私　立　東　海　大　學

資 訊 工 程 研 究 所

碩　士　論　文

指 導 教 授：呂 芳 懌 博士

基於往返延遲的串流控制傳輸協定路徑切換方法

A Path Switching Scheme for SCTP Based on Round Trip Delays

研 究 生：陳 永 倫

中 華 民 國 九 十 九 年 一 月

摘要

由於網際網路迅速地發展，網路已經成爲我們日常生活中不可或缺的角色。使用者總是希望網路速度在存取時沒有延遲，但是網路研究在未來的方向不僅要考慮到更快的速度，更要考慮到網路的堅固性並且減少封包的遺失。目前新的網路串流控制傳輸協定 Stream Control Transmission Protocol (SCTP) 提供了 Multi-homing 以及 Multi-streaming 兩項重要的功能。Multi-homing 提供了多條預備路徑使得再網路發生斷線時能夠即時切換到另外一條來傳輸。Multi-streaming 可以允許一條路徑使用多條串流來提升效率。在此論文中，我們提出了新的切換路徑方法。當發生網路斷線時，我們考慮了加密封包、擁塞窗口、路由路徑、頻寬大小等因素來計算封包來回延遲的時間，藉以判斷出應該選擇的最佳路徑。

關鍵字：串流控制傳輸協定、擁塞控制、重傳、路由路徑、頻寬

Abstract

Due to the rapid development of network applications, today the Internet plays an important role in our everyday life. Users hope that the network is always speedy enough to help them access the Internet without any delay. But, the real situation is far from the ideal case. In the future, network researchers will continuously improve network speed, and try to develop networks that are robust, without any crashes or packet loss. Recently, a relatively new protocol, called stream control transmission protocol (SCTP), which provides multi-homing and multi-streaming features, was released. With the former feature, when the original transmission path is unstable or fails, the sender and receiver nodes can quickly change transmission path to another one to prevent the transmission from crash or resulting in poor transmission quality. With the multi-streaming feature, a node can separately transmit packets to another node or nodes without occurrence of the head-of-line problem. In this paper, we propose an aggressive path switching scheme for SCTP. Before data transmission, the scheme selects the fastest path as the primary path to transmit packets. When the path fails or transmission quality is poor, this scheme evaluates all alternate paths, and selects the one with the best quality as the new primary to substitute for the original one. After that, packets are delivered through the new path. Several factors are considered in the evaluation, including bandwidth, encryption/decryption, size of the congestion window, retransmission policy, routing policy, etc.

Keywords: SCTP, congestion control, retransmission, routing, bandwidth

Table of Content

List of Figures

List of Tables

Chapter 1: Introduction

In recent years, stream control transmission protocol (SCTP) and its applications have been widely deployed and quickly developed, respectively. Its importance in wireless communication is greater every day. Leu [1] employed SCTP as a key mechanism of network mobility to achieve a seamless handover, particularly for delivering multimedia data. Many modified versions, like CS-SCTP [2] and nSCTP [3], have been proposed. However, most SCTP systems consider only a few performance-affecting factors, implying that their performance can be further improved. In fact, a packet travels through several network layers before it successfully arrives at its destination [4]. We consult the OSI model [4] as our reference model. When a transmission starts, packets flow from the application layer to the physical layer, and then go across switches/routers. When the packets arrive at the receiver, they go in the reverse direction from the physical layer to the application layer. This is a complicated transmission process in which the transport layer plays an important role in handling flow control.

Current TCP and UDP protocols have several drawbacks, e.g., head-of-line blocking [5] and denial-of-service attacks [6]. This study aims to improve the performance of the SCTP protocol. Our opinion is that the SCTP improvement should not be limited to the transport layer. Also, a factor may be affected by others, which means a factor may be a function of other factors. For example, current available bandwidth is affected by the size of the sender's congestion window.

In this study, we develop a new path switching scheme for SCTP, called the path selection and switching process (PSASP), which chooses the best path for SCTP by evaluating mechanisms and activities that influence on SCTP transmission efficiency, including size of encrypted/decrypted data [2], size of congestion window [7][8], retransmission policies [9], length of a routing path [10], a packet's round trip time (RTT) [11], network delays [12], hardware speed and bandwidth, etc. These mechanisms and activities are dispersed in layers of the OSI model. For example, routing is a layer-three task, and hardware speed is a layer-one concern. We also propose a path switching scheme based on evaluation of the results of the related mechanisms and activities, which can help the SCTP to select primary path. Experimental results show that this scheme can truly select the best path. The contributions of this research are as follows:

(1) The PSASP evaluates cross-layer mechanisms and activities to select a primary path for SCTP.

(2) We derive PSASP's cost model, including the processing delay, transmission delay, propagation delay and queuing delay, each of which is evaluated based on the cross-layer mechanisms and activities.

(3) We calculate the total cost for the PSASP when retransmission k times is considered given a path's retransmission probability, k=0,1,2…n.

This article is organized as follows. Section 2 introduces relevant background and related work. Section 3 describes our system architecture. The experimental results are presented in section 4. Section 5 concludes this article and addresses our feature work.

Chapter 2: Background and Related Work

2.1. SCTP

The SCTP inherits features and attributes from TCP, but provides new features for users [13], including multi-homing, multi-streaming, heartbeat, four-way handshake, and chunk bundling.

(1) Multi-homing: with this, the SCTP establishes an association between sender and receiver before transmitting packets. An association often contains multiple paths, each of which is an ip-to-ip connection. Therefore, this protocol needs multiple IPs. Initially the SCTP chooses a path as the primary path to transmit packets. When transmission quality is poor, it chooses the secondary path (know as alternate path) to substitute for the primary path. With multi-homing, SCTP transmission is more reliable than that of TCP and UDP.

(2) Multi-streaming: this divides a path into multiple subpaths, called streams. All streams are independent of each other in transmission. Before data transmission, SCTP defines a number of streams and assigns packets to streams for transmission to prevent the head of line problem [5].

(3) Heartbeat: this is implemented for each node to periodically send packets telling other nodes that it is still active. Through heartbeats, a node can know which paths are currently available.

(4) Four-way handshake: this is used to establish a connection. Before data transmission, the sender sends an INIT to the receiver. The receiver on receiving the INIT responds with an INIT-ACK which includes a state cookie and connection information, neither saving state information, nor allocating resources for the connection. Next, the sender replies with a corresponding COOKIE-ECHO to

confirm the state cookie. After the confirmation, the receiver replies with a COOKIE-ACK. After that, an association is established and the sender can transmit data to the receiver. Meanwhile, the receiver allocates cpu time and memory capacity to the association.

(5) Chunk bundling: this is related to the SCTP packet format. A SCTP packet includes control chunks and data chunks. Control chunks carry information for SCTP controlling. Data chunks convey data messages. The SCTP can bundle several small chunks into a big one, or vice versa. However, the packet size cannot in any circumstance exceed the maximum transmission limit.

2.2. The SCTP variations and applications

There are several SCTP variations. Mobile-SCTP (mSCTP) [14], an extension of SCTP, is used for mobility management in a wireless environment. It allows an endpoint to add, delete and change IPs by sending address configuration (ASCONF) messages to its peer while their SCTP association is still active. An SCTP-based handoff scheme is designed by using the mSCTP protocol. The SCTP can use the multi-homing feature to improve handoff problem.

Satellite networks are global internet that provides broadband transmission, television, and navigation services. A number of satellite link characteristics may limit the performance of transport protocols over satellite networks. Fu et al. [15] investigated and evaluated the SCTP features that can be exploited to increase the satellite network performance.

The Internet Protocol Television (IPTV) has been regarded application in the next generation networks. Kim et al. [16] used the SCTP to support the real-time IPTV. Through the multi-streaming, the streams can dispatch stream 0 as service manager, stream 1 as channel 1, …, stream n as channel

n. All the channels can easily transfer with different stream identifiers. It actually reduces the impact of the head of line problem.

2.3. Related Work and Influential Factors

According to previous studies [2,7-11], network transmission is influenced by several factors. Yang, Chang and Huang [2] mentioned that encryption, due to requiring additional overheads, makes a data chunk include much more information than transmitting plain text does. The overheads consume extra packet processing time and transmission time. Generally, a relatively higher security level often generates more overheads than a lower level one does. Kim et al. [7] pointed out that different congestion-window increasing/shrinking polices result in different throughputs. Qiao et al. [9] described how retransmission policies, e.g., different parameters such as Path.Max.Retrans (PMR) and Retransmission Time-Out (RTO), cause different failover performance.

Routing policies, e.g., static and dynamic, also affect transmission performance since different policies select different paths for data transmission. Hassan et al. [10] analyzed two routing protocols: proactive (table-driven) and reactive (on-demand). Proactive protocols, such as Destination-sequence Distance-vector Routing (DSDV) [17], maintain routing information by periodically exchanging routing-table contents with neighbors, whereas reactive protocols, such as Ad hoc On Demand Distance Vector (AODV) [18] and Dynamic Source Routing (DSR) [19], build routing paths when they need to route packets.

Dahel and Saikia [11] stated that round-trip time (RTT) which responds to the current available bandwidth of the path can help to determine how to adjust congestion window to transmit data. The RTT based congestion avoidance (RBCA) Scheme that calculating RTT on receipt of each SACK uses the Timestamp option. It adjusts the *cwnd* by $CwndIncr = DSize * \frac{RTT_{threshold} - RTT}{\max(RTT, RTT_{threshold})}$. Where the *DSize* is the minimum of the number of newly ACKed byte in a SACK and maximum packet size. The $RTT_{threshold}$ is base RTT * *F,* where *F*>1. The RTT value is monitored for every SACK continuously that arrives and the *cwnd* size is updated accordingly.

Ribeiro et al. [12] stated that symmetric paths and asymmetric paths perform differently. The delays of asymmetric paths are usually shorter because they choose the path with the lowest delay to transmit packets. Al-kaisan et al. [20] presented a modified version of the SCTP, called the optimized SCTP, which uses modified congestion control to improve SCTP performance. It proposed the new modifications in the SCTP congestion control approach when a packet has been retransmitted by the fast retransmission procedure. Only the 1st detection of a lost packet will cause the path variable to be changed. Once the variables are changed, they will not be changed by any lost message until a transmission time-out occurs. After this, it will change the value of path variables for the lost message. A packet lost detection will cause to reduce the *cwnd* relatively slowly, i.e., *cwnd* = *cwnd* – [0.05 * *cwnd*] instead of reducing it by half.

Further, the standard SCTP does not clearly define how to select one of the alternate paths as the primary path. In fact, the round-trip time is a good method to evaluate alternate paths. But, round-trip delay is a complicated delay consisting of many path-performance affecting factors which are dispersed among different network layers. A round-trip delay of a path actually reflects the real condition of a path since a packet and its acknowledgement are really delivered through the path. In this study, we will analyze how the factors affect path performance. Based on the analysis, we can then select the best path as the primary path to delver messages. The path switching scheme is important, because the multi-homing's feature is connecting several paths. The different paths may own different transmission performance. It also affects the transmission speed, path maximum retransmission count and retransmission time out. Choosing the fast path can decrease the retransmission probability and improve the total transmission performance. So, we develop the new path switching scheme to choose the fastest path.

Chapter 3: The Proposed Scheme

In a multi-homing environment, the PSASP has two main steps in selecting the best path for an SCTP association. Step 1 is selecting an initial primary path. Before data transmission, the PSASP first checks to see whether any network flow flows through the path with the widest bandwidth or not. If not, the path will be selected as the initial primary path. Otherwise, the PSASP enters step 2 which evaluates performance for all paths of the association. In addition, when transmission fails or communication quality is poor, the PSASP will also invoke the step-2 process. But, this time, only the remaining paths of the association are evaluated. In this process, dynamic influential factors are involved to compute the packet delivery delay of a transmission path. The one with the minimum delay or default path will be selected as the initial or the new primary path. Fig. 3-1 shows the flow chart of the primary path selection. In the following, we assume that 1) the bandwidth of an association and that of each path involved are known; 2) initial bandwidth = current_available bandwidth + occupied_bandwidth; 3) packet arrival rate of each path segment along a path, e.g., the path segment between nodes i and i+1, follows Poisson distribution.

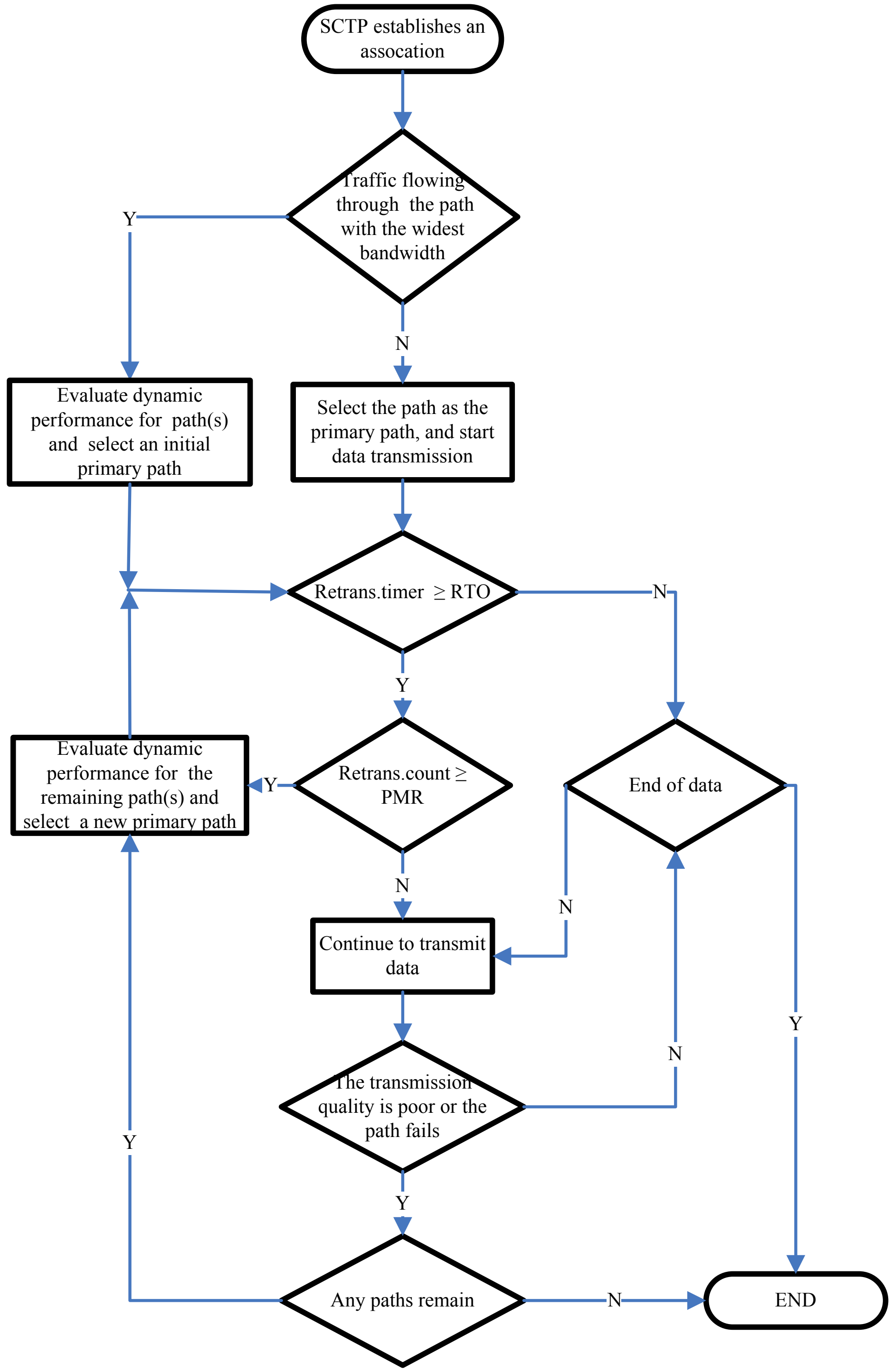

Fig. 3-1 The flow chart for selecting an initial and a newly selected primary path

3.1. Dynamic Factors

The following mechanisms and activities, including switchover/retransmission policies, size of encrypted/decrypted data, size of congestion window and round-trip delay, are considered as key factors in selecting a primary path.

3.1.1 Switchover/retransmission policies

There are two main factors that strongly influence on retransmission policies. One is PMR which is the maximum retransmission counts of a path. The other is RTO which is the counted time of a retransmission period. Fallon et al. [9] claimed that PMR and RTO should both be considered before an appropriate switchover/retransmission policy can be ensured. That is, when retransmission timer exceeds the RTO, an underlying packet will be retransmitted. When the retransmission count is over the PMR (i.e., a path's transmission-failure count ≥ PMR+1), implying the quality of the path is poor or the path fails, a new path will be selected, and the SCTP will switch over to the new path to continue delivering packets for the sender. Often, the default value of PMR is 5, and that of RTO is 60 sec.

3.1.2 Size of encrypted/decrypted data

According to [2], an encrypted packet has a longer delivery delay than its original packet has because of additional processing efforts, such as packet encryption and decryption, and additional transmission overheads. Generally, encrypted packets can be classified into four security levels [2]. Level 0 does not provide any security facilities. Level 1 provides authentication and integrity checking for established associations. With level 2, only a part of chunks, instead of the whole, is encrypted. Level 3 provides encryption, authentication and integrity checking for all chunks. Higher security levels often have more overheads.

3.1.3 Size of congestion window

According to [7], when the size of a congestion window is relatively larger, implying the transmission path has better quality, and current available bandwidth defined as initial_bandwidth – traffic_occupied bandwidth is wider, then a sender can transmit more data per second to receiver. When the window size is small, it often means the available bandwidth of the path is limited, and the network quality is not good. Once packets are lost, the window size will be reduced to mitigate data flow and shorten the packet waiting time. In this case, the SCTP can only use a portion of currently available bandwidth to transmit packets. In this study, we assume that available bandwidth = initial_bandwidth – traffic_occupied bandwidth – SCTP_occupied bandwidth where SCTP_occupied bandwidth results from shrunken congestion window size. If packets can be successfully and continuously delivered to the destination, the window size will be slowly enlarged, which is known as a slow start.

3.2. Round Trip Delay

According to [11], round trip time/delay is the time period from when the sender sends a packet to receiver to the time point when the sender receives the reply. The round trip time as stated above more accurately reflects real network speeds. Many systems employ it as an important performance parameter. A shorter round trip time implies the network transmission speed is high. Ribeiro et al. [12] used the round trip time to judge the paths. But, the authors did not analyze details of the delay. In this study, we consider round trip delay as one of the most important performance-measure parameters. The delay can be further divided into processing, transmission, propagation and queuing delays.

3.2.1 The timings of delivering a data packet

In the following, we assume the SCTP association has H paths, and a concerned path has n+2 nodes, including the source node, denoted by S_0, the destination node, denoted by S_{n+1}, and n intermediate nodes (i.e., n routers), denoted by $S_1, S_2, \ldots, S_n$. Definitions of the concerned terms are listed and described in Table 3-1. Their formal expressions will be expressed in the following.

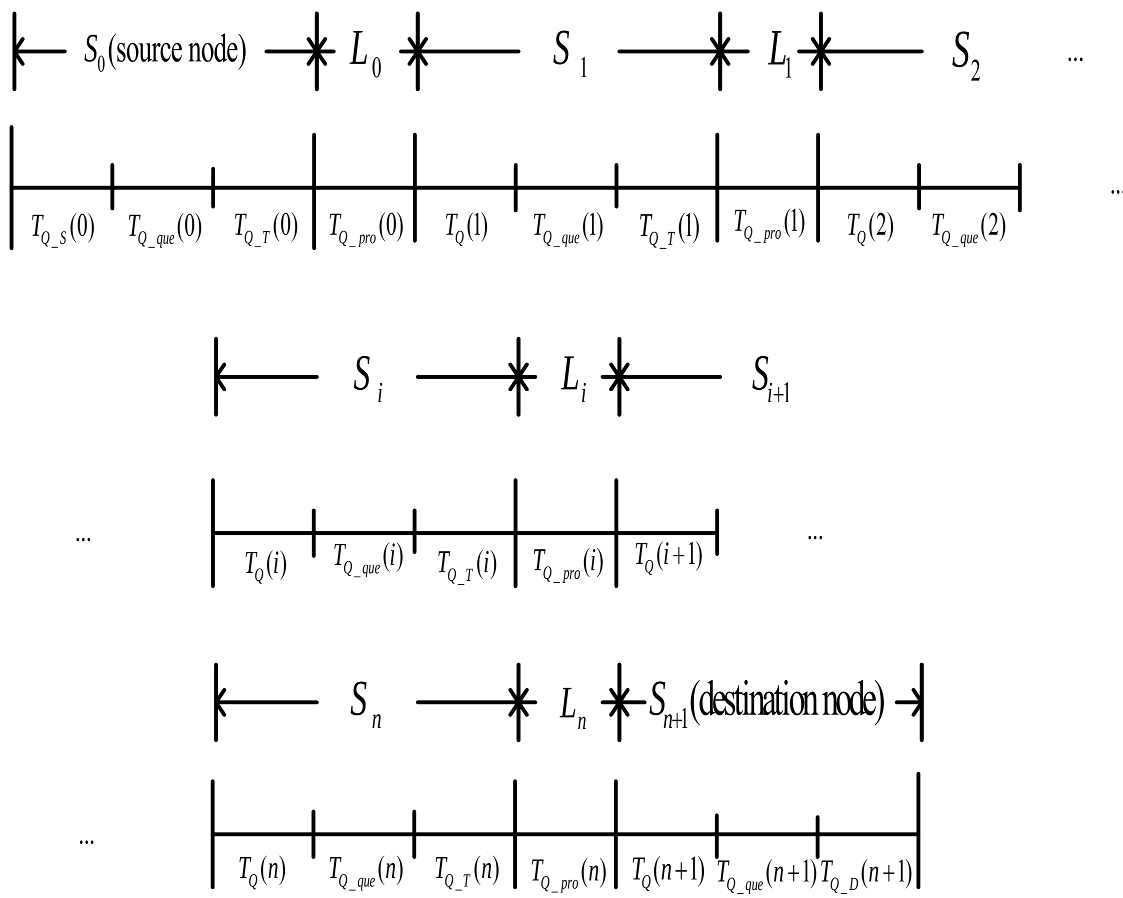

Fig. 3-2 The Timings of a path with n+2 nodes ($S_0, S_1, ..., S_{n+1}$) in which S_0 is the source node which generates a packet Q. S_0 is also the destination of the corresponding ACK. S_{n+1} is the destination node of Q and S_{n+1} is also the source node of the ACK.

Table 3-1. Definitions of concerned terms on data packet Q

Items	Description
$T_{Q\_S}(0)$	The time required by S_0 to generate and encrypt a packet Q – processing delay at source node
$T_{Q\_que}(i)$ i=0,1,2,…n+1	The time Q waiting in S_i's message queue – (waiting time)
$T_{Q\_T}(i)$ i=0,1,2,…n	The time required by S_i to transmit Q from its first bit to last bit – transmission delay
$T_{Q\_pro}(i)$ i=0,1,2,…n	The time required by Q to propagate from S_i to S_{i+1} through link L_i. It is defined as the time from when a bit is sent out by S_i to the time point when the bit arrives at S_{i+1} – propagation delay
$T_Q(i)$ i=0,1,2,…n+1	The time required by S_i to receive Q. It is defined as the receiving time period from when Q's first bit arrives at S_i to the time point when Q's last bit arrives at S_i – receiving delay

$T_{Q\_D}(n+1)$	The processing time required by S_{n+1} to process *Q* (i.e., to decrypt *Q*) – processing delay at destination node

In Fig. 3-2, S_0 first generates a packet (i.e., *Q*) which will be delivered to S_1 through link 0, denoted by L_0. The time required to generate and encrypt a packet by S_0 is $T_{Q\_S}(0)$. If $S_0's$ packet generating speed is higher than the delivery speed, packets will be queued in $S_0's$ message buffer. The time a packet waiting in $S_i's$ message buffer is $T_{Q\_que}(i)$. All *n*+2 nodes have their own queues. That is why the indexes of $T_{Q\_que}(i)$ are from 0 to *n*+1. The time required to transmit *Q* by S_i is $T_{Q\_T}(i)$. Only $S_0, S_1, ..., S_n$ transmit packets to their immediate downstream nodes. Therefore, the indexes of $T_{Q\_T}(i)$ are between 0 and *n*. Once *Q* is delivered by S_i, it will travel through L_i to S_{i+1}. The time required by a bit to propagate from S_i to S_{i+1} through L_i is $T_{Q\_pro}(i)$. *Q* should travel through *n*+1 links ($L_0, L_1, ..., L_n$) before it can arrive at S_{n+1}. So, the indexes of $T_{Q\_pro}(i)$ are between 0 and *n*. When the first bit of *Q* arrives at S_i, S_i starts receiving *Q*. The time required to receive *Q* at S_i is $T_Q(i)$. Only $S_1, S_2, ..., S_{n+1}$ receive *Q* from their upstream nodes. So, the indexes of $T_Q(i)$ are between 1 and *n*+1.

3.2.2 The timings in delivering an ACK

The timings in delivering an *ACK* and their indexes are shown in Fig. 3-3. The definitions of these terms are listed in Table 3-2. Their descriptions are similar to those of delivering *Q*, with *Q* being substituted by the *ACK*.

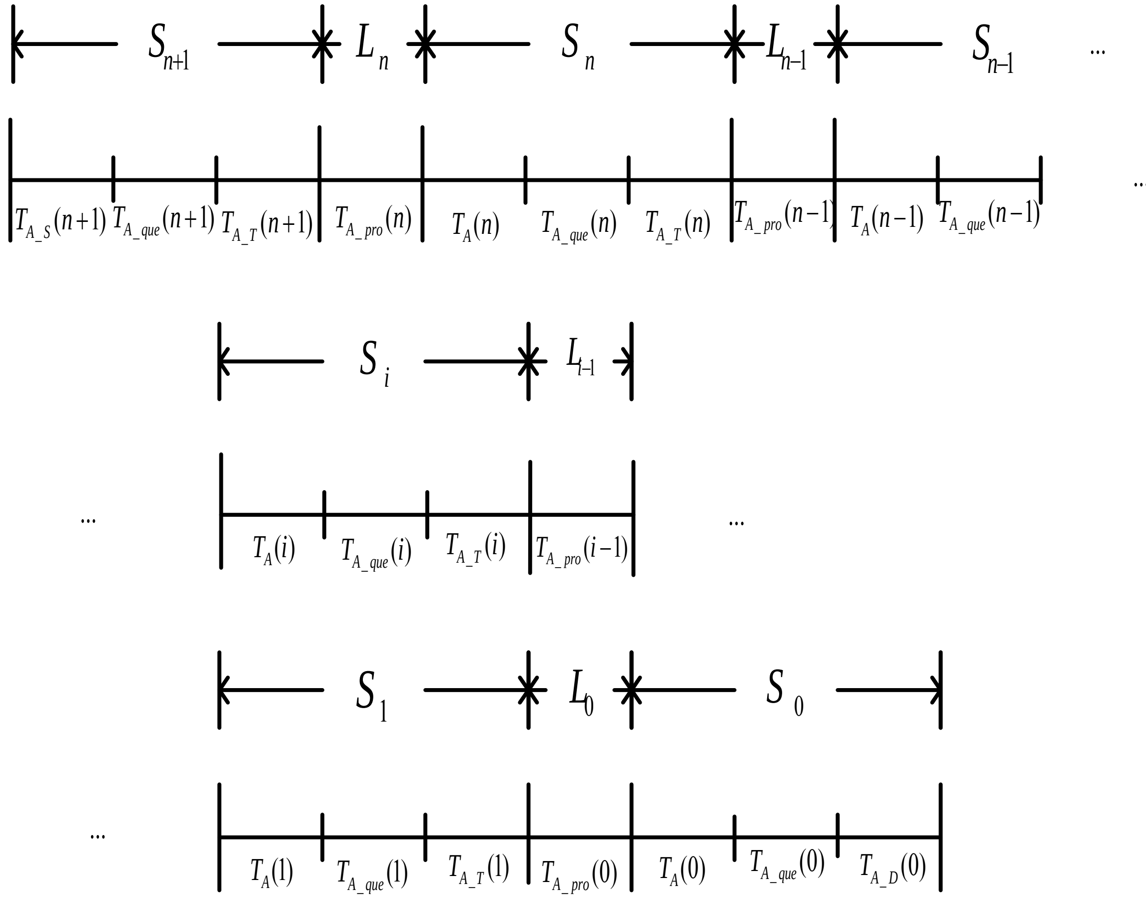

Fig. 3-3 The timings in delivering an *ACK*

Table 3-2. Definitions of concerned terms on *ACK*

Items	Description
$T_{A\_S}(n+1)$	The time required by S_{n+1} to generate an *ACK*
$T_{A\_que}(i)$ i=0,1,2,…n+1	The time the *ACK* waiting in S_i's message queue
$T_{A\_T}(i)$	The time required by S_i to send the *ACK* out from

i=1,2,…n+1	the first bit to the last bit (S_0 is the destination)
$T_{A\_pro}(i)$ i=0,1,2,…n	The time required by the *ACK* to travel from S_i to S_{i-1}. It is defined as the propagation time from when a bit is sent out by S_i to the time point when the bit arrives at S_{i-1}
$T_A(i)$ i=0,1,2,…n+1	The time required by S_i to receive the *ACK*. It is defined as the receiving time period from when the *ACK's* first bit arrives at S_i to the time point when the *ACK's* last bit arrives at S_{i-1}
$T_{A\_D}(0)$	The time required by S_0 to process the *ACK*

3.2.3 Processing delay

Processing delay is the time required to prepare and receive a packet, and encrypt and decrypt SCTP chunks. The purpose of these activities is basically getting the data ready for the next activity, e.g., to be transmitted. Performance of the activities is mainly influenced by hardware processing speed and time complexities of encryption and decryption algorithms. The concerned items include size of encrypted/decrypted data, a node's data generating speed, encryption speed, decryption speed, receiving speed and processing speed. The latter five (i.e., speeds) are described in Table 3-3. The time required to generate a data packet varies dramatically. For example, a control system on receiving a user command may consume a very long time to perform a complicated time-consuming computation. A sensor of a wireless sensor network may on the contrary spend only a few microseconds to transform environmental changes to formatted data.

Table 3-3. The speeds involved in packet processing delays

Items	Description
data generating speed	The speed with which a node generates a bit. After the generation of a packet, the packet is ready to be transmitted or encrypted
encryption speed	The speed with which a node encrypts a bit
decryption speed	The speed with which a node decrypts a bit
receiving speed	The speed with which a node receives a bit
processing speed	The speed with which a node processes a bit

Basically, the data generating speed and processing speed of a computer strongly depend on its cpu performance. Ohlendorf et al. [21] presented that cpu processing speed can be expressed by $\frac{cpu\_count \times cpu\_clock}{clock\_per\_instruction} MIPS$ where *cpu_count* is number of cpus that a node has, *cpu_clock* is the cpu clock rate and *clock_per_instruction* represents the number of clocks required to finish the execution of an instruction. As an example, Kim et al. [22] presented the fact that when a TCP connection is established and a packet of the maximum-sized (1460 byte) is sent with 100 Mb/s (or 900 Mb/s), the required TCP layer's instruction count is 1286 (1356), which is also the number of instructions required to generate a TCP packet. So, we can infer that the packet generating speed at S_0 , denoted by *gen_speed*(0), is *gen_speed*(0)= $\frac{million\_instructions\_per\_second}{instruction\_count\_per\_packet}$ (1)

To express packet encryption/decryption speed, we use the Advanced Encryption Standard (AES) [23], a symmetric encryption mechanism, as an example, and assume that the encryption speed is equal to decryption speed. The time required to encrypt a bit can be derived from the penalty of the AES data encryption [24] through the Regression Analysis [25]. When the lengths of encryption keys are 128, 192, and 256 bits, we can respectively obtain three linear equations, y=18.929x+500, y=22.5x+214.29 and y=26.25x+535.71, which can be generally expressed by $y=\alpha x+\beta$ where x is length of the encrypted data in kilobytes, y in microseconds is the time required to encrypt the data, and α and β are constants once the encryption key length is given.

(1). cost for processing a data packet

S_0 only generates and encrypts Q. Hence, we can derive the formula for $T_{Q\_S}(0)$,

$$T_{Q\_S}(0) = \frac{size\ (Q)/max\_Packet\_size}{gen\_speed\ (0)} + (\alpha \cdot \frac{size(encrypted\ data\ x)}{10^3} + \beta) \cdot 10^{-6} \cdot z \quad (2)$$

where Q is the packet generated before encryption, x is the portion of Q that is encrypted, $(\alpha \cdot \frac{size(encrypted\ data\ x)}{10^3} + \beta) \cdot 10^{-6} \cdot z$ is the encryption cost of AES and z is decision variable. Let Q' be the encrypted Q. Then, $|Q'| = |Q|$ + encryption overheads. If S_0 does not encrypt Q, then $z = 0$, i.e., the encryption cost = 0 and $Q' = Q$.

A node's receiving speed is basically depending on the network interface's current input data rate (receiving rate) and input drop rate. Generally, popular network interfaces are 10 Mbps, 100Mbps and 1Gbps. Since $S_1, S_2, ..., S_n$ do not encrypt and decrypt *Q'*, based on the definition of $T_Q(i)$, we can derive formula

$$T_Q(i) = \frac{size(Q')}{rec\_speed(i) - drop\_rate_{in}(i)},\ i=1,2,\ldots n+1 \tag{3}$$

where $rec\_speed(i)$ is S_i's receiving speed, and $drop\_rate_{in}(i)$ is S_i's arriving data's drop rate, rather than packet drop rate. Formula (2) is also applicable to S_{n+1} for the receipt of *Q'*.

Let

$$R_i = rec\_speed(i) - drop\_rate_{in}(i) \tag{4}$$

which is the actual receiving speed of S_i. Let T_{Q_in} be accumulated processing time consumed by the *n* intermediate nodes to deliver *Q'*.

$$T_{Q\_in} = \sum_{j=1}^{n} T_Q(j) \tag{5}$$

For the destination node S_{n+1}, the time required to recover *Q* (i.e., $T_{Q\_D}(n+1)$) is

$$T_{Q\_D}(n+1) = \frac{size(Q')}{R_{n+1}} \tag{6}$$

Let $T'_{Q\_D}(n+1) = T_{Q\_D}(n+1) + (\alpha \cdot \frac{x}{10^3} + \beta) \cdot 10^{-6} \cdot z = \frac{size(Q')}{R_{n+1}} + (\alpha \cdot \frac{x}{10^3} + \beta) \cdot 10^{-6} \cdot z$ (7)

Let $T_{Q\_processing}$ be the total time for processing *Q'* in the *n*+2 nodes, $S_0, S_1, ..., S_{n+1}$,

$$T_{Q\_processing} = T_{Q\_S}(0) + T_{Q\_in} + T'_{Q\_D}(n+1) \tag{8}$$

(2). Cost for processing an ACK packet

Not that an *ACK* is often not encrypted. Based on the definition stated above, we can derive the formula for S_i's processing cost which only includes *ACK's* generation cost.

$$T_{A\_S}(n+1) = (\alpha \cdot \frac{size(ACK)}{10^3} + \beta) \cdot 10^{-6} \tag{9}$$

$T_A(i)$ can be also derived as

$$T_A(i) = \frac{size\ (ACK)}{R_i} \tag{10}$$

The accumulated processing time consumed by the n intermediate nodes,

$$T_{A\_in} = \sum_{j=1}^{n} T_A(j) \tag{11}$$

In S_0, the cost for processing the *ACK*, then $T_{A\_D}(0) = 0$

Let $T'_{A\_D}(0)$ be the time required by S_0 to receive and process the *ACK*,

$$T'_{A\_D}(0) = T_A(0) + T_{A\_D}(0) = \frac{size\ (ACK)}{R_0} \tag{12}$$

Let $T_{A\_processing}$ be total cost for processing an *ACK*,

$$T_{A\_processing} = T_{A\_S}(n+1) + T_{A\_in} + T'_{A\_D}(0) \tag{13}$$

Let

$$T_{processing} = T_{Q\_processing} + T_{A\_processing} \tag{14}$$

3.2.4 Transmission delay

Transmission delay is the time period from when the first bit of *Q'* is sent out to the time point when the last bit of *Q'* is transmitted. The items concerning transmission delay include the size of *Q'* and actual delivery speed, instead of data rate.

S_i's transmission delay, $T_{Q\_T}(i) = \frac{size\ (Q')}{M_i}$, i=0,1,2,…n (15)

where M_i is S_i's actual delivery speed which is defined as

$$M_i = data\_rate(i) - drop\_rate_{out}(i) \qquad (16)$$

Here, $drop\_rate_{out}(i)$ is drop rate of S_i's departing data, instead of departing packets.

Let $T^{'}_{Q\_T}$ be the transmission delay of the data packet *Q'*,

$$T^{'}_{Q\_T} = \sum_{i=0}^{n} T_{Q\_T}(i) = \sum_{i=0}^{n} \frac{size\ (Q')}{M_i} \qquad (17)$$

Let T_{Q_T} be the transmission delay caused by the n intermediate nodes to deliver *Q'*

$$T_{Q\_T} = \sum_{i=1}^{n} T_{Q\_T}(i) = \sum_{i=1}^{n} \frac{size\ (Q')}{M_i} \qquad (18)$$

Let $T^{'}_{A\_T}$ be the transmission delay for delivery the corresponding *ACK*.

$$T^{'}_{A\_T} = \sum_{i=1}^{n+1} \frac{size\ (ACK)}{M_i} \qquad (19)$$

Let T_{a_T} be the transmission delay caused by the n intermediate nodes to deliver the *ACK*.

$$T_{A\_T} = \sum_{i=1}^{n} \frac{size\ (ACK)}{M_i} \qquad (20)$$

3.2.5 Propagation delay

Propagation delay of a link L_i connecting S_i and S_{i+1} (*i*=0,1,2,…*n*) is the time period from the time point when a bit of *Q'* is sent out by S_i to the time point when the bit arrives at S_{i+1}(i.e., the time required by the bit to travel from S_i to S_{i+1}). The items included are the initial bandwidth, occupied bandwidth and S_i's output drop rate $drop\_rate_{out}(i)$.

$$T_{Q\_pro}(i) = \frac{1}{bandwidth(i) - bandwidth_{occupied}(i) - drop\_rate_{out}(i)} = \frac{1}{M_i} \qquad (21)$$

where $bandwidth(i)$ and $bandwidth_{occupied}(i)$ respectively represent initial bandwidth and occupied bandwidth of the link L_i, i=0,1,2,…n and

$$data\_rate(i) = bandwidth(i) - bandwidth_{occupied}(i) \tag{22}$$

Based on the definition above, when a packet is transmitted, the total propagation delay T'_{Q_pro},

$$T'_{Q\_pro} = \sum_{i=0}^{n} T_{Q\_pro}(i) = \sum_{i=0}^{n} \frac{1}{M_i} \tag{23}$$

Let T_{Q_pro} be the propagation delay caused by the n intermediate nodes to deliver *Q'*.

$$T_{Q\_pro} = \sum_{i=1}^{n} T_{Q\_pro}(i) = \sum_{i=1}^{n} \frac{1}{M_i} \tag{24}$$

For an *ACK* packet, we assume the propagation delay $T_{A\_pro} = T_{Q\_pro}$ (i.e., input bandwidth and output bandwidth are the same. This is reasonable in real situation since the network interface is the same one) to simplify the scope of the following analyses.

3.2.6 Queuing delay

The items concerning queuing delay include packet arrival and departure rates. Here, we assume the processing mechanism pertaining to a queue is the M/M/1 queuing model [26]. One may figure out that the SCTP's multi-streaming mechanism can be viewed as a multi-server system. It is true. But, from the physical layer viewpoint, the interface is the only mechanism (i.e., server of the queue) that sends out packets, no matter which streams the packets belong to. The arrival rate λ is a function of several independent variables, including the packet size, bandwidth, occupied bandwidth, packet drop rate, length of queue, etc. The departure rate μ (i.e., service rate) is also a function of several independent variables, including hardware processing speed, data rates, etc. Due to involving too many influential factors, it is hard to derive the mathematical models for them. But, the arrival (departure) rate of a link can be observed on the receiver (sender) side of a path segment.

Based on queuing theory, $T_{Q\_que}(i) = \dfrac{\lambda_i}{\mu_i(\mu_i - \lambda_i)}$. Let $T^{'}_{Q\_que}$ be the total queuing delay of the concerned path, $T^{'}_{Q\_que} = \sum_{i=0}^{n+1}(\dfrac{\lambda_i}{\mu_i(\mu_i - \lambda_i)})$ [26], which does not contain service time (i.e., transmission time) and is derived under the assumption that no packets are dropped upon arriving and departing. If we consider actual arrival and departure rates, and assume that they follow Poisson distribution, then

$$T_{Q\_que}(i) = \frac{R^{'}_i}{M^{'}_i(M^{'}_i - R^{'}_i)} \tag{25}$$

$$T^{'}_{Q\_que} = \sum_{i=0}^{n+1} T_{Q\_que}(i) = \sum_{i=0}^{n+1}(\frac{R^{'}_i}{M^{'}_i(M^{'}_i - R^{'}_i)}) \tag{26}$$

where $R^{'}_i$ and $M^{'}_i$ are respectively S_i's actual packet arrival and departure rates, and

$$R^{'}_i = \frac{R_i}{size(Q')} \tag{27}$$

$$M^{'}_i = \frac{M_i}{size(Q')} \tag{28}$$

Let T_{Q_que} be the queuing delay generated by the n intermediate nodes to process *Q'*

$$T_{Q\_que} = \sum_{i=1}^{n} T_{Q\_que}(i) = \sum_{i=1}^{n}(\frac{R^{'}_i}{M^{'}_i(M^{'}_i - R^{'}_i)}) \tag{29}$$

For an *ACK* packet, we also assume the queuing delay $T_{A\_que} = T_{Q\_que}$ to simplify the scope of the following analyses.

3.3. Total Cost without Retransmission

Assume the initial primary path fails. There are two methods to choose a new primary path. One is that, we evaluate the remaining *H*-1 paths and sort the paths based on their evaluation results. The one with the highest performance is then selected as the new primary path. Another method is comparing arbitrary two paths, e.g., paths *q* and *r*. The one with higher performance, e.g., *q*, will be chosen. We then select an uncompared path as the new *r*, and compare *q* and *r* again. This procedure repeats until all paths are compared. Then, the one with the highest performance can be selected. With either method, *H*-1 times of comparison are required. But, using the second approach, we can omit the evaluation of many items, e.g., $S_0's$ and $S_{n+1}'s$ costs since the source nodes and destination nodes of two paths, e.g., paths q and r, belonging to the same association are themselves the same, and the two paths deliver the same packet *Q*/*Q'* and *ACK*. The cost difference CD_{qr} between the two paths only results from involving different numbers of intermediate nodes and different intermediate nodes.

Let *TC* be the total cost of packet delivery without retransmission,

$$TC = T_{processing} + (T_{Q\_T} + T_{A\_T}) + (T_{Q\_pro} + T_{A\_pro}) + (T_{Q\_que} + T_{A\_que}) \tag{30}$$

According to equations (5), (11), (18), (20), (24) and (29).

$$CD_{qr} = TC_q - TC_r = (T^q_{Q\_in} - T^r_{Q\_in} + T^q_{A\_in} - T^r_{A\_in})$$

$$+(T^q_{Q\_T} - T^r_{Q\_T} + T^q_{A\_T} - T^r_{A\_T}) + 2(T^q_{Q\_pro} - T^r_{Q\_pro}) + 2(T^q_{Q\_que} - T^r_{Q\_que}) \tag{31}$$

From equations (3), (4), (5), (10) and (11), we can see that the expression $(T^q_{Q\_in} - T^r_{Q\_in} + T^q_{A\_in} - T^r_{A\_in})$ is a function of R_i, *i*=1,2,… $n_q + n_r$ once *size(Q')* is given where n_q and n_r are respectively numbers of path *q's* and path *r's* immediate nodes. Similarly, based on equations (18) and (20), the expression $(T^{q}_{Q\_T} - T^{r}_{Q\_T} + T^{q}_{A\_T} - T^{r}_{A\_T})$ is a function of M_i, i=0,1,2,… $n_q + n_r$. Based on equations (24) and (29), the remaining two expressions are respectively functions of M_i, $R_i^{'}$ and $M_i^{'}$. Let $CD_{qr} = | CD_{qr}^{'} |$, then

$$
\begin{aligned}
CD_{qr}^{'} = &\sum_{j=1}^{n_q} \frac{size\ (Q')}{R_{j,q}} - \sum_{k=1}^{n_r} \frac{size\ (Q')}{R_{k,r}} + \sum_{j=1}^{n_q} \frac{size\ (ACK)}{R_{j,q}} - \sum_{k=1}^{n_r} \frac{size\ (ACK)}{R_{k,r}} + \\
&\sum_{j=1}^{n_q} \frac{size\ (Q')}{M_{j,q}} - \sum_{k=1}^{n_r} \frac{size\ (Q')}{M_{k,r}} + \sum_{j=1}^{n_q} \frac{size\ (ACK)}{M_{j,q}} - \sum_{k=1}^{n_r} \frac{size\ (ACK)}{M_{k,r}} + \\
&2(\sum_{j=1}^{n_q-1} \frac{1}{M_{j,q}} - \sum_{k=1}^{n_r-1} \frac{1}{M_{k,r}}) + 2(\sum_{j=1}^{n_q} \frac{R'_{j,q}}{M'_{j,q}(M'_{j,q} - R'_{j,q})} - \sum_{k=1}^{n_r} \frac{R'_{k,r}}{M'_{k,r}(M'_{k,r} - R'_{k,r})})
\end{aligned}
\tag{32}
$$

where $R_{j,q}$ ($R_{k,r}$) is actual receiving speed of $S_{j,q}$ (i.e., node *j* on the path *q*) (of $S_{k,r}$ (i.e., node *k* on path *r*)), $M_{j,q}$ ($M_{k,r}$) is actual delivery speed of $S_{j,q}$ (of $S_{k,r}$), and $R'_{j,q}$ and $M'_{j,q}$ ($R'_{k,r}$ and $M'_{k,r}$) are respectively actual packet arrival rate and departure rate of $S_{j,q}$ (of $S_{k,r}$).

Since an *ACK* is a packet of fixed length, given an encrypted packet *Q'* and an association that has two paths, *q* and *r* of lengths n_q+2 and n_r+2, respectively, from equations (4), (16), (27), and (28), we can see that only (*rec_speed(j,q)*, $drop\_rate_{in}(j,q)$, *data_rate(j,q)*, $drop\_rate_{out}(j,q)$) and (*rec_speed(k,r)*, $drop\_rate_{in}(k,r)$, *data_rate(k,r)*, $drop\_rate_{out}(k,r)$) are unknown, *j*=1,2,3… n_q, *k*=1,2,3… n_r. On the other hand, if we can access the $n_q + n_r$ intermediate nodes' network management information through a network management protocol (e.g., Simple Network Management Protocol (SNMP)), then we can retrieve the quadruples (*rec_speed()*, $drop\_rate_{in}()$, *data_rate()*, $drop\_rate_{out}()$) from all immediate nodes. So, we further assume that all immediate nodes' management information bases (MIBs) are available, and can be accessed. However, accessing network management information takes time. It is hard to retrieve the information of concern for each path in a real time manner right before choosing a primary path. And, before current accurate information is gathered, we cannot make a right decision and choose the right path. On the other hand, if we access the information before choosing the best path, delivery of *Q'/Q* will be delayed. To solve this problem, we predict the quadruple values for each node by using the exponential average algorithm [27], $\tau_{n+1} = \alpha\tau_n + (1-\alpha)T_n$, where τ_{n+1} and τ_n are respectively the $(n+1)^{th}$ and n^{th} predicted values of one of the quadruple elements, and T_n is the n^{th} actual value of the feature retrieved from the corresponding MIB.

Here,

$$\overline{R_{j,q}}^{n+1} = \alpha_{L_{j,q}} \cdot \overline{R_{j,q}}^{n} + (1-\alpha_{R_{j,q}}) \cdot R_{j,q}{}^{n} \tag{33}$$

$$\overline{M_{j,q}}^{n+1} = \alpha_{M_{j,q}} \cdot \overline{M_{j,q}}^{n} + (1-\alpha_{M_{j,q}}) \cdot M_{j,q}{}^{n} \tag{34}$$

$$\overline{drop\_rate_{in}}^{n+1}(j,q) = \alpha_{dp_{in}(j,q)} \cdot \overline{drop\_rate_{in}}^{n}(j,q) + (1-\alpha_{dp_{in}(j,q)}) \cdot drop\_rate_{in}{}^{n}(j,q) \tag{35}$$

$$\overline{drop\_rate_{out}}^{n+1}(j,q) = \alpha_{dp_{out}(j,q)} \cdot \overline{drop\_rate_{out}}^{n}(j,q) + (1-\alpha_{dp_{out}(j,q)}) \cdot drop\_rate_{out}{}^{n}(j,q) \tag{36}$$

where the $\overline{R_{j,q}}^{n+1}$, $\overline{M_{j,q}}^{n+1}$, $\overline{drop\_rate_{in}}^{n+1}(j,q)$ and $\overline{drop\_rate_{out}}^{n+1}(j,q)$ ($\overline{R_{j,q}}^{n}$, $\overline{M_{j,q}}^{n}$, $\overline{drop\_rate_{in}}^{n}(j,q)$ and $\overline{drop\_rate_{out}}^{n}(j,q)$) are respectively the $(n+1)^{th}$ (the n^{th}) predicted receiving speed, delivery speed, input drop rate and output drop rate of $S_{j,q}$, $\alpha_{R_{j,q}}$, $\alpha_{M_{j,q}}$, $\alpha_{dp_{in}(j,q)}$ and $\alpha_{dp_{out}(j,q)}$ are respectively weights of $S_{j,q}$'s receiving speed, delivery speed, input drop rate and output drop rate, and $R_{j,q}{}^{n}$, $M_{j,q}{}^{n}$, $drop\_rate_{in}{}^{n}(j,q)$ and $drop\_rate_{out}{}^{n}(j,q)$ are respectively the n^{th} actual receiving speed, actual delivery speed, actual input drop rate and actual output drop rate. Since SCTP's path change and switch do not occur frequently, we often have enough time to access the actual values of the four terms from each intermediate node's MIB [28]. For example, the Cisco provides the CISCO-IETF-SCTP-EXT-MIB [29] to supply the MIB module.

In an MIB, the items ipInReceives(t) (OID= {1.3.6.1.2.1.4.3}), ipInDiscards(t) (OID= {1.3.6.1.2.1.4.8}) , ipOutRequests(t) (OID= {1.3.6.1.2.1.4.10}) and ipOutDiscards(t) (OID= {1.3.6.1.2.1.4.11}) are respectively defined as accumulated numbers of packets that the underlying router has so far received, dropped on the input side, sent and dropped on the output side since the router started up. By retrieving the four items from intermediate node $S_{j,q}$, $\overline{R_{j,q}}^{n+1}$, $\overline{M_{j,q}}^{n+1}$, $\overline{drop\_rate_{in}}^{n+1}(j,q)$ and $\overline{drop\_rate_{out}}^{n+1}(j,q)$, at time point t_{n+1} can be derived where the $R_{j,q}{}^{n}$, $M_{j,q}{}^{n}$, $drop\_rate_{in}{}^{n}(j,q)$ and $drop\_rate_{out}{}^{n}(j,q)$ respectively in equations (33) ~ (36) can be obtained by accessing the MIB twice at t_{S2} and t_{S1} right after the previous (i.e., the n^{th}) switchover at time t_S by the following equations,

$$R_{j,q}{}^{n} = \frac{(ipInReceives(t_{S2}) - ipInDiscards(t_{S2})) - (ipInReceives(t_{S1}) - ipInDiscards(t_{S1}))}{t_{S2} - t_{S1}},$$

$$M_{j,q}{}^{n} = \frac{(ipOutRequests(t_{S2}) - ipOutDiscards(t_{S2})) - (ipOutRequests(t_{S1}) - ipOutDiscards(t_{S1}))}{t_{S2} - t_{S1}},$$

$$drop\_rate_{in}{}^{n}(j,q) = \frac{ipInDiscards(t_{S2}) - ipInDiscards(t_{S1})}{t_{S2} - t_{S1}},$$

$$drop\_rate_{out}{}^{n}(j,q) = \frac{ipOutDiscards(t_{S2}) - ipOutDiscards(t_{S1})}{t_{S2} - t_{S1}}$$

in which $t_{S+1} >> t_{S2} > t_{S1} > t_S$, n=0,1,2…. Here, n=0 represents the time point right after the SCTP started up (i.e., the time point when the initial primary path has just been selected).

3.4. Total Cost with Retransmission

In the SCTP, as stated above, when a packet is sent out and the sender cannot receive the corresponding *ACK* within *RTO* seconds, the packet will be retransmitted, and the $RTO \leftarrow 2 \times RTO$ (i.e., the SCTP duplicates its *RTO* value)**.** Each time when a packet cannot be successfully delivered within *PMR* times of retransmission (i.e. *PMR*+1 transmission), the SCTP will evaluate remaining alternate paths and choose a new primary path [30]. Each time when a transmission fails, and $RTO.MIN \leq RTO \leq RTO.MAX$ **,** as shown in Fig. 3-4, the relationship among *RTO*, *PMR*, and the opportunity to choose a new primary is shown in Fig. 3-5. When *k* times of transmission (rather than retransmission) fail, the accumulated costs due to timeout are $\sum_{i=1}^{k} 2^{i-1} \cdot RTO \ \ (= (2^0 + 2^1 + ... + 2^{k-1}) \cdot RTO)$. Now, we assume the source node S_0 currently experiences *k*-1 retransmission failures (i.e., *k* transmission failures, including the initial transmission failure) and the k^{th} retransmission (i.e., $(k+1)^{th}$ transmission) succeeds.

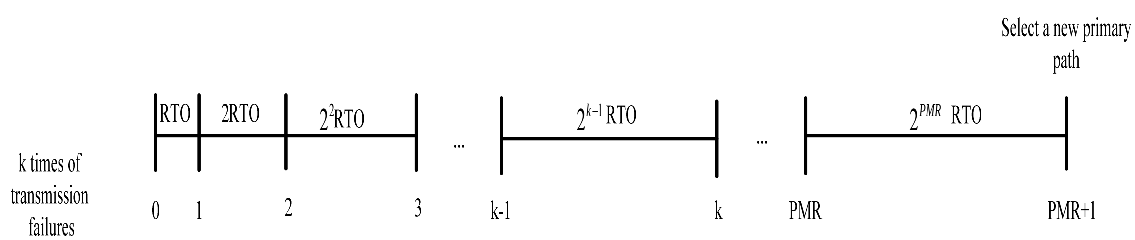

Fig. 3-4 Timings of *k* transmission failures

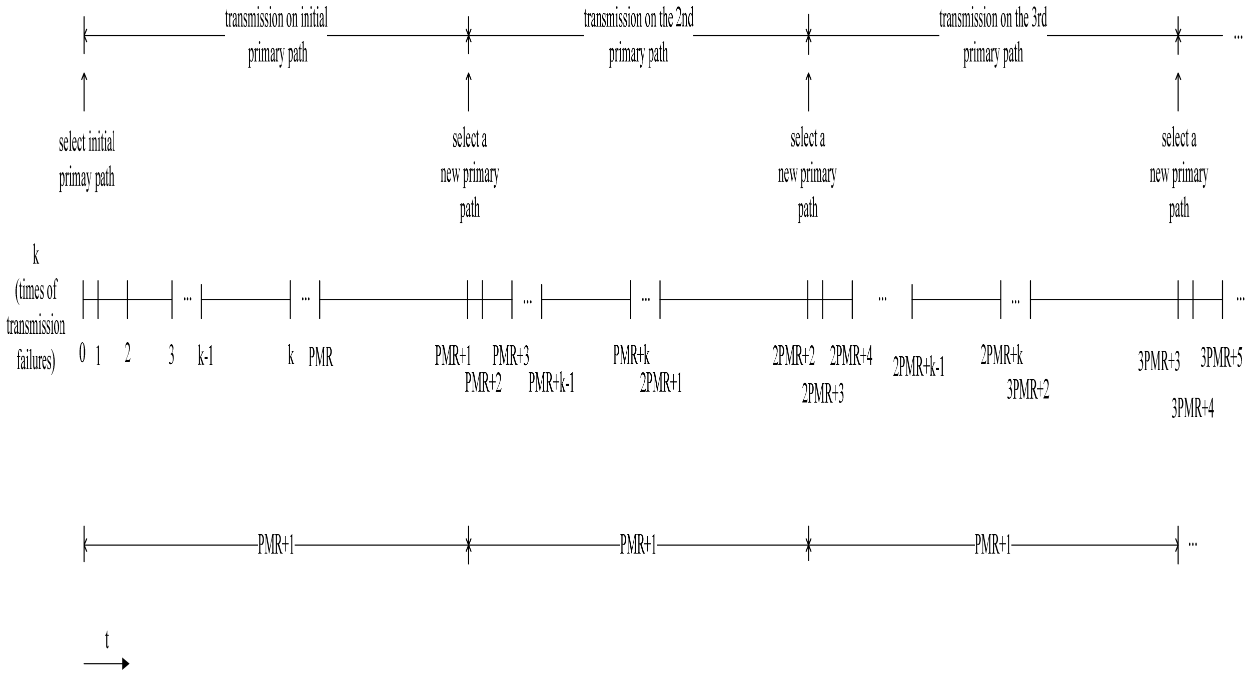

Fig. 3-5 Timings of transmission/retransmission and primary path selection

(1). when $0 \le k < PMR + 1$

$k < PMR + 1$ implies the $(k+1)^{th}$ transmission also goes through the initial primary path where the "1" represents the initial transmission. The total cost for successfully delivering *Q*/ *Q'* at $(k+1)^{th}$ transmission, denoted by T_0, is

$$\begin{aligned} T_0 &= T_{evaluation(0)} + (T_{Q\_S}(0) + \sum_{i=1}^{k} 2^{i-1} \cdot RTO + (T_{Q\_processing} - T_{Q\_S}(0) \\ &\quad + T_{Q\_T} + T_{Q\_pro} + T_{Q\_que})) + (T_{A\_processing} + T_{A\_T} + T_{A\_pro} + T_{A\_que}) \\ &= T_{evaluation(0)} + \sum_{i=1}^{k} 2^{i-1} \cdot RTO + T_{processing} + T_{Q\_T} + T_{A\_T} + 2(T_{Q\_pro} + T_{Q\_que}) \end{aligned} \quad (37)$$

$T_{evaluation(i)}$ represents the cost of evaluating all remaining *H-i* paths of the underlying association. When *i*=0, $T_{evaluation(0)} = f(H)$. The first $T_{Q\_S}(0)$ is the time that the S_0 requires to initially prepare *Q'*. After *Q'* is sent out, the *RTO* timer is then initiated. Usually, S_0 keeps *Q'* in its message buffer until it receives the corresponding *ACK*. So, when *Q'* due to some reason has to be retransmitted, S_0 just retrieves *Q'* from the buffer without regenerating *Q'* again. That is why equation (37) substrates an item $T_{Q\_S}(0)$ from $T_{Q\_processing}$.

Let $T_{delivery}$ be the time required by $(k+1)^{th}$ transmission which successfully delivers *Q'* without any retransmission, i.e., $T_{delivery} = TC$ in equation (30).

$$T_{delivery} = T_{processing} + T_{Q\_T} + T_{A\_T} + 2(T_{Q\_pro} + T_{Q\_que}) \tag{38}$$

Then, T_0 in (37) can be expressed by

$$T_0 = T_{evaluation(0)} + \sum_{i=1}^{k} 2^{i-1} RTO + T_{delivery} \tag{39}$$

Assume the packet loss rate of the underlying primary path (i.e., the initial primary path) is P_0, which is also *Q'* retransmission probability. If the i^{th} delivery failure is denoted by *DF*(*i*), based on the Bayes' Theorem [31], $P(A \mid B) = \frac{P(A \cap B)}{P(B)}$, the average delivery cost T_{av} is

$$
\begin{aligned}
T_{av} &= T_{evaluation(0)} + RTO \cdot P_0 + 2 \cdot RTO \frac{P(DF(2) \cap DF(1))}{P(DF(1))} + 2^2 \cdot RTO \frac{P(DF(3) \cap (\bigcap_{i=1}^{2} DF(i)))}{P(DF(2) \cap DF(1))} \\
&+ \ldots + 2^{m-1} \cdot RTO \frac{P(DF(m) \cap (\bigcap_{i=1}^{m-1} DF(i)))}{P(\bigcap_{i=1}^{m-1} DF(i))} + \ldots + 2^{k-1} \cdot RTO \frac{P(DF(k) \cap (\bigcap_{i=1}^{k-1} DF(i)))}{P(\bigcap_{i=1}^{k-1} DF(i))} \\
&+ T_{delivery}(1-P_0) \\
&= T_{evaluation(0)} + RTO \cdot P_0 + 2 \cdot RTO \frac{P_0^2}{P_0} + 2^2 \cdot RTO \frac{P_0^3}{P_0^2} + \ldots + 2^{m-1} \cdot RTO \frac{P_0^m}{P_0^{m-1}} + \ldots \\
&+ 2^{k-1} \cdot RTO \frac{P_0^k}{P_0^{k-1}} + T_{delivery}(1-P_0) \\
&= T_{evaluation(0)} + \sum_{i=1}^{k} 2^{i-1} \cdot RTO \cdot P_0 + T_{delivery}(1-P_0)
\end{aligned} \tag{40}
$$

where $RTO \cdot P_0$ represents the time on the 1^{st} timeout, and

$2^{m-1} \cdot RTO \frac{P(DF(m) \cap (\bigcap_{i=1}^{m-1} DF(i)))}{P(\bigcap_{i=1}^{m-1} DF(i))}$ is the time required on the m^{th} timeout, $1 \le m \le k$.

(2). when *k*≥*PMR*+1

k≥*PMR*+1 implies that the SCTP has selected the best alternate path as the new primary path *r* times, *r*=1,2… $\left\lfloor \frac{k+1}{PMR+1} \right\rfloor$, …, or $H-1$, where *H* is the total number of paths that the underlying association has, and $\left\lfloor \frac{k+1}{PMR+1} \right\rfloor$ means the underlying primary path is the $\left\lfloor \frac{k+1}{PMR+1} \right\rfloor^{th}$ newly selected primary path (excluding the initial selection). The maximum value of r is $H-1$ instead of H also due to excluding the initial. Total cost from when the SCTP starts transmission after an association is established to the time point the $(k+1)^{th}$ transmission succeeds is

$$\begin{aligned}
T &= (T_{evaluation(0)} + \sum_{i=1}^{PMR+1} 2^{i-1} \cdot RTO) + (T_{evaluation(1)} + \sum_{i=1}^{PMR+1} 2^{i-1} \cdot RTO) + \ldots \\
&\quad + (T_{evaluation(\left\lfloor \frac{k+1}{PMR+1} \right\rfloor - 1)} + \sum_{i=1}^{PMR+1} 2^{i-1} \cdot RTO) + (T_{evaluation(\left\lfloor \frac{k+1}{PMR+1} \right\rfloor)} + \sum_{i=1}^{S} 2^{i-1} \cdot RTO \\
&\quad + T_{delivery(\left\lfloor \frac{k+1}{PMR+1} \right\rfloor)}) \\
&= \sum_{i=0}^{\left\lfloor \frac{k+1}{PMR+1} \right\rfloor} T_{evaluation(i)} + \left\lfloor \frac{k+1}{PMR+1} \right\rfloor \sum_{i=1}^{PMR+1} 2^{i-1} \cdot RTO + \sum_{i=1}^{S} 2^{i-1} \cdot RTO + T_{delivery(\left\lfloor \frac{k+1}{PMR+1} \right\rfloor)}
\end{aligned} \tag{41}$$

where $T_{evaluation(i)}$, the cost of the i^{th} evaluation of paths, is proportional to the number of remaining paths, S(=k-$\left\lfloor \frac{k+1}{PMR+1} \right\rfloor$($PMR$+1)) is the times of timeouts (i.e., transmission failures) on the $\left\lfloor \frac{k+1}{PMR+1} \right\rfloor^{th}$ primary path ($\left\lfloor \frac{k+1}{PMR+1} \right\rfloor^{th} = 0$ means the initial primary path) before *Q'* is successfully delivered, and there are a total of $\left\lfloor \frac{k+1}{PMR+1} \right\rfloor + 1$ times of path evaluation (including the initial evaluation). Assume if there are r remaining paths, then r+i=H. Let $T_{evaluation}(i) = f(r) = f((H-i)$, where i=$\left\lfloor \frac{k+1}{PMR+1} \right\rfloor$, $0 \le i \le H$ and $T_{delivery(\left\lfloor \frac{k+1}{PMR+1} \right\rfloor)}$ is the cost required to successfully deliver *Q'* and receive the corresponding *ACK* (refer to equation (38)) through the $\left\lfloor \frac{k+1}{PMR+1} \right\rfloor^{th}$ selected primary path. Let $T_{delivery}$ in equation (39) be $T_{delivery}(0)$, and $\sum_{i=1}^{0} T_{evaluation(i)} = 0$, then we can conclude that equation (41) is the general equation of T.

Assume path failure rate of the i^{th} primary path is P_i which is also retransmission probability of *Q'* on path *i*, *i*=0,1,2…,*H*-1. Let $k_i = k - \left\lfloor \frac{k+1}{PMR+1} \right\rfloor (PMR+1)$ which is the times of timeouts on the i^{th} primary path before *Q'* is successfully delivered on this path, *i*=0,1,2,…,*H*-1.

Let c_i be the cost that the SCTP consumes to successfully deliver Q' on the i^{th} primary path. Let $T_{retrans\_time\_out}$ be $\sum_{i=1}^{PMR+1} 2^{i-1} \cdot RTO$ which is the total cost of PMR+1 times of transmission failures, i.e., the SCTP will choose a new path.

$$c_0 = T_{evaluation(0)} + \sum_{i=1}^{k_0} 2^{i-1} \cdot RTO \cdot P_0 + T_{delivery(0)}(1-P_0)$$

$$c_1 = (T_{evaluation(0)} + T_{retrans\_time\_out}) + T_{evaluation(1)} + \sum_{i=1}^{k_1} 2^{i-1} \cdot RTO \cdot P_1 + T_{delivery(1)}(1-P_1)$$

$$c_2 = \sum_{i=0}^{1} (T_{evaluation(i)} + T_{retrans\_time\_out}) + T_{evaluation(2)} + \sum_{i=1}^{k_2} 2^{i-1} \cdot RTO \cdot P_2 + T_{delivery(2)}(1-P_2)$$

…

$$c_m = \sum_{i=0}^{m-1} (T_{evaluation(i)} + T_{retrans\_time\_out}) + T_{evaluation(m)} + \sum_{i=1}^{k_m} 2^{i-1} \cdot RTO \cdot P_m + T_{delivery(m)}(1-P_m)$$

…

$$c_{H-1} = \sum_{i=0}^{H-2} (T_{evaluation(i)} + T_{retrans\_time\_out}) + T_{evaluation(H-1)} + \sum_{i=1}^{k_{H-1}} 2^{i-1} \cdot RTO \cdot P_{H-1} + T_{delivery(H-1)}(1-P_{H-1})$$

$$
\begin{aligned}
T_{av} &= \frac{1}{H}\sum_{i=0}^{H-1} c_i \\
&= \frac{1}{H}\left(\sum_{j=0}^{H-2}\sum_{i=0}^{j}(T_{evaluation(i)} + T_{retrans\_time\_out}) + \sum_{i=0}^{H-1} T_{evaluation(i)} + \sum_{h=0}^{H-1}\sum_{i=1}^{k_h} 2^{i-1}\cdot RTO \cdot P_h + \sum_{i=0}^{H-1} T_{delivery(i)}(1-P_i)\right)
\end{aligned}
\tag{42}
$$

Chapter 4: Experimental Results

4.1. Simulation Environment Setup

Our simulations were carried out by running a revision of Delaware University's SCTP module [32] for NS-2 [33]. The simulation topology is shown in Fig. 4-1. The two end nodes, sender and receiver, both have 4 IP addresses. Routers 1-1 and 1-2, routers 2-1 and 2-2,…, and routers 4-1 and 4-2 are set up between the two end nodes. Router i-1 is connected to router i-2 and i=1,2,3,4. The bandwidth of path 1 is 2Mbps, and those of paths 2, 3 and 4 are 1.5Mbps, 1.8Mbps and 1Mbps, respectively. The SCTP parameters are all default values except those mentioned above. The sender continuously sends 2 Mbps FTP data to the receiver. Switchover occurs at the 10^{th} sec. Five experiments were performed in this study. The first evaluated the PSASP's end-to-end delays, jitters, throughputs and packet drop rates. The second studied the four QoS parameters of the PSASP given different numbers of routers along a tested routing path. The third redid the first experiment but given different error rates to each tested path. The fourth measured the scales of delays. The fifth evaluated switching costs when H paths are given.

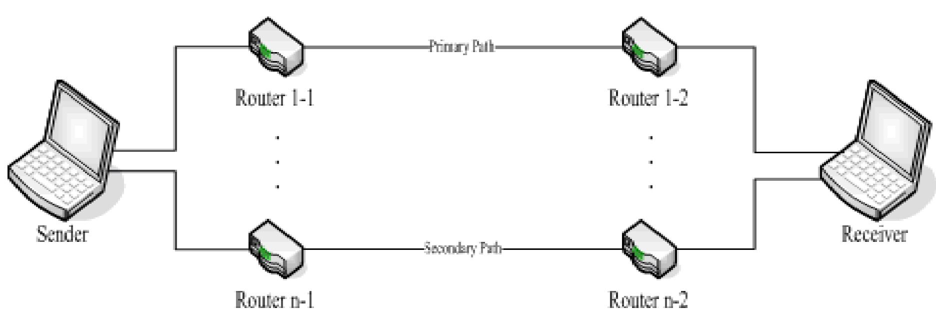

Fig. 4-1 The Simulation topology

Table 4-1. Simulation parameters

Parameters	Value
Sending rate	2 Mbps FTP data
Propagation delay	50 ms
SCTP chunk size	1468 bytes
SCTP MTU	1500 bytes
Path 1's bandwidth	2Mbps
Path 2's bandwidth	1.5Mbps
Path 3's bandwidth	1.8Mbps
Path 4's bandwidth	1Mbps

4.2. Simulation Results of the First Experiment

In the first experiment, three state-of-the-art systems, including the standard SCTP [13], Optimized SCTP [20], and RTT Based SCTP [11], are tested and compared with the PSASP. The default primary path of the standard SCTP is set to path 2.

The experimental results of the four schemes on the end-to-end delays as illustrated in Fig. 4-2 are initially almost the same. But, after the first switchover, the PSASP had less delays than others, and right after the switchover, the end-to-end delays of the four systems between the 11^{th} and 12^{th} seconds do not increase sharply because they all have enough bandwidth to transmit packets. When time passed and more packets and overheads were sent and involved, respectively, the delays increased quickly. But the PSASP had less delay because it selected the best path. The Optimized SCTP as stated above reduced its congestion window size slowly. That is why its end-to-end delays after congestion are longer. The RTT based SCTP can adjust the congestion window based on round trip time, so the delay is lower than the stand SCTP. The PSASP calculates path delays to select the fastest path, but the standard SCTP, Optimized SCTP and RTT based SCTP do not specify how to select alternate paths as the primary paths when necessary. They select according to the order the paths are specified when the underlying association was established. For example, RTT based SCTP measures RTT to adjust its congestion window instead of selecting a path.

In the best case, when the fastest path is the 1st alternate path, the second fast is the 2nd alternate path …, and the slowest one is the last alternate path, then the order of the path selected of the four tested schemes will be the same. This can not discriminate the characteristics of path selection. So, we do not consider the case only. The Optimized SCTP and RTT based SCTP adjust their sizes of congestion windows when necessary. So when congestion is not severe and we do not need to hugely reduce the size of congestion window, in this case the two schemes are better than the PSASP and standard SCTP. But this situation is not always true. We should consider general case which is the case that congestion may or may not be severe, and current default path may or may not be the fastest one. Hence, a way to keep the association performing the best is required. Choosing the best path and adjust congestion are the solutions. The SCTP and RTT based SCTP do not select the best path. The effect of adjusting congestion window due to limited bandwidth of current path is sometimes not significant. That is why the two schemes' delays are longer.

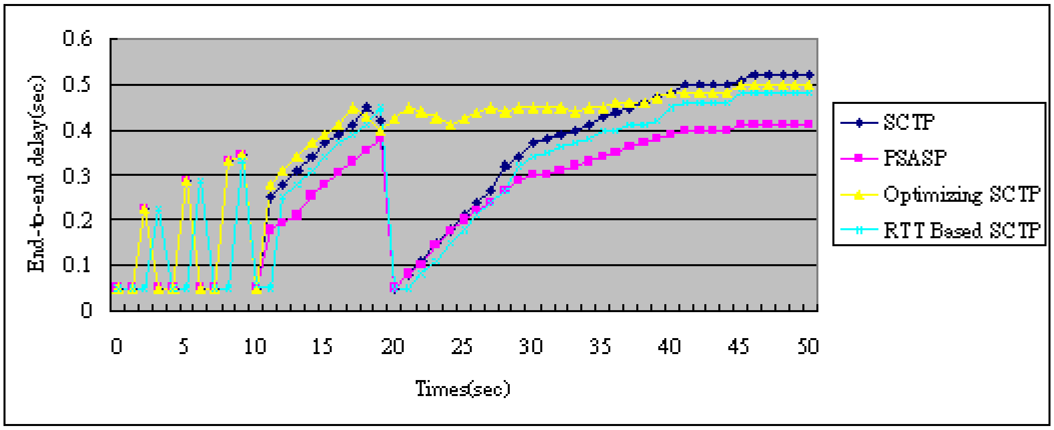

Fig. 4-2 End to end delays of the four tested schemes

The experimental results of jitters as illustrated in Fig. 4-3 show that the PSASP had smaller jitters than others had. At the point when the primary path begins its transmission, the jitters vibrated because the two sides of the path need to exchange information, e.g., four-way handshake, resulting in more transmission overheads. However, the transmission and jitters were soon stable. When switchover occurs, the jitters vibrated again, and the other three schemes' are larger than they were. The PSASP had a similar phenomenon, but the vibration is smoother and smaller since the PSASP always chooses the path currently with the widest bandwidth as the new primary path. The reason is the same as that of the first experiment. Generally, longer transmission delays result in larger jitters, and lower traffic often causes shorter and more smooth vibration. But, the Optimized SCTP's vibration is often still huge because its congestion window decreases slowly when traffic is congested.

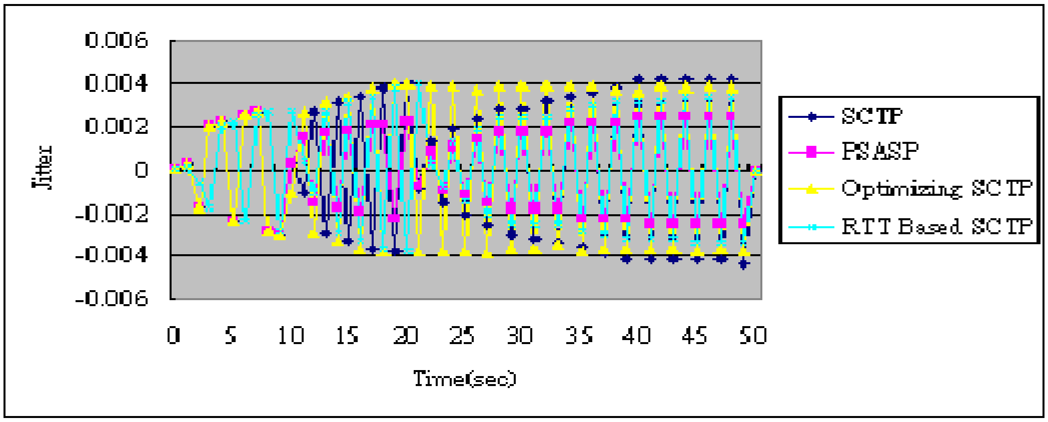

Fig. 4-3 Jitters of the four tested schemes

The experimental results of throughputs are illustrated in Fig. 4-4. Before switchover, throughputs of the four schemes are not significantly different. After the switchover, since the path with the shortest RTT is selected, the PSASP outperforms the others. The reason is the same as that of the first experiment. Although Optimized SCTP and RTT based SCTP adjust their congestion window, but their performance is limited by current path's bandwidth. The PSASP selects the best path which has the widest bandwidth. The wider bandwidth can transmit more packets than smaller bandwidth.

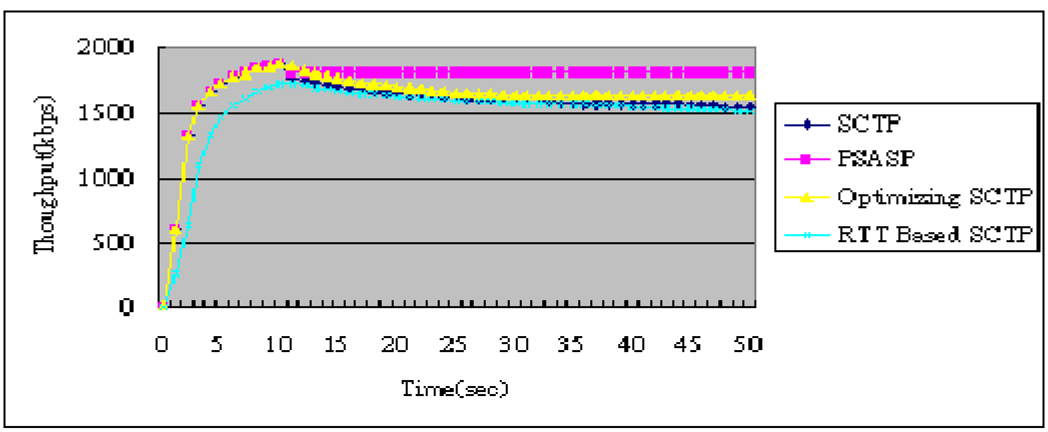

Fig. 4-4 Throughputs of the four tested schemes

The experimental results of packet loss rates are illustrated in Table 4-2. The PSASP exhibits the best also due to choosing the best alternate path (i.e., path 3) which provides a higher transmission quality and stable environment than the default path (i.e., path 2) does. The reason is the same as that of the first experiment. The wider bandwidth can transmit and process many more packets and reduce probability of network congestion. So, it can decrease packet loss rate and probability of packet retransmission.

Table 4-2. Packet loss rates of the four tested schemes

Protocols	No. of packets sent	No. of packets received	No. of packets lost	Packet loss rate(%)
SCTP	22826	22724	102	0.446
PSASP	27077	27010	67	0.247
OPT SCTP	23543	23390	153	0.649
RTT SCTP	22483	22402	81	0.360

Generally, the Optimized SCTP has better throughputs than standard SCTP has. Since when packets got lost, the size of its congestion window shrinks slowly. But this also causes its high packet loss rate. Its delays and jitters are also relatively huge. The RTT based SCTP sacrifices a portion of its throughputs by adjusting congestion window frequently to exploit lower delays, jitters and packet loss rates than standard SCTP. Compared to the three schemes, the PSASP can diminish the delays, maintain smooth jitters, improve performance and decrease the packet loss rates.

4.3. Performance on Different Numbers of Routers

Fig. 4-5 illustrates the experimental results of end-to-end delays of the second experiment. The numbers of routers given are 1, 3, 5, 7 and 9. When the number of routers increases, the end-to-end delays are obviously longer, resulting from longer accumulated transmission, propagation and queuing delays. The congestion occurs at the 15^{th} second, making longer delays for number of routers=1 and 3. But, the influence on the other three numbers of routers is not significant because when a node, e.g., node i, is congested, only the size of its congestion window is reduced. Other intermediate nodes are not temporarily affected. Node i's downstream nodes continue transmitting packets originally queued in their message buffers to their immediate downstream nodes. Node i's immediate upstream node keeps queuing packets in its message buffer. Before downstream nodes' buffers are all empty and upstream node's buffers are all full, node i's congestion window may be large enough again to supply enough packets for downstream nodes to continue their transmission. We call this phenomenon packet-flow regulation. A path with many more routers has a better regulation effect since many more packets are accumulated in the message buffers.

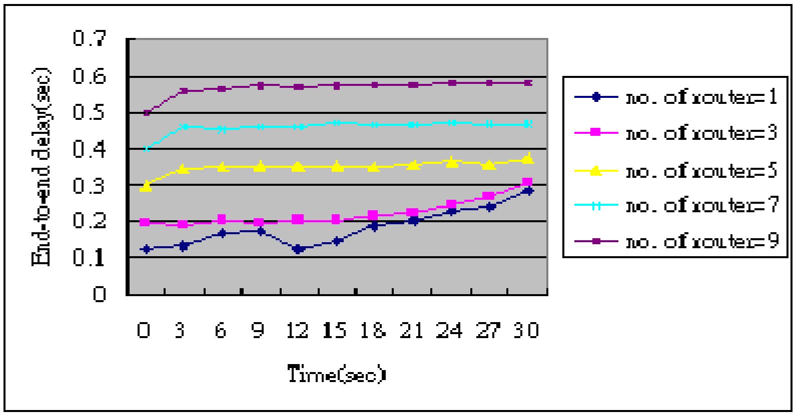

Fig. 4-5 End-to-end delays of the PSASP given different numbers of routers

The experimental results of jitters are illustrated in Fig. 4-6. Jitters are relatively huge on number of routers=1, particularly after the 15^{th} second because of network congestion. But, no. of routers=5, 7 and 9 are not significantly affected, also due to packet-flow regulation. Generally, longer transmission delays result in larger jitters, and lower traffic often causes shorter and smoother vibration. In this case, the number of routers=1 which has less packet-flow regulation effect yields higher jitters.

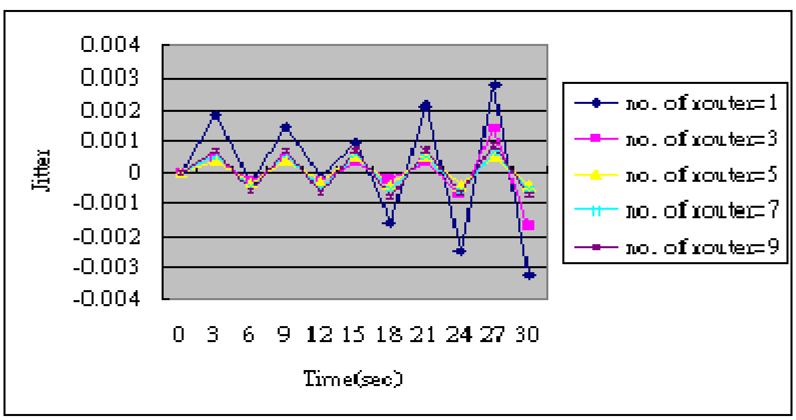

Fig. 4-6 Jitters of the PSASP given different numbers of routers

The experimental results of throughputs are illustrated in Fig. 4-7. We can see the larger the number of routers, the lower the performance because packets flow through more routers producing many more unnecessary overheads. That is when a packet P arrives at a routers R, P will enter R's message queue and wait for being processed and transmitted. As stated above, more routers result in longer accumulated queuing and transmission delays. Further, many more routers also cause higher probability of packet loss rate and flow congestion. Hence, the performance is lower.

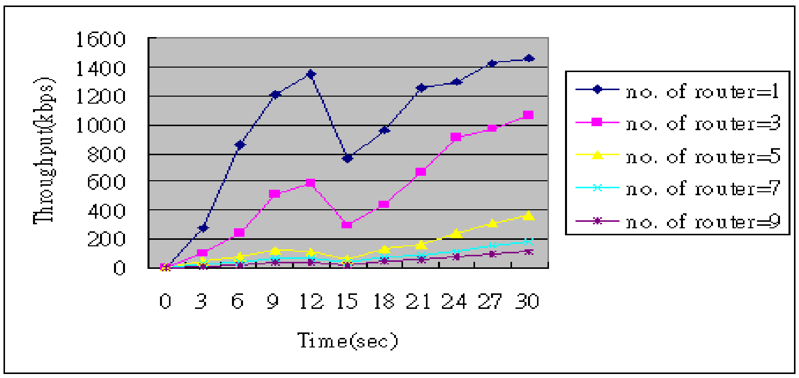

Fig. 4-7 Throughputs of the PSASP given different numbers of routers

The experimental results of packet loss rates are illustrated in Table 4-3. When the number of routers increases, packets are transmitted though more nodes, resulting in higher drop rates, of course higher packet loss rates. For example, if there are n routers on a path and their drop rates are respectively $P_1, P_2, P_3, \ldots,$ and P_n. Then, the probabilities that a packet can be successfully delivered by them, denoted by P_S, $P_S = \prod_{i=1}^{n}(1-P_i)$. A larger n will yield a smaller P_S.

Table 4-3. Packet loss rates for different numbers of routers

Routers	No. of	No. of	No. of	Packet loss

	Packets sent	Packets received	packets lost	rate(%)
1	26375	26305	70	0.265
3	24276	24206	70	0.288
5	22140	22073	67	0.302
7	16123	16070	53	0.328
9	12597	12548	49	0.388

Now, we can conclude that many more routers, e.g., n routers, will cause more transmission overheads since a packet when passing through a router has to wait to be processed and sent, and the packet has to propagate and be transmitted n+1 times. An ACK has similar phenomena. Both increase the total waiting time and degrade the performance.

4.4. Performance on Different Error Rates

In the third experiment, we evaluate the four tested systems given different error rates, including 2%, 5%, 8%, 10% and 15%, to see how error rates affect a system. We can see the end-to-end delays of the four schemes as shown in Fig. 4-8 are not significant different. Due to selecting the path with the shortest delay, the end-to-end delays of the PSASP are less than those of others. The reason is the same as that of the first experiment. The path with wider bandwidth and better transmission performance can decrease the queuing delay and end-to-end delays.

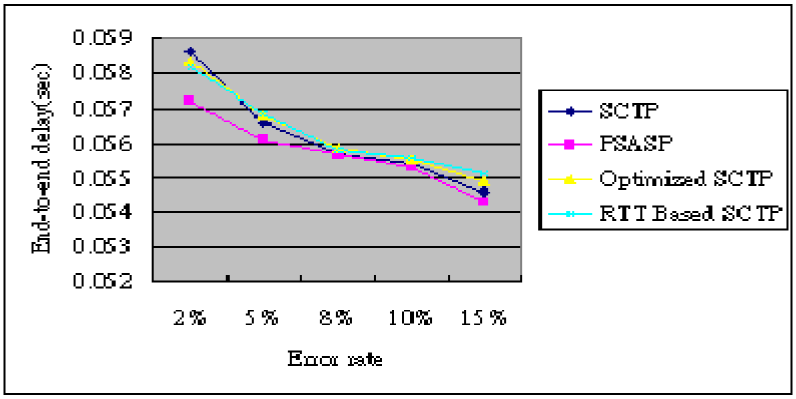

Fig. 4-8 End-to-end delays of the four tested schemes given different error rates

Fig. 4-9 illustrates the experimental results of average jitters. We can see the four schemes have different ranges of jitters. But, due to choosing the path with less delay the PSASP's jitters are the smallest compared to other schemes'. So, it is suitable for transmitting multi-media and audio data.

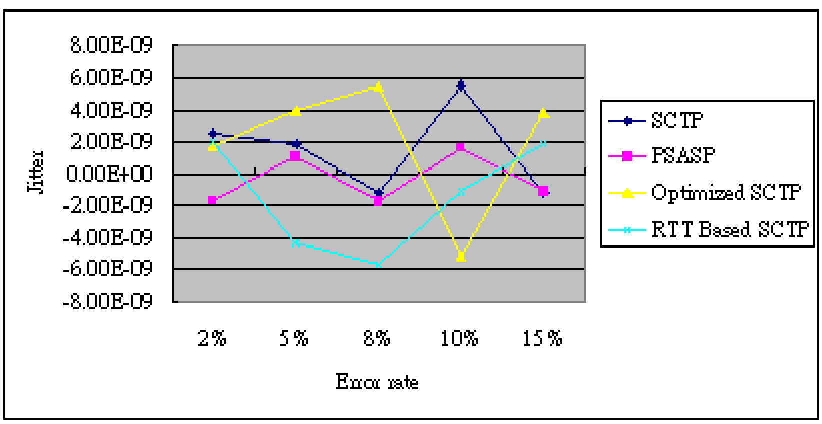

Fig. 4-9 Jitters of the four tested schemes given different error rates

Fig. 4-10 illustrates the experimental results of average throughputs. When the error rates increase, the performances decrease sharply. But, the PSASP outperforms the others. In theory, when the error rates increase from 2% to 15 %, the throughputs will decrease from 98% to 85%. But, the resulting throughputs are rather small. Since each time when a packet due to loss is retransmitted, the RTO increases doubly. Hence, a packet transmitted on a high error-rate path needs to wait for longer RTO time, thus lowering throughputs.

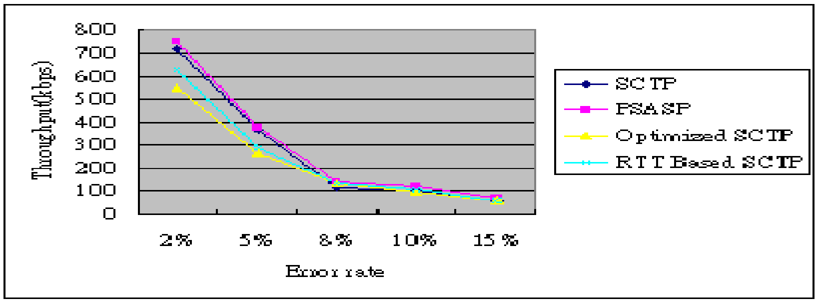

Fig. 4-10 Throughputs of the four tested schemes given different error rates

At last, we can conclude that higher error rates cause higher packet loss rates, many more retransmitted packets and many more overheads. Particularly, when many ACKs are lost, the corresponding data packets will be retransmitted at least twice, consequently consuming wider bandwidth, causing longer end-to-end delays, and resulting in worse throughputs and jitters.

4.5. The Scale of Four Delays

In this experiment, we analyze the four abovementioned delays and their scales given a sender and receiver.

Table 4-4. The scale of four delays

Delays	Scale (sec)

Average processing delay	0.00629
Average transmission delay	0.005872
Average propagation delay	0.05
Average queuing delay	0.20353
Average end to end delay	0.266836

The experimented results as listed in Table 4-4 show that the average processing delay and average transmission delay are only 5 and 6 ms, respectively. Their scales are relatively smaller than the other two's because their performance heavily depends on hardware speed and bandwidth of network interfaces. Processing data with hardware can often obtain very good performance. The average propagation delay and queuing delay are about 50 ms and 200 ms, respectively. The latter is closely to our average end to end delay. Now, it is clear that bottleneck of data transmission is queuing delay. This is, packets wait for being processed and delivered in the message queue. When the queue is full, the following packets will be dropped. The longer waiting time will cause longer end to end delay. The phenomenon is also true in the second experiment, i.e., the more numbers of routers result in higher queuing delay. Besides, we measured the average queuing delays which are listed in Table 4-5 given different number of routers. We can see when numbers of routers increase, the queuing delays also increase. Our conclusion is that the path with wider bandwidth and better transmission performance can decrease the queuing delay and end to end delay.

Table 4-5. Average queuing delay with given different number of routers

Routers	Average queuing delay (sec)	Average end-to-end delay (sec)	Percentage (%) (=Average queuing delay / Average end-to-end delay)
1	0.21206	0.26906	78
3	0.23038	0.28538	80
5	0.302201	0.36820	82
7	0.414312	0.476312	86
9	0.526548	0.586548	89

4.6. H Paths Switching Cost

In this experiment, we evaluate the path's switch cost and discuss how numbers of paths affect the delays. Switchover occurs at 10^{th} sec, and we measure the time period from when the tested system starts up to the time point when the secondary path's fist packet is successfully delivered and processed. The time measured is 10.01008 sec, indicating that switchover costs 10 ms. In our simulation environment, when the primary path fails, the SCTP calculates the remaining H-1 paths' cost differences. If we compare paths by pair, time of comparison of the remaining H-1 paths is H-2. The average cost for evaluating cost difference of two paths is above 5 ms. If H = 10, the cost will be 50 ms. So, when H is higher, the evaluation cost will be also higher. The SNMP packet format [34] consists of three parts, including SNMP Header (4 bytes), SNMP PDU Header (12 byte) and PDU data. The PDU data are OID length and OID data. We only need to access the four OIDs from MIB in section 3.3, so the PDU data does not exceed 4 bytes. Hence, the SNMP packet size is 20 bytes in length, which is very smaller than the data packet size (1468 byte). So, we can conclude that the SNMP packets do not significantly affect occupied bandwidth and the following data transmission. If the bandwidth of a path is 2 Mbps that means it can transmit 250000 bytes, the percentage of SNMP packets generated on each MIB retrieval from the n routers is 20*n / 250000 (bytes). If n = 10, it means SNMP packets only increase 0.0008 % of network traffic which is negligible.

Chapter 5: Conclusion and Future Research

In this study, we develop a new path selection and switching scheme for the SCTP, the PSASP, which considers the key path performance influential factor, round-trip delay, to select a primary path for the SCTP so as to provide the SCTP network transmission with wider bandwidth and a more reliable environment. The round-trip delay is the time required to successfully deliver a packet and receive the corresponding ACK. We further decompose the round-trip delay into processing, transmission, propagation and queuing delays, and analyze the influential factors of the four delays.

We also consider the PSASP's retransmission costs on different retransmission counts (i.e., $k < PMR+1$ and $k \geq PMR+1$) and different paths' packet loss rates where a packet's packet loss rate is also the path's transmission failure probability. This helps us to infer the average costs of packet delivery and retransmission.

Experimental results show that the PSASP can accurately evaluate performance of alternate paths so as to select the one with the widest bandwidth as the primary path. This is why the PSASP outperforms the other three tested schemes.

In the future, we would like to derive the PSASP's mathematical model of reliability which is a formal model. So, a user can realize the reliability of the PSASP before using it. We will also study how the considered parameters affect the arrival and service rates of a path segment. So that we can more preciously estimate the two paths. The purpose is to accurately estimate queuing delay. Those constitute our future research.

References

[1] F.Y. Leu, “A Novel Network Mobility Handoff Scheme using SIP and SCTP for Multimedia Applications,” Journal of Network and Computer Applications, Vol. 32, 2009, pp. 1073 – 1091.

[2] C.Z. Yang, W.K. Chang and I.H. Huang, “CS-SCTP: A Collaborative Approach for Secure SCTP over Wireless Networks,” In Proc. of the IEEE International Region 10 Conference 2007, pp. 1 – 4.

[3] P. Behbahani, V. Rakocevic and J. Habermann, “nSCTP: A New Transport Layer Tunnelling Approach to Provide Seamless Handover for Moving Networks,” the IFIP International Conference on Mobile Wireless Communications Networks, 2007, pp. 71 – 75.

[4] Information technology - Open System Interconnection Basic Reference Model: The Basic Model, ISO/IEC Std. 7498-1, 1994.

[5] P. Natarajan, R.I. Janardhan, D.A. Paul and R. Stewart, “SCTP: An Innovative Transport Layer Protocol for the Web,” the International World Wide Web Conference, 2006, ACM 1-59593-323-9/06/0005.

[6] E.P. Rathgeb, C. Hohendorf and M. Nordhoff, “On the Robustness of SCTP against DoS Attacks,” the International Conference on Convergence and Hybrid Information Technology, Vol. 2, 2008, pp. 1144 – 1149.

[7] H.C. Kim, Y.H. Kim, K.J. Kim and J.W. Chung, “High-Performance Data Transfer Using SCTP-based Compact Association Scheme,” In Proc. of the International Conference on Computational Science and its Applications, 2007, pp. 389 – 398.

[8] M.M. Monowar, M.O. Rahman, C.S. Hong, "Multipath Congestion Control for Heterogeneous Traffic in Wireless Sensor Network," International Conference on Advanced Communication Technology, 2009, pp. 1711 – 1715.

[9] S. Fallon, P. Jacob, Y.S. Qiao, L. Murphy, E. Fallon and A. Hanley, “SCTP Switchover Performance Issues in WLAN Environments” In Proc. of the IEEE International Conference on Consumer Communications and Networking Conference, 2008, pp. 564 – 568.

[10] H. Hassan, M. El-Shehaly and A. Abdel-Hamid, “Routing and Reliable Transport Layer Protocols Interactions in MANETs,” In Proc. of the International Conference on Computer Engineering and Systems, 2007, pp. 359 – 364.

[11] M. Dahal and D. kr. Saikia, “RTT Based Congestion Control and Path Switching Scheme for SCTP,” In Proc. of the International Conference on Communication Technology, 2006, pp. 1 – 4.

[12] E.P. Ribeiro and V.C.M. Leung, “Minimum Delay Path Selection in Multi-homed Systems with Path Asymmetry,” IEEE Communications Letters, Vol. 10, I. 3, Mar 2006, pp. 135 – 137.

[13] Internet Engineering Task Force (IETF), “RFC 2960 (Stream Control Transmission Protocol)”, 2000, http://www.faqs.org/rfcs/rfc2960.html.

[14] S.J. Koh, M.J. Chang and M. Lee, ”mSCTP for Soft Handover in Transport Layer,” IEEE Communications Letters, Vol. 8, I. 3, March 2004, pp. 189 – 191.

[15] S. Fu, M. Atiquzzaman, and W. Ivancic, “SCTP over Satellite Networks,” IEEE Workshop on Computer Communications, 2003, pp. 112–16.

[16] S.T. Kim, S.J. Koh, and Y.J. Kim, “Performance of SCTP for IPTV Applications,” The 9th Advanced Communication Technology International Conference, Vol. 3, 2007, pp. 2176 – 2180.

[17] C.E. Perkins and P. Bhagwat, “Highly Dynamic Destination-Sequenced Distance Vector (DSDV) for Mobile Computers,” ACM SIGCOMM Conference on Communications Architectures, Protocols and Applications, 1994, pp. 234-244.

[18] C. Perkins, E. Royer and S. Das, “Ad hoc On-demand Distance Vector (AODV) Routing,” IETF RFC 3561, July 2003.

[19] D. Johnson, D. Maltz, and Y.C. Hu, “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks for IPv4,” IETF RFC 4728, Feb. 2007.

[20] A. Al-kaisan, Md. Ashrafuzzaman, and S.M.M. Ahsan, “Reducing Congestion Collapse and Promoting Fairness in the Internet by Optimizing SCTP,” In Proc. of the IEEE Computer and Information Technology, 2008, pp. 1 - 5.

[21] R. Ohlendorf, A. Herkersdorf, T. Wild, "FlexPath NP: a network processor concept with application-driven flexible processing paths," International Conference on Hardware/Software Codesign and System Synthesis, 2005, pp. 249 – 284.

[22] H.Y. Kim, S. Rixner, "Performance Characterization of the FreeBSD Network Stack,", Rice University Computer Science Technical Report TR05-450, June, 2005.

[23] National Institute Of Standards and Technology, “Federal Information Processing Standard Publication 197, the Advanced Encryption Standard (AES),” Nov. 2001.

[24] K. Kamphenkel, M. Blank, J. Bauer, G. Carle, "Adaptive encryption for the realization of real-time transmission of sensitive medical video streams," International Symposium on World of Wireless, Mobile and Multimedia Networks, 2008, pp. 1 – 6.

[25] J. Fox, *Applied Regression Analysis, Linear Models, and Related Methods*, Sage Publications, February 5, 1997.

[26] D. Gross and C.M. Harris, *Fundamentals of Queuing Theory*, Third Edition, Wiley-Interscience, 2005.

[27] Exponentially Weighted Moving Average (EWMA), http://www.itl.nist.gov/div898/handbook/pmc/section3/pmc324.htm, June 3 2009.

[28] Internet Engineering Task Force (IETF), "RFC 4293 (Management Information Base for the Internet Protocol (IP))", 2006, http://tools.ietf.org/html/rfc4293.

[29] CISCO-IETF-SCTP-EXT-MIB, http://www.oidview.com/mibs/9/CISCO-IETF-SCTP-EXT-MIB.html, Jun 31 2010.

[30] Y. Qiao, E. Fallon, J. Murphy, L. Murphy, X. Zhu, G. Hayes, A. Matthews and A. Hanley, "Performance Analysis of Multi-homed Transport Protocols with Network Failure Tolerance," Communications, IET Vol. 2, I. 2, Feb. 2008, pp. 336 – 345.

[31] Bayes' Theorem, http://plato.stanford.edu/entries/bayes-theorem/, Oct. 15 2009.

[32] A. Caro and J. Iyengar, "ns-2 SCTP module", Version 3.5, http://www.armandocaro.net/software/ns2sctp/.

[33] UC Berkeley, LBL, USC/ISI, and Xerox Parc, "ns-2 documentation and software", Version 2.33, May 31 2008, http://www.isi.edu/nsnam/ns.

[34] R.Z. Lin, A. Liao and H.C. Chao, "Implementing SIP-Based Technology for Management Framework," the International Conference on Mobile Technology, Applications and System, 2008, ACM 978-1-60558-089-0.