修士論文

IP/DVB 伝送方式の改良とその評価

Master's Thesis

Improvement of IP over DVB Transport Mechanism and its evaluation

東京大学院　新領域創成学研究科　基盤情報学専攻

Graduate School of Frontier Sciences, The University of Tokyo,

Department of Frontier Informatics

47- 46324

ズルヒルミビンズルキフリ

Zul Hilmi Bin Zulkifli

指導教官　中山雅哉助教授

January 2006

In Loving Memories of My Mother

Contents

Abstract

As an open standard DVB system provides an affordable mean to use broadcast channel such as satellite for IP services. DVB uses MPEG2 Transport Stream at Layer 2. Size of MPEG2-TS cell is fixed to 188 bytes but generally IP packet could not fit perfectly into integral number of the cells. The left over space whether will be ignored (padding mode) or will be filled with next incoming packets. This thesis studies the shortcomings of these modes and propose section packing mode with threshold setting to overcome the deficiency. In this thesis, a complete system has been developed to emulate the behavior of IP/DVB over satellite links for the performance evaluation. Efficiency, packing delay and Delay Differences performance evaluation between proposed method and conventional method has been carried out.

Acknowledgment

I would like to thank Associate Prof. Masaya Nakayama for the enthusiastic supervision, advice and support throughout this research. I also would like to thank Prof. Yasushi Wakahara for advices and comments on my work and to Dr. Fumitaka Nakamura for the technical counsel.

I would like to thank to all my lab mates for direct and indirect support in finishing the thesis especially Mr. Kamiyama, Mr. Okumura, Mr. Murota and Mr. Saito. I am also grateful to the staff of the Information Technology Center for their support especially secretary Ms. Yoshizawa and secretary Ms. Kawasaki.

Finally to my wife, thank you for your love and boundless support.

Chapter 1

Introduction

1.1 Motivation

Transporting Internet Protocol (IP) packet over satellite channel is an alternative to provide IP services for those in areas and countries not covered by good terrestrial link.

Digital Video Broadcast (DVB) is an open standard proposed by European Telecommunication Standard (ETSI) and provides affordable means for IP communication using satellite. As an open standard, DVB system is expected to reduce the deployment cost since related equipments can be mass produced.

DVB standard utilized MPEG-2 Transport Stream (MPEG-2 TS) as the link layer technology. MPEG-2 TS cell is a fixed size 188 bytes cell. Generally, IP packets do not fit into integral number of TS cells. The left over part will be ignored and padded and hence reduce efficiency to transport IP.

Efficiency can be improved by packing multiple IP packets together in a TS cell. This mechanism is also known as section packing mode. However, usage of section packing mode may cause a packet to wait forever to be packed. This will lead to packing delay problem where IP packet can not be fully transmitted until next packet arrives.

In this thesis, I propose setting threshold values to section packing mode so that section packing and padding mode can be utilized together. I also propose a testbed architecture to emulate IP over DVB transmission without having to use real satellite link for performance analysis. Performance comparison of the proposed method is carried out with the padding mode and section packing mode.

Results of this thesis will enable us to understand relationship between threshold values and the efficiency as well as Packing Delay and Delay Differences.

1.2 Thesis Organization

Chapter 2 will give an overview and the introduction of basic DVB architecture. It also covers encapsulation mechanism in DVB system with the details of section packing mode and padding mode mechanism and the efficiency comparison. Chapter 3 discusses issues in current implementation of section packing mode and my proposal will be described in Chapter 4. Chapter 5 will shows the details about the performance evaluation experiments and the results will be discussed in Chapter 6. Finally, in Chapter 7, the conclusions have been drawn based on the analysis results and some possible future work have been suggested.

Chapter 2

Background

This chapter gives an overview of the current Digital Video Broadcast (DVB) System architecture specifically the mechanism to transport Internet Protocol (IP) packet over DVB satellite. This chapter also gives details about Ultra Light Encapsulation (ULE), the main encapsulation mechanism that we use throughout the thesis.

2.1 DVB System Architecture

ETSI defined various DVB standards to adapt to different link technologies. Initially, DVB is intended for digital audio video broadcasting. However since it has the potential to transport IP through a broadcast medium, ETSI provide extension to transport IP based packet into DVB systems. Example of DVB standard is DVB for cable (DVB-C), DVB for terrestrial wave (DVB-T) and etc. For satellite link, ETSI defined DVB for satellite (DVB-S) and DVB for return channel via satellite (DVB-RCS).

In IP over DVB-S, the forward link is a broadcast channel with receive only characteristic as seen by the satellite interface terminals (SIT). The return link could be a one-way channel and also could be a network connection permitting two-way operation using ordinary terrestrial link such as PSTN. This kind of connection is called UniDirectional Link (UDL) and various works have been done to address routing problems in UDL. Figure 2.1 shows the DVB-s topology.

In recent development, DVB with return channel via satellite (DVB-RCS) have been standardized enabling full independency to the terrestrial network. DVB-RCS is a modified version of DVB-S with additional standards on how to create interactive return channel using satellites. Figure 2.2 shows DVB-RCS topology, which can be characterized by having broadcast channel in forward link and point to point (PPP) channel in return the link.

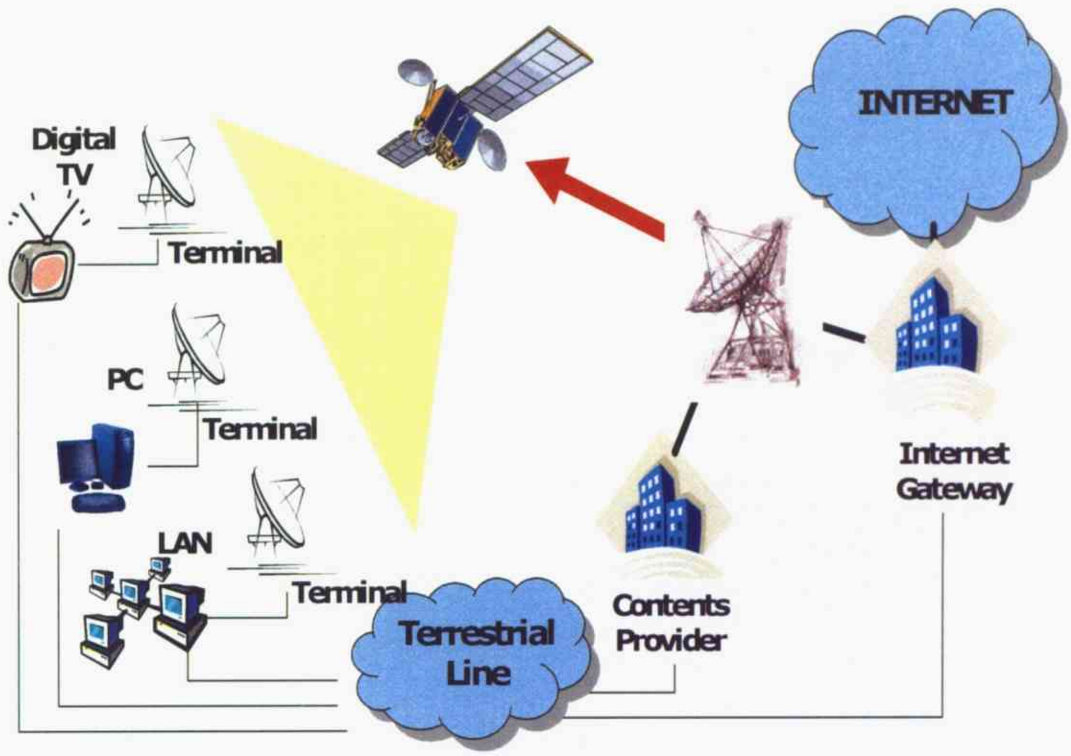

Figure 2.1: DVB-s Topology

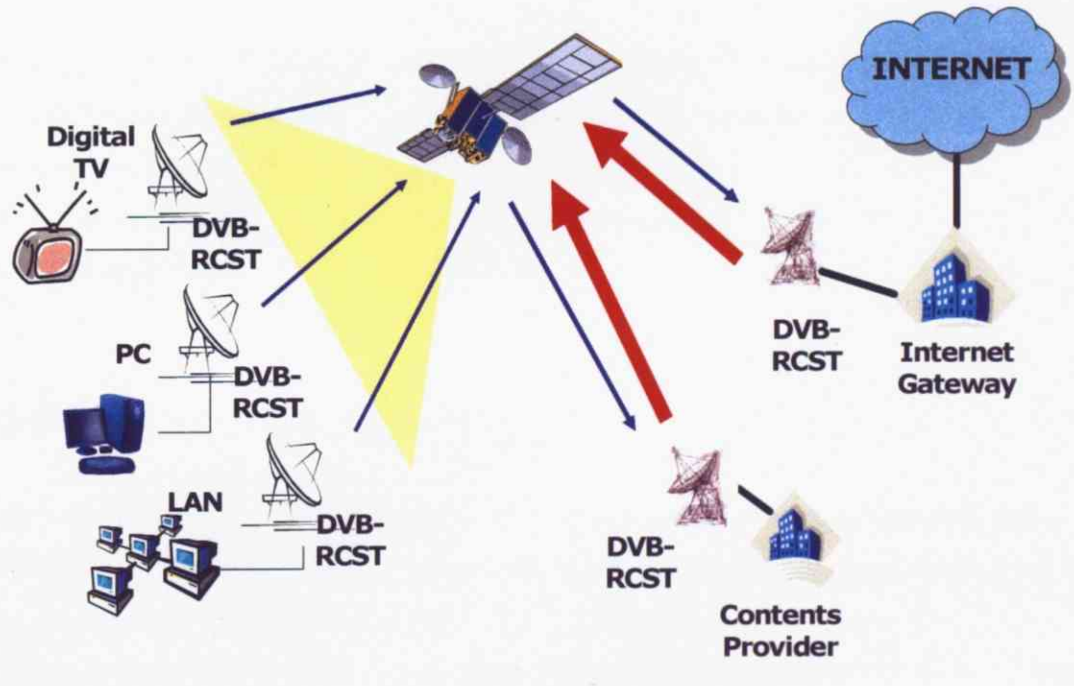

Figure 2.2: DVB-RCS Topology

DVB systems basically utilize fixed size 188 bytes MPEG-2 Transport stream (MPEG-2 TS) cell to carry packetized data in the forward link based on ISO/IEC 13818-1 MPEG-2 systems standard. DVB-RCS in other hand uses ATM/AAL5 53 bytes cells by default but also defines optional MPEG-2 TS usage in the return link.

2.2 Link Layer Technologies in DVB System

Combining DVB-S and DVB-RCS, DVB systems in satellite environment have two different layer 2 formats; using MPEG-2 Transport Stream (TS) cell (broadcast link) or Asynchronous Transport Mode (ATM) cell (DVB-RCS return link).

2.2.1 Asynchronous Transport Mode

When using ATM cell, IP packet is encapsulated according to AAL5 Virtual Circuit (VC) based multiplexing. AAL5 VC based encapsulation appends 8 bytes trailer to Protocol Data Unit (PDU) or in this case the IP packet to make up Sub Network Data Unit (SNDU). SNDU then will be fragmented into the payload of multiple 53 bytes ATM cells. One ATM cell consists of 5 bytes header and 48 bytes payload. The padding bytes will be stuffed between the IP packet and AAL5 trailer whenever the resulted SNDU can't be fit perfectly into ATM cells (padding mode).

Detail discussions about ATM encapsulation will not be discussed since it is not in the scope of this thesis.

2.2.2 MPEG-2 Transport Stream

MPEG-2 TS (transport stream) initially used to transport compressed digital audio and video data. However MPEG-2 TS is also able to carry defined data containers such as IP packets in addition to audio and video.

Figure 2.3 shows 188 byte fixed sized TS cells. Each TS cell consists 4 bytes of header and 184 bytes payload. Error recovery is easier when using constant cell length (essential in error prone line).

Compressed data from a single source (video, audio etc) and additional control data for the source information form elementary streams (ESs). It then will be packetized into packetized ES (PES). Each PES packet consists of a header and payload and PESs from various elementary streams are combines to form a program.

Several program combine to form the TS with other descriptive data called program-specific information (PSI). PSI contains descriptive data about the network and also assignments of PESs and PIDs into the program. Examples of main PSIs are program association table (PAT), program map table (PMT), and network information table (NIT). Details about PSI are out of this paper's scope and will not be discussed in detail.

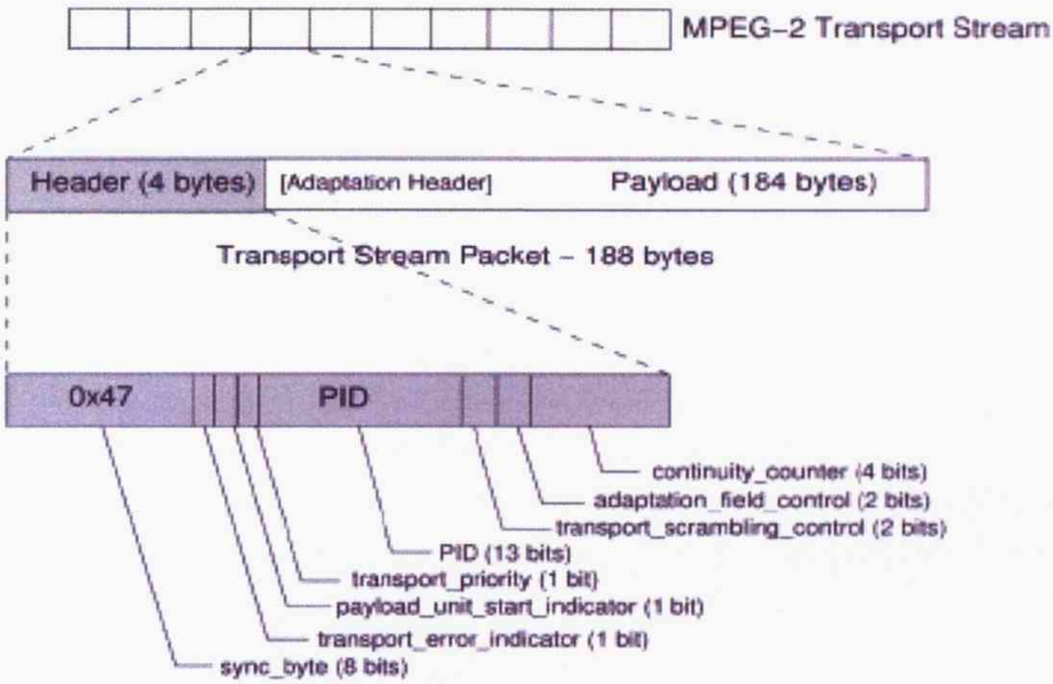

Figure 2.3: MPEG-2 TS structure

Semantic of TS packet header is shown below.

1. sync_byte (8 bit): 0x47 fixed value. Indicates the start of a TS cells.
2. transport_error_indicator (1 bit): 1 indicates an uncorrectable bit error in current TS.
3. Payload_unit_start_indicator (1 bit): 1 indicates presence of a new PES packet or PSI section. If PUSI = 1, first byte in the payload contains pointer_field that will be described later.
4. Transport_priority (1 bit): 1 indicates a higher priority that other packets.
5. Packet Identifier (PID) (13 bit): values 0x1FFF is null packet (ignored by the receiver). PIDs are used to distinguish between elementary streams (therefore each elementary stream have different PID) and different PSI. PIDs enable receiver to

differentiate the stream to which current received packet belongs.

6. Transport_scrambling_control (2 bits) Indicated the scrambling mode of the packet payload (refer table 2.1). It is used to encrypt payload.

Table 2.1. Coding of transport_scrambling_control

Value	Description
00	no scrambling of TS packet payload
01	reserved for future DVB use
10	TS packet scrambled with Even key
11	TS packet scrambled with Odd key

7. Adaptation_field_control (2 bits): 01 indicates only payload and no adaptation field, 10 indicates only adaptation field in the payload, and 11 indicates presence of adaptation field followed by payload.

8. Continuity_counter (4 bits): One continuity counter for a PID. Incremented for each non-repeated TS received for certain PID.

Total 32 bits header mentioned above can be followed by below optional header.

9. Payload_pointer_field (8bit) – presence if and only if PUSI equal to 1 in PSI packet. It indicates the number of bytes until the new section of PSI started after the pointer_field.

2.3 IP Packet Encapsulation into MPEG-2 TS Cell

To encapsulate IP packet into MPEG-2 TS cells, current standard for IP encapsulation is MultiProtocol Encapsulation (MPE). However in the near future, Ultralight Encapsulation (ULE) is expected to replace MPE since it has less and simpler overhead.

This section will give an overview about the encapsulation mechanism involved in MPE and ULE.

2.3.1 MultiProtocol Encapsulation (MPE)

MPE is the standard for IP encapsulation defined in DVB family of standards. It allows transmission of IP packets or Ethernet style frames in the control plane associated with audio/video transport. Data is formatted as if it were a Table Section.

MPE makes use of a medium access control (MAC) level device address and the address format follows the ISO/IEEE standards for LAN/MAN. IP packets can be as large as 65636 bytes in length and have to be fragmented so they do not exceed the Maximum transfer Unit (MTU) of the payload portion in DVB-MPE datagram section (DVB-MPE specified MTU as 1500 bytes).

Table 2.2: MPE Overhead

Name	Bits
Table_id (0x3E)	8
Section_syntax_indicator	1
Private_indicator	1
Reserved	2
Section_length	12
MAC_6	8
MAC_5	8
Reserved	2
Payload_scrambling_control	2
Address_scrambling_control	2
LLC_SNAP_flag	1
Current_next_indicator	1
Section_number	8
Last_section_number	8
MAC_4	8
MAC_3	8
MAC_2	8
MAC_1	8
LLC/SNAP_header (optional)	8

MPE packets have 12 bytes of header and 4 bytes of cyclic redundancy check (CRC) as the tail. Table 2.2 shows the MPE's header starting from the most significant bit.

MPE packets are suboptimal to carry IP packets since not all the header fields added are required to deliver IP packets to the destinations.

2.3.2 Ultralight Encapsulation

Ultra Lightweight Encapsulation (ULE) has been introduced in an Internet draft to eliminate unnecessary overhead in MPE. ULE encapsulates IP packets directly into a sequence of TS cells. Unlike MPE, ULE only has 4 bytes of header, cutting 8 bytes from the header in MPE and has 4 bytes of CRC or checksum trailer. Table 2.3 shows ULE's header format starting from most significant bit.

Destination Address Present Field is the most significant bit of the Length field. Default value of 0 indicates the presence of the Destination Address Field in the payload and value 1 indicates the opposite.

Length Field indicates the length in bytes of the payload, counting from the byte following Type field including CRC. Type field indicates the type of payload carried in the payload.

Table 2.3: ULE Overhead

Name	Bits
Destination_address_field	1
Length	15
Type	16
Destination_address (optional)	48

2.3.3 Encapsulation summary

Figure 2.4 summarizes the IP packet encapsulation into MPEG-2 TS cells in comparison to the Ethernet and ATM/AAL5.

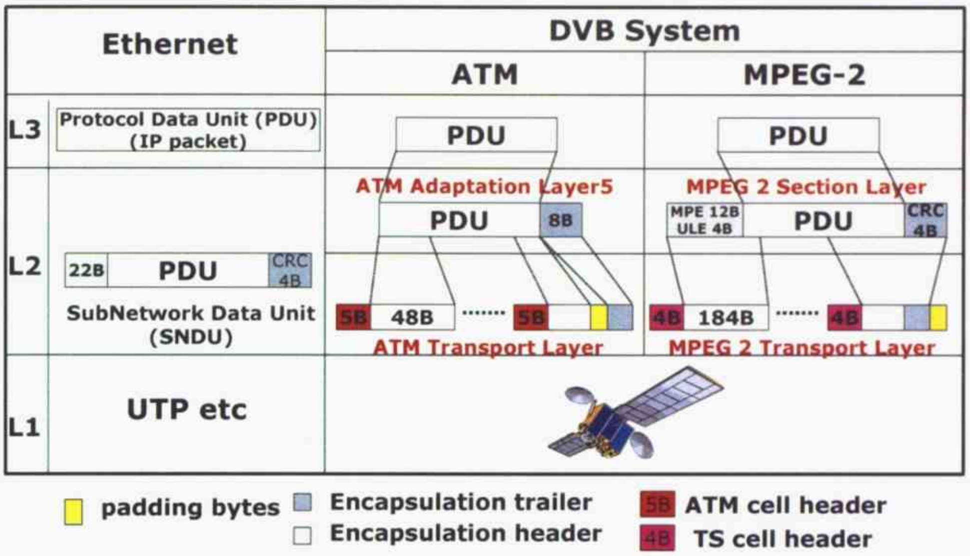

Figure 2.4. Encapsulation Summary

2.4 Padding and Section Packing

Inserting padding bytes into left over space in ATM cell as well as MPEG-2 TS cell in padding mode increases total overhead involved in transmitting an IP packet. By eliminating the padding bytes, overall efficiency could be increased.

In contrast to IP packet encapsulation into ATM cells that only defines padding mode, encapsulation into MPEG-2 TS cells also defines padding and section packing mode. Section packing mode is a mechanism to fully eliminate padding bytes.

Figure 2.5 shows the padding mechanism and figure 2.6 illustrates the section packing mechanism for ULE encapsulation into MPEG-2 TS cell. When a SNDU is not perfectly fit into MPEG-2 TS cell (or cells) and left some free space in the final cell, instead of inserting the space with padding bytes, section packing mode forces the cell to wait until next incoming SNDU. Then part of the next SNDU will be inserted into the remaining space. In case of ULE, this mechanism takes place only when more than 2 bytes, if payload pointer exists or 3 bytes when not, leftover space remains.

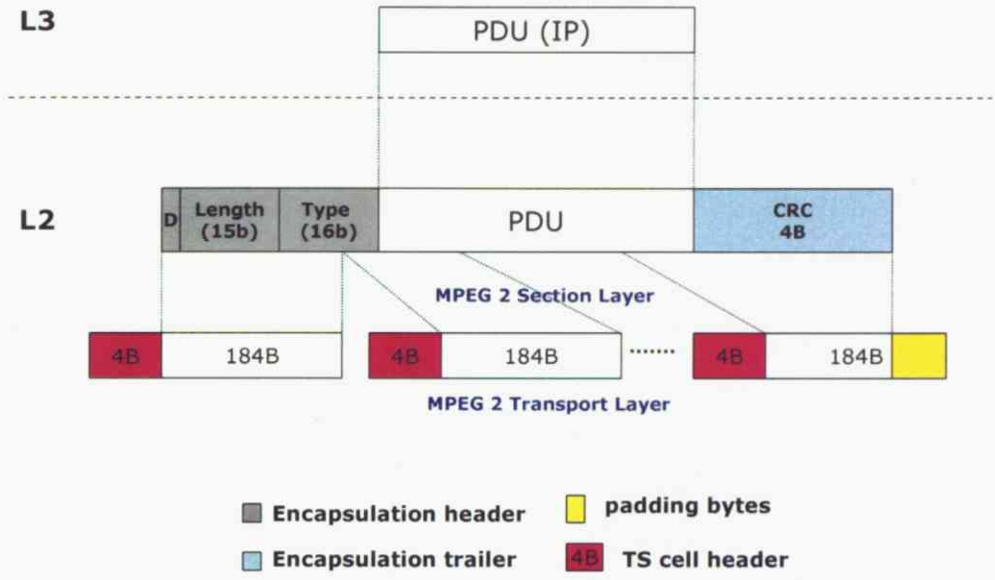

Figure 2.5: Padding Mechanism

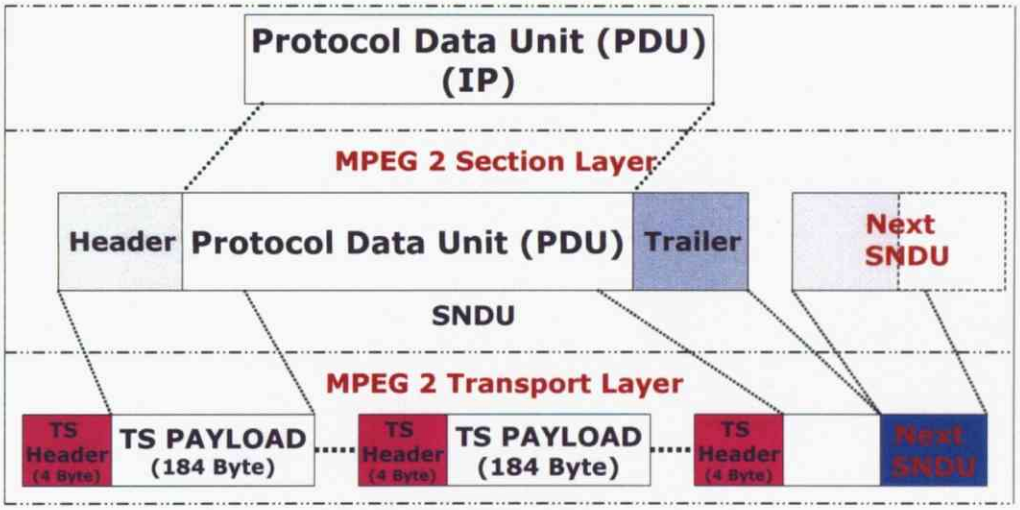

Figure 2.6 Section Packing Mechanism

2.5 Efficiency comparison

This section will compare the efficiency to transport IP packet over DVB system with the well known Ethernet encapsulation and also with ATM/AAL5, the default encapsulation mechanism for return link in DVB-RCS.

First I present the efficiency equation for each encapsulation method. Later I show efficiency comparison between each method. In this paper I define efficiency as the efficiency to transport IP packet into layer 2 frame or cell.

2.5.1 MPE/ULE Efficiency Equation

MPE and ULE have similar efficiency equation. MPE has 16 bytes total overhead without LLC/SNAP and 24 bytes overhead with LLC/SNAP.

Meanwhile total overhead for ULE is 8 bytes without Destination Address, 14 bytes with Destination Address, 22 bytes for Ethernet bridging, and 28 bytes for Ethernet bridging with Destination Address.

There are two scenarios to calculate the efficiency. First when a single IP packet is encapsulated into TS cells without concatenation with other IP packets (Padding Mode). The later is when multiple IP packets can be concatenated into a TS cell (Section Packing Mode).

Each MPEG cells have 4 bytes of header and 184 bytes of payload. The first cell will have 1 byte payload pointer, so it will have only 183 bytes of payload. If section packing is OFF, then IP packet only start at the beginning of Transport stream cell, and the remainder will be padded by stuffing bytes.

If L denotes the total overhead of the section layer (8, 14, 16, 22, 24, 28) and S denotes IP packet length, the total cells n required to transmit an IP packet can be denoted by:

$$n = \left\lceil \frac{S+L+1}{184} \right\rceil \qquad (1)$$

Where $\lceil x \rceil$ is the smallest integer greater or equal to x. Then the number of padding bytes, p can be defined as:

$$p = 184 - [S + L + 1 - (n-1) \times 184] \qquad (2)$$

The efficiency will be:

$$E = \frac{S}{n \times 188} \qquad (3)$$

Now if section packing is ON, there will be no padding bytes. There will be 2 cases for efficiency calculation:

1. If $S > 183 - L$ then there will be one byte payload pointer for every IP packet, and for simplicity we can add the overhead into the section layer overhead. The transport overhead will be 4 bytes for each TS cells. This means total transport layer overhead per IP packet will be:

$$\frac{S+L+1}{184} \times 4 \qquad (4)$$

And the efficiency will be:

$$E = \frac{S}{S+L+1+[\frac{S+L+1}{184} \times 4]} \qquad (5)$$

2. If $S < 183 - L$ then there will be one payload per transport packet. The overhead (payload pointer) could be added to TS overhead making it 5 bytes per TS packet. Therefore total transport layer overhead per IP packet will be:

$$\frac{S+L}{184} \times 5 \qquad (6)$$

And the efficiency will be:

$$E = \frac{S}{S+L+[\frac{S+L}{184} \times 5]} \qquad (7)$$

3. If $S = 183 - L$, Efficiency equal to (3).

4. In ULE, TS packet can contain multiple SNDU. TS packet must at least contains 2 bytes of additional space before accepting another SNDU. Therefore when $S > 183 - L$ and padding byte p is equal to 0, 1 or 2, or $S < 183 - L$ and $p = 1$, overhead is equivalent to (3). The logic is when $S > 183 - L$; IP packet is devided into multiple TS packets. Then the last TS packet will not contain any payload pointer. In order to fit in a

new SNDU, at least 2 bytes must be free so that length field will not be divided into multiple TS packets. Then another byte is needed to put payload pointer. Therefore minimum 3 free bytes needed to put a new SNDU. Meanwhile when $S < 183 - L$, TS cell will already contain payload pointer and only needs 2 extra bytes for the next SNDU.

2.5.2 ATM Efficiency Equation

ATM AAL5 has the following overhead.

1. 8 bytes of AAL5 trailer.
2. 5 bytes overhead for every ATM cells.

At AAL5 layer, 8 bytes of trailer will be added to each IP packets. Each ATM cell is 53 bytes long and each cell has 5 bytes of overhead. Section packing could not be done in ATM encapsulation. If S denotes size of the IP packets, the total cells number n required to transport the IP packet can be express by following equation:

$$n = \left\lceil \frac{S+8}{48} \right\rceil \quad (8)$$

And the efficiency, E is:

$$E = \frac{S}{n \times 53} \quad (9)$$

2.5.3 Ethernet Efficiency Equation

Ethernet adds the following overhead to IP packets:

1. 8 bytes of preamble
2. 14 bytes of header (MAC add 12 bytes, Ethertype 2 byte)
3. 4 bytes of CRC

Total Overhead is 26 bytes.

Minimum Ethernet frame is 64 bytes (excluding preamble), therefore, packet less than

46 bytes will be padded. If IP packet size in bytes is denoted by S, Efficiency, E will be defined as:

1. $S \leq 46$ **:**

$$E = \frac{S}{72} \qquad (10)$$

2. $46 \leq S \leq 1500$:

$$E = \frac{S}{S + 26} \qquad (11)$$

2.5.3 Comparison

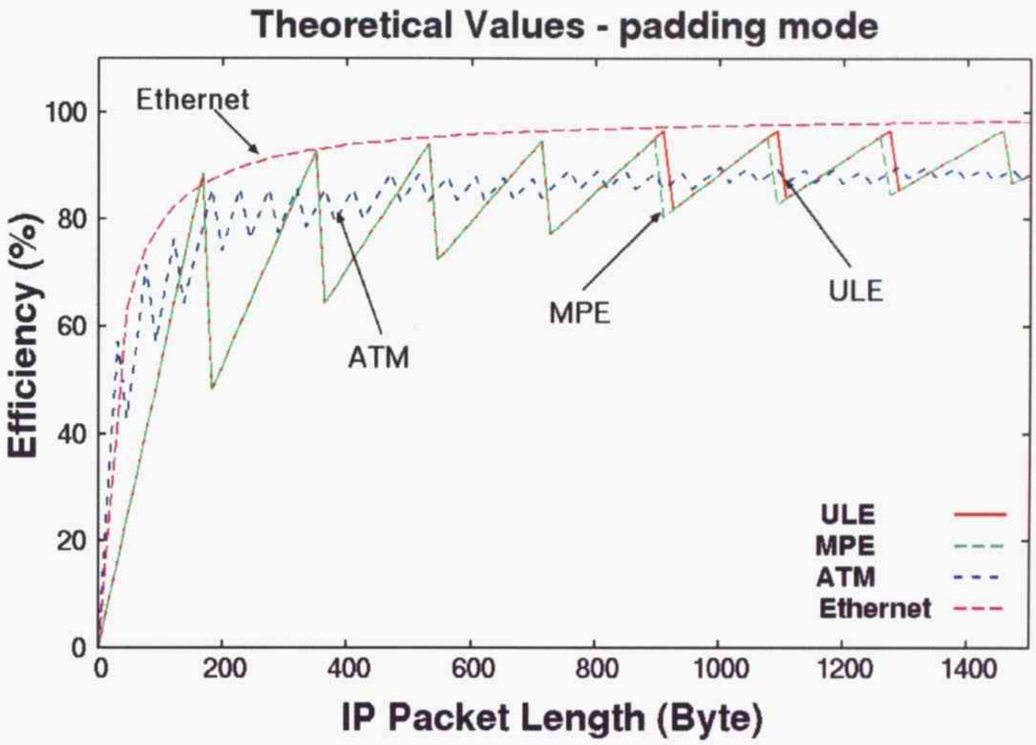

Figure 2.7: Efficiency in Padding Mode

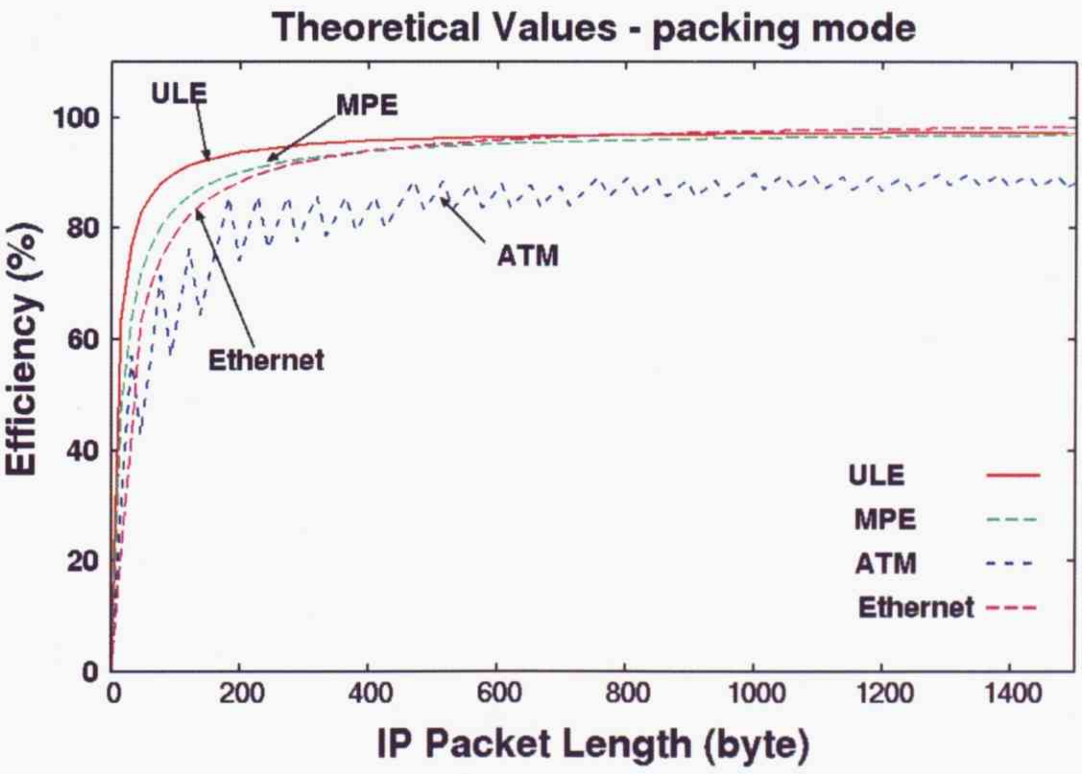

Figure 2.8: Efficiency in Section packing mode

From figure 2.7, we can see that most of the time MPE and ULE have the same efficiency values except for certain IP packet length where the packet can be perfectly fitted into TS cells without or less padding bytes. This is caused by padding bytes overhead that compensate the ULE header reduction. For small packets (around 180 bytes and below), ATM encapsulation seems to have better efficiency. From the graph we can conclude that ATM efficiency is more stable across the graph compare to MPEG-2 TS since ATM cell is much smaller than MPEG-2 TS cell.

Efficiency in section packing mode as illustrated in figure 2.8 shows MPEG-2 encapsulations perform better than Ethernet for IP packet less than 600 bytes long and ATM. In comparison, MPEG-2 section packing encapsulation is much more efficient than padding mode since it eliminates all the overhead by padding bytes.

Section packing mode shows the potential of MPEG-2 encapsulation to transport IP packets efficiently. When the packet length is big enough, it marks at least more that 10% better efficiency compare to ATM.

Chapter 3

Issues of Section Packing

In efficiency point of view, section packing is an excellent way to increase transmission efficiency as unnecessary overhead from padding bytes can be eliminated. However section packing also has some disadvantages compare to padding mode. This chapter discuss about the issues in section packing mode.

3.1 Section Packing and Packing Delay

The necessity to wait for next incoming packets in section packing mode introduces a new issue. In Figure 3.1, for SNDU3 to be fully transmitted, it has to wait until t5 where the last TS cell for SNDU3 is inserted with SNDU4 and a part of SNDU5. However compare to padding mode, sending SNDU3 could be completed in t3, the SNDU3 arrival time, when assuming the processing delay and queuing delay equal to 0 and multiple cells can be transmitted simultaneously. As the result, this will generate delay when comparing to padding mode. The delay is called packing delay (PD).

In this paper, I define packing delay (PD) as the time difference between packet arrival time and time when the SNDU can be fully transmitted. For example in SNDU3 case, the PD will be (t5 – t3).

PD is determined by the packet interval time. PD might take 1 interval or multiple interval time according to packet length sequence. Naturally, the longer interval time, means the higher PD values will be. Hence, when the traffic is heavier, PD is expected to be lower.

PD might hampers quality of the communications and affects the applications. It is especially not preferable for delay sensitive applications such as telesurgery, Voip and Video Conferencing [10,11,12]. In addition, TCP session might be effected if PD is long enough and able to trigger TCP timeout mechanism.

Current ULE implementation does not discuss the usage of threshold setting based on packing delay. It also does not discuss the effects of threshold setting to the packing delay property.

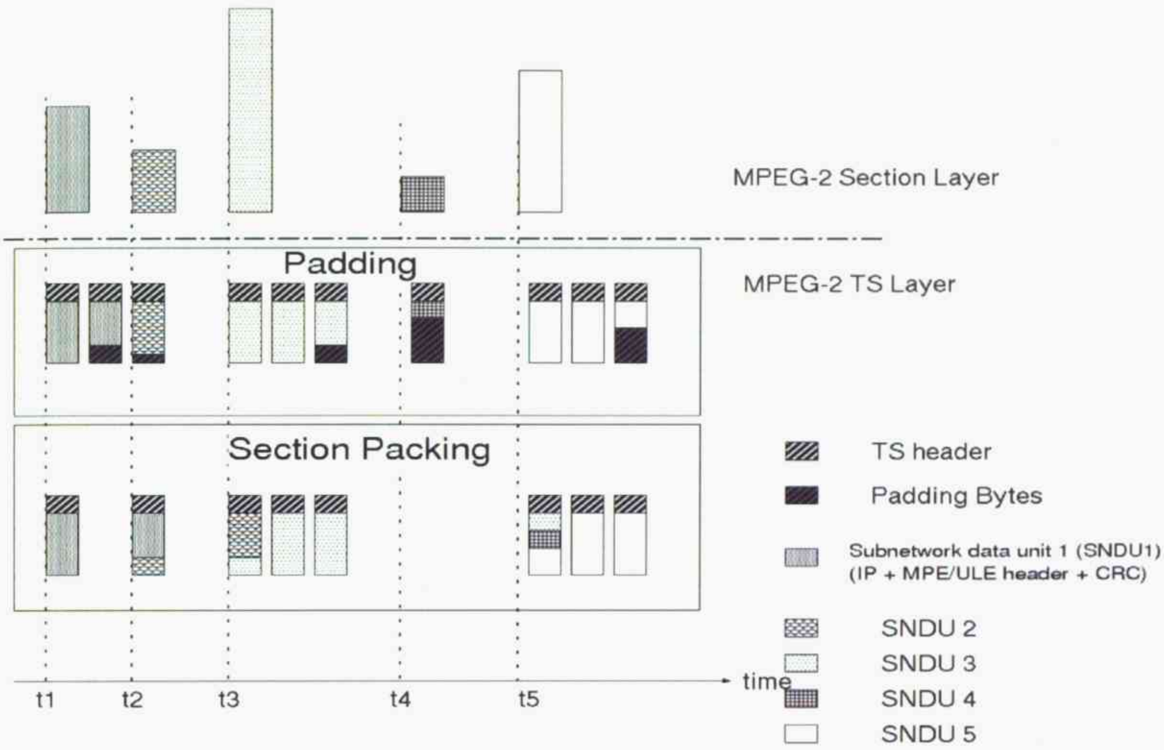

Figure 3.1: Packing Delay problem

3.2 Section Packing and Delay Differences

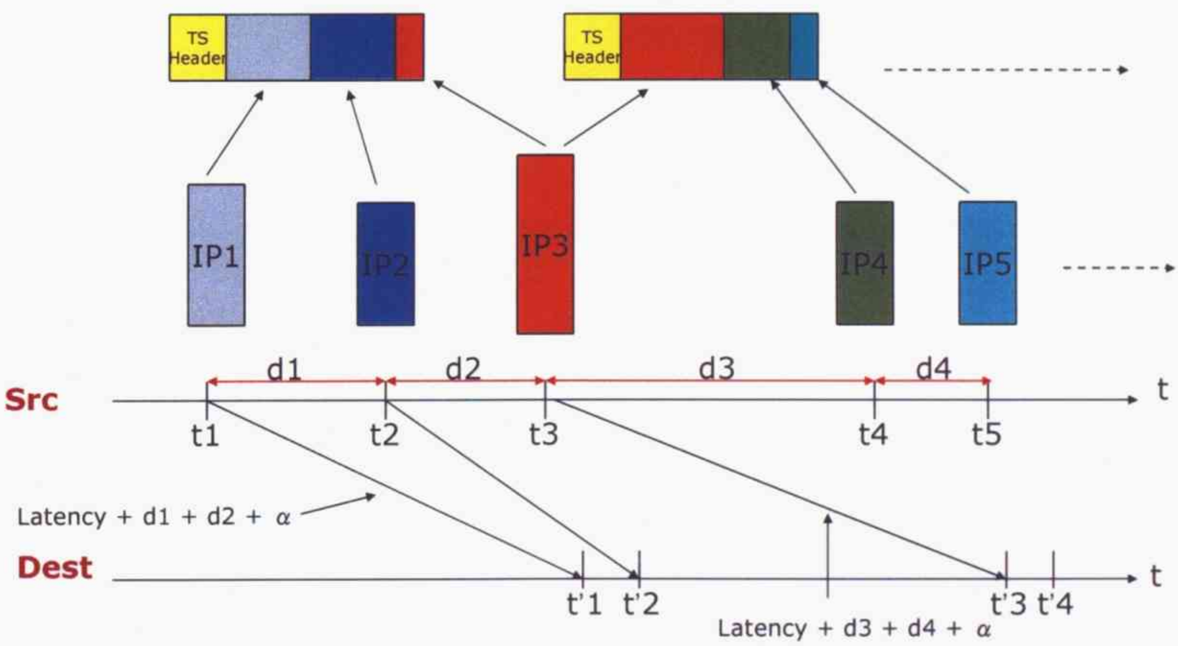

Figure 3.2: Delay Differences

Another issue of section packing is Delay Differences. Packing multiple IP packets into a single MPEG-2 TS will introduce Delay Differences to the connection. In padding mode, every packet is sent according to the time they arrive at the encapsulator. Interval time between packets in padding mode is supposed to be the same as the interval time between packets when they enter the encapsulator.

In section packing mode, IP packet that is packed together with packet that came before, will arrive earlier than expected. However the packet before is required to wait for it to be packed together. Therefore it will arrive later than it should be.

The differences between the arrival times at the destination introduce Delay Differences to the connection. Delay Differences is crucial to application like VoIP since it concern the feeling of the user and related to connection jitter.

Figure 3.2 illustrates the Delay Differences definition. Delay Difference for IP2 is equal to (t'2 – t'1) – (t2 – t1). Since for IP2, packet inter-arrival time at destination is shorter than at the source, the Delay Difference will have negative value. Meanwhile for IP3, Delay Difference is equal to (t'3 – t'2) – (t3 – t2). IP3 will have positive Delay Difference value since packet inter-arrival time is longer at destination.

3.3 Efficiency, Delay Differences and Packing Delay Tradeoff

For better efficiency, section packing should be used. But to improve Delay Differences and Packing Delay performances, padding mode should be used and bandwidth utilization has to be sacrificed. In section packing mode, no threshold values mean a packet may have to wait forever for the next packet to be packed and therefore might have infinite PD values and large variation in Delay Differences.

Chapter 4

Proposal of Packing Delay Threshold in Section Packing Mode

From chapter 2, we can observe that section packing mode is more efficient compare to padding mode. However in chapter 3 it is shown that section packing mode also introduces packing delay problems. This chapter will discuss the proposal to overcome the problem.

4.1 Packing Delay Threshold Setting

Ignoring the quality of section packing mode to overcome Packing Delay and Delay Differences problems is not efficient. Utilizing both section packing mode and padding mode in a link may reduce packing delay problem while increasing transmission efficiency.

Conventionally padding mode and section packing mode is used independently and not co-existing. Thus, we propose utilizing both modes by limiting packing delay as the threshold in section packing mode to overcome or at least reduce the Packing Delay and Delay Differences problems.

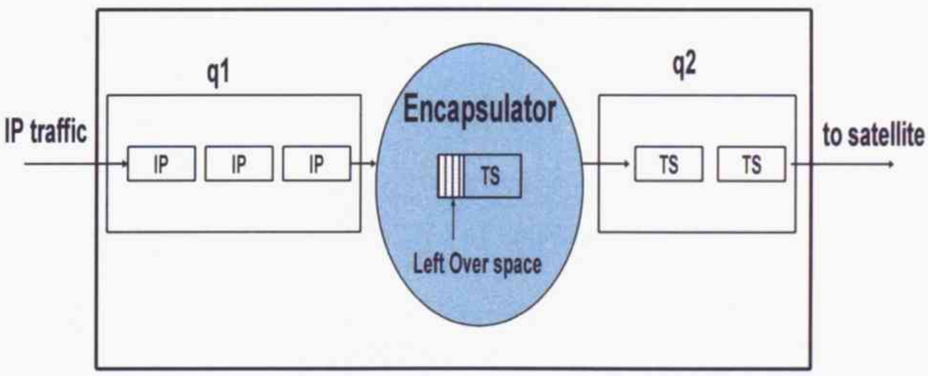

Figure 4.1 ULE encapsulation model

Figure 4.1 illustrates an ULE encapsulator model in satellite invironment. Incoming IP traffic will be queued in q1 before being processed. In padding mode however, the left over space will be padded and sent to q2 before being transmitted. Regardless of the queue condition in q1, all IP packets will be padded.

This thesis proposes threshold setting for section packing mode base on limiting Packing Delay (PD). In chapter 3, PD is defined as the time difference between packet arrival time (before being queued) and the time when the packet fully transmitted.

Threshold for section packing mode can be set according to network preferences. It is more efficient for one packet to wait a while at the encapsulator, if q1 is empty. Unlike section packing mode, the packet does not have to wait forever. The transmission can be delayed until total processing time is equal to threshold value to wait for the next incoming packet. Figure 4.2 shows the basic algorithm to implement the section packing mode with threshold.

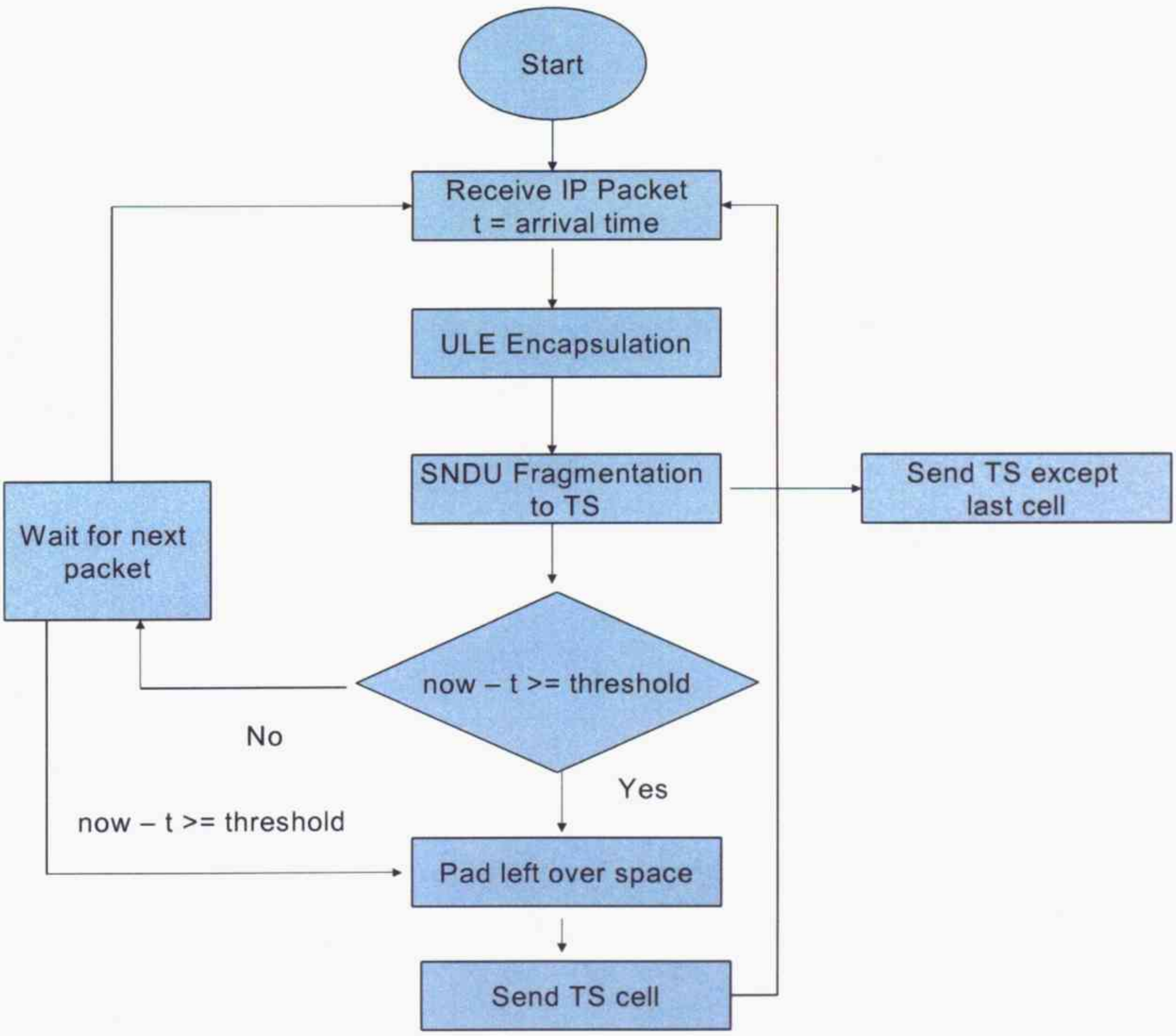

Figure 4.2 Algorithm for section packing mode with threshold

Chapter 5

Evaluation Method

This chapter will focus on the implementation details involve in the evaluation. Evaluation is done only to ULE encapsulation since it is expected to replace MPE as the standard to transport IP over DVB system. Performance of the ULE encapsulation in padding mode and section packing mode depends upon the actual traffic patterns, specifically packet length distribution and packet inter time.

We do analysis based on 3 different scenarios as illustrated in figure 5.1. These scenarios represent different traffic characteristic and therefore could produce different results. Client-server scenarios represent the extreme traffic characteristic. Client's traffic is usually small in size since there are mostly consists of request packets. However traffic from the server mostly contains the maximum size of IP packet length (MTU) since server has to answer request from the client and send data in a big bulk. Meanwhile network scenario consists of a mixture of these traffics. Client traffic represents the worst case scenario for efficiency performance and server traffic represents the best case scenario.

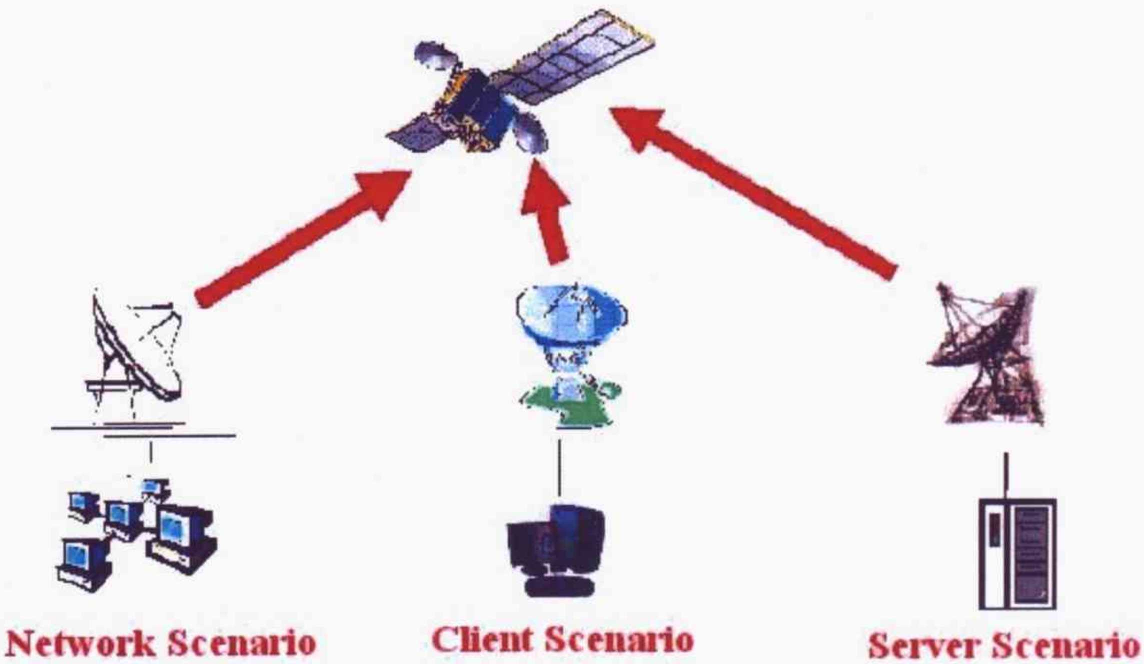

Figure 5.1 Evaluation Scenarios

Analyses for the efficiency to transport IP packets using MPEG-2 TS cell have been done. In other word, the efficiency to transport Layer 3 packet into Layer 2 cells.

Evaluation is divided into two parts: simulation and emulation. The details of the experiment setup are discussed below.

5.1 Simulation

Evaluation for simulation analysis is done to 3 scenarios. The simulation analysis is done by collecting traffic data from these scenarios and records it using tcpdump tool. The data then will be analyzed using a spreadsheet C-based calculation tools that have been develop for the experiment. The tools allowed analysis of the inter-packet arrival time and hence could predict the performance of our proposal compare to padding mode and section packing mode.

Simulation evaluation is done in assumption that we have enough buffer size to buffer the incoming packet. Processing time and queuing delay for every packet is assumed equal to zero. Section packing with threshold will send the IP packet immediately after the threshold and multiple MPEG2-TS cell can be sent simultaneously.

To evaluate my proposal, I present 2 different analysis based on 3 scenarios: single personal computer (PC), server, and local area network (LAN) shown in figure 5.1. These scenarios have different traffic characteristic and therefore could produce different results.

5.1.1 Network Scenario

For Network scenario, evaluation is done using a series of real IP over DVB traffic captured over 48 hours from a 10 Mbps unidirectional DVB link. The link had been used to provide inbound IP traffic for annually held WIDE Project Japan [9] 2005 spring camp. Figure 5.2 describe the basic network topology for the link. This link is a point-to-point unidirectional link with the outbound link is supported by other connections.

The traffic data is dumped in a machine located at the camp's venue which receives all IP packets via DVB link and relay it to local network. We assume that the traffic captured in the venue is equal to the traffic fed in the feeder from the Internet.
Below is the list of steps involved in doing the simulation evaluation.

1. Decide when to record traffic data
2. Decide which machine to record the data
3. Setup recording environment
4. Analyze traffic data using analysis tools to obtain evaluation results for each condition.
5. Post process the data for visualization.

The parameter involved in the experiment is listed below

1. Section packing threshold → Random threshold values of 20 ms and 200 ms are chosen for the evaluation. Padding mode has threshold values of 0 ms and section packing mode is infinity.
2. Bandwidth → According to link bandwidth
3. Latency → Natural latency of the link

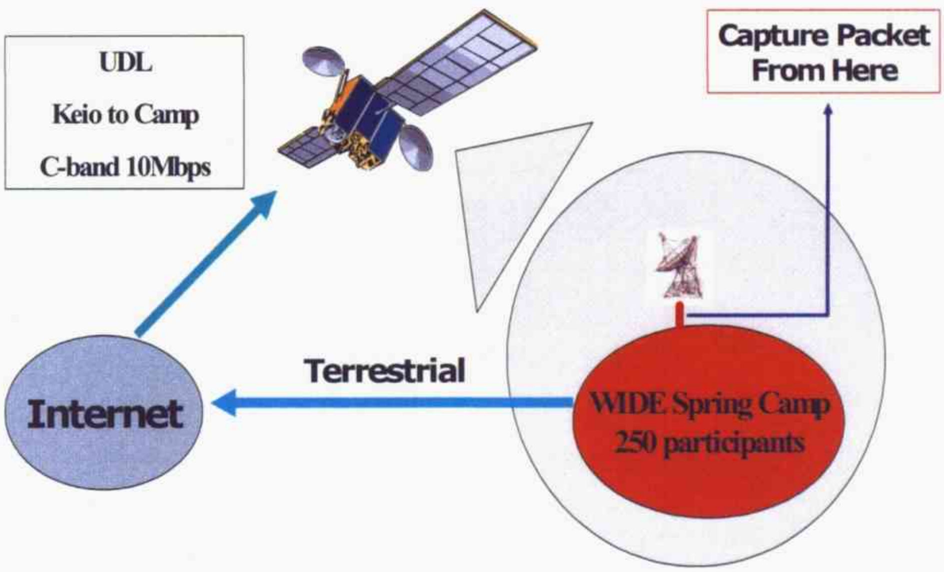

Figure 5.2 Wide spring camp DVB-RCS topology

5.1.2. Client Server Scenario

To simulate packet flows for a client and a server scenario, a testbed illustrated by figure 5.3 has been constructed. TCP connection is chosen to show the effect of a high latency satellite link to the packet inter-arrival time. Unlike UDP, TCP's packet rate is determined by the quality of the links between the client and the server. TCP client-server traffic characteristic represents the worst and best case scenario for efficiency performance and therefore I decided to do evaluation based on TCP client-server mode.

Satellite link emulation is done using a Dummynet machine. Client will request data from a HTTP server through the Dummynet. Dummynet machine will shape the traffic between the client and the server according to the bandwidth parameter. We choose 12 Mbps as the bandwidth parameter since it is the logical bandwidth for a satellite channel. Latency is set according to common GEO satellite latency of 250 ms since digital video broadcast satellite usually operates in GEO orbit.

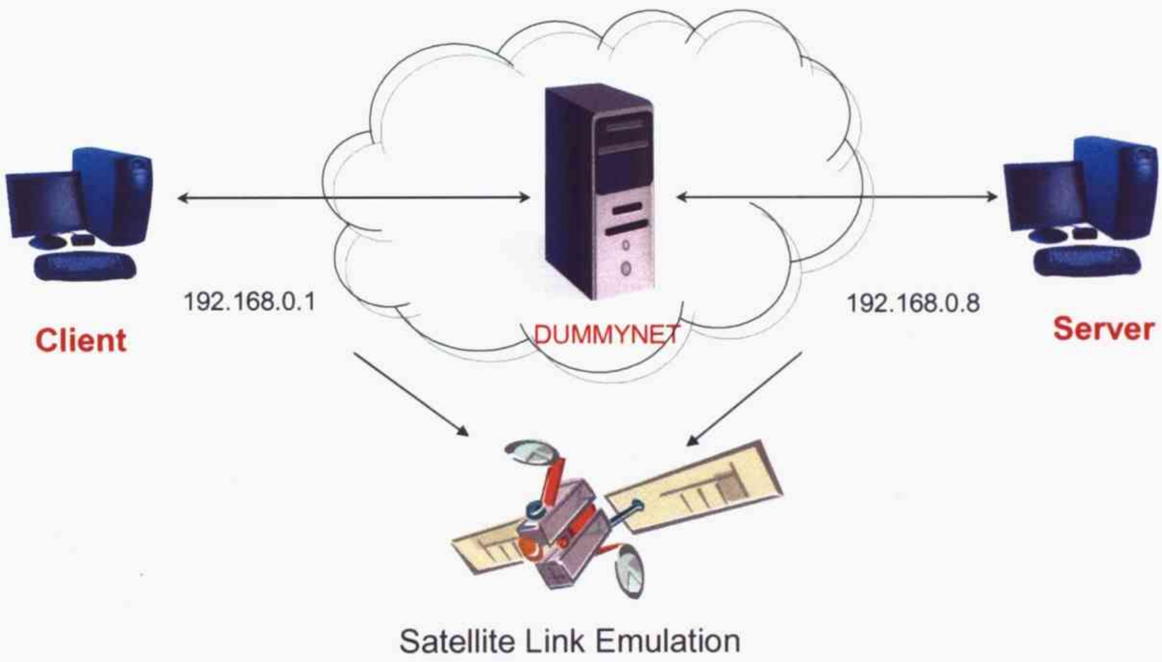

Figure 5.3 Client – Server Testbed Topology

Below are the lists of the parameters and the values involved in the experiments.

1. Section packing threshold → I choose threshold values of 0 (padding mode), 50 ms, 100 ms, 200 ms and infinity (section packing mode)
2. Link Latency → To emulate satellite link I choose 250 ms one way latency.
3. Link bandwidth is equal to 12 Mbps

Below is the list the steps involved in conducting the client-server connection experiment for ULE encapsulation simulation.

1. Decide what kind of TCP connection to be evaluated
2. Decide which file to be fetched from the server
3. Setup server to be able answering request from the client
4. Write shell script for request automation
5. Set the DUMMYNET attribute like bandwidth shaping and one way latency
6. Connect client and server through DUMMYNET
7. Run the experiment
8. Dump traffic data at client side and server side.
9. Run analysis tool to the traffic data to obtain evaluation results for each condition.
10. Post process the data for visualization

Client-server connection experiment is designed to be reproducible. Client fetches the same set of files 2 times from the server. Below is the list of the file fetched by the client.

1. Apache Ver. 2.0.51 manual page manual/install.html with all the files which necessary to display the page
2. Apache Ver. 2.0.51 manual page manual/invoking.html with all the files which necessary to display the page
3. Apache Ver. 2.0.51 manual page manual/stopping.html with all the files which are necessary to display the page
4. Apache Ver. 2.0.51 manual page manual/bind.html with all the files which are necessary to display the page
5. Apache Ver. 2.0.51 manual page manual/sections.html with all the files which are necessary to display the page
6. 1249 KB PDF file
7. 545 KB PDF file
8. 572 KB zip file

Data set 6 to 8 creates a heavy load connection with high packet rate. Packet 1 to 5 is consists of multiple small files and the packet rate during this time is lower.

Table 5.1 shows the environment and the specification for every computer utilized in the experiment. All kernel settings are set to default.

Table 5.1: Machine specification for client – server simulation

Machine	Client	Server	Dummynet
OS	Mandrake 10.1	Fedora Core 2	FreeBsd
CPU (Mhz)	Pentium 3 750 MHz	Pentium 3 850 MHz	Pentium 4 2.53GHz
Memory (MB)	256 MB	256 MB	1GB

5.2 Emulation

Simulation can only estimate the effectiveness of section packing mode with threshold setting. But in the real world, threshold values could determine the characteristic of the connection. Queuing delay and processing time might influence the outcome. Furthermore the encapsulator and decapsulator are not perfect and have their own capacity to handle the packets.

This factor can not be represented by simulation only. In addition, the influence of section packing mode and Delay Differences, can not be evaluated in the simulation. Delay Differences is a very important network property and particularly crucial in time sensitive application like VoIP.

To have the accurate evaluation of the section packing mode with threshold setting in satellite link, we have to perform the ULE encapsulation and the fragmentation into MPEG-2 TS cells. MPEG-2 TS cells then have to be transmitted via a satellite link.

Figure 5.4 shows the basic setup to transport IP over satellite link. A DVB-S modulator is needed to modulate MPEG-2 TS cell into the satellite wave using DVB-S standard for the decapsulator.

Dedicating a channel for the experiment appears to be very expensive. But the experiment needs an environment where at least satellite latency, bandwidth and the encapsulation process could be emulated. Therefore we proposed an architecture for ULE emulator. The emulator will emulate the behavior of the ULE encapsulation and also SNDU fragmentation into TS cells. Replacing the DVB modulation, MPEG-2 TS cell is transported over User Datagram Protocol (UDP) to the decapsulator. We utilize Ethernet as the link layer technology for the emulation. Figure 5.5 shows the methodology of the emulator.

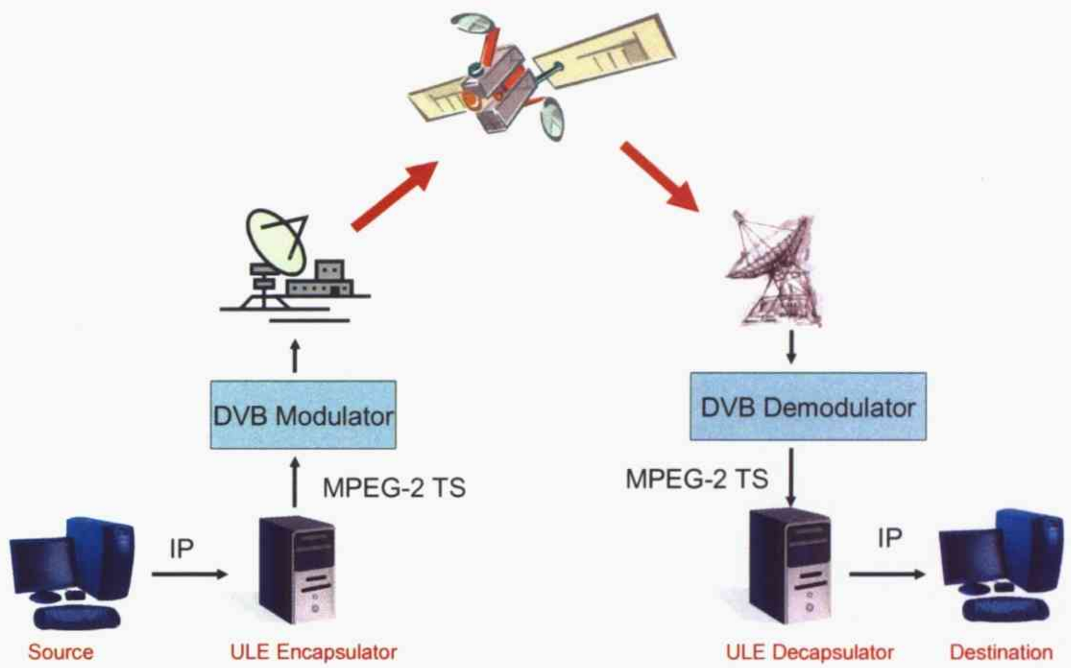

Figure 5.4: Setup for IP/ULE/DVB over satellite

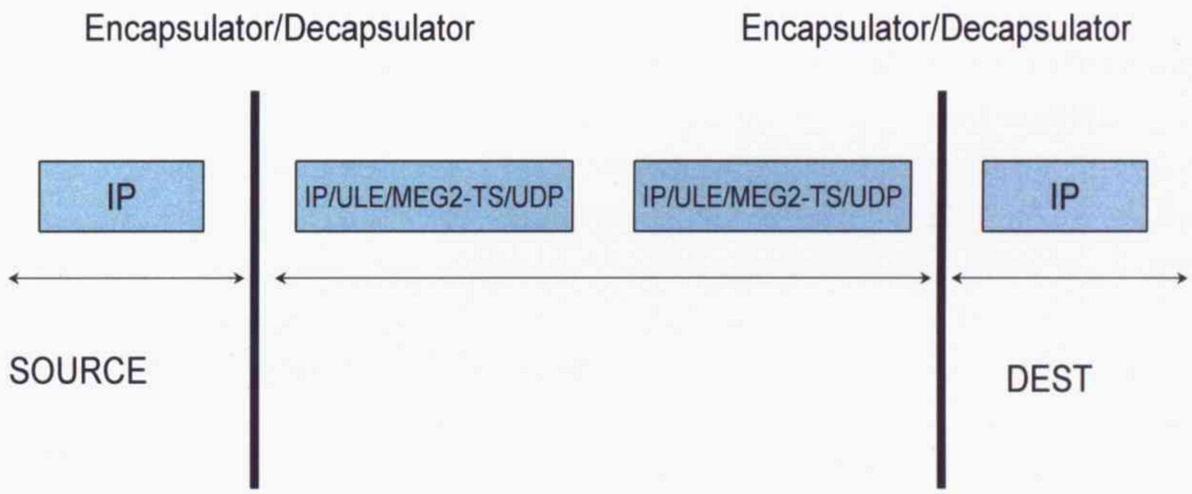

Figure 5.5: Emulator Methodology

Encapsulator acts as the gateway for IP packet. The steps involved in the emulation are described below.

1. Destination IP header is set to receiver IP address.
2. Source sends the IP packet over Ethernet frame to Encapsulator
3. Encapsulator receives Ethernet frames, and extracts IP packets.
4. Encapsulator performs ULE encapsulation to IP packet to make SNDU.
5. Encapsulator builds MPEG-2 TS cells and fragment SNDU into the cell.

6. Encapsulator send completed MPEG-2 TS cells using UDP protocol addressed to decapsulator.
7. Decapsulator receives UDP packets and extracts TS cells.
8. TS cells are defragmented into SNDUs.
9. Decapsulator decapsulates SNDUs to IP packets.
10. Decapsulator transmits IP packets using Ethernet frame to the destination.
11. Destination receives the IP packet.

5.2.1 Implementation of ULE encapsulator

This section describes the implementation of the ULE encapsulator used for the emulation experiment. The encapsulator is implemented using multi-threaded approach. Main process is responsible for receiving IP packets and does ULE encapsulation to produce SNDUS. The SNDUs then is stored in a circular buffer.

Another thread will fetch SNDU from the buffer, do the fragmentation and send MPEG-2 TS cells using UDP packets. Multi-threaded approach will enable the encapsulator to work independently to receive IP packets and to process the MPEG-2 TS cells. If fragmentation process requires more processing time, the encapsulator will still be able to receive IP packets from the source and perform ULE encapsulation.

Encapsulator is equipped with 2 network interfaces. An interface will receive IP packets from the source. Another interface is used to send MPEG-2 TS cell to the decapsulator. The encapsulator could also be considered as a bridge connecting source to the destination.

ULE encapsulator is implemented in user space. Encapsulator receives IP packet from the source by using SOCK_RAW socket API and sends MPEG-2 TS cells using SOCK_DGRAM socket API. Figure 5.6 shows software architecture for ULE encapsulator and program 5.1 shows the data structure involved in implementing ULE encapsulator.

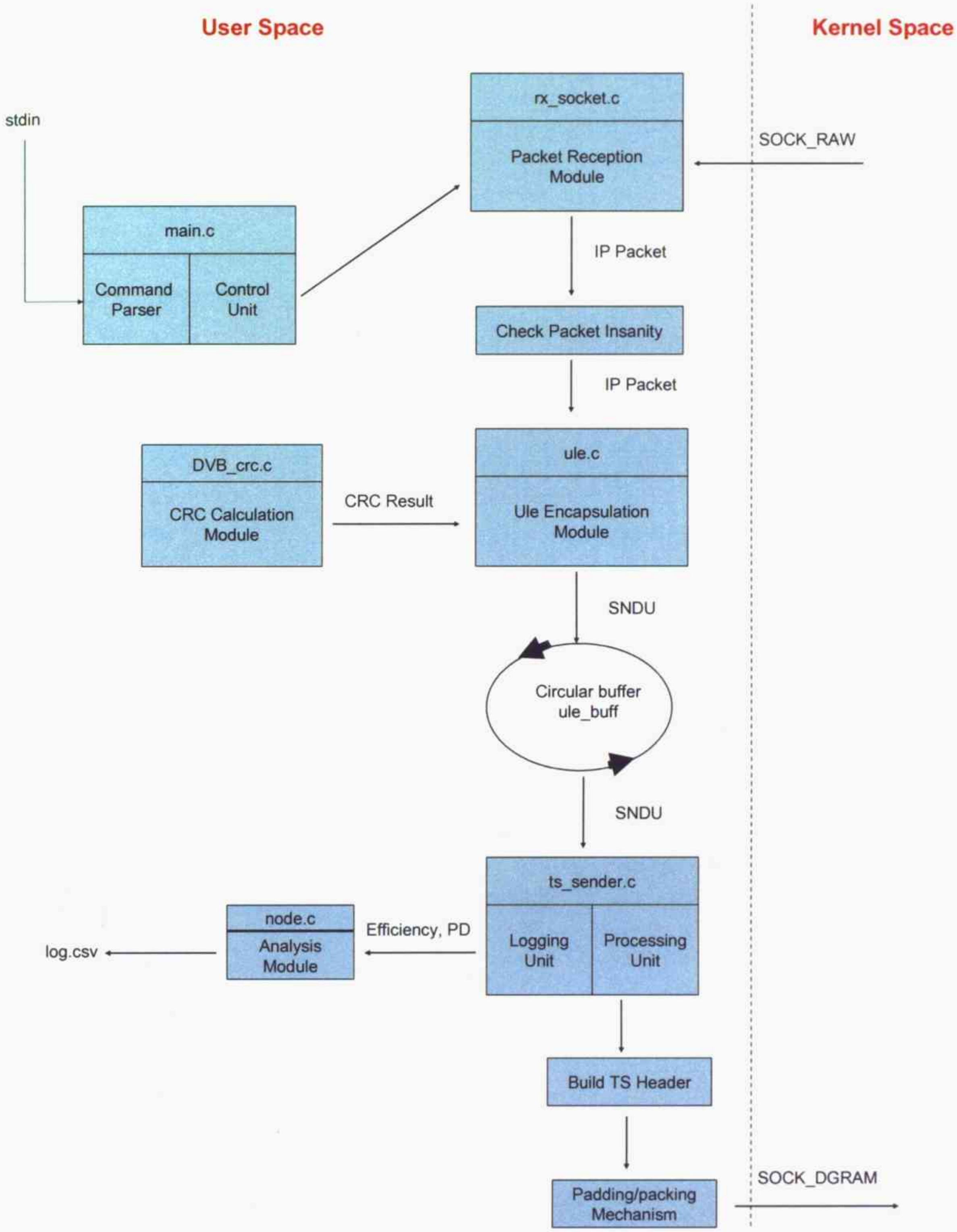

Figure 5.6: ULE encapsulator's software architecture

```
typdef struct _ule_hdr {
        u_short Dlength;
#define D_absent_mask 0x8000
#define Length_mask 0x7FFF
        u_short Type;
}

typedef struct _ts_header {
        char syn;
        char PID_hi : 5;            //Upper 5 bit from 13 bit PID
        char trans_prior : 1;       //Transport priority flags
        char P_U_S_I :     1;       //PUSI flags
        char trans_error : 1;       //Transport error indicator
        char PID_lo;                //Lower 8bit of 13 bit PID
        char CC : 4;                //Continuity counter field
        char AFC : 2;               //Adaptation field control
        char scramble : 2;          //Scrambling control field
} ts_header;
//ULE circular buffer
typedef struct _buff {
        struct timeval ule_tv;              //Arrival time
        int ule_length;                     //SNDU total length
        unsigned char ule_pack [1600];      //SNDU
} ule_buff;
```

Program 5.1 Data structure in ULE encapsulator

5.2.2 Implementation of ULE decapsulator

This section describes ULE decapsulator implementation. Decapsulator is also equipped with 2 network interfaces. An interface will receive UDP packet from the encapsulator. Another interface is used to send the bridged IP packet to the destination.

ULE decapsulator is implemented in user space. It receives UDP packets from encapsulator by using SOCK_DGRAM socket API. Then the decapsulator processes MPEG-2 TS cell and reconstruct the SNDU. The SNDU will be verified using CRC calculation and the decapsulated before being sent to destination through SOCK_RAW socket API.

Implementation of padding mode and section packing mode do not influence the architecture of decapsulator. Decapsulator is designed to be able to handle any incoming MPEG-2 TS cell, whether it is packed or padded. No data buffering is required in ULE decapsulator since the reconstructed IP packet will be sent immediately. Data structure for ULE header and MPEG-2 TS header is shown in program. Figure 5.7 shows the software architecture for ULE decapsulator.

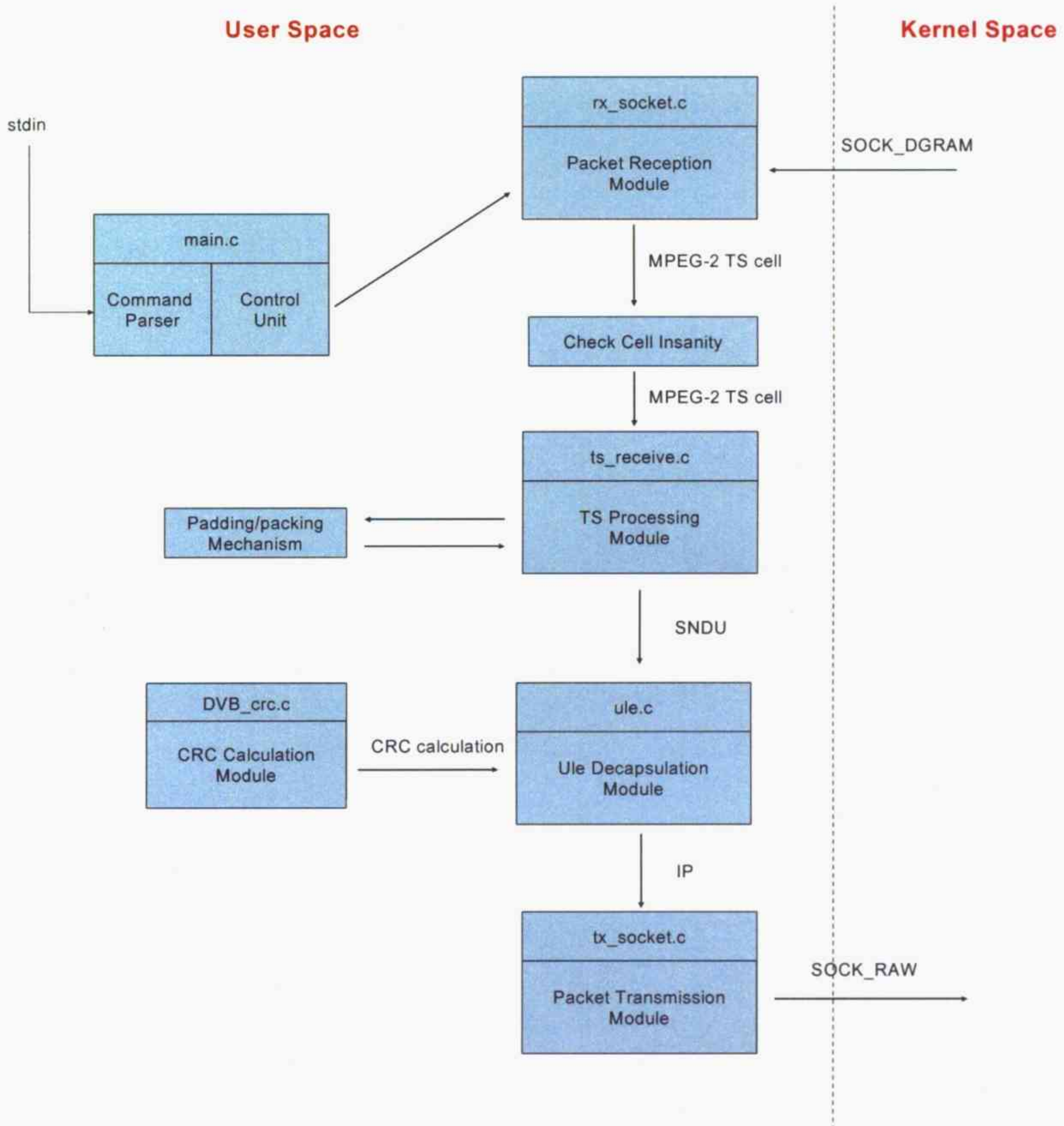

Figure 5.7: ULE decapsulator's software architecture

5.2.3 Experiment Setup

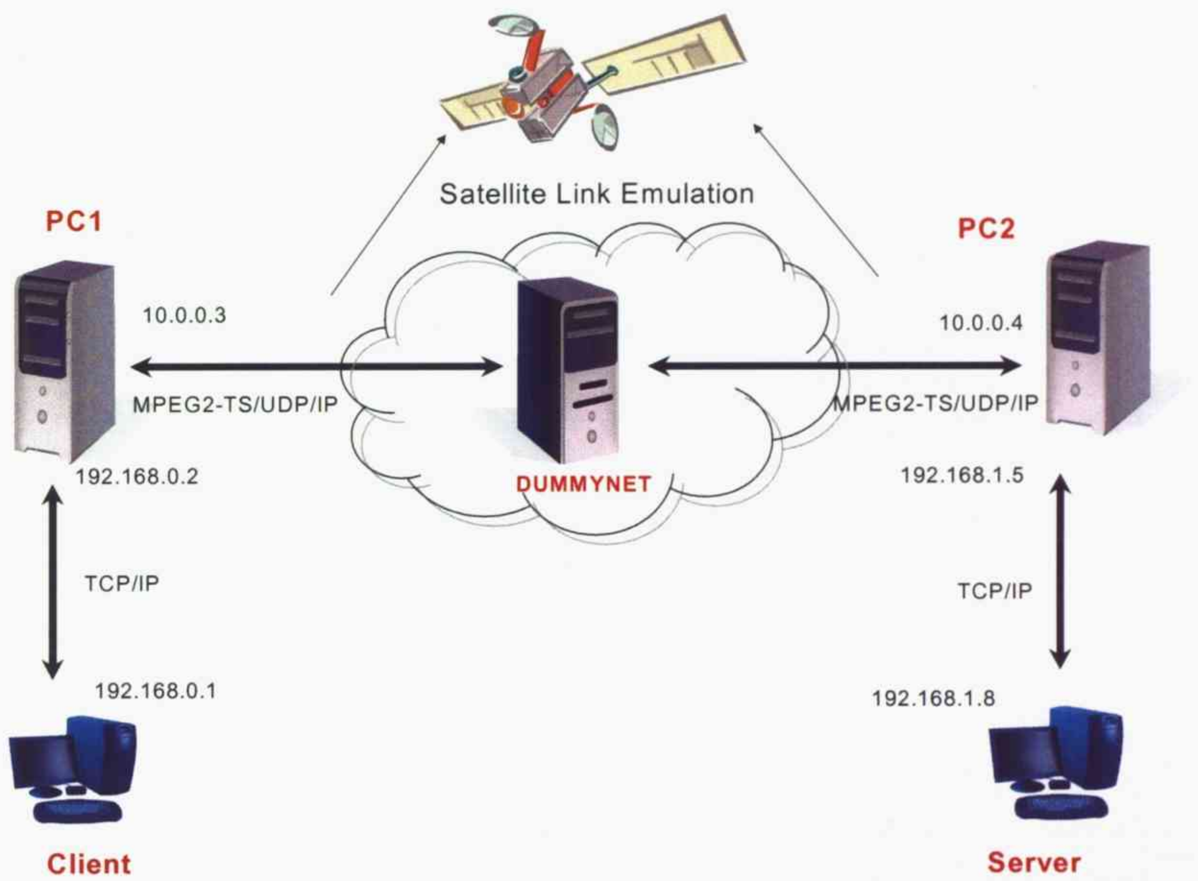

Figure 5.8: Experiment setup for ULE emulation.

Figure 5.8 shows the experiment setup for the experiment. PC1 and PC2 act as default gateway for client and the server. ULE encapsulator is installed in PC1 and PC2. Dummynet is used to emulate satellite link latency and bandwidth. Below is the list of steps involves in conducting experiment for client-server connection through ULE emulator

1. Set the Dummynet attribute like bandwidth shaping and one way latency
2. Connect client to PC1 and server to PC2
3. Connect PC1 and PC2 through Dummynet
4. Use same shell script in simulation experiment for request automation
5. Set the threshold parameter (0 for padding) to PC1 and PC2.
6. Run the experiment
7. Dump traffic data at client side and server side.
8. Log encapsulation efficiency and packing delay at PC1 and PC2
9. Repeat steps 5-8 for every threshold values.

10. Post process the data for visualization.

Below is the list of the parameters and the values involved in the experiments.

1 Section packing threshold. I choose threshold values of 0 (padding mode), 50 ms, 100 ms, 200 ms and infinity (section packing mode).
2 Link Latency. To emulate satellite link I choose 250 ms one way latency.
3 Link bandwidth is equal to 12Mbps
4 Encapsulator circular buffer is set to 100,000 packet

Table 5.2 shows the specification and the environment in every machine used in the experiment. All kernel settings are set to default

Table 5.2: Machine specification in emulation experiment

Machine	Client	Server	PC1	PC2	Dummynet
OS	Mandrake Linux 10.1	Fedorra Core 2	Redhat	Redhat	FreeBsd
CPU	Pentium3 750 MHz	Pentium3 850 MHz	Pentium3 870 MHz	Pentium3 870 MHz	Pentium4 2.53 GHz
Memory	256 MB	256 MB	256 MB	256 MB	1 GB

Emulation for network traffic is not being done in the experiment. This is because I insist to emulate real communication traffic. Replaying traffic data through the emulator according to the time stamp can not accurately reproduce the traffic characteristic since different threshold values may have different impact on each connection or traffic flow. Therefore in emulation experiment, we only conduct client – server analysis.

Delay Differences characteristic is one of the important things in network performance. Threshold values and its impact to Delay Differences performance are not able to be done in simulation experiment. In emulation experiment we also conduct Delay Differences analysis and discuss the influence of threshold values to Delay Differences characteristic.

CHAPTER 6

Results and Discussions

This chapter shows the results of the experiments and the discussions. Section 6.1 summarizes results for simulation and Section 6.2 discussed about emulation results.

6.1 Simulation

This section shows the results for simulation experiments and the discussion. Every subsection will shows all the results in the first part followed by discussion at the end of the subsection.

6.1.1 Network Scenario

6.1.1.1 Results

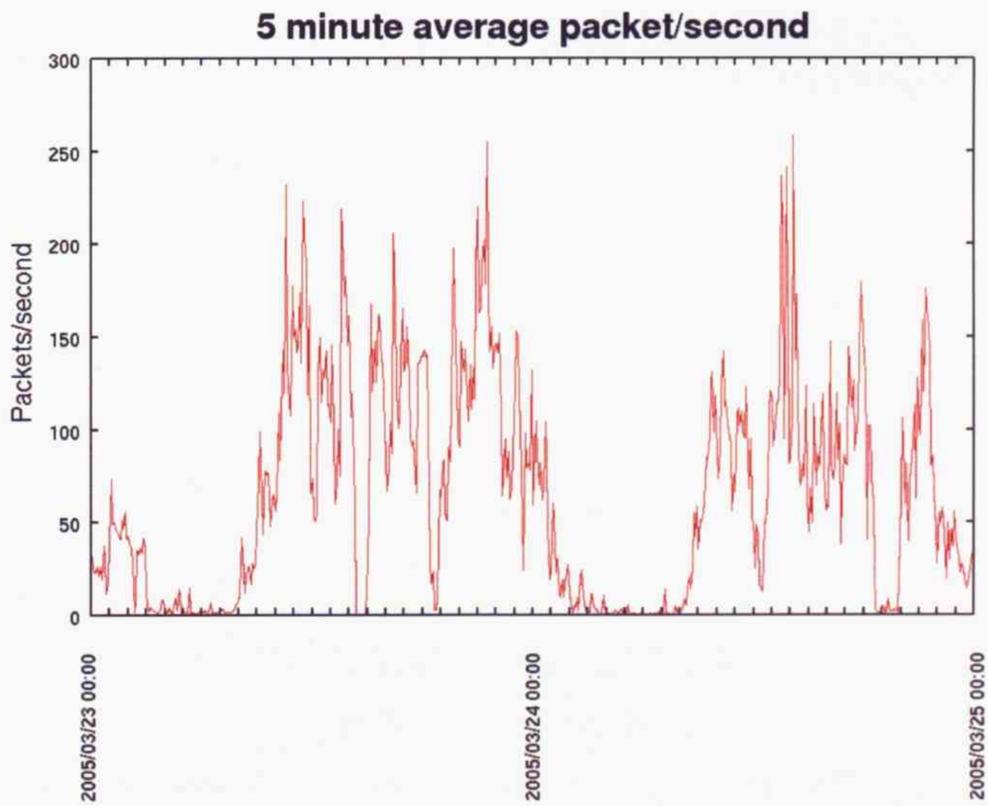

Figure 6.1: Traffic volume for 48 hours on DVB-RCS link – packet/second

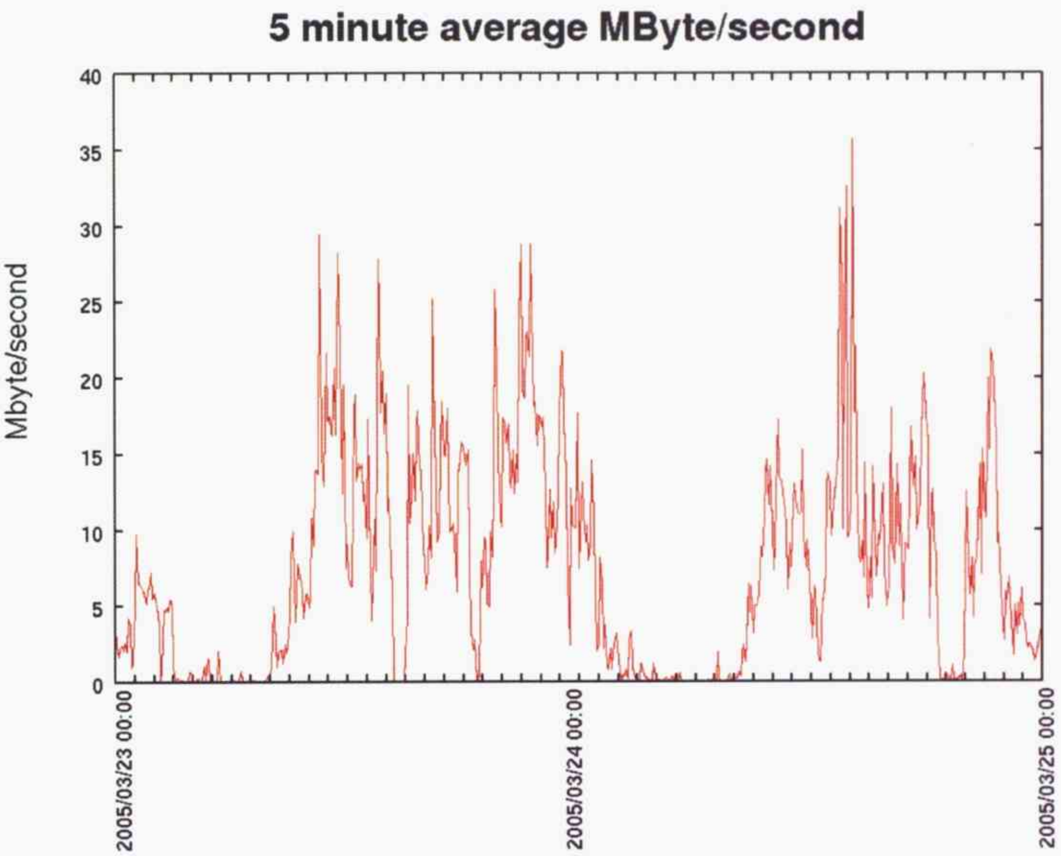

Figure 6.2: Trafic volume for 48 hours on DVB-RCS link – Mbyte/second

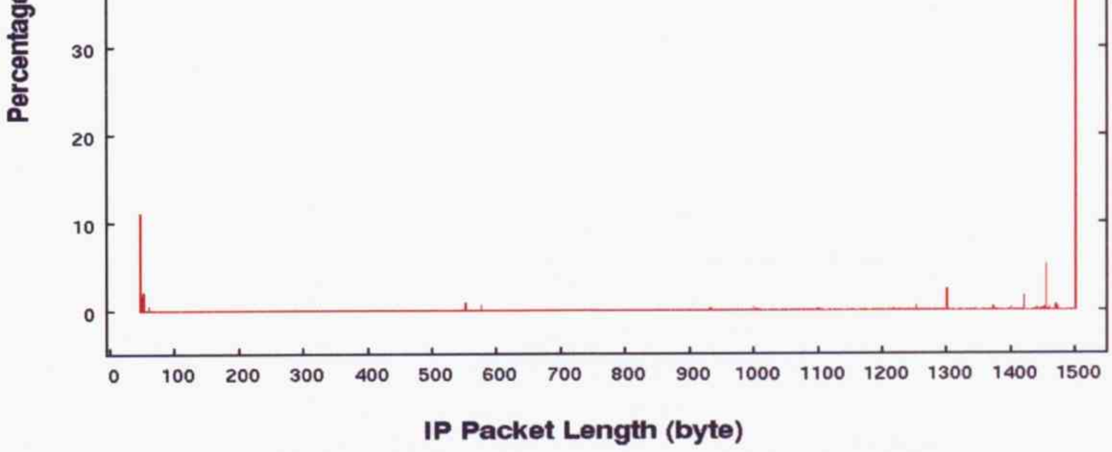

Figure 6.3: IP packet Length Distribution 2005/03/23

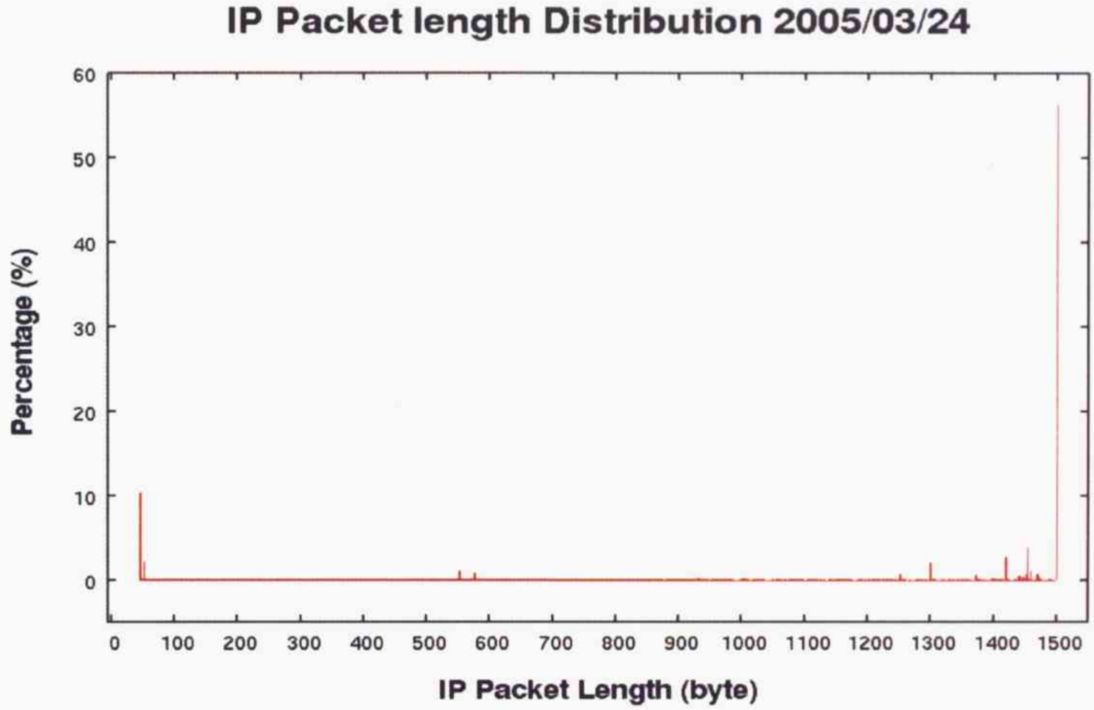

Figure 6.4: IP packet Length Distribution 2005/03/24

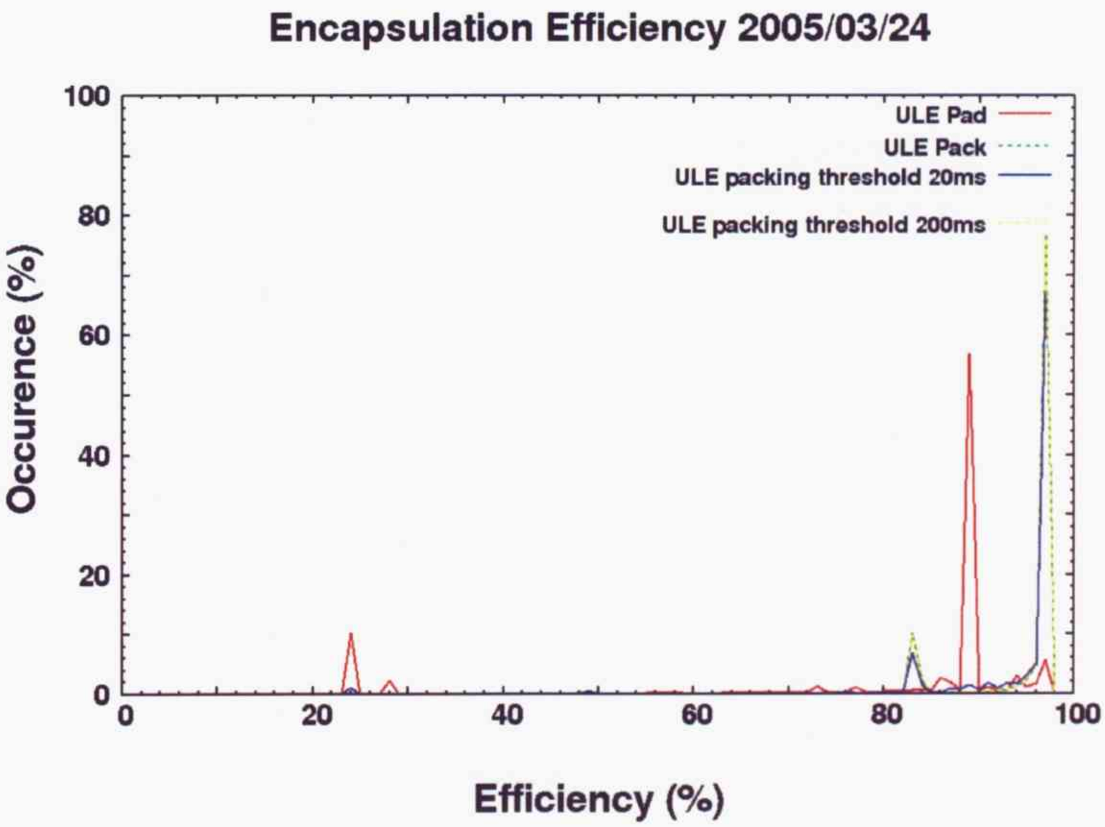

Figure 6.5: Encapsulation Efficiency Distribution

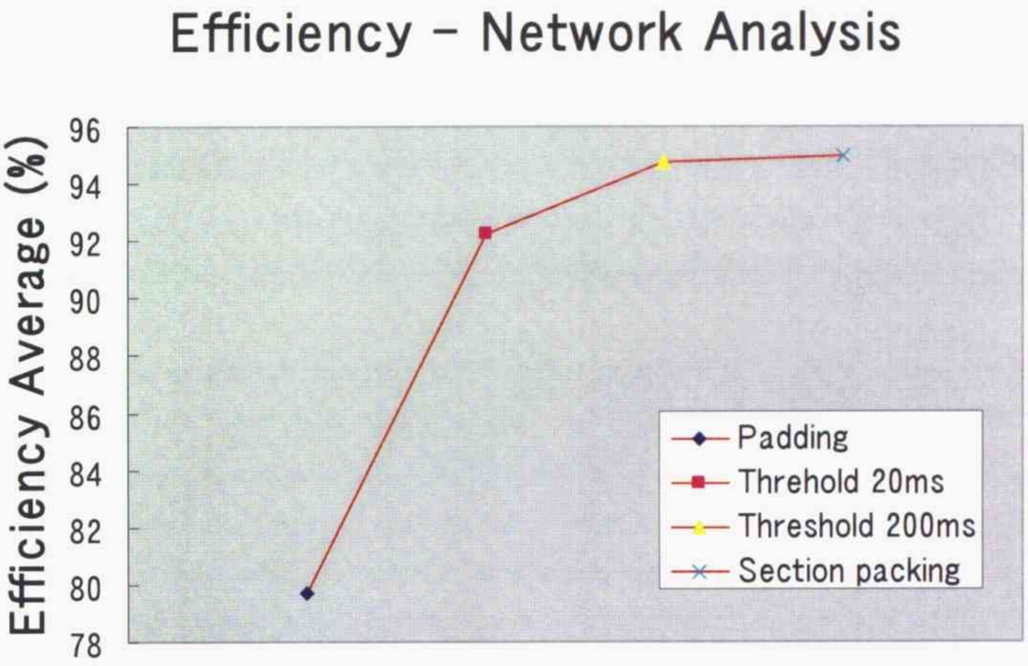

Figure 6.6: Average efficiency values

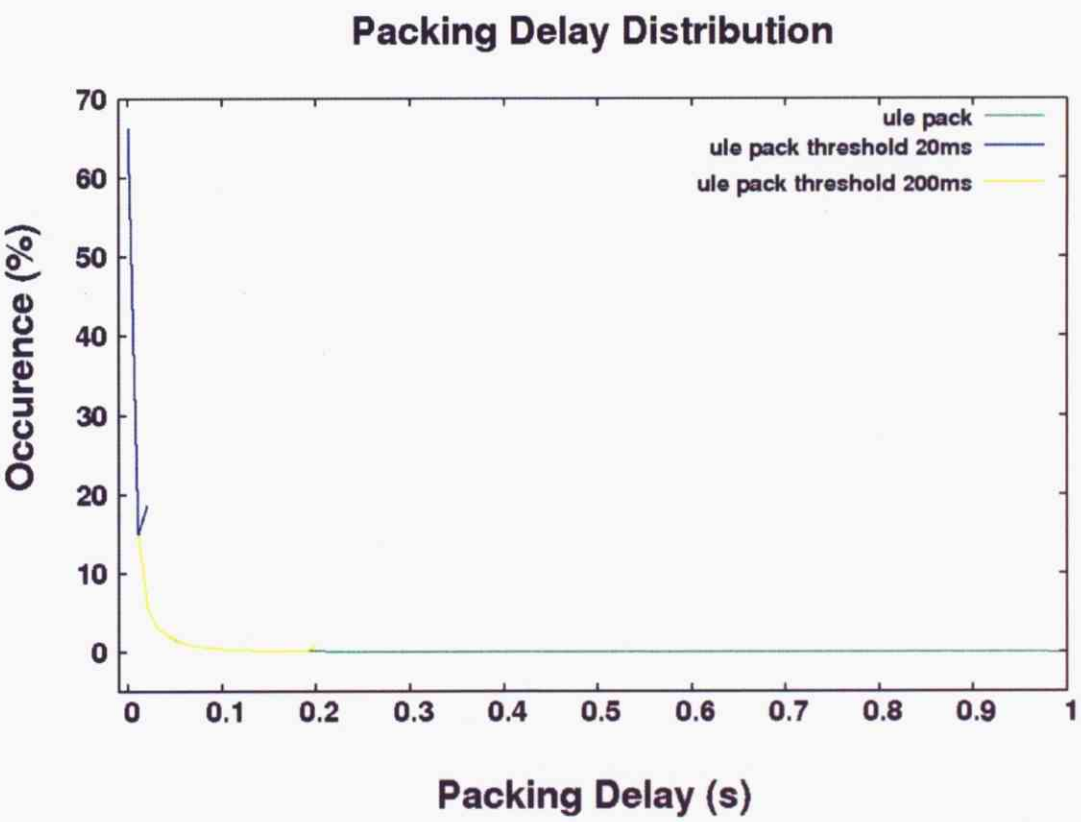

Figure 6.7: Packing Delay Distribution

6.1.1.2: Discussion

During the observation on the traffic from the DVB-RCS link, a total of 11,087,324 IP packets have been recorded in 48 hours. Figure 6.1 shows the 5 minutes average of packet per second. The packet numbers reach the peak during noon onwards when every participant was having their discussion. The volumes then decreased to nearly zero late at night. The pattern repeats itself every 24 hours. Similar tendency can also be seen in Figure 6.2.

On rare occasions we can se the packet rate drop to 0 packets per second when the DVB-RCS link had problems and lost the connection. The same pattern can be observed in figure which shows the traffic pattern by bytes/second.

Figure 6.3 and figure 6.4 show the IP packet length distribution every 24 hours. The traffic pattern for each day is comparable and verifies that the DVB-RCS link usage pattern is indeed similar every day during the conference. The left portion of the x-axis shows the proportion of small packets mainly contain of TCP ACKs and other TCP control packets. In contrast, the maximum packet length is 1500 bytes. This is the maximum Ethernet frame size and therefore a common Maximum Transmission Unit (MTU) of most current link technology. From the graph it is obvious that most of the traffic is dominated by 2 packet length values; the TCP Ack Packet and the MTU.

The low percentage of small ACK packet is typical for DVB satellite inbounds links which are asymmetric in nature. Nevertheless the traffic shows the mixture of multiple clients-server traffic by having 2 obvious peaks that represent the characteristic of a client and server traffic patterns. This character is essential to show how ULE encapsulation with section packing threshold performs in a moderately distributed packet length as oppose to the extreme distribution of client and server traffic.

We decided to do efficiency analysis only for 24 hours traffic on 24th March 2005 since the traffic pattern is obviously similar every 24 hours. For the analysis we developed a spreadsheet calculation tool which is able to evaluate packing delay for each packet. In the analysis, we assume that processing time and queuing delay is equal to 0. PD for padding mode in the analysis is also assumed to be 0.

For the sake of evaluation, we choose a random section packing threshold time of 20 ms and 200 ms to show the effectiveness of our proposal for different threshold values as compare to section packing mode without threshold and padding mode for ULE encapsulation.

Figure 6.5 illustrates the comparison of the ULE encapsulation efficiency for padding mode, section packing mode, section packing with 20ms threshold and section packing with 200ms threshold. Meanwhile figure 6.6 shows the efficiency average for each sample.

The results shows that although section packing is limited by the threshold, we can still observe the efficiency gain when compare to padding mode and comparable to section packing mode.

In average threshold setting of 20 ms improved 12% efficiency from padding mode and 3% less that section packing mode without threshold. Meanwhile threshold setting of 200 ms is 15% more efficient than padding mode and less than 1% different compare to section packing mode.

The efficiency results of the scenario show that values of threshold time are not giving much influence to the efficiency performances because of the small inter-arrival times between packets. Nevertheless section packing mode execution is still more efficient than padding mode.

Figure 6.7 shows the packing delay distribution of the analysis. With threshold setting, packing delay is limited to the threshold values. As the result, unlike section packing mode without threshold, a packet doesn't have to wait forever for incoming packet to be packet and transmitted.

Efficiency gain varies depending on the threshold times. Nevertheless threshold setting evidently increases transmission efficiency compare to padding mode and at the same time reduce the packing delay to values below than threshold, as shown in figure 6.7.

6.1.2 Client Scenario

6.1.2.1 Results

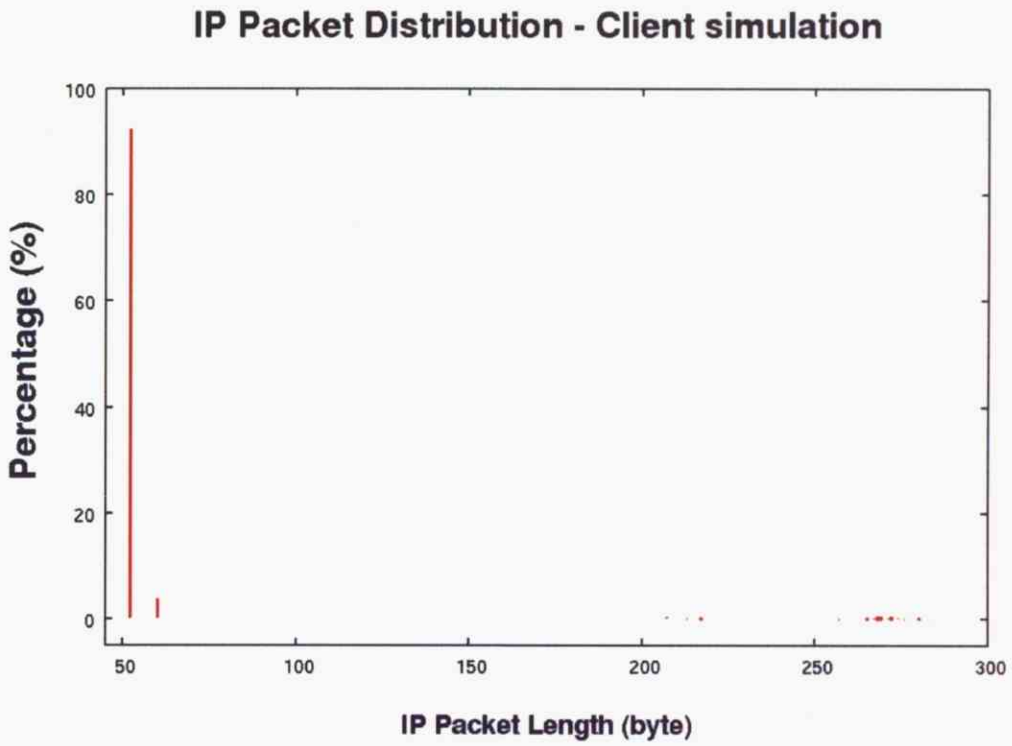

Figure 6.8: IP Packet Length Distribution – Client Simulation

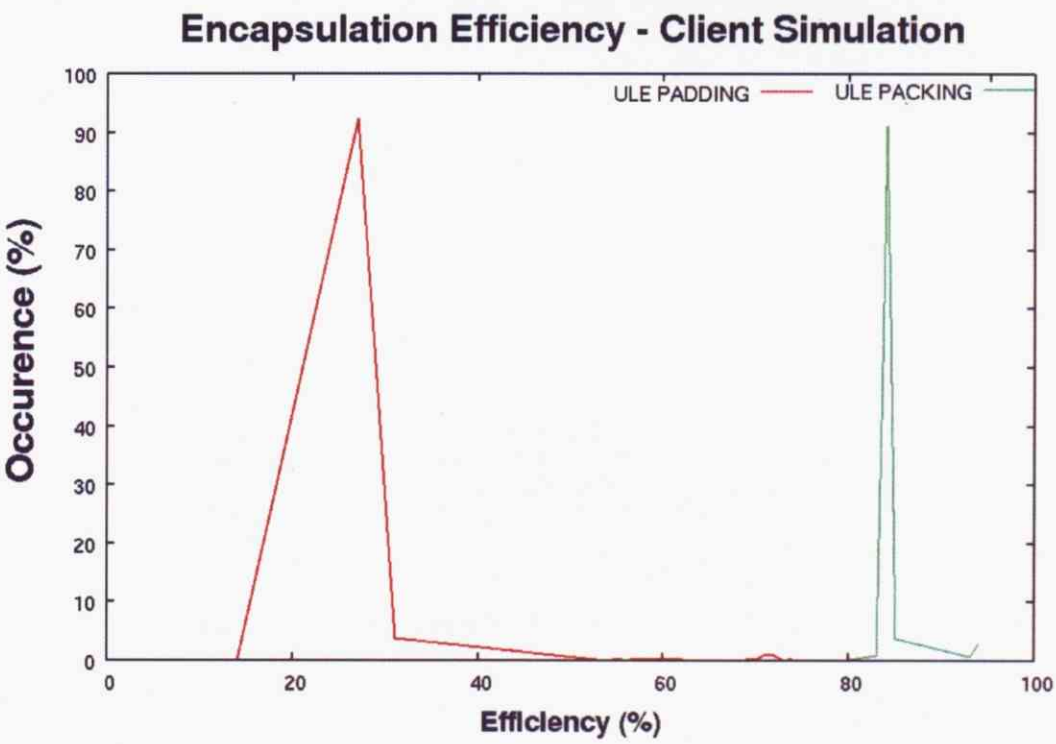

Figure 6.9: Efficiency distribution 1 – Client Simulation

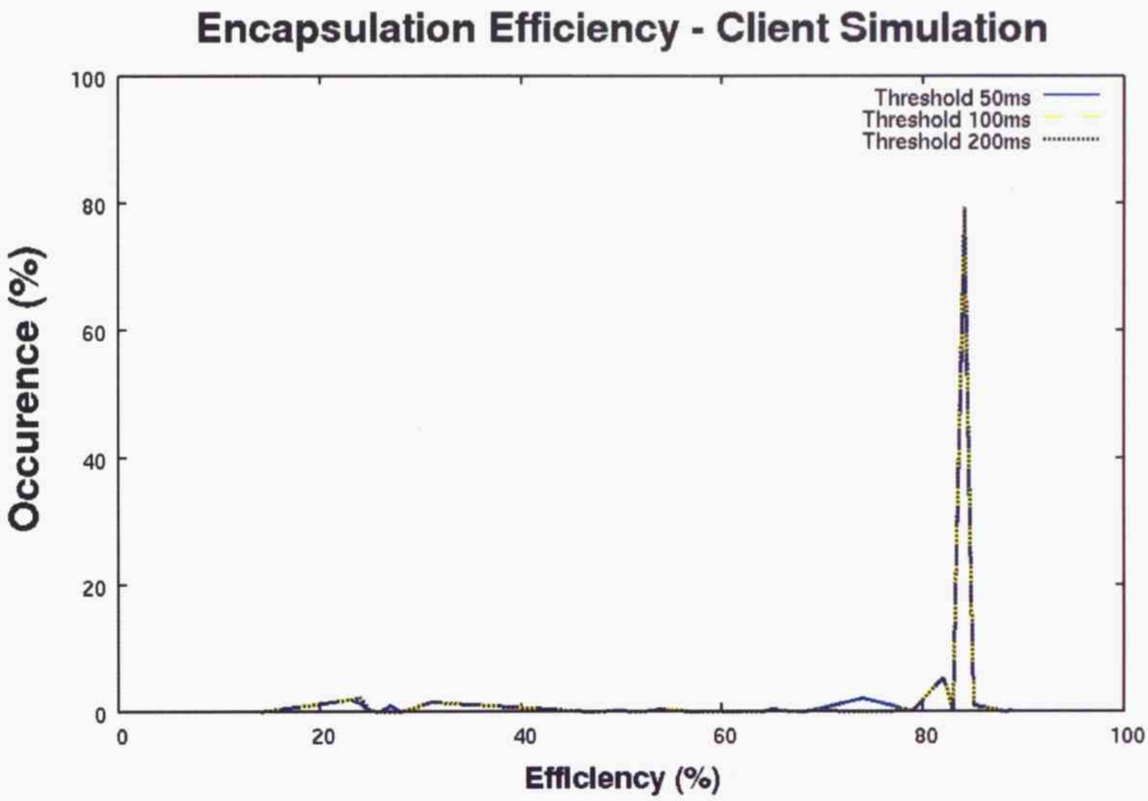

Figure 6.10: Efficiency Distribution 2 - Client Simulation

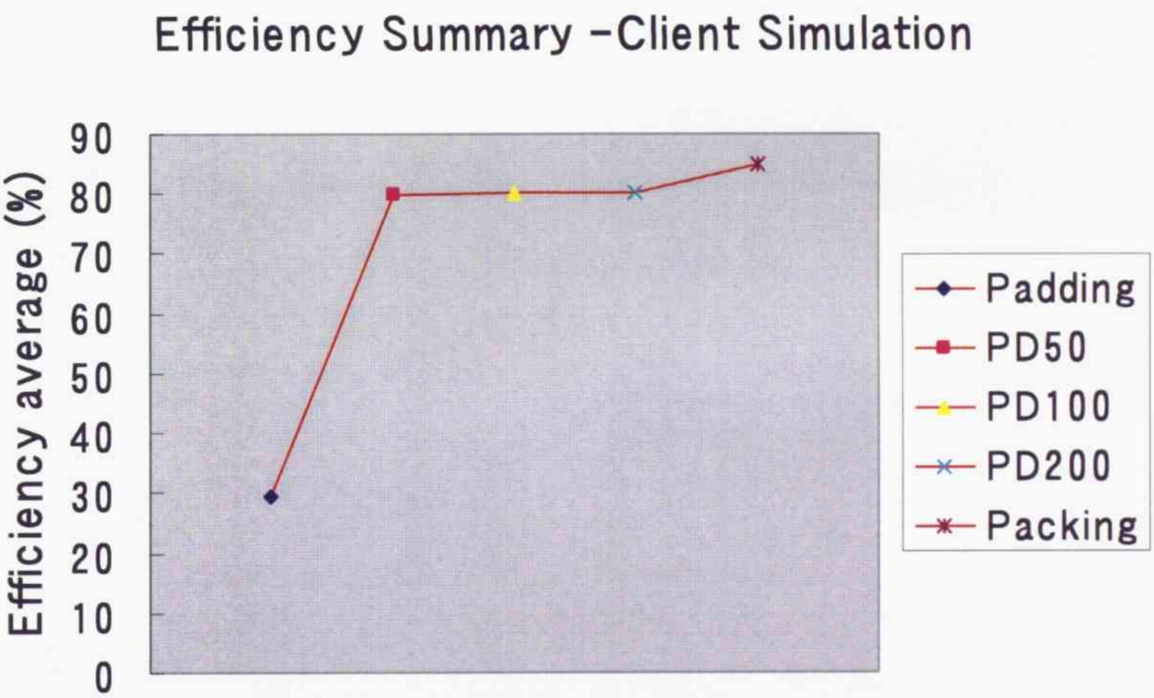

Figure 6.11: Efficiency Average – Client Simulation

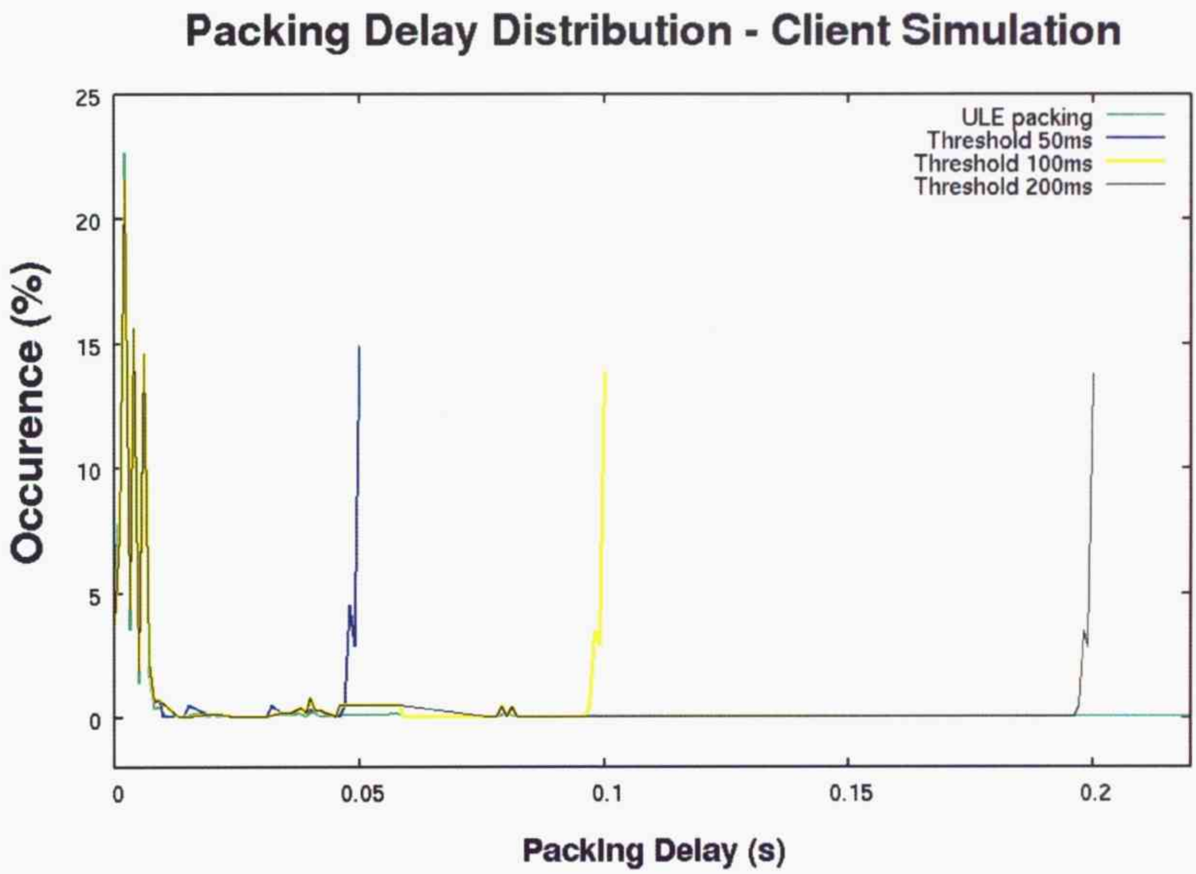

Figure 6.12: Packing Delay Distribution – Client Simulation

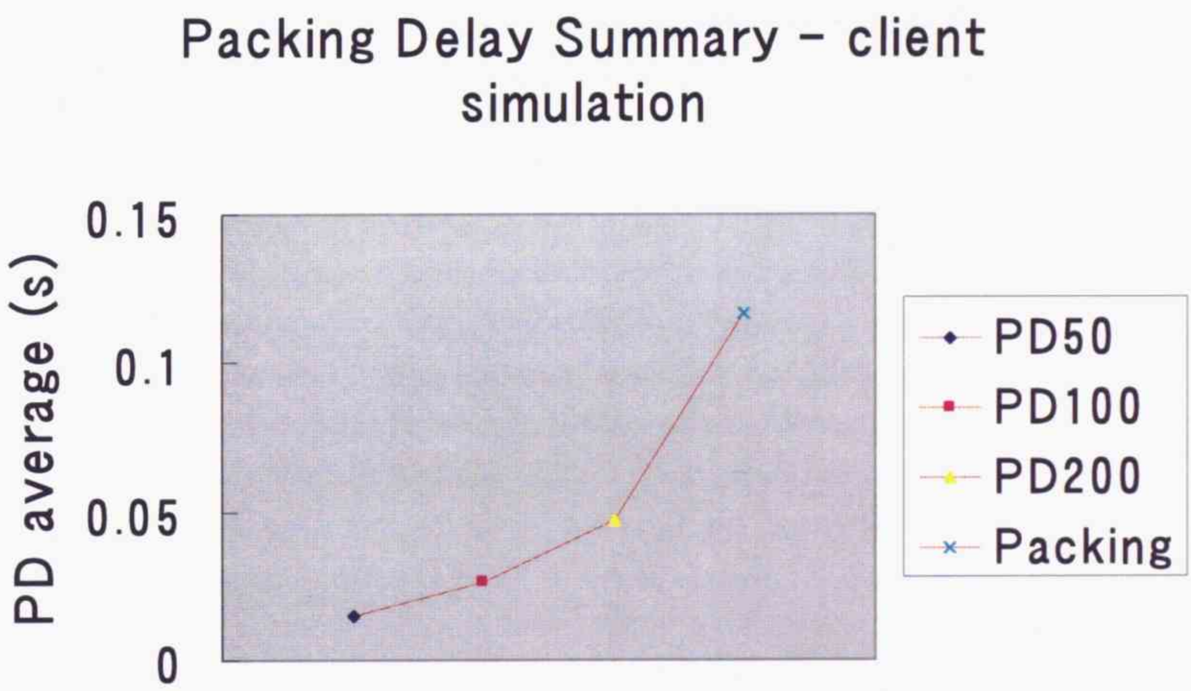

Figure 6.13: Packing Delay Average – Client Simulation

6.1.2.2 Discussion

Figure 6.8 shows the IP packet distribution. IP packet length in client traffic is extreme to the left with over 90% of the packets are 52 bytes TCP ACK packet. Beside ACK packets, client also generates a small number of other packet sizes. Since simulation is performed by making the client requests particular files from the server, the packet other than ACK packet comprise requests to the individual file name with relative path or information on user name and password.

Significant majority of small packets could deterrent ULE efficiency in padding mode. Furthermore, longer inter-arrival times between packets will cause packet to wait longer in section packing mode.

Client will have to wait data from server before sending ACK packet to the server after sending the request packet. Server will wait ACK packet from client to shift the TCP's sliding window. When the ACK packet from client is delayed by packing mechanism, TCP might mistakenly recognize it as congestion and reduce server's window size. The congestion control will affect the throughput of the flows

Setting threshold to section packing mode can diminish the issues. ACK packets from client don't have to wait forever and will be padded and sent immediately if no incoming packets exist until threshold time.

Figure 6.9 and figure 6.10 shows the efficiency distribution of client traffic. Reflecting the packet distribution, around 92% of the packets have 27% efficiency in padding mode. In contrary around 91% of the packets are 84% efficient.

Threshold setting of any threshold values is more than 50% efficient than padding mode in average and only 3% less efficient than unbounded section packing mode. Observation from figure 6.9 to 6.11 shows that efficiency values for each threshold setting is almost the same.

From packing delay distribution illustrated in figure 6.12, we can observe that nearly 15% of the packets have to be padded for each threshold setting. However since, one 188 bytes TS cell can contains multiple TS packets and padding bytes itself could be less, the efficiency is much better than padding one small packet alone in a TS cell. That is the reason why threshold setting scheme is comparable to section packing mode without threshold although 15% of the packets reach the threshold time. Naturally the average values of packing delay increase in consequence of threshold values as shown in figure 6.13.

6.1.3: Server traffic

6.1.3.1 Results

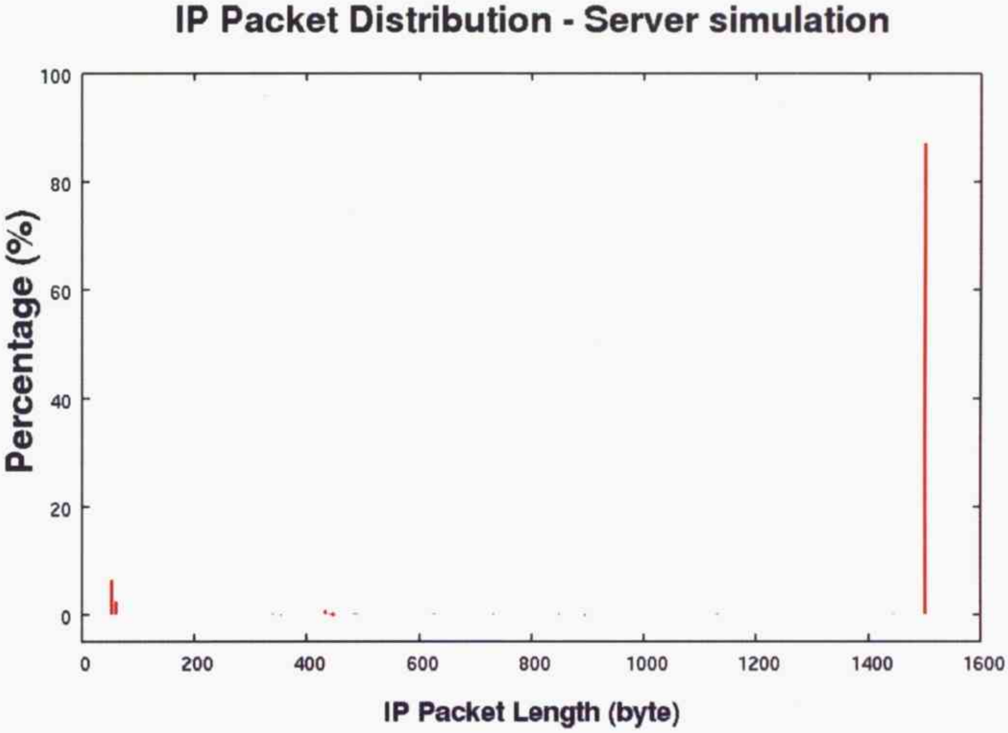

Figure 6.14: IP packet Distribution – Server simulation

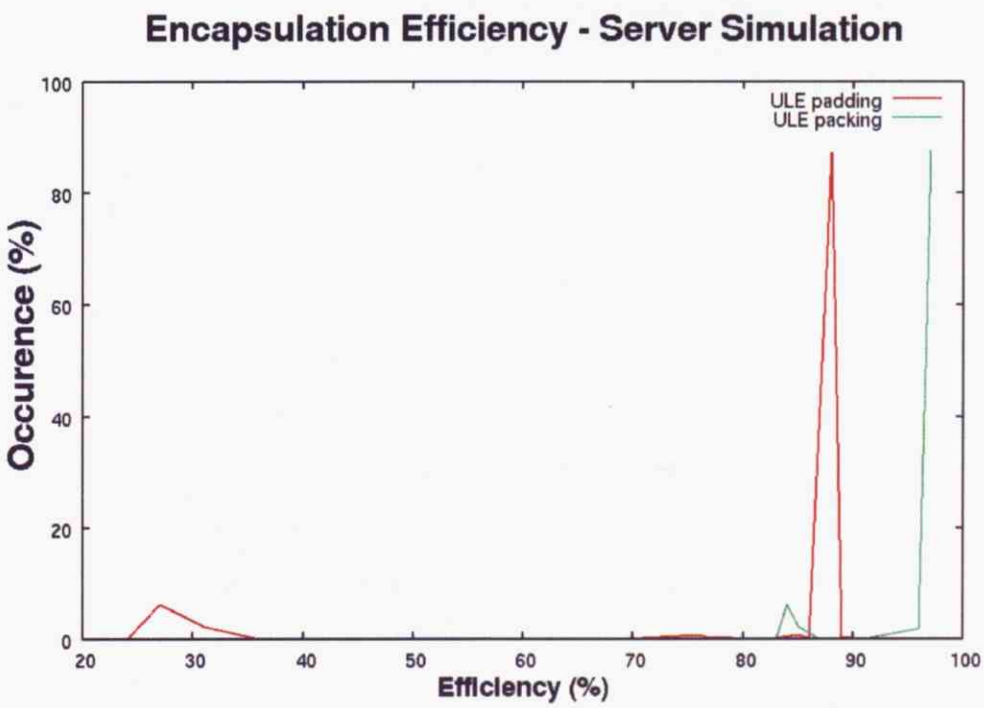

Figure 6.15: Efficiency Distribution 1 – Server Simulation

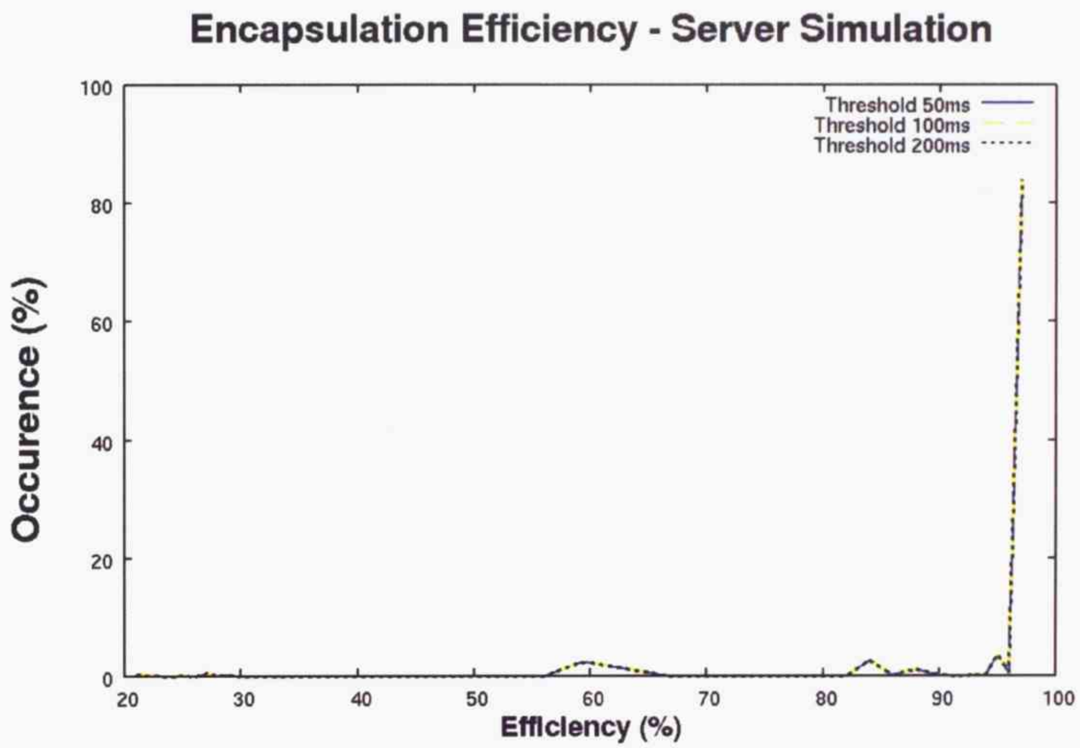

Figure 6.16: Efficiency Distribution 2 – Server Simulation

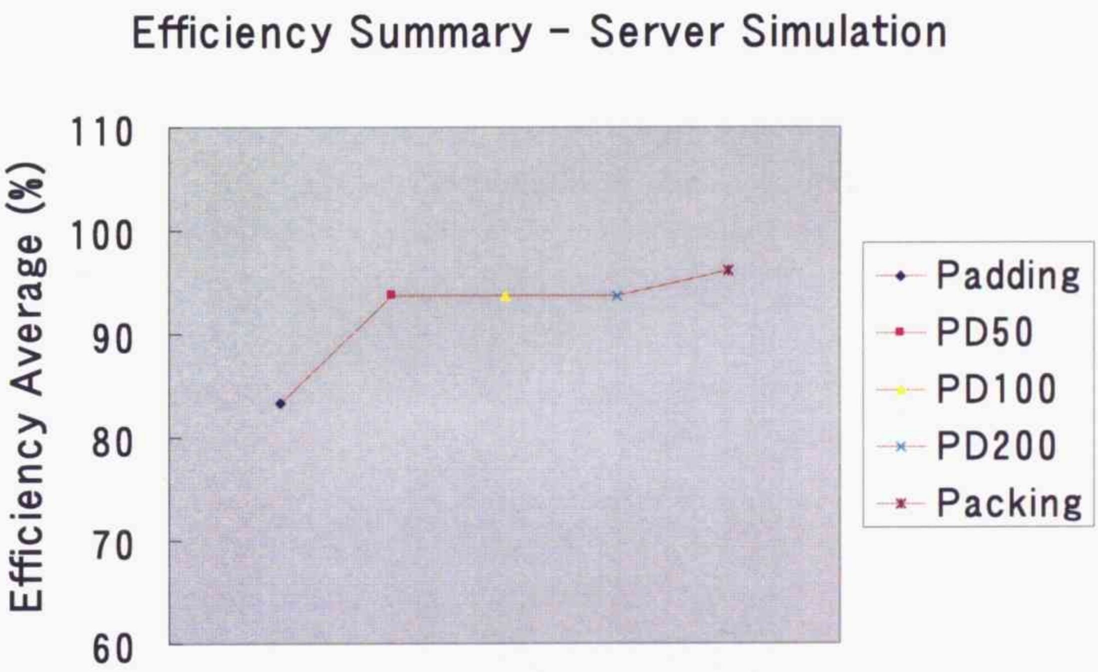

Figure 6.17: Efficiency Average – Server Simulation

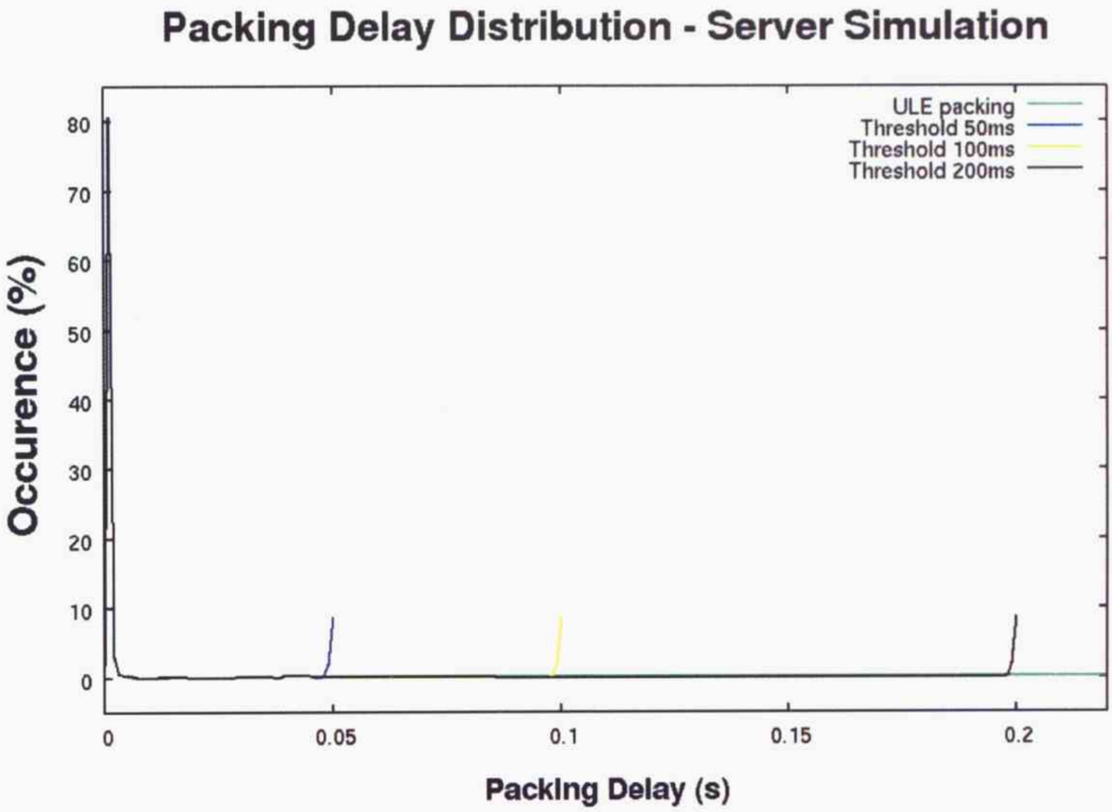

Figure 6.18: Packing Delay – Server Simulation

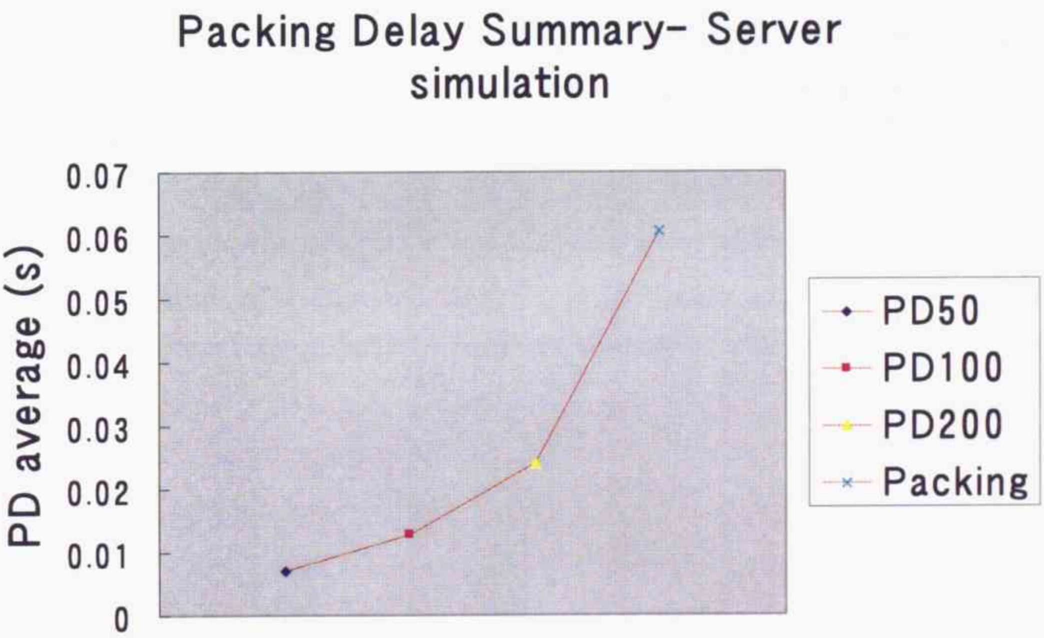

Figure 6.19: Packing Delay Summary – Server Simulation

6.1.3.2 Discussion

Figure 6.14 shows the packet length distribution of the server traffic. The histogram depicts that as oppose to client server, significant majority of packets from server have packet length equal to maximum frame size of Ethernet (MTU). We can also see a few percentages of small packets. These packets are TCP control packets (ACK, SYN, FIN) from the server.

Big IP packets mean better efficiency in padding mode. Compare to client traffic, padding mode in server traffic mark 83% in average and only around 10% less efficient than unlimited section packing mode.

Figure 6.15 to 6.17 figure show the efficiency performance for each mode. Small packet interval time in server traffic makes the efficiency results similar for different threshold. Interestingly similar to client traffic, average efficiency differences between unbounded section packing mode and section packing mode with threshold is only 3%. This result prove that in both client traffic:- the worst scenario for section packing where most of the packet is small and have higher packet inter-interval time, and server traffic:- the best condition for section packing with low packet interval time and big packets, difference between section packing mode and section packing mode with threshold is almost the same.

Figure 6.18 shows the packing delay distribution for server traffic and figure 6.19 shows the average values. Only 10% of the packet reach threshold time and have to be padded for each threshold setting because of the smaller inter-arrival time between packets from the server.

These evaluations shows that in server traffic, section packing mode with threshold setting is comparable to unbounded section packing mode and much efficient than padding mode. The packing delay for every packet also is limited to the threshold time and eliminates the possibility of the packet has to wait forever to be packed for the sake of efficiency.

6.2 Emulation

This section shows the results of emulation experiments and the discussion. Every subsection will shows all the results in the first part followed by discussion at the end of the subsection.

6.2.1 Client Scenario

6.2.1.1 Result

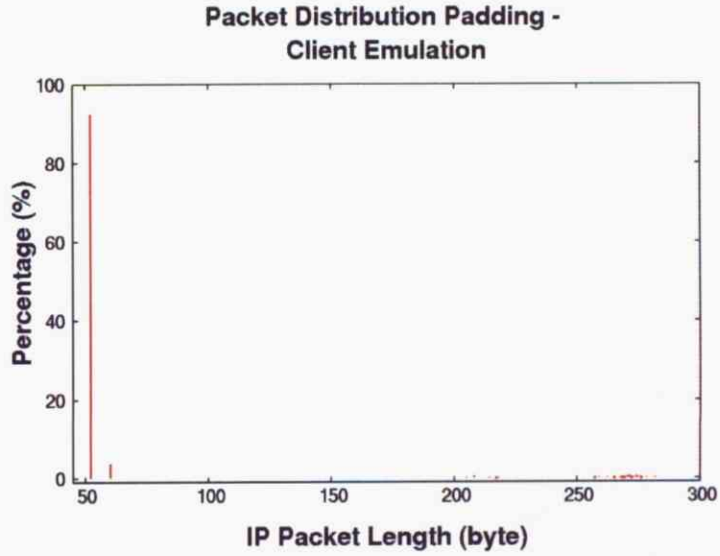

Figure 6.20: IP Packet Percentage Distribution – Client Padding Emulation

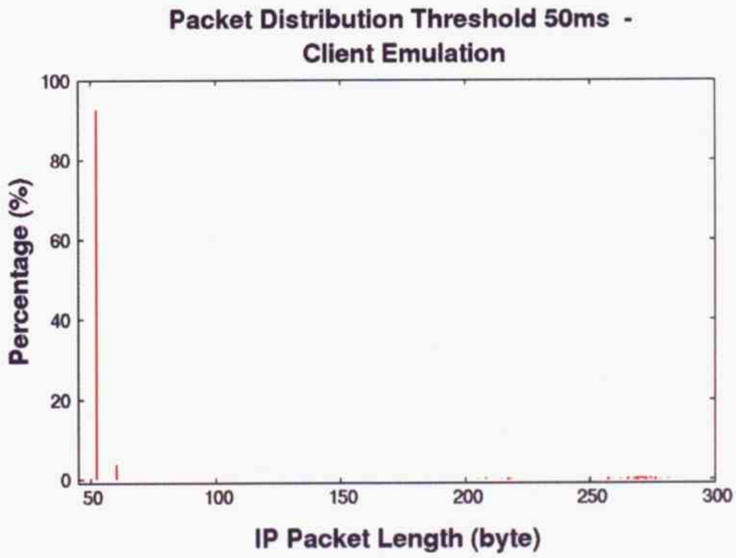

Figure 6.21: IP Packet Percentage Distribution – Client Section Packing Emulation with 50 ms Threshold

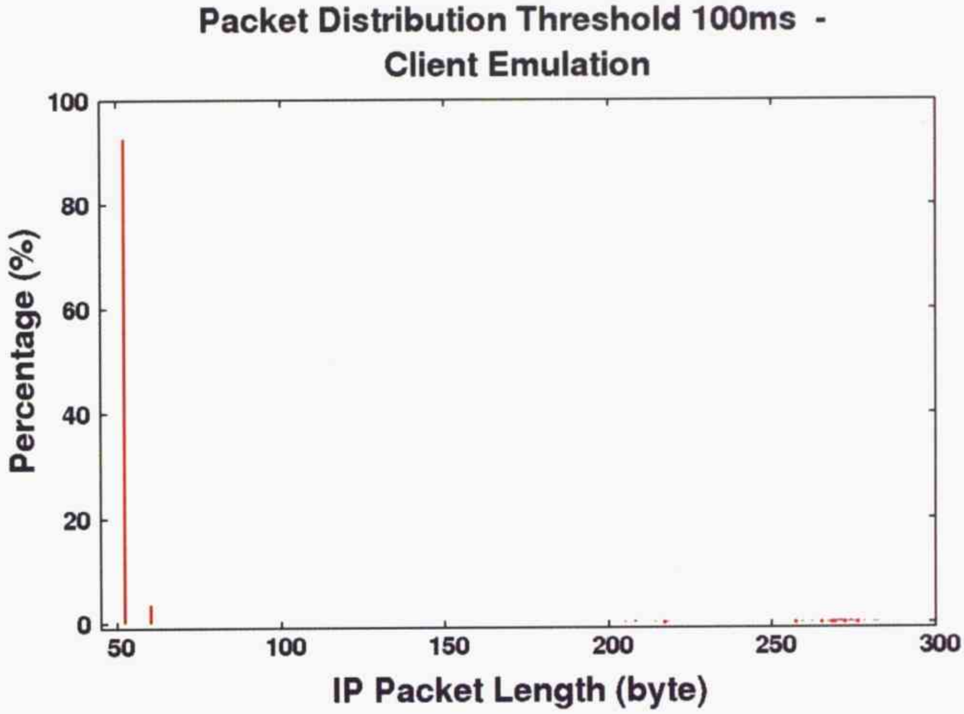

Figure 6.22: IP Packet Percentage Distribution – Client Section Packing Emulation with 100 ms Threshold

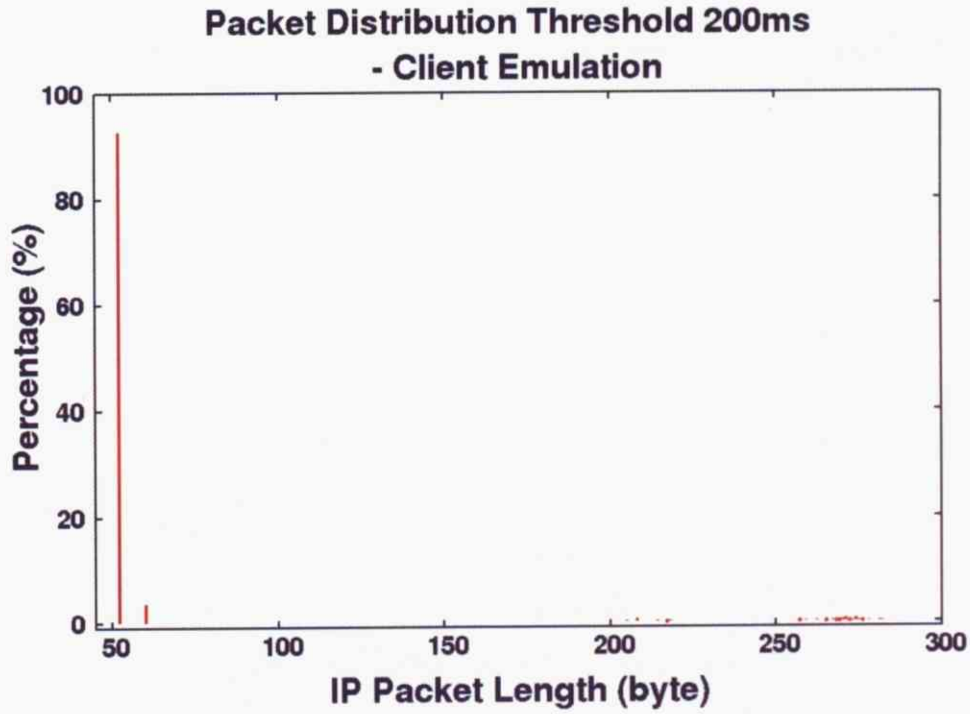

Figure 6.23: IP Packet Percentage Distribution – Client Section Packing Emulation with 200 ms Threshold

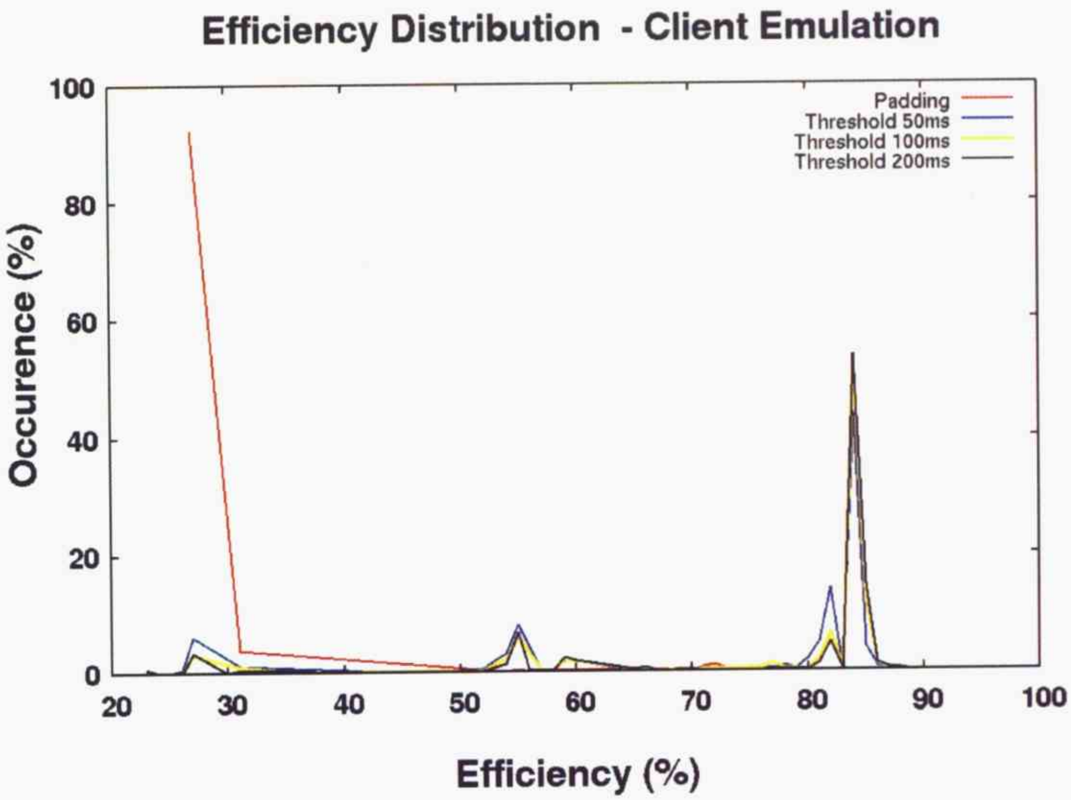

Figure 6.24: Encapsulation Efficiency Percentage Distribution

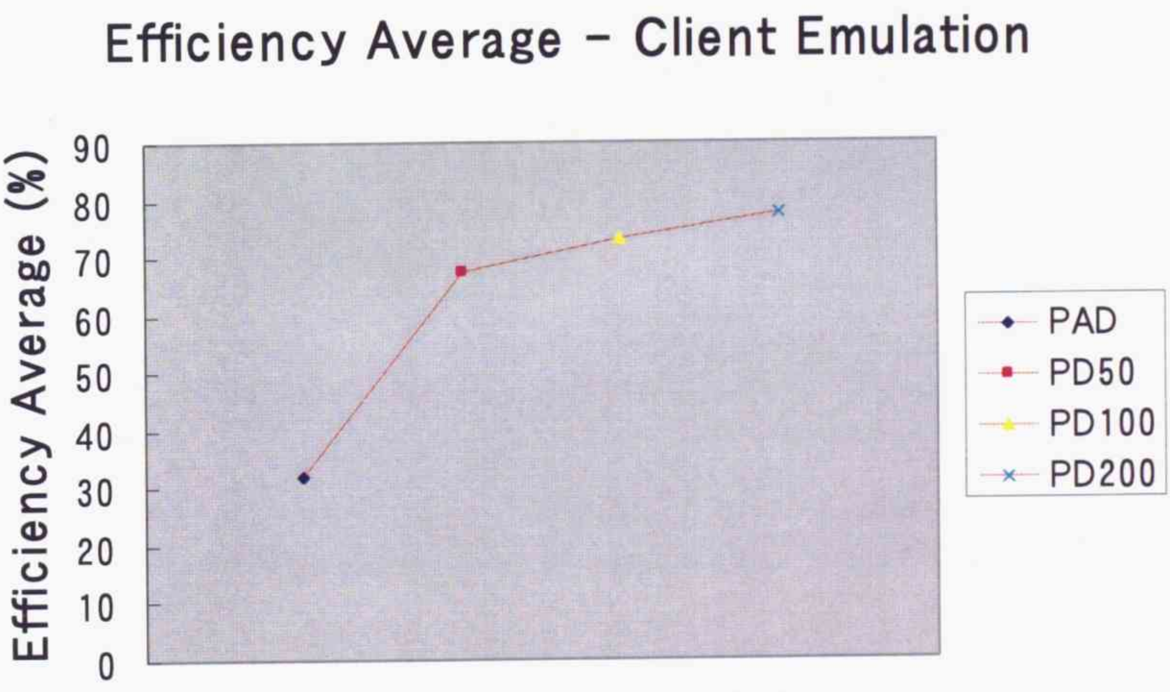

Figure 6.25: Efficiency Average for Client Emulation

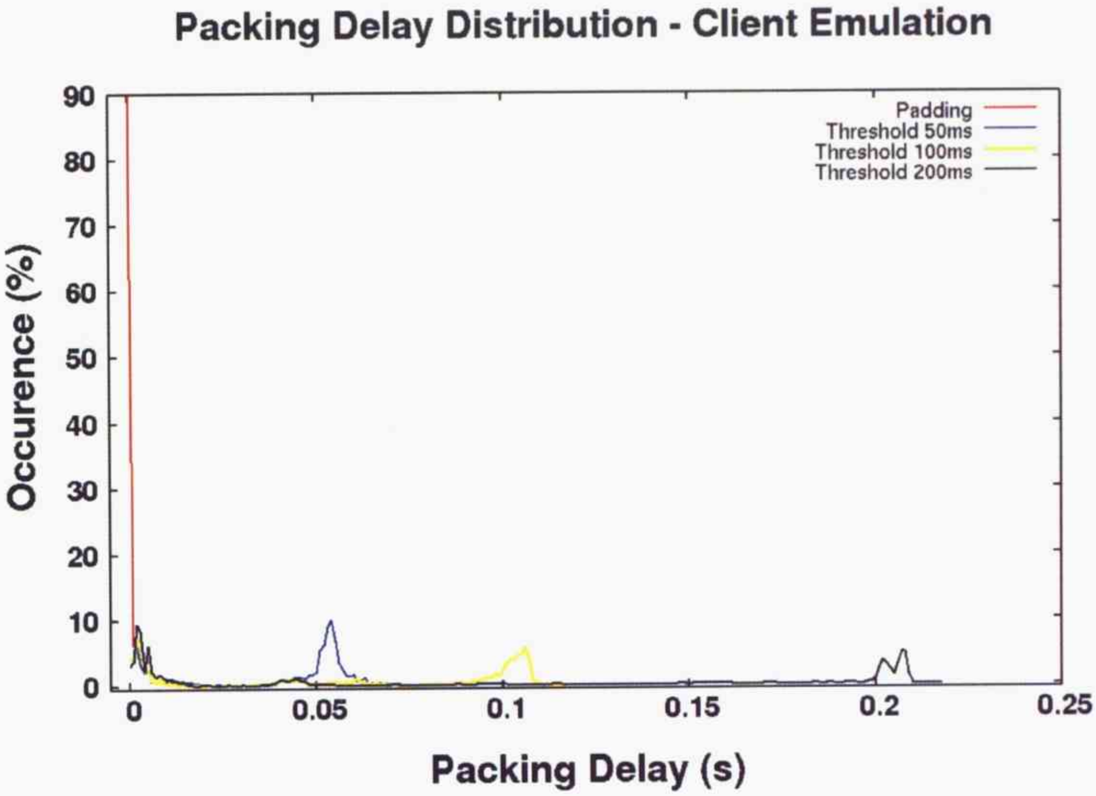

Figure 6.26: Packing Delay Percentage Distribution

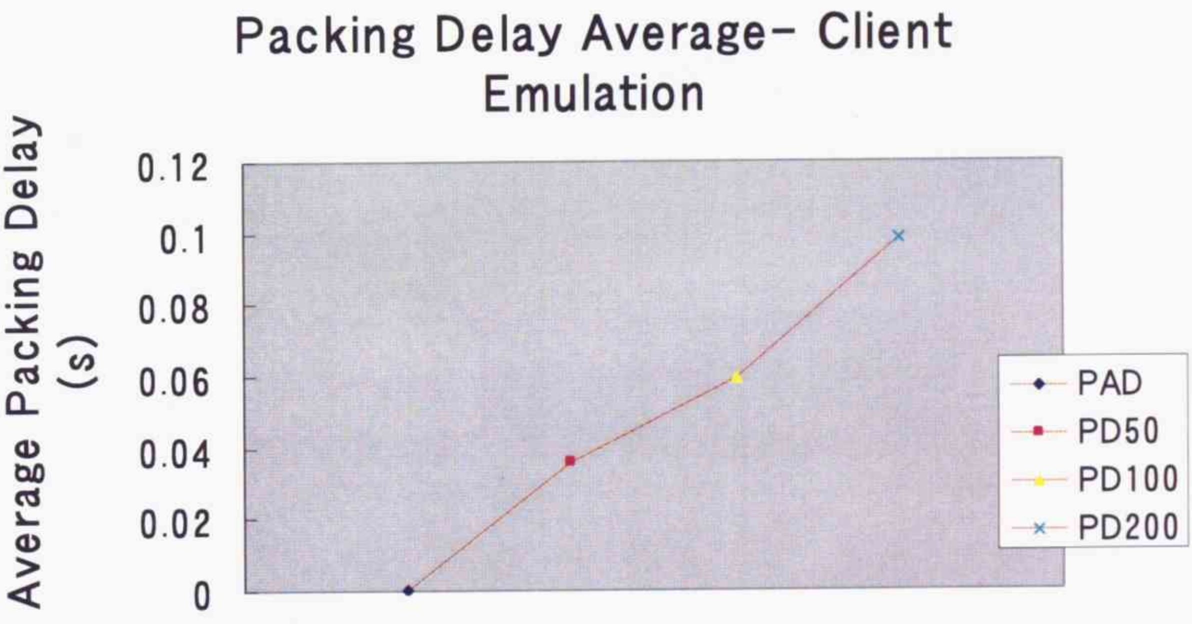

Figure 6.27: Packing Delay Average for Client Emulation

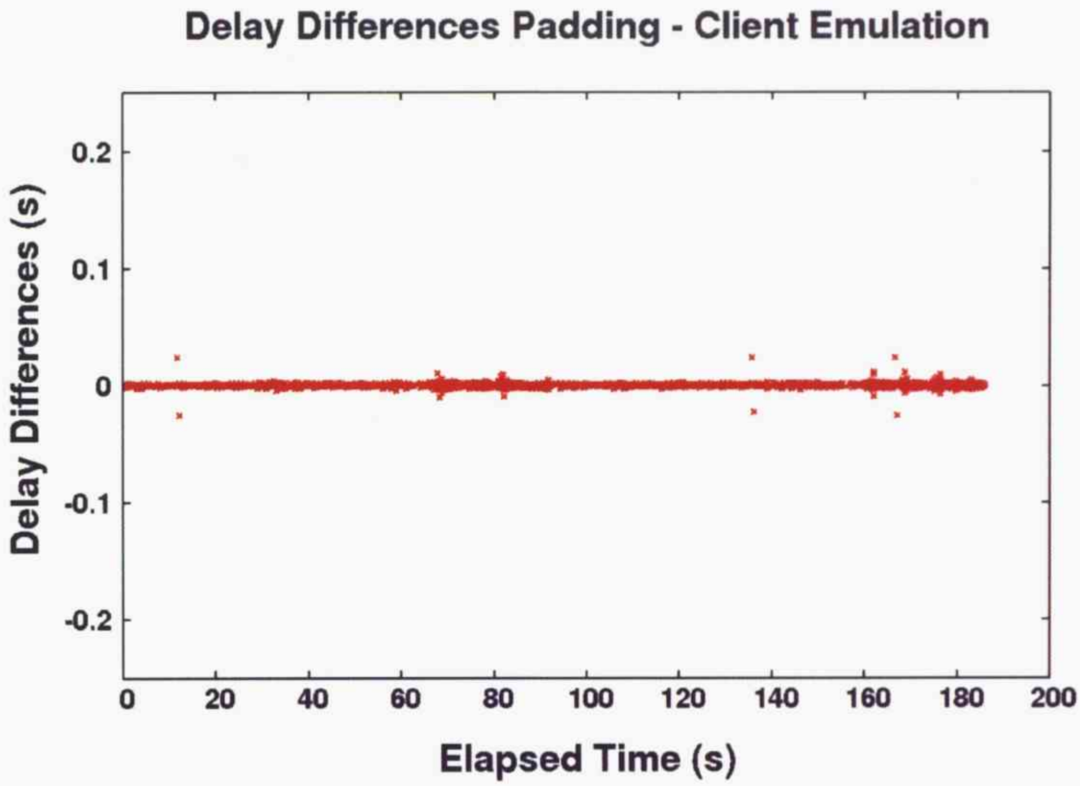

Figure 6.28: Delay Differences vs. Elapsed Time Distribution Result – Client Padding Emulation

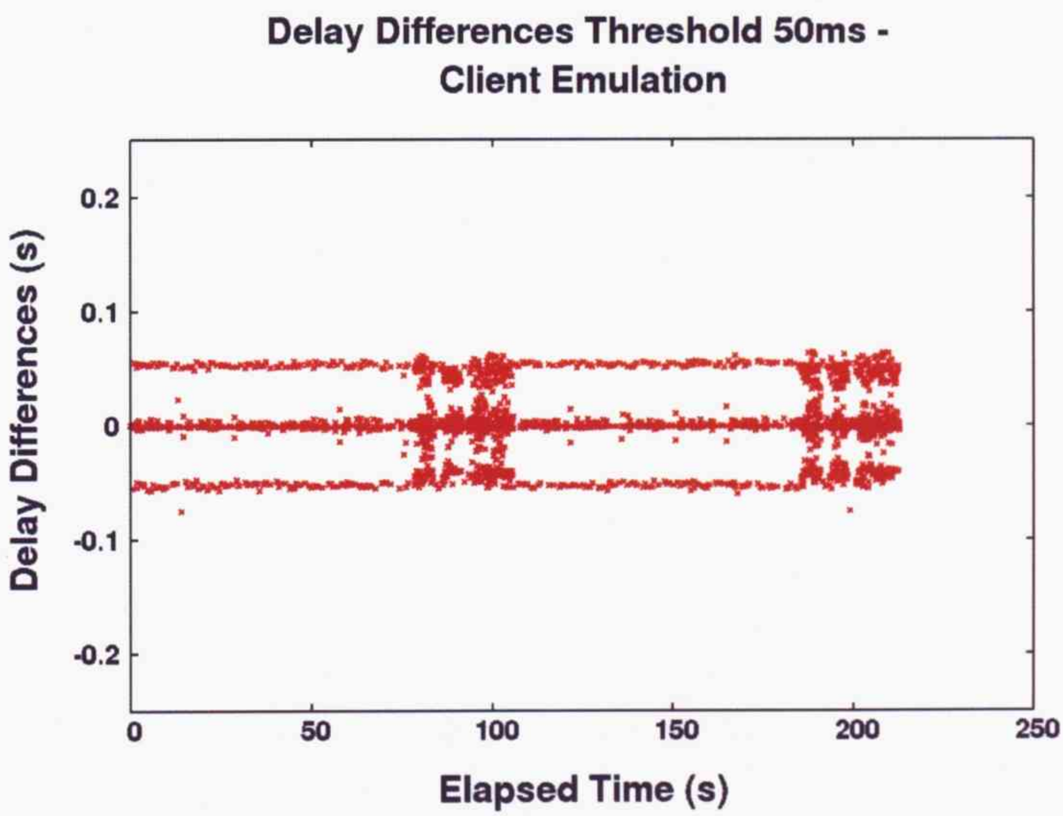

Figure 6.29: Delay Differences vs. Elapsed Time Distribution Result – Client Section Packing Emulation with 50 ms Threshold

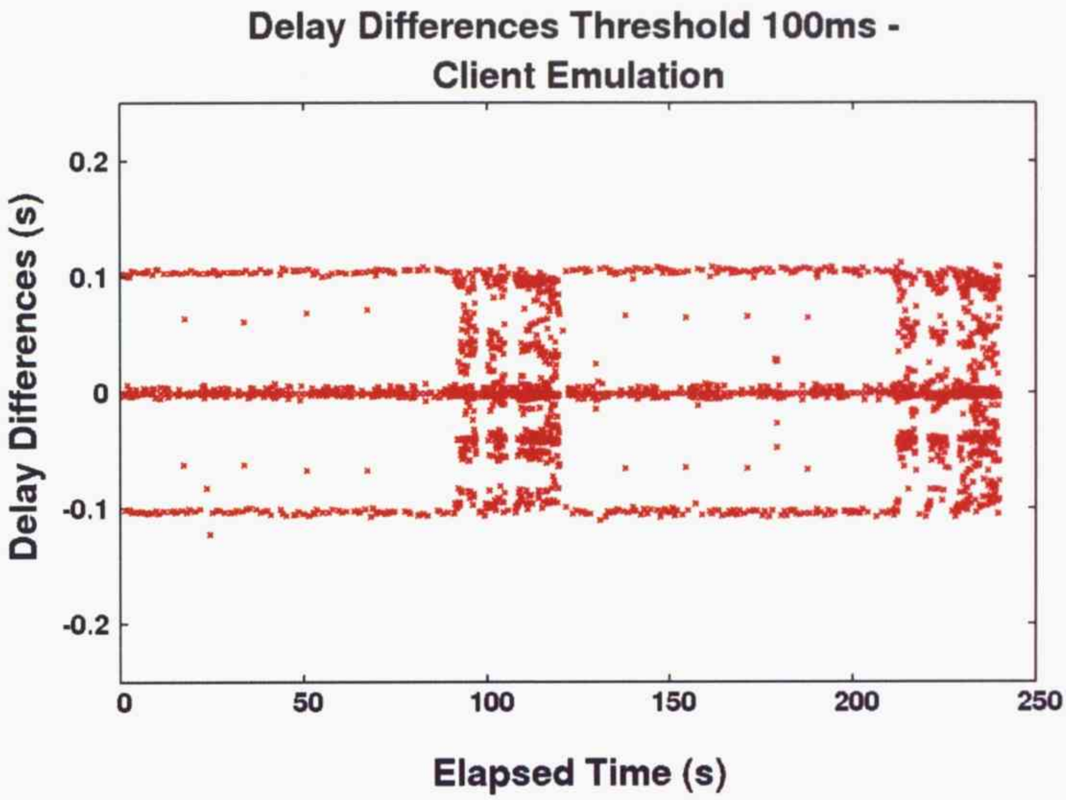

Figure 6.30: Delay Differences vs. Elapsed Time Distribution Result – Client Section Packing Emulation with 100 ms Threshold

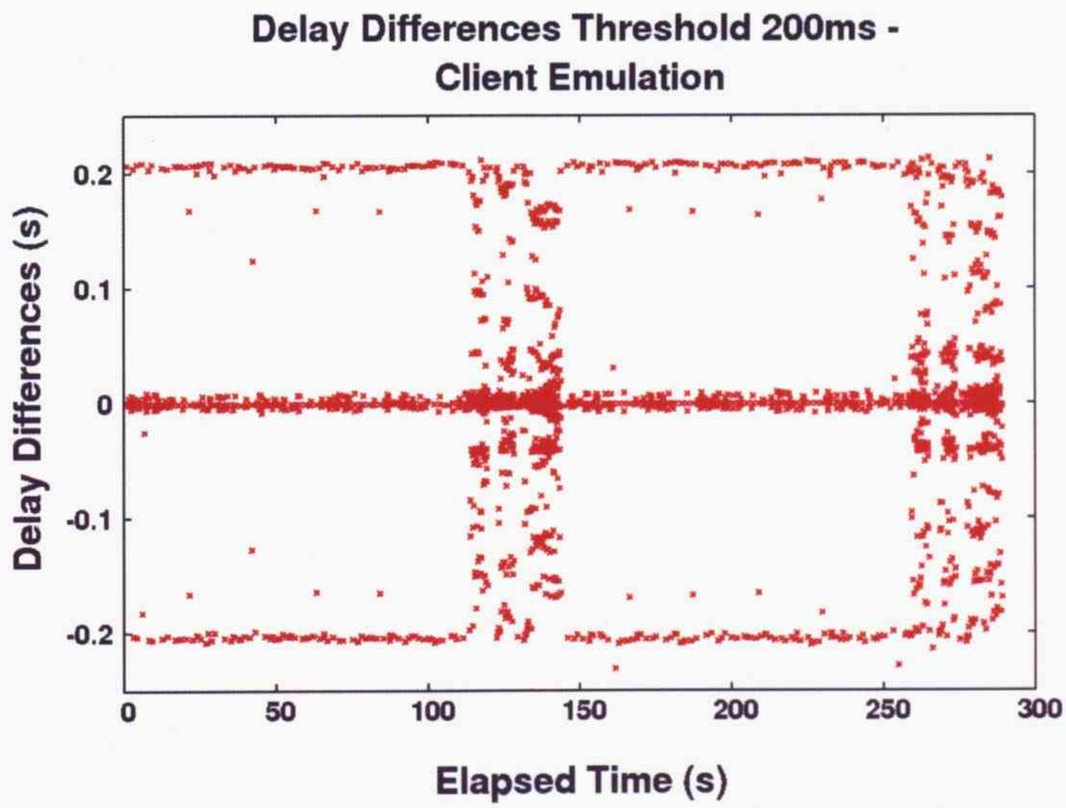

Figure 6.31: Delay Differences vs. Elapsed Time Distribution Result – Client Section Packing Emulation with 100 ms Threshold

6.2.1.2 Discussion

Figure 6.20 – 6.23 shows the IP packet length distribution from the client for every sample. Packet distribution in client emulation is almost similar to the client simulation. From these results we could expect efficiency results are also comparable to client simulation.

Figure 6.24 shows the efficiency distribution for the emulation in client traffic. Padding mode efficiency distribution is almost the same as padding mode efficiency results in client simulation. However for packing mode, since in TCP connection client will wait replies from server before sending another request, more packets from client cannot be packed together. Hence the packing mode efficiency results are not as efficient as client simulation. In client traffic emulation, no real PD involved in the connection and packets from clients are assumed can be packed together if the interval time is lower than threshold. On the other hand in client traffic simulation, packet will not be generated if no reply came from the server and server will not reply if no request packet came from the client. As the results, ACK or request packet that still in the encapsulator will time out and have to be padded. Therefore the occurrence of high efficiency peak is less compare to client traffic simulation.

Figure 6.25 shows the efficiency average values in client traffic simulation. Like client traffic simulation, our proposed method increases efficiency by more than 40% regardless of threshold values. When threshold values increase, naturally the average efficiency will increase. Nevertheless the difference is around 10% when compare between 50 ms threshold and 200 ms threshold.

Figure 6.26 shows the packing delay distribution. In packing delay distribution we can observe that unlike simulation, time-out for section packing do not sharply equal to the threshold values. ULE emulator is implemented in Linux's user space. There will be margin of error for the process to receive signals from the kernel for the time-out mechanism. Multi-threaded implementation may contribute more to the error range as the overhead for the mutual-exclusiveness.

However we can observe that higher threshold values means fewer packets have to be padded by assessing the occurrence of the packing delay timeout. This will be the natural results of section packing since a packet will have a better chance to be packed if it waits longer. The result is especially true in client traffic where client have to wait data packet from the server before issuing packets to acknowledge it and has longer packet inter-arrival time compare to server traffic.

Figure 6.27 shows the packing delay average for each sample and dictates that the average packing delay goes almost in a linear line according to threshold values.

Bigger threshold value, means a packet has to wait longer to be packed and therefore bigger PD value.

Figure 6.28 to 6.31 figure show the Delay Differences distribution in second vs. elapsed time in second. One straight line at 0 second can be seen in padding mode result illustrated in figure 6.28. From the graph we can say that in padding mode virtually no Delay Differences can be observed since all packets will be sent immediately.

However in section packing mode, 3 straight lines can be seen: 1 at 0 second, 1 at +threshold value and 1 at –threshold value. When the packet rate is high, more packets have delay difference between + and – threshold values. Same similarity can be seen for every threshold values.

From the results we could conclude that the proposed method is as efficient as packing mode but the Delay Differences between packets will be larger according to threshold values.

In the emulation we are not able to do analysis for unlimited section packing mode. This is because, when establishing connection to the server, the client has to wait acknowledgement from the server to its request. However with infinity waiting time, the request packets have to wait almost forever to be packed and sent, since there are no other packets exist in the testbed. Eventually client will time out and no connection to the server could be established. Unlimited section packing mode might be functional if there are a lot of packets to be encapsulated like in a scenario of an Internet gateway.

6.2.2 Server Scenario

6.2.2.1 Results

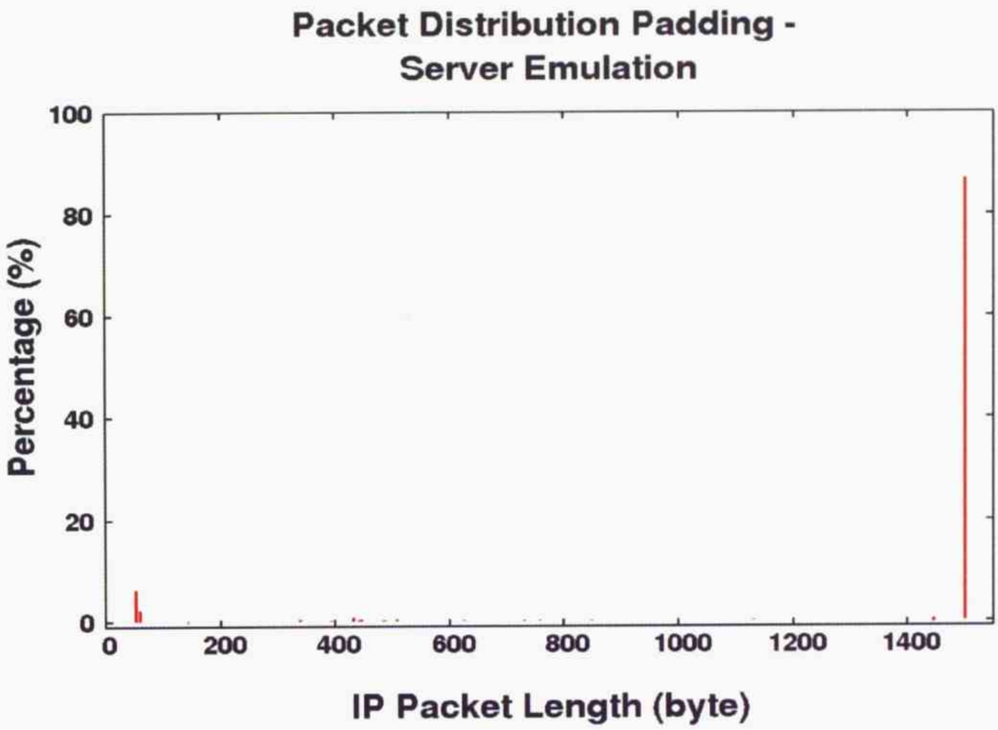

Figure 6.32: IP Packet Percentage Distribution – Server Padding Emulation

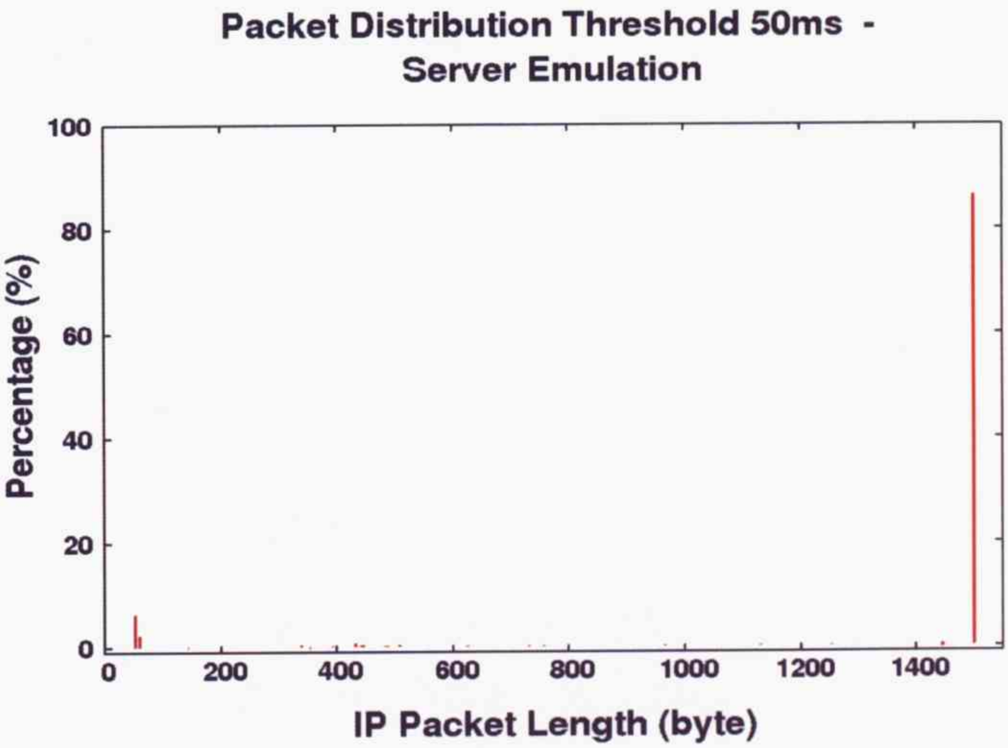

Figure 6.33: IP Packet Percentage Distribution – Server Section Packing Emulation with 50ms Threshold

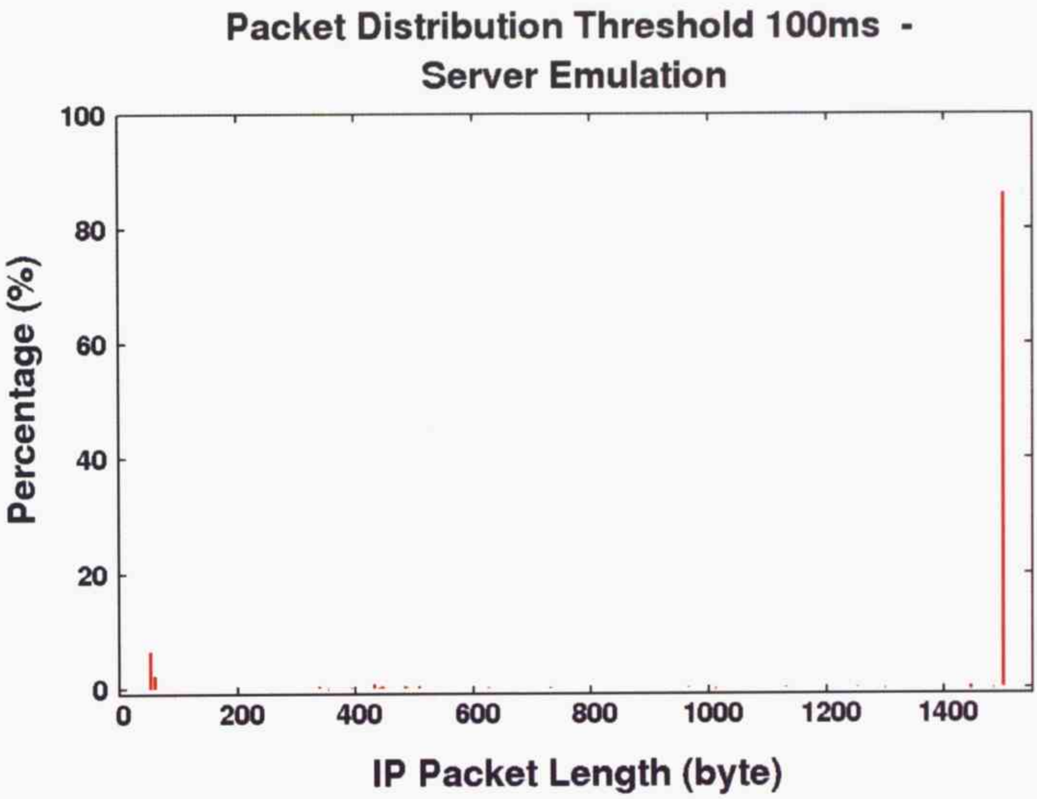

Figure 6.34: IP Packet Percentage Distribution – Server Section Packing Emulation With 100ms Threshold

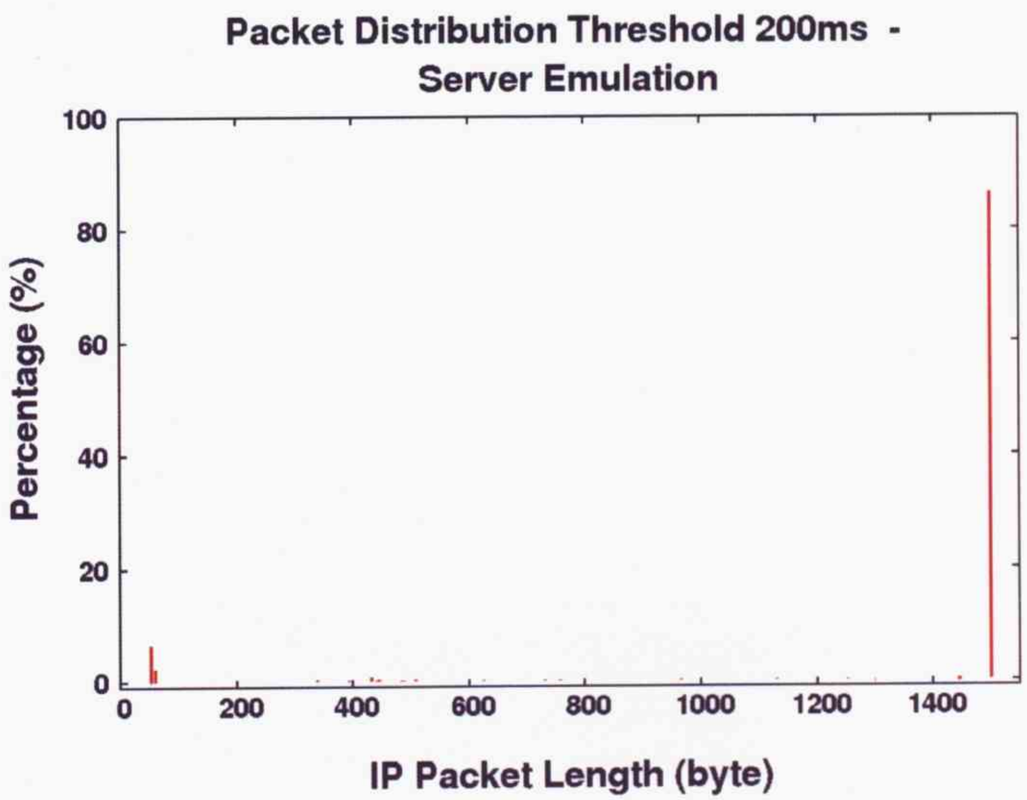

Figure 6.35: IP packet Percentage Distribution – Server Section Packing Emulation with 200ms Threshold

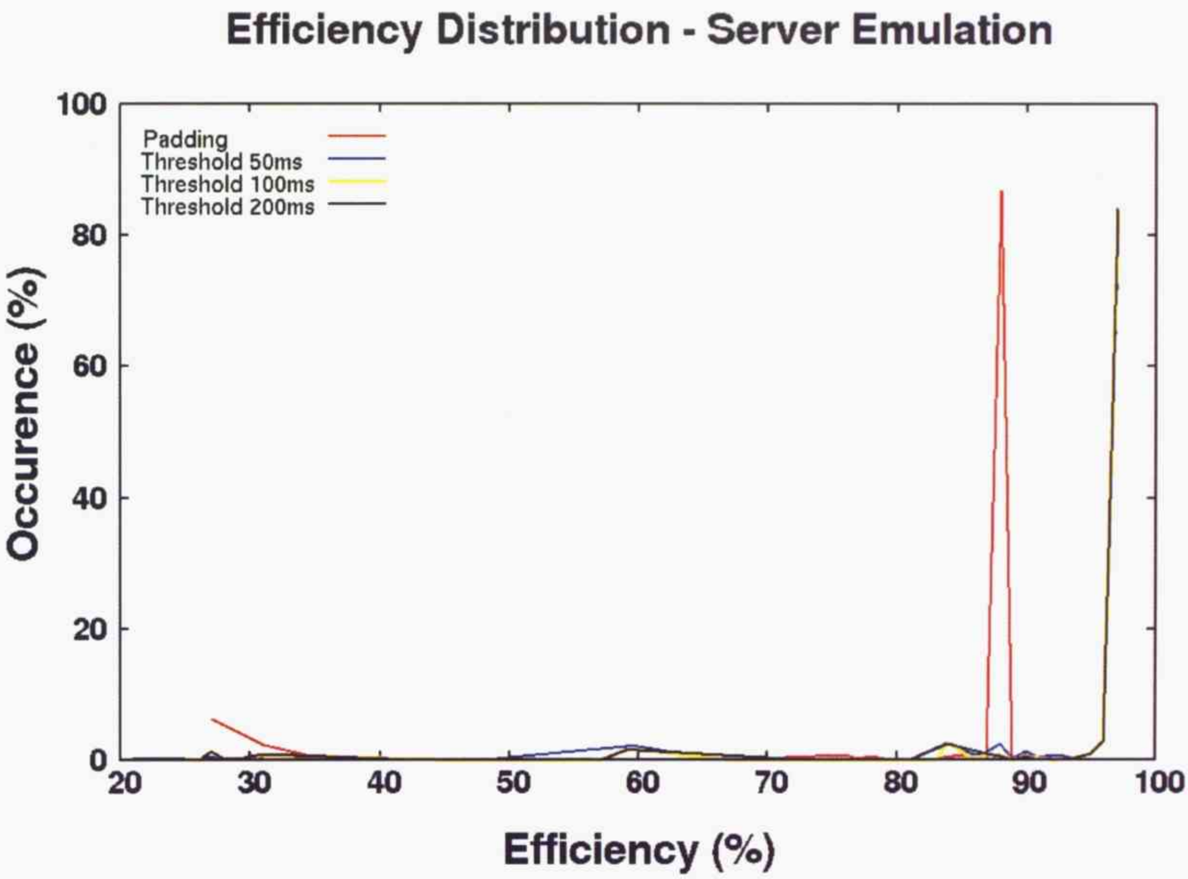

Figure 6.36: Encapsulation Efficiency Percentage Distribution – Server Emulation

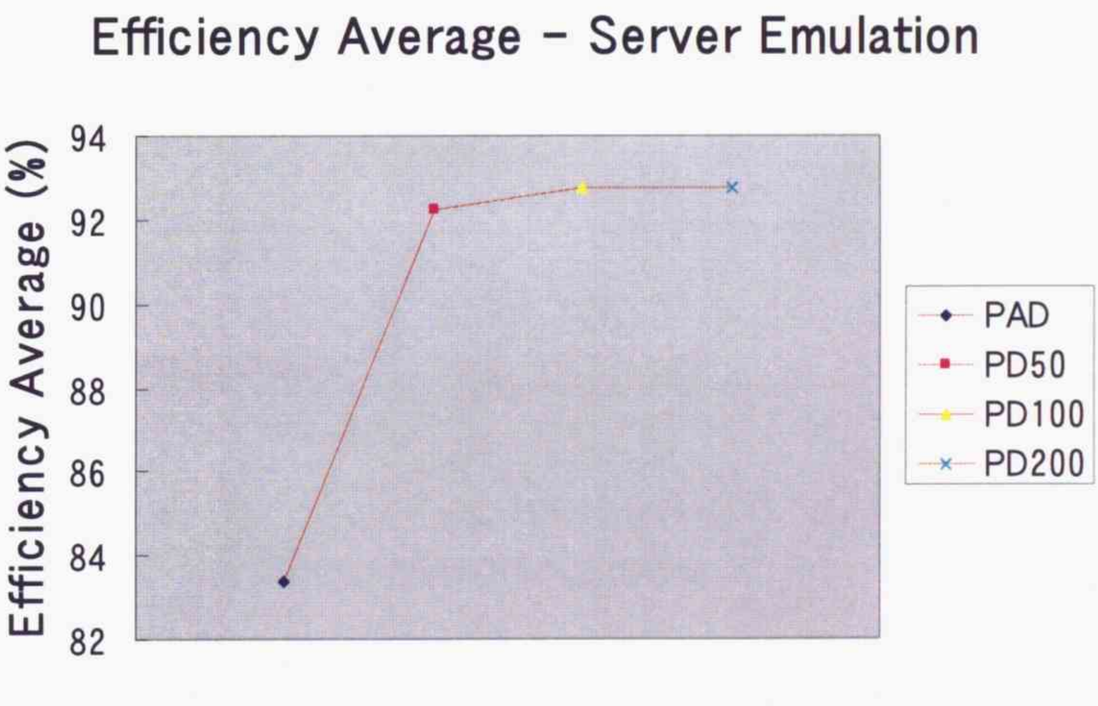

Figure 6.37: Efficiency Average for Server Emulation

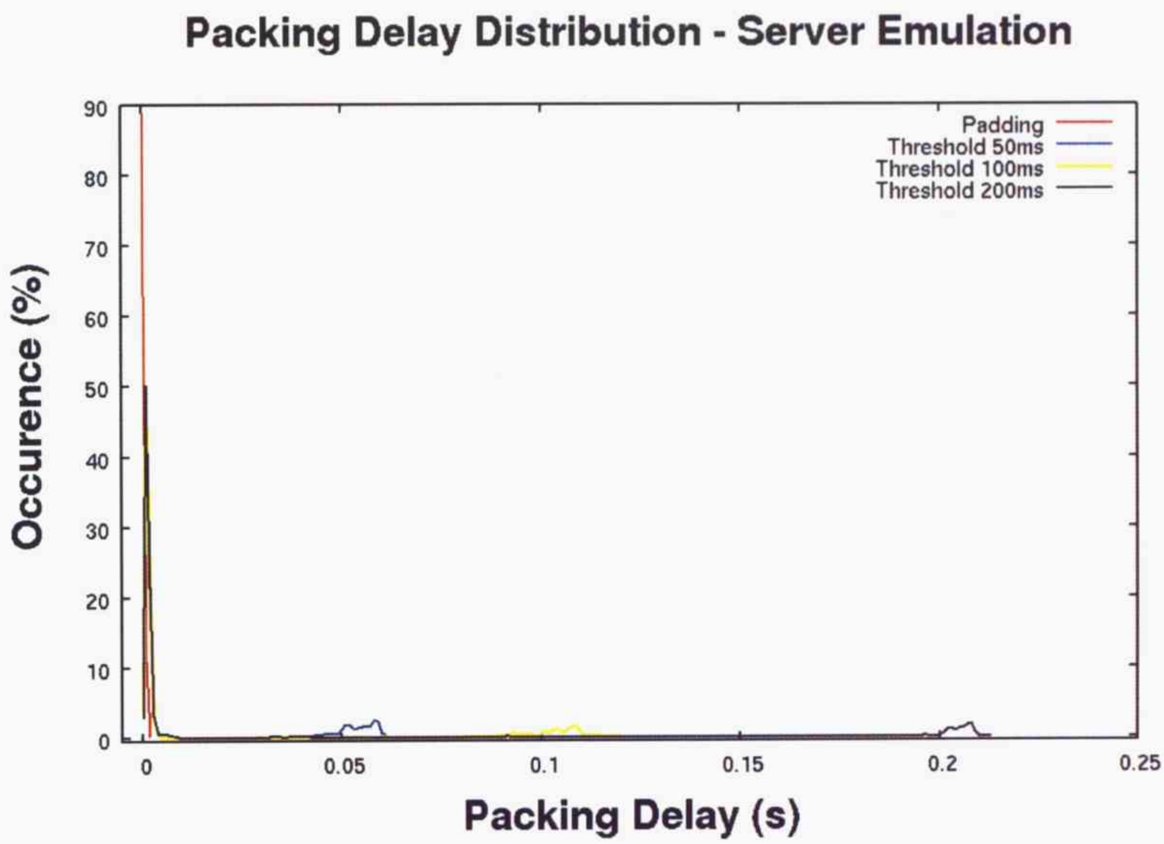

Figure 6.38: Packing Delay Percentage Distribution – Server Emulation

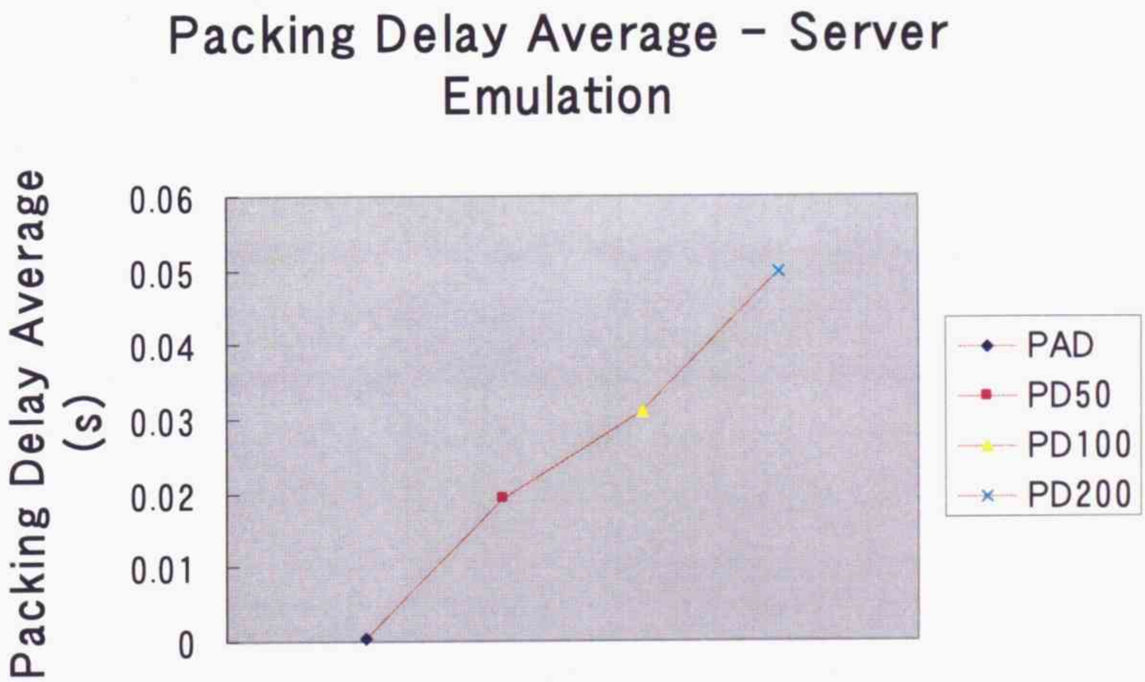

Figure 6.39: Packing Delay Average for Server Emulation

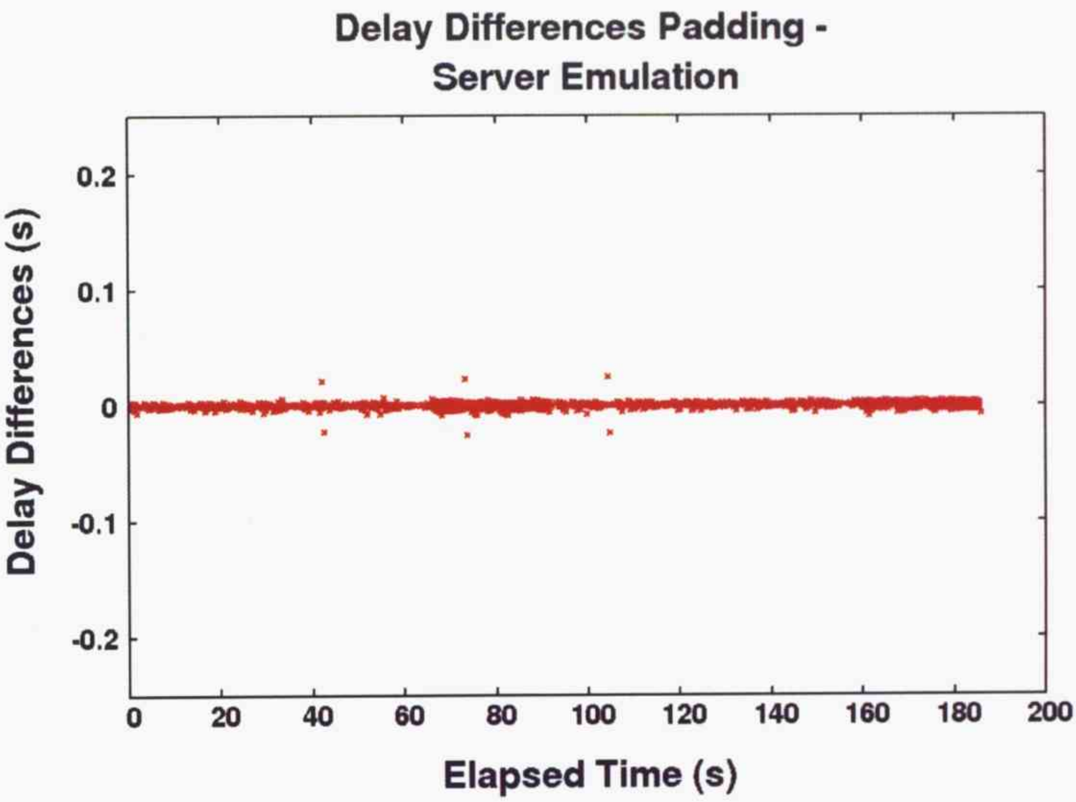

Figure 6.40: Delay Differences vs. Elapsed Time Distribution Result – Server Padding Emulation

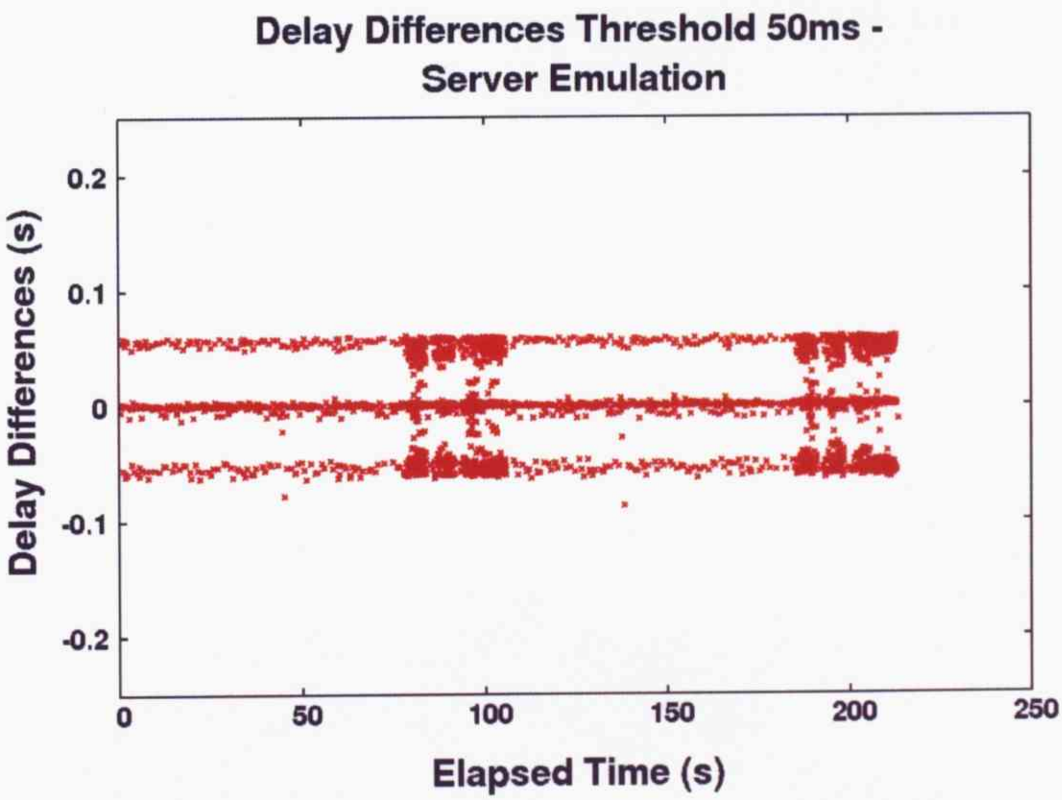

Figure 6.41: Delay Differences vs. Elapsed Time Distribution Result – Server Section Packing Emulation with 50 ms Threshold

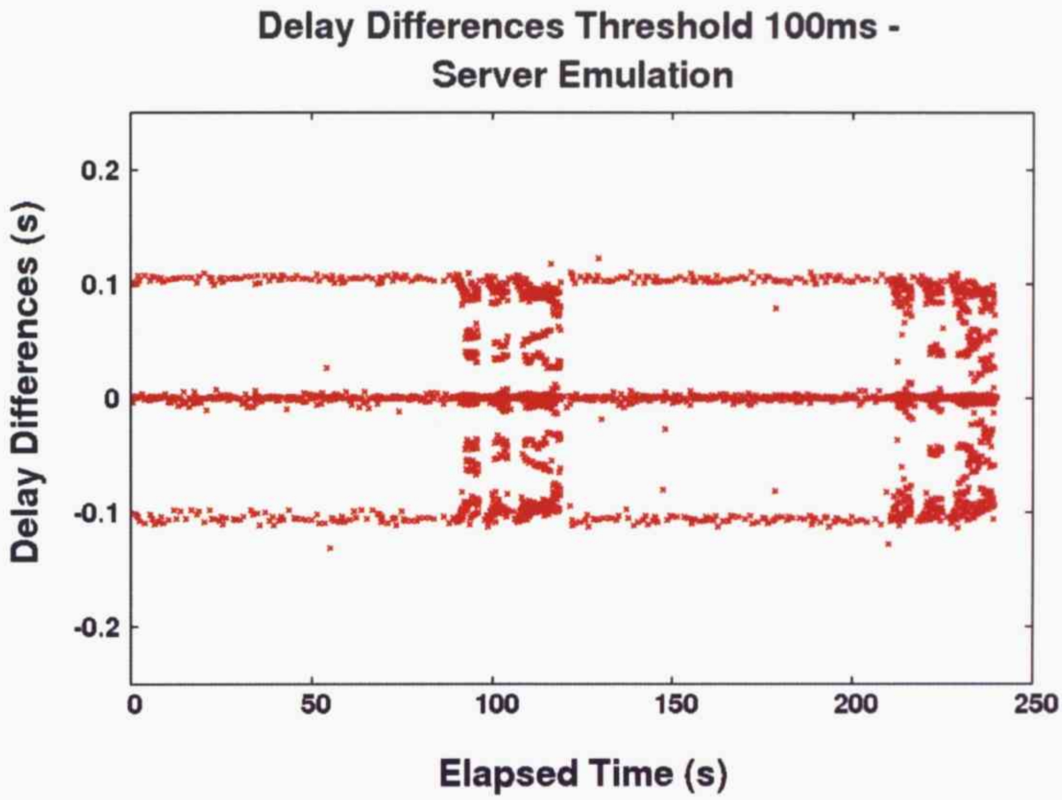

Figure 6.42: Delay Differences vs. Elapsed Time Distribution Result – Server Section Packing Emulation with 100 ms Threshold

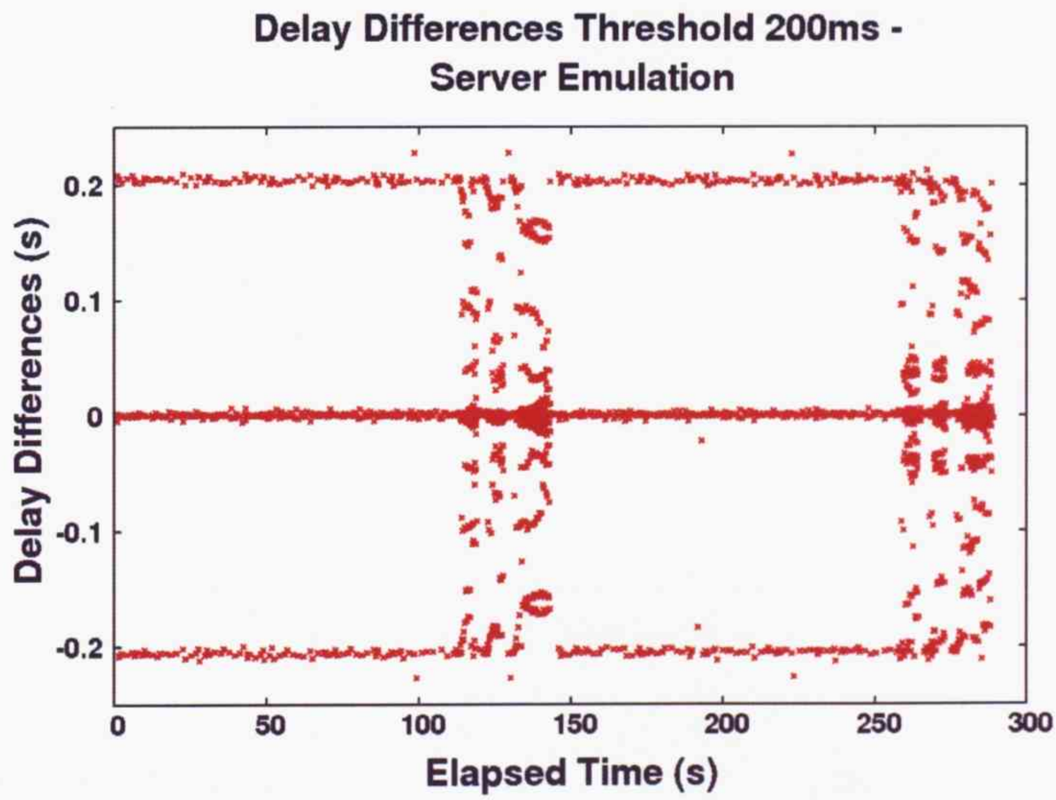

Figure 6.43: Delay Differences vs. Elapsed Time Distribution Result – Server Section Packing Emulation with 200 ms Threshold

6.2.2.2 Discussion

Figure 6.32 to 6.35 show the IP packet length distribution from the server for every sample. Packet proportion of the server traffic is almost similar for every case and almost identical to server traffic simulation. Like the simulation results, MTU sized (1500 bytes) IP packets dominate over 80% of the traffic. These distributions are expected since the server only sends big data packets to answer requests from the server. The small percentage of small packets below 100 bytes is the TCP control packet for handshaking and connection termination when the request has been completed.

Figure 6.36 shows the efficiency distribution for server traffic. Since most of the packet is 1500 byte in size, padding mode have maximum peak at 88% efficiency. Meanwhile section packing mode for every threshold setting have maximum peak at 97% efficiency, 9% more efficient that the padding mode.

Figure 6.37 shows the efficiency average for the server emulation. Section packing of any threshold values averagely is around 8% more efficient compare to padding mode. But contrary to client emulation, the difference between each threshold setting is less than 1%. From figure 6.36 we can also observe that 3 lines indicating 3 different threshold values almost overlapping each other showing that they have similar efficiency values. The different is small in server traffic thanks to the higher packet rate per second and therefore smaller packet inter-arrival times. That means larger threshold values have less influence to the efficiency results.

Together with client traffic emulation results, these results suggest that traffic with lower traffic aggregation like client traffic is more sensitive to section packing threshold and section packing mode efficiency improve more than padding mode. Traffic with smaller packet inter-arrival times is less sensitive to threshold values like traffic from server.

Figure 6.38 shows the packing delay distribution for sever traffic. Similar to client emulation results, due to the time granularity and margin error of the encapsulator, time out accuracy is not as perfect as the simulation results.

Figure 6.39 shows packing delay average for each sample. The tendency is similar to client emulation where the result almost forming a linear line. However the average values for each sample is less than client emulation due to the smaller packet inter-arrival times.

Figure 6.40 to 6.43 show the Delay Differences distribution for the server traffic. Delay Differences results in server traffic are almost similar to client traffic results where padding mode virtually have no Delay Differences and our proposed section packing mode with threshold method have Delay Differences values between +

and – threshold value.

Delay Differences results show the sensitivity of Delay Differences to the threshold time. We can observe that although server traffic's inter-arrival time is shorter than client traffic's and have better PD compare to client traffic, Delay Differences results is almost identical.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

The research work described in this thesis is concluded as follows.

- Section packing mode with threshold setting has been proposed
- Implementation of an emulator to emulate behavior of IP/ULE/ over DVB satellite has been done and tested
- Efficiency, packing delay and Delay Differences evaluations have been carried out for padding mode, section packing mode and proposed method. Evaluation is done by simulation and emulation
- Performance of proposed method in term of efficiency is comparable to section packing mode and in average increase more than 40% in client traffic and more than 10% in server traffic when compare to padding mode.
- Efficiency increases when comparing to padding mode are more obvious in client traffic where most of the packets is small request packets.
- During emulation, efficiency difference between proposed method and section packing mode is less obvious in server traffic compare to client traffic. This is because server traffic has smaller packet inter-arrival time therefore more packet can be packed before time out. Bigger IP packets also contribute to this characteristic since encapsulation efficiency is high when IP packet is big.
- Packing Delay and Delay Differences performance is sensitive to threshold values in padding mode, section packing mode and proposed method.
- Delay Differences is similar regardless of traffic characteristic. It will be between + and – threshold values. 3 lines will exist in the graph at 0 second, -threshold value and +threshold value.
- Threshold value tuning is an important parameter for efficiency, Delay Differences and Packing Delay tradeoff. The threshold should be tuned according to network preference. Naturally time sensitive application

should have less threshold value to improve Delay Differences and Packing Delay performances. Other traffic should be packed as often as possible to increase bandwidth utilization.

7.2 Future Work

Below is the possible future work suggested from the outcome of this research.

- This thesis only conducts experiment for satellite link. However DVB system is also supported in other broadcast media such as terrestrial wave, cable television and direct broadcast satellite with return channel via terrestrial links which have different link characteristic and latency. We suggest a future work to evaluate efficiency performance to transport IP over MPEG-2 TS as well as Delay Differences and Packing Delay performances in those links.
- Packing multiple IP packets in one MPEG-2 TS cell may increase link error rate since one TS cell lost means multiple IP packets is lost. We suggest an evaluation be conducted to find the influence of section packing mode to link error rate.
- Section packing introduces more Delay Differences to the network. We suggest a future work to be done on the mechanism to reduce the effect especially for time sensitive application.

References

[1]EBU/ETSI, "Digital Video Broadcasting (DVB), Interaction channel for satellite distribution systems", ETSI EN 301 790, 2003

[2]EBU/ETSI, "Digital Video Broadcasting (DVB): DVB specification for data broadcasting", EN 301 192, 1999

[3]ETSI, "Multiprotocol Encapsulation", Draft EN 301 192 V1.1.1, 1997-08

[4]H. D. Clausen, H. Linder, B. Collini-Nocker, "Internet over direct broadcast satellites", IEEE Commun. Mag (1999), vol. 37. no.6, pp.146–151.

[5]G. Fairhurst, B. Collini-Nocker, "Ultra Ligthweight Encapsulation (ULE) for transmission of IP datagrams over MPEG-2/DVB networks", IETF, Internet draft, draft-ietf-ipdvb-ule-06.txt , 2004.

[6]E. Duros, W. Dabbous, H. Izumiyama, N. Fujii, and Y. Zhang, "A Link-Layer Tunneling Mechanism for Unidirectional Links", RFC 3077, March 2001.

[7]B.G Haskell, A.Puri, A.N Netravali, Digital Video: An Introduction to MPEG-2, Chapman & Hall, New York, 1997.

[8]EBU/ETSI, Digital Video Broadcasting (DVB): Implementation Guidelines for the Use of MPEG-2 Systems, Video an Audio in Satellite, Cable and Terrestrial. ETR 154, 1997.

[9] WIDE Project Japan, http://www.wide.ad.jp/

[10] J. Marescaux, J. Leroy, M. Gagner, F. Rubino, D. Mutter, M.Vix, S.E. Butner and M. K. Smith, "Transatlantic robot-assisted telesurgery", Nature, Vol. 413, pp. 379-380, Sep. 2001.

[11] "Understanding Delay in Packet Voice Networks", CISCO
White paper, http://www.cisco.com/warp/public/788/voip/delay
details.html, Oct 2002.

[12]"Video conferencing", A1 project, Institute of Communication Networks, Austria,
http://www.ikn.tuwien.ac.at/ftw-a1/services.htm