

콘텐츠 중심 네트워킹에서 캐시 적중률 향상을 위한 파일 확장자 인지 캐싱 구조

이진원[○], 홍충선*
경희대학교 컴퓨터공학과
{notwonz, cshong}@khu.ac.kr

Filename- Extension Aware Caching Architecture for Enhancing Cache Hit Ratio in Content Centric Networking

Jinwon Lee[○], ChoongSeon Hong*
Department of Computer Science and Engineering, Kyung Hee University

요 약

최근 인터넷 기술과 어플리케이션의 발전으로 사용자들의 콘텐츠에 대한 요청이 급격하게 증가하였다. 콘텐츠를 얻기 위해 서버와 같은 하나의 콘텐츠 생산자의 IP 주소로 접근하는 사용자가 많아짐에 따라 TCP/IP 통신은 병목현상과 불필요한 반복전송 등의 문제점들을 가지게 되었다. TCP/IP 통신의 문제점을 해결할 수 있는 대안으로 IP주소 대신 콘텐츠 이름을 사용하고, In-Network Cache를 지원하는 콘텐츠 중심 네트워킹(CCN)이 주목받고 있다. 본 논문에서는 CCN 노드가 CCN Packet 수신 시 파일 확장자를 기반으로 Cache에 저장되어 있는 콘텐츠들을 탐색, 교체하여 콘텐츠 탐색 시간을 줄이고 전체 네트워크의 Cache 적중률을 향상시킬 수 있는 Caching 구조를 제안한다. 그리고 기존 CCN과의 시뮬레이션을 통해 제안하는 구조의 타당성을 확인한다.

1. 서 론

최근 몇 년 사이에 유, 무선 네트워크 기술과 스마트폰, 태블릿 PC와 같은 단말들의 발전으로 인터넷 트래픽은 폭발적으로 증가하였다. 또한 수백, 수천만의 사용자들은 웹페이지, 비디오와 같은 동일한 서비스를 요구하는 특징을 가지게 되었다. 하지만 현재의 인터넷은 1970년대 이동성이 없는 분산된 사용자들에게 서비스를 제공하기 위해 개발된 IP주소 기반의 TCP/IP 통신을 사용한다. TCP/IP 통신방식은 다수의 사용자들이 동일한 콘텐츠를 얻기 위해서는 콘텐츠 서버와 같은 하나의 IP주소에 접근해야하고 이는 불필요한 위치탐색 및 반복전송, 병목현상 등의 문제점들을 발생시킬 수 있다.

따라서 새로운 미래 네트워크 구조의 필요성이 대두되었고 TCP/IP 통신의 문제점들을 해결할 수 있는 콘텐츠 중심 네트워킹(Content Centric Networking, 이하 CCN)이 Van Jacobson에 의해 제안되었다[1].

CCN은 In-Network Cache를 지원하며 Cache의 성능이 CCN의 성능과 직접적으로 연관되어 있다. 기존의 Cache 관리정책에서는 콘텐츠를 탐색하거나 교체할 때

콘텐츠를 구분하지 않으며 하나의 Cache에서 일괄적으로 처리한다. 하지만 해당 방식으로 콘텐츠를 탐색할 때는 순차적으로 탐색하기 때문에 지연이 발생할 수 있고 교체 시에는 용량이 큰 콘텐츠와 용량이 작은 콘텐츠가 교체될 수 있어 교체의 공정성이 이뤄지지 않는다. 따라서 사용자의 Cache 적중률이 감소하게 된다. 본 논문에서는 사용자의 Cache 적중률을 향상시키기 위해 파일 확장자를 바탕으로 Cache를 나누고, 콘텐츠를 탐색하거나 교체할 때 해당 Cache에서 처리하여 탐색의 지연을 줄이고 공정한 콘텐츠 교체를 이룰 수 있는 Cache 구조를 제안한다.

2. 관련 연구

2.1 콘텐츠 중심 네트워킹(Content Centric Network)

TCP/IP 통신은 IP 주소를 사용하여 콘텐츠를 획득하지만 CCN에서는 하나의 콘텐츠를 여러 개의 Chunk로 나누고 각 Chunk에 식별 가능한 고유한 이름을 부여하기 때문에 사용자는 해당하는 콘텐츠 Chunk 이름을 사용하여 콘텐츠를 획득한다.

CCN은 2종류의 Packet을 사용하는데, 콘텐츠를 요청할 때 사용하는 Interest Packet, 요청에 응답하고 콘텐츠를 전달할 때 사용하는 Data Packet이 있다. 또한 각 CCN 노드들은 콘텐츠를 저장하는 Content Store(이하 CS),

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (B0101-16-0033, CCN 기반의 다차원 scalability를 활용한 5G 이동통신 기술 연구개발)

*Dr. CS Hong is the corresponding author

콘텐츠를 요청하는 Interest Packet의 기록을 저장하는 Pending Interest Table(이하 PIT), Interest Packet의 Forwarding을 위한 Forwarding Information Base(이하 FIB)라는 3종류의 In-Network Cache들로 구성되어 있다. 사용자는 콘텐츠의 이름이 담긴 Interest Packet을 전송하여 콘텐츠를 요청하고 Interest Packet을 수신한 CCN 노드는 요청받은 콘텐츠의 이름을 바탕으로 CS를 탐색한다. CS에 콘텐츠가 존재하면 요청받은 Interface로 Data Packet을 전송하고 CS에 콘텐츠가 존재하지 않으면 PIT, FIB로 Interest Packet을 Forwarding한다. 언급한 바와 같이 CCN은 CS, PIT, FIB와 같은 In-Network Cache를 지원하기 때문에 사용자들은 콘텐츠 서버와 같은 특정 IP주소에 접근하지 않아도 해당 콘텐츠를 Cache하고 있는 중간 노드에서 콘텐츠를 획득할 수 있어 TCP/IP 방식보다 빠르고 효율적인 통신을 가능하게 한다[2].

2.2 Cache 교체정책

CCN 노드는 기본적으로 노드를 거쳐 가는 모든 콘텐츠를 Cache한다. 이 때, 더 이상 콘텐츠를 저장할 공간이 없다면 CCN 노드는 CS에 새로운 콘텐츠를 저장하기 위해 CS에서 방출할 콘텐츠를 선별하는데 이를 Cache 교체정책이라고 한다. Cache 교체정책은 자주 요청되어지는 콘텐츠를 CCN 노드에 Cache하여 네트워크 대역폭과 서버의 부하를 감소시키며 사용자로 하여금 콘텐츠를 빠르게 얻게 한다. Least Recently Used(이하 LRU), First In First Out(FIFO), Least Frequently Used(LFU) 등의 Cache 교체정책이 널리 사용되고 있지만 좀 더 효율적인 Cache 교체정책을 찾기 위한 연구가 활발하게 진행되고 있다 [3],[4].

3. 기존 연구의 문제점 및 제안사항

그림 1은 기존 CCN과 본 논문에서 제안하는 구조의 CCN Packet Forwarding 방식을 나타낸다. 기존 CCN에서는 CCN 노드가 CCN Packet 수신 시 하나의 CS에 저장되어 있는 모든 콘텐츠와 비교하기 때문에 콘텐츠를 탐색하는 시간이 길어지고, 콘텐츠 교체 시에도 다른 확장자를 가진 콘텐츠와 교체될 수 있기 때문에 불공정한 교체가 이루어진다. 예를 들어, 크기가 큰 비디오 콘텐츠는 특성상 인기가 있는 콘텐츠라도 단위 시간동안 요청 횟수가 적을 수 있고 크기가 작은 문서 콘텐츠는 단위 시간동안 많은 요청을 받을 확률이 높다. 따라서 LRU Cache 교체 정책에 의해서 인기가 있지만 요청 횟수가 적은 비디오 콘텐츠와 요청 횟수가 많은 문서 콘텐츠를 교체한다면 기존에 비디오 콘텐츠를 요청하던 사용자는 Cache 적중률이 감소하게

된다.

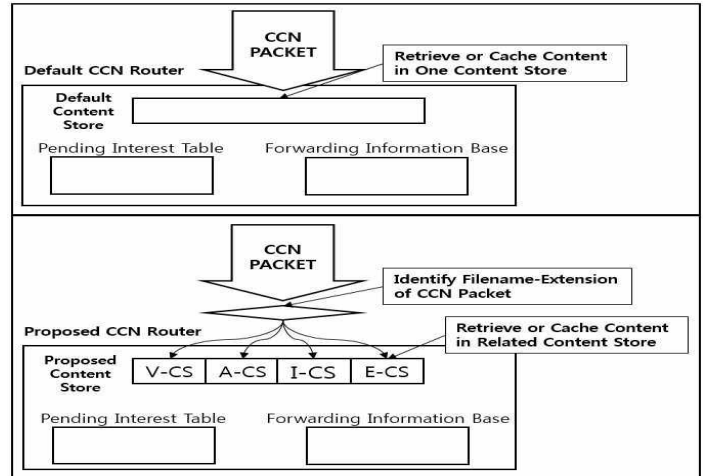


그림 1. 기존 CCN과 제안하는 구조의 CCN Packet Forwarding 비교

제안하는 구조에서는 표 1에 따라 CS를 4종류로 나누어, CCN 노드가 CCN Packet 수신 시 확장자를 식별한 뒤 해당 CS에서 처리한다. 제안하는 구조는 확장자로 대표되어지는 비슷한 콘텐츠끼리 교체하여 공정한 콘텐츠 교체를 가능하게 한다.

표 1. 제안하는 Content Store 구조

Video-CS	Audio-CS	Image-CS	Etc-CS	
.avi	.aac	.bmp	.doc	.alz
.fla	.cda	.dib	.htm	.apk
.fli	.m3u	.img	.html	.app
.flm	.mp2	.jpe	.hwp	.exe
.gdb	.mp3	.jpeg	.pdf	.jar
.mp4	.wav	.jpg	.php	.js
.mpeg	.ogg	.lcd	.php3	.rar
.mpg		.png	.phtml	.zip
.pls		.tif	.ppt	.xls
		.tiff	.txt	

알고리즘 1은 CCN 노드가 Interest Packet에 포함된 확장자를 식별하여 해당하는 CS에서 콘텐츠 탐색을 진행하는 과정을 보여준다. CS에서 매칭되는 콘텐츠를 찾지 못하면 기존 CCN과 같이 PIT와 FIB를 탐색한다.

Algorithm 1 Filename-Extension based Interest Forwarding

```

1: On Interest Packet Arrival:
2: Extract Filename-Extension(FE) from the Interest
3: Search FE's Content Store(CS) with requested Content Name
4: if the requested Content is found then
5:   Send the Data to requesting Client
6: end if
7: Check in PIT
8: else if Request have been already Forwarded then
9:   Requested content's arrival face is Added on the PIT
10: end if
11: else if Matching entry in the FIB then
12:   Forward the Interest to a list of faces
13: end if
14: else
15:   Discard the Interest
16: end if
    
```

알고리즘 2는 CCN 노드가 Data Packet의 확장자를 식별하여 해당하는 CS에서 Cache결정 알고리즘을 수행하는 과정을 보여준다. CS에 새로운 콘텐츠를 저장할 공간이 부

족할 경우 기존 CCN과 같이 LRU Cache 교체정책으로 기존의 콘텐츠와 교체한다.

Algorithm 2 Filename-Extension based Cache Decision

```

1: On Data Packet Arrival:
2: Extract Filename-Extension(FE) from the Data
3: Search FE's Content Store(CS) with requested Content Name
4: if FE's CS is not full then
5:   Cache the Content
6: end if
7: else
8:   Replace according to LRU
9: end if
    
```

4. 성능평가

4.1 시뮬레이션 환경

제안하는 CCN 구조의 시뮬레이션을 위해 OMNET++ 기반의 시뮬레이터인 ccnSim(0.3)을 사용하였다. 그림 2는 시뮬레이션을 위한 네트워크 토폴로지를 보여준다. 콘텐츠 서버와 10개의 CCN 노드, 40개의 Client로 구성되었다. 콘텐츠 서버는 100개(각 1 Chunk)의 콘텐츠를 가지고 있으며 각 Client는 Zipf distribution 분포에 의해 무작위로 콘텐츠를 요청한다.

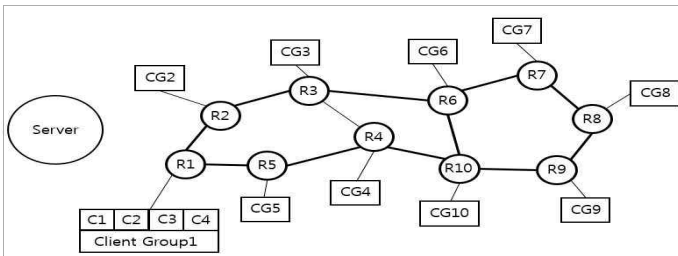


그림 2. 네트워크 토폴로지

Packet Forwarding 전략과 Cache 교체정책, Cache 결정 알고리즘은 NRR1(Nearest Neighbor Routing), LRU, LCE(Leave Copy Everywhere)를 적용하였다.

4.2 시뮬레이션 결과

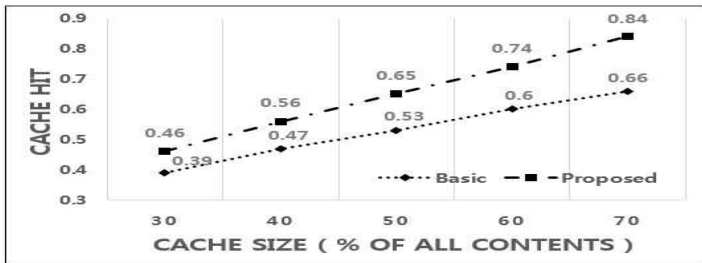


그림 3. Cache 크기에 따른 Cache Hit- Ratio 비교 (Zipf distribution parameter $\alpha = 0.8$)

그림 3, 4는 고정된 Zipf parameter(0.8, 1.2) 값에 대한 Cache 크기에 따른 Cache 적중률의 변화를 보여준다. 기존의 CCN은 하나의 CS 안에서 LRU를 바탕으로 한 콘텐츠 간 불공정한 교체가 이루어지기 때문에 Cache 적중률이 제안하는 구조보다 낮은 폭으로 증가했다. 제안하는 구

조는 특정 확장자 CS안에서 비슷한 확장자를 가진 콘텐츠 간의 공정한 교체가 이루어지기 때문에 다른 CS에 저장되어있는 콘텐츠들의 Cache 적중률에 영향을 끼치지 않는다. 결과적으로 제안하는 구조는 기존 CCN보다 전체 네트워크의 Cache 적중률을 향상시켰으며 그림 3에 비해 그림 4에서는 Zipf parameter 값이 증가함에 따라 Cache 적중률이 더 크게 증가함을 보였다.

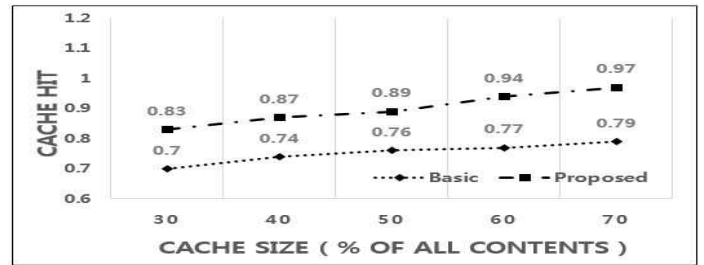


그림 4. Cache 크기에 따른 Cache Hit- Ratio 비교 (Zipf distribution parameter $\alpha = 1.2$)

5. 결론

본 논문에서는 TCP/IP 통신에서 발생할 수 있는 병목현상, 불필요한 반복전송 등의 문제점들을 해결할 수 있는 새로운 네트워크 패러다임으로 주목받는 콘텐츠 중심 네트워킹에서, 콘텐츠의 확장자에 따라 Content Store를 나누어 콘텐츠를 탐색, 교체함으로써 콘텐츠 탐색시간을 줄이고 전체 네트워크의 Cache 적중률을 향상시킬 수 있는 파일 확장자 인지 Caching 구조를 제안하였고 시뮬레이션을 통해 기존 CCN과 비교하여 20% 이상의 Cache 적중률의 향상을 확인하였다.

참고 문헌

[1] Van Jacobson, et al. "Networking Named Content", ACM CoNEXT 2009, pp. 1-12, Dec. 2009

[2] Xingliang Yuan, Jianping Wang, "Enabling Secure and Efficient Video Delivery Through Encrypted In-Network Caching", IEEE Journal on Selected Areas in Communication, vol. 34, no. 8, pp. 2077-2090, Aug. 2016

[3] Yonggong Wang, et al. "Design and Evaluation of the Optimal Cache Allocation for Content-Centric Networking", IEEE Transactions on Computers, pp. 95-107, Sep. 2015

[4] Kyi Thar, Choong Seon Hong, et al. "Applying Multi-Armed Bandit Problem into Cache Decision Algorithm for Content Centric Networking", Korea Computer Congress, pp. 1180-1182, June 2016