Summary:

The internet is formed because of two main goals and several minor goals. The first and most important goal of initializing an internet is multiplexing. That is, all kinds of networks can connect to the internet system. This goal appeared because the following reason: according to the economic reason, the ISPs always want to make more profits with low costs. That is, asking everyone to do some engineering for the internet is unachievable. The developer of the internet can only give them the lowest cost option to enable the existence of the internet.

In addition, how to apply the multiplexing is another problem. There are three aspects need to be considered: first, they use networks as the basic component and connect all of them to form the internet. The basic function or service they need is to enable data flowing. So, the developer chosen the packet switching as the basic method to implement, because the packet switching has been broadly used by different networks. Second, because there are different types of networks, they need to define a unified system to integrate transmission media, which will result in higher degree of integration and better performance. Third, they need a technique to achieve or control the data flow, so the gateway (a layer of Internet packet switches interconnected networks, and it is the interface that connect separate networks with different protocols) is defined.

Based on those assumptions above, the internet was created, and all the networks are connected to each other. Now, there is a second major issue residents within the internet system. It is effectiveness. How to make the data that the sender wants to send accurately and fast to arrive its destinations is the problem that this goal wants to solve. The developer listed 7 sub goals under the goal of effectiveness. They are: 1. Internet communication must continue despite loss of networks or gateways. 2. The Internet must support multiple types of communications service. 3. The Internet architecture must accommodate a variety of networks. 4. The Internet architecture must permit distributed management of its resources. 5. The Internet architecture must be cost effective 6. The Internet architecture must permit host attachment with a low level of effort. 7. The resources used in the internet architecture must be accountable. These goals are separated by the importance to achieve. But it is not the absolute importance level list, it might change according to the requirements.

Now it comes to the first sub-goal of the effectiveness: Internet communication must continue despite loss of networks or gateways. It is also called survivability. In other words, it should have some fault tolerant mechanism. For achieving this, the developer does the following: Instead of storing nodes of packet switching and replicate (high accuracy of replicate is expansive and hard to achieve), the fate-sharing method is developed (also accompany with the development of datagram. It just like the USPS delivery service. The sender includes the destination and source in the “header” and send it out.). The developer just took the state information and store it at the end point of the internet. The fate sharing mechanism accepts losing states if the data is lost at the same time. It brings two more benefits: 1: the packet switching nodes don’t need to include any state information (lower costs). 2: because the mechanism puts more trusts on hosts rather than a good internet, it lowers the difficulty to build up an internet and make it has more fault tolerance.

Second, the internet should support different types of communication services. Initially the developer tried to use TCP-like services to cover all circumstances. The TCP supported the reliable data transmission (there are several “beta” editions of TCP like the protocol enables both byte level and packet level data flow control, but the developer realized that the packet level is not useful because inserting control information into bytes is simpler, and the byte level flow control enables the multiplexing of the internet. That is, the byte level can dissemble large packets into smaller parts and be passed to the networks with worse bandwidth. Although the packet level flow control eliminated the situation that some only one-byte packages arrives earlier than the larger packages and once those one-byte-packets fail, the retransmission will occur, the packet level flow control will decrease the work efficiency. So, that’s the reason why the TCP looks like what it looks like nowadays). But, later, the developer noticed that the TCP cannot cover all of the situations (It cannot cover the data transmission situations that do not need reliability, or the reliability even put some negative efforts on it. For example, the debugging environment does not need reliability. It just needs the faults occur. Or like YouTube video, it does not need reliability. Lacking a few frames does not influence at all) Thus, the internet protocol (IP) was separated from the TCP. The IP only served for transmitting data without caring anything. It is the basic building block of the entire internet. Furthermore, the economic factors contributed to the invention of IP as well. The ISPs are always reluctant. that is, they always desired to consume as less money as possible to make largest profits. Thus, the internet developer needed to design a long-lasting architecture that people do not need to change often, but it does not satisfy the design of a system. The requirements will come into the system over time. So, designing a flexible enough system satisfied the requirement (people can build all kinds of protocols under IP).

Third, other goals either had low significance or has already been deployed into the internet. (like the #5 goal, it is unable to be achieved if the data is only 1byte but with 40-bytes header. Or like #4 goal, the internet must permit distributed management of its resources. It has been achieved because not all of gateways or networks are controlled by only one administer).

Now, most of major goals are settled, but how to implement them into the real world also have some difficulties (The developer called this process as realization). The developer finally decided to give the implementor a guidance to do the realization. Although there is another way which is called simulation (Simulating some realizations and put them into the different circumstances might be able to solve the problem above), lacking experience related to it make that approach hard.

Finally, the internet architecture was designed to be insecure. even though the developer know that problem exist in early stage of the internet. It is because the requirements of the security cannot be address. It is hard to know which part of security is most important. And it also because of the economic issue. The people who suffer from the security issue is not the people who pay for the security improvement. If the internet wants to increase the security, the entire architecture must be changed. Everyone uses the internet will pay for it, but not everyone wants to pay for it. In addition, the availability issue (Availability, use trustworthy source. But the user normally doesn’t care about whether the source is trustworthy. So, the internet architecture cannot solve this problem. It should be solved at higher layers.) can also make that happen.

Take out:

It should be shallow from deep. First, I learned the way that the developer built the internet architecture. The developer just addresses the main goal of the internet according to several dimensions of considerations. It is astonishing that although it is a computer science problem, the considerations of economic, politic (like what the developer said, sometimes they do not pay for it does not because it is an advanced technology, but just not that profitable) and so on can even influence the architecture of the “pure” computer world.

Second, the materials make me have better understanding of the four layers of the internet (especially link layer and network layer). I know the reason why every packet that we sent need to include a header (actually, it is just the datagram). It is because the internet should be fault tolerance. Then, I learned some knowledge with TCP. In the class, I cannot understand why the reliability is not important. In my imagination, the internet itself should be reliable enough. After reading the paper, I noticed that sometimes the reliability can put some negative effort to the sufficiency. For example, when we are using zoom meeting, the internet automatically checks whether the video streaming lack one frame. If it lacks one frame, the entire video streaming will be disrupted and retransmitted to the destination. That would be a disaster. It might cause the end-hosts who watch the streaming facing some sudden silent or cause the end-hosts watch the video that is performed several minutes before (The source or the streamer asked a question a few minutes before, and it just delivered to the watcher several minutes later).

Finally, in that part, I get a deeper understanding of the fate-sharing mechanism or principle. In the class, there are only several words to describe it, which confused me. Now I addressed that the fate sharing is the relationship between the packets and the states information (when does it arrive, what is the destination…). The state information will be lost with the packets. And now, not only the mechanism is clear to me, but also the reason why internet need the fate sharing. It increases the fault tolerance(don’t need to keep connection) and increase the efficiency(don’t need to add state information in end nodes).

Third, the biggest and deepest take way from the material is that I learned the way that they consider a system problem. That is, I learned the basic way of how to build a system. Next time if I need to design a new system for some purpose, I could follow the way that the developer of the internet did. I could first set up the most important goal of my system. I need to consider several points: first, what is the most important or most basic function that my system needs? Second, to achieve the most important or basic goal, what kind of technology/algorithm I need to implement? Third, is the cost of the system can be profitable? Is my system able to fit into the current situations without as less as effort in engineering? Then, after the most important goal is addressed, I might consider what should be the basic building block of my system, and what are the benefits and takeoffs of each type of building block. (The building block could be the minimum data size, like the bytes level data flow in internet, or an infrastructure or services that enables other services, like the IP on the internet.) After meeting all of those basic requirements of the system, the most important and necessary service of the system should be considered. In addition, whether the service can be divided into several sub problems should also be considered. If so, I could find out the most priorities sub-services and remove the sub-services with low priority. Then, the basic infrastructure and the basic service is built. After having the basic functions of the system, I need to consider the flexibility of the system. Can the system be flexible enough to enable other developers who add and accomplish requirements succeed? How could I build the interface to enable them to do so? After completing those three major things, a good infrastructure of a system could be built. Later, the only problem is just realization: how could I implement the system in the most economical way and how could I give a guideline for implementors to keep building the system? Finally, if all of the questions are solved, a sophisticated system is constructed.

Finally, I awarded that building a system is just the process of the bottom-up approach (basically, the development of the infrastructure of the system do not need to take care about coming requirements (except the requirement is extremely necessary like separating IP from TCP)).