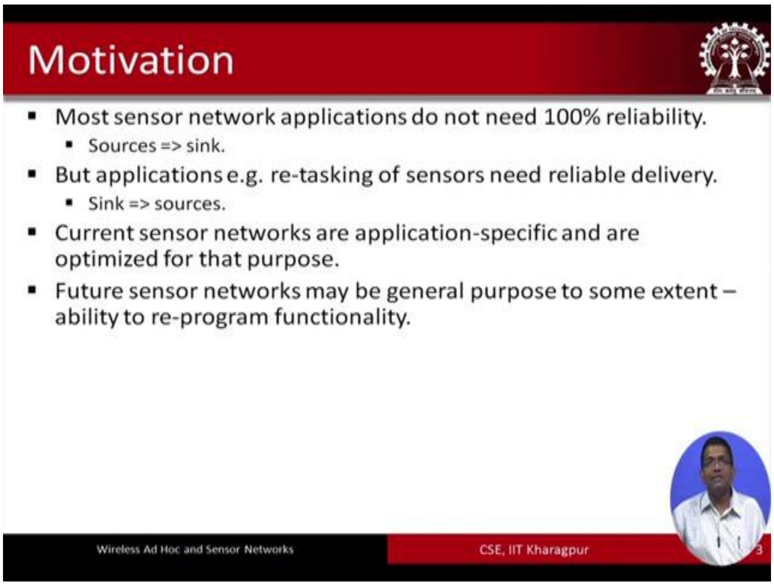
Wireless Ad Hoc and Sensor Networks
Prof. Sudip Misra
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 32
Congestion and Flow Control in Wireless Sensor Networks- Part-II

Now, we are at the second part of the congestion and flow control in sensor networks. We have already, in the first part we have already looked at the issues of congestion and flow control in sensor networks and what are the different solutions that are available the list of all these different solutions and we have already gone through the highlights of two very well known protocols for congestion control, the ESRT and the coder protocol now we are going to look at few other protocols in, and highlights of them in the next few moments.

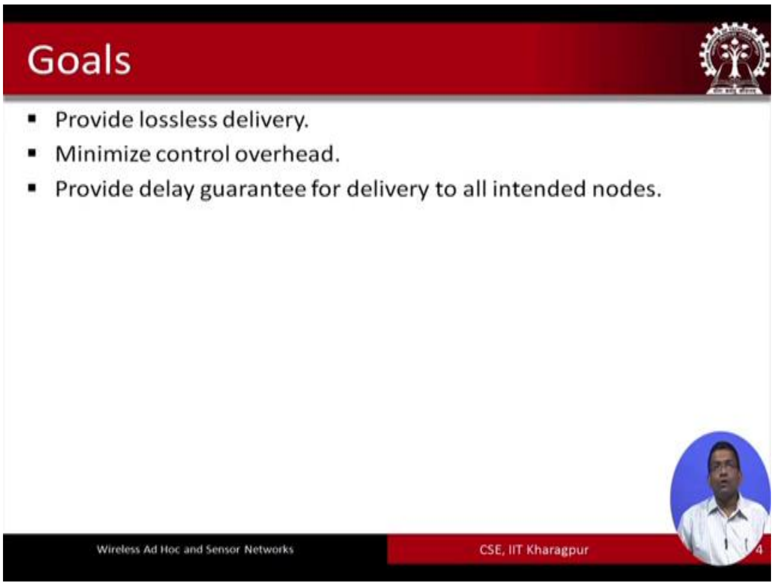
So, let us now start with the next protocol which is the PSFQ protocol. The full form of which is pump slowly and fetch quickly. So, this name basically pump slowly fetch quickly says it all about what has to be done by PSFQ.

(Refer Slide Time: 01:13)



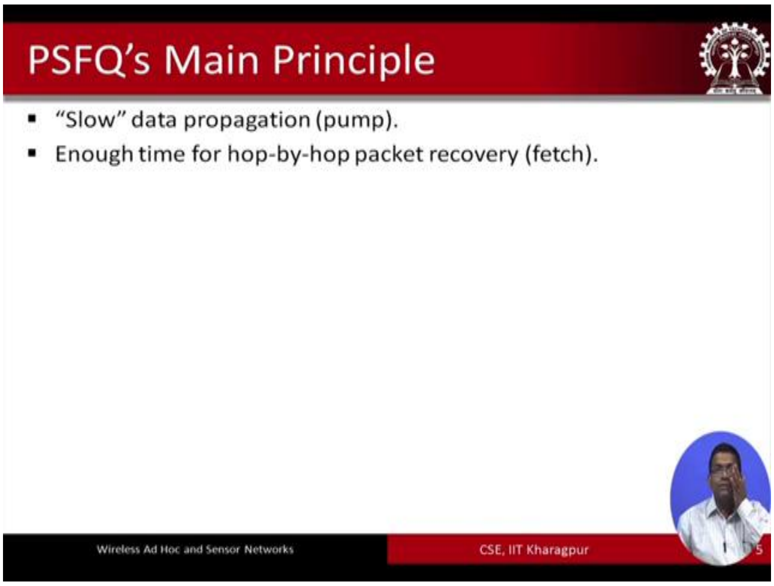
So, the motivation for the PSFQ is a few observations, the observations are like this that the most sensor network applications they do not require 100 percent reliability of packet delivery and, but for applications example the re tasking of sensors they need reliable delivery. So, in both of these cases as you can see that they in the first case the data transmission takes place from the source nodes towards the sink node and in the second case it is from the sink node towards the source node.

(Refer Slide Time: 01:54)



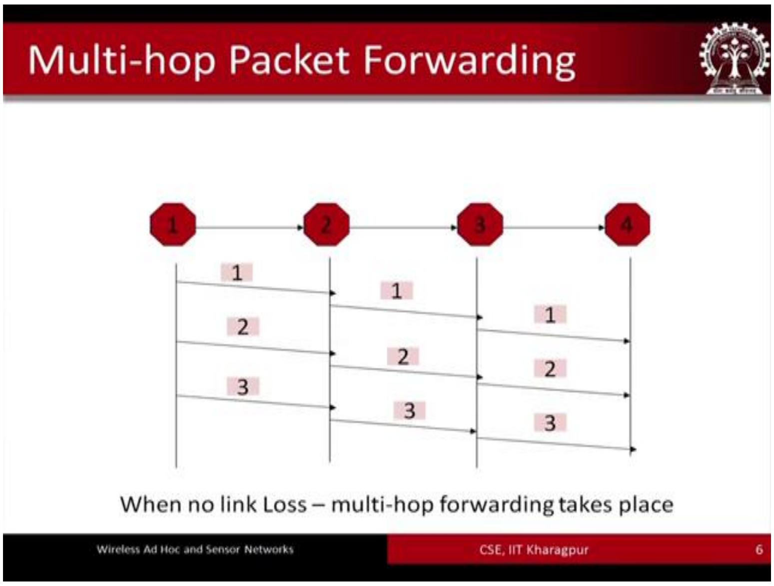
So, the goal of PSFQ is to provide lossless delivery minimize control overhead because you know the more number of control packets you send the more control overhead, more control overhead means more energy consumption and we have a highly energy constant network the sensor networks are highly energy constant and we want to minimize this kind of control overheads. The third goal is to provide delay guarantees for delivery to all the intended nodes.

(Refer Slide Time: 02:24)



PSFQs main principle of working is the slow and the fetch combination, slow is the pumping slow pumping. So, slow data propagation in the network from the source node and fetch is basically to offer enough time for hop-by-hop packet recovery.

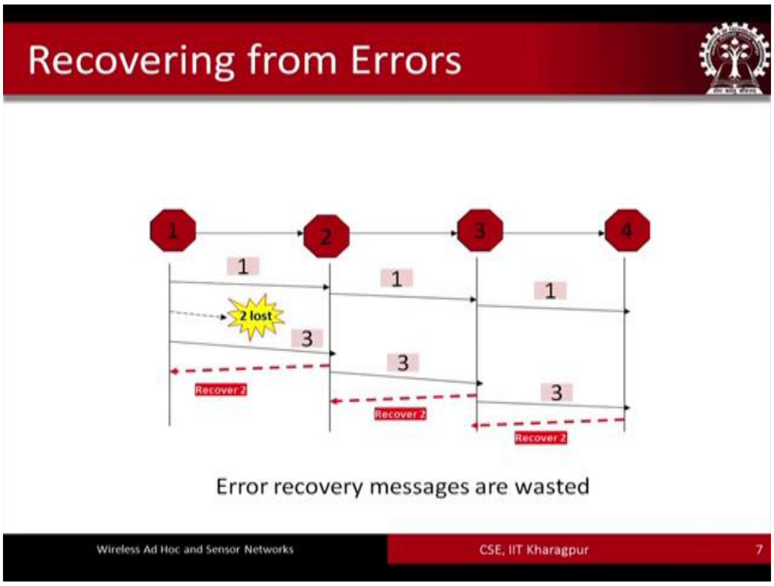
(Refer Slide Time: 02:45)



So, if we first look at the scenario of packet transferred in a multi hop scenario. So, I have been talking about you know the I have been using the PDU term packet, but you know if we are talking about congestion and at the transport layer we are talking about segments. So, in this particular case it is not you know packets, but segments if we are talking about transport layer.

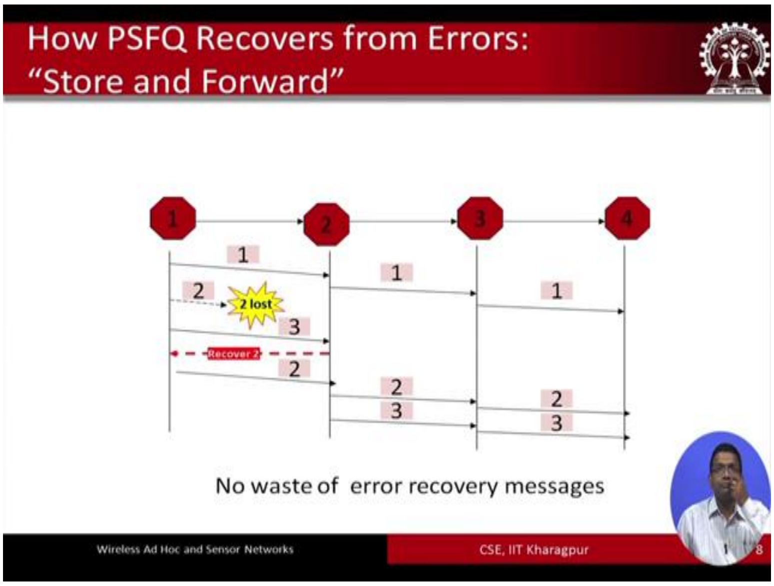
So, but you know so what I am doing is basically I am interchangeably using this name packet or you know segmented etcetera, but basically they all mean the same thing and you know so the concept has not changed at all. So, if we are talking about a multi hop packet forwarding scenario we have something like this as you can see in this figure. So, we have four nodes 1 2 3 and 4 and it is a multi hop packet forwarding scenario from 1 towards 3. So, let us say that this is the, you know this is a scenario where there is no link loss and how packet forwarding takes place. So, this is how packet forwarding takes place. So, first packet or segment 1 is sent to sink node 2 from 2 to 3 from 3 to 4 and so on then for 2 also the packet number 2, packet id 2 also the similar kind of thing for 3 again the similar kind of approach.

(Refer Slide Time: 04:20)



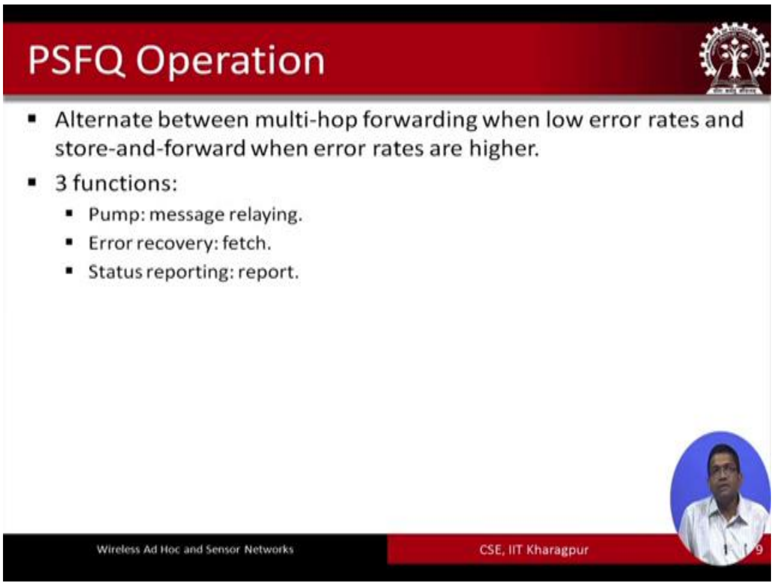
So, now let us say that we consider how it looks like when we have there is some packet loss. So, in this particular example we see that packet 1 is sent 2 is then lost and then when 3 is sent two nodes that after one it was expecting 2 this node 2 knows that after packet 1 it was expecting packet 2 and that has been lost and it has received 3 then it will do what it will explicitly send recovery message for 2 and this process will be followed in the next of nodes as well. So, everybody is going to send a recovery explicit recovery message for two towards the source node to the intermediate nodes like this and that basically leads to you know wastage of different resources too many recovery messages are being sent. So, that is not very ideal.

(Refer Slide Time: 05:31)



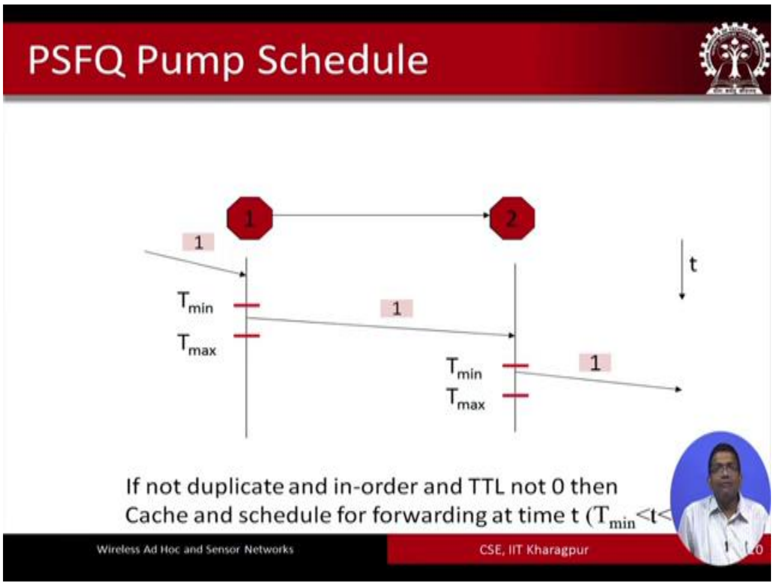
So, the way it can be handled is that if 2 is lost packet 2 is lost and 3 is received. So, this 3 is going to be buffered and then an explicit recovery message like before will be sent towards 1 and only after receiving 2 both the packets 2 that has been received successfully or correctly and the buffered 1 3 will be sent towards the next of neighbour. So, this is how PSFQ basically recovers from errors. So, it stores the packet 3 and then forwards both 2 and 3 on successful receipt of 2 store and forming forward mechanism is followed for discovering from error in PSFQ.

(Refer Slide Time: 06:31)



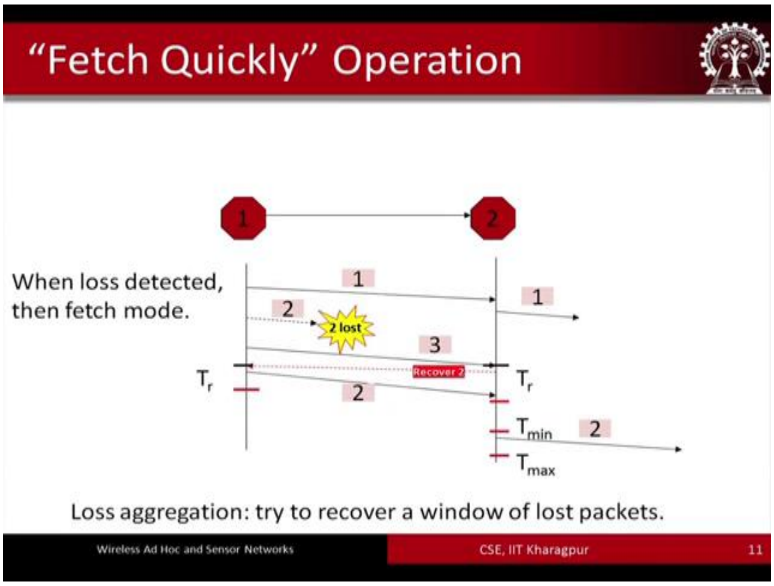
So, there are three functions in PSFQ - pump which talks about how the messages have to be pumped into the network and have to be relate error recovery which is basically the fetch operation and the third is the reporting which is basically about status reporting

(Refer Slide Time: 06:51)



So, we have the PSFQ schedule you know pumps schedule. So, we have a scenario like this. So, one is received then received by node 1, packet 1 is received by node 1 then it will send this packet one within a certain interval bounded by T min and T max. And the same thing is going to be repeated at node 2 as well, 1 is going to be forwarded packet 1 is going to be forwarded by node 2 within a certain bound T min T max. So, if the packet is not duplicated and packet received are in order and TTL is not 0 then cache and schedule for packet forwarding at time T and that T is bounded by T min and T max.

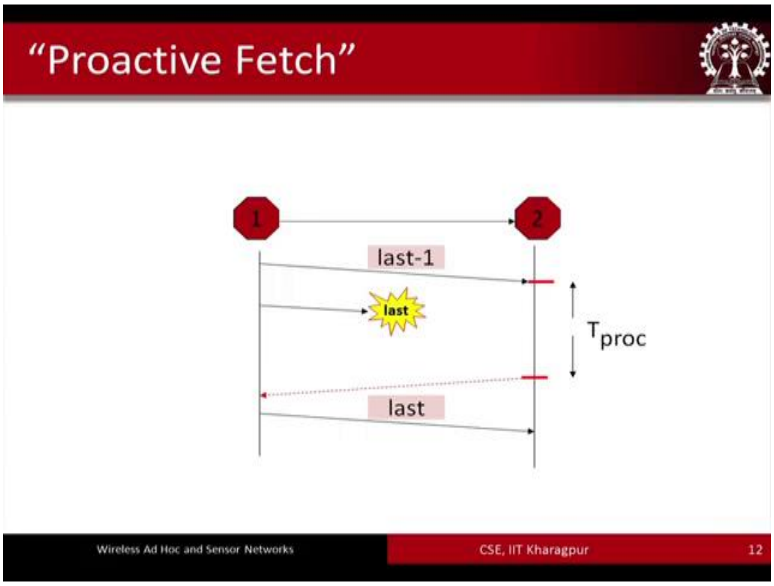
(Refer Slide Time: 07:53)



When loss is detected then it is in the fetch mode. So, this is how it looks like in this particular mode. So, this node 2 explicitly fetches the packet on packet 2, on receipt of node 3 and non receipt of sorry packet 3 and non receipt of packet 2.

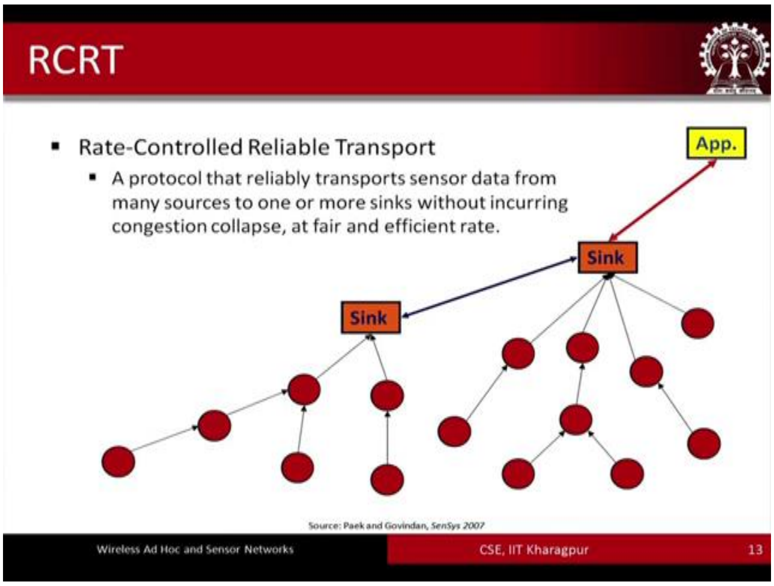
So, then explicitly this will be fetched on receipt of the package to at node 2 this will be sent forward within a certain time bound of T min T max. So, losses like this are going to be aggregated through the fetch operation and sent this is what PSFQ tries to do using their protocol.

(Refer Slide Time: 08:57)



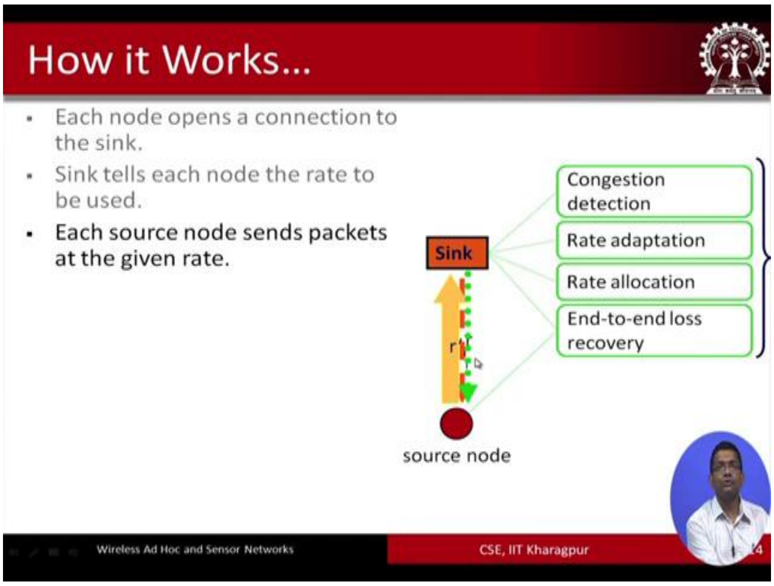
Now, let us look at another suggestion by PSFQ which is a proactive fetch. In this mechanism as we can see that the last packet was not received and then you know this node it will wait for certain duration of time T proc it will timeout and then it will send an explicit request proactive fetch request for the last packet.

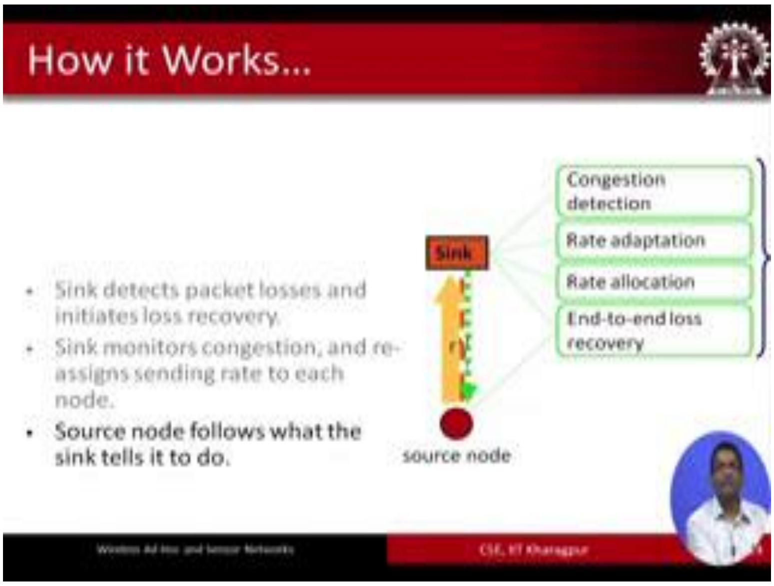
(Refer Slide Time: 09:30)



Now, we come to the last protocol that we are going to discuss in this course. So, we the last congestion control protocol it is called the RCRT protocol, it is a rate controlled reliable transport protocol. This protocol it reliably transports sensor data from many sources to one or more sinks without incurring congestion collapse at fair and efficient rate. So, in this particular protocol we are considering that not only one sink you can it will have multiple sinks as shown in the figure in front of you.

(Refer Slide Time: 10:10)

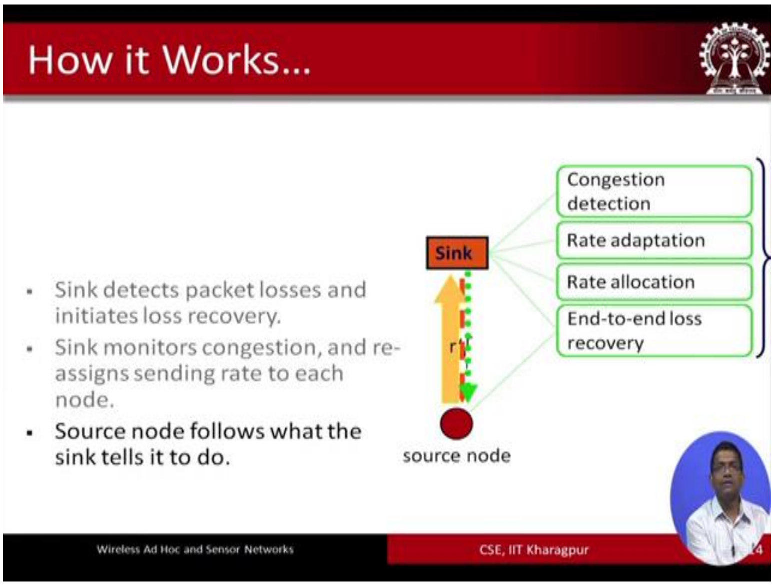




So, how does it work? This particular protocol how does it work? So, in this protocol as we can see there are few things that are controlled - first thing is that we have a source node, then we have a sink node. So, this, the source node basically reports the data towards the sink node at a particular rate maybe r prime and the sink also sends the requests in the other direction towards the source node in this manner. So, the sink node is particularly involved in these activities congestion detection, rate adaptation and rate allocation and also end-to-end loss recovery. The source node is also involved in the end to end loss recovery. So, unlike congestion detection rate adaptation and rate allocation which are basically controlled by the sink the end-to-end loss detection loss recovery is done by both the sink as well as the source node.

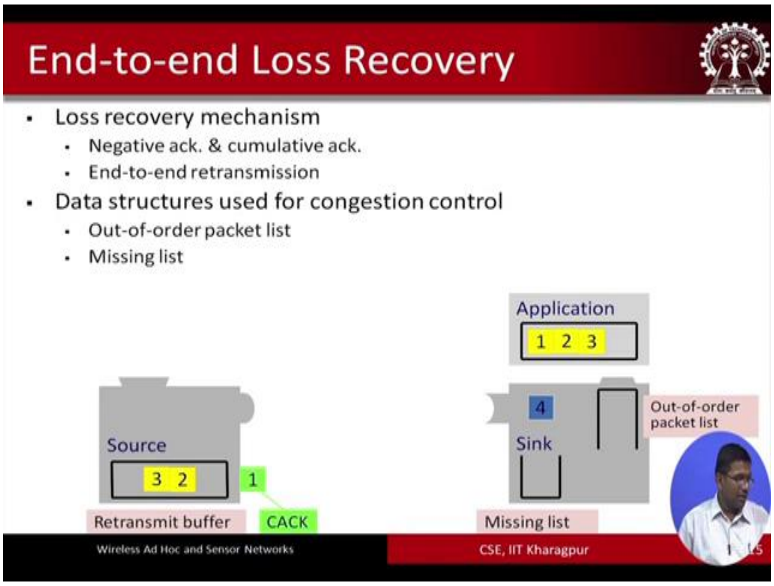
So, each node in this particular protocol it opens the connection to the sink first you know like this manner it opens the connection towards the sink. The sink tells each node the rate that has to be used the sink will tell, so this green colored arrow shows that the sink is informing about. So, first of all what happens is the source node opening a connection to the sink will tell the source node that this is the rate that you are going to use for sending the data. Each source node sends the packets at that particular rate only. So, this is that rate you know that r prime. So, that r prime rate was once again informed by the sink to the source node that you should send at this rate only.

(Refer Slide Time: 12:22)



The sink detects the packet losses and initiates loss risk recovery the sink monitors congestion and re assigns the sending rate to each node. The source node follows what the sink tells it to do.

(Refer Slide Time: 12:35)



So, how the end-to-end loss recovery is done. So, this is done using couple of mechanisms. So, let us look at few of these features - the loss recovery mechanism is done with the help of two acknowledgments, one is the negative acknowledgement the one is the negative acknowledgement and the other one is the cumulative acknowledgement. So, two acknowledgments negative acknowledgement is the nack and the cack cumulative acknowledgement are taken care of, you know the help of these are taken care of in order to recover from loss. And the second is the end to end retransmission mechanism in the end-to-end retransmission mechanism we will see that you know end to end a packet is going to be retransmitted if you know there is some kind of loss it has been detected.

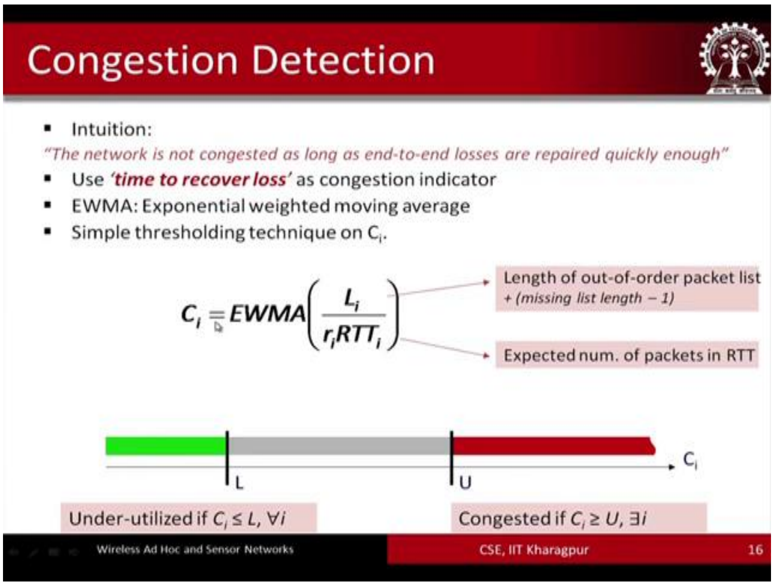
So, there are few other things also, there are two specific data structures that are used by this particular protocol - one is the out of order packet list, the second one is the missing list. So, these two are maintained these lists are maintained by this particular protocol. So, out of order packet list is about the packets it is a list of packets which have been received out of order. So, this list is buffering those packets and missing list is that it was expecting some packet and it got missed. So, the missing list has the missed list of all these missing packets.

So, let us now say look at how this particular protocol functions. So, what we have just witnessed is that a packet one was sent by the source node, it was sent by the source node and it was sent via the sink node to the application which is running on this particular network. So, the application is (Refer Time: 14:37) through the sink node right. So, as we can see that once it has been sent it is also buffered at the source in the retransmit buffer, in the retransmit buffer it has been buffered at the source node. This is done by this particular protocol thinking that once it is sent it does not know that the packet is going to be successfully delivered to the sink and the application or not, it might get it might get dropped in between. So, it basically once it is sent it will also keep a copy of it in the retransmit buffer. The sink node what it will do is it will now, you know it will now expect once it has handle two handle packet one it will now expect the second packet.

So, the second packet is sent and let us say that it was dropped before it was received at the sink node. Source node had sent packet two, but it got dropped in between and was not received by the sink node. Now mind you that the sink node is still expecting packet two to arrive, now you see that what has happened in the retransmit buffer of the source node in the retransmit buffer you have one and then two. Now packet 3 comes it is it has again been appended in the retransmit buffer now this packet 3 it was successfully delivered to the sink node, but the sink node basically does not accept it as such and it does not send to the application layer. So, what it does is it will first send 2 to the missing list and then send 3 to the out of order packet list because it was you know it did not accept 3 as such because it was expecting packet 2 which did not arrive. So, once three is sent and has been successfully received by the sink node it is sent to the out of order packet list and the missing list is populated with 2.

Then from the sink towards the source a cumulative acknowledgement for 1 is sent and a negative acknowledgement for 2 is sent. Then once that is done you see as the animation showed that the source basically then sent packet 1 and then in both the missing list and out of order packet list at the sink node the packets 2 and 3 they were finally, delivered from those lists to the application.

(Refer Slide Time: 17:40)

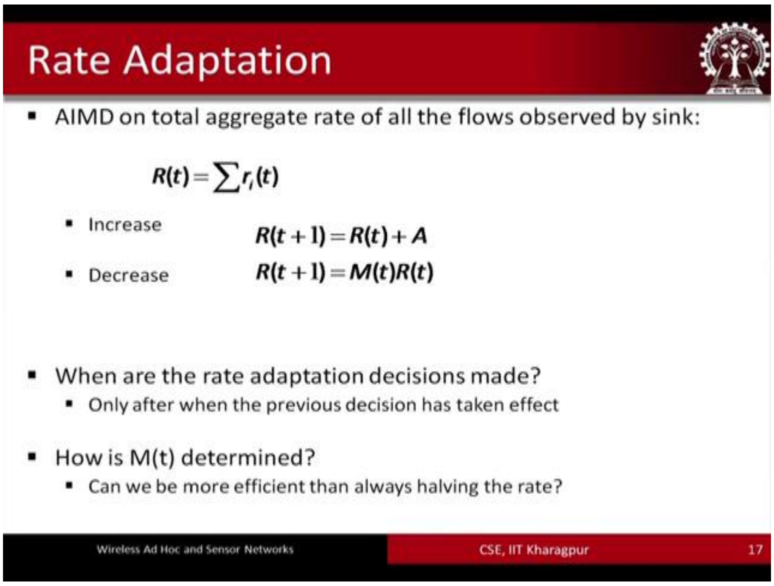


And then this process continues and what we have is a control mechanism by which successfully error free the packets can be delivered from the source node to the sink node and finally, to the application which is running on top.

So, the congestion detection is done through an intuition that the network is not congested as long as end-to-end processes, end to end losses are repaired quickly enough. It uses this particular protocol uses a time to recover loss as a congestion indicator. So, you know, what it does is it keeps track of and a metric which is the exponential weighted moving average. So, exponential you know, if we look at this particular formula. So, you know, L i divided by r i times RTT and an exponential weighted moving average of that is kept track of where L i stands for the length of out of order packet plus the missing length list length minus 1 and this product of r i times RTT i denotes the expected number of packets in the particular time RTT.

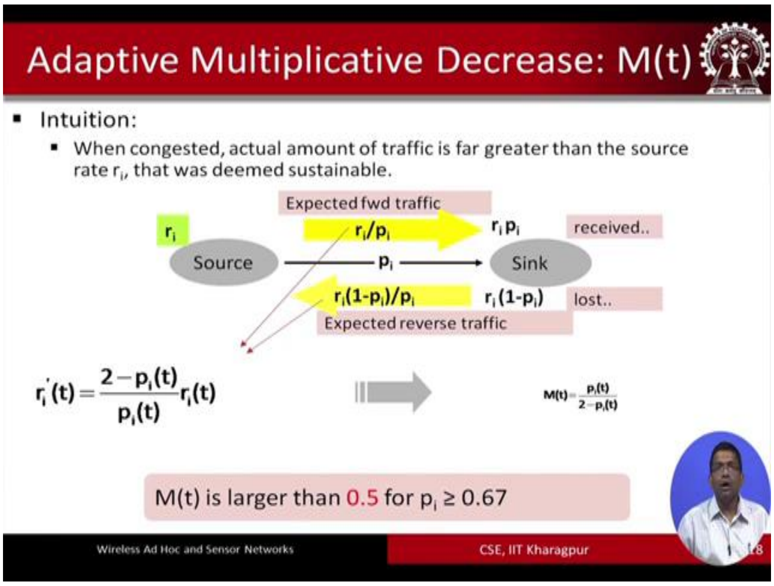
So, what we get is this expected weighted moving average and this value gives the this parameter C i know it is denoted this moving average is denoted as C i. So, this particular protocol says that you maintain two thresholds one is the L the other one is the U. So, two thresholds are maintained L and U lower and upper and to denote how much is the congestion C i. So, it is saying that the channel is underutilized if C i is less than equal to channel utilization is less than or equal to the lower threshold and it is congested the channel is congested if C i is greater than or equal to upper threshold.

(Refer Slide Time: 20:11)



This protocol it adopts the AIMD rate adaptation technique, AIMD stands for additive increase multiplicative decrease. So, as we can see through this formula the rate is adaptively increased in this manner and it is decreased in this manner. So, if there is some kind of congestion etcetera etcetera the rate is going to be multiplicatively decreased and this is the multiplication factor empty. So, when are the rate adaptation decisions made? So, these are made only after when the previous decision has taken effect and the next question is that how is this empty this multiplicative factor determined. This particular factor can be determined in this particular way, let us see how.

(Refer Slide Time: 21:12)

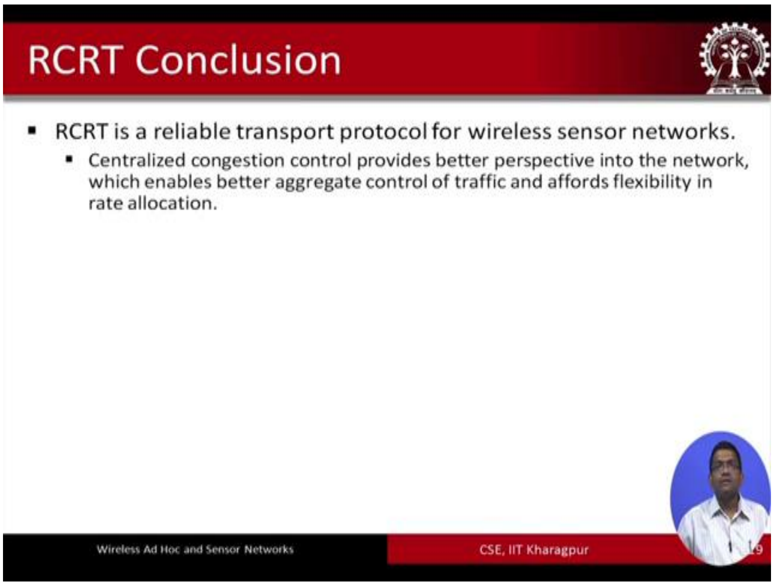


Let us see we have a source and we have a sink the scent. So, when the channel is congested the actual amount of traffic is going to be far greater than the source rate r i that was deemed sustainable. So, let us say that the source sends at the particular rate r i towards the sink and the probability of the packet that is sent by the source to the sink being delivered at the sink (Refer Time: 21:49) p i. So, the number of packets intuitively, number of packets over a time interval that are received at the sink is conceptually the product of r i and p i these are the total number of packets and received at the sink successfully the ones that are lost are r i times 1 minus p i and that is quite obvious because p is the probability of successful delivery of a packet. So, r i times 1 minus p i will be the number of packets at a lost.

So, they you know. So, what we do over here is calculate we means this particular protocol what it does is it calculates two things one is expected forward traffic which is basically r i over p i and the expected reverse traffic which is r i times 1 minus p i over p i. So, the sum of these if we take we get this particular term r i prime t which is equal to r i by p plus r i times 1 minus p by p. So, this one times r t. So, basically these two sums will lead to r i over p plus r i times 1 minus p i divided by p will give 2 minus p i t by p i t times r i.

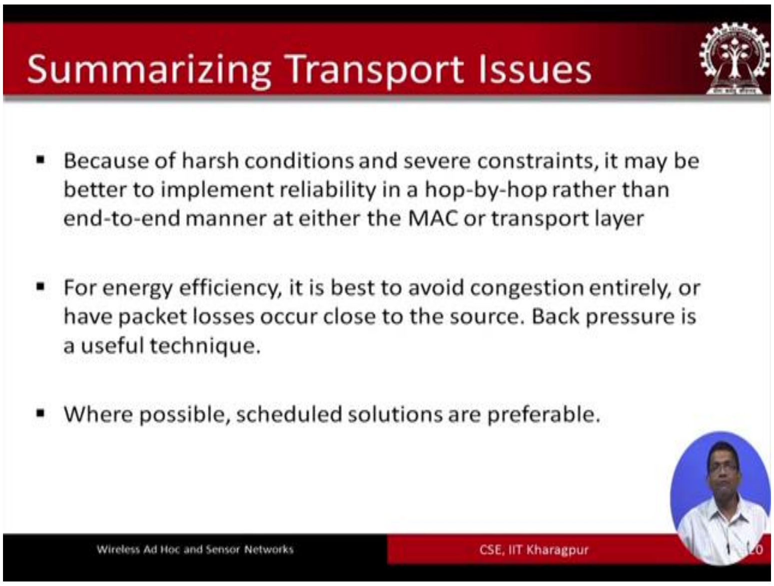
So, these particular entity 2 minus p i t by p i t is basically defined as 1 over M t and this is you recall that this is the factor the multiplicative tag factors that we attempted to calculate. So, this will give 1 over, this factor will give is the 1 over M t. So, in other words M t will be equal to p t over 2 minus p t and it has been analytically shown in the paper that if M t is larger than 0.5, if M t is larger than 0.5 for p greater than or equal to 0.67. So, this is something that has been established in the paper and we do not need to know how.

(Refer Slide Time: 24:19)



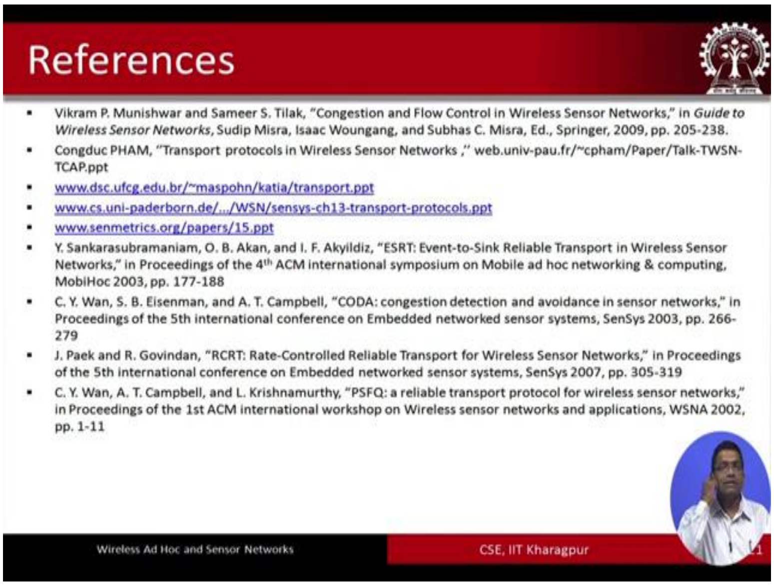
So, few concluding remarks RCRT is a reliable transport protocol for wireless sensor networks. Centralized congestion control provides better perspective into the network which enables better aggregate control of traffic and a force flexibility in rate allocation.

(Refer Slide Time: 24:37)



And here are a few you know summarized points about transport because of the harsh conditions and severe constraints it may be difficult to implement reliability in a hop-by-hop manners rather than end-to-end manner at either the MAC or the transport layers. For energy efficiency it is best to avoid condition entirely or have packet losses occur close to the source back, pressure is a useful technique when possible schedule solutions are preferable.

(Refer Slide Time: 25:10)



So, here at the references once again, these references you had also seen when we saw the first part of way at the end of the first part of congestion control in sensor networks and as I said in the first part as well here as well some of these slides and the particular simulation animations have been adopted from different sources, the sources are mentioned over here, some of these PPTs are useful for this particular purpose for getting access to these different resources the slides that we have talked about. Some of the contents with minor modifications we have adopted from these particulars references.

Thank you.