COMP/ELEC 429/556
Introduction to Computer Networks

Let's Build a Scalable Global Network - IP

Some slides used with permissions from Edward W. Knightly, T. S. Eugene Ng, Ion Stoica, Hui Zhang

Some Perspectives

- We started with the simplest kind of networks – i.e. broadcast networks
  - Explored access control problems and solutions
  - Explored their performance and range limitations
  - Therefore they are "Local Area Network" (LAN)

- We tried to relieve these limitations by bridging
  - Explored how Ethernet bridges can transparently link up Ethernet segments, thereby enlarging the network
  - Don't forget bridged Ethernet still needs to maintain the broadcast network abstraction
  - Explored address learning as a way to reduce cross-bridge traffic
  - Explored spanning tree protocol as a way to make packet flooding safe

We Have Reached a Road Block

- Cannot build a global network using Ethernet bridges
  - Forwarding table explosion
    - imagine telephone numbers are randomly assigned and don't have country/area codes
  - Inefficiency of spanning tree protocol and restricted forwarding
    - root bridge is the bottleneck
  - Inefficiency/danger of flooding
- Additionally, a global network should allow heterogeneous technologies (e.g. Aloha Net, Ethernet, ARPANET, 4G LTE, Wi-Fi, Bluetooth, etc)
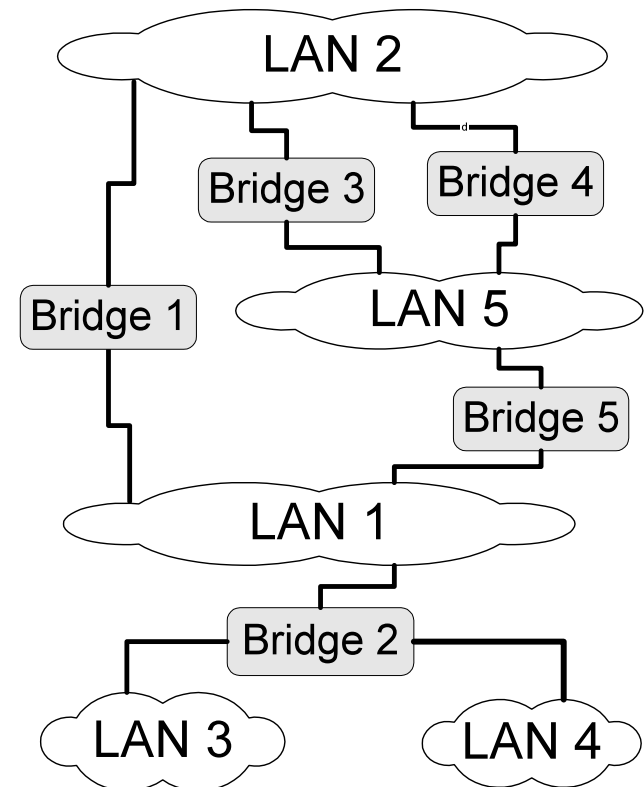
# Design Philosophy of the DARPA Internet Protocols by David D. Clark (1988)

Fundamental Goal: An effective technique for multiplexed utilization of existing interconnected networks.

Secondary Goals:

1. Internet communication must continue despite loss of networks or gateways
2. The Internet must support multiple types of communications service
3. The Internet architecture must accommodate a variety of networks
4. The Internet architecture must permit distributed management of its resources
5. The Internet architecture must be cost effective
6. The Internet architecture must permit host attachment with a low level of effort
7. The resources used in the Internet architecture must be accountable

New Word: Internetwork

- Multiple incompatible LANs can be physically connected by specialized computers called *routers.*
  - *Routers provide inter-operability*
- The connected networks are called an *internetwork*.
 - The "*Internet*" is one (very big & successful) example of an internetwork
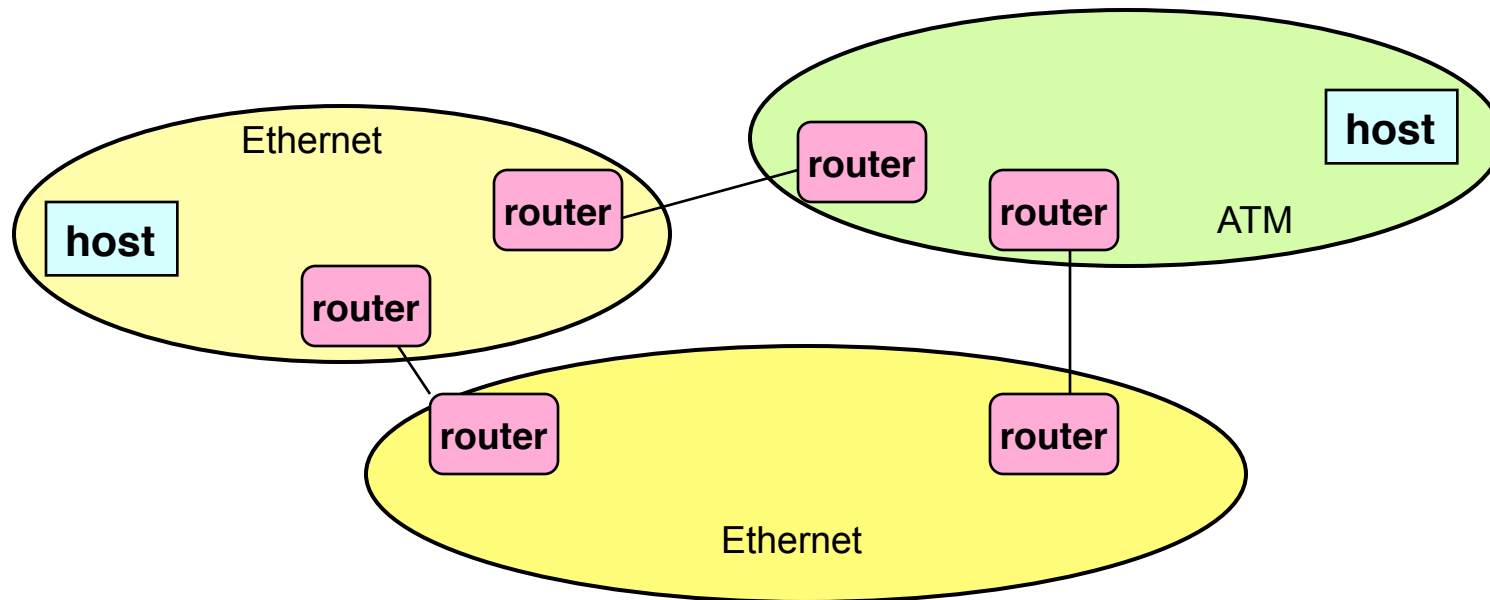
Structure of Internet
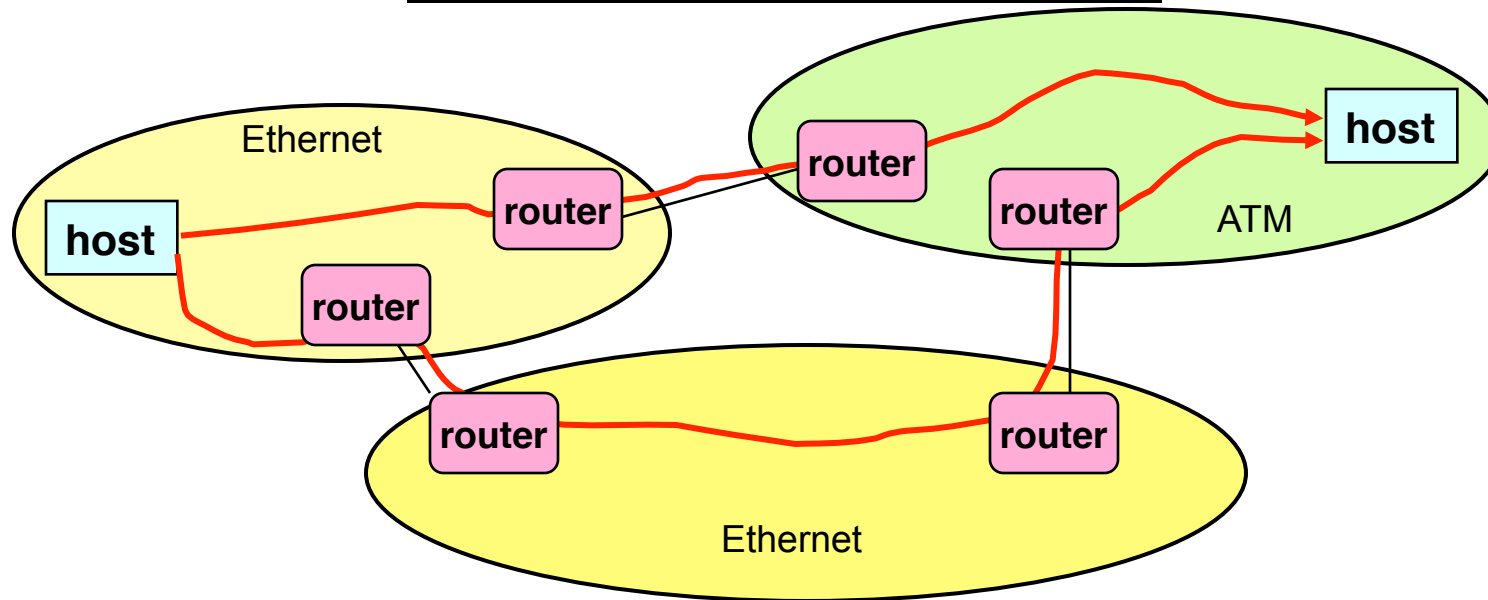
– Ad hoc interconnection of networks
  • No topological restriction
  • Vastly different router & link capacities
– Send packets from source to destination by hopping through networks
  • Router connects one network to another
  • Different packets may take different routes

Internet Protocol (IP) Provides Interoperability
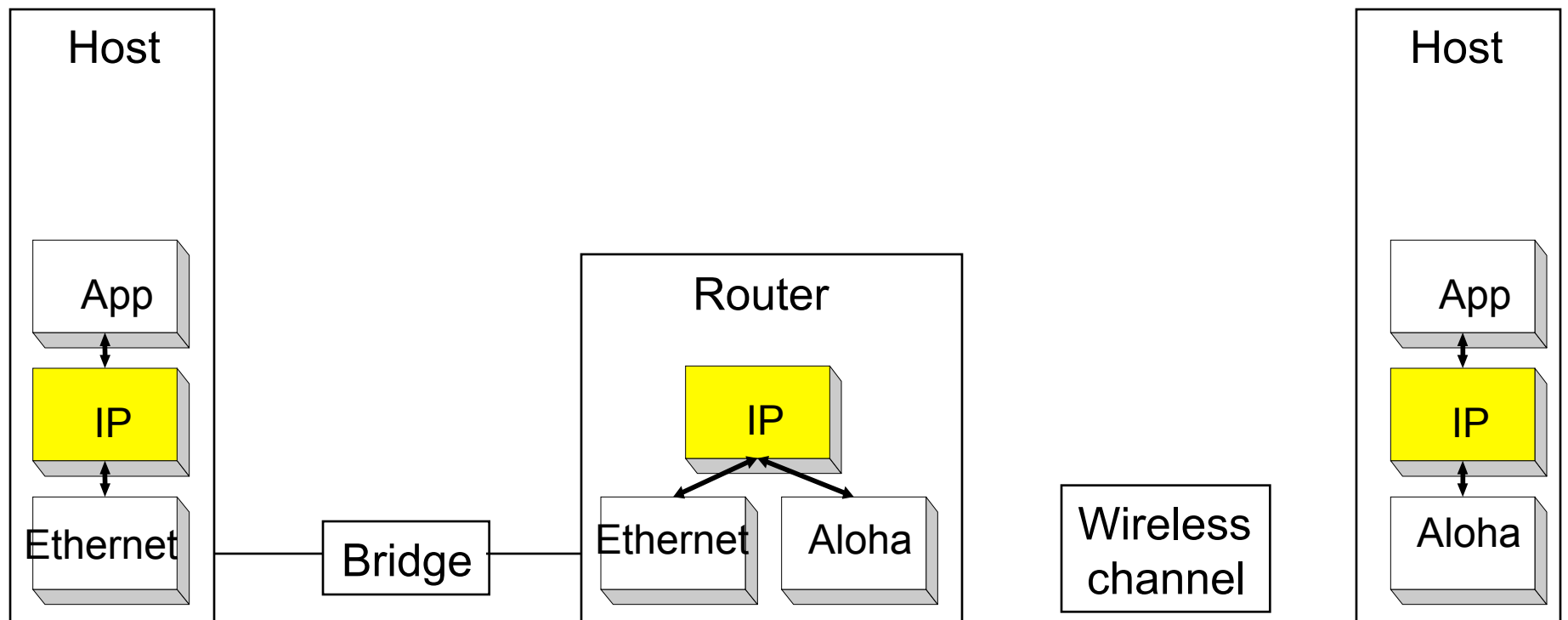
Issues in Designing an Internetwork

- How do I designate a distant host?
  - Addressing / naming

- How do I send information to a distant host?
  - Underlying service model
    - What gets sent?
    - How fast will it go?
    - What happens if it doesn't get there?
  - Routing

- Challenges
  - Scalability
    - Ensure ability to grow to worldwide scale
  - Robustness
    - Ensure loss of individual router/network has little global impact
  - Heterogeneity
    - Packet may pass through variety of different networks

IP Service Model

- ## Datagram
  - Each packet self-contained
    - All information needed to get to destination
    - No advance setup or connection maintenance
  - No performance or reliability guarantee

IPv4 Packet Format

0	4	8	12	16	19	24	28	31

ver-sion	HLen	TOS	Length			
Ident			Flags	Offset		
TTL		Protocol	Checksum			
Source Address						
Destination Address						
Options (if any)						
Data						

Header

IP Header Fields: Word 1

| 0 | 4 | 8 | 12 | 16 | 19 | 24 | 28 | 31 |

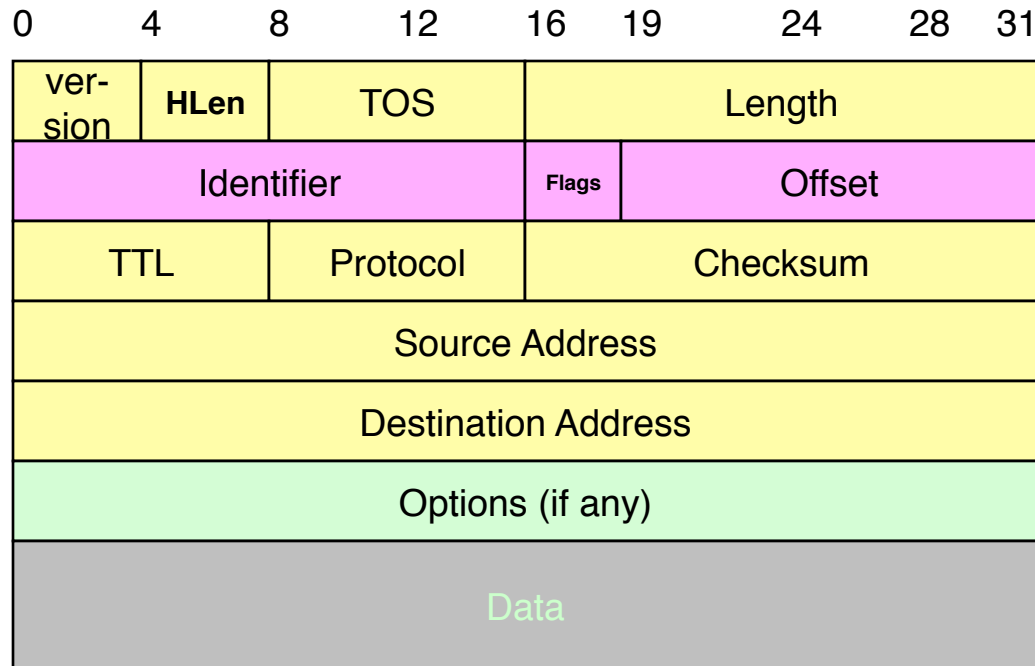| version	HLen	TOS	Length
Identifier		Flags	Offset
TTL	Protocol		Checksum
Source Address			
Destination Address			
Options (if any)			
Data			
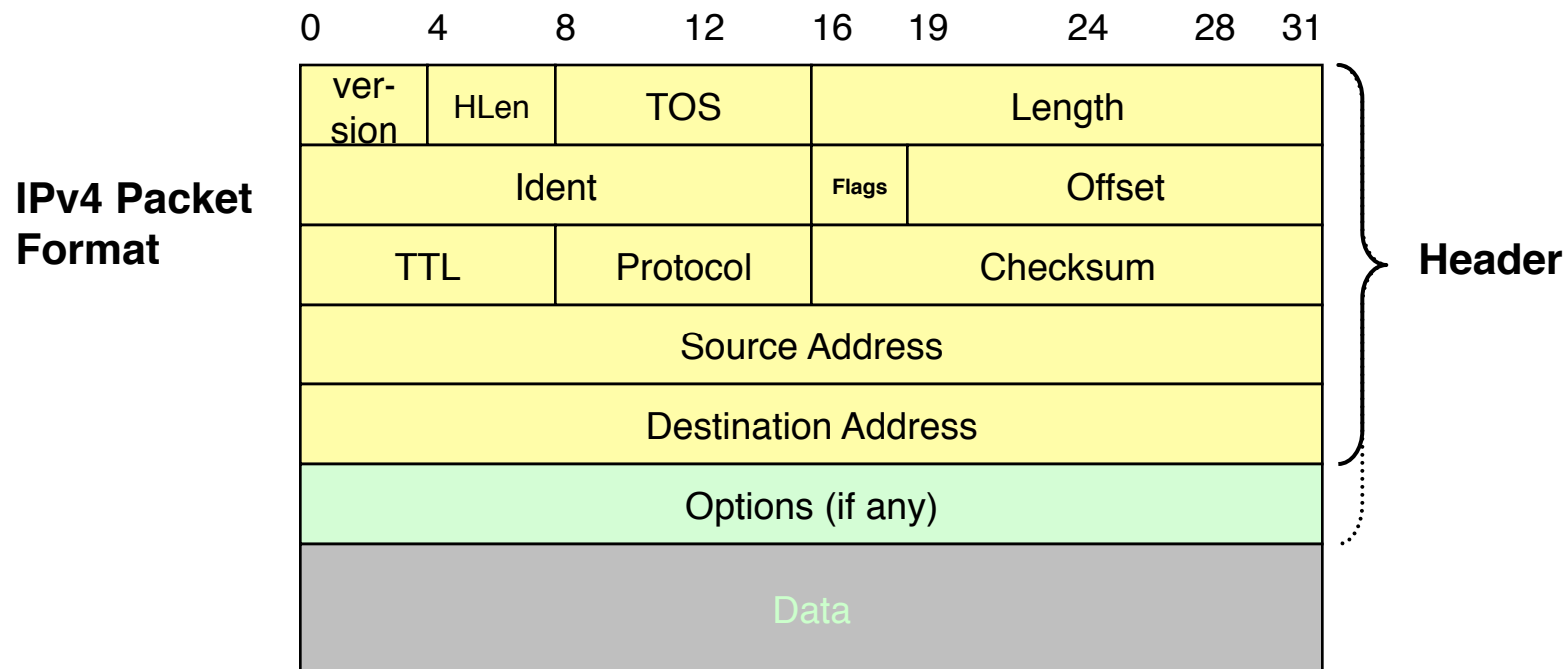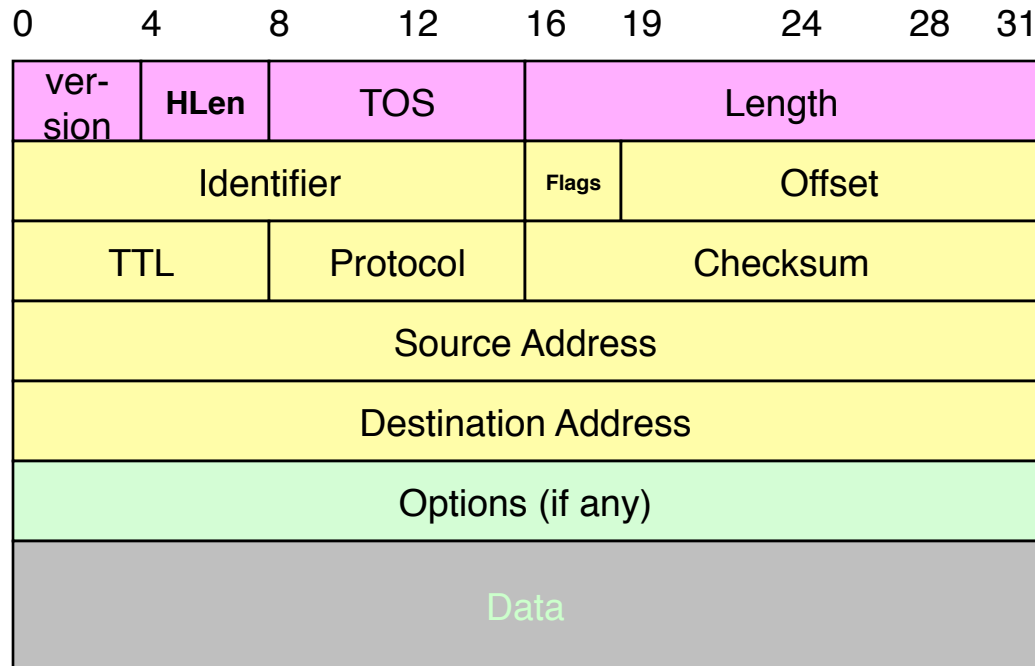
- **Version: IP Version**
  - 4 for IPv4
- **HLen: Header Length**
  - 32-bit words (typically 5)
- **TOS: Type of Service**
  - Priority information
- **Length: Packet Length**
  - Bytes (including header)

- Header format can change with versions
  - First byte identifies version
- Length field limits packets to 65,535 bytes

IP Header Fields: Word 2

0	4	8	12	16	19	24	28	31

ver-sion	HLen	TOS	Length					
Identifier				Flags	Offset			
TTL		Protocol		Checksum				
Source Address								
Destination Address								
Options (if any)								
Data								

- •Identifier
  - – Unique identifier for original datagram
    - • Typically, source increments counter every time sends packet
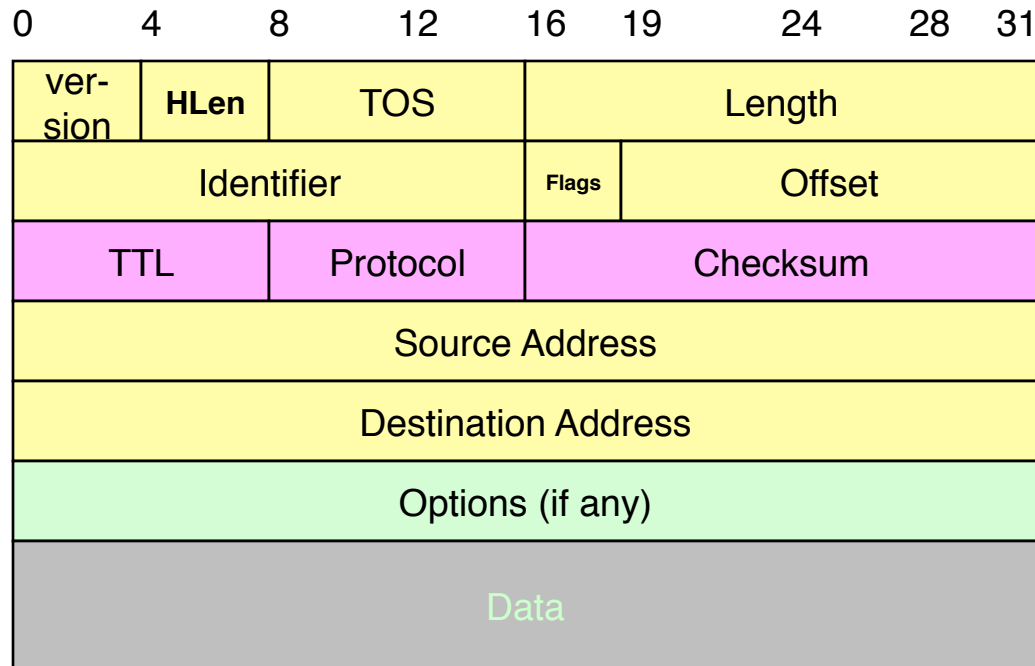- •Flags (3 bits)
  - – M flag: This is not the last fragment
- •Offset
  - – Byte position of first byte in fragment ÷ 8
  - – Byte position must be multiple of 8

- – Each fragment carries copy of IP header
  - • All information required for delivery to destination
- – All fragments comprising original datagram have same identifier
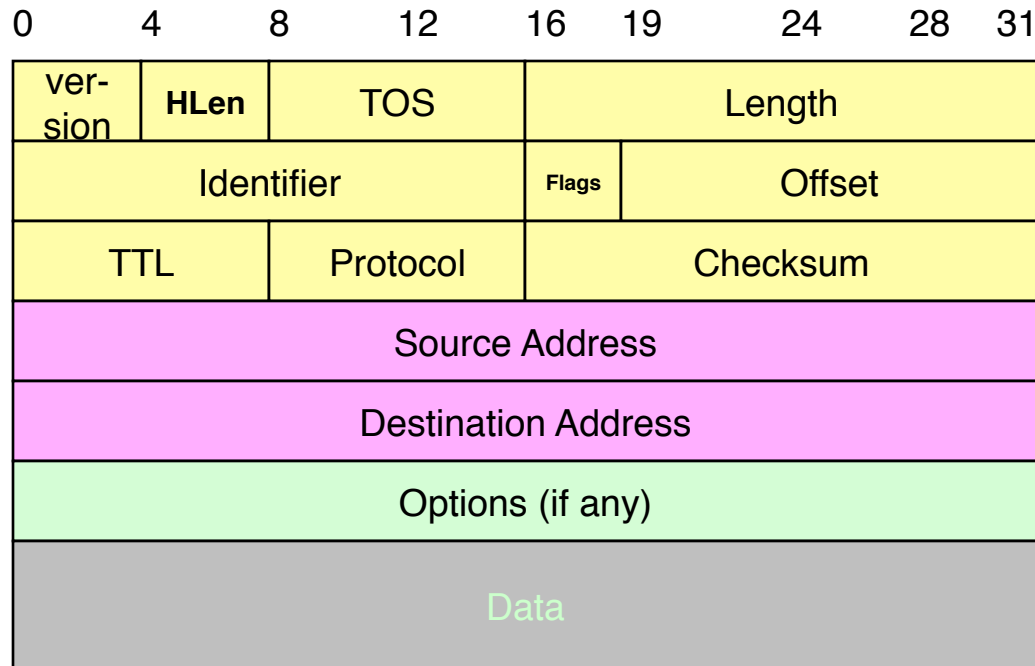- – Offsets indicate positions within datagram

IP Header Fields: Word 3

```
  0     4     8     12    16  19      24    28  31
```

ver-sion	HLen	TOS	Length	
Identifier			Flags	Offset
TTL		Protocol	Checksum	
Source Address				
Destination Address				
Options (if any)				
Data				

- •TTL: time to live
  - – Decrement by one at each intermediate router
  - – Prevent looping forever
- •Protocol
  - – Protocol of the "Data"
  - – E.g. TCP (6), UDP (17)
- •Checksum
  - – Of IP header

- – Protocol field (encodes TCP, UDP, etc.) used for demultiplexing
- – Checksum re-computed at each router
  - • Why?
- – TTL field is used in the traceroute tool

IP Header Fields: Words 4&5

ver-sion	HLen	TOS	Length
Identifier		Flags	Offset
TTL	Protocol		Checksum
Source Address			
Destination Address			
Options (if any)			
Data			

Bit positions: 0 4 8 12 16 19 24 28 31

- **Source Address**
  - 32-bit IP address of sender
- **Destination Address**
  - 32-bit IP address of destination

  - Like the addresses on an envelope
  - In principle, globally unique identification of sender & receiver

IP Addressing

- ## IPv4: 32-bit addresses
  - Typically written in dotted decimal format
    - E.g., 128.42.198.135
    - Each number is the decimal representation of a byte
  - Big-Endian Order

0	8	16	24	31	
128	42	198	135		**Decimal**
80	2a	c6	87		**Hexadecimal**
1000 0000	0010 1010	1100 0110	1000 0111		**Binary**

Possible Addressing Schemes

- Flat
  - e.g., In an Ethernet, every host identified by its 48-bit MAC address
  - Router would need entry for every host in the world
    - Too big
    - Too hard to maintain as hosts come & go
- Hierarchical
  - Address broken into segments of increasing specificity
    - 713 (Houston) – 348 (Rice area) – 2000 (Particular phone)
  - Route to general region and then work toward specific destination
  - As people and organizations shift, only update affected routing tables

IP Addressing and Forwarding

- Routing Table Requirement
  - For every possible destination IP address, give next hop
  - Nearly 2^{32} (4.3×10^9) possibilities!

- Hierarchical Addressing Scheme

network	host

  - Address split into network ID and host ID
  - All packets to given network follow same route
    - Until they reach destination network

How to split an address into "network" and "host" parts?

- Evolved over time

- Began with designating different address classes
  - Class A, B, C
  - Different classes imply different network/host split
  - Unfortunately proved too inflexible

- Today, an explicit prefix length is used to designate how network/host split
  - e.g. 128.42.0.0/16 (netmask 255.255.0.0)

Classless Interdomain Routing

- CIDR, pronounced "cider"

- Arbitrary Split Between Network & Host IDs
  - Specify either by mask (below) or prefix length (16)

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

network	host

  - E.g., Rice's original set of IP addresses can be specified as
    - 128.42.0.0 with netmask 255.255.0.0
    - 128.42.0.0/16

Routing Table Entry Examples

Address	Prefix Length	Third Byte	Next Hop
207.46.0.0	19	0-31	R1
207.46.32.0	19	32-63	R2
207.46.64.0	19	64-95	R3
207.46.128.0	18	128-191	R4
207.46.192.0	18	192-255	R5

This column does not exist in the real routing table, it's here just to make it more clear.

Note hole in table: Nothing covers third byte 96 – 127

What If We Have This?

Address	Prefix Length	Third Byte	Next Hop
207.46.0.0	19	0-31	R1
207.46.32.0	19	32-63	R1
207.46.64.0	19	64-95	R1
207.46.128.0	18	128-191	R4
207.46.192.0	18	192-255	R5

Visualizing 207.46.0.0/16

207.46.0.0

207.46.0.0/19   R1

207.46.32.0/19   R1

207.46.64.0/19   R1

207.46.128.0/18   R4
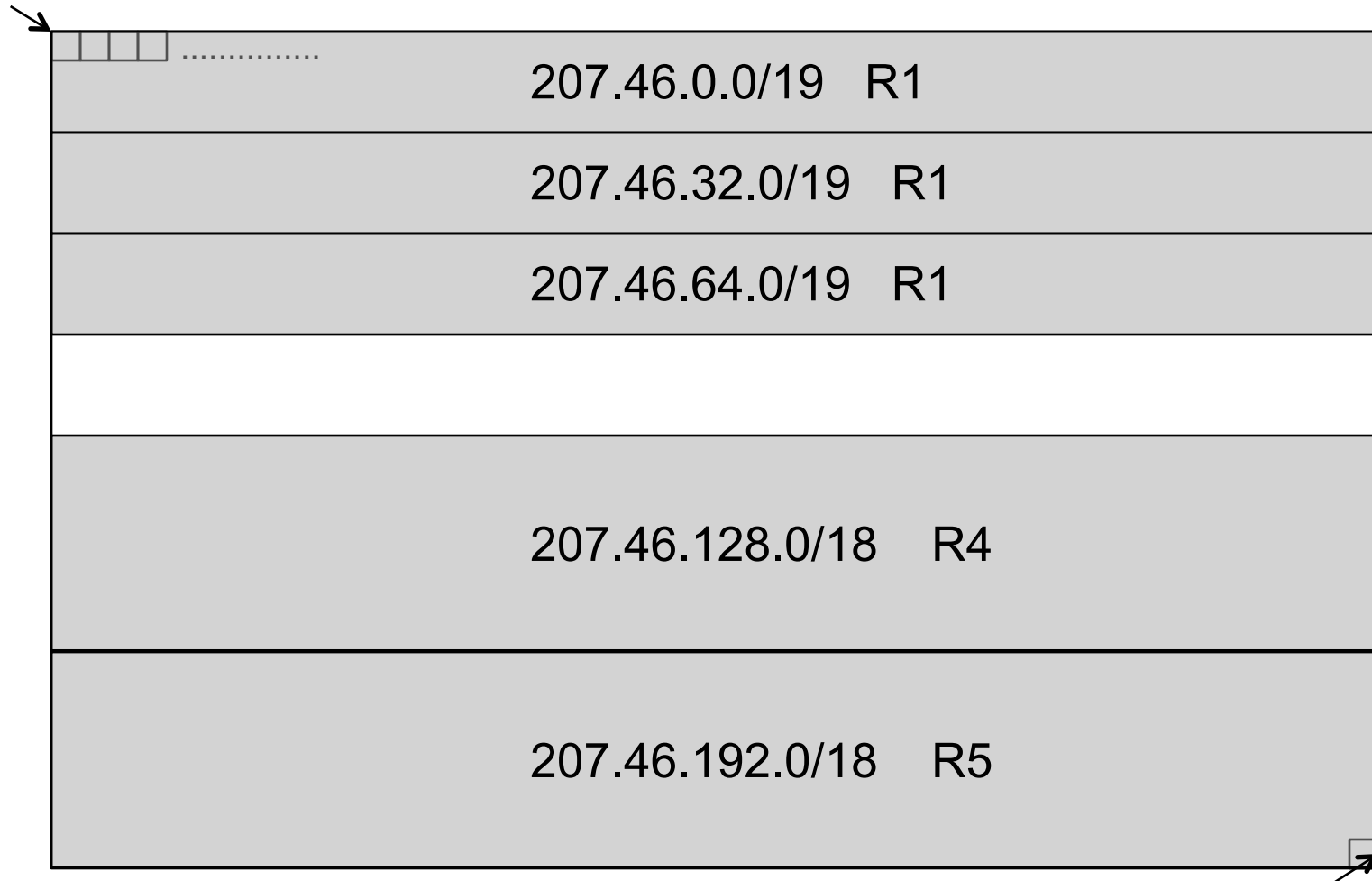
207.46.192.0/18   R5

207.46.255.255

Aggregate Entries to Compress Table

Address	Prefix Length	Third Byte	Next Hop
207.46.0.0	19	0-31	R1
207.46.32.0	19	32-63	R1
207.46.64.0	19	64-95	R1
207.46.128.0	18	128-191	R4
207.46.192.0	18	192-255	R5

Key requirement: Same routing behavior for all active addresses after aggregation

Address	Prefix Length	Third Byte	Next Hop
207.46.0.0	**17**	**0-127**	R1
207.46.128.0	18	128-191	R4
207.46.192.0	18	192-255	R5

What about this?

Address	Prefix Length	Third Byte	Next Hop
207.46.0.0	**17**	**0-127**	R1
207.46.128.0	18	128-191	R1
207.46.192.0	18	192-255	R5

Address	Prefix Length	Third Byte	Next Hop
207.46.0.0	**16**	**0-255**	R1
207.46.192.0	18	192-255	R5

Same routing behavior for all active addresses?

Yes if router performs longest prefix matching

Longest Prefix Matching

Address Pattern	Subnet Mask	Next Hop
128.42.222.0	255.255.255.0	R1
128.42.128.0	255.255.128.0	R2
18.0.0.0	255.0.0.0	R3
0.0.0.0	0.0.0.0	R4
128.42.0.0	255.255.0.0	R5

- Address 128.42.222.198 matches 4 entries

- Longest Prefix Match
  - Select entry with longest sequence of 1's in mask
  - Most specific case

Size of Complete Routing Table
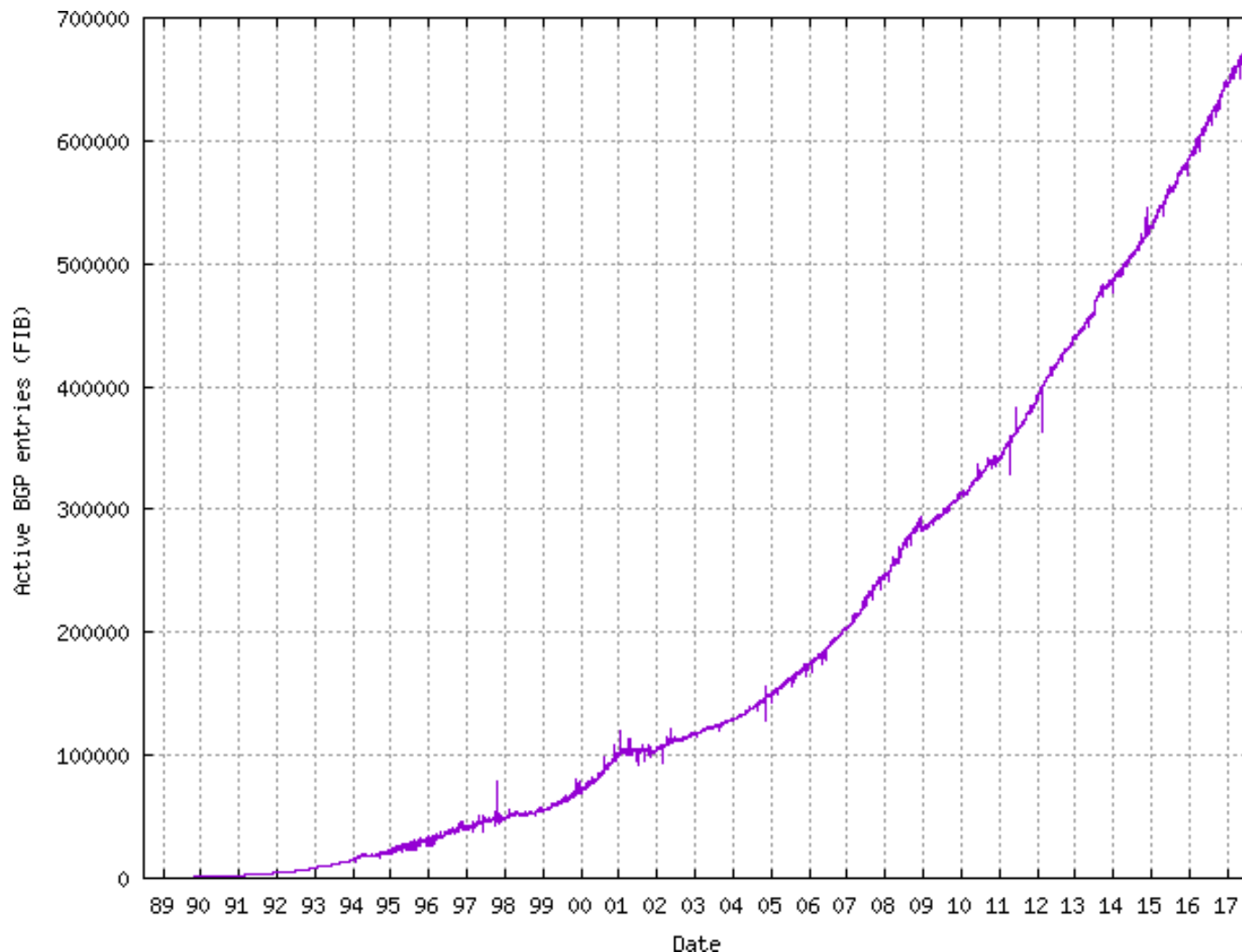
– Source: www.cidr-report.org

Remember This?

BGPMON

What caused today's Internet hiccup

Posted by Andree Toonk - August 13, 2014 - *BGP instability* - *No Comments*
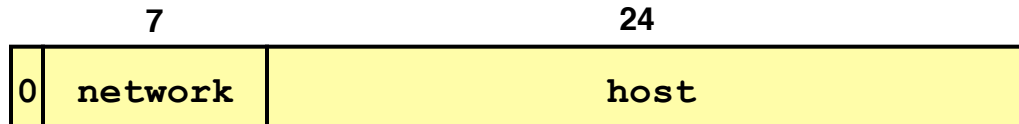
Like others, you may have noticed some instability and genera
today. In this post we'll take a closer look at what happened, i

At around 8am UTC Internet users on different mailing lists, fo

"limitation in older Cisco routers. These routers have a default limit of 512K routing entries"

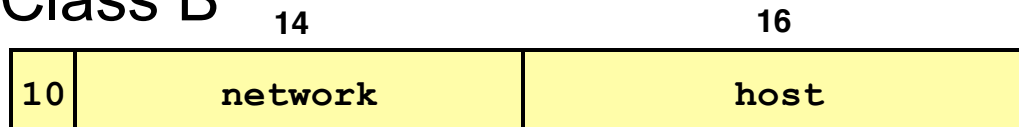# Original IP Address Classes (Obsolete today)

- ## Class A

7	24
0 network	host

  First octet: 1–126

  – e.g. MIT: 18.7.22.69

- ## Class B

14	16
10 network	host

  First octet: 128–191

  – e.g. Rice: 128.42.129.23

- ## Class C

21	8
110 network	host
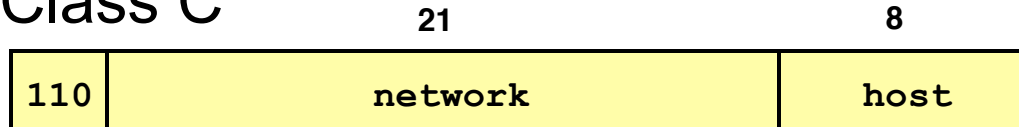
  First octet: 192–223

  – e.g. adsl-216-63-78-18.dsl.hstntx.swbell.net: 216.63.78.18

- ## Classes D, E, F

  – Not commonly used

  – D for multicast service (more later in the course)

Implications

Class	Count	Hosts
A	$2^7-2 = 126$ (0 & 127 reserved)	$2^{24}-2 = 16,777,214$ (all 0s, all 1s reserved)
B	$2^{14} = 16,398$	$2^{16}-2 = 65,534$ (all 0s, all 1s reserved)
C	$2^{21} = 2,097,512$	$2^8-2 = 254$ (all 0s, all 1s reserved)
Total	2,114,036	

- Partitioning too Coarse
  - No local organization needs 16.7 million hosts
    - Large organization likely to be geographically distributed
  - Many organizations must make do with multiple class C's
- Too many different Network IDs
  - Routing tables may have 2.1 million entries

Important Concepts

- Hierarchical addressing critical for scalable system
  - Don't require everyone to know everyone else
  - Reduces amount of updating when something changes
- Non-uniform hierarchy useful for heterogeneous networks
  - Class-based addressing too coarse
  - CIDR helps
- Implementation Challenge
  - Longest prefix matching more difficult to implement and to make fast than exact matching

Remainder of this set of slides are for your reference only. They will not be needed for assignments.
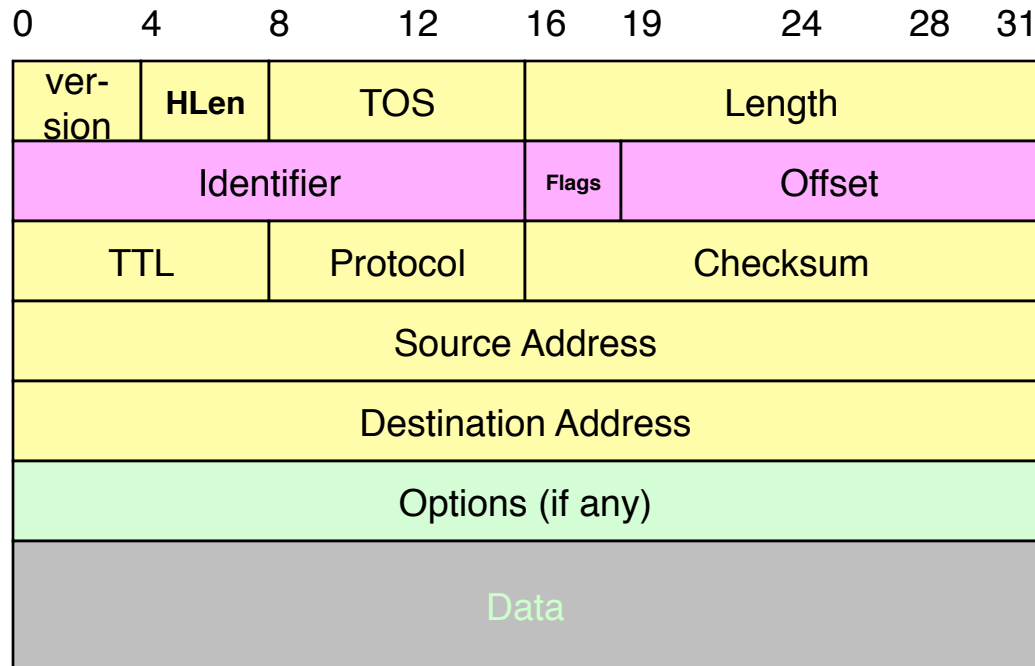
IP Fragmentation

MTU = 2000

router

host

host

router

MTU = 1500

MTU = 4000
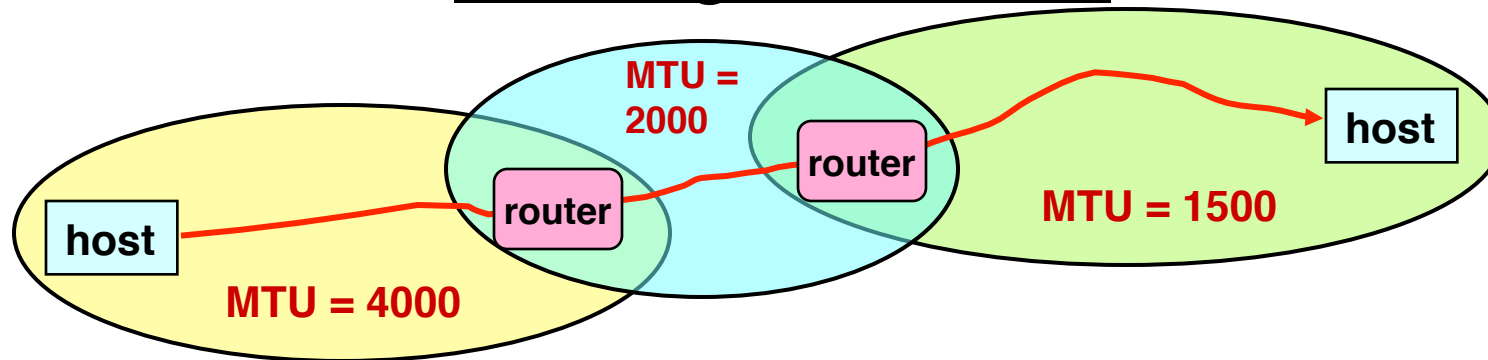
- Every Network has Own Maximum Transmission Unit (MTU)
  - Largest IP datagram it can carry within its own packet frame
    - E.g., Ethernet is 1500 bytes
  - Don't know MTUs of all intermediate networks in advance
- IP Solution
  - When hit network with small MTU, fragment packets
    - Might get further fragmentation as proceed farther
  - Reassemble at the destination
    - If any fragment disappears, delete entire packet

IP Header Fields: Word 2

ver-sion	HLen	TOS	Length		
Identifier			Flags	Offset	
TTL		Protocol	Checksum		
Source Address					
Destination Address					
Options (if any)					
Data					

Bit positions: 0 4 8 12 16 19 24 28 31

- **Identifier**
  - Unique identifier for original datagram
    - Typically, source increments counter every time sends packet
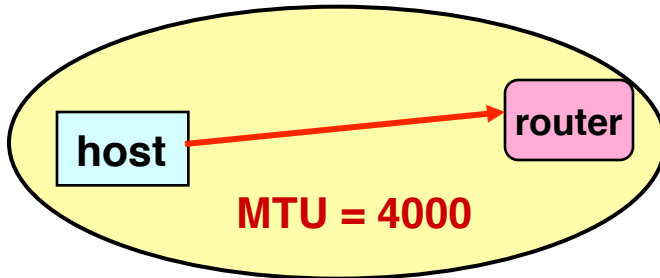- **Flags (3 bits)**
  - M flag: This is not the last fragment
- **Offset**
  - Byte position of first byte in fragment ÷ 8
  - Byte position must be multiple of 8

- Each fragment carries copy of IP header
  - All information required for delivery to destination
- All fragments comprising original datagram have same identifier
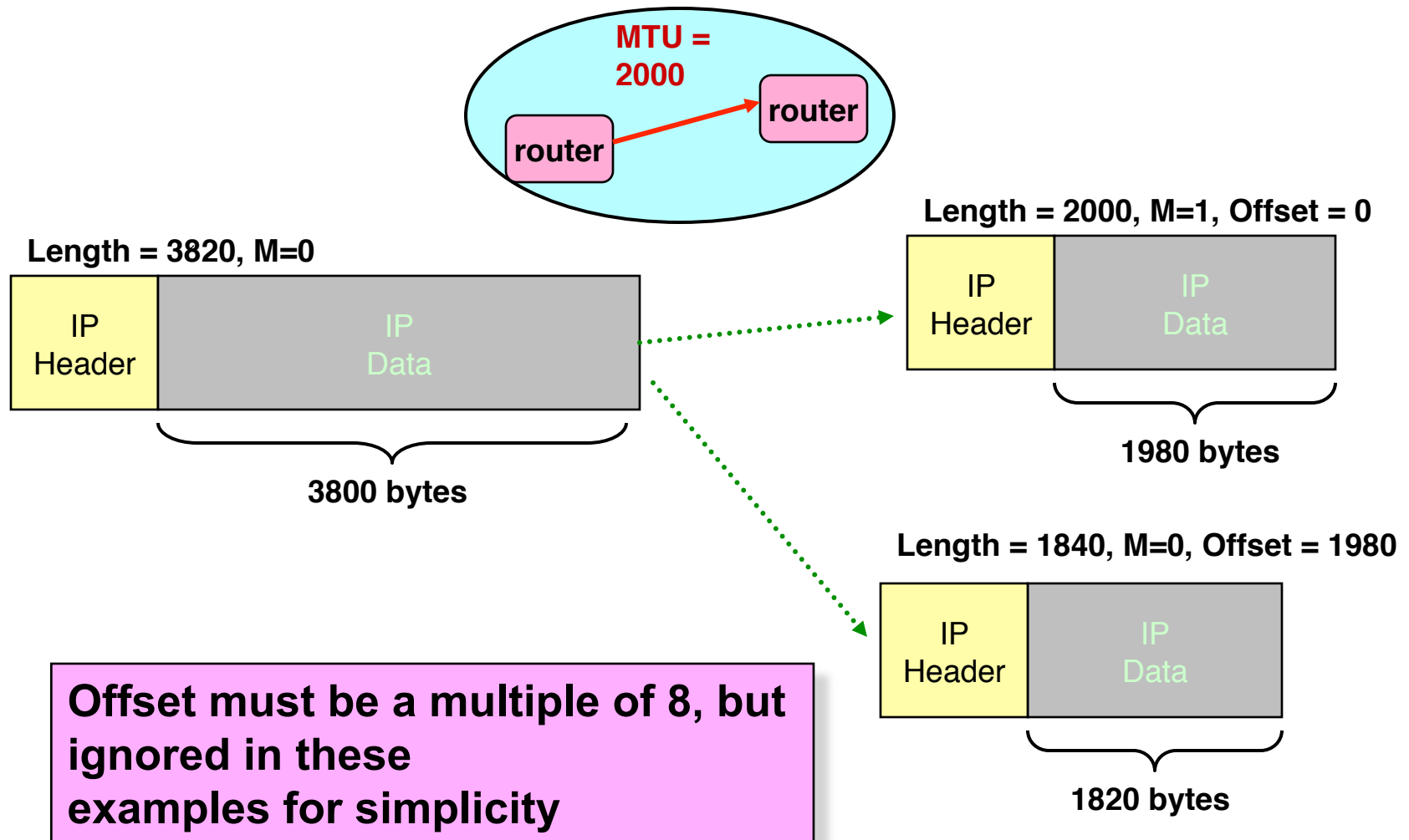- Offsets indicate positions within datagram
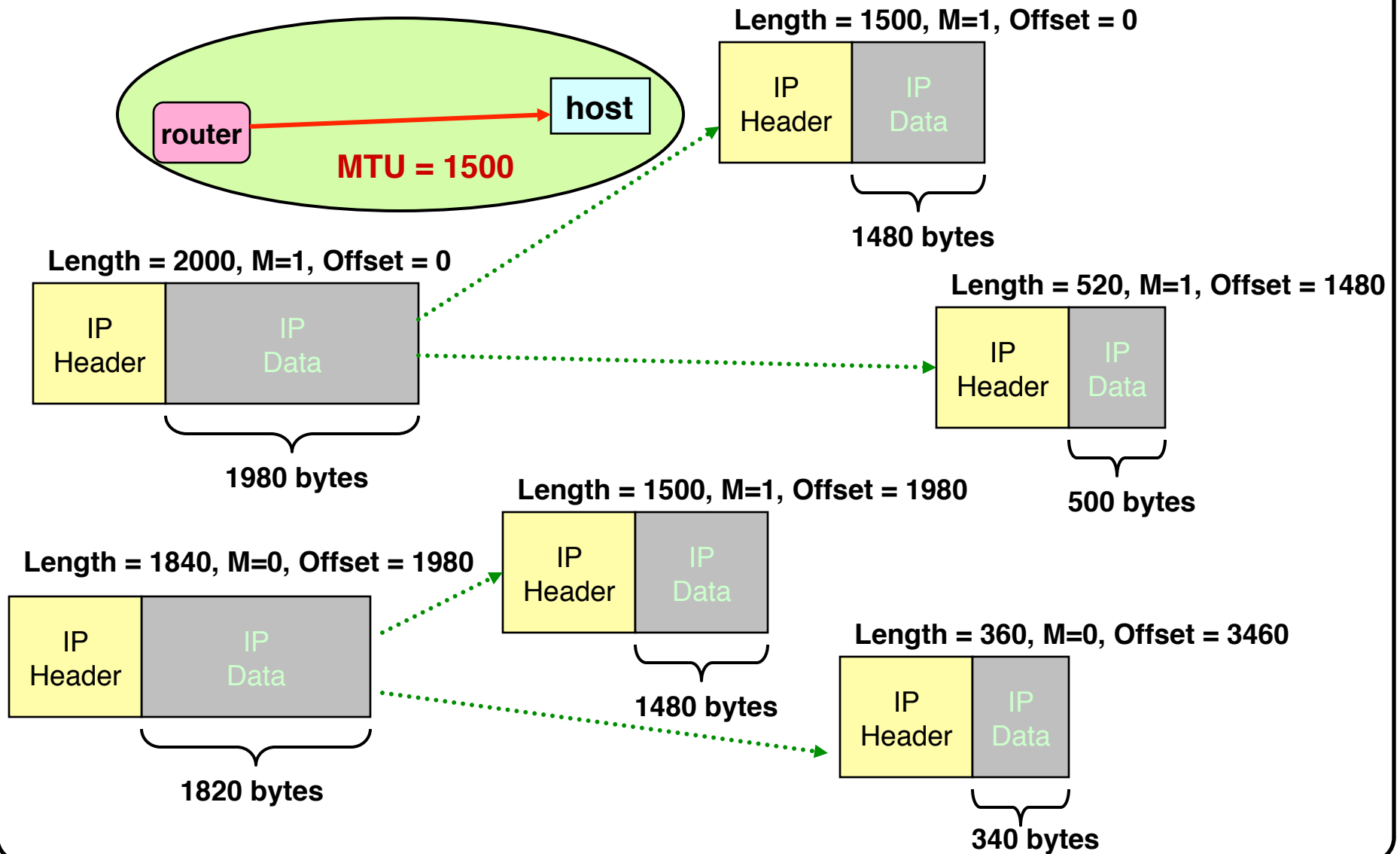
IP Fragmentation Example #1

host → router

MTU = 4000

Length = 3820, M=0

IP Header	IP Data

IP Fragmentation Example #2

MTU = 2000

router → router

Length = 3820, M=0

IP Header	IP Data

3800 bytes

Length = 2000, M=1, Offset = 0

IP Header	IP Data

1980 bytes

Length = 1840, M=0, Offset = 1980

IP Header	IP Data

1820 bytes

Offset must be a multiple of 8, but ignored in these examples for simplicity

IP Fragmentation Example #3

Length = 1500, M=1, Offset = 0

IP Header | IP Data

1480 bytes

router → host

MTU = 1500

Length = 2000, M=1, Offset = 0

IP Header | IP Data

1980 bytes

Length = 520, M=1, Offset = 1480

IP Header | IP Data

500 bytes

Length = 1840, M=0, Offset = 1980

IP Header | IP Data

1820 bytes

Length = 1500, M=1, Offset = 1980

IP Header | IP Data

1480 bytes

Length = 360, M=0, Offset = 3460

IP Header | IP Data

340 bytes

IP Reassembly

Length = 1500, M=1, Offset = 0
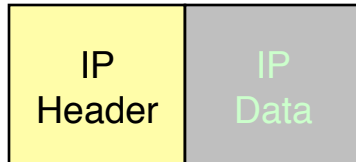
IP Header	IP Data

Length = 520, M=1, Offset = 1480

IP Header	IP Data

Length = 1500, M=1, Offset = 1980

IP Header	IP Data

Length = 360, M=0, Offset = 3460

IP Header	IP Data

IP Data	IP Data	IP Data	IP Data
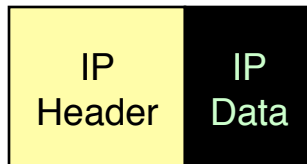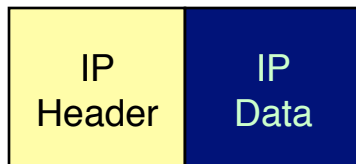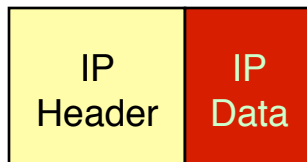
- Performed at final destination
- Fragment with M=0 determines overall length

- Challenges
  - Fragments might arrive out-of-order
    - Don't know how much memory required until receive final fragment
  - Some fragments may be duplicated
    - Keep only one copy
  - Some fragments may never arrive
    - After a while, give up entire process
  - Significant memory management issues

Frag. & Reassembly Concepts

- Demonstrates many Internet concepts
- Decentralized
  - Every network can choose MTU
- Connectionless datagram protocol
  - Each (fragment of) packet contains full routing information
  - Fragments can proceed independently and along different routes
- Fail by dropping packet
  - Destination can give up on reassembly
  - No need to signal sender that failure occurred
- Keep most work at endpoints
  - Reassembly

Frag. & Reassembly Reality

- Reassembly fairly expensive and hurts performance
  - Copying, memory allocation
  - Want to avoid
- MTU discovery protocol
  - Protocol to determine MTU along route
    - Send packets with "don't fragment" flag set
    - Keep decreasing message lengths until packets get through
    - May get a "can't fragment error" message from router which contains the correct MTU
  - Assumes every packet will follow same route
    - Routes tend to change slowly over time
- Common theme in system design
  - Fragmentation is handled as a special case by slower general processor in router
  - Assure correctness by implementing complete protocol
  - Optimize common cases to avoid full complexity