Networking Basics 5: DNS

1. TCP/IP Introduction
2. Routing and Ports
3. TCP/IP Protocols
4. Ethernet
5. **DNS**
   - **hostnames**
   - **the Domain Name System**
   - **DNS lookups**
   - **DNS records**
   - **Linux networking tools**

Hostnames and DNS

While TCP/IP identifies hosts using numeric IP addresses, humans do better with names.

The **Domain Name System** (**DNS**) was created to map between alphanumeric hostnames and numeric IP addresses.

Hostname: a designator for a particular host (machine) on the network (format depends on the naming system used, such as DNS, SMB, etc.).

Domain name: DNS designation for a realm of authority (e.g., siu.edu), but note that in DNS a hostname is also considered to be domain name.

Hostnames and DNS (contd.)

Fully qualified domain name (FQDN): domain name that uniquely identifies domain within DNS tree (so starts from top level).

A FQDN identifies a host/domain on the Internet, without need for any context: e.g., `www.cs.siu.edu.`
(trailing dot technically required for FQDN).

Domain names are written starting with highest level domain at end (on right) and lowest level at start (on left).

FQDN has a toplevel domain at far right: . (dot).

Hostnames and DNS (contd.)

Host/domain names may be *incomplete*, relying on domain context to uniquely identify a host: e.g., `www`

Target host will depend on domain context, e.g., `www.cs.siu.edu` or `www.siu.edu`, etc.

FQDNs are limited to 255 bytes total and 63 bytes for each (dot separated) **label**.

Labels contain only (Latin) alphabet characters (a-z), digits (0-9), plus the dash/hyphen character (-), and are *not case sensitive*.

ICAAN has approved an *internationalized character* set called **IDNA** (using **Punycode**).

DNS

DNS is a **hierarchical, distributed database**:

- tree structured (hierarchical), **root name servers** at top
- mappings distributed across servers, no single server has all mappings

There is to be at least one **name server** that contains the hostname to IP address mapping for each host on Internet (this is the **authoritative name server** for the hostname).

Each host must also have one or more **local name servers** that it can contact to resolve hostnames.

The so-called **resolver** code is a set of routines that handle requests to map a hostname to an IP address, using local resources (`/etc/hosts`) or initiating DNS lookups to the local name servers.

DNS Lookups

The DNS lookup procedure can be fairly complex.

We will consider an example of trying to connect to `www.cs.siu.edu` from a computer outside of the SIU network:

- the web browser code makes a "**resolver**" call to get the IP address for the hostname `www.cs.siu.edu`
- the resolver code (part of the OS networking stack) sends a DNS query to (first listed) **local name server**
- if local name server knows the IP address (because it is authoritative for the hostname or because it has cached it from a previous lookup), it simply returns the IP address
- if local name server does *not* know the address, it (the name server) automatically makes a series of queries until it can return the IP address...

DNS Lookups (contd.)

...continued...

- the local name server first queries a **root name server** to get a name server for the **toplevel domain** (TLD), `.edu`
- once the TLD server is known, a query is sent to it
- the TLD name server responds with the name server for the next lower domain level, `.siu.edu`
- the local name server then queries that nameserver, which may respond with the IP address if it is authoritative for the host or if it has the address cached, otherwise it will respond with another, lower level name server, `cs.siu.edu`
- the process continues until the local name server reaches an authoritative name server and gets the IP address (or finds that there is no mapping, i.e., the hostname is invalid)

DNS Records

Name servers store information as records, of various **types**:

- **A** – hostname to IPv4 mapping
- **AAAA** – hostname to IPv6 ("quad A")
- **CNAME** – map aliases to canonical name
- **MX** – mail servers for the domain
- **NS** – delegate to subdomain name servers
- **PTR** – IPv4 address to hostname mapping (reverse lookup)

Key organizations/sites:
icann.org, iana.org, root-servers.org

Key Linux Networking Tools

- `netstat` – show network connections
- `ifconfig` – show/configure network interfaces
- `route` – show/configure the IP routing table
- `ip` – show/manipulate network devices, routing, tunnels
- `ping/ping6` – send ICMP echo requests to hosts
- `traceroute` – trace packet route to host
- `arp` – show/configure ARP cache
- `iptables` – show/configure Linux firewall
- `wireshark` – network packet sniffer
- `nmap` – network port scanner