A2 - End to End

Linzh (Zhongyu Zhao)

1. Two examples of "functions that make up a computer system" and... (1 point)

We use two layers in OSI model to explain this concept.

First, we will talk about the Physical Layer. In this layer, the information is electronic signals, e.g. high level and low level. All information transmission between computers must be interpreted into electronic signals and then emit via cable, optical fiber or other medium after modulation. However, this layer does not provide any check or inspect on the content that it carry. Thus, we could give a conclusion that Physical Layer provides the physical transmission methods and provides the accessibility for the content to the upper layer.

Moreover, we can discuss about the Data Link Layer. This layer have a strong connect with the Physical Layer, because devices in this layer, e.g. modem, will try to process the physical signal. For example, the modem will distinguish the terminate address and the repeaters will regenerate the signal to reduce noise. This layer will process the content of the signal, and regard them as frames, which means this layer could process some useless information to provides a continuous, error-corrected (try to do, but not guarantee) information.

2. Example of the interfaces between the communication system and the rest of a system... (1 point)

We will talk about the modulation and demodulation of wireless signals. If we have a tower which could emit the signals, the operator is able to broadcast any signal she/he want, e.g. FM, PM, AM. What she/he need to do is modulate the signal onto the carrier wave. Meanwhile, the receiver, e.g. the radio, will also need to demodulate the signal. However, different modulate method will need different circuit.

In this case, we could find that the interface is the signal tower (emitter) and radio (reviser), and the rest of the system are operator and people who want to listen to radio. Moreover, the communication subsystem is the transmission of wireless signal and its modulator and demodulator. The key users are operators and receiver. This interface provides a function between two people who want to talk (simplex, half duplex or duplex). Additionally, users can also use different 'functions' to transmit their information. FM and AM are the most commonly used modulation methods, but PM is also theoretically feasible. Besides, for digital signals, there are ASK, PSK, FSK, QAM, MSK and other methods in use.

3. Can you elaborate on what is the end-to-end argument?

3.1 what you understand by the term "low-level function implementation" (0.25 points)

The 'low-level' wants to indicate that the function is implementation at low-level. However, in the end-to-end arguments, the check is happened at the upper level, maybe the application layer, which make the check in low-level useless. Additionally, the process also leads to extra time or other resources.

3.2 can you detail your understanding on the authors arguments against low-level function implementation (0.25 point)

There are too many steps in a single task, and any step has a probability that go wrong. Therefore, if steps in the task only focus on their own job, the whole task cannot be done perfectly.

Meanwhile, if we introduce some check into this workflow, the accuracy rate will increase. However, it will also lead to a overhead if we check at every step. Therefore, the author proposed the end-to-end method, which set the check on the end of the workflow. In author's opinion, the less extra function lower layer provide, the better performance the application has.

3.3 what is the end-to-end argument in your opinion (0.5 points)

In my opinion, this concept is worth considering. The pros are introduced by the authors, e.g. more flex on development to implement some features to improve the user experience and let developer make trade-off instead the system does. However, I should also point out that if any lower layer does not provide some 'useless' function, the developer has to handle any potential problem, like the memory error and storage fail. Meanwhile, we know that the developers in a small companies and individual developers may not have enough skills to catch every potential error. Namely, the cost of development will increase, and it might be unacceptable for small companies.

As a result, I do not think the end-to-end should be extend to the whole computer system. Instead, the current solution, in my opinion, is a compromise solution. The lower layer provides limited correct for the data, and the application (not for user maybe) will ask for another copy if the data is corrupt.

4. In the conclusion, the authors state:...

4.1 can you specify if any devices you use are using any devices that use the RISC architecture, and which architecture is your phone using? (0.25 points)

I had a NVIDIA Jetson for machine vision. It is aarch_x64.

The macbook have a M1 processor, which also using RISC. Specifically, Arm64.

The architecture of my phone is arm64-v8a via `adb`.

4.2 can you give an example of "an esoteric feature" that RISC does not provide and perhaps the CPU architecture of your laptop/desktop provides? (0.25 points)

For example, the AVX-512 (or the previous version AVX-2) instruction set could accelerate the training and inferring process a lot by enhancing data rearrangement for more efficient data access and other features. However, this feature is only support on desktop. The desktop processors which designed after 2011 most support AVX. (from Wikipedia: Advanced Vector Extensions)

4.3 how is the RISC argument related to the end-to-end argument? (0.5 points)

CISC could add some common function into its instruction set. However, CISC also provides too many functions to the user, which the authors believe is wrong. Conversely, the RISC only support basic function, and any future function will need some combinations. In this way, the programmer could make use of the resources and the program will also be more flexible.

The similarity between them, in my perspective, is that both of them want simplify and design system efficient and highly customizable.

5. How is the end-to-end argument... APIs and services you provide... virtualization... (1 point)?

For me, maybe I will reduce the extra process on data behind the environments user use. Also, if I have full control to the physical devices and protocols, I would rewrite the driver codes to reduce the extra process. Moreover, I should not remove the check, because the check is necessary to ensure that the data we transfer is not broken. As a result, I will add the only check process at the last layer of this workflow, which means I check it just before using it by users.

Besides, I also should ensure that the visualization will isolate different users. In this way, I can also isolate some error or crash from other users. Additionally, I also will build a monitor system to identify if there is any unusual traffic or other weird stuff, like some websites are under attack. Therefore, I can prevent some potential problems and cluster crashes.