Design of an Efficient and Robust Multimedia Gateway for Pervasive Communication

Fung Po Tso, Yufei Du and Weijia Jia
Department of Computer Science, City University of Hong Kong
83 Tat Chee Avenue, Kowloon, Hong Kong
{posco.cs, itjia}@cityu.edu.hk, yufeidu2@student.cityu.edu.hk

Abstract

In this paper, we present the design and implementation details of an efficient and robust multimedia gateway which enables ubiquitous multimedia wireless communication. Primarily, we studied current technological trend and realized that in order to support various network technologies and to handle high call traffic conditions, a robust and efficient gateway is required for the universal adaptability and transcoding. To achieve the performance improvement of the gateway, we focus on efficiently implementing the H.245, which is the main call handling and signaling protocol. In our method, sets of H.245 messages are compiled, used and stored in a table so as to be reused for further incoming calls. After implementing this procedure, we have successfully tested our 3G-IP gateway that is robust enough for smoothly handling up to one million concurrent calls. It is also experimentally verified that our gateway provides the feature of invariant call setup time even in high traffic condition. As ubiquitous communications among the heterogeneous networks are in demanding today and our gateway will play the key role in this area.

Key words - Universal gateway, universal communication environment, 3G-IP gateway, SIP, H.245.

1. Introduction

These days, mobile devices like iPods, PDAs, smart-phones, tablet PCs and high end mobile phones etc have become like inexhaustibly essential part in day to day life. It is because, these equipments provide not only the features like phone calling, music playing, photo and vide capturing, but also they do have incredible features like seamless accessibility and connectivity that enables the modern life to be pervasively connected to any equipment around him or to any point in this world. These advanced communication devices posses the capability of connecting through various network types such as, infrared, Bluetooth, cellular networks, traditional landlines and internet. On the mobile devices, the demand of multimedia applications such as video conferencing, VOD, IP TV, Mobile trading, on-line games, geographical and vehicular information systems etc have become a general trend in today's networked life style. Moreover, if we study the current technological trend, it is prominent that there is a high demand on the complete convergence and unification of the traditional telephone, mobile phone and the internet networks so as to provide any type of seamless services to subscribers on any device at any place, any time.

This ubiquitous communication environment among the heterogeneous networks is in demanding today. In this situation, a gateway is required for bridging these gaps between different network domains (e.g. PSTN and IP). As the ubiquitous communication is expected to grow rapidly and is also a major research direction and thus we believe our gateway will play the key role for this sort of applications.

Basically, communication networks can be classified into two categories, such as circuit switching network (CSN) and packet switching network (PSN). The telecommunication networks mainly use the circuit-switching technique due to its service reliability and provision of fixed bandwidth. On the other hand, the internet uses the packet-switching method in order to deal with the burst traffic pattern and working over a range of available bandwidth in the network.

Presently we observe that the mobile networks are in increasing demand of using packet switching network in order to meet their heavy demand of internet and multimedia applications. These bulky multimedia contents have eventually caused largely growing traffic load over both mobile phone and internet networks. Moreover, as mentioned earlier, cell phones are now being used to transmit and receive not only voice but also high quality real-time video and bulky data. From these observations we realized that in next generation networking and in 3G and beyond wireless networks, cross network multimedia service delivery will assume an increasingly important role. A multimedia gateway that facilitates establishing service sessions and delivery of

978-1-4244-1870-1/08/$25.00 (c) 2008 IEEE

multimedia services to users and terminals of differing capabilities is highly recommended. In this case, the supporting gateway must be giant, robust and efficient enough for call handling. Otherwise the gateway faces not only the loss of QoS, but also a severe hazard of performance bottleneck.

There have already been some practical developments of gateways [1, 2, 3] that successfully bridge the gaps among the traditional landline phone, 3G mobile and internet networks in order to fulfill many advanced application aspects. The ITU-T provides the standards H.245 [4], H.223 [5], H.324M [6] and 3GPP's 3G-324M [7] standards are the building blocks of an up-to-date multimedia gateway. There are some commercially available 3G gateways in the market. Radvision has developed a 3G gateway called Scopia [1]. The Scopia 3G video gateway supports video telephony as well as video streaming between 3G-324M based mobile handsets/devices and IP based video media servers. In the meantime, Tandberg and Dilithium have also developed similar gateways [2, 3].

During this research, we have developed our own 3G-324M-and-SIP based 3G-IP gateway in the City University of Hong Kong. We aim at developing a gateway which is robust enough to meet the challenge of handling large number of calls concurrently and at the same time to ensure a good QoS to all the subscribers. In our research, we have given emphasis on efficient implementation of the H.245 protocol so as to enhance the efficiency and robustness of the gateway that can smoothly handle high traffic condition with a satisfactory level of QoS.

2. Previous work

3GPP has adopted the H.324 with some modifications in codec and error handling requirements to create 3G-324M standard for 3G wireless networks. Adequate attention has been focused on 3G-324M in recent years. Some of those researches specifically concerned about implementation issues of this standard. Sanghyun Park et al [8] have implemented H.324M, however their simulation result involve serious performance degradation produced by the header because of corrupted flags. A.Basso and H.Kalva [9] have studied this limitation of 3G-324M for supporting streaming and messaging services and proposed a set of requirements that imply some extensions and clarifications of the standard to better support them. Some researchers focused their works in the ASN.1 messaging encoding. Rein Vesilo [10] proposed the design of ASN.l coders and compilers using recursive descent techniques from their theoretical basis in compiler theory with software engineering considerations. Since BER encoding/decoding have low efficiency and redundant codes; so, Qian Lv B.H. et al [11] have proposed a tree like method called "database definition" to store the necessary ASN.1 syntax information to increase the efficiency of ASN.1 message handling. Also Yuen M.C et al [12] have studied the drawbacks of tree approach and proposed the Single-step Direct Message Transformation (SDMT) for the optimization of tree-structured message processing in H.245 module. Meanwhile, Most of the works are about video quality improvement. Lee Yen-Chi et al [13] have proposed a cross-layer decoder design that efficiently recovers more video data in the presence of transmission errors for 3G-324M video telephony over WCDMA networks with supporting simulation results. Miki T. et al [14] described the error-resilient audio-visual coding, MPEG-4/GSM-AMR, and the terminal standard, H.324 mobile extensions for 3G mobile multimedia terminals. However, the performance improvement of a 3G gateway, especially in case of large number of calls handling, is seldom studied.

3. Background

According to ITU-T standards, H.245 is a control channel protocol capable of conveying information for multimedia communication. In voice and video telephony as well as in VoIP, this sub-protocol (of 3G-324M) is basically responsible for call initialization, setup and for continuing the conversation. This protocol manages information exchange through a set of predefined messages. Some members of this message set are vital for the call initialization and setup, while some are responsible for continuing the conversation. So any attempt to improve the performance of the H.245 procedure is nothing but to handle this message exchange process efficiently. As this protocol is responsible for the call setup and conversation continuation, by efficiently handling its message exchanges will eventually lead to shorter call setup time and better conversation quality. Also in high traffic condition, for handling large number of call concurrently and to provide good conversational quality (after all the QoS), H.245 is the only vital protocol which should be taken care of properly.

According to ITU-T X.691 recommendation [15], H.245 messages are initially in the form of ASN.1 and then they are converted into binary streams, when the call is initiated. After a requested terminal receives the bit stream, it reconstructs those messages back in to meaningful ASN.1 text and then it sends proper responses to react to the requester terminal.

In H.245 module, messages are defined in a tree-like structure as is depicted in Figure 1. This defines a general message type called, MultimediaSystemControlMessage (MSCM). MSCM further comprises four different types of special messages, namely, request, response, command and indication. A request message corresponds to a

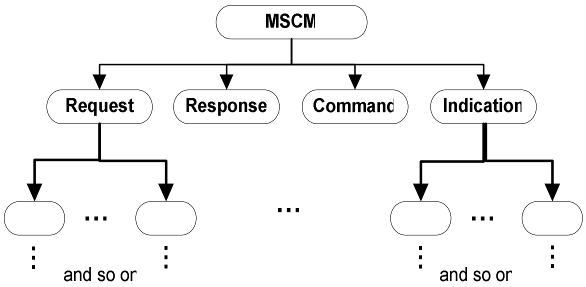

Figure 1. Hierarchical representation of the H.245 messages

specific action and requires an immediate response through a response message. A command message requires an action but no explicit response. An indication message contains information that does not require action or response.

3.1. Characteristics of H.245 messages in 3G system

The most common H.245 messages for a 3G video call are:

1. TerminalCapabilitySet
2. TerminalCapabilitySetAck
3. MasterSlaveDetermination
4. MasterSlaveDeterminationAck
5. VendorIndentification
6. MultiplexEntrySend
7. MultiplexEntrySendAck
8. OpenLogicalChannel
9. OpenLogicalChannelAck
10. MiscellaneousCommand:VideoFastUpdatePicture
11. RoundTripDelayRequest
12. EndSessionCommand:disconnect

Here only messages-7&8 are dynamic because their content depends on messages sent by remote terminals. Normally, message-1 to message-9 are only exchanged during call setup phase and message-10&11 are used to maintain the call, and finally conversation is terminated using 'EndSessionCommand:disconnect' command.

To present a clear idea about the details of a message and its sub-messages, let's take TerminalCapabilitySet as an example. The message, TerminalCapabilitySet is used by sending terminal to inform receiving terminal about its multiplexer capabilities and also about its supported media codecs etc. An example of the hierarchical message structure of TermnalCapabilitySet is shown in Figure 2. Here, we may observe that most contents can be kept unchanged since they describe the capability of a multimedia mobile terminal (as they are fixed for that specific terminal). However, there is a field called 'sequenceNumber' which is used to label instances of 'TerminalCapabilitySet' so that the corresponding response can be identified. If there appear multiple instances of 'TerminalCapabilitySet' with the same content, then only the 'sequenceNumber' filed can easily be changed dynamically.

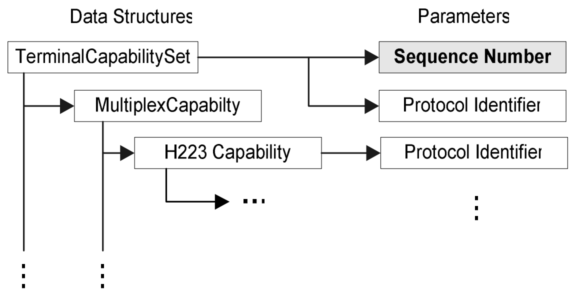

Figure 2. Message structure of TerminalCapabilitySet

3.2. Suggestion of a novel procedure

Traditionally H.245 messages are encoded in a chunk-by-chunk style [16]. Each chunk corresponds to a sub-message under the main message. For each chunk of input, encoding process must be called every time and the chunks are being encoded serially. So in this serial and individual encoding process for each and every message eventually leads to high cost of system time.

Also we note some important points here:

1. Each main message and its sub-messages are the same for all the individual calls.
2. From our experiment, we observe that some sub-messages under a main message are also repeated more than once.
3. Messages like TerminalCapabilitySet request are no doubt same for all calls. But still, the response is also very few in number; it is because, in reality, terminal capabilities are fixed and there are a few number of different mobile terminals produced by different brands. So for most calls, these requests just the same and responses are just little varied.

Hence, it seems that the earlier mentioned serial coding process is just unnecessary killing of system's CPU time. In this contrast, if there is a low-cost process serves the same objective of producing the same encoded output, that process is definitely worth adopting. In this scenario, with the motivation of improving the call handling efficiency of the 3G gateway, we would like to suggest a more efficient method on an experimental basis. In this method, we consider the possibility of reusing of a replicated encoded message set which is already compiled earlier for a previous call, unless the message data is changed. And if there is any change in some sub-messages, that part can easily be handled by dynamically

Table 1. Traditional approach for H.245 message encoding

Input	Process	Output
M1	Encoding	A
M2	Encoding	B
M2	Encoding	B
M3	Encoding	C
M1	Encoding	A
M1	Encoding	A

Table 2. Suggested table lookup approach for efficient H.245 message encoding

Input	Process	Output
M1	Encoding +Reuse	A
M2	Encoding +Reuse	B
M2	Reuse	B
M3	Encoding +Reuse	C
M1	Reuse	A
M1	Reuse	A

compiling that part and updating the returned value in the whole encoded message string. We describe the method in detail as follows.

Considering the programming level of message encoding, it is not necessary to encode the whole H.245 message for each call, rather the produced binary stream can be just updated with a minor changes in few bits. Here the precompiled message streams may be saved in a *lookup-table* and that can be accessed for each call instead of recompilation each time. Eventually it saves an appreciable amount of encoding compilation time on the fly.

A visual description of the above suggested reusable pre-compiled message-data set and its lookup table implementation method is compared to the traditional message encoding method is depicted in the above tables. Here we suppose that M1, M2 and M3 are H.245 messages and A, B and C are their encoded binary streams respectively. Table 1 shows the traditional approach for H.245 message encoding where each message is encoded individually. Table 2 illustrates our suggestion of improved approach for H.245 message encoding in which reuse of previous encoded bit streams is deployed along with a minor update by dynamic compilation process. Comparing Tables 1 and 2, we get exactly the same output but at least three extra encoding processes can be skipped. So this leads to saving of at least 50% compilation time and thus provides extra free system resources and time for better quality conversation in the post setup phase in this example.

Algorithm1: Table lookup message processing.
Input: H.245 ASN.1 messages m
Output: binary stream $B(m)$

1. initialize system
2. pre-compile a message set $M(m)$ and store into a table $T(m)$
3. for each i[th] message H.245 ASN.1 message m_i
4. if $m_i \in M(m)$ then
5. retrieve bit stream B(m_i) from table $T(m_i)$
6. if replacement needed then
7. replace changed filed(s) and return
8. *else* return the bit stream
9. else dynamically compile it and store it into table $T(m)$
10. end for
11. repeat step 3 to step 10

Figure 3. Table lookup message processing algorithm

4. Implementation details of lookup-table based message encoding

Basing on the above discussions and suggestions, here with we present the technical details of the *lookup-table* based message processing approach. An algorithmic representation is shown in Figure 3 and its process flow is illustrated diagrammatically in Figure 4. Basically this algorithm consists of two major functions. In the first process, the system is initialized and then a set of messages, i.e., M(*m*) are (pre)compiled and followed by storing the return data (bit-streams) into a table. Here we defined M(*m*) as a set of frequently used messages. Each time, when system comes across a new message that is not in the set, then the message is added automatically to the database in the table for future use. In the second part of the algorithm, simply message matching is performed. The input of this algorithm is a single H.245 message, m_i (m_i is the i[th] message to be encoded), and the output is it's corresponding bit stream B(m_i). Looking from the programming point of view, during the system initialization, a H.245 message set is compiled into bit streams and then is stored in a table, *T(m)*. For each H.245 message m_i (to be encoded), it is looked for in the precompiled message database (i.e., table *T(m)*), if a match is found, then it is retrieved and returned. After that, the returned bit stream B(m_i) is further checked whether a replacement is needed, if no, B(m_i) is returned directly, if yes, then the changed field(s) is dynamically compiled and replaced and returned. However, if the message m_i is not found, then it is just compiled and returned

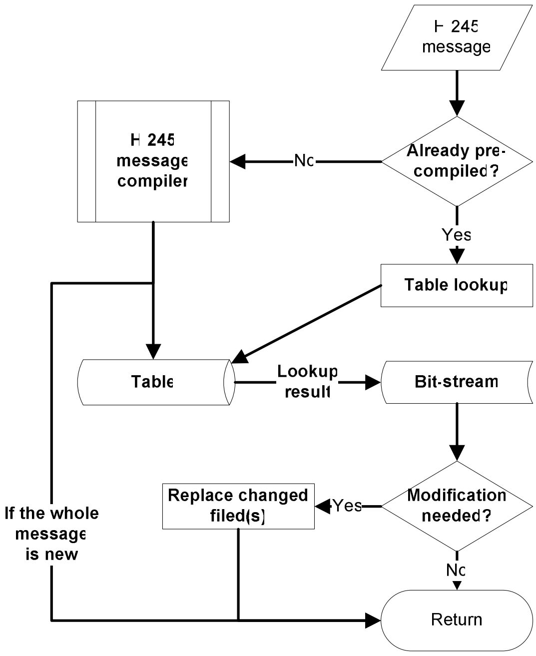

Figure 4. Table-lookup process flow

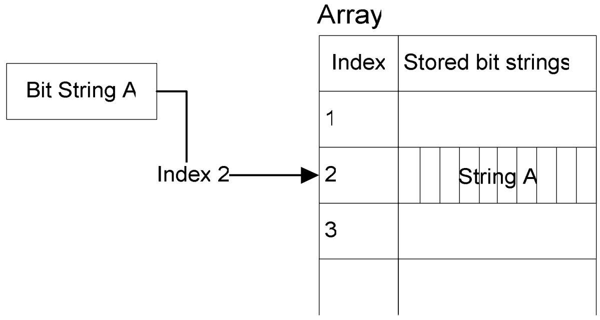

Figure 5. Array implementation for the lookup-table creation and management

dynamically and is also stored in the database for further use (for next incoming calls).

One of the key issues of *lookup-table* approach is to find the efficient way to manage the table data storage and retrieval system. We propose the table should be managed in an *index-based* fashion. Each encoded message string is saved as an array in that table with an index number. So in the retrieval process, each entity is returned in terms of its reference number (integer). This approach is used because, it is much less time consuming way to locate and return each table entry as compared to the traditional algorithms involving special key generation. Thus, an appreciable amount of time can be saved for data retrieval in the table and eventually leading to a big gain in the overall system performance. The Figure 5 describes this procedure pictorially.

5. Performance evaluation and discussion

In this section, we evaluate the performance and efficiency of our method through experimental results. As we have already mentioned earlier, we have developed our IP-3G bidirectional call handling gateway. Our system leverages PC to PSTN/PC and vice versa video, voice and text chatting/conferencing provision.

We have successfully tested our 3G-gateway for the compatibility test after embedding the currently proposed performance enhancement algorithm into it. We made phone calls to different brands/makes of 3G handsets and also to the Dilithium 3G Network Analyzer (model no. FXPAG-P42G). All attempts of 3G video calls were setup and conversations were hold successfully. It proves that the proposed algorithm is fully compatible with the existing 3G protocol as well as with all commercially available 3G handsets/mobile-devices too.

Although the gateway is verified to be fully compatible and interoperable with SIP and 3G-324M protocol stack, we have still have some difficulties in subscribing a T1 line from local operator. Due to this constraint, we can not conduct our experiments in large scale wise in real environment. For evaluating the performance of our gateway, we have carried out a systematic procedure of experimentation. During this experiment, in order to track the call handling efficiency, we added two test modules namely “MessageFeeder” and “DataCollector” to our gateway. MessageFeeder is responsible for selecting a large number (say 10000 at a time and then iteratively increasing in arithmetic progress) of H.245 message sets to be input to the 3G gateway. And the DataCollector is employed to record the time at the start of encoding the messages and the time corresponding to bit stream is returned at the end of the encoding process. It is practically obvious that these two factors can be compared so as to let us study the performance and efficiency of our 3G gate way in the view of handling high traffic of incoming calls.

In our experiment, the number of messages we chose are ranging from 10,000 to 1,000,000. Here we take care of another important factor. In principle, the longer the message length, better will be the performance of our gateway. This is because, the longer time for compilation is saved through our message handling method. But in practical, some messages are short length while some are long. In order to keep balance in the testing process, we have taken consideration of both these types of message structures and have chosen samples randomly, so that the collected data in the respective tables are nearly the average values.

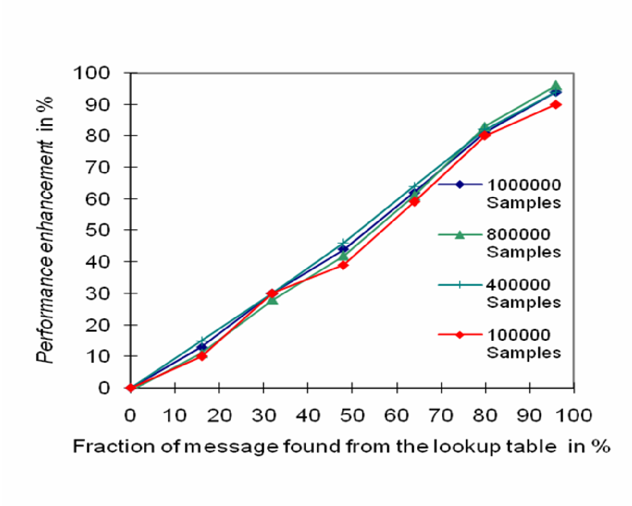

Figure 6. Experimental result of message processing in table-lookup procedure

The proposed method is compared with the method that uses the traditional way i.e., if there is a message, then encode it and output it. The traditional procedure is time consuming as it follows this 'message-encode-output' procedure repeatedly for each incoming message corresponding to each individual incoming call request. In the contrast, we handle H.245 message processing through our lookup-table based method which can save the call setup and continuation time because of shortened message processing time and thus is claimed the efficiency of our procedure.

In our experiment, for the same set of messages, we calculated the time taken for message encoding in the traditional approach (t_1) and that in our proposed approach (t_2) respectively. The *performance enhancement* (in terms of reducing call setup time) can be calculated by the following formula:

$$\frac{t_1 - t_2}{t_1} \times 100\%$$

If the result is positive, then it is obvious that we have achieved the enhancement of the performance.

The experimental result of performance enhancement in our *lookup-table* method of H.245 message processing is illustrated in Figure 6. From this graph, two important observations can be noted. Firstly, the performance remains almost unchanged when number of calls is increasing. Secondly, the performance is linearly and proportionately improving with the fraction of messages found in the lookup table. That means the performance of the gateway increases when more number of calls encountered through it.

The first observation just proves the very purpose of the currently introduced message processing approach, i.e., in this approach, the call handling performance of the 3G gateway remains invariant in spite of the number of concurrent calls increase consistently to large amount. So our procedure enables the gateway to successfully face the challenge of facing any amount of traffic load without degrading its performance.

Let us consider the second observation in the light of the discussion in Section 3. There are twelve common messages for a normal 3G video call where ten out of them are responsible for call setup and are almost same in case of all calls. So, these messages can be pre-compiled and can be stored in the lookup-table and then those can be directly retrievable (or reusable) for next incoming calls. Thus while processing the messages, the percentage of retrievable precompiled messages is $10/12 \approx 83.3\%$. So, the *performance improvement* should also be predicted to be improved up to 83.3%. Our experimental result in proves this prediction and it is visually obvious in the Figure 6.

Thus, if we consider a physical interpretation, the second observation implies that the efficiency of call setup is improved appreciably. In other words, the call setup time is reduced greatly, when the gateway handles large number of calls.

Here another important advantage of our procedure may be predicted. An appreciable amount of improvement in the conversation quality (for the ongoing calls) can be achieved in our procedure. This is because our procedure also uses the same *lookup-table* approach for handling the post connection messages too. So the system saves lots of resource and time that may eventually enable a better quality in the ongoing conversation. Thus this suggested procedure claims for better QoS. But, it is beyond the scope of our laboratory experiment to provide the data in support of this claim. However this claim can be proved when the gateway is deployed to handle high call traffic situation in real life application e.g. by a commercial 3G service provider.

6. Conclusion

On the perspective of wireless telecommunication, we aim at developing a system which can provide seamless multimedia and data communication provision among all types of IP and PSTN applications. For this communication environment, a robust and efficient multimedia gateway is also required for universal compatibility and transcoding.

In this paper, we have presented our success of attaining appreciable improvement in robustness and efficiency of the multimedia gateway which is suitable for handling high call traffic in telecommunication scenario. We have adopted an efficient 'compile-store-reuse' model for speeding up the process of encoding the ITU-T H.245

ASN.1 formatted messages in to binary stream. In this method, H.245 sets of messages are compiled, used and stored in a table. For encoding further incoming messages, the table is queried and the matching result is used instead of recompiling the messages again and again. Experimental results show that our algorithm greatly improves call handling efficiency of a 3G gateway. Thus, we have proved that our lookup-table based message encoding algorithm can provide stable performance and a satisfactory QoS assurance even under high traffic conditions. Also our method is tested to be fully compatible with existing 3G protocols.

7. Acknowledgements

The work described in this paper was fully supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 113906) and from CityU Strategic Research Grant (SRG) (Project Nos. 7002217 and 7002102).

This is also to acknowledge Dr Pranab Sabitru Naik for his continuous advice throughout this research.

8. References

[1]http://www.radvision.com/Products/3GProductsApplications/

[2] http://www.tandberg.com/products/tandberg_gateways.jsp

[3] http://www.dilithiumnetworks.com/products/DTG3000.htm

[4] ITU-T Rec. H.245, Control protocol for multimedia communication, July 2003.

[5] ITU-T Rec. H.223, Multiplexing protocol for low bit rate mobile multimedia communication, July 2001.

[6] ITU-T Rec. H.324, Terminal for low bit rate multimedia communication, March 2002.

[7] ITU-R Rec. PDNR WP8F, Vision, Framework and Overall Objectives of the Future Development of IMT-2000 and Systems beyond IMT-2000, 2002.

[8] Sanghyun Park; Jongsung Yoon; Soon-Key Jung; Sung-Jea Ko, Implementation and performance analysis of mobile multimedia terminal (H.324M), Proc. of the IEEE Region 10 Conference TENCON 99, vol: 2, pp978-981, 1999.

[9] R2. Basso A.; Kalva H., Beyond 3G video mobile conversational services: an overview of 3G-324M based messaging and streaming, Multimedia Software Engineering, Proceedings of the IEEE Sixth International Symposium on Multimedia Software Engineering (ISMSE'04), pp474-481, 2004.

[10] Recursive Design Technique for ASN.1 Decoding and Encoding, IEEE Region 10 Conference Tencon 1992, pp327-331, 1992.

[11] Qian Lv Benxiong Huang, Furong Wang, The Mechanism of ASN.1 Encoding & Decoding Implementation in Network Protocols(ITCC 03), pp622-626, 2003.

[12] M.-C. Yuen, J. Shen, Weijia Jia and B Han, Simplified Message Transformation for Optimization of Message Processing in 3G-324M Control Protocol, Proc. of International Conference on Computer Networks and Mobile Computing (ICCNMC 2005), pp64-73, 2005

[13] Yen-Chi Lee; Ming-Chang Tsai; El-Maleh, K., Efficient video data recovery for 3G-324M telephony over WCDMA networks, Proceeding of the IEEE GLOBECOM 2005, vol.1, pp47-51, 2005.

[14] R4. Miki T., Hotani S., Kawahara T., Error resilience features of MPEG-4 audio-visual coding and their application to 3G multimedia terminals, Proceedings of IEEE ICSP 2000, pp40-43, 2000.

[15] ITU-T Rec. X.691, Information technology - ASN.1 encoding rules: Specification of Packed Encoding Rules (PER), July 2002.

[16] Gerald W. N. and Yueli Y., The Design and Implementation of an ASN. 1-C Compiler, IEEE Transactions on Software Engineering. Vol. 16, No. 10, 1990.