The Effect of IPv6 Packet Size on Implementation of CRC Extension Header

Supriyanto
Electrical Engineering Department
University of Sultan Ageng Tirtayasa (UNTIRTA) Indonesia

Iznan H. Hasbullah
National Advanced IPv6 Centre
Universiti Sains Malaysia (USM) Malaysia

Rahmat Budiarto
School of Computer Sciences
Universiti Sains Malaysia (USM) Malaysia

***Abstract*—The traditional TCP/IP stack requires verification and regeneration of the CRC code in every router to detect transmission errors. For IPv6 packet transmission over high speed networks, this task is redundant because of the very rare transmission error and thus introduces unnecessary latency. The CRC Extension Header (CEH) was proposed to eliminate the unnecessary duplication of error detection. Since IPv6 packets are transmitted over Ethernet, the packet size ranges from 1280 up to 1500 bytes. In the near future, the size will increase with the advance of underlying technology such as 100 Gigabit Ethernet. This paper studies the effect of various IPv6 packet sizes with the implementation of CEH on the processing time and network latency. Experiments were done by transmitting various IPv6 packet sizes over high speed networks. The results showed that higher packet sizes increases processing time and network latency.**

***Index Terms*—Routing, TCP/IP, IPv6, error correction.**

I. Introduction

ROUTERS in a computer network have the important role in the packet forwarding process in order to ensure that each packet reaches its correct destination. An IPv6 packet transmitted through the Internet has to pass through many routers along the network. A router can be defined as three of the five layers of the TCP/IP stack, namely the Physical layer, the Data Link layer and the Network layer. In general, a router needs to decide to which node a packet has to be forwarded next after the IPv6 header processing are done. The decision is made in the Network layer of the router. However, before the packet reaches the Network layer, it must pass through the error detection process in the Data Link layer. The Data Link layer always verifies the CRC code attached with the frame received to ensure that the frame from the previous hop is free from transmission error. Only the correct frame will be delivered up to the Network layer for forwarding and hop limit updating.

The Network layer within a router on an IPv6 network does a simpler task compared to the network layer of a IPv4 router. In an IPv6 router there are no error detection processing (header checksum checking) and fragmentation. In addition, the size of an IPv6 header is fixed. This means that IPv6 packets are processed faster on router. However, the Data Link layer of the router not only needs to perform CRC calculation for the received frame but also for each frame that needs to be forwarded to the next hop. On a large network that with numerous routers this double CRC calculation will be repeated in every router, as shown in Figure 1.

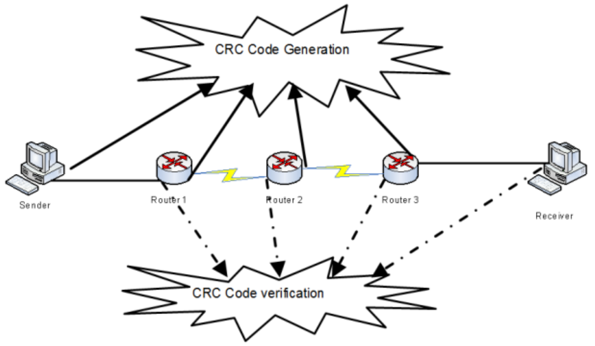

Figure 1: CRC Code Generation and Verification

Figure 1 shows a small network with three routers as intermediate nodes. In this network, there will be eight CRC calculations: four CRC code generations and four CRC code verifications. For IPv6 packet transmission over high speed networks which has an extremely small probability of transmission error, those repeated tasks are unnecessary. This is more so in the case of a reliable medium such as fiber optic which is not influenced by either the magnetic field or electric field. Errors within this kind of medium will be infinitesimal that practically we can say it has a near zero probability of occurring.

The IPv6 protocol has been designed with features that provide better performance than the former IPv4 protocol,

Manuscript received October 26, 2010. This work was supported by the Ministry of National Education of the Republic of Indonesia.

Supriyanto is with the Electrical Engineering Department, University of Sultan Ageng Tirtayasa, Cilegon, Banten, Indonesia (e-mail: supriyanto@ ft-untirta.ac.id).

Iznan H. Hasbullah is with the National IPv6 Centre (NAv6) Universiti Sains Malaysia, Penang, Malaysia. (e-mail: iznan@nav6.org).

Rahmat Budiarto is with the School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia (e-mail: rahmat@cs.usm.my).

including the extensibility of IPv6 extension header. IPv6 allows the definition of a new extension header in the future without impacting the existing infrastructure. In this paper we propose a CRC Extension Header (CEH) as a new IPv6 extension to perform error control within the Network layer and eliminating error control from Data Link layer [1]. In the near future, the IPv6 packet size will be increased [2]. Implementation of a CEH for a larger packet size is still debatable. This paper attempts to understand the effect of varying IPv6 packet sizes on the implementation the IPv6 packet transmission over high speed networks.

II. Overview of CRC Extension Header

The CEH is a new extension header that is proposed to reduce the bottleneck in IPv6 packet transmission over high speed networks by eliminating the CRC verification and regeneration in each and every router, as shown in Figure 1. This idea is entirely different from the existing method that conducts error detection and correction in the Data Link layer of a router. This new mechanism does not require the Data Link layer to verify and regenerate the CRC code for error detection. However, the concept of CEH does not eliminate overall CRC code generation and verification during IPv6 packet transmission. Generation of CRC code is done in the sender's machine while the verification of CRC code will be done at Network layer of the destination node by processing the CEH inside the received IPv6 packet. Routers are only required to process IPv6 packet at their Network layer in the usual manner, which consists simply of a forwarding decision and hop limit updating.

There are three major rationales of the idea of the CEH: Firstly, we would like to optimally utilize one of the advantages of IPv6 features, specifically the IPv6 extension header. IPv6 offers the opportunity to develop new extension headers for IPv6 packet transmission improvement in the future. Furthermore, it is not limited to 40 bytes only. In addition, applying a new extension header within an existing network will not disturb current IPv6 network operations. Secondly, within new high speed networks medium — such as fiber optic — the probability of transmission errors occurring is very small. This allows the error detection function in higher layer to be bypassed. Thus, the duplication of CRC calculation could be eliminated from the Data Link layer. Thirdly, the high transfer rate of current underlying technology, especially gigabit Ethernet, allows the transmission rates of more than 100 Gbps. Eliminating the error detection process in the intermediate nodes will decrease round trip delay from the sender to receiver. Hence, low latency of IPv6 packet transmission over high speed networks could be achieved.

Since all the existing extension headers already have specific functions, designing a CEH cannot make use of any existing extension header. Therefore we need to define a new one. The aim is to still utilize the advantages of the current error control but at the same time avoid its weaknesses. The approach uses the widely deployed CRC-32 error detection mechanism to detect transmission errors. To obtain an optimal result, we select an appropriate generator polynomial to be implemented in the CEH [3]. In terms of error correction, the CEH applies retransmission procedure to overcome erroneous packets, such as *Automatic Repeat reQuest* (ARQ). Details of the CEH format and mechanism design will be presented in the next section.

A. Format of IPv6 with CEH

Currently, there is no standard format for IPv6 extension headers. The existing IPv6 extension headers have different formats compared to one another. Thus, the format of the CEH follows the draft of the generic IPv6 extension header (GIEH) proposed in [4]. The CEH is placed after the destination address field of the IPv6 header, as shown in Figure 2. It has three main fields: the *Next Header* field, the *Reserved* field and *CRC-32 code* field. The next header is an 8-bit indicator to point to other extension headers following the CEH or upper layer data. As a new extension header, the CEH has not obtained specific number from Internet Assigned Numbers Authority (IANA). The Reserved field is allocated for future use instead of specific extension header type. The CRC-32 code field is the important field for the CEH. It contains the CRC code generated by the sender machine. This code is used only by the destination node for verification.

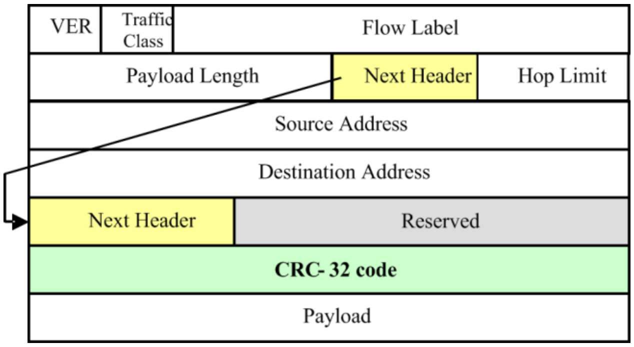

Figure 2: IPv6 Packet with CRC Extension Header

B. CEH Generation

The CRC-32 code field is the most important field of the CEH. CEH generation is determined by CRC-32 code generation. The code is generated from entire IPv6 packet excluding hop limit field. The CRC-32 code generation requires a generator polynomial [5]. The generator polynomial used in the CEH was selected from three candidates, namely the generator polynomial standardized in IEEE802.3 [6], the generator polynomial proposed by Guy Castagnoli [7] and the generator polynomial proposed by Philip Koopman [8]. The test conducted showed that Castagnoli's CRC-32C is the most suitable generator polynomial implement for the CEH [3].

The generation process of the CRC-32 code follows the usual algorithm used in the Data Link layer, which is already adapted for the Network layer as shown in Figure 3 [3]. The sender generates a CRC-32 code and then inserts the code into the CRC-32 code field of the CEH. When the CEH is created in an IPv6 packet, the extension header field is first initialized to zero except for the next header field. A complete IPv6 packet includes a main header with next header value indicating the type of CEH, an extension header which is CEH and a payload. The packet is now ready for transmission along the network path until it reaches its destination.

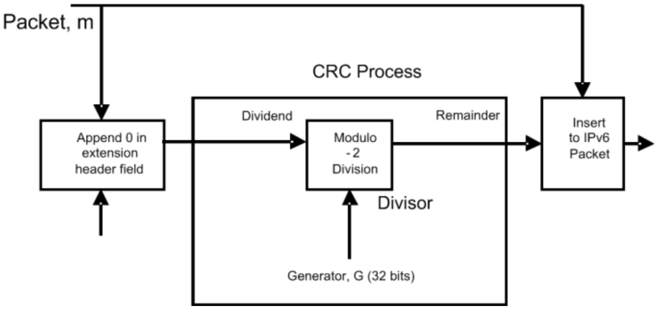

Figure 3: CEH Generation

C. *CEH Verification*

The verification of CEH will be done at the final destination by following the algorithm depicted in Figure 4 [3]. Once the receiver receives an IPv6 packet from previous node, it will check the next header field of the packet. When the value indicates there is a CEH inside the packet, it extracts the CEH and generates a new CRC-32 code from the entire IPv6 packet (omitting the hop limit field). The CRC-32 code generated by the receiver will be compared to the CRC-32 code found inside the IPv6 packet, which is located within the CRC-32 code field of the CRC extension header. If both CRC-32 code the same, then there is no transmission error in the IPv6 packet received. The receiver then forwards the packet into higher layer. Otherwise it will discard the packet and wait for retransmission.

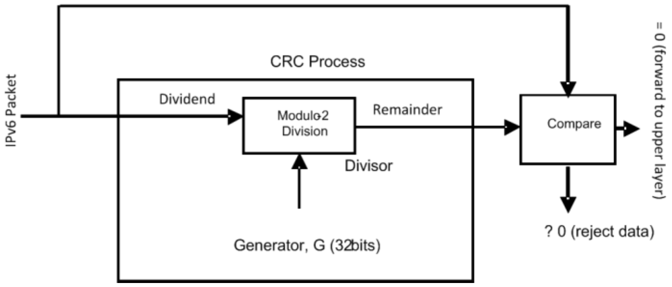

Figure 4: CEH Verification

III. Experiments

The CEH is a new concept for error detection for the entire IPv6 packet which is done at the Network layer. To verify the acceptability of the new concept we performed an experiment using the network topology depicted in Figure 5. The experiment has two scenarios: the transmission of IPv6 packets with the CEH for error detection in Network layer and the transmission of IPv6 packets with Frame Check Sequence (FCS) for error detection in Data Link layer. The first experiment simulates the new concept, while the later simulates the existing system. The two scenarios yielded an adequate data to justify the performance gain of the new error detection approach compared to the current error detection.

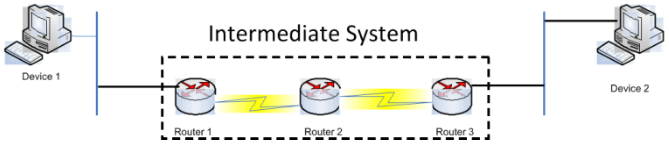

Figure 5: Network Topology for IPv6 Packet Transmission

The topology in Figure 5 has five nodes: Device 1 as a sender, Router 1, Router 2, Router 3 and Device 2 are the receivers. All nodes are PC computers equipped with Core 2 Duo processors that were configured as a mini IPv6 network. The sender is an end system installed with an IPv6 packet generator program written in JAVA that is able to generate both IPv6 packet with the CEH and IPv6 packet without the CEH. Router 1, 2 and 3 represent the intermediate system of the network whose task is to forward the packets.

In the first experiment, the routers simply forwarded the IPv6 packets with the CEH rather than checking each and every packet for error detection. In the second experiment the intermediate system performs error checking for each packet before forwarding the packet to next node in the path. The last node is the receiver which is a PC installed with the same program as the sender to verify all packets received.

The sender generates an IPv6 packet by adding the IPv6 main header to a TCP segment received from the upper layer. The Network layer then generates the CRC-32 code from both the IPv6 header and the payload using algorithm in Figure 3. The CRC code obtained is then inserted as the CRC extension header into the IPv6 packet. In the first scenario the Data Link layer just adds the link layer header. There is no trailer in the Data Link layer frame that usually performs the error detection task. The frame without frame check sequence (FCS) generated by the sender is then transmitted through the experiment network.

When the IPv6 packet reaches the intermediate nodes (namely Router 1, 2 and 3) no CRC code calculation will be performed by the routers for error detection. The ingress frame is just verified for its frame header by the incoming port of router and passed to the Network layer to determine the next path (forwarding process). The egress frame also does not have a Data Link layer trailer. It simply obtains the link layer header.

The only node that will verify the IPv6 packet is the destination node (receiver), as indicated by destination address field of the IPv6 packet. The verification follows algorithm in Figure 4 to check whether the packet received is free from error. Thus, the receiver node's task is to receive the packet in Data Link layer, release the Data Link layer header without computing the CRC code, and then to deliver the packet to the Network layer for verification.

In the second experiment which is a simulation of the current error detection mechanism, the sender generates an IPv6 packet without the extension header. The packet is then encapsulated by Data Link layer by adding Data Link layer header and trailer. We refer to this frame as the IPv6 packet with FCS. Each intermediate node will receive the packet and process it as usual. They process the Data Link layer header including CRC code computation to detect transmission errors for the corresponding hop. Only the packets that are verified as free from error will be delivered up to Network layer for forwarding process without processing any extension header. Before the IPv6 packet is forwarded to the next hop, the packet will be encapsulated again in Data Link layer of the outgoing port of the router including CRC code regeneration. The receiver node in the second experiment captures the IPv6 packet that was addressed to it and then verifies the packet for error detection. The correct packet will be passed to Network layer. Otherwise, it will discard the erroneous packet and wait for retransmission.

In order to know the effects of IPv6 packet size with the implementation of the CEH, we measure two main metrics: processing time and error detection capability. Processing time is the packet processing time in each node including the CRC-32 code computation, packet generation and packet verification. Here it is very important to know the network latency, which is the most important aspect of network performance with various packet sizes. With regards to error detection capability, it is also important to know the new error detection capabilities to detect transmission error on varying sizes of IPv6 packets.

IV. Results and Discussion

The sender's processing time is the time required to generate an IPv6 packet with the CEH using the algorithm in Figure 3. While the receiver's processing time is the time required to verify the IPv6 packet received and the CEH verification using the algorithm in Figure 4. Experiments of IPv6 packets with the CEH, using varying packet sizes from 64 bytes to 1500 bytes yielded the results as shown in Table 1. The table shows the relationship between the sender's processing time and packets size of the two error detection mechanisms on TCP/IP, namely the transmission of IPv6 packets with the CEH as a proposed mechanism and the IPv6 packet transmission with FCS (existing system) as a comparison. This is done to learn about the impact of packet sizes on the performance of the two error detection mechanisms.

Table 1: Correlation between Processing Time and Packet Size at the Sender

Size	64	128	256	512	1024	1280	1492	Average
CEH	0.689	0.718	0.772	0.883	1.039	1.067	1.133	0.900
FCS	0.682	0.686	0.778	0.788	0.834	0.876	0.909	0.793
Δ	0.007	0.132	0.006	0.105	0.205	0.191	0.224	0.107

Both the IPv6 packet with CEH and the packet with FCS require a certain amount of time to process at the sender as well as at the receiver. The processing time of the packet at the sender increases with the increase in packet size. This is because CRC code generation was done byte-per-byte of the packet. Thus more bits in a packet means longer time to perform processing of the IPv6 packet. Table 1 also shows that the average of processing time for all packet sizes for the transmission of IPv6 packet with CEH is 13.5% higher than the transmission of IPv6 packet with FCS. This is due to the fact that the IPv6 packet with CEH generation at the Network layer (shown in Figure 3) is more complex compared to FCS generation at the Data Link layer. The process to generate the CRC code from the whole IPv6 packet requires firstly the exclusion of the hop limit and then secondly the insertion of the CRC code as the extension header. In contrast the FCS generation simply divides the whole frame with the generator polynomial and followed by appending the CRC code in the last part of the frame.

Similar to the sender, the receiver's processing time of the IPv6 packet which is listed in Table 2 showed that the processing time is affected by the packet size. The larger the packet, the higher it's processing time. On average the receiver's processing time of IPv6 packet transmission with CEH is 16.3% higher than IPv6 packet transmission with FCS. Note that the percentage value is higher than in the sender. This is because the average processing time for both IPv6 with CEH and IPv6 with FCS at the receiver is less. In other words, the processing time of IPv6 packet at the receiver is faster than at the sender.

Table 2: Correlation between Processing Time and Packet Size at the Receiver

Size	64	128	256	512	1024	1280	1492	Average
CEH	0.568	0.568	0.583	0.600	0.642	0.662	0.670	0.613
FCS	0.495	0.491	0.512	0.514	0.548	0.566	0.565	0.527
Δ	0.073	0.077	0.071	0.086	0.094	0.096	0.085	0.086

The processing times of IPv6 packets with the CEH as listed in Table 1 and Table 2 is based on the algorithm in Figure 3 and Figure 4. Even though the process of generating the CEH at the sender and at the receiver is similar, the sender has to generate the IPv6 packet first before it can generate the CEH. In contrast the receiver simply has to receive IPv6 packets without needing to generate any packets.

However, note that the transmission of IPv6 packets exploiting the CEH as an error detection method in the Network layer has eliminated the need for CRC code calculation and regeneration in every router. This resulted in the faster processing of IPv6 packets in every router in the network, where a router only processes the packets at the Network layer (routing and forwarding). Hence, although processing time at the end systems (sender and receiver) is higher, the total network latency of IPv6 with CEH transmission is visibly lower than IPv6 with FCS. Figure 6 shows the correlation between network latency in milliseconds (ms) and packet sizes in bytes. In the figure, network latency of the IPv6 packet with CEH transmission is 68 % lower than IPv6 with FCS. The network latency as shown in Figure 6 increases when the packet size increases. Compared to FCS, the increasing network latency of IPv6 with the CEH is lower as shown by the two graphs in Figure 6.

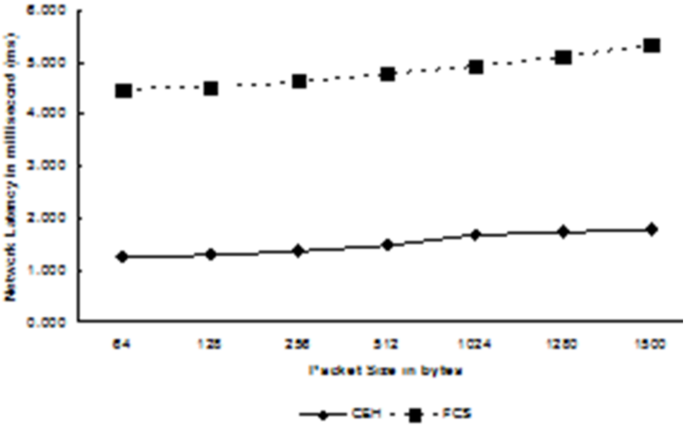

Figure 6: Network Latency

Another measurement is the error detection capability. Error detection capability of an error control mechanism is the ability to catch transmission errors during transmission from the sender to the receiver. Recent high speed network technologies have made the occurrence of transmission errors very rare. Employing more error detection tools is inefficient and also redundant. However, we cannot eliminate the tools entirely because if an error occurred during the communications then data will be corrupted. Proposing to exploit the CEH as an error detection tool is the correct way. It can reduce the redundant error control process but at the same time it does not ignore the error.

Error is unpredictable. Thus an error control mechanism should be able to detect the error inside the packet transmitted and correct it. In order to know ability of CEH to detect transmission errors, we send erroneous IPv6 packet with the CEH. The result illustrated that all erroneous IPv6 packets sent are successfully detected by the receiver.

V. Conclusion

In this paper we have proposed a new concept of handling transmission errors by exploiting the IPv6 extension header. The new concept utilizes the CRC extension header (CEH) as an error control method at the Network layer and eliminates error control at the Data Link layer of the intermediate nodes. Typically, the extension header is generated by the sender and is processed by the receiver. Thus, the CEH that contains the CRC code is also processed by the receiver.

Using the CEH as a new IPv6 extension header for error detection at the Network layer increases the processing time of IPv6 packets both at the sender and at the receiver. The increase in processing time is influenced by the packet size. The higher the packet size the longer it will take to process the packet. However, our proposed new concept has eliminated error detection routine at the Data Link layer.

Elimination of error detection at the Data Link layer means eliminating CRC code computation and regeneration in each and every router. This reduces the total network latency on IPv6 packet transmission. The results show that overall network latency decrease by 68% compared with normal latency. It also reduces the Data Link layer frame size due to the elimination of 4 bytes of frame check sequence field. The proposed error control mechanism also shows good ability to detect transmission error inside the transmitted packet. The experiment showed that all IPv6 packets sent with errors are successfully detected by the receiver.

References

[1] Supriyanto, Raja Kumar Murugesan, Rahmat Budiarto, Sureswaran Ramadass, *CRC Extension Header (CEH): A New Model to Handle Transmission Error for IPv6 Packets over Fiber Optic Links,* Proceedings of IEEE Symposium on Industrial Electronics and Applications (ISIEA 2009), October 4-6, 2009, pp. 569 – 573.

[2] Supriyanto, Iznan H. Hasbullah, Rahmat Budiarto, *Exploiting IPv6 Extension Header to Handle Transmission Error for IPv6 Packets Over High Speed Networks*, Proceedings of International Conference on Advanced Computer Science and Information System, 7 – 8 December 2009.

[3] Supriyanto, Abidah M. Taib, Rahmat Budiarto. *Selecting a Cyclic Redundancy Check (CRC) Generator Polynomial for CEH (CRC Extension Header),* Proceedings of International Conference on Quality in Research, Jakarta. 3 – 6 August 2009, ISSN 114-1284 pp. 245 – 250

[4] S. Krishnan, et. al, *An uniform format for IPv6 extension headers,* Internet Draft IETF, 2008. http://ietfreport.isoc.org/idref/draft-krishnan-ipv6-exthdr/

[5] A.S. Tanenbaum, *Computer Networks*, Fourth Edition, Prentice Hall of India. New Delhi, 2006.

[6] ANSI/IEEE Standard for Local Area Networks, *Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications.* 1984. http://standards.ieee.org/getieee802/802.3.html.

[7] Castagnoli, G., Brauer, S. & Herrmann, M. *Optimization of cyclic redundancy-check codes with 24 and 32 parity bits*. IEEE Transactions on Communications, 1993, pp. 883-892.

[8] Koopman, P. *32-bit cyclic redundancy codes for Internet applications,* Proceedings. International Conference on Dependable Systems and Networks (DSN). 2002. http://www.ece.cmu.edu/~koopman/networks/dsn02/dsn

Supriyanto received the B.Eng. degree in Electrical Engineering from Brawijaya University Malang, Indonesia in 1999, and and M.Sc in Computer Science from Universiti Sains Malaysia (USM) in 2010. Currently, he is a lecturer and also department secretary in the Electrical Engineering Department at the University of Sultan Ageng Tirtayasa (UNTIRTA). He is a researcher in the Computer Network Laboratory at the university. His research interest includes computer networks especially, IPv6, IPTV over overlay network and wireless communication.

Iznan H. Hasbullah graduated with the B.Sc degree in electrical engineering in 1998 from Rensselaer Polytechnic Institute, Troy, New York. He is currently a Research Officer with National Advanced IPv6 Centre (NAv6) at University Sains Malaysia, Pulau Pinang. He is the Domain Head of HD Video Conference for IPv6 group under Next Generation Multimedia & Telemedicine Cluster at NAv6. Prior to joining NAv6 in 2009, he was an R&D Consultant with MLABS System Berhad (MLABS) seconded to JMCS Sdn. Bhd. to spearhead a project titled High Definition Video Conferencing for IPv6 Networks. He held the post of Chief Technology Officer (CTO) while with JMCS Sdn. Bhd. He was involved in two network security audit exercise commissioned by Malaysian Communications and Multimedia Commission (MCMC) in the year 2003 and 2004. He was also part of a team that brought Advance Military Mobile Conferencing System (AMMCS) to the Maritime Langkawi International Maritime & Aerospace Exhibition (LIMA'03) in 2003. His area of expertise includes multimedia conferencing application for computer supported cooperative work (CSCW), software architecture, Graphical User Interface (GUI), Component Object Modeling (COM), Microsoft Foundation Classes (MFC) and C++ Language.

Rahmat Budiarto received the B.Sc. degree from Bandung Institute of Technology (ITB) in 1986, and the M.Eng and Dr.Eng degree in Computer Science from Nagoya Institute of Technology in 1995 and 1998 respectively. Currently, he is an associate professor at the School of Computer Sciences at the Universiti Sains Malaysia (USM) as well as the advisor of Indonesia IPv6 Task Force. His research interest includes IPv6, network security and Intelligent Systems. He was chairman of APAN security working group.