Analysis of IP Encapsulation Methods over DVB Satellite
DVB 衛星における IP パケットカプセル化方式の解析

Zul Hilmi Zulkifli

***ABSTRACT* – Despite the fact that there are some problems with latency and high error rate with satellite connections, they are expected to play an important role in providing Internet Protocol (IP) services to complementing next-generation terrestrial network.**

In this paper we will do efficiency analysis for various encapsulation methods to transport IP packets via satellites used in Digital Video Broadcast (DVB) open standard. We do analysis based on theoretical calculations and also based on real traffic from three different scenarios or network models.

1. INTRODUCTION

Satellite communication technology has been developed for nearly 50 years. For many years Geosynchronous earth orbit (GEO) transparent (bent-pipe) satellites have been the most important element of telecommunication networks, serving particularly long distance telephone services and television broadcasting.

In the latest trend of global telecommunication, Internet Protocol (IP) traffics hold most of the share in total traffics. Consequently demand to use satellite in IP networks are increasing to complement existing terrestrial communication networks. IP over satellites offer significant advantages such as wide geographic coverage, broadcast capability, rapid deployment and support for mobile stations. This is important for Internet connections in areas and countries which are not covered by good terrestrial connections due to the rough terrains.

Currently there are many standards exist on IP transportation via satellites. However, most of the standard is exclusive only for certain satellite connection provider. As the result, the equipments tend to be very expensive. To reduce deployment cost, open standards are required so that equipments could be mass produced. European Telecommunication Standards Institute (ETSI) defined 2 open standards in satellite communication; Digital Video Broadcast for Satellite (DVB-S) and DVB Return Channel via satellite (DVB-RCS). These standards are widely in used currently.

DVB standards initially intended for digital audio video broadcasting using MPEG-2 Transport Stream (TS) and not optimized for IP packets. Therefore in this paper, we present IP packets encapsulation analysis for various encapsulation methods defined by DVB standards. We do analysis based on theoretical values and also based on the real traffic for various satellite network models. The analysis result is vital for research to enhance efficiency for transporting IP traffic in satellite networks.

2. SYSTEM ARCHITECTURE

In IP over DVB-S, the forward link is a broadband 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 [1] using ordinary terrestrial link such as PSTN. This kind of connection is called UniDirectional Link (UDL) and various work such as [2] have been done to address routing problems in UDL. Figure 1 shows the DVB-s topology.

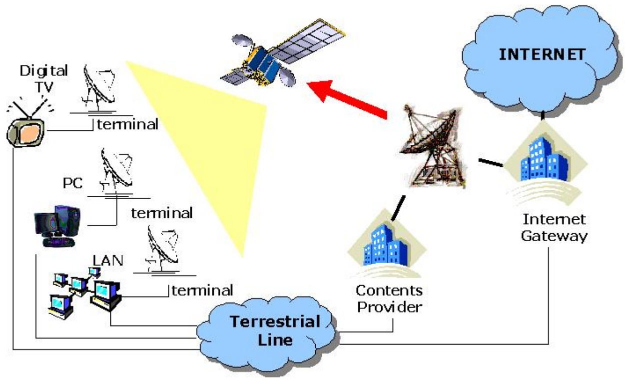

Figure 1. DVB-s Topology

However, in recent development, DVB with return channel via satellite (DVB-RCS) have been standardized enabling full independency to the terrestrial network [1]. DVB-RCS is a modified version of DVB-S with additional standards on how to create interactive return channel using satellites. Figure 2 shows DVB-RCS topology, which can be characterized by having broadcast channel in forward link and point to point (PPP) channel in return link.

The DVB systems family is based on cell-oriented packet transmission system defined by ISO/IEC 13818-1 MPEG-2 systems standard [3].

DVB systems family use fixed size MPEG-2 Transport stream (MPEG-2 TS) to carry packetized data in the forward link. Meanwhile in case of interactive system, return link can use various systems to transport TCP/IP based data according to the link technology in used. For example DVB-RCS use ATM/AAL5 cells by default and also define optional MPEG-2 TS usage to be used in the return link. Since ATM is a well known system, the details about it will not be discussed in details.

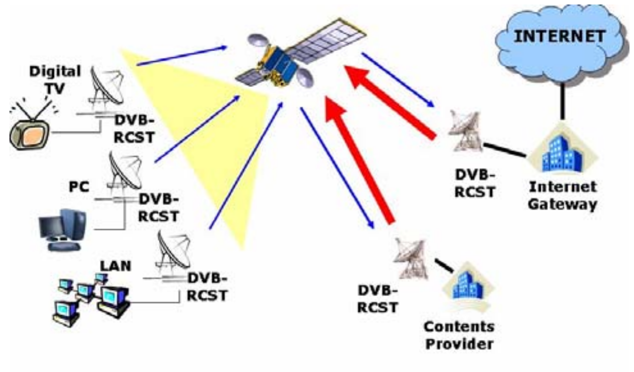

Figure 2. DVB-RCS Topology

3. MPEG-2 TRANSPORT STREAM

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

Figure 3 shows 188 byte fixed sized TS cells [4]. Each TS cells consist 4 bytes of header and 184 bytes payload. Error recovery is easier when using constant cell length (essential in error prone line).

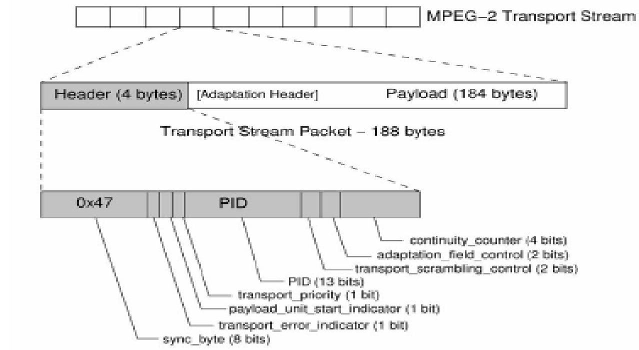

Figure 3. MPEG-2 TS structure

Details about 3 important header fields in this paper are described below.

1. Payload_unit_start_indicator (1 bit): 1 indicates presence of a new PES packet or section layer packet. In case of section layer packet, if PUSI = 1, first byte in the payload contain payload_pointer_field.
2. Packet Identifier (PID) (13 bit): values 0x1FFF is null packet (ignored by the receiver). PID are used to distinguish between different logical channels
3. Payload_pointer_field (8bit, optional) – presence if and only if PUSI equal to 1 in PSI packet. Indicate the number of byte until the new section layer packet started after the field.

Compressed data from a single source (video, audio, and data) and additional control data for the source information form elementary streams (ESs). ESs then are packetized into packetized ES (PES). Each PES packets consist of a header and payload and PES from various elementary streams are combines to form a program [5]

Several programs 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 is program association table (PAT), program map table (PMT), and network information table (NIT).

Details about PSI and other header fields are out of this paper's scope and will not be discussed in detail.

4. IP ENCAPSULATION INTO MPEG-2 TS

Figure 4 shows 3 different methods to encapsulate IP packets into MPEG-2 TS cells [6]. Below is the description of the methods.

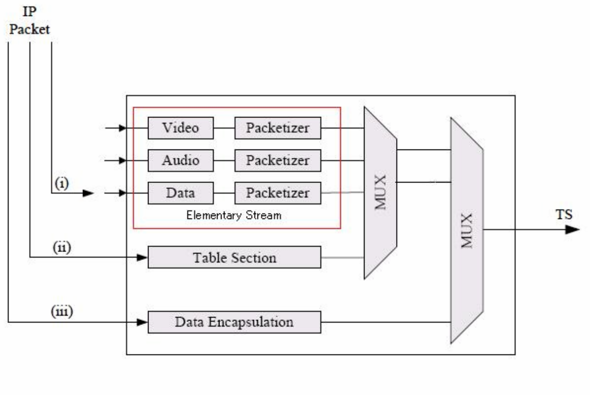

Figure 4: Possible entries for IP packets in MPEG-2 TS

(1) Data streaming: IP packet encapsulation into PES packets.

(2) Multi Protocol Encapsulation (MPE): IP packet encapsulation into Digital storage medium – command and control (DSM-CC) table section packet.

(3) Data piping: IP packet encapsulation directly into TC cells. Current work is Ultra Light Encapsulation(ULE).

4.1 MPE

MPE is the IP encapsulation standard defines 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 DSM-CC Table Section data.

MPE makes use of a medium access control (MAC) level device address and the address format follows the ISO/IEEE standards for LAN/MAN.

MPE packets have 12 bytes of header and 4 bytes of cyclic redundancy check as the tail [6].

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

4.2 ULE

Ultra Lightweight Encapsulation (ULE) has been introduced in an Internet draft [7] to eliminate unnecessary overhead in MPE. ULE encapsulate IP packets directly into a sequence of TS. Unlike MPE, ULE only have 4 bytes of header cutting 8 bytes from the header in MPE and have 4 bytes of CRC or checksum trailer [7].

Destination Address Present Field is the most significant bit of the header. A value of 0 indicates the presence of optional Destination Address Field (6 byte) in the payload and value 1 indicates the field is not present.

ULE is still in the draft and yet to be standardized. Currently the standard method to encapsulate IP into MPEG-2 TS is MPE. MPE is neither elegant nor efficient solution but for the time being seems to be generally accepted.

4.3 Padding and Section Packing

There are 2 modes to encapsulate IP packets into MPEG-2 TS using MPE or ULE. First mode is padding mode and the second one is section packing mode.

In both modes, one IP packet is encapsulated independently into one section layer packet using MPE or ULE respectively. Both modes differ only how section layer packets is divided or inserted into TS cells.

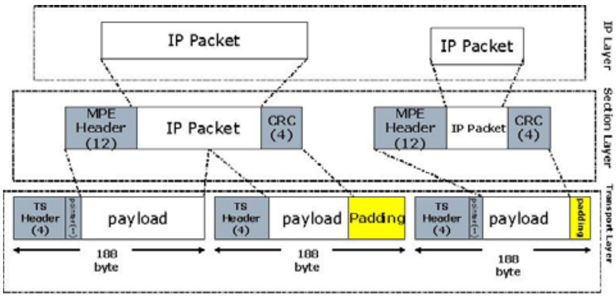

Figure 5: MPE padding mode

In padding mode, one section layer packet is encapsulated independently into TS cell or cells. Since TS cell is fixed size cell with 184 bytes of payload, a section layer packet is not necessary to perfectly fit into one or multiple TS cells. In that case, the leftover space will be padded with padding bytes and are considered as overhead. In padding mode, one IP packet is encapsulated into section layer packet and is instantly inserted into TS cell or cells. Therefore certain IP packet will not have to wait for other IP packets to come. Jitter for each packet is expected to be minimal. Figure 5 shows MPE encapsulation method in padding mode. ULE encapsulation is similar to MPE and only differs in overhead number.

Meanwhile in section packing mode, leftover space will no be filled with padding bytes. If there are leftover spaces, it will be filled with the next section layer packet. In this mode, ULE draft defines the leftover space in TS cell should be more than 2 bytes before new section layer packet is inserted. It is because first 2 bytes of the ULE header can't be divided into multiple cells. However MPE specification does not define explicitly how to do section packing. Figure 6 shows section packing mode in MPE encapsulation. ULE have similar mechanism except having different total overhead. In section packing mode, if section layer packet is smaller than TS cell payload, it has to wait for the next packet to come before being transmitted. If it is longer than TS cell size, it will be divided into multiple cells. All filled TS cells will be transmitted immediately while the last cell will have to wait for the next packet. Therefore, in this mode since no padding byte is necessary; the efficiency is expected to be higher with the cost of having longer jitter produced by different packet waiting time.

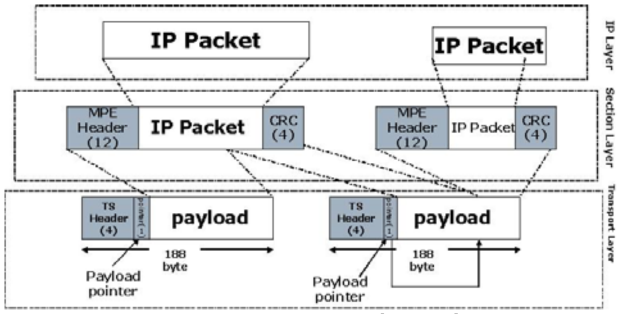

Figure 6. MPE section packing mode

5. EFFICIENCY

In this section we describe efficiency calculation for various IP packet encapsulation methods that will be used in our analysis later. The methods are MPE, ULE, ATM/AAL5 and Ethernet

5.1 MPE/ULE Efficiency equation

MPE and ULE have similar efficiency equation. MPE have 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 encapsulated in TS cells without concatenation with other IP packets (padding mode). The later is when multiple IP packets can be concatenated into TS cells (section packing ON).

Each TS cells have 4 bytes of header and 184 bytes of payload. The first cells will have 1 byte payload pointer, so it will have only 183 bytes payload. In padding mode, IP packets only start at the beginning of Transport stream cells, and the remainder will be padded by padding 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 denote by:

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

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

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

The efficiency will be:

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

Now if section packing mode 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:

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

And the efficiency will be:

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

2. IF $S < 183 - L$ then there will be one payload per transport packet. We could add the overhead into TS overhead making it 5 bytes per TS packet. Therefore total transport layer overhead per IP packet will be:

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

And the efficiency will be:

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

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

4. In ULE, TS cells must at least contain 2 bytes of additional space before accepting another section layer packet. 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 $S > 183 - L$; we divide IP packets into multiple TS cells. Then the last TS cell 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 we need another byte to put payload pointer. Therefore total free bytes needed to put a new SNDU are minimum 3 bytes. Meanwhile when $S < 183 - L$, TC cells will already contain payload pointer and only need 2 extra bytes for the next section layer packet.

5.2 Ethernet Efficiency equation

Ethernet adds the following overhead to IP packets:

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

Total Overhead is 26 byte.

Minimum Ethernet frame is 64 bytes (excluding preamble), therefore, packet less than 46 bytes will be padded. If IP packets size in bytes is denoted by S, Efficiency, *E* will be defined as:

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

$$E = \frac{S}{72} \quad (8)$$

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

$$E = \frac{S}{S+26} \quad (9)$$

5.3 ATM Efficiency equation

ATM/AAL5 have the following overhead.

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

At AAL5 layer, 8 bytes of trailer will be added to each IP packets. Then each ATM cells have fixed size (53 bytes) 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:

$$n = \left\lceil \frac{S+8}{48} \right\rceil \quad (10)$$

And the efficiency, E is:

$$E = \frac{S}{n \times 53} \quad (11)$$

AAL5 encapsulation into ATM cells only have one mode; padding mode. Section packing mode is not defined in ATM/AAL5 standards and therefore all the leftover spaces will be padded with padding byte.

6. ANALYSIS METHODS

In this paper we present analysis based on the theoretical values and also based on real Internet traffic patterns. We divide the analysis into 2 modes; padding mode and section packing mode.

Theoretical analysis is based on the equation introduced in chapter 4. We do analysis for MPE and ULE for IP packet encapsulation into MPEG-2 cells and also ATM/AAL5 for encapsulation into ATM cells used as default in DVB-RCS's return link. For comparison purpose, we also perform analysis for Ethernet, the well known encapsulation method for IP packets.

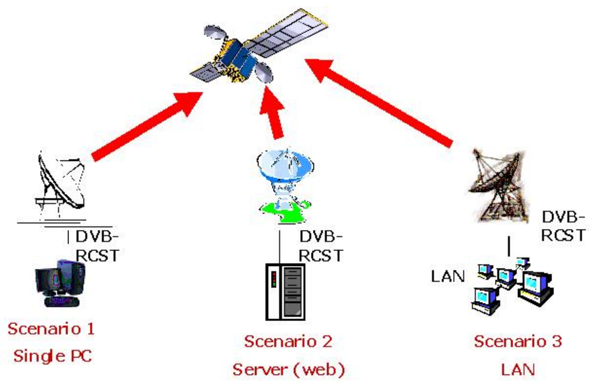

Figure 7: Real traffic analysis scenarios

For real traffic, we do analysis based on 3 different scenarios; single personal computer (PC), web server and local area network (LAN) shown in figure 7. These scenarios have different traffic pattern and therefore have different efficiency characteristic.

In section packing mode, we assume that we have enough buffer size to buffer the packet while waiting the next packet to be concatenated together. We also assume that if there is leftover space in a TS cell, the cell will wait long enough until the next section layer packet arrived. The objective is to fully eliminate overhead introduced by padding bytes.

7. RESULTS

We divide this section into four part; result for theoretical values, result for scenario 1 (single PC), result for scenario 2 (web server) and finally result for scenario 3 (LAN).

Within each part, there will be 2 kind of results; result for padding method and result for section packing method.

7.1 Theoretical values

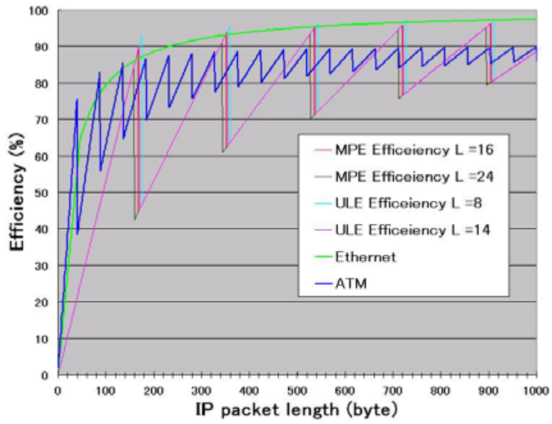

Figure 8. Theoretical values- padding

Graphs in figure 8 and figure 9 have the same axis. Y axis denotes the efficiency percentage and X axis denotes the IP packet length. Meanwhile the colored lines show the efficiency percentage for each encapsulation methods corresponding to the certain IP packet length.

Figure 8 present efficiency graphs in padding mode meanwhile figure 9 present the efficiency in section packing mode. Both figures show efficiency percentage for MPE encapsulation with total overhead, L equal to 16 and 24, graph for ULE with L equal to 8 and 14 and graph for ATM/AAL5 efficiency. For comparison we also include Ethernet efficiency graph.

In padding mode as shown in figure 8, since all the left over space have to be padded and consequently introduce additional overhead, MPE and ULE efficiency seems to be equal except for certain small range areas. It is because although ULE have much smaller header overhead, padding bytes overhead will mostly compensate the smaller header and make the total overhead same as MPE encapsulation except for certain IP packet length where it is can perfectly fit into TS cells without or less padding bytes.

For small packets (around 180 bytes and below) ATM encapsulation seems to have better efficiency comparing to Ethernet, MPE and ULE. MPE and ULE relatively have bigger oscillation compare to ATM. ATM oscillation will converge into around 90% efficiency when IP packet size is big enough. Meanwhile, MPE and ULE have better performance in certain areas compare to ATM but the areas' range is very small and will return to much smaller efficiency area just after the highest efficiency point.

Meanwhile figure 9 shows efficiency graph in section packing mode. Since no overhead is introduced by padding byte and packet concatenation is allowed, all the lines tend to be very smooth. However since ATM can't utilize section packing, ATM's graph in this figure is exactly the same with ATM's in figure 8.

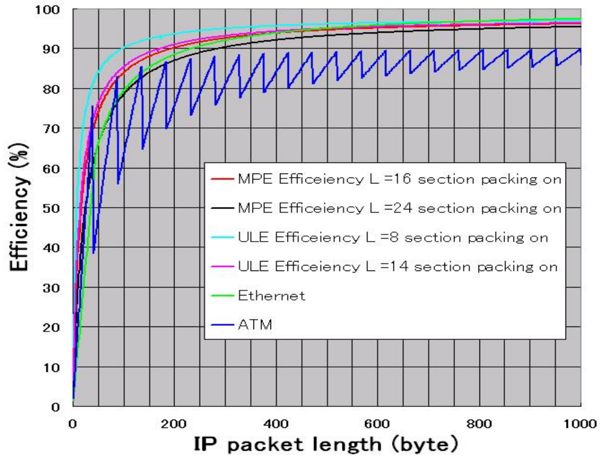

Figure 9. Theoretical values- Section packing

In this graph we can see ULE has better efficiency compare to MPE. Both ULE and MPE are better than Ethernet for small packets specifically below around 800 bytes for ULE and around 400 bytes for MPE. If IP packet is long enough, MPE and ULE efficiency converge to around 95%. It is 5 % better compare to ATM encapsulation.

7.2 Real Traffic

Figure 10 shows the legend used in graphs for figure 11 to 19. MPE and ULE efficiency are calculated using the minimum total overhead; 8 for ULE and 16 for MPE.

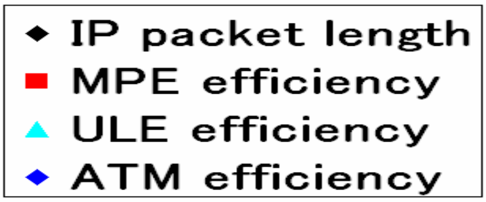

Figure 10. Legend for figure 10 to 19.

Figure 11 to 20 shows the result for real traffic analysis based on 3 different scenarios stated above. All the graphs have the same XY axis. X axis denotes time in second started from observation time. Y axis in the left denotes the IP packet length produced in each scenario at a certain time represented by black points. Meanwhile Y axis in the right represents the efficiency (%) and is used to plot efficiency points, correspondent to each IP packet produced.

7.2.1 Scenario 1

Figure 11 shows encapsulation efficiencies in scenario 1 in padding mode while Figure 12 show the result in section packing mode. IP packets produced in scenario 1 tend to be small in size. Therefore when in padding mode, ATM encapsulation proved to be the best among the others. Average ATM efficiency is around 55 % while MPE and ULE have average around 33%. From figure 10 we can see most of time, MPE and ULE efficiency is plotted in the same place make it hard to distinguish their points. It is because as stated before, in padding mode MPE and ULE have the same efficiency most of the time.

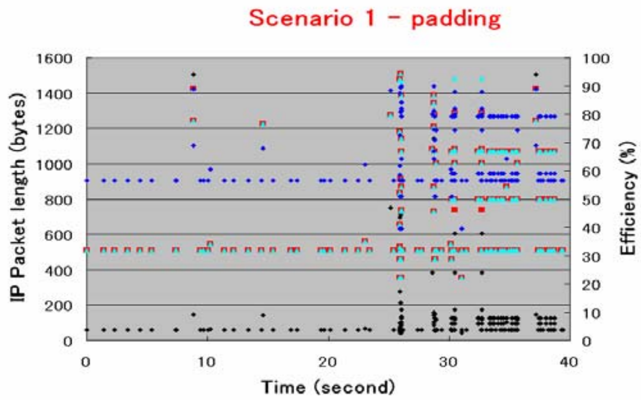

Figure 11. Scenario 1- padding

However in section packing mode, as we can see from figure 12, ULE and MPE have higher efficiency compare to ATM that did not have section packing mode. ULE is relatively has higher efficiency then MPE but the difference is small.

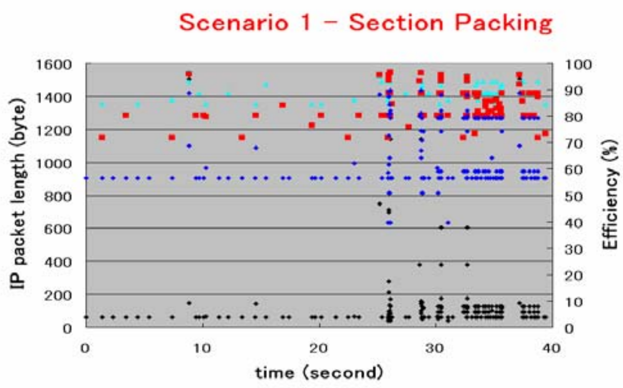

Figure 12. Scenario 1- section packing

7.2.2 Scenario 2

Figure 13 to 16 shows the result for scenario 2. In this scenario IP traffic seems to have more big size IP packets since it is the traffic from a server which provides contents to the users.

Figure 13 shows ATM efficiency in this scenario for padding mode. ATM efficiency has bigger range than scenario 1; from 60% to 90 %. When comparing to MPE and ULE in figure 14, we can see ATM have higher efficiency overall. MPE and ULE visually have the same efficiency scattered between 30% to around 95%. The same efficiency percentage make MPE values (pointed with red color) hard to distinguish since they share the same place with ULE's points most of the time

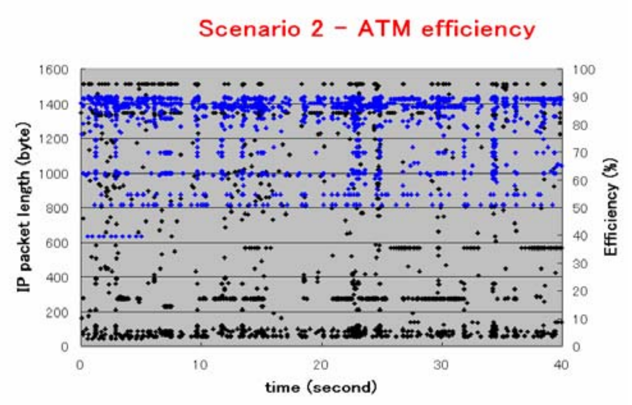

Figure 13. Scenario 2 – ATM efficiency

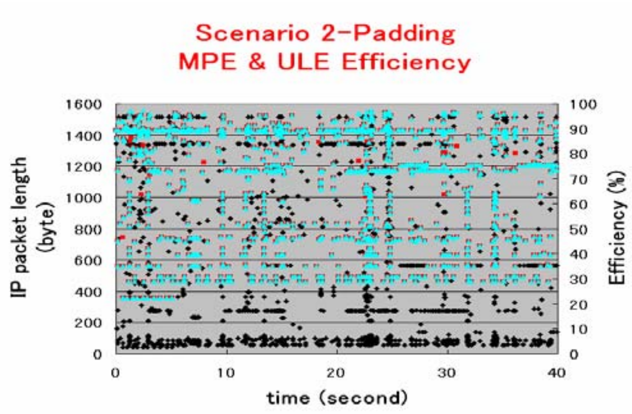

Figure 14. Scenario 2 – MPE & ULE efficiency in padding mode

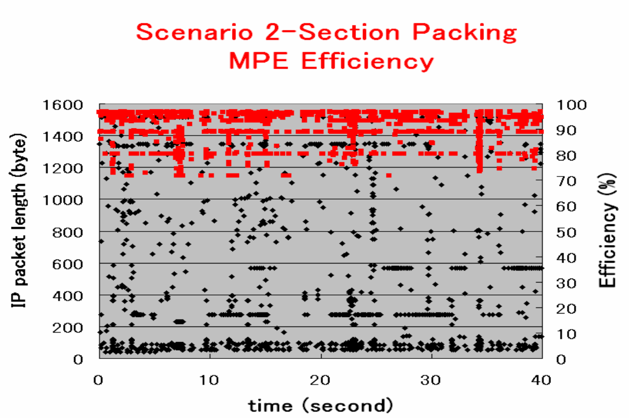

Figure 15. Scenario 2 – MPE efficiency in packing mode

Meanwhile in section packing mode as shown in figure 15 and 16, MPE and ULE obviously have much higher efficiency compare to ATM. MPE's average efficiency is around 90 % and ULE's is around 93%. Although the different is small, ULE obviously is better that MPE in section packing mode.

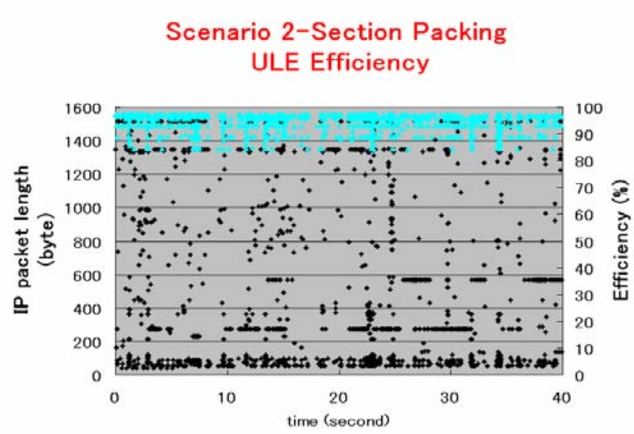

Figure 16. Scenario 2 – ULE efficiency in packing mode

7.2.3 Scenario 3

Figure 17 to 20 shows the result for scenario 3. From the graphs we can see scenario 3 have the heaviest traffic compare to other scenarios.

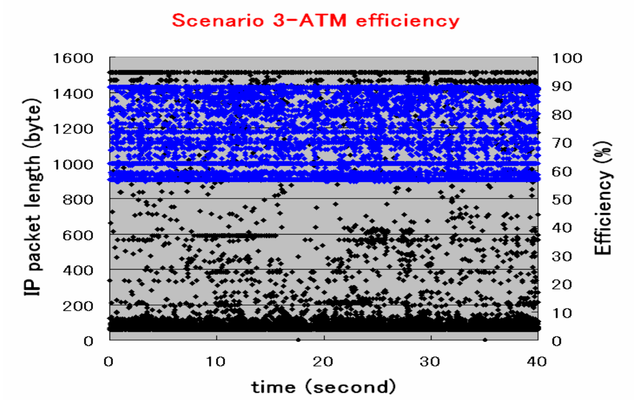

Figure 17. Scenario 3 – ATM efficiency

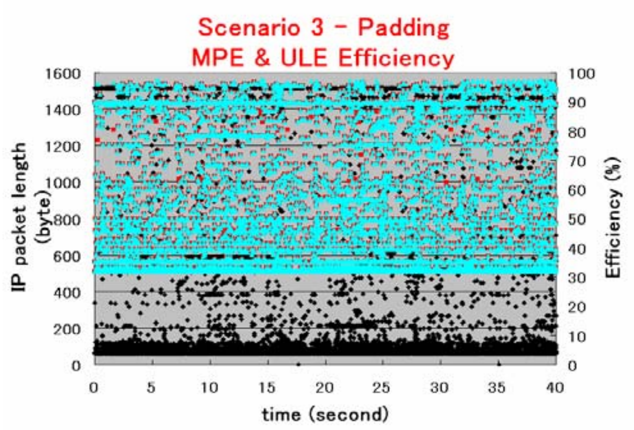

Figure 18. Scenario 3 – MPE & ULE efficiency in padding mode

Figure 1 7 shows ATM efficiency in scenario 3. As we can see majority of the packet size is relatively small packets (below 150 bytes) but there are also many IP packets with size around 1500 bytes. ATM efficiency can be seen scattered around 60% to 90 % with average around 75 %. It has relatively similar characteristic with scenario 2. Figure 18 shows MPE and ULE efficiency for padding mode. The efficiency percentage scattered from 30% to 95% with average efficiency at around 60%. The distribution is bigger than ATM thus we can say that MPE and ULE have less stable efficiency compared to ATM in padding mode.

However in section packing mode shown in figure 19 and 20, as expected MPE and ULE has better performance compare to ATM. MPE efficiency distributed between 70% and 97% and have average at around 85%. Obviously ULE is better than MPE. ULE efficiency scattered less and have average at around 92%, 5% higher that ULE.

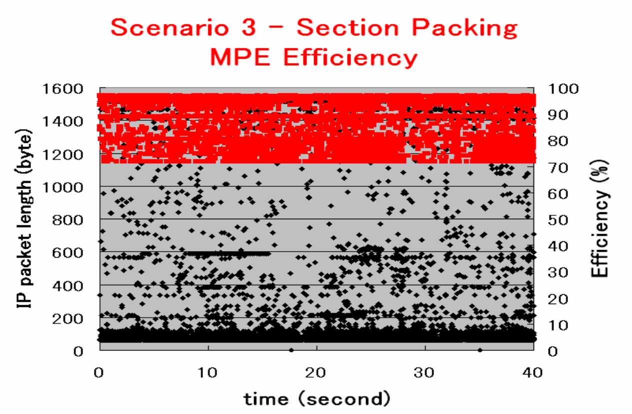

Figure 19. Scenario 3 – MPE efficiency in packing mode

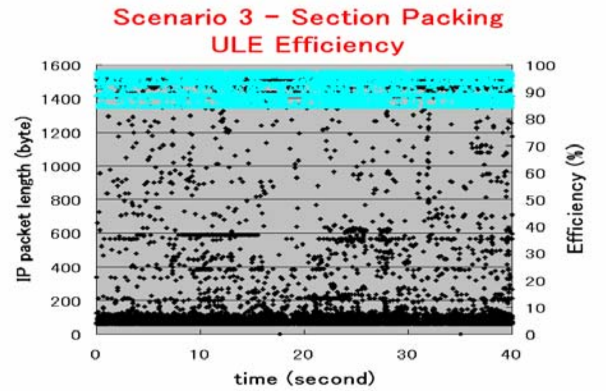

Figure 20. Scenario 3 –ULE efficiency in packing mode

8. CONCLUSION

From the result we can see each scenario have different traffic characteristic and therefore have different encapsulation performance. However overall we can conclude ATM encapsulation has better performance when comparing to MPE and ULE in padding mode. MPE and ULE have the same efficiency in most of the time in padding mode since smaller total header overhead in ULE is compensate by padding byte overhead. Therefore although ULE have smaller header then MPE, it has similar efficiency with MPE unless section packing mode is used.

However in section packing mode, MPE and ULE have much better efficiency compare to ATM encapsulation that doesn't have section packing mode defined. In this mode ULE have around 5% better efficiency compare to MPE as seen in results for both scenario 2 and scenario 3 although total overhead for ULE have been reduced 50% from total overhead in MPE.

Section packing modes have much better efficiency compare to padding mode. But it is expected to have higher jitter and delay because of having to wait next packet to arrive to be concatenated together.

When the traffic is light like scenario 1, it is wise to consider ATM usage since the jitter introduced by section packing mode in MPE/ULE is high. This can be seen from figure 11 where the points for MPE and ULE efficiency seems to be sparsely distributed compare to incoming IP packets. Meanwhile in scenario 2 and scenario 3 where the traffic is heavier, the sparse distribution is not obvious. It is because the traffic is heavy and the incoming packet doesn't have to wait long before the next packet arrives.

However jitter and delay properties introduced by section packing mode have to be analyzed thoroughly before anything can be concluded. In our future work we plan to perform jitter and processing delay analysis introduced by section packing mode. After we have the result we will examine traffic characteristic for all scenarios before proposing packing method best suited for each scenarios to reduce jitter and enhance efficiency.

9. REFERENCE

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

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

[3]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, Sept. 1997.

[4] H. D. Clausen, H. Linder, and B. Collini-Nocker, "Internet over direct broadcast satellites," *IEEE Commun. Mag.*, pp. 146–151, June 1999.

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

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

[7] G. Fairhurst, and B. Collini-Nocker, "Ultra Ligthweight Encapsulation (ULE) for transmission of IP datagrams over MPEG-2/DVB networks", IETF, Internet draft, draft-ietf-ipdvb-ule-01.txt, May 2004.

本論文は、東京大学新領域創成科学研究科基盤情報学輪講（2004/12/03）資料として執筆された。