제 18 회 한국정보처리학회 추계학술발표대회 논문집 제 9 권 제 2 호 (2002.11)

무선 인터넷 접속 장치에서 실시간 인증을 위한 AAA 구현 방법

이훈기*, 김순철, 류원
한국전자통신연구원, 유무선인터넷정합팀
e-mail : lhk@etri.re.kr

A Implementation Method of AAA for Real-Time Authentication on MiDAS

Hoon-Ki Lee, Soon-Chul Kim, Won Ryu
Wire and Wireless Internet Interworking Team
Electronics and Telecommunications Research Institute

요 약

본 논문은 무선망과 유선 인터넷 망을 연결하여 실시간 인터넷 접속 서비스를 받을 수 있는 IS-95C 패킷 데이터 서비스 노드(PDSN: Packet Data Service Node) 기능을 수행하는 개방형 무선 인터넷 접속장치(MiDAS: Mobile Interface Data Access System)에서 사용자 인증 및 권한 검증을 수행하는 AAA 프레임웍의 설계 방법에 관한 것이다. 실시간으로 접속되는 무선 인터넷 접속 사용자의 수가 증가할수록 PDSN 에서의 사용자 관리가 중요한 문제로 대두되고 이러한 문제점을 효율적으로 처리할수 있는 방법을 제시한다. MiDAS 에서의 AAA 프레임웍 구현을 위해 RADIUS 프로토콜을 이용하였으며 실시간으로 요구하는 무선인터넷 사용자의 인증을 수행하기 위해 AAA 클라이언트에서 사용자 관리를 위한 구현방법, 통신 구현방법, 그리고 타이머를 통하여 인증 처리에서 서버 응답에 대한 임계시간을 두어 구현하는 방법에 관한 것이다.

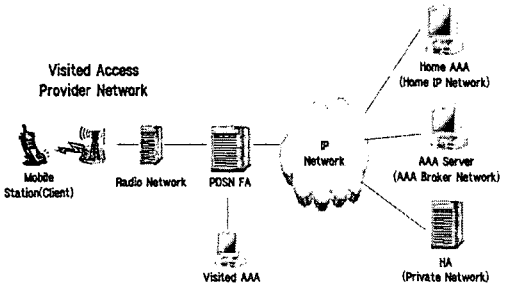
1. 서론

정보 통신과 무선 단말기의 발달에 따라 장소와 시간에 관계없이 사용자는 무선 데이터 서비스를 요구하고 이동 통신 사업자는 이러한 사용자의 요구를 수용하는 방향으로 나아가고 있다. 또한 네트워크의 유무선 인프라 통합에서부터 서비스 연역에 이르기까지 다방면의 유선과 무선 통신망 자원을 통합하려는 활발히 진행되고 있다[1]. 이러한 환경하에서 무선 인터넷 서비스 사용자가 증가함에 따라 안전한 서비스를 위하여 사용자 인증에 대한 중요성이 증대되고 인증 과정에서의 대기 시간 지연으로 발생되는 문제점이 대두될 수 있다. 본 논문에서는 무선망과 유선 인터넷 망을 연결하여 실시간 인터넷 접속 서비스를 받을 수 있는 IS-95C 패킷 데이터 서비스 노드(PDSN: Packet Data Service Node) 기능을 수행하는 개방형 무선 인터넷 접속장치(MiDAS: Mobile Interface Data Access System)에서 사용자 인증 및 권한 검증을 수행하는 AAA 프레임웍 구현 방법에 관한 것으로 MiDAS 에서의 AAA 프레임웍 구현을 위해 RADIUS 프로토콜[2]을 이용하였으며 실시간으로 요구하는 무선인터넷 사용자의 인증을 수행하기 위해 AAA 클라이언트에서 사용자 관리를 위한 구현방법, 통신 구현방법, 그리고 타이머를 통하여 인증 처리에서 서버 응답에 대한 무한 대기를 억제할 수 있는 방법에 관한 것이다.

본 논문의 구성은 2 장에서는 MiDAS 시스템이 운용될 망구조와 시스템의 전반적인 소프트웨어 블록들 간의 구성에 대하여 알아보고 3 장에서는 효율적인 인증 과정을 수행하기 위해 각각의 AAA 구성 프로세서들의 구현 방법에 대하여 알아본다. 4 장에서는 3 장에서 언급한 구현 방법을 토대로 실제 MiDAS 시스템에서 적용된 방식에 대하여 알아보고 마지막으로 결론 및 향후 과제에 대하여 논한다.

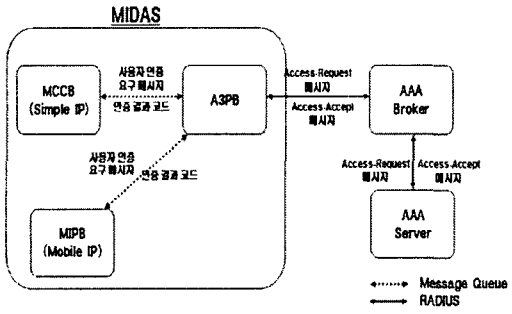
2. MiDAS 시스템에서의 AAA Framework

MiDAS 가 운용될 망 구조도는 <그림 1>에서 보는 것과 같이 일반적인 무선 망과 유선 망 사이의 패킷 데이터 서비스를 위해 사용되는 PDSN 장치다[2][3]. <그림 1>에 나타낸 AAA 연동 구조는 RADIUS 프로토콜을 이용하고 각각의 AAA 를 위하여 서버, 클라이얼드 그리고 서버간의 중계역할을 하는 브로커로 구성된다. AAA 클라이언트는 MiDAS 시스템의 하나의 단위 블록으로 존재한다.



<그림 1> 무선 인터넷 망 구성도

<그림 2>은 MiDAS 시스템을 구성하는 단위 블록들 간의 인터페이스와 메시지 흐름도이다. MiDAS 시스템이 지원하는 무선인터넷 서비스는 Simple IP 와 Mobile IP 서비스를 지원한다. 인증처리를 위하여 인증 및 권한 검증의 AAA 클라이언트를 담당하는 A3PB(AAA Processing Block)은 Simple IP 서비스의 세부 역할을 담당하는 MCCB(Main Call Control Block) 블록과 Mobile IP 서비스를 담당하는 MIPB(Mobile IP Processing Block)블록과 인터페이스르 가진다[3].



<그림 2> 인증 메시지 흐름도

A3PB 블록은 내부적으로 3 개의 프로세서인 구성되는 데 메인 프로세서, 타이머 프로세서, 통신 프로세서로 구성되며 프로세서를 세부적으로 나눈것도 실시간으로 사용자의 인증을 처리하기 위해서이다. AAA 메인 프로세서는 무선 단말기로 사용자가 인증을 요구하면 서비스 종류(Simple IP, Mobile IP)에 따라 RADIUS 인증 요구 메시지를 생성 및 폐기, 그리고 분석단계를 수행하고 타이머 프로세서는 메인 프로세서에서 서버로 인증 요구 메시지를 전송한 시점부터 일정 임계 시간동안 처리 결과 메시지가 수신되지 않을 경우 AAA 클라이언트로 하여금 처리 결과 메시지를 생성하여 사용자 단말로 결과를 전송하는 기능을 담당한다. 통신 프로세서는 실시간으로 서버로의 처리 결과 메시지를 수신하여 메인 프로세서로 메시지를 전달하는 기능을 수행한다. 또한 다중 서버로 구성된 경우 각 사용자가 속한 realm 범위에 따라 선택된 서버로 인증 요구 메시지를 전달하는 기능을 담당한다[4].

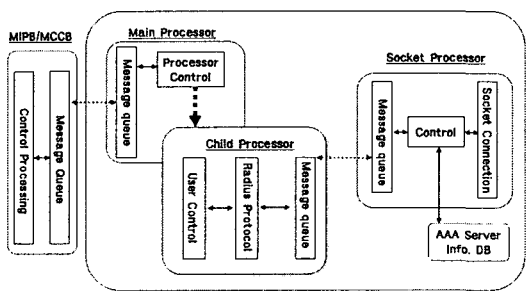
3. MiDAS 의 프로세서 구현 방법

3.1 메인 컨트롤 프로세서 구현

인터넷 사용자가 급증할수록 실시간으로 인증을 처리하기 위해서 인증처리를 위한 대기 시간이 증가 할 수가 있다. 이런 단점을 줄이기 위해 AAA 를 구현하기 위한 두가지 방법을 제시한다. 여기서 논의된 구현방법에서 서버와 클라이언트가 직접적으로 수행해야할 RADIUS 프로토콜 처리 부분과 서버의 사용자 인증 처리 방법에 대해서는 논의하지 않는다. 서버와 클라이언트 구별없이 하나의 인증 메시지를 처리하기 위한 프로세서의 효율적 구현 방법이다.

- 프로세서(Child Processor) 생성 방식

AAA 프레임웍 구현을 위한 방식으로 사용자가 인증을 요구한 메시지가 MiDAS 에 수신되면 AAA 메인 프로세서에서 Pork 를 통하여 자식 프로세서를 생성하여 인증 처리가 완료될 동안 유지하도록 한다. 이때 AAA 서버와의 통신 방법에 대해서도 두가지 방법이 존재하는데 각 프로세서에서 AAA Server 와의 통신을 담당하는 방법과 별도의 통신 프로세서와의 인터페이스를 통하여 하나의 프로세서가 서버와의 통신을 담당하는 방법이 있다. 전자는 별로의 통신 프로세서와의 인터페이스를 해야한다는 단점이 있으나 고정 포트를 사용하여 서버와의 통신을 전담할 수 있다. 후자는 자식 프로세서에서 통신 부분을 담당하기 때문에 인터페이스로 인한 지연시간은 줄일수 있는 반면 각 스레드에게 통신 포트를 할당해야 하므로 AAA 메인 프로세서가 포트 자원을 관리해야하는 단점이 있다.



<그림 3> 프로세서 생성 방식 블록도

또한 수십만 가입자의 인증 요구 메시지가 수신될 경우 프로세서 생성의 한계로 인하여 인증 시간이 지연될 수도 있다. <그림 3> 메인 프로세서에서 자식 프로세서 생성 방식을 통하여 사용자 인증 처리를 수행하고 별도의 통신프로세서를 두어 전담하는 방식의 블록 구성도이다.

메인 프로세서는 자식 프로세서를 생성시에 RADIUS 메시지의 패킷 포맷에서 RADIUS Id 필드 값의 중복을 피하기위해 프로세서 생성시 MCCB 와 MIPB 에서 수신된 인증 데이터와 메인 프로세서에서 할당한 RADIUS Id 를 가지고 AAA 클라이언트 처리를 수행한다.

- 리스트(List) 관리 방식

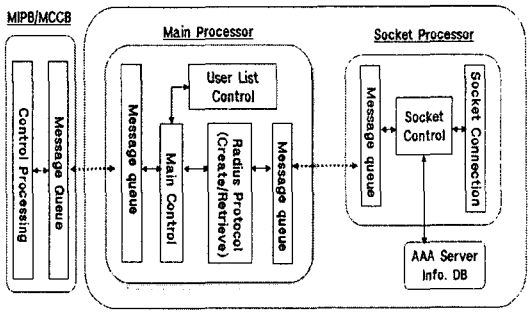
메인 프로세서에서 인증 요구 메시지가 수신되면 내부적으로 리스트로 데이터를 관리하고 처리하는 방식이다. 리스트 관리 방식은 메인 프로세서에서 모든 사용자에 대하여 인증 처리를 수행해야 하므로 리스를 관리를 위한 정책이 필요하다. 하나의 리스트로 모든 사용자를 관리하는 것보다는 몇 개의 개별 리스트를 두어 리스트 탐색 시간을 줄일수 있는 방법을 적용하여 인증 처리 시간을 단축할 수가 있다. 본 논문에서는 2 개의 리스트를 관리하도록 한다. Simple IP 서비스를 요구하는 사용자 리스트, Mobile IP 서비스를 요구하는 사용자 리스트를 따로 관리하여 처리한다. <표 1>은 서비스 사용자를 관리하기 위한 구조체이다. iCallId 는 MiDAS 내부에 동작하는 각 블록들에서 공통적으로 사용하는 ID 이고 iRadiusId 는 RADIUS 프로토콜 메시지의 구성요소에 들어였는 값으로 이 값을 통하여 서버로부터 수신된 정보가 어떤 서비스 사용자 인지를 판별하고 리스트를 검색한다. stAttrBuf 는 인증 서버의 정책에 따라 인증 실패 요인과 관련하여 재 전송이 가능하도록 초기 작성된 RADIUS 메시지중에서 Attributes 영역에 해당되는 데이터를 저장한다. 재 전송이 발생하였을 경우 메시지를 다시 만들어야하는 시간적 손실을 억제하기 위해 반복 전송이 발생하더라도 수정되지 않는 영역에 대해서는 리스트에 함께 관리한다.

```
typedef struct radUserList{
        unsigned int        iCallId;
        unsigned char       cUserName[maxSize];
        unsigned char       cRealm[maxSize];
        uint8_t             iRadiusId;
        MsgReqAvps          *stAttrBuf;
        struct radUserList *next;
}
```

<표 1> Simple IP 서비스 사용자 리스트

RADIUS Id 의 부여 방법은 Simple IP 서비스 사용자는 짝수 번호로 부여하고 Mobile IP 사용자는 홀수 번호를 부여하여 탐색에서 걸리는 시간을 줄일 수가 있다. 서버로부터 수신된 메시지 중에서 각 개별 사용자의 인증에 따른 식별값으로 사용할 수 있는 중요한 데이터이다.

<그림 4>는 메인 프로세서에서 리스트 관리를 통하여 AAA 클라이언트 처리 부분을 수행하는 블록도이다. 본 블록도는 앞서 언급한 스레드 방식에 비해서 실시간으로 사용자 요구가 있을 경우 리스트 자원의 확장 이외에는 기존 자원의 추가 확보가 없으므로 스레드 생성으로 발생하는 자원 점유를 어느정도 보상할 수 있는 장점이 있다.



<그림 4> 리스트 관리 방식 블록도

3.2 타이머 프로세서 구현

AAA 클라이언트에서 AAA 서버로 인증 메시지(Access Request)를 송신하고 처리 결과 메시지를 수신하기 위해 MiDAS 시스템에서는 타이머 프로세스를 지원하다. 타이머의 시작, 종료 제어를 통한 구현 방법이 있다.

- 타이머 시작과 종료 제어

시작과 종료를 제어하는 타이머는 AAA 프로세서에서 자식 프로세서를 생성하여 인증처리를 할 경우 이용하는 구현 방식이다. 자식 프로세서로부터 전송 받은 메시지에는 타이머 구동에 필요한 구동 시간(Sleep Time), 반환시 식별 코드(Call ID)가 포함되고 타이머 스레드를 구동하게 된다. 각각의 타이머 스레드는 Microseconds 로 구동하며 각각의 프로세서로 부터 종료 메시지가 수신되었는지 타임 간격으로 메시지 큐나 공유 메모리를 검사한다. 타이머 종료전에 메인 프로세서로부터 타이머 종료 메시지를 수신하면 타이머 스레드는 처리결과 메시지의 생성 없이 타임 스레드를 종료하게 된다. 이런 방법의 구현은 시스템 자원 낭비를 줄일 수가 있다.

- 타이머 시작 제어

시작 제어만 구현하는 경우는 타이머 프로세서가 타이머 구동을 지시하고 타이머 스레드는 종료가 될 때까지 어떠한 메시지도 송/수신하지 않는다. 다만 타이머 스레드가 종료가 되었을 경우 종료 메시지를 작성하여 메인 프로세서로 전송하게 된다. 메인 프로세서는 인증타이머 구동을 요구한 후 인증 처리결과에 관계 없이 항상 타임 종료 메시지를 받게 된다. 종료 메시지 수신 시점에서 현재의 인증 처리 과정에 따라 인증 결과 메시지를 작성하여 인증을 요청한 타 블록으로 전송하게 된다. 시스템 자원측면 보다는 필요 이

상의 메시지를 줄일 수가 있다.

3.3 통신 프로세서 구현

AAA 서버와의 통신은 RADIUS 규격에 따라 UDP 소켓 통신을 이용한다. 대부분의 AAA 프레임웍에서는 한 개 이상의 AAA 서버를 둘수 있으며 각각의 서버들은 사용자의 NAI 정보를 통하여 구별하도록 한다. 사용자 영역에서 정의된 서버와의 통신 실패로 인하여 타 서버와의 통신을 재게하여야 할 경우 AAA 클라이언트는 다른 포트를 통하여 새로운 통신 선로를 연결하거나 다중 서버 대신 이중화 시스템을 지원할 경우 Primary AAA 서버가 다운 되면 Secondary AAA 서버로 통신을 하는 방법을 사용한다. 통신 프로세서 구현 방법은 통신 프로세서가 송신과 수신을 병행하는 방법과 송신은 메인 프로세서에서 담당하고 수신만 통신 프로세서에서 담당하는 방법이 있다.

- 통신 프로세서에서 송/수신을 담당

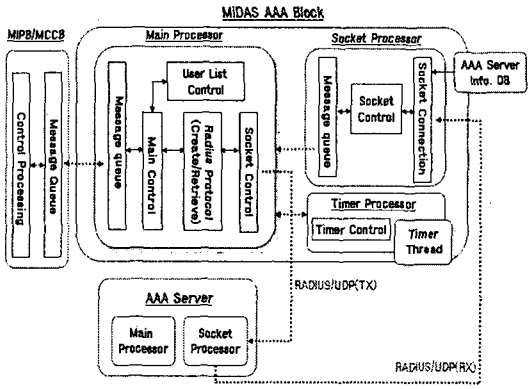
송수신을 모두 담당하는 방법은 다수의 Primary AAA 서버를 지원하는 경우다. AAA 클라이언트와 관련된 모든 서버의 통신 선로에 대한 바인딩정보를 가지고 있으며 각 인증 요구 메시지가 통신 프로세서로 수신되면 수신 메시지에 포함된 Realm 정보를 이용하여 해당 서버의 바인딩정보로부터 통신을 수행한다. 또한 선택된 서버와의 통신이 실패되었을 경우는 임의의 서버나 미리 정해진 서열에 의해 재 전송을 수행한다. 이 때의 재전송은 즉시 전송과 임의 시간 경과 후 전송방법을 택할 수 있다. 다수의 AAA 서버에 대한 정보를 메인 프로세서가 가지고 있어야할 부담을 줄이는 대신 통신 프로세서의 역할을 강조하게 된다.

- 통신 프로세서가 수신만 담당

인증 메시지에 대한 송신은 메인프로세서에서 수행하고 수신만 통신 프로세서가 담당하는 경우로 다수의 AAA 서버보다는 Primary/Secondary 서버로 구성된 이중화 시스템에 적합하다. 이런 경우는 메인 프로세서에서 인증 메시지를 생성하고 즉시 전송을 수행한다. 메인 프로세서에서는 통신 프로세서로 메시지를 전달하는 시간을 줄일 수 있고 AAA 서버의 바인딩 정보에 대한 것만 유지하므로서 통신 프로세서가 수신 메시지에 대한 폴링을 수행하고 메시지 수신 후 메인 프로세서로 인증 처리 결과 메시지를 전송하므로써 통신 프로세서가 단순한 수신 창고로써의 역할을 하도록 한다.

4. MiDAS 의 프로세서 구현 결과

MiDAS 에서의 실제 구현은 각각의 프로세서 구현 방법에서 메인 프로세서는 Simple IP 서비스 사용자와 Mobile IP 사용자를 분리하여 관리하는 리스트 관리 방법으로 구현하였고, 타이머 프로세서는 단순 시작 제어를 통한 5 초 임계 시간을 가지는 스레드 생성방식을 사용하였다. 통신 프로세서는 현재 구현되는 MiDAS 시스템의 망 구성도로 다수의 AAA 서버를 지원하는 것이 아니라 Primary/Secondary AAA 서버 구조를 사용하므로 통신 프로세서가 수신만 담당하는 밥법을 사용하였다. <그림 5> 앞서 언급된 구현 방법을 토대로 작성된 전체 블록도이다.



<그림 5> MiDAS AAA 처리 블록도

메인 프로세서가 AAA 서버와의 네트워크 속도적 측면에서 얻을 수 없는 인증 시간의 증가부분을 각 프로세서의 처리 방법에 따라 어느 정도 보상할 수 있는 방법으로 구현하였다. 또한 하나의 프로세서로 모든 부분을 관리하는 측면도 가능하나 스레드나 프로세서를 생성하는 부분에서 자원의 비 효율적인 면을 간과해서는 안되는 문제가 있으며 초기 생성한 프로세서의 자원 이상을 사용하지 않도록 프로세서의 분리는 필수적이다.

5. 결론

본 논문에서는 AAA 를 설계하기 위한 방법으로 세 가지 블록으로 구성된다는 가정하에 각각의 프로세서 구현 방법에 대하여 알아보았다. 실시간으로 인증을 요구하는 사용자가 증대될수록 인증 서버뿐만아니라 AAA 클라이언트도 인증 대기 시간의 증대가 따를 수 밖에 없다. 이럴 경우 최소한의 시간에 인증 결과를 수집하기 위해서는 다수의 동시 사용자 관리 및 처리 시간을 줄임으로써 사용자가 신뢰할수 있는 시스템을 제공한다. 본 논문에서 제시한 방법으로 구현한 MiDAS 에서의 사용자 인증 처리 블록은 처리 시간적 측면 뿐만아니라 효율적인 사용자 관리에도 중점을 두어 구현하였다.

참고 문헌

[1] 한국전산원, " 유무선 통합을 위한 통신망 진화 방안에 관한 연구" 보고서, pp242-245, 2001.12.

[2] C. Rigney, et al., " Remote Authentication Dial In User Service" , RFC2865, June 2000.

[3] 3GPP2, P.S0001-A, "Wireless IP Network Standard", July 2000

[4] TIA/EIA/IS-835-A, "cdma2000 wireless IP network standard," May. 2000.