• The solutions provided hereon are for reference and study tools and are not to be considered final exam-assessment criteria

2. (Corrected solution) Calculate how much time it takes to transfer a file which size is 1473 Bytes from a host A to a host B assuming a direct connection between them. The **Rtt is 20 µs** and the transmission speed is 10 Mbps. Assume that the sending application at host A uses the following protocol stack: UDP/IP/Ethernet. The involved header sizes follow:

- Size of a UDP datagram header is ~~20~~ <u>8</u> Bytes
- Size of an IP packet header is 20 Bytes
- Ethernet Datalink header is comprised of:
  - Dest. Address (DST MAC): 6 Bytes
  - Src. Address (SRC MAC): 6 Bytes
  - Ethertype: 2 Bytes
- The Max payload size that can be encapsulated into an Ethernet frame is 1500 Bytes (Known as MTU, or Max. Transfer Unit)
- The Physical-layer Ethernet header has two sections:
  - Preamble: 64 bits
  - CRC: 32 bits

The problem statement requests that we calculate the time that it takes transferring the full 1473-Byte file from host A to host B assuming thaboth hosts are directly connected at an unspecified distance for which we are given the Rtt = $20\mu s$. Consequently, $T_p = T_{prop} = \frac{Rtt}{2} = \frac{20}{2}\,\mu s = 10\mu s$.

In general, UDP's socket interface will accept write sizes larger than 1473 bytes, but that entirely depends on the implementation and on certain technical details about the IP configuration such as whether PMTUD is in use or not. We'll assume the following context for responding to the question:

- The MTU is 1500 bytes according to the problem statement which leads us to be sure that the system will fragment any transmitted IP packet larger than that size.
- The system is not doing any kind of Path MTU discovery.

We can expect the datagram socket to accept the 1473-byte sized block of data in a single write operation. UDP will build a datagram containing the 1473 bytes, its total size will be 1473 bytes + 8 bytes (UDP header) = 1481 bytes. Can this size be sent in a single IP packet? An IP packet must fit the MTU, then we ask which is the maximum payload that can be encapsulated into one such IP packet.

Since a *normal (No options)* IP packet has a 20-byte header, the max payload it can carry is = 1500 bytes (MTU) – 20 bytes (IP Header) = 1480 bytes. The UDP datagram that is to be sent weighs 1481 bytes, therefore, IP will fragment it into two IP fragments:

Fragment 1: 20 bytes (Header) + 1480 bytes (Payload)
Fragment 2: 20 bytes (Header) + 1 bytes (Payload)

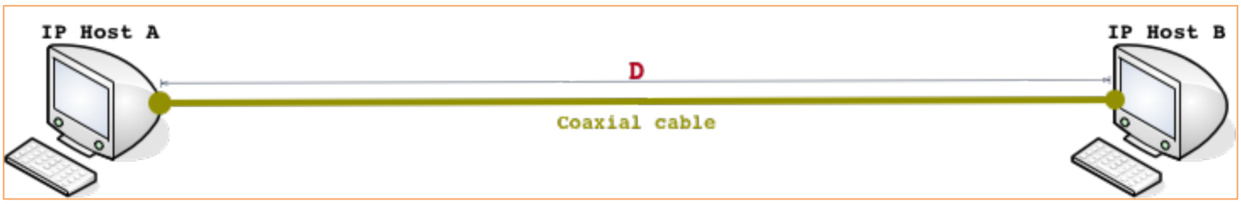

Figure 1. Hosts A and B directly connected at distance D

All in all, the file transfer entails sending two Ethernet frames --probably consecutively, but not necessarily so.

Furthermore, if these two frames were actually sent consecutively, the Ethernet protocol requires that in between the two the sender wait a period of time known as IFG (Inter-frame Gap) which standard time length is 12.5μs. We introduce this technical requirement of Ethernet LANs now, however, since we'll formally introduce it when we take up Ethernet, we will insert no IFG in the solution proposed below (Observe the light blue notice on Figure 2).

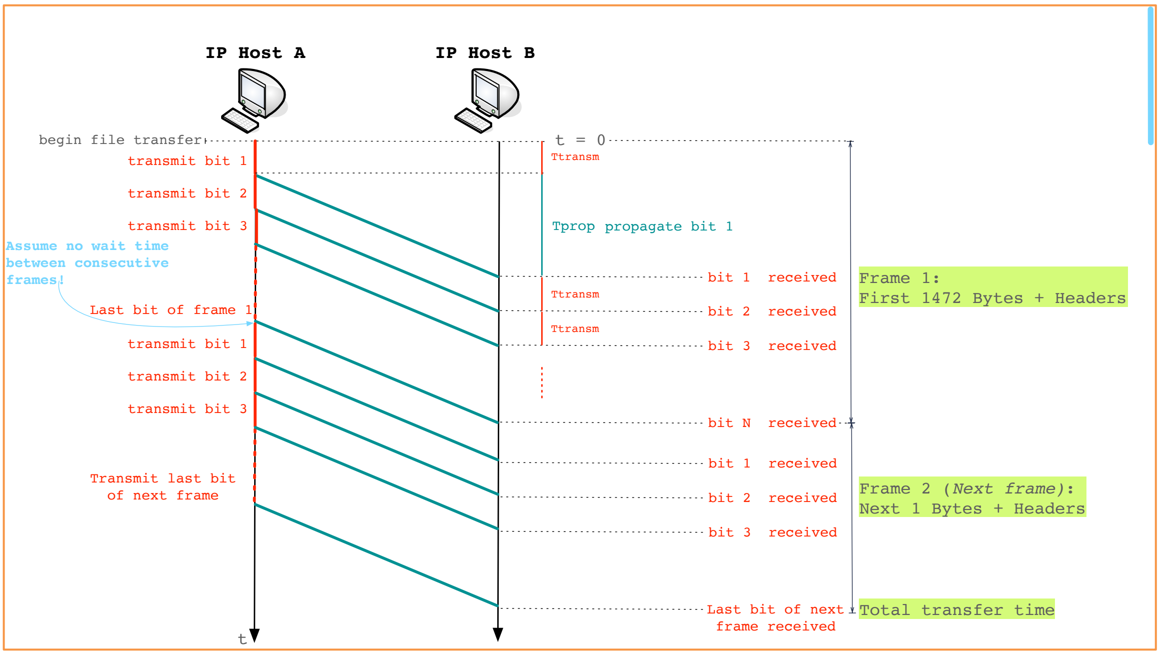

Figure 2. Total transfer time of two back-to-back frames. (In reality, these two frames won't necessarily be transmitted consecutively, but we assume so to enrich this exercise a little).

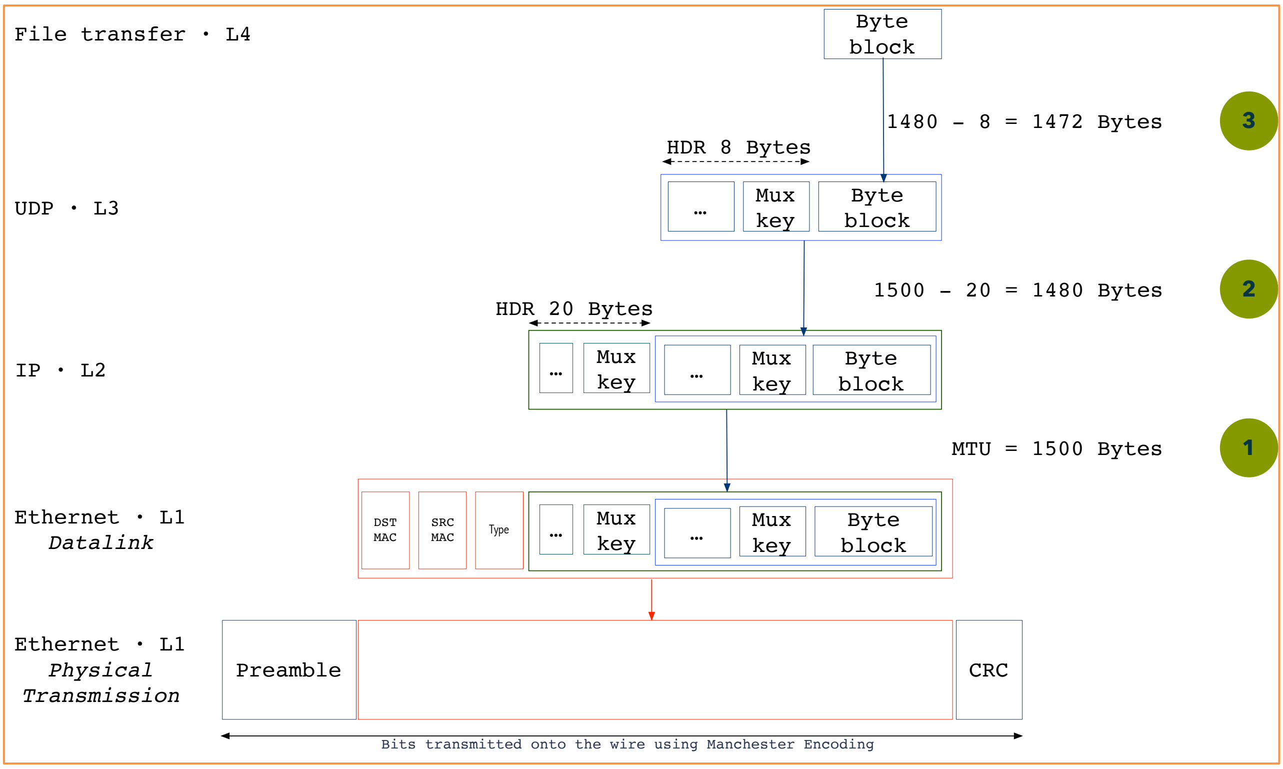

Figure 3. Protocol stack of file application over UDP/IP/Ethernet

By observing the conceptual chronogram in Figure 2, we calculate the total number of bits transmitted (On the wire), assuming the two frames were sent back-to-back and **with no IFG**. The multiplexing hierarchy is developed in Figure 3.

1. **Frame 1/2 (1472 B of payload)**

$$N_{bit-frame\ 1} = 1472\ Byte + UDP\ hdr + IP\ hdr + Ethernet\ Datalink\ hdr + Ethernet\ Physical\ hdr$$

$$N_{bit-frame\ 1} = 1472\ B \cdot 8\frac{b}{B} + 8B \cdot 8\frac{b}{B} + 20B \cdot 8\frac{b}{B} + 6B \cdot 8\frac{b}{B} + 6B \cdot 8\frac{b}{B} + 2B \cdot 8\frac{b}{B} + 64\ b + 32\ b$$

$$N_{bit-frame\ 1} = 11776\ b + 64\ b + 160\ b + 48\ b + 48\ b + 16\ b + 64\ b + 32\ b$$

$$N_{bit-frame\ 1} = 12208\ b$$

- The transmission time is the inverse of the transmission speed (100 Mbps): $T_t = \frac{1}{10Mbps} = \frac{1}{10 \cdot 10^6 \frac{1}{s}} = 0{,}1 \cdot 10^{-6} s = 100 \cdot 10^{-9} s = 100\ ns$
- The propagation time calculated above is: $T_p = 10\mu s$

2. **Frame 2/2 (1 B of payload)**

$$N_{bit-frame\ 2} = 1\ Byte + IP\ hdr + Ethernet\ Datalink\ hdr + Ethernet\ Physical\ hdr$$

$$N_{bit-frame\ 2} = 1\ B \cdot 8\frac{b}{B} + 20B \cdot 8\frac{b}{B} + 6B \cdot 8\frac{b}{B} + 6B \cdot 8\frac{b}{B} + 2B \cdot 8\frac{b}{B} + 64\ b + 32\ b$$

$$N_{bit-frame\ 2} = 376\ b$$

- Clarification about the propagation time: Since we are **assuming** the two frames are transmitted consecutively and **with IFG = 0 s**, we add no new T_p since it remains concurrent with the transmission time of the ensuing bits, just as happens in the transmission of frame 1.

- The transfer of the two back-to-back frames takes a final time:

$$Total\ transfer\ time = N_{bit-TOTAL} \cdot T_t + T_p = (N_{bit-frame\ 1} + N_{bit-frame\ 2}) \cdot T_t + T_p$$
$$Total\ transfer\ time = (12208 + 376)b \cdot 100\ \frac{ns}{b} + 10\mu s$$
$$\boldsymbol{Total\ transfer\ time = 1258.4\ ms + 10\ \mu s = 1258.410\ \mu s}$$