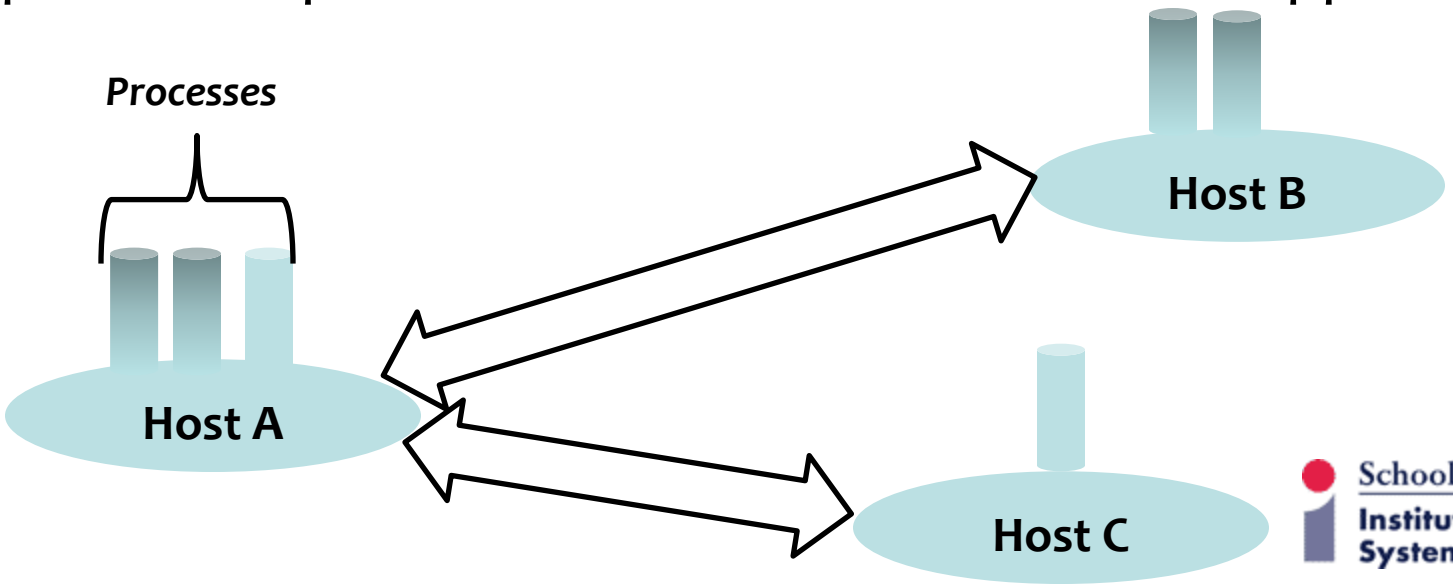
The Network Layer: Part I

These slides are adapted from those provided by Jim Kurose and Keith Ross with their book "Computer Networking: A Top-Down Approach (6th edition)."

School of Informatics
Institute for Computing Systems Architecture

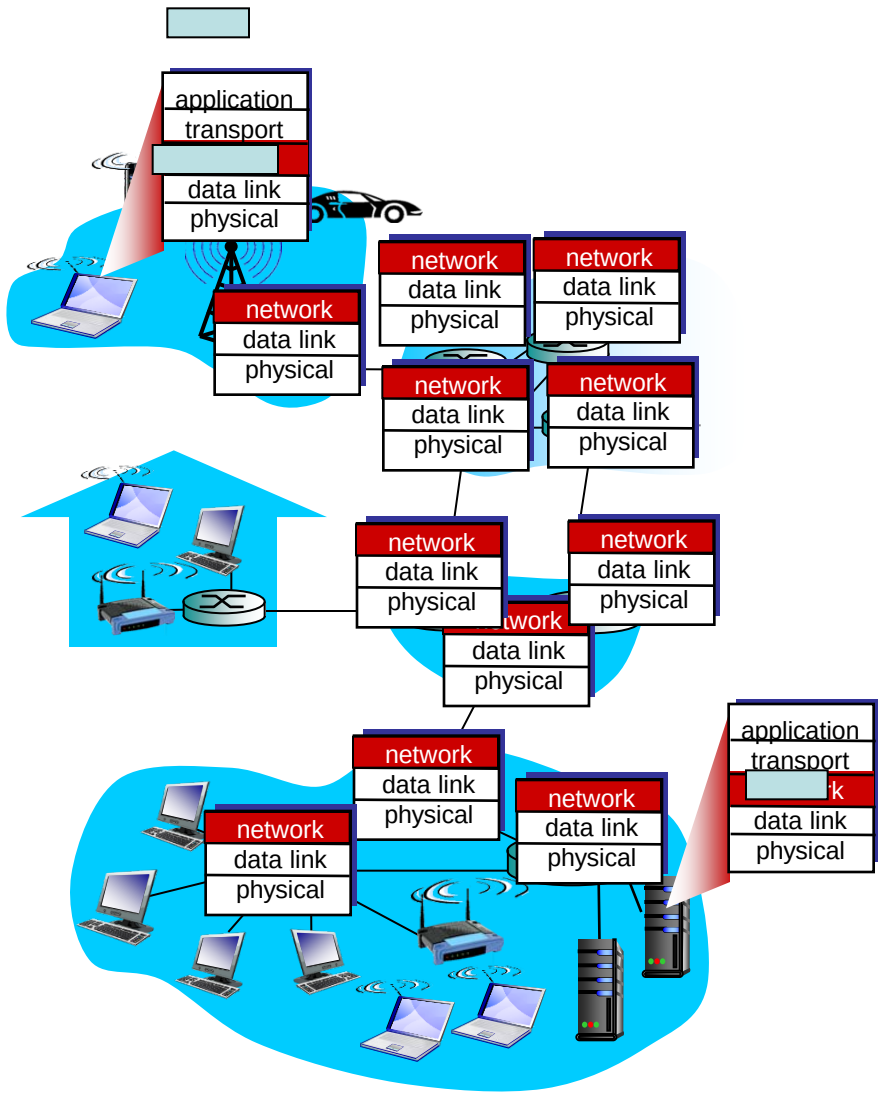
Overview

- Network layer provides the host-to-host communication service
  - Can be connectionless (datagram networks) or connection-oriented (virtual-circuit networks) but not both
  - Contrast with Internet transport layer which features both connection-oriented (TCP) and connectionless (UDP) protocols
- Transport layer protocols rely on this service to provide process-to-process communication service for applications



Overview (contd.)

- Made use of in all nodes of a communication path, including end-hosts ***and*** routers, unlike application and transport layers whose use for data packets is confined to end-hosts



Two Broad Classes of Computer Networks

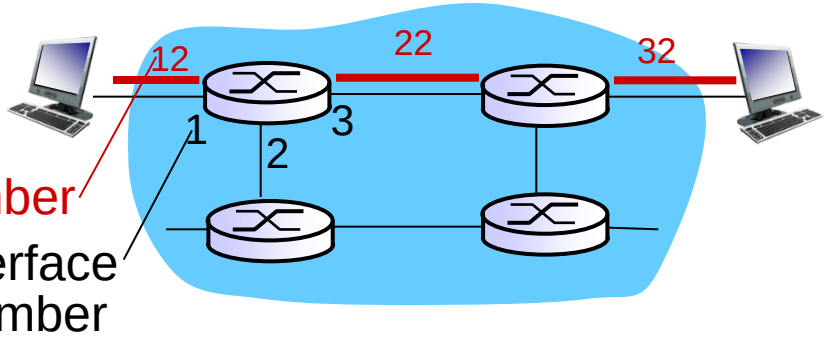
- ***Virtual-circuit networks*** (e.g., ATM, MPLS) provide a connection-oriented network layer service

- ***Datagram networks*** (e.g., the Internet) provide a connectionless service at the network layer (e.g., the Internet) ← Our focus

Virtual-Circuit Networks

- Provide ***connection-oriented*** host-to-host communication service
- Examples: ATM, MPLS, frame relay
- Network-layer connections are called ***virtual circuits (VCs)***
- VC consists of a path, VC numbers, forwarding table at each intermediate router
  - Beneficial to allow VC numbers to be different on different links of a VC path: easier to realise unique VC numbers locally and also reduced header field for VC number
- Packets are stamped with VC number that may need to be replaced with a new VC number at each router on the path
- Routers must maintain connection state information as VCs are per-connection

VC Forwarding Table



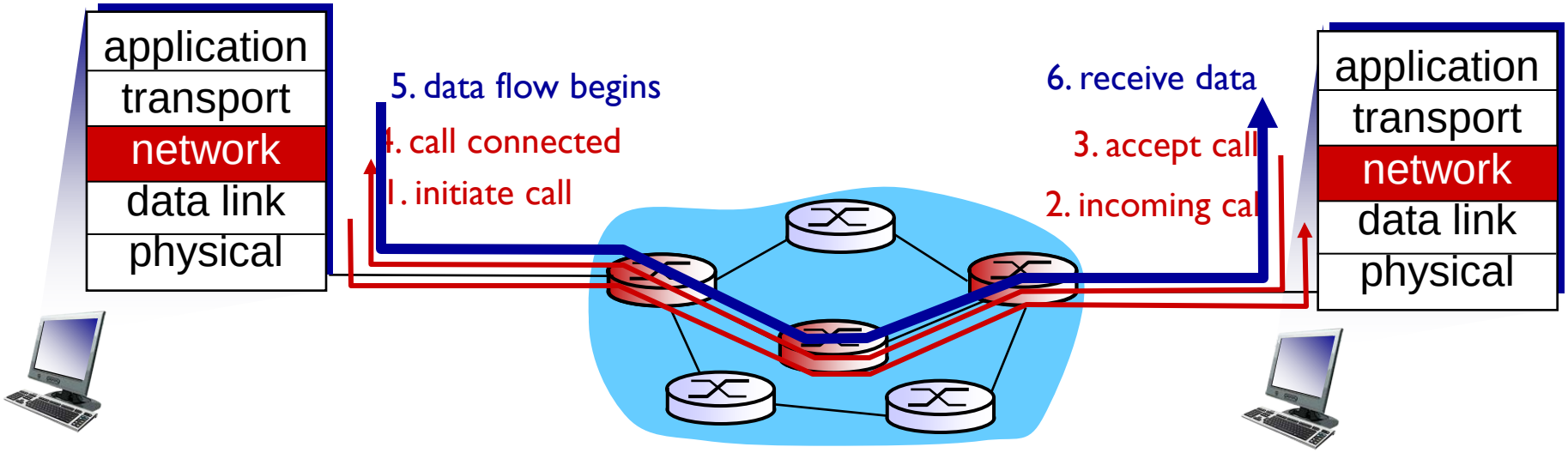
forwarding table in northwest router:

Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
…	…	…	…

VC routers maintain connection state information!

VC Phases

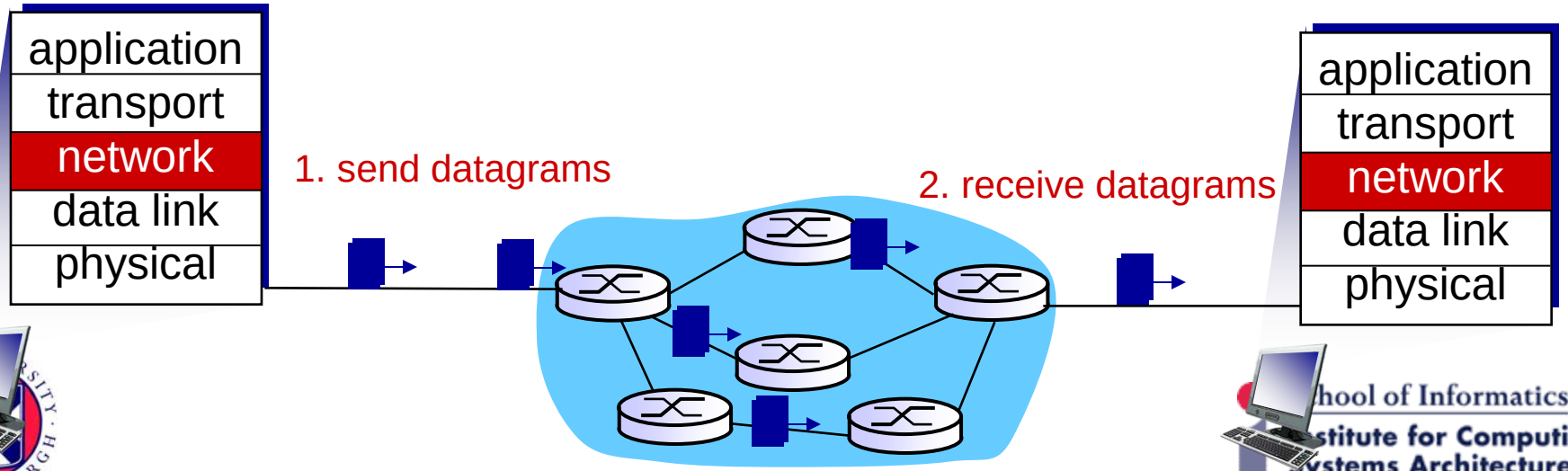
- Setup ➔ data transfer ➔ VC teardown



- VC setup involves all intermediate routers whereas transport layer connections (e.g., TCP) are handled solely by end-hosts
- Messages and protocols to initiate, setup and terminate VCs are called ***signalling messages*** and ***signalling protocols***, respectively.

Datagram Networks

- Example: the Internet
- No connection setup before sending data packets
- Forwarding tables indexed by destination addresses and maps destination addresses to output link interfaces
- Packets stamped with destination address
- Different packets can take different routes ➔ they can arrive out-of-order at the destination



Optimising the Forwarding Table Size

- Having a forwarding table with an entry for each destination address is not practical
  - E.g., 32-bit IP address ➔ 2^{32} potential destination addresses = ~ 4 billion addresses
- Solution:
  - Use ***address prefixes*** instead and match destination address in data packet with prefixes in the forwarding table
  - If multiple matches, pick the ***longest*** match (which corresponds to the most specific route); more on this later

Virtual-Circuit vs. Datagram Networks

- VC networks are relatively more complex within the network as they require per-connection state information at each router
- Datagram networks (simple network, complexity at the edge) ➔ make it easy to interconnect diverse types of networks
- Forwarding tables in the Internet (a datagram network) updated by the routing algorithm every 1-5 minutes unlike VCs which can come and go at a much faster timescale (e.g., in the order of microseconds)
- Harder to provide any service guarantees with datagram networks; more on this shortly

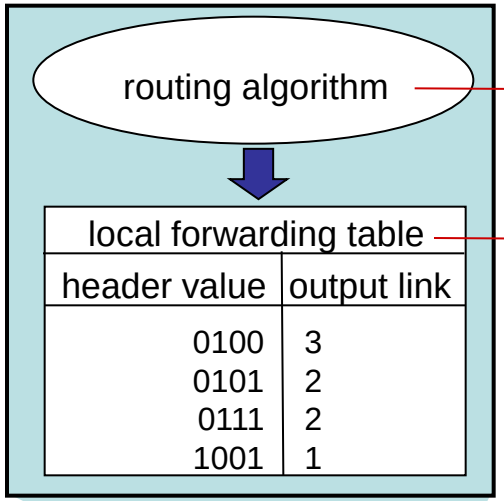
Network Layer Functions

- ***Forwarding:*** transfer of a packet from an incoming link to an outgoing link within a single router

- ***Routing:*** involves all routers, who interact via routing protocols to create and maintain forwarding table entries at each router

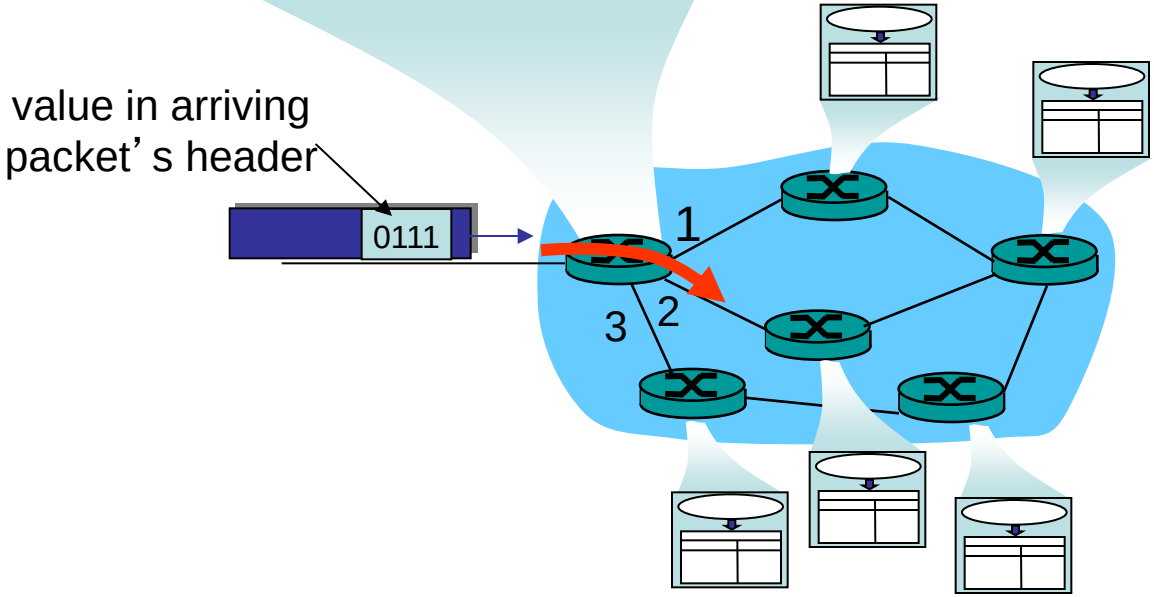
- ***Connection Setup*** (in Virtual-Circuit Networks)

Distinction between Routing and Forwarding Illustrated



routing algorithm determines end-end-path through network

forwarding table determines local forwarding at this router



Network Service Model

- Defines the characteristics of services provided by a network layer, beyond the connectionless vs. connection-oriented aspect we have already seen
- These can include:
  - Guaranteed delivery
  - Guaranteed delivery with bounded delay
  - In-order packet delivery
  - Guaranteed minimum bandwidth
  - Guaranteed maximum jitter
  - Security services

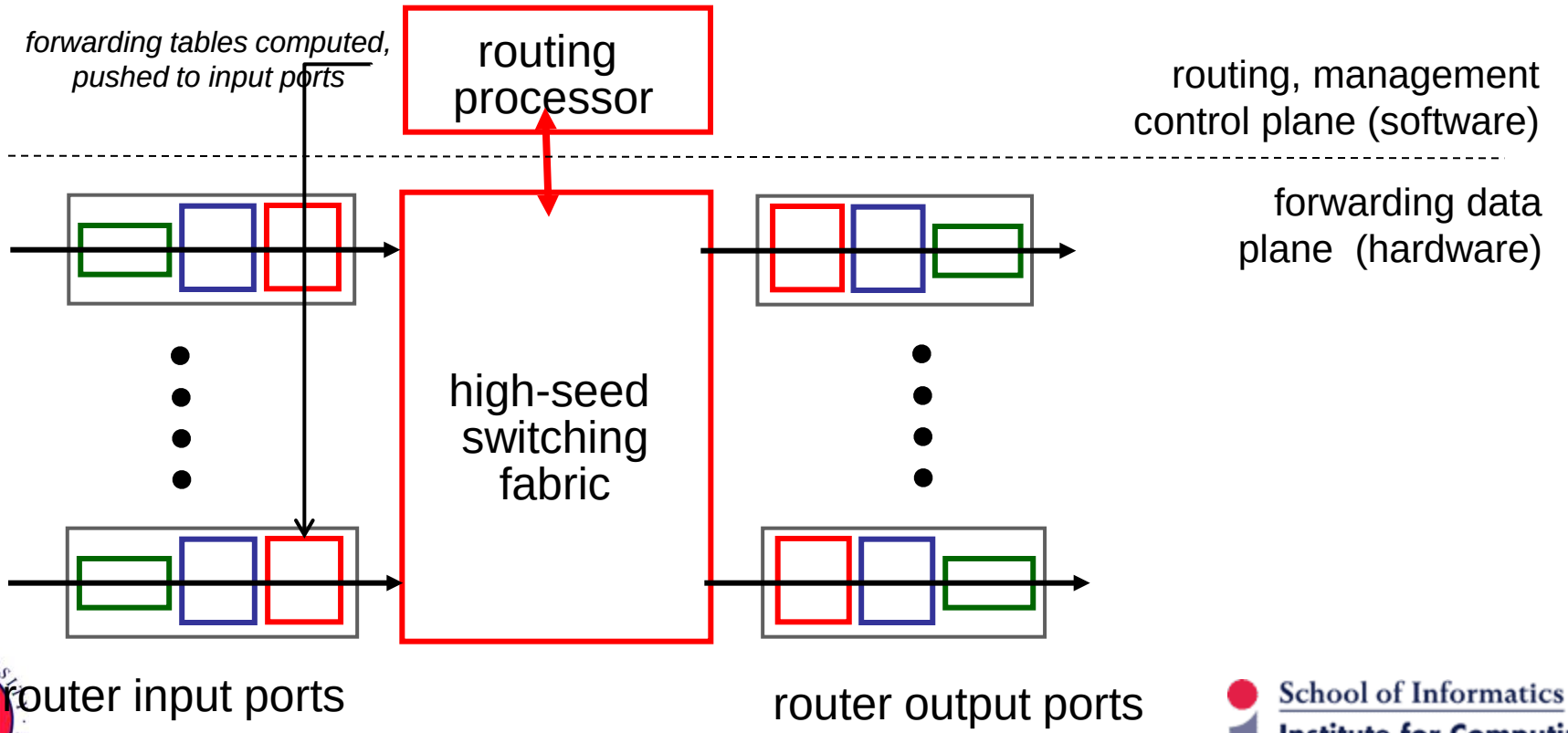
Example Network Service Models

Network Architecture	Service Model	Guarantees ?				Congestion feedback
		Bandwidth	Loss	Order	Timing	
Internet	best effort	none	no	no	no	no (inferred via loss)
ATM	CBR	constant rate	yes	yes	yes	no congestion
ATM	VBR	guaranteed rate	yes	yes	yes	no congestion
ATM	ABR	guaranteed minimum	no	yes	no	yes
ATM	UBR	none	no	yes	no	no

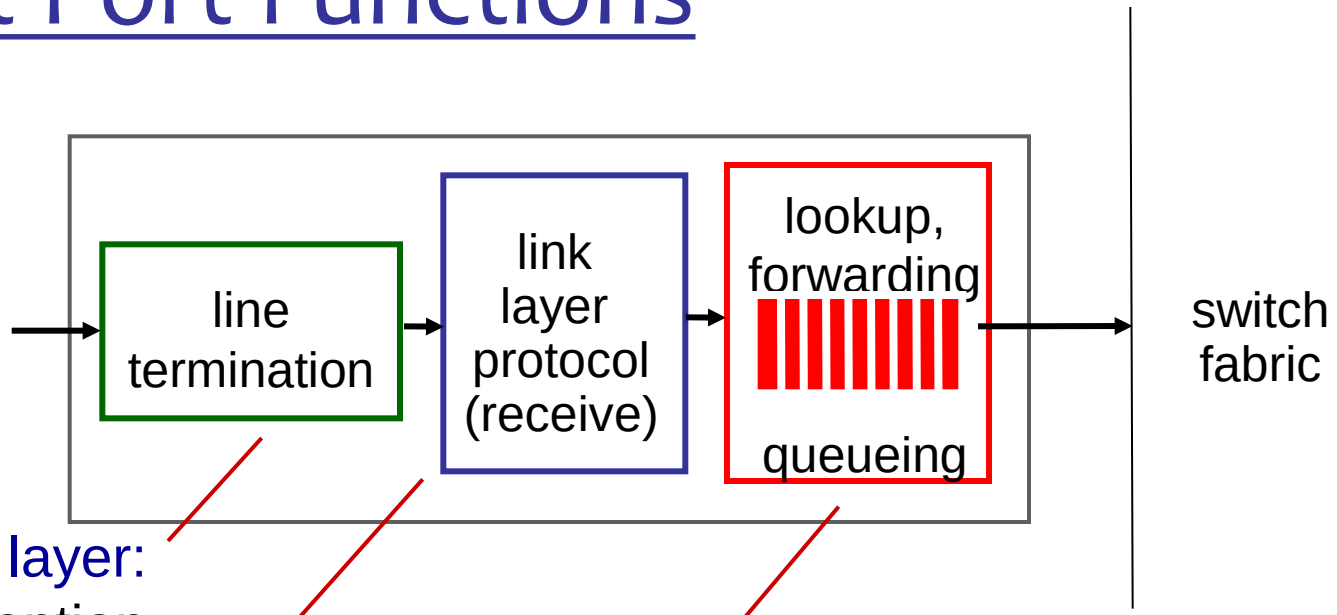
Router Architecture Overview

Two key router functions:

- Run routing algorithms/protocols (RIP, OSPF, BGP)
- *Forwarding* datagrams from incoming to outgoing links



Input Port Functions



physical layer:
bit-level reception

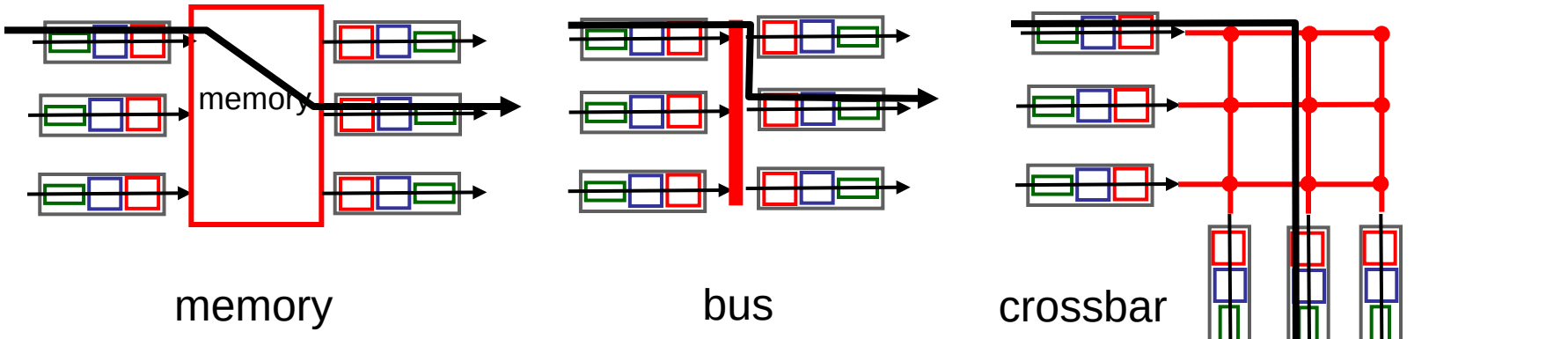
data link layer:
e.g., Ethernet

Decentralized switching:

- Given datagram dest., lookup output port using forwarding table in input port memory (*"match plus action"*)
- Goal: complete input port processing at 'line speed'
- Queuing: if datagrams arrive faster than forwarding rate into switch fabric

Switching Fabrics

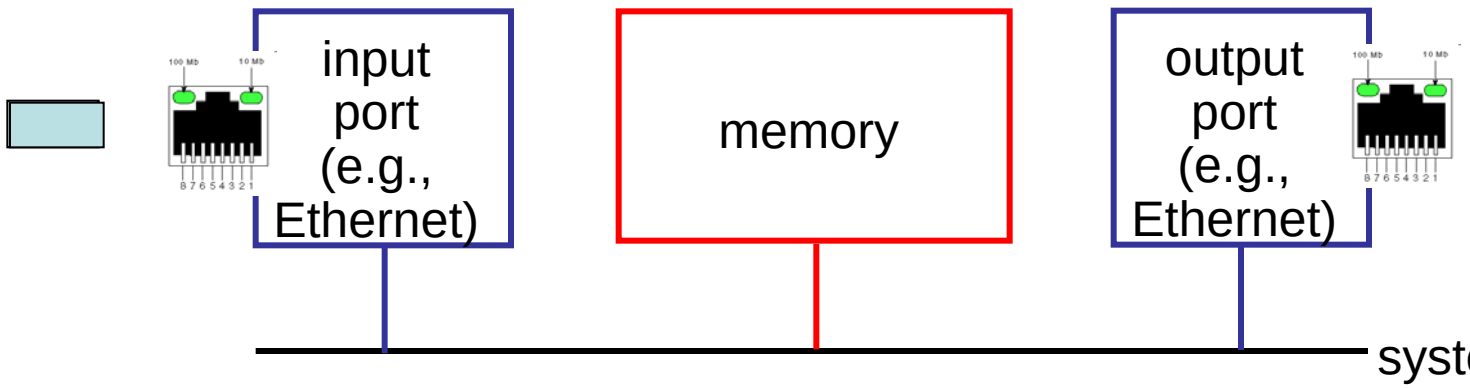
- Transfer packet from input buffer to appropriate output buffer
- Switching rate: rate at which packets can be transferred from inputs to outputs
  - often measured as multiple of input/output line rate
  - N inputs: switching rate N times line rate desirable
- Three types of switching fabrics:



Switching via Memory

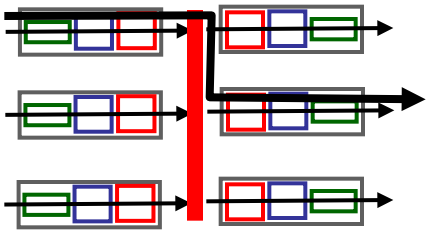
First generation routers:

- Traditional computers with switching under direct control of CPU
- Packet copied to system's memory
- Speed limited by memory bandwidth (2 bus crossings per datagram)



Switching via a Bus

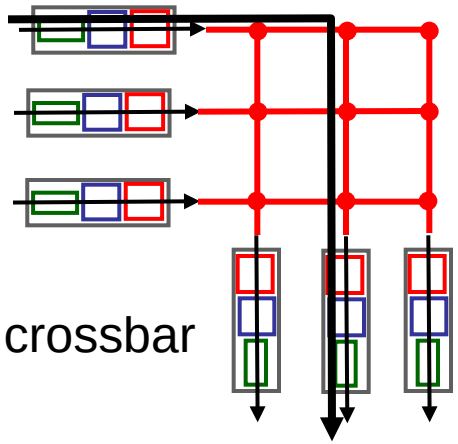
- Datagram from input port memory to output port memory via a shared bus
- *Bus contention:* switching speed limited by bus bandwidth
- 32 Gbps bus, Cisco 5600: sufficient speed for access and enterprise routers



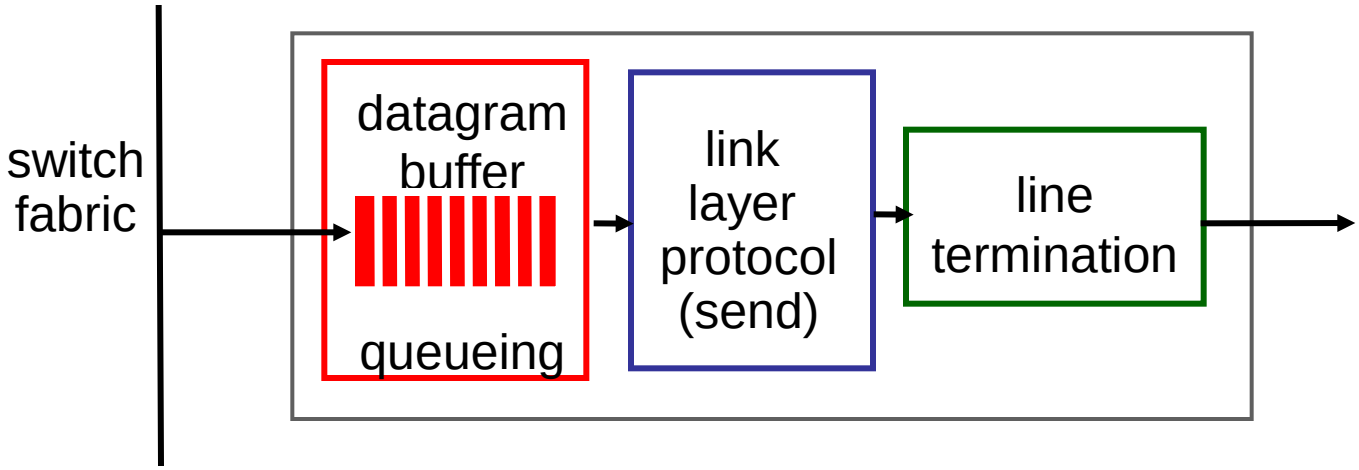
bus

Switching via Interconnection Network

- Overcome bus bandwidth limitations
- Banyan networks, crossbar, other interconnection nets initially developed to connect processors in multiprocessor
- Advanced design: fragmenting datagram into fixed length cells, switch cells through the fabric.
- Cisco 12000: switches 60 Gbps through the interconnection network

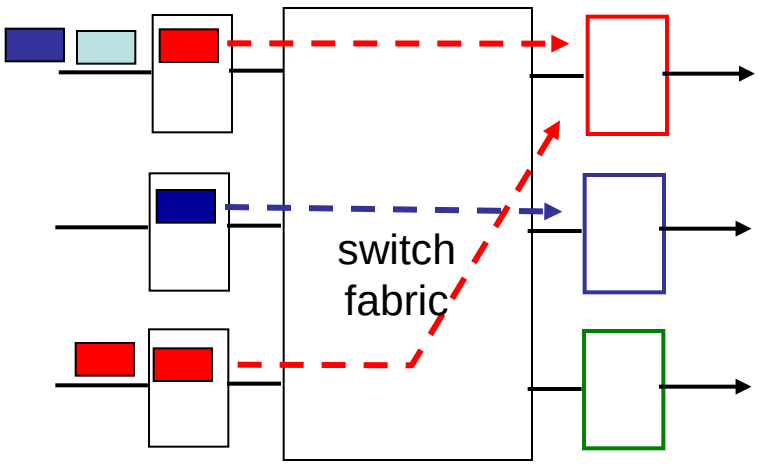


Output Ports

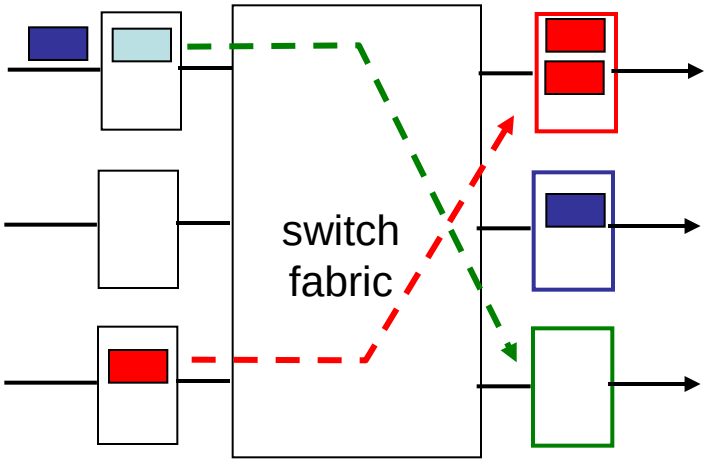


- *Buffering* required when datagrams arrive from fabric faster than the transmission rate
- *Scheduling discipline* chooses among queued datagrams for transmission

Output Port Queueing



at *t*, packets more from input to output



one packet time later

- Buffering when arrival rate via switch exceeds output line speed
- *Queueing (delay) and loss due to output port buffer overflow!*

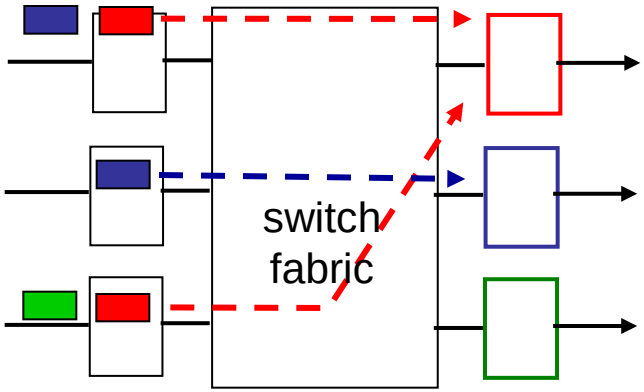
How much buffering?

- RFC 3439 rule of thumb: average buffering equal to "typical" RTT (say 250 msec) times link capacity C
  - e.g., C = 10 Gpbs link: 2.5 Gbit buffer
- Recent recommendation: with *N* flows, buffering equal to

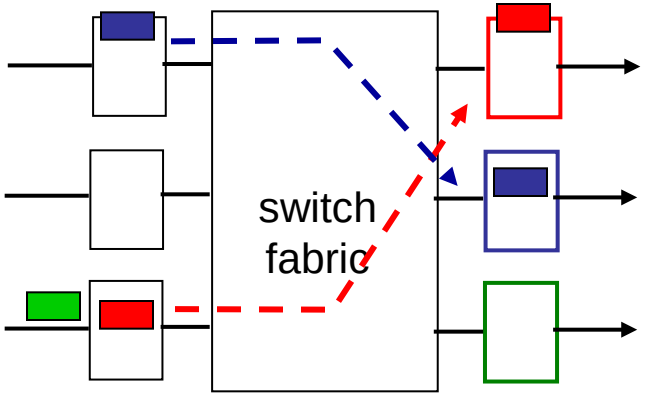
$$\frac{RTT \cdot C}{\sqrt{N}}$$

Input Port Queuing

- Fabric slower than input ports combined ➔ queueing may occur at input queues
  - *queueing delay and loss due to input buffer overflow!*
- Head-of-the-Line (HOL) blocking: queued datagram at front of queue prevents others in queue from moving forward



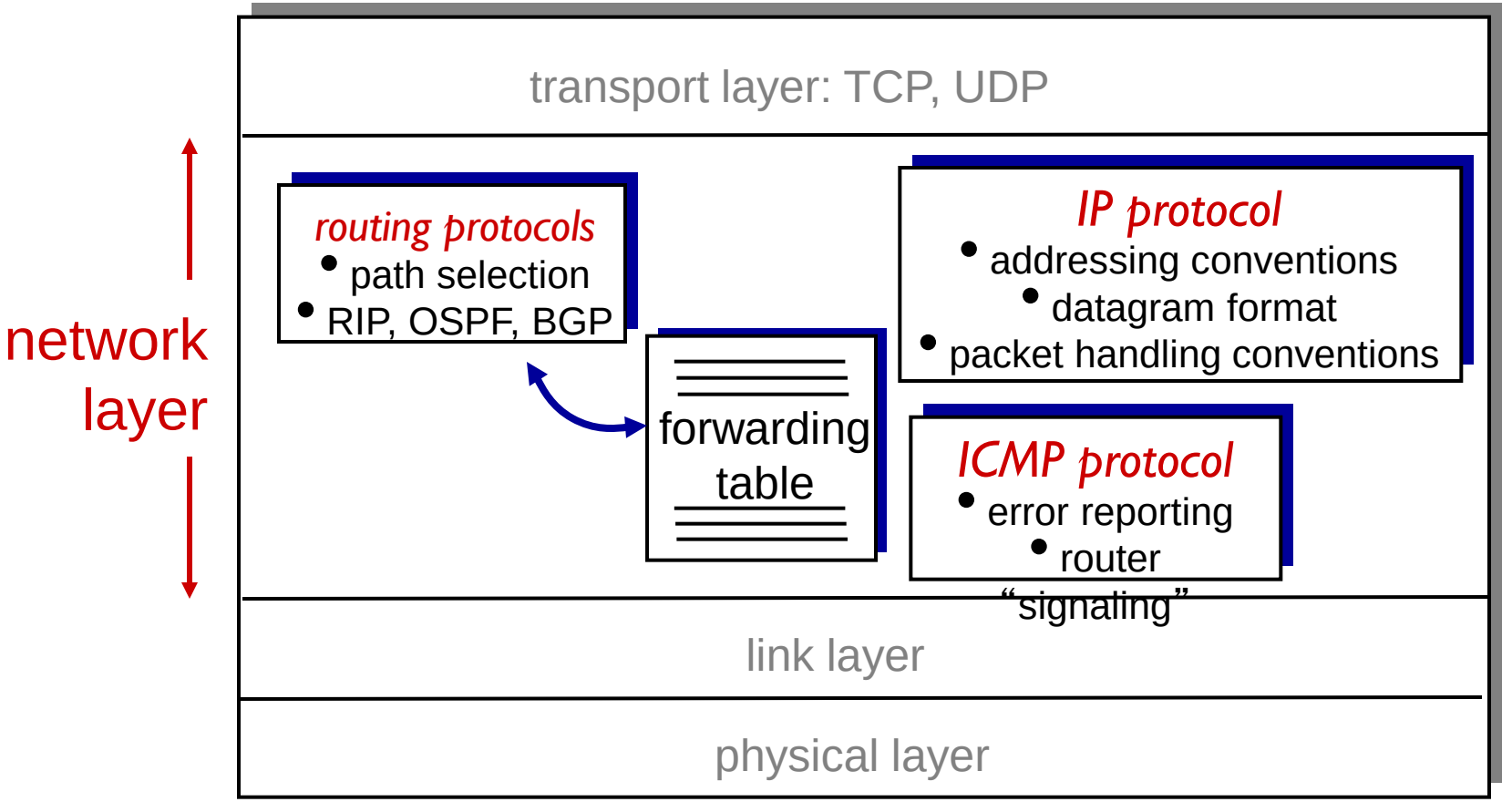
output port contention:
only one red datagram can be transferred.
lower red packet is blocked



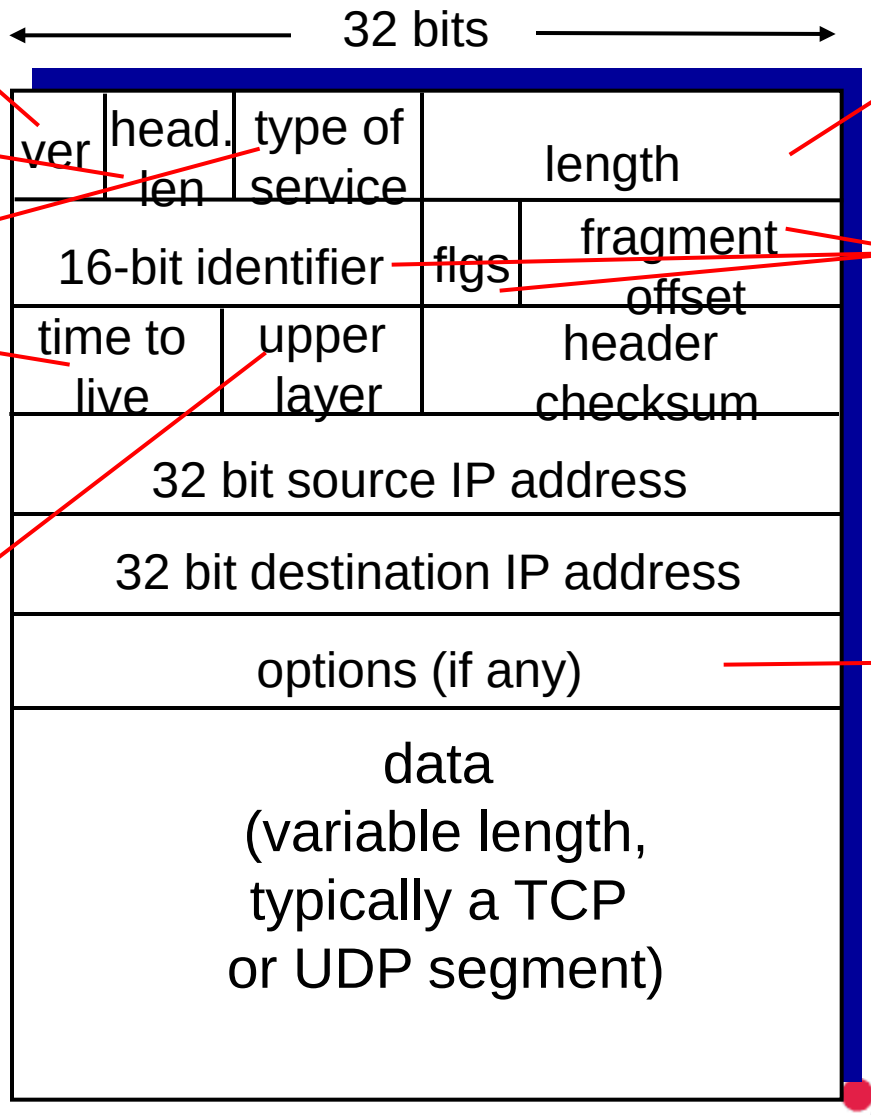
one packet time later: green packet experiences HOL blocking

Internet's Network Layer

host, router network layer functions:



IPv4 Datagram Format



IP protocol version number

header length (bytes)

"type" of data

max number remaining hops (decremented at each router)

upper layer protocol to deliver payload to

total datagram length (bytes)

for fragmentation/ reassembly

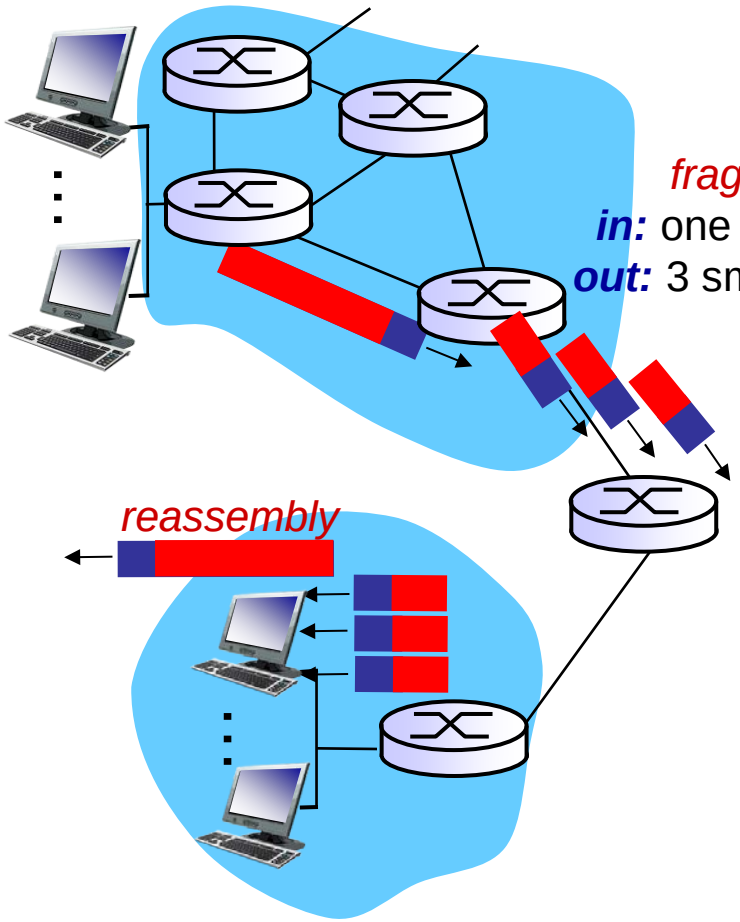
e.g. timestamp, record route taken, specify list of routers to visit.

how much overhead?

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

IPv4 Fragmentation & Reassembly

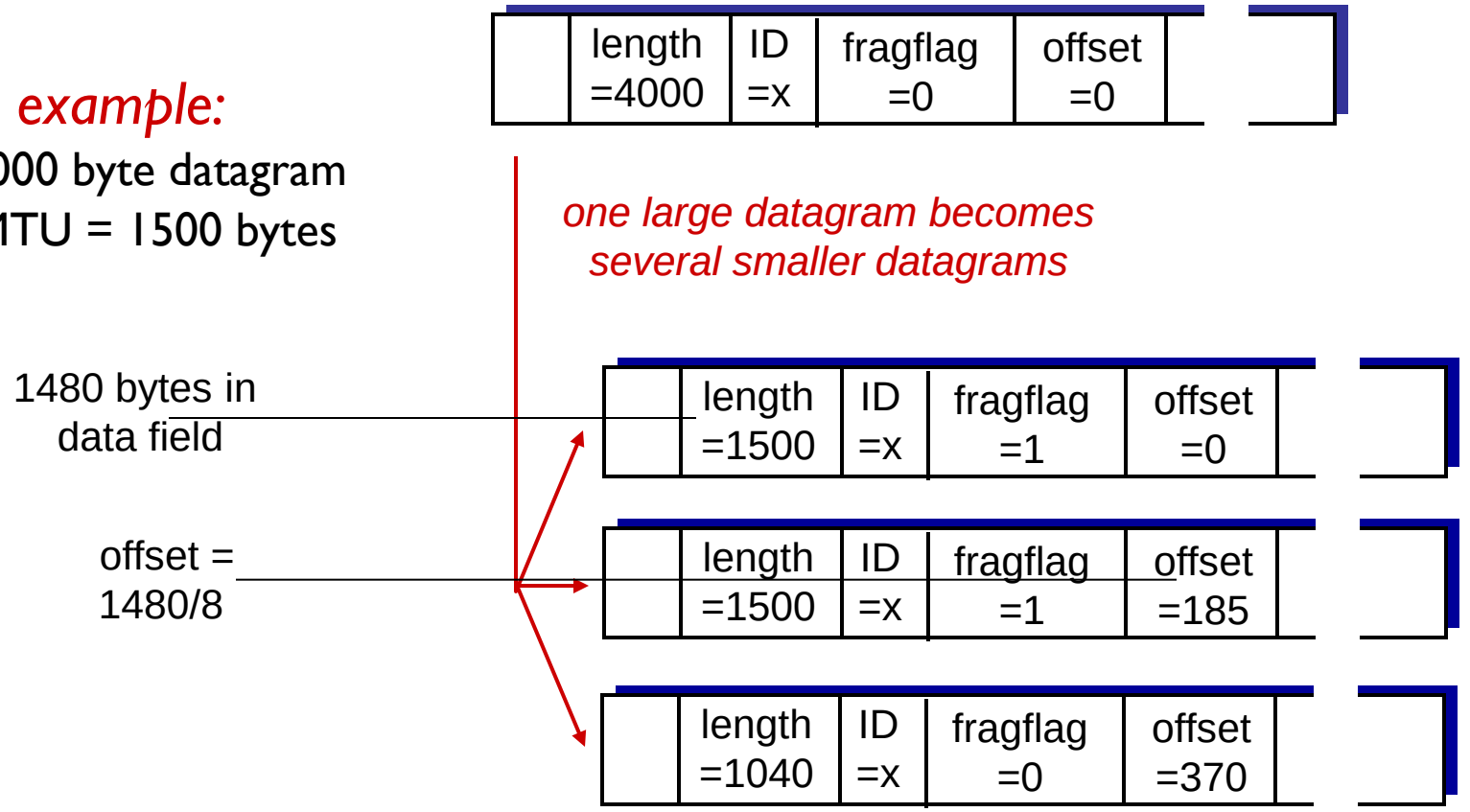
- Network links have MTU (max.transfer size) - largest possible link-level frame
  - different link types, different MTUs
- Large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify, order related fragments



IPv4 Fragmentation & Reassembly

example:

- 4000 byte datagram
- MTU = 1500 bytes



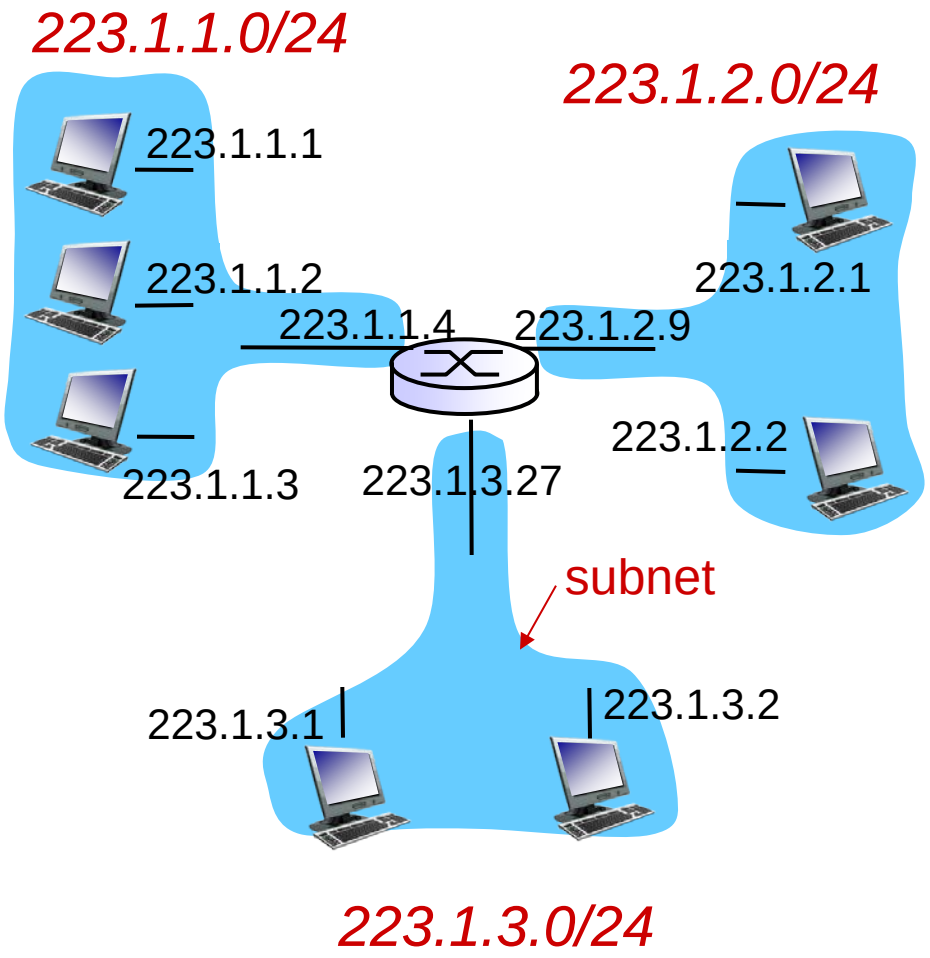
IPv4 Addressing

- In the Internet, addressing and routing are intertwined
- IP address is assigned to an interface because:
  - Routing requires unique addresses, and
  - A router or host may have more than one interface (and correspondingly multiple incident links)
- ***Dotted-decimal notation*** for IP addresses
- Example: 223.1.1.1

Subnets and Addressing within a Subnet

- ***Subnet*** refers to the set of interfaces that are *directly* reachable from each other from IP perspective
- An interface's IP address determined by the ***subnet*** it is connected to
- Interfaces within a subnet are assigned IP addresses that have a common prefix, usually called ***subnet mask***
- Subnet mask is represented using the ***/x notation***, meaning that *x* most significant bits of each IP address in a subnet corresponds the subnet part of the address

Subnets Example



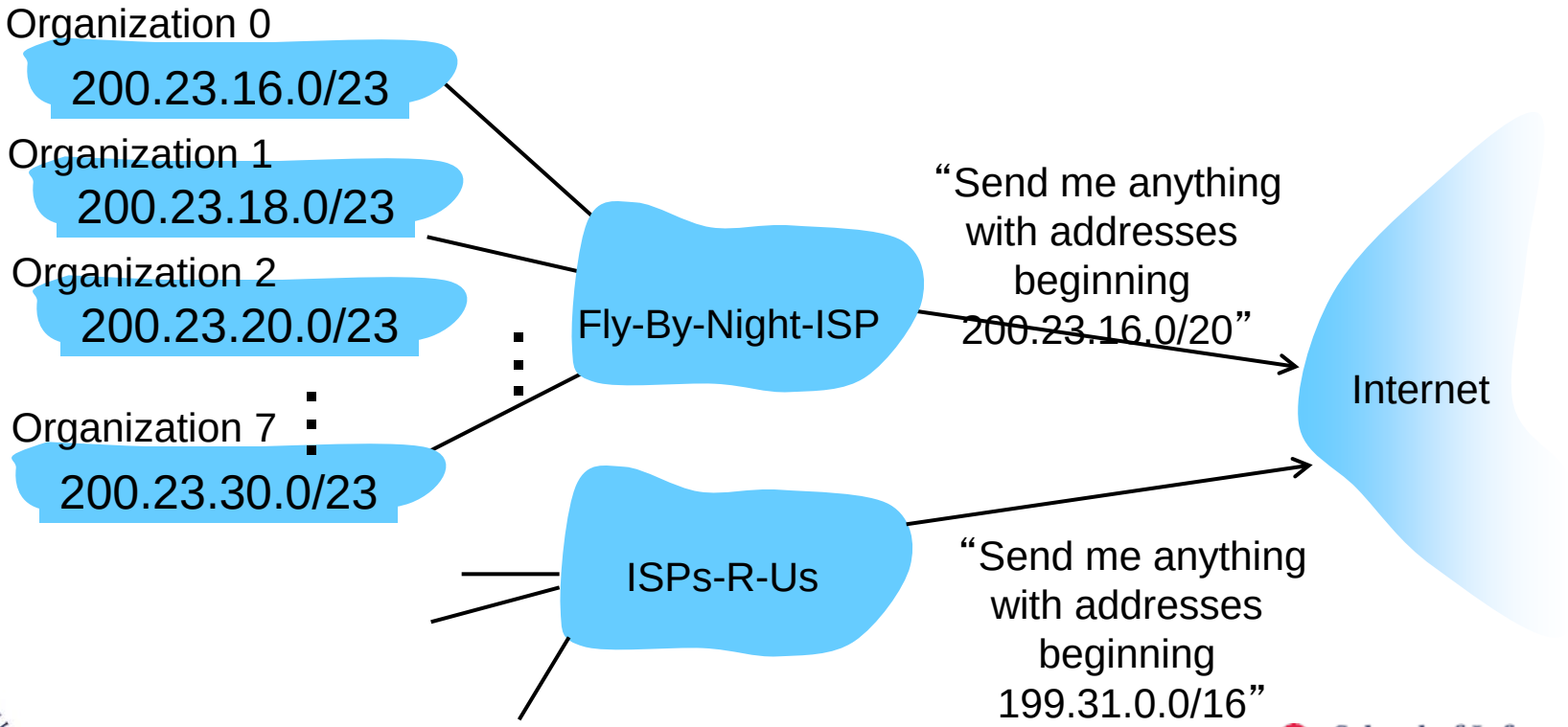
Internet Address Assignment Strategy

- Known as ***Classless Inter-Domain Routing (CIDR)***
- Generalises the notion of subnet addressing
- 32-bit IP address is divided into two parts and represented using the notation ***a.b.c.d/x***
- Here *x* indicates the number of most significant bits forming the first part of the address, referred to as ***(network) prefix***
- Each organisation assigned a set of addresses that share the same prefix
- As a result, only the prefix is sufficient for routers external to the organisation to determine a route to any destination within the organisation

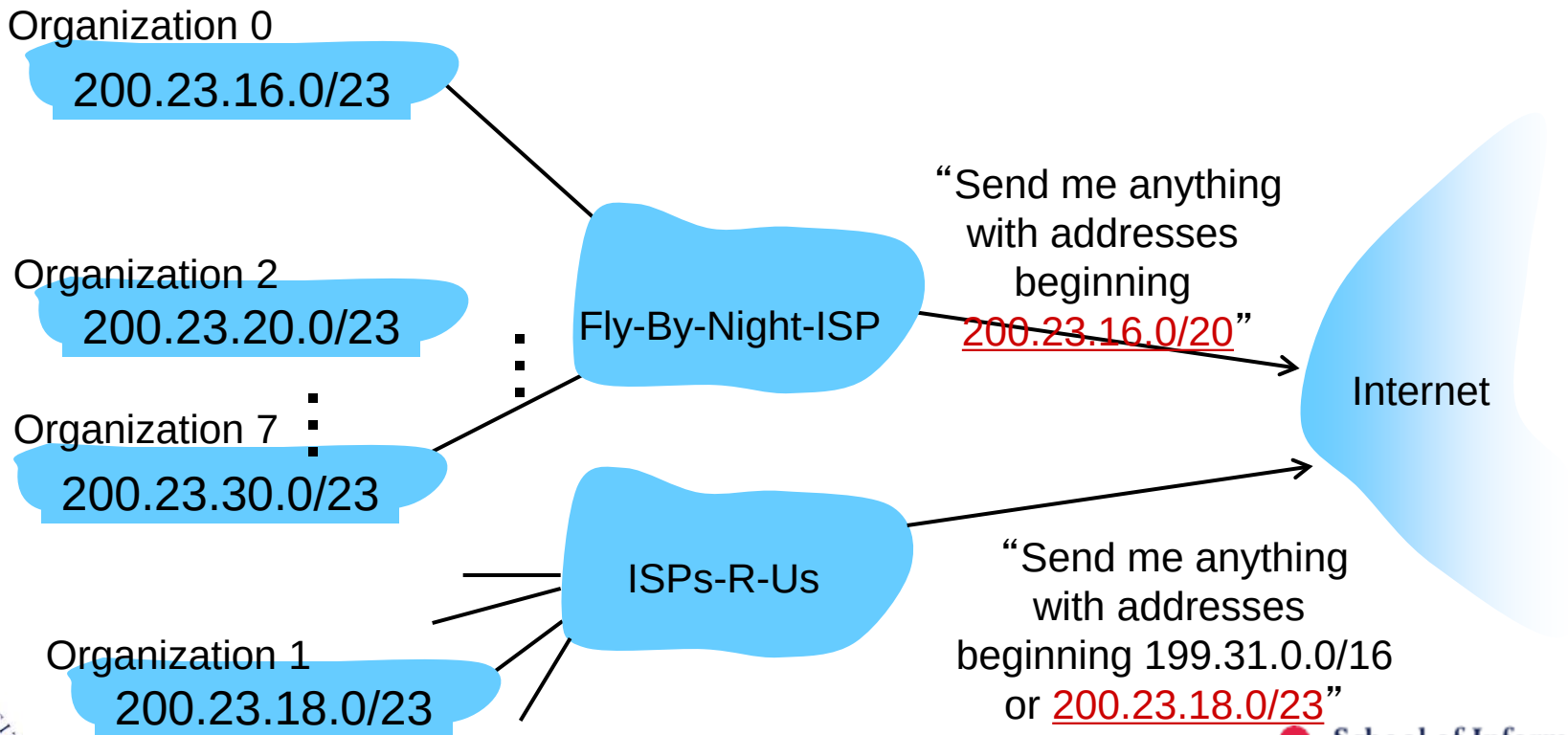
Route Aggregation

- *32-x* bits in the IP address are used for distinguishing between different interfaces of hosts/routers inside the organisation
  - Could be further divided to use multiple levels of subnetting ➔ **address/route aggregation** or **route summarisation**
  - Exceptions handled by advertising any additional (longer and more specific) prefixes
  - Then the use of longest prefix matching based route selection ensures that packets are delivered correctly even with exceptions

Route Aggregation Example



Route Aggregation with More Specific Routes: Example



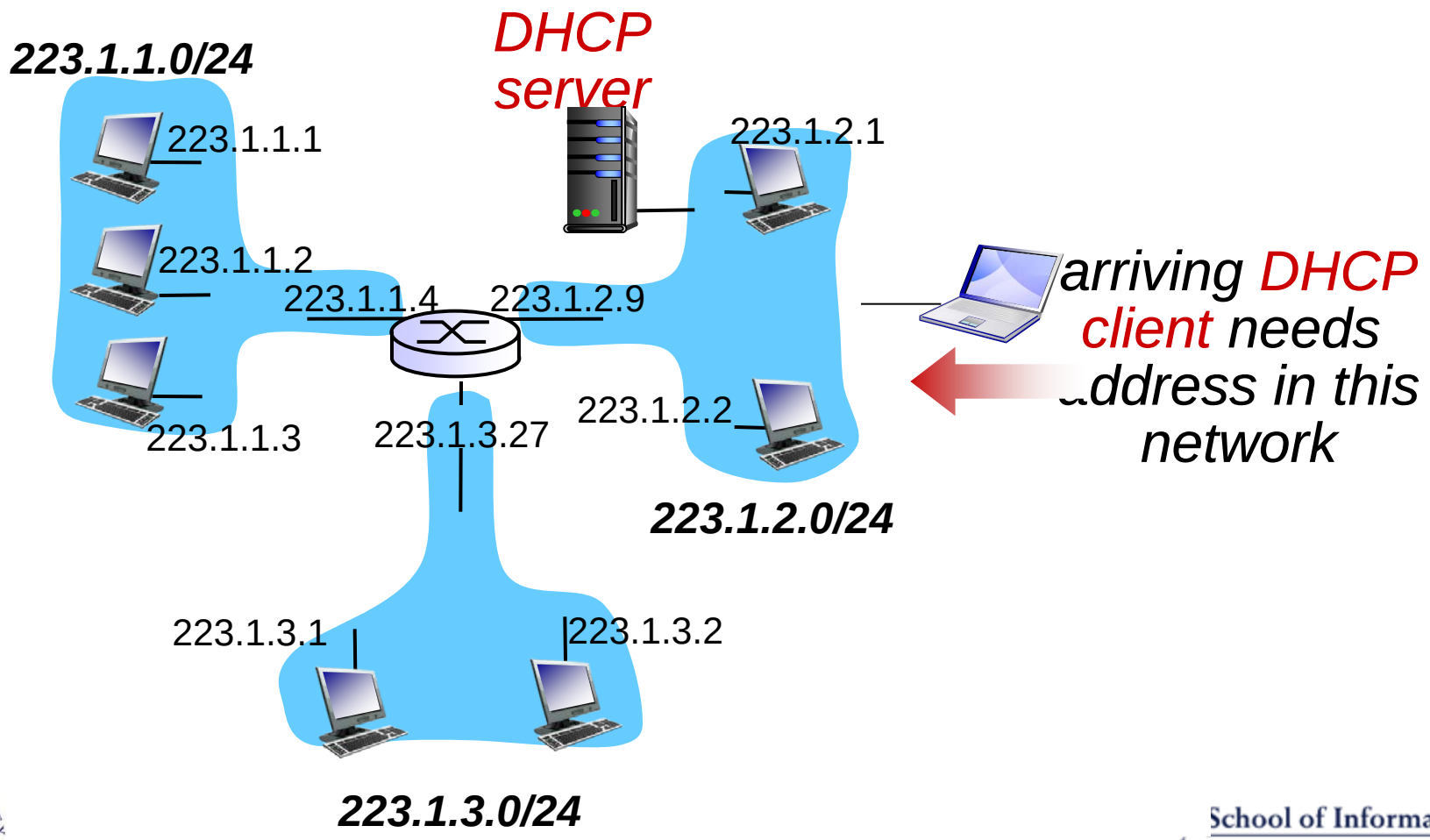
Obtaining a Block of IP Addresses

- From an organisation that already holds a larger block (e.g., ISP)
- Or from ICANN (Internet Corporation for Assigned Names and Numbers), the global authority responsible for managing IP addresses, root DNS servers, assigning domain names via regional Internet registries (e.g., RIPE)

Obtaining an IP Address for a Host

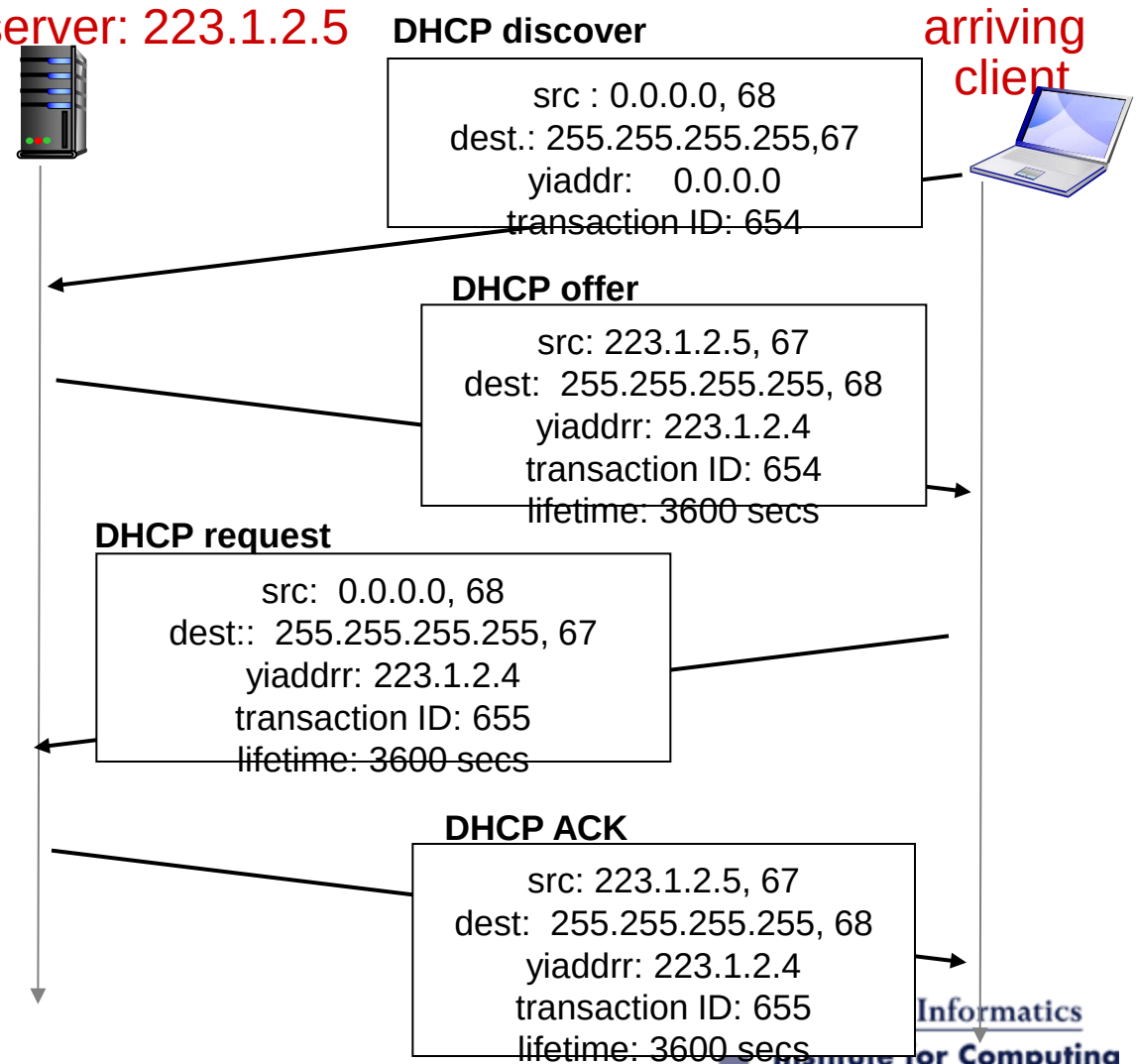
- Can be done manually but typical to use ***Dynamic Host Configuration Protocol (DHCP)*** for automatically getting an IP address
- DHCP can be configured to provide the same or a different address each time a host connects to the network
  - Additionally provides subnet mask, default gateway (first-hop router), local DNS server, and a mechanism to renew address lease

- DHCP is a client-server protocol
  - If no DHCP server in the local subnet, then use a remote DHCP server via the router (which acts as a DHCP relay agent)



DHCP in action

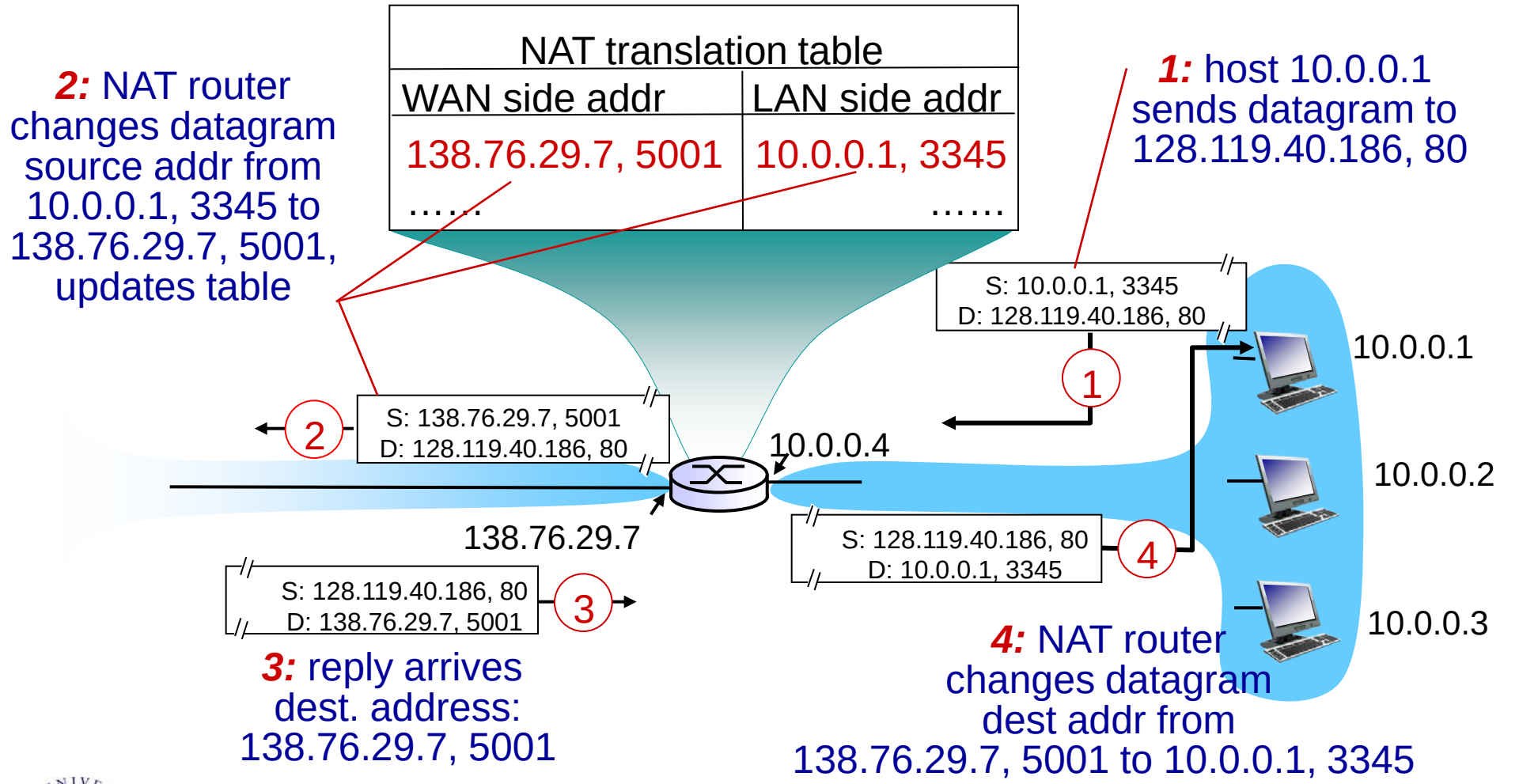
- Assuming a local DHCP server, address acquisition with DHCP for a new host is a 4-step process:
  1. DHCP server discovery
  2. DHCP server offer(s)
  3. DHCP request
  4. DHCP ACK



Network Address Translation (NAT)

- Motivation: not enough addresses for every end-host
- Idea: have a NAT-enabled router with a public (globally unique) IP address hide the many end-hosts which are assigned private (non-unique) addresses
  - Role of the NAT router is to translate between private and public addresses for each packet to/from the Internet

NAT Example



Network Address Translation (NAT)

- Motivation: not enough addresses for every end-host
- Idea: have a NAT-enabled router with a public (globally unique) IP address hide the many end-hosts which are assigned private (non-unique) addresses
  - Role of the NAT router is to translate between private and public addresses for each packet to/from the Internet
- Note that NAT is a patch that violates the original intended use of port numbers, layering and end-to-end argument
- Still not straightforward for a host behind a NAT to act as a server
  - Universal Plug and Play (UPnP) protocol eases this problem