Current Events

1.35 Tb/s DDoS on GitHub

https://www.wired.com/story/github-ddos-memcached/
https://githubengineering.com/ddos-incident-report/
Current Events

Reminder: The Cloud is just other people’s computers.

AWS US-EAST-1 Region downtime leads to outages and connectivity degradation for Atlassian’s Bitbucket, Confluence, and Jira, GitHub, MongoDB, NewVoiceMedia, Slack, Twilio, Zillow.

https://is.gd/gvI38X
Keeping track...

- http://www.devopsweekly.com/
- https://sreweekly.com/
- https://www.nanog.org/
- https://puck.nether.net/mailman/listinfo/outages
In the beginning...
In the beginning...
In the beginning...
In the beginning...
In the beginning...
In the beginning...

https://is.gd/DdPNCo
In the beginning...

# Host Database
# This file should contain the addresses and aliases
# for local hosts that share this file.
#
127.0.0.1 localhost localhost.
#
# RFC 1918 specifies that these networks are "internal".
# 10.0.0.0 10.255.255.255
# 172.16.0.0 172.31.255.255
# 192.168.0.0 192.168.255.255
10.0.0.1 UCLA-TEST
10.0.0.2 SRI-SPRM
10.0.0.4 UTAH-CS
But then...
The Domain Name System

Computers like numbers.

1001101111101100101100110011111
The Domain Name System

Computers like numbers.

10011011  11110110  01011001  10011111

155    .    246    .    89    .    159
The Domain Name System

People like names.

ash.cs.stevens-tech.edu
The Domain Name System

DNS; HTTP

March 5, 2018
The New Phonebook is here!

https://is.gd/XXp2sC

wget -q -0 - https://is.gd/XXp2sC | grep -c "^HOST"
DNS: A distributed database
The domain name space consists of a tree of *domain* names.
DNS: A hierarchical system
The Domain Name Space

The domain name space consists of a tree of *domain* names.

A subtree divides into *zones*. 
The domain name space consists of a tree of domain names.

A subtree divides into zones.

Each node may contain resource records.
The Domain Name Space

- resource records associated with name
- zone of authority, managed by a name server
- delegated subzone
  - When a system administrator wants to let another administrator manage a part of a zone, the first administrator's name server delegates part of the zone to another name server.
  - NS RR ("resource record") names the name server authoritative for delegated subzone

see also: RFC 1034 4.2: How the database is divided into zones.
Domain Names

ash.cs.stevens-tech.edu

Domain Names are read from right to left and components separated by a “.”.
Domain Names

ash.cs.stevens-tech.edu.

The *root* is known as “.”, but is usually left out.
Domain Names

ash.cs.stevens-tech.edu.

There is a small number of top level domains.
Domain Names

ash.cs.stevens-tech.edu.

There is a number of top level domains.

wget -O - ftp://rs.internic.net/domain/root.zone | \
grep "IN<tab>*NS<tab>" | awk '{print $1}' | sort -u | wc -l

https://data.iana.org/TLD/tlds-alpha-by-domain.txt
Domain Names

ash.cs.stevens-tech.edu.

Each *domain* can be divided into any number of *sub domains*. 
Domain Names

ash.cs.stevens-tech.edu.

Each *domain* can be divided into any number of *sub domains*.
Domain Names

ash.cs.stevens-tech.edu.

The left-most component of a domain name may be a *hostname*. 
A hostname with a domain name is known as a **FQDN**.

ash.cs.stevens-tech.edu.
The Original IANA
NIC and Network Solutions

Before the DNS, the Network Information Center (NIC) at Stanford Research Institute (SRI) allocated domain names. IANA (effectively: Jon Postel) assigned, NIC published.

https://www.internic.net

In 1991, this was contracted out to Network Solutions, Inc. (NSI), which held the monopoly on DNS registrations (within .com, .org, .mil, .gov, .edu, and .net) until around 1998.
Registries

IANA manages the root zone (.), arpa.; gTLD registries handle gTLDs, ccTLD registries handle ccTLDs. ICANN accredits domain name registries.

- may function as a Domain Name Registrar
- may delegate Domain Name registration
- control policies of allocations
- can (and do) censor, revoke, change, ... entries (e.g. vb.ly)

The domain name space is a tree; if you control one node, you control all the branches and subtrees.
DNS servers come in two flavors

- Authoritative Nameservers
- Recursive Nameservers
Hostname resolution

Resolution on a recursive nameserver (aka *resolver*) involves a number of queries:

```
$ nslookup ash.cs.stevens-tech.edu
Server:    127.0.0.1
Address:   127.0.0.1#53

Non-authoritative answer:
Name: ash.cs.stevens-tech.edu
Address: 155.246.89.159
```

$
Hostname resolution

Resolution on a *resolver* involves a number of queries:

IP panix.netmeister.org.62105 > i.root-servers.net.domain:
   11585 [1au] A? ash.cs.stevens-tech.edu. (52)
IP i.root-servers.net.domain > panix.netmeister.org.62105:
   11585- 0/8/8 (494)
IP panix.netmeister.org.53168 > a.gtld-servers.net.domain:
   46575 [1au] A? ash.cs.stevens-tech.edu. (52)
IP a.gtld-servers.net.domain > panix.netmeister.org.53168:
   46575- 0/6/3 (609)
IP panix.netmeister.org.41071 > nrac.stevens-tech.edu.domain:
   24322 [1au] A? ash.cs.stevens-tech.edu. (52)
IP nrac.stevens-tech.edu.domain > panix.netmeister.org.41071:
   24322*- 1/2/3 A[|domain]
Hostname resolution

Resolution on a *resolver* involves a number of queries:

```
$ host -t ns .
  . name server I.ROOT-SERVERS.NET.
  . name server D.ROOT-SERVERS.NET.
  . name server C.ROOT-SERVERS.NET.
  . name server M.ROOT-SERVERS.NET.
  . name server F.ROOT-SERVERS.NET.
  . name server A.ROOT-SERVERS.NET.
  . name server E.ROOT-SERVERS.NET.
  . name server L.ROOT-SERVERS.NET.
  . name server H.ROOT-SERVERS.NET.
  . name server J.ROOT-SERVERS.NET.
  . name server B.ROOT-SERVERS.NET.
  . name server G.ROOT-SERVERS.NET.
  . name server K.ROOT-SERVERS.NET.
$
```
Hostname resolution

Resolution on a *resolver* involves a number of queries:

```
$ dig -t ns edu.
[...]
;; ANSWER SECTION:
edu. 172800 IN NS l.edu-servers.net.
edu. 172800 IN NS f.edu-servers.net.
edu. 172800 IN NS c.edu-servers.net.
edu. 172800 IN NS g.edu-servers.net.
edu. 172800 IN NS a.edu-servers.net.
edu. 172800 IN NS d.edu-servers.net.

;; ADDITIONAL SECTION:
c.edu-servers.net. 36626 IN A 192.26.92.30
d.edu-servers.net. 13274 IN A 192.31.80.30
l.edu-servers.net. 36626 IN A 192.41.162.30
[...]
$```

DNS; HTTP

March 5, 2018
Hostname resolution

Resolution on a *resolver* involves a number of queries:

```
$ dig @c.edu-servers.net -t ns stevens.edu.
[...]
;; AUTHORITY SECTION:
stevens.edu. 172800 IN NS nrac.stevens-tech.edu.
stevens.edu. 172800 IN NS sitult.stevens-tech.edu.

;; ADDITIONAL SECTION:
nrac.stevens-tech.edu. 172800 IN A 155.246.1.21
sitult.stevens-tech.edu. 172800 IN A 155.246.1.20
[...] 
$
```
Hostname resolution

- Client
- Recursive DNS
- Root DNS
- .edu DNS
- nrac.stevens-tech.edu
- ash.cs.stevens-tech.edu
Hostname resolution

Resolution on a resolver involves a number of queries:

$ nslookup ash.cs.stevens-tech.edu
Server: 127.0.0.1
Address: 127.0.0.1

Non-authoritative answer:
Name: ash.cs.stevens-tech.edu
Address: 155.246.89.159

$
Hostname resolution
Hostname resolution

$ ftp -o - ftp.internic.net:/domain/db.cache | more
https://www.internic.net/zones/named.root
Operation Global Blackout

https://pastebin.com/XZ3EGsbc
DNS: A distributed system

There are 13 root servers.
DNS: A distributed system

There are 13 root servers.

Except... there are more.
DNS: A distributed system

There are 13 root authorities.
DNS: A distributed system

There are 13 root server addresses.
DNS: A distributed system

There are hundreds of root servers.
DNS: A distributed system

See e.g.: https://e.root-servers.org/
Operation Global Blackout

@anonops

GlobalBlackOut is another Fake Operation. No intention of #Anonymous to cut Internet.
DNS: A distributed database
DNS Resource Records

More than just A and AAAA:

- **CAA** – certificate authority authorization
- **CNAME** – the canonical name for an alias
- **MX** – mail exchange
- **NS** – an authoritative name server
- **SOA** – marks the start of a zone of authority
- **SRV** – service locator (e.g. for kerberos)
- **PTR** – a domain name pointer
- **TXT** text strings
- ...
DNS Resource Records

You’ve all seen PTR records:

```
$ host ash.cs.stevens-tech.edu
ash.cs.stevens-tech.edu has address 155.246.89.159
ash.cs.stevens-tech.edu mail is handled by 0 guinness.cs.stevens-tech.edu.
$ host 155.246.89.159
159.89.246.155.in-addr.arpa domain name pointer ash.cs.stevens-tech.edu.
$
```

Stevens doesn’t have write access to the `in-addr.arpa` domain. How does this work?
Creative uses of DNS Resource Records

- identifying sources of SPAM (via e.g. an RBL)
- detect email spoofing (via e.g. SPF)
- find out if the internet is on fire:
  ```
  dig +short txt istheinternetonfire.com
  ```
- find ASN numbers by IP addresses:
  ```
  dig +short 159.89.246.155.origin.asn.cymru.com TXT
  ```
- check a resolver's source port randomization (to help mitigate DNS Cache Poisoning attacks):
  ```
  dig +short porttest.dns-oarc.net TXT
  ```
- using DNS to publish SSH key fingerprints (RFC4255, ssh_config(5) VerifyHostKeyDNS; for best results combine with DNSSEC)
Hooray!

5 Minute Break
Hypertext Transfer Protocol

Today’s Universal Internet Pipe
HTTP: Hypertext

WWW

“The World Wide Web is the only thing I know of whose shortened form takes three times longer to say than what it’s short for.” – Douglas Adams
HTTP: Hypertext

This proposal concerns the management of general information about accelerators and experiments at CERN. It discusses the problems of loss of information about complex evolving systems and derives a solution based on a distributed hypertext system.

Keywords: Hypertext, Computer conferencing, Document retrieval, Information management, Project control

https://is.gd/JnZaN6
HTTP

Hypertext Transfer Protocol

RFC2616
HTTP is a request/response protocol.
The Hypertext Transfer Protocol

HTTP is a request/response protocol:

1. client sends a request to the server
2. server responds
The Hypertext Transfer Protocol

HTTP is a request/response protocol:

1. client sends a request to the server
   - request method
   - URI
   - protocol version
   - request modifiers
   - client information

2. server responds
HTTP: A client request

$ telnet www.google.com 80
Trying 173.194.75.147...
Connected to www.google.com.
Escape character is '^[].
GET / HTTP/1.0
The Hypertext Transfer Protocol

HTTP is a request/response protocol:

1. client sends a request to the server
   - request method
   - URI
   - protocol version
   - request modifiers
   - client information

2. server responds
   - status line (including success or error code)
   - server information
   - entity metainformation
   - content
HTTP: a server response

HTTP/1.0 200 OK
Date: Sun, 31 Mar 2013 01:54:40 GMT
Set-Cookie: PREF=ID=c5eb56d629b347cc:FF=0:TM=1364694880:LM=1364694880:
S=sIdRFdxV9YvtQ01G; expires=Tue, 31-Mar-2015 01:54:40 GMT; path=/;
domain=.google.com
Set-Cookie: NID=67=hvBn0ob2NoZW4haTJVfajbcyn_jips501KRe-8nawzdCZ6AukNR
_s8CNHD6ZA-Z2721nA3TpLnXt-2zyIui23j4kdsoF8Gg--PmGsM0J3Jv5frEzQG1elHJv92HL-w2;
expires=Mon, 30-Sep-2013 01:54:40 GMT; path=/; domain=.google.com; HttpOnly
Server: gws

<!doctype html><html itemscope="itemscope" itemtype="http://schema.org/WebPage">
<head><meta content="Search the..."
The Hypertext Transfer Protocol

Server status codes:

- **1xx** – Informational; Request received, continuing process
- **2xx** – Success; The action was successfully received, understood, and accepted
- **3xx** – Redirection; Further action must be taken in order to complete the request
- **4xx** – Client Error; The request contains bad syntax or cannot be fulfilled
- **5xx** – Server Error; The server failed to fulfill an apparently valid request
$ telnet www.cs.stevens.edu 80
Trying 155.246.89.84...
Connected to www.cs.stevens-tech.edu.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.1 301 Moved Permanently
Date: Mon, 05 Mar 2018 20:41:06 GMT
Server: Apache
Location: https://www.cs.stevens.edu/
Vary: Accept-Encoding
Content-Length: 235
Connection: close
Content-Type: text/html; charset=iso-8859-1
HTTP: A client request

$ printf "HEAD / HTTP/1.1\r\nHost: www.cs.stevens.edu\r\n\n" |

HTTP/1.1 302 Found
Date: Mon, 05 Mar 2018 20:53:38 GMT
Server: Apache
Location: https://www.stevens.edu/ses/cs
Vary: Accept-Encoding
Content-Type: text/html; charset=iso-8859-1
HTTP: A client request

$ printf "HEAD /ses/cs HTTP/1.1\nHost: www.stevens.edu\n\n" | openssl s_client -quiet -ign_eof -connect www.stevens.edu:443 2>/dev/null

HTTP/1.1 301 Moved Permanently
Date: Mon, 05 Mar 2018 20:54:51 GMT
Content-Type: text/html; charset=UTF-8
Location: https://www.stevens.edu/schaefer-school-engineering-science/departments/computer-science
HTTP: A client request

```
$ printf "HEAD /schaefer-school-engineering-science/departments/computer-science HTTP/1.1\r\nHost: www.stevens.edu\r\n\openssl s_client -quiet -ign_eof -connect www.stevens.edu:443 2>/dev/null
```

HTTP/1.1 200 OK
Date: Mon, 05 Mar 2018 20:56:37 GMT
Content-Type: text/html; charset=utf-8
Connection: keep-alive
Expires: Sun, 19 Nov 1978 05:00:00 GMT
Last-Modified: Mon, 05 Mar 2018 16:44:39 GMT

[...]
HTTP: A client request
HTTP - more than just text

HTTP is a *Transfer Protocol* – serving *data*, not any specific text format.

- **Accept-Encoding** client header can specify different formats such as gzip or deflate for compression etc. communications, etc.

- **corresponding server headers**: **Content-Type** and **Content-Encoding**
HTTP - more than just static data

HTTP is a *Transfer Protocol* – what is transferred need not be static; resources may generate different data to return based on many variables.

- CGI – resource is *executed*, needs to generate appropriate response headers
- server-side scripting (ASP, PHP, Perl, ...)
- client-side scripting (JavaScript/ECMAScript/JScript,...)
- applications based on HTTP, using:
  - AJAX
  - RESTful services
  - JSON, XML, YAML to represent state and abstract information
HTTP Proxy Servers

- HTTP traffic usually is very asymmetric
- a lot of the content is static
- network ACLs may restrict traffic flow
HTTP overload

Ways to mitigate HTTP overload:

- DNS round-robin to many web servers
- load balancing
- web cache / accelerators (reverse proxies)
- content delivery networks

These solutions depend on the location within the network and the scale of the environment.
Load Balancing

Client PC → Internet → Load Balancer → Servers

Incoming Requests → Divided Requests
Load Balancing: Inbound

Client PC 172.10.12.24

Layer 2 Switch

Local Router 192.168.0.1/24

VIP: 192.168.0.200:80

Load Balancer 192.168.0.10/24

src: 192.168.0.10
dst: 192.168.0.102

Real Servers

192.168.0.100
192.168.0.101
192.168.0.102
Load Balancing: Outbound

- Client PC
  - IP: 172.10.12.24
- Load Balancer
  - VIP: 192.168.0.200:80
  - IP: 192.168.0.10/24
- Layer 2 Switch
- Local Router
  - IP: 192.168.0.1/24
- Server to Client (by way of load balancer)
- Real Servers
  - IP: 192.168.0.100
  - IP: 192.168.0.101
  - IP: 192.168.0.102

Traffic flow:
- Client PC sends request to VIP: 192.168.0.200:80
- Load Balancer directs request to Layer 2 Switch
- Layer 2 Switch forwards request to local router
- Local router directs request to real server(s)
- Real server(s) respond to client PC
Load Balancing: Direct Server Return

1. Client (1.1.1.1) sends a request to the Internet.
2. The request is received by the Load Balancer (VIP: 2.2.2.2, VIP MAC: AAAA).
3. The Load Balancer forwards the request to the server (Loopback IP: 2.2.2.2, NIC IP: 3.3.3.3, NIC MAC: BBBB).
4. The server (1.1.1.1) sends the response to the Load Balancer (2.2.2.2).
5. The Load Balancer forwards the response to the client (1.1.1.1).

DNS; HTTP
Content Delivery Networks

DNS; HTTP
Content Delivery Networks

- cache content in strategic locations
- determine location to serve from via geomapping of IP addresses (beware IPv6 aggregation!)
- often uses a separate domain to distinguish small objects/large objects or dynamic content/static content
- either out-sourced or in-house (if your organization is a Tier-1 or Tier-2 peering partner)
- request routing happens via Global Server Load Balancing, DNS-based request routing, anycasting etc.
- provides vast amounts of interesting data about your clients (see https://www.akamai.com/stateoftheinternet/)
CDN Implications

- your CDN sees all your traffic
- your CDN controls your TLS certificate keys
- your CDN is a multi-tenant environment
- your CDN may impose restrictions on your clients
- separation of cache-able content may require multiple (second-level) domains
HTTP and DNS

Both HTTP and DNS are trivial to set up.

Both HTTP and DNS are not trivial to get right.
Reading

HTTP etc.:

- RFC 2616, 2818, 3875
- https://httpd.apache.org/docs/
- https://www.w3.org/Protocols/
- REST: https://is.gd/leSvGa
- CDNs: https://is.gd/R5DoxA
  - https://www.edgecast.com/
  - https://aws.amazon.com/cloudfront/
  - https://www.akamai.com/
  - https://www.limelight.com/
  - ...