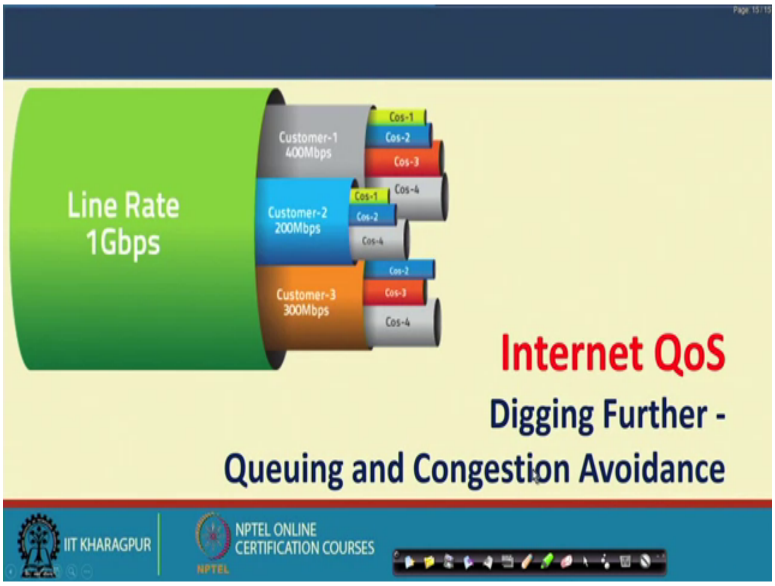
Computer Networks and Internet Protocol
Prof. Sandip Chakraborty
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 34
Internet QoS – IV (Traffic Scheduling)

So welcome back to the course on Computer Network and Internet Protocols.

(Refer Slide Time: 00:21)



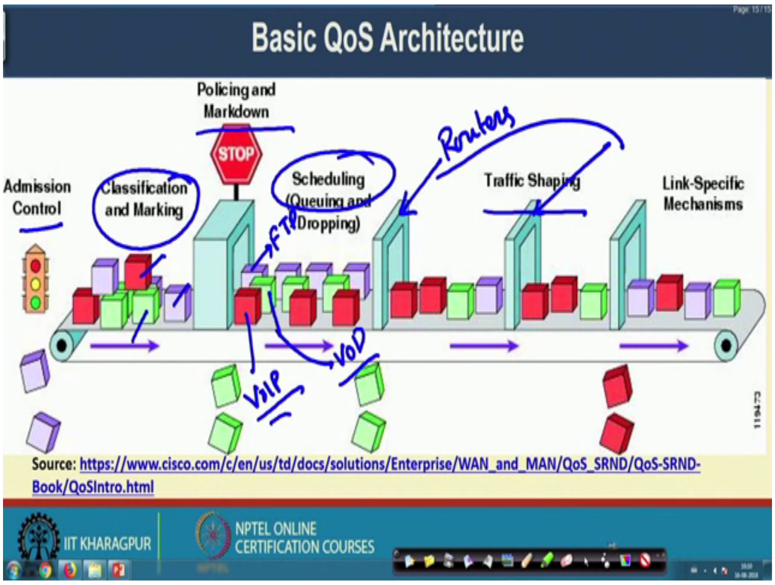
Internet QoS

Digging Further -

Queuing and Congestion Avoidance

So, in the last 3 lectures we have looked into the basic quality of service architecture to provide quality of service over the internet. So, today we will dig again further to look into different kind of a queuing and congestion avoidance strategies, which we apply to provide the quality of service to different applications.

(Refer Slide Time: 00:38)



Basic QoS Architecture

Source: https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/WAN_and_MAN/QoS_SRND/QoS-SRND-Book/QoSIntro.html

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, we start from this point that we have already seen earlier, in the basic quality of service architecture what we have seen that whenever the packets are coming to the network. First you have to use this admission control strategy to enter the flows in the network for which you can support the desired quality of service requirement.

Then we do something called a classification and marking. So, this classification and marking it is used to mark individual packets, based on their quality of service classes like the red packets, blue packets or green packets.

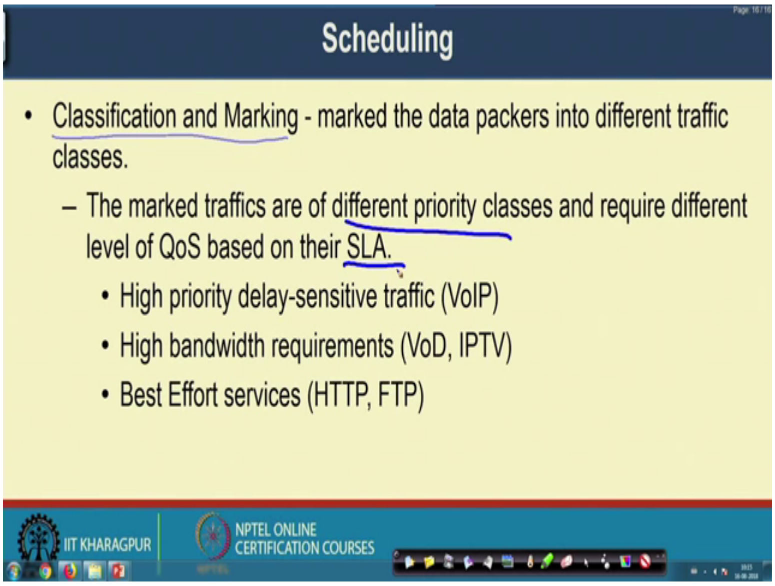
Then we apply something called traffic policing, in case of traffic policing we have seen we have seen that different type of traffic shaping and the policing mechanisms are applied in intermediate routers. So, we apply different kind of traffic policing and different type of traffic shaping mechanism.

Now, in between we have this module which we called as the traffic scheduling. So, what is scheduling actually does? So, whenever you have this different marked packets by like, the red packets, green packets, and the blue packets. So, now different mark packets require different level of quality of service. So, because different mark packets require different level of quality of service, you need to schedule those packets accordingly at the intermediate routers. So, all these different gates that you are observing here you can think of a intermediate routers of the network.

So, in that routers we need to apply the scheduling strategies to ensure the desired quality of service for different mark packets either red packet, green packets and the blue packet. So, you can think of the red packets as the voice applications like this VoIP applications, which require strict quality of service. Say the green packet may be something like a video on demand services, which also require another class of quality of service. It requires more bandwidth and requires a less jitter whereas, this blue packets are normal based for services like this ftp based traffic delivery.

So, you need to give more priority to VoIP than the video and the less priority will be the FTP. So, that way you need to design a scheduling strategy at individual routers so, that different level of quality of service can be ensured of different classes of packets. So, today we will see all these quality of service in details. So, let us start our journey on this quality of service concept.

(Refer Slide Time: 03:17)



Scheduling

- Classification and Marking - marked the data packers into different traffic classes.
  - The marked traffics are of different priority classes and require different level of QoS based on their SLA.
    - High priority delay-sensitive traffic (VoIP)
    - High bandwidth requirements (VoD, IPTV)
    - Best Effort services (HTTP, FTP)

So, to do the traffic scheduling as I have mentioned the first stage is this classification and marking. So, this classification and marking it ensures that the marked data packets they enter into the network and we divide them into different traffic classes like from the users.

Say for example, whenever you are connecting your smartphone to the internet you are enabling the data services over your smartphone during that time. So, if you have certain level of quality of service associated with your service provider, then the smartphone will or the smartphone or your sim card will actually create one service level agreement with your service provider, say the Airtel or BSNL that I want this much amount of service and I am paying you money for this particular service so, treat my packets accordingly.

Now in that case say for example, if you are going to have a VoIP service voice over IP services. So, this voice over IP services is still not very popular in India, but in many countries they are very popular. So, if you are going to support this voice over IP services from your smartphone, then your network service provider at the first half router say the base station where your smartphone is connected it should understand that will you are going to transfer this VoIP data, voice over IP data. Now, whenever it understand that you need to transfer this voice over IP data during that time it marked this particular packets which are coming from your smartphone as the voice over IP data.

Now, remember that in your smartphone you can run multiple applications you can run Facebook, you can run YouTube and at the same time you can run voice over IP applications. Now the network need to understand that this particular application is actually voice over IP application and for that I should give the required resources for ensuring quality of service of that VoIP application. So, that is why this kind of classification and marking is requirements.

So, this classification and marking it marked the data packets into different traffic classes. So, the mark traffics are of different priority classes and require different level of quality of service based on the service level agreements. So, this SLA stands for service level agreement, the service level agreement is something like whenever you are subscribing to a particular network during that time, you say the network service provider that I am going to use VoIP services and you should give me this much amount of data for or this much amount of services you should ensure from your n to transfer my voice data or VoIP IP data over your network.

So, indeed I think you have seen some level of service level agreement, whenever you are purchasing certain packs from Airtel or Vodaphone or any other service providers, you will see they mention something like that will provide you these minutes of free calling send them 100 SMS per day then 1.2 GB of uplink data and 5 GB of downlink data per day. This is one example of service level agreement that you have with your service provider.

So, this is not application level service level agreement, but this is like user level service level agreement. So, whatever data you are going to transfer you will get that 1.5 GB of uplink bandwidth and 5 GB of downlink bandwidth if you have made a service level agreement like that. So, that way all your applications will use that bandwidth, but you can also do that application levels service level agreement to it the network service provider.

So, as I have mentioned just a couple of minutes back that these things are not very popular in India, because you are not using VoIP services right now. And that is why you do not see this kind of application level service level agreements, but once this VoIP becomes popular and our network service provider migrates to the 5 G cellular network and start using or start providing VoIP services possibly we will see this kind of agreements which are coming whenever you are going to purchase some packs from the service providers.

Now, we have this different classes of traffics like it can be high priority delay sensitive traffic like voice over IP, it can be high bandwidth requirement traffic like video on demand or IPTV kind of applications, it can be best effort services like HTTP or FTP. So, best effort service means that whatever bandwidth you get you try to transfer your data over that bandwidth only.

(Refer Slide Time: 08:06)

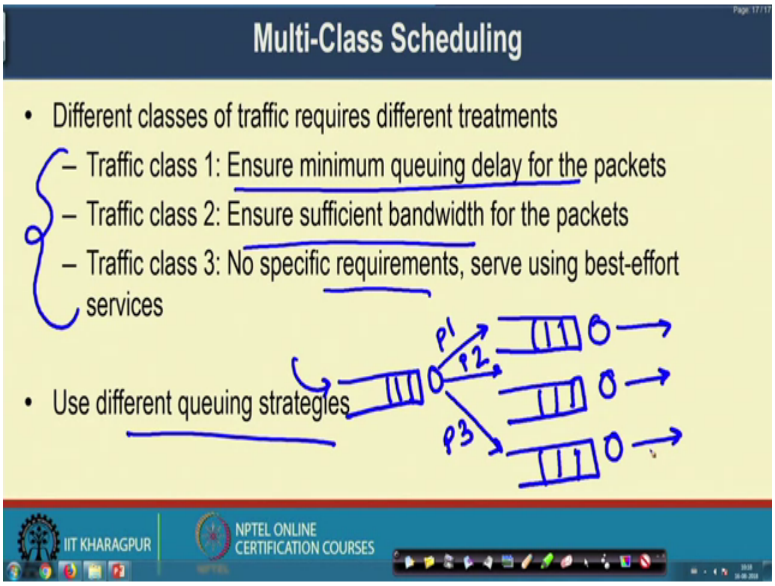
Multi-Class Scheduling

- We call them as different "class"es of traffic
  - Traffic class 1: High priority delay sensitive traffic
  - Traffic class 2: Medium priority bandwidth hungry traffic
  - Traffic class 3: Low priority best effort traffic
- Different classes of traffic requires different treatments

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Now, in case of scheduling whenever you have this different classes of traffic like what we say here that we have multiple traffic classes here, say the traffic class 1 denotes the high priority delay sensitive traffic. The traffic class 2 denotes the medium priority bandwidth hungry traffic and the traffic class denote the low priority best effort traffic. So, these different classes of traffic require different treatments. So, that is why, what we go for is called multi class scheduling. So, this multi class scheduling ensures that different traffic classes are treated differently and you provide the specific services by your scheduling algorithm to the corresponding service class.

(Refer Slide Time: 08:49)



Multi-Class Scheduling

- Different classes of traffic requires different treatments
  - Traffic class 1: Ensure minimum queuing delay for the packets
  - Traffic class 2: Ensure sufficient bandwidth for the packets
  - Traffic class 3: No specific requirements, serve using best-effort services
- Use different queuing strategies

So, the solution is that in case of multiclass scheduling here is a possible solution to do that say because your traffic class 1 was a high priority traffic. So, you ensured a minimum queuing delay for this packets for the packets from those traffic class. So, what we have seen earlier that queuing delay is the dominant component of delay and because of the queuing delay we expect a significant loss in quality of service.

So, for traffic class 1 you ensure minimum queuing delay, for traffic class 2 you ensure sufficient bandwidth because those are bandwidth hungry applications and traffic class 3 it is a best effort traffic. So, you do not have any specific requirements. So, you using you start using the best effort services. So, whatever bandwidth you have you try to serve using that.

So, now, to differentiate among this different traffic classes based on their requirement we used different queuing strategies. So, this queuing strategies ensures that I have multiple different queues in my device rather than maintaining a so, you have a single packet buffer queue where all the incoming packets are getting entered.

Now, from that packet you apply the marking policy, the classification policy to classify the packets into different traffic classes and put the packets into different queues. Say this is a high priority traffic, this is a medium priority traffic and this is the low priority traffic. So, you put it in different queues now different queues will be treated differently based on the their queuing based on their a class requirements are based on their service requirements.

So, in this queue we will apply one scheduling strategy, in this queue we will apply another scheduling strategy and in the third queue we may apply a third scheduling strategy. So, that way we will try to provide the quality of service support for this different classes of services in together. So, that we call as the multi class scheduling.

(Refer Slide Time: 10:54)

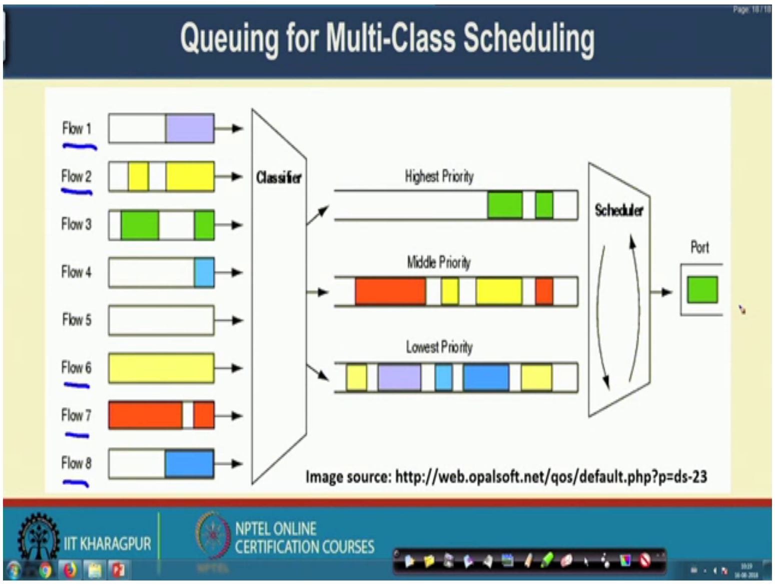

Queuing for Multi-Class Scheduling

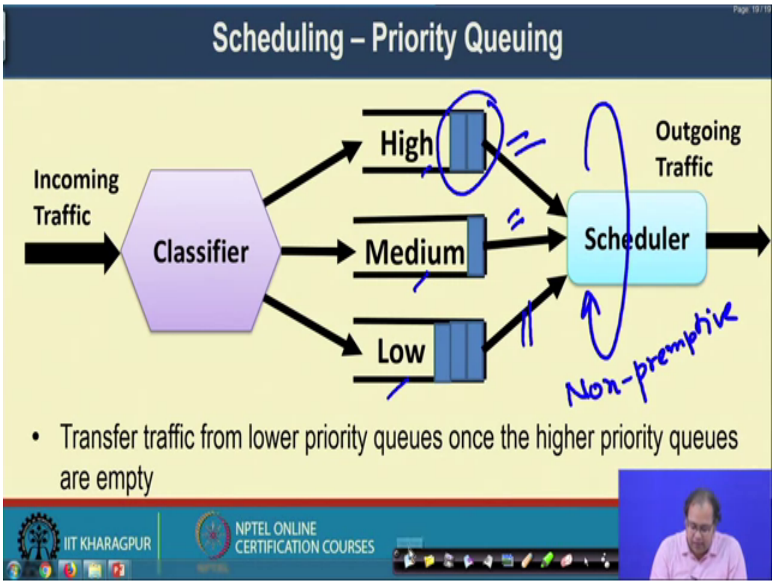
Image source: http://web.opalsoft.net/qos/default.php?p=ds-23

So, here is the example of this multiclass scheduling, the similar kind of figure that I have drawn earlier.

So, you have multiple different flows. So, this from this different flows the classifier identifies that what are the different priority of traffic, say the green flow; that means, flow 3 has the highest priority. So, it is put in the highest priority queue, then you have this red and the red means flow 7 and yellow means flow 2 these 2 different flows which are medium priority traffic. And this blue, indigo and green; that means, flow 6 flow 8 and flow 1 they are the low priority traffic so, you put them in that respectively. Then the scheduler will run over these individual queues and send the traffic based on this scheduling strategy to the output port.

So, we look into this different scheduling strategy in details and this scheduling strategy are based on the queuing principles that we have. So, we look into different kind of queuing principles for a multiclass scheduling.

(Refer Slide Time: 12:06)



Scheduling – Priority Queuing

- Transfer traffic from lower priority queues once the higher priority queues are empty

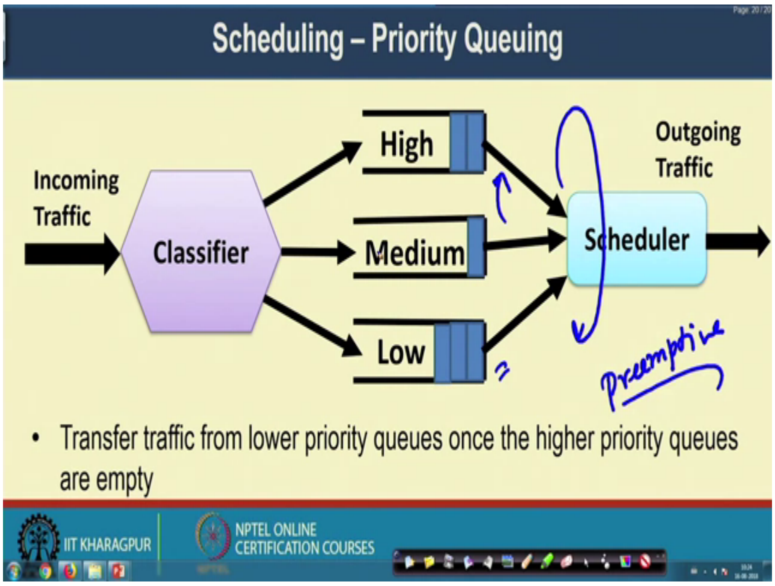
So, the first scheduling that we are going to look into we call it as the priority scheduling. So, what happens in case of priority scheduling, we have multiple queues of different priority. Now we have a incoming traffic, the classifier classifies the incoming traffic and put it into different queuing queues either in the high priority queue or in the medium priority queue or in the low priority queue.

Now the scheduler in case of priority queuing the idea is that if you have some packets in the high priority queue you first serve that packets. So, you first serve the packets from this high priority queue only when this high priority queue becomes empty then you come to the medium priority queue and serve it, when the medium priority queue becomes empty then you come to the low priority queue and serve it.

Now, here the scheduler can be a preemptive scheduler on a non preemptive scheduler in case of non preemptive scheduler you work in a round robin fashion and in that round robin fashion it works in this way that, whenever there is some data packet in the high priority queue you first transfer all the packets of high priority queue. When it becomes empty you come to the medium priority queue transfer all the packets. When the medium priority queue becomes empty you come to the low priority queue transfer all the packets from the low priority queue and then go to the high priority queue the, that we call as the non preemptive scheduling.

So, you are not non-preemptive scheduling. So, in case of non preemptive scheduling what you are trying to do, you are actually applying a scheduling strategy where the scheduler is not preempted or not broken in between. So, it serves this 3 priority queues in a round robin fashion and whenever one particular queue becomes empty it moves to the next priority queue.

(Refer Slide Time: 14:05)



Scheduling – Priority Queuing

- Transfer traffic from lower priority queues once the higher priority queues are empty

The second type of priority queuing strategy you call is the preemptive priority scheduling. In case of preemptive priority scheduling what happens that the scheduler serves in the round robin way, but it may get preempted, preempted in the sense like say for example, it had served all the packets from the high priority queue then it comes to serve the medium priority queue. When it is serving the medium priority queue by that time some packets come to the high priority queue. Then it will preempt the service at the medium priority queue and immediately goes back to the high priority queue and serve the packets from that high priority queue.

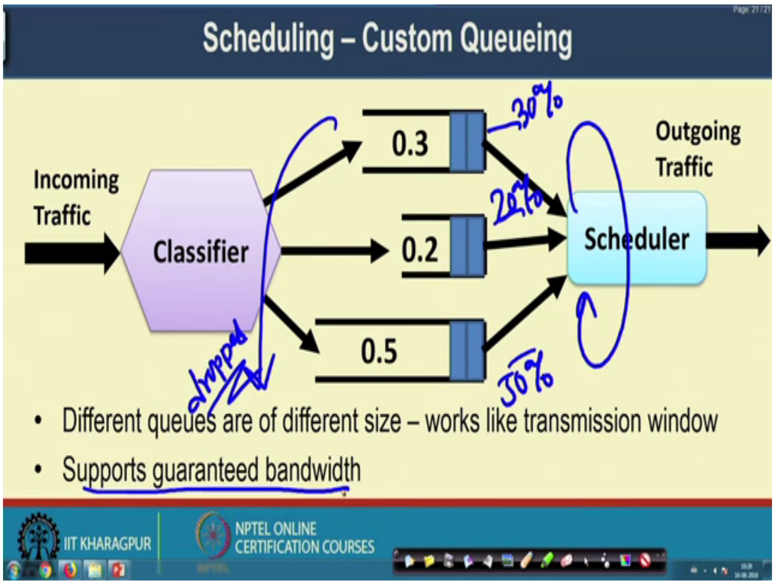
Again when the high priority queue becomes empty it will come to the medium priority queue and then once the medium priority queue becomes empty it will come to the low priority queue. But while serving the low priority queue again if a packet comes to the high priority queue or the medium priority queue. It will preempt the service at the low priority queue immediately you will return back and the serve the packets from that high priority queue or the medium priority queue.

Now, in case of a preemptive service as you can understand that sometime the low priority queue may get start because always you are receiving the high priority queue, high priority packets or the medium priority packets. So, the scheduler is never able to serve the low priority packets, but the advantage is that you are providing very less amount of delay and you are ensuring low jitter for the packets at the high priority queue and the medium priority queue.

So, it is like that whenever you can just think of the high priority queue as the VIP passed line. So, the VIP's need not to wait whenever they are going to that particular queue they are immediately send 2 in side. So, in a airport you can think of that as a as the VIP gate. So, that VIP is coming and they are served immediately. So, they do not need to wait, similarly in the network perspective you can think of certain packets as those VIP packets which who do not need to wait at those gates or the queues they are served immediately. Even there is no one to served and no one to serve them then from other gates people are taken and they are serve those VIP peoples. So, that is the concept which is applied in case of a priority queue.

Now, that is the idea behind priority queue. So, we applied priority queuing to give different priority to different packets say for example, you can think of the network control packets. The network control packets are very high priority packets. So, if you make a delay in the network control packets your entire network operation may get affected. So, that is why whenever certain network control packets are generated immediately those packets are served, if those are the high priority traffic. So, we do not keep them waiting inside the waiting queues. So, this is one type of scheduling strategy.

(Refer Slide Time: 17:01)



Scheduling – Custom Queueing

- Different queues are of different size – works like transmission window
- Supports guaranteed bandwidth

The second type of scheduling strategy that we are going to discuss it called as the custom queuing. So, what happens in case of custom queuing? So, you have different queues of different lengths say for example, I am normalizing the queue length to 1 and in this example the first queue has a length of 0.3 the second queue has a length of 0.2 and the third queue has a length of 0.5.

Now, in this context remember one thing that if you do not have sufficient number of packets and if your network is very lightly loaded then it does not matter actually, then quality of service indeed does not matter because you have a sufficient amount of capacity and everyone will get served within their time bound. But the problem starts occurring when the network capacity is not sufficient and during that time you are going to push the packets in the network.

So, you can just think of the airport scenario whenever it is a non peak time say at the around 2 PM in the noon when there are not much passenger. So, you in you go to the any of the gates you will need to wait for a minimal amount of time. But if you go at the peak hours when there are huge numbers of passengers in the airport then you have to really think about this kind of quality of service. So, you have possibly seen that well during the non peak hours whenever you are going at least that happened to myself a quite a few number of times that I normally prefer flights at the non peak hours and during that time whenever I go to the airport I find that well even I am being allowed through the VIP gates.

So, it is something like that. So, no one cares about what is the quality of service because the load is not very high, but the problem starts occurring when the load is very high and you have certain congestion in the network and during that time you have to really think of that what is happening inside network. So, this particular concept is actually important in the context of this custom queuing, why? Let us see.

So, you have 3 different queues and in that 3 different queues of 3 different plane. So, the first queue has length of 0.2, the second queue has a length of 0 point sorry the first queue has a length of 0.3, the second queue has a length of 0. 2 and the third queue has a length of 0.5. Now, just think of what will happen at the peak hours. So, in the peak hours all the queues are full and what the scheduler is doing the scheduler is simply applying a round robin scheduling.

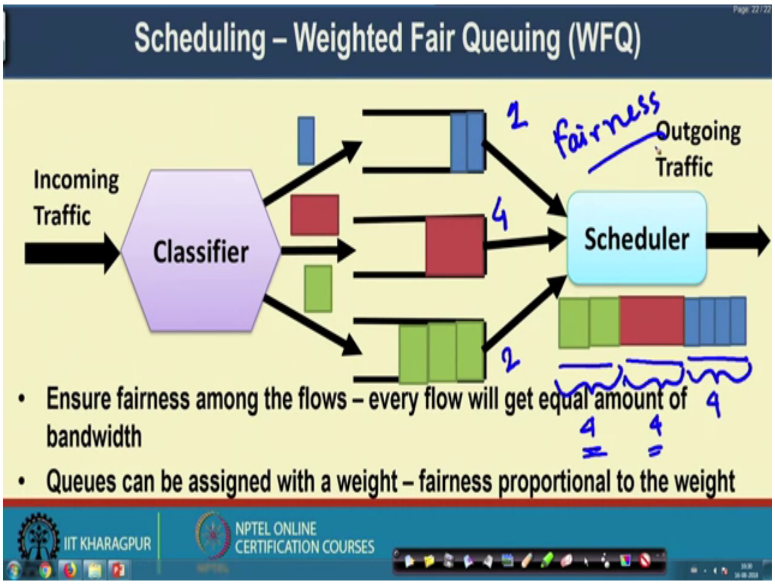
A round robin scheduling means it is just taking one packet from the first queue, then one packet from the second queue, then one packet from the third queue, then one packet from the fourth queue, one packet from the again, one packet from the first queue, one packet from the second queue, one packet from the third queue. Then one packet from the first queue, one packet from the second queue, one packet from the third queue. So, it is scheduling it in a round robin fashion.

But in the peak hours the queues are always full. So, when the peak queues are always full and you are getting certain traffic if you do not have any passage in a queue the packet will actually get dropped. So, at what you are necessarily doing here. So, you are actually providing 30 percent of your capacity to this particular queue, 20 percent of the capacity to this particular queue and 50 percent of the capacity to this third queue.

So, that way this particular custom queuing mechanism where you have different queue length and in the peak hour so, and there are lots of traffics which are coming for these 50 percent queue size it has more amount of spaces it can hold more traffic. So, it can serve more amount of traffic from that particular queuing. And it can serve very less amount of traffic from this 20 percent queue.

So, that way this custom queuing mechanism it supports what we call as the guaranteed bandwidth. So, you can provide guaranteed bandwidth with the help of this kind of custom queuing strategy. So, whenever you require guaranteed bandwidth like this video kind of application you can use custom queuing mechanism.

(Refer Slide Time: 21:26)



Scheduling – Weighted Fair Queuing (WFQ)

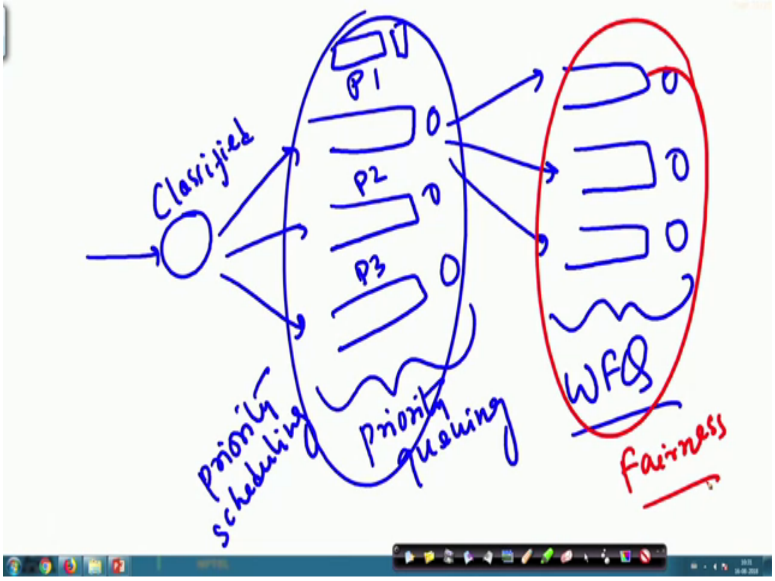
- Ensure fairness among the flows – every flow will get equal amount of bandwidth
- Queues can be assigned with a weight – fairness proportional to the weight

Now, let us see the third queuing mechanism which we call as the weighted fair queuing, again in the case of weighted fair queuing we have 3 different queues. But here you consider that well the packet sizes may vary; in the earlier cases we have considered a scenario when there are fixed packet sizes, but here the packet size can vary. So, what is happening here? You can think of that the blue packets are of size 1 unit the red packets are of size 4 unit and this green packets are of size say 2 units. Then in that case in case of weighted fair scheduling what we try to do? We want to ensure fairness among different classes of traffic.

So, we want to ensure that all these different classes of traffic should get almost equal amount of bandwidth, then what you have to do, you have to transfer 4 packets of one unit then 1 packet of 4 unit then 2 packets of 2 unit. So, you can see that now total amount of blue packet is 4 unit, total amount of red packet is 4 unit and total amount of green packet is again 4 unit. So, you are providing what we call as the fairness in this particular system and remember that normally what we do that we apply multiple queuing strategies together.

So, sometime you require providing priority classes and at the same time you need to provide certain level of a fairness among the priority classes of their different traffic.

(Refer Slide Time: 23:04)

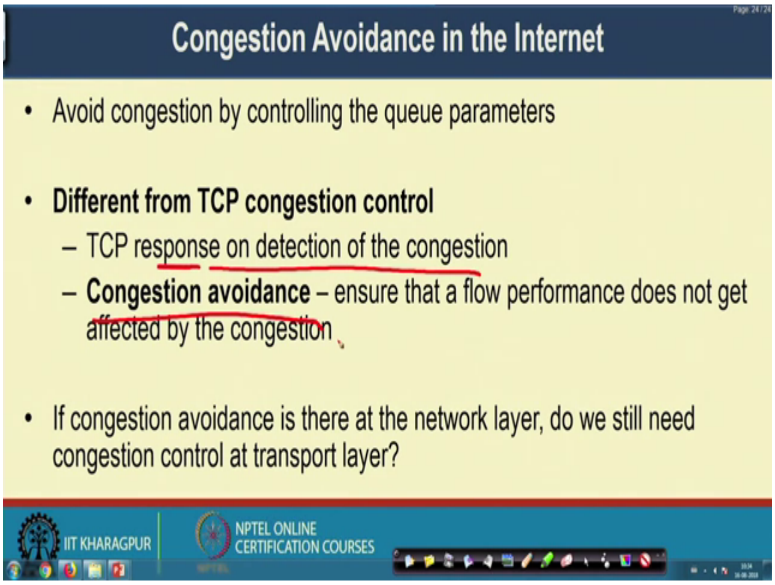


So, in that particular architecture what you can do, that after your packets are getting classified then you put it into different priority classes say this is a priority 1, this is a priority 2, this is a priority 3. Now you know that in priority class 1 you can have different packets of different sizes.

So, here in the first level we are applying say priority queuing, now at the second level say for priority 1 classes it can have different size packets it can have small packets as well as large packets. So, that is why from here you can again apply something called this weighted fair queuing. So, the second level of scheduling can be a weighted fear queuing scheduling.

So, that way we can certain sometime applies multilevel queue scheduling. So, here this first level of scheduling it ensures a priority scheduling whereas, this second level of scheduling it supports fairness in the system. So, that way we will be able to support both priority as well as fairness in your system.

(Refer Slide Time: 24:31)



Congestion Avoidance in the Internet

- Avoid congestion by controlling the queue parameters
- **Different from TCP congestion control**
  - TCP response on detection of the congestion
  - **Congestion avoidance** – ensure that a flow performance does not get affected by the congestion
- If congestion avoidance is there at the network layer, do we still need congestion control at transport layer?

Now, that these are the different type of queue scheduling which we have. Now we look into another interesting concept which we call as the congestion avoidance in the internet.

So, as we have discussed earlier that TCP it does not avoid congestion, what it does that, whenever congestion occurs in the network then it responses on detection of the congestion in the internet. So, what TCP does that, TCP detects congestion based on packet loss and whenever there is a congestion detected then it ensures that a flow performance does not get affected by the congestion and it tries to drop or reduce the sending rate.

So, these congestion avoidance that we are talking in the perspective of internet that is different from TCP congestion control. So, we are not actually controlling congestion rather we are avoiding congestion. So, what we are doing, that we are ensuring that congestion does not occur in the internet. So, this is like before the congestion actually happens we are considering certain measures so that we can ensure that the high priority traffic does not get affected due to congestion.

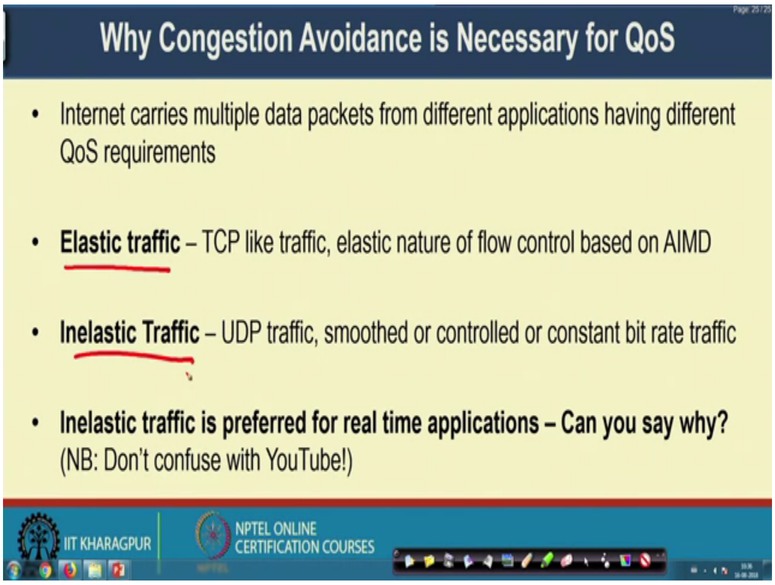
Now one interesting question that you can think of that if congestion avoidance is there in the network layer, do we still need congestion control at the transport layer? Do you still need the service from the TCP?

The answer is yes, we need why we need that particular service because whenever you are applying the congestion avoidance algorithm you will see that we are actually again applying congestion avoidance on class based.

So, we are ensuring that the high priority traffic does not go into the congestion, if atom condition occurs in the net network that should occur under low priority traffic site. Say for example, if you have VoIP services over your internet and at the same time you have FTP services then this congestion avoidance algorithm ensures that the VoIP does not get into the congestion. But well FTP can always get into congestion and in that case you require congestion control algorithm for the TCP which is running the FTP to make FTP come out of the congestion. So, that is the difference between the congestion control and congestion avoidance.

So, in perspective of this congestion avoidance we actually require both in the internet, we require both congestion control and congestion avoidance to support services over the internet. Now, the reverse question is also there say I have already mention that if congestion control is there we also require congestion avoidance to support quality of service. Otherwise the voice traffic or the high priority traffic will also get into congestion. Now, let us see that how we avoid congestion in the internet.

(Refer Slide Time: 27:36)



Why Congestion Avoidance is Necessary for QoS

- Internet carries multiple data packets from different applications having different QoS requirements
- **Elastic traffic** – TCP like traffic, elastic nature of flow control based on AIMD
- **Inelastic Traffic** – UDP traffic, smoothed or controlled or constant bit rate traffic
- **Inelastic traffic is preferred for real time applications – Can you say why?** (NB: Don't confuse with YouTube!)

So, that is a another problem that why congestion avoidance is necessary for quality of service. So, internet carries multiple data packets from different applications having different quality of service requirements and broadly we have 2 different classes of traffics we call them as the elastic traffic and inelastic traffic.

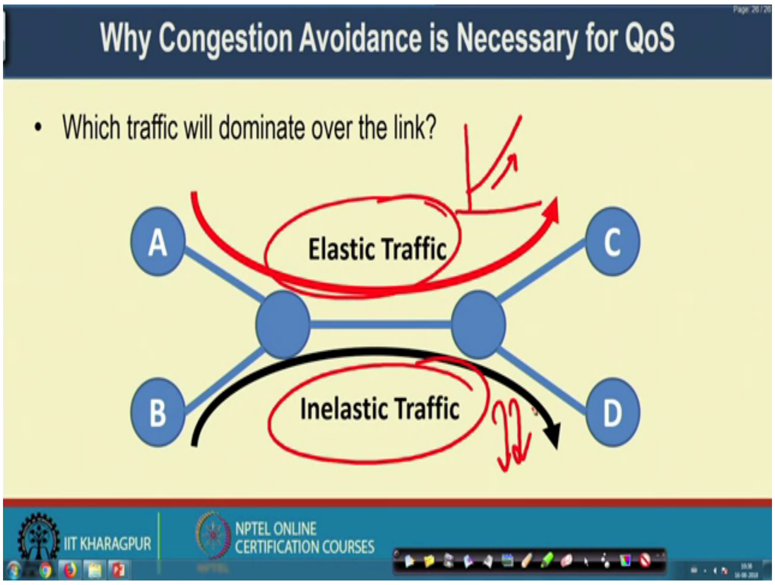
So, this elastic traffics are the TCP like traffic which ensure elastic nature of flow control based on the AIMD principal that we have learned earlier. So, it increases the rate whenever there is no congestion and on detection of congestion it reduces the rate. So, it has certain kind of elastic behavior. So, expand the rate and then reduce the rate, again expand the rate reduce the rate.

In case of in a inelastic traffic they are the kind of UDP traffics they are kind of smoothed or the controlled or constant bit rate traffic. Now, these kinds of inelastic traffic are preferred for real time applications; so why? Because they do not get affected due to the overrate of TCP that we have. So, TCP congestion control is always a overrate further quality of service or associated traffic. So, first of all you can think of that in case of TCP because of this elastic nature you are actually introducing jitter in the network.

Because whenever you have in; whenever you are increasing the capacity we will have less amount of delay, whenever you are dropping the rate you will have more amount of delay for the application data. So, that way by TCP congestion control you are actually introducing jitter in the network. So, that is why for real time traffic there are protocol like real time streaming protocol or the real time protocol, RTP they prefer UDP based constant bit rate delivery.

But do not get confused with YouTube, YouTube is not a real time, YouTube live is real time, but your standard YouTube the thing that you are watching now it is not real time, it is the video has been already recorded and now it is getting streamed.

(Refer Slide Time: 29:37)



Why Congestion Avoidance is Necessary for QoS

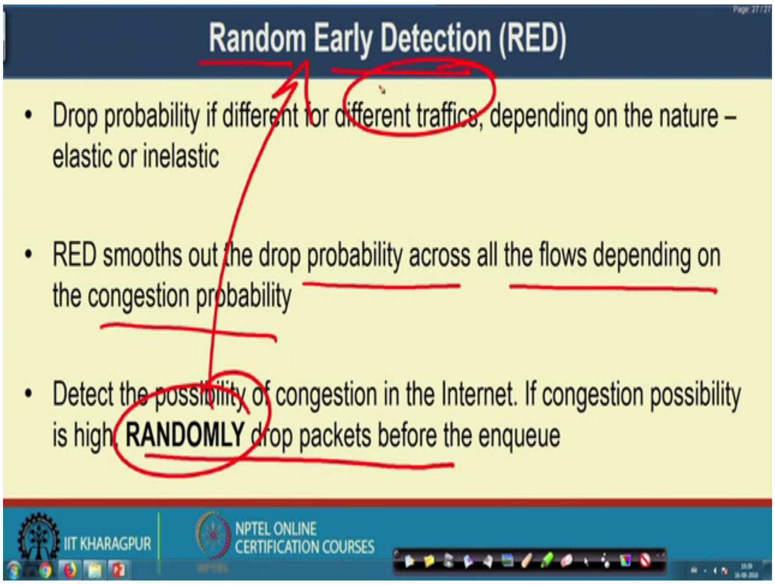
- Which traffic will dominate over the link?

So, just a kind of practice question for you that if you have electric traffic and inelastic traffic in your network which traffic will dominate over the link. So, what will happen here actually, if you have a elastic traffic, the elastic traffic will try to increase it is bandwidth.

So, whenever it will try to increase in bandwidth the inelastic traffic does not have any control over that congestion and you will experience a significant amount of loss from that inelastic traffic. So, there will have a adverse effect whenever you are transferring elastic traffic and inelastic traffic together over the internet and that is why you require congestion avoidance.

We required that the inelastic traffic that we have which is used to transfer a multimedia data they do not get into congestion bit, due to this congestion control algorithm of the elastic traffic because this elastic traffic they will increase the rate, whenever they will increase the rate they will take more bandwidth. But inelastic traffic do not have any control over that. So, you will experience more packet drop from that inelastic traffic.

(Refer Slide Time: 30:43)



Random Early Detection (RED)

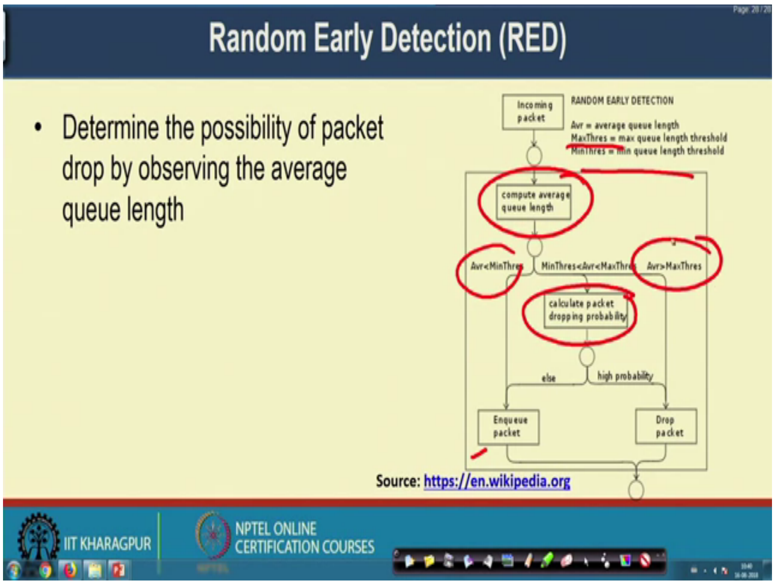
- Drop probability if different for different traffics, depending on the nature – elastic or inelastic
- RED smooths out the drop probability across all the flows depending on the congestion probability
- Detect the possibility of congestion in the Internet. If congestion possibility is high, **RANDOMLY** drop packets before the enqueue

So, the algorithm that we applied for congestion avoidance in the internet we call it as random early detection or RED. So, in case of random early detection what we do first, the first principle is that we drop the packets; we drop the packets for certain applications to avoid the congestion. So, remember that to avoid the congestion the only principle is that if you are expecting that certain application is sending too much traffic which is more than the capacity then you drop the traffic for those applications.

Now, if you drop the packets for those applications. So, what we apply in case of random early detection that this drop probability it is different for different traffic. So, it depends on the nature of the traffic, whether you are using elastic traffic or inelastic traffic or what type of quality of service classes for that particular traffic has. So, this RED it smooths out the drop probability across all the flows depending on the congestion probability. So, it detects the possibility of congestion in the internet, if congestion probability is high you randomly drop packets before enqueueing the packets.

So, this random term is important because of which we call this mechanism as a Random Early Detection or RED. So, there are 2 term random and early detection. This early detection is early detection of congestion we will see how we are applying early detection of congestion and then we apply this random principle to randomly drop the packets.

(Refer Slide Time: 32:15)



Random Early Detection (RED)

- Determine the possibility of packet drop by observing the average queue length

Incoming packet
RANDOM EARLY DETECTION
Avr = average queue length
MaxThres = max queue length threshold
MinThres = min queue length threshold
compute average queue length
Avr<MinThres
MinThres<Avr<MaxThres
Avr>MaxThres
calculate packet dropping probability
else
high probability
Enqueue packet
Drop packet
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

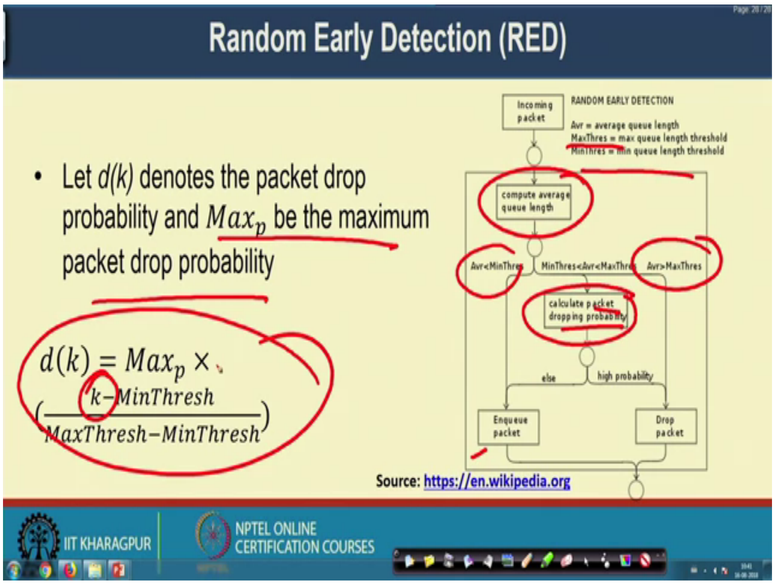
Source: https://en.wikipedia.org

So, here is the principle of RED. So, first you determine the possibility of packet drop by observing the average queue length. So, you have the incoming packet after incoming packet you compute the average queue length, now you have 2 different threshold. One is the maximum threshold, this maximum threshold is the maximum queue length threshold a minimum queue length threshold and you have this average queue length.

Now, if your average queue length is less than the minimum threshold; that means, you are in a safe zone. So, you enqueue the packet if your minimum threshold, if your average queue length is in between this minimum threshold and the maximum threshold; that means, you are going to the danger zone. So, you calculate some packet dropping probability; if the packet drop probability is high you drop the packet otherwise you enqueue the packet.

And if your average queue length is more than the maximum threshold; that means, you are already within the danger zone. So, to avoid the congestion you drop the packet. So, that is the principle of random early detection.

(Refer Slide Time: 33:22)



Random Early Detection (RED)

- Let $d(k)$ denotes the packet drop probability and Max_p be the maximum packet drop probability

$$d(k) = Max_p \times \left(\frac{k-MinThresh}{MaxThresh-MinThresh}\right)$$

RANDOM EARLY DETECTION

Avr = average queue length
MaxThres = max queue length threshold
MinThres = min queue length threshold

Incoming packet

compute average queue length

Avr<MinThres

MinThres<Avr<MaxThres

Avr>MaxThres

calculate packet dropping probability

else

high probability

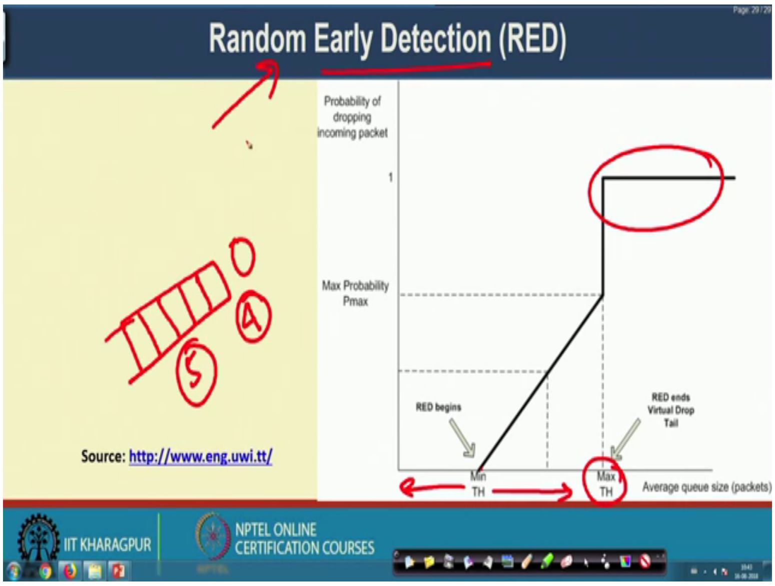
Enqueue packet

Drop packet

Source: https://en.wikipedia.org

So, this is the way we calculate this packet drop probability. So, we have to calculate the packet dropping probability here. So, we calculate the packet drop probability here say assume that Max p is the maximum packet drop probability and d k denotes the drop probability, then d k will be Max p into k minus MinThresh divided by MaxThresh minus MinThresh. So, k is the current queue length. So, we calculate the packet drop probability from the current queue length. Now, let us see what is the significance of this equation.

(Refer Slide Time: 34:00)



Random Early Detection (RED)

Source: http://www.eng.uwi.tt/

Page 29 / 29
Probability of dropping incoming packet
1
Max Probability Pmax
RED begins
RED ends Virtual Drop Tail
Min TH
Max TH
Average queue size (packets)
0
4
5
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

So, to look into the significance of this equation we plot this packet drop probability with respect of average queue sells. So, if you look into the packet drop probability of this average queue size you will see whenever it is less than this minimum threshold your packet the probability is 0. After that the packet drop probability increases linearly based on the equation that we have written the packet drop probability increases linearly.

Now, whenever you are crossing this maximum threshold your packet drop probability becomes equals to 1. So, that is the significance here that as you are moving from this minimum threshold to the maximum threshold you are gradually increasing the packet drop probability and once you have reached to the maximum threshold your packet drop probability becomes 1 and you drop all the packets for that particular application.

Now, here the interesting fact is that what we are doing here, we are ensuring that whenever things are going good we do not do anything, but when things are moving towards the bad side we you do some kind of early detection of a congestion by observing the current queue length, because the current queue length gives you a reliable indication of the congestion. If the queue length becomes side; that means, you have more number of packets in the queue and a queue has say length 5 and you have already filled up the queue with 4 packets; that means you are gradually going towards the congestion.

The moment there will be 5 packets in the queue and the queue become full you will start experiencing congestion. So, that is why as you are increasing the queue length you are moving more towards congestion and accordingly you detect it early based on the average queue length and then randomly drop the packets to ensure that things are going out of congestion.

Now, this random drop has an implication you remember that in case of TCP we detect something as congestion whenever you have 3 consecutive packet loss. So, you are getting 3 duplicate acknowledgements or you are having a time out. Now, if you have a random packet loss then it is just like that one of the packet will get lost and we will get a single duplicate acknowledgement or you will not experience any time out for that particular packet.

So, that way TCP will not get regard a congestion control there, but as you are gradually moving towards congestion you will detect a drop more packet and TCP will trigger the congestion control algorithm at that instants of time.

So, that is all about this congestion avoidance algorithm in the internet, which helps you to come out of congestion or have a early signature of congestion, but as you have seen that as the load increase gradually it moves towards the scenario of congestion and then TCP should come in the picture and run it congestion control algorithm, to make the system come out of congestions. So, to support quality of service we need to run both congestion avoidance as well as congestion control in our system. So, that is all in the next class we will look into 2 specific QoS architecture in the internet call integrated service under differentiated service architecture.

Thank you all for attending this class.