Online Loss Differentiation Algorithm with One-Way Delay for TCP Performance Enhancement

Minsu Shin†, Mankyu Park†, Byungchul Kim††, Jaeyong Lee††, and Deockgil Oh†

†Dept. of Satellite Wireless Conv., Electronics and Telecommunications Research Institute, Daejeon, 305-700 KOREA
††Dept. of Information Communication Engineering, Chungnam National University, Daejeon, 305-764 KOREA

Summary
In this paper for the performance enhancement of TCP protocol in the mixed wired/wireless network, packet loss differentiation algorithm is proposed which distinguishes the nature of losses and decides congestion control action accordingly. TCP protocol takes packet loss as an indication of network congestion so that it reduces its congestion window into half. However, packet loss can be more likely induced by wireless bit error in the wireless network and the performance of TCP in such a network could be unnecessarily degraded by the lack of the ability to identify the cause of the losses. To reduce the effect of the problem, we propose the accurate online loss differentiation algorithm making use of the one-way delay on the forward path. Our contributions are in twofold. First, we introduce the detection and removal of the clock offset existing between sender and receiver clocks only with the information available at the sender side which make our algorithm easy to deploy, and therefore calculate accurate one-way delay on the forward path. Second, we consider the possibility that wireless loss would occur even in the situation of network congestion, and to the best of our knowledge it is the only algorithm that takes this probability into account. We propose two-phase loss differentiation algorithm that uses the ratio of the average one-way delay in the lossless period and lossy period to decide the existence of congestion in the first phase, and the statistical property of the one-way delay for the congestion loss in the second phase. From the simulation results, we showed that our algorithm can provide better accuracy both for the congestion and wireless losses.

Key words:
TCP, loss differentiation, one-way delay, clock synchronization, congestion loss

1. Introduction

As the congestion control algorithm of TCP is basically to take packet loss as an indication of network congestion, thus consequently throughput performance of TCP protocol depends on the network congestion state. However, in the networks including wireless link as a part of the connection many packet losses can be induced by wireless bit error. Since standard TCP treats these wireless losses as an indication of the congestion and reduces its congestion window accordingly, unnecessary performance degradation problem is caused. Many researches are devoted for performance enhancement of TCP in mixed wired/wireless networks by addressing the decision problem of congestion and most of them have focused on the differentiation of the cause of packet losses[1-4]. Packet loss differentiation algorithms that can be applied to the sender side would be classified into 2 groups. First, in the network based schemes receiver and intermediate node within the path can inform the sender of network congestion status or information for the specific packet loss explicitly[6-8], and the sender controls its transmission rate accordingly. While this method can provide most accurate results, it requires that all the network equipment, sender and receiver should be modified to support such an interaction, which limits its use in the real network. Second, end-to-end schemes estimate the level of congestion only with the information measured in the sender and/or receiver and identifies the nature of the packet loss according to the estimated congestion status. Because these methods take use of the estimation with limited information acquired from the sender and /or receiver side measurements, their decision accuracy may be lower to some extents comparing to the network based schemes. However, since this type of schemes is easy to implement and thus readily deployed in the real networks, in the paper we are focus on the end-to-end schemes that can be applied for any transport layer protocol by requiring only sender side modifications.
Among the measurable data in the sender side, we can utilize Round trip time(RTT) and ACK arrival rate to differentiate the cause of packet losses. However it is very hard to find the relation between those information and the characteristics of packet losses because there is no enough correlations between them. Timestamp option in the TCP header was proposed to provide a way to measure RTT accurately for the purpose of performance enhancement of TCP in the Long-Fat Network such as broadband satellite network[9]. This timestamp option can be used to classify the cause of packet losses by measuring the variation of packet inter-arrival time[13], or by estimating buffer state

Manuscript received April 5, 2011
Manuscript revised April 20, 2011

in the network according to the estimated relative one-way delay[14]. While for these algorithms to have expected performance, clock synchronization between two end systems should be assumed, their clocks are usually not synchronized in the real network because they are operating on the different platforms. In Fig. 1, the trend of one-way delay is shown which is measured at the receiver using sender timestamp in the environment that they are connected through a router. While we have expected that one-way delay at the initial step is negligibly small, it is turned out to be around 200ms of initial time difference between the sender and the receiver. This time difference is caused by the relative difference of the time reported by two clocks and is called *clock offset*. Also it is shown that there is a linearly increasing trend in the one-way delay measurement. This is caused by the difference of the frequency the clock progresses, which is called *clock skew*. Due to the clock skew, we can find that it can cause 370 ms of measurement error during the period of 230 minutes. There have been much efforts to detect and remove the clock offset and skew in the literature[15][16].

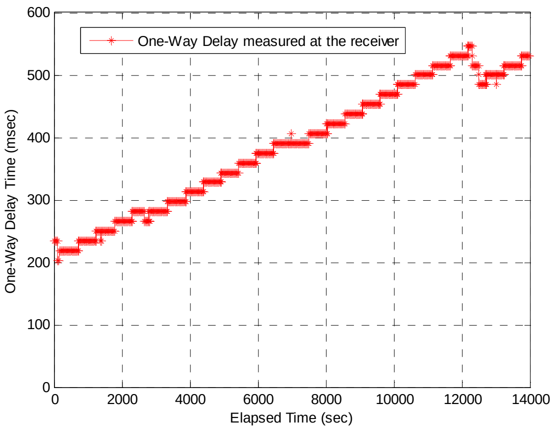

Fig. 1 Effect of the Lack of Clock Synchronization between Sender and Receiver Clocks.

Previous works to classify the cause of packet losses have the limitations as following. First, TCP timestamp option is widely used for the measurement of time information in this area. However, it is hard to estimate network congestion status with the information due to the clock dynamics existing between two end systems such as clock offset and clock skew. Even though many researchers have made their efforts to have clock synchronization[17][18], most of the works have the limitations in their use for the end-to-end protocols because for the accurate estimation they require large measurement samples during the long period. Second, with the fact that relative one-way delay (ROTT) can more accurately estimate the queuing delay than RTT does[15], many algorithms take ROTT as a mean to differentiate the nature of packet losses[1][13][19]. However, these algorithms require the receiver side modifications to measure ROTT using sender timestamps and moreover the signaling protocol to deliver this ROTT value measured at the receiver to the sender. These modifications pose limitations to realize the algorithms. Third, since most algorithms are considering the pattern of the packet delay under a specific target network topology, their accuracy may not be guaranteed in the different scenarios.

With these considerations in mind, we propose efficient packet loss differentiation algorithm in the paper and our contributions to solve the problems stated above are twofold as follows. First, since the performance of the TCP depends mainly on the packet losses over the forward path, it is critical to estimate forward one-way delay(FOTT) accurately. Therefore, we also take variation of forward one-way delay as a criterion to decide whether it is congestion loss or not. Since we first have clock synchronization between sender and receiver only using measurable information at the sender side and thereby estimate forward one-way delay accurately, we can avoid the aforementioned problems which receiver based schemes usually have. Second, in this paper we are concerned wireless losses which still exist with the same probability even in the congestion situation. Recognizing these wireless losses with the statistical characteristics of forward one-way delay variations, we can provide better accuracy performance than the previous works do as it is verified through the simulation.

The paper is organized as follows. In Sec. 2, we briefly review some of the previous works in the arena and describe the proposed online packet loss differentiation algorithm using forward one-way delay in Sec. 3. Performance evaluation with the NS-2 Simulator is shown in Sec. 4 and we conclude the paper with summary and further considerations in Sec. 5.

2. Previous Works

Many studies for the packet loss differentiation have been done to enhance wireless TCP performance until recently. In this section, we describe some of the previous works to look into their characteristics and benefits. As stated in Sec. 1, there are two types of approaches to differentiate the nature of packet losses at the sender side. One is for the receiver and intermediate node over the path to send congestion information to the sender explicitly. This type of algorithm is called network based scheme. The other is for the sender to estimate the network congestion status with the measurable data at the sender and/or receiver, and to decide whether it is congestion loss or not according to the estimation results. This type of algorithms is called end-to-end scheme. The proposed algorithm falls into this end-to-end scheme in that we do not need any information from the intermediate node in the network and just require sender only modifications for implementation. Therefore, we focus on the end-to-end schemes to review in the section.

Biaz[1] proposed the method to differentiate the cause of packet losses with packet inter-arrival times measured at the receiver. The receiver measures inter-arrival times for each receiving packets and sends this information to the sender to allow it to control its transmission rate appropriately. In his scheme, n missing packets are assumed to be lost due to the wireless bit error when the packet inter-arrival time, T_i, is satisfied with Eq.(1). Otherwise, they are identified as congestion losses.

$$(n+1)\times T_{\min} \le T_i < (n+2)\times T_{\min}\,. \tag{1}$$

where T_i denotes the time between the arrivals of the last in-sequence packet P_i received by the receiver before a loss happened and the first out-of-order packet P_{i+n+1} received after the loss, T_{min} denotes the minimum packet inter-arrival time observed so far by the receiver during the connection, and n is the number of packets losses.

The concept here is that based on the arrival time of P_i, if P_{i+n+1} arrives around the time that it should have arrived, we can assume the missing packets were properly transmitted and lost to wireless errors. Otherwise, the missing packets are probably queued or dropped at the buffer, thus assumed to be lost due to congestion. The Biaz scheme works best when the last link is both the wireless link and the bottleneck link of the connection, and is not shared by other connections competing for the link. It has been proven that it misclassifies a significant number of congestion losses and thus induces more congestion in the network since it prevents the sender from reducing its congestion window[11].

Tobe *et. al.*[3] proposed the spike scheme which differentiate loss types using ROTT measured at the receiver. ROTT is a measure of the time a packet takes to travel from the sender to the receiver. Since the sending and receiving times are measured at the sender and receiver separately, the absolute value of delay is difficult to obtain due to the clock skew between the two clocks and thus it is named *relative*. They define the spike-train as the continuous observation of high variations of the ROTT and assume that it occurs only during the period of congestion which is called *spike state*. If the connection is in the spike state, losses are assumed to be due to congestion. Otherwise, losses are assumed to be wireless. The spike state is determined according to the threshold values in Eq.(2). If the connection is currently not in the spike state, and the ROTT for packet i exceeds the threshold $B_{\text{spikestart}}$, then the algorithm enters the spike state. Otherwise, if the connection is currently in the spike state, and the ROTT for packet i is less than a second threshold B_{spikeend}, the algorithm leaves the spike state. When the receiver detects a loss because of a gap in the sequence number of received packets, it classifies the loss based on the current state.

$$\begin{aligned} B_{\text{spikestart}} &= ROTT_{\min} + 20ms \\ B_{\text{spikeend}} &= ROTT_{\min} + 5ms \end{aligned}\,. \tag{2}$$

Since the threshold values in spike scheme is fixed as in Eq.(2), the performance of the spike scheme tends to degraded depending on the network traffic conditions. Therefore several variants to modify the formula for the threshold values have been presented to make them adapt to the overall network delays[11][21][29].

JTCP [13] was proposed to use the jitter ratio (J_r) to predict the reason of packet loss, and further adapt its congestion control. To identify the cause of packet losses, JTCP calculates average jitter ratio J_r of the inter-arrival jitter during one round trip time as follows.

$$J_r = \frac{(t_i^r - t_{i-w}^r) - (t_i^s - t_{i-w}^s)}{(t_i^r - t_{i-w}^r)}\,. \tag{3}$$

where w denotes the current congestion window size, t_i^r and t_i^s denote receiving time and sending time respectively of the earliest unacked packet i during one RTT period. When the sender receives three duplicate ACKs, it determines if the period for three duplicate ACKs extends to the next RTT and J_r is larger than *1/cwnd*. If the two conditions are satisfied, it ascribes the packet loss to congestion. JTCP has been proven to have low detection accuracy for congestion losses and, on the other hand, high detection accuracy for wireless losses. The reason for low detection accuracy for congestion losses is that the threshold *1/cwnd* is too high to differentiate congestion loss appropriately. From this observation, we can expect that the performance of JTCP could be more adversely affected by large congestion window size when it is applied for the high BDP(Bandwidth-Delay Product) networks.

NCPLD(Non-Congestion Packet Loss Detection) proposed to use delay variation to differentiate the cause of packet losses in the wireless backbone networks[2]. NCPLD decides that the loss is due to the wireless transmission error when the measured RTT is less than the delay threshold as in Eq.(4).

$$\begin{aligned} &if\,(measuredRTT < delayThreshold) \\ &\quad \text{wireless loss} \\ &delayThreshold = RTT_{\min} + 0.5\times RTT \times \frac{BDP}{TotalPipeSize} \end{aligned}\,. \tag{4}$$

where $RTT_{\min}$ is the minimum of measured RTT, *BDP* is the measured bandwidth delay product when TCP sender experiences the minimum RTT, and *TotalPipeSize* is an estimation of the total number of bytes in the network.

NCPLD scheme does not serve much gains at the low error rate comparing to the standard TCP, but on the other hand it provide performance enhancement when the channel error rate is high.

Lim *et. al.* suggested RELDS(Robust End-to-End Loss Differentiation Scheme) to precisely discriminate between congestion losses and wireless losses in the diverse network conditions[20]. RELDS scheme defines threshold as a function of the minimum RTT and the average RTT so as to make the algorithm adaptively react to the level of network congestion. By doing so, the threshold value for identifying the wireless losses increases as the level of congestion increases, and consequently the algorithm can have higher probability to correctly detect congestion losses. The decision rule for classifying the cause of packet losses is as follows.

$$\frac{T - T_{ave}}{T_{dev}} > 2\left(\frac{T_{min}}{T}\right)^3 - 1 \cdot \qquad (5)$$

where T is the current RTT value, T_{ave} is the averaged RTT, and T_{dev} is the deviation of RTT which can be calculated as T_{dev} = (3/4) T_{dev} +(1/4)|T- T_{ave} |. If the rule is satisfied when the sender receives the third duplicate ACK, it assumes the packet loss as congestion losses. RELDS has been initially verified in the mixed wired/wireless network with wireless last hop, and the results have shown that it has high detection accuracy for congestion losses regardless of the channel error rate and network buffer size.

TCPW-BR(Bulk Repeat)[21] was proposed for the wireless networks with high error rate of more than 30%. TCP-W[22] is known to work well in the high BDP networks with reasonably low error rates. On the other hand, TCPW-BR was designed to enhance the performance of TCP-W and provide better performance even in the networks with higher error rates. To this end, TCPW-BR adopts packet loss differentiation mechanism. When the algorithm determines that the loss is due to wireless error, it continuously increases congestion window and does not back off the RTO value. Otherwise, if the algorithm recognizes the congestion losses, the same action is taken as TCP-W. TCPW-BR adopts modified Spike scheme presented by Cen *et. al.*[11], but uses RTT at the sender side instead of ROTT. Based on RTT, the algorithm computes the two thresholds to identify the current network state as follows.

$$\begin{aligned} B_{spikestart} &= RTT_{\min} + 0.4 \times (RTT_{\max} - RTT_{\min}) \\ B_{spikeend} &= RTT_{\min} + 0.05 \times (RTT_{\max} - RTT_{\min}) \end{aligned} \quad . \qquad (6)$$

where $RTT_{\min}$ and $RTT_{\max}$ are the minimum and the maximum of measured RTT, respectively.

The detailed operations are the same as in the Spike scheme[3].

LDA_EQ scheme[14] proposed packet loss differentiation algorithm for the multi-hop wireless network to avoid performance degradation of TCP in such a network by using different TCP congestion control method according to the cause of packet loss. Based on the idea that the queue usage in the links can be accurate measure for the level of congestion, LDA_EQ estimates queue usage in the links from the EROTT(Estimated ROTT) calculation and classifies the packet losses into congestion-related losses and congestion-unrelated losses according to the estimated queue usage.

Most of the works stated before have some disadvantages that their performance can be affected by the network fluctuations over the reverse path because of using RTT value as a measure to identify the loss type and that they require modifications of signaling protocol for the receiver to deliver its measurement data such as packet inter-arrival time to the sender. In this paper, we present efficient forward one-way delay(FOTT) estimation scheme at the sender side and the packet loss differentiation algorithm taking the estimated forward one-way delay as a measure of identifying packet loss types to avoid aforementioned limitations.

3. LDA based on FOTT

To estimate the network congestion status, RTT value is widely used. TCP congestion control needs to take an action only against the congestion losses on the forward path. If RTT is increased by the increase of the one-way delay on the reverse path and TCP takes this RTT increase as an indication of network congestion on the forward path, TCP would trigger its congestion control and thus result in the spurious throughput degradation. In the paper, to reduce the impact of the changes on the reverse path we use forward one-way delay(FOTT) for the measure of packet loss differentiation. As stated in Sec. 1, for the accurate FOTT estimation, we detect and remove the clock offset between two clocks of sender and receiver. We propose simple and rather accurate clock offset estimation scheme with sender only modification to have more accurate estimate of FOTT, and also packet loss differentiation scheme with the FOTT estimate for the mixed wired/wireless networks. When this packet loss differentiation algorithm incorporates with TCP congestion control so as to trigger it only for the congestion losses, the performance of TCP would be drastically enhanced depending on the wireless bit error rate.

As shown in Fig. 2, wireless losses due to the deterioration of link quality can occur almost at the same rate regardless of the congestion status. Therefore, to increase packet loss differentiation accuracy, we need to detect wireless losses occurred within the congested period as well as to determine accurately whether the network is congested.

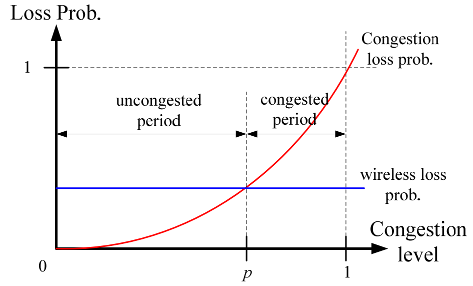

Fig. 2 Occurrences of Congestion Loss and Wireless Loss.

In the paper, we adopt the statistical characteristics of FOTT measured in the congested period to identify the wireless losses occurred within this congested period. We describe the proposed FOTT estimation scheme with clock offset removal and packet loss differentiation scheme using the estimate of FOTT in the following sections.

3.1 FOTT estimation

Before proceeding to the proposed algorithms in detail, we define the relations between the time instances and the related delays during the packet delivery to better understand the basic ideas of the removal algorithms of clock offset and the corresponding effects of clock offset. in this section, we try to keep the consistency in notations for clocks, timestamps and delays with previous works [15][16].

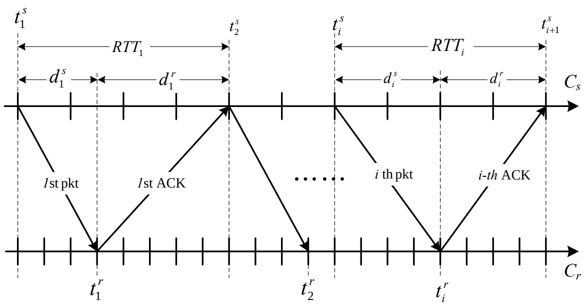

Fig. 3 Relation of time information in packet transfer.

- C_s, C_r : clocks of the sender and the receiver
- t_i^r : i-th packet arrival time at the receiver
- t_i^s : i-th packet sending time at the sender
- d_i^r : one-way delay over the reverse path for i-th packet
- d_i^s : one-way delay over the forward path for i-th packet
- RTTi : RTT for i-th packet measured at the sender
- θ : clock offset between two clocks

We assume in the paper that the receiver clock progresses faster than the sender clock, i.e. $C_r = C_s + \theta$, for simplicity and without loss of generality. As shown in Fig. 3, i-th packet arrival time measured at the receiver and i-th packet sending time at the sender is derived as follows.

$$t_i^r = t_i^s + d_i^s + \theta . \quad (6)$$

$$t_{i+1}^s = t_i^r + d_i^r - \theta . \quad (7)$$

Since the RTT value for i-th packet can be expressed by $t_{i+1}{}^s = t_i^s + RTT_i$, clock offset is formulated as follows.

$$\theta = t_i^r - t_i^s + d_i^r - RTT_i . \quad (8)$$

We need to know the one-way delay for i-th ACK packet, d_i^r, to calculate clock offset θ. However, this value is hard to know without the clock synchronization between two clocks. In current standard TCP, since first packet of the connection is assumed to experience minimum delay during the connection, network bandwidth is estimated by the RTT value for the packet and used to calculate the slow start threshold, *ssthresh*[23]. Based on this implication, we estimate the clock offset with the RTT value of the first sending packet as in Eq.(9).

$$\theta = t_1^r - t_1^s + d_1^r - RTT_1 = t_1^r - t_1^s - 0.5 \times RTT_1 . \quad (9)$$

Since the first packet of the connection can be assumed to experience the minimum delays both on the forward and the reverse paths, we take the one-way delay of the first packet on the reverse path as the half of RTT value for the first packet, RTT_1, of the connection. Therefore the clock offset is estimated using the arrival time of the first ACK packet at the sender as in Eq.(9). With the estimate of the clock offset θ, the sending time of i-th packet at the sender t_i^s which is echoed by the ACK packet to the sender and the receiving time of i-th packet at the receiver t_i^r, we can calculate forward one-way delay every time ACK packet arrives at the sender as in Eq.(10)

$$d_i^s = t_i^r - t_i^s - \theta . \quad (10)$$

As you can see in Eq.(10), forward one-way delay can be derived only with the measurable data at the sender side. It is important that this does not cause any problem for implementing the algorithm within the TCP protocol in the real network since it requires sender only modifications.

3.2 Packet Loss Differentiation with FOTT

We have presented simple FOTT estimation algorithm conducted at the sender in the previous section. In the section, we present how our algorithm classifies packet losses into congestion losses and wireless losses due to the problem of link quality using the variation of FOTT values. When the packet losses are determined to be a wireless losses, it is required that congestion window be kept as it is, the lost packets be retransmitted, and thereby the transmission rate should not be reduced unnecessarily.
The basic concept of our algorithm is that the forward one-way delay will increase as packets are lost by the network congestion and the average FOTT of the congested period will be higher than that of the non-congested period. Moon *et. al.*[16] studied the correlation between the packet loss and the corresponding delay. She proposed the concept of loss-conditioned average delay to quantify the correlation between delay and loss because it is not possible to calculate the average delay of packets conditioned on their own loss.

$$E[d_i^s \mid l_{i-j}=1] = \sum_{k\in P} d_k^s / |P| \quad , \text{where } P=\{k: l_{k-j}=1 \text{ and } l_k=0\} \qquad (10)$$

Where l_i indicates a loss and l_i = 1 means that i-th packet is lost, j denotes a *lag* and l_{i-j} indicates whether the packet received j packets before is lost or not, and therefore Eq.(10) means the average packet delay conditioned on a loss occurring at a time lag j packets in the past.
From the simulation results in [16], it is shown that the loss-conditioned average delay has higher value near lag 0. It means that packets entering the queue just before an overflow experience a large queuing delay. Similarly, those packets successfully entering the buffer soon after an overflow event will also see a large queuing delay, since the queue is still nearly full. Based on this observation, we propose to differentiate the loss type by comparing average FOTT over non-congested period with average FOTT over packet loss period.

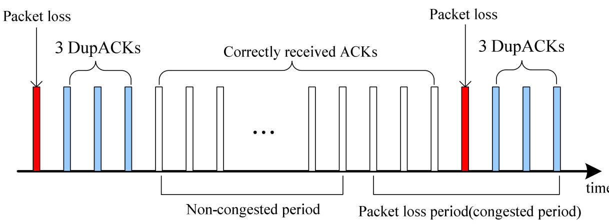

Fig. 4 Non-congested period and packet loss period for average FOTT calculation.

Average FOTT during non-congested period can be calculated at the moment of packet loss detection as follows.

$$\hat{d}_i^{s,uc} = E[d_i^s \mid l_{i-j}=1, l_{i+3}=1] = \sum_{k\in P} d_k^s / |P| \quad , \text{where } P=\{k: l_{k-j}=1, l_k=0 \text{ and } l_{k+3}=1\} \qquad (11)$$

Eq.(11) is calculated for the packets from the first packet which was correctly received after the previous packet loss event to the packet received 3 packets before the current packet loss event. On the other hand, the average FOTT over packet loss period is calculated with the 3 duplicate ACK packets and 3 correctly received packets before the first duplicate ACK packet. In other words, 3 ACK packets just before and just after the packet loss are used for the average FOTT calculation during the packet loss period as in Eq.(12).

$$\hat{d}_i^{s,c} = E[d_i^s \mid l_i=1] = \sum_{k\in P} d_{i+k}^s / |P| \quad , \text{where } P=\{-3<k<3: l_{i+k}=0, l_i=1\} \qquad (12)$$

If the packet losses are occurred due to the network congestion, average FOTT of the packet loss period calculated by Eq.(12) will be higher than that of non-congested period. This idea is supported by the fact that the autocorrelation of loss conditioned average delay is turned out to have peak value when a lag j is near 0[16], which means that average delay of the packet loss period would be higher than that of the period before the loss event.
As described so far, we can infer network congestion state from the average one-way delay at the moment of packet loss event. However, we also have to consider that it is possible for wireless losses to occur even in the congestion state as mentioned in Sec. 3. To this end, we additionally adopt statistical characteristics of average FOTT in the packet loss period as a second measure for decision. It is known that TCP global synchronization happens in the network during the congestion period because each sender will reduce its transmission rate at the same time when packet losses occur. Because of this TCP synchronization, it is expected that the network status will be similar at every packet loss event. Therefore, the average FOTTs in the congestion period can have some relation with the function of their average $\bar{d}_{mean}^{s,c}$ and standard deviation $\bar{d}_{dev}^{s,c}$.
Our proposed packet loss differentiation algorithm can be implemented using the clock offset removal scheme for accurate FOTT estimation and the packet loss differentiation scheme with two criteria as in Table 1.

Table 1: Pseudo-code for the proposed loss differentiation algorithm

INIT $\hat{d}_i^{s,uc} = \hat{d}_i^{s,c} = 0$
$\theta = t_1^r - t_1^s - 0.5 \times RTT_1$
while($i-$th ACK received)
  $d_i^s = t_i^r - t_i^s - \theta$
  update $\hat{d}_i^{s,uc}$
when(3 DupACKs detected)
  update $\hat{d}_i^{s,c}$
  if $(\hat{d}_i^{s,c} / \hat{d}_i^{s,uc} > \delta)$
    if $\{\hat{d}_i^{s,c} > (\bar{d}_{mean}^{s,c} - \bar{d}_{dev}^{s,c})\}$
      it's congestion loss
    else
      it's wireless loss
  else
    it's wireless loss

In Table 1, we assumed that the network is in congestion state when $\hat{d}_i^{s,c} / \hat{d}_i^{s,uc} > \delta$, and from the simulation results in Sec. 4 best performance is shown when $\delta = 1.1$.

4. Performance Evaluation

In this section, we present the simulation model and performance metrics that are used for comparison and the simulation results of proposed LDA scheme conducted by NS-2 simulator[28].

4.1 Simulation Model

All the simulations are carried out with the network model in Fig. 5.

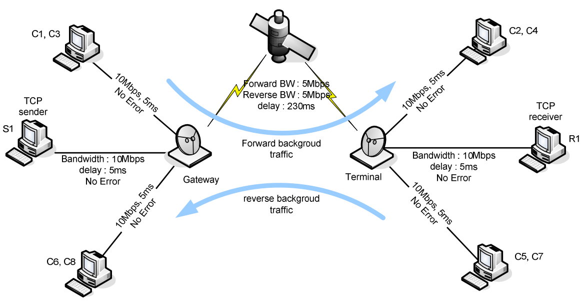

Fig. 5 Simulation Topology.

We have two TCP connections(C1-C2, C3-C4) to make forward cross traffic and another two TCP connections(C5-C6, C7-C8) to make reverse cross traffic on the network. To make the environment more realistic, each connection starts at the different time. C1, C5, C3, C7 start from 0 second with 10 second interval, respectively, and finally S1 starts at 40 seconds when all other connections are at the steady state.

4.2 Simulation Results

We accomplished two different simulations to evaluate the proposed algorithm. First simulation is to verify the estimation accuracy of initial clock offset and second simulation is to evaluate the performance of the packet loss differentiation algorithm through the comparison with some of the previous algorithms.

4.2.1 Initial Clock Offset Estimation Accuracy

In the proposed algorithm, to calculate the initial clock offset, we assumed that the reverse one-way delay for the first packet of the connection would be half of the first RTT value. In this section, we want to verify the estimation accuracy of the initial clock offset in the various network conditions and several different initial clock offset involved.

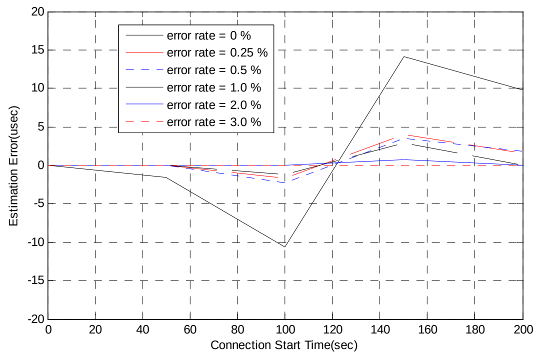

Fig. 6 Initial Offset Estimation Accuracy according to the different connection start times.

For this verification, we make the sender and the receiver clocks have random initial offset value between 10ms ~ 500ms. We conducted several simulations for this with different connection start times and with the different wireless error rates between 0%~3% to see the effect of network conditions to our estimation method. From the simulation results as in the Fig. 6, when the error rate is low and the connection starts at the high network traffic load, it is shown that the estimation error is rather high. Let us see the first case in which the error rate is 0 % and the connection start time is between 100sec and 150sec. The actual estimation value of the first case is around 10 usec, and this is far less than the usual clock granularity which means the estimation error value of our method does not make any inaccuracy in the operation because it would be rounded up to higher order in the real measurement. This result can also imply the accuracy of our forward one-way delay estimation and that is, the accuracy of our forward one-way delay estimation will be kept within the order of around 10 microseconds.

4.2.2 Packet Loss Differentiation Accuracy

The main standard of LDAs is the accuracy in distinguishing wireless losses and congestion losses. As the accuracy is higher, the performance of TCP will be improved much when it comes with the LDA scheme. To evaluate the detection accuracy performance for the congestion loss and wireless loss, we carried out our simulations with the 3 different conditions such as congestion loss only case, wireless loss only case and mixed loss case. We adopted performance metrics for each case as in Eq.(13)

$$A_c = \frac{N_c}{D_c},\quad A_w = \frac{N_w}{D_w},\quad A_t = \frac{A_c + A_w}{2}. \qquad (13)$$

The accuracy of congestion loss detection (A_c) is the fraction of packet losses due to congestion that are correctly diagnosed (N_c) out of total congestion losses happened (D_c). The accuracy of wireless loss detection (A_w) can be similarly calculated. Finally, the accuracy of mixed congestion loss and wireless loss (A_t) is calculated as the average of the two accuracies.

Let us first make a selection for delta (δ) which is the first measure for the congestion condition in our algorithm. We conducted the same test with several different delta values from 0.8 to 1.2 as in Fig. 7.

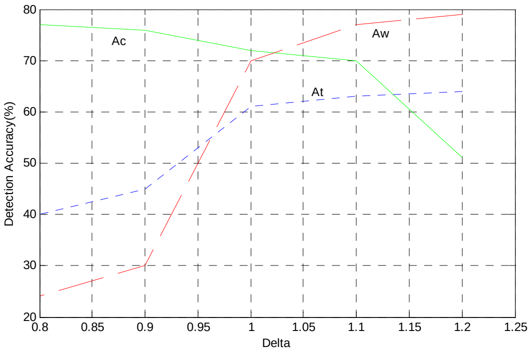

Fig. 7 Impact of delta(δ) on the performance of proposed LDA.

From the result in Fig. 7, it suggests that δ = 1.1 have the most reasonable performance because at this point there is a drastic decrease in A_c and not much increases in A_w and A_t. Therefore, all the simulations in the paper use this value for the delta.

To evaluate our loss differentiation algorithm, we use 3 pervious works for the comparisons which are RELDS, NCPLD and LEA_EQ. In Fig. 8, our algorithm has a lowest accuracy for congestion losses and highest accuracy for wireless losses. Even though our algorithm has the lowest accuracy for the congestion losses, the accuracy level is around 80% which is acceptable. On the other hand, our algorithm has the highest accuracy for the wireless losses that means we can avoid unnecessary performance degradation by using this algorithm with TCP congestion control. It is noted that some of the previous algorithms tends to regard most of the packet losses as the congestion losses

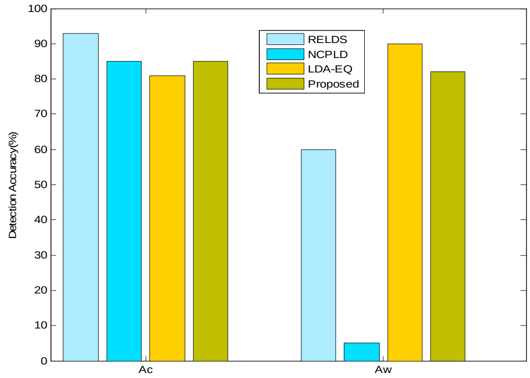

Fig. 8 Accuracies of Congestion(A_c) and Wireless(A_w) losses.

We evaluate our algorithm in the situation of mixed congestion and wireless losses. To make this environment, we start all the connections in Fig. 5 to generate bi-directional cross traffic and put error rate of 0.25% on the wireless link. From the simulation results as in Fig. 9, it is shown that our algorithm has the highest A_t performance thanks to high accuracies of both A_c and A_w. As expected from A_c and A_w performances in Fig. 8, NCPLD and LDA_EQ still show their tendency for the congestion losses and thus they have around 50% accuracy for the mixed congestion and wireless losses.

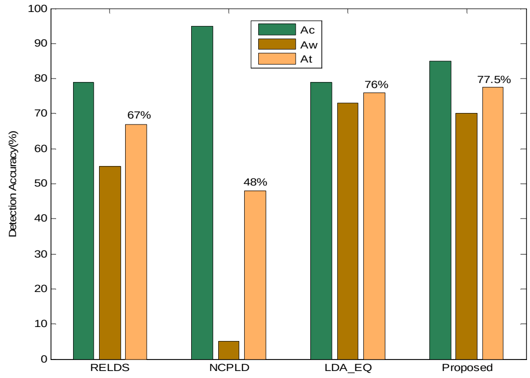

Fig. 9 Accuracy of mixed congestion/wireless losses(A_t).

5. Conclusion

In this paper for the performance enhancement of TCP protocol in the mixed wired/wireless network, packet loss differentiation algorithm is proposed which distinguishes the nature of losses and help TCP decide congestion control action accordingly. Standard TCP protocol takes packet losses as an indication of network congestion so that it reduces its congestion window into half. However, packet losses can be more likely induced by wireless bit error in the wireless network and the performance of TCP in such a network could be unnecessarily degraded by the lack of the ability to identify the cause of the losses. To reduce the effect of the problem, we propose the accurate online loss differentiation algorithm making use of the one-way delay on the forward path. Our contributions are in twofold. First, we introduce the detection and removal of the clock offset existing between sender and receiver clocks only with the information available at the sender side which make our algorithm easy to deploy, and therefore calculate accurate one-way delay on the forward path. Second, we consider the possibility that wireless loss would occur even in the situation of network congestion, and to the best of our knowledge it is the only algorithm that takes this probability into account. We propose two-phase loss differentiation algorithm that uses the ratio of the average one-way delay in the lossless period and lossy period to decide the existence of congestion in the first phase, and the statistical property of the one-way delay for the congestion loss in the second phase. We conducted extensive simulations to evaluate our algorithm. First of all, we have verified the accuracy of the initial offset estimation which is the essential part for our entire algorithm. Even though we have intuitive aspect in our initial offset estimation method, the result showed that our estimation algorithm can estimate initial offset value added randomly with the accuracy to the level of around 10 usec in several network conditions. This also implies that the forward one-way delay can be estimated accurately. We have also evaluated the detection accuracies for each type of losses. The results show that our proposed algorithm has the reasonable performance both for the congestion and wireless losses and thus provides the highest accuracy for the mixed congestion and wireless losses comparing to the 3 previous works. From the simulation results, it is turned out that our proposed algorithm has rather low accuracy for the congestion losses and this would cause more congestion in the network by injecting more packets to the already congested network. Therefore for the further work, this aspect has to be enhanced to keep the friendliness with the other TCP variants.

Acknowledgment

This work was supported by the IT R&D program of KCC[2009-S-039-02, Development of Satellite Return Link Access Core Technology for High Efficient Transmission].

References

[1] S. Biaz and N. Vaidya, “Discriminating congestion losses from wireless losses using interarrival times at the receiver,” in Proc. IEEE Symp. Application Specific Systems and Software Engineering and Technology, pp. 10-17, Richardson, TX, Mar. 1999.

[2] N. Samaraweera, “Non-congestion packet loss detection for TCP error recovery using wireless links," Int. Elect. Eng. Proc. Commun., vol. 146, no. 4, pp. 222-230, Aug. 1999.

[3] Y. Tobe, Y. Tamura, A. Molano, S. Ghosh, and H. Tokuda, "Achieving moderate fairness for UDP flows by path-status classification," in Proc. 25th Annu. IEEE Conf. Local Computer Networks (LCN 2000), pp. 252-261, Tampa, FL, Nov. 2000.

[4] S. Biaz and N. Vaidya, "Distinguishing congestion losses from wireless transmission losses: a negative result," in Proc. 7th Int. Conf. Computer Communications and Networks, pp. 722-731, Lafayette, LA, Oct. 1998.

[5] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," IEEE/ACM Trans. Networking, vol. 5, pp. 756-769, Dec. 1997.

[6] S. Biaz and N. H. Vaidya, "De-randomizing congestion losses to improve TCP performance over wired-wireless networks," ACM/IEEE Trans. Networking, vol. 13, no 3, pp. 596-608, 2005.

[7] K. Xu, Y. Tian, N. Ansari, "Improving TCP performance in integrated wireless communications networks," Computer Networks, vol. 47 no. 2, pp. 219-237, Feb. 2005.

[8] C. Parsa, J. J. Garcia-Luna-Aceves, "Improving TCP congestion control over Internets with heterogeneous transmission media," in Proc. IEEE Int. Conf. Network Protocols(ICNP99), pp. 213-221, Toronto, Canada, Oct. 1999.

[9] V. Jacobson, R. Braden, and D. Borman, "TCP Extensions for High Performance," IETF RFC 1323, May 1992.

[10] C. P. Fu and S. C. Liew, "TCP veno: TCP enhancement for transmission over wireless access networks," IEEE J. Select. Areas Commun., vol. 21, no. 2, pp. 216-228, Feb. 2003.

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

[12] S. Biaz and N. H. Vaidya, "Is the round-trip time correlated with the number of packets in flight?," in Proc. 3rd ACM SIGCOMM Conf. Internet Measurement, pp. 273‐278, Karlsruhe, Germany, Aug. 2003.

[13] E. H. K. Wu and M. Z. Chen, "JTCP: jitter based TCP for heterogeneous wireless networks," IEEE J. Select. Areas Commun., vol. 22, no. 4, pp. 757-766, May 2004.

[14] 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," in Proc. Int. Computer Software and Applications Conf., 2009.
[15] V. Paxson, "End-to-End Internet Packet Dynamics," IEEE/ACM Trans. Networking, vol. 7, no. 3, pp. 277-292, Jun. 1999.
[16] S. B. Moon, "Measurement and Analysis of End-to-End Delay and Loss in the Internet" Ph.D dissertation, University of Massachusetts Amherst, Massachusetts, January 2000.
[17] J. Bi, Q. Wu, and Z. Li, "On estimating clock skew for one-way measurements," Computer Communications, vol. 29, no. 8, pp. 1213-225, May 2006.
[18] H. Khlifi and J. C. Gréegoirea, "Low-complexity offline and online clock skew estimation and removal," Computer Networks, vol. 50, no. 11, pp. 1872-884, Aug. 2006.
[19] M. C. Weigle, K. Jeay, and D. Smith, "Delay-based early congestion detection and adaptation in TCP : impact on web performance," Computer Communications, vol. 28, no. 8, pp. 837-850, May 2005.
[20] 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.
[21] G. Yang, R. Wang, M. Gerla, "TCPW bulk repeat," Comput. Commun., vol 28, no 5, pp. 507-518, 2005.
[22] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP Westwood: end-to-end congestion control for wired/wireless networks," Wireless Networks, Kluwer Academic Publisher, vol. 8, pp. 467-479, 2002.
[23] J. C. Hoe, "Improving the Start-up Behavior of A Congestion Control Scheme for TCP," in Proc. ACM SIGCOMM, pp. 270-280, 1996.
[24] A. Gurtov and S. Floyd, "Modeling wireless links for transport protocols," ACM SIGCOMM Comput. Commun. Rev., vol. 34, no. 2, pp. 85-96, 2004.
[25] J. Postel, Transmission control protocol, RFC 793, Sept. 1981.
[26] 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, pp.1465-1480, Oct. 1995.
[27] C. P. Fu, S. C. Liew, "TCP Veno: TCP Enhancement for Transmission Over Wireless Access Networks" IEEE J. Select. Areas Commun., vol. 21, no. 2, pp. 216-228, Feb. 2003.
[28] ns-2 Network Simulator (Ver 2). LBL. [Online]. Available : http://www.isi.edu/nsnam/ns
[29] H. F. Hsiao, A. Chindapol, J. Ritcey, Y. C. Chen, and J. N. Hwang, "A new multimedia packet loss classification algorithm for congestion control over wired/wireless channels," IEEE ICASSP, vol.2, pp.1105-1108, Mar. 2005.

Minsu Shin received the B.E. and M.E. degrees, from FKorea Aerospace Univ. in 1998 and 2000, respectively. Since he joined the Electronics and Telecommunications Research Institute (ETRI) in 2000, he has worked for satellite communication systems and broadcasting systems until now. Now his research interests are satellite communication network design, wireless resource management, wireless TCP and cross-layer enhancement.

Mankyu Park received the B.S. and M.S. degrees from the Department of Information and Communication Engineering, Kongju National University, Korea in 1999 and 2001, respectively. He worked as a system programmer at venture companies named Syswill and KDnet from 2001 to 2005, and he was a researcher and full time Ph.D student in Chungnam national university from 2006 to 2009. Now he is a senior researcher in Satellite Broadcasting & Telecommunications Convergence Research Team at ETRI.

Byungchul Kim received the B.S. degree in Electronics engineering from Seoul National University and M.S. and Ph.D degrees in electronic engineering from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 1988, 1990 and 1996. He is currently a professor at the Department of Information and Communication Engineering of Chungnam National University, Korea since 1999. Also, from 1993 to 1999, he worked as a Research Engineer at the Samsung Electronics. His research interests include computer networks, wireless internet, sensor networks and mobile communications.

Jaeyong Lee received the B.S. degree in Electronics engineering from Seoul National University and M.S. and Ph.D degrees in electronic engineering from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 1988, 1990 and 1995. He is currently a professor at the Department of Information and Communication Engineering of Chungnam National University, Korea since 1995. Also, from 1990 to 1995, he worked as a research engineer at the Digicom Institute of Information and Communications. His research interests include Internet protocols, traffic control, performance analysis and mobile internet.

Deockgil Oh received his BS degree in Electronics Engineering Department from Seoul National University (SNU) in 1980. He received the MS and Ph.D degrees in Electronics Engineering from SNU, Korea, in 1984 and 1996, respectively. He joined ETRI in 1982, where he is currently working as a director in Dept. of Satellite Broadcasting and Communication Convergence. His research interests include wireless access technology, mobile communication and broadcasting system and future generation satellite broadcasting architectures.