

# 콘텐츠 중심 네트워킹에서 IGPT를 이용한 적시성 및 보안성 향상 방안

정승훈\*, 박흥순°, 권태욱\*

## A Method for Enhancing Timely-Delivery and Security Using IGPT in Content-Centric Networking

Seunghoon Jung\*, Heungsoon Park°, Taewook Kwon\*

### 요약

최근에 기존 IP기반과는 다른, 정보 중심 네트워킹이 부각되고 있으며 그 중의 하나인 Content-Centric Networking(CCN)은 Van Jacobson에 의해 처음 제안되어 대표적인 정보 중심 네트워킹 패러다임으로 활발히 연구되고 있다. CCN은 콘텐츠 자체에 대한 접근과 전달에 목표를 두고 라우터에서 콘텐츠의 배분이 가능토록 하여 IP 방식보다 좀 더 빠르게 콘텐츠가 전달되도록 하는 장점을 가지고 있다. 하지만, CCN은 정보요청자 기준의 통신(pull-based)으로 사전 요청패킷의 전송없이 정보를 분배하는 것이 제한된다. 또한, PIT(Pending Interest Table)의 사용으로 DDoS 공격 등을 받을 시 PIT 메모리 오버로드의 문제점이 있다. 이에 본 논문은 CCN에서 요청패킷 없이 정보를 전달하는 push-based 기법과 PIT 오버로드 상황에서 데이터 전달이 가능하도록 Interest Group Pushing Table(IGPT)을 활용한 정보전송 방법을 제안한다. 제안하는 기법은 데이터 전송의 신속성 및 보안성 향상 측면에서 기존의 기법보다 동일 시간동안 많은 양의 콘텐츠 전송이 가능하다.

**Key Words** : Content-Centric Networking, Pending Interest Table, PIT overload, Distributed Denial of Service

### ABSTRACT

Recently, Information-Centric Networking(ICN), different from traditional IP-based networking, has been highlighted. Content-Centric Networking(CCN), proposed by Van Jacobson, is a representative scheme of the ICN architectures. It can deliver messages slightly faster than the IP-based networking by focusing on the access and delivery to the content itself. However, CCN is restricted to distribute the information without transmitting the request packet in advance because it is pull-based architecture by content requester. In addition, it has a problem that the Pending Interest Table(PIT) could be overloaded easily when DDoS attack happens. In this paper, we suggest an algorithm using a push-based scheme without request packets and overcoming PIT overload situation by Interest Group Push Table(IGPT). The proposed scheme enables to transmit a large amount of content than an existing scheme during the same amount of time in terms of timely-delivery and security.

\* First Author : Dept. of Computer Science & Engineering, Korea National Defense University, hoonyjung9999@gmail.com, 학생회원  
 ° Corresponding Author : Dept. of Computer Science & Engineering, Korea National Defense University, heungsoon.park@gmail.com, 학생회원  
 \* Dept. of Computer Science & Engineering, Korea National Defense University, kwontw9042@naver.com, 종신회원  
 논문번호 : KICS2014-05-207, Received May 30, 2014; Revised August 19, 2014; Accepted November 11, 2014

## I. 서 론

Content-Centric Networking(CCN)은 기존의 IP 네트워킹 방식과는 달리, 정보 그 자체에 대한 접근을 우선으로 하여 원하는 곳에 빠르게 전달하는 것을 목표로 하고 있다<sup>1)</sup>. CCN의 정보전달 방식은 정보 요청자 중심의 자료전송으로, pull-based의 전송방식이다. 이는 정보 요청자가 전송한 콘텐츠 이름을 갖고 있는 Interest packet에 대한 응답으로 정보 제공자가 해당되는 Data packet을 전송함을 의미한다. 하지만, 특정 어플리케이션들은 정보 제공자 중심으로 정보를 신속하게 전달하는 push-based의 데이터 전송방법을 필요로 한다. 예를 들면, 실시간 통화를 요구하는 VoIP 서비스 같은 경우나<sup>2)</sup> 인터넷 뉴스와 같은 동영상 스트리밍 서비스를 이용하여 실시간으로 방송을 보기 위해서도 필요하며, 재난 관련 긴급상황과 관련된 알림 서비스 등에 이용되기 위해서도 정보제공자 중심의 push-based의 데이터 전송방식이 필요하다<sup>3)</sup>.

CCN은 동일한 정보의 요청에 대한 처리를 줄이기 위해 Pending Interest Table(PIT)을 활용한다. 이는 정보 제공자에게 요청되는 요청메세지의 숫자를 줄여 네트워크의 부하를 낮추는 효과를 갖고 있지만, 실제 존재하지 않는 콘텐츠를 정상적인 prefix와 결합하여 Spoofed interest packet을 만들어 지속적으로 전송하는 경우나 병목현상이 발생하는 집중된 노드에 단기간에 여러 종류의 콘텐츠를 요청하는 경우에 PIT 엔트리가 기하급수적으로 증가하여 오버플로우되는 문제가 발생 할 수 있다. 따라서, 본 논문에서는 CCN에서 Interest Group Pushing Table(IGPT)를 제안하여 통해 신속한 정보 제공을 필요로 하는 상황에서 정보 제공자가 생산하는 정보가 신속하게 전달됨은 물론 PIT 오버 발생 시에도 데이터 포워딩이 가능토록 설계하여 단시간에 많은 정보를 수신할 수 있도록 제안한다.

본 논문의 나머지 부분에 대한 구성은 다음과 같다. 2장에서 관련 연구를 살펴보고, 3장에서는 기존에 제안된 CCN 데이터 전달 방법을 비교하고, 4장에서는 IGPT를 제안한다. 5장에서는 시뮬레이션 환경 및 시뮬레이션 측정결과를 제시하고 분석한다. 마지막으로 6장에서는 향후 연구방향을 제시하며 결론을 맺는다.

## II. 관련 연구

### 2.1 Content-Centric Networking(CCN)

CCN은 현 IP 주소를 이용하여 콘텐츠를 찾는 것과

는 달리 콘텐츠 이름에 기반하여 ‘어디에’가 아니라 ‘무엇을’ 중심으로 콘텐츠를 찾는 방식을 말한다. CCN의 구현을 위해서 두 가지의 패킷타입이 사용된다. 하나는 Interest Packet이며, 다른 하나는 Data Packet이다. Interest Packet은 사용자가 콘텐츠를 요청하기 위해 사용되는 타입이며, Data Packet은 요청된 콘텐츠를 사용자에게 제공하는 패킷으로 콘텐츠 뿐만 아니라 보안 및 신뢰성을 보장하기 위한 Signature 등을 포함하고 있다. 이 두 가지의 패킷은 CCN을 구성하는 노드를 통해 전송된다. CCN 노드는 Content Store(이하 CS), Forwarding Information Base(이하 FIB), Pending Interest Table(이하 PIT)의 구조를 가진다. CS는 노드를 통해 전달되는 콘텐츠를 저장하고 LRU, LFU, FIFO와 같은 교환정책을 적용하고 있다. PIT는 Interest Packet에서 요청하는 콘텐츠가 CS에 없을 경우 다른 노드로 요청을 내보내고 그 리스트를 유지하는 테이블이다. 만약 동일한 콘텐츠를 요구하는 Interest Packet이 수신되는 경우 콘텐츠 이름을 Exact-Matching으로 검색하여 결과가 일치하면 중복요청을 다른 노드로 보내지 않고 해당 패킷은 폐기한다. PIT의 엔트리는 일정시간이 지나거나 요청에 대한 응답이 오는 경우 소멸된다. FIB는 해당 노드에 들어온 Interest Packet에 해당되는 콘텐츠 또는 관련 정보가 CS, PIT에 없으면 다음 노드로 Interest Packet을 전송하기 위한 인터페이스 경로를 저장하는 테이블이다.

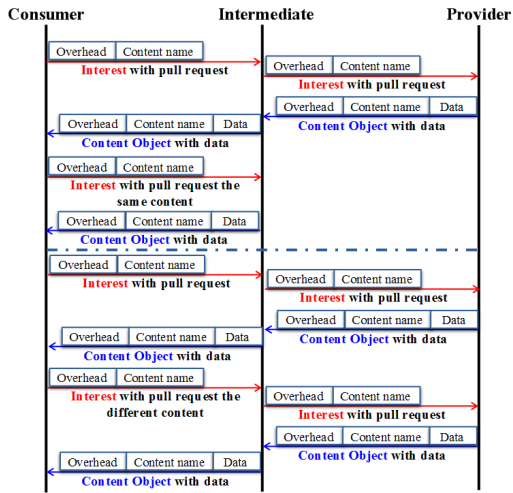
### 2.2 Methods to Data Transport in CCN

CCN에서의 데이터 전송방식은 앞서 살펴본바와 같이 기본적으로 pull-based의 전송방식 따르고 있다. 이를 개선하여 다양한 어플리케이션에 적합하고 효율적인 데이터 전송을 하고자 하는 연구가 활발히 진행되고 있다<sup>4-9)</sup>. 기존의 IP기반의 인터넷 환경에서 Push 기술들의 사용이 용이하였으나, CCN 환경에서는 서버와 클라이언트 관계 성립이 필요없어 Push 방법의 적용에는 제한이 발생한다<sup>10)</sup>. 그러므로 CCN 환경에서 Data Push 기법을 적용하기 위해서는 CCN의 구조를 고려한 새로운 개념의 접근이 필요하다. 상기 사항을 고려하여 CCN에서의 데이터 전송방식이 표 1과 같이 다양하게 연구되고 있다.

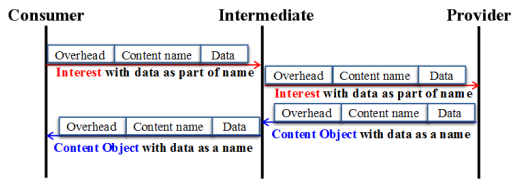
본 논문에서는 표 1의 1번은 EIP, 2번은 LIP, 4번은 4HP, 6번은 PEP로, 또한, 3, 5번의 전송방법이 Publish/Subscribe로 비슷하여 3번은 Simple Pub-Sub, 5번은 Long Pub-Sub으로 구분하였다. 기존 CCN과 표 1의 전송방법을 그림 1과 같이 표현할 수 있다.

표 1. CCN에서 Data 전송방법  
Table 1. Methods of Data transport in CCN

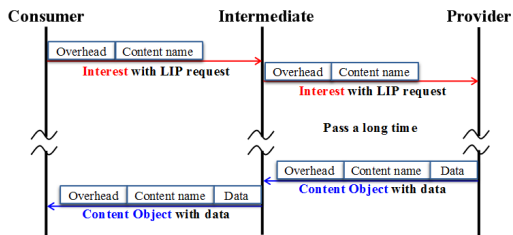
No	Methods	Year
1	Embedded Interest Push <sup>[4]</sup>	2009
2	Long Lived Interest Push <sup>[5]</sup>	2011
3	Content-Based Publish/Subscribe <sup>[6]</sup>	2011
4	4-Way Handshake Push <sup>[7]</sup>	2012
5	Optimized by design for IoT <sup>[8]</sup>	2013
6	Push Enabling Packets <sup>[9]</sup>	2013



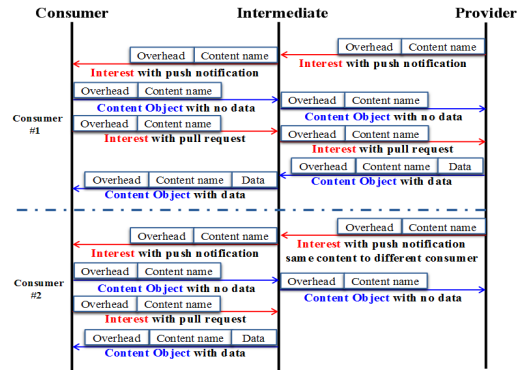
(a) Pull 방법(기본 CCN)



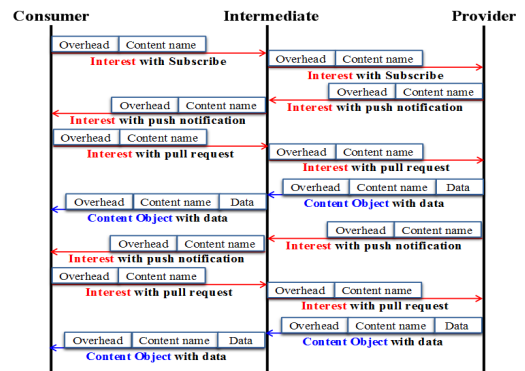
(b) EIP(Embedded Interest Push) 방법



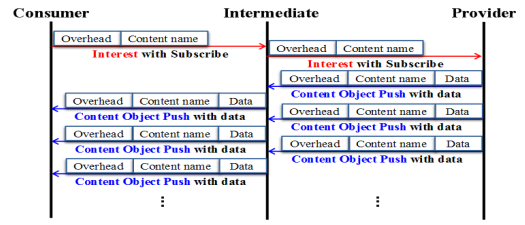
(c) LIP(Long Lived Interest Push) 방법



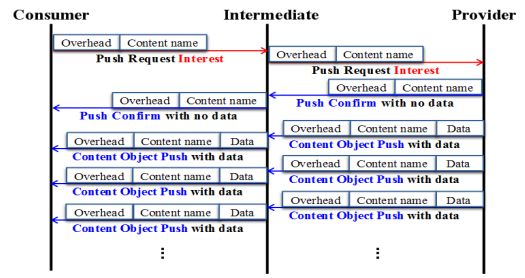
(d) 4HP(4-Way Handshake Push) 방법



(e) Simple Pub-Sub(Content-Based Publish/Subscribe) 방법



(f) Long Pub-Sub(Optimized by design for IoT) 방법



(g) PEP(Push Enabling Packets) 방법

그림 1. CCN에서 Interest packet과 content object packet의 전송방법 비교  
Fig. 1. Comparison of the delivery methods between Interest and content object

기본 CCN의 동작방식<sup>[11]</sup>을 살펴보면 그림 1(a)와 같이 요청자가 기본적으로 하나의 콘텐츠를 요청한 다음 동일한 콘텐츠를 재요청하는 경우와 요청자가 하나의 콘텐츠를 요청한 다음 다른 콘텐츠를 요청하는 경우로 크게 2가지로 구분 할 수 있다. 첫 번째와 두 번째 경우에서 보듯이 기본 CCN은 최초로 Interest packet을 전달하여 콘텐츠를 요청함에 따른 응답으로 Data packet이 전송되는 Pull 방식을 따르고 있으며 Push 방식에 대해서는 고려가 되어 있지 않음을 알 수 있다. EIP 동작방식은 그림 1(b)와 같이 Interest packet에 데이터를 포함하여 전송하는 특징이 있으며, LIP 동작방식은 그림 1(c)와 같이 기존의 Interest packet에 Lifetime을 조정하여 전송하는 특징이 있다. 4HP 동작방식은 그림 1(d)와 같이 요청자가 Interest packet을 먼저 보내는 것이 아니라 정보제공자가 Interest packet을 먼저 보내서 Push 정보를 알리는 방식을 취하고 있으며, Simple Pub-Sub 동작방식은 그림 1(e)와 같이 Subscribe 동작과 Publish 동작으로 구분되어 진행되며 IP 주소체계처럼 요청자의 위치를 바탕으로 Push가 필요한 데이터가 있을 경우 쉽게 전파가 가능하다는 특징이 있으며, Long Pub-Sub 동작방식은 그림 1(f)와 같이 하나의 Interest packet의 Subscribe 정보를 바탕으로 다수의 Data packet을 보낼 수 있도록 되어 있다. 마지막으로 PEP 동작방식은 그림 1(g)와 같이 표현할 수 있으며 Long Pub-Sub 방식과 그 동작방식이 유사하며 맨 처음 Interest packet에 따라 데이터를 포함하지 않은 응답 packet을 보낸다는 특징을 가지고 있다. 이처럼 다양한 데이터 전송 방법들을 활용해 어플리케이션이 요구하는 다양한 사항(실시간 스트리밍 등)을 만족할 수 있을 것이다.

### 2.3 Methods to mitigate PIT overload

CCN에서는 서론에서 언급한 PIT 오버로드 등과 같은 취약한 점이 있다. CCN에서의 PIT 오버로드와 관련하여 표 2와 같이 다양한 연구가 진행되고 있다.

본 논문에서는 표 2의 1번은 DiPIT, 2번은 SDP, 3번은 MFB로 구분하였다. 기존 CCN과 표 2의 전송방법을 그림 2와 같이 표현할 수 있다.

표 2. CCN에서 PIT overload 감소 방법  
Table 2. Methods to mitigate PIT overload in CCN

No	Methods	Year
1	DiPIT <sup>[10]</sup>	2012
2	Spoofed Data Packet <sup>[11]</sup>	2013
3	Maintaining Flow Balance <sup>[12]</sup>	2013

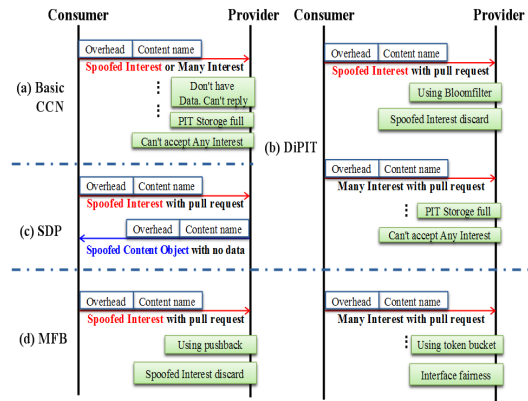


그림 2. CCN에서 PIT Overload의 해결 방법  
Fig. 2. Methods to solve the PIT overload problem

그림 2에서처럼 Spoofed interest packet이나 단기간에 다수의 콘텐츠가 많이 요청되는 경우 PIT 공간을 다 사용하게 되어 그 이후 요청되는 Interest packet을 버리게 되고, 오랫동안 응답이 없다고 판단한 정보 요청자는 또다시 재요청을 반복하게 되는 상황이 지속될 여지가 있다. 이는 네트워크 효율을 저하시키게 되어 정상적인 정보요청자가 원하는 데이터를 받는데 드는 시간이 증가하게 되는 결과를 가져올 수 있다. DiPIT, SDP, MFB 등을 활용하면 이러한 문제점을 어느 정도 해결할 수 있다. 하지만 PIT overload 상황 발생에 따른 추가 Interest packet을 받아들일 수 없는 경우가 발생한다. 이에 따라 본 논문에서는 PIT overload 상황이 발생하여도 지속적으로 요청되는 Interest packet을 수용하여 재요청 전송시간을 단축시킴으로써 정상적인 요청자의 입장에서 좀 더 빠르게 원하는 콘텐츠를 받을 수 있도록 하는 방법을 연구하였다.

## III. CCN 데이터 전송 기법 분석

### 3.1 분석환경

CCN의 다양한 데이터 전송방법들을 상호 비교하고 그 중 하나의 방식을 이용하여 IGPT를 설계하고자 한다. 기존 CCN과 비교하여 동일 시간, 동일 환경에서 더 많은 콘텐츠를 받기 위한 IGPT 사용 목적을 고려하여 각 전송방법들의 대역폭 소모율을 분석하여 가장 낮은 소모율을 보이는 전송방법을 확인하였다. 대역폭 소비율은 정보 전송방법 중 Pull, LIP, 4HP, EIP를 대상으로 연구된 선행연구<sup>[13]</sup>를 발전시켜 Simple Pub-Sub, Long Pub-Sub, PEP를 추가하여 분

석하였다. 먼저, 그림 3과 같이 Interest packet과 Data packet을 name과 overhead로 분리하여 평균 크기를 정의하였다. 또한, 별도의 P<sub>1</sub>, P<sub>co</sub> overhead를 추가하였다. 해당 overhead는 push 구현에만 사용된다.

그림 4와 같이 요청자와 제공자 사이에 중간 노드가 있고, 한쪽은 m개의 요청자와 반대쪽은 1개의 제공자를 각각 위치시키고 해당 노드를 통하여 데이터 전송이 된다고 가정하였다. 이 때 링크의 대역폭은 충분히 모든 데이터의 수용이 가능하며, 콘텐츠를 요청 또는 전파할 때 중간 노드에 해당 콘텐츠가 캐싱되어 있을 확률, 즉 Cache hit ratio를 D로 정한다. Pull, LIP, 4HP 방법에서는 확률 D가 사용되는 반면에 EIP, Simple Pub-Sub, Long Pub-Sub, PEP 방법은 중단과 중단사이에 데이터 전송에 중점을 두고 있으므로 확률 D가 사용되지 않는다.

Pull와 LIP 방법의 경우 중간 노드에 해당 콘텐츠가 캐싱되어 있지 않다고 가정하면 그림 4 #1, #2 모두 패킷이 전송되며 한번 요청에 2개의 Interest packet과 2개의 Data packet이 전송된다. 중간 노드에 해당 콘텐츠가 캐싱되어 있다면 그림 4 #1에만 패킷이 전송되어 각 1개씩의 Interest packet과 Data packet이 전송된다. EIP 방법은 확률 D가 사용되지 않고 제공자와 요청자 사이에 지속적으로 패킷이 전달된다. 4HP 방법의 경우 중간 노드에 해당 콘텐츠가 캐싱되어 있지 않다고 가정하면 그림 4 #1, #2 모두 패킷이 전송되며 한번의 콘텐츠 전달에 push

notification 정보가 포함된 2개의 Interest packet, 2개의 일반적인 Interest packet, data를 포함하지 않고 push reply 정보를 포함한 2개의 Data packet, data를 포함한 2개의 Data packet이 전송된다. 하지만 다른 요청자에게 동일한 콘텐츠를 전송하고자 할 때 이미 중간 노드에 전파하고자 하는 콘텐츠가 캐싱되어 있다면 그림 4 #2에서는 최초 push notification 역할의 Interest packet 및 그에 따라 data를 포함하지 않은 Data packet만 전송되며, 그림 4 #1에서만 data 전달이 이루어지게 된다. Simple Pub-Sub, Long Pub-Sub, PEP 방법의 경우 push 기법 구현을 위해 Interest packet 및 Data packet에 push request/reply overhead가 추가되어 동작된다. 먼저 Simple Pub-Sub 방법에서 요청자는 제공자에게 콘텐츠 구독 신청을 한다. 이 때 구독정보가 포함된 2개의 Interest packet이 발생한다. 해당 구독정보는 최초 1회만 필요하다. 제공자는 보내야 할 콘텐츠가 생성되면 요청자에게 해당 콘텐츠를 요청할 것을 알리게 되고 push notification 정보가 포함된 2개의 Interest packet이 발생한다. 요청자는 제공자에게 해당 콘텐츠를 요청하여 data를 수신하게 되며 일반적인 Interest packet 2개, Data packet 2개가 전송된다. Long Pub-Sub 방법의 경우 요청자는 제공자에게 구독신청을 하며 이 때 구독정보가 포함된 2개의 Interest packet이 발생한다. 해당 구독정보는 최초 1회만 필요하다. 제공자는 보내야할 콘텐츠가 생성되면 바로 구독요청을 한 요청자에게 data를 보내고 push reply overhead가 포함된 2개의 Data packet이 전송된다. PEP 방법은 Long Pub-Sub 방법과 거의 유사하다. 단지 최초 구독요청시 제공자가 요청자에게

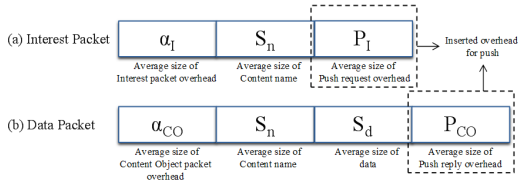


그림 3. Interest packet와 Data packet의 평균 size  
Fig. 3. Average size of Interest packet & Data packet

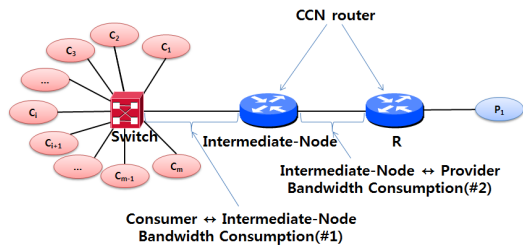


그림 4. 대역폭 소모율 측정을 위한 토폴로지  
Fig. 4. Topology for bandwidth Consumption Rate determination

표 3. consumer와 provider 간 대역폭 소모율  
Table 3. Bandwidth Consumption method between consumer and provider

Methods	Mathematical bandwidth consumption model
Pull and LIP	$m\{D \times (2S_n + S_d + \alpha_1 + \alpha_{co}) + (1 - D) \times 2(2S_n + S_d + \alpha_1 + \alpha_{co})\}$
EIP	$m \times 2(2S_n + 2S_d + \alpha_1 + \alpha_{co})$
4HP	$m\{D \times (6S_n + S_d + 3\alpha_1 + 3\alpha_{co} + 2P_1 + 2P_{co}) + (1 - D) \times 2(4S_n + S_d + 2\alpha_1 + 2\alpha_{co} + P_1 + P_{co})\}$
Simple Pub-Sub	$2(4S_n + S_d + 3\alpha_1 + \alpha_{co} + 2P_1) + (m - 1) \times 2(3S_n + S_d + 2\alpha_1 + \alpha_{co} + P_1)$
Long Pub-Sub	$2(2S_n + S_d + \alpha_1 + \alpha_{co} + P_1 + P_{co}) + (m - 1) \times 2(S_n + S_d + \alpha_{co} + P_{co})$
PEP	$2(3S_n + S_d + \alpha_1 + 2\alpha_{co} + P_1 + 2P_{co}) + (m - 1) \times 2(S_n + S_d + \alpha_{co} + P_{co})$

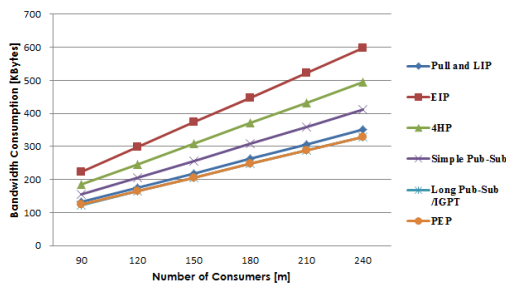
구독완료가 되었다고 응답을 주는 것이 추가되어 있다.

각 전송방법에 따른 요청자와 제공자 사이의 대역폭 소모율의 식을 종합하면 표 3과 같다.

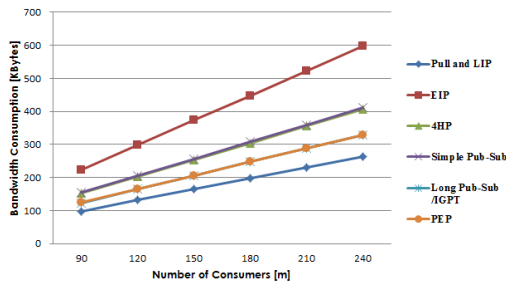
### 3.2 분석결과

대역폭 소모율 측정을 위해  $S_n = 70$ ,  $S_d = 512$ ,  $\alpha_i = 27$ ,  $\alpha_{co} = 52$  으로 값을 설정하였다<sup>[13]</sup>. 또한, Push 구현을 위해 추가된 오버헤드의 값은 기존의 Interest packet과 Data packet의 값과 동일하게  $P_i = 27$ ,  $P_{co} = 52$  으로 값을 설정하였다. 그 이유는 Push 구현을 위한 추가 오버헤드의 값은 각 data 전송방법에 따라 차이가 있기 때문이다. 따라서 전송방법간의 비교판단

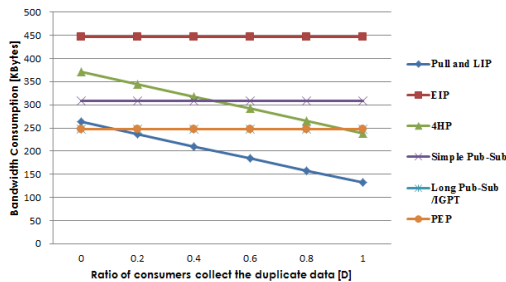
을 위해서 동일한 값으로 설정하였다. 그림 5(a)에서는 D를 0으로, 그림 5(b)에서는 D를 0.5로 설정하고 M값을 90에서 240까지 변화시켜 결과값을 확인하였다. 그림 8(c)는 M을 180으로 고정하고, D를 0에서부터 1까지 변화되었을 때의 결과를 그래프로 나타내었다. Long Pub-Sub, PEP 방식은 해당 값이 거의 동일하였으며, 그림 5(a)에서처럼 D값이 0인 경우, 즉 정보요청자가 동일한 콘텐츠를 요청하지 않고 계속해서 다른 콘텐츠를 요청하는 경우에 가장 작은 대역폭 소모를 하는 것을 알 수 있다. 그림 5(b)는 요청자가 동일한 콘텐츠를 요청할 확률이 50%인 경우 Pull, LIP 방식이 가장 낮은 대역폭 소모를 하는 것을 보여준다. 그림 5(c)에서는 D값이 증가함에 따라 확률을 사용하는 Pull, LIP, 4HP 방식은 점차 대역폭 소모율이 낮아지는 반면 이를 제외한 방식들은 대역폭 소모가 일정함을 알 수 있다. 따라서 그림 5 결과에 따라 동일한 콘텐츠를 다수의 노드가 가지고 이를 주변에서 요청하는 경우 Pull, LIP, 4HP 방식이 낮은 대역폭을 소모하며, 요청자와 제공자간에 새로운 데이터의 전송(예를 들면 스트리밍 서비스 등)과 같은 상황에서는 Long Pub-Sub, PEP 방식이 낮은 대역폭 소모를 가진다는 것을 알 수 있다. 데이터 전송의 신속성 관점에서 보면 Long Pub-sub 방식(그림 1(f))이 다른 방식과 비교하여 단일 시간에 좀 더 많은 데이터를 받을 수 있다는 것을 알 수 있다. 따라서, 본 논문에서는 Long Pub-sub 방식을 이용한 IGPT의 사용을 통해 push-based 데이터 통신 및 PIT 오버로드시 데이터 통신을 구현하고자 한다.



(a) 전송기법별 대역폭 소모량(D=0, m=90~240)



(b) 전송기법별 대역폭 소모량(D=0.5, m=90~240)



(c) 전송기법별 대역폭 소모량(D=0~1, m=180)

그림 5. Consumer의 data 요청 횟수 및 동일 data 요청 비율 대비 Bandwidth consumption 결과

Fig. 5. Bandwidth consumption(y) versus number of data consumers/ratio of consumers collect the same data(x)

## IV. Interest Group Pushing Table (IGPT)

Long Pub-sub 방식의 구현을 위해서는 우선 요청자는 제공자에게 데이터를 받고 싶다는 요청을 해야 한다. 이를 통해 요청을 받은 제공자는 새로 생성하는 콘텐츠를 요청자에게 지속적으로 보낼 수 있게 된다. 이를 고려하여 기존 패킷에 pull-based와 push-based를 구분하는 옵션값을 추가하였다. Interest packet에는 00/01/10/11의 값을 추가하고, Data packet에는 0/1값을 추가하였다. Interest packet에 추가한 옵션인 00은 기존의 CCN과 동일하게 pull-based로 동작하도록 하며, 01/10/11은 push-based로 동작한다. 기존의 Interest packet과 동일하게 IGPT에서도 lifetime을 지정하여 요청할 수 있으나, Interest packet에 해당하는 Data packet이 도착하여도 계속적인 data push를 위해 lifetime 동안 계속해서 IGPT에 남아 있게 되므로,



신규/갱신/삭제 등의 상황을 조절하기 위한 옵션으로 01/10/11을 설정하였다. Data packet에 추가한 옵션 0은 기존과 동일한 pull-based의 콘텐츠로, 옵션 1은 push-based의 콘텐츠로 동작하도록 하였다. 이에 따라 pull-based 및 push-based의 데이터 전송방식 모두를 적용할 수 있으며, push-based의 데이터 전송을 원하는 경우 우선 01의 옵션값을 가지고 있는 Interest packet을 요청자가 전송하고, 이에 따라 제공자는 본인이 push하려는 콘텐츠에 1의 옵션값을 부여하고 IGPT를 통해 데이터 push가 이루어지게 된다. 추가된 옵션값은 표 4와 같다.

기본적으로 CCN에서는 라우터가 Data Packet을 수신하면 CS, PIT 순으로 참조하여 패킷을 처리한다. 표 4의 CCN Packet 옵션값에 따라 Data Packet이 Push Contents인 경우 별도의 테이블을 참고한다면 즉시 다음 노드로 해당 Data Packet이 전송되어 신속한 정보 전파가 가능하다. 따라서, 요청자가 Push를 요청하는 경우 별도의 테이블에 저장되어 push-based의 데이터 통신이 가능하도록 설계하였다. 이 때 다수의 사용자가 소수의 제공자에게 data를 요청하는 경우 동일한 Contents Prefix는 해당 인터페이스만 추가하도록 하고 Push Data가 생성되어 전송되면 해당 콘텐츠는 인터페이스를 참고하여 해당 인터페이스로 동시에 전파되므로 멀티캐스트형식으로서의 전송도 가능하다. 이 별도의 테이블을 ‘Interest Group Pushing Table (IGPT)’이라 정의하였다. IGPT의 구성은 표 5와 같다.

IGPT는 인터페이스 정보 이외에도 유효시간 정보를 가지고 있고 카운트 필드를 가지고 있다는 특징이 있다. 현재의 인터넷 환경이 다양한 네트워크 구조를 가지고 있고 이에 따라 요청자와 제공자 사이에 네트워크 구조가 바뀔 가능성을 고려하여 유효시간 정보를 갱신토록 하였다. 유효시간이 겹치게 되는 경우 해

표 4. Push-based 전달을 위한 패킷 추가 옵션  
Table 4. Additional options of packets for push-based delivery

Packet types	Option	Description	
Interest Packet	00	Pull Interest	
	01	Push Interest	Insert IGPT entry
	10		Refresh the IGPT entry time
	11		Delete the IGPT entry
Data Packet	0	Contents to be pulled	
	1	Contents to be pushed	

표 5. Interest Group Pushing Table(예시)  
Table 5. Interest Group Pushing Table example

Contents Prefix	Interface	Lifetime	Count
CEO.streaming.com	1	12h	-
CEO.streaming.com/11DEPT	0, 1	8h	60
CEO.streaming.com/11DEPT/ 111UNIT/1SXN	1	-	30

당 노드에 자료 요청이 많다고 판단되어 최대요청시간이 반영되도록 하였다. 예를 들면 최초의 요청자가 제공자에게 30분간의 push 요청하였는데, 중간노드에 연결된 다른 요청자가 동일한 제공자에게 1시간의 push를 요청하는 경우 Lifetime이 30분에서 1시간으로 변경된다. 이때 요청되는 시간은 각 어플리케이션의 특징, 토폴로지, 중요도에 따라 다르게 설정될 수 있다. 기본적으로 패킷이 수신되면 Data Packet인지 Interest Packet인지에 따라 동작되며, 여기에 pull-based와 push-based 통신을 각각 구분하여 작동되는 알고리즘을 추가하였으며 Push Contents인 경우에는 CS, PIT, FIB를 거치지 않고 IGPT만 참고하여 바로 전파가 가능토록 구성하였다. 또한, 카운트에는 PIT 오버로드 상황에서 PIT가 가득차서 더 이상 요청되는 Interest packet을 처리하지 못하게 된다면 IGPT를 통해 정보를 입력 가능토록 구성하였으며 기존의 PIT와는 다르게 Prefix 단위로 Interest Packet을 카운트하도록 하여 저장공간에 대한 부담을 줄이면서 하나의 테이블을 가지고 push-based의 데이터 통신 및 PIT 오버로드시 모두 참조토록 설계하였다. 정상적인 PIT 오버로드 상황과 더불어 DDoS의 공격과 같이 공격자의 변조된 임시 Interest Packet들에 따른 PIT 오버로드 상황으로 PIT 저장공간이 더 이상 없거나 사용이 불가능한 경우 요청된 Interest Packet을 버리는 것이 아니라 IGPT에 추가하고, 이 때 count를 1 증가시킨다. 그리고 다시 전달된 Interest Packet은 동일한 절차를 거치게 된다. 마찬가지로 이에 대응하는 Data Packet이 도착하면 카운트를 1 감소시켜 카운트가 0이 되면 해당 entry는 삭제된다. 이 때 비정상적인 Interest Packet이라 할지라도 IGPT에 저장되는 상황이 발생하는데 그렇다면더라도 prefix 단위로 count만 증가시키게 되므로 메모리 공간의 추가 사용은 크지 않다. PIT overload 상황에서 적용되는 Interest/Data packet은 pull option을 보유한 packet에 경우에 한한다. IGPT에서 Time과 Count가 동시에 존재하는 경우에 count 값이 0이 되어도 push를 지원하기 위해 존재하는 해당 entry는 삭제되지 않는다. 이를 종합해 보

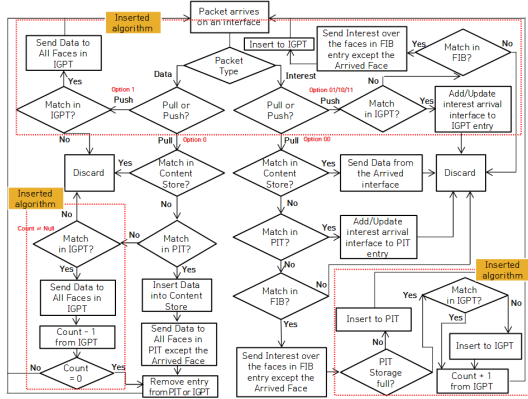


그림 6. 수정된 CCN Packet 흐름도  
Fig. 6. Modified CCN Packet Flow Chart

면 Push interest와 data의 경우에는 IGPT의 Lifetime을 활용하고, Pull interest와 data의 경우에만 IGPT의 count를 활용한다. push의 구현을 위해 lifetime 동안만 테이블 entry에 존재하며, PIT overload 상황을 위해 count를 활용하고 Pull interest를 그룹화하여 관리하기 때문이다.

Lifetime과 count가 둘 다 존재하는 경우 둘 다 모두 null값이 되어야 해당 entry가 삭제된다. count만 있는 경우 계속 잔존하는 문제 해결을 위해 기존 PIT 처럼 동일하게 해당 Interest packet의 lifetime 시간이 지나면 삭제가 되며, 삭제 이전에 새로운 interest가 도착하여 count가 증가하면 해당 entry의 lifetime이 갱신된다. IGPT를 적용한 수정된 CCN Packet Flow Chart는 그림 6과 같다.

V. 실험 및 분석

5.1 실험환경

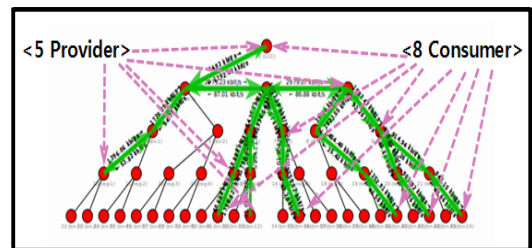
앞서 CCN환경에서 push-based의 데이터 통신 기능 적용 및 PIT 오버로드 상황에서 data 전송이 가능토록 IGPT를 설계하였고 이를 실험을 통해 효율성을 검증하고자 한다. 기존의 CCN과 IGPT가 적용된 CCN의 콘텐츠 수신량을 비교 확인하였고, PIT 오버로드 상황에서의 data 전송 확인을 위해 기존 CCN 상황에 spoofed된 다량의 Interest packet을 전송하여 PIT 저장공간이 가득차도록 하여 IGPT 적용 전/후를 비교하였다. 이상의 실험환경 변수를 정리하면 표 6과 같으며, 시나리오를 기준으로 실험 1, 실험 2로 구분하고 ndnSIM<sup>[14]</sup>을 이용하여 실험하였다.

실험 1에서는 IGPT를 적용하기 전과 후를 비교하여 초당 콘텐츠를 얼마나 받는지 확인하기 위하여 그

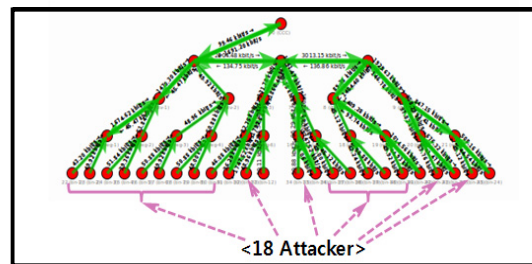
표 6. 실험환경 변수  
Table 6. Simulation Parameters

Simulation scenario	Number of		Contents request rate(sec)	Payload
	Node (Router)	Attacker		
1 Contents Received per second	46	.	100 to 1,000	1,024
2 Contents Received of PIT overload	46	18	100 to 1,000	1,024

림 7과 같이 46개의 노드를 계층적인 구조로 만들고 8개의 요청자가 5개의 제공자로부터 데이터를 요청하도록 설계하였다. IGPT의 적용을 위해서 4장에서 추가한 패킷 추가 옵션을 사용하였으며 pull 방식의 경우는 기존과 동일하므로 push 옵션이 추가된 Interest packet 및 Data packet이 사용된다고 가정하였다. 또한 정보 제공자 중심으로 정보를 신속하게 전달하는 목적의 Push 방식이 적용된 IGPT와 기존 CCN의 CCN의 상호 비교를 위해서 정보 제공자는 스트리밍 서비스 등과 같이 지속적으로 새로운 콘텐츠를 생성하여 배포하고자 하고 이에 따라 각 요청자는 지속적으로 매번 다른 이름을 가지는 콘텐츠를 요청한다고 가정하였다. 시뮬레이션 시간을 기준으로 변화량을 측정하여 분석가능토록 실험하였다. 실험에서 링크 대역



(a) 기본 토폴로지 / PIT attack(or overload)이 없는 상황



(b) PIT overload 상황의 토폴로지 / PIT attack이 있는 상황

그림 7. 46개 노드를 계층적으로 배치한 토폴로지  
Fig. 7. Tree topology that is consisted of 46 nodes



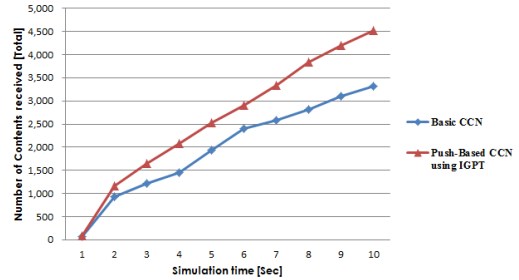
폭은 2Mbps, delay는 100ms, queue는 100으로 설정하였으며 Interest packet은 초당 100번에서부터 1,000번을 요청토록 하였다. 시뮬레이션 시간은 1초부터 10초까지 변화를 확인하였다. 구성된 토폴로지는 그림 7(a)와 같다.

실험 2을 위해 그림 7과 같이 실험 1와 동일한 토폴로지를 이용하였으며, 그림 7(b)에서 보는 것처럼 중간 edge router들에 강제적으로 PIT overload 상황을 부여하기 위해 특정시간 동안 18개의 공격자들이 spoofed된 Interest packet을 최상단 정보제공자에게 요청토록 하였다. 공격자들은 10초 동안 임의의 interest packet을 보내게 하였고 PIT에 존재하는 Interest packet의 lifetime은 ndnSIM의 기본 설정값을 기준으로 2초로 하였다. 시뮬레이션 시간은 20초까지 설정하여 변화를 확인하였다.

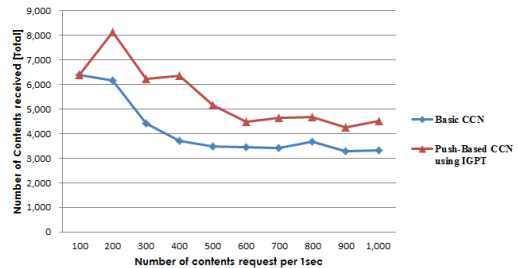
## 5.2 실험결과 및 분석

### 5.2.1 실험 1. 기존 CCN과 IGPT가 적용된 CCN의 콘텐츠 수신량

그림 8(a)에서 CCN에 IGPT를 적용하기 전·후에 초당 전송되는 콘텐츠의 양을 확인할 수 있다. 결과에서 처럼 시간이 지남에 따라 기존 CCN과 비교하여 IGPT가 적용된 CCN의 경우 최초에는 기존 CCN과 비슷한 양의 콘텐츠를 수신하지만 시간이 지날수록 제공자가 생산하는 콘텐츠를 push-based의 데이터 통신 기법을 통하여 생산 즉시 요청자에게 전달되게 되므로 동일한 시간에 더 많은 신규 콘텐츠를 요청자가 수신할 수 있다는 것을 알 수 있다. 그림 8(b)에서는 기존 CCN의 경우 초당 콘텐츠 요청이 100개에서 200개의 경우는 실험환경의 대역폭인 2Mbps이내에 모두 동작하여 변화가 적은 반면 초당 콘텐츠 요청이 300개 이상부터는 대역폭 범위를 넘어서게 되어 Interest packet의 재전송 요청 증가 등에 따라 실제 전송되는 콘텐츠 수가 줄어들게 된다. IGPT의 경우 초당 콘텐츠 요청이 100개인 경우는 기존 CCN과 동일하나 초당 콘텐츠 요청이 200개인 경우는 실험환경의 대역폭인 2Mbps이내에서 최적의 전송률을 보이며 초당 콘텐츠 요청이 300개 이상부터는 전반적으로 기존 CCN보다 IGPT가 더 많은 콘텐츠를 수신하는 것을 알 수 있다. 본 실험에서 기존 CCN과 동일하게 IGPT의 경우에도 지속적으로 Push interest packet을 요청하여 초당 콘텐츠 요청 건수가 증가함에 따라 Interest packet의 수도 같이 증가하게 하였다. 상기 결과를 통해 동일하게 콘텐츠를 요청하는 경우 동일시간에 기



(a) 시간 변화에 따른 전체 노드의 콘텐츠 수신량



(b) 콘텐츠 요청 패킷 변화에 따른 전체 노드의 콘텐츠 수신량

그림 8. 시뮬레이션 시간 및 콘텐츠 요청에 따라 모든 요청자가 받는 콘텐츠 수  
Fig. 8. All consumer of contents received (y) versus simulation time and contents request (x)

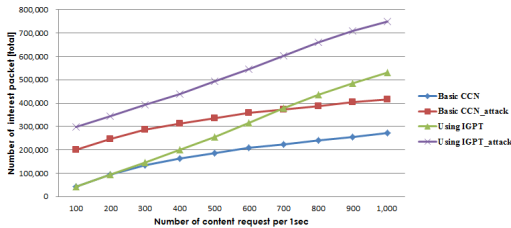
존 CCN 보다 IGPT가 좀 더 많은 콘텐츠를 수신할 수 있다는 것을 알 수 있다. 하지만 IGPT의 경우 매번 새로운 콘텐츠를 요청할 필요가 없이 Lifetime을 증가시킨 Push 요청을 보낸다면 추가의 Interest packet이 없이도 콘텐츠의 전송이 가능하므로 동일시간에 좀 더 많은 콘텐츠의 수신에 가능할 것이다.

### 5.2.2 실험 2. PIT 오버로드 상황에서의 콘텐츠 수신량

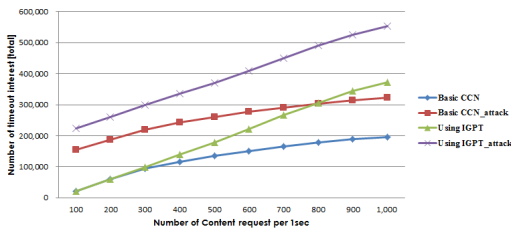
그림 9(a)에서 기존 CCN의 경우 초당 요청하는 콘텐츠 수가 증가함에 따라 PIT 오버로드 상황에 도달하면 요청자에서부터 제공자까지 중간에 위치하는 노드 라우터들이 추가로 요청되는 Interest packet을 drop시켜 일정수준을 유지하게 되는 것을 알 수 있으며, IGPT를 사용하는 CCN의 경우 모든 요청되는 Interest packet을 처리하므로 Interest packet 수가 지속적으로 증가하는 것을 알 수 있다. 그림 9(a)와 그림 9(b)를 비교해 보면 먼저 초당 700개의 콘텐츠 요청시에 공격을 받은 기존 CCN과 IGPT의 Interest packet의 수가 비슷하고 초당 800개의 콘텐츠 요청시에 공격을 받은 기존 CCN과 IGPT의 Timeout interest

packet의 수가 비슷함을 알 수 있다. 이는 초당 요청하는 콘텐츠 수가 증가하여 모든 Interest packet이 처리되지 못하여 발생하는 Timeout interest packet은 약간의 차이를 두고 그림 9(a)와 마찬가지로의 흐름으로 증가하는 것을 보여준다. 그림 9(c)에서 throughput의 효율은 basic CCN > basic CCN\_attack > Using IGPT > Using IGPT\_attack 순임을 알 수 있다. IGPT의 경우 기존 CCN과 비교하여 동일 시간에 더 많은 Interest packet을 받고, 이를 처리하여 더 많은 Data packet을 보내기 때문에 초당 요청되는 콘텐츠 수가

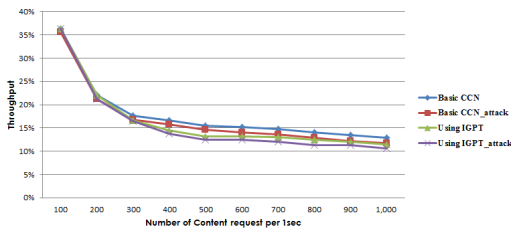
증가함에 따라 throughput 효율이 낮아지게 된다. 이는 Interest packet의 경우 Data packet과 비교하여 크기가 작아 더 많이 전송이 가능한 반면, Data packet은 대역폭 한계에 따라 일정 수치가 넘어가면 병목구간이 발생하여 전송이 일부 지연되므로 이에 따라 효율이 낮아지게 된다. 하지만 그림 9(d)에서 보듯이 기존 CCN의 경우 초당 요청하는 콘텐츠 수가 증가하여도 해당 Interest packet을 PIT에서 처리를 하지 못하여 수신되는 콘텐츠가 일정수준을 유지하게 되는 반면에 IGPT를 사용하는 CCN의 경우 초당 요청하는 콘텐츠 수 증가에 따른 Interest packet을 모두 처리함으로써 더 많은 콘텐츠를 받을 수 있게 된다. 그림 9(d)를 통해서 기존 CCN 및 IGPT를 적용한 CCN 둘다 Spoofed interest packet 공격 전·후가 크게 차이 없이 동작되며, 특히 IGPT를 적용한 CCN이 기존 CCN 보다 PIT 오버로드 상황에서 더 많은 콘텐츠를 수신하는 것을 알 수 있다.



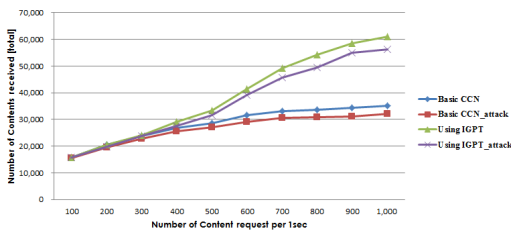
(a) Interest packet 발생량



(b) Timeout Interest packet 발생량



(c) Throughput 비교 결과



(d) 콘텐츠를 포함한 Data packet 수신량

그림 9. PIT overload 상황에서 패킷량 분석 결과  
Fig. 9. Result of Packets received of PIT overload

## VI. 결론 및 향후 연구

본 논문에서는 기존 CCN의 Interest Packet과 Data Packet에 옵션값을 추가하고 IGPT 설계하여 생성된 Push Interest Packet에 따라 IGPT에 해당 값이 입력되고, 제공자가 생산하는 Push Contents는 생산 즉시 CS, PIT, FIB를 거치지 않고 IGPT만 참고하여 전송되게 하는 push-based의 데이터 통신이 가능하도록 제안하였다. 이를 통해 push-based 동작이 필요한 어플리케이션 등에서의 활용에 따른 신속성 향상이 기대된다. 또한, push-based가 아닌 경우 pull-based가 동작되도록 설계하여 기존 CCN과 동일하게 Packet처리가 가능하도록 하여 활용도를 높였다. 뿐만 아니라 IGPT를 통해 단기간에 다량의 Interest packet이 요청되는 경우 이를 수용하고 처리하도록 하여 PIT 오버로드 상황에서의 콘텐츠 수신량 증가가 가능토록 제안하였다. 이를 통해 Spoofed interest packet attack 등의 상황에서도 더 많은 데이터를 수신하도록 보안성을 향상시켜 유연하게 대처가 가능하다. 또한, CCN에서는 IP 주소가 없이 콘텐츠 중심으로 전송이 일어나며, Pull-based 기반으로 동작됨에 따라 기존 IP 네트워크에서 멀티캐스트 기법 등의 적용이 제한된다. IGPT를 통하여 1hop 단위의 멀티캐스트 그룹 관리 역할을 할 수 있으며 실제적으로 Subscribe를 신청한 요청자들에게 콘텐츠의 전파가 가능하여 CCN에 기존 IP 네트워크의 멀티캐스트 기법 적용이 가능토록 한 것이라 할 수 있다. 이와 더불어 Lifetime 적용으로 지

정한 시간 동안만 콘텐츠를 수신토록 조정가능한 장점이 있다. 이처럼 CCN에서 IGPT를 활용한다면 push-based의 데이터 통신 및 PIT 오버로드 상황에서 기존 CCN 대비 동일 시간에 더 많은 데이터의 수신 이 가능할 것으로 기대된다. 하지만, IGPT 적용을 위해서는 옵션값을 추가한 패킷 설정, 처리 및 Table 관리 등의 추가사항이 있으며 Prefix Match를 통해 데이터가 전송되므로 제공자가 생성하는 Push Data가 무제한적으로 요청 Consumer로 전달되는 과정에서 발생하는 네트워크 효율성 부분의 문제 등이 발생할 수 있다. 또한, 단말의 이동성, Congestion control, PIT와 IGPT 메모리 할당 등에 대한 고민도 필요하다. 향후 이러한 고려사항을 포함하여 확장된 topology를 구성하여 콘텐츠 사이즈 및 Segment 변화에 따른 판단 등을 데이터 전송방법별로 추가 실험할 예정이다. 이를 통해 최적화된 CCN 데이터 전송 기법 및 선별적 data push 방법 도출이 기대된다. 또한, 유·무선의 다양한 환경에서의 CCN 데이터 전송을 위해 간접적 push-based<sup>[16]</sup> 등의 기법들을 접목한 전송방식에 대해 연구 예정이다.

## References

- [1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proc. CoNext'09*, pp. 1-12, 2009.
- [2] J. P. Martin-Flatin, "Push vs. Pull in web-based network management," in *Proc. Sixth IFIP/IEEE Int. Symp.*, pp. 3-18, Boston, MA, 1999.
- [3] E. Bozdag and A. Mesbah, "A comparison of push and pull techniques for Ajax," in *Proc. 9th IEEE Int. Workshop WSE 2007*, pp. 15-22, 2007.
- [4] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, J. D. Thornton, and R. L. Braynard, "VoCCN: Voice-over content-centric networks," in *Proc. ACM Int. Workshop Re-Architecting Internet (ReArch '09)*, pp. 1 - 6, Rome, Italy, Dec. 2009.
- [5] Z. Zhu, S. Wang, X. Yang, V. Jacobson, and L. Zhang, "ACT: Audio conference tool over named data networking," in *Proc. ACM SIGCOMM Workshop Inf.-Centric Netw. (ICN '11)*, pp. 68-73, Toronto, Canada, Aug. 2011.
- [6] A. Carzaniga, M. Papalini, and A. L. Wolf, "Content-based publish/subscribe networking and information-centric networking," in *Proc. ACM SIGCOMM Workshop Inf.-Centric Netw., ACM*, pp. 56-61, 2011.
- [7] J. Kim, M. Jang, Y. Bae, and B.-J. Lee, "Named content sharing in virtual private community," in *Proc. 9th IEEE Consumer Commun. Netw. Conf. (CCNC '12)*, pp. 50-51, Las Vegas, USA, Jan. 2012.
- [8] J. François, T. Cholez, and T. Engel, "CCN traffic optimization for IoT," in *Proc. 4th Int. Conf. Netw. Future(NoF)*, pp. 1-5, Pohang, 2013.
- [9] K. Kim, S. Choi, S. Kim, and B. H. Roh, "A push-enabling scheme for live streaming system in content-centric networking," in *Proc. CoNEXT Student Workshop '13, ACM*, pp. 49-52, 2013.
- [10] W. You, B. Mathieu, P. Truong, J. Peltier, and G. Simon, "Dipit: a distributed bloom-filter based pit table for ccn nodes," in *Proc. of ICCCN '12, Munich, Germany*, Jul. 2012.
- [11] H. Dai, Y. Wang, J. Fan, and B. Liu. "Mitigate ddos attacks in ndn by interest traceback," in *Proc. NOMEN '13, Turin, Italy*, Apr. 2013.
- [12] A. Afanasyev, P. Mahadevany, I. Moiseenko, E. Uzuny, and L. Zhang, "Interest flooding attack and countermeasures in named data networking," in *Proc. IFIP Netw. Conf.*, pp. 1-9, Brooklyn, NY, May 2013.
- [13] S. S. Seo, J. M. Kang, Y. Han, and J. W. K. Hong, "Analysis and performance evaluation of data transport methods in content-centric networking," in *Proc. 2013 15th Asia-Pacific, IEEE Network Operations and Management Symp. (APNOMS)*, pp. 1-3, 2013.
- [14] A. Afanasyev, I. Moiseenko, and L. Zhang, *ndnSIM: NDN simulator for NS-3*, Technical Report NDN-0005, NDN Project, 2012.
- [15] S. Jung, H. Park, and T. Kwon, "A study on the data push using push table in content networking," in *Proc. KICS winter Conf.*, pp.

667-668, Pyeongchang, Korea, Jan. 2014.

- [16] H. Park and T. Kwon, "Military routing scheme using content-centric mobile ad hoc networks," *J. KICS*, vol. 39C, no. 04, pp. 334-342, 2014.
- [17] S. Jung, H. Park, and T. Kwon, "Data pushing using IPT in content-centric networking," in *Proc. 2014 6th Int. Conf. IEEE, Ubiquitous and Future Netw.(ICUFN)*, pp. 164-165, Jul. 2014.

**정 승 훈 (Seunghoon Jung)**



2003년 : 한국해양대학교 전자통신공학과 졸업  
2013년 1월~현재 : 국방대학교 컴퓨터공학과 석사과정  
<관심분야> Next Generation Networking, Content-Centric Networking, Computer Security

**박 흥 순 (Heungsoon Park)**



2002년 : 육군사관학교 전산학과 졸업  
2007년 : 미 공군대학원 컴퓨터공학과 석사  
2012년 1월~현재 : 국방대학교 컴퓨터공학과 박사과정  
<관심분야> Next Generation Networking, Military Communications, MANET, Content-Centric Networking, Network Security

**권 태 욱 (Taewook Kwon)**



1986년 : 육군사관학교 전산학과 졸업  
1995년 : 미 해군대학원 컴퓨터공학과 석사  
2001년 : 연세대학교 컴퓨터공학 박사  
2007년~현재 : 국방대학교 컴퓨터공학과 교수

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