20170058 Keonwoo Kim · September 24, 2020
CS341 Introduction to Computer Networks

Reading Assignment #2

The *end-to-end argument* is the principle that any functions specific to some applications only should not belong in intermediate nodes of the communication, but must be implemented at the end points of the system. It is not a brand-new principle that the authors of the paper introduced for the first time, but rather it is accumulated over the years and used in many protocols from various authors in history.

The paper introduces some examples to spotlight the benefits of the end-to-end argument, starting with a case study of applications in the communication network: careful file transfer. According to the authors, there could be bunch of incorrect data transfer using the system, however making the communication system do lots of additional works, such as sending every data three times, in order to ensure the system to be completely correct is not beneficial at all compared to the payoffs we got from being correct. Instead of that, we check the end-to-end checksum to see the integrity of the transmitted data.

However, as discussed in the paper, some minimum level of reliability check could make the system better also in performance aspects. As mentioned above, too much reliability leads to a performance loss. However, too little reliability also ends up with a performance loss. Let p be the probaility of failure sending a packet and n be the number of packets needed to send the data. Then, the expected number of trials using 'end-to-end check and retry' will be

$$\mathbb{E}\left[\#\text{ of trials}\right] = (1-p)^n + (1-(1-p)^n)(\mathbb{E}\left[\#\text{ of trials}\right] + 1)$$
$$\implies \mathbb{E}\left[\#\text{ of trials}\right] = \left(\frac{1}{1-p}\right)^n \approx e^{pn}$$

which grows exponentially with n, the size of the data and approximately exponentially with p. When p is not ignorable or n is quite large, the number of trials to get the desired data using end-to-end check and retry method increases quickly. This example shows some sort of reliability check is needed, and we need to consider such tradeoffs. Note that this does not mean that the system should provide perfect reliability, but it is enough for the system to provide some minimal levels of reliability to not decrease the performance due to lacks of reliability.

The end-to-end argument can be applied for other concepts as well: guaranteeing the conveyance of messages, secure transfer of data, filtering duplicate messages, guaranteeing the order of the messages, and so on.

The criteria we use to identify the end nodes and what should the system do varies depending on the problem and the situation. For example, in real-time living conversation of two people, it is not that important to filter the missing or duplicated messages. However, in the case of transmitting recorded voice messages, packet ordering and duplicate message filtering is important. This implies that the end-to-end argument is not a fixed principle that every protocol should satisfy.