International Journal of u- and e- Service, Science and Technology
Vol. 4, No. 3, September, 2011

A Comparative Study on Responding Methods for TCP's Fast Recovery in Wireless Networks

Mi-Young Park, Sang-Hwa Chung, Kyeong-Ae Shin, Guangjie Han

Dept. of Computer Engineering, Pusan National University, South Korea
openmp@korea.com,
Dept. of Computer Engineering, Pusan National University, South Korea
shchung@pusan.ac.kr,
Dept. of Medical Information System, Dongju College University, South Korea
kashin@dongju.ac.kr,
Dept. of Information & Communication System, Hohai University, P.R.China
kashin@dongju.ac.kr

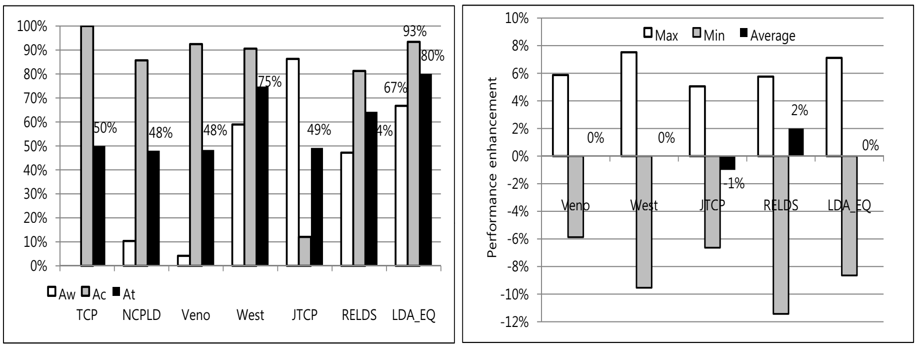
Abstract. When TCP operates in wireless networks, it suffers severe performance degradation because TCP's fast retransmit/fast recovery (FR/FR) is often triggered regardless of congestion when a packet is lost due to wireless transmission errors. Although several loss differentiation schemes have been proposed to distinguish the cause of triggered FR/FRs, there has been a little try to find out an appropriate responding method to utilize the detection information of a LDA. In this paper, we observe the relationship between a responding method and TCP's performance, and investigate the best way to respond to the FR/FRs non-related to congestion in order to improve TCP's performance to the max. Our experiment results emphasize the importance of the responding method, which is overlooked in previous works, and suggest the best response for the FR/FRs non-related to congestion.

1 Introduction

One of the objectives in TCP [5, 8] is to use available bandwidth to the max, and to prevent congestion in the networks by appropriately controlling its transmission rate. When a TCP sender receives three successive duplicate acknowledgements (3DUPACK) indicating a packet loss, it assumes that the packet is lost due to congestion. Then, TCP triggers a fast retransmit/fast recovery (FR/FR) to retransmit the lost packet as well as to reduce its transmission rate [8].

Such a mechanism works very well in wired networks where most packets are lost due to congestion. However, in wireless networks where packets are lost due to wireless transmission errors, TCP's FR/FRs are often triggered regardless of congestion [6, 12]. We call the FR/FRs non-related to congestion *wireless FR/FRs*, and we call the FR/FRs triggered by congestion *congestion FR/FRs*. Due to wireless FR/FRs, TCP unnecessarily reduces its transmission rate, and then underutilizes the available bandwidth. Consequently, TCP's performance severely degrades in wireless networks.

Since TCP has no ability to distinguish if a FR/FR is triggered due to congestion or due to wireless transmission errors, several loss differentiation schemes (LDAs) have



(a) (b)

Fig. 1. Relation between accuracy of LDA and TCP's performance enhancement

been proposed to distinguish the cause of triggered FR/FRs. These LDA schemes can detect wireless FR/FRs by distinguishing congestion losses from wireless losses when a TCP sender receives 3DUPACK. To improve TCP's performance in wireless networks, these LDAs assume that the detected wireless FR/FRs will be responded by setting TCP's congestion control state to the previous state instead of being responded by halving TCP's transmission rate. Unfortunately, our previous work [9] showed that restoring TCP's congestion control state to the previous state does not always improve TCP's performance; TCP's performance degrades even with the best LDA in some simulation scenarios.

In this paper, we will observe why TCP's performance decreases in some cases, and investigate the best way to respond to wireless FR/FRs in order to improve TCP's performance to the max. For this, we designed about 120 different simulation scenarios by setting different values for network parameters such as the buffer size, the number of hops, and the loss rate, and then we grouped all the scenarios into two groups (W, and C group) according to our intention. In each group, we measure how much each responding method can improve TCP's performance when all wireless FR/FRs are detected, and we measure how much each responding method degrades TCP's performance when a congestion FR/FR is mistakenly detected as a wireless FR/FR. Our extensive experiment results emphasize the importance of the responding method on improving TCP's performance, and suggest the best response for the detected wireless FR/FRs.

In the following section, we introduce previous loss differentiation schemes which can detect wireless FR/FRs, and then we explain our motivation in section 3. In section 4, we investigate the performance variation of TCP based on different responding methods, and suggest the best response for the detected wireless FR/FRs. Lastly, we conclude this paper with our future work.

2 Related Works

To avoid TCP's performance degradation caused by wireless FR/FRs in wireless networks, several loss differentiation schemes (LDAs) have been proposed to detect wireless FR/FRs by distinguishing congestion losses from wireless losses when a TCP sender receives 3DUPACK. When any wireless loss is detected by a LDA, the LDA assumes that TCP's FR/FR is triggered regardless of congestion, and treats it as a wireless FR/FR, otherwise, it treats it as a congestion FR/FR assuming that the FR/FR is triggered due to congestion. Here, we introduce five end-to-end loss differentiation algorithms (LDAs): NCPLD [11], Veno [1], West [4], JTCP [3], and RELDS [2].

Samaraweera [11] 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, and it assumes the triggered FR/FR as a wireless FR/FR.

TCP Veno [1] estimates the backlog packets (N) in the buffer using Vegas's mechanism [7]. When a TCP sender receives 3DUPACK, Veno compares N with a threshold 3. If $N < 3$, Veno assumes the packet loss as wireless losses, and treats the triggered FR/FR as a wireless FR/FR.

Yang [4] computes the two thresholds, $B_{spikestart}$ and $B_{spikeend}$ using RTT in order to identify the spike state of the current connection. Any packet losses in the spike state are considered as congestion losses, and the triggered FR/FRs are assumed as congestion FR/FRs. Otherwise, all the triggered FR/FR will be assumed as wireless FR/FRs.

JTCP [3] calculates a threshold (*Jr*) which is the average of the inter arrival jitter during one round-trip time. When a TCP sender receives 3DUPACK, it checks if the time receiving the 3DUPACK 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 not satisfied, it ascribes the packet loss to wireless transmission errors and it assumes the triggered FR/FR as a wireless FR/FR.

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 not satisfied when a TCP sender receives 3DUPACK, it assumes the packet loss as wireless losses, and treats the triggered FR/FR as a wireless FR/FR.

3 Motivation and observation

In our previous work, we also suggested an end-to-end loss differentiation scheme (LDA_EQ) [10] to distinguish the cause of the triggered FR/FR by estimating the rate of queue usage. If the estimated queue usage is smaller than a certain threshold when 3DUPACK is received, our scheme assumes the packet loss as wireless losses, and treats the triggered FR/FR as a wireless FR/FR. In the work, we measured three accuracies:

(1) how accurately it can detect wireless FR/FRs (A_w), (2) how accurately it can detect congestion FR/FRs (A_c), and (3) the average (A_t) of A_w and A_c. Figure 1(a) shows the result. In the figure, A_w of our scheme (LDA_EQ) is 67%, and its A_c is 93%, thus, its average accuracy (A_t) is 80% which is the highest among the LDAs. In case of TCP which has no ability to detect wireless FR/FRs, its A_w, A_c and A_t are 0%, 100%, and 50% respectively.

In our another work [9], we analyzed the effect of LDAs on improving TCP performance. Whenever a wireless FR/FR is triggered, it is responded by restoring TCP's congestion control state to the previous state instead of being responded by reducing the transmission rate in the experiments of the work [9]. That is because the existing LDAs basically assume that TCP's performance will be improved by not reducing the transmission rate when a wireless FR/FR is detected.

Figure 1(b) shows TCP's performance improvement when each LDA is applied to TCP. In the figure, each bar graph in each LDA shows the lowest, the highest, and the average of the performance enhancement when each LDA is applied. For example, when Veno is applied to TCP, the lowest performance enhancement is -6%, the highest one is 6%, and the average is 0%. With our scheme (LDA_EQ), its lowest enhancement is -8%, the highest one is 7%, and the average is 0%.

This result shows that TCP's performance is not always improved with any of the LDAs. In some scenarios, TCP's performance degraded even with the best LDA. To find out why, we traced and analyzed all the trace files of the simulation scenarios carefully, and then, we found the reason in the responding method. How to respond is very critical on improving TCP's performance because there will be no performance enhancement if all wireless FR/FRs are responded by the same way of TCP.

From now on, let us explain our observations. TCP uses the two variables, the congestion window size (*cwnd*) and the slow start threshold (*ssthresh*), to control its transmission rate according to the network traffic. For example, when it is assumed that the network is congested, TCP reduces its transmission rate by decreasing the values of *cwnd* and *ssthresh* while TCP increases its transmission rate by increasing *cwnd* when the network is not congested.

Firstly, let us suppose that a wireless FR/FR is detected by a LDA when *cwnd* and *ssthresh* are too small values. Just recovering *cwnd* and *ssthresh* to the previous values does not make any big difference between the performances of TCP with/without a LDA. If the wireless FR/FR is detected when *cwnd* and *ssthresh* are large values, however, TCP's performance will be improved significantly. This is the reason why one LDA improves TCP's performance significantly in some scenarios while it hardly improves the performance in other scenarios.

In the meantime, with the larger values of *cwnd* and *ssthresh*, TCP's performance with a LDA can be degraded if the bandwidth is not enough or if the packet loss rate is high. This is because the larger values of *cwnd* and *ssthresh* might cause serious congestion or might cause burst losses. In such cases, TCP's performance can be reduced even with the best LDA.

Secondly, let us suppose that one LDA (A) detects 10 wireless FR/FRs when *cwnd* and *ssthresh* are too small values, and another LDA (B) detects 1 wireless FR/FR when *cwnd* and *ssthresh* are large values in the same scenario. In this case, B algorithm may

Table 1. Responding Methods

Responding Methods	TCP's Congestion Control State	
0	TCP's original response	
1	restoring to the previous congestion control state	
	Slow Start Part	
	cwnd	**ssthresh**
11	the size of one segment	65535
12	TCP's response (*cwnd*)	65535
13	the previous value of *cwnd*	65535
14	the average of *cwnd*	65535
15	the maximum of *cwnd*	65535
16	the size of sender's buffer	65535
17	bandwidth-delay product	65535
	Congestion Avoidance Part	
	cwnd	**ssthresh**
21	the size of one segment	
22	TCP's response (*cwnd*)	
23	the previous value of *cwnd*	
24	the average of *cwnd*	
25	the maximum of *cwnd*	
26	the size of sender's buffer	
27	bandwidth-delay product	

improve TCP's performance much more than A algorithm even though the accuracy of A algorithm is higher. This indicates that the accuracy of a LDA can be meaningless without a good responding method.

All existing LDA schemes assume that the responding method of recovering TCP's congestion control state to the previous state will improve TCP's performance, and this assumption seems reasonable in a common sense. But, our observations inform us that recovering to the previous state does not always improve TCP's performance. Furthermore there has been a little try to find out the best responding method to utilize the detection information of a LDA. Thus, in this paper, we observe the relationship between responding methods and TCP's performance, and investigate the best way to respond to wireless FR/FRs in order to improve TCP's performance to the max.

4 A comparative study on responding methods

4.1 Responding Methods

To find the best response, we made 14 different responding methods for wireless FR/FRs based on our observations. Table 1 shows all the responding methods used in our experiments. First of all, we classified the responding methods into two groups: the slow start part and the congestion avoidance part. This is to check if TCP should enter the

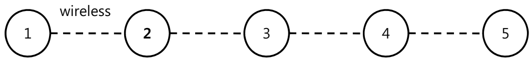


Fig. 2. 5-hops wireless chain topology

slow start phase or the congestion avoidance phase when a wireless FR/FR is detected. Although TCP enters the congestion avoidance phase when a FR/FR is triggered, it might be necessary to make TCP enter the slow start phase to quickly utilize available bandwidth to the max if the FR/FR is triggered regardless of congestion. This is why we classified the methods into the two groups.

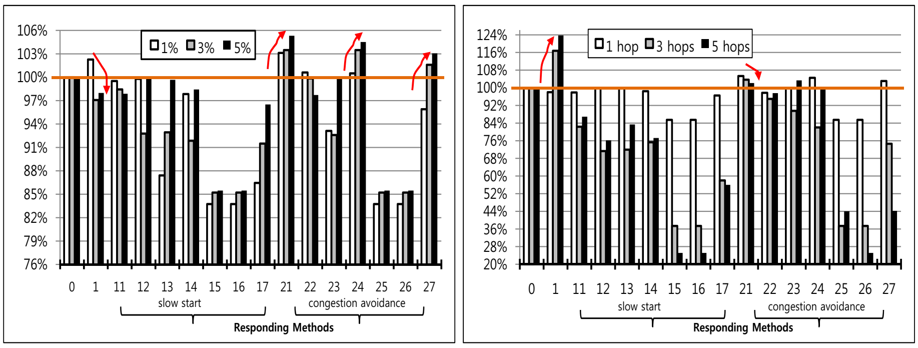
Then, each group is again classified into seven methods according to $cwnd$'s size. This is to check if $cwnd$ should be set by small or larger values when a wireless FR/FR is detected. Deciding the size of $cwnd$ is a difficult problem because there is always a trade-off relationship. With a small size of $cwnd$, TCP's transmission rate is low while there is a little possibility to have burst losses. With a big size of $cwnd$, TCP's transmission rate is high while there is a high possibility to have burst losses and additional congestion. By testing various sizes of *cwnd*, we will check the appropriate size of *cwnd* for the detected wireless FR/FRs.

In the table 1, the method 0 is TCP's response for the triggered FR/FR when a TCP sender receives 3DUPACK. This method will be used to compare with other methods to check if other methods is better than TCP's response. With the method 1, a TCP sender recovers its transmission rate to the previous state just before a wireless FR/FR is detected; all LDAs assume that the method 1 will be used when a wireless FR/FR is triggered.

With the methods from 11 to 17, TCP enters the slow start phase when a wireless FR/FR is detected. Thus, $ssthresh$ is set to the max value (65535) and $cwnd$ is set to one of the values shown at "$cwnd$" column of the table 1. For $cwnd$, seven different sizes are used: the size of one segment, TCP's response for $cwnd$, the previous value just before a wireless FR/FR is detected, the average of $cwnd$, the maximum value of $cwnd$, the size of sender's buffer, and bandwidth-delay product. For example, when the method 11 is used as a responding method for the wireless FR/FRs, TCP's $cwnd$ and $ssthresh$ will be set to "the size of one segment", and "65535" respectively, instead of halving *cwnd* and *ssthresh*. With the methods from 21 to 27, TCP enters the congestion avoidance phase when a wireless FR/FR is detected. For example, when the method 21 is used, TCP's $cwnd$ and $ssthresh$ will be set to "the size of one segment" when a wireless FR/FR is detected.

4.2 Simulation Scenarios

To evaluate and compare TCP's performance enhancement according to different responding methods, we conducted simulation-based experiment using QualNet 4.5 [13]. Figure 2 shows the 5-hops chain topology of IEEE 802.11 wireless nodes used in our experiment. The bandwidth of the wireless channel is 11Mbps, and we set DropTail as its queuing policy. The maximum segment size of TCP is equal to 1K bytes, and



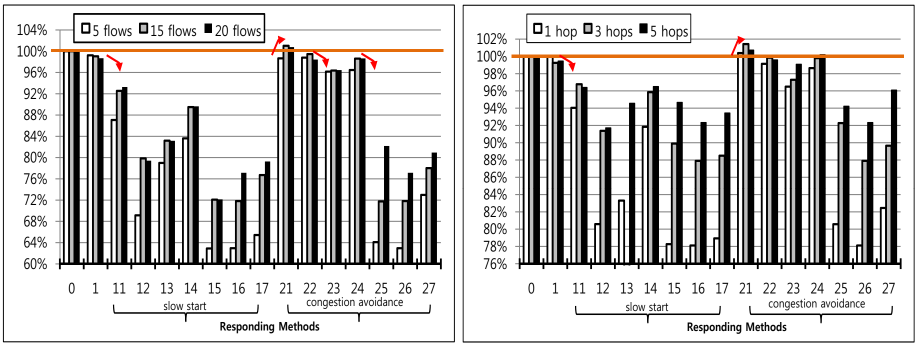
(a) (b)

Fig. 3. TCP's performance enhancements according to the responding methods in W group

the packet size equal to 1K bytes. The congestion window size *cwnd* is limited to 48 packets, and all TCP flows are originated at the first node (node 1) and destined to one of nodes on the right side of Figure 2. In our experiment, each simulation scenario lasts about 200 seconds, and data packets of TCP are continually transmitted during the simulation time after the warm-up period (35 seconds). In this experiment, we aimed to find the best response for the wireless FR/FRs. We assume that the best responding method should improve TCP's performance as highly as possible when a wireless FR/FR is detected and also should cause little performance degradation when a LDA misjudges congestion FR/FRs to wireless FR/FRs. Thus, to find the best response, we measure how much each responding method can improve TCP's performance when all wireless FR/FRs are detected, and also we measure how much each responding method degrades TCP's performance when all congestion FR/FRs are mistakenly detected as wireless FR/FRs. For this, we designed about 120 different simulation scenarios by setting different values for network parameters such as the buffer size, the number of hops, and the loss rate, and then we grouped all the scenarios into two groups: W, and C group.

W group, which consists of 60 different simulation scenarios, is designed to measure how much each responding method can improve TCP's performance when all wireless FR/FRs are detected. All simulation scenarios in this group have packet losses caused by only wireless transmission errors, and only one TCP flow is used to avoid causing congestion. Thus, all TCP's FR/FRs triggered in this group are wireless FR/FRs.

C group, which consists of 60 different simulation scenarios, is designed to measure the performance degradation of TCP when each responding method is applied to congestion FR/FRs. Each scenario in this group has packet losses caused by only congestion, and we increased the number of TCP flows gradually to make different levels of congestion. Thus, all the triggered FR/FRs in this group are congestion FR/FRs.



(a) (b)

Fig. 4. TCP's performance degradation according to the responding methods in C group

4.3 Results and Analysis

Figure 3 shows TCP's performance enhancement according to the different responding methods in W group. In this group, we checked how much each responding method can improve TCP's performance when it can detect all wireless FR/FRs. Whenever we applied each responding method in the table 1 to the same scenario, we modified TCP's fast recovery to react like the responding method when FR/FRs are triggered.

In the figure, x axis represents the responding methods, and y axis represents TCP's performance enhancement when each responding method is applied. To check if each responding method improves TCP's performance, we set the performance of TCP as the criterion for comparison. Thus, when the method 0 (TCP's original response) is used, we set the performance of TCP as 100%, and then we compared it with the enhancements when the other methods are used. For example, in the figure 3(a), we can see that the methods (1, 21, 24, and 27) improve TCP's performance compared to TCP's original response (the responding method 0) while the other responding methods degrade TCP's performance.

Figure 3(a) shows TCP's performance enhancement according to the packet loss rate when each responding method is used at a 1-hop TCP communication. Each bar graph in each responding method shows TCP's performance enhancement when the loss rate is 1%, 3%, and 5% respectively. For example, with the responding method 1, TCP's performance is improved when the loss rate is 1%, but the performance degrades as the loss rate increases (3% and 5%). While the responding method 1 tends to degrade TCP's performance as the loss rate increases, the method 21 tends to increase TCP's performance as the loss rate increases. Among the four methods (1, 21, 24, and 27), the responding method 21 is the best when the loss rate is high.

Figure 3(b) shows TCP's performance enhancement according to the number of hops in each responding method. Each bar graph in each responding method shows TCP's performance enhancement in a 1-hop, 3-hops, and 5-hops TCP communication, respectively. In this graph, we can see that the responding method 1 enhances TCP's performance significantly as the number of hops increases. Although the method 21 improves TCP's performance, it tends to decrease as the number of hops increases. Thus, the responding method 1 is the best when the number of hops increases.

From the figure 3, we can see that all the responding methods in the slow start part (from 11 to 17) do not improve TCP's performance while some of the methods in the congestion avoidance part (from 21 to 27) enhance TCP's performance. Additionally, we can see that as the size of $cwnd$ increases (specially the methods 15, 16, 25, and 26) TCP's performance degrades severely. This is because, as $cwnd$ becomes larger, there is high probability not only to have burst losses when the packet loss rate is high, but also to have additional congestion when the bandwidth is not sufficient. This observation informs us that it is better to set *cwnd* with small values and to let a TCP sender enter the congestion avoidance phase.

Figure 4 shows TCP's performance according to the different responding methods in C group. In this group, we aimed to measure how each responding method degrades TCP's performance when all congestion FR/FRs are mistakenly detected as wireless FR/FRs. This is necessary because there is no perfect LDA and all LDA have a possibility to misjudge congestion FR/FRs to wireless FR/FRs. When a LDA misjudges a congestion FR/FR as a wireless FR/FR, the best responding method should hardly degrade TCP's performance.

Figure 4(a) shows TCP's performance according to the number of TCP flows when each responding method is used in a 1-hop TCP communication. Each bar graph in each responding method shows the performance when the number of TCP flows is 5 flows, 15 flows, and 20 flows, respectively. In this figure, we can see that the responding methods (1, 22, and 24) show the similar performance degradation, and the responding method 21 shows the least performance degradation. The other methods degrade TCP's performance severely.

Figure 4(b) shows TCP's performance according to the number of hops in each responding method. Each bar graph in the each responding method shows the performance in a 1-hop, 3-hops, and 5-hops TCP communication, respectively. This graph also shows the similar result with that in Figure 4(a). In the graph, we can see that the responding method 21 hardly degrades TCP's performance compared to the other responding methods. From the two graphs in Figure 4, we can know that the responding method 21 is the best response for the case when congestion FR/FRs are mistakenly detected as wireless FR/FRs.

5 Conclusions

To find the best response for the detected wireless FR/FRs, in this paper, we compared TCP's performance based on the 15 different responding methods as the loss rate increases or the number of hops increases. Our experiment results show that the responding method 1 is the best as the number of hops increases, and the responding method 21 is the best as the loss rate increases. Also, it shows that, when a LDA detects mistakenly congestion FR/FRs as wireless FR/FRs, the responding method 21 is the best in avoiding TCP's performance degradation. Among the 15 different methods, there is no single responding method which is the best for all network conditions. This informs us that the responding method needs to dynamically react to wireless FR/FRs according to the network conditions. In the near future we will develop a dynamic responding method for the detected wireless FR/FRs in order to improve TCP's performance in wireless networks.

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. Allman, V. Paxson, and W. Stevens, TCP Congestion Control. RFC 2581, April 1999.
9. M. Park, and S. Chung, "Analyzing Effect of Loss Differentiation Algorithms on Improving TCP Performance", *12th International Conference on Advanced Communication Technology*, pp. 734-742, Jan. 2010.
10. M. Park, S. Chung, P. Sreekumari, "End-to-end loss differentiation algorithm based on estimation of queue usage in multi-hop wireless networks", *IEICE Trans. Inf. & Syst.*, vol.E92-D, no.10, pp.2082-2093, Oct. 2009.
11. 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.
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. http://www.scalable-networks.com/index.php