HY 335

Φροντιστήριο 6ο

Χειμερινό Εξάμηνο 2010-2011

Παπακωνσταντίνου Άρτεμις
artpap@csd.uoc.gr

25/11/2010

Roadmap

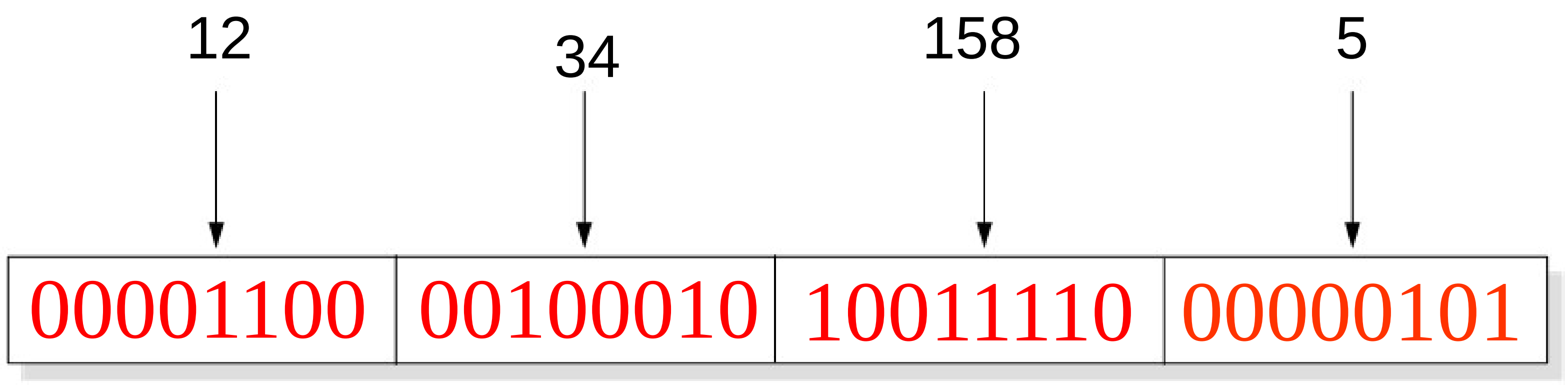
- IP: The Internet Protocol
  - IPv4 Addressing
  - Transporting a datagram from source to destination
  - IP Fragmentation & Reassembly
  - ICMP
  - DHCP
  - ARP
  - IPV6
- Routing Algorithms: Dijkstra's Example

Roadmap

- IP: The Internet Protocol
  - IPv4 Addressing
  - Transporting a datagram from source to destination
  - IP Fragmentation & Reassembly
  - ICMP
  - DHCP
  - ARP
  - IPV6
- Routing Algorithms: Dijkstra's Example

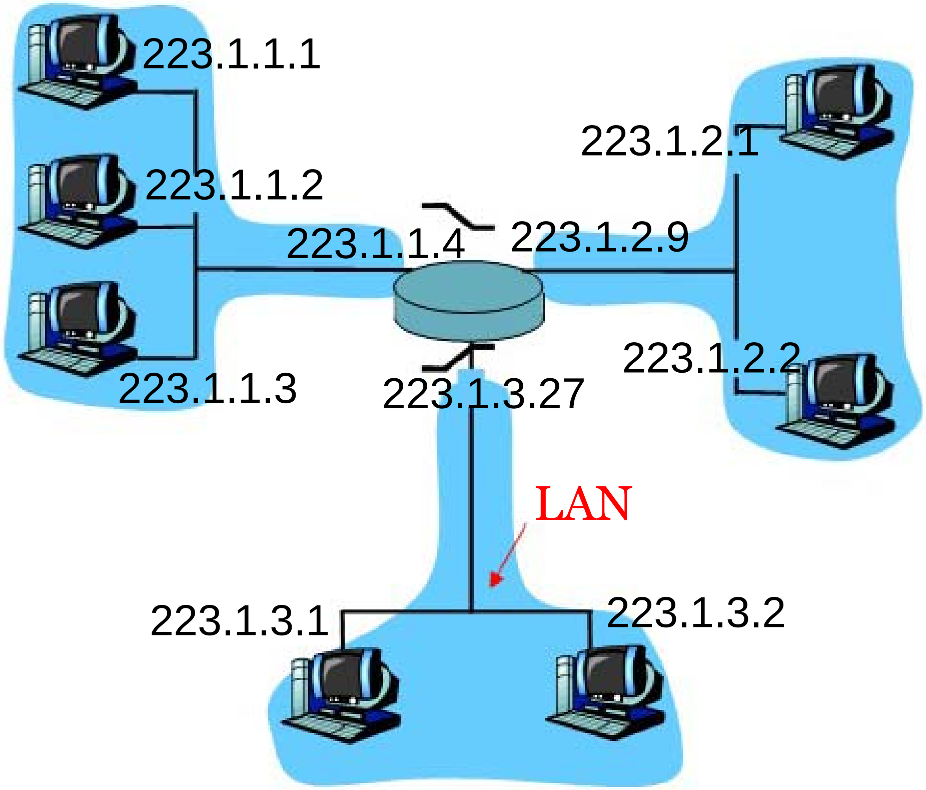
IP Address (IPv4)

- A unique 32-bit number
- Identifies an interface (on a host, on a router, …)
- interface: connection between host/router and physical link
  - router's typically have multiple interfaces
  - host may have multiple interfaces
  - IP addresses associated with each interface
- Represented in dotted-decimal notation

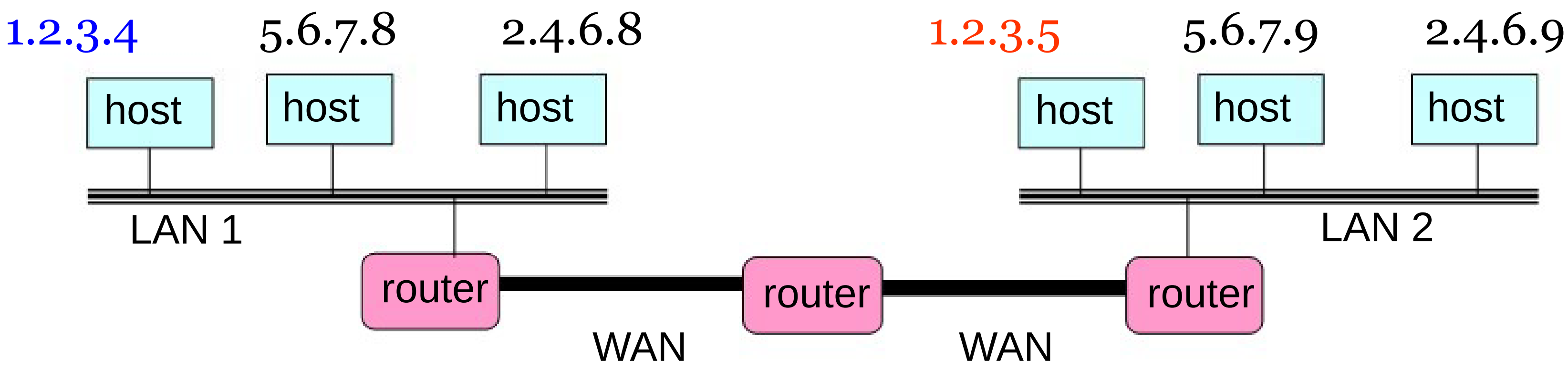


Grouping Related Hosts

- The Internet is an "inter-network"
  - Used to connect networks together, not hosts
  - Needs a way to address a network (i.e., group of hosts)



Scalability Challenge



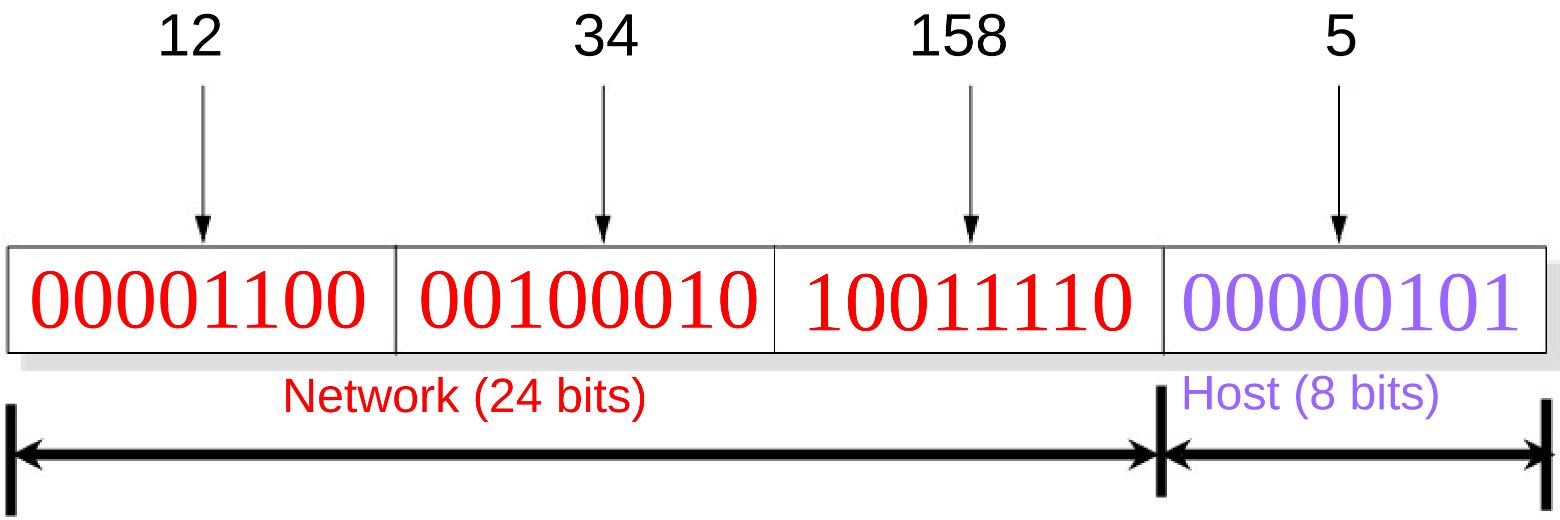
- Suppose hosts had arbitrary addresses
  - Then every router would need a lot of information
  - ...to know how to direct packets toward the host

1.2.3.4	←
1.2.3.5	→
⋮	

forwarding table

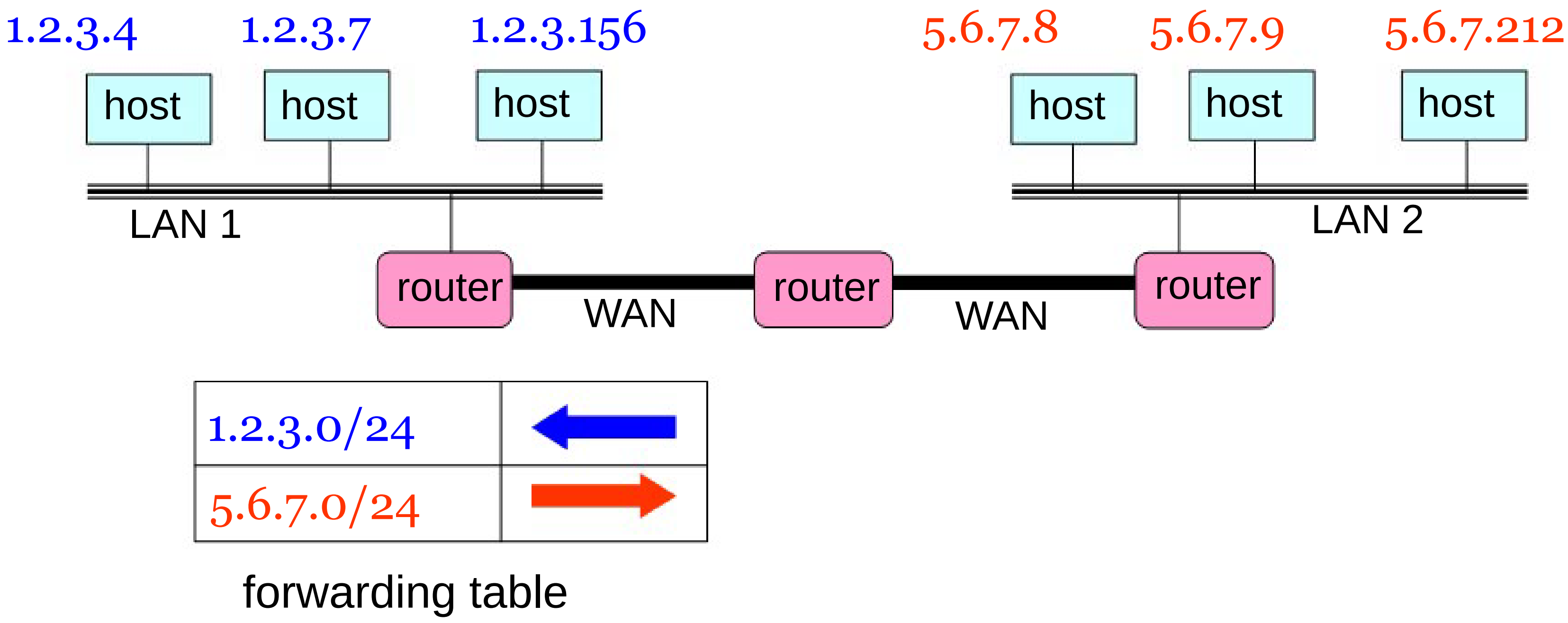
Hierarchical Addressing: IP Prefixes

- Divided into network & host portions (left and right)
- Forming subnets:
  - device interfaces with same network part of IP address
  - can physically reach each other without intervening router
- 12.34.158.0/24 is a 24-bit prefix with 2^8 addresses



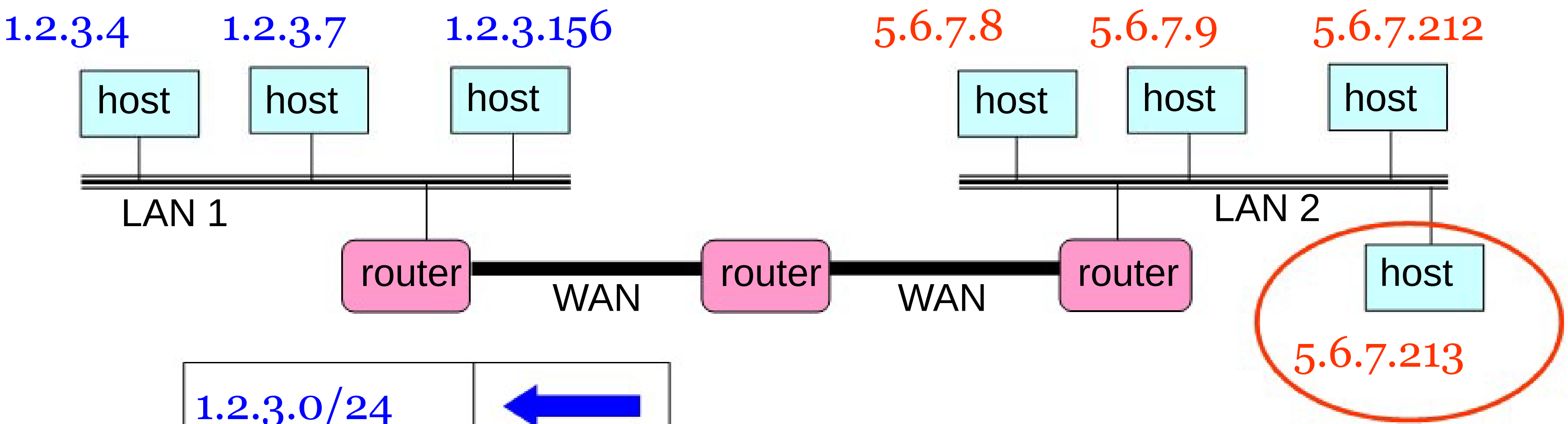
Scalability Improved

- Group related hosts from a common subnet
  - 1.2.3.0/24 on the left LAN
  - 5.6.7.0/24 on the right LAN



Easy to Add New Hosts

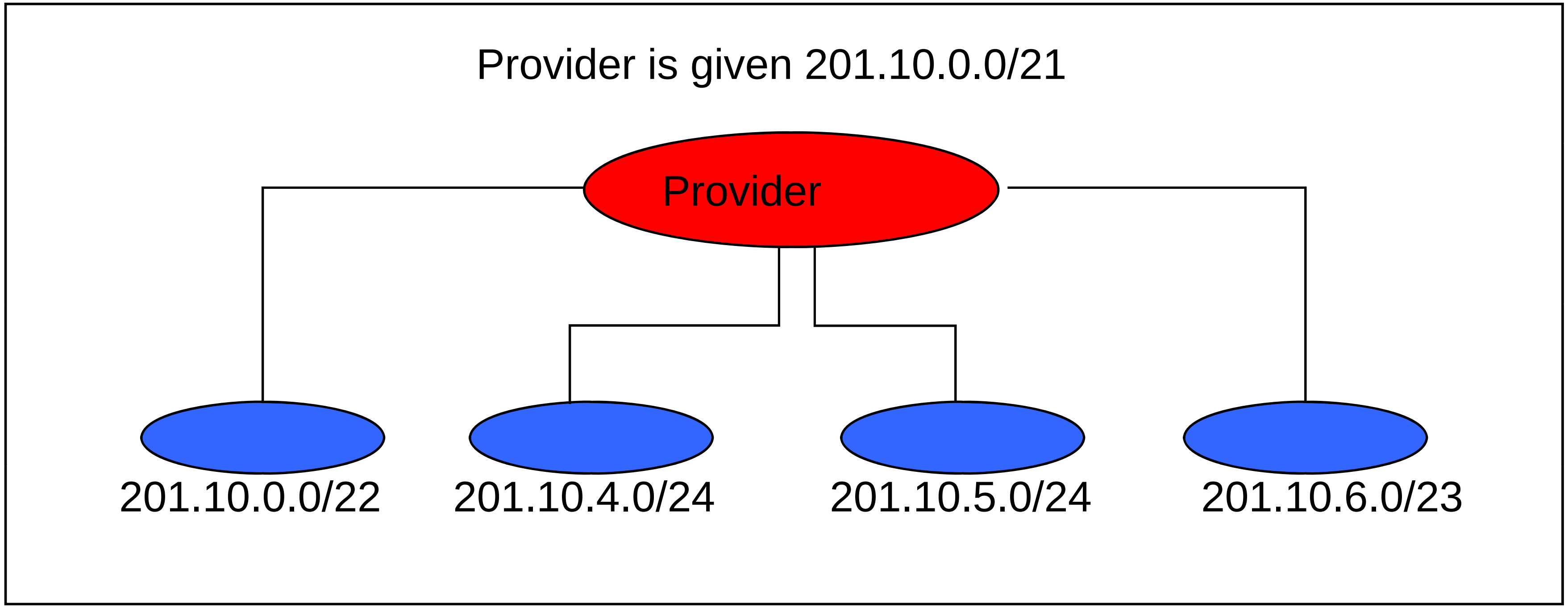
- No need to update the routers
  - E.g., adding a new host 5.6.7.213 on the right doesn't require adding a new forwarding-table entry



1.2.3.0/24	←
5.6.7.0/24	→

forwarding table

Scalability: Address Aggregation



Routers in the rest of the Internet just need to know how to reach 201.10.0.0/21. The provider can direct the IP packets to the appropriate customer.

Class-full Addressing

- In the older days, only fixed allocation sizes
  - Class A: 0*
    - Very large /8 blocks (e.g., MIT has 18.0.0.0/8)
  - Class B: 10*
    - Large /16 blocks (e.g,. Princeton has 128.112.0.0/16)
  - Class C: 110*
    - Small /24 blocks (e.g., AT&T Labs has 192.20.225.0/24)
  - Class D: 1110*
    - Multicast groups

IP addressing: CIDR

- Class-full addressing:
  - inefficient use of address space, address space exhaustion
  - e.g., class B net allocated enough addresses for 65K hosts, even if only 2K hosts in that network
- CIDR: Classless InterDomain Routing
  - network portion of address of arbitrary length
  - address format: a.b.c.d/x, where x is # bits in network portion of address

network part | host part

11001000 00010111 00010000 00000000

200.23.16.0/23

IP addresses: how to get one?

Q: How does host get IP address?

- hard-coded by system admin in a file
- DHCP: Dynamic Host Configuration Protocol: dynamically get address from server
  - "plug-and-play"

Obtaining a Block of Addresses

- Separation of control
  - Prefix: assigned to an institution
  - Addresses: assigned by the institution to their nodes
- Who assigns prefixes?
  - Internet Corporation for Assigned Names and Numbers
    - Allocates large address blocks to Regional Internet Registries
  - Regional Internet Registries (RIRs)
    - E.g., ARIN (American Registry for Internet Numbers)
    - Allocates address blocks within their regions
    - Allocated to Internet Service Providers and large institutions
  - Internet Service Providers (ISPs)
    - Allocate address blocks to their customers
    - Who may, in turn, allocate to their customers…

Roadmap

- IP: The Internet Protocol
  - IPv4 Addressing
  - Transporting a datagram from source to destination
  - IP Fragmentation & Reassembly
  - ICMP
  - DHCP
  - ARP
  - IPV6
- Routing Algorithms: Dijkstra's Example

Hop-by-Hop Packet Forwarding

- Each router has a forwarding table
  - Maps destination addresses…
  - … to outgoing interfaces
- Upon receiving a packet
  - Inspect the destination IP address in the header
  - Index into the table
  - Determine the outgoing interface
  - Forward the packet out that interface
- Then, the next router in the path repeats
  - And the packet travels along the path to the destination

Getting a datagram from source to dest.

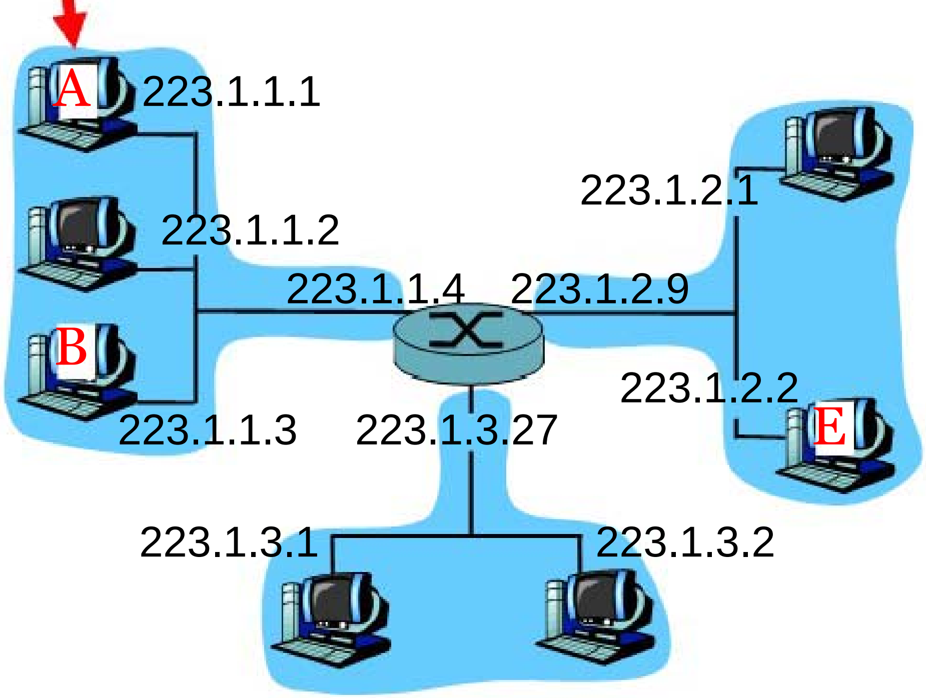
misc fields	223.1.1.1	223.1.1.3	data

forwarding table in A

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2

Starting at A, send IP datagram addressed to B:

- look up net. address of B in forwarding table
- find B is on same net. as A
- link layer will send datagram directly to B inside link-layer frame
  - B and A are directly connected



Getting a datagram from source to dest.

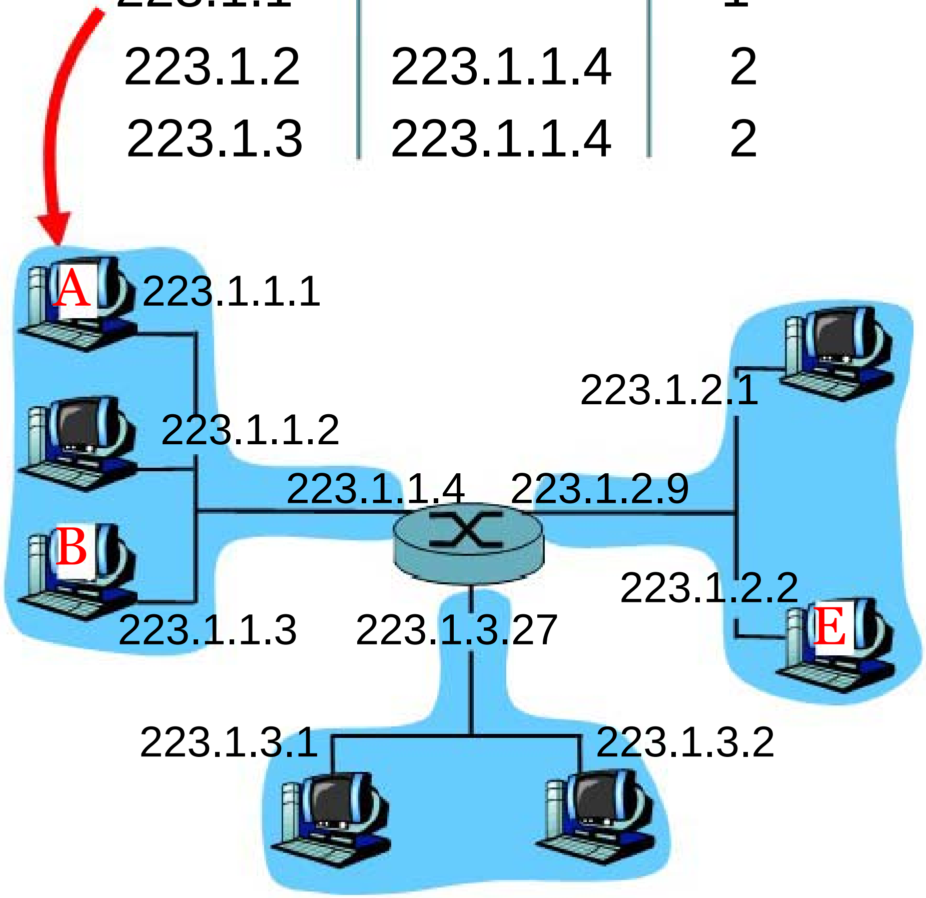
misc fields	223.1.1.1	223.1.2.2	data

Starting at A, dest. E:

- look up network address of E in forwarding table
- E on different network
  - A, E not directly attached
- routing table: next hop router to E is 223.1.1.4
- link layer sends datagram to router 223.1.1.4 inside link-layer frame
- datagram arrives at 223.1.1.4
- continued…..

forwarding table in A

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



Getting a datagram from source to dest.

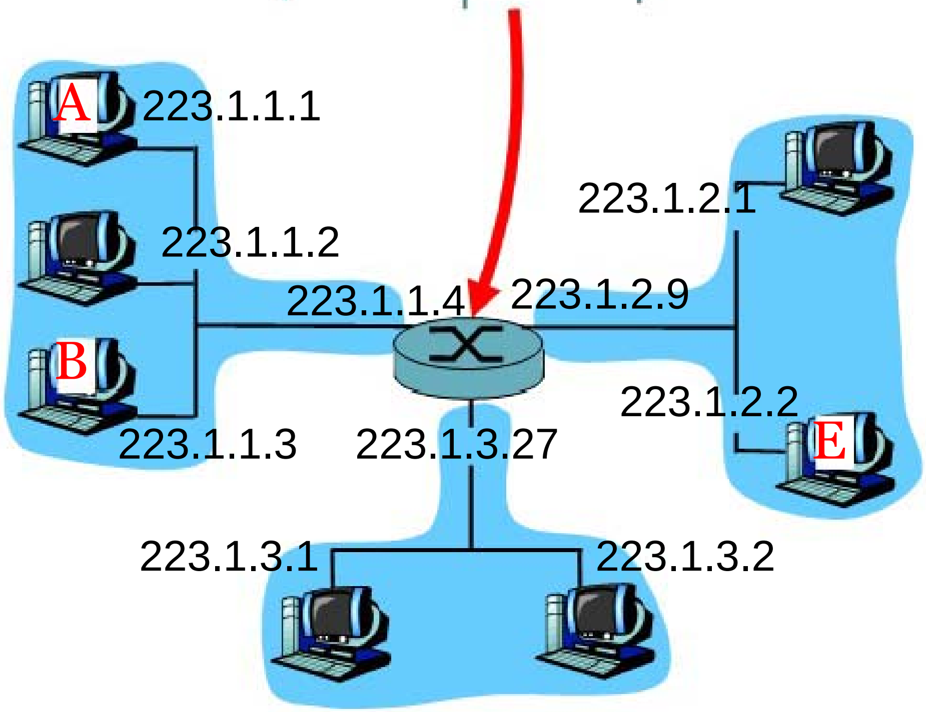
misc fields	223.1.1.1	223.1.2.2	data

Arriving at 223.1.4, destined for 223.1.2.2

- look up network address of E in router's forwarding table
- E on same network as router's interface 223.1.2.9
  - router, E directly attached
- link layer sends datagram to 223.1.2.2 inside link-layer frame via interface 223.1.2.9
- datagram arrives at 223.1.2.2!!!

forwarding table in router

Dest. Net	router	Nhops	interface
223.1.1	-	1	223.1.1.4
223.1.2	-	1	223.1.2.9
223.1.3	-	1	223.1.3.27

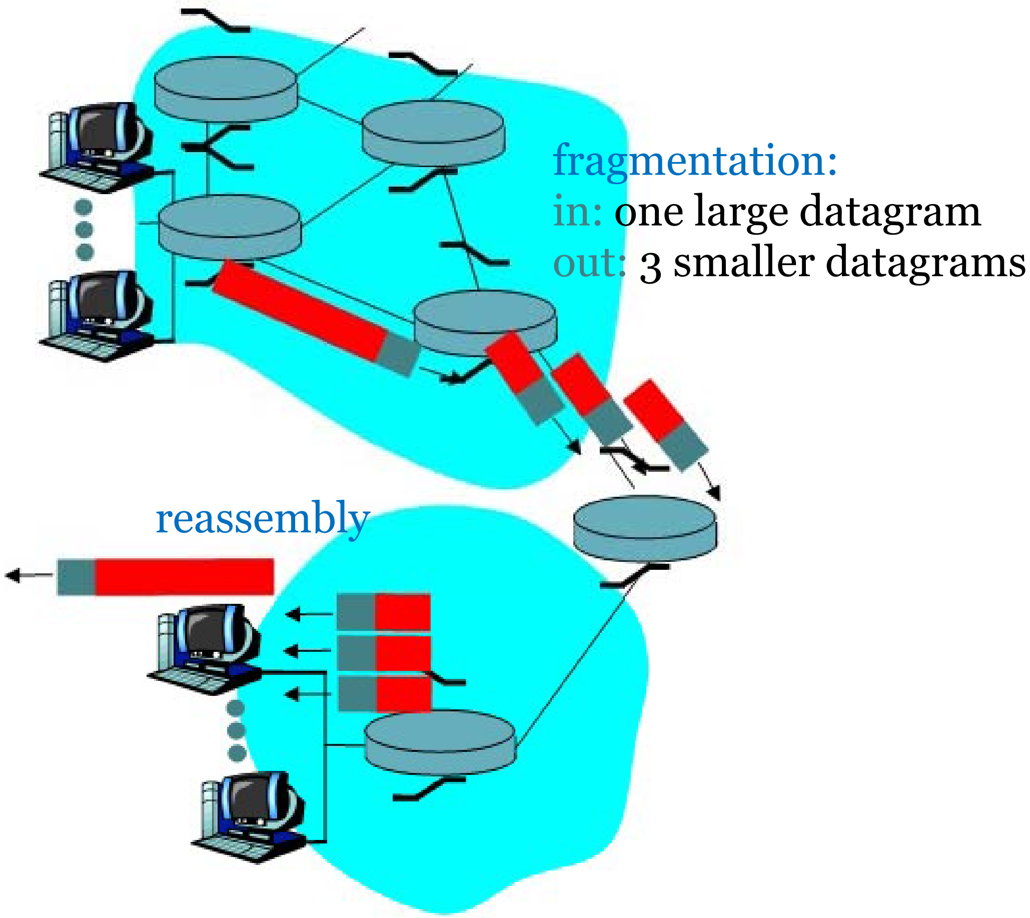


Roadmap

- IP: The Internet Protocol
  - IPv4 Addressing
  - Transporting a datagram from source to destination
  - IP Fragmentation & Reassembly
  - ICMP
  - DHCP
  - ARP
  - IPV6
- Routing Algorithms: Dijkstra's Example

IP Fragmentation & Reassembly

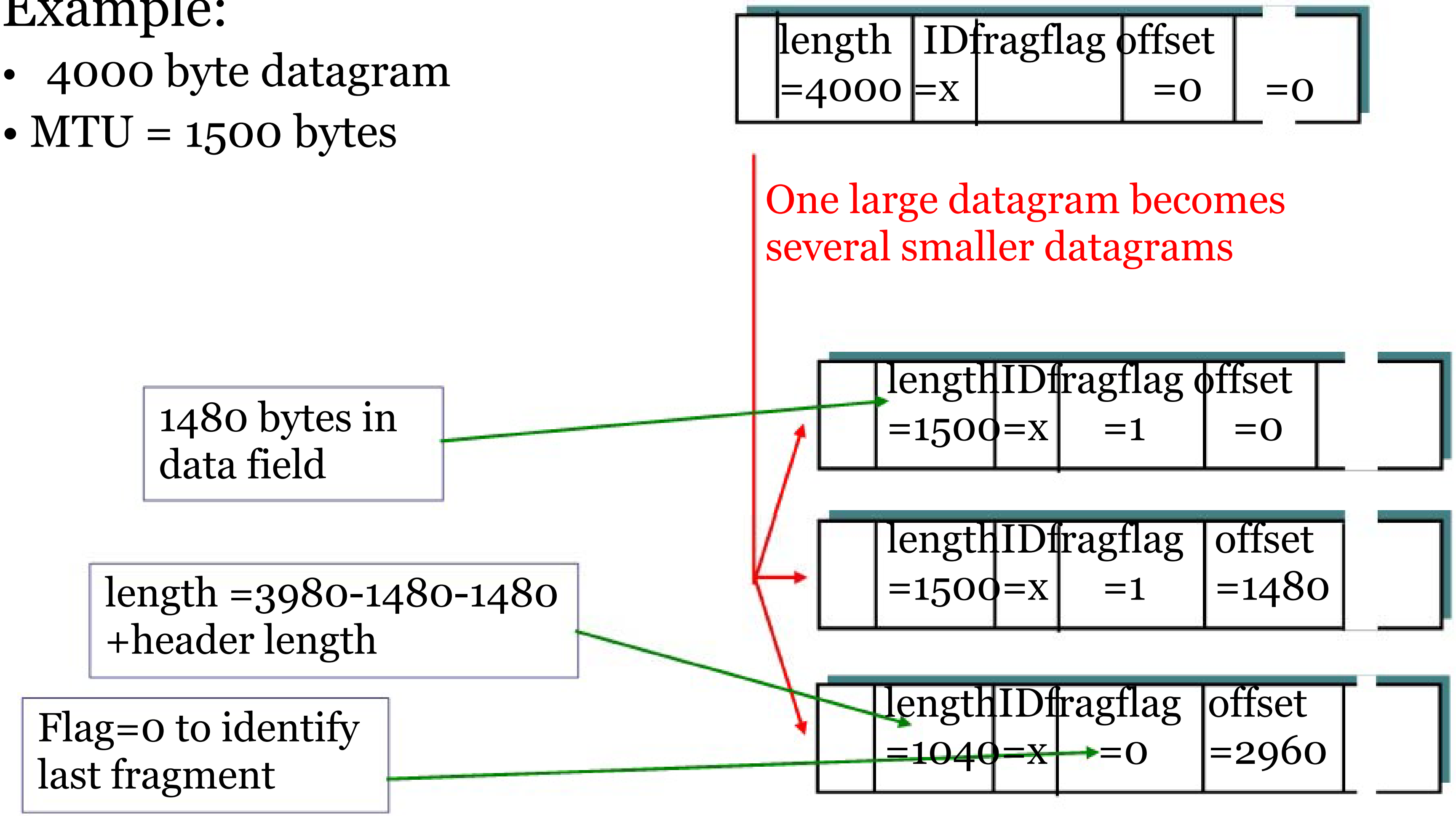
- network links have MTU (max.transfer unit) - largest possible link-level frame.
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify,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



IP Fragmentation & Reassembly

Example:

- 4000 byte datagram
- MTU = 1500 bytes



Fragment Loss

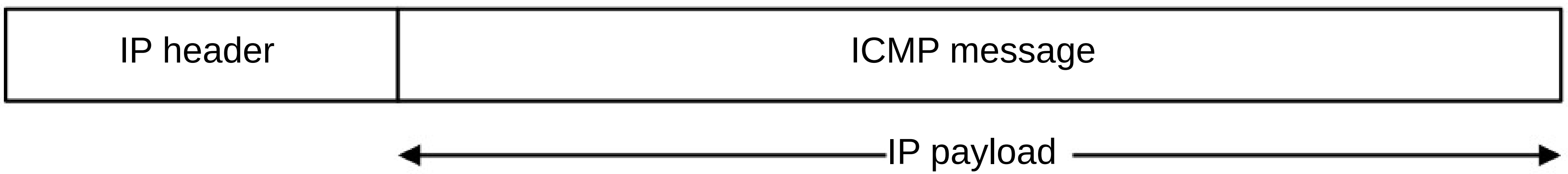
- IP does not guarantee datagram delivery
- Some fragments may be delayed or lost
- Datagrams with lost fragments cannot be reassembled
- If TCP is used in the transport layer the original datagram can be retransmitted
- Fragments may be saved temporarily.
- IP specifies a maximum time to hold fragments.
- After a timer expires, saved fragments are discarded.

Roadmap

- IP: The Internet Protocol
  - IPv4 Addressing
  - Transporting a datagram from source to destination
  - IP Fragmentation & Reassembly
  - ICMP
  - DHCP
  - ARP
  - IPV6
- Routing Algorithms: Dijkstra's Example

ICMP Overview

- The Internet Control Message Protocol (ICMP) is a helper protocol that supports IP with:
  - Error reporting (unreachable host, network, port, protocol)
  - Simple queries (echo request/reply, used by ping)
- ICMP message: type, code plus first 8 bytes of IP datagram causing error
- ICMP messages are encapsulated as IP datagrams:

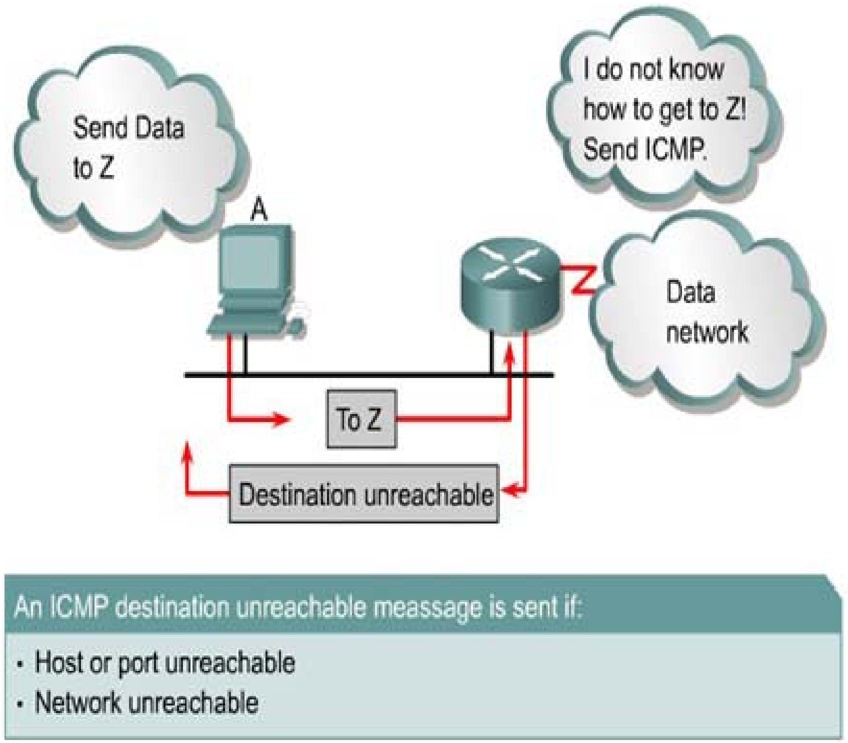


ICMP Message Types

Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion
		control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Examples of errors/problems

- Unreachable Network
  - Sender sends datagram to a non-existent IP address
  - Destination device is disconnected from its network.
  - Router's connecting interface is down
  - Router does not have the information necessary to find the destination network.
- Port Unreachable
  - No process is waiting in destination port of destination host



ICMP use in Traceroute

- Command to determine the active route to a destination address
- How?
  - Send a UDP message to an unused port on the target host with ttl=1
  - When ttl becomes 0, router has to return an ICMP time exceed massage
    - It includes IP address & name of router
  - Traceroute set ttl = 2 and retransmits, this time go one more hop
  - ttl++ until UDP reach the destination
  - The target returns an ICMP service unreachable because there is no UDP port service

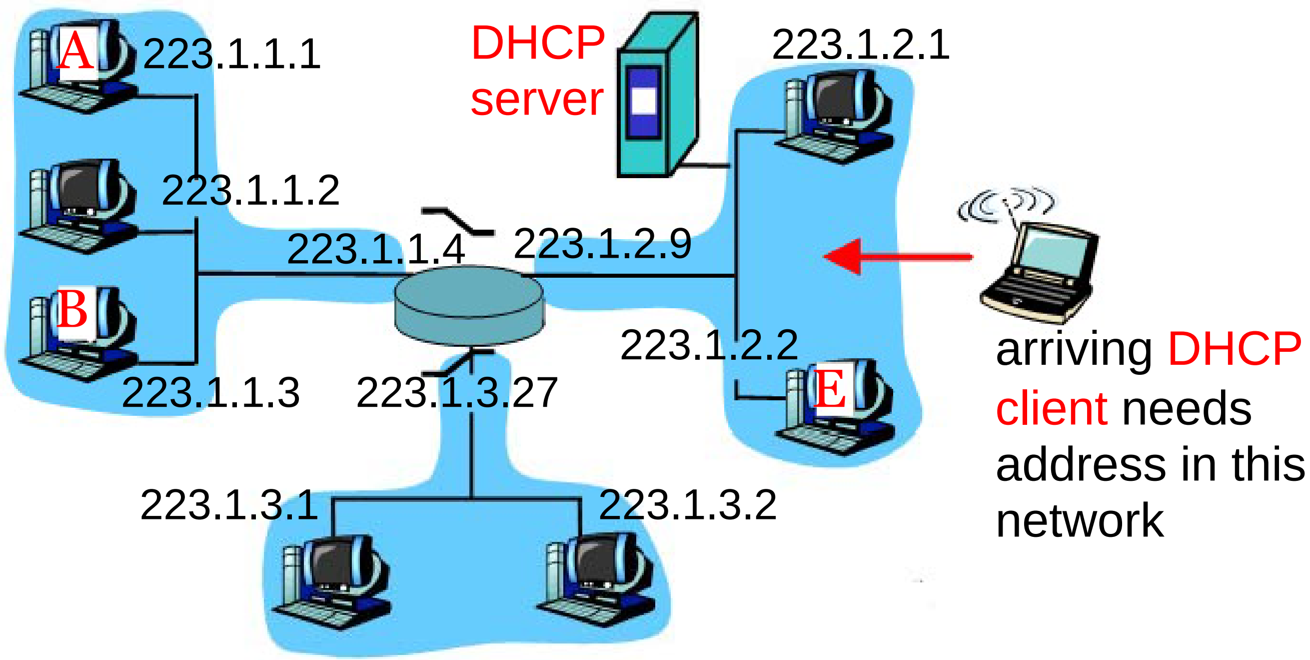
Roadmap

- IP: The Internet Protocol
  - IPv4 Addressing
  - Transporting a datagram from source to destination
  - IP Fragmentation & Reassembly
  - ICMP
  - DHCP
  - ARP
  - IPV6
- Routing Algorithms: Dijkstra's Example

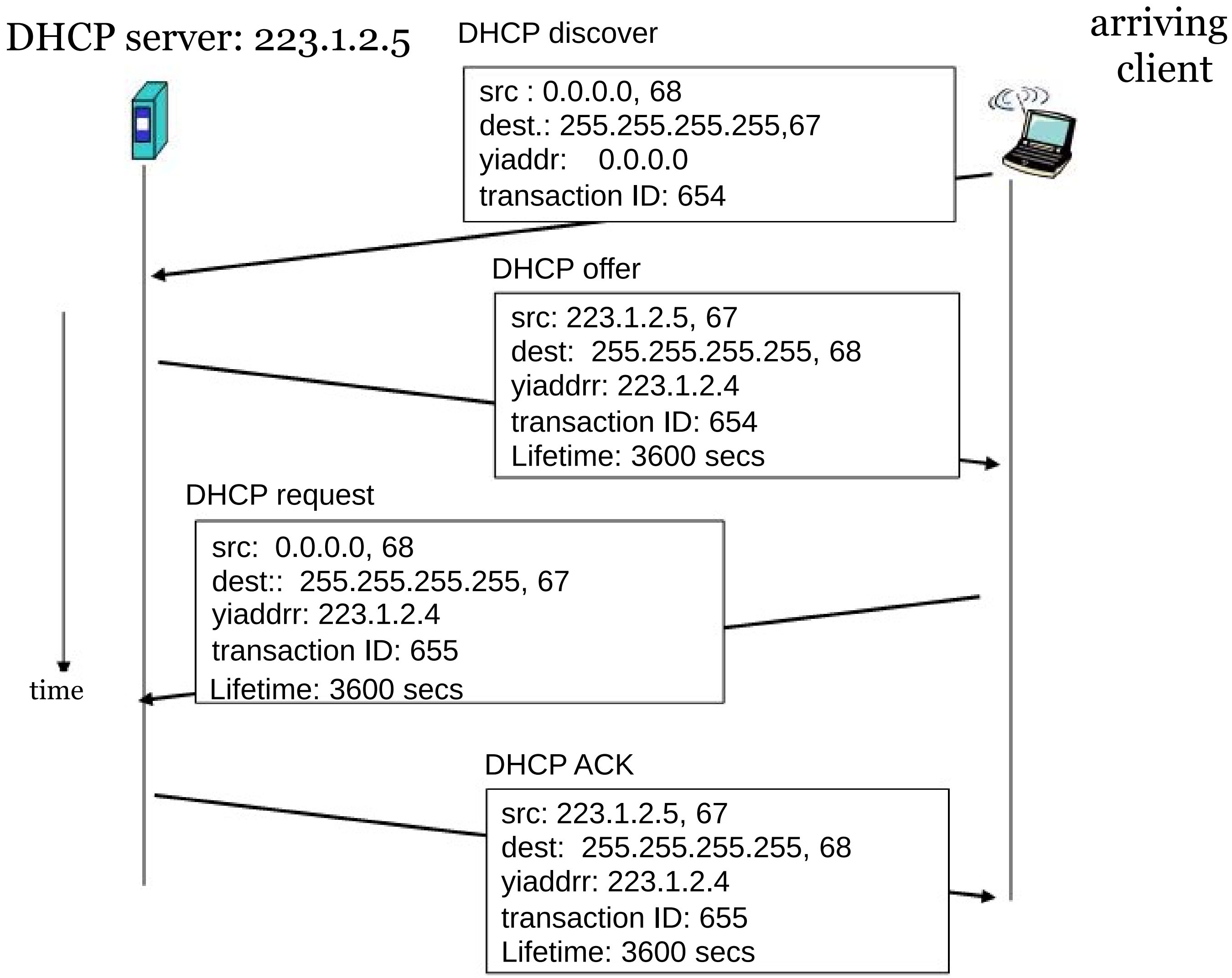
DHCP: Dynamic Host Configuration Protocol

- Allows host to dynamically obtain its IP address from network server when it joins network
  - Can renew its "lease" on address in use
  - Allows reuse of addresses (only hold address while connected)
  - Support for mobile users who want to join networks
- DHCP Overview
  - host broadcasts "DHCP discover" msg
  - DHCP server responds with "DHCP offer" msg
    - Several servers may respond
  - host requests IP address: "DHCP request" msg
  - DHCP server sends address: "DHCP ack" msg

DHCP client-server scenario



DHCP client-server scenario

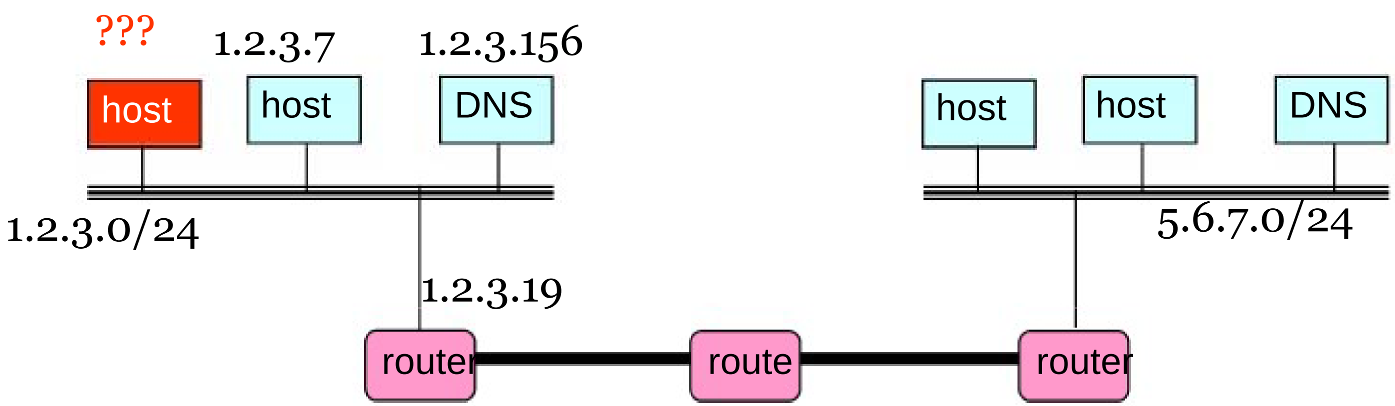


Roadmap

- IP: The Internet Protocol
  - IPv4 Addressing
  - Transporting a datagram from source to destination
  - IP Fragmentation & Reassembly
  - ICMP
  - DHCP
  - ARP
  - IPV6
- Routing Algorithms: Dijkstra's Example

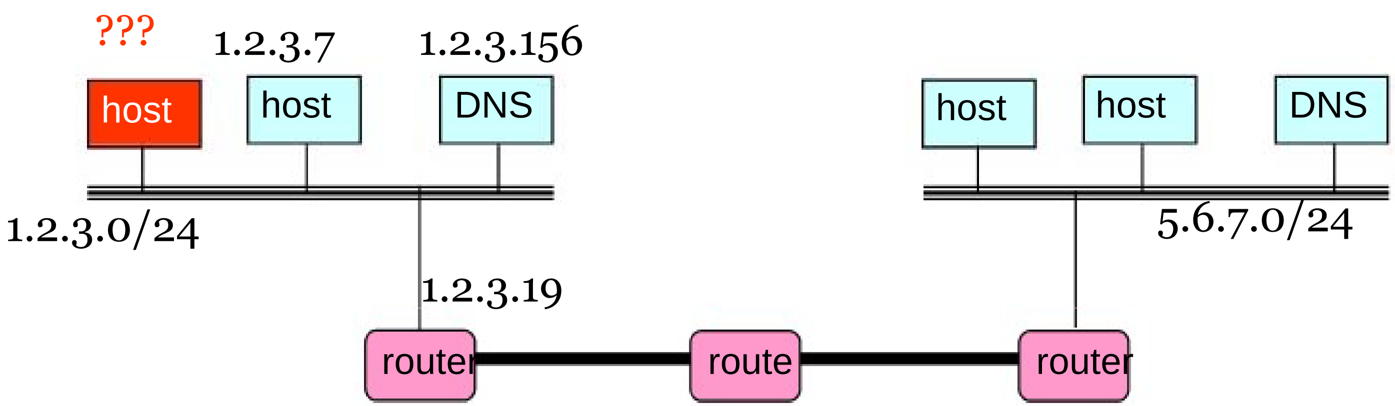
How To Bootstrap an End Host?

- What local Domain Name System server to use?
- What IP address the host should use?
- How to send packets to remote destinations?
- How to ensure incoming packets arrive?

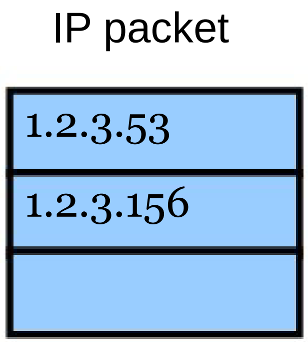


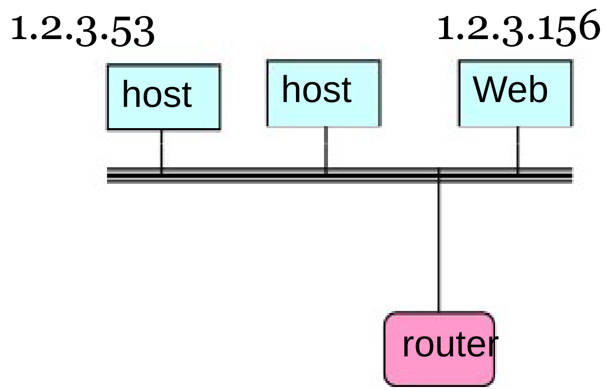
Avoiding Manual Configuration

- Dynamic Host Configuration Protocol (DHCP)
  - End host learns how to send packets
  - Learn IP address, DNS servers, and gateway
- Address Resolution Protocol (ARP)
  - Others learn how to send packets to the end host
  - Learn mapping between IP address & interface address



Sending Packets Over a Link



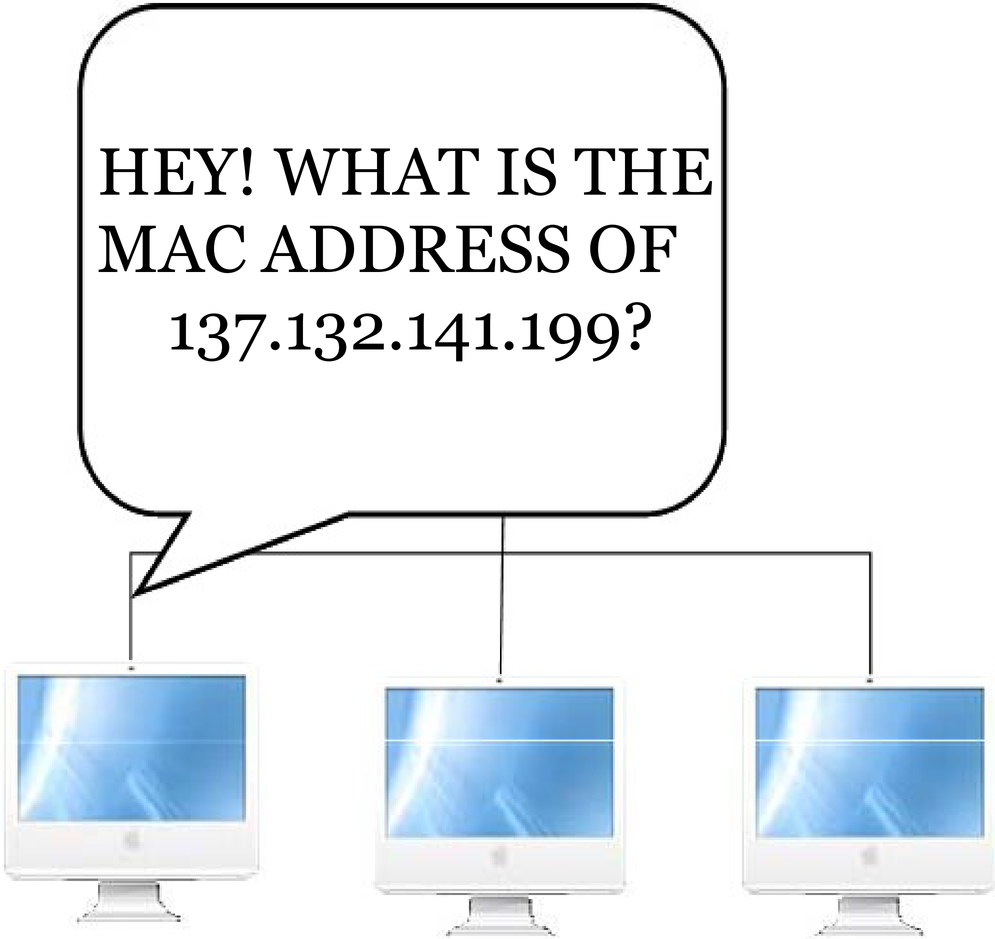


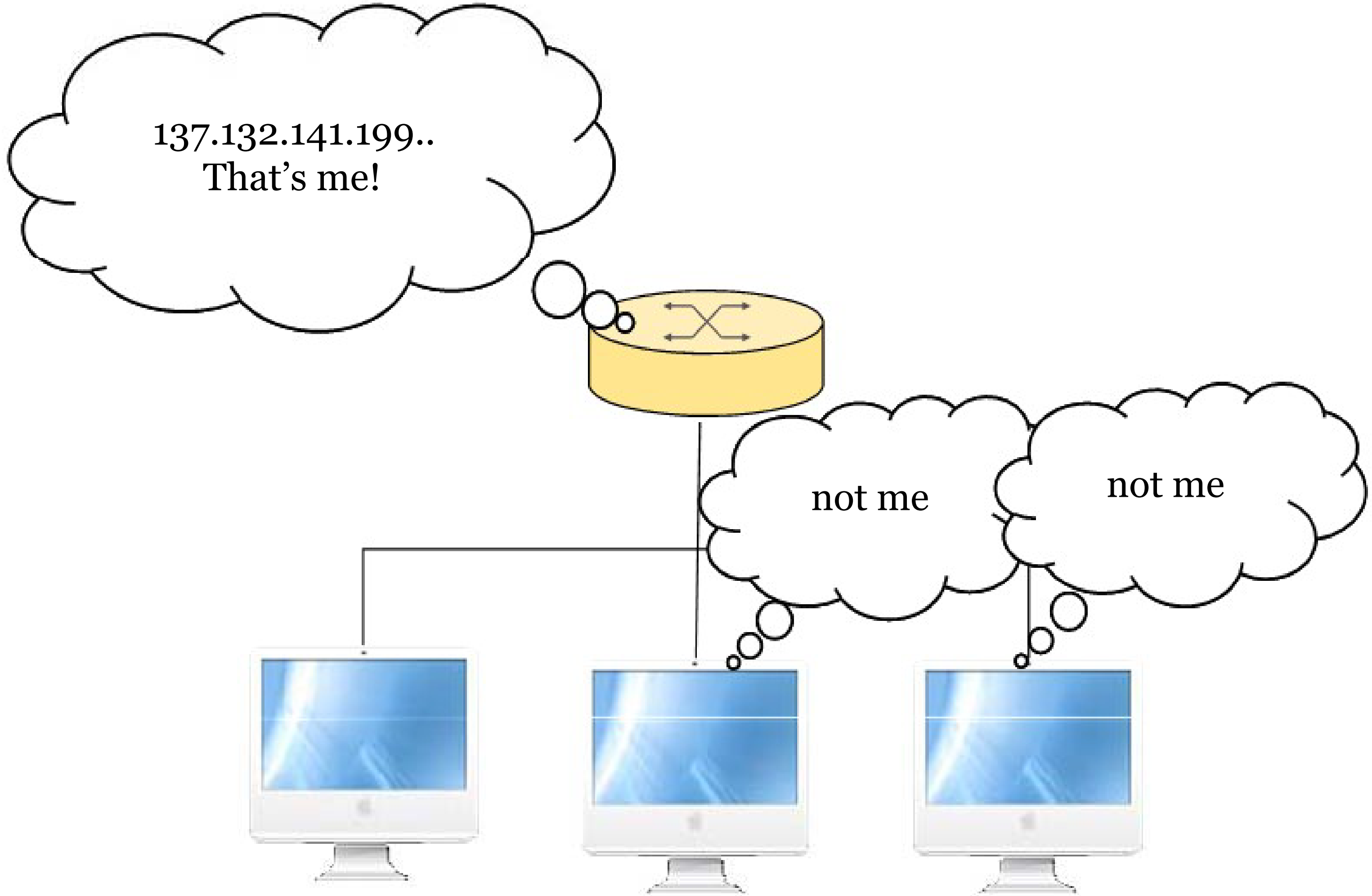
- Adaptors only understand MAC addresses
  - Translate the destination IP address to MAC address
  - Encapsulate the IP packet inside a link-level frame

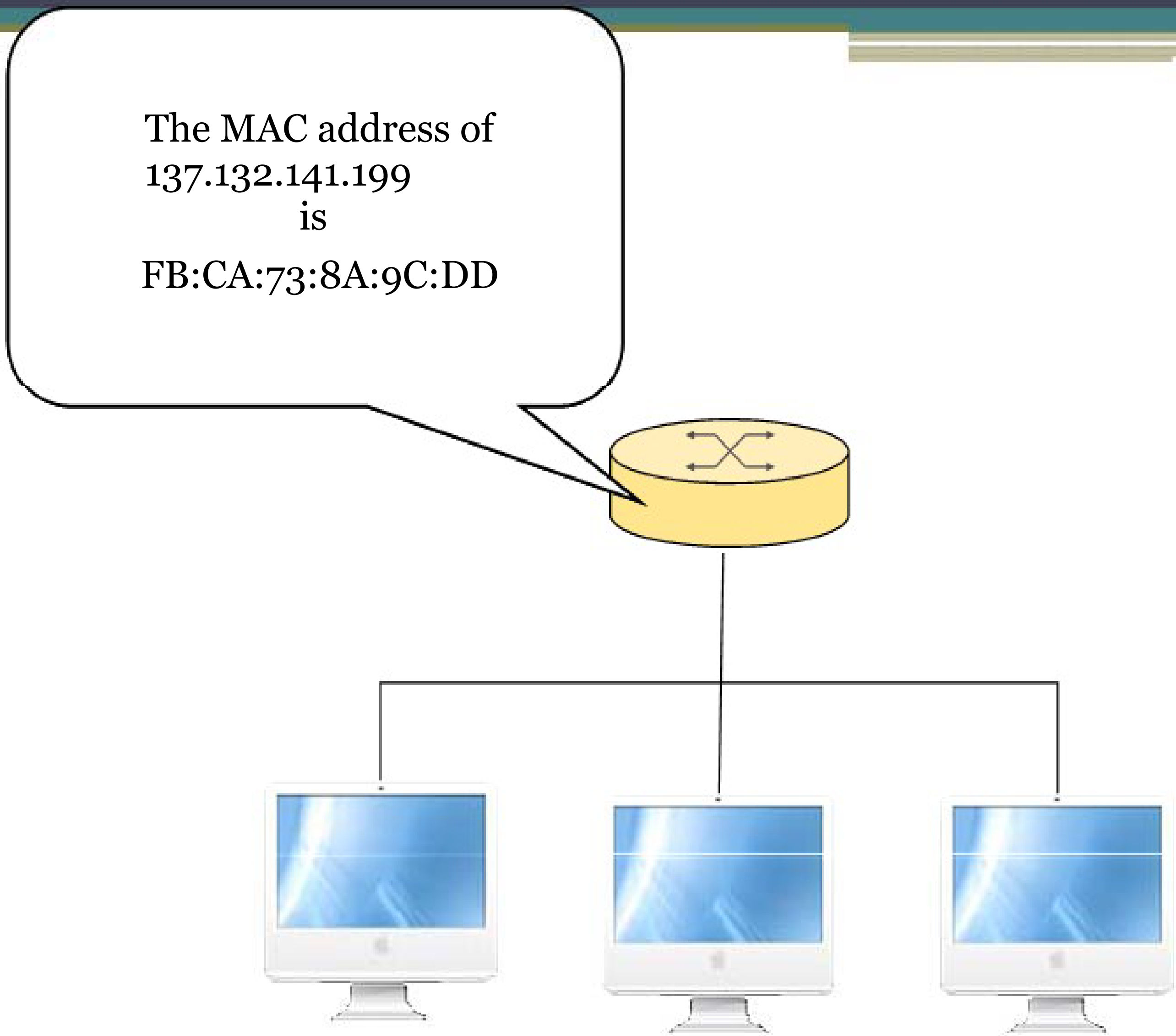
Address Resolution Protocol Table

- Every node maintains an ARP table
  - (IP address, MAC address) pair
- Consult the table when sending a packet
  - Map destination IP address to destination MAC address
  - Encapsulate and transmit the data packet

- But, what if the IP address is not in the table?
  - Sender broadcasts: "Who has IP address 1.2.3.156?" (ARP query)
  - Receiver responds: "MAC address 58-23-D7-FA-20-B0" (unicast)
  - Sender caches the result in its ARP table
    - Entries in ARP table have a timer and an entry is removed when its timer expires
- No need for network administrator to get involved


HEY! WHAT IS THE MAC ADDRESS OF 137.132.141.199?


137.132.141.199..
That's me!
not me
not me


The MAC address of
137.132.141.199
is
FB:CA:73:8A:9C:DD

Roadmap

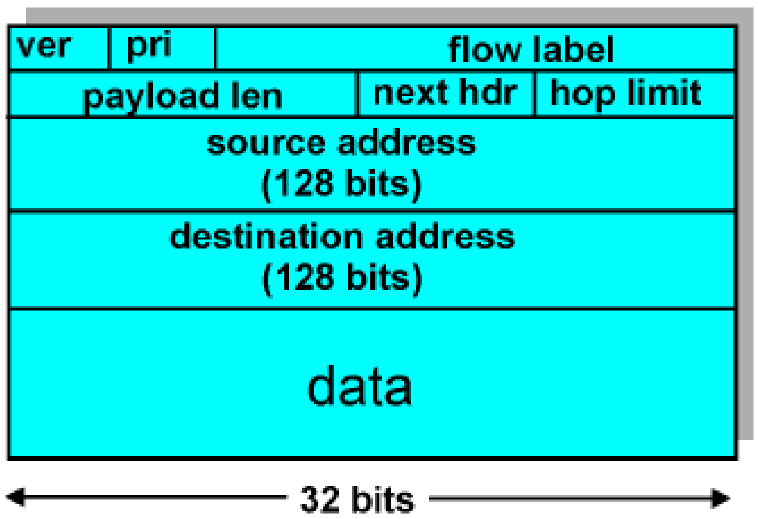
- IP: The Internet Protocol
  - IPv4 Addressing
  - Transporting a datagram from source to destination
  - IP Fragmentation & Reassembly
  - ICMP
  - DHCP
  - ARP
  - IPV6
- Routing Algorithms: Dijkstra's Example

IPv6

- Initial motivation
  - 32-bit address space soon to be completely allocated
  - 2^{32} = 4,294,967,296 (just over four billion)
  - Plus, some are reserved for special purposes
  - Great need for IPs(Computers, PDAs, routers, mobiles..)
- Additional motivation:
  - header format helps speed processing/forwarding
  - header changes to facilitate QoS
- IPv6 has 128-bit addresses ($2^{128} = 3.403 \times 10^{38}$)
  - every grain of sand on the planet can be IP-addressable!
- Short-term solutions: limping along with IPv4
  - Network address translation (NAT)
  - Dynamically-assigned addresses (DHCP)
- IPv6 datagram format:
  - fixed-length 40 byte header
  - no fragmentation allowed

IPv6 Header

- Priority: identify priority among datagrams in flow or give priority to datagrams from certain apps (ICMP)
- Flow Label: identify datagrams in same "flow."
  - Special handling for some flows (e.g. real time app.)
  - Flows of high priority users (paying for better service)
- Next header: identify upper layer protocol for data (TCP/UDP)



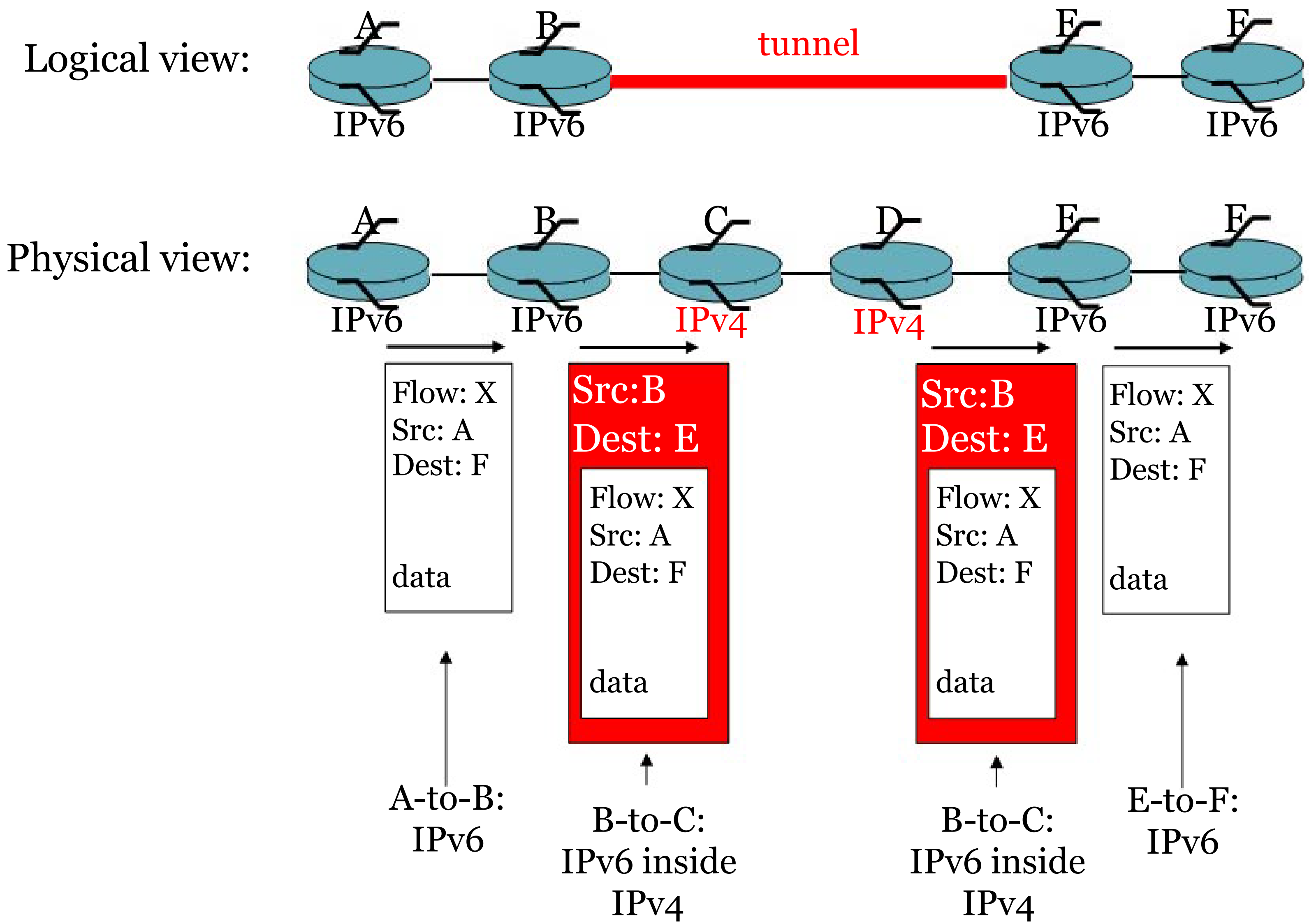
Other Changes from IPv4

- Checksum: removed entirely to reduce processing time at each hop (after change of TTL)
- Options: allowed, but outside of header, pointed to by "Next Header" field
- ICMPv6: new version of ICMP
  - additional message types, e.g. "Packet Too Big"
  - multicast group management functions

Transition From IPv4 To IPv6

- Not all routers can be upgraded simultaneous
  - no “flag days” & huge size of Internet
  - How will the network operate with mixed IPv4 and IPv6 routers?
- Dual Stack approach
  - IPv6 nodes also have a complete IPv4 implementation
  - Nodes must have both IPv6 & IPv4 addresses
  - Must be able to determine if other nodes are IPv6 capable
- Tunneling: entire IPv6 packet carried as payload in IPv4 datagram among IPv4 routers

Tunneling



Logical view:
A
B
tunnel
E
F
IPv6
IPv6
IPv6
IPv6
Physical view:
A
B
C
D
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6
Flow: X
Src: A
Dest: F
data
Src:B
Dest: E
Flow: X
Src: A
Dest: F
data
Src:B
Dest: E
Flow: X
Src: A
Dest: F
data
Flow: X
Src: A
Dest: F
data
A-to-B:
IPv6
B-to-C:
IPv6 inside
IPv4
B-to-C:
IPv6 inside
IPv4
E-to-F:
IPv6

Roadmap

- IP: The Internet Protocol
  - IPv4 Addressing
  - Transporting a datagram from source to destination
  - IP Fragmentation & Reassembly
  - ICMP
  - DHCP
  - ARP
  - IPV6
- Routing Algorithms: Dijkstra's Example

Link-State (LS) Routing Algorithm

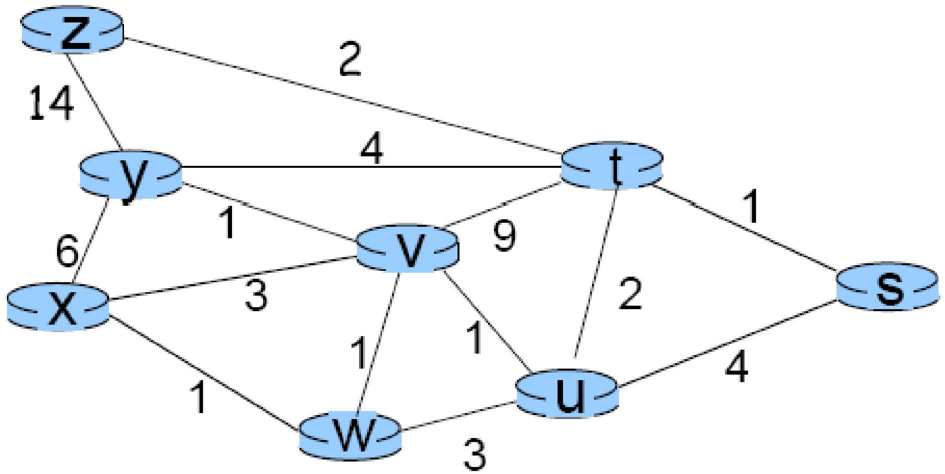
Dijkstra's algorithm

- topology and link cost known to all nodes
  - accomplished via "link state broadcast"
  - all nodes have same info
- computes least cost paths from one node (source) to all other nodes
  - gives forwarding table for that node
- iterative: after k iterations, know least cost path to k destination nodes

Notation:

- **c(x,y)**: link cost from node x to y; set to infinite if a and y are not direct neighbours
- **D(v)**: current value of cost of path from source to dest. v
- **p(v)**: v's predecessor node along path from source to v
- **N'**: set of nodes whose least cost path is definitively known

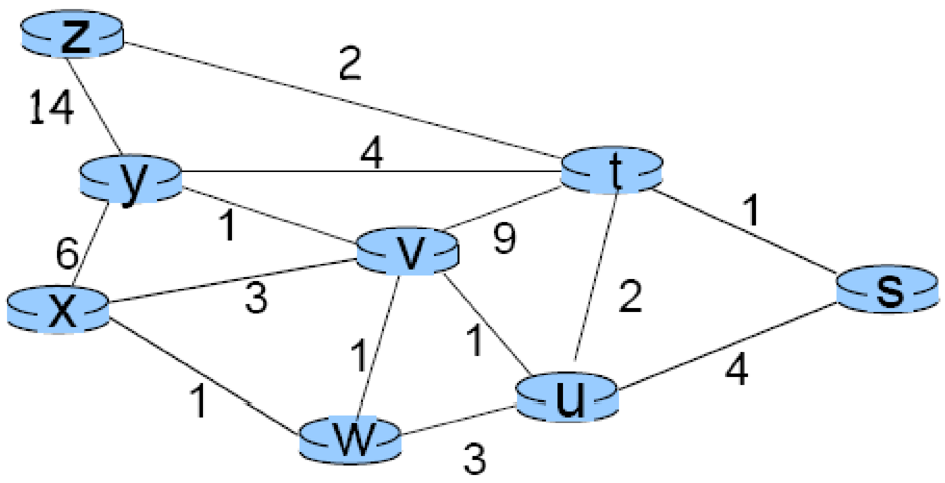
Dijkstra’s Algorithm Example



Step	N’	D(s),p(s)	D(t),p(t)	D(u),p(u)	D(v),p(v)	D(w),p(w)	D(y),p(y)	D(z),p(z)
0	x	∞	∞	∞	3,x	1,x	6,x	∞

- Initialization:
  - Store source node x in N’
  - Assign link cost to neighbours (v,w,y)
  - Keep track of predecessor to destination node

Dijkstra's Algorithm Example (cont.)

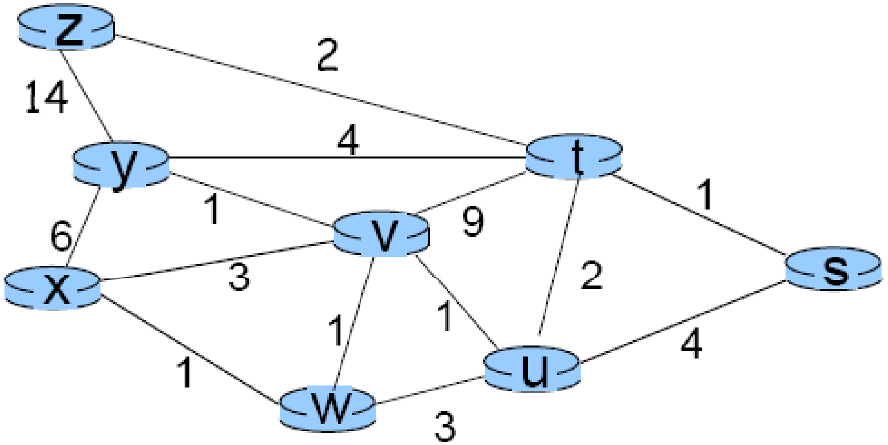


Node and its minimum cost are colour-coded in each step

Step	N'	D(s),p(s)	D(t),p(t)	D(u),p(u)	D(v),p(v)	D(w),p(w)	D(y),p(y)	D(z),p(z)
0	x	∞	∞	∞	3,x	1,x	6,x	∞
1	xw	∞	∞	4,w	2,w		6,x	∞

- Loop – step 1:
  - For all nodes not in N', find one that has minimum cost path (1)
  - Add this node (w) to N'
  - Update cost for all neighbours of added node that are not in N'
- repeat until all nodes are in N'

Dijkstra’s Algorithm Example (cont.)



We can now build x’s forwarding table. E.g. the entry to s will be constructed by looking at predecessors along shortest path:
6,t 5,u 3,v 2,w (direct link)
So forward to s via w

Step	*N’*	*D(s),p(s)*	*D(t),p(t)*	*D(u),p(u)*	*D(v),p(v)*	*D(w),p(w)*	*D(y),p(y)*	*D(z),p(z)*
0	x	∞	∞	∞	3,x	1,x	6,x	∞
1	xw	∞	∞	4,w	2,w		6,x	∞
2	xwv	∞	11,v	3,v			3,v	∞
3	xwvu	7,u	5,u				3,v	∞
4	xwvuy	7,u	5,u					17,y
5	xwvuyt	6,t						7,t
6	xwvuyts							7,t
7	xwvuytsz							