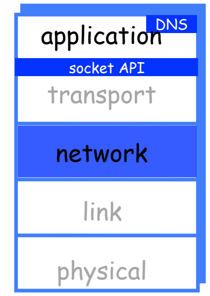
eecs489 Computer Networks

Lecture 6: IPv4, CIDR, ICMP

Network Layer

Where are we now?



Previously . . . the Internet is a packet switched network:

- data is parceled into packets
- each packet carries a destination address
- each packet is routed independently

Packet and Packet Header

Protocols are rules ("syntax" and "grammar") governing communication between nodes

Just as with the postal system, the "content" you want to send must be put into an envelope and the envelope must be addressed

The "envelope" in this case is the packet header
The format of a packet header is part of the protocol

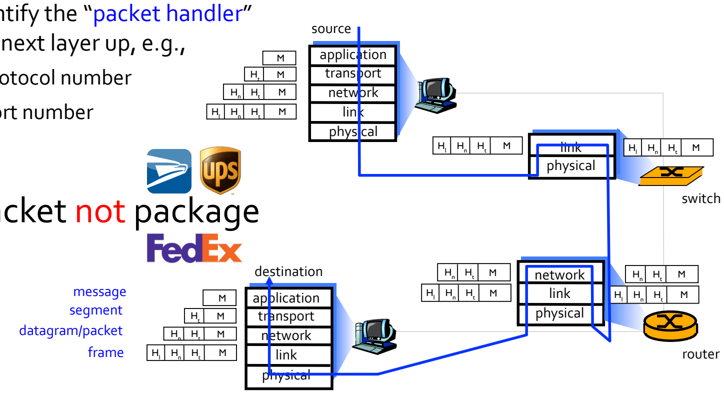
For the Internet, the network-layer protocol is the Internet Protocol (IP)

Encapsulation

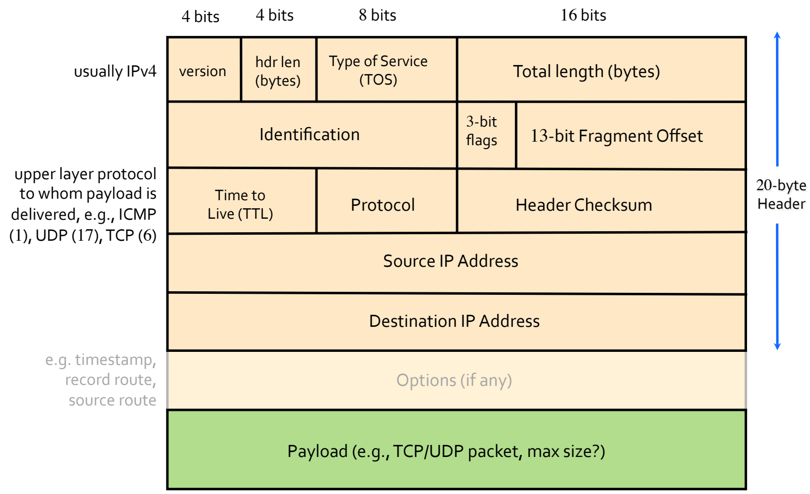
Each protocol has its own "envelope"

- each protocol attaches its header to the packet
- so we have a protocol wrapped/encapsulated inside another protocol
- each layer of header contains a protocol de-multiplexing field to identify the "packet handler" the next layer up, e.g.,
  - protocol number
  - port number

packet not package



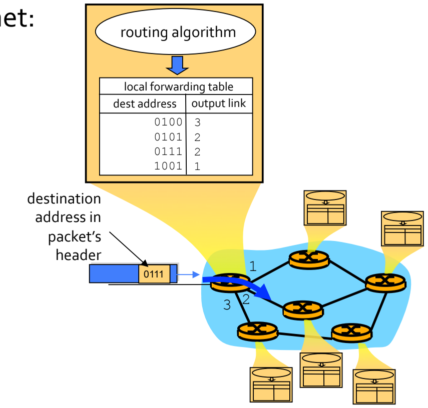
IPv4 Packet Header Format



Packet Forwarding

Goal: deliver packets through routers from source to destination

- source node puts destination address in packet header
- each router node on the Internet:
  - looks up destination address in its routing/forwarding table
    - we'll study several path selection (i.e., routing) algorithms
  - sends the packet to the next hop towards the destination
    - routes may change during session
    - analogy: driving, asking directions

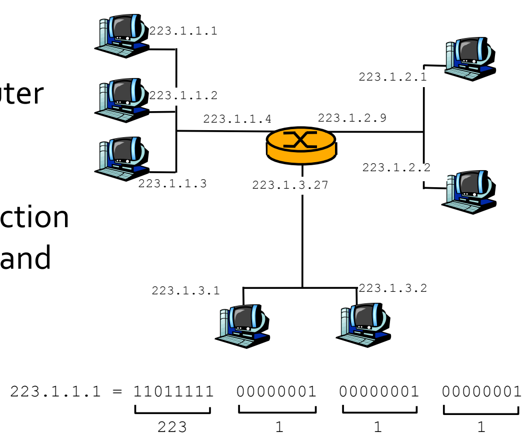


IPv4 Addressing: Introduction

IPv4 address: 32-bit identifier for host/router interface

interface (`if`): connection between host/router and physical link

- routers typically have multiple interfaces
- host may have multiple interfaces



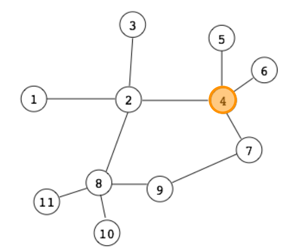
Flat vs. Hierarchical Addressing

Flat addressing:

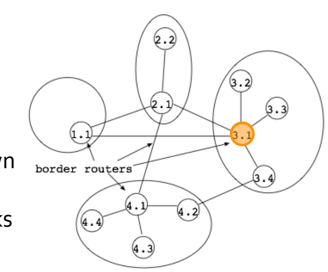
- each router needs 10 entries in its routing table

Hierarchical addressing:

- hosts only need to know the default router, usually its border router
- each border router keeps in its routing table:
  - addresses of all hosts within its own network
  - next hop address of other networks (Cf. forwarding table stores the outgoing interface number)



4	
1	2
2	2
3	2
4	-
5	5
6	6
7	7
8	2
9	7
10	2
11	2



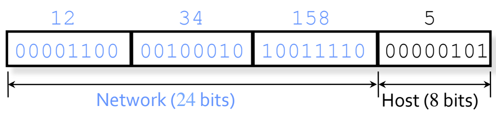
3.1	
2.*	2.1
1.*	1.1
4.*	2.1
3.2	3.2
3.3	3.3
3.4	3.4

IPv4 Addressing

Independent of physical hardware address

32-bit number represented as dotted decimal:
- for ease of reference
- each # is the decimal representation of an octet

Divided into two parts:
- network prefix, globally assigned
  - route to network first
- host ID, assigned locally

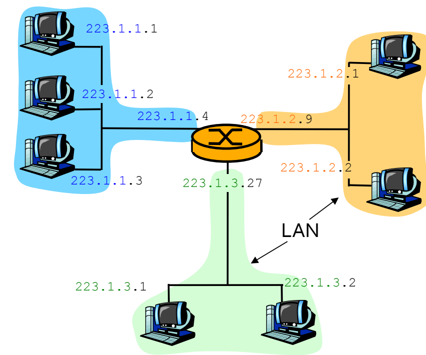


Example: `12.34.158.0/24`
is a 24-bit network prefix with 2^8 host addresses

Subnets

A network can be further divided into subnets

What makes a subnet ?
- the IP addresses of all interfaces within a subnet have the same network prefix
- hosts within a subnet can physically reach each other without intervening router



a network consisting of 3 subnets

Classfull Addresses

For the example network prefix: `12.34.158.0/24`
- how many hosts can the network have?

What is a good partition of the 32-bit address space between the network and host parts?

Historically . . . classfull addresses:

Class A: `0[10]*`, very large /8 blocks (e.g., MIT has `18.0.0.0/8`)

Class B: `10[10]*`, large /16 blocks (e.g,. UM has `141.213.0.0/16`)

Class C: `110[10]*`, small /24 blocks (e.g., AT&T Labs has `192.20.225.0/24`)

Class D: `1110[10]*`, multicast groups

Class E: `11110[10]*`, reserved for future use

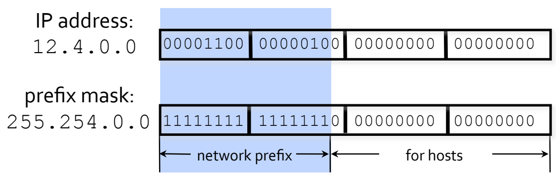
Classfull Addresses

Problems:
1. everybody wanted a Class B address (the Goldilock problem)
2. address space usage became inefficient
3. routing table explosion
4. and then, address space became scarce…
   - by 1992, half of Class B had been allocated, would have been exhausted by 3/94

Solution:
Make network portion of address arbitrary length, determined by a prefix mask
- uses two 32-bit numbers to represent a network address
  - network address = IP address & prefix mask

Classless InterDomain Routing (CIDR)

Usually written as `a.b.c.d/x`, where `x` is number of bits in the network portion of the address: `12.4.0.0/15`

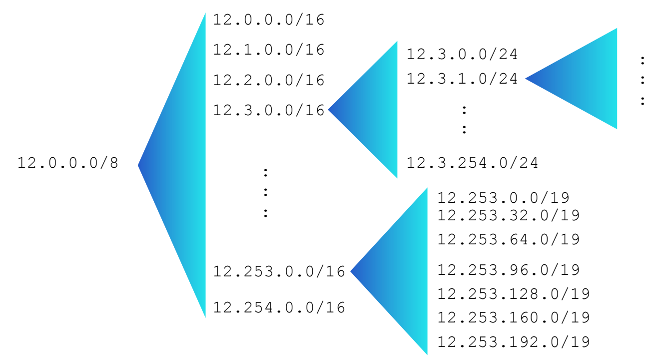


Another example (not at octet boundary!):

CIDR: Hierarchical Address Allocation

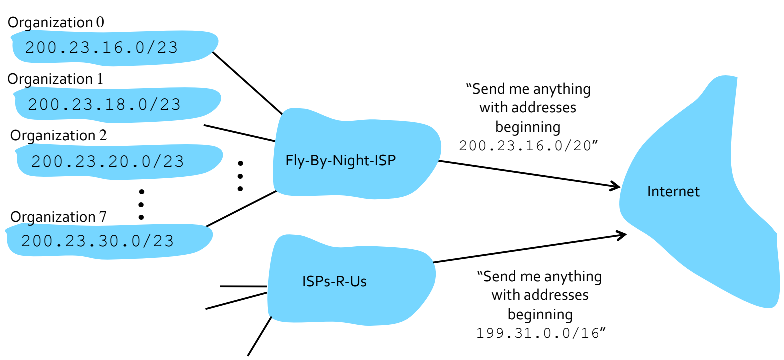
Prefixes are key to Internet routing scalability

- address allocation by ICANN, ARIN/RIPE/APNIC and by ISPs
- routing protocols and packet forwarding based on prefixes
- today, routing tables contain ~150,000-200,000 prefixes



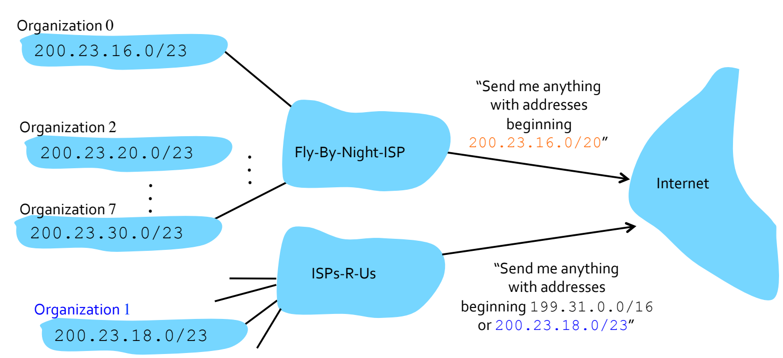
CIDR: Route Aggregation

Hierarchical addressing allows efficient advertisement of routing information:



Longest Prefix Match: More Specific Routes

ISPs-R-Us has a more specific route to Organization 1



Packet Forwarding Summary

Forwarding is destination-based:

- packet has a destination address
- router identifies longest-matching prefix

Forwarding table:

- maps each IP prefix to next-hop link(s)
- entries can be statically configured
  - e.g., "map `12.34.158.0/24` to Serial`0/0.1`"

forwarding table

destination `12.34.158.5` →

```
4.0.0.0/8
4.83.128.0/17
12.0.0.0/8
12.34.158.0/24
126.255.103.0/24
```

outgoing link Serial`0/0.1`

But, this doesn't adapt to

- failures
- new equipment
- the need to balance load
- …

That is where routing protocols come in… [more on routing later in the term]

Special IPv4 Addresses

Network identification:

- `0`s on host part, e.g. ,`141.212.0.0` (cannot be used to send packets)

Directed broadcast:

- `0xffff` on host part, e.g., `141.212.255.255`
- broadcast to all hosts on a network (e.g., `141.212`) (not implemented)

Limited broadcast:

- `0xffffffff`, received by all hosts on LAN, not forwarded beyond LAN

This computer:

- `0.0.0.0` used at startup to solicit IP address with RARP (deprecated)

Loopback address:

- `127.*.*.*` (usually `127.0.0.1`), named `localhost`
- packets sent to `localhost` traverse down the kernel networking code to the network layer and back up to the application without traversing the network, useful for testing networking code

Internet Control Message Protocol

ICMP is used by hosts and routers to communicate network-level information

- error reporting: unreachable host, network, port, protocol
- echo request/reply (used by `ping`)
- ICMP error message does not trigger another ICMP message

ICMP runs on network-layer, but "above" IP:

- ICMP messages are carried inside IP datagrams

ICMP message format:

type (8 bits)	code (8 bits)	checksum (16 bits)
The first 8 bytes of IP datagram causing error		

Internet Control Message Protocol

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	4	frag needed but DF set
3	6	dest network unknown
3	7	dest host unknown
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Traceroute and ICMP

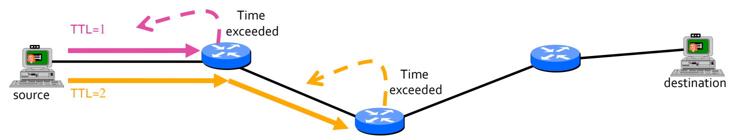
How to discover the routers along a path?

Source sends a series of UDP packets to destination

- first 3 packets have TTL set to 1, spaced 3 secs apart
- next 3 packets have TTL set to 2, etc.
- packets are all sent to an unused port number

When packet's TTL expired:

- router discards packet
- and sends to source an ICMP message (type 11, code 0)
- message includes IP address of router

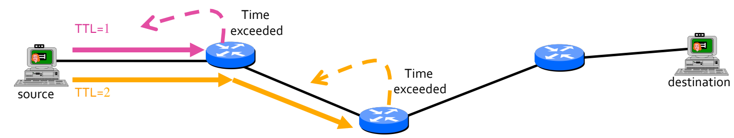


Traceroute and ICMP

When ICMP message arrives back at the source, source calculates round-trip time (RTT)

Stopping criterion:

- UDP packets eventually arrive at destination host
- destination returns ICMP "destination port unreachable" message (type 3, code 3)
- when source gets this ICMP message, it stops sending UDP packets



"Real" Internet Delays and Routes

Three delay measurements to 141.212.110.1

traceroute to eurocom.fr (92.243.13.96), 64 hops max, 40 byte packets

```
 1  141.212.110.1 (141.212.110.1)  1.356 ms  0.560 ms  1.277 ms
 2  vss-cse.engin.umich.edu (141.213.127.134)  0.486 ms  0.291 ms  0.291 ms
 3  l3-caen-bin-arb.r-bin-arbl.umnet.umich.edu (192.12.80.177)  0.373 ms  0.348 ms  0.333 ms
 4  l3-barb-bseb-2.r-bin-seb.umnet.umich.edu (192.12.80.11)  0.852 ms  0.576 ms  0.709 ms
 5  v-bin-seb-inet-aa2.aa2.mich.net (192.12.80.37)  0.527 ms  0.544 ms  0.515 ms
 6  ae0x76.wsu5.mich.net (198.108.23.9)  1.876 ms  1.928 ms  1.864 ms
 7  ae1x19.sfld-cor-123net.mich.net (198.108.23.20)  2.234 ms  2.261 ms  2.247 ms
 8  xe-8-3-0.1018.asbn0.tr-cps.internet2.edu (198.71.47.25)  22.498 ms  22.594 ms  34.036 ms
 9  64.57.20.106 (64.57.20.106)  22.737 ms  22.698 ms  22.681 ms
10  * * *
11  xe-1-1-0.mpr1.bwi9.us.above.net (64.125.22.134)  24.450 ms  24.394 ms  24.113 ms
12  64.125.192.86 (64.125.192.86)  115.416 ms  115.496 ms  115.670 ms
13  xe3-4-core4-d.paris.gandi.net (217.70.176.233)  116.071 ms  116.125 ms  115.971 ms
14  xe2-6-3-dist3-d.paris.gandi.net (217.70.176.182)  115.550 ms  115.652 ms  115.690 ms
15  srv4.felten.biz (92.243.13.96)  116.896 ms  116.925 ms  116.768 ms
```

* means no reponse (probe lost, router not replying, 3 secs timer)

trans-oceanic link