CS 118 – Computer Network Fundamentals

Fall 2024 – Professor George Varghese

LANs

Datalink: Connects multiple senders and receivers

- Low error rates → Little need for error recovery
    - Issue: Medium Access: Multiple senders attempt to grab link
- Sequential: Only one sender sends at a time
- Multiaccess: Multiple nodes can access the link
- Broadcast: Every transmission is heard by all other stations

Local Area Networks: Geographical area serviced is local and small

- Medium Access
    - Cost of Medium Access increases with propagation delay in the link
- Multicast: Refined broadcast with sends frame to set of receivers
    - All receivers receive all transmissions, however the hardware filters out any non-addressed messages
- Cheap: Saves wiring cost compared to router ports

Fixing Medium Access: Multiplexing

- Strict Multiplexing: Fixed allocations regardless of whether user has to send data or not (B/N)
    - FDM (Frequency Division Multiplexing): Assign frequencies to all broadcasters
    - TDM: Assign timeslots when broadcasters can send
    - Issue: User Traffic is bursty (maximum traffic/average ratio)
- Statistical Multiplexing: Busy users can access bandwidth of idle users (B/x)
    - Higher throughput, lower latency
    - Fairness: bandwidth is allocated fairly across all users who want to use the link
    - Centralization: Send all requests to a manager
        - Efficient for single node containing all users
        - Inefficient for LANs

        - Manager is bottleneck and failure prone
        - Cost: Requests need to be parsed recursively
        - Increased latency (propagation delay) in idle system
    - Decentralization used to solve LAN medium access

Ethernet

- CSMA/CD
    - Listen before transmitting
        - If channel is idle, send entire packet, otherwise defer
    - Carrier Sense: Would-be speakers wait for pause in conversation
        - Detect DC voltage or sense Manchester signal receipt
        - Hidden Terminal Problem: A & C cannot hear each other
            - A sends Request to Send (RTS) to B
            - B sends a Clear to Send (CTS)
    - Collision Detection: Multiple speakers try to speak, some may detect it and stop talking
        - Idea: It takes time for data signals and collisions to propagate along the Ethernet wire
        - Detect BEFORE transmission → Less wasted bandwidth
        - Manchester Encoding with non-zero net DC bias
    - Backoff: Speaker waits random amount of time before trying again
        - Issue: Static schemes of retransmission do not work for large amounts of colliders
        - Binary Exponential Backoff: Backoff time grows exponentially for every unsuccessful collision, reset after successful transmission
- Semi-Reliability: Ethernet does its best to retransmit (in hardware) all frames lost due to collisions
    - Thus senders must be aware of all collisions to retransmit
    - Slot: 2T (maximum propagation delay between two stations)
        - Max time to detect a collision involving own transmission

    - Min Size Packet: Necessitates length field so receivers know length of real data
    - Jamming: Add more bits to collided bits to enforce a collision

Aloha

- Problem: Multiple islands on Hawaii need to communicate with central computer
- Solution: Two channels
    - Contention Channel: Shared with everyone to send to computer
    - Reverse Broadcast: For computer to broadcast to all clients
    - Collision detection after frame transmission (in software)
    - Random backoff algorithm uses fixed number of random slots
- Issue: Cannot handle dynamic loads
    - Vulnerable to frames that start before another transmits
    - Vulnerable to frames that start after another transmits
        - We send the entire frame before we get an ACK

Ethernet Etc

- Every Ethernet has its own MAC Address
- Repeaters output same voltage at stronger magnitudes
- Issues:
    - Limited Range
        - Fixed with bridges
    - Performance gets worse at high loads
    - Does not work well on fiber and twisted pair
        - Token Rings: Use token to speak
            - Poor latency at low loads
            - Good throughput at high loads

Token Rings

- Small Token Passing Delays
    - Cut-through forwarding: forward before everything is received
- Frame Stripping: A source can strip a frame its send to prevent loops

    - Only source can strip otherwise we won't be able to multicast

IBM Ring

- Free Token and Start of Frame differ by at most 1 bit
- Sender does not pass token until last bit of last frame arrives (head of next)
- Initialized by monitor node (estimate of initial passing delay)

Bridging

Issues

- Ethernet throughput bottoms out as with enough senders, it only retransmits
    - Not true
- Ethernet is limited to ~1.5Km
- Routers were way slower than Ethernet
- Multiple protocols are used, routers need to adapt to multiple

Ideas:

- Repeaters connect via physical layer information
- Bridges connect via Data Link layer
    - Bridges make multiple Ethernets look like one big Ethernet

Components:

- Packet Repeater: Bridges pick up entire packet, buffers it, and transmits at next opportunity
- Filtering Repeater: Bridges have table that maps station addresses to Ethernets
- Filtering Repeater with Learning: Bridges check source addresses to learn additional information (about where the sender is)

Wire-Speed Forwarding: Bridge looks up destination and source addresses in time it takes for min size packet o arrive on Ethernet

- Fixes packet drops due to slower, overfilled buffers
- Spanning Tree Algorithm: Creates loop-free topology by having redundant bridges turn off bridge ports

Scaling Lookups to Higher Speeds:

- Binary Search Forwarding does not scale to FDDI Speeds
    - Upgrade to SRAM is expensive
    - 2 Binary Search lookups eat up all 2T time
- Gigaswitch: Perfect Hashing
    - CAM Lookup in parallel with hasp lookup

Issues

- 802 Address is flat, Routers only learn addresses within each level fo hierarchy
- Spanning Tree is inefficient and does not provide shortest path
- Flooding wastes throughput

IP Addressing

IP

- Connectionless Routing: No call is set up, each packet routes itself
    - Paths are unique most of the time
- Hierarchical Addressing: variable length prefixes
- Domain Name Service (DNS): Translates IP address to domain

Forwarding

- Final Hop Reached → Map to local address
- Lookup Router Table: If exists, forward to nexthop, otherwise go to default router

Addressing

- Classes
    - Core router stores 3 sets of hash tables
    - Ethernet addresses start with 0 → Class A
        - Net numbers
    - 10 → Class B
        - Any network with 255+ addresses
        - CIDR scheme: New organizations receive contiguous Class C addresses, queried by a common prefix
    - 110 → Class C
        - NAT: Using TCP port numbers in every packet to extend IP address space
            - Use IP addresses locally
            - NAT box contains single unique global address
                - Distinguishes from port numbers in reply
                - Issue: NAT end hosts are not reachable from the internet
                - Issue: End hosts may not be aware of external IP Address
        - IPv6

Lookups

- Unibit Trie → 32 steps in worst case, too slow
- Multibit Trie → 11 steps in worst case, too slow
- Ternary Cam → Parallel, 1 step in worst case, too much power

Prefix Notation: Decimal points separate bytes

Neighbor Routing

- Routers needs a network of endnode addresses
    - All endnodes on same LAN use same prefix, thus check prefix
- Routers need Data Link addresses of nodes
    - IP sends ARP request to entire LAN for the MAC
    - address
- Endnodes need Data Link address of at least one router
    - DHCP servers accessed through multicast addresses returns an endnode's prefix and router
    - DHCP: Given a MAC Address, assign a unique IP address
- Endnode traffic should not go through routers: endnodes check for matching prefixes before sending to router
    - If match, send directly
    - If not, ARP
- Endnode traffic eventually goes through routers: Routers send redirect packets to sender. Sender caches it
- If routers are down, endnodes should still communicate: Modified ARP

Route Computation

Problem: How do we route between subnets that are connected by some path of routers?

Topology:

- Company is given a domain name and an Autonomous System number
- Wide Area Network: Collection of ISPs (with own AS numbers)

Intra-domain routing (within a domain)

- Distance vector (in enterprises)
    - Gossiping: Bridges send (root, distance, parent) to all neighbors
        - No neighbor can look down/upstream
        - Occasionally send hellos to ensure everyone still exists
    - Router contains:
        - Vector of (ID, distance)
        - Port database for each neighbor
            - Stores new incoming distance vectors
            - Run distance vector algorithm on vector, update central database and forward if needed
        - Central Database
            - Cost and port number to reach subnet
    - Link Failures:
        - Count-to-infinity: Packet ping pongs between two routers and a packet does not actively avoid previously visited routers
            - Fix: Each distance vector update lists its path of routers on top of distance
- Link State Routing (in ISPs)
    - Approach
        - LSP Generation: Create link state packet containing a node's neighbors and outgoing costs
            - LSP updates if link dies → More respondent to fails
        - LSP Propagation: Each source broadcasts LSP to all other nodes using intelligent flooding

        - Intelligence: Add sequence number to LSP packet
            - Take the better packet, then take the most recent packet, otherwise send back newer LSP
        - Jumping: Don't drop packets, send back corrections
        - Aging: All LSPs age out after a half hour
    - All nodes know all other nodes now. Use Dijkstra to compute shortest path from S → D
        - May run multiple times as more LSP packets update

Inter-domain routing

- Policy routing

BGP (Border Gateway Protocol)

Ideas

- Use prefixes (denoted by / or *) to become stateless
- Shorter AS paths are preferred
- Allows integration of policies

Process

- Establish session
- Exchange all active routes
- Exchange incremental updates repeatedly
    - Announcement when ID is added to new active route
    - Withdrawal if active route is no longer available

Issues:

- Neighbor's best routes may not be your best route
- AS Path Length doesn't actually measure distance or latency

Interdomain protocols (BGP) & EGPs

- Concerned with providing reachability information
- Facilitates routing policy implementation in scalable manner
- Routes from B→A means B agrees to forward packets send by A

Interior Gateway Protocols

- Concerned with optimizing path metrics

Inter-AS Relationships

- Provider-consumer-transit
    - ISP provides access to all destinations in its routing tables
    - We want higher access, so we fatten the pipeline to connect to as many ASes as we can (consumers)
- Peering
    - Two ASes provide mutual access to subset of each others' routing tables
        - No financial settlement
    - We provide routes as needed
- Selective Transit: transit and peer when you feel like it

- Priorities: Customer > Peer > Provider

BGP

- Sessions
    - eBGP: Standard, exchanges network routing information across ASes
    - iBGP: Between BGP routers in same AS, allows internal routers to exchange information about external routes
- Route Reflector: Selects best route to each destination, announces it to all clients, readvertise newer, better routes
- Policies
    - Next Hop Attribute : IP address of next router
    - AS Path : Sequence of ASes traversed
        - Avoids loops, used to select when no preference
    - Local Preference : Criterium to pick routes
    - Multiple Exit Discriminator : Compares routes of neighbor AS
        - Ignored if no financial incentive, hot potato routing
    - Community: Data used for tag routing
- Inefficient failure and recovery
    - Fault detected → Future propagation is damped/delayed

TCP

Sockets: provides abstraction of shared queue between two machines

- TCP takes data from socket queue and reliably delivers it to remote socket queue

Main concepts:

- Packetizing: TCP breaks data into packets (not data streams),
- Sliding Window protocol: Use seq numbers, current window size as states
- Disconnection after finished

Nuances

- Network (not FIFO) → Packets can be delayed, reordered, and duplicated
    - Sequence number applied per byte (not per packet)
- Creating connections: 3-way handshake
    - Sender sends a SYN
    - Receiver sends SYN-ACK with new number Y
    - Sender sends SYN-ACK with Y and new data
- Congestion Control
    - Go-Back-N: Does not accept out of order packets, retransmit all packets from point of loss
    - Selective Reject: ACKs specify which packets were received out of order
    - No congestion → Increase window size
    - Congestion detection → Decrease window size
        - Implicit: Packet loss, packet delay
    - RED Router: Detects congestion before all buffers are full
        - DECbit: Router sends congestion experienced bit to source when average queue size goes beyond threshold
        - Drop a packet with small probability: Use fast retransmit
- Retransmit timers: Dynamically compute round-trip delay
- Fast Retransmit: Use 3 duplicate ACKs to detect a loss and retransmit
- Slow Start: Find the equilibrium speed
- Fast recovery: Avoid stall after loss

QUIC

- Simplified handshake

- Places multiple streams in one connection, losing one doesn’t affect performance

Applications

HTTP

- Server is stateless: Has no memory of previous requests
- Non-persistent
    - TCP connection opened
    - At most 1 object sent over TCP, TCP closed
    - Requires 2 RTTs per object
- Persistent
    - TCP connection opened to a server
    - Multiple objects can be sent between client and server, TCP closed
    - Client requests as soon as it encounters referenced object
- Cookies maintain state
    - First class: Tracks user behavior on website
    - Third class: Tracks user behavior across multiple websites

CLNP

- 20 byte address
    - Bigger than IPv6
    - Top 14 bytes: Prefix shared by all nodes in a cloud
        - Allows for levels of hierarchy
    - Bottom 6 bytes: Once you get to the cloud, route based on endnode ID
    - No need to ARP as the address is already stored
        - Allows you to follow “true layering”
- Zero configuration inside of cloud (1 prefix per cloud vs per link)

Worst Decision Ever

- 1992: Internet could have adapted CLNP
- Idea: Modify TCP to work on top of CLNP
- Issues with IPv6
    - 8 bytes on bottom is too much