Transport Layer

provide logical communication between app processes running on different hosts

UDP (User Datagram Protocol)

- Header:
    - source port #
    - dest port #
    - length
    - checksum

- no connection establishment (which can add delay)
- simple: no connection state at sender, receiver
- small header size
- no congestion control: UDP can blast away as fast as desired
- sender:
    - treat segment contents, including header fields, as sequence of 16-bit integers
    - checksum: addition (one’s complement sum) of segment contents
    - sender puts checksum value into UDP checksum field

- receiver:
    - compute checksum of received segment
    - check if computed checksum equals checksum field value:

RDT (reliable data transfer)

- **Go-back-N**:

    - sender can have up to N unack’ed packets in pipeline
    - receiver only sends cumulative ack
        - doesn’t ack packet if there’s a gap

    - sender has timer for oldest unack’ed packet
        - when timer expires, retransmit all unack’ed packets

- **Selective Repeat**:

  - sender can have up to N unack’ed packets in pipeline
  - rcvr sends individual ack for each packet
  - sender maintains timer for each unack’ed packet

TCP (Transmission Control Protocol)

- TCP socket identified by 4-tuple:
  - source IP address
  - source port number
  - dest IP address
  - dest port number
- **demux**: receiver uses all four values to direct segment to appropriate socket
- point-to-point
- reliable, in-order byte stream
- connection-oriented
- full duplex
- pipelined
- flow control
- congestion control

Application Layer

Client-server architecture

- server:
  - always-on host
  - permanent IP address
  - data centers for scaling
- clients:
  - communicate with server
  - may be intermittently connected

  - may have dynamic IP addresses
  - do not communicate directly with each other

P2P architecture

- no always-on server
- peers request service from other peers, provide service in return to other peers
  - => **self scalability**

- peers are intermittently connected and change IP addresses
  - => **complex management**

Web and HTTP

HTTP: hypertext transfer protocol

1. client initiates TCP connection to server
2. server accepts TCP connection from client
3. HTTP messages exchanged between browser (HTTP client) and Web server (HTTP server)
4. TCP connection closed

HTTP is "stateless": server maintains no information about past client requests

- **non-persistent HTTP**:

  - at most one object sent over TCP connection
  - downloading multiple objects required multiple connections
  - **non-persistent HTTP response time** = 2RTT+ file transmission time

- **persistent HTTP**:

  - multiple objects can be sent over single TCP connection between client, server
  - server leaves connection open after sending response
  - subsequent messages sent over open connection

Non-persistent HTTP

1. HTTP client initiates TCP connection to HTTP server
2. HTTP server send accepts connection, notify server
3. HTTP client sends HTTP request message into TCP connection socket.

4. HTTP server receives request message, forms response message containing requested object, and sends message into its socket
5. HTTP server closes TCP connection
6. HTTP client receives response message containing html file, displays html

HTTP request message: ASCII (human-readable format)

HTTP/1.0: GET, POST, HEAD

HTTP/1.1: GET, POST, HEAD, PUT, DELETE

User-server state: cookies

- authorization
- shopping carts
- recommendations
- user session state

Web Caches (procy server)

Goal: satisfy client request without involving origin server.

- cache acts as both client and server
  - server for original requesting client
  - client to origin server

- typically cache is installed by ISP
- Why?
  - reduce response time for client request
  - reduce traffic on an institution’s access link

Conditional GET

cache: specify date of cached copy in HTTP request

Server: response contains no object if cached copy is up-to-date

E-mail

- **user agetnts**

    - a.k.a. mail reader
    - composing, editing, reading mail messages
    - outgoing, incoming message stored on server

- **mail servers**:

    - **mailbox**: contains incoming messages for user
    - **message queue**
    - **SMTP protocok**: between mail servers to send email mesages (for send and recv)

SMTP

- uses TCP to reliably transfer email message from client to server
- **direct transfer**
- three phases:
    - handshaking
    - transfer of messages
    - closure

- **commands** (ASCII) and **response** (status code and phrase)

HTTP vs SMTP

- PULL (HTTP) VS PUSH (SMTP)
- both have ASCII command/response, status codes
- SMTP uses persistent connections
- HTTP: each object encapsulated in its own response message
- SMTP: multiple objects sent in multipart message

Mail access protocols

- **SMTP**: delivery/storage to receiver's server
- **mail access protocol**: retrieval from server

    - **POP**: Post Office Protocol
    - **IMAP**: Internet Mail Access Protocol
    - **HTTP**

DNS: domain name system

- **distributed database** implemented in hierarchy of many name servers
- **application-layer protocol**: hosts, name servers communicate to resolve names (address/name translation)

DNS services

- hostname to IP address translation
- host aliasing
    - canonical, alias names

- mail server aliasing
- load distribution

- **root name servers**: contacted by local name server that can not resolve name

    - contact authoritative name server if name mapping not known
    - gets mapping
    - returns mapping to local name server

- **top-level domain (TLD) servers**:

- **authoritative DNS servers**:

    - organization’s own DNS server(s), providing authoritative hostname to IP mappings for organization’s named hosts

- **Local DNS name server**:

    - each ISP has one
    - DNS query is sent to its local DNS server
    - **iterative query** VS **recursive query**
    - **Caching**

Multimedia networking

3 application types

- **streaming, stored** audio, video

    - **streaming**: can begin playout before downloading entire file
    - stored (at server): can transmit faster than audio/video will be rendered => buffer.

- **conversational** voice/video over IP

  - interactive nature of human-to-human conversation limits delay tolerance

- **streaming live** audio, video

streaming stored video

continuous playout constraint: once client playout begins, playback must match original timing

client-side buffering and playout delay: compensate for network-added delay, delay jitter

initial playout delay tradeoff: buffer starvation less likely with larger delay, but larger delay until user begins watching

- UDP Streaming
  - server sends at rate appropriate for client (constant rate)
  - short playout delay to remove network jitter
  - Drawback:
    - Due to unpredictable and varying amount of bandwith, fail to provie continuous playout
    - Requires a media control server to process client-to-server interactivity request and track client state => overhead
    - UDP may not go throught filewall

- HTTP Streaming
  - file retrieved via HTTP GET
  - send at maximunm possible rate under TCP
  - fill rate fluctuates due to TCP congestion control, retransmissions
  - larger playout delay: smooth TCP delivery
  - HTTP/TCP passes more easily through firewalls

- Video Streaming and CDNs
  - Challenge: **scale**, **heterogeneity**
  - Solution: **distributed**, **application-level infrastructure**: **content distribution network (CND)**
  - store/serve multiple copies of videos at multiple geographically distributed sites (CDN)
    - enter deep: push CDN servers deep into many access networks
    - bring home: smaller number of larger clusters in POPs near access networks

  - how does CDN select “good” CDN node to stream to client
    - geographically closet
    - shortest delay

        - let client decide

- **DASH: Dynamic, Adaptive Streaming over HTTP**

    - client determines
        - **when** to request chunk
        - **what** encoding rate to request
        - **where** to request chunk

Real-Time Protocol (RTP)

- RTP libraries provide transport-layer interface that extends **UDP**:

    - port numbers, IP addresses
    - payload type identification
    - sequence number
    - time-stamp

- RTP does not provide any mechanism to ensure timely data delivery or other guarantees
- RTP encapsulation only seen at end systems

payload type	seq number	time stamp	Synchronization Source ID	Miscellaneous fileds

- **payload type (7 bits)**: indicates type of encoding currently being used.
- **sequence # (16 bits)**: increment by one for each RTP packet sent
- **timestamp field (32 bits long)**: sampling instant of first byte in this RTP data packet

    - for audio, timestamp clock increments by one for each sampling period

- **SSRC field (32 bits long)**: identifies source of RTP stream. Each stream in RTP session has distinct SSRC

Real-Time Control Protocol (RTCP)

- works in conjunction with RTP
- each participant in RTP session periodically sends RTCP control packets to all other participants
- each RTCP packet contains sender and/or receiver reports
- feedback used to control performance

Voice-over-IP (VoIP)

- **VoIP end-end-delay requirement**: needed to maintain “conversational” aspect
  - higher delays noticeable, impair interactivity
  - < 150 msec: good
  - > 400 msec bad
- **session initialization**
- **value-added services: call forwarding, screening, recording**
- **Packet loss, delay**
  - **network loss**: IP datagram lost
  - **delay loss**: arrives too late
  - **loss tolerance**: between 1% and 10% can be tolerated
- sender generates packets every 20 msec during talk spurt.
- receiver attempts to playout each chunk exactly q msecs after chunk was generated
- **tradeoff** in choosing q:
  - **large q**: less packet loss
  - **small q**: better interactive experience

Adaptive playout delay

goal: low playout delay, low late-loss rate

approach: adaptive playout delay adjustment:

adaptively estimate packet delay: di = (1 - a)d_(i-1) + a(ri - ti)

estimate average deviation of delay: vi = (1-b)vi-1 + b lri – ti – dil

playout-timei = ti + di + Kvi

Determine whether packet: difference of successive stamps > 20 msec and sequence numbers without gaps => talk spurt begins

VoiP: recovery from packet loss

Forward Error Correction (FEC): send enough bits to allow recovery without retransmission

- **Simple FEC**: for every group of n chunks, create redundant chunk by exclusive OR-ing n original chunks
  - increasing bandwidth by factor 1/n
  - can reconstruct original n chunks if at most one lost chunk from n+1 chunks
- **piggyback lower quality stream**: send lower resolution audio stream as redundant information
  - non-consecutive loss: receiver can conceal loss
- **interleaving to conceal loss**: audio chunks divided into smaller units, packet contains small units from different chunks. if packet lost, still have most of every original chunk
  - no redundancy overhead, but increases playout delay.

The Network Layer

transport segment from sending to receiving host

network layer protocols in **every** host, router

- **Forwarding**: (router local action) packet arrives at router's input link, router must move the packet to the appropriate link.
- **Routing**: (network-wide process) Determine the route or path taken by packets as they flow from sender to a receiver.

Every router has a **forwarding table**

A router forwards a packet by examining the value of a field in the arriving packet’s header, and then using this header value to index into the router’s forwarding table

the routing algorithm determines the values that are inserted into the routers’ forwarding tables

- **Data plane**:
  - local, per-router function
  - determines how arriving packet is forwarded
- **Control plane**
  - network-wide logic
  - determines how datagram routed among routers from sender to receiver

  - 2 approaches:
    - **traditional routing algorithms**: implemented in router
    - **software-defined networking (SDN)**: implemented in (remote) server

Inside a Router:

- **Input ports**:

  - **line termination**: (physical layer) bit-level reception
  - **link layer protocol (receive)**: data link layer
  - **lookup, forwarding** (queue):

    - use header values to lookup output port using forwarding table
    - queuing: if datagrams arrive faster than forwarding rate into switch fabric
    - **Longest prefix matching**: when looking for forwarding table entry for given destination address, use longest address prefix that matches destination address.

- **Switching fabrics**:

  - transfer packet from input buffer to appropriate output buffer
  - switching rate: rate at which packets can be transfer from inputs to outputs
  - three types of switching fabrics:
    1. Memory: (traditional), limited by memory bandwidth
    2. bus: shared buss, **bus contention**
    3. crossbar: up to Tbits/sec combined capacity

- **Output ports**:

  - **buffering** required from fabric faster rate
  - scheduling datagrams (Priority scheduling)
  - How much buffering: **RTT*C/sqrt(N)**, N flows, link capacity C

Scheduling mechanisms

- discard policy
  - tail drop
  - priority
  - random

- Scheduling:
  - **priority scheduling**: send highest priority queued packet

    - **Round Robin scheduling**
    - **Weighted Fai Queuing (WFQ)**

IP datagram

- 20 bytes header
- Version number: 4 bits specify the IP protocol version of the datagram
- Header length: due to variable header length
- Datagram length
- 16-bit id, flags, fragment offset: used for fragmentation and reassembly
- Time to live: max number remaining hops
- upper layer protocol: TCP or UDP
- 32 bit src IP address
- 32 bit dest IP addr
- options (if any)
- data (typically TCP or UDP segment)

network links have MTU (maximun transfer size) - largest possible link-level frame

=> large IP datagram divided in to small datagrams

- flag: 0 for last fragment, 1 for rest
- ID: same for all
- offset: the offset of first byte in total data

IP Addressing

32-bit identifier for host, router interface,

IP addresses for each interface

Subnet

Subnet: each isolated network after detaching each interface

subnet address: high order of IP address

host part: low order bits of IP address

local subnet: attached to local interface, no routing needed.

remote subnet: reachable via some gateway, internal structure of subnet unknown.

subnet matching: longest prefix matching

Classless Interdomain Routing (CIDR)

CIDR generalizes the notion of subnet addressing

a.b.c.d/x: x is number of bits in subnet portion of address.

The Link Layer

node: device thatr runs a link-layer protocol, including hosts, routers, siwtches and Wifi access points.

links: communications channels that connect adjacent nodes along the communication path.

link-layer frame: the transmitting node encapsulates the datagram in a link-layer frame.

Obtain a IP address

- Hard-coded by system admin
- **DHCP: Dynamic Host Configuration Protocol**:

  - temporary IP address that will be different each time the host connects to the network
  - plug-and-play
  - Steps:
    1. host broadcasts “DHCP discover” msg
    2. DHCP server responds with “DHCP offer” msg
    3. host requests IP address: “DHCP request” msg
    4. DHCP server sends address: “DHCP ack” msg

ICMP: Internet Control Message Protocol

- used by hosts & routers to communicate network- level information
  - error reporting
  - echo request/reply

- architecturelly "above" IP
- ICMP message: type, code plus first 8 bytes of IP datagram causing error
- **Traceroute and ICMP**:

    - set TTL = 1 ... n and a unlikely port number
    - When datagram in n-th arrives to n-th router, router discards due to TTL = 0, and send ICMP message with orouter and IP address
    - When last reach host, due to a unlikely port, return a ICMP "port unreachable" message

NAT: Network Address Translation

- one public IP address for network of devices
- can change ISP without changing internal addresses
- devices inside local net not "visible" by outside world
- all datagram leaving local network have same single source NAT IP address
- NAT router use port for different hosts within the network.
    - Outgoing datagrams: (source IP address, port #) to (NAT IP address, new port #)
    - remember (in NAT translation table) every (source IP address, port #) to (NAT IP address, new port #) translation pair
    - incoming datagrams: replace (NAT IP address, new port #) with (source IP address, port #) stored in NAT table

- NAT is **controversial**:

    - routers should only process up to layer 3
    - address shortage should be solved by IPv6
    - violates end-to-end argument

IPv6

- **Motivation**:

    - 32-bit address shortage
    - header formats (fixed length) helps speed speed processing / forwarding
    - header changes to facilitate label-switching

- **IPv6 datagram format**:

    - fixed-length 40 byte header
        - priority: = type of service in IPv4
        - Flow label: this 20-bit field is used to identify a flow of datagrams.
        - payload length
        - next header: upper layer protocol for data
        - hop limit: = time to live (TTL) in IPv4
        - source address (128 bytes)

        - destination address (128 bytes)
    - no fragmentation allowed

- **Tunnelling**: IPv6 datagram carried as payload in IPv4 datagram among IPv4 routers

Virtual Circuit Forwarding

- A virtual circuit (VC) is a means of transporting data over a packet switched computer network in such a way that it appears as though there is a dedicated physical layer link between the source and destination end systems of this data
- "Connection-oriented" network
- signalling establishes a connection along path
- forwarding based on flat labels
- **routers maintain connection state information**

Multi-Protocol Label Switching (MPLS)

- improve the forwarding speed of IP routers by adopting a key concept from the world of virtual-circuit networks: a fixed-length label
- selectively labeling datagrams and allowing routers to forward datagrams based on fixed-length labels
- fast lookup using fixed length identifier (rather than shortest prefix matching)
- MPLS-enhanced frame can only be sent between routers that are both MPLS capable
- An MPLS-capable router is often referred to as a **label-switched router**
- MPLS-capable router need not extract the destination IP address and perform a lookup of the longest prefix match in the forwarding table
- **flexibility**: MPLS forwarding decisions can differ from those of IP
    - use destination and source addresses to route flows to same destination differently
    - re-route flows quickly if link fails: pre-computed backup paths
- **IP routing**: path to destination determined by destination address alone
- **MPLS routing**: path to destination can be based on source and destination address

Generalized Forwarding and SDN

- Each router contains a **flow table** that is computed and distributed by a **logically centralized routing controller**
    - Local flow table: headers, counters, actions
- simple packet-handling rules:

    - Pattern: match values in packet header fields
    - Actions: for matched packet: drop, forward, modify, send
    - Priority: disambiguate overlapping patterns
    - Counters: #bytes and #packets

Routing

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

Dijkstra’s algorithm (Link State)

- net topology, link costs known to all nodes
- computes least cost paths from one node to all other nodes
    - gives forwarding table for that node

```
Initialization:
   N' = {u}
   for all nodes v
       if v adjacent to u
           then D(v) = c(u, v)
       else D(v) = inf

Loop
   find w not in N' such D(w) is minimal
   add w to N'
   update D(v) = min(D(v), D(w) + c(w, v))
until all nodes in N'
```

Distance Vector algorithm

Bellman-Ford equation (dynamic programming)

```
let
    dx(y) = cost of least-cost path from x to y
then
    dx(y) = min{c(x,v) + dv(y)}
```

- Dx(y) = estimate of least cost from x to y

- node x:
  - knows cost to each neighbor v: c(x,v)
  - maintains its neighbors’ distance vectors. For each neighbor v, x maintains Dv = [Dv(y): y in N]

Key idea:

- each node sends its own DV estimate to neighbors
- when x receives new DV estimate, it updates its own DV
  - Dx(y) ← minv{c(x,v) + Dv(y)} for each node y ∊ N

- **iterative, asynchronous**
- **distributed**

```
Initialization:
    for all destinations y in N:
        Dx(y) = c(x,y) /* if y is not a neighbor then c(x,y) = ∞ */
     for each neighbor w
         Dw(y) = ? for all destinations y in N
      for each neighbor w
          send distance vector Dx = [Dx(y): y in N] to w
loop
    wait (until I see a link cost change to some neighbor w or
    until I receive a distance vector from some neighbor w)
        for each y in N:
            Dx(y) = minv{c(x,v) + Dv(y)}
         if Dx(y) changed for any destination y
            send distance vector Dx = [Dx(y): y in N] to all neighbors
forever
```

- link cost changes:
  - node detects local link cost change
  - **bad news travels slow**

- **poisoned reverse**

  - Z tells Y its (Z’s) distance to X is infinite (so Y won’t route to X via Z)

LS VS DV

- message complexity
  - LS: O(nE)
  - DV: convergence time varies

- speed of convergence
    - LS: O(n^2) algorithm requires O(nE) messages
    - DV: convergence time varies
        - may be routing loops
        - count-to-infinity problem

- robustness
    - LS
        - node can advertise incorrect link cost
        - each node computes only its own table

    - DV
        - DV node can advertise incorrect path cost
        - each node’s table used by others => error propagation through network

Scalability

- With billions of destinations
- => can't store all destinations in routing table
- => routing table exchange would swamp links

"autonomous systems" (AS)

a.k.a. "domains

an autonomous system (AS) is a collection of connected Internet Protocol (IP) routing prefixes under the control of one or more network operators on behalf of a single administrative entity or domain that presents a common.

- **Intra-AS routing**:

    - routing among hosts, routers in same AS (“network”)
    - all routers in AS must run same intra-domain protocol
    - routers in different AS can run different intra-domain routing protocol
    - gateway router: at “edge” of its own AS, has link(s) to router(s) in other AS’es

- **inter-AS routing**：

    - routing among AS’es
    - gateways perform inter- domain routing (as well as intra-domain routing)

- forwarding table configured by both intra- and inter-AS routing algorithm
  - intra-AS routing determine entries for destinations within AS
  - inter-AS & intra-AS determine entries for external destinations

OSPF (Open Shortest Path First)

- OSPF routing is widely used for intra-AS routing in the Internet.
- OSPF indicates that the routing protocol specification is **publicly available**
- OSPF is a **link-state protocol** that uses flooding of link-state information and a **Dijkstra** least-cost path algorithm
- router floods OSPF link-state advertisements to all other routers in entire AS

  - carried in OSPF messages directly over IP

1. A router constructs a complete topological graph of the entire autonomous system
2. Then locally runs Dijkstra algorithm to determine a shortest-path tree to all **subnets** (Individual link costs are configured by the network administrator, may set to 1)

- With OSPF, a router broadcasts routing information to **all** other routers in the autonomous system
- A router broadcasts link-state information whenever there is a change in a link’s state (eg, cost or up/down status)
- broadcasts a link’s state periodically even if the link’s state has not changed
- The OSPF protocol also checks that links are operational and allows an OSPF router to obtain a neighboring router’s database of network-wide link state
- **Security**: Exchanges between OSPF routers can be authenticated
- **Multiple same-cost paths**: When multiple paths to a destination have the same cost, OSPF allows multiple paths to be used
- **Integrated support for unicast and multicast routing**: Multicast OSPF provides simple extensions to OSPF to provide for multicast routing
- **Support for hierarchy within a single routing domain**

Hierarchical OSPF

- **two-level hierarchy**: local area, backbone

  - link-state advertisements only in area
  - each node has detailed area topology

- **area border routers**: summarize distances to nets in own area, advertise to other Area Border routers
- **backbone routers**: run OSPF routing limited to backbone

- **boundary routers**: connect to other ASes

In-network duplication

- **flooding**: when nodereceives broadcast packet, sends copy to all neighbors
    - *problem*: cycles & broadcast sotrm
- **controlled flooding**: node only broadcasts packet, if it hasn't broadcast same packet before
    - node keep track of pack ids already broadcasted
- **reverse path forwarding**: only forward packet if it arrived on shortest path to source
    - requires unicast routing
- **spanning tree**:
    - no redudant packets received by any node

1. First construct a spanning tree
2. nodes then forward/make copies only along panning tree

BGP (Border Gateway Protocol)

**BGP (Border Gateway Protocol): the de facto inter-domain routing protocol

** Used for determining paths of source-destination pairs that span multiple ASs.

- BGP provides each AS a means to:
    - **eBGP**: obtain subnet reachability information from neighboring ASes
    - **iBGP**: propagate reachability information to allAS-internal routers
    - determine "good" routes to other networks based on reachability information and policy

- advertised prefix includes BGP attributes
- two important attributes:
    - **AS-PATH**: list of ASes through which prefix advertisement has passed
    - **NEXT-HOP**: indicates specific internal-AS router to next- hop AS

- **Policy-based routing**:
    - gateway receiving route advertisement uses import policy to accept/decline path
    - AS policy also determines whether to advertise path to other neighboring ASes

- gateway router may learn about multiple paths to destination

In BGP, pairs of routers exchange routing information over semipermanent TCP connections

There are also semipermanent BGP TCP connections between routers within an AS, two routers at the end of the connection are called **BGP peers**, connection is called a **BGP session**

Gateway routers run both eBGP and iBGP protocols.

- policy:
  - inter-AS: admin wants control over how its traffic routed, who routes through its net
  - intra-AS: single admin, so no policy decisions needed

- scale:
  - hierarchical routing saves table size, reduced update traffic

- performance:
  - intra-AS: can focus on performance
  - inter-AS: policy may dominate over performance

The SDN control plane

distinct (typically remote) controller interacts with local control agents (CAs) in routers to compute forwarding tables

- **Data plane switches**:

  - fast, simple, commodity switches implementing generalized data-plane forwarding in hardware
  - switch flow table computed, installed by controller
  - API for table-based switch control
  - protocol for communicating with controller

- **SDN controller (network OS)**:

  - maintain network state information
  - interacts with network control applications “above” via northbound API
  - interacts with network switches “below” via southbound API
  - implemented as distributed system for performance, scalability, fault-tolerance, robustness

- **network-control apps**:

  - “brains” of control: implement control functions using lower-level services,API provided by SND

controller
    - unbundled: can be provided by 3rd party – distinct from routing vendor, or SDN controller

The Link Layer

Link Layer Services

Framing: encapsulate each network-layer data-gram within a link-layer frame. A frame consists of a data filed and a numebr of header fields.

Link access: A medium access control (MAC) protocol specifies the rules by which a frame is transmitted onto the link.

Reliable delivery: guarantees to move each network-layer datagram across the link without error. Many wired link-layer protocols do not provide a reliable delivery service (due to low bit-error).

Error detection and correction: Bit errors are introduced by signal attenuation and electro noise. Error detection in the link layer is usually more sophisticated and is implemented in hardware.

The link layer is implemented in a **network adapter**, known as **network interface card (NIC)**. At the heart of the network adapter is the link-layer controller (a single chip) implements many of the link-layer services.

Sending side: controller takes a datagram from higher layer, encapsulates the datagram in a link-layer frame and transmits the frame into the communication link, following the link-access protocol. (May set error-detection bits in frame header)

Receiving side: controller receives the frame, extracts the network-layer datagram. (If error-detection bits are set) perform error detection.

The link layer is combination of hardware and software.

Error-Detection and Error-Correction

Goal: develop and intuitive feel for the capabilities that error-detection and correction techniques provide.

Parity Checks: use a single **parity bit**. In even(odd) parity scheme. The sender simply includes one additional bit and chooses its value such that the total number of 1s in the d+1 bits is even(odd). The probability of

undetected errors using single-bit parity can approach 50%.

Two-dimensional parity: the parity of both the collumn and the row containing the flipped bit will be in error. The receiver can not only detect single-bit error has occurred but also correct that error.

Checksumming Methods: bytes of data are treated as 16-bit integers and summed. The 1s complement of this sum forms the Internet checksum that carried in the segment header.

Cyclic Redundancy Check (CRC)

Hamming Distance: n-bit errors can be detected if H(C) > n; n-bit errors can be corrected if H(C) > 2n.

Framing

split bit stream in discrete units

signal start and end of message

Frame Delimiter: designate special bit pattern as delimiter. Double the delimiter to escape to represent a data.

Multiple Access Links and Protocols

point-to-point link: consists of a single sender at one end of the link and a single receiver at the other end of the link.

broadcast link: have multiple sending and receiving nodes all connected to the same, single, sharedbroadcast channel. When any one node transmits a frame, the channel broadcasts the frame and each of the other nodes receives a copy.

Multiple access problem: how to coordinate the access of multiple sending and receiving nodes to a shared broadcast channel.

Multiple Access Protocols: nodes regulate their transmission into the shared broadcast channel. (need in a wide variety of network settings, including bot wired and wireless access networks)

When there is a collision, none of the receiving nodes can make sense of any the frames that were transmitted. All the frames involved in the collision are lost.

To ensure broadcast channel perform useful work => coordinate the transmissions of the active nodes. (responsibility of the multiple access protocol)

Multiple access protocols belonging to one of **three categories**:

- **Channel partitioning protocols**
- **Random access protocols**
- **Take-turns protocols**

Ideal protocol:

- When only on nodes, throughput R bps.
- When M nodes, throughput R/M bps.
- Decentralized
- Simple, inexpensive to implement

Channel Partitioning Protocols

Time-Division Multiplexing (TDM) divides time into time frames and further divieds each time frame into N time slots. Each time slot is assigned to one of the N nodes.

When a node has a packet to send, it transmits the packet's bits during its assigned time slot.

TDM eliminates collisions and is perfectly fair but with two drawbacks

1. A node is limited to an average rate of R/N bps even when it is the only node with packets to send.
2. A node must always wait for its turn in the transmission sequence even it is the only active node.

Frequency-Division Multiplexing (FDM) divides the R bps channel into different frequency and assignes each frequency to one of the N nodes. **FDM shares both the advantages and drawbacks of TDM**.

Code division multiple access (CDMA) assignes a different **code** to each node, each node then uses its unique code to encode the data bits it send.

Random Access Protocols

A transmitting node always transmits at the full rate of the channel (R bps). When there is a collision, each node involved in the collision repeatedly retransmits its frame until its frame gets through without a collision.

Each node will choose a independent random delays before retransmitting the frame.

Slotted ALOHA

- All frames exactly L bits

- Time is divided into slots of L/R seconds
- Nodes start to transmit frames only at beginnings of a slot.
- The node are synched so that each node knows when the slots starts
- If two or more frames collide in a slot, then all nodes detect the collision before event ends.

Slotted ALOHA allows a node to transmit continuously at full rate R, when node is the only active node.

Highly **decentralized**, each node detects collisions and independently decides when to retransmit.

Extremely simple

Maximun efficiency slotted ALOHA:

- each node attempts to transmit a frame in each slot with probability p.
- N nodes
- probability a given node has a success is p(1 - p)^(N-1)
- probability that an one of the N nodes has a success is Np(1 - p)^(N-1)
- Maximum efficiency of the protocl is given by 1/e = 0.37

=> When a large number of nodes have many frames to transmit, then (at best) only 37 percent of the slots do useful work

Pure (unslotted) ALOHA

- When a frame first arrives, the node immediately transmits the frame in its entirety into the broadcast channel.
- If a transmitted frame experiencs a collision whth one or more other transmissions, the node will then immediately retransmit with probability p. Otherwise, the node waits for a frame transmission time.
- After this wait, it then transmit with probability p, or waits for another frame time with probability 1 - p.

Maximun efficiency of pure ALOHA

- At any given time ,the probability that a node is transmitting a frame is p.
- Suppose frame begins transmission at time t0.
- In order to successful, no other nodes can begin transmission in the interval of time [t0 - 1, t0]. The probability that all other nodes do not begin a transmission in this interval is (1 - p)^(N-1)
- The probability that a given node has a successful transmission is p(1 - p)^(2(N-1))

=> The maximun effeciency of the pure ALOHA is 1/(2e)

Carrier Sense Multiple Access (CSMA)

- **Carrier sensing**: a node listens to the channel before transmitting. If other node is currently transmiting into the channel, it then waits until it detects no transmissions.
- **Collision detection**: a transmitting node listens to the channel. If it detects that another node is transmitting an interfering frame, it stops transmitting and waits a random amount of time before repeating.

It is evident that the end-to-end **channel propagation delay** of a broadcast channel will play a crucial role in determining its performance.

=> The longer the propagation delay, the longer the chanse of collision.

Carrier Sense Multiple Access / Collision Detection (CSMA/CD)

1. Datagram arrived from network layer, prepares a link-layer frame and puts into adapter buffer
2. If adapter senses that channel is idle, it starts to transmit the frame. Otherwise, waits until it senses not signal energy.
3. When transmitting, the adapter monitors for the presense of signal energy comming from other adapters using the broadcast channel.
4. If the adapter done transmission without detecting signal energy from other adapter, it finishes with the frame. Otherwise, it aborts the transmission.
5. After aborting, the adapter waits a random amount of time and then returns to step2.

Binary Exponential Backoff: When transmitting a frame that has already experienced n collisions, node chooses the value K at random from {0, 1, 2, ... 2^n - 1}

CDMA/CD Efficiency

- When only one node active, it transmit at full channel rate.
- Dprop denote the maximum time it takes signal energy to propagate between any two adapters
- Dtrans be the time to transmit a maximun-size fram.
- Efficiency = 1 / (1 + 5Dprop/Dtrans)

Taking-Turns Protocols

R bps if one active, R/M bps if M nodes active

Pooling Protocol

One of the nodes is master node. The master node polls each of the nodes in a round-robin fashion.

Eliminates the collisions and empty slots that plague random access protocols => achieve a much higher efficiency.

Has a fiew drawbacks:

- Introduces a polling delay: tiem required to notify a node that it can transmit.
- Single point of failure (master node)

Token-Passing Protocol

- A **token** is exchanged among the nodes in some fixed order.
- When a node receives a token, it holds token only if it has frames to transmit. Otherwise, it immediately forwards the token to the next node.
- If node has frames to transmit, it sends up to a maximum number of rames and then forwards the token to the next node.

Decentralized and **highly efficient**.

Problem: **token overhead**, **latency**, **single point of failure (token)**

Cable Internet Access

A cable access network typically connects several thousand residential cable modems to a cable modem termination system (CMTS) at the cable network headend.

Data-Over-Cable Service Interface Specifications (DOCSIS) uses FDM to devide downstream (CMTS->modem) and upstream (modem->CMTS) network segments into multiple frequency channels.

Each upstream and downstream channel is a broadcast channel. Frames transmitted on the downstream channel by the CMTS are received by all cable modems receiving that channel.

For upstream, multiple cable modems share the same upstream channel to the CMTS, collisions can potentially occur. Each upstream channel is divided into intervals of time (TDM-like), each containing a sequence of mini-slots during which cable modems can transmit to the CMTS. CMTS explicitly grants permission to individual cable modems to transmit durint specific mini-slots. **The CMTS accomplishes this by sending a control message (MAP msg) on a downstream channel to specify which cable modem can transmit durint which mini-slot**.

Cable modems send **mini-slot-request** frames to delicated for notifying CTMS it has frames to send. The mini-slot-request frames are sent in random access manner so collision may occur. Detect collision by no reply. When collision is inferred, use binary exponential backoff ot defer retransmission.

MAC Addresses

Switches: Operate at the link layer, they switch link-layer frames, do not recognize network-layer addresses. Don't use routing algorithms to determine paths.

Adapters (Network interfaces) have **link-layer addresses**. Host or router with **multiple network interfaces** will have **multiple link-layer addresses**.

Link-layer switches do not have link-layer addresses associated with their interfaces. Because the job of **link-layer switch** is to carry datagrams between hosts and routers.

A link-layer address is called **LAN address**, or **physical address** or a **MAC address**.

Macaddress is **6 bytes** long. eg. 1A-23-F9-CD-06-9B.

Adapter's MAC address is **fixed** and **unique**. (IEEE manages the MAC address space)

An adapter's MAC address has a **flat structure**, does not change not mater where the adapter goes. While IP addresses have a **hierarchical structure**, and a host's IP address changed when the host moves.

- When an adapter wants to send a frame to dest adapter, the sender insert the dest adapter's MAC address into the frame and sends into the LAN.
- When an adapter receives a frame, it check to see if the dest MAC address matches its own MAC address.
    - If matches: adapter extracts the datagram and pass up the protocol stack
    - If not match, discards the frame.

- Special MAC **broadcast address** with 48 consecutive 1s (FF-FF-FF-FF-FF-FF). All receiver adapter will process the frame.

Address Resolution Protocol (ARP)

Address Resolution Protocol (ARP): translate between IP address (network-layer) and MAC addresses (link-layer).

ARP module in the sending host takes any IP address on the same LAN as input, and returns the MAC address. (**ARP resolves IP addresses only for interfaces on the same subnet**)

Each node has an **ARP table** in its memory, contains mappings of IP addresses to MAC addresses and **time-to-live (TTL)**

Sending host obtain MAC address for givin IP addresses:

1. If mapping does not exists in node's ARP table. Sender constructs a **ARP packet** with sending and

receiving IP and MAC address. (Both query and response have same format)

2. Send the packet to MAC **broadcast address **(FF-FF-FF-FF-FF-FF). The adapter encapsulates the ARP packet in a link-layer frame, uses the broadcast address for the frame's destination address and transmits the frame into the subnet.
3. Frame received by all other adapters and adapter passes it to ARP module. ARP module checks to see if IP address matches the dest IP in ARP packet. The matching one sends back to the querying host a **response ARP packet** with desired mapping.
4. Sending host received response ARP packet and update ARP table.

plug-and-play: ARP tables built automatically without intervention from net/system administrator. When host disconnected from the subnet, its entry is eventually deleted.

Sending a Datagram off the subnet:

1. Sending host passes datagram to its adapter with MAC address of the connecting router interface for the receiver's subnet. And send the datagram to local subnet.
2. The router receives the datagram and determine the correct interface to forward, by consulting the **forward table**.

Ethernet

widely used LAN technology, simpler and cheaper than others, kept up with speed race 10Mbps-10Gbps.

Bus Topology: broadcast LAN: all transmitted frames travel to and are processed by all adapters connected to the buss

Star-based Topology: Hosts (and routers) are directly connected to a hub (**later replaced with a switch**) with copper wire. A **hub** is a physical-layer device acts on bits. When a bit arrives, it transmits the bit onto all other interfaces.

Preamble	dest addr	src addr	Type	Data	CRC

preamble: 7 bytes with pattern 10101010 followed by one byte with pattern 10101011, used to **synchronize receiver, sender clock rates**

addresses: 6 byte source, destination MAC address. If receiver match dest address or broadcast address, it passes data to network layer. Otherwise, discard frame

type: indicates higher layer protocol (mostly IP) **CRC**: cyclic redundancy check at receiver. If error detected, frame is dropped.

connectionless: no handeshaking.

unreliable: receiving NIC doesn't send acks. Frame is droped if error detected.

- Ethernet comes in **many** different flavors, such as 10BASE-T, 10BASE-2, 100BASE-T, 1000BASE-LX and 10GBASE-T.
- The first part refers to the speed of the standard.
- "BASE" refers to baseband Ethernet,(physical media only carries Ethernet traffic)
- The final part refers to the physical media itself (eg. T for twisted-pair copper wires)

Link-Layer Switches

Receive incoming link-layer frames and forward them onto outgoint links

Switch is **tranparent** to hosts and routers in the subnet: They unware that a switch will be receiving the frame and forwarding it.

plug-and-play and **self-learning**

- **Filtering**: determines whether a frame should be forwared to some interface or should just be dropped.
- **Forwarding**: determines the interfaces to which a frame should be directed and moves the frame to those interfaces.
- Filtering and forwarding are done with **switch table**. Swtich table contains entries for some but not necessarily all of the hosts and routers on a LAN.
- An entry in the switch consists of
  - a MAC address
  - the switch interface leads to taht MAC address
  - the time when the entry was placed in table

- Switches forward packets based on MAC addresses rather than on IP addresses.
- When a frame arrives at the switch on interface x.
  1. If there is no entry for the destination address, the switch broadcasts the frame.
  2. If there is an entry for destination table and have a interface x. Discard the frame. (filtering)
  3. There is an entry in the table, with interface y!=x. The frame need to be forwared to interface y.

- **Self-Learning**: table is built automatically, dynamically and autonomously

  1. The switch table is initially empty.
  2. For each incoming frame received on an interface, the switch stores in its table
     - The MAC address in the frame's source address field
     - The interface from which the frame arrived

        - Current time

3. Switch deletes an address in the table if no frames are received with that address as the source address after some period of time (**aging time**).

- Properties of link-layer switching
    - **Elimination of collisions**: buffer used, no wasted bandwidth due to collision.
    - **Heterogeneous links**: a switch isolates one link from another.
    - **Management**: enhanced security, eases network management.

Virtual Local Area Networks (VLANs)

- Three drawbacks of switched LAN in read world:
    1. Lack of traffic isolation
    2. Inefficient use of switches
    3. Managing users

- Theses difficulties can be handled by switch that supports **virtual local area networks (VLANs)**

    1. **Traffic Isolation**: frames to/from ports 1-x can only reach ports 1-x
    2. **Dynamic membership**: ports can be dynamically assigned among VLANs
    3. **Forwarding between VLANs**:

Trunk Port: A more scalable approach to interconnecting VLAN switches. A special port on each switch is configured as a trunk port to inter connect the two VLAN switches. The trunk port belongs to all VLANs, and frames sent to any VLAN are forwareded over the trunk link to the other switch. **VLAN tag** added into the header that carries the identity of the VLAN to which the frame belongs. It is added into a frame by the switch at the receiving side of the trunk.

Wireless and Mobile Networks

a network in which wireless (possibly mobile) users are connected into the larger network infrastructure by a wireless link at the network’s edge.

- **Wireless hosts**: hosts are the end-system devices that run applications.

    - May be stationary or mobile

- **Wireless links**: A host connects to a base station or to another wireless host through a wireless communication link.

    - Different wireless link have different transmission rates and distances.
    - also used as backbone link

- **Base station**: relay - responsible for sending and receiving data to and from a wireless host.
    - tipically connected to wired network
    - **Handoff**: a mobile host moves beyond the range of one base station and into the range of another one.

- **Network infrastructure**: The larger network with which a wireless host may wish to communicate
- **Infrastructure mode**: Base station connects mobile into wired network
- **ad hock networks**: No base stations. Nodes only transmit to other node, organize themselves into a network (route among themselves).

	single hop	**multiple hops**
intrastructure	host connects to base station which connects to larger Internet	host may relay through several wireless nodes to connect to larger Internet
no infrastructure	No base station, no connection to larger Internet (Bluetooth, ad hoc nets)	no base station, no connection to larger Internet, may have to relay to reach other wireless node

- **Decreasing signal strength**
- **Interference from other sources**
- **Multipath propagation**: occurs when portions of the electromagnetic wave reflect off objects and the ground, taking paths of different lengths between a sender and receiver.
- => bit errors will be more common in wireless links than in wired links
- **Signal-to-noise Ratio (SNR)**: relative measure of the strength of the received signal and this noise. (in decibels[dB])
- **Bit Error Rate (BER)**: probability that a transmitted bit is received in error at the receiver
- Given modulation scheme, the higher the SNR, the lower the BER.
    - increase transmission power to decrease BER. => tradeoff: more energy expended

- For a given SNR, a modulation technique with a higher bit transmission rate will have a higher BER.
- Dynamic selection of the physical-layer modulation technique can be used to adapt the modulation technique to channel conditions.

Challenges different from wired transmission:

- **Hidden Terminal Problem**: Physical obstructions in the enviroment my prevent two node hearing from each other's transmissions even though they are indeed interfering each other.

- **Signal attenuation**: Results in undetectable collisions at the receiver results from the **fading** of signal's strength as it propagates through the wireless medium.

Bluetooth

- < 10m diameter
- replacement for cables
- Ad hoc: no infrastructure
- master/slaves:
  - slave request to send, master grant request

Wifi: 802.11 Wireless LANs

IEEE 802.11 wireless LAN, also known as WiFi

There are several 802.11 standards for wireless LAN technology

The three (a,b,g) 802.11 standards share many characteristics:

- same medium access protocol, CSMA/CA.
- same frame structure for link-layer frames
- ability to reduce their transmission rate in order to reach out over grater distances
- allow for both "infrastructure mode" and "ad hoc mode".

802.11b:

- direct sequence spread spectrum in physical layer
  - modulate signal on high frequency "chip rate"
  - redundancy / robustness
  - single channel or CDMA

802.11 LAN architecture:

- wireless host communicates with base station.
- A **basic service set (BSS)** contains one or more wireless hosts and a cnentral base station, known as an **access point (AP)**.
- Each wireless hosts and AP has a MAC address.

Channels, association

- 802.11 divided into 11 channels at different frequencies.

    - AP admin chooses frequency for AP
    - interference possibe (same frequency with neighbour AP)

- **Service Set Identifier (SSID)** are assigned to AP by admin.
- Each wireless host needs to associate with an AP.
- Scans channels, listenning for beacon frames containing AP's name (SSID) and MAC address
- may perform authentication
- typically use DHCP to get IP address in AP's subnet.

- **Passive Scanning**:

    1. Beacon frames sent from APs
    2. Host send association request frame
    3. AP granted and send back association response frame

- **Active Scanning**:

    1. Probe request frame broadcast from host.
    2. Probe Response frames sent from AP
    3. Association Request frame sent to AP
    4. Association Response frame sent from AP to host

Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA)

- Difficult to receive (sense collisions) when transmitting due to weak received signals (fading)
- Can't sense all collisions in any case: hidden terminal or fading
- Uses a link-layer **acknowledgement/retransmission(ARQ)** scheme
- **Distributed Inter-frame Space (DIFS)**
- **Short Inter-frame Spacing (SIFS) **
- Sender:
    1. If sense channel idle for DIFS, transmit entire frame
    2. if sense channel busy, start random backoff time until channel idle and transmit.
    3. Wait for an acknowledgement.
    4. If an ack is received, the host knows that its frame has been correctly received. If ack not received, it reenteres the backoff phase in step2.

- Receiver:
    1. return ACK after SIFS.

Dealing with Hidden Terminals: RTS and CTS

- A (optional) reservation scheme that helps avoid collisions even in the presence of hidden terminals.

- Allow a host to use a short **Request to Send (RTS)** and **Clear to Send (CTS)** frame to **reserve** access to the channel.
- When a sender wants to send a DATA frame, it send an RTS to the AP, include total time required to transmit the DATA frame and the ACK frame.
- When AP receives the RTS frame, it responds a CTS frame, heard by all hosts.
- It gives the sender explicit permission ot send and also intruct the other host not to send for the reserved duration.

frame control(2)	duration(2)	address 1 (6)	address 2 (6)	address 3 (6)	seq control (2)	address 4 (6)	payload	CRC (4)

- **payload**: consists of an IP datagram or an ARP packet.
- **Address 1**: MAC address of wireless host or AP to receive this frame
- **Address 2**: MAC address of wireless host or AP transmitting this frame
- **Address 3**: MAC address of roouter interface to which AP (BSS) is attached.

Advanced capabilities

- **Rate adaption**: base station, mobile dynamically change transmission rate as mobile moves, SNR varies.
- **Power management**:
  - Mobile node to AP:
    - A node is able to explicitly alternate between sleep and wake states
    - AP knows not to transmit frames to this node
    - node wakes up before next beacon frame
  - beacon frame: contains list of mobiles with AP-to-mobile frames waiting to be sent

a day in the life of a web request

1. connecting laptop needs to get its own IP address, addr of first-hop router, addr of DNS server: use **DHCP**
2. DHCP request encapsulated in **UDP**, encapsulated in **IP**, encapsulated in **802.3 Ethernet**
3. Ethernet frame **broadcast** (dest: FFFFFFFFFFFF) on **LAN**, received at router running **DHCP server**
4. Ethernet demuxed to IP demuxed, UDP demuxed to DHCP
5. DHCP server formulates **DHCP ACK **containing client’s IP address, IP address of first-hop router for client, name & IP address of DNS server
6. encapsulation at **DHCP server**, **frame forwarded** (switch learning) through LAN, demultiplexing at client
7. DHCP client receives **DHCP ACK** reply

8. before sending **HTTP request**, need IP address of www.google.com: **DNS**
9. DNS query created, encapsulated in UDP, encapsulated in IP, encapsulated in Ethernet frame. To send frame to router, need **MAC address** of router interface: **ARP**
10. **ARP query broadcast**, received by router, which replies with **ARP reply** giving MAC address of router interface
11. client now knows MAC address of first hop router, so can now send frame containing **DNS query**
12. IP datagram containing DNS query forwarded via LAN switch from client to 1st hop router
13. IP datagram forwarded from campus network into Comcast network, routed (tables created by **RIP, OSPF, IS-IS and/or BGP routing protocols**) to **DNS server**
14. demuxed to DNS server
15. DNS server replies to client with **IP address** of www.google.com
16. to send **HTTP request**, client first opens **TCP socket** to web server
17. **TCP SYN segment** (step 1 in 3- way handshake) inter-domain routed to web server
18. web server responds with **TCP SYNACK** (step 2 in 3-way handshake)
19. TCP connection **established**
20. **HTTP request** sent into TCP socket
21. IP datagram containing HTTP request routed to www.google.com
22. web server responds with **HTTP reply** (containing web page)
23. IP datagram containing HTTP reply routed back to client

Mobility

- **home network**: permanent “home” of mobile
- **home agent**: entity that will perform mobility functions on behalf of mobile, when mobile is remote
- **permanent address**: address in home network, can always be used to reach mobile
- **care-of-address**: address in visited network
- **foreign agent**: entity in visited network that performs mobility functions on behalf of mobile

- **let routing handle it**: routers advertise permanent address of mobile-nodes-in-residence via usual routing table exchange. => **NOT SCALABLE**

- **let end-systems handle it**:

    - **indirect routing**: communication from correspondent to mobile goes through home agent, then forwarded to remote
    - **direct routing**: correspondent gets foreign address of mobile, sends directly to mobile

mobile uses two addresses:

- **permanent address**: used by correspondent

- **care-of-address**: used by home agent to forward datagrams to mobile

overcome triangle routing problem =>

non-transparent to correspondent: correspondent must get care-of-address from home agent

Mobile IP

- three components to standard:
  - indirect routing of datagrams
  - agent discovery
  - registration with home agent
- **agent advertisement**: foreign/home agents advertise service by broadcasting ICMP messages

cellular networks

- Cell:
  - covers geographical region
  - base station (BS) analogous to 802.11 AP
  - mobile users attach to network through BS
  - air-interface: physical and link layer protocol between mobile and BS
- **MSC (Mobile Switching Center)**
  - connects cells to wired tel. net.
  - manages call setup
  - handles mobility

home network: network of cellular provider you subscribe to

visited network: network in which mobile currently resides

Cloud Compution

- trends
  - mainframe => workstation => client/server computing
  - virtualization
  - scaling

Software Systems

- Docker: build once, run anywhere
- CloudFoundry
- Kubernetes
- OpenStack

Data Center Networks

- Load balancer: application-layer routing
  - receives external client requests
  - directs workload within data center
  - returns results to external client

- rich interconnection among switches, racks

Virtualization

- virtual machines
  - emulate hardware interface
  - run multiple kernels
  - clean isolation, difficult sharing

- container (Linux) / jail (FreeBSD) / etc.
  - emulate multiple kernel instances
  - private name spaces
  - less strict isolation, better sharing

Virtual Networking

- network-level addressing of instance
- traffic isolation:
  - pretection
  - resource control

- connectivity
  - connect services across instances
  - virtual subnets spanning physical machines

- **Tunnel**:
  - virtual interface(s) connected via tunnels
  - traffic isolation?
  - connectivity: complex management