

데이터 이름 기반 네트워킹의 데이터 캐싱 관리 기법

김 대 업*

A Data Caching Management Scheme for NDN

DaeYoub Kim[†]

ABSTRACT

To enhance network efficiency, named-data networking (NDN) implements data caching functionality on intermediate network nodes, and then the nodes directly respond to request messages for cached data. Through the processing of request messages in intermediate node, NDN can efficiently reduce the amount of network traffic, also solve network congestion problems near data sources. Also, NDN provides a data authenticate mechanism so as to prevent various Internet accidents caused from the absence of an authentication mechanism. Hence, through applying NDN to various smart IT convergence services, it is expected to efficiently control the explosive growth of network traffic as well as to provide more secure services. Basically, it is important factors of NDN which data is cached and where nodes caching data is located in a network topology. This paper first analyzes previous works caching content based on the popularity of the content. Then ii investigates the hitting rate of caches in each node of a network topology, and then propose an improved caching scheme based on the result of the analyzation. Finally, it evaluates the performance of the proposal.

Key words: ICN, NDN, Cache Management, Redundancy

1. 서 론

초기 인터넷의 주요 개발 목적은 원격 호스트들 사이의 안전한 네트워크 연결을 제공하는 것이었다. 그러므로 현재와 같은 멀티미디어 데이터 전송 증가로 인한 잦은 네트워크 병목현상, 취약한 보안으로 인한 심각한 침해 사고, 호스트의 빈번한 이동으로 인한 비효율성과 같은 다양한 문제점들과 그 해결 방안을 고려하지 않았다[1]. 특히, 다양한 스마트 IT 융복합 서비스의 증가는 이와 같은 문제점들을 더욱 심화시킬 수 있으며, 인터넷의 기존 문제점들은 IT 융복합 서비스의 저변확대를 방해하는 주요 요인이

될 수 있다. 그러므로 이와 같은 문제들을 해결하고 인터넷을 통하여 다양한 데이터 및 정보를 보다 효과적으로 지원하기 위한 미래 인터넷 기술 연구가 여러 방향으로 진행되고 있다[2-4]. 특히, 미래 인터넷 기술 중 하나인 정보 중심의 네트워킹 기술(Information Centric Networking, ICN)은 데이터 소스에게 집중되는 데이터 요청 메시지를 효율적으로 분산 처리하기 위하여 멀티미디어 프락시 시스템(Multimedia Proxy System)이나 네트워크 기기(Network Node)에 데이터를 임시 저장한 후, 데이터 소스를 대신하여 이들 기기들이 데이터 요청 메시지를 직접 처리하는 기술을 제공하고 있다.

* Corresponding Author : DaeYoub Kim, Address: (445-743) 17, Wauan-gil, Bongdam-eup, Hwaseong-si, Gyeonggi-do, Korea, TEL : +82-31-229-8352, FAX : +82-31-229-8352, E-mail : daeyoub69@suwon.ac.kr
Receipt date : Jan. 22, 2016, Approval date : Jan. 26, 2016
[†] Dept. of Information Security, Suwon University

* This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2013R1A1A 2008389)

ICN 기술 중 하나인 데이터 이름 기반 네트워킹 (Named-Data Networking, NDN)은 데이터의 계층화된 고유 이름에 기반한 패킷 전송과 네트워크 기기에 데이터 임시 저장 기능(Caching)을 구현하여 효과적으로 데이터를 전송할 수 있도록 설계되었다 [3, 4]. 기본적으로 네트워크 노드에 저장된 데이터를 이용하여 네트워크의 효율성을 높이고 있기 때문에 캐싱되는 데이터의 종류와 데이터를 캐싱하는 네트워크 노드의 네트워크 토폴로지 상에서 위치가 NDN의 성능에 매우 큰 영향을 미친다.

본 논문에서는 데이터의 이용 빈도를 기반으로 데이터 캐싱 노드를 선택/운영하는 NDN 데이터 캐싱 기법들을 살펴보고, 캐싱 기법의 성능 개선을 위하여 네트워크 토폴로지 상에서의 노드별 캐싱 이용률을 관찰하고, 관찰 결과를 토대로 개선된 NDN 캐싱 기법을 제안한다.

2. NDN Overview

인터넷의 성능 향상을 위하여 제안된 NDN은 다음과 같은 특징을 갖고 있다:

- (1) 데이터의 고유 이름 기반의 패킷 포워딩
- (2) 중간 네트워크 노드에서의 데이터 캐싱 및 포워딩

- (3) 전자 서명 기반의 데이터 인증

또한, 이와 같은 특징을 구현하기 위하여 NDN은 다음과 같은 추가적인 요소들을 정의한다:

(1) FIB (Forwarding Information Base) 테이블: 데이터 요청 메시지(Interest)에 포함되어 있는 데이터 이름/식별자를 기반으로 해당 Interest를 포워딩할 전송 인터페이스 (Face)를 결정할 때 필요한 정보를 제공한다.

(2) PIT (Pending Interest Table): 응답 메시지 (Data) 수신 시, 해당 Data를 사용자들에게 전송하기 위하여, 수신된 Interest와 incoming Face 정보를 기록/관리한다.

(3) 네트워크 캐쉬 (Content Store, CS): 수신된 Data를 임시 저장한다.

(4) 데이터 서명 및 검증: 전송되는 Data는 각각 처음 Data를 생성한 생성자(Publisher)의 전자 서명을 포함하고 있으며, 수신자는 이 전자 서명 값을 이용하여 Publisher와 데이터 위/변조 여부를 확인한다.

Fig. 1은 NDN의 Interest와 Data 처리 절차를 설명한다.

- (1) 수신 노드의 인터페이스 (예를 들어, Face 0)로

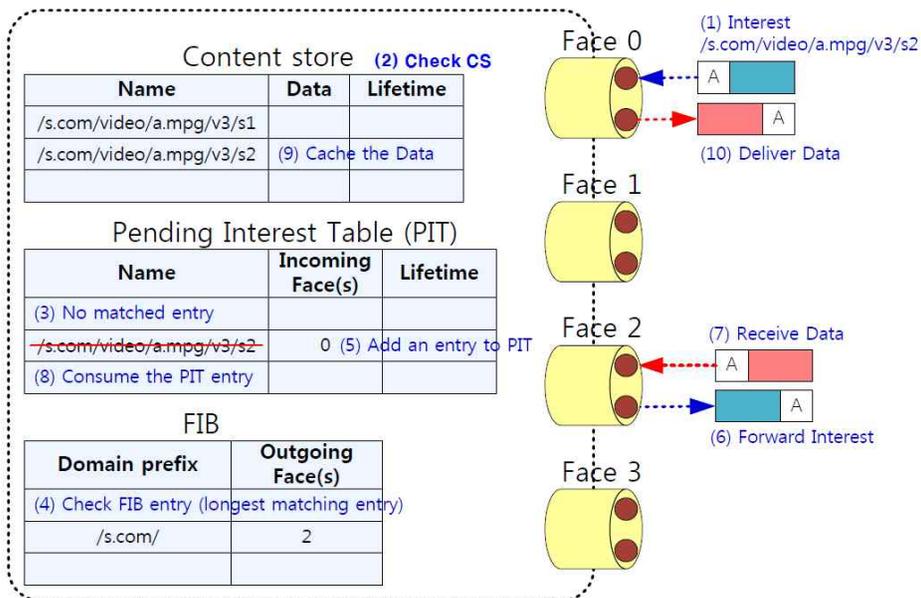


Fig. 1. NDN Interest/Data Forwarding Process.

Interest를 수신한다.

(2) Interest에 해당하는 Data를 CS에 저장하고 있는지 여부를 확인한다. 만약 해당 Data를 저장하고 있다면, 저장하고 있는 Data를 Face 0를 통하여 전송한 후, Interest 처리를 완료한다.

(3) Interest에 해당하는 entry가 PIT에 이미 존재하는지 확인한다. 만약 대응되는 entry가 PIT에 존재한다면, 해당 entry에 Interest의 유입 Face (Face 0)를 추가한 후, Interest 처리를 완료한다.

(4) FIB 테이블을 참조하여, Interest를 포워딩할 Face (예를 들어, Face 2)를 선택한다.

(5) PIT에 Interest를 위한 새로운 entry를 추가한다.

(6) 4단계에서 선택된 Face로 Interest를 전송한 후, Interest 처리 절차를 완료한다.

(7) 이 후, 노드의 인터페이스 (Face 2)로 Data를 수신한다.

(8) 수신된 Data에 해당하는 entry가 PIT에 존재하는지 확인한다. 만약 해당 entry가 PIT에 존재하지 않는다면, 수신된 Data를 폐기한 후, 처리 절차를 종료한다.

(9) Data를 CS에 저장한다.

(10) 단계 8에서 검색된 entry에 기록되어 있는 Face들로 Data를 전송한 후, 해당 entry를 PIT에서 삭제한다.

3. NDN 캐쉬 관리 기술

기본적인 NDN은 전송되는 모든 데이터를 전송 경로상의 모든 네트워크 노드에 저장한 후, 이를 이용하여 네트워크 성능을 향상시킬 수 있음을 보여주고 있다. 그러나 전송되는 모든 데이터를 캐싱할 경우, 각 노드마다 데이터 캐싱을 위해 대용량의 CS를 운영해야 한다. 또한, 이와 같은 대용량의 CS 운영은 저장 공간에 대한 오버헤드 뿐만 아니라, Interest를 수신할 때마다 저장된 많은 수의 데이터를 매번 검색해야 하기 때문에 전체적인 응답 시간을 지연시킬 수 있다. 그러므로 전송되는 모든 데이터를 모든 노드가 저장하는 것은 매우 비효율적이다. 이와 같은 문제를 해결하기 위하여 데이터의 이용 빈도에 따라 캐싱 노드를 선택/운영하는 다양한 기법들이 다음과 같이 제안되었다[5-9].

(1) LCD: 임의의 노드가 저장하고 있는 데이터에 대한 Interest를 수신하여, 해당 데이터를 전송하는 경우, 해당 데이터의 전송 경로 위에 있는 노드들 중에서 첫 번째로 데이터를 수신한 노드만 해당 데이터를 캐싱 한다. 즉, 전송 경로 위에 있는 나머지 노드들은 데이터를 캐싱 하지 않고 전송만한다. 그림 2-(A)는 LCD의 운영 방법을 설명한다. 데이터를 저장하고 있는 Source가 Interest를 수신하면, Interest의 전송 경로의 반대 방향을 따라 Data가 전송될 때, Source와 직접 연결된 A-Node만 Data를 캐싱 한다. 이 후, 해당 데이터를 요청하는 Interest를 A-Node가 수신하면, Data 전송 경로 위에 있는 노드 중에서 A-Node와 직접 연결된 B-Node만 Data를 캐싱 한다. 이와 같은 캐싱 정책을 구현할 때, 데이터의 요청 빈도에 따라 데이터를 캐싱 하는 노드의 수를 점진적으로 증가시킬 수 있으며, 요청 빈도가 높은 데이터의 경우 사용자가 포함된 네트워크 (Edge Network)의 노드에 캐싱 된다.

(2) MCD: LCD는 데이터를 캐싱하는 노드의 수를 점진적으로 늘려간다. 그러나 Interest 전송 경로의 제일 앞단에 위치한 노드에 데이터가 캐싱되어 있는 경우, 전송 경로 상에서 해당 노드보다 뒤에 위치한 노드들에 캐싱된 데이터는 사용되지 않는다. 즉, 사용되지 않는 불필요한 데이터 캐싱이 발생하게 된다. 이와 같은 문제를 해결하기 위해서 MCD는 데이터의 이용 빈도에 따라 데이터를 캐싱하는 노드를 사용자의 Edge Network 방향으로 점진적으로 이동시킨다. 즉, 데이터 소스가 아닌 중간 노드의 경우, CS에 저장된 데이터를 전송한 후, CS에서 해당 데이터를 삭제한다. Fig. 2-(B)는 MCD의 운영 방법을 설명한다. A-Node가 캐싱하고 있는 데이터에 대한 Interest를 수신하여 대응되는 Data를 전송하면, 전송 경로 상에서 A-Node와 직접 연결된 B-Node만 Data를 캐싱 한다. 이 때, A-Node는 Data를 전송한 후, CS에서 해당 Data를 삭제한다. MCD를 적용할 경우, Interest 전송 경로 위에 있는 노드들 중에서 데이터를 캐싱하고 있는 노드는 한 개뿐이다. 그러나 A-Node에서 데이터가 삭제된 후, 다른 경로를 통하여 A-Node가 삭제된 데이터에 대한 Interest를 수신하면, A-Node는 이를 처리할 수 없기 때문에 Interest를 데이터 소스에게 전송해야 한다. 이러한 경우, 네트워크 성능 저하를 초래할 수 있다.

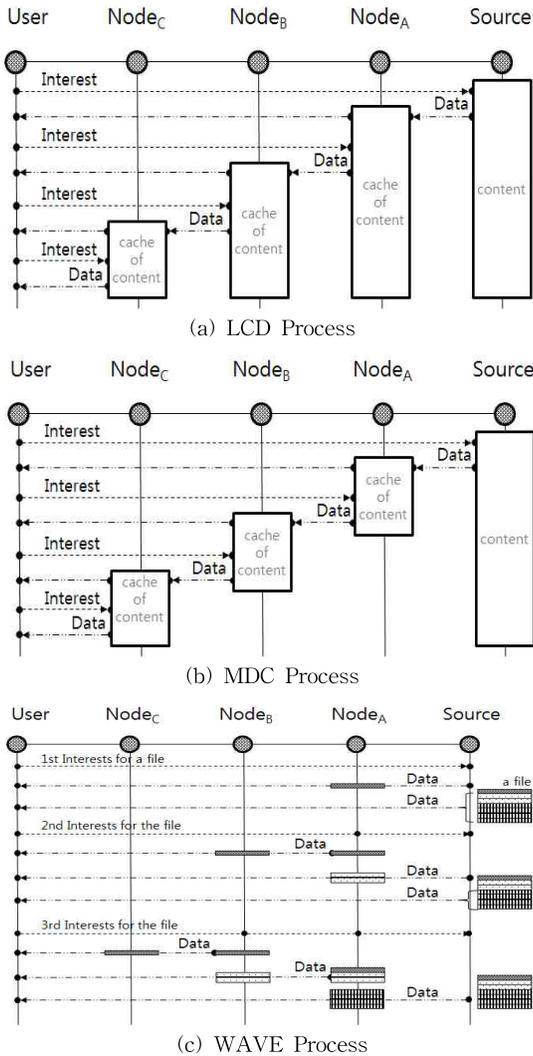


Fig. 2 NDN Cache Management Schemes.

(3) WAVE: NDN은 파일(File)을 일정 크기 이하의 여러 segment로 단편화 한 후, 각각의 segment를 단일 데이터로 간주하여 처리한다. LCD와 MCD는 이렇게 단편화된 데이터 단위로 캐싱을 운영한다. 즉, LCD와 MCD는 개별 데이터의 이용 빈도에 따라 해당 데이터의 캐싱을 결정한다. 반면에, WAVE는 파일의 이용 빈도를 이용하여 캐싱을 결정한다. 특히, WAVE는 데이터 캐싱 단위를 파일의 요청 빈도에 따라 지수 승으로 증가 시킨다. 예를 들어 Fig. 2-(C)에서와 같이, 파일에 대한 첫 번째 요청 Interest 들을 수신하며, 해당 파일의 첫 번째 segment의 Data가 캐싱 되기 시작하고, 해당 파일에 대한 두 번

째 요청 Interest들이 수신되면, 두 번째 segment와 세 번째 segment의 Data들이 캐싱 되기 시작하며, 세 번째 요청에는 네 번째부터 일곱 번째 segment의 Data들이 캐싱 되기 시작한다. 이와 같은 캐싱을 통해서 WAVE는 LCD와 같이 점진적으로 데이터를 사용자의 Edge Network에 캐싱 하지만, WAVE는 이용 빈도에 따라 캐싱 증가 속도를 지수 승으로 증가 시킬 수 있다.

이와 같은 기법들의 공통적인 특징은 데이터의 이용 빈도가 높을수록 사용자의 Edge Network의 노드에 데이터를 캐싱하고, 해당 데이터에 대한 Interest를 사용자의 Edge Network 안에서 응답 처리되게 함으로써, Core Network에서 교환되는 패킷의 수를 효과적으로 줄일 수 있을 것으로 예상했다. 또한, 전송되는 데이터를 모두 캐싱하는 것이 아니라, 이용 빈도에 따라 캐싱되는 노드의 수를 증가시키기 때문에, 데이터 캐싱을 위한 노드의 저장 공간을 보다 효율적으로 유지/관리할 수 있을 것으로 기대되었다.

그러나 이와 같은 결과를 얻기 위해서는 사용자가 요청한 데이터가 사용자의 Edge Network에 캐싱되어 있어야 한다. 즉, 해당 데이터의 이용 빈도가 매우 높은 데이터의 경우에만 이와 같은 성능 구현이 가능하다. 실제로 2015년 Youtube의 이용 통계 자료에 의하면, 10억 명 이상의 사용자가 지난 1년 동안 Youtube를 통해 비디오를 시청했으며, 매일 평균 40억 개의 비디오가 시청되고 있으며, 가장 높은 누적 시청 회수는 2억 5천회 이상의 시청 기록을 갖고 있다. 그러나 9개의 주요 카테고리 별 누적 평균 시청 회수를 분석할 때, 가장 많은 평균 이용 회수 기록을 갖는 카테고리는 Entertainment로 평균 9,816건의 시청을, 가장 적은 카테고리는 People and Blogs로 2,354건의 평균 시청 회수를 기록하고 있다 [10-13]. 즉, 누적 평균을 고려할 때 대부분의 데이터들이 일만 건 이하의 시청 기록을 갖고 있다. 이는 데이터 당 평균 0.001%의 이용자들만이 시청한다는 것을 의미한다. 그러므로 사용자의 Edge Network에 데이터가 캐싱 되는 경우는 전체 데이터 이용 건수를 고려할 때 매우 적을 수 있다. Fig. 3은 이와 같은 분석을 바탕으로 계층적으로 구성된 네트워크 도메인의 각 Level에 캐싱된 데이터의 이용 빈도를 조사한 결과이다. 계층적 네트워크 구조의 Root에 해당하는 Core Network Node에 데이터가 저장될수록 데이터

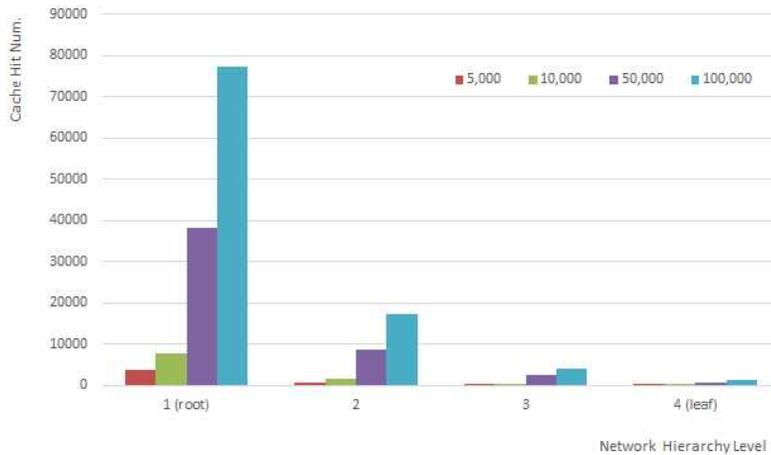


Fig. 3 Cache Hitting Rate Comparison Considering Network Hierarchy Level.

의 이용 빈도가 높고, Leaf Node에 해당하는 Edge Network Node에 데이터가 저장될수록 데이터 이용 빈도가 낮게 조사된다. 그러므로 이와 같은 각각의 노드별 데이터 이용 빈도를 고려하여 NDN 데이터 캐싱 기법을 개선할 필요가 있다.

4. 노드 이용 빈도 기반 NDN 캐쉬 관리 기술

네트워크 노드에 캐싱된 데이터의 이용 빈도를 높이기 위하여 다음과 같은 두 가지 원칙을 기준으로 데이터 캐싱을 운영하는 방안을 제안 한다:

(1) 네트워크 성능 개선: 사용자가 생성한 Interest의 전송 경로 위에 있는 노드들 중에서, 많은 Interest들이 공통적으로 경유하는 노드에 데이터를 캐싱 한다.

(2) 저장 공간의 효율적 운영: 데이터의 이용 빈도에 따라 데이터 캐싱 영역을 점진적으로 확장한다.

이와 같은 데이터 캐싱 운영 기준에 따라 NDN 데이터 캐싱을 개선하기 위하여 네트워크 토폴로지 구성과 캐싱 프로세스를 다음과 같이 제안한다.

4.1 도메인 식별자 기반 계층적 네트워크 토폴로지

많은 수의 노드들을 네트워크로 연결하고, 이들 노드들 사이의 전송 효율성을 높이기 위하여 네트워크 도메인에 포함된 노드들을 계층적으로 구성하는 것이 일반적이다. 또한, NDN은 Domain Name Server와 같은 Resolution Server를 이용하지 않고 Interet/Data를 전송하기 위하여 데이터의 유일한 식

별자인 데이터 이름을 계층화하여 운영한다.

본 논문에서는 NDN의 계층화된 데이터 이름과 네트워크 도메인의 계층을 연계하여 운영하는 방안을 제안한다. 즉, Fig. 4처럼, 명명된 계층화된 네트워크 도메인(Named-Network Doamin Hierarchy)을 다음과 같이 운영한다.

(1) 각각의 네트워크 도메인 (CU-HD과 CS_HD)은 Border Gateway를 Root Node로 하고, 계층화된 서브 도메인마다 Gateway를 운영한다. 또한, 이들 도메인(서브도메인)/사용자 호스트는 계층화된 도메인/호스트 식별자를 각각 갖고 있으며, 도메인/서브도메인의 Gateway 관리자는 도메인/서브도메인 식별자를 해당 Gateway에 설정되고, 사용자는 사용자 호스트에 호스트 식별자를 설정한다.

(2) 데이터의 유일한 식별자는 이들 계층화된 호스트 식별자를 이용하여 생성한다. 예를 들어 데이터 소스의 호스트가 연결된 네트워크의 계층의 도메인/서브도메인의 식별자를 구성하는 요소가 Root Node로부터 각각 'dn1', 'dn2', 'dn3' 라하고, 해당 호스트의 식별자 요소를 'sou'라고 한다면, 해당 호스트의 네트워크 계층 식별자는 'dn1/dn2/dn3/son'이 된다. 이 때 해당 호스트에서 생성/배포하는 데이터의 이름은 'dn1/dn2/dn3/son'을 데이터 이름 접두어(Content Name Prefixs)로 사용한다.

4.2 도메인 식별자 기반 데이터 캐싱(DNbc)

Fig. 4에서와 같이, 계층화된 도메인 식별자를 기

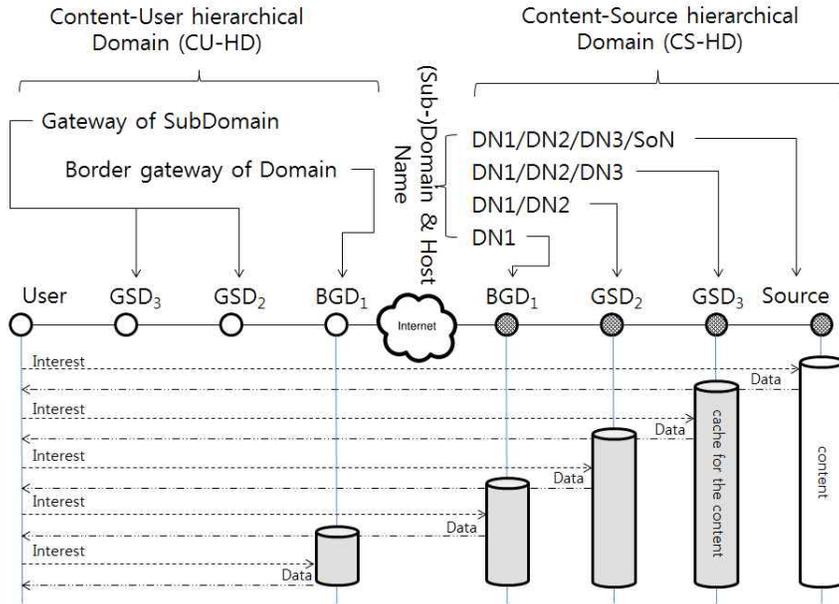


Fig. 4 Named-Network Domain Hierarchy.

반으로 데이터를 점진적으로 캐싱 하기 위하여, 본 논문에서는 기존 Data 구성 형식에 Caching Flag Bit (CFB)를 추가한다. Interest를 수신한 노드가 해당 Interest에 의해 요청된 데이터를 CS에 캐싱 하고 있다면, 다음과 같은 절차를 따라 Interest를 처리 한다:

- (1) PIT 검사를 수행한다. 수행 절차는 기본적인 NDN 수행 절차와 동일하다.
- (2) CS 검사를 수행한다. 수행 절차는 기본적인 NDN 수행 절차와 동일하다. 가정에 따라 해당 노드의 CS에는 대응하는 데이터가 저장되어 있다.
- (3) CS에 저장되어 있는 데이터의 CFB를 True로 설정한 후, 해당 Interest의 유입 Face를 통하여 전송 한다.
- (4) PIT에서 해당 Interest에 대응하는 entry를 삭제한 후, Interest 처리 절차를 종료한다.

Data를 수신한 노드는 다음과 같은 절차를 따라 수신된 Data를 처리한다.

- (1) PIT 검사를 수행한다. 수행 절차는 기본적인 NDN 수행 절차와 동일하다.
- (2) 수신된 Data의 CFB를 확인한다. CFB가 False 이면, 수신된 Data를 CS에 저장하지 않는다.
- (3) CFB가 True이면, 다음과 같은 두 가지 경우에

만 해당 Data를 CS에 저장한다.

- ① 수신 노드의 네트워크 계층 이름이 Data의 데이터 계층 이름의 prefix에 포함된 경우. 즉, 수신 노드가 데이터 소스의 네트워크 도메인의 Root Node와 데이터 소스 사이의 전송 경로 위에 존재하는 Gateway인 경우.
- ② 수신 노드가 네트워크 도메인의 Root Node인 경우. 즉, 수신 노드가 Core Network 망을 구성하는 노드인 경우.
- (4) 수신된 Data의 CFB를 False로 수정한다.
- (5) Data를 PIT에 기록된 관련 entry의 Face들을 통해서 전송한다. 이 때, 어떤 경우에든지 Data의 CFB는 항상 False이다.
- (6) PIT에서 해당 entry를 삭제한 후, Data 처리 절차를 종료한다.

5. 성능 평가

5.1 성능 평가를 위한 환경 구성

제안된 데이터 캐싱 기법의 성능 평가를 위하여 NDN, LCD 그리고 제안된 DNbC의 성능을 비교 평가한다. 이를 위하여, 본 논문에서는 랜덤하게 선택된 사용자들이 생성된 데이터를 랜덤하게 요청하는 시뮬레이션을 구현하여 그 결과를 관찰하였다. 시뮬

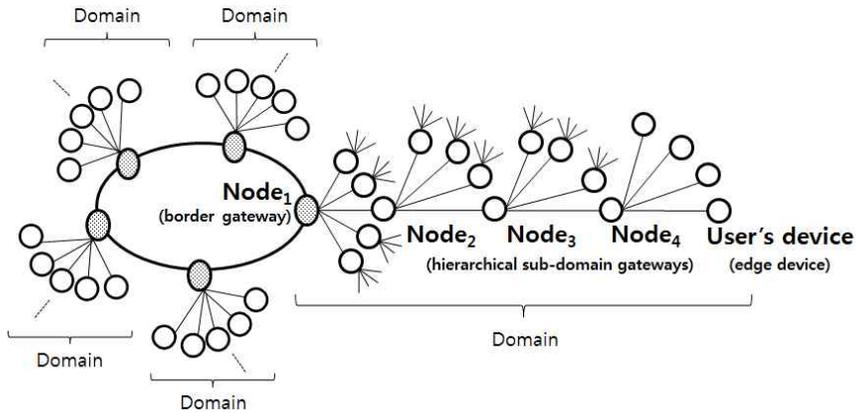


Fig. 5 Network Topology for Simulation.

레이션은 Fig. 5와 같이 5개의 서로 다른 Network Domain으로 구성된 Core Network 망을 가정하고 있으며, 각각의 Network Domain은 다음과 같이 계층적으로 구성하였다.

- (1) Root Node는 5개의 Sub-domain들을 Child로 갖으며, Leaf Node를 제외한 나머지 노드들은 각각 4개의 Sub-domain들을 Child로 갖는다.
- (2) Leaf node는 사용자 기기로 간주하며, Interest의 최초 생성자는 Leaf node로 한정한다.
- (3) 각각의 노드는 Interest 전송을 위한 FIB 테이블을 구성/관리한다고 가정한다.

네트워크를 구성하는 각각의 노드들의 환경 설정은 다음과 같다:

- (1) 네트워크를 통해 접근 가능한 데이터는 16,000개이며, 랜덤하게 선택된 노드들에서 생성한다. 또한, 이들 데이터는 동일한 이용 빈도를 갖는다.
- (2) 각각의 노드는 1000개의 데이터를 동시에 저장할 수 있는 CS를 운영한다.
- (3) 캐싱된 데이터의 생명주기(Lifetime)는 모든 노드에서 50초로 설정한다.
- (4) 랜덤하게 선택된 Leaf Node에서, 16,000개의 데이터 중 하나를 랜덤하게 선택하여 요청한다.
- (5) 평균 응답률은 96.7%로 설정한다.

5.2 성능 평가 결과

Fig. 6은 시뮬레이션 과정 중에서 각각의 노드에 캐싱된 데이터의 평균 개수의 총합을 나타낸다. LCD와 DNbC는 데이터의 요청 빈도에 따라서 해당 데이

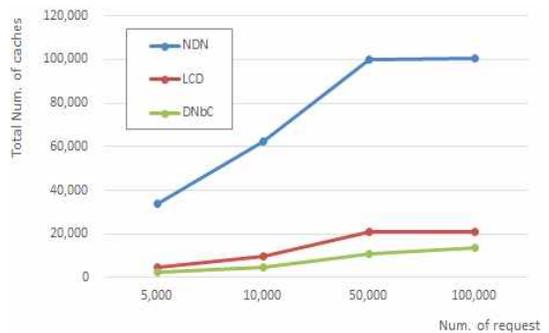


Fig. 6 The Total Number of Caches.

터의 캐싱 노드의 수를 점진적으로 늘려나가기 때문에 모든 노드에 데이터를 캐싱 하는 NDN에 비하여 효율적으로 네트워크 캐시를 운영할 수 있다. 그림 6에서와 같이 NDN에 비해 20% 정도의 캐시 만으로도 서비스 운영이 가능하다. 또한, DNbC의 경우, 점진적으로 캐싱 되는 노드를 Core Network으로 한정시켜 운영하기 때문에, 사용자의 Edge Network에 추가적으로 캐싱되는 경우는 데이터 소스가 포함된 네트워크 도메인으로 한정된다. 그러므로 LCD에 비해 30% 정도의 추가 절감 효과를 기대할 수 있다.

Fig. 7은 NDN, LCD, DNbC의 Interest 전송량을 비교 분석한 결과를 나타낸다. 최대 100,000개의 Interest가 랜덤하게 선택된 데이터를 요청하기 위하여 생성/전송될 때, 네트워크 노드에서 해당 Interest들을 포워딩한 수를 누적하여 측정하였다: NDN과 비교할 때 LCD와 DNbC는 2% 정도의 Interest 전송량이 증가한다. 그러나 LCD와 DNbC만을 비교할 때, 오히려 0.2% 정도 감소하는 것으로 나타났다. 이는

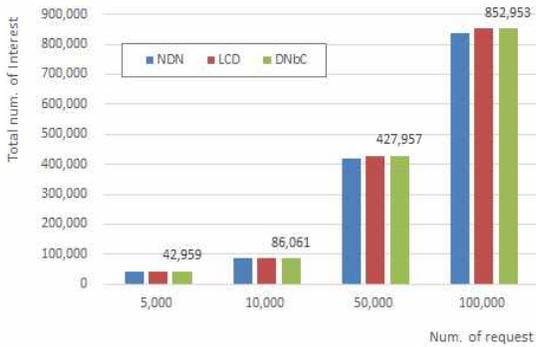


Fig. 7 The Total Number of transmitted Interests.

데이터 이용 빈도와 오차를 고려할 때 무시할만한 수치이지만, 데이터를 사용자의 Edge Network에까지 캐싱하는 것에 비해, Core Network에까지 캐싱하는 것이 전송 효율에 큰 영향을 미치지 않음을 알 수 있다.

또한, 내부 도메인과 Core Network을 구성하는 Border Gateway로 캐싱 구간을 이중화 하여 제한함으로써, Interest가 집중되는 영역을 분산시키는 NDN의 장점을 그대로 유지시키면서도 네트워크와 스토리지의 오버헤드를 개선할 수 있다.

6. 결 론

NDN은 네트워크 노드에 캐싱된 데이터를 활용하여 네트워크의 효율성을 높이기 위해 제안되었다. 그러므로 어떤 데이터를 어느 위치에 캐싱하여 운영할지를 결정하는 것이 무엇보다 중요하다. 지금까지 발표된 대부분의 연구들은 사용자 노드가 위치한 Edge Network에 데이터를 캐싱하는 방안들이 대부분을 차지하고 있다. 이는 Interest를 사용자의 Edge Network 내에서 응답 처리되게 함으로써 Core Network으로 유입되는 Interest/Data의 수를 줄일 수 있을 것으로 예상했기 때문이다. 그러나 본 논문에서는 이와 같은 연구 결과는 일부 인기 데이터에 대해서만 도출이 가능하고, 대부분의 데이터는 일부 Edge Network에만 캐싱 되기 때문에, 해당 Interest들이 여전히 Core Network로 유입될 수 있음을 지적하였다.

이와 같은 문제들을 해결하기 위하여 본 논문은 기존의 데이터 요청 빈도에 따른 점진적으로 데이터 캐싱 범위를 확대하던 기존의 기술에 노드의 접근 빈도를 추가적으로 감안하여 데이터 캐싱 여부를 결

정하도록 제안한다.

본 논문의 제안을 시뮬레이션을 통해 관찰한 결과 기존 캐싱 기법과 비교할 때 유사한 네트워크 효율성을 나타내면서도 스토리지 효율성을 개선할 수 있음을 알 수 있다. 또한, 내부 도메인과 Core Network을 구성하는 Border Gateway로 캐싱 구간을 이중화하여 제한함으로써, Interest가 집중되는 영역을 분산시키는 NDN의 장점을 그대로 유지시키면서도 네트워크와 스토리지의 오버헤드를 개선할 수 있다. 그러므로 네트워크 캐시의 효율적인 운영을 통하여 점차 다양해지고 있는 스마트 IT 융복합 서비스의 안정적인 구현이 가능할 것으로 기대된다.

REFERENCE

- [1] D. Clark, "The Design Philosophy of the DARPA Internet Protocols," *ACM SIGCOMM Computer Communications Review*, Vol. 18, No. 1, pp. 106-114, 1988.
- [2] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlmann, "A Survey of Information-Centric Networking," *IEEE Communications Magazine*, Vol. 50, No. 7, pp. 26-36, 2012.
- [3] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking Named Content," *Proceeding of 5th International Conference on Emerging Networking Experiments and Technologies*, pp. 1-12, 2009.
- [4] D. Kim, "Content Centric Networking Naming Scheme for Efficient Data Sharing," *Journal of Korea Multimedia Society*, Vol. 15, No. 9, pp. 1126-1132, 2012.
- [5] I. Psaras, W.K. Chai, and G. Pavlou, "In-network Cache Management and Resource Allocation for Information-centric Networks," *IEEE Transactions on Parallel Distributed Systems*, Vol. 25, No. 11, pp. 2920-2931, 2013.
- [6] N. Laoutaris, H. Che, and I. Stavrakakis, "The lcd Interconnection of Lru Caches and Its Analysis," *Performance Evaluation*, Vol. 63,

No. 7, pp. 609-634, 2006.

[7] G. Zhang, Y. Ki, and T. Lin, "Caching in Information Centric Networking: A Survey," *Computer Networks*, Vol. 57, Issue 16, pp. 3128-3141, 2013.

[8] M. Zhang, H. Luo, and H. Zhang, "A Survey of Caching Mechanisms in Information-Centric Networking," *IEEE Communication Surveys & Tutorials*, Vol. 17, No. 3, pp. 1473-1499, 2015

[9] K. Cho, M. Lee, K. Park, T. Kwon, Y. Choi, and S. Pack, "WAVE: Popularity-based and Collaborative In-network Caching for Content-Oriented Networks," *Proceeding of IEEE INFOCOM Workshop on Emerging Design Choices in Name-Oriented Networking*, pp. 316-321, 2012.

[10] Youtube Statistics, <https://www.youtube.com/yt/press/en-GB/statistics.html>, 1. Feb. 2016.

[11] Youtube Trend Map, <https://www.youtube.com/trendsmap>, 1. Feb. 2016.

[12] How Many Views Does A Youtube Video Get? Average Views By Category, <http://www.reelseo.com/average-youtube-views/> 1. Feb. 2016.

[13] X. Cheng, C. Dale, and J. Liu, "Statistics and Social Network of Youtube Videos," *Proceeding of 16th International Workshop on Quality of Service*, pp. 229-238, 2008.



김 대 엽

1997년 3월~2000년 2월 고려대학교 대학원 수학과 이학박사
 1997년 9월~2000년 3월 (주)텔리맨 CAS팀 연구원
 2000년 4월~2002년 8월 시큐아이 닷컴 정보보호연구소 차장

2002년 9월~2012년 2월 삼성전자 종합기술원 전문연구원
 2012년 3월~현재 수원대학교 정보보호학과 조교수
 2016년 1월~현재 수원대학교부설 IT연구소 소장
 관심분야 : 보안프로토콜, 네트워크/시스템 보안, 콘텐츠 보안, 미래 인터넷 보안