Reading 3

姓名：程可

学号：518021910095

1. Use your own word to describe “end-to-end” argument

"End-to-end" argument states that "Occam's razor" should be applied when it comes to choosing the functions to be provided by the lower level of some system. The author argues that many lower-level functions of a system may be redundant or of little value when compared with the cost of providing them at that low level because first, the lower level subsystem is common to many applications, those applications that do not need the function will pay for it anyway; second, the low-level subsystem may not have as much information as the higher levels, so it cannot do the job as efficiently. So low-level funcions should pay more attention to performance instead of "perfect" reliability which will be done by high-level application.

2. Give at lease three cases that are suitable to use this principle

- Delivery guarantees: a data communication network is not able to know whether or not the application acted on the message delivered, so it should be the application instead of the data communication network that sends an acknowledgement to the sender application.
- Secure transmission of data: there is no need for the communication subsystem to provide for automatic encryption of all traffic because many applications have to obtain required authentication check and handle key management to its satisfaction.
- Duplicate message suppression: some duplicate requests may be originated by application because of its own failure/retry procedures, and low-level network cannot distinguish these duplicate requests, so it should be applications' job to detect their own duplicates.

3. Give at lease three cases that are NOT suitable to use this principle

- Also secure transmission of data: there may be some misbehaving user or application do not deliberately transmit information that should not be exposed. In this case, there should be the automatic encryption of all data put into the network to act as a one more firewall, although this encryption can be quite unsophisticated.
- Nowadays, the Internet has become an untrustworthy medium because there are many frauds and attacks. And also, nowadays most people using the Internet don't have enough professional knowledge about network, so many things such as authentication check and attack detection should be done by the middleware such as DNS servers instead of the "end".
- 《END-TO-END ARGUMENTS IN SYSTEM DESIGN》 was written in 40 years ago, at that time, every host can have a unique IPV4 address, and the end-to-end argument was based on that condition. But nowadays, because of the limited space of IPV4 address, many hosts have to

share the same IP address, and many middleware such as router, switch and gateway come out to handle the problem of limited address space. It is these middleware that handle and cover the problem and difficulty of the limited IPV4 address space for the end applications.

4. Consider the design of the file system based on inode (as we learn from class and the lab). Which part(s) of the design do you think can be removed? In another word, which part(s) of the design is(are) not flexible enough and should be implemented by the user?

I think file layer and above should be implemented by user. Because file system's design of inode constricts file's max size, the max number of files and which type of attributes of the file can be recorded, which is very inflexible and maybe go beyond the core function of file system. So the management of inode should be done by user.

5. The same question, for the OS.

I think most functions of the OS should be implemented by the user. If an application wants to use the functions provided by the OS, there must be a context switch to the kernel mode, which is very costly and inflexible. Now there is micro-kernel OS, whose memory only implements minimal abstraction, and most of its functions are implemented as different modules and run in user mode, which is very flexible, scalable and fault tolerant because if there are some mistakes in some module, the mistakes will only be restricted in that module and will not affect other modules.