

Routers/Switches: Devices that forward packets arriving on one link to another link

End Host: A device attached to the network that sends or receives packets.

- Ex: mobile phone, laptop, etc.

Packets: Bags of bits with:

- Header: info for network and network stack to make decisions
- Body: contains a payload. For example, a file, image, application header, etc.
- The network doesn't care about what's in the payload

Naming

Network name: which host it is

Network address: where host is located

When you move a server to a new building, its name doesn't change, but its address does.

Network "Stack": Networking software on host.

Replicates some router/switch functionality and adds some additional functionality before passing the body of packets to the application. This is the networking code inside the OS.

ISP (Internet Service Provider): A network of packet switches and links that provide network access (Comcast, ATT, Sonic)

ASes (Autonomous Systems): Groups of routers under the same control

- Usually each ISP has one AS, but may have multiple ASes.
- Routers within the same AS will have info about each other.

Ping: Simple utility that lets you poke a website and see if it moves. Say hi and see if the server says hi back and how long the reply took to come back.

Latency: Time between when a request is sent and when the response is heard

Traceroute: Tool to trace the route that

packets take from your computer to the destination. Specifically lets you see the routers/switches that are forwarding your packets. Takes advantage of something called a TTL in the packet IP header.

TTL: how many times a packet should be forwarded before it is discarded.

When packets are dropped because of TTL expiring, routers send back a message telling us. You can use the source of this notification to identify the routers along the packet's path

Traceroute gives us:

- Latency to every step of the way
- Router names/locations
- Stars → routers that time out/don't respond

Dig

Internet thinks in terms of addresses not names. (1.2.3.4 not google.com). Dig lets you look up the address of a website by name. It is a CLI tool to query the Domain Name Service (DNS).

dig +trace shows the DNS lookup step by step. It asks a root server, then top level domain servers, then more servers until the IP was found.

Delays

Time for a packet to travel through a network depends on:

- How much data and link speed → transmission delay
- Distance from the destination → propagation delay
- Traffic pattern → Queuing delay

Transmission Delay

How long it takes for all bits in the packet to get on the wire.

- Time between first & last bits entering
- Limited by bandwidth → # of bits you can send through a wire per unit of time
- Transmission Delay = $\frac{\text{Packet Size}}{\text{bandwidth}}$

Propagation Delay (latency)

- End to end transmission time of one bit.

- Depends on the length of the link.

- Limited by the propagation speed (usually speed of light)

- Capacity = Delay × Bandwidth

- Delay = $\frac{\text{length of link}}{\text{speed of light}}$

Pipe Diagrams: Height = bandwidth
Width = prop. delay

Transient Overload: Packets arrive faster than the router can send them. Packets form a queue and the router sends out packets in that order.

Queuing Delay

How long the packet waits to get transmitted, how long it is in the queue.

- Only happens when arrival rate > transmission rate.

Persistent Overload

- Not enough capacity to hold incoming packets (even in queue) → packets might get dropped.

- Solve by telling senders to slow down or upgrade link.

Life of a Packet

Packet Delay = Transmission + Propagation + Queuing Delays

Lifecycle:

- Sender puts payload in packet, adding headers
- Packet travels along a link.
- Packet arrives at a router, router forwards packet
- Might be queued or dropped
- Repeat until packet is dropped or reaches dest.

Round Trip Time (RTT): Time for a packet to reach its destination and receive a response

$RTT = 2 \times \text{Propagation Delay}$

Statistical Multiplexing

Users can share a router without splitting it since traffic tends to be bursty

Layers in Open Systems Interconnected (OSI) Model

- L7: Application header
- L6: Presentation header
- L5: Session
- L4: Transport → beyond delivery, reliability, packet assembly, etc.
- L3: Network → global delivery
- L2: Data Link → local delivery
- L1: Physical → Physical transfer of bits

Sockets - Endpoint for sending/receiving data across network

- Allows L7 Apps to work on data streams (not packets)
- Open a socket between a source and dest IP address.
- Types: Server and Client
 - Servers listen for clients to connect to them
 - Accept and dispatch connection
 - Usually many clients at once
 - Clients initiate new connections to servers.
 - Ex: Server → berkeley.edu
Client → Your browser
- Sockets use unique IP:port pairs!
Ports are associated with a socket.
- Servers listen on well known ports so clients know where to connect. Clients also have their own port so replies come back to the right process

Routing - builds a path backwards (starting from dest)

Forwarding uses that to send packets towards the dest. Router stores only the next hop.

Distance Vector Algorithm

For each destination:

- If you hear advertisement, update table & reset TTL if:
 - dest isn't in table
 - Advertised cost + Cost to neighbor < best known cost
 - Advertisement is from current next hop
 - Includes Poison advertisements.
- Advertise to all your neighbors when table updates and periodically
- But don't advertise back to the next hop
- Or advertise poison back to the next hop
- Any cost ≥ 16 is advertised as ∞
- If a table entry expires, make the entry poison and advertise it.

Link State Routing

- Each router knows its own link state
 - State of each link to its neighbor
 - Associated costs

① Router floods its link state to all other routers

② Each router learns global network topology

③ Computes shortest path themselves (with Dijkstra's, etc)

Distance Vector vs Link State

Distance Vector	Link State
- Global Computation (distributed across all nodes)	- Local Computation
- Only local data (local node plus whatever neighbors told us)	- Using global data (from all parts of the network)

Addressing

Requirements:

- Scalable Routing
 - Minimize state exchange needed to create paths
- Efficient Forwarding
 - Small forwarding tables
 - Fast lookups
- Host must be able to recognize packet is for them
 - An end to end check on routing
 - L3: IP Addresses (dynamically assigned)

IP Address

Example: 1.2.2.0/23

0000 0001 0000 0010 0000 0010 0000 0000

23 bits for network address last 9 bits for host address

To tell if an address is in a network check if the first /n bits match the network address first /n bits. In this case /n is 23.

This is called Classless Inter-Domain Routing (CIDR)

- Easy to add new hosts since forwarding table doesn't need to be updated.

Interdomain Routing: between autonomous systems

- Want scalability and policy compliance

Export & Selection

Each AS makes two policy decisions:

- Selection (outgoing): choosing which route to use to send a packet to a destination.
- Export (incoming): which neighbors you tell about your route, therefore whose traffic you agree to carry.

Types of ASes:

- Stub: only sends/receives traffic for its users
 - companies, universities, etc.
 - Transit: carries traffic for other ASes
 - Global ISPs (tier 1), Regional ISPs (tier 2), Local ISPs (tier 3)
- All ISPs are ASes, but not all ASes are ISPs
 - Two ASes will connect only if they have a business relationship
 - Provider A carries customer B's traffic for a fee
 - Peers A, B carry each other's traffic for free
 - Peer relationships can be cyclic, customer-provider relationships must be acyclic.

Gateway Protocols

eBGP - Between border routers in different ASes

- Learn about external routes

iBGP - Between border routers and other routers within a single AS

- Learn which border router to use to reach external destinations

IGP - protocol used for intradomain routing (OSPF)

- shortest path to internal or external destinations

BGP Route Attributes

Attributes: Parameters used in route selection

① Local Attributes:

- ASes keep them private
- Not included in eBGP route announcements
- Ex: LOCAL_PREF

② Nonlocal Attributes:

- propagated with eBGP route announcements
- Ex: AS_PATH

Route Selection - Rule → Tiebreak → ...

Highest LOCAL_PREF → shortest AS_PATH → lowest IGP cost to next hop → lowest MED → Smallest IP Address of next hop routers

Gao-Rexford Provider $\rightarrow$ Customer

ASes act like businesses:

- When choosing a route, prefer the ones that make you money, then ones that are free, then ones that cost money. Provider chooses to participate.
- When sharing routes, only carry traffic if getting paid. Tell everyone about routes to your customers, but only tell your customers about routes to peers and providers.

TCP - L4 Protocol that turns IP's unreliable packet delivery into a reliable connection.

Properties:

- ① Byte Stream: apps send and receive continuous stream of bytes. TCP handles splitting it into packets.
- ② Bidirectional: Both sides can send at once
- ③ Reliable: Lost packets get retransmitted
- ④ In-order: Bytes are delivered in the order they were sent even if packets arrive out of order.

TCP Header:

- ① Host port numbers - used for (*) multiplexing
- ② Sequence number - byte offset of the packet's first data byte.
- ③ Acknowledgement number - next byte the receiver expects
- ④ Header Length - header size, so the receiver knows where the data starts
- ⑤ Flags - connection control (SYN start, ACK, FIN close, RST abort)
- ⑥ Advertised Window - how much the receiver can accept (flow control)
- ⑦ Checksum - detects corruption
- ⑧ Urgent - marks urgent data (rarely used)
- ⑨ Options - optional extras (assume none)

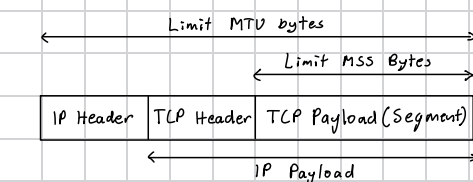
TCP Segments

TCP/IP Packet: IP Packet with TCP header and TCP data inside.

Size limits: $\rightarrow$ Fragmentation if packet size $>$ MTU

- IP packet: Maximum transmission unit (MTU)
- TCP Segment: Maximum segment size (MSS)

$MSS = MTU - (IP\ Header) - (TCP\ Header)$



Three Way Handshake

How TCP sets up a connection. Each side shares its starting state (initial sequence number, receive window, port) and confirms it got the others.

- ① SYN (client $\rightarrow$ server): Want to connect, sequence #s start at M
- ② SYN-ACK (server $\rightarrow$ client): Got it (ACK M+1), sequence #s start at N.
- ③ ACK (client $\rightarrow$ server): Got it (ACK N+1). This packet can carry data.

After ③, both sides are ESTABLISHED and can send freely. The ACKs are +1 because a SYN counts as one byte of sequence space.

Sequence #s start at a random value so stray packets from an old connection on the same ports don't get mistaken for new data, and so attackers can't easily guess them to inject fake packets. Each side picks its own, so they have to tell each other via SYN and SYN-ACK.

Teardowns

TCP Connections end in one of two ways:

- ① Normal Teardown (4 way handshake): each direction is closed separately.
 - a) Client sends FIN to signal done sending
 - b) Server ACKs it. Server can keep sending data.
 - c) When the server is done, it sends its own FIN.
 - d) Client ACKs and connection is closed
- ② Reset (RST): An abrupt teardown when something's wrong, most commonly a SYN sent to a port nobody is listening on. One side sends RST and the connection dies immediately.

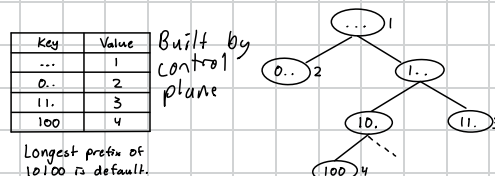
TCP Reliability

TCP uses cumulative ACKs. If a packet goes missing, the ACK number stops advancing, revealing the loss. TCP detects loss in two ways

- ① Timeout (RTO)
 - Sender keeps one timer, based on its estimate of the RTT plus its variance
 - Each ACK for new data resets the timer
 - If the timer expires, the sender retransmits the oldest unACKed packet.
 - Downside is that if the RTO is too long, the sender sits idle for a while before resending.
- ② Duplicate ACKs (faster way)
 - Ex: packet 3000 dropped, but 4000, 5000, 6000 all arrive
 - Each one makes the receiver repeat ACK 3000, since its missing byte 3000.
 - Repeated ACKs for the same number signal a loss or reordering, so the sender can retransmit right away without waiting for the timer.

Longest Prefix Matching

If a dest matches several ranges, pick longest prefix. Ex: 4.29.1.2 matches 4.0.0.0/8 and 4.29.0.0/16. Implemented w/ tries. $O(1)$ Runtime since there are 32 nodes max. Longest prefix to external network is 24.



Router Hardware

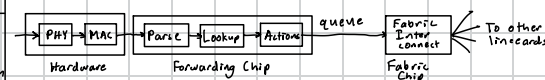
- line rate := Speed of a port
- innovation focused on $\uparrow$ line rate
- Planes
 - Data
 - local operation (forwarding)
 - ns timescale
 - Control
 - preforms routing protocols
 - Second time scale
 - Management
 - How operator interacts w/ router
 - seconds/minutes timescale
 - assign costs to links
 - how much traffic is being sent on each link?

Types of Packets

- User: sent to dest.
 - forwarding chip looks up which port to forward packet
 - if outgoing port is on diff linecard, send there via fabric
- control: packets intended for router
 - ex: ads in routing protocols
 - forwarding chip sends to controller for processing
- punt: packets intended for a user, but requiring extra processing
 - ex: packet TTL has expired, send back error
 - forwarding chip 'punts' to controller for processing

Forwarding Pipelines

- 1) receive packet from other systems
 - PHY (L1) decode into 1s 0s
 - MAC (L2) ops
 - 2) process packet
 - parse packet headers
 - look up next hop in forwarding table
 - update packet
 - dec ttl, update checksum, fragment packet
 - 3) send packet onward
 - fabric chip sends packet to other linecards
- * Forwarding logic built into hardware



Fabric interconnect has limited features $\rightarrow$ v. fast

IPv4 addresses are 32 bits $\rightarrow 2^{32}$ exist
 IPv6 are 128 bits $\rightarrow 2^{128}$ addresses
 $\rightarrow$ uses hex: 2001:0:0:0:0:0:0:1
 2001:0:0:0:1

BGP

- Goals
 - scalability: hierarchical addressing
 - privacy
 - autonomy

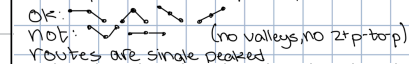
Gao-Rexford Rules

- making money is good
- don't do work for free

If all ASes follow Gao-Rexford

- can guarantee reachability $\rightarrow$ convergence

* Valid paths (AS will participate) if someone is paying it



eBGP vs. iBGP

eBGP: border routers in different ASes exchange routes

iBGP: border routers share external routes with routers inside their own AS. Needs N*B sessions (N routers, B border routers).

To reach an external destination: BGP picks the egress border router, then the IGP gets you there.

LOCAL-PREF

- Set by border routers when they send iBGP messages. It's never sent over eBGP, so it is private to the AS.

- Encodes Gao-Rexford: customer route $>$ peer route $>$ provider route

Store and Forward and Queuing

x = transmission delay, z = propagation delay per link

- A router must receive the whole packet before forwarding it, so each hop costs $x+z$.

- Path through one router with no queue: $2x+z$

- With n packets queued at the router, your packet finishes arriving at $x+z$ and the queue needs $n \cdot x$ to drain. Total = $\max(x+z, n \cdot x) + x + z$

- No wait if $n \cdot x \leq x+z$, so the max queue that causes no delay is $(x+z)/x$

Link State

- Hello count = $2 \times (\# \text{ router-router links})$. Each router says hello to each neighbor; host links don't count.

- Loops happen when routers have inconsistent topology views (Ex: a dropped advertisement)

- Periodic re-advertisement fixes stale views

- Routers never change their tables because of user packets; they just keep forwarding

Traceroute

(Ex: 33434)

- Sends UDP probes with TTL=1,2,3... to a high port

- The router where TTL hits 0 replies with ICMP Time Exceeded, source = the destination's IP.

- The ICMP reply payload is the original IP header plus the first 8 bytes, so the UDP ports appear in the reply

- Replies are forwarded via table [original src]. A router generating replies handles those packets as punt traffic.

End to end principle says that only end hosts alone are sufficient to guarantee reliability.