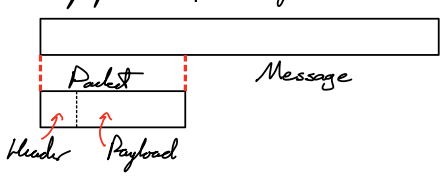
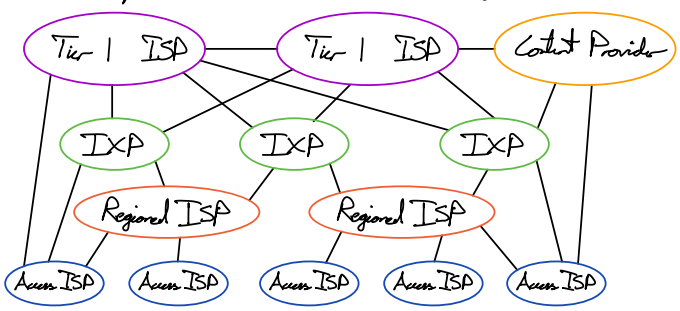
The Internet: a "Nuts & Bolts" view

- a number of connected computing devices
- Protocol: the format and order of messages exchanged as well as the resulting actions
- Network Edge: "end" systems running network applications, i.e. clients & servers
- Access Networks
  - Cable: single transmission line for multiple channels using frequency division multiplexing
    - operates using combination of fiber optic and cable → HFC
  - DSL (Digital Subscriber Line)
    - separate phone line use into internet upload, download, and two-way telephone through frequency encoding
  - Within Home/Enterprise
    - create a local area network (LAN) to connect end system to edge router
    - can employ Ethernet and/or WiFi technologies
  - Wide Area Access Networks: 3G, LTE, 4G, 5G
    - uses cellular technology to operate across tens of kilometers
- Physical Mediums of Transmission
  - Guided
    - Twisted Pair: two insulated copper wires
      - 100 Mbps – 10 Gbps
    - Coaxial Cable: two concentric copper conductors
      - broadband → 100 Mbps+ per channel
    - Fiber Optic: glass fiber carrying pulses of light
      - 100 Gbps+
  - Unguided
    - Wireless Radio: signal radiated through space
      - Terrestrial Microwave: 45 Mbps per channel
      - WiFi: 100 Mbps+
      - Cellular: 10 Mbps+
      - Satellite: 45 Mbps, 270 millisecond delay
- Network Core: interconnected routers; a network of networks
  - Packets: broken up pieces of message with a header



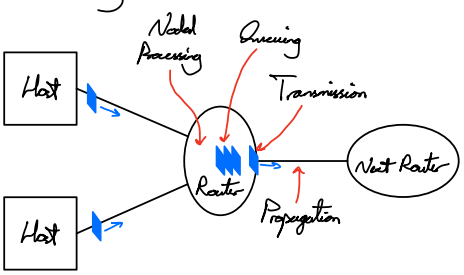
- Packet-Switching: devices that receive and send packets
  - Store and Forward Transmission: packet switch receives entire packet before transmitting
    - Time Delay = $N \cdot \frac{L}{R}$ (N = # of switches; L = length of message (bits); R = rate of transfer (bit/sec))

- Internet Structure
  - multiple ISPs connected through Internet Exchange Points



- large content provides can set up a private network for their own traffic
- high tier ISPs mainly connect low tier ISPs
- IXPs connect ISPs together
- lower tier ISPs mainly connect users

- Packet Delay

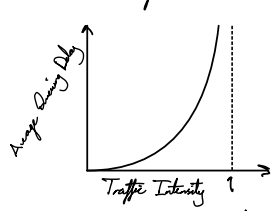


- Processing delay: examine packet's header and determine next destination
- Queuing delay: earlier packets waiting for transmission
- Transmission delay: push all bits into link
- Propagation delay: transfer from beginning of link to next router

$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

($d_{trans} = L/R$; $d_{prop} = \frac{\text{length of link } (d)}{\text{propagation speed } (s)}$)

- Queuing delay depends upon traffic arrival rate, transmission rate, and frequency
  - Traffic Intensity: $\frac{La}{R}$ (a = Average Arrival Rate)
    - metric for estimating extent of queuing delay



- Packet Loss can result from high traffic intensity backing up the queue in a router

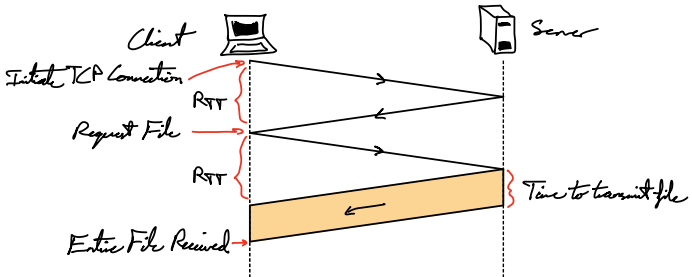
- Throughput: rate at which bits move along link
  - Bottleneck: link with lowest transfer rate
- Network Security
  - Malware
    - Virus: requires medium of transfer (i.e. e-mail) to access host
    - Worm: can spread independently to host
  - Denial of Service: overwhelm host with spam packets
  - Packet Interception: a network interface reads / records all packets
- Internet Protocol Stack: a set of protocols that interact in a certain order
  - Application: supports network applications (i.e. HTTP)
  - Transport: process to process data transfer (i.e. TCP, UDP)
  - Network: routes datagrams from host to host
  - Link: sends data on to physical medium of transfer (i.e. WiFi, Ethernet)
  - Physical: bits travelling through medium
  - Encapsulation
    - each layer adds its own header component to the segment

Application Layer

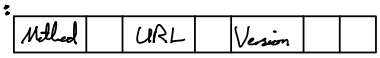
- specifies shared communications protocols and interface methods used by hosts
- Client-Server Architecture
  - Server: always on, permanent IP
  - Client: only communicates with server, dynamically assigned IP
- Peer to Peer Architecture
  - end systems directly communicate
- Processes in different hosts communicate by sending messages through network
  - Socket: where a computer process sends messages into, and receives messages from
    - the interface between application layer and transport layer
    - also known as API
  - Addressing
    - to identify receiving process, sender must know IP address (host) & port number (process)

- Protocols
  - TCP
    - "handshake" setup required
    - reliable & flow/congestion control
  - UDP
    - no setup
    - lightweight & minimal services
- Application Layer Protocol defines:
  - message type, syntax and semantics
  - rules for process message reception & creation

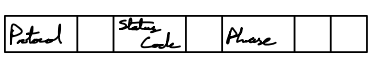
- Web & HTTP
  - webpage consists of objects such as HTML file, JPEG image, audio file, etc.
  - accessible through a URL which contains server hostname and object's path
  - HyperText Transfer Protocol: structure of messages sent between client & server
    - Persistent HTTP: multiple objects sent over single connection
    - Non-Persistent HTTP: connection with only 1 object sent



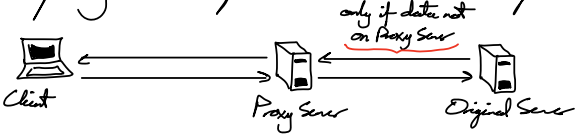
  - Request Message consists of:
    - Request Line
    - Header Lines



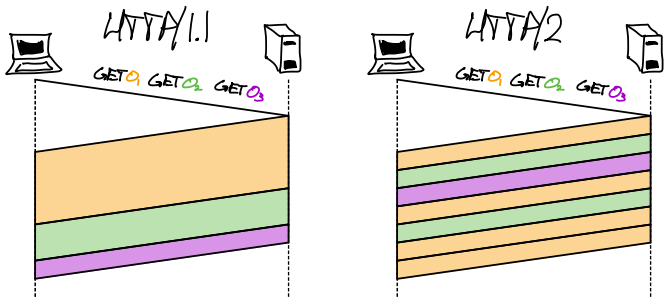
  - Response Message consists of:
    - Status Line
    - Header Lines
    - Data



  - Cookies: data used to maintain a state in transactions in the event of disconnect
    - after user sends HTTP request, server responds with cookie ID; every subsequent request made by user includes cookie header
  - Caches: temporary data copied to close servers for high usage



- Conditional "GET"
  - origin server does not send data in response if there is no new changes
  - sent in "Phrase" part of response message
- HTTP/2: decreased delay in multi object requests
  - objects are divided into frames to interleave transmission to avoid Head Of Line blocking



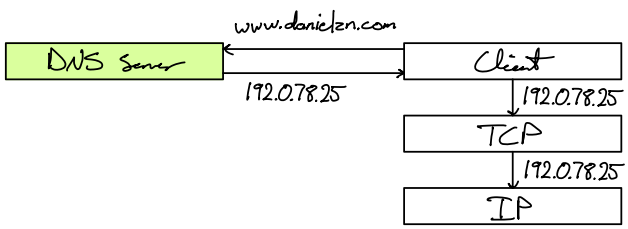
- HTTP/3 adds security, object error & congestion control by using UDP instead of TCP

- Email
  - 3 major components:
  1. User Agents: applications for creating/reading emails
  2. Mail Servers: contains incoming messages, use SMTP to send email between servers
  3. Mail Transfer Protocol (SMTP)

  - SMTP
    - header lines include: "From:", "To:", "Subject:"
    - body contains message; can only contain ASCII characters

  - Mail Access Protocols
    - SMTP: delivery/storage of email messages to server
    - IMAP: retrieval/deletion/folders for stored messages on server
    - HTTP: web-interface for retrieval of email messages

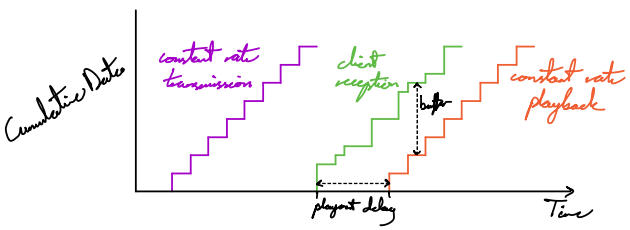
- Domain Name System: decentralized naming system used to identify hosts on the internet
  - DNS Server: translates domain names into IP addresses



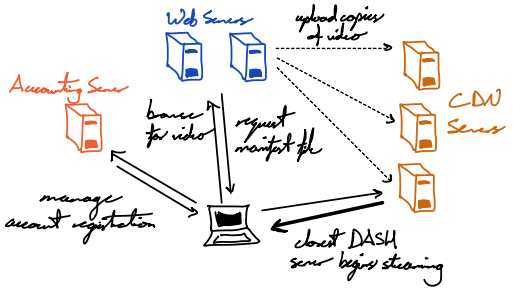
- DNS Services
  - hostname to IP translation
  - long hostname to alias hostname translation (web & mail servers)
  - load distribution: manages single hostname with many IP addresses corresponding to different web servers
  - Hierarchal Database: levels of DNS servers used to narrow search for IP
    1. Root: provides ".com" DNS server
    2. .com: provides "danielzr.com" DNS server
    3. danielzr.com: provides IP address for "www.danielzr.com"
- Name Resolution: servers may forward DNS query to another server, or provide address to client of another server to contact
- DNS servers cache entries to expedite future queries
- DNS Records
  - stored in format: (name, value, type, TTL)
- Insert DNS Records
  - registrar inserts NS, A RRs into root TLD server

- Peer to Peer Applications
  - "peers" intermittently connected with dynamic IP addresses
  - File Distribution Time: time to distribute file (F) to N clients with upload speed u:
    - Client-Server: $D_{c-s} \geq \max\left\{N\frac{F}{u_s}, \frac{F}{d_{min}}\right\}$
    - P2P: $D_{P2P} > \max\left\{\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{NF}{u_s + \sum_{i=1}^{N} u_i}\right\}$
      - multiple peers containing fragments of file can upload individually
  - BitTorrent
    - user requests chunks of file (rarest chunks first), then provides chunks to peers

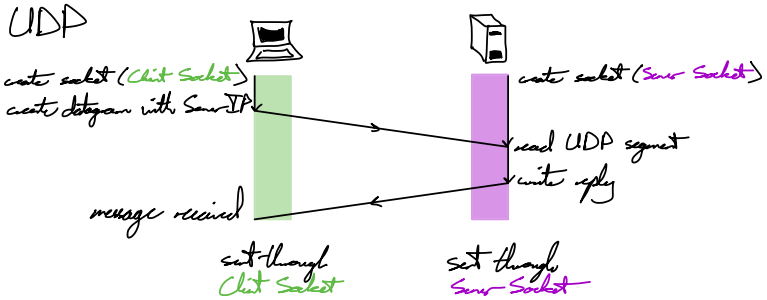
- Video Streaming
  - Streaming Stored Video
    - network delay variations can cause buffering



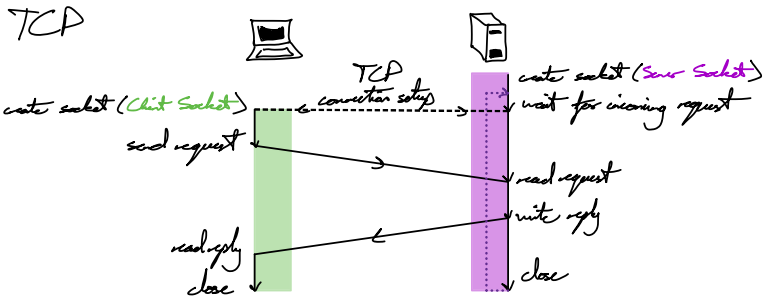
- Dynamic Adaptive Steaming over HTTP
  - video file divided into chunks that are sent at different rates
  - Manifest File: provides URLs for different file rates for client to request
- Content Distribution Network
  - place many servers in access ISPs, or a few at IXPs
  - typical structure uses different servers to perform tasks:



- Socket Programming
  - UDP



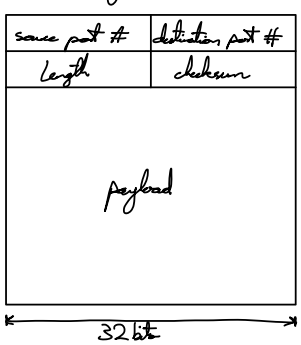
  - TCP



Transport Layer

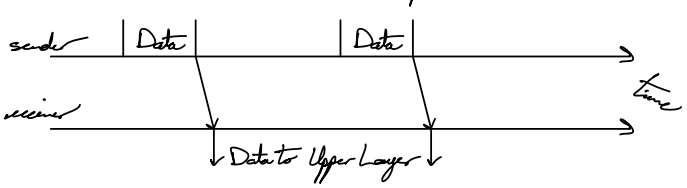
- Logical communications between processes
- Sender: breaks application message into segments that are passed to network layer
- Receiver: receives segment from IP and demultiplexes message to application via socket
- Multiplexing: handle data from multiple sockets and adds transport header
- Demultiplexing: use header info to deliver segment to correct socket
  - TCP requires source & destination IP & port number

- UDP
  - used for streaming, DNS, SNMP, HTTP/3
  - add reliability & congestion control in application layer
  - format:

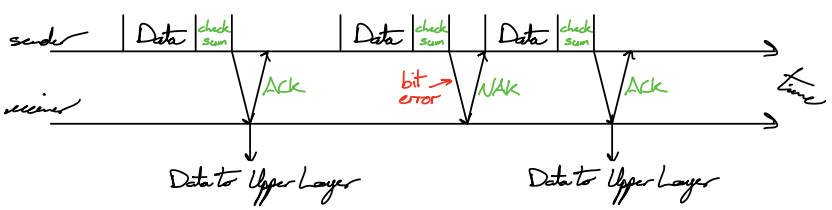


  - "checksum" detect errors (flipped bits) in segment

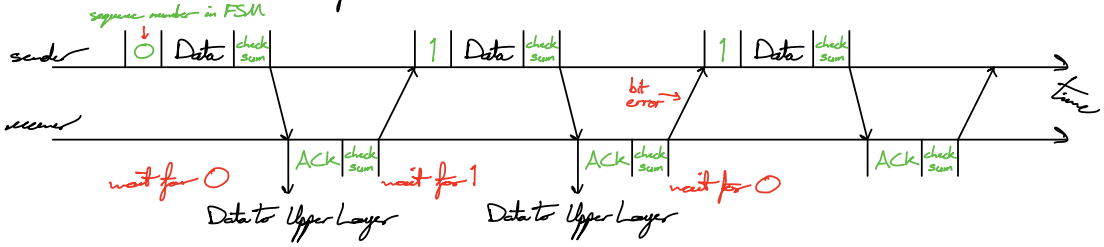
- Reliable Data Transfer
  - sender/receiver use FSMs to handle reception/transmission
  - rdt1.0: no bit errors or loss of packets



  - rdt2.0: channel with bit errors from sender
    - receiver uses acknowledgment (ACK) or negative acknowledgment (NAK) to communicate with sender

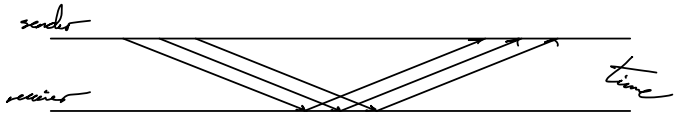


- rdt2.1: channel with bit errors from sender & receiver



- rdt 3.0: channel with errors and losses including packet loss
  - sender retransmits after timeout if no ACK or incorrect ACK is received
  - utilization of bandwidth: $U_{sender} = \frac{L/R}{RTT + L/R}$; inefficient use of bandwidth

- Pipelining
  - increase utilization of bandwidth by sending multiple packets



$$U_{sender} = \frac{3L/R}{RTT + L/R}$$

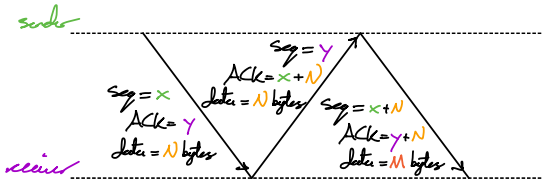
- Go-Back-N: Sender
  - a window of up to N transmitted, unACKed packets
  - Cumulative ACK (ACK(n)): acknowledges all packets <= sequence number n
  - Timeout(n): retransmit all packets >= sequence number n
  - ACK-only: always send ACK for received packet with highest in-order sequence # in the event of a packet loss

- Selective Repeat
  - buffer packets while waiting for lost packet to be retransmitted
  - maximum window size should be half sequence number size to avoid dilemma

- TCP: connection-oriented transport
  - Segment Structure:

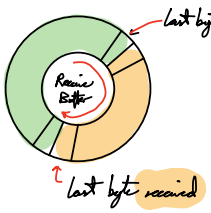
SRC Port #	DST Port #
Sequence #	
ACK #	
	Receive Window
checksum	
TCP options	
data	

- Sequence & Acknowledgment



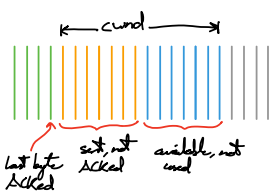
  - ACK field is +N from received seq field
  - after each transmission, the sequence # increases by number of bytes in payload

- Timeout
  - $EstimatedRTT = (1-\alpha) \cdot EstimatedRTT + \alpha \cdot SampleRTT$
  - $TimeoutInterval = EstimatedRTT + 4 \cdot DevRTT$
    - $DevRTT = (1-\beta) \cdot DevRTT + \beta \cdot |SampleRTT - EstimatedRTT|$
- Fast Retransmit
  - if 3 ACKs of same value are received by sender, retransmit the corresponding segment as it was probably lost
- Flow Control — Buffering

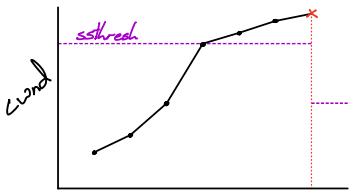


  - free buffer space = Buffer − (last received − last read)
  - sender will send 1 byte of data to check if buffer (rcvwd) has room — if sender receives rcvwd > 0, then it can send

- Congestion Control
  - End-End: congestion is inferred from observed loss & delay
  - Network-Assisted: routers provide direct feedback to hosts
  - increase sending rate when ACK received, and decrease when there is loss
  - AIMD
    - Additive Increase: increase rate by 1 segment for each RTT
    - Multiplicative Decrease: cut sending rate by half in event of loss
  - Congestion Window (cwnd)
    - $TCP\ Rate = \frac{cwnd}{RTT}$ bytes/sec
    - procedure is: send "cwnd" bytes, wait RTT seconds for ACK, send next bytes

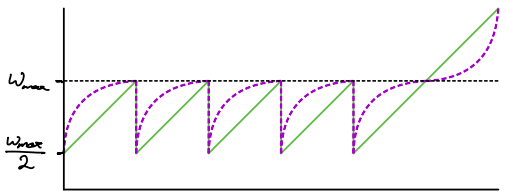


- Slow Start
  - initial cwnd = 1 MSS, then double cwnd every RTT until loss occurs
  - when cwnd surpasses ssthresh, transition from Slow Start to Congestion Avoidance



- Congestion Avoidance
  - $cwnd = cwnd + MSS \cdot \frac{MSS}{cwnd}$; $cwnd > ssthresh$
  - on loss event set "ssthresh" to $\frac{cwnd}{2}$

- CUBIC
  - increase rate by a decreasing amount



TCP Reno (AIMD)
TCP CUBIC: $W = C(t-K)^3 + W_{max}$
(t: time since last loss was detected)

- Bottleneck Link: Location in network where packet loss occurs
- Delay Based Congestion Control: maximize throughput while keeping delay low

$$\text{measured throughput} = \frac{\#\text{bytes sent in last RTT interval}}{RTT_{measured}}$$

- Explicit Congestion Notification: network-assisted congestion control
  - 2 bits in IP header indicate congestion

- Quick UDP Internet Connections: application layer protocol on top of UDP
  - includes loss detection & congestion control from TCP
  - 1 handshake, no HOL blocking

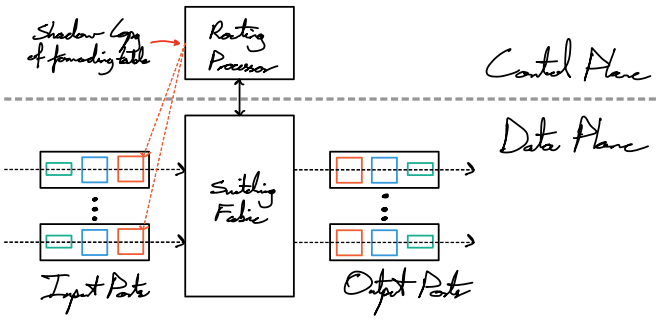
Network Layer

- Sender: encapsulates segments into datagrams, passes to link layer
- Receiver: delivers segments to transport layer

- Forwarding: move packets from input link to output link of router
- Routing: determine route taken by packets from source to destination

- Data Plane: determines how datagram is Forwarded (local logic)
- Control Plane: determines how datagram is routed through network (network logic)

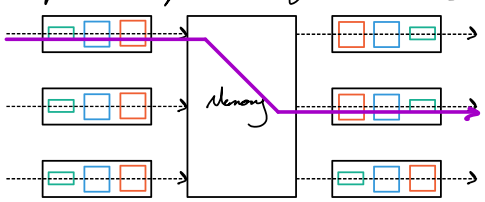
- Router Anatomy



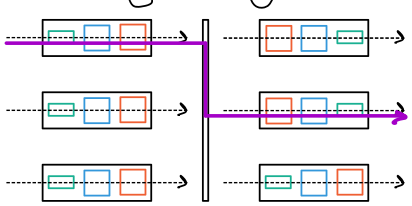
- Decentralized Switching: find output port using forwarding table
- Link Layer Protocol
- Line Termination: data reception
- Forwarding Table: maps destination addresses to router's outbound links

- Switching Fabrics: transfer packet from input link to output link

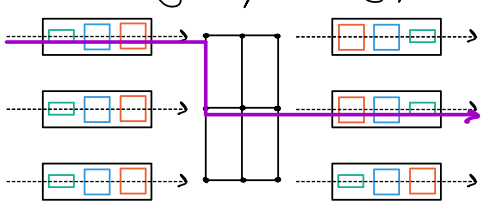
1. Memory Switch
   - packet copied to system memory, each datagram must pass through bus twice



2. Bus Switch
   - switching limited by bus bandwidth

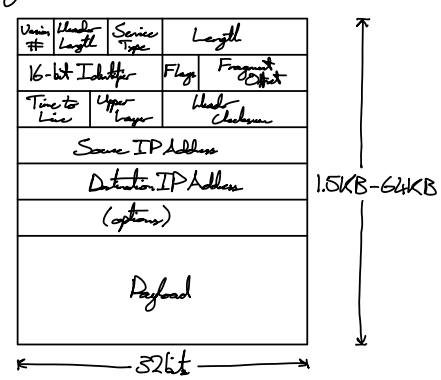


3. Interconnect Network Switch: connect grid of smaller switches
   - scale using multiple switching "planes" in parallel to increase throughput



- Input Port Queuing
  - occurs when switching fabric is slower than input ports
- Output Port Queuing
  - occurs when datagrams arrive from fabric faster than link transmission rate
  - Buffering = $\frac{RTT \cdot C}{\sqrt{N}}$ ← C: link capacity, N: # flows
  - Packet Scheduling: choosing amongst queued datagrams for transmission
    1. First Come First Serve: packets transmitted in order of arrival to output port
    2. Priority Scheduling: incoming packets are queued by class and then sent in order of priority
    3. Round Robin: incoming packets are queued by class and then sent cyclically, one class after another
    4. Weighted Fair Queuing: similar to RR, each class' share of cycle is dependent on a weight

- Internet Protocol
  - Datagram Format



  - IPv4 Fragmentation
    - Maximum Transmission Unit: size of largest data unit that can be communicated in a single network layer transaction
    - Path MTU Discovery: determine smallest MTU on path to avoid fragmentation

  - IP Addressing
    - 32-bit identifier associated with each host interface ← connection between host & physical link
    - Classless Inter-Domain Routing:

      a.b.c.d/x ← subnet mask → 32-bit address

    - Subnet
      - network of hosts that can communicate without routers
      - Subnet Mask: # bits of subnet portion of address
        - remaining bits are assigned to hosts in subnet

- Longest Prefix Matching: choose the forwarding table entry with longest address prefix that matches destination address
- Dynamic Host Configuration Protocol: host is given IP address when it joins network

- Network Address Translation
  - all devices in local network share 1 IPv4 address for main network
  - secondary, private IP addresses are assigned to hosts by router
  - NAT Router Implementation:
    - Outgoing Datagrams: replace source IP address & port # with NAT IP & new port number
    - Translation Table: remember every pair of source & replaced address/port #
    - Incoming Datagrams: replace destination field with source IP address & port number
  - violates end to end argument → port # manipulation by network-layer device

- IPv4 to IPv6
  - Tunneling: IPv6 datagram carried as payload in IPv4 datagram among IPv6 routers

- Generalized Forwarding
  - many header fields can determine multiple possible actions (drop/copy/modify packet)
  - flow tables allow network wide behaviour

- Middlebox
  - modifies network traffic for purposes other than packet forwarding
  - violates "end to end" principle

Network Layer: Control Plane

- Routing Protocols
  - Path: sequence of routers that packets traverse from source host
  - Link State Algorithms: all routers have complete network topology and generate best paths to other routers
  - Distance Vector Algorithms: iterative process to determine path using information from nearby routers

- Dijkstra's Link-State Routing Algorithm
  - computes least cost paths from one node to all others

1. $N' = \{u\}$
2. for all nodes v
   if v adjacent to u
   then $D(v) = c_{u,v}$
   else $D(v) = \infty$
3. loop for all nodes in N'
   find w not in N' such that $D(w)$ is a minimum
   add w to N'
   update $D(v)$ for all adjacent to w and not in N':
   $D(v) = \min\{D(v), D(w) + c_{w,v}\}$

N': set of nodes whose least cost path is known
$c_{x,y}$: direct link cost from node x to y
$D(v)$: current estimate of cost from source to destination "v"
$p(v)$: predecessor node along path from source to destination "v"

- Bellman-Ford Distance Vector Algorithm:

$$D_x(y) = \min_v \{c_{x,v} + D_v(y)\}$$

$D_x(y)$: cost of least cost path from x to y

  - periodically receive new DV estimate from neighbor (y_n) → use Bellman equation to update D_x

- Intra-AS Routing
  - routing among AS's (Autonomous Systems; domains)
  - Routing Information Protocol
    - classic DV: DVs exchanged every 30 seconds
  - Enhanced Interior Gateway Routing Protocol
    - DV based
  - Open Shortest Path First
    - LS based
    - Hierarchical OSPF
      - Two-Level Hierarchy: local area & backbone
      - routers can access their local area & area border routers

- Inter-AS Routing
  - Border Gateway Protocol
    - eBGP: obtain subnet accessability information from nearby ASes
    - iBGP: propagate accessability information to all AS-internal routers

- Software Defined Networking
  - decoupling of network control logic from devices performing functions
  - a remote controller computes & installs forwarding tables in routers
  - easier network management
  - centralized routing table computations
  - Data-Plane Switches
    - implements generalized data-plane forwarding in hardware
  - Components of SDN Controller

interface layer to network control apps
network-wide state management
communication with controlled devices → OpenFlow

- Internet Control Message Protocol
  - communicate network-level information; ie. error reporting, echo request

- Network Management
  - Command Line Interface: operator issues types/scripts direct to individual devices
  - SNMP/MIB: operator queries/sets device data
  - NETCONF/YANG: multi-device configuration management
    - ↳ data modeling language
    - ↳ actively manage devices