

콘텐츠 중심 네트워크에서 성능 향상을 위한 인기도 기반 캐시 교체 기법

우 태 희*, 박 흥 순°, 김 호 길*

Popularity Based Cache Replacement Scheme to Enhance Performance in Content Centric Networks

Taehee Woo*, Heungsoon Park°, Hogil Kim*

요 약

콘텐츠 중심 네트워크(CCN)는 기존의 IP 방식과는 다른 새로운 네트워킹 패러다임으로써 데이터를 찾는 데 필요한 라우팅 정보를 콘텐츠 이름에서 찾는 방식이다. CCN은 노드마다 콘텐츠를 저장할 수 있는 캐시를 가지고 있어서 반복적인 콘텐츠 요청에 효율적으로 처리할 수 있다. 본 논문은 콘텐츠의 인기를 활용한 캐시 교체 기법을 제안하여 기존의 기법보다 캐시의 히트율을 향상시켰고, 이에 따라 서버의 부하 및 Round Trip Time(RTT) 시간이 감소하여 성능이 향상됨을 증명하였다.

Key Words : Content Centric Networking, Cache Replacement Scheme, Popularity

ABSTRACT

Unlike an existing IP routing methods, the Content Centric Network(CCN) is new networking paradigm to find the contents by content name. The CCN can effectively process the content requested repeatedly by the user because of the cache which can be storing a content. This paper proposes a popularity based cache replacement scheme. The proposed scheme improves the hit rate better than a existing scheme. Accordingly reducing the load of server and Round Trip Time(RTT).

1. 서 론

1970년대 초 알파넷의 등장 이후 TCP/IP 기반의 컴퓨터 통신은 오늘날까지 널리 사용되고 있는 가장 기본적인 컴퓨터 네트워크 모델이다. 하지만 단말기의 급격한 증가와 저장매체 비용 감소, 전 세계적 규모로의 네트워크 범위 확장 등 초기와 다른 컴퓨터 통신 환경의 변화는 기존 TCP/IP 모델에서 불필요한 위치 탐색 및 반복 전송, 병목 현상 등과 같은 문제점이 대두하였다. 이에 따라 기존 네트워크 모델을 개선한 콘텐츠

중심 네트워크(Content Centric Network, 이하 CCN)와 같은 새로운 정보 중심 네트워크(Information Centric Network) 모델이 등장하게 되었다^[1]. CCN은 노드마다 캐시 기능을 추가하여 수신된 콘텐츠를 저장하고, 이후에 동일한 요청이 올 경우에 콘텐츠의 최초 위치로 요청을 보내지 않고 중간 노드에서 즉시 콘텐츠를 전달하기 때문에 기존의 IP 방식보다 성능이 좋다^[2].

한편, 웹 캐시에 대해 연구한 기존 연구에 따르면 캐시 교체 기법은 네트워크 대역폭과 서버의 부하를

* First Author : Korea Army Academy at Young-Chung(KAAY) Department of Cyber Warfare, seeles0@gmail.com, 학생회원

° Corresponding Author : Joint Forces Military University, heungsoon.park@gmail.com, 중신회원

* Korea Army Academy at Young-Chung(KAAY) Department of Cyber Warfare, khglex@gmail.com, 정회원

논문번호 : KICS2015-08-260, Received August 20, 2015; Revised November 10, 2015; Accepted November 10, 2015

감소시키며, 사용자가 정보를 인식하는데 소요되는 지연시간을 줄일 수 있다³⁾. 그래서 캐시의 히트율(hit ratio)을 높이기 위해서 많은 연구자들이 연구를 해왔고, 캐시의 히트율이 높을수록 성능이 향상됨을 증명하였다⁴⁾. 노드마다 캐시 기능을 가지는 CCN에서도 기존의 교체 기법보다 더 높은 히트율을 가지는 교체 기법을 적용한다면 더 좋은 성능을 보일 것이다. 우리는 CCN에서 캐시의 히트율을 향상시키기 위해 콘텐츠 인기도를 기반으로한 캐시 교체 기법을 제안한다.

본 논문의 구성은 다음과 같다. 2장은 CCN과 캐시 교체 기법에 대해서 소개하고, 3장에서는 CCN에서 캐시의 히트율 향상을 위한 인기도 기반 캐시 교체 기법을 소개한다. 4장은 제안한 모델의 성능을 평가하고 분석하였으며, 5장에서 결론으로 마무리 지었다.

II. 관련연구

2.1 콘텐츠 중심 네트워크³⁾

CCN은 사용자가 정보를 요청하기 위한 Interest 패킷과 Interest 패킷에 대해 응답하고 콘텐츠 청크(chunk)를 수반한 Data 패킷으로 나뉜다. Data 패킷은 Interest 패킷의 역경로로 이동하며 사용자에게 전달된다. CCN의 라우터 구조는 Content Store(CS), Pending Interest Table(PIT), Forwarding Information Base(FIB)로 구성되며 다음과 같이 동작한다.

Interest 패킷이 CCN 라우터에 도착하면, 라우터는 CS에서 도착한 Interest 패킷과 이름이 일치하는 콘텐츠가 있는지 확인을 한다. 만약 있다면, 해당 콘텐츠를 Interest 패킷이 도착한 역경로로 보낸다. 일치하는 콘텐츠가 없다면 PIT에 일치하는 항목이 있는지 확인을 하고, PIT 항목에 있다면 그 항목에 Interest 패킷의 인터페이스를 추가한다. PIT에도 일치하는 항목이 없다면 FIB에 일치하는 항목이 있는지 확인을 하고, 일치하는 항목이 있다면 라우터는 PIT에 새로운 항목을 생성한 후에 FIB의 항목에 따라 Interest 패킷을 전달한다. FIB 항목에도 존재하지 않는다면 Interest 패킷은 폐기된다.

Data 패킷이 CCN 라우터에 도착하면, 라우터는 CS에 도착한 Data 패킷과 일치하는 콘텐츠가 존재하는지 확인을 하고, 존재한다면 폐기한다. CS에 일치하는 콘텐츠가 없다면 PIT에 일치하는 항목이 존재하는지 확인을 하고, 존재한다면 라우터는 도착한 Data 패킷을 CS에 저장하고 일치하는 PIT 항목에 표기된 인터페이스로 Data 패킷을 전송한다. 그리고 Data 패킷을 전송한 후에 해당 PIT 항목은 삭제한다. PIT에도

일치하는 항목이 없으면 Data 패킷은 폐기된다.

2.2 캐시 교체 기법

네트워크에서의 캐시 교체 기법은 데이터를 어디에 위치시킬 것인가에 대한 부분을 주로 다룬다. 노드들의 독립적인 캐시 교체 기법이 아닌 네트워크 전반적인 관점에서 어디에 캐시를 할 것인가에 대한 기법으로 Leave Copy Everywhere(LCE), Leave Copy Down(LCD), Move Copy Down(MCD), Prob 등이 있다^{5,6)}.

그림 1은 LCE, LCD, MCD, Prob의 캐시 교체 기법 알고리즘에 대해서 묘사하였다. A와 B는 중간노드, C는 콘텐츠를 가지고 있는 서버이며, A의 하위에서 콘텐츠 요청이 이루어지고 있다. LCE부터 알아보면, A가 데이터를 요청받았을 때, 데이터를 가지고 있지 않다면 B와 C를 거치는 일련의 경로를 이동하여 데이터의 요청이 발생한다. C에서 해당 데이터를 보유하고 있어 요청에 대한 데이터가 발송되면 경로에 있는 A와 B 모두 해당 데이터에 대해 캐싱한다. 결과적으로 경로에 있는 모든 중간 노드(A, B)는 데이터를 캐싱한다. 동일한 상황에서 LCD방식은 데이터의 캐싱이 전달경로에서 하위 1홉의 B에게만 발생하며, MCD에서는 하위 1홉의 B에게 데이터가 전달되고 전달된 데이터는 서버 C에서 전달과 동시에 삭제된다. Prob방식은 LCE방식과 유사하나 캐싱 여부를 데이터의 생성시기와 캐시 크기 등을 고려하여 확률적인 방법으로 수행한다.

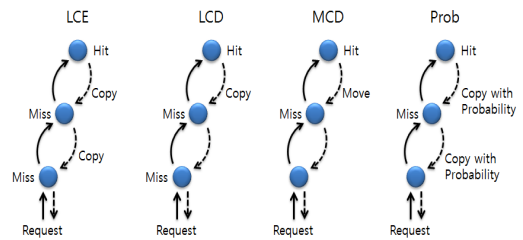


그림 1. LCE, LCD, MCD, Prob 알고리즘 동작
Fig. 1. LCE, LCD, MCD, Prob algorithm operation

2.3 CCN에서의 캐시 교체 기법

CCN노드는 단일 노드 관점에서 LRU 또는 LFU방식의 교체 기법을 기본으로 사용하고 있으며⁵⁾, 네트워크적 관점에서는 자신을 점유하는 모든 콘텐츠를 저장하는 LCE방식을 따른다. 하지만 이러한 LCE방식은 그림 2와 같이 다수의 클라이언트들이 인기 있는 콘텐츠를 요청하여 콘텐츠가 제공될시, 중간 지점에 위치한 노드 B~G에 동일한 콘텐츠의 중복 저장

이 발생하여 상대적으로 다른 콘텐츠가 캐싱될 확률을 낮추고 CCN 노드가 가지는 저장 공간을 효율적으로 사용하지 못하는 것을 볼 수 있다.

이러한 네트워크에서의 캐싱 영역에 중복을 줄이기 위한 많은 방안이 연구되고 있다⁷⁻⁹⁾. [7]은 요청 횟수에 따라서 콘텐츠의 청크를 요청자에게 가까운 노드로 순차적으로 전달하는 방안을 제안하였다. 하지만 LCD 방식을 사용하여 하위 노드에 콘텐츠를 전달하고도 상위 노드에서 콘텐츠를 보유하므로 캐싱 영역의 중복성을 야기시킨다. [8]은 중간에 교차되는 노드에 대해서만 캐싱하는 방안을 제안하였으나, 많은 교차점이 발생하는 네트워크에서는 기존의 LCD 방식과 동일하게 동작되므로 많은 중복된 콘텐츠들이 네트워크상에 존재하게 된다. [9]는 요청이 일정한 임계값 (Popularity Threshold) 수치에 이르면 연결된 모든 인터페이스에 콘텐츠를 전파를 하는 방안에 대해서 제안하였다. 하지만 임계값에 이르기 전까지 콘텐츠는 일반적인 서버-클라이언트 방식을 따르며, 임계값이 넘어 연결된 모든 인터페이스에 콘텐츠가 전달되더라도 기존의 상위 노드에는 콘텐츠를 보유하므로 중복된 콘텐츠가 발생한다.

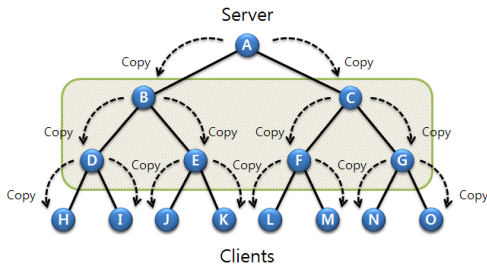


그림 2. 콘텐츠 중복 문제
Fig. 2. Contents Duplication Problem

III. 인기도 기반 캐시 교체 기법 설계

3.1 기본개념

사용자가 많이 찾는 즉, 인기 있는 콘텐츠의 캐싱이 되는 위치는 사용자에게 가까울수록 효과적일 것이다. 따라서 그림 3과 같이 서비스를 제공하고 받는 입장에서 인기가 있는 콘텐츠 일수록 사용자에게 가깝게 위치하고 인기가 낮은 콘텐츠일수록 서버 쪽에 위치하는 것이 이상적인 콘텐츠의 캐싱 위치가 된다.

인기도 기반 캐시 교체 기법에서는 네트워크를 계층적으로 보고 콘텐츠를 제공하는 노드를 상위노드, 콘텐츠를 요청하는 노드를 하위노드로 본다. 하위 노

드에 전송된 콘텐츠들은 소비자에게 전송되었거나 하위 노드에 캐싱이 이루어져 향후 추가적인 요청이 발생하지 않을 것이므로 중간 노드에서 해당 콘텐츠에 대한 캐싱은 불필요하며, 이에 따라 다른 콘텐츠보다 우선적인 교체가 이루어진다. 또한 인기 있는 콘텐츠가 신속하게 전파되기 위해 지역 내에서 인기 있는 콘텐츠를 미리 하위노드에서 캐싱 한다면 중간 노드에서 캐싱이 이루어질 가능성이 더욱 증가한다. 즉, 중간 캐싱 영역에서 콘텐츠의 중복을 줄이고 인기 있는 콘텐츠를 신속히 전파하기 위한 방안으로 인기 있는 콘텐츠를 전달하고 중복되는 콘텐츠는 신속히 제거될 필요가 있다.

이를 위해 각 CCN 노드들은 데이터의 전송이 이루어진 후 모든 페이스로 해당 데이터가 전송되었는지를 확인하여 이에 해당할 경우 해당 데이터를 최우선 교체 우선순위로 할당한다. 이를 통해 CCN 노드는 인기 있는 콘텐츠를 신속하게 하위 CCN노드에게 배포하고 자신의 CS에서 우선적으로 삭제하여 하위 순위의 인기 있는 콘텐츠를 받아들일 수 있다.

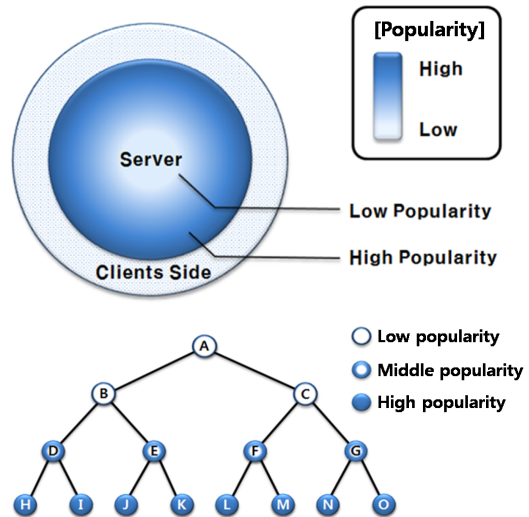


그림 3. 최적화된 콘텐츠 분배
Fig. 3. Optimized contents distribution

3.2 시스템 설계

인기도가 높은 콘텐츠는 높은 확률로 자신의 하위 노드들에게 해당 콘텐츠를 전파한다. 이를 위해 시스템은 자신의 인접 노드들이 얼마나 해당 콘텐츠를 요청하였는지를 기록하며 이를 위해 CS에 콘텐츠별로 요청한 노드들을 기록하기 위한 공간을 설계하였다. 인기도에 따른 확률적 전파 시에는 브로드캐스트로 전파가 되며 이후 해당 콘텐츠는 우선적으로 교체가

된다.

3.2.1 인기도 판단

네트워크에서 인기도는 일반적으로 알려진 Zipf 분포^[10]를 따른다고 가정한다. 만약 각각의 CCN 노드가 콘텐츠별 인기도를 미리 알고 있을 경우 Zipf 분포에 따라 인기도 높은 콘텐츠를 미리 캐싱하는 것이 적절하지만, 전체 네트워크에서 각 콘텐츠별 인기도를 알 수 있는 방법은 제한될 수밖에 없으며 특정 지역에서의 콘텐츠 인기도가 전체 네트워크에서의 콘텐츠 인기도를 항상 따른다고 보기는 어렵다. 또한 CCN에서는 콘텐츠가 매번 서버에서 제공되는 것이 아닌 중간 캐시에서 제공될 수 있으며, 노드 간 콘텐츠의 인기도에 대한 정보가 교환되지 않기 때문에 노드는 자신이 수신하고 전송하는 콘텐츠가 전체 네트워크에서 어느 정도의 인기도를 갖고 있는지 더더욱 알 수 없다.

따라서 콘텐츠별 전체 인기도에 대한 정보를 별도로 수신하지 않는 한, 노드의 콘텐츠 인기도 판단은 지역적인 방법을 사용할 수밖에 없다. 노드가 인기도를 판단할 수 있는 기준으로는 콘텐츠의 참조횟수와 연결되어 있는 Face별 요청여부를 삼을 수 있다. 우리가 제한한 기법에서는 자신과 인접한 노드들 중에서 얼마나 많은 노드들이 해당 콘텐츠를 요청하는가를 통해 판단하고, 이를 전송확률(P)이라고 정의하며 다음과 같이 수식으로 표현한다.

$$P = \frac{\# \text{ of Req.Face} + \# \text{ of Rcv.Face}}{\# \text{ of Total Face}} \times 100 \quad (1)$$

수식 (1)에서 노드의 전체 Face는 자신과 연결된 인접 노드들의 수이며, 이는 자신과 통신이 이루어진 Face의 수를 의미한다. 콘텐츠의 인기도 판단은 각 콘텐츠별로 계산되며 전체 Face 대비 얼마나 많은 Face에서 해당 콘텐츠를 요청했는지를 통합하여 계산한다. 계산된 전송확률은 해당 데이터를 요청하지 않은 하위 다른 노드들로 전송할 확률이 되며, 전송확률이 높은 콘텐츠는 더 높은 확률로 자신의 하위 노드들에게 전파된다.

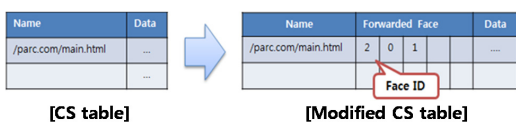


그림 4. 수정된 CS 테이블
Fig. 4. Modified CS Table

3.2.2 CS 구조 변경

전송확률을 판단하기 위해서는 전송확률을 판단하고 기록할 수 있는 공간이 필요하다. 이 공간을 위해서 그림 4와 같이 CCN 노드 모델에서 Interest 패킷과 Data 패킷이 출입되는 Face를 저장할 수 있는 공간인 “Forwarded Face” 열을 추가하였다. Forwarded Face에는 콘텐츠를 수신한 Face ID와 PIT 항목에 있는 수신된 Interest 패킷의 Face를 기록한다. 요청한 콘텐츠가 도착하여 CS에 등록될 때 콘텐츠를 수신한 Face 번호가 최초 Forwarded Face에 기록되고 PIT 항목에 기록되어 있던 Face 번호들이 순차적으로 기록된다. CS에 존재하는 콘텐츠를 다른 Face에서 요청할 경우 해당 Face가 해당 콘텐츠의 Forwarded Face에 추가적으로 기록된다. Forwarded Face에 이미 기록된 Face에서 재요청이 발생하는 경우 Face 번호는 신규로 추가되지 않으며 콘텐츠를 인접 노드로 전파하는 것에 대한 판단만 이루어진다.

3.2.3 콘텐츠의 전파

콘텐츠는 수식 (1)의 전송확률에 의해 확률적으로 전파되며, 인기도 판단과 전파여부 결정은 해당 콘텐츠가 참조될 때마다 발생한다. 전송확률이 결정되면 CCN 노드는 최초 콘텐츠를 수신한 Face를 제외한 자신 주변의 모든 Face로 콘텐츠를 전파한다. 콘텐츠의 전송이 이루어진 후 노드는 해당 콘텐츠를 더 이상 참조될 가치가 없는 자료로 판단하고 우선적으로 교체한다. 상위 노드에서 데이터가 전파된 동작을 푸싱(Pushing)이라고 정의하며, 이렇게 전파된 콘텐츠는 CS에 신규로 저장한다. 이렇게 푸싱에 의해 캐싱된 콘텐츠는 CCN 노드에서 Interest 패킷에 의해 수신된 다른 콘텐츠와 동일한 캐싱 기법을 적용받는다.

그림 5는 CCN 노드에서 콘텐츠 요청에 따른 Forwarded Face의 변화와 콘텐츠의 전파 이후 캐싱의 우선 교체과정을 나타낸다. (a) ‘Face 0’에서 콘텐츠 요청 발생시 PIT에 해당 Face번호가 기록되며, (b) 콘텐츠가 도착하면 콘텐츠가 도착한 ‘Face 2’와 PIT에 있던 ‘Face 0’가 Forwarded Face에 기록되고 인기도 판단을 시행한다. (c) 이후 ‘Face 1’에서 추가적인 콘텐츠 요청이 발생하면 노드A는 Forwarded Face에 해당 Face를 기록하고 전송확률을 판단한다. (d) 인기도 판단결과 전파확률은 100%라고 가정했을 때, 노드는 해당 콘텐츠를 보내준 최초 Face를 제외한 모든 Face로 해당 콘텐츠를 전파하고, 추가적인 캐싱 상황 발생 시 모든 Face로 보낸 콘텐츠를 우선적으로 교체한다.

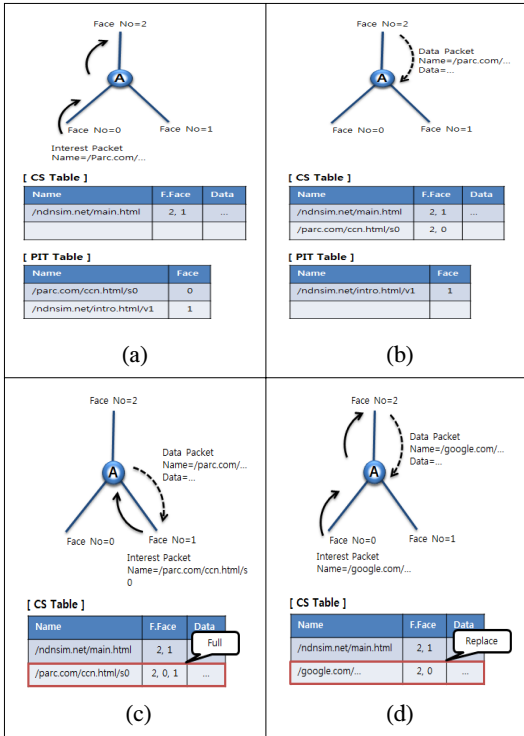


그림 5. 수정된 CS 테이블 동작
Fig. 5. Operation of modified CS Table

3.2.4 알고리즘 설계

제안한 기법은 기존의 포워딩 알고리즘을 그림 6과 같이 수정하였으며, 점선으로 표시된 부분이 추가된 부분이다. 먼저 Interest 패킷이 도착하면, Prefix의 콘텐츠 이름을 확인하여 CS에 존재하는지 확인을 한다. 존재하지 않을 경우에는 기존의 절차에 의거 동작을 하고, 존재하는 경우에는 요청한 Face로 Data 패킷을 발송한 후 콘텐츠를 다른 Face에 전파하는지에 대해 판단한다. 이때 전송확률 계산도 같이 이루어지며, 계산된 전송확률에 따라 Data 패킷을 발송 또는

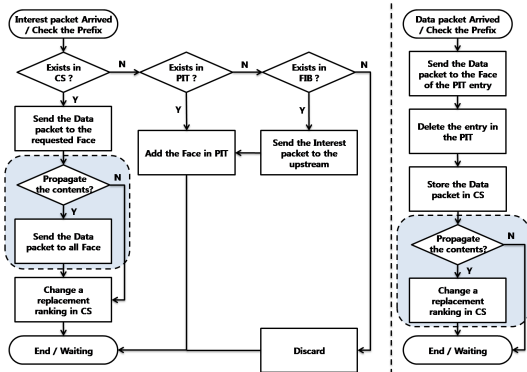


그림 6. 수정된 노드 알고리즘
Fig. 6. Modified Node Algorithm

발송하지 않게 된다.

데이터를 요청한 Face가 아닌 다른 Face로 Data 패킷이 도착하는 예외의 경우에는 그림의 오른쪽 절차에 의해서 처리된다. Data 패킷이 도착하면, CS에 저장되는 절차까지는 기존과 동일하다. 이어서 콘텐츠에 대해서 전파를 할지에 대한 판단을 하고, 만약 전송확률에 의해서 전파가 된다면 콘텐츠 우선순위를 변경한다. 만약 콘텐츠 전파가 되지 않는다면 동작을 종료한다.

IV. 그림과 표

4.1 실험환경 구성

제안한 기법의 성능분석을 위해 ndnSIM 시뮬레이터^[11]를 이용하였으며, 실험환경은 그림 7과 같이 구성하였다. 최상단에 위치한 루트 노드가 서버와 연결되어 콘텐츠를 제공하고, 각 노드는 하나의 종단 노드와 3개의 하위 노드를 갖고 종단 노드들이 Zipf 분포를 따르는 확률로 콘텐츠를 요청한다. 라우팅 전략은 ndnSIM에서 기본으로 제공하는 BestRoute를 사용하였으며, 모든 노드들은 동일한 캐시 크기를 갖도록 설계하였다.

최상단에 위치한 루트 노드가 서버와 연결되어 콘텐츠를 제공하고, 각 노드는 하나의 종단 노드와 3개의 하위 노드를 갖고 종단 노드들이 Zipf 분포를 따르

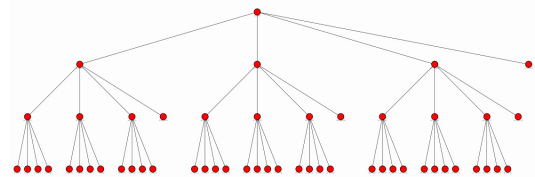


그림 7. 시뮬레이션 토폴로지
Fig. 7. Simulation topology

표 1. 시뮬레이션 파라미터
Table 1. Simulation parameters.

Simulation time	5000sec
Total cache size	0.5%, 2.5%, 5.0%, 10.0% (of all contents size)
Bandwidth	0.04, 0.13, 0.4Mbps
Node delay	50ms
Chunk size	1024
Zipf parameter α	0.8
Request frequency	10/sec
Link delay	50ms

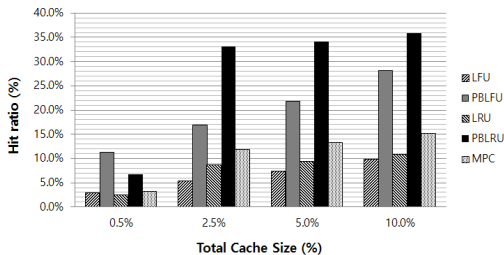
는 확률로 콘텐츠를 요청한다. 라우팅 전략은 ndnSIM에서 기본으로 제공하는 BestRoute를 사용하였으며, 실험을 위한 기타 세부적인 설정 값은 표 1과 같다. 표에서 변수인 Zipf 지수 α 는 [10]에서 실제 트래픽 분석을 통해 제안한 0.8로 설정하였다.

4.2 실험결과 및 성능분석

실험은 전체 데이터의 0.5%, 2.5%, 5%, 10%의 캐시 크기에 따른 히트율, 서버부하 감소율, Round Trip Time(RTT) 값을 측정하였다. 측정된 모델은 CCN에서 기본적으로 캐시 교체 기법으로 사용되고 있는 LFU, LRU와 2장에서 설명한 MPC[9] 모델, 우리가 제안하는 기법 2가지이다. 제안하는 기법 2가지는 LFU를 기본으로 하여 수정 설계한 PBLFU, LRU를 기본으로 하여 수정 설계한 PBLRU 라고 명명하였다. MPC 모델은 본 논문의 실험환경에 맞췄서 동작을 구현하였으며, 임계값(Popularity Threshold)은 [9]에서 최적의 결과를 보인 5로 설정하였다.

4.2.1 히트율 비교

캐시 크기의 증가에 따른 히트율 측정 결과는 그림 8과 같다. 전체적으로 네트워크에서 수용 가능한 캐시 크기가 증가할수록 히트율이 높아지며, 제안한 기법이 LFU, LRU, MPC 방식 보다 히트율이 더 좋은 것으로 나타났다. PBLFU 방식은 캐시 크기 0.5%일 때, LFU에 비해 약 3.9배까지 히트율이 증가하였으며, PBLRU 방식도 캐시 크기 2.5%일 때, LRU에 비해 최대 약 3.9배까지 히트율이 증가하였다. 이것은 캐시 크기와 상관없이 제안한 기법의 히트율이 더 우수한 결과를 보인다. 많이 요청되는 인기 콘텐츠가 반영되



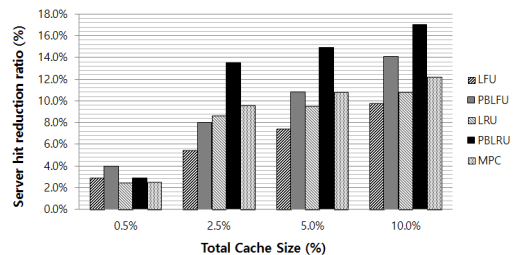
Cache size	LFU	PBLFU	LRU	PBLRU	MPC
0.5%	2.9%	11.3%	2.4%	6.8%	3.2%
2.5%	5.4%	16.9%	8.6%	33.1%	11.8%
5.0%	7.4%	21.8%	9.4%	34.1%	13.2%
10.0%	9.8%	28.1%	10.8%	35.8%	15.1%

그림 8. 캐시 크기에 따른 히트율 결과
Fig. 8. Result of hit ratio by cache size

어 전송확률에 따라 사전에 사용자에게 가까운 노드에 전파하므로 더 높은 히트율이 발생하게 된 것으로 분석된다. 한편 MPC 방식은 제안하는 임계값 이상의 여러 번 반복되는 요청이 있어야 그 콘텐츠를 주변의 노드들에게 전파하게 된다. 이렇게 인기가 있는 콘텐츠를 주변에 전송하여 LFU, LRU보다 좋은 성능을 보이지만 인접 노드 간에 콘텐츠 중복이 야기되므로 제안하는 방식보다는 낮은 성능을 보였다.

4.2.2 서버부하 감소율 비교

서버의 부하를 측정하기 위해 중단 노드들의 전체 요청대비 루트 노드에서의 Cache miss를 각 교환 방식별로 측정하여 비율로 나타냈으며 결과는 그림 9와 같다. 전체적으로 캐시의 크기가 커질수록 서버에 부하 감소율이 증가하고 있으며 제안한 기법이 더 많은 감소율을 보이고 있다. 특히, 캐시 크기 5%일 때, PBLRU는 LRU에 비해 약 1.6배 이상 감소율을 보인다. 캐시 크기가 0.5%일 경우에 LFU, LRU 방식과 제안한 PBLFU 방식은 비슷한 결과를 보인다. 이것은 캐시 크기에 비해 방대한 양의 콘텐츠들이 요청에 따라 빈도가 높게 교체되어 발생한 것과 제안한 기법이 동작하기 위한 추가된 알고리즘 및 시스템에 대한 소요가 반영되어 나타난 것으로 분석된다. PBLRU와 MPC 기법 비교 결과 PBLRU가 캐시 크기 2.5%일 때 1.41배로 PBLRU가 더 좋은 감소율을 보였고, PBLFU와 MPC 기법 비교 결과에는 캐시 크기 0.5%일 때를 제외하고 PBLFU가 좋은 성능을 보였다. MPC가 LRU를 기본 캐시 전략으로 동작하기 때문에 LRU와 비슷한 특성을 보였으며, 0.5%일 때 PBLFU가 나쁜 성능을 보인 것은 앞서 설명한 대로 추가 알



Cache size	LFU	PBLFU	LRU	PBLRU	MPC
0.5%	2.9%	4.0%	2.4%	2.9%	2.5%
2.5%	5.4%	8.0%	8.6%	13.5%	9.6%
5.0%	7.4%	10.8%	9.5%	14.9%	10.8%
10.0%	9.8%	14.1%	10.8%	17.0%	12.2%

그림 9. 캐시 크기에 따른 서버의 히트 감소율 결과
Fig. 9. Result of server hit ratio reduction ratio by cache size

고리증 동작을 위해 전송확률 계산과 수정된 CS 테이블 등을 적용하는데 시간이 소요되었기 때문에 분석된다.

4.2.3 RTT 값 비교

RTT 값은 Interest 요청에 대한 Data 패킷 수신시간을 측정하였으며 결과는 그림 10과 같다. 전체적으로는 캐시의 크기가 커질수록 줄어드는 결과를 보이고 있으며, LFU, LRU, MPC 방식보다 제안방식의 RTT 값이 더 많이 감소하였다. 캐시 크기가 10%일 때, PBLFU는 LFU에 비해 약 1.34ms 빠른 결과를 보였으며, 특히 캐시 크기 2.5%일 때, PBLRU는 LRU에 비해 약 6.52ms 만큼의 큰 격차로 좋은 결과를 보였다. 그리고 MPC는 LRU보다 짧은 RTT 값을 보여주고 있으나 제안하는 기법에 비해서는 나쁜 성능을 보인다. 이것은 임계값을 초과하여 요청이 들어와야 인접 노드들에게 전파하는 특성과 전파한 후에도 해당 콘텐츠를 중복 보유하는 특성 때문으로 보인다.

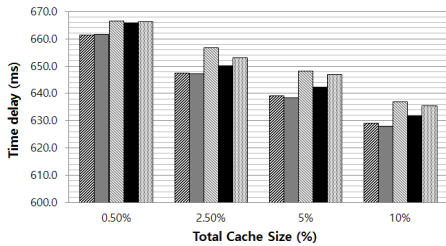
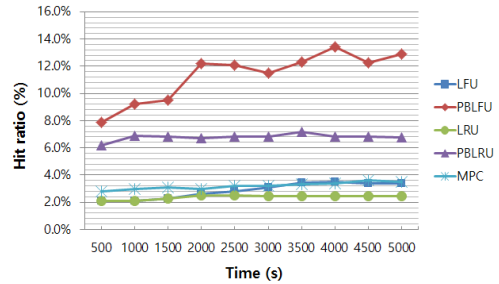


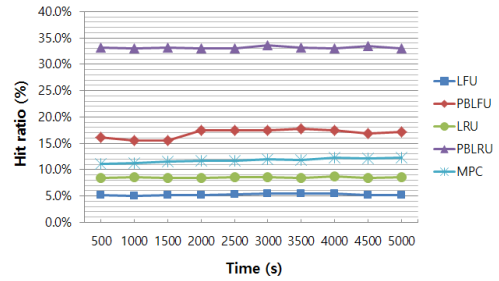
그림 10. 캐시 크기에 따른 RTT 결과
Fig. 10. Result of Round Trip Time by cache size

4.2.4 안정성 비교

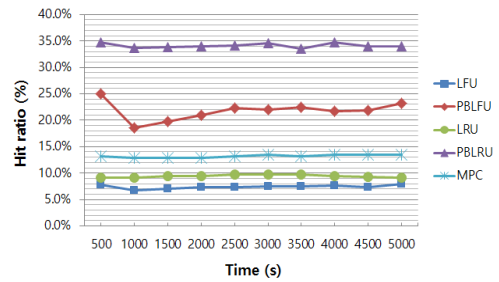
안정성 비교를 위해 시뮬레이션 시간별로 히트율을 측정하여 결과를 그림 11에서 보여주고 있다. LFU, LRU, MPC 기법은 전체적으로 안정된 결과를 보여주고 있다. 하지만 제안하는 기법 PBLFU는 시간이 진행됨에 따라서 결과 값이 급변하는 모습을 보여주고 있으며, 특히, 캐시 크기 5%일 때, 500s와 1000s 사이 구간에서는 약 14%가 변화되고 있다. 콘텐츠의 인기도와 요청하는 횟수에 종속되어 동작을 하므로 낮은 확률로 나타나는 다수의 예외적인 콘텐츠 요청에 쉽게 영향을 미쳤으며, 그 결과가 그대로 히트율에 반영이 되었다.



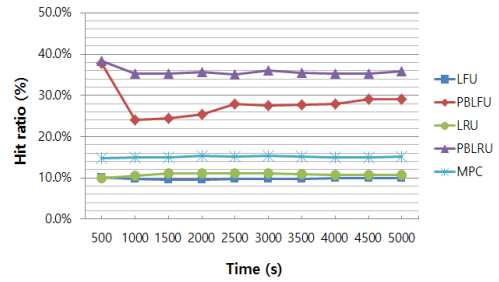
(a) Cache size 0.5%



(b) Cache size 2.5%



(c) Cache size 5.0%



(d) Cache size 10.0%

그림 11. 안정성 결과
Fig. 11. Result of reliability

V. 결론

본 논문에서는 CCN 네트워크의 성능 향상을 위한 방안으로 인기도 기반 캐시 교체 기법을 제안하였다. 특히, 캐시 교체 기법인 LFU 및 LRU 방식을 개선하

여 새롭게 PBLFU, PBLRU를 설계하였다. 제안한 기법은 하위 노드에 확률적으로 콘텐츠를 전달하는 푸싱 동작을 하여 인기 있는 콘텐츠를 하위 노드에 우선적으로 캐싱 하도록 수정하였다. 실험을 통해 제안한 기법이 LFU, LRU, MPC 방식보다 향상된 캐시 히트율을 보여주었으며, 서버의 부하와 RTT 값은 감소시켰다. 하지만 제안한 기법이 안정성 및 강건성 측면에서는 정반대의 결과를 보이므로 이 단점에 대한 보완이 필요하다.

제안한 기법은 다음에 대한 향후 연구가 필요하다. 첫째, 다양한 토폴로지에서의 실험이 요구된다. 본 논문에서는 간단한 트리 구조의 토폴로지에서 실험을 했지만 실제 네트워크 환경은 다양한 토폴로지가 존재할 것이다. 그래서 많은 토폴로지 환경에서의 성능 비교 및 분석이 필요하다. 둘째, Zipf 지수 α 의 값에 따른 결과 도출이다. 이번 실험에서는 하나의 값을 이용하여 결과를 분석하였지만, 실제 네트워크에서의 콘텐츠 요청은 이 지수 값에 따라 많은 변화가 있을 것이므로 지수 값의 변화에 대한 결과도 연구되어야 한다. 셋째, 앞서 언급한 안정성과 강건성을 보완하기 위해 기존 알고리즘을 수정하고, 인기도를 정확히 알고 있는 기법과 다양한 성능 비교를 통해서 제안한 기법의 우수함을 증명해야겠다.

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] B. Davison, "A web caching primer," *J. IEEE Internet Comput.*, vol. 5, no. 4, pp. 38-45, Jul. 2001.

[4] S. Podlipnig and L. Boszormenyi, "A survey of web cache replacement strategies," *ACM Computing Surveys*, vol. 35, no. 4, pp. 374-398, Dec. 2003.

[5] G. Rossini and D. Rossi, "Evaluating CCN multi-path interest forwarding strategies," *ELSEVIER Computer Commun.*, vol. 36, pp. 771-778, Apr. 2013.

[6] N. Laoutaris, S. Syntila, and I. Stavrakakis,

"Meta algorithms for hierarchical web caches," in *Proc. IEEE ICPC*, pp. 445-452, Phoenix, USA, Apr. 2004.

[7] K. Cho, et al., "WAVE: popularity-based and collaborative in-network caching for content-oriented networks," in *Proc. IEEE INFOCOM WKSHPS*, pp. 316-321, Orlando, USA, Mar. 2012.

[8] W. Chai, et al., "Cache less for more in information-centric networks (extended version)," *Computer Commun.*, vol. 36, no. 7, pp. 758-770, Apr. 2013.

[9] C. Bernardini, et al., "MPC : popularity-based caching strategy for content centric networks," in *Proc. IEEE Commun.(ICC)*, 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*, pp. 126-134, New York, USA, Mar. 1999.

[11] A. Afanasyev, I. Moiseenko, and L. Zhang, *Ndnsm: ndn simulator for ns-3*(2012), Retrieved Nov., 24, 2015, from <http://named-data.net/publications/techreports/trndnsm>.

우 태 희 (Taehee Woo)



2006년 : 숭실대학교 정보통신
전자공학부 졸업
2015년 : 국방대학교 컴퓨터공
학과 석사
2015년 1월~현재 : 육군3사관학
교 사이버전학과 순환직 교
수

<관심분야> Next Generation Networking,
Content-Centric Networking, MANET,
Information Assurance, Cyber Warfare

박 흥 순 (Heungsoon Park)



2002년 : 육군사관학교 전산학과 졸업

2007년 : Air Force Institute of Technology 컴퓨터공학과 석사

2015년 : 국방대학교 컴퓨터공학과 박사

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

김 호 길 (Hogil Kim)



1992년 : 육군사관학교 전자공학과 졸업

1996년 : 연세대학교 전자공학 석사

2007년 : Texas A&M 대학교 전산학 박사

1998년~현재 : 육군3사관학교 사이버전학과 교수

<관심분야> Wireless Sensor Network, Computer Architecture, Information Security, Cyber Warfare