# Analyzing Effect of Loss Differentiation Algorithms on Improving TCP Performance

Mi-Young Park, Sang-Hwa Chung*

*Department of Computer Engineering, Pusan National University,*
*Busan, South Korea*
*openmp@korea.com*, shchung@pnu.ac.kr

***Abstract*—When a TCP sender receives three successive duplicate ACKs indicating a packet loss, loss differentiation algorithms (LDAs) distinguish wireless losses from congestion losses to improve TCP's performance in wireless networks. Although previous LDAs improved the accuracy of wireless loss discrimination, each LDA has a trade-off relationship between the accuracy of wireless loss discrimination and the accuracy of congestion loss discrimination based on its own threshold. To suggest good guidelines for deciding the best threshold to improve TCP's performance to the max, we observe the relationship between LDA's accuracy and TCP's performance improvement throughout the extensive simulations. Contrary to our expectations, the simulation results show that LDA's detection information itself is not sufficient to improve TCP's performance due to frequent spurious retransmission timeouts (RTO). The results emphasize that such spurious timeouts should be detected and the RTO recovery should be modified to utilize the detection information of a LDA to improve TCP's performance in wireless networks.**

***Keywords*— TCP, retransmission timeouts, wireless networks, loss differentiation algorithm**

## I. INTRODUCTION

When TCP [5] operates in wireless networks, it suffers from severe performance degradation because of the different characteristics of wireless networks and wired networks [6], [12]. The performance degradation is mainly caused by TCP's basic assumption that any packet loss is an indication of congestion. Although this assumption works very well in wired networks where most packets are lost due to congestion, the assumption is not appropriate for wireless networks where most packet losses are caused by wireless transmission errors [3].

The appropriate behavior of TCP for the packet loss due to wireless transmission errors is just to retransmit the lost packet without reducing its sending rate. Unfortunately, TCP considers every packet loss as congestion signals, and unnecessarily decreases its sending rate by halving its congestion window size. To avoid such performance degradation, it is important for TCP to differentiate between wireless losses and congestion losses.

For this reason, several loss differentiation algorithms (LDAs) [1]–[4], [7]–[9], [11] have been proposed to improve TCP performance by distinguishing wireless losses from congestion losses in wireless networks. In our previous work, we also suggested an end-to-end loss differentiation algorithm (LDA) which has the highest accuracy among the previous LDAs. One of our simulation results in the previous work showed that each LDA has a trade-off relationship between the accuracy of wireless loss discrimination ($A_w$) and the accuracy of congestion loss discrimination ($A_c$). For example, if $A_c$ is high in a LDA, its $A_w$ is low, or vice-versa. In addition, the trade-off relationship between $A_c$ and $A_w$ changed according to a threshold of a LDA.

This paper is motivated from the observation in our previous work [8]. First of all, we aim to suggest good guidelines for deciding the best threshold for all LDAs by investigating the trade-off relationship between $A_c$ and $A_w$ as well as by inspecting the relationship between LDA's accuracy and TCP's performance improvement. Additionally, we aim to measure the performance improvement when our LDA is applied in TCP, and show that its improvement is the highest among the previous LDAs as its accuracy is the highest.

For our aims, we design more than 200 different simulation scenarios by setting different values for network parameters using QualNet [15] and group all scenarios into three small groups: a group with packet losses caused by only wireless transmission errors (*W* group), a group with packet losses caused by only congestion (*C* group), and the last group which is mixed with the two types of packet losses (*M* group). Before we simulate all scenarios, we modify the fast recovery algorithm [13] of TCP to utilize the detection information of a LDA, and then we measure and compare both TCP's performances with and without a LDA.

The simulation results are quite different from what we expected. The results show that spurious retransmission timeouts (RTO) due to wireless losses prevent TCP from increasing its congestion window size, and make little performance improvement even though TCP avoids

---

*Corresponding author. This work was supported by the Grant of the Korean Ministry of Education, Science and Technology (The Regional Core Research Program/Institute of Logistics Information Technology)

considerable spurious fast retransmits with the help of LDA. The observation in our simulation shows that 1. LDA's detection information itself is not sufficient to improve TCP's performance, and 2. the performance degradation due to wireless losses could not be avoidable without detecting spurious retransmission timeouts and modifying the RTO recovery algorithm of TCP based on the detection information of a LDA.

In the following section, we introduce previous LDAs including our scheme, and then, in Section 3, we explain our motivation. In Section 4 we investigate the relationship between LDA's accuracy and TCP's performance improvement throughout the extensive simulations. Lastly we summarize and discuss our simulation results and conclude this paper with our future work.

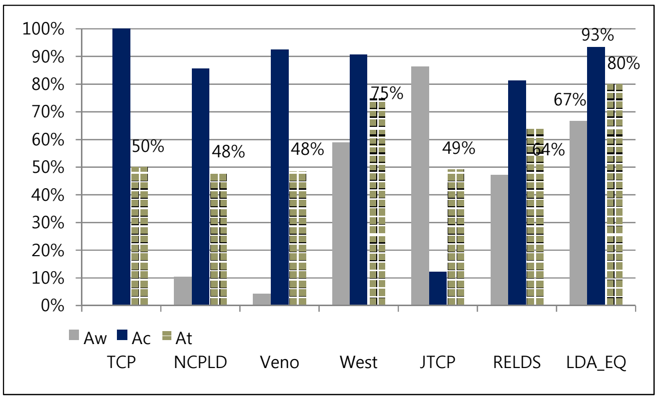


**Figure 1. Comparisons of accuracy of existing LDAs**

## II. EXISTING LOSS DIFFERENTIATION ALGORITHMS

Several solutions for distinguishing the cause of packet losses have been proposed to improve TCP [5] performance in wireless networks. These solutions can be broadly classified in two classes: those that require support from the intermediate network nodes, and those that work purely on an end-to-end basis which retains TCP semantics. Since it is difficult to deploy the solutions which require support from the network, end-to-end solutions are more desirable. Here, we introduce six end-to-end loss differentiation algorithms (LDAs): NCPLD [9], Veno [1], West [4], JTCP [3], RELDS [2], LDA_EQ [8].

Samaraweera [9] proposed an non-congestion packet loss detection (NCPLD) to implicitly detect the type of packet loss using the variation of delay experienced by TCP packets. On detection of a packet loss, the scheme compares the currently measured round trip time (*RTT*) with a calculated delay threshold. If the RTT is less than the threshold, the scheme treats the packet loss as wireless losses. Otherwise, it treats the packet loss as congestion losses.

TCP Veno [1] estimates the backlog packets ($N$) in the buffer using Vegas's mechanism [7]. When a packet is lost, Veno compares $N$ with a threshold 3. If $N < 3$, Veno ascribes the packet loss to wireless transmission errors; otherwise, it assumes the loss as congestion losses.

Yang [4] adopted Spike [11] scheme suggested by Cen and Voelker as its loss differentiation scheme. While Spike scheme uses the relative one-way trip time (*ROTT*) taken by a packet to travel from the sender to the receiver, Yang (West) uses RTT instead of ROTT at the sender side. Based on *RTT*, it computes the two thresholds, $B_{spikestart}$ and $B_{spikeend}$, to identify the spike state of the current connection. Any packet losses in the spike state are considered as congestion losses.

Wu and Chen [3] proposed a jitter-based TCP (JTCP) to adapt sending rates to the packet losses and jitter ratios. To distinguish congestion losses from wireless losses, JTCP calculates a threshold (*Jr)* which is the average of the inter arrival jitter during one round-trip time. When a TCP sender receives three duplicate ACKs, it checks if the time receiving the three duplicate ACKs exceeds one *RTT* as well as if *Jr* is larger than the inverse value of the current congestion window size. If the two conditions are satisfied, it ascribes the packet loss to congestion; otherwise, it is assumed as wireless losses.

Lim and Jang [2] suggested a robust end-to-end loss differentiation scheme (RELDS) to precisely discriminate between congestion losses and wireless losses. This scheme employs a moving threshold which is defined as a function of minimum and sample *RTT*. If the moving threshold is satisfied when a TCP sender receives the third duplicate ACK, it assumes the packet loss as congestion losses; otherwise it assumes the packet loss as wireless losses.

In our previous work [8], we suggested an end-to-end loss differentiation scheme (LDA_EQ) which estimates the rate of queue usage using information available to TCP. If the estimated queue usage is larger than a certain threshold when a packet is lost, our scheme diagnoses the packet loss as congestion losses. Otherwise, it assumes the packet loss as wireless losses.

As described above, each LDA has its own threshold to classify congestion losses and wireless losses. Based on the threshold, the trade-off relationship between $A_c$ and $A_w$ changes; For example, if a threshold has a low value, $A_c$ increases while $A_w$ decreases or if a threshold has a high value, $A_c$ decreases while $A_w$ increases. When we developed our own scheme, we also had to spend much time on deciding the appropriate value for our threshold. Whenever we changed the value of the threshold, the trade-off relationship between $A_c$ and $A_w$ changed. Thus, by investigating the trade-off relationship between $A_c$ and $A_w$, in this paper, we aim to suggest good guidelines for deciding the best threshold for all LDAs, and show the relationship between LDA's accuracy and TCP's performance improvement.

## III. MOTIVATION

The accuracy in distinguishing congestion losses and wireless losses is the main appraisal standard of loss differentiation algorithms (LDAs), because it is assumed that as the accuracy is higher TCP's performance can be improved much more. Thus, in our previous work, we suggested an end-to-end loss differentiation algorithm which has the highest accuracy among the previous LDAs.

Figure 1 shows one of our simulation results in our previous work. As shown in the figure, $A_w$ of our scheme is 67%, and its $A_c$ is 93%, so its average accuracy ($A_t$) is 80% which is the highest among the LDAs. In case of TCP which assumes all packet losses as congestion losses, its $A_c$ and $A_w$ is 100%, 0% respectively, and its $A_t$ is 50%. In this graph, we also found that each LDA has a trade-off relationship between the accuracy of wireless loss discrimination ($A_w$) and the accuracy of congestion loss discrimination ($A_c$). For example, if $A_c$ of a scheme is high, its $A_w$ is low like NCPLD, and Veno. On the other hand, if $A_w$ of a scheme is high, its $A_c$ is low like JTCP. Such trade-off relationship is unavoidable due to the misclassification and limited information at Transport layer.

This paper is motivated from the observation in our previous work. Although $A_t$ of our scheme is the highest, its $A_w$ is not highest, and our scheme sacrifices $A_c$ compared to $A_c$ of TCP. Thus, first of all, we wonder if our scheme really achieves the highest performance improvement of TCP with its highest $A_t$.

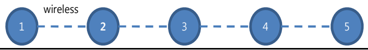
Secondly, we wonder which scheme among TCP, NCPLD, Veno and JTCP achieves better performance of TCP. As shown in the figure, Each $A_t$ of TCP, NCPLD, Veno and JTCP is around 50%, but their trade-off relationship between $A_w$ and $A_c$ is quite different each other. For example, in case of TCP, its $A_w$ is 0% and its $A_c$ is 100%. In case of NCPLD and Veno, their $A_w$ are about 10% and their $A_c$ are around 90%. In case of JTCP, its $A_w$ is about 80% and its $A_c$ is less than 20%.

Thus, we wonder if the TCP' performances achieved by each of NCPLD, Veno, and JTCP are almost same because their $A_t$ is almost same, or if the performances are quite different each other because the trade-off relationships between $A_w$ and $A_c$ are different. By comparing the performance improvements achieved by each of NCPLD, Veno, and JTCP, we can know if the average accuracy ($A_t$) is critical or if one of $A_w$ and $A_c$ is more critical to improve the performance of TCP.

Lastly, we wonder how serious the performance degradation of TCP is due to the sacrificed $A_c$. Although all LDAs improved $A_w$, these could not avoid sacrificing $A_c$ because of misclassifying congestion losses as wireless losses. By measuring the performance degradation of TCP due to the reduced $A_c$, we can know how much $A_c$ can be sacrificed to improve $A_w$ achieving the best improvement of TCP.

Throughout the extensive simulations in this paper, we plan to find all answers about the questions came from the figure 1, and then we intend to suggest good guidelines for deciding the best threshold for all LDAs by investigating the trade-off relationship between $A_c$ and $A_w$ as well as by inspecting the relationship between LDA's accuracy and TCP's performance improvement.

**Table 1. Simulation Parameters**

| | |
|---|---|
| Simulator | QualNet 4.5 |
| Topology | 5-hop wireless chain topology<br><br> |
| Bandwidth | 2MB |
| Application | FTP/Generic |
| Transport Protocol | TCP Reno |
| Queuing Policy | DropTail |
| Link Protocol | IEEE 802.11b |
| Wireless Error Model | Deterministic, Uniform, Exponential |
| Maximum Segment Size | 1024Bytes |
| Packet Size | 1024Bytes |
| Simulation Time | 200s (including 35s warm-up) |

## IV. COMPARATIVE STUDY ON RELATIONSHIP BETWEEN ACCURACY AND PERFORMANCE

### *A. Simulation methodology*

To find right answers for our questions, we have designed about 255 different scenarios by setting different values for network parameters such as the queue size, the number of hops, and the loss rate. Table 1 shows the common values used in all scenarios. Throughout the extensive simulations, we aim 1. to measure the performance improvement of TCP due to the improved $A_w$, 2. to measure the performance degradation of TCP due to the reduced $A_c$, 3. to check which accuracy among $A_t$, $A_c$, and $A_w$ is critical to improve TCP's performance, 4. to check that our scheme achieves the highest performance enhancement of TCP.

For this, we grouped all scenarios into three groups: a group with packet losses caused by only wireless transmission errors (*W* group), a group with packet losses caused by only congestion (*C* group), and the last group which is mixed with the two types of packet losses (*M* group).

*W* group is designed to observe the performance improvement of TCP due to the improved $A_w$ in each LDA (TCP's $A_w$ is 0%). Thus, all packet losses in this group are caused by only wireless transmission errors, and we measured $A_w$ and TCP's performance achieved by each LDA according to the rate of packet losses, and the number of hops. The rate of packet losses ranges from 1% to 6%, and we used three different error models for each packet loss rate: deterministic, uniform, and exponential. Each scenario in *W* group has different value in terms of *loss rate*, *hop count*, and *wireless error model.* Thus, *W* group consists of 80 different scenarios by combining the three factors differently.

In a similar way, *C* group is planned to observe the performance degradation of TCP due to the sacrificed $A_c$ (TCP's $A_c$ is 100%). Thus, all packet losses in this group are caused by only congestion, and we measured $A_c$ and TCP's performance achieved by each LDA according to the rate of packet losses, the number of hops, and the queue size. To make different levels of congestion, we increased the number of TCP flows gradually, and the rate of packet losses due to congestion ranges from 1% to almost 15%. Each scenario in *C* group has different value in terms of *the number of TCP flows*, *hop count*, and *queue size* (20KB - 60KB). Thus, *C* group consists of 150 different scenarios.

*M* group is designed to observe and measure the improved performance of TCP achieved by each LDA under a more realistic network environment. For this, we mixed the two types of packet losses (wireless losses, congestion losses). The rate of packet losses in each scenario ranges from 4% to 8%, and the ratio of wireless losses to congestion losses is approximately 5:5, 2:8, or 8:2 under different network parameters. Each scenario in *M* group has different value in terms of *wireless error model, TCP flows, hop count*, and *queue size*. Thus, *M* group consists of 25 different scenarios.

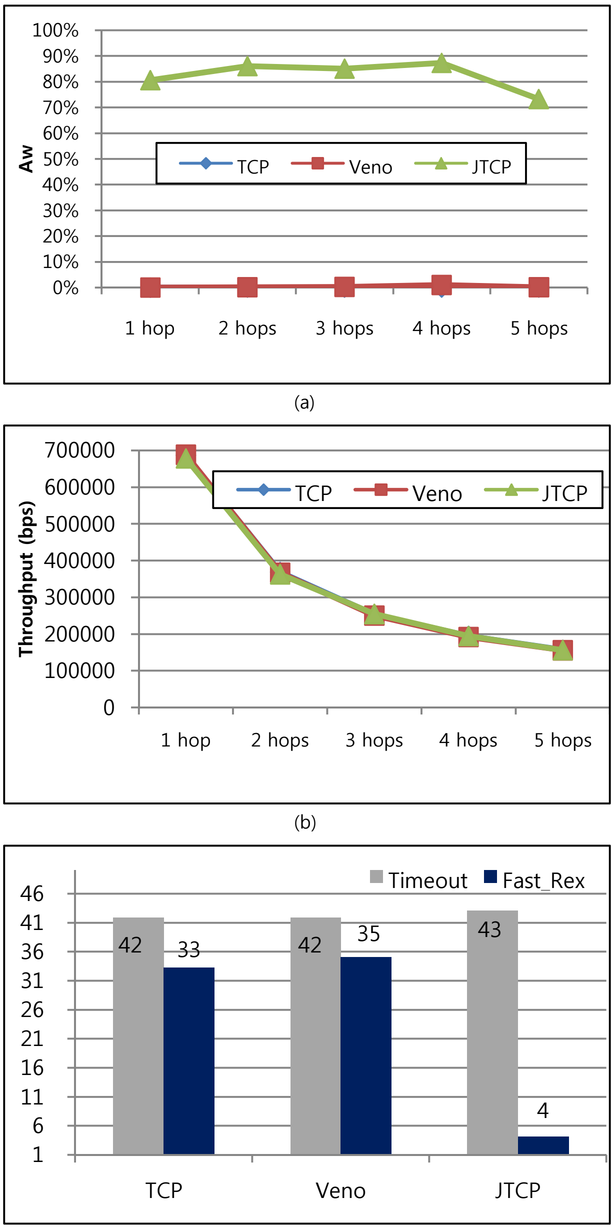
### B. Simulation Results

**1) Measuring the performance improvement due to the improved $A_w$:** In *W* group, we measured $A_w$s of Veno and JTCP, and compared the performance improvements achieved by these; Veno has the lowest $A_w$ and JTCP has the highest $A_w$ among the LDAs. This is to know the relationship between $A_w$ and TCP's performance improvement, and to measure how much TCP's performance can be improved according to $A_w$.

Figure 2(a) shows the average accuracy of TCP, Veno and JTCP according to the number of hops, and Figure 2(b) shows the corresponding TCP's performance when Veno or JTCP is applied to TCP. Veno's $A_w$ ranges from 0% to 4.5% while JTCP's $A_w$ ranges from 65% to 95%. If we compare Figure 2(a) and Figure 2(b), we can find that there is little difference between TCP's performance improvements achieved by Veno and JTCP although there is a big difference between accuracies of Veno and JTCP. In case of Veno it improved TCP's performance by 1% at the best case and by -4% at the worst case. In case of JTCP, TCP's performance is improved by -1% at the best case and by -5% at the worst case.

This simulation result is quite different from what we expected. To find the reason that there is little performance improvement of TCP with the highest $A_w$, we checked trace files of all scenarios, and found that there are not only many spurious fast retransmits but also many spurious retransmission timeouts (RTO). All fast retransmits and timeouts in *W* group are spurious because *W* group is designed to have only one TCP flow in order to avoid causing congestion. Thus, those spurious fast retransmits and retransmission timeouts are caused by wireless losses.

Figure 2(c) shows the average number of fast retransmits and retransmission timeouts in each of TCP, Veno and JTCP. For example, when Veno is applied to TCP in *W* group, the average number of fast retransmits and that of timeouts are respectively 35 and 42. In case of JTCP, the number of fast retransmits and that of timeouts is 4 and 43 respectively. If we compare the number of fast retransmits of TCP with that of those of JTCP, we can see that the number of fast retransmits of JTCP is significantly reduced while that of fast retransmits of Veno is similar with that of TCP. It means that the number of spurious fast retransmits decreases as $A_w$ increases. On the other hand, the number of spurious retransmission timeouts is not reduced at Veno and JTCP. Such frequent spurious timeouts tend to suddenly and significantly reduce congestion window size to one segment. From this observation, we can understand why TCP's performance is little improved even with the highest $A_w$. Due to the frequent retransmission timeouts, there is no chance for TCP to increase its congestion window size even though TCP avoids spurious fast retransmits with the help of a LDA.



(a)

(b)

(c)

**Figure 2. Simulation results in *W* group**

***2) Measuring the performance degradation due to the sacrificed $A_c$:*** While TCP's $A_c$ is 100%, all LDAs could not avoid sacrificing $A_c$ to improve $A_w$. Thus, we planned to observe how TCP's performance degrades due to the sacrificed $A_c$ in *C* group. For this, we measured the misclassification ($M_c$= 100% - $A_c$) of JTCP since JTCP's $A_c$ is the lowest among previous LDAs. Figure 3(a) shows the average misclassification of JTCP according to the number of hops in *C* group. In all scenarios, its misclassification ranges from 73% to 100%. It means that it misclassifies more than 7 congestion packet losses as wireless losses among 10 congestion losses. Due to the misclassification, JTCP does not reduce its sending rate even though those packets are lost due to congestion. We expected that such inappropriate responding of JTCP might cause more serious congestion.

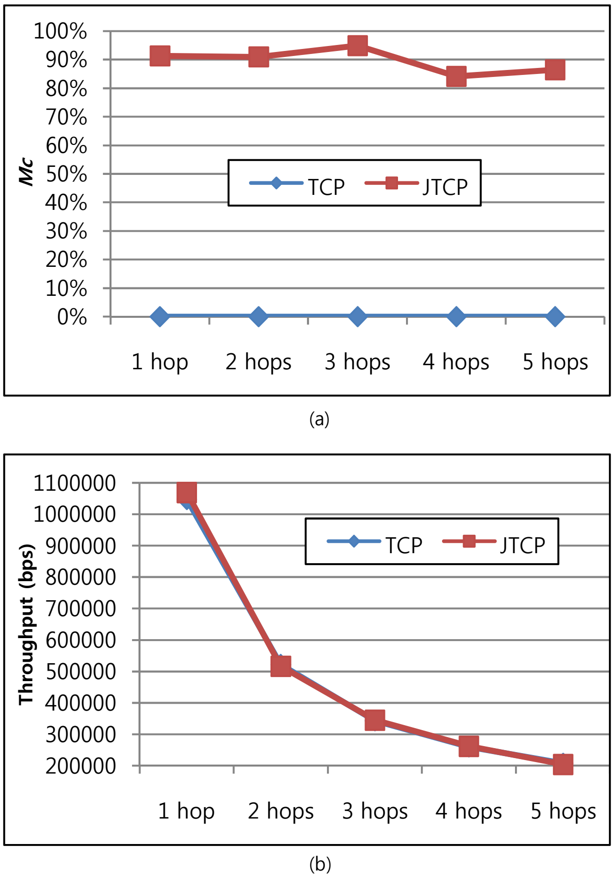
Figure 3(b) shows the performances of TCP and JTCP corresponding to Figure 3(a). Although there is a big difference between the misclassifications of TCP (0%) and JTCP (> 70%), there is little difference between the performances of these. The performance enhancement of TCP when JTCP is applied ranges -16% to 12%. In some cases the performance is improved while it is reduced in other cases, and the average improvement in *C* group is 0%.

Contrary to our expectations, there is little performance degradation due to the reduced $A_c$. To investigate why, we also checked the number of fast retransmits and retransmission timeouts in *C* group. We found that there are several hundred of retransmission timeouts in each scenario. Due to the frequent timeouts, there is little chance for the congestion window to increase, and with a small congestion window size there is no chance to cause serious congestion by the sacrificed $A_c$. This is the main reason why the TCP's performance did not degrade much with the reduced $A_c$.

Although all scenarios in *C* group are designed to have no wireless losses, there are also spurious retransmission timeouts. Those timeouts happen due to delay spikes which are defined as a sudden and significant change in the round-trip time between a TCP sender and its receiver. Such high delay variability has been observed in fixed wired networks and can be caused by route flipping [14]. Thus, without removing spurious retransmission timeouts in *C* group, it is not easy to exactly observe the relationship between the sacrificed $A_c$ and TCP's performance degradation.

***3) Comparison of TCP's Performances achieved by LDAs:*** Figure 4 shows the lowest, highest, and average of the performance improvement when each LDA is applied to TCP in *M* group. For example, when Veno is applied to TCP, the lowest performance enhancement is -6%, the highest one is 6%, and the average is 0%. In our scheme, its lowest enhancement is -8%, the highest one is 7%, and the average is 0%.

To know the relationship between LDA's accuracy and TCP's performance improvement, we compared Figure 4 with Figure 1 which shows the accuracy of each LDA. While the difference in accuracy ranges from about 10% to 90%, the difference in the average of performance enhancement ranges from -1% to 2%. In other words, there is little difference in the performance enhancement even with a big difference in accuracy. Throughout the observation in the simulation, we ascribe the reason to many spurious retransmission timeouts (RTO).



(a)

(b)

**Figure 3. Simulation results in *C* group**

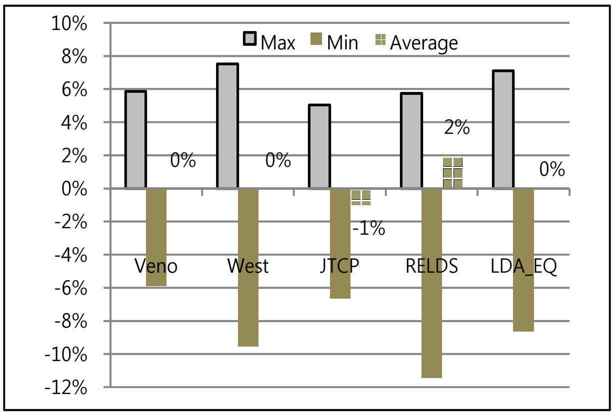


**Figure 4. Simulation Results in *M* group**

The simulation results show that without modifying TCP's RTO recovery based on the detection information of a LDA, it is difficult to improve TCP's performance even though the accuracy is high. Detecting spurious retransmission timeouts or modifying RTO recovery is another emerging research issue in wireless networks, and is out of the scope in this paper. Thus, we leave it for the future work.

## V. DISCUSSION AND CONCLUSION

TCP assumes any packet loss as an indication of congestion, and provides two methods to detect packet loss: fast retransmits for light congestion, and retransmission timeouts (RTO) for heavy congestion. If TCP receives three successive duplicate acknowledgements, it halves its congestion window size assuming a packet is lost due to light congestion. If a TCP sender does not receive a new ACK before the retransmission timeout expires, it initializes its congestion window size to one segment assuming a packet is lost due to heavy congestion.

Such TCP's assumption that all packets are lost due to congestion is not appropriate in wireless networks because most packet losses are caused by wireless transmission errors. Thus, TCP unnecessarily reduces its sending rate due to wireless losses in wireless networks. To avoid the performance degradation of TCP, previous LDAs have been proposed to detect if a packet is lost due to wireless transmission errors or due to congestion. Then, the detection information of a LDA is supposed to be used in the fast recovery [13] of TCP to avoid reducing the congestion window size in half in case of wireless losses. Although these schemes improved $A_w$, these could not avoid sacrificing $A_c$. Thus, each LDA has the trade-off relationship between $A_c$ and $A_w$ which changes based on its threshold.

In this paper, throughout the extensive simulations, we aimed to suggest good guidelines for deciding the best threshold for all LDAs by investigating the trade-off relationship between $A_c$ and $A_w$. Additionally, we aimed to measure the performance improvement as $A_w$ increases, and the performance degradation as $A_c$ decreases. For this, we modified TCP's fast recovery algorithm not to halve congestion window size when a TCP sender receives three duplicate ACKs due to wireless losses. Then, we simulated more than 200 different scenarios in QualNet [15] to investigate the relationship between LDA' accuracy and TCP's performance improvement.

The simulation results were quite different from what we expected. We could not find any relationship between LDA's accuracy and TCP performance improvement. From a commonsense standpoint, TCP's performance should be improved as the accuracy increases. After considerable investigation, we found that spurious retransmission timeouts happened frequently and these prevent TCP from increasing congestion window size.

There is no doubt that the detection information of a LDA is very important for TCP to avoid unnecessarily reducing its sending rate when a packet is lost due to wireless losses. The observation in this paper, however, shows that the detection information itself is not sufficient to improve TCP's performance and the performance degradation due to wireless losses could not be avoidable without detecting spurious retransmission timeouts and modifying the RTO recovery based on the detection information of a LDA.

Detecting spurious retransmission timeouts or modifying the RTO recovery is another emerging research issue in wireless networks, and it is out of the scope in this paper. Although there are several previous works about detecting spurious retransmission timeouts, unfortunately, there is no research so far how to modify the fast recovery and the RTO recovery based on the detection information to maximize TCP's performance. Without such research, it is very difficult to avoid TCP's performance reduction caused by wireless losses. Therefore, in the near future, we will work on how to utilize the detection information of a LDA to improve TCP's performance to the max.

## REFERENCES

[1] C.P. Fu, and S.C. Liew, "TCP Veno: TCP enhancement for transmission over wireless access networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 2, pp. 216-228, Feb. 2003.

[2] C.-H. Lim, and J.-W. Jang, "Robust end-to-end loss differentiation scheme for transport control protocol over wired/wireless networks," *IET Communications*, vol. 2, no. 2, pp. 284-291, Feb. 2008.

[3] E. H.-K. Wu, and M.-Z. Chen, "JTCP: jitterbased TCP for heterogeneous wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 4, pp. 757-766, May 2004.

[4] G. Yang, R. Wang, M. Gerla, and M.Y. Sanadidi, "TCPW bulk repeat," *Computer Communications*, vol. 28, no. 5, pp. 507-518, Mar. 2005.

[5] J. Postel, *Transmission control protocol, RFC 793*, Sept. 1981. [TCP].

[6] L.-P. Tung, W.-K. Shih, T.-C. Cho, Y.S. Sun, and M.C. Chen, "TCP Throughput Enhancement over Wireless Mesh Networks," *IEEE Communications Magazine*, vol. 45, no. 11, pp. 64-70, Nov. 2007.

[7] L.S. Brakmo, and L.L. Peterson, "TCP Vegas: End to end congestion avoidance on a global Internet," *IEEE J. Select. Areas. Commun.*, vol. 13, no. 8, pp. 1465-1480, Oct. 1995.

[8] M.-Y. Park, S.-H. Chung, and P. Sreekumari, "Estimating Rate of Queue Usage to Differentiate Cause of Packet Loss in Multi-hop Wireless Networks," *33rd Annual IEEE International Computer Software and Applications Conference*, pp. 500-508, USA, July 2009.

[9] N.K.G. Samaraweera, "Non-congestion packet loss detection for TCP error recovery using wireless links," *IEE Proceedings Communications*, vol. 146, no. 4, pp. 222-230, Aug. 1999.

[10] P. Sarolahti, M. Kojo, and K. Raatikainen, "F-RTO: an enhanced recovery algorithm for TCP retransmission timeouts", *SIGCOMM Comput. Commun. Rev.* vol. 33, no. 2, Apr. 2003.

[11] S. Cen, P.C. Cosman, and G.M. Voelker, "End-to-end differentiation of congestion and wireless losses," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 703-717, Oct. 2003.

[12] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on TCP throughput and loss," *Proc. of 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1744-1753, San Francisco, USA, Mar. 2003.

[13] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control. RFC 2581, April 1999.

[14] A. Gurtov, and R. Ludwig, "Responding to spurious timeouts in TCP," *Proc. Of 22nd Annual Joint Conference of the IEEE Computer and Communications*, pp. 2312-2322, Mar. 2003.

[15] http://www.scalable-networks.com/index.php