Studies on Data Transport Protocols for Wireless Cellular Networks

Masahiro Miyoshi

June 2002

Department of Informatics and Mathematical Science

Graduate School of Engineering Science

Osaka University

Preface

Mobile Internet technology based on wireless cellular networks has been rapidly deployed over the last several years. In the Internet, TCP (Transmission Control Protocol) is used as a standard transport layer protocol. TCP was developed in the early 1970s and has been continuously improved. Though it was designed for a wired network, it has also been used for wireless networks but without enough study or evaluation. Digital mobile phone services based on wireless cellular networks, such as PDC (Personal Digital Cellular) or GSM (Global System for Mobile communications), were first provided in the early 1990s. These services will be further developed based on high-speed packet data transmission as IMT-2000 (International Mobile Telecommunications 2000) services in the early 2000s. In this thesis, we study TCP as a data transport protocol for wireless cellular networks based on the IMT-2000 system.

In studying the use of TCP over wireless cellular networks, certain issues need to be considered. First, we should consider the characteristics of the underlying data link protocol. Secondly, packet losses due to transmission errors on the wireless link cause unexpected degradation of TCP throughput.

In this thesis, we first evaluate TCP performance by explicitly modeling the performance characteristics of the underlying data link layer protocol. We adopt slotted ALOHA, a multiple access method, as the data link layer protocol. Then, we construct an analytical model for TCP throughput in the wireless cellular that takes into consideration the influence of slotted ALOHA. Using our method, we next evaluate TCP throughput through the wireless cellular network and show that improving throughput at the data link layer level does not necessarily lead to improved TCP throughput. We also show that the greater reliability provided by data transmission enhancement in the lower layer protocols, such as ARQ (Automatic Repeat reQuest) and FEC (Forward Error Correction), can improve TCP performance. It is especially important for error correction to adequately change the

FEC parameters according to the noise level.

Next, we study the problem of packet losses due to transmission errors causing unexpected degradation of TCP throughput in a wireless cellular network. Many approaches to improving TCP throughput in a wireless cellular network have been reported. We summarize these approaches and evaluate TCP performance, through which we find the appropriate FEC code for a given BER (Bit Error Rate). Based on above these results, we propose a new adaptive FEC scheme that combines these approaches, and show that our proposed method suppresses TCP performance degradation.

For the sake of simplicity and easy implementation, we also propose a method for improving the TCP performance through a minor change in the way ACK packets are treated. In our proposed method, a TCP receiver sends multiple ACKs, rather than just one, and this make TCP more robust regarding wireless link errors. We describe an analytical method, and show that two ACKs for each TCP packet are sufficient to improve the performance.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Backgrounds

The application of Internet technology has spread wider than anyone expected over the past several years, and people worldwide have benefited in various ways from the growth of Internet. The Internet is based on TCP/IP, a combination of TCP (Transmission Control Protocol) [1] and IP (Internet Protocol) [2]. Through continuous improvement since it was developed in the early 1970s, TCP has enabled fairly reliable Internet communication. Although it was initially designed for a wired network, TCP is now also used for the wireless networks developed in response to the rapidly growing demand for mobile communication. However, this application of TCP has been implemented with surprisingly little study evaluation.

When studying the use of TCP over wireless cellular networks, several issues should be considered. First, we need to consider the characteristics of the underlying data link protocol that is used for communication between the BS (Base Station) and wireless terminals. The performance of data link protocols only has been studied [3-5], where the data packet transmission was considered for CDMA (Code Division Multiple Access) channels. TCP performance on wireless cellular networks has also been examined [6,7], but the influence of the lower layer protocols was not considerd.

Secondly, we need to realize that packet losses caused by transmission errors can unexpectedly degrade TCP throughput in wireless cellular networks. In the Internet, TCP recognizes a congestion occurrence within the network by a packet loss, and performs congestion control by throttling the congestion window. Then, we encounter the problem that the packet losses due to the transmission errors cause unexpected degradation of TCP throughput in the wireless cellular networks. Many approaches have been proposed to improve TCP throughput in wireless cellular networks. The split connection approach, such as Indirect-TCP [8], is an early proposal in the field. It involves splitting each TCP connections between a sender and a receiver into two separate connections. Snoop Protocol [9] is an improved scheme of split connection approaches, and it retains end-to-end semantics; it uses the Snoop Agent to cache the TCP segment and retransmits the segment only on the wireless link. ELN (Explicit Loss Notification) [6] is a more precise approach with a capability of distinguishing the packet loss due to congestion from the one by transmission errors. Another approach to improving TCP performance is to enhance the lower layer protocol, i.e., the data link layer protocol, by incorporating, e.g., ARQ (Automatic Repeat reQuest) and/or FEC (Forward Error Correction) [10]. FEC is a simplest solution to improve the bit error ratio seen by the higher layer protocol. However, it is inefficient if the error condition of the wireless channel varies greatly. Accordingly, an adaptive error correction scheme is proposed in [11] in order to compensate such a drawback of FEC. However, utilizing the adaptive error correction solely is not a realistic solution because it is complicated to measure BER (Bit Error Rate) for each bit or the unit of several bits and change the FEC code appropriately against irregular wireless errors.

Of course, above solutions are effective. However, these solutions have not been realized because these require major changes to network infrastructures. For example, ELN needs to change both the BSs and wireless terminals to observe packet losses on the radio link. On this account, we should approach considering easy for implementation.

1.2 Congestion Control Mechanisms of TCP

TCP applies a window-based congestion control mechanism. It recognizes the congestion occurrence of the network and throttles the window size, when the packet loss is detected. The TCP sender determines that a packet is lost through two mechanisms [14]. The one is Fast Retransmit and Fast Recovery mechanisms, which throttle the congestion window size to half, if the TCP sender detects triple duplicate ACKs continuously. The other is Retransmission Timeout mechanism, which draws back the congestion window size to 1 MSS (Maximum Segment Size), if the ACK for the TCP segment is not received before the retransmission timeout timer expiration.

Fast Retransmit and Fast Recovery

Fast Retransmit and Fast Recovery mechanism works as shown in Figure 1.1.

(1) The TCP sender sends the TCP segment added on sequence number.
(2) The TCP receiver sends the ACK added on sequence number, which should be received.
(3) If triple duplicate ACKs are received continuously, the packet is assumed to be lost.

In this case, the TCP sender sends a TCP segment that throttle the congestion window size to half.

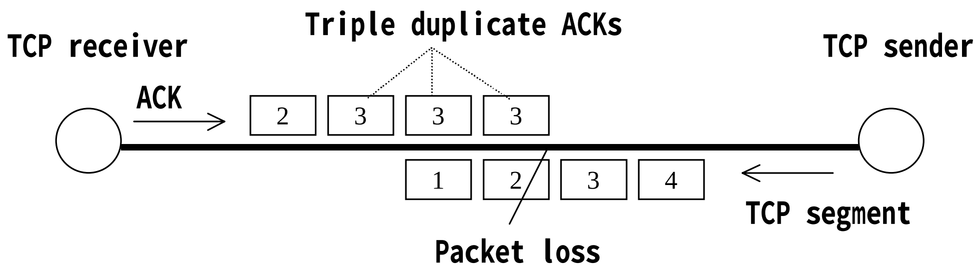


Figure 1.1 Fast Retransmit and Fast Recovery

Retransmission Timeout

Retransmission Timeout mechanism applies the observed RTT (Round Trip Time) value and RTO (Retransmission TimeOut) value, which is calculated as

$$To = rtt\_old + 4rtt\_var \tag{1.1}$$

where *To* is the calculated RTO value, *rtt_old* is the last RTT value, and *rtt_var* is the RTT variance. Retransmission Timeout mechanism works as shown in Figure 1.2.

(1) The TCP sender calculates *To*.
(2) The TCP sender sets the retransmission timer.
(3) If the ACK for the timed packet is not received, before the timer expiration, the packet is assumed to be lost.

In this case, the TCP sender sends a TCP segment that draws back the congestion window size to 1 MSS.

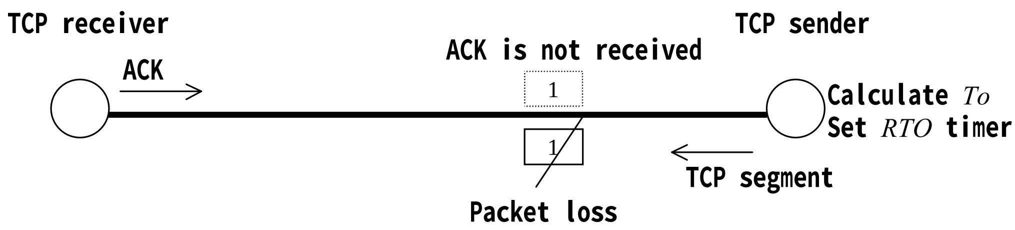


Figure 1.2 Retransmission Timeout

1.3 Wireless Error Model

In wireless cellular network environments, propagating radio waves are subject to reflection, diffraction, and scattering. As the results, these different propagation patterns affect the free space loss, shadowing, and multipath fading. These effects and random noise are the main causes of wireless errors. In studying TCP over a wireless link, we considered wireless errors to be either random bit errors caused by the influence of random noise, or burst errors caused by the propagation characteristics of the wireless link.

Empirical error models have been proposed, such as the trace-based model [11]. In this case, the empirical error model is defined according to the trace conditions, since the wireless error depends on the wireless environment.

Gilbert model [15] is the conventional method, and it generates burst errors under the condition of shadowing or multipath fading. As shown in Figure 1.3, the model consists of two states; “Good” and “Bad” states. Here, the state is defined randomly between “Good” and “Bad ” with transition probabilities p_{GB} (Good→Bad), p_{BG} (Bad→Good). The burst errors are generated only in “Bad” states. The appearance of generated burst errors from the Gilbert model is shown in Figure 1.4. As show in the figure, the state is changed by turns and the term of state is random. We used the Gilbert model to generate burst errors in this thesis.

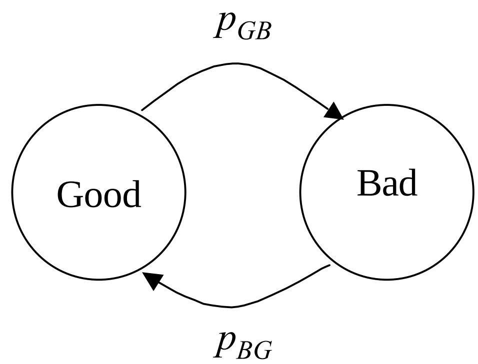


Figure 1.3 Gilbert Model

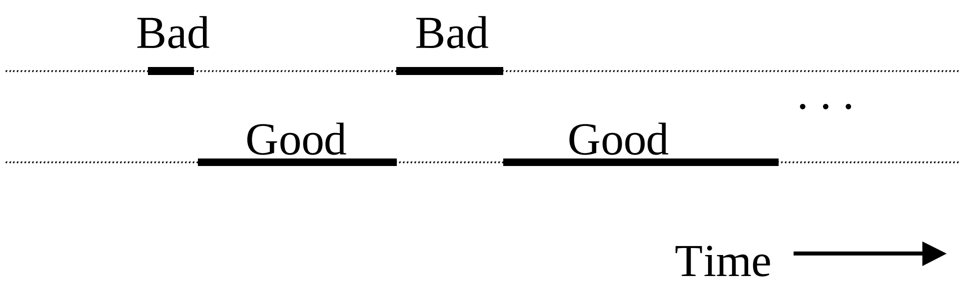


Figure 1.4 Generating Burst Errors by Gilbert Model

1.4 Outline of Thesis

TCP has provided reliable communication in wireless cellular networks for some time. Nevertheless, the performance of TCP for wireless cellular networks has not been sufficiently evaluated. Several issues need to be considered the use of TCP in wireless cellular networks. First, we should consider the characteristics of the underlying data link protocol, used for the communication between the BS and wireless terminals. Secondly, the frequent occurrence of transmission errors on a wireless link has to taken into account. Therefore, we focus on two objectives in this thesis.

- Performance evaluation of TCP for wireless cellular networks taking into consideration the lower layer protocol.
- Improving TCP performance on wireless cellular networks

Here, we summarize these objectives with regard to related works.

Performance Evaluation of TCP for Wireless Cellular Networks Considering Lower Layer Protocol [30-32]

When studying the use of TCP for wireless cellular networks, we should take into account the characteristics of the underlying data link protocol that is used for communication between the BS and wireless terminals. The performance of only the data link protocols has been studied [3-5], where the data packet transmission on CDMA channels was considered. TCP performance on wireless cellular networks has also been studied [6,7], but the influence of the lower layer protocols was not considered. Therefore, we have evaluated TCP performance by explicitly modeling the performance characteristics of the underlying data link layer protocol. For IMT-2000 (International Mobile Telecommunications 2000) [16], we adopted slotted ALOHA as the data link layer protocol. Our performance model clearly shows that improving the throughput of the data link layer level does not necessarily improve the TCP throughput. It is shown that reliability enhancements for the data transmission in the lower layer protocols such as ARQ and FEC can improve the TCP performance. And we show that FEC is more effective than ARQ to prevent TCP performance degradation. It is especially important to adequately change the FEC parameters for an error correction capability according to the noise level.

Improving TCP Performance on Wireless Cellular Networks [33-38]

In the Internet, where TCP is used as a standard transport layer protocol, TCP recognizes network congestion from the packet loss, and performs congestion control by adjusting the congestion window. The problem then arises that packet losses due to transmission errors unexpectedly degrade the TCP throughput in a wireless cellular network. Many approaches have been taken to improve TCP throughput in a wireless cellular network. The split connection approach, such as Indirect-TCP [8], was an early proposal. This involves splitting each TCP connections between a sender and a receiver into two separate connections. Snoop Protocol [9] is an improved split connection approach that retains the end-to-end semantics; it uses the Snoop Agent to cache the TCP segment and retransmits the segment only on the wireless link. ELN [6] is a more precise approach that can distinguish between packet loss caused by congestion and that caused by transmission errors. Another approach to improving TCP performance is to enhance the lower layer protocol, i.e., the data link layer protocol, by incorporating, e.g., ARQ and/or FEC [10]. FEC is the simplest way to improve the bit error ratio as seen by the higher layer protocol. However, it is inefficient if the error condition of the radio channel varies greatly. Accordingly, an adaptive error correction scheme has been developed [11] that compensates for this drawback of FEC. However, using adaptive error correction only is not a realistic solution because of the complexity of measuring the BER for each bit or each unit of several bits and appropriately changing the FEC code in response to irregular wireless errors. For this reason, we propose a new adaptive FEC scheme combined with ELN, and we show a method for deciding threshold value at which to change the adaptive FEC code, which was not available for the earlier scheme [11]. In our method, transmission errors on the wireless link are measured at the packet level and the TCP sender is notified of the error status through ELN. According to this information, an appropriate FEC code is selected. We first evaluate the TCP performance using Snoop Protocol, ELN, and the fixed FEC, through which we find the appropriate FEC code for a given BER. We then show how adaptive FEC can be realized by using our solution, and also examine the appropriate observation period for measuring BER for the fading speed on a noisy wireless link. After that, we evaluate our proposed method by using the Gilbert model as a wireless error model, and show that our method enables better performance than the conventional fixed FEC.

A shortcoming of previous solutions has also been that they cannot be realized without making considerable changes to the network infra-structures. For example, ELN requires that both the BSs and the wireless terminals be changed to allow the observation of packet losses on wireless links. To avoid this problem, we improve the TCP performance through a minor change in the way ACK packets are handled. In this method, the TCP receiver sends multiple ACKs, rather than just one when the packet loss probability exceeds the predefined threshold. Since the major applications of the current Internet are to download the Web documents from the fixed servers to the wireless terminal, proposed method only requires changes at the wireless terminal side for performance improvement. Wireless terminals monitor a packet loss probability, and estimate the error probability of the wireless channel. Then, TCP becomes robust against wireless link errors. We present an analytical method, and show that sending two ACKs for each packet is sufficient to improve TCP performance.

Chapter 2

Modeling TCP Considering Lower Layer Protocols

In studying TCP for wireless cellular networks, we should consider the characteristics of the underlying data link protocol. Performance of only the data link protocols is studied in [3-5], where the data packet transmission is considered on CDMA (Code Division Multiple Access) channels. TCP performance on the wireless cellular networks is studied in [6,7], but the authors in those papers do not consider the influence of the lower layer protocols.

In this chapter, we construct an analytical model for TCP throughput in wireless cellular networks. In our analytic model, the influence of data link layer and transmission errors on the wireless link has been explicitly modeled. Actually, we consider slotted ALOHA as multiple access control protocol and ARQ (Automatic Repeat reQuest) and/or FEC (Forward Error Correction) as transmission error detection/correction protocol. We also show that an accuracy of our analysis model is examined by comparing with simulation results.

Next, we evaluate TCP throughput on the wireless cellular networks and show that improving throughput at the data link layer level does not necessarily lead to the TCP throughput improvement. Furthermore, we evaluate TCP throughput by considering transmission errors on the wireless link. It is shown that when we introduce FEC (Forward Error Correction) as an error correction method, we show that TCP throughput can be improved by selecting an appropriate error correction code with careful consideration on the overhead and the error correction capability according to the quality of the wireless channel.

2.1 Network Model

The network model for our analysis is shown in Figure 2.1, which consists of the wireless terminal, BS (Base Station), wired terminal. Parameters are summarized in Table 2.1. We consider that the bandwidth of the wireless link is 2Mbps for maximum service in IMT-2000 (International Mobile Telecommunications 2000) [15], and the wired link is 125kbps supporting for the dual link N-ISDN (Narrowband Integrated Services Digital Network). We assume that TCP segments are transmitted towards the wired terminal from the wireless terminal, and ACK segments are in the reverse direction. We also assume the TCP Reno version, which is a major implementation in the current TCP code. In this configuration, packet loss is mainly caused by the transmission error on the wireless link, and the packet loss due to buffer overflow occurs at the BS buffer.

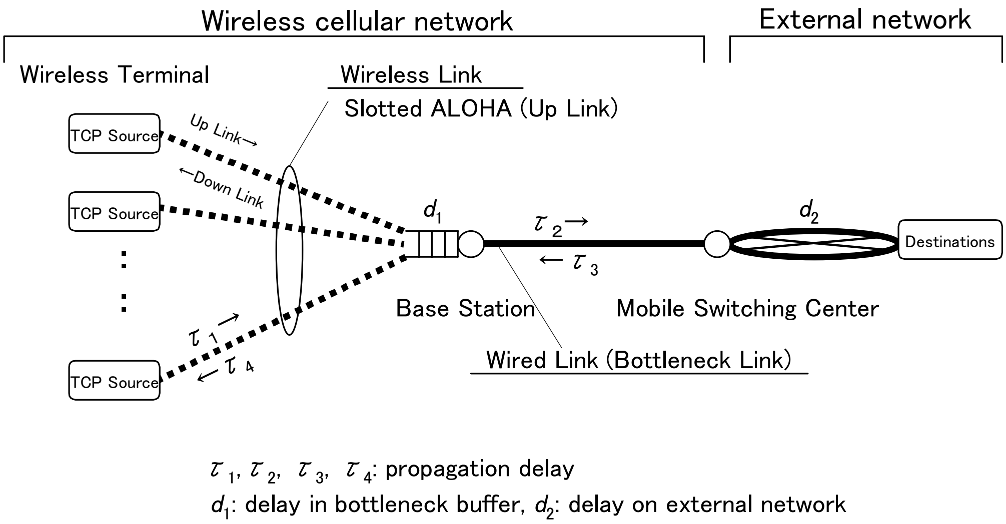


Figure 2.1 Model of Wireless Cellular Networks

Table 2.1 Evaluation Parameters (comparison of analysis and simulation)

Bandwidth of wireless link	2 Mbps
Bandwidth of wired link	125 kbps
TCP segment size	100 bytes
Packet transmission interval N	70 packets (5 nodes) 35 packets (30 nodes)
External network delay $d2$	100 ms

The data link layer model for our analysis is shown in Figure 2.2, which is contained the function of multiple access control and transmission error detection/correction. In the wireless terminal, there are two functions of retransmission due to transmission error and retransmission due to slotted ALOHA located below. At the BS, there are two functions of transmission error detection/correction and collision detection by slotted ALOHA located below. Here, the configuration of slot is show in Figure 2.3, which is assumed one slot of data link layer equal to one TCP segment. This is because of simplified our analysis.

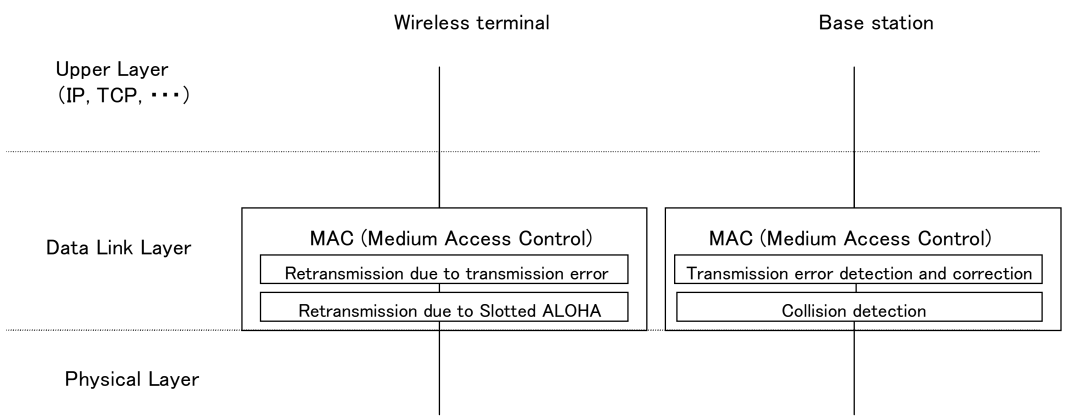


Figure 2.2 Model of Data Link Layer

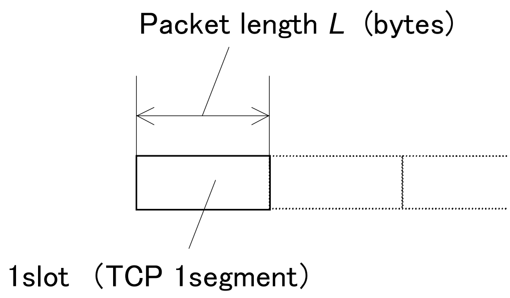


Figure 2.3 Configuration of Slot

We assume that retransmission due to slotted ALOHA is implemented using *Stop and Wait* manner. On the other hand, we assume that retransmission due to error correcting protocol is implemented using *Go back N* manner, which does not wait for each packet arrival. In the view of TCP, former are considered as delay but not packet losses, latter are considered as packet losses. Finally, we explain traffic flow, which is assumed slotted ALOHA as the lower layer protocol in Figure 2.4.

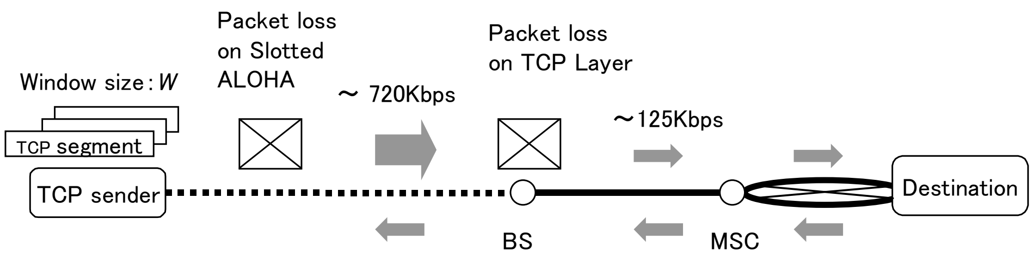


Figure 2.4 Traffic Flow

(1) Due to collisions by slotted ALOHA, the traffic volume, which is transmit from TCP sender, is restricted under 2Mbps (wireless link)×0.36 (maximum throughput using slotted ALOHA) = 720kbps.

(2) The traffic volume is restricted under 125kbps. Since the packet losses occur due to buffer overflow at the BS and due to the transmission errors on the wireless link.

(3) The traffic volume does not change through reverse direction, because we assume no factor, which changes traffic volume.

(4) The TCP sender, which received ACK, transmits TCP segment whose window size is W.

Thus, the traffic volume is controlled by TCP congestion control. In the view of slotted ALOHA, the traffic load are changed due to congestion control of TCP.

2.2 Analytical Model of TCP Throughput

In this section, we construct an analytical model of TCP considering lower layer protocol. Here, we explain about *RTT* (Round Trip Time) in Subsection 2.2.1, and p in Subsection 2.2.2.

2.2.1 Round Trip Time : *RTT*

We explain the analytical TCP throughput model for wireless cellular networks. The network model for our analysis is shown in Figure 2.1. TCP throughput, S_{TCP}, in our analysis is characterized by three parameters *RTT*, *To* (Timeout), p and is given as [16].

$$S_{TCP} = \frac{1}{RTT\sqrt{\frac{2bp}{3} + To\min(1,3\sqrt{\frac{3bp}{8}})p(1+32p^2)}} \tag{2.1}$$

$$W = \frac{2+b}{3b} + \sqrt{\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b}\right)^2} \tag{2.2}$$

where W is the average window size of the TCP connection, and b is a delayed ACK parameter. Normally, b=2.

In wireless cellular network environments, values of RTT and To must be influenced by the packet loss characteristics of slotted ALOHA because of frame retransmissions at the data link layer. TCP segment loss probability p is also affected by the transmission error on a wireless link. In what follows, we will show RTT and To values encountered by TCP when the slotted ALOHA is used as the data link layer protocol. To clearly investigate the error characteristics of the wireless networks, we consider that the packet loss probability p is given by the sum of p_{err} (packet loss due to transmission error) and p_{buff} (packet loss due to buffer overflow). For the slotted ALOHA, we first determine an expected value of RTT as follows:

$$E[RTT] = \tau_1 + \tau_2 + \tau_3 + \tau_4 + d_1 + d_2 \tag{2.3}$$

where, d_1 is delay in bottleneck buffer, d_2 is delay on external network, τ_1, τ_2, τ_3, τ_4 is propagation delay. See Figure 2.1 each of them.

Due to the link contention, the packet transmission delay experienced in the slotted ALOHA network, τ_1, is likely to be much larger than the propagation delays τ_2 through τ_4. Eq. (2.3) thus may be simply rewritten as

$$E[RTT] \approx \tau_1 + d_2 \tag{2.4}$$

We next want to determine the packet transmission delay, τ_1. It is well known that the throughput of the slotted ALOHA can be determined by

$$S_{ALOHA} = G\exp(-G) \tag{2.5}$$

$$G = \frac{n}{N} \tag{2.6}$$

where n shows the number of terminals and N is the packet transmission interval. G is offer load. The underlying assumption for the above equation is that the number of terminals is large and the Poisson arrivals can be assumed for the traffic generation. Apparently, it is not true in our case: nevertheless we introduce it for the following derivation, and will validate its accuracy by comparing with simulation results. In our analysis, the traffic load, G, is given by

$$G = \frac{nW}{N} \tag{2.7}$$

An expected value of the up-link delay, τ_1 and each packet time L_t is then represented the following equation by S_{ALOHA} when packet collision happens i times in the slotted ALOHA.

$$E[\tau_1]_i = (i+1)NL_t(1-S_{ALOHA})^i S_{ALOHA} \tag{2.8}$$

The sum of Eq. (2.8) correspond to expected value of the up-link delay, τ_1,

$$E[\tau_1] = \sum_{i=0}^{\infty}(i+1)NL_t(1-S_{ALOHA})^i S_{ALOHA} \tag{2.9}$$

Then we can derive $E[RTT]$ by adding the external network delay d_2 as:

$$E[RTT] = \sum_{i=0}^{\infty}(i+1)NL_t(1-S_{ALOHA})^i S_{ALOHA} + d_2 \tag{2.10}$$

An expected value of the timeout time is given in [13] as:

$$To = rtt\_old + 4rtt\_var \tag{2.11}$$

In our case, it is written by:

$$E[To] = E[RTT] + 4\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\left|E[\tau_1]_i - E[\tau_1]_j\right| \tag{2.12}$$

2.2.2 Packet Loss Rate : *p*

As shown in Figure 2.5, we assume that the packet loss rate p_{err} (packet loss due to transmission error) and p_{buff} (packet loss due to buffer overflow) is serial. Then we can derive the packet loss rate *p*. By assuming that the transmission error on the wireless link and buffer overflow occur independently, *p* is given by p_{err} and p_{buff} as follows:

$$p = p_{err} + p_{buff} - p_{err}p_{buff} \tag{2.13}$$

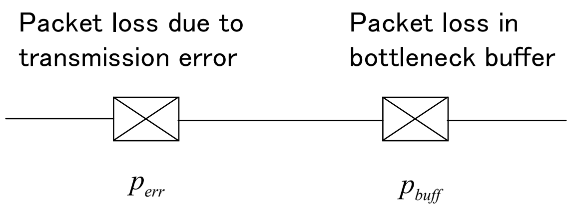


Figure 2.5 Deriving Packet Loss Rate *p*

Above all, we can now determine the TCP throughput on the wireless channels using following parameter based on Eqs. (2.1) and (2.2).

n: the number of terminals
N : the packet transmission interval

L_t: each packet time
d_2 : delay on external network
p_{err} : packet loss rate due to transmission error
p_{buff} : packet loss rate due to buffer overflow
,

2.3 Validation of Analytical Model

As show in Figure 2.6, we validated the accuracy of our analysis by comparing with simulation using ns-2 [17], with the library of the slotted ALOHA.

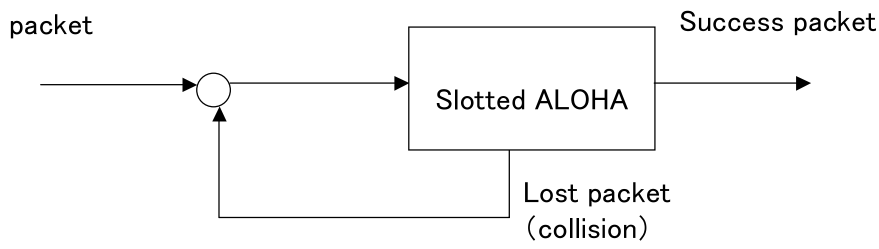


Figure 2.6 Model of Slotted ALOHA

According to Figure 2.7, we explain generating packet. In the case of wireless terminal “A”, the new traffic, which is the number of window size W, is generated after RTT time when the traffic was generated. As show in Figure 2.8, retransmission is simulated for each packet. Here, the interval (slot) of retransmission is selected, in $1 \leqq t \leqq 2N$ (t = 1, 2, $\cdots$,$2N$, where N is the packet transmission interval), randomly.

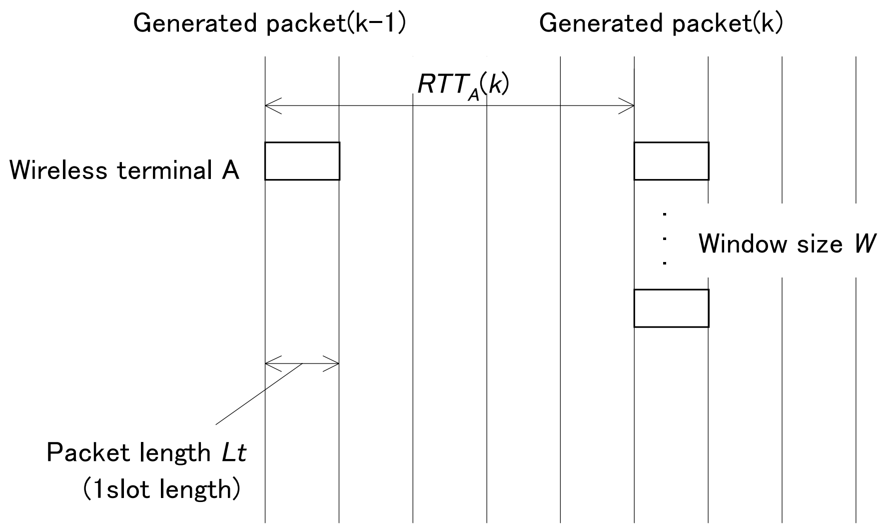


Figure 2.7 Model of Generating Packet

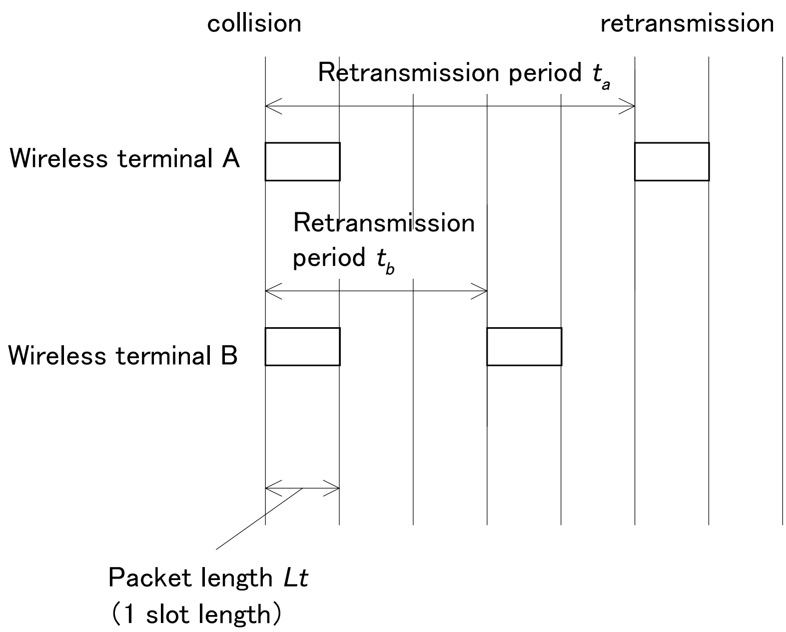


Figure 2.8 Model of Retransmission for Packet Loss

We evaluated TCP throughput per connection versus p, which is represented in Eq. (2.13) by p_{err} (packet loss rate due to transmission error) and p_{buff} (packet loss rate due to buffer overflow). The parameters are summarized in Table 2.1, and the results of simulation and analysis are shown in Figure 2.9 and Figure 2.10.

In comparison, we first store the data of the p_{buff}, which changed the p_{err} by simulation. Thus, we get the combination of (p_{buff}, p_{err}). Then we calculated the p by the expression (2.13) and simulated using this combination of (p_{buff}, p_{err}). We show this combination in Table 2.2 and Table 2.3.

Table 2.2 Packet Loss Rate
(the number of wireless terminals is 5)

p	p_{err}	p_{buff}
0.138	0.054	0.089
0.140	0.098	0.047
0.148	0.113	0.039
0.226	0.213	0.017
0.55	0.55	0

Table 2.3 Packet Loss Rate
(the number of wireless terminals is 30)

p	p_{err}	p_{buff}
0.163	0.054	0.115
0.167	0.098	0.077
0.171	0.113	0.065
0.213	0.213	0
0.55	0.55	0

When we change the traffic load G, changing the number of wireless terminal n and the packet transmission interval N, the simulated values are lager than the analytical values in Figure 2.9. On the other hand, the analytical values are lager than the simulated values in Figure 2.10.

This is due to approximation by Eq. (2.1), which is derived for the TCP throughput S_{TCP} and Eq. (2.2), which is derived for the slotted ALOHA throughput S_{ALOHA}. In Eq. (2.1), S_{TCP} would reach smaller than actual value because this equation does not consider slow start mechanism of TCP as the mention [16], which increase window size by exponential until the predefined threshold value. In Eq. (2.2), it is known that S_{ALOHA} would reach smaller than actual value when the traffic load G is smaller, and then RTT, To would reach smaller value according to Eq. (2.10) and Eq. (2.11), respectively. Then, S_{TCP} would reach smaller than actual value.

In short, analyzed S_{TCP} would reach smaller than actual value, but it would reach larger when the traffic load G is large. For this reason, analyzed S_{TCP} would reach smaller than simulated value in Figure 2.9. On the other hand, analyzed S_{TCP} would reach larger than simulated value in Figure 2.10 because of larger G. On the whole, the analysis results are in a good agreement with simulation results.

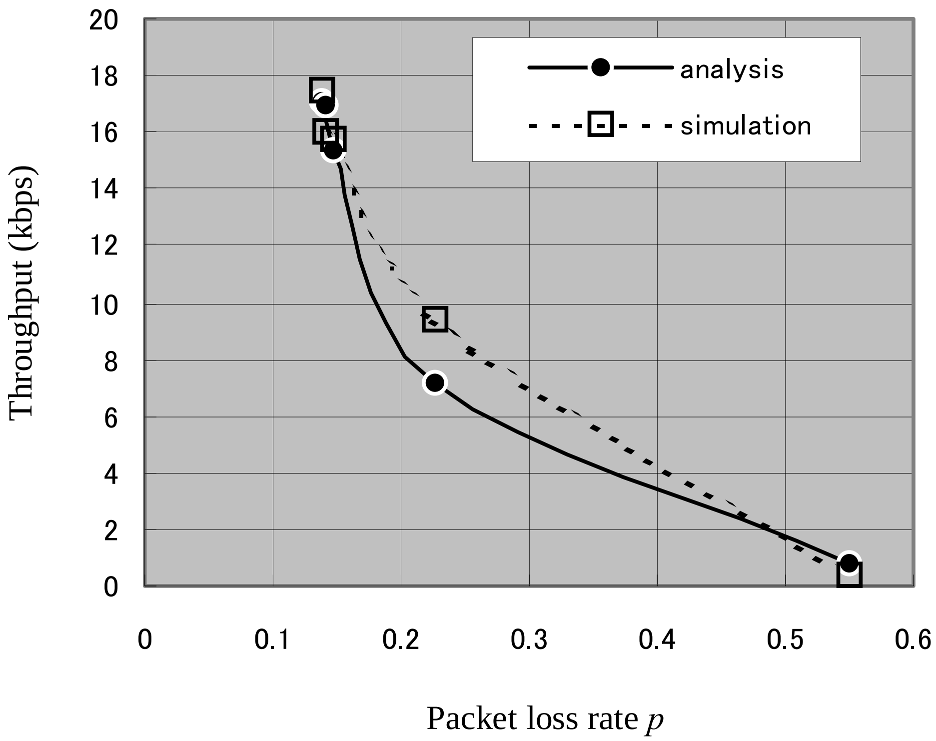


Figure 2.9 TCP Throughput Comparison
(the number of wireless terminal is 5)

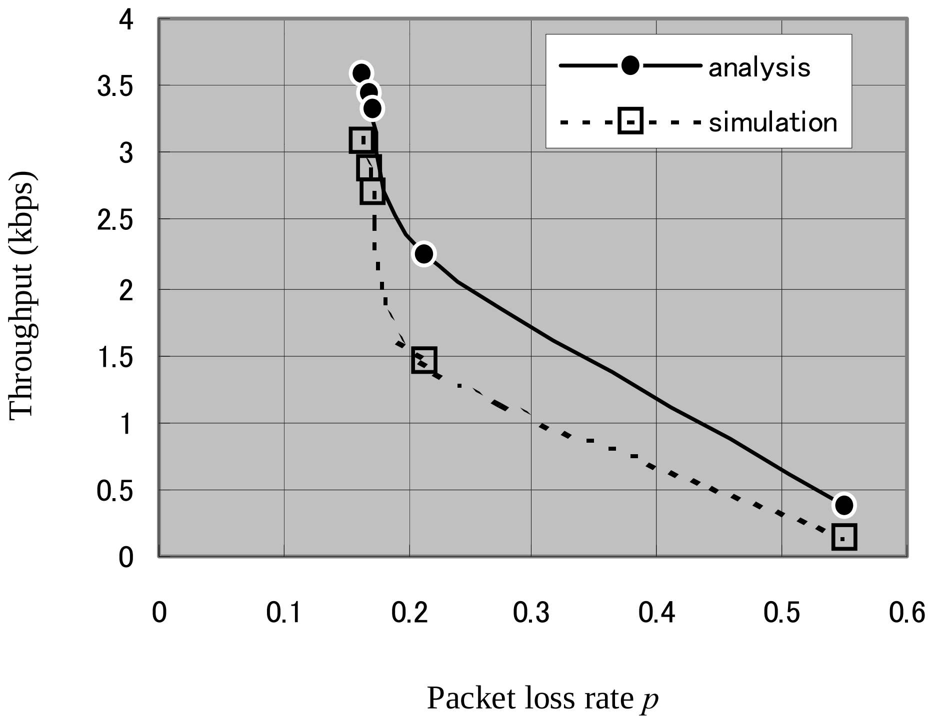


Figure 2.10 TCP Throughput Comparison
(the number of wireless terminal is 30)

2.4 Performance Evaluation

In this section, we evaluate the influence of slotted ALOHA on TCP throughput performance and the influence of the transmission errors on the wireless link on TCP performance. We show some numerical results and add to our study.

2.4.1 Influence of Slotted ALOHA Protocol

In this subsection, the influence of slotted ALOHA on TCP throughput performance is evaluated using our analysis method. Table 2.4 shows parameters set that we used in evaluation. We show the throughput values of slotted ALOHA and TCP in Figure 2.11, and Figure 2.12, respectively. In these figures, the horizontal axis represents the packet retransmission interval. As shown in the figure, as N is increased, slotted ALOHA throughput reaches at its maximum (around N = 80 in the current case), and then decreases. However, TCP throughput exhibits different results. It is because at N = 80, packet loss occurs frequently as shown in Figure 2.12. Thus, it is clear that improving throughput on the data link layer level does not necessarily lead to the TCP throughput improvement.

Table 2.4 Evaluation Parameters (comparison of S_{ALOHA} and S_{TCP})

Bandwidth of radio link	2 Mbps
Bandwidth of wired link	125 Kbps
Number of Terminal	30 node
TCP segment size	100 bytes
BS buffer size	50 packets
External network staying delay $d2$	0 ms

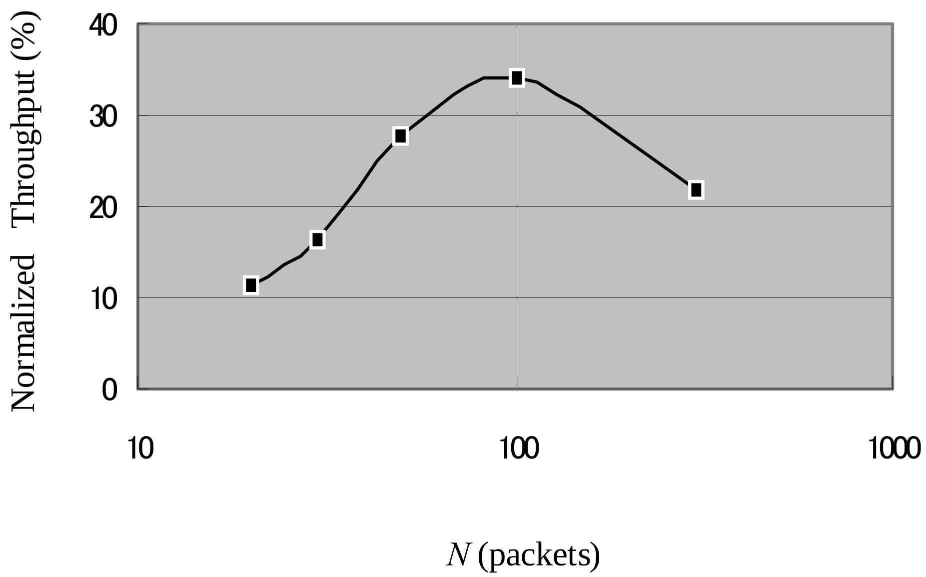


Figure 2.11 Slotted ALOHA Throughput

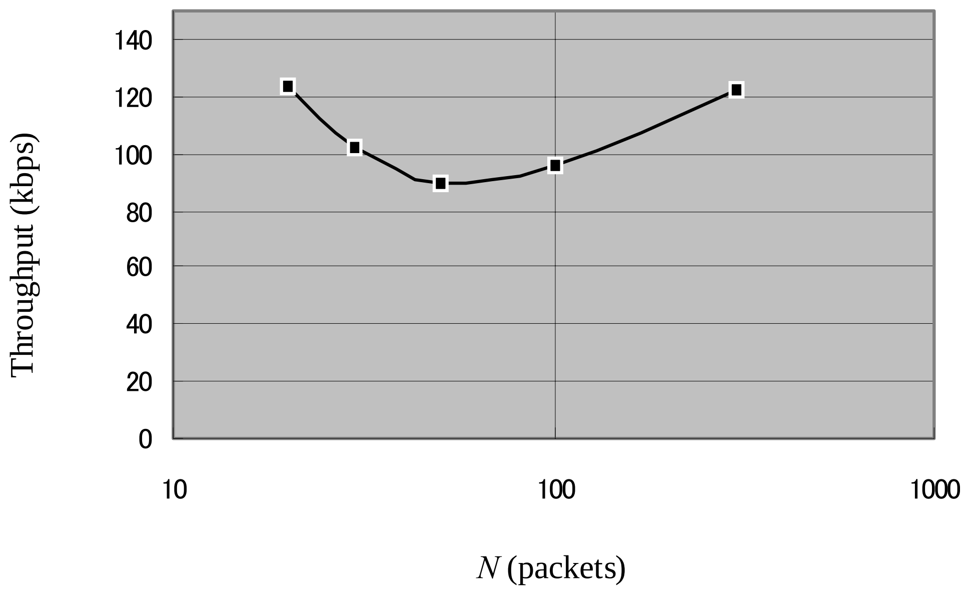


Figure 2.12 TCP Throughput

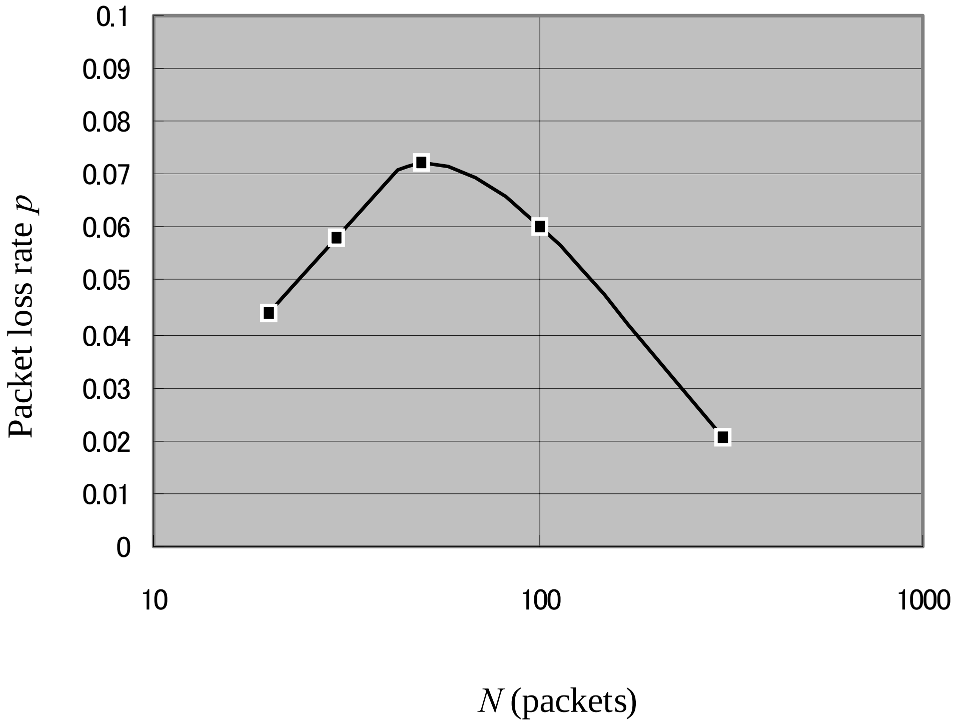


Figure 2.13 Packet Loss Rate at BS Buffer

2.4.2 Influence of Transmission Errors on Wireless Link

We next evaluate the influence of the transmission errors on the wireless link for TCP performance. We adopt Reed Solomon (127.117) or (127.87) as the FEC code. We assume the ARQ overhead to be 5% as mentioned in [18]. To choose the number of packet retransmissions by ARQ, we change it and compare the TCP performance. Table 2.5 shows parameters set that we used in evaluation. It is shown in Figure 2.14, where the horizontal axis Eb/N_0 is a noise level, which affects communication errors on the radio link. We show the relation between Eb/N_0 with p_{err}. Here, we assume that the physical layer uses QPSK modulation, which is used in CDMA method of IMT-2000. In the case of QPSK (Quadrature Phase Shift Keying) modulation, it is known that p_{err} is given by the following expression.

$$p_b = \frac{1}{2} erfc\left(\sqrt{\frac{E_b}{N_0}}\right) \tag{2.14}$$

Without ARQ or FEC, p_{err} is represented by using bit error rate p_b and packet length l as:

$$p_{err} = 1-(1-p_b)^l \tag{2.15}$$

When we use ARQ, p_{err} is represented by using bit error rate p_b, packet length l and retransmission number r as:

$$p_{err} = \{1-(1-p_b)^l\}^{r+1} \tag{2.16}$$

In the case of using FEC, p_{err} is represented by the following binomial distributed expression by using bit error rate p_b, packet length l and error correct ability c.

$$p_{err} = 1 - \sum_{i=0}^{c} \binom{l}{i} p_b^i (1 - p_b)^{l-i} \tag{2.17}$$

As shown in the figure, the effect of increasing the number of packet retransmissions at ARQ is very limited. Figure 2.14 shows the comparative results for ARQ and FEC. In the case of ARQ, the number of packet retransmissions is set to be one. As shown in the figure, FEC is more effective than ARQ to prevent TCP throughput degradation. It can also be observed that there exists the optimal FEC parameter to achieve the best performance dependent on the noise level. The last result shown in Figure 2.15 investigates the effect of packet transmission interval *N*. As indicated in the figure, the points that the TCP throughput is suddenly degraded are almost unchanged. Its reason is that the Retransmission Timeout algorithm does not work correctly because *RTT* changes are small in the current case.

Table 2.5 Evaluation Parameters (transmission errors on wireless link)

Bandwidth of radio link	2 Mbps
Bandwidth of wired link	125 Kbps
Terminals number	5 node
TCP segment size	100 bytes
Packet transmission interval *N*	70 packets
BS buffer size	50 packets
External network staying delay *d*2	0 ms

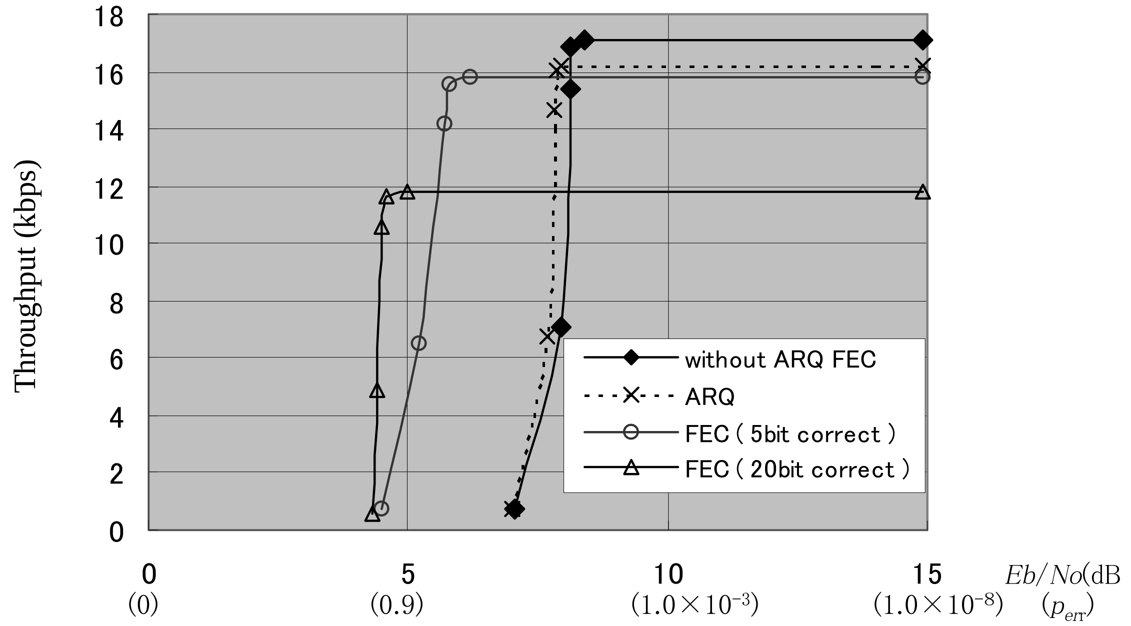


Figure 2.14 TCP Throughput Using ARQ, FEC

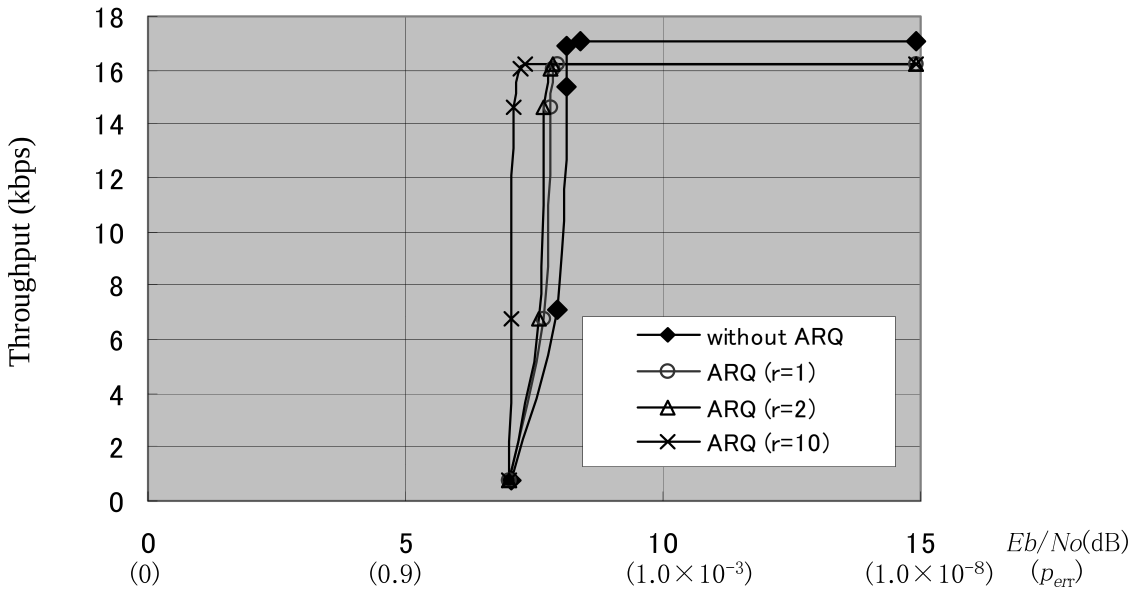


Figure 2.15 Influences of the Number of Packet Retransmission for TCP Throughput

2.5 Conclusion

In this chapter, we have constructed an analytical model for TCP throughput in wireless cellular networks. In our analytic model the influence of data link layer and transmission errors on the wireless link has been explicitly modeled, where the slotted ALOHA protocol is adopted as a data link layer. Finally, we have shown that the analysis results are in a good agreement with simulation results by comparing with simulation results.

Next, we have evaluated TCP throughput on the wireless cellular networks using our analytic model. We have shown that improving throughput at the data link layer level does not necessarily lead to the TCP throughput improvement. It is also shown that TCP throughput performance is much degraded by transmission errors on the wireless link. The use of FEC is effective to prevent TCP throughput degradation, but it is necessary to appropriately choose an error correction code with careful consideration on its overhead and the quality level of the wireless channel.

In our study, we have assumed that TCP segments are transmitted towards the wired terminal from the wireless terminal. As further work, we need to study in the case of the reverse direction, since the major application of the current Internet is to download the Web documents from the fixed servers to the wireless terminal.

Chapter 3

Improving TCP Performance on Wireless Cellular Networks by Adaptive FEC Combined with Explicit Loss Notification

Several approaches have been study for improving TCP performance on wireless cellular networks. TCP recognizes a congestion occurrence within the network by a packet loss, and performs congestion control by throttling the congestion window. Then, we encounter the problem that the packet losses due to the transmission errors cause unexpected degradation of TCP throughput in a wireless cellular network environment. Many approaches have been proposed to improve TCP throughput in the wireless cellular network. The split connection approach such as Indirect-TCP [8] is an early proposal in the field. It involves splitting each TCP connections between a sender and a receiver into two separate connections. Snoop Protocol [9] is an improved scheme of split connection approaches, and it retains end-to-end semantics; it uses Snoop Agent to cache the TCP segment and retransmits the segment only on the wireless link. ELN (Explicit Loss Notification) [6] is a more precise approach with a capability of distinguishing the packet loss due to congestion from the one by transmission errors.

Another approach to improving the TCP performance is to enhance the lower layer protocol, i.e., the data link layer protocol, by incorporating, e.g., ARQ (Automatic Repeat reQuest) and/or FEC (Forward Error Correction) [10]. FEC is a simplest solution to improve the bit error ratio seen by the higher layer protocol. However, it is inefficient if an error condition of the wireless channel varies greatly. Accordingly, an adaptive error correction scheme is proposed in [11] in order to compensate such a drawback of FEC. However, utilizing the adaptive error correction solely is not a realistic solution because it is complicated to measure BER (Bit Error Rate) for each bit or the unit of several bits and change the FEC code appropriately against irregular wireless errors. In reference [11], it is shown an adaptive error correction method for wireless LAN where the packet size and the degree of FEC redundancy are adaptively controlled according to the packet error rate. But it is needed specially-formatted UDP (User Datagram Protocol) [19] datagrams as probed packets to measure the packet error rate, and that these datagrams compress available bandwidth. In short, it has not been proposed Adaptive FEC scheme using no probed packets.

In this chapter, we propose a new adaptive FEC scheme combined with ELN, and we show a method for deciding threshold value to change adaptive FEC code, which is not clarified by [11]. In our method, transmission errors on the wireless link are measured at the packet level and the error status is notified the TCP sender with ELN. According to this information, an appropriate FEC code is selected. We first evaluate the TCP performance using Snoop Protocol, ELN and the fixed FEC, through which we find the appropriate FEC code against given BER. We will then show how the adaptive FEC can be realized using our solution, and also examine the appropriate observation period of measuring BER enough for the fading speed on noisy wireless link. We will finally evaluate our proposed method by Gilbert model [14] as wireless error model, and show that our method achieves better performance than the conventional fixed FEC.

3.1 Network Model

We use the network configuration shown in Figure 3.1 with IMT-2000 (International Mobile Telecommunications 2000) support [15] to investigate the TCP performance. Parameters are summarized in Table 3.1. Here, the bandwidth of the downlink is 384kbps for best effort service in the first deployment in Japan. In this configuration, since the major application of the current Internet is to download the Web documents from the fixed server to the wireless terminal, we assume that TCP segments are transmitted towards the wireless terminal from the wired terminal, and ACK segments are in the reverse direction. We consider that packet loss is mainly caused by the transmission error on the wireless link, and the packet loss due to buffer overflow is assumed to be negligible.

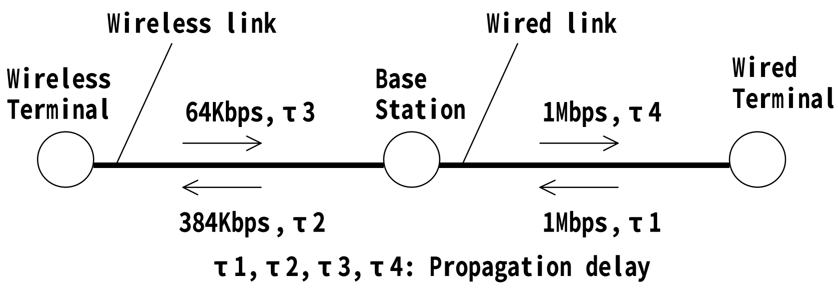


Figure 3.1 Network Model

Table 3.1 Network Parameters

TCP segment size	100 byte
ACK size	40 byte
Buffer size	50 kbyte
Propagation delay (τ 1, τ 2, τ 3, τ 4)	1ms

3.2 Performance Comparison for TCP on Wireless Cellular Networks

In this section, we first describe major schemes of improving for TCP throughput on wireless cellular network, and then evaluate TCP throughput for major schemes. TCP provides reliable end-to-end data communication using the following two main congestion control mechanisms. The one is Fast Retransmit and Fast Recovery mechanisms, which throttle the congestion window size to half, if the TCP sender detects triple duplicate ACKs continuously. The other is a Retransmission Timeout mechanism, which draws back the congestion window size to 1 MSS (Maximum Segment Size), if the ACK for the TCP segment is not received before retransmission timeout timer expiration.

Those congestion control mechanisms perform well on wired links since most of packet losses occur due to congestion. However, on wireless links, it does not because the packet loss may occur due to transmission errors and TCP cannot distinguish the packet loss due to congestion and transmission errors. Therefore, it is well known that the packet losses due to transmission errors cause unexpected degradation of TCP throughput in a wireless cellular network environment.

Many approaches have been proposed to improve TCP throughput in the wireless cellular network. These are summarized in the IETF (Internet Engineering Task Force) [20]. Here, we describe Snoop Protocol, ELN, and the link-layer scheme as major schemes. Though above approaches is due to reduce degradation TCP throughput by transmission errors, we comment on the relation with standardization aspects such as fast recovery and fast retransmission, and slow start [21], selective acknowledgements [22], and ECN (Explicit Congestion Notification) [23]. We study based on Reno version, which is already implemented fast retransmission, fast recovery, and slow start mechanisms. Since Reno is current de facto standard for TCP implementations. Selective acknowledgements scheme, the sender needs to retransmit only the segments that have actually been lost, is a little effective. Since this scheme is more effective when the window size is larger, but the window size is usually small due to transmission errors in the wireless cellular network. ECN, the sender controls congestion explicitly using congestion mark received from the router, but in ECN scheme, sender cannot recognize the transmission error on the wireless link.

For above reasons, we focus on Snoop Protocol, ELN and the data link layer scheme.

Snoop Protocol

Snoop Protocol is an improved scheme of split connection approaches. It retains end-to-end semantics of TCP, such that it uses Snoop Agent to cache the TCP segment and retransmits the segment only on the wireless link. Snoop Protocol performs the following sequence in the case of the wired terminal being the TCP sender.

(1) The TCP segments are cached at the BS
(2) The TCP segments are retransmitted at the BS, if packet loss is detected on the wireless link.

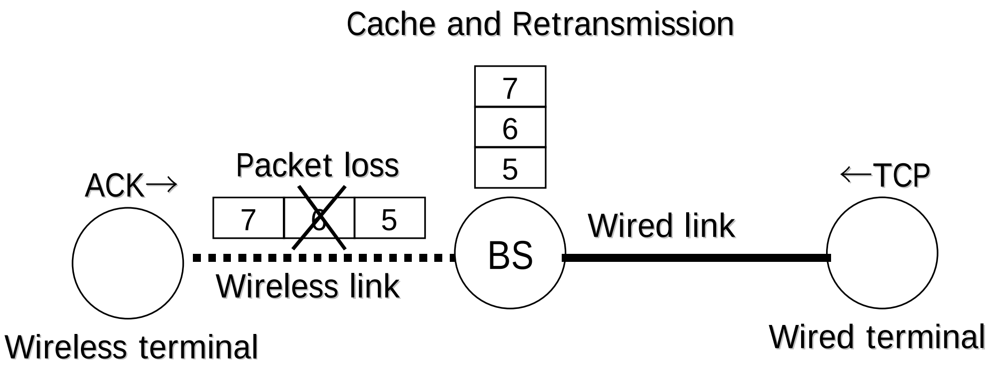


Figure 3.2 Snoop Protocol

ELN

ELN is an approach that enables distinguishing the packet loss due to congestion from transmission errors. If the packet loss occurs due to transmission errors, the TCP sender takes appropriate actions. That is, ELN performs the following sequence in the case of the wired terminal being the TCP sender.

(1) The TCP sequence numbers are cached at the base station.
(2) The ACK packet is attached "ELN bit active" at the base station, if the packet, of which sequence number is cached, is lost.
(3) The TCP sender takes an appropriate action when receiving "ELN bit active". One typical realization is that the TCP sender does not perform congestion control.

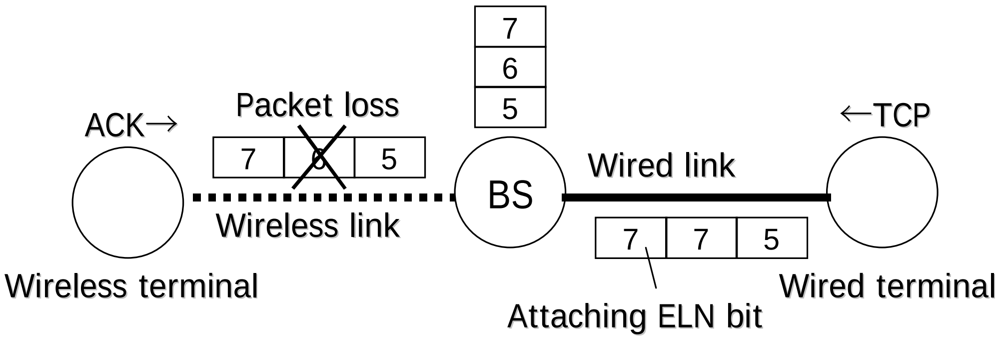


Figure 3.3 ELN

The Data Link layer scheme

Enhancements of the link-layer protocol include ARQ and FEC. FEC is a simplest solution to improve the bit error ratio seen by the higher layer protocol. However, it is inefficient if an error condition of the radio channel varies greatly. Accordingly, an adaptive error correction scheme is proposed in [11] in order to compensate such a drawback of FEC. However, utilizing the adaptive error correction solely is not a realistic solution because it is complicated to measure BER for each bit or the unit of several bits and change the FEC code appropriately against irregular wireless errors.

We evaluate TCP throughput by using the TCP Reno version, which is a current de facto standard for TCP implementations. For enhancement of the TCP performance, we consider the following three schemes.

(1)ELN, which distinguishes between packet loss due to congestion and others.
(2) Snoop+ELN, which is a combination of Snoop and ELN.
(3) FEC as an enhancement of the data link layer protocol.

We used the ns-2 simulator [17]. Simulation results are shown in Figure 3.4, where the horizontal axis shows the bit error rate on the wireless link. We adopt BCH (127, 92) as the FEC code. It can be observed in the figure that FEC is most effective to prevent TCP throughput degradation in the range where BER is greater than 0.0001. On the other hand, in the lower BER region, TCP throughput of FEC becomes smaller than other schemes due to its overhead and Snoop+ELN achieves the best performance.

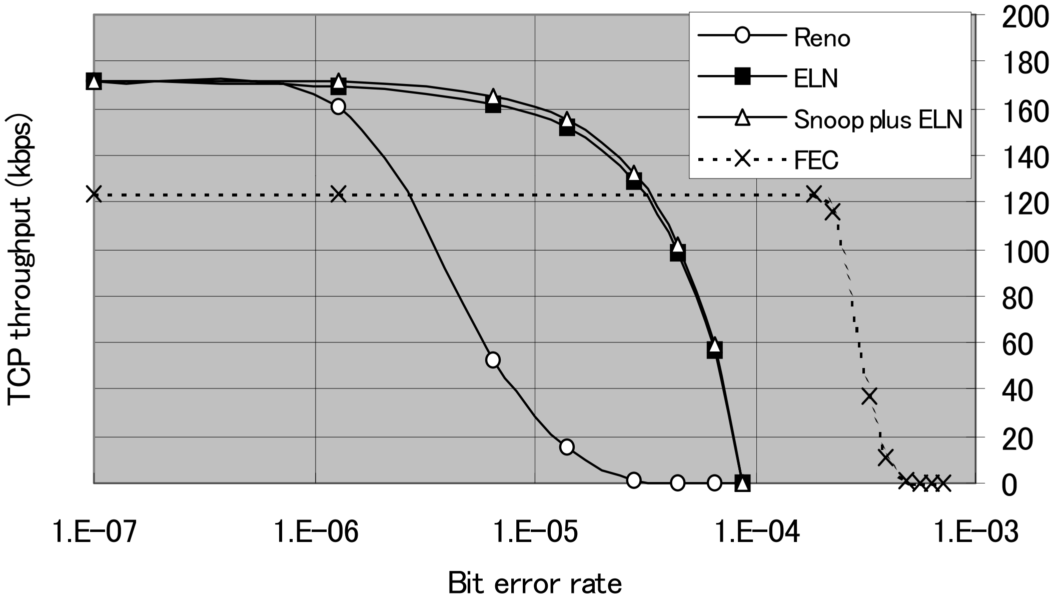


Figure 3.4 Comparison of TCP Throughput

3.3 Adaptive FEC Combined with Snoop+ELN

In Figure 3.4, we present the TCP throughput of Snoop+ELN combined with FEC. The FEC code is changed as BCH (127, 92) as 5-bit correct code, BCH (127, 64) as 10-bit correct code and BCH (127, 36) as 15-bit correct code. The figure clearly shows an existence of the optimal FEC parameter to achieve the best performance dependent on BER. In short, an adaptive FEC combined with Snoop+ELN scheme is more effective in order to prevent TCP throughput degradation. In this paper, we propose a new adaptive FEC scheme combined with ELN, and we show a method for deciding threshold value to change adaptive FEC code. This is not clarified by [11]. In the current Internet, network congestion has been solved to improve network infrastructures. We consider that packet loss is mainly caused by the transmission error on the wireless link, and the packet loss due to network congestion is assumed to be negligible. In this assumption, TCP senders recognize the transmission error on the wireless link by ELN, and perform no congestion control.

Of course, we should take account of the fact that the packet error rate is dynamically changed according to the condition of the wireless link.

In the adaptive FEC scheme, it is important how FEC code should be changed according to the packet error rate. In this chapter, we adopt a reactive-based adaptive algorism by observed errors. In the reactive-based adaptive algorism, it is important that the algorism is robust in the sense that it is not influenced by the sensitive errors. The authors in [11] proposed a technique of adaptively changing FEC; the FEC code is changed when two or more segments among ten segments encounter the transmission error. Our proposed method is based on that idea. Based on the above numerical results, we show a method for changing the FEC capability according to p_{err} obtained by ELN as shown in Figure 4.5. Here, the current FEC code is changed to the more error correctable code when p_{err} exceeds the threshold value. Otherwise, it is changed to the less overhead code. In Figure 4.5, the symbol of the A, B, C, and D is the threshold value of p_{err} and correspond to the same symbol in Figure 4.6.

We finally examine the time period of measuring p_{err}. The smaller value of the time period is expected so that it can be adapted to the time changing packet loss rate. However, it must be large enough for the fading speed. The fading speed is given by f_d (Doppler frequency) [24] as:

$$f_d = \frac{v\, f_c}{c} \tag{3.1}$$

where f_c is the carrier frequency, v is the vehicle speed, and c is the speed of light. For the 900-MHz carrier frequency, the above formula yields a Doppler frequency of 3Hz/m/sec. Therefore, for the high-speed wireless transmission (e.g., in the order of Mbps), the fading speed can be viewed as a roughly constant value for the time period of measuring p_{err} (e.g., 10 sampling with the length of 1000-byte segment).

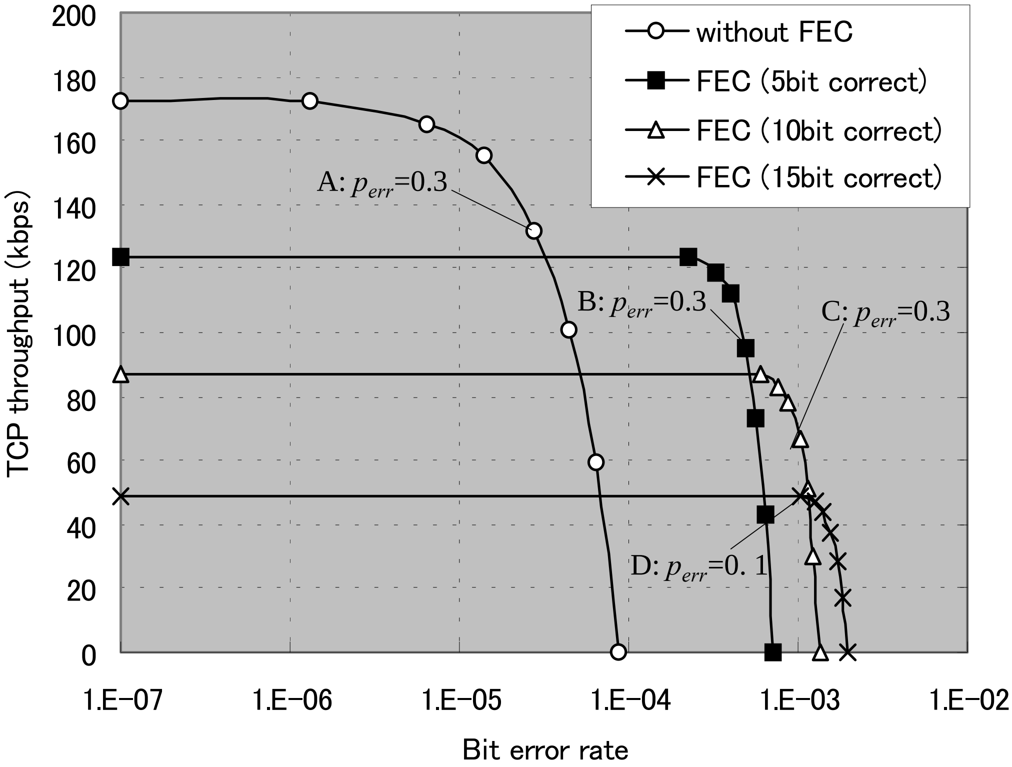
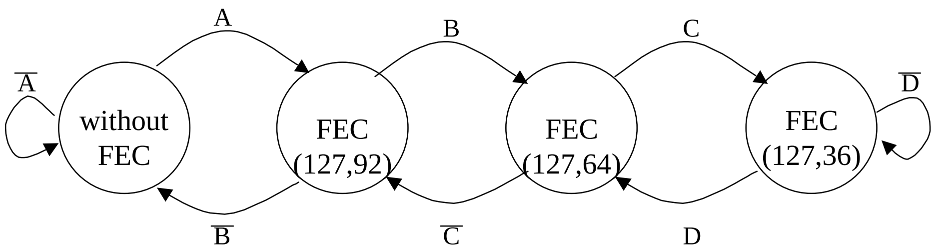


Figure 3.5 TCP Throughput by Adaptive FEC



A: p_{err}>0.3 (In the case of two or more packets in a window of ten)
B: p_{err}> 0.3 (In the case of two or more packets in a window of ten)
C: p_{err}> 0.3 (In the case of two or more packets in a window of ten)
D: p_{err}<0.1 (In the case of two or more packets in a window of ten)

Figure 3.6 Adaptive FEC Combined with Snoop+ELN

3.4 Evaluation of Adaptive FEC Combined with Snoop+ELN

In this section, we evaluate our proposed method by the Gilbert model as a wireless error model. It has been widely used to model the noisy link with time-varying errors. In the Gilbert model, two states of "Error-free" and "Error" are expressed in terms of transition probabilities p_{GE} and p_{EG}., and average error-free length L_G and error length L_E. See Figure 3.7.

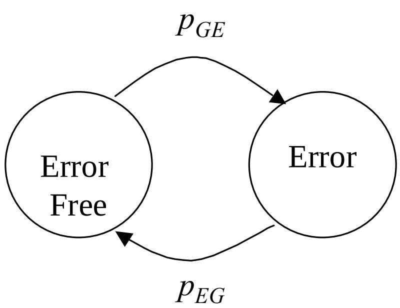


Figure 3.7. Gilbert Model

$$p_{GE} = \frac{1}{L_G} \tag{3.2}$$

$$p_{EG} = \frac{1}{L_E} \tag{3.3}$$

The Gilbert model is a two-stare Markov model and each state is memory-less. Recalling that the geometric distribution is a discrete equivalent of the exponential distribution [25], we determine the length $G(p)$ of staying in each state as follows.

$$G(p) = \frac{\ln(u)}{\ln(1-p)} \tag{3.4}$$

where u is a random number uniformly distributed from 0 to 1, and p is the leaving probability from the state. Using the above model, we generate wireless errors and simulate whether lost or success for each sending packet. See the flow chart illustrated in Figure 3.8. In simulation, the number of sending packets is 50,000, and packets length is fixed at 800 bits, which is commonly seen in the mobile phone data transfer services. The parameter set of the Gilbert model is shown in Table 3.2. Here, we assume that bit errors in each packet occur randomly. We show the simulation results in Table 3.3 in the case of a rather lower value of BER (p_b = 0.00015, p_{err} = 0.090), and in Table 3.4 in the case of the higher BER (p_b = 0.00075, p_{err} = 0.445), respectively. We also show the values obtained through the simple analysis for the case without FEC and for the one with fixed FEC.

Table 3.2 A Set of Parameters for Gilbert Model

	p_{GE}	p_{EG}
lower BER case	0.006	0.05
higer BER case	0.006	0.007

Here, we show how the numbers of overhead packets and successfully received packets are calculated in the case of without FEC and fixed FEC.

[The Case without FEC]

By assuming that the bit error in each packet occurs randomly, packet error rate p_{err} is represented by using the bit error rate p_b and the packet length l as:

$$p_{err} = 1 - (1 - p_b)^l \tag{3.5}$$

Using p_{err}, the number of lost packets, P_{loss}, is determined as:

$$P_{loss} = p_{err} P_{send} \tag{3.6}$$

Similarly, the number of successfully received packets, P_{rcv}, is given by:

$$P_{rcv} = P_{send} - P_{loss} \tag{3.7}$$

[The Case of Fixed FEC]

If the bit errors in each packet occur randomly, p_{err} is represented by the following binomial distributed expression by using bit error rate p_b, packet length l and error correct ability c.

$$p_{err}(l,c) = 1 - \sum_{i=0}^{c} \binom{l}{i} p_b^i (1-p_b)^{l-i} \tag{3.8}$$

Using packet length l and data length d, overhead by fixed FEC is calculated by following expression:

$$H = \frac{d}{l} \tag{3.9}$$

Using p_{err} and overhead H, the numbers of lost packets P_{loss}, overhead packets P_{head} , and successfully received packets P_{rcv} are obtained by the following expressions:

$$P_{loss} = p_{err} P_{send} \tag{3.10}$$

$$P_{head} = HP_{send} \tag{3.11}$$

$$P_{rcv} = P_{send} - P_{loss} - P_{head} \tag{3.12}$$

In Tables 3.3 and 3.4, the second column shows the number of lost packets by wireless packet errors and the third column does the number of received packets. The fourth column shows the number of overhead packets, which is the case with fixed FEC obtained by Eq. (3.9). The final column shows the number of successfully received packets, in which we omit the number of overhead packets from the number of totally received packets. We can observe that our adaptive FEC can achieve the best performance. That is, by using the adaptive FEC, the successfully received packets are improved by 2% in Table 3.3, and by 10% in Table 3.4. It is because our adaptive FEC encounters less packet losses than without FEC, and needs less overhead packets than the fixed FEC.

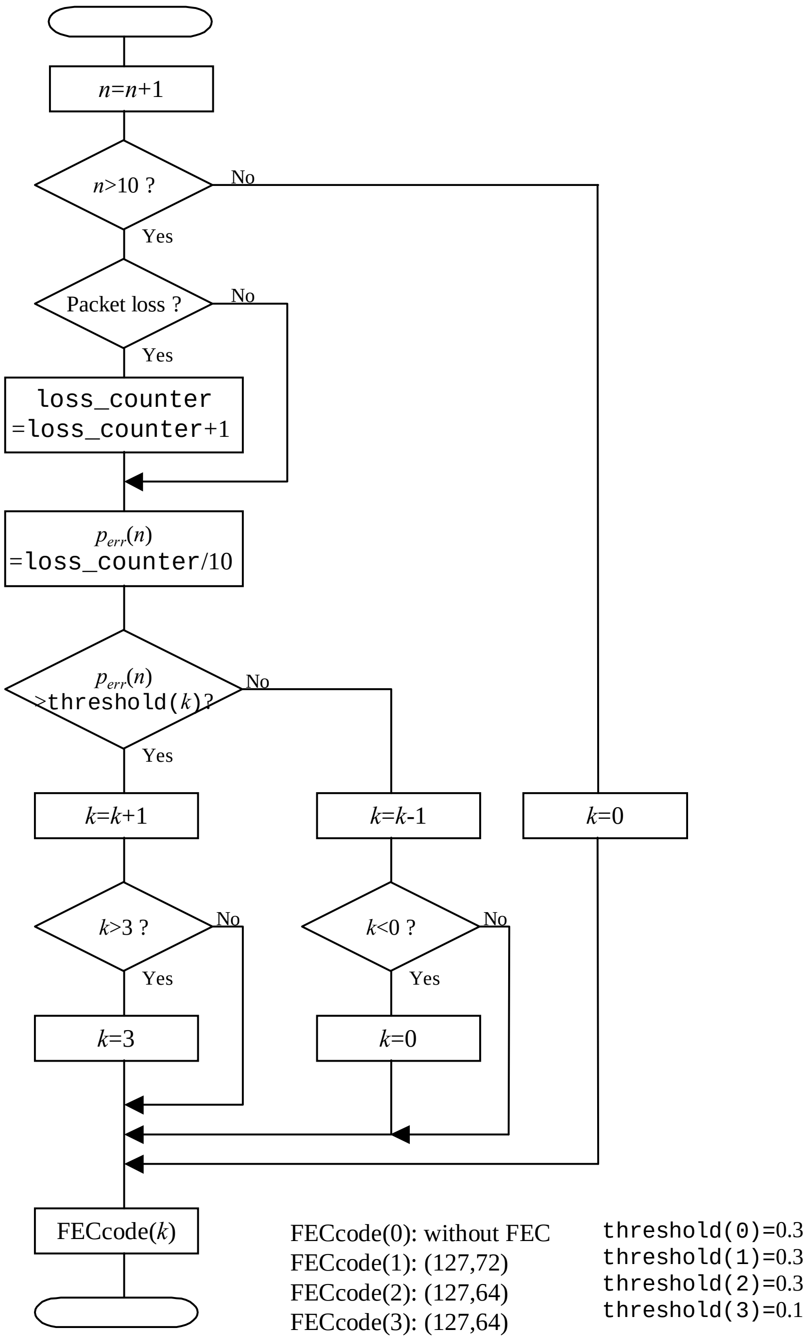


Figure 3.8 Adaptive FEC Flow

Table 3.3 Comparison of Received Packets (lower BER case)

	lost packets	received packets	overhead packets	received packets (available)
without FEC	4461	45539	0	45539
FEC(127,94)	35	49965	12983	36982
FEC(127,64)	0	50000	24803	25197
FEC(127,36)	0	50000	35827	14173
Adptive FEC	2292	47708	1002	46706

Table 3.4 Comparison of Received Packets (higher BER case)

	lost packets	received packets	overhead packets	received packets (available)
without FEC	22242	27758	0	27758
FEC(127,94)	15903	34097	8860	25237
FEC(127,64)	934	49066	24340	24726
FEC(127,36)	11	49989	35819	14170
Adptive FEC	8213	41787	9223	32564

3.5 Conclusion

In this chapter, we have proposed a new adaptive FEC scheme combined with ELN. In our method, transmission errors on the wireless link are measured at the packet level and the error status is notified the TCP sender with ELN. According to this information, an appropriate FEC code is selected. We have evaluated the TCP performance using Snoop Protocol, ELN and the fixed FEC, through which we found the appropriate FEC code against given BER. We have also shown how the adaptive FEC can be realized using our solution. We have finally evaluated our proposed method by the Gilbert model as the wireless error model, and shown that our method achieves better performance than conventional fixed FEC. As a next step, we need to implement our adaptive FEC scheme and demonstrate its effectiveness.

Chapter 4

Improving TCP Performance in Wireless Cellular Networks by Acknowledgement Control

It is well known that the packet losses due to the transmission errors cause unexpected degradation of TCP throughput in a wireless cellular network environment. Especially when ACK is lost, in spite of having sent the data segment correctly, the performance of TCP deteriorates terribly.

Many researches have been presented to improve TCP throughput in the wireless cellular network. For example, ELN (Explicit Loss Notification) [6] controls the window size appropriately by observing packet losses on the radio link. WTCP [7] changes window based congestion control to rate based congestion control. However, those solutions have not been realized because those require major changes to network infrastructures. ELN needs to change both the BSs (Base Stations) and wireless terminals to observe packet losses on the wireless link. WTCP needs to change both clients and servers for the new TCP congestion control algorithm.

In this section, we propose a method for improving the performance of TCP by a minor change in treating the ACK packets. In our proposed method, the TCP receiver does not send one ACK, but multiple ACKs when the packet loss probability exceeds the predefined threshold. Since the major application of the current Internet is to download the Web documents from the fixed servers to the wireless terminal, it only requires changes at the wireless terminal side for performance improvement. Wireless terminals monitor a packet loss probability, and estimate the error probability of the wireless channel. Then, TCP becomes robust against wireless link errors.

We present an analytical method, and show that two ACKs for each TCP packet are sufficient to improve the performance.

4.1 Network Model

In this section, we investigate the performance of TCP for the network configuration shown in Figure 4.1, with IMT-2000 (International Mobile Telecommunications 2000) [15] support. Parameters are summarized in Table 4.1. In this model, it is assumed that a TCP segment is transmitted towards the wireless terminal from the wired terminals (or servers). We mainly consider packet losses due to buffer overflow at the wireless terminals and transmission errors on the wireless link. We assume the TCP Reno version, which is a major implementation in the current TCP code.

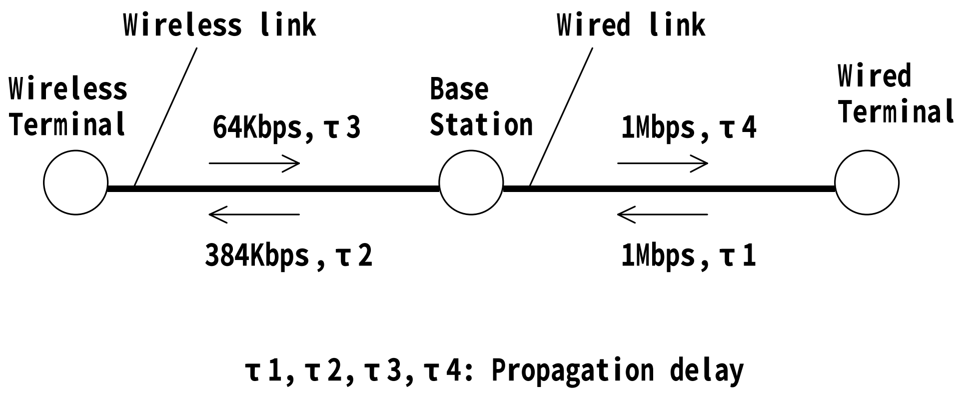


Figure 4.1 Network Model

Table 4.1 Network parameters

TCP segment size	100 byte
ACK size	40 byte
Buffer size (Wireless Terminal, BS, Wired Terminal)	50 kbyte
Propagation delay (*τ1,τ2,τ3,τ4*)	1 ms

4.2 Proposed Method

TCP provides reliable end-to-end data communication using the following two main congestion control mechanisms. The one is Fast Retransmit and Fast Recovery mechanisms, which throttle the congestion window size to half, if the TCP sender detects triple duplicate ACKs. The other is a Retransmission Timeout mechanism, which draws back the congestion window size to 1 MSS (Maximum Segment Size), if the ACK for the TCP segment is not received before retransmission timeout timer expiration.

Those congestion control mechanisms perform well on wired links since the most of packet losses occurs due to congestion. However, on wireless links, it does not because the packet loss may occur due to transmission errors and TCP cannot distinguish the packet loss due to congestion and transmission errors. Therefore, it is well known that the packet losses due to transmission errors cause unexpected degradation of TCP throughput in a wireless cellular network environment.

In our proposed method, to make TCP robust against ACK losses caused by transmission errors, the wireless terminal sends multiple ACKs when the packet error probability of the wireless network p_{err} exceeds the threshold value. See Figure 4.2. When the wired terminal sends a TCP segment and the wireless terminal returns multiple ACKs, the following three cases are considered.

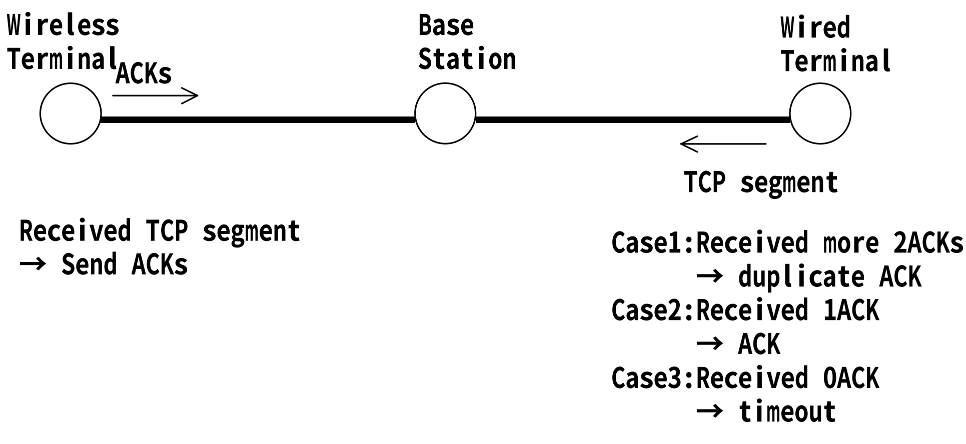


Figure 4.2 Control of the Number of Acknowledgment

(1) The wired terminal receives one ACK, and it continues the normal operation.
(2) The wired terminal receives more than two ACKs and it recognizes the second and proceeding ACKs as duplicate ACKs. It invokes the fast recovery in this case.
(3) Only if all of multiple ACKs are lost, the timeout occurs at the wired terminal.

While transmitting two or more ACKs decreases the loss of ACK due to the transmission error on the wireless link, duplicate ACK may be received at the wired terminal as in the case (2) above. Thus, it is effective in transmitting two or more ACKs in the range of the high transmission error probability. However, if the error probability is low, the TCP performance might deteriorate by the fast recovery. Therefore, it is necessary to determine the appropriate number of ACKs according to the quality of wireless channel. In the proceeding subsections, we explain how to estimate the packet error probability in the wireless link, and the method of deriving the appropriate number of ACKs analytically. We last note that even when the wired network actually falls into congestion, the multiple ACKs do not affect the proper congestion control operation.

4.3 Estimation on Packet Loss Rate on Wireless Link

As shown in Figure 4.3, we define p_{err_uplink}, $p_{err\_downlink}$, p_{buff} and p. p_{err_uplink} represents the packet error probability on the uplink. $p_{err\_downlink}$ represents the packet error probability on the downlink. p_{buff} represents the packet loss probability at the bottleneck buffer. In our network model, as shown in Figure 4.1, the bottleneck link corresponds to uplink, so the bottleneck buffer is in the wireless terminal. p is packet loss probability, which can be monitored as three duplicate ACK at TCP sender (wired terminal). Here, we assumed that p is informed to wireless terminal from wired terminal.

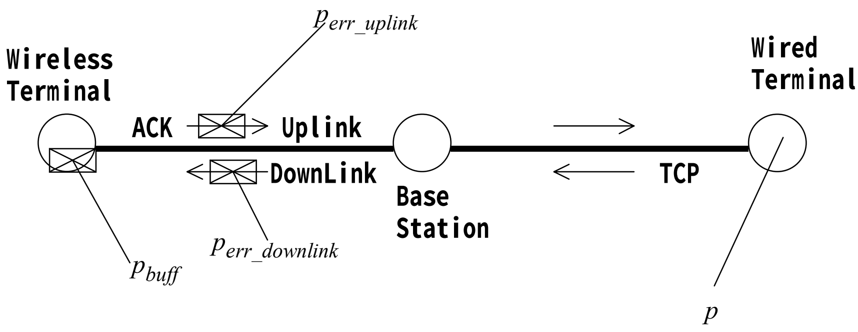


Figure 4.3 Packet Loss Model

Packet loss on the radio link and the buffer overflows at the bottleneck buffer take place independently. Then the total packet error probability observed at wireless terminal, p is expressed as follows:

$$p = 1 - (1 - p_{err\_uplink})(1 - p_{buff})(1 - p_{err\_downlink}) \tag{4.1}$$

When the error probability of uplink is represented p_{err}, downlink is assumed to be equal to kp_{err}, where k is the ratio of the packet length uplink versus downlink. We have

$$p_{err\_uplink} = p_{err} \tag{4.2}$$

$$p_{err\_downlink} = kp_{err} \tag{4.3}$$

Then we can calculate p_{err} by Eq. (4.1), (4.2), and (4.3) using parameter p, p_{buff} , k. Here, p_{buff} can be observed at the wireless terminal because the wireless terminal is bottleneck in the current network. Otherwise, p_{buff} can be given as a fixed parameter if the router employs RED [12].
In monitoring p, at the wireless terminal, its change would become too large

if the time unit of measurement is large. Conversely, when the time unit of measurement is small, more memories at the wireless terminal are necessary. We therefore introduce a method of calculating p by the moving average method, following an estimation method of *RTT* (Round Trip Time) values in TCP [13]. The estimation probability is given by

$$p = \frac{7}{8} p\_now + \frac{1}{8} p\_old \tag{4.4}$$

where p is the estimation value, p_now is the current value, and p_old is the last value.

Next, we performed the simulation which the number of connections of TCP is changed dynamically. In this section, the simulation has been done in order to check the effect of this estimation method. We would show the evaluation result of TCP performance by means of simulation in Section 4.5. Here, we adopt the model of Figure 4.1 as the network under consideration and applied the value of Table 4.1 as for the parameter. The result at the time of changing the number of TCP connections from one to five dynamically is shown in Figure 4.4. In this figure, asterisk points show the observation values of p for every per second, and the line represents the result of the moving average by the formula (4.2). In Figure 4.4, the curve of moving average is in a good agreement with each p, the curve of moving average is in a good agreement with each p. We have also shown influence of initial value of p in Figure 4.5, when we change the value 0 to 0.3. There are no differences due to initial value of p in the case of over ten times sampling as this figure is shown.

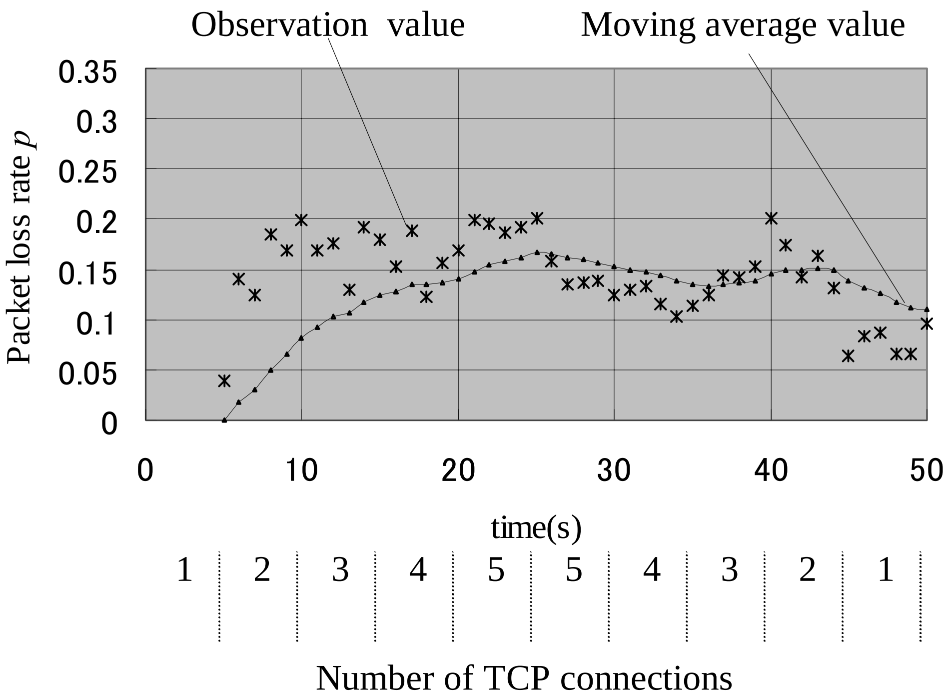


Figure 4.4 Calculating p by Moving Average Method

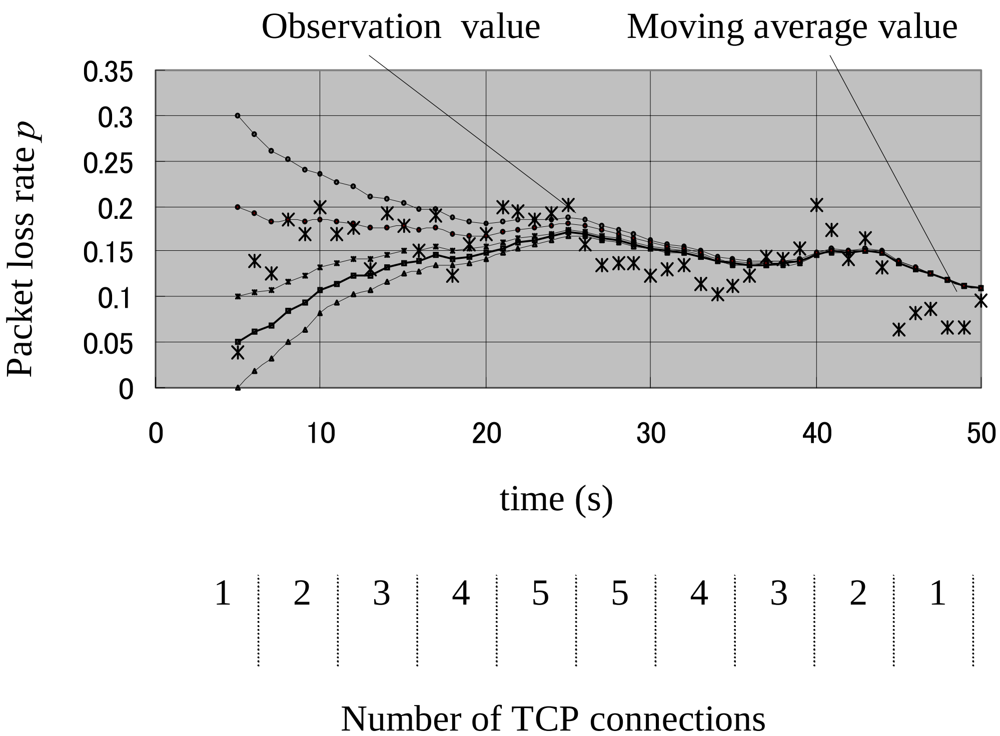


Figure 4.5 Calculating p by Moving Average Method
(initial value)

4.4 Derivation of Appropriate Number of ACKs

Since TCP recognizes packet loss occurrences due to buffer overflow when the TCP sender has received more than three duplicate ACKs, $p(n)$ can be determined as the probability of receiving more than three duplicate ACKs.

$$
\begin{aligned}
p(n) = {} & P_{dACK0} P_{rcv3}(n) + P_{dACK1} P_{rcv2}(n) \\
& + P_{dACK2} P_{rcv1}(n) + P_{dACK3} \\
n = {} & 1,2,3...
\end{aligned}
\tag{4.5}
$$

where the first term of Eq. (4.5) represents the probability of receiving more than three duplicate ACKs, in the case of receiving three or more ACKs out of n ACKs, from the state of receiving no duplicate ACK. The second term represents the probability of receiving more than three duplicate ACKs, in the case of receiving two or more ACKs out of n ACKs, from the state of receiving one duplicate ACK. The third term represents the probability of receiving more than three duplicate ACKs, in the case of receiving one or more ACKs out of n ACKs, from the state of receiving two duplicate ACK. The final term represents the probability in the state of receiving already more than three duplicate ACKs. Thus, $p_{buff}(n)$ can be represented by the sum of above four cases. Here, P_{dACK0}, P_{dACK1}, P_{dACK2} and P_{dACK3} are probabilities in the state of receiving zero, one, two, and more than three duplicate ACKs. These are represented by the following expressions;

Probability in the state of receiving no duplicate ACK:

By assuming p_{buff} occur independently from the former state, the cubic root of p_{buff} represents the probability in the state of receiving more than one ACK. Since P_{dACK0} is the inverse probability of the receiving more than one ACK, it can be determined as:

$$P_{dACK0} = 1 - \sqrt[3]{p_{buff}} \tag{4.6}$$

Probability in the state of receiving one duplicate ACK:

The cubic root of p_{buff} represents the probability in the state of receiving more than one ACK, and then its square represents the probability in the state of receiving more than two duplicate ACKs. Since P_{dACK1} is represented as the difference, it can be determined as:

$$P_{dACK1} = \sqrt[3]{p_{buff}} - (\sqrt[3]{p_{buff}})^2 \tag{4.7}$$

Probability in the state of receiving two duplicate ACKs:

The first item on Eq. (4.8) represents the probability in the state of receiving more than two duplicate ACKs, and the second item represents the probability in the state of receiving more than three duplicate ACKs. Since P_{dACK2} is represented as the difference, it can be determined as:

$$P_{dACK2} = (\sqrt[3]{p_{buff}})^2 - p_{buff} \tag{4.8}$$

Probability in the state of receiving more than three duplicate ACKs:

p_{buff} is the probability of receiving more than three duplicate ACKs.

$$P_{dACK3} = p_{buff} \tag{4.9}$$

Furthermore, $P_{rcv1}(n)$, $P_{rcv2}(n)$ and $P_{rcv3}(n)$ are probabilities of receiving ACKs out of n ACKs. These are calculated from the binomial distribution with a generating probability p_{err}.

Probability of receiving one or more ACKs out of *n* ACKs:

It can be determined as the binomial distribution using p_{err} as the lost probability and (1- p_{err}) as the receiving probability, which is represented the sum of the probability of receiving from one to n ACKs out of n ACKs.

$$P_{rcv1}(n) = \sum_{i=1}^{n} {}_n\mathrm{C}_i (1 - p_{err})^i p_{err}^{n-i} \tag{4.10}$$

Probability of receiving two or more ACKs out of *n* ACKs:

In the same way, it is represented the sum of the probability of receiving from two to n ACKs out of n ACKs.

$$P_{rcv2}(n) = \sum_{i=2}^{n} {}_n\mathrm{C}_i (1 - p_{err})^i p_{err}^{n-i} \tag{4.11}$$

Probability of receiving three or more ACKs out of *n* ACKs:

Likewise, it is also represented the sum of the probability of receiving from three to n ACKs out of n ACKs.

$$P_{rcv3}(n) = \sum_{i=3}^{n} {}_n\mathrm{C}_i (1 - p_{err})^i p_{err}^{n-i} \tag{4.12}$$

Thus, we can determine $p(n)$ by using Eqs.(4.3) through (4.12) by parameters n, p_{err} and p_{buff}. Next, we derive TCP throughput for each number of ACKs analytically using the formula shown in [16]. In our analysis, it is characterized by three parameters RTT, To (Timeout) and p as follows;

$$S_{TCP} = \frac{1}{RTT\sqrt{\frac{2bp}{3}} + To\min(1,3\sqrt{\frac{3bp}{8}})p(1+32p^2)} \tag{4.13}$$

where b is a delayed ACK parameter. Normally, b = 2.

Figures 4.7 and 4.8 plot the packet loss probability and TCP throughput when changing the packet error probability on the radio link, p_{err}. We set p_{buff} = 0.01, RTT = 100 ms and To = 400 ms. As shown in Figure 4.8, transmission of two ACKs can achieve the best performance in the range of 0.1 < p_{err} < 0.47. The reason is that as shown in Figure 4.7, transmission of two ACKs results in smallest p in the above-mentioned range.

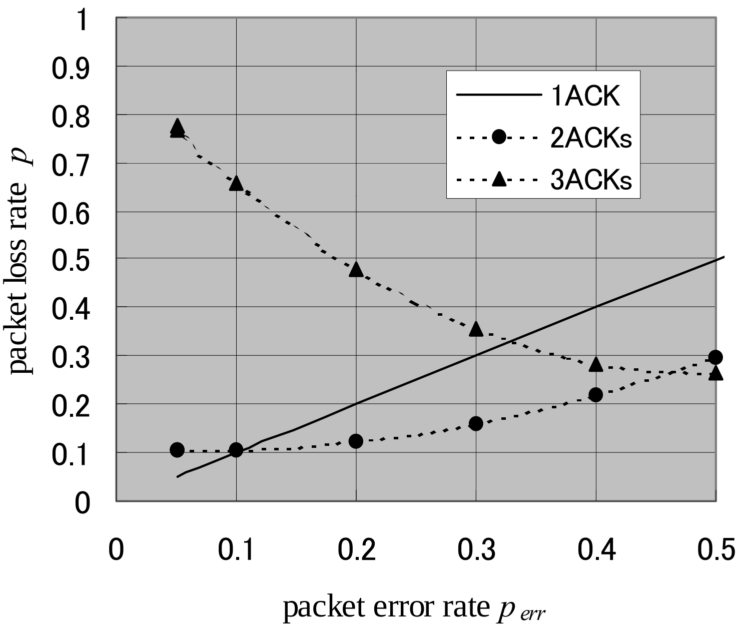


Figure 4.6 Packet Loss Rate (analysis)

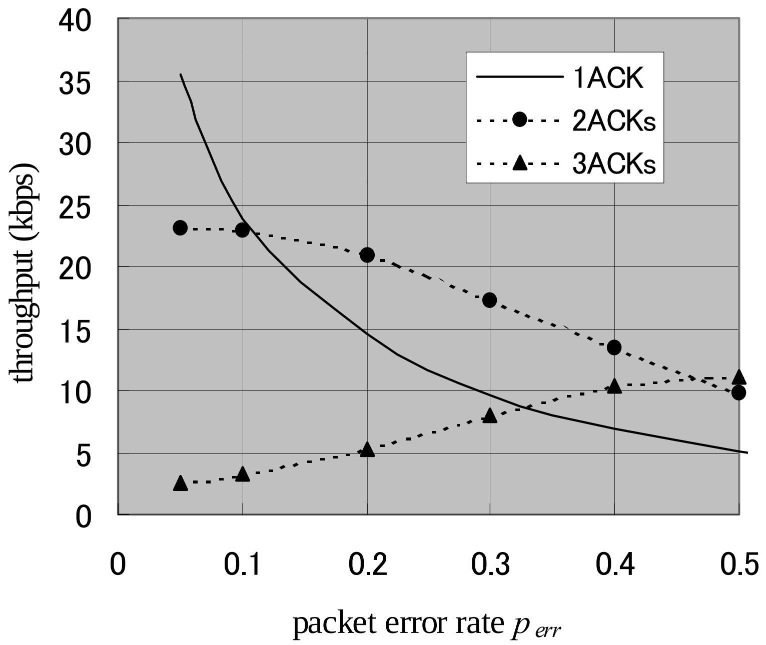


Figure 4.7 TCP Throughput (analysis)

4.5 Simulation Results

In this section, we evaluate our proposed method by ns-2 simulator [17]. In our simulation, it is not concerned to Eqs. (4.1) - (4.13), but it is simulated that each TCP segment is transmitted towards the wireless terminal from the wired terminals in Figure 4.1 as the network model. See Section 4.2 for detail. Here, a set of parameters is summarized in Table 4.1. In Table 4.1, we assume that the TCP segment size is 100 bytes and the ACK packet size is 40 bytes, which is commonly seen in the mobile phone data transfer services. And the buffer size and the propagation delay are a set of default values in ns-2 simulator.

First, we show the simulation results in Table 4.2 about the case where the number of wireless terminals is one. From this result, it is shown that transmission of two ACKs can improve the TCP throughput where the range of p_{err} is 0.05 or more.Then we show the window size appearance of TCP in the case of our method being effective (p_{err} = 0.1) in Figure 4.8. As shown in the figure, our proposed method keeps larger window sizes than the original method. On the other hand, as shown in Figure 4.9, the window size is not larger than the original one with p_{err} = 0.01. It is because transmissions of two ACKs per each data packet causes much duplicate ACKs and the TCP sender recognizes it as packet loss to invoke the fast recovery.

Moreover, we show the result in Table 4.3 about the case of the number of terminals is five. It is represented to adding four terminals to the Base Station in Figure 4.1. This result shows that transmission of two ACKs can improve the throughput with p_{err} larger than 0.1. It turns out that the effective range is decreasing as compared with the case where the number of terminals is one. This is because transmissions of two ACKs per each data packet causes much more duplicate ACKs due to increase in the number of wireless terminals. Thus, it becomes important to choose the number of ACKs according to the value of p_{err}. As shown in the analysis, our method can easily incorporate such a control method by monitoring p and observing p_{buff} at the wireless terminal in the current network model.

Table 4.2 TCP Throughput (1 node)

perr	1ACK (Kbps)	2ACK (Kbps)
0	214.1	211.7
0.01	202.9	200.2
0.05	68.55	74.55
0.1	20.42	20.78
0.2	2.058	2.92
0.3	0.388	0.55
0.4	0.114	0.152
0.5	0.044	0.073

Table 4.3 TCP Throughput (5 node)

perr	1ACK (Kbps)	2ACK (Kbps)
0	292.8	286.2
0.01	259	247.6
0.05	202.2	192.3
0.1	89.25	105.45
0.2	18.86	20.8
0.3	1.845	2.93
0.4	0.486	0.56
0.5	0.11	0.126

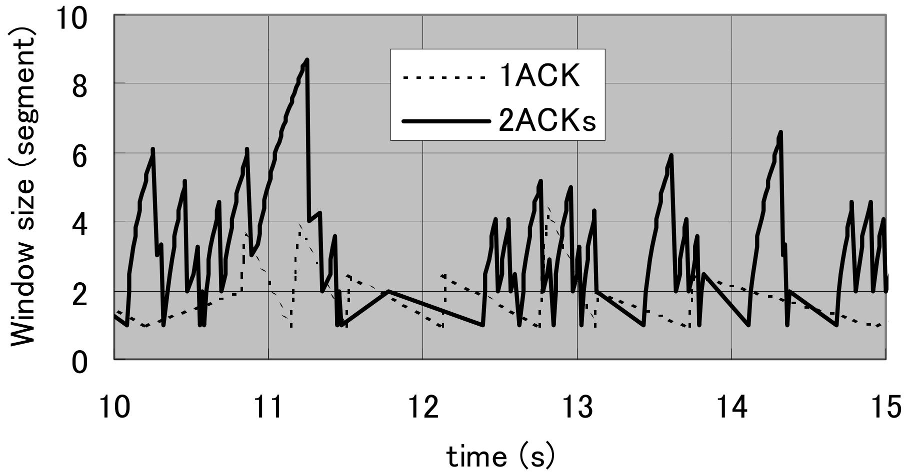


Figure 4.8 TCP Window Size (p_{err}= 0.1)

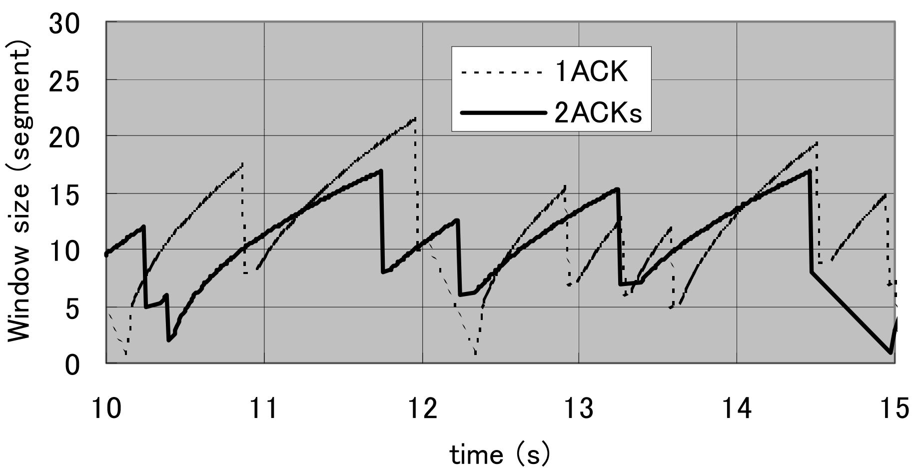


Figure 4.9 TCP Window Size (p_{err}= 0.01)

4.6 Conclusion

In this chapter, we have presented a method for improving TCP throughput in wireless cellular networks, which needs changes of TCP layer only by the side of a wireless terminal. In our method, the wireless terminal sends multiple ACKs. By introducing it, it is expected that TCP becomes robust against radio link errors. We have introduced the estimation method of the packet error probability on the wireless link, and have determined the appropriate number of ACKs by the analytical method. By means of simulation, we have revealed that TCP throughput can be improved in the range with the high error probability of the wireless link as we have expected. In our proposed method, the effective range exists. As further work, we are going to study additional proposals in order to extend this effective range.

Chapter 5

Conclusion

TCP has provided for reliable communication on Internet with several improvements since early 1970s. Though it has been designed for wired network, nowadays it is used for wireless network due to user's desire for mobility without enough study and evaluation. TCP has been providing reliable communication for wireless Internet environment as typified by wireless cellular networks. Nevertheless, it is not saying that performance evaluation of TCP for wireless cellular networks are enough. For this purpose, we have done case study in this thesis.

We have first focused on the characteristics of the underlying data link protocol, which is used for communication between the BS and wireless terminals. About this issue, we have proposed an analytical method for evaluating the TCP throughput performance in a wireless cellular network environment, where the slotted ALOHA protocol is adopted as a data link layer. By using our method, we have shown that improving throughput at the data link layer level does not necessarily lead to the TCP throughput improvement. Furthermore, we have evaluated TCP throughput by considering transmission errors on the wireless link. It was shown that when we introduce FEC as an error correcting method, we have shown that TCP throughput can be improved by selecting an appropriate error correction code with careful consideration on the overhead and the error correction capability according to the quality of the wireless channel.

Secondly, we have focused on the problem that the packet losses due to the transmission errors cause unexpected degradation of TCP throughput in a wireless cellular network environment. About this problem, many approaches have been proposed. To begin with, we have compared the TCP performance using Snoop Protocol, ELN and the fixed FEC, through which we find the appropriate FEC code against given BER. Based on above study, we have proposed a new adaptive FEC scheme combined with ELN that was proposed for improving TCP performance in wireless cellular networks. In our method, transmission errors on the wireless link are measured at the packet level and the error status is notified the TCP sender with ELN. According to this information, an appropriate FEC code is determined in order to maximize the TCP performance. We then have shown how the adaptive FEC can be realized using our solution, and also examined the appropriate observation period of measuring BER enough for the fading speed on the noisy wireless link. We have demonstrated that our method can achieve better performance than the conventional fixed FEC by using the Gilbert model as a wireless error model.

However, these solutions have not been realized because these require major changes to network infrastructures. For this purpose, we have presented a simple method for improving TCP performance in a wireless cellular network. In our proposed method, the TCP receiver does not send one ACK, but multiple ACKs when the packet loss rate exceeds the predefined threshold. Since the major application of the current Internet is to download the Web documents from the fixed servers to the wireless terminal, it only requires changes at the wireless terminal side for performance improvement. Then, TCP becomes robust against radio link errors. We have presented an analytical method, and show that two ACKs for each TCP packet are sufficient to improve the performance.

In this thesis, we have evaluated TCP performance for wireless cellular networks, using analytical or simulated method. Based on our evaluation, we have shown actual proposals and several indications for improving TCP throughput on wireless cellular networks. We believe that our advanced work must be precious for actual design of wireless cellular networks.

In wireless cellular networks, a user moves between cells, and then the connection is switched from old BS to new BS. This is called a handoff. The packet loss or delay during handoff causes degradation of TCP throughput. In order to maintain connection, it is necessary to study procedure during handoff. We need to research this problem.

Acknowledgements

This work has been accomplished with a great number of supports.
I would like to express my respect here.

Prof. Masayuki Murata, He has been leading me to the best way in any conditions with his appropriate advice. His advice covered for my English expression nearly every pages of this work. He spared his precious time for my research. I really appreciate his thoughtful leading and advice.

Prof. Hideo Miyahara, Prof. Makoto Imase and Prof. Teruo Higashino,
I am most grateful for their survey as readers of my thesis committee. Their insightful suggestions have been invaluable.

Associate Prof. Masashi Sugano of Osaka Prefecture College of Health Sciences, He has supported my research with his expert advice. He always reviewed my research, patiently. I am heartfelt thank for his expert advice and review.

Associate Prof. Naoki Wakamiya, Associate Prof. Hiroyuki Ohsaki, Associate Prof. Go Hasegawa, and Dr. Shingo Ata of Osaka City University, I have referred their research and gotten several hints for my research. Especially, Associate Prof. Go Hasegawa, I have referred his Ph.D. thesis.
I am particularly thanking for them.

I thank many friends and colleagues in the Graduate School of Engineering Science, Osaka University and DENSO Corporation for their support.

Finally, I have my family to thank for their encouragement and understanding.

Bibliography

[1] J. B. Postel, "Transmission control protocol," *Request for Comments 793*, September 1981.

[2] J. B. Postel, "Internet protocol," *Request for Comments 791*, September 1981.

[3] R. D. J. van Nee, R. N. van Wolfswinkel, and R. Prasad, "Slotted ALOHA and code division multiple access techniques for land-mobile satellite personal communications," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 2, pp.382-388, February 1995.

[4] Z. Liu and M. El Zarki, "Performance analysis of DS-CDMA with slotted ALOHA random access for packet PCNs," *Wireless Networks*, vol. 1, no. 1, pp. 1-16, 1995.

[5] O. A. González and R. Kohno, "A spread slotted CDMA/ALOHA system with hybrid ARQ for satellite multiple access," *IEEE Journal on Selected Areas in Com-munications*, vol. 18, no. 1, pp. 123-131, January 2000.

[6] H. Balakrishnan and R. H. Katz, " Explicit loss notification and wireless web performance," in *Proceeding of IEEE GLOBECOM'98 Inter-net Mini-Conference*, November 1998.

[7] P. Sinha, N. Venkitaraman, R. Sivakumar, and V. Bharghavan, " WTCP: A reliable transport protocol for wireless wide-area networks," in *Proceeding of ACM MOBICOM'99*, pp.231-241, August 1999.

[8] A. Bakre and B. R. Badrinath, "I-TCP: Indirect TCP for mobile hosts," in *Proceeding of IEEE International Conference on Distributed Computing Systems (ICDCS ' 95)*, pp.136-143, May 1995.

[9] H. Balakrishnan, S. Seshan and R. H. Katz, " Improving reliable transport and handoff performance in cellular wireless networks," *Wireless Networks*, vol.1, December 1995.

[10] E. Ayanoglu, S. Paul, T. F. LaPorta, K. K. Sabnani, and R. D. Gitlin. "AIRMAIL: A link-layer protocol for wireless networks," *Wireless Networks*, vol.1, pp.47-60, February 1995.

[11] D. A. Eckhardt and P. Steenkiste, "A trace-based evaluation of adaptive error correction for a wireless local area network," *Mobile Networks and Applications*, vol.4, no.4, pp. 273-287, December 1999.

[12] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transaction on Networking*, vol.1, pp.397-413, August 1993.

[13] W. R. Stevens, *TCP/IP Illustrated, Volume 1*: The Protocols, Reading, Massachusetts: Addison-Wesley, 1994.

[14] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Systems Technical Journal*, vol. 39, pp. 1253-1265, September. 1960.

[15] IMT-2000 Home Page, available from *http://www.itu.int/imt*.

[16] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *Proceeding of ACM SIGCOMM'98*, September 1998.

[17] The Network Simulator ns-2 Home Page, available form *http://www.isi.edu/nsnam/ns/*.

[18] PIAFS Standard (ARIB STD) Home Page, available form

http://www.arib.or.jp/piaf/piaf_index.htm.

[19] J. B. Postel, "User datagram protocol," *Request for Comments 768*, August 1980.

[20] G. Montenegro, S. Dawkins, M. Kojo, V. Magret, and N. Vaidya, "Long thin networks," *Request for comments 2757*, January 2000.

[21] W. R. Stevens, "TCP slow-start, congestion avoidance, fast retransmission, and fast recovery algorithms," *Request for comments 2001*, January 1997.

[22] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP selective acknowledgement options," *Request for comments 2018*, October 1996.

[23] S. Floyd, "TCP and explicit congestion notification," *Computer Communication Review*, 24(5): 10-23, October 1994.

[24] A. Kumar and J. Holtzman, "Comparative performance analysis of versions of TCP in a local network with a mobile radio link," *IEEE/ACM Transactions on Networking*, vol. 6, no. 4, pp. 485-498, August 1998.

[25] R. Jain, "The art of computer systems performance analysis: Techniques for experimental design, measurement, simulation, and modeling," *Wiley*, 1991.

[26] M. Miyoshi, M. Sugano, and M. Murata, "Performance evaluation of TCP throughput with consideration on lower layer protocols for wireless cellular networks," *The Transactions of IEICE* (in Japanese), vol. J85-B No.5, pp. 732-745, May 2002.

[27] M. Miyoshi, M. Sugano, and M. Murata, "Performance evaluation of TCP throughput on wireless cellular networks," in Proceedings of *the IEEE Vehicular Technology Conference 2001 Spring (VTC2001 Spring),* Rhodes (Greece), CD-ROM, May 2001.

[28] M. Miyoshi, M. Sugano, and M. Murata, "Performance evaluation of TCP throughput on wireless cellular networks," *Technical Report of IEICE* (CQ2000-46) (in Japanese), pp. 49-54, October 2000.

[29] M. Miyoshi, M. Sugano, and M. Murata, "Improving TCP performance for wireless cellular networks by adaptive FEC combined with explicit loss notification," to appear in *IEICE Transactions on Communications Special Issue on Mobile Multimedia Communications*, December 2002.

[30] M. Miyoshi, M. Sugano, and M. Murata, "Performance improvement of TCP on wireless cellular networks by adaptive FEC combined with explicit loss notification," in Proceedings of *the IEEE Vehicular Technology Conference 2002 Spring (VTC2002 Spring)*, Birmingham (USA), CD-ROM, May 2002.

[31] M. Miyoshi, M. Sugano, and M. Murata, "Improving TCP performance for wireless cellular networks by adaptive FEC combined with explicit loss notification," *Technical Report of IEICE* (RCS 2002-24, CQ2002-24) (in Japanese) pp. 39-45, April 2002.

[32] M. Miyoshi, M. Sugano, and M. Murata, "An acknowledgement approach for performance improvement of TCP in wireless cellular network environment," submitted to *IEICE Transactions on Communications*, June 2002.

[33] M. Miyoshi, M. Sugano, and M. Murata, "Performance improvement of TCP in wireless cellular network based on acknowledgement control," in Proceedings of *the 7th Asia-Pacific Conference on Communications (APCC 2001)*, Tokyo, pp. 505-508, September 2001.

[34] M. Miyoshi, M. Sugano, and M. Murata, "A study for improving TCP performance on radio terminal," *Technical Report of IEICE* (SST2000-131, AP2000-265, RCS2000-265, MW2000-256) (in Japanese), pp. 137-141, March 2001.