2- Application Level Protocols
Where we are in the Course

Application
Transport
Network
Link
Physical
Implementation

• Application layer protocols are often part of “the app”
  – Libraries running in untrusted space
Application Communication Needs

• Vary widely; must build on Transport services

Web
- Series of variable length, reliable request/reply exchanges
- TCP

DNS
- Short, reliable request/reply exchanges
- UDP

Skype
- Real-time (unreliable) stream delivery
- UDP
OSI Session/Presentation Layers

• Two relevant concepts ...

Considered part of the application, not strictly layered!

<table>
<thead>
<tr>
<th>Layer</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Application</td>
</tr>
<tr>
<td>6</td>
<td>Presentation</td>
</tr>
<tr>
<td>5</td>
<td>Session</td>
</tr>
<tr>
<td>4</td>
<td>Transport</td>
</tr>
<tr>
<td>3</td>
<td>Network</td>
</tr>
<tr>
<td>2</td>
<td>Data link</td>
</tr>
<tr>
<td>1</td>
<td>Physical</td>
</tr>
</tbody>
</table>

- Provides functions needed by users
- Converts different representations
- Manages task dialogs
- Provides end-to-end delivery
- Sends packets over multiple links
- Sends frames of information
- Sends bits as signals
Session Concept

• A session is a series of related network interactions in support of an application task
  — Often informal, not explicit
  — Often related to an individual user

• Examples:
  — Web page fetches multiple resources
  — Skype call involves audio, video, chat
Presentation Concept

- Apps need to identify the *type of content*, and *encode it for transfer*
  - These are Presentation functions
- Examples:
  - Media (MIME) types, e.g., image/jpeg, identify content type
  - Transfer encodings, e.g., gzip, identify the encoding of content
  - Application headers are simple/readable versus packed for efficiency
Which Apps Matter?

Much of the content here is from
Which Devices Matter?

Graph of connected devices

* Figures (n) refer to 2017, 2022 device share
Source: Cisco VNI Global IP Traffic Forecast, 2017–2022
Which Devices Matter?

Graph of traffic generated by devices

* Figures (n) refer to 2017, 2022 traffic share
Source: Cisco VNI Global IP Traffic Forecast, 2017–2022
M2M Connections By Industry

19% CAGR 2017–2022

Billions of M2M Connections

Source: Cisco VNI Global IP Traffic Forecast, 2017–2022
Application Data

26% CAGR 2017-2022

Exabytes per Month

* Figures (n) refer to 2017, 2022 traffic share
Source: Cisco VNI Global IP Traffic Forecast, 2017-2022
Peak to Average Load

[Graph showing the growth of Busy Hour Internet Traffic (CAGR 37%) and Average Internet Traffic (CAGR 30%) from 2017 to 2022.

Source: Cisco VNI Global IP Traffic Forecast, 2017-2022]
DOMAIN NAME SYSTEM
DNS

• Human-readable host names, and more
Names and Addresses

- **Names** are higher-level identifiers for resources
- **Addresses** are lower-level locators for resources
  - Multiple levels, e.g. full name → email → IP address → Ethernet addr
- **Resolution** (or lookup) is mapping a name to an address

- Name, e.g. “Andy Tanenbaum,” or “flits.cs.vu.nl”
- Address, e.g. “Vrijie Universiteit, Amsterdam” or IPv4 “130.30.27.38”
Before the DNS – HOSTS.TXT

• Directory was a file HOSTS.TXT regularly retrieved for all hosts from a central machine at the NIC (Network Information Center)
• Names were initially flat, became hierarchical (e.g., lcs.mit.edu) ~1985
• Not manageable or efficient as the ARPANET grew ...
DNS

• A naming service to map between host names and their IP addresses (and more)
  – www.uwa.edu.au → 130.95.128.140

• Goals:
  – Easy to manage (esp. with multiple parties)
  – Efficient (good performance, few resources)

• Approach:
  – Distributed directory based on a hierarchical namespace
  – Automated protocol to tie pieces together
DNS Namespace

- Hierarchical, starting from “.” (dot, typically omitted)
TLDs (Top-Level Domains)

• Run by ICANN (Internet Corp. for Assigned Names and Numbers)
  – Starting in ‘98; naming is financial, political, and international 😊
• 700+ generic TLDs
  – Initially .com, .edu, .gov, .mil, .org, .net
  – Unrestricted (.com) vs Restricted (.edu)
  – Added regions (.asia, .kiwi), Brands (.apple), Sponsored (.aero) in 2012
• ~250 country code TLDs
  – Two letters, e.g., “.au”, plus international characters since 2010
  – Widely commercialized, e.g., .tv (Tuvalu)
  – Many domain hacks, e.g., instagr.am (Armenia), kurti.sh (St. Helena)
DNS Zones

- A zone is a contiguous portion of the namespace
DNS Zones (2)

• Zones are the basis for distribution
  – EDU Registrar administers .edu
  – UW administers washington.edu
  – CSE administers cs.washington.edu

• Each zone has a nameserver to contact for information about it
  – Zone must include contacts for delegations, e.g., .edu knows nameserver for washington.edu
DNS Resource Records

- A zone is comprised of DNS resource records that give information for its domain names

<table>
<thead>
<tr>
<th>Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOA</td>
<td>Start of authority, has key zone parameters</td>
</tr>
<tr>
<td>A</td>
<td>IPv4 address of a host</td>
</tr>
<tr>
<td>AAAA (“quad A”)</td>
<td>IPv6 address of a host</td>
</tr>
<tr>
<td>CNAME</td>
<td>Canonical name for an alias</td>
</tr>
<tr>
<td>MX</td>
<td>Mail exchanger for the domain</td>
</tr>
<tr>
<td>NS</td>
<td>Nameserver of domain or delegated subdomain</td>
</tr>
</tbody>
</table>
## DNS Resource Records (2)

; Authoritative data for cs.vu.nl

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Class</th>
<th>TTL</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>cs.vu.nl</td>
<td>IN</td>
<td>SOA</td>
<td>86400</td>
<td>star boss (9527,7200,7200,241920,86400)</td>
</tr>
<tr>
<td>cs.vu.nl</td>
<td>IN</td>
<td>MX</td>
<td>86400</td>
<td>1 zephyr</td>
</tr>
<tr>
<td>cs.vu.nl</td>
<td>IN</td>
<td>MX</td>
<td>86400</td>
<td>2 top</td>
</tr>
<tr>
<td>cs.vu.nl</td>
<td>IN</td>
<td>NS</td>
<td>86400</td>
<td>star</td>
</tr>
<tr>
<td>star</td>
<td>IN</td>
<td>A</td>
<td>86400</td>
<td>130.37.56.205</td>
</tr>
<tr>
<td>zephyr</td>
<td>IN</td>
<td>A</td>
<td>86400</td>
<td>130.37.20.10</td>
</tr>
<tr>
<td>top</td>
<td>IN</td>
<td>A</td>
<td>86400</td>
<td>130.37.20.11</td>
</tr>
<tr>
<td>www</td>
<td>IN</td>
<td>CNAME</td>
<td>86400</td>
<td>star.cs.vu.nl</td>
</tr>
<tr>
<td>ftp</td>
<td>IN</td>
<td>CNAME</td>
<td>86400</td>
<td>zephyr.cs.vu.nl</td>
</tr>
<tr>
<td>fits</td>
<td>IN</td>
<td>A</td>
<td>86400</td>
<td>130.37.16.112</td>
</tr>
<tr>
<td>fits</td>
<td>IN</td>
<td>A</td>
<td>86400</td>
<td>192.31.231.165</td>
</tr>
<tr>
<td>fits</td>
<td>IN</td>
<td>MX</td>
<td>86400</td>
<td>1 fits</td>
</tr>
<tr>
<td>fits</td>
<td>IN</td>
<td>MX</td>
<td>86400</td>
<td>2 zephyr</td>
</tr>
<tr>
<td>fits</td>
<td>IN</td>
<td>MX</td>
<td>86400</td>
<td>3 top</td>
</tr>
<tr>
<td>rowboat</td>
<td>IN</td>
<td>A</td>
<td>86400</td>
<td>130.37.56.201</td>
</tr>
<tr>
<td></td>
<td>IN</td>
<td>MX</td>
<td></td>
<td>1 rowboat</td>
</tr>
<tr>
<td></td>
<td>IN</td>
<td>MX</td>
<td></td>
<td>2 zephyr</td>
</tr>
<tr>
<td>little-sister</td>
<td>IN</td>
<td>A</td>
<td></td>
<td>130.37.62.23</td>
</tr>
<tr>
<td>laserjet</td>
<td>IN</td>
<td>A</td>
<td></td>
<td>192.31.231.216</td>
</tr>
</tbody>
</table>
$ dig @june.cs.washington.edu attu.cs.washington.edu ANY

;; Got answer:
;; ->>>HEADER<<- opcode: QUERY, status: NOERROR, id: 34244
;; flags: qr aa rd; QUERY: 1, ANSWER: 4, AUTHORITY: 5, ADDITIONAL: 5
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;attu.cs.washington.edu. IN ANY

;; ANSWER SECTION:
attu.cs.washington.edu. 60 IN A 128.208.1.139
attu.cs.washington.edu. 60 IN A 128.208.1.138
attu.cs.washington.edu. 60 IN A 128.208.1.137
attu.cs.washington.edu. 60 IN A 128.208.1.140

;; AUTHORITY SECTION:
cs.washington.edu. 86400 IN NS holly.s.uw.edu.

;; ADDITIONAL SECTION:
june.cs.washington.edu. 1 IN AAAA 2607:4000:200:17::104
lumpy.cs.washington.edu. 86400 IN AAAA 2607:4000:200:17::102
june.cs.washington.edu. 86400 IN A 128.95.1.4
lumpy.cs.washington.edu. 86400 IN A 128.95.1.2
DNS Resolution

- DNS protocol lets a host resolve any host name (domain) to IP address

- If unknown, can start with the root nameserver and work down zones
  - But can use cache to do as much resolution as possible

- Let’s see an example first ...
DNS Resolution (2)

- flits.cs.vu.nl resolves robot.cs.washington.edu
Iterative vs. Recursive Queries

• Recursive query
  – Nameserver resolves and returns final answer
  – E.g., flits $\rightarrow$ local nameserver

• Iterative (Authoritative) query
  – Nameserver returns answer or who to contact for answer
  – E.g., local nameserver $\rightarrow$ all others
Iterative vs. Recursive Queries (2)

• Recursive query
  – Lets server offload client burden (simple resolver) for manageability
  – Lets server cache over a pool of clients for better performance

• Iterative query
  – Lets server “file and forget”
  – Easy to build high load servers
DNS Resolution (2)

- flits.cs.vu.nl resolves robot.cs.washington.edu
Caching

• Resolution latency should be low  
  – Adds delay to web browsing

• Cache query/responses to answer future queries immediately  
  – Including partial (iterative) answers  
  – Responses carry a TTL for caching
Caching (2)

• flits.cs.vu.nl now resolves eng.washington.edu
  – And previous resolutions cut out most of the process
Local Nameservers

• Local nameservers often run by IT (enterprise, ISP)
  – But may be your host or AP
  – Or alternatives e.g., Google public DNS

• Clients need to be able to contact local nameservers
  – Typically configured via DHCP
Root Nameservers

• Root (dot) is served by 13 server names
  – a.root-servers.net to m.root-servers.net
  – All nameservers need root IP addresses
  – Handled via configuration file (named.ca)

• There are >250 distributed server instances
  – Highly reachable, reliable service
  – Most servers are reached by IP anycast (Multiple locations advertise same IP! Routes take client to the closest one.)
  – Servers are IPv4 and IPv6 reachable
Root Server Deployment

DNS Query Format

<table>
<thead>
<tr>
<th>Transaction ID</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question count</td>
<td>Answer RR count</td>
</tr>
<tr>
<td>Authority RR count</td>
<td>Additional RR count</td>
</tr>
</tbody>
</table>

- **Question entries** (variable length)
- **Answer RRs** (variable length)
- **Authority RRs** (variable length)
- **Additional RRs** (variable length)

<table>
<thead>
<tr>
<th>Request/Response</th>
<th>Operation Code</th>
<th>= 1 bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authoritative Answer</td>
<td>Truncation</td>
<td></td>
</tr>
<tr>
<td>Recursion Desired</td>
<td>Recursion Available</td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td>0 0 0</td>
<td></td>
</tr>
<tr>
<td>Return Code</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RR name</th>
<th>(variable length)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record type</td>
<td>16 bits</td>
</tr>
<tr>
<td>Record class</td>
<td>16 bits</td>
</tr>
<tr>
<td>TTL RR</td>
<td>32-bits</td>
</tr>
<tr>
<td>Resource data length</td>
<td>16 bits</td>
</tr>
<tr>
<td>Resource data</td>
<td>variable length</td>
</tr>
</tbody>
</table>
DNS Protocol

• Query and response messages
  – Built on UDP messages, port 53
  – UDP is unreliable
    • time out and repeat request
    • server is stateless (and requests are idempotent)
  – Query/response linked by a 16-bit ID field
DNS Protocol (2)

• Service reliability via replicas
  – Run multiple nameservers for domain
  – Return the list; clients use one answer
  – Helps distribute load too
DNS Protocol (3)

• Security is a major issue
  – Compromise redirects to wrong site!
  – Not part of initial protocols ..

• DNSSEC (DNS Security Extensions)
  – Mostly deployed

Um, security??
Goal and Threat Model

• Naming is a crucial Internet service
  – Binds host name to IP address
  – Wrong binding can be disastrous ...
Goal and Threat Model (2)

• Goal is to secure the DNS so that the returned binding is correct
  – Integrity/authenticity vs confidentiality

• Attacker can tamper with messages on the network
DNS Spoofing

• How can attacker corrupt the DNS?

• Can trick nameserver into caching the wrong binding by using the DNS protocol itself!
  – This is called DNS spoofing
DNS Spoofing (2)

• To spoof, Trudy returns a fake DNS response that appears to be true
  – Fake response contains bad binding
DNS Spoofing (3)

• Lots of questions!
  1. How does Trudy know when the DNS query is sent and what it is for?
  2. How can Trudy supply a fake DNS reply that appears to be real?
  3. What happens when the real DNS reply shows up?

• There are solutions to each issue ...
DNS Spoofing (4)

1. How does Trudy know when the query is sent and what it is for?

• Trudy can make the query herself!
  – Nameserver works for many clients
  – Trudy is just another client
2. How can Trudy supply a fake DNS reply that appears to be real?

• A bit more difficult. DNS checks:
  – Reply is from authoritative nameserver (e.g., .com)
  – Reply ID that matches the request
  – Reply is for outstanding query

• (Nothing about content though ...)
2. How can Trudy supply a fake DNS reply that appears to be real?

- **Techniques:**
  - Put IP of authoritative nameserver as the source IP address
  - ID is 16 bits (64K). Send many guesses! (Or if a counter, sample to predict.)
  - Send reply right after query

- **Good chance of succeeding!**
3. What happens when real DNS reply shows up?

- Not likely a problem
  - There is no outstanding query after fake reply is accepted
  - So real reply will be discarded
DNSSEC (DNS Security Extensions)

• Extends DNS with new record types
  – RRSIG for digital signatures of records
  – DNSKEY for public keys for validation
  – DS for public keys for delegation
  – First version in ‘97, revised by ’05

• Deployment requires software upgrade at both client and server
  – Root servers upgraded in 2010
  – Followed by uptick in deployment
$ dig edu ANY

; Got answer:
; >>HEADER<<
;; Got answer:
;; flags: qr ra ad
;; QUERY SECTION:
;enu.

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:;
;; QUESTION SECTION:
edu.

;; ANSWER SECTION:
edu.

;; AUTHORITY SECTION:
edu.

;; ADDITIONAL SECTION:

a.eduervers.net. 92995 IN NS l.eduervers.net.
c.eduervers.net. 92995 IN NS d.eduervers.net.
d.eduervers.net. 92995 IN NS f.eduervers.net.

.g.eduervers.net. 92995 IN NS g.eduervers.net.
l.eduervers.net. 92995 IN NS c.eduervers.net.
l.eduervers.net. 92995 IN NS f.eduervers.net.
l.eduervers.net. 92995 IN NS d.eduervers.net.
l.eduervers.net. 92995 IN NS c.eduervers.net.

c.eduervers.net. 92995 IN NS a.eduervers.net.

;; ADDITIONAL SECTION:
an.eduervers.net. 92995 IN A 192.5.6.30
c.eduervers.net. 92995 IN A 192.26.92.30
d.eduervers.net. 92995 IN A 192.31.80.30
d.eduervers.net. 92995 IN A 192.35.51.30
d.eduervers.net. 92995 IN A 192.42.93.30
g.eduervers.net. 92995 IN A 192.42.93.30
g.eduervers.net. 92995 IN AAAA 2001:503:cc2c::2:36

Sb3xPGSU2Wux7nc9BEF04o5wTr1hnt+dE1Hv28/iwOD5YwCjEGVpZonU |l|j[|j/yOlj|j|6RXZMDX8vrmv04GVFf6ndeeQQVq7Q/qlq77MTxX17 r/WV9LDENEG57AzeE/JawZCruYAwvchYB9w86rwb68BM89Nrgn6P5QG
hhf3IlEvBaiPaeBceteADTX0A7nwmw5NWDEeGZGzGheGfD8Bj713n5j FrhWCTICABrNgupY30BcsdC1BSQx51VGhGrC3yinp/Kif2zAf2/T VKx2vmd6D+VQyhtd/sbpt45OmQdpCmXEPkg2f8vufUg4LMWYp0G66sG 12H6Pg==