

Bandwidth Estimation Algorithm of WestwoodNR for Wireless Network

Pandya Vyomal N^{1*}, P. Rahul Reddy², Abhishek Choubey³

¹Associate Professor, Dept. of ECE, Sreenidhi Institute of Science & Technology, Ghatkesar, India.

²Associate Professor, Dept. of ECE, Sreenidhi Institute of Science & Technology, Ghatkesar, India.

³Associate Professor, Dept. of ECE, Sreenidhi Institute of Science & Technology, Ghatkesar, India.

*Corresponding author E-mail: vyomalpandya@sreenidhi.edu.in

Abstract

Two widely known parameters of Transmission Control Protocol (TCP) used to control the flow of packets are Congestion Window (*cwnd*) & Slow Start Threshold (*ssthresh*). After congestion, slow start phase or fast-retransmit phase come in action wherein TCP has an important role in the reduction of these parameters. This is in response to packet loss identified by TCP. This in turn will cause unnecessary reduction of data flow & degradation of TCP throughput. Researchers have developed some algorithms to come out of this problem, WestwoodNR is one of them. WestwoodNR is using Bandwidth Estimation algorithm to estimate available bandwidth, to make effective use of available network capacity even after the congestion episode. It allows higher values of *ssthresh* & *cwnd* when it enters the fast-retransmit phase and slow start phase. In turn this algorithm claims better performance in terms of bandwidth utilization. The focus of this paper is on error recovery mechanisms suitable for WestwoodNR operating over the wireless sub path. These mechanisms have to address the increased bit error probability and temporary disruptions of wireless links. The efficiency of WestwoodNR within wireless scenarios is investigated and possible modifications that lead to higher performance are pointed out.

Keywords: *cwnd*; *ssthresh*; TCP; WestwoodNR

1. Introduction

TCP Congestion Control Evaluation Policy

Throughput

Throughput refers to how much data can be transferred from one location to another in a given amount of time. It is used to measure the performance of hard drives as well as Internet and network connections.

Goodput

The maximum throughput is often not reliable parameter of used bandwidth, for example the file transmission data rate in bits per seconds. The achieved efficiency is many times lesser than the maximum throughput. And, the protocol extra bits may affect the available bandwidth. The throughput is a metric which fail when it comes to how to deal with protocol overhead. It is typically measured at below the network layer and above the physical layer.

The simplest word are the number of bits per second that are physically delivered. Ethernet network is a typical example where this definition is proven. In this case the maximum throughput is the gross bit rate or raw bit rate. To regulate the actual data rate the "goodput" measurement definition may be used. For example in file transmission, the "goodput" corresponds to the file size (in bits) divided by the file transmission time. The "goodput" is the amount of suitable data that is handed over per second to the application layer protocol. Dropped packets or packet retransmissions as well as protocol overhead are excluded. Because of that, the "goodput" is lower than the throughput.

Transmission Efficiency

Transmission Efficiency can be defined as the ratio of actual transmitted data packet to the successfully transmitted data packet. This will indicate the unnecessary retransmission of the data packet to deliver the actual data packets. As TCP will fail in congestion control algorithm it will increase number of retransmitted data packet. This will not allow TCP to utilize the available bandwidth. Transmission Efficiency will be defined as under.

$$\eta = (\text{Total Number of Packet Transmitted}) / (\text{Successfully Transmitted Packet})$$

Friendliness

When numbers of TCP connections are established on a single link, all TCP connections tries to use equal part of the available bandwidth. This tendency of TCP is called as friendliness. If the transmission efficiency of TCP is lower than it will also affect this factor of the performance measurement.

Fairness

When any other traffic and TCP traffic is flowing through a single link than TCP tries to take an equal part of the available bandwidth this is called as fairness of TCP.

Challenges to TCP over Wireless Networks

TCP is basically designed for wired network. Present scenario is, all networks are using wireless link to communicate. Wireless link has some parameters which are affecting TCP performance widely. Few of them are shown below.

Packet Error Rate (PER)

The packet error rate (PER) is the ratio of packets that are dropped to the overall number of packets received in a transmission, usually conveyed in ten to a negative power. For example, a transmission might have a PER of 10 to the minus 6, meaning that, out of 1,00,000 packets transmitted, one packet was in error.

$PER = \text{number of erroneous packets} / \text{number of transmitted packets}$

Link Delay

Link delay is the delay of the link used between sender and the receiver. This delay will also affect TCP performance. As TCP working according to the rate of ACK packet comes. So, if delay is higher than ACK will come with slow rate and because of that TCP will not be able to send data with higher rate.

Data Rate of Link

Number of bits transferred through link within a second that is called as data rate of link. As data rate of any link is low it will not allow TCP to transmit more data packets. This will not allow TCP to increase its flow which will reflect in terms of throughput.

Mobility of Nodes

Mobile community is categorized by handoffs as the user may have some movements. It will initiate packet losses and delay due to temporary disconnections. Transmission will ache a lot if it classifies each loss in congestion loss instead of network failure loss. The handoffs are probable to be more common in next generation cellular networks as the micro-cellular structure is embraced to accommodate an increasing number of users.

As these all terms are defining the performance criteria of TCP. These terms define how the TCP is giving better performance. These terms will also define the bandwidth utilization of by TCP for any connection.

Effects of all these limits on the presentation of different TCP variants is shown in this paper. On basis of this analysis one of the better performing variant will be selected as a base algorithm.

2. Literature survey

TCP WestwoodNR

This variant of TCP allows detecting multiple losses. It uses bandwidth estimation algorithm to set sending rate. It works similar to the NewReno but main advantage of WestwoodNR is it uses bandwidth estimation and always sets the ssthresh value to some higher value than in NewReno. WestwoodNR TCP differs from Westwood TCP in that it employs a new bandwidth estimation algorithm that works properly also in the presence of ACK compression [1].

So, these all flavors are used in end-to-end approach. There are some more flavors are also available but these all are related to receiver side modification and as only sender side modifications are giving higher performance we are focusing on these flavors only [2].

Demerit of TCP WestwoodNR

TCP WestwoodNR is upgraded version of Westwood so; it gives better performance than Westwood. Its algorithm is designed such that it can estimate available bandwidth accurately. And because of that it can utilize available bandwidth higher than any other flavor of TCP. Its major problem is that it takes larger time in slow start. Because of this bandwidth utilization is lower in Westwood & WestwoodNR in slow start phase.

As this congestion control algorithms tries to give better performance in terms of throughput, goodput, transmission efficiency, friendliness and fairness. Let's try to understand these all terms in brief.

Basic Congestion Control Using WestwoodNR

TCP WestwoodNR improves the window control and back off process. Namely, a TCP WestwoodNR sender displays the acknowledgment sequence it has got and from that number it assessments

the speed of connection in terms of data rate. If once, the transmitter detects a loss of packet (i.e., a timeout occurs or 3 DUPACKs are received), the transmitter tries to estimate the bandwidth to exactly set new values of the cwnd and the ssthresh. After setting new calculated values to cwnd and ssthresh, TCP WestwoodNR avoids more shrinkage of cwnd and ssthresh that can be extreme or deficient. In this way TCP WestwoodNR confirms rrecovery of the losses faster and enters in to the congestion avoidance phase as soon as possible. Experiment says the benefits of the newly established back off strategy in TCP WestwoodNR: better throughput, goodput, and delay performance, as well as fairness even when competing connections differ in their end-to-end propagation times. In addition, our studies of TCP WestwoodNR friendliness when co - existing with TCP Reno is reassuring since we have observed that TCP Reno connections are not starved in the presence of TCP WestwoodNR connections. Most importantly, TCP WestwoodNR is good in taking care of losses in the wireless networks. This is because TCP WestwoodNR uses the new calculated rate as reference for giving new values to the parameters. Ongoing data rate will be slightly wedged by loss (as long as loss is a very small fraction of available data rate) [3]. The equal share values are given as a feedback in the TCP header and informed to the TCP transmitter. The latter uses it to properly set its cwnd and ssthresh parameters. BA-TCP and TCP WestwoodNR are similar as they are relying on bandwidth information to set congestion controlling parameters. While BA-TCP needs new network layer purposes to measure available bandwidth and compute equal share, TCP WestwoodNR relies only on information willingly available in the current TCP header.

Figure.1 shows the basic algorithm working of TCP WestwoodNR. Here we can see that in the beginning only Slow Start phase is taking higher time $t_{slowstart}$ to reach ssthresh. But, after that if 3 DUPACKs are coming than BWE (Bandwidth Estimation) algorithm is coming into action and instead of setting ssthresh to lower value it is giving somewhat higher value of ssthresh. So we can understand that BE algorithm is being applied to agreed higher rate of ssthresh. Westwood is setting ssthresh value to a higher value after getting three DUPACK, so that in next phase it can reach to that point faster. By this way it will utilize available bandwidth. Here we can see that to estimate available bandwidth it is using RTT and it always uses the instant RTT. If the RTT at that instant is lower than RTT_{min} than it will replace RTT_{min} value with the present RTT value.

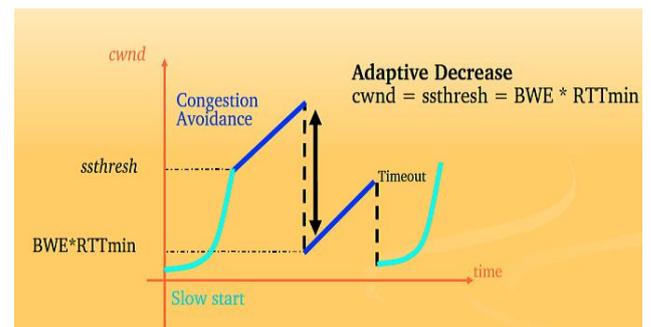


Figure 1: WestwoodNR Algorithm

3. Bandwidth Estimation (BW) in WestwoodNR

The network is taken into account a "Un predictive Bunch", giving no help to the TCP theme. The main new plan in transmission control protocol WestwoodNR is that the transmission control protocol sender ceaselessly computes the affiliation information in Bandwidth Estimate (BE) that is estimating the bottleneck information measure presently on the market to the affiliation. Thus, BE additionally represents the share of the bottleneck information measure - and so the speed presently utilized by the affiliation. The information measure estimate relies on information measure samples fetched by the supply because the ACKs area unit received. When

a packet loss in experienced may well be thanks to either congestion or link errors, the sender uses the calculable on the market information measure to properly set the congestion window and also the slow begin threshold. Packet loss symptoms utilized by a sender area unit the reception of three duplicates, or coarse timeout expiration.

3.1. Setting *cwnd* and *ssthresh* in WestwoodNR

Gives a chance to accept that a sender has decided the association data transmission evaluate BE. In this segment we portray the utilization of BE in setting *cwnd* and *ssthresh* after a parcel misfortune sign. In the first place, we take note of that in TCP Westwood, clog window progression amid moderate begin and blockage shirking are unaltered, so the increment exponentially and directly, individually, as in available TCP Reno and Newreno. A bundle misfortune is demonstrated by (1) the gathering of 3 duplicate acknowledgements, or (2) timeout lapse [4]. If the loss indication is 3 Duplicate Acknowledgements, TCP WestwoodNR keeps *cwnd* and *ssthresh* as under:

```

if (3 DUP ACKs are received)
  ssthresh = (BandEstimate * RoundTripTime min) / segment_size;
  if (cwin > ssthresh) /* congestion avoid. */
    cwin = ssthresh;
  endif
endif

```

In given code, *segment_size* gives the main length of a TCP data in terms of bits. In standard TCP RENO after receiving *n* DUPACKs in Fast Retransmit algorithm, sender has to retransmit the missing segment. This happens when time-out to illustrate, once the window is born to one and a random loss happens throughout slow begin. During this case we tend to don't reset the window however permit it to continue its exponential increase. In case a packet loss is indicated by a timeout, *cwnd* and *ssthresh* are set as follows:

```

if (timeout)
  cwin = 1;
  ssthresh = (BandwidthEstimate * RoundTripTime min) / seg_size;
  if (ssthresh < 2)
    ssthresh = 2;
  endif;
endif;

```

Here, the Reno characteristics is being still followed, while a fast recovery is also achieved by keeping *ssthresh* value with the help of Bandwidth Estimation.

3.2. Strategy for BE

The TCP Westwood sender uses Acknowledgements to decide BE. The sender uses the under said logics:

- (1) Reception of ACK in unit time interval and,
- (2) The amount of data delivered which is conveyed by the destination in the form of ACK.

We will take the information in (2) received from the receiver in upcoming portion. For now, let us consider that Acknowledgement arrived at sender with time $time_k$, indicates that $data_k$ bytes are received at the destination. With this one can amount the model bandwidth for that flow as $bw_k = data_k / (time_k - time_{k-1})$, where $time_{k-1}$ is the time the previous ACK was received [4].

Letting $\Delta time_k = time_k - time_{k-1}$, then $bw_k = data_k / \Delta time_k$

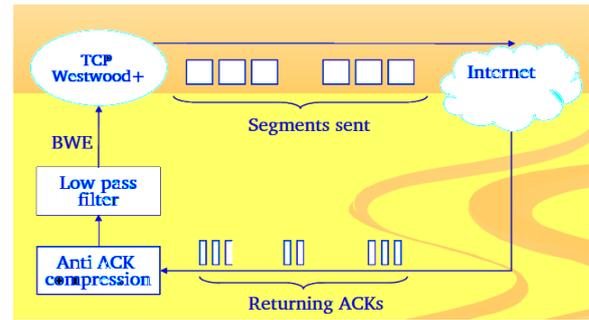


Figure 2: Bandwidth Estimation in Westwood

One can without much of a stretch check that the example data transfer capacity calculation is steady with existing transmission capacity estimation strategies. Specifically, it gives a measure of the division of bottleneck data transmission accessible to the association. In the event that the association is the only flow of the single link and the two acknowledgements being referred as to have a place with a similar window cycle, the example data transfer capacity is an exact gauge of aggregate bottleneck transmission capacity. Note that the example data transfer capacity is for the most part unique in relation to the genuine rate accomplish by the association. Here we are not discussing the part of keeping things filtering. As there are so many low pass filters are available with NS-2.30 source codes for WestwoodNR.

The Additive Increase Multiplicative Decrease algorithm is considered as a method to obtain a “coarse” and healthy quantity. The primary endeavor to abuse ACK parcels to enhance data transmission estimation is the bundle combine packet pair (PP) calculation, which tries to gather the bottleneck accessible transfer speed at the beginning of an association by estimating the bury landing time between the ACKs of two bundles that are sent consecutive [5]. Hoe gives a refined PP method for guessing the available bandwidth in order to properly reset the *ssthresh*; the bandwidth is calculated by using the least-square estimation on the reception time of three ACKs corresponding to three closely-spaced packets [6]. Allman and Paxson evaluate the PP techniques and show that in practice they perform less well than expected [7]. Lai and Baker propose an evolution of the PP algorithm for measuring the link bandwidth in FIFO-queuing networks [8]. The method consumes less network information measure whereas maintaining roughly constant accuracy of different strategies. That is less for methods longer than small networks. The quality of algorithms supported the packet try method is because of the actual detail that the repose arrival times in between successive segments at destination is totally dissimilar from the repose arrival times among the consistent ACKs at the source. It will be discussed in next portion that this result is way a lot of important within the occurrence of crowding on the reverse path. Faith and Dovrolis gives idea to use array of inquiring data to live the source-to-destination accessible information measure that is outlined because the most rate that the trail will give to a flow, while not dipping the speed of remainder of desired flow. The approximation is totaled over associate in nursing averaging pause [9]. At the end, they emphasis on the association with the available BW in a prescribed path they compute and throughput of a tireless TCP flow. Observation from this experiment is that be around throughput of a TCP flow is about 20-30% marginally more than the existing BW unrushed by experimental device. We can say, TCP is taking more share of available bandwidth initially. But, if a new connection wants to be established it will take share from the on-going flows of TCP.

WestwoodNR TCP suggests an approximation of the “best strength” BW by correctly totaling and riddling the movement of coming ACKs [11]. Example of accessible BW $bw_k = data_k / \Delta time_k$ is calculated after each Round Trip Time. The part of $data_k$ is calculated by a proper counting procedure. Bandwidth samples bw_k are low-pass filtered since congestion is due to low frequency components, and because of delayed ACK option [12]. In [13] the following

time-invariant low pass filter has been proposed as an alternative to the original time varying filter of Westwood TCP [11]:

$$bw_k = \alpha * bw_{k-1} + (1 - \alpha) * bw_k \quad (3.1)$$

Where α is a constant set equal to 0.9. The filter reveals to be particularly suited for kernel code implementation, where floating point operations should be avoided [14]. It should be taken care that bw_k are samples of used bandwidth that agree with the “best-effort” available bandwidth when the connection reaches the network capacity and start facing the victims.

After this discussion we can say that WestwoodNR is using accumulation of ACK packets over a RTT and then it tries to compress it. Here we can see the difference that there is a block called Anti ACK Compression which is used to compress the coming ACK. It always estimates available bandwidth after an RTT period. Now we can say that when any DUPACK is coming than also it is an ACK so, we have to consider it in calculation of bandwidth. Now we have to consider some points for ACK packet rates.

3.3. Processing of Acknowledgement Sequence Number

Returning acknowledgements are carrying some information regarding the delivery of the data packets. We must take care of each case in depth [4]. The sentences given under are summarizes these considerations:

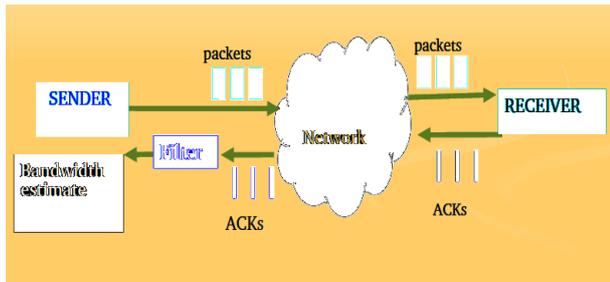


Figure 3: WestwoodNR Bandwidth Estimation

1. If an acknowledgement received by the sender, it indicates the success of reception of a packet. A DUPACK too indicates the reception of the transmitted packet, activating the send action if the received a DUPACK. So, one cannot ignore the DUPACK while estimating the channel bandwidth. A DUPACK affects the calculation of the BWE, a fresh calculation should be done after the reception of it.
2. As TCP acknowledgements may be “cumulative”, we must take care while calculating the data bytes which are delivered and only those bytes should be counted to estimate bandwidth. We should avoid the repeated counting of bytes in this.
3. TCP ACKS can be “delayed,” delayed ACKs must be carefully processed as it may give cumulative acknowledgement also.
4. One cannot predict that by which segment a DUPACK is triggered. And, it cannot have exact idea of the lost segment size. To avoid this miss-understanding, TCP is using an average segment size for that connection.

3.4. Rate Estimation

TCP WestwoodNR is providing significant better output using the estimation technique shaped by above mentioned specimen and observing process. It is observed in the scenarios where more losses are occurring. In addition, observe that in scheduled transmission if a router is using round robin policy, bandwidth estimation provides exact estimation and also provides a fair share in the network. With a router with the mechanism of dropping the tail transport layer protocol is sending whole window and waits for an acknowledgement. At present scenario, competitive processes could “take turns” in inserting their entire segments into the provided link, in this each

sender thinks that the link is solely provided to it, but as it is a bottleneck link it will over-estimate the available bandwidth. This will not allow it to have a fair share of the bandwidth. Let us take another on the market information measure sample, outlined because the quantity of information rumored to be received by the receiver with all ACKs that arrived within the last X units of time, divide it by X [4]. In further explanation after in additional part, performed activity points out towards the speed really achieved newly.

3.5. Fair share of bandwidth by TCP WestwoodNR

The main aim of TCP design is to use all bandwidth which is available by keeping fairness in allocations given to different connections [4].

We are taking the following terms to calculate the fair share of the available bandwidth that can be utilized by TCP connection:

- a) K different TCP WestwoodNR flows are available in one bottleneck link with bandwidth capacity C. The fair share is C/K.
- b) If we have total K connections of both protocols Newreno and Westwood, and if we assume that there is no random errors available in the network. Both the protocols should take equal share of the available bandwidth and provide equal output in terms of throughput. So, each protocol will have throughput equals to C/K.

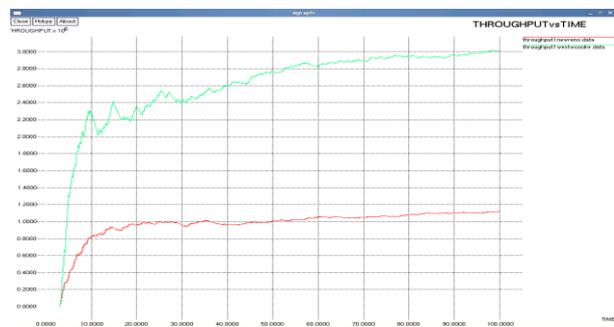


Figure 4: Fair share of bandwidth by TCP variants

- c) If CBR type of flow is already there in the network, it will take away large portion of the bandwidth. This portion is being calculated by their available transmission rates. In this case we will take only the remaining part of bandwidth and that part will be used for the fair share of WestwoodNR. This is already discussed in the above two points.

4. WestwoodNR Performance

In previous section we have discussed about the selection of algorithm. In this chapter we will try to understand the behavior of WestwoodNR in different network conditions. We will also try to compare it with the existing TCP variant which is widely accepted by the researchers that is TCP Newreno. WestwoodNR is taking help of Newreno to set ssthresh and cwnd after any congestion episode. This indicates that WestwoodNR must not degrade its performance than Newreno.

4.1. Impact of Higher ssthresh after a Timeout

Some simulations are carried out to understand the behavior of WestwoodNR in congested network. Here instead of congestion PER is used to introduce packet loss in network. Figure 3.1 shows comparison of WestwoodNR and Newreno performance on the same network. Here we can see that Newreno is always setting ssthresh value to 2 after a timeout occurs and WestwoodNR will set ssthresh to a higher value calculated with the use of ABE algorithm. cwnd is also set to a higher value according to the algorithm of WestwoodNR. This will help WestwoodNR to utilize available bandwidth.

In above figure we can observe that WestwoodNR is trying to pull cwnd up to the probed network capacity. Because of this it is trying to fill the network as soon as the link is re-established. Now as Figure 4 Comparison of Newreno and WestwoodNR in network with PER = 0.05%

If network is not out of congestion, than it will again fill the network unnecessarily and it will again cause the loss of those data. This timeout will again pull down cwnd to 2. Now, as this condition occurs it is not allowing WestwoodNR to reach to the probed capacity. This way TCP WestwoodNR is suffering with more number of timeout in the slow start phase.

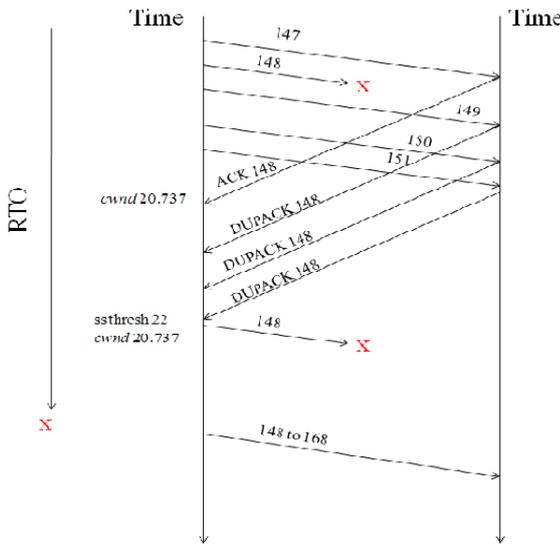


Figure 5: Comparison of Newreno and WestwoodNR

Table 1: Comparison of Newreno and WestwoodNR in Terms of Number of Timeouts

PER (%)	Timeout (SS\FR)	
	Newreno	WestwoodNR
0	0	0
0.01	1\3	1\5
0.02	1\13	4\7
0.05	6\40	14\19
0.1	21\66	49\52

In table 1 we can see that WestwoodNR is facing problem in terms of number of timeouts. As in WestwoodNR is more aggressive in nature, it is facing this kind of problem. Main result for this problem is ABE algorithm. ABE is setting higher value of ssthresh after a timeout event. Timeout is an indication of a severe congestion into the network. WestwoodNR is trying to fill the network after a timeout event to its maximum probed capacity. And, if within this time period network is not restored than it will again face the same problem of timeout.

In Figure 6, it is shown that how more number of timeouts is occurring in slow start phase. In this figure observation can be done with the help of a trace file. In this file we are taking help of

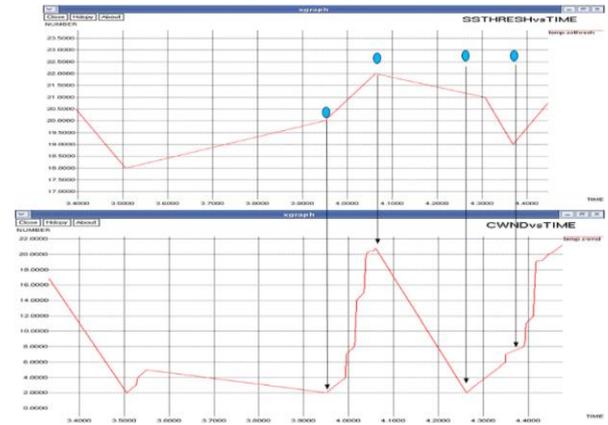


Figure 6: Reason for More Timeouts in WestwoodNR

different trace files. Suppose that TCP WestwoodNR is sending one packet on reception of ACK 147 with sequence number 167 and at that time cwnd is 20.373.

After these times suppose the packet with 148 which was sent earlier was dropped by intermediate node. Because of this reason TCP will now get DUPACK of 147. After receiving 3rd DUPACK it will enter into the fast retransmit phase and deflates the cwnd. TCP will retransmit the packet for which it got 3 DUPACK. At this time WestwoodNR is calculating ssthresh according to ABE. But if that packet is also dropped than the receiver will still send DUPACK. There is no mechanism to detect the loss of re transmitted packet in TCP. Because of this reason it will wait to RTO occur. After some time when RTO will expire TCP will reduce its cwnd to 2 and start transmissions again. Here after timeout WestwoodNR will retransmit number of packets it can send in the cwnd which was calculated when it entered into fast retransmit phase. In this case the value of cwnd was 20. As it is at higher side so it will allow WestwoodNR to pump more data in the network.

Table 2 is showing comparison of timeout in Newreno and WestwoodNR in dumbbell wired topology. As PER is increased than number of timeouts in WestwoodNR is also increased. This will pull down the cwnd of WestwoodNR below ssthresh. This indicates that WestwoodNR is now not utilizing network capacity.

Table 2: Comparison of Newreno and WestwoodNR in Dumbbell topology

PER (%)	Timeout (SS\FR)	
	Newreno	WestwoodNR
0	0	0
0.01	0\2	2\10
0.02	1\14	6\16
0.05	6\45	28\44
0.1	25\87	57\63

This way we can say WestwoodNR is more aggressive in nature as it tries to regain its flow after a timeout event. In this we can see that number of timeouts in WestwoodNR is lower. But it is having more number of timeouts in slow start phase. This is not allowing TCP WestwoodNR to increase the throughput.

Table 3: Comparison Efficiency of Newreno and WestwoodNR

PER (%)	No. on Packets Transmitted		No. of Retransmissions		Total Packet Transmitted		Transmission Efficiency	
	Newreno	WestwoodNR	Newreno	WestwoodNR	Newreno	WestwoodNR	Newreno	WestwoodNR
0	81037	81037	0	0	81037	81037	1	1
0.01	26682	59556	288	434	26970	27116	0.98	0.98
0.02	17289	45736	374	662	17663	17951	0.97	0.96
0.05	9036	22944	548	1182	9584	10218	0.94	0.88
0.1	4497	7274	560	1271	5057	5768	0.88	0.77

