Journal of Institute of Control, Robotics and Systems (2010) 16(8): 816-822
DOI 10.5302/J.ICROS.2010.16.8.816 ISSN 1976-5622

무선 망에서의 연속적 ECN을 이용한 TCP 성능 개선 방법

Improving TCP Performance for Wireless Networks Based on Successive ECN

변 희 정*
(Heejung Byun)
[1]Suwon University

Abstract: In this paper, we propose an algorithm to improve TCP performance over wireless links. TCP is known to have poor performance over wireless links because TCP has no mechanism to differentiate congestion loss from wireless loss, and treats all losses as congestive. We present a simple method to determine the cause of packet loss using the successive ECN. In addition, we present an algorithm to control the congestion window size based on the estimated queue state in order to guarantee the fairness and high link utilization.

Keywords: wireless TCP, ECN, congestion control

I. INTRODUCTION

TCP (Transmission Control Protocol) is designed for the wired networks where the channel error rates are very low and congestion is the primary cause of packet loss. TCP assumes that every packet loss is an indication of the network congestion and reduces the transmission rate. However, when a wireless link forms a part of a network, the packet losses due to the wireless link error are often more significant than the ones due to the congestion. Since TCP has no mechanism to differentiate these losses from congestion, it treats all losses as congestive by reducing its transmission rate, and therefore TCP shows greatly degraded performance over wireless networks.

Many algorithms to improve wireless TCP performance have been proposed [1-16]. In [1,2], the source is informed of the reason for a packet loss explicitly. The protocols in [3-8] decide whether packet losses are likely to be due to congestion or wireless link errors using the network information. The protocols proposed in [9-11] eliminate the packet loss due to the buffer overflows, so the source only sees the wireless losses. The algorithms in [12,13] estimate the packet rate of the connection by monitoring ACK reception rate. In [14-16], they develop a variant of RCP (Rate Control Protocol) under the max-min criterion and α -fairness. In [15,16], they study the local stability of RCP for a single resource with a large or small buffer. However, there is far less result on methods for wireless TCP to provide fair sharing of the bandwidth and high link utilization. In this paper, we propose an algorithm to identify losses due to congestion and those due to noise on a wireless link based on successive ECN (Explicit Congestion Notification), and propose a congestion window control algorithm to guarantee fair sharing of the bandwidth among active connections.

II. PROPOSED ALGORITHM

1. Preliminaries

* 책임저자(Corresponding Author)
논문접수: 2010. 4. 9., 수정: 2010. 5. 4., 채택확정: 2010. 7. 4.
변희정: 수원대학교 IT대학 정보통신공학과(heejungbyun@suwon.ac.kr)

In our proposal, we assume that ECN is implemented in the network and we consider a TCP network with a single bottleneck link. Let M be the number of active connections routed through the bottleneck link and let τ_i be a RTT (Round Trip Time) of connection i. In addition, τ_{if} and τ_{ib} are the forward /backward delay of connection i. The capacity of outgoing bottleneck link is denoted by C [packet/s].

Assume that there exists a positive integer Δ_i $(1 \le i \le M)$ proportional to τ_i such that

$$\frac{\tau_1}{\Delta_1} = \frac{\tau_2}{\Delta_2} = \cdots = \frac{\tau_{M-1}}{\Delta_{M-1}} = \frac{\tau_M}{\Delta_M} = \tau_s$$

with $\tau_1 < \tau_2 < \cdots < \tau_{M-1} < \tau_M$ where τ_s is a constant.

For a ECN router, we propose that the router marks the packet with the following probability p, where p is a function of the queue length:

$$p(n) = \begin{cases} \max_p \cdot \dfrac{q(n)}{\max_{th}} & if \quad q(n) \le \max_{th} \\ 1 & else \end{cases} \tag{1}$$

where $q(n)$ is the queue length of the bottleneck link at time n, $\max_p$ is the maximum packet marking probability, and $\max_{th}$ is the maximum threshold of the queue length.

Since one-bit congestion indication by the ECN message is insufficient to judge the network state properly, we use a number of ECN messages to estimate the packet marking probability and the queue state of the bottleneck link. That is, sending host i counts the number of received ECN-Echo messages, N_e, in N number of the ACK packets during $[n, n+1)$. Then we define $e_i(n) \equiv N_e / N$. Consider the wired networks where the transmission errors are negligible. Then from the fact the ECN algorithm marks the incoming packet proportionally to the window size of each connection [18], the overall ratio of N_e and N

is equal to the probability that the ECN algorithm randomly marks the packet. Then without loosing much generality, we show

$$e_i(n) = p(n - d_{ib}) \tag{2}$$

where $d_{ib} = \tau_{ib} / \tau_s$.

2. Differentiation of congestion and wireless loss

Out-of-sequence packets are the indication of the packet losses caused by the congestion or the wireless link error. Then the receiver sends the duplicate acknowledgment to the sending host irrespective of the cause of packet loss. When the third duplicate ACK (acknowledgment) arrives, the sending host should determine the cause of the packet loss and execute the proper action based on the result. The proposed method to determine the cause of packet losses consists of two steps: estimation of ECN marking for the lost packet and detection of the cause of packet loss.

First, we describe the method to estimate whether the lost packet was marked by ECN or not. From the result in [4], congestion neither happens nor disappears suddenly and congestion losses are normally preceded and followed by marked packets. Based on this result, we define the set of adjacent ACKs of ACK k, ACK_k^N, as follows:

$$ACK_k^N = \{ACK_{k-3}, ACK_{k-2}, ACK_{k-1}\}$$

where ACK_i is the ACK of i th packet. When a sending host receives the third duplicate ACK, it checks whether any ACK in the adjacent ACKs is an ECN-Echo. Then,

- if any ACK in the set of adjacent ACKs contains an ECN-Echo, we estimate that the lost packet corresponding to the duplicate ACK was marked by ECN
- otherwise, when the set of adjacent ACKs contain no ECN-Echo, we estimate that the lost packet corresponding to the duplicate ACK was not marked by ECN.

Second, the sending host calculates $e_i(n)$ with consideration of the estimated ECN-Echo from the first step. Then the cause of the loss is classified as

- if $e_i(n) \ge \max_p$, then the duplicate ACK is caused by a congestion loss.
- otherwise, the duplicate ACK is caused by a wireless loss.

Fig. 1 shows the error rate in determining the cause of packet losses. The proposed algorithm makes the right estimation in most cases, but it makes mistake when the wireless packet error rate is very high. Therefore, when the wireless packet error rate lies in the acceptable range (0-10%), we can say that (2) is also satisfied in the wireless network. Then from (1)-(2), $e_i \ge \max_p$ implies $q \ge \max_{th}$ and so, we can estimate the queue state.

3. Congestion window control

When the duplicate ACK is caused by a congestion loss, the sending host triggers its congestion control mechanism by halving its congestion window. However, this AIMD (Additive Increase/ Multiplicative Decrease) algorithm leads to the significantly degraded throughput and it does not guarantee the fair bandwidth sharing among connections. In this section, we propose a congestion window control algorithm to guarantee the fair sharing of the available bandwidth and high link utilization in TCP over wireless link networks. We assume that a sending host updates its congestion window size every its RTT (Round-Trip Time). Let w_i denote the congestion window size of the sending host i. Then we introduce the following congestion window control algorithm:

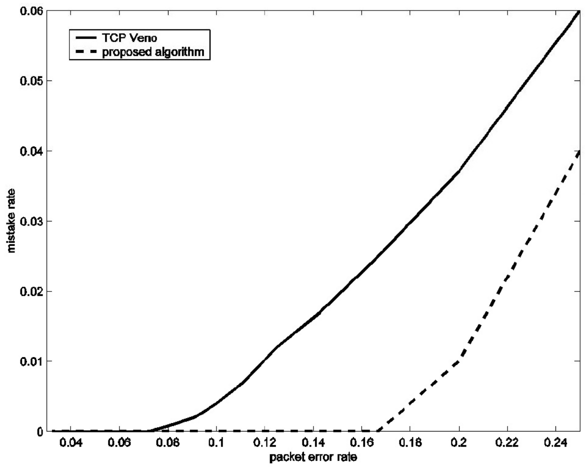

그림 1. 구별 오류율.

Fig. 1. Differentiation error rate.

$$w_i \leftarrow [\alpha(\max_p - e_i)]^+ \cdot RTT_i \tag{3}$$

where α is a control gain to be chosen, RTT_i is the round-trip time of connection i and $[\cdot]^+ = \max(0,\cdot)$. When $\max_p \le e_i$, it means that the queue length of bottleneck link exceeds $\max_{th}$, and so the sending host sets the congestion window size to zero. In contrast when $\max_p > e_i$, the sending host adapts the congestion window size in proportional to $(\max_{th} - q)$.

III. ANALYSIS

1. Network modeling

We assume that the waiting time of a packet at the router is negligible in comparison with the round-trip propagation time. Then the system can be represented by a discrete-time model where τ_s is the duration of a normalized time slot [17].

Let $w_i(n)$ denote the congestion window size of the sending host i at time n. From the assumption that the sending host updates its congestion window size every τ_i, (3) can be rewritten as follows:

$$w_i(n+1) = \begin{cases} [\alpha(\max_p - e_i(n))]^+ \cdot \tau_i & if \quad ((n+1) \bmod \Delta_i) = 0 \\ w_i(n) & else \end{cases} \tag{4}$$

Let $p_e(n)$ denote the stationary process of packet loss ratio at time n on the wireless link. Then, the dynamics of the bottleneck queue is described by the following difference equation:

$$q(n+1) = \left[q(n) + \left(\sum_{i=1}^{M} \frac{w_i(n+1-d_{if})}{\tau_i} - C(1-p_e(n)) \right) \tau_s \right]^+ \tag{5}$$

Originally, the window-based congestion control mechanism allows the sending host to send w_i per its RTT. Since each sender updates the window size independently every its RTT, it is difficult to analyze the asymptotic stability in the heterogeneous network. For the analysis, we approximate the network model based on the sense of average. Specifically, we assume that connection i sends w_i/Δ_i on the average and updates the window size every normalized time slot. Thus, the window dynamics of (4) and queue dynamics of (5) are rewritten by the following equations.

$$\hat{w}_i(n+1) = [\alpha(\max_p - e_i(n))]^+ \cdot \tau_s \quad (6)$$

$$q(n+1) = \left[q(n) + \sum_{i=1}^{M} \hat{w}_i(n+1-d_{if}) - C(1-p_e(n))\tau_s \right]^+ \quad (7)$$

We consider that the wireless packet error rate lies in the acceptable range. Substituting (2) and (6) to (7), we derive the closed-loop dynamics for the network model:

$$q(n+1) =$$
$$\left[q(n) + \alpha \frac{\max_p}{\max_{th}} \sum_{i=1}^{M} (\max_{th} - q(n-\Delta_i))\tau_s - C(1-p_e(n))\tau_s \right]^+ \quad (8)$$

Let

$$x(n) = \frac{q(n)}{q_c} \quad (9)$$

and $k_1 = \varepsilon_a/\varepsilon, k_2 = \varepsilon_b/\varepsilon$, where $\varepsilon_a = \alpha \max_p \tau_s / \max_{th}$, $\varepsilon_b = C\tau_s/q_c$, $\varepsilon = \min(\varepsilon_a, \varepsilon_b)$, and q_c is the total queue capacity. Then (8) is rewritten as:

$$x(n+1) = x(n) + \varepsilon\Phi(x(n), x(n-1), \cdots, x(n-D), p_e(n)) \quad (10)$$

where

$$\Phi(x(n), x(n-1), \cdots, x(n-D), p_e(n))$$
$$= -\left[k_1 \sum_{i=1}^{M} \left(x(n-\Delta_i) - \frac{\max_{th}}{q_c} \right) + k_2(1-p_e(n)) \right]$$

and $D = \max_{\forall i}(\Delta_i)$.

Since $q_c >> 1$ without loosing much generality, we assume that $\varepsilon << 1$ Note that since Φ can take arbitrarily large values but with negligibly small probabilities, (10) represents a slow-in-the-average Markov walk process [21-23]. Let $y(n)$ denote the averaged value of $x(n)$. Applying the asymptotic theory for such processes [21-23], we obtain the following asymptotic approximation:

$$y(n+1) = y(n) - \varepsilon \left(k_1 \sum_{i=1}^{M} \left(y(n-\Delta_i) - \frac{\max_{th}}{q_c} \right) + k_2(1-p_e) \right) \quad (11)$$

where $p_e = E[p_e(n)]$.

2. Steady state and asymptotic stability

Let q_s and $\hat{w}_{is}$ denote the steady state solution of $q(n)$ and $\hat{w}_i(n)$. Note that in the neighborhood of the steady state, we ignore the saturation nonlinearity. Thus from (6)-(9) and (11), if $\max_{th} > C(1-p_e) \cdot \max_{th}/(\alpha M \cdot \max_p)$, the steady states are obtained from

$$q_s = \max_{th} - \frac{C(1-p_e)}{\alpha M} \frac{\max_{th}}{\max_p} \quad (12)$$

$$\hat{w}_{is} = \frac{C\tau_s}{M}(1-p_e) \quad (13)$$

Note that q_s cannot be greater than $\max_{th}$ and can be stabilized at a certain value smaller than $\max_{th}$. Since $\hat{w}_{is}$ is the steady state solution of (6), which represents the approximated window dynamic at every normalized time slot, we obtain the congestion window size of connection i at every RTT by multiplying Δ_i to (13). That is $w_{is} = C\tau_i(1-p_e)/M$ for all i. Then the transmission rate λ_i of connection i at the steady state is the same as $C(1-p_e)/M$. Consequently, the transmission rates of all the connections routed through the same bottleneck link are same as $C(1-p_e)/M$ and this result guarantees the fair sharing of the bandwidth among the connections. Therefore, the fairness index [25] equals to the value of 1. Here, the average link capacity is $C(1-p_e)$. Therefore, if $\max_{th}$ is chosen such that $\max_{th} > C(1-p_e) \cdot \max_{th}/(\alpha M \cdot \max_p)$, the system guarantees the fair sharing of the bottleneck link bandwidth and high link utilization.

Now we investigate the asymptotic stability of the steady state shown in (12). Let $\bar{y}(n) = y(n) - y_s$, where $y_s = q_s/q_c$. Then

$$\bar{y}(n+1) = \bar{y}(n) - \varepsilon_a \sum_{d=0}^{D} l_d \bar{y}(n-d) \quad (14)$$

Let $\bar{Y}(n)$ be the state vector with respect to the queue dynamics of the bottleneck link that is represented by

$$\bar{Y}(n) = [\bar{y}(n) \quad \bar{y}(n-1), \cdots, \bar{y}(n-D)]^T$$

Then, (14) can be written by the following state equation

$$\bar{Y}(n+1) = A\bar{Y}(n) \quad (15)$$

where

$$\mathbf{A} = \begin{bmatrix} 1-\epsilon_a l_0 & -\epsilon_a l_1 & \cdots & -\epsilon_a l_{D-1} & -\epsilon_a l_D \\ 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

For the bottleneck link, the characteristic polynomial of A of (15) is obtained as follows:

$$P(z) = z^{D+1} - z^D + \varepsilon_a \sum_{d=0}^{D} l_d z^{D-d} \quad (16)$$

From [24], $P(z)$ has all zeros within the unit circle and then, the steady state is asymptotically stable if the control gain α satisfies the following relation:

$$0 < \alpha < \frac{2}{M} \frac{\max_{th}}{\max_p \cdot \tau_s} \sin\left(\frac{\pi}{4D+2} \right) \quad (17)$$

Accordingly, all eigenvalues of A are located within the unit circle. Therefore, for the overall system, the steady state is asymptotically stable. Finally, from the conditions of the high utilization and of the stability, we conclude that if the control gain α satisfies the following relation, the system guarantees the high utilization and stability.

$$\frac{C(1-p_e)}{M\cdot\max_p}<\alpha<\frac{2}{M}\frac{\max_{th}}{\max_p\cdot\tau_s}\sin\left(\frac{\pi}{4D+2}\right) \tag{18}$$

3. Steady state with high packet error rate

In this section, we investigate the steady state behavior when the packet error rate is high and the mistake rate is not zero. We define Q, E, E_R and T as

- Q : the number of ACKs with ECN-echo measured during RTT
- E : the number of the lost packets during RTT
- E_R : the number of the lost packets which was ECN marked
- T : the number of the packets transmitted during RTT

We denote the packet marking probability at the steady state as p. From the ECN marking estimation in Section 2.2, we show $p_m = 1-(1-p)^3$ where p_m is the probability that the lost packet is estimated to be ECN marked. Then we can express e_i at the steady state as

$$\begin{aligned} e_{is} &= \frac{Q+E\cdot(1-(1-p)^3)}{T} \\ &= \frac{Q+E_R+E\cdot(1-(1-p)^3)-E_R}{T} \\ &= p+\delta \end{aligned} \tag{19}$$

where

$$\delta=\frac{E(1-(1-p)^3)-E_R}{T}$$

Since $0\le E_R\le E$, we can determine the range of δ, i.e., $|\delta|\le E/T$.

When the packet error rate is high, the mistake rate is not zero and so there is a deviation from e_i in (2) as δ. Because of this deviation, we cannot determine the exact level of the steady states of the window size and the queue length, but only the range of theirs. From (6)-(7) and (19), we obtain the ranges of q_s and $\hat{w}_{is}$ as

$$\begin{aligned} &\Gamma-\zeta\le q_s\le\Gamma+\zeta \\ &\frac{C\tau_s}{M}(1-p_e)-\frac{E}{T}\alpha\tau_s\le\hat{w}_{is}\le\frac{C\tau_s}{M}(1-p_e)+\frac{E}{T}\alpha\tau_s \end{aligned} \tag{20}$$

where

$$\Gamma=\max_{th}-\frac{C(1-p_e)}{\alpha M}\frac{\max_{th}}{\max_p},\quad \zeta=\frac{E}{T}\frac{\max_{th}}{\max_p}.$$

Similarly with Section 3.2, we obtain the range of the steady state of the congestion window size of connection i at every RTT by multiplying Δ_i to (20):

$$\frac{C\tau_i}{M}(1-p_e)-\frac{E}{T}\alpha\tau_i\le w_{is}\le\frac{C\tau_i}{M}(1-p_e)+\frac{E}{T}\alpha\tau_i \tag{21}$$

The steady state of the transmission rate λ_i of connection i is equal to the congestion window size divided by its round-trip time (w_{is}/RTT_i). We have

$$\frac{C}{M}(1-p_e)-\frac{E}{T}\alpha\le\lambda_{is}\le\frac{C}{M}(1-p_e)+\frac{E}{T}\alpha \tag{22}$$

From (22), we can see that as the packet error rate is larger, the transmission rate varies largely. However, the transmission rates of all the connections are converged to the same range, and we can conclude that the proposed algorithm guarantees the fair sharing even with high packet error rate.

IV. SIMULATION RESULTS

In this section, we describe simulation results of the proposed algorithm. We compare our proposed algorithm with TCP Veno [5] and RCP [14]. Let's consider a simplified network model shown in Fig. 2. In the network model, there are two routers connected through wireless bottleneck link and the wired network is a LAN speed network. We use the following network parameters: The wireless link capacity C is 1500 [packet/s], $\max_p$ is 1, and $\max_{th}$ is 100 packets.

The number of TCP connection M is 3 and the minimum RTT, τ_i, of each TCP connection is 10, 20, and 30ms respectively. The packet size is 512 bytes and we choose $\alpha=600$ according to (18).

Fig. 3 shows the results of TCP Veno [5] in terms of the window size, queue length, throughput, and link utilization for a

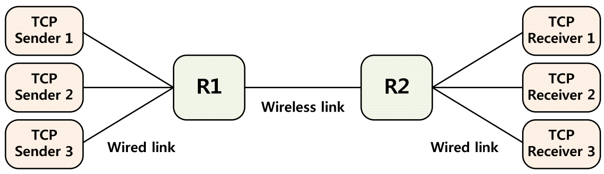

그림 2. 망 모델.

Fig. 2. Network model.

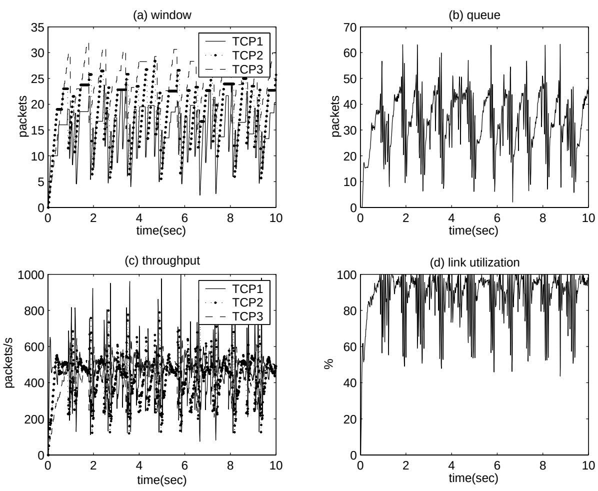

그림 3. TCP Veno [5]: (a) 혼잡 윈도우 크기 (b) 버퍼 길이 (c) 처리율 (d) 링크 효율.

Fig. 3. TCP Veno [5]: (a) Congestion window size (b) Queue length (c) Throughput (d) Link utilization.

wireless BER (Bit Error Rate) of 1×10^{-5}, which corresponds to the PER (Packet Error Rate) of 4%. As shown in Fig. 3, there is an excessive oscillation in the window size and queue length, though this algorithm employs the modified Reno that reduces the sending rate less aggressively when random loss is detected. In addition, the fair sharing of the bandwidth and the high link utilization are not achieved for a considerable period.

Fig. 4 shows the results of RCP [14]. As shown in Fig. 4, the link is fully utilized but TCP 1 occupies most of the link capacity. Therefore the fair sharing of bandwidth is not achieved among TCP connections. Moreover, the queue length is smaller than $\max_{th}$ but fluctuates rapidly.

Fig. 5 shows the results of the proposed algorithm. In Fig. 5(a), the congestion window size controlled by the proposed algorithm shows a superior behavior, i.e., the convergence speed is much faster and there's no excessive oscillation. In addition, the window size of the connection with larger RTT is larger than that of connection with smaller RTT. This is due to the proper adaptation of the window size in order to obtain fair bandwidth allocation. As shown in Fig. 5(b), the queue length converges around the steady state smaller than $\max_{th}$. The throughput has been converged to the fair share and the high link utilization is achieved.

To show the fairness in the throughput, we measure the fairness index [25], I(n), of the throughput of all connections, which is given by

$$I(n) = \frac{(\sum_{i=1}^{M} \lambda_i(n))^2}{M \sum_{i=1}^{M} \lambda_i^2(n)}$$

The value of this index is between 0 and 1. If the throughput of

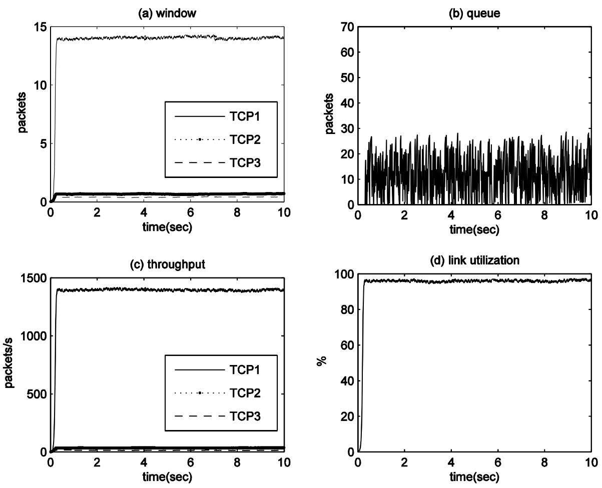

그림 4. RCP [14]: (a) 혼잡 윈도우 크기 (b) 버퍼 길이 (c) 처리율 (d) 링크 효율.

Fig. 4. RCP [14]: (a) Congestion window size (b) Queue Length (c) Throughput (d) Link utilization.

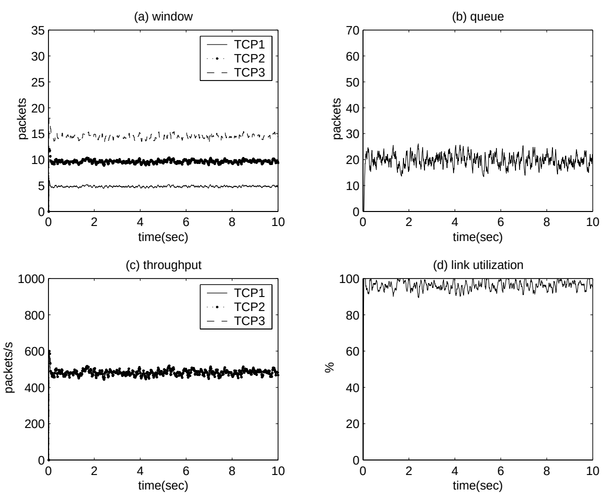

그림 5. 제안 알고리즘: (a) 혼잡 윈도우 크기 (b) 버퍼 길이 (c) 처리율 (d) 링크 효율.

Fig. 5. Proposed algorithm: (a) Congestion window size (b) Queue length (c) Throughput (d) Link utilization.

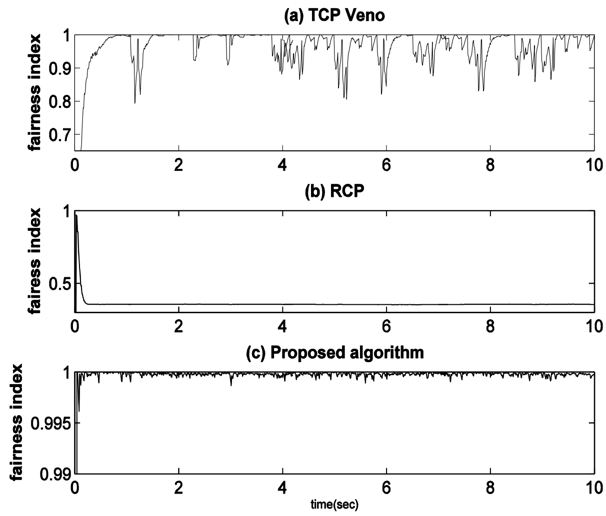

그림 6. 공정성 지수: (a) TCP Veno (b) RCP (c) 제안 알고리즘.

Fig. 6. Fairness index: (a) TCP Veno (b) RCP (c) Proposed algorithm.

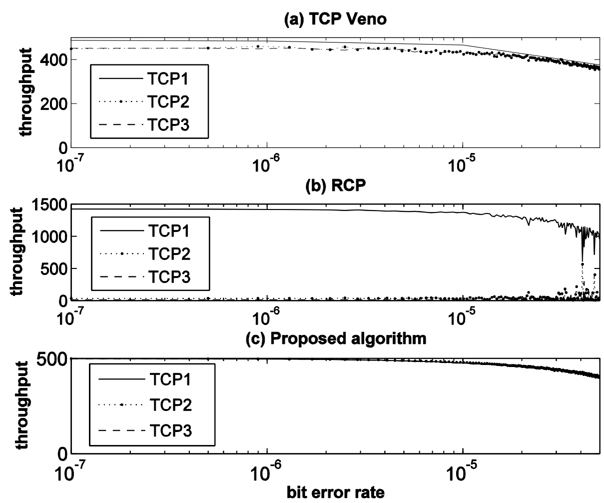

그림 7. 처리율 vs 비트 오류율: (a) TCP Veno (b) RCP (c) 제안 알고리즘.

Fig. 7. Throughput vs BER: (a) TCP Veno (b) RCP (c) Proposed algorithm.

each connection is exactly the same each other, this index will be the value of 1. The smaller the index, the larger the unfairness in throughput. Fig. 6 shows the fairness index. Compared with RCP, the fairness index of TCP Veno and that of the proposed algorithm tend to converge to 1, but TCP Veno has the smaller index.

Fig. 7 shows the throughput averaged during 10 second of TCP Veno, RCP, and the proposed algorithm for bit error rate ranging from 1×10^{-7} to 5×10^{-5}. TCP Veno performs reasonably well when the bit error rate is small, but as the bit error rate increases, its performance degrades more quickly than that of the proposed algorithm. The results of RCP shows the similar behavior with TCP Veno but RCP does not achieve the fairness in terms of throughput among TCP connections. However, we find that the proposed algorithm guarantees the fair sharing of the bandwidth for a wide range of bit error rate.

In Fig. 8, we illustrate the performance for the proposed algorithm with high wireless packet loss rate (PER=20%). This figure shows a considerable oscillatory behavior because the mistake rate is not negligible for high PER. However, the queue length still lies around a value smaller than $\max_{th}$.

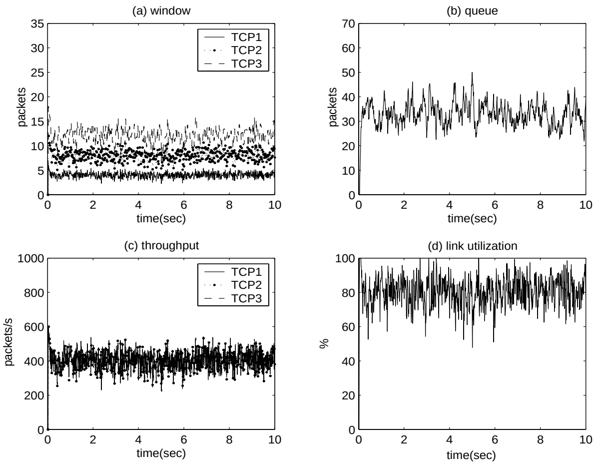

그림 8. 패킷 오류율이 높을 때의 제안 알고리즘 : (a) 혼잡 윈도우 크기 (b) 버퍼 길이 (c) 처리율 (d) 링크 효율.

Fig. 8. Proposed algorithm with high PER: (a) Congestion window size (b) Queue length (c) Throughput (d) Link utilization.

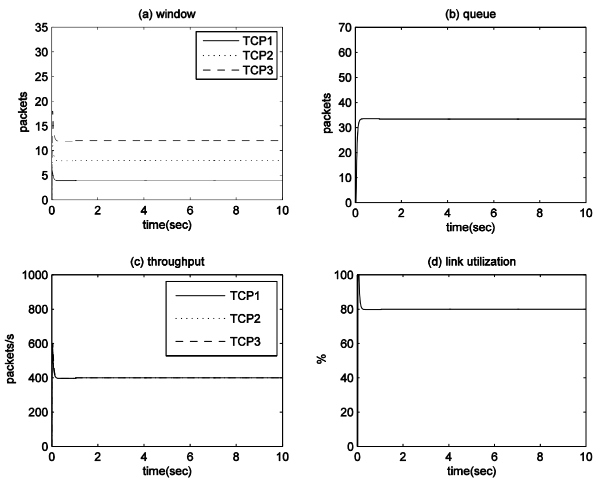

그림 9. 제안 알고리즘 근사화 : (a) 혼잡 윈도우 크기 (b) 버퍼 길이 (c) 처리율 (d) 링크 효율.

Fig. 9. Approximate results for the proposed algorithm with high PER: (a) Congestion window size (b) Queue length (c) Throughput (d) Link utilization.

Fig. 9 shows the approximate results of the proposed algorithm for the high wireless packet loss rate in terms of (20)-(22), i.e., the derived the range of the window, queue and throughput. Compared with the results of Fig. 8, we observe that the behaviors of the proposed algorithm are almost the same as the approximate behaviors.

V. CONCLUSIONS

In this paper, we suggest a new algorithm to determine the cause of packet losses using the ratio of the number of ECN-Echo to ACK. In addition, based on that information, the window-based congestion control algorithm is proposed. We observe significant improvements in the performance over TCP wireless networks in our proposed algorithm.

REFERENCES

[1] O. Shagdar, M. N. Shirazi, and B. Zhang, "Improving ECN-based TCP performance over wireless networks using a homogeneous implementation of EWLN," *International Conference on Telecommunications*, vol. 1, pp. 812-817, 2003.

[2] W. Ding and A. Jamarlipour, "A new explicit loss notification with acknowledgment for wireless TCP," *Personal, Indoor and Mobile Radio Communications*, San Diego, USA, pp. B.65-B.69, 2001.

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

[4] C. Liu and Raj Jain, "Approaches of wireless TCP enhancement and a new proposal based on congestion coherence," *Proc. of the 36th Hawaii International Conference on System Sciences*, pp. 307-316, 2002.

[5] 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, 2003.

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

[7] C. Parsa and J. J. Garcia, "Differentiating congestion vs. random loss: A method for improving TCP performance over wireless links," *Proc. of IEEE Wireless Communications and Networking Conference*, Chicago, USA, vol. 1, pp. 90-93, 2000.

[8] R. Ramani and A. Karandikar, "Explicit congestion notification (ECN) in TCP over wireless network," *IEEE International Conference on Personal Wireless Communications*, Hyderabad, India, pp. 495-499, 2000.

[9] S. Athuraliya, S. H. Low, V. H. Li, and Q. Yin, "REM: active queue management," *IEEE Networks*, vol. 15, no. 3, pp. 48-53, 2001.

[10] M. Gerla, W. Weng, and R. L. Cigno, "Bandwidth feedback control of TCP and real time sources in the Internet," *IEEE GLOBECOM*, San Francisco, USA, vol. 1, pp. 561-565, 2000.

[11] M. Gerla, W. Weng, and R. L. Cigno, "BA-TCP: a bandwidth aware TCP for satellite networks," *Proc. of Computer Communications and Networks*, Boston, USA, pp. 204-207, 1999.
[12] M. Gerla, M. Y. Sanadidi, R. Wang, and A. Zanella, "TCP Westwood: congestion window control using bandwidth estimation," *IEEE GLOBE*COM, San Antonio, USA, pp. 1698-1702, 2001.
[13] F. Martignon and A. Capone, "TCP with bandwidth estimation over wireless networks," *Proc. Vehicular Technology Conference*, vol. 3, pp. 1422-1426, 2002.
[14] F. Kelly, G. Raina, and T. Voice, "Stability and fairness of explicit congestion control with small buffers," *ACM SIGCOMM Computer Communication Review*, vol. 38, issue 3, pp. 51-62, 2008.
[15] H. Balakrishnan, N. Dukkipati, N. McKewon, and C. Tomlin, "Stability analysis of explicit congestion control protocols," *IEEE Communications Letters*, vol. 11, no. 10, pp. 823-825, 2007.
[16] T. Voice and G. Raina, "Stability analysis of a max-min fair rate control protocol (RCP) in a small buffer regime," *IEEE Trans. Automatic Control*, vol. 54, no. 8, pp. 1908-1913, 2009.
[17] H. Ohsaki and M. Murata, "Steady state analysis of the RED gateway: stability, transient behavior, and parameter setting," *IEICE Trans. Commun.*, vol. E85-B, no. 1, pp. 107-115, 2002.
[18] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Net.,* vol. 1, no. 4, pp. 397-413, 1993.
[19] R. G. Mukhtar, S. V. Hanly, and L. L. H. Andrew, "Efficient Internet traffic delivery over wireless networks," *IEEE Commun. Mag.*, vol. 41, no. 12, pp. 46-53, 2003.
[20] L. Kalampoukas, A. Varma, and K. K. Ramakrishnan, "Explicit window adaptation: a method to enhance TCP performance," *IEEE/ACM Trans. Net.*, vol. 10, no. 3, pp. 338-350, 1991.
[21] J.-T. Lim, S. M. Meerkov, and T. Zeng, "Simplified description of slow-in-the-average Markov walks," *J. Math. analys. and applic.*, vol. 158, no. 2, pp. 476-486, 1991.
[22] J.-T. Lim and K. H. Shim, "Asymptotic performance evaluation of Token Passing Networks," *IEEE Trans. Industrial Electronics*, vol. 40, no. 3, pp. 384-385, 1993.
[23] K. H. Shim and J.-T. Lim, "Performance analysis and design of Token-passing networks with two message priorities," *IEE Proc. Commun.*, vol. 44, no. 1, pp. 11-16, 1997.
[24] K.-H. Shim and J.-T. Lim, "Extreme-point Robust Stability of a class of discrete-time polynomials," *Electronics Letters*, vol. 32, no. 15, pp. 1421-1422, 1996.
[25] D. Chiu and R. Jain, "Analysis of the increase/decrease algorithms for congestion avoidance in computer networks," *Journal of computer networks and ISDN*, vol. 17, no. 1, pp. 1-14, 1989.

변 희 정

1999년 숭실대학교 정보통신공학과 학사. 2001년 한국과학기술원 전자전산학과 석사. 2005년 한국과학기술원 전자전산학과 박사. 2010년 삼성종합기술원, 삼성전자. 2010년~현재 수원대학교 IT대학. 관심분야는 네트워크 제어.