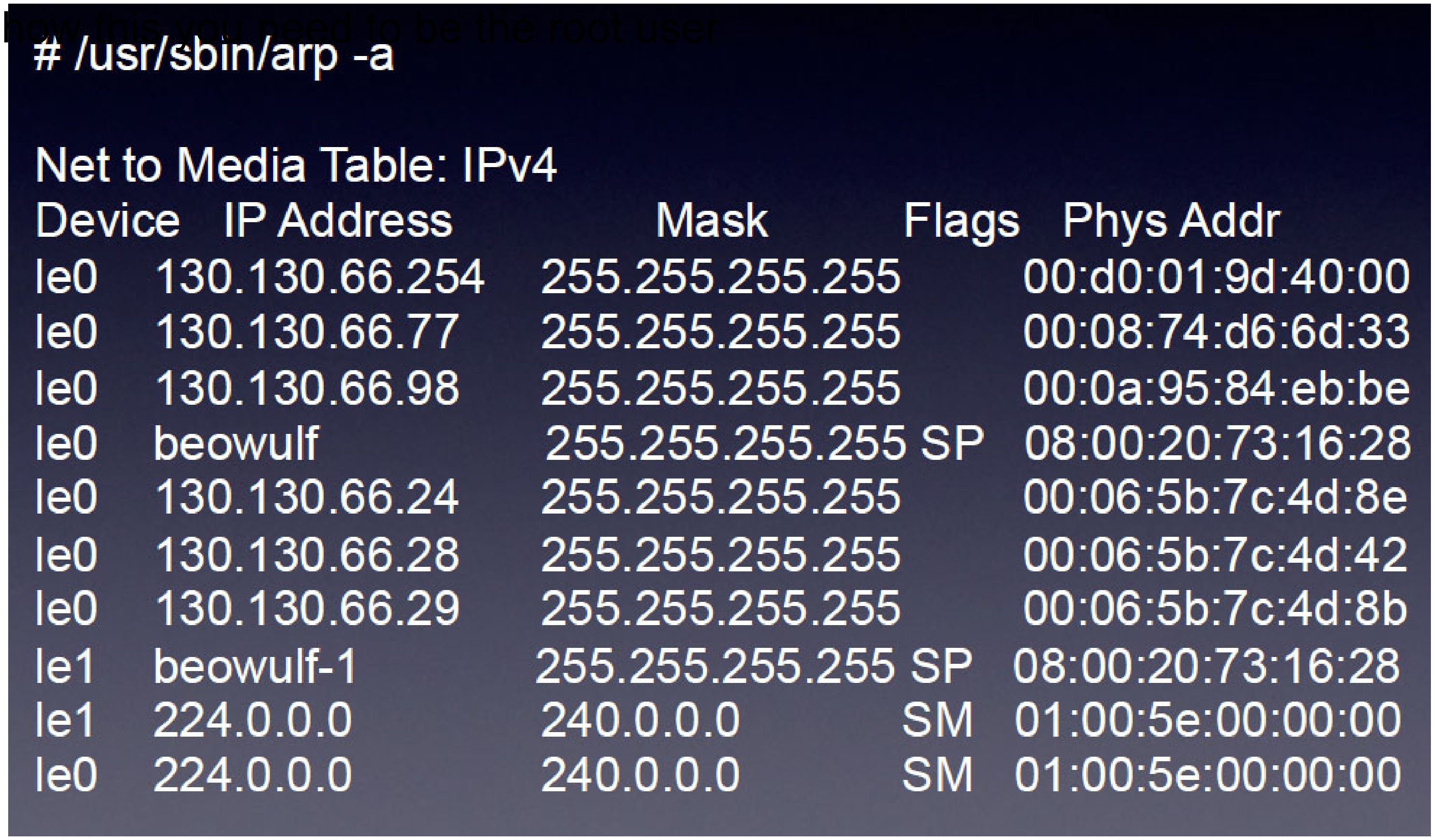
Lecture 5

Address resolution

- When a packet needs to be transmitted to another machine (on the same subnet), the sender needs to be able to determine the hardware (Ethernet) address of the destination.
- This is done via a broadcast asking "who has IP address X".
- This is an ARP (address resolution protocol) packet.
- Machines keep a cache of IP to MAC address translations in the arp table.
- This prevents having to send ARP requests too often.

Here is the *arp* table on a Sun Workstation. Typically to show this you need to be the root user



```
# /usr/sbin/arp -a

Net to Media Table: IPv4
Device   IP Address              Mask            Flags   Phys Addr
le0    130.130.66.254    255.255.255.255          00:d0:01:9d:40:00
le0    130.130.66.77     255.255.255.255          00:08:74:d6:6d:33
le0    130.130.66.98     255.255.255.255          00:0a:95:84:eb:be
le0    beowulf           255.255.255.255 SP       08:00:20:73:16:28
le0    130.130.66.24     255.255.255.255          00:06:5b:7c:4d:8e
le0    130.130.66.28     255.255.255.255          00:06:5b:7c:4d:42
le0    130.130.66.29     255.255.255.255          00:06:5b:7c:4d:8b
le1    beowulf-1         255.255.255.255 SP       08:00:20:73:16:28
le1    224.0.0.0         240.0.0.0       SM       01:00:5e:00:00:00
le0    224.0.0.0         240.0.0.0       SM       01:00:5e:00:00:00
```

- You can save the need for your system to do these ARP broadcasts to find a hosts MAC level address by modifying the file /etc/ethers.
- The /etc/ethers file contains mappings of IP numbers to MAC addresses

RARP

- Reverse Address Resolution Protocol (RARP) can be used by a machine that does not know its own IP address.
- It uses a machine to find out its address.
- To do this it normally consults /etc/ethers.
- People these days tend to stay away from RARP and use DHCP/ BOOTP instead (Dynamic Host Configuration Protocol) (Boot Protocol).

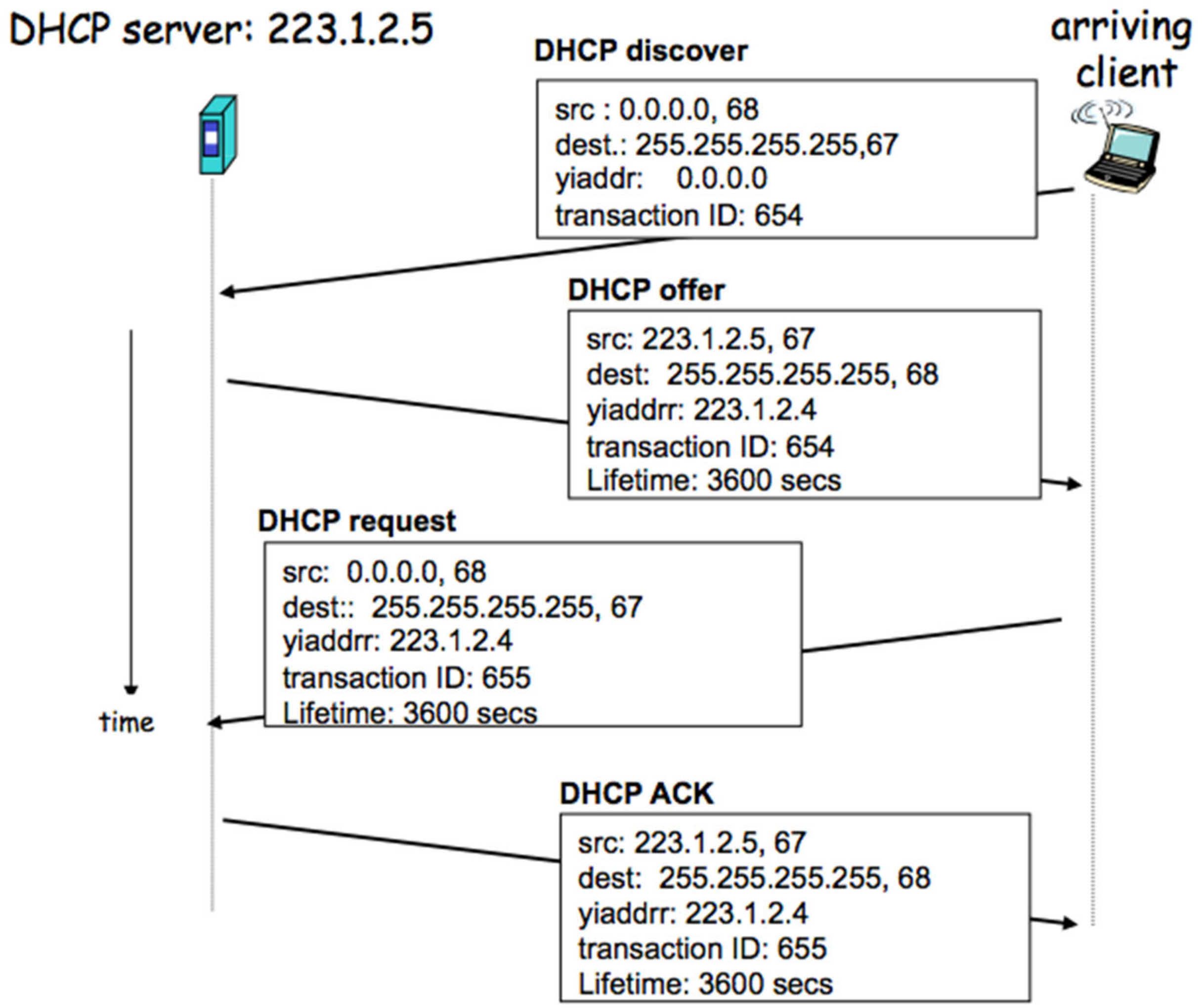
DHCP

- DHCP – Dynamic Host Control Protocol
- Described by RFC 2131 and 2132
- DHCP is a “plug and play” service that operates when your device first accesses a network
- DHCP provides your device with
  - a unique IP address (that belongs to the subnet you are on)
  - an address mask
  - the gateway address(es) of the router(s)
  - DNS address(es)
  - and other features such as Syslog server, WINS (Windows Internet Name) Server, Proxy Servers and NTP (network time protocol) servers which are all described in the above RFCs.

- DHCP overview:
  - host broadcasts “DHCP discover” msg
  - DHCP server responds with “DHCP offer” msg
  - host requests IP address: “DHCP request” msg
  - DHCP server sends address: “DHCP ack” msg
- DHCP discover – The arriving host first needs to find a DHCP server. This is done using the DHCP discover message which is a UDP packet with a source port address of 68 and destination port of 67. The IP datagram has destination address of 255.255.255.255 and source address of 0.0.0.0.

- DHCP Server offer(s) - A DHCP server receiving a DHCP discover message responds with a DHCP offer message. Its possible that more than two servers can respond with an offer. Each offer message contains a transaction ID from the received discover message. Each offer message contains an IP address, mask, and IP address lease time – the amount of time for which the IP address will be valid. It is common for the lease time to be set for several hours or days. This is sent back to the host as a broadcast but has port 68 as its (now) destination port
- DHCP Request message – the host will choose from one of the offer messages and send back a DHCP request message back to the relevant DHCP server echoing back the configuration parameters

- DHCP ACK – The servers responds with an ACK message. Once the client receives the ACK message the transaction is complete and the host can use the allocated IP address for the lease duration.
- There is a mechanism that enables the host to renew its lease on an IP address.
- It is also possible for DHCP servers to be found on another subnet – DHCP relay message is used to relay information across routers
- It is also possible to allocate a permanent IP address to a machine (using their MAC address)


DHCP server: 223.1.2.5
arriving client
DHCP discover
src : 0.0.0.0, 68
dest.: 255.255.255.255,67
yiaddr: 0.0.0.0
transaction ID: 654
DHCP offer
src: 223.1.2.5, 67
dest: 255.255.255.255, 68
yiaddrr: 223.1.2.4
transaction ID: 654
Lifetime: 3600 secs
DHCP request
src: 0.0.0.0, 68
dest:: 255.255.255.255, 67
yiaddrr: 223.1.2.4
transaction ID: 655
Lifetime: 3600 secs
DHCP ACK
src: 223.1.2.5, 67
dest: 255.255.255.255, 68
yiaddrr: 223.1.2.4
transaction ID: 655
Lifetime: 3600 secs
time

DHCP software

- The Internet Systems Consortium (ISC) maintains a range of open source implementations of DHCP (version 2,3, 4)
  - Version 3 – allows for backup DHCP servers
  - Version 4 – IPv6
- LINUX flavours run versions of ISC's DHCP server software (dhcp3-server on Ubuntu for example)
- There are proprietary versions of DHCP.
- Client side DHCP software comes with the device so should not need tampering with

- ISC’s server daemon in LINUX is called dhcpd
- The associated configuration file is called dhcpd.conf
- As is the practice with .conf files you should find it in the /etc directory (or /etc/dhcp3/)
- The file is text but its format is a bit finicky
- Another file is required to hold lease database file. This file is normally found in the /var directory in LINUX.

- To set up a dhcpd.conf file you will need the following information:
- The subnets for which dhcpd should manage IP addresses and the range of addresses it needs to allocate
- A list of static IP addresses assignments you want to make (if any) along with the MAC addresses of the recipients
- The length of initial and maximum lease time durations in seconds
- Any other options the server should pass to clients such as netmasks, default routes, DNS name servers.

```
subnet 192.168.1.0 netmask 255.255.255.0 {
        option routers                  192.168.1.254;
        option subnet-mask              255.255.255.0;

        option domain-name              "example.com";
        option domain-name-servers       192.168.1.1;

        option time-offset              -18000;     # Eastern Standard Time

        range 192.168.1.10 192.168.1.100;
}
```

Example 25-1. Subnet Declaration

```
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2;
option domain-name "example.com";

subnet 192.168.1.0 netmask 255.255.255.0 {
   range 192.168.1.10 192.168.1.100;
}
```

Example 25-4. Range Parameter

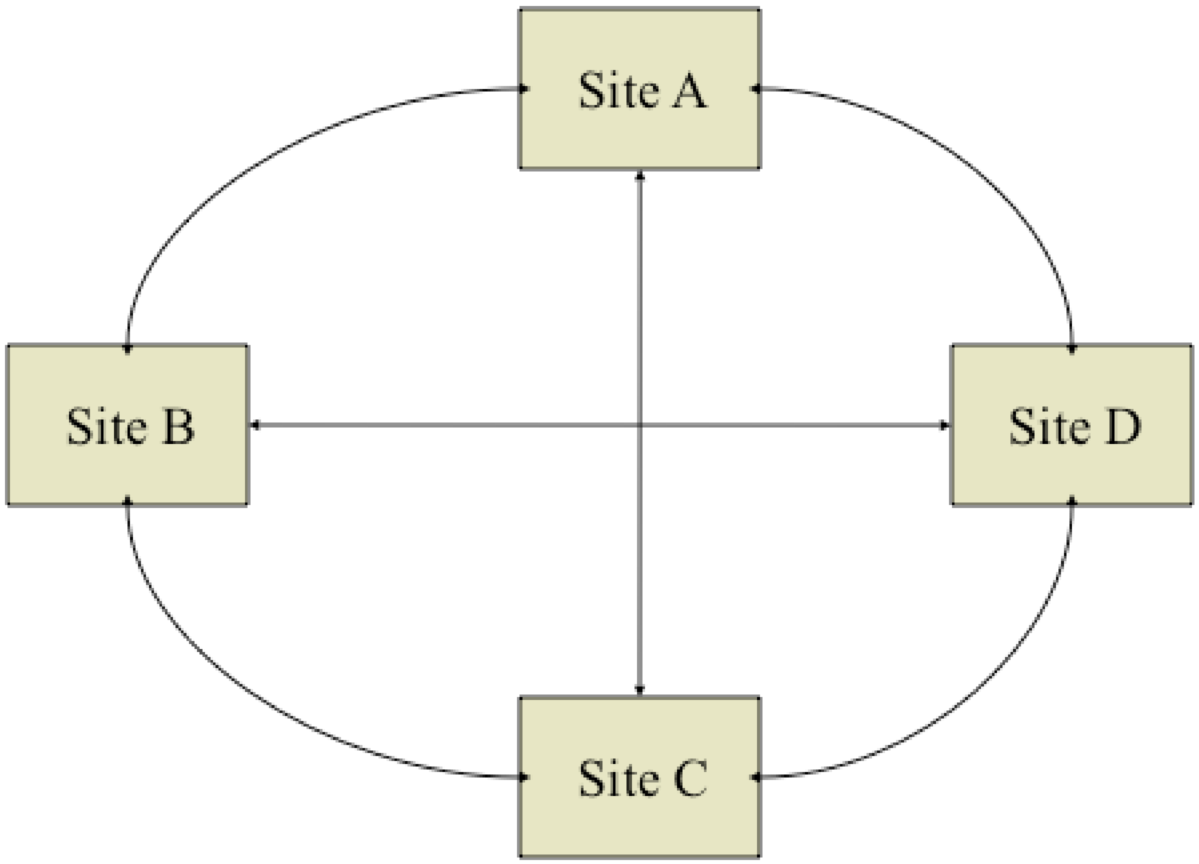
Domain Name System (DNS)

- There are hundreds of countries, millions of networks and millions of hosts.
- How do we keep track of all the nodes and networks.
- Well we use a service called DNS (Domain Name Service).
- The Domain Name Service is a ***distributed database*** which provides a lookup table for ***hostnames*** and ***IP addresses***.
- Humans prefer to refer to nodes on the internet by names, rather than IP addresses. In order for this to work, we need to convert these high level names into addresses. DNS performs name resolution to IP address (and also the opposite direction).
- The Domain Name System also gives us a network hierarchy. It is responsible for giving structure to the Internet (from the users perspective).

The History Behind the Domain Name Service

- When the Internet was first born (ARPANET)- the mapping between Hostnames and IP addresses was done in small files.
- If a site wished to add a new computer to the network, they choose a name and added it to such files.
- These files were distributed to other sites. Such a process consumed considerable bandwidth.
- That said there were consistent problems keeping the data between sites up to date and in sync.
- It soon became clear that this technique simply did not scale.
- This mapping between names and IP addresses is extremely important because as humans we prefer to remember names as opposed to long sequences of numbers.
- On Unix, you have already seen how the `/etc/hosts` file may contain such mappings;

```
121.194.14.142   www.seu.edu.cn
```



- The exchange of host files between sites was too expensive, complex and inefficient.

The History Behind the Domain Name Service

- The Domain Name System was created to address two problems;

  - The organization of the internet.

    As ARPANET grew researchers began to realize that names for hosts were running out. So they needed to impose a hierarchy/ structure.

  - The synchronization of host databases.

    Again as ARPANET grew it become more and more difficult to keep track of nodes. Data quickly became dated.

The History Behind the Domain Name Service

- DNS was formally introduced by Paul Mockapetris in RFC’s 882 and 883 (around 1983).
- The original version of DNS was designed and implemented at the University of California at Berkeley in 1984.
- In 1985 the Berkeley Computer Systems Research Group absorbed the graduate students work and rolled it into the BSD project (Berkeley Software Distribution)
- Kevin Dunlap in 1985 released BIND (Berkeley Internet Name Domain system) which has grown in popularity over the years.
- Bind is currently in release 9 and an organisation called the **Internet Software Consortium** (ISC) has evolved from the work done at Berkeley. The ISC is a non-profit organisation funded by the USA government, community and private sector.

The History Behind the Domain Name Service

- DNS has evolved since those early days and continues to evolve. Initiatives include:
  - DNSSEC
    Secure DNS transactions i.e. the ability to move data between servers in a secure and trusted fashion.
  - IPv6 Support
    With the adoption of IPv6, DNS has had to change only slightly.
- It has also been an interesting time because vendors such as Microsoft have come up with several good ideas which have been rolled back into the DNS system.
  - For example Microsoft has developed the notion of Dynamic DNS which has had a profound impact.

So What *is* DNS?

- In order to understand DNS, you need to understand the Internet.
- The internet is made up of sites. These sites reflect networks or organisations.
- Organisations can be grouped together depending on type.
- Entities can be grouped together based upon geographic locality.
- Ultimately DNS provides the following kind of functionality;
  - A hierarchal namespace for IP addresses.
  - A distributed database of host information.
  - A mechanism for finding services and nodes on a network.
  - A protocol for exchanging naming information to keep data current and synchronised between sites.

The Domain Name System Architecture

- But how does it work?
  - The Domain name system defines a tree of domains. Each domain represents a chunk or grouping of the network which is managed by a single administrative entity.
  - The Domain Name Service is a global infrastructure managed by ICANN.
  - The Domain Name System defines what are known as Top Level Domains (TLDs).
  - These TLDs had not been changed in years. It used to be that certain TLDs represented specific kinds of networks.
  - But as the Internet has grown, ICANN added a few top level domains. And more recently, hundreds more!

*g*TLD

- Top level domains are sometimes referred to in the literature are *g*TLD's (generic top level domains).

Domain	Use	Domain	Use
com	Commercial Entities	net	Network Providers
edu	Education Entities	org	Non profit organisations
gov	Government	int	International
mil	Military Networks	arpa	Old Style Arpanet

*cc*TLD

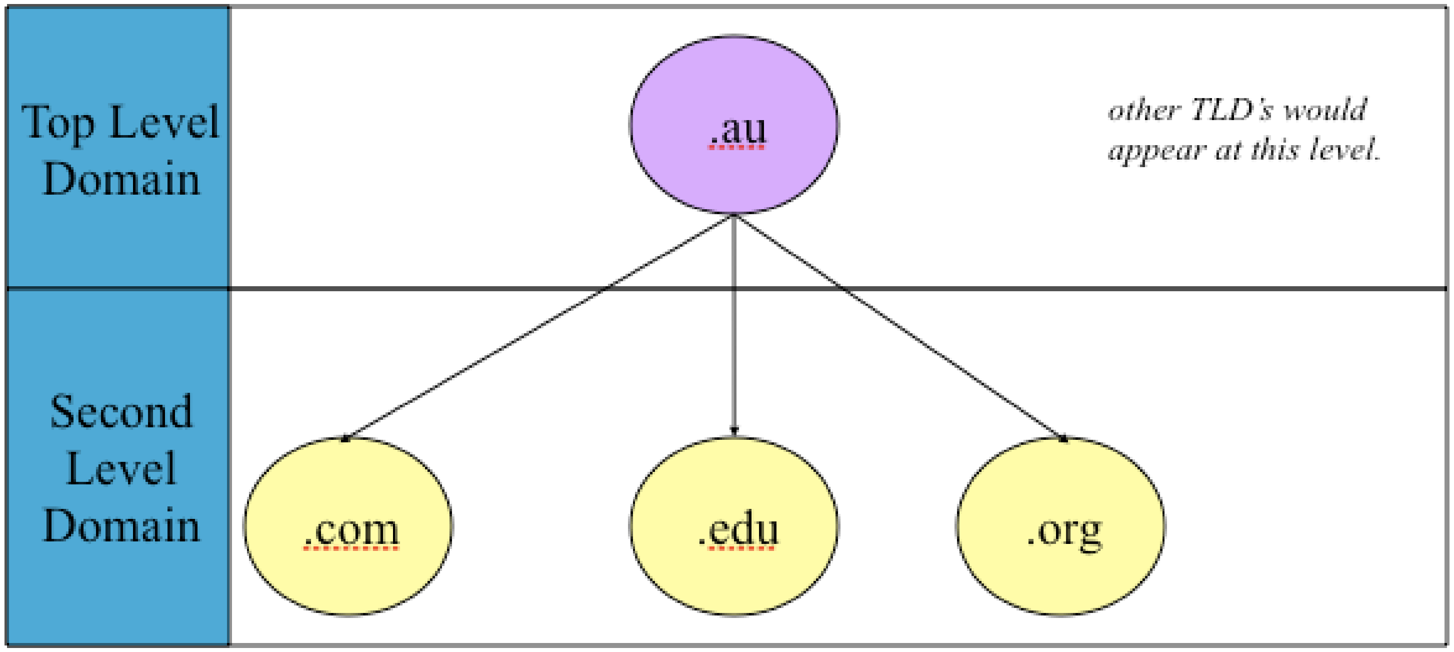
- Most of the top level domains are used within the United States. Outside the United states there are a number of top level country code domains *cc*TLD’s.

Some common country codes are

Code	**County**	**Code**	**Country**	**Code**	**Country**
cn	China	fi	Finaland	hk	Hong Kong
ca	Canada	fr	France	ch	Switzerland
br	Brazil	jp	Japan	mx	Mexico
de	Germany	se	Sweden	mu	Hungary

- Most of these countries produce second level domains which are similar to the top level domains offered by the US.

- For example the internet has the following structure.



- So considering this, **`www.seu.edu.cn`** belongs to the **`seu`** domain which is part of the **`edu`** domain which is in **`cn`**’s namespace.

Subdomain

- You can infer from the DNS model that domains can be delegated to other authorities (sub domains).
- **`.cn`** has its own authority
- And at the University, faculties may have their own subdomain beneath the **`seu`** domain. The management of this **`cs.seu.edu.cn`** domain is partially delegated.

Obtaining a Domain.

- Depending upon where you wish to create a domain you need to work with different agencies.
- ICANN is responsible for delegating ‘authority’ to agencies to handle certain top level domains.
- In China, China Internet Network Information Center (CNNIC) has a registry of all the domain and organisations that are managed.

- Small organisations can run servers on their own hosts or ask their ISP to supply DNS service
- Medium-sized sites should run multiple DNS servers to reduce query latency
- A very large site can divide its DNS domain into subdomains and run several server for each subdomain

Technical Details of DNS

- Inside `seu.edu.cn` there is a sub domain `cs.seu.edu.cn` (plus many others).
- Although it is a sub domain, it is not a delegation point since the `seu.edu.cn` DNS server is authoritative for that domain.
- A DNS server may be authoritative for sub domains or may delegate responsibility to other DNS servers.

Types of DNS

- There are several kinds of DNS Server
  - Authoritative

  An authoritative server is one that is authorised to answer queries for a domain. It is guaranteed to be up to date. It is considered the official representative of a zone/domain.

  - Non-Authoritative.
    A non-authoritative server answers a query typically from a cache. It does not know if the data is still valid.
- In the Authoritative grouping we typically have these kinds of servers
  - Masters (Primary)
  - Slaves (Secondary)
- The slave gets a copy of its records from the master using the SOA (Start of Authority) record for the zone.
- Data is maintained between the master and slave using Zone Transfers (over TCP). We can control zone transfers using software e.g. ndc.
- Non-Authoritative servers typically include things such as caches and forwarders.

Recursive vs Non Recursive Queries

- There are two ways in which DNS servers can answer queries

  - Non-Recursive
    If a DNS server is non-recursive it will only answer queries that it has the answer for. These may be cached or they may be authoritative.

    If it does not have the answer it returns the address of the server that is likely to know the answer.

  - Recursive
    A recursive DNS server returns only names or errors. It follows the referrals to yield the answer.

    The way this is done is by forwarding the query to a *root* server that is known.

Zone

- A zone is essentially a domain minus its subdomains
- Each zone typically has one master name server that keeps the official copy of the zone's data on disk
- A zone can have several slave name servers that get their data from the master server through a zone transfer operation
- Typically we have two zones;
  - Forward zone name to address mapping.
  - Reverse zone address to name mapping.
- You may hear things like catching-only name server, non-authoritative name server, recursive name server, etc.

DNS Records

- The information in the DNS system consists of records

SOA	Start of Authority
NS	Identifies Zones Servers
A	Name to Address Mapping
PTR	Address to Name Mapping
MX	Mail Exchange (used for mail routing)
CNAME	Synonyms for hosts i.e. Nicknames
Glue	A Record for a Nameserver

SOA

- The `SOA` record indicates the beginning of a Zone.
- In many regards a zone can be considered a grouping in the DNS namespace.
- Typically we have two zones
  - Forward zone for name to address mapping.
  - Reverse zone for address to name mapping.
- The `SOA` record typically contains the following
  - The name of the Zone.
  - The administrator for the Zone.
  - Timeout information e.g.
    - `Serial` is typically used to indicate last change.
    - `Refresh` specifies how often slave servers check with the master to see if the serial has changed.
    - `Retry` specifies how long after a failure a slave checks the master.
- The `SOA` information is used by slaves to identify when it is necessary to do a Zone Transfer.
- A Zone Transfer allows master and slave DNS servers to remain in sync.

NS

- The `NS` record identifies which servers are authoritative for a zone. This is how we define master and secondary DNS servers.
- If we do the following we get some interesting results:

```
$ dig seu.edu.cn NS
```

- Each domain should have a primary server and at least one secondary.
- Primary and secondary name servers talk among themselves to ensure the consistency of the database.

A

- The `A` records provide the name to address mappings in the DNS database. They basically represent the `/etc/hosts` file.
- An A record would typically look like this:

```
yoshi IN  A  130.130.64.68
```

This entry would be in the zone file representing the cs subdomain for the University.

PTR

- The `PTR` records do the reverse of the A records.
- A reverse DNS lookup for an IP address is NOT a search of the A records in reverse; it is a search of the PTR records for the IP address.
- PTR records exist in a special part of the DNS namespace called *`in-addr.arpa`*.

In-addr.arpa

- Domains under `in-addr.arpa` are named liked IP addresses with their bytes reversed.

  For example at the University the `cs` zone is (was)

  ```
  130.130.64.0/24
  ```

  Thus the `in-addr.arpa` file will be called *`64.130.130.in-addr.arpa`*.

- It will typically contain records such as:

  ```
  68   PTR   yoshi
  ```

- Some software requires names to be reversed for security purposes e.g. `Telnet` or `SSH`.

MX

- MX records are created to help routing of email.
- A MX record knows where mail should be routed.
- For a domain we can have more then one MX record. The mail router uses the MX record with the lowest priority that is available.

MX Example

- If I run:

```
$ dig seu.edu.cn MX
  ;; ANSWER SECTION:
  seu.edu.cn. 3600 IN  MX  1 voidc50.seu.edu.cn.
```

I will get the `MX` record for the domain which points to the host `voidc50.seu.edu.cn.`

It has a priority of 1.

- If I do the same thing at gmail

```
$ dig gmail.com MX
   ;; ANSWER SECTION:
   gmail.com. 455 IN MX  10 alt1.gmail-smtp-in.l.google.com.
   gmail.com. 455 IN MX   5 gmail-smtp-in.l.google.com.
   gmail.com. 455 IN MX  40 alt4.gmail-smtp-in.l.google.com.
   gmail.com. 455 IN MX  30 alt3.gmail-smtp-in.l.google.com.
   gmail.com. 455 IN MX  20 alt2.gmail-smtp-in.l.google.com.
```

we see five hosts. The priority decides where mail goes. We use this for redundancy purposes.

CNAME

•CNAMES describe canonical names for hosts in the DNS namespace.

In the namespace we will typically see things like this:

```
wyrm        IN    A          130.130.68.3
smtp        IN    CNAME      wyrm
ftp         IN    CNAME      wyrm
```

Other records

- There are a few other records which are important to mention:

  - `SRV` records
    SRV records describe priority, ports and other information for network resources/ hosts. This is particularly important in Zero Configuration networking.

  - `TXT` records
    A TXT record adds arbitrary text to a hosts DNS record.

Adding a new machine to DNS (Bind server)

1. Assign an unused name and address to the new machine
2. Log into the master names server (use *dig* to find it)
3. Find the name server configuration file (usually /etc/named.conf). Find the zone data files in the zone statement, e.g.,

```
zone “example.com” {
   type master;
   file “filename”;
zone “188.77.208.in-addr.arpa” {
   type master;
   file “filename”;
```

4. Identify the record of a similar machine in the same subnet, e.g.,

```
template      IN   A    208.77.188.100
                   IN   MX 10 mail-hub
```

5. Duplicate this record and change it accordingly
6. Edit the reverse zone file

```
100         IN   PTR   template.example.com.
```

7. Change the serial number in the SOA record at the beginning of the file
8. Reload the domain

```
sudo rndc reload forward-zone-filename
sudo rndc reload reverse-zone-filename
```

Hosts on the Network.

- Machines may have many IP addresses, either on different hardware interfaces or the same hardware interface.
- Each IP address may have its own name or not.
- The name of the machine (uname) may not match any interface name.
- Sensible people adopt a sensible naming scheme.
- For example,
    - Machine name matches DNS name for main interface.
    - Each IP address has its own name.
    - Each interface has a single IP address and name.
    - Secondary interface names are derived from the primary name e.g. wraith-1

The Importance of DNS

- Management of their IP addresses and DNS configuration is a complex and onerous task for an organisation of any size.
- Do not underestimate the size of this problem or the complexity of DNS configuration.
- Remember that nearly all network services depend on name resolution.
- ***If DNS fails, most of your services will be down.***
- After your network fabric, DNS is probably the most important network service you will ever have.

The Resolver.

- The resolver on a host is responsible for converting names to addresses. It forwards queries to the appropriate DNS servers.
- Name server configuration is typically via a single configuration file. For Unix, that configuration file is called `/etc/resolv.conf`. (not a typo!)

- Such a file looks like this:

```
domain     seu.edu.cn
nameserver aa.bb.cc.dd
nameserver xx.yy.zz.nn
options    retry:2 retrans:5
search     seu.edu.cn cs.seu.edu.cn
```

`resolv.conf` example

- The file has a number of directives. The first is the Primary DNS domain for the current host.

```
domain  seu.edu.cn
```

- The next group of directives specify the name server to use to resolve queries. Queries are sent to each server in order.

```
nameserver    aa.bb.cc.dd
nameserver    xx.yy.zz.nn
```

- For each server we can specify a series of option. In this case we are saying:
  - `options   retry:2 retrans:5`
  - Wait 5 seconds for a reply.
  - Retry each nameserver 2 times before moving on and trying the next.
- This is a very aggressive configuration.
- If name does not resolve, try these domains as well.
- `search     seu.edu.cn cs.seu.edu.cn`
- This is typically used when we do not provide a fully qualified DNS name.

Multi-homed Hosts

- Machines may have multiple interfaces. We often see these devices in the device tree.
  - `/dev/le0`
  - `/dev/hme4`
- Individual interfaces may be up or down.
- The `ifconfig` command is used to display information about or configure an interface.
- Information displayed includes the hardware address, the IP address, the broadcast address, the netmask, the MTU and some flags.

Important Files

  - Under Solaris, `/etc/hostname.X` controls the name (and therefore the IP address) of each interface.
  - `/etc/nodename` contains the nodename or system name of the machine.
  - Name/IP information comes from `/etc/hosts` (needed since before interface operational, no way to get information from DNS)
  - Netmask information from `/etc/netmasks`
  - Network/name translation from `/etc/networks`

- As a systems administrator you need to remember these files.
- On most Unix implementations they are the same.

Questions?