On Improving Wireless TCP Performance Using Supervisory Control

변희정
(Heejung Byun)
'Suwon University

Abstract: This paper proposes a systematic approach to the rate-based feedback control based on the supervisory control framework for discrete event systems. We design the supervisor to achieve the desired behavior for TCP wireless networks. From the analysis and simulation results, it is shown that the controlled networks guarantee the fair sharing of the available bandwidth and avoid the packet loss caused by the buffer overflow of TCP wireless networks.

Keywords: wireless TCP, supervisory control

I. INTRODUCTION

When a wireless link forms a part of a network, the noncongestion losses may occur due to the wireless link error and TCP will half down its congestion window size unnecessarily. A number of algorithms have been proposed to improve TCP’s performance over the wireless networks [1-10]. In [1-4], they decide whether packet losses are likely to be due to congestion or wireless link errors using the network congestion information or packet delay. The protocols proposed in [5,6] eliminate the packet loss due to the buffer overflows, so they consider the packet loss as due to the wireless link error on detecting the packet loss and only retransmit the lost packet. In [7], the bandwidth is controlled based on the radio conditions to enhance the wireless link capacity. The explicit window adaptation algorithms [8-10] suggest a congestion control scheme to explicitly inform the optimal window size to the TCP source by modifying the receiver’s advertised window (AWND) field carried by TCP ACKs. In [11-13], they develop a variant of RCP (rate control protocol) under max-min criterion and study the local stability of RCP for a single resource with a large or small buffer. However, there is far less result on methods to provide fair sharing of the bandwidth. In addition, they are heuristic in nature and they do not present the formal methods to analyze their methods.

To deal with these problems, we present a formal method to develop a rate-based feedback control scheme over the wireless TCP networks based on the supervisory control framework [14,15]. This framework can effectively capture the dynamic behavior of discrete event systems and provide the existence conditions of a supervisor so that the controlled system meets a desired specification. However, there are still few control applications based on supervisory control due to its computational complexity on state space explosion. In this paper, we propose a discrete event model of TCP networks and design a simple supervisor to improve the wireless TCP’s performance in terms of throughputs and fairness with low computational complexity.

II. PROPOSED ALGORITHM

1. Supervisory control

We briefly describe the supervisory control framework [14,15]. In the supervisory control of discrete event systems, the system to be controlled is modeled by an automaton

\[ G = (\Sigma, Q, \delta, q_0, Q_f) \]

where \( \Sigma \) is the set of events, \( Q \) is the set of states, \( q_0 \in Q \) is the initial state, \( Q_f \subseteq Q \) is the set of marker states, and \( \delta : \Sigma \times Q \rightarrow Q \), the transition function, is a partial function defined at each state in \( Q \) for a subset of \( \Sigma \). Let \( \Sigma^* \) denote the set of all finite strings (sequences) over \( \Sigma \), including the empty string \( \epsilon \). A subset of \( \Sigma^* \) is called a language over \( \Sigma \). The behavior of \( G \) is characterized by a language

\[ L(G) := \{ s \in \Sigma^* | \delta(s, q_0) \text{ is defined} \} \]

which is the set of event sequences generated in \( G \).

To impose supervision on the system, we identify some of its events as controllable and the others as uncontrollable. The event set \( \Sigma \) is partitioned into controllable and uncontrollable events, i.e., \( \Sigma = \Sigma_c \cup \Sigma_u \). The controllable events in \( \Sigma_c \) can be disabled by a supervisor, while the uncontrollable events in \( \Sigma_u \) are permanently enabled. A supervisor is then an agent which observes a sequence of events as it generated by \( G \) and enables or disables any of the controllable events.

To find out the existence of the supervisor, the following notion is necessary: Let us define \( K \) as the specification. Then a language \( K \subseteq \Sigma^* \) is called controllable with respect to \( L(G) \) if the following condition is satisfied [15]:

\[ K \cap L(G) \subseteq K \]

where \( K \) is a prefix of \( K \), i.e.,

\[ K = \{ s \in \Sigma^* | \exists t \in \Sigma^* \text{ st } s \in K \} \]

This condition requires that extension of any prefix of \( K \) by an uncontrollable event that is feasible in the system should also be a prefix of \( K \).
2. Network modeling and window rate control

Consider the wireless TCP network where the base station is the bottleneck point of the network and \( M \) TCP connections are connected to the base station. Let \( X = [C] \) where \( C \) is the capacity of outgoing link of the base station and \( [x] \) is the smallest integer larger than \( x \). Assume that the feedback rate (FR) is determined by the supervisor at the base station. The feedback rate is the available bandwidth for a connection and same to all connections routed through the base station. We partition the feedback rate into \((X + 1)\) distinct elements, i.e., \( FR \in \{0, 1, \ldots, X\} \).

To reflect the queue, we propose the free buffer function and the value of \( f_{\text{b}} \) is updated every \( T \) as follows:

\[
f_{\text{b}} = \lambda (q^\text{up} - q)
\]

where \( 0 \leq f_{\text{b}} \leq R_{\text{max}} \), \( R_{\text{max}} \) is the maximum increase/decrease of the feedback rate, \( q \) is the queue length of the base station, \( q^\text{up} \) is the queue thresholds, and \( \lambda (\lambda > 0) \) is a weighting constant to be chosen. Specifically, the value of \( T \) affects both the transient response and the control overhead. The choice of the update period \( T \) can be approached as a multicroteria optimization problem and the optimal update period can be calculated [16].

The basic idea of the proposed algorithm is to control the window rate based on the free buffer function in order to guarantee the fair sharing of the bandwidth and avoid the congestion. From this operation, our proposed algorithm can consider that the packet loss occurs due to the wireless link error by keeping the queue length at the desired queue length. To do this, the window size is controlled by a simple rule as:

\[
w \leftarrow w + f_{\text{b}}
\]

The window size increases as the free buffer length increases. On the contrary, the window size decreases as the free buffer length decreases. Since the window size is adapted every round-trip time, the network systems can be represented as the discrete event systems. Based on the supervisory control framework, we analyze the controlled system by capturing the dynamic behavior of TCP wireless networks and provide the existence conditions of a supervisor so that the controlled system meets our desired goal.

3. Supervisor design

In this section, we design the supervisor in order to guarantee the fair sharing of the bandwidth and avoid the congestion. After then, based on the supervisory control framework, we provide the existence conditions of a supervisor. Some of the notations used in this paper are listed in Table 1.

<table>
<thead>
<tr>
<th>( FR )</th>
<th>Feedback rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{\text{b}} )</td>
<td>Free buffer function</td>
</tr>
<tr>
<td>( E_{\text{F}} ), ( E_{\text{G}} ), ( E_{\text{H}} )</td>
<td>Events representing that the value of ( f_{\text{b}} ) is updated and ( f_{\text{b}} &gt; 0, f_{\text{b}} &lt; 0, f_{\text{b}} = 0 ), respectively</td>
</tr>
<tr>
<td>( E_{\text{D}}^i ) ((0 \leq i \leq X))</td>
<td>Events representing the increase or decrease of the feedback rate (up-transition or down-transition to state ( R_i ))</td>
</tr>
<tr>
<td>( E_{\text{FUP}} )</td>
<td>Event representing the maintenance of the current feedback rate</td>
</tr>
<tr>
<td>( R' )</td>
<td>State representing that the feedback rate corresponds to ( i )th element of ( FR ) and the value of ( f_{\text{b}} ) is waiting for the next update</td>
</tr>
<tr>
<td>( R'_0, R'_1, R'_p )</td>
<td>State representing that the feedback rate is ( i )th element of ( FR ), the value of ( f_{\text{b}} ) is updated, and ( q &lt; q^\text{up}, q &gt; q^\text{up}, q = q^\text{up} ), respectively</td>
</tr>
<tr>
<td>( BW_{\text{p}} )</td>
<td>Available bandwidth for TCP flows at the base station</td>
</tr>
</tbody>
</table>

Table 1. Notations.

Fig. 1. State transition diagram.

\[
\text{Act}(R'_i) = \text{Act}(R'_i \cup R'_j) = \text{Act}(R'_i \cup R'_j) = \{E_{\text{FUP}}, E_{\text{G}}, E_{\text{H}}, E_{\text{F}}\}
\]

The state transition diagram is shown in Fig. 1. When the value of \( f_{\text{b}} \) is updated at state \( R'_i \), one of uncontrollable events \( E_{\text{G}}, E_{\text{H}}, E_{\text{F}} \) is triggered. At state \( R'_i, R'_0, R'_j \), the supervisor adapts feedback rate by enabling or disabling controllable events according to the free buffer function. The objectives of the proposed method are to guarantee the fair sharing and avoid the congestion-induced packet loss by keeping the queue length at the desired queue length. The marked states represent the desired endpoints of the system. Therefore, as shown in Fig. 2, the marked states are \( R' \) and \( R'_p \) where \( p = BW_{\text{p}} / M \).

Let us define \( K_d \) as the specification to meet our objectives. We define

\[
S = \{E_{\text{G}}, E_{\text{F}}\} \cup \{E_{\text{G}}, E_{\text{F}}\}(0 \leq i \leq X)(0 \leq j \leq X - 1)
\]

Then the specification \( K_d \) is described as
The supervisor should enable $G$ or event $E$ means that the network is over-utilized, cannot occur as long as the network parameters $GK$ do not change. As it follows from the above, since state $q$, the supervisor should decrease the feedback rate by $\Delta E$. Moreover, if the queue length becomes the queue threshold, event $E$ cannot occur as long as the network parameters do not change. As it follows from the above, since $K_s$, the supervisor adapts the feedback rate according to the free buffer function so that the queue length at steady state, the sum of incoming window rates $(r_i)$ to the base station equals to $BW_f$:

$$\sum_{i=1}^{M} r_i = \sum_{i=1}^{M} FR = BW_f$$

That is, if $K_d$ is achieved, the controlled network guarantees the fair sharing of the bandwidth, avoids the congestion, and considers that the packet loss occurs due to the wireless link error by keeping the queue length at the desired queue length. Therefore, the proposed supervisor can drive the network to reach the marked states.

### III. SIMULATION RESULTS

We describe simulation results of our proposed window control. First, we compare the proposed method with TCP Veno [2] and XCP/RCP [12]. TCP Veno monitors the network congestion level and uses this information to decide whether packet losses are likely to be due to congestion or random bit error. The simulation model is shown in Fig. 3. We divide the connections into 3 Groups, which have $M_i (1 \leq i \leq 3)$ connections respectively. The round-trip propagation time of all the connections in Group $i$ is $T=100\cdot i$ ms. We use the following network parameters: The wired network is a LAN speed network and the base station is linked to the destinations via 10Mb/s noisy channel. The full buffer FTP traffic is generated and the packet size is 1Kbytes. We set $q_0=30$ Kbytes, $T=100$ms, and $\lambda=1$.

Fig. 4 shows the throughput with $M_1=M_2=M_3=1$. As shown in Fig. 4(a), TCP Veno has a large oscillation in the throughput and the fair sharing of the bandwidth is not achieved for a considerable period. Fig. 4(b) shows the throughput of

<table>
<thead>
<tr>
<th>state</th>
<th>enabled event</th>
<th>condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_p^l$</td>
<td>$E_{TMD}$</td>
<td>$i \leq (X-\Delta)$</td>
</tr>
<tr>
<td>$R_p^l$</td>
<td>$E_{TMD}$</td>
<td>$(X-\Delta) &lt; i &lt; X$</td>
</tr>
<tr>
<td>$R_p^l$</td>
<td>$E_{TMD}$</td>
<td>$i = X$</td>
</tr>
<tr>
<td>$R_p^l$</td>
<td>$E_{TMD}$</td>
<td>$0 &lt; i &lt; \Delta$</td>
</tr>
<tr>
<td>$R_p^l$</td>
<td>$E_{TMD}$</td>
<td>$i = 0$</td>
</tr>
</tbody>
</table>

**Table 2. Supervisor $S$.**
It shows the stable behavior but the fair sharing of bandwidth is not achieved among TCP connections. However, the proposed algorithm as shown in Fig. 4(c) guarantees the fair share of the bandwidth with rather small oscillation.

Fig. 5 shows the normalized throughput as the packet loss rate varies. For the proposed algorithm and XCP/RCP, the link is run close to capacity. Especially at the high error rate, the proposed and XCP/RCP method still have satisfactory throughput whereas Veno experiences degradation in the throughput. This is because the proposed supervisor and XCP/RCP method keep the queue length nonzero.

To show the fairness in the throughput, we measure the fairness index [4]. The values of fairness range from 1/M to 1, with 1 corresponding to the best fair allocation among all connections. We consider multiple connections (1-10 for each Group). As shown in Fig. 6, XCP/RCP does not guarantee the fairness among TCP connections. TCP Veno and the proposed method achieve satisfactory fairness index, but the fairness index for the proposed method is higher than TCP Veno. This is because the proposed supervisor determines the feedback rate with the common buffer length and all connections routed through the base station transmit the packets with the same window rate. From the above results, we conclude that the controlled network significantly improves wireless TCP performance and guarantees the fair sharing of the available bandwidth.

**IV. CONCLUSIONS**

In this paper we propose the discrete event system approach to accommodate the rate-based feedback control scheme and make the supervisory control feasible for the network application. From the simulations, the controlled network has revealed better performance compared with the conventional schemes.

**REFERENCES**


변 희 정