COMPUTER NETWORKS

18CS46

UNIT 3

Congestion Control Algorithms

Dr. Minal Moharir

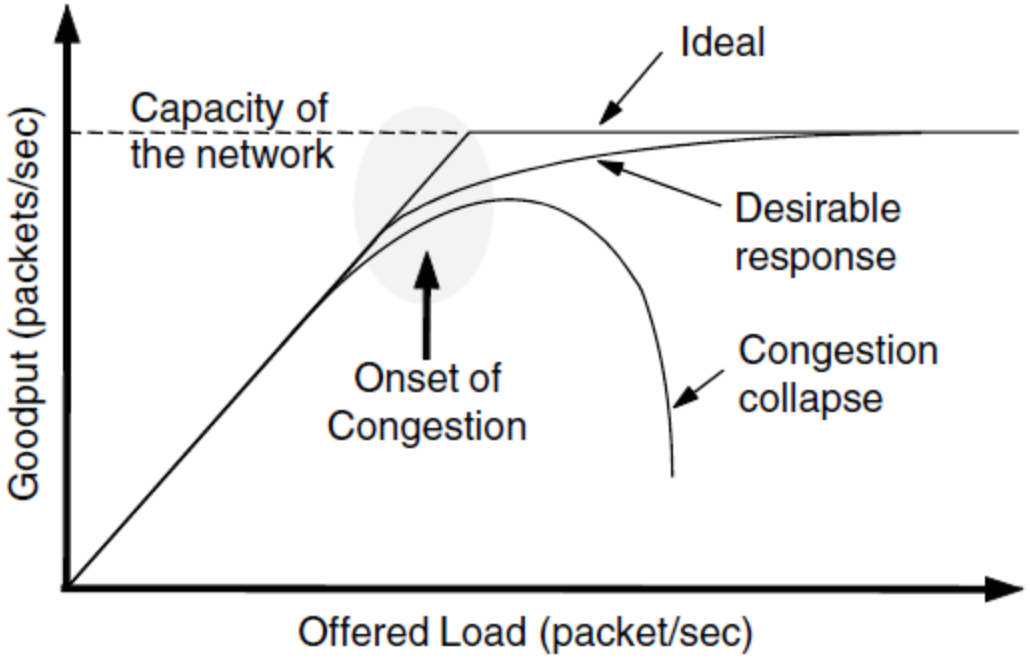
Congestion Control Algorithms

- Too many packets present in (a part of) the network causes packet delay and loss that degrades performance.

- This situation is called **congestion**.

- The network and transport layers share the responsibility for handling congestion.

Congestion Control Algorithms



When too much traffic is offered, congestion sets in and performance degrades sharply.

Congestion Control Algorithms

- When the number of packets hosts send into the network is well within its carrying capacity, the number delivered is proportional to the number sent.
- the offered load approaches the carrying capacity, bursts of traffic occasionally fill up the buffers inside routers and some packets are lost.
- These lost packets consume some of the capacity, so the number of delivered packets falls below the ideal curve.
- The network is now congested.

Congestion Control Algorithms

- Unless the network is well designed, it may experience a **congestion collapse**
- different failure mode occurs when senders retransmit packets that are greatly delayed, thinking that they have been lost.
- In this case, copies of the same packet will be delivered by the network, again wasting its capacity.
- To capture these factors, the y-axis of
- Fig is given as **goodput**, which is the rate at which *useful* packets are delivered by the network.

Congestion Control Algorithms

- We should design networks that avoid congestion where possible and do not suffer from congestion collapse
- Unfortunately, congestion cannot wholly be avoided.

Congestion Control Algorithms

- If all of a sudden, streams of packets begin arriving on three or four input lines and all need the same output line, a
- queue will build up.
- If there is insufficient memory to hold all of them, packets will be lost.
- Solution: Adding more memory
- Nagle (1987) realized that if routers have an infinite amount of memory, congestion gets worse, not better.
- This is because by the time packets get to the front of the queue, they have already timed out (repeatedly) and duplicates have been sent.
- This makes matters worse, not better—it leads to congestion collapse.

Congestion Control Algorithms

- Low-bandwidth links or routers that process packets more slowly than the line rate can also become congested.
- In this case, the situation can be improved by directing some of the traffic away from the bottleneck to other parts of the network.
- Eventually, however, all regions of the network will be congested.
- In this situation, there is no alternative but to shed load or build a faster network.

Congestion Control Algorithms

- Difference : congestion control and flow control
- Congestion control has to do with making sure the network is able to carry the offered traffic.
- It is a global issue, involving the behavior of all the hosts and routers.
- Flow control, in contrast, relates to the traffic between a particular sender and a particular receiver.
- Its job is to make sure that a fast sender cannot continually transmit data faster than the receiver is able to absorb it.

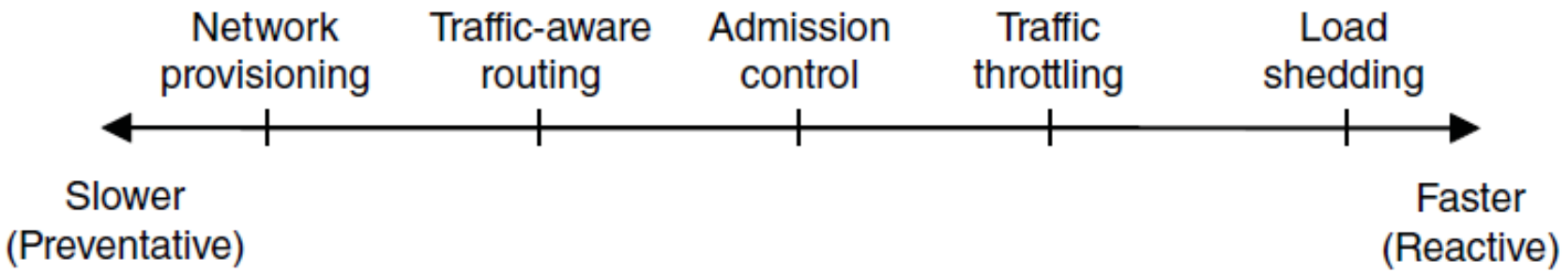
Congestion Control Algorithms

- 100-Gbps fiber optic links on which a supercomputer is trying to force feed a large file to a personal computer that is capable of handling only 1 Gbps.
- Although there is no congestion (the network itself is not in trouble), flow control is needed to force the supercomputer to stop frequently to give the personal computer a chance to breathe.
- At the other extreme, consider a network with 1-Mbps lines and 1000 large computers, half of which are trying to transfer files at 100 kbps to the other half.
- Here, the problem is not that of fast senders overpowering slow receivers, but that the total offered traffic exceeds what the network can handle.

Congestion Control Algorithms

- The reason congestion control and flow control are often confused is that the
- best way to handle both problems is to get the host to slow down.
- Thus, a host can get a ‘‘slow down’’ message either because the receiver cannot handle the load or because the network cannot handle it.

Approaches to Congestion Control



- The presence of congestion means that the load is (temporarily) greater than the resources (in a part of the network) can handle.
- Two solutions come to mind: increase the resources or decrease the load.

Approaches to Congestion Control

- The most basic way to avoid congestion is to build a network that is well matched to the traffic that it carries.
- If there is a low-bandwidth link on the path along which most traffic is directed, congestion is likely.
- Sometimes resources can be added dynamically when there is serious congestion.

Approaches to Congestion Control

- for example:
- turning on spare routers or enabling lines that are normally used only as backups (to make
- the system fault tolerant) or purchasing bandwidth on the open market.
- More often, links and routers that are regularly heavily utilized are upgraded at the earliest opportunity.
- This is called **provisioning**

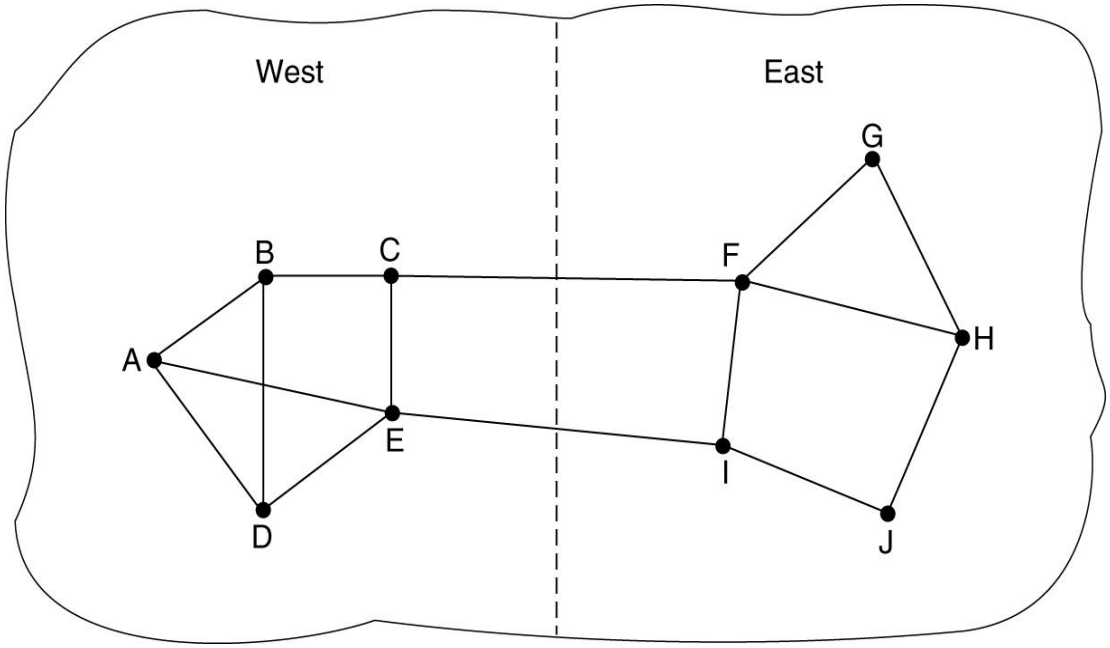
Traffic-Aware Routing

- The routing
- schemes we looked at in Sec 5.2 used fixed link weights.
- These schemes adapted to changes in topology, but not to changes in load.
- The goal in taking load into account when computing routes is to shift traffic away from hotspots

Traffic-Aware Routing

- The most direct way to do this is to set the link weight to be a function of the (fixed) link bandwidth and propagation delay plus the (variable) measured load

Traffic-Aware Routing



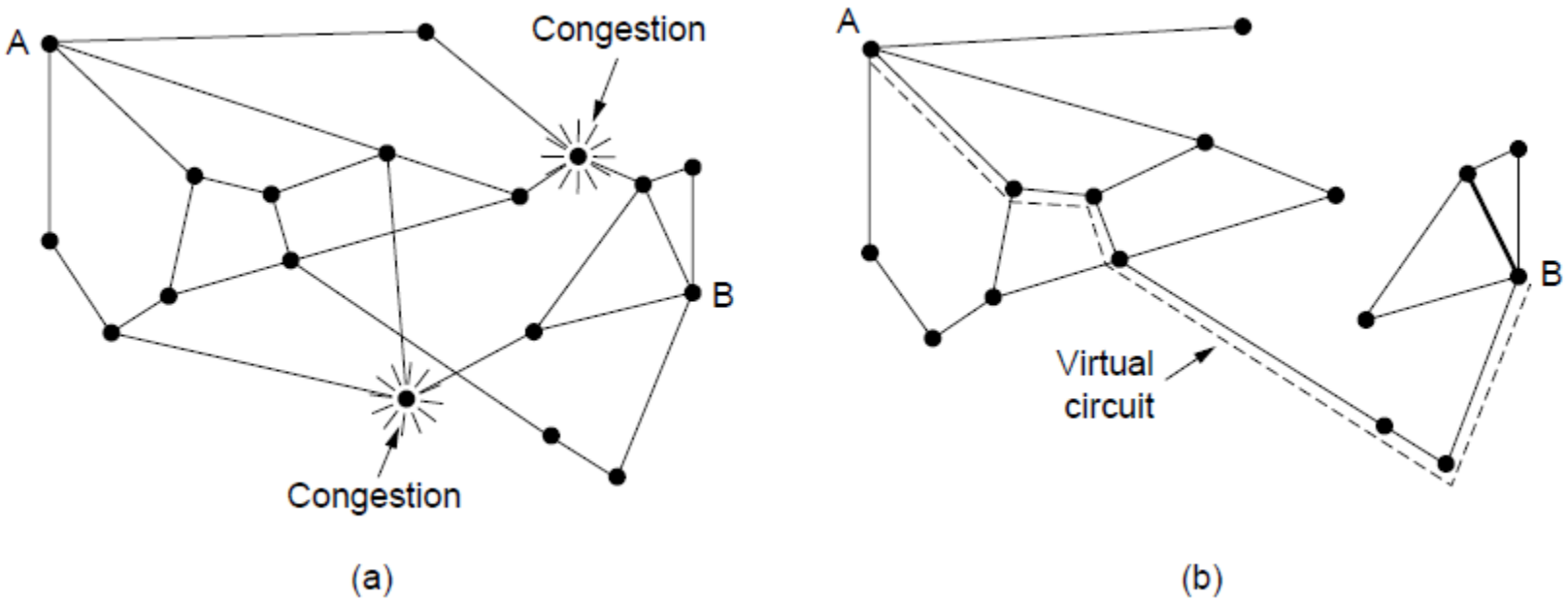
wild oscillations of routing tables, erratic routing

solution : distribution of load over multiple lines

Admission Control

- One technique that is widely used in virtual-circuit networks to keep congestion at bay is **admission control**.
- The idea is simple: do not set up a new virtual circuit unless the network can carry the added traffic without becoming congested.
- A commonly used descriptor that captures
- this effect is the **leaky bucket** or **token bucket**.

Traffic Throttling (1)



(a) A congested network. (b) The portion of the network that is not congested. A virtual circuit from A to B is also shown.

Traffic Throttling

- In the Internet and many other computer networks, senders adjust their transmissions to send as much traffic as the network can readily deliver.
- In this setting, the network aims to operate just before the onset of congestion.
- When congestion is imminent, it must tell the senders to throttle back their transmissions and slow down.
- **congestion avoidance**

Approaches To Throttling Traffic

- Each approach must solve two problems.
- First, routers must determine when congestion is approaching, ideally before it has arrived.
- To do so, each router can continuously monitor there sources it is using.
- Three possibilities are the utilization of the output links, the buffering of queued packets inside the router, and the number of packets that are lost due to insufficient buffering.

Approaches To Throttling Traffic

- The second problem is that routers must deliver timely feedback to the senders that are causing the congestion.
- To deliver feedback, the router must identify the appropriate senders.
- It must then warn them carefully, without sending many more packets into the already congested network.

Choke Packets

- The most direct way to notify a sender of congestion is to tell it directly.
- In this approach, the router selects a congested packet and sends a **choke packet**
- back to the source host, giving it the destination found in the packet.
- The original packet may be tagged (a header bit is turned on)

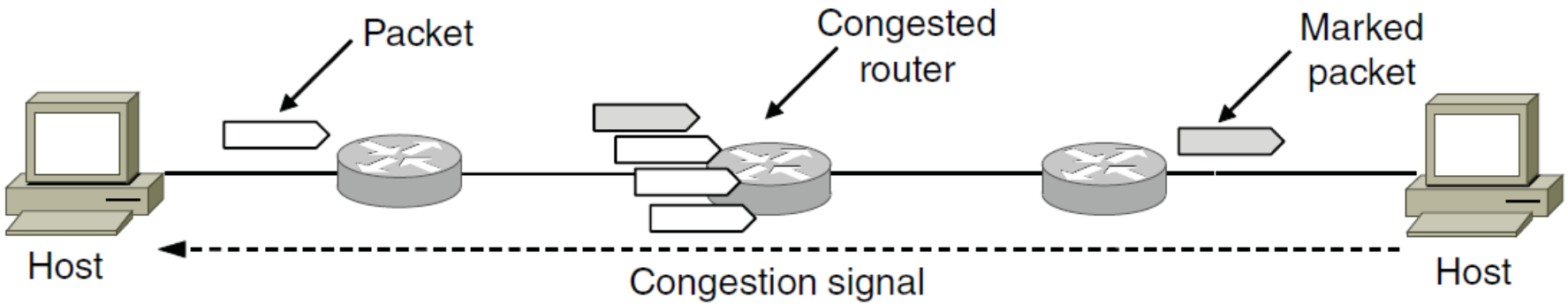
Choke Packets

- When the source host gets the choke packet, it is required to reduce the traffic sent to the specified destination, for example, by 50%.
- it is likely that multiple choke packets will be sent to a given host and destination.
- The host should ignore these additional chokes for the fixed time interval until its reduction in traffic takes effect.
- After that period, further choke packets indicate that the network is still congested.

Explicit Congestion Notification

- Instead of generating additional packets to warn of congestion, a router can tag any packet it forwards (by setting a bit in the packet's header) to signal that it is experie
- This design is called **ECN** (**Explicit Congestion Notification**) and is used in the Internet
- Two bits in the IP packet header are used to record whether the packet has experienced congestion

Traffic Throttling (2)



Explicit congestion notification

Traffic Throttling (2)

- If any of the routers they pass through is congested, that router will then mark the packet as having experienced congestion as it is forwarded.
- The destination will then echo any marks back to the sender as an explicit congestion signal in its next reply packet.
- This is shown with a dashed line in the figure to indicate that it happens above the IP level (e.g., in TCP).
- The sender must then throttle its transmissions, as in the case of choke packets.

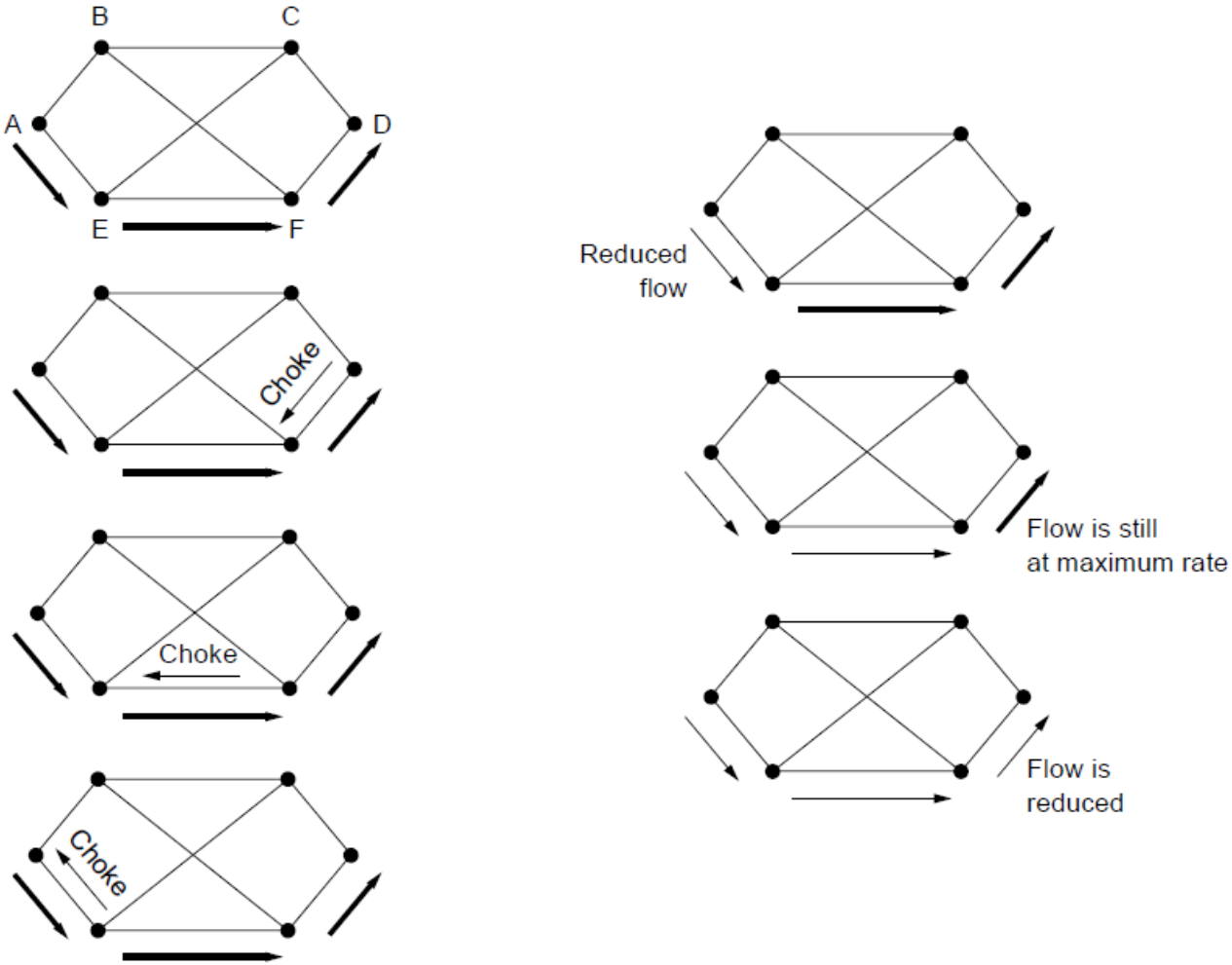
Hop-by-Hop Backpressure

- At high speeds or over long distances, many new packets may be transmitted after congestion has been signaled because of the delay before the signal takes effect.

Hop-by-Hop Backpressure

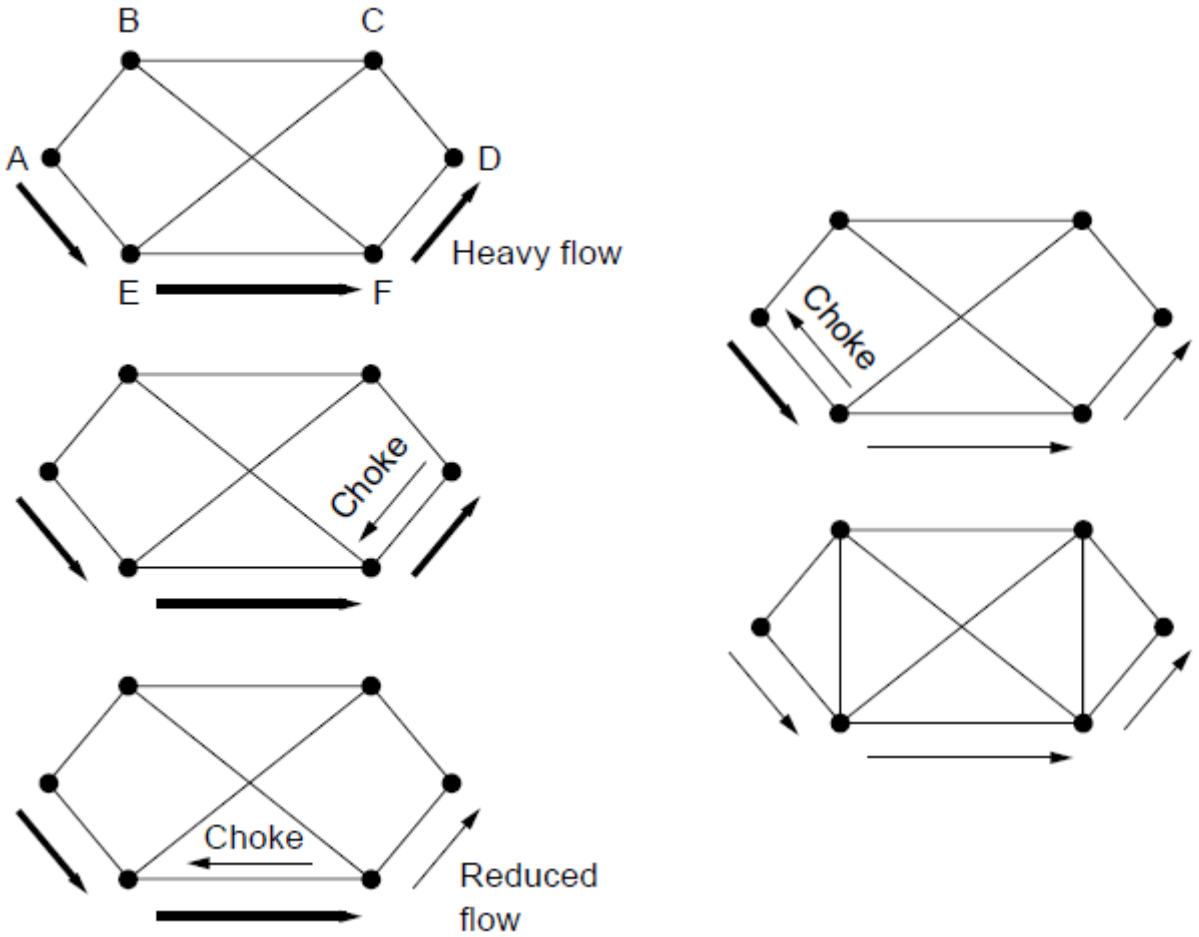
- Consider, for example, a host in San Francisco (router *A* in Fig. 5-26) that is sending traffic to a host in New York (router *D* in Fig. 5-26) at the OC-3 speed of 155 Mbps.
- If the New York host begins to run out of buffers, it will take about 40 msec for a choke packet to get back to San Francisco to tell it to slow down.
- An ECN indication will take even longer because it is delivered via the destination.
- Choke packet propagation is illustrated as the second, third, and fourth steps in

Load Shedding (1)



A choke packet that affects only the source..

Load Shedding (2)



A choke packet that affects each hop it passes through.

Hop-by-Hop Backpressure

- An alternative approach is to have the choke packet take effect at every hop it passes through, as shown in the sequence of Fig. 5-26(b).
- Here, as soon as the choke packet reaches *F*, *F* is required to reduce the flow to *D*.
- Doing so will require *F* to devote more buffers to the connection, since the source is still sending away at full blast, but it gives *D* immediate relief,
- In the next step, the choke packet reaches *E*, which tells *E*
- to reduce the flow to *F*.
- This action puts a greater demand on *E*'s buffers but gives *F* immediate relief.
- Finally, the choke packet reaches *A* and the flow genuinely slows down.

Load Shedding

- When none of the above methods make the congestion disappear : load shedding.
- **Load shedding** is a fancy way of
- saying that when routers are being inundated by packets that they cannot handle, they just throw them away.
- The term comes from the world of electrical power generation, where it refers to the practice of utilities intentionally blacking out certain areas to save the entire grid from collapsing on hot summer days

Load Shedding

- The key question for a router drowning in packets is which packets to drop.
- The preferred choice may depend on the type of applications that use the network.
- For a file transfer, an old packet is worth more than a new one.
- This is because dropping packet 6 and keeping packets 7 through 10, for example, will only force the receiver to do more work to buffer data that it cannot yet use.
- In contrast, for real-time media, a new packet is worth more than an old one.
- This is because packets become useless if they are delayed and miss the time at which they must be played out to the user.

Load Shedding

- The former policy (old is better than new) is often called **wine** and the latter (new is better than old) is often called **milk** because most people would rather drink new milk and old wine than the alternative.
- More intelligent load shedding.

Random Early Detection

- Popular algorithm for doing this is called **RED** (**Random Early Detection**) (Floyd and Jacobson, 1993).
- To determine when to start discarding, routers maintain a running average of their queue lengths.
- When the average queue length on some link exceeds a threshold, the link is said to be congested and a small fraction of the packets are dropped at random

Random Early Detection

- Dealing with congestion when it first starts is more effective than letting it gum up the works and then trying to deal with it.
- This observation leads to an interesting twist on load shedding, which is to discard packets before all the buffer space is really exhausted.
- The affected sender will notice the loss when there is no acknowledgement, and then the transport protocol will slow down.

QUALITY OF SERVICE

- The techniques we looked at in the previous sections are designed to reduce congestion and improve network performance.
- However, there are applications (and customers) that demand stronger performance guarantees from the network
- Eg: Multimedia applications: need a minimum throughput and maximum latency

QUALITY OF SERVICE

- Here also will continue our study of network performance, but
- now with a sharper focus on ways to provide **quality of service** that is matched to application needs.

overprovisioning

- An easy solution to provide good quality of service is to build a network with enough capacity.
- The resulting network will carry application traffic without significant loss and, assuming a decent routing scheme, will deliver packets with low latency.

overprovisioning

- The trouble with this solution is that it is expensive.
- Quality of service mechanisms let a network with less capacity meet application requirements just as well at a lower cost.

Four issues must be addressed to ensure quality of service:

- 1. What applications need from the network.
- 2. How to regulate the traffic that enters the network.
- 3. How to reserve resources at routers to guarantee performance.
- 4. Whether the network can safely accept more traffic.

Four issues must be addressed to ensure quality of service:

- No single technique deals efficiently with all these issues.
- Instead, a variety of techniques have been developed for use at the network (and transport) layer.
- Practical quality-of-service solutions combine multiple techniques.

Application Requirements

- A stream of packets from a source to a destination is called a **flow.**
- A flow might be all the packets of a connection in a connection-oriented network, or
- all the packets sent from one process to another process in a connectionless network.

Application Requirements

- The needs of each flow can be characterized by four primary
- parameters: bandwidth, delay, jitter, and loss. Together,
- these determine the
- **QoS** (**Quality of Service**) the flow requires.

Several common applications and the stringency of their network requirements

Application	Bandwidth	Delay	Jitter	Loss
Email	Low	Low	Low	Medium
File sharing	High	Low	Low	Medium
Web access	Medium	Medium	Low	Medium
Remote login	Low	Medium	Medium	Medium
Audio on demand	Low	Low	High	Low
Video on demand	High	Low	High	Low
Telephony	Low	High	High	Low
Videoconferencing	High	High	High	Low

QoS

- To accommodate a variety of applications, networks may support different categories of QoS.
- An ATM networks,
- 1. Constant bit rate (e.g., telephony).
- 2. Real-time variable bit rate (e.g., compressed videoconferencing).
- 3. Non-real-time variable bit rate (e.g., watching a movie on demand).
- 4. Available bit rate (e.g., file transfer).

Traffic Shaping

- Before the network can make QoS guarantees, it must know what traffic is being guaranteed.
- In the telephone network, this characterization is simple.
- For example, a voice call (in uncompressed format) needs 64 kbps and consists of one
- 8-bit sample every 125 µsec.

Traffic Shaping

- However, traffic in data networks is **bursty**. It typically arrives at nonuniform rates as the traffic rate varies, users interact with applications, and computers switch between tasks.
- Bursts of traffic are more difficult to handle

Traffic Shaping

- **Traffic shaping** is a technique for regulating the average rate and burstiness of a flow of data that enters the network.
- The goal is to allow applications to transmit a wide variety of traffic that suits their needs, including some bursts, yet have a simple and useful way to describe the possible traffic patterns to the network.
- When a flow is set up, the user and the network (i.e., the customer and the provider) agree on a certain traffic pattern (i.e., shape) for that flow.
- In effect, the customer says to the provider ‘‘My transmission pattern will look like this; can you handle it?’’

Traffic Shaping

- This agreement is called an **SLA** (**Service Level Agreement**) when it is made over aggregate flows and long periods of time.
- Traffic shaping reduces congestion and thus helps the network live up to its promise.
- However, to make it work, there is also the issue of how the provider can tell if the customer is following the agreement and what to do if the customer is not.

Traffic Shaping

- Packets in excess of the agreed pattern might be dropped by the network, or
- they might be marked as having lower priority.
- Monitoring a traffic flow is called **traffic policing**.
- Shaping and policing are not so important for peer-to-peer and other transfers, but they are of great importance for real-time data, such as audio and video connections, which have stringent quality-of-service requirements.

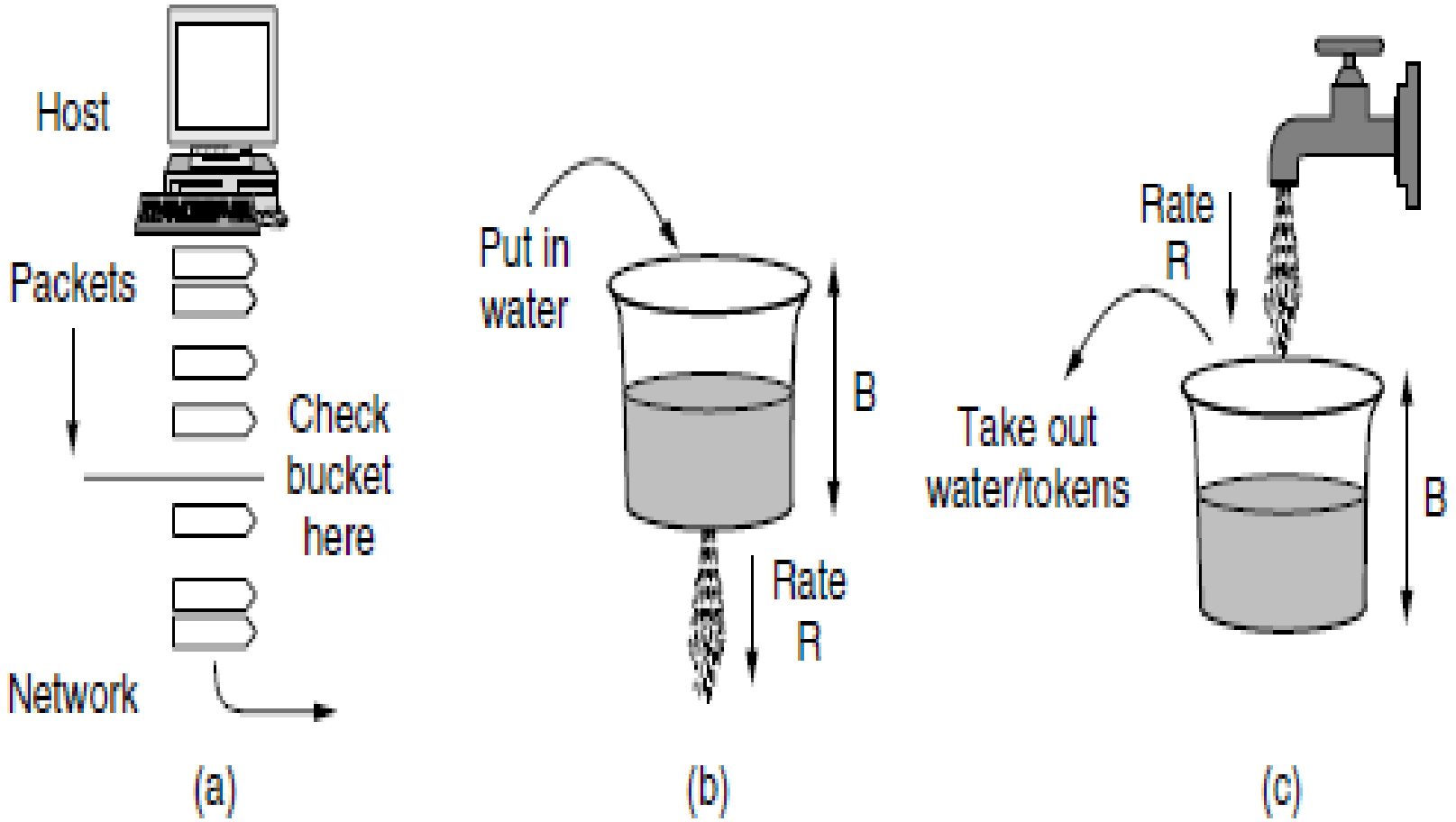
Leaky and Token Buckets

- Try to imagine a bucket with a small hole in the bottom, as illustrated in Fig.
- No matter the rate at which water enters the bucket, the outflow is at a constant rate, *R*, when there is any water in the bucket and zero when the bucket is empty.
- Also, once the bucket is full to capacity *B*, any additional water entering it spills over the sides and is lost.

Leaky and Token Buckets

- Try to imagine a bucket with a small hole in the bottom, as illustrated in Fig b.
- No matter the rate at which water enters the bucket, the outflow is at a constant rate, *R*, when there is any water in the bucket and zero when the bucket is empty.
- Also, once the bucket is full to capacity *B*, any additional water entering it spills over the sides and is lost.

Leaky and Token Buckets



Leaky and Token Buckets

- This bucket can be used to shape or police packets entering the network, as shown in Fig a.
- Conceptually, each host is connected to the network by an interface containing a leaky bucket.
- To send a packet into the network, it must be possible to put more water into the bucket.
- If a packet arrives when the bucket is full, the packet must either be queued until enough water leaks out to hold it or be discarded.
- This technique was proposed by Turner (1986) and is called the **leaky bucket algorithm**.

Leaky and Token Buckets

- A different but equivalent formulation is to imagine the network interface as a bucket that is being filled, as shown in Fig. (c).
- The tap is running at rate *R* and the bucket has a capacity of *B*, as before.
- Now, to send a packet we must be able to take water, or tokens, as the contents are commonly called, out of the bucket (rather than putting water into the bucket). No more than a fixed number of tokens, *B*, can accumulate in the bucket, and if the bucket is empty, we must wait until more tokens arrive before we can send another packet.
- This algorithm is called the **token bucket algorithm**.

Packet Scheduling

- Being able to regulate the shape of the offered traffic is a good start.
- However, to provide a performance guarantee, we must reserve sufficient resources along the route that the packets take through the network.
- Algorithms that allocate router resources among the packets of a flow and between competing flows are called **packet scheduling algorithms**

Packet Scheduling

- Three different
- kinds of resources can potentially be reserved for different flows:
- 1. Bandwidth.
- 2. Buffer space.
- 3. C
- Packet scheduling algorithms allocate bandwidth and other router resources by determining which of the buffered packets to send on the output line next.

Packet Scheduling

- **1. FIFO (First-In First-Out)**, or equivalently **FCFS (First-Come First-Serve)**.
- 2. **fair queuing**
- **3. WFQ (Weighted Fair Queuing)**.

Packet Scheduling

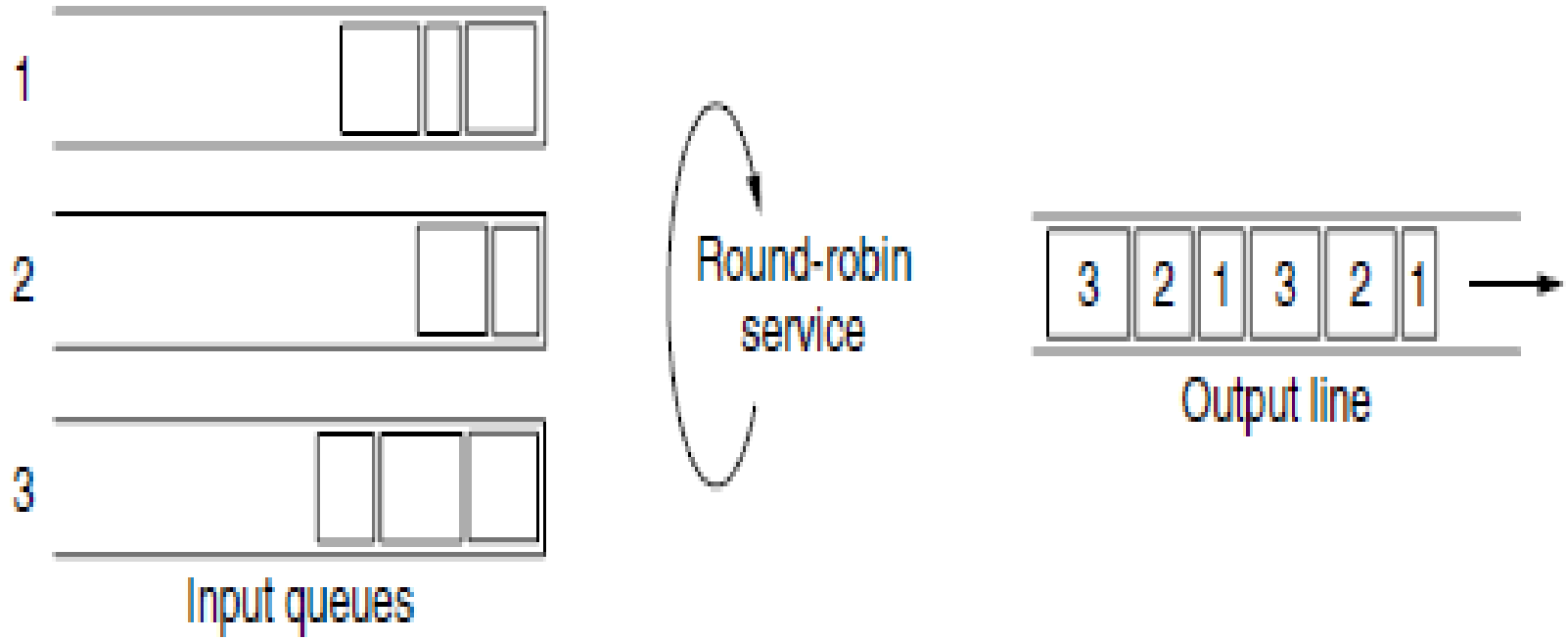
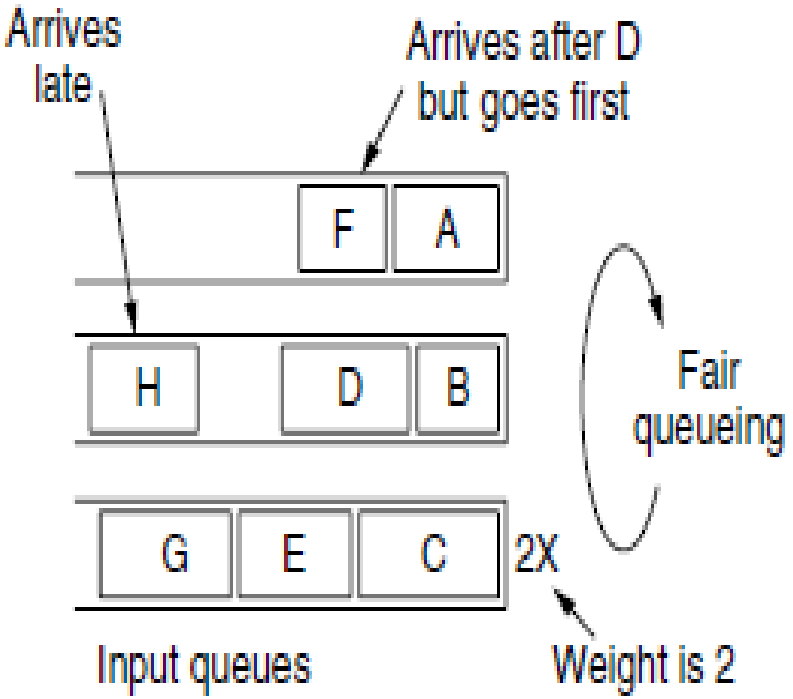


Figure 5-30. Round-robin fair queueing.

Packet Scheduling



(a)

Packet	Arrival time	Length	Finish time	Output order
A	0	8	8	1
B	5	6	11	3
C	5	10	10	2
D	8	9	20	7
E	8	8	14	4
F	10	6	16	5
G	11	10	19	6
H	20	8	28	8

(b)

Admission Control

- QoS guarantees are established through the process of admission control.
- The user offers a flow with an accompanying QoS requirement to the network.
- The network then decides whether to accept or reject the flow based on its capacity and the commitments it has made to other flows.
- If it accepts, the network reserves capacity in advance at routers to guarantee QoS

Admission Control

- The reservations must be made at all of the routers along the route that the packets take through the network.
- This is called **QoS routing**.

Integrated Services

- Architecture for streaming multimedia.
- Supports unicast and multicast applications.
- An example of the former is a single user streaming a video clip from a news site.
- An example of the latter is a collection of digital television stations broadcasting their programs as streams of IP packets to many receivers at various locations.

Integrated Services

- Architecture for streaming multimedia.
- Supports unicast and multicast applications.
- An example of the former is a single user streaming a video clip from a news site.
- An example of the latter is a collection of digital television stations broadcasting their programs as streams of IP packets to many receivers at various locations.

RSVP—The Resource reSerVation Protocol

- The main part of the integrated services architecture that is visible to the users of the network is **RSVP**.
- It is described in RFCs 2205–2210.
- This protocol is used for making the reservations; other protocols are used for sending the data.

RSVP—The Resource reSerVation Protocol

- RSVP allows multiple senders to transmit to multiple groups of receivers,
- permits individual receivers to switch channels freely,
- and optimizes bandwidth use while
- at the same time eliminating congestion.

RSVP—The Resource reSerVation Protocol

- In its simplest form, the protocol uses multicast routing using spanning trees.
- Each group is assigned a group address.
- To send to a group, a sender puts the group's address in its packets.
- The standard multicast routing algorithm then builds a spanning tree covering all group members.
- The routing algorithm is not part of RSVP.
- The only difference from normal multicasting is a little extra information that is multicast to the group periodically to tell the routers along the tree to maintain certain data structures in their memories.

RSVP—The Resource reSerVation Protocol

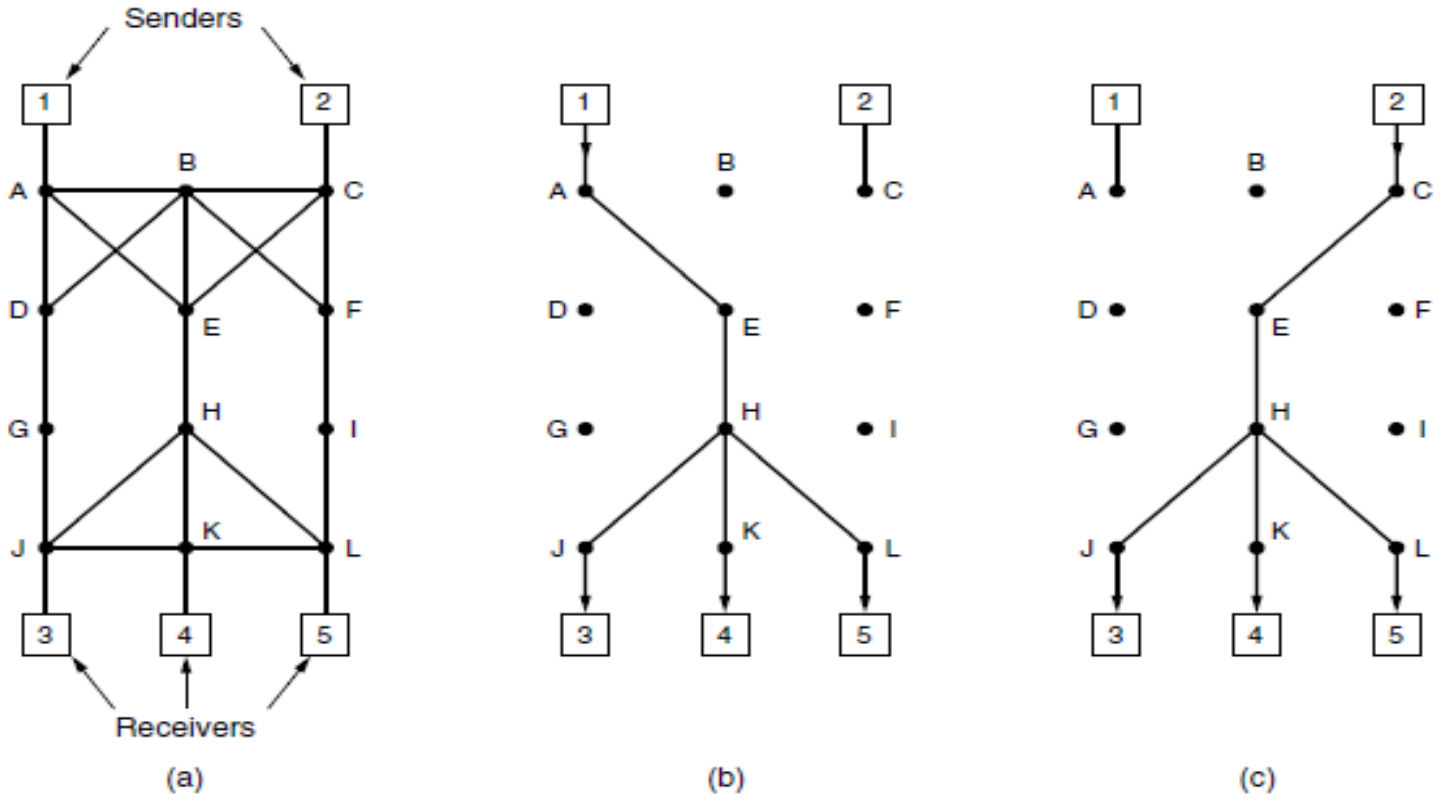


Figure 5-34. (a) A network. (b) The multicast spanning tree for host 1. (c) The multicast spanning tree for host 2.

RSVP—The Resource reSerVation Protocol

- As an example, consider the network of Fig.
- Hosts 1 and 2 are multicast senders, and hosts 3, 4, and 5 are multicast receivers.
- The multicast trees for hosts 1 and 2 are shown in Fig. 5-34(b) and Fig. 5-34(c), respectively.

RSVP—The Resource reSerVation Protocol

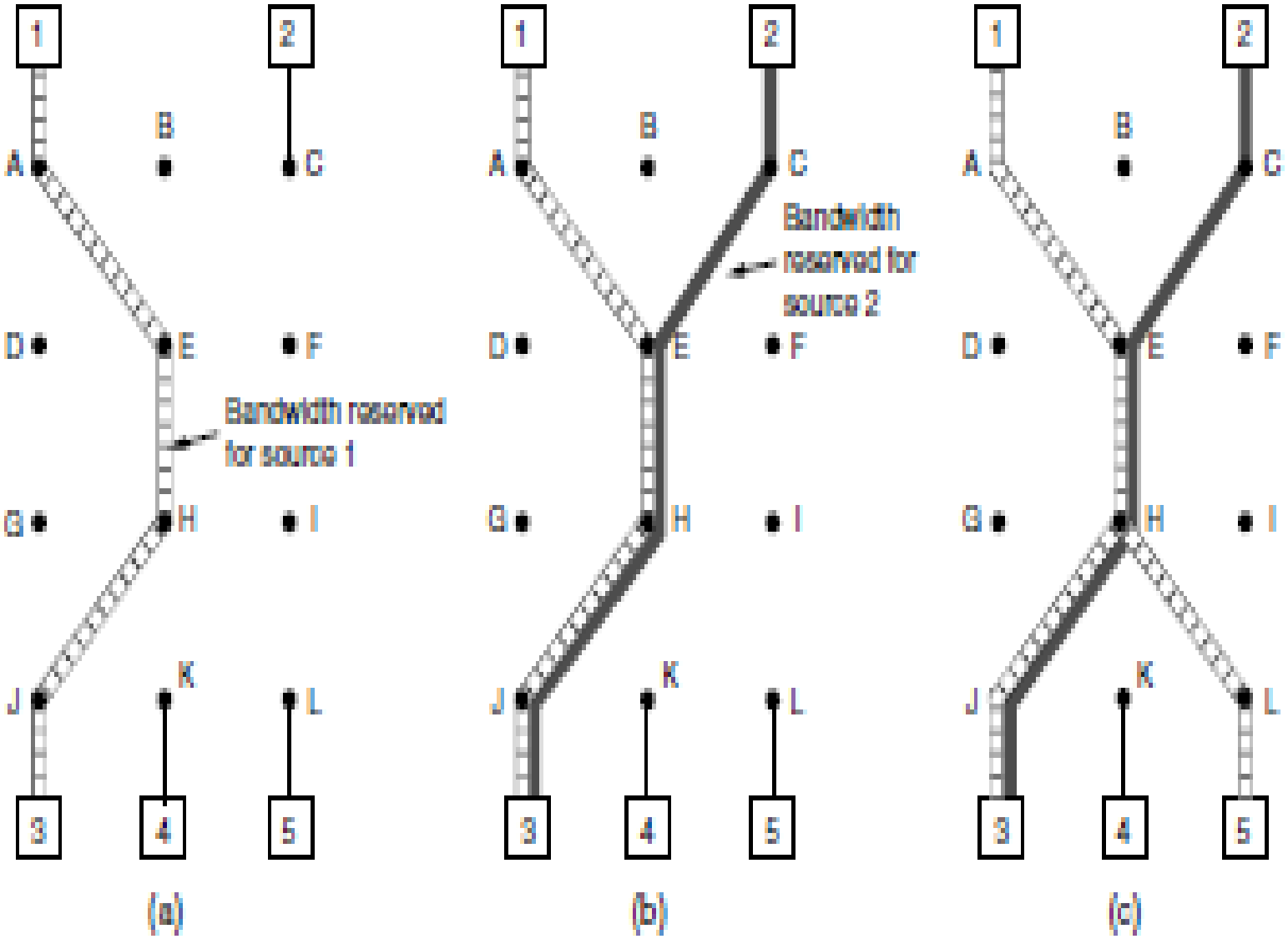
- To get better reception and eliminate congestion, any of the receivers in a group can send a reservation message up the tree to the sender.
- The message is propagated using the reverse path forwarding algorithm discussed earlier.
- At each hop, the router notes the reservation and reserves the necessary bandwidth.

RSVP—The Resource reSerVation Protocol

- weighted fair queueing scheduler can be used to make this reservation.
- If insufficient bandwidth is available, it reports back failure.

RSVP—The Resource reSerVation Protocol

- An example of such a reservation is shown in Fig. (a).
- Here host 3 has requested a channel to host 1.
- Once it has been established, packets can flow from 1 to 3 without congestion.
- Now consider what happens if host 3 next reserves a channel to the other sender, host 2, so the user can watch two television programs at once.
- A second path is reserved, as illustrated in Fig. (b).
- Note that two separate channels are needed from host 3 to router *E* because two independent streams are being transmitted


1
2
A
B
C
D
E
F
G
H
I
J
K
L
3
4
5
Bandwidth reserved for source 1
Bandwidth reserved for source 2
(a)
(b)
(c)

RSVP—The Resource reSerVation Protocol

- Finally, in Fig. (c), host 5 decides to watch the program being transmitted by host 1 and also makes a reservation.
- First, dedicated bandwidth is reserved as far as router *H*.
- However, this router sees that it already has a feed from host 1, so if the necessary bandwidth has already been reserved, it does not have to reserve any more.

Differentiated Services

- Flow-based algorithms have the potential to offer good quality of service to one or more flows because they reserve whatever resources are needed along the route.

Differentiated Services

- Flow-based algorithms have the potential to offer good quality of service to one or more flows because they reserve whatever resources are needed along the route.
- However, they also have a downside.

Differentiated Services

- They require an advance setup to establish each flow, something that does not scale well when there are thousands or millions of flows.
- Also, they maintain internal per-flow state in the routers, making them vulnerable to router crashes.
- Finally, the changes required to the router code are substantial and involve complex router-to-router exchanges for setting up the flows.

Differentiated Services

- For these reasons, IETF has also devised a simpler approach to quality of service, one that can be largely implemented locally in each router without advance setup and without having the whole path involved.
- This approach is known as **class-based** (as opposed to flow-based) quality of service.
- IETF has standardized an architecture for it, called **differentiated services**, which is described in RFCs2474, 2475,

Differentiated Services

- The administration defines a set of service classes with corresponding forwarding rules.
- If a customer subscribes to differentiated services, customer packets entering the domain are marked with the class to which they belong.
- This information is carried in the *Differentiated services* field of IPv4 and IPv6 packets (described in Sec. 5.6).
- The classes are defined as **per hop behaviors** because they correspond to the treatment the packet will receive at each router, not a guarantee across the network.
- Better service is provided to packets with some per-hop behaviors (e.g., premium service) than to others (e.g., regular service).

Differentiated Services

- no advance setup, no resource reservation, and no time-consuming end-to-end negotiation for each flow, as with integrated services.
- This makes differentiated services relatively easy to implement.
- Eg. Package delivery, Airlines, Train etc

Computer Networks
Network Layer

Topics to be discussed:

- **internetworking**

- **the network layer in the Internet**

23.03.2020 17:09
83

Computer Networks

Network Layer

Internetworking

networks are not homogeneous

different networks (LANs, WANs, MANs) and protocols are widely used in every layer

two or more networks are connected to form an ***internet***

varieties of different networks and protocols will stay :

- installed / legacy base is large
- decisions for implementations are decentralized

23.03.2020 17:09
84

Computer Networks

Network Layer

Internetworking

Connection of varieties of networks --- LANs, MANs, WANs --- with numerous protocols in every layer

(a) TCP / IP, Novell NCP/IPX, AppleTalk used by PCs / LANs

(b) TCP / IP used by Unix workstations

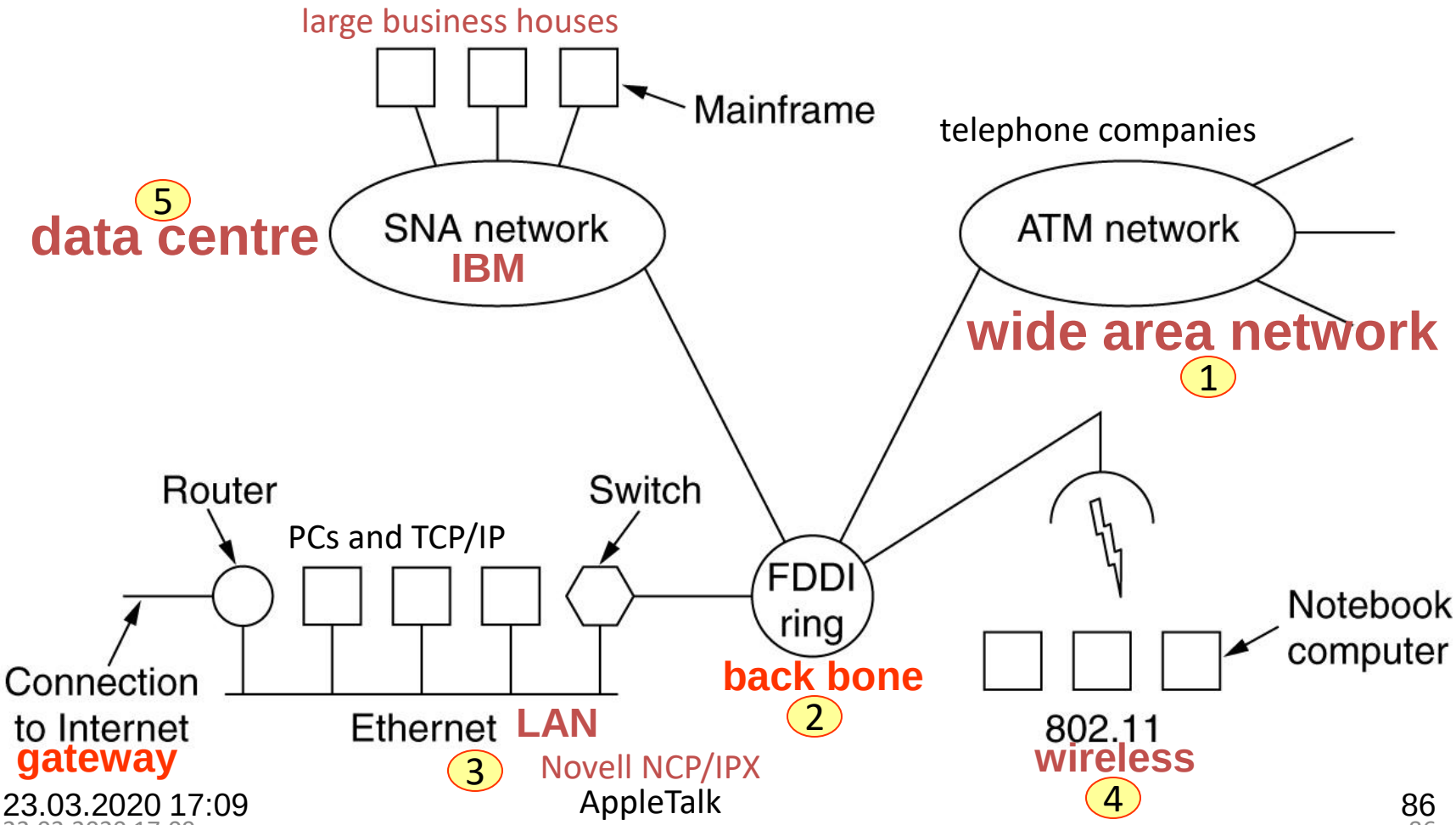
(c) SNA used by Mainframes

(d) ATM used by telephone companies

(e) Protocols used by Wireless Networks

23.03.2020 17:09
85

Computer Networks
Network Layer
Internetworking



Computer Networks
Network Layer
Internetworking
Differences in networks

Item	Some Possibilities
Service offered	Connection oriented versus connectionless
Protocols	IP, IPX, SNA, ATM, MPLS, AppleTalk, etc.
Addressing	Flat (802) versus hierarchical (IP)
Multicasting	Present or absent (also broadcasting)
Packet size	Every network has its own maximum
Quality of service	Present or absent; many different kinds
Error handling	Reliable, ordered, and unordered delivery
Flow control	Sliding window, rate control, other, or none
Congestion control	Leaky bucket, token bucket, RED, choke packets, etc.
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, by packet, by byte, or not at all

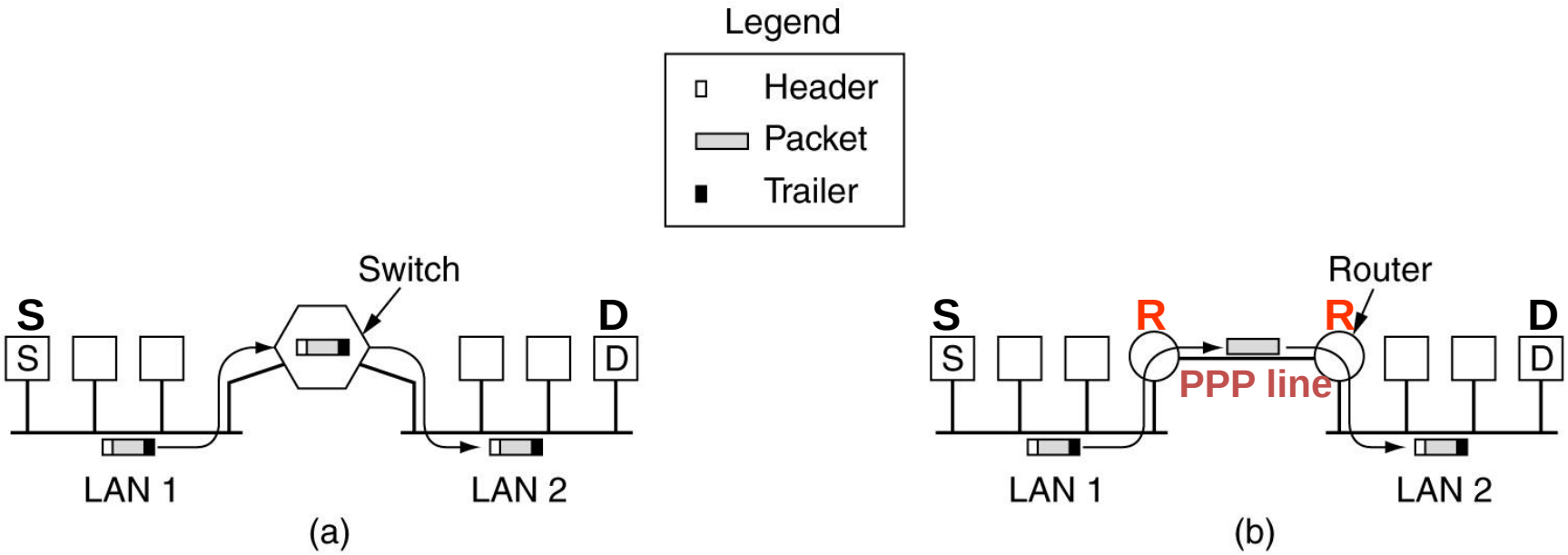
23.03.2020 17:09
87

Computer Networks

Network Layer

Internetworking

Connection of different networks



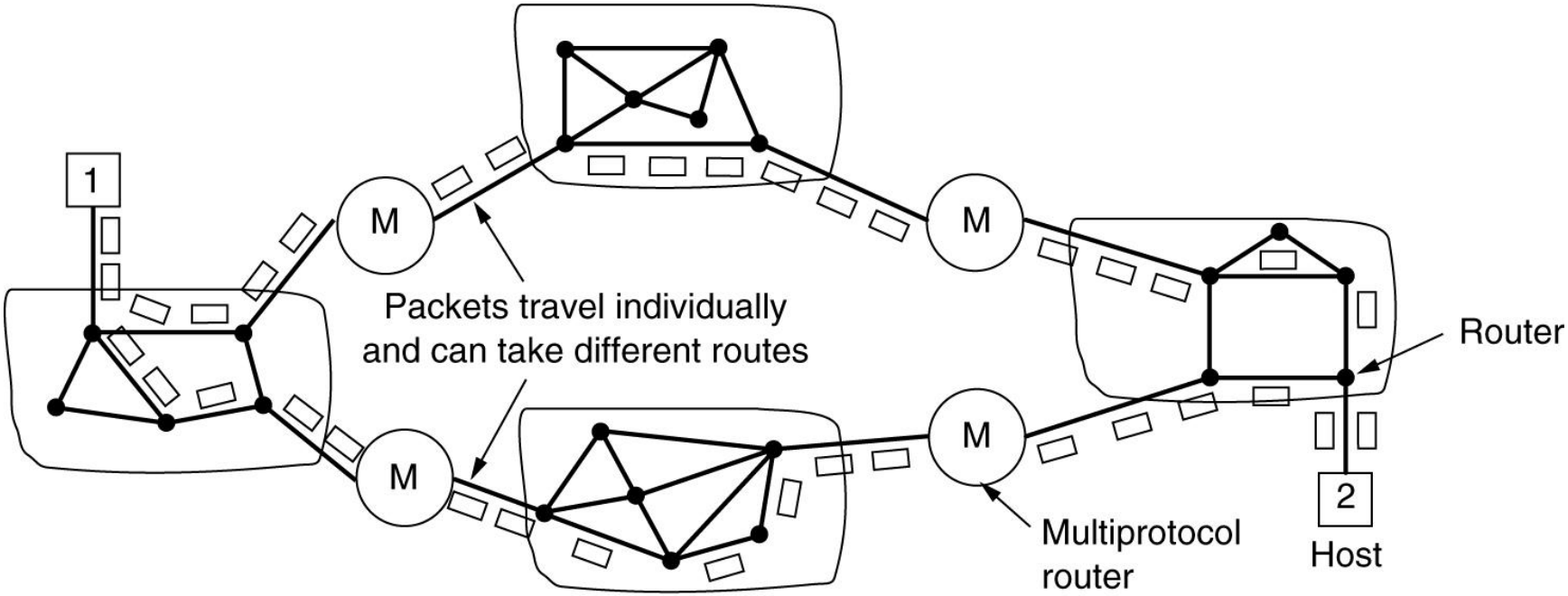
Connected by SWITCH (uses MAC address)

Connected by ROUTER (uses IP address)

23.03.2020 17:09
88

Computer Networks
Network Layer
Internetworking
Connectionless Internetworking
(datagram model)

multi-protocol routers translate packets of different formats belonging to two networks that have dissimilar network layers



23.03.2020 17:09
89

Computer Networks

Network Layer

Internetworking

Connectionless Internetworking

the only service the network layer offers to the transport layer is the ability to introduce datagrams into the subnet

packets can traverse thro' different gateways

routing decision is traffic-dependent, and is made separately for each packet

no assured delivery / in-order delivery

23.03.2020 17:09
90

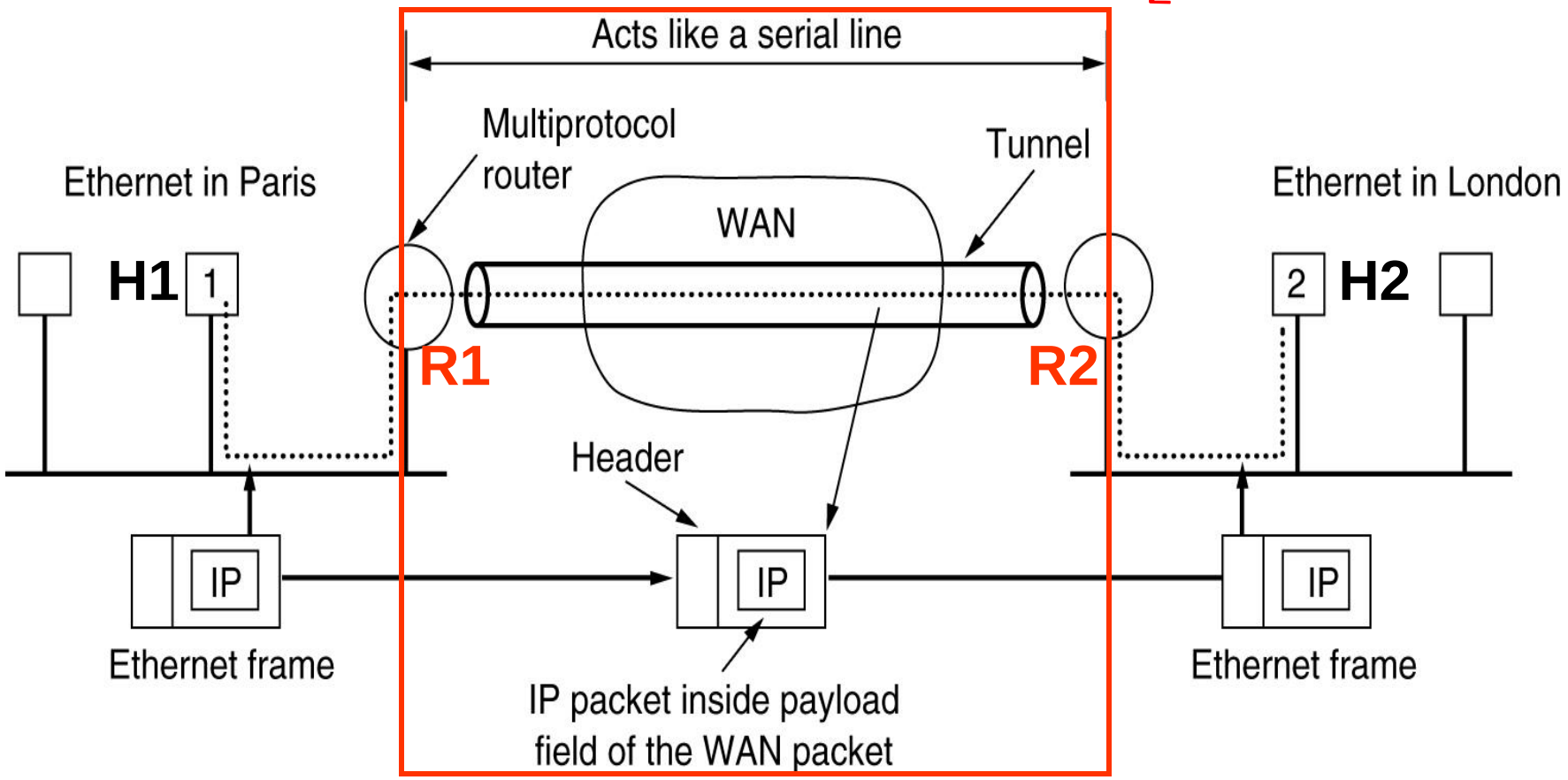
Computer Networks

Network Layer

Internetworking

source and destination hosts connected to same type of network, but there is a different intermediary network

Tunnelling



Computer Networks

Network Layer

Internetworking

Tunnelling

Host H1 is required to send an IP packet to a remote (large distance) host H2

(a) H1 constructs an IP packet containing IP address of H2

(b) H1 inserts IP address of H2 into an Ethernet frame addressed to MP Router R1

(c) R1, on receipt, removes the IP packet

23.03.2020 17:09
92

Computer Networks
Network Layer
Internetworking

Tunnelling

(d) R1 inserts frame in the payload field of WAN Network Layer packet

(e) R1 addresses frame to the WAN address of R2

(f) R2 removes the IP packet inserted in step (b)

(g) R2 sends frame to H2 inside an Ethernet frame

23.03.2020 17:09
93

Computer Networks
Network Layer
Internetworking

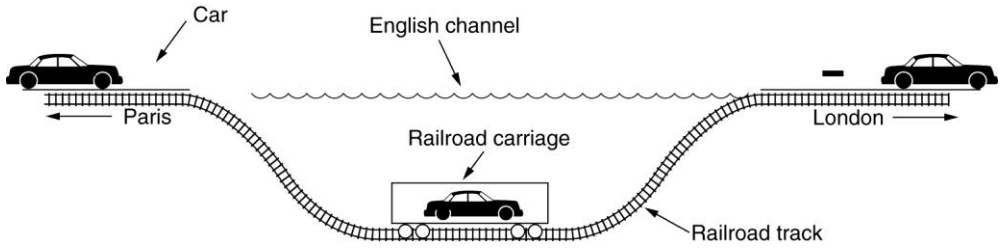
Tunnelling

The WAN can be perceived as a big tunnel extending from MP- R1 to MP - R2

IP packet, encapsulated, just travels in the tunnel ; not required to deal with WAN

MP - R is required to deal with IP and WAN packets

Analogy



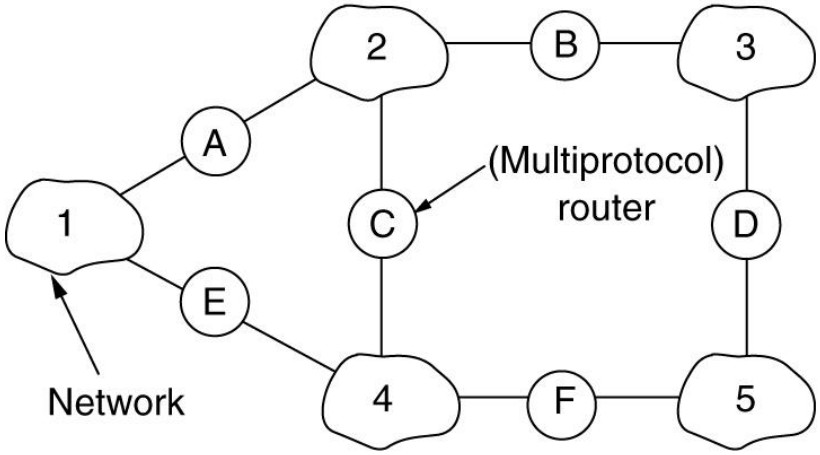
23.03.2020 17:09
94

Computer Networks

Network Layer

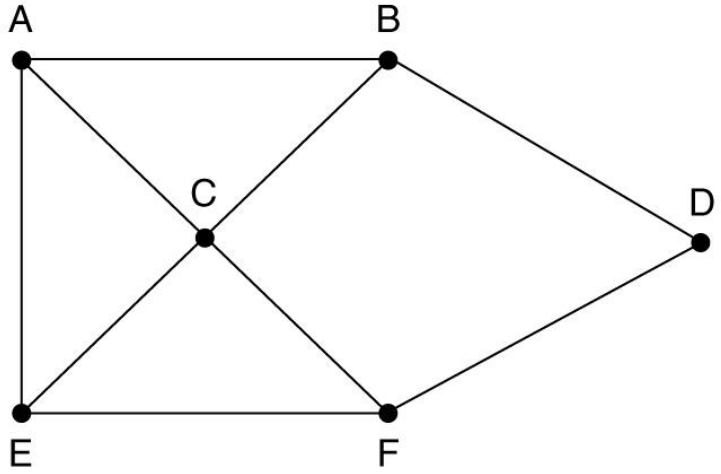
Internetworking

Internetwork Routing



(a)

An internetwork



(b)

A graph of the internetwork

23.03.2020 17:09
95

Computer Networks

Network Layer

Internetworking

Internetwork Routing

(a) graph is constructed

(b) known routing algorithms, such as DV, LS are applied to MP routers (gateways)

routing algorithms are used at two-levels :

- *interior* gateway protocol, *within* each NW
- *exterior* gateway protocol, *between* NWs

each network is independent (referred to as Autonomous System - AS) of all others and may use different algorithms

23.03.2020 17:09
96

Computer Networks

Network Layer

Internetworking

Internetwork Routing

(i) internet packet starts from its LAN, addressed to Multi Protocol Router (MPR)

(ii) MPR decides, using its own routing tables, which ***way*** it is to be forwarded

(iii) if the distant MPR can be reached, packet is forwarded directly

(iv) else, packet is encapsulated in the protocol required by the intervening network and tunneled

(v) process is repeated until packet reaches destination network

23.03.2020 17:09
97

Computer Networks
Network Layer
Internetworking

Fragmentation

each network imposes some maximum size on its packets; reasons :

(a) hardware (e.g. size of Ethernet frame)

(b) OS (e.g. all buffers are, say, 512 bytes)

(c) protocols (e.g. the no. of bits in packet length field)

(e) compliance with some standard (national / international)

(f) need to reduce error-induced retransmissions to some level

(g) need to reduce packet dwell time in a channel

23.03.2020 17:09
98

Computer Networks
Network Layer
Internetworking

Fragmentation

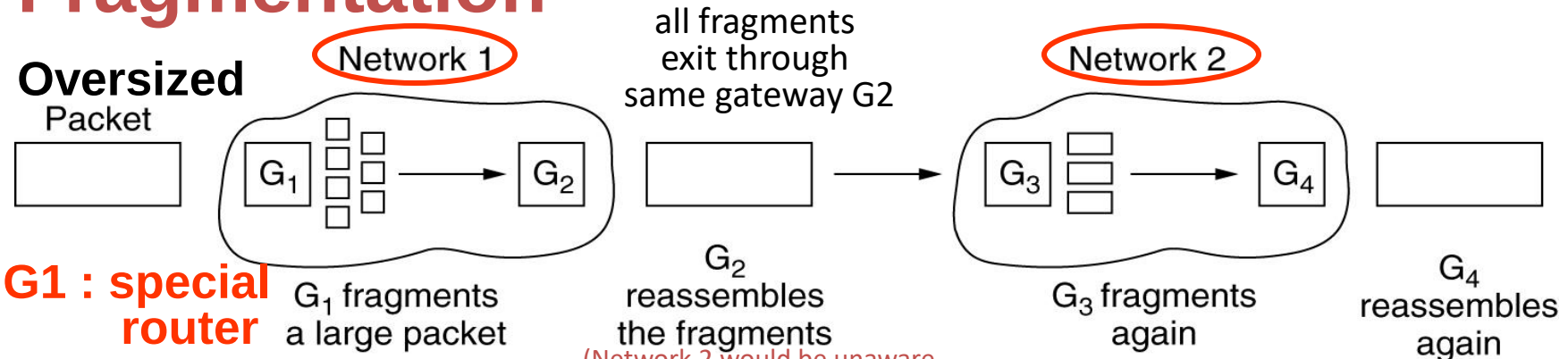
network designers have constraints on maximum payloads : 48 bytes (ATM cells) and 65515 (IP packets) →
larger payloads of higher layers need to be fragmented

even if routing algorithms take care to ensure routing through networks which can accommodate higher payloads, … if destination network cannot, fragmentation is inevitable

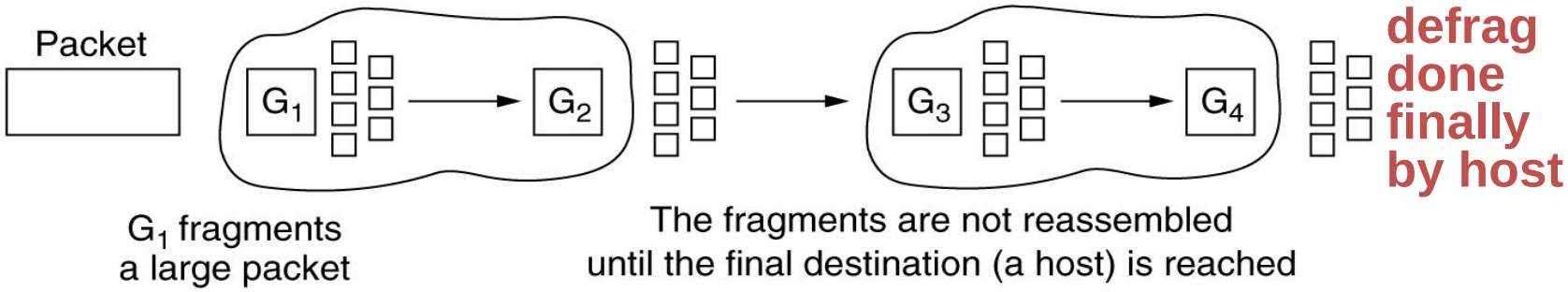
23.03.2020 17:09
99

Computer Networks
Network Layer
Internetworking

Fragmentation



(a) **Transparent fragmentation**



(b) **Nontransparent fragmentation**

Computer Networks
Network Layer
Internetworking

Fragmentation

Transparent : (e.g. ATM networks) (#)
exit gateway must keep packet count / end of packet field to ensure arrival of all fragments before reassembly
all fragments to exit thro' the same router
… performance issues
large overhead of repeated de- / re- frag.

(#) ATM networks have special hardware to fragment large size packets into cells and then reassemble cells into packets

23.03.2020 17:09
101

Computer Networks
Network Layer
Internetworking
Fragmentation

Nontransparent : (e.g. IP network)
every **host has to be able to reassemble fragments into original packets**
increase of overhead due to headers for every fragmented packet

23.03.2020 17:09
102

Computer Networks
Network Layer
The Network Layer in the Internet
Ten design principles as per RFC 1958

Sl. No.	Principle
1.	Make sure it works ; make / integrate many prototypes
2.	Keep it simple ; leave out not-essential features
3.	Make clear choices ; standards have many options, choose one
4.	Exploit modularity ; have protocol stacks, independent
5.	Expect heterogeneity ; simple, general, flexible design
6.	Avoid static options and parameters ; rather negotiate
7.	Look for a good design ; it need not be perfect
8.	Be strict when sending and tolerant when receiving
9.	Think about scalability; spread loads over resources
10.	Consider performance and cost

23.03.2020 17:09
106

Computer Networks

Network Layer

The Network Layer in the Internet

- the Internet may be viewed as a collection of interconnected sub-networks or ***Autonomous Systems*** at the network layer
- no real structure, but several backbones
- high-speed Lines and fast Routers
- attached to backbones are regional networks comprising LANs
- the Network Layer protocol viz. ***Internet Protocol*** (***IP***) holds the Internet together

23.03.2020 17:09
107

Computer Networks
Network Layer
The Network Layer in the Internet

Network layer :
job is to provide a *best-efforts* (not guaranteed) way to *transport datagrams* from source to destination …….

irrespective of whether these are on the same network or have many networks in between them

23.03.2020 17:09
108

Computer Networks

The Network Layer in the Internet

Communication in the Internet : steps

(a) transport layer takes data streams, breaks them up into datagrams

(b) each datagram is transmitted thro' the Internet, possibly fragmented

(c) when all fragments reach the destination, they are reassembled by the network layer into the original datagram

(d) datagram is then handed to the transport layer at the destination

(e) transport layer inserts it into the input stream of the receiving process

23.03.2020 17:09
109

Computer Networks

The Network Layer in the Internet

The IP Protocol : format of IP datagram

an IP datagram consists of :

- header part
- data part

the header has :

- a 20-byte fixed part
- a variable length optional part

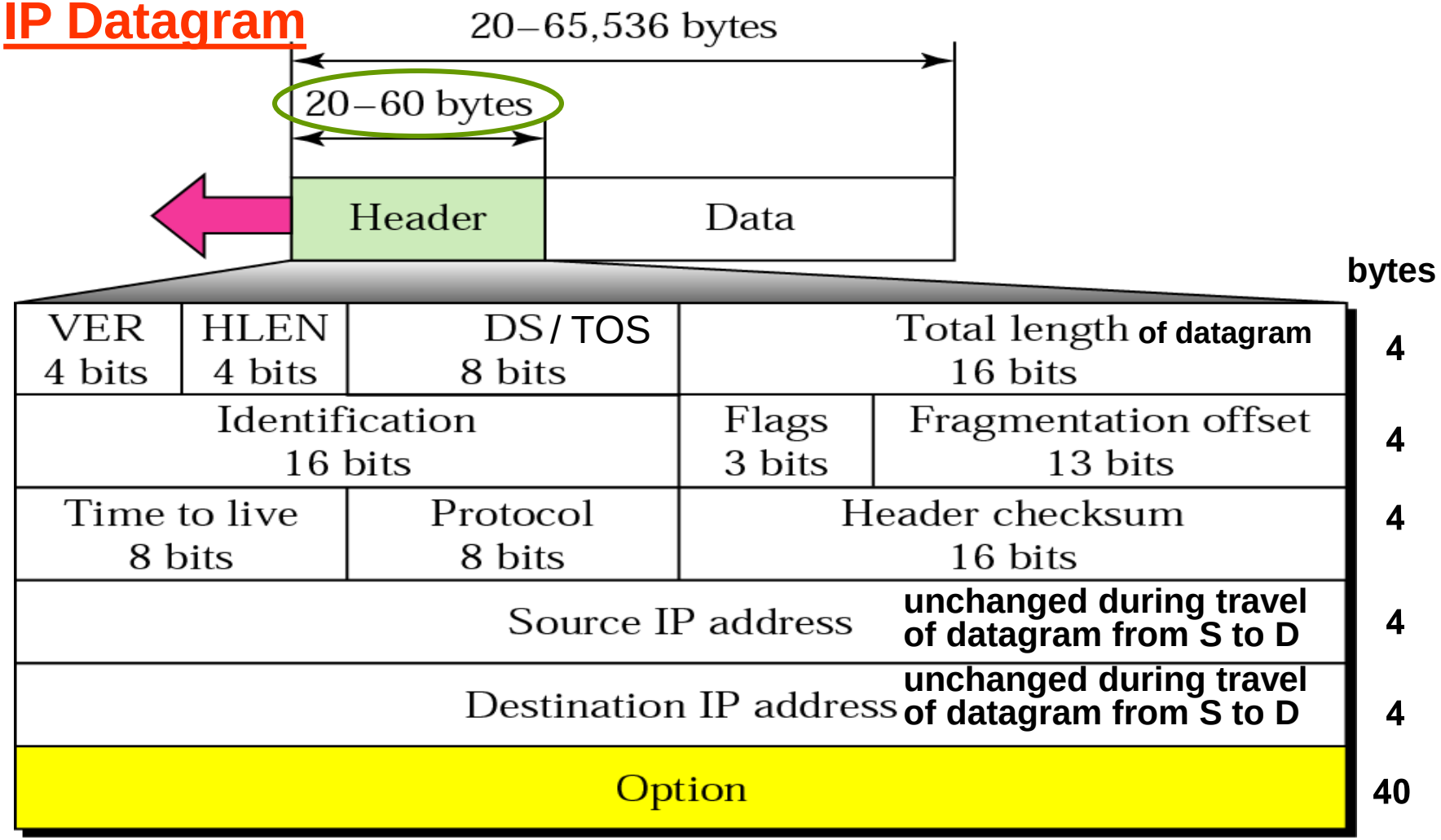
header format →

23.03.2020 17:09
111

Computer Networks

The IP Protocol : key part is the IP datagram

IP Datagram



The IPv4 Header

23.03.2020 17:09
112

Computer Networks

Network Layer

The Network Layer in the Internet

The IP protocol : IPv4 header

IPv4 header fields : (12 fixed fields)

Version **: keeps track of which version of protocol the datagram belongs to**

HLEN **: length of the header in 32-bit (4-byte) words;**

min. = 5 ; max. = 15

23.03.2020 17:09
113

Computer Networks

Network Layer

The Network Layer in the Internet

IPv4 header fields : (contd.)

Type of Service **: to distinguish between various classes of service : reliability, speed, ….**

Total length **: includes everything in the datagram (header + data) max. = 65535 bytes**

Identification **: to allow destination host to determine which datagram an arrived fragment belongs to**

23.03.2020 17:09
114

Computer Networks

Network Layer

The Network Layer in the Internet

IPv4 header fields : (contd.)

***DF* : 'do not fragment' bit**

***MF* : 'more fragment' bit ; indicates 'more to come' ; last fragment will have MF = 0**

***Fragment offset* : indicates where in the current datagram this fragment belongs; all fragments (except last) must be n x 8 bytes long ; 13 bits are provided i.e. max. of 8192 fragments per datagram**

23.03.2020 17:09
115

Computer Networks
Network Layer
The Network Layer in the Internet

IPv4 header fields : (contd.)

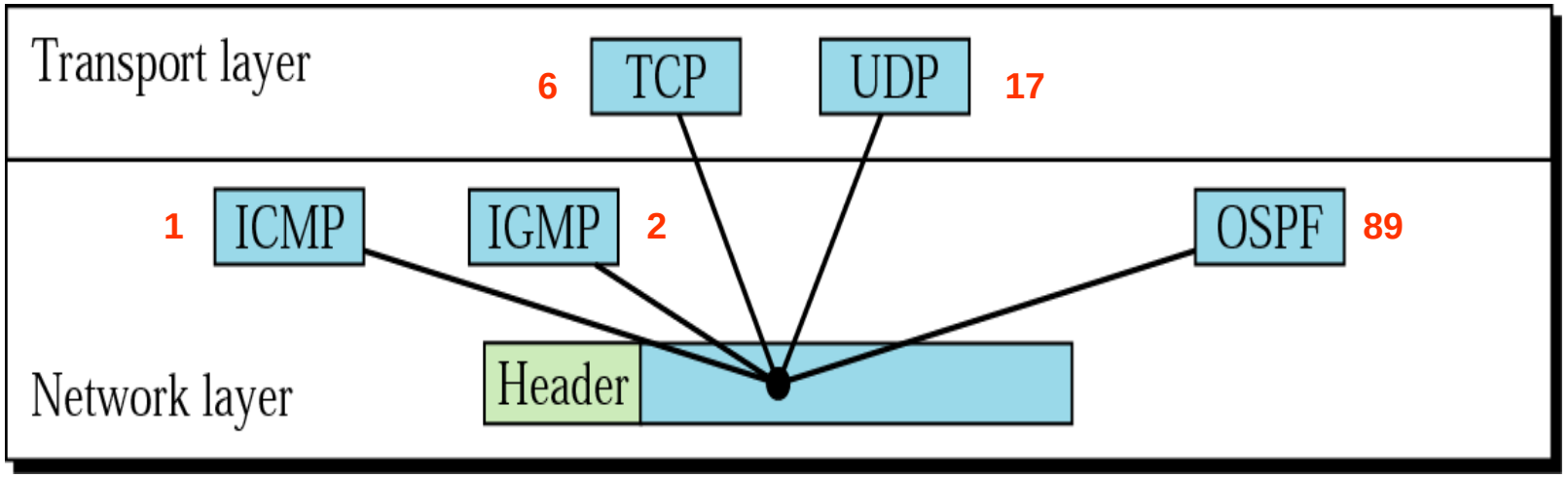
Time to live **: is a seconds counter used to limit packet life times (max. = 255) ; decremented on every hop ; if *TTL* = 0, packet is discarded, warning sent to host**

Protocol **: tells the NL which transport process the completely assembled datagram is to be handed over to ; protocols are universally numbered ; RFC 1700 / www.iana.org (on-line database)**

23.03.2020 17:09
116

Computer Networks

Protocol :



23.03.2020 17:09
117

Computer Networks

Network Layer

The Network Layer in the Internet

IPv4 header fields : (contd.)

Header Checksum **: for error detection ; recomputed at each hop because at least one field (TTL) always changes**

Source Address, Destination Address **: indicate network number and host number**

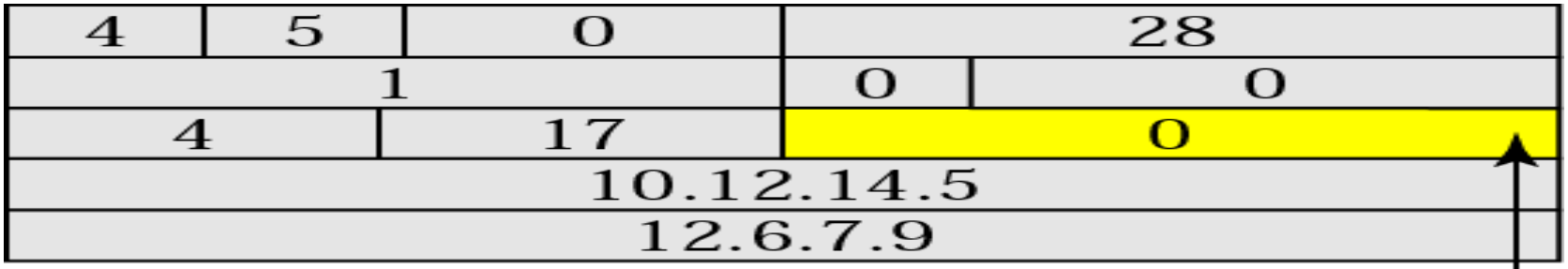
Options **: for expandability, experimentation ; variable length, 1-byte option id (coded), 1-byte option length, one or more data bytes (option field padded to n x 4 bytes)**

23.03.2020 17:09 118

Computer Networks

IP header : checksum - 16 bits : example

(covers header only, not data)



4, 5, and 0	→	0100010100000000
28	→	0000000000011100
1	→	0000000000000001
0 and 0	→	0000000000000000
4 and 17	→	0000010000010001
0	→	0000000000000000
10.12	→	0000101000001100
14.5	→	0000111000000101
12.6	→	0000110000000110
7.9	→	0000011100001001
Sum	→	0111010001001110
Checksum	→	1000101110110001

23.03.2020 17:09
119

Computer Networks
Network Layer
The Network Layer in the Internet
IPv4 header - option fields :

Option	Description
Security	Specifies how secret the datagram is e.g. military use
Strict source routing	Gives the complete path to be followed
Loose source routing	Gives a list of routers not to be missed political / economical
Record route	Makes each router append its IP address for tracking
Timestamp	Makes each router append its address and timestamp

Strict source routing **:**
specifies exact path - used for emergency packets or for measurement purposes

23.03.2020 17:09
120

IPv4 datagram : example

BF4e20.23

An IPv4 datagram has arrived with the following information in the header (in hex)

45 00 00 54 00 03 00 00 20 06 58 50 7C 4E 03 02 B4 0E 0F 02

(a) How can it be ascertained whether the packet is corrupted ?
(b) Are there any options ?
(c) Is the packet fragmented ?
(d) What is the size of the data ?
(e) How many more routers can the packet travel to ?

…contd/-

IPv4 datagram : example

BF4e20.23

….contd

(f) What is the identification number of the packet ?

(g) What is the type of service ?

(h) Where has the packet come from ?

(i) What is the destination address ?

(j) What is the transport layer protocol to which this packet would be delivered to ?

Computer Networks

IPv4 datagram : example

BF4e20.23

VER = 0x4 = 4

HLEN =0x5 = 5 → 5 × 4 = 20

Service =0x00 = 0

Total Length = 0x0054 = 84

Identification = 0x0003 = 3

Flags and Fragmentation = 0x0000 →

D = 0; M = 0; Offset = 0

Time to live = 0x20 = 32

Protocol = 0x06 = 6

Checksum = 0x5850

Source Address: 0x7C4E0302 = 124.78.3.2

Destination Address: 0xB40E0F02 = 180.14.15.2

IPv4 datagram : example

BF4e20.23

(a) checksum is calculated. If it works out to 0x0000. The packet is not corrupted.

(b) since the length of the header is 20 bytes, there are no options.

(c) since M = 0 and offset = 0, the packet is not fragmented.

(d) the total length is 84. Data size is 64 bytes (84 - 20).

IPv4 datagram : example

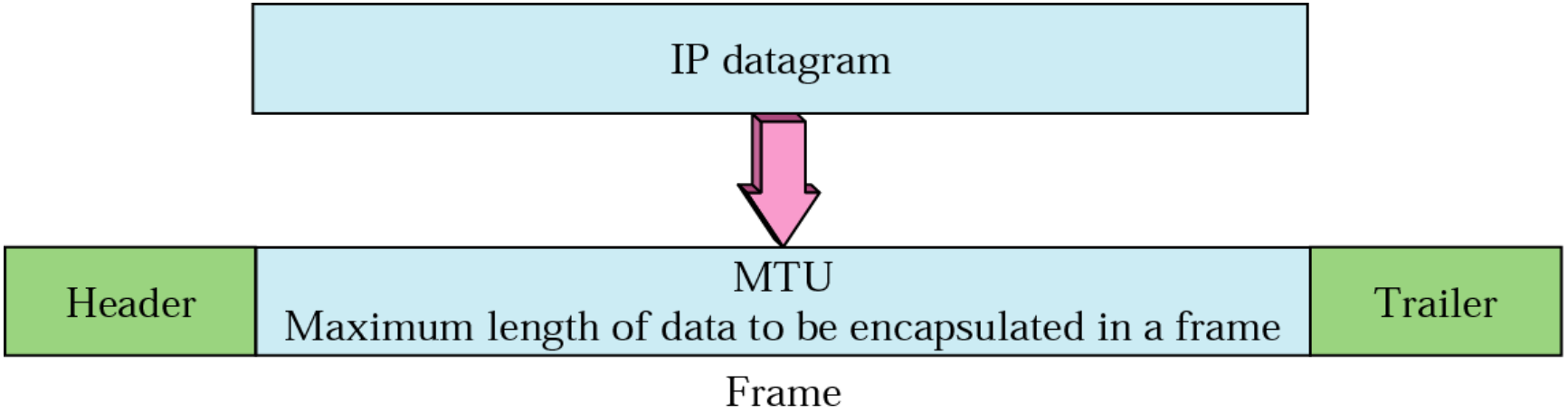
BF4e20.23

(e) since the value of time to live = 32, the packet may visit up to 32 more routers.
(f) the identification number of the packet is 3.
(g) the type of service is normal.
(h) the packet has come from 124.78.3.2
(i) the destination address is 180.14.15.2
(j) since the Protocol field = 6, the transport layer protocol to which this packet would be delivered to is TCP

Computer Networks

Fragmentation and Reassembly

Maximum transfer Unit (MTU)



Ex : Ethernet : 1500 bytes (max.)

FDDI : 4500 bytes (max.)

packets that are too large to be transmitted over a given technology are fragmented and reassembled

23.03.2020 17:09
126

Computer Networks
Network Layer
Internetworking
Maximum transmission unit (MTU)

MTU refers to the size (in bytes) of the largest packet that a given layer of a protocol can pass onwards.

MTU parameters usually appear in association with a NIC, serial port, etc.

The MTU may be fixed by standards (as is the case with Ethernet) or decided at connect time (as is usually the case with point-to-point serial links).

A higher MTU means higher bandwidth efficiency.

However, large packets can block up a slow interface for some time, increasing the lag on other packets. For example, a 1500 byte packet, the largest allowed by Ethernet (and hence most of the Internet), would block up a 14.4 Kbps Modem for about one second.

23.03.2020 17:09 127

Computer Networks
Network Layer
Internetworking

Fragmentation

Packet size in networks are required to be limited by NW designers due to the following

(a) Hardware (size of ethernet frame)
(b) OS (all buffers are xxx bytes)
(c) Protocols (no. of bits in the *packet length* field)
(d) Compliance with some standard
(e) Need to reduce retransmissions
(f) Need to reduce packet dwell time in a channel

23.03.2020 17:09
128

Computer Networks

Network Layer

Internetworking

Fragmentation

through use of proper algorithms, avoid sending large packets which cannot be handled

if destination network is not capable of handling original (large) source packet ?

→ → → Fragmentation of packets

Routers to break up large packets into *fragments* ; to send each fragment as a separate internet packet

23.03.2020 17:09
129

Computer Networks

Fragmentation and Reassembly

when a datagram is fragmented, each fragment has its own header

a fragmented datagram *may* itself be fragmented several times along the route,

initially by the source host or any other router, and ….
if necessary, before it reaches the destination

reassembly is done at the final destination

23.03.2020 17:09
130

Computer Networks

Fragmentation and Reassembly

Fields in IP header

(a) Identification (16 bits) :

- chosen by the sending host
- all fragments have the same identification number, which is also the same as the original datagram
- facilitates reassembly

23.03.2020 17:09
131

Fragmentation and Reassembly

fragmentation typically occurs in a Router
all fragments of a packet carry the same identification field to facilitate reassembly at the receiving host
if all the fragments of a packet do not arrive at the receiving host, the reassembly process is not done
IP does not attempt to recover from missing fragments

Computer Networks

Fragmentation and Reassembly

Fields in IP header

(b) Flag (3 bits) :

- first bit - reserved
- second bit - “do not fragment”
- third bit - “ more fragments”

23.03.2020 17:09
133

Computer Networks
Fragmentation and Reassembly

Fields in IP header

(c) Fragmentation offset (13 bits) : shows relative position of fragment

can represent a sequence of bytes upto 8191

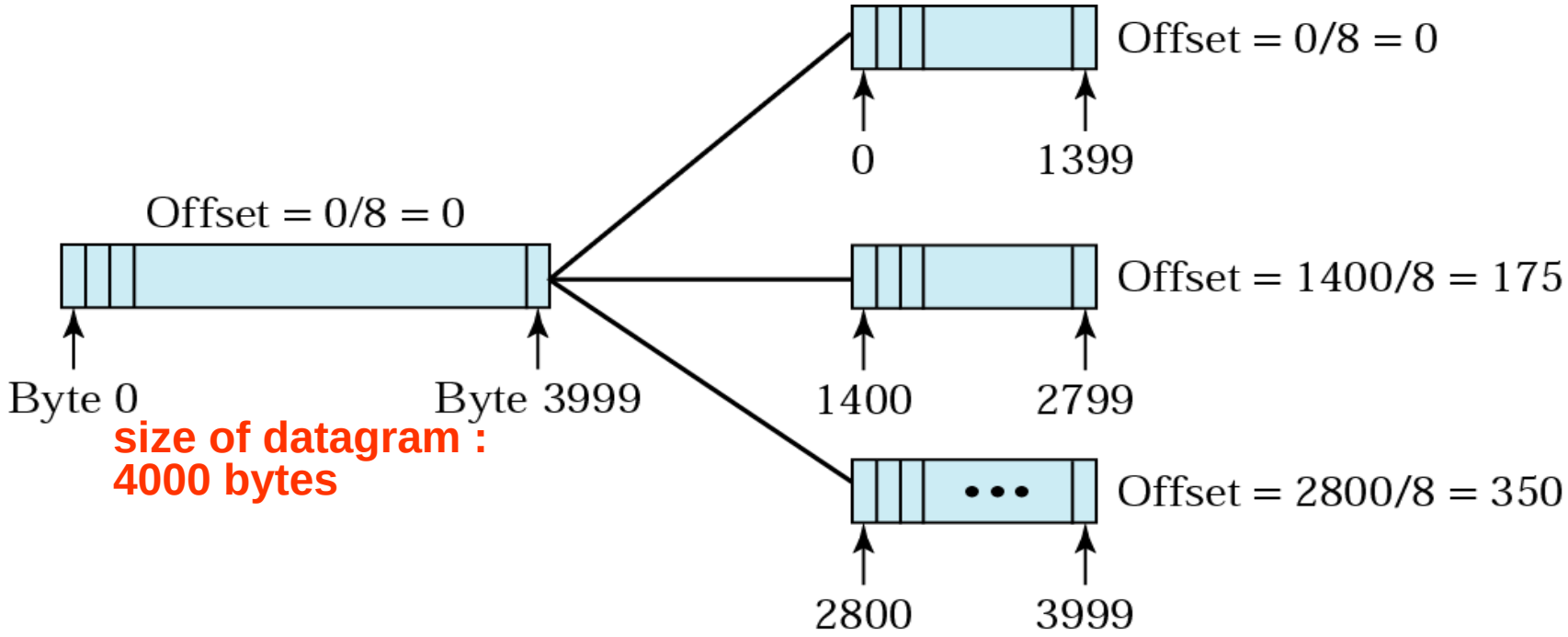
the value of the offset is measured in units of 8 bytes

hosts or routers that fragment datagrams choose a fragment size such that the first byte number is divisible by 8

Computer Networks
Fragmentation and Reassembly

Fields in IP header

Fragmentation offset (13 bits) : shows relative position of fragment



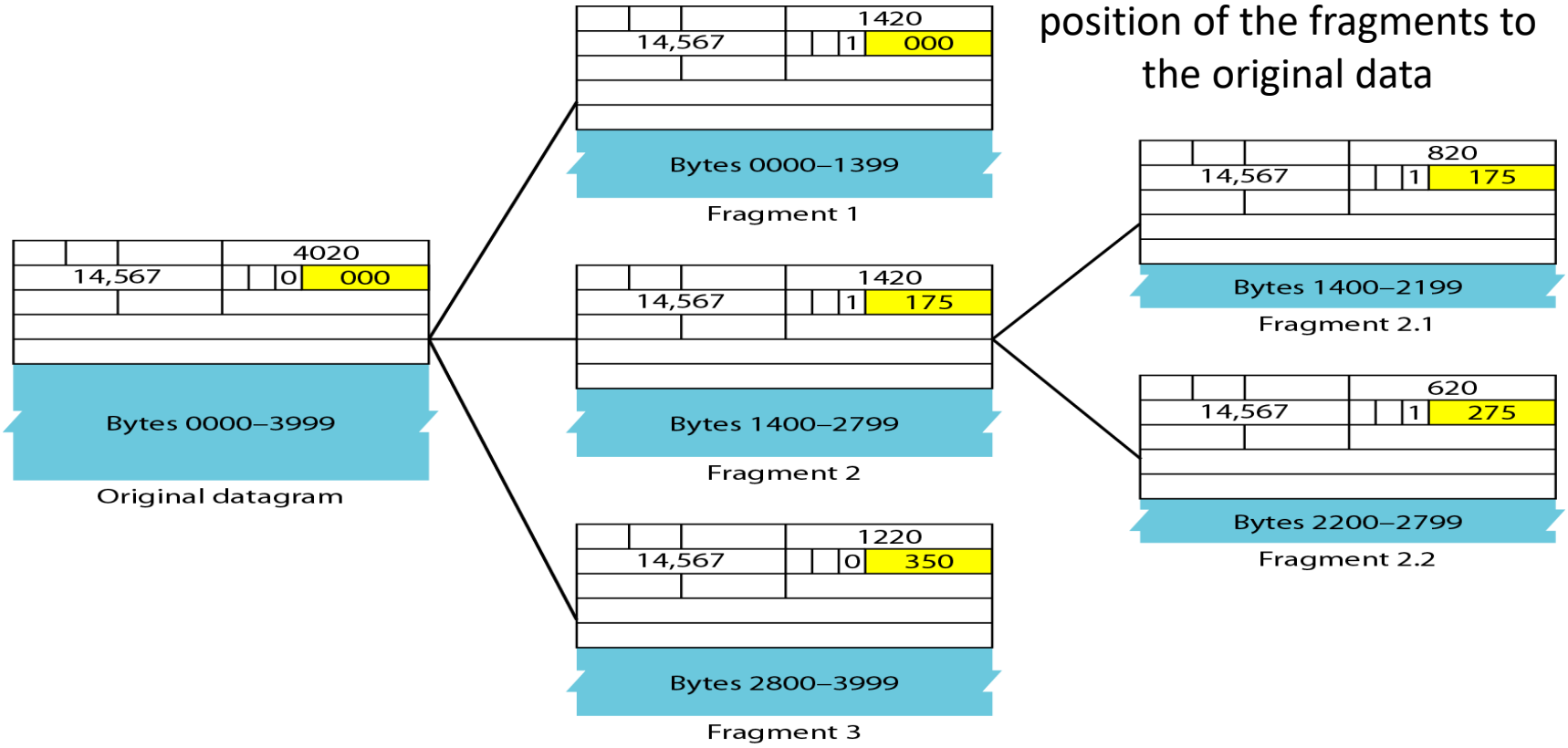
23.03.2020 17:09
135

Computer Networks

Fragmentation and Reassembly

Fields in IP header

Fragmentation example



Computer Networks
Fragmentation and Reassembly

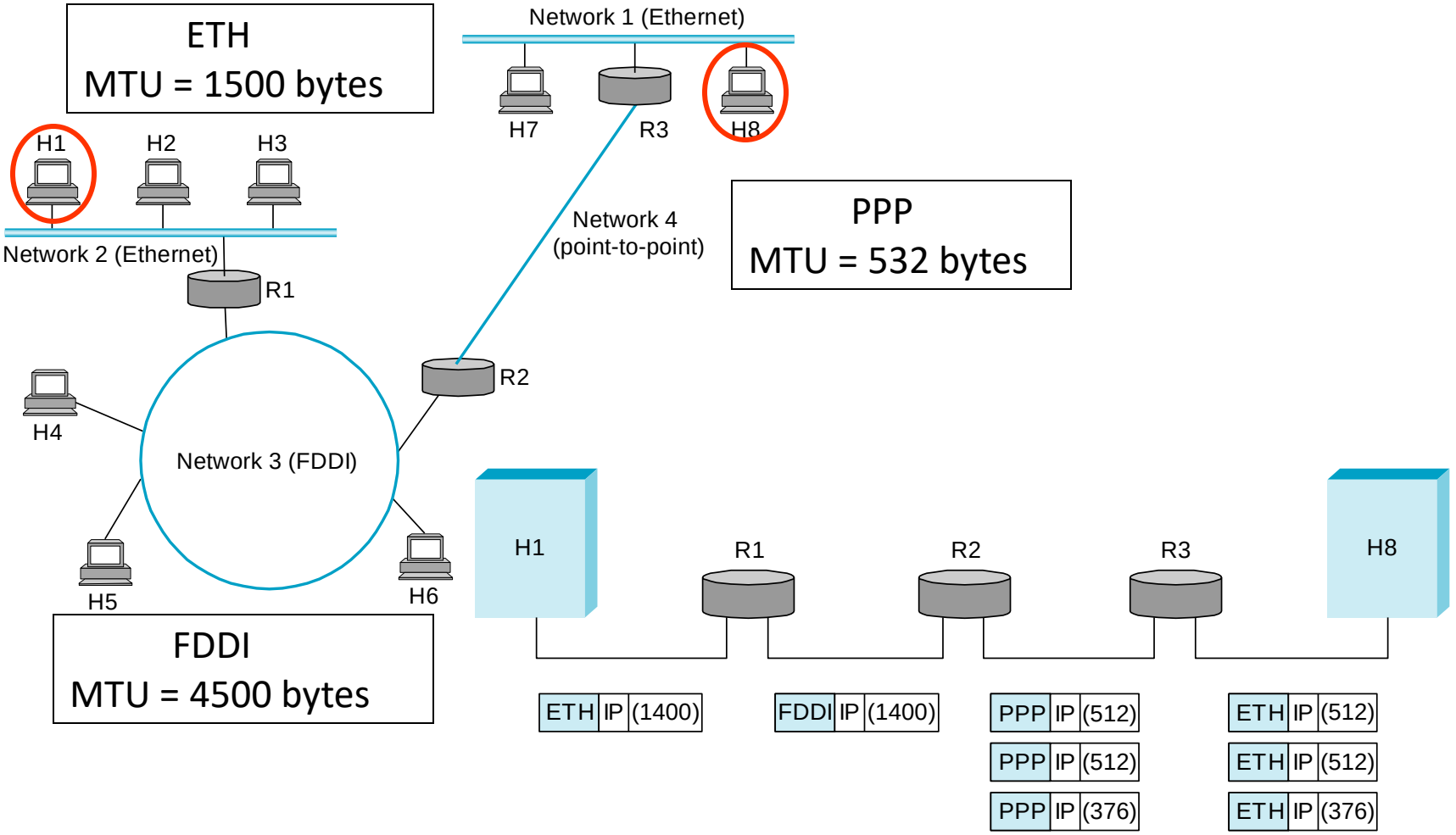
Fields in IP header : fragmentation

final destination host reassembles the original datagram from the fragments received by using the following strategy :

1. **offset value of first fragment = 0**
2. **offset value of second fragment = { length of first fragment ÷ 8 }**
3. **offset value of third fragment = { length of (first + second fragment } ÷ 8)**
4. **last fragment has a *more* bit value of 0**

Computer Networks

Fragmentation and Reassembly

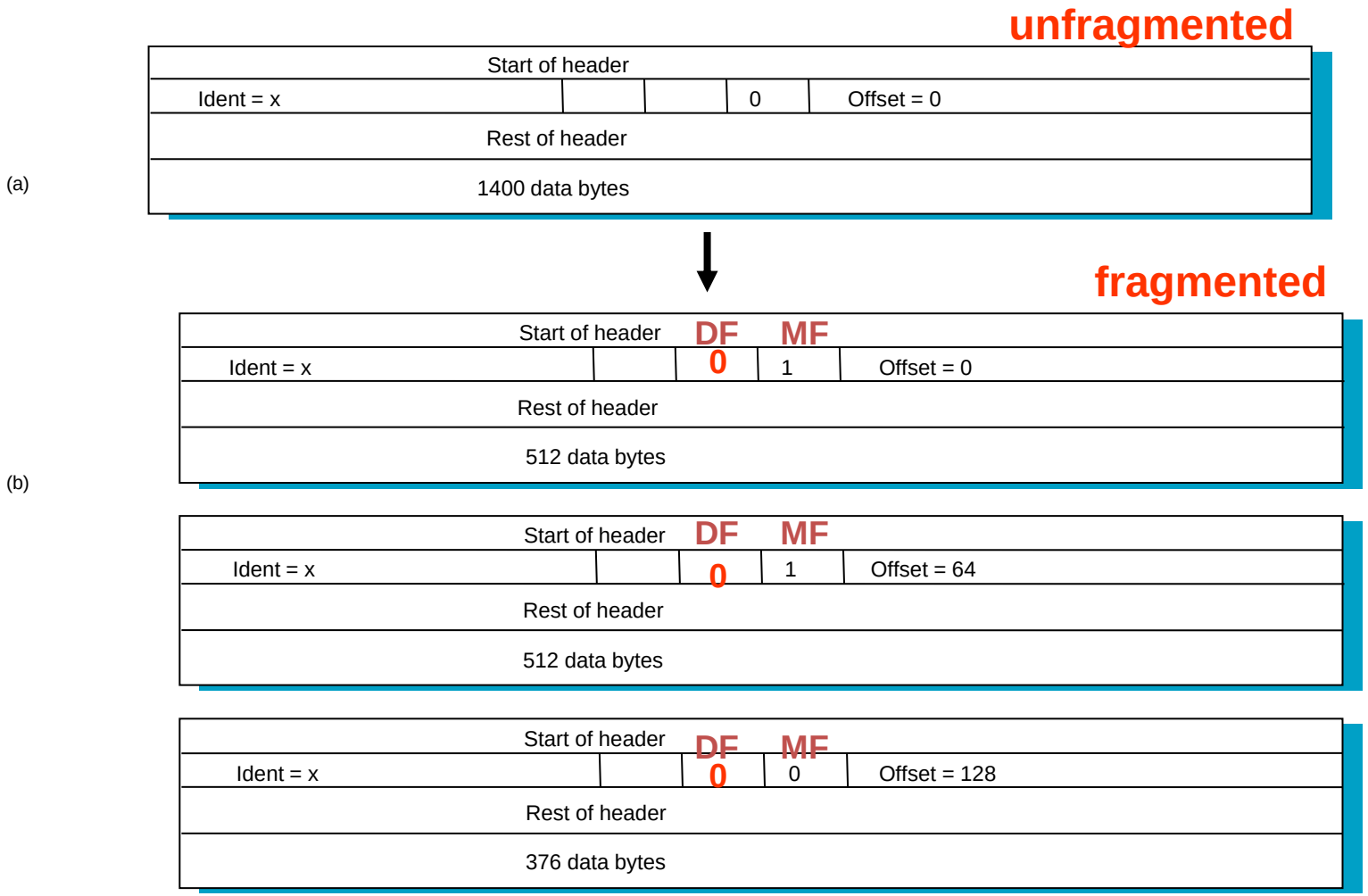


23.03.2020 17:09

138

Computer Networks

Fragmentation and Reassembly



23.03.2020 17:09
139

Computer Networks
Network Layer

Example : **AT 5.36**

An IP datagram using the *strict source routing* option is required to be fragmented.

Is the above *option* required to be copied into each fragment, or is it sufficient to just put it into the first fragment ?

Explain the answer.

23.03.2020 17:09
146

Computer Networks

Example - 1 on IP fragmentation PD 4.4

Suppose a TCP message containing 2048 bytes of data + 20 bytes TCPH is passed to IP for delivery across two networks N1 & N2

N1 : 14-bytes header ; MTU 1024 bytes

N2 : 8-bytes header; MTU 512 bytes

Give the sizes and offsets of the sequence of fragments delivered to the network layer at the destination host

Computer Networks

Example - 1 on IP fragmentation PD 4.4

N1 : MTU = 1024 →
available = 1024-20-14 = 990

for 2068 bytes, fragments are 984 + 984 + 100

N2 : MTU = 512 →
available = 512-20-8 = 484

for 984 bytes, fragments are 480 + 480 + 24

Computer Networks

Example - 2 on IP fragmentation AT 5.34

Suppose that a host A is connected to Router R1, R1 is connected to R2 and R2 to host B.
Suppose that a TCP message that contains 900 bytes of data and 20 bytes of TCP header is passed to the IP code at host A for delivery to B.

Show the *Total length, Id, DF, MF, Offset* fields of the IP header in each packet transmitted over the three links.

Assume that the link A - R1 can support an MTU of 1024 bytes, including a 14-byte frame header, R1 - R2 can support a maximum frame size of 512 bytes, including an 8-byte frame header and R2 - B can support a maximum frame size of 512 bytes including a 12-byte frame header.

23.03.2020 17:09
151

Computer Networks

Example - 2 on IP fragmentation AT 5.34

The initial IP datagram will be fragmented into two IP datagrams at R1

Link A-R1:
Length = 940; ID = x; DF = 0; MF = 0; Offset = 0

Link R1-R2:
F1 : Length = 500; ID = x; DF = 0; MF = 1; Offset = 0
F2 : Length = 460; ID = x; DF = 0; MF = 0; Offset = 60

Link R2-B:
F1 : Length = 500; ID = x; DF = 0; MF = 1; Offset = 0
F2 : Length = 460; ID = x; DF = 0; MF = 0; Offset = 60

23.03.2020 17:09
152