* Definition of a network
  + 2 major components
    - Link
    - Node
* Bursty traffic
  + Handled with packet switching
* Verify data
  + Checksum
  + Send sequence number in the header info
* Statistical Multiplexing
  + Analyzing efficiency of network
  + If you have a 100 Mbps you can give 150Mbps worth of service
* Replicating
  + Don’t hold stuff on 1 server, distribute it to many servers so you have more places to get it from
  + Use closest physical location
* Network boundaries
* Use traceroute to track packets through network
* Bandwith and latency to determine network performance
* Layering
  + OSI - 7
    - Application
    - Presentation
    - Session
    - Transport
    - Network
    - Data Link
    - Physical
      * blutooth
      * Wifi
      * Ethernet
      * Fiber optics
  + TCP/IP - 4
    - Application
      * HTTP, HTTPS, DNS, FTP...
      * Includes from OSI:
        + Application
        + Presentation
        + Session
    - Transport
      * TCP, UDP
    - Network
      * IP
    - Network Access/Data link
      * Ethernet, WiFi
      * Includes from OSI:
        + Data link
        + Physical
  + Encapsulation
    - Wrap application in presentation in session…
* Demultiplexing
* History of internet
  + ARPANET 1969
  + NSFNET 1985
  + Internet 1993
* DNS, HTTP
  + DNS replaces HOST.TXT
  + Namespace
    - .com, .latech
  + Top-level Domain
    - .com, .edu, .org
  + Top-Level Country
    - Mostly used by business
    - .tv, .be
  + Zone
    - Contiguous portion, uses same IP address
  + DNS Resolution
    - Local computer sends to Local DNS (recursive)
    - Local DNS sends to root server (iterative)
      * Root server is top level domains
        + .com, .edu
        + Find zone server for .latech
    - And to zone server
      * Basically this is .latech, responds with coes.latech
      * Then returns it to local DNS
    - Other zone servers if needed
    - When Local DNS gets info, sends it back to computer
* DNS Caching
  + Local DNS store the info of some servers
  + Is able to skip asking root server and zone server
  + DNS based on UDP
* DNSSEC
  + Secure DNS
* HTTP
  + Based on TCP
  + Port 80
  + GET
    - Request the webpage
  + POST
    - Update webpage
  + Performance determined by page load time
  + HTTP 1.0
    - Every time you want to get something, have to establish connection, get info, then release connection
    - Very inefficient
  + HTTP 1.1
    - After establishing, holds connection for a set amount of time (5 min)
    - Persistent connection
    - If re-connect, reset time
    - Close connection after timeout
  + HTTP Caching
    - Temporarily save page content
    - Check expiration date of cached file
    - Ask server for last modification date
* Content Delivery Network
  + Push content to servers at critical locations
  + Use replication
* TCP vs UDP
  + TCP
    - 3 way handshake
    - Flow control
      * Sliding window
        + Go back in - if error, resend everything
        + Selective repeat - if error, only resend those
        + Window size

GBI - max sequence number

SR - half of max sequence number

* + - * Slow start
        + Start with small window
        + Slowly bump up until frequent data loss
  + UDP
    - No error control
    - No connection
    - Lost packet it lost
    - No congestion control
    - Just blast content
    - UDP Header
      * Source port
      * Destination port
      * Length
      * Checksum