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
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
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
Administrivia

• 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!
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?
What if we weren’t constrained by the EM spectrum?

How else can we transmit data?
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
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

NIC → UNMIE PACKET
  ↓
  TIE IT TO PIDBOX

PRINT OUT PACKET
  ↓
  TIE IT TO PIDBOX
  ↓
  NIC
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
Pigeon-powered Internet takes flight

One of the Internet's most audacious ideas is about to take flight: transmitting network data through birds. http://www.cnet.com/news/pigeon-carries-data-bundles-faster-than-telkom/
Today: microSD card: ~250mg, 1TB
Today: microSD card: ~250mg, 1TB
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>Feature</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>9 in x 6 in x 3 in</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

The Internet Header Format [RFC-791]
### The Burrito Internet Header Format

<table>
<thead>
<tr>
<th>Obvious 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>
</tr>
<tr>
<td>Given Delivery Time</td>
<td>Guacamole</td>
<td>Receipt</td>
</tr>
<tr>
<td>Lettuce</td>
<td>Rice</td>
<td>Beef</td>
</tr>
</tbody>
</table>
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 Fools' 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.

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!
DNS
Hierarchical namespace broken into zones

```
cslab1a.cs.brown.edu
```

- **Root**
- **Top Level Domain (TLD)**
- **Sub Domain**
- **Name of Some Org/Entity**
- **Name of Host/Service**
- **CS**
- **Managed by Brown**

**Original TLDs**
- com, org, net, gov, edu
- ISO Country Codes
- Now > 1500!
Two types of DNS servers

- "Authoritative" server
  holds/maintains records for some domain(s)

- Resolution: you query it to look up name => finds
  finds authoritative server of answer
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 (may overlap)
  – Authoritative servers: “owners” of certain DNS records
  – Resolvers: process lookups, caches authoritative records
How a (recursive) DNS query works

Want: CS, Brown, EDU

Root server

DNS name server for EDU is at 1.2.3.4

NS Brown?

CS?

Brown server

128.147...

→ This is a recursive DNS query (no caching)

→ Can take a long time (100+ or ms)

→ Caching!
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