

Lecture 11: Internet Addressing

CSE 123: Computer Networks

Aaron Schulman



Today's Lecture



- IP Fragmentation
- Addressing in internetworking
 - ◆ How long should addressees be?
 - ◆ How do you address both hosts and networks?
- How does a router look up where to route a packet?



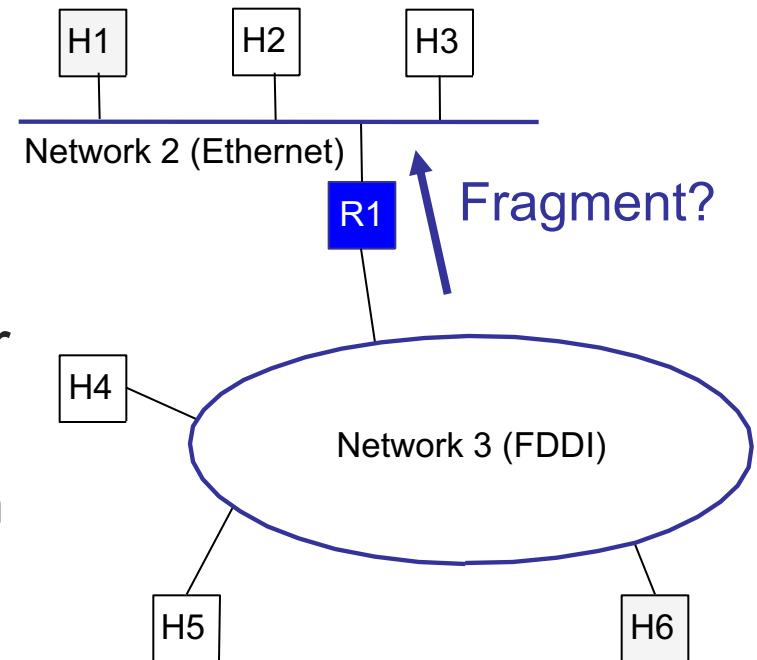
So what *does* IP do?

- Addressing to/from hosts and networks
- Fragmentation (handling different link layer protocols)
 - ◆ E.g. FDDI's maximum packet is 4500 bytes while Ethernet is 1500 bytes, how to manage this?
- *Some* error detection
- Routers only forward packets to next hop
 - ◆ They do not:
 - » Detect packet loss, packet duplication
 - » Reassemble or retransmit packets
- Today we'll talk about fragmentation and forwarding



Fragmentation

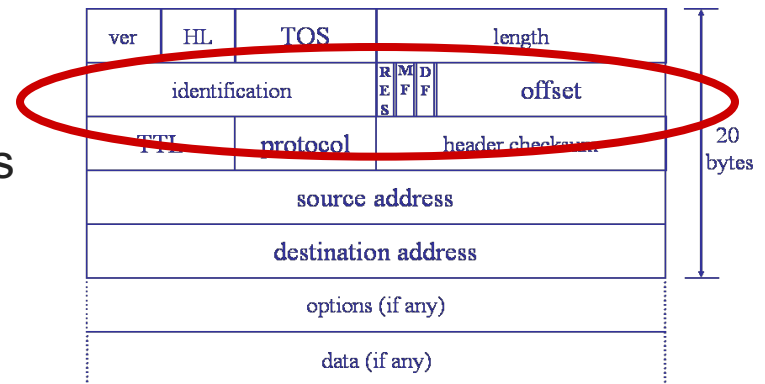
- Different networks may have different maximum frame sizes
 - ◆ Maximum Transmission Unit (MTUs)
 - ◆ Ethernet 1500, WiFi 2,346
- Router breaks up single IP packet into two or more smaller IP packets
 - ◆ Each **fragment** is labeled so it can be correctly **reassembled**
 - ◆ *End host* reassembles them into original packet



IP ID and Bitflags



- Source inserts unique value in identification field
 - ◆ Also known as the IPID
 - ◆ If packet is fragmented, the router copies this value into any fragments
- Offset field indicates position of current fragment (in bytes/8)
 - ◆ Zero for non-fragmented packet
- Bitflags provide additional information
 - ◆ More Fragments bit helps identify last fragment
 - ◆ Don't Fragment bit prohibits (further) fragmentation
 - ◆ Note recursive fragmentation easily supported—just requires care with More Fragments bit





Fragmentation Example

	length =4000	ID =x	MF =0	offset =0	
--	-----------------	----------	----------	--------------	--

One large datagram becomes
several smaller datagrams

(Offset actually
encoded as bytes/8)

	length =1500	ID =x	MF =1	offset =0	
--	-----------------	----------	----------	--------------	--

0

	length =1500	ID =x	MF =1	offset =1480	
--	-----------------	----------	----------	-----------------	--

185

	length =1040	ID =x	MF =0	offset =2960	
--	-----------------	----------	----------	-----------------	--

370



Costs of Fragmentation

- Interplay between fragmentation and retransmission
 - ◆ A **single** lost fragment may trigger retransmission
 - ◆ Any retransmission will be of entire packet (why?)
- Packet must be completely reassembled before it can be consumed on the receiving host
 - ◆ Takes up buffer space in the receiver in the mean time
 - ◆ When can it be garbage collected?
- Why not reassemble at each router?



Discovering MTU on path

- Path Maximum Transmission Unit (MTU) is the smallest MTU along path
 - ◆ Packets less than this size don't get fragmented
- Fragmentation is a burden for routers
 - ◆ We already avoid reassembling at routers
 - ◆ Avoid fragmentation too by having hosts learn path MTUs
- Hosts send packets, routers return error if too large
 - ◆ Hosts can set "don't fragment" flag
 - ◆ Hosts discover limits, can size packets at source
 - » ICMP protocol: special IP packet format for sending error msgs
 - ◆ Reassembly at destination as before



IP Forwarding Tables

- Router needs to know where to forward a packet
- Forwarding table contains:
 - ◆ List of **network names (e.g., LANs)** and **next hop** routers
 - ◆ Local networks have entries specifying which interface
 - » Link-local hosts can be delivered with Layer-2 forwarding
- Address of incoming internetwork packet needs to say:
 - ◆ What is the destination **network**?
 - ◆ What is the destination **host** in that network?

IPv4 Addresses – Network + Host



- 32-bits in an IPv4 address
 - ◆ Dotted decimal format a.b.c.d
 - ◆ Each represent 8 bits of address
- **Hierarchical:** Network part and host part
 - ◆ E.g. IP address 128.54.70.238
 - ◆ 128.54 refers to the UCSD campus network
 - ◆ 70.238 refers to the host ieng6.ucsd.edu
- How do we define which part is network (e.g., LAN) vs. host?



Class-based Addressing

- Most significant bits determines “class” of address

Class A

0	Network	Host
---	---------	------

127 nets, 16M hosts

Class B

14		16	
1	0	Network	Host

16K nets, 64K hosts

Class C

21			8	
1	1	0	Network	Host

2M nets, 254 hosts

- Special addresses
 - ◆ Class D (1110) for multicast, Class E (1111) experimental
 - ◆ 127.0.0.1: local host (a.k.a. the loopback address)
 - ◆ Host bits all set to 0: network address
 - ◆ Host bits all set to 1: broadcast address

Subnetting (network in network)

- Individual networks may be composed of several subnetworks
 - ◆ Only want to route traffic to hosts on a specific subnetwork
 - ◆ Routers need a way to know which hosts are in which subnetwork
- Networks can be arbitrarily decomposed into **subnets**
 - ◆ Each subnet is simply a prefix of the host address portion
 - ◆ Subnet prefix can be of any length, specified with **netmask**




Prefix

- ◆ Netmask specified by admins and set on hosts/routers



Subnet Addresses

- Every (sub)network has an address and a **netmask**
 - ◆ Netmask tells which bits of the network address is subnet
 - ◆ Convention suggests it be a proper prefix (must start from high order bits of address)
- Netmask written as an all-ones IP address
 - ◆ E.g., Class B netmask is 255.255.0.0
 - ◆ Sometimes expressed in terms of number of 1s, e.g., /16
- Need to size subnet appropriately for each subnetwork
 - ◆ Only have remaining bits to specify host addresses

IP Address Problem (1991)



- Address space depletion
 - ◆ In danger of running out of classes A and B
- Why?
 - ◆ Addresses only allow for fixed-size address allocations
 - » Class C too small for most organizations (only ~250 addresses)
 - » Very few class A – very careful about giving them out (who has 16M hosts anyway?)
 - Also few options for hierarchy of subnetworks
 - » Class B – greatest problem

CIDR



- Classless Inter-Domain Routing (1993)
 - ◆ Networks described by variable-length prefix and length
 - ◆ Allows arbitrary allocation between network and host address



Prefix

Mask=# significant bits representing prefix

- ◆ e.g. 10.95.1.2 contained within 10.0.0.0/8:
 - » 10.0.0.0 is network and remainder (95.1.2) is host
- Pro: Finer grained allocation; hierarchical aggregation
- Con: More expensive forwarding table lookup:
longest prefix match



Longest Matching Prefix

- Forwarding table contains many prefix/length tuples
 - ◆ They *need not* be heirarcical.
 - ◆ E.g. 200.23.16.0/20 and 200.23.18.0/23 both in table
 - ◆ What to do if a packet arrives for destination 200.23.18.1?
 - ◆ Need to find the longest prefix in the table which matches it (200.23.18.0/23)
- Not a simple table, requires multiple memory lookups
 - ◆ Lots and lots of research done on this problem
 - ◆ Lots of this work was historically done by UCSD faculty

Address length considerations



- Fixed length or variable length addresses?
- Issues:
 - ◆ Flexibility
 - ◆ Processing costs (tricky hardware design)
 - ◆ Header size
- Engineering choice: IP uses fixed length addresses

t 2013

routed used
5.3M, 31.5%
connected to a
computer and
in use

**routed
unused**
5.1M, 30.4%

present on the
Internet, but not
connected to a
computer

reserved
2.3M, 13.7%
reserved for
special or future
use

3





IPv6 vs IPv4 header

Ver.	Traffic Class	Flow Label		
Payload Length		Next Header	Hop Limit	
Source Address				
Destination Address				

Ver.	Hdr Len	Type of Service	Total Length	
Identification			Flg	Fragment Offset
Time to Live		Protocol	Header Checksum	
Source Address				
Destination Address				
Options...				

- Gray bits are unique to each header
- Changes
 - Eliminate fragmentation-related fields
 - Eliminate header checksum
 - Added flow label
 - Quadruple size of addresses
 - IPv6 header (40 bytes) vs IPv4 (20 bytes)
 - Hop limit == TTL



Extension Headers

- Effectively a linked list of headers
 - ◆ The “next header” field is the pointer
- Two different types
 - ◆ **Destination**, intended for the IP end point. E.g.,
 - » 44: Fragmentation Header (it’s baaack!)
 - » 43: Routing header (dictates how to route the packet)
 - ◆ **Hop-by-hop**, processed by each node on the path
 - » Each node takes a look at it and can modify it



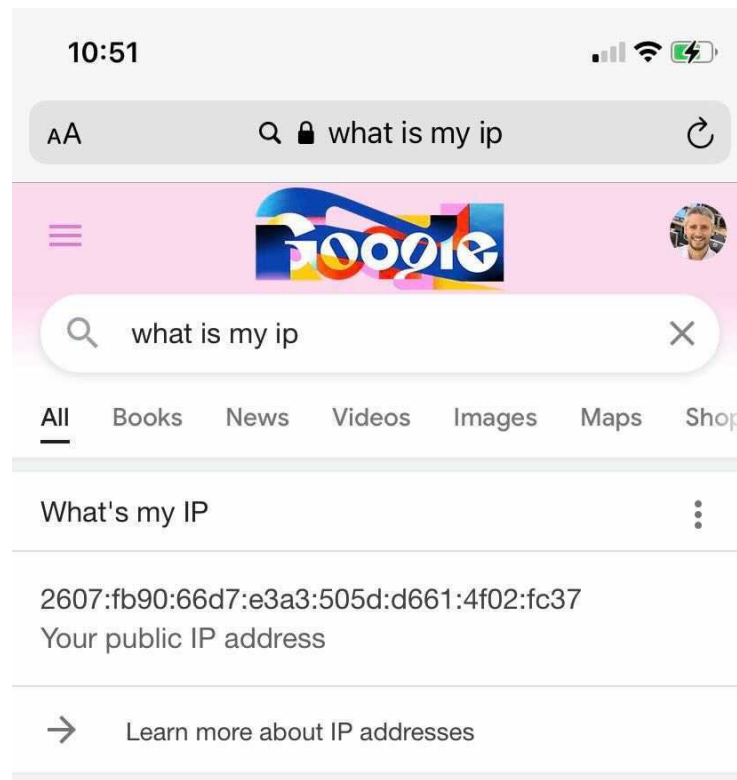
IPv6 transition is difficult

- Need to support both protocols at the same time
 - ◆ Complicated... if a destination has both a IPv4 and IPv6 address which to use?
- Less need in developed world -> slower adoption
- That said...
 - ◆ All major operating systems now support IPv6
 - ◆ All major router vendors
 - ◆ US Mobile carriers (e.g., Tmobile, Verizon, etc)
 - ◆ Offered by many US wired ISPs (Spectrum, Xfinity)
- IPv6 is dominating mobile carrier traffic

Check your phone now!



Try with WiFi on and WiFi off. Cellular almost 100% should be v6 to Google



Summary



- You can't route efficiently on flat address spaces
 - ◆ You'd need a table the size of all hosts on the Internet
 - ◆ You'd need to send updates about that table to everyone
- Network-layer addressing is done hierarchically
 - ◆ Routing prefix + host suffix
 - ◆ Originally, this split was done statically (class-based addressing)
 - ◆ Now it is done dynamically (CIDR)
 - ◆ Requires more complex forwarding table lookup
 - ◆ Allows contiguous chunks of address space to be aggregated (for the purposes of routing) into fewer prefixes
- IPv6 fixed address shortcomings of IPv4



For Next Time

- Project 2 and Homework 2!
- Read 3.3, 4.2