Introduction to Computer Networks

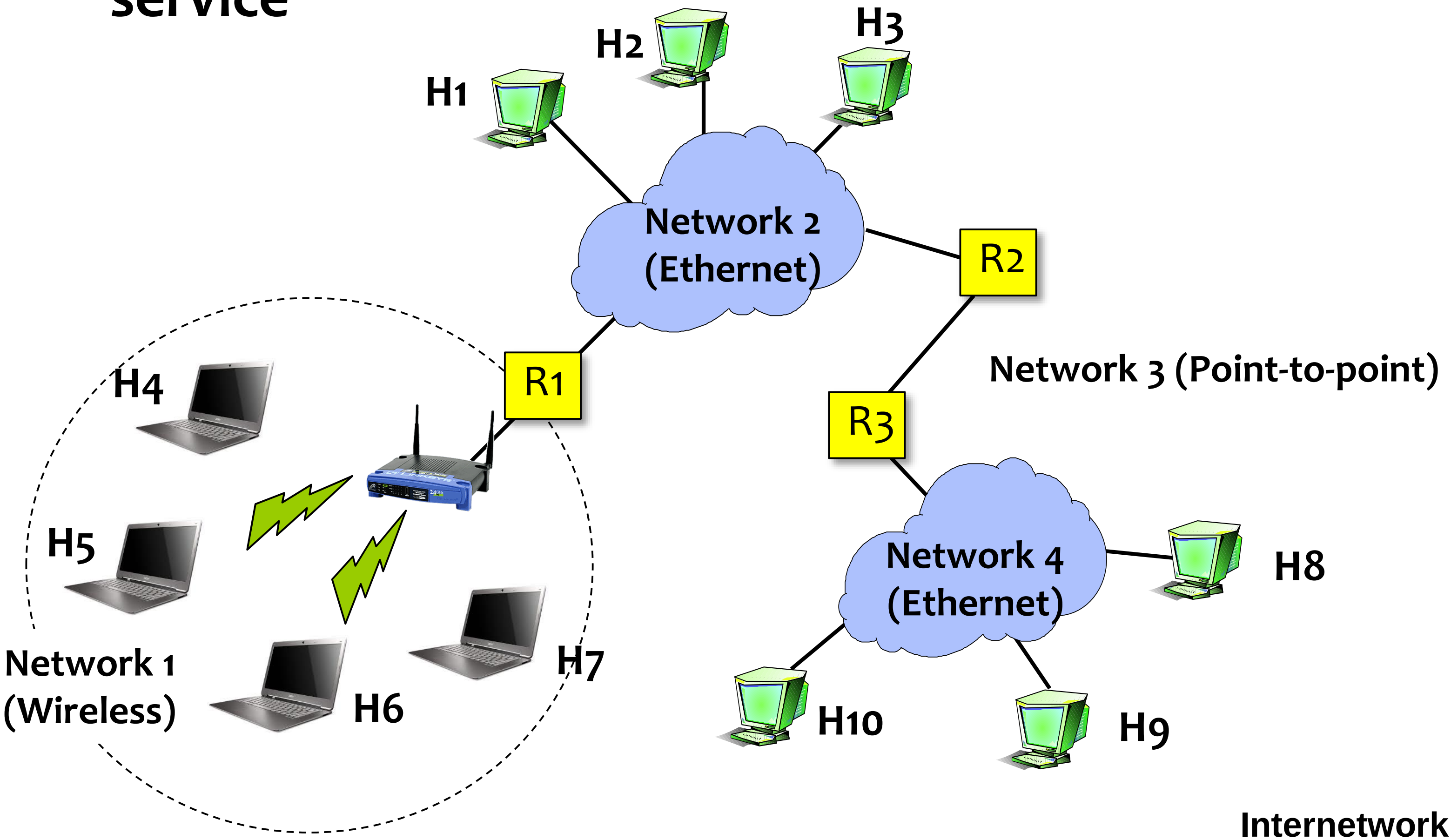
Internetworking

Outline

- Introduction
- IP and Routers
- IP Subnetting
- Classless Addressing
- Routing protocols
- Distance Vector protocol
- Link State protocol

Internetworking

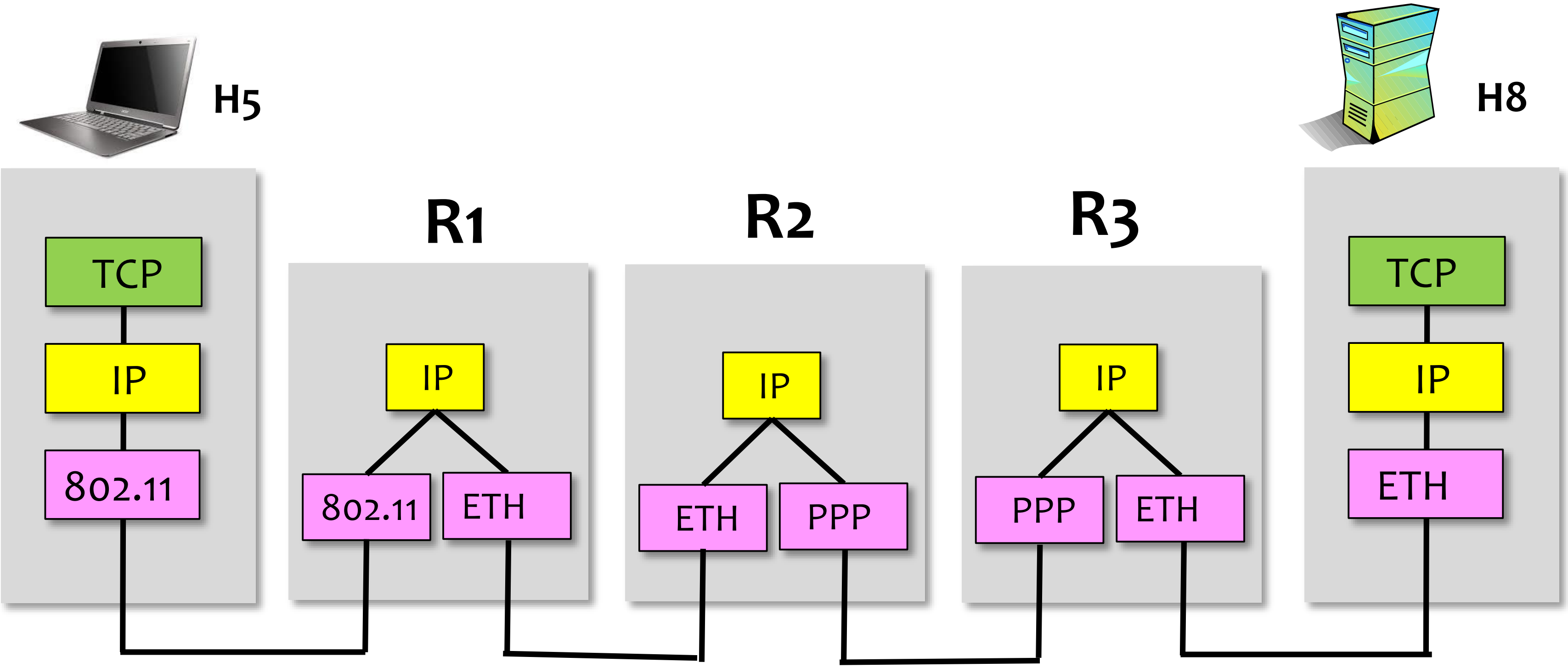
- What is internetwork ?
  - An arbitrary collection of networks interconnected to provide some sort of host-host to packet delivery service



Internetworking

- What is IP ?
  - IP stands for Internet Protocol
  - Key tool used today to build scalable, heterogeneous internetworks
  - It runs on all the nodes in a collection of networks
  - Defines the infrastructure that allows these nodes and networks to function as a single logical internetwork

Internetworking



A simple internetwork showing the protocol layers

IP Service Model

- Packet Delivery Model
  - Connectionless model for data delivery
  - Best-effort delivery (unreliable service)
    - packets are lost
    - packets are delivered out of order
    - duplicate copies of a packet are delivered
    - packets can be delayed for a long time
- Global Addressing Scheme
  - Provides a way to identify all hosts in the network

How Layer 3 Routers Work ?

- Layer 3 router uses store and forward scheme to forward incoming IP packets (datagrams).
  - IP Address Lookup (Forwarding Table constructed by routing protocols, such as RIP, OSPF, BGP, etc)
  - IP/MAC mapping table

IP	MAC
IP(A)	MAC(A)
IP(B)	MAC(B)
IP(Y)	MAC(Y)
IP(X)	MAC(X)

IP	Next
140.114.77.0	Directly
140.114.78.0	Directly
140.114.79.0	Router Z

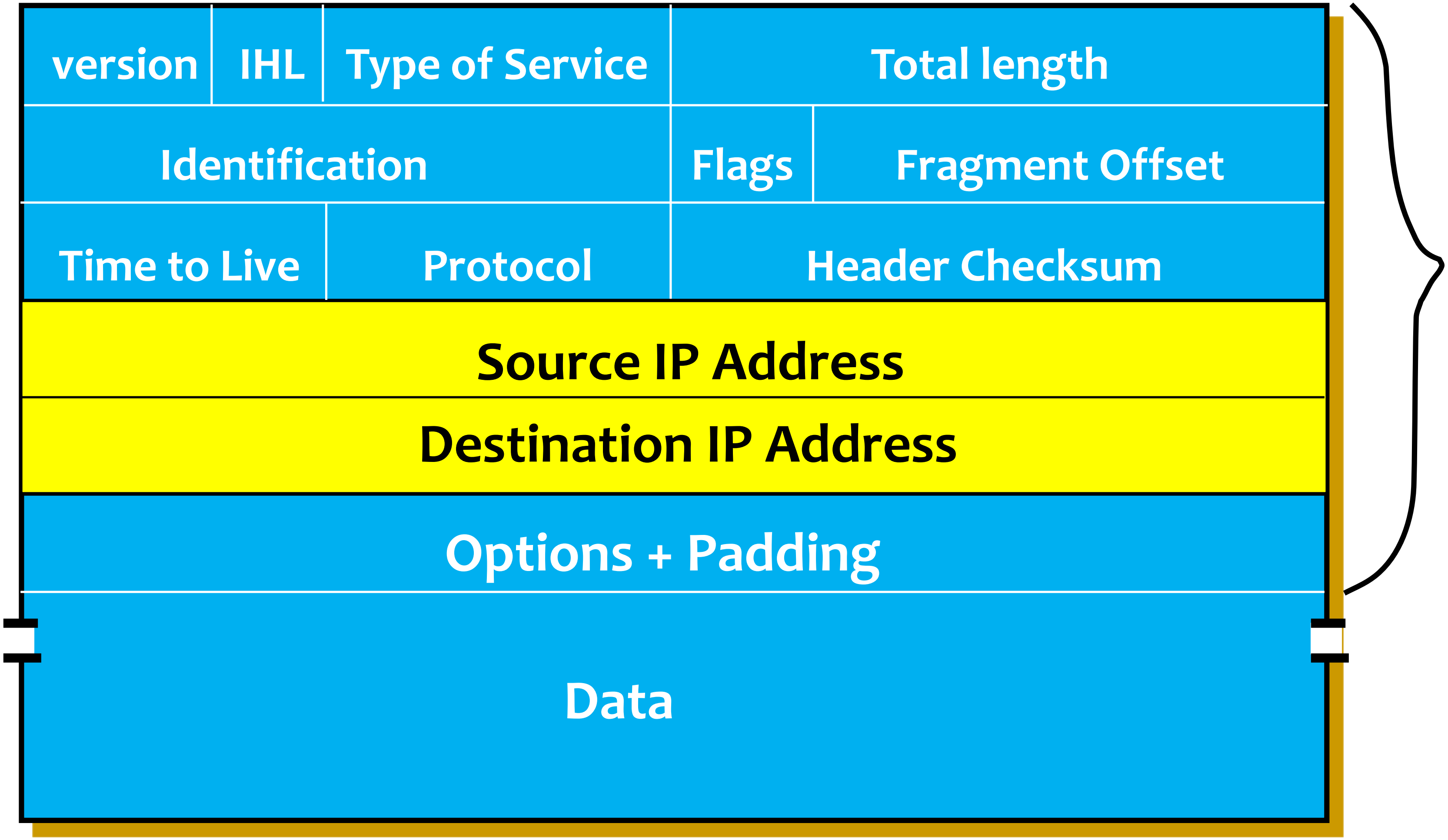
How Layer 3 Routers Work ?

- Forward IP packet into next hop if the destination IP is found in the Forwarding Table. Otherwise, forward to default port.
- New router Architecture with L3 switching Fabric ASICs and IP address lookup ASICs (hardware lookup)
- Wire-speed forwarding design Gbps, 10Gbps, 100Gbps, …
- Not Plug-and-Play

Outline

- Introduction
- IP and Routers
- IP Subnetting
- Classless Addressing
- Routing protocols
- Distance Vector protocol
- Link State protocol

IP Datagram Header Format



version | IHL | Type of Service | Total length

Identification | Flags | Fragment Offset

Time to Live | Protocol | Header Checksum

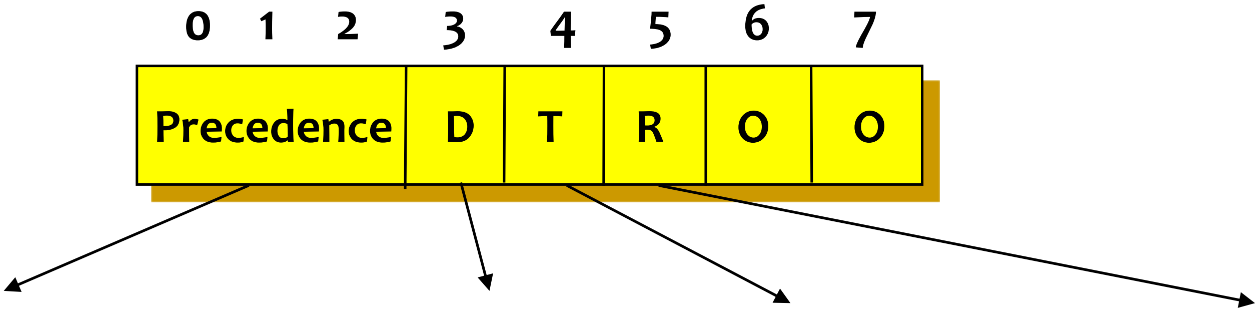
Source IP Address

Destination IP Address

Options + Padding

Data

Type of Service (ToS) of IP



Precedence

111 Network Control

110 Internetwork Control

101 CRITIC/ECP

100 Flash Override

011 Flash

010 Immediate

001 Priority

000 Routine

Delay

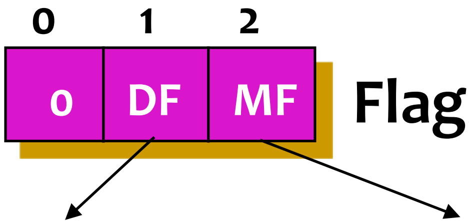
0 Normal

1 Low

Throughput, Reliability

0 Normal

1 High



Flags

DF

0 May Fragment

1 Don't Fragment

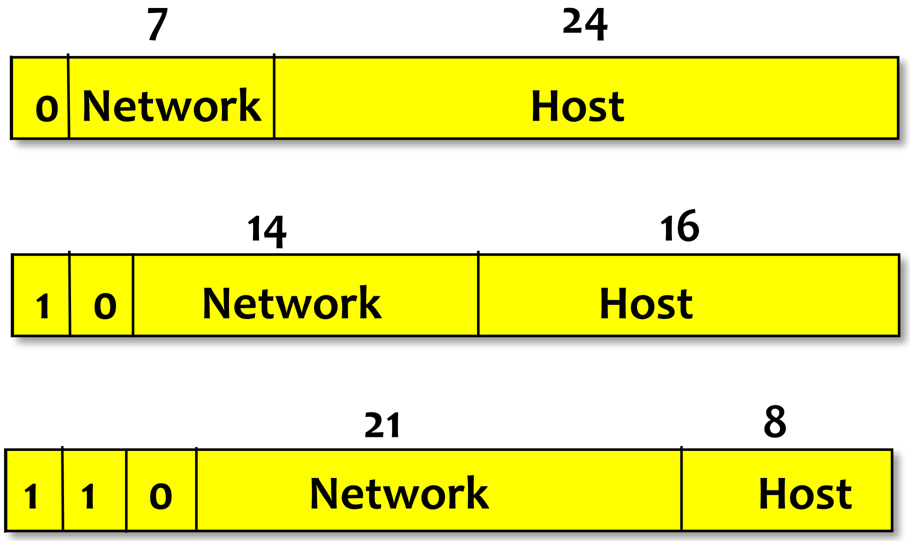
MF

0 Last Fragment

1 More Fragment

IP Addresses

- Properties
  - Globally unique 32 bits address
  - Hierarchical: network + host
  - 4 Billion IP addresses
  - Class A type (1/2)
  - Class B type (¼)
  - Class C type (1/8)
- Dot notation
  - 10.3.2.4
  - 128.96.33.81
  - 192.12.69.77

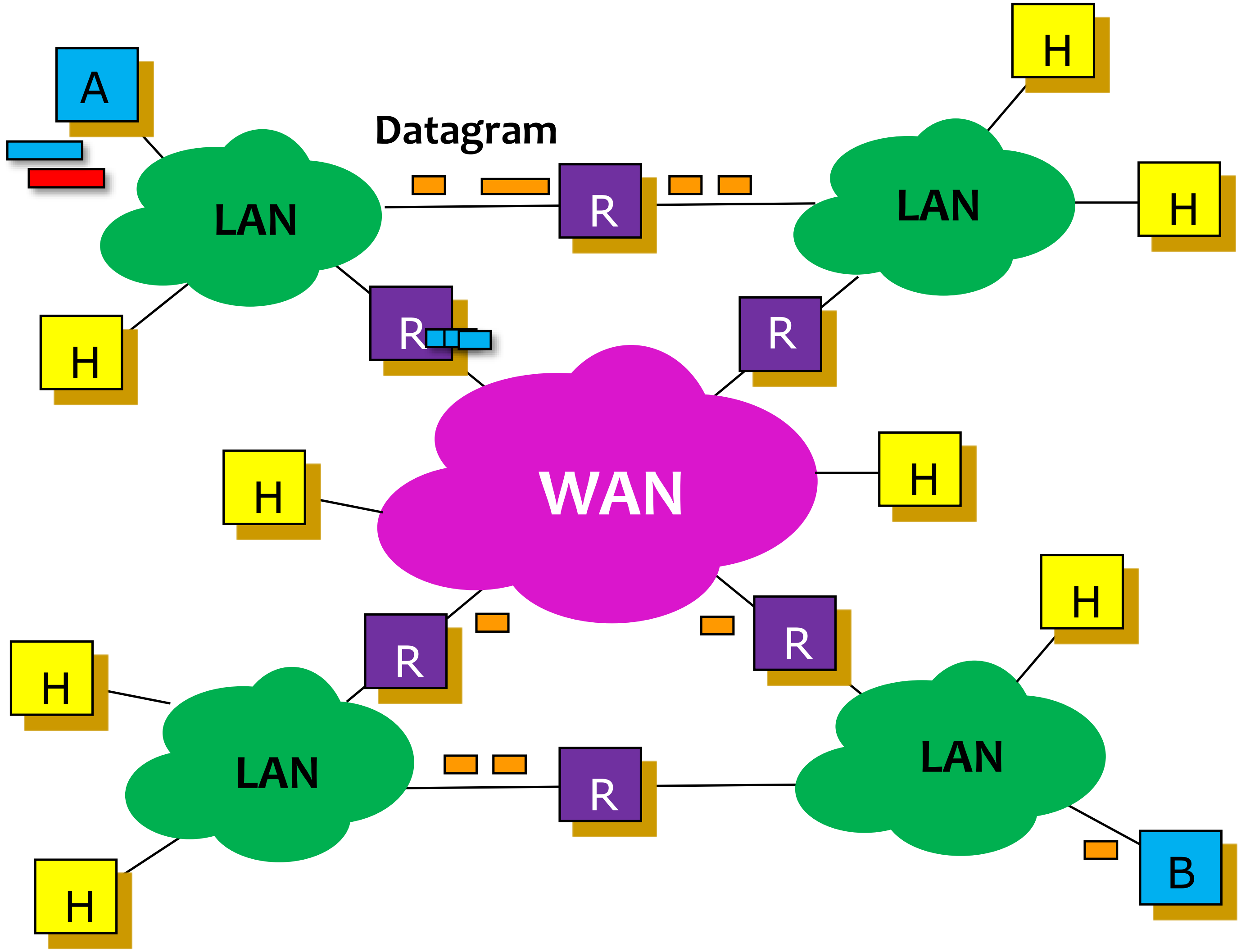


	7	24
0	Network	Host

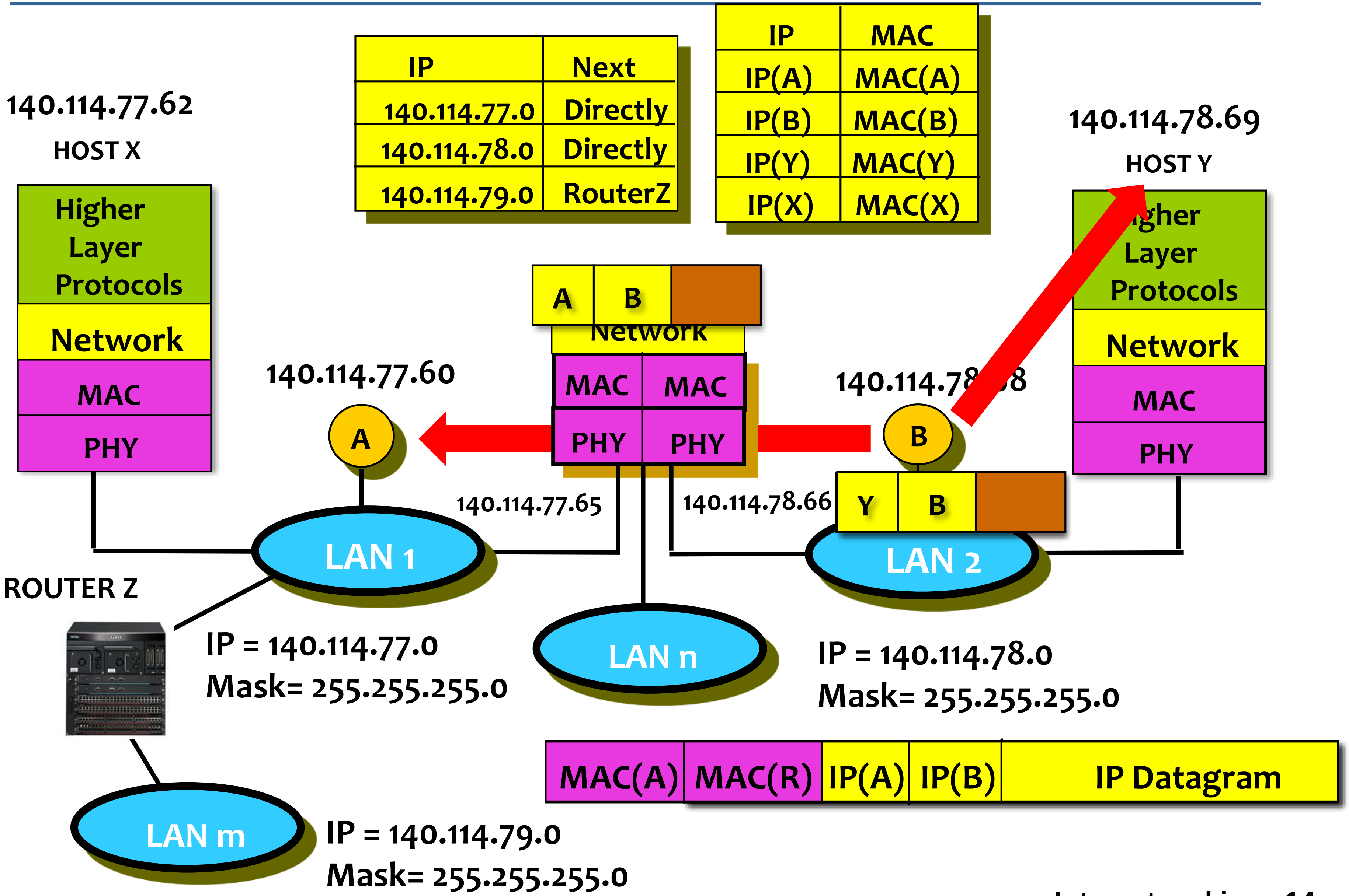
		14	16
1	0	Network	Host

			21	8
1	1	0	Network	Host

How datagrams are delivered in an Internet ?



Routers

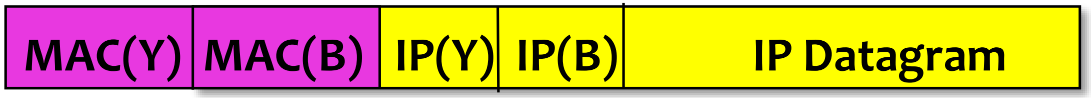


IP	Next
140.114.77.0	Directly
140.114.78.0	Directly
140.114.79.0	RouterZ

IP	MAC
IP(A)	MAC(A)
IP(B)	MAC(B)
IP(Y)	MAC(Y)
IP(X)	MAC(X)

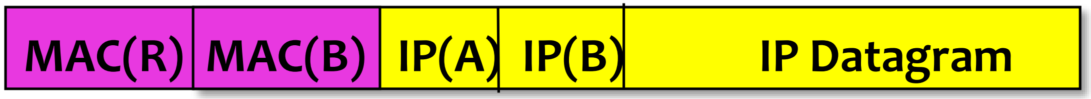
Intra-LAN and Inter-LAN Communications

- B -> Y (Intra LAN):
  - Send the frame to the destination directly

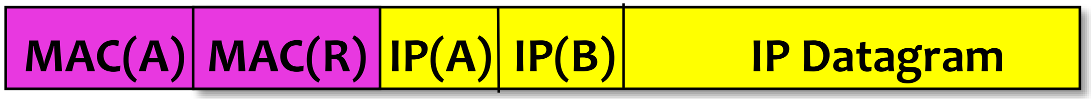


MAC(Y)	MAC(B)	IP(Y)	IP(B)	IP Datagram

- B -> A (Inter-LAN):
  - Send the frame to attached Router first.
  - Router will forward to the destination.

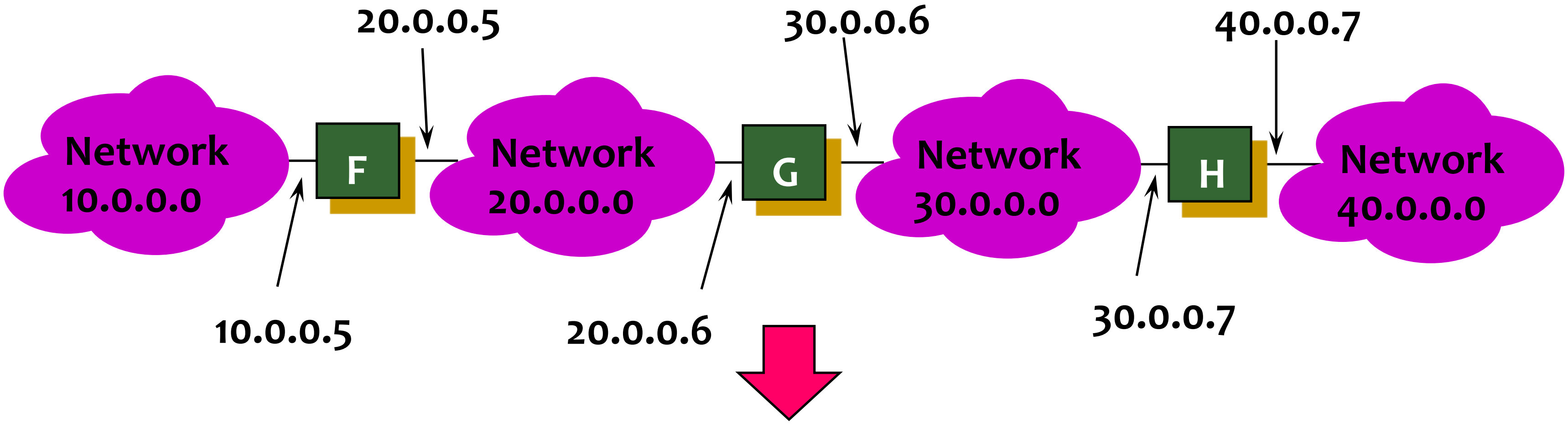


MAC(R)	MAC(B)	IP(A)	IP(B)	IP Datagram



MAC(A)	MAC(R)	IP(A)	IP(B)	IP Datagram

An Internet Routing Example



To reach hosts on network	Route to this address
20.0.0.0	Deliver Direct
30.0.0.0	Deliver Direct
10.0.0.0	20.0.0.5
40.0.0.0	30.0.0.7

- Routing Table

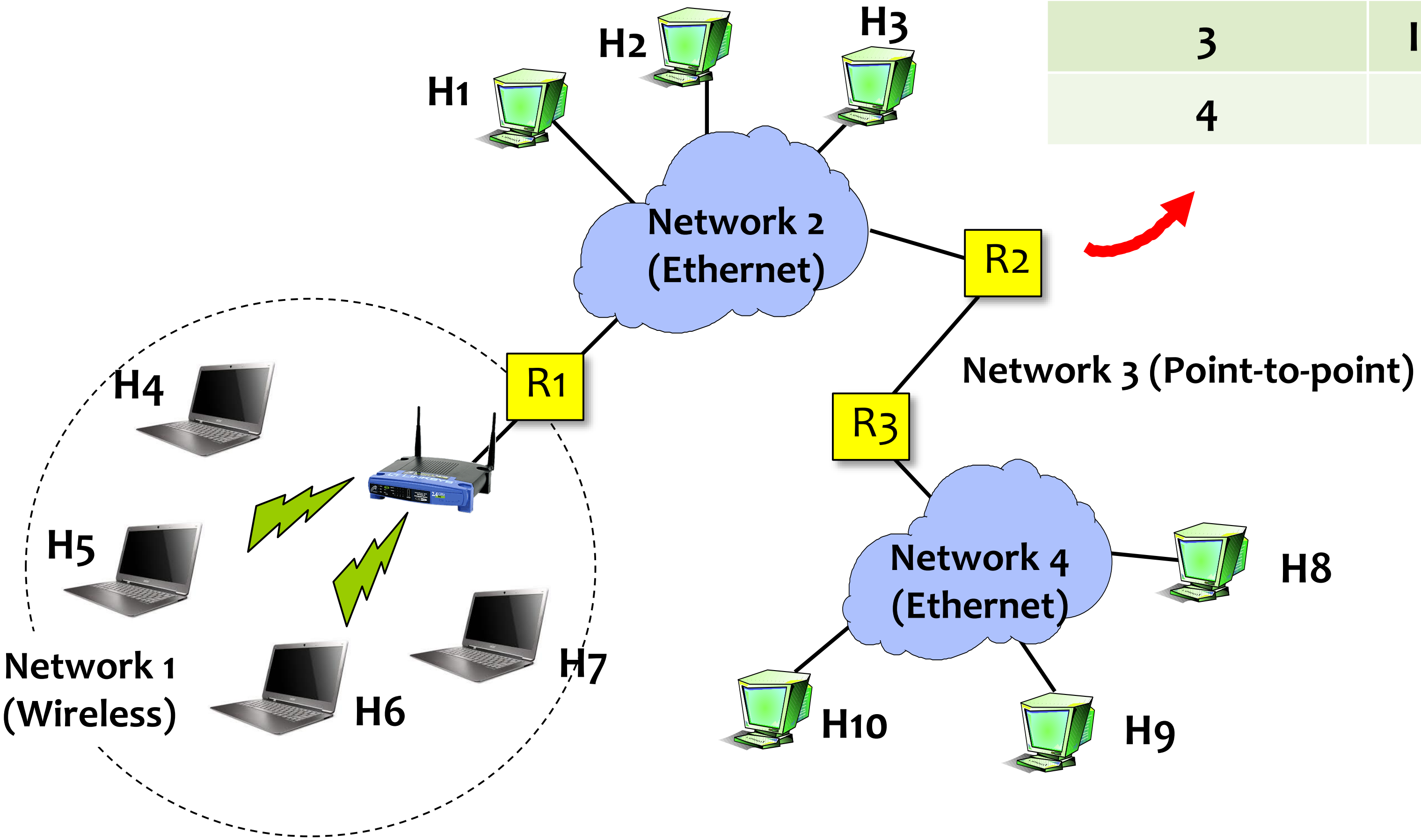
IP Datagram Forwarding

- Strategy
  - every datagram contains destination's address
  - if directly connected to destination network, then forward to host
  - if not directly connected to destination network, then forward to some router
  - forwarding table maps network number into next hop
  - each host has a default router
  - each router maintains a forwarding table

IP Datagram Forwarding

- Example (router R2)

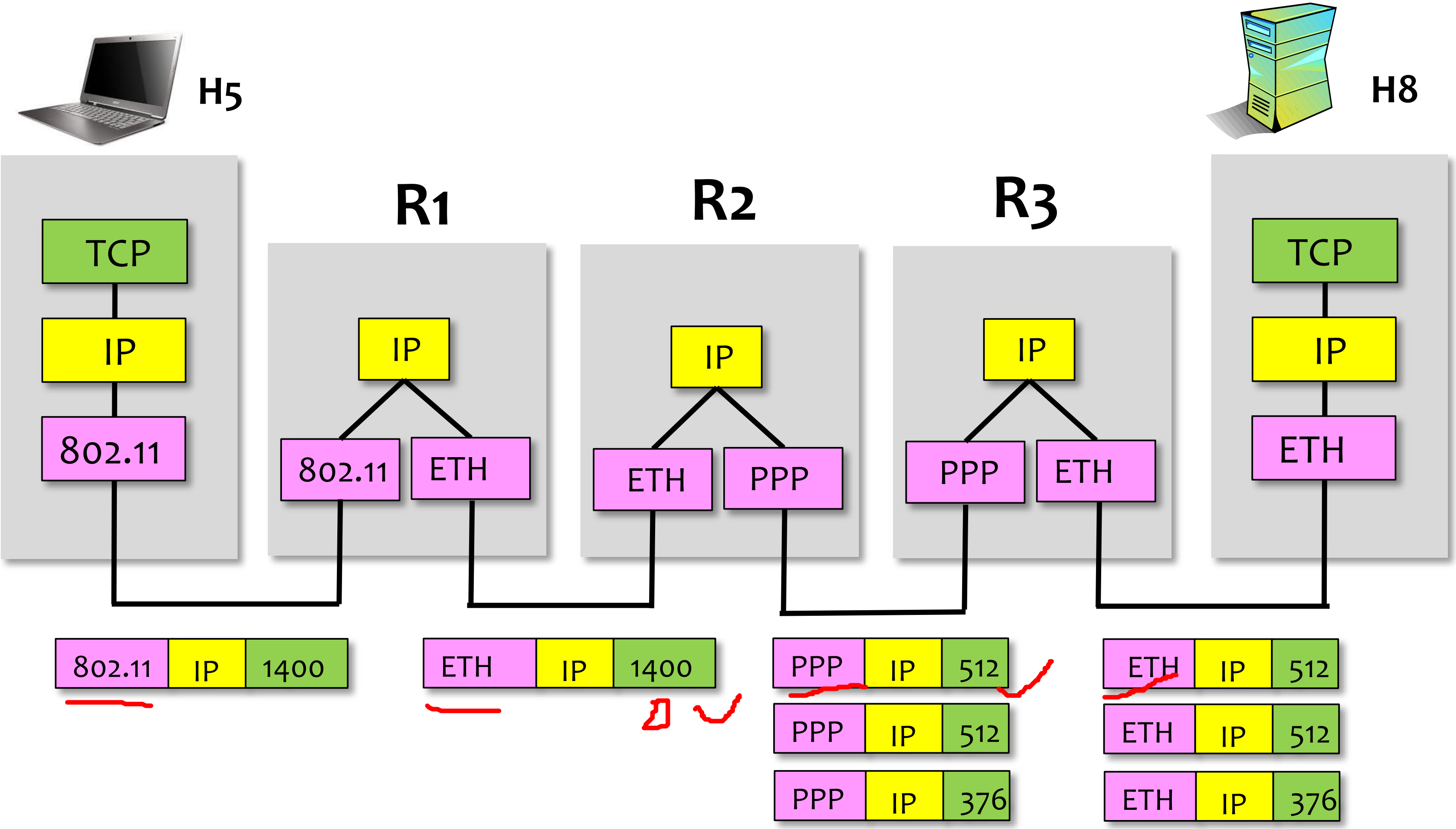
NetworkNum	NextHop
1	R1
2	Interface 1
3	Interface 0
4	R3



IP Fragmentation and Reassembly

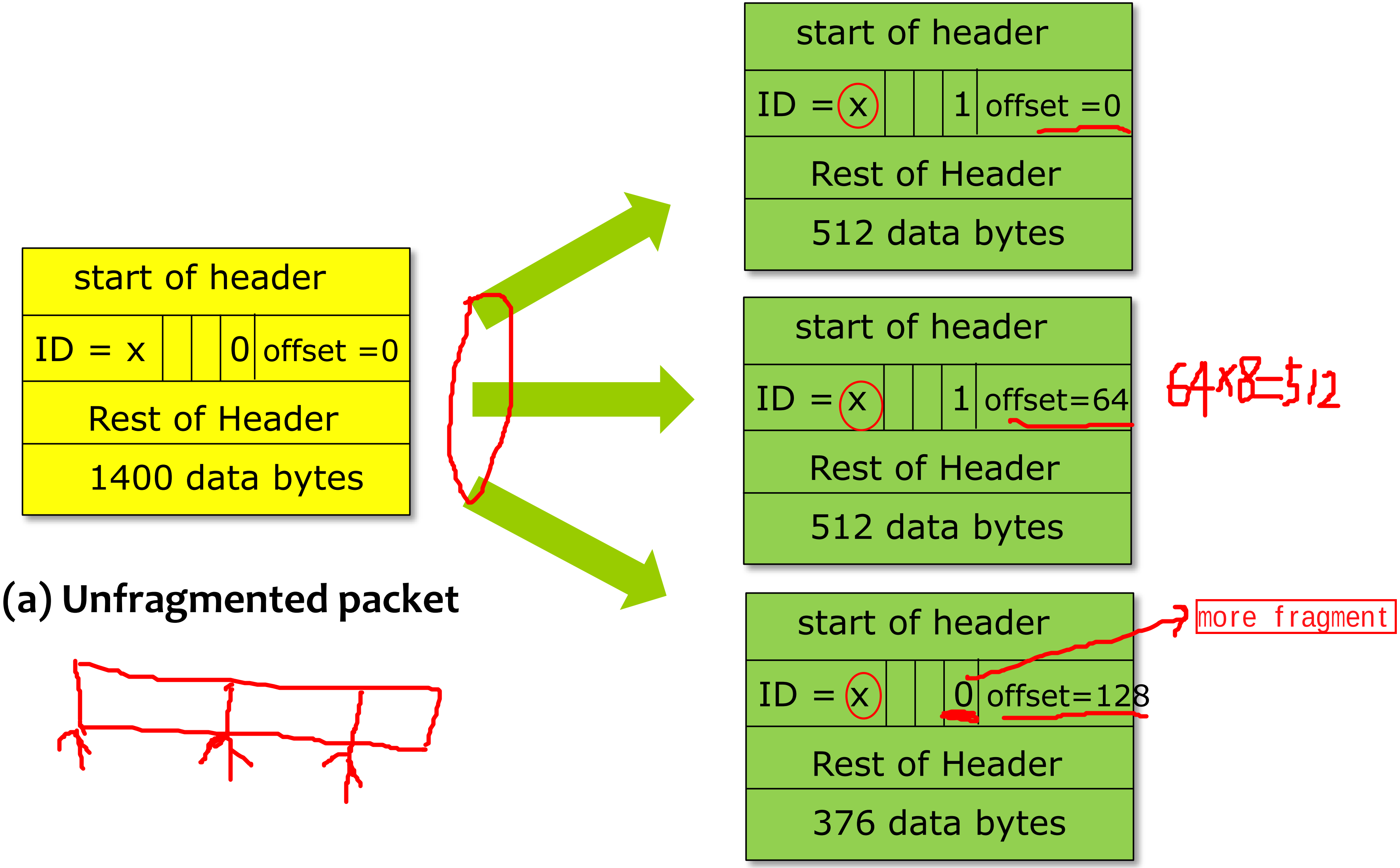
- MTU (Maximum Transmission Unit)
  - Ethernet (1518 bytes),
  - IEEE 802.11 Wireless (2312 bytes)
  - FDDI (4500 bytes)
- Strategy
  - Fragmentation occurs in a router when it receives a datagram that it wants to forward over a network which has MTU < datagram 封包的size > 那个网络的MTU
  - Reassembly is done at the receiving host
  - All the fragments carry the same identifier
  - Fragments are self-contained datagrams
  - IP does not recover from missing fragments

IP Fragmentation and Reassembly



IP datagrams traversing the sequence of physical networks

IP Fragmentation and Reassembly



start of header

ID = x | | | 0 | offset =0

Rest of Header

1400 data bytes

(a) Unfragmented packet

start of header

ID = x | | | 1 | offset =0

Rest of Header

512 data bytes

start of header

ID = x | | | 1 | offset=64

Rest of Header

512 data bytes

64×8=512

start of header

ID = x | | | 0 | offset=128

Rest of Header

376 data bytes

more fragment

•

(b) fragmented packets

Router Characteristics

Network Layer Routing

- Network layer protocol dependent
- Filter MAC broadcast and multicast packets
- Easy to support mixed media
- Packet fragmentation and reassembly
- Filtering on network (IP) addresses and information
- Accounting

Direct Communication Between Endpoints and Routers

- Highly configurable and hard to get right
- Handle speed mismatch
- Congestion control and avoidance

Router Characteristics (Continued)

- Routing Protocols
  - Interconnect layer 3 networks and exploit arbitrary topologies
  - Determine which route to take
  - Static routing
  - Dynamic routing protocol support
    - RIP: Routing Information Protocol
    - OSPF: Open Shortest Path First
  - Provides reliability with alternate routes
- Router Management
  - Troubleshooting capabilities

Differences Between Bridges and Routers

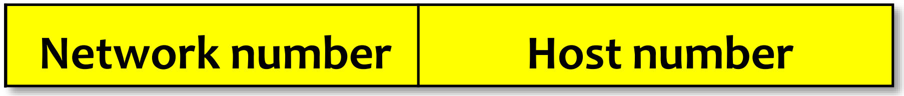
Bridges	Routers
Operation at Layer 2	Operation at Layer 3
Protocol Independent	Protocol Dependent
Automatic Address Learning/Filtering	Administration Required for Address,Interface and Routes
Pass MAC Multicast/Broadcast	MAC M/B can be Filtered
Lower Cost	Higher Cost
No Flow/Congestion Control	Flow/Congestion Control
Limited Security	Complex Security
Transparent to End Systems	Non-Transparency
Well Suited for Simple/Small Networks	For WAN, Larger Networks
No Frames Segmentation/Reassembly	IP packets Segmentation/Reassembly
Spanning Tree Based Routing	Optimal Routing and Load Sharing
Plug and Play	Requires Central Administrator

Outline

- Introduction
- IP and Routers
- IP Subnetting
- Classless Addressing
- Routing protocols
- Distance Vector protocol
- Link State protocol

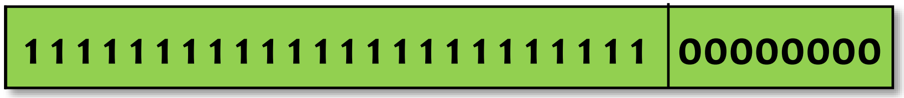
Subnetting

- Add another level to address/routing hierarchy: *subnet*
- *Subnet masks* define variable partition of host part of class A and B addresses



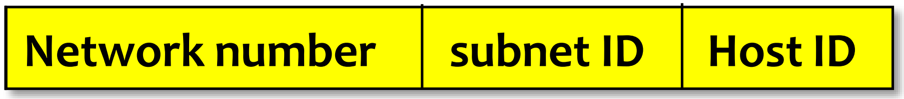
Network number | Host number

Class B address



1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 | 00000000

Subnet Mask (255.255.255.0)



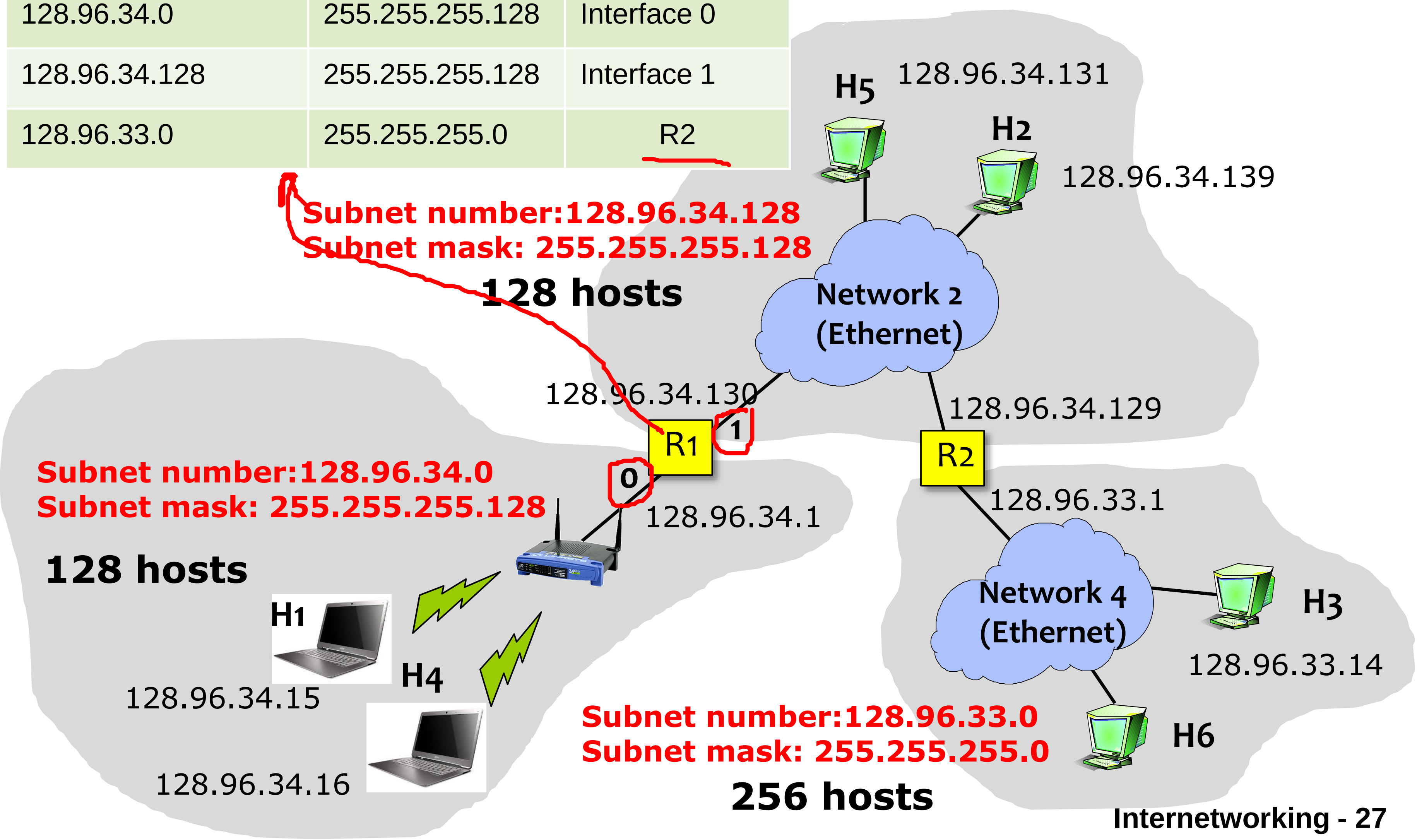
Network number | subnet ID | Host ID

Subnetted address

Subnetting

Forwarding Table at Router R1

SubnetNumber	SubnetMask	NextHop
128.96.34.0	255.255.255.128	Interface 0
128.96.34.128	255.255.255.128	Interface 1
128.96.33.0	255.255.255.0	R2



Subnet number:128.96.34.128
Subnet mask: 255.255.255.128
128 hosts

Subnet number:128.96.34.0
Subnet mask: 255.255.255.128
128 hosts

Subnet number:128.96.33.0
Subnet mask: 255.255.255.0
256 hosts

Subnetting

Forwarding Algorithm

R1

SubnetNumber	SubnetMask	NextHop
128.96.34.0	255.255.255.128	Interface 0
128.96.34.128	255.255.255.128	Interface 1
128.96.33.0	255.255.255.0	R2

```
D = destination IP address
for each entry < SubnetNum, SubnetMask, NextHop>
    D1 = (SubnetMask) AND (D)
    if D1 = SubnetNum
        if NextHop is an interface
          deliver datagram directly to destination
        else
          deliver datagram to NextHop (a router)
```

Subnetting

Example 1

Forwarding Table at Router R1

SubnetNumber	SubnetMask	NextHop
128.96.34.0	255.255.255.128	Interface 0
128.96.34.128	255.255.255.128	Interface 1
128.96.33.0	255.255.255.0	R2

D = 128.96.34.15 (H1)

D1 = SubnetMask & D = 255.255.255.128 1000 0000

128. 96. 34. 15 0000 1 1 1 1

---

128. 96. 34. 0 0000 0000

D1 = SubnetNum 128. 96. 34. 0 → Interface 0

Subnetting

Example2

Forwarding Table at Router R1

SubnetNumber	SubnetMask	NextHop
128.96.34.0	255.255.255.128	Interface 0
128.96.34.128	255.255.255.128	Interface 1
128.96.33.0	255.255.255.0	R2

D = 128.96.34.131 (H5)

D1 = SubnetMask & D = 255.255.255.128   1000 0000

128. 96. 34. 131   1000 0011

128. 96. 34. 128   1000 0000

D1 = SubnetNum 128. 96. 34. 128 → Interface 1

Subnetting

Example 3

Forwarding Table at Router R1

SubnetNumber	SubnetMask	NextHop
128.96.34.0	255.255.255.128	Interface 0
128.96.34.128	255.255.255.128	Interface 1
128.96.33.0	255.255.255.0	R2

D = 128.96.33.14 (H3)

D1 = SubnetMask & D = 255.255.255.0 0000 0000

128. 96. 33. 14 0000 1110

128. 96. 33. 0 0000 0000

D1 = SubnetNum 128. 96. 33. 0 → R2

Subnetting

Notes

- A default router is used if nothing matches
- Not necessary for all ones in subnet mask to be contiguous
- Can put multiple subnets on one physical network
- Subnets not visible from the rest of the Internet

Outline

- Introduction
- IP and Routers
- IP Subnetting
- Classless Addressing
- Routing protocols
- Distance Vector protocol
- Link State protocol

Classless Addressing

- Classless Inter-Domain Routing (CIDR)
  - A technique that addresses two scaling concerns in the Internet
    - The growth of backbone routing table as more and more network numbers need to be stored in them
    - Potential exhaustion of the 32-bit address space

Classless Addressing

- Address assignment efficiency
  - Arises because of the IP address structure with class A, B, and C addresses
  - Forces us to hand out network address space in fixed-size chunks of three very different sizes
    - A network with two hosts needs a class C address
      - Address assignment efficiency = 2/255 = 0.78
    - A network with 256 hosts needs a class B address
      - Address assignment efficiency = 256/65535 = 0.39

Classless Addressing

- Exhaustion of IP address space centers on exhaustion of the class B network numbers
- Solution
  - Say “NO” to any Autonomous System (AS) that requests a class B address unless they can show a need close to 64K addresses
  - Instead give them an appropriate number of class C addresses
- What is the problem with this solution?
  - Large storage requirement for routing table at the routers.

Classless Addressing

- For example, if a single AS has 16 class C network numbers
  - Every Internet backbone router needs 16 entries in its routing tables for that AS even if the path to every one of these networks is the same
- If we had assigned a class B address to the AS
  - The same routing information can be stored in one entry
  - But Efficiency = $16 \times 255 / 65,536 = 6.2\%$

Classless Addressing

- CIDR uses aggregate routes
  - Uses a single entry in the forwarding table to tell the router how to reach a lot of different networks
  - Breaks the rigid boundaries between address classes

Classless Addressing

- For example, an AS with 16 class C network numbers
- Instead of handing out 16 addresses at random, hand out a block of contiguous class C addresses
- Suppose we assign the class C network numbers from 192.4.16 through 192.4.31
- Observe that top 20 bits of all the addresses in this range are the same (11000000 00000100 0001)
  - We have created a 20-bit network number (which is in between class B network number and class C number)
- Requires to hand out blocks of class C addresses that share a common prefix

Classless Addressing

192.4.16	1100 0000	0000 0100	0001	0000
192.4.17	1100 0000	0000 0100	0001	0001
192.4.18	1100 0000	0000 0100	0001	0010
192.4.19	1100 0000	0000 0100	0001	0011
192.4.20	1100 0000	0000 0100	0001	0100
192.4.21	1100 0000	0000 0100	0001	0101
192.4.22	1100 0000	0000 0100	0001	0110
192.4.23	1100 0000	0000 0100	0001	0111
192.4.24	1100 0000	0000 0100	0001	1000
192.4.25	1100 0000	0000 0100	0001	1001
192.4.26	1100 0000	0000 0100	0001	1010
192.4.27	1100 0000	0000 0100	0001	1011
192.4.28	1100 0000	0000 0100	0001	1100
192.4.29	1100 0000	0000 0100	0001	1101
192.4.30	1100 0000	0000 0100	0001	1110
192.4.31	1100 0000	0000 0100	0001	1111

192.4.16/20

Classless Addressing

- The convention is to place a “/X” after the prefix where X is the prefix length in bits
- For example, the 20-bit prefix for all the networks 192.4.16 through 192.4.31 is represented as 192.4.16/20
- By contrast, if we wanted to represent a single class C network number 192.4.16, which is 24 bits long, we would write it 192.4.16/24

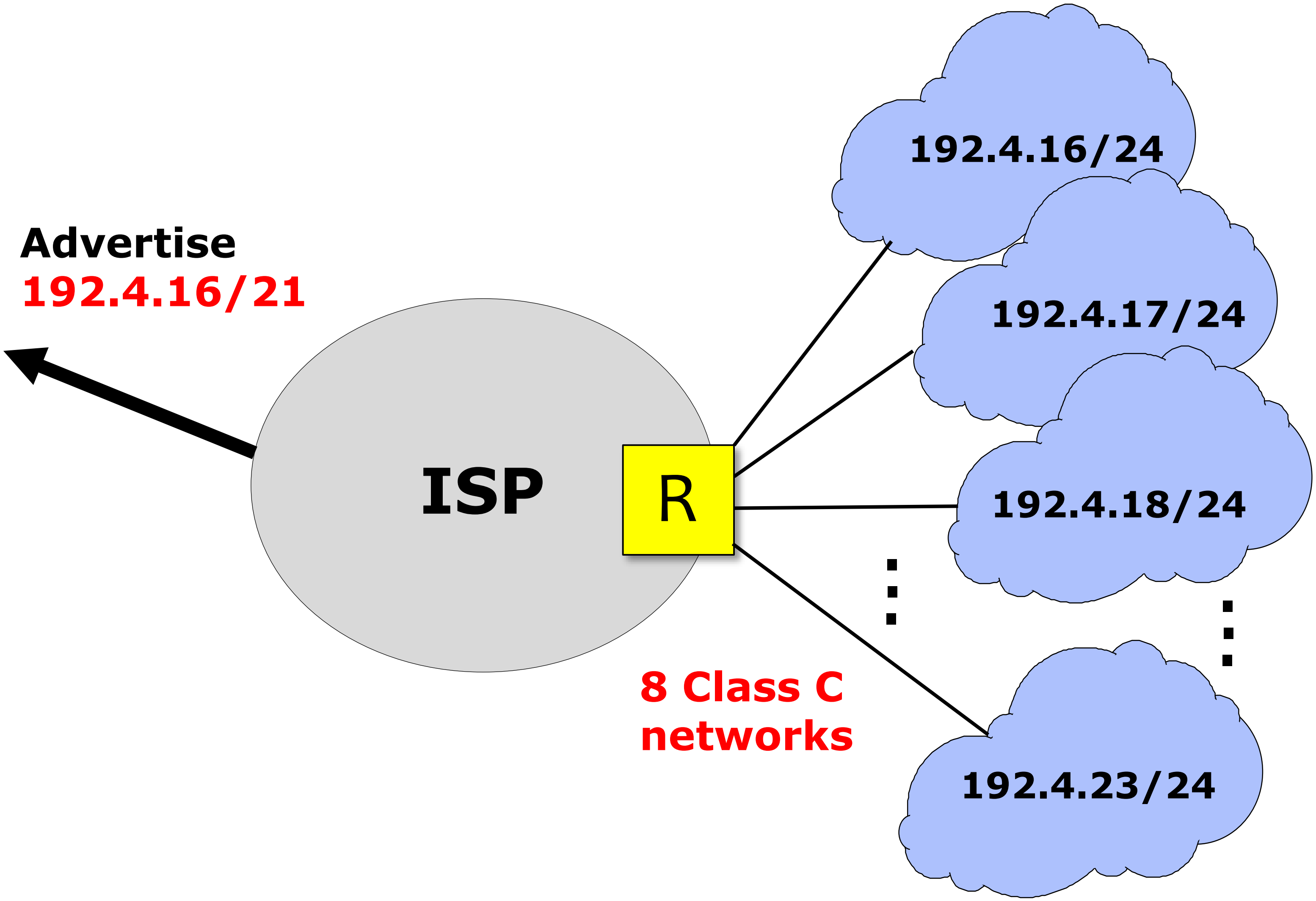
Classless Addressing

- How do the routing protocols handle this classless address ?
  - It must understand that the network number may be of any length
- Represent network number with a single pair

  <length, value>

- All routers must understand CIDR addressing
- CIDR means that prefixes may be of any length, from 2 to 32 bits

Classless Addressing



Route aggregation with CIDR

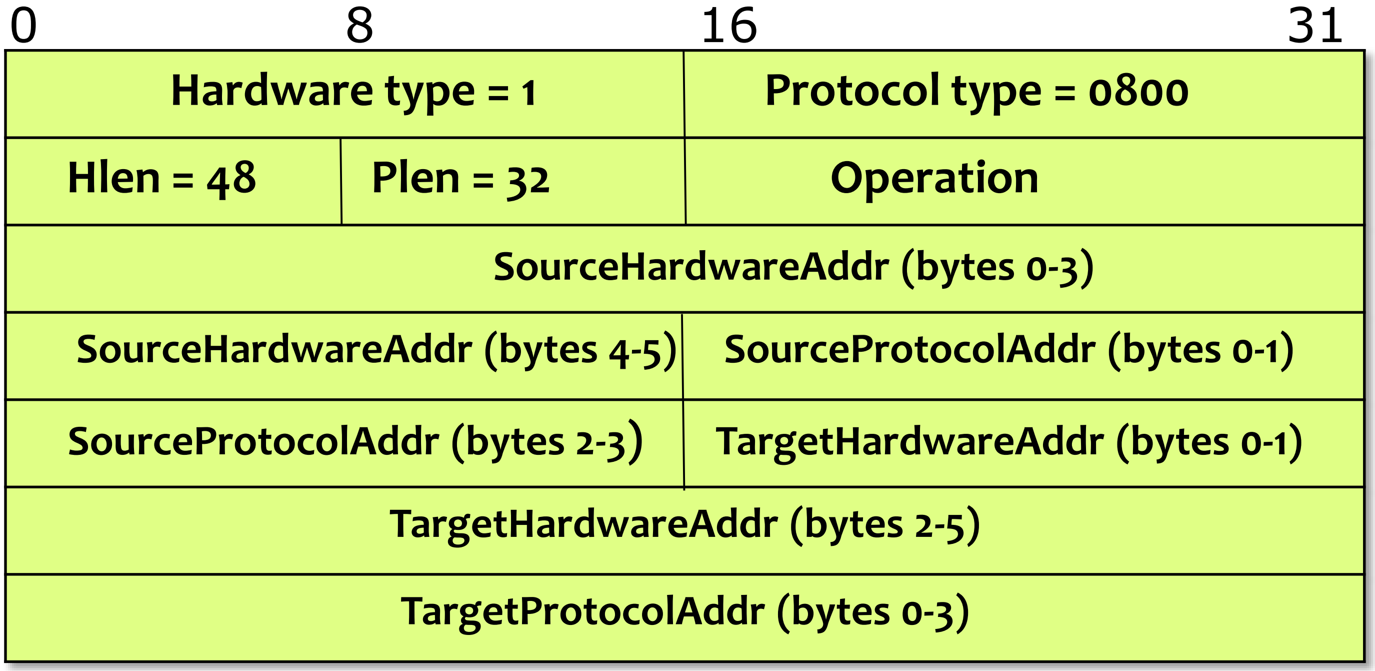
Longest prefix matching

- It is also possible to have prefixes in the forwarding tables that overlap
  - Some addresses may match more than one prefix
- For example, we might find both
  - 171.69 (a 16 bit prefix) and
  - 171.69.10 (a 24 bit prefix) in the forwarding table of a single router
- A packet destined to 171.69.10.5 clearly matches both prefixes.
  - The rule is based on the principle of "longest prefix match"
    - 171.69.10 in this case
- A packet destined to 171.69.20.5 would match 171.69 and not 171.69.10

Address Resolution Protocol (ARP)

- Map IP addresses into physical (MAC) addresses
  - destination host, next hop router
- ARP (Address Resolution Protocol)
  - table of IP to physical address bindings
  - broadcast request if IP address not in table
  - target machine responds with its physical address
  - table entries are discarded if not refreshed

ARP Packet Format



0 8 16 31

Hardware type = 1 | Protocol type = 0800

Hlen = 48 | Plen = 32 | Operation

SourceHardwareAddr (bytes 0-3)

SourceHardwareAddr (bytes 4-5) | SourceProtocolAddr (bytes 0-1)

SourceProtocolAddr (bytes 2-3) | TargetHardwareAddr (bytes 0-1)

TargetHardwareAddr (bytes 2-5)

TargetProtocolAddr (bytes 0-3)

- HardwareType: type of physical network (e.g., Ethernet)
- ProtocolType: type of higher layer protocol (e.g., IP)
- HLEN & PLEN: length of physical and protocol addresses
- Operation: request or response
- Source/Target Physical/Protocol addresses

Host Configurations

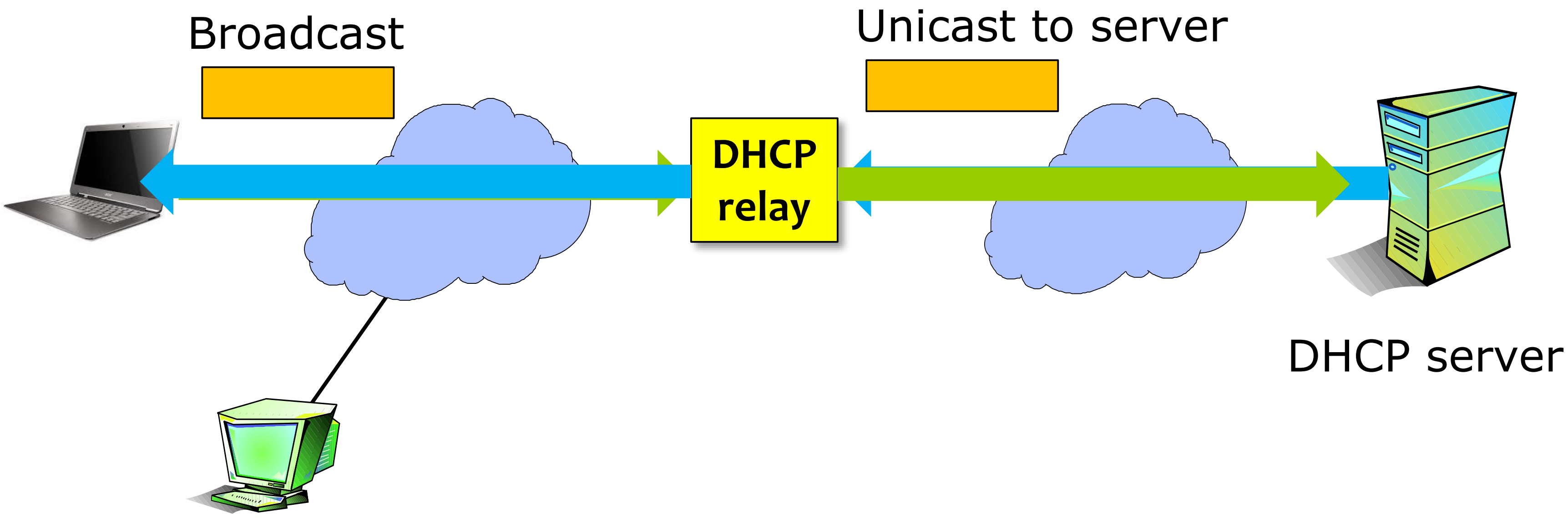
- Ethernet addresses are configured into network by manufacturer and they are unique
- IP addresses must be unique on a given internetwork but also must reflect the structure of the internetwork
- Most host Operating Systems provide a way to manually configure the IP information for the host
- Drawbacks of manual configuration
    - A lot of work to configure all the hosts in a large network
- Automated Configuration Process is required

Dynamic Host Configuration Protocol (DHCP)

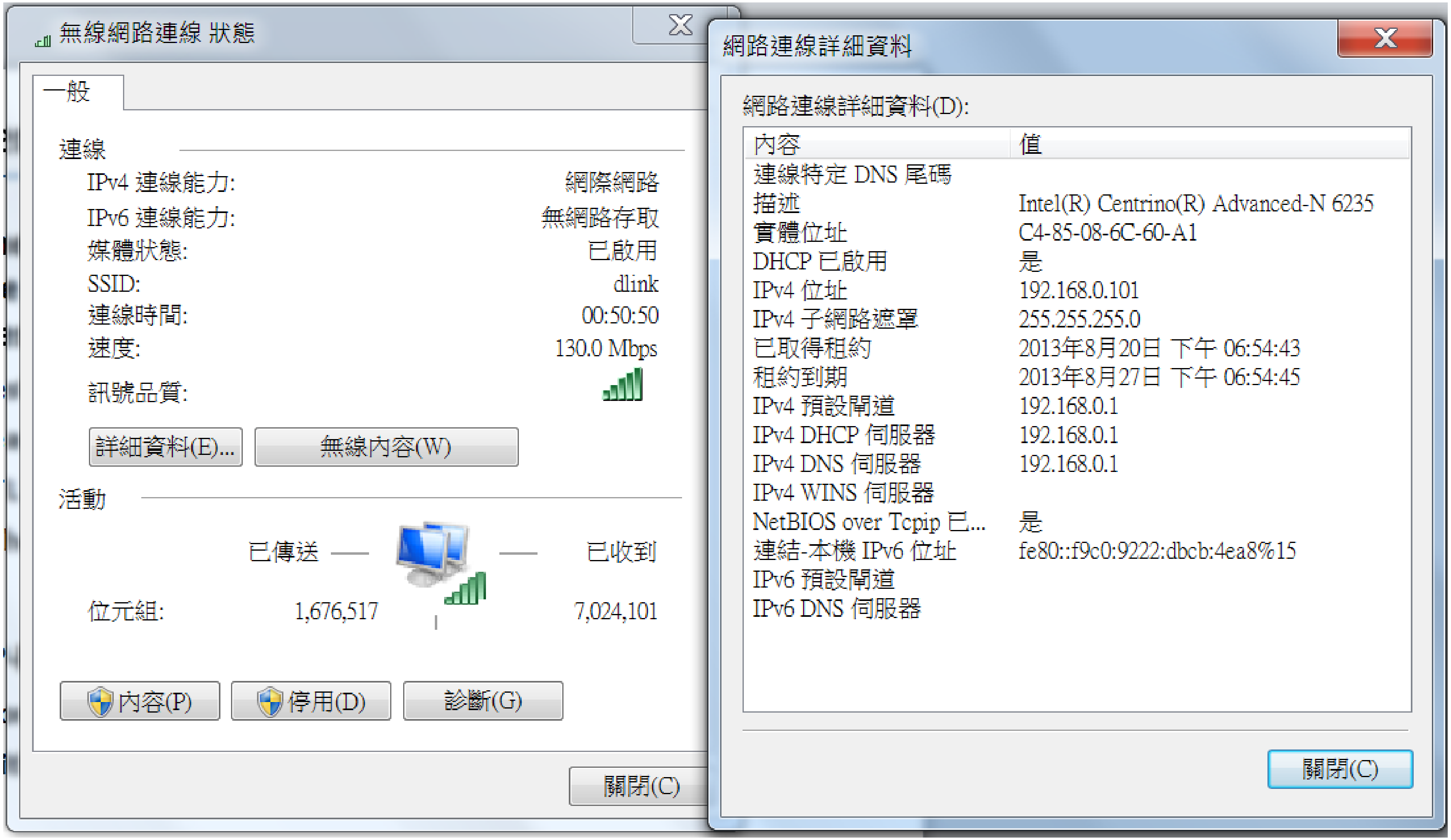
- DHCP server is responsible for providing configuration information to hosts
- There is at least one DHCP server for an administrative domain
- DHCP server maintains a pool of available addresses

DHCP

- Newly booted or attached host sends **DHCP DISCOVER** message to a special IP address (255.255.255.255)
- DHCP relay agent unicasts the message to DHCP server and waits for the response



DHCP Example



連線

IPv4 連線能力:	網際網路
IPv6 連線能力:	無網路存取
媒體狀態:	已啟用
SSID:	dlink
連線時間:	00:50:50
速度:	130.0 Mbps
訊號品質:	

活動

	已傳送	已收到
位元組:	1,676,517	7,024,101

網路連線詳細資料(D):

內容	值
連線特定 DNS 尾碼	
描述	Intel(R) Centrino(R) Advanced-N 6235
實體位址	C4-85-08-6C-60-A1
DHCP 已啟用	是
IPv4 位址	192.168.0.101
IPv4 子網路遮罩	255.255.255.0
已取得租約	2013年8月20日 下午 06:54:43
租約到期	2013年8月27日 下午 06:54:45
IPv4 預設閘道	192.168.0.1
IPv4 DHCP 伺服器	192.168.0.1
IPv4 DNS 伺服器	192.168.0.1
IPv4 WINS 伺服器	
NetBIOS over Tcpip 已...	是
連結-本機 IPv6 位址	fe80::f9c0:9222:dbcb:4ea8%15
IPv6 預設閘道	
IPv6 DNS 伺服器	

Internet Control Message Protocol (ICMP)

- Defines a collection of error messages that are sent back to the source host whenever a router or host is unable to process an IP datagram successfully
  - Destination host unreachable due to link /node failure
  - Reassembly process failed
  - TTL had reached 0 (so datagrams don't cycle forever)
  - IP header checksum failed

- ICMP-Redirect
  - From router to a source host
  - With a better route information

Outline

- Introduction
- IP and Routers
- IP Subnetting
- Classless Addressing
- Routing protocols
- Distance Vector protocol
- Link State protocol

Routing

- Forwarding versus Routing
  - **Forwarding:**
    - to select an output port based on destination address and routing table
  - **Routing:**
    - process to build the routing table

Routing

- Forwarding table vs. Routing table
  - Forwarding table
    - Used when a packet is being forwarded
    - An entry in the forwarding table contains the mapping from a network number to an outgoing interface and some MAC information, such as Ethernet Address of the next hop
  - Routing table
    - Built by the routing algorithm
    - Generally contains mapping from network numbers to next hops

Routing

Prefix/Length	Next Hop
140.114/16	171.34.45.12

(a)

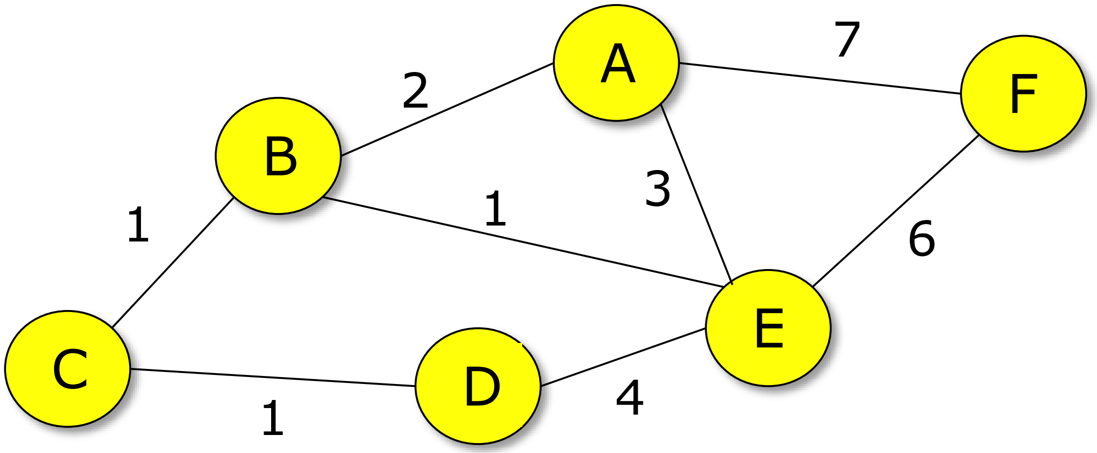
Prefix/Length	Interface	MAC Address
140.114/16	0	8:0:2c:e3:b:20

(b)

Example rows from (a) routing and (b) forwarding tables

Routing

- Network as a Graph



- The basic problem of routing is to find the lowest-cost path between any two nodes
  - Where the cost of a path equals the sum of the costs of all the edges that make up the path

Routing

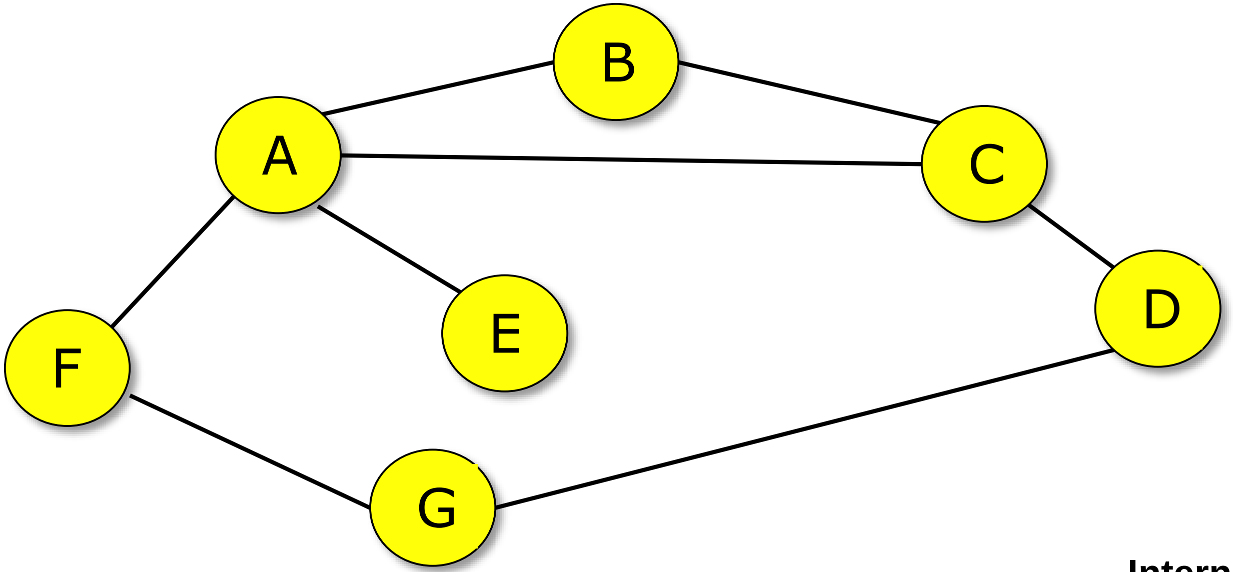
- For a simple network, we can calculate all shortest paths and load them into each node.
- Such a static approach has several shortcomings
  - It does not deal with node or link failures
  - It does not consider the addition of new nodes or links
  - It implies that edge costs cannot change
- What is the solution ?
  - Need a distributed and dynamic protocol
  - Two main classes of protocols
    - Distance Vector
    - Link State

Outline

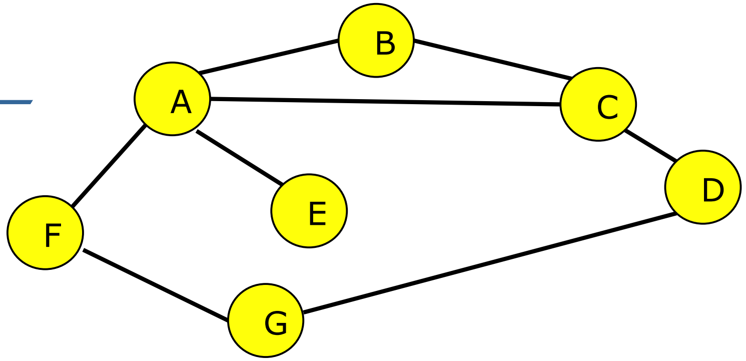
- Introduction
- IP and Routers
- IP Subnetting
- Classless Addressing
- Routing protocols
- Distance Vector protocol
- Link State protocol

Distance Vector

- Each node constructs a one dimensional array (a vector) containing the “distances” (costs) to all other nodes and distributes that vector to its immediate neighbors
- Assume that each node knows the cost of the link to each of its directly connected neighbors



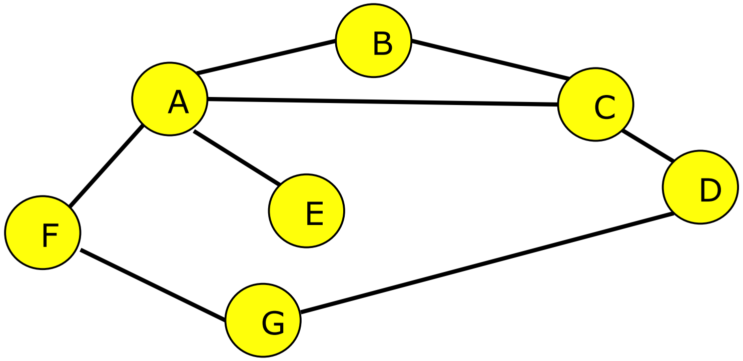
Distance Vector



Initial distances stored at each node (global view)

Information at node	Distance to reach node						
	A	B	C	D	E	F	G
A	**0**	**1**	**1**	∞	**1**	**1**	∞
B	**1**	**0**	**1**	∞	∞	∞	∞
C	**1**	**1**	**0**	∞	∞	∞	**1**
D	∞	∞	**1**	**0**	∞	∞	**1**
E	**1**	∞	∞	∞	**0**	∞	∞
F	**1**	∞	∞	∞	∞	**0**	**1**
G	∞	∞	∞	**1**	∞	**1**	**0**

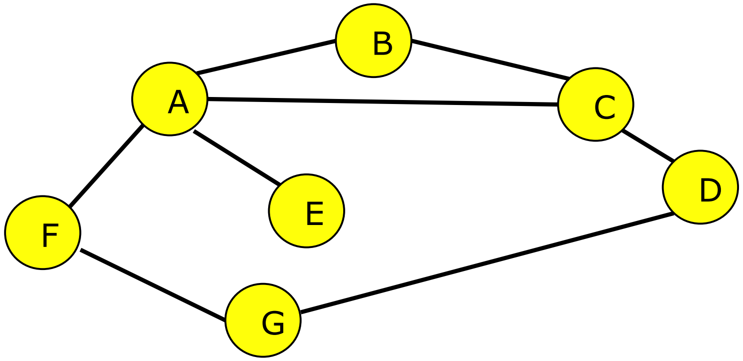
Distance Vector



Destination	Cost	NextHop
B	1	B
C	1	C
D	∞	--
E	1	E
F	1	F
G	∞	--

Initial routing table at node A

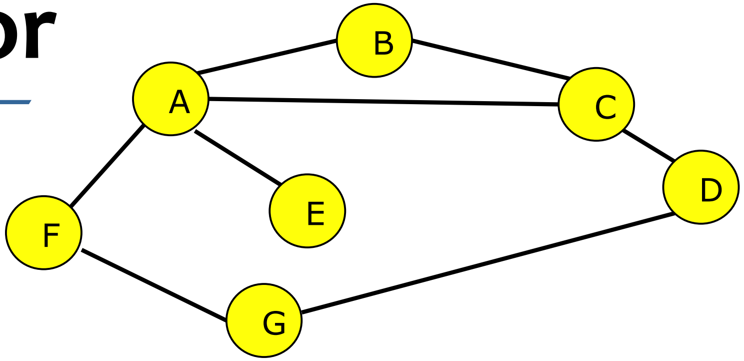
Distance Vector



Destination	Cost	NextHop
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

Final routing table at node A

Distance Vector



Final distances stored at each node (global view)

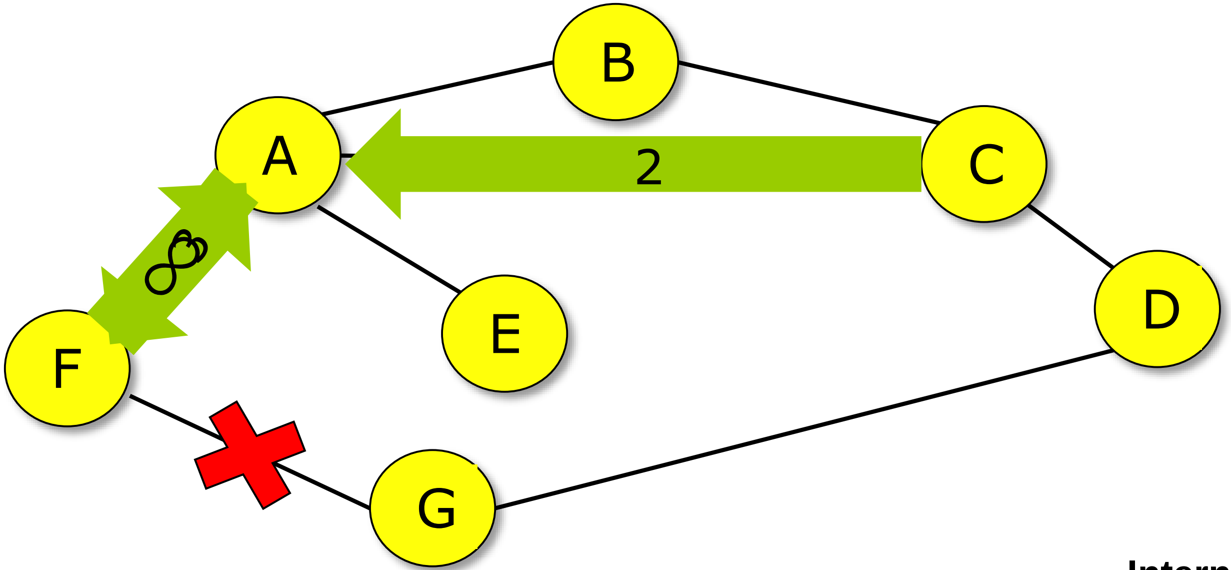
Information at node	Distance to reach node						
	A	B	C	D	E	F	G
A	**0**	**1**	**1**	**2**	**1**	**1**	**2**
B	**1**	**0**	**1**	**2**	**2**	**2**	**3**
C	**1**	**1**	**0**	**1**	**2**	**2**	**2**
D	**2**	**2**	**1**	**0**	**3**	**2**	**1**
E	**1**	**2**	**2**	**3**	**0**	**2**	**3**
F	**1**	**2**	**2**	**2**	**2**	**0**	**1**
G	**2**	**3**	**2**	**1**	**3**	**1**	**0**

Distance Vector

- Every T seconds each router sends its routing table to its neighbors
- Each router then updates its routing table based on the new information
- Problems include
  - fast response to good news
  - slow response to bad news
  - Too many messages to update

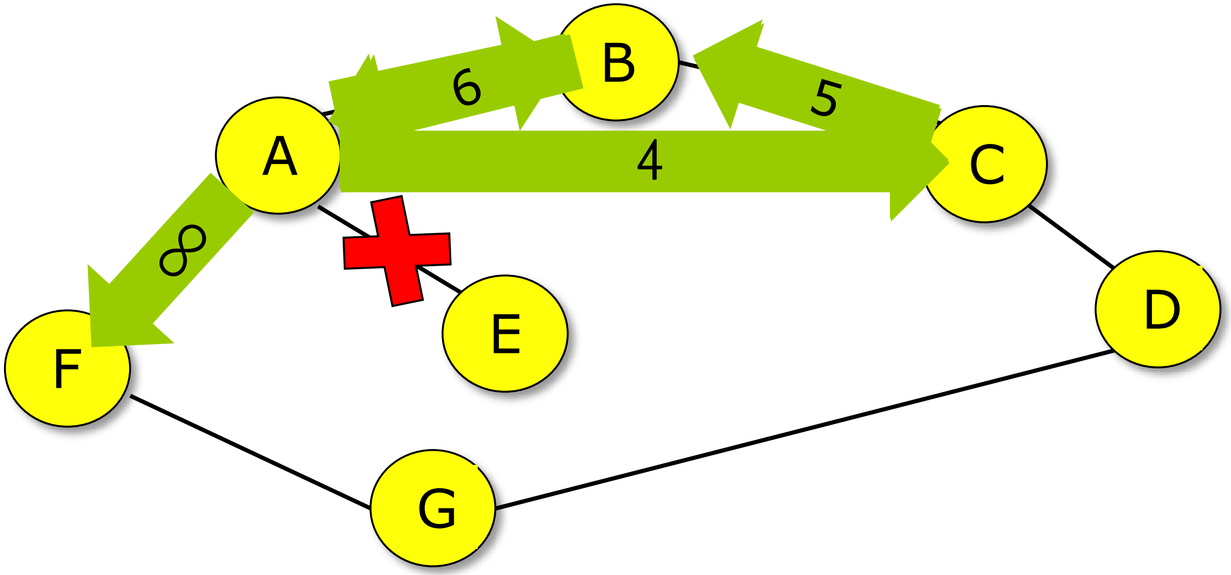
Distance Vector

- When a node **detects a link failure**
  - F detects that link to G has failed
  - F sets distance to G to infinity and sends update to A
  - A sets distance to G to infinity since it uses F to reach G
  - A receives periodic update from C with 2-hop path to G
  - A sets distance to G to 3 and sends update to F
  - F decides it can reach G in 4 hops via A



Distance Vector

- **Count-to-infinity problem**
- **Slightly different cases may cause the network unstable**
  - **Suppose the link from A to E goes down**
  - **In the next round of updates, A advertises a distance of infinity to E, but B and C advertise a distance of 2 to E**



Distance Vector

- Depending on the exact timing of events, the following might happen
  - Node B, upon hearing that E can be reached in 2 hops from C, concludes that it can reach E in 3 hops and advertises this to A
  - Node A concludes that it can reach E in 4 hops and advertises this to C
  - Node C concludes that it can reach E in 5 hops; and so on.
  - This cycle stops only when the distances reach some number that is large enough to be considered infinite
    - **Count-to-infinity problem**

Count-to-infinity Problem

- In fact, some relatively small number is used to approximate the infinity
- For example, the maximum number of hops to get across a certain network is less than 16
- One technique to improve the time to stabilize routing is called ***split horizon***
  - When a node sends a routing update to its neighbors, it **does not send those routes it learned from each neighbor back to that neighbor**
  - For example, if B has the route (E, 2, A) in its table, then it knows it must have learned this route from A, and so whenever B sends a routing update to A, it does not include the route (E, 2) in that update

Count-to-infinity Problem

- *split horizon with poison reverse,* a stronger version of split horizon
  - B actually sends that back route to A, but it puts negative information in the route to ensure that A will not eventually use B to get to E
  - For example, B sends the route (E, ∞) to A

Outline

- Introduction
- IP and Routers
- IP Subnetting
- Classless Addressing
- Routing protocols
- Distance Vector protocol
- Link State protocol

Link State Routing

Strategy: Send to all nodes (not just neighbors) information about directly connected links (not entire routing table).

- Link State Packet (LSP)
  - ID of the node that created the LSP
  - Cost of link to each directly connected neighbor
  - Sequence number (SEQNO)
  - Time-to-live (TTL) for this packet

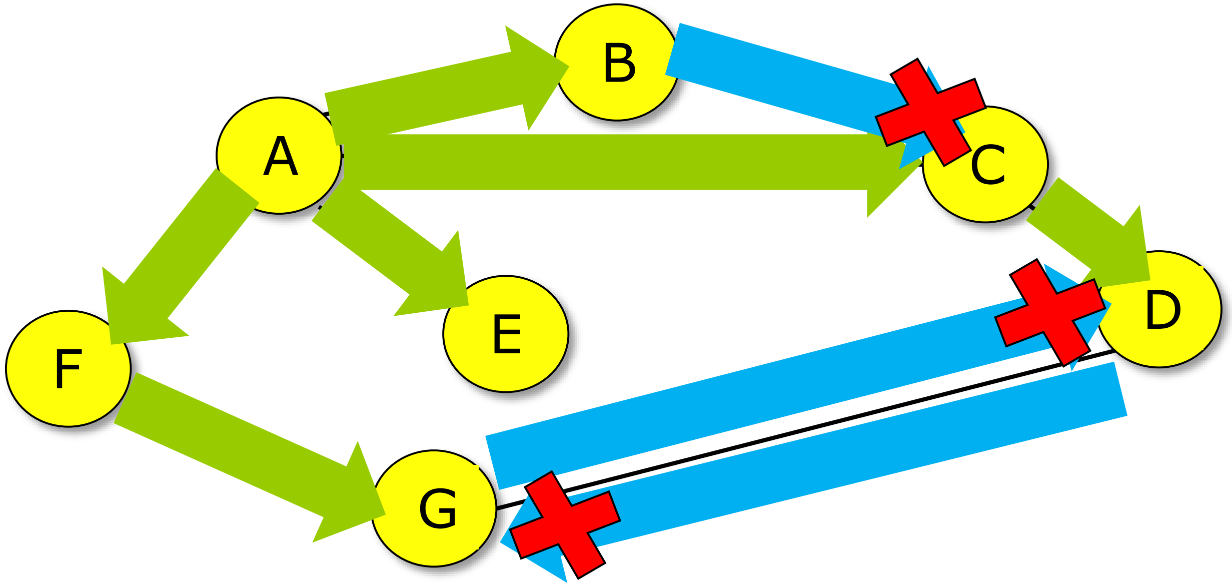
Link State Routing

- **Reliable Flooding**
  - Store most recent LSP from each node
  - Forward LSP to all nodes but one that sent it
  - Generate new LSP periodically; increment SEQNO
  - Start SEQNO at 0 when reboot
  - Decrement TTL of each stored LSP; discard when TTL=0

Link State Routing

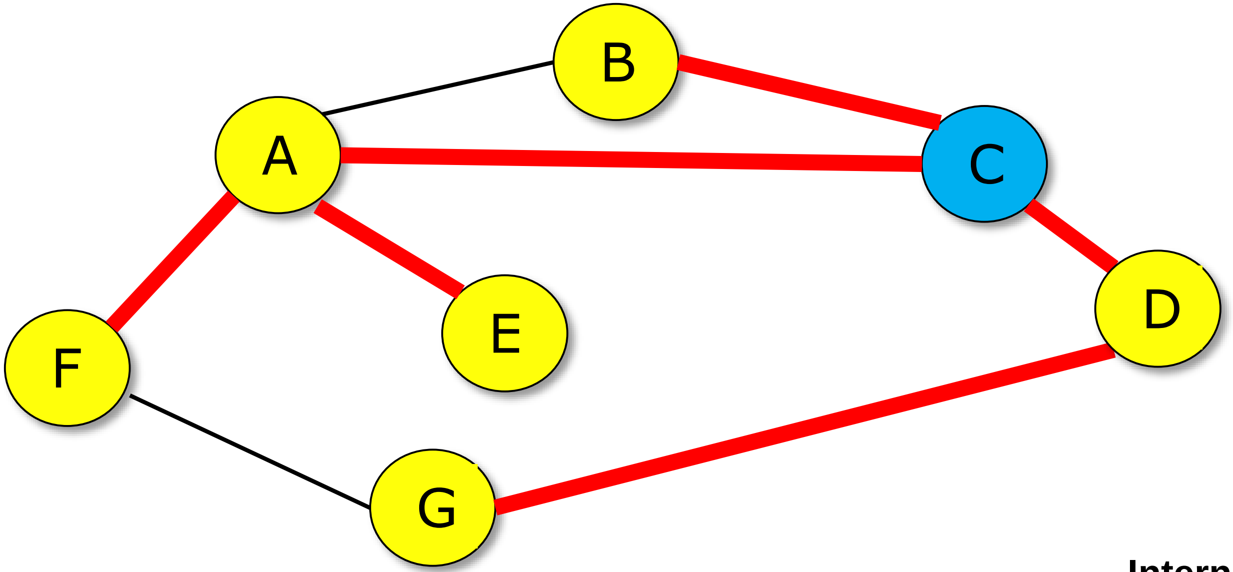
Example of reliable flooding of LSP packets

From node A



Shortest Path Routing

- OSPF (Open Shortest Path First)
- Each router computes its routing table directly from the LSP's it has collected using the Dijkstra's algorithm
- Find the shortest path from the router to each other node.



Summary

- Introduced issues for building scalable and heterogeneous networks by using routers to interconnect networks.
- Internet Protocol (IP)
  - Connectionless model for data delivery
  - Best-effort delivery (unreliable service)
    - packets are lost
    - packets are delivered out of order
    - duplicate copies of a packet are delivered
    - packets can be delayed for a long time
- Routers and Routing protocols
  - Routing Table lookup

Summary

- IP Subnetting
  - Subnetmask → 255.255.255.0
  - Subnet number
- Classless Inter-Domain Routing (CIDR)
  - Routes aggregated to reduce routing table size
  - Prefix and Prefix length
    - 192.4.16/21 presents 8 class C networks
    - 192.4.16/22 presents 4 class C networks
- DHCP (Dynamic Host Configuration Protocol)
- ICMP (Internet Control Message Protocol)

Summary

- Two major classes of routing protocols
  - Distance Vector
    - Send entire routing table to directly connected neighbors.
    - fast response to good news
    - slow response to bad news
    - Count-to-infinite problem
    - Split-horizon solution
    - *Split horizon with poison reverse solution*
    - RIP (Routing Information Protocol)

Summary

- Link State
  - Send to all nodes information about directly connected links
  - Reliable broadcasting LSPs
  - Each node has the entire network topology of the AS
  - Calculate the shortest path to each other node
  - OSPF (Open Shortest Path First)