(1) What is a protocol (define “network protocol” by providing the three keywords)? Answer this question by filling the three blanks below:

A protocol is (a) a set of rules (b) that govern communication between two hosts and (c) implemented as software.

Give two examples of the existing network protocols.

IP, TCP, UDP, FTP, HTTP, SMTP, IPsec, …

(2) Why we want “layered structure” for a protocol? Name the two benefits (please mention those we discussed in the class – you do NOT have to describe them).

- Any change in network service does not require re-compilation of the entire system (“isolation”)
- Combine different protocols (“flexibility”) without changing anything in any other protocol layers.

(3) What is the counter concept of “peer-to-peer”?

client-and-server

Note: end-to-end is not the counter concept of “peer-to-peer” (but it is the counter-concept of “point-to-point”).
(4) What are the major advantages in using “network protocols”? What are the major disadvantages in using “network protocols”?

**Advantages:**

- Using existing (network) protocols, your host computer can communicate with any other host computers or network applications (because of standard, universal protocols).
- Since existing (standard and universal) protocols are usually implemented as “libraries”, you (as an application programmer), do not have to implement them (you can just “call” them from within your network application programs, which lets you focus on developing your network applications).

**Disadvantages:**

- Making a protocol is a time-consuming process (before a protocol becomes available to public use, it usually takes years …
- If a protocol is developed by a private organization, it may cause “lock-in” (you will be under the strong control of the private organization).


- To prevent the receiving host computers from being overloaded.
- To prevent network congestions from happening.