Network Security

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

Week - 01

Lecture - 07

Review of Basic Concepts and Terminology in Communication Networks: Part 5

Hello, in this lecture we review different metrics that are used to measure the performance of a communication network. So, there are four commonly used network performance metrics. These are delay, packet loss, throughput, and jitter. Let's discuss each of these in turn. We start with delay and packet loss.

Suppose A and B are end systems in a packet-switched network. So, A and B are illustrated in this figure. A is a desktop computer downloading a file from a server, B, and the communication is bidirectional, say. So, there is also traffic, there is a flow from A to B and a flow from B to A. Delay is the time taken for a packet to travel from A to B. So, notice that delay is measured on a per packet basis.

The delay is, in general, different for different packets, and for a packet, the delay is the time required for the packet to travel from the source A to the destination B. Packet loss is the phenomenon where packets are sent by the source A, but they are dropped by intermediate routers. Typically this happens because there is congestion in the network, because of which routers run out of buffer space and they have to drop packets. So, let's discuss queuing and packet loss. Packets arrive at a router on incoming links. In this figure, consider this router.

Packets come in at this router from source A and from source C. So, packets coming from A are shown in red, and packets coming from C are shown in blue. Now a packet that comes in at a router that is transmitted immediately on the output link if the output link is free. So, for example, if a packet arrives from C, and at that point there are no packets waiting to be transmitted on this link. In that case the packet from C will be transmitted immediately on this output link. But if there are other packets at the router, which are either being transmitted on this link or are waiting to be transmitted on this link.

In that case, the incoming packet waits in a queue for its turn to be transmitted on this output link. So, there is some time spent in waiting in the queue. This delay is known as queuing delay. And the queuing delay in a network varies with time. It depends on the current congestion level.

If there is high congestion, then there will be high queuing delays and vice versa. The queue grows whenever the total rate of incoming packets exceeds the output link capacity. For example, suppose the rate of this output link is 10 Mbps. Whenever the total rate at which packets sent by A and C exceeds 10 Mbps, the queue will start growing. As the queue grows, the buffer space at this router starts getting consumed.

Eventually, there will be no free space in this buffer for storing new packets that are arriving. In that case, the newly arrived packets are dropped since there's no space to store them. Now, if packets are dropped at the router, then they can be retransmitted by the source. We discussed earlier that the source starts a timer when it transmits a packet, and if it doesn't receive an acknowledgement until the timer expires, then it retransmits the packet. But the downside of retransmissions is that it results in additional delays.

So, the packet loss has to be detected and then another packet has to be sent and that new transmission has to travel all the way to the destination again. So, this results in delay, and it also wastes bandwidth in the network. Recall that congestion control seeks to limit the queue lengths and packet losses. Now there are four sources of packet delay at each router on the path from a source to a destination. Let's discuss these four sources of packet delay.

These four sources are transmission delay, propagation delay, nodal processing delay and queuing delay. So, let's discuss these and again I point out that these four delays are incurred at every router on the path from a source to a destination. So, the processing delay is the following. Once a packet arrives at an incoming link, for example a packet arrives from A to this router on this incoming link, the packet is checked for bit errors. We discussed earlier that checksum is added by the source of a packet.

Checksum might be, for example, Hamming code or parity checksum, and so on. So, using this checksum, the router checks for bit errors. If any bit errors are detected, then the packet is dropped; otherwise, the packet is further processed. Now, if the checksum verification passes, then the output link to send the packet on is determined using the destination IP address. How is the output link determined?

So, each router has a routing table. It is a table of the output link for every destination IP address or group of destination IP addresses. So, each router has a routing table like this one, which provides output link on which the packet should be forwarded for every destination IP address or group of destination IP addresses. And this routing table is populated by routing algorithms, which run in the network periodically, such as OSPF and RIP. So, these routing protocols, they populate the routing tables at routers.

Fine, in this step, the output link to send the packet on is determined using the destination IP address. All these steps, namely checksum verification and determination of the output link to send the packet on, these steps take time, and that time is known as processing delay. So, a processing delay is incurred at every router. Then another delay that is incurred at a router is the queuing delay. So, as we discussed on the previous slide, packets to be sent out on a given output link are stored in a buffer for their turn to be transmitted.

A packet that is stored in a buffer must wait in a queue for its turn to be transmitted. So, the time spent waiting in a queue is known as queuing delay. The queuing delay varies with time significantly. It depends on the current congestion level. If there is no congestion, then the buffer will be empty or nearly empty.

In that case, the queuing delay will be low and vice versa. If there's high congestion, then the buffers will be full or a significant part of the buffers will be occupied. In that case, the queuing delay will be high. Then, the third source of delay is the transmission delay. The transmission delay is the time required to send out the bits of a packet onto the output link.

In this figure, this packet is being currently transmitted on the output link. This transmission takes some time, and that is the transmission delay. So, how much time is taken for transmission? Let R be the output link bandwidth in bits per second. For example, in the case of DSL, the downstream bandwidth is 8 Mbps, and upstream bandwidth is 1 Mbps in a typical scenario.

So, this R is that bandwidth of 8 Mbps or 1 Mbps. And let L be the packet length in bits. For example, if the packet is 1,000 bytes in length, then L will be 8,000 bits. The transmission delay is the time to send the bits into the output link. So, each bit takes time, $1/R$, to be sent on the output link.

So, the time to send all the L bits is L/R. Thus, the transmission delay is L/R. Then, the fourth source of delay at a router is propagation delay. Let d be the length of the physical link. So, this is the output link and d is its length. So, this length can vary significantly.

For example, if it's a link in a local area network, then it might be a few hundreds of meters or so. But if it is a link in an intercontinental context, for example, a link between two routers on different continents, in that case, the length of the link might be thousands of kilometers. So, this d is the length of that link, and s is the propagation speed of the signal in the medium. The signal is an electromagnetic wave, and it travels at the speed of light in the medium. For example, the speed of light in copper is around 2 into 10 to the power 8 meters per second.

The propagation delay is the time required for the signal to travel from one end of the link to the other end of the link. The propagation delay is d/s, since d is the length of the link and s is the speed in the medium. So, the propagation delay is d/s. So, it's important not to confuse these two sources of delay, transmission delay and propagation delay. Transmission delay is the time required to send out a packet on the output link. Transmission delay is independent of the length of the link, but it depends on the length of the packet and the bandwidth of the link.

Propagation delay is the time required for the signal to travel from one end of the link to the other end of the link. It depends on the length of the link. It is independent of the length of the packet or the bandwidth of the link. Now, consider packets that are sent from a source S to a destination D via N intermediate routers, P_1, P_2, up to P_N. The end-to-end delay from S to D is obtained by adding all the delays on each hop from S to D. The end-to-end delay includes the four basic delays which we discussed above, namely nodal processing delay, queuing delay, transmission delay, and propagation delay.

There may be additional delays on some hops from the source to the destination. For example, if one of the links is a wireless link, in that case, there will be a medium access control delay for acquiring the medium, or one of the links may be a shared link to which many nodes are connected. So again, the source node has to first acquire the medium for transmission by contending with the other nodes attached to the medium. So, that will result in additional delays. In summary, at each hop from source to destination, these four delays are incurred, and in addition, some additional delays may also be incurred.

So, this concludes our discussion of delay. So next, we discuss average throughput. Suppose A and B are end systems in a packet switch network. This is A, which is a server, and B is an end system. Suppose a file of size F bits is transferred from A to B in T seconds.

Average throughput is the rate of data transfer. It is F by T, the number of bits transferred divided by the time taken for the transfer. So, in this definition, only the useful bits are

counted. Recall that the file is broken into several packets and a header is added to each packet. But in this calculation of average throughput, we don't count the header bits.

We only count the application data, which is useful to the application. So, the file itself, the bits of the file itself, which is F bits, only those are taken into account in this definition. So, the average throughput is F by T. For example, the average throughput if an attachment of size 10 to the power 6 bits is downloaded from Gmail in 4 seconds is, by definition, it's 10 to the power 6 bits divided by 4 seconds. That's 0.25 Mbps.

So, this is the average throughput. It's the rate of transfer. Now, we have another definition of throughput, which is instantaneous throughput. It is intuitively the rate at which the destination is receiving bits at the current instant. For example, when we download a file through a file transfer application or through a browser, then the browser displays either the amount of time left or the number of bits or how much data is still to be downloaded.

So, this number displayed by browsers or file transfer applications is the instantaneous throughput. That is, if it displays the bit rate at which the file is being downloaded, so then that is the instantaneous throughput and the remaining time and the number of bytes yet to be transferred. These depend on the instantaneous throughput. So, now, what do we mean by rate at which the destination is receiving bits at the current instant? Bits are discrete, so in practice, to measure the instantaneous throughput, we measure the average throughput in a small time window of maybe 100 milliseconds or so.

So, the average throughput in a small time window is used as the instantaneous throughput. So, this instantaneous throughput keeps on varying with time, depending on the congestion level in the network and other factors. Another performance metric is jitter. Again, suppose A and B are end systems in a network and there is a flow from A to B. This is the horizontal axis is the time axis and the upper set of packets are the stream of packets at the source, that is, A in this case. Suppose A sends packets, which are uniformly spaced in time as shown by the upper set of packets.

But at the destination, the same packet stream looks like the set of packets at the bottom of the figure. So, the spacing between adjacent packets is different for different packets. So, packets are evenly spaced at the source but they are unevenly spaced at the destination. That is because different packets get delayed by different amounts from source to destination. So, recall that the end-to-end delay varies from packet to packet mainly because of varying congestion levels in the network.

This variation in the end-to-end delay is called jitter. Jitter can be quantified in different ways. For example, we can talk about the standard deviation of the end to end delay. So, that's a measure of jitter, or we can talk about the maximum end-to-end delay minus the minimum end to end delay. So, that's another measure of the jitter.

So, at first sight, it might seem that only the end-to-end delay is important. Why is the jitter important? So, here is one example, where the jitter is also important from the point of view of a user. Consider video streaming of a live event over a network, such as a sporting event or a musical concert, and so on. 30 video frames are transmitted per second by the source, each in a separate packet.

Hence, one packet is transmitted every 33 milliseconds. Assume that the receiver starts playing the video as soon as the first frame is received. So, the receiver needs to receive a new frame every 33 milliseconds to display it. Now, suppose the receiver has just displayed a frame, and the next frame is received before 33 milliseconds from the play out of the first frame. In that case, the frame can be saved and displayed later, but if the next frame is received after 33 milliseconds from the play out of a frame, then the video is stuck.

So, the spacing between adjacent packets is not necessarily 33 milliseconds, even though the spacing is 33 milliseconds at the source, and that's because of jitter in the network. Different packets are delayed by different amounts from source to destination. So, the first packet may reach the destination very fast and it is played out immediately. But the second packet may be delayed. In that case, the spacing between reception of the first packet and the second packet is more than 33 milliseconds, and the video is stuck.

So, hence, the quality of the video perceived by the user that suffers adversely because of jitter in this example. In this example, jitter impacts the perceived quality more than the end-to-end delay. In a live stream, it doesn't matter much if the whole stream is delayed by 2 seconds instead of 1 second. That delay is negligible. But the video quality is not smooth because of jitter.

It starts and stops because of jitter. In this example, it is straightforward to hide jitter. What the receiver should do is that once it receives the very first frame, it should not play it out immediately. Instead, it should wait for some time and keep on buffering some frames so that it builds a stock of sufficient number of frames. And after that, it should play out the first frame.

So, hence, a new frame will be always available to it for playing out, even though the frames that are coming from the source, they are getting delayed. So, in this example, it is easy to hide jitter, but in some cases, it may not be so easy to counter jitter. For example, consider a live telephone call over the network between A and B. In that case, end-to-end delay should be small. So, typically end-to-end delay should be less than 200 milliseconds or so. So, in that case, the receiver cannot buffer packets by a large amount before playing out the packet.

So, it becomes difficult to counter jitter in such cases. Now, a network can be viewed as a communication infrastructure that supports distributed applications, such as web, email, games, e-commerce, file sharing, internet telephony, video streaming, and so on. The quality of service experienced by applications depends on the metrics that we discussed, namely throughput, packet loss, delay, and jitter. The higher the throughput, the better it is. The lower the packet loss, the better it is, and lower the delay and lower the jitter, the better it is for the application.

The QoS requirements vary across applications. Which ones of these metrics are important and which are less important, that depends on what kind of application it is. Broadly, applications can be classified into the following categories, but there are also hybrids. So, these categories into which applications can be classified are elastic and real-time applications. Examples of elastic applications are file transfer and web downloads.

Here, the delay and jitter requirements are not stringent. We only care about when the entire file or the entire web page is downloaded. The delays of individual packets don't matter much. Now, the packet loss is not tolerable. For example, if we are downloading a PDF file or a webpage, in that case, even if one byte gets corrupted, the receiver is not able to render the file properly.

So, if the network loses packets, then they must be detected by the source and the source must retransmit those packets. Now, the throughput must be as high as possible for elastic applications. Recall that the average throughput is F/T. The higher the throughput, the lower the value of T, that is, the transfer time is less, so the better it is for the user. So, elastic applications require as high a throughput as possible.

The other kind of applications are real-time applications. Examples are internet telephony and live video streaming. These applications have stringent delay and jitter requirements. For example, in internet telephony, the end-to-end delay should be less than 200

milliseconds. If it is more than 200 milliseconds, then after a user speaks, there is some interval before the other user can hear what the first user said.

So, the quality of the call suffers. So, for good quality of service, the end-to-end delay should be less than 200 milliseconds, and also the jitter should be low. Otherwise, we'll face the kind of problems that we saw in the example. Some packet loss is tolerable. If there is packet loss, then there is some slight disruption in the sound or picture quality, but that is acceptable for users, typically.

And what are the throughput requirements? The instantaneous throughput must be above some threshold. For example, 24 kbps for a voice call of a certain quality. So, any throughput above this threshold is equally good. It is unlike elastic applications for which the throughput must be as high as possible.

So, these are broadly two different kinds of applications, elastic applications and real-time applications, which have different kinds of QoS requirements, which are summarized in this slide. This concludes our discussion of network performance metrics. Thank you.