## Improvement Congestion Avoidance over TCP Westwood Protocol

Mounir Frikha, Meriem Chekir Network department Ecole supérieure des communications de Tunis Route Raoued Km 3.5, 2080 ARIANA TUNISIA

*Abstract:* TCP WestwoodNR is a sender-side modification of the TCP congestion window algorithm that improves upon the performance of TCP New Reno in wired as well as wireless networks. The improvement is most significant in wireless networks with lossy links, since TCP WestwoodNR relies on end-to-end bandwidth estimation to discriminate the cause of packet loss congestion or wireless channel effect) which is a major problem in TCP New Reno.

The key idea of our improvement consists in increasing the throughput of the transmission by modifying the algorithm of congestion avoidance. The rationale of this strategy is simple: in contrast with TCP Reno, which reduce "blindly" to slow start threshold the congestion window after three duplicate ACKs, TCP WestwoodNR stabilize during a certain period the congestion window to new threshold larger then ssthresh, then it increases it linearly.

The proposed mechanism is particularly effective over wireless links where sporadic losses due to radio channel problems are often misinterpreted as a symptom of congestion by current TCP schemes and thus lead to an unnecessary window reduction.

Experimental studies reveal improvements in throughput performance, as well as in fairness. In addition, friendliness with TCP New Reno was observed in a set of experiments showing that TCP New Reno connections are not starved by TCP WestsoodNR connections.

Key-Words: Congestion Avoidance, TCP, TCP Reno, TCP New Reno, TCP Westwood, Simulation

## 1 Introduction

There is not doubt any more today: the world of telecommunications is in blooming. It opens the way to new objectives and challenges for the development of news transmission techniques but also for the setting to systems.

Today, the great majority of the networks are made up of a central infrastructure, equipped with broad band transmission capacity, and most of the time built on the basis of wired network. As this one does not allow a wide mobility to equipments to which it is connected, a news generation of networks was defined. By this one, peripherals hitherto dedicated in a fixed place, are equipped with radio operator equipment which can enter in dialogue with others while ensuring a more or less broad mobility.

This inauguration of new behaviors revealed some weaknesses in the data transmissions following up the heterogeneity of the propagation environment. The wired networks initially conceived for end to end communications forwarded exclusively by cabled way are put today

at contribution for transit between and to the wireless networks. These modifications of behavior are not without consequences for the majority of the algorithms designed for control of information. For example a data loss could be interpreted in various manners according to whether it comes from a saturation of a link in the network or a loss caused by a little reliable propagation environment, or as a consequence of a technical deficiency which has occurred in equipment carrying out transmission. Algorithms developed until now do not make distinction between these different the interpretations.

We can observe that if no modification is made, they will contribute to make the performances of transmission rather disastrous on heterogeneous networks. In fact, information transport protocols as previously conceived and which are based on the packets losses to adjust their parameters are less powerful than the algorithms in which our estimation is based on the bandwidth available for adjusting these parameters, as well. Random losses of packets suitable for wireless network an ill-treated. We are interested in the problem of evolution of the congestion window during the detection of a loss.

We present in section 2 an outline of the evolution of the congestion window in TCP Westwood. Our contribution will be detailed in section 3. After that, in section 4, we will simulate our proposal in the wired and wireless networks. In section 5, we explain the choice of our new parameter. Then in section 6, we verify if our contribution does not affect the other properties of TCP Westwood. And we conclude by the perspectives which can open our work.

## 2 An overview of TCP Westwood

In TCP Westwood the sender continuously computes the connection BandWidth Estimate (BWE) which is defined as the share of bottleneck bandwidth used by the connection. Thus, BWE is equal to the rate at which data is delivered to the TCP receiver. The estimate is based on the rate at which ACKs are received and on their payload. After a packet loss indication, (ie, reception of 3 duplicate ACKs, or timeout expiration). The sender resets the congestion window and the slow start threshold based on BWE. More precisely, cwnd = BWE x RTT.

To understand the rationale of TCPW, note that BWE varies from flow to flow sharing the same bottleneck; it corresponds to the rate actually achieved by each INDIVIDUAL flow. Thus, it is a FEASIBLE (ie, achievable) rate by definition. Consequently, the collection of all the BWE rates, as estimated by the connections sharing the same bottleneck, is a FEASIBLE set.

When the bottleneck becomes saturated and packets are dropped, TCPW selects a set of congestion windows that correspond exactly to the measured BWE rates and thus

reproduce the current individual throughputs. The solution is feasible, but it is not guaranteed per se to be fair share. An important element of this procedure is the RTT estimation. RTT is required to compute the window that supports the estimated rate BWE. Ideally, the

RTT should be measured when the bottleneck is empty. In practice, it is set equal to the overall minimum round trip delay (RTTmin) measured so far on that connection (based on continuous monitoring of ACK RTTs).

We note that in TCPW, congestion window increments during slow start and congestion avoidance remain the same as in Reno, that is they are exponential and linear,

respectively. A packet loss is indicated by (a) the reception of 3 DUPACKs, or (b) a coarse timeout expiration. In case the loss indication is 3 DUPACKs, TCPW sets

cwnd and ssthresh as follows:

if (3 DUPACKs are received)

ssthresh = (BWE \* RTTmin) / seg\_size;

if (cwnd > ssthresh) /\* congestion avoid. \*/

cwnd = ssthresh;

endif

endif

In the pseudo-code, seg size identifies the length of a TCP segment in bits. Note that the reception of n DUPACKs is followed by the retransmission of the missing segment, as in the standard Fast Retransmit implemented by TCP Reno. Also, the window growth after the cwnd is reset to ssthresh follows the rules established in the Fast Retransmit algorithm (i.e., cwnd grows by one for each further ACK, and is reset to ssthresh after the first ACK acknowledging new data). During the congestion avoidance phase we are probing for extra available bandwidth. Therefore, when n DUPACKs are received, it means that we have hit the network capacity (or that, in the case of wireless links, one or more segments were dropped due to sporadic losses). Thus, the slow start threshold is set equal to the window capable of producing the measured rate BWE when the bottleneck buffer is empty (namely, BWE\*RTTmin). The congestion window is set equal to the ssthresh and the congestion avoidance phase is entered again to gently probe for new available bandwidth.. Note that after ssthresh has been set, the congestion window is set equal to the slow start threshold only if cwnd > ssthresh. It is possible that the current cwnd may be below threshold. This occurs after time-out for example, when the window is dropped to 1 as discussed in the following section. During slow start, cwnd still features an exponential increase as in the current implementation of TCP Reno. In case a packet loss is indicated by a timeout expiration, cwnd and ssthresh are set as follows:

if (coarse timeout expires) cwnd = 1;

```
ssthresh=(BWE.RTTmin) /seg_size;
if (ssthresh < 2)
sthresh = 2;
endif;
endif
```

The rationale of the algorithm above is that after a timeout, cwnd and the ssthresh are set equal to 1 and BWE, respectively. Thus, the basic Reno behavior is still captured, while a speedy recovery is ensured by setting ssthresh to the value of BWE.

# 3 Modifying congestion avoidance algorithm

We propose in this part how to improve the throughput without increasing the number of packets lost.

On the basis of the observation, we do not note a fluctuation of the curve of bandwidth used only when there is a loss of packet. In fact, the packet loss is sign of congestion, it is also the cause of the release of the congestion avoidance phase which decreases considerably the size congestion window.

We estimate that the reduction of the congestion window cwnd is still very significant; we propose to reduce this variation. During a congestion detection, we propose to not reduce cwnd to ssthresh but to divide this reduction interval by two. Instead of beginning the window incrementing from this new threshold, we stabilize it with this threshold to form a stage. When the stage coincides with the initial curve we swing towards a linear increase.

Thus, we preserve the same shape of the curve and where there is a fall of the window size, we replace the fall by the stage.

Now, we propose to calculate the period during which the window cwnd will remain constant. This period is not fixed but varies according to the maximum size of cwnd during the congestion detection. In fact, the larger cwnd, the longer the stage. Thus, to estimate the length of the stage, it is necessary to calculate the time that

the congestion window takes to be incremented during the part of the curve starting from threshold ssthresh to reach the new Newssthesh threshold, which is the stage value. The incrementing is carried out from the incrementing function described in the algorithm of congestion avoidance of TCP Westwood. Our proposal is described in this pseudo code: Newssthresh = ssthresh + (cwnd - ssthresh)/2 (1) New increment = ssthresh (2) if (cwnd > ssthresh) then when ack receved : New increment = New increment +1/New increment (3) If (New increment < Newssthresh) then cwnd = Newssthresh (4) else cwnd = New increment (5)

The New increment variable is used as meter in the stage phase to calculate its duration. This variable is even useful to us as an increment in the linear increase phase in the curve.

## 4 Simulation

To make sure that our theoretical idea is verified by simulations, we chose to treat different cases of networks. In this part, we will see what happens in the case of a wired network resenting bottleneck and in the case of a wireless network presenting random packets loss.

We will indicate by TCP WestwoodNR+ the protocol TCP Westwood applied to New Reno integral our contribution.



Figure 1 — Simple topology with bottleneck

### 4.1 Wired Network case

The figure 1 shows a network topology that we simulate with 2 portions including a 100-Mbps link intercalated by a bottleneck with a bandwidth of 5 Mbps and delay a 35 ms.

To visualize the modification that we carried out, we chose to represent on the same graph the evolution of the congestion window and its threshold ssthresh which enables us to clearly distinguish the modification.

The figure 2 represents, in red, the new pace of evolution in time of the congestion window. We clearly observe the existence stages.



Graphically, we can conclude that the fixed objectives are carried out since the amplitude of the cwnd size decreased and we also observe that there is no increase in the packets loss. Following this curve, we can deduce the bandwidth utilisation ratio.

The figure 3 shows us a graph with two phases. Even if the use of the bandwidth fluctuates a little more at the beginning of simulation, the second phase is much more stable. The throughput calculation proves this tendency since we have under the same conditions an average throughput of 4.538 Mbits/s whereas it was 4.436 Mbits/s before our contribution, which gives us an improvement of the throughput of 2.30%.

We conclude that our contribution presents a throughput improvement and verifies our starting conditions. But, the improvement is not very significant. Let us observe now what happens in the wireless network.



Figure 3 — Use of the bandwidth in TCP WestwoodNR+

#### 4.2 Wireless Network case

We integrated our contribution in TCP WestwoodNR and we applied it in the case of a network presenting packets loss.



Figure 4 — Evolution of cwnd in the wireless network

That is what we obtain: a curve 4 presenting stages intercalated by linear incrementing phases. We notice that the stages don't have the same length, this is explained by the random aspect of the packets loss which leads to a variable size of the congestion window at the decrementation time. Although the curve of cwnd presents clearly this random aspect, it does not affect the threshold ssthresh calculation. We notice that it very little varies since the 7th second of simulation. In fact, our contribution preserves the characteristics of TCP Westwood.

Thus, we obtain a graph 5 of the bandwidth use which does not vary with each packet loss. It is also another characteristic of TCP Westwood that we find. The curve evolves exponentially to reach and stabilize very close to physical limit of the bottleneck fixed to 5 Mbit/s.

Our contribution presents an improvement in the window evolution cwnd and in the use of the bandwidth. The average throughput of simulation is of 4.913 Mbit/s where as it was 4.510 Mbit/s given an improvement of 8.94% (see figure 6).

This improvement is considerable since our theoretical work stroke is of a maximum improvement of 10.86%. In fact, we will not be able to have an improvement





Figure 6 — Throughput befor and after the modification

only when the throughput found is included between the throughput of reference before our contribution which is 4.510 and the physical bandwidth of the bottleneck which is 5 Mbit/s.

#### 5 Choice of the new threshold

The choice of the Newsstresh threshold of equation 1 is the result of several simulations. Our principal objective is to place this stage at a value between ssthresh and cwnd.

- a value of Newsstresh too close to cwnd and we fall into the case where we are still in congestion.

- a value very close to ssthresh and our contribution will not be significant.

To know on which level the stage should be placed, we have made several simulations by dividing the interval between ssthresh and cwnd on several values. The general form of the equation is:

Newssthresh = ssthresh+(cwnd - ssthresh)/cte (6)

According to the values of cte we find different throughputs. We preserve the constant which gives us the best throughput. but we have noted according to simulations that the best throughput in without loss network intervenes for different cte value in case of a network presenting the random packets losses.

- the best cte value for the wired network is 3.6 to have a throughput of 4.718 Mbit/s.

- the best cte value for the wireless network is 2 to have a throughput of 4.913 Mbit/s.

Following this divergence, we choose to privilege wireless case since the contribution is better there. We find therefore equation 1.

## 6 Evaluation of performances : TCP Friendliness of WestwoodNR+

We showed previously that our contribution has well increased the throughput. It should be seen now if this improvement does not deteriorate the notion of the friendship. We will simulate in this part our improved protocol to see whether it is compatible with the other existing protocols. We will simulate the scenario described in figure 7.



Figure 7 — TCP freindliness

#### 6.1 with TCP New Reno

We test compatibility between two sources S1 and S2 emitting simultaneously towards the same nodes. The source S1 emits according to our protocol TCP WestwoodNR+, its relative curves are schematized in red in the following graphs.

The S2 source emits according to protocol TCP New Reno, his relative curves are schematized in green. Thus, we choose to start by comparing with the protocol New Reno since it is the base of our protocol.

The figure 8 represents the number of received acknowledgements by the sources. These curves give us an idea on rate of each source. We notice that the S1 source rate emitting with TCP WestwoodNR+ is more significant than that of S2 which is emitted with TCP New Reno without crushing it.

The figure 9 confirms this tendency since we distinguish that cwnd from TCP WestwoodNR+ is above that of TCP New Reno. We can also say that protocol TCP New Reno does not die of starvation since its window of congestion periodically becomes extensive.





The calculation of the throughput of our modified protocol, under the same conditions of simulation, indicates that it is improved by 44.29% compared to TCP New Reno.

#### 6.2 With TCP WetwoodNR

The curves in red relate to TCP WestwoodNR+ and the curves in green relate to the reference TCP WestwoodNR protocol. We can observe now the compatibility of our improvement with the same protocol without our contribution.

The figure 10 shows that the two WestwooodNR protocols with and without modification are similar. The slight superiority of the red curve indicates the presence of our contribution.



WestwoodNR

In the figure 11, and from the 10th second, we can see that the congestion windows of both curves are practically superimposed. Thus, we can conclude that the improvement of throughput is due to the presence of the stage in our modification.



Figure II — Cwnd of TCPWestwoodNR with and without modification

#### 7 Conclusion

The study of protocol TCP Westwood enabled us to rase the problem of congestion window evolution. To improve the transfer of data, it is necessary to resolve this problem. We proposed in this paper an hypothesis to reduce the margin in which can evolve the congestion window and modify hallure of the curve of the window evolution according to time. Our contributions made it possible to increase the throughput without increasing the packetloss rate. After that, we cheked whether this modification affected the other aspects of protocol and more precisely the aspect of the friendliness which is one of the strong points of protocol TCP Westwood. In conclusion, we can confirm that our contribution proves to be an amelioration.

For a later work, we project to make an analytical model of our work and to adapt it to the ad hoc networks.

References:

[1] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links", In Proceedings of Mobicom 2001, Rome, Italy, Jul. 2001.

[2] M. Gerla, M. Y. Sanadidi, R. Wang, A. Zanella, C. Casetti, S. Mascolo, "Westwood: Congestion Window Control Using Bandwidth Estimation", In Proceedings of IEEE Globecom 2001, Volume: 3, pp 1698-1702, San Antonio, Texas, USA, November 25-29, 2001.

[3] G. Yang, R. Wang, M. Y. Sanadidi, M. Gerla, "TCPW with Bulk Repeat in Next Generation Wireless and Satellite Networks", ICC 2003, Alaska, May 2003.

[4] I Lengliz, H. Touati, F. Kammoun, M. Y. Sanadidi "Experimentation towards TCP Westwood application in ad-hoc mobile networks" in Adhoc Mobile Networks Med-Hoc Net 2003, Tunis, Tunisia, June 2003.

[5] C. Casetti, M. Gerla, S. S. Lee, S. Mascolo and M. Sanadidi, "TCP with Faster Recovery", In Proceedings of MILCOM 2000, Los Angeles, CA, October 2000.

[6] S. Floyd, "The NewReno Modification to TCP's Fast Recovery Algorithm", RFC 3782, INTERNET DRAFT, February 1999.

[7] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", RFC2001, janvier 1997.

[8] M. Chekir, M. Frikha, "Etude et Amélioration du Contrôle de Congestion dans IP", Maser in telecommunications, Sup'Com Tunisia, May 2005.