Processing Delay: Time to examine packet's header, determine where to direct the packet, check for bit-level errors in the packet, etc.

Queuing Delay: Packet waits to be transmitted onto link. Depends on number of earlier packets. Queuing delay is 0 if queue empty. be zero. Queuing delay long if traffic heavy.

Transmission Delay: Length of the packet = L bits, transmission rate of the link from router A to router B = R bits/sec, Transmission delay is $\frac{L}{R}$.

Propagation Delay: Time for bit to propagate from start of link to other router. Depends on medium of link.

Traffic Intensity: Average rate packets arrive at queue = α packets/sec. Transmission rate (bits pushed out of queue) is R bits/sec. All packets consist of L bits. Average rate bits arrive at queue is $L\alpha$ bits/sec. Assume queue can hold infinite bits, $L\alpha/R$ is traffic intensity.

End-to-End Delay:
$d_{\text{end-end}} = N(d_{\text{proc}} + d_{\text{trans}} + d_{\text{prop}})$, $d_{\text{trans}} = L/R$ where L is packet size.

Internet Protocol Stack: **Application Layer**: Support data exchange. FTP, SMTP, HTTP. **Transport Layer**: Handle delivery reliability, multiplex within host. TCP, UDP. **Network Layer**: Forward packets from source to destination. IP, routing protocols. **Link Layer**: Transfer data between directly connected network elements. Ethernet protocol. **Physical Layer**: Bits "on the wire".

Estimated Round Trip Time:
EstimatedRTT =
$(1-\alpha)\cdot\text{EstimatedRTT} + \text{SampleRTT}$
DevRTT =
$(1-\beta)\cdot\text{DevRTT} + |\text{SampleRTT} - \text{EstimatedRTT}|$
TimeoutInterval = EstimatedRTT + 4 · DevRTT
Typically $\alpha = \frac{1}{8}, \beta = \frac{1}{4}$

TCP Congestion Control: Add "Congestion Control Window" `cwnd`: `LastByteSent - LastByteAdded` $\leq$ `cwnd`.

"Slow" Start (`cwnd` < `ssh-thresh`): (1) `cwnd` = 1 MSS (2) Send packets (3) If packets got ACKed, `cwnd` = 2· `cwnd` (3) goto 2 (4) If not ACKed, `ss-thresh` = `cwnd / 2`, `cwnd` = 1 MSS, goto 2.

Congestion Avoidance (`cwnd` $\geq$ `ssh-thresh`): (1) If packets got ACKed, `cwnd` = `cwnd` + 1 MSS (2) Otherwise `cnwd` = `cnwd / 2`.

Fast Recovery / Retransmit (Reno): If loss detected through 3 dup ACKs, reduce `cwnd` by half and quickly recover from loss.

IPv4 a.b.c.d/x (CIDR): **#bits**: x, **#endpoint addr**: 2^{32-x}, **net addr**: a.b.c.d, **net mask**: leading x 1s the rest 0s, **broadcast addr**: (addr | (∼mask))

IPv4 Datagram: **Version (4)**: IP protocol version. **Header Length (4)**: Length of header (32 bit words). **Type of Service (8)**: Low delay, high throughput, reliability, etc. **Datagram Length (16)**: Header + data, in bytes. **Identifier, Flags, Fragmentation Offset (32)** : For IP fragmentation. **Time-to-Live (8)**: Ensure datagrams don't circulate forever. **Protocol (8)**: Type of transport-layer protocol. 6: TCP, 17: UDP. **Header Checksum (16)**: Computed with IP header. **Source, Destination Address**: IP address of source and destination. **Options**: Allows header to be extended. **Data**: Payload.

IP Fragmentation: MTU (maximum transmission unit) is max data that link-layer can transfer. (1) Stamps with identification (same for all fragments) (2) Set flag = 1 (last fragment flag = 0) (3) Offset field specifies where fragment fits in original IP datagram.

Routing Table: Match prefix to outgoing link. Exists on every host and router. **Longest Prefix Matching**: Longest net addr prefix with net mask. Routing table next-hop.

ICMP (Internet Control Message Protocol):

type	code	message
0	0	echo reply
8	0	echo request
3	0	dest net unreachable
3	3	dest port unreachable
11	0	TTL expired
12	0	IP header bad

Link-State (Dijkstra): Router computes shortest paths to all dest.
`c(x,y)`: cost from `x` to `y`
`D(v)`: current path cost from source to `v`
`p(v)`: node right before `v` on least-cost path
`N'`: set of nodes with least cost paths found

```
Initialization (source = A):
  N' = {A}
  for all nodes v
    if v adjacent to A
      then D(v) = c(A,v)
    else D(v) = infty
LOOP
  find w not in N' st D(w) is min
    add w to N'
    update D(v) for all v adjacent to w
      and not in N'
    D(v) = min(D(v),D(w)+c(w,v)), p(v) = w
UNTIL all nodes in N'
```

Distance-Vector (Bellman-Ford): Router computes shortest paths based on shortest paths of all neighbors.
$D_x(y)$: cost of best path from x to y.
$D_x(y) = \min\{c(x,v) + D_v(y)\}$, with min over all neighbors v of x.
Node x: - knows cost to neighbor v, $c(x,v)$
- maintains distance vector $\boldsymbol{D_x} = \{D_x(y) : y \in N\}$
- sends $\boldsymbol{D_x}$ to all neighbors
- $\boldsymbol{D_v}$ from each neighbor v
- calculates $D'_x(y) = \min\{c(x,v) + D_v(y)\}$
– If $D'_x(y) < D_x(y)$: $D_x(y) = D'_x(y)$, next hop to $y = v$, send out updated $\boldsymbol{D_x}$

Poisoned Reverse: Node z will advertise to y that $D_x(z) = \infty$ if z goes through y to get to x. Thus y will never route through z to get to x.

OSPF (Open Shortest Path First): Each node knows neighbors & link distance to each neighbor. Each node periodically broadcasts link-state to entire network.
Sends LSP (Link-State Packet) with (1) ID of node (2) list of neighbors with link cost to each (3) sequence (SEQ) for LSP (4) time-to-live (TTL) for information in LSP.

BGP (Border Gateway Protocol): BGP routers establish BGP session over semi-permanent TCP connection to exchange routing info (propagate reachable IP prefixes), advertising paths to dest network prefixes ("path vector" protocol) Provides each AS (Autonomous Systems) means to (1) obtain subnet reachability (IP address prefix) from neighbors AS (2) propagate reachability info to all internal routers (3) determine "good" routes to dest prefix based on reachability info (4) advertise its own prefixes to internet.

eBGP: Session between different AS.

iBGP: Session between routers in same AS.

Prefix + Atrributes = "Route", **Atributes**: (1) **AS-PATH**: list of ASes through which prefix has passed (2) **NEXT-HOP**: specific internal-AS router to next-hop AS (3) **Local Preference**: policy preference in path selection, injected into BGP update at border router.

BGP Routing Policy: Provider advertises all prefixes to customer AS. Customer does not advertise prefix between providers.

Sequence-Number-Controlled Flooding: Source node puts its address & broadcase sequence number in broadcast packet. Each node maintains list of source address & sequence number of packets received. If in list, drop packet.

RPF (Reverse Path Forwarding): Router accepts packets that is on shortest unicast path back to source.

Spanning-Tree Broadcast: Each node generate minimum spanning tree. Broadcast packet only on tree links.

IGMP (Internet Group Management Protocol): **Membership Query**: Sent by router to all hosts on attached interface to determine set of all multicast groups that have been joined by hosts on that interface. **Membership Report**: Response to membership query by host. Also sent when application first joins multicast group without waiting for membership query. **Leave Group**: To leave group. Optional bc router infers if host doesn't respond to membership query.

HDLC Byte Stuffing: Frame delimitter `01111110`. Adds extra `0` after sequence of five `1` bits.

PPP Byte Stuffing: Frame delimitter `01111110`. Replaces `0x7e` (`01111110`) with "control sequence" `0x7d 0x5e`. Replaces `0x7d` with `0x7d 0x5d`.

COBS(Consistent Overhead Byte Stuffing): Frame delimitter `0x00`. Adds byte representing # non-zero bytes in stream+1, adds non-zero bytes, adds byte representing # non-zero bytes to follow + 1, ...

Link Access: MAC (Medium Access Control) specifies how frames are transmitted onto link.

Random Access Protocols (RAP)

ALOHA & Slotted ALOHA: Retransmit with probability p. Otherwise wait certain time period (next time slot Slotted ALOHA).

Carrier Sense Multiple Access (CSMA): If channel is idle then transmit.

1-persistent: Retransmit immediately.

p-persistent: Retransmit immediately with probability p.

non-persistent: Retransmit after random interval.

ARP (Address Resolution Protocol): If IP-MAC not in ARP table, send ARP request, wait for ARP response, then cache IP-MAC-TTL in ARP table.

Switches: Build forwarding table by self learning.

(1) If frame source MAC address not in table, record source MAC address, interface, timestamp.

(2) If table entry found for dest: if dest on segment from which frame arrived, drop frame, otherwise forward to outgoing interface.

(3) If table entry not found for dest: forward to all interfaces except source (broadcast frame).

Switch v. Router: Switch is layer 2 (link), forward with MAC address. Router is layer 3 (network).

802.11 Architecture: **Basic Service Set (BSS)**: Contains wireless stations and central **base station**, or **Access Point (AP)**. The APs in each BSS are connected to interconnecition device (switch, router).

Passive Scanning: 802.11 requires AP periodically send **beacon frames**, including AP's SSID (Service Set Identifier) and MAC addr.

Active Scanning: Probe Request frame broadcast from host. Probes Response sent from APs.

Collision Avoidance: Host sends **Request to Send (RTS)** frame to reserve access to channel. AP broadcasts **Clear to Send (CTS)** frame to sender to give permission and tells other senders to not send. Once done, send ACK and notify other senders free to send.

802.11 Frame: **Addr 1**: MAC of receiver. **Addr 2**: MAC of sender. **Addr 3**: MAC of router in BSS. **Addr 4**: Only for ad hoc. **Sequence Number**: Used for sending ACKs.