BLM– Advanced Computer Networks

Slides are taken from
Computer Networks by Tanenbaum & Wetherall,

Outline

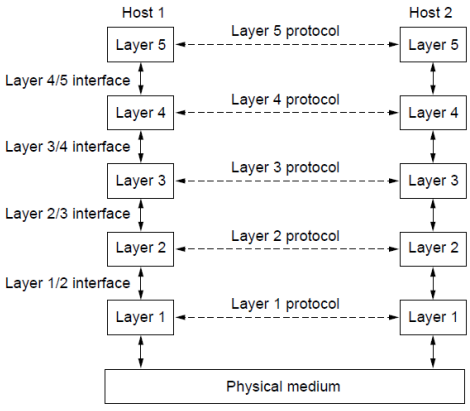
- Network software
- Reference models
- Physical layer
- Data link layer
- MAC layer
- Data link switching

Network Software

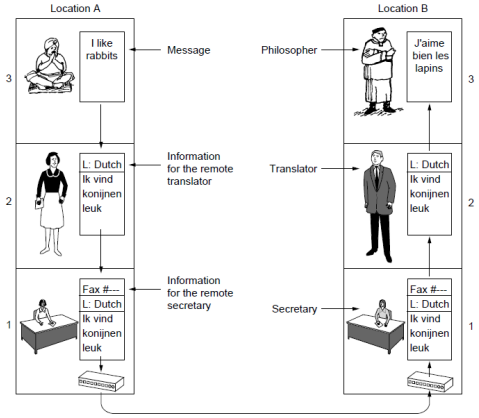
- Protocol layers
- Design issues for the layers
- Connection-oriented vs. connectionless service
- Service primitives
- Relationship of services to protocols

Protocol Layers – I

- Protocol layering is the main structuring method used to divide up network functionality.
  - Each protocol instance talks virtually to its peer
  - Each layer communicates only by using the one below
  - Lower layer services are accessed by an interface
  - At bottom, messages are carried by the medium



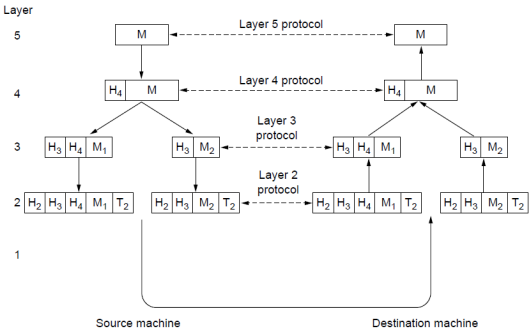
Protocol Layers – II



- Example: the philosopher-translator-secretary architecture
- Each protocol at different layers serves a different purpose

Protocol Layers – III

- Each lower layer adds its own <u>header</u> (with control information) to the message to transmit and removes it on receive
- Layers may also split and join messages, etc.



Design Issues for the Layers

- Each layer solves a particular problem but must include mechanisms to address a set of recurring design issues

Issue	Example mechanisms at different layers
Reliability despite failures	Codes for error detection/correction (§3.2, 3.3) Routing around failures (§5.2)
Network growth and evolution	Addressing (§5.6) and naming (§7.1) Protocol layering (§1.3)
Allocation of resources like bandwidth	Multiple access (§4.2) Congestion control (§5.3, 6.3)
Security against various threats	Confidentiality of messages (§8.2, 8.6) Authentication of communicating parties (§8.7)

Connection-Oriented vs. Connectionless

- Service provided by a layer may be kinds of either:
  - Connection-oriented, must be set up for ongoing use (and torn down after use), e.g., phone call
  - Connectionless, messages are handled separately, e.g., postal delivery

	Service	Example
Connection-oriented	Reliable message stream	Sequence of pages
	Reliable byte stream	Movie download
	Unreliable connection	Voice over IP
Connectionless	Unreliable datagram	Electronic junk mail
	Acknowledged datagram	Text messaging
	Request-reply	Database query

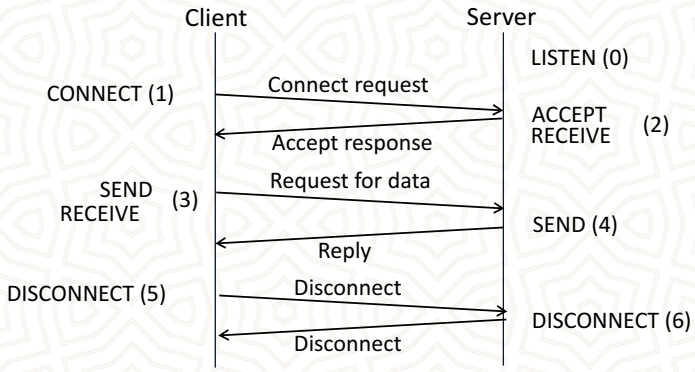
Service Primitives – I

- A service is provided to the layer above as primitives
- Hypothetical example of service primitives that may provide a reliable byte stream (connection-oriented) service:

Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
ACCEPT	Accept an incoming connection from a peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection

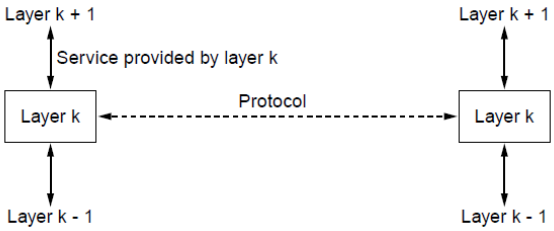
Service Primitives – II

- Hypothetical example of how these primitives may be used for a client-server interaction



Relationship of Services to Protocols

- Recap:
  - A layer provides a service to the one above [vertical]
  - A layer talks to its peer using a protocol [horizontal]



Reference Models

- Reference models describe the layers in a network architecture
  - OSI reference model »
  - TCP/IP reference model »
  - Critique of OSI and TCP/IP »

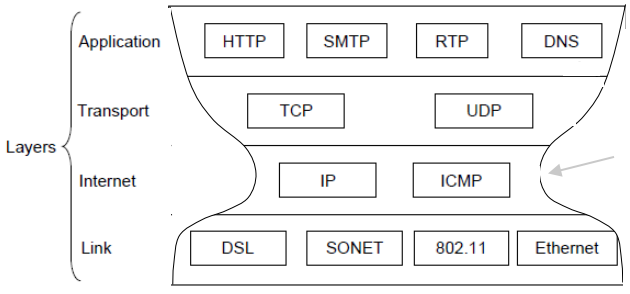
OSI Reference Model

- A principled, international standard, seven layer model to connect different systems

7	Application	– Provides functions needed by users
6	Presentation	– Converts different representations
5	Session	– Manages task dialogs
4	Transport	– Provides end-to-end delivery
3	Network	– Sends packets over multiple links
2	Data link	– Sends frames of information
1	Physical	– Sends bits as signals

TCP/IP Reference Model

- A four layer model derived from experimentation; omits some OSI layers and uses the IP as the network layer.



Protocols are shown in their respective layers

Critique of OSI & TCP/IP

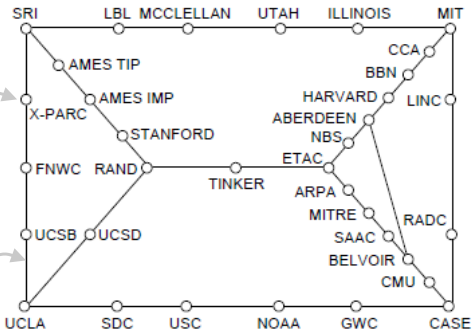
- OSI:
  - + Very influential model with clear concepts
  - Models, protocols and adoption all bogged down by politics and complexity
- TCP/IP:
  - + Very successful protocols that worked well and thrived
  - Weak model derived after the fact from protocols

Internet – I

- Before the Internet was the ARPANET, a decentralized, packet-switched network based on Baran's ideas.

Nodes are IMPs, or early routers, linked to hosts

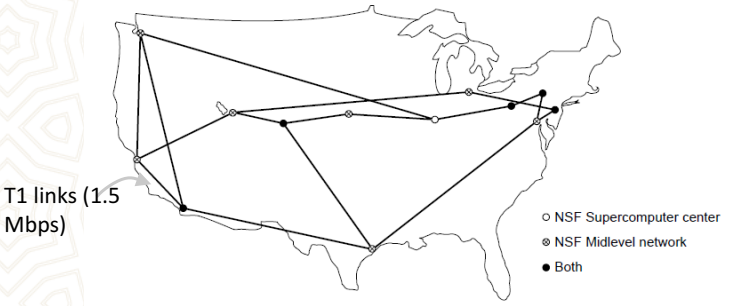
56 kbps links



ARPANET topology in Sept 1972.

Internet – II

- The early Internet used NSFNET (1985-1995) as its backbone; universities connected to get on the Internet

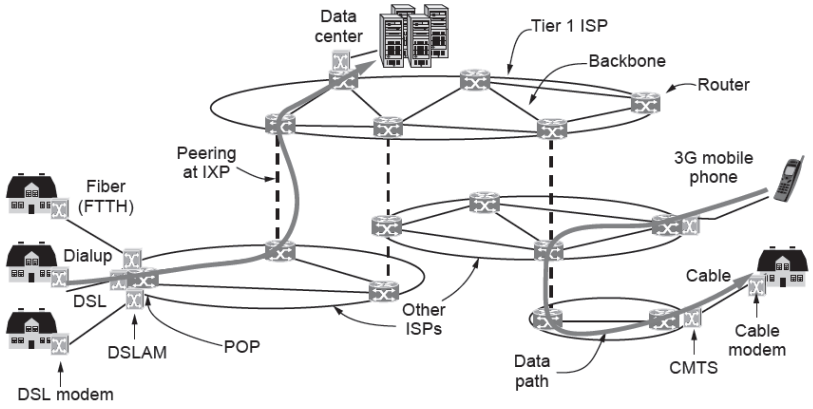


NSFNET topology in 1988

Internet – III

- The modern Internet is more complex:
  - ISP networks serve as the Internet backbone
  - ISPs connect or peer to exchange traffic at IXPs
  - Within each network routers switch packets
  - Between networks, traffic exchange is set by business agreements
  - Customers connect at the edge by many means
  - Cable, DSL, Fiber-to-the-Home, 3G/4G wireless, dialup
  - Data centers concentrate many servers (“the cloud”)
  - Most traffic is content from data centers (esp. video)
  - The architecture continues to evolve

Internet – IV



The Physical Layer

- Foundation on which other layers build
  - Properties of wires, fiber, wireless limit what the network can do
- Key problem is to send (digital) bits using only (analog) signals
  - This is called modulation

Application
Transport
Network
Link
Physical

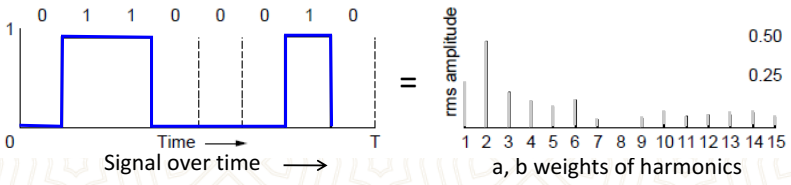
Theoretical Basis for Data Communication

- Communication rates have fundamental limits
  - Fourier analysis
  - Bandwidth-limited signals
  - Maximum data rate of a channel

Fourier Analysis

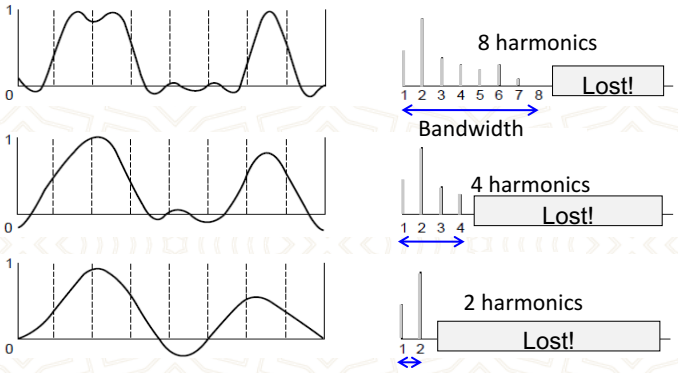
- A time-varying signal can be equivalently represented as a series of frequency components (harmonics):

$$g(t) = \frac{1}{2}c + \sum_{n=1}^{\infty} a_n \sin(2\pi nft) + \sum_{n=1}^{\infty} b_n \cos(2\pi nft)$$



Bandwidth-Limited Signals

- Having less bandwidth (harmonics) degrades the signal



Maximum Data Rate of a Channel

- Nyquist's theorem relates the data rate to the bandwidth (B) and number of signal levels (V):

$$\text{Max. data rate} = 2B \log_2 V \text{ bits/sec}$$

- Shannon's theorem relates the data rate to the bandwidth (B) and signal strength (S) relative to the noise (N):

$$\text{Max. data rate} = B \log_2(1 + S/N) \text{ bits/sec}$$

How fast signal can change

How many levels can be seen

Link Terminology

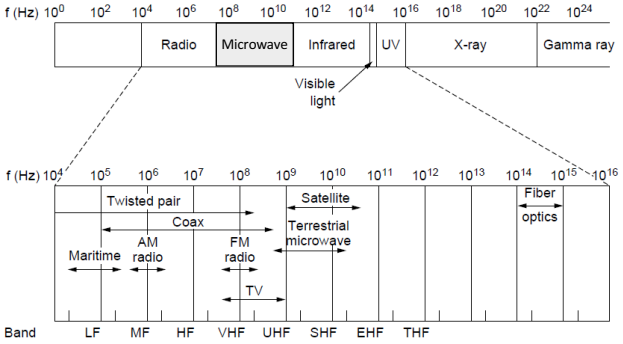
- Full-duplex link
  - Used for transmission in both directions at once
  - e.g., use different twisted pairs for each direction
- Half-duplex link
  - Both directions, but not at the same time
  - e.g., senders take turns on a wireless channel
- Simplex link
  - Only one fixed direction at all times; not common

Wireless Transmission

- Electromagnetic Spectrum
- Radio Transmission
- Microwave Transmission
- Light Transmission
- Wireless vs. Wires/Fiber

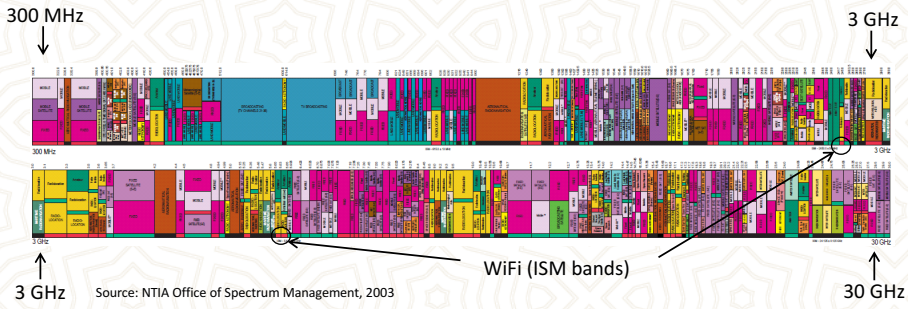
Electromagnetic Spectrum – I

- Different bands have different uses:
  - Radio: wide-area broadcast; Infrared/Light: line-of-sight
  - Microwave: LANs and 3G/4G;



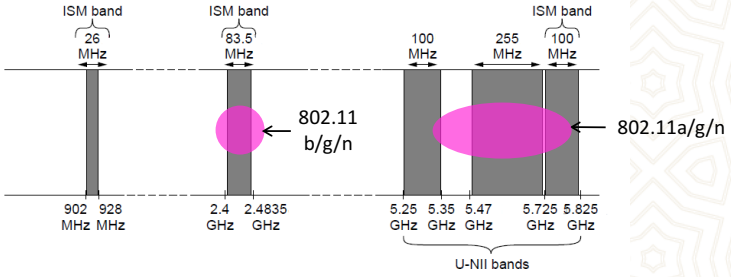
Electromagnetic Spectrum – II

- To manage interference, spectrum is carefully divided, and its use regulated and licensed, e.g., sold at auction.



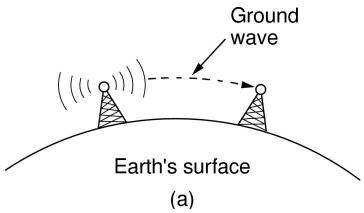
Electromagnetic Spectrum – III

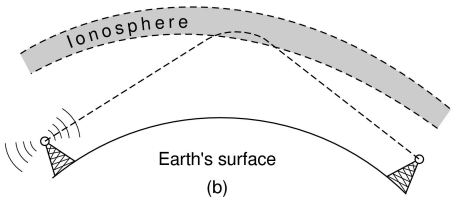
- Fortunately, there are also unlicensed ("ISM") bands:
  - Free for use at low power; devices manage interference
  - Widely used for networking; WiFi, Bluetooth, Zigbee, etc.



Radio Transmission

- In the VLF, LF, and MF bands, radio waves follow the curvature of the earth

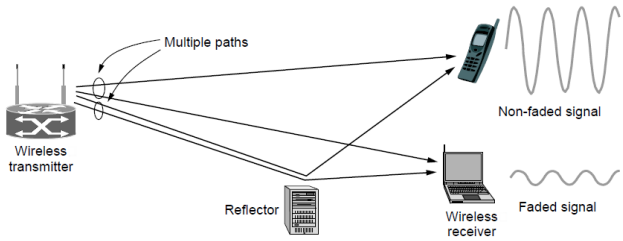




- Radio signals penetrate buildings well and propagate for long distances with path loss
- In the HF band, radio waves bounce off the ionosphere.

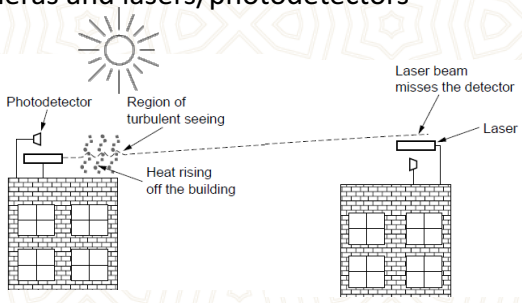
Microwave Transmission

- Microwaves have much bandwidth and are widely used indoors (WiFi) and outdoors (3G, satellites)
  - Signal is attenuated/reflected by everyday objects
  - Strength varies with mobility due multipath fading, etc.



Light Transmission

- Line-of-sight light (no fiber) can be used for links
  - Light is highly directional, has much bandwidth
  - Use of LEDs/cameras and lasers/photodetectors



Wireless vs. Wires/Fiber

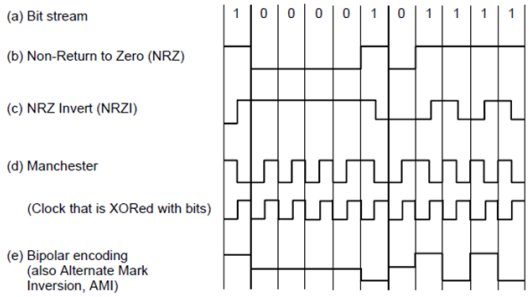
- Wireless:
  - + Easy and inexpensive to deploy
  - + Naturally supports mobility
  - + Naturally supports broadcast
  - Transmissions interfere and must be managed
  - Signal strengths hence data rates vary greatly
- Wires/Fiber:
  - + Easy to engineer a fixed data rate over point-to-point links
  - Can be expensive to deploy, esp. over distances
  - Doesn't readily support mobility or broadcast

Digital Modulation and Multiplexing

- Modulation schemes send bits as signals; multiplexing schemes share a channel among users.
  - Baseband Transmission
  - Passband Transmission
  - Frequency Division Multiplexing
  - Time Division Multiplexing
  - Code Division Multiple Access

Baseband Transmission

- Line codes send symbols that represent one or more bits
- NRZ is the simplest, literal line code (+1V="1", -1V="0")
- Other codes tradeoff bandwidth and signal transitions



Clock Recovery

- To decode the symbols, signals need sufficient transitions
  - Otherwise long runs of 0s (or 1s) are confusing, e.g.:

1 0 0 0 0 0 0 0 0 0 0 um, 0? er, 0?

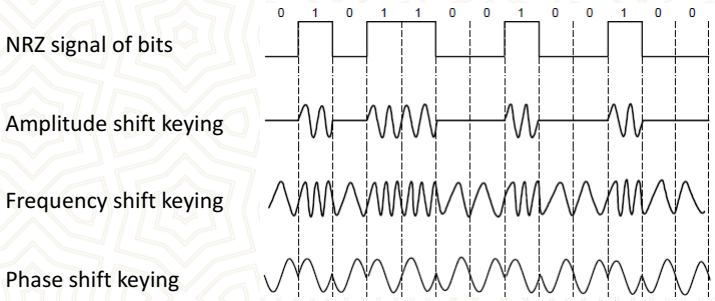
- Strategies:
  - Manchester coding, mixes clock signal in every symbol
  - 4B/5B maps 4 data bits to 5 coded bits with 1s and 0s:

Data	Code	Data	Code	Data	Code	Data	Code
0000	11110	0100	01010	1000	10010	1100	11010
0001	01001	0101	01011	1001	10011	1101	11011
0010	10100	0110	01110	1010	10110	1110	11100
0011	10101	0111	01111	1011	10111	1111	11101

  - Scrambler XORs tx/rx data with pseudorandom bits

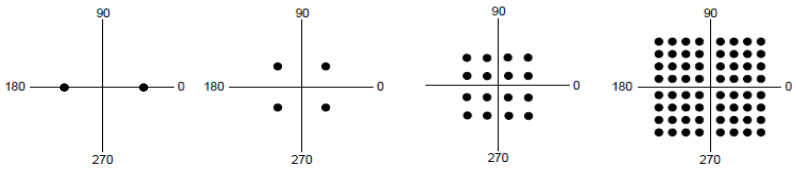
Passband Transmission – I

- Modulating the amplitude, frequency/phase of a carrier signal sends bits in a (non-zero) frequency range



Passband Transmission – II

- Constellation diagrams are a shorthand to capture the amplitude and phase modulations of symbols:



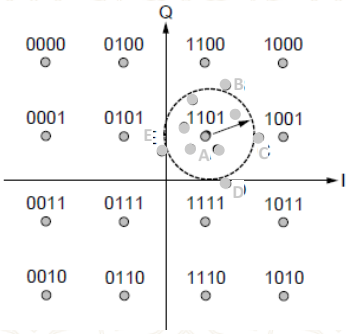
BPSK	QPSK	QAM-16	QAM-64
2 symbols	4 symbols	16 symbols	64 symbols
1 bit/symbol	2 bits/symbol	4 bits/symbol	6 bits/symbol

BPSK/QPSK varies only phase

QAM varies amplitude and phase

Passband Transmission – III

- Gray-coding assigns bits to symbols so that small symbol errors cause few bit errors:

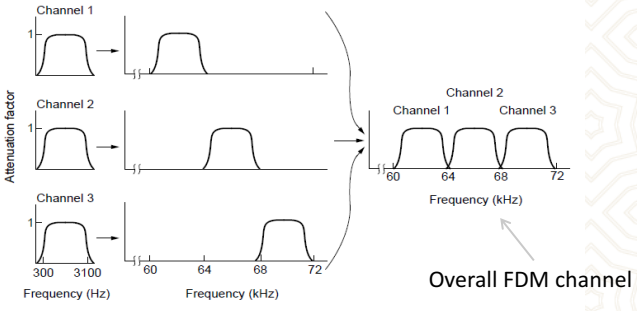


When 1101 is sent:

Point	Decodes as	Bit errors
A	1101	0
B	1100	1
C	1001	1
D	1111	1
E	0101	1

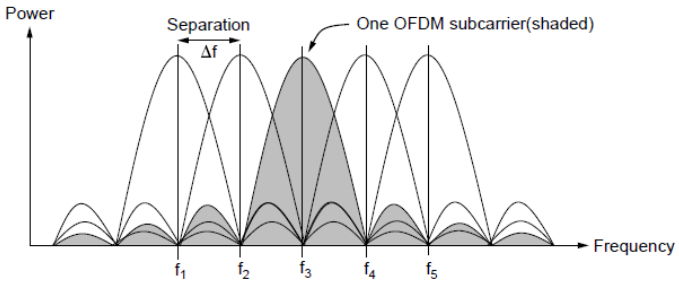
Frequency Division Multiplexing – I

- FDM (Frequency Division Multiplexing) shares the channel by placing users on different frequencies:



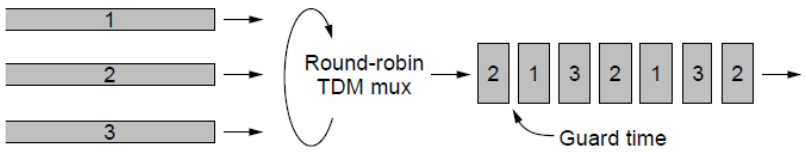
Frequency Division Multiplexing – II

- OFDM (Orthogonal FDM) is an efficient FDM technique used for 802.11, 4G cellular and other communications
  - Subcarriers are coordinated to be tightly packed



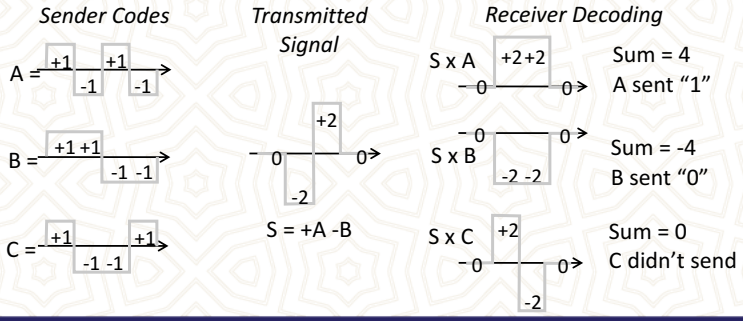
Time Division Multiplexing (TDM)

- Time division multiplexing shares a channel over time:
  - Users take turns on a fixed schedule; this is not packet switching or STDM (Statistical TDM)
  - Widely used in telephone / cellular systems



Code Division Multiple Access (CDMA)

- CDMA shares the channel by giving users a code
  - Codes are orthogonal; can be sent at the same time
  - Widely used as part of 3G networks



The Data Link Layer

- Responsible for delivering frames of information over a single link
  - Handles transmission errors and regulates the flow of data

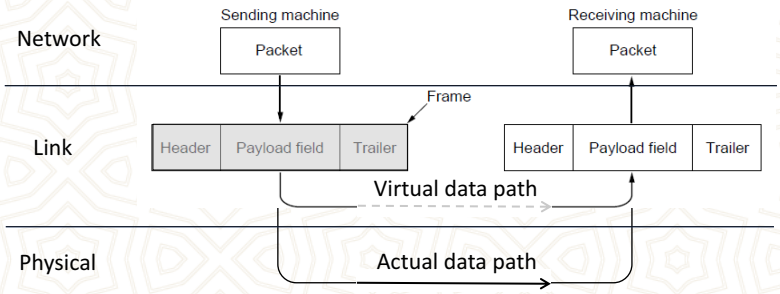
Application
Transport
Network
Link
Physical

Data Link Layer Design Issues

- Frames
- Possible services
- Framing methods
- Error control
- Flow control

Frames

- Link layer accepts packets from the network layer, and encapsulates them into frames that it sends using the physical layer; reception is the opposite process



Possible Services

- Unacknowledged connectionless service
  - Frame is sent with no connection / error recovery
  - Ethernet is example
- Acknowledged connectionless service
  - Frame is sent with retransmissions if needed
  - Example is 802.11
- Acknowledged connection-oriented service
  - Connection is set up; rare

Framing Methods

- Byte count
- Flag bytes with byte stuffing
- Flag bits with bit stuffing
- Physical layer coding violations
- Use non-data symbol to indicate frame

Error Control

- Error control repairs frames that are received in error
  - Requires errors to be detected at the receiver
  - Typically retransmit the unacknowledged frames
  - Timer protects against lost acknowledgements
- Detecting errors and retransmissions are next topics.

Flow Control

- Prevents a fast sender from out-pacing a slow receiver
  - Receiver gives feedback on the data it can accept
  - Rare in the Link layer as NICs run at “wire speed”
    - Receiver can take data as fast as it can be sent
- Flow control is a topic in the Link and Transport layers.

Error Detection and Correction

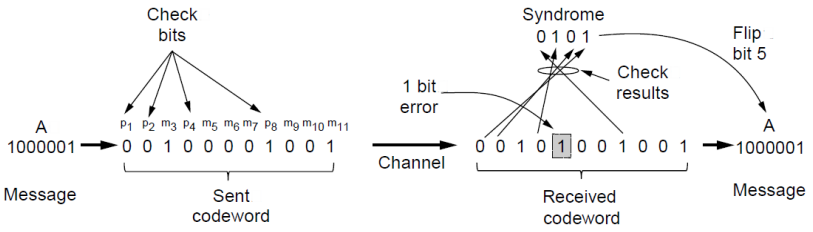
- Error codes add structured redundancy to data so errors can be either detected, or corrected.
- Error correction codes:
  - Hamming codes
  - Binary convolutional codes
  - Reed-Solomon and Low-Density Parity Check codes
    - Mathematically complex, widely used in real systems
- Error detection codes:
  - Parity
  - Checksums
  - Cyclic redundancy codes

Error Bounds – Hamming distance

- Code turns data of n bits into codewords of n+k bits
- Hamming distance is the minimum bit flips to turn one valid codeword into any other valid one.
  - Example with 4 codewords of 10 bits (n=2, k=8):
    - 0000000000, 0000011111, 1111100000, and 1111111111
    - Hamming distance is 5
- Bounds for a code with distance:
  - 2d+1 – can correct d errors (e.g., 2 errors above)
  - d+1 – can detect d errors (e.g., 4 errors above)

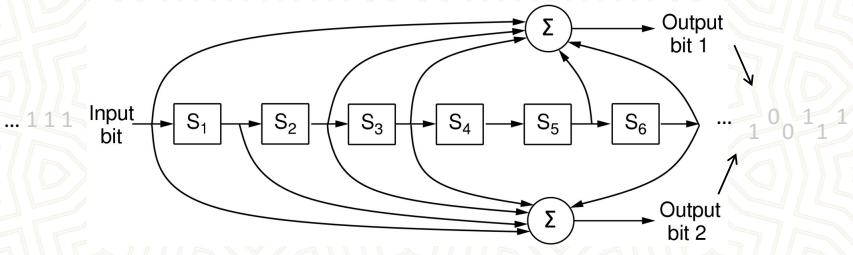
Error Correction – Hamming code

- Hamming code gives a simple way to add check bits and correct up to a single bit error:
  - Check bits are parity over subsets of the codeword
  - Recomputing the parity sums (syndrome) gives the position of the error to flip, or 0 if there is no error



Error Correction – Convolutional codes

- Operates on a stream of bits, keeping internal state
  - Output stream is a function of all preceding input bits
  - Bits are decoded with the Viterbi algorithm



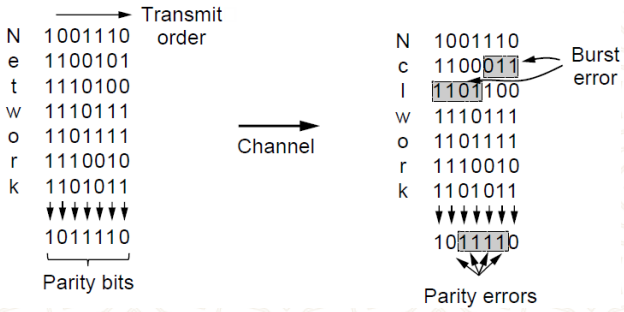
Popular NASA binary convolutional code (rate = ½) used in 802.11

Error Detection – Parity – I

- Parity bit is added as the modulo 2 sum of data bits
  - Equivalent to XOR; this is even parity
  - Ex: 1110000 → 11100001
  - Detection checks if the sum is wrong (an error)
- Simple way to detect an *odd* number of errors
  - Ex: 1 error, 11100101; detected, sum is wrong
  - Ex: 3 errors, 11011001; detected sum is wrong
  - Ex: 2 errors, 11101101; *not detected*, sum is right!
  - Error can also be in the parity bit itself
  - Random errors are detected with probability ½

Error Detection – Parity – II

- Interleaving of N parity bits detects burst errors up to N
  - Each parity sum is made over non-adjacent bits
  - An even burst of up to N errors will not cause it to fail

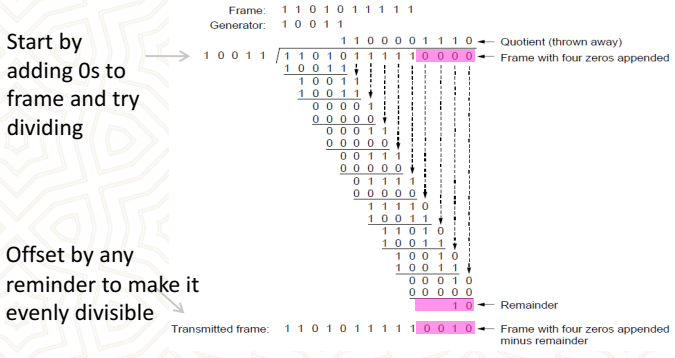


Error Detection – Checksums

- Checksum treats data as N-bit words and adds N check bits that are the modulo 2^N sum of the words
  - Ex: Internet 16-bit 1s complement checksum
- Properties:
  - Improved error detection over parity bits
  - Detects bursts up to N errors
  - Detects random errors with probability $1-2^N$
  - Vulnerable to systematic errors, e.g., added zeros

Error Detection – CRCs – I

- Adds bits so that transmitted frame viewed as a polynomial is evenly divisible by a generator polynomial



Error Detection – CRCs – II

- Based on standard polynomials:
  - Ex: Ethernet 32-bit CRC is defined by:

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1$$

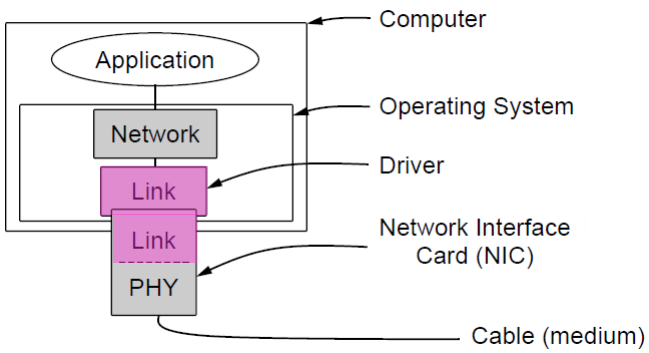
  - Computed with simple shift/XOR circuits
- Stronger detection than checksums:
  - E.g., can detect all double bit errors
  - Not vulnerable to systematic errors

Elementary Data Link Protocols

- Link layer environment
- Utopian Simplex Protocol
- Stop-and-Wait Protocol for Error-free channel
- Stop-and-Wait Protocol for Noisy channel

Link Layer Environment – I

- Commonly implemented as NICs and OS drivers;
- Network layer (IP) is often OS software



Link Layer Environment – II

- Link layer protocol implementations use library functions
  - See code (`protocol.h`) for more details

Group	Library Function	Description
Network layer	from_network_layer(&packet) to_network_layer(&packet) enable_network_layer() disable_network_layer()	Take a packet from network layer to send Deliver a received packet to network layer Let network cause “ready” events Prevent network “ready” events
Physical layer	from_physical_layer(&frame) to_physical_layer(&frame)	Get an incoming frame from physical layer Pass an outgoing frame to physical layer
Events & timers	wait_for_event(&event) start_timer(seq_nr) stop_timer(seq_nr) start_ack_timer() stop_ack_timer()	Wait for a packet / frame / timer event Start a countdown timer running Stop a countdown timer from running Start the ACK countdown timer Stop the ACK countdown timer

Utopian Simplex Protocol

- An optimistic protocol (p1) to get us started
  - Assumes no errors, and receiver as fast as sender
  - Considers one-way data transfer
  - That's it, no error or flow control …

```
void sender1(void)
{
  frame s;
  packet buffer;

  while (true) {
      from_network_layer(&buffer);
      s.info = buffer;
      to_physical_layer(&s);
  }
}
```

Sender loops blasting frames

```
void receiver1(void)
{
  frame r;
  event_type event;

  while (true) {
      wait_for_event(&event);
      from_physical_layer(&r);
      to_network_layer(&r.info);
  }
}
```

Receiver loops eating frames

Stop-and-Wait – Noisy channel – I

- ARQ (Automatic Repeat reQuest) adds error control
  - Receiver acks frames that are correctly delivered
  - Sender sets timer and resends frame if no ack)
- For correctness, frames and acks must be numbered
  - Else receiver can't tell retransmission (due to lost ack or early timer) from new frame
  - For stop-and-wait, 2 numbers (1 bit) are sufficient

Stop-and-Wait – Noisy channel – II

- Sender loop (p3):

```
void sender3(void) {
  seq_nr next_frame_to_send;
  frame s;
  packet buffer;
  event_type event;

  next_frame_to_send = 0;
  from_network_layer(&buffer);
  while (true) {
      s.info = buffer;
      s.seq = next_frame_to_send;
      to_physical_layer(&s);
      start_timer(s.seq);
      wait_for_event(&event);
      if (event == frame_arrival) {
            from_physical_layer(&s);
            if (s.ack == next_frame_to_send) {
                  stop_timer(s.ack);
                  from_network_layer(&buffer);
                  inc(next_frame_to_send);
            }
      }
  }
}
```

Send frame (or retransmission)
Set timer for retransmission
Wait for ack or timeout

If a good ack then set up for the next frame to send (else the old frame will be retransmitted)

Stop-and-Wait – Noisy channel – III

- Receiver loop (p3):

```
void receiver3(void)
{
  seq_nr frame_expected;
  frame r, s;
  event_type event;

  frame_expected = 0;
  while (true) {
      wait_for_event(&event);
      if (event == frame_arrival) {
            from_physical_layer(&r);
            if (r.seq == frame_expected) {
                  to_network_layer(&r.info);
                  inc(frame_expected);
            }
            s.ack = 1 – frame_expected;
            to_physical_layer(&s);
      }
  }
}
```

Wait for a frame

If it's new then take it and advance expected frame

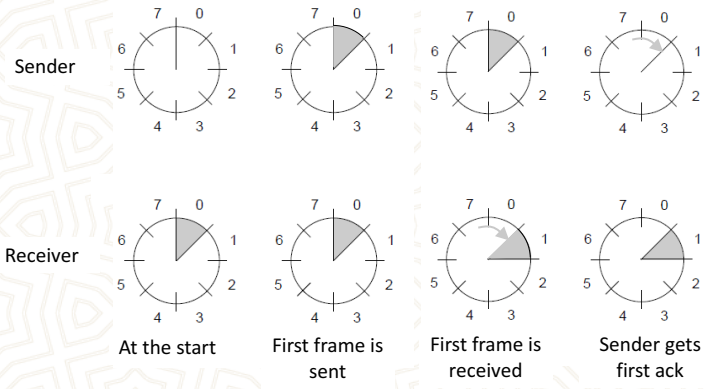
Ack current frame

Sliding Window Concept – I

- Sender maintains window of frames it can send
  - Needs to buffer them for possible retransmission
  - Window advances with next acknowledgements
- Receiver maintains window of frames it can receive
  - Needs to keep buffer space for arrivals
  - Window advances with in-order arrivals

Sliding Window Concept – II

- A sliding window advancing at the sender and receiver
  - Ex: window size is 1, with a 3-bit sequence number.

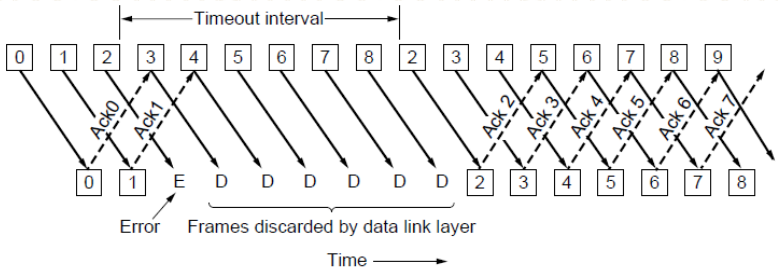


Sliding Window Concept – III

- Larger windows enable pipelining for efficient link use
  - Stop-and-wait (w=1) is inefficient for long links
  - Best window (w) depends on bandwidth-delay (BD)
  - Want w ≥ 2BD+1 to ensure high link utilization
- Pipelining leads to different choices for errors/buffering
  - We will consider Go-Back-N and Selective Repeat

Go-Back-N – I

- Receiver only accepts/acks frames that arrive in order:
  - Discards frames that follow a missing/errored frame
  - Sender times out and resends all outstanding frames

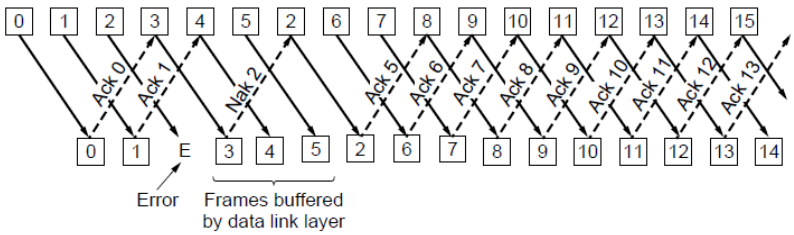


Go-Back-N – II

- Tradeoff made for Go-Back-N:
  - Simple strategy for receiver; needs only 1 frame
  - Wastes link bandwidth for errors with large windows; entire window is retransmitted
- Implemented as p5 (see code in book)

Selective Repeat – I

- Receiver accepts frames anywhere in receive window
  - Cumulative ack indicates highest in-order frame
  - NAK (negative ack) causes sender retransmission of a missing frame before a timeout resends window

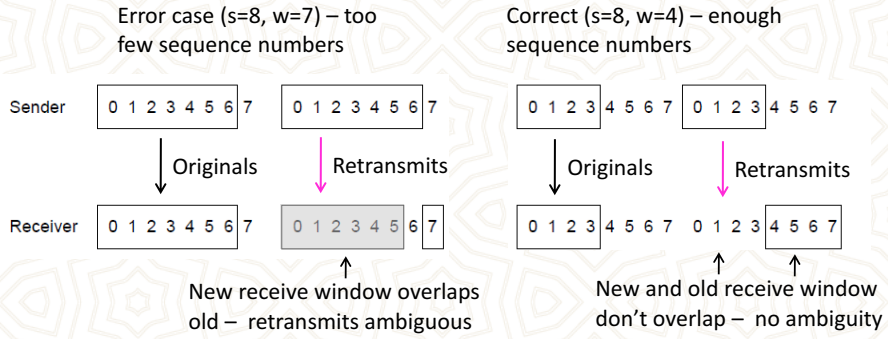


Selective Repeat – II

- Tradeoff made for Selective Repeat:
  - More complex than Go-Back-N due to buffering at receiver and multiple timers at sender
  - More efficient use of link bandwidth as only lost frames are resent (with low error rates)
- Implemented as p6 (see code in book)

Selective Repeat – III

- For correctness, we require:
  - Sequence numbers (s) at least twice the window (w)

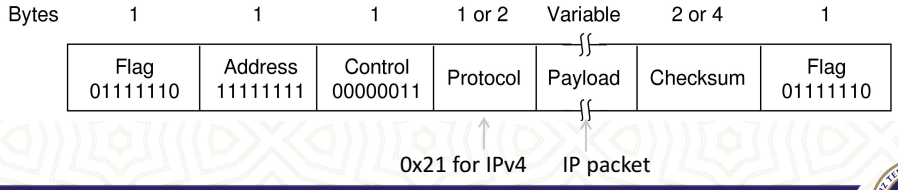


Example Data Link Protocols

- Packet over SONET
- PPP (Point-to-Point Protocol)
- ADSL (Asymmetric Digital Subscriber Loop)

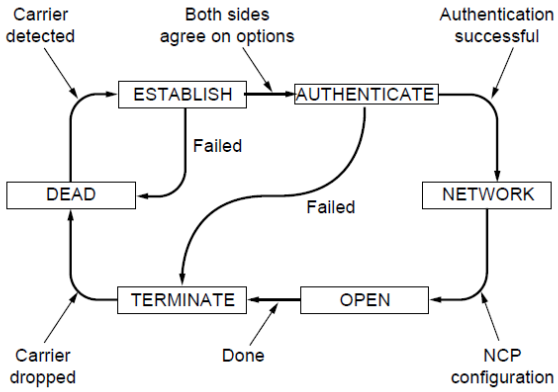
PPP – I

- PPP (Point-to-Point Protocol) is a general method for delivering packets across links
  - Framing uses a flag (0x7E) and byte stuffing
  - "Unnumbered mode" (connectionless unacknow-ledged service) is used to carry IP packets
  - Errors are detected with a checksum



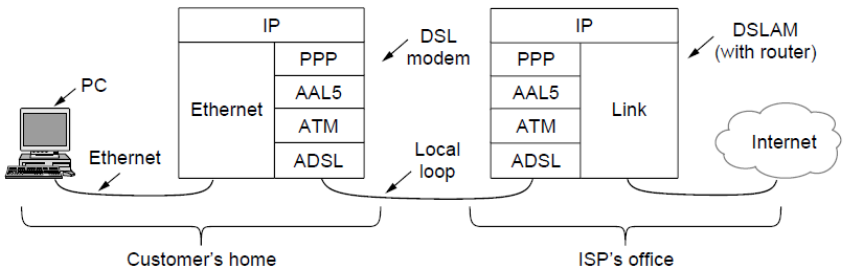
PPP – II

- A link control protocol brings the PPP link up/down



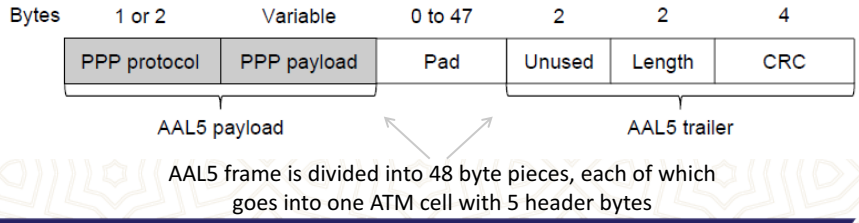
ADSL – I

- Widely used for broadband Internet over local loops
  - ADSL runs from modem (customer) to DSLAM (ISP)
  - IP packets are sent over PPP and AAL5/ATM (over)



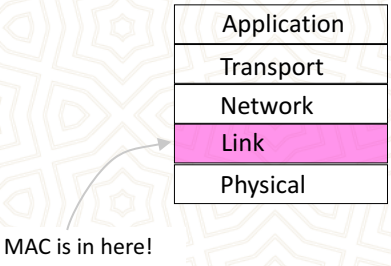
ADSL – II

- PPP data is sent in AAL5 frames over ATM cells:
  - ATM is a link layer that uses short, fixed-size cells (53 bytes); each cell has a virtual circuit identifier
  - AAL5 is a format to send packets over ATM
  - PPP frame is converted to a AAL5 frame (PPPoA)



The MAC Sublayer

- Responsible for deciding who sends next on a multi-access link
  - An important part of the link layer, especially for LANs



Channel Allocation Problem – I

- For fixed channel and traffic from N users
  - Divide up bandwidth using FTM, TDM, CDMA, etc.
  - This is a static allocation, e.g., FM radio
- This static allocation performs poorly for bursty traffic
  - Allocation to a user will sometimes go unused

Channel Allocation Problem – II

- Dynamic allocation gives the channel to a user when they need it. Potentially N times as efficient for N users.
- Schemes vary with assumptions:

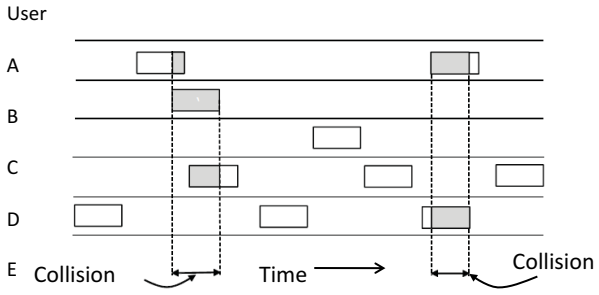
Assumption	Implication
Independent traffic	Often not a good model, but permits analysis
Single channel	No external way to coordinate senders
Observable collisions	Needed for reliability; mechanisms vary
Continuous or slotted time	Slotting may improve performance
Carrier sense	Can improve performance if available

Multiple Access Protocols

- ALOHA
- CSMA (Carrier Sense Multiple Access)
- Collision-free protocols
- Limited-contention protocols
- Wireless LAN protocols

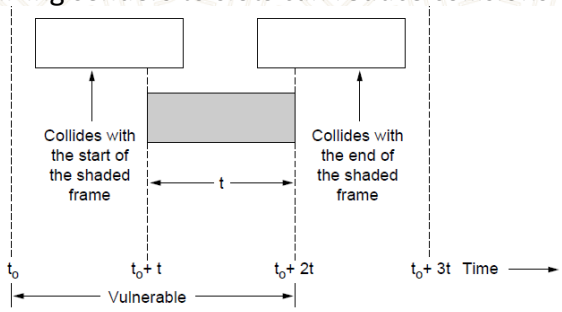
ALOHA – I

- In pure ALOHA, users transmit frames whenever they have data; users retry after a random time for collisions
  - Efficient and low-delay under low load



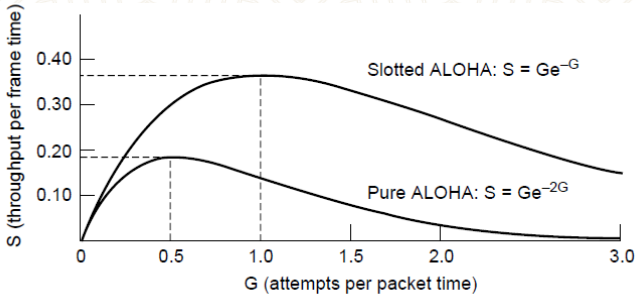
ALOHA – II

- Collisions happen when other users transmit during a vulnerable period that is twice the frame time
  - Synchronizing senders to slots can reduce collisions



ALOHA – III

- Slotted ALOHA is twice as efficient as pure ALOHA
  - Low load wastes slots, high loads causes collisions
  - Efficiency up to 1/e (37%) for random traffic models

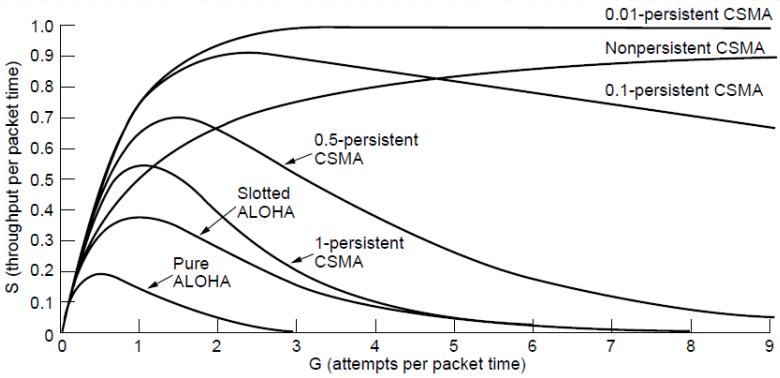


CSMA – I

- CSMA improves on ALOHA by sensing the channel!
  - User doesn't send if it senses someone else
- Variations on what to do if the channel is busy:
  - 1-persistent (greedy) sends as soon as idle
  - Nonpersistent waits a random time then tries again
  - p-persistent sends with probability p when idle

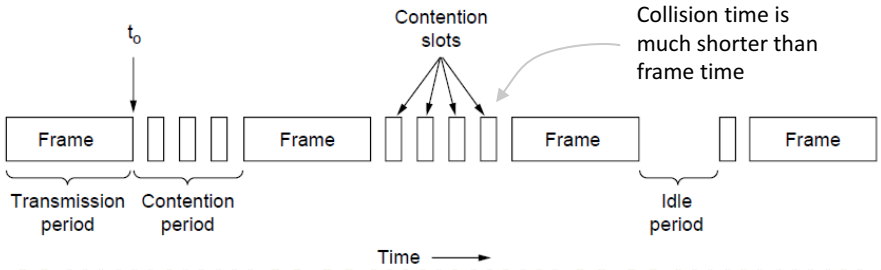
CSMA – II

- CSMA outperforms ALOHA, and being less persistent is better under high load



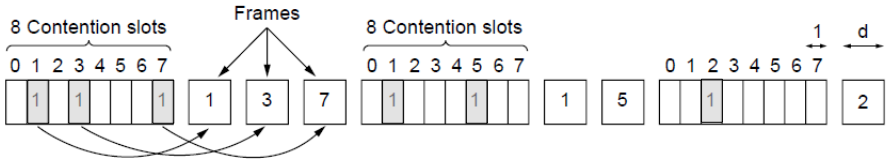
CSMA – III Collision Detection

- CSMA/CD improvement is to detect/abort collisions
  - Reduced contention times improve performance



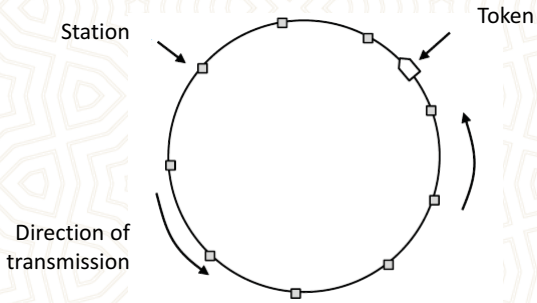
Collision-Free – I – Bitmap

- Collision-free protocols avoid collisions entirely
  - Senders must know when it is their turn to send
- The basic bit-map protocol:
  - Sender set a bit in contention slot if they have data
  - Senders send in turn; everyone knows who has data



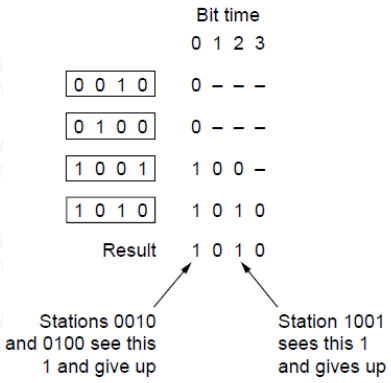
Collision-Free – II – Token Ring

- Token sent round ring defines the sending order
  - Station with token may send a frame before passing
  - Idea can be used without ring too, e.g., token bus



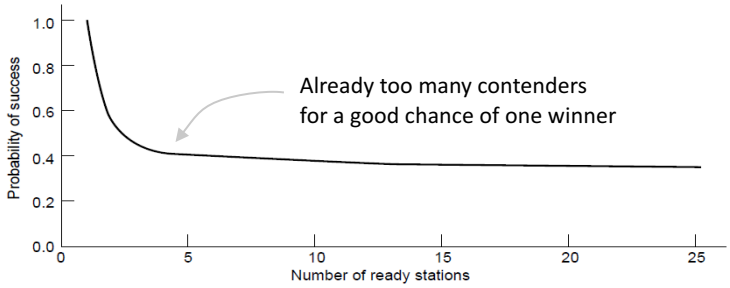
Collision-Free – III – Countdown

- Stations send their address in contention slot (log N bits instead of N bits)
- Medium ORs bits; stations give up when they send a "0" but see a "1"
- Station that sees its full address is next to send
- Binary countdown **improves** on the bitmap protocol



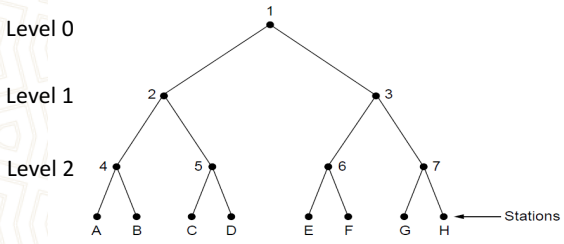
Limited-Contention Protocols – I

- Idea is to divide stations into groups within which only a very small number are likely to want to send
  - Avoids wastage due to idle periods and collisions



Limited-Contention Protocols – II

- Tree divides stations into groups (nodes) to poll
  - Depth first search under nodes with poll collisions
  - Start search at lower levels if >1 station expected

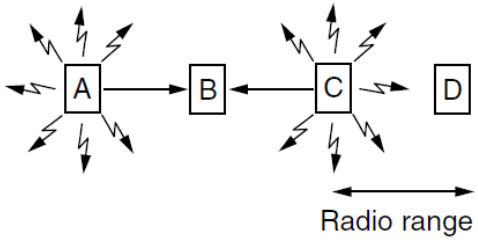


Wireless LAN Protocols – I

- Wireless has complications compared to wired.
- Nodes may have different coverage regions
  - Leads to hidden and exposed terminals
- Nodes can't detect collisions, i.e., sense while sending
  - Makes collisions expensive and to be avoided

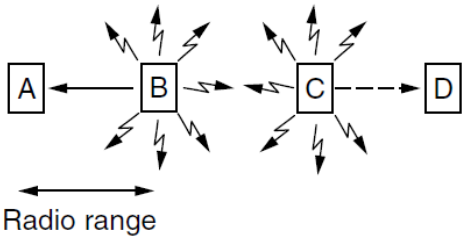
Wireless LAN Protocols – II

- Hidden terminals are senders that cannot sense each other but nonetheless collide at intended receiver
  - Want to prevent; loss of efficiency
  - A and C are hidden terminals when sending to B



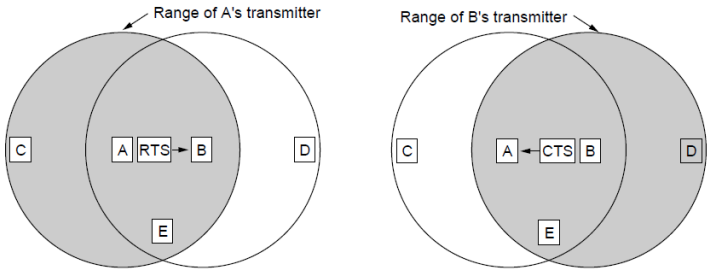
Wireless LAN Protocols – III

- Exposed terminals are senders who can sense each other but still transmit safely (to different receivers)
  - Desirably concurrency; improves performance
  - B → A and C → D are exposed terminals



Wireless LAN Protocols – IV

- MACA protocol grants access for A to send to B:
  - A sends RTS to B [left]; B replies with CTS [right]
  - A can send with exposed but no hidden terminals

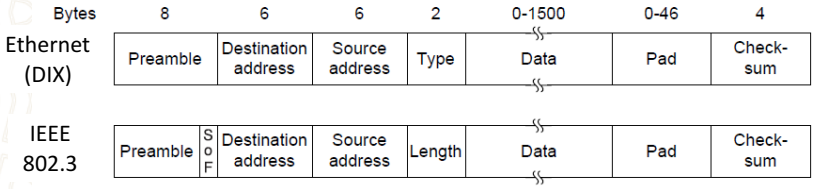


Ethernet

- Classic Ethernet
- Switched/Fast Ethernet
- Gigabit/10 Gigabit Ethernet

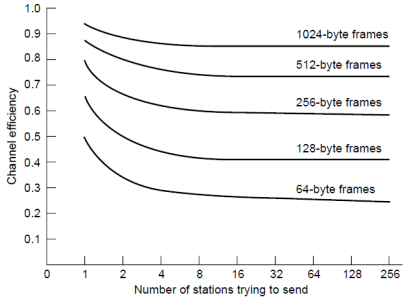
Classic Ethernet – MAC

- MAC protocol is 1-persistent CSMA/CD (earlier)
  - Random delay (backoff) after collision is computed with BEB (Binary Exponential Backoff)
  - Frame format is still used with modern Ethernet.



Classic Ethernet – Performance

- Efficient for large frames, even with many senders
  - Degrades for small frames (and long LANs)



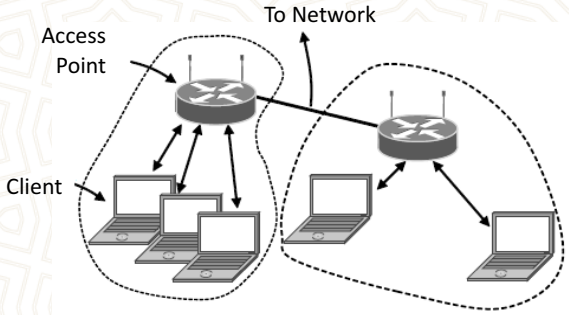
10 Mbps Ethernet,
64 byte min. frame

Wireless LANs

- 802.11 architecture/protocol stack
- 802.11 physical layer
- 802.11 MAC
- 802.11 frames

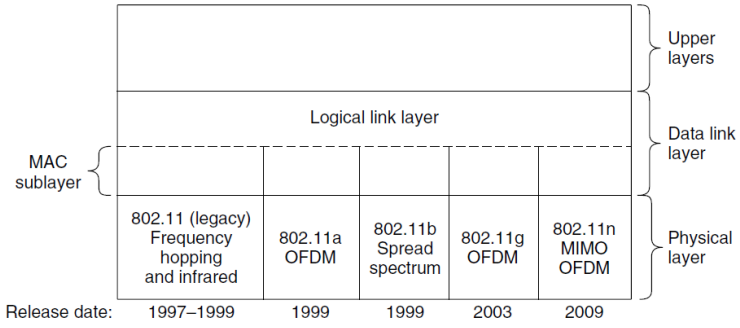
802.11 Architecture/Protocol Stack – I

- Wireless clients associate to a wired AP (Access Point)
  - Called infrastructure mode; there is also ad-hoc mode with no AP, but that is rare.



802.11 Architecture/Protocol Stack – II

- MAC is used across different physical layers



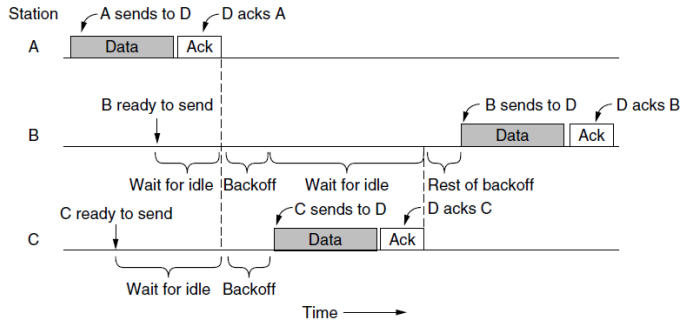
802.11 Physical Layer

- NICs are compatible with multiple physical layers
  - E.g., 802.11 a/b/g

Name	Technique	Max. Bit Rate
802.11b	Spread spectrum, 2.4 GHz	11 Mbps
802.11g	OFDM, 2.4 GHz	54 Mbps
802.11a	OFDM, 5 GHz	54 Mbps
802.11n	OFDM with MIMO, 2.4/5 GHz	600 Mbps

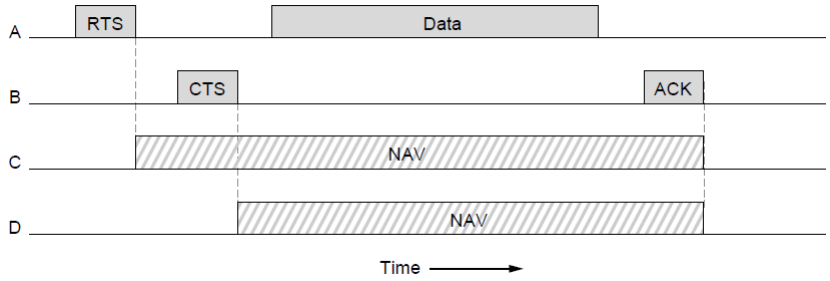
802.11 MAC – I

- CSMA/CA inserts backoff slots to avoid collisions
- MAC uses ACKs/retransmissions for wireless errors



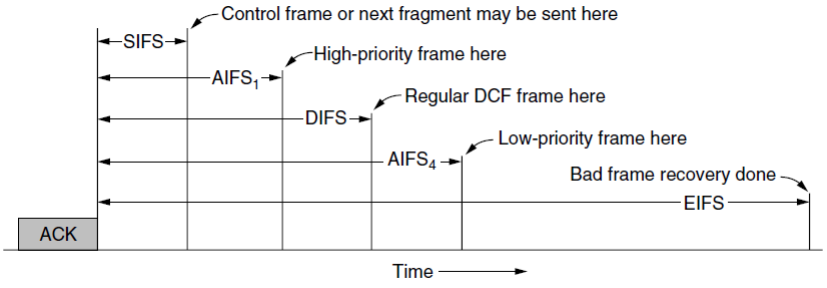
802.11 MAC – II

- Virtual channel sensing with the NAV and optional RTS/CTS (often not used) avoids hidden terminals



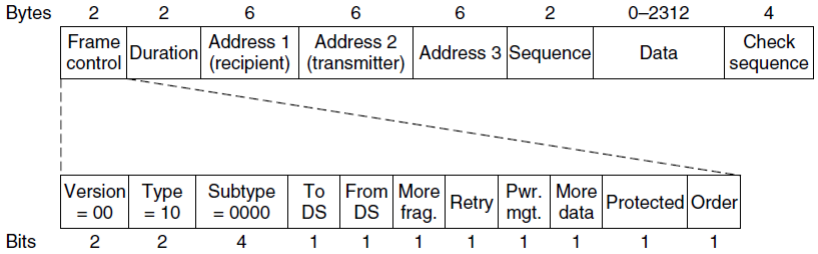
802.11 MAC – III

- Different backoff slot times add quality of service
  - Short intervals give preferred access, e.g., control, VoIP
- MAC has other mechanisms too, e.g., power save



802.11 Frames

- Frames vary depending on their type (Frame control)
- Data frames have 3 addresses to pass via APs

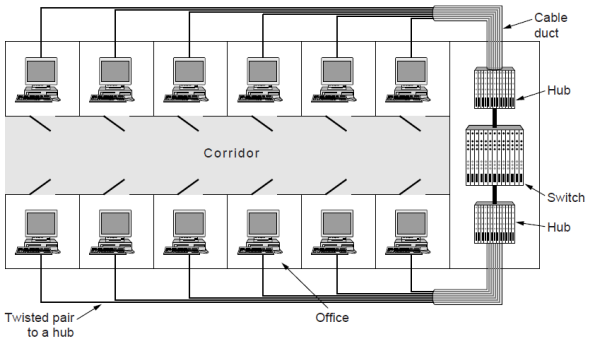


Data Link Layer Switching

- Uses of Bridges
- Learning Bridges
- Spanning Tree
- Repeaters, hubs, bridges, .., routers, gateways
- Virtual LANs

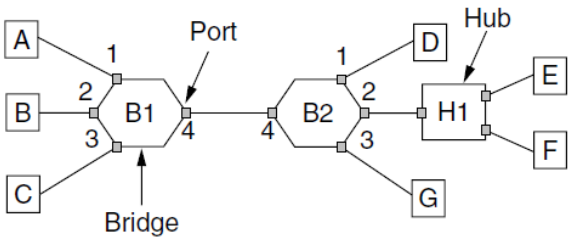
Uses of Bridges

- Common setup is a building with centralized wiring
  - Bridges (switches) are placed in or near wiring closets



Learning Bridges – I

- A bridge operates as a switched LAN (not a hub)
  - Computers, bridges, and hubs connect to its ports

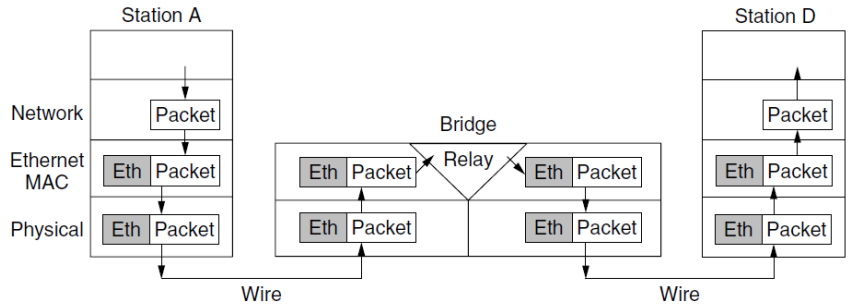


Learning Bridges – II

- Backward learning algorithm picks the output port:
  - Associates source address on frame with input port
  - Frame with destination address sent to learned port
  - Unlearned destinations are sent to all other ports
- Needs no configuration
  - Forget unused addresses to allow changes
  - Bandwidth efficient for two-way traffic

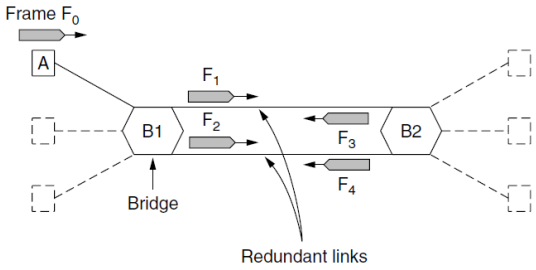
Learning Bridges – III

- Bridges extend the Link layer:
  - Use but don't remove Ethernet header/addresses
  - Do not inspect Network header



Spanning Tree – I

- Bridge topologies with loops and only backward learning will cause frames to circulate for ever
  - Need spanning tree support to solve problem



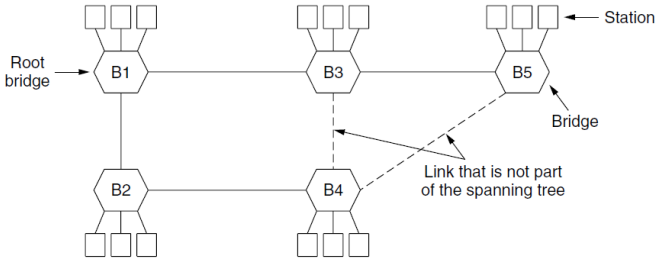
Spanning Tree – II

- Subset of forwarding ports for data is used to avoid loops
- Selected with the spanning tree distributed algorithm by Perlman

I think that I shall never see
A graph more lovely than a tree.
A tree whose crucial property
Is loop-free connectivity.
A tree which must be sure to span.
So packets can reach every LAN.
First the Root must be selected
By ID it is elected.
Least cost paths from Root are traced
In the tree these paths are placed.
A mesh is made by folks like me
Then bridges find a spanning tree.
– Radia Perlman, 1985.

Spanning Tree – III

- After the algorithm runs:
  - B1 is the root, two dashed links are turned off
  - B4 uses link to B2 (lower than B3 also at distance 1)
  - B5 uses B3 (distance 1 versus B4 at distance 2)



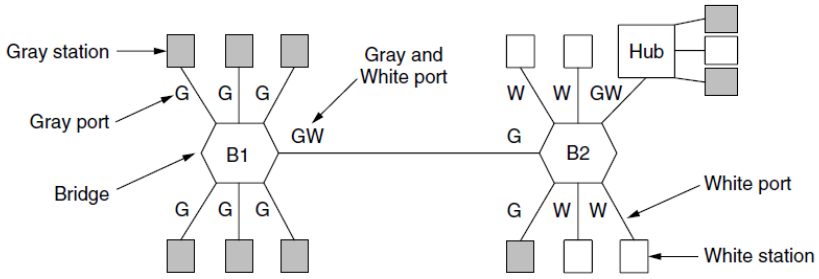
Repeaters, Hubs, Bridges, Switches, Routers, & Gateways

- Devices are named according to the layer they process
  - A bridge or LAN switch operates in the Link layer

Application layer	Application gateway
Transport layer	Transport gateway
Network layer	Router
Data link layer	Bridge, switch
Physical layer	Repeater, hub

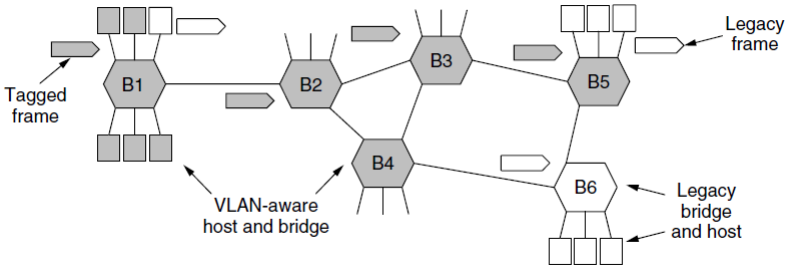
Virtual LANs – I

- VLANs (Virtual LANs) splits one physical LAN into multiple logical LANs to ease management tasks
  - Ports are "colored" according to their VLAN



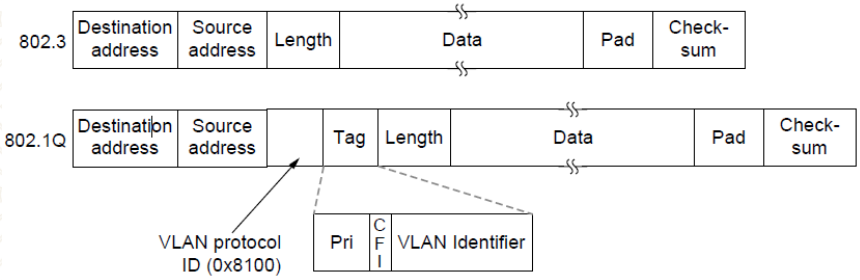
Virtual LANs – II

- Bridges need to be aware of VLANs to support them
  - In 802.1Q, frames are tagged with their "color"
  - Legacy switches with no tags are supported



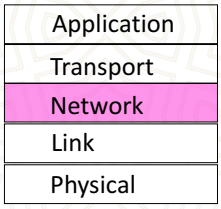
Virtual LANs – III

- 802.1Q frames carry a color tag (VLAN identifier)
  - Length/Type value is 0x8100 for VLAN protocol



The Network Layer

- Responsible for delivering packets between endpoints over multiple links

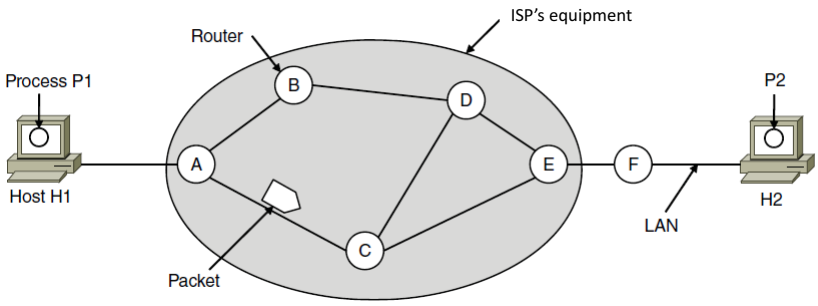


Design Issues

- Store-and-forward packet switching »
- Connectionless service – datagrams »
- Connection-oriented service – virtual circuits »
- Comparison of virtual-circuits and datagrams »

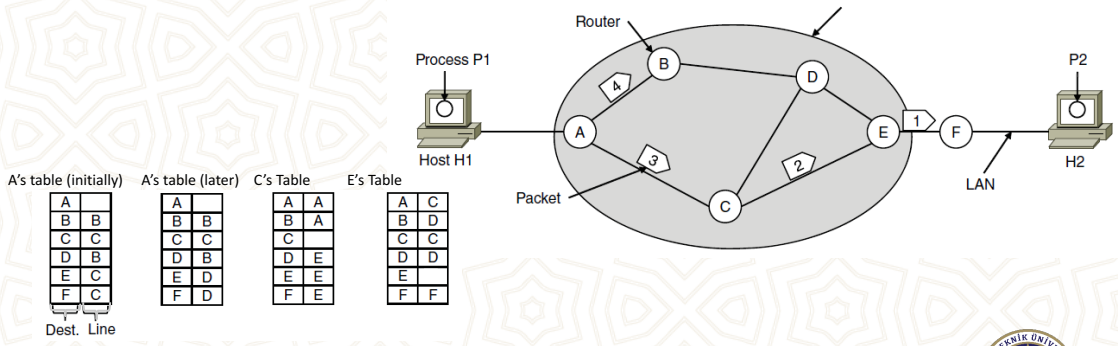
Store-and-Forward Packet Switching

- Hosts send packets into the network; packets are forwarded by routers



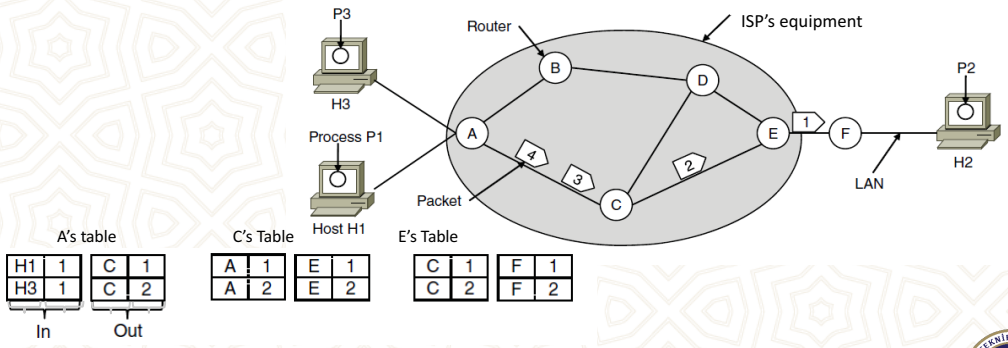
Connectionless Service – Datagrams

- Packet is forwarded using destination address inside it
  - Different packets may take different paths



Connection-Oriented – Virtual Circuits

- Packet is forwarded along a virtual circuit using tag inside it
  - Virtual circuit (VC) is set up ahead of time



Comparison of Virtual-Circuits & Datagrams

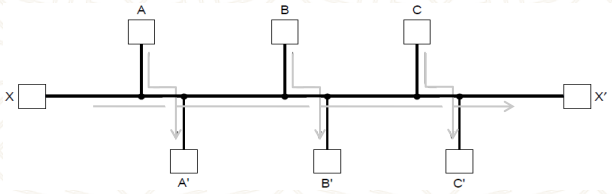
Issue	Datagram network	Virtual-circuit network
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Routing Algorithms – I

- Optimality principle
- Shortest path algorithm
- Flooding
- Distance vector routing
- Link state routing
- Hierarchical routing
- Broadcast routing
- Multicast routing
- Anycast routing
- Routing for mobile hosts
- Routing in ad hoc networks

Routing Algorithms – II

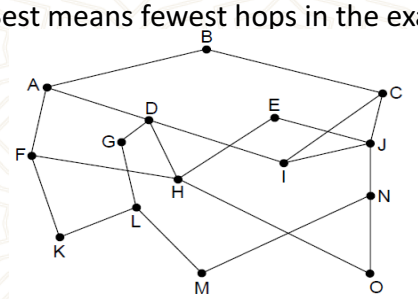
- Routing is the process of discovering network paths
  - Model the network as a graph of nodes and links
  - Decide what to optimize (e.g., fairness vs efficiency)
  - Update routes for changes in topology (e.g., failures)



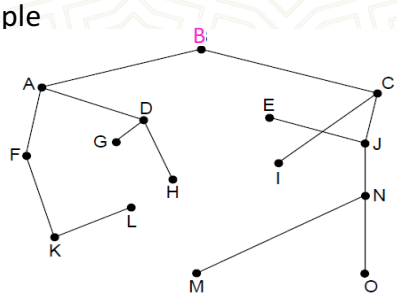
- Forwarding is the sending of packets along a path

The Optimality Principle

- Each portion of a best path is also a best path; the union of them to a router is a tree called the sink tree
  - Best means fewest hops in the example



Network

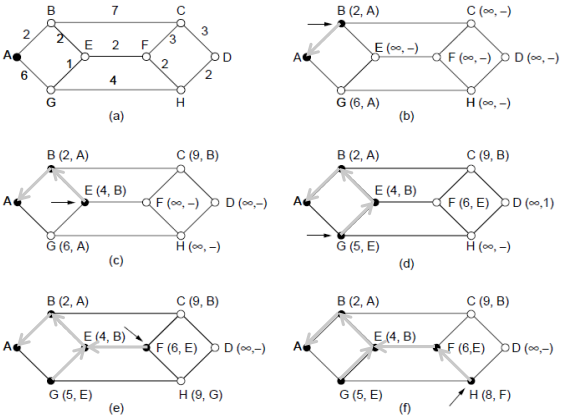


Sink tree of best paths to router B

Shortest Path Algorithm – I

- Dijkstra's algorithm computes a sink tree on the graph:
  - Each link is assigned a non-negative weight/distance
  - Shortest path is the one with lowest total weight
  - Using weights of 1 gives paths with fewest hops
- Algorithm:
  - Start with sink, set distance at other nodes to infinity
  - Relax distance to other nodes
  - Pick the lowest distance node, add it to sink tree
  - Repeat until all nodes are in the sink tree

Shortest Path Algorithm – II



- A network and first five steps in computing the shortest paths from A to D. Pink arrows show the sink tree so far.

Shortest Path Algorithm – III

```
for (p = &state[0]; p < &state[n]; p++) {        /* initialize state */
    p->predecessor = –1;
    p->length = INFINITY;
    p->label = tentative;
}
state[t].length = 0;  state[t].label = permanent;
k = t;
do {
    for (i = 0; i < n; i++)
        if (dist[k][i] != 0 && state[i].label == tentative) {
            if (state[k].length + dist[k][i] < state[i].length) {
                state[i].predecessor = k;
                state[i].length = state[k].length + dist[k][i];
            }
        }
. . .
```

Start with the sink, all other nodes are unreachable

Relaxation step. Lower distance to nodes linked to newest member of the sink tree

Shortest Path Algorithm – IV

```
. . .
    k = 0; min = INFINITY;
    for (i = 0; i < n; i++)
        if (state[i].label == tentative && state[i].length < min) {
            min = state[i].length;
            k = i;
        }
    state[k].label = permanent;
} while (k != s);
```

Find the lowest distance, add it to the sink tree, and repeat until done

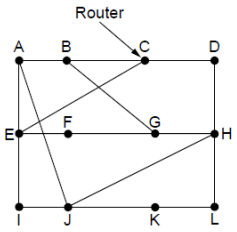
Flooding

- A simple method to send a packet to all network nodes
- Each node floods a new packet received on an incoming link by sending it out all of the other links
- Nodes need to keep track of flooded packets to stop the flood; even using a hop limit can blow up exponentially

Distance Vector Routing – I

- Distance vector is a distributed routing algorithm
  - Shortest path computation is split across nodes
- Algorithm:
  - Each node knows distance of links to its neighbors
  - Each node advertises vector of lowest known distances to all neighbors
  - Each node uses received vectors to update its own
  - Repeat periodically

Distance Vector Routing – II



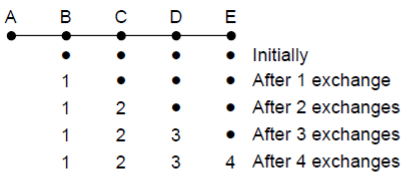
Network

To	A	I	H	K	New estimated delay from J	Line
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6		

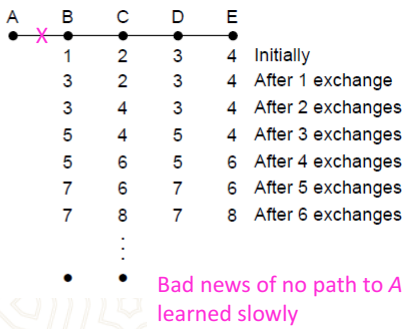
Vectors received at J from Neighbors A, I, H and K

New vector for J

The Count-to-Infinity Problem



Good news of a path to *A* spreads quickly



Bad news of no path to *A* is learned slowly

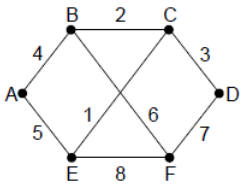
- Failures can cause DV to "count to infinity" while seeking a path to an unreachable node

Link State Routing – I

- Link state is an alternative to distance vector
  - More computation but simpler dynamics
  - Widely used in the Internet (OSPF, ISIS)
- Algorithm:
  - Each node floods information about its neighbors in LSPs (Link State Packets); all nodes learn the full network graph
  - Each node runs Dijkstra's algorithm to compute the path to take for each destination

Link State Routing – II

- LSP (Link State Packet) for a node lists neighbors and weights of links to reach them



Network

Link	State			Packets	
A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

LSP for each node

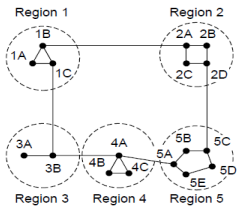
Link State Routing – III (Reliable flooding)

- Seq. number and age are used for reliable flooding
  - New LSPs are acknowledged on the lines they are received and sent on all other lines
  - Example shows the LSP database at router B

			Send flags			ACK flags			
Source	Seq.	Age	A	C	F	A	C	F	Data
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

Hierarchical Routing

- Hierarchical routing reduces the work of route computation but may result in slightly longer paths than flat routing



Full table for 1A

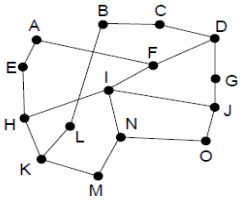
Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

Hierarchical table for 1A

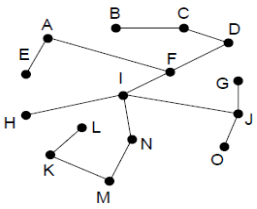
Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

Best choice to reach nodes in 5 except for 5C

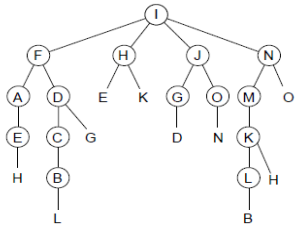
Broadcast Routing



Network



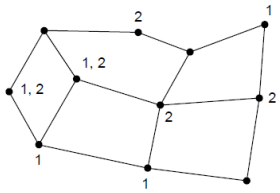
Sink tree for *I* is efficient broadcast



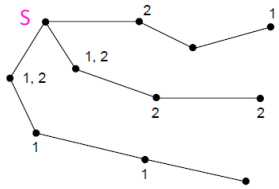
RPF from *I* is larger than sink tree

- Broadcast sends a packet to all nodes
  - RPF (Reverse Path Forwarding): send broadcast received on the link to the source out all remaining links
  - Alternatively, can build and use sink trees at all nodes

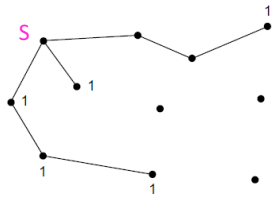
Multicast Routing – I – (Dense Case)



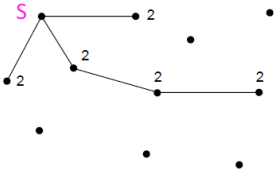
Network with groups 1 & 2



Spanning tree from source S



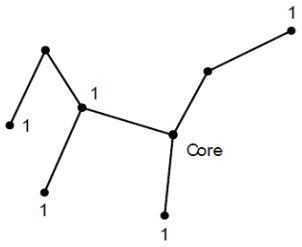
Multicast tree from S to group 1



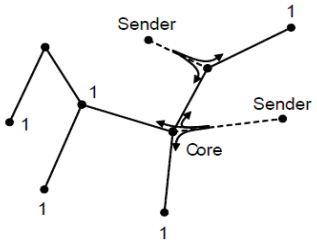
Multicast tree from S to group 2

Multicast Routing – II – (Sparse Case)

- CBT (Core-Based Tree) uses a single tree to multicast
  - Tree is the sink tree from core node to group members
  - Multicast heads to the core until it reaches the CBT
- p 1.



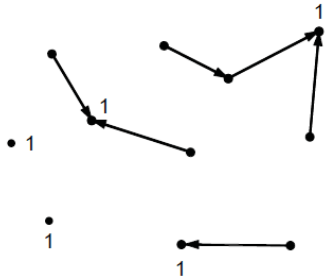
Sink tree from core to group 1



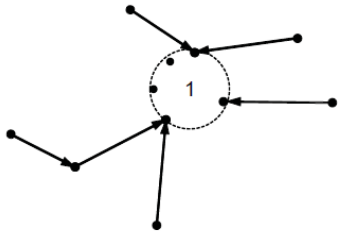
Multicast is send to the core then down when it reaches the sink tree

Anycast Routing

- Anycast sends a packet to one (nearest) group member
  - Falls out of regular routing with a node in many places



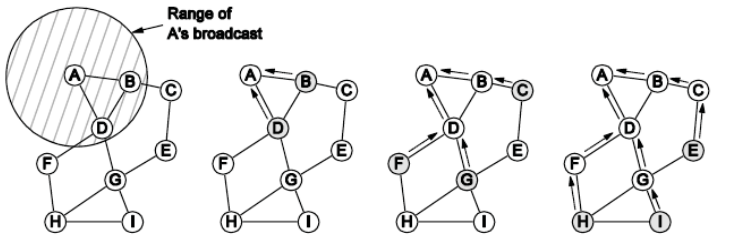
Anycast routes to group 1



Apparent topology of sink tree to "node" 1

Routing in Ad Hoc Networks

- The network topology changes as wireless nodes move
  - Routes are often made on demand, e.g., AODV (below)



A's starts to find route to *I*

A's broadcast reaches *B* & *D*

B's and *D*'s broadcast reach *C*, *F* & *G*

C's, *F*'s and *G*'s broadcast reach *H* & *I*

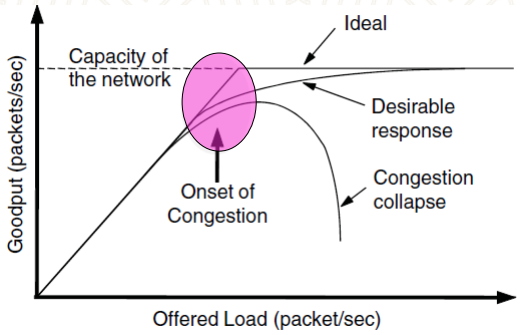
Congestion Control – I

- Handling congestion is the responsibility of the Network and Transport layers working together
  - We look at the Network portion here

- Traffic-aware routing »
- Admission control »
- Traffic throttling »
- Load shedding »

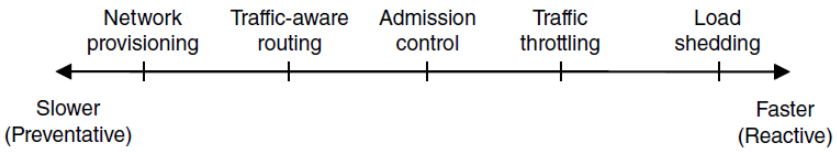
Congestion Control – II

- Congestion results when too much traffic is offered; performance degrades due to loss/retransmissions
  - Goodput (=useful packets) trails offered load



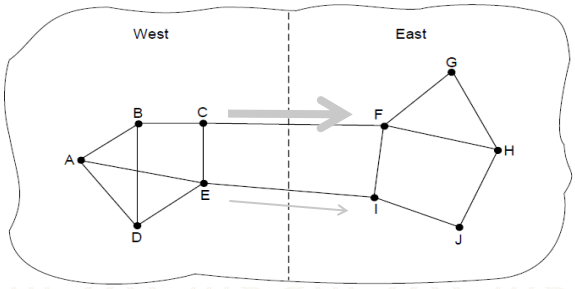
Congestion Control – III

- Network must do its best with the offered load
  - Different approaches at different timescales
  - Nodes should also reduce offered load (Transport)



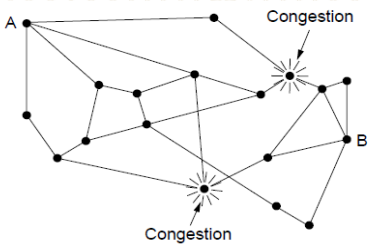
Traffic-Aware Routing

- Choose routes depending on traffic, not just topology
  - E.g., use *EI* for West-to-East traffic if *CF* is loaded
  - But take care to avoid oscillations

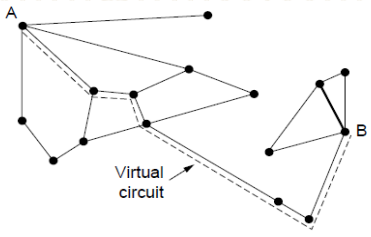


Admission Control

- Admission control allows a new traffic load only if the network has sufficient capacity, e.g., with virtual circuits
  - Can combine with looking for an uncongested route



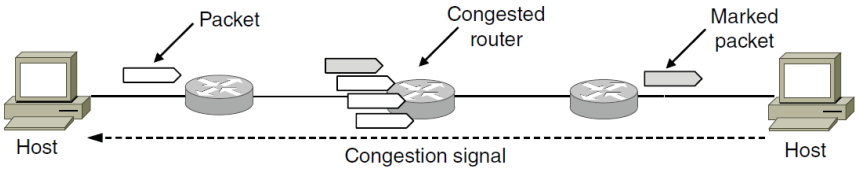
Network with some congested nodes



Uncongested portion and route *AB* around congestion

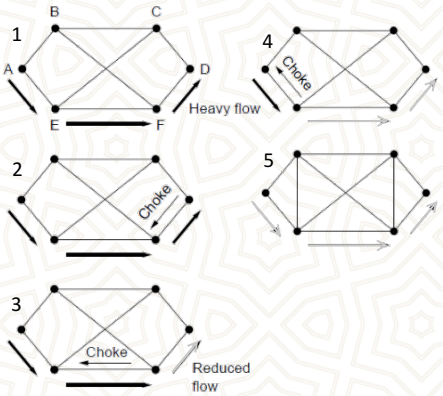
Traffic Throttling

- Congested routers signal hosts to slow down traffic
  - ECN (Explicit Congestion Notification) marks packets and receiver returns signal to sender



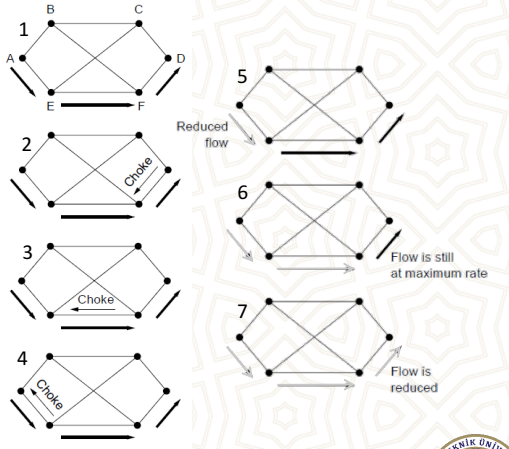
Load Shedding – I

- When all else fails, network will drop packets (shed load)
- Can be done end-to-end or link-by-link
- Link-by-link (right) produces rapid relief



Load Shedding – I

- End-to-end (right) takes longer to have an effect, but can better target the cause of congestion



Internetworking

- Internetworking joins multiple, different networks into a single larger network
  - How networks differ
  - How networks can be connected
  - Tunneling
  - Internetwork routing
  - Packet fragmentation

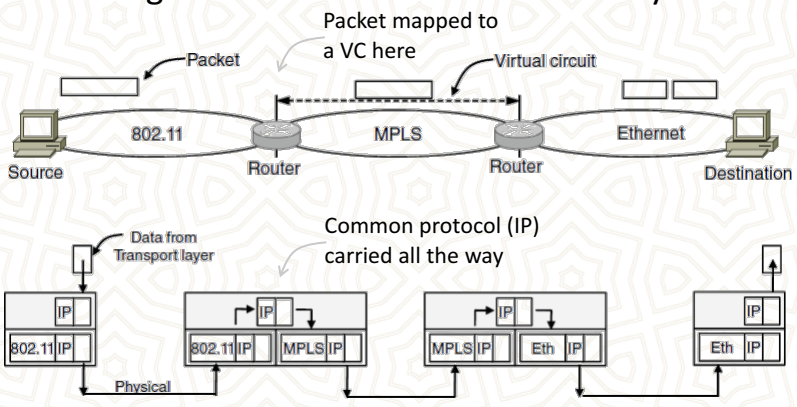
How Networks Differ

Item	Some Possibilities
Service offered	Connectionless versus connection oriented
Addressing	Different sizes, flat or hierarchical
Broadcasting	Present or absent (also multicast)
Packet size	Every network has its own maximum
Ordering	Ordered and unordered delivery
Quality of service	Present or absent; many different kinds
Reliability	Different levels of loss
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, packet, byte, or not at all

Differences can be large; complicates internetworking

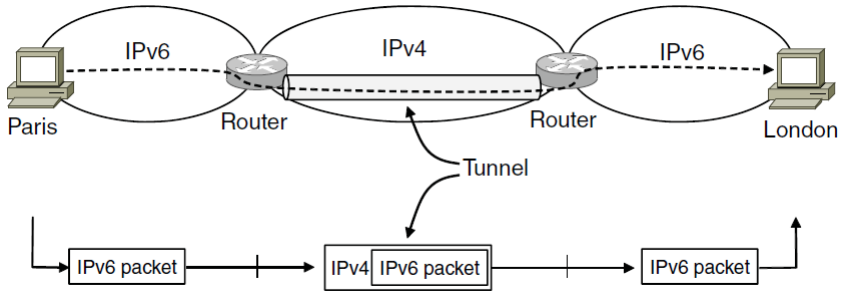
How Networks Can Be Connected

- Internetworking based on a common network layer – IP



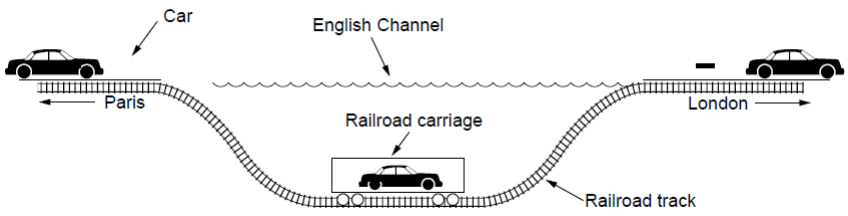
Tunneling – I

- Connects two networks through a middle one
  - Packets are encapsulates over the middle



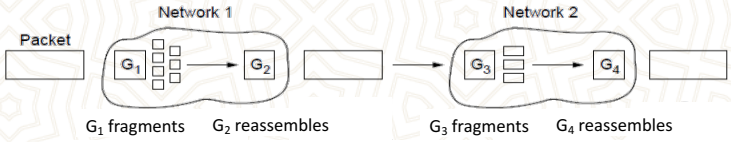
Tunneling – II

- Tunneling analogy:
  - tunnel is a link; packet can only enter/exit at ends

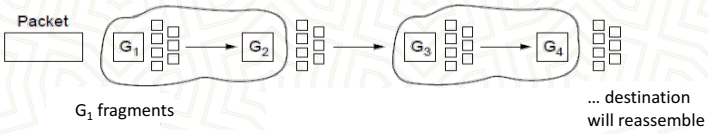


Packet Fragmentation – I

- Networks have different packet size limits for many reasons
  - Large packets sent with fragmentation & reassembly

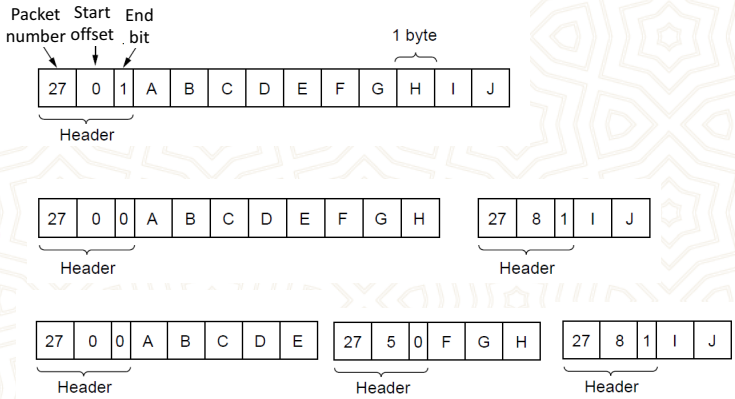


Transparent – packets fragmented / reassembled in each network



Non-transparent – fragments are reassembled at destination

Packet Fragmentation – II



Packet Fragmentation – III

- Path MTU Discovery avoids network fragmentation
  - Routers return MTU (Max. Transmission Unit) to source and discard large packets