한국사물인터넷학회논문지 Vol. 2, No. 2, pp. 37-49, 2016 http://dx.doi.org/10.20465/KIOTS.2016.2.2.037

SDN환경에서 DHT를 이용한 네트워크 기반 위치자-식별자 분리 기술

이찬행[1], 민성기[1], 최창원[2]
[1]고려대학교 정보대학, [2]한신대학교 컴퓨터공학부

A Network-based Locator-Identifier Separation Scheme using DHT in SDN

Chan-Haeng Lee[1], Sung-Gi Min[1], Chang-Won Choi[2]
[1]College of Information, Korea University,
[2]Division of Computer Engineering, Hanshin University

요 약 IP 주소는 기존 인터넷에서 단말과 어플리케이션을 바인딩하기 위해 식별자와 위치자의 역할을 모두 갖고 있다. 이러한 바인딩을 제거하기 위해 몇몇 프로토콜들이 제안되었고, 제안된 대부분의 연구들은 기존의 인터넷과 호환성을 유지하기 위해 IPv6기반의 단말 식별자를 사용하였다. 하지만, 단말 식별자는 일반적으로 IP 도메인 상에서 라우터블하지 않은 주소이기 때문에 기존의 표준 IPv6 라우터들은 이와 같은 단말 식별자를 처리할 수 없었고, 때문에 단말 식별자는 보통 단말에서 위치자로 변경될 수 밖에 없었다. 본 논문에서는 분산 해시 테이블(DHT)을 이용한 단말의 위치자와 식별자를 분리하는 기술을 제안하였다. 제안한 기술은 기존의 네트워크를 가상의 Host Identity 도메인과 실제의 IP 도메인으로 나누어 host Identity 도메인을 IP 도메인 위에서 오버레이된 방식으로 사용할 수 있도록 설계함으로써, 단말 식별자를 자신의 주소로 갖는 패킷을 직접 다룰 수 있고, 가상의 도메인을 통해 라우터블하지 않은 식별자를 라우터블하게 처리할 수 있다. 제안 기술의 평가는 기존의 연구와 시그널링과 패킷 전송에 소모되는 비용을 비교하였으며, 비교 결과는 현대의 모바일 위주의 환경에는 제안한 기술이 더 적합하다는 것을 보여주고 있다.

주제어 : SDN 위치자 식별자 분리, 분산해시 테이블, IPv6, 네트워크 기반

Abstract An IP address is used as a host identifier and a locator to bind hosts and applications to their location in existing Internet. Several protocols are proposed to eliminate this binding. Most of these protocols use IPv6-based host identifiers to maintain compatibility with existing Internet, but these identifiers cannot be handled by standard IPv6 routers because such identifiers are unroutable. Therefore, host identifiers need to be usually converted to locators at hosts, and the standard IPv6 protocol should be modified to interoperate with these protocols. In this paper, we propose a network-based host identifier locator separating scheme in software-defined networking. The proposed scheme separates the underlying network into Host Identity and IP domains in order to directly forward unroutable identifiers. The Host Identity domain operates as an overlaid network over IP domain, and it makes the unroutable identifiers to be routable using distributed hash table based routing strategy. For the evaluation, we compared the proposed scheme with the previous scheme using signaling costs and packet delivery costs. The result shows that the proposed scheme is more suitable in the recent mobile-based environments.

Key Words : SDN, ID-LOC separation, DHT, IPv6 address, network-based

교신저자 : 민성기(sgmin@korea.ac.kr)
접수일자 2016년 6월 23일

1. 서론

최근 인터넷을 이용한 네트워크 환경이 무선 모바일 환경으로 변화됨에 따라 모바일 시장의 성장 또한 급격하게 증가하고 있다. 모바일 장치의 폭발적인 증가로 인해 단말의 이동성은 모바일 환경을 위한 필수적인 요소 중의 하나가 되었다. 이러한 모바일 장치는 상호 통신을 위해 IP 주소를 사용하게 되며, IP주소는 모바일 장치를 식별하고 모바일 장치가 위치한 액세스 네트워크를 식별하기 위해 사용된다. 뿐만 아니라, 네트워크에서 단말과 어플리케이션을 바인드(bind)하기 위해 사용되기도 한다. 초기의 인터넷(Internet)은 유선의 고정된 네트워크 환경만을 고려하여 설계되었기 때문에 단말과 어플리케이션의 바인딩은 이러한 환경에서는 장점으로 작용하였으나, 오늘날의 인터넷 환경은 모바일이라는 기능적인 면을 주로 고려하고 있기 때문에 이동성이나 멀티호밍(Multihoming), 확장성과 같은 문제를 야기하게 되었다. 특히, 오늘날의 모바일 환경은 스마트 장치를 비롯한 사물인터넷 영역으로 확대되어감에 따라 네트워크를 사용하는 모바일 장치의 수가 기하급수적으로 증가하고 있기 때문에 모바일 장치의 배치나 라우팅과 어드레싱(Addressing)에 대한 확장성(Scalability)문제도 발생하고 있으며, 백본 지역의 라우팅 테이블이 기하급수적으로 증가하게 되는 현상도 발생하고 있다. 그렇기 때문에 IP주소의 이러한 바인딩은 현재의 모바일 지향적인 네트워크 환경에는 적합하지 않다. 미래 인터넷에서 보다 많은 모바일 장치의 라우팅과 어드레싱의 확장성 및 이동성을 제공하기 위해서 IP 주소의 두 가지 기능인 ID로서의 기능과 LOC로서의 기능이 분리되어야 할 필요가 있다.

IP주소의 두 가지 역할의 구분을 위한 다양한 프로토콜들이 제안되었다[1-4]. 이러한 제안 기술들은 ID와 LOC로써 두 개의 새로운 이름공간(namespace)을 사용하고 있으며, 크게 호스트 기반의 프로토콜과 네트워크 기반의 프로토콜로 나누어 진다. Host Identity Protocol(HIP), Shim6와 같은 호스트 기반 프로토콜들은 단말 기기의 프로토콜 스택을 수정해야하고 ID와 LOC의 매핑 정보를 관리 및 유지하기 위한 랑데부(Rendezvous) 서버가 반드시 필요하다. 그에 반해, 네트워크 기반의 프로토콜들은 호스트의 프로토콜 스택을 수정할 필요가 없으며, 모든 시그널링과 관련된 프로세스는 네트워크를 구성하는 상위 구성요소(entities)에서 수행되기 때문에 네트워크에 연결된 단말 장치들은 어떠한 시그널링 절차에도 참여할 필요가 없다. ID와 LOC의 매핑 또한 라우터와 같은 네트워크의 다른 구성체에 의해 처리되어지고, 매핑 정보에 따라 터널링(tunneling)을 이용하여 패킷을 전달할 수 있다. 하지만 터널링을 사용함에 따라 더 많은 대역폭이 필요하게 되고, 패킷의 캡슐화(Encapsulation)와 역캡슐화(Decapsulation)에 따른 프로세싱 오버헤드도 필요할 수밖에 없다는 문제가 있다.

이러한 문제를 해결하기 위해 본 논문에서는 Software-Defined Networking(SDN)과 Distributed Hash Table(DHT)를 이용한 네트워크 기반의 ID와 LOC 분리 기술을 제안하였다[5-7]. SDN에서는 물리적 네트워크와 가상 네트워크에 서로 다른 라우팅 기법을 운영하는 것이 가능하기 때문에, 네트워크를 더 유연하게 만들 수 있다는 점에 집중하여, 가상 네트워크에 대해서 DHT기반의 라우팅 알고리즘을 적용하고, 물리 네트워크에 대해서는 기존의 IP 라우팅 알고리즘을 적용함으로써 오버레이(Overlay) 형태의 네트워크를 구성하였다[5],[6]. 오버레이 된 네트워크를 위한 네트워크 구조 및 라우팅 알고리즘으로 DHT기반 알고리즘의 하나인 Contents Addressable Network(CAN)의 구조와 알고리즘을 적용하였다[7].

제안 기술에서 단말들은 Host Identity Tag(HIT)기반의 ID(HIT)를 주소로 사용하는 패킷을 전송할 수 있으며, 어떠한 ID-LOC 변환 프로세스에도 관여하지 않는다. 제안한 네트워크에 존재하는 컨트롤러들은 목적지로 패킷을 포워딩할 수 있도록 HIT-LOC 매핑정보를 유지/관리하고, 컨트롤러들은 자신이 관리하는 OpenFlow 스위치의 Flow table을 설정할 수 있도록 전용의 안전한 채널을 이용하여 OpenFlow 메시지를 전송할 수 있다. 기존의 네트워크 기반 프로토콜들에서 보이는 터널링을 제거하기 위해서, 본 연구에서는 주소 필드(address field)에 대한 필드 교체(field replacement)를 사용하였다. 제안 기술의 비용 모델을 제시하고, 기존의 제안된 기술과 비교/분석을 위한 시그널링 비용과 패킷 전달 비용을 계산하였다.

2. 관련연구

IP주소의 식별자와 위치자의 역할을 분리하기 위한

다양한 연구가 진행되어 왔으며, 이러한 연구는 크게 호스트 기반 프로토콜과 네트워크 기반 프로토콜로 구분되어 진다.

먼저 호스트 기반 프로토콜은 HIP, Shim6, MOFI(Mobile Oriented Future Internet) 등이 존재한다[1],[2],[9],[10]. 호스트 기반 프로토콜의 가장 대표적인 HIP는 단말의 식별자와 위치자를 분리하는 새로운 네임스페이스인 HI(Host Identity) 계층을 정의하고, IP계층과 전송계층 사이에 추가하였다[1]. 이 HI계층은 상위 계층과는 식별자로써 단말의 ID를 사용하고, 하위 계층은 위치자에 대한 IP주소를 사용하여 상호 통신할 수 있도록 고안되었다. 이를 위해서 HI계층은 식별자와 위치자 사이의 매핑 정보를 유지하는 기능을 갖는다. Shim6는 식별자와 위치자를 분리하는 기술로, HIP와 기본 개념은 동일하나 별도의 식별자나 위치자를 사용하지 않고, 기존의 IP주소를 사용한다. Shim6는 IP계층 내부에 Shim6 계층을 추가하여 식별자와 위치자의 매핑을 담당하도록 제안되었다[2].

MOFI도 두 가지의 이름공간을 사용하고 있으며, 단말의 고유한 식별을 위해 HID(Host ID)를 사용하고 위치정보를 위해 LOC를 사용한다. LOC는 지역 네트워크 내에서의 라우팅을 위한 A-LOC(Access network LOC)와 백본 네트워크에서 사용되기 위한 B-LOC(Backbone network LOC)의 두 가지 LOC가 사용된다. HID와 LOC의 매핑을 유지하고 관리하기 위해, 액세스 라우터(Access Router, AR)에는 Hash table을 포함하는 Local Mapping Controller(LMC)와 HID-LOC Register(HLR)가 포함되어 있다. HID와 LOC의 매핑 정보는 분산된 방식으로 AR들에서 관리되어진다[9],[10].

하지만, 이러한 호스트 기반 기술의 단점은 모든 단말의 프로토콜 스택을 수정해야한다는 점과 ID와 LOC의 매핑을 위해서 중앙 집중형의 랑데뷰 서버가 필요하다는 점이다. 이는 네트워크에 단말 장치가 전개되는 것을 어렵게 할 뿐만 아니라, 중앙 집중 방식의 매핑 서버 운영으로 단일 장애 지점 발생의 요인이 될 수 있다.

이에 반해서 네트워크 기반의 프로토콜은 단말의 프로토콜 스택을 수정할 필요가 없고, 모든 시그널링과 관련된 프로세스는 네트워크에서 처리되어지기 때문에 각 단말은 어떠한 시그널링 프로시저에도 관여할 필요가 없다. 대표적인 네트워크 기반의 프로토콜로는 LISP(Locator Identifier Separation Protocol) 프로토콜을 들 수 있다. LISP은 IP 주소를 두 개의 공간으로 구분하여 각 역할에 따라 구분하여 사용하고자 하였으며, 단말의 식별자를 위한 Endpoint Identifier(EID)와 단말의 위치자를 위한 Routing Locator(RLOC)를 새로운 이름공간으로 정의하였다[3]. EID는 라우터의 로컬 네트워크에서만 사용가능하고, 글로벌하게 라우팅될 수 없는 특징이 있는 반면, RLOC는 글로벌하게 라우팅이 가능한 IP주소로 표현될 수 있다. EID와 RLOC의 매핑과 Lookup은 각 라우터에서 진행되며, 패킷의 전달은 터널링 방식을 이용하여 출발지에서 목적지로 전달된다.

최근에는 또 다른 네트워크 기반의 프로토콜로, 분산 해시 테이블 기반의 식별자-위치자 매핑 기술(DHT-MAP)이 제안되었다. DHT-MAP 프로토콜의 네트워크는 패킷 포워딩을 위한 Customer network와 ID와 LOC 매핑정보의 유지관리를 위해 CAN(Content Addressable Network)을 이용한 overlay network로 구성되며, 모든 AS(Autonomous System)는 EID와 LOC의 매핑정보를 저장하기 위한 한 개 이상의 resolver를 유지하고 있다[4],[7].

네트워크 기반 프로토콜의 문제는 패킷을 포워딩하기 위해서는 항상 터널링을 사용해야한다는 점이다. 터널링의 사용으로 인해 패킷을 전달하고 처리하기 위해서는 패킷 인캡슐레이션과 디캡슐레이션 과정이 추가적으로 필요하게 되며, 이는 더 많은 대역폭과 패킷 처리를 위한 프로세싱 오버헤드가 필요하다.

DHT는 일종의 분산 시스템으로, 시스템 내의 각 노드가 특정한 키와 그에 해당하는 값을 <key, value>의 쌍으로 갖고 있으며, 시스템에 참여하는 노드는 주어진 키를 이용하여 중앙 서버 없이도 특정한 값을 검색할 수 있다. 특히 부하가 집중되지 않고 분산된다는 특징으로 인해 극단적으로 큰 규모의 노드들에 대한 관리도 가능하기 때문에 일반적으로 토렌트와 같은 P2P 시스템에서 사용된다.

최근 많은 관심을 받고 있는 SDN은 소프트웨어 정의 네트워킹의 약자로, 스위치와 같은 네트워크 장비의 제어 부분을 데이터 전송 부분으로부터 분리하고, 네트워크 장비의 기능을 정의할 수 있는 오픈된 API를 외부에 제공함으로써 프로그래밍 된 소프트웨어로 다양한 네트워크 경로의 설정과 직접적인 제어 등을 가능하게 하는 기술이다[6]. SDN에서는 물리적 네트워크와 가상 네트워크에 서로 다른 라우팅 기법을 운영하는 것이 가능하

기 때문에, 네트워크를 더 유연하게 만들 수 있다.

OpenFlow 규격은 SDN 아키텍처의 제어 평면과 전송 평면 사이에 정의된 최초의 표준 프로토콜이다[5],[8]. Open Networking Foundation(ONF)에서 SDN과 OpenFlow에 대한 규격을 정의하고 업데이트하고 있다. OpenFlow는 트래픽을 직접적으로 전송하고 관리할 수 있도록 설계되었기 때문에 SDN 컨트롤러가 네트워크 장치의 전송 평면과 직접적으로 상호작용할 수 있다. OpenFlow를 사용함으로써, 컨트롤러들이 원격으로 스위치의 플로우 테이블을 관리하는 기능을 제공할 수 있다.

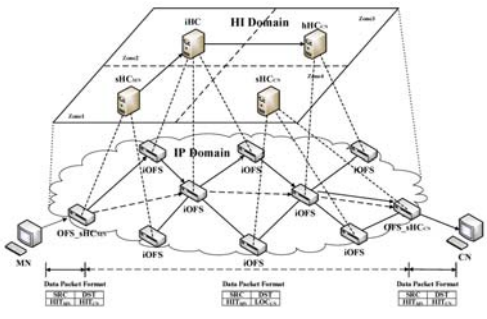
3. 네트워크기반 ID-LOC 분리 기술

3.1 새로운 namespace 정의: HIT와 LOC

본 연구의 제안 기술에서는 단말의 식별자와 위치자의 역할을 위한 두 개의 이름공간을 사용하였으며, 각각 HIT와 LOC로 나타낸다. HIT는 IP기반 네트워크에서는 라우팅되지 않는 IPv6 주소로 HIP의 HIT 생성 메커니즘을 사용하여 생성되는 주소이다[1],[11]. 단, 본 연구에서 사용되는 HIT는 기존의 HIT생성 방식과는 다르게 공개키를 사용하지 않고, 단말의 Network Access Identifier(NAI)를 이용하여 생성한다.

LOC는 기존 IP기반의 네트워크에서 라우팅이 가능한 IPv6형태의 주소로 생성되며, 기존의 IPv6주소 생성 메커니즘의 방식에 따라 현재의 액세스 네트워크의 프리픽스(prefix)와 단말의 Media Access Control(MAC)주소를 이용하여 생성된다[12],[13]. LOC는 IP망에서 일반적인 IP 주소로 사용되어지며, 단말이 현재 붙어 있는 위치를 나타낸다.

3.2 시스템 구성



[그림 1] 시스템 구성도

제안 기술의 네트워크는 HI 도메인과 IP 도메인으로 구분되어지고, 네트워크 자원은 HI 도메인과 IP 도메인을 모두 수용하도록 가상화되어진다.

네트워크를 구성하는 각 구성요소들은 SDN과 마찬가지로 OpenFlow 스위치(OFSs)와 컨트롤러들로 구성된다. 위의 [그림 1]은 제안한 네트워크의 구조를 보여주고 있으며, IP 도메인에 오버레이 된 HIT 도메인의 모습을 함께 묘사하고 있다. [그림 1]에서 HI 도메인은 네 개의 HIT 존으로 구분되고, 각 존은 HC에 의해 관리된다. HI 도메인은 IP 도메인 위에 오버레이 되어 구성되며, 패킷들은 실제로는 IP 도메인 상에서 전달된다. 그렇기 때문에 IP 도메인 위에 오버레이 되어진 HIT 도메인의 물리적인 경로는 최적화되지 않을 수도 있다.

IP 도메인은 일반적인 IP 라우팅을 위한 도메인으로 일반적인 IPv6 주소를 갖는 패킷들은 SDN 환경에서와 마찬가지로 IP 라우팅을 통해 전송될 수 있다. 그에 반해, HI 도메인은 HIT 패킷을 전달하기 위해 사용되며, HIT 패킷들은 HI 도메인의 논리적인 최단 경로를 통해 이동될 수 있다. HI 도메인은 DHT기반의 CAN 네트워크와 마찬가지로 가상의 d-차원의 카르테시안 좌표공간으로 구성되고, IP 네트워크 위에 논리적으로 오버레이 되어지며, 물리적인 좌표 시스템과 어떠한 연관도 존재하지 않는다.

HI 도메인의 가상 좌표 공간은 <HIT, LOC> 쌍을 유지 및 관리하고 저장하기 위한 공간으로 사용된다. 전체의 가상 좌표 공간은 시스템의 컨트롤러의 수에 따라 동적으로 나누어지며, 모든 HIT 주소들은 존(Zone)의 수와 동일한 수의 HIT 블록에 나누어 할당되며, HIT 블록은 각 존에 할당된다. HIT 관리 모듈을 갖는 컨트롤러는 각 존에 위치하고 있으며, HC 컨트롤러라 한다. 각각의 HC는 HIT를 갖는 단말들과 할당된 HIT 블록을 관리해하고, HIT-LOC 매핑 정보를 등록해야할 책임이 있다. HI 도메인의 모든 HC들은 좌표 공간 내에 구별되는 개별적인 존을 갖고 있으며, DHT메커니즘을 통해 <HIT, LOC>쌍을 관리한다. HC의 기본적인 기능은 다음과 같다.

1) 단말의 네트워크 접근 탐지
2) 단말의 HIT와 LOC 매핑 및 HIT-LOC 매핑 정보 업데이트
3) 매핑정보의 유지 및 관리
4) HIT 라우팅을 위한 스위치 flow table 설정

시스템 내의 모든 HC들은 자신의 OFS들의 flow table을 미리 설정해야 한다. 이러한 사전 설정을 통해 HC들이 HIT 패킷의 목적지까지 도달하기 위한 가장 인근의 존에 대한 경로를 획득할 수 있도록 하고, OFS들이 해당되는 존으로 HIT 패킷을 전송할 수 있도록 한다.

3.3 HC와 OFS의 동작

HC들의 역할은 크게 세 가지로 구분된다. 첫 번째 역할은 단말의 home HC로써의 동작으로, 단말의 HIT가 어떤 HC의 HIT 블록에 포함될 경우, 해당 HC는 그 단말의 홈 컨트롤러(hHC)가 된다. 이 경우 hHC는 해당 단말의 HIT와 LOC의 매핑 정보를 관리해야하고, HIT 쿼리에 대해 적절한 LOC를 전달해야할 책임이 주어진다.

두 번째 역할은 단말이 현재 위치한 곳을 관리하는 영역의 컨트롤러로, 이때의 컨트롤러는 해당 단말에 서비스를 제공해 주는 serving HC(sHC)가 된다. sHC는 새로운 단말의 접근을 탐지하고, 해당 단말의 HIT에 대한 LOC를 할당하고 매핑 정보를 유지한다. 또한 sHC는 해당 단말의 HIT-LOC 바인딩 정보를 단말의 hHC에게 알려주거나 업데이트한다. 또한 sHC는 단말이 현재 붙어있는 네트워크의 컨트롤러이기 때문에, 소스 주소로 HIT를 사용하는 단말(MN)이 목적지의 주소로 HIT를 사용하는 단말(CN)로 패킷을 보내고자 하는 경우 sHC는 CN의 LOC를 얻기 위한 쿼리를 전송한다. 일단 sHC가 MN과 CN의 LOC 정보를 획득하기만 하면, 획득한 정보를 이용하여 자신의 OFS들의 flow table을 설정한다.

HC의 마지막 역할은 중간 전달자의 역할로, 단말의 hHC와 sHC의 중간에 위치하는 모든 HC가 intermediate HC(iHC)가 될 수 있다. iHC는 HIT 라우팅에 대해 사전 설정된 flow table에 따라 목적지를 향하는 가장 가까운 지역으로 패킷을 전달하는 역할을 한다.

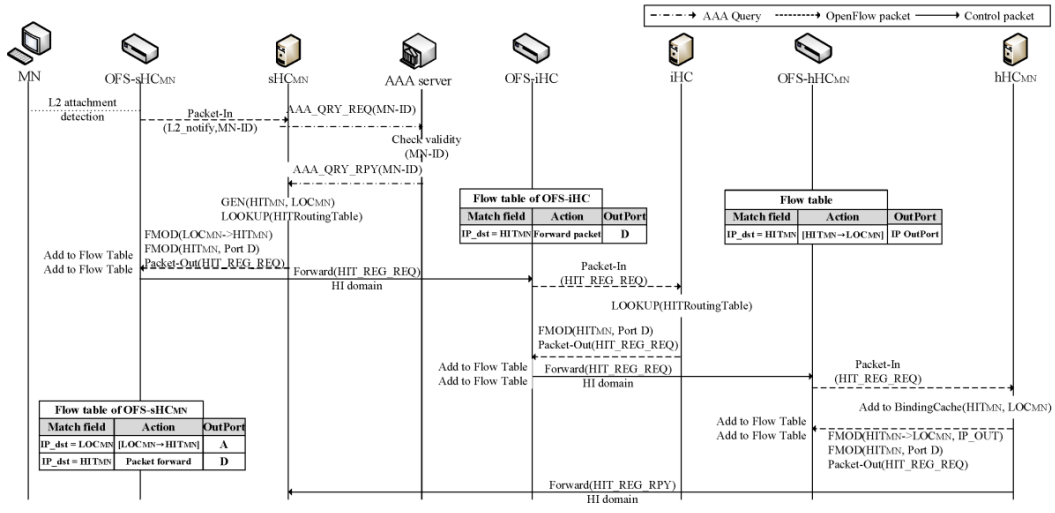
HC들과 달리, OFS들은 보통 자신의 flow table의 매칭 룰에 따라 수신한 패킷을 처리한다. 수신 패킷에 대한 룰이 자신의 flow table에 없을 경우, OFS는 자신이 속한 HC로 시큐어 채널을 통해 해당 패킷을 전달하고, 룰이 존재하는 경우는 룰에 따라 적절한 처리를 수행한다.

OFS의 주소 필드 변환 기능은 최적의 경로를 찾아 패킷을 전송하기 위한 기능이다. 단말(MN)이 연결된 OFS는 IP 도메인으로부터 수신한 패킷의 목적지 주소가 MN의 LOC인 경우, 해당 목적지 주소를 HIT로 변환하여 패킷을 전달할 수 있다.

MN과 CN이 연결된 OFS사이에 위치한 OFS들은 중간 OFS(iOFS)로 간주되고, iOFS의 기본적인 역할은 iHC와 마찬가지로 목적지를 향하는 가장 가까운 존의 OFS로 수신한 패킷을 전달하는 것이다.

3.4 등록 과정

[그림 2]는 새로운 노드가 네트워크에 추가될 때 발생하는 등록 절차를 나타내고 있다. 새로운 단말이 네트워크에 연결되면, hHC와 sHC에 단말의 상태 정보를 등록



[그림 2] 새로운 단말의 등록 절차

하고 업데이트해야 한다. 새로이 추가되는 단말을 MN이라 하면, 이 MN이 처음 연결을 시도하는 OFS는 먼저 자신의 flow table을 검색한다. 이 OFS는 MN에 대한 어떠한 정보도 갖고 있지 않기 때문에 자신의 HC에게 새로운 노드가 붙었다는 알림 메시지를 L2 attachment로부터 획득한 MN의 ID 정보를 포함하여 전달한다. 이 때, 이 HC는 MN의 sHC의 역할을 수행한다. sHC는 획득한 MN의 ID와 함께 AAA 서버로 MN에게 HIT 서비스를 제공해야 하는지 여부를 확인하기 위한 쿼리 요청을 전달한다. AAA 서버로부터 해당 ID의 유효성이 검증되면, sHC는 MN의 HIT와 LOC 주소를 생성한다. sHC는 먼저 MN의 HIT(HIT_{MN})에 대한 next hop port를 찾기 위해 자신의 HITRoutingTable을 검색하고, MN의 hHC에게 등록 요청 쿼리를 전송한다. 그와 함께 자신이 관리하는 OFS에게 flow table add 명령을 보내어 해당 MN의 HIT와 LOC 매핑 정보를 등록한다. 등록된 flow table entry는 LOC를 사용하여 내부로 유입되는 패킷의 주소 필드 변환에 사용된다. 이에 대한 과정은 [그림 2]에 표현되고 있다.

[표 1] 메시지 형식

메시지 형식	의미
pkt	일반적인 IP 패킷 또는 관리용 패킷
pkt.type	패킷 타입
MN-ID	L2 attachment notify 메시지로 획득한 단말의 ID. (NAI 또는 MAC)
pkt.dst	패킷의 목적지 주소
MY_HIT_ADDRESS	HC의 HIT 주소
MY_HIT_ADDR_BLOCK	HC가 관리하는 HIT 주소 블럭
BindingUpdateList	바인딩 리스트 sHC에서 관리/유지
BindingCache	하나이상의 단말에 대한 다수의 바인딩을 포함 hHC에서 관리/유지되는 캐시
HITRoutingTable	모든 HC에서 관리/유지되는 라우팅 정보가 저장된 라우팅 테이블
L2_ATT_NOTIFY	링크계층에서 단말의 연결에 대한 알림
AAA_QRY_REQ	HC가 AAA서버로 보내는 쿼리 요청 메시지
AAA_QRY_RPY	쿼리 요청에 대한 응답으로 AAA 서버에서 보내는 응답 메시지
HIT_REG_REQ	sHC가 자신의 파라미터 셋에 대한 등록을 요청하기위해 보내는 쿼리 메시지
HIT_REG_RPY	HIT_REG_REQ에 대한 응답 메시지
LOOKUP	알맞은 next hop 포트를 찾기 위한 쿼리
FMOD	Flow table을 수정하기 위한 OpenFlow 제어 메시지

모든 HC들은 HC 처리 알고리즘(HC Processing Algorithm, HPA)에 의해 운영되어진다. HPA 알고리즘은 HC의 역할에 따라 세 단계의 프로세스로 구성된다. [그림 3]은 전체 HPA 알고리즘을 나타내고 있으며, [표 1]은 제안한 알고리즘과 제안 기술에서 사용되는 메시지의 형식을 보여주고 있다.

Algorithm 1 HC Processing Algorithm

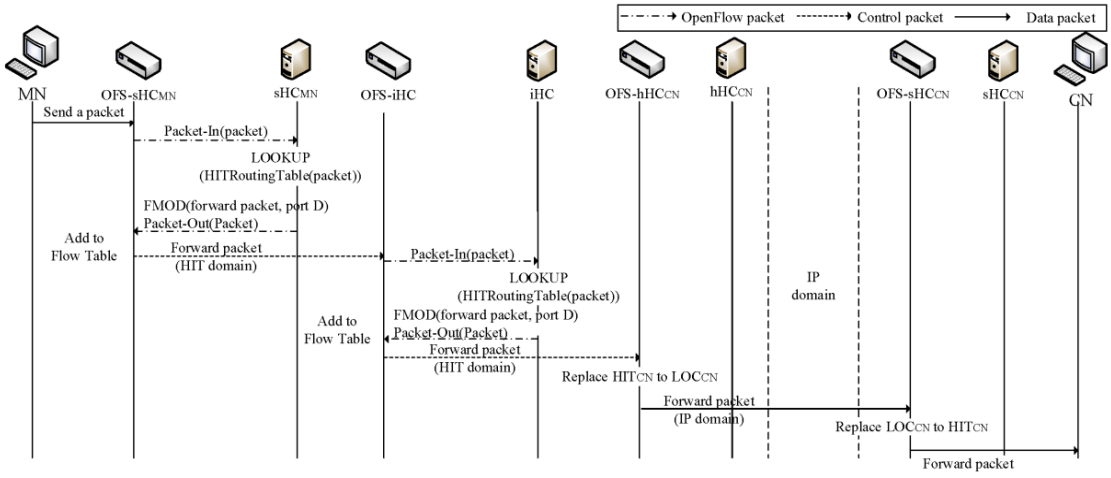
```
1:  receive a packet pkt from the OFS
2:  if pkt.dst is MY_HIT_ADDRESS(HIT_sHC or LOC_sHC) then
3:      if pkt.type is L2_ATTACHMENT_NOTIFY then
4:          save MN-ID from pkt
5:          send AAA_QRY_REQ(MN-ID)
6:      else if pkt.type is AAA_QRY_RPY(MN-ID) then
7:          generate HIT_MN using NAI_MN
8:          generate LOC_MN using MAC_MN
9:          LOOKUP HITRoutingTable for the next hop port(D)
10:         send HIT_REG_REQ(HIT_MN,LOC_MN) to
            hHC(HIT_MN)
11:         add (HIT_MN, LOC_MN) to BindingUpdateList
12:         add a flow table entry (LOC_MN → HIT_MN)
13:     else if pkt.type is HIT_REG_RPY then
14:         done
15:     end if
16: else if pkt.dst is in MY_HIT_ADDR_BLOCK then
17:     if pkt.type is HIT_REG_REQ then
18:         add (pkt.dst,pkt.loc) to BindingCache
19:         add a flow table entry(pkt.dst → pkt.loc)
20:         send HIT_REG_RPY to pkt.src
21:     end if
22: else if pkt.dst is not in MY_HIT_ADDR_BLOCK then
23:     LOOKUP HITRoutingTable(pkt.dst) for next hop port(D)
24:     add a flow table entry (HIT_MN, D)
25:     Packet-Out(pkt)
26: else
27:     drop packet
28: end if
```

[그림 3] HC 처리 알고리즘(HPA)

3.5 패킷 전달 과정

MN이 sHC와 자신의 hHC에 등록이 되면, MN은 목적지 단말에게 데이터를 요청할 수 있다.

이 때 목적지 단말을 CN이라고 하면, MN은 자신의 HIT_{MN}과 CN의 HIT 주소(HIT_{CN})를 사용하여 패킷을 전달할 수도 있다. MN이 연결된 OFS는 CN의 HIT 주소에 대한 flow table entry를 검색하고, 존재하지 않을 경우 해당 패킷을 자신의 HC로 전달한다. sHC (sHC_{MN})는 MN의 등록절차와 유사한 방법으로 HIT_{CN}에 대한 HITRoutingTable 검색을 진행하고 적절한 next hop port를 찾아서 HI 도메인을 통해 목적지와 가장 인접한 HC로 패킷을 전달한다. 동시에 hHC는 [그림 3]의 알고리즘에 따라 HIT_{CN}에 대한 flow table entry를 추가하는 명령을 자신의 OFS에게 전달한다. 해당 패킷은 HC와 iOFS를 경유하여 CN의 hHC



[그림 4] 패킷 포워딩 절차

(hHC_{CN})의 OFS로 전달된다. hHC_{CN}의 OFS는 이미 CN의 HIT에 대한 flow table entry를 갖고 있기 때문에, flow table entry에 적용된 룰에 따라 주소 필드의 값인 HIT를 LOC로 변환하여 CN의 IP 도메인을 통해 현재 CN이 연결되어 있는 sHC(sHC_{CN})의 OFS까지 직접 패킷을 전송할 수 있다. sHC_{CN}의 OFS가 이 패킷을 수신하게 되면, 마찬가지로 자신의 flow table에 의해 CN의 LOC(LOC_{CN})는 원래의 주소값인 CN의 HIT (HIT_{CN})로 변환되어 최종 목적지에 도달한다. 첫 번째 패킷이 목적지에 도달하게 되면 CN과 MN의 통신 경로가 설정되고, 이후의 패킷들은 설정된 경로를 따라 전달될 수 있다. [그림 4]는 이러한 패킷의 전달 과정을 흐름에 따라 나타내고 있다.

3.6 경로 최적화 과정

설정된 경로를 따라 이후의 패킷들이 전달될 때, 패킷들은 항상 hHC_{CN}의 OFS와 중간의 몇몇 iOFS를 경유하는 경로를 통해서만 전달될 수밖에 없다. IP 도메인을 경유하는 경로는 IP 라우팅을 사용함으로써 최적화되어질 수 있지만, 그 이외의 MN에서 CN까지의 경로는 최적화되어 있지 않을 수도 있다. MN과 CN이 각각 연결되어 있는 OFS가 IP 도메인을 경유하여 패킷을 포워딩시킬 수 있는 적절한 flow table entry를 갖고 있다면, MN에서 CN까지의 전체 경로는 최적화될 수 있다. 이를 위해서, 본 연구에서는 추가적인 경로 최적화(Route Optimization, RO)를 위한 알고리즘을 제안하였다. RO를 위한 알고리즘은 [그림 5]를 통해 확인할 수 있고, 이 알고리즘은 앞서 HPA 알고리즘의 hHC와 sHC에 해당하는 알고리즘의 말미에 추가적으로 삽입되어야 한다.

Algorithm 2 RO Algorithm for sHC and hHC

```
Require: the HC acts as sHC
 1: if pkt.src is in BindingUpdateList then
 2:     send HIT_QRY_REQ to pkt.dst
 3: else if pkt.dst is MY_HIT_ADDRESS(HIT_sHC) then
 4:     if pkt.type is HIT_QRY_RPY then
 5:         add a flow table entry (pkt.dst → pkt.loc)
 6:     else
 7:         drop packet
 8:     end if
 9: else
10:     follow line 16 of Algorithm 1
11: end if
Require: the HC acts as hHC
12: if pkt.dst is in MY_HIT_ADDR_BLOCK then
13:     if pkt.type is HIT_QRY_REQ then
14:         lookup BindingCache(pkt.dst)
15:         send HIT_QRY_RPY with (LOC_CN) to pkt.src
16:     else
17:         drop packet
18:     end if
19: else
20:     follow line 22 of Algorithm 1
21: end if
```

[그림 5] 경로 최적화 알고리즘

앞서의 패킷 포워딩 과정 중 sHC_{MN}이 HIT_{CN}의 next hop port를 찾고 패킷을 전달하면서 flow table entry를 생성하는 단계에서, sHC_{MN}은 hHC_{CN}에게 CN의 LOC(LOC_{CN})를 요청하는 쿼리를 전송하게 되며, 출발지의 주소는 sHC_{MN} 자신의 HIT 주소로 설정

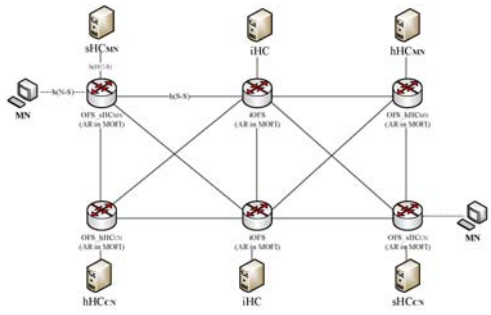
한다. hHC_{CN}이 해당 쿼리 요청을 수신한 후에, 이 sHC_{MN}에게 LOC_{CN}의 정보를 포함하는 응답 메시지를 전달하게 되고, 이 응답 메시지를 수신한 sHC_{MN}은 응답 메시지로부터 LOC_{CN}을 획득하여 <HIT_{CN}, LOC_{CN}>의 매핑 정보를 자신의 BindingUpdateList에 저장한다.

sHC_{MN}은 자신이 관리하는 OFS들에게 <HIT_{CN}, LOC_{CN}>에 대한 flow table entry를 수정 또는 추가하는 명령을 전달하여, 목적지 주소가 HIT_{CN}인 패킷에 대해 목적지 주소를 LOC_{CN}으로 변경하고 IP 도메인으로 나가는 포트로 직접 내보낼 수 있게 함으로써 경로 최적화 과정이 이루어진다.

[그림 6]은 RO 과정에 대한 패킷 흐름을 보여주고 있으며, RO가 설정된 이후에 MN이 연결된 OFS로부터 IP 도메인을 통해 전달되는 패킷이 목적지로 도달되는 과정을 보여주고 있다.

4. 비용 분석

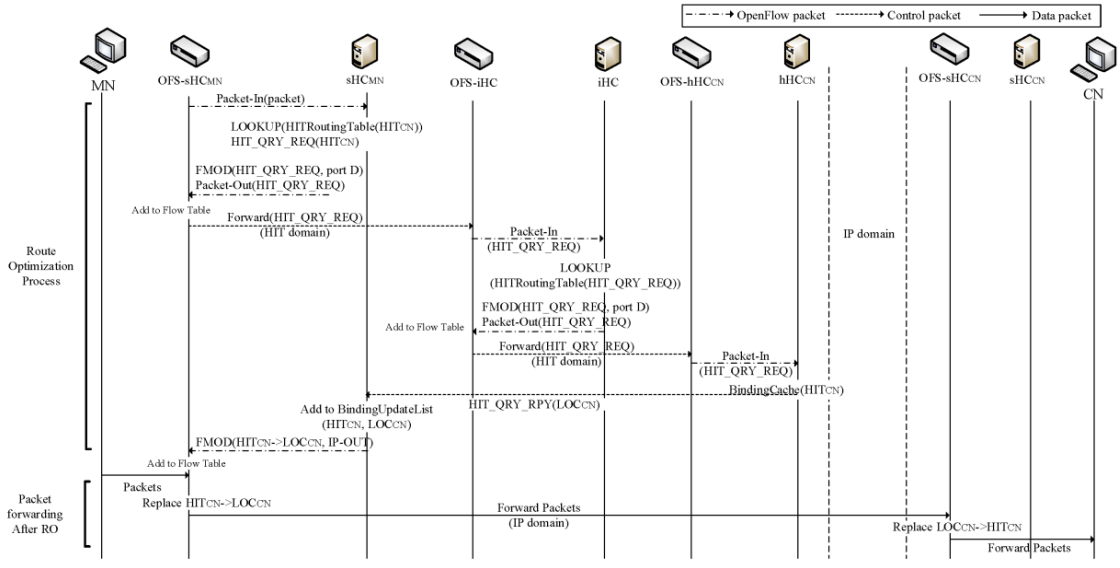
앞서 제안한 기술의 패킷 전달 및 시그널링 비용을 계산하기 위해서 본 연구와 유사한 방식의 MOFI를 비교 대상으로 정하였으며, 참고문헌 [14]를 기반으로 비용 분석에 대한 모델을 선정하고, 아래의 [그림 7]과 같이 네트워크를 구성하였다.



[그림 7] 비교분석을 위한 네트워크 구성

전체 네트워크는 6개의 HC와 6개의 OFS, 2개의 최종 단말로 구성하였다. 각 OFS는 서로 연결되어 있고, 각각의 OFS는 모바일 장치가 연결할 수 있는 액세스 포인트를 포함하고 있다.

MOFI와 본 논문의 제안 기술에서의 사용하는 네트워크 구성요소는 서로 다르기 때문에, 프로토콜에 따라 네트워크의 일부 구성요소의 기능은 대체되었다고 가정한다. MOFI에서 MN이 CN과 통신할 때, 각 AR에는 Hash table과 HLR을 갖는 LMC가 포함되어 있고, HID와 LOC의 매핑 정보는 분산된 방식으로 AR들에 의해 관리되어진다.



[그림 6] 경로 최적화(RO) 절차

그에 반해 제안한 방식에서 각 AR은 OpenFlow 스위치의 모든 기능을 포함할 수 있다. [그림 7]에서 $h_{(X-Y)}$는 네트워크를 구성하는 엔티티인 X와 Y의 평균 홉 수를 나타낸다. 시그널링 비용(C_S)과 패킷 전송 비용(C_{PD})은 바이트 단위의 메시지 크기로 표현될 수 있으며 전송 경로를 경유하는 홉 카운트의 경로 길이의 곱으로 계산하고, 전체 비용은 시그널링과 패킷 전송 비용의 합으로 계산한다.

4.1 시그널링 비용(Signaling Cost)

4.1.1 MOFI의 시그널링 비용 계산

MOFI에서 HID와 LOC의 바인딩과 LOC 쿼리 프로시저는 패킷을 전달하기 전에 선행되어야 한다. 따라서, MOFI의 시그널링 비용(C_S^M)은 다음과 같이 나타낼 수 있다.

$$C_S^M = C_{BI}^M + C_{LQ}^M \quad (1)$$

위의 (1)에서 C_{BI}^M와 C_{LQ}^M는 각각 HID-LOC 바인딩과 LOC 쿼리에 따르는 비용을 나타낸다. 링크 계층의 L2 attachment가 진행된 후에, MN은 연결된 AR (LMC_{AR-MN})로 Binding Request 메시지(BR)를 보내고, 그 후 LMC_{AR-MN}은 MN의 LMC(LMC_{MN})에게 BR을 전달한다. BR을 처리한 후에 LMC_{MN}은 Binding Acknowledgement(BA) 메시지를 MN에게 보낸다. MN의 HIT-LOC 바인딩 비용과 CN의 비용이 같다고 가정하면, 바인딩 비용은 다음과 같이 계산된다.

$$C_{BI}^M = 2(L_{BR} + L_{BA})(h_{(N-S)} + h_{(S-S)}) \quad (2)$$

L_{BR}과 L_{BA}는 BR과 BA의 길이를 의미한다.

LMC_{AR-MN}이 MN에서 CN으로 보내는 첫 번째 데이터 패킷을 수신하게 되면, CN의 적합한 LOC를 구하기 위한 LOC 쿼리 프로시저를 진행하게 되고, LOC 쿼리 요청(LQR) 메시지는 LMC_{CN}을 거쳐 LMC_{AR-CN}로 전달된다. LOC 쿼리 응답 메시지(LQP)의 경우, LMC_{AR-CN}에서 LMC_{AR-MN}로 전달되기 때문에 LOC 쿼리에 대한 비용은 다음과 같이 계산된다.

$$C_{LQ}^M = 2h_{(S-S)}L_{LQR} + h_{(S-S)}L_{LQA} \quad (3)$$

4.1.2 제안한 기술의 시그널링 비용 계산

MN과 CN은 우선 각각의 hHC에 등록이 되어야 한다. L2 attachment 후에 sHC_{MN}은 Registration Request (RR) 메시지를 hHC_{MN}에게 보내게 되고, 경로 최적화를 위해 쿼리 요청도 함께 보낼 수 있다. 그리고 MN과 CN이 자신의 hHC에 등록을 위한 평균적인 비용은 같다고 가정한다. 따라서 제안한 기술의 시그널링 비용(C_S^P)은 아래와 같다.

$$C_S^P = 2(C_{RRQ}^P + C_{RO}^P) \quad (4)$$

C_{RRQ}^P와 C_{RO}^P는 등록과 RO에 대한 비용을 나타내고, C_{RRQ}^P는 다시 등록 요청(RR)과 그 응답(RP)에 대한 비용의 합으로 계산된다. 따라서 다음과 같이 계산식을 생성할 수 있다.

$$C_{RRQ}^P = C_{RR}^P + C_{RP}^P \quad (5)$$

C_{RR}^P은 RR 메시지를 전달하는 비용과 OpenFlow signaling에 대한 비용으로 표현할 수 있기 때문에 다음과 같이 계산된다.

$$C_{RR}^P = C_{DRR}^P + C_{ORR}^P \quad (6)$$

RR 메시지는 sHC_{MN}에서 hHC_{MN}로 전달되기 때문에, C_{DRR}^P은 다음과 같이 표현된다.

$$C_{DRR}^P = h_{(S-S)}L_{RR} \quad (7)$$

식에서 L_{RR}은 RR 메시지의 길이를 나타낸다.

C_{ORR}^P은 HC의 역할에 따라 분류된다. 각 역할에 따라 RR 메시지를 처리하는 HC의 OpenFlow 시그널링에 대한 비용을 각각 C_{RR}^{sHC}, C_{RR}^{hHC}, C_{RR}^{iHC}로 구별하였으며, 전체 비용 C_{ORR}^P은 C_{RR}^{sHC}, C_{RR}^{hHC}와 C_{RR}^{iHC}의 합

과 같고, 각 컨트롤러에서 처리하는 비용은 아래와 같이 계산될 수 있다.

$$C_{ORR}^{P} = C_{RR}^{sHC} + C_{RR}^{hHC} + C_{RR}^{iHC} ,$$
$$C_{RR}^{sHC} = h_{(HC-S)}(2L_{FMOD} + L_{RPOUT}) \quad (8)$$
$$C_{RR}^{iHC} = (h_{(S-S)} - 1) \times$$
$$(L_{RPIN} + 2L_{FMOD} + L_{RPOUT}) h_{(HC-S)} \quad (9)$$
$$C_{RR}^{hHC} = h_{(HC-S)}(2L_{FMOD} + L_{RPIN}) \quad (10)$$

수식 (8)에서 L_{FMOD}는 FMOD 컨트롤 메시지의 길이를 나타내고, L_{RPOUT}은 RR 메시지를 포함하는 Packet-out(POUT) 메시지의 길이를 나타내며, 반대로 식(9)에서 L_{RPIN}은 RR 메시지를 포함하는 Packet In(PIN) 메시지의 길이를 나타낸다.

이와 유사한 과정을 거쳐 RP 메시지에 대한 비용도 마찬가지로 계산될 수 있으며, 이는 아래와 같이 표현된다.

$$C_{RP}^{P} = C_{DRP}^{P} + C_{ORP}^{P} \quad 11)$$

여기서 RP의 전달에 대한 비용과 OpenFlow를 이용하는 시그널링의 비용은 아래와 같다.

$$C_{DRP}^{P} = h_{(S-S)} L_{RP} ,$$
$$C_{ORP}^{P} = h_{(HS-S)}(L_{RPPIN} + L_{RPPOUT}) \quad (12)$$

경로 최적화에 대한 비용(C_{RO}^{P})도 마찬가지로 쿼리 요청(ROQ)과 응답(ROR)의 비용에 대한 합으로 나타낼 수 있다.

$$C_{RO}^{P} = C_{ROQ}^{P} + C_{ROR}^{P} \quad (13)$$

ROQ 비용은 다시 ROQ 전달에 필요한 비용(ROQD)과 OpenFlow에 대한 비용(ROQO)로 구분할 수 있으며, C_{ROQ}^{P}는 다음과 같다.

$$C_{ROQ}^{P} = C_{ROQD}^{P} + C_{ROQO}^{P} ,$$
$$C_{ROQD}^{P} = h_{(S-S)} L_{ROQ} ,$$
$$C_{ROQO}^{P} = h_{(HC-S)}(L_{RQPIN} + L_{RQPOUT}) \quad (14)$$

위의 식에서 L_{RQPIN}과 L_{RQPOUT}은 ROQ를 포함하는 PIN과 POUT 메시지의 길이를 나타낸다.

C_{ROR}^{P}은 ROR 전달 비용(C_{RORD}^{P})과 OpenFlow 시그널링 비용 C_{RORO}^{P}의 합으로 계산되고, C_{RORD}^{P}는 $h_{(S-S)} L_{ROR}$ 와 같이 계산할 수 있다.

$$C_{ROR}^{P} = h_{(S-S)} L_{ROR} + C_{RORO}^{P} \quad (15)$$

각각의 HC는 ROR 메시지를 전달하기 위해서 flow table entry를 설정해야 한다. 따라서 ROR 메시지는 hHC_{CN}에서 sHC_{MN}으로 전달될 수도 있다. ROR이 sHC_{MN}에 도착하고 나면, sHC_{MN}은 HIT를 LOC로 변경하는 flow table entry를 생성하게 된다. 유사한 방법으로 OpenFlow signaling에 대한 비용(C_{RORO}^{P})은 다음과 같이 계산된다.

$$C_{RORO}^{P} = C_{ROR}^{sHC} + C_{ROR}^{hHC} + C_{ROR}^{iHC} \quad (16)$$

각 HC에 대한 비용 계산은 아래의 식으로 통해 확인할 수 있다.

$$C_{ROR}^{sHC} = h_{(HC-S)} L_{FMOD} + L_{ROPOUT} ,$$
$$C_{ROR}^{sHC} = h_{(HC-S)}(h_{(S-S)} - 1) \times$$
$$(L_{ROPIN} + L_{FMOD} + L_{ROPOUT}) ,$$
$$C_{ROR}^{hHC} = h_{(HC-S)} L_{ROPIN} + L_{FMOD} \quad (17)$$

4.2 패킷 전달 비용(Packet Delivery Cost)

패킷 전달 비용은 패킷 수와 길이로 계산되지만, MOFI에서의 패킷 전달 방식과 제안한 기술의 방식이 서로 다르기 때문에 비용 계산식 또한 달라진다.

4.2.1 MOFI의 패킷 전달 비용

패킷의 전달을 위해서 MOFI는 터널링을 사용하기 때문에, LOC정보를 포함하는 추가적인 헤더가 필요하다.

헤더의 크기는 동일하기 때문에, MOFI의 패킷 전달 비용(C_{PD}^{M})은 다음과 같이 계산된다.

$$C_{PD}^{M} = N(p)(L_{DP}+L_{LH})(2h_{(N-S)}+h_{(S-S)}) \quad (18)$$

위의 식에서 $N(p)$와 L_{DP}는 데이터 패킷의 수와 데이터 패킷의 길이를 나타내고, L_{LH}는 LOC 정보의 헤더 길이를 나타낸다.

4.2.2 제안 기술의 패킷 전달 비용

패킷 전달 비용은 데이터 패킷의 전달(FDP) 비용과 OpenFlow를 사용하여 데이터를 전달하는데 소모되는 (ODP) 비용의 합으로 계산한다. RO 전까지는 패킷이 최적화되지 않은 경로를 통해 전달되지만, RO가 완료된 후에는 모든 패킷들은 최적의 경로를 통해 이동되어야 한다. 그에 따라 패킷 전달 비용(C_{PD}^{P})는 다음과 같이 표현되어 진다.

$$C_{PD}^{P} = C_{FDP}^{P} + C_{ODP}^{P} \quad (19)$$

여기서 C_{FDP}^{P}는 다음과 같이 계산된다.

$$C_{FDP}^{P} = (r_{no}N(p)L_{DP}(2h_{(N-S)}2h_{(S-S)})) + ((1-r_{no})N(p)L_{DP}(2h_{(N-S)}+h_{(S-S)})) \quad (20)$$

식에서 r_{no}는 최적화되지 않은 경로를 사용하여 패킷이 전달될 비율을 나타낸다. ODP에 대한 비용은 sHC_{MN}과 iHC들에서만 발생되기 때문에 ODP의 전체 비용은 sHC_{MN}과 iHC의 비용의 합과 같고, 다음과 같이 표현한다.

$$C_{ODP}^{P} = C_{sODP}^{P} + C_{iODP}^{P} \quad (21)$$

첫 패킷이 목적지에 도달되더라도, 응답 메시지가 전송되기 전까지는 역경로에 대한 flow table entry는 생성될 수 없다. 따라서 C_{sODP}^{P}와 C_{iODP}^{P}는 다음과 같이 계산될 수 있다.

$$C_{sODP}^{P} = h_{(HC-S)}(L_{fPIN}+L_{FMOD}+L_{fPOUT}) \quad (22)$$

$$C_{iODP}^{P} = h_{(HC-S)}(h_{(S-S)}-1)\times (L_{fPIN}+L_{FMOD}+L_{fPOUT}) \quad (23)$$

(23)의 식에서 L_{fPIN}과 L_{fPOUT}은 각각 첫 데이터 패킷을 포함하는 PIN과 POUT 메시지의 길이를 나타낸다.

5. 분석 결과

MOFI와 제안 기술의 비교를 위해 시그널링 비용과 패킷 전달 비용을 평가하였다. [표 2]는 평가에서 사용된 패킷의 길이와 시스템 파라미터를 나타내고 있다.

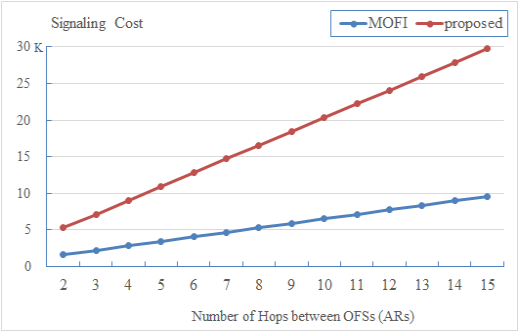
[표 2] 평가에 사용된 패킷 크기와 파라미터

notation	size (byte)	notation	size (byte)
L_{BR}	88	L_{BA}	88
L_{LQR}	88	L_{LQA}	88
L_{RR}	88	L_{FMOD}	116
L_{RPOUT}	246	L_{RPIN}	254
L_{RP}	88	L_{ROQ}	88
L_{RPPOUT}	246	L_{RPPIN}	254
L_{ROPOUT}	246	L_{ROPIN}	254
L_{RQPOUT}	246	L_{RQPIN}	254
L_{ROPOUT}	246	L_{ROPIN}	254
L_{ROR}	88	L_{DP}	128
L_{fPOUT}	334	L_{fPIN}	342
L_{LH}	40	r_{no}	0.1

5.1 시그널링 비용 비교

[그림 8]은 OFS 사이의 홉 수를 변화함에 따라 변화되는 시그널링 비용을 보여주고 있다. 그림에서 보듯 제안 기술의 시그널링 비용은 항상 MOFI의 비용보다 크게 나타나고 있다. 이러한 이유는 HI 도메인에서 컨트롤 메시지를 라우팅하기 위한 OpenFlow 메시지들은 HC들과

OFS들 사이에서 교환되기 때문에 MOFI 보다 더 많은 시그널링이 발생하기 때문이다. MOFI에서는 LMC가 AR과 같은 위치에 존재하기 때문에 LMC와 AR간의 통신 비용은 발생하지 않는다.

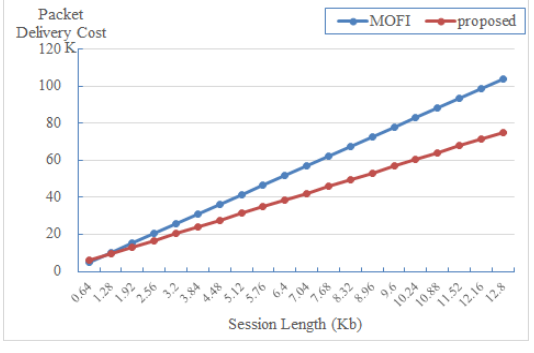


[그림 8] OFS간 홉 수에 따른 시그널링 비용

5.2 패킷 전송 비용 비교

패킷 전달 비용의 측정을 위해 OFS(또는 AR)간의 홉 수는 5로 설정하였다. 세션 길이는 패킷 수와 패킷 길이의 곱으로 계산하였고, 그에 따라 0.64KB에서 12.8KB까지 변화하였다.

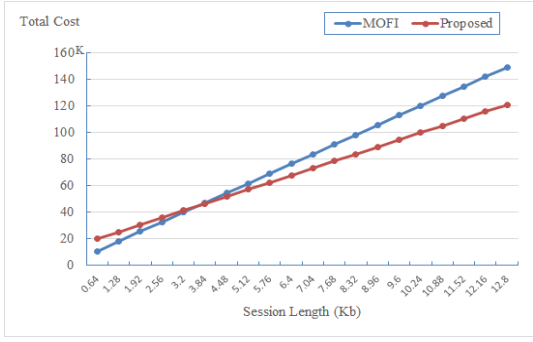
[그림 9]는 세션 길이의 변화에 따른 패킷 전송 비용의 변화를 나타내고 있으며, 시그널링 비용의 그래프와는 반대로 제안 기술의 패킷 전달 비용이 MOFI 보다 낮다는 것을 볼 수 있다. 제안 기술은 필드의 교체만을 진행하는 반면, MOFI는 터널링으로 인해 세션의 길이가 길어지면 질수록 두 기술 사이의 패킷 전송 비용의 격차는 커지게 된다.



[그림 9] 세션 길이에 따른 패킷전송 비용

5.3 전체 비용 비교

전체 비용은 시그널링의 비용과 패킷 전송 비용의 합으로 나타낼 수 있다. [그림 10]에서 보는 바와 같이, 세션 길이가 3.84KB 이하의 경우는 MOFI가 우수하지만, 세션의 길이가 3.84KB를 넘어서면서 부터는 제안 기술이 더 좋은 효과를 보이는 것으로 나타났다.



[그림 10] 전체 비용의 비교

6. 결론

본 논문에서는 SDN과 DHT를 이용하여 IP의 기능인 식별자와 위치자의 역할을 분리하기 위한 기술을 제안하였으며, 등록 절차에서 패킷 전달 과정 및 경로 최적화 과정에 대한 프로세스를 구체적으로 묘사하였고, 각 단계에 필요한 컨트롤러의 알고리즘을 설명하였다. 제안한 기술은 기존의 네트워크를 HI 도메인과 IP 도메인으로 오버레이 방식으로 구성하여, 라우팅할 수 없는 단말의 식별자 ID를 라우팅 가능하도록 설계하였다.

제안 기술은 네트워크 기반으로 동작하기 때문에, 단말이 시그널링 과정에 참여하지 않을 뿐만 아니라 ID와 LOC 간의 변환이나 매핑 정보의 관리 등의 과정을 고려할 필요도 없다는 장점이 있고, 기존의 네트워크 기반 프로토콜에서 사용하던 터널링을 제거함으로써 패킷 처리에 대한 오버헤드를 감소시킬 수 있다. 분석 결과로부터 제안한 기술의 시그널링에 대한 비용은 비교대상인 MOFI의 시그널링 비용보다 크게 보이지만, 패킷 전송 비용을 고려한 전체 비용의 비교 그래프는 세션 길이가 특정 범위를 넘어설 경우, MOFI의 운영은 제안한 방식보다 더 많은 비용을 요구한다는 것을 확인할 수 있었다.

REFERENCES

[1] R. Moskowitz, P. Nikander, P. Jokela, ed., T. Henderson, Host Identity Protocol, IETF, RFC 5201, 2008.

[2] E. Nordmark, M. Bagnulo, shim6: Level 3 Multihoming Shim Protocol for IPv6, IETF, RFC 5533, 2009.

[3] D. Farinacci, V. Fuller, D. Meyer, D. Lewis, The Locator/ID Separation Protocol(LISP), IETF, RFC 6830, 2013.

[4] H. Luo, Y. Qin, H. Zhang, A DHT-Based Identifier-to-Locator Mapping Approach for a Scalable Internet, IEEE Transactions on Parallel and Distributed Systems, vol.20, issue 12, pp. 1790-1802, Dec. 2009.

[5] http://www.opennetworking.org

[6] ONF White Paper, Software-Defined Networking: The New Norm for Networks, https://www.opennetworking.org/images/stories/downloads/sdnresources/white-papers/wp-sdn-newnorm.pdf, Open Networking Foundation.

[7] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, A scalable content-addressable network, SIGCOMM'01 , pp. 161-172, Aug. 2001.

[8] Open Networking Foundation, OpenFlow Switch Specification Version 1.3.0 (Wire Protocol 0x04), Jun. 2012.

[9] H. Kang, J. Kim, S. Koh, DHT-based Identifier-Locator Mapping Management for Mobile Oriented Future Internet, 18th Asia-Pacific Conference on Communications (APCC), pp. 786-791, Oct. 2012.

[10] J. Kim, H. Jung, S. Koh, Mobile Oriented Future Internet(MOFI):Architectureal Design and Implementations, ETRI Journal, vol.35, no.4 pp. 666-676, Aug. 2013.

[11] P. Nikander, J. Laganier, F. Dupont, An IPv6 Prefix for Overlay Routable Cryptographic Hash Identifiers(ORCHID), IETF, RFC 4843, 2007.

[12] S. Deering, R. Hinden, Internet Protocol, Version 6(IPv6), IETF, RFC 2460, 1998.

[13] S. Thomson, T. Narten, T. Jinmei, IPv6 Stateless Address Autoconfiguration, IETF, RFC 4862, 2007.

[14] J. Lee, T. Chung, and S. Gundavelli, A Comparative Signaling Cost Analysis of Hierarchical Mobile IPv6 and Proxy Mobile IPv6, IEEE 19th International Symposium, Personal, Indoor and Mobile Radio Communications, pp. 1-6, Sep. 2008.

이 찬 행(Chan-Haeng Lee)

- 2004년 2월 : 한신대학교 정보시스템공학과 졸업(학사)
- 2006년 2월 : 한신대학교 컴퓨터학과 졸업(석사)
- 2016년 2월 : 고려대학교 컴퓨터학과 졸업(박사)

<관심분야>

모바일 네트워크, SDN, VANET, WSN

민 성 기(Sung-Gi Min)

- 1988년 2월 : 고려대학교 컴퓨터학과 졸업(학사)
- 1989년 2월 : 런던대학교 컴퓨터학 졸업(석사)
- 1993년 2월 : 런던대학교 컴퓨터학 졸업(박사)
- 1994년~2000년 : LG정보통신
- 2000년~2001년 : 동의대학교 컴퓨터학과 교수
- 2001년~현재 : 고려대학교 교수

<관심분야>

SDN, 모바일 네트워크, QoS, 이동성 관리, VANET, 네트워크 프로토콜

최 창 원(Chang-Won Choi)

- 1990년 2월 : 고려대학교 전산과학과 졸업(학사)
- 1992년 2월 : 고려대학교 전산과학과 졸업(석사)
- 1995년 2월 : 고려대학교 전산과학과 졸업(박사)
- 1996년~현재 : 한신대학교 컴퓨터공학부 교수

<관심분야>

유무선 네트워크, 시스템 분석, 미래 인터넷