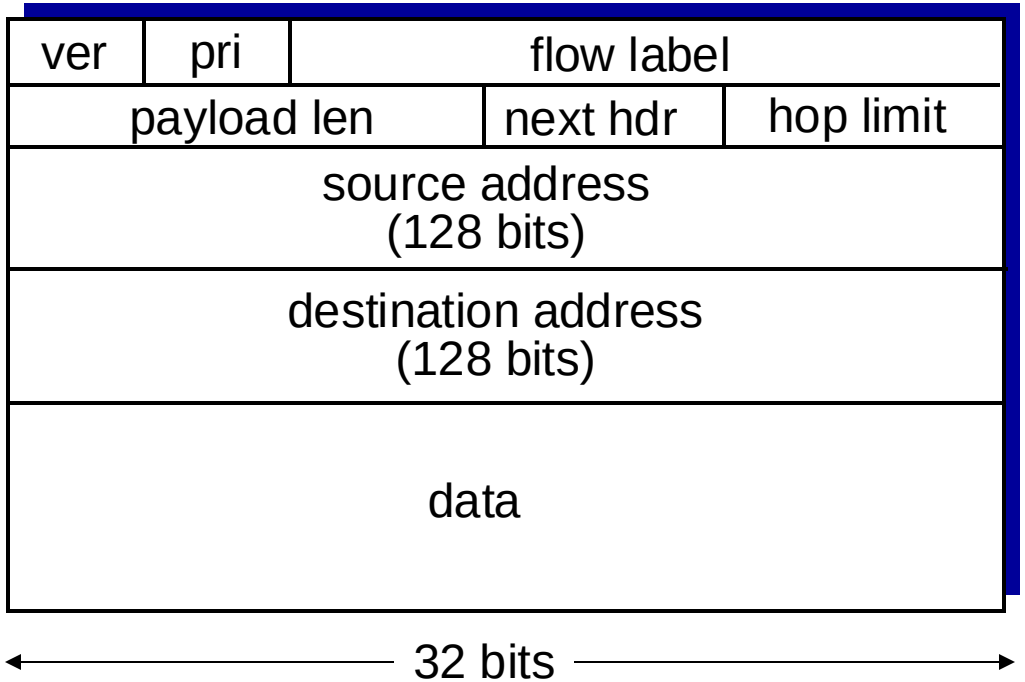
  - fixed-length 40 byte header

IPv6 datagram format

priority: identify priority among datagrams in flow

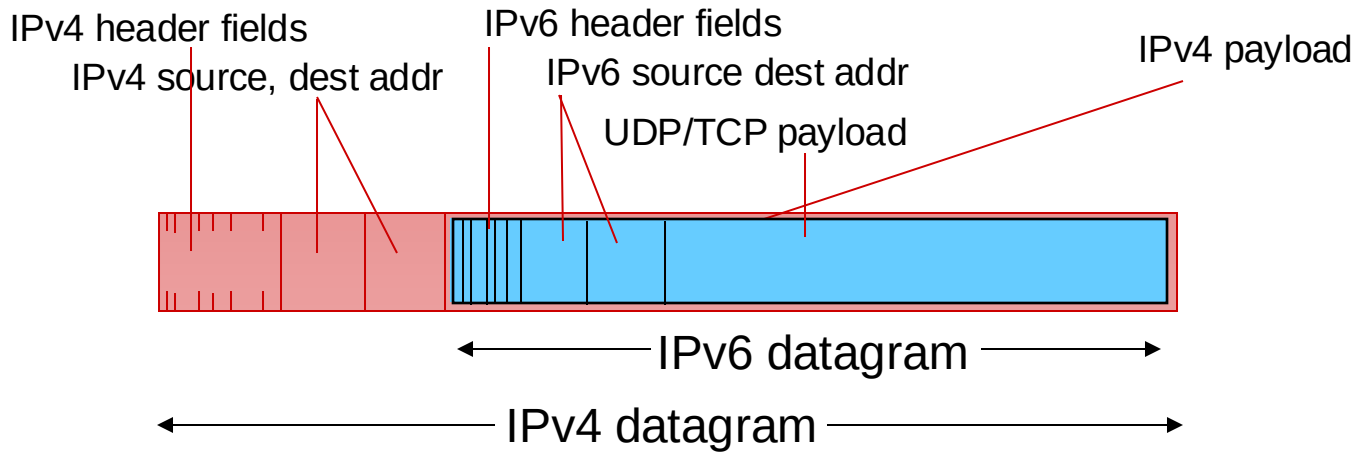
flow Label: identify datagrams in same "flow." (concept of"flow" not well defined).

next header: identify upper layer protocol for data

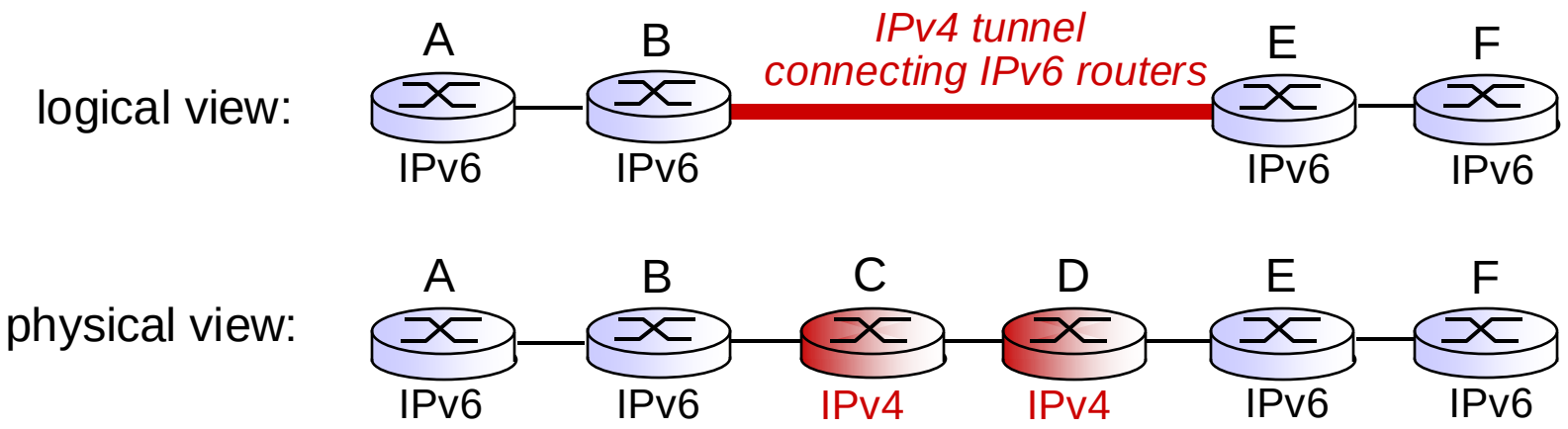


Transition from IPv4 to IPv6

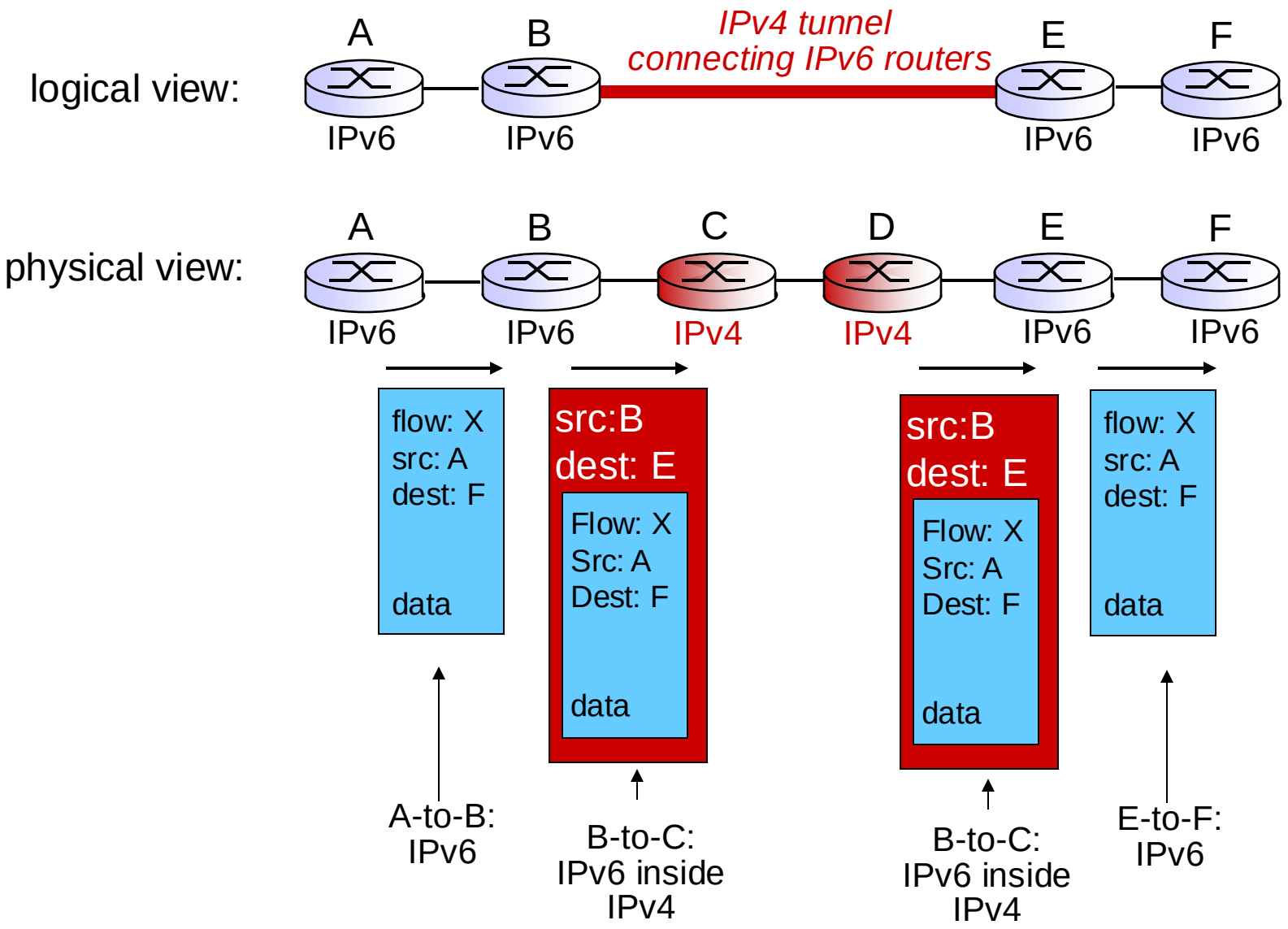
- not all routers can be upgraded simultaneously
  - how will network operate with mixed IPv4 and IPv6 routers?
- *tunneling:* IPv6 datagram carried as *payload* in IPv4 datagram among IPv4 routers



Tunneling



Tunneling



京都大学

IPv6: adoption

- *Long (long!) time for deployment, use*
  - 20 years and counting!
  - think of application-level changes in last 20 years: WWW, Facebook, streaming media, Skype, …