CSE3213 Computer Network I

Network Layer
(7.1, 7.3, 8.2.1-8.2.3)

Course page:
http://www.cse.yorku.ca/course/3213

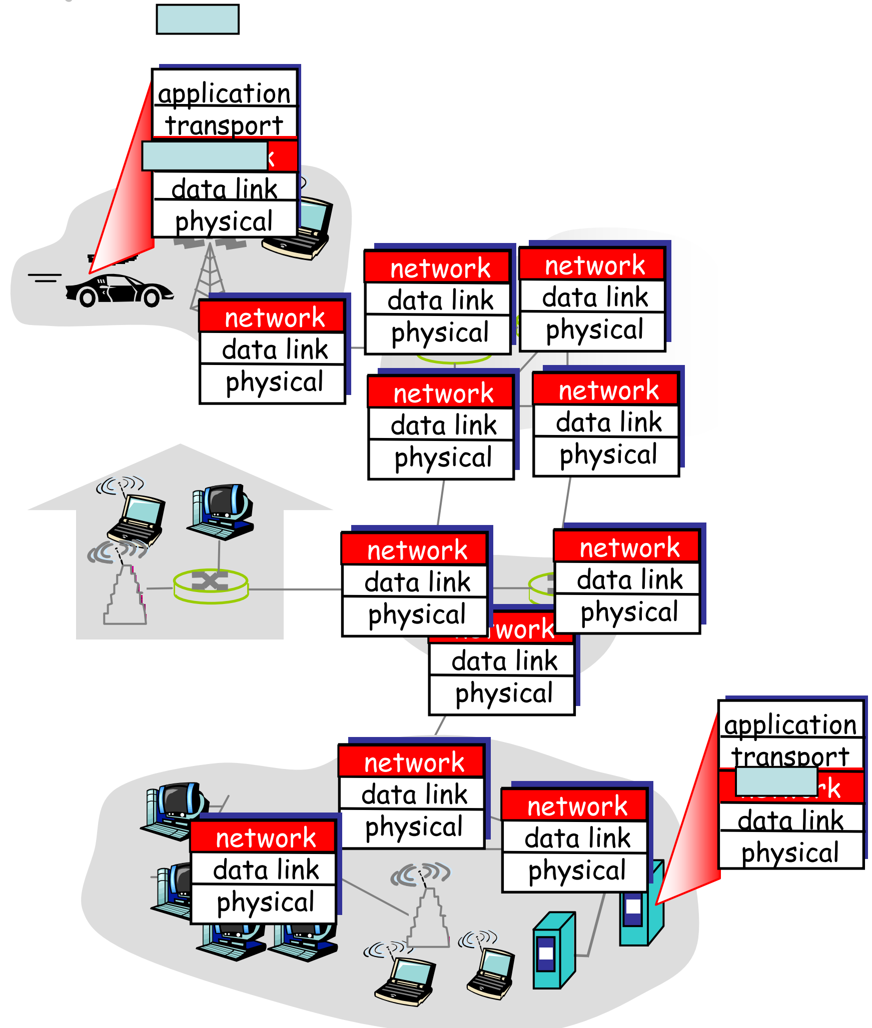
Slides modified from Alberto Leon-Garcia and Indra Widjaja and Jim Kurose

Network Layer

- Introduction
- Virtual circuit and datagram networks
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing

Network layer

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



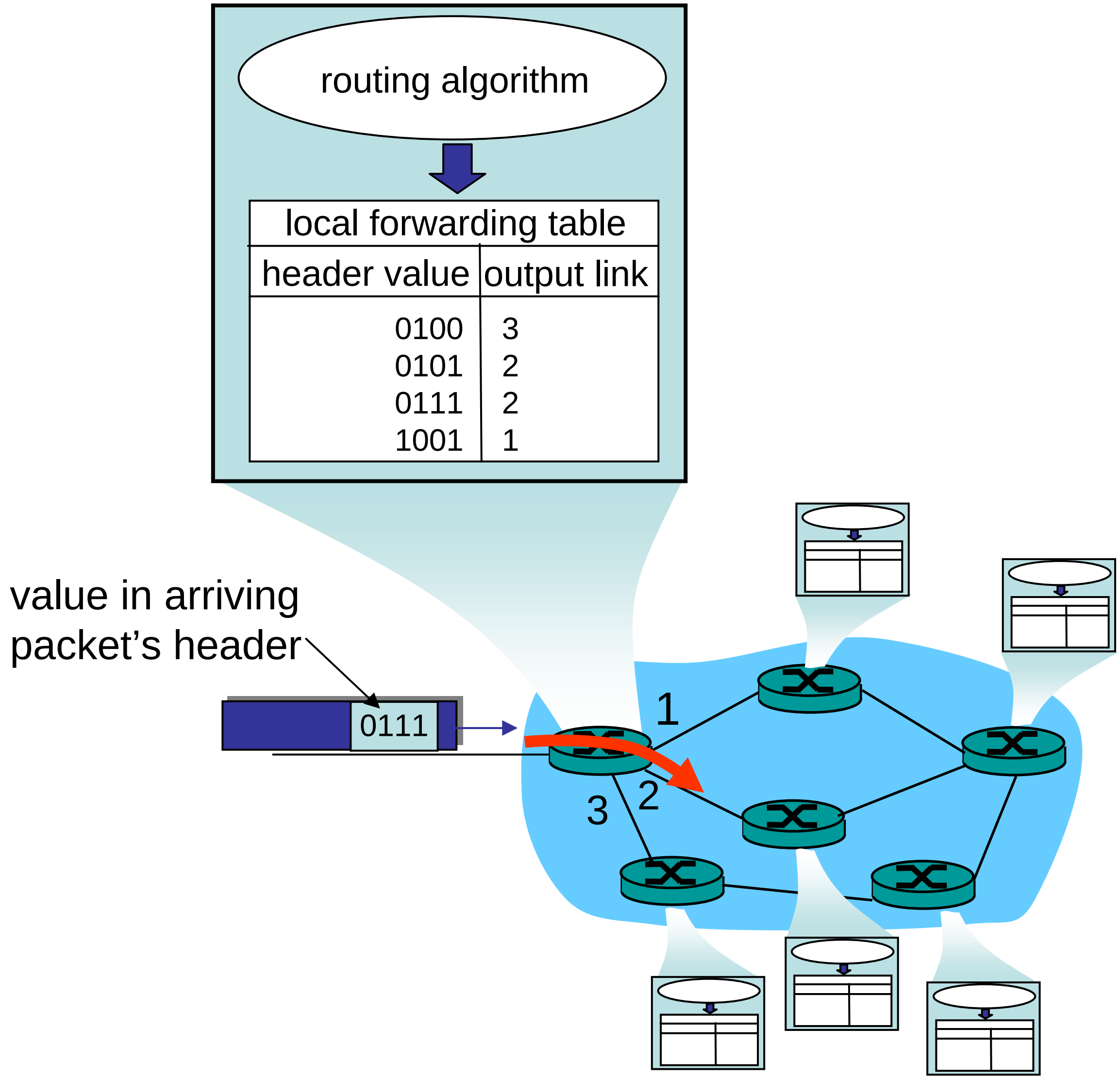
Two Key Network-Layer Functions

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

analogy:

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

Interplay between routing and forwarding



Connection setup

- 3rd important function in *some* network architectures:
  - ATM, frame relay, X.25
- before datagrams flow, two end hosts *and* intervening routers establish virtual connection
  - routers get involved
- network vs transport layer connection service:
  - network: between two hosts (may also involve intervening routers in case of VCs)
  - transport: between two processes

Network service model

Q: What *service model* for "channel" transporting datagrams from sender to receiver?

Example services for individual datagrams:

- guaranteed delivery
- guaranteed delivery with less than 40 msec delay

Example services for a flow of datagrams:

- in-order datagram delivery
- guaranteed minimum bandwidth to flow
- restrictions on changes in inter-packet spacing

Network Layer

- Introduction
- Virtual circuit and datagram networks
- What's inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing

Network layer connection and connection-less service

- datagram network provides network-layer connectionless service
- VC network provides network-layer connection service
- analogous to the transport-layer services, but:
  - service: host-to-host
  - no choice: network provides one or the other
  - implementation: in network core

Virtual circuits

"source-to-dest path behaves much like telephone circuit"
- performance-wise
- network actions along source-to-dest path

- call setup, teardown for each call *before* data can flow
- each packet carries VC identifier (not destination host address)
- *every* router on source-dest path maintains "state" for each passing connection
- link, router resources (bandwidth, buffers) may be *allocated* to VC (dedicated resources = predictable service)

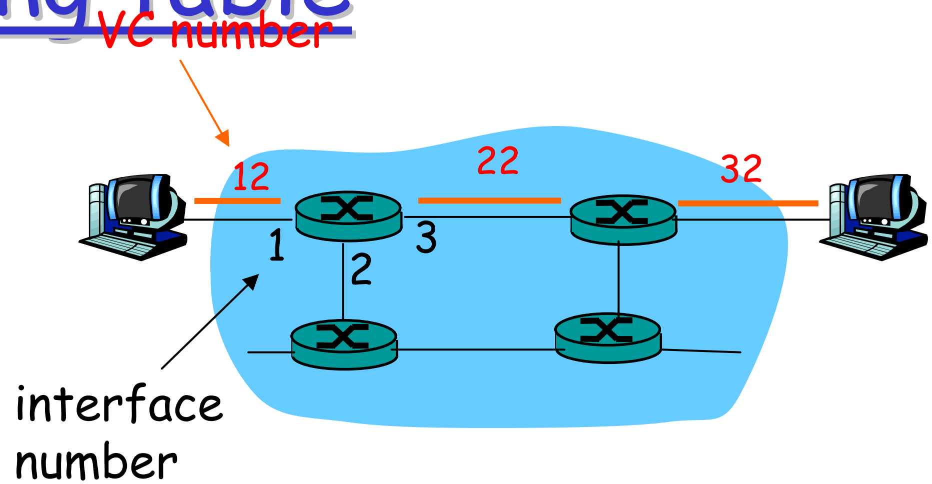
VC implementation

a VC consists of:

1. path from source to destination
2. VC numbers, one number for each link along path
3. entries in forwarding tables in routers along path

- packet belonging to VC carries VC number (rather than dest address)
- VC number can be changed on each link.
  - New VC number comes from forwarding table

Forwarding table



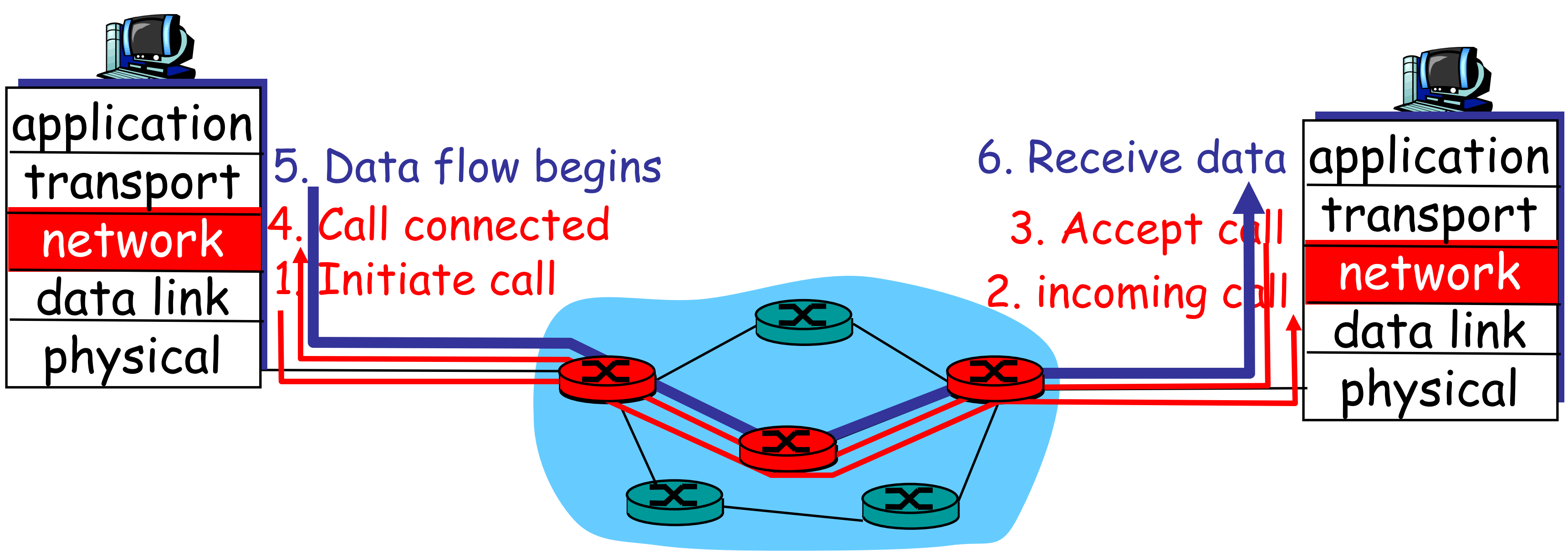
Forwarding table in northwest router:

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

Routers maintain connection state information!

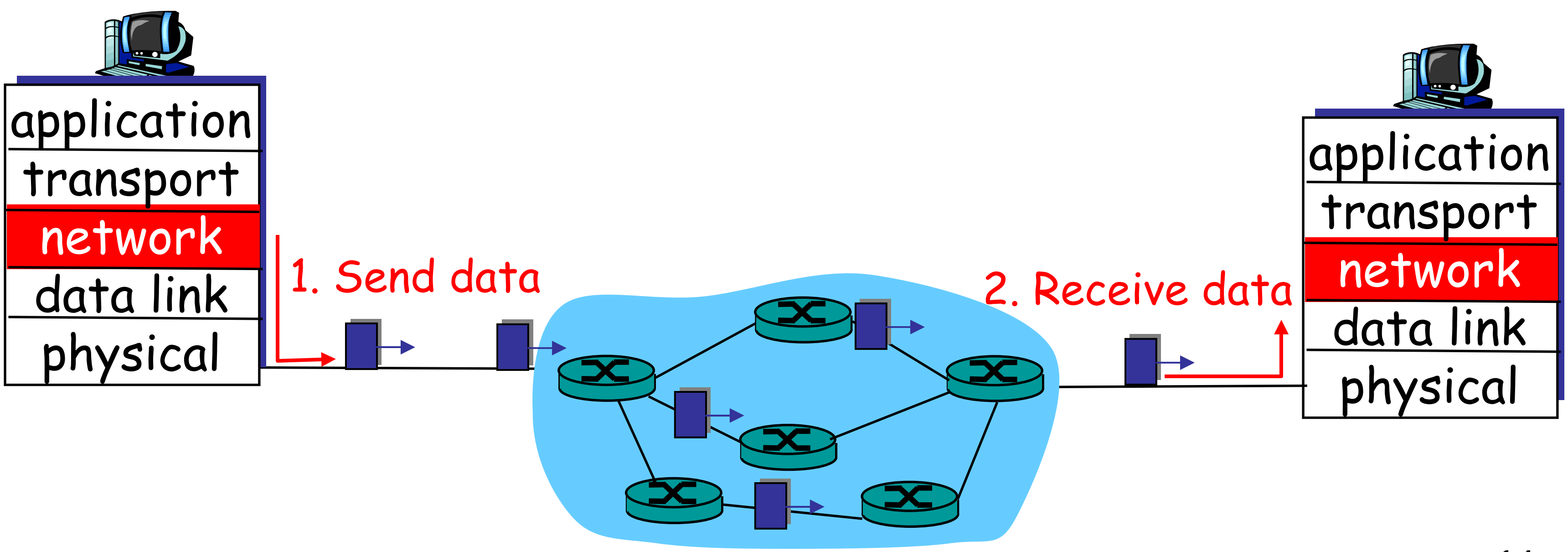
Virtual circuits: signaling protocols

- used to setup, maintain teardown VC
- used in ATM, frame-relay, X.25
- not used in today's Internet



Datagram networks

- no call setup at network layer
- routers: no state about end-to-end connections
  - no network-level concept of "connection"
- packets forwarded using destination host address
  - packets between same source-dest pair may take different paths



Forwarding table

4 billion possible entries

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

Datagram or VC network: why?

Internet (datagram)

- data exchange among computers
  - "elastic" service, no strict timing req.
- "smart" end systems (computers)
  - can adapt, perform control, error recovery
  - simple inside network, complexity at "edge"
- many link types
  - different characteristics
  - uniform service difficult

ATM (VC)

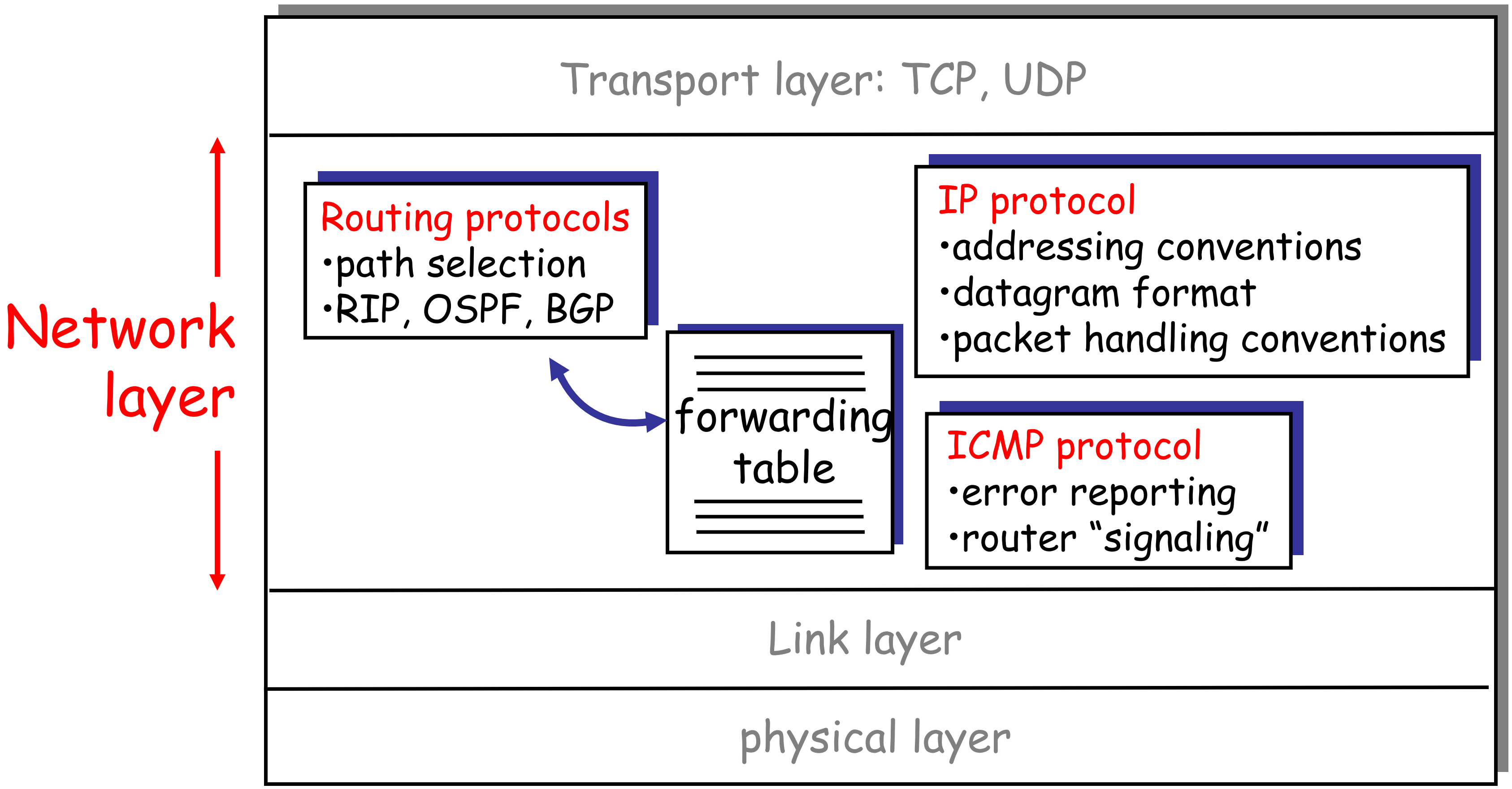
- evolved from telephony
- human conversation:
  - strict timing, reliability requirements
  - need for guaranteed service
- "dumb" end systems
  - telephones
  - complexity inside network

Network Layer

- Introduction
- Virtual circuit and datagram networks
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing

The Internet Network layer

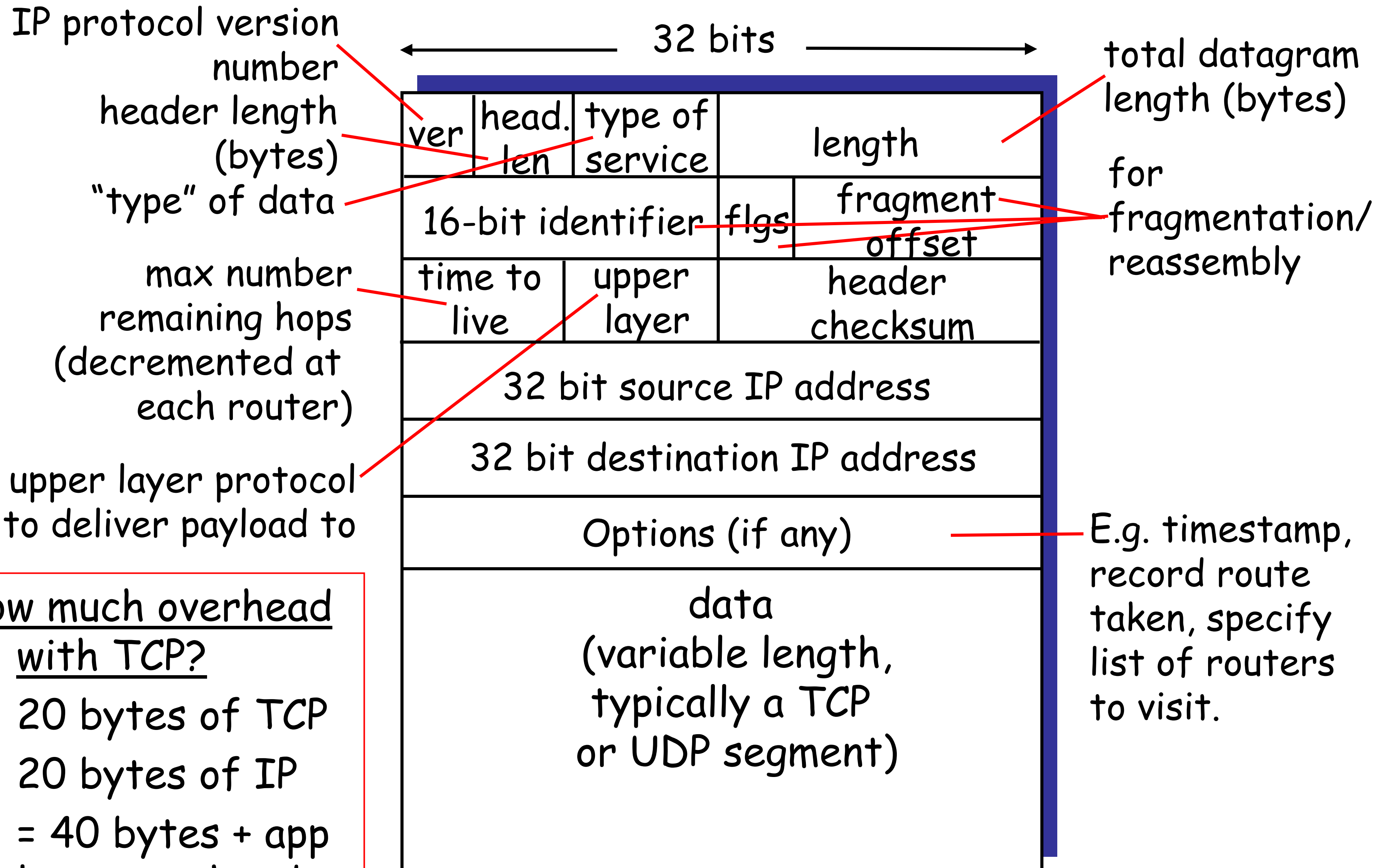
Host, router network layer functions:



Network Layer

- Introduction
- Virtual circuit and datagram networks
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing

IP datagram format

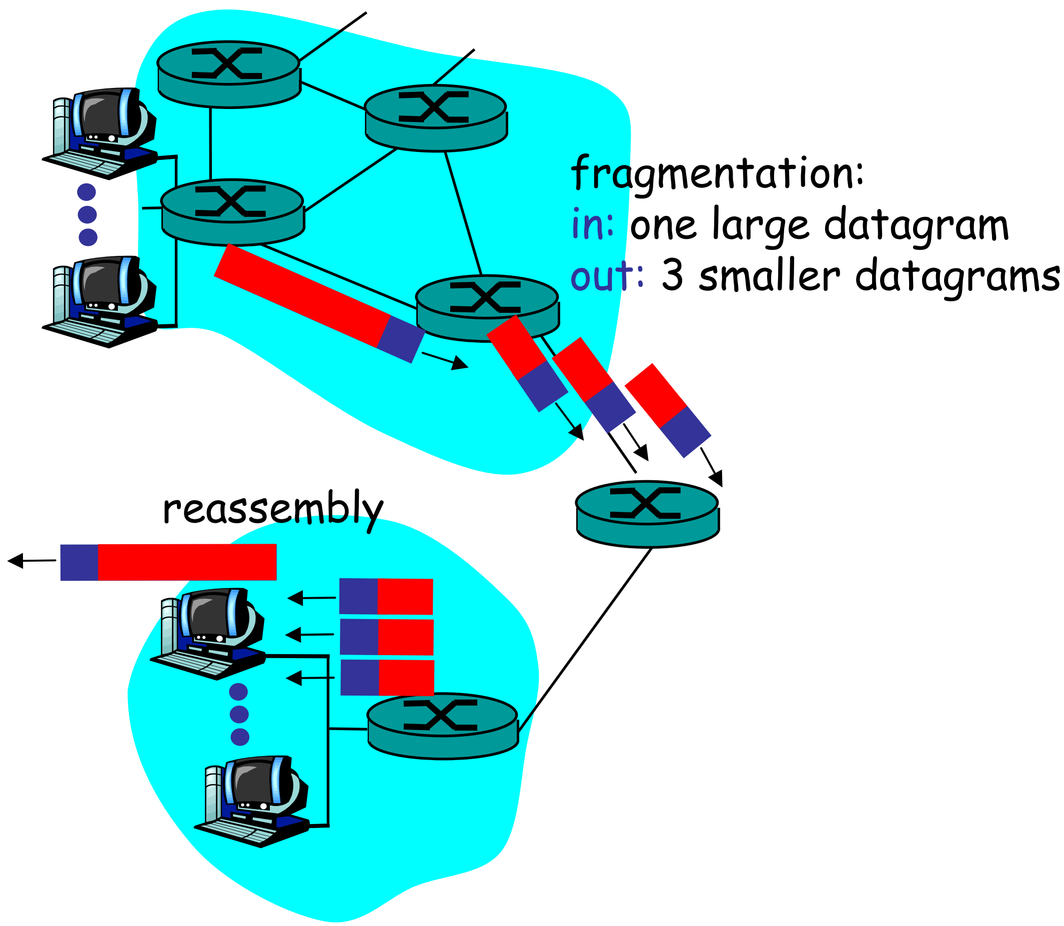


<u>how much overhead with TCP?</u>

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

IP Fragmentation & Reassembly

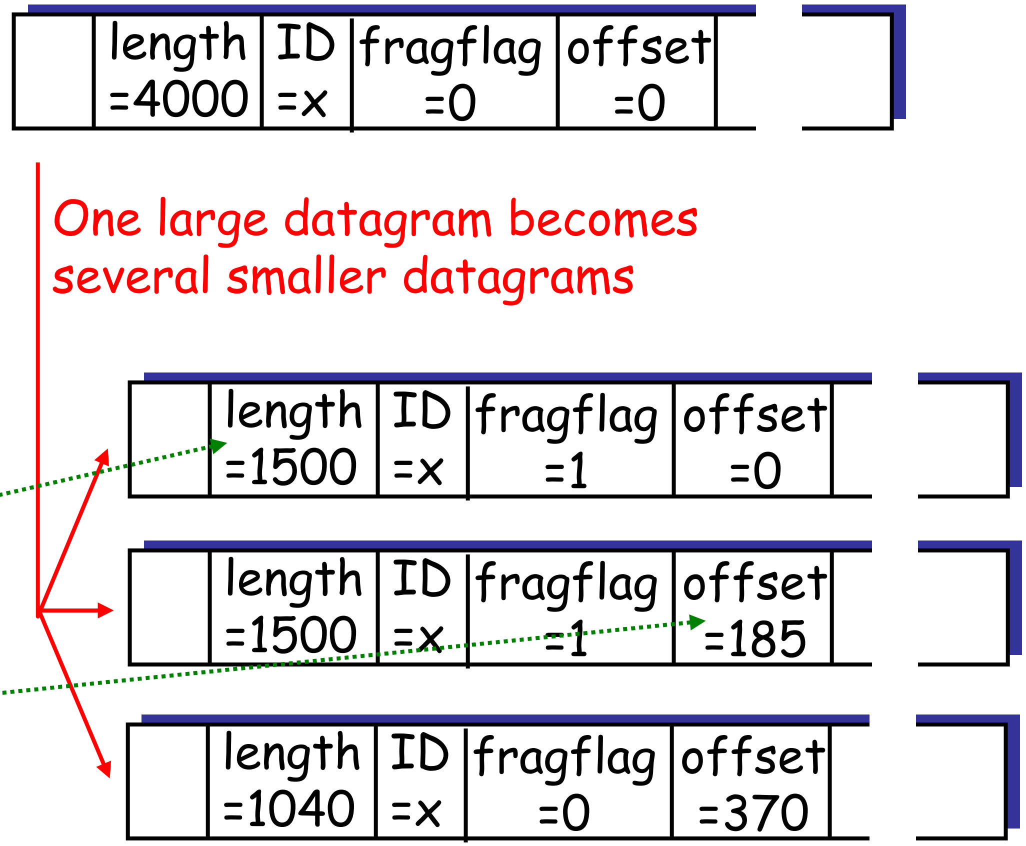
- network links have MTU (max.transfer size) - largest possible link-level frame.
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify,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



IP Fragmentation and Reassembly

Example

- 4000 byte datagram
- MTU = 1500 bytes



1480 bytes in data field

offset = 1480/8

Network Layer

- Introduction
- Virtual circuit and datagram networks
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing

Classful Addresses

Class A

	7 bits	**24 bits**
0	netid	hostid

- 126 networks with up to 16 million hosts

1.0.0.0 to 127.255.255.255

Class B

		14 bits	**16 bits**
1	0	netid	hostid

- 16,382 networks with up to 64,000 hosts

128.0.0.0 to 191.255.255.255

Class C

			22 bits	**8 bits**
1	1	0	netid	hostid

- 2 million networks with up to 254 hosts

192.0.0.0 to 223.255.255.255

Class D

28 bits

1	1	1	0	multicast address

224.0.0.0 to 239.255.255.255

- Up to 250 million multicast groups at the same time
- Permanent group addresses
  - All systems in LAN; All routers in LAN;
  - All OSPF routers on LAN; All designated OSPF routers on a LAN, etc.
- Temporary groups addresses created as needed
- Special multicast routers

Reserved Host IDs (all 0s & 1s)

Internet address used to refer to network has hostid set to all 0s

0	0	0	0		0	0	this host (used when booting up)

0	0		0	**host**	a host in this network

Broadcast address has hostid set to all 1s

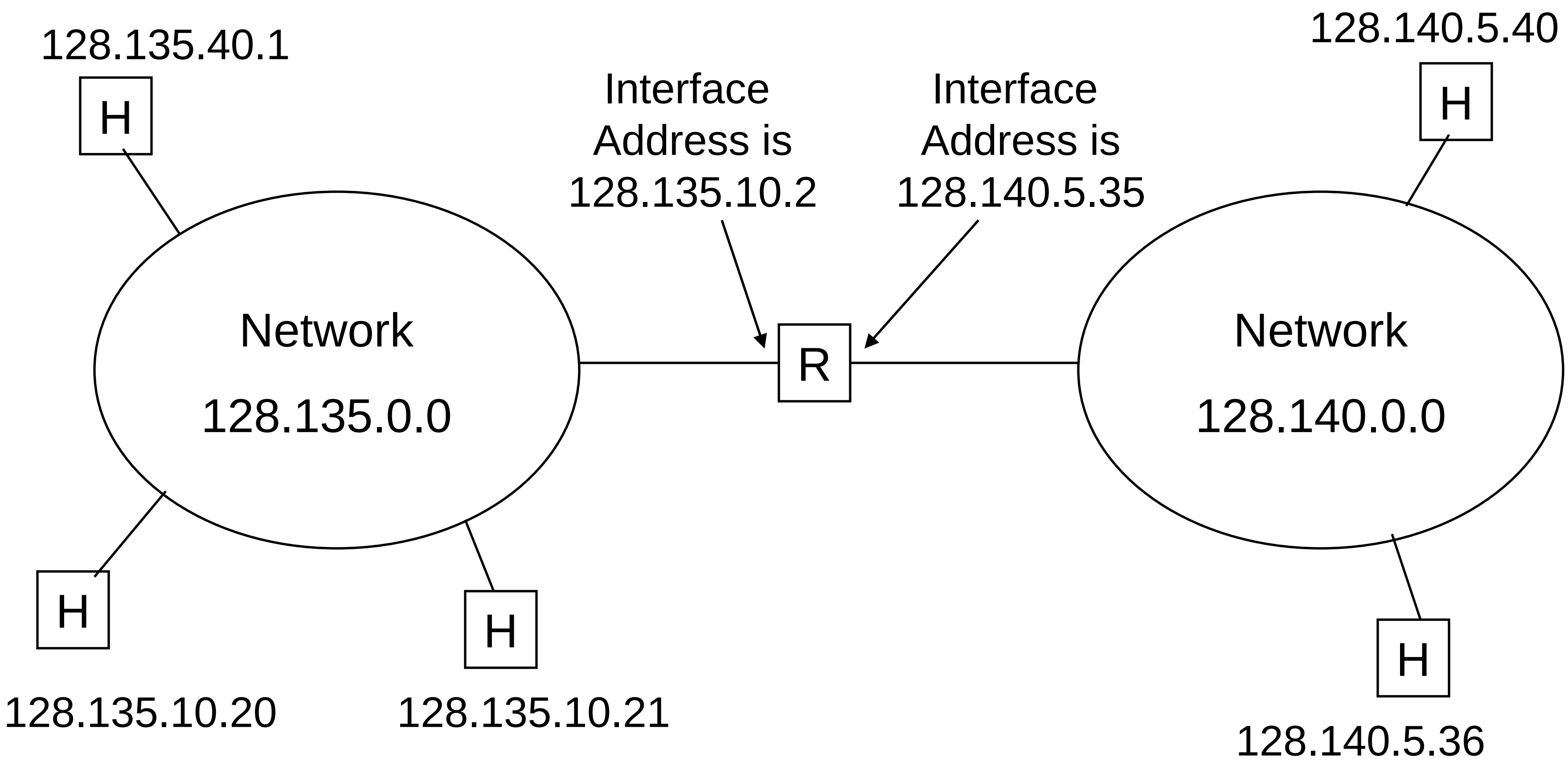
1	1	1	1		1	1	broadcast on local network

netid	1	1	1		1	1	1	1	broadcast on distant network

Private IP Addresses

- Specific ranges of IP addresses set aside for use in private networks (RFC 1918)
- Use restricted to private internets; routers in public Internet discard packets with these addresses
- Range 1: 10.0.0.0 to 10.255.255.255
- Range 2: 172.16.0.0 to 172.31.255.255
- Range 3: 192.168.0.0 to 192.168.255.255
- Network Address Translation (NAT) used to convert between private & global IP addresses

Example of IP Addressing



Address with host ID=all 0s refers to the network

Address with host ID=all 1s refers to a broadcast packet

R = router

H = host

Subnet Addressing

- Subnet addressing introduces another hierarchical level
- Transparent to remote networks
- Simplifies management of multiplicity of LANs
- Masking used to find subnet number

Original address	1	0	Net ID	Host ID

Subnetted address	1	0	Net ID	Subnet ID	Host ID

Subnetting Example

- Organization has Class B address (16 host ID bits) with network ID: 150.100.0.0
- Create subnets with up to 100 hosts each
  - 7 bits sufficient for each subnet
  - 16-7=9 bits for subnet ID
- Apply subnet mask to IP addresses to find corresponding subnet
  - Example: Find subnet for 150.100.12.176
  - IP add = 10010110 01100100 00001100 10110000
  - Mask = 11111111 11111111 11111111 10000000
  - AND = 10010110 01100100 00001100 10000000
  - Subnet = 150.100.12.128
  - Subnet address used by routers within organization

Subnet Example