Chapter 1

Introduction

1.1 Introduction

Data transmission like Internet, e-mail, and etc. plays an important role in a third-generation communication system. There are many researches on data communication between wireline network and wireless network. TCP is used as the transport protocol for reliable transmission in most network applications [1-2]. TCP initially providing a control mechanism for network congestion has been continuously developed and investigated for better performance [3-4] for both the wireline and wireless communications. However, the characteristics of wireless links could mislead the TCP into a wrong congestion control. In wireless environment, the high Frame Error Rate (FER) and the variation of transmission bandwidth might cause significant degradation of end-to-end TCP performance since the high FER might trigger the wrong congestion control, the TCP timeout. Thus, Radio Link Protocol (RLP) is proposed to reduce the over-the-air errors in the wireless network [5-8]. In this way, the TCP timeout might be suppressed by means of RLP retransmission of lost packets.

The fading channel is always the hot topic and it is considerably relative to TCP performance [9-11]. Fading channel causes not only the higher FER but also the bandwidth variation [12-15] and long retransmission delay at RLP layer might cause

TCP timeout. Besides the priority scheduling will also cause the high delay variation. If the Round-Trip Time (RTT) increases severely, the spurious TCP timeout will occur.

The objective of this thesis is to provide a method to decide whether the timeout is spurious or genuine. As a result, how to estimate the RTT [17-19] or available transmission rate more accurately are important for the decision. Therefore, a RTO (Retransmission TimeOut) algorithm is proposed to resolve the spurious or genuine problems. A better TCP throughput performance can be achieved by the proposed RTO algorithm.

1.2 Organization of This Thesis

This thesis is organized as follows: we will introduce packet transmission flow in wireless communication systems in Chapter 2. In Chapter 3, we will point out TCP transmission problems and propose an Adaptive TCP to improve the system performance. Our simulation model and the selection of parameters are addressed in Chapter 4. Simulation results and discussions are concluded in Chapter 5. Finally, conclusions are described in Chapter 6.

Chapter 2

TCP & RLP Transmission in Wireless Networks

In this chapter, we will introduce the concept of reliable transmission which is used to enhance the correct reception of data transmission in the third generation cellular system. TCP retransmission provides better performance over wireline network; however, it might not be sufficient for wireless link since the higher frame error rate will mislead TCP into wrong trigger of TCP timeout. To reduce the over-the-air errors, RLP is proposed to resolve erroneous packets by retransmission before the expiration of TCP timer.

2.1 Overview of data transmission flow

Data transmission initializes after a TCP connection is established. In Figure 2-1, we can observe the packet is transmitted from a data server through Packet Data Service Node (PDSN) to Base Station Controller (BSC). The connection between data server and BSC are wireline. The characteristic of wireline connection is stable with low error rate, however, there are still some network congestion and routing delay issues. Transmission Control Protocol (TCP) and Internet Protocol (IP) will provide solutions to these issues by means of congestion control and routing schemes.

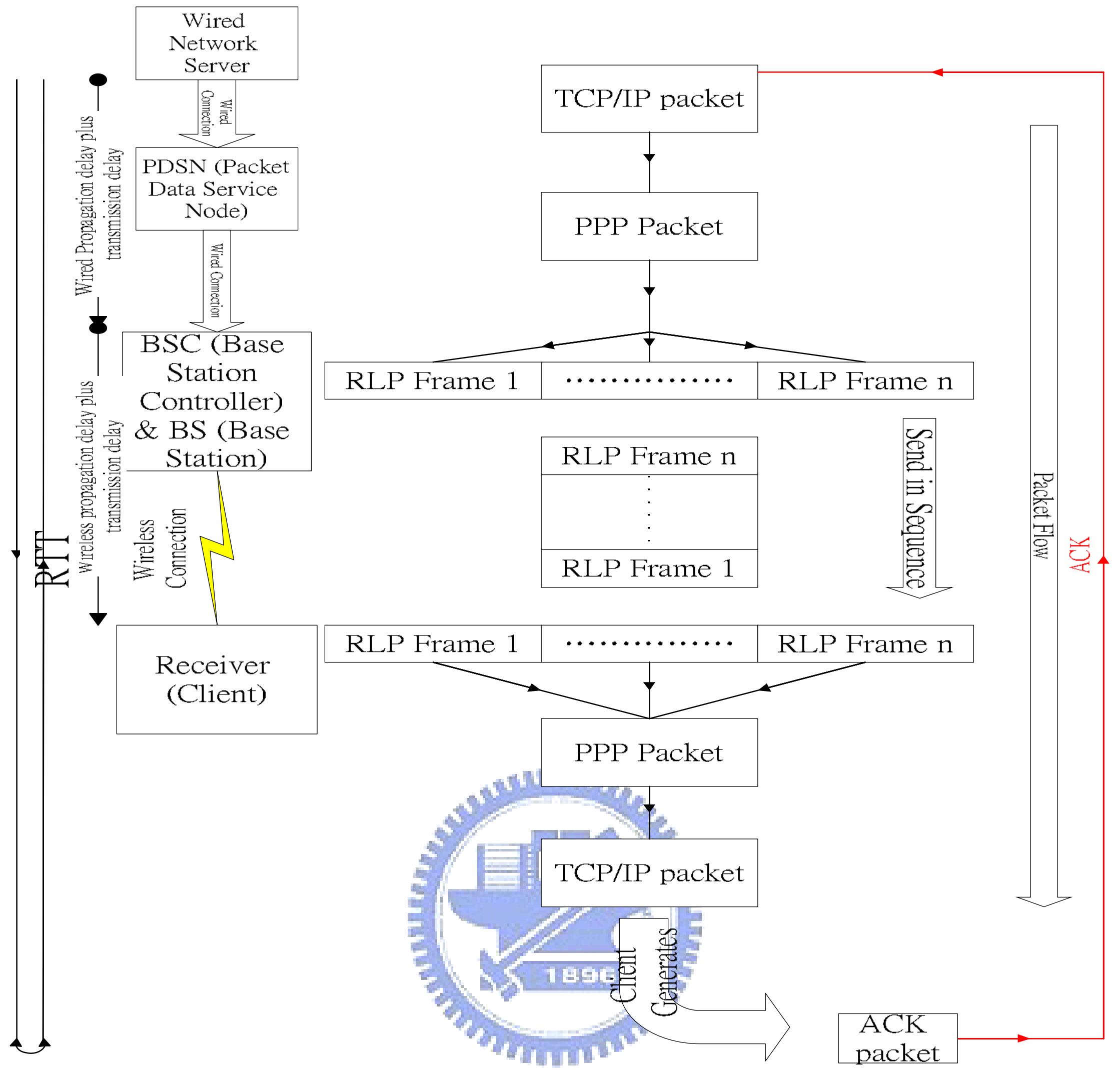

Figure 2-1 Packet Flow

Radio Link Protocol divides the PPP packet into several frames and then, these frames are transmitted over wireless link. Radio Link Protocol retransmission can significantly reduce the packet error rate in TCP level and the retransmission delay of TCP layer. If all frames which belong to the same TCP packet are received correctly by the client, these frames will be reassembled back to TCP packet and be delivered to the upper layer. Once the client receives the TCP packet, TCP at the client side will generate an ACK to inform TCP layer at data server of correct reception of TCP packet. Besides, the time between the transmission of TCP packet and the receiving acknowledgement of TCP packet is called Round Trip Time (RTT).

2.2 Transmission Control Protocol (TCP)

In the data network environment, the Transmission Control Protocol (TCP) [1,2] is used for a reliable data transport and in the Open Systems Interconnection (OSI) model, the TCP is in the forth layer – transport layer. The main advantages of TCP are reliable transmission and the avoidance of network congestion. First, Reliable transmission is provided by a sliding window algorithm which is the heart of TCP transmission.

2.2.1 Sliding Window Algorithm

Sliding window algorithm provides several advantages which are

1. it promises reliable data transmission
2. it makes sure the data will arrivals TCP in sequence
3. it provides flow control between transmitter and receiver

TCP at the sender side maintains a buffer. This buffer is used to store data that has been sent but not yet acknowledged, as well as data that has been written by the sending application, but not transmitted. At the receiver side, TCP maintains a buffer as well. This buffer holds data that arrives out of order, as well as data that is in order which has not been delivered to the upper layer.

Another purpose of sliding window algorithm is to make the pipe full. For example, there is one link which has a delay multiply bandwidth product of 8 KB and frames are of 1-KB size. The algorithm is to the sender is ready to transmit the ninth frame at pretty much the same moment that the ACK for the first frame arrives. Before introducing the algorithm, there are several parameters which are pointed out. The sender maintains three variables: The send window size, denoted SWS, gives the sender the upper bound to the number of outstanding (unacknowledged) frames; LAR denotes the sequence number of the last acknowledgment received; and LFS denotes the sequence number of the last frame sent. The sender also maintains the following invariant:

$$LFS - LAR \leq SWS \tag{1}$$

When an acknowledgment arrives, the sender moves LAR to the right, thereby allowing the sender to transmit another frame. Also, the sender associates a timer with each frame it transmits, and it retransmits the frame if the ACK is not received before the expiration of TCP timer. Notice that the sender needs to buffer up SWS frames since it needs to prepare for retransmission until the sent packet is acknowledged.

The receiver maintains the following three variables: The receive window size, denoted RWS, gives the receiver the upper bound on the number of out of order frames; LAF denotes the sequence number of the largest acceptable frame; and LFR denotes the sequence number of the last frame received. The receiver also maintains the following invariant:

$$LAF - LFR \leq RWS \tag{2}$$

When a frame with sequence number SeqNum arrives, the receiver takes the following action. If SeqNum $\leq$ LFR or SeqNum > LAR, then, the frame which is beyond the receiver's window will be discarded. If LFR < SeqNum $\leq$ LAF, then the frame which is within the receiver's window will be accepted. After that, the receiver needs to decide whether or not to send an ACK. Let SeqNumToAck denote the largest sequence number but not yet acknowledged, such that all frames with sequence numbers less than or equal to SeqNumToAck have been received. The receiver acknowledges the receipt of SeqNumToAck, even if the high-indexed packets have been received. This acknowledgment is said to be cumulative. It then sets LFR = SeqNumToAck and adjusts LAF=LFR+RWS. The flow control of TCP uses the receiver advertises a window size to the sender rather than fixed-size sliding window. This will be done by using the AdvertisedWindow field in the TCP header. Detail of TCP format and sliding window operation will be discussed in Appendix A and [23]. Because TCP guarantees the reliable delivery of data, it retransmits each segment if an

ACK is not received in a certain period of time. TCP sets this timeout as a function of the RTT. After client receives the TCP packet, it will send back an Acknowledgement for signaling the server that the packet is already correct received. If the timer expires and server still does not receive the ACK, it will be thought that the packet is lost and the packet will be retransmitted.

2.2.2 Three-Way Handshake

As mentioned, TCP is connection-oriented. Before TCP starts to transfer the data, it must make sure that the connection has been already setup. Figure 2-2 shows how to connect the end-to-end transmission path.

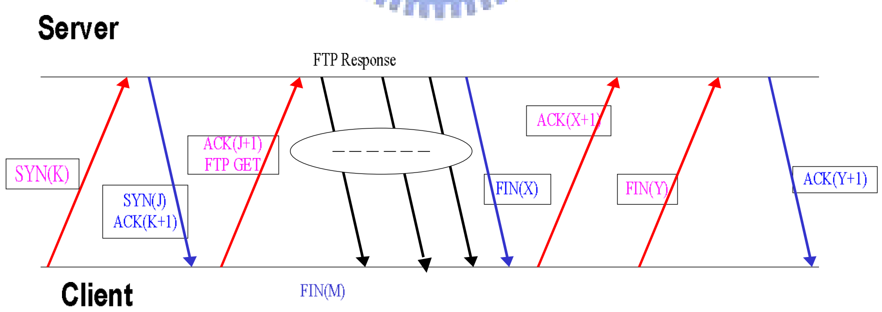

Figure 2-2 Three-way handshake and Four-way handshake

First, the client sends the SYN k to the server, which brings the start sequence number of next segment. After server receives the request, it will send a response for the message combining with the SYN J and ACK k+1. Finally, when client receives the message from the server, it sends an ACK J+1 back. After that, the connection is setup successfully. This is called **three-way handshake**.

Terminating the connection is similar with the initialization of connection. The difference is the two way connections are not closed simultaneously. Therefore, it needs to send four packets for it, which is called **four-way handshake**. All mentioned above are the TCP basic concepts. Next, we introduce the four TCP transmission algorithms. These algorithms are used to control the data transmission.

2.2.3 TCP Transmission Algorithms

In TCP transmission, there are various controls that are related to each other. The controls are **slow start** with **congestion avoidance** and **fast retransmission** with **fast recovery.** When segments which stand for the packets by TCP are transferred, TCP uses the congestion window (cwnd) as the unit of transmission. In the network environment, the packets are transferred from the sender to the receiver continuously until the advertised window used by the receiver shows that there is not enough space for any packets. But the end-to-end users do not how many capacity in network is available. In order to avoid the problem, the slow start algorithm is implemented. The slow start means that after the connection has established, the congestion window is initialized to one. After the first successful receiving segment, it will send back an ACK for the acknowledgement. Every time when the sender receives one ACK, it sends two segments. In other words, the number of segment increases at the order of 2^n, for n=0, 1, 2, and etc. Figure 2-3 shows the implementation of the slow start. In this way the sender can probe the capacity which how many packets it can safely have in transit. Besides slow start can effectively increases the congestion window exponentially, rather than linearly.

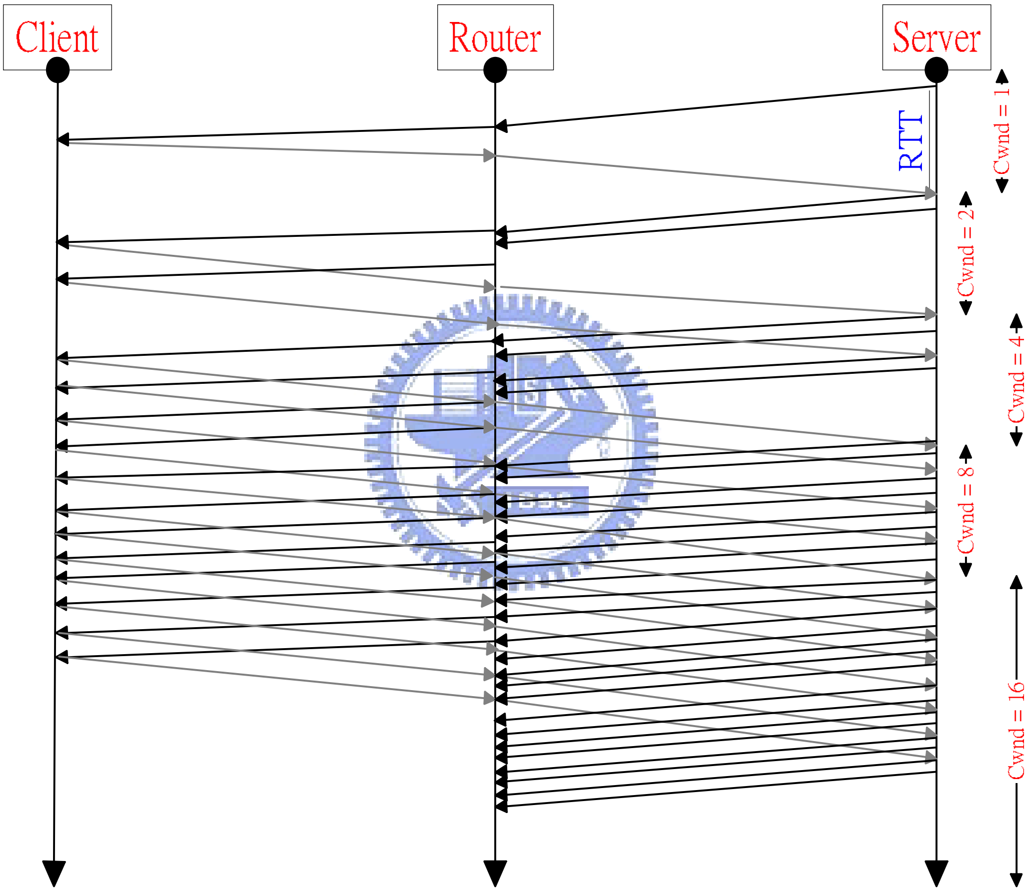

Figure 2-3 Slow start

The congestion window increases until the slow start threshold (ssthresh) is reached. After that, TCP will go into the congestion avoidance state. Hence the slow start threshold is taken as an indication to tell sender change the form of the cwnd increase. Congestion avoidance means that the cwnd increases linearly. The equations

(3)-(4) show that the cwnd increase form.

Slow start cwnd < ssthresh

$$cwnd(n+1\text{-th}) = 2*cwnd(n\text{-th}) \tag{3}$$

Congestion avoidance when cwnd >= ssthersh

$$cwnd\ (n+1\text{-th}) = cwnd\ (n\text{-th}) + 1 \tag{4}$$

the equation (3) is the slow start phase and the n-th, n+1-th means the round-trip number. In slow start phase the cwnd increases as the mentioned at the order of 2^n for each round-trip. And the equation (4) is the congestion avoidance state, each round-trip the cwnd will increase one.

And in TCP, there is two indications of packet loss are timeout and duplicate ACKs. Because the wireline transmission is stable, the probability of packet error is very low. Besides the RTT in the equation means the time between transmission of a segment and reception of ACK is called Round Trip Time (RTT). The RTT is one of important factors to calculate RTO (Retransmission timeout). When the timeout occurs, cwnd is down to one segment and ssthresh reduces by half to its current cwnd. Then the algorithm goes through the slow start procedure again. From the above algorithms, we can see that when timeout occurs, it will significant decrease TCP throughput. The ssthresh value changes when the round-trip timeout occurs. As the equation (5) shows

$$ssthresh(n+1-th) = \frac{cwnd(n-th)}{2} \qquad \text{when timeout occurs} \qquad (5)$$

The fast retransmission and fast recovery is used by the receiver to signaling the packet out of order. Fast retransmission will be trigger when the sender received more the three the same acknowledgement sequence number. After that TCP server will reset the retransmission timeout timer and retransmits the packet immediately. The fast recovery algorithm will change the congestion window to half of original rather than one when the fast retransmission is successfully. But in the wireless transmission it might not proper. The transmission rate of wireline network is very high hence the transmission delay is very short. After receive more than three ACK, the timer will not expire. But it is not the same for the wireless transmission, the lower transmission rate causes larger transmission delay, when server receives more than three ACK the timer is possible to exceed the RTO. So we do not consider these two algorithms in this problem.

2.3 Radio Link Protocol (RLP)

The Radio Link Protocol (RLP) is introduced in wireless communication systems to solve the TCP performance problem in wireless environment. Besides of the Automatic Repeat request (ARQ) scheme, RLP further improves the TCP performance by segmentizing a TCP packet into smaller segments. For ARQ method, RLP is a negative acknowledgement (NACK) based on ARQ protocol, that is, RLP only sends the negative acknowledgements for the lost or erroneous frames rather than sending acknowledgments for successfully received frames [9]. There are several optional retransmission schemes, such as (1, 1, 1), (1, 2, 3), (2, 3), and etc. [2]. Take scheme (1,1,1) for example: when RLP at the receiver side finds out that a frame is in error (missing), it sends back one NACK requesting for retransmission of this frame. A timer is set for the missing frame. When the timer expires but the frame is still not received, another NACK will be sent again. The timer is reset. For each NACK received at the sender, the sender will retransmit the missing frame. If the timer expires again, the RLP sends third NACK for the retransmission of the frame. Finally, if the timer still expires before the successful reception of this frame, RLP sends NACK to the TCP layer. As a result, it means RLP can not recover this frame and RLP hands this job to TCP which is going to take care of the associated packet. The figure 2-4 shows the procedure of scheme (1,1,1). The purpose of RLP is trying to recovers the lost RLP frame before the error occurs in the TCP layer, i.e., before TCP times out. In this way, it not only can reduce the transmission delay, i.e., the time required for transmission from upper to lower layer, also can reduce the error probability comparing with larger size of a TCP packet over the fading channel.

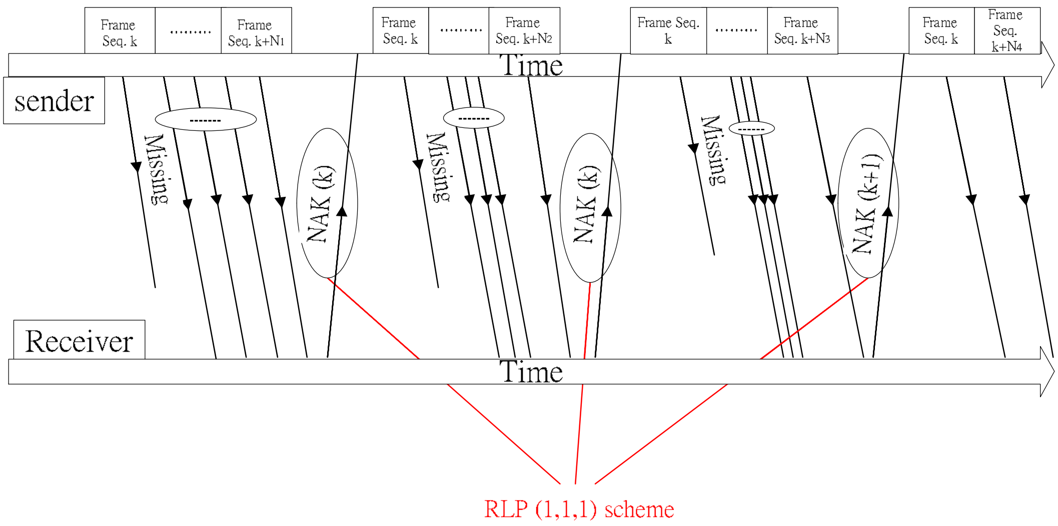

Figure 2-4 RLP retransmission procedure and retransmission scheme (1,1,1)

2.4 Interaction between TCP and RLP

TCP and RLP both use retransmission to improve the system throughput. But RLP is used to reduce the packet error rate from the TCP point of view. Figure 2-5 presents the relationship between TCP and RLP. As shown, we know that each TCP retransmission will cost more delay than RLP. Hence, retransmission in RLP layer will significantly improves system throughput over wireless link with higher FER as compared to the wireline networks.

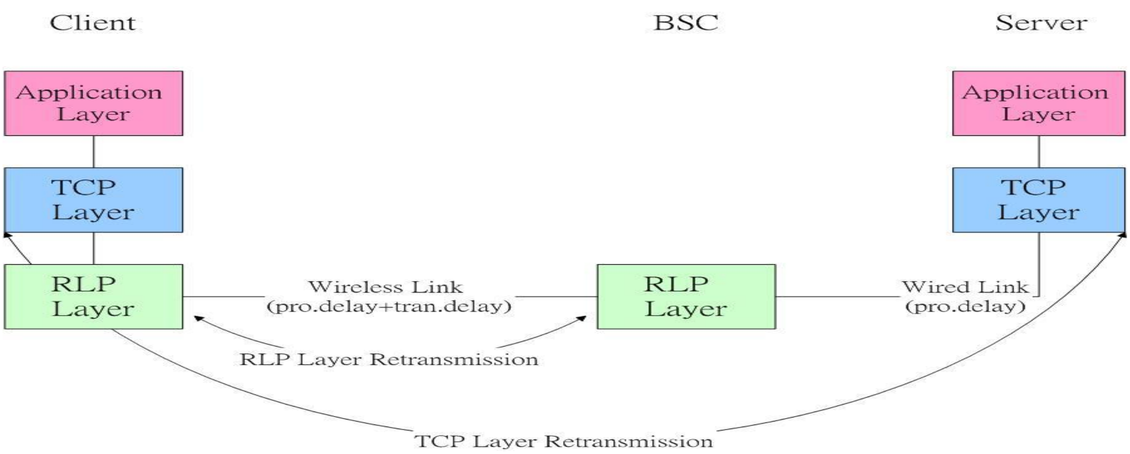

Figure 2-5 Relationship between TCP and RLP

Chapter 3

Problem Statement and Adaptive TCP Algorithm

The scope of this chapter is to describe the problem of TCP over wireless environment and its reaction to the wireless environment. Also, the adaptive TCP algorithm is proposed to improve the overall TCP throughput.

3.1 Wireless Transmission

The wireless channel is commonly regarded as a "Fading channel" as shown in Figure 3-1.

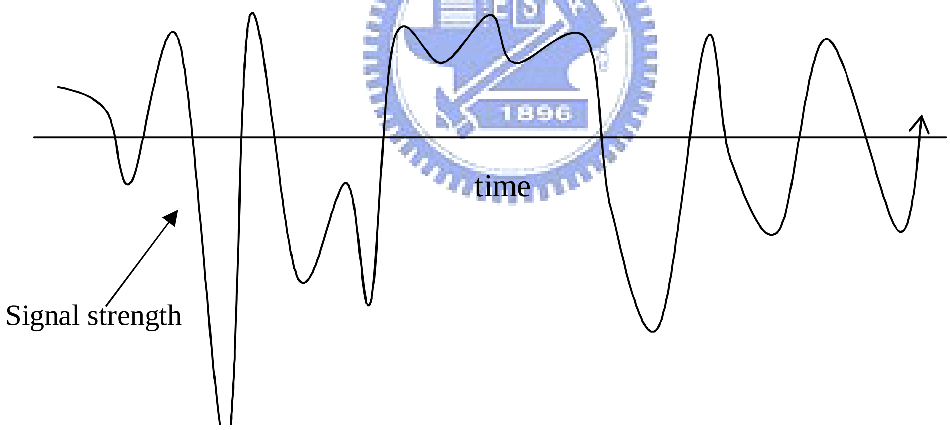

Fig. 3-1 The behavior of fading channel

Besides of this phenomenon, "fades", the multiplicative-noise effect is also induced in the channel, which results in the burst errors. Although the power control can solve the deterioration of the signal strength, it can not solve the multiplicative-noise-wise problem. This is because adding too much transmitted power will considerably degrade the system performance. In this thesis, the Automatic Repeat Request (ARQ) in RLP layer will solve this issue. Although ARQ can successfully recover most lost packets by retransmitting it, it takes extra time to do so.

The fading channel could cause the so-called error burst, i.e., the transmission could fail in some consecutive transmission slots. As a result, uniformly distributed error model is not adequate to describe the wireless channel. Consequently, to model the error burst channel, a two-state Markov model, also called Gilbert model, is adopted in the bit level performance [20].

The following diagram is to descript the first order two-state Markov model as shown in Figure 3-2. The channel condition is represented by either “good” or “bad” states. For simplicity, the simple Markov model is assumed in this paper. In a good state, we assume there is no transmission error. On the other hand, in a bad state, all packet transmission will result in errors. As a result, the two-state Markov model is characterized by a transition matrix, **T**, which is derived in Equation (6) [20] .

$$\mathbf{T} = \begin{pmatrix} p & 1-p \\ 1-q & q \end{pmatrix} \qquad (6),$$

where, p and q stand for the probability of staying in good and bad state, respectively. In addition, (1-p) and (1-q) represent the transition probability from a good state to bad state and from a bad state to good state, respectively.

Also, the average symbol error rate e could be acquired [20]:

$$e = \frac{1-p}{2-p-q} \qquad (7)$$

We assume the burst length (bl) is geometrically distributed, where we can attain:

$$E[bl] = \frac{1}{1-q} \qquad (8)$$

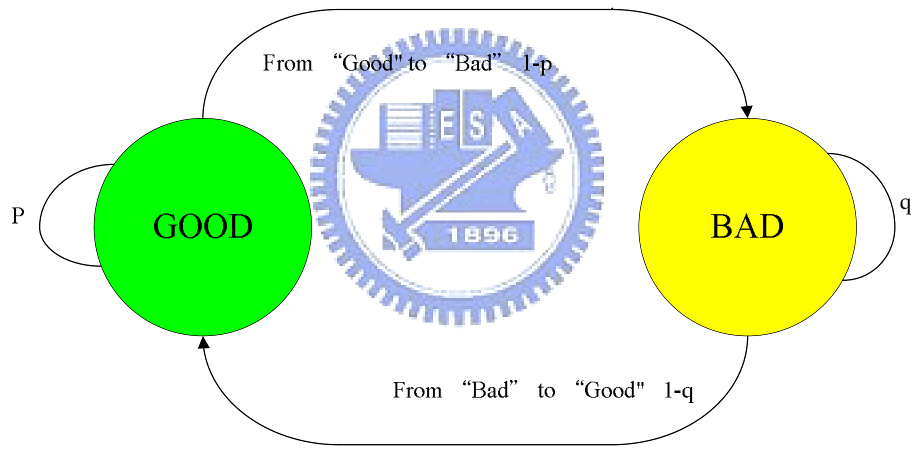

Fig. 3-2 Transition state of Gilbert channel

With transition matrix **T,** the sequence of burst error can be generated and the characteristics of wireless channel can be described by the matrix **T**.

3.2 Problem Statement

Transmission Control Protocol (TCP) is widely used for reliable data transmission in wireline networks whose error rate is low. The major reason for packet loss is network congestion. In wireless communications, unlike wireline network, the higher FER and the variation in transmission bandwidth could have serious impacts on the TCP performance. Figure 3-3 shows how the bandwidth variation changes due to the resource allocation or channel condition. For example at t1, it shows the abrupt decrease of bandwidth from 307.2kbps to 19.2kbps. Besides, at t2, it is another abrupt decrease of bandwidth variation. The potential impact on bandwidth variation is that the Round trip time will change accordingly. The bandwidth reduction will cause negative impact on TCP performance.

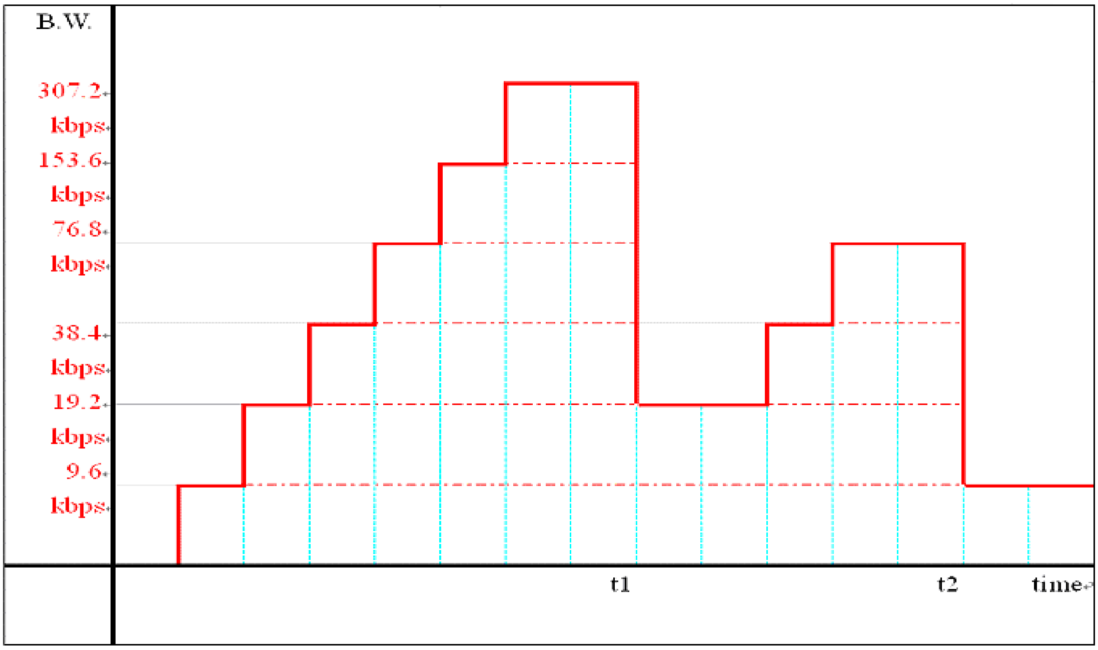

Figure 3-3 Bandwidth variation with the time.

Figure 3-4 shows the relationship between RTT and RTO associated with bandwidth variation which in the figure 3-3. The RTT continuously decreases dues to the increase of bandwidth step by step.

RTO calculation is based on the RTT; hence, the RTO will change with the RTT value. At t1, bandwidth varies from high to low suddenly as shown in figure 3-3, then it will cause the RTT increases suddenly, which exceeds the RTO which is calculated based on the original bandwidth.

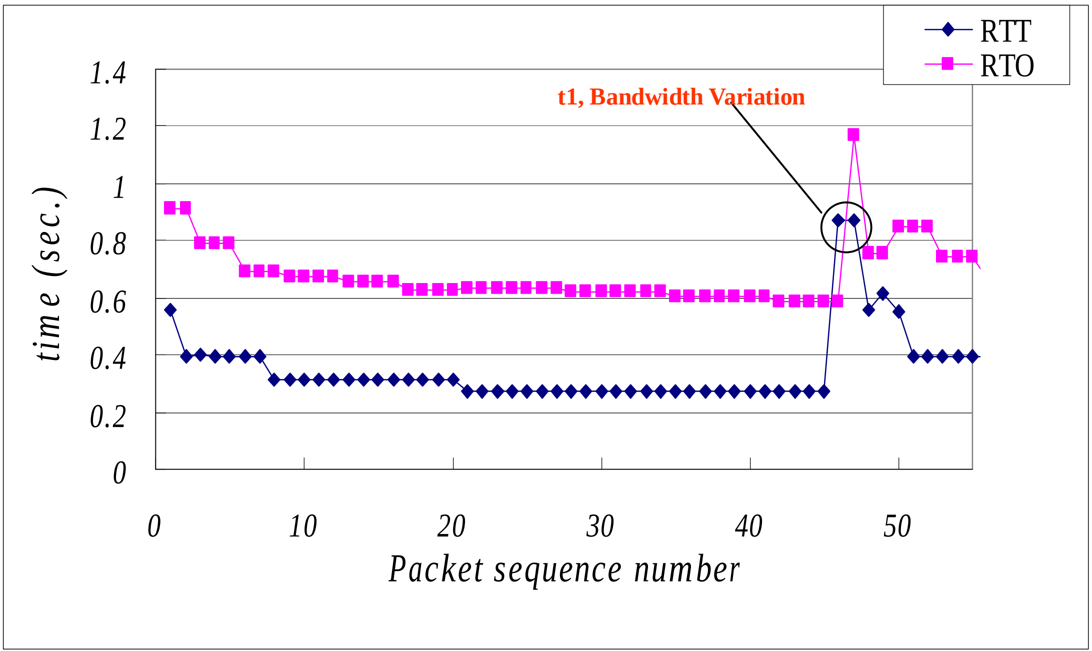

Figure 3-4 Relationship of RTT and RTO relative to bandwidth variation

Because of the TCP congestion control mechanism, once the timeout happens, the TCP in the server will take it as the network congestion, and then it triggers the mechanism, congestion control. The concept of congestion control is to decrease the data rate in order to eliminate (or reduce) the network congestion. Figure 3-5 is an example of cwnd and ssthersh variation with the congestion control. The cwnd continuously grows regardless of being in slow start phase or in linear phase until the timeout happens or cwnd reaches the maximum value. Timeout events could be caused by the network congestion, bandwidth variation or over-the-air errors. Therefore, if the timeout happens, the cwnd will be down to one and cwnd will grow up based on the slow-start manner. Slow start threshold (ssthresh) is reduced by half of the cwnd final. Figure 3-5 also shows the slow start and congestion avoidance algorithm. From the figure, we can see that the red line means slow start threshold and blue line means congestion window size. When blue line is under the red line, the TCP stays at slow start state and when blue line is equal to or over the red line, the TCP go to the congestion avoidance state. From the figure, we can see that if TCP timeout occurs, the ssthresh will become half of the cwnd.

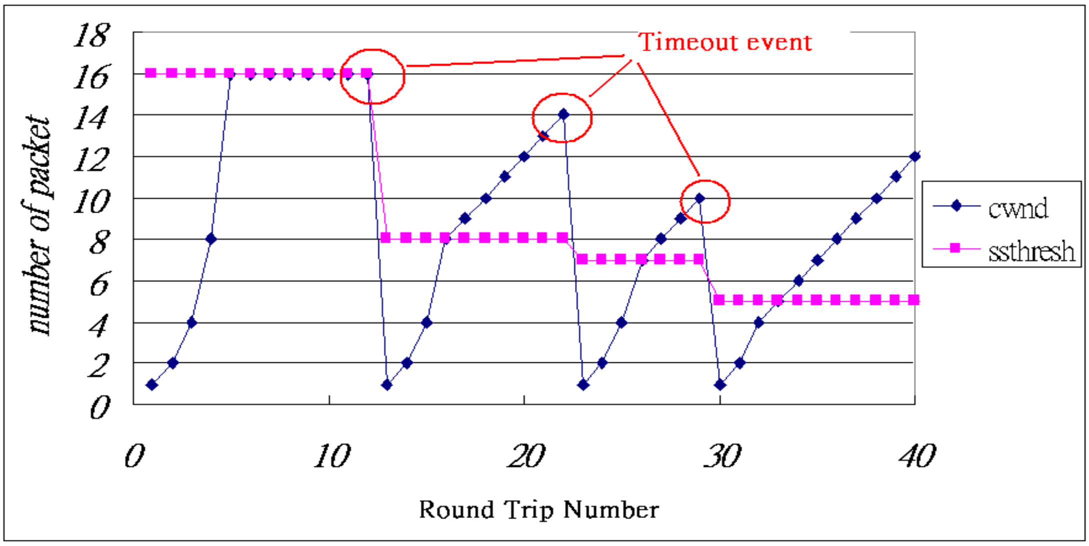

Figure 3-5 Variation of transmitter's congestion window and slow start threshold

The calculation of RTO is based on the wireline network and it is composed of RTT, estimation of RTT (R) and deviation of RTT (D). Over a wireline network, data transmission is very stable, so the deviation of transmission rate is very small. However in the wireless network, the assumption might not be standed because of the fading channel.

3.3 Adaptive TCP

The objective of the studies is to avoid a spurious timeout by moving into the slow start state. The Characteristics of wireline and wireless network are different. Therefore, the analysis of TCP should be divided into two parts. One is the wireline and another is wireless network. First, the reason of TCP timeout could be the traditional congestion packet lost over the network. Of course, there could be some timeout caused by the wireline network routing delay. Another reason why TCP times out is wireless environment issue, bandwidth variation. This might increase RTT suddenly. Therefore, the avoidance of TCP timeout is important.

3.3.1 Indirect TCP (I-TCP)

Split the end-to-end TCP into two, wireline network and wireless network, as shown figure 3-6 the I-TCP (Indirect-TCP) [21] author describes the advantages of it are

1. It separates the flow and congestion control on wireless link from the wireline link.

This separation is necessary because the characteristic of wireline and wireless network are different. Transmission rate of wireline network is from several tens Mbps to several Gbps. But in wireless system the transmission rate is between several kbps to several Mbps. In addition, the wireless network is unreliable because of the fading channel.

2. the separate protocol can provides the notification of events for instance disconnection, available transmission rate, handoff and other feature of the wireless link to higher layers which can be used by supporting the link aware and location aware mobile applications.

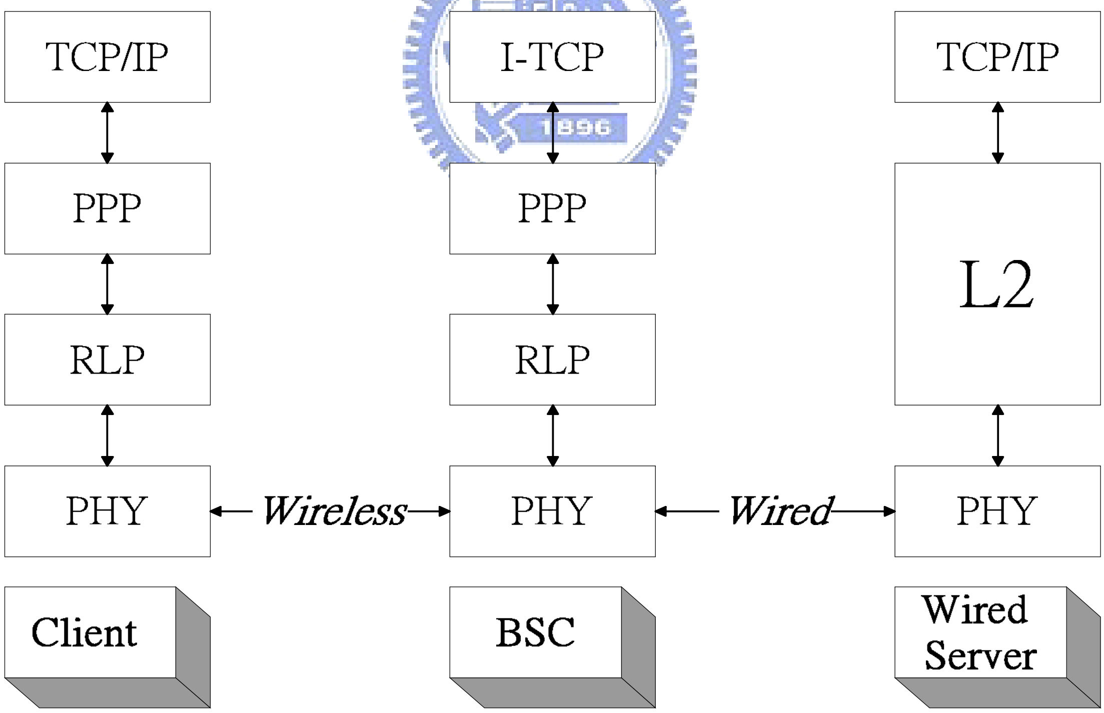

Figure 3-6 End-to-end protocol stack

Adaptive RTO algorithm is proposed based on the I-TCP to notification the TCP about the bandwidth variation which may cause unnecessary TCP layer packet timeout and get into the slow start phase. The main object of this algorithm is to detect the timeout causes only by bandwidth oscillation in order to prevent the TCP timeout. Original RTT and RTO estimation values are proper for wireline environment which is not expected the unstable transmission rates. The calculations are shown as below:

$$R = \frac{7}{8} \times R + \frac{1}{8} \times RTT \tag{9}$$

$$D = \frac{3}{4} \times D + \frac{1}{4} \times abs(R - RTT) \tag{10}$$

$$RTO = R + 4 \times D \tag{11}$$

- RTT means the measurement of packet round-trip time
- R means estimation for the packet transmission time.
- D means the deviation estimator of RTT
- RTO means the Round trip timeout

The RFC793 [2] algorithm which shows above is used for estimation of the mean of the Round-Trip Time. And at the beginning the first round-trip, three-way handshake, does not have the timeout mechanism. It is not only provides the sequence number of first data segment but also used for measure the time duration of packet transmission. The duration is the first RTT. After the process, TCP will set the initial value for R, D, RTO [1, 2]. The initial value of R is equal to first RTT, and the D is equal to half of first RTT. Using the equation (8), TCP can get the first RTO value.

3.3.2 Wireless Adaptive-RTO TCP Algorithm (WA-RTO)

The original of the idea is to avoid the spurious timeout which is caused by the bandwidth variation. If we can disable the TCP timeout mechanism when bandwidth variation, then problem will be solved. But when should we recover the mechanism or if network congestion occurs during the moment of RTO disable. Both above will introduce another problem in TCP transmission. Finally we think the better solution for spurious timeout is to use the Wireless Adaptive-RTO TCP Algorithm. By usage of the algorithm we can not only avoid the spurious timeout but also prevent the network congestion during the moment.

In the data transmission, RTT can be divided into transmission delay and propagation delay [23]. And the spurious timeout occur dues to the bandwidth variation. That means the increase of RTT is the transmission delay. Transmission delay is shown as (12)

$$\text{Transmission delay} = \frac{packet\_size}{transmission\_rate} \tag{12}$$

The goal of this algorithm is to provide a proper RTO value for the data transmission which the bandwidth variation has occurred.

Once the TCP detects the bandwidth oscillation occurs, TCP sends a packet to the server as an indication of the RTO value changes. But the bandwidth variation could be a positive or negative, and the RTO will be changed based on the bandwidth variation. The equation represents as

$$RTT\_esti(t+1) = (RTT\_esti(t) - pro.delay) * \left(\frac{previous\_bw}{present\_bw}\right) + pro.dealy \qquad (13)$$

$$RTO = RTT\_esti + 4 \times D \qquad (14)$$

which $\quad pro.delay = propagation\_delay = RTT - \left(\frac{MTU}{BW}\right) \qquad (15)$

D is previous calculation and the $\frac{MTU}{BW}$ is data transmission delay over a wireless link

In equation 13, the RTT_esti(t) is the estimation for RTT before the bandwidth variation. And the RTT_esti(t+1) means new estimation for RTT after bandwidth variation. The propagation delay stands for the packet total propagation delay which includes the wireline and wireless propagation delay. Hence the item of RTT_esti(t) subtracts propagation delay will be equal to wireless transmission delay because the wireline transmission delay is too small that we can ignore it. From the equation (12), we know that the bandwidth variation will cause the variation of RTT. Wireless transmission delay could increase or decrease dues to the bandwidth variation. According to this we can estimate the new wireless transmission delay. Taking the value which is equal to the original transmission delay multiplies the bandwidth variation as the new transmission delay. After that adds the new transmission delay and the propagation delay then we can get a new estimation for the RTT. Finally using the new estimation for RTT, the new RTO value will be generated. The MTU which is seen in the equation (15) is Maximum Transfer Unit which is used by TCP.

If we just increase the RTO value it can avoid the spurious timeout but once the network congestion occur TCP must spend more time to wait for the retransmission. Thus, the adaptive TCP algorithm is considered in this issue.

Chapter 4

Simulation model

In this chapter, we will introduce the simulation platform and list all of assumption parameters for the simulation. Finally, the performance metric is proposed.

4.1 Assumption Parameters

The purpose of this thesis is to focus on the unreliable transmission over the wireless link. To achieve that, model an end-to-end TCP transmission as shown in Figure4-1.

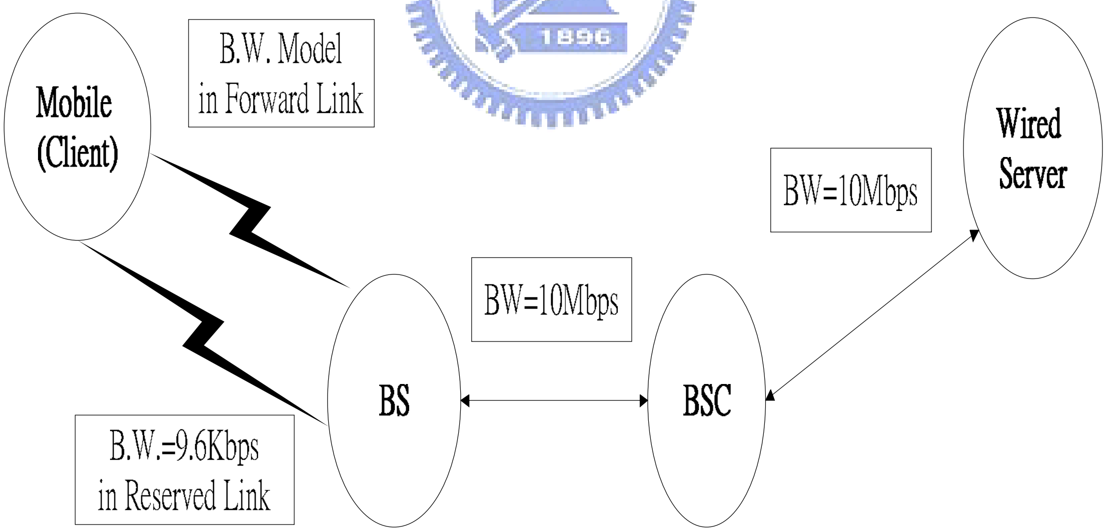

Figure 4-1 End-to-end simulation model

The links between the server and BS are wireline. Also, the bandwidth of wireline link is 10Mbps that is a common assumption for Ethernet [23]. In addition, TCP segment size is assumed to be 1500 bytes. Under the assumption, the transmission delay of wireline equal to 1500bytes divide 10Mbps and it is about 1.2ms which is very small in the whole TCP segment transmission. As a result, we will ignore its impact on the system throughput calculations. The summary of the parameters are list in table 4-1. In the data transmission, the RTT can be divided into the propagation delay and the transmission delay [23] and in our simulation model the processing time is included in the propagation delay. For more detail, please refer to [10]. The propagation delay from the BSC to the client is about 30ms. Besides, the buffer size of server is about 16Kbytes.

Table 4-1 Propagation delay and bandwidth models of each link

Link	Bandwidth	Propagation Delay
Wired Server -> BSC	10Mbps	100ms
BSC -> BS	10Mbps	10ms
BS -> Client	BW Model	20ms
Client -> BS	9.6kbps	20ms
BS -> BSC	10Mbps	10ms
BSC -> Wired Server	10Mbps	100ms

4.2 Simulation Flow

The Figure 4-2 shows the flowchart of simulation program, and IP layer routing issues in the wireline network are not considered in our program.

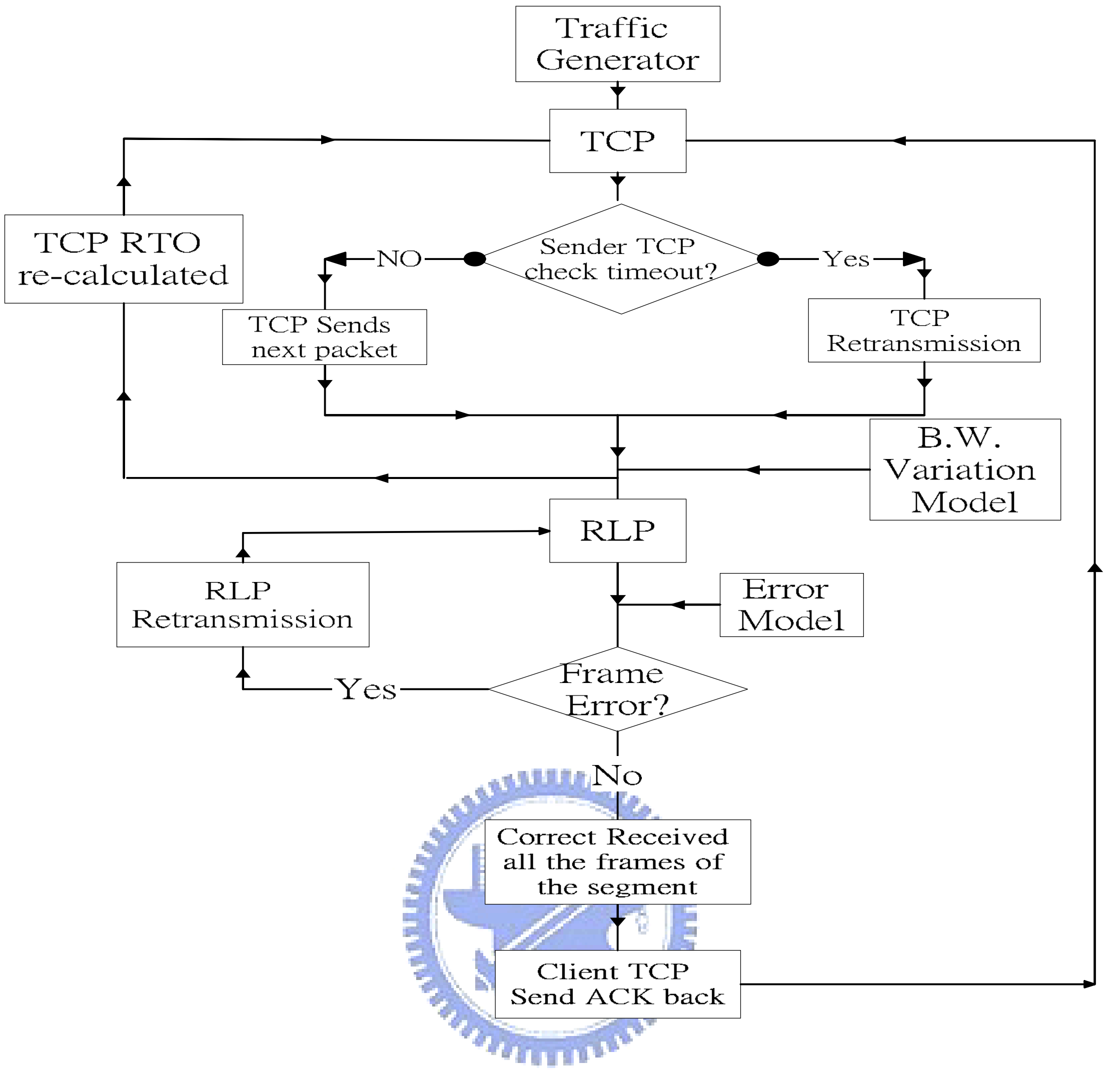

Figure 4-2 Simulation Flow

Traffic Generator generates either the HTTP or FTP traffic for TCP transmission and the parameters are adopted from 1xEV-DV specification [22]. These parameters are list in table 4-2 and table 4-3. From the [22], we know the characteristics of HTTP are different from the FTP such that the utilization of channels might be different. Generally, FTP traffic will have better performance than the HTTP traffic.

Table 4-2 FTP Traffic Parameter

Component	Distribution	Parameters	PDF
File size (S)	Truncated Lognormal	Mean=2Mbytes Std=0.722 Mbytes Max=5 Mbytes	$f_x = \frac{1}{\sqrt{2p}\, sx} \exp\left[\frac{-(\ln x - m)^2}{2s^2}\right]$ $x \geq 0,\ s = 0.35,\ m = 14.45$
Reading time (D_{pc})	Exponential	Mean = 180 sec.	$f_x = l e^{-lx}, x \geq 0$ $l = 0.083$

Table 4-3 HTTP Traffic Parameter

Component	Distribution	Parameters	PDF
Main object size (S_M)	Truncated Lognormal	Mean=10710 bytes Std=25032 bytes Min=100 bytes Max=2 Mbytes	$f_x = \frac{1}{\sqrt{2p}\, sx} \exp\left[\frac{-(\ln x - m)^2}{2s^2}\right]$ $x \geq 0,\ s = 1.37,\ m = 8.35$
Embedded object size (S_E)	Truncated Lognormal	Mean=7758 bytes Std=126168 bytes Min=50 bytes Max=2 Mbytes	$f_x = \frac{1}{\sqrt{2p}\, sx} \exp\left[\frac{-(\ln x - m)^2}{2s^2}\right]$ $x \geq 0,\ s = 2.36,\ m = 6.17$
Number of embedded objects per page (N_d)	Truncated Pareto	Mean=5.64 Max=53	$f_x = \frac{a k^a}{x^{a+1}}, k \leq x < m$ $f_x = \left(\frac{k}{m}\right)^a, x = m$ $a = 1.1, k = 2, m = 55$
Reading time (D_{pc})	Exponential	Mean=30 sec	$f_x = \lambda e^{-\lambda x}, x \geq 0$ $\lambda = 0.033$
Parsing time (T_p)	Exponential	Mean=0.13 sec	$f_x = \lambda e^{-\lambda x}, x \geq 0$ $\lambda = 7.69$

When data traffic is generated, TCP gets into connection phase and sends connection packet for estimation of RTT value. As mentioned, RTT measurement is used for the RTO calculation. At TCP layer, packet data traffic will be divided into several TCP segments and be transferred to the client. TCP will check each packet whether times out or not. If the packet times out, TCP will trigger the congestion control and retransmit the lost packet. If it does not, TCP, then, will deliver new segment.

The bandwidth variation model is used for modeling the wireless bandwidth variation. The system assigns the different bandwidth for each TCP packet dues to channel condition or resource allocation. When BSC detects the bandwidth variation, then, it will send the signaling information which includes the previous bandwidth and the current bandwidth for recalculation of the RTO. In next section, we will talk about the performance metric and bandwidth variation model.

In order to solve the wireless transmission issue, RLP is implemented in the simulation. RLP performs the retransmissions according to its protocol and (1,1,1) retransmission scheme is used here. Once the frame is error, it will be retransmitted because of ARQ in the RLP layer. In addition, the error model is based on two state Markov model and Frame error rate is assumed to be 1%~15% in the simulation.

4.3 Performance Metrics

The focus of the simulation is based on the wireless behavior since the characteristics of wireless links have an impact on the TCP throughput. The first case focuses on the bandwidth change frequency. Secondly, the standard deviation of bandwidth variation will be investigated. Thirdly, the TCP throughput will be discussed in the FER point of view.

Chapter 5

Simulation Results

The sensitivity study will be performed in our simulation to analyze the impacts on the system performance due to different circumstances. At the beginning, we will present the impact on the adaptive TCP algorithm. The major purpose of the adaptive TCP algorithm is to eliminate the TCP spurious timeout. In this way, we will conclude the relationship between RTO and RTT with/without the adaptive TCP algorithm. Later, we will compare the impact of bandwidth change frequency on performance. In addition, we also use different transmission models for data transmissions. Thirdly, we studied the standard deviation of bandwidth variation in the system performance. Finally the different frame error rate impacts will be considered in the thesis.

5.1 RTT and RTO with/without Adaptive TCP

In this section, the comparison of RTT and RTO with/without adaptive TCP algorithm is studied. In the simulation, both FTP and HTTP traffic are used because they are the popular application layer protocol in the data transmission. And we want to show the relationship between RTT and RTO with the algorithm. The following two diagrams figure 5-1(a) and figure 5-1(b) show the distribution of RTT and RTO with/without adaptive TCP algorithm. It is shown that with adaptive TCP algorithm, the RTO is always larger than RTT slightly, even if bandwidth variation occurs. Everytime when the bandwidth varies, RTT will change but the RTO is not so sensitive to the abrupt variation generally.

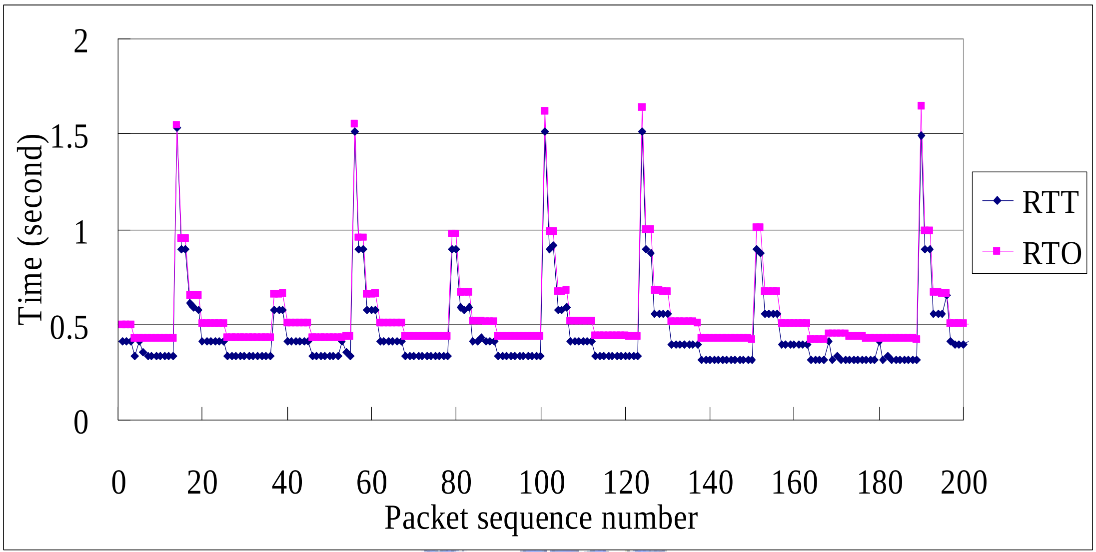

Figure 5-1 (a) Relationship of RTT and RTO with adaptive TCP algorithm

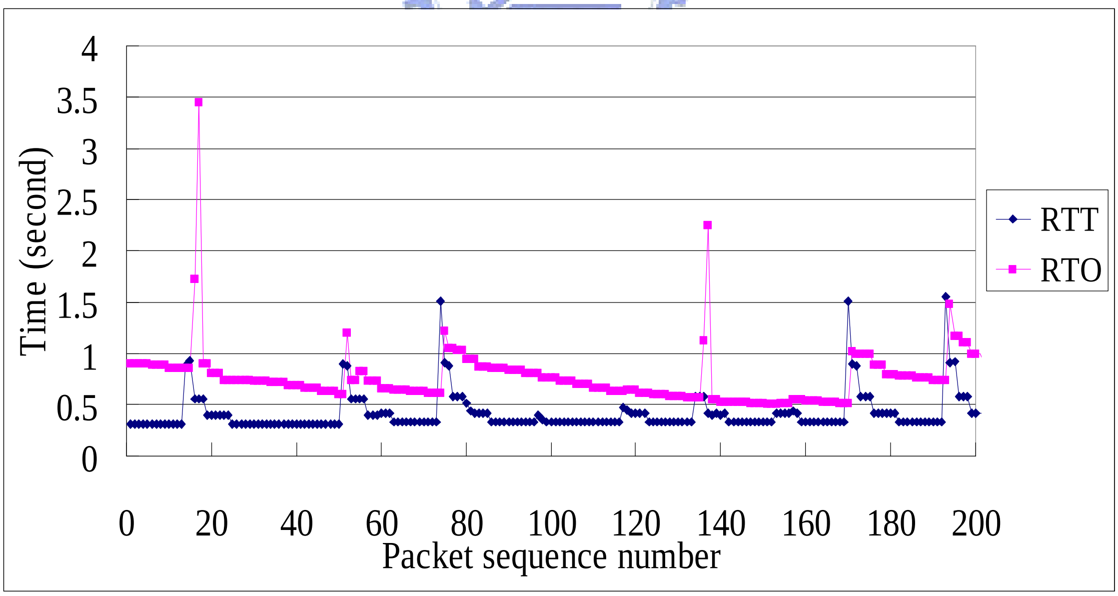

Figure 5-1 (b) Relationship of RTT and RTO without adaptive TCP algorithm

Besides, the wireline congestion is not considered in our simulation since the stable transmission in the wireline networks is assumed and we focus only on the wireless link issues. In other words, the wireline congestion never happens in both with/without adaptive TCP algorithm cases. When the bandwidth variation is from low to high, RTT estimation will rapidly reduce. As a resul, the RTO value will change to avoid the wireline congestion problem, which means once the wireline congestion occurs right after the bandwidth changes from low to high. However, we can see that the problems are resolved with the adaptive algorithm.

5.2 Bandwidth Change Frequency

In this section, we present different bandwidth variation models. Bandwidth model 1 as shown in Figure 5-2 is commonly used [5, 9,10]. IS-2000 also has an enhancement mode for the data transmission, called finite burst. In the finite burst mode, Supplement Channel (SCH) is periodically assigned in a fixed amount of time slots by the terminal.

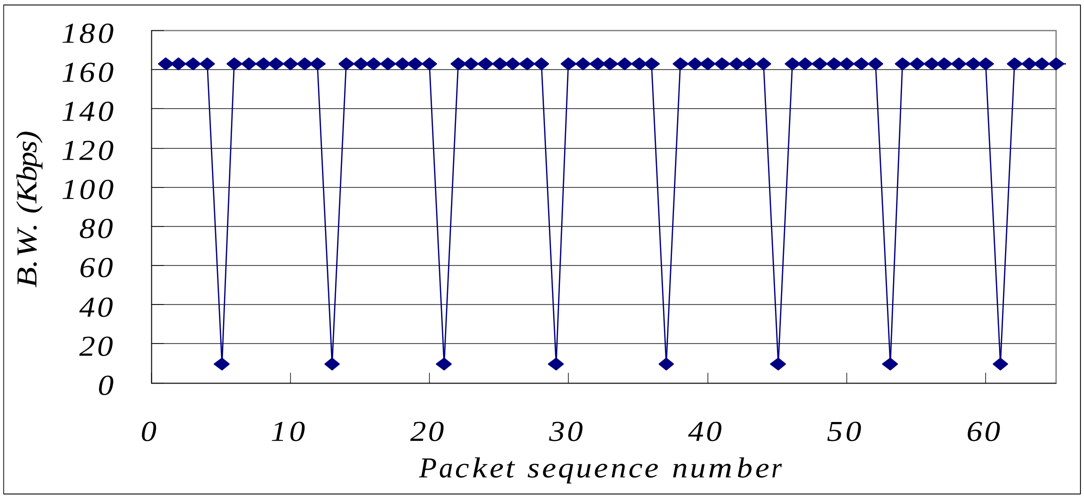

Figure 5-2 B.W. variation model 1

In this finite burst mode, system will assign more resources to the user. However, at the end of data transmission, the extra resource for data transmission will be removed and the RTT will increase, especially, when the abrupt transmission drop from 163.2kbps to 9.6kbps. As a result, in such the case, the bandwidth changing frequency's problem will reveal. Also, the spurious TCP timeouts will be severe. Therefore, we showed that the adaptive TCP algorithm could suppress these spurious timeout successfully.

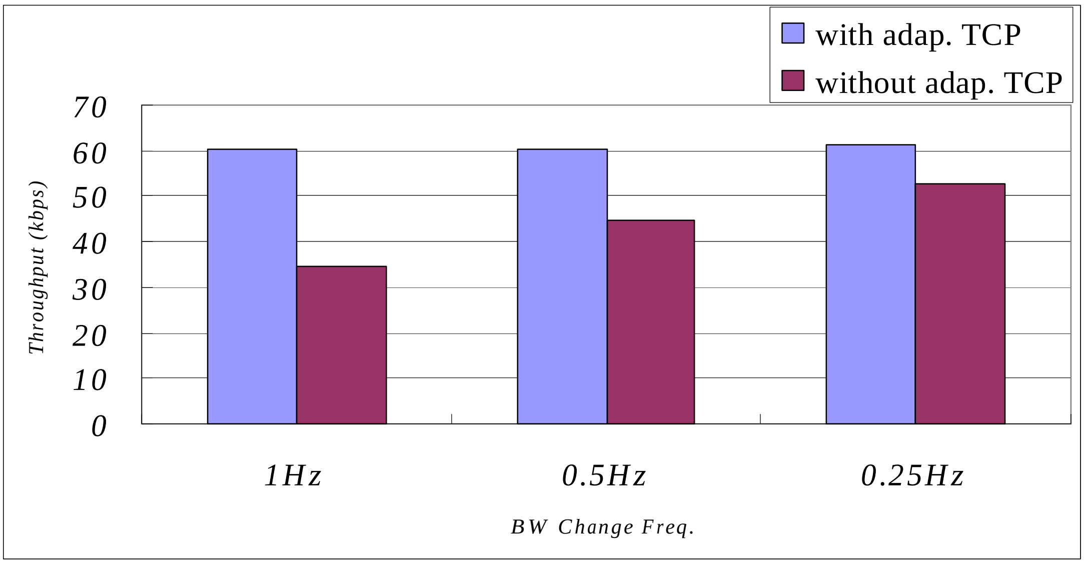

Figure 5-3 Bandwidth change freq. and TCP throughput with FTP

Table 5-1 Improvement with/without Adaptive TCP with FTP using model 1

Change Freq.	1Hz	0.5Hz	0.25Hz
with adap. TCP throughput (kbps)	60.1	60.4	61.4
without adap. TCP throughput (kbps)	34.5	44.8	52.6
Improvement (%)	78	35	16

As Figure 5-3 shown above, we can find that adaptive TCP algorithm improves TCP throughput under different frequnecy of bandwidth variation. Moreover, the throughput with the adaptive TCP algorithm is almost the same because of the suppression of spurious timeouts. As the frequency decreases, the throughput improvement will decrease. Since the probability of spurious timeout, P , is positively correlated with the frequency of bandwidth variation. As mentioned before, the bandwidth is aware of the TCP layer rather than physical layer. This is because the duration of the bandwidth variation could be very fast. However, the time for RTT measurement of a TCP packet is much larger than the time for the bandwidth variation. Therefore, this is why we say that the bandwidth variation is only aware of by the physical layer rather than the TCP layer.

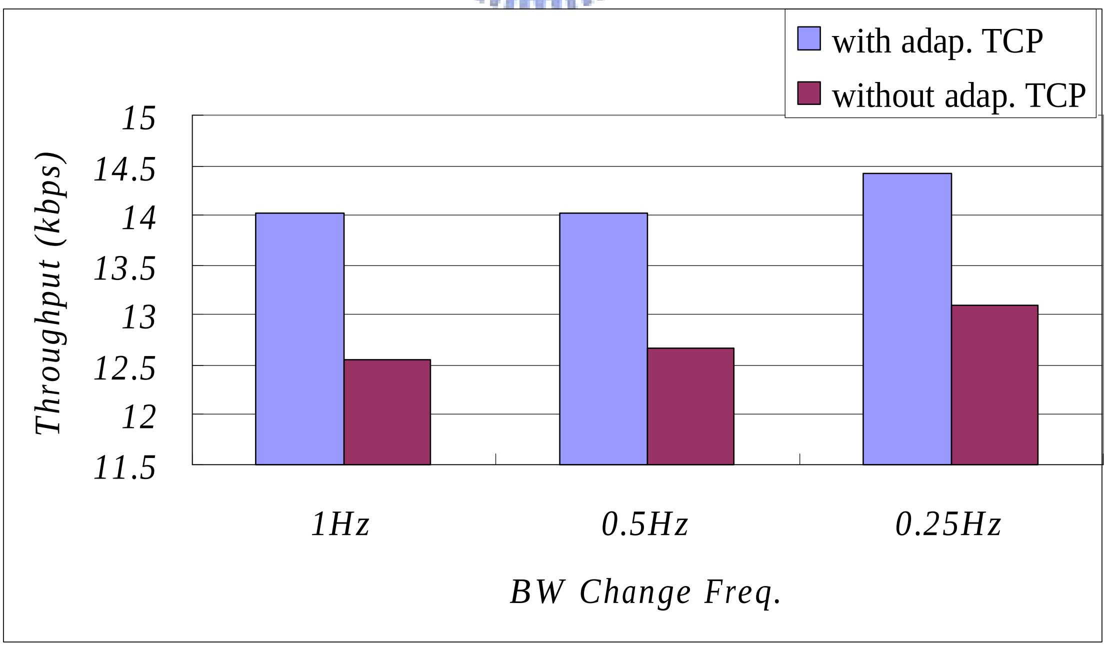

Figure 5-4 Bandwidth change freq. and TCP throughput with HTTP

Figure 5-4 presents the relationship between variation frequency and TCP throughput which uses the HTTP data traffic. By observing the table5-2, we find that HTTP data traffic has the similar improvement distribution.

Table 5-2 Improvement with/without Adaptive TCP with HTTP using model 1

Change Freq.	1Hz	0.5Hz	0.25Hz
with adap. TCP throughput (kbps)	14	14	14.4
without adap. TCP throughput (kbps)	12.6	12.7	13.1
Improvement (%)	11.6	10.7	10.1

The bandwidth model 2 uses different transmission rate such as 9.6, 19.2, 38.4, 76.8, 153.6 kbps which shows in Figure 5-5. The probability of each transmission rate is followed by the normal distribution.

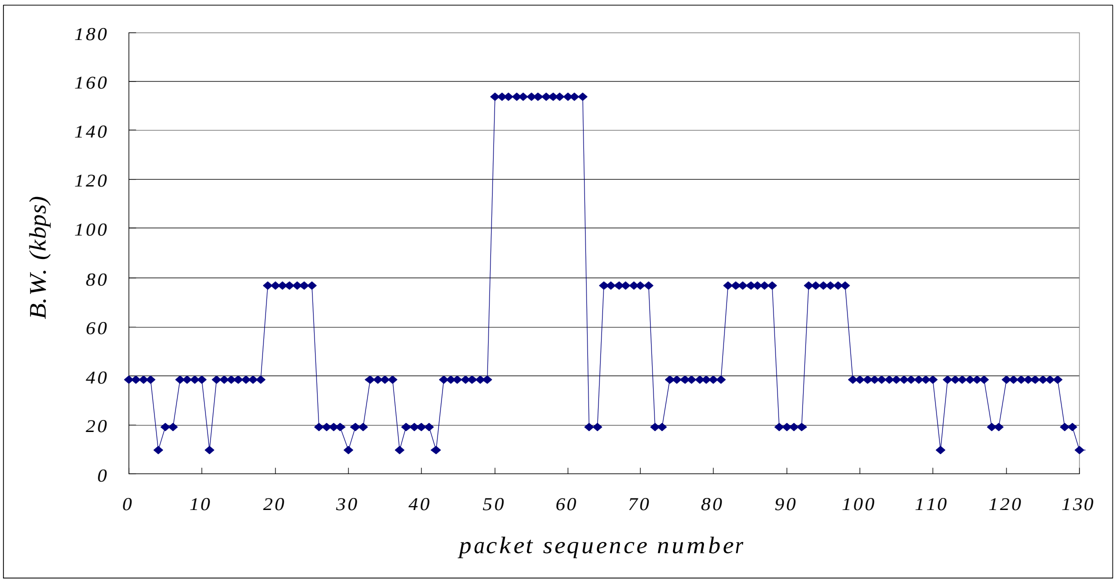

Figure 5-5 B.W. variation model 2

The bandwidth variation model 2 is a more realistic case since the assigned transmission rate is unpredictable.

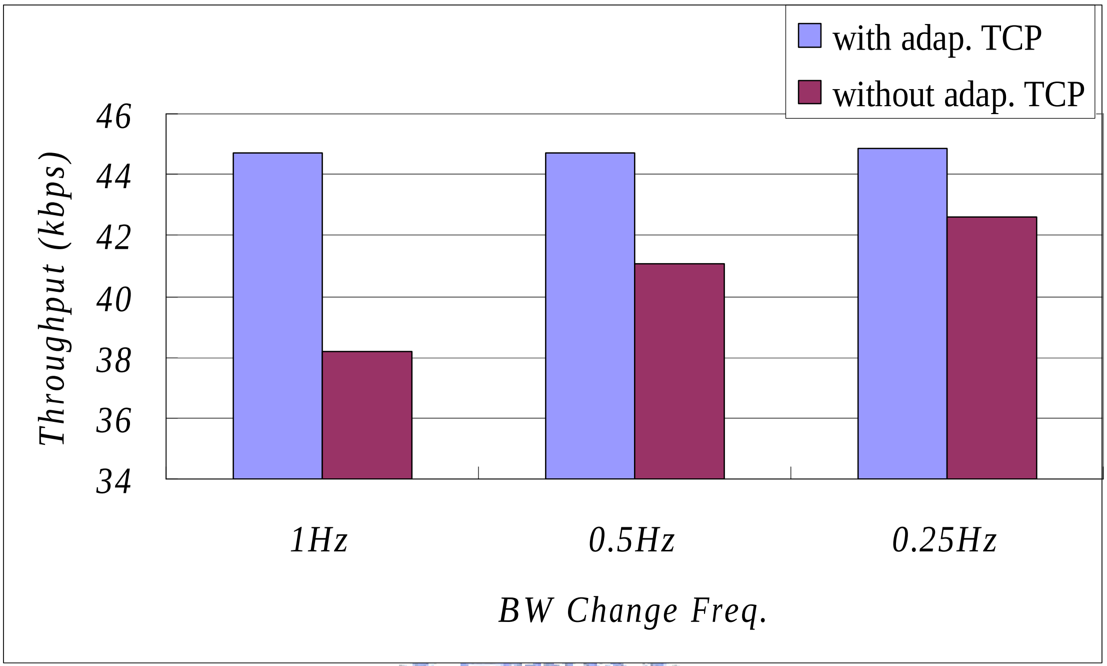

Figure 5-6 Bandwidth change freq. and TCP throughput with FTP

Table 5-3 Improvement with/without Adaptive TCP with FTP using model 2

Change Freq.	1Hz	0.5Hz	0.25Hz
with adap. TCP throughput (kbps)	44.69	44.74	44.82
without adap. TCP throughput (kbps)	38.14	41.05	42.62
Improvement (%)	17.1	9	5.2

Figure 5-6 and table 5-3 show the relationship between the frequency of bandwidth variation and TCP throughput. With the same assumptions, the only difference is bandwidth variation model. Also, the model 2 is adopted in the simulation. It is clear that the system improvement of model2 is smaller than model1. Average transmission rate of model 2 is lower than model 1 but model 2 is more complex than model 1 in the sense of variation. In other words, the transmission rate in next state could not be predictable.

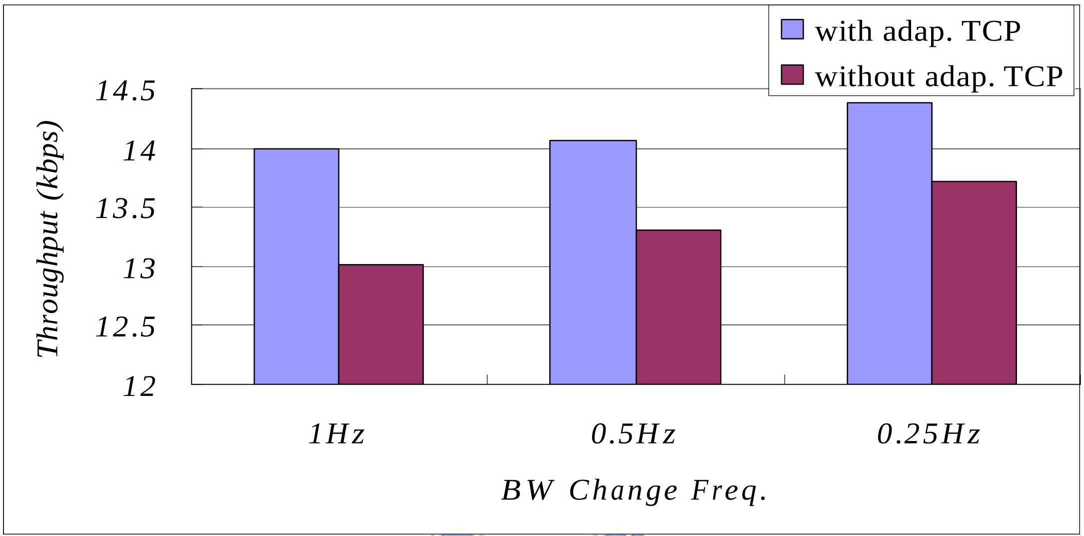

Figure 5-7 Bandwidth change freq. and TCP throughput with HTTP

Table 5-4 Improvement with/without Adaptive TCP with HTTP using model 2

Change Freq.	1Hz	0.5Hz	0.25Hz
with adap. TCP throughput (kbps)	14	14.1	14.4
without adap. TCP throughput (kbps)	13	13.3	13.7
Improvement (%)	7.6	5.7	4.8

By observing the results that in different bandwidth model the HTTP also has less improvement. Because the characteristics of the HTTP traffic which include the reading time, packet call size, embedded packet call size will impact the probability of the timeout occur. Besides the bandwidth change could happen during the different packet call, it will not cause the throughput decrease.

5.3 Standard Deviation of Bandwidth Variation

In this section, we want to study the relationship between TCP system throughput and the standard deviation of bandwidth. As mentioned, the model 2 is a normal distribution; hence we only consider the transmission model in this case. We also present the system improvement in different data traffic, too.

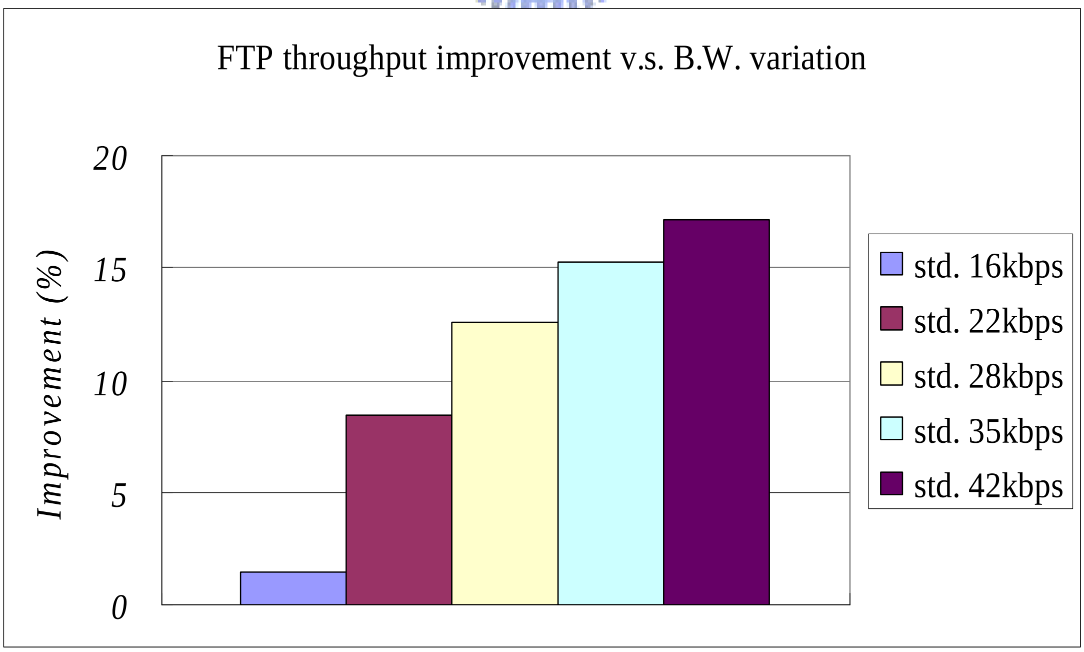

Figure 5-8 Throughput improvement and Standard Deviation

Table 5-5 Throughput improvement and Standard Deviation with FTP

Std. devi. (kbps)	16	22	28	35	42
with adap.TCP throughput (kbps)	41.16	42.24	42.07	43.16	44.69
without adap.TCP throughput (kbps)	40.59	38.96	37.38	37.43	38.15
improvement (%)	1.4	8.4	12.5	15.3	17.2

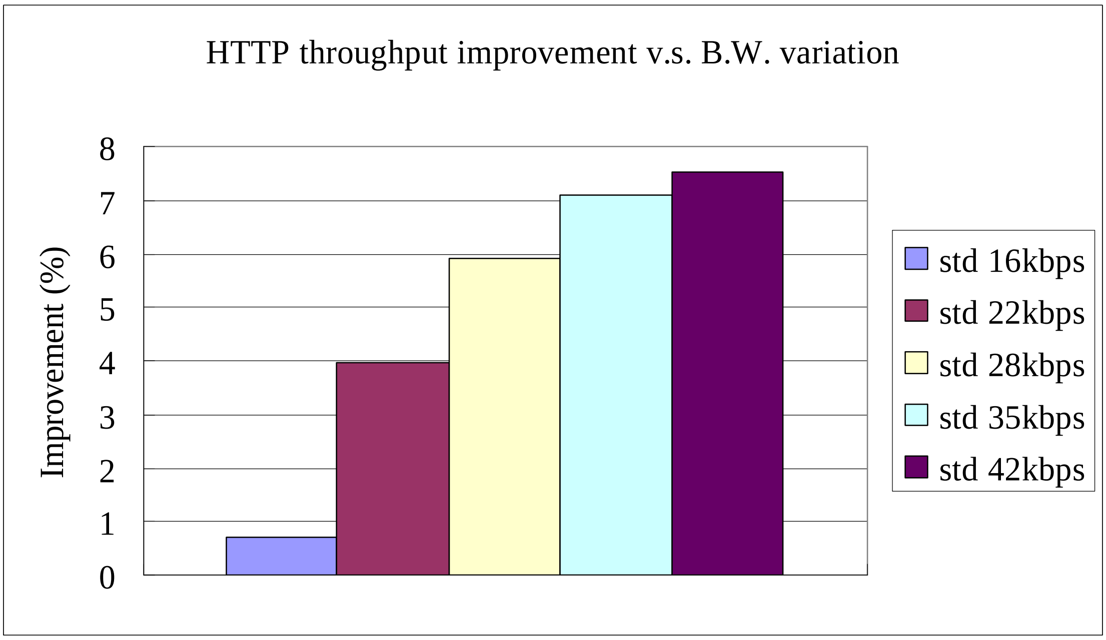

Figure 5-9 Throughput improvement and Standard Deviation

Table 5-6 Throughput improvement and Standard Deviation with HTTP

Std. devi. (kbps)	16	22	28	35	42
with adap.TCP throughput (kbps)	14.19	14.21	13.95	13.9	13.94
without adap.TCP throughput (kbps)	14.09	13.67	13.17	12.98	12.96
improvement (%)	0.7	4	5.9	7.1	7.5

From the Figure 5-8, 5-9 and Table 5-5, 5-6, we can find that when the standard deviation is larger and throughput improvement is larger as well. Larger deviation of transmission rate means the transmission rate will has larger probability of changing from high to low or from low to high. By observing the result, we know that transmission rate variation should be as small as possible. In this way, the fading effect and bandwidth variation will be mitigated. Therefore, the smaller deviation is considered to be suitable for the fading channel.

From the above we can conclude that the spurious timeout is relative to standard deviation of bandwidth variation. Hence if the STD of bandwidth variation is limited below some range like 9.6kbps or 19.2kbps, the deviation of RTT will be small and the spurious timeout will be avoided. In wireless network, the channel condition and system available resource is not expected. But if the system can reduce the probability of larger BW variation, the impact of fading channel on the throughput will be reduced.

5.3 FER (frame error rate)

We are going to discuss the different FER in this section. The improvement of system throughput is discussed in different FER with the adaptive TCP algorithm. . High FER will not only cause the RTT increasing but increase the total time of reception of a FTP call. Figure 5-10 presents the system throughput improvements under different FER cases. And table 5-7 shows the system throughput with/without adaptive TCP algorithm in different FER conditions.

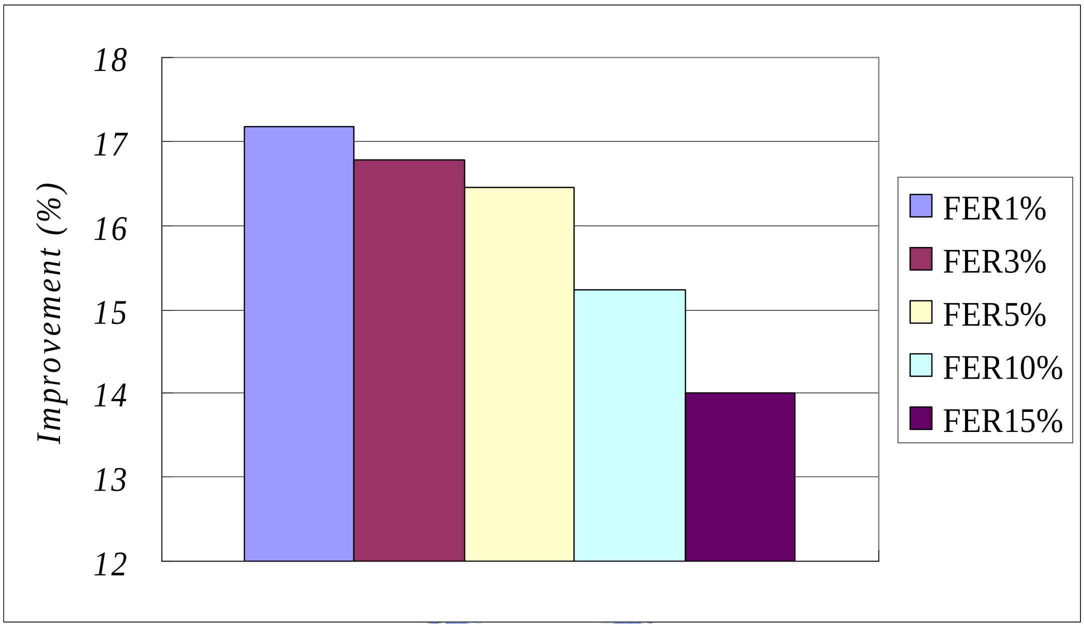

Figure 5-10 FTP throughput improvement in different FER

Table 5-7 Summary of FTP throughput with/without adaptive TCP V.S. FER

FER (%)	1	3	5	10	15
with adap.TCP throughput (kbps)	44.69	44.29	43.96	43.08	42.29
without adap.TCP throughput (kbps)	38.15	37.92	37.75	37.39	37.09
improvement (%)	17.2	16.8	16.5	15.2	14

By observing the figures and tables, we know that the throughput at higher FER will be worse than the case at lower FER. Besides, the throughput improvement will decrease as the FER increases. The reasons are as following. Higher FER causes RTT increase and further it might cause the TCP timeouts. Finally the throughput will decrease. And the adaptive TCP algorithm can avoid spurious timeout so that cwnd can maintain higher. On the other hand, the average number of cwnd without adaptive algorithm is smaller than the case with the algorithm. However, once the higher FER causes TCP packet to timeout, the damage on that which with adaptive TCP will be larger than the case without the adaptive algorithm. That is why the improvement will decrease with FER increase. Figure 5-11 and table 5-8 presents the FER impacts on the throughput in the data traffic of HTTP.

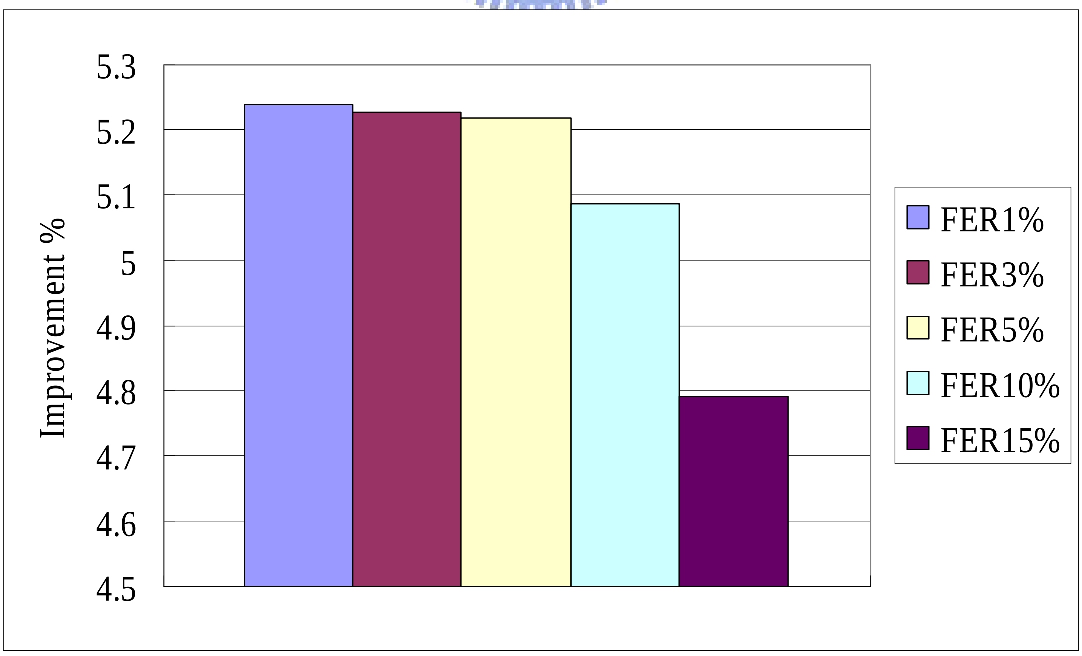

Figure 5-11 HTTP throughput improvement in different FER

Table 5-8 Summary of HTTP throughput with/without adaptive TCP V.S. FER

FER (%)	1	3	5	10	15
with adap.TCP throughput (kbps)	11.93	11.88	11.83	11.75	11.67
without adap.TCP throughput (kbps)	11.34	11.29	11.24	11.18	11.14
improvement (%)	5.24	5.22	5.21	5.09	4.79

From the results we can conclude that the high FER at wireless connection can not be avoided. Even if by usage of adaptive TCP algorithm the throughput will decrease with the increase of FER. So all of that we can do is to decrease degree the impact of FER. Power control is one of the solutions. Another solution could be to reduce the time of retransmission, like using the MAC retransmission.

Chapter 6

Conclusion

In this thesis, our contributions are (1) establishing a TCP & RLP simulation platform (2) proposing an Adaptive TCP RTO algorithm to eliminate error timeout triggers in wireless communication system (3) providing sensitivity studies in bandwidth changing frequency & the STD of BW changes. At the beginning, we showed that the BW changing frequency has negative relationship to the system throughput. Decreasing the probability of BW change will make the data transmission more stable and the timeout will be decreased. And the STD of BW variation is another important factor which is relative to the system throughput. When the STD is larger, the probability of transmission rate change from high to low or from low to high will also increase. The STD of bandwidth variation should be limited below some range like 9.6kbps or 19.2kbps, the deviation of RTT will be small and the spurious timeout could be avoided. Finally the FER causes the throughput decrease, and it can not be avoided. Even if by usage of adaptive TCP algorithm the throughput will decrease with the increase of FER. So power control and reduce the time of retransmission (like using the MAC retransmission) could be another solutions for this.

References

[1] W. Stevens, "TCP/IP Illustrated Volume l," Addison Wesley, 1994

[2] Postel, Jon, ed. Transmission Control Protocol; RFC793. In ARPANET Working Group Requests for Comments, no. 793. SRI International, Menlo Park, Calif., Sep. 1981

[3] Mo, J.; La, R.J.; Anantharam, V.; Walrand, J.; "Analysis and Comparison of TCP Reno and Vegas" INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE , Volume: 3 , 21-25 March 1999, Pages:1556 - 1563 vol.3

[4] Kevin Fall and Sally Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP" ACK SIGCOMM Computer Communication Review archive Volume 26, Issue 3 (July 1996) table of contents. Pages: 5-21. Year of Publication:1996. ISSN:0146-4833

[5] Khafizov, F.; Yavuz, M.; "Running TCP over IS-2000," Communications, 2002. ICC 2002. IEEE International Conference on , Volume: 5 , 28 April-2 May 2002 Pages:3444 - 3448 vol.5

[6] Yong Bai; Ogielski, A.T.; Gang Wu, "TCP over IS-707" in Proc. of PIMRC'99, Sept. 1999

[7] Khan, F.; Kumar, S.; Medepalli, K.; Nanda, S., "TCP performance over cdma2000 RLP," Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st , Volume: 1 , 15-18 May 2000

[8] Yong Bai; Ogielski, A.T.; Gang Wu, "Interactions of TCP and radio link ARQ protocol"; Vehicular Technology Conference, 1999. VTC 1999 - Fall. IEEE VTS 50th , Volume: 3 , 19-22 Sept. 1999, Pages:1710 - 1714 vol.3.

[9] Abouzeid, A.A.; Roy, S.; Azizoglu, M.; "Comprehensive performance analysis of a TCP session over a wireless fading link with queueing" Wireless Communications, IEEE Transactions on , Volume: 2 , Issue: 2 , March 2003 Pages:344 - 356

[10] Cheng Peng Fu; Liew, S.C., " TCP Veno: TCP enhancement for transmission over wireless access networks" Selected Areas in Communications, IEEE Journal on , Volume: 21 , Issue: 2 , Feb. 2003 Pages:216 – 228

[11] Chockalingam, A.; Zorzi, M.; Rao, R.R., "Performance of TCP on Wireless Fading Links with Memory," Communications, 1998. ICC 98. Conference Record.1998 IEEE International Conference on , Volume: 1 , 7-11 June 1998 Pages:595 - 600 vol.1

[12] Chaponniere, E.; Kandukuri, S.; Hamdy, W., "Effect of physical layer bandwidth variation on TCP performance in cdma2000," Vehicular Technology Conference,

2003. VTC 2003-Spring. The 57th IEEE Semiannual , Volume: 1 , 22-25 April 2003

[13] Yavuz, M.; Khafizov, F., "TCP over wireless links with variable bandwidth," Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th , Volume: 3 , 24-28 Sept. 2002 Pages:1322 - 1327 vol.3

[14] M. Chan; R. Ramjee, "TCP/IP Performance over 3G Wireless Links with Rate and Delay Variation, "Mobicom'02,

[15] Meyer, M.; Sachs, J.; Holzke, M., "Performance evaluation of a TCP proxy in WCDMA networks," Wireless Communications, IEEE [see also IEEE Personal Communications] , Volume: 10 , Issue: 5 , Oct 2003 Pages:70 – 79

[16] M. Allman, V. Paxson, W. Stevens, "TCP Congestion Control, RFC 2581 ," April 1999

[17] M. Allman, V. Paxson, "On Estimating End-to-End Network Path Properties," ACM SIGCOMM, September 1999

[18] H. Jiang, C. Dovrolis, "Passive Estimation of TCP Round-Trip Times" ACM SIGCOOM, July 2002

[19] P. Karn, C. Partridge, "Improving Round-Trip Time Estimates in Reliable Transport Protocols, " ACM SIGCOMM, August 1987

[20] Yee, J.R.; Weldon, E.J., Jr, "Evaluation of the performance of error-correcting codes on a Gilbert channel"; Communications, IEEE Transactions on, Volume: 43, Issue:8 , Aug. 1995, Pages:2316-2323

[21] A. Bakre, B.R. Badrinath, "I-TCP:Indirect TCP for Mobile Hosts," in Proc. of the 15th International Conference on Distributed Computing Systems, Vancouver, BC, May 1995

[22] 1xEV-DV Evaluation Methodology – Addendum (V6). 3GPP2, July 25, 2001

[23] Larry L. Peterson, Bruce S. Davie "Computer Networks",Margan Kaufmann

Appendix A

TCP are informed of some important information by its header. The header is added at the front of the data. Each layer has its own header, and the header is used to indicate how to process the received data. The block of data which TCP passes to IP, including the TCP header and applications data, is usually called a **segment**. TCP header is shown in Figure A-1.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |           |U|A|P|R|S|F|                               |
| Offset| Reserved  |R|C|S|S|Y|I|            Window             |
|       |           |G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure A-1 TCP header

Source port field length is 16 bits and it shows the ID of the sender’s application. And so does the **Destination port field**, the only difference is it shows the ID of the receiver’s application. Using the two fields we can use many applications through the TCP transferring data at the same time. **Sequence number field** has 32bits, it is used for error control. It records the sequence number of the sending segment to identify from other segments. And the sequence number is assigned randomly in the beginning, it is in order to prevent using the same sequence number in the network. Each time the segment sent the sequence number will be increased by 1. According the sequence number, even if the segment be send or arrive out of order, it will be put the correct position. **Acknowledgement number field** is used to acknowledge the correct segment sequence number. But there one thing must be noticed, the number of the field is not correct received segment number. It shows the number for the next time the receiver expects to receive. In other words the number of the segment which is correctly received in sequence is the acknowledgement number-1. It also expresses the segments up to the acknowledgement number-1 are correctly received. The acknowledgement number is only valid if the ACK flag is set. **Data offset** shows the TCP header size. Flags are used to indicate the validity of other fields and connection control. The flags are URG、ACK、PSH、RST、SYN、FIN. Each flag is only one bit. **ACK** indicates the acknowledgement number field is valid and it is usually on. **PSH** indicates the receiver TCP layer to pass the segment to application layer immediately. **RST** is used to reset the connection. **SYN** is synchronization flag and it is used at the beginning of the connection setup. That means when the connection setups, the flag is set to be one. **FIN** is used when the connection is closed. **URG** indicates the urgent pointer field is valid. It means the data should be process before any other data. **Window field** is 16 bits and it show how much buffer space does this connection allow to send. **Checksum** is used for check the data and the header. **Urgent pointer** field show the data should be considered as urgent and requires process immediately. This field is only valid if the URG flag is set. **Options** field size is variable length. Mostly it is used to show the Maximum Segment Size (MSS), it means to tell the client TCP layer the maximum size of the segment should be sent.