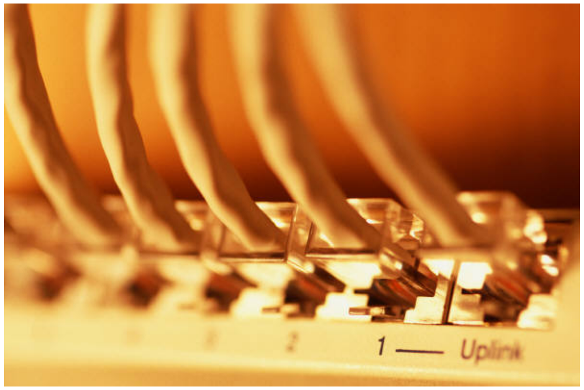
Lecture 8: Internetworking

CSE 123: Computer Networks
Stefan Savage

Last Time

- How to interconnect LANs



- Repeaters/Hubs
- Bridges
- Learning Bridges
- Spanning trees
- Switches

Today

- Recall problems with L2 bridging/switching from yesterday
  - **Homogeneous link layer (all Ethernet)**
  - Broadcast traffic to all members (scaling…VLANS helped a bit)
  - No control over topology (and root is bottleneck)
  - Single administrative domain (who controls?)

- Goal: Scalably interconnect large numbers of networks of different types

First some history…

- 1968: DARPAnet/ARPAnet (precursor to Internet)
  - Advanced Research Projects Agency Network
  - Bob Taylor, Larry Roberts create program to build first wide-area packet-switched network
- 1978: new networks emerge
  - SATNet, Packet Radio, Ethernet
  - All “islands” to themselves – didn’t work together
- Big question: how to connect these networks?

Note: If you want to learn more about Internet history, read “Where Wizards Stay Up Late” by Hafner and Lyon

DARPAnet/Internet
Primary Goal: Connect Stuff

- "Effective technique for multiplexed utilization of existing interconnected networks" – David Clark
  - **Minimal** assumptions about underlying networks
    - » No support for broadcast, multicast, real-time, reliability
    - » Extra support could actually get in the way
  - Packet switched, store and forward
    - » Matched application needs, nets already packet switched
    - » Enables **efficient resource sharing**/high utilization
  - "Gateways" interconnect networks
    - » Routers in today's nomenclature

What is it hard to inter-connect networks?

- Main challenge is heterogeneity of link layers:
  - Addressing
    - Each network media has a different addressing scheme
  - Bandwidth
    - Modems to terabits
  - Latency
    - Seconds to nanoseconds
  - Frame size (Maximum Transmission Unit – MTU)
    - Dozens to thousands of bytes
  - Loss rates
    - Differ by many orders of magnitude
  - Service guarantees
    - Send and pray vs reserved bandwidth

Lecture 8 Overview

- Internet Protocol
  - Service model
  - Packet format
- Fragmentation
- Addressing
  - Subnetting
  - CIDR

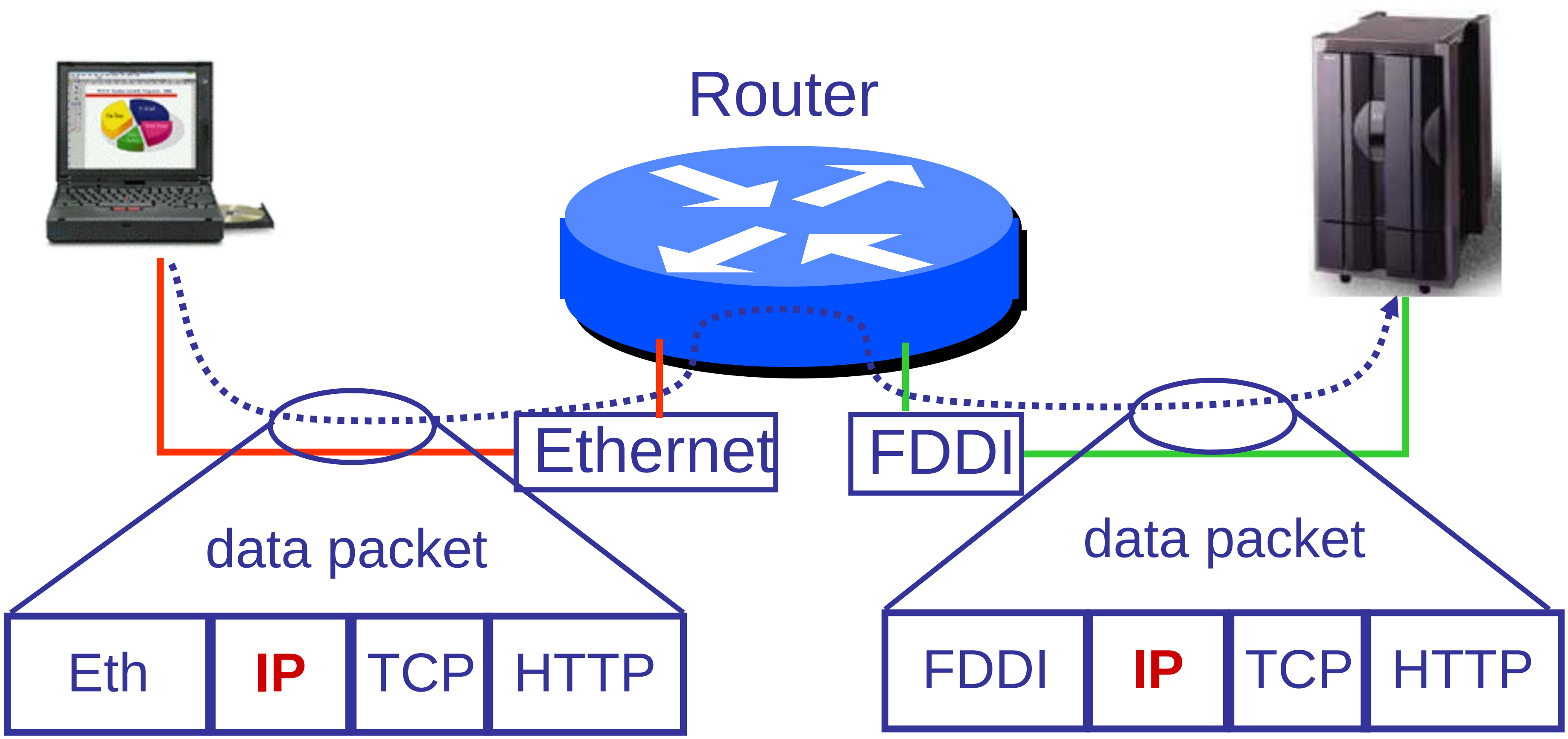
Internetworking

- Cerf & Kahn74,
  "*A Protocol for Packet Network Intercommunication*"
  - Foundation for the modern Internet

- Routers forward packets from source to destination
  - May cross many separate networks along the way

- All packets use a **common** Internet Protocol
  - *Any* underlying data link protocol
  - *Any* higher layer transport protocol

IP Networking



Routers

- A router is a store-and-forward device
  - Routers are connected to multiple networks
  - On each network, looks just like another host
  - A lot like a bridge/switch, except at the **network** layer
- Must be explicitly addressed (L2) by incoming frames
  - Not at all like a switch, which is transparent
  - Removes link-layer header, parses IP header
- Looks up next hop, forwards on appropriate network
  - Each router need only get one step closer to destination

IP Philosophy

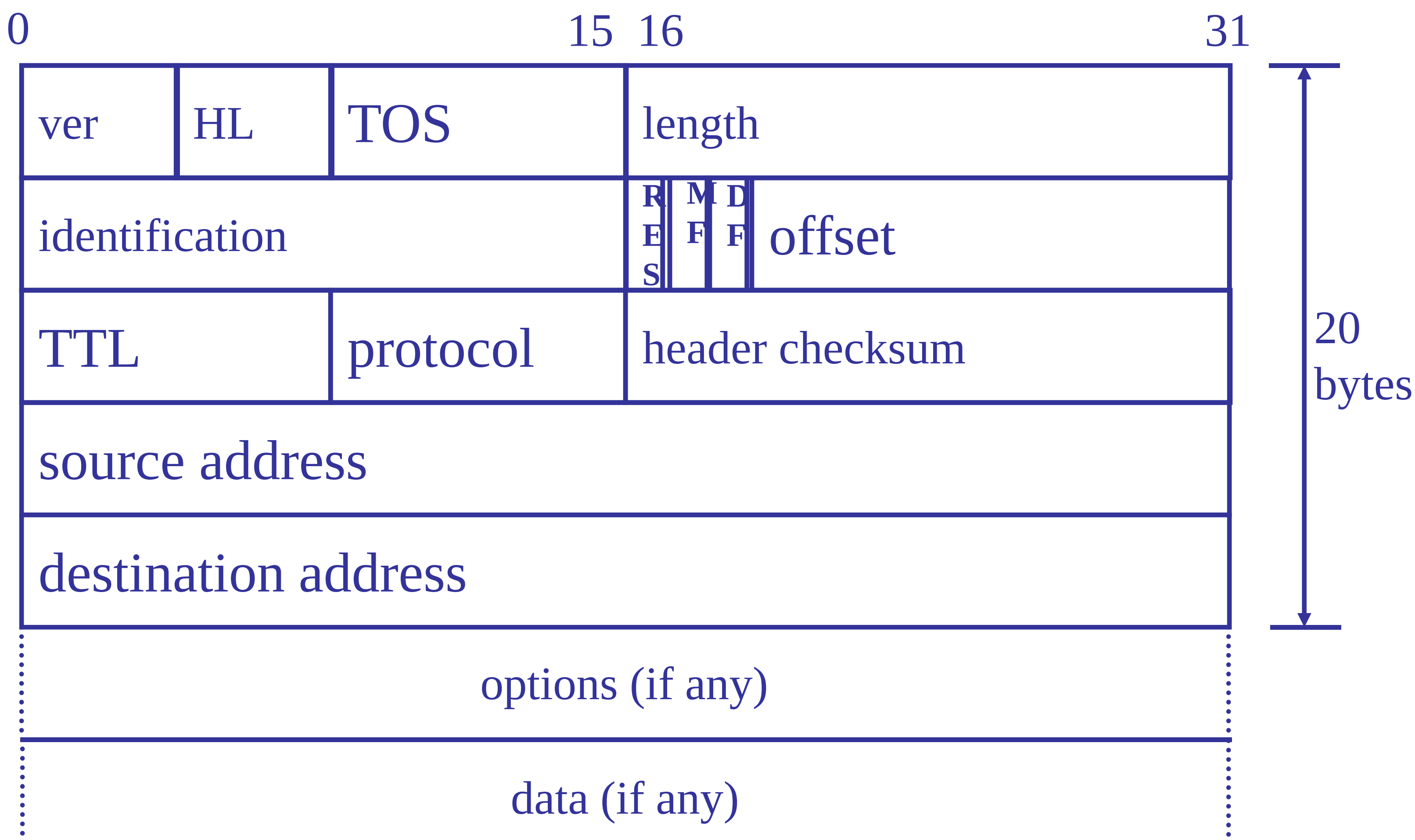
- Impose few demands on network
  - Make few assumptions about what network can do
  - No QoS, no reliability, no ordering, no large packets
  - No persistent state about communications; no connections
- Manage heterogeneity at hosts (not in network)
  - Adapt to underlying network heterogeneity
  - Re-order packets, detect errors, retransmit lost messages…
  - Persistent network state only kept in hosts (fate-sharing)
- Service model: best effort, a.k.a. *send and pray*

So what *does* IP do?

- Addressing
- Fragmentation
  - E.g. FDDI's maximum packet is 4500 bytes while Ethernet is 1500 bytes, how to manage this?
- *Some* error detection

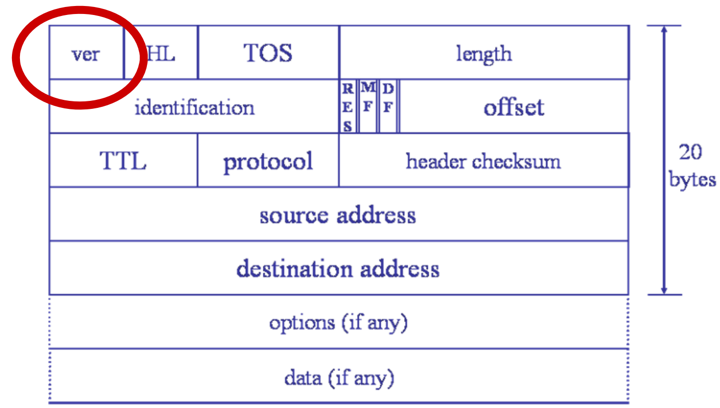
- Routers only forward packets to next hop
  - They do not:
    - » Detect packet loss, packet duplication
    - » Reassemble or retransmit packets

IP Packet Header



Version field

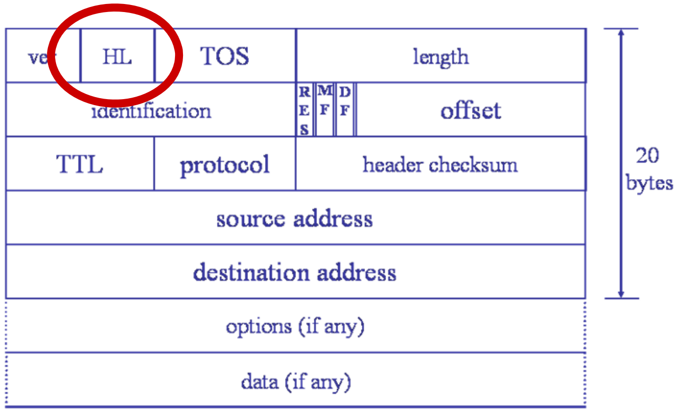
- Which version of IP is this?
  - Plan for change
  - Very important!



- Current versions
  - 4: most of Internet today
  - 6: new protocol with larger addresses
  - What happened to 5? Standards body politics.

Header length

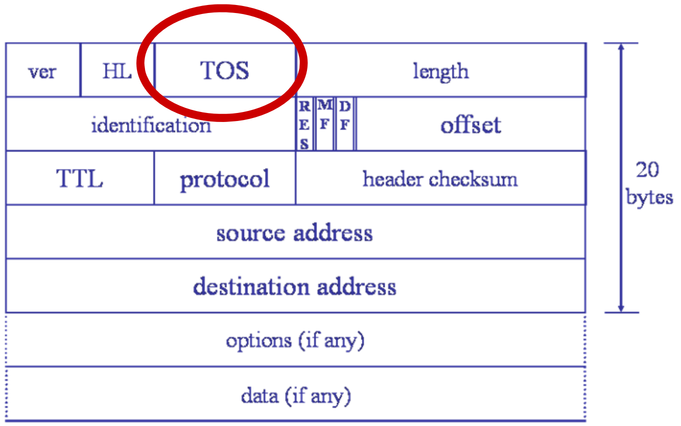
- How big is IP header?
  - In bytes/octets
  - Variable length
    - Options
  - Engineering consequences of variable length…



- Most IP packets are 20 bytes long

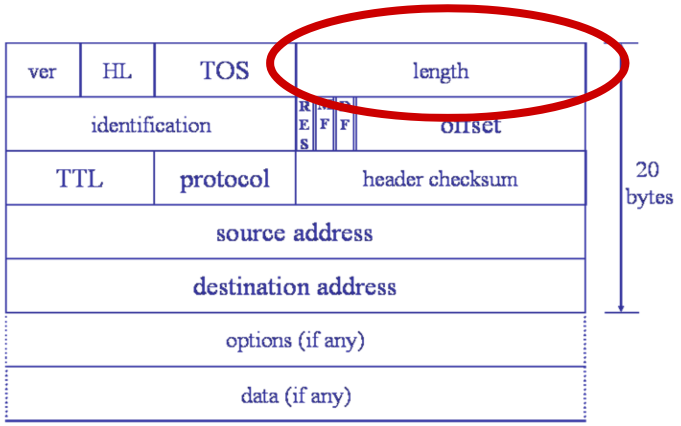
Type-of-Service

- How should this packet be treated?
  - Care/don't care for delay, throughput, reliability, cost
  - How to interpret, how to apply on underlying net?
  - Largely unused until 2000 (hijacked for new purposes, ECN & Diffserv)



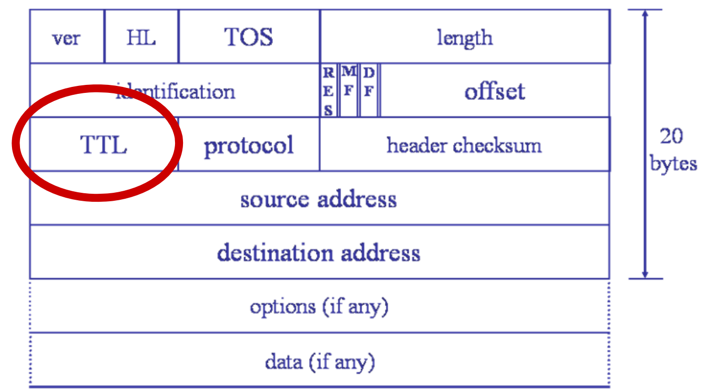
Length

- How long is whole packet in bytes/octets?
  - Includes header
  - Limits total packet to 64K
  - Redundant?



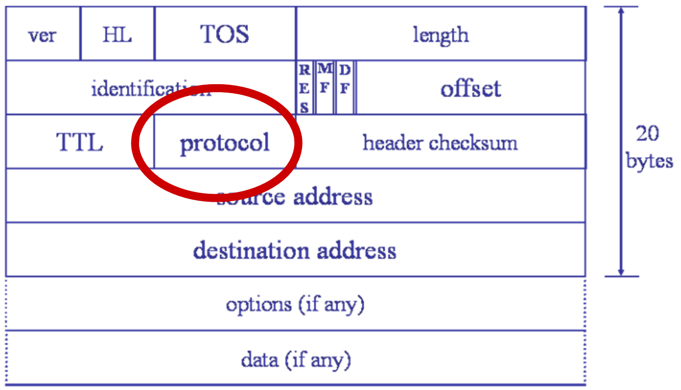
TTL (Time-to-Live)

- How many more routers can this packet pass through?
  - Designed to limit packet from looping forever
- Each router decrements TTL field
- If TTL is 0 then router discards packet



Protocol

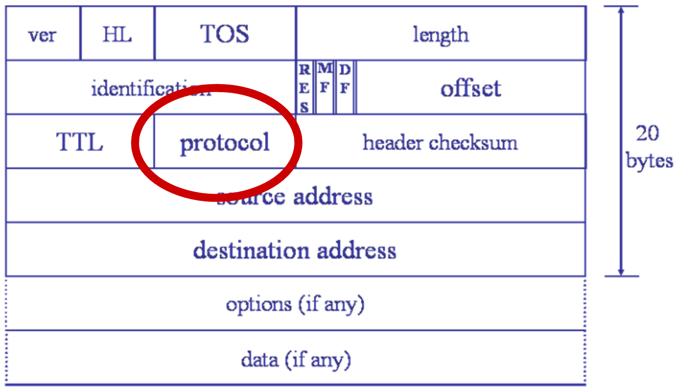
- Which transport protocol is the data using?
  - i.e. how should a host interpret the data



- TCP = 6
- UDP = 17

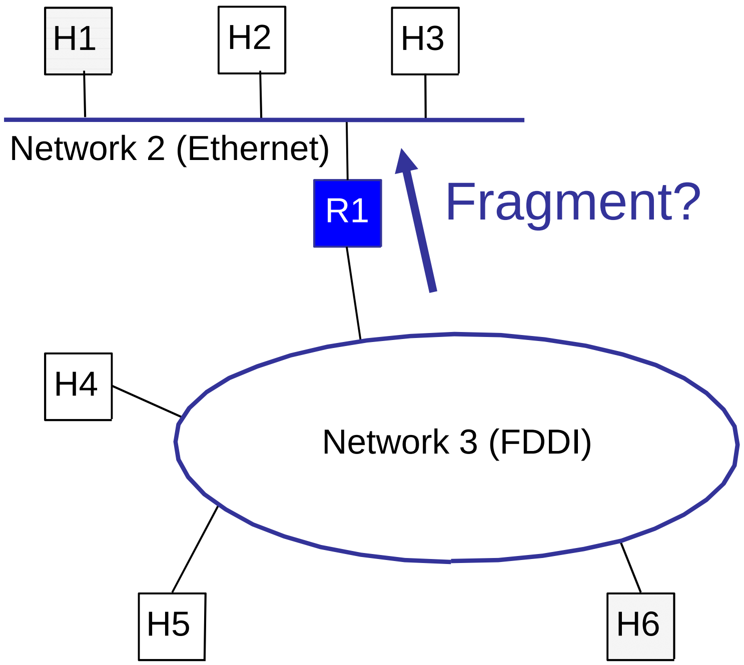
IP Checksum

- Header contains simple checksum
  - Validates content of header *only*
- Recalculated at each hop
  - Routers need to update TTL
  - Hence straightforward to modify
- Ensures *correct* destination receives packet



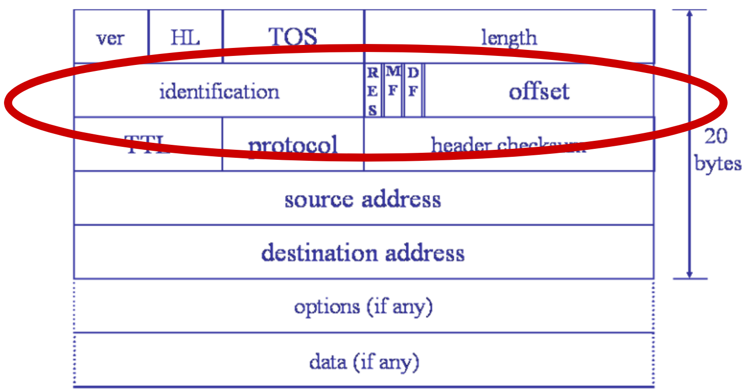
Fragmentation

- Different networks may have different frame limits (MTUs)
  - Ethernet 1.5K, FDDI 4.5K
- 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



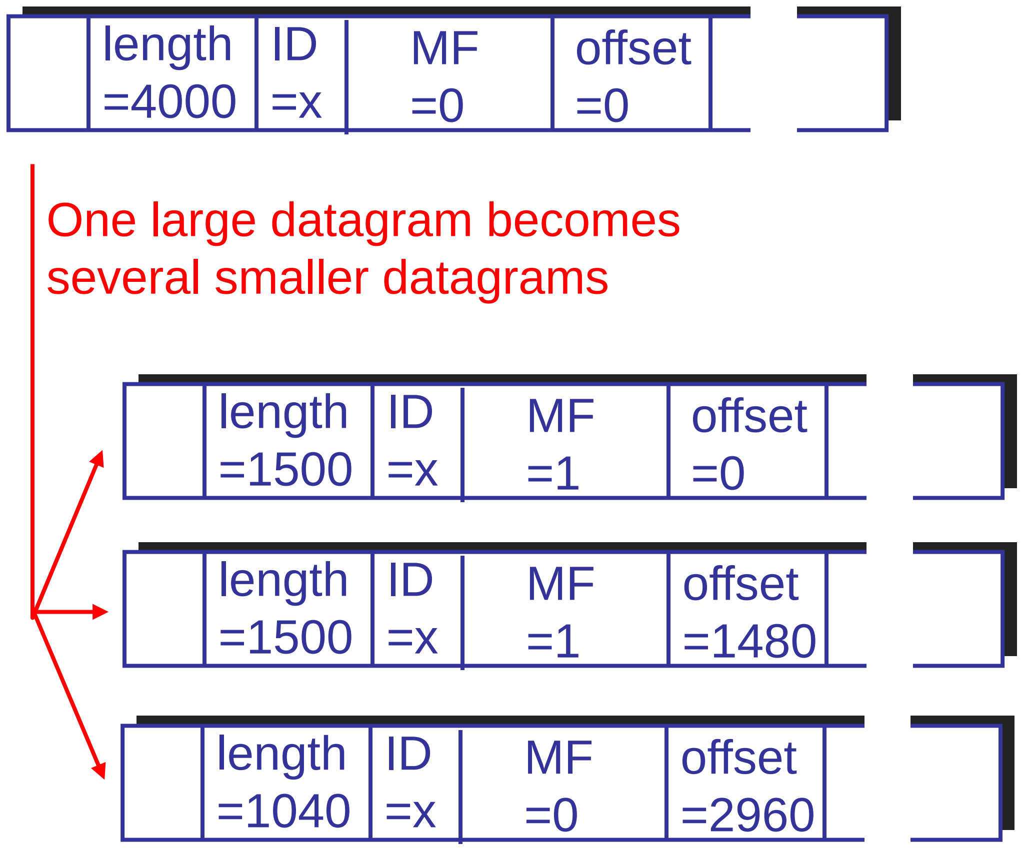
IP ID and Bitflags

- Source inserts unique value in identification field
  - Also known as the IPID
  - Value is copied into any fragments
- Offset field indicates position of current fragment (in bytes)
  - 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



Problems w/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 mean time
  - When can it be garbage collected?
- Why not reassemble at each router?

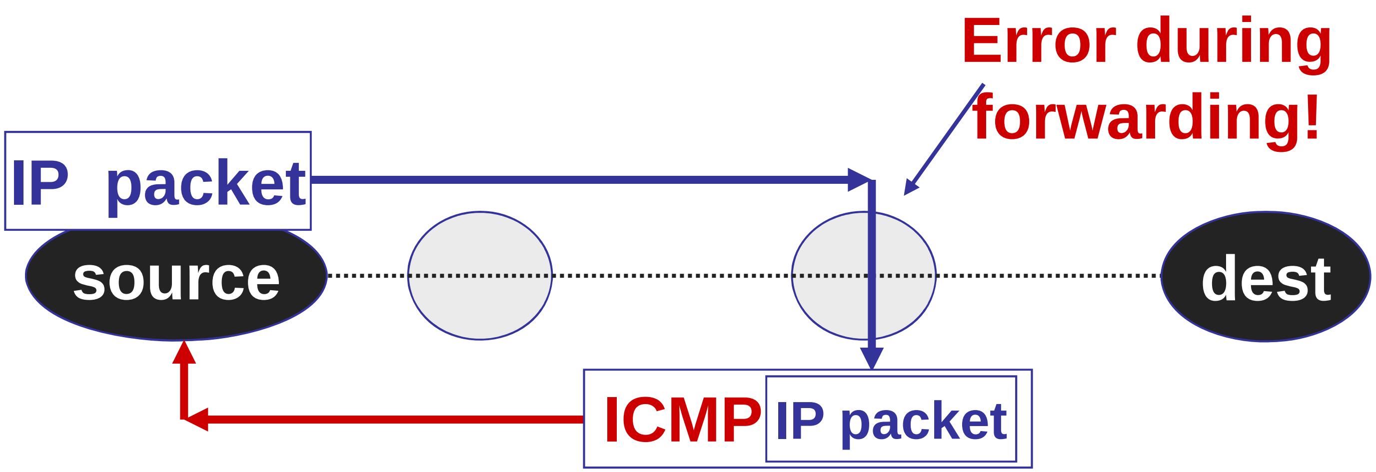
Solution: Path MTU Discovery

- Path MTU is the smallest MTU along path
  - Observation: 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; causes router to send error
    - ICMP protocol: special IP packet format for sending error msgs
  - Hosts discover limits, can size packets at source
  - Reassembly at destination as before

Aside: ICMP

- What happens when things go wrong?
  - Need a way to test/debug a large, widely distributed system
- ICMP = Internet Control Message Protocol (RFC792)
  - Companion to IP – required functionality
- Used for error and information reporting:
  - Errors that occur during IP forwarding
  - Queries about the status of the network

ICMP Error Message Generation



Common ICMP Messages

- Destination unreachable
  - "Destination" can be host, network, port, or protocol
- Redirect
  - To shortcut circuitous routing
- TTL Expired
  - Used by the "traceroute" program
    - » traceroute traces packet routes through Internet
- Echo request/reply
  - Used by the "ping" program
    - » ping just tests for host liveness
- ICMP messages include portion of IP packet that triggered the error (if applicable) in their payload

ICMP Restrictions

- The generation of error messages is limited to avoid cascades … error causes error that causes error…

- Don’t generate ICMP error in response to:
  - An ICMP error
  - Broadcast/multicast messages (link or IP level)
  - IP header that is corrupt or has bogus source address
  - Fragments, except the first

- ICMP messages are often rate-limited too
  - Don’t waste valuable bandwidth sending tons of ICMP messages

Addressing Considerations

- Fixed length or variable length addresses?

- Issues:
  - Flexibility
  - Processing costs
  - Header size

- Engineering choice: IP uses fixed length addresses

Addressing Considerations (2)

- Hierarchical vs flat
  - How much does each router need to know?
- Original DARPAnet IP addressing (24 bits)
  - Global inter-network address (8 bits)
  - Local network-specific address (16 bits)

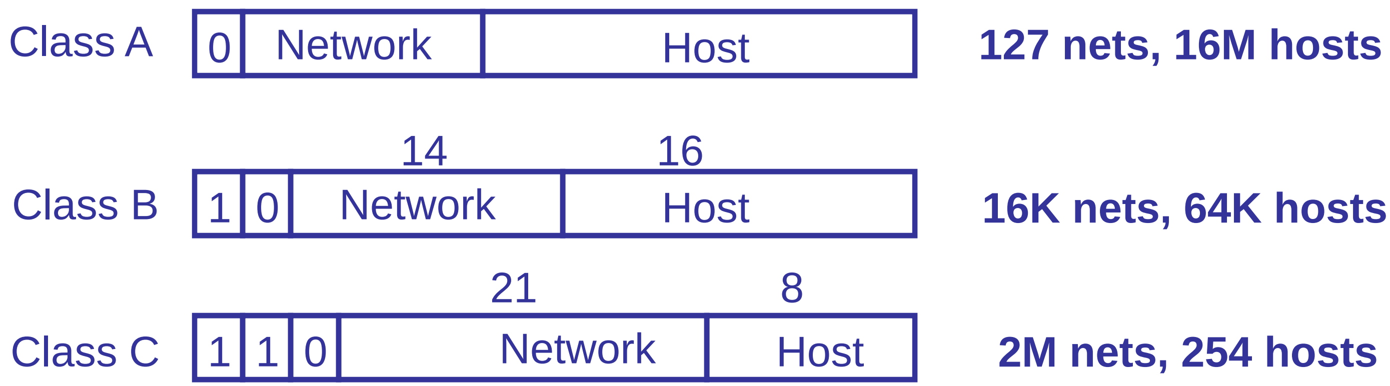
- Very successful, but now obsolete… what assumption do you think was problematic?

Modern IP Addresses

- 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
- Which part is network *vs.* host?

Class-based Addressing

- Most significant bits determines “class” of address



- 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

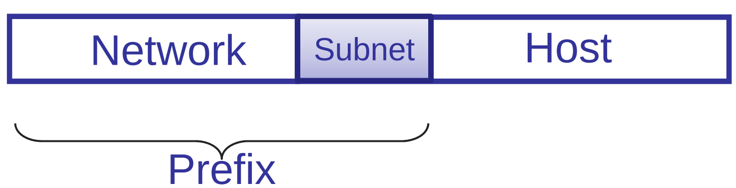
IP Forwarding Tables

- Router needs to know where to forward a packet
- Forwarding table contains:
  - List of network names and next hop routers
  - Local networks have entries specifying which interface
    - Link-local hosts can be delivered with Layer-2 forwarding
- E.g. www.ucsd.edu address is 132.239.180.101
  - Class B address – class + network is 132.239
  - Lookup 132.239 in forwarding table
  - Prefix – part of address that really matters for routing

Subnetting (inside a network)

- Individual networks may be composed of several LANs
  - Only want traffic destined to local hosts on physical network
  - Routers need a way to know which hosts on which LAN

- 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



Subnet Addresses

- Every (sub)network has an address and a netmask
  - Netmask tells which bits of the network address is important
  - Convention suggests it be a proper prefix
- 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 LAN
  - 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

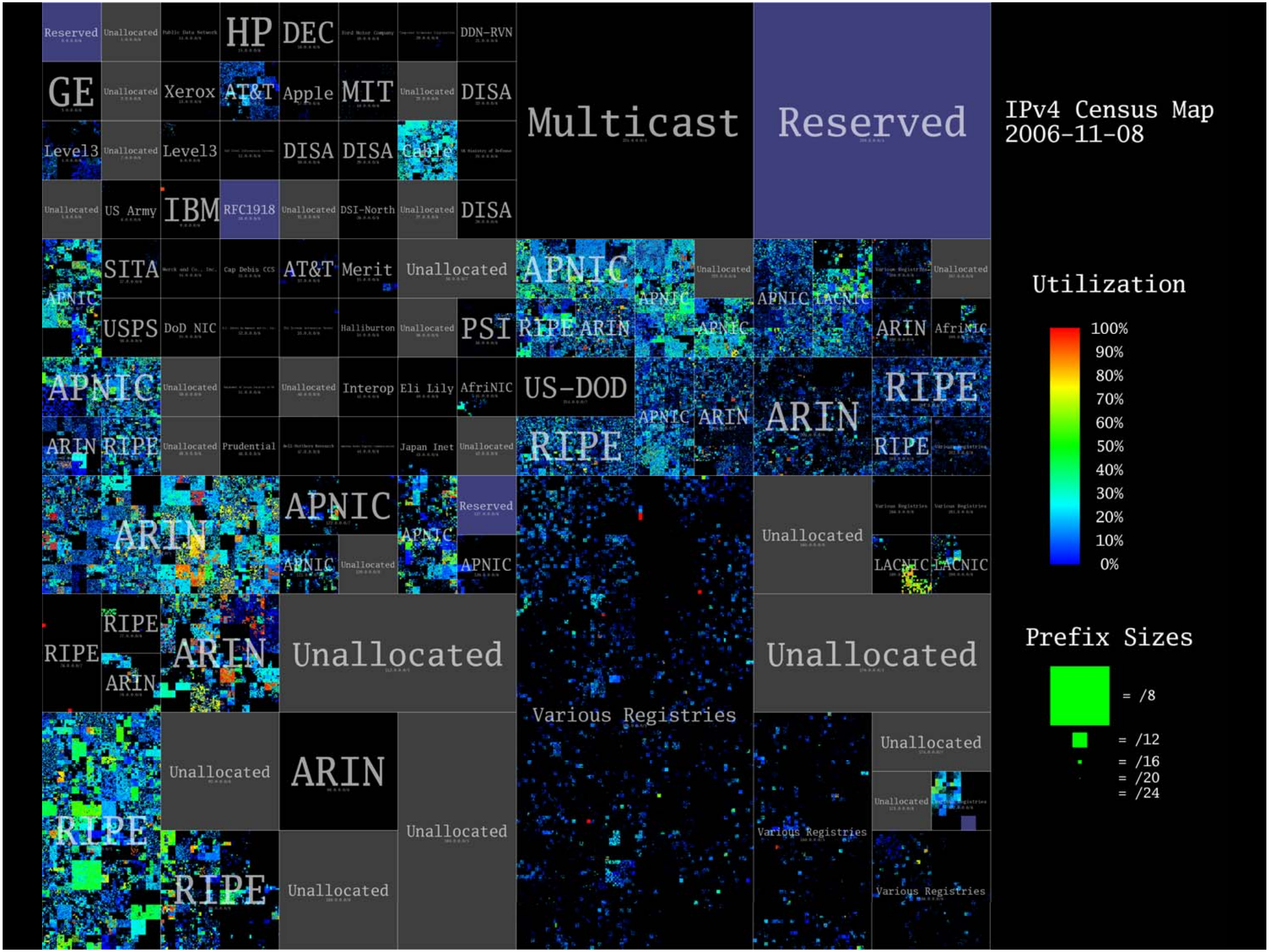
- Why?
  - Class C too small for most organizations (only ~250 addresses)
  - Very few class A – very careful about giving them out (who has 16M hosts anyway?)
  - Class B – greatest problem

CIDR

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

  - e.g. 10.95.1.2/8: 10 is network and remainder (95.1.2) is host

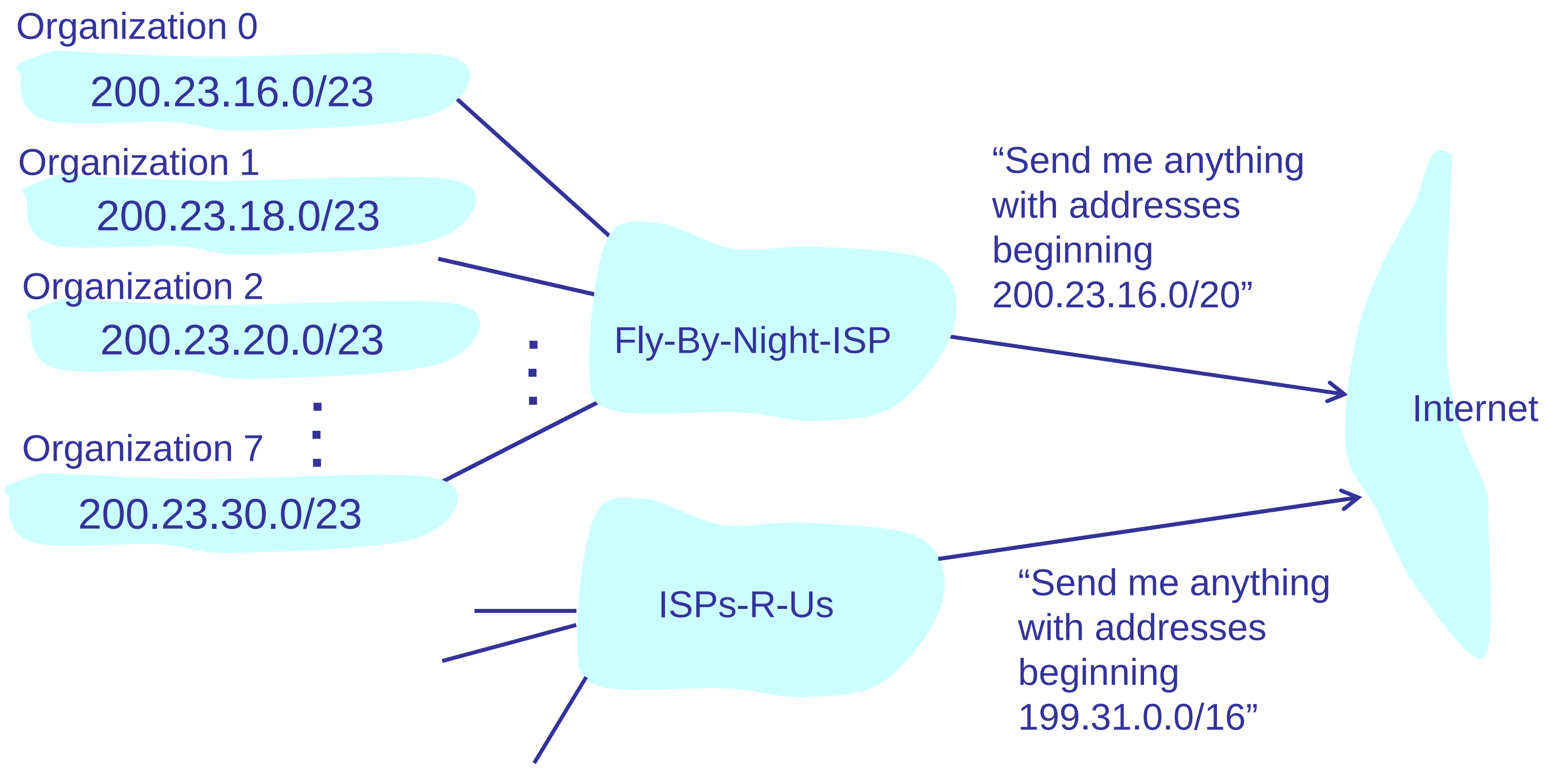
- Pro: Finer grained allocation; aggregation
- Con: More expensive lookup: longest prefix match



IPv4 Census Map
2006-11-08
Reserved
Unallocated
HP
DEC
DDN-RVN
GE
Xerox
AT&T
Apple
MIT
DISA
Multicast
Reserved
Level3
Level3
DISA
DISA
Cable
US Army
IBM
RFC1918
DSI-North
DISA
APNIC
SITA
Cap Debis CCS
AT&T
Merit
Unallocated
APNIC
USPS
DoD NIC
Halliburton
PSI
RIPE
ARIN
LACNIC
AfriNIC
APNIC
Interop
Eli Lily
AfriNIC
US-DOD
RIPE
ARIN
Prudential
Japan Inet
ARIN
APNIC
Reserved
Unallocated
Various Registries
Unallocated
LACNIC
RIPE
ARIN
Unallocated
Unallocated
ARIN
Various Registries
Unallocated
Utilization
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Prefix Sizes
= /8
= /12
= /16
= /20
= /24

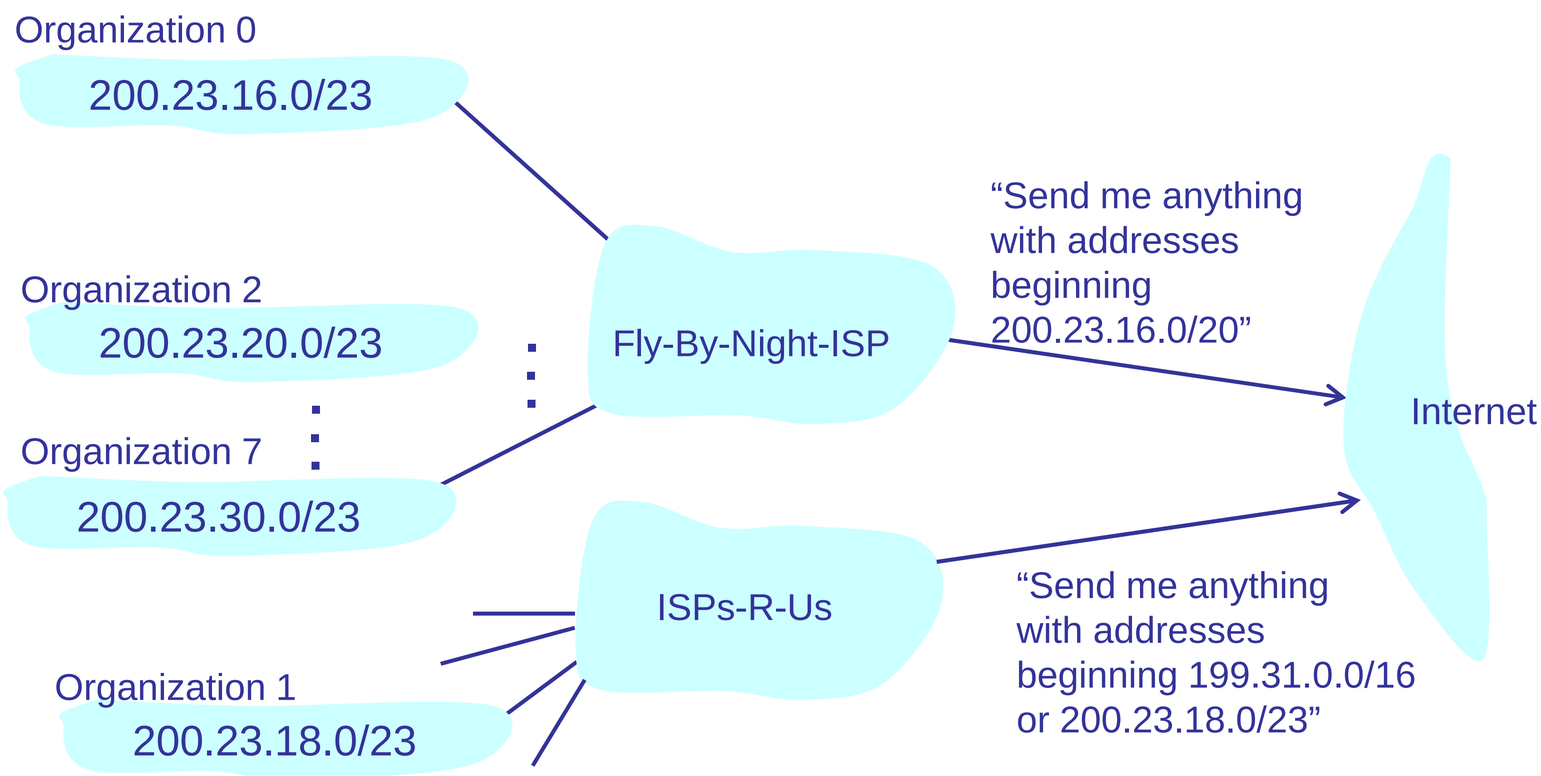
Route Aggregation

- Combine adjacent networks in forwarding tables
  - Helps keep forwarding table size down



Most Specific Route

- But what if address range is not contiguous?



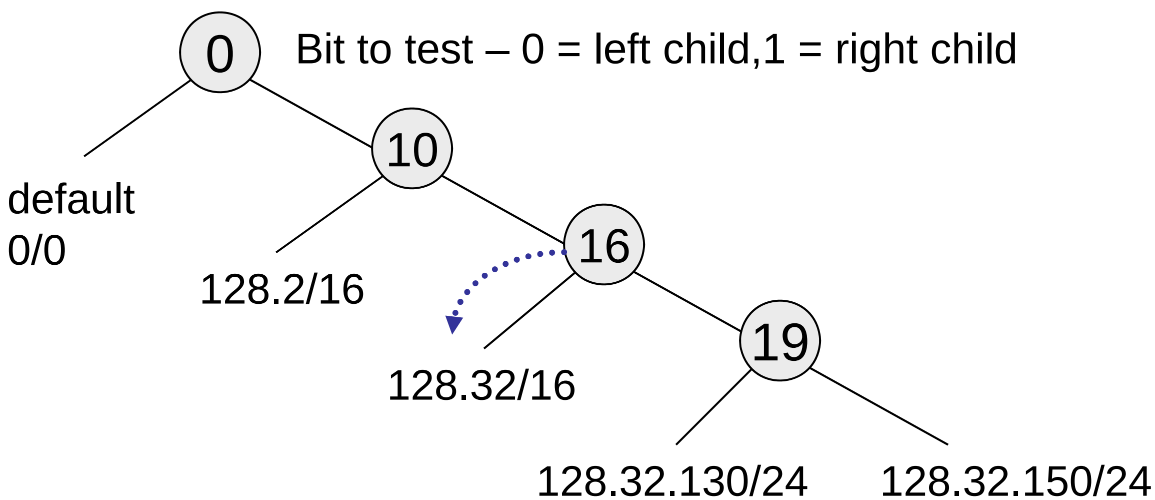
Longest Matching Prefix

- Forwarding table contains many prefix/length tuples
  - Again, they *need not* be disjoint!
  - E.g. 200.23.16.0/20 and 200.23.18.0/23
  - 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
    - Hardware solutions: Content Addressable Memories
    - Software solutions: clever optimized data structures
  - Our own George Varghese is the master of this domain

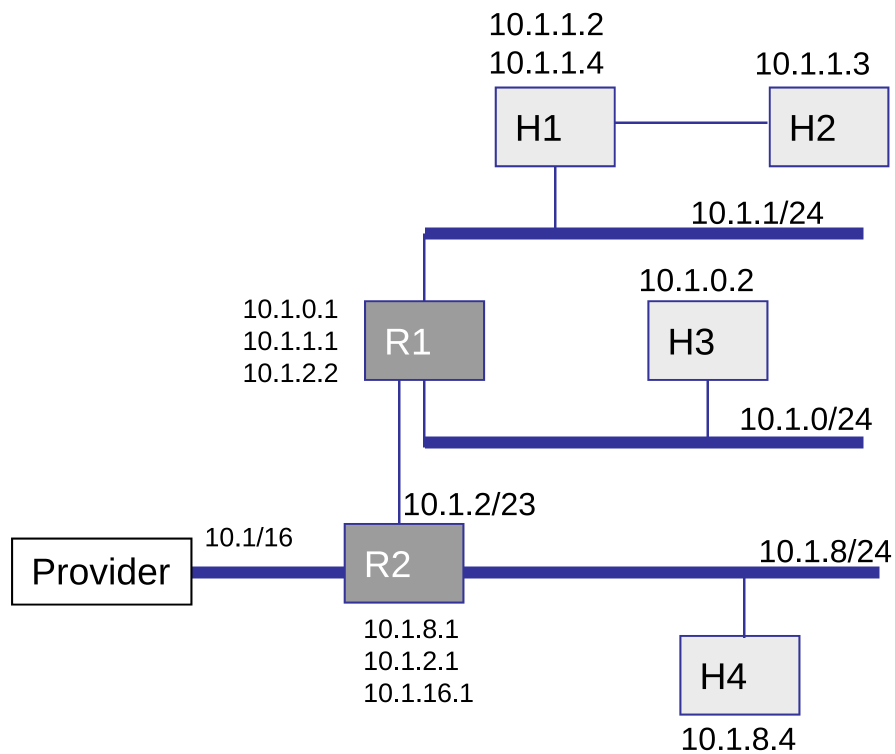
Simplest approach: PATRICIA Trie

- Straightforward way to look up LMP
  - Arrange route entries into a series of bit tests
  - Worst case = 32 bit tests
  - Problem: memory speed is a bottleneck



Forwarding example

- Packet to 10.1.1.3 arrives
- Path is R2 – R1 – H1 – H2

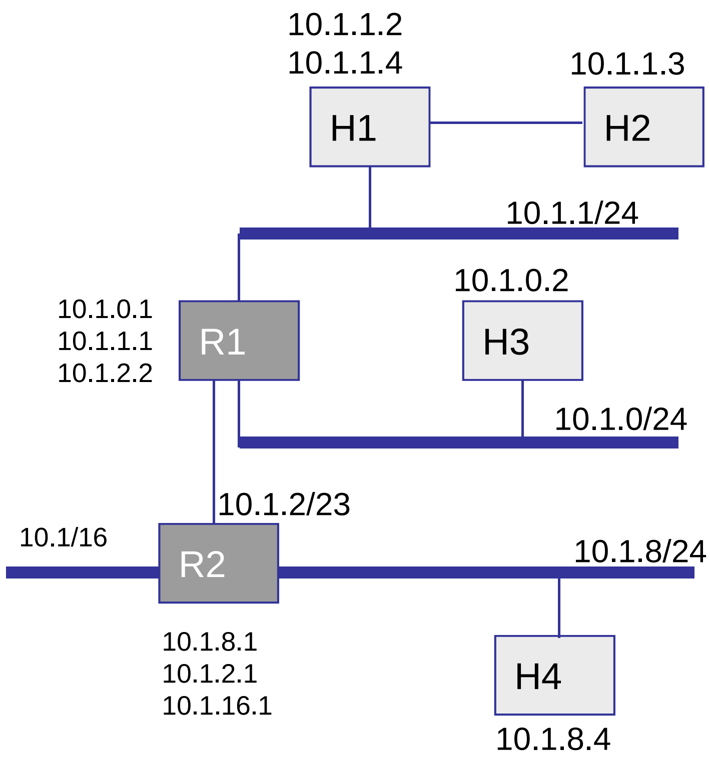


Forwarding example (2)

- Packet to 10.1.1.3
- Matches 10.1.0.0/23

Forwarding table at R2

Destination	Next Hop
127.0.0.1	loopback
Default or 0/0	10.1.0.1
10.1.8.0/24	interface1
10.1.2.0/23	interface2
10.1.0.0/23	10.1.2.2

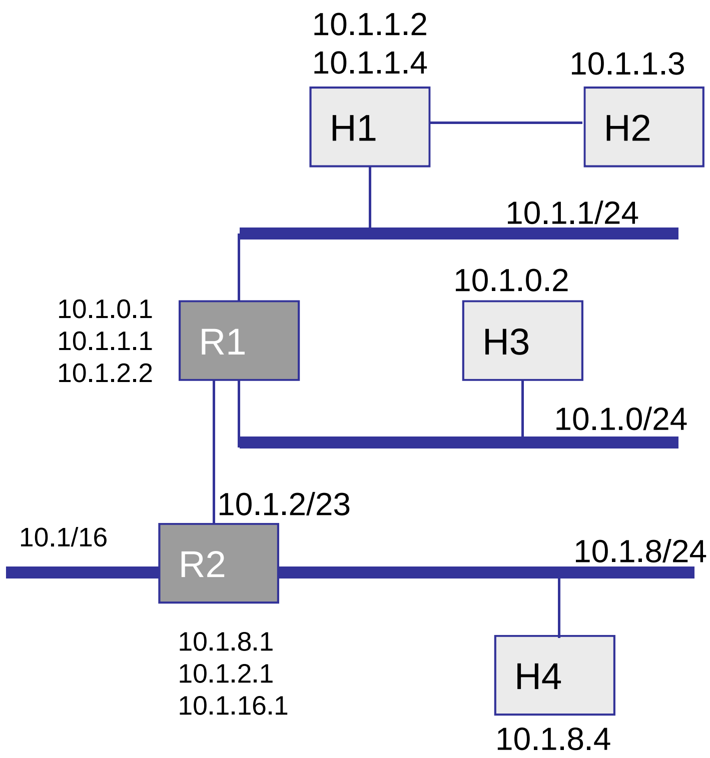


Forwarding example (3)

- Packet to 10.1.1.3
- Matches 10.1.1.2/31
  - Longest prefix match

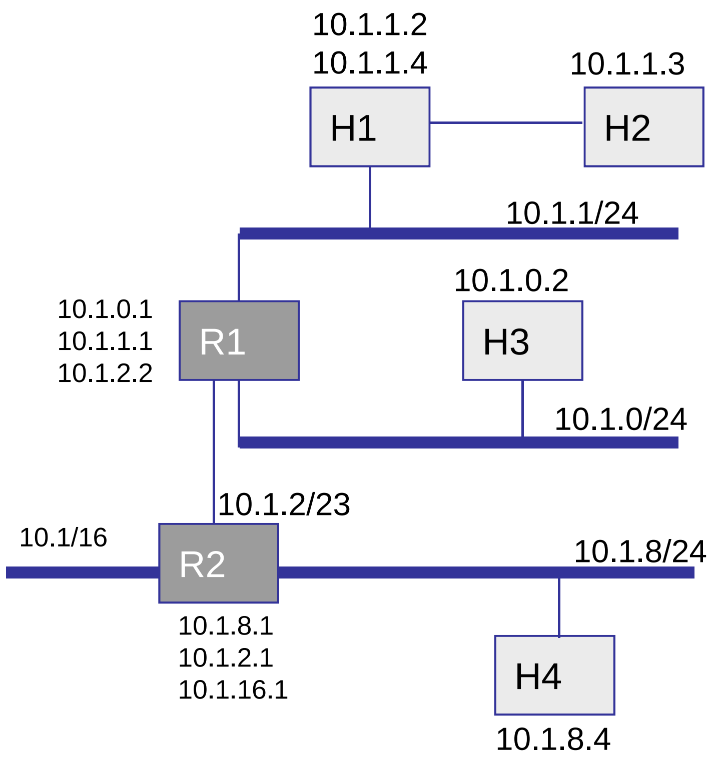
Routing table at R1

Destination	Next Hop
127.0.0.1	loopback
Default or 0/0	10.1.2.1
10.1.0.0/24	interface1
10.1.1.0/24	interface2
10.1.2.0/23	interface3
10.1.1.2/31	10.1.1.2



Forwarding example (4)

- Packet to 10.1.1.3
- Direct route
  - Longest prefix match



Routing table at H1

Destination	Next Hop
127.0.0.1	loopback
Default or 0/0	10.1.1.1
10.1.1.0/24	interface1
10.1.1.3/31	interface2

Remaining addressing issues

- How do you get IP addresses?
  - Registries and DHCP

- How do IP addresses get mapped to link layer addresses (e.g., Ethernet)?
  - ARP

Whence come IP Addresses?

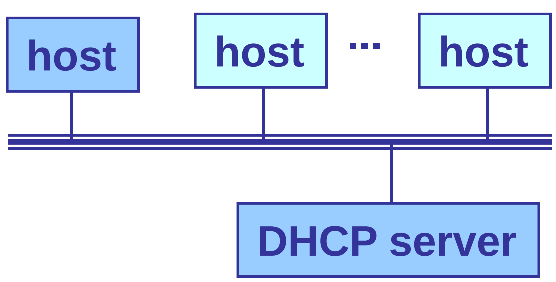
- You already have a bunch from the days when you called Jon Postel and asked for them (e.g. BBN)
- You get them from another provider
  - E.g. buy service from Sprint and get a /24 from one of their address blocks
- You get one directly from a routing registry
  - ARIN: North America, APNIC (Asia Pacific), RIPE (Europe), LACNIC (Latin America), etc.
  - Registries get address from IANA (Internet Assigned Numbers Authority)

How Do You And I Get One?

- Well from your provider!
- But how do you know what it is?
- Manual configuration
  - They tell you and you type that number into your computer (along with the default gateway, DNS server, etc.)
- Automated configuration
  - Dynamic Host Resolution Protocol (DHCP)

Bootstrapping Problem

- Host doesn't have an IP address yet
  - So, host doesn't know what source address to use
- Host doesn't know who to ask for an IP address
  - So, host doesn't know what destination address to use
- Solution: shout to discover a server who can help
  - Install a special server on the LAN to answer distress calls



DHCP

- Broadcast-based LAN protocol algorithm
  - Host broadcasts “DHCP discover” on LAN (e.g. Ethernet broadcast)
  - DHCP server responds with “DHCP offer” message
  - Host requests IP address: “DHCP request” message
  - DHCP server sends address: “DHCP ack” message w/IP address
- Easy to have fewer addresses than hosts (e.g. UCSD wireless) and to *renumber* network (use new addresses)
- What if host goes away (how to get address back?)
  - Address is a “lease” not a “grant”, has a timeout
  - Host may have different IP addresses at different times?

Mapping IP to link-layer

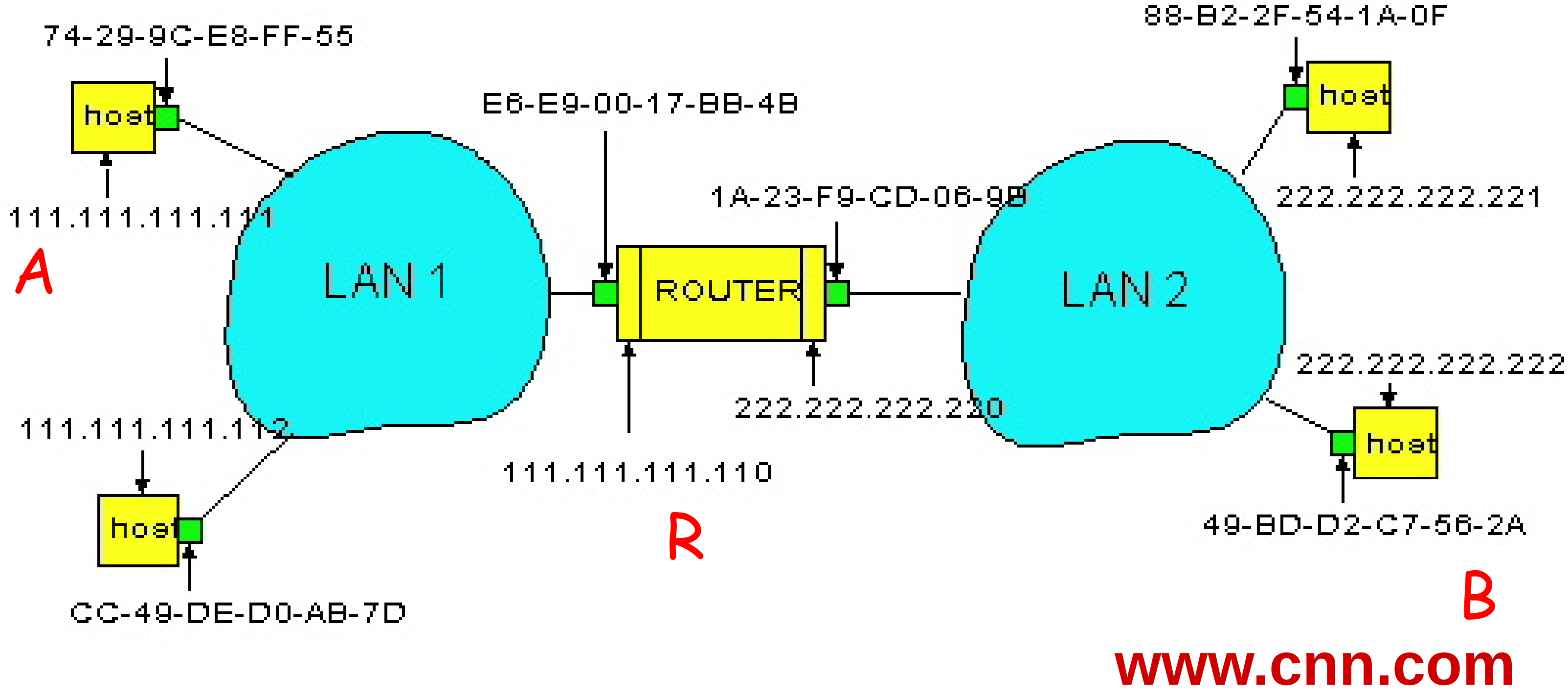
- Ok, you have an IP address you want to send to
  - If its not on your LAN then it goes to the router
  - What if it is on your LAN?
  - Now that you mention it, where's the router on your LAN?

- Key question: how to map IP addresses to link-layer addresses?
  - What should I put in the destination field of the Ethernet packet?

Address Resolution Protocol

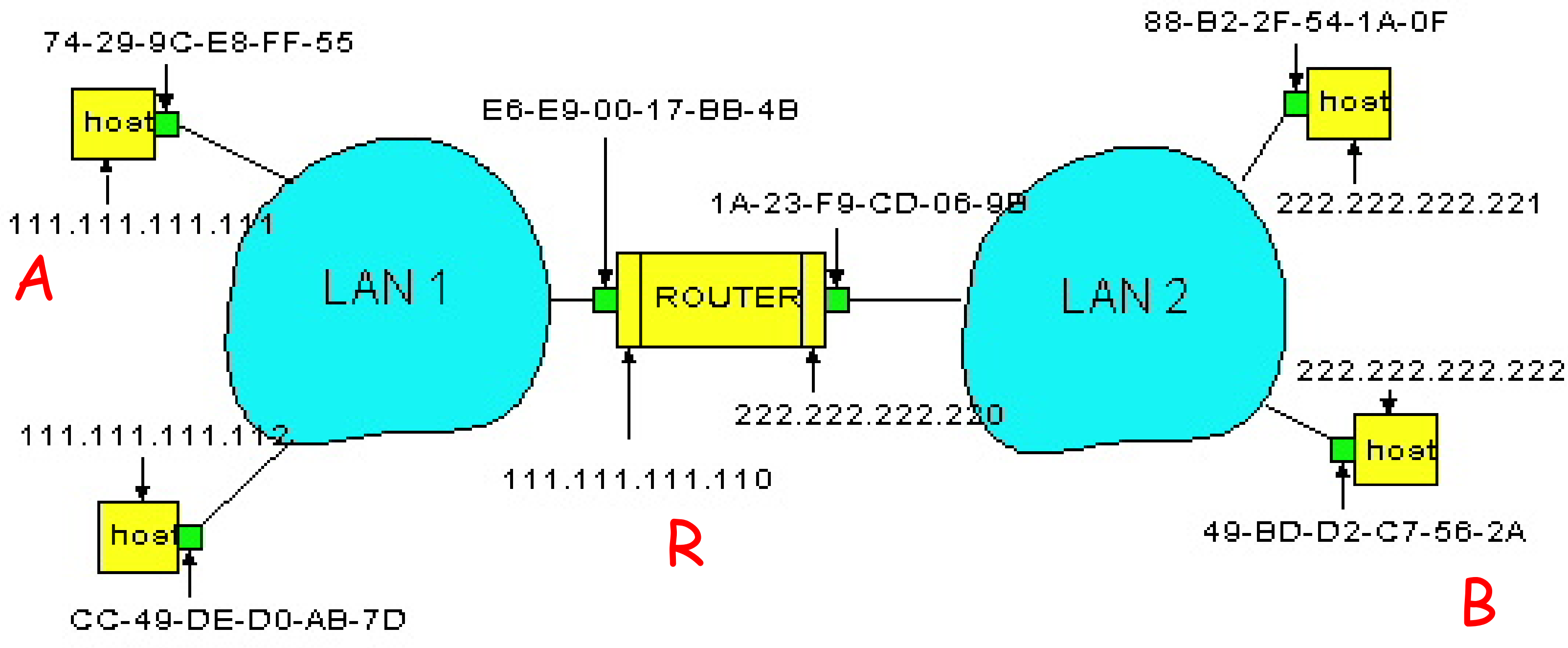
- Every node maintains an ARP table
  - (IP address, MAC address) pair
- Consult the table when sending a packet
  - Map destination IP address to MAC address
  - Encapsulate and transmit the data packet
- What if the IP address is not in the table?
  - Broadcast: “Who has IP address *x.x.x.x*?”
  - Response: “MAC address *yy:yy:yy:yy:yy:yy*”
  - Sender caches the result in its ARP table

Example: Sending to CNN



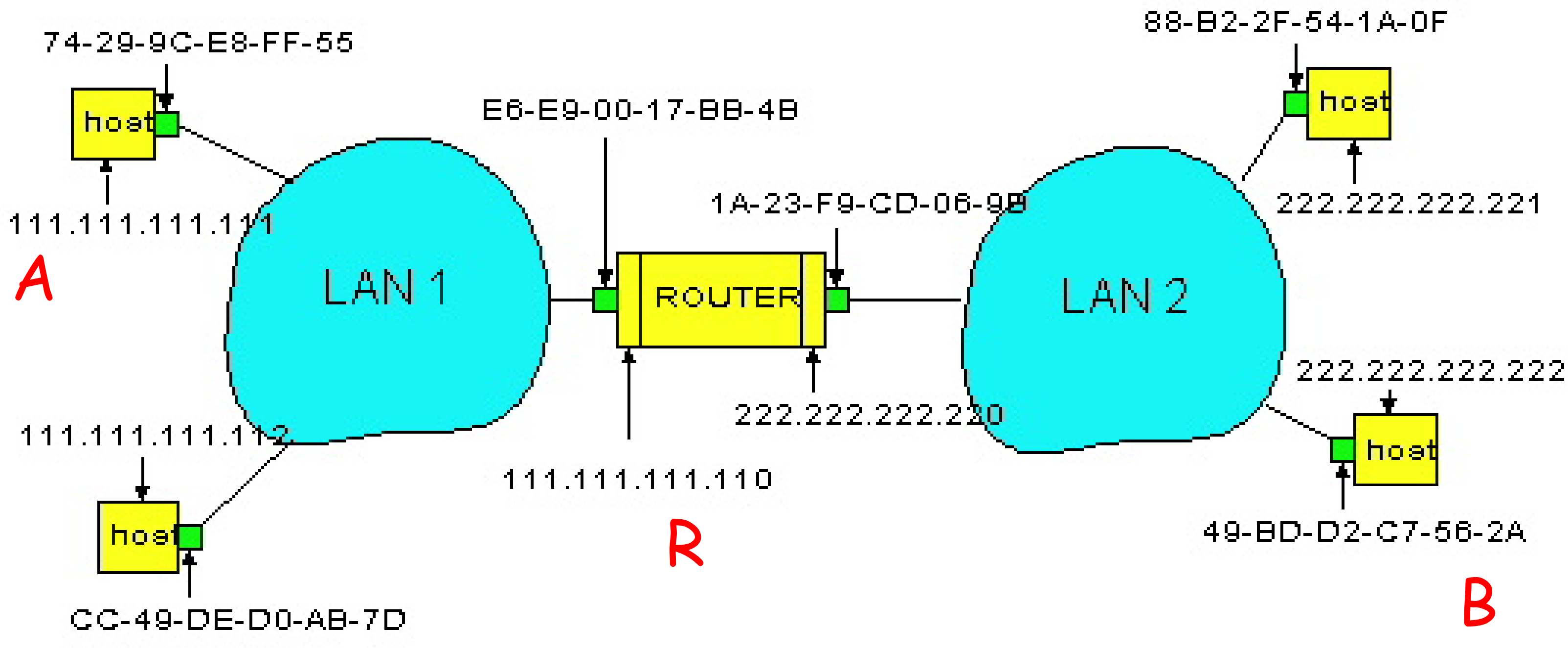
Basic Steps

1. Host *A* must learn the IP address of *B* (via DNS)
2. Host *A* uses gateway *R* to reach external hosts
3. Router *R* forwards IP packet to outgoing interface
4. Router *R* learns *B*'s MAC address and forwards frame



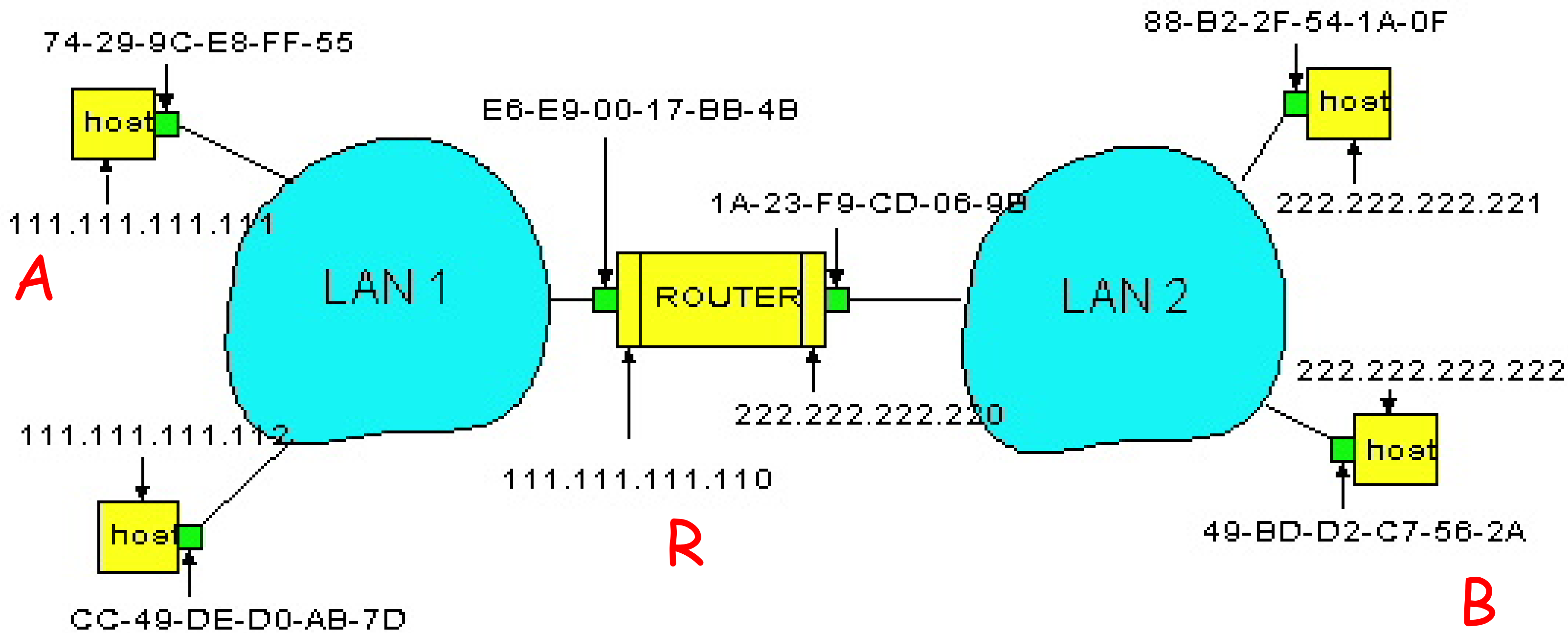
Host *A* Learns *B*'s IP Address

- Host *A* does a DNS query to learn *B*'s address
  - DNS service returns 222.222.222.222 (more in later class)
- Host *A* constructs an IP packet to send to *B*
  - Source 111.111.111.111, dest 222.222.222.222



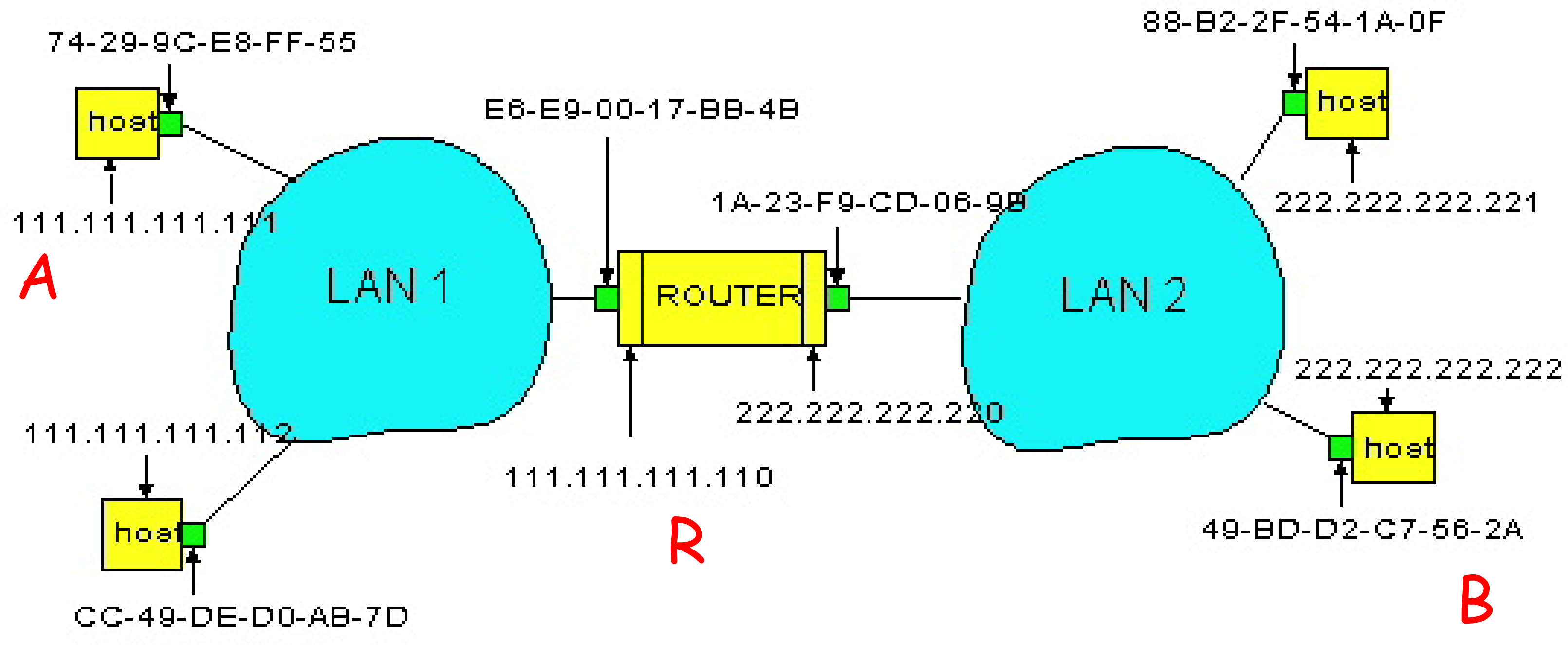
Host *A* Learns *B*'s IP Address

- IP packet
  - From *A*: 111.111.111.111
  - To *B*: 222.222.222.222
- Ethernet frame
  - From *A*: 74-29-9C-E8-FF-55
  - To gateway: ????



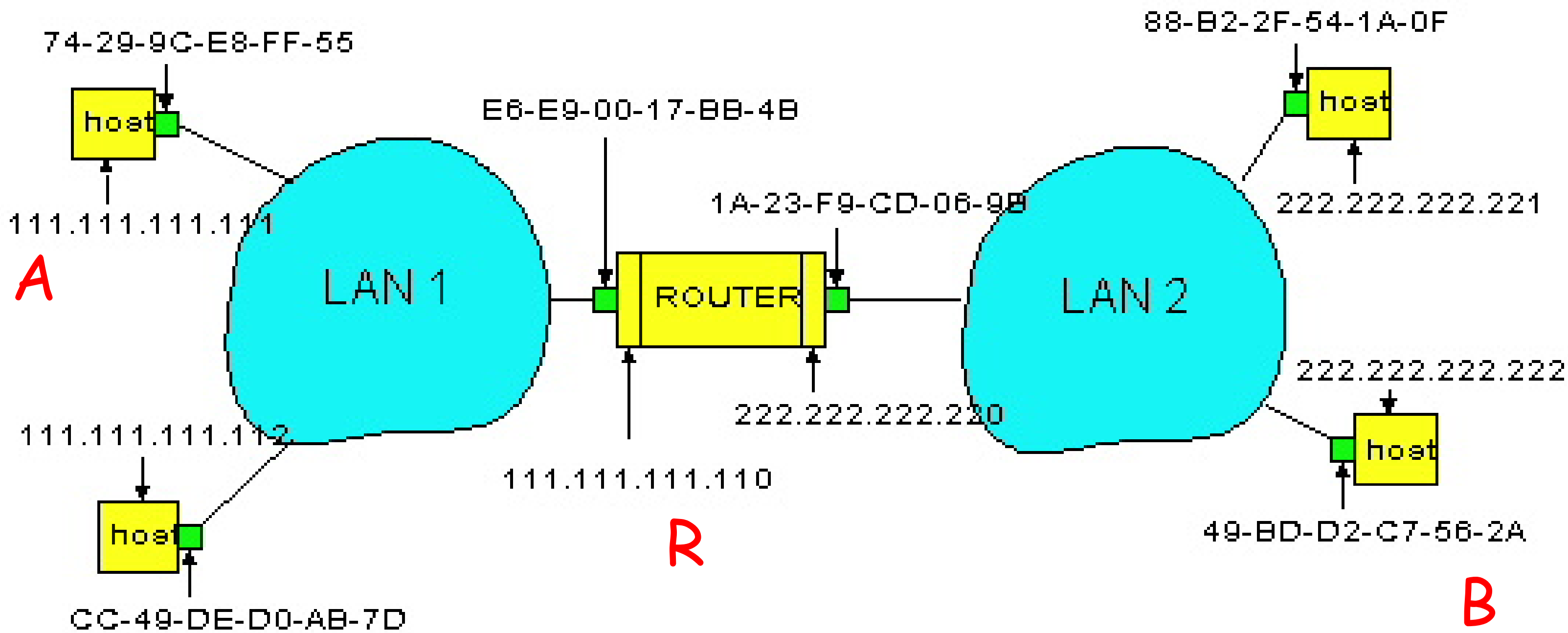
A Decides to Send Through *R*

- Host *A* has a gateway router *R*
  - Used to reach dests outside of 111.111.111.0/24
  - Address 111.111.111.110 for *R* learned via DHCP
- But, what is the MAC address of the gateway?



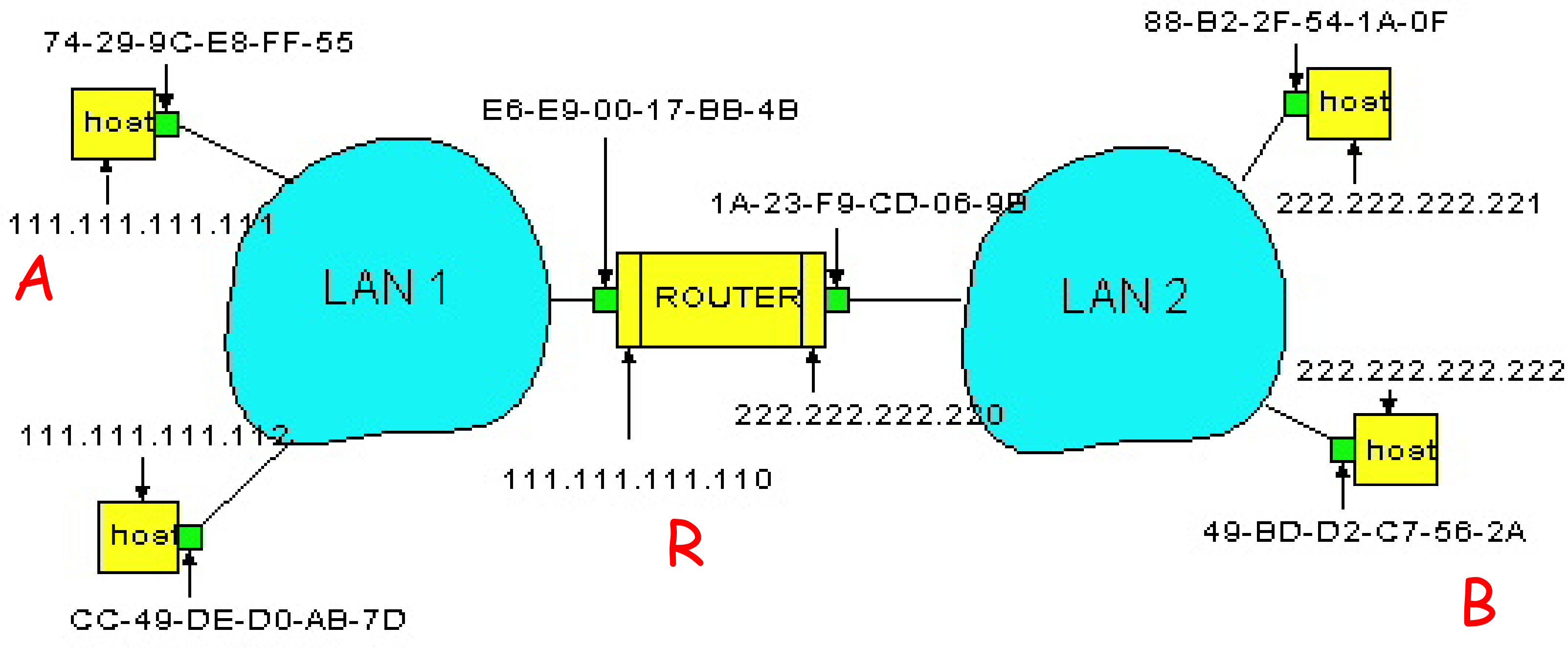
A Sends Packet Through *R*

- Host *A* learns the MAC address of *R*'s interface
  - ARP request: broadcast request for 111.111.111.110
  - ARP response: R responds with E6-E9-00-17-BB-4B
- Host *A* encapsulates the packet and sends to *R*



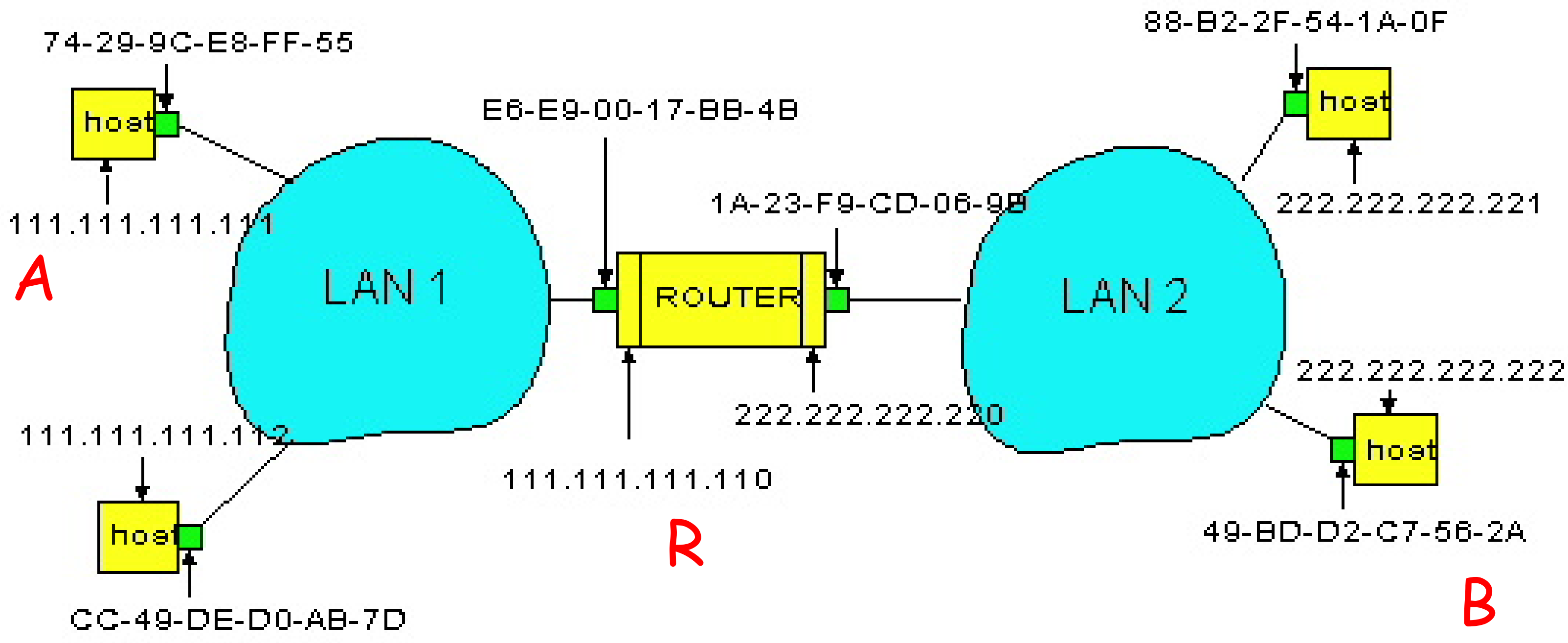
A Sends Packet Through *R*

- IP packet
  - From *A*: 111.111.111.111
  - To *B*: 222.222.222.222
- Ethernet frame
  - From *A*: 74-29-9C-E8-FF-55
  - To *R*: E6-E9-00-17-BB-4B



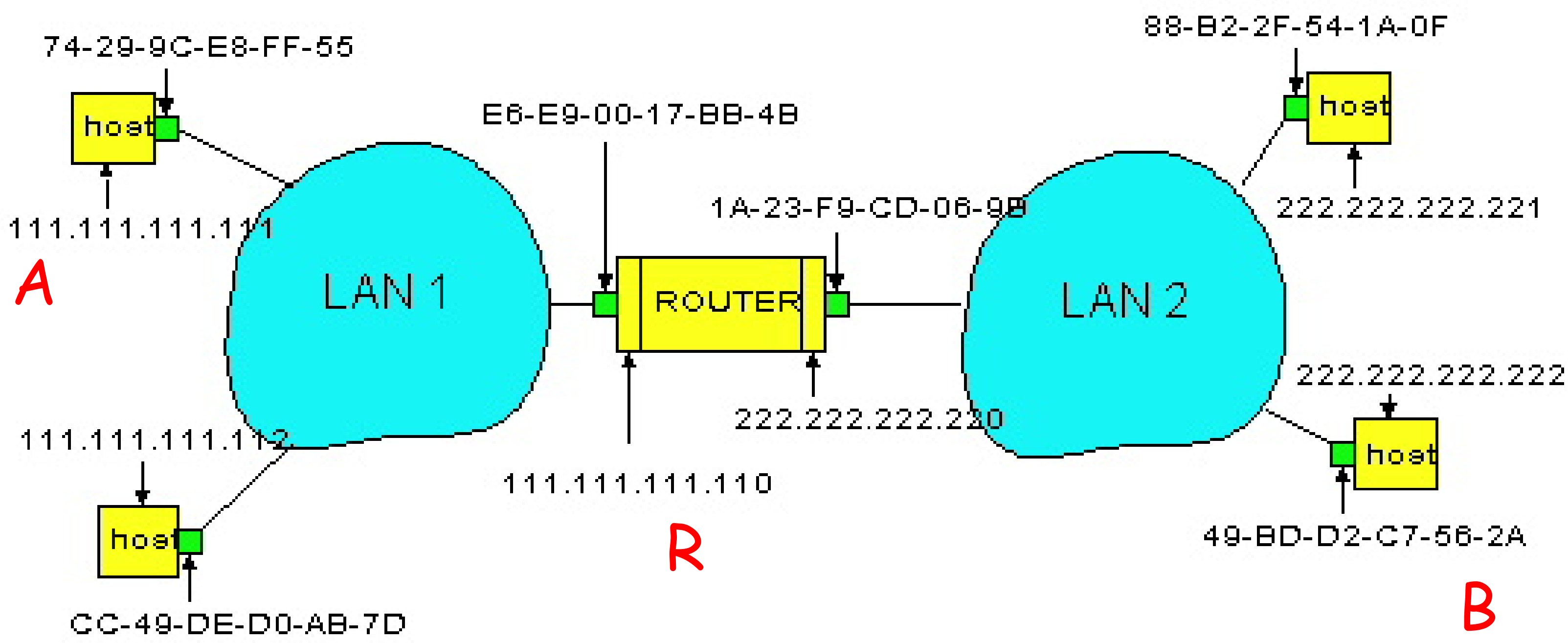
R Looks up Next Hop

- Router *R*'s adapter receives the packet
  - *R* extracts the IP packet destined to 222.222.222.222
- Router *R* consults its forwarding table
  - Packet matches 222.222.222.0/24 via other interface



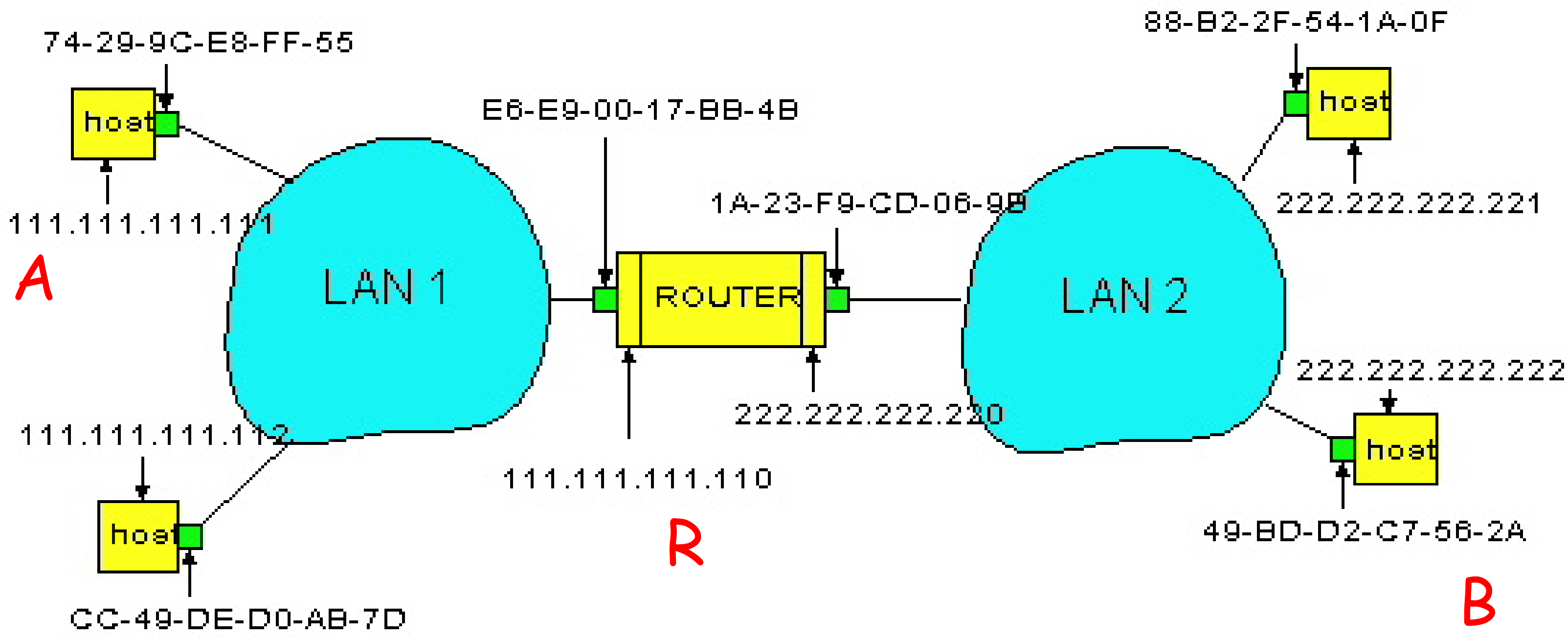
R Wants to Forward Packet

- IP packet
  - From *A*: 111.111.111.111
  - To *B*: 222.222.222.222
- Ethernet frame
  - From *R*: 1A-23-F9-CD-06-9B
  - To *B*: ???



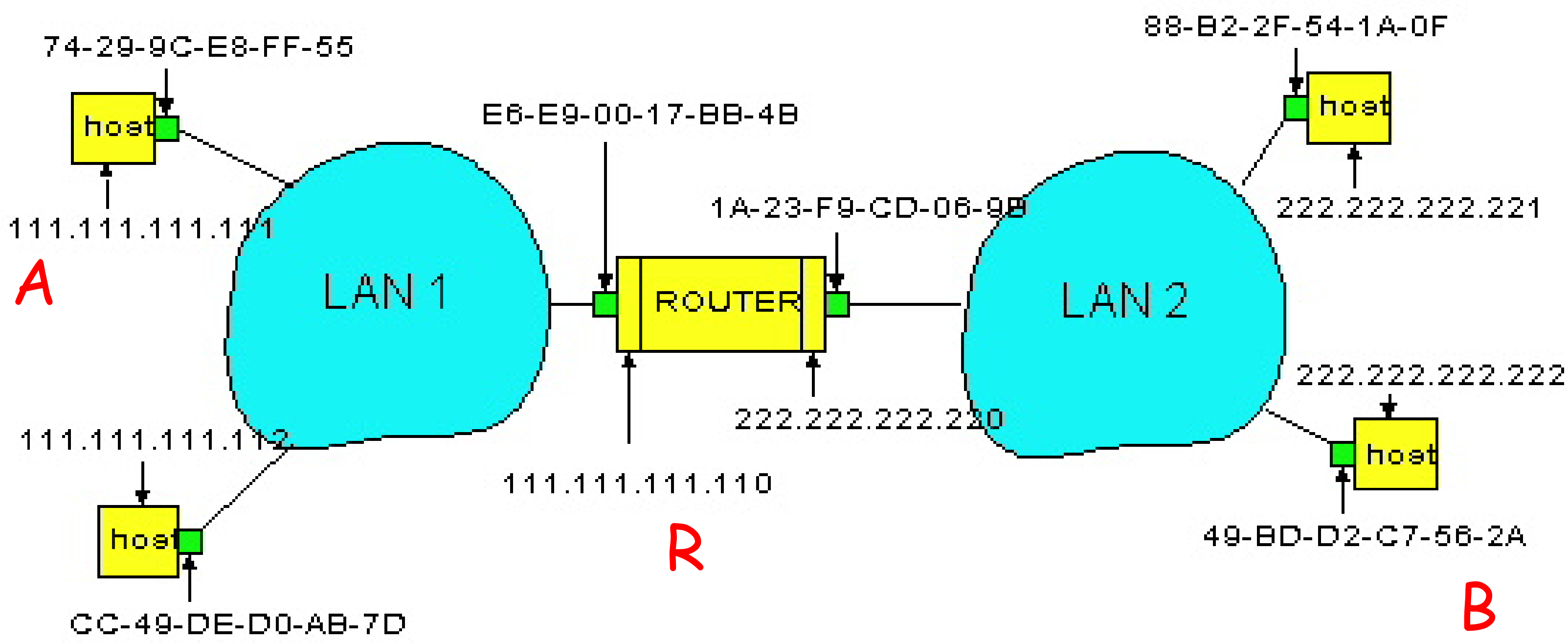
R Sends Packet to *B*

- Router *R*'s learns the MAC address of host *B*
  - ARP request: broadcast request for 222.222.222.222
  - ARP response: *B* responds with 49-BD-D2-C7-56-2A
- Router *R* encapsulates the packet and sends to *B*



R Wants to Forward Packet

- IP packet
  - From *A*: 111.111.111.111
  - To *B*: 222.222.222.222
- Ethernet frame
  - From *R*: 1A-23-F9-CD-06-9B
  - To *B*: 49-BD-D2-C7-56-2A



Some observations…

- The Internet was designed
  - There is no natural law that says TCP/IP, network routing, etc.. had to look the way it does now
  - It could (and maybe should?) have been done differently
- The Internet evolves
  - The Internet today is not the same Internet as 1988 or 1973
  - IP (and other protocols) have changed considerably over the years (and continue to change -> IPv6)
- Many of these design issues are deep
  - Seemingly straightforward decisions can have very subtle correctness and performance implications
  - E.g. Implications of fragmentation
- This concludes our discussion of IP

For Next Time

- Read P&D 3.2: routing