Chapter 2

DNS and Active Directory

Outline

Part 1

- DNS
  - Overview
  - Recursive and Iterative Queries
  - Recursive or Caching DNS Server
  - Resource record and DNS query
  - DNS Protocol
  - Reverse DNS lookup
- Active Directory (AD)
  - Overview and applications of AD
  - Structure and trust
  - AD objects
  - SRV RR
  - The Open Directory (OD)

DNS evolution

- Historically the host file is located on the local machine
  - E.g., c:\windows\system32\drivers\etc\hosts
  - Need to be maintained and updated by an administrator
- Maintaining the hosts files for all Internet domain names and sub domains is not feasible
  - Hence the birth of a distributed database that is called DNS
  - A service run by a myriad of organizations, ISP's and Internet authorities (ICANN)
  - To facilitate the mapping of URL to IP addresses

DNS (Domain Name System)

- Internet host's and router interfaces:
  - IPv4 address (32 bit): used for addressing datagrams
  - "name", e.g., www.cnn.com, used by humans
- Translating from www.cnn.com to 64.236.16.20 in order to deliver a datagram

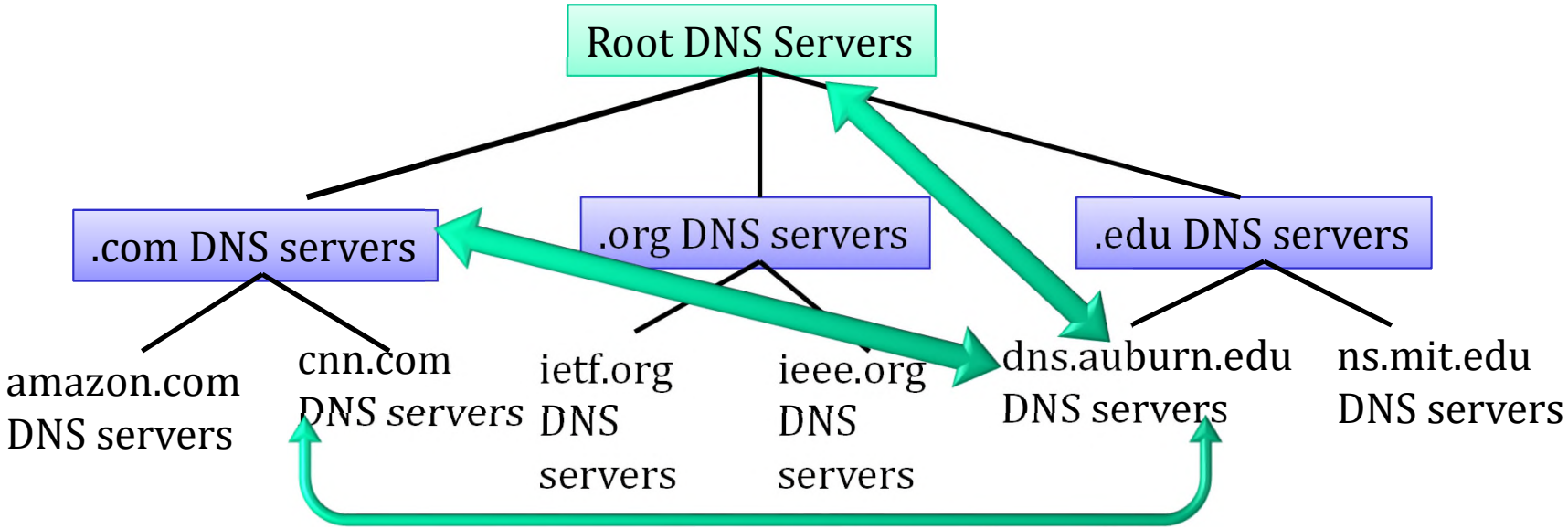
- Domain Name System (DNS)
  - An infrastructure for translating between IP addresses and a name
  - DNS is an application-layer protocol used by hosts to query DNS servers
  - A distributed database implemented in a hierarchy of name servers
  - Distributed DNS is more reliable and faster than centralized DNS
  - DNS is defined in RFC 1034 and RFC 1035

Disadvantages of centralized DNS

- Single point of failure: poor reliability
- RTT is long for a distant centralized database
- Difficult to maintain and update
- Poor performance

Distributed, Hierarchical Database for DNS

- A host needs the IP address of www.cnn.com and queries the local DNS server
- If the local DNS server does not have the RR (resource record) for www.cnn.com in its cache, then the local DNS server
  - Queries a root server to find .com DNS server
  - Queries .com DNS server to get cnn.com DNS server
  - Queries cnn.com DNS server to get IP address for www.cnn.com



DNS: Root name servers

- Contacted by local DNS name server that can not resolve a name
- A local DNS system is pre-configured with the known addresses of the root servers in a file using root hints
  - This file must be updated periodically by the local administrator
- Root name servers:
  - The root name servers know which servers are responsible for the top-level domains (TLD), such as .edu
  - Each top-level domain (such as .edu) has its own set of servers
    - TLD servers in turn delegate to the name servers responsible for individual domain names (such as ns.mit.edu)
- 13 organizations manage the root DNS servers
- The locations of the root servers are listed in the following table: (source www.root-servers.org)

Operator	Locations	IP Addresses
VeriSign, Inc.	Dulles VA	IPv4: 198.41.0.4 IPv6: 2001:503:BA3E::2:30
Information Sciences Institute	Marina Del Rey CA	IPv4: 192.228.79.201 IPv6: 2001:478:65::53
Cogent Communications	Herndon VA; Los Angeles; New York City; Chicago	192.33.4.12
University of Maryland	College Park MD	128.8.10.90
NASA Ames Research Center	Mountain View CA	192.203.230.10
Internet Systems Consortium, Inc.	**43 sites:** Ottawa; Palo Alto; San Jose CA;bvNew York City; San Francisco; Madrid; Hong Kong; Los Angeles; Rome; Auckland; Sao Paulo; Beijing; Seoul; Moscow; Taipei; Dubai; Paris; Singapore; Brisbane;Toronto; Monterrey; Lisbon; Johannesburg; Tel Aviv; Jakarta; Munich; Osaka; Prague; Amsterdam; Barcelona; Nairobi;Chennai; London; Santiago de Chile; Dhaka; Karachi; Torino; Chicago; Buenos Aires; Caracas; OsloPanama; Quito	IPv4: 192.5.5.241 IPv6: 2001:500:2f::f
U.S. DOD Network Information Center	Columbus OH	192.112.36.4
U.S. Army Research Lab	Aberdeen MD	IPv4: 128.63.2.53 IPv6: 2001:500:1::803f:235
Autonomica/NORDUnet	**31 sites:** Stockholm; Helsinki; Milan; London; Geneva; Amsterdam; Oslo; Bangkok; Hong Kong; Brussels; Frankfurt; Ankara; Bucharest; Chicago; Washington DC; Tokyo; Kuala Lumpur; Palo Alto; Jakarta; Wellington; Johannesburg; Perth; San Francisco; New York; Singapore; Miami; Ashburn (US); Mumbai; Beijing; Manila; Doha	192.36.148.17
VeriSign, Inc.	**41 sites:** Dulles (3 locations), Vienna, Miami, Atlanta, Seattle, Chicago, New York, Los Angeles, Mountain View, San Francisco (2 locations), Dallas (US); Amsterdam (NL); London (UK); Stockholm (2 locations) (SE); Tokyo (JP); Seoul (KR); Bejing (CN); Singapore (SG); Dublin (IE); Kaunas (LT); Nairobi (KE); Montreal, Quebec (CA); Sydney (AU); Cairo (EG); Warsaw (PL); Brasilia, Sao Paulo (BR); Sofia (BG); Prague (CZ); Johannesburg (ZA); Toronto (CA); Buenos Aries (AR); Madrid (ES); Vienna (AT); Fribourg (CH); Hong Kong (HK); Turin (IT)	IPv4: 192.58.128.30 IPv6: 2001:503:C27::2:30
Reseaux IP Europeens - Network Coordination Centre	**17 sites:** London (UK); Amsterdam (NL); Frankfurt (DE); Athens (GR); Doha (QA); Milan (IT); Reykjavik (IS); Helsinki (FI); Geneva (CH); Poznan (PL); Budapest (HU); Abu Dhabi (AE); Tokyo (JP); Brisbane (AU); Miami (US); Delhi (IN); Novosibirsk (RU)	IPv4: 193.0.14.129 IPv6: 2001:7fd::1
Internet Corporation for Assigned Names and Numbers	Los Angeles (US); Miami(US)	IPv4: 199.7.83.42 IPv6: 2001:500:3::42
WIDE Project	**6 sites:** *Tokyo (JP); Seoul (KR); *Paris (FR);	IPv4: 202.12.27.33 IPv6: 2001:dc3::35

TLDs Types

- Country-code TLDs (ccTLDs)
  - There are more than 240 ccTLDs. Examples include .uk, .in, and .jp.
- Sponsored generic TLDs (gTLDs)—specialized domains with a sponsor representing a community of interest
  - These TLDs include .edu, .gov, .int, .mil, .aero, .coop, and .museum.
- Unsponsored generic TLDs (gTLDs)—domains without a sponsoring organization
  - The list of unsponsored gTLDs includes .com, .net, .org, .biz, .info, .name, and .pro.

TLD name servers

- .edu TLD
  - a.gtld-servers.net. 192.5.6.30 2001:503:a83e:0:0:0:2:30
  - c.gtld-servers.net. 192.26.92.30
  - .....
- .com TLD
  - a.gtld-servers.net. 192.5.6.30 2001:503:a83e:0:0:0:2:30
  - b.gtld-servers.net. 192.33.14.30 2001:503:231d:0:0:0:2:30
  - c.gtld-servers.net. 192.26.92.30
  - .....
- .org
  - a0.org.afilias-nst.info. 199.19.56.1 2001:500:e:0:0:0:0:1
  - b0.org.afilias-nst.org. 199.19.54.1 2001:500:c:0:0:0:0:1
  - .....
- .fr TLD
  - a.nic.fr. 192.93.0.129 2001:660:3005:3::1:1
  - c.nic.fr. 192.134.0.129 2001:660:3006:4:0:0:1:1
  - .....

Source: http://iana.org/domains/root/db/

TLD and Authoritative Servers

- Top-level domain (TLD) servers
  - Responsible for .com, .edu, .org, .net, .gov, etc, and all top-level country domains uk, fr, ca, jp
  - ICANN (Internet Corporation for Assigned Names and Numbers) is responsible for managing the assignment of domain names and IP addresses
  - Network Solutions maintains servers for .com TLD
  - Educause maintains servers for .edu TLD
- Authoritative DNS servers
  - Such as ns.mit.edu
  - Organization's DNS servers, providing authoritative hostname to IP mappings for organization's servers (e.g., Web, and mail)
  - Can be maintained by organization or service provider
  - Primary and secondary (may be more than one)

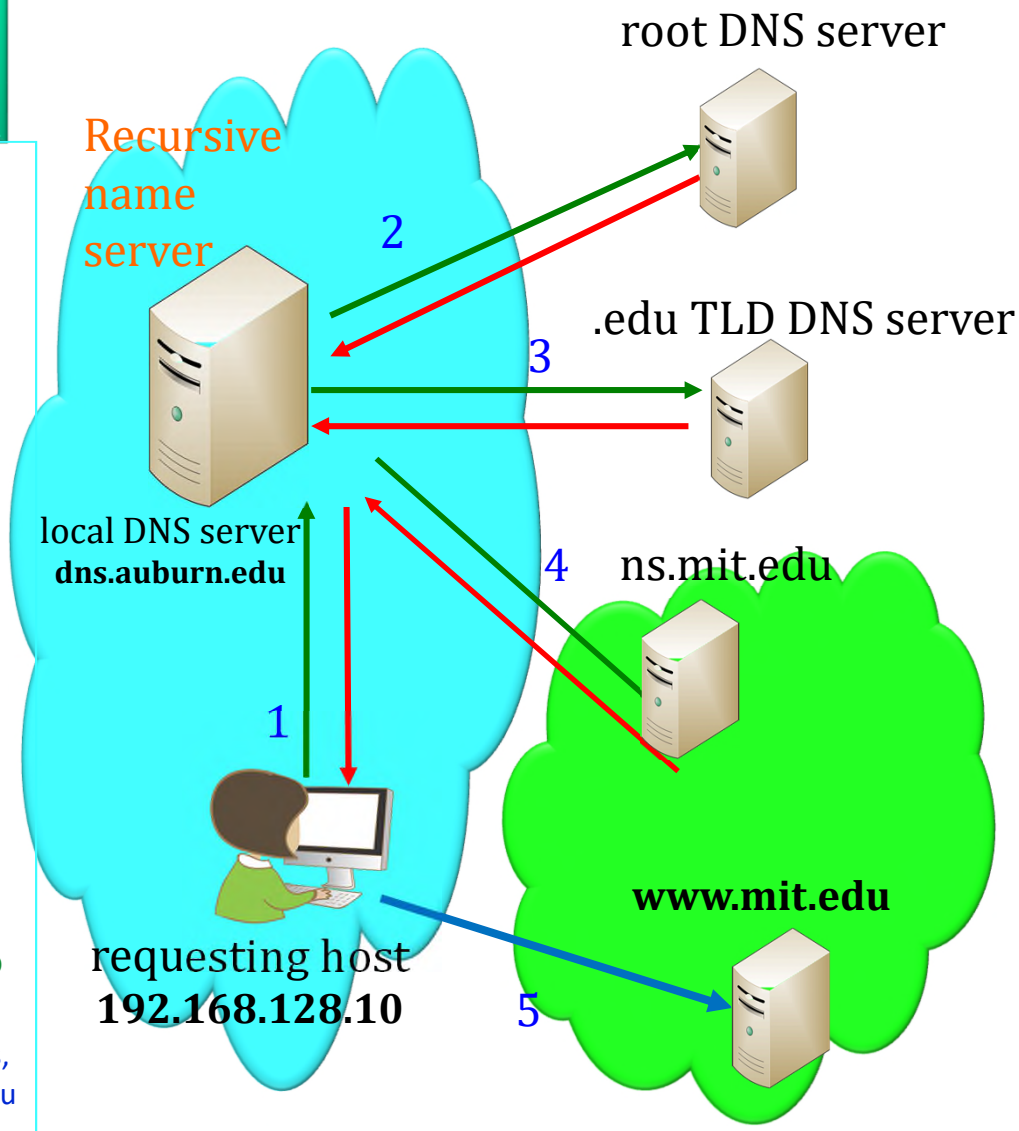
Outline

Part 1

- DNS
  - Overview
  - Recursive and Iterative Queries
  - Recursive or Caching DNS Server
  - Resource record and DNS query
  - DNS Protocol
  - Reverse DNS lookup
- Active Directory (AD)
  - Overview and applications of AD
  - Structure and trust
  - AD objects
  - SRV RR
  - The Open Directory (OD)

DNS queries

- A host at auburn.edu wants IP address for mit.edu
  - If the RR is not in the cache of the local DNS server, then the local DNS server will carry out the recursive query for the local client
- Recursive query
  - Arrow 1
  - DNS.auburn.edu performs recursive query for the host
    - dns.auburn.edu plays the role of a recursive/caching name server
    - Only serving recursive query for hosts in the same domain in order to reduce the load
- Iterated (non-recursive) query
  - Arrows 2, 3, and 4
  - E.g., root DNS replies to dns.auburn.edu to contact .edu TLD DNS
    - Root DNS says, "I do not know the IP address, but ask .edu TLD DNS server who will help you out!"
  - ns.mit.edu plays the role of authoritative name server



Authoritative DNS servers (1)

- Authoritative name servers:
  - An authoritative answer to a DNS query
    - Cache name server provides a cached answer that was given by another name server
  - Primary DNS server, also known as master server, contains the original set of data
  - Secondary or slave name server contains data copies usually obtained from synchronization with the master server
  - It is recommended that three servers be provided for most organizations (in RFC 2182)
- The IP addresses of authoritative DNS servers are maintained by ICANN and kept in TLD DNS servers
- All Authoritative name server are initially treated equally
- Resolvers often measure the performance of the various servers, choose the server with the best performance for most queries

Authoritative DNS servers (2)

- Each ISP, company, university, organization has at least one default name server
- When host makes a DNS query,
  - Query is sent to its local authoritative DNS server or a recursive server
  - DNS server acts as proxy, forwards query into the DNS hierarchy: recursive query
- The DNS information for one domain name is stored as resource record(s) (RR's)
- A DNS zone is a portion of the global Domain Name System (DNS) namespace for which administrative responsibility has been delegated

Outline

Part 1

- DNS
  - Overview
  - Recursive and Iterative Queries
  - Recursive or Caching DNS Server
  - Resource record and DNS query
  - DNS Protocol
  - Reverse DNS lookup
- Active Directory (AD)
  - Overview and applications of AD
  - Structure and trust
  - AD objects
  - SRV RR
  - The Open Directory (OD)

DNS Resolver

- Inside a host, a process called DNS resolver obtains the mapping from name to IP address
  - RESOLVERS are programs that obtain information from name servers in response to client requests
  - A cache preserves a mapping for certain amount of time
  - A DNS resolver can be running inside a computer that is
    - A client computer
    - A web server, mail server, etc.
    - A DNS server
- Resolvers must have access to at least one name server
  - Use that name server's information to answer a query directly
  - Perform the query using referrals to other name servers

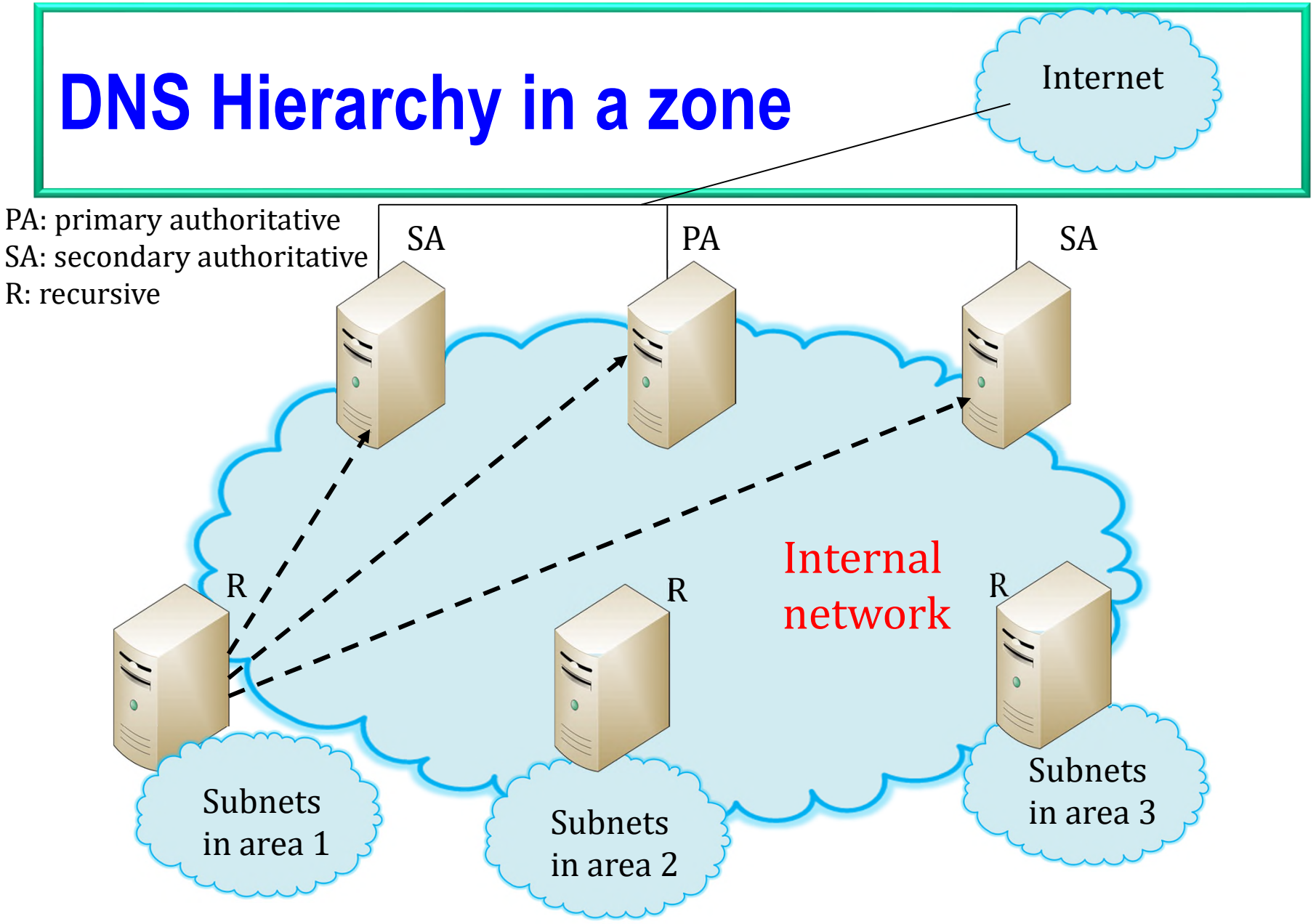
Caching Name Server/Recursive Name Server

- The terms recursive server and caching server are often used synonymously as in BIND (Berkeley Internet Name Domain)
- Typical implementation:
  - Move the resolver function out of the local machine and into a name server which supports recursive queries
    - Produces an easy method of providing domain service in a PC which lacks the resources to perform the resolver function
    - Centralizes the cache for a whole local network
  - Each PC must have a list of name server addresses that will perform the recursive requests
  - A router that connects a home network to DSL/cable modem provides caching/recursive name service
    - E.g. 192.168.1.1 is the LAN interface that provides caching DNS
    - E.g. 192.168.x.1
    - Some vendors refer to it as DNS relay

Forwarder and Firewall

- A caching name server does not necessarily perform the complete recursive lookup itself
  - Instead, it can forward some or all of the queries that it cannot find from its cache to another caching name server, commonly referred to as a forwarder
  - Caching servers unable to pass packets through the firewall would forward to the server that can traverse the firewall, and that server would query the Internet DNS servers on the internal server's behalf
  - Ref: BIND9 manual, http://www.bind9.net/manual/bind/9.3.2/Bv9ARM.ch01.html#id2546254

DNS Hierarchy in a zone



DNS Reliability and Security

- Mitigate risk by duplicating the DNS function of an organization onto two servers, one primary and one secondary, so that if the one goes down, DNS is still available
- Split DNS
  - Only let local users that are part of the domain query a private DNS server (or stealth DNS server) to ensure confidentiality of resource naming conventions and other sensitive information only available to internal hosts
  - Set up a public DNS server outside the firewall or in a DMZ for outsiders to learn the IP addresses for a web server or mail server
  - From a security perspective, only publish a mapping to the public domain when it is necessary
  - Split Horizon is normally used to describe a DNS server that will give different responses based on the source IP address, or some other characteristic

DNS protocol and port

- DNS uses UDP port 53 for lookups and transfers
  - This port must be opened to the VPN through the firewall if a remote user needs to use the internal/private DNS for lookups
  - Note: this decision will be defined in the planning phase and should be used with a virtual private network (VPN)
- TCP port 53 comes into play only when the response data size exceeds 512 bytes, or for tasks such as a zone transfer
  - Zone transfer: from primary authoritative to secondary authoritative server

DNS: caching and updating records (1)

- Once a name server learns mapping, it caches mapping
  - Cache entries timeout after some time
  - TLD servers are typically cached in a local name server (such as dns.auburn.edu)
    - A local name server can be an authoritative or recursive name server
    - Thus root and TLD name servers not often visited
  - Update/notify mechanisms in RFC 2136
- Caching/Recursive Servers:
  - If a server is going to provide caching services, then it must provide recursive queries
  - Recursive queries need access to the root servers which is provided via the 'type hint' statement: root servers' IP addresses are in a file
    - A caching server using BIND will typically have a named.conf file which includes

```
type hint;
file "root.servers";
```

DNS: caching and updating records (2)

- For Windows DNS server:
  - A root hints file, Cache.dns, that is stored in the systemroot\System32\Dns folder on the server computer
  - The contents of this file are preloaded into server memory when the service is started and contain pointer information to root servers for the DNS namespace
- Caching/Recursive Servers:
  - To create a caching-only name server, install the DNS service but do not configure any zones
  - Configure client computer's TCP/IP properties to use the caching-only DNS server for name resolution
  - Provide DNS name resolution for computers in the same domain
  - Cache the result to answer potential future queries within a certain expiration (time-to-live) period
  - Servers with Recursion Access Control provide control over which hosts are permitted to use DNS recursive lookups

Outline

Part 1

- DNS
  - Overview
  - Recursive and Iterative Queries
  - Recursive or Caching DNS Server
  - Resource record and DNS query
  - DNS Protocol
  - Reverse DNS lookup
- Active Directory (AD)
  - Overview and applications of AD
  - Structure and trust
  - AD objects
  - SRV RR
  - The Open Directory (OD)

RR Format

- (www.auburn.edu, 131.204.2.251, A)
- Type (16 bit): A

RR format: **(name, [pref.], value, type, [ttl])**

- www.auburn.edu. IN A 131.204.2.251
- www IN A 131.204.2.251
- Class 16 bit: IN which identifies a protocol family or instance of a protocol is the Internet system

RR format: **name [ttl] [Class] Type [pref.] value**

DNS resource record (RR) Type (1)

RR format: **(name, [pref.], value, type, [ttl])**

- Type=A
  - Name is host's name
  - Value is IP address
- Type=NS
  - Name is domain name (e.g. auburn.edu)
  - Value is name of authoritative name server for this domain (e.g. dns.auburn.edu)
- Type=MX
  - Name is domain name (e.g. auburn.edu)
  - Value is name of mail server designed for the domain (e.g. aumail.duc.auburn.edu)
  - A preference value is designated for each mail server if there are multiple MX RR's in a domain

DNS resource record (RR) Type (2)

RR format: **(name, [pref.], value, type, [ttl])**

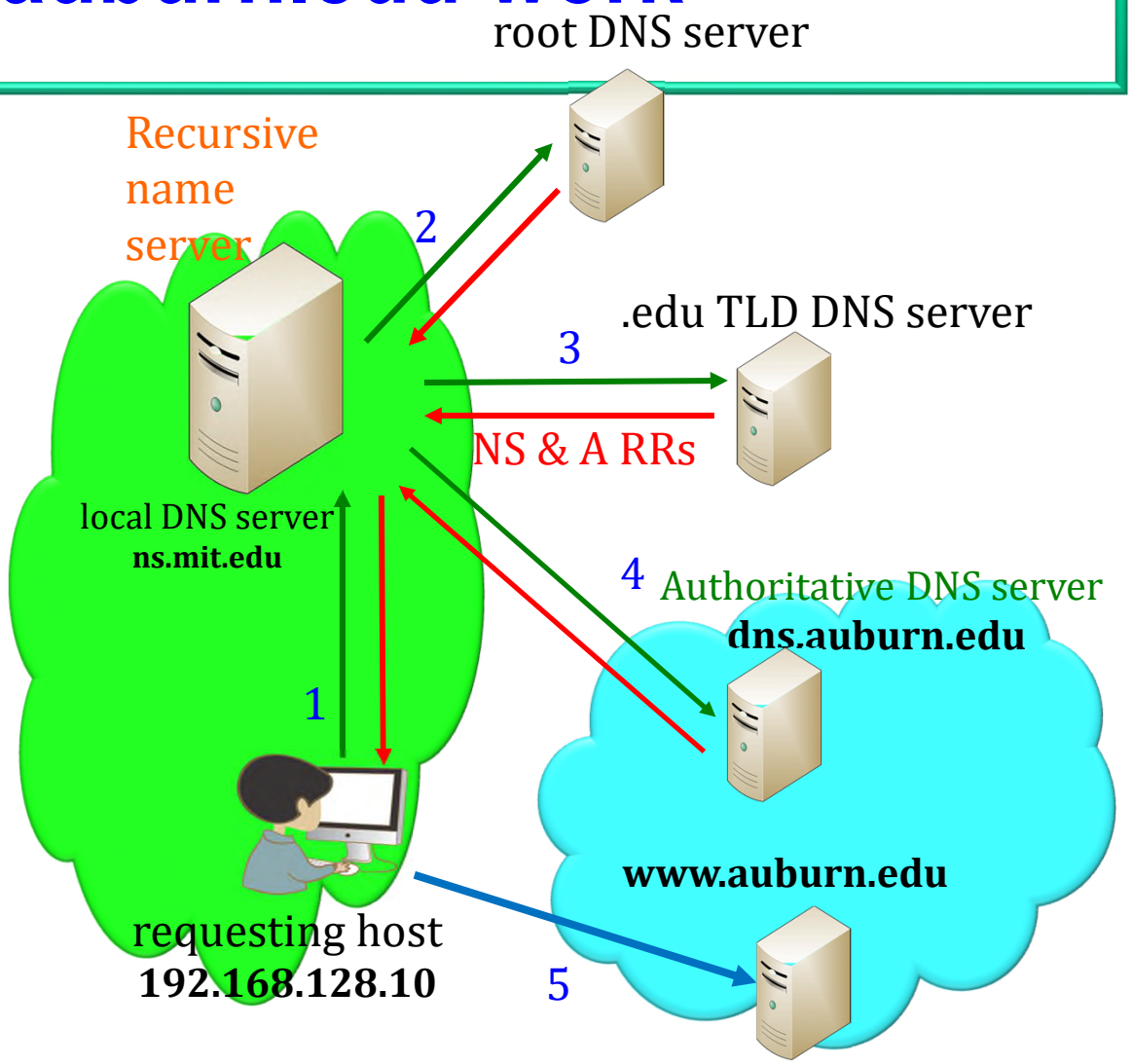
- Type=CNAME
  - Name (such as www.ibm.com) is alias name for “canonical” (real) name
  - Value is canonical name (such as servereast.backup2.ibm.com)
  - www.ibm.com (name) is really servereast.backup2.ibm.com (value)
- Type=AAAA
  - IPv6 host address (AAAA) resource record
  - Maps a DNS domain name to an Internet Protocol (IP) version 6 128-bit address
- TTL: time to live in cache
  - 32 bit integer for the number of seconds

RR example

- Company x has a webserver w.x.com with IP address 131.204.2.5
- General public uses www.x.com or x.com to access the website
- One Type A RR for the host (w.x.com, 131.204.2.5, A, 3 hours)
- One Type CNAME RR for aliasing (www.x.com, w.x.com, CNAME, 3 hours)
- One Type CNAME RR for aliasing (x.com, w.x.com, CNAME, 3 hours)

How to make auburn.edu work

- .edu TLD server contains
  - 3 NS RR's
  - 3 A RR's
- dns.auburn.edu: authoritative name server contains
  - 1 A RR for web server
  - 1 MX RR for mail server
  - 1 A RR for mail server



Inserting records into DNS (1)

- Example: auburn.edu
- Auburn University registers name auburn.edu at DNS registrar (e.g., Educause)
  - Provide names, IP addresses of authoritative name server (primary and secondary)
  - Registrar inserts six RRs (3 authoritative DNS servers) into edu TLD server:

```
(auburn.edu, dns.auburn.edu, NS)
(dns.auburn.edu, 131.204.41.3, A)
(auburn.edu, dns.eng.auburn.edu, NS)
(dns.eng.auburn.edu, 131.204.10.13, A)
(auburn.edu, dns.duc.auburn.edu, NS)
(dns.duc.auburn.edu, 131.204.2.10, A)
```

Inserting records into DNS (2)

- Auburn University created 3 RR's in authoritative server dns.auburn.edu inside auburn.edu domain
  - Type A RR for www.auburn.edu
    - `(www.auburn.edu, 131.204.2.251, A)`
  - Type MX (mail exchange) RR for aumail.duc.auburn.edu
    - `(auburn.edu, 10 aumail.duc.auburn.edu, MX)`
    - `(aumail.duc.auburn.edu, 131.204.2.83, A)`
  - When multiple mail servers are available, each server has a type MX RR and one type A RR
  - Preference = 10 as default value for mail server
  - When multiple MX RR available, mail server with smallest Preference value is used
  - No CNAME RR for multiple mail servers
- For a small organization, one can put RR's for authoritative servers in an ISP's DNS server hosting the web and mail service for the organization

MX and CNAME: Microsoft's recommendation (1)

- DNS MX Records and CNAMEs: Article ID: 153001 (http://support.microsoft.com/kb/153001)
  - RFC 1123 explicitly states that SMTP mail should be addressed to canonical name hosts
  - To be canonical, the DNS entry must be an A record or an MX record
  - CNAME records are not canonical and should not be mixed with MX records
  - Domain Name System (DNS) entries for Mail Exchanger (MX) records can be pointed to canonicalized (CNAME) host records
  - However, doing so is not advised
  - Increased administrative overhead and the possibility of misrouted messages can result

MX and CNAME: Microsoft's recommendation (2)

- Microsoft recommends that Mail/DNS administrators always link MX records to fully qualified principal names or domain literals
- The Internet Mail Connector (IMC) uses DNS to resolve Internet Protocol (IP) addresses when sending mail
  - A sending Simple Mail Transfer Protocol (SMTP) server also uses DNS to determine which host on the destination network is appropriate to receive mail
  - To determine mail hosts, the sending server checks for an MX record
  - Next, the sending server resolves the MX record to an IP address by checking for an address record (A record)
  - If an A record is found, the address is fully canonicalized and mail can be delivered

MX and CNAME: Microsoft's recommendation (3)

- If an alias record (CNAME) is used for the hostname listed in the MX record, the sending host might re-write the envelope and redirect the RCPT command to the alias hostname and not the original recipient
  - This might cause the destination SMTP host to reject the message.
  - Example:

    auburn.edu. MX 10 mail.auburn.edu.

    mail.auburn.edu. IN CNAME server.auburn.edu.
  - When a mail is sent to "admin@auburn.edu" with the above configuration, the sending host might detect the fact that the "mail.auburn.edu" is an alias and rewrite the RCPT-TO command to "server.auburn.edu".
  - Thus, the mail envelope written during SMTP mail transmission might be changed to admin@server.auburn.edu

MX and CNAME: Microsoft's recommendation (4)

- If the mail system isn't configured to accept mail for "server.auburn.edu" the message may be returned as undeliverable
- This issue can be difficult to detect since the body of the message with the TO: line is left unchanged
- Desired Configuration:

  auburn.edu. MX 10 mail.auburn.edu.

  mail.auburn.edu. IN A 131.204.12.17
- In the above example, the MX record resolves directly to an IP address
- This causes the sending host to realize that the resolved address is canonical and the final destination

Zone file

```
; zone file for auburn.edu
; zone file name master.localhost
$TTL 2d    ; Two days or 172800 seconds as the default TTL for zone
$ORIGIN auburn.edu.
@             IN      SOA   dns.auburn.edu. master.auburn.edu. (
                        2003080800 ; serial number
                        12h        ; refresh (h: hour)
                        15m        ; update retry (m: minute)
                        3w         ; expiry (w: week)
                        3h         ; minimum
                        )
              IN      NS      dns.auburn.edu.
              IN      MX  10  aumail.duc.auburn.edu.
dns           IN      A   131.204.10.13
webserver     IN      A   131.204.2.251
aumail.duc    IN      A   131.204.2.83
www           IN      CNAME   webserver.auburn.edu.
@             IN      CNAME   webserver.auburn.edu.
```

Fully Qualified Domain Name (FQDN)

- A zone file consists of Comments, Directives and Resource Records
  - Comments start with ';' (semicolon) and are assumed to continue to the end of the line
    - Comments can occupy a whole line or part of a line
  - Directives start with '$' and are standardized, such as $ORIGIN and $TTL
    - The $TTL directive should be present, appear before the first RR (RFC 2308 implemented in BIND 9) and defines the default RR TTL value
    - $ORIGIN defines the base name (aka label) to be used for *unqualified* name substitution
      - If there is a dot at the end of a name in a resource record or directive, the name is *qualified*
      - if it contains the whole name including the host then it is a *Fully Qualified Domain Name* (FQDN)
      - In this case the name as it appears in the RR is used unchanged
      - E.g. `dns.auburn.edu.  IN  A  131.204.10.13`

Unqualified name

- If there is NO dot at the end of the name, the name is unqualified and DNS software adds the value of the $ORIGIN directive
  - For example, the type A RR of

    dns IN A 131.204.10.13
  - is expanded to

    dns.auburn.edu. IN A 131.204.10.13
- The symbol @ forces substitution of the current (or synthesized) value of $ORIGIN.
  - The @ symbol is replaced with the current value of $ORIGIN. For example,

    @ IN CNAME webserver.auburn.edu.
  - becomes

    auburn.edu. IN CNAME webserver.auburn.edu.

SOA: Start of Authority (1)

- The first Resource Record must be the SOA (Start of Authority) record
  - The SOA defines global parameters for the zone (domain)
  - There is only one SOA record allowed in a zone file
  - The master.auburn.edu. represents the email address of master@auburn.edu
- The generic format is described below:
  - The serial number
    - An unsigned 32 bit value in range 1 to 4294967295 with a maximum increment of 2147483647
    - In BIND implementations this is defined to be a 10 digit field
    - This value must increment when any resource record in the zone file is updated

SOA: Start of Authority (2)

- Refresh
  - A signed 32 bit time value in seconds and indicates the time when the slave will try to refresh the zone from the master (by reading the master DNS SOA RR)
- Retry
  - A signed 32 bit value in seconds and indicates the time between retries
  - If the slave (secondary) fails to contact the master when refresh has expired
- Expire
  - How long a secondary will still treat its copy of the zone data as valid if it cannot contact the primary
  - A signed 32 bit value in seconds and indicates when the zone data is no longer authoritative

SOA: Start of Authority (3)

- Minimum
  - Signed 32 bit value in seconds
    - The default TTL (time-to-live) for resource records
  - RFC 2308 (implemented by BIND 9) redefined this value to be the negative caching time
    - The time a NAME ERROR = NXDOMAIN (the domain name is not defined) result may be cached by any resolver
      - Negative caching provides the caching of the non-existence of an RR or domain name
    - The maximum value allowed by RFC 2308 for this parameter is 3 hours

BIND 9 named.conf

- BIND 9 needs
  - A standard resolver (Caching-only DNS Server) config. file: named.conf
  - Zone file: master.localhost,
  - Other files: localhost.rev and root.servers
- Listen-on defines the port and IP address(es) on which BIND will listen for incoming queries
  - The default is port 53 on all server interfaces

named.conf file

```
options {
  directory "C:\Windows\system32\dns\etc";
  version “BIND 9";
  recursion yes;
  allow-recursion {131.204.0.0/16;};
listen-on {131.204.10.13;};
};

zone "." {
  type hint;
  file "root.servers";
};

zone “auburn.edu" in{
  type master;
  file "master.localhost";
  allow-update{none;};
};
zone "0.0.127.in-addr.arpa" in{
  type master;
  file "localhost.rev";
  allow-update{none;};
};
```

Hint and root.servers

- When a name server cannot resolve a query it uses the file root.servers
  - The file root.servers defines a list of name servers (a.root-servers.net - m.root-servers.net) where BIND can get a list of TLD servers for the particular TLD e.g. .com
  - The root.servers file can be obtained from ICANN
    - Using anonymous FTP for file /domain/named.root on server ftp.internic.net or rs.internic.net
- The root server file is defined using a normal zone clause with type hint as outlined in the following example:
  - The dot (".") zone identifies the DNS server as a root server

```
;
. 3600000 IN NS A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET. 3600000 A 198.41.0.4
;
. 3600000 NS B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET. 3600000 A 192.228.79.201
…….
```

localhost.rev file

- The localhost.rev file maps the IP address 127.0.0.1 to the name 'localhost'
  - This special zone allows reverse mapping of the loopback address 127.0.0.1 to satisfy applications which do reverse or double lookups.
  - Any request for the address 127.0.0.1 using this name server will return the name localhost
  - The 0.0.127.IN-ADDR.ARPA zone is defined as shown below
- This file should not require modification

```
$TTL 86400 ;
; could use $ORIGIN
0.0.127.IN-ADDR.ARPA.
@ IN SOA localhost.
root.localhost. (
1997022700 ; Serial
3h ; Refresh
15 ; Retry
1w ; Expire
3h ) ; Minimum
IN NS localhost.
1 IN PTR localhost.
```

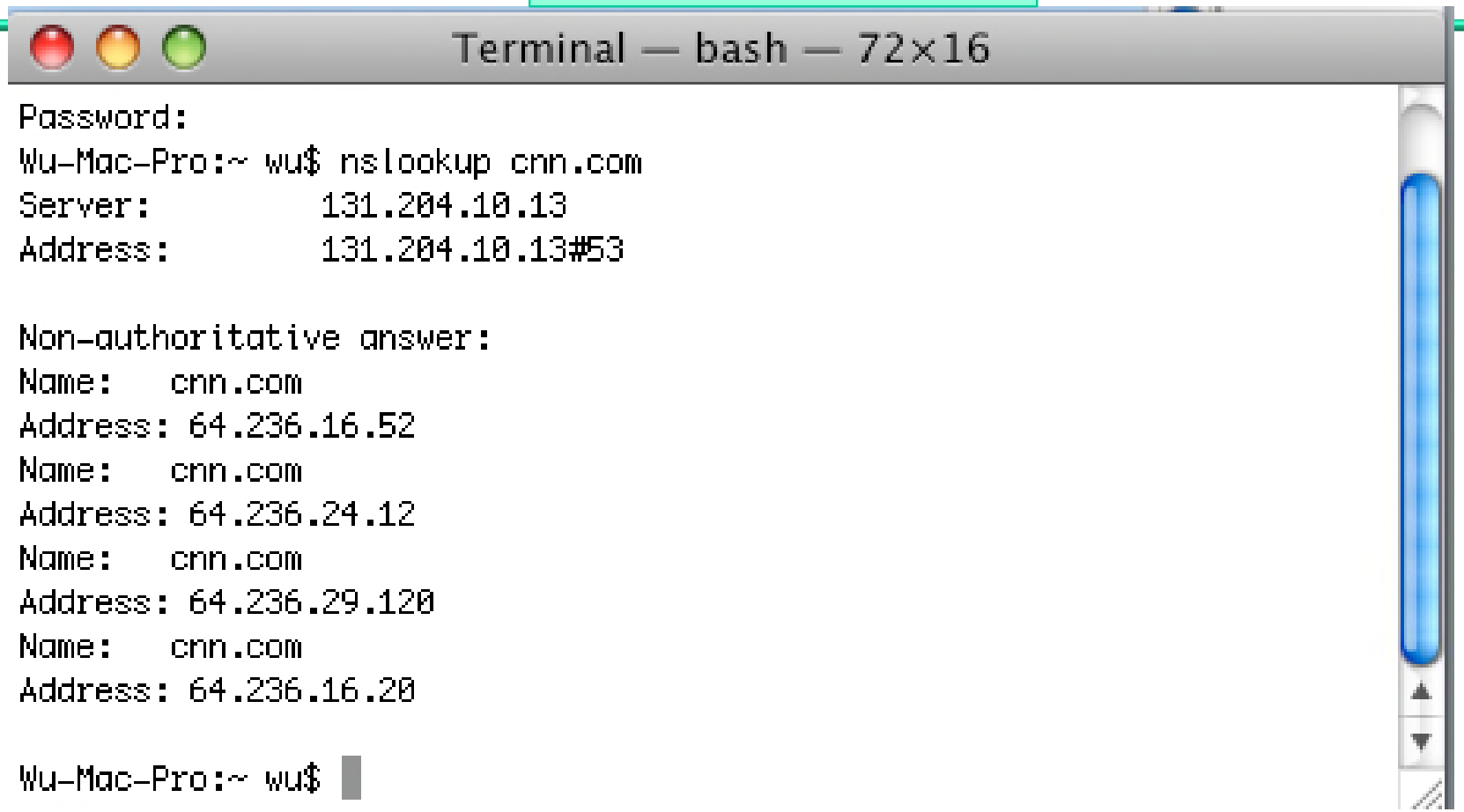
Outline

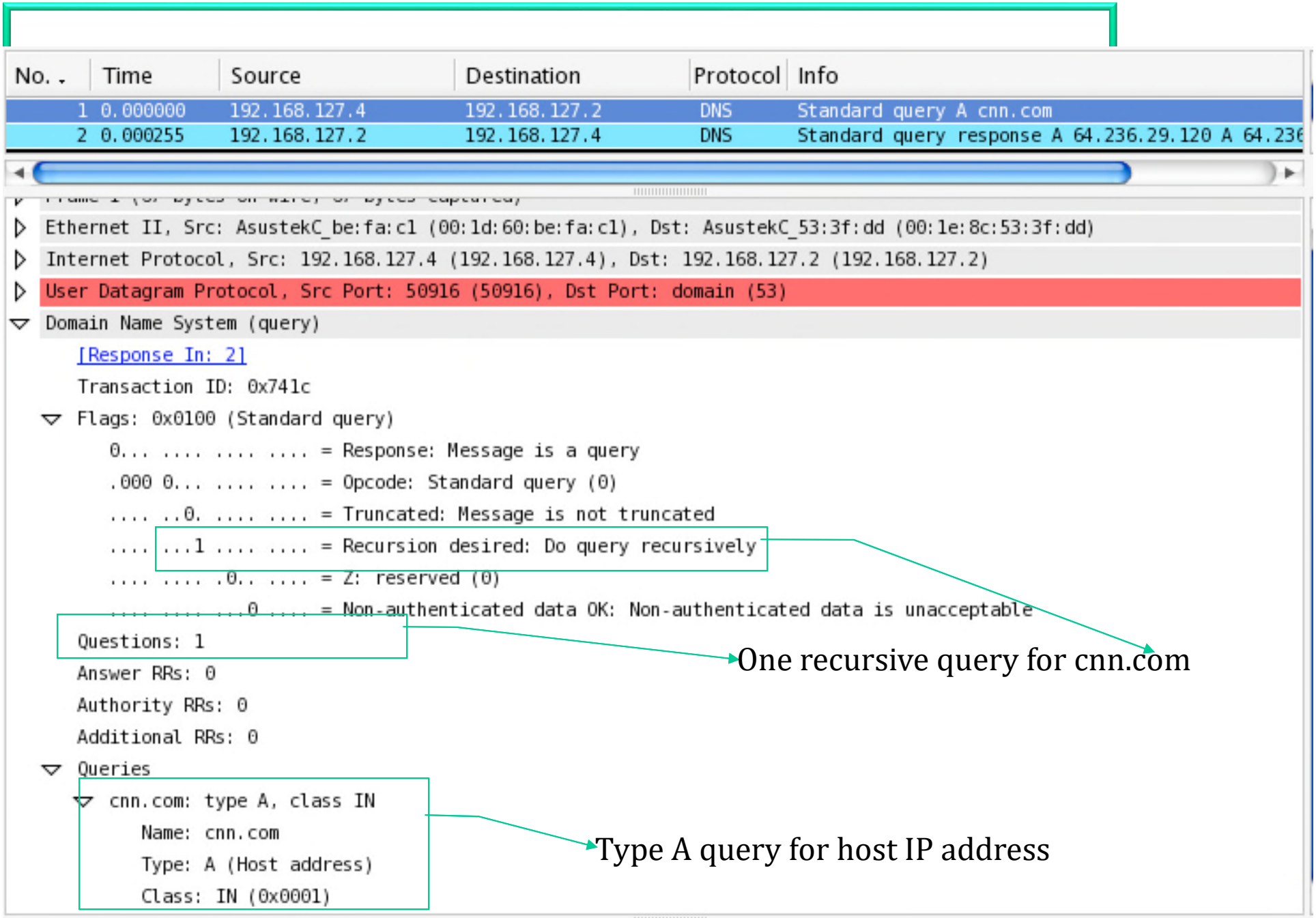
Part 1

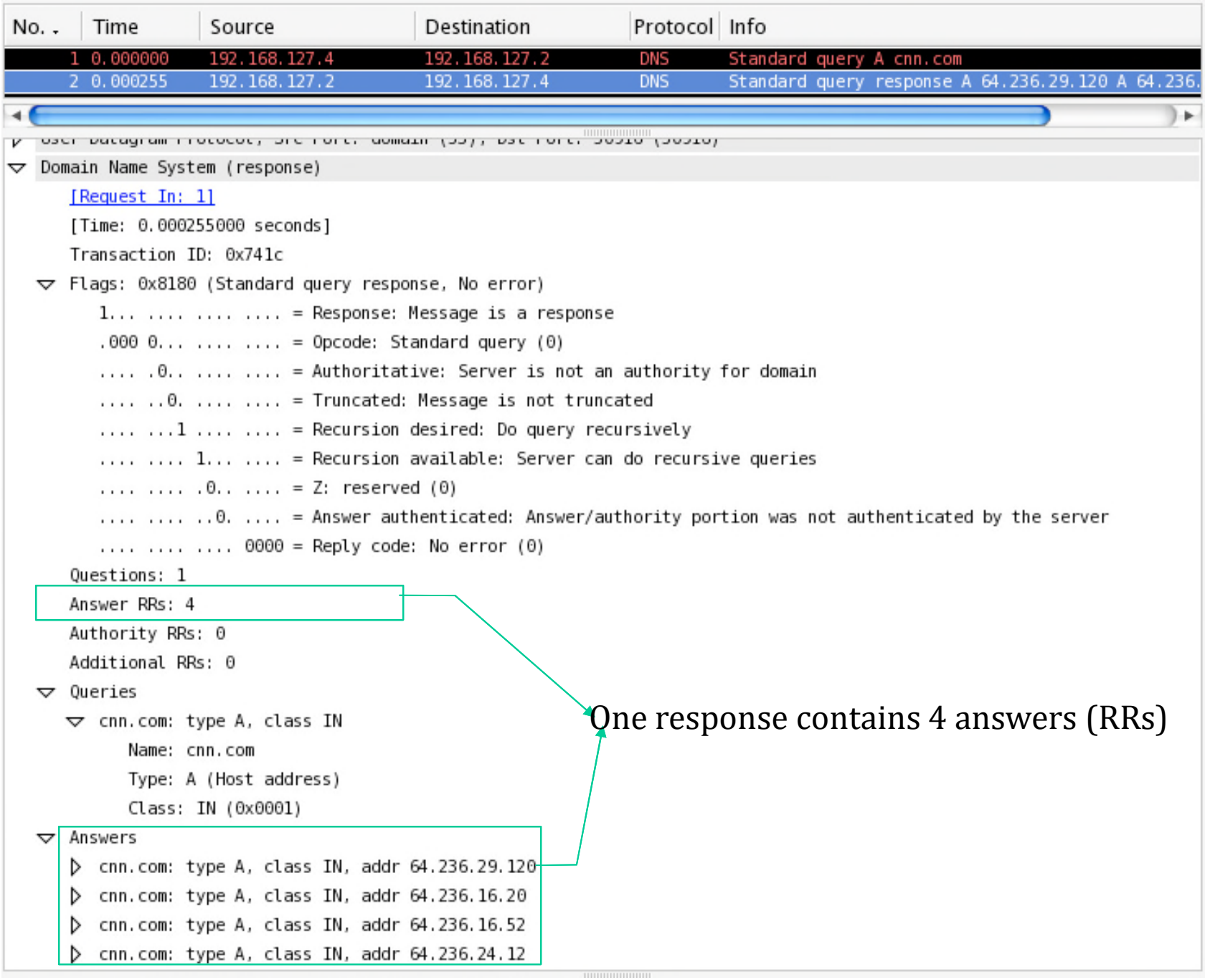
- DNS
  - Overview
  - Recursive and Iterative Queries
  - Recursive or Caching DNS Server
  - Resource record and DNS query
  - DNS Protocol
  - Reverse DNS lookup
- Active Directory (AD)
  - Overview and applications of AD
  - Structure and trust
  - AD objects
  - SRV RR
  - The Open Directory (OD)

nslookup

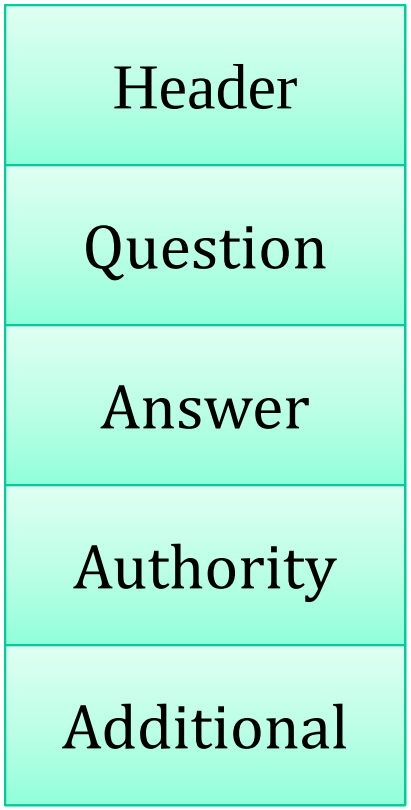
Troubleshooting tool




No. Time Source Destination Protocol Info
1 0.000000 192.168.127.4 192.168.127.2 DNS Standard query A cnn.com
2 0.000255 192.168.127.2 192.168.127.4 DNS Standard query response A 64.236.29.120 A 64.236
Ethernet II, Src: AsustekC_be:fa:c1 (00:1d:60:be:fa:c1), Dst: AsustekC_53:3f:dd (00:1e:8c:53:3f:dd)
Internet Protocol, Src: 192.168.127.4 (192.168.127.4), Dst: 192.168.127.2 (192.168.127.2)
User Datagram Protocol, Src Port: 50916 (50916), Dst Port: domain (53)
Domain Name System (query)
[Response In: 2]
Transaction ID: 0x741c
Flags: 0x0100 (Standard query)
0... .... = Response: Message is a query
.000 0... .... = Opcode: Standard query (0)
.... ..0. .... = Truncated: Message is not truncated
....1 .... = Recursion desired: Do query recursively
.... .0.. .... = Z: reserved (0)
....0 .... = Non-authenticated data OK: Non-authenticated data is unacceptable
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
cnn.com: type A, class IN
Name: cnn.com
Type: A (Host address)
Class: IN (0x0001)
One recursive query for cnn.com
Type A query for host IP address


No. | Time | Source | Destination | Protocol | Info
1 0.000000 192.168.127.4 192.168.127.2 DNS Standard query A cnn.com
2 0.000255 192.168.127.2 192.168.127.4 DNS Standard query response A 64.236.29.120 A 64.236.
Domain Name System (response)
[Request In: 1]
[Time: 0.000255000 seconds]
Transaction ID: 0x741c
Flags: 0x8180 (Standard query response, No error)
1... .... = Response: Message is a response
.000 0... .... = Opcode: Standard query (0)
.... .0.. .... = Authoritative: Server is not an authority for domain
.... ..0. .... = Truncated: Message is not truncated
....1 .... = Recursion desired: Do query recursively
.... 1... .... = Recursion available: Server can do recursive queries
.... .0.. .... = Z: reserved (0)
....0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
.... 0000 = Reply code: No error (0)
Questions: 1
Answer RRs: 4
Authority RRs: 0
Additional RRs: 0
Queries
cnn.com: type A, class IN
Name: cnn.com
Type: A (Host address)
Class: IN (0x0001)
Answers
cnn.com: type A, class IN, addr 64.236.29.120
cnn.com: type A, class IN, addr 64.236.16.20
cnn.com: type A, class IN, addr 64.236.16.52
cnn.com: type A, class IN, addr 64.236.24.12
One response contains 4 answers (RRs)

DNS Message format



RFC 1035

The question for the name server

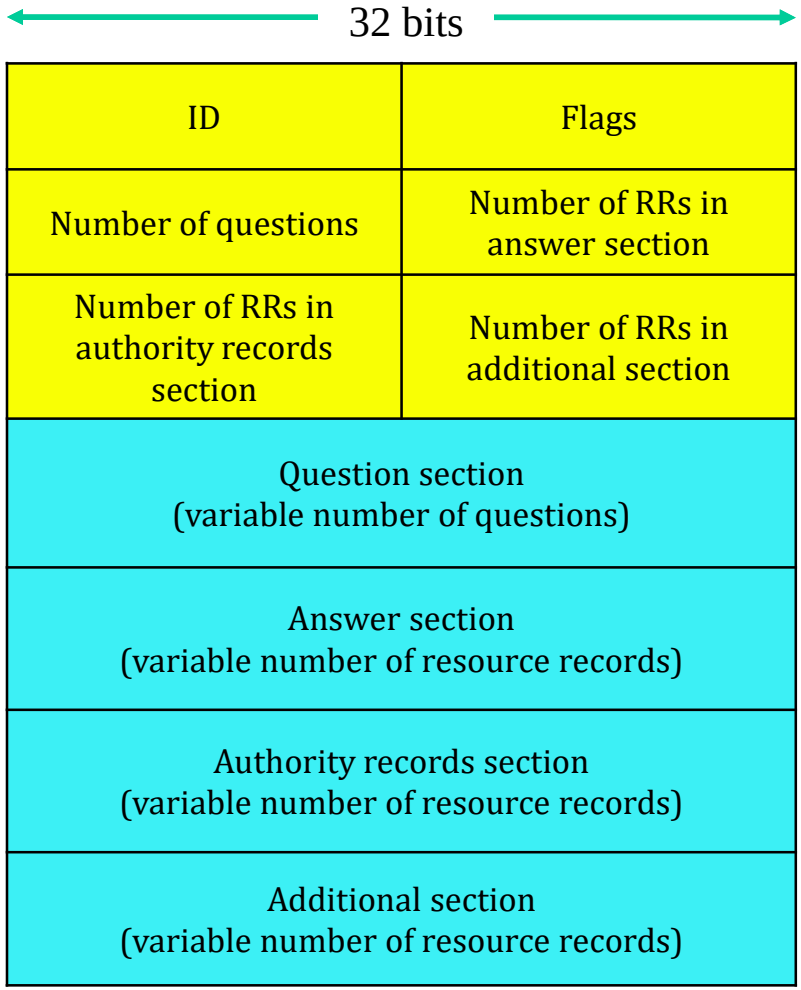
RRs answering the question

RRs pointing toward an authority

RRs holding additional information

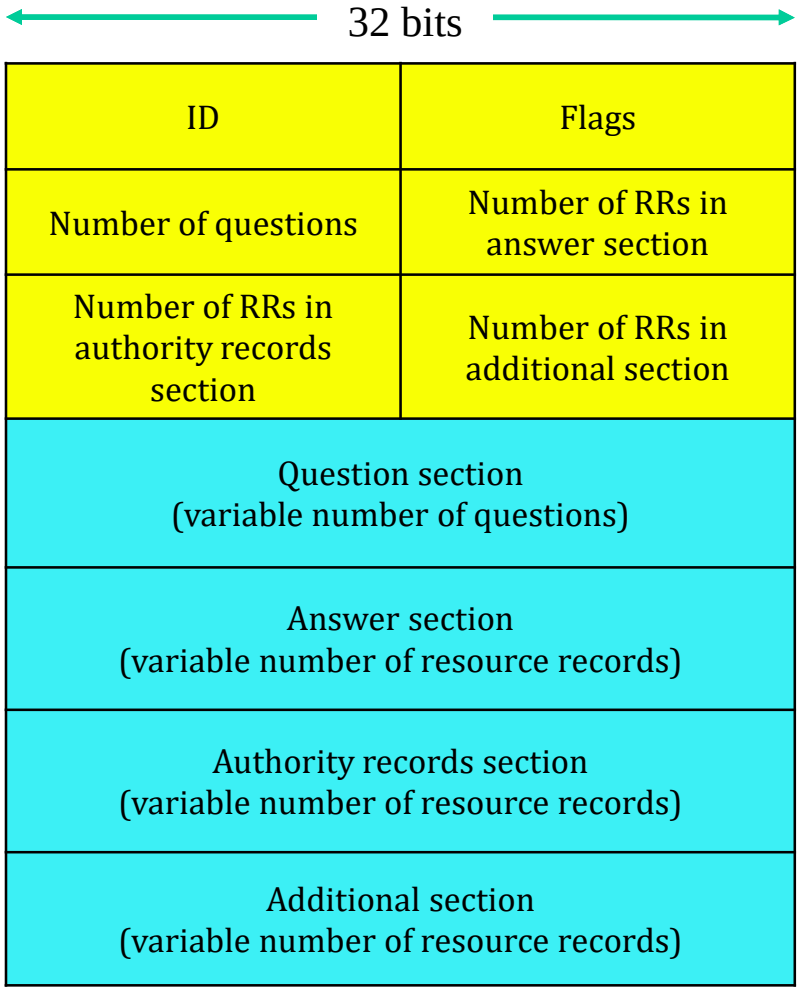
DNS protocol and message format (1)

- DNS query and reply messages both in same message format
- Message header
  - ID: 16 bit # for query and reply
  - A particular query and reply use same ID #
  - Flags:
    - QR bit: query (=0) or reply (=1)
    - RD bit: recursion desired
    - RA bit: recursion available
    - AA bit: Authoritative Answer
    - RCODE (Response code): 4 bit
      - Code = 3: Name Error (no existent domain name, etc.)
      - Code = 0: no error

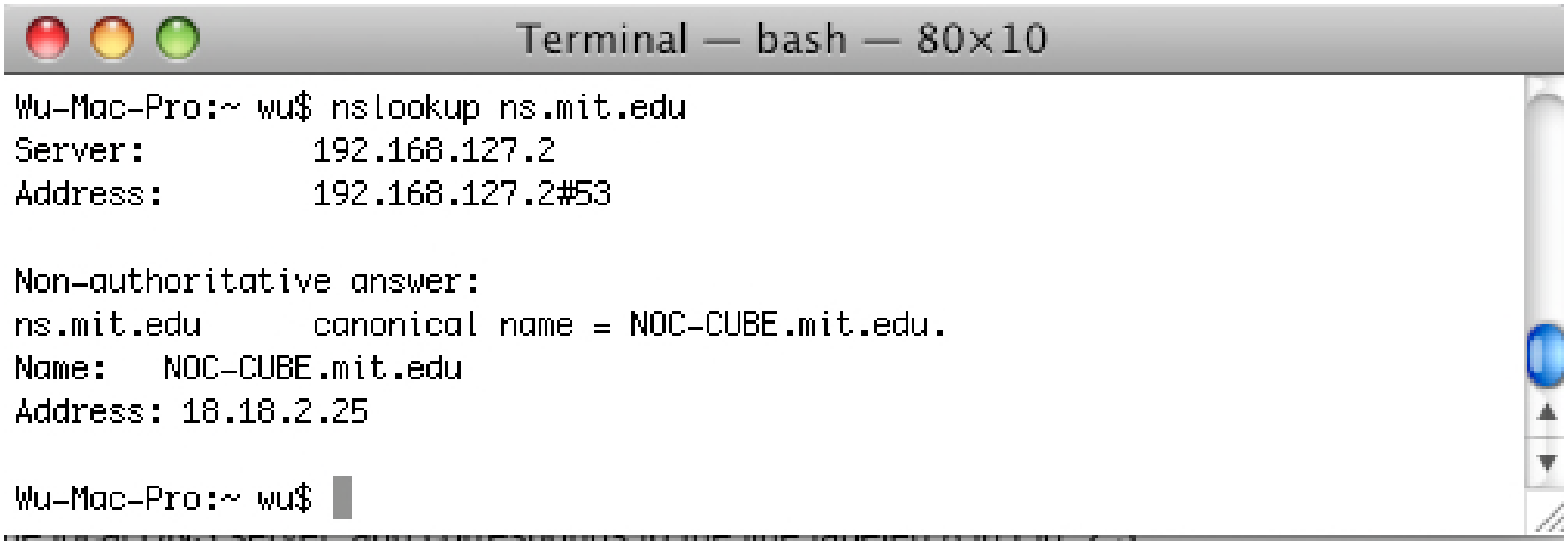


DNS protocol and message format (2)

- Question
  - Name and type for a query
- Answer
  - RRs as the answer
- Authority records
  - Resource records point toward another authoritative name server
  - Non-recursive reply contains no answer and delegates to another DNS server
- Additional
  - Additional "helpful" RR, e.g. suggestion to ask another DNS server (plus server's IP address) that may have answer



Find the IP address of a name server



mit.edu has a name server with an alias name of ns.mit.edu, but the real name of the DNS server (canonical name) is NOC-CUBE.mit.edu

Query auburn.edu's mail RR

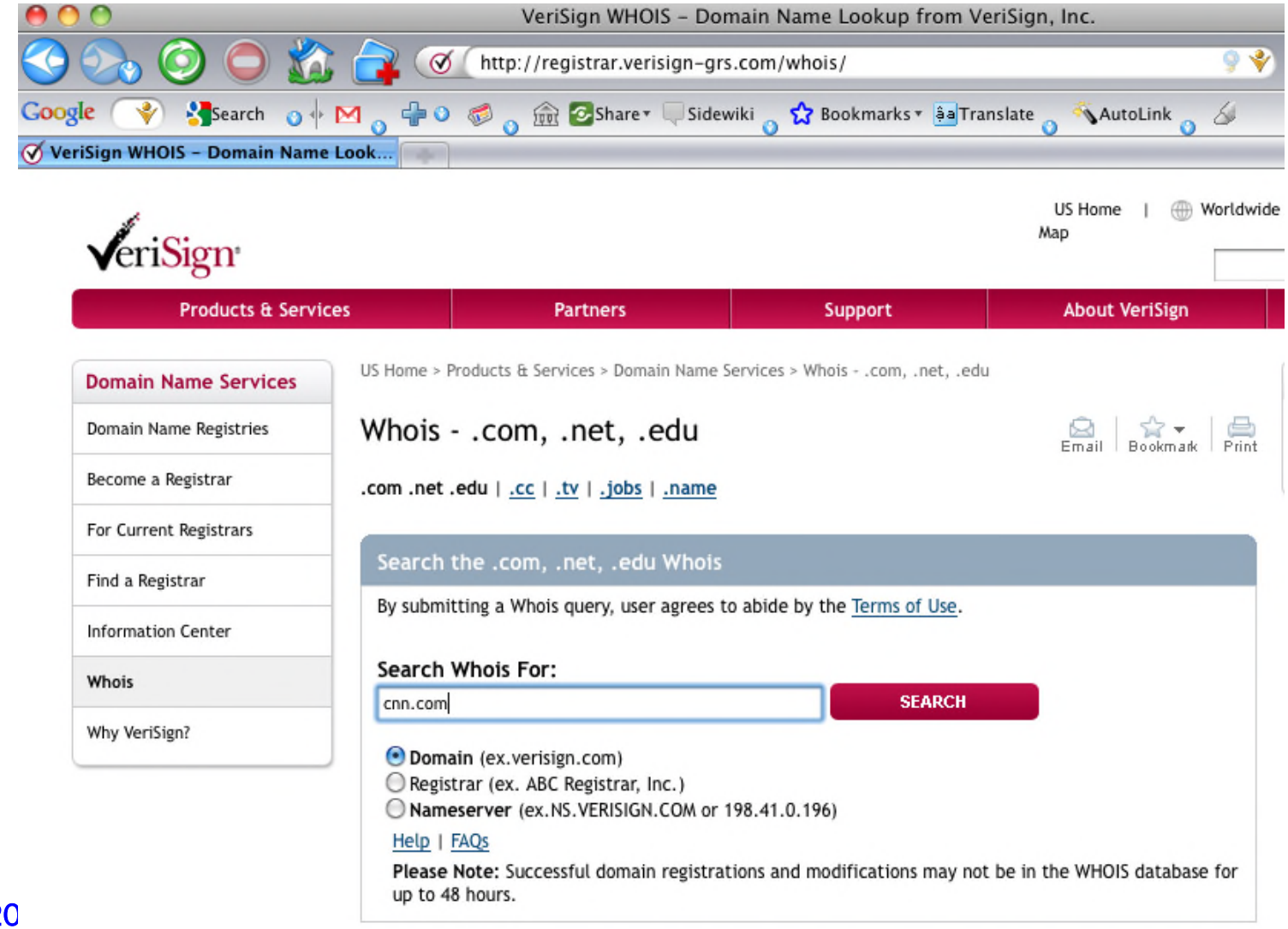
- Set q=mx

```
Mac-Pro:~ wu$ nslookup
> 131.204.10.13
Server:         131.204.10.13
Address:        131.204.10.13#53

Non-authoritative answer:
13.10.204.131.in-addr.arpa      name = dns.eng.auburn.edu.

Authoritative answers can be found from:
10.204.131.in-addr.arpa nameserver = dns.eng.auburn.edu.
dns.eng.auburn.edu      internet address = 131.204.10.13
> set q=mx
> auburn.edu
Server:         131.204.10.13
Address:        131.204.10.13#53

Non-authoritative answer:
auburn.edu      mail exchanger = 10 aumail.duc.auburn.edu.

Authoritative answers can be found from:
auburn.edu      nameserver = dns.auburn.edu.
auburn.edu      nameserver = dns.eng.auburn.edu.
auburn.edu      nameserver = dns.duc.auburn.edu.
aumail.duc.auburn.edu   internet address = 131.204.2.83
dns.auburn.edu  internet address = 131.204.41.3
dns.eng.auburn.edu      internet address = 131.204.10.13
dns.duc.auburn.edu      internet address = 131.204.2.10
>
```

Query google.com's MX RR's

```
google.com mail exchanger = 10 smtp4.google.com.
google.com mail exchanger = 10 smtp1.google.com.
google.com mail exchanger = 10 smtp2.google.com.
google.com mail exchanger = 10 smtp3.google.com.
google.com nameserver = ns2.google.com.
google.com nameserver = ns3.google.com.
google.com nameserver = ns4.google.com.
google.com nameserver = ns1.google.com.
smtp1.google.com        internet address = 209.85.237.25
smtp2.google.com        internet address = 64.233.165.25
smtp3.google.com        internet address = 64.233.183.25
smtp4.google.com        internet address = 72.14.221.25
ns4.google.com          internet address = 216.239.38.10
ns1.google.com          internet address = 216.239.32.10
ns2.google.com          internet address = 216.239.34.10
ns3.google.com          internet address = 216.239.36.10
```

Whois service (1)



Whois service (2)

Search the .com, .net, .edu Whois

By submitting a Whois query, user agrees to abide by the Terms of Use.

Search Whois For:

SEARCH

Domain (ex.verisign.com)
Registrar (ex. ABC Registrar, Inc.)
Nameserver (ex.NS.VERISIGN.COM or 198.41.0.196)

Help | FAQs

Please Note: Successful domain registrations and modifications may not be in the WHOIS database for up to 48 hours.

Whois Server Version 2.0

Domain names in the .com and .net domains can now be registered with many different competing registrars. Go to http://www.internic.net for detailed information.

Domain Name: CNN.COM
Registrar: CSC CORPORATE DOMAINS, INC.
Whois Server: whois.corporatedomains.com
Referral URL: http://www.cscglobal.com
Name Server: NS1.TIMEWARNER.NET
Name Server: NS3.TIMEWARNER.NET
Name Server: NS5.TIMEWARNER.NET
Status: clientTransferProhibited
Updated Date: 04-feb-2010
Creation Date: 22-sep-1993
Expiration Date: 21-sep-2018

>>> Last update of whois database: Tue, 09 Mar 2010 21:23:26 UTC <<<

cnn.com uses Timewarner to host its authoritative DNS servers

Details about cnn.com

DNS for load balancing

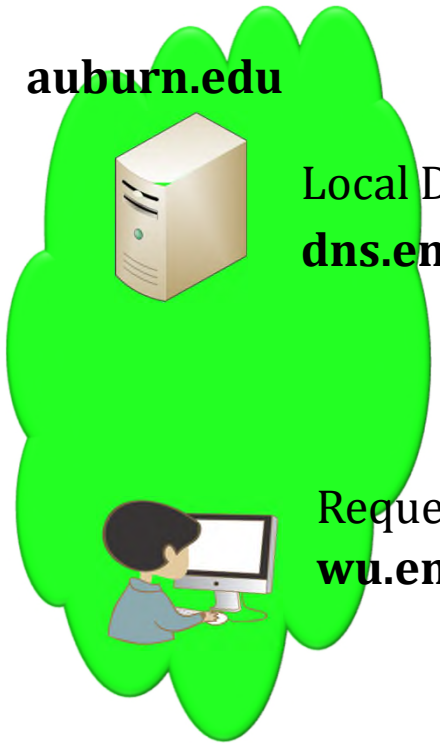
- Load balancing
  - Replicated Web/mail or other servers
  - A set of IP addresses for one name
  - Balance the load of each replicated server: Round robin DNS
    - When the request comes to the DNS server to resolve the domain name, it provides one of the several canonical names in a rotated order
    - This redirects the request to one of the several servers in a server group
    - Once the BIND feature of DNS resolves the domain to one of the servers, subsequent requests from the same client are sent to the same server
  - Name: google.com
    - Address: 74.125.67.100 (name = gw-in-f100.google.com)
    - Address: 74.125.45.100 (yx-in-f100.google.com)
    - Address: 209.85.171.100 (cg-in-f100.google.com)
    - ……

BIND for CNAME

- BIND 4 name servers
  - Multiple CNAMES are not considered as a configuration error
    - www IN CNAME srv1.auburn.edu.
    - IN CNAME srv2.auburn.edu.
    - IN CNAME srv3.auburn.edu.
- BIND 8 name servers
  - Signal an error for multiple CNAMES
  - This can be avoided by an explicit multiple CNAME configuration option
    - options {multiple-cnames yes;};
- BIND 9
  - Multiple CNAMES for one domain name are not a valid DNS server configuration for BIND 9
  - Multiple A records are used
    - www.auburn.edu. 60 IN A 131.204.2.3
    - www.auburn.edu. 60 IN A 131.204.2.4
    - www.auburn.edu. 60 IN A 131.204.2.5

Short life time for load balancing

How to get an IP address of www.cnn.com using recursive mode

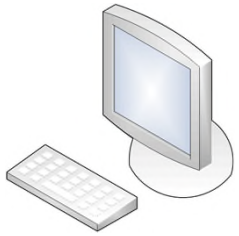


Root DNS Server: A.ROOT-SERVERS.NET
198.41.0.4

TLD DNS Server:
a.gtld-servers.net
192.5.6.30

Authoritative DNS Server

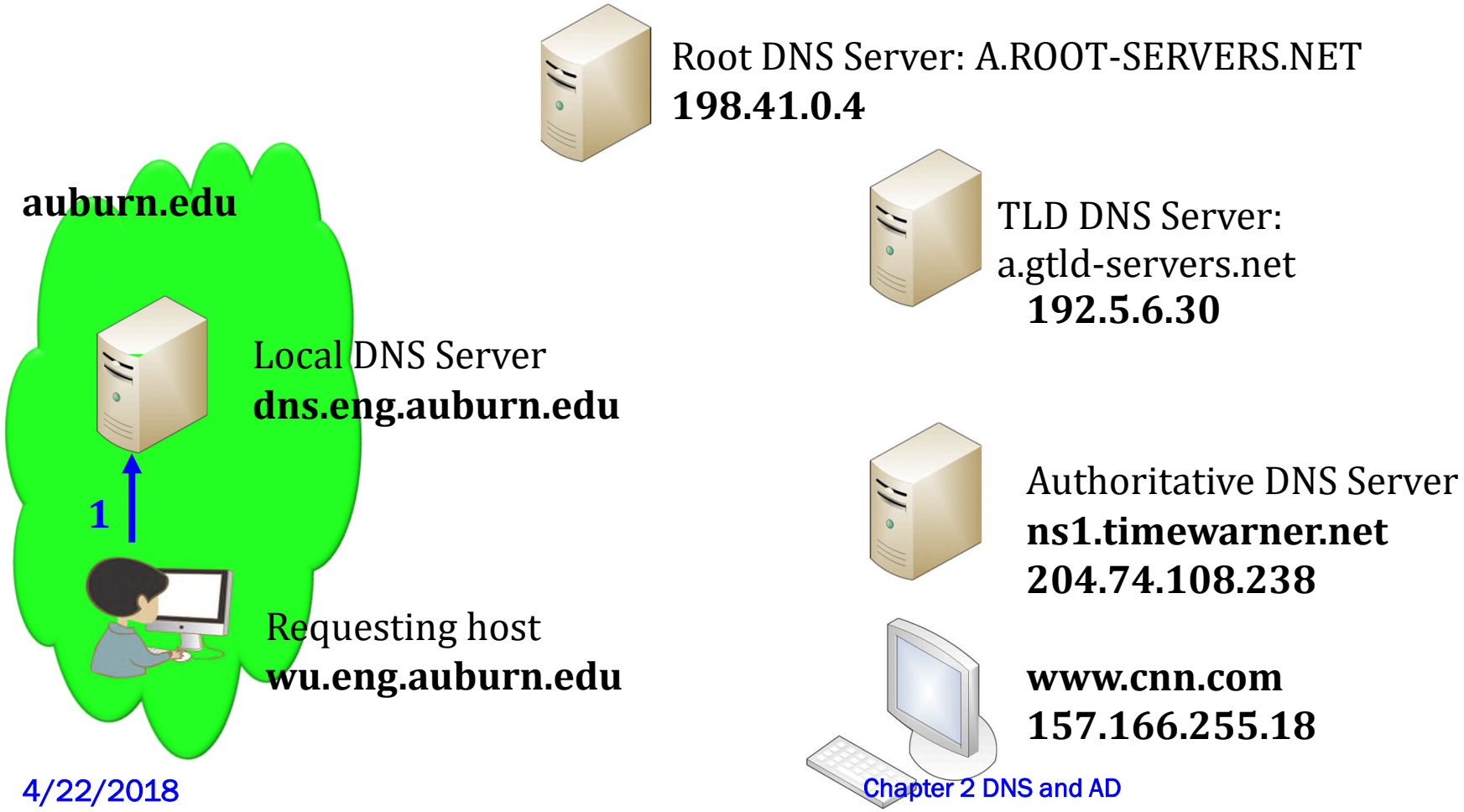
ns1.timewarner.net
204.74.108.238



www.cnn.com
157.166.255.18

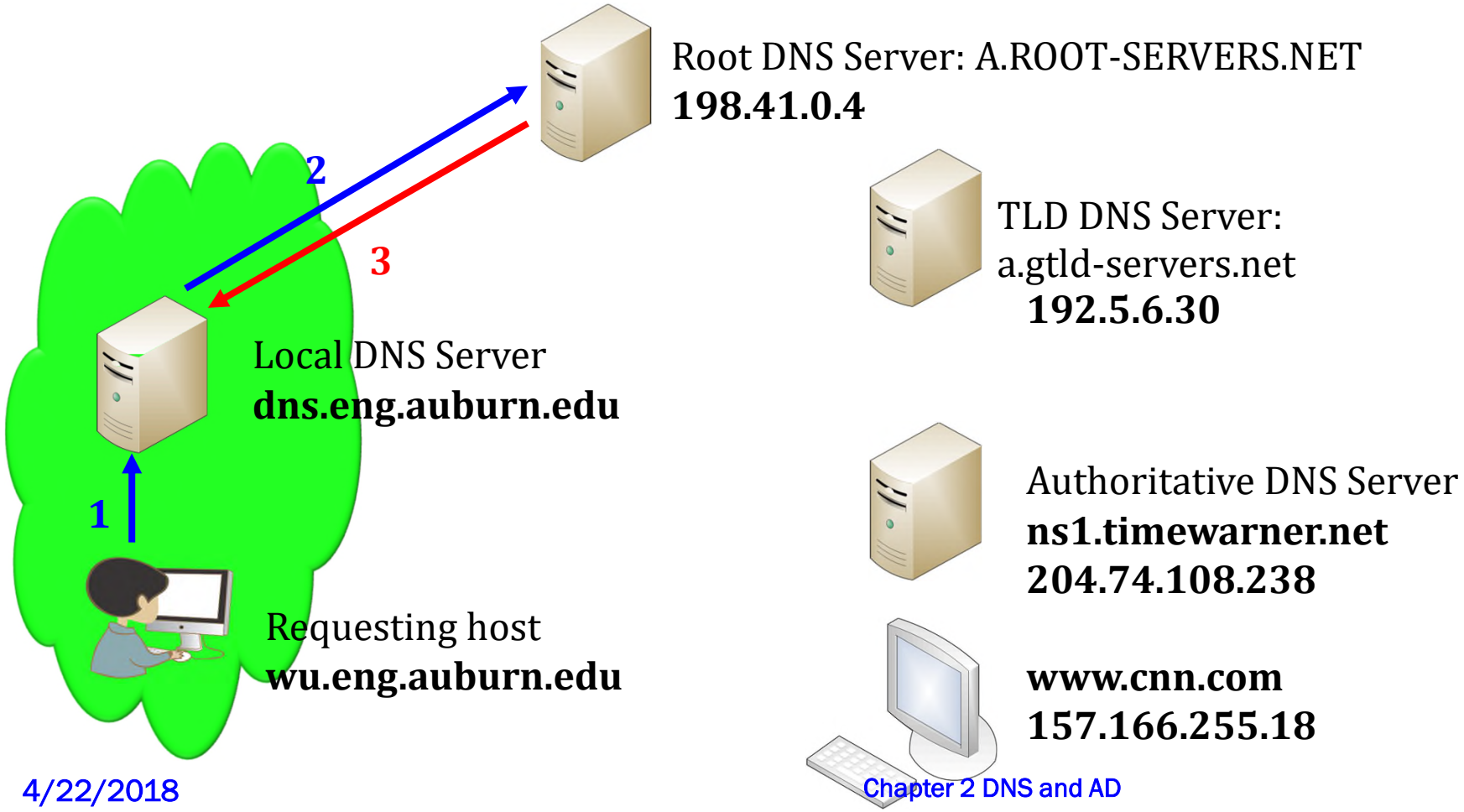
Step 1. From Client to Local DNS Server

- Requesting host queries local DNS Server to find www.cnn.com.
- Data Format
  * Header Section's Flag: QR - 0 (Query), RD - 1 (Recursive Query Desired)
  * Questions Section: QNAME- www.cnn.com, QTYPE – A



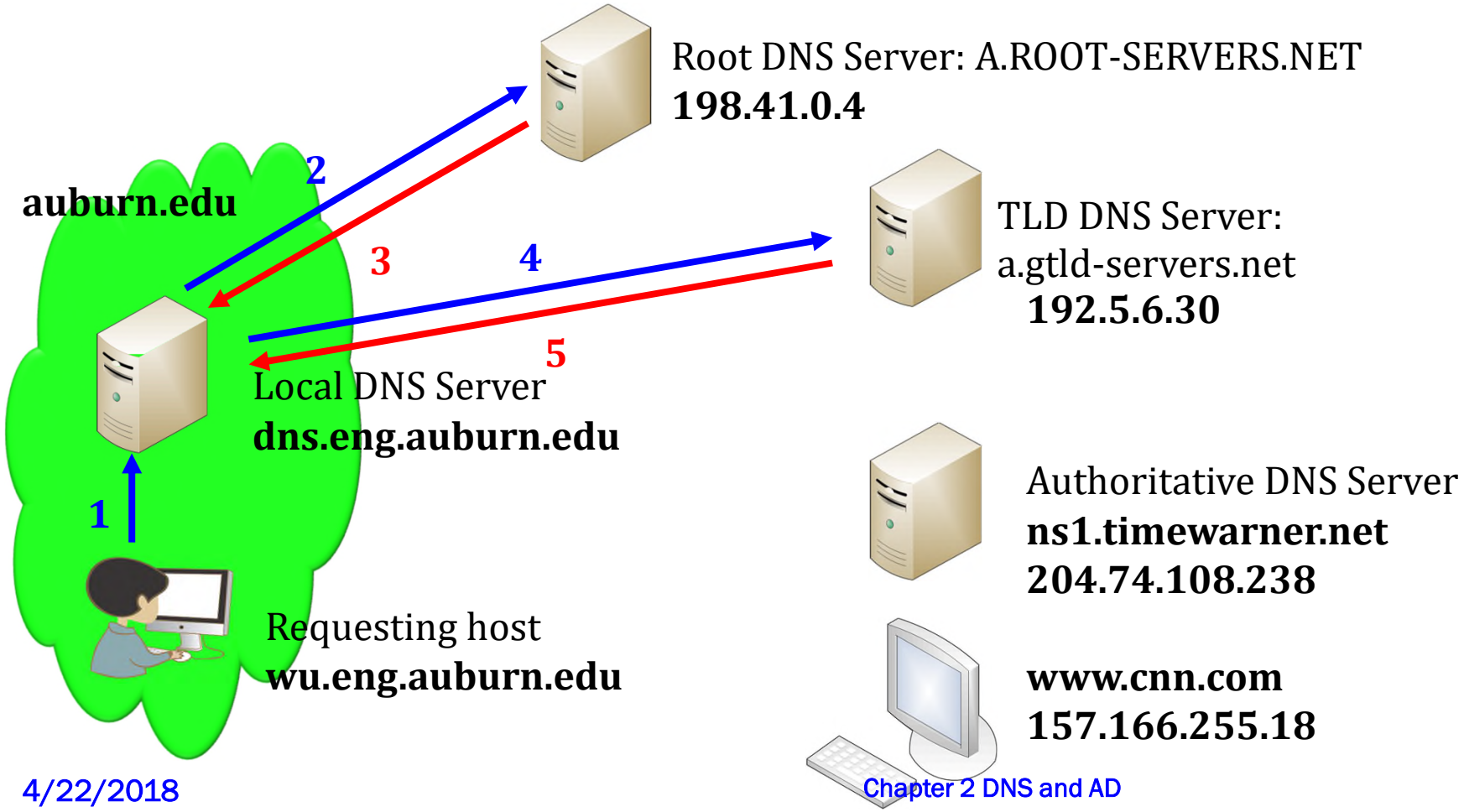
Step 2~3. Between Local and Root DNS Server

- Step 2's Data Format
  * Header Section's Flag: QR - 0 (Query), RD - 0 (Non-Recursive Query Desired)
  * Question Section: QNAME- www.cnn.com, QTYPE - A
- Step 3's Data Format
  * Header Section's Flag: QR - 1 (Response)
  * Authority Section: (*com*, a.gtld-servers.net, NS)
  * Additional Section: (*a.gtld-servers.net*, **192.5.6.30**, A)



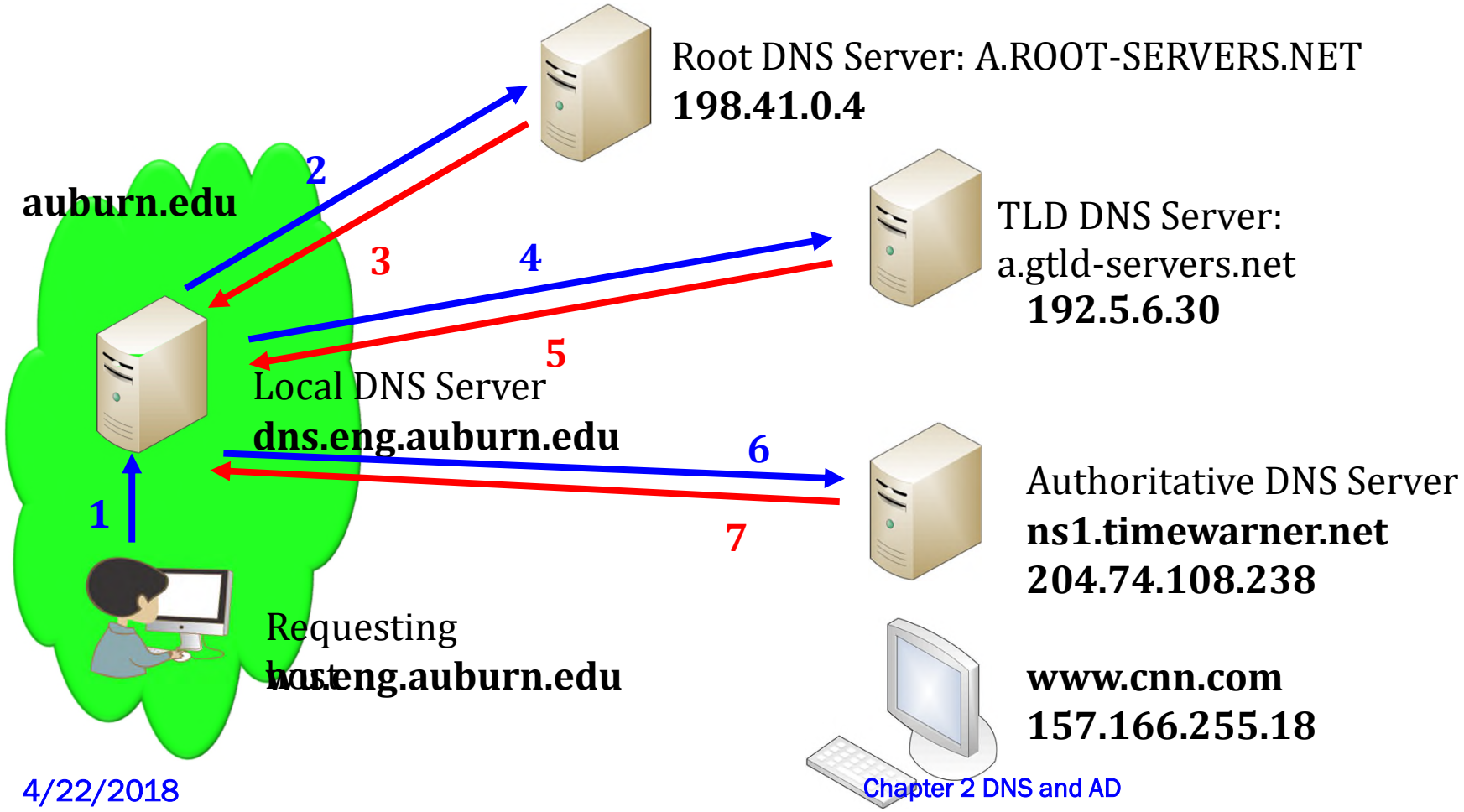
Step 4~5. Between Local and TLD DNS Server

- Step 4's Data Format
  * Header Section's Flag: QR - 0 (Query), RD - 0 (Non-Recursive Query Desired)
  * Question Section: QNAME- www.cnn.com, QTYPE - A
- Step 5's Data Format
  * Header Section's Flag: QR - 1 (Response)
  * Authority Section: (cnn.com, ns1.timewarner.net, NS)
  * Additional Section: (ns1.timewarner.net, 204.74.108.238, A)



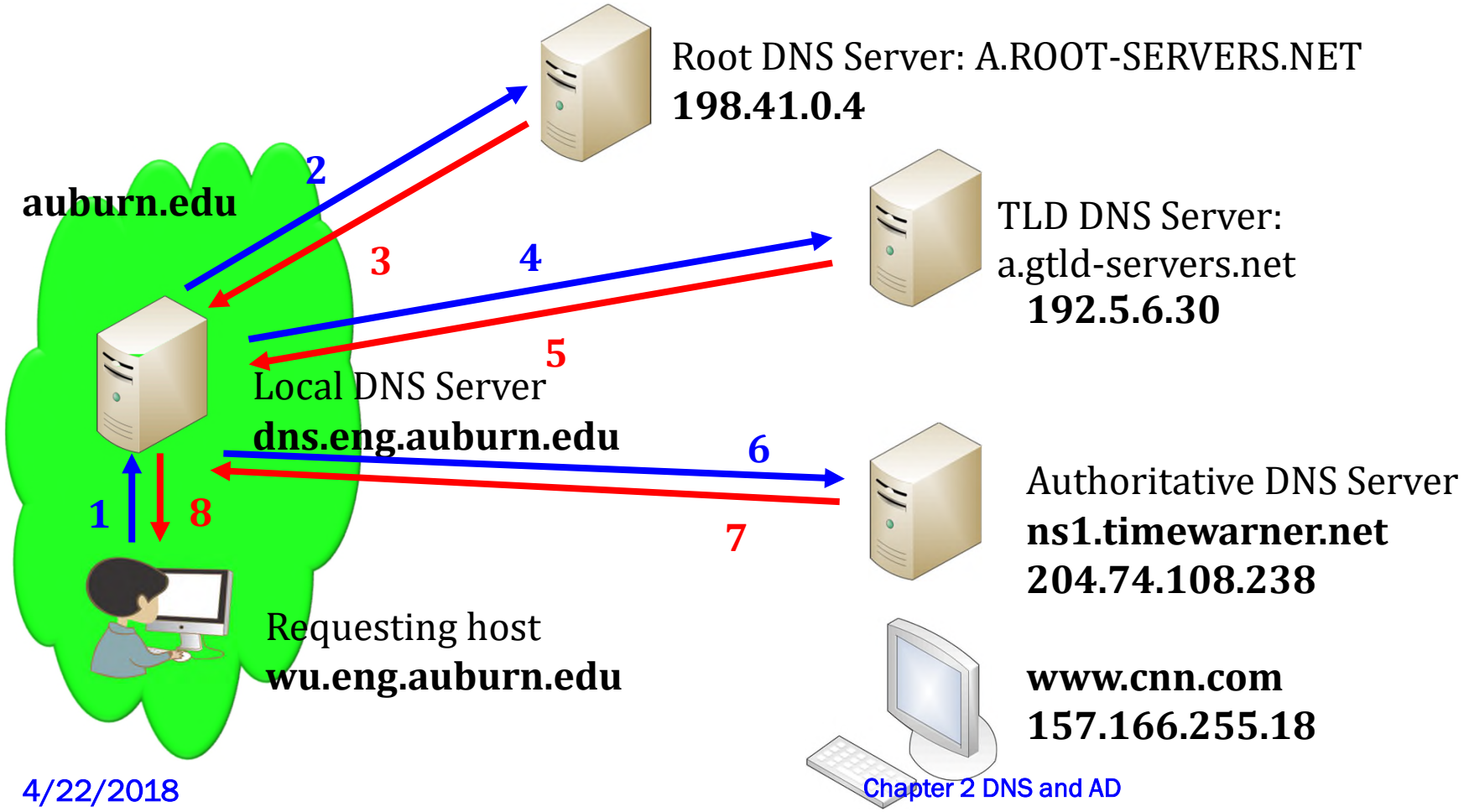
Step 6~7. Between Local and Authoritative DNS Server

- Step 6's Data Format
  * Header Section's Flag: QR - 0 (Query), RD - 0 (Non-Recursive Query Desired)
  * Question Section: QNAME- www.cnn.com, QTYPE - A
- Step 7's Data Format
  * Header Section's Flag: QR - 1 (Response)
  * Answer Section: (www.cnn.com, 157.166.255.18 , A)
  * Authority Section: (cnn.com, ns1.timewarner.net, NS)
  * Additional Section: (ns1.timewarner.net, 204.74.108.238, A)



Step 8. From Local DNS Server to Client

- Step 8's Data Format
  * Header Section's Flag: QR - 1 (Response), RA - 1 (Recursive Query Available)
  * Answer Section: (www.cnn.com, 157.166.255.18 , A)
  * Authority Section: (cnn.com, ns1.timewarner.net, NS)
  * Additional Section: (ns1.timewarner.net, 204.74.108.238, A)



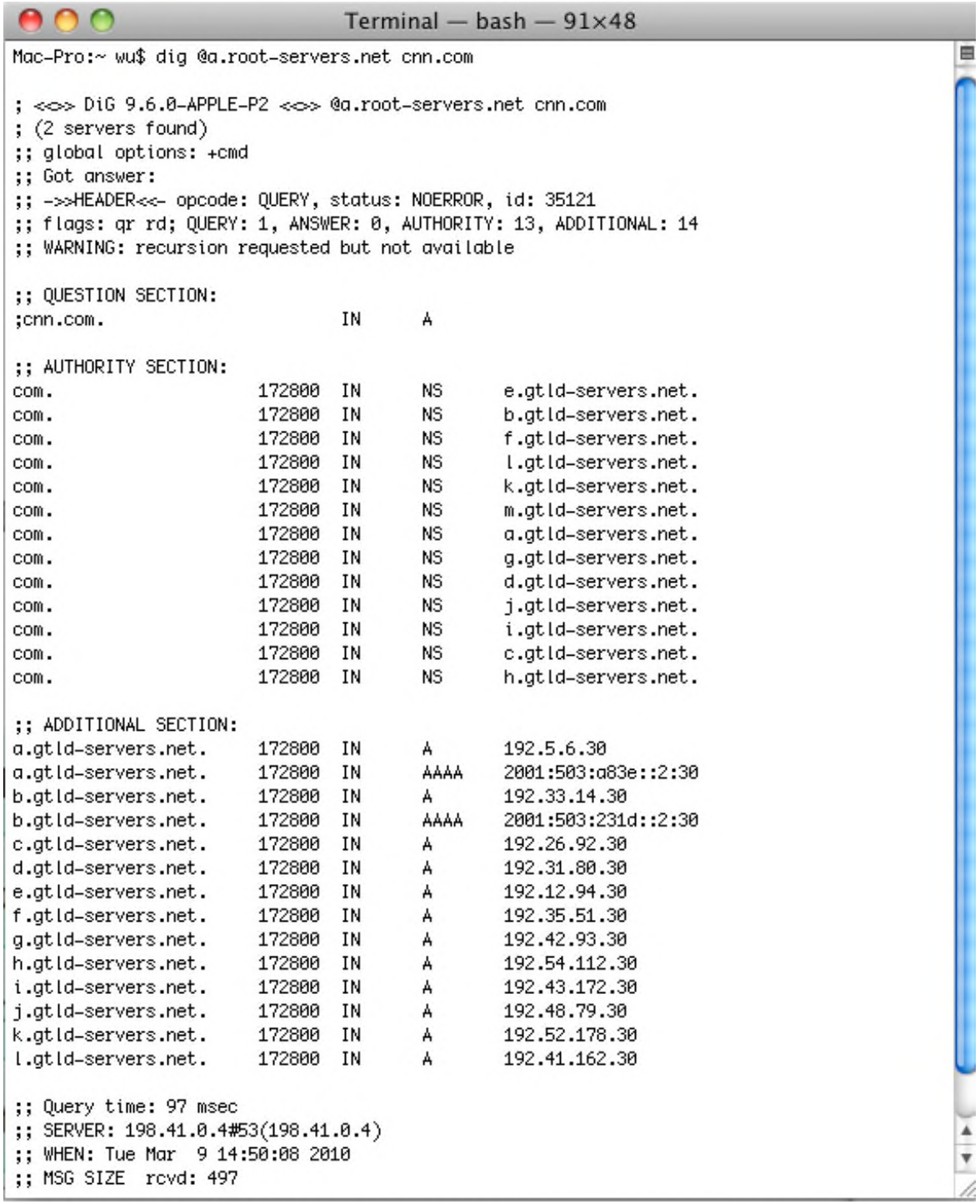
Root

Dig is not available in Windows;
Add-on can be installed

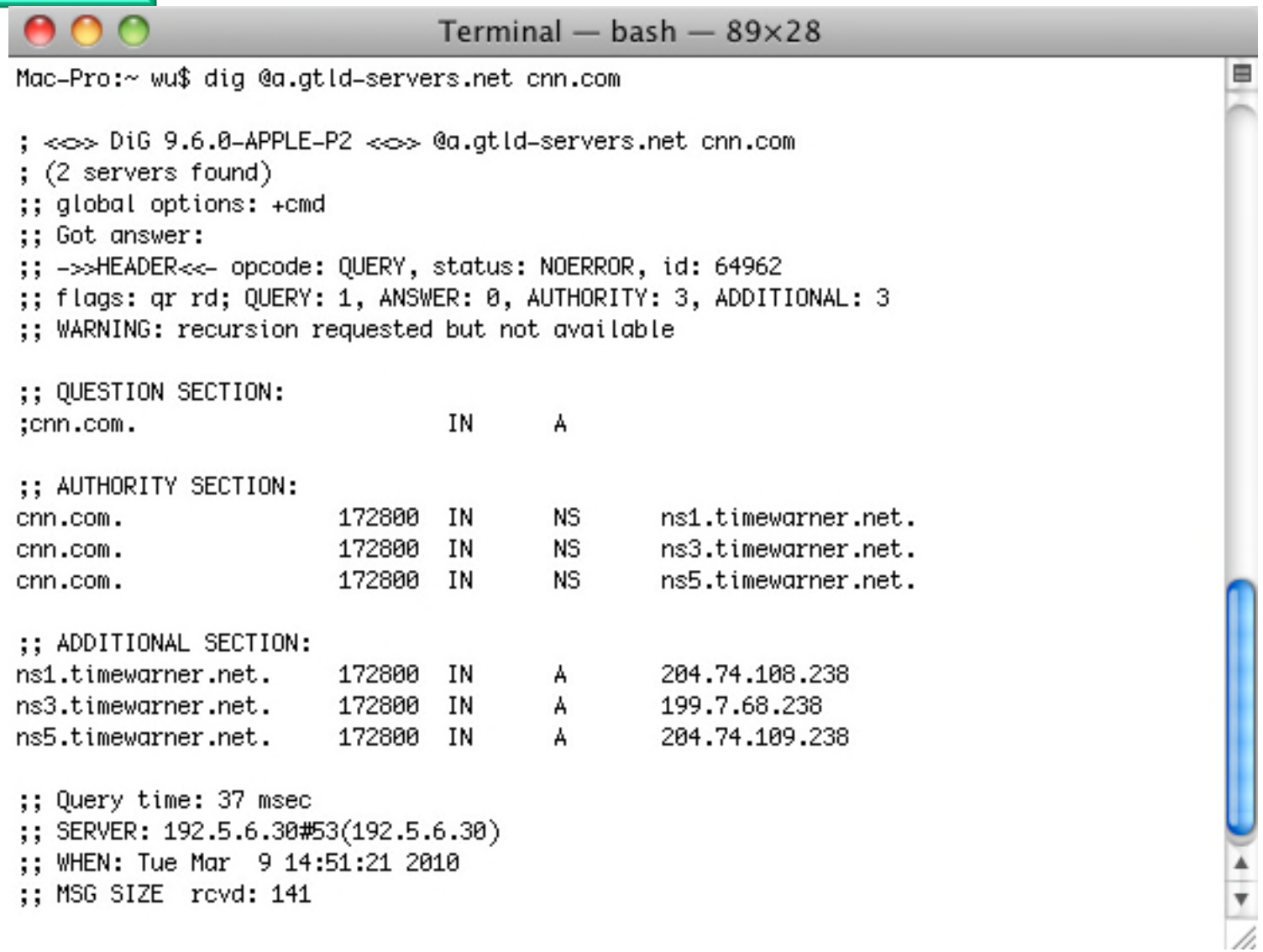
Terminal — bash — 91×48

```
Mac-Pro:~ wu$ dig

; <<>> DiG 9.6.0-APPLE-P2 <<>>
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18305
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 15

;; QUESTION SECTION:
;.                              IN      NS

;; ANSWER SECTION:
.                       362436  IN      NS      H.ROOT-SERVERS.NET.
.                       362436  IN      NS      B.ROOT-SERVERS.NET.
.                       362436  IN      NS      E.ROOT-SERVERS.NET.
.                       362436  IN      NS      K.ROOT-SERVERS.NET.
.                       362436  IN      NS      I.ROOT-SERVERS.NET.
.                       362436  IN      NS      F.ROOT-SERVERS.NET.
.                       362436  IN      NS      J.ROOT-SERVERS.NET.
.                       362436  IN      NS      A.ROOT-SERVERS.NET.
.                       362436  IN      NS      L.ROOT-SERVERS.NET.
.                       362436  IN      NS      C.ROOT-SERVERS.NET.
.                       362436  IN      NS      M.ROOT-SERVERS.NET.
.                       362436  IN      NS      D.ROOT-SERVERS.NET.
.                       362436  IN      NS      G.ROOT-SERVERS.NET.

;; ADDITIONAL SECTION:
H.ROOT-SERVERS.NET.     253637  IN      A       128.63.2.53
B.ROOT-SERVERS.NET.     253788  IN      A       192.228.79.201
E.ROOT-SERVERS.NET.     253788  IN      A       192.203.230.10
K.ROOT-SERVERS.NET.     253806  IN      A       193.0.14.129
I.ROOT-SERVERS.NET.     253637  IN      A       192.36.148.17
F.ROOT-SERVERS.NET.     429048  IN      A       192.5.5.241
F.ROOT-SERVERS.NET.     429048  IN      AAAA    2001:500:2f::f
J.ROOT-SERVERS.NET.     429048  IN      A       192.58.128.30
A.ROOT-SERVERS.NET.     426616  IN      A       198.41.0.4
A.ROOT-SERVERS.NET.     426616  IN      AAAA    2001:503:ba3e::2:30
L.ROOT-SERVERS.NET.     253806  IN      A       199.7.83.42
C.ROOT-SERVERS.NET.     253637  IN      A       192.33.4.12
M.ROOT-SERVERS.NET.     253806  IN      A       202.12.27.33
D.ROOT-SERVERS.NET.     253788  IN      A       128.8.10.90
G.ROOT-SERVERS.NET.     253637  IN      A       192.112.36.4

;; Query time: 8 msec
;; SERVER: 131.204.10.13#53(131.204.10.13)
;; WHEN: Tue Mar  9 14:47:47 2010
;; MSG SIZE  rcvd: 492
```

From Root


Terminal — bash — 91×48
Mac-Pro:~ wu$ dig @a.root-servers.net cnn.com
; <<>> DiG 9.6.0-APPLE-P2 <<>> @a.root-servers.net cnn.com
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35121
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 13, ADDITIONAL: 14
;; WARNING: recursion requested but not available
;; QUESTION SECTION:
;cnn.com. IN A
;; AUTHORITY SECTION:
com. 172800 IN NS e.gtld-servers.net.
com. 172800 IN NS b.gtld-servers.net.
com. 172800 IN NS f.gtld-servers.net.
com. 172800 IN NS l.gtld-servers.net.
com. 172800 IN NS k.gtld-servers.net.
com. 172800 IN NS m.gtld-servers.net.
com. 172800 IN NS a.gtld-servers.net.
com. 172800 IN NS g.gtld-servers.net.
com. 172800 IN NS d.gtld-servers.net.
com. 172800 IN NS j.gtld-servers.net.
com. 172800 IN NS i.gtld-servers.net.
com. 172800 IN NS c.gtld-servers.net.
com. 172800 IN NS h.gtld-servers.net.
;; ADDITIONAL SECTION:
a.gtld-servers.net. 172800 IN A 192.5.6.30
a.gtld-servers.net. 172800 IN AAAA 2001:503:a83e::2:30
b.gtld-servers.net. 172800 IN A 192.33.14.30
b.gtld-servers.net. 172800 IN AAAA 2001:503:231d::2:30
c.gtld-servers.net. 172800 IN A 192.26.92.30
d.gtld-servers.net. 172800 IN A 192.31.80.30
e.gtld-servers.net. 172800 IN A 192.12.94.30
f.gtld-servers.net. 172800 IN A 192.35.51.30
g.gtld-servers.net. 172800 IN A 192.42.93.30
h.gtld-servers.net. 172800 IN A 192.54.112.30
i.gtld-servers.net. 172800 IN A 192.43.172.30
j.gtld-servers.net. 172800 IN A 192.48.79.30
k.gtld-servers.net. 172800 IN A 192.52.178.30
l.gtld-servers.net. 172800 IN A 192.41.162.30
;; Query time: 97 msec
;; SERVER: 198.41.0.4#53(198.41.0.4)
;; WHEN: Tue Mar 9 14:50:08 2010
;; MSG SIZE rcvd: 497

From TLD


Terminal — bash — 89×28
Mac-Pro:~ wu$ dig @a.gtld-servers.net cnn.com
; <<>> DiG 9.6.0-APPLE-P2 <<>> @a.gtld-servers.net cnn.com
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64962
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 3, ADDITIONAL: 3
;; WARNING: recursion requested but not available
;; QUESTION SECTION:
;cnn.com. IN A
;; AUTHORITY SECTION:
cnn.com. 172800 IN NS ns1.timewarner.net.
cnn.com. 172800 IN NS ns3.timewarner.net.
cnn.com. 172800 IN NS ns5.timewarner.net.
;; ADDITIONAL SECTION:
ns1.timewarner.net. 172800 IN A 204.74.108.238
ns3.timewarner.net. 172800 IN A 199.7.68.238
ns5.timewarner.net. 172800 IN A 204.74.109.238
;; Query time: 37 msec
;; SERVER: 192.5.6.30#53(192.5.6.30)
;; WHEN: Tue Mar 9 14:51:21 2010
;; MSG SIZE rcvd: 141

From Name Server

Terminal — bash — 83×34

```
Mac-Pro:~ wu$ dig cnn.com

; <<>> DiG 9.6.0-APPLE-P2 <<>> cnn.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42840
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 3, ADDITIONAL: 3

;; QUESTION SECTION:
;cnn.com.                       IN      A

;; ANSWER SECTION:
cnn.com.                147     IN      A       157.166.226.25
cnn.com.                147     IN      A       157.166.224.26
cnn.com.                147     IN      A       157.166.224.25
cnn.com.                147     IN      A       157.166.255.19
cnn.com.                147     IN      A       157.166.255.18
cnn.com.                147     IN      A       157.166.226.26

;; AUTHORITY SECTION:
cnn.com.                74994   IN      NS      ns1.timewarner.net.
cnn.com.                74994   IN      NS      ns5.timewarner.net.
cnn.com.                74994   IN      NS      ns3.timewarner.net.

;; ADDITIONAL SECTION:
ns1.timewarner.net.     1159    IN      A       204.74.108.238
ns5.timewarner.net.     917     IN      A       204.74.109.238
ns3.timewarner.net.     2706    IN      A       199.7.68.238

;; Query time: 2 msec
;; SERVER: 131.204.10.13#53(131.204.10.13)
;; WHEN: Tue Mar  9 14:44:37 2010
;; MSG SIZE  rcvd: 237
```

Outline

Part 1

- DNS
  - Overview
  - Recursive and Iterative Queries
  - Recursive or Caching DNS Server
  - Resource record and DNS query
  - DNS Protocol
  - Reverse DNS lookup
- Active Directory (AD)
  - Overview and applications of AD
  - Structure and trust
  - AD objects
  - SRV RR
  - The Open Directory (OD)

Reverse DNS lookup (1)

- Forward resolution
  - Name to IP address mapping
  - DNS zones associated with such a process are often referred to as forward zones
- Reverse zone
  - IP address to Name mapping
- Reverse DNS lookup (rDNS) is a process to determine the hostname or host associated with a given IP address or host address
  - PTR record: Pointer to a canonical name
  - Reverse DNS lookup is accomplished using a "reverse IN-ADDR entry" in a DNS server

Reverse DNS lookup (2)

- If a company is assigned B class IP addresses such as 131.204.X.Y, then a reverse lookup zone of 204.131.in-addr.arpa. will be created
  - PTR record: 204.131.in-addr.arpa
  - ARPA: Address and Routing Parameter Area
- May contain delegations to some other domains (sub-domains) such as
  - 1.204.131.in-addr.arpa.
  - 2.204.131.in-addr.arpa., etc.

Example

- Reverse nslookup

```
Mac-Pro:~ wu$ nslookup 74.125.45.100
Server:         131.204.10.13
Address:        131.204.10.13#53

Non-authoritative answer:
100.45.125.74.in-addr.arpa      name = yx-in-f100.google.com.

Authoritative answers can be found from:
125.74.in-addr.arpa     nameserver = ns1.google.com.
125.74.in-addr.arpa     nameserver = ns2.google.com.
125.74.in-addr.arpa     nameserver = ns3.google.com.
125.74.in-addr.arpa     nameserver = ns4.google.com.
ns1.google.com  internet address = 216.239.32.10
ns2.google.com  internet address = 216.239.34.10
ns3.google.com  internet address = 216.239.36.10
ns4.google.com  internet address = 216.239.38.10
```

Common uses of the reverse DNS

- The original use of rDNS was primarily that of a network troubleshooting tool, such as traceroute, and ping
- e-mail anti-spam technique
  - Check the domain name in the rDNS to ensure they are from a legitimate domain
  - A Forward Confirmed Reverse DNS (FCrDNS) verification can create a form of authentication showing a valid relationship between the given IP address and domain name
  - This validation is not strong enough but often is used as the first round of defense against spammers and phishers who usually cannot pass this verification test when they use zombie computers to forge domains
  - A blacklist is available for blocking spammers' domains

Most widely used DNS implementation

- BIND (Berkeley Internet Name Domain) is the most commonly used DNS server on the Internet
- BIND9, a ground-up rewrite of BIND, provides full DNSSEC (Domain Name System Security Extensions, RFCs 4033, 4034, 4035) support including
  - Origin authentication of DNS data
  - Data integrity
  - Authenticated denial of existence
    - In the "split-horizon DNS" approach, the DNS server denies the existence of names to some outside clients, and provides correct information to the remaining clients
    - However, since DNSSEC information is cryptographically signed as authoritative, an attacker could request the signed "does not exist" record, then retransmit the record to cause a denial of service
    - RFC 5155 was developed to solve the problem
  - DNSSEC is expected to enhance Internet security
- Developed by Internet Systems Consortium

Outline

Part 1

- DNS
  - Overview
  - Recursive and Iterative Queries
  - Recursive or Caching DNS Server
  - Resource record and DNS query
  - DNS Protocol
  - Reverse DNS lookup
- Active Directory (AD)
  - Overview and applications of AD
  - Structure and trust
  - AD objects
  - SRV RR
  - The Open Directory (OD)

Up Another Notch: Directory Services

- Administration of network objects, including:
  - Users
  - Computers
  - User/Group access
  - Network resources
    - Servers/services/applications
    - Storage/Database/information
    - Printer/scanner/fax
  - Security policies for authenticating a network object for authorizing access to
    - information
    - applications
    - services

Security & Reliability

- Microsoft Active Directory (AD) protects
  - Network objects from unauthorized access
  - Network objects by using appropriate communication and storage methods
- AD replicates information about objects across the entire network to prevent object loss if one domain controller fails
  - Replication also improves the network performance by shortening the access distance

AD Advantages

- Save manpower
  - To update every computer software/firmware, a group policy can be used to push the actions
  - Centralize shared information using a service for syncing files/database
  - Centralize the management of network objects
  - Log in only once (single sign on) for a user to access the allowed resources in an active directory
  - Better reliability and performance by replication
- An information infrastructure for accessing resources
  - Access control of resources/services

Foundation of AD

- Built upon DNS: AD is a distributed database
- LDAP (Lightweight Directory Access Protocol)
  - The industry standard for directory access: RFC 4510
  - AD supports LDAP version 2 and 3
  - The primary directory access protocol used to
    - Add, modify, and delete information stored in AD
    - Query and retrieve data from Active Directory
- Kerberos for authentication

Domain controllers

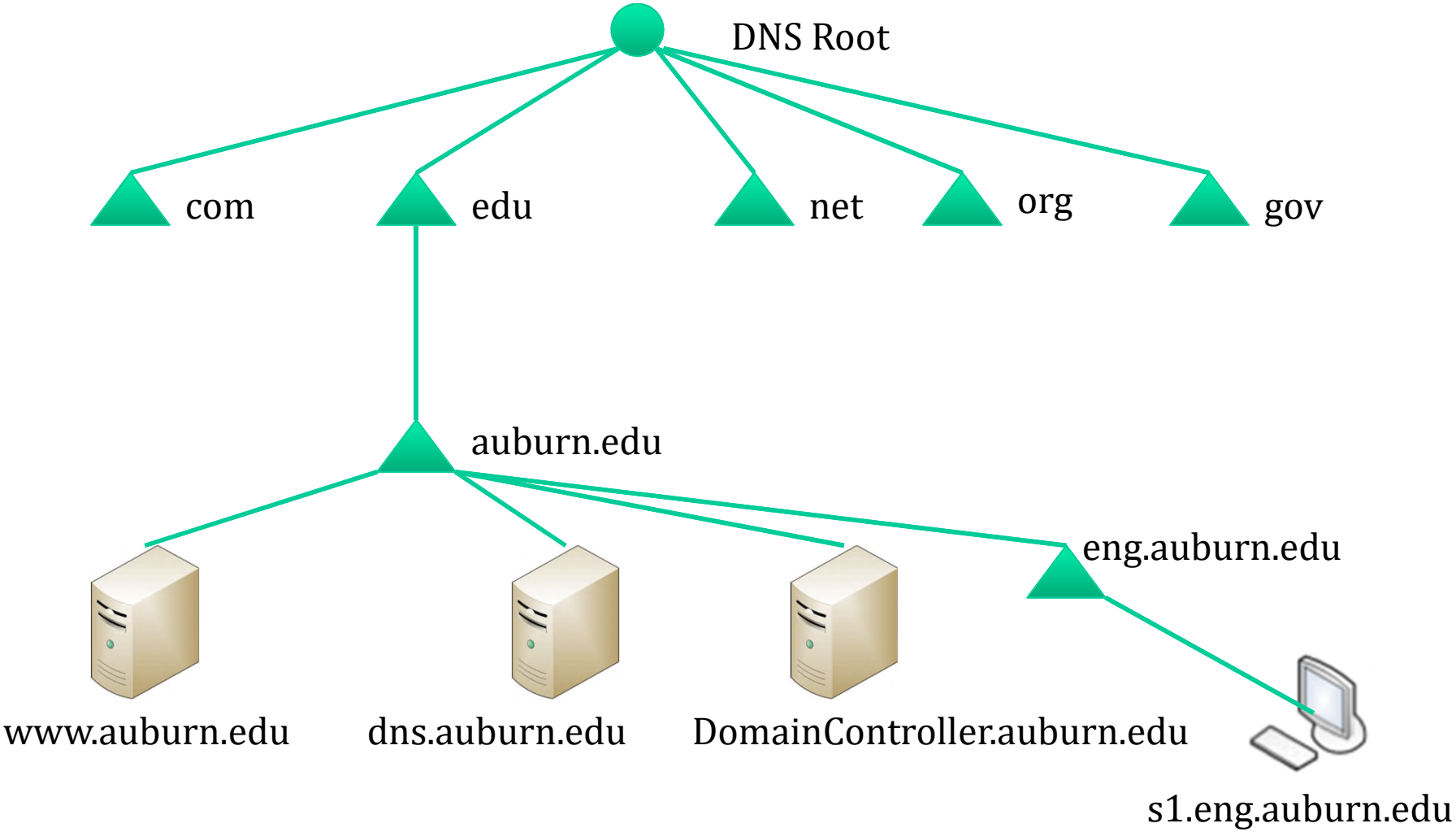
- Domains
  - A single security boundary for a network
  - A domain tree comprised of several domains sharing a common schema, configuration and forming a contiguous namespace
- Each domain may have a few domain controllers (DC's)
  - Only one primary DC (PDC) in a domain
  - A Domain controller is a unit of Replication
    - Domain controllers in a domain participate in replication and contain a complete copy of the AD information for their domain
  - Each DC has a copy of the AD by replication and synchronization
  - Changes on one computer/user/service are synchronized (converged) between all the DC computers by multi-master replication
- Servers, joined to AD, are not domain controllers, but called Member Servers

Outline

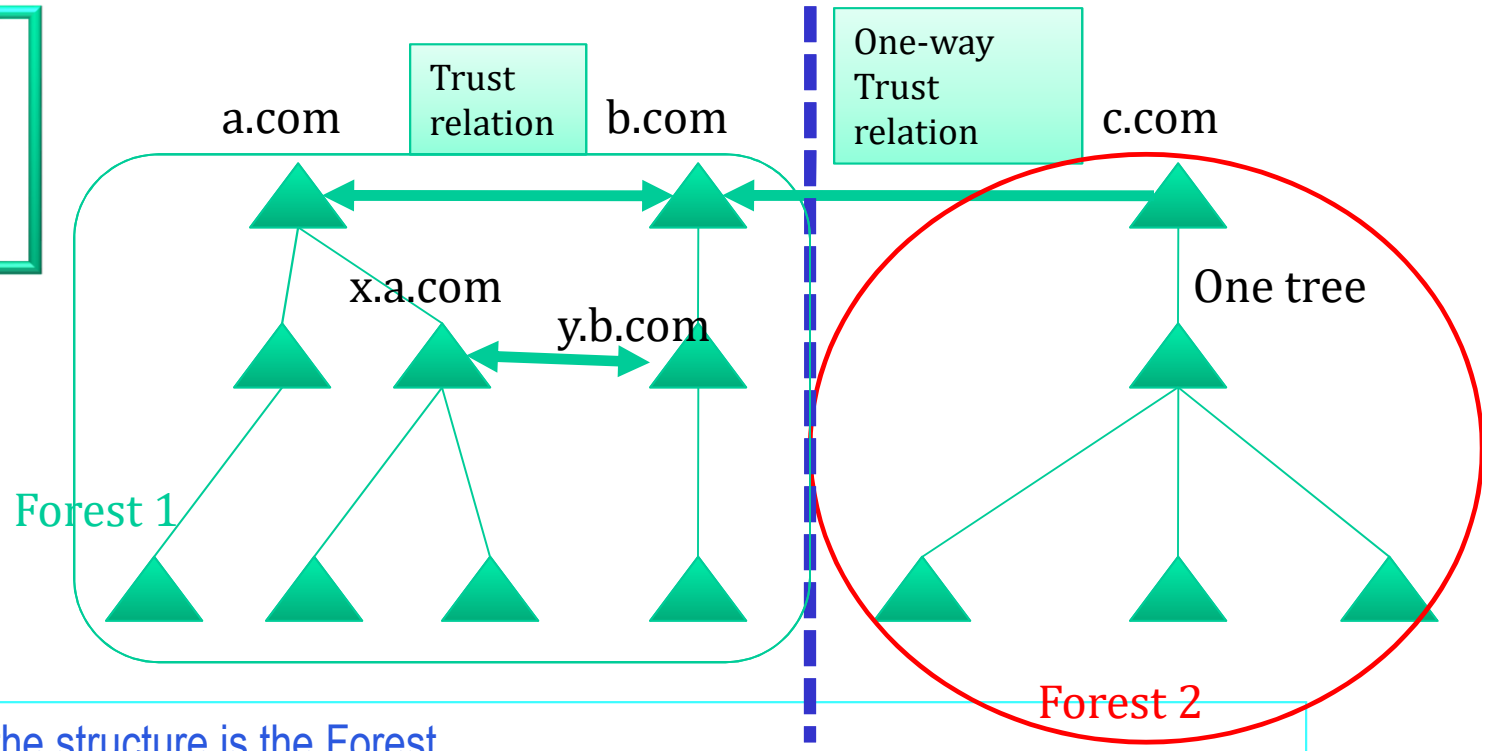
Part 1

- DNS
  - Overview
  - Recursive and Iterative Queries
  - Recursive or Caching DNS Server
  - Resource record and DNS query
  - DNS Protocol
  - Reverse DNS lookup
- Active Directory (AD)
  - Overview and applications of AD
  - Structure and trust
  - AD objects
  - SRV RR
  - The Open Directory (OD)

Hierarchical structure



Forest



- At the top of the structure is the Forest
  - A collection of every object, their attributes, and rules (attribute syntax) in the AD
  - A forest holds one or more transitive, trust-linked Trees
  - All trees in a forest share a common schema, configuration and global catalog
  - A forest does not need a distinguished name
  - The Forest Root Domain Is the first domain created in a Forest
- The figure illustrates one forest with two domain trees (a.com and b.com) and another tree c.com
- The three root domains are not contiguous with each other

Trust (1)

- a.com and b.com are the roots of separate trees in Forest 1
  - a.com is the forest root domain that is the very first domain created in the forest
- The two-way, transitive, tree-root trust between a.com and b.com provides complete trust between all domains in the two trees of Forest 1
  - A transitive trust means that if Domain X trusts Domain Y and Domain Y trusts Domain Z, then Domain X trusts Domain Z
- A forest is a collection of multiple trees
  - They share a common global catalog, and schema
  - A forest has automatic two-way transitive trust relationships for all domains
  - A forest is the default boundaries of trust, not a domain

Trust (2)

- y.b.com frequently uses resources in x.a.com
  - To shorten the trust path between the two domains, x.a.com and y.b.com can trust each other directly
- Domain c.com in Forest 2 implements an explicit one-way external trust relationship with domain b.com in Forest 1 so that users in domain b.com can be granted access to resources in domain c.com
  - Because the trust is non-transitive, no other domains in Forest 1 can be granted access to resources in c.com
  - c.com cannot access b.com since communication is one-way

Catalog

- The Global Catalog (GC) servers are used to provide a global listing of all objects in a Forest
  - a.com and b.com are in one GC
- The Global Catalog is held on domain controllers configured as global catalog servers
- Global Catalog servers replicate all objects from all domains to themselves
  - All GC servers contain all objects of a.com and b.com
- Global Catalog servers provide a global listing of objects in the forest

Outline

Part 1

- DNS
  - Overview
  - Recursive and Iterative Queries
  - Recursive or Caching DNS Server
  - Resource record and DNS query
  - DNS Protocol
  - Reverse DNS lookup
- Active Directory (AD)
  - Overview and applications of AD
  - Structure and trust
  - AD objects
  - SRV RR
  - The Open Directory (OD)

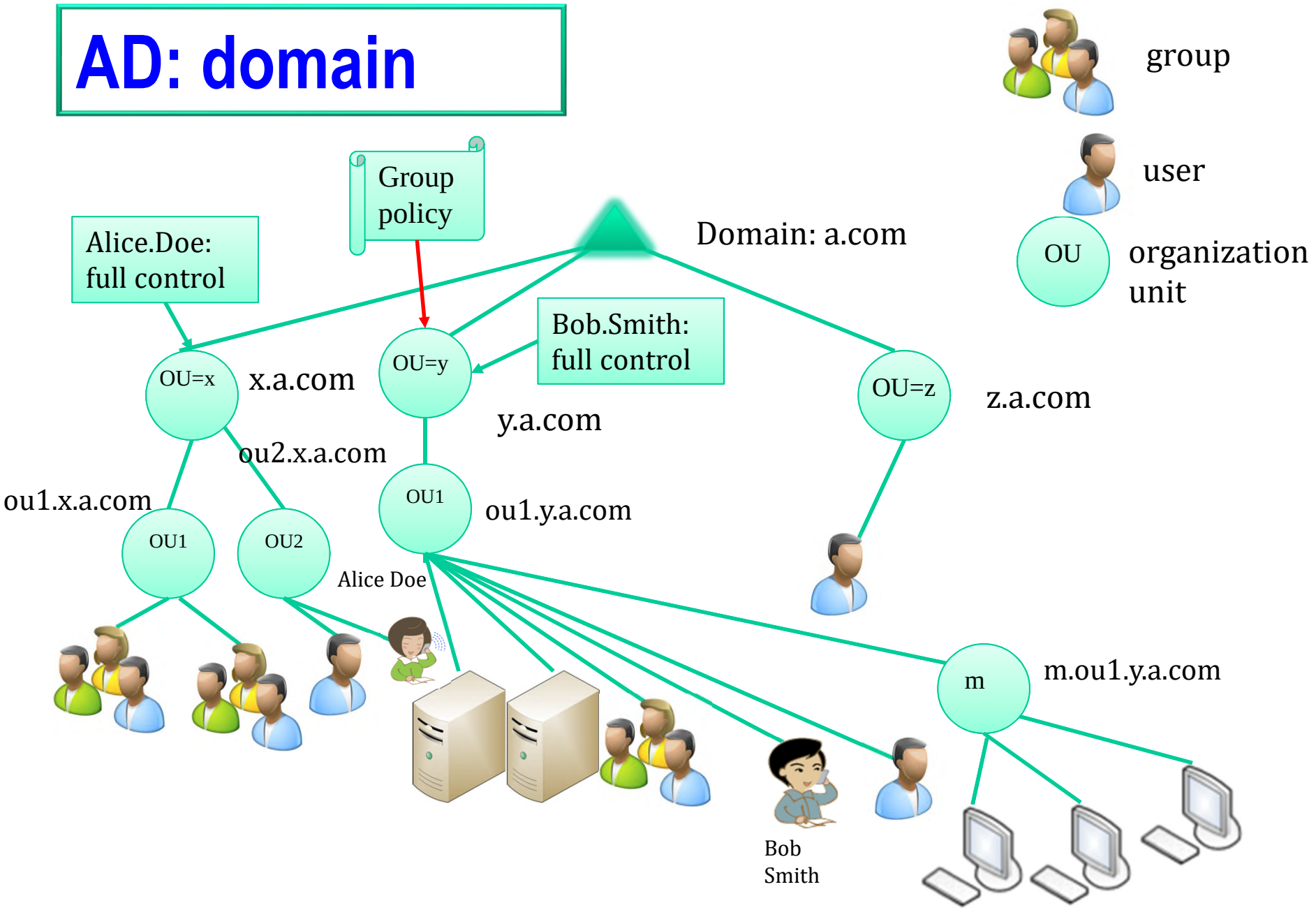
Active Directory Objects (1)

- Objects (including the schema)
  - AD objects are the entities that make up a network:
    - A user, a computer, a device, a service, an application or a group of objects
    - The schema defines the types of objects that are available to the AD service
  - AD can store, retrieve, and validate the data that it manages, regardless of the application that is the original source of the data
  - The schema is the AD component that defines all the objects and attributes that the directory service uses to store data
- An AD installation process creates the forest and also generates the default schema
  - The default schema replicates each new domain controller during installation of the AD

Active Directory Objects (2)

- Object naming conventions: including security principal names, LDAP-related names, and logon names
- Domains: including, trees, forests, trusts, and organizational units
  - Sites (including replication): Location in a network (subnets) that contains AD servers
- How delegation and Group Policy apply to OUs (organization units), domains, and sites

AD: domain


group
user
OU
organization unit
Group policy
Domain: a.com
Alice.Doe: full control
Bob.Smith: full control
OU=x
x.a.com
OU=y
y.a.com
OU=z
z.a.com
ou2.x.a.com
ou1.x.a.com
OU1
OU2
OU1
ou1.y.a.com
Alice Doe
m
m.ou1.y.a.com
Bob Smith

Structure and Group Policy

- x.a.com is a child domain of a.com
- a.com is a parent domain of x.a.com
- Two-way transitive trust relationships between parent and child domains
- The hierarchical structure of AD allows for
  - Delegating authority
  - Applying administrative and security policies (Group Policies)

Administration of permissions

- Use OU to group objects into a logical hierarchy that best suits the needs of your organization
  - A group of accountants
  - A group of servers
  - A group of students
- Delegate administrative control over the objects within an OU by assigning specific permissions to users and groups
  - Alice Doe is assigned to have full control of the OU, x.a.com
  - Alice Doe modifies specific attributes of any object in a single OU

AD object definitions

- Object definitions control the types of data that the objects can store, and the syntax of the data
  - Only data that has an existing object definition in the schema can be stored in the AD
  - If a new type of data needs to be stored, a new object definition for the data must first be created in the schema
    - All objects are defined in the schema
- AD stores and retrieves information from a wide variety of applications and services
  - AD standardizes how data is stored in the directory
  - Directory service can retrieve, update, and replicate data
- Directory service uses objects as units of storage
  - Each time the AD handles data, the directory queries the schema for an appropriate object definition
  - Based on the object definition in the schema, the AD creates the object and stores the data

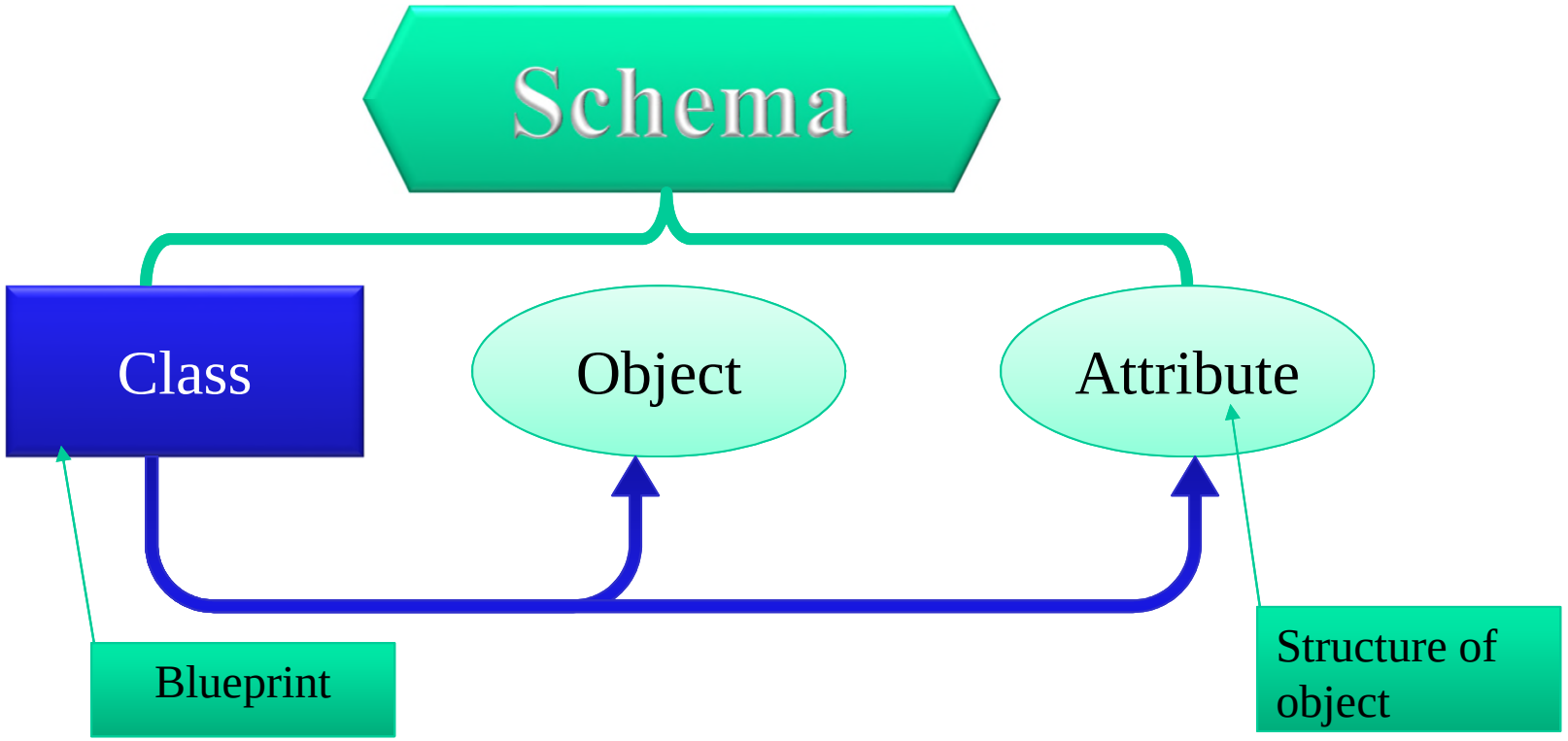
Schema (1)

- Schema contains definitions of the following:
  - Objects that are used to store data in the AD
  - The rules that govern the structure of those objects
  - The structure and the content of the AD itself
- Schema definitions consist of objects, attributes, and classes (three components)
  - Objects are structures that store both data the objects represent and data that controls the content and structure of the objects
    - For example, a user account object contains a user's logon name and data that indicate the proper syntax for storing the user's logon name in the user object

Schema (2)

- AD uses attribute syntax to ensure that information is stored in a legitimate format and that the information is a valid data type
  - Each attribute in the AD has an associated syntax. For example, integer, string, numeric, and so on
  - A phone number attribute can only store digits 0 through 9 and max. digits are 13
- AD uses objects to store data while the data are maintained in the directory
  - When the AD stores an object, some associated data stored along with the object are the object's attributes

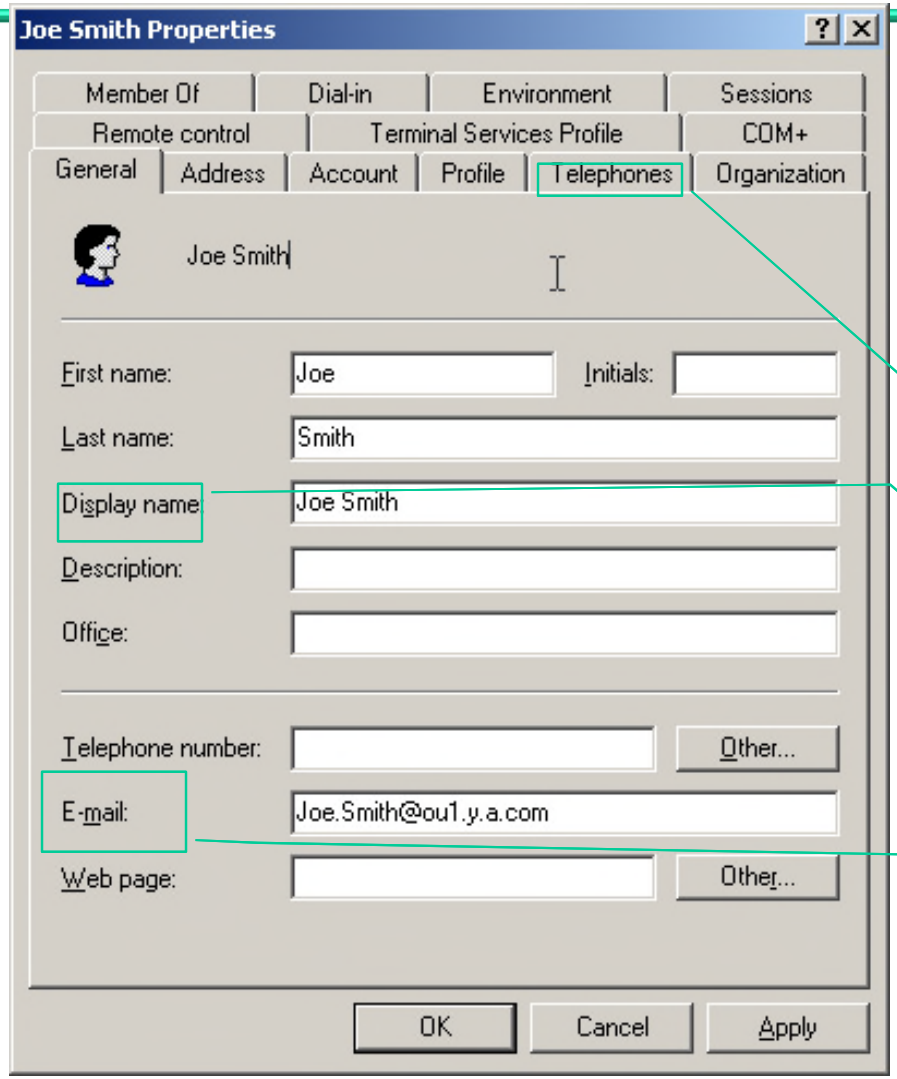
Schema overview



Attributes

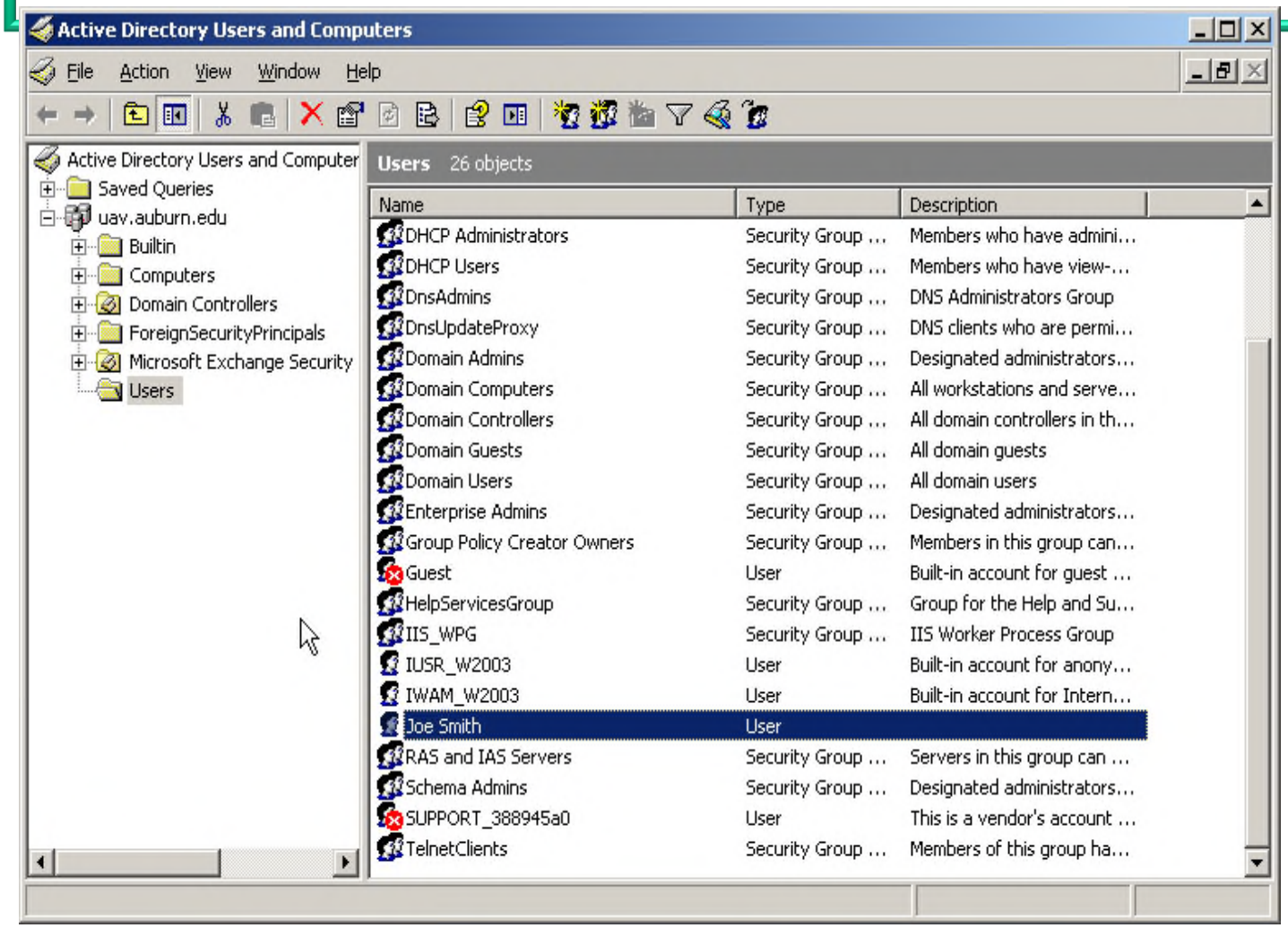
- Attributes contain data that defines the information that is stored in an object or in another attribute
  - For example, a user account object has attributes that store user information
    - Such as the user's first name, last name, password, Building/office number, and telephone number
  - Different types of objects have different attributes
    - An object definition is really an association of various attributes that are used to describe the characteristics of an object that stores specific pieces of data
- Efficiently define many different types of objects using common attributes
  - For example, many objects have a security descriptor to define who is allowed to access and make changes to the contents of the object
  - Rather than create a separate security descriptor definition for each object definition, the schema defines a single security descriptor object, and lets all other object definitions refer to it

A User Object in Active Directory



Schema attributes

Active Directory for Users



Class

- Classes are used as blueprints that can be used each time a new object is created
  - When a new object is created in the AD, the object's class determines the attributes that are associated with the new object, including which attributes are required or optional
    - For example, when a new user account object is created in the AD, its definition comes from the classUser class (think C++ Class)
    - The class dictates that the new account object is required to have a user name attribute and a password attribute, and optionally it might have an office number attribute
  - A schema object (classSchema object) defines each class in the schema
  - Another schema object (attributeSchema object) defines each attribute in the schema
- Every class is an instance of the classSchema class, and every attribute is an instance of the attributeSchema class

Distinguished name (DN) and RDN (Relative DN)

- The full path to the object is defined by the DN (distinguished name)
  - LDAP API references an LDAP object by its distinguished name (DN)
- The name of the object itself is defined by the RDN (relative DN)
  - The RDN is that segment of an object's DN that is an attribute of the object itself
- Example:
  - DN: cn=Bob.Smith, ou=OU1, ou=y, dc=a, dc=com
    - A DN is a sequence of relative distinguished names (RDN) connected by commas
  - The RDN of the Bob.Smith user object is cn=Bob.Smith
    - A RDN is an attribute with an associated value in the form attribute=value
  - The RDN of OU1 (the parent object of Bob.Smith) is ou=OU1, and so on
- AD tools do not display the LDAP abbreviations for the naming attributes (dc=, ou=, or cn=)
  - These abbreviations are shown only to illustrate how LDAP recognizes portions of the DN

LDAP naming conventions

LDAP DN (Distinguished Name) and RDN (relative distinguished name) Naming Convention	Corresponding Active Directory Naming Convention
cn=common name	cn=common name
ou=organizational unit	ou=organizational unit
o=organization	dc=domain component
c=country	(not supported)

- **Note:** cn=, ou=, etc are *attribute types*
- The attribute type used to describe an object's RDN is called the *naming attribute*
- The AD naming attributes, shown on the right above, are for the following AD object classes:
  - cn is used for the *user* object class
  - ou is used for the *organizational unit* (OU) object class
  - dc is used for the *domainDns* object class

Schema Object Name

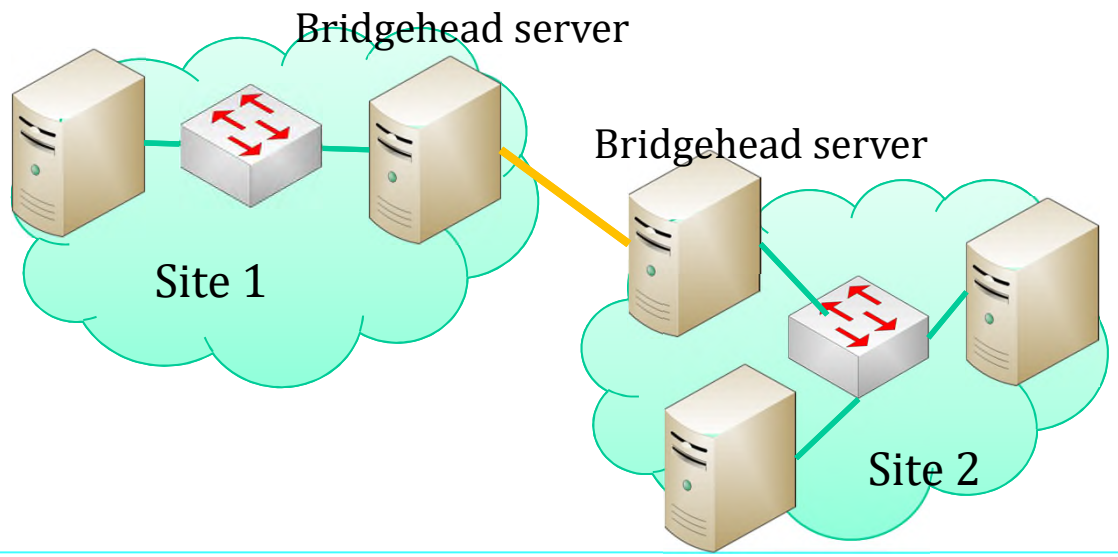
- A schema object can be referenced by each of the following types of names:
  - LDAP display name
    - The LDAP display name consists of one or more words combined, using initial caps for words after the first word
    - For example, mailAddress is the LDAP display name for email address
  - Common name
    - The common name is a more readable version of the LDAP display name
    - For example, SMTP-Mail-Address is the common name for mailAddress
    - It is the relative distinguished name (RDN) of the object in the schema that represents the object class
  - Object identifier (OID)
    - A schema object's identifier is a number issued by an issuing authority such as the International Organization for Standardization (ISO) and the American National Standards Institute (ANSI)
    - For example, the OID for the SMTP-Mail-Address attribute is 1.2.840.113556.1.4.786
    - OIDs are guaranteed to be unique across all networks worldwide
    - Once a root OID is obtained from an issuing authority, one can use it to allocate additional OIDs to form a hierarchy

OID

- For example, Microsoft has been issued the root OID of 1.2.840.113556
  - Microsoft manages further branches from this root internally
  - One of the branches is used to allocate OIDs for AD schema classes, and another for attributes
  - To continue the example, the OID in Active Directory is 1.2.840.113556.1.5.4, which identifies the Built-in Domain class

1	A branch called Active Directory that includes…
5	A branch called classes that includes…
4	A branch called Built-in Domain

Sites



- Sites are established based on the performance requirement for replication
  - Reduce distance or propagation time
- Two sites connected by a site link
  - Each site's preferred bridgehead server is used preferentially for inter-site information exchange
  - Of the 5 domain controllers in the figure, two are bridgehead servers
- The bridgehead servers are the preferred servers for replication, replicating AD changes between sites

Outline

Part 1

- DNS
  - Overview
  - Recursive and Iterative Queries
  - Recursive or Caching DNS Server
  - Resource record and DNS query
  - DNS Protocol
  - Reverse DNS lookup
- Active Directory (AD)
  - Overview and applications of AD
  - Structure and trust
  - AD objects
  - SRV RR
  - The Open Directory (OD)

Relationship between DNS and AD (1)

- In order for a DNS server to be able to support AD (to advertise the AD directory service), the DNS server is required to support:
  - Service (SRV) resource record type (RFC 2782)
  - Dynamic update protocol for higher rates of update (RFC 2136)
    - DNS has low update rate
    - AD uses LDAP
- AD uses DNS as the location mechanism for domain controllers, enabling computers on the network to obtain the IP addresses of domain controllers
  - SRV resource records map the name of a service to the name of a server offering that service
  - AD clients and domain controllers use SRV records to determine the IP addresses of domain controllers

SRV RR (1)

- RFC 2052 describes a DNS RR which specifies the location of the server(s) for a specific protocol and domain
- Resembles a more general form of MX
- The SRV RR allows administrators to use several servers for a single domain, to move services from host to host, and to designate some hosts as primary servers for a service and others as backups
- The SRV RR format is

```
http.tcp.www SRV 0 0 80 webserver.auburn.edu.
```

Service.Proto.Name TTL Class SRV Priority Weight Port Target

RR format: **`name ttl [Class] Type [pref.] value`**

SRV RR (2)

- http.tcp SRV 0 0 80 webserver.auburn.edu.
- A RR: webserver A 131.204.7.11
- The format of the SRV RR

  `Service.Proto.Name TTL Class SRV Priority Weight Port Target`

  - Service
    - The symbolic name of the desired service, as defined in Assigned Numbers or locally
    - The Service is case insensitive
  - Proto (protocol)
    - TCP or UDP
    - The Proto is case insensitive
  - Name
    - The domain name of this RR
  - TTL
  - Class

SRV RR (3)

- Priority
  - Similar to MX, and the priority of this target host
  - A client MUST attempt to contact the target host with the lowest-numbered priority it can reach; target hosts with the same priority should be tried in pseudorandom order
  - The range is 0-65535
- Weight Load balancing mechanism
  - When selecting a target host from among those with the same priority, the chance of trying this one first SHOULD be proportional to its weight
  - The range of this number is 1-65535
  - Domain administrators should use Weight 0 when load balancing is not necessary, thus making the RR easier to read for humans (less noisy).
- Port
  - The port on the target host for this service. The range is 0-65535
  - This is often specified in Assigned Numbers but need not be
- Target
  - Similar to MX, and the domain name of the target host
  - There must be one or more A records for this name

Relationship between DNS and AD (2)

- During the installation of Active Directory, the service (SRV) and address (A) resource records are dynamically registered in DNS
  - Both types of records are necessary for the functionality of the domain controller (Locator) mechanism
  - To find domain controllers in a domain or forest, a client queries DNS for the SRV and A resource records of the domain controller
  - The resource records provide the client with the names and IP addresses of the domain controllers
  - In this context, the SRV and A resource records are referred to as Locator DNS resource records
- When a domain controller is added to a forest, a DNS zone hosted on a DNS server is updated with the Locator DNS resource records for that domain controller

Relationship between DNS and AD (3)

- The SRV record is used to map the name of a service (in this case, the LDAP and Kerberos service) to the DNS computer name of a server that offers that service
- An SRV RR is displayed for the following services:
  - _kerberos
    - **_kerberos._tcp.***DnsDomainName*
    - **_kerberos._udp.***DnsDomainName*
      - _kerberos._tcp.auburn.edu SRV 0 0 88 domainController.auburn.edu.
      - _kerberos._udp.auburn.edu SRV 0 0 88 domainController.auburn.edu.
  - _ldap
    - **_ldap._tcp.***DnsDomainName*
    - _ldap._tcp.auburn.edu
      - _ldap._tcp.auburn.edu SRV 0 0 389 domainController.auburn.edu.
  - A RR: domainController.auburn.edu A 131.204.79.10

Outline

Part 1

- DNS
  - Overview
  - Recursive and Iterative Queries
  - Recursive or Caching DNS Server
  - Resource record and DNS query
  - DNS Protocol
  - Reverse DNS lookup
- Active Directory (AD)
  - Overview and applications of AD
  - Structure and trust
  - AD objects
  - SRV RR
  - The Open Directory (OD)

Open Directory

- Open directory (OD): standards-based directory service
- Apple supports both OD and AD
- Mac OS X client and server systems are compatible with other standards-based LDAP servers, and can even plug into environments that use proprietary services such as Microsoft's Active Directory
- Linux computers can also be managed by AD using 3rd party tools
  - Centrify's DirectControl for Mac
  - Likewise Software's Likewise Enterprise 4.0