

콘텐츠 중심 네트워킹을 위한 인기도 기반 캐싱 기법

권봉용^o 홍충선*
 경희대학교 컴퓨터공학과
 {by1130,cshong}@khu.ac.kr

Popularity-based Caching Scheme for Content Centric Networking

BongYong Kwon^o ChoongSeon Hong*
 Kyung Hee University

요약

콘텐츠 중심 네트워킹은 현재 활발히 진행되고 있는 미래 인터넷 연구 중 하나이며, 네트워크망을 구성하는 라우터들이 자신을 지나는 패킷을 캐싱하고, 콘텐츠 이름을 기반으로 라우팅이 가능한 미래인터넷 아키텍처이다. 본 논문에서는 패킷이 지나는 경로에 있는 모든 라우터에서 패킷을 캐싱하는 기존의 기법과 다른, 인기 콘텐츠의 캐싱 비율을 높이고 클라이언트에 가까워질수록 가중치를 증가시켜서 캐싱하는 효율적인 기법을 제안한다.

1. 서론

지금까지 널리 사용되고 있는 인터넷은 IP 주소를 이용하여 송신자와 수신자 간에 1:1로 서비스를 요청하고 제공받는 구조이다. 그러나 인터넷 TV, 음성/화상 통신, 사물통신(Internet of Things:IoT) 등 인터넷을 이용한 다양한 분야와 용도로 사용하고 있는 시점에서 1:1구조의 인터넷은 주소체계 고갈로 인한 확장성, 보안성, 및 이동성 등에서 한계점을 가진다[1].

인터넷의 한계를 극복하기 위해 미래 인터넷에 관한 연구가 진행되고 있으며, 콘텐츠 중심 네트워킹(CCN: Content Centric Networking)은 현재 활발히 진행되고 있는 혁신적 접근 방식의 미래 인터넷 연구 중 하나이다 [2]. CCN은 자신을 거쳐 간 패킷을 저장할 수 있는 캐싱 기능을 가진 라우터로 구성된 네트워크망에서 콘텐츠의 이름을 기반으로 라우팅이 가능하여 동일한 콘텐츠의 반복 전송을 피하고[2], 네트워크 부하를 감소시켜 사용자의 서비스 질을 향상시킬 수 있는 미래인터넷 아키텍처이다.

현재 CCN에 대한 다양한 연구들이 진행 중이며, 본 논문에서는 인기 콘텐츠의 캐싱 비율을 높이고 클라이언트에 가까워질수록 가중치를 증가시켜서 캐싱하는 효율적인 기법을 제안한다.

2. 관련연구

2.1 CCN(Content Centric Networking)

CCN은 V. Jacobson이 제안하고 PARC 연구소에서 연구가 진행되고 있는 미래 인터넷 아키텍처이다[3]. CCN에서는 데이터를 식별하는 이름을 가진 Interest 패킷을 유저가 보냄으로써 데이터 전달 과정이 시작된다. 즉

Interest 패킷을 보냄으로써 콘텐츠를 요청을 하고, 해당 콘텐츠를 가진 서버 혹은 중간 라우터가 Data 패킷으로 응답하는 것이다. Interest 패킷이 요청된 데이터를 가진 노드 혹은 콘텐츠 서버에 도착할 때, Data 패킷은 Interest 패킷에 의해 만들어진 경로를 추적해서 그 역경로로 보내진다.

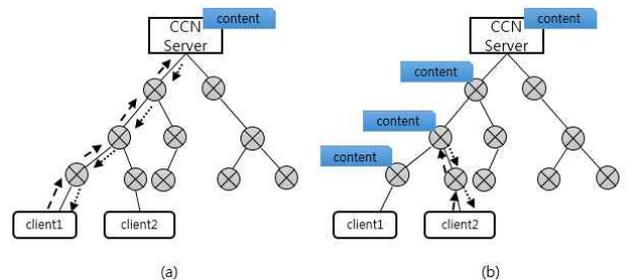


그림 1. CCN Routing

그림 1에서 볼 수 있듯이, Client1이 Interest 패킷(그림 1 파선)을 보낸 후 Data패킷을 받으면(그림1 점선)(a), 패킷이 지나간 중계라우터들은 패킷을 Content Store(CS)에 저장하게 되고, Client2가 Client1과 같은 콘텐츠를 요청하면 라우터가 CS에 저장하고 있는 패킷을 전송하게 된다(b). 이러한 CCN의 특성으로 인해 동일한 데이터를 수많은 사용자 수만큼 반복 전송되는 방식의 현재의 인터넷보다 높은 효율성을 가지게 된다. CCN 기술은 서버가 데이터를 전송한다는 것에 제한을 두지 않고, 데이터를 가지고 있는 라우터도 데이터를 전송할 수 있게 함으로써, 빠른 데이터 전송과 데이터가 네트워크에서 반복되어 전송되는 횟수를 줄이는 장점을 가진다.

3. 제안사항

본 연구는 미래창조과학부가 지원한 2014년 정보통신/방송(ICT) 연구개발사업의 연구결과로 수행되었음. *Dr. CS Hong is corresponding author.

Basic CCN에서는 클라이언트가 서버 혹은 라우터에게 콘텐츠를 요청하는 경우, Data 패킷을 그 경로에 있는 모든 라우터의 Content Store(CS)에 저장하게 되며, 각각의 라우터에 중복 저장된 만큼 CS를 비효율적으로 사용하게 된다. 현재 CS를 효율적으로 사용하기 위한 확률적 캐싱[4], 점진적 캐싱 기법[5]등 다양한 연구가 진행되고 있으며, 본 논문에서는 클라이언트에 가까운 라우터일수록 높은 가중치를 주기 위한 TSI(Time Since Inception), TSB(Time Since Birth)[4]와 콘텐츠 인기도 테이블(Content Popularity Table:CPT)을 이용한 인기도 기반 캐싱 기법을 제안한다.

3.1 캐시 가중치(Cache Weight)

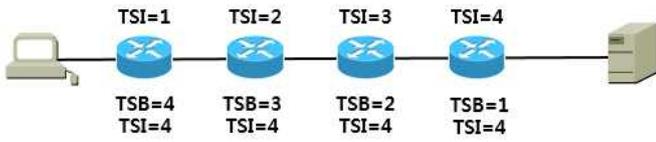


그림 2. TSI, TSB Operation

TSI 필드는 Interest 패킷과 Data 패킷의 헤더에, TSB 필드는 Data 패킷의 헤더에 포함이 되어 그림 2와 같이 TSI는 Interest 패킷이 서버 혹은 Data 패킷을 가지고 있는 라우터에 도착할 때까지 각 라우터를 지날 때 마다 1씩 증가하며 서버/라우터에 도착한다. 그 후, 서버/라우터는 고정된 TSI와 클라이언트에 도착할 때까지 증가하는 TSB 필드를 가지는 Data 패킷을 전송하게 되며, 이 TSI와 TSB를 이용해 식1을 계산하여 각 라우터에서의 캐시 가중치를 계산한다[4].

$$\text{Cache Weight} = \frac{\text{TSB}}{\text{TSI}}$$

식 1. 캐시 가중치

3.2 콘텐츠 인기도 테이블(Content Popularity Table)

표 1. 콘텐츠 인기도 테이블

Content Name	I-FWD	D-Hit	Popularity
/contents/a	3	4	
/contents/b	4	4	

표 1에서 볼 수 있듯이 CPT는 Content Name, I-FWD, D-Hit으로 구성되며, I-FWD와 D-Hit는 각 콘텐츠의 첫 번째 청크(ChunkID가 0인)로 집계를 한다. Content Name은 콘텐츠의 이름, I-FWD는 라우터에서 Interest 패킷에 매칭되는 Data 패킷이 없는 경우 다른 라우터에게 전송한 Interest 패킷의 수, D-Hit는 라우터에 캐싱된 Data 패킷을 전송한 수이며, 이 두 개의 값들을 통해 식 2를 계산하여 인기도를 구한다.

그림 3은 라우터가 Interest 패킷을 받았을 때 CPT의 변화를 보여준다. 라우터 A가 콘텐츠 a/0(콘텐츠 a의 ChunkID가 0인 청크)에 대한 요청메시지(Interest 패킷)를 받으면, 요청된 콘텐츠가 CS에 저장되어 있는지, CPT 리스트에 있는지 확인을 한다. 만약 CS와 CPT에 콘텐츠 a가 존재하지 않으면(그림 3-1) CPT에 콘텐츠 a를 추가

한 후, 콘텐츠 a의 I-FWD 값을 1을 증가시키고 라우터

$$\text{Popularity} = \frac{\text{I - FWD and D - Hit of content}}{\text{total number of I - FWD and D - Hit}}$$

식 2. 인기도

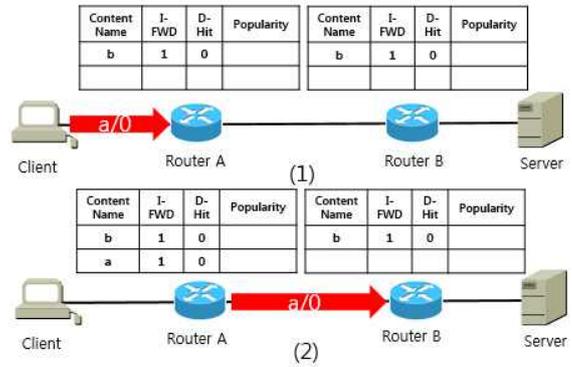


그림 3. Interst 패킷을 받은 경우의 CPT

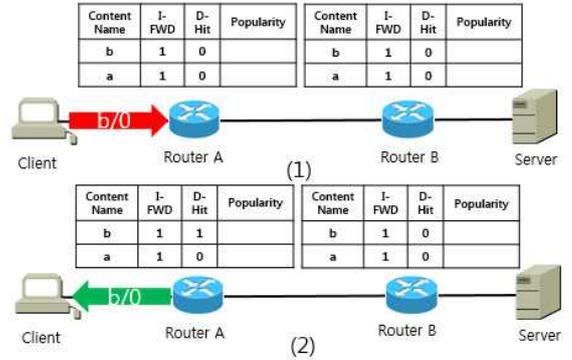


그림 4. Data 패킷을 받은 경우의 CPT B로 요청메시지를 전달한다(그림 3-2). 그림 4는 Data 패킷을 주고받을 때 CPT의 변화를 보여준다. 라우터 A가 b/0 패킷의 요청 패킷을 받았는데, 콘텐츠 b/0이 존재하면(그림 4-1) 콘텐츠 b의 D-Hit 값을 1 증가 시킨 후, Data 패킷을 전송한다(그림 4-2). 라우터가 Data 패킷을 받으면, CPT에서 그 콘텐츠의 인기도와 Data 패킷에서 TSI, TSB를 확인하여 이를 기반으로 그 패킷의 캐싱 유무를 결정 한 후, 다음 라우터 혹은 클라이언트로 패킷을 전송한다.

Alg1. Router receive Interest packet

```

1: if chunkID of Content is 0 then
2:   if content name not exist at CPT then
3:     add content name to CPT
4:   end if
5:   if chunk not exist at CS then
6:     I-FWD is increased by 1
7:     TSI is increased by 1
8:     send interest packet to other router
9:   else
10:    D-Hit is increased by 1
11:    TSB is increased by 1
12:    send Data packet
13:  end if
    
```

```

14: else
15:   if chunk not exist at CS then
16:     TSI is increased by 1
17:     send interest packet to other router
18:   else
19:     TSB is increased by 1
20:     send Data packet
21:   end if
22: end if
    
```

본 논문에서 제안하는 알고리즘은 라우터에서 Interest 패킷을 받을 때(알고리즘1)와 Data 패킷을 받을 때(알고리즘2)로 나누어진다.

먼저 라우터가 Interest 패킷을 받으면(알고리즘 1) 요청된 패킷이 콘텐츠의 첫 번째 패킷인지 확인을 한다. 만약 첫 번째 패킷이면, CPT에서 콘텐츠 이름의 유무를 확인하고, 없으면 추가를 한다. 그 후 CS에 캐싱되어 있는지 확인을 한 후, CS에 없으면 FWD와 TSI를 1 증가시키고 다른 라우터로 플로딩하며, CS에 캐싱되어 있으면, D-Hit과 TSB를 1 증가시키고 Data 패킷을 전송하게 된다. 또한, 요청된 패킷이 첫 번째 패킷이 아니라면, I-FWD와 D-Hit 증가시키는 것을 제외하고 위의 동작을 반복한다.

Alg2. Router receive Data packet

```

1: if CS is full then
2:   apply LRU policy
3:   if no content name x at CS then
4:     content name x is eliminated at CPT
5:   end if
6:   if popularity of content x is greater then or
7:   equals to highest popularity × weight then
8:     cache Data packet
9:   end if
10: TSB is increased by 1
11: send Data packet
    
```

라우터가 Data 패킷을 받으면(알고리즘 2) CS가 꽉 차 있는지 확인하고, 꽉 차있으면 LRU 정책을 이용하여 공간을 만들며, 이때 지워진 체크가 CS에 캐싱되어있던 콘텐츠의 마지막 체크이면, CPT에서 콘텐츠의 이름을 삭제한다. 그리고 받은 패킷의 인기도가 캐싱 가중치와 가장 높은 인기도 곱한 값보다 크거나 같으면 그 패킷은 캐싱이 되며, 그 후 TSB를 1 증가시키고 Data 패킷을 전송하게 된다.

4. 성능평가

본 절에서는 CCN의 캐싱 기법과 본 논문의 제안한 기법의 평균 캐시 적중률을 비교하여 어느 정도 성능이 향상되었는지를 OmNET++4.4.1과 ccnSim-0.2를 이용해 평가하였다. 각각 200개의 체크를 캐싱 할 수 있는 30개의 노드로 구성된 이진트리 구조의 토폴로지에서 Mzip가 1.5이고 50개의 체크로 분할된 10개의 콘텐츠를 16개의

클라이언트에서 요청하는 실험을 하였다.

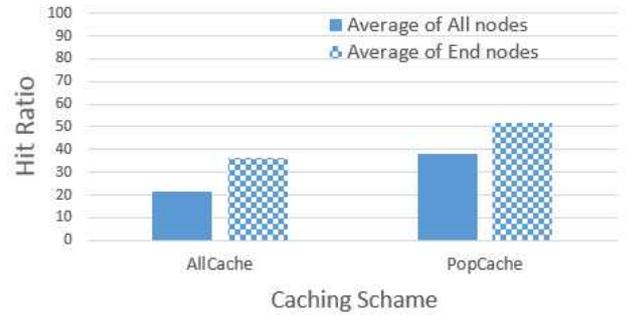


그림 5. 캐시 적중률

그림 5는 CCN에서 사용하는 캐싱 기법과 본 논문에서 제안하는 인기도 기반 캐싱 기법의 캐시 적중률을 보여준다. 실험 결과 평균 적중률에서는 약 17%, 종단 노드들의 평균 적중률에서는 약 18%의 성능이 향상되었으며 표본의 수를 증가시킬 경우, 캐시 적중률이 증가될 것이라 예상된다.

5. 결론 및 향후계획

본 논문에서는 패킷이 지나는 경로에 있는 모든 라우터에서 패킷을 캐싱하는 기존의 기법과 다른, 인기 콘텐츠의 캐싱 비율을 높이고 클라이언트에 가까워질수록 가중치를 증가시켜서 캐싱하는 효율적인 기법을 제안하였다.

향후 최적화 된 표본의 수와 콘텐츠의 인기도에 대해 연구하고, CCN Streaming에 적용을 해서 본 연구가 스트리밍 서비스의 효율성에 어느 정도 영향을 미치는지에 대한 연구를 고려하고 있다.

참고 문헌

- [1] 변성혁, “미래인터넷 아키텍처 연구동향”, 전자통신동향분석, 제 24권, 제 3호, 1-12쪽, 2009년 6월
- [2] 김정임, 정희영, 박우구, “콘텐츠 중심의 네트워크 기술”, 전자통신동향분석, 제25권, 제6호, 2010년 12월
- [3] V. Jacobson, D. K. Smetters, J. D. Thronton, M. F. Plass, N. H. Briggs, and R. L. Braynard, “Networking Named Content,” in Proc. ACM CoNEXT, 2009.
- [4] Psaras, Ioannis, Wei Koong Chai, and George Pavlou. “Probabilistic in-network caching for information-centric networks” Proceedings of the second edition of the ICN workshop on Information-centric networking. ACM, 2012.
- [5] Kideok Cho, Munyoung Lee, Kunwoo Park, Ted Taekyoung Kwon, Yanghee Choi, and Sangheon Pack, “WAVE: Popularity-based and Collaborative In-network Caching for Content-Oriented Networks,” in Proc. IEEE INFOCOM Workshop on Emerging Design Choices in Name-Oriented Networking (NOMEN), March 2012