Answers

Q8-1. Serial

a. **Physical**: Simple wires; minimum requirement of ground, transmit, and receive.

b. **Data link**: serial communication that is determined by the host clock and framed by a start bit and a stop bit.

Q8-2. I^2C

a. **Physical**: 2 wires; serial data line (sdl) and serial clock line (scl)

b. **Data link**: multi-master networking protocol in which collisions are detected by having the sender listen to the network to make sure that what it “hears” is what it “said.”

c. **Network**: 1 byte transmissions framed by a start signal and a stop signal

d. **Transport**: Defined to be a series of 1 byte transmissions consisting of an address followed by one or more data bytes.

Q8-3. 10Base-T Ethernet

a. **Physical**: Twisted pair or coaxial cabling. Should provide a signal and ground

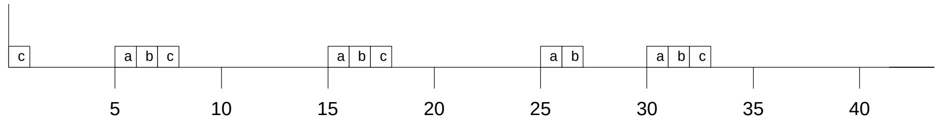
b. **Data link**: CSMA/CD combined with Ethernet packets

c. **Network**: Whatever is running on top of Ethernet; probably IP

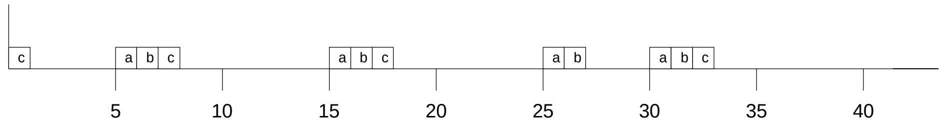
d. **Transport**: Whatever is running on top of the lower 3 level; should provide connection-oriented services. This will probably be something akin to TCP

Q8-4.

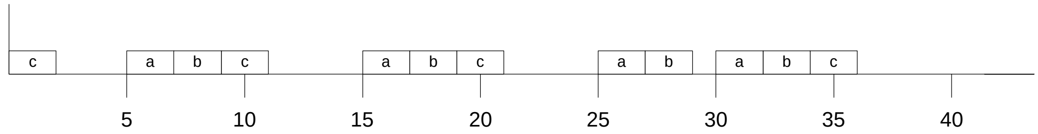
a.



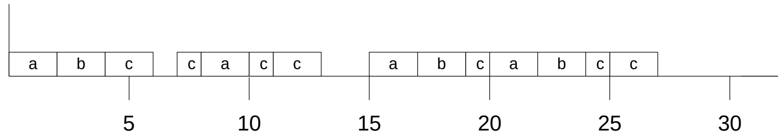
b.



Q8-5.



Q8-6. Assume these requests can be preempted.



Q8-7.

The key feature here is the router, which is a device that essentially "routes" traffic on one network to its proper destination. In this case, client 1 will attempt to send out an IP packet with client 2's address in the packet's destination field. The router, which is continuously monitoring the network, sees that client 2's address does not exist on this particular network. It then proceeds to look up client 2's address in its configuration tables. Once client 2's network is found, the packet is forwarded to that network. Client 2 will then see the packet on it's Ethernet and receive it.

Q8-8. Javacam

a.
Application: Java Applet

b.
Presentation: HTTP server that allows users to request pictures

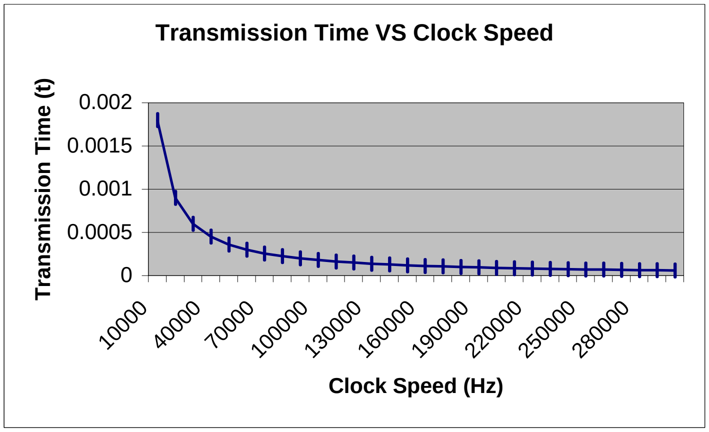
c.
Session: Quickcam server running on top of Java nanokernel

d.
Transport: TCP

Q8-9.

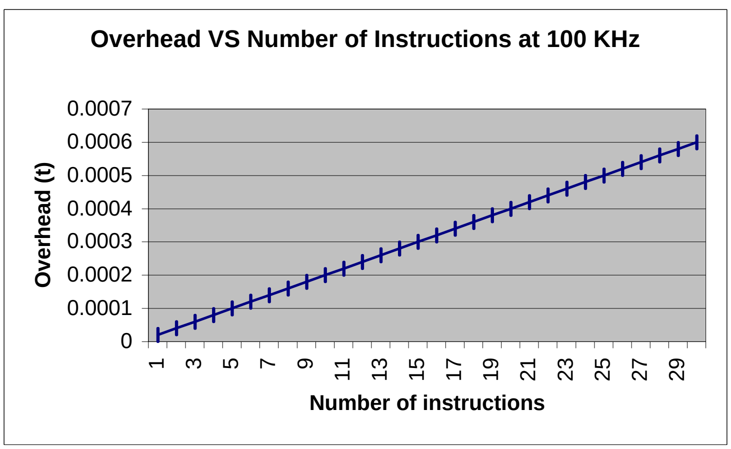
Number of bits per packet = n_{packet} = 18

t_n= n_{packet} * t_{bit} = 18 * 1/Clock Speed



Transmission overhead = t_x = t_r =number of instructions * Clock Speed
Assuming the average number of instructions is x, we find that the total overhead is:

t_m = 2 * x* 1/Clock Speed



Thus, at a given Clock Speed, the overhead will equal the transmission delay if there are 9 instructions that need to be executed.

Q8-10.

A situation in which this could occur goes as follows: Imagine that PE1 acquire a hold of the bus. However, due to a quirky transmission model, it only tries to acquire the data that it needs to send after it has acquired the bus. Unfortunately, in this case, the data that it needs to transfer is on a blocked and locked system resource. Since it cannot acquire a lock on the resource that it needs, it will block indefinitely while it has control of the bus. Hence, the other processing elements will be starved.

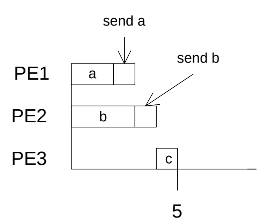
Q8-11.

They do not need to actually coordinate their transactions. If PE1 and PE2 have the highest priories in the system, and it happens that they are always makes request of the bus, this situation becomes possible. This is simply because the high priorities of the above name processing elements will always guarantee them use of the bus. Hence, in the case where they are continuously asking for the bus and their transactions each require the same amount of time, it will be possible for them to starve out the other processing elements.

Q8-12.

N time units, where N is the number of processing elements in the system.

Q8-13.



Q8-14.