Translating Addresses

Goals of Today's Lecture

Three different kinds of addresses

- Host names (e.g., www.cnn.com)
- IP addresses (e.g., 64.236.16.20)
- MAC addresses (e.g., 00-15-C5-49-04-A9)

Protocols for translating between addresses

- Domain Name System (DNS)
- Dynamic Host Configuration Protocol (DHCP)
- Address Resolution Protocol (ARP)

Two main topics

- Decentralized management of the name space
- Boot-strapping an end host that attaches to the 'net

Host Names vs. IP addresses

Host names

- Mnemonic name appreciated by humans
- Variable length, alpha-numeric characters
- Provide little (if any) information about location
- Examples: www.cnn.com and ftp.eurocom.fr

IP addresses

- Numerical address appreciated by routers
- Fixed length, binary number
- Hierarchical, related to host location
- Examples: 64.236.16.20 and 193.30.227.161

Separating Naming and Addressing

Names are easier to remember

- www.cnn.com vs. 64.236.16.20

Addresses can change underneath

- Move www.cnn.com to 173.15.201.39
- E.g., renumbering when changing providers

Name could map to multiple IP addresses

- www.cnn.com to multiple replicas of the Web site

Separating Naming and Addressing

Map to different addresses in different places

- Address of a nearby copy of the Web site
- E.g., to reduce latency, or return different content

Multiple names for the same address

- E.g., aliases like ee.mit.edu and cs.mit.edu

Solution #1: Local File

Original name to address mapping

- Flat namespace
- /etc/hosts
- Downloaded regularly

Count of hosts was increasing: moving from a machine per domain to machine per user

- Many more downloads
- Many more updates

Solution #2: Central Server

Central server

- One place where all mappings are stored
- All queries go to the central server

Many practical problems

- Single point of failure
- High traffic volume
- Distant centralized database
- Does not scale

Need a distributed, hierarchical collection of servers

Domain Name System (DNS)

Properties of DNS

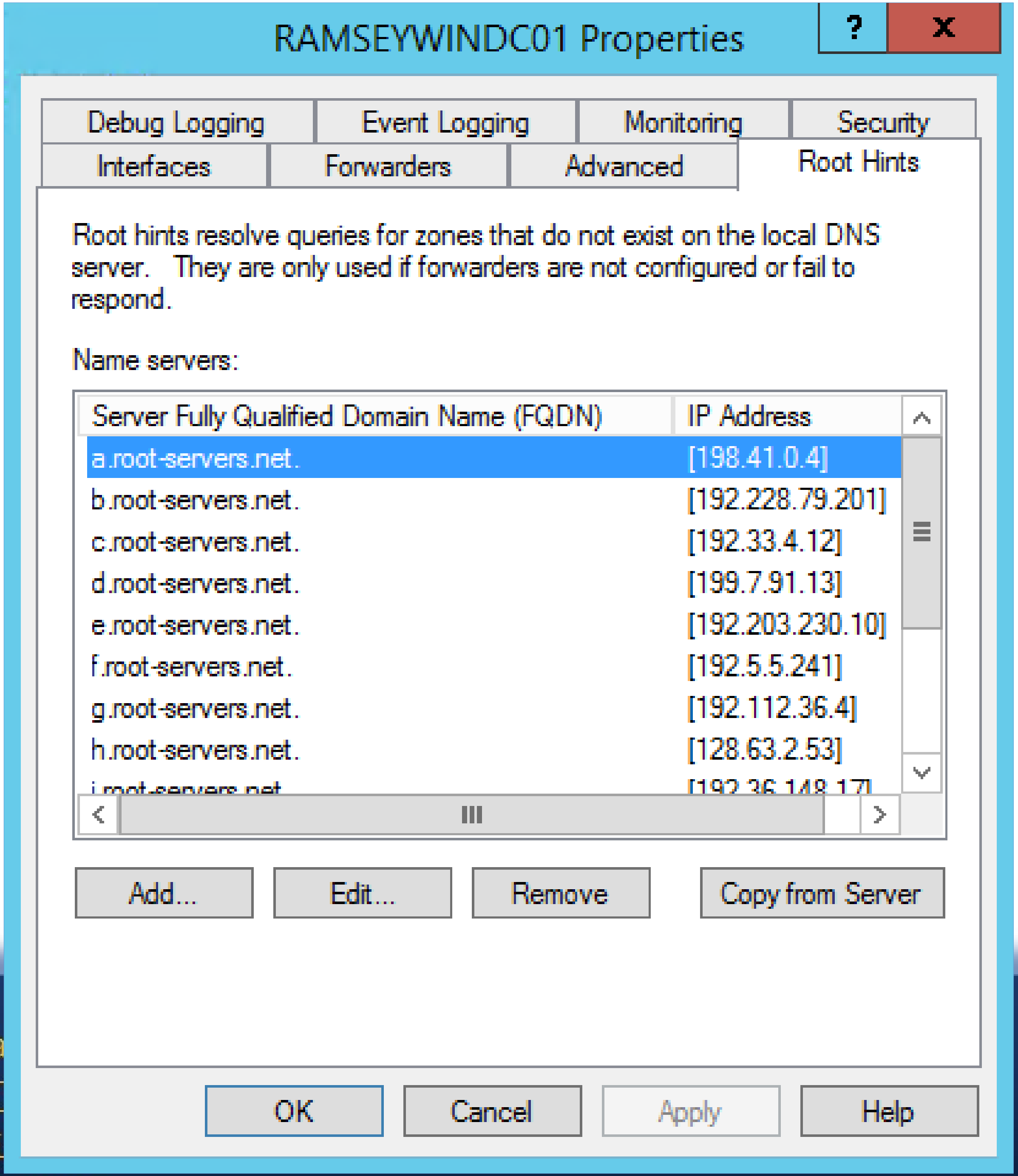
- Hierarchical name space divided into zones
- Distributed over a collection of DNS servers

Hierarchy of DNS servers

- Root servers (13 clusters, labelled A-M)
- Top-level domain (TLD) servers
- Authoritative DNS servers

Greenland
Iceland
Suomi
Finland
Norge
Norway
Россия
Russia
Canada
United States
Kazakhstan
Mongolia
中国
China
Japan
México
North Atlantic Ocean
España
Spain
Italia
Italy
Algeria
Libya
Egypt
Saudi Arabia
Iraq
Mauritania
Mali
Niger
Chad
Sudan
Nigeria
Kenya
DR Congo
Tanzania
Angola
Namibia
Madagascar
South Africa
Venezuela
Perú
Brasil
Brazil
Bolivia
Argentina
South Atlantic Ocean
Indian Ocean
Thailand
Indonesia
Papua New Guinea
Australia
New Zealand
Legend
Multiple instances
Single instance

DNS Root Servers



TLD and Authoritative DNS Servers

Top-level domain (TLD) servers

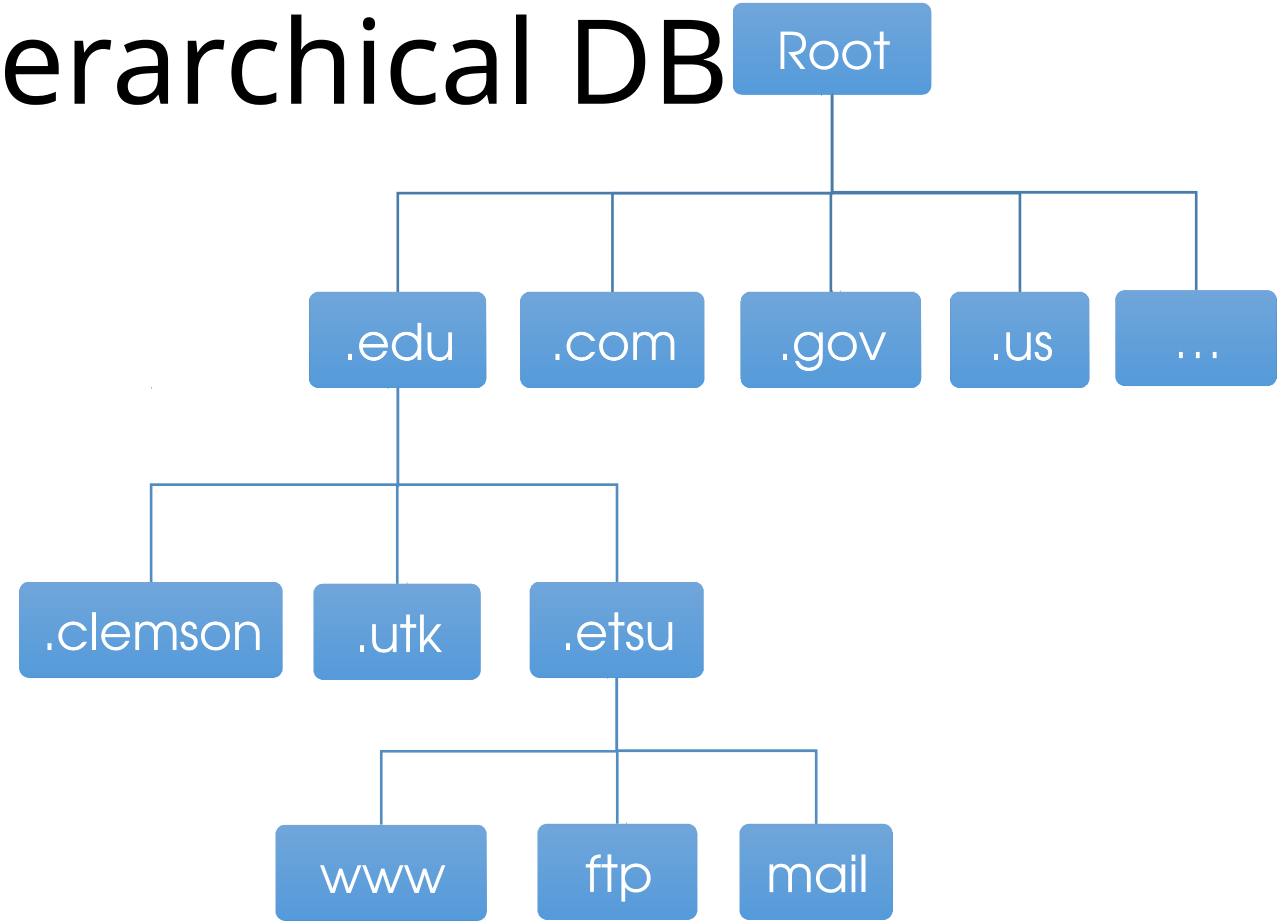
- Generic domains (e.g., com, org, edu)
- Country domains (e.g., uk, fr, ca, jp)
- Typically managed professionally
  - Network Solutions maintains servers for “com”
  - Educause maintains servers for “edu”

TLD and Authoritative DNS Servers

Authoritative DNS servers (e.g., RAMSEYDC01)

- Provide public records for hosts at an organization
- For the organization's servers (e.g., Web and mail)
- Can be maintained locally or by a service provider

Distributed Hierarchical DB



Using DNS

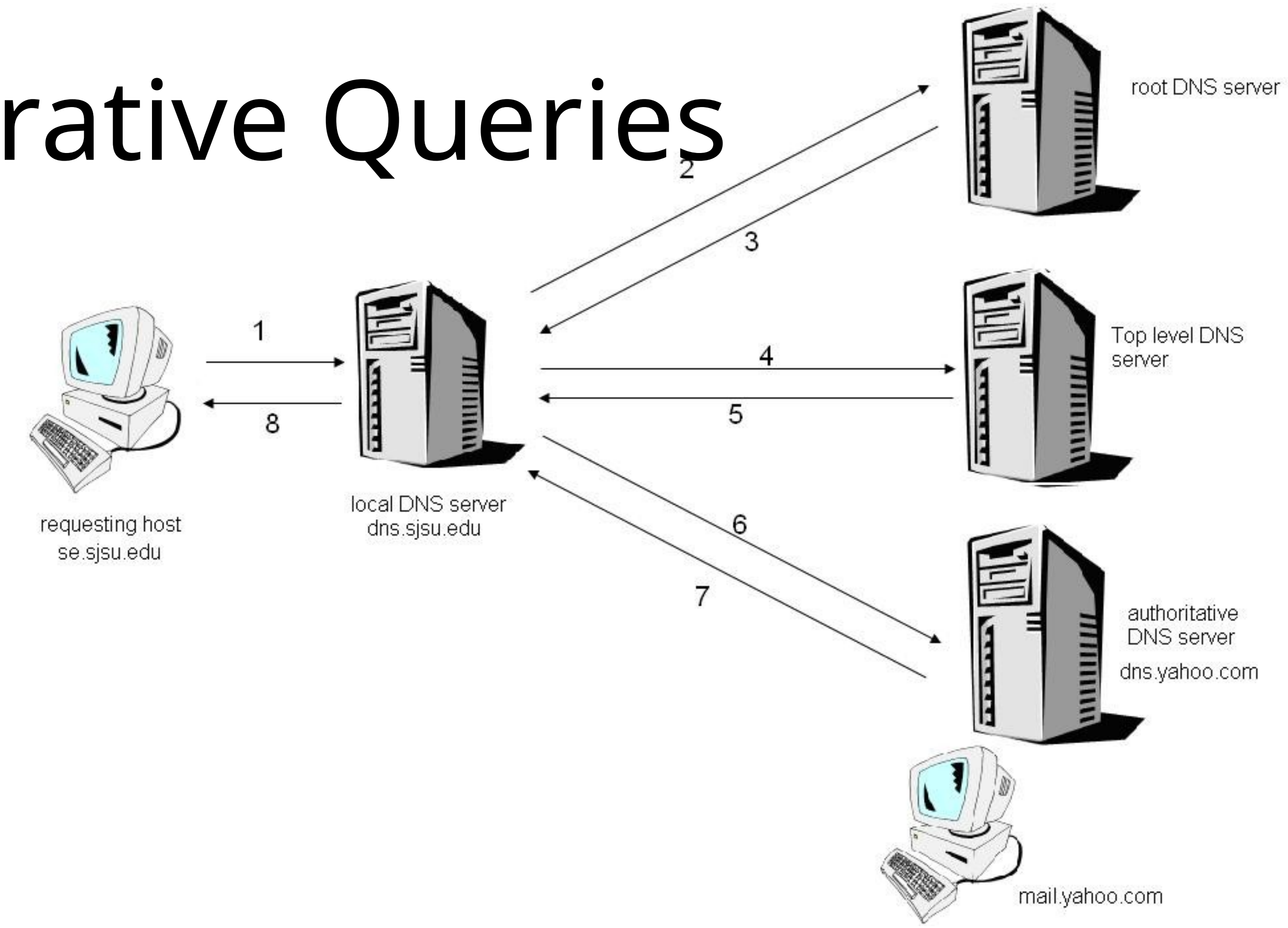
Local DNS server ("default name server")

- Usually (geographically) near the end hosts who use it
- Local hosts configured with local server (e.g., /etc/resolv.conf) or learn the server via DHCP

Recursive vs. Iterative Queries

Iterative query

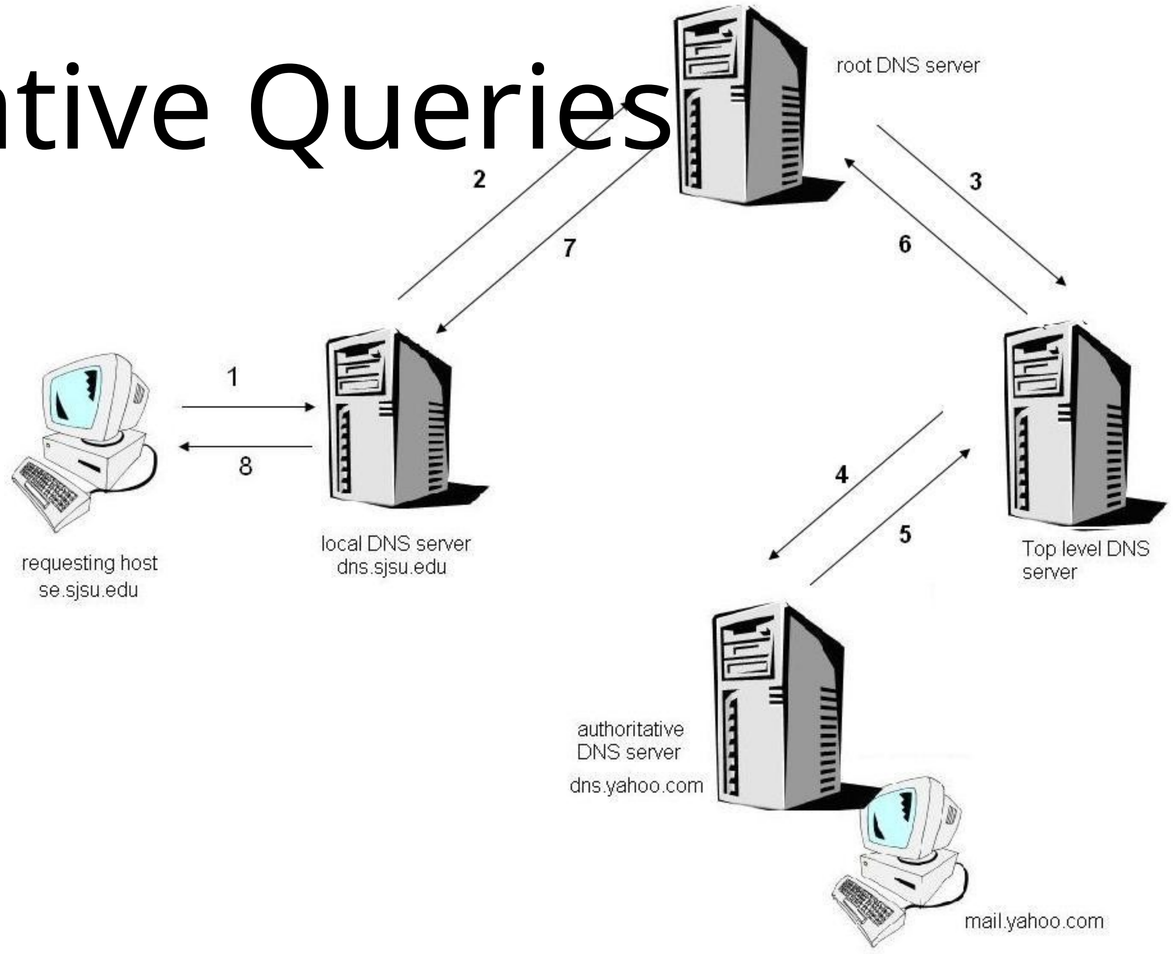
- Ask server who to ask next
- E.g., all other request-response pairs



Recursive vs. Iterative Queries

Recursive query

- Ask server to get answer for you
- E.g., request 1 and response 8



DNS Caching

Performing all these queries take time

- And all this before the actual communication takes place
- E.g., 1-second latency before starting Web download

Caching can substantially reduce overhead

- The top-level servers very rarely change
- Popular sites (e.g., www.cnn.com) visited often
- Local DNS server often has the information cached

DNS Caching

How DNS caching works

DNS servers cache responses to queries

Responses include a “time to live” (TTL) field

Server deletes the cached entry after TTL expires

Negative Caching

Remember things that don't work

- Misspellings like www.cnn.comm and www.cnnn.com
- These can take a long time to fail the first time
- Good to remember that they don't work
- … so the failure takes less time the next time around

DNS Resource Records

DNS: distributed db storing resource records (RR)

RR format: (name, value, type, ttl)

Type=A
- name is hostname
- value is IP address

Type=NS

`name` is domain (e.g. foo.com)

`value` is hostname of authoritative name server for this domain

DNS Resource Records

DNS: distributed db storing resource records (RR)

RR format: (name, value, type, ttl)

Type=CNAME

- `name` is alias name for some "canonical" (the real) name

  `www.ibm.com` is really `servereast.backup2.ibm.com`
- `value` is canonical name

Type=MX

- `value` is name of mailserver associated with `name`

Reliability

DNS servers are replicated

- Name service available if at least one replica is up
- Queries can be load balanced between replicas

UDP used for queries

- Need reliability: must implement this on top of UDP

Try alternate servers on timeout

Same identifier for all queries

- Don’t care which server responds

Inserting Resource Records into DNS

Example: just created startup "FooBar"

Register foobar.com at Network Solutions

- Provide registrar with names and IP addresses of your authoritative name server (primary and secondary)
- Registrar inserts two RRs into the com TLD server:
  - (foobar.com, dns1.foobar.com, NS)
  - (dns1.foobar.com, 212.212.212.1, A)

Inserting Resource Records into DNS

Put in authoritative server dns1.foobar.com

- Type A record for www.foobar.com
- Type MX record for foobar.com

Playing With Dig on UNIX

- Dig program
  - "domain information grouper"
  - Allows querying of DNS system
  - Use flags to find name server (NS)
  - Disable recursion so that operates one step at a time

Terminal - jack@azure: ~

```
; <<>> DiG 9.9.5-3-Ubuntu <<>> +recurse etsu.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34482
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0005 , udp: 4000
;; QUESTION SECTION:
;etsu.edu.                      IN      A

;; ANSWER SECTION:
etsu.edu.               5       IN      A       151.141.8.110
etsu.edu.               5       IN      A       151.141.8.101
etsu.edu.               5       IN      A       151.141.5.62
etsu.edu.               5       IN      A       151.141.8.100

;; Query time: 5 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Thu Mar 19 12:09:45 EDT 2015
;; MSG SIZE  rcvd: 101

jack@azure:~$
```

ETSU

ipconfig

Windows Utility

- Can display client network information

C:\Windows\System32\cmd.exe

```
C:\Windows\System32>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : Hurin
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : wifi.etsu.edu

Wireless LAN adapter Local Area Connection* 3:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Microsoft Wi-Fi Direct Virtual Adapter
   Physical Address. . . . . . . . . : 80-86-F2-7E-AB-13
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Bluetooth Device (Personal Area Network)
   Physical Address. . . . . . . . . : 80-86-F2-7E-AB-16
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : wifi.etsu.edu
   Description . . . . . . . . . . . : Intel(R) Dual Band Wireless-N 7260
   Physical Address. . . . . . . . . : 80-86-F2-7E-AB-12
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::f5e6:25f3:638a:1922%4(Preferred)
   IPv4 Address. . . . . . . . . . . : 151.141.227.37(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.248.0
   Lease Obtained. . . . . . . . . . : Tuesday, October 14, 2014 8:01:55 AM
   Lease Expires . . . . . . . . . . : Tuesday, October 14, 2014 7:01:44 PM
   Default Gateway . . . . . . . . . : 151.141.224.1
```

ipconfig

Windows Utility

- Can display client network information
- Can also display/flush cached DNS information

C:\Windows\System32\cmd.exe

```
C:\Windows\System32>ipconfig /displaydns

Windows IP Configuration

    i.creativecommons.org
    ----------------------------------------
    Record Name . . . . . : i.creativecommons.org
    Record Type . . . . . : 1
    Time To Live  . . . . : 885
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 213.138.111.164


    api.mixpanel.com
    ----------------------------------------
    Record Name . . . . . : api.mixpanel.com
    Record Type . . . . . : 1
    Time To Live  . . . . : 75242
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 174.36.240.85


    Record Name . . . . . : api.mixpanel.com
    Record Type . . . . . : 1
    Time To Live  . . . . : 75242
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 174.36.240.86


    Record Name . . . . . : api.mixpanel.com
    Record Type . . . . . : 1
    Time To Live  . . . . : 75242
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 174.36.240.84
```

Boot-Strapping an End Host

DHCP and ARP

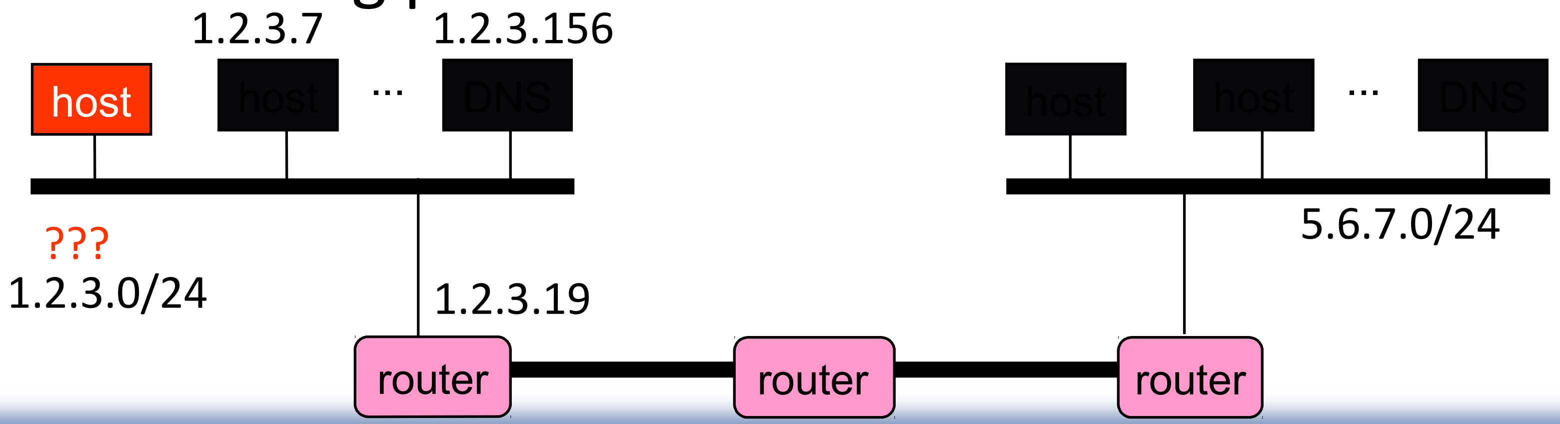
How To Bootstrap an End Host?

What local Domain Name System server to use?

What IP address the host should use?

How to send packets to remote destinations?

How to ensure incoming packets arrive?



Avoiding Manual Configuration

Dynamic Host Configuration Protocol (DHCP)

- End host learns how to send packets
- Learn IP address, DNS servers, and gateway

Address Resolution Protocol (ARP)

- Others learn how to send packets to the end host
- Learn mapping between IP address & interface address

Key Ideas in Both Protocols

Broadcasting: when in doubt, shout!

- Broadcast query to all hosts in the local-area-network
- … when you don't know how to identify the right one

Caching: remember the past for a while

- Store the information you learn to reduce overhead
- Remember your own address & other host's addresses

Key Ideas in Both Protocols

Soft state: … but eventually forget the past

- Associate a time-to-live field with the information
- … and either refresh or discard the information
- Key for robustness in the face of unpredictable change

Need Yet Another Kind of Identity

- LANs are designed for arbitrary network protocols
  - Not just for IP and the Internet
- Using IP address would require reconfiguration
  - Every time the adapter was moved or powered up
- Broadcasting all data to all adapters is expensive
  - Requires every host on the LAN to inspect each packet

Motivates separate Media Access Control (MAC) addresses

MAC Address vs. IP Address

MAC addresses

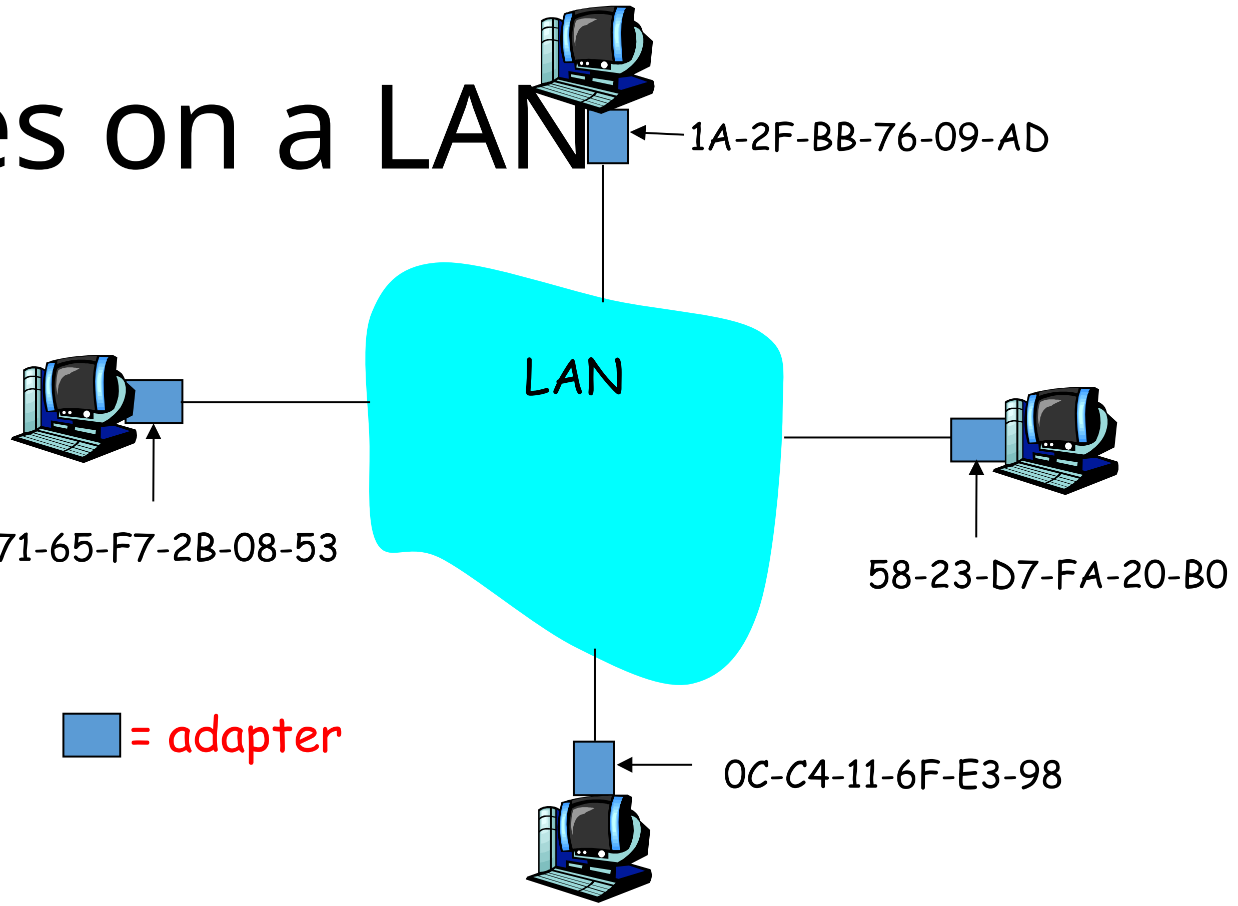
- Hard-coded in read-only memory when adaptor is built
- Like a social security number
- Flat name space of 48 bits (e.g., 00-0E-9B-6E-49-76)
- Portable, and can stay the same as the host moves
- Used to get packet between interfaces on same network

MAC Address vs. IP Address

IP addresses

- Configured, or learned dynamically
- Like a postal mailing address
- Hierarchical name space of 32 bits (e.g., 12.178.66.9)
- Not portable, and depends on where the host is attached
- Used to get a packet to destination IP subnet

MAC Addresses on a LAN



Bootstrapping Problem

Host doesn’t have an IP address yet

- So, host doesn’t know what source address to use

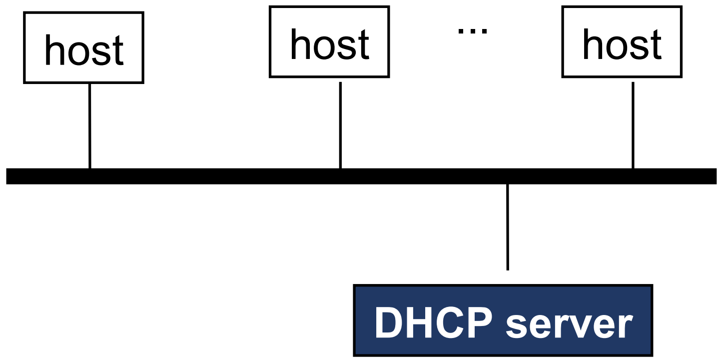
Host doesn’t know who to ask for an IP address

- So, host doesn’t know what destination address to use

Bootstrapping Problem

Solution: shout to discover a server who can help

- Broadcast a server-discovery message
- Server sends a reply offering an address



Broadcasting

Broadcasting: sending to everyone

- Special destination address: FF-FF-FF-FF-FF-FF
- All adapters on the LAN receive the packet

Delivering a broadcast packet

- Easy on a “shared media”
- Like shouting in a room – everyone can hear you

Response from the DHCP Server

DHCP “offer message” from the server

- Configuration parameters (proposed IP address, mask, gateway router, DNS server, ...)
- Lease time (the time the information remains valid)

Multiple servers may respond

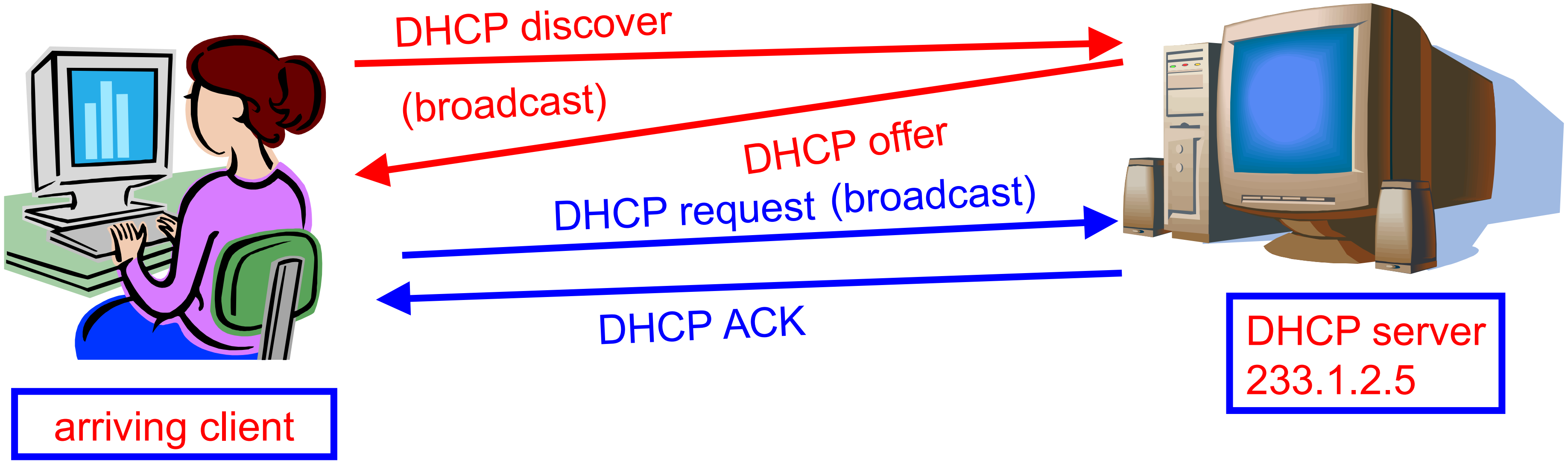
- Multiple servers on the same broadcast media
- Each may respond with an offer
- The client can decide which offer to accept

Response from the DHCP Server

Accepting one of the offers

- Client sends a DHCP request echoing the parameters
- The DHCP server responds with an ACK to confirm
- … and the other servers see they were not chosen

Dynamic Host Configuration Protocol



Deciding What IP Address to Offer

Server as centralized configuration database

- All parameters are statically configured in the server
- E.g., a dedicated IP address for each MAC address
- Avoids complexity of configuring hosts directly
- … while still having a permanent IP address per host

Deciding What IP Address to Offer

Or, dynamic assignment of IP addresses

- Server maintains a pool of available addresses
- … and assigns them to hosts on demand
- Leads to less configuration complexity
- … and more efficient use of the pool of addresses
- Though, it is harder to track the same host over time

Soft State: Refresh or Forget

Why is a lease time necessary?

- Client can release the IP address (DHCP RELEASE)
  - E.g., “ipconfig /release” at the DOS prompt
  - E.g., clean shutdown of the computer
- But, the host might not release the address
  - E.g., the host crashes
  - E.g., buggy client software
- And you don’t want the address to be allocated forever

Soft State: Refresh or Forget

Performance trade-offs

- Short lease time: returns inactive addresses quickly
- Long lease time: avoids overhead of frequent renewals

So, Now the Host Knows Things

IP address

Mask

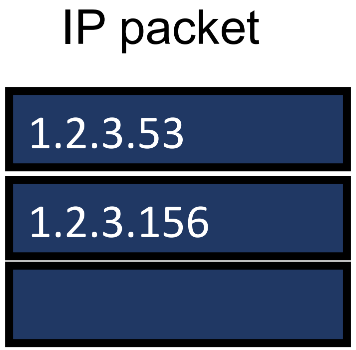
Gateway router

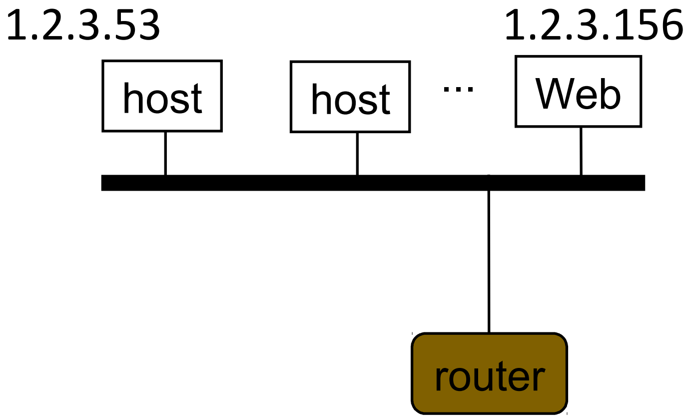
DNS server,

And can send packets to other IP addresses

- But, how to learn the MAC address of the destination?

Sending Packets Over a Link





Adaptors only understand MAC addresses

- Translate the destination IP address to MAC address
- Encapsulate the IP packet inside a link-level frame

Address Resolution Protocol Table

Every node maintains an ARP table

- (IP address, MAC address) pair

Consult the table when sending a packet

- Map destination IP address to destination MAC address
- Encapsulate and transmit the data packet

Address Resolution Protocol Table

But, what if the IP address is not in the table?

- Sender broadcasts: “Who has IP address 1.2.3.156?”
- Receiver responds: “MAC address 58-23-D7-FA-20-B0”
- Sender caches the result in its ARP table

No need for network administrator to get involved

C:\Windows\System32\cmd.exe

```
C:\Windows\System32>arp -a

Interface: 151.141.227.37 --- 0x4
  Internet Address      Physical Address      Type
  151.141.224.1         58-bf-ea-be-c2-80     dynamic
  151.141.231.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 192.168.92.1 --- 0x17
  Internet Address      Physical Address      Type
  192.168.92.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 192.168.244.1 --- 0x18
  Internet Address      Physical Address      Type
  192.168.244.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\Windows\System32>
```

What happens if there’s no DHCP?

APIPA

- “Automatic Private IP Addressing”
- Microsoft creation allows a client to self-configure its IP and subnet when no DHCP server available
- Reserved address range: 169.254.0.1 through 169.254.255.254
- Continues to check for DHCP server, obtains network IP address – replacing APIPA address – if DHCP is found

Conclusion

Domain Name System

- Distributed, hierarchical database
- Distributed collection of servers
- Caching to improve performance

Bootstrapping an end host

- Dynamic Host Configuration Protocol (DHCP)
- Address Resolution Protocol (ARP)

Copyrights

Presentation prepared by and copyright of John Ramsey, East Tennessee State University, Department of Computing . (ramseyjw@etsu.edu)

•Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

•IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

•Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

•Oracle is a registered trademark of Oracle Corporation.

•HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

•Java is a registered trademark of Sun Microsystems, Inc.

•JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

•SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

•Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

•ERPsim is a registered copyright of ERPsim Labs, HEC Montreal.

•Other products mentioned in this presentation are trademarks of their respective owners.