CSCI-1680
DNS
Nick DeMarinis

Based partly on lecture notes by Rodrigo Fonseca, Scott Shenker and John Jannotti
Breathe

i am a tiny cactus
and i believe
in you

you can do the thing
TCP officially due tonight (Tuesday, Nov 22)

- Office hours 3-5pm (Me; Zoom); 5-7pm (Rhea, Nathan; CIT201/Zoom)
- Like with IP: you can continue to make small bugfixes after the deadline
  - OK: Fixing small bugs, README, capture files, code cleanup
  - Not OK: eg. implementing sendfile/recvfile, connection teardown, submitting broken/untested code
- Grading meetings: Week of 11/28 (slots available after break)

If you want to submit late:
Monday 11/28 by 11:59pm EST => one day late
• Final project: Out after break, document online now
  – …maybe skim it before break?
  – Group assignment form: Due Tuesday, 11/29 (out after class)
    • Keep your current groups, or form new ones
  – Project proposal: Due Friday, 12/2
  – Final submission: Due Monday, 12/12

More details after break => but remember this is supposed to be light!
IPoAC
How can we improve the physical layer?

Traditional links have fixed bandwidth
- Media limits what frequencies can be used for signal
- Places upper bound on channel capacity
What if we weren’t constrained by the EM spectrum?

How else can we transmit data?
RFC1149: IPoAC

IP over Avian Carriers (1 April 1990)

- High delay, low throughput, low altitude datagram service
- Nearly unlimited movement in 3D etherspace
- Intrinsic collision avoidance
- Typical MTU: 256 milligrams
IPoAC: Design
IPoAC: Implementation

Proof of concept: 28 April 2001
Bergen, Norway
$ ping -c 9 -i 900 10.0.3.1
PING 10.0.3.1 (10.0.3.1): 56 data bytes
64 bytes from 10.0.3.1: icmp_seq=0 ttl=255 time=6165731.1 ms
64 bytes from 10.0.3.1: icmp_seq=4 ttl=255 time=3211900.8 ms
64 bytes from 10.0.3.1: icmp_seq=2 ttl=255 time=5124922.8 ms
64 bytes from 10.0.3.1: icmp_seq=1 ttl=255 time=6388671.9 ms

--- 10.0.3.1 ping statistics ---
9 packets transmitted, 4 packets received, 55% packet loss round-trip
min/avg/max = 3211900.8/5222806.6/6388671.9 ms
IPoAC: (more) Modern implementations

Pigeon-powered Internet takes flight

One of the Internet’s newest forms of life: transmitting no data.

Today: microSD card: ~250mg, 1TB

Pigeon carries data bundles faster than Telkom

Stephen Shankland
Jan. 2, 2002 4:43 p.m. PT

Staff Reporter 10 Sep 2009

= ???
But actually

What happens if you have a LOT of data to move into the cloud?
But actually

What happens if you have a LOT of data to move into the cloud? Example: AWS
## Feature comparison matrix

<table>
<thead>
<tr>
<th></th>
<th>AWS SNOWCONE</th>
<th>AWS SNOWBALL EDGE STORAGE OPTIMIZED</th>
<th>AWS SNOWBALL EDGE COMPUTE OPTIMIZED</th>
<th>AWS SNOWMOBILE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usable HDD Storage</td>
<td>8 TB</td>
<td>80 TB</td>
<td>N/A</td>
<td>100 PB</td>
</tr>
<tr>
<td>Usable SSD Storage</td>
<td>14 TB</td>
<td>1 TB</td>
<td>28 TB</td>
<td>No</td>
</tr>
<tr>
<td>Usable vCPUs</td>
<td>4 vCPUs</td>
<td>40 vCPUs</td>
<td>104 vCPUs</td>
<td>N/A</td>
</tr>
<tr>
<td>Usable Memory</td>
<td>4 GB</td>
<td>80 GB</td>
<td>416 GB</td>
<td>N/A</td>
</tr>
<tr>
<td>Device Size</td>
<td>9in x 6in x 3in</td>
<td>548 mm x 320 mm x 501 mm</td>
<td>548 mm x 320 mm x 501 mm</td>
<td>45 ft. shipping container</td>
</tr>
<tr>
<td>Device Weight</td>
<td>4.5 lbs. (2.1 kg)</td>
<td>49.7 lbs. (22.3 kg)</td>
<td>49.7 lbs. (22.3 kg)</td>
<td>N/A</td>
</tr>
<tr>
<td>Storage Clustering</td>
<td>No</td>
<td>Yes, 5-10 nodes</td>
<td>Yes, 5-10 nodes</td>
<td>N/A</td>
</tr>
<tr>
<td>256-bit Encryption</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>HIPAA Compliant</td>
<td>No</td>
<td>Yes, eligible</td>
<td>Yes, eligible</td>
<td>Yes, eligible</td>
</tr>
</tbody>
</table>
### RFC791: IPv4 Header

<table>
<thead>
<tr>
<th>Version</th>
<th>IHL</th>
<th>Type of Service</th>
<th>Total Length</th>
<th>Identification</th>
<th>Flags</th>
<th>Fragment Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-----------------</td>
<td>--------------</td>
<td>---------------</td>
<td>-------</td>
<td>-----------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-----------------</td>
<td>--------------</td>
<td>---------------</td>
<td>-------</td>
<td>-----------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-----------------</td>
<td>--------------</td>
<td>---------------</td>
<td>-------</td>
<td>-----------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-----------------</td>
<td>--------------</td>
<td>---------------</td>
<td>-------</td>
<td>-----------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time to Live</th>
<th>Protocol</th>
<th>Header Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Destination Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

The Internet Header Format [RFC-791]
# IP over Burrito Carriers

<table>
<thead>
<tr>
<th>Obvious</th>
<th>Onion</th>
<th>Jalapenos</th>
<th>Physical Length (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number Written on Foil</td>
<td>Bean Type</td>
<td>Number of Beans</td>
<td></td>
</tr>
<tr>
<td>Given Delivery Time</td>
<td>Guacamole</td>
<td>Receipt</td>
<td></td>
</tr>
<tr>
<td>Lettuce</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rice</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Beef</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The Burrito Internet Header Format
April Fool’s Day RFCs

April Fools’ Day Request for Comments

From Wikipedia, the free encyclopedia
(Redirected from Peg DHCP)

A Request for Comments (RFC), in the context of Internet governance, is a type of publication from the Internet Engineering Task Force (IETF) and the Internet behaviors, research, or innovations applicable to the working of the Internet and Internet-connected systems.

Almost every April Fool’s Day (1 April) since 1989, the Internet RFC Editor has published one or more humorous Request for Comments (RFC) documents, for RFC 527 called ARPAWOCKY, a parody of Lewis Carroll’s nonsense poem "Jabberwocky". The following list also includes humorous RFCs published on other dates.

Contents [hide]

1 List of April Fools’ RFCs
2 Other humorous RFCs
3 Non-RFC IETF humor
4 Submission of April Fools’ Day RFCs
5 References
6 Further reading
7 External links

List of April Fools’ RFCs  [edit]

1978


A parody of the TCP/IP documentation style. For a long time it was specially marked in the RFC index with "note date of issue".

1989

https://en.wikipedia.org/wiki/April_Fools%27_Day_Request_for_Comments

Enjoy!
How it works

Hierarchical namespace broken into zones

cslab1a.cs.brown.edu
• 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
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
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
How a resolver works
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
Where is the 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?
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 (…)

• 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

• 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?

• 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
$ 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: 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

• 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.
# DNS record types

<table>
<thead>
<tr>
<th>RR Type</th>
<th>Purpose</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>IPv4 Address</td>
<td>128.148.56.2</td>
</tr>
<tr>
<td>AAAA</td>
<td>IPv6 Address</td>
<td>2001:470:8956:20::1</td>
</tr>
<tr>
<td>CNAME</td>
<td>Specifies an alias (&quot;Canonical name&quot;)</td>
<td>systems.cs.brown.edu. 86400 IN CNAME systems-v3.cs.brown.edu. systems-v3.cs.brown.edu. 86400 IN A 128.148.36.51</td>
</tr>
<tr>
<td>MX</td>
<td>Mail servers</td>
<td>MX &lt;priority&gt; &lt;ip&gt; eg. MX 10 1.2.3.4</td>
</tr>
<tr>
<td>SOA</td>
<td>Start of authority</td>
<td>Information about who owns a zone</td>
</tr>
<tr>
<td>PTR</td>
<td>Reverse IP lookup</td>
<td>7.34.148.128.in-addr.arpa. 86400 IN PTR quanto.cs.brown.edu.</td>
</tr>
<tr>
<td>SRV</td>
<td>How to reach specific services (eg. host, port)</td>
<td>_minecraft._tcp.example.net 3600 SRV &lt;priority&gt; &lt;weight&gt; &lt;port&gt; &lt;server IP&gt;</td>
</tr>
</tbody>
</table>

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 (more on this later)
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

• Can you trust this DNS server?
DNS: 8.8.8.8, 8.8.4.4
Alternative: 8.8.4.4
Great Firewall of CIT

If attacker is on the path (say, it is the ISP, or a malicious version of TStaff), what could they do?

- Can sniff all DNS queries
- Send fake responses back first
- Could do this selectively, to direct facebook.com to cs.brown.edu, for example…
Great Firewall of CIT

If attacker is on the path (say, it is the ISP, or a malicious version of TStaff), what could they do?
https://blog.thousandeyes.com/monitoring-dns-in-china/
“Helpful” ISPs

• Many ISPs hijack NXDOMAIN responses to “help” by offering search and advertisement related to the domain
• E.g., www.bicycleisntadomain.com doesn’t (currently) exist
  – Could return a page with search and ads on bicycles (or domain registrations?)
What can be done?

Some defenses against DNS spoofing/hijacking
Some defenses against DNS spoofing/hijacking

- **DNSSEC**: protocol to sign/verify hierarchy of DNS lookups
  - Expensive to deploy, hierarchy must support at all levels
  - APNIC DNSSEC monitor: [https://stats.labs.apnic.net/dnssec](https://stats.labs.apnic.net/dnssec)

- **Tunneling DNS**: client uses DNS via more secure protocol
  - DNS over HTTPS
  - DNS over TLS
More on DNS
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 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.