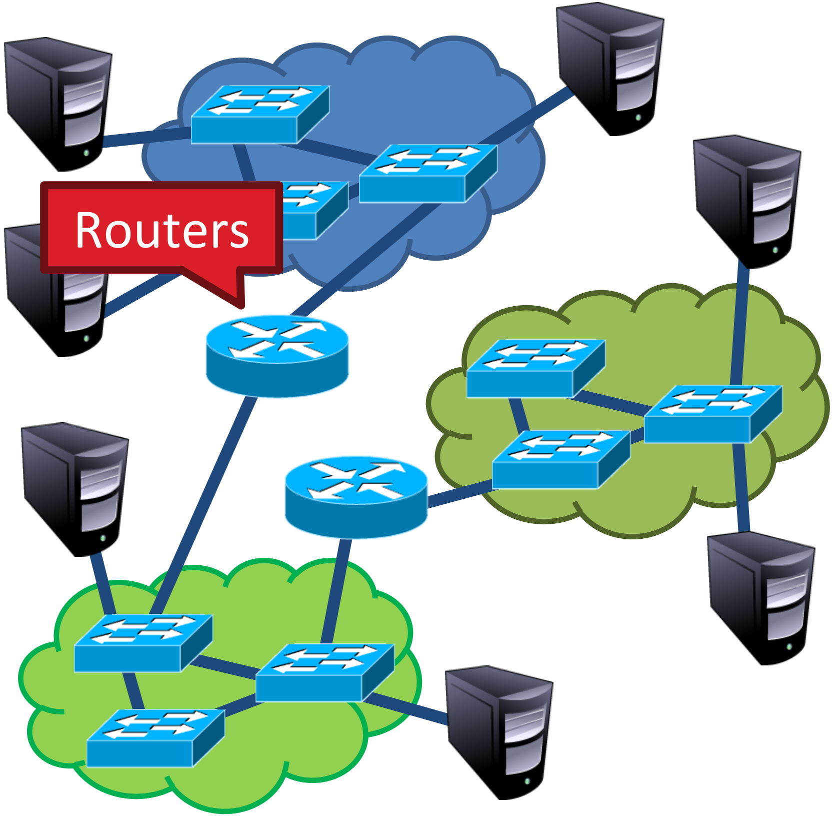
IF2230 Jaringan Komputer Network (IP) Layer

Internetworking

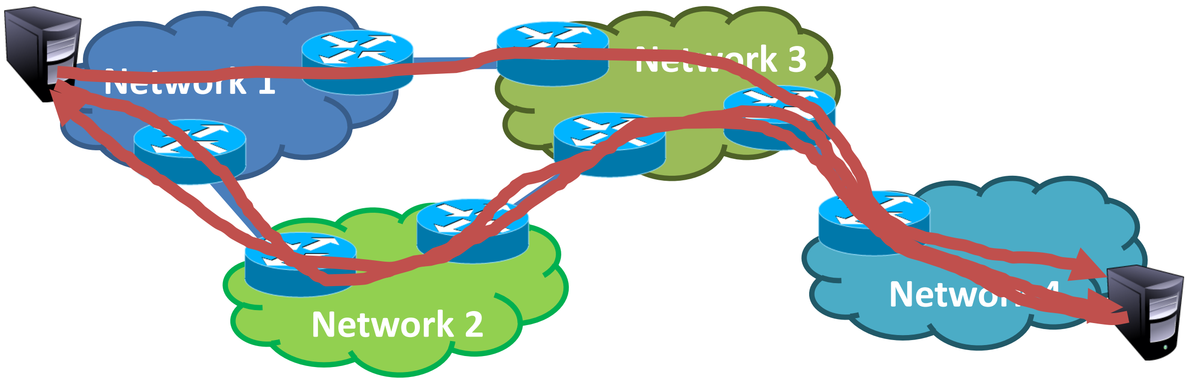
Robithoh Annur
Andreas Bara Timur
Monterico Andrian

Introductions

- How to connect multiple LANs?
- LANs may be incompatible
  - Ethernet, Wifi, etc…
- Connected networks form an internetwork
  - The Internet is the best known example



Structure of the Internet



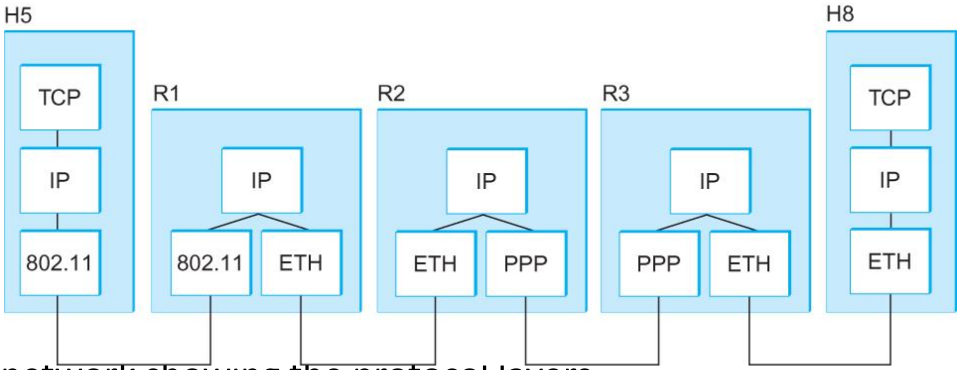
- Ad-hoc interconnection of networks
  - No organized topology
  - Vastly different technologies, link capacities
- Packets travel end-to-end by hopping through networks
  - Routers "peer" (connect) different networks
  - Different packets may take different routes

Outline

- Addressing
  - Class-based
  - CIDR
  - IP forwarding
  - NAT
- IPv4 Protocol Details
  - Packed Header
  - Fragmentation
- IPv6

Internet Protocol

- 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 and defines the infrastructure that allows these nodes and networks to function as a single logical internetwork



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

IP Addressing

- Globally unique (for “public” IP addresses)
- IP address: IPv4 32-bit identifier for host, router *interface*
- *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
- Usually written in dotted notation, e.g. 192.168.21.76
- Each number is a byte
- Stored in Big Endian order

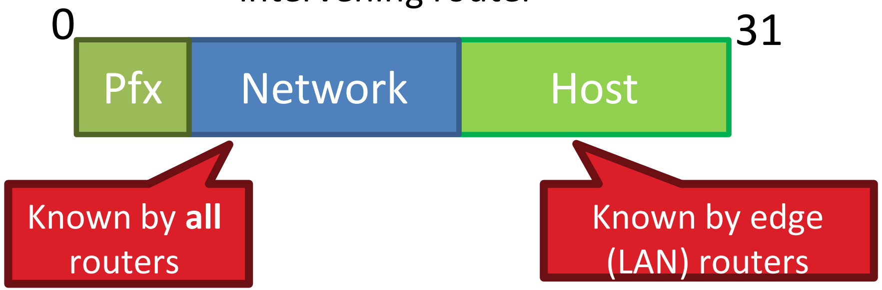
	0	8	16	24 31
Decimal	192	168	21	76
Hex	C0	A8	15	4C
Binary	11000000	10101000	00010101	01001100

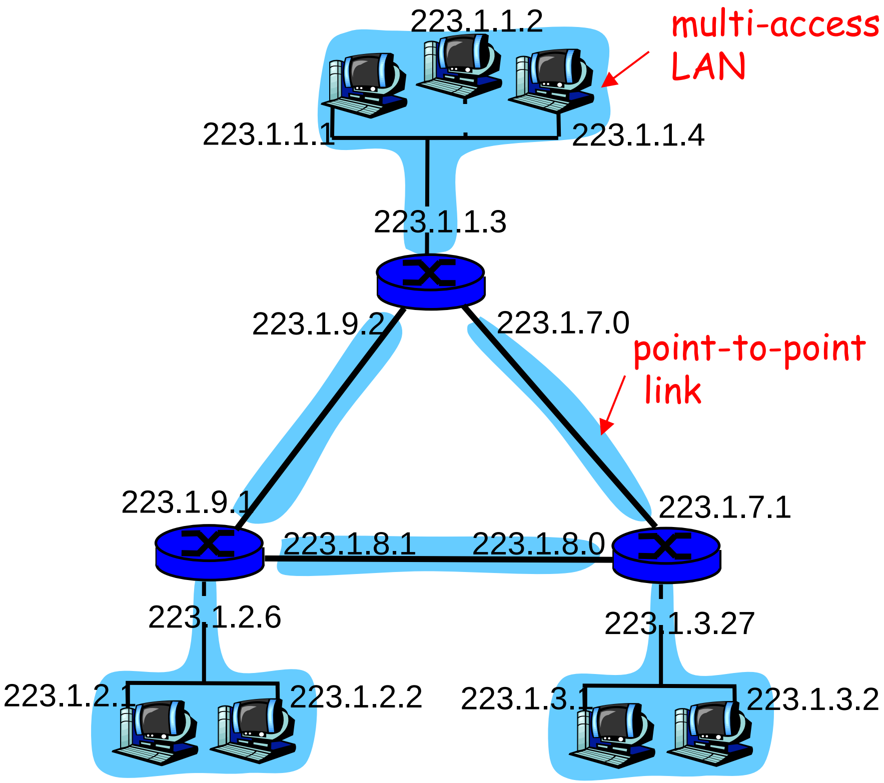
IP Addressing: Network vs. Host

- Two-level hierarchy →Separate the address into a network and a host
  - network part (high order bits)
  - host part (low order bits)
- *What's a network ?*

(from IP address perspective)

  - device interfaces with same network part of IP address
  - can physically reach each other without intervening router





"Classful" IP Addressing

class	0–7	8–15	16–23	24–31	range	hosts
A	0network	host			1.0.0.0 to 127.255.255.255	$2^{24} - 2 = 16{,}777{,}214$ (All 0 and all 1 are reserved)
B	10 network		host		128.0.0.0 to 191.255.255.255	$2^{16} - 2 = 65{,}534$ (All 0 and all 1 are reserved)
C	110 network			host	192.0.0.0 to 223.255.255.255	$2^{8} - 2 = 254$ (All 0 and all 1 are reserved)
D	1110 multicast address				224.0.0.0 to 239.255.255.255	

← 32 bits →

- Disadvantage: 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

Classless Addressing

CIDR: Classless InterDomain Routing

- 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
- 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: CIDR

CIDR: Classless InterDomain Routing

- Network portion of address is of arbitrary length
- Addresses allocated in contiguous blocks
  - Number of addresses assigned always power of 2
- Address format: a.b.c.d/x
  - x is number of bits in network portion of address

network part | host part

11001000 00010111 00010000 00000000

200.23.16.0/23

Classless Addressing

- CIDR tries to balance the desire to minimize the number of routes that a router needs to know against the need to hand out addresses efficiently.

- 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

Representation of Address Blocks

- “Human Readable” address format: a.b.c.d/x
  - x is number of bits in network portion of address, the network portion is also called the network prefix
- machine representation of a network (addr block):
  using a combination of
  - first IP of address blocks of the network
  - network mask (x “1”’s followed by 32-x “0”’s

network w/ address block: 200.23.16.0/23

first IP address of address block:

11001000 00010111 00010000 00000000

network mask:

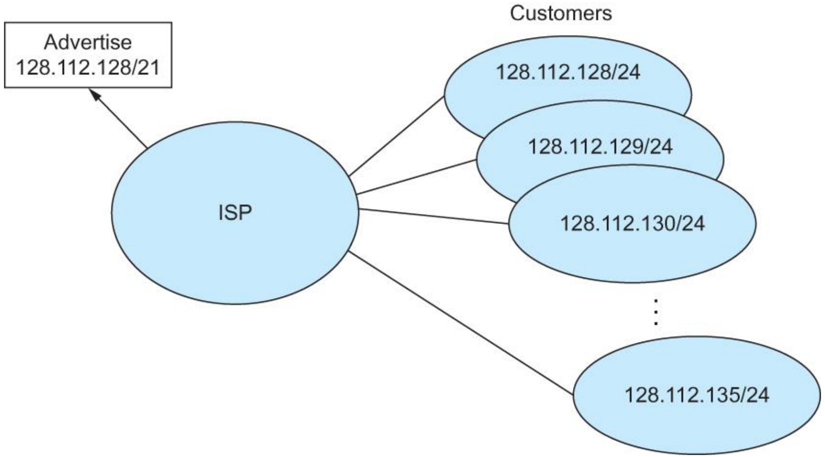
11111111 11111111 11111110 00000000

Classless Addressing

- Consider an Autonomous System (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
- 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, which is 24 bits long, we would write it 192.4.16/24

Classless Addressing



Route aggregation with CIDR

IP Addresses: How to Get One? …

Q: How does a *network* get network part of IP addr?

A: gets an allocated portion of its provider ISP's address space

ISP's block	11001000 00010111 00010000 00000000	200.23.16.0/20
Organization 0	11001000 00010111 00010000 00000000	200.23.16.0/23
Organization 1	11001000 00010111 00010010 00000000	200.23.18.0/23
Organization 2	11001000 00010111 00010100 00000000	200.23.20.0/23
...	….. ….	….
Organization 7	11001000 00010111 00011110 00000000	200.23.30.0/23

Host Configurations

- Notes
  - 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
    - Configuration process is error-prune
  - Automated Configuration Process is required

Dynamic Host Configuration Protocol (DHCP)

Goal: allow host to *dynamically* obtain its IP address from network DHCP server when it joins network

- Can renew its lease on address in use
- Allows reuse of addresses (only hold address while connected as “on”)
- Support for mobile users who want to join network (more shortly)

- 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

Outline

- Addressing
  - Class-based
  - CIDR
  - IP forwarding
  - NAT
- IPv4 Protocol Details
  - Packed Header
  - Fragmentation
- IPv6

IP Forwarding: Destination in Same Net

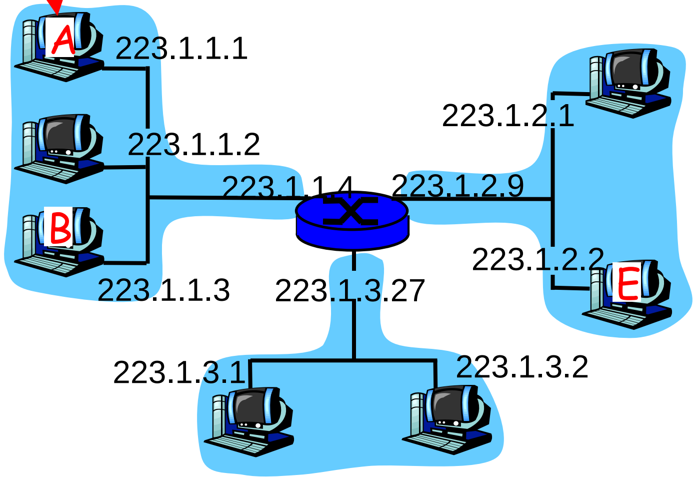
misc fields	223.1.1.1	223.1.1.3	data

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:

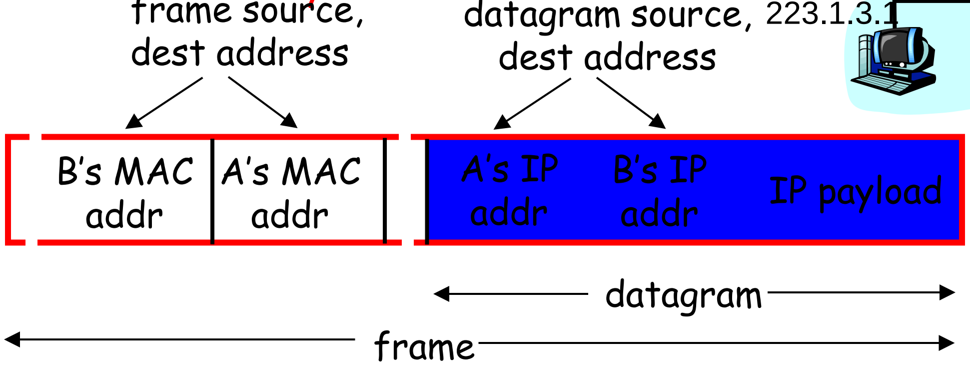
- look up net. address of B in forwarding table
- find B is on same net. as A
- link layer will send datagram directly to B inside link-layer frame
  - B and A are directly connected

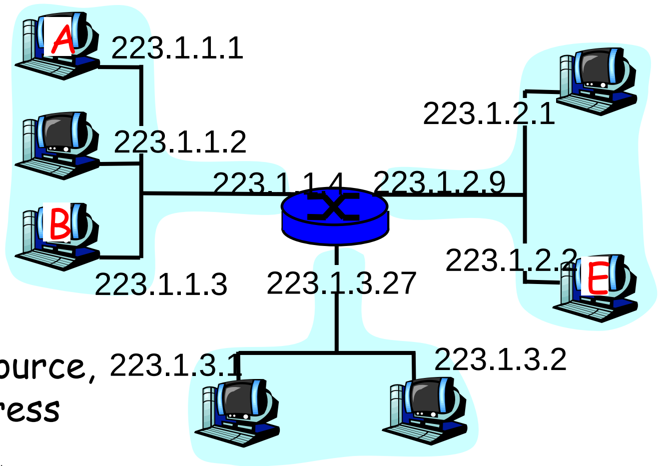


IP Datagram Forwarding on Same LAN: Interaction of IP and data link layers

Starting at A, given IP datagram addressed to B:

- look up net. address of B, find B on same net. as A
- link layer send datagram to B inside link-layer frame





MAC (Physical) Addresses -- Revisited

- used to get frames from one interface to another physically-connected interface (same physical network, i.e., p2p or LAN)
- 48 bit MAC address (for most LANs)
  - fixed for each adaptor, burned in the adapter ROM
  - MAC address allocation administered by IEEE
    - 1st bit: 0 unicast, 1 multicast.
    - all 1’s : broadcast
- MAC flat address -> portability
  - can move LAN card from one LAN to another
- MAC addressing operations on a LAN:
  - each adaptor on the LAN “sees” all frames
  - accept a frame if dest. MAC address matches its own MAC address
  - accept all broadcast (MAC= all1’s) frames
  - accept all frames if set in “promiscuous” mode
  - can configure to accept certain multicast addresses (first bit = 1)

MAC vs. IP Addresses

32-bit IP address:

- *network-layer* address, logical
  - i.e., not bound to any physical device, can be re-assigned
- IP hierarchical address NOT portable
  - depends on IP network to which an interface is attached
  - when move to another IP network, IP address re-assigned
- used to get IP packets to destination IP network
  - Recall how IP datagram forwarding is performed
- IP network is "virtual," actually packet delivery done by the underlying physical networks
  - from source host to destination host, hop-by-hop via IP routers
  - over each link, different link layer protocol used, with its own frame headers, and source and destination MAC addresses
    - Underlying physical networks do not understand IP protocol and datagram format!

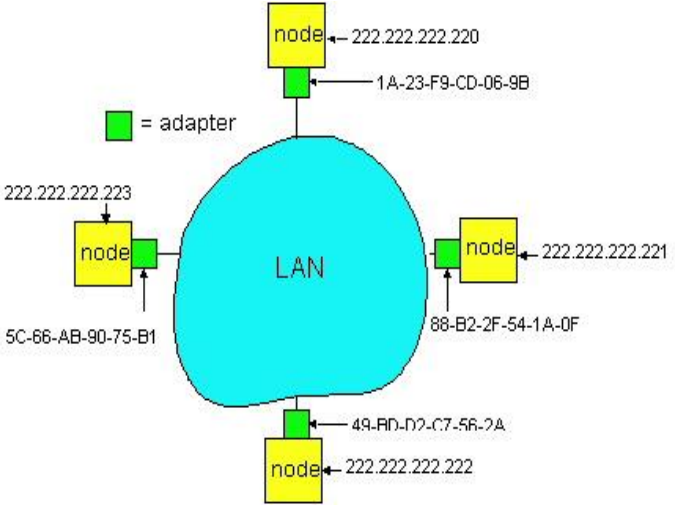
ARP: Address Resolution Protocol

- Each IP node (host, router) on LAN has ARP table
- ARP Table: IP/MAC address mappings for some LAN nodes

< IP address; MAC address; timer>

  - timer: time after which address mapping will be forgotten (typically 15 min)

Question: how to determine MAC address of B knowing B's IP address?



What does ARP do?

- The main functions of ARP
  - Obtaining the MAC address of an destination IP.
  - Forming the ARP table with lookup entry of “destination IP to MAC address”
- Issued by a host OS that tries to obtain the MAC address of an destination IP (automatically).
- After obtaining the MAC address of the “desired destination IP” thru ARP, the host will use the information to form an entry in the ARP table (or ARP cache)
  - Remember that the Frame MUST have the destination MAC address before sending out thru the wire.
- There are two parts of the ARP
  - ARP request (issued by the source host)
  - ARP reply (issued by the destination host)

```
D:\Documents and Settings\Administrator>arp -a

Interface: 172.16.10.16 --- 0x4
  Internet Address      Physical Address      Type
  172.16.10.1           00-15-f9-04-57-93     dynamic
  172.16.10.3           00-15-fa-a4-99-4a     dynamic
  172.16.10.129         00-13-46-32-af-45     dynamic
```

ARP Table

ARP Protocol

- A wants to send datagram to B, and A knows B’s IP address.
- A looks up B’s MAC address in its ARP table
- Suppose B’s MAC address is not in A’s ARP table.
- A broadcasts (why?) ARP query packet, containing B's IP address
  - all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
  - frame sent to A’s MAC address (unicast)
- A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
  - soft state: information that times out (goes away) unless refreshed
- ARP is “plug-and-play”:
  - nodes create their ARP tables without intervention from net administrator

ARP Messages

<table>
<tr><th>0</th><th>8</th><th>16</th><th>24 … 31</th></tr>
<tr><td colspan="2">HARDWARE ADDRESS TYPE</td><td colspan="2">PROTOCOL ADDRESS TYPE</td></tr>
<tr><td>HADDR LEN</td><td>PADDR LEN</td><td colspan="2">OPERATION</td></tr>
<tr><td colspan="4">SENDER HADDR (first 4 octets)</td></tr>
<tr><td colspan="2">SENDER HADDR (last 2 octets)</td><td colspan="2">SENDER PADDR (first 2 octets)</td></tr>
<tr><td colspan="2">SENDER PADDR (last 2 octets)</td><td colspan="2">TARGET HADDR (first 2 octets)</td></tr>
<tr><td colspan="4">TARGET HADDR (last 4 octets)</td></tr>
<tr><td colspan="4">TARGET PADDR (all 4 octets)</td></tr>
</table>

Hardware Address Type: e.g., Ethernet
Protocol address Type: e.g., IP
Operation: ARP request or ARP response

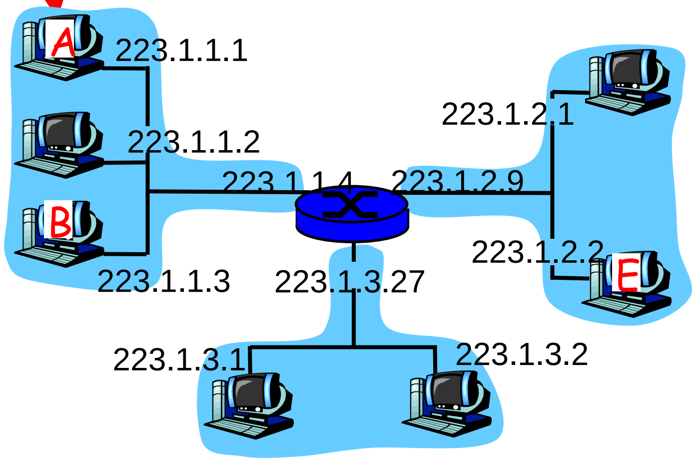
IP Forwarding: Destination in Different Network

misc fields	223.1.1.1	223.1.2.3	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



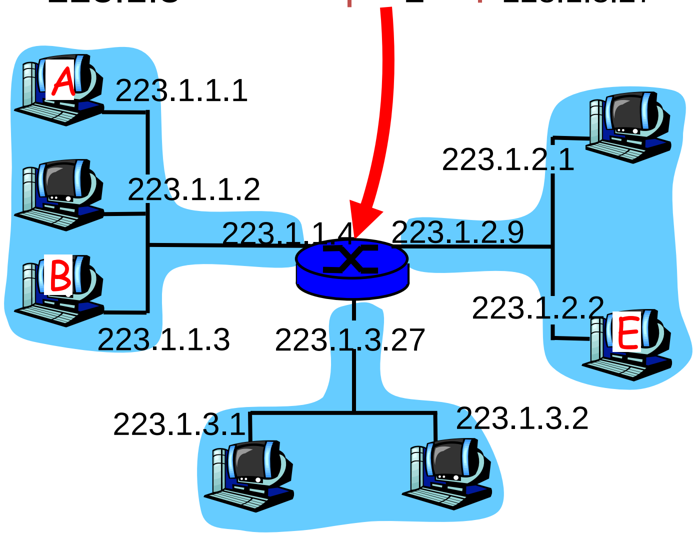
IP Forwarding: Destination in Diff. Net …

misc fields	223.1.1.1	223.1.2.3	data

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

- 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!!! (hooray!)

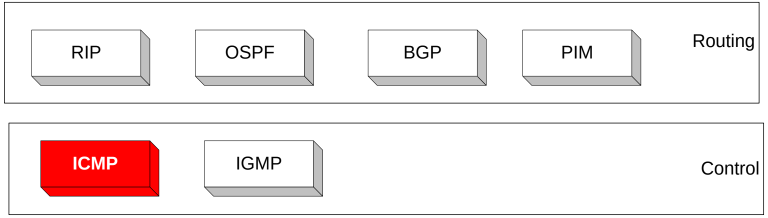


IP Forwarding Revisited

- IP forwarding mechanism assumes that it can find the network number in a packet and then look up that number in the forwarding table
- We need to change this assumption in case of CIDR
- CIDR means that prefixes may be of any length, from 2 to 32 bits
- 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 match”
    - 171.69.10 in this case
- A packet destined to 171.69.20.5 would match 171.69 and not 171.69.10

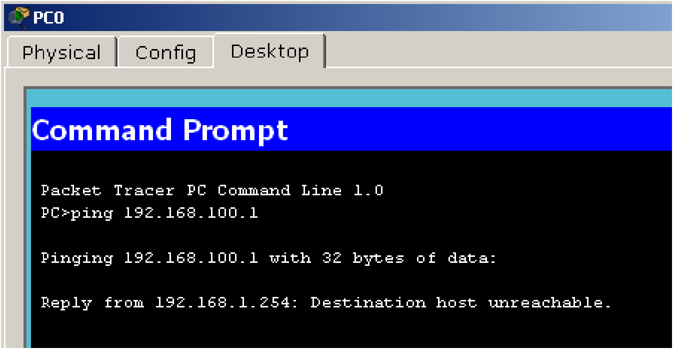
Internet Control Message Protocol (ICMP)

- The IP (Internet Protocol) relies on several other protocols to perform necessary control and routing functions:
  - Control functions (ICMP)
  - Multicast signaling (IGMP)
  - Setting up routing tables (RIP, OSPF, BGP, PIM, …)

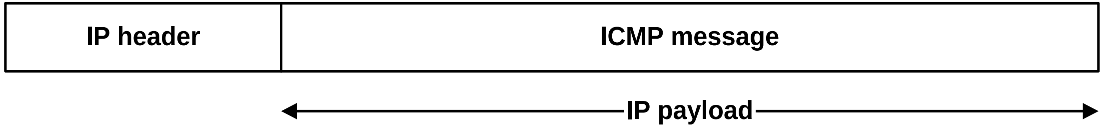


- The **Internet Control Message Protocol (ICMP)** is a helper protocol that supports IP with facility for
  – Error reporting
  – Simple queries

example



- ICMP messages are encapsulated as IP datagrams:



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

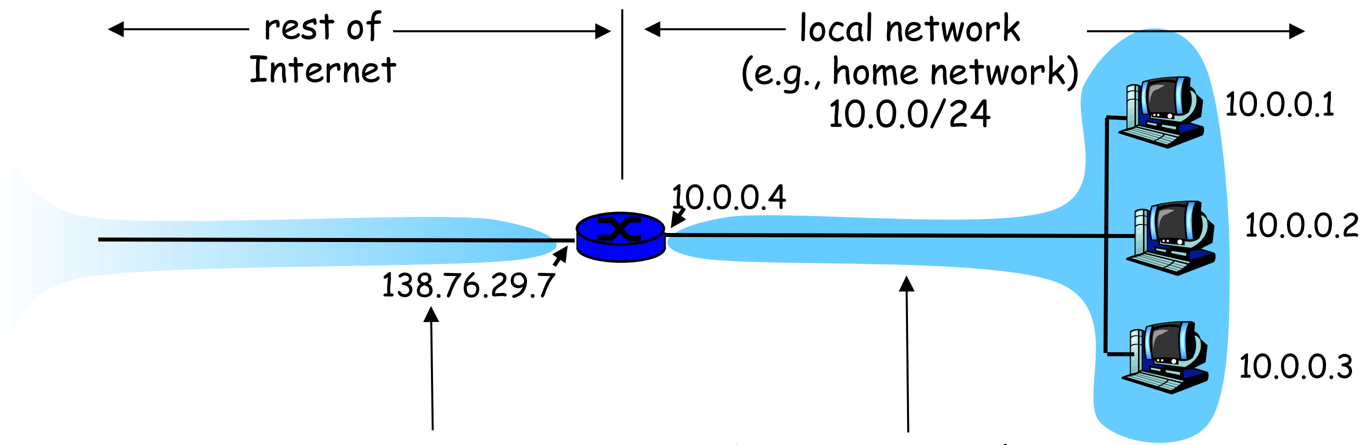
Frequent ICMP Error message

Type	Code	Description	
3	0–15	Destination unreachable	Notification that an IP datagram could not be forwarded and was dropped. The code field contains an explanation.
5	0–3	Redirect	Informs about an alternative route for the datagram and should result in a routing table update. The code field explains the reason for the route change.
11	0, 1	Time exceeded	Sent when the TTL field has reached zero (Code 0) or when there is a timeout for the reassembly of segments (Code 1)
12	0, 1	Parameter problem	Sent when the IP header is invalid (Code 0) or when an IP header option is missing (Code 1)

Outline

- Addressing
  - Class-based
  - CIDR
  - IP forwarding
  - NAT
- IPv4 Protocol Details
  - Packed Header
  - Fragmentation
- IPv6

NAT: Network Address Translation



All datagrams *leaving* local network have *same* single source NAT IP address: 138.76.29.7, different source port numbers

Datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

10.0.0.0/8 has been reserved for private networks!

Why do we need this “NAT”?

- NAT is used for three major reasons:
  - IPv4 address exhaustion
  - Masquerading for security purpose
  - TCP load sharing
- NAT for alleviating the consequences of IPv4 address exhaustion.
  - It has become a standard, indispensable feature in routers for home and small-office Internet connections.
  - One public IP can be used by thousands of private network computers.
- NAT as IP masquerading
  - Obscures an internal network's structure,
  - All network traffic appears to outside network as if it is originated from the one IP address of a router.
- NAT for TCP load sharing
  - Useful for server farm
  - A few servers with similar functions represented by one single IP address.

NAT: Network Address Translation

- Then: local network uses just one IP address as far as outside world is concerned:
  - no need to be allocated range of addresses from ISP: - just one IP address is used for all devices
  - can change addresses of devices in local network without notifying outside world
  - can change ISP without changing addresses of devices in local network
  - devices inside local net not explicitly addressable, visible by outside world (a security plus).

NAT: Network Address Translation

Implementation: NAT router must:

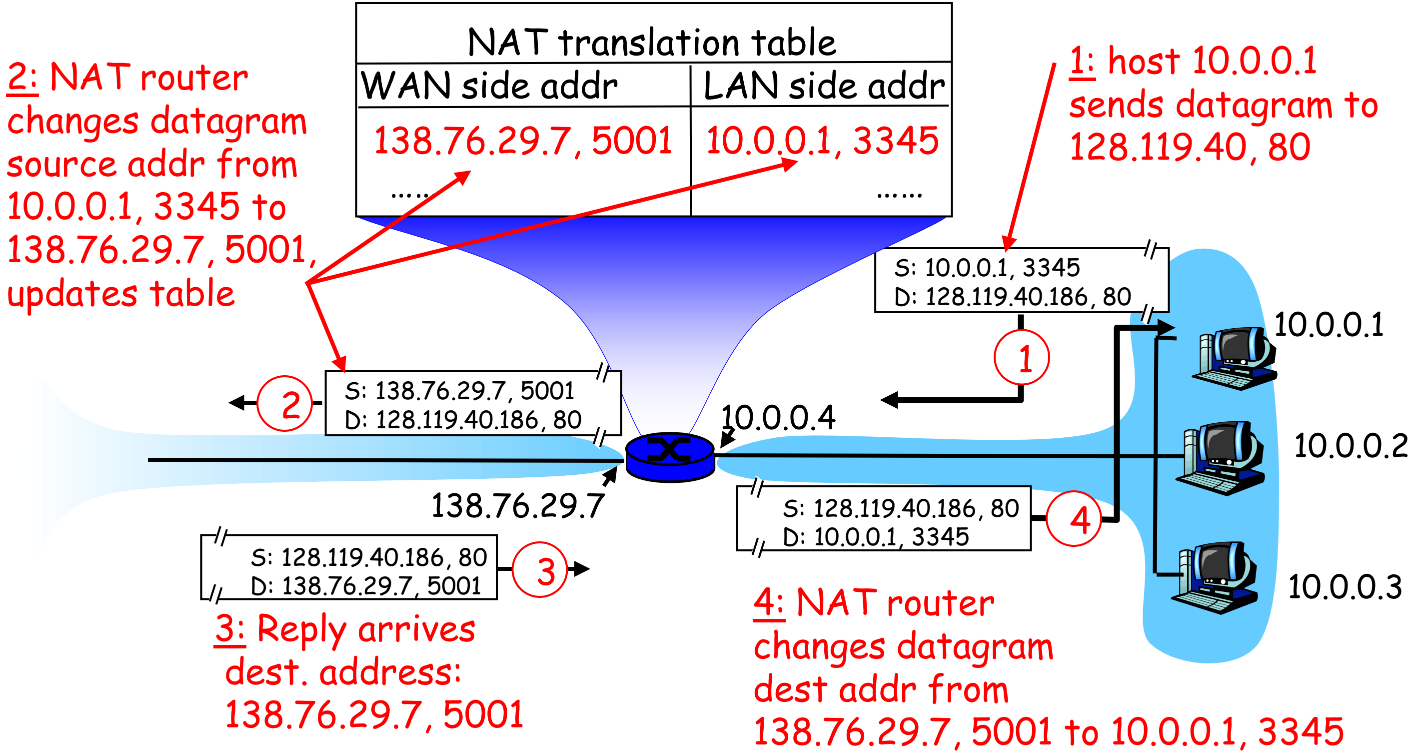
- *outgoing datagrams: replace* (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)

  . . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr.

- *remember (in NAT translation table)* every (source IP address, port #) to (NAT IP address, new port #) translation pair

- *incoming datagrams: replace* (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

NAT: Network Address Translation



Four Types of NAT

- Static Network Address Translation (static NAT)
  - 1 private IP to 1 global IP address translation

- Dynamic Network Address Translation (dynamic NAT)
  - Many private IP to many global IP address translation (a pool of IP)

- Port Address Translation (PAT)
  - Many private IP to 1 global IP address translation.
  - Is also called NAT overloading.
  - 2 sub-mode: Interface mode & pool mode

- Port Forwarding (Type of static NAT)
  - Accessing a inside local network service from outside global host.

More Details in in the practical labs

Outline

- Addressing
  - Class-based
  - CIDR
  - IP forwarding
  - NAT
- IPv4 Protocol Details
  - Packed Header
  - Fragmentation
- IPv6

IP Datagrams

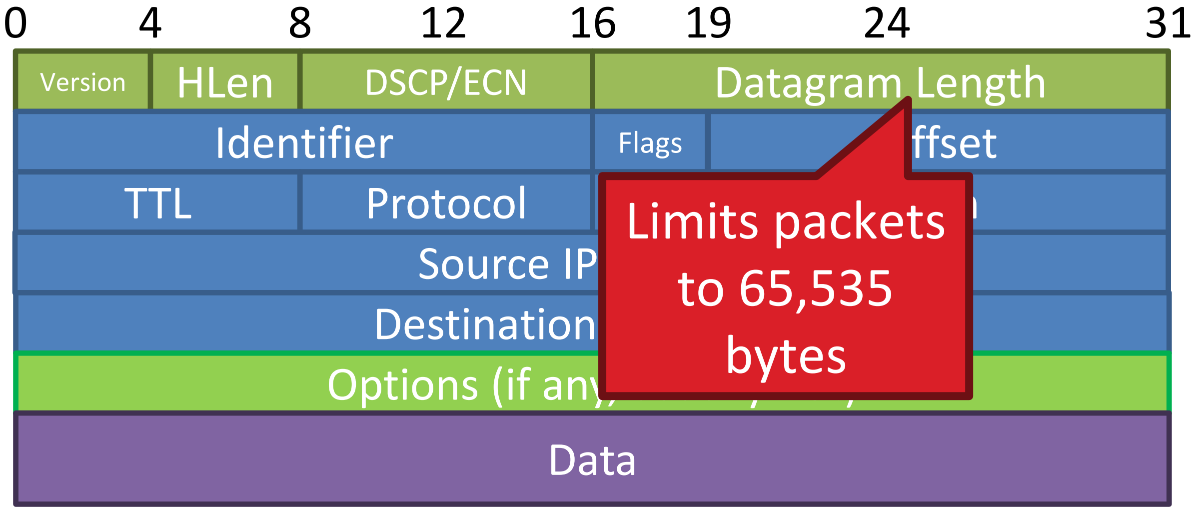
- IP Datagrams are like a letter
  - Totally self-contained
  - Include all necessary addressing information
  - No advanced setup of connections or circuits

0 4 8 12 16 19 24 31

<table>
<tr><td>Version</td><td>HLen</td><td>DSCP/ECN</td><td colspan="3">Datagram Length</td></tr>
<tr><td colspan="3">Identifier</td><td>Flags</td><td colspan="2">Offset</td></tr>
<tr><td colspan="2">TTL</td><td>Protocol</td><td colspan="3">Checksum</td></tr>
<tr><td colspan="6">Source IP Address</td></tr>
<tr><td colspan="6">Destination IP Address</td></tr>
<tr><td colspan="6">Options (if any, usually not)</td></tr>
<tr><td colspan="6">Data</td></tr>
</table>

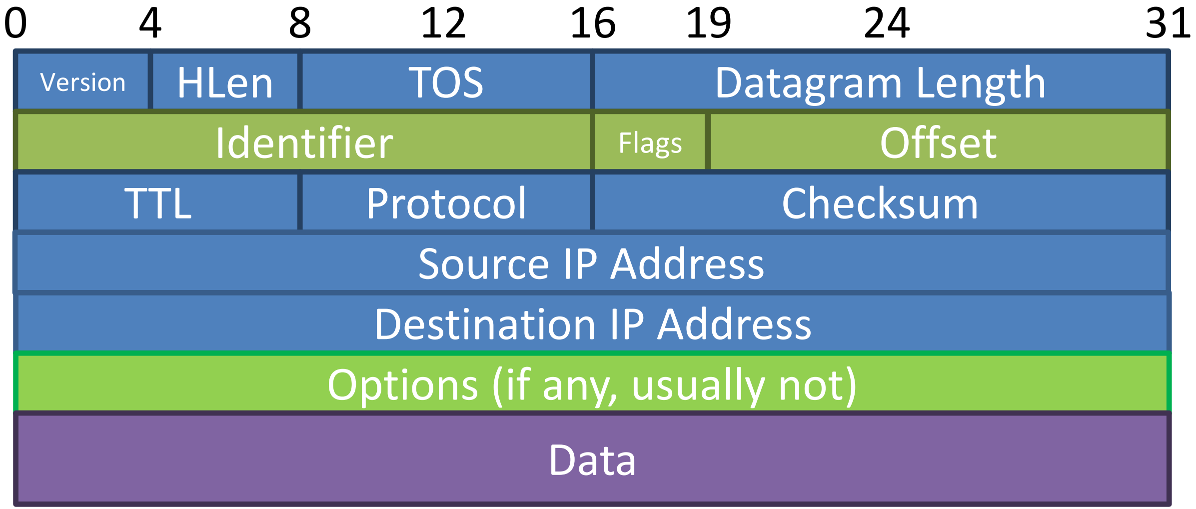
IP Header Fields: Word 1

- Version: 4 for IPv4
- Header Length: Number of 32-bit words (usually 5)
- Type of Service: Priority information (unused)
- Datagram Length: Length of header + data in bytes



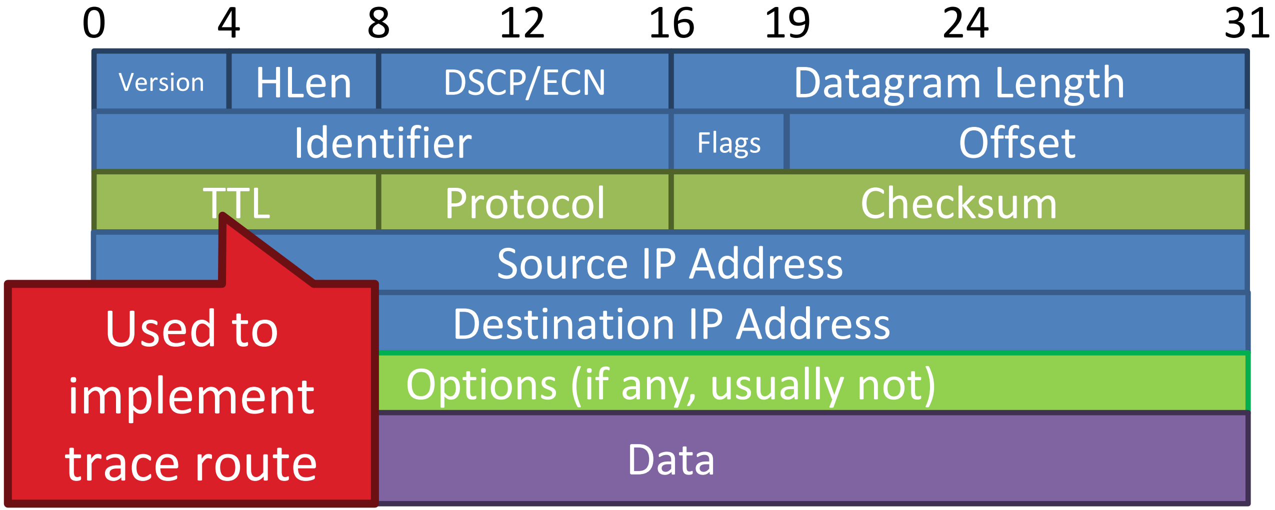
IP Header Fields: Word 2

- Identifier: a unique number for the original datagram
- Flags: M flag, i.e. this is the last fragment
- Offset: byte position of the first byte in the fragment
  - Divided by 8



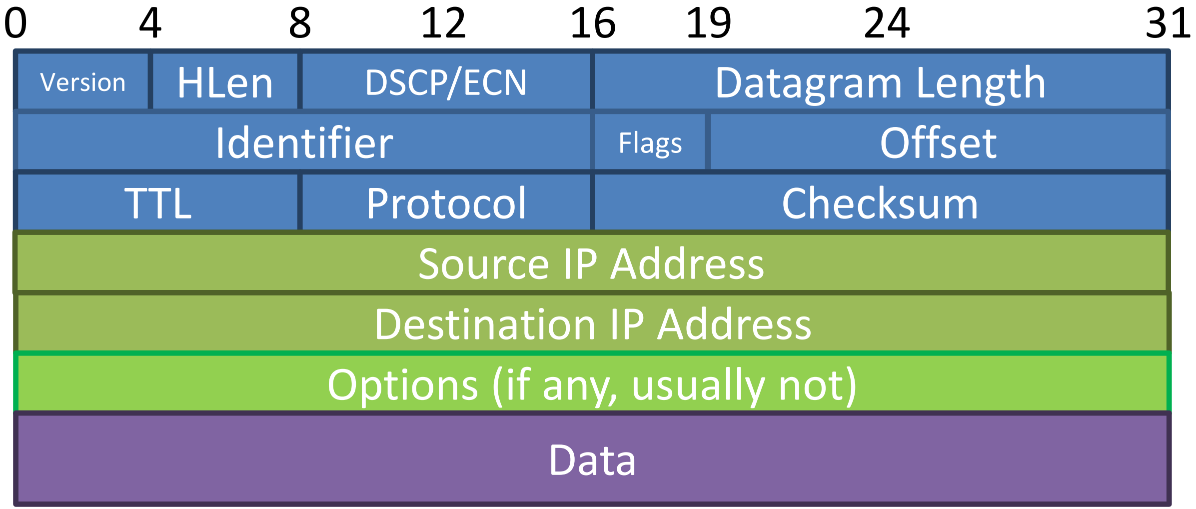
IP Header Fields: Word 3

- Time to Live: decremented by each router
  - Used to kill looping packets
- Protocol: ID of encapsulated protocol
  - 6 = TCP, 17 = UDP
- Checksum



IP Header Fields: Word 4 and 5

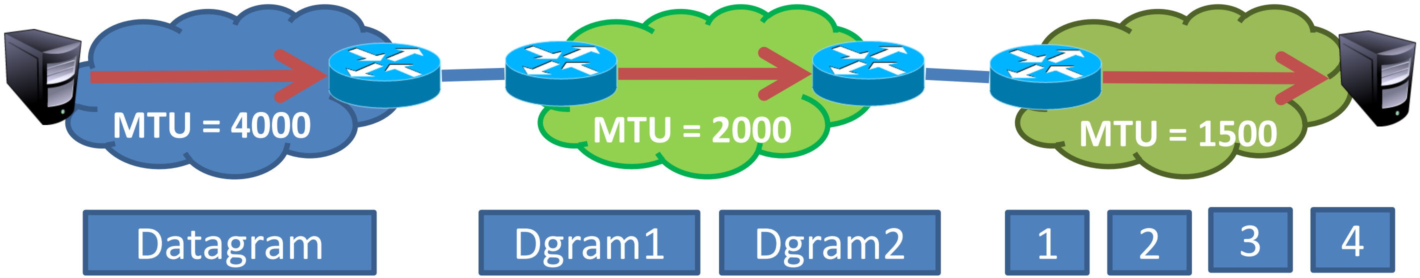
- Source and destination address
  - In theory, must be globally unique
  - In practice, this is often violated



Outline

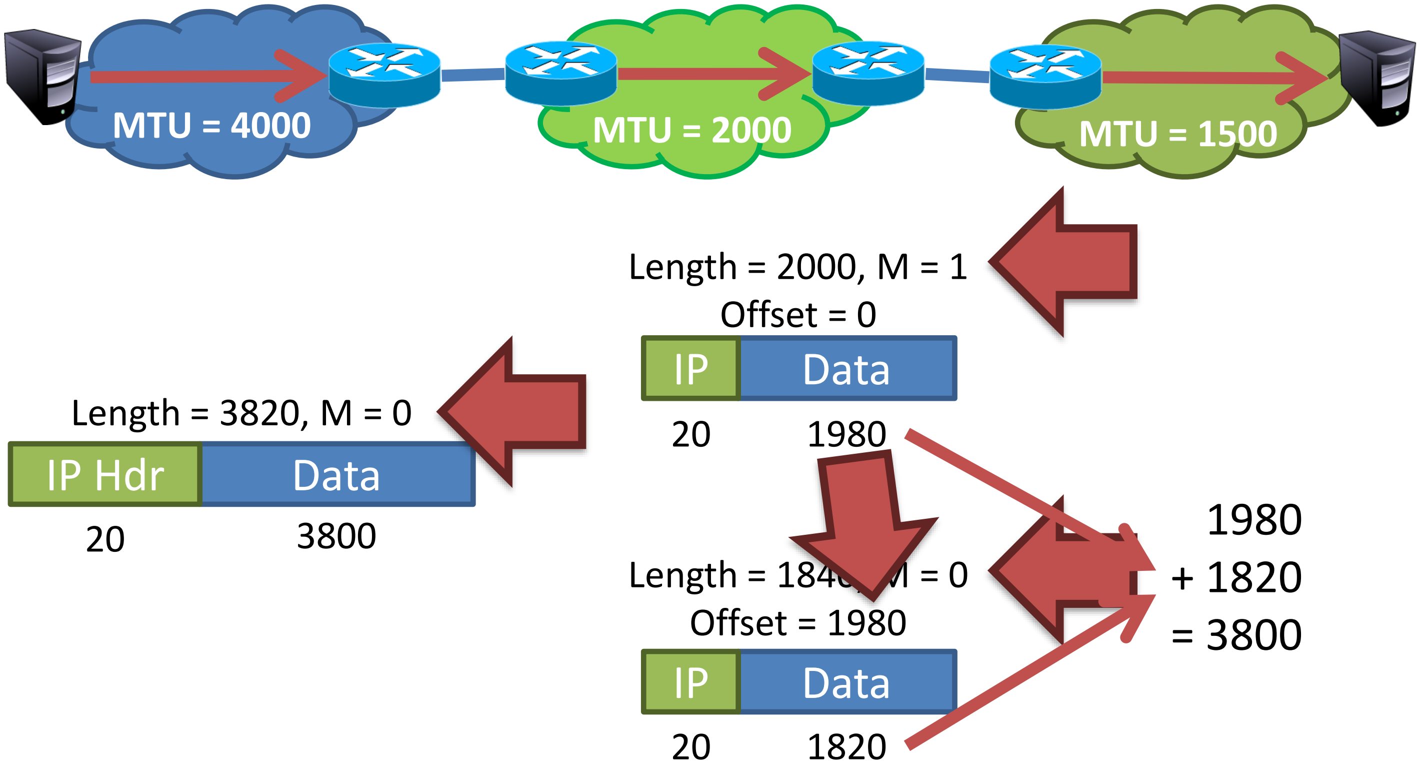
- Addressing
  - Class-based
  - CIDR
  - IP forwarding
  - NAT
- IPv4 Protocol Details
  - Packed Header
  - Fragmentation
- IPv6

Problem: Fragmentation

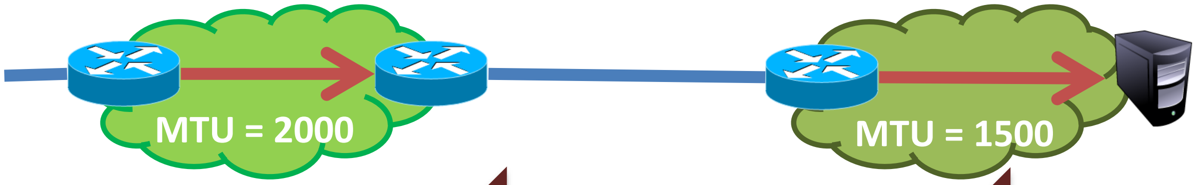


- Problem: each network has its own MTU
  - DARPA principles: networks allowed to be heterogeneous
  - Minimum MTU may not be known for a given path
- IP Solution: fragmentation
  - Split datagrams into pieces when MTU is reduced
  - Reassemble original datagram at the receiver

Fragmentation Example



Fragmentation Example



Length = 2000, M = 1
Offset = 0

IP	Data
20	1980

Length = 1840, M = 0
Offset = 1980

IP	Data
20	1820

Length = 1500, M = 1
Offset = 0

IP	Data
20	1480

Length = 520, M = 1
Offset = 1480

IP	Data
20	500

Length = 1500, M = 1
Offset = 1980

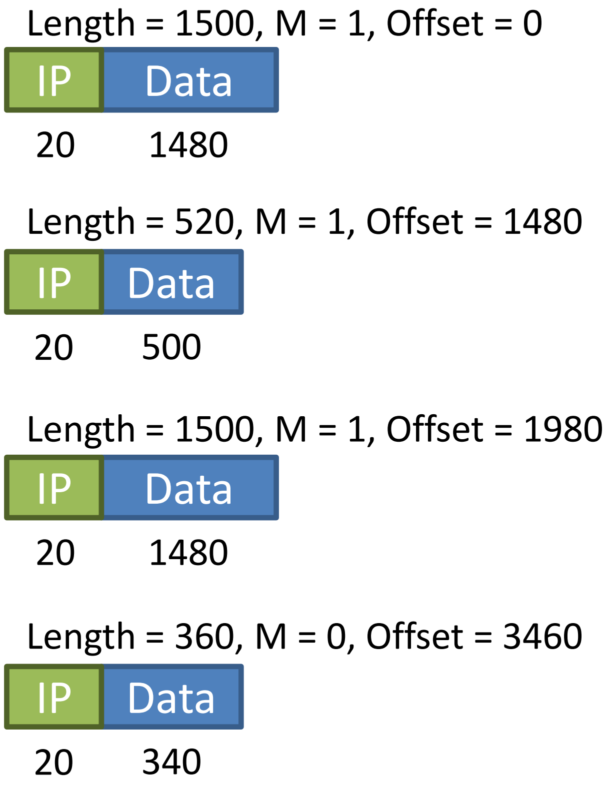
IP	Data
20	1480

1480
+ 500
= 1980

Length = 360, M = 0
Offset = 3460

IP	Data
20	340

IP Fragment Reassembly



- Performed at destination
- M = 0 fragment gives us total data size
  - 360 – 20 + 3460 = 3800
- Challenges:
  - Out-of-order fragments
  - Duplicate fragments
  - Missing fragments
- Basically, memory management nightmare

Fragmentation Concepts

- Highlights many key Internet characteristics
  - Decentralized and heterogeneous
    - Each network may choose its own MTU
  - Connectionless datagram protocol
    - Each fragment contains full routing information
    - Fragments can travel independently, on different paths
  - Best effort network
    - Routers/receiver may silently drop fragments
    - No requirement to alert the sender
  - Most work is done at the endpoints
    - i.e. reassembly

Outline

- Addressing
  - Class-based
  - CIDR
  - IP forwarding
  - NAT
- IPv4 Protocol Details
  - Packed Header
  - Fragmentation
- IPv6

The IPv4 Address Space Crisis

- Problem: the IPv4 address space is too small
  - 2^{32} = 4,294,967,296 possible addresses
  - Less than one IP per person
- Parts of the world have already run out of addresses
  - IANA assigned the last /8 block of addresses in 2011

Region	Regional Internet Registry (RIR)	Exhaustion Date
Asia/Pacific	APNIC	April 19, 2011
Europe/Middle East	RIPE	September 14, 2012
North America	ARIN	13 Jan 2015 (Projected)
South America	LACNIC	13 Jan 2015 (Projected)
Africa	AFRINIC	17 Jan 2022(Projected)

IPv6

- IPv6, first introduced in 1998(!)
  - 128-bit addresses
  - 4.8 * 10^{28} addresses per person
- Address format
  - 8 groups of 16-bit values, separated by ‘:’
  - Leading zeroes in each group may be omitted
  - Groups of zeroes can be omitted using ‘::’

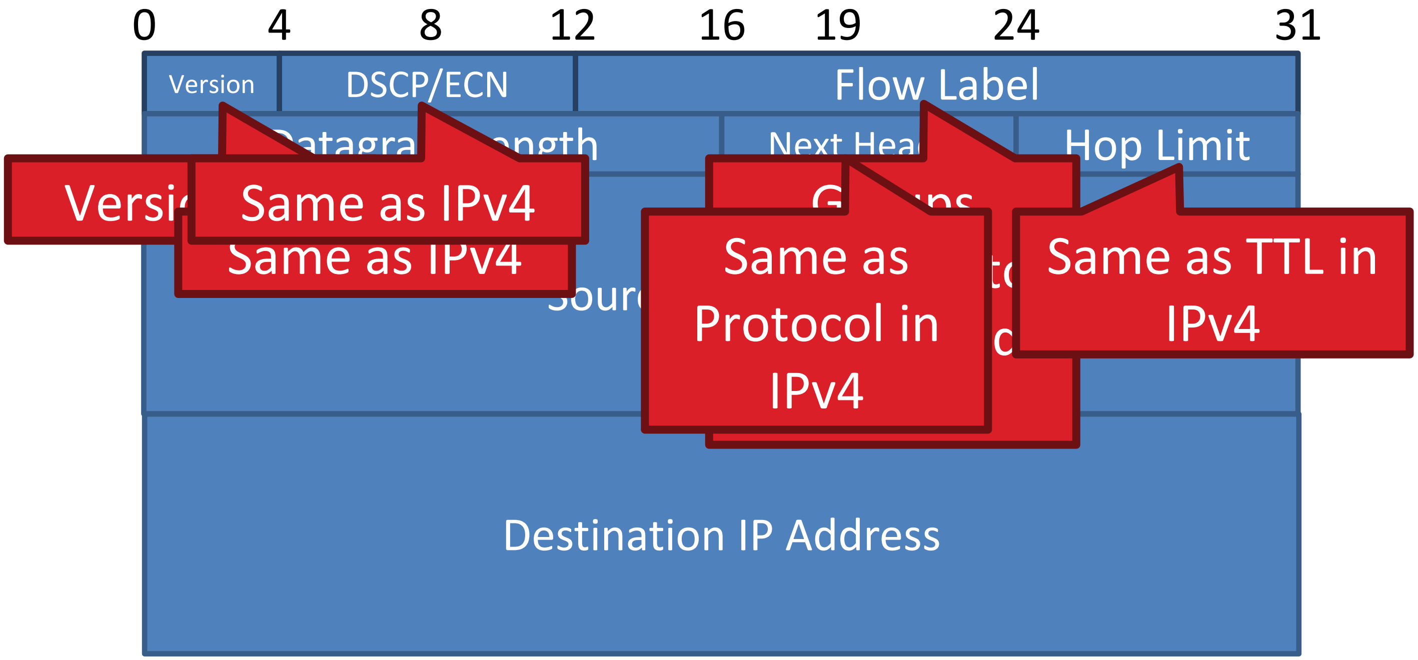
2001:0db8:0000:0000:0000:ff00:0042:8329

2001:0db8:0:0:0:ff00:42:8329

2001:0db8::ff00:42:8329

IPv6 Header

- Double the size of IPv4 (320 bits vs. 160 bits)



Differences from IPv4 Header

- Several header fields are missing in IPv6
  - Header length – rolled into Next Header field
  - Checksum – was useless, so why keep it
  - Identifier, Flags, Offset
    - IPv6 routers do not support fragmentation
    - Hosts are expected to use path MTU discovery
- Reflects changing Internet priorities
  - Today's networks are more homogeneous
  - Instead, routing cost and complexity dominate

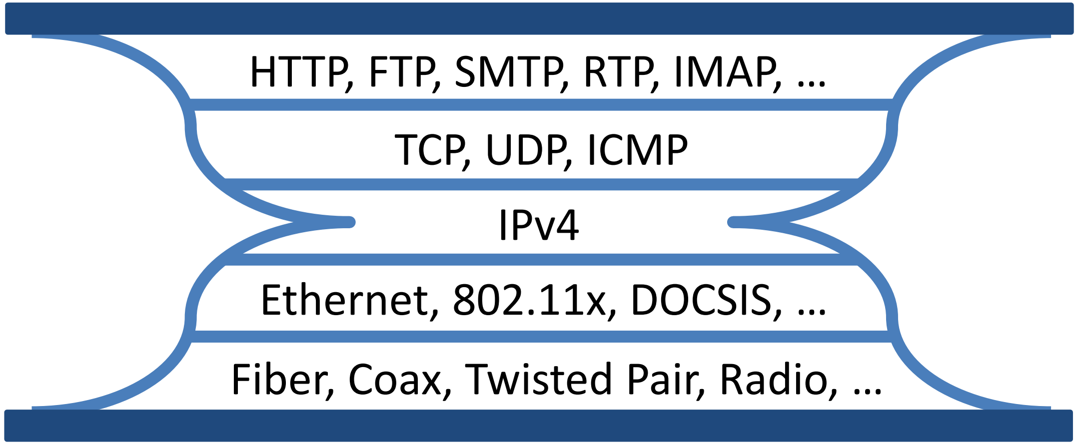
Performance Improvements

- No checksums to verify
- No need for routers to handle fragmentation
- Simplified routing table design
  - Address space is huge
  - No need for CIDR (but need for aggregation)
  - Standard subnet size is 2^{64} addresses
- Simplified auto-configuration
  - Neighbor Discovery Protocol
  - Used by hosts to determine network ID
  - Host ID can be random!

Additional IPv6 Features

- Source Routing
  - Host specifies the route to wants packet to take
- Mobile IP
  - Hosts can take their IP with them to other networks
  - Use source routing to direct packets
- Privacy Extensions
  - Randomly generate host identifiers
  - Make it difficult to associate one IP to a host
- Jumbograms
  - Support for 4Gb datagrams

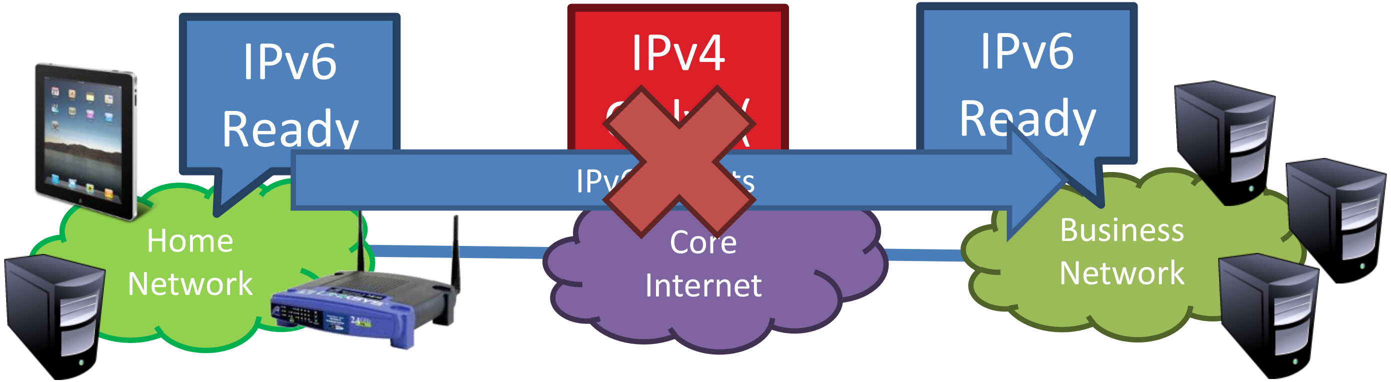
Deployment Challenges



- Switching to IPv6 is a whole-Internet upgrade
  - All routers, all hosts
  - ICMPv6, DHCPv6, DNSv6
- 2013: 0.94% of Google traffic was IPv6, 2.5% today

Transitioning to IPv6

- How do we ease the transition from IPv4 to IPv6?
  - Today, most network edges are IPv6 ready
    - Windows/OSX/iOS/Android all support IPv6
    - Your wireless access point probably supports IPv6
  - The Internet core is hard to upgrade
  - … but a IPv4 core cannot route IPv6 traffic

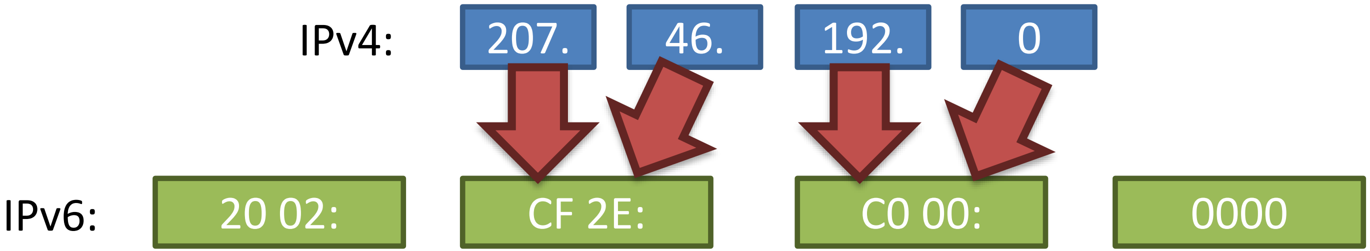


Transition Technologies

- How do you route IPv6 packets over an IPv4 Internet?
- Transition Technologies
  - Use tunnels to encapsulate and route IPv6 packets over the IPv4 Internet
  - Several different implementations
    - 6to4
    - IPv6 Rapid Deployment (6rd)
    - Teredo
    - … etc.

6to4 Basics

- Problem: you've been assigned an IPv4 address, but you want an IPv6 address
  - Your ISP can't or won't give you an IPv6 address
  - You can't just arbitrarily choose an IPv6 address
- Solution: construct a 6to4 address
  - 6to4 addresses always start with 2002::
  - Embed the 32-bit IPv4 inside the 128-bit IPv6 address



Problems with 6to4

- Uniformity
  - Not all ISPs have deployed 6to4 relays
- Quality of service
  - Third-party 6to4 relays are available
  - …but, they may be overloaded or unreliable
- Reachability
  - 6to4 doesn’t work if you are behind a NAT
- Possible solutions
  - IPv6 Rapid Deployment (6rd)
    - Each ISP sets up relays for its customers
    - Does not leverage the 2002:: address space
  - Teredo
    - Tunnels IPv6 packets through UDP/IPv4 tunnels
    - Can tunnel through NATs, but requires special relays