Network Layer:
Network Layer and IP Protocol

Required reading: Garcia 7.3.3, 8.1, 8.2.1

CSE 3213, Winter 2010
Instructor: N. Vlajic

1. **Introduction**
2. Router Architecture
3. Network Layer Protocols in the Internet
4. IPv4
5. IP Addressing and Subnetting

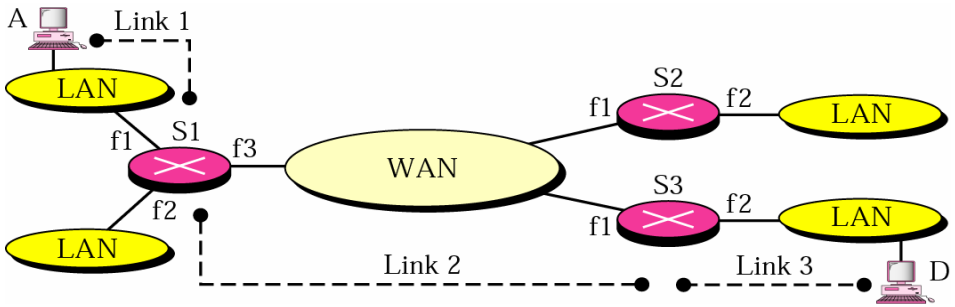
Introduction

Network Layer – supervises host-to-host packet delivery – hosts could be separated by several physical networks

- data-link layer provides node-to-node delivery, transport layer provides process-to-process delivery

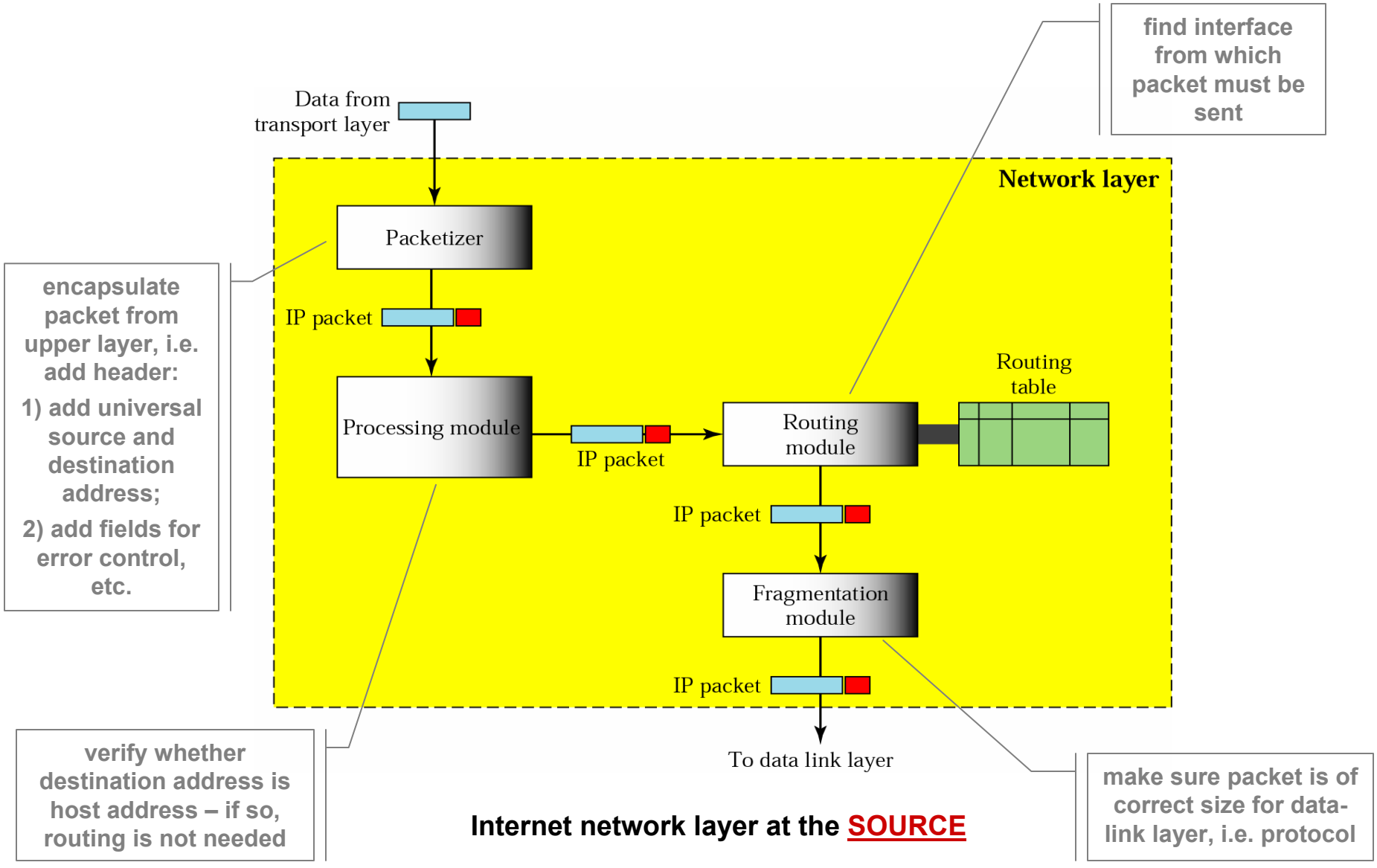
Major (Basic) Network Layer Duties

- **addressing**: identify each device uniquely to allow global communication
- **routing**: determine optimal route for sending a packet from one host to another
- **packetizing**: encapsulate packets received from upper-layer protocols
- **fragmenting**: decapsulate packets from one and encapsulate them for another network



Introduction (cont.)

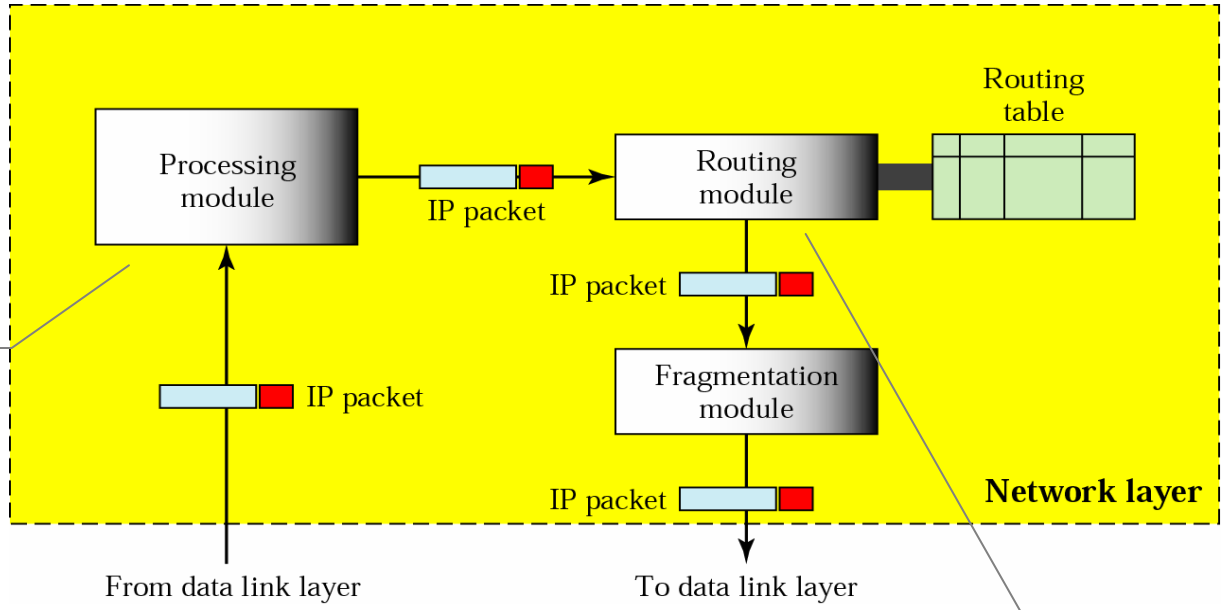
Example **[network layer duties in the Internet, at the SOURCE]**



Internet network layer at the SOURCE

Introduction (cont.)

Example cont. [network layer duties in the Internet, at a ROUTER]



check if packet has reached its final destination or needs to be forwarded (TTL!)

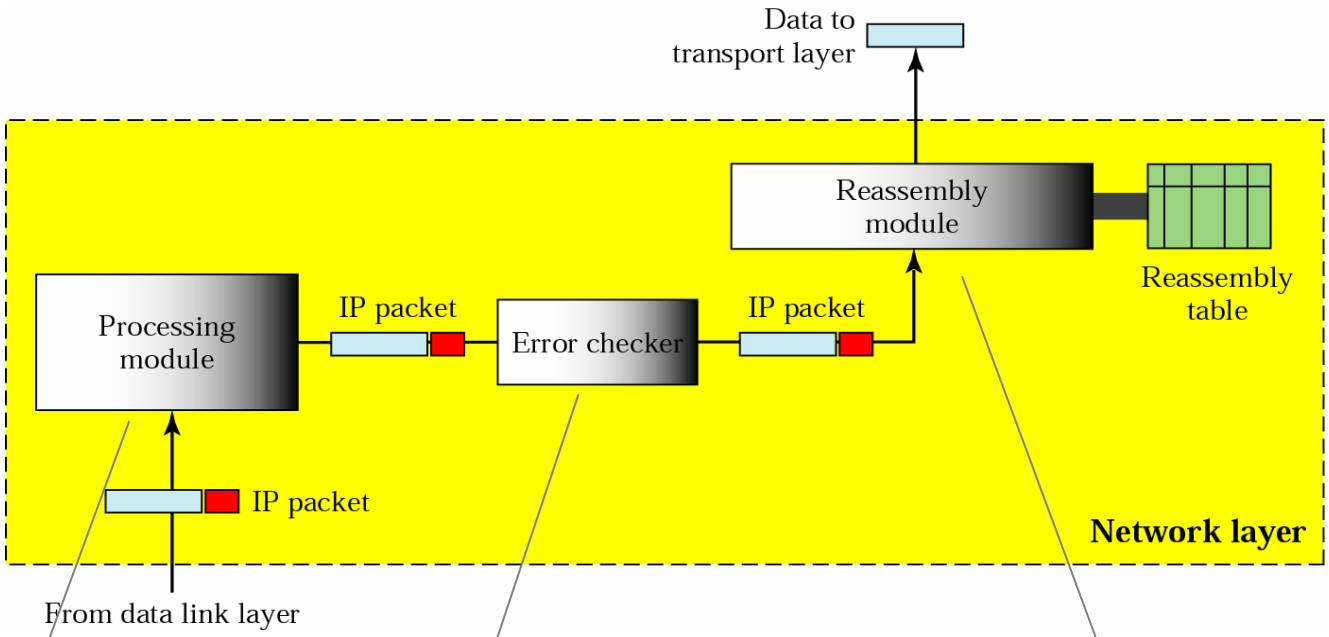
+ header error checking !!!

find interface from which packet must be sent

Internet network layer at a ROUTER

Introduction (cont.)

Example cont. **[network layer duties in the Internet, at the DESTINATION]**



verify whether destination address is host address

Internet network layer at the DESTINATION

check if packet has been corrupted during transmission

if packet has been fragmented, wait until all fragments have arrived, reassemble them, and then deliver the reassembled packet to transport layer

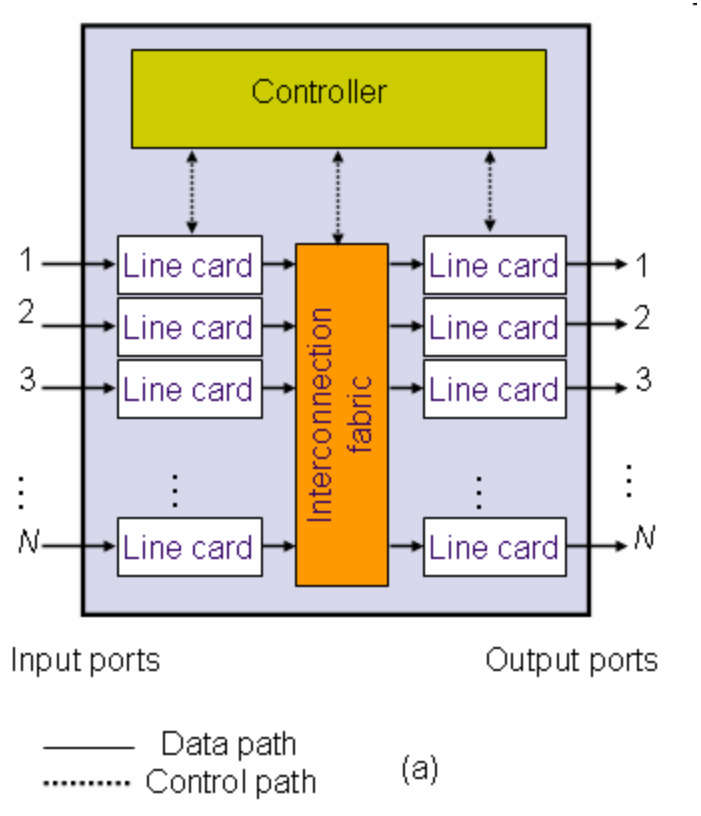
1. Introduction
2. **Router Architecture**
3. Network Layer Protocols in the Internet
4. IPv4
5. IP Addressing and Subnetting

Internet Router Architecture

Router – **3-layer (physical, data-link, network) device, with 3 key functions:**

- **run routing algorithms/protocols (RIP, OSPF, BGP)**
- **forward/switch IP packets from incoming to proper outgoing links**
- **manage congestion**

Router Architecture



- **input ports / interfaces** (see pp. 10)
- **interconnection (switching) fabric** (see pp. 11)
- **output ports / interfaces** (see pp. 12)
- **routing processor (switch controller)** – **general-purpose processor in charge of**
  1) **executing routing protocol**
  2) **maintaining routing information and forwarding tables, etc.**

Example [forwarding / routing table]

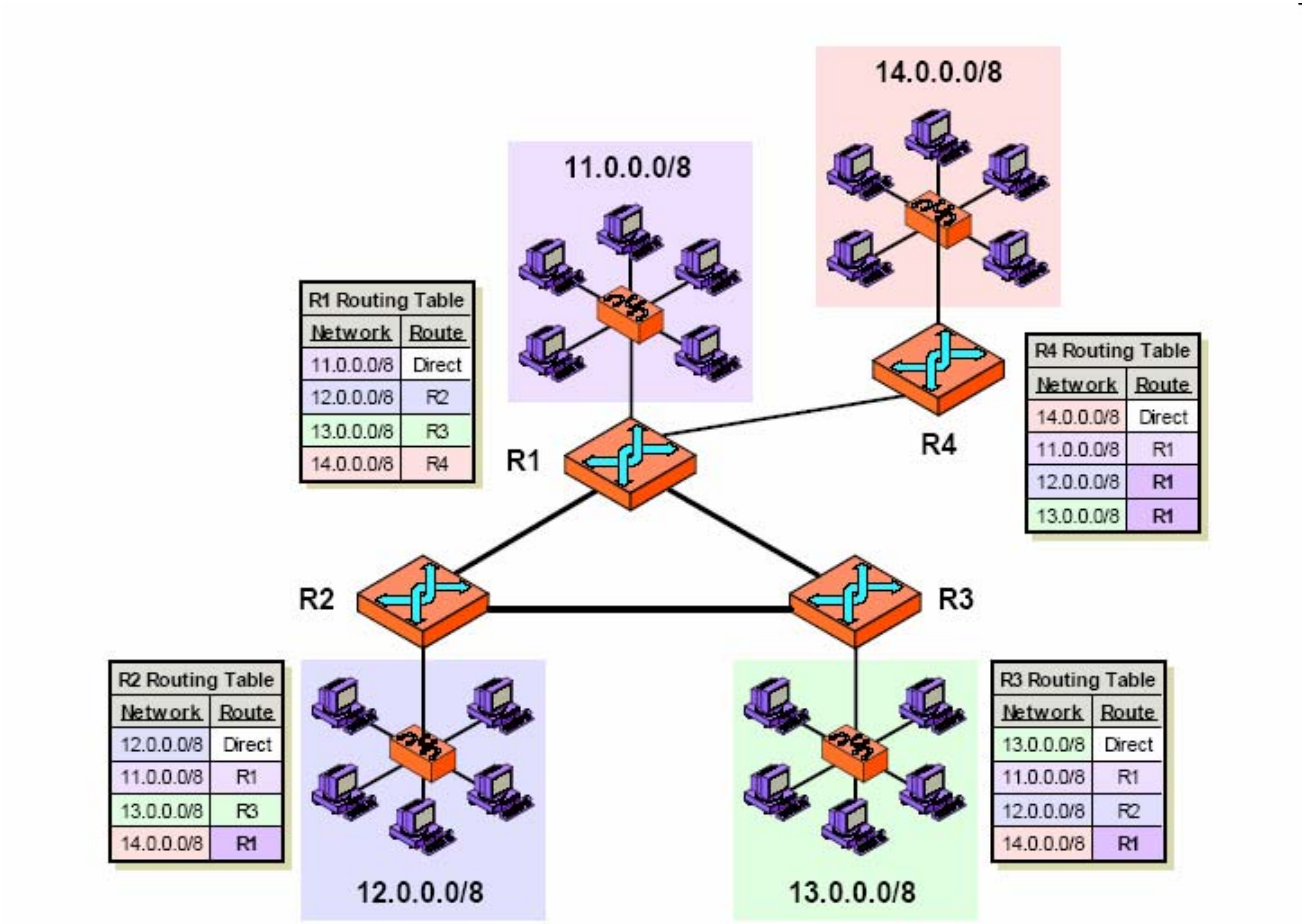


Figure 93: IP Routing and Routing Tables

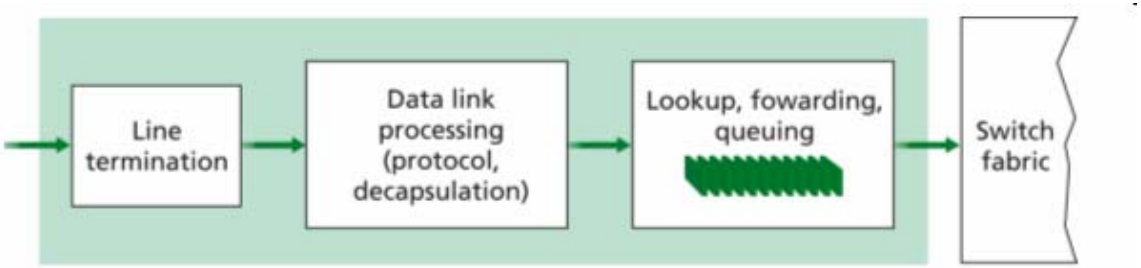
This diagram shows a small, simple internetwork consisting of four LANs each served by a router. The routing table for each lists the router to which datagrams for each destination network should be sent, and is color coded to match the colors of the networks. Notice that due to the "triangle", each of R1, R2 and R3 can send to each other. However, R2 and R3 must send through R1 to deliver to R4, and R4 must use R1 to reach either of the others.

Internet Router Architecture (cont.)

Input Port – **has an associated line card (NIC) which implements physical and data-link layer functions, as well as certain network layer functions**

Input Line Card Functions

- **physical layer: bit-level reception**
- **data-link layer: decapsulation, error checking, etc.**
- **network layer: decentralized switching / packet forwarding = decide to which output line to forward each packet based on packet header**
  - **looks up output port using forwarding table in input line card memory** (table is created and updated by routing processor)

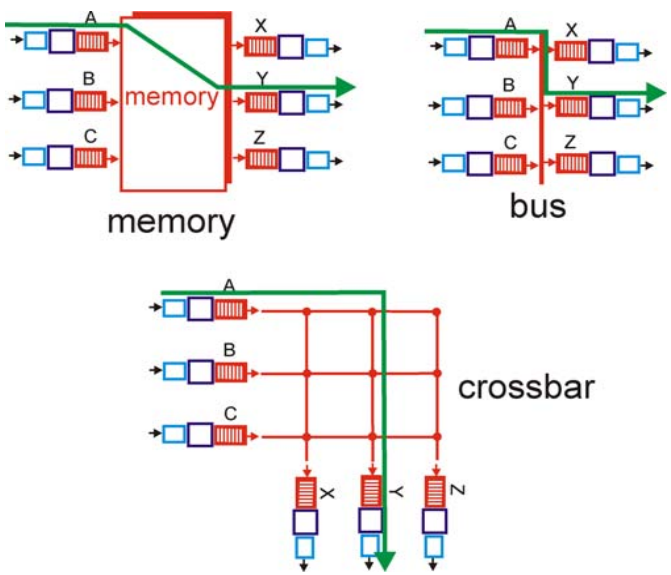


Decentralized switching prevents creating a processing bottleneck at a single point within the router.

Internet Router Architecture (cont.)

Switching Fabric Function – **(physically) transfer packets between input and output line cards**

Types of Switching Fabric



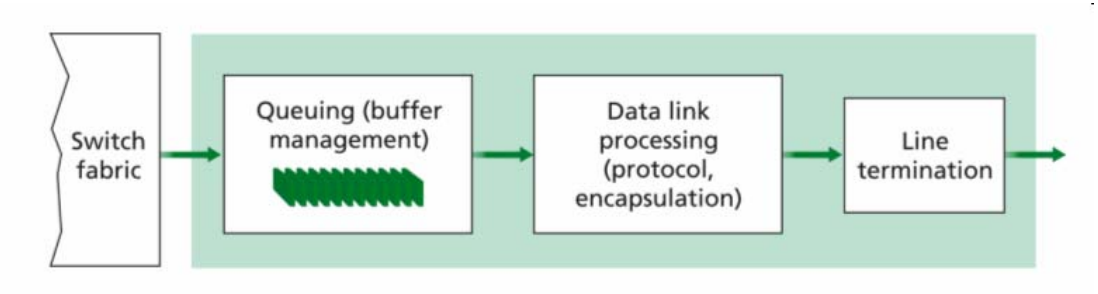
- **via memory**: **datagram is received through input port, stored in memory, then send to appropriate output port – slow ☹**
- **via a bus**: **datagram is sent directly from input port to output port via a shared bus ⇒ does not scale well ☹**

  (packets are send serially so buss speed needs to be N-times input line speed)
  - **today's bus bandwidths ≥ 1 Gbps ⇒ switching via bus is sufficient for routers in LANs**
- **via a crossbar**: **interconnection network consisting of 2N busses that interconnect N input and N output**
  - **packet travels along horizontal bus until it intersects with vertical bus leading to desired output port – if vertical bus is busy, queueing at input port is needed**
  - **Cisco 12000 Family – 60 Gbps routers**

Internet Router Architecture (cont.)

Output Line Card Functions

- network layer:
  1) **buffering** – required when datagrams arrive from fabric at rate faster than output line transmission rate
  2) **buffer management** – decide when and which packets to drop if there is not enough memory to store all incoming packets
  3) **scheduling / packet classification** – decide which packet, of those queued, to send out next
     - packet scheduling plays crucial role in in providing quality-of-service (QoS)
- data-link layer: encapsulation, address mapping, etc.
- physical layer: bit-level forwarding

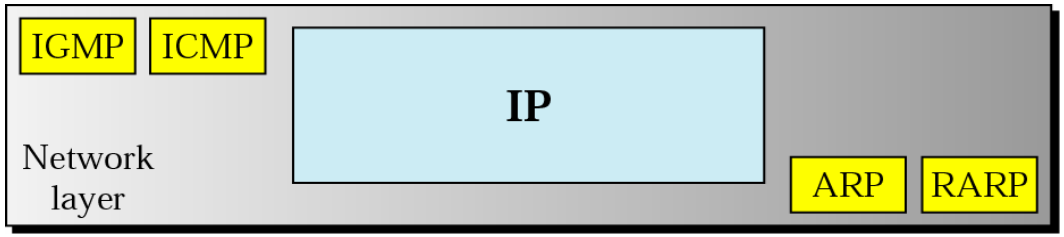


1. Introduction
2. Router Architecture
3. **Network Layer Protocols in the Internet**
4. IPv4
5. IP Addressing and Subnetting

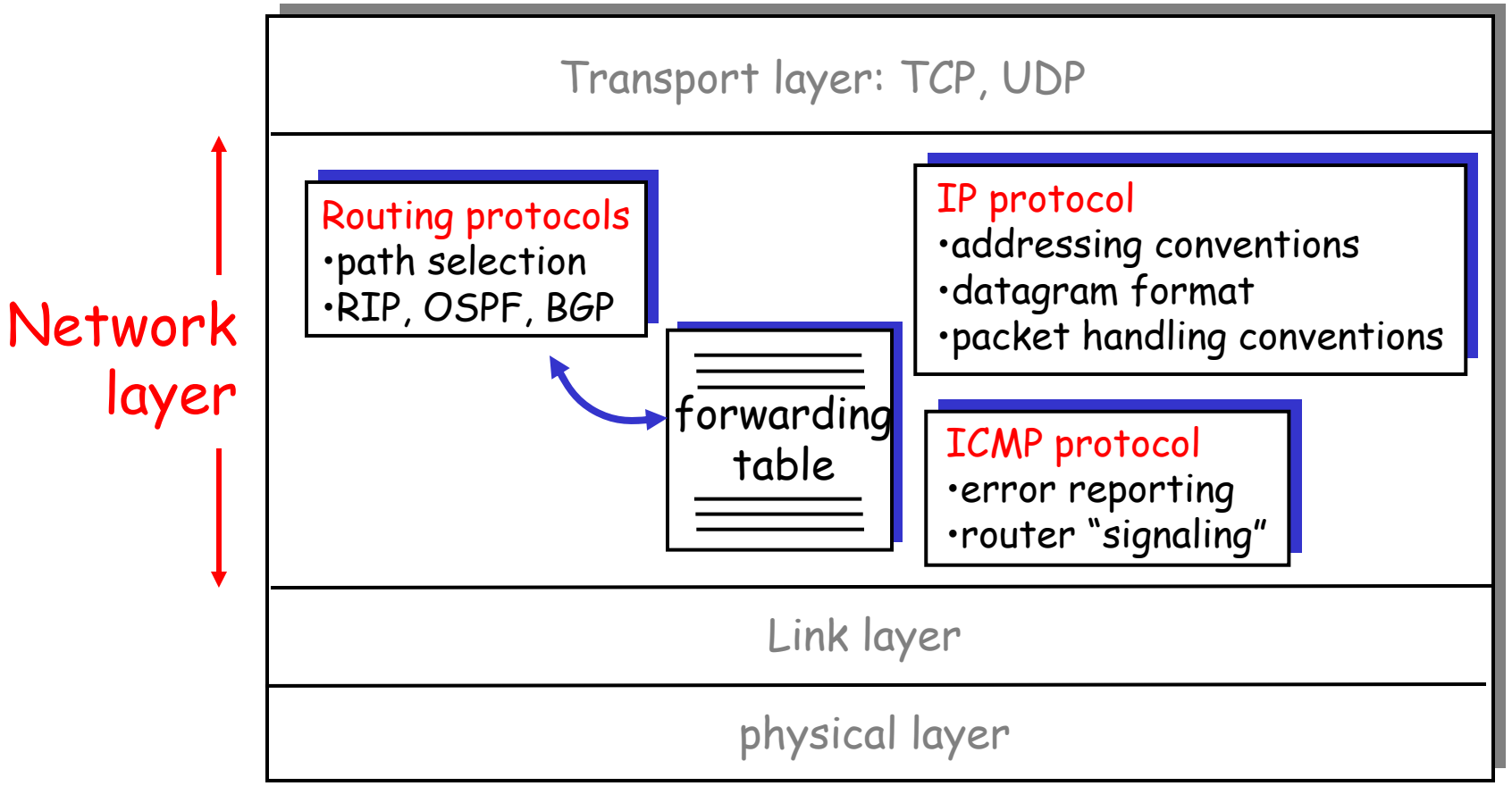
Internet Network Layer Protocols

Network Layer Protocols in the Internet

- **IP** – main protocol, responsible for 'best effort' host-to-host delivery
- **ARP** – maps IP address of next hop to its MAC/physical address (used when passing packets to lower data-link layer)
- **RARP** – maps MAC/physical address to IP address (used at diskless machines for IP address recovery)
- **ICMP** – used by hosts and routers to handle unusual situations such as IP packet-header errors, unreachable hosts and networks, etc.
- **IGMP** – used by host and routers to achieve efficient network-layer multicasting
- **Routing Protocols** – responsible for routing table maintenance



Internet Network Layer Protocols (cont.)



1. Introduction
2. Router Architecture
3. Network Layer Protocols in the Internet
4. **IPv4**
5. IP Addressing and Subnetting

IP Protocol

Internet Protocol (IP) – host-to-host network-layer delivery protocol for the Internet with following properties

- connectionless service – each packet is handled independently (possibly along different path)
- best-effort delivery service
  1) does its best to deliver packet to its destination, but with no guarantees
  2) limited error control – only error detection, corrupted packets are discarded
  3) no flow control
- must be paired with a reliable transport- (TCP) and/or application- layer protocol to ensure reliability

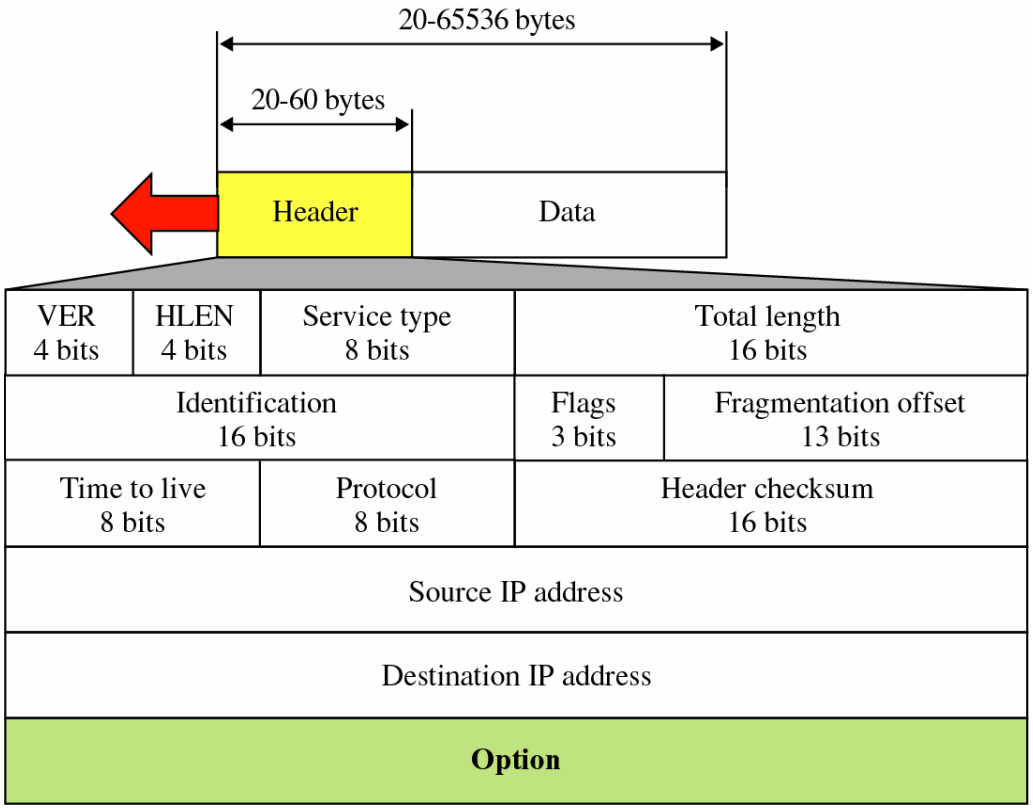
IP Protocol Versions

- IPv4 – version currently in wide use (1981)
- IPv6 – new version of IP protocol created to correct some of significant problems of IPv4 such as exhaustion of address space (1996)
- Mobile IP – enhanced version of IPv4 – supports IP in mobile environments (1996)

IP Datagram Fields

Datagram – **IP packet = variable length packet consisting of header & data**

- **header – 20 to 60 bytes in length, contains information essential to routing and delivery**
- **data – length determined by Maximum Transmission Unit (MTU) of link layer protocol** (theoretically between 20 to 65536 bytes)



IP Datagram Fields (cont.)

Version Number – 4-bit field – specifies IP protocol version of the datagram (IPv4 or IPv6)

- different version of IP use different datagram formats
- by looking at version number router can determine how to interpret remainder of datagram

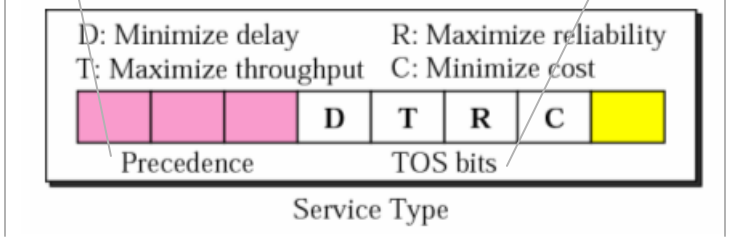
Header Length – 4-bit field – defines total length of datagram header in 4-byte words

- when there are no options header length is 20 ⇒ HLEN = 5

Differentiated Service (formerly Service Type) – 8-bit field – allows different types of datagrams to be distinguished from each other based on their associated / requested QoS

- e.g. datagrams particularly requiring low delay, high throughput, or reliability

Precedence defines the priority of datagram in case of congestion. If a router is congested and needs to discard some datagrams, those datagrams with lowest precedence are discarded first. **Network management datagrams have the highest precedence!**

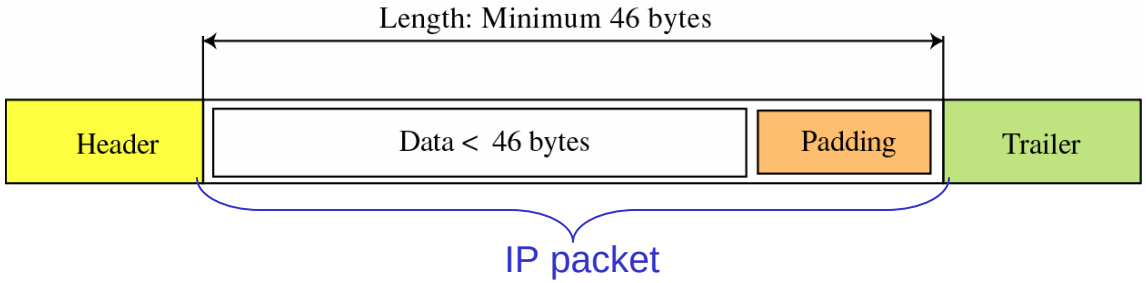


Although each TOS bit has a special meaning, only one bit can be set to 1 in each datagram.
0000 – normal type of service
0001 – minimize cost
0010 – maximize reliability
0100 – maximize throughput
1000 – minimize delay

IP Datagram Fields (cont.)

Total Length – 16-bit field – defines total datagram length in bytes, including header

- 16 bits ⇒ maximum size = 65,535 bytes
- some physical networks are not able to encapsulate a datagram of 65,535 bytes, so datagram must be fragmented to be able to pass through those networks
- some physical networks have restriction on minimum size of data that can be encapsulated in a frame, so datagram must be padded (e.g. Ethernet min size of data – 46 bytes)



Identifier, Flags, Fragmentation Offset – 3 fields used in fragmentation

- IPv6 does not allow fragmentation at routers since it is time consuming operation – if an IPv6 packet is too big, it is simply dropped and an ICMP message is sent back to the source

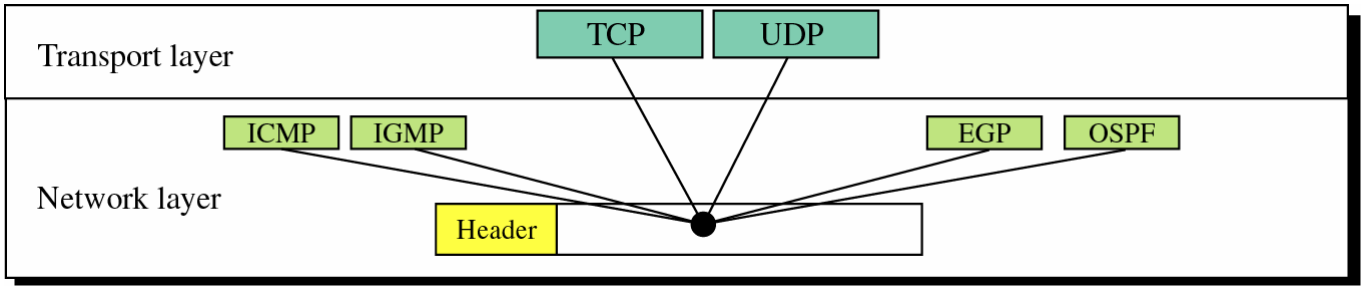
IP Datagram Fields (cont.)

Time-To-Live (TTL) – **8-bit field – controls max number of hops visited by datagram and/or time spend in the network**

- **field is decremented by one each time datagram is processed by a router – when TTL reaches 0, datagram must be dropped**
- **ensures that**
  1) **datagram does not circulate/loop forever, or**
  2) **to limit its journey** (e.g. LAN only: TTL = 1)

Protocol – **8-bit field – indicates specific transport-layer protocol to which data portion of this IP datagram should be passed**

- **used only at final destination to facilitate demultiplexing process**
- protocol number is glue that binds network & transport layer, while port number is glue that binds transport & application layer
- **values: 1 – ICMP, 2 – IGMP, 6 – TCP, 17 – UDP, 89 – OSPF**



IP Datagram Fields (cont.)

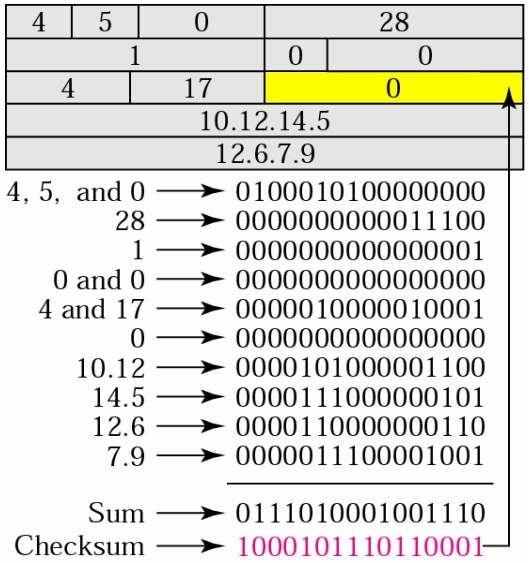
Header Checksum – **16-bit field – aids in detecting errors in header only!**

- **checksum must be recomputed & stored again at each router as TTL and some options fields may change**
- **routers discard datagrams for which an error is detected**
- **checksum calculation:**
  1) **divide header into 16-bit (2-byte) sections – checksum field itself is set to 0**
  2) **sum all sections using 1s complement arithmetic**

Each intermediate router must:

1) verify / recompute checksum on every incoming packet

2) compute checksum for every outgoing packet



Error detection / correction is not the responsibility of network-layer.
Why is, then, IP willing to perform error detection on IP headers?!

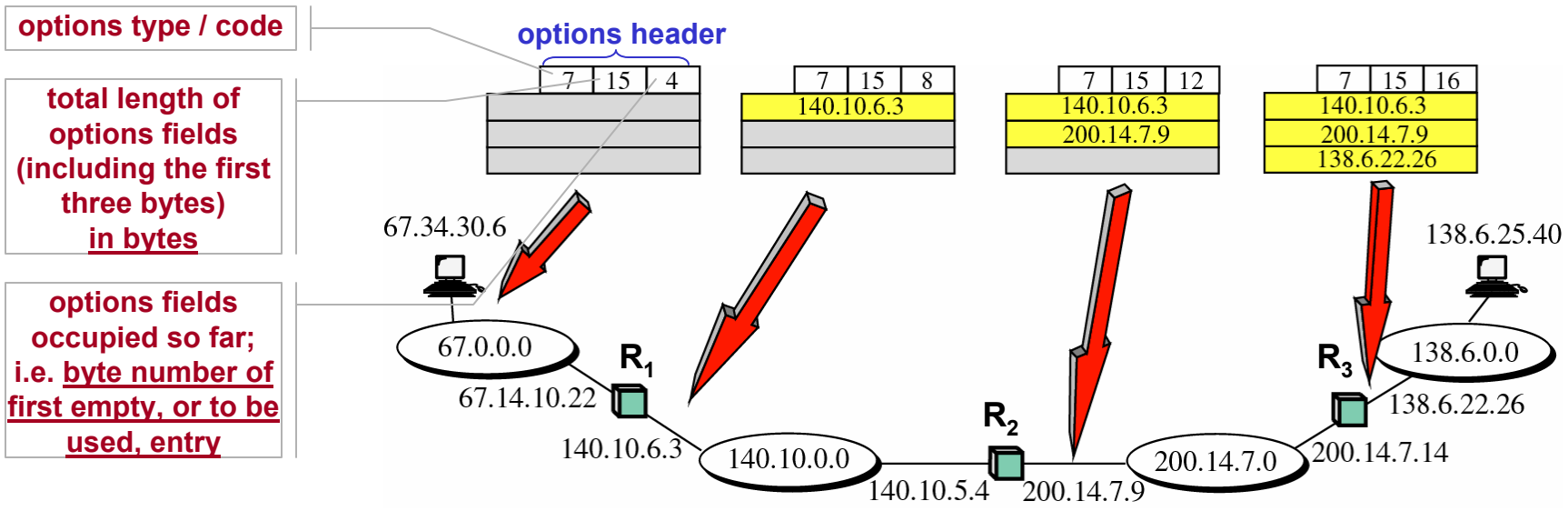
IP Datagram Fields (cont.)

Source and Destination IP Addresses – **32-bit fields – must remain unchanged until IP datagram reaches its final destination**

Options – **32-bit field(s) – not required for every datagram! – allows expansion of IP header for special purposes**

(a) **Record Route option** – **used to trace route that datagram takes**

- **source creates empty fields for IP addresses – up to 9**
  (40 bytes options – 4 bytes option header) / 4 bytes for IP address
- **each router that processes datagram inserts its outgoing IP address**



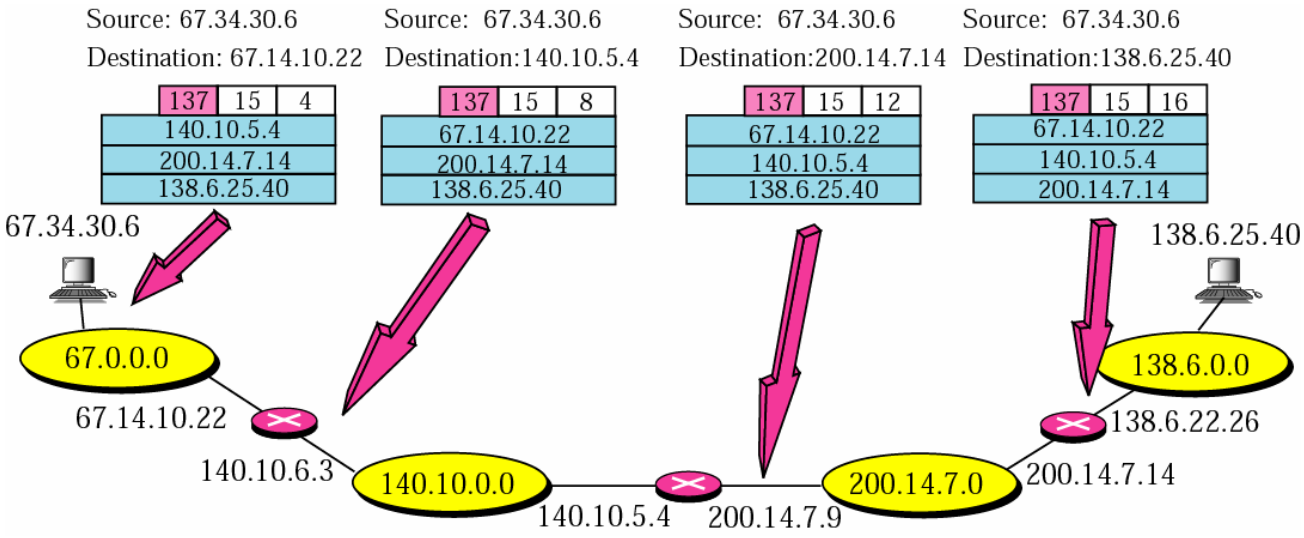
IP Datagram Fields (cont.)

Options (cont.)

(b) **Timestamp option** – similar to (a), plus records datagram end-processing time by each router, in milliseconds

(c) **Strict Source Route option** – used by source to predetermine route for datagram

- source provides a list of IP addresses (sequence of routers) that datagram must (is allowed) to visit on its way to destination



(d) **Loose Source Route option** – similar to (c), but it is more relaxed – each router in the list must be visited, though datagram can visit other routers as well

IP Datagram Fields (cont.)

Example [IP Datagram fields]

An IP packet has arrived with the first 8 bits as shown: 01000010
The receiver discards the packet. Why?

Solution:

There is an error in this packet. The 4 left-most bits (**0100**) show the version, which is correct. The next 4 bits (**0010**) show the header length, which means ($2 \times 4 = 8$), which is wrong. The minimum number of bytes in the header must be 20. The packet has been corrupted in transmission.

Example [IP Datagram fields]

In an IP packet, the value of HLEN is 1000 in binary. How many bytes of options are being carried by this packet?

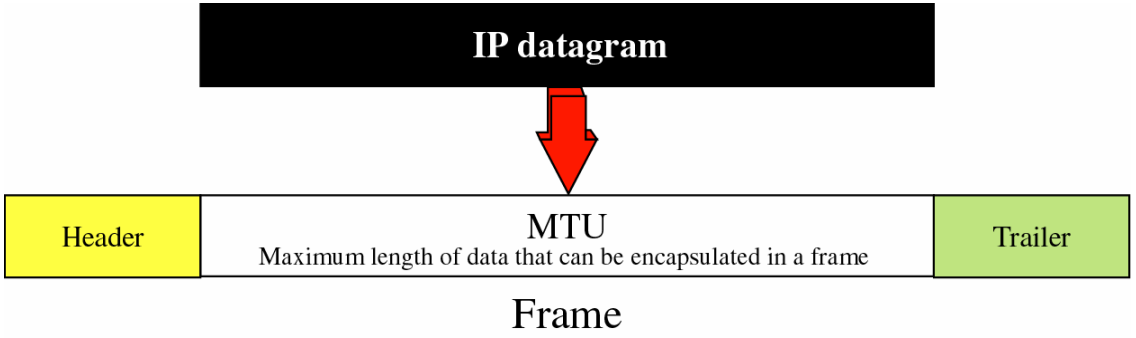
Solution:

The HLEN value is 8, which means the total number of bytes in the header is 8x4 or 32 bytes. The first 20 bytes are the main header, the next 12 bytes are the options.

IP Datagram Fragmentation

Maximum Transfer Unit (MTU) – **maximum amount of data that link-layer frame can carry = hard limit on IP datagram length**

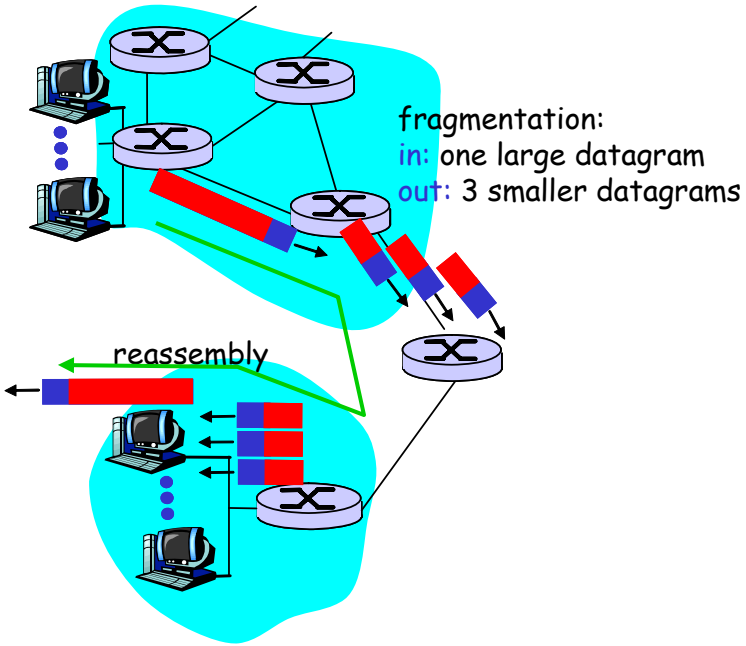
- **MTU differs from one data-link layer protocol to another**
  - **(a) Token Ring (4 Mbps): MTU = 4,464 bytes**
  - **(b) Ethernet: MTU = 1,500 bytes**
  - **(c) PPP: MTU = 296 bytes**



Hard limit on IP datagram size is not a problem.
What is a problem is that each of the links along the route between sender and receiver can use different link-layer protocols, and each of these protocols can have different MTUs.

IP Datagram Fragmentation (cont.)

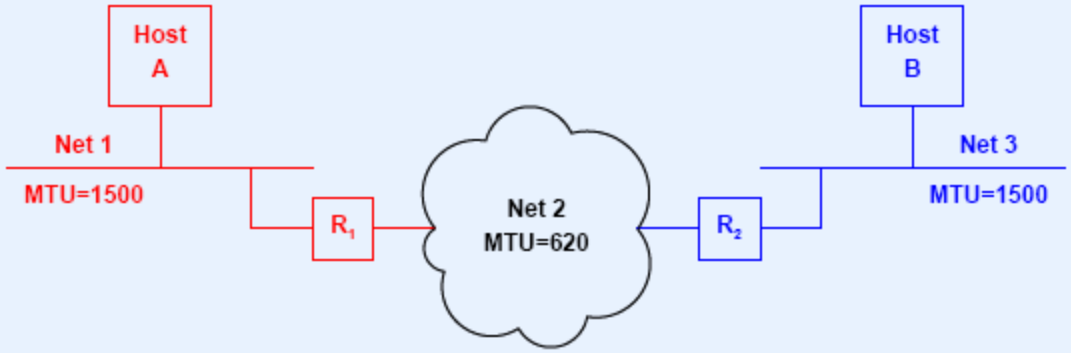
IP Datagram Fragmentation – **process of dividing datagram into smaller fragments that meet MTU requirements of underlying data-link layer protocol**



- **datagram can be fragmented by source-host or any other router in the path; however reassembly of datagram is done only by destination host! – parts of a fragmented datagram may take different routes !!!**
- **once fragmented datagram may be further fragmented if it encounters network with even smaller MTU**
- **when a datagram is fragmented, each fragment gets its own header with most fields repeated, but some changed**
  - **host or router that fragments datagram must change values of three fields: *flags*, *fragmentation offset* and *total length***

IP Datagram Fragmentation (cont.)

Example **[Example, from the book by D. E. Comer]**



- Hosts A and B send datagrams of up to 1500 octets
- Router R_1 fragments large datagrams from Host A before sending over Net 2
- Router R_2 fragments large datagrams from Host B before sending over Net 2

IP Datagram Fragmentation (cont.)

Identification – 16-bit field – uniquely identifies datagram originating from source host

- to guarantee uniqueness, IP uses counter to label each datagram
- when IP sends a datagram, it copies current counter value to identification field, and increments counter by one
- when datagram is fragmented, identification field is copied into all fragments
- identification number helps destination in reassembling datagram – all fragments with same identification value should be assembled into one datagram

Flags – 3-bit field

- 1st bit is reserved
- 2nd bit is called "do not fragment" bit
  - if its value is 1, machine must NOT fragment datagram
  - if fragment cannot pass through physical network router discards packet and sends ICMP error message back to source host
- 3rd bit is called "more fragment" bit
  - if its value is 1, datagram is not last fragment – there are more fragments after this one
  - if its value is 0, this is last or only fragment

D: Do not fragment
M: More fragments

IP Datagram Fragmentation (cont.)

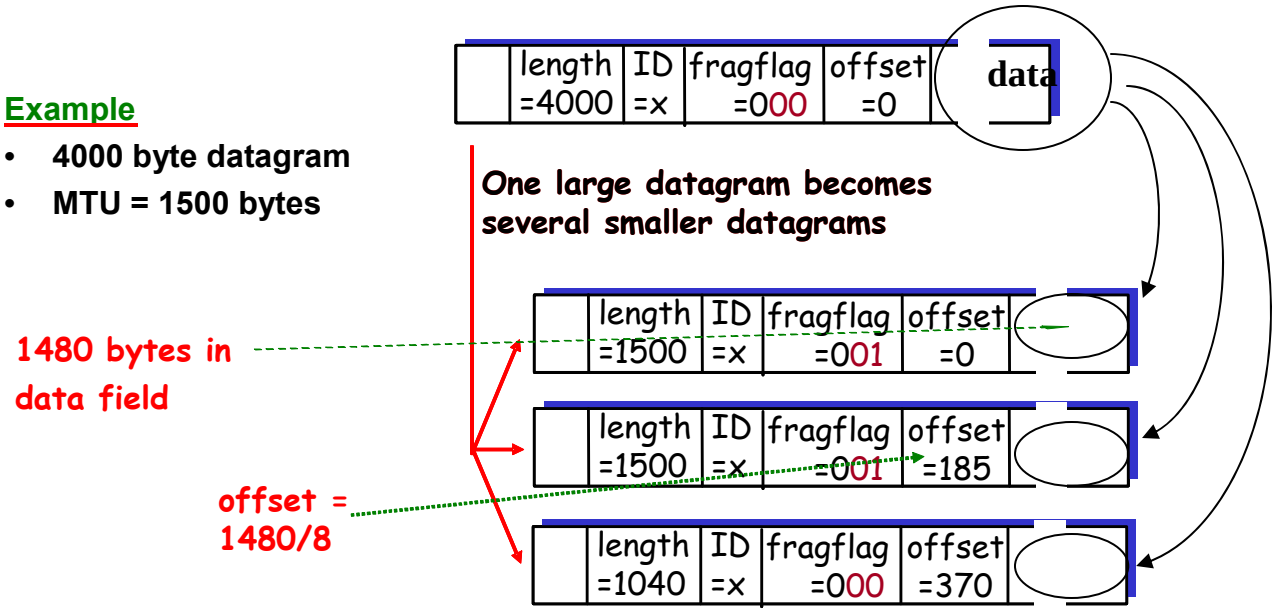
Fragmentation Offset – **13-bit field – shows relative position of fragment's data with respect to whole datagram**

- **the offset is measured in units of 8 bytes – this is done because offset field is only 13 bits long and otherwise could not represent sequences greater than 8191**
- **this forces hosts and routers to choose fragment sizes divisible by 8**

Example [fragmentation]

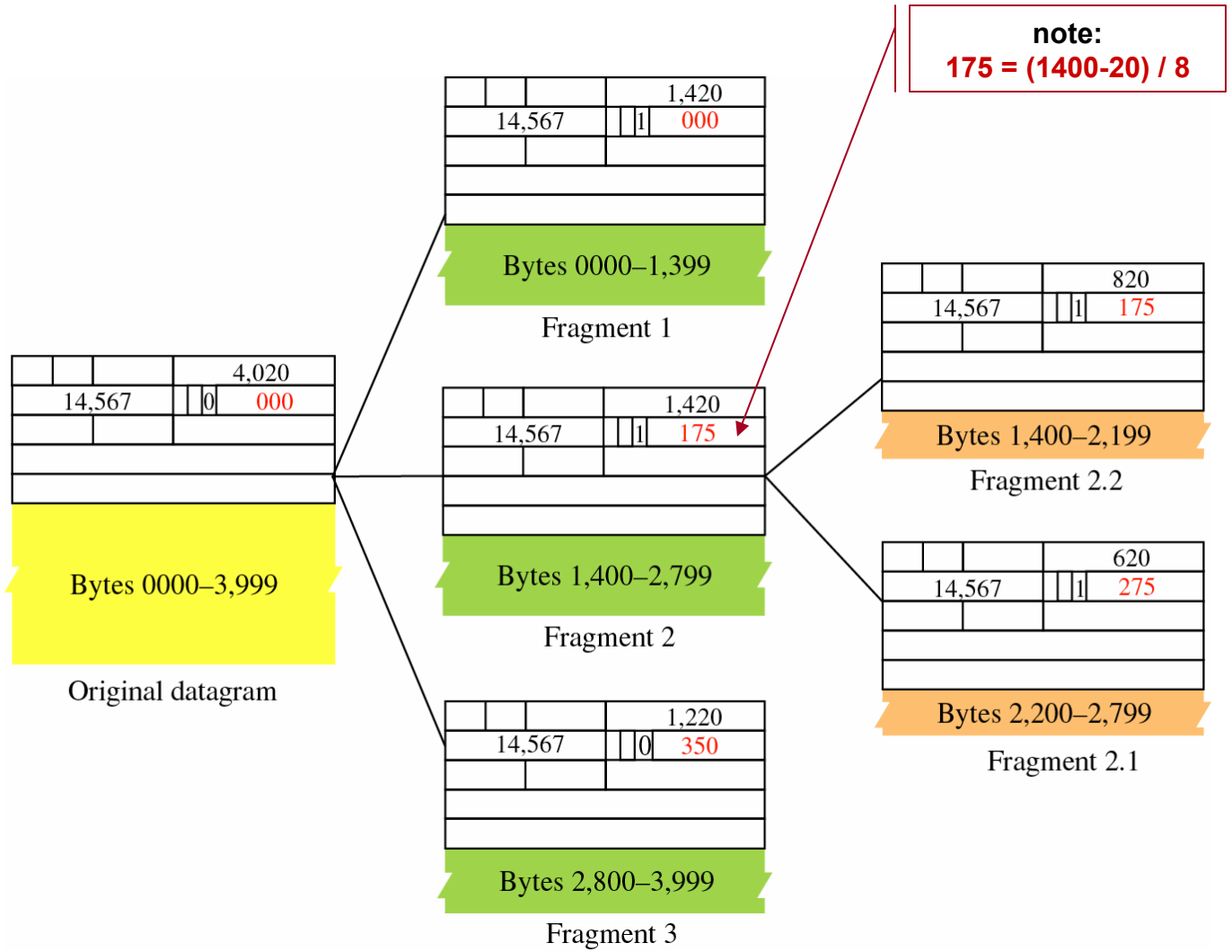
Example

- **4000 byte datagram**
- **MTU = 1500 bytes**

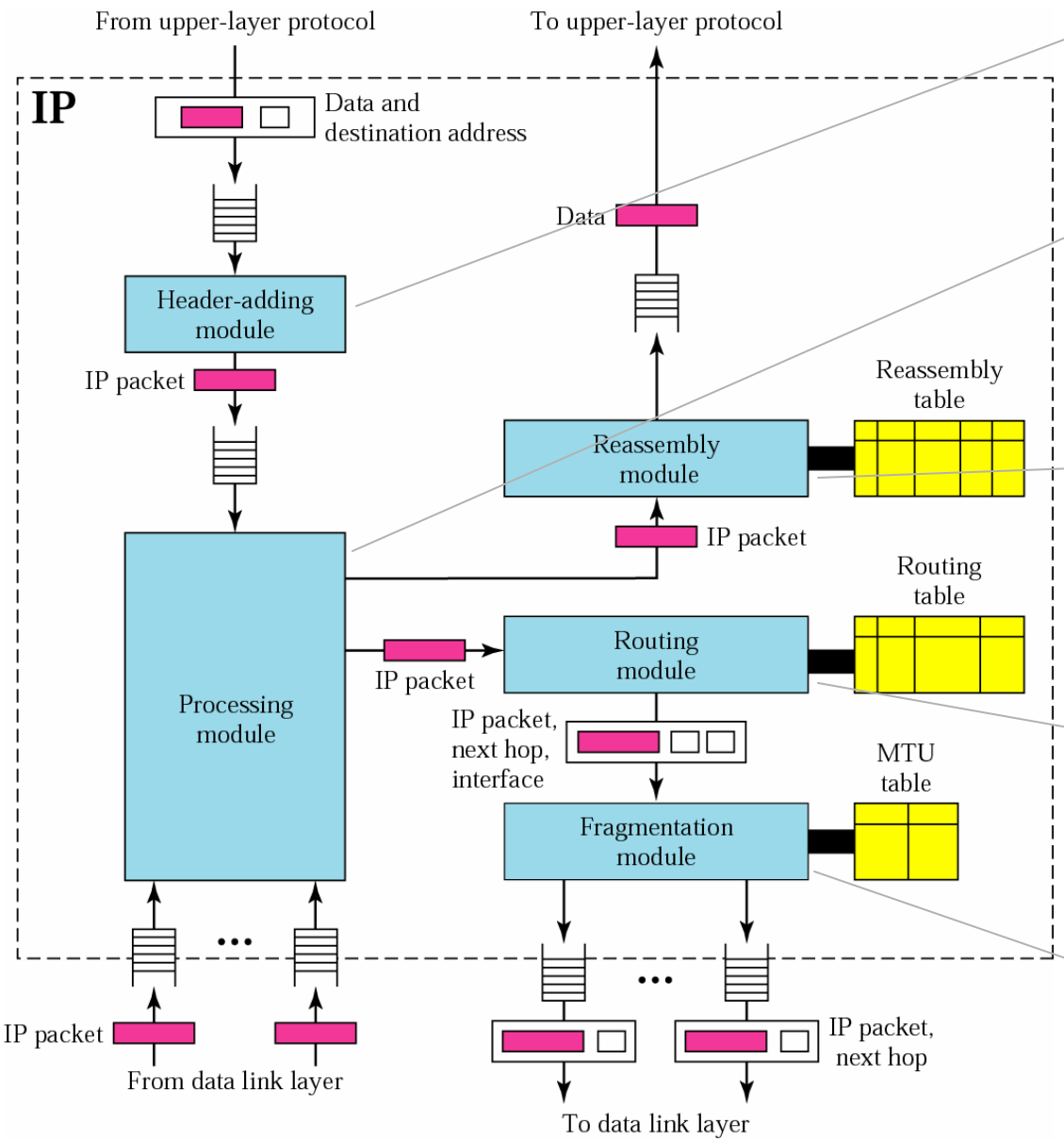


IP Datagram Fragmentation (cont.)

Example [fragmentation of a fragment]



IP Datagram Processing



Header-adding module
encapsulates the data from the transport-layer protocol by adding the IP header

Processing module
1) checks if destination IP address = host IP address
2) decrements TTL by 1
(done only at router-nodes)

Reassembly module
orders and reassembles the fragments belonging to the same datagram. Reassembly table keeps track of received fragments

Routing module
finds the IP address of the next station along with the interface number to which the packet should be sent.

Fragmentation module
consults MTU table to find the MUT for the specific interface number.

Exercise

1. **A packet has arrived with Flag's M bit value of 0. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?**

2. **A packet has arrived with an *M* bit value of 1. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?**

3. **A packet has arrived with an *M* bit value of 1 and a fragmentation offset value of zero. Is this the first fragment, the last fragment, or a middle fragment?**

4. **A packet has arrived in which the offset value is 100. What is the number of the first byte? Do we know the number of the last byte?**

5. **A packet has arrived in which the offset value is 100, the value of HLEN is 5 and the value of the total length field is 100. What is the number of the first byte and the last byte?**