A Window-Based Congestion Control Algorithm for Wireless TCP in Heterogeneous Networks

Hee-Jung Byun* and Jong-Tae Lim*

*Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea

(Tel: +82-42-869-3441; Fax: +82-42-869-3410; Email:jtlim@stcon.kaist.ac.kr)

Abstract: This paper describes a feedback-based congestion control algorithm to improve TCP performance over wireless network. In this paper, we adjust the packet marking probability at the router for Max-Min fair sharing of the bandwidth and full utilization of the link. Using the successive ECN (Explicit Congestion Notification), the proposed algorithm regulates the window size to avoid the congestion and sees the packet loss only due to the wireless link error. Based on the asymptotic analysis, it is shown that the proposed algorithm guarantees the QoS of the wireless TCP. The effectiveness of the proposed algorithm is demonstrated by simulations.

Keywords: wireless TCP, ECN, congestion control

1. Introduction

An important assumption of TCP (Transmission Control Protocol) is that every packet loss is an indication of network congestion and the source should reduce its 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. Therefore a TCP sender misunderstands the packet loss from this link error as the packet loss due to the congestion in the network, and reduces the congestion window size unnecessarily. Recently, several algorithms have been proposed to improve TCP's performance in the wireless networks [1]-[8]. In [2], the source is informed that the losses are due to the wireless link and so it is not necessary to reduce its window size after retransmission. The protocols in [3]-[5] decide whether packet losses are likely to be due to congestion or wireless link errors using the network congestion information. The protocols proposed in [6]-[8] eliminate the packet loss due to the buffer overflows, so they only see the wireless losses. However, there is no result on Max-Min fair sharing of the bandwidth for the heterogeneous network case.
In this paper, we propose an algorithm to improve TCP's performance over wireless link in the heterogeneous network. We adjust the packet marking probability at the router for Max-Min bandwidth sharing and full utilization of the link. Using the successive ECN [9], we develop a window control algorithm to eliminate the congestion in the wired network. We consider the case of data transfer from fixed hosts to mobile hosts.

2. Proposed Algorithm

2.1. Network Modeling

In our proposal, we assume that ECN is implemented in the wired network. At first, we define the notion of the locally/remotely bottlenecked connections. Let Ω_l be the set of all connections that are routed through link l. Since the connections pass through the multiple bottleneck links, there will be some connections from set Ω_l which are bottlenecked elsewhere in the network. Let M_l be the set of remotely bottlenecked connections that are routed through link l and whose window sizes are limited by the routers other than the router under consideration. The rest of the connection set S_l where $S_l = \Omega_l - M_l$ are locally bottlenecked connection set at link l. Without loosing much generality, we assume that there are at least one link with $S_l = \Omega_l$.
The network model is shown in Fig. 1. Let M be the number of active connections routed through the bottleneck link and let τ_i be a minimum RTT of connection i ($1 \le i \le M$), which is obtained when the network is not congested. In addition, τ_{if} and τ_{ib} are the forward/backward delay of connection i. The capacity of outgoing link which is the bottleneck of the network is denoted by C [packet/s].

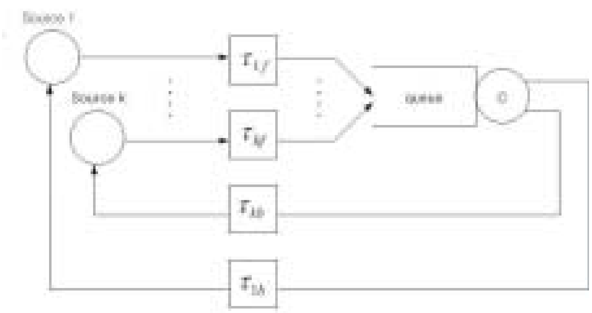

Fig. 1. Network model

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

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

where τ_s is a constant. Provided that the propagation delay is dominant over the waiting time of a packet at the router, the system can be represented by a discrete-time model where τ_s is the duration of a time slot [10]. Let $w_i(n)$ denote the window size of the sending host i at time n, and

This paper is supported by Institute of Information Technology Assessment under the fundamental research fund (sponsored by the Ministry of Information and Communication).

$q_l(n)$ denote the queue length at router l at time n. Let $p_e(n)$ denote the stationary process of packet loss ratio at time n on the wireless link. Then the queue length at router l at time $n+1$ is described by

$$\begin{aligned} q_l(n+1) &= [q_l(n) + (\sum_{i\in S_l} \frac{w_i(n+1-d_{if})}{r_i(n+1-d_{if})} \\ &+ \sum_{i\in M_l} \frac{w_i(n+1-d_{if})}{r_i(n+1-d_{if})} - C\,(1-p_e(n)))\,\tau_s]_0^{q_c} \end{aligned} \quad (1)$$

where $d_{if} = \frac{\tau_{if}}{\tau_s}$, $d_{ib} = \frac{\tau_{ib}}{\tau_s}$, $[\cdot]_0^K = \min[K, \max[0,\cdot]]$, q_c is the total queue capacity and $r_i(n+1) = \tau_i + \sum_{l\in\tilde{L}_i} \frac{q_l(n-d_{ib})}{C}$, where $\tilde{L_i}$ is the subset of links that connection i uses.

2.2. Modified RED Algorithm

Let q^0 be the desired value of the queue length. The modified RED algorithm uses the queue length to determine the packet marking probability $p_l(n)$ at every τ_s as

$$p_l(n) = \left[\frac{\left(\frac{C\tau_s(1-p_e)-R_l(n)}{N_{s_l}} - \alpha_l(q_l(n)-q^0)\right)}{C\tau_s} \right]_0^1 \quad (2)$$

where $p_e = E[p_e(n)]$, $R_l(n)$ is the sum of window in M_l, N_{S_l} is the number of connections in S_l, and α_l is the control gain to be chosen. As shown in (2), $p_l(n)$ is adjusted around $\frac{C\tau_s(1-p_e)-R_{ls}}{N_{s_l}\cdot C\tau_s}$ where R_{ls} is the steady state of $R_l(n)$.

The probability $P_i(n)$ that is the rate of ECN bit received by source i is the sum of all ECN marking rate along its end-to-end path [6], [11]. Therefore the sending hosts that go through more bottleneck links experience larger P_i. On the internet, all the bottlenecked links of a source to the destination path contributes to the aggregate congestion notification that controls the transmission rate. These networks referred SumNet [11] do not achieve Max-Min fairness.

To solve this problem, the desired end-to-end probability $P_i(n)$ should be the minimum value among the packet marking probabilities along the path of connection i because the proposed packet marking probability decreases as the queue length increases. That is the source rate should be controlled by only one link, the most severely bottlenecked link on the end-to-end path. To achieve this, SumNet [11] includes bits with the packet format to communicate the complete congestion information. Each link replaces the current congestion information in the packet if the link's congestion information is greater than the one in the packet. However, in our paper, we propose a new packet marking algorithm that does not require to change the packet format and to relay the complete congestion information. To do this, we introduce a register *strike* which is maintained for each connection and *pre_ecn*. The register *pre_ecn* is the ECN bit from the previous upstream path. From the value of *strike*, it is determined whether ECN field will be marked or not. Moreover, the register *strike* is increased when a ECN bit is set and it is decreased when a ECN bit is reset. The detailed algorithm for the packet marking strategy is described in Table 1. From this procedure, the overall end-to-end probability, i.e., the minimum value among the marking probabilities along the path of the connection is conveyed to the sender as the

Table 1. Packet marking procedure

```
Initialization: strike_i = 0, ∀i
for incoming a packet of connection i   {
   if(ECN bit=1)   pre_ecn = 1;
   else   pre_ecn = 0;
   generation of random R in [0,1]
   if (R > p)
      if (pre_ecn = 1)
         if (strike_i ≥ 0)   {
            ecn = 0;
            strike_i − −; }
         else   {
            ecn=1
            strike_i + +;   }
      else   no marking;
   else   {
      if (strike_i ≥ 0) {
         ecn = 0;
         strike_i − −;   }
      else   {
         ecn = 1;
         strike_i + +;   }
      }
}
```

feedback information. Then, the probability that a packet received with ECN bits set at sending host i becomes the packet marking probability at the link where connection i undergoes the severest congestion.

2.3. Window Control Algorithm

In general, TCP should throttle the window size when a single marked packet is received by the sending host. However, the window updating algorithm of TCP is a class of AIMD (Additive Increase/Multiplicative Decrease) which is the main cause of the unfairness in TCP. Thus the problem is how to update the window size to achieve the fairness once a ECN message is received, but one-bit congestion indication by the ECN message is insufficient to adjust the window size properly [6]. We therefore use a number of ECN messages to obtain the end-to-end packet marking probability. That is, a sending host i counts the number of received ECN messages, N_e, in N number of the ACK packets during $[n, n+1)$. Then we define $e_i(n)$ as

$$e_i(n) = \frac{N_e}{N} \quad (3)$$

From the fact the RED algorithm marks the incoming packet proportionally to the window size of each connection [12] and the packet marking strategy in Section 2.2, the overall ratio of N_e and N is equal to the end-to-end probability that the RED algorithm randomly marks the packet. For example, we assume that the most severely bottlenecked link on the end-to-end path of connection i is link l, i.e., $P_i(n) = p_l(n)$. Then without loosing much generality, according to the algorithm described in Section 2.2, we show

$$e_i(n) = p_l(n - d_{i_b}) \tag{4}$$

Using the measured $e_i(n)$, we introduce a simple congestion window control scheme:

$$w_i(n+1) = \begin{cases} C\tau_i e_i(n) & \text{if (n mod } \Delta_i) = 0 \\ w_i(n) & \text{else} \end{cases} \tag{5}$$

That is, a sending host i is controlled by the packet marking probability of the most severely bottlenecked link encountered on its path.

2.4. Asymptotic Stability

The window-based congestion control mechanism allows the sending host i to send w_i per its RTT. We approximate the network model based on the sense of average. Specifically, we assume that connection i sends w_i/Δ_i per a time slot on the average. Thus, the window dynamic of (5) is rewritten by the following equation.

$$\hat{w}_i(n+1) = \omega^0 e_i(n) \tag{6}$$

where $\omega^0 = C\tau_s$. This represents the evolution of the approximate congestion window size which connection i updates every τ_s. As a result, the approximate dynamics of q_l is given by

$$\begin{aligned} \hat{q}_l(n+1) &= [\hat{q}_l(n) - \alpha_l \sum_{i \in S_l} (\hat{q}_l(n - \Delta_i) - q^0) \\ &- \frac{1}{N_{s_l}} \sum_{i \in S_l} R_l(n - \Delta_i) + \sum_{i \in M_l} \omega^0 p_{l_i}(n - \Delta_i) \\ &- C\tau_s(p_e - p_e(n))]_0^{q_c} \end{aligned} \tag{7}$$

where p_{l_i} is the minimum value of the packet marking probability encountered on the path of connection i in M_l. Since $R_l(n) = \sum_{i \in M_l} \omega^0 p_{l_i}(n)$, (7) can be rewritten:

$$\begin{aligned} \hat{q}_l(n+1) &= [\hat{q}_l(n) - \alpha_l \sum_{d=0}^{\bar{D}_l} l_d(\hat{q}_l(n-d) - q^0) \\ &- C\tau_s(p_e - p_e(n)) - \sum_{i \in M_l} \alpha_{l_i} \{(\hat{q}_{l_i}(n - \Delta_i) - q^0) \\ &- \frac{1}{N_{s_l}} \sum_{d=0}^{\bar{D}_l} l_d(\hat{q}_{l_i}(n-d) - q^0)\}]_0^{q_c} \end{aligned} \tag{8}$$

where $\bar{D}_l = \max_{i \in S_l}(\Delta_i)$.

Let $\hat{q}_{ls}$, $\hat{w}_{is}$ and R_{ls} respectively denote the steady-state solution of $\hat{q}_l(n)$, $\hat{w}_i(n)$ and $\sum_{i \in M_l} \hat{w}_{is}$. Note that in the neighborhood of the equilibrium point, we ignore the saturation nonlinearity. Thus the equilibrium point of the closed-loop system given by (8) is obtained from

$$\begin{aligned} \hat{w}_{is} &= \frac{\omega^0(1-p_e) - R_{ls}}{N_{s_l}} \\ &= \frac{\omega^0(1-p_e)}{N_l} + \frac{(N_l - N_{S_l}) \cdot \frac{\omega^0(1-p_e)}{N_l} - R_{ls}}{N_{S_l}} \\ \hat{q}_{ls} &= q^0 \end{aligned} \tag{9}$$

where N_l is the total number of the connections routed through link l. From (9), it shows that the Max-Min fairness is achieved. That is, if N_l connections share a link, each connection achieves $1/N_l$ of the available link capacity $(\omega^0(1-p_e))$. If $(N_l - N_{S_l})$ connections use less than their shares, the unused portion $\left((N_l - N_{S_l}) \cdot \frac{\omega^0(1-p_e)}{N_l} - R_{ls}\right)$ is equally distributed among the remaining N_{S_l} connections that are locally bottlenecked connections at link l. In addition, the queue length can be stabilized at q^0. Hence an increase in the number of the connections does not lead to the increase in the queue length. This suggests that there will be no packet loss due to the congestion if each router has adequate queue size to hold the sudden bursts of packets. Therefore, when a packet loss occurs, the proposed algorithm treats it as an indication of the wireless link error and only retransmits the lost packet.

Now we investigate the asymptotic stability of the equilibrium point shown in (9). Let $x_l(n) = \frac{\hat{q}_l(n) - q^0}{q_c}$. Then

$$\begin{aligned} x_l(n+1) &= x_l(n) - \alpha_l \sum_{d=0}^{\bar{D}_l} l_d x_l(n-d) - \frac{C\tau_s}{q_c}(p_e - p_e(n)) \\ &- \sum_{i \in M_l} \alpha_{l_i} \left(x_{l_i}(n - \Delta_i) - \frac{1}{N_{s_l}} \sum_{d=0}^{\bar{D}_l} l_d x_{l_i}(n-d) \right) \end{aligned} \tag{10}$$

Let $k_1 = \frac{\epsilon_a}{\epsilon}, k_2 = \frac{\epsilon_b}{\epsilon}$, $k_{3_i} = \frac{\epsilon_{a_i}}{\epsilon}$ where $\epsilon_a = \alpha_l, \epsilon_b = \frac{C\tau_s}{q_c}$, $\epsilon_{a_i} = \alpha_{l_i}$ and $\epsilon = \min_{\forall i \in M_l}(\epsilon_a, \epsilon_b, \epsilon_{a_i})$.

$$\begin{aligned} x_l(n+1) &= x_l(n) - \epsilon(k_1 \sum_{d=0}^{\bar{D}_l} l_d x_l(n-d) + k_2(p_e - p_e(n)) \\ &+ \sum_{i \in M_l} k_{3_i}(x_{l_i}(n - \Delta_i) - \frac{1}{N_{s_l}} \sum_{d=0}^{\bar{D}_l} l_d x_{l_i}(n-d))) \end{aligned} \tag{11}$$

Since $q_c \gg 1$, without loosing much generality, we assume that $\epsilon \ll 1$. Then (11) represents a slow-in-the-average Markov walk process [13]-[15] and we can apply the asymptotic theory for such processes [13]-[15]. Let $y_l(n)$ denote the averaged value of $x_l(n)$. Then we obtain the following asymptotic approximation:

$$\begin{aligned} y_l(n+1) &= y_l(n) - \alpha_l \sum_{d=0}^{\bar{D}_l} l_d y_l(n-d) \\ &- \sum_{i \in M_l} \alpha_{l_i} \left\{ y_{l_i}(n - \Delta_i) - \frac{1}{N_{s_l}} \sum_{d=0}^{\bar{D}_l} l_d y_{l_i}(n-d) \right\} \end{aligned} \tag{12}$$

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

$$\mathbf{Y}_l(n) = [y_l(n) \;\; y_l(n-1) \;\; \cdots \;\; y_l(n-D)]^T$$

where $D = \max_{\forall l}(D_l)$ and $D_l = \max_{i \in \Omega_l}(\Delta_i)$. Let $\mathbf{Y}_l^M(n)$ be the state vector of the severely congested link of connections in M_l. Then $\mathbf{Y}_l^M(n)$ is represented by

$$\mathbf{Y}_l^M(n) = [\mathbf{Y}_{l_1}^T(n) \;\; \mathbf{Y}_{l_2}^T(n) \;\; \cdots \;\; \mathbf{Y}_{l_c}^T(n)]^T \tag{13}$$

where c is the cardinality of M_l and l_j is the severely congested link along the path of j-th connection in M_l. Then, (12) can be written by the following state equation

$$\mathbf{Y}_l(n+1) = \mathbf{A}_l \mathbf{Y}_l(n) + \mathbf{B}_l \mathbf{Y}_l^M(n) \quad (14)$$

where

$$\begin{aligned}
\mathbf{A}_l &= \begin{bmatrix} \tilde{\mathbf{A}}_{(\bar{D}_l+1)\times(\bar{D}_l+1)} & \mathbf{0}_{(\bar{D}_l+1)\times(D-\bar{D}_l)} \\ \mathbf{0}_{(D-\bar{D}_l)\times(\bar{D}_l+1)} & \mathbf{0}_{(D-\bar{D}_l)\times(D-\bar{D}_l)} \end{bmatrix}, \\
\mathbf{B}_l &= \begin{bmatrix} \mathbf{b}_{l_1} & \cdots & \mathbf{b}_{l_c} \\ & \mathbf{0}_{D\times c(D+1)} & \end{bmatrix}, \\
\mathbf{b}_{l_j} &= [b_{l_j}^0 \;\; b_{l_j}^1 \;\; \cdots \;\; b_{l_j}^{D_l} \;\vdots\; \mathbf{0}_{1\times(D-D_l)}] \\
\tilde{\mathbf{A}} &= \begin{bmatrix} 1-\alpha_l l_0 & -\alpha_l l_1 & \cdots & -\alpha_l l_{\bar{D}_l-1} & -\alpha_l l_{\bar{D}_l} \\ 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}
\end{aligned}$$

and $\mathbf{b}_{l_i}$ is $1 \times (D+1)$ vector represented by the product of α_{l_i} and $\frac{l_d}{N_{s_l}}$. Now we regroup the bottleneck link set L_b to $L_0, \cdots, L_G$ where G is the number of bottleneck link set, L_0 is the set of the links with $S_l = \Omega_l$, and L_i is the set of links l where $\mathbf{x}_l^M$ is represented by only the state of links in $L_0, \cdots$, or L_{i-1}. Let $\bar{L}_i$ be the cardinality of L_i. For the links in L_0, the second term in the right hand side of (14) is removed. Let $\mathbf{Y}(n)$ be the overall state vector of the network which is arranged by the state $\mathbf{Y}_l$ of all links in the order of $L_0, \cdots, L_G$. Then $\mathbf{Y}(n)$ is denoted by

$$\mathbf{Y}(n) = [\mathbf{Y}_0(n) \;\; \mathbf{Y}_1(n) \;\; \cdots \;\; \mathbf{Y}_G(n)]^T \quad (15)$$

where $\mathbf{Y}_i(n) = [\mathbf{Y}_{i_1}^T(n) \;\; \mathbf{Y}_{i_2}^T(n) \;\; \cdots \;\; \mathbf{Y}_{i_{\bar{L}_i}}^T(n)]$. Thus the state equation of the overall network is shown as follows:

$$\mathbf{Y}(n+1) = \mathbf{A}\mathbf{Y}(n) \quad (16)$$

with

$$\begin{aligned}
\mathbf{A} &= \begin{bmatrix} \mathbf{A}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{B}_1 & \mathbf{A}_1 & \cdots & \mathbf{0} \\ & \vdots & \ddots & \vdots \\ & \mathbf{B}_G & & \mathbf{A}_G \end{bmatrix}, \\
\mathbf{A}_i &= \begin{bmatrix} \mathbf{A}_{i_1} & \cdots & \mathbf{0} \\ \cdots & \ddots & \cdots \\ \mathbf{0} & \cdots & \mathbf{A}_{i_{\bar{L}_i}} \end{bmatrix}, \\
\mathbf{B}_i &= \begin{bmatrix} \mathbf{B}_{i_1} \\ \vdots \\ \mathbf{B}i_{\bar{L}_i} \end{bmatrix}
\end{aligned}$$

where $\mathbf{A}_i$ is a $(D+1)\bar{L}_i \times (D+1)\bar{L}_i$ matrix, $\mathbf{B}_i$ is a $(D+1)\bar{L}_i \times (D+1)(\bar{L}_0 + \bar{L}_1 + \cdots + \bar{L}_{i-1})$ matrix, and $\mathbf{A}_{i_j}$ and $\mathbf{B}_{i_j}$ are the same form of $\mathbf{A}_l$ and $\mathbf{B}_l$ in (14).
For link l, the characteristic polynomial of $\mathbf{A}_l$ of (14) is obtained as follows:

$$\Phi_l(z) = z^{\bar{D}_l+1} - z^{\bar{D}_l} + \alpha_l \sum_{d=0}^{\bar{D}_l} l_d z^{\bar{D}_l-d} \quad (17)$$

Table 2. Minimum RTTs of TCP connections.

Conn.	TCP1	TCP2	TCP3	TCP4	TCP5-TCP8
Delay(ms)	50	100	150	200	250

From [16], $\Phi_l(z)$ has all zeros within the unit circle and then, the equilibrium point is asymptotically stable if the control gain α_l satisfies the following relation:

$$0 \;\; < \alpha_l \;\; < \frac{2}{N_{S_l}} \sin\left(\frac{\pi}{4\bar{D}_l + 2}\right) \quad (18)$$

Accordingly, $\mathbf{A}_i$ for $i = 1, \cdots, G$ in (16) also have their own eigenvalues within the unit circle and all eigenvalues of $\mathbf{A}$ are located within the unit circle. Therefore, for the overall system, the equilibrium point is asymptotically stable.

3. Simulation Results

In this section, we describe simulation results of our proposed window control algorithm. We compare our proposed algorithm with the algorithm in [8]. Figure 2 illustrates the simulation model. We use the following network parameters: The link capacity C is 1500 [packet/s] and q^0 is 90 packets. The number of TCP connection M is 8 and the minimum RTT, τ_m, of each TCP connection is shown in Table 2. In addition, we fix the duration of time slot τ_s as $50ms$ and therefore the positive integers Δ_m $(1 \leq m \leq 8)$ become $1, 2, 3, 4$ for connection 1,2,3,4 respectively and 5 for connection 5-8. The packet losses by the wireless link error are occurred randomly with $p_e = 0.05$. In Fig. 2, four TCP connections share link 1 and six TCP connections share link 2. That is, connection $3, 4$ pass multiple bottleneck links (link 1 and 2) and the others pass single bottleneck link. Then connection $1, 2$ become the locally bottlenecked connections and the others (connection $3, 4$) become the remotely bottlenecked connections for router 1 (R1), while all connections routed through link 2 become the locally bottlenecked connections for router 2 (R2).

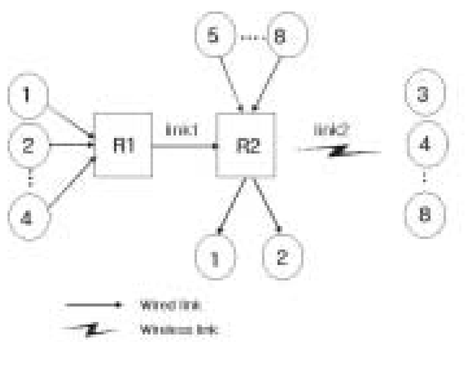

Fig. 2. Simulation model

Figure 3 shows the results of the algorithm in [8] in terms of the queue length at R2, window size, throughput and link utilization. Since it uses the bandwidth-delay product to compute the window size, the queue length at the steady state in Fig. 3 (a) is very close to zero. Figure 3 (b) shows each window size of TCP connections 1-8. In Fig. 3 (c), connections 1-2 obtain about $C/4$ bandwidth and connections

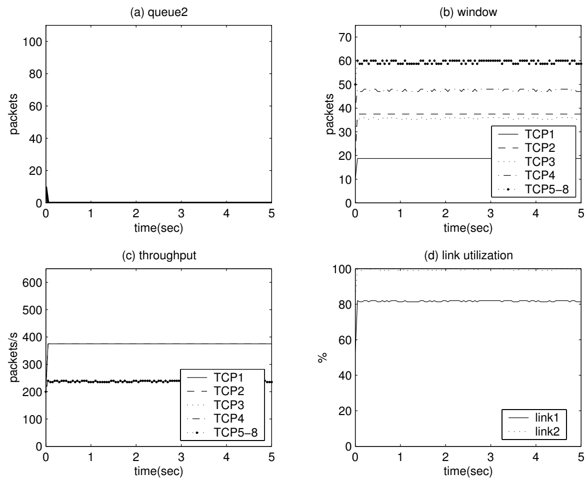

Fig. 3. Algorithm in [8]: (a) queue length, (b) window size, (c) throughput, (d) link utilization

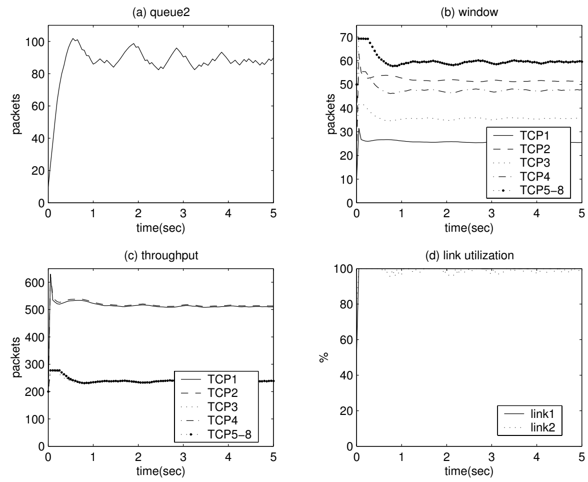

Fig. 5. Simulation result of proposed algorithm: (a) queue length, (b) window size, (c) throughput, (d) link utilization

3-8 obtain about $C/6$ bandwidth. In spite of the unused bandwidth which connection 3 and 4 cannot use, connection 1 and 2 do not share it. Therefore the algorithm in [8] doesn't utilize the link resource fully as shown in Fig. 3 (d) and doesn't guarantee the Max-Min fairness.

Figure 4 illustrates the simulation results of the approximate behavior based on (6)-(7) of the proposed algorithm. Figure 4 (a) shows the queue length at R2 and it converges to a desired value, q^0 and Fig. 4 (b) shows each window size of TCP connections 1-8. As shown in Fig. 4 (c), the throughput converges to the Max-Min fair bandwidth, i.e., TCP connections 1-2 and 3-8 send at about $C/3$ (500 packet/s) and $C/6$ (250 packet/s) respectively. That is, connections $1, 2$ which are locally bottlenecked connections in R1 share equally the unused bandwidth which the remotely bottlenecked connections cannot use. Therefore, it fully uses the link bandwidth as shown in Fig. 4 (d).

Figure 5 exhibits the simulation results of the proposed algorithm based on (1) and (5). The performances are almost same as the approximate behaviors shown in Fig. 4. However, there is a small difference as compared with Fig. 4. This small difference is caused by the assumptions that for the network described by (6) and (7), each connection sends w_m/Δ_m $(1 \leq m \leq 8)$ simultaneously every time slot (τ_s) on the average and the asymptotic approximation for the network described by (1) and (5). After considering all the factors, the proposed algorithm eliminates the unnecessary window reduction caused by the wireless link error and achieves the enhanced performance over the wireless link.

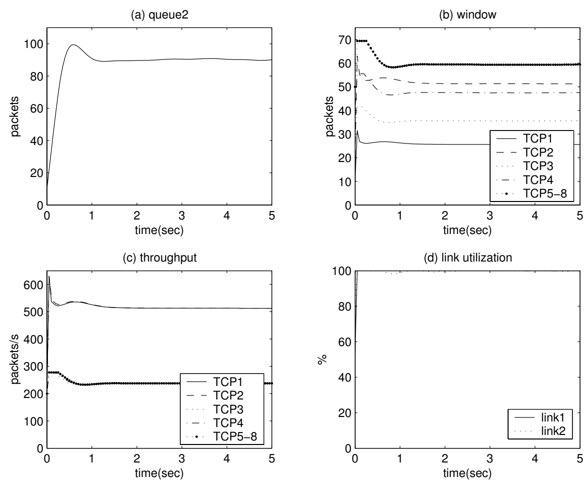

Fig. 4. Approximate behavior of proposed algorithm : (a) queue length, (b) window size, (c) throughput, (d) link utilization

4. Conclusion

In this paper, we propose a window-based congestion control algorithm in wireless TCP networks. We propose the packet marking probability to guarantee the Max-Min fairness and improve the link utilization. Specifically, we use the successive ECN congestion indications to obtain the delayed packet marking probability of the router and control the window size. The proposed algorithm avoids the congestion in the wired network and a TCP source only retransmits the lost packet on detecting the packet loss. Our simulation results show that the proposed algorithm improves TCP performance over wireless link.

References

[1] H. Balakrishnan, V. N. Padmanabhan, S. Seshan and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *IEEE/ACM Trans. Networking*, vol. 5, no. 6, pp. 756-769, 1997.

[2] W. Ding and A. Jamalipour, "A new explicit loss notification with acknowledgment for wireless TCP," *Personal, Indoor and Mobile Radio Communications*, 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. P. Fu and S. C. Liew, "TCP Veno: TCP enhancement for transmission over wireless access networks," *IEEE J. Select. Areas in Commun.*, vol. 21, no. 2, pp. 216-228, 2003.

[5] C. Liu and R. 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.

[6] 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.

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

[8] M. Gerla, W. Weng and R. L. Cigno, "BA-TCP: A bandwidth aware TCP for satellite networks," *Proc. of Computer Communications and Networks*, pp. 204-207, 1999.

[9] S. Floyd, "TCP and explicit congestion notification," *ACM Computer communication review*, vol. 24, no. 5, pp. 8-23, 1994.

[10] H.-J. Byun and J.-T. Lim, "On window-based congestion control with explicit congestion notification," *IEICE Trans. on Communications*, vol. E86-B, no. 1, pp. 421-427, 2003.

[11] B. Wydrowski and M. Zukerman, "MaxNet: A congestion control architecture," *IEEE Communications Letters*, vol.6, no. 11, pp. 512-514, 2002.

[12] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 1. no. 4, pp. 397-413, 1993.

[13] J.-T. Lim, S. M. Meerkov and T. Zeng, "Simplified description of slow-in-the-average Markov walks," *J. Math. analys and applic.*, vo. 158, no. 2, pp. 476-486, 1991.

[14] J.-T. Lim and K.-H. Shim, "Asymptotic performance evaluation of Token Passing Networks," *IEEE Trans. on Industrial Electronics*, vo. 40, no. 3, pp. 384-385, 1993.

[15] K.-H. Shim and J.-T. Lim, "Peroformance analysis and design of Token-passing networks with two message priorities," *IEE Proceedings-Communications*, vol. 44, no. 1, pp. 11-16, 1997.

[16] 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.