yz709-network-sup2

Architecture

Question 1

Prior to the Internet, wide-area networks were joined together at the level of application protocols, using gateways. Explain why this approach limited application development.

- Application development was more complex without layering and abstraction because it needed to implement all sorts of underlying network technology the end-users would probably use. For example, there are various transport protocols (e.g., TCP and UDP) and data link protocols (e.g., Ethernet, wireless and point-to-point links).

Comments:

Question 2

Explain how the design of the Internet protocol, i.e. IP, addressed this problem of application development. You should explain how the term “hourglass model" describes IP's approach to network layering.

- IP has interoperability, it defines a common method for exchanging packets among networks; on top of it resides various transport protocols in the transport layer, each offering a different channel abstraction to applications, at the bottom, there are various

data link protocols in the link layer, ranging from Ethernet to wireless to the single point-to-point network.

- This allows the Internet to adapt rapidly to new user demands and changing technologies because new applications only need to implement once according to the abstraction IP/network layer provided.

💡 Comments:

Question 3

The design of IP makes explicit provision for fragmentation, i.e. the ability to split an individual packet into pieces during its journey across the network. By considering the hourglass model, suggest why this feature is essential.

- Because different network technology has different maximum frame sizes, the IP datagram needs to fit in the payload of the link-layer frame, so we define a maximum transmission unit (MTU) for each network technology (e.g., Ethernet, 802.11).
- In addition, higher-level transport protocols such as UDP and TCP have different segment sizes as well. In the hourglass model, since IP has the property of interoperability, it defines a common way for exchanging packets among networks, so the packet/segment size cannot be fixed on all networks.

💡 Comments:

Question 4

Comparing (data) delays in packet and circuit-switched networks: Compare the time it takes to transfer a file of data under circuit-switching and packet-switching. Consider a network consisting of n links in a row, each of bandwidth B and latency L. [Note: I'd like to see your work on this, so it might be easier for you to work it out by hand and either scan or photograph (with camscanner!) the page.]

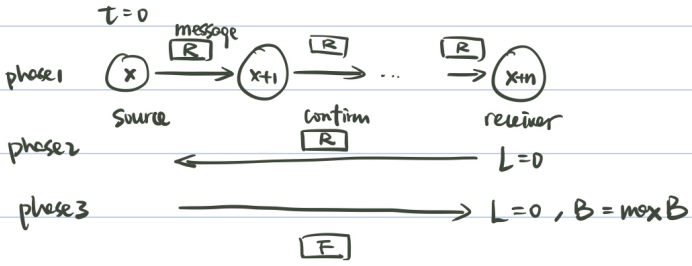
1. Circuit-switching: At time $t = 0$, the first node sends out a circuit reservation packet (of size R) which is sent to the second node, which then receives the full packet and then forwards it to the next node. This is continued at each node until the reservation packet arrives at the last node (after traversing n links). After this reservation message is processed at the last node (the destination), the last node sends back a reservation confirmation message (also of size R) back to the first hop. Because the circuit is

established before this confirmation is sent, this packet must not be processed at each node; instead, the bits flow through the nodes without delay. Once the confirmation message is received at the first node (the source), the source immediately starts sending the file (which is of size F) at the full bandwidth of the link. Note that when the file is transferred, the data is not stored-and-forwarded at any of the intermediate nodes but is just passed through without delay. Also, ignore the teardown message, since it is only sent after the file arrives.

a. Assuming no problems in transmission along the way, at what time does the last bit of the file arrive at the last node (the destination)?

n links, each bandwidth B, latency L

1. circuit switching: has fixed bandwidth B



phase1: establish the connection

$$t_1 = T_{transmission} + T_{delay} = n\frac{R}{B} + (n-1)L$$

phase2: confirmation message

$$t_2 = T_{transmission} + T_{delay} = n\frac{R}{B} + 0 = \frac{nR}{B_2}$$

phase3: sending the file

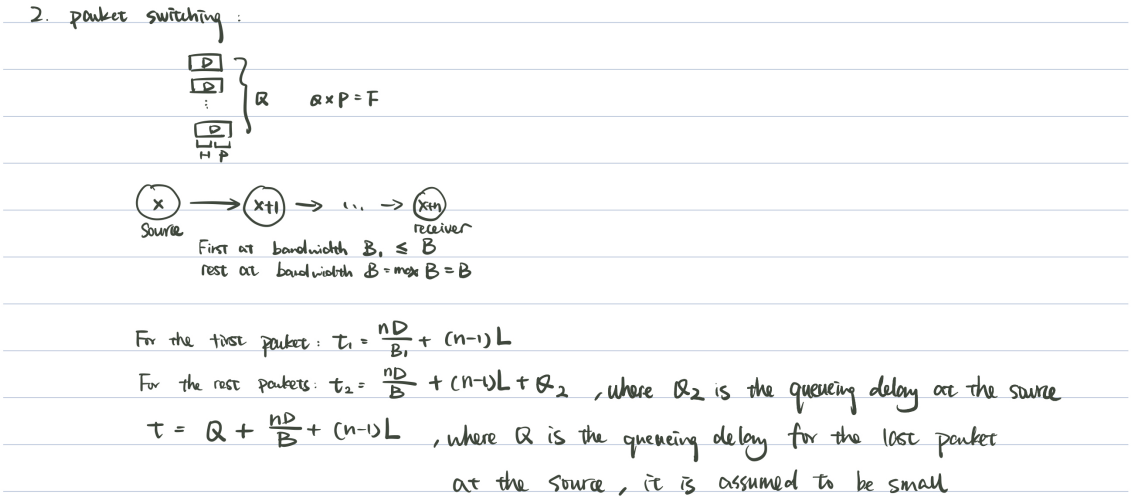
$$t_3 = T_{transmission} + T_{delay} = \frac{nF}{B} + 0 = \frac{nF}{B}$$

$$t = t_1 + t_2 + t_3 = n\left(\frac{R}{B} + \frac{R}{B} + \frac{F}{B}\right) + (n-1)L = \frac{n(2R+F)}{B} + (n-1)L$$

2. Packet-switching: Here the file is broken into Q packets of size D, each with header size H and payload size P. Since the entire file must be carried, Q x P = F. At time *t* = 0, the source (the first node) sends the first packet, which is stored-and-forwarded at each of the subsequent nodes until it reaches the destination (the last node). As soon as the source finishes sending the first packet, it sends the second packet (at full link

bandwidth). Note that the source does not wait until the first packet arrives at the next node before starting the next transmission, it starts sending the next packet as soon as it has finished transmitting the previous packet. We assume that a node can immediately send a packet out on the next link as soon as the last bit has arrived from the previous link (i.e., there is no time required to process the packet before sending it on the next link)

a. Assuming no packet drops or other errors, at what time does the last bit of the file arrive at the destination?



3. In the following questions, refer to cases where some quantities are big. By that we mean consider the limit where that quantity becomes infinitely large or infinitesimally small. Note that some quantities are linked (i.e., if the payload P gets smaller, the number of packets Q must get larger to keep Q x P = F). For each question, the answer could be either: circuit-switched is faster, or packet-switched is faster. Even if you didn't get the formulae above completely correct, you should understand how these perform relative to each other in the limit. Use this as a way to check your answers for the two previous questions.

   a. If the file size F is very large, which is faster? (Assume that the header size H has not changed.)

      - If file size F is very large, then the transmission time is the dominant term, and the latency is negligible, in that case, packet switching, which does not require any setup overhead and packets can transmit in-fly simultaneously, would be much faster.

   b. If the payloads P become small (but the header size remains constant), which is faster?

- Payload P becomes small, then Q becomes large, hence the packet head overhead becomes a dominant term in packet switching, thus circuit switching is much faster as it is not affected by the payload size P

c. If the bandwidth B is very large, which is faster? And by what ratio (in the limit)?

- If bandwidth B is very large, then the transmission time is negligible and the latency is the dominant term, in that case, circuit switching which requires much less latency would be faster, at a ratio of $\frac{(n-1)L}{Q+(n-1)L}$ compared to the latency (queueing latency + transmission latency) of packet switching.

Comments:

Data Link Layer

Question 5

Multiple access multiplexing in local area networks:

1. Give examples of multiple access and point-to-point links.
   - Multiple access: (1) Shared wire, e.g., coax cabled Ethernet; (2) Shared radio frequency, e.g., 802.11 WiFi and satellite
   - Point-to-point links: (1) telephone call, (2) microwave relay links, (3) leased wire
2. Explain [wired] Ethernet's use of *carrier sense, collision detection*. As part of your explanation, explain how the probability of collision on retransmission is minimised.
   - Carrier sense: all nodes can distinguish between an idle and a busy link to defer transmission until the link is idle
   - Collision detection: a transmitting adaptor detects collision by sensing transmissions from other adaptors while it is transmitting a frame. So in the worst case, an adaptor sends an RTT amount of data before receiving the collision signal.
   - Minimise the probability of collision using exponential backoff:
     - When a collision is detected, the adaptor sends a jamming sequence then stops transmission and waits for a random time interval before retransmitting.
     - This random time interval follows an exponential backoff, so the more the collisions, the longer the sender would wait, the time interval increases exponentially.

3. Why is collision detection not sufficient for wireless communication, and how does collision avoidance fix this deficiency?
    - Problems:
        - Half-duplex: wireless nodes cannot transmit and receive at the same time, so they cannot receive a collision signal whilst it is transmitting
        - A lack of global collision because of the hidden terminal and exposed terminal problems
            - **Hidden terminal problem**: If A and C are hidden from each other (i.e., A and C are both within the range of B, but not each other), their signal can collide at B without awareness
            - **Exposed terminal problem**: If B and C are exposed to each other's signal (i.e., B can send to A and C; C can send to B and D), there is no interference of B transmits to A while C transmits to D, but C can not transmit in the CSMA/CD algorithm
    - Collision avoidance with RTS/CTS fixes this deficiency:
        - the sender (e.g., C) sends an RTS (ready-to-send) to the receiver, the receiver (e.g., B) sends back CTS (clear-to-send) if the packet is received, this CTS can be heard by hidden terminals (e.g., A), so all terminals within the range will wait for a period indicated in the CTS packet and a random time period before sending.
        - using RTS/CTS, in the exposed terminal problem, when the sender (e.g., B) sends RTS to the receiver (e.g., A), the exposed terminals (e.g., C) do not have to wait because it heard RTS from B but no CTS, so can assume the destination (A) is out of range, so sending packets from C to other nodes will not affect node B.

Comments:

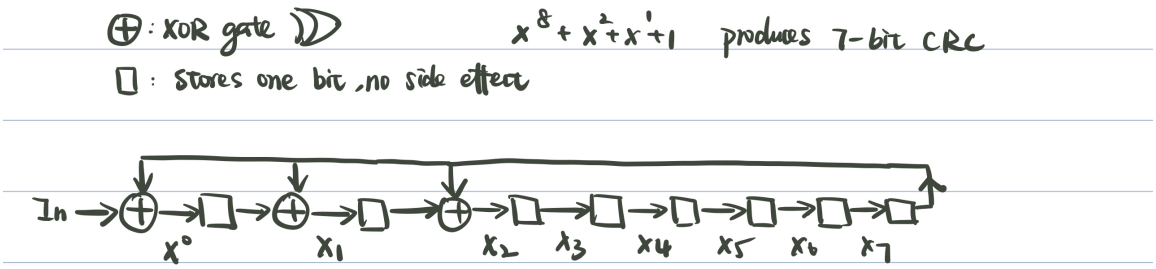
Question 6

CRCs:

1. What are the benefits of CRCs compared to other error detection mechanisms?
    - More powerful and reliable because it can detect various kinds of errors, including the 2-bit flip errors that the parity method is not able to detect (i.e., flip one bit from 1 to 0 and another one from 0 to 1)

- The hardware overhead (i.e., number of redundant bits attached to the message) is not high compared to the multiple copy method, which needs to send $m \times n$ bits for sending n bits of information with m copies.

2. Draw a shift register for the CRC-8 polynomial x^8 + x^2 + x^1 + 1.
   - The final CRC is stored in the shift registers □



3. Given the CRC polynomial x^3 + x^2 + 1, what CRC value is transmitted for the message 10101010?
   - The CRC being transmitted is `110`, the encoded message `10101010 110` would be sent to the receiver end; if no errors occurred, it could be divisible by the shared 4-bit divisor `1101`

```
4-bit divisor: 1101
Message M = 10101010
calculate 3-bit CRC = rem(M⨯2^3, 1101), using binary division = 110
         11011 110
     ----------------
1101| 10101010 000
      1101
      ----
      01111
       1101
       ----
       00100
        0000
        ----
        01001
         1101
         ----
         01000
          1101
          ----
          0101 0
           110 1
           -----
           011 10
            11 01
            -----
            00 110
             0 000
             -----
               110
```

4. Show that a flip of the final bit can be detected.

```
Message received M' = 10101011
CRC = 110
since rem(M'⨉2^3 + CRC, 1101) != 0, the message contains some errors

         11011 111
     ----------------
1101| 10101011 110
      1101
      ----
      01111
       1101
       ----
       00100
        0000
        ----
        01001
         1101
         ----
         01001
          1101
          ----
          0100 1
           110 1
           -----
           010 01
            11 01
            -----
            01 000
             1 101
             -----
               101
```

Comments:

Question 7

Digital channels, modulation, and transmission:

1. Distinguish between baud rate and bit rate.
    - Baud rate is the number of signals sent per second; the signals are indivisible data elements encoded, for example, 4-bits per signal.
    - Bit rate is the number of bits sent per second. If a signal includes x bits, then the baud rate is $\frac{1}{x}$ of bit rate.
2. Give an example where baud rate > bit rate and one where bit rate > baud rate and justify your answer?

- Baud rate > bit rate: suppose a bit 0 represents a transition from high voltage to low, a bit 1 represents low voltage to high, then we can use 1 bit to convey two signals, hence if we send x bits over the channel within a period of t seconds, then bit rate x/t bits per second, but the baud rate is $2x/t$ signals per second.
- Bit rate > baud rate: suppose we have 4 possible voltages, so we need two bits to represent each signal ($2^2 = 4$ voltages); hence if we send x bits over the channel within a period of t seconds, the bit rate is x/t bits per second, but the baud rate is $x/2t$ signals per second.

3. What synchronous transmission problem(s) does Manchester encoding solve?
    - It prevents the case where consecutive 0s (low voltages) or consecutive 1s (high voltages) stayed for an extended period by mapping 0 to a low-to-high transition and mapping 1 to a high-to-low transition.
4. Why might a line scrambler (e.g., 64B/66B) be preferred over a block code such as 4B/5B (or 8B/10B)?
    - Block codes have a lower efficiency compared to a line scrambler, 4B/5B and 8B/10B are 80% efficient because for sending 4(or 8) bits of information, we need to send in a total 5(or 10) bits.
    - While in a line scrambler, it has an efficiency of $64/66 \approx 97\%$

> 💡 Comments:

Question 8

Code division, multiple access:

1. What is a code?
    - A unique chipping sequence per client node is used to encode and decode data at the sender and receiver ends.
2. Give 2 pros and 2 cons of using CDMA.
    - Pros: (1) allow multiple users to transmit simultaneously with minimal interference, (2) it is hard to eavesdrop on the data because it is often a superposition of the data and multiple chipping sequences.
    - Cons: (1) the code/chipping sequence gets very long with a large user base, (2) the chipping sequence is hardware overhead because the receiver needs to know the chipping sequence of the sender in order to decode the data.

3. Two transmitters, A and B, both want to transmit a four-bit message simultaneously using CDMA. Transmitter A has code 10010111 and message 1001. Transmitter B has code 00111101 and message 0011. [Each bit is transmitted as the exclusive OR of the code sequence with the bit value.]

    a. Write down the bit sequences transmitted by A and B.

```
codeA = 10010111
dataA = 1001
encode data into signal using xor:
signal0 = (1,0,0,1) ⊕ (1, 0, 0, 1, 0, 1, 1, 1)
        = [(1, 0, 0, 1, 0, 1, 1, 1),
           (0, 1, 1, 0, 1, 0, 0, 0),
           (0, 1, 1, 0, 1, 0, 0, 0),
           (1, 0, 0, 1, 0, 1, 1, 1)]
=> sends 10010111 01101000 01101000 10010111

codeB = 00111101
dataB = 0011
encode data into signal using xor:
signal1 = (0,0,1,1) ⊕ (0, 0, 1, 1, 1, 1, 0, 1)
        = [(1, 1, 0, 0, 0, 0, 1, 0),
           (1, 1, 0, 0, 0, 0, 1, 0),
           (0, 0, 1, 1, 1, 1, 0, 1),
           (0, 0, 1, 1, 1, 1, 0, 1)]
=> sends 11000010 11000010 00111101 00111101
```

    b. For the rest of this question, it is helpful to remember that the bit sequence is transmitted as a wave, with highs and lows, so you can think of the bit sequences as voltage levels of +1 (for 1) and -1 (for 0). Write down the voltage levels seen by a receiver, assuming the messages from A and B perfectly align.

```
codeA = 10010111
dataA = 1001
(1) encode data into 1s and -1s: 1001 -> (1,-1,-1,1)
(2) encode code into 1s and -1s: 10010111 -> (1, -1, -1, 1, -1, 1, 1, 1)
(3) encode data into signal using xor:
signal0 = (1,-1,-1,1) ⊕ (1, -1, -1, 1, -1, 1, 1, 1)
        = [(1, -1, -1, 1, -1, 1, 1, 1),
           (-1, 1, 1, -1, 1, -1, -1, -1),
           (-1, 1, 1, -1, 1, -1, -1, -1),
           (1, -1, -1, 1, -1, 1, 1, 1)]

codeB = 00111101
dataB = 0011
(1) encode data into 1s and -1s: 0011 -> (-1,-1,1,1)
(2) encode code into 1s and -1s: 00111101 -> (-1, -1, 1, 1, 1, 1, -1, 1)
(3) encode data into signal using xor:
signal1 = (-1,-1,1,1) ⊕ (-1, -1, 1, 1, 1, 1, -1, 1)
        = [(1, 1, -1, -1, -1, -1, 1, -1),
           (1, 1, -1, -1, -1, -1, 1, -1),
           (-1, -1, 1, 1, 1, 1, -1, 1),
           (-1, -1, 1, 1, 1, 1, -1, 1)]

signal = signal0 + signal1 (a superposition)
```

```
       = [(2, 0, -2, 0, -2, 0, 2, 0),
          (0, 2, 0, -2, 0, -2, 0, -2),
          (-2, 0, 2, 0, 2, 0, -2, 0),
          (0, -2, 0, 2, 0, 2, 0, 2)]
```

c. Show that the original messages of both A and B may be recovered.

```
signal = signal0 + signal1 (a superposition)
       = [(2, 0, -2, 0, -2, 0, 2, 0),
          (0, 2, 0, -2, 0, -2, 0, -2),
          (-2, 0, 2, 0, 2, 0, -2, 0),
          (0, -2, 0, 2, 0, 2, 0, 2)]
codeA = 10010111 -> (1, -1, -1, 1, -1, 1, 1, 1)
decodeA = (inner product of signal ⨯ codeA)/len(codeA)
        = [8, -8, -8, 8]/8 = [1, -1, -1, 1]
        -> has meaning 1001

codeB = 00111101 -> (-1, -1, 1, 1, 1, 1, -1, 1)
decodeB = (inner product of signal ⨯ codeB)/len(codeB)
        = [-8, -8, 8, 8]/8 = [-1, -1, 1, 1]
        -> has meaning 0011
```

💡 Comments:

Admin

How long did this take (reading + assignments)?

- Question 1 - Question 3: 20 mins
- Question 4: 20 mins
- Question 5 & 6: 50 mins
- Question 7: 15 mins
- Question 8: 30 mins
- In total 135 mins

💡 Comments:

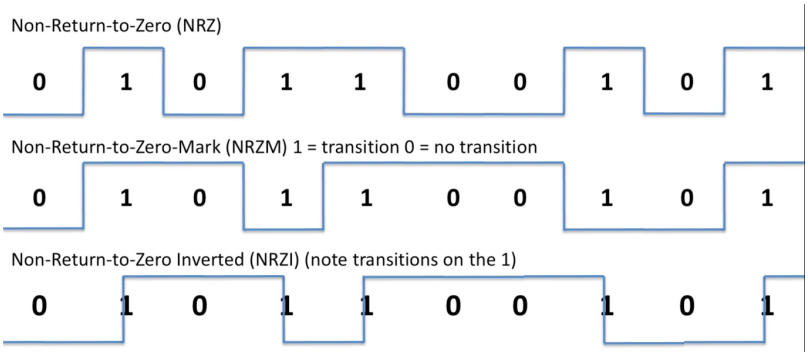
Notes on chapter 2 & 3

Data-link layer

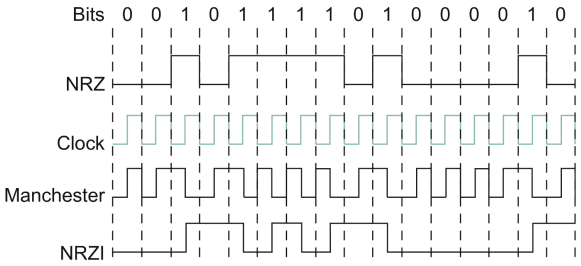
- A link layer is a combination of hardware (network adaptor), software (to control timing, and system addresses) and firmware (to represent MAC address).
- There are five central problems on the data link layer: (1) bits encoding to signals, (2) framing problem (network adaptor needs to recognise among a bit stream where the frame starts and ends), (3) error detection problem, (4) reliable link, (5) media access control for multiple-access links (how all nodes got chances to transmit data) - Ethernet with CSMA/CD & Wi-Fi with CSMA/CA.
- A link is some physical material that can propagate signals (e.g., electromagnetic radiation), so in order to send bits over the link, we have to encode bits into signals.
- A network adaptor is a hardware that connects a node to a link, it contains a signalling component that encodes bits into signals (and add error checking bits) at the sending node and decodes at the receiving node. Signals travel over a link between two signalling components.

- Encoding scheme:
  - Non-return to zero (NRZ) - map 0 to low signal, 1 to high signal → P(consecutive 0s or 1s means the signal stays low or high for an extended period):
    - Consequence(1) baseline wander (receiver uses the signal average to distinguish between signals, consecutive 0s or 1s cause the average to change, making signal detection hard); (2) lost synchronisation/clock recovery (receiver uses received signal to sync the clock with the sender, but consecutive steady signals make it hard to detect clock edges)
  - Non-return-to-zero-mark (NRZM) - map 1 to a transition, 0 to stays the same → P(consecutive 0s means signal stays low for an extended period) → still cause synchronisation problem

▼ Non-return to zero inverted (NRZI) - → transition occurred at the beginning of the bit interval → 1; no transition at the beginning of the bit interval → 0



▼ Manchester encoding - synchronise clock encoding technique, map 0 to a low-to-high transition, map 1 to a high-to-low transition → P(two states represent a bit because each bit is a transition, so double the bandwidth/data rate required to transmit the same amount of data, otherwise will be 50% efficient) + P(Additional circuitry or firmware needed to generate and interpret the Manchester-encoded data)



▼ Block encoding (4B/5B) - every 4 bits of actual data are encoded in a 5-bit block (scheme makes sure ≤ 1 leading 0 and ≤ 2 tailing 0s, so 5-bit block has > 3 consecutive 0s), transmitted using NRZI(solved problem of consecutive 1s) → P(80% efficient, but more efficient than Manchester encoding and has no synchronisation issues)

- For 16 possible data symbols, we have $2^5 = 32$ patterns can be selected, so 16 valid ones are selected
- For the 16 other ones, where 11111 is used when the line is idle, 00000 is used when line is dead, 00100 means halt, remaining 7 out of 13 is invalid, the remaining 6 valid ones are used as control symbols.

Table 2.2 4B/5B Encoding

4-Bit Data Symbol	5-Bit Code
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

- Line coding scrambling is used to provide synchronisation without increasing the number of bits needed to convey information - message xor with a shared scrambling sequence between sender and receiver
  - (1) based on the statistical idea that a message has an equal number of zeros and ones; (2) can add secrecy by securely distributing the scrambling sequence between sender and receiver; (3) not reliable at the start, so needs to send redundant bits initially to indicate "start of the frame"
- CDMA (Code division multiple access) - each client node uses a unique chipping sequence, all users share some frequency, but encode and decode the data sent or received by XOR with the chipping sequence.
  - (1) allows multiple users to transmit simultaneously with minimal interference; (2) used in cellular and satellite standards; (3) chipping sequence gets long with large user base

---

- Framing encapsulates datagram into frames, adding a header and a trailer. MAC addresses are used in headers to identify source and destination

▼ Framing problem (network adaptor needs to recognise among a bitstream where the frame starts and ends)

Name	Binary Synchronous Communication Protocol (BSCP) & Point-to-point Protocol (PPP)	DECNET's DDCMP	High-level Data Link Control (HDLC) Protocol
Type of protocol	Byte-oriented	Byte-oriented	Bit-oriented
Principle	- special sentinel character to indicate frame start and endpoints - character stuffing, insert escape character DLE before using sentinel characters in the frame body	- include a number of bytes in a frame in the header	- denotes both the beginning and end of a frame with bit sequence `01111110` - other data is coded to ensure it never contains more than five 1s - bit stuffing, every time consecutive five 1s are transmitted, sender inserts a bit 0 at the end; depending on the next bit - receiver decides (1) end-mark if receive 0, (2) error if receive 1

Name	Binary Synchronous Communication Protocol (BSCP) & Point-to-point Protocol (PPP)	DECNET's DDCMP	High-level Data Link Control (HDLC) Protocol
Problem	- more characters needed (extra DLE) - transmission error corrupt the ETX(end of text) field	- framing error: transmission error corrupts the header field, cause back-to-back frames incorrectly received	- receiver might fail to receive two consecutive frames - framing error can go undetected when an entire end-of-frame pattern is generated by error

- Bit errors are caused by electrical interference/signal attenuation, thermal noise, digitisation (converting between digital and analogue) and signal distortion.

▼ Error detection method - goal is to maximise the probability of detecting errors using a small number of redundant bits:

	Multiple copies	Parity	Checksum	Cyclic redundancy check (CRC)
Principle	- transmitting multiple same copies of data, compare whether they are the same	- for each 7-bit code block, append one extra parity bit to ensure even number of #1s	- sum all bits together, append this sum to the message	- an CRC is appended to the end of the message M, so that the result becomes exactly divisible by a chosen k-bit divisor P (e.g., 3-bit divisor 101) - (1) T = add k zeros to the end of M, (2) CRC = rem(T, P), (3) send the message T + CRC (can get it by XOR T with P), (4) receiver check received message can be divisible by P
Advantages		- low hardware overhead because only few redundant bits - easy to implement	- low hardware overhead because only few redundant bits	- more powerful than parity as it can detect various kinds of errors, including 2-bit errors

	Multiple copies	Parity	Checksum	Cyclic redundancy check (CRC)
Problems	- m * n redundant bits for sending n bits of information (m = # copies) - Errors might corrupt all copies	- not reliable as we cannot detect an even number of errors (e.g., two-bit errors)	- cannot detect errors where one bit (0 → 1), one bit (1 → 0)	- more complex because it involves multiplication and binary division
Usage	-	used by BISYNC protocol for transmitting ASCII chars	used by several internet protocol	used in all data-link level protocols

- Error correction: (1) notify the sender to resend, (2) reconstruct the correct message at receiving end using Forward Error Correction (FEC) - replace erroneous data with its closest error-free data

	Error detection	Error correction
Pros	Fewer check bits, only resent when an error occurs	No need to re-send, so no additional bandwidth and latency overhead
Cons	Need to re-send	Require a greater number of redundant bits for every transmission
Usage	(1) less noise, low probability of getting errors (2) easy and cheap for retransmission	(1) errors are quite probable (e.g., in the wireless environment) (2) cost of retransmission is too high

- Reliable transmission may be provided at the data link level, but more frequently provided at higher levels (e.g., transport layer or application layer)
    - All uses (1) acknowledgements (a control frame/header without data indicates received) and (2) timeouts (sender will retransmit the data frame if no ACK is received after a timeout)
        - A protocol can piggyback an ACK on a data frame if it will later send data frames
    - Algorithm 1 - **stop-and-wait**: (1) after transmitting one frame, the sender waits for an ACK before transmitting the next frame; (2) retransmit if no ACK received after timeout; (3) attach 1-bit sequence number (0 or 1), alternating between frames, so

the receiver can deduplicate copies, and the sender will not be confused about delayed ACK

- Problem: only one outstanding frame on the link at a time, far below the link capacity (i.e., delay $\times$ bandwidth)

▼ Algorithm 2 - **sliding window**: aims to (1) reliably deliver messages across the unreliable networks; (2) preserve the frame orders; (3) flow control (a feedback mechanism where the receiver can control the sender)

- Cumulative acknowledgement:(1) attach a sequence number for each frame; (2) receiver only accept frames within the range, send ACK for next expected frame, buffer frames with higher sequence number
  - Problem: when packet loss, can no longer keep the pipe full because cannot advance the sliding window → Solution (Selective acknowledgement)
- Selective acknowledgement: ACK for all received frames individually, so the sender can send more frames to keep the pipe full even some packets are lost, but adds complexity to the implementation.

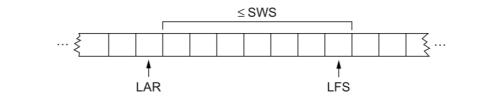


FIGURE 2.20 Sliding window on sender.

FIGURE 2.21 Sliding window on receiver.

Two invariants:
(1) LFS(last frame sent) - LAR(last ack received) ≤ SWS (sender window size)
(2) LAF(largest acceptable frame) - LFR(last frame received) ≤ RWS(receiver window size)

- Media access control (MAC) protocols are distributed algorithm that determines how nodes share channel (i.e., when nodes can transmit):

	Channel partitioning MAC protocols	Random access MAC protocols	Taking turns MAC protocols

	Channel partitioning MAC protocols	Random access MAC protocols	Taking turns MAC protocols
Principle	Divide the channel into smaller pieces (time slots/frequency/code) and allocate pieces to the node for exclusive use	- Channel is not divided, node can transmit packets at full data rate - Need to detect and recover from collisions - Wait a random time period after a collision and before sending (exponential backoff)	- Nodes take turns, but some may be allocated a longer turn
Typical algorithms	- Time division multiple access - Frequency division multiple access	- CSMA/CD in Ethernet - CSMA/CA in wireless	- ATDM in Bluetooth - Polling (primary node invites subordinate nodes to transmit in turn - Token passing (control token passed from one node to next sequentially)
Under Low Load	Inefficient, delay in channel access, 1/N bandwidth allocated even if only 1 active node	Efficient since a single node can fully utilise the channel	Efficient
Under High Load	Efficient and fair	Inefficient because of high collision overhead	Efficient

- In random access MAC protocols in a network with latency d:
  - The minimum frame size is $\geq 2d$ and maximum distance $\leq d$:
    - A sender can transmit at most $\mathrm{bandwidth} \times 2\mathrm{d}$ amount of data before receiving a collision signal from the receiver, so the sender will only see the collision signal at $2d$ after the sending time
  - The performance (efficiency) is $E \sim \frac{\frac{p}{b}}{\frac{p}{b}+Kd'}$ where the time spend transmitting a packet is $\frac{p}{b}$ (packet length over bandwidth) and the time wasted in the collision is proportional to distance d': Kd'
    - (1) For large packets and small distances $E \sim 1$; (2) As bandwidth increases, E decreases, hence high-speed LANs do not use CSMA/CD but use a switched network.
- CSMA/CD (Carrier sense, multiple access with collision detection) technology:
  - A media access control (MAC) method used in multi-access Ethernet for LAN: (1) carrier sense (all nodes can distinguish between an idle and a busy link) to defer transmission until the link is idle; (2) collision detection (a transmitting adaptor

detects collision by sensing transmissions from other adaptors while it is transmitting a frame)

- When a collision is detected, the adaptor sends a jamming sequence then stops transmission, waits for a random time interval before retransmitting
  - In the worst case, an adaptor sends an RTT amount of data before receiving the collision signal
  - Random time interval follows an **exponential backoff**

▼ Multi-access Ethernet uses CSMA/CD: (1) data transmitted by any host on that Ethernet reaches all the other hosts, (2) all hosts are competing for access to the same link

- The frame format is the interface of Ethernet (e.g., where to put the data, what size, what information, how to compute CRC), it can accommodate many changes, implementation is hidden behind the interface
  - Each host has a unique Ethernet address(also called MAC address), typically belongs to the adaptor (burned in the ROM)
- Frame must contain at least 46 bytes of data (including paddings) to allow collision detection

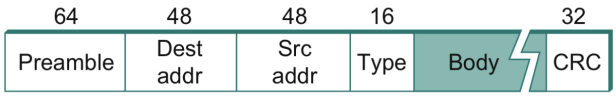


FIGURE 2.25 Ethernet frame format.

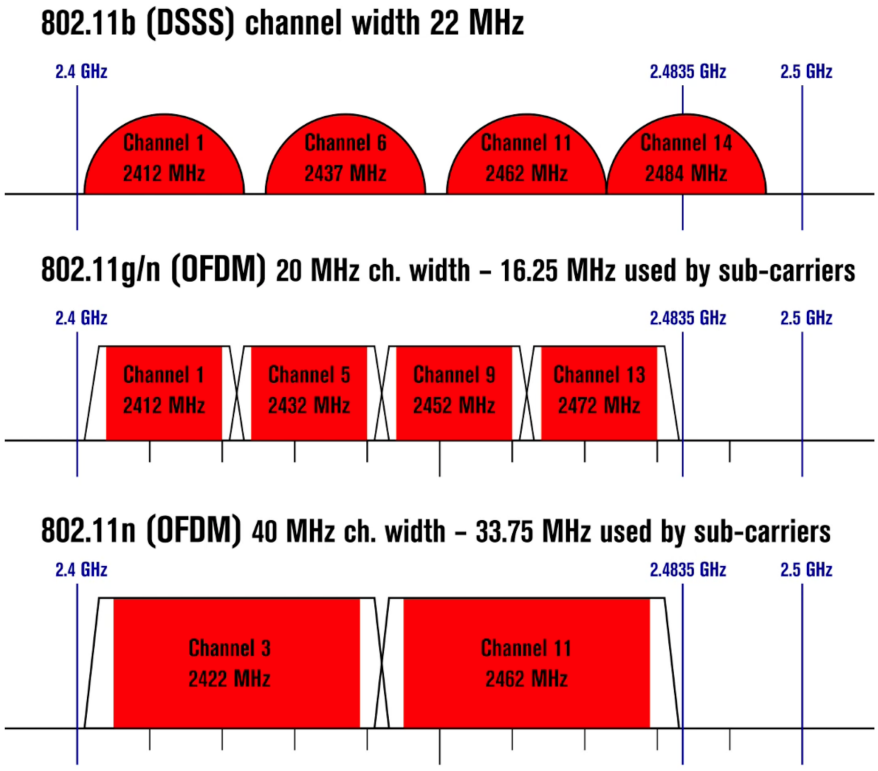
Preamble - allow receiver to synchronise with the signal
Type - demultiplexing key, identify higher-level protocol this frame should deliver
Preamble and CRC is attached and stripped by network adaptors

---

- Similarities and differences between Wireless and Ethernet:
  - Similarities include: (1) bit errors are of great concern - wireless concerns more because its unpredictable noise environment; (2) framing and reliability have to be addressed; (3) multi-access, so media access control is a central issue;
  - Differences: (1) power is a big issue for wireless because power is limited on mobile phones; (2) much harder to control who receives your signal over the air, so eavesdropping is an issue;
- Wireless links all share the same medium, so need to divide frequency and space effectively.

- Frequency bands that can penetrate the walls are suitable for communication, they are divided for (1) government use, (2) AM radio, (3) FM radio, (4) television, (5) satellite communication, (6) cellular phones. Exclusive use of a particular frequency in a particular geographic area can be allocated to an individual entity.
- Frequency spectrum can be partitioned into several channels, nodes within interference range can use separate channels. Channels must be independent/orthogonal to be used to avoid power bleeding from one channel into adjacent channels

▼ Spread spectrum: spread the signal over a wider frequency band, to minimise the impact of interference from other devices. Techniques include (1) frequency hopping: sender and the receiver uses the same pseudorandom number generator, signals are transmitted via an agreed random sequence of frequencies. (2) direct sequence: adds redundant data bits in transmission (xor with a chipping sequence), to tolerant loss of a single data bit.



Non-overlapping channels for 2.4 GHz WLAN (unregulated)
The US only standard is 802.11b (DSSS) channel, which has 3 independent and orthogonal channels (1, 6, 11)

- Wireless network topology:

	Wireless star topology	Mesh topology

	Wireless star topology	Mesh topology
Implementation	- A wireless base station sends radio waves that multiple client nodes can receive - Communication between client nodes via the base station.	- Messages are forwarded via a chain of peer nodes within the range
Property	- Single point of failure - Limited range - Only the base station needs hardware & software support	- Fault tolerance - Extensive range - Higher hardware & software cost in each client node, critical for battery-powered devices

- Similarities and differences between wireless technologies:
  - Similarities: all broadcast and multi-access medium, but signals sent by sender don't always end up at receiver intact

	Bluetooth (802.15.1)	Wi-Fi (802.11)	3G Cellular
Link length	10 m	100 m	Tens of kilometres
Data rate	2 Mbps (shared), low power consumption	54 Mbps (shared)	Hundreds of kbps per connection
Usage	Link a peripheral to a computer	Link a computer to a wired base	Link a mobile phone to a wired tower
Wired tech analogy	USB	Ethernet	DSL

- Wi-Fi (802.11) - aims to mediate access to a shared communication medium in a limited geographical area
  - Defines a number of different physical layers that operate in various frequency bands with a range of different data rates:
    - e.g., common 4 - `802.11a, b, g, and n`:(1) the license-exempt 2.4GHz frequency band has 802.11b defined, which provides data rate up to 11 Mbps; (2) the license-exempt 5GHz frequency band has 802.11a defined, which provides data rate up to 54 Mbps.
  - A base station might support multiple 802.11 standards and choose an appropriate data rate for a particular noise environment: (1) lower bit rate is easier when decoding transmitted signals in the presence of noise; (2) high bit rate gives more redundant information, implies higher resilience to bit errors, but lowering the effective data rate (as most are redundant).
    - Approach of picking a bit rate implemented by vendors: (a) estimate the bit error rate by directly measuring the signal-to-noise ratio (SNR) or estimating SNR by measuring how often packets are successfully transmitted and acknowledged; (b) a sender probe a higher bit rate by sending some packets to see if it succeeds.

    - Some devices become access pointers (AP), which set to a specific channel, host scan all the channels to discover the AP's and associated with the strongest one.
- Wi-Fi uses CSMA/CA (collision avoidance using RTS/CTS) for transmission:

Problems with the CSMA/CD algorithm implemented in Ethernet
- **Half duplex**: Wireless nodes cannot transmit and receive at the same time because power generated by the transmitter is higher than any received signal.
- **Hidden terminal problem**: If A and C are hidden from each other (i.e., A and C are both within the range of B, but not each other), their signal can collide at B without awareness
- **Exposed terminal problem**: If B and C are exposed to each other's signal (i.e., B can send to A and C; C can send to B and D), there is no interference of B transmits to A while C transmits to D, but C can not transmit in the CSMA/CD algorithm

    - (1) carrier sense (all nodes can distinguish between an idle and a busy link) to defer transmission until the link is idle
        - P(but because of the hidden terminal problem, waiting for absence does not guarantee idle link) → S(use an explicit ACK from the receiver to the sender if the packet is successfully received)
    - (2) collision avoidance using RTS/CTS - to solved the problem caused by a lack of global collision:
        - (1) the sender sends an RTS (ready-to-send) to the receiver, (2) the receiver sends back CTS (clear-to-send) if the packet received → **the CTS can be heard by a hidden terminal;** (3) so all terminals within the range will wait for a period indicated in the CTS packet + random period before sending
        - (4) If two nodes send RTS frames simultaneously, both frames dropped, the sender would notice the timeout for CTS and wait an amount of time defined by an exponential backoff algorithm
        - (5) After a successful RTS-CTS exchange, the sender sends its data packet and the receiver sends ACK
        - (6) If a client node hears RTS but not CTS, then can assume the destination is out of range → but this causes problems when a CTS is lost

Network layer - part 1

- Forwarding: the process of sending a packet from input out to correct output, based on a forwarding table built by routing
- Routing: the process of building up the tables that allow the correct output for a packet to be determined

- Switches are used to connect same network links; Routers are used to connect disparate network links.

	Router	Switch
Layer	Network layer	Data link layer
Definition	Device that interconnect different type of links	Device that interconnect same type of links
Usage	LAN, MAN	LAN
Type	- Adaptive routing - Non-adaptive routing	- Circuit switching - Packet switching
Handle header	Does strip the header, adds a new link layer header, then sends to the destination	Does not strip the header, store-and-forward frames based on MAC address

- A switch can form a wired star topology by having a physical point-to-point link with each host: (1) a large networks can be built by interconnecting a number of switches; (2) all hosts can transmit to the switch at full speed using their direct link; (3) more scalable than shared-media networks because the ability to support many hosts
- A switch forwards packets by:

Approach	Procedure	Property
Datagram (connectionless)	- every packet contains full destination address - for any arriving packet, the switch consults the forwarding table to find the correct port number connects to the destination link - forwarding table is updated dynamically by routing	- forward immediately via forwarding table - sender has no idea whether forwarding would be successful - packet forwarded independently - robust to switch or link failure
Virtual circuit (connection-oriented)	- connection setup phase: an entry in the VC table set up to uniquely identify the connection ends (small identifier rather than the full address) - data transfer phase: hop-by-hop flow control, to allocate resources fairly to each node	- virtual connection between sender and receiver needs to be set up before forwarding (introduce RTT delay) - each packet only contains a small identifier - sender knows a lot about the network before sending - robust to switch or link failure

- Internet protocol is invented to deal with interconnection of disparate network types, to build scalable, heterogeneous internetwork.
    - It runs on all the nodes (hosts & routers) in a collection of networks and define the infrastructure that allows these nodes and networks to function as a single logical internetwork
- The IP service model is made to be undemanding for underlying network technology (e.g., simple routers) with (1) global addressing scheme, (2) unreliable datagram data delivery model.

- Global addressing scheme: global uniqueness + hierarchical addressing + route aggregation & subnetting + ARP
  - Hierarchical addressing: `IP address` = `network part` (identify the network the host attached) + `host part` (unique host/interface identifier on that network)
    - A router connects to two networks needs to have two IP addresses, one for each interface
  - Subnetting used in CIDR: (1) allocate a single IP network-part number to several physical networks that are close to each other, (2) A subnet mask is used to determine whether a host is on the local subnet or a remote network, (3) **route aggregation**: not all parts of the Internet see the same routing information, enable routing system scaling.
  - Address resolution protocol (ARP) for address translation: (1) each host on a network dynamically build a table of mappings between IP addresses and link-level addresses, (2) because physical devices only understand link-level addresses (e.g., a 48-bit Ethernet address)
  - Process:
    - (1) each host is allocated an IP address and a subnet mask (same across hosts on the subnet), the bitwise AND gives the subnet number of the host;
    - (2) when sending a packet, do bitwise AND on its subnet mask and destination IP address, if equals to the subnet number of the sender, then the destination is on the same subnet; otherwise send to a router to forward to the correct subnet via a modified forwarding table;
    - (3) once on the same subnet, the host checks mappings in ARP cache. If no entry, then broadcast the ARP query onto the network (because link-level support broadcast). A matching receiver would response, ARP table updated.

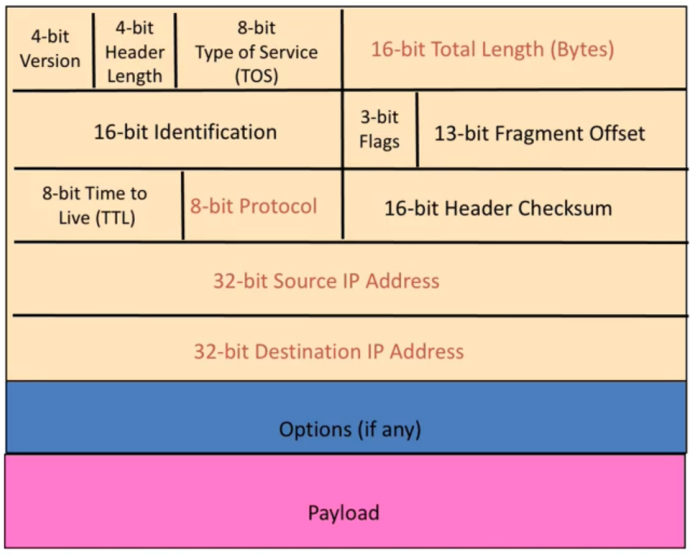
	Classful IP addressing	Classless Inter-Domain Routing (CIDR)
Definition	Divides IP addresses into fives groups (A, B, C, D, E) each with a different format, hence different # networks in that class and different # hosts that can be supported - e.g., A (0 + 7-bit network + 24-bit host), C(1 + 1 + 0 + 21-bit network + 8-bit host (255 hosts))	Use classless subnetting - assign blocks of contiguous addresses to a subnet - the network size can depends on the prefix `/x` - e.g., 192.4.16/20 for all networks 192.4.16 through 192.4.31, the top 20 same bits used as the network part

	Classful IP addressing	Classless Inter-Domain Routing (CIDR)
Problem → Solution	(1) not sufficient, (2) for a lot of organisations, class B is too big, class C is too small, (3) the router is slow because of the big forwarding table if there are lots of networks	(1) more IP addresses, (2) more balanced use of IP address ranges (3) **route aggregation** makes forwarding table smaller because each entry (192.4.16/20) links to multiple networks

- Datagram data delivery model: (2.1) every datagram packet has enough information to enable network forwarding without advance setup, (2.2) unreliable, no recovery, best-effort service (e.g., packet loss, out of order, repeated delivery):

  ▼ IP datagram packet format: header + data

```
Version: ipv4 or ipv6, so software can process the rest using the correct techniques
Header Length: counts words, usually 5 words (20 bytes)
TOS: allow packets treated differently based on application needs
Total Length: length of the datagram, counts bytes, maxSize = 65535 bytes
Identification & Flags & Fragment Offset: fragmentation, flag (e.g., more to follow)
TTL: max hop counts for routers, discard packets stuck in routing loops (e.g., 64)
Protocol: demultiplexing key that identifies the higher-level protocol (e.g., TCP, UDP)
Checksum: check for packet corruption, not as strong as CRC
Source Addr & Destination Addr: reply, forward
```



  - Fragmentation and reassembly is used because different network technology have different maximum frame size, and the IP datagram needs to fit in the payload of the link-layer frame, so a maximum transmission unit (MTU) is defined as the largest IP datagram for each network technology (e.g., Ethernet, 802.11)
    - Fragmentation occurs in a router when forwarding a datagram onto a network with a smaller MTU than the datagram

- Reassembly at the receiving end uses the `identification` field (unique among datagrams at destination from source)