주 제

HIP 등 새로운 식별체계 관련 연구 동향

목포해양대학교 김건웅, 서경대학교 송병권, 한국인터넷진흥원 김 원

차례

I. 서 론

현재 전 세계적으로 차세대 인터넷 망을 구축하기 위한 많은 연구와 작업이 진행 중이다. 1990년대 중반 이후 점차 고갈되어 가는 IPv4의 주소 공간 문제를 해결하기 위해 시작된 IPv6 연구에서는 주소 공간 문제뿐만 아니라 보안, 망 서비스 품질 보장, 처리의 효율성 증대를 위해 많은 연구들이 진행 중이며, 이에 관련하여 ICMPv6, DNSEXT 등 많은 연구들이 병행되고 있다[1] [2].

또한 서비스 환경에서도 급격한 변화를 맞고 있는데, 과거에는 고정된 위치에서 하나의 IP 주소를 갖는 경우가 일반적이었지만, 이제는 하나의 시스템이 여러 주소를 가지고 서비스를 제공하는 경우도 생겨나고 있으며, 무선망의 확산은 고정된 위치에서의 서비스가 아닌 이동망 환경에서의 서비스 제공도 고려해야만 하는 시대로 바뀌었다. 이러한 멀티홈(multi-home) 기능과 이동성(mobility)을 제공하기 위한 연구로 Multi6 WG나 Mobile IP에 관련된 연구들을 예로 들 수 있으며 유비쿼터스(ubiquitous) 환경으로 진화하는 현재, 이러한 연구에 대한 중요성은 더욱 커지고 있다[3] [4] [5] [6] [7] [8] [9].

호스트 정체성(HI: Host Identity) 이름공간은 현재의 중요한 이름 공간인 IP(Internet Protocol) 주소와 DNS(Domain Name System) 이름 공간 사이를 채우기 위해 생겨난 것으로서, 호스트 식별자(HI: Host Identifier)들로 구성되는데, 이러한 호스트 정체성은 하나의 호스트를 식별한다. HIP는 시스템들 간에 제한된 형태의 신뢰를 제공하며, 이동성, 멀티홈, 동적 IP 주소 변경을 확장하며, 프로토콜 번역/변환을 지원하고 DOS (denial-of-service)와 같은 종류의 공격을 줄여줄 수 있다[10] [11].

본 고에서는 차세대 인터넷을 위한 여러 식별체계에 대한 연구 중에서 이동성과 멀티홈을 지원하기 위

한 연구들의 현황을 살펴보고 현재 활발히 논의 중인 HIP에 대해 소개한다.

II. 멀티홈과 식별체계

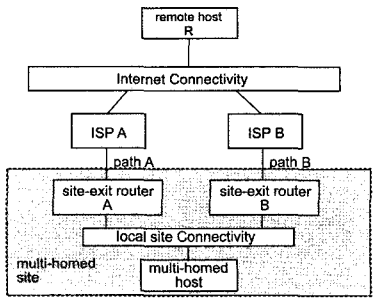
현재 고려되고 있는 멀티홈 형태는 호스트 멀티홈과 사이트 멀티홈이다. 호스트 멀티홈의 경우 호스트가 전역 IP 주소를 2개 이상 가지는 경우인데, 이러한 주소들은 하나의 ISP에 속해 있을 수도 있고, 하나의 인터페이스에 2개 이상의 주소를 부여하거나, 또는 복수개의 인터페이스에 각각의 주소가 부여될 수도 있다. 사이트 멀티홈의 경우 공중 인터넷과의 연결을 2개 이상 가지고 있는 경우를 뜻하는데, 이때 동일한 ISP에 복수개의 연결을 가질 수 도 있고, 복수개의 ISP에 연결을 가질 수도 있다[8] [9].

이러한 멀티홈을 구성하게 되는 이유는 첫째, 하나의 ISP가 다운되어도 다른 ISP를 이용하여 서비스를 계속 할 수 있도록 하는 장애 극복(fault tolerance)과 예비(redundancy) 기능이 가능하며, 둘째, 2개 이상의 ISP를 통해 서비스함으로써 전체 처리율(throughput)을 높일 수 있는 부하 균형(load balancing)이 가능하며, 셋째, 서비스 가격이나 정책에 따라 해당 서비스의 비용이 제일 저렴한 ISP를 선택할 수 있도록 하기 위해서이다.

이러한 멀티홈을 지원하는 것은 IPv4보다는 IPv6에서 훨씬 적합한데, 그것은 복수 주소에 대한 동적인 감지와 상속이 훨씬 용이하기 때문이다. IPv6 호스트가 망에 접속하는 경우 IPv6 라우터들로부터 라우터 광고 메시지를 받을 수 있고, 또한 자동설정 방법으로 각각의 망에 대해 전역적으로 유일한 주소를 할당할 수도 있다. 따라서 IPv6 호스트는 부팅이 되는 순간 멀티홈이 가능하다. 현재 이러한 멀티홈을 IPv6 환경에서 구축하기 위해 multi6 WG이 구성되어 활동하였으며, 이러한 연구 결과를 계승하여 shim6 WG이 활동 중이다.

이러한 멀티홈을 지원하기 위한 여러 가지 방안들을 분류해보면 다음과 같다.

- IPv4 접근 방식 : IPv4의 접근방식과 같이 사이트의 지역 프리픽스를 도메인간 라우팅 시스템에 알리는 방식이다. 이러한 프리픽스가 최상위 라우팅 시스템 DFZ(default free zone)까지 전파되어야 하는데, 이 때문에 확장성 문제가 있다.
- IPv6 라우팅 방식 : IPv6의 라우팅에 새로운 방법을 추가하여 문제를 해결하기 위한 방법들이다.
- 식별자/위치 (2 공간) 방식 : 여러 가지 방식들이 제안되고 있는데, 이들은 모두 노드를 지칭하는 식별자와 인터넷 위치주소를 구별하는 것이다.
- 이동성 방식 : 이동성을 멀티홈의 특별한 경우로 보는 방식으로 Mobile IPv6를 비롯해 여러 가지 방식이 제안되고 있다.
- 트랜스포트 방식 : 기존의 트랜스포트나 다른 상위 계층 프로토콜이 멀티홈을 지원하지 않는데 반해 새로 제안되고 있는 방식들은 주소 변화에 따른 지원이 상위 계층에 포함되어야 한다는 방식이다.



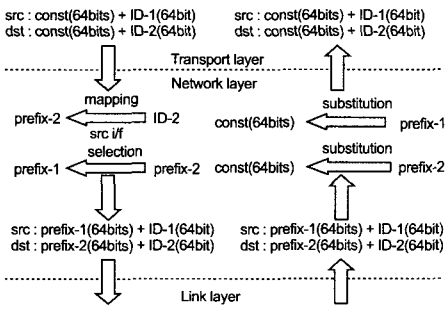
(그림 1) 멀티홈 관련 개체

- 사이트 출구 라우터와 호스트 동작 : 이 부류의 방식들을 다음 (그림 1)에서 보이는 사이트 출구 (Site Exit) 라우터와 호스트들의 동작을 수정하여 멀티홈을 지원하는 것이다.

그 중에서 위치주소와 식별자를 분리하는 방식으로 제안된 것들은 다음과 같다.

- Multihoming without IP Idetifiers (NOID)
- Weak Identifier Multihoming Protocol (WIMP/WIMP-F)
- LIN6
- HIP
- Cryptographic based Identifiers
- Strong Identity Multihoming
- Hashed Based Addressing

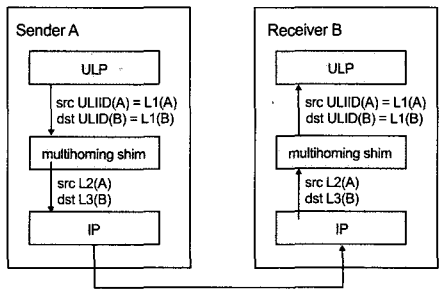
예를 들면 LIN6에서는 (그림 2)에서 보이는 바와 같이, IPv6 주소를 분리하여 일부는 식별자로, 일부는 위치 정보로 이용한다[12].



(그림 2) LIN6에서의 통신 과정

2004년 61차 IETF 회의 때 IPv6 사이트 멀티호밍의 표준을 담당하는 Multi6 워킹그룹에서 여러 가지 제안된 솔루션들 중 식별자와 위치정보를 분리하여 멀티호밍을 지원하는 기고서들을 종합하여 Shim을 제안하였고, 2004년 62차 IETF 회의에서 Multi6 WG를 종료하고, Shim6 워킹그룹에서 Shim의 표준화를 진행하도록 결정하였다.

Shim은 기본적인 개념은 HIP와 유사하지만 오직 IPv6 사이트 멀티호밍만을 지원하기 위한 방법으로 제안되었다. 이것은 HIP와는 달리 다른 에이전트의 도움 없이 오직 호스트에 식별자와 위치 주소를 매핑하는 Shim 계층만 존재하면 멀티호밍이 가능하며 기존 단말들과의 호환성을 보장한다. 이를 위해 먼저 호스트의 식별자는 새로운 이름 공간을 사용하지 않고 기존 IP 주소를 사용한다.



(그림 3) Shim6에서 ULID와 Locator의 관계

(그림 3)은 Shim6에서의 식별자와 위치정보와의 관계를 보여주는데, Shim6에서는 상위 계층의 식별자로 하위 위치정보(IP 주소)들 중 하나를 선택하여 이용하며, 통신 중인 링크의 문제로 인해 다른 링크를 사용해야 하는 경우 위치 주소가 변경되어도 처음 세션을 맺은 식별자는 유지된다[13].

III. 이동성과 식별체계

인터넷에 있는 이동 노드들에게 이동성 제공을 위한 대표적인 연구가 Mobile IP인데, 이것에 대한 요구 사항은 다음과 같다. 먼저 이동 노드는 IP 주소를 바꾸지 않고 인터넷에 링크층의 접촉점을 바꾼 후에

다른 노드들과 통신할 수 있어야 한다. 또한 이동 노드는 이러한 이동 기능들을 하지 않는 다른 노드들과 통신할 수 있어야 한다. 새로운 구조 하에서 작동하지 않는 호스트 또는 라우터에서는 프로토콜의 향상이 요구되지 않으며, 이동 노드의 위치에 대하여, 또 다른 노드들을 갱신하는데 사용되는 메시지들은 원거리 수신 전환 발생을 막기 위해서 인증되어져야 한다[3].

이동 노드가 인터넷에 직접 접속되는 링크는 종종 무선 링크가 될 수 있는데, 이 경우 링크는 낮은 대역폭을 가질 수 있고 일반적인 유선망보다 더 높은 에러율을 가질 수 있다. 또한 이동 노드는 전력소비를 최소화하는 것이 중요하다. 그러므로 링크 상에 보내지는 관리 메시지의 수가 최소화 되어져야 하고 이들 메시지의 크기는 가능한 한 작아야 한다.

Mobile IP는 다음과 같은 새로운 엔티티를 제공한다.

- 이동 노드: 네트워크나 서브네트워크에서 다른 것으로의 접속점을 바꾸는 호스트나 라우터. 이동 노드는 그것의 IP 주소를 바꾸지 않고 위치를 바꿀 수 있다. 이때 IP 주소는 이용 가능한 접속점에 링크층 연결을 떠맡는다.
- 홈에이전트: 이동 노드가 홈에서부터 떨어져 있을 때 이동 노드로 데이터그램을 보내는 이동 노드의 홈 네트워크상의 라우터. 여기서 이동 노드의 현 위치정보를 유지한다.
- 외부에이전트: 경로설정서비스를 이동 노드에 제공하는 이동 노드가 방문한 네트워크상의 라우터. 외부 에이전트는 등록되어진 이동 노드에 대해서 기본형 라우터처럼 행동한다.

홈 네크워크 상에서 장기의 IP 주소가 이동 노드에 부여된다. 이 홈 주소는 고정된 호스트에 주어진 영구 IP 주소와 같은 방식으로 관리되어진다. 이동 노드가 그것의 홈 네트워크에서 멀어졌을 때 COA(care-of address)가 이동 노드와 관계되어지고, 이동 노드의 현재 접속점을 나타낸다. 이동 노드는 모든 IP 데이터그램에 대해 소스 주소로 자신의 홈 주소를 사용한다.

IV. HIP

1. HIP 연구 배경

인터넷은 3가지 중요 요소에 의해 만들어진다. 이것들은 연산 플랫폼(종단), 패킷 전송(망간 연결) 하부구조, 그리고 서비스(응용)이다. HIP 구조에서는 이름에 관련된 연산 플랫폼과 패킷 전송 요소들만을 집중해 고려한다[11].

이들 요소들을 위한 두 가지 중요 이름 공간이 존재하는데, 그것들이 IP 주소와 DNS 이름이다. IP 주소는 호스트의 망 인터페이스의 이름과 위치의 이름의 혼동(confounding)이라고 볼 수 있는데, 여기서 혼동이라고 한 것은 다른 의미 두가지가 하나로 결합되는 과정에서 정보의 손실이 일어난다는 것을 뜻한다.

IP 주소는 일반적으로 망에 연결되어 있을 때만 망 인터페이스를 이름 짓는다. 원래 IP 주소는 고정된 호스트에게 부여하여 장기간 의미를 가질 목적으로 만들어졌다. 따라서 IP 주소를 망 인터페이스의 이름과 위치의 의미로 같이 써도 무방하였다. 그러나 인터넷 망이 확산되면서 많은 인터페이스들이 DHCP, NAT 등의 프로토콜을 이용하여 단기간에 이용되면서 유일하지 않은 IP 주소를 이용하게 되었다. 이 경우 망 인터페이스의 이름과 위치 정보를 동일시 할 수 없는 상황이 발생한다.

도메인 이름은 어떤 연산 플랫폼이나 서비스들에

게 계층적으로 명명된 이름을 제공하며, 각 계층은 한 단계 위 레벨에서 위임을 받는 형식이다. 따라서 도메인 이름에서는 익명성이 없다.

인터넷에서 쓰이는 현재의 두 가지 이름 공간에는 3가지의 치명적인 결함이 있다. 첫째, 동적인 재주소가 바로 관리될 수 없다. 둘째, 일관되고 신뢰할 수 있는 형태로 익명성이 제공될 수 없다. 셋째, 시스템이나 데이터그램에 대한 인증이 제공되지 않는다. 이러한 모든 결함들은 현재의 이름 공간에서 연산 플랫폼이 제대로 명명되지 않는데서 비롯된다.

연산 플랫폼에 대한 독립적인 이름 공간은 망 계층의 진화와 독립적으로, 많은 망 계층들을 거치면서 종단간 작업을 수행하는데 이용될 수 있다. 이것은 이동성이나 리홈(rehoming), 번호재할당(renumbering)으로 인한 망 계층에서의 주소 재할당을 지원한다. 또한 연산 플랫폼에 대한 이름 공간이 공개키 기반 암호화에 기반을 두고 있다면 인증 서비스를 제공할 수도 있다. 만약 이러한 이름 공간이 등록절차 없이 지역적으로 생길 수 있다면 익명성을 제공할 수도 있다.

HIP에서는 이러한 연산 플랫폼을 위한 이름 공간에 대한 요구 사항을 다음과 같이 정리했다.

- 이름공간은 IP 커널에 적용되어야 한다. 이때 IP 커널은 응용들과 패킷 전송 하부구조 사이의 요소이다.
- 이름 공간은 망 계층과 상위 계층을 완전히 분리시켜야 한다. 이 이름은 응용에서 나타나는 모든 IP 주소를 대체하여야 한다. 이것은 현재의 API들을 변화시킬 수도 있다. 결국에는 아마도 새로운 API들이 필요할 것이다.
- 이름 공간의 도입이 관리 하부구조로 권한을 이양하는 것으로 이어지면 안된다. 도입은 아래에서부터 위로 진행되어야 한다.
- 이름은 데이터그램의 헤더와 현재의 프로그래밍 인터페이스 쉽게 포함할 수 있도록 고정된 길이의 표현을 가져야 한다.
- 이름 공간은 프로토콜에서 적용 가능하여야 한다. 이것은 주로 패킷 크기에 관련된 문제이다. 이러한 적용 가능성에는 연산에 관련된 고려사항이 있을 수 있다.
- 이름은 통계적으로 전세계적으로 유일해야 한다. 64비트는 충돌이 일어날 가능성을 충분히 낮게 하는 데에는 적합하지 않다. 따라서 100비트나 혹은 그 이상의 비트가 이용되어야 한다.
- 이름은 현재 존재하는 프로토콜이나 API들에서 이용될 수 있도록 지역화된 축약을 가져야 한다.
- 지역적으로 이름을 생성하는 것이 가능해야 한다. 이것은 분해하기는 어려우면서 익명성을 제공할 수 있다.
- 이름 공간은 인증 서비스를 제공하여야 한다.
- 이름의 수명은 길지만 언제라도 바꿀 수 있어야 한다. 이것은 ACL(Access Control List)에 영향을 미치는데, 짧은 수명은 리스트 유지를 어렵게 하거나 접근 리스트를 중앙에서 관리하는 이름 공간 하부구조를 필요로 하는 경향이 있다.

이러한 요구사항들을 만족하도록 제안된 새로운 이름 공간을 호스트 정체성(Host Identity) 이름 공간이라고 한다. 이러한 이름 공간을 이용한다는 것은 망 계층과 트랜스포트 계층 사이에 새로운 프로토콜 계층 – Host Identity Protocol을 필요로 한다. 이들 이름은 인증 서비스를 제공하기 위하여 공개키 암호화에 기반을 두고 있다. 적절히 설계가 되면 위에 언급한 모든 요구사항들을 만족할 수 있다.

2. 호스트 정체성 이름 공간

호스트 정체성 이름 공간의 하나의 이름, 즉 하나

의 호스트 식별자(HI)는 IP 스택을 가지고 있는 어떤 시스템을 명명할 수 있는, 통계적으로 전세계에 걸쳐 유일한 이름을 나타낸다. 이러한 정체성은 일반적으로 하나의 IP 스택에 연관되어 있는데, 반드시 그렇게 제한되어 있는 것은 아니다. 하나의 시스템은 여러 개의 정체성을 가질 수 있는데, 이때 그들 중 일부는 '잘 알려진' 것들이고 일부는 미발표되거나 익명으로 된 것들이다. 시스템은 스스로 자기 자신의 정체성을 주장할 수도 있고, 어떤 것들은 DNSSEC, PGP, X.509와 같이 정체성을 증명하기 위해 제 3의 인증자를 이용할 수 있다. HI는 처음에 DNSSEC을 통해 인증 받도록 되어 있고, 따라서 구현에서는 최소한 기본적으로 DNSSEC을 지원해야 한다[11] [14].

이론적으로는 통계적으로 전세계에 걸쳐 유일한 어떠한 이름도 HI로 이용될 수 있다. 그러나 이 구조를 제안한 저자들의 의견으로는 공개키 쌍의 공개키가 가장 좋은 HI로 판단하고 있다. HIP 프로토콜 문서에서 언급된 것과 같이 공개키 기반 HI는 HIP 패킷을 인증하고, man-in-the-middle 공격을 방어할 수 있다. HIP의 denial-of-service 공격을 방어하기 위해 데이터그램의 인증이 필수적이므로 HIP의 Diffie-Hellman 교환은 인증되어야 한다. 따라서 실제로는 공개키 기반 HI와 인증된 HIP 메시지만이 지원된다.

호스트 정체성은 인터넷 프로토콜에 두 가지 중요한 기능을 제공한다. 먼저 망과 트랜스포트 계층을 분리시킨다. 이러한 분리는 두 계층의 독립된 진화를 가능하게 한다. 또한 여러 망이 연결된 환경에서 종단간 서비스를 제공한다. 두 번째 기능은 호스트 인증 기능이다. HI가 하나의 공개키이므로 이 키는 IPSec과 같은 보안 프로토콜에서의 인증에서 이용될 수 있다.

호스트 정체성으로 완전히 정의된 유일한 구조는 공개키/비밀키 쌍의 구조이다. 이 경우 호스트 정체성은 그들 중 공개된 요소, 즉 공개키이다. 따라서 호스트 정체성 이름 공간에서 하나의 호스트 정체성을 표현하는 이름은 공개키이다. 따라서 그 비밀키를 소유한다는 것은 정체성 자체를 정의한다. 만약 비밀키를 하나 이상의 노드가 가지고 있다면 정체성은 분산된 것으로 볼 수 있다.

구조적으로는 어떠한 인터넷 이름 관습도 호스트 식별자로 유용하게 쓰일 수 있다. 그러나 암호화되지 않은 이름들은 신뢰도가 높고 손실 위험이 작은 상황에서만 쓰여야 한다. 즉 인증이 필요 없는 곳과 IPSec을 이용하지 않는 곳에만 국한된다. 그러나 적어도 여러 운영 영역에 펼쳐있는 상호 연결된 망에서는, 암호화되지 않은 호스트 식별자를 이용함으로써 스푸핑이 허용될 수 있는 환경이 허용되는 경우는 없다. 따라서 현재의 HIP 문서에서는 오직 공개키만을 이용하는 방안만이 제시되어 있다.

실제로 인터넷 프로토콜에서는 호스트 식별자가 직접적으로 이용되지 않는다. 대응하는 호스트 식별자는 다양한 DNS나 LDAP 디렉토리에 저장되어 있고, HIP 기본 교환(base exchange)에서 교환된다. 다른 프로토콜에서는 HIT(Host Identity Tag)가 호스트 정체성을 나타낸다. 호스트 정체성의 다른 표현인 LSI(Local Scpoer Identifier)도 프로토콜과 API들에서 이용될 수 있다.

HIT는 호스트 정체성을 128비트로 표현한 것이다. 일반적으로 이것은 호스트 식별자의 암호화 해쉬로 얻어지는데, 이러한 HIT의 사용은 고정된 길이로 인해 프로토콜 코딩이 용이하고, 작은 패킷 크기로 인해 전송 과정에서 이득을 볼 수 있다는 점과 암호화 알고리즘에 독립적으로, 일관된 형태로 식별이 가능할 수 있다.

HI나 HIT를 사용자 데이터 패킷의 프로토콜 헤더에 운반하는 것은 패킷에 대한 오버헤드를 증가시킨다. 따라서 모든 패킷에 그것들을 운반하는 것이 아니고, 데이터 패킷과 대응하는 HI와 매핑하는 다른 메

커니즘이 사용된다. 따라서 사용자 데이터 패킷에 추가되는 헤더 없이 HIP를 이용할 수 있다. 예를 들면 데이터 트래픽의 보호를 위해 ESP가 이용된다면, ESP 헤더의 SPI(Security Parameter Index)가 암호화된 데이터 패킷과 해당 HIP 연관을 매핑시킬 수 있다.

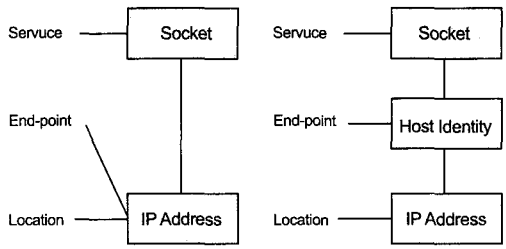
LSI 역시 호스트 정체성의 다른 표현인데 32비트의 길이를 갖는다. 이것은 현재의 프로토콜과 API들에서 호스트 정체성을 이용하고자 할 때 유용하게 쓰일 수 있으며, 약점은 작은 크기로 인해 전 세계적으로 이용할 수는 없고, 지역적으로만 이용이 가능하다는 점이다.

공개된 호스트 식별자는 DNS에 저장되어야 한다. 미발표된 호스트 식별자는 통신하는 호스트들을 제외한 다른 장소에는 저장되지 않는다. 공개된 HI는 새로 정의할 새로운 RR 타입으로 저장된다. 이러한 RR 타입은 IPSecKEY RR과 아주 유사하다.

또 다른 대안으로 혹은 호스트 식별자를 DNS에 저장한 것에 추가하여 그것들은 다양한 PKI(Public Key Infrastructure)에 저장될 수 있다. 이러한 적용들은 순수한 호스트 식별이 아닌 다른 응용에서도 쓰일 수 있다.

3. 새로운 프로토콜 스택

호스트 정체성과 현재의 인터넷을 비교하면, IP 주소는 라우팅 방향 벡터와 인터페이스 이름이 혼동된 형태이다. IRTF 이름 공간 연구그룹의 용어 정의에 따르면 IP 주소는 위치정보(locator)와 종단 식별자(end-point identifier)의 역할을 동시에 수행하고 있다. HIP 구조에서는 종단 이름과 위치정보가 분리되어 있다. 여기서 호스트 정체성에 기반을 둔 종단 식별자는 인터페이스 이름과 다르다는 점이 중요하다. 하나의 호스트 정체성은 여러 인터페이스를 통해 접근할 수도 있다. (그림 4)는 현재의 인터넷 구조와 HIP 구조를 비교하고 있다[11] [14].



(그림 4) 현재의 인터넷과 HIP 구조의 논리 개체 비교

따라서 HIP에서는 구조적으로 다른, 트랜스포트 계층 프로토콜과의 바인딩을 제공한다. 즉 트랜스포트 계층의 연관(TCP 연결, UDP 연관)은 더 이상 IP와 묶이지 않고 호스트 정체성과 연관된다.

하나의 물리적인 컴퓨터에서 여러 개의 논리적인 종단을 가지는 것이 가능한데, HIP에서는 각 종단이 각기 다른 호스트 정체성을 가질 수 있다. 또한 트랜스포트 연관이 호스트 정체성에 묶이므로, HIP는 프로세스 이주(process migration)와 클러스터 서버(clustered server)를 지원한다. 또한 호스트 정체성이 하나의 물리적인 컴퓨터에서 다른 컴퓨터로 이동될 수 있다면 클라이언트 종단에 변화를 주지 않고도 클러스터 기반 서비스를 제공할 수 있다.

4. HIP 프로토콜

HIP에서 HIP 교환을 시작하는 측이 초기자(Initiator)가 되며, 상대측은 응답자(Responder)로 지칭한다. 이러한 구별은 기본 교환(base exchange)이 끝나면 사라진다. HIP 기본 교환은 초기자와 응답자 사이의 상태 설정을 관리한다. 첫 번째 패킷 I1이 교환을 초기화하고, 이어지는 3개의 패킷 R1, I2, R2가 세션키를 생성하는 Diffie-Hellman

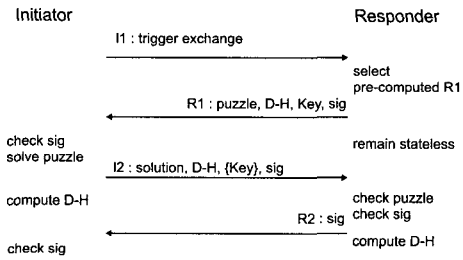
표준 키 교환을 구성한다. D-H 키 교환 중에 하나의 키 요소가 생성되는데, HIP 연관은 이렇게 생성된 키 요소로부터 유도된다. 만약 또 다른 키들이 필요한 경우, 역시 같은 키 요소에서 유도될 수 있을 것으로 예상된다[14].

처음에 초기자는 트리거 패킷인 I1을 응답자에게 보낸다. 이 패킷에는 초기자의 HIT만을 담고 있고, 가능한 경우 응답자의 HIT를 담을 수 있다. 어떤 경우에는 다른 트리거를 보낼 수도 있는데, 이 경우에는 응답자가 R1 패킷을 보냄으로써 프로토콜이 시작된다.

두 번째 패킷인 R1이 실제 교환을 시작한다. 여기에는 퍼즐(puzzle)을 담고 있는데, 교환을 계속하기 위해서는 초기자가 반드시 이것을 풀어야 한다. 이러한 퍼즐의 난이도는 초기자의 신뢰도, 현재의 부하 상태 또는 다른 요소에 따라 조정된다. 여기에 덧붙여 R1에는 D-H 파라메터와 메시지를 보호하는 서명을 포함하고 있다. 어떤 필드들은 미리 만들어진 R1을 지원하기 위해 서명 밖에 있을 수 있다.

I2 패킷에서는 초기자가 받은 퍼즐에 대한 해답을 보여야 한다. 이때 정답이 아닌 경우 I2는 폐기된다. 또한 I2에는 응답자를 위한 정보를 운반하는데 필요한 D-H 파라메터를 담고 있고, 전송자에 의해 서명된다.

R2 패킷은 기본 교환을 완료하며, 서명되어져 있다. (그림 5)는 기본 교환을 보여주고 있다.



(그림 5) HIP Base Exchange

5. 종단 호스트 이동성과 멀티홈 기능 제공

HIP는 트랜스포트 계층과 IP 계층을 분리하고, 트랜스포트 계층의 연관을 호스트 정체성과 묶었다. 결과적으로 HIP는 낮은 인프라 비용으로 이동성과 멀티홈을 제공할 수 있다. HIP 이동성은 IP 주소 변화를 지원하는 것인데, PPP, DHCP, IPv6 프리픽스 재할당, NAT 매핑 등의 여러 이유로 IP 주소가 동적으로 변화하는 것을 지원한다면 이동성이 있다고 볼 수 있다[15].

HIP에서는 상대방에게 접근 가능한 다른 IP 주소를 알려주기 위해 LOCATOR 파라메터를 이용한다. LOCATOR는 주소의 일반화된 표현이다. 이러한 LOCATOR는 망의 종단 인터페이스를 정의할 수도 있고, 추가로 터널링이나 디멀티플렉싱 정보가 추가될 수도 있다.

호스트가 다른 주소로 이동하는 경우, 호스트는 LOCATOR 파라메터를 가지고 있는 HIP UPDATE 패킷을 보냄으로써 상대방에게 새로운 주소를 통보할 수 있다. 이러한 UPDATE 패킷은 상대방으로부터 확인되고 재전송으로 보호되고 있다. 상대방은 서명과 키를 이용한 해쉬를 통해 UPDATE 패킷의 내용을 인증할 수 있다. 이때 호스트는 새로운 보안 연관을 설정하기 위해 키를 재생성하도록 판단할 수 도 있으며, 이러한 모든 일들은 UPDATE 패킷의 옵션을 추가하여 가능하다.

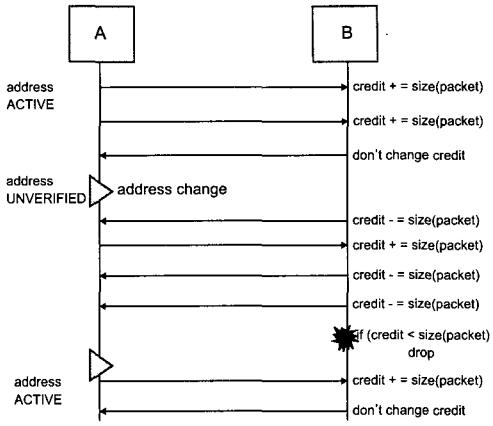
마찬가지로 하나의 시스템이 전세계적으로 경로 설정이 가능한 IP 주소를 동시에 하나 이상 가질 수 있다면, 멀티홈으로 볼 수 있다. HIP는 여러 개의 IP 주소를 하나의 호스트 정체성으로 묶을 수 있고, 하나의 주소가 사용이 불가능하거나 다른 주소를 선호한다면, 현재 존재하고 있는 트랜스포트 연결을 쉽게 다른 주소로 옮겨갈 수 있다.

통신이 진행 중인 노드가 이동한다면, 주소 변화는 보다 간단하다. 이동 노드의 상대방은 HIP나 IPSec으로 보호된 패킷만을 받아들이고 출발지 주소 부분은 무시할 수 있다. 이때 이동 노드는 HIP 주소 재설정 패킷을 통해 상대방에게 새로운 주소를 알려주어야 하고, 상대방은 이들 주소를 통해 접근 가능한지를 확인해야 한다.

다른 HIP 호스트로부터 새로운 주소가 담긴 LOCATOR를 받은 경우, 호스트는 그 주소들로 접근이 가능한지 반드시 확인할 필요는 없다. 그러나 플로딩 같은 악의적인 공격에 대비하기 위해서 HIP 호스트는 새로운 주소로 상대방이 접근 가능한지를 확인할 필요가 있다. 주소 확인은 새로운 주소로 추정이 불가능한 정보를 보내고 상대방으로부터 그것을 받았다는 것을 확인할 수 있는 응답을 기다리는 것으로 가능하다. 이 과정에는 새로운 임시정보(nonce)의 교환, 새로운 SPI의 생성, 새로운 SPI를 통한 데이터 도착 관찰 과정 등이 포함된다.

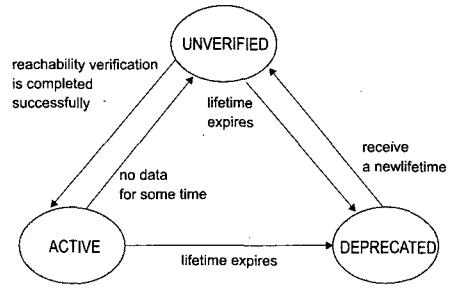
신용기반(Credit-based) 권한부여는 호스트가 LOCATOR에 포함된 주소에 대한 확인이 되지 않은 경우에도 새로운 LOCATOR를 안전하게 이용할 수 있도록 한다. 이 방법은 상대방에게 받은 패킷의 양을 신용 값으로 누적하는 것인데, 상대방이 주소를 바꾸었다고 통보하는 경우, 주소 확인이 되기 전에는 그 전 주소에서 누적된 신용값을 기반으로 해서 패킷을 보내고, 보낸 만큼 신용값을 감소시킨다. 이때 새로운 주소로부터 데이터가 오면 그만큼 신용값을 증가시키며, 주소 확인이 끝나면 더 이상 신용값을 감소시키지 않는다. (그림 6)은 신용기반 권한 부여의 동작 원리를 보여준다.

이런 방법은 플로딩 같은 공격을 하기 위해 악의로 다른 주소를 알려준 경우에도 신용값에 따른 양의 데이터만을 보내고 더 이상의 데이터를 보내지 않기 때문에 그러한 위협을 방지할 수 있다.



(그림 6) 신용기반 권한부여 동작 원리

LOCATOR의 상태는 3가지가 있는데, UNVERIFIED는 접근 확인이 아직 안된 상태를 뜻하고, ACTIVE는 접근확인이 끝나서 이용하고 있는 상태, DEPRECATED는 수명이 끝난 상태를 뜻한다. (그림 7)에서는 LOCATOR의 상태변화를 보여준다.



(그림 7) LOCATOR 상태변이도

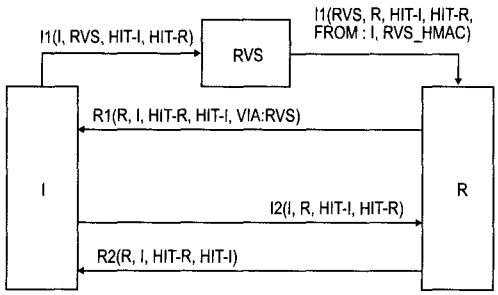
LOCATOR 정보를 보내는 시점은 호스트의 지역 정책에 따르는데, IP 주소가 변경되면 LOCATOR 정보를 보내도록 권장되고 있으며, 이때 보내는 IP 주소는 수초 이상 지속될 것이라는 확신이 있어야 한다.

HIP를 지원하는 호스트는 I1을 제외한 다른 HIP

패킷의 LOCATOR를 처리할 준비가 되어 있어야 하는데, LOCATOR 파라메터가 있는 HIP 패킷을 받는 경우 다음 절차를 따른다. (1) 먼저 LOCATOR 안의 주소들이 적법한 주소인지를 확인한다. (2) LOCATOR 안의 주소들과 이미 연계된 SPI가 있는지 확인한다. 연계된 IP 주소들인 경우 수명을 갱신한다. DEPRECATED 상태였던 것들은 UNVERIFIED로 상태를 변경시킨다. 연계되지 않은 주소들인 경우 주소를 추가하고 UNVERIFIED 상태로 설정한다. (3) LOCATOR와 ESP_INFO가 같이 있는 경우 ESP_INFO 파라메터를 처리한다. (4) LOCATOR 파라메터 안에 없는 IP주소들의 상태를 모두 DEPRECATED로 변화시킨다.

이동 노드에 처음 접근하는 것은 더 많은 점이 고려되어야 하는데, HIP 교환을 하기 전에 시작 노드는 이동 노드에 어떻게 접근할 수 있는지를 알고 있어야 한다. 자주 이동하지 않는 노드의 경우에는 DNS에 자신의 정보를 갱신하기 위해 Dynamic DNS를 이용할 수도 있지만, 랑데부 서비스라고 부르는 새로운 대안도 제시되어 있다[16]. 랑데부 메커니즘에서는 클라이언트가 처음 접근하는 창구 역할을, 제3자인 랑데부 서버(RVS: Rendezvous Server)가 담당한다. RVS의 클라이언트들은 그들의 HIT->IP 주소 매핑 과정에서 RVS를 이용하기 위해 HIP Registration Protocol을 이용하는 HIP 노드들이다. 이러한 등록 절차가 완료되면 다른 HIP 노드들은 그들이 접근하고자 하는 노드의 IP 주소가 아닌 RVS의 IP 주소로 기본 교환을 수행할 수 있다.

HIP 노드는 그들 위치 변화에 상관없이 상대방과 연결될 수 있기를 원할 수 있다. HIP 구조에서 RVS는 이러한 호스트들의 HIT와 현재 IP 주소 등록을 담당한다. RVS는 이러한 이들의 HIT를 가지고 접근하는 HIP 패킷을 등록된 노드의 IP 주소로 전달한다. HIP 노드가 RVS에 등록되는 경우, HIPRVS DNS RR로 DNS에 등록되어 있어야 한다. (그림 8)은 RVS가 참여하는 HIP 기본 교환을 보여준다.



(그림 8) RVS의 기본교환 전달

여기서 HIP 노드 R은 이미 HIP 등록 프로토콜로 HIT와 현재 IP 주소가 RVS에 등록되어 있는 상태이다. 초기자 I가 응답자 R과 연결을 설정하고자 하는 경우, 기본 교환의 I1은 R의 IP 주소 중 하나로 보내지거나 R의 RVS 중 하나로 보내질 수 있다. 여기서 RVS로 보내지는 경우, RVS는 I1이 자신의 것이 아님을 확인하고 전달 필요성이 있는지 확인한다. 여기서 I1이 R의 것임을 확인하고 R에게 전달한다. 이것을 받은 R은 RVS의 지원 없이 바로 I에게 응답한다. 현재까지 RVS의 클라이언트는 항상 응답자 역할을 담당하는데, 앞으로 NAT나 파이어월의 경우 확장될 필요도 있다.

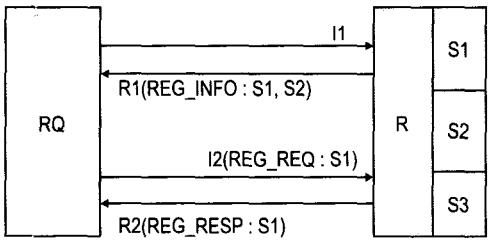
랑데부 메커니즘은 두 노드가 동시에 주소를 바꾸는 경우에도 반드시 필요하다. 이러한 경우는 두 노드가 모두 이동 노드이면서 동시에 이동을 한 경우, 둘 중 하나가 잠시 연결이 끊어진 경우, 또는 또 다른 이유로 발생할 수 있다.

6. HIP 등록 메커니즘

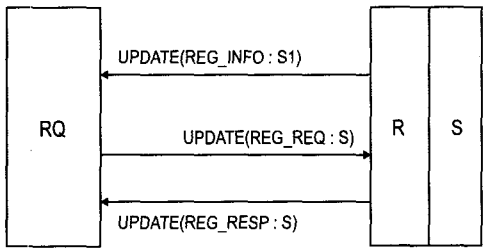
HIP 등록 메커니즘은 호스트들이 랑데부 서버나 미들박스와 같은 서비스를 등록할 수 있도록 해준다.

등록기관으로 역할을 할 수 있고, 그 역할을 하고자 하는 호스트는 모든 기본 교환과정에서 R1 패킷에 REG_INFO 파라메터를 포함해서 보낸다. 만약 제공하고자 하는 서비스가 일시적으로 제공할 수 없는 상태가 되면, 비어있는 REG_INFO 파라메터를 보낸다. 나중에 이러한 서비스가 가능한 경우 UPDATE 패킷에 새로운 REG_INFO 파라메터를 실어서 보내어 관련된 모든 호스트가 이를 알게 한다[17].

서비스 등록을 원하는 요청자는 다음 그림들과 같이 I2 또는 UPDATE 패킷에 REG_REQUEST 파라메터를 실어서 등록기관에게 보낸다.



(그림 9) R1에 서비스 정보를 포함



(그림 10) UPDATE에 서비스 정보를 포함

요청자와 등록기관 사이에 HIP 연관이 없는 경우에는 먼저 I2 패킷에 REG_REQUEST를 보내야 한다. 이러한 방식은 등록기관과 요청자 사이에서 교환될 메시지의 수를 줄여준다.

등록기관은 REG_REQUEST를 담고 있지 않은 HIP 연관을 REG_REQUIRED 타입 통고를 R2에 포함시켜 보냄으로써 HIP 연관을 종료시킬 수 있다. 이 경우 HIP 연관은 생성되지 않는다.

등록이 요청된 경우 등록기관은 I2에 포함된 HIT를 기반으로 해서 요청자의 신원 인증을 한다. 이때 등록기관은 자신의 정책에 따라 호스트 정체성이 등록기관에 요청된 서비스를 등록할 수 있는지 여부를 판정한다. 이러한 권한확인 작업은 요청된 서비스나 등록기관의 내부 정책에 따른다.

7. HIP를 위한 DNS 확장

많은 응용에서 노드들을 명명할 때 DNS 서비스를 이용하고 있으므로, HIP 정보를 DNS를 통해 유지하는 것은 단순한 일이다. HIP에서는 IP 주소를 종단간 호스트의 통신에서만 쓰이고, ULP(Upper Layer Protocol)나 다른 응용들은 HI나 HIT를 대신 쓰도록 하고 있다. 결과적으로 도메인 이름을 HI로 번역하는 기능이 필요한데, 이때 DNS를 이용하면 아주 단순해진다. HIP를 위해 두개의 RR을 새로 추가했는데, 그것들은 HIPHI와 HIPRVS이다[18].

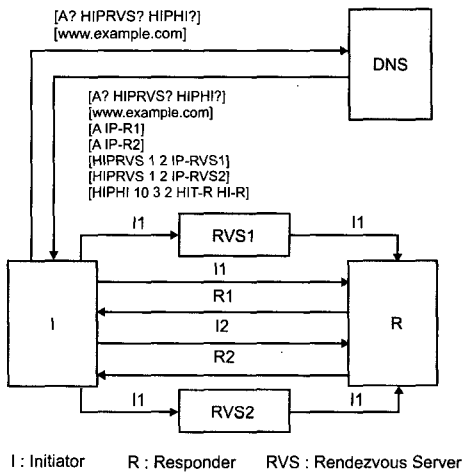
DNS 서비스를 이용할 때 DNSSEC을 이용하지 않는 경우 Man-in-the-Middle 공격에 취약하기 때문에 DNSSEC을 이용하지 않는 HIP 핸드쉐이크 역시 Man-in-the-Middle 공격에 취약하다는 점은 주의해야 한다.

HIP에서는 대부분의 응용과 ULP가 실제 패킷을 운반하는 IP 주소를 모르는 상태이다. 결과적으로 HIP 노드는 대부분의 응용과 ULP에게 투명하게 다수의 IP 주소를 이용하는데 따른 장애 극복, 예비, 이동성 또는 번호 재설정 등의 이점을 가질 수 있다. 이런 경우 노드는 다음과 같은 정보를 DNS에 저장하고자 할 수 있다.

- IP 주소 집합(A 또는 AAAA RR)
- HIPHI RR들을 통한 HI와 HIT

• HIPRVS를 통한 RVS의 IP주소 또는 DNS 이름

(그림 11)은 HIP에서 DNS를 이용하는 시나리오를 보여주고 있다.



(그림 11) HIP에서 DNS를 이용하는 시나리오

8. HIP와 IPsec

HIP를 구현하는 적당한 방안은 IPsec이 실제 데이터 트래픽을 운반하는 것이다. 현재 완전히 정의된 방법은 IPsec ESP(Encapsulated Security Payload)뿐인데, 앞으로는 다른 방법도 개발될 수 있을 것이다[19].

실제로 HIP 기본 교환은 암호화된 호스트 식별자를 통해, 종단간 ESP 교환이 가능하도록 한쌍의 ESP 보안 연관(SA : Security Association)을 설정할 수 있다. 이론적으로는 IKEv2와 같은 현재의 암호화 프로토콜과 호스트 식별자를 같이 쓸 수 있지만, HIP는 필요한 SA를 설정하기 위해 새로운 프로토콜을 정의했다.

이것은 역사적으로나 구조적으로 여러 가지 이유가 있는데, 첫째 IKE(IKEv2)는 미들 박스(middle box)에 대한 고려가 없는 상태에서 개발되었지만, HIP는 미들 박스에 적합한 구조이며, 둘째, ESP의 IPsec SPI(Security Parameter Index)는 HIT의 단순 축약 형태로 볼 수 있고, HIT마다 두 쌍의 SA들(SPI들)을 요구하고 있으며, 그대로 이용하는 경우 키 관리 메커니즘의 기능을 감소시키는 결과를 가져올 수 있다. HIP가 게이트웨이나 BITW(Bump-in-the-Wire) 구현을 고려하지 않고, 호스트간 사용만을 고려하고 있으므로, ESP 트랜스포트 모드만 지원된다. 따라서 HIP와 IKE는 서로 경쟁 관계가 아니며, 서로 보완적인 관계로 볼 수 있다.

V. 결 론

IETF HIP WG에서 작업 중인 HIP는 IP 주소가 가지고 있는 종단 식별자 기능과 위치 지시자 역할을 분리시키기 위한 시도로서 이를 통해 이동성, 멀티홈 제공, 익명성 제공, 인증 서비스를 동시에 지원하고자 연구되고 있다. 현재는 HIP 구조와 HIP 프로토콜 자체에 관한 문서 보완이 진행 중이며, 그 외에도 랑데부 메커니즘과 DNS의 확장, HIP를 이용한 이동성, 멀티 홈 제공 시나리오 등이 연구 중이다. HIP의 기본 교환의 경우 이미 5가지 이상의 구현 결과물이 발표된 상태이며, 하부 구조와 관련된 연구가 활발히 진행 중이다.

현재 식별자와 주소 정보를 분리하는 다른 노력으로 Shim6 WG의 연구도 주목할 필요가 있다. 멀티홈을 지원하기 위한 노력으로 진행 중인 이 연구는 HIP와 유사한 점도 있지만, 멀티 홈 제공으로 문제를 한정함으로써 보다 단순한 해결방안 제공이 가능하고 호환성 확보가 용이하다는 장점이 있다.

현 상태에서는 어느 접근방식이 채택될지 판단하기 어려운 상태이며, 따라서 앞으로 이들 HIP WG와

Shim6 WG의 연구를 주시하며, 이들 연구에 대한 의견이 모여지면 실제 도입을 추진하여 기술을 선점할 필요성이 있다고 판단된다.

[참고문헌]

[1] http://www.ietf.org/ipv6-charter.html

[2] http://www.ietf.org/dnsext-charter.html

[3] http://www.ietf.org/mip4-charter.html

[4] http://www.ietf.org/mip6-charter.html

[5] http://www.ietf.org/mipshop-charter.html

[6] http://www.ietf.org/monami6-charter.html

[7] http://www.ietf.org/nemo-charter.html

[8] http://www.ietf.org/shim6-charter.html

[9] http://www.ietf.org/multi6-charter.html

[10] http://www.ietf.org/hip-charter.html

[11] R. Moskowitz, P. Nikander, "Host Identity Protocol (HIP) Architecture", RFC4423, IETF, 2006

[12] Mitsunobu Kunishi, Masahiro Ishiyama, Keisuke, Uehara, Hiroshi Esaki, and Fumio Teraoka, "LIN6: A New Approach to Mobility Support in IPv6," Proceedings of the Third Inernational Sympsium on Wireless Personal Multimedia Communications, Nov. 2000.

[13] E. Nordmark, M. Bagnulo, "Level 3 multihoming shim protocol", draft-ietf-shim6-proto-06, IETF, 2006

[14] R. Moskowitz, P. Nikander, P. Jokela , T. Henderson, "Host Identity Protocol", draft-ietf-hip-base-06, IETF, 2006

[15] T. Henderson , "End-Host Mobility and Multihoming with the Host Identity Protocol", draft-ietf-hip-mm-04, IETF, 2006

[16] J. Laganier, L. Eggert, "Host Identity Protocol (HIP) Rendezvous Extension", draft-ietf-hip-rvs-05, IETF, 2006

[17] J. Laganier, T. Koponen, L. Eggert, "Host Identity Protocol (HIP) Registration Extension", draft-ietf-hip-registration-02, IETF, 2006

[18] P. Nikander, J. Laganier, "Host Identity Protocol (HIP) Domain Name System (DNS) Extensions", draft-ietf-hip-dns-08, IETF, 2006

[19] P. Jokela, R. Moskowitz, P. Nikander, "Using ESP transport format with HIP", draft-ietf-hip-esp-04, IETF, 2006

김건웅

1990년 고려대학교 전자전산공학과 (공학사)
1994년 고려대학교 대학원 전자공학과 (공학석사)
1998년 고려대학교 대학원 전자공학과 (공학박사)
1999년 ~ 현재 목포해양대학교 해양전자 · 통신 공학부 부교수
관심분야 : 네트워크 프로토콜, IPv6, 인터넷 식별체계

송병권

1984년 고려대학교 전자공학과 (공학사)
1986년 고려대학교 대학원 전자공학과 (공학석사)
1995년 고려대학교 대학원 전자공학과 (공학박사)
1984년 ~ 1997년 삼성종합기술원 선임연구원
1995년 ~ 현재 서경대학교 정보통신공학과 교수
관심분야 : High-speed Network, 분산처리시스템, Mobile computing

김 원

1984년 한양대학교 전자공학과 졸업.
1989년 한양대학교 대학원 전자공학과 공학석사
2002년 : 경희대학교 전자공학과 공학박사
1984년 ~ 1987년 국방과학연구소(연구원)
1989년 ~ 1992년 (주)데이콤(주임연구원)
1992년 ~ 1999년 한국전산원(팀장)
1999년 ~ 현재 한국인터넷진흥원(기술개발단장)
관심분야 : 로봇에이전트, 컴퓨터네트워킹, 차세대인터넷식별자