TCP Milestone II: Schedule meeting by this Friday, April 21
- Meeting slots available soon
- Should by sending/receiving with sliding window—no retransmissions/shutdown/close yet
TCP Milestone II Gearup: tonight (April 19), 7pm, CIT368 (and also on Zoom)
HW3: Out soon, short, due next Tuesday
The story so far

Transport layer: send packets to IP:port,
eg. 128.148.10.3 port 80

Is this how users interact with the network? No!
A new abstraction

What we have: **IP Addresses**
- Numerical address appreciated by routers
- Fixed length, binary numbers
- Hierarchical, related to host’s location in the network

Examples: 128.148.32.110, 212.58.224.138

Want: Host names
- Mnemonics appreciated by humans
- Variable length, string characters
- Provide little (if any) information about location

Examples: google.com, www.cs.brown.edu, bbc.co.uk
Separating Naming and Addressing

cs.brown.edu => 128.148.32.110
Separating Naming and Addressing

\[ \text{cs.brown.edu} \rightarrow 128.148.32.110 \]

Why?

- Names are easier to remember
- Addresses can change underneath
  - e.g., renumbering when changing providers
- Useful Multiplexing/sharing
  - One name -> multiple addresses
  - Multiple names -> one address

- CAN ADJUST W/O CHANGING HOW END USERS INTERACT W/ SERVICE
Another Change in Layers...

• Remember ARP
  – ARP: maps IP addresses to MAC addresses

L3 L2

DNS: Human Readable
  Application/Service-Based

Network-Layer Info.
Scalable (Address <-> Name) Mappings
Scalable (Address <-> Name) Mappings

Original way: one file: hosts.txt
Scalable (Address <-> Name) Mappings

Original way: one file: hosts.txt

- Flat namespace
Scalable (Address <-> Name) Mappings

Original way: one file: `hosts.txt`

- Flat namespace
- Central administrator kept master copy (for the Internet)
Scalable (Address <-> Name) Mappings

Original way: one file: hosts.txt

- Flat namespace
- Central administrator kept master copy (for the Internet)
- To add a host, emailed admin
Scalable (Address <-> Name) Mappings

Original way: one file: hosts.txt
  • Flat namespace
  • Central administrator kept master copy (for the Internet)
  • To add a host, emailed admin
  • Downloaded file regularly
<table>
<thead>
<tr>
<th>Host Name</th>
<th>Host Address</th>
<th>Sponsor</th>
<th>Liaison</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC</td>
<td>16.2.0.54</td>
<td>VDH ARPA</td>
<td>Lockwood, Gregory (LOCKWOOD@BBNC) Associated Computer Consultants 414 East Cota Street Santa Barbara, California 93101 (805) 965-1023</td>
</tr>
<tr>
<td>CPU Type: PDP-11/70(UNIX)</td>
<td></td>
<td></td>
<td>McBride, William T. (MCBRIDE@USC-ISI) Naval Ocean Systems Center Code 8321 271 Catalina Boulevard San Diego, California 92152 (714) 225-2883 (AV) 953-2083</td>
</tr>
<tr>
<td>ACCAT-TIP</td>
<td>10.2.0.55</td>
<td>ARPA</td>
<td>Nelson, Louis C. (LOUSAEROSPACE) Aerospace Corporation AZ/1013 P.O. Box 92957 Los Angeles, California 90009 (213) 615-4424</td>
</tr>
<tr>
<td>CPU Type: H-316</td>
<td></td>
<td></td>
<td>Nelson, Louis C. (LOUSAEROSPACE) Aerospace Corporation AZ/1013 P.O. Box 92957 Los Angeles, California 90009 (213) 615-4424</td>
</tr>
<tr>
<td>AEROSPACE</td>
<td>10.2.0.65</td>
<td>AFSC</td>
<td>Nelson, Louis C. (LOUSAEROSPACE) Aerospace Corporation AZ/1013 P.O. Box 92957 Los Angeles, California 90009 (213) 615-4424</td>
</tr>
<tr>
<td>CPU Type: VAX-11/780(UNIX)</td>
<td></td>
<td></td>
<td>Cosentino, Antonio (COSENTINO@AFSC-HQ) Air Force Geophysics Laboratory SUNA Mail Stop 30 Hanscom Air Force Base, Massachusetts 01731 (617) 861-4161 (AV) 478-4161</td>
</tr>
<tr>
<td>AFGL</td>
<td>10.1.0.66</td>
<td>AFSC</td>
<td>Cosentino, Antonio (COSENTINO@AFSC-HQ) Air Force Geophysics Laboratory SUNA Mail Stop 30 Hanscom Air Force Base, Massachusetts 01731 (617) 861-4161 (AV) 478-4161</td>
</tr>
<tr>
<td>CPU Type: PDP-11/50(RSX11M) -&gt; CDC-6600(NOS/BE)</td>
<td></td>
<td></td>
<td>Cosentino, Antonio (COSENTINO@AFSC-HQ) Air Force Geophysics Laboratory SUNA Mail Stop 30 Hanscom Air Force Base, Massachusetts 01731 (617) 861-4161 (AV) 478-4161</td>
</tr>
<tr>
<td>AFGL-TAC</td>
<td>10.2.0.66</td>
<td>AFSC</td>
<td>Cosentino, Antonio (COSENTINO@AFSC-HQ) Air Force Geophysics Laboratory SUNA Mail Stop 30 Hanscom Air Force Base, Massachusetts 01731 (617) 861-4161 (AV) 478-4161</td>
</tr>
<tr>
<td>CPU Type: C/30</td>
<td></td>
<td></td>
<td>Cosentino, Antonio (COSENTINO@AFSC-HQ) Air Force Geophysics Laboratory SUNA Mail Stop 30 Hanscom Air Force Base, Massachusetts 01731 (617) 861-4161 (AV) 478-4161</td>
</tr>
</tbody>
</table>
Scalable (Address <-> Name) Mappings
Scalable (Address <-> Name) Mappings

Original way: one file: hosts.txt
Scalable (Address <-> Name) Mappings

Original way: one file: hosts.txt
- Flat namespace
Scalable (Address <-> Name) Mappings

Original way: one file: hosts.txt

- Flat namespace
- Central administrator kept master copy (for the Internet)
Scalable (Address <--> Name) Mappings

Original way: one file: `hosts.txt`
- Flat namespace
- Central administrator kept master copy (for the Internet)
- To add a host, emailed admin
Scalable (Address <-> Name) Mappings

Original way: one file: hosts.txt

- Flat namespace
- Central administrator kept master copy (for the Internet)
- To add a host, emailed admin
- Downloaded file regularly
Scalable (Address <-> Name) Mappings

Original way: one file: hosts.txt
- Flat namespace
- Central administrator kept master copy (for the Internet)
- To add a host, emailed admin
- Downloaded file regularly

Is this feasible today? Lol no.
Domain Name System (DNS)

- Originally proposed by RFC882, RFC883 (1983)
- Distributed key-value store, before it was cool

- Distributed protocol to translate hostnames -> IP addresses
  - Human-readable names
  - Load-balancing/content delivery
  - So much more…
Goals for DNS

- Scalability
  - Must handle a huge number of records
    - With some software synthesizing names on the fly
  - Must sustain update and lookup load
Goals for DNS

• Scalability
  – Must handle a huge number of records
    • With some software synthesizing names on the fly
  – Must sustain update and lookup load

• Distributed Control
  – Let people control their own names
Goals for DNS

- **Scalability**
  - Must handle a huge number of records
    - With some software synthesizing names on the fly
  - Must sustain update and lookup load

- **Distributed Control**
  - Let people control their own names

- **Fault Tolerance**
  - Minimize lookup failures in face of other network problems
The good news
The good news

- Properties that make these goals easier to achieve
  1. Read-mostly database
     Lookups MUCH more frequent than updates
  2. Loose consistency
     When adding a machine, not end of the world if it takes minutes or hours to propagate
The good news

• Properties that make these goals easier to achieve
  1. Read-mostly database
     Lookups MUCH more frequent than updates
  2. Loose consistency
     When adding a machine, not end of the world if it takes minutes or hours to propagate

• These suggest aggressive caching
  – Once you’ve lookup up a hostname, remember
  – Don’t have to look again in the near future
How it works

Hierarchical namespace broken into zones

From there, one system can allocate more names and more hierarchy.

Top-level domain

Root namespace

One system

Names can be arbitrarily long.

Owners can further delegate names to continue.
How it works

• Hierarchical namespace broken into zones
  – root (.), edu., brown.edu., cs.brown.edu.,
  – Zones separately administered :: delegation
  – Parent zone tells you how to find servers for subdomains
• Each zone served from multiple replicated servers
• Lots and lots of caching
DNS Architecture

- Hierarchy of DNS servers
  - Root servers
  - Top-level domain (TLD) servers
  - Authoritative DNS servers
- Two “types” of DNS servers


DNS Architecture

- Hierarchy of DNS servers
  - Root servers
  - Top-level domain (TLD) servers
  - Authoritative DNS servers

- Two “types” of DNS servers (may overlap)
  - Authoritative servers: “owners” of certain DNS records
  - Resolvers: process lookups, caches authoritative records
Resolver operation

- Apps make **recursive** queries to local DNS server (1)
  - Ask server to get answer for you
- Server makes **iterative** queries to remote servers (2, 4, 6)
  - Ask servers who to ask next
  - Cache results aggressively
Want: CS, Brown, etc. 

This is called a recursive query — no caching.

- Can take a long time (100s of ms)

- Caching!
RESIDENTIAL DNS

- Given to your router w/ DHCP
- Next level cache
- Maybe recursive

VZON DNS

ROUTE
DNS
SERVER

- Local
- Cache
- Not Recursive

LAPTOP
- Local Resolver
- Another Cache
- 192.168.1.100
- IP: ~ DNS: 192.168.1.1
- Network
- Gateway
DNS Root Server

- Located in New York
- How do we make the root scale?

Verisign, New York, NY
DNS Root Servers

- 13 Root Servers (www.root-servers.org)
  - Labeled A through M (e.g., A.ROOT-SERVERS.NET)
- Does this scale?
DNS Root Servers

- 13 Root Servers (www.root-servers.org)
  - Labeled A through M (e.g., A.ROOT-SERVERS.NET)
- Remember anycast?

A Verisign, New York, NY (also Frankfurt, HK, London, LA)
B USC-ISI Marina del Rey, CA (plus 157 other locations)
C Cogent, Herndon, VA (also Los Angeles, NY, Chicago, Frankfurt and 3+)
D U Maryland College Park, MD (also in 106 other locations)
E NASA Mt View, CA (+70)
F Internet Software Consortium, Palo Alto, CA (and 57 other locations)
G US DoD Columbus, OH (+5) K RIPE London (plus 41 other locations)
H ARL Aberdeen, MD (also San Diego)
J Verisign (118 locations)
L ICANN Los Angeles, CA (plus 157 other locations)
M WIDE Tokyo plus Seoul, Paris, San Francisco, Osaka

ONE IP ADVERTISED FROM MULTIPLE PLACES (VIA BGP)
DNS Root Servers

- 13 Root Servers (www.root-servers.org)
  - Labeled A through M (e.g., A.ROOT-SERVERS.NET)
- Remember anycast?

- A Verisign, New York, NY (also Frankfurt, HK, London, LA)
- C Cogent, Herndon, VA (also Los Angeles, NY, Chicago, Frankfurt and 3+)
- D U Maryland College Park, MD (also in 106 other locations)
- G US DoD Columbus, OH (+5)
- H ARL Aberdeen, MD (also San Diego)
- J Verisign (118 locations)
- E NASA Mt View, CA (+70)
- F Internet Software Consortium, Palo Alto, CA (and 57 other locations)
- B USC-ISI Marina del Rey, CA
- L ICANN Los Angeles, CA (plus 157 other locations)
- I Netnod, Stockholm (plus 49 other locations)
- M WIDE Tokyo plus Seoul, Paris, San Francisco, Osaka

- 13 Root Servers (www.root-servers.org)
  - Labeled A through M (e.g., A.ROOT-SERVERS.NET)
- Remember anycast?

- A Verisign, New York, NY (also Frankfurt, HK, London, LA)
- C Cogent, Herndon, VA (also Los Angeles, NY, Chicago, Frankfurt and 3+)
- D U Maryland College Park, MD (also in 106 other locations)
- G US DoD Columbus, OH (+5)
- H ARL Aberdeen, MD (also San Diego)
- J Verisign (118 locations)
- E NASA Mt View, CA (+70)
- F Internet Software Consortium, Palo Alto, CA (and 57 other locations)
- B USC-ISI Marina del Rey, CA
- L ICANN Los Angeles, CA (plus 157 other locations)
- I Netnod, Stockholm (plus 49 other locations)
- M WIDE Tokyo plus Seoul, Paris, San Francisco, Osaka

- 13 Root Servers (www.root-servers.org)
  - Labeled A through M (e.g., A.ROOT-SERVERS.NET)
- Remember anycast?

- A Verisign, New York, NY (also Frankfurt, HK, London, LA)
- C Cogent, Herndon, VA (also Los Angeles, NY, Chicago, Frankfurt and 3+)
- D U Maryland College Park, MD (also in 106 other locations)
- G US DoD Columbus, OH (+5)
- H ARL Aberdeen, MD (also San Diego)
- J Verisign (118 locations)
- E NASA Mt View, CA (+70)
- F Internet Software Consortium, Palo Alto, CA (and 57 other locations)
- B USC-ISI Marina del Rey, CA
- L ICANN Los Angeles, CA (plus 157 other locations)
- I Netnod, Stockholm (plus 49 other locations)
- M WIDE Tokyo plus Seoul, Paris, San Francisco, Osaka
DNS Root Servers: Today

From: www.root-servers.org
“Types” of DNS servers

• Top Level Domain (TLD) servers
  – Generic domains (e.g., com, org, edu)
  – Country domains (e.g., uk, br, tv, in, ly)
  – Special domains (e.g., arpa)
  – Corporate domains (…)
TYPE OF DNS SERVICES

- Authoritative
- Organization/ISP to local-only names
- Public DNS
  - Google: 8.8.8.8 8.8.4.4
  - Cloudflare: 1.1.1.1 1.1.2.2
SEWA
COMPANY
South
DNS
COMPANY
MEN
MINS
DEN
LON
I
CLIENT
CLIENT
OPTIMIZED
DNS
FOR
CONTENT
DELIVERY
"Types" of DNS servers

• Top Level Domain (TLD) servers
  – Generic domains (e.g., com, org, edu)
  – Country domains (e.g., uk, br, tv, in, ly)
  – Special domains (e.g., arpa)
  – Corporate domains (…)

• Authoritative DNS servers
  – Provides public records for hosts at an organization
  – Can be maintained locally or by a service provider
“Types” of DNS servers

- **Top Level Domain (TLD) servers**
  - Generic domains (e.g., com, org, edu)
  - Country domains (e.g., uk, br, tv, in, ly)
  - Special domains (e.g., arpa)
  - Corporate domains (...)

- **Authoritative DNS servers**
  - Provides public records for hosts at an organization
  - Can be maintained locally or by a service provider

- **Recursive resolvers**
  - Big public servers, or local to a network
  - Lots of caching
DNS Caching

- Recursive queries are expensive
- Caching greatly reduces overhead
  - Top level servers very rarely change
  - Popular sites visited often
  - Local DNS server caches information from many users
- How long do you store a cached response?
  - Original server tells you: TTL entry
  - Server deletes entry after TTL expires
Negative Caching
Negative Caching

• Remember things that don’t work
  – Misspellings like www.cnn.comm, ww.cnn.com
  – Is the cost of these two queries the same?
Negative Caching

- Remember things that don’t work
  - Misspellings like www.cnn.comm, ww.cnn.com
  - Is the cost of these two queries the same?

- These can take a long time to fail the first time
  - Good to cache negative results so it will fail faster next time
Negative Caching

• Remember things that don’t work
  – Misspellings like www.cnn.com, ww.cnn.com
  – Is the cost of these two queries the same?

• These can take a long time to fail the first time
  – Good to cache negative results so it will fail faster next time
Negative Caching

• Remember things that don’t work
  – Misspellings like www.cnn.comm, ww.cnn.com
  – Is the cost of these two queries the same?
• These can take a long time to fail the first time
  – Good to cache negative results so it will fail faster next time
• But negative caching is optional, and not widely implemented
Reverse DNS

How do we get the other direction, IP address to name?
Reverse DNS

How do we get the other direction, IP address to name?

- Addresses have a natural hierarchy:
  - 128.148.32.12

- Idea: reverse the numbers: 12.32.148.128 ...
  - and look that up in DNS

- Under what TLD?
  - Convention: in-addr.arpa
  - Lookup 12.32.148.128.in-addr.arpa
  - in6.arpa for IPv6
DNS Protocol

• TCP/UDP port 53
• Most traffic uses UDP
  – Lightweight protocol has 512 byte message limit
  – Retry using TCP if UDP fails (e.g., reply truncated)
• TCP requires messages boundaries
  – Prefix all messages with 16-bit length
• Bit in query determines if query is recursive
Example

% dig cs.brown.edu @10.1.1.10

; <<>> DiG 9.10.6 <<>> cs.brown.edu @10.1.1.10
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 8536
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1220
;; QUESTION SECTION:
cs.brown.edu. IN A

;; ANSWER SECTION:
cs.brown.edu. 1800 IN A 128.148.32.12

;; Query time: 69 msec
;; SERVER: 10.1.1.10#53(10.1.1.10)
;; WHEN: Tue Apr 19 09:03:39 EDT 2022
;; MSG SIZE  rcvd: 57
Example

dig . ns

dig +norec www.cs.brown.edu @a.root-servers.net

dig +norec www.cs.brown.edu @a.edu-servers.net

dig +norec www.cs.brown.edu @bru-ns1.brown.edu

www.cs.brown.edu. 86400 IN A 128.148.32.110
Resource Records

• All DNS info represented as resource records (RR)

  name [ttl] [class] type rdata

  – name: domain name
  – TTL: time to live in seconds
  – class: for extensibility, normally IN (1) “Internet”
  – type: type of the record
  – rdata: resource data dependent on the type

• Important RR types
  – A – Internet Address (IPv4);   AAAA – IPv6
  – NS – name server;

• Example RRs

  www.cs.brown.edu.  86400  IN  A   128.148.32.110
  cs.brown.edu.      86400  IN  NS   dns.cs.brown.edu.
  cs.brown.edu.      86400  IN  NS   ns1.ucsb.edu.
Some important details

• How do local servers find root servers?
  – DNS lookup on a.root-servers.net?
  – Servers configured with root cache file
  – Contains root name servers and their addresses
    .                       3600000    IN    NS    A.ROOT-SERVERS.NET.
    A.ROOT-SERVERS.NET.    3600000    A    198.41.0.4
    ...

• How do you get addresses of other name servers?
  – To obtain the address of www.cs.brown.edu, ask a.edu-servers.net, says a.root-servers.net
  – How do you find a.edu-servers.net?
  – Glue records: A records in parent zone
Other uses of DNS

• Local multicast DNS
  – Used for service discovery
  – Made popular by Apple
  – This is how you learn of different Apple TVs in the building

• Load balancing

• CDNs
Structure of a DNS Message

- Same format for queries and replies
  - Query has 0 RRs in Answer/Authority/Additional
  - Reply includes question, plus has RRs
- Authority allows for delegation
- Additional for glue, other RRs client might need

```
+------------------+
|       Header     |
+------------------+
| Question         | the question for the name server
+------------------+
| Answer           | RRs answering the question
+------------------+
| Authority        | RRs pointing toward an authority
+------------------+
| Additional       | RRs holding additional information
+------------------+
```
Header format

- **Id**: match response to query; **QR**: 0 query/1 response
- **RCODE**: error code.
- **AA**: authoritative answer, **TC**: truncated,
- **RD**: recursion desired, **RA**: recursion available
Other RR Types

- **CNAME (canonical name):** specifies an alias
  
  
  - `www.l.google.com. 300 IN A 72.14.204.147`

- **MX record:** specifies servers to handle mail for a domain (the part after the @ in email addr)
  - Different for historical reasons

- **SOA (start of authority)**
  - Information about a DNS zone and the server responsible for the zone

- **PTR (reverse lookup)**
  
  - `7.34.148.128.in-addr.arpa. 86400 IN PTR quanto.cs.brown.edu.`
Reliability

- Answers may contain several alternate servers
- Try alternate servers on timeout
  - Exponential backoff when retrying same server
- Use same identifier for all queries
  - Don’t care which server responds, take first answer
Inserting a Record in DNS

Your new startup helpme.com
Inserting a Record in DNS

- Your new startup helpme.com
- Get a block of addresses from ISP
  - Say 212.44.9.0/24
- Register helpme.com at namecheap.com (for ex.)
  - Provide name and address of your authoritative name server (primary and secondary)
  - Registrar inserts RR pair into the .com TLD server:
    - helpme.com NS dns1.helpme.com
    - dns1.helpme.com A 212.44.9.120
- Configure your authoritative server (dns1.helpme.com)
  - Type A record for www.helpme.com
  - Type MX record for helpme.com
Inserting a Record in DNS, cont

- Need to provide reverse PTR bindings
  - E.g., 212.44.9.120 -> dns1.helpme.com
- Configure your dns server to serve the 9.44.212.in-addr.arpa zone
  - Need to add a record of this NS into the parent zone (44.212.in-addr.arpa)
- Insert the bindings into the 9.44.212.in-addr.arpa zone
DNS Security

• You go to starbucks, how does your browser find www.google.com?
  – Ask local name server, obtained from DHCP

• How can you know you are getting correct data?
  ➔ By default, no authentication/verification of responses.
DNS $\rightarrow$ Resolver can lie!
    Insert Bad Records

$H \rightarrow$ Local DNS $\rightarrow$ Authentic DNS

$\leftarrow$ DNS
DNS: 8.8.8.8
Alternatif: 8.8.4.4