Network Security

Professor Gaurav S. Kasbekar
Department of Electrical Engineering
Indian Institute of Technology, Bombay

Week - 01

Lecture - 08

Review of Basic Concepts and Terminology in Communication Networks: Part 6

Hello, in this lecture we will discuss layering, which is a concept used in the design, development, and maintenance of communication networks. The motivation for using layering is that, as we have seen by now, networks are complex. There are a lot of components, such as end systems, routers, and communication links, which are illustrated in this figure here. So, these are routers, and these are end systems, and connecting different routers and end systems. We have communication links. There are also a lot of networking functions, such as reliable data transfer, routing, congestion control, and medium access control, which we discussed briefly in the previous few lectures.

So, what approach do we typically use in engineering to simplify the design of a large and complex system, such as a network? We break down the system into smaller parts. This is the concept of modularity. Examples of modularity in some other contexts are as follows. In programming, a large program is divided into functions and procedures, and different functions and procedures can be developed by different programmers.

So, different programmers can develop different functions or procedures after they have agreed on the interfaces between different functions and procedures. In particular, the arguments to be passed to different functions and the return values of different functions and so on. So, once programmers have agreed upon these, different programmers can develop different functions and procedures. Another example is from the context of circuit design. Here, a large circuit is divided into blocks and different circuit designers may develop different blocks.

So to illustrate this, consider a circuit. It may have different blocks which perform different functions. And there are I/O ports between different blocks. So, a circuit has been broken into different parts as illustrated here. And these are the input-output ports of the different parts of the circuit. Once the circuit designers have agreed on what signals will be present

at different I/O ports, then different circuit designers can develop different blocks of the circuit.

So, this is another example of modularity. Now, different parts may be designed independently, provided that the interfaces have been standardized. Different parts may be functions or procedures in the programming context or they may be different blocks in a circuit context. And the interfaces, examples of interfaces are function return values and arguments and I/O ports in the case of circuit design. So, this shows examples of interfaces, function return values and arguments, and signals at the I/O ports of circuit blocks.

These are some examples of interfaces. Once interfaces are standardized, different parts may be designed independently. This is also the idea used in network design. The concept used is called layering, and these different parts or modules into which the networking functions are divided, these parts are known as layers. So, the concept of modularity in networks differs from that in programming or circuit design in the sense that each layer is present not only at one location but it is present across many locations.

So, each layer is distributed across many nodes which may be end systems or routers. There are five layers in the Internet. These layers are application, transport, network, link, and physical layer. The application and transport layers are only present in end systems, but the lower three layers are present in end systems as well as at routers. So, we can see from this picture that the physical layer is present at every end system as well as at every router.

And the link layer is also present at every end system as well as at every router. The application layer is present at every end system. And the transport layer is also present at every end system, but not at routers. There are also devices called switches, which are conceptually similar to routers, but there's a technical difference. They operate at only the link layer. They don't have the network layer.

So, from this picture, we can see that each layer is distributed across multiple entities. It is not restricted to only one entity. So, in a layered architecture, layer n provides a service to layer n+1 by itself performing some actions and using the services provided by the layer below it, that is, n-1. So, in this protocol stack, these layers are numbered as follows. The physical layer is layer 1, link layer is layer 2, network layer is layer 3, transport layer is layer 4, and application layer is layer 5.

And layer n provides a service to the layer above it by itself performing some actions and using the services provided by the layer below it. One example is as follows. So, consider

the example of n equals 4, that is, the transport layer. The transport layer uses the services of the network layer, which is the layer below it. In particular, the network layer provides unreliable packet transfer between end systems A and B.

That is, the network layer tries its best to transfer packets from A to B, but some packets may be dropped, some packets may be corrupted, packets may be reordered, and so on. So, this layer, layer number 3, network layer, provides unreliable packet transfer between end systems A and B to the transport layer, that is, layer 4. Now, the transport layer uses the unreliable packet transfer services of the network layer, that is, the layer below it, n equals 3. And transport layer itself performs some actions, such as error detection, retransmissions, and so on. The result of the use of these services and the performance of these actions: the result is that the transport layer provides reliable packet transfer services to the application layer, which is layer 5.

So, this is one example with n equals 4, of the transport layer using some services of the network layer and itself performing some actions to provide reliable packet transfer services to layer number 5, that is the application layer. In a layered architecture, the concept of information hiding is crucial. Layer n can access the services provided by the layer below it, that is, layer n-1, using only the service interface between layer n-1 and layer n. It does not need to know how layer n-1 implements the service. That in particular, layer n views layer n-1 as a black box. So, the service interface between two layers is typically an application programming interface, or API, or it may be some system calls in an operating system, and so on.

More generally, a layer does not need to know the implementation details of any other layer to perform its own function. So, because of this property of information hiding, different layers can be independently designed and maintained. While developing a layer n, a developer does not need to know about the details of the other layers. So, that's the property of information hiding that simplifies network design and maintenance. The advantages of layering are as follows.

A complex task, namely performing the set of all the networking functions, is divided into smaller and manageable tasks at different layers. Another advantage is that the implementation of a layer can be changed or updated without affecting the other layers, provided that the interfaces are unchanged. An example which occurred after the advent of fiber optic technology was the following. Earlier, long-distance links in the telephone

network were based on copper cable. They were upgraded to fiber optic cables after fiber optic technology was invented.

And this transition happened without having to change the higher layers. Only the physical layer and link layer were changed when cables were upgraded from copper cable to fiber optic. The higher layers did not have to be changed. Another common example is, we know that there are often updates in software programs at a layer. When these updates are installed, then the other layers are not affected.

So, this is another advantage of layering. Once interfaces are standardized, different teams or companies can develop hardware or software for different layers. This is another advantage of layering. Now, let's discuss the layers in the internet individually in some more detail. So, recall that the layers in the internet are application, transport, network, link, and physical.

This slide summarizes the functions of different layers in the internet. The application layer is where network applications and associated protocols reside. Examples are web browsers. Web browsers, in the context of the web application, the applications which reside in the application layer are web browsers, such as Firefox, Chrome, Microsoft Edge, and so on. And web servers, such as Apache and Microsoft server.

These are applications which reside in the application layer for the web application. Then, for the email application, mail readers, such as Gmail and Webmail and mail servers, where user mailboxes are stored. These are the applications which reside in the application layer. And also the application layer protocols, which support various applications, such as web, file transfer, and email, they also reside in the application layer. So, examples are HTTP, FTP, and SMTP.

HTTP is Hyper Text Transfer Protocol and it supports the web application. FTP is File Transfer Protocol, which supports the file transfer application. And SMTP is Simple Mail Transfer Protocol, which is for email. So, this is the application layer. It is where network applications and the corresponding protocols for supporting the applications reside.

Now the transport layer; the main function of the transport layer is multiplexing and demultiplexing. Specifically, there may be multiple application processes in an end system. The transport layer takes data from different application processes and creates a single data flow, and sends it over the network. So, multiplexing is the function wherein the transport layer takes data from different applications and passes it to the network layer for being

transferred over the network. So, this is the multiplexing function, and the demultiplexing function is when the transport layer takes data received from the network layer and routes it to the correct application process.

So, this multiplexing and demultiplexing is done using different identifiers corresponding to different application processes, which are called port numbers. So, this is the basic function of the transport layer, that is multiplexing-demultiplexing. Popular transport layer protocols in the internet are TCP and UDP. TCP is Transmission Control Protocol and UDP is User Datagram Protocol. A transport layer protocol may perform functions other than multiplexing and demultiplexing.

For example, TCP performs reliable data transfer as well as congestion control. UDP performs only multiplexing and demultiplexing. Now, the network layer has the main function of routing packets from the source to the destination end system. So, computation of routes is done by the network layer. And also, it assigns IP addresses hierarchically to different nodes in the network.

So, routing and addressing are functions of the network layer. The link layer is responsible for transferring packets between neighboring end systems and/or routers. So, the transfer over an individual link in the network is accomplished by the link layer. And the physical layer is responsible for bit transfer on the physical link. The physical layer takes care of issues, such as modulation, demodulation, bit synchronization, timing, and so on.

Now, let's discuss the functions of the lower four layers in some more detail. So, the transport layer: we discussed that transport layer performs multiplexing and demultiplexing. Specifically, multiple applications in an end system may be communicating with applications in other end systems. For example, multiple browser windows may be open, an email application may also be open, and so on. The transport layer performs the basic function of multiplexing and demultiplexing using unique identifiers, namely port numbers assigned to each application process.

So, using these port numbers, the transport layer passes the received packets from the network layer to the correct application process. And the transport layer also adds port numbers to packets sent by the application layer so that these port numbers can be used by the transport layer on the other side to pass the packets to the correct application process. The transport layer also performs some additional services. So, these additional services are optional. Transport layer protocols in the internet are the following.

Transmission control protocol, TCP and user datacom protocol, that is, UDP. TCP performs reliable data transfer, that is, it implements recovery from bit errors and packet losses and provides in-order delivery of packets. TCP also performs congestion control, that is, ensuring that the rate of data transmission is regulated so as not to cause too much congestion in the network. TCP also performs flow control. Flow control is ensuring that the sending rate is regulated so the receiver's buffer should not overflow.

So that is ensured by flow control. So, the difference between congestion control and flow control is that in the case of congestion control, the sender ensures that the routers on the path from source to destination do not overflow. And flow control ensures that the buffer at the receiving end system, that buffer does not overflow. So, TCP performs reliable data transfer, congestion control, and flow control in addition to multiplexing and demultiplexing. User datagram protocol is a minimal protocol.

It doesn't perform reliable data transfer, congestion control, or flow control. So, it is useful for real-time services, such as internet telephony and video streaming. So, UDP is useful for such applications because it does not have the overhead resulting from these functions, such as congestion control, flow control, and so on. So, because of this lack of overhead, UDP can be effectively used by real-time services. Now, let's discuss the network layer.

The transport layer at the source end system passes the message along with the destination IP address to the network layer. The network layer routes the packet over the network from the source end system to the destination end system, typically over a large number of intermediate routers. There might be, for example, 10, 15 routers on the path from source to destination. The network layer routes the packet from the source end system to the destination end system. So, the network layer, to be able to perform this routing, the network layer periodically computes paths from different pairs of source and destination end systems.

That is, the network layer periodically computes paths from different end systems A to different end systems B. So, the network layer does this by using routing algorithms, such as OSPF and RIP. OSPF is Open Shortest Path First and it is an implementation of Dijkstra algorithm, which we discussed earlier. And RIP is Routing Information Protocol. It's an implementation of Bellman-Ford's algorithm.

So, the network layer uses routing algorithms like these to periodically compute paths And after these paths are computed, the network layer populates the routing tables at routers, and the routers use these routing tables to decide on which link each incoming packet

should be sent out. Now, the network layer routes packets over multiple hops from the source end system to the destination end system. For transferring a packet over an individual link on the path from the source to the destination, the network layer takes the help of the link layer. Specifically, to move a packet from a node A to the next node B in a path, the network layer at A passes the packet to the link layer at A, the link layer at A transfers the packet over the communication link to the link layer at B, and finally the link layer at B passes the packet to the network layer at B. So, the link layer accomplishes transfer over a single link in a communication network.

Examples of link layer protocol are PPP, which is Point-to-Point Protocol. It was used in the context of dial-up internet. Ethernet and Wi-Fi; these are other link layer protocols. So, there are some other link layer functions, which may optionally be performed. So, if the physical link is a shared medium, then media access control is another function that the link layer must perform.

This is done by the MAC sublayer, which is a part of the link layer. So, for example, the old version of Ethernet and Wi-Fi; these are shared media, so, they require a media access control sublayer for performing media access control. The link layer may also perform reliable data transfer, for example, recovery from bit errors. Reliable data transfer is typically performed by communication links, which are wireless links, because the rate of bit errors is high on wireless links. So, it is better to do reliable data transfer locally on the link itself rather than rely on the end-to-end reliable data transfer protocol to recover from bit errors on a particular wireless link.

Another function performed by the link layer is encryption. So, encryption is often performed on communication links, which are wireless links and/or which are shared links. Encryption is also performed on point-to-point links, which may be tapped by intruders. So, this is for confidentiality, so that even if a receiver taps the transmission from source to destination, it is not able to decipher that information because of the encryption added to the message. Then, the lowest layer is the physical layer.

Recall that the link layer moves packets between neighboring nodes. The physical layer moves the individual bits in a packet across the physical link. In particular, the physical layer is responsible for functions, such as modulation and demodulation and bit synchronization and timing. So, these functions are performed by the physical layer. The physical layer provides a bit pipe.

As bits are sent by the link layer to the physical layer, it keeps on sending the bits to the receiver. The physical layer is unaware of where packets start and end. Now in a layered architecture, the concept of encapsulation is used. Specifically, when a packet is passed from layer n to the layer below it, that is n-1, then layer n-1 adds some additional information. This additional information is called a header.

And this header includes information, such as source and destination IP addresses, port numbers, checksum bits, sequence numbers, and so on. So these headers are illustrated in this figure. The application layer passes a message M to the layer below it, that is the transport layer. This message M is to be sent to the application process at this destination site. So, the transport layer adds a header to the message M. The header is H_t.

This header includes information such as port numbers, source port number, and destination port number. Then the network layer adds another header, that is, H_n. So, this H_n includes the source IP address and destination IP address among other fields. Then the link layer adds another header H_l, which is the link layer header and this header is removed when the packet is passed from layer n-1 to layer n at the peer node. So, for example, when the packet reaches its destination and the physical layer passes the bits to the link layer, then the link layer removes the link layer header and then passes the rest of the packet to the network layer. So, H_l is removed by the link layer, then H_n is removed by the network layer, and so on and so forth. So, this header is removed after its function has been already performed.

So, for example, the network layer consults this H_n to find out what is the source IP address. So, to find out from which source the packet arrived. So, once the source IP address has been used, after that the header can be removed. So, this header is added to perform some functions by the source and it is removed by the destination after the functions have been performed. And this header removal also happens at intermediate nodes.

For example, when the packet is transferred from this switch to this router, the router removes this link layer header H_l and then the packet is passed to the network layer and then the network layer adds another new link layer header, that is, H_l; this H_l and then sends the packet to the destination. So, these headers are also removed and added at the routers during transit from the source to the destination. So, layering has several advantages as we discussed earlier, but there are some disadvantages as well. One disadvantage is that there is duplication of functionality. For example, error-checking is often done at both link layer as well as transport layers.

In particular, if the transport layer protocol uses TCP, then we know that it performs error-checking, but some link layers protocols, such as the link layer protocols in wireless links, they also perform error-checking. The reason is the following. Consider a packet being sent by a source S over several communication links to a destination D. Now, one of the links may be a wireless link. So, this is a wireless link. But the other links may be wired links.

Now, in a wireless link, error-checking is done by the link layer protocol. But the other links are wired links, and error-checking is not performed on these wired links because these wired links are quite reliable, and the overhead of error-checking need not be incurred because these links are quite reliable, so it is not necessary to add error-checking on a link basis. So because of this, it is possible that, with a high probability, a bit error might occur on one of the wired links between S and D. So, for this reason, error-checking still needs to be performed by the transport layer. By the transport layer protocol that runs between S and D. Even though error-checking is performed by some of the link layer protocols between S and D. So, there is this duplication of functionality.

Error-checking is performed both at some link layer protocols and at the transport layer protocol. So, this is one disadvantage of layering. Another disadvantage of layering is that functions at different layers cannot be jointly optimized. There's an example in the next slide in which there is such loss of performance because of layering. Cross layer design is an alternative paradigm to a layered architecture.

Here, two or more layers are combined and their functions are jointly performed. This results in better performance than layered architecture. But the advantages of layering are lost in a cross-layer design. So, the advantages like simplicity and ease of development, they are not there in a cross-layer design. A cross-layer design is complicated to design and maintain.

But it can provide a higher performance than a layered architecture. So, in the internet, the architecture is a layered architecture, but there are occasional cross-layer functionalities, where violating the layered architecture results in much better performance. We violate the strict layered architecture and use a partial cross-layer design. So, here's an example where there is some loss in performance because of layering. Consider file transfer from an end system A to an end system B via several intermediate routers.

A is connected to the next hop router by a wireless link. So, this is A and it is connected to the next hop router by a wireless link, and then there are other intermediate routers, and then there is the destination B. So, the first link is a wireless link on the path from A to B.

So, this wireless link is prone to errors. The transport layer at A adapts its transmission rate to the packet loss rate. So, whenever a packet loss happens, the transport layer at A reduces the transmission rate.

And when an acknowledgement is received, it increases the transmission rate. But the transport layer at A cannot distinguish between packet losses because of congestion in the network, and those because of errors on the wireless link. Some packet losses happen because of errors on this wireless link, so the packets are lost on this link itself. Some packets are lost because of congestion in the network at one of these other routers. So, the transmission rate adaptation algorithm performance could have been improved if this information were available.

For example, a packet loss on the wireless link may be because of a wireless channel fade, which may be very temporary. So, it is not necessary actually to reduce the transmission rate much. Whereas the congestion in the network may be more long-lived. So, in that case, it is better to significantly reduce the transmission rate. But the transport layer at A cannot distinguish between these two kinds of packet losses.

Those packet losses because of errors in the wireless link, and those packet losses because of congestion in the network. The transport layer at A cannot distinguish between these kinds of packet losses, so it is not able to adapt its transmission rate optimally so as to get the best performance. In this example, notice that, information about packet losses because of congestion is present at the network layers of these routers where the congestion occurs. Whereas information about packet losses because of errors on the wireless link, this information is there at the MAC layer of this wireless link. And the transmission rate adaptation is done by the transport layer of A.

These different functions are performed by different layers, and hence, we know that the information from one layer is not available to the other layer. Because of this, there is some loss in performance in this example. This is one example where there is some loss in performance because of a layered architecture. So to summarize, layering has a lot of advantages, but in some cases, there can be some loss in performance because of layering. This concludes our review of basic communication networks.

In the next lecture, we will discuss different types of attacks on networks. Thank you.