CS244a: An Introduction to Computer Networks

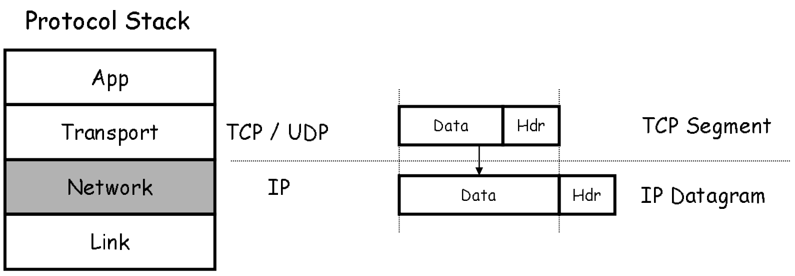
Handout 4: Layer 3 and the Internet Protocol (IP)

Nick McKeown
Professor of Electrical Engineering and Computer Science, Stanford University
nickm@stanford.edu
http://www.stanford.edu/~nickm

Outline

- IP: The Internet Protocol
  - Service characteristics
  - The IP Datagram format
  - IP addresses
  - Classless Interdomain Routing (CIDR)
  - An aside: Turning names into addresses (DNS)
  - Forwarding IP Datagrams

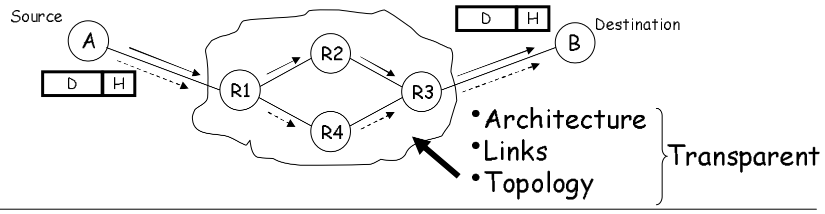
The Internet Protocol (IP)



The Internet Protocol (IP)

- Characteristics of IP

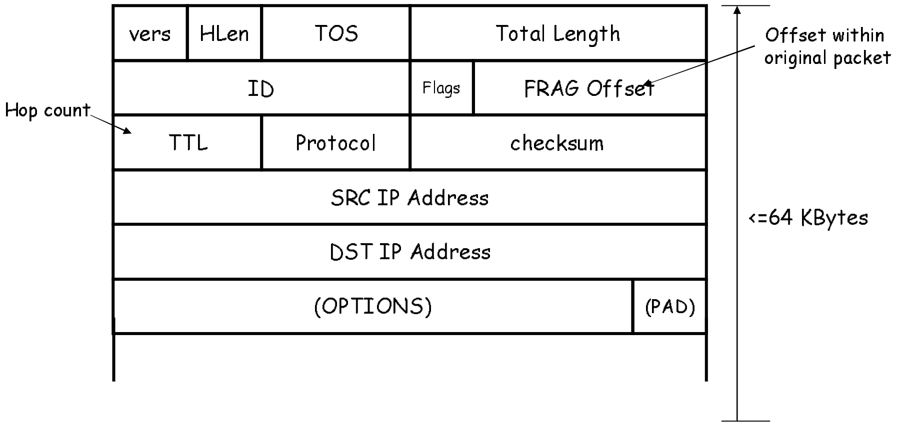
• CONNECTIONLESS:	mis-sequencing
• UNRELIABLE:	may drop packets…
• BEST EFFORT:	… but only if necessary
• DATAGRAM:	individually routed



- Architecture
- Links
- Topology

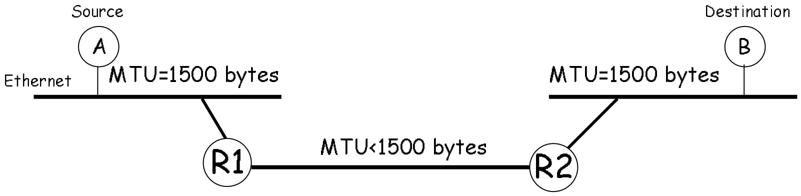
Transparent

The IP Datagram

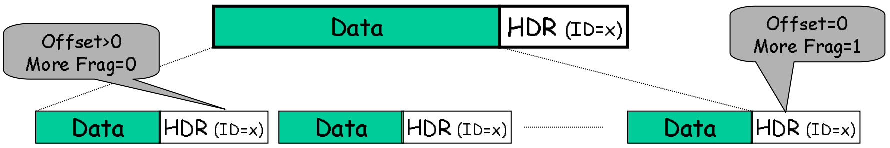


Fragmentation

Problem: A router may receive a packet larger than the maximum transmission unit (MTU) of the outgoing link.



Solution: R1 fragments the IP datagram into mutiple, self-contained datagrams.



Fragmentation

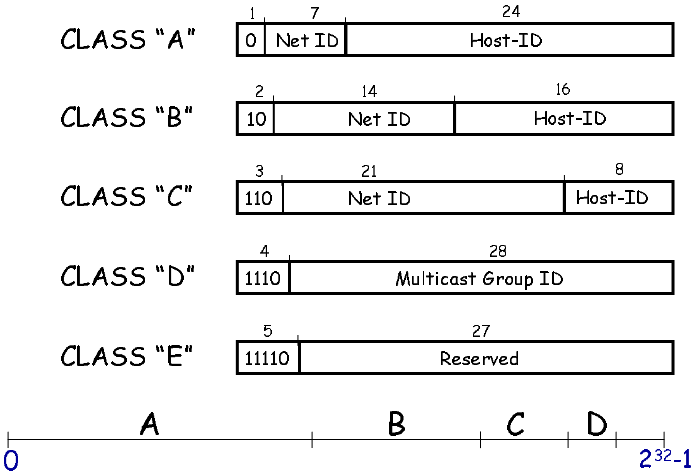
- Fragments are re-assembled by the destination host; not by intermediate routers.
- To avoid fragmentation, hosts commonly use path MTU discovery to find the smallest MTU along the path.
- Path MTU discovery involves sending various size datagrams until they do not require fragmentation along the path.
- Most links use MTU>=1500bytes today.
- Try:
  `traceroute -f berkeley.edu 1500` and
  `traceroute -f berkeley.edu 1501`
- (DF=1 set in IP header; routers send "ICMP" error message, which is shown as "!F").
- Bonus: Can you find a destination for which the path MTU < 1500 bytes?

IP Addresses

- IP (Version 4) addresses are 32 bits long
- Every interface has a unique IP address:
  - A computer might have two or more IP addresses
  - A router has many IP addresses
- IP addresses are hierarchical
  - They contain a network ID and a host ID
  - E.g. Stanford addresses start with: 171.64…
- IP addresses are assigned statically or dynamically (e.g. DHCP)
- IP (Version 6) addresses are 128 bits long

IP Addresses

Originally there were 5 classes:



IP Addresses

Examples

Class "A" address: www.mit.edu
18.181.0.31
(18<128 => Class A)

Class "B" address: mekong.stanford.edu
171.64.74.155
(128<171<128+64 => Class B)

IP Addressing

Problem:

- Address classes were too “rigid”. For most organizations, Class C were too small and Class B too big. Led to inefficient use of address space, and a shortage of addresses.
- Organizations with internal routers needed to have a separate (Class C) network ID for each link.
- And then every other router in the Internet had to know about every network ID in every organization, which led to large address tables.
- Small organizations wanted Class B in case they grew to more than 255 hosts. But there were only about 16,000 Class B network IDs.

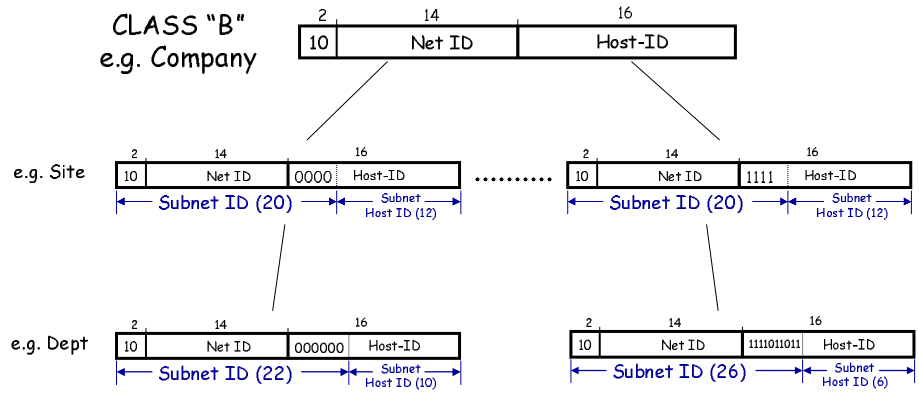
IP Addressing

Two solutions were introduced:

- Subnetting within an organization to subdivide the organization's network ID.
- Classless Interdomain Routing (CIDR) in the Internet backbone was introduced in 1993 to provide more efficient and flexible use of IP address space.

- CIDR is also known as “supernetting” because subnetting and CIDR are basically the same idea.

Subnetting

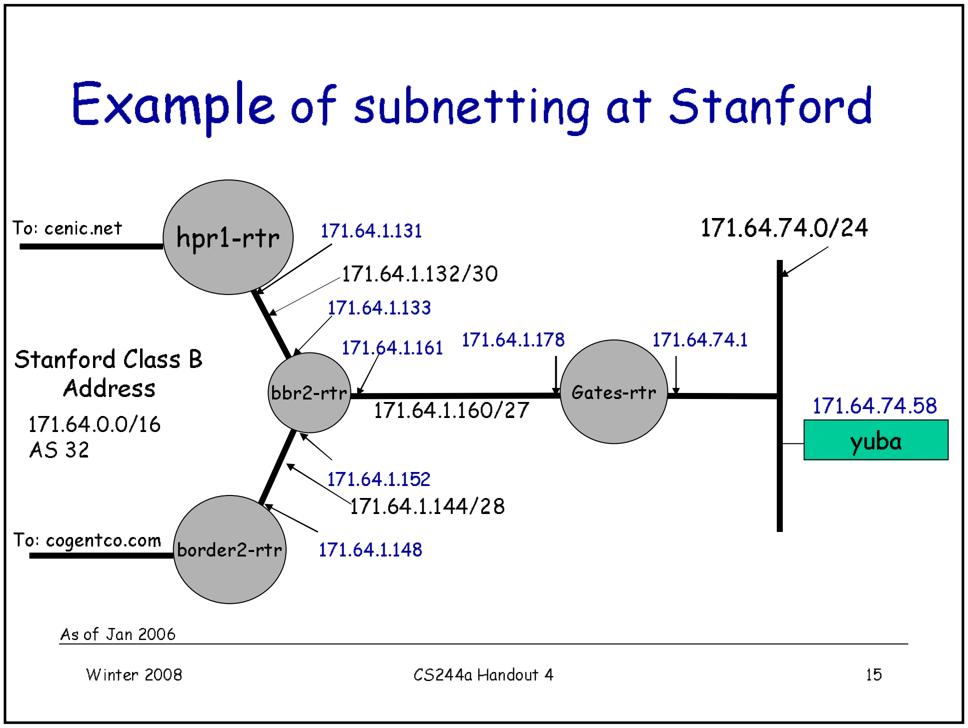


Subnetting

- Subnetting is a form of hierarchical routing.
- Subnets are usually represented via an address plus a subnet mask or "netmask".
- e.g.

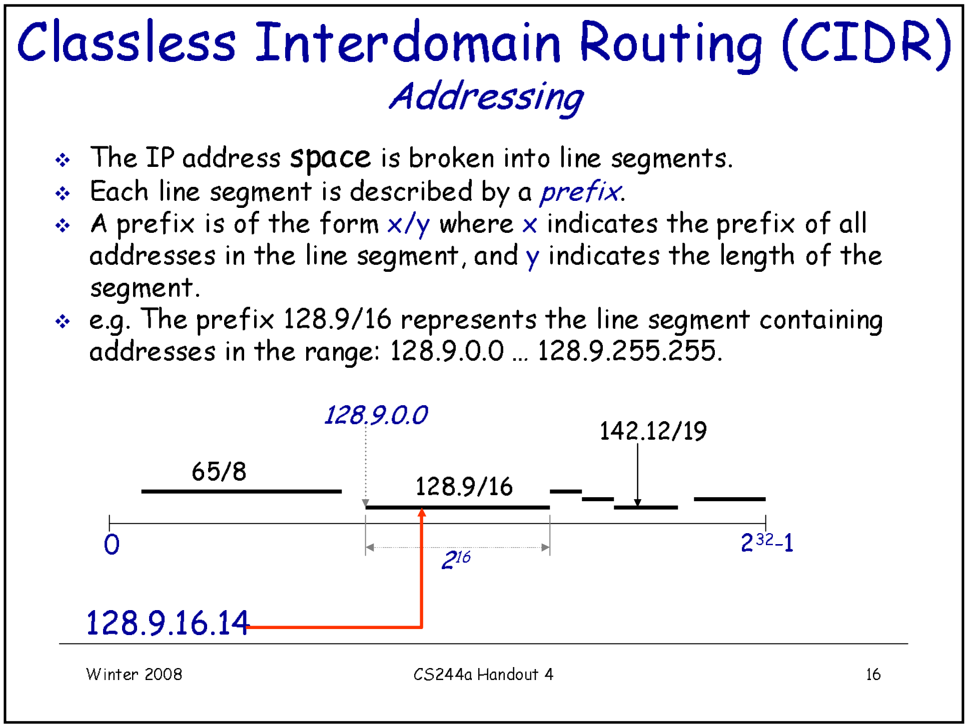
```
nickm@elaine17.Stanford.EDU > ifconfig hme0
 hme0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
 inet 171.64.15.82 netmask ffffff00 broadcast 171.64.15.255
```

- Netmask ffffff00: the first 24 bits are the subnet ID, and the last 8 bits are the host ID.
- Can also be represented by a "prefix + length", e.g. 171.64.15.0/24, or just 171.64.15/24.



Example of subnetting at Stanford

As of Jan 2006



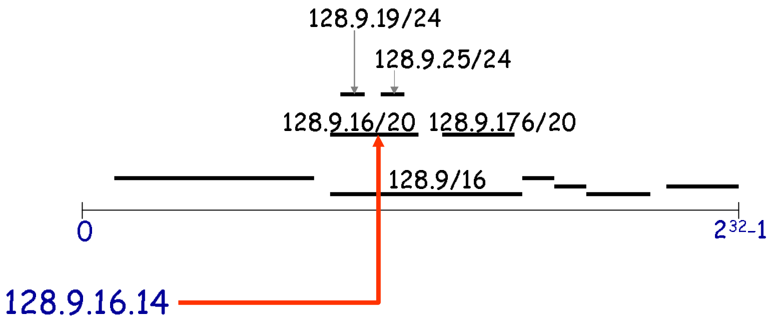
Classless Interdomain Routing (CIDR)

Addressing

- The IP address space is broken into line segments.
- Each line segment is described by a *prefix*.
- A prefix is of the form x/y where x indicates the prefix of all addresses in the line segment, and y indicates the length of the segment.
- e.g. The prefix 128.9/16 represents the line segment containing addresses in the range: 128.9.0.0 … 128.9.255.255.

Classless Interdomain Routing (CIDR)

Addressing



Most specific route = "longest matching prefix"

Classless Interdomain Routing (CIDR)

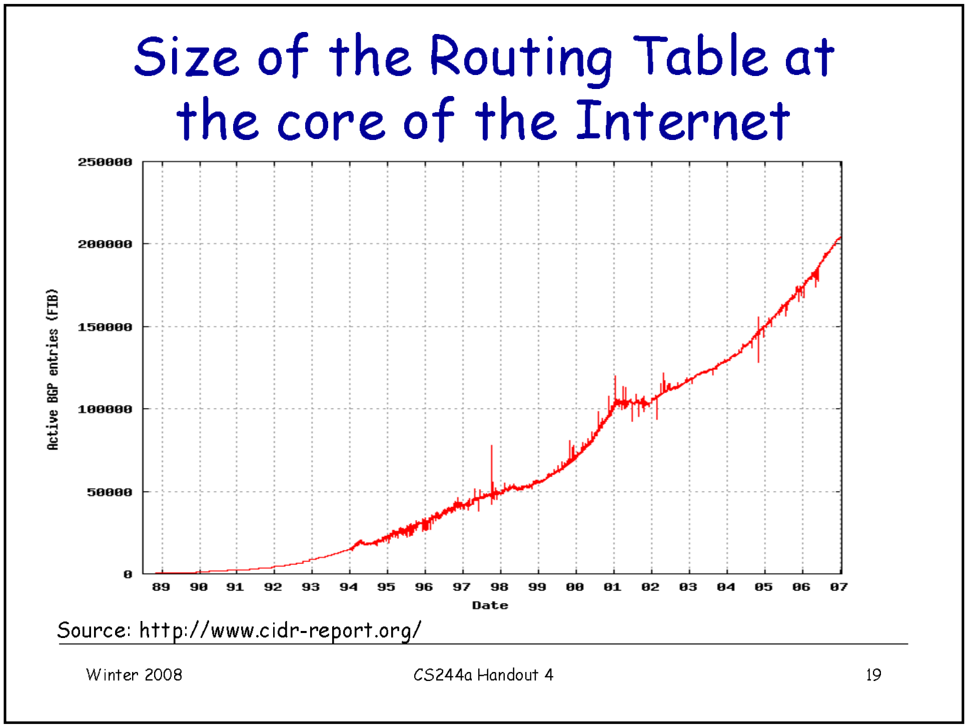
Addressing

Prefix aggregation:

- If a service provider serves two organizations with prefixes, it can (sometimes) aggregate them to form a shorter prefix. Other routers can refer to this shorter prefix, and so reduce the size of their address table.
- E.g. ISP serves 128.9.14.0/24 and 128.9.15.0/24, it can tell other routers to send it all packets belonging to the prefix 128.9.14.0/23.

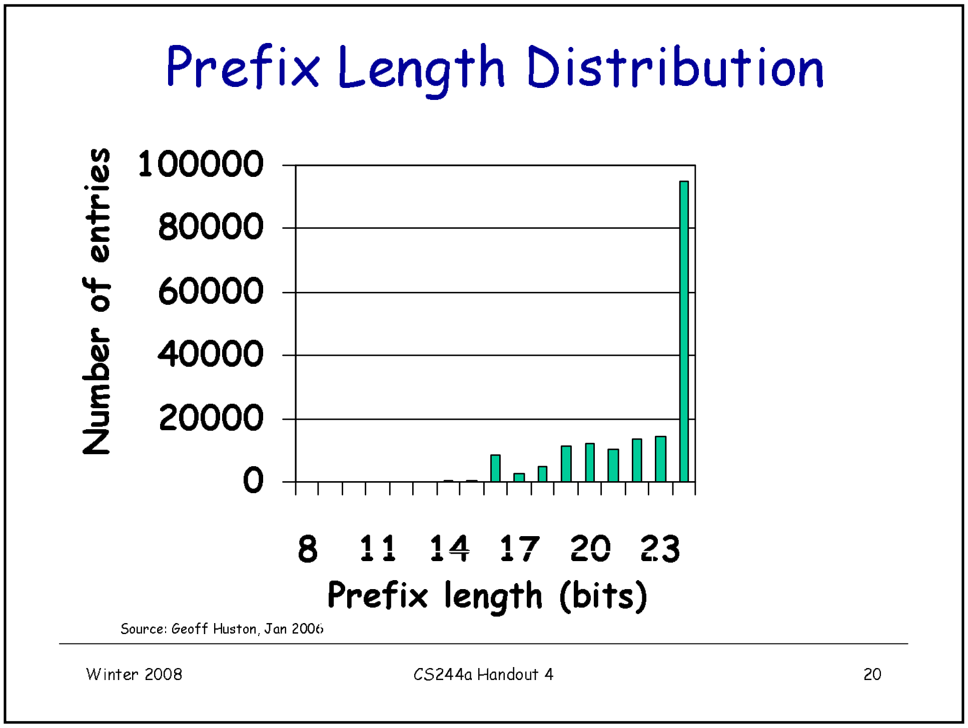
ISP Choice:

- In principle, an organization can keep its prefix if it changes service providers.



Size of the Routing Table at the core of the Internet

Source: http://www.cidr-report.org/



Prefix Length Distribution

Source: Geoff Huston, Jan 2006

Mapping Computer Names to IP addresses
The Domain Naming System (DNS)

Names are hierarchical and belong to a domain:

- e.g. elaine17.stanford.edu
- Common domain names: .com, .edu, .gov, .org, .net, .uk (or other country-specific domain).
- Top-level names are assigned by the Internet Corporation for Assigned Names and Numbers (ICANN).
- A unique name is assigned to each organization.

DNS Client-Server Model

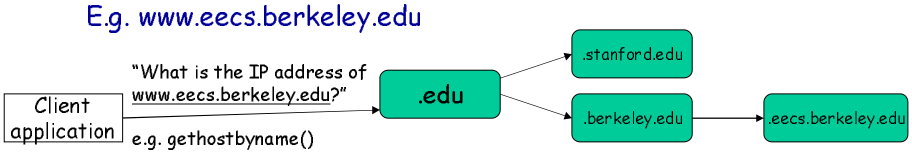
- DNS maintains a hierarchical, distributed database of names.
- Servers are arranged in a hierarchy.
- Each domain has a "root" server.
- An application needing an IP address is a DNS client.

Mapping Computer Names to IP addresses
The Domain Naming System (DNS)

A DNS Query

1. Client asks local server.
2. If local server does not have address, it asks the root server of the requested domain.
3. Addresses are cached in case they are requested again.

E.g. www.eecs.berkeley.edu



Example: On elaine machines, try "host www.mit.edu"

An example of names and addresses

Mapping the path between two hosts

```
nickm@yuba.Stanford.EDU > host yuba
     yuba.Stanford.EDU. has address 171.64.74.58

nickm@yuba.Stanford.EDU >traceroute www.mit.edu
traceroute to www.mit.edu (18.7.22.83), 30 hops max, 38 byte packets
 1  Gates-rtr (171.64.74.1)  0.539 ms  0.381 ms  0.388 ms
 2  bbr2-rtr (171.64.1.161)  0.373 ms  0.366 ms  0.368 ms
 3  hpr1-rtr (171.64.1.131)  0.832 ms  0.992 ms  0.902 ms
 4  hpr-svl-hpr--stan-ge.cenic.net (137.164.27.161)  1.596 ms  0.988 ms  1.363 ms
 5  lax-hpr--svl-hpr-10ge.cenic.net (137.164.25.12)  9.065 ms  8.895 ms  8.948 ms
 6  abilene-LA--hpr-lax-gsr1-10ge.cenic.net (137.164.25.3)  8.884 ms  8.816 ms 9.080 ms
 7  snvang-losang.abilene.ucaid.edu (198.32.8.95)  16.414 ms  16.562 ms  16.763ms
 8  dnvrng-snvang.abilene.ucaid.edu (198.32.8.2)  42.058 ms  41.515 ms  41.068 ms
 9  kscyng-dnvrng.abilene.ucaid.edu (198.32.8.14)  52.901 ms  56.792 ms  51.923ms
10  iplsng-kscyng.abilene.ucaid.edu (198.32.8.80)  71.886 ms  61.674 ms  64.450ms
11  chinng-iplsng.abilene.ucaid.edu (198.32.8.76)  64.987 ms  64.824 ms  64.839ms
12  nycmng-chinng.abilene.ucaid.edu (198.32.8.83)  85.108 ms  84.846 ms  85.009ms
13  nox230gw1-PO-9-1-NoX-NOX.nox.org (192.5.89.9)  90.206 ms  90.227 ms  90.044ms
14  nox230gw1-PEER-NoX-MIT-192-5-89-90.nox.org (192.5.89.90)  89.922 ms  90.332ms
    90.242 ms
15  B24-RTR-3-BACKBONE.MIT.EDU (18.168.0.26)  90.193 ms  90.373 ms  90.374 ms
16  WWW.MIT.EDU (18.7.22.83)  90.762 ms  90.996 ms  90.271 ms
```

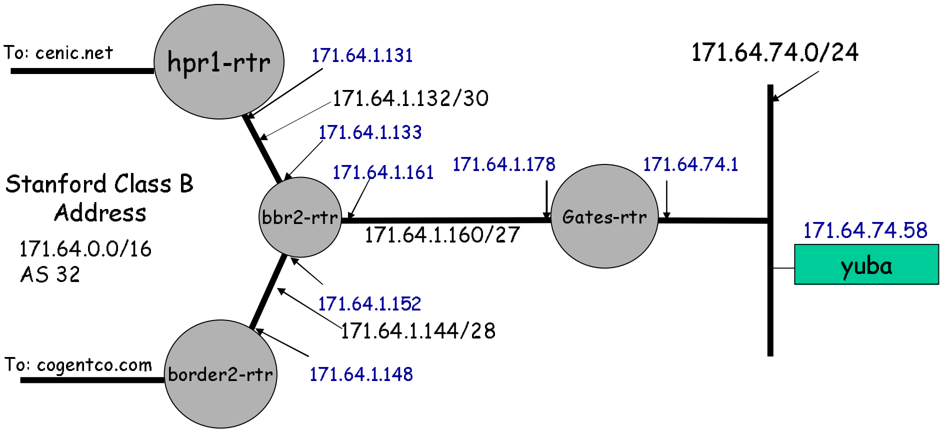
Example

Mapping the path between two hosts

```
nickm@yuba.Stanford.EDU > host bbr2-rtr.stanford.edu | sort -n
    bbr2-rtr.Stanford.EDU has address 128.12.1.49
    bbr2-rtr.Stanford.EDU has address 171.64.0.126
    bbr2-rtr.Stanford.EDU has address 171.64.1.133
    bbr2-rtr.Stanford.EDU has address 171.64.1.152
    bbr2-rtr.Stanford.EDU has address 171.64.1.161
    bbr2-rtr.Stanford.EDU has address 171.64.1.242
    bbr2-rtr.Stanford.EDU has address 171.64.1.26
    bbr2-rtr.Stanford.EDU has address 171.64.1.9
    bbr2-rtr.Stanford.EDU has address 171.64.1.97
    bbr2-rtr.Stanford.EDU has address 171.64.3.242
    bbr2-rtr.Stanford.EDU has address 171.64.7.60
    bbr2-rtr.Stanford.EDU has address 171.66.1.249
    bbr2-rtr.Stanford.EDU has address 171.66.16.1
    bbr2-rtr.Stanford.EDU has address 171.67.1.193
    bbr2-rtr.Stanford.EDU has address 171.67.20.1
    bbr2-rtr.Stanford.EDU has address 171.67.254.242
    bbr2-rtr.Stanford.EDU has address 171.67.255.126
    bbr2-rtr.Stanford.EDU has address 172.24.1.9
    bbr2-rtr.Stanford.EDU has address 172.27.20.1
    bbr2-rtr.Stanford.EDU has address 192.168.2.129
    bbr2-rtr.Stanford.EDU has address 192.168.7.154
```

Example

Mapping the path between two hosts



As of Jan 2006

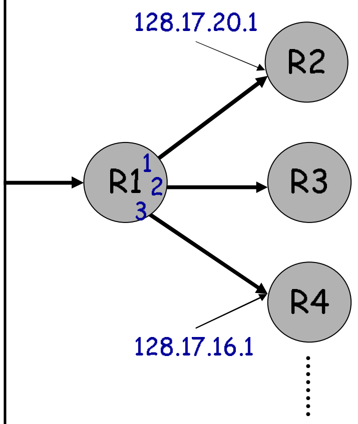
An aside:

Error Reporting (ICMP) and traceroute

Internet Control Message Protocol:

- Used by a router/end-host to report some types of error:
- E.g. Destination Unreachable: packet can't be forwarded to/towards its destination.
- E.g. Time Exceeded: TTL reached zero, or fragment didn't arrive in time. `Traceroute` uses this error to its advantage.
- An ICMP message is an IP datagram, and is sent back to the source of the packet that caused the error.

How a Router Forwards Datagrams



e.g. 128.9.16.14 => Port 2

Prefix	Next-hop	Port
65/8	128.17.16.1	3
128.9/16	128.17.14.1	2
128.9.16/20	128.17.14.1	2
128.9.19/24	128.17.10.1	7
128.9.25/24	128.17.14.1	2
128.9.176/20	128.17.20.1	1
142.12/19	128.17.16.1	3

Forwarding/routing table

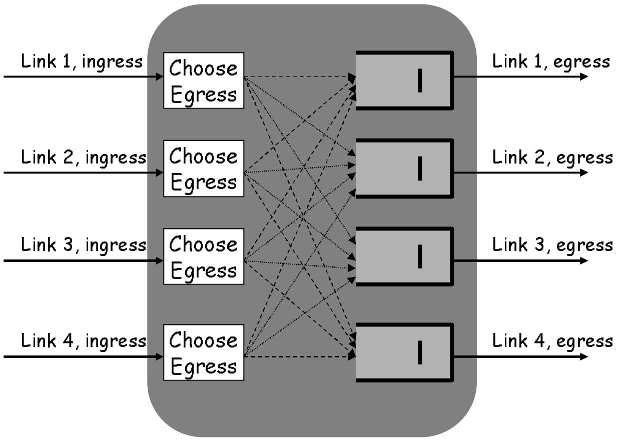
How a Router Forwards Datagrams

- Every datagram contains a destination address.
- The router determines the prefix to which the address belongs, and routes it to the"Network ID" that uniquely identifies a physical network.
- All hosts and routers sharing a Network ID share same physical network.

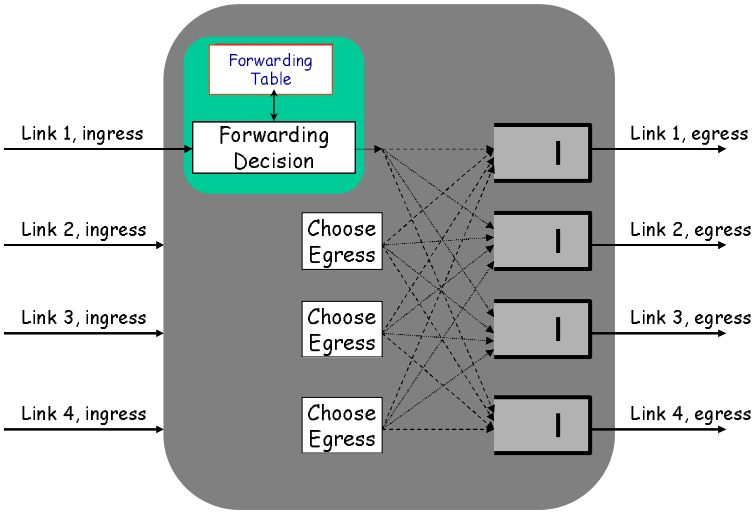
Forwarding Datagrams

- Is the datagram for a host on a directly attached network?
- If no, consult forwarding table to find *next-hop*.

Inside a router



Inside a router



Forwarding Table

Forwarding Decision

Link 1, ingress

Link 2, ingress

Link 3, ingress

Link 4, ingress

Choose Egress

Choose Egress

Choose Egress

Link 1, egress

Link 2, egress

Link 3, egress

Link 4, egress

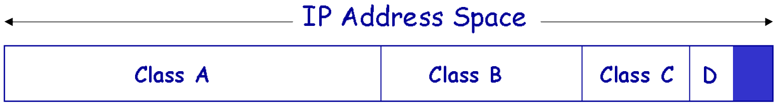
Forwarding in an IP Router

- Lookup packet DA in forwarding table.
  - If known, forward to correct port.
  - If unknown, drop packet.
- Decrement TTL, update header Checksum.
- Forward packet to outgoing interface.
- Transmit packet onto link.

Question: How is the address looked up in a real router?

Making a Forwarding Decision

Class-based addressing

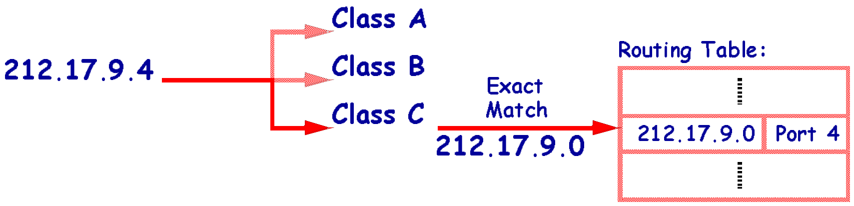

IP Address Space

Class A

Class B

Class C

D


212.17.9.4

Class A

Class B

Class C

Exact Match

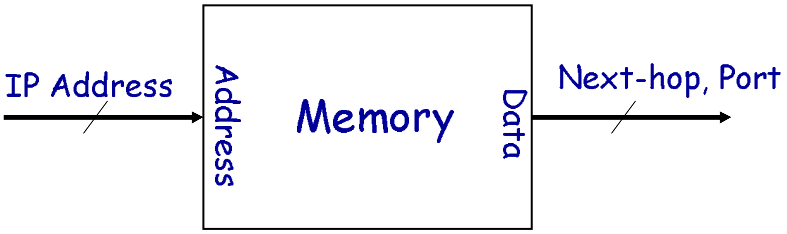
212.17.9.0

Routing Table:

⋮	
212.17.9.0	Port 4
⋮	

Exact Match: There are many well-known ways to find an exact match in a table.

Direct Lookup


IP Address

Address

Memory

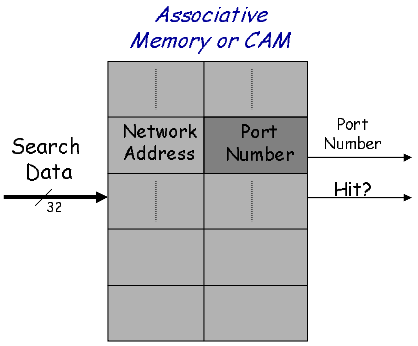
Data

Next-hop, Port

Problem: With 2^{32} addresses, the memory would require 4 billion entries.

Associative Lookups

"Contents addressable memory" (CAM)


Associative Memory or CAM

Search Data

32

Network Address

Port Number

Port Number

Hit?

Search data is compared with every entry in parallel

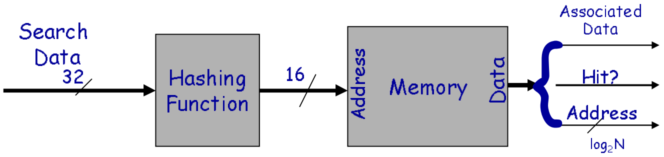
Advantages:

- Simple

Disadvantages

- Slow
- High Power
- Small
- Expensive

Hashed Lookups


Search Data

32

Hashing Function

16

Address

Memory

Data

Associated Data

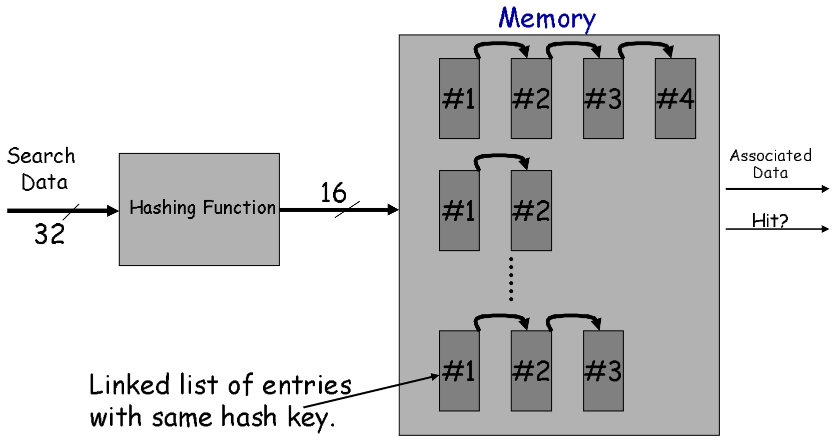
Hit?

Address

$\log_2 N$

Lookups Using Hashing

An example



Lookups Using Hashing

Advantages:

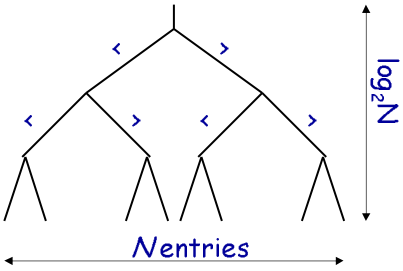
- Simple
- Expected lookup time can be small

Disadvantages

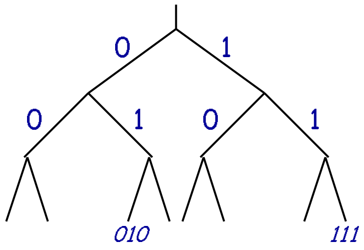
- Non-deterministic lookup time
- Inefficient use of memory

Trees and Tries

Binary Search Tree:



Binary Search Trie:
("reTRIEval")

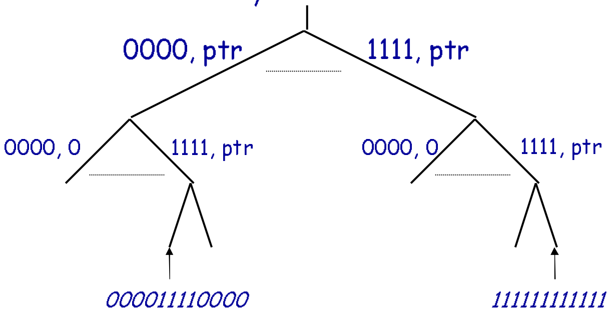


Requires 32 memory references, regardless of number of addresses.

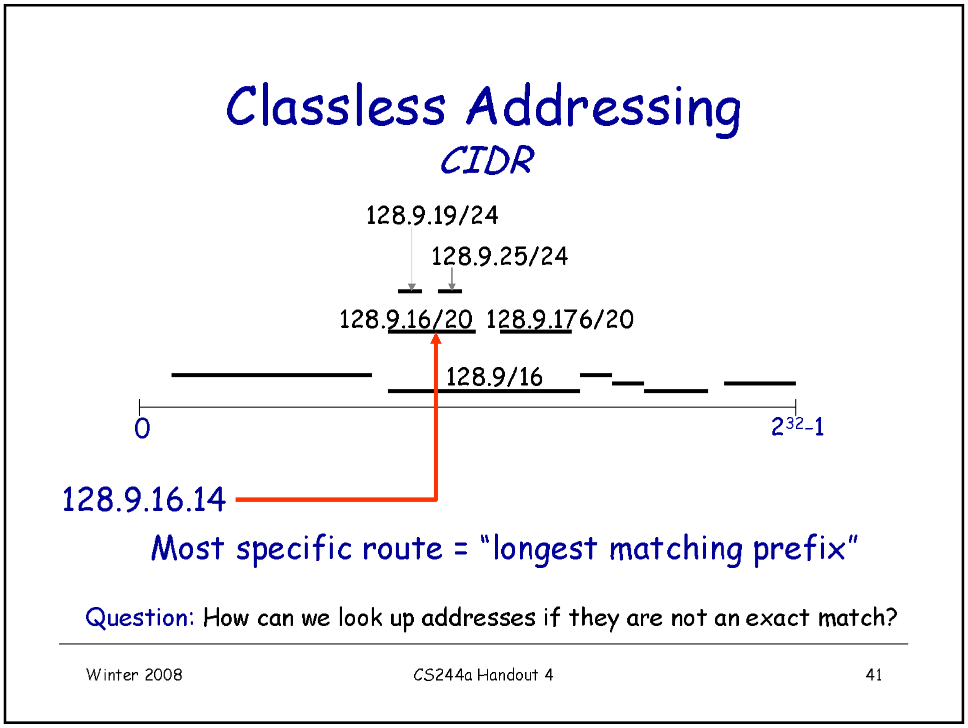
Search Tries

Multiway tries reduce the number of memory references

16-ary Search Trie



Question: Why not just keep increasing the degree of the trie?

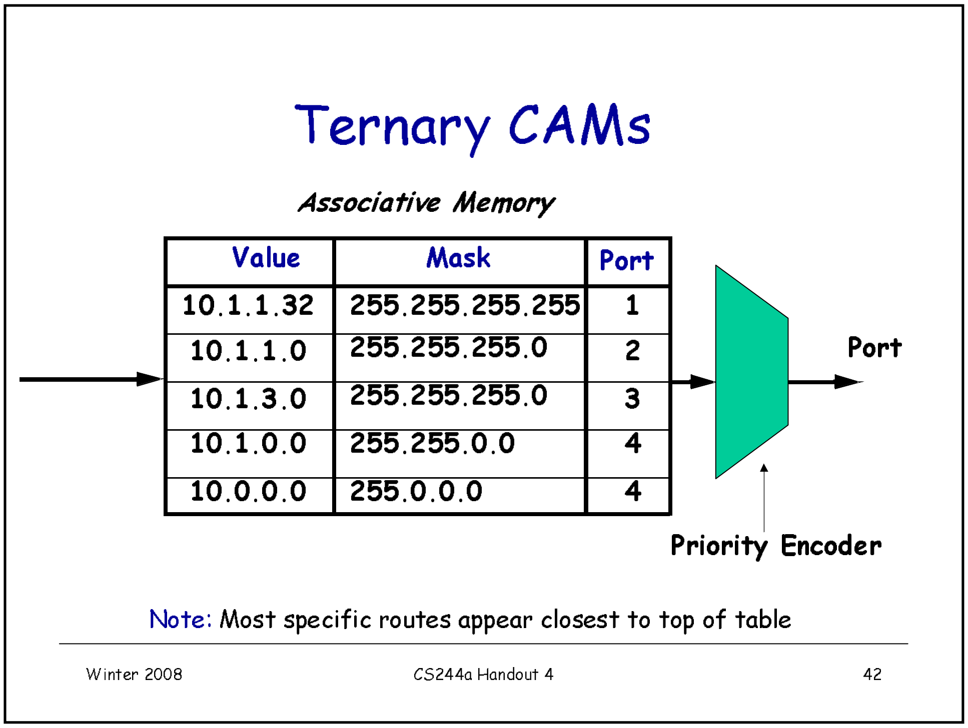


Classless Addressing

CIDR

Most specific route = "longest matching prefix"

Question: How can we look up addresses if they are not an exact match?



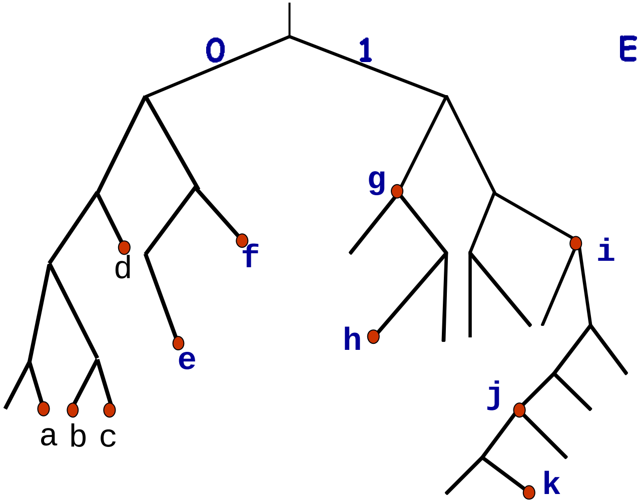
Ternary CAMs

Associative Memory

Value	Mask	Port
10.1.1.32	255.255.255.255	1
10.1.1.0	255.255.255.0	2
10.1.3.0	255.255.255.0	3
10.1.0.0	255.255.0.0	4
10.0.0.0	255.0.0.0	4

Note: Most specific routes appear closest to top of table

Longest prefix matches using Binary Tries

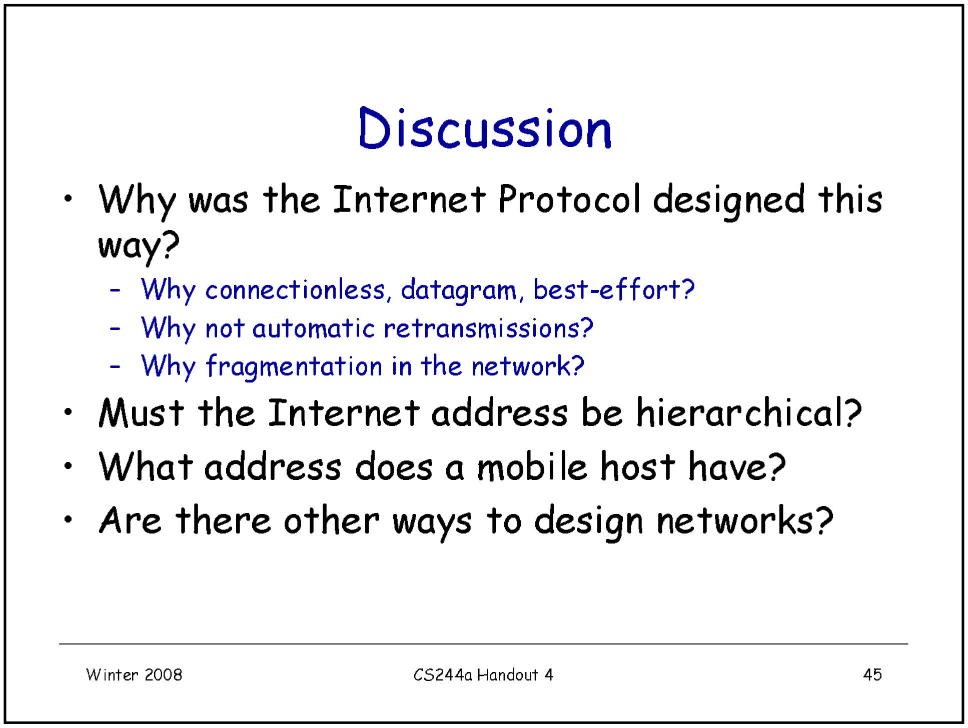


Example Prefixes:

a) 00001
b) 00010
c) 00011
d) 001
e) 0101
f) 011
g) 10
h) 1010
i) 111
j) 111100
k) 11110001

Lookup Performance Required

Line	Line Rate	Pkt-size=40B	Pkt-size=240B
T1	1.5Mbps	4.68 Kpps	0.78 Kpps
OC3	155Mbps	480 Kpps	80 Kpps
OC12	622Mbps	1.94 Mpps	323 Kpps
OC48	2.5Gbps	7.81 Mpps	1.3 Mpps
OC192	10 Gbps	31.25 Mpps	5.21 Mpps



Discussion

- Why was the Internet Protocol designed this way?
  - Why connectionless, datagram, best-effort?
  - Why not automatic retransmissions?
  - Why fragmentation in the network?
- Must the Internet address be hierarchical?
- What address does a mobile host have?
- Are there other ways to design networks?