CS519: Computer Networks

Lecture 2, part 2: Feb 4, 2004

IP (Internet Protocol)

More ICMP messages

- These were added over time
- RFC1191: Path MTU Discovery
  - Added the size of the limiting MTU to the ICMP Packet Too Big message
- RFC1256: Router Discovery
  - Allows a host to dynamically discover a default router
  - Router Advertisement, Router Solicitation

Path MTU discovery (PMTU)

- Host tries some large MTU, sends packets with the DF (Don't Fragment) bit set
- If it gets an ICMP Packet Too Big, it tries the MTU in the ICMP (if there is one), or a lower MTU if not
- There are various "well-known" MTUs it can try
- Without PMTU, hosts default to 1500 for local Ethernet destinations, and 576 for non-local destinations

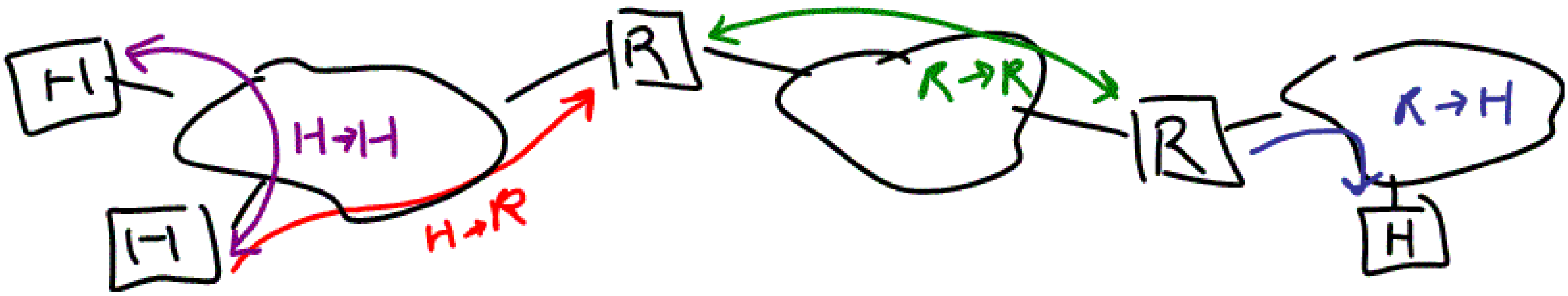
Recall new functions required by IP architecture

- Address resolution
  - How to determine the subnet address of the next hop (router or host)
  - A hard problem in the general case
- Fragmentation and reassembly
  - How to accommodate different MTUs (Maximum Transmission Unit) in different subnets

Router discovery and address resolution

CS519

- How do hosts discover routers over subnets
- How do hosts and routers find each other's subnet addresses?



But first, IP addresses

- Now we are getting to one of the two “cores” of the Internet
  - The IP address space
  - (The other is the DNS name space)
- Can’t understand router discovery and address resolution without understanding the IP address

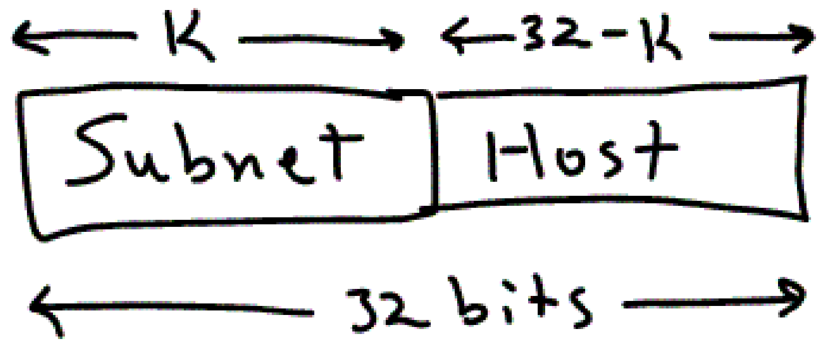
The most basic basics

- IP address is 32 bits long
- They are written like this: 128.93.44.6
  - “dotted-decimal” notation
  - Each decimal number represents 8 bits
  - (We’ll look at the “slash” notation later)
- Each host or router *interface* is identified by an IP address
  - The role of IP is to get packets to their corresponding interfaces, *not* to boxes per se

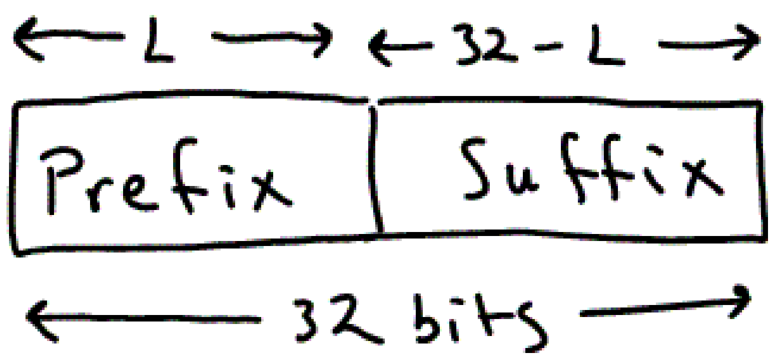
Structure of the IP address

CS519

From the point of view of the host or "last-hop" router



From the point of view of a router



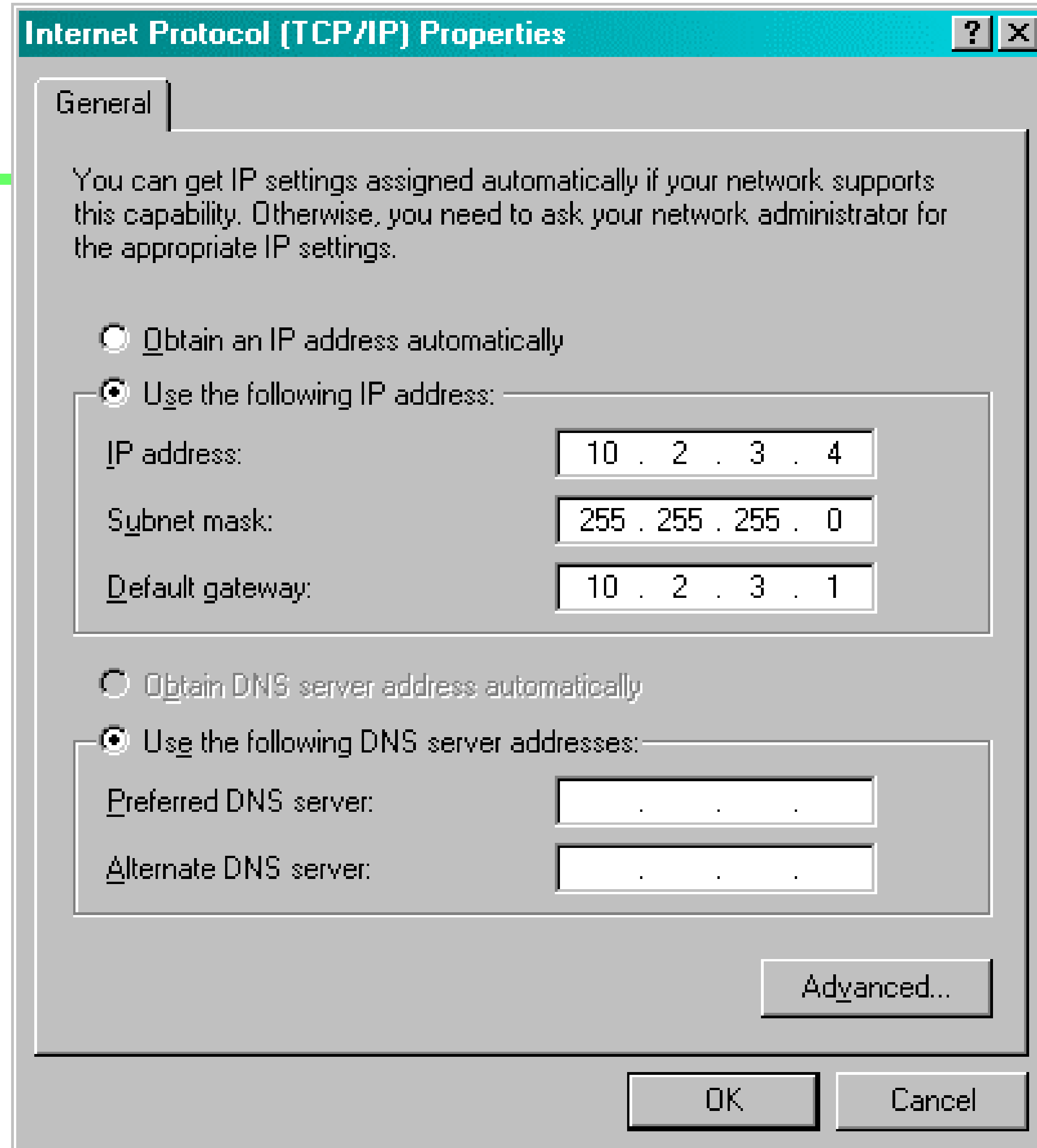
Host forwarding algorithm

- Upon receiving a packet either from an interface or from the upper layer
  - Is the destination me?
  - If not, is the destination on my subnet?
    - If so, discover subnet address of destination and transmit packet
  - If not, send the packet to my default router

This begs several questions

- How does the host know its own IP address?
- How does the host know the destination is on its own subnet?
- How does the host know its default router?
- How does the host discover the subnet address of another node?

In a nutshell . . .



Last-hop router forwarding algorithm

- Upon receiving a packet either from an interface or from the upper layer
  - Is the destination me?
  - If not, is the destination on my subnet?
    - If so, discover subnet address of destination and transmit packet
  - If not, send the packet to ~~my default~~ the next hop router

Non-Last-hop router forwarding algorithm

CS519

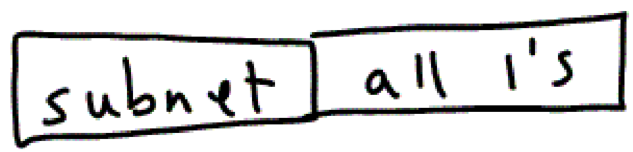
- Upon receiving a packet either from an interface or from the upper layer
  - Is the destination me?
  - ~~If not, is the destination on my subnet?~~
    - ~~If so, discover subnet address of destination and transmit packet~~
  - If not, send the packet to ~~my default~~ the next hop router

Other special IP addresses

Loopback address 127.0.0.1
(send to self
via interface)

Local broadcast address 255.255.255.255

Broadcast address
(sends packet to all interfaces
on the subnet!)



Multicast addresses

Router discovery and address resolution

- We saw how routers are discovered (through configuration)
- Hosts on a subnet don’t have to be “discovered” per se
  - Because the IP address is obtained from packet reception
- But the host subnet address needs to be discovered
  - This is called *address resolution*

Address resolution

CS519

- Approach differs depending on whether subnet is broadcast capable or not
- If broadcast capable:
  - An Address Resolution Protocol (ARP) query is broadcast to all nodes:
    - ARP_query(IP_addr)
  - The node with IP_addr responds:
    - ARP_reply(IP_addr, subnet_addr)
  - The querying node caches this for a while

Address resolution

- If not broadcast capable, two choices:
- Embed subnet address inside IP address!
  - This was done with the Arpanet
  - But most subsequent non-broadcast subnets had addresses at least as big as IP's (X.25, SMDS, ATM, …)
  - Can be done in IPv6 though
- Configure address resolution tables
  - In all nodes, or in a directory that nodes can query

Address resolution

- Large non-broadcast subnets to which hosts attach are essentially non-existent now---everything has “gone IP”
  - Though these still support routers (manually configured)

Where are we?

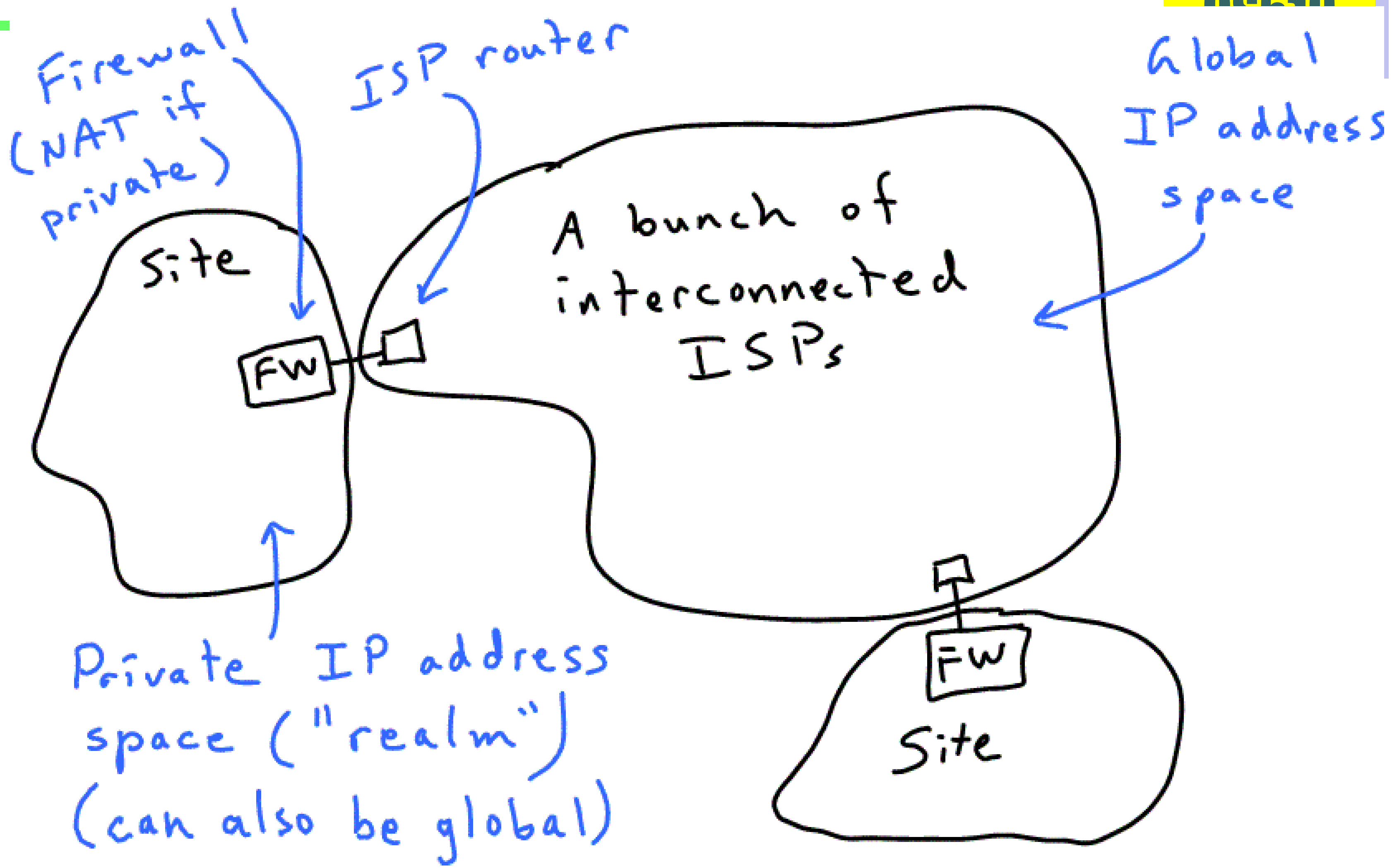
CS519

- We've looked down from IP:
  - We've examined **subnet** structure of the Internet and of IP addresses
  - We've seen how to resolve IP addresses to subnet addresses
  - We'll look inside large Ethernet networks later
- Now lets look at the larger structure of the IP Internet itself
  - From several vantage points

The firewalled Internet

- The Internet consists of *sites* interconnected by *ISPs*
  - Site = enterprise network, campus network, *your home!,* corporate network, etc.
  - ISP = Internet Service Provider
- The sites are protected by *firewalls*
- The sites often use a *private address space*, or *IP address realm*

The firewalled Internet



Firewalls

CS519

- Firewalls are a type of IP “router”
- They protect the site from unwanted packets (to an extent)
- Typically they allow “flows” to be initiated outgoing, but not incoming
  - Though they may prevent some types of outgoing flows
  - And allow some incoming flows to some hosts

Address realms and NAT

CS519

- Certain blocks of IP addresses have been designated “private addresses”
  - RFC 1918
  - 10/8, 172.16/12, and 192.168/16
- These can be used in any sites, but are not “visible” in the “global” address space
  - Like a one-way mirror: nodes in private networks can “see” the global internet, but nodes in the global internet cannot “see” hosts in private networks

What is this “slash” stuff?

- The “slash” notation (10/8, 72.16/12, etc.) denotes an address range
- P/B means a prefix P of length B bits
- 10/8 = 10.0.0.0 – 10.255.255.255
- 72.16/12 = 72.16.0.0 – 72.31.255.255

Address realms and NAT

CS519

- Two hosts in the same site cannot have the same private address, but two hosts in different sites can and do
- A private host can establish a flow with a public host (through a NAT box)
- A public host cannot generally establish a flow with a public host
- Two private hosts in different realms cannot generally establish flows with each other
  - Though we now know how to do this with the help of a global host

Address realms and NAT

CS519

- When a private host talks to a public host, the NAT box translates its private address into a public address
  - And remembers the private/public mapping
- Why do we want private addresses?
  - This is a low-cost way to effectively increase the IP address space to way beyond 32 bits
- We'll examine this in detail later

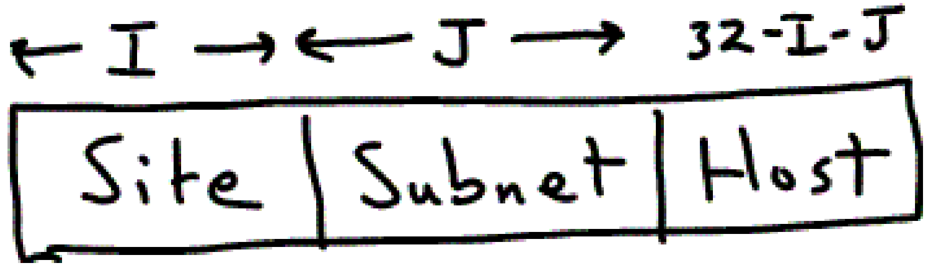
Routing from a site router's point of view

CS519

- We saw how a host checks to see if the address prefix is "my subnet", and if not forwards the packet to a default router
- A router in a site (kindof) checks to see if the address prefix is "my site", and if not forwards the packet by default towards the global internet

Does this mean the IP address really looks like this?

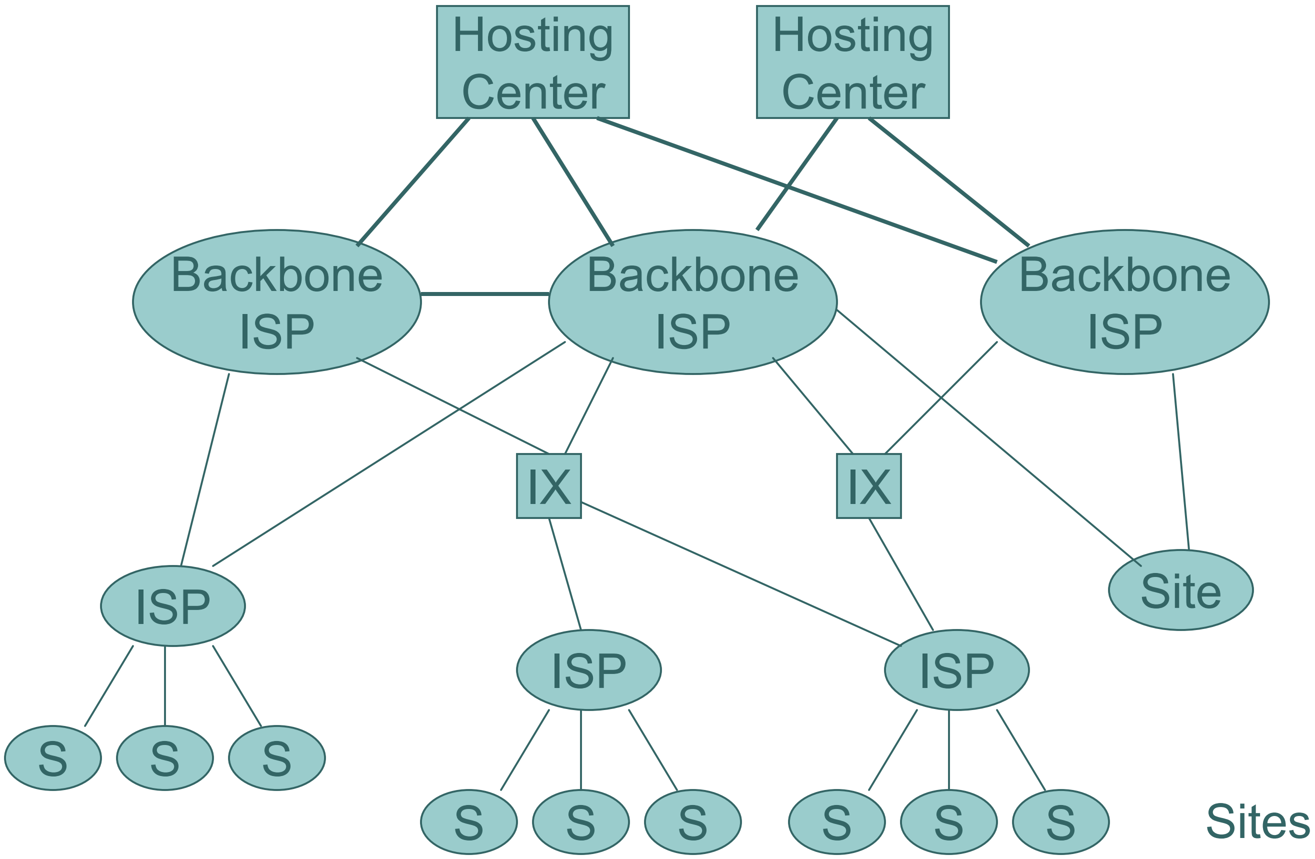
CS519



- Yes, to a site router, in the sense that this is what the site router has to “know” to correctly forward the packet
- But, no, in a global sense this is still an incomplete picture of the address
  - If this was the complete picture, it means that every global router would need to know explicitly of every site!

The global Internet

CS519

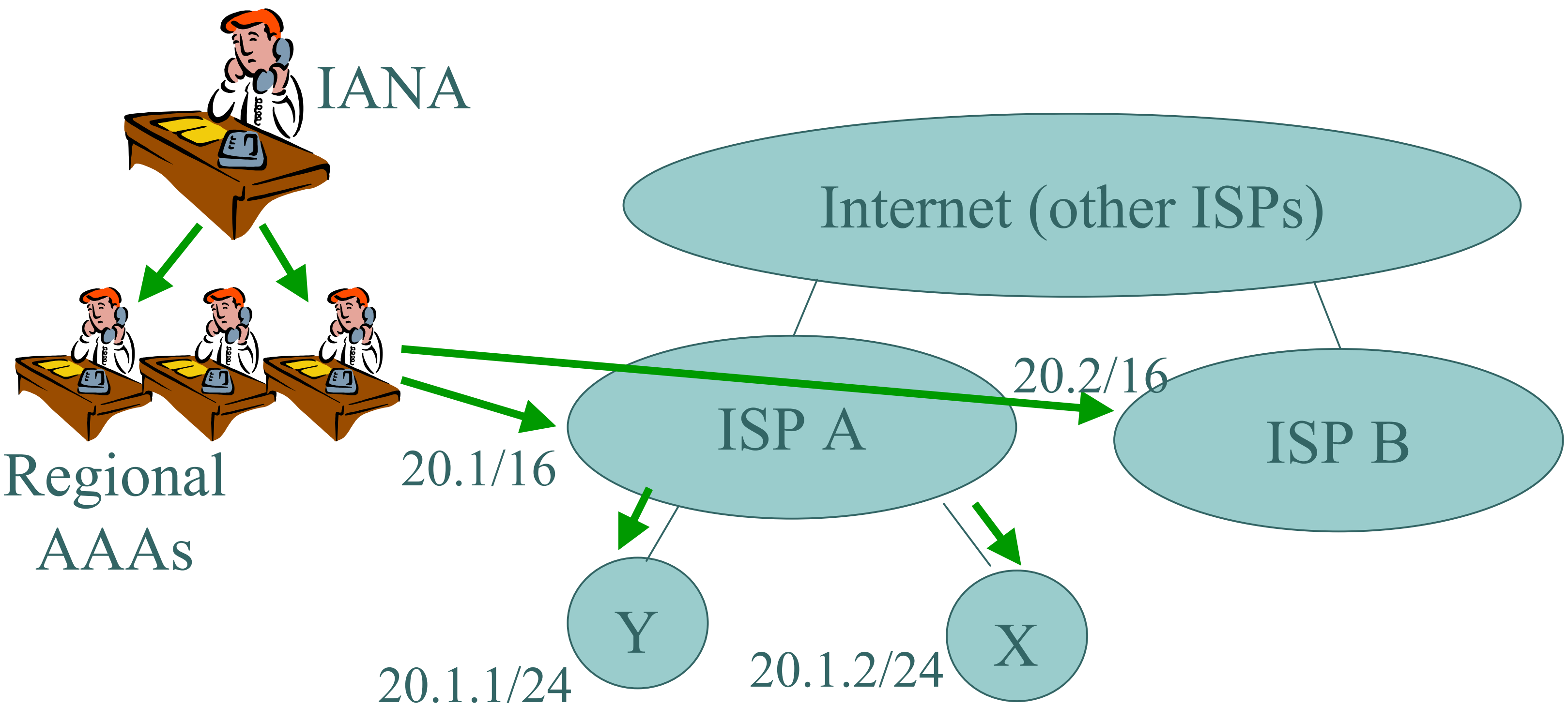


IXs came first

IXs tend to be performance bottlenecks

Hosting centers and bilateral peering are a response to poor IXs

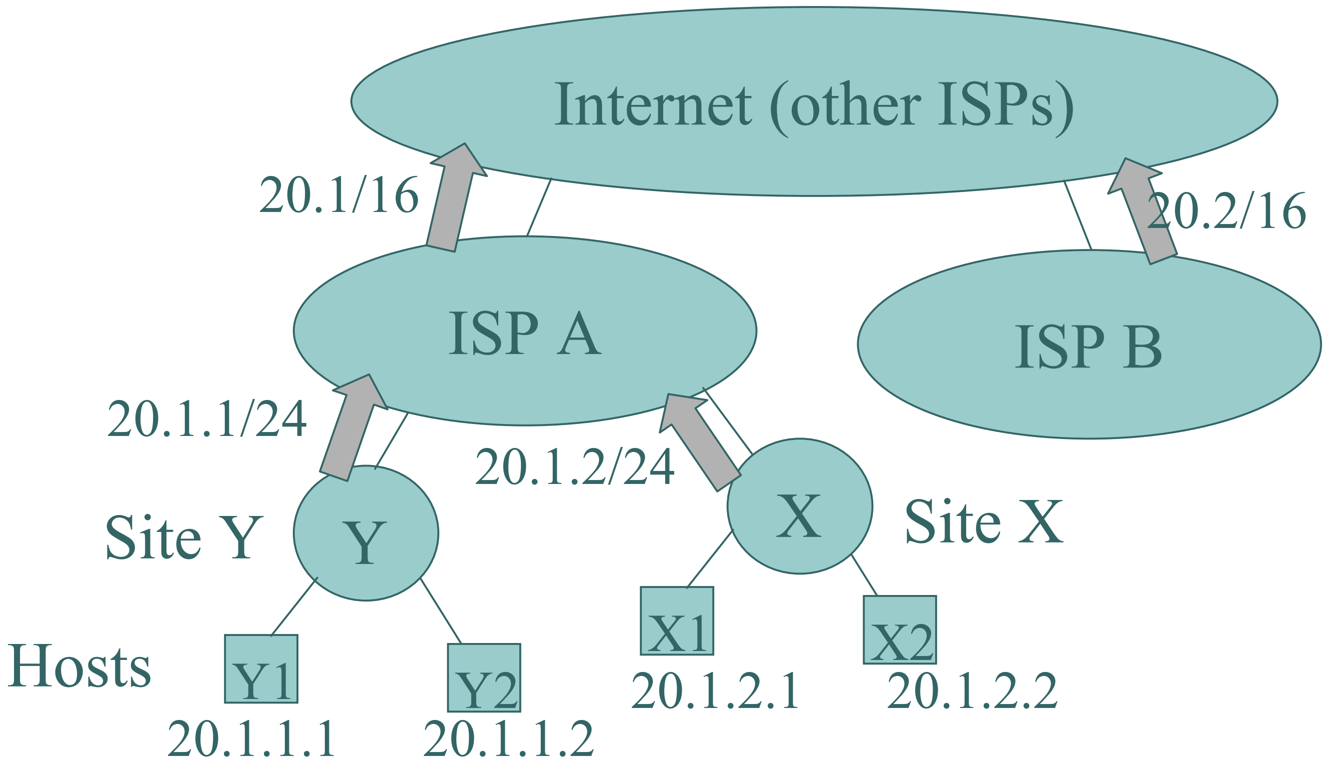
Address assignment



Route Aggregation Basics

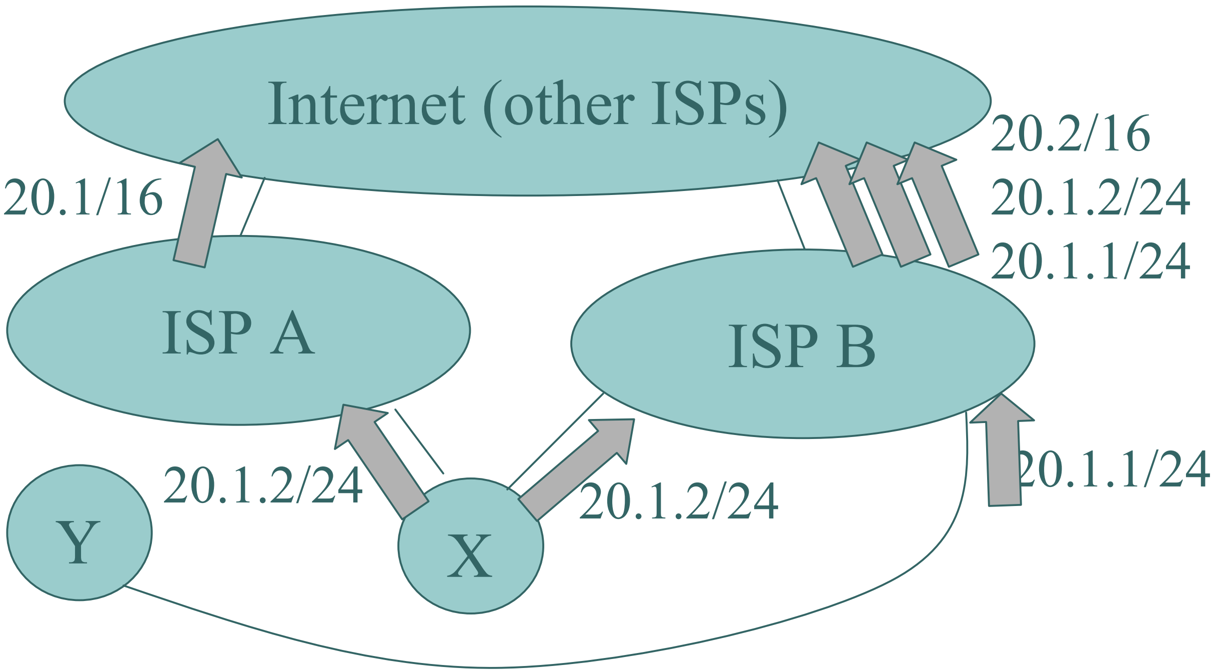
CS519

- Address hierarchy ⇔ topological hierarchy

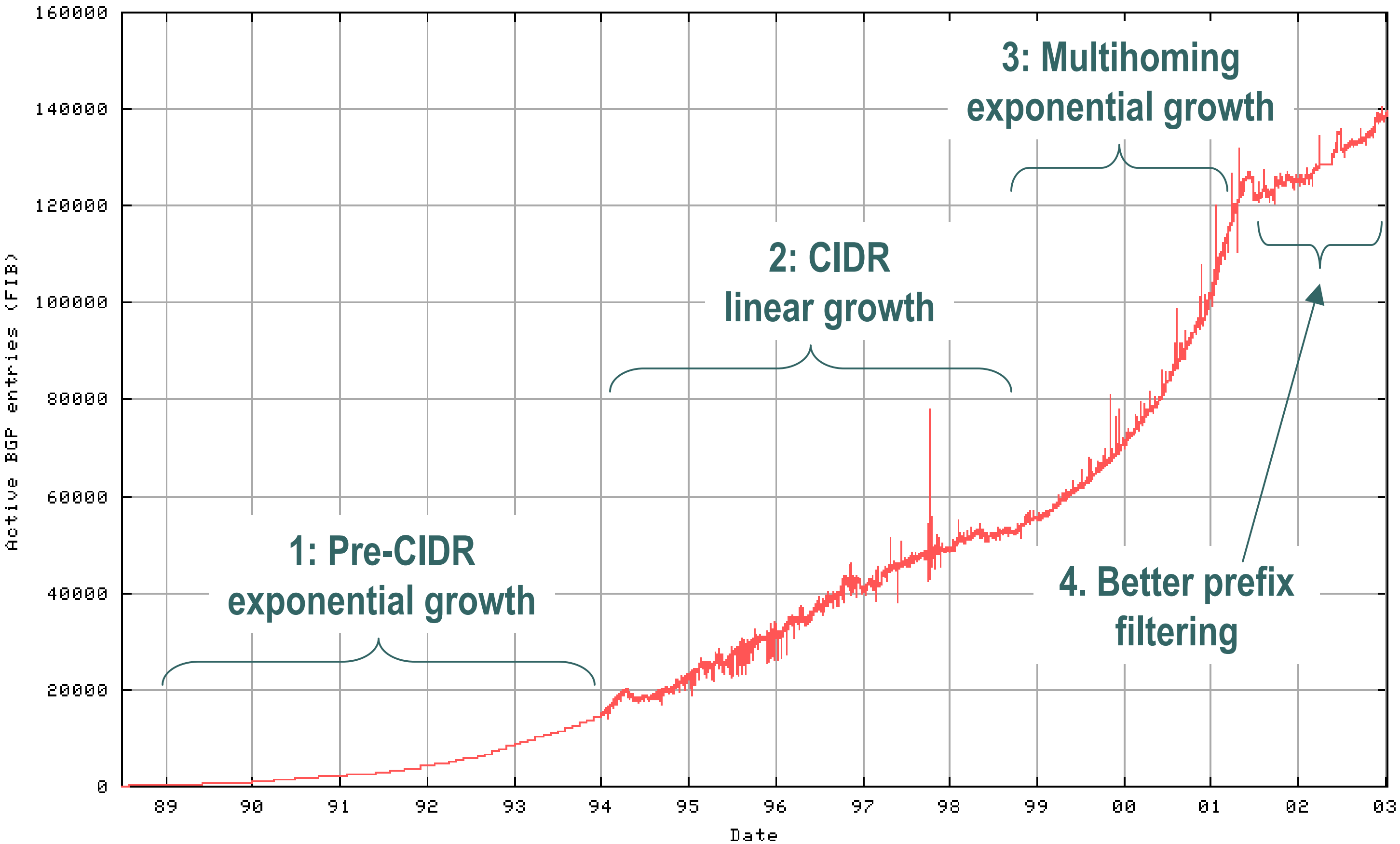


But we don’t always get good aggregation in the Internet

CS519



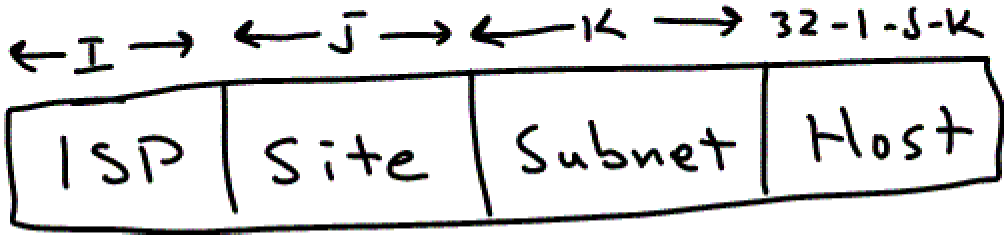
Size of the backbone router forwarding tables (BGP)



Source: The CIDR Report, www.cidr-report.org

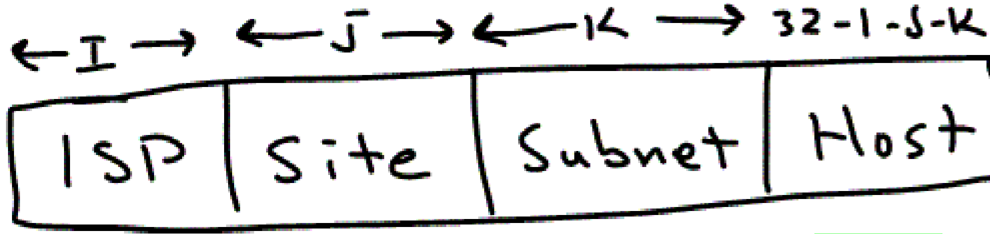
The "complete" structure of the IP address

CS519



- But this is a misleading picture
  - Which is why I didn't show it at the start, and which is why you rarely see this picture

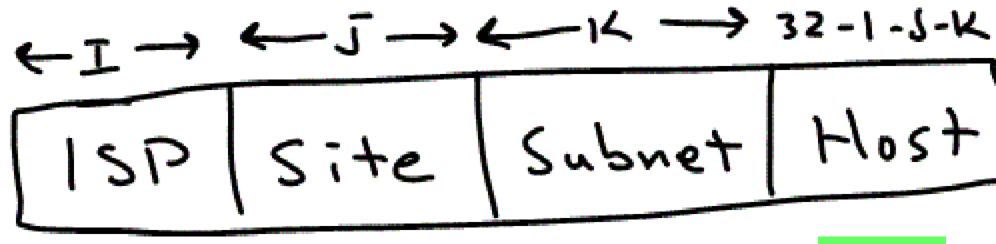
Why misleading?



CS519

- Bit-wise field boundaries aren't fixed
- Hosts and last-hop/site routers don't "see" the higher-level structure
  - As long as they have a default route
- Global routers don't "see" the lower-level structure
  - In fact, they don't "see" any structure except prefix/suffix
  - But sometimes the prefix is pretty long (i.e. into the "site" level)

Why misleading?



- Some sites didn't in fact get their prefix from an ISP
  - And multihomed sites advertise their site via the "wrong" ISP
- Ultimately, only humans are aware of the whole structure…but routers and hosts don't care what we think!

On the other hand…

- IPv6 does draw these kinds of complete address structure pictures
  - Even though IPv6 nodes don't understand the structure
- The difference is that IPv4 addresses evolved over time bottom-up, whereas IPv6 addresses were defined top-down from the start