

콘텐츠 중심 네트워크에서 정보제공자의 이동성 지원을 위한 인기도 기반 푸싱 기법

우 태 희*, 박 흥 순°, 권 태 욱*

Popularity-Based Pushing Scheme for Supporting Content Provider Mobility in Content-Centric Networking

Taehee Woo*, Heungsoon Park°, Taewook Kwon*

요 약

콘텐츠 중심 네트워크(CCN)는 데이터를 찾는데 필요한 라우팅 정보를 콘텐츠 이름에서 찾는 방식으로 기존의 IP 방식과는 다른 새로운 네트워킹 패러다임이다. CCN의 도전과제 중 하나인 이동성 관리는 크게 정보요청자의 이동과 정보제공자의 이동을 들 수 있다. 그 중에서도 정보제공자 이동의 경우 해당되는 라우터의 경로 정보를 갱신해야 하므로 많은 오버헤드와 시간이 필요하다. 이에 본 논문은 CCN에서 효율적인 정보제공자의 이동성 지원을 위해 콘텐츠의 인기를 고려한 Popularity-based Pushing CCN(PoPCoN)을 제안한다. 제안하는 알고리즘은 기존의 기법과 비교하여 요청자의 콘텐츠 다운로드 시간을 단축시키고 네트워크의 오버헤드를 감소시킨다.

Key Words : Content-Centric Networking, Content Provider Mobility, Mobility Management, Wireless Networking

ABSTRACT

Content-Centric Networking(CCN) is a new networking paradigm to search for the routing information needed to find a data from the content name, unlike conventional IP networks. In CCN, the mobility management, one of the CCN challenges, is consists of consumer mobility and content provider mobility. Among both, in the case of the content provider mobility, it requires too much overhead and time to update routing information on the corresponding routers. In this paper, we propose Popularity-based Pushing CCN(PoPCoN) which considers the content popularity to support effective mobility of content provider in CCN. Our proposed algorithm shortens content download time for the consumer and reduces the network overhead during mobility as compared to the existing approaches.

I. 서 론

최근 이동통신 가입자의 급격한 증가에 따라 인터넷에서 전송되는 데이터의 양도 기하급수적으로 증가

하고 있다. 이에 급증하는 데이터 트래픽을 해결을 위해서 많은 연구들이 진행 중에 있으며 특히, 정보중심 네트워크(Information-Centric Networking)는 효과적으로 콘텐츠를 전파하기 위한 미래의 네트워크 아키텍처

* First Author : Dept. of Computer Science & Engineering, Korea National Defense University, seeles0@gmail.com, 학생회원

° Corresponding Author : Dept. of Computer Science & Engineering, Korea National Defense University, heungsoon.park@gmail.com, 종신회원

* Dept. of Computer Science & Engineering, Korea National Defense University, kwontw9042@naver.com, 종신회원

논문번호 : KICS2014-09-357, Received September 22, 2014; Revised December 7, 2014; Accepted January 5, 2015

텍처로 부각되고 있다¹⁾.

그 중에서도 본 논문에서는 대표적인 정보중심 네트워크의 하나인 Van Jacobson의 Content-Centric Networking(CCN)²⁾에 초점을 맞춰 연구하였다. CCN에서 정보요청자는 콘텐츠의 위치에 상관없이 원하는 콘텐츠의 이름으로 요청패킷을 전송하고, 패킷을 받은 라우터나 정보제공자는 상응하는 콘텐츠를 찾아 응답한다. 따라서 CCN은 정보요청자의 이동성 지원에 유리한 구조를 가지나 정보제공자가 이동할 경우 다양한 문제가 대두되며 이를 해결하기 위한 기법이 연구되고 있다^{3,4)}.

그림 1에서 정보제공자의 이동성 문제를 살펴보자. 정보요청자는 콘텐츠 ‘A.com/movie.avi’를 정보제공자에게 요청중이며, 중간의 CCN 라우터 R1~R3의 라우팅 테이블에는 해당 콘텐츠의 위치정보가 도메인 A로 되어있다. 이 때 정보제공자가 도메인 A에서 도메인 B로 이동을 하면, 정보요청자는 R2와 R3의 라우팅 테이블이 갱신될 때까지 해당 콘텐츠를 찾을 수 없으며, 변경을 위해 소요되는 비용도 크다³⁾. 본 논문에서는 정보제공자의 이동에서 기인되는 문제점을 해결하기 위해서 Popularity-based Pushing CCN(PoPCoN)모델을 제안한다. 정보제공자는 이 기법을 활용한 이동성 지원으로 콘텐츠 전송지연을 감소시키며, 정보요청자도 불필요한 오버헤드를 줄일 수 있다.

본 논문의 구성은 다음과 같다. 2장은 CCN에 대한 소개와 정보제공자 이동성지원에 대한 관련연구를 제시한다. 3장에서는 정보제공자의 이동성 문제를 해결하기 위한 PoPCoN 모델을 소개하고, 4장은 제안한 모델의 성능을 평가하고 분석한다. 마지막 5장에서 결론과 향후연구를 제시한다.

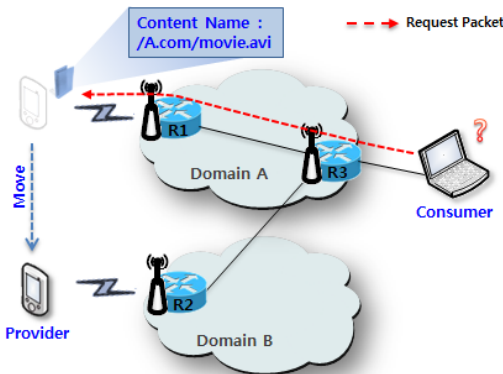


그림 1. CCN에서 정보제공자 이동성 문제
Fig. 1. Provider mobility issue in CCN

II. 관련 연구

2.1 Content-Centric Networking

CCN에서의 패킷은 Interest 패킷과 Data 패킷으로 구분된다. Interest는 정보요청자가 콘텐츠를 찾기 위해서 발행하는 패킷이며, Data는 Interest에 대한 응답으로써 콘텐츠 청크(chunk)를 수반하고, Interest의 역경로로 이동하여 정보요청자에게 전달된다²⁾.

CCN의 라우터의 구조는 Content Store(CS), Pending Interest Table(PIT), Forwarding Information Base(FIB)로 구성되며 CCN 패킷의 포워딩 시 다음과 같이 순차적으로 활용된다²⁾. Interest의 포워딩 시 Interest가 CCN 라우터에 도착하면, 라우터는 CS에서 Interest의 이름과 일치하는 콘텐츠가 있는지 확인을 한다. 만약 있다면, 해당 콘텐츠를 Interest가 들어온 인터페이스로 보낸다. 일치하는 콘텐츠가 없다면 라우터는 PIT에 일치하는 항목이 있는지 확인을 하며, PIT 항목에 있다면, 라우터는 단순히 새로 들어온 Interest의 인터페이스를 추가한다. 만약 PIT에 일치하는 항목이 없다면, 라우터는 FIB에 일치하는 항목이 있는지 확인을 하고, FIB에 일치하는 항목이 있다면 라우터는 PIT에 새로운 항목을 추가한 후에 일치하는 FIB 항목에 따라 Interest를 보낸다. FIB 항목에도 없다면 해당 Interest는 폐기된다. Data의 포워딩 절차는 다음과 같다. Data 패킷이 CCN 라우터에 도착하면, 라우터는 CS에 도착한 Data와 일치하는 콘텐츠가 있는지 확인을 한다. 이미 일치하는 항목이 있다면 도착한 Data 패킷은 삭제한다. CS에 없다면 라우터는 PIT에 일치하는 항목이 있는지 확인을 한다. 일치하는 항목이 있다면 라우터는 도착한 Data 패킷을 CS에 저장하고, 일치하는 PIT 항목에 명시된 인터페이스로 Data 패킷을 보낸다. Data를 보낸 후에 해당 PIT 항목을 삭제한다. PIT에 일치하는 항목이 없다면 Data 패킷은 삭제된다.

2.2 CCN에서 정보제공자의 이동성 지원

정보제공자가 다른 네트워크 도메인으로 이동하는 경우, 제공자는 기존의 경로를 통해 보내어지는 Interest를 더 이상 수신할 수 없게 된다. 가장 단순한 해결책으로는 정보제공자가 새로운 도메인에서 자신이 제공하는 콘텐츠 이름을 전체 네트워크로 알리는 것이다. 즉 정보제공자에서 요청자까지 연결된 라우터들의 FIB 테이블이 교체되어야 하는데 시간이 많이 소요되고, 이동 단말이 빈번히 이동하는 경우에는 많은 오버헤드 발생을 초래한다.

정보제공자의 이동성 지원에는 크게 에이전트 지원 방식과 네트워크 지원방식이 있다^[5]. 에이전트 지원방식은 정보제공자가 다른 도메인으로 이동하였을 때 이전 콘텐츠 이름과 새로운 콘텐츠 이름을 바인딩 시켜주는 대행자를 두는 방식으로, Rendezvous-based(RD)^[6]와 Indirection-based(ID)^[7] 등의 기법이 있다. RD 기법에서 정보제공자가 새로운 도메인으로 이동하게 되면, Home Repository(HR)라는 에이전트로 자신의 위치정보가 포함된 Interest를 전송한다. HR은 콘텐츠 이름과 새로운 위치정보가 포함된 라우팅 정보를 서로 바인딩 한다. 정보요청자는 정보제공자의 위치에 관계없이 HR에게 Interest를 전송하게 되고, HR은 정보제공자에게 Interest를 전달한다. 하지만 RD 기법은 Data가 Interest의 역경로를 따르므로 항상 HR을 통해 전송될 수밖에 없게 되어 전달 지연 시간이 증가하게 된다. ID 기법은 RD의 문제점을 Data에 위치정보를 관리함으로써 해결한다. 정보제공자가 이동한 후에 HR에게 위치정보를 알리면 정보요청자는 Interest를 HR에게 보내는 원리는 RD와 동일하다. 하지만 정보요청자가 첫 번째 Data 패킷을 받게 되면, 수신된 위치정보를 활용하여 정보제공자에게 직접 Interest를 보낼 수 있다. 하지만 ID기법은 정보제공자가 자주 이동을 하는 환경에서는 정보제공자의 위치정보가 갱신되기 전에 정보요청자가 Interest를 전송하는 경우가 발생한다. 또한 위와 같은 에이전트 지원방식은 단일 경로를 사용하기 때문에, 다중 경로가 가능한 CCN에서는 비효율적이다.

네트워크 지원방식은 네트워크의 라우팅 테이블을 갱신하는 방법으로 Interest forwarding(IF)^[6] 기법과 Zone flooding(ZF)^[5] 기법이 있다. IF기법에서 정보제공자는 이동하기 전에 라우터에게 이동정보를 알리고, 이동정보를 받은 라우터는 정보제공자의 위치가 갱신될 때 까지 Interest를 버퍼에 저장한다. 정보제공자의 이동 완료 후 정보제공자의 가상 Interest 전송으로 중간 라우터들은 FIB 정보를 갱신하며 Interest를 갱신된 위치로 포워딩 한다. 이 방법은 버퍼로 인해 Interest의 손실을 줄일 수 있고, 지역적으로 연관된 라우터만 라우팅 정보를 갱신하여 비용을 줄일 수가 있다. ZF 기법은 다중 Interest와 Zone 개념을 추가하였다. Zone은 고정된 라우터의 지리적인 위치에 의해서 정해지며 정보제공자가 이동하게 될 후보 라우터 위치에 대한 정보를 갖는다. 정보제공자 이동 시 정보요청자는 Zone 정보를 바탕으로 다중 Interest를 존 내부의 모든 라우터에게 전송함으로써 정보제공자가 Zone의 어느 위치를 이동하든지 상관없이 Interest를

받을 수 있다. Zone flooding 방법을 이용하면 다른 기법에 비해 지연시간을 줄일 수 있다. 하지만 다중 Interest로 인해 네트워크에 불필요한 트래픽이 발생하게 된다.

2.3 인기도 기반의 콘텐츠 관리

인기도가 높은 콘텐츠는 정보요청자들로 하여금 많은 중복된 요청이 있기 때문에 이 콘텐츠를 우선적으로 관리하는 기법이 필요하다^{[8],[9]}. [8]에서는 네트워크를 여러 개의 클래스와 셀로 구분하고 각각의 셀에는 콘텐츠가 반드시 하나 이상 유지될 수 있도록 알고리즘을 구현하였다. 조밀한 셀에는 인기가 높은 콘텐츠가 유지되며, 인기가 낮은 콘텐츠는 넓은 셀에 콘텐츠가 유지된다. 높은 인기도의 콘텐츠들이 조밀하게 위치하고 있기 때문에 정보요청자들은 해당 콘텐츠를 낮은 인기도의 콘텐츠보다 상대적으로 빠르게 접근을 할 수 있으며, 그 결과 전체적인 홉 수 및 트래픽이 감소한다.

[9]에서는 정확하고 신속한 콘텐츠 접근을 위해 인기 있는 콘텐츠를 요청하지도 않은 인접 노드에게 전달하는 기법을 제안한다. 동일한 콘텐츠의 요청횟수가 일정 수치에 도달하면 이웃노드에게 해당콘텐츠를 수신하라고 제안하는 메시지를 전달한다. 이는 인접 노드에 확산되며 많은 노드들은 해당 인기도 높은 콘텐츠를 보유하게 된다. 차후 정보요청자들이 해당 콘텐츠를 필요로 하면, 주변 노드에서 콘텐츠를 찾게 되어 전체적인 네트워크 자원 소비를 줄일 수 있다.

지금까지 설명한 콘텐츠 관리 기법은 정보요청자에게 최대한 가까운 위치에 콘텐츠를 놓았기 때문에 콘텐츠 요청을 하였을 때 신속하게 콘텐츠를 받을 수 있었다. 이와 같이 정보제공자도 인기 콘텐츠의 분포 특징을 이용한다면 자원을 보다 효율적으로 사용할 수 있을 것으로 판단된다.

III. 인기도 기반 푸싱 기법 (PoPCoN)

본 논문에서는 CCN에서 콘텐츠 제공자의 이동성 지원을 위해 인기도를 활용한 Popularity-based Pushing CCN(PoPCoN)을 제안한다. 먼저 PoPCoN의 기본개념에 대해서 살펴보고 시스템 설계와 동작 알고리즘에 대해서 설명한다. PoPCoN은 가정사항은 다음과 같다.

- 정보제공자와 정보요청자 사이의 중간 네트워크 노드는 이동성이 없다.

- 링크계층 이동성 예측 기법에 의해서 정보제공자의 이동 임박시점을 알 수 있다(사용된 이동성 예측기법은 4장에서 설명한다).

3.1 기본개념

PoPCoN은 인기있는 콘텐츠는 많은 노드들이 보유한다는 전제하에 동작하며, 이동에 임박한 정보제공자는 보유 콘텐츠를 ‘요청한 이웃노드(이하 요청노드)’에게 전달하고 이동하는 것이 기본원리이다. 정보제공자가 이동하기 전에 콘텐츠를 주변노드에 전달을 하면 정보요청자는 지연없이 콘텐츠를 받을 수 있으며, 이동한 후에도 추가적인 처리를 하지 않아도 된다. 하지만 주변에 이미 해당 콘텐츠를 가지고 있는 이웃노드가 있다면 불필요하게 콘텐츠를 전달하지 않고 이동해도 될 것이다. 해당 콘텐츠의 인기도가 높으면 주변 노드에 콘텐츠를 보유할 가능성이 높기 때문에 인기도가 높은 콘텐츠일수록 높은 확률로 주변을 탐색한다. 주변 노드를 탐색하여 해당 콘텐츠의 중복보유가 확인되면 정보제공자는 해당 콘텐츠의 위치정보를 요청노드들과 공유하고 요청노드들은 콘텐츠를 보유한 노드에게 콘텐츠를 요청하여 받을 수 있다. 마찬가지로 낮은 인기도의 콘텐츠는 주변에 존재할 가능성이 낮기 때문에 정보제공자는 낮은 확률로 주변을 탐색한다. 이 확률적인 값을 인기율(Popularity ratio)로 정의하였으며, 이는 콘텐츠의 인기도에 의해서 계산된다. 이 인기율에 따라서 탐색을 하고 콘텐츠의 보유 여부를 확인하는 동작을 SCAN이라고 하며, 이웃노드 정보 또는 콘텐츠를 전달하는 동작을 PUSH라고 정의한다. 인기율과 SCAN, PUSH는 3.3절에서 자세하게 설명한다.

3.2 시스템 설계

3.2.1 패킷 설계

PoPCoN의 동작을 위해 새로운 유형의 패킷을 설계하였으며 M-Interest와 M-Data로 구분된다. M-Interest는 SCAN이 동작할 때, 정보제공자가 주변 이웃노드들의 콘텐츠 보유여부를 질의하기 위해 보내는 패킷이다. M-Interest는 기존의 Interest와 구조는 같으나 Mobility 옵션을 추가하여 콘텐츠의 보유여부가 확인되면 ACK로 수신된다. M-Data는 정보제공자가 PUSH 동작을 할 때 이웃노드 정보나 콘텐츠 전달을 위해 사용되는 패킷으로, 콘텐츠의 유통기한(Freshness)과 MAC주소 옵션을 갖는다. 유통기한은 M-Data로 보낸 콘텐츠의 불필요한 점유와 네트워크

에 무기한 캐싱되는 콘텐츠를 방지하기 위해서 사용되며, MAC은 이웃노드 주소정보를 전달하기 위해 사용된다.

3.2.2 포워딩 엔진 설계

PoNCoN은 기존 CCN의 포워딩 엔진과 패킷 포워딩 절차^[2]를 일부 수정하였다. 또한 콘텐츠에 대한 추가 정보를 기록하기 위해서 CS의 콘텐츠 항목 리스트에 4가지 항목을 추가하였다. M-Data의 유통기한을 저장하기 위한 항목과 인기율 항목, 정지와 이동 상태를 구분하기 위한 상태(state) 항목, 콘텐츠를 보유한 이웃노드 정보를 기록하기 위한 보유자(possessor) 항목이다. 상태 항목은 정지 상태일 때 N, 이동 상태일 때 M으로 기록된다.

Interest 포워딩 절차는 그림 2와 같으며, 점선으로 표시된 부분이 기존과 달리 새롭게 수정된 부분이다. Interest가 PoPCoN 노드에 들어오면 패킷 형태를 확인한다. 패킷 형태가 기존의 Interest이면 CCN의 처리 절차와 동일하게 처리하며, M-Interest 패킷이면 이동성 옵션을 확인한다.

Data 패킷 포워딩 절차는 그림 3과 같다. Data 패

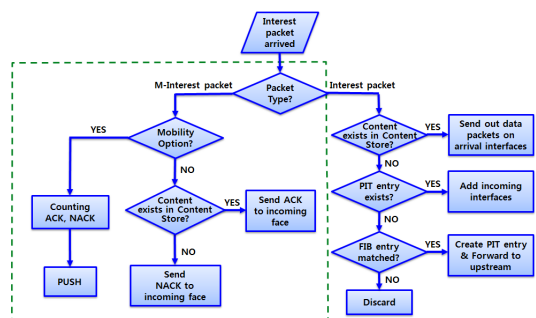


그림 2. PoPCoN에서 Interest 패킷 포워딩 절차
Fig. 2. Interest packet forwarding process in PoPCoN

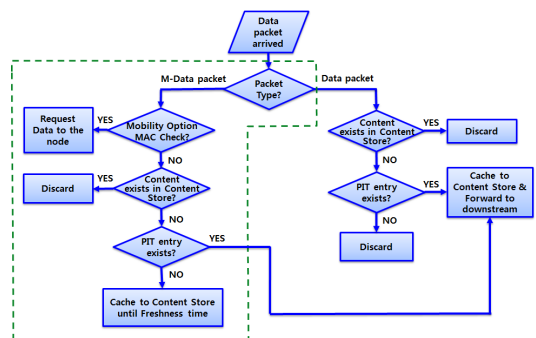


그림 3. PoPCoN에서 Data 패킷 포워딩 절차
Fig. 3. Data packet forwarding process in PoPCoN

킷이 PoPCoN 노드에 들어오면 패킷 형태를 확인하여 M-Data 패킷이면 PoPCoN 라우터는 이동성 옵션의 MAC 항목을 확인하고, 활성화가 되어 있다면 데이터 부분의 이웃노드 정보를 이용하여 콘텐츠를 요청한다. 만약 비활성화 되어 있다면, CS에 중복된 콘텐츠가 있는지 확인을 하고, 존재한다면 폐기한다. 존재하지 않으면, PIT를 확인하여 요청된 항목이 있으면 콘텐츠를 CS에 저장하고 해당 인터페이스로 보낸다. PIT에 없으면 콘텐츠를 CS에 저장하고 콘텐츠 유통기한까지 콘텐츠를 보관한다.

3.3 동작 알고리즘

PoPCoN은 이동에 임박했을 때 정보제공자의 이동성 지원을 위해 SCAN과 PUSH 동작을 수행한다. SCAN은 인기율에 따라 확률적으로 동작되며, 정보제공자가 이웃노드들에 콘텐츠 중복 보유여부를 확인하는 동작이다. PUSH는 정보제공자가 콘텐츠를 요청한 이웃노드에 Data 패킷이나 이웃노드 정보를 전달하는데 필요한 동작이다. 먼저 SCAN 동작에 필요한 인기율을 알아보고, 이어서 SCAN과 PUSH 동작을 설명한다.

3.3.1 인기율(Popularity ratio)

콘텐츠의 인기도는 정보요청자들이 콘텐츠를 참조하는 정도에 따라 정해진다고 보고, 이 참조확률(reference probability)은 Zipf-like 분포를 따른다고 가정한다^[10]. Zipf-like 분포에서 i 번째($1 \leq i \leq M$) 랭킹의 콘텐츠 C_i 의 참조확률은 $P_N(i)$ 으로 표현되며 식 (1)과 같다.

$$P_N(i) = \frac{1}{i^\alpha} \cdot \left(\sum_{j=1}^N \frac{1}{j^\alpha} \right)^{-1} \quad (1)$$

SCAN은 확률적으로 작동하게 되는데, SCAN시 사용되는 확률을 인기율 $P_p(i)$ 이라고 하며 식(2)와 같이 표현한다.

$$P_p(i) = \frac{P_N(i)}{P_N(1)} = \frac{1}{i^\alpha} \quad (2)$$

정보제공자는 콘텐츠의 랭킹을 알고 있다고 가정하며, $P_p(i)$ 는 가장 인기 있는 콘텐츠의 참조확률 $P_N(1)$ 을 기준으로 판단될 콘텐츠의 참조확률 $P_N(i)$ 의 비율로 표현하였다. 식 (2)에서 인기율은 콘텐츠의 순위와 Zipf 지수에 따라 가변하며, 어떤 콘텐츠가 선택되느냐에 따라서 그에 상응하는 확률이 주어진다. 인기율을 산출하는 알고리즘은 알고리즘 1과 같다.

나에 따라서 그에 상응하는 확률이 주어진다. 인기율을 산출하는 알고리즘은 알고리즘 1과 같다.

Algorithm 1. Pseudo-code of calculating process for popularity ratio

Input : DATA($C(i)$), $P_N(i)$

let i is content ranking
 let $C(i)$ is a content which has i th rank
 let $P_N(i)$ is $C(i)$'s Popularity concerning Zipf-like distribution
 let $P_p(i)$ is $C(i)$'s Popularity ratio

1 : **if** $C(i)$ is arrived **then**
 2 : $P_p(i) \leftarrow$ calculate $(1 / i^\alpha)$
 3 : write $P_p(i)$ in Popularity ratio entry of Content Store
 4 : **end if**

Output : $P_p(i)$

3.3.2 SCAN

SCAN은 정보제공자가 콘텐츠의 인기율에 따라서 확률적으로 구동되며 주변에 중복된 콘텐츠 보유노드가 있는지 확인한다. 정보제공자는 CS의 인기율 항목의 인기율과 랜덤함수를 이용하여 SCAN을 구동시킨다. SCAN이 구동되면 M-Interest를 브로드캐스팅 하고, CS에 상태 항목을 N에서 M으로 변경한다. M-Interest 패킷을 받은 이웃노드는 M-Interest의 콘텐츠 이름을 확인하고, 해당 콘텐츠보유를 확인한다. ACK를 보낸 이웃노드 정보는 CS의 Possessor 항목에 기록된다.

SCAN의 동작은 그림 4와 같이 묘사된다. 설명을 위한 시나리오로써 그림과 같이 1개의 정보제공자와 1개의 정보요청자, 4개의 이웃노드(N1~N4), 1개의

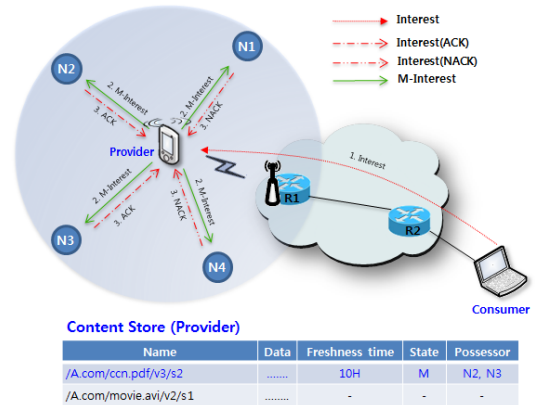


그림 4. SCAN 동작절차 예
 Fig. 4. An example of the SCAN process

요청노드(R1), 중간노드(R2)가 있다. N1, N4는 콘텐츠를 보유하지 않으며, N2, N3은 콘텐츠를 보유하고 가정하자.

- ① 정보제공자는 SCAN이 구동되면 M-Interest 패킷을 브로드캐스팅 하고, CS에 상태 항목을 N에서 M으로 변경한다.
- ② M-Interest를 받은 N1, N2, N3, N4는 해당 콘텐츠가 있는지 CS를 확인한다. N2, N3는 ACK를 정보제공자에게 전송하고, N1, N4는 NACK를 전송한다.
- ③ 정보제공자는 ACK를 보낸 N2, N3을 CS의 Possessor 항목에 기록한다.

정보제공자에서 처리되는 SCAN은 알고리즘 2와 같으며, 이웃노드들에서의 동작은 알고리즘 3과 같다.

Algorithm 2. Pseudo-code of SCAN (Provider)

Input : Popularity Ratio
 let i is content ranking, $C(i)$ is Content(i) in PIT
 let $Pp(C(i))$ is Popularity Ratio for Content(i)
 let nm is the number of neighbor nodes
 let s is State, p is Possessor in Content Store

```

1 : if s is N then
2 :   process packet forwarding as CCN
3 : else if s is M then
4 :   if Random([0,1]) <= Pp(C( $i$ )) then
5 :     forward M_INTEREST(C( $i$ )) to neighbor nodes
6 :     Timer start during 2 · RTT(1_hop)
7 :     For (until nm equal nack+nack and Timer is 0)
8 :       nack ← count a number of the received ACKs
9 :       nnack ← count a number of the received NACKs
10 :      p ← MAC addresses of nodes with ACKs
11 :    End For
12 :    PUSH(p), stop processing
13 :  end if
14 : else PUSH(p), stop processing
15 : end if
    
```

Output : M_INTEREST(C(i)), p

Algorithm 3. Pseudo-code of SCAN (Neighbor nodes)

Input : M_INTEREST(C(i))
 let i is content ranking
 let $C(i)$ is Content(i) in provider's Content Store

```

1 : if receive a M_INTEREST packet then
2 :   if possess C( $i$ ) then
3 :     send a ACK to Provider
4 :   end if
5 : else if not possess C( $i$ ) then
6 :   send a NACK to Provider
7 : end if
    
```

Output : ACK, NACK

알고리즘 2에서 타이머는 이웃노드들이 ACK, NACK를 보내기 위한 충분한 시간으로 설정하였으며, 본 논문에서는 평균 RTT의 2배로 설정하였다.

3.3.3 PUSH

PUSH는 SCAN에서 판단한 결과를 바탕으로 이웃노드 정보 또는 콘텐츠를 요청노드에게 전달하는 절차이다. 이 때, 이웃노드 정보를 전달할 경우 PUSH(MAC), Data 패킷을 전달할 경우에는 PUSH(Data)로 표현하자. PUSH(MAC)은 정보제공자가 이웃노드로부터 ACK를 수신했을 때 수행하는 동작으로, 정보제공자가 ACK를 수신하게 되면 ACK를 보낸 이웃노드의 노드정보를 요청노드에게 전달한다. M-Data를 받은 요청노드는 M-Data에 명시된 이웃노드의 노드정보를 확인하고, 해당노드에 Interest를 보내서 콘텐츠를 요청한다. 그림 5는 PUSH(MAC)의 동작을 제시한다. 정보제공자는 CS의 보유자 목록(N2, N3)을 M-Data 옵션의 MAC을 이용하여 R1에게 전달한다. 이 패킷을 받은 R1은 N2또는 N3에게 콘텐츠를 요청하여 수신 받고, 최종적으로 정보요청자에게 전달된다. N2, N3의 선택은 기존 CCN 포워딩 전략²⁾에 따라서 선택되어 진다. 정보제공자가 PUSH(Data) 동작을 구동하게 되면 콘텐츠를 M-Data를 이용하여 요청노드에게 전송한다. M-Data 패킷을 받은 노드는 PoPCoN 포워딩 절차에 따라서 PIT 항목에 있으면 Data 패킷을 해당 인터페이스로 전송하고, 항목에 없으면 Data를 CS에 저장한다. M-Data는 유통기한이 설정되어 보내지며, 요청노드는 M-Data를 유통기한 동안 CS에 저장한다. 이는 캐시교체정책에 의해 교체되지 않도록 우선권을 부여하여 해당기한 동안 정보요청자가 콘텐츠를 수신할 수 있도록 한다. 그림 6은

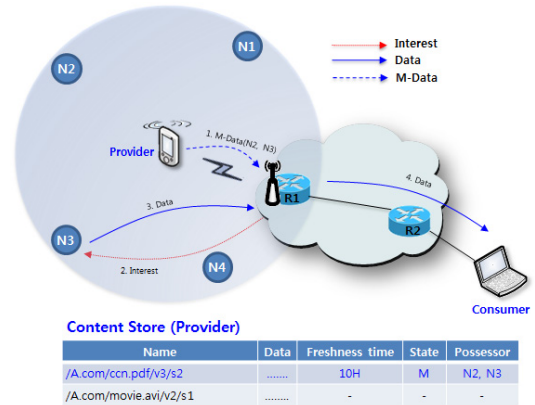


그림 5. PUSH(MAC) 동작절차 예
 Fig. 5. An example of the PUSH(MAC) process

IV. 성능평가

4.1 시뮬레이션 환경

PoPCoN의 성능을 평가하기 위해서 시뮬레이터인 ndnSIM^[11]을 수정하였고, 그림 7과 같이 실험 시나리오를 구성하였다. 전체 네트워크 크기는 500m × 500m이며, PoPCoN 라우터를 도메인 A에는 2개, 도메인 B에는 1개를 배치하였다. 도메인 A내 PoPCoN 무선노드는 10개를 배치하였고, Random Waypoint Mobility Model을 적용하였다. 정보제공자와 정보요청자는 각각 1개씩 배치하였고, 도메인 A에 위치한다. 무선 파라미터는 802.11g 모델을 사용하였고, 정보제공자 및 무선노드의 전송률은 24Mbps로 설정하였다. 정보요청자 연결링크는 1Mbps 속도에 지연시간 10ms로 설정하였으며, R1과 R3의 구간과 R2와 R3의 구간은 1Mbps 용량에 100ms 지연으로 설정하였다. 콘텐츠 크기는 300KB라고 가정하였으며, 각각 1KB의 chunk로 분할되어 진다. 콘텐츠의 인기도 설정을 위해서 정보요청자가 요청하는 콘텐츠는 ndnSIM의 Zipf Random Variable Model을 사용하였으며, 전체 콘텐츠 수 $n=100$, Zipf 지수 α 는 0.8로 설정하였다. 이웃노드들이 콘텐츠를 사전에 캐시되어 있는 환경을 조성하기 위해서 시뮬레이션을 일정시간 실행하여 이웃노드들에 콘텐츠가 캐시되도록 하였다. M-Data 패킷의 콘텐츠 유통기한은 이동성 실험을 위해서 실험의 영향에 미치지 않도록 시뮬레이션 동안 지속되도록 설정하였다.

정보요청자는 초당 10개의 Interest를 PoPCoN 라우터 R_3-R_1 를 경유하여 보낸다. 정보제공자는 도메인 A에서 도메인 B로 이동을 하며, 이동속도는 3~15m/s까지 3m/s씩 늘려가면서 실험하였다. 정보제공

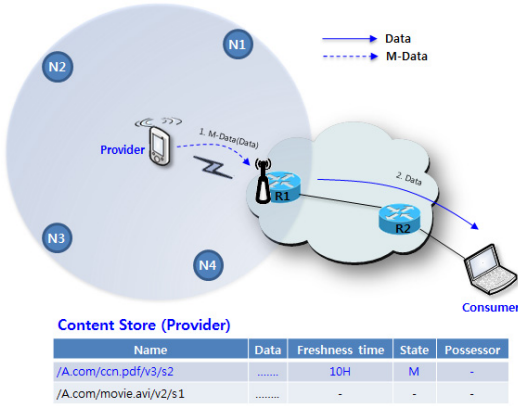


그림 6. PUSH(Data) 동작절차 예
Fig. 6. An example of the PUSH(Data) process

PUSH(Data) 동작을 보여준다. SCAN에서 이웃노드들이 아무도 콘텐츠를 보유하지 않았다고 판단했을 때, 정보제공자는 R1에게 콘텐츠를 전달한다. R1은 CS에 콘텐츠를 보관하고 정보요청자에게 콘텐츠를 제공한다. 정보제공자에서의 PUSH 동작(알고리즘 4)과 요청노드에서의 PUSH 동작(알고리즘 5)은 다음과 같다.

```

Algorithm 4. Pseudo-code of PUSH (Provider)

Input : p
let i is content ranking
let C(i) is Content(i) in provider's Content Store
let s is State in Content Store
let p is Possessor in Content Store

1 : if s is M and p is not empty then
2 :   push M-DATA(p) to incoming interface in PIT_ENTRY(C(i))
3 : else s is M and p is empty then
4 :   push M-DATA(C(i)) to incoming interface in PIT_ENTRY(C(i))
5 : else stop processing
6 : end if
Output : M-DATA(C(i)), M-DATA(p)
    
```

```

Algorithm 5. Pseudo-code of PUSH (Request nodes)

Input : M-DATA(C(i)), M-DATA(p)
let i is content ranking
let C(i) is Content(i) in provider's Content Store

1 : if receive a M_DATA(C(i)) packet then
2 :   store a Content Store with C(i)
3 : else if receive a M_DATA(p) packet then
4 :   send a Interest(C(i)) to p
12 : end if
Output : Interest(C(i))
    
```

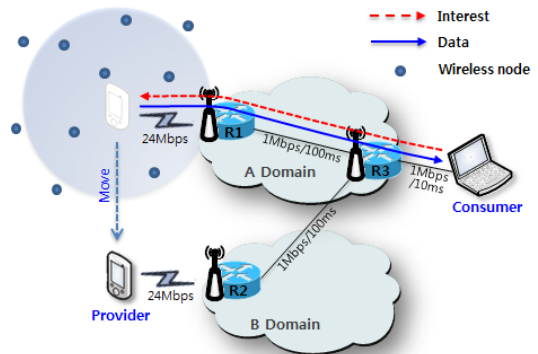


그림 7. 시뮬레이션 시나리오
Fig. 7. Simulation scenario

자의 이동성지점을 알기위해 이동성 예측기법을 이용하였으며, 이동성 예측 기법은 Dividing Sensitive Ranges Based Mobility Prediction Algorithm^[12]을 참조하였다. 예측기법에 필요한 파라미터로써 이동성 예측주기 $T=5s$, 모바일 노드의 최대 이동속도 $V_{max}=15m/s$, 1 range threshold $D=75m$, 모바일 노드의 평균이동속도 $\bar{v}=9m/s$, 2 range threshold $d=45m$ 를 사용하였다. d보다 크고 D보다 작은 구역을 2 range라고 불리며, 이곳에 위치하면 PoPCoN의 정보제공자는 SCAN을 시작한다. PoPCoN의 성능 평가를 위해서 CCN^[2], RD^[6], ID^[7], IF^[6], ZF^[5] 기법과 비교·분석을 하였다. RD, ID, IF, ZF는 [5]의 연구결과를 참조하여 구현하였으며, 앞서 제시한 동일한 파라미터를 사용하였다. RD, ID의 HR은 R_1 이 그 기능을 하도록 구현하여 실험하였다.

4.2 시뮬레이션 결과 및 분석

실험결과는 정보요청자의 관점에서 콘텐츠 다운로드 완료시간(download time)과 재요청수(number of retransmitted packets) 결과를 비교·분석 하였다. 다운로드 시간은 1개의 완전한 콘텐츠를 받기 위해서 소요되는 시간을 측정하였고, 재요청수는 1개의 콘텐츠를 받는데 필요한 300개의 Interest 패킷보다 타임아웃등의 이유로 추가적으로 재 발행된 패킷들을 측정하였다.

4.2.1 다운로드 완료시간

그림 8은 정보제공자 이동속도에 따른 다운로드 완료시간을 측정한 결과이다. 전체적으로 모든 방식들은 속도가 증가함에 따라서 다운로드 완료시간이 증가하는 추세를 보인다. 이는 이동속도가 빨라짐에 따라서 패킷 손실이 발생하여 생기는 현상으로 판단된다.

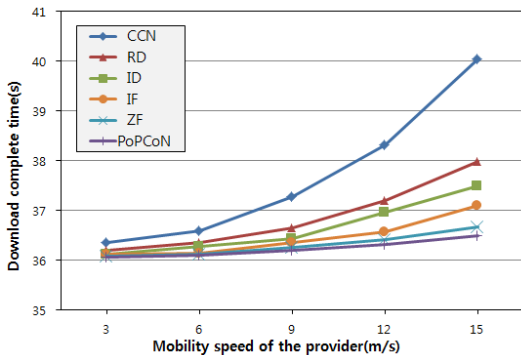


그림 8. 콘텐츠 다운로드 완료시간
Fig. 8. Content download time

CCN의 경우 이동속도의 증가 할수록 다른 방식들에 비해서 다운로드 완료시간이 가장 급격히 증가하는 것을 볼 수 있다. CCN에서 이 같은 결과는 속도가 증가함에 따라서 재전송 횟수가 많아지고 정보제공자의 도메인이 변경되어 FIB 테이블이 수정되는데 많은 시간이 소요되기 때문이다. 에이전트 지원 방식인 RD와 ID는 CCN에 비해서 빠른 결과를 보였지만 네트워크 지원 방식과 PoPCoN 보다는 느린 결과를 보인다. RD와 ID는 도메인 이동후에 HR인 R_1 과 소통하며 처리하는데 지연시간이 발생하고, R_1 에서 이전 콘텐츠 이름과 새로운 콘텐츠 이름을 바인딩 시키는데 추가로 처리시간이 발생하였기 때문에 나온 것으로 분석된다. ID는 새로운 장소로 이동한 후 R_1 이 정보요청자에게 Data 패킷을 보낼 때, 위치정보를 같이 보내기 때문에 지연시간이 줄어들어 RD보다 빠른 결과를 보인다.

네트워크 지원방식인 IF와 ZF는 CCN, RD, ID보다는 빠른 결과를 보였다. IF는 이동한 후에 R_1 에게 가상 Interest 패킷을 보내면서 R_3 의 FIB 테이블까지 변경하여 지연시간을 감소시켰으며, R_1 의 버퍼기능으로 재전송 패킷으로 인한 지연시간을 제거하였기 때문에 빠른 결과를 보였다. ZF의 경우는 이동한 후에 추가로 R_1 에게 요청을 하지 않아도 Interest 패킷이 수신되므로 IF보다 더 빨랐다. ZF와 많은 차이를 보이지 않지만 가장 좋은 결과를 보인 PoPCoN은 이동하기 전에 콘텐츠를 R_1 에게 전달하거나 콘텐츠를 가진 이웃노드 정보를 전달하였기 때문에 도메인 변경에 따른 지연시간이 발생하지 않아서 빠른 다운로드 완료시간을 갖는다. 한편, 15m/s의 가장 빠른 이동속도 환경에서 PoPCoN은 CCN, RD, ID, IF, ZF에 비교해서 각각 약 3.5초, 1.5초, 1초, 0.6초, 0.2초정도 차이를 보였다.

4.2.2 Interest 패킷 재전송 수

그림 9는 정보제공자 이동속도에 따라서 발행되는 정보요청자의 Interest 패킷 재전송 수를 보여준다. 전체적으로 모든 기법들이 높은 이동속도 환경에서 낮은 속도보다 많은 패킷을 전송하였음을 보여준다. 이것은 높은 이동환경에 따라서 Interest 패킷이 손실되어 재전송한 패킷수가 늘어났기 때문이다. PoPCoN은 이동하기 전에 콘텐츠를 전달하거나 위임하여 이동하는 동안 공백을 없앴기 때문에 가장 적은수의 재요청수를 보인다. CCN은 도메인을 이동한 후에 FIB가 갱신되기 전까지의 시간동안 정보요청자는 도메인 A로

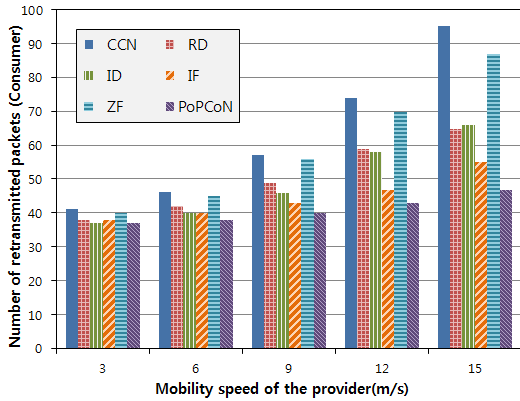


그림 9. 재전송된 Interest 패킷 수
Fig. 9. The number of retransmitted Interest packets

계속해서 Interest 패킷을 보내므로 가장 많은 재전송수가 발생하였다. RD와 ID는 정보제공자가 이동한 후 R_1 으로 새로운 위치정보를 보내기 전까지 이전 위치로 Interest 패킷을 보내게 되므로 많은 패킷 손실이 발생한다. 하지만 CCN의 FIB 갱신시간보다는 새로운 위치로 이동과 동시에 R_1 으로 위치정보를 보내는 시간이 짧기 때문에 적은 패킷손실이 발생하고 재요청수가 더 적었다. IF의 경우는 R_1 에 버퍼기능을 두었기 때문에 패킷손실이 적게 발생하였다. 하지만 정보제공자가 이동하는 도중에 R_1 에 저장된 Interest 패킷이 새로운 장소의 정보제공자에게 되돌아오는 과정에서 R_3 의 기존 PIT에 요청된 항목이 있기 때문에 R_2 로 패킷이 전달되지 못하고 타임아웃되어 재전송이 발생된다. 다운로드 완료시간에서 PoPCoN과 비슷한 결과를 보인 ZF는 CCN 다음으로 많은 수의 재전송 패킷을 발행하였다. 이 결과는 정보제공자가 이동하는 시간동안 R_2 뿐만 아니라 R_1 에도 패킷을 지속적으로 발행하였기 때문에 많은 패킷 손실이 발생하였다. ZF는 정보제공자가 이동하고 콘텐츠를 전송하는 시간에 대해서는 좋은 성능을 보이지만, 이 성능을 위해서 정보요청자의 많은 패킷발행이 필요하게 된다. 따라서 많은 정보요청자가 콘텐츠를 요청하는 경우에는 대역폭 낭비 및 중간 라우터들의 부하발생 등 성능저하가 예상된다. 한편, 15m/s의 가장 빠른 이동속도 환경에서 PoPCoN은 CCN, RD, ID, IF, ZF에 비교해서 각각 48개, 18개, 19개, 8개, 40개 차이를 보였다.

V. 결론 및 향후연구

본 논문에서는 미래 네트워크 아키텍처로 주목받고

있는 CCN의 정보제공자 이동성 지원 향상을 위해 PoPCoN을 제안하였다. PoPCoN은 콘텐츠의 인기도를 고려하여 해당 콘텐츠를 전달하고 이동한다. 이 동작을 지원하기 위해서 패킷과 포워딩 엔진을 재설계하였고, 주변의 콘텐츠를 탐색하는 SCAN과 콘텐츠 및 이웃노드 정보를 전달하는 PUSH 기능을 구현하였다. 이를 시뮬레이션을 통해서 성능평가하였으며, CCN과 기존의 여러 기법과 비교·분석한 결과 다운로드 완료시간과 Interest 패킷의 재전송수에 대해서 PoPCoN이 다른 기법에 비해 성능이 향상되었다.

PoPCoN은 다음에 대한 향후 연구가 필요하다. 첫째, 여러 노드들이 동시에 움직이는 MANET상황을 고려한 복합적인 이동성 지원이 필요하다. 둘째, 여러 정보요청자들의 각각 다른 콘텐츠 요청과 실시간 데이터를 요청하는 경우의 이동성 지원이다. 단일 정보요청자 환경은 현실적인 네트워크 환경이 아니며, 정보요청자들은 저장된 데이터뿐만 아니라 스트리밍 데이터와 같은 형태의 데이터들도 요청한다. 여러 정보요청자들의 콘텐츠를 요청하는 상황이라면 우선순위를 고려하여 콘텐츠를 선택하여 전달하는 방안과 스트리밍 데이터 요청에 대해서 정보제공자가 이동하기 전의 PUSH와 이동한 후의 PUSH를 연계하는 방안이 추가 연구되어야 한다. 마지막으로, 이동할 시점을 예측하기 위해서 링크계층의 이동성 예측 기법에 의존하였는데, 이동성 예측 기법이 실패할 경우와 연계한 방안도 고려하여 PoPCoN의 강건성을 높여야 할 것이다.

References

- [1] B. Ahlgren, et al., "A survey of information-centric networking," *IEEE Commun. Mag.*, vol. 50, no. 7, pp. 26-36, Jul. 2012.
- [2] V. Jacobson, et al., "Networking named content," in *Proc. 5th Int. Conf. Emerging Netw. Experiments and Technol.*, pp. 1-12, Rome, Italy, Dec. 2009.
- [3] G. Tyson, et al., "A survey of mobility in information-centric networks," *Commun. ACM*, vol. 56, no. 12, pp. 90-98, Dec. 2012.
- [4] J. Wang, R. Wakikawa, and L. Zhang, "DMND: collecting data from mobiles using named data," in *Proc. IEEE Veh. Netw. Conf.*, pp. 49-56, Jersey City, NJ, Dec. 2010.
- [5] D. Kim, et al., "End-to-end mobility support in content centric networks," *Int. J. Commun.*

Syst., Feb. 2014.

- [6] D. Kim, et al., "Mobility support in content centric networks," in *Proc. ACM ICN Workshop on Information-Centric Netw. (ICN '12)*, pp. 13-17, South Carolina, USA, Aug. 2012.
- [7] F. Hermans, E. Ngai, and P. Gunningberg, "Global source mobility in the content-centric networking architecture," in *Proc. 1st ACM workshop on Emerging Name-Oriented Mob. Netw. Design - Architecture, Algorithms, Appl. (NoM '12)*, pp. 13-18, Jun. 2012.
- [8] S. Lee, et al., "Content management in a mobile ad hoc network: Beyond opportunistic strategy," *Int. J. Commun. Netw. and Distrib. Syst.*, vol. 10, no. 2, pp. 123-145, Jan. 2013.
- [9] C. Bernardini, T. Silverston, and O. Festor, "MPC : Popularity-based caching strategy for content centric networks," in *Proc. IEEE Int. Conf. Commun.*, pp. 3619-3623, Budapest, Hungary, Jun. 2013.
- [10] L. Breslau, et al., "Web caching and Zipf-like distributions: evidence and implications," in *Proc. IEEE INFOCOM '99*, vol. 1, pp. 126-134, New York, USA, Mar. 1999.
- [11] A. Afanasyev, I. Moiseenko, and L. Zhang, *ndnSIM: NDN simulator for NS-3*, NDN, Tech. Rep. NDN-0005, Los Angeles, CA, Oct. 2012.
- [12] Y. Tang, et al., "Dividing sensitive ranges based mobility prediction algorithm in wireless networks," in *Proc. 6th Int. Wirel. Commun. Mob. Computing Conf. (IWCMC '10)*, pp. 1223-1227, 2010.

우 태 희 (Taehee Woo)



2006년 : 숭실대학교 정보통신
전자공학부 졸업
2015년 : 국방대학교 컴퓨터공
학과 석사
2015년 1월~현재 : 육군 3사관
학교 사이버전학과 순환직
교수

<관심분야> Next Generation Networking, Content-Centric Networking, MANET

박 흥 순 (Heungsoon Park)



2002년 : 육군사관학교 전산학과
졸업
2007년 : Air Force Institute of
Technology 컴퓨터공학과 석사
2012년 1월~현재 : 국방대학교
컴퓨터공학과 박사과정

<관심분야> Military Communications, MANET, Content-Centric Networking, Internet of Things, Network Security, Cyber Warfare

권 태 욱 (Taewook Kwon)



1986년 : 육군사관학교 전산학
과 졸업
1995년 : 미국 해군대학원 컴퓨
터공학과 석사
2001년 : 연세대학교 컴퓨터공
학 박사
2007년~현재 : 국방대학교 컴퓨
터공학과 교수

<관심분야> Next Generation Networking, Sensor Networking, Cloud Computing, RFID/USN, Virtual Reality, Content-Centric Networking, Software Defined Networking