실시간 선불 서비스를 위한 모바일 IPv6 권한검증 구현☆

Implementation of Mobile IPv6 Fast Authorization for Real-time Prepaid Service

김 현 곤
HyunGon Kim

요 약

차세대 무선 네트워크에서는 응용들이 실시간 선불 서비스를 제공해야 하며, 최종 사용자에게 서비스를 제공하기 이전에 요청된 서비스에 대해 신용을 사전 체크하여야 한다. 또한, 선불서비스를 효과적으로 제공하기 위해서는 신용제어 기능이 최소한의 지연만을 가져야 한다. 본 논문에서는 모바일 IPv6 환경에서 실시간 신용제어가 가능한 권한검증 구현 모델을 제안하였다. 제안한 모델은 일반적인 신용제어 권한검증 절차와 모바일 IPv6 인증 절차를 통합한 구조를 갖는다. 지연을 최소화하기 위해 제안한 모델을 싱글 서버 내에 구현하였으며, 이 시스템은 권한검증과 인증을 동시에 수행한다. 구현한 시스템의 구현구조를 소프트웨어 기능 블록과 유니트 형태로 제시하였다. 구현한 모델의 feasibility를 검증하기 위해서 구현한 시스템의 지연을 측정하였으며 측정 시, 몇 가지 인증 확장 프로토콜 (EAP)을 적용하였다. 측정된 결과에 따르면 신용제어 권한검증과 인증이 분리된 기존 모델과 비해서 제안한 통합 모델이 상대적으로 지연시간이 적었다.

Abstract

In next generation wireless networks, an application must be capable of rating service information in real-time and prior to initiation of the service it is necessary to check whether the end user's account provides coverage for the requested service. However, to provide prepaid services effectively, credit-control should have minimal latency. In an endeavor to support real-time credit-control for Mobile IPv6 (MIPv6), we design an implementation architecture model of credit-control authorization. The proposed integrated model combines a typical credit-control authorization procedure into the MIPv6 authentication procedure. We implement it on a single server for minimal latency. Thus, the server can perform credit-control authorization and MIPv6 authentication simultaneously. Implementation details are described as software blocks and units. In order to verify the feasibility of the proposed model, latency of credit-control authorization is measured according to various Extensible Authentication Protocol (EAP) authentication mechanisms. The performance results indicate that the proposed approach has considerably low latency compared with the existing separated models, in which credit-control authorization is separated from the MIPv6 authentication.

☞ Keyword : prepaid service, credit-control, Mobile IPv6, authentication, AAA

1. 서 론

The prepaid model has proved to be very successful in applications such as GSM networks, where network operators offering prepaid services have experienced substantial growth of their customer base and revenues. Prepaid services are now cropping up in many other wireless and wire line based networks as well. In next generation wireless networks, additional functionality is required beyond that specified in the Diameter base protocol [1]. For example, the 3GPP Charging and Billing requirements state that an application must be able to rate service information in real-time [2]. In addition, it is necessary to check whether the end user's account provides coverage for the requested service, prior to initiation of that service. When an account is ex-

* 정 회 원 : 목포대학교 정보통신공학부 교수
hyungon@mokpo.ac.kr(제1저자)
[2005/06/11 투고 - 2005/06/28 1차 심사 - 2005/10/11
2차 - 2005/11/09 심사완료]
☆ 본 논문은 2005학년도 목포대학교 학술연구비지원에 의하여 연구되었음.

hausted or expired, the user must be denied the capacity to compile additional chargeable services. A mechanism that informs the user of the charges to be levied for a requested service is also needed. In addition, there are services such as gaming and advertising that may credit as well as debit from a user account.

For general purposes, the Diameter Credit-Control Application [3] was proposed to support prepaid services. It can be used to implement real-time credit-control for a variety of end user services such as network access, Session Initiation Protocol (SIP) services, messaging services, download services, etc. However, in the event that long latency is induced by authentication in the home network e.g., MIPv6 authentication [4], a practical approach should be considered to ensure real-time processing. Also on the other hand, relative research on credit-control authorization is mainly being devoted by IETF standardization and its implementation issues are out of scope in the standardization. Considering these, we design and implement an architecture model that is capable of MIPv6 service specific fast authorization for prepaid services. The remainder of the paper is organized as follows. Chapter 2 briefly presents an architecture model of credit-control authorization on a MIPv6 infrastructure. Chapter 3 describes the proposed credit-control authorization procedure and chapter 4 presents the implementation architecture in detail. Chapter 5 describes performance results and conclusions are presented in chapter 6.

2. Architecture of Credit-Control Authorization with Embedded MIPv6 Authentication

Fig. 1 illustrates the architecture of a credit-control authorization model with embedded MIPv6 authentication. It consists of a service element with embedded Diameter credit-control client, a Diameter credit-control server, a business support system, and MIPv6 authentication servers. i.e., a foreign authentication, authorization, and accounting server (AAAF) in the Mobile Node (MN)'s foreign network and a home AAA server (AAAH) in the MN's home network [5].

The Diameter Credit-Control Application [3] defines the framework for credit control; it can provide generic credit-control authorization. In order to support real-time credit-control with an embedded MIPv6 authentication mechanism [6], a new type of server is needed, i.e., Diameter credit-control server. This server is the entity responsible for credit authorization for prepaid subscribers. It also acts as a prepaid server, performing real-time rating and credit control. It is located in the home domain and is accessed by service elements or Diameter AAA servers in real-time for the purpose of price determination and credit-control before a service event is delivered to the end user. It may also interact with business support systems. The service element can be an Access Router (AR), which is defined in MIPv6 basic service [4] or an AAA server in the foreign domain.

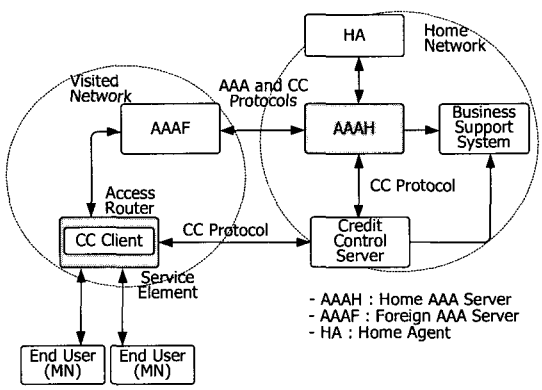

Fig. 1 Architecture of credit-control authorization with embedded MIPv6 authentication

A Diameter credit-control client is an entity that interacts with a credit-control server. It monitors the usage of the granted quota according to instructions returned by the credit-control server. A business support system is usually deployed, and it includes at least the billing functionality. The credit-control protocol is the Diameter base protocol with the Diameter Credit-Control application. The AAA protocol with embedded credit-control protocols is used between the credit-control client and AAA server. However, the credit-control protocol is only used between the credit-control server and the credit-control client and between the credit-control server and AAA server. In this paper, to launch the credit control authorization model on a MIPv6 authentication infrastructure, it is assumed that credit-control client functionality is performed by AR in the MN's visited domain. It is also assumed that the AAAH server performs both basic AAAH server functionality and credit-control server functionality. The latter means that two server functionalities are implemented on a single host even if it is explicitly defined as an external interface.

3. Embedded Credit-Control Authorization Procedure

3.1 Diameter Messages for Credit-Control Authorization

The Diameter Credit-Control Application [3] defines two Diameter messages, Credit-Control-Request (CCR) and Credit-Control-Answer (CCA). The CCR message requests credit-control authorization for a given service. When an end user requests a service, the request is issued by the credit-control client and is forwarded to the credit-control server. The CCA message acknowledges the CCR message. The CCR and CCA have four types of interrogation, initial, intermediate, final, and one-time events. First-interrogation (CCR-Initial) is used to first interrogate a requested prepaid service. The credit-control server will check whether the end user is a prepaid user and will rate the service event in real-time. It also makes a credit-reservation from the end user's account that covers the cost of the service event.

Intermediate-interrogation (CCR-Update) is used to make a new credit reservation while the service is ongoing. When all of the granted service units are spent by the end user or the validity time is expired, the credit-control client sends a new CCR to the credit-control server. The server deducts the used amount from the end user's account. The CCR-Update may rate the new request and make a new credit-reservation from the end user's account that covers the cost of the new requested service event. Final-interrogation (CCR-Termination) is used to close credit-control authorization. When the end user terminates the service session, the credit-control client sends a final CCR message to the credit-control server. After the final interrogation, the server refunds the unused reserved credit amount to the end user's account and deducts the used monetary amount from the end user's account. One-time event (CCR-Event) is used when there is no need to maintain any state in the credit-control server, for example, requiring the price of the service. The use of a one-time event implies that the user has been authenticated and authorized beforehand.

3.2 Initial Credit-Control Authorization Procedure

Fig. 2 illustrates the initial credit-control au-

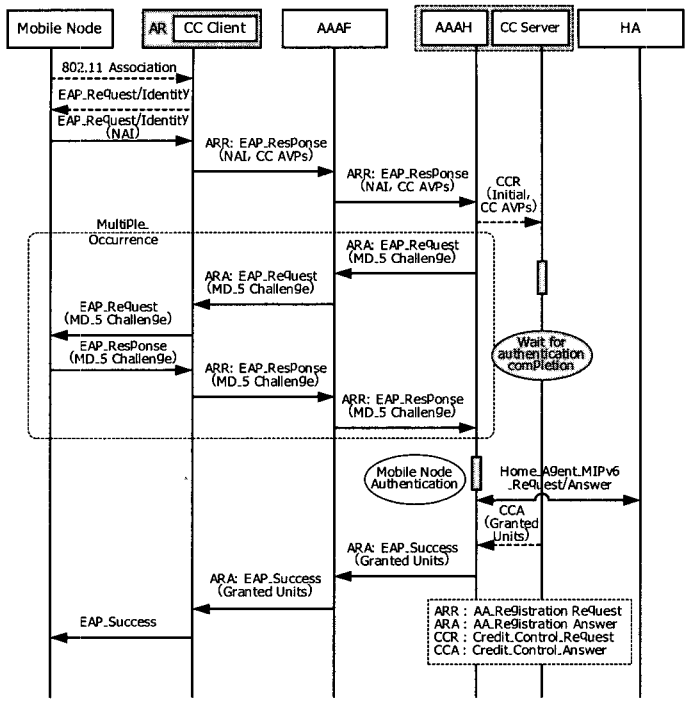

Fig. 2 Initial Credit-Control authorization with embedded MIPv6 authentication

thorization procedure with an embedded MIPv6 authentication procedure for Initial-interrogation (CCR-Init). It proceeds as follows:

(1) A user logs onto the network. An MN may make a data link connection using a data link protocol such as IEEE 802.11.

(2) Upon receipt of a Network Access Identifier (NAI) from the network, the AR with a credit-control client populates the Diameter ARR (Authorization-Authentication Registration Request) message with the Credit-Control AVPs (Attribute-Value Pair) set to CREDIT-AUTHORIZATION. MIPv6 specific AVPs are included. The ARR message requests MIPv6 authentication and credit-control authorization from the MIPv6 authentication server and credit-control server, respectively, in the home network. Then, the AAAF forwards the request to the AAAH.

(3) The AAAH may perform MIPv6 specific authentication and authorization as usual. It determines whether the user is a prepaid user and identifies from the Credit-Control AVPs. It then sends a Diameter CCR with CC-Request-Type set to INITIAL-REQUEST to the internal credit-control server to perform credit authorization and to establish a credit-control session (the AAAH may forward MIPv6 specific AVPs as received from the AR as input for the rating process).

(4) The credit-control server waits for authentication completion of the MN by the AAAH.

(5) After challenges and responses are processed between the MN and AAAH, the AAAH performs authentication of the

MN. If it is successful, the AAAH sends a HOR (Home-Agent-MIPv6-Request) message to the Home Agent (HA) to perform binding update of the MN. It may receive a HOA (Home-Agent-MIPv6-Answer) from the HA.

(6) The reserved quota thus, Granted Units may be sent to the AAAH by the credit-control server. It may be included in the Diameter ARA (Authorization-Authentication Registration Answer) message. The AAAH sends it to the credit-control client through the AAAF.

(7) Upon receipt of a successful ARA, the AR starts the credit-control service and starts monitoring the granted units. The AR grants access to the end user.

4. Implementation of Credit-Control Server

4.1 System Configuration and Parameters

This chapter presents implementation of the Credit-Control (CC) server in detail. Fig. 3 presents the system configuration for implementation of the CC server and Table 1 describes the system parameters used in this paper. Certificate parameters are used in the Transport Layer Security (TLS). Each entity has a public IPv6 address and IP signaling and traffic are routed by a standard IPv6 routing scheme. Regarding real deployment for the MIPv6 service, the network is segmented by two subnets, the MN's visited network and the MN's home network. In order to launch credit-control authorization, it is assumed that CC client functionality is performed by the AR in the MN's visited networks and the CC server performs both basic CC server functionality and AAAH functionality.

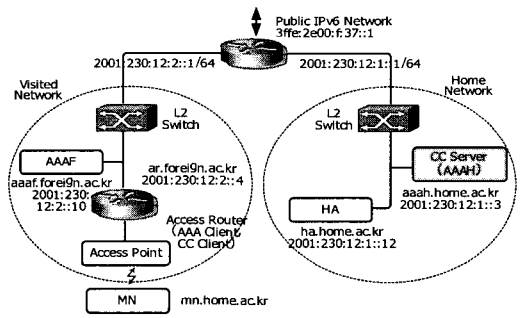

Fig. 3 System configuration for implementation

Table. 1 System parameters

	System parameters
Platform	CC server; SUN V880
	MN/AR/HA/AAAF; ZION Linux P-III
OS	Solaris 8/Red Hat 9(Kernel v2.4.20)
Link Capacity	100Mbps(100 Base-T)
Routing	IPv6 and Diameter message routing
Certificates Parameters	Certificate size; 493 bytes
	ClientHello/ServerHello size; 60/66 bytes
	ClientKeyExchange message size; 64 bytes
	Finished message size; 12 bytes
	CertificateVefiry message size; 64 bytes

4.2 Software Architecture and Functional Blocks

This section describes the implemented software architecture, blocks, and units in the CC server. Fig. 4 illustrates the implemented software architecture of the CC server. There are five software blocks, Low-layer Transport Block (LTB), Diameter Base Engine Block (DBEB), MIPv6 Security Application Block (MSAB), Credit-Control Block (CCB), EAP Block (EAPB), and Operation and Management Block (OMB). Each block acts as a process, and the UNIX System V IPC queue is utilized to communicate between blocks, i.e., processes.

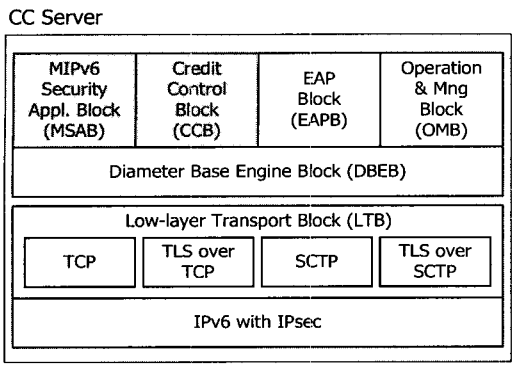

Fig. 4 Software architecture of CC server

The DBEB performs the Diameter Base Protocol followed by [1], the MSAB performs the Diameter Mobile IPv6 Security Application followed by [6], the CCB performs the Diameter Credit-Control Application followed by [3], and the EAPB performs EAP Authentication Protocol followed by [7]. The OMB performs CC server operation and management functionality.

LTB (Lower-layer Transport Block) : Based on the IPv6 routing scheme, the LTB performs low-layer transport functionality followed by [8]. It has capable of processing multiple transport layer protocols such as TLS, Stream Control Transmission Protocol (SCTP) [9], TLS over SCTP, and TCP. Thus, the DBEB in the upper layer may choose a proper transport protocol corresponding to the peer's transport capability.

DBEB (Diameter Base Engine Block) : The DBEB provides the following facilities: Delivery of AVPs, capabilities negotiation, error notification, extensibility through addition of new commands and AVPs, and basic services necessary for applications such as handling of user sessions or accounting. It consists of eight software units and three libraries, which provide the following functionalities respectively:

- Main Controller performs initialization for power-up/reset and sending/receiving primitives from/to internal units.
- PCMU (Peer Connection Managing Unit): performs low-layer peer connection management based on LTB functionality, such as connection establishment, monitoring, and release.
- RSU (Routing Service Unit): performs Diameter message routing service. A request is sent toward its final destination using a combination of Destination-Realm and Destination-Host AVPs. If the request cannot be processed locally, it is forwarded to other Diameter agents such as proxies, redirects, or relays.
- OMB_MU (OMB_Managing Unit): performs local operation and management functionality and cooperates with external OMB.
- AMU (Accounting Managing Unit): performs real-time accounting management by handling accounting messages. It collects accounting records for roaming subscribers according to an accounting state machine. Accounting records include session time, input octets, output octets, input packets, output packets, etc.
- PAMU (Path Authorization Managing Unit): checks that its peers are authorized to act in their roles, before initiating a connection.
- CEMU (Capabilities Exchange Managing Unit): performs capabilities negotiation in

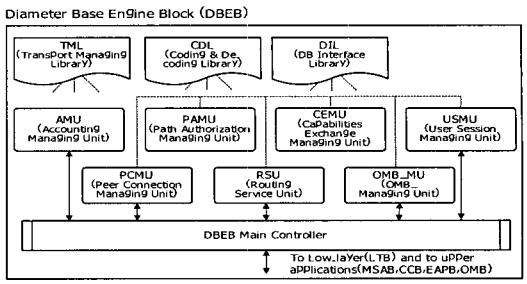

Fig. 5 DBEB software block in CC server

order to determine what Diameter applications are supported by each peer.

- USMU (User Session Managing Unit): manages user sessions according to a user session state machine.
- Three common libraries: provide transport layer management, CODEC, and DB interface by using a TML (Transport Managing Library), CDL (Coding & Decoding Library), and DIL (DB Interface Library).

CCB (Credit Control Block) : As described above, the CC server provides credit-control authorization with embedded MIPv6 authentication. The MSAB performs MIPv6 authentication and, afterwards, the CCB performs authorization for a MN. To do this, MSAB and CCB are tightly coupled to exchange authentication results and credit-control authorization results. After successful MIPv6 authentication of the MN, the result is informed to the CCB to authorize the MN. The CCB consists of five software units, which provide the following functionalities respectively:

- Main Controller/Interface Handler: performs initialization for power up/reset and sending/receiving primitives from/to internal units.
- Credit Handler: performs credit authorization and debits end user credit or refunds credit.
- CC Session Handler: manages credit-control sessions according to a Diameter credit-control session state machine.
- Diameter Message Handler: decodes messages such as CCR and CCA according to Diameter message format, and also encodes messages to be used by credit-control peer.
- Error Handler: handles errors and exceptions.

MSAB (MIPv6 Security Application Block) : The MSAB consists of six software units and

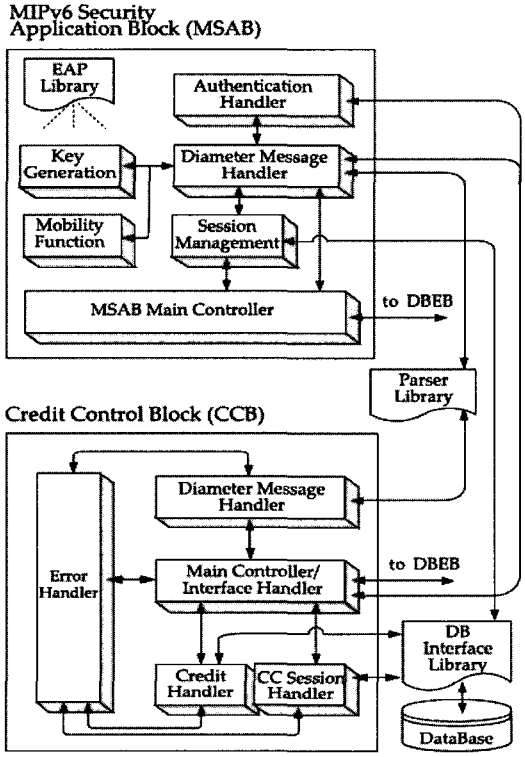

Fig. 6 CCB and MSAB blocks in CC server

one library, which provide the following functionalities respectively:

- Main Controller: performs initialization and sending/receiving primitives from/to internal units.
- Mobility Function: performs HA assignment and home address assignment of MN.
- Key Generation: performs key generation to be used for IPSec security association.
- Authentication Handler: performs authentication of MN and interacts with CCB for credit-control authorization.
- Diameter Message Handler: handles Diameter messages such as ARR, ARA, HOR, and HOA and message encoding and decoding.
- Session Management: manages MIPv6 authentication sessions according to a Diameter MIPv6 user session state machine.

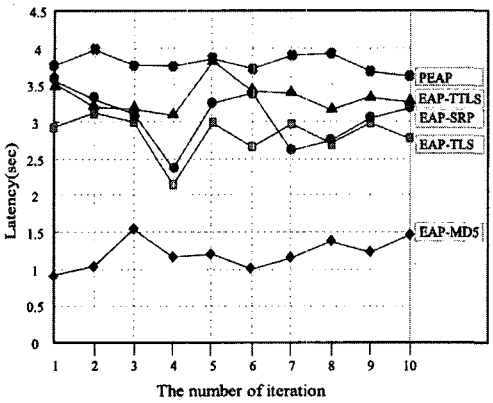

Fig. 7 Latency of CC authorization with MIPv6 authentication

· EAP Library: provides five EAP authentication mechanisms, EAP-MD5, EAP-TLS, EAP-TTLS, EAP-SRP, and PEAP [10].

EAPB (EAP Block) and **OMB** (Operation and Management Block) : The EAPB provides a standard mechanism for support of various EAP authentication mechanisms. It carries EAP packets between the Network Access Server (NAS), working as an EAP Authenticator, and a back-end authentication server, i.e., a CC server with AAAH. The OMB performs operation and management of the CC server itself. It interacts with other software blocks in the CC server.

Table. 2 Average latency of CC authorization and CC server's processing time

EAP authentication	Average latency and processing time
EAP-MD5	Average latency : 1.2 sec
	CC server's processing time: 399msec
EAP-TLS	Average latency : 2.82 sec
	CC server's processing time: 839msec
EAP-TTLS	Average latency : 3.3 sec
	CC server's processing time: 1,105msec
EAP-SRP	Average latency : 3.06 sec
	CC server's processing time: 1,018msec
PEAP	Average latency : 3.8 sec
	CC server's processing time: 1,128msec

5. Performance Results

Latency measurement is performed in order to evaluate the proposed credit-control authorization approach with embedded MIPv6 authentication according to several EAP authentication mechanisms. The latency is the amount of time between the start of an MN's MIPv6 service request with authentication and credit-control authorization, and the end of the service. To measure the latency, MN generates a few hundred MIPv6 authentication requests and inserts into the implemented CC server simultaneously. The latency is measured at the MN. Fig. 7 shows the measured performance results.

Table 2 shows the average latency and CC server' processing time. It should be noted that the measured server's processing time depends on system processing capability such as CPU processing time, I/O operation capability, memory resources, and so on. In the case of EAP-MD5, the average latency is about 1.2 sec for one credit-control authorization with one MIPv6 authentication. The results indicate that EAP-TLS, EAP-TTLS, EAP-SRP, and PEAP require about 2.82 sec, 3.3 sec, 3.06 sec, and 3.8 sec latency, respectively.

Under the same conditions, we measure the latency of MIPv6 authentication alone for comparison with the latency of the proposed credit-control authorization scheme with MIPv6 authentication. The latter requires only 2% additional latency relative to the former. In the case of EAP-MD5, the latter requires 1.224 sec additional latency. Thus, considerably lower latency is realized relative to that of the existing, separated models, wherein the credit-control authorization server is separated from the MIPv6 authentication server. In addition, the separated model may have additional latency such as the

server's processing time, transport layer latency, data link latency, physical latency, and so on. Since a CC server must have a Diameter agent's transport capability [3,8] such as TCP, TLS, and SCTP, the transport layer may also require a hundred msec order delay.

6. Conclusions

This paper has presented an architecture model of credit-control authorization for MIPv6 services, which have recently drawn remarkable attention in IETF. From a real deployment point of view, we have attempted to realize a practical approach to credit-control authorization. The proposed integrated model combines a credit-control authorization procedure into the MIPv6 authentication procedure. In order to verify the feasibility of the proposed approach, we implemented it and measured the latency of credit-control authorization with MIPv6 authentication and compared the results with those yielded by existing schemes. From the performance results, we found that the proposed integrated model has considerably low credit-control authorization latency compared with the separated models, which separate credit-control authorization from MIPv6 authentication. Thus, the proposed integrated model is more effective than the separated models in terms of real-time processing. As a further work, we will implement a full kernel space implementation to get high performance.

References

[1] Pat R. Calhoun, J. Loughney, E. Guttman, G. Zorn, J. Arkko, "Diameter Base Protocol", IETF RFC 3588, 2003.

[2] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects, Service aspects; Charging and Billing(release 5)", 3GPP TS22.115 (v.5.2.1), 2002.

[3] Harri Hakala, Leena Mattila, Juha-Pekka Koskinen, Macro Stura, John Loughney, "Diameter Credit-Control Application", IETF draft, draft-ietf-aaa-diameter-cc-06.txt, 2004.

[4] D. Johnson, C. Perkins, J. Arkko, "Mobility Support in IPv6", IETF RFC 3775, 2004.

[5] Stefano M. Faccin, Frank Le, et al., "Mobile IPv6 Authentication, Authorization, and Accounting Requirements", IETF draft, draft-le-aaa-diameter-mipv6-requirements-03.txt, 2004.

[6] Stefano M. Faccin, Frank Le, Basavaraj Patil, Charles E. Perkins, "Diameter Mobile IPv6 Application", IETF draft, draft-le-aaa-diameter- mobileipv6-03.txt, 2004.

[7] P. Eronen, Ed., T. Hiller, G. Zorn, "Diameter Extensible Authentication Protocol (EAP) Application", IETF draft, dreaft-ietf-aaa-eap-10.txt, 2004.

[8] PB. Aborba, J. Wood, "Authentication, Authorization and Accounting (AAA) Transport Profile", IETF RFC 3539, 2003.

[9] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, et al., "Stream Control Transmission Protocol", IETF RFC 2960, 2000.

[10] B. Aborba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz, Ed, "Extensible Authentication Protocol (EAP), "IETF RFC 3748, 2004.

◐ 저 자 소 개 ◑

김 현 곤 (HyunGon Kim)
1992년 금오공과대학교 전자공학과 졸업(학사)
1994년 금오공과대학교 대학원 전자공학과 졸업(석사)
2003년 충남대학교 대학원 전자공학과 졸업(박사)
1994년~2004년 한국전자통신연구원 정보보호연구단 팀장
2005년~현재 목포대학교 정보공학부 교수
관심분야 : RFID/USN 네트워크, 무선통신 보안, RFID/USN 보안 등
E-mail : hyungon@mokpo.ac.kr