National University of Computer & Emerging Sciences

CS 3001 - COMPUTER NETWORKS

Lecture 08

Chapter 2

14th September, 2023

Nauman Moazzam Hayat

nauman.moazzam@lhr.nu.edu.pk

Office Hours: 02:30 pm till 06:00 pm (Every Tuesday & Thursday)

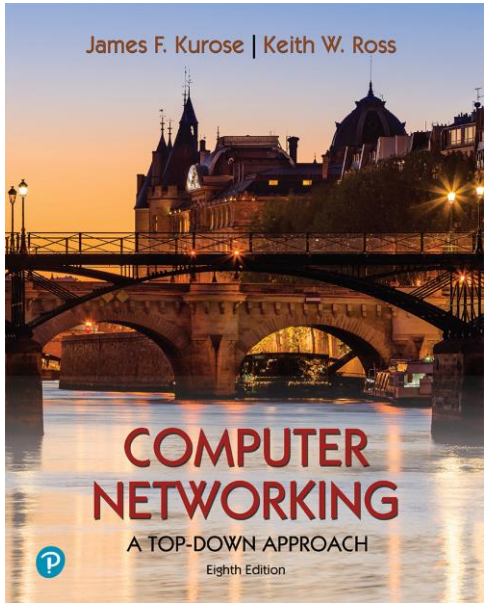
Chapter 2
Application Layer

A note on the use of these PowerPoint slides:
We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)
- If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

For a revision history, see the slide note for this page.

Thanks and enjoy! JFK/KWR

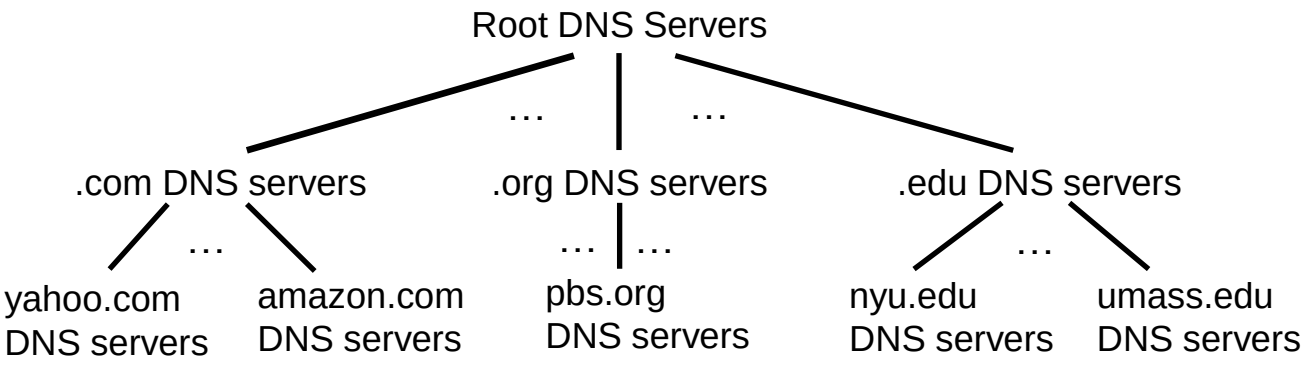


Computer Networking: A Top-Down Approach
8th edition n
Jim Kurose, Keith Ross
Pearson, 2020

Application Layer: Overview

- Principles of network applications
- Web and HTTP
- E-mail, SMTP, IMAP
- **The Domain Name System DNS**
- P2P applications
- video streaming and content distribution networks
- socket programming with UDP and TCP

DNS: a distributed, hierarchical database

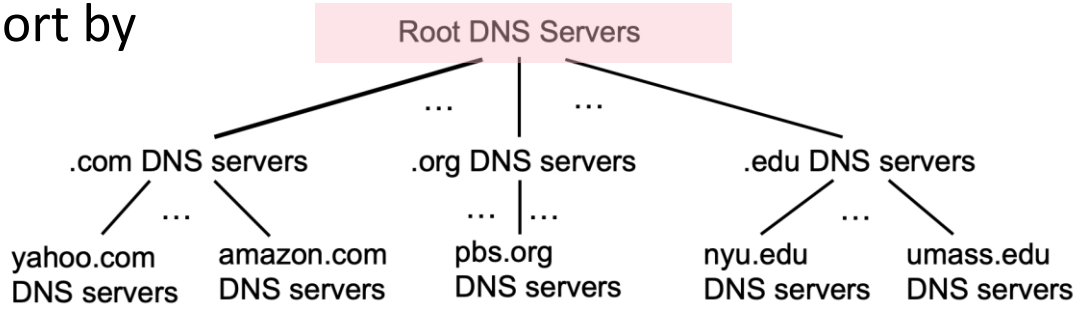


Client wants IP address for www.amazon.com; 1st approximation:

- client queries root server to find .com DNS server
- client queries .com DNS server to get amazon.com DNS server
- client queries amazon.com DNS server to get IP address for www.amazon.com

DNS: root name servers

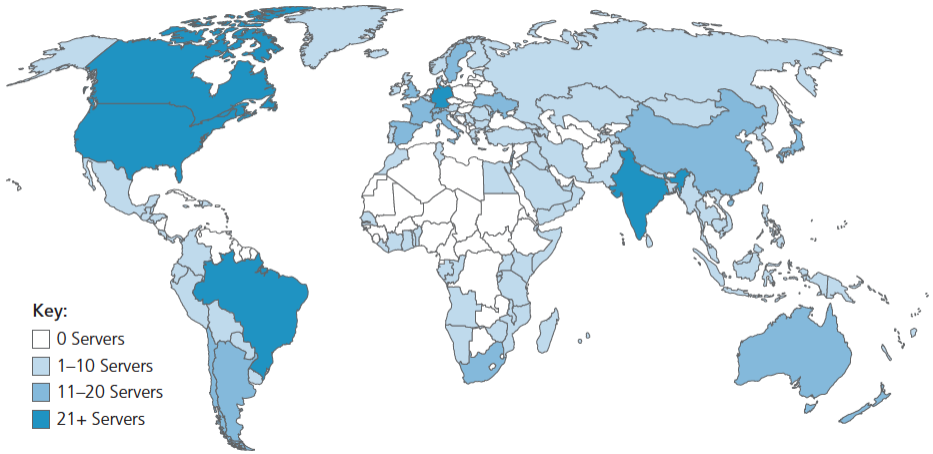
- official, contact-of-last-resort by name servers that can not resolve name



DNS: root name servers

- official, contact-of-last-resort by name servers that can not resolve name
- *incredibly important* Internet function
  - Internet couldn't function without it!
  - DNSSEC – provides security (authentication, message integrity)
- ICANN (Internet Corporation for Assigned Names and Numbers) manages root DNS domain

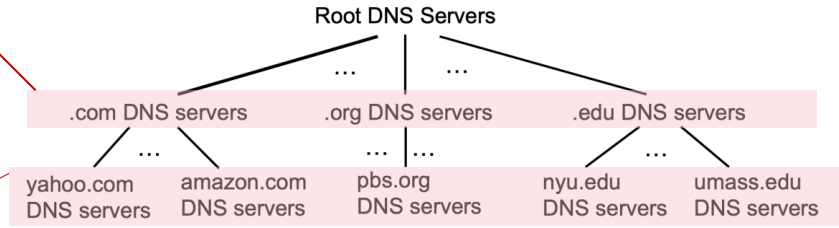
13 logical root name "servers" worldwide each "server" replicated many times (~200 servers in US)



Top-Level Domain, and authoritative servers

Top-Level Domain (TLD) servers:

- responsible for .com, .org, .net, .edu, .aero, .jobs, .museums, and all top-level country domains, e.g.: .cn, .uk, .fr, .ca, .jp, .pk etc.
- Network Solutions: authoritative registry for .com, .net TLD
- Educause: .edu TLD



authoritative DNS servers:

- organization's own DNS server(s), providing authoritative hostname to IP mappings for organization's named hosts
- can be maintained by organization or service provider

Local DNS name servers (Default Name Server)

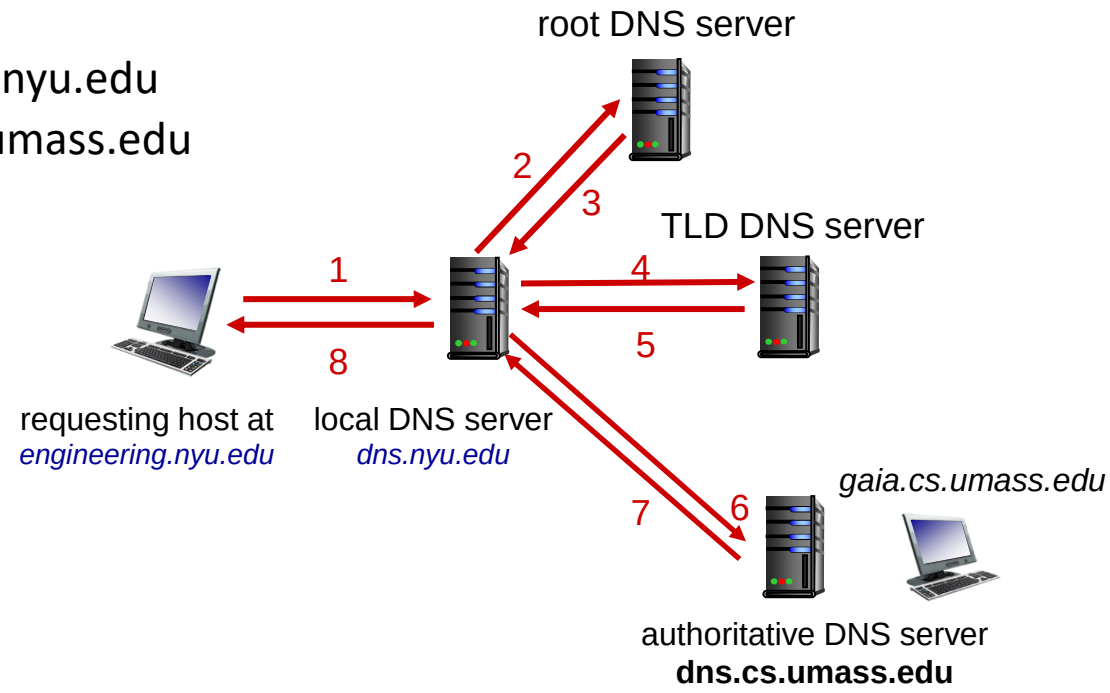
- when host makes DNS query, it is sent to its *local* DNS server
  - Local DNS server returns reply, answering:
    - from its local cache of recent name-to-address translation pairs (possibly out of date!)
    - forwarding request into DNS hierarchy for resolution
  - each ISP has local DNS name server; to find yours:
    - MacOS: `% scutil --dns`
    - Windows: `>ipconfig /all`
- local DNS server doesn't strictly belong to hierarchy

DNS name resolution: iterated query

Example: host at engineering.nyu.edu wants IP address for gaia.cs.umass.edu

Iterated query:

- contacted server replies with name of server to contact
- "I don't know this name, but ask this server"

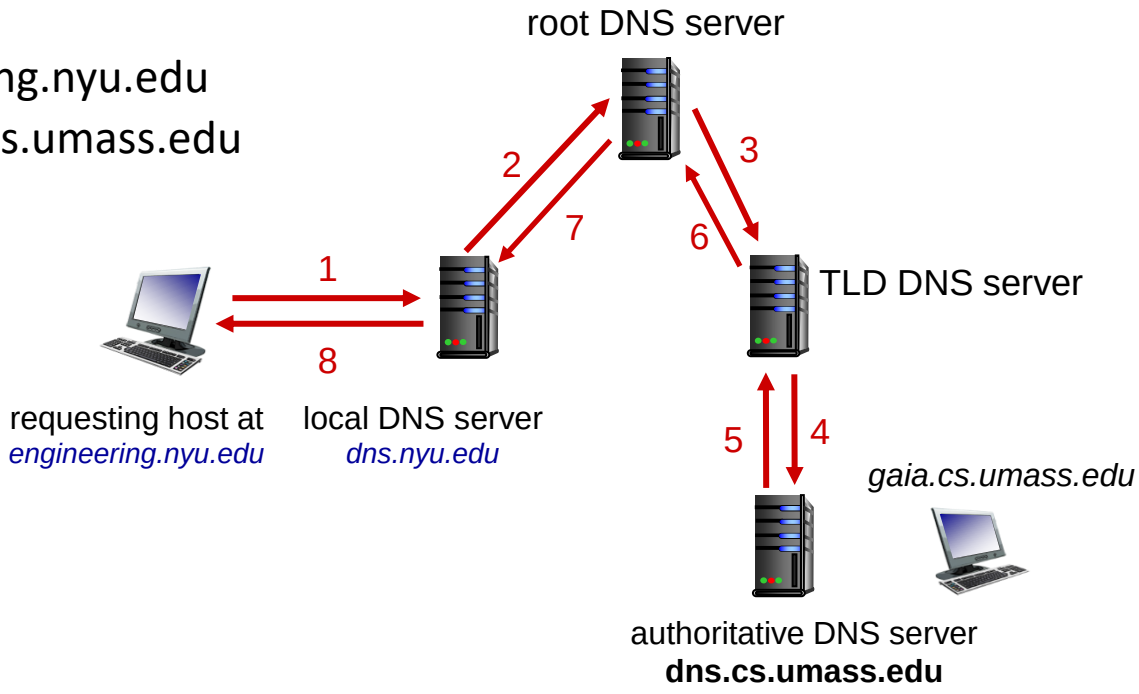


DNS name resolution: recursive query

Example: host at engineering.nyu.edu wants IP address for gaia.cs.umass.edu

Recursive query:

- puts burden of name resolution on contacted name server
- heavy load at upper levels of hierarchy?



Caching DNS Information

- once (any) name server learns mapping, it *caches* mapping, and i*mmediately* returns a cached mapping in response to a query
  - caching improves response time
  - cache entries timeout (disappear) after some time (TTL)
  - TLD servers typically cached in local name servers
- cached entries may be *out-of-date*
  - if named host changes IP address, may not be known Internet-wide until all TTLs expire!
  - *best-effort name-to-address translation!*

DNS records

DNS: distributed database storing resource records (RR)

RR format: (`name, value, type, ttl`)

type=A

- `name` is hostname
- `value` is IP address

type=NS

- `name` is domain (e.g., foo.com)
- `value` is hostname of authoritative name server for this domain

type=CNAME

- `name` is alias name for some "canonical" (the real) name
- www.ibm.com is really servereast.backup2.ibm.com
- `value` is canonical name

type=MX

- `value` is name of SMTP mail server associated with `name`

DNS records

DNS: distributed db storing resource records (RR) (RFC 1035)

RR format: `(name, value, type, ttl)`

if type=A (Address Mapping Record, RFC 1035)

- **`name`** is hostname
- **`value`** is IP address
- Used to map (point) a domain name to an IP address
- E.g. (relay1.bar.foo.com, 145.37.93.126,A)

if type=CNAME (Canonical Name Record, RFC 1035)

- **`name`** is alias (mnemonic) name for some "canonical" (the real) name
- **`value`** is canonical name (real/actual name)
- **`www.ibm.com`** is really **`servereast.backup2.ibm.com`**
- Used to map (point) a domain name to another domain name (for example your website is example.com, but you have also registered examples.com, thus examples.com can be redirected towards example.com via this record
- E.g. (foo.com, relay1.bar.foo.com, CNAME)

if type=NS (Name Server Record, RFC 1035)

- **`name`** is domain
- **`value`** is hostname of authoritative name server for this domain
- NS records specifies which DNS server is authoritative for this domain
- E.g. (foo.com, dns.foo.com, NS)

if type=MX (Mail Exchange Record, RFC 1035)

- **`name`** is alias name for some "canonical" (the real) name
- **`value`** is canonical name of mailserver associated with alias **`name`**
- Same as CNAME but for mailserver
- Used by SMTP to locate mail server name for that domain (thus mail server name must also have a Type A record.)
- E.g. (foo.com, mail.bar.foo.com, MX)

DNS RR Summary

RR format: `(name, value, type, ttl)`

TTL specify the time to leave the resource record. It means it determines the Time when resource should be removed from cache.The meaning of Name and Value depends on Type

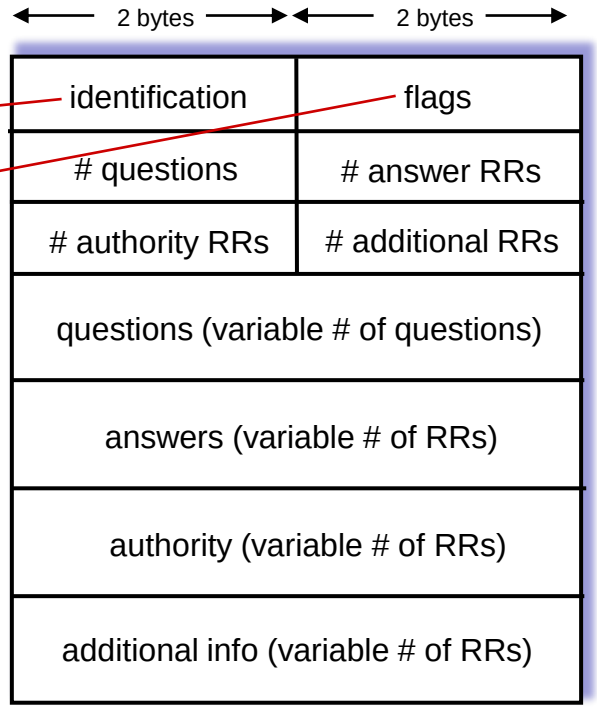
Type	Name	Value	Example	Application
A - Provides hostname to IP translation	Hostname	IP address of Host specified in Name	www.3schools.com , 10.0.0.1, A, 10	Host to IP Translation
NS	Domain Name	Hostname of authoritative server for domain specified in name	Foo.com, dns.foo.com, NS, 10	To get IP of Authoritative Server
CNAME	Alias Hostname	Canonical Hostname	Fb.com, www.facebook.com,CNAME, 10	Host Aliasing
MX	Alias Mail Server Name	Canonical Mail Server Name	Hotmail.com, 123.hotmail.com, MX, 10	Mail server Aliasing

DNS protocol messages

DNS *query* and *reply* messages, both have same *format:*

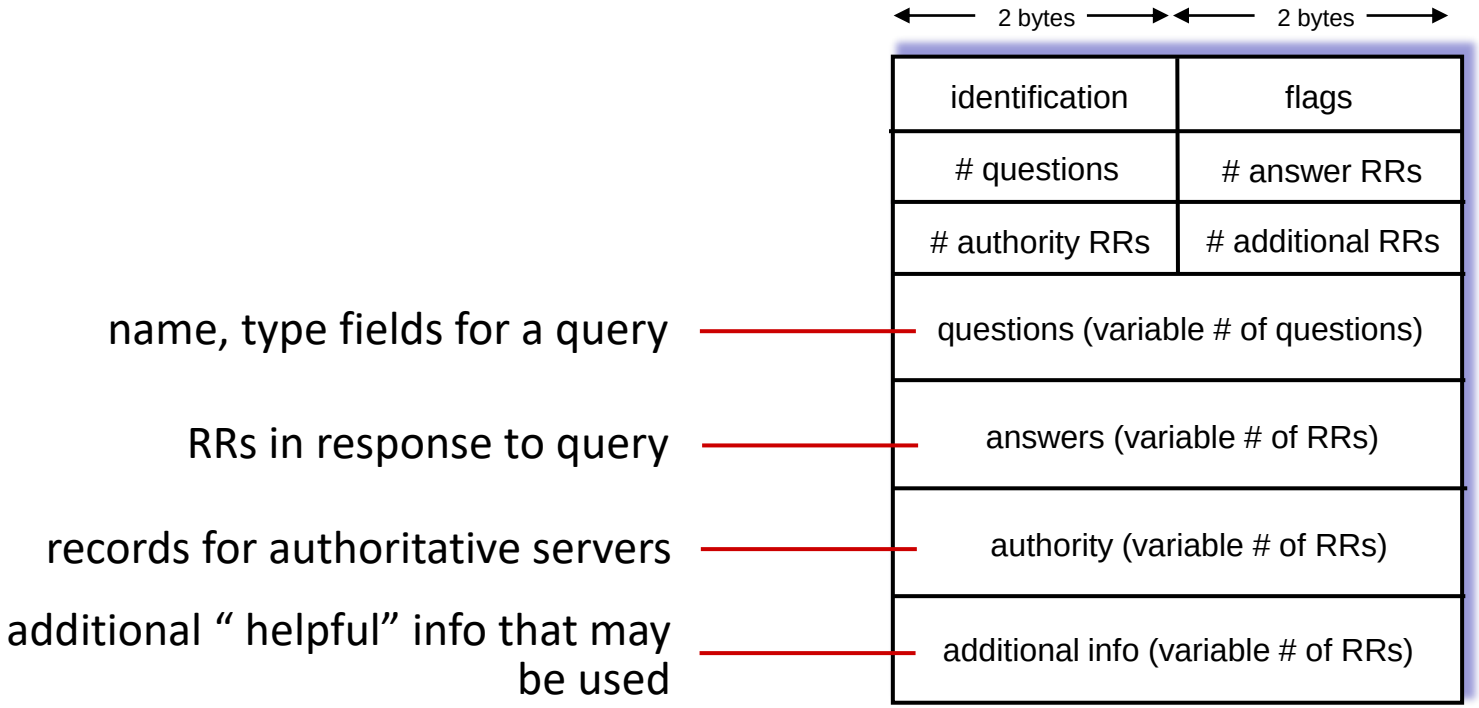
message header:

- identification: 16 bit # for query, reply to query uses same #
- flags:
  - query or reply
  - recursion desired
  - recursion available
  - reply is authoritative



DNS protocol messages

DNS *query* and *reply* messages, both have same *format:*



DNS Messages (Header Section)

- **Header Section:** The first 12 bytes is the header section, which has a number of fields:

  - The first field is a 16-bit number that identifies the query. This identifier is copied into the reply message to a query, allowing the client to match received replies with sent queries.

  - Flags: There are a number of flags in the flag field. A one-bit query/reply flag indicates whether the message is a query (0) or a reply (1). A one-bit authoritative flag is set in a reply message when a DNS server is an authoritative server for a queried name. A one-bit recursion-desired flag is set when a client (host or DNS server) desires that the DNS server perform recursion when it doesn't have the record. A one-bit recursion available flag is set in a reply if the DNS server supports recursion.

- **# Fields (Four number-of fields):** These fields indicate the number of occurrences of the four types of data sections that follow the header.

  - **Question Section:** contains information about the query that is being made. This section includes (i) a name field that contains the name that is being queried, and (ii) a type field that indicates the type of question being asked about the name

  - **Answer Section:** In a reply from a DNS server, the answer section contains the resource records for the name that was originally queried.

  - **Authority Section:** contains records of other authoritative servers.

  - **Additional Section:** contains other helpful records

Getting your info into the DNS

example: new startup "Network Utopia"

- You should register name networkuptopia.com at *DNS registrar* (e.g., Network Solutions, GoDadday etc.)
  - provide names, IP addresses of authoritative name server (primary and secondary)
  - registrar inserts NS, A RRs into .com TLD server: (and CNAME RR if alias name also exists)

    `(networkutopia.com, dns1.networkutopia.com, NS)`

    `(dns1.networkutopia.com, 212.212.212.1, A)`
- You should create authoritative server locally with IP address `212.212.212.1`
  - type A record for www.networkuptopia.com i.e. if DNS query is initiated by HTTP for a webserver, then RRs will be
    - (networkutopia.com, 212.212.212.2, A) assuming webserver IP address is 212.212.212.2
    - and also another RR can be (www.networkutopia.com, 212.212.212.2, A) so that webserver is reached in both the scenarios i.e. if user either types url "networkutopia.com" or types "www.networkutopia.com"
  - type MX record for networkutopia.com i.e. if the DNS query is initiated by SMTP for a mail server, then RRs will be
    - (networkutopia.com, mail.networkutopia.com, MX)
    - (mail.networkutopia.com, 212.212.212.3, A) assuming mail server IP address is 212.212.212.3

DNS security

DDoS attacks

- bombard root servers with traffic
  - not successful to date
  - traffic filtering
  - local DNS servers cache IPs of TLD servers, allowing root server bypass
- bombard TLD servers
  - potentially more dangerous

Spoofing attacks

- intercept DNS queries, returning bogus replies
  - DNS cache poisoning
  - RFC 4033: DNSSEC authentication services

Chapter 2: Summary

our study of network application layer is now complete!

- application architectures
  - client-server
  - P2P
- application service requirements:
  - reliability, bandwidth, delay
- Internet transport service model
  - connection-oriented, reliable: TCP
  - unreliable, datagrams: UDP
- specific protocols:
  - HTTP
  - SMTP, IMAP
  - DNS
  - P2P: BitTorrent
- video streaming, CDNs
- socket programming: TCP, UDP sockets

Chapter 2: Summary

Most importantly: learned about *protocols*!

- typical request/reply message exchange:
  - client requests info or service
  - server responds with data, status code
- message formats:
  - *headers*: fields giving info about data
  - *data:* info(payload) being communicated

important themes:

- centralized vs. decentralized
- stateless vs. stateful
- scalability
- reliable vs. unreliable message transfer
- "complexity at network edge"

Chapter 2

Quiz 1 – Chapter 1