# 2017 P5 Q4

1. In a hop-by-hop network, the functionality the functionality which ensures reliability and security of data is performed at every hop, whereas an end-to-end network requires all such functionality to be implemented in the two end nodes, with the intermediate network segments having to do as little as possible.

End-to-end networks provide better security. For example, when sending an encrypted webpage from a server to a client, the intermediate nodes in an end-to-end network, the packets would not be decrypted until they are received by the client node. However, this eliminates the potential for more efficient transmission which is present in hop-by-hop networks.

* 1. For networks in which the nodes are geographically far apart, propagation delay is the largest, typically on the order of milliseconds, as the latency of any given link is limited by the speed of light.

Queueing delay is the next largest if the network is under high load. This is because there is less “idle time” in which the queue can be emptied entirely.

Transmission delay is relatively small, on the order of microseconds, if the bandwidth of the network is large and the packet is small. The transmission delay is inversely proportional to the bandwidth because more bits can be transmitted at once.

The processing delay is defined to be negligible.

* 1. // TODO

# 2018 P5 Q4

1. Embedding can result in each layer adding its own header to the packet, which can result in a lot of additional data being transmitted. For example, a packet might be transmitted with TCP, IP, and Ethernet headers.

Having distinct layers might hurt overall performance, as internal state/data cannot be shared between layers.

* 1. When transmitting a byte, an additional (“parity”) bit is included. If the byte has an odd number of 1s, the parity bit is a 1. Otherwise, the parity bit is a zero. Therefore, the combined byte and parity bit are guaranteed to contain an even number of 1s. If the receiver detects an odd number of 1s, then at least one bit must have been corrupted in transmission. This can detect any odd number of errors.
  2. Assuming that the bits of the data are numbered from 0 to 31, 6 parity bits are required. For parity bits 0-4, bit *i* will ensure that the parity is even amongst the data bits for which the *i*th bit of their index number is set. Parity bit 5 will ensure the parity is even for the entire block. Therefore, the receiver can generate a 5-bit number whose *i*th bit is set if and only if parity bit *i* is incorrect. If this number is non-zero, it will be the index of the erroneous data bit. Otherwise, parity bit 5 will be correct if there was no error, and incorrect if the error is in data bit 0. This is a Hamming code.
  3. This transmission would be successful because there is no collision.
  4. This transmission will be unsuccessful. A and C both think that B is available when the transmission starts, and so a collision will occur.
  5. This transmission will be unsuccessful because A has no way of knowing that C is transmitting, and thus believes that B is free. This will cause a collision.
  6. B will see that A is transmitting and so will be aware that its own transmission to C will fail. There will be no collision but B will not send the packet to C.

# 2019 P5 Q5

* 1. 500TB = 500 TB \* 8000Gbit/TB = 4,000,000Gbit

4,000,000Gbit / 10Gbit/s = 400,000s = 111 hours = 4.63 days

This is much slower than overnight package delivery even with 0 propagation delay, and so the latter should be used.

* 1. 500TB / 400Gbit/s = 1000s = 2.78 hours

The propagation delay would likely be negligible when compared to the transmission delay (on the order of milliseconds, even over 2600km), and so this would likely be much faster than overnight delivery.

* 1. FlyByNight would be suitable for the 500TB/hour task, as this requires the transmission of only 1.11Tbit/s which is comfortably less than 16Tbit/s.

However, the maximum latency of 4 minutes makes FlyByNight unsuitable for the VR system, which requires latency of less than 5ms.

* 1. 4 pairs of twisted cabling (2 pairs for incoming data and 2 pairs for outgoing data) each with 250MHz bandwidth means that effectively 500,000 tokens can be sent/received per second, making the effective bandwidth 500MHz. Tokens with 4 possible levels can encode 2 bits each, and so 1,000,000 bits can be sent/received per second, which is 1GHz.
  2. Error-correction techniques such as Hamming codes can be used. Physical media errors can also be reduced by using more reliable materials for the cables.

1. // TODO