Chapter 4

Network-layer

Network-layer services and protocols

- takes segment from sending host to receiving host
  - sender: encapsulates segment into datagram, passes to link layer
  - receiver: delivers segment to transport layer protocol
- network layer protocols in every Internet device: hosts, routers
- **Router**:
  - examines header fields in all **IP datagrams** passing through it
  - moves datagram from input port to output port along end-end path

Two key network-layer functions

network-layer functions:

- **routing:** determine route taken by packets from source to destination – routing algorithm (running on the control plane)
- **forwarding:** move packets from a router's input link to appropriate router's output link

Data plane, control plane

Data plane:

- local, per-router function
- determines how datagram arriving on router input port is forwarded to router output port

Contorl plane:

- network-wide logic
- determines how datagram is routed among routers along end-end path from source host to destination host
- two control-plane approaches:
  - traditional routing algo: implemtned in routers
  - software-defined networking(SDN): implemented in remote servers

Per-router control plane – traditional routing algo

- Individual routing algorithm components in each and every router interact in the control plane
- every router has a forwarding table where it maps from header field value to the one of the output links of the router
- routing algorithm is on control plane while forwarding table is on the data plane

SDN control plane

- Remote controller computes, installs fowrading table in routers
- everything is running on remote controllers

Network service model

services for individual datagrams:

- guaranteed delivery
- guaranteed delivery with less than 40 mses delay

services for a flow of datagrams:

- in-order datagram delivery
- guaranteed minimum bandwidth to flow
- restrictions on changes in interpacket spacing

Internet "best effort" service model:

- easy to deploy and adopt

NO guarantees on:

- successful datagram delivery to destination
- timing or oder of delivery
- bandwidth available to end-end flow

Router

architecture

- high-speed switching fabric:
  - forwarding data plane, operates in nanosecond timeframe
- routing processor:
  - creating, maintaining and updating the forwarding rules
  - routing, managment control plane operates in millisecond time frame

Input port functions

- physcial layer: bit-level reception; reciving bits
- link layer: rebuilding the frame and extract the datagram
- network layer protocal:
  - processing the datagram
  - using header field values, lookup output port using forwaring table in input port memory; try to match to the forwarding table
  - goal: complete input port processing at "line speed"
  - input port queuing: if datagrams arrive faster than forwarding rate into switch fabric
  - **destination-based forwarding:** forward based only on destination IP address (traditional)
  - **generalized forwarding:** forward based on any set of header field values

Longest prefix matching: when looking for forwarding table entry for given destination address, use longest address prefix that matches destinations address

- IP address is 4 bytes – 32 bits
- in case the matching are the same, choose the longest prefix matching
- often performed using ternacy content addressable memories (TCAMs)
  - **content addressable**: present address to TCAM: retrieve the mapping between an address and output port in one clock cycle, regardless of table size

Switching fabrics

- Transfer packet from input link to appropriate output link
- switching rate: rate at which packets can be transfer from inputs to outputs
  - often measured as multiple of input/output line rate
  - N inputs: switching rate N times line rate desirable
  - rate ideally NR
- Input port queuing:
  - queueing delay and loss due to input buffer overflow
  - Head-of-the-line (HOL) blocking: queeued datatgram at front of queue prevents others in queue from moving forward

Output port queueing

- Buffering required when datagrams arrive from fabric faster than link transmission rate.
  - datagrams can be lost due to congestion, lack of buffers
- Drop policy: depends on the implementation on the router
  - scheduleing discipline chooses among queued datagrams for transmission
    - priority scheduling: packet has higher priority will be send first
    - network neutrality: each packet has equal priority
- size of buffer: RTT $\times$ link capcity R
  - too much buffereing can increase delays
  - long RTTs: poor performance for real-time apps, just keep buffer busy enough

Buffer management:

- drop: which packet to add, drop when buffers are full
  - tail drop: drop arriving packet
  - priority: drop/remove on priority basis
- marking: which packets to mark to signal congestion (ECN, RED)

Packet Scheduling: FCFS

- packets transmitted in order of arrival to output port

Scheduling policies: priority

- arriving traffic classified, queued by class
  - any header fields can be used for classification
- send packet from highest priority queue that has buffered packets
  - FCFS within priority class

Scheduling policies: round robin

- arriving traffic classifed, queued by class
  - any header fileds can be sued for classification
- sercer cyclically, repeatedly scans class queues, sending one complete packet from each class in turn

Scheduling policies: weighted fair queueing

- generalized Round Robin
- each class, i, has weight, w_i, and gets weighted amount of service in each cycle :
$$\frac{w_i}{\sum_j w_j}$$
- that is established a priority for each class. Larger weight with more packets in the class

Internet network layer

It contains

- Path-selection algorithms:
  - implemented in routing protocols / SDN controllers
  - it is for forwarding table
- IP protocol
  - dictate the IP datagram format
  - the way how hosts, routers and network interfaces are addressed
  - dictates packet handling conventions
- ICMP protocol
  - dictates the way errors have to be reported
  - dictates how router signal themselves

IP fragmentation/ressembly

- network links have MTU(max, transfer size) – largest possible link-level frame
  - different link types, different MTUs
  - MTU: largest possible link layer frame size
- large IP datagram divided within net
  - one datagram becomes several datagrams
  - "ressembled" only at destination
  - IP header bits used to identify, order related fragments
  - When a large datagram has been forward to a relatively small MTU, unless the datagram has been marked, the large datagram will be splitted into 3 smaller datagrams. However, even next hub has larger MTU, they will not be assembled into the initial IP datagram; only the receiver will reassembled all the framgment datagrames.
  - e.g. 4000 bytes datagram; MTU 1500 bytes
    * 4000 - 20 = 3980 bytes payload data; as the header always takes up 20 bytes
    * 1500 bytes MTU = 1500 bytes datagram = 20 bytes header + 1480 bytes payload
    * total fragment = $\lceil 3980/1480 \rceil = 3$
    * **first** fragment datagram: 1480 bytes payload + 20 bytes header = 1500(length); offset = 0
    * **second** fragment datagram: 1480 bytes payload + 20 bytes header = 1500(length); offset = $1480 / 2^3 = 185$
    * **third** fragment datagram: 3980 - 1480 - 1480 = 1020 bytes payload + 20 bytes header = 1040(length); offset = $\frac{(1480+1480)}{2^3} = 370$
  - each fragflag has been set to 1 except the last one which has been set to 0 (as it is the last fragment).
  - Why we want to divide by 8? because the fragment offset field (13 bits) is three bits shorter than the length field(16 bits), thus $2^3 = 8$.

IP address

- 32-bit identifier associated with each host or router interface
- interface: connection between host/router and physcial link
  - router's typically have multiple interfaces
  - host typically has one or two interfaces s

Subnet

- device interface that can physically reach each other without passing through an intervening router
- IP addresses have structure:
  - subnet part: devices in same subnet have common high order bits
  - host part(right most byte): remaining low order bits
    - # of bit host part = 32 - # of bit subnet part
  - 223.1.1.1 $\rightarrow$ 223.1.1 subnet part, last 1 to be host part
- subnet mask: /24 (high-order 24 bits: subnet part of IP address)
- subnets defined by address and mask
- CIDR: Classless InterDomain Routing
  - subnet portion of address of arbitrary length
  - address format: a.b.c.d/x, where x is the # bits in subnet portion of address; x is less than 32

DHCP: Dynamic host configuration protocol

The machine will be assigned an IP address by DHCP, once the machine is on in a new subnet, it will request an IP address in a pool of available IP address from DHCP server on the network.
goal: host dynamically obtain IP address from network server when it joins network

- it can renew its lease on address in use
- allow resue of addresses(only hold address while connected/on)
- support for mobile user who join/leave network

DHCP overview:

- host broadcast DHCP discover msg [option]
- DHCP server responds with DHCP offer msg [optional]
- host requests IP address: DHCP request msg
- DHCP server sends address: DHCP ack msg

DHCP can return more than just allocated IP address on subnet:

- address of first-hop router for client
- name and IP address of DNS server
- network mask (indicating network vs host portion of address)

Connection example:

- Connecting laptop will use DHCP to get IP address, address of first-hop router, address of DNS server
- DHCP REQUEST message encapsulated in UDP, encapsulated in IP(source:0.0.0.0, dest: 255.255.255.255), encapsulated in Ethernet
- Then ethernet frame broadcast(dest:FFFFFFFFFFFF) on LAN, recevied at router running DHCP server
- Ethernet demux'ed(abstract) to IP demux'ed, UDP demuxed to DHCP
- DHCP server formulates DHCP ACK contining client's IP address, IP address of first-hop router for client, name& IP address of DNS server
- encapsulated DHCP server reply forwarded to client, demuxing up to DHCP at client
- client now knows its IP address name and IP address of DNS server, IP address of its first-hop router

Hierarchical addressing: route aggregation

- hirerarchical addressing allows efficient advertisment of routing information

NAT: network address translation

NAT: all devices in local network share just one IPV4 address as far as outside world is concerend.

- all datagram leaving local network have same source NAT IP address; but with different source port number
- datagrams with source or destination in this network have xx.x.x/xx address for source destination (from host to router)
- all devices in local network have 32-bit addresses in a "private" IP address space that can only be used in local network

Advantages:

- one IP address needed from provider ISP for all devices

- can change address of host in local network without notifying outside world
- can change ISP without changing addresses of devices in local network
- security: devices inside local net not directly addressable, visible by outside world

Implementation: NAT router must (transparently):

- outgoing datagrams: replace (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
  - remote clients/servers will respond using (NAT IP address, new port #) as destination address
- record in NAT translation table every (source IP address, port number) to NAT IP address, new port number translation pair
- incoming datagrams: replace (NAT IP address, new port #) in destination fields of every incoming datagram with corresponding source IP address, port #) stored in NAT table

Disadvantages:

- routers should only process up to layer 3
- address shortage should be solved by IPV6
- violates end-to-end argument (port numbe rmanipulation by network-layer device)
- NAT traversal

Summary:

- NAT is playing a gather role, it establishes a "public" IP address for the outside world so that every devices in the subnet does not need to know the outside, but when sending packets to outside NAT will create a mapping table of host source address/port to "public" IP address/ port

IPV6 motivation

- initial motivation: 32-bit IPV4 address space would be completely allocated
- additional motivation:
  - speed proessing/forwarding: 40-byte fixed length header
  - enable different network-layer treatment of "flows"
- Missing component:

    - no checksum to speed processing at routers
    - no fragmentation/reassembly
    - no options (available as upper-layer, next-header protocol at router)

- tunneling: IPV6 datagram carried as payload in IPV4 datagram among IPV4 routers
    - tunneling used extensively in other contexts

Generalized forwarding: match plus action

- destination-based forwarding: forwward based on destiation IP address
- generalized forwarding:
    - many header fields can determine acation
    - many action possible: drop/copy/modify/log packet

Flow table abstraction

- flow: defined by header field values (in link, netowrk, transport-layer fields)
- generalized forwarding: simple packet-heandling rules
    - match: pattern values in packet header field
    - actions: fow matched packet: drop, forward, modify, matched packet or send matched packet to controller
    - priority: the highest priority detemines the forwarding behaviour of datagram
    - counters: number of bytes and number of packets

Openflow abstraction

match + action: abstraction unifies different kinds of devices

- router:
    - match: longest destination IP prefix
    - action: forward out a link
- Switch:
    - match: destination MAC address
    - action: forward or flood
- Firewall:

  - match: IP addresses and TCp/UDP port numbers
  - action: permit or deny

- NAT:
  - match: IP address and port
  - action: rewrite address and port

Chapter 5

Routing protocols

Routing protocol goal: determine "good" paths from sending hosts to receiving host, through network of routers

- path: sequence of routers packets traverse from given initial source host to final destination host
- good: least cost

Routing algorithm classification

- global: all routers have complete topology, link cost info
  - link state algorithm
- decentralized: iterative process of computation, exchange of info with neightbors
  - routers initially only know link costs to attached neighbors
  - distance vector algorithms

Dijkstra's link-state routing algorithm

- centralized: network topology, link costs known to all nodes
  - accomplished via "link state broadcast"
  - all nodes have same info
- computes least cost paths from one node ("source") to all other nodes
  - gives forwarding table for that node
- iterative: after k iterations, know least cost path to k destinations
- algorithm complexity: n nodes

  - each of n iteration: need to check all nodes, w, not in N
  - n(n+1) / 2 comparisons: $O(n^2)$ complexity
  - more efficient implementations possilbe: O(nlogn)

- message complexity:
  - each router must broadcast its link state information to other n routers
  - efficient broadcast algorithms: O(n) link crossings to disseminate a broadcast message from one source
  - each router's message crosses O(n) links: overall message complexity: $O(n^2)$

Distance vector algorithm

Key ideas:

- from time-to-time, each node sends its own distance vector estimae to neighbors
- when x receives new DV estimate from any neighbor, it updates its own DV using B-F equaiton:
$$D_x(y) = \min_v\{c_{x,v} + D_v(y)\} \text{ for each node } y \in N$$
- $D_{source,target} = \min\{D_{neightbor,target} + c_{source,neighbor}\}$ for all neightbor
- under minor, natural conditions, the estimate $D_x(y)$ converge to the actual least cost $d_x(y)$
- iterative, asynchronous, distributed, self-stopping
- good news travels fast, bad news travels slow

Comparision of LS and DV algorithms

message complexity

- LS: n routers, $O(n^2)$ messages sent
- DV: exchange betweeen neighbors; convergence time varies

Speed of convergence

- LS: $O(n^2)$ algorithm, $O(n^2)$ messages
  - may have oscillations

- osillation: suppose two routes(say A and B) to target router, suppose one route is relatively cost-free, then all the packets have been sent on route A which results in congestion problem. It will result in the increasing cost on route A. Once cost A is higher than cost B, then all the packets switch to send through route B which also results in congestion problem. It will result in the increasing cost on route B. Therefore, they switch back send through route A. Thus A to B, B to A.

- DV: convergence time varies
  - may have routing loops
  - count-to-infinity problem

Robustness:

- LS:
  - router can advertise incorrect link cost
  - each router computes only its own table
- DV:
  - DV router can advertise incorrect path cost
  - each router's table used by others: errors propagate through network

OSPF(Open Shortest Path First)

Internet approach to scalable routing

aggregate routers into regions known as "autonomous systems" (AS)

- **intra-AS (intra-domain)**: routing among within same AS
  - all routers in AS must run same intra-domain protocol
  - routers in different AS can run different intra-domain routing protocols
  - gateway router: at "edge" of its own AS, has links to routers in other AS'es
- **inter-AS (inter-domain)**: routing among AS'es
  - gateways perform inter-domain routing

Interconnected ASes

- forwarding talbe configured by intra- and inter-AS routing algorithms
  - intra-AS routing determine entries for destinations within AS
  - inter-AS and intra-AS determine entires for external destinations

Inter-AS routing: a role in intradomain forwarding

Inter-domian routing must

- learn which destinations reachable through AS2, which through AS3
- propagate this reachability info to all routers in AS1 so that each router knows which gateway router they should forward to

Intra-domain routing

- it is intra-domain routing's job to tell each router how to reach gateway router

OSPF routing

- public avaliable
- classic link-state
  - each router floods OSPF link-state advertisements to all other routers in entire AS
  - multiple link costs metrics possible: bandwidth, delay
  - each router has full topology, uses Dijkstra's algorithm to compute forwarding table
- security: all OSPF messages authenticated

Hierarchical OSPF

- Two-level Hierarchichy: local area(area 1, ..., area n), backbone(area 0)
- all routers in the same area called internal routers or local routers have the same topology table/map (link state database)
- each router will have a different routing table as OSPF compute the best path for each router depending on its location within the network or the area topology
- **The goal of having area:**
  - the area boundaries will give the opportunity of using summerization – reducing the routing table
    - summerize network prefixes within the same area into a single prefix to be advertised to other areas
  - help the containment to the area to update messages that are sent out when a change occurs in the state of a link – reducing the number of routing messages exchanged over the netwrok

- **Routers in OSPF**
  - area border routers: summerize distances to destinations in own area, advertise in backbone
  - local/internal routers: compute routing within area/ forward packets to outside via area border router
  - boundary router: connects to other ASes
  - backbone router: runs OSPF limited to backbone
  - designated router:
    - elected in local routers – collecting and flooding link state advertisment and link state updates
    - update to all area routers using multicast(implemented in IP level) address – 224.0.0.5
    - receive the advertisment and update each router within area and propagate these updates to the rest of the routers within the area
  - backup designated router:
    - elected along with the designated router when the designated router fail

Routing among ISPs: BGP

Inter-domain routing protocol – glue holds the internet together

- allow subnet to advertise its existence and the destinations it can reach to rest of Internet
- BGP provides each AS a means to:
  - eBGP(external BGP): obtain subnet reachability information from neighboring ASes
  - iBGP(internal BGP): propagate reachability information to all AS-internal routers.
  - determine ”good” routes to other networks based on reachability information and policy
- gateway routers run both eBGP and iBGP protocols

BGP basics

- BGP session: Two BGP routers (”peers”) exchange BGP messages over semi-permanent TCP connection:
  - advertising paths to different destination network prefixes (BGP is a ”path vector” protocol)

- When new subnet X has been attached to ASn, ASn gateway will advertises path ASn, X to ASm gateway:
  - ASn promises to ASm it will forward datagrams towards X

Path attributes and BGP routes

- BGP advertised route: IP prefix + attributes
  - prefix: destination being advertised
  - two important attributes:
    - AS-PATH: list of ASes through which prefix advertisement has passed
    - NEXT-HOP: indicateds specific internal-AS route to next-hop AS
- Policy-based routing:
  - gateway receiving route advertisement uses import policy to accept/decline path
  - AS poslicy also determines whehther to adverstise path to other neightboring ASes

BGP messages

- BGP messages exchanged between peers over TCP connection
- BGP messages:
  - OPEN: opens TCP connection to remote BGP peer and authenticates sending BGP peer
  - UPDATE: advertises new path (or withdraws old
  - KEPPALIVE: keep conection alive in absence of UPDATES; also ACKs OPEN reques
  - NOTIFICATION: reports errors in previous msg; also used to close connection

Intra-, Inter-AS routing

- **Policy**:
  - inter-AS: admin wants control over how its traffic routed, who routes through its network
  - intra-AS: single admin, so policy less of an issue
- **Scale**:
  - hierarchical routing saves table size, reduced update traffic

- **Performance**
  - intra-AS: can focus on performance
  - inter-AS: policy dominates over performance

Hot potato routing

- choose local gateway that has least intra-domaincost(internal cost) don't worry about inter-domain cost(external cost)

BGP route selection

- router may learn about more than one route to destination AS, selects route based on
  - local preference value attribute: policy decision
  - shortest AS-PATH
  - closest NEXT-HOP router: hot potato routing
  - additional criteria

ISP only wants to route traffic to/from its customer network

Software Defined networking(SDN)

- Pre-router control plane:
  - Individual routing algorithm components in each and every router interact in the control plane to compute forwarding tables
- SDN: remote controller computes, installs forwarding tables in routers
- Why a logically centralized control plane?
  - easier network management: avoid router misconfigurations, greater flexibility of traffic flows
  - Table-based fowarding allows "programming" routers
    * centralized programming easier: compute talbes centrally and distribute
    * distributed programming more difficult: compute tables as result of distributed algorithm implemented in each-and-every router
  - open (non-proprietary) implementation of control plane

Four Features:

- generalized "flow-based" forwarding

- control, data plane separation
- control plane functions extenral to data-plane switches
  - block a flow for security concern
  - split a flow for load balancing
  - route a flow on the lowest cost path based on Dijkstra algorithm
- programmable controal applications
  - route and access control and load balancing

Data-plane switches:

- fast, simple, commodity switches implementing generalized data-plane forwarding in hardware
- flow table computed, installed under controller supervision
- API for table-based switch control
  - defines what is controllable what is not
- protocoal for communicating with controller

SDN controller

- maintain network state information
- interacts with network control applications "above" via northbound API
- interacts with network swtiches "below" via southbound API
- implemented as distributed system for performance, scalability, faultolerance, robustness
- Structures:
  - interface layer to network control apps: abstrcations API
  - network-wide state management: state of networks links, switches services: a distributed database
  - communication: communicate between SDN controller and contolled switches

network-control apps

- "brains" of control: implement control functions using lower-level services, API provided by SDN controller

- unbundled: can be provided by thrid party: distinct from routing vendor or SDN controller

OpenFlow protocol

- operates between controller, switch
- TCP used to exchange messages
- three classes of OpenFlow messages:
  - controller-to-switch
  - asynchronous (switch to controller)
  - symmetric
- distinct from OpenFlow API
  - API used to specify generalized forwarding actions

OpenFlow Controller

- key controller-to-switch messages:
  - features: controller queries switch features, switch replies
    * switch must reply features reply message that specified the features and the capability that are supported by the swtich
  - configure: controller queries/sets switch configuration parameters
    * allows the controller to set and query configuration params in the switch
    * the query only responds to a query from a controller
  - modify-state: add, delete, modify flow entries in the Openflow tables
    * sent by the controller to manage the state of the switches
  - packet-out: controller can send this packet out of specific switch port
- Key switch-to-controller messages
  - packet-in: transfer packet to controller.
    * switch that do not support internal buffering or run out of buffering will send the full packet to controller
  - flow-removed: flow table entry deleted at switch
    * add a hard timeout value: the entry should be removed regardless of the activity
    * flow modify messages also indicates if the switch should send a flow removal message to the controller when flow expires
  - port status: inform controller of a change on a port

SDN: control/data plane interaction example

1. xx experiencing link failure uses OpenFlow port status message to notify controller
2. SDN controller receives OpenFlow message, updates link status info
3. Dijkstra's routing algorithm application has previously registered to be called when ever link status changes. The routing algorithm has been called.
4. Dijkstra's routing algorithm access network graph info, link state info in controller, computes new routes.
5. link statte routing app interacts with flow-table-computation compoent in SDN controller, which computes new flow tables needed
6. controller uses OpenFlow to install new tables in switches that need updating

ICMP: internet control message protocol

- used by hosts and routers to communicate network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer "above" IP:
  - ICMP messages carried in IP datagram
- ICMP message: type, code plus first 8 bytes of IP datagram causing error

Chapter 6 – Link layer

Introduction

- hosts, routers: nodes
- Communication channels that connect adjacent nodes along communication path: links
- layer-2 packet: frame, encapsulates datagram

Link layer services

- Framing: encapuslate datagram into frame, adding header, trailer
  - "MAC" addresses used in frame headers to identify source, destination
    - different from IP address
  - a frame is a link layer packet which consists of a data field which the network layer datagram is inserted and the number of header fields
- Link access: channel access if shared medium
- reliable delivery between adjacenet nodes
  - seldom used on low bit-error link
  - wireless links: high error rates
- error detection:
  - errors caused by signal attenuation, noise
    - having transmission note include error detection bits
    - having the receiving node perform an error check
  - receiver detects presence of errors: singals sender for retransimission or drops frame

Link layer implementation

- in each and every host
- link layer implemented in adaptro (Network interface card (NIC)) or on a chip
- attaches into host's system buses
- link layer is a combination of hardward software, firmware

Multiple access links, protocoals

- two types of links:
  - point-to-point
    - PPP for dial-up access
    - point-to-point link between Ethernet switch, host
  - broadcast
    - old-fashioned Ethernet
    - upstream HFC
    - WIFI

  Multiple access protocol:

  - single shared broadcast channel
  - if two or more simultaneous transmission by nodes: interference, then collision will happen. The packets will be lost.
  - the protocol is a distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
    - so that it can avoid collision
  - Communication about channel sharing must use channel itself
    - no out-of-band channel for coordination
- An ideal multiple access protocol
  - broadcast channel of rate R bps
  - when one nodes want to broadcast, it can send at rate R
  - when M nodes want to broadcast, it can send at rage R/M
  - fully decentralized:
    - no special node to coordinate transmissions
    - no synchrnoization of clocks, slots
  - simple

  MAC protocols: taxonomy

  - Three broad classes:
    - channel partitioning
      - divide channel into smaller "pieces

        · allocate piece to node for exclusive use
    * random access
        · channel not divided, allow collisions
        · recover from collision
    * taking turns
        · nodes take turns, but nodes with more to send can take longer turns

Channel partitioning MAC protocols: TDMA

TDMA: time division multiple access

- Access to channel in "rounds"
- each station gets fixed length slot in each round
- unused slots go idle
- avoids collisions and divides the bandwidth fairly among the end nodes
- Two drawbacks:
    * nodes are limited to the average rate of r/n even when it is the only node with packet to send
    * a node must always wait for its turn in the transmission sequence even when it is the only node with packet to send

Frequency divdision multiple access: FDMA

- channel spectrum divided into frequency bands
- each station assigned fixed frequeny band
- unused transmission time in frequency bands go idle
- avoids collisions and divides the bandwidth fairly among the end nodes
- Two drawbacks:
    * nodes are limited to a bandwidth of r/n even when it is the only node with packet to send
    * a node must always wait for its turn in the transmission sequence even when it is the only node with packet to send

Random access protocols

- when node has packet to send
    * transmit at full channel data rate R

    * no a priori coordination among nodes
- two or more transmitting nodes
- MAC protocol specifies:
    * how to detect collision
    * how to recover from collisions
- examples of random access MAC protocols: CSMA, CSMA/CD, CSMA/CA

CSMA/CA(collision avoidance)

- if channel sensed idle: transmit entire frame
    * if channel sensed busy, defer transmission
- CSMA collisions:
    * collisions can still occur propagtion delay means two nodes may not hear each other's transmission
    * collision: entire packet transmission time wasted
    * distance & propagation delay play role in determining collision probability
    * the longer the propagation delay of the broadcast channel (a signal to propagate from one node to other nodes), the higher the chance the of the career sensing node is not yet able to sense a transmission that has already begun (the higher the chance of the collision rate)
    * the longer the broadcast career sensing channel, the higher the collision rate on the channel

CSMA/CD(collision detection) – CSMA with collision detection

Carrier sensing, deferral as in CSMA

- collisions detected within short time
- colliding transmission aborted, reducing channel wastage

Collision detection:

- A transmitting node listens to the channel while it is transmiitning its frame, if detected other node is transmitting and interfering with its own frame, then it will stop transmitting and wait for a certain amount of time before repeating the sense one idle cycle
- easy in wired LANs: detect signal incoming signal energy while transmitting
- difficult in wireless LANs: received signal strength overwhelmed by local transmission signal strength, that is why CSMA/CA is used in wireless LANs.

Ethernet CSMA/CD algorithm

- NIC receives datagram from network layer, create frame
- If NIC sense channel idle, starts frame transmission. If NIC sense channel busy, waits until channel idle, then transmits.
- If NIC transmits entire frame without detecting another transmission, NIC is done with frame
- If NIC detects another transmission while transmitting, aborts(stops) and sends jam signal
- After aborting, NIC enters binary exponential backoff
  - At m^{th} collision, NIC chooses K at random from 0 - $2^m - 1$. NIC waits $K \cdot 512$ bits times, returns to step 2
  - longer backoff interval with more collisions.
  - therefore it is possible for a NEEW node to get in while other nodes are in the exponential backoff

"Taking turns" MAC protocols

channel partitioning MAC protocols:

- share channel efficiently and fairly at high load
- inefffficient at low load" delay in channel access, 1/N bandwidth allocated even if only 1 active node

random access MAC protocols:

- efficient at low load: single node can fully utilize channel
- high load: collision overhaed

Taking turns MAC protocols:

- Polling:
  - master node "invidtes" slave nodes to transmit in turn
  - typically used with dumb slave devices
  - concers:
    - polling overhaed
    - latency: polling delay: the amount of time required to notify a node that it can transmit

        * single point of failure (master dead)

- token passing:
    - contorl token passed from one node to next passed from one node to next sequentially; virtually put them into a circle
    - token message
    - drawbacks:
        * token overhead
        * latency
        * single point of failure (token dead)

MAC address and ARP

The job of link layer switches is to carry datagram between switches and routers.
The host and router do not have to define the frame to the switches.

- 32-bit IP address:
    - network-layer address for interface
    - used for layer 3 forwarding
- MACE address:
    - function: used 'locally' to get frame from one interface to another physically-connected interface (same network, in IP-addressing sense)
    - 48 bit MAC address burned in NIC ROM, also sometimes software settable

Each adapter on LAN has unique LAN address. LAN addresses:

- MAC address allocation daministered by IEEE
- Manufacturer buys portion of MAC address space to assure uniqueness
- Analogy:
    - MAC address: like social security number
    - IP address: like postal address
- MAC flat address: portability
    - can move LAN card from one LAN to another
- IP hierarchical address not portable

    - address depends on IP subnet to which not is attached

- When an adpater sends a frame to a destination adapter, the sending adpater inserts destination adapter's MAC address into the frame and sent the frame into the LAN.

- When an adpater receives a frame, it checks to see if the destination MAC address of that frame mataches its own MAC address

    - if there is a match, the adapter enclosed datagram and pass it to the protocal stack
    - if there is not a match, the adapater discards the frame without passing the network layer datagram; unless the destination MAC address is a broadcase address (48 consecutive 1 – ff-ff-ff-ff-ff-ff)

ARP: address resolution protocal

To determine interface's MAC address, knowing its IP address:

- ARP table: each IP node on LAN has table

    - IP/MAC address mappings for some LAN nodes
    - TTL(Time to live): time after which address mapping will be forgotten (typically 20 mins)

To send a packet from host A and host B.

- Host A must provide the both IP address and MAC address of host B to A's adpater

- If A's ARP talbe does not have an entry fro the destination node(IP/MAC address to LAN B node)

    - The sender will use the ARP protocoal to resolve the destination MAC address
        1. sender constructs an arp packet(including the sending & receiving IP and MAC address)
        2. Both ARP query and response packets have the same format
        3. The goal of ARP packet is to query all the rest host in the subnet and determine the MAC address corresponding the IP address included in the query
        4. The adapte encapsulates the ARP packet in link layer frame, uses the broadcast address for the frame's destination address and transmit the frame into the subnet
        5. The frame contatin the ARP query is received by all the other adapters in the subnet, then send to the ARP module.
        6. ARP module checks if its IP address mathces the IP address in the ARP query; if matches, send the mapping for IP to MAC back to A

  - A update its ARP table by using the mapping

- The sent ARP message is boradcast, while the response ARP message is unicast.
- ARP table is built automatically.
- If a host leaves the subnet, the entry for that will eventually deleted because of TTL
- ARP is similar to DNS(host name to IP address)
  - DNS is resovle the host name and host in the internet
  - ARP is resolve hosts and routers interface only on the same subnet. (ARP query never leave subnet)

Addressing: routing to anthoer LAN
e.g. A and B are in different subnet, with 111.111.111.xxx and 222.222.222.xxx. They are connected by a router *R*. When A wants to send a packet to B, it needs to do the following:

- A creates IP datagram with IP source A, destination B
- A creates link-layer frame with R's MAC address as destination, frame contains A-to-B IP dataframe.(see slike 34)
- R forwards datagram with IP source A, destination B
- R create link-layer frame with B's MAC address as destination, frame contains A-to-B IP datagram.

Ethernet protocal and switches

"dominant" weird LAN technology

- cheap for NIC
- first widely used LAN technology
- simpler, cheaper than token LANs and ATM
- kept up with speed race: 10 Mbps - 10 Gbps

Ethernet: physical topology

- bus: popular through mid 90s
  - all nodes in same collision domain
- star: prevails today
  - active swtich in certer
  - each "spoke" runs a separate Ethernet protocal (nodes dont collide with each other)

Etherenet Frame structure

Sending adapter encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame

- preamble:
  - 7 bytes with pattern 10101010 followed by one byte with pattern 10101011
  - used to synchronize receiver, sender clock rates; as the other adapter do not know how the clock has drifted
- addresses: 6 byte source, destination MAC address
  - if adapter receives frame with matching destination address, or with broadcast address, it passes data in frame to network layer protocol
  - otherwise, adapter discards frame
- type: indicates higher layer protocal (the type of data(payload between 46 - 1500 bytes))) – mostly IP, but possible novell IPX, AppleTalk
  - payload is between 46 bytes to 1500 bytes
  - if exceed 1500 bytes, then send into different frames
  - if less than 46 bytes, the data field has to be stuffed
- CRC: cyclic redundancy check at receiver
  - error detected: frame is dropped

Ethernet: unreliable, connectionless

- connectionless: no handshaking between sedning and receiving NICs
- unreliable: receiving NIC does not send acks or nacks to sending NIC
  - data in dropped frames recovered only if initial sender uses higher layer RDT, otherwise dropped data lost
  - checking by CRC, no matter what the results are, never send the ACK
- Ethernet's MAC protocoal: unslotted CSMA/CD with binary backoff

Link & phsycial layer

- 100 mbps ethernet is limited to 100 meters distance over twisited pair, several kilometers over fiber
- GBps ehternet standards offer a raw data rate of 1000 mbps while maintaining full compatibility with the huge installed base of internet equipment
  - it allows point to point link(using switches) as well as shared broadcast channels(using hubs)
  - using CSMA/CD for shared broadcast channels
  - allows for full duplex operation at 1000 mbps in both directions for point to point channels
- Although ethernet has changed a lot, but the ethernet frame never changed $\rightarrow$ the timeless centrepiece of ethernet standards

Ethernet Switch

- link-layer device: takes an active role
  - store, forward Ethernet frames
  - examine incoming frame's MAC address, selectively forward frame to one-or-more outgoing links when frame is to be forwarded on segment, uses CSMA/CD to access segment
  - Switch table contains
    - a mac address of adapter on the LAN
    - a switch interface that leads to that adapter
    - the time at which the entry was placed in the table
- transparent
  - hosts are unware of presence of switches
  - that is host will address a frame to another host instead of another switch
- plug-and-play, self-learning
  - switch do not need to be configured
  - switch forward packets based on MAC addresses rather than IP addresses; switch table constructs in a very different manner than a router table

Switch: multiple simultaneous transimissions

- hosts have dedicated direct connection to switch
- switches buffer packets; it never transmit more than one frame on a segment at any one time
- Ethernet protocol used on each incoming link, but no collisions; full duplex
- every link in the LAN can operate at different speed and can run on different media
- switching: A-to-A' and B-to-B' can transmit simultaneously without collisions
- switch also eases network management; if a jabbering adapter happens(continuously sends ethernet frames), the switch will internally disconnet the not functioning adapter.
- Switch also collect data usage

Swith: self-learning

- switch learns which hosts can be reached through which interfcaes
  - when frame received swtich "learns" location of sender: incoming LAN segment
  - records sender/location pair in switch table

Switch: frame filtering/forwarding

- if switch table is empty, then it will do the following:
- firstly, with incoming frame, it will store the mac address in the frame's source address field
- then the interface from which frame arrived, then the current time
- index switch table using incoming frame's MAC destination address
- if entry was found for destination
  - if destination on segment from which frame arrived: forward frame on interface indicated by entry
  - else: forward on all interfaces except arriviing interface

Switches vs routers

both are store-and-forward:

- routers: network-layer devices (examine network-layer headers)
- switches: link-layer devices (examine link-layer headers)

both have forwarding tables:

- routers: compute tables using routing algorithms, IP addresses
- switches: learn forwarding table using flooding, learning MAC addresses
  - if they do not learn the specific MAC addres is, they will flood the output link

in terms of network size:

- switches are good for relatively small size of network, say tens or hundreds of hosts (campus, enterprise, home), but when the size of hosts is getting larger, it will create flood storms
- routers are good for relatively large size of network. It can help to find the lowest cost path in a large nework.

Virtual LAN

- virtually isolate traffic from a group of hosts from the rest of the traffic
- single broadcast domain:
  - scaling: all layer-2 broadcast traffic (ARP, DHCP, unknown MAC) must cross entire LAN
  - efficiency, security, privacy issues
- administrative issues:
  - locate at A physically attached to A switch but want to remain logically attached to B switch

Port-based VLAN

- switches supporting VLAN capabilities can be configured to define multiple virutal LANS over single phsyical LAN infrastructure
- **port-based VLAN**: switch ports grouped by switch management software so that single physical switch operates as multiple virtual switches

- **Traffic isolation**: frames to/frome ports 1-8 can only reach port 1-8 even it is a broadcast message to a host connected to port 9 - 16
  - can also define VLAN based on MAC addresses of endpoints, rather than switch port
- **dynamic membership**: ports can be dynamically assigned among VLANs
- **forwarding between VLANs:** done via routing (just as with separate switches)
  - we need to go to layer-3 to forward packet across VLANs
  - in practice vendors sell combined swithces plus routers

VLANs spanning multiple switches

- trunk port: carries frames between VLANs defined over multiple physical switches
  - frames forwarded within VLAN betwen switches cannot be vanilla 802.3 frames
  - 802.1 q protocal adds/removed additional header fields for frames forwarded between trunk ports

RECAP

1. Connecting laptop needs to get its own IP address, address of first-hop router, address of DNS server; using DHCP
2. DHCP request encapsulated in UDP, encapsulated in IP, encapsulated in 802.3 Ethernet
3. Ethernet frame broadcast(dest:FFFFFFFFFF) on LAN, recevied at router running DNCP server
4. Ethernet demuxed to IP demuxed, UDP demuxed to DNCP
5. DHCP server formulates DNCP ACK containing client's IP address, IP address of first-hop router for client, name and IP address of DNS server
6. encapsulation at DHCP server, frame forwarded(switch learning) through LAN, demulitplexing at client
7. DHCP client receives DNCP ACK reply
   - client now has IP address, knows name and address of DNS server, IP address of its first-hop router
8. before sending HTTP request, need IP address of xxxx.com $\rightarrow$ create DNS query
9. DNS query created, encapsulated in UDP, encaluspted in IP, encapsulated in Ethernet. to send frame to router, need MAC address of router interface $\rightarrow$ create ARP query
10. ARP query broadcast, received by router, which replies with ARP replaying giving MAC address of router interface
    - clinet now knows MAC address of first hop router, so can now send frame containing DNS query
11. IP datagram contiaining DNS query forwarded via LAN switch from client to first hop router
12. IP datargram forwarded from campus network into comcast network, routed to DNS server
13. DNS server replies to client with IP address of xxx.com
14. To send HTTP request, client first opens TCP socket to web server
15. TCP SYN segment inter-domain routed to web server
16. web server responds with TCP synack ; TCP connection established

17. HTTP request sent into TCP socket;
18. IP datagram containing HTTP request routed to xxx.com
19. web server responds with HTTP reply
20. IP datagram containing HTTP reply routed back to client