Network Security

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

Week - 01

Lecture - 05

Review of Basic Concepts and Terminology in Communication Networks: Part 3

Hello, in the previous lecture, we discussed routing and reliable data transfer. We now discuss two other important basic networking functions namely, congestion control and medium access control. We start with congestion control. So, congestion is the phenomenon of too many sources of data sending too much data, too fast, for the network to handle. It is similar to traffic jams in transportation networks.

The effects of congestion are the following. If the total rate at which data arrives at the incoming links of a router, if that rate exceeds the maximum rate at which data can be sent out from the output links of a router, then the queues at the router start growing. So, the packets which go through the routers, they experience long delays. So, that's one ill effect of congestion. Another ill effect is that as these queues grow, eventually they overrun the available space in the router buffers, then the incoming packets have to be dropped because there is no more space to store the incoming packets.

So, packets are dropped, this results in lost packets. In the reliable data transfer part, we discussed that the sender can retransmit packets if some packets are lost, but that results in waste of bandwidth and additional delays. So, lost packets is another ill effect of congestion. As an example, in this figure, there are N users. The first user has source S_1 and destination D_1, and the N^{th} user has source S_N and destination D_N.

All these N users share a 1 Mbps link. This is a bottleneck link. By bottleneck link, we mean that the other links through which the traffic of each source passes. So, this link for example and this link. These other links have plenty of bandwidth, but the bottleneck comes from this link, which has limited bandwidth of 1 Mbps to be shared among all the N sources of traffic.

Now, if the sum of the data rates of the N senders exceeds 1 Mbps, then the total rate at which data comes in at this router is more than the rate at which it can be sent out, which

is 1 Mbps. So, the queue at this router buffer grows. So, this leads to long delays and eventually packet losses. So, because of these reasons, in networks, congestion must be detected somehow and the sources must reduce their transmission rates. This function is known as congestion control.

There are two broad approaches towards congestion control which have been adopted in networks. One is end-to-end congestion control and the other is network-assisted congestion control. In end-to-end congestion control, routers do not provide any feedback to the sender about how much congestion they are experiencing. Then, how do senders infer about how much congestion there is in the network? So, if a sender sends some packets and the acknowledgements come after a long time, then that is an indication that the delays in the network are large, and hence, that the network is congested.

Similarly, if the sender is sending packets and a lot of them are getting lost in the network, then that is another indication that there is congestion. Conversely, if the sender sends data and it is getting acknowledgements very fast, then that is an indication that there is not much congestion in the network and the sender can try sending information at a higher rate. So, in this way, through the end system, observed packet loss and delay, the congestion is inferred, and the transmission rates are adapted to the observed congestion levels. This is the approach taken by TCP, which stands for Transmission Control Protocol. The other broad approach for congestion control is network assisted congestion control.

So, consider the sender S, which is sending information to a destination D over a network. So, the traffic goes through several intermediate routers. These intermediate routers provide some feedback to the senders about how much congestion there is in the network. So, for example, this router R, it provides feedback to the sender S indicating how much congestion it is experiencing. This feedback may be of different types.

It may be a single bit indicating congestion. The bit is 1 if there is high congestion and the bit is 0 if there is low congestion. So, this router might send a single bit indicating to the source whether there is a high congestion or a low congestion, or the feedback can be more detailed. This router can send a field, which might be a natural number, for example. So, the router can send a natural number to the sender indicating how much rate it can support. So, for example, this router might indicate that I can support a rate of 1 Mbps.

So, the source should send information along this path at a maximum rate of 1 Mbps. So, the feedback from the router to the source can be more detailed in this form. So, this approach in which routers provide feedback to the senders, this is the approach taken by

ATM, which stands for Asynchronous Transfer Mode. So, this is another architecture which is an alternative to the TCP/IP architecture. This was proposed earlier, but it did not become very popular.

So, this is the network-assisted congestion control approach whereby routers provide some feedback to the sources. We now discuss TCP congestion control in some more detail. As we said, it's an end-to-end scheme. It is implemented by the sender without any help from routers. There are two components of TCP congestion control.

One is the sender estimates how much congestion there is from itself to the destination. And the second is the rate adaptation, that is, the sender adapts its transmission rate to the observed congestion level. So, let's discuss each of these components in turn. We start with congestion level estimation. So, the problem is that the sender does not know how much free bandwidth there is at the routers along the path from itself to the destination.

When the sender sends a data packet, either its acknowledgement is received or a packet loss is detected. So, the sender needs to infer the congestion level using just this information. The basic idea used is the following. If the sender sends a packet and it is lost in the network, then that indicates that there must be high congestion in the network, and some routers must be running out of buffer space. So, a packet loss indicates a high congestion.

In this case, the sender reduces its transmission rate. On the other hand, if the sender sends a packet and it gets back its acknowledgement for the packet, then that indicates that the network is able to transport the packets, which the sender is sending. So, the sender can try and increase its transmission rate. The network may be able to support even the higher transmission rate. So, when an acknowledgement is received for a previously unacknowledged packet, then the sender increases its rate.

So, the TCP congestion control algorithm is an adaptive algorithm, which adapts its transmission rate to the observed packet losses and received acknowledgements. So, now we discuss the second component. After inferring how much is the congestion level, how does the sender adapt its transmission rate? The sender adapts its transmission rate by adjusting its window. Recall that in the previous lecture we discussed the concept of window.

The window size equals the number of packets that can be sent before the acknowledgement of the first of those packets is received. In particular, if the window size

is N, where N is a positive integer, for example, N equals 10, then the sender can send N packets before it receives the acknowledgement of the first of those packets. So, the sender sends N packets and then it has to wait for the acknowledgements. After it receives the acknowledgements of those N packets, it sends the next batch of N packets, and so on and so forth. Now, let's consider a simple model, where the round trip times are constant.

Recall that round trip time is the time taken for a data packet to be sent from the sender to the receiver and for the acknowledgement to be received at the sender. The round trip time typically varies in networks because of varying queuing delays and other delays. But for simplicity, assume that the round trip times are constant. So, this constant round trip time is shown in this figure here. This is the round trip time.

Assume that there are no corrupted or lost packets and no corrupted or lost acknowledgements. So, what is the average rate if a window size of N is used? We see that in every interval of RTT, N packets are sent. So, N packets are sent in this RTT, then another N packets are sent in this second RTT, then another N packets are sent in this third RTT, and so on. So, N packets are sent every RTT time.

Hence the average rate is N by RTT. This shows us that since RTT is constant, by adjusting the value of N, the sender can adjust the transmission rate. So, TCP's congestion control algorithm adapts the transmission rate by adapting the window size N. So, whenever the sender detects a packet loss, it reduces the value of N, and when it receives an acknowledgement for a previously unacknowledged packet, then it increases the transmission rate, that is, increases the value of N. This shows the trajectory of a congestion window with time. On the x-axis, there is time.

In particular, there is transmission round. So, in the previous figure, this is the first transmission round where the first batch of N packets is sent. Then this is the next transmission round where the second batch of N packets is sent. This is the third transmission round, and so on. So, this is proportional to time.

So, this transmission round is on the x-axis and on the y-axis is the evolution of the congestion window size. So, in this example, the congestion window initially increases rapidly until it reaches a fairly high value. Then the congestion window increases linearly. At this point, there is a packet loss. So, the congestion window decreases to half its previous value.

Then, it again starts increasing as acknowledgements are received for transmitted packets. At this point, there is another packet loss, so the congestion window reduces. Then it starts increasing again. So, the congestion window size keeps on evolving with time. It keeps on changing with time as shown in this example.

So, this concludes our discussion of congestion control. We now discuss another networking function, medium access control. There are broadly two types of communication links. Recall that communication links connect together different routers and systems. The two types of communication links are point-to-point link and broadcast link.

Point-to-point link is conceptually very simple. There is a single sender at one end of the link. There is a single receiver at the other end of the link. When the sender has a packet for the receiver, it just sends the packet over the link. There is no question of acquiring the link for sending the packet.

Examples of point-to-point links are dial-up and DSL. DSL is Digital Subscriber Line. The other type of link is a broadcast link. Here, N nodes are connected to a shared medium as shown here. N is 5 in this example, so, 5 nodes are connected to this medium.

This is known as a broadcast medium, which means that when one node transmits a packet, the packet reaches every other node that is connected to the medium. So, for example, when node 3 transmits, the packet reaches nodes 1, 2, 4, and 5. When node 5 transmits, the packet reaches nodes 1 to 4. So, this is a broadcast medium. Examples of broadcast medium are as follows.

One is Wi-Fi, which is shown in this figure on the right. Whenever one of the nodes transmits, for example, when this node transmits, the wireless transmission reaches all the other nodes. For example, this node, this node, this node, and this base station. Another example of a broadcast medium is a shared cable, which was used in the old version of Ethernet, which is a wired protocol commonly used in local area networks. So, in this case, there are computers which connect to a shared cable.

This might be, say, 1 kilometer long. This cable is 1 kilometer long, for example. And in this example, three computers are connected to this cable. We zoom into the first computer to show its internal details. So, it is connected by what is known as a TAP to the cable.

And the property of this cable is that whatever information is transmitted by one of the nodes connected to the cable, that information propagates everywhere in the cable. So, for

example, if this node sends some information on this cable, then the information reaches this node as well as this node. So, this is another example for broadcast medium. So, now consider N nodes connected to a broadcast medium. If exactly one node transmits at a time, then the packet is successfully received by its receiver.

This figure shows on the x-axis is time. And consider this interval of time when node 1 is the only node that transmits a packet. So, that packet transmission is shown by this rectangle. So, if exactly one node transmits at a time, then the packet is successfully received. But there is a collision if the transmissions of two or more nodes overlap in time.

So, this interval is an interval where both nodes 2 and 3 are transmitting. Then a collision occurs, that is the transmissions overlap in time, and none of the packets can be decoded because of interference. This is another collision. The transmissions of nodes 2 and 3 overlap in time, and none of the transmissions can be decoded by the respective receivers. This is because a packet is transmitted as an electromagnetic waves, and the electromagnetic waves of the multiple transmitters they superimpose at the receiver, and the receiver is not able to decode the individual transmissions.

So, we need a way to make sure that exactly one node transmits at a time. So, we want to maximize intervals such as this one and this one where only one node is transmitting, exactly one node is transmitting. A medium access control protocol is a distributed algorithm that determines how nodes share the channel. That is, it determines when a given node can transmit and when it should not transmit. A MAC protocol should attempt to maximize successful transmissions like this one and this one.

It should attempt to minimize collisions like this one and this one. And it should minimize idle time intervals, such as this one where no node is transmitting. So, the bandwidth is wasted. So, this decision of when a given node should transmit, that has to be taken in a distributed manner. That is, the nodes are at different locations, and they must decide in a distributed manner when a given node should transmit and when it should not transmit.

So, that's a challenge in the design of a medium access control protocol. Broadly, there are two types of MAC protocols. Channel partitioning protocols and random access protocols. There are also hybrids of these two types of protocols. A channel partitioning protocol divides the channels statically into smaller pieces.

For example, these pieces might be time slots or they may be frequency bands. So, the channel is divided statically into smaller pieces and one piece is allocated to each node for

its exclusive use. Examples of such partitioning are FDMA, that is, Frequency Division Multiple Access, which is shown in this figure. And another example is Time Division Multiple Access (TDMA), which is shown in this figure. In the case of FDMA, on the vertical axis we have a frequency, and the frequency spectrum is divided into pieces called frequency bands.

So, this is the first frequency band, this is the second frequency band, and so on. Frequency band 1 is assigned to node 1, frequency band 2 is assigned to node 2, and so on and so forth. This frequency band is assigned to this node. And this allocation is done on a static basis, that is, for example, frequency band 1 is permanently assigned to node 1. So, now this prevents collisions because node 1 will always transmit on this band, node 2 will always transmit on this band, and so on, so collisions are prevented.

In the case of time division multiple access, time is divided into frames of equal durations, and each frame is divided into N slots, where N is the number of nodes, as shown here. And the first slot of every frame is assigned to node 1, second slot of every frame is assigned to node 2, and so on. Again, because of the static partitioning, collisions are avoided. But channel partitioning protocols have a disadvantage that when the bandwidth assigned to a particular node is not being used by that node, then the bandwidth is wasted. For example, when node 1 is not transmitting anything, in that case, this band assigned to node 1 is wasted.

So, that's one disadvantage of channel partitioning protocols. Another type of MAC protocol is random access based MAC protocol. Here, the channel is not divided into pieces, so collisions can occur. So, how are collisions avoided? Each node with a packet to transmit does not transmit it immediately, but it waits for a random duration before transmitting to avoid collisions.

So, this shows the basic principle of a random access protocol. Suppose packets are generated at nodes 1 and 2 at roughly the same time. If they transmit immediately, then there will be a collision. The transmissions of nodes 1 and 2 will collide. If they retransmit, then there will be another collision.

But if each node waits for a random amount of time before transmitting, then there will be a successful transmission with some probability. This example illustrates this. This horizontal axis is the time axis. Packets are generated at nodes 1 and 2 almost simultaneously at this instant, shown by this dashed line. So, as in a randomized MAC

protocol, node 1 chooses a random interval for which it has to wait before it transmits its packet.

Node 1 happens to choose this interval before transmission, and then, it transmits at the end of this interval. Node 2 chooses another random interval independently of node 1. It chooses this interval before it transmits, and then, it transmits its packet. So, in this case, we see that collision is avoided because node 2 transmits in this interval, node 1 transmits in this interval, so collision is avoided. So, that's the principle of a randomized MAC protocol.

Since different nodes choose their random waiting intervals independently of each other, with some probability, nodes will choose very different random waiting intervals, and their transmissions will not overlap in time. So, there will be no collision. This is the principle behind the random access MAC protocol. And that is used, for example, in the case of Wi-Fi. Wi-Fi uses a randomized MAC protocol.

Thank you.