

콘텐츠 중심 네트워크를 위한 효율적인 자료구조

손동현, 최대진, 김동현, 임영빈, 한진영, 권태경*, 최양희*

서울대학교 컴퓨터 공학부

{dhson, dhkim, djchoi, ybim, jyhan}@mmlab.snu.ac.kr, *{tkkwon, yhchoi}@snu.ac.kr

An Efficient Data Structure for Content-Centric Networks

Donghyun Son, Daejin Choi, Donghyun Kim, Youngbin Im, Jinyoung Han,

Ted “Taekyong” Kwon, Yanghee Choi

School of Computer Science and Engineering, Seoul National University

요약

인터넷의 구조는 수십 년 동안 거의 동일하나, 사용자의 관점이 호스트 중심에서 콘텐츠 중심으로 변화했다. 이에 따라, 호스트 주소로 통신하는 기존의 인터넷을 탈피하여, 콘텐츠의 이름으로 통신하는 Content Centric Network (CCN) 이라는 새로운 패러다임이 소개되었으며 지금까지도 많은 연구가 진행되고 있다. CCN에서는 라우터가 수많은 콘텐츠의 요청을 빠르게 처리해야 하기 때문에 패킷을 빠르게 처리하는 것이 필수적이다. 이를 위해서는 CCN 라우터에서 핵심적인 역할을 하는 FIB, PIT, CS의 처리속도가 매우 중요하다. 하지만 지금까지의 CCN 연구에서는 CCN 라우터의 FIB, PIT, CS의 효율적인 처리를 위한 자료구조에 대해서는 다루지 않았다. 따라서 본 논문에서는, 빠른 CCN을 지원하기 위한 효율적인 자료구조를 제안하고 성능을 검증한다.

I. 서론

현재의 인터넷의 구조는 호스트간 통신을 목적으로 설계되었다. 하지만 오늘날의 인터넷 사용자들은 어떤 호스트가 콘텐츠를 전송하는지에 대해서가 아니라 어떤 콘텐츠를 전송하는가에 대해 주로 관심을 가지고 있다[1]. 이런 관점 변화에 따라, 콘텐츠 중심 네트워크 (Content Centric Network, 이하 CCN)라는 새로운 인터넷 패러다임이 소개되었으며 많은 연구가 진행 중에 있다. CCN은 현재 인터넷과 같이 IP 주소를 기반으로 통신을 하는 것이 아니라 콘텐츠의 이름을 기반으로 해서 통신을 하기 때문에 불필요한 중복 전송이 최소화 되고, 효율적인 통신을 가능하게 한다.

CCN에서는 콘텐츠 요청과 전달을 위해서 Interest packet과 Data packet의 두 가지 유형이 사용된다[1]. Interest packet은 사용자가 특정 콘텐츠를 요청하기 위해 사용되는 packet으로 콘텐츠 배포자를 향해 전달된다. Interest packet을 받은 콘텐츠 배포자는 Data packet에 해당 콘텐츠를 담아 사용자에게 전달하게 된다.

CCN의 핵심적인 역할을 하는 라우터는 Forwarding Information Base (FIB), Pending Interest Table (PIT), Content Store (CS)로 구성된다 [1]. FIB는 Interest packet을 해당 콘텐츠가 있는 곳으로 안내하는데 필요한 테이블로 현재 인터넷의 라우팅 테이블과 그 기능이 유사하다. PIT는 처리 중인 Interest packet이 어느 곳에서 왔는지에 대한 정보가 담겨있는 테이블이다. CS는 전달했던 콘텐츠를 저장하는 곳으로, 향후 CS에 저장된 동일한 콘텐츠에 대한 요청이 들어오는 경우에 해당 콘텐츠를 요청자 쪽으로 바로 전달한다.

CCN에서는 라우터가 수많은 콘텐츠의 요청을 빠르게 처리해야 하기 때문에 Interest packet과 Data packet을 빠르게 처리하는 것이 필수적이다. 이를 위해서는 CCN 라우터에서 핵심적인 역할을 하는 FIB, PIT, CS의 처리속도가 매우 중요하다. 하지만 지금까지의 CCN 연구에서는 CCN 라우터의 FIB, PIT, CS의 효율적인 처리를 위한 자료구조에 대해서는 다루지 않았다. 따라서 본 논문에서는, 빠른 CCN을 지원하기 위한 효율적인 자료구조를 제안하고 성능을 검증한다.

II. 본론

2.1 CCN을 위한 자료구조의 요구사항

CCN의 FIB, PIT, CS를 위해 사용될 수 있는 대표적인 자료구조에는 연결리스트를 이용한 선형구조와 이진탐색트리구조가 있다. 연결리스트를 이용한 선형구조가 적용될 경우 콘텐츠 이름 lookup시 연결리스트의 모든 entry를 검색해야 하기 때문에 시간복잡도가 $O(n)$ 이다. PIT의 경우에는 단순히 lookup만 수행되는 것이 아니라 entry에 대한 update도 이루어지는데, update의 경우에는 연결리스트를 사용하면 $O(1)$ 만 소요된다. 이진탐색트리를 적용하면, lookup의 경우 시간 복잡도가 $O(\log n)$ 이나, 한 쪽으로 치우쳐진 트리가 구성되면 최악의 경우에는 $O(n)$ 까지 될 수가 있다. PIT entry에 대한 update를 위해서는 $O(\log n)$ 가 소요된다. 연결리스트와 이진탐색트리 모두 콘텐츠의 개수인 n 이 커질수록 처리속도가 심각하게 느릴 수 있다. 따라서 빠른 CCN을 위해서는 lookup과 update를 모두 빠르게 처리할 수 있는 CCN에 특화된 효율적인 자료구조가 필요하다.

2.2 CCN을 위한 자료구조 제안: 이중 해시 기반 해시 테이블

본 논문에서는 CCN을 위한 자료구조로 이중해시 기반 해시테이블을 제안한다. 제안한 자료구조는, 하나의 해시테이블과 두 개의 해시를 기반으로 하고, Figure 1에서 동작과정이 묘사되어 있다. lookup의 경우, 대부분 해시한 콘텐츠의 이름에 해당되는 entry를 바로 돌려주면 되기 때문에 $O(1)$ 이 소요될 것이다. 만약 다수의 콘텐츠의 이름이 같은 entry에 해당되게 되면 각 entry에 대한 연결리스트를 검색하면 되는데, 이때 복잡도는 $O(k)$ 가 소요될 것이다. PIT의 update의 경우에도 소요되는 시간복잡도는 이와 유사할 것으로 예상된다.

콘텐츠의 이름을 해시할 때 사용하는 해시함수는 city-hash 64와 spooky 64이다[2]. 해당 해시함수는 비교적 충돌(collision) 확률이 적고 성능이 좋은 것으로 보고되고 있다[2]. 두 개의 해시함수를 사용하는 이유는 해시값과 콘텐츠의 이름이 일대일 대응이 아닐 수 있기 때문에, 동일한

해시값을 가지는 콘텐츠의 경우 연결리스트에서 추가로 검색을 해야 하는데, 이때 가능한 문자열 비교를 피하기 위해 한 개의 해시함수를 추가로 사용해서 연결리스트를 빠르게 검색하도록 한다.

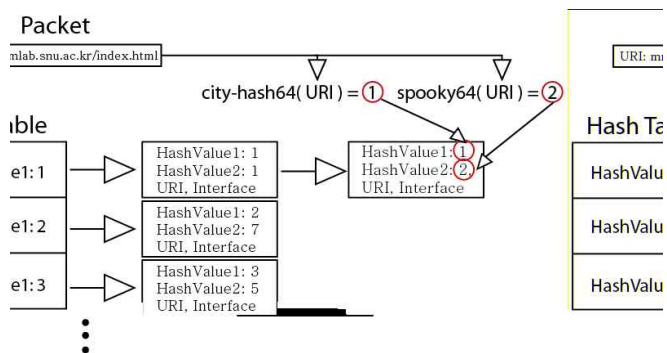


Figure 1. 이중해시 기반 해시테이블 자료구조

2.3 실험

2.3.1 실험환경

리눅스 버전	CentOS release 5.8 (Final)
CPU사양	Intel(R) Xeon(R) i7,
메모리	8G
콘텐츠 요청 패턴	Zipf distribution (parameter = 1.0)
FIB에 배포된 콘텐츠 개수	10K, 100K, 200K

2.3.2 콘텐츠 개수에 따른 FIB lookup 성능비교

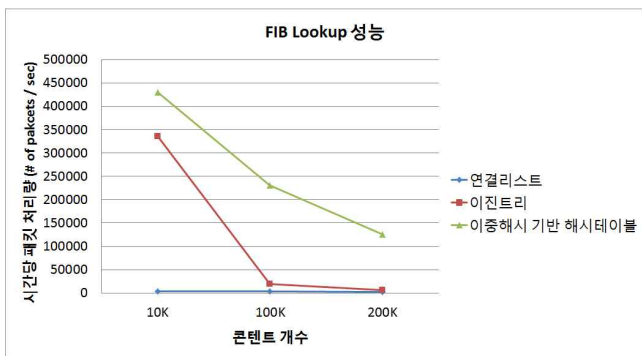


Figure 2. 콘텐츠 개수에 따른 FIB lookup 성능 비교

Figure 2는 제안한 자료 구조와 이진탐색트리로 구성된 트리구조, 그리고 링크드리스트로 구성된 선형구조를 이용하여 구성된 FIB의 entry의 수가 증가할 때 패킷처리량을 비교한 것이다. 여기에서는 Interest 패킷만 생성하였으며, PIT 처리는 제외하였다.

결과에서 볼 수 있듯이, 제안한 자료구조가 다른 두 자료구조에 비해, FIB의 Entry의 수가 증가할수록 더욱 성능이 우월한 것을 보여주고 있다. 연결리스트의 경우에는 처리량이 다른 두 자료구조에 비해 가장 적었고, 이진트리의 경우에는 콘텐츠 개수가 적을 경우에는 많은 처리량을 보여주었지만 콘텐츠 개수가 증가할수록 급격하게 처리량의 감소를 나타내었다. 제안한 자료구조는 콘텐츠 개수가 증가하더라도 계속 많은 처리량을 유지하였다.

2.3.3 Interest와 Data packet에 대한 FIB/PIT의 성능 비교

Figure 3. PIT, FIB에서 시간당 패킷처리량 비교

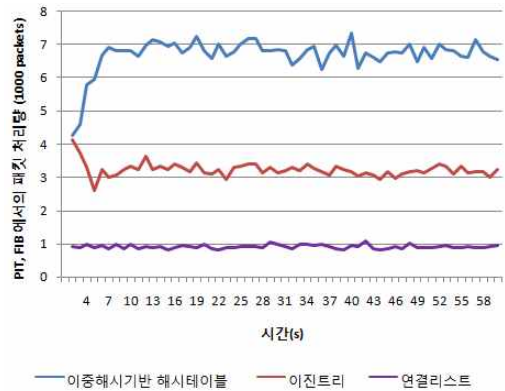


Figure 3은 PIT, FIB에 대한 시간당 패킷의 처리량을 비교한 것이다. 본 실험은 Interest packet과 이에 대응되는 Data packet을 모두 발생하여서 FIB에서는 lookup이 수행되고, PIT에서는 update가 수행되는 복잡한 상황에 대해 검증하였다. 이때, FIB에 배포된 콘텐츠의 개수는 100K 개 이다. 단순 lookup 뿐만 아니라 update가 복합적으로 이루어지는 상황에서도 제안한 자료구조의 성능이 이진탐색트리의 약 2.5배 정도 뛰어난 것을 확인할 수 있었다. 연결리스트에 비해서는 약 7배 이상 많은 패킷을 처리하는 것을 확인할 수 있었다.

III. 결론

본 논문에서는 CCN을 빠르게 지원하기 위한 효율적인 자료구조인, 이중해시 기반 해시테이블을 제안하였다. 제안된 이중해시 기반 자료구조를 사용한다면 해시테이블을 구성하기 위해 소모되는 메모리는 늘어날 수 있겠지만, 메모리 디바이스의 가격이 하락하고 있기 때문에 여기에 관한 문제는 희석될 것으로 보인다. 그 뿐만이 아니라, IP기반 네트워크 보다 콘텐츠의 개수가 많아지는 CCN의 특징을 감안하였을 때, CCN 라우터에서의 빠른 처리가 더욱더 중요해질 것이기 때문에 제안한 자료구조의 적용이 필요하다. 향후 연구에서는 콘텐츠 라우터의 성능을 더욱 최적화 하기 위해 수많은 패킷을 동시에 처리하는 멀티코어 기반의 라우터를 개발할 계획이다. 또한 제안한 자료구조와 멀티코어를 함께 고려한 라우터 아키텍처를 설계해서 효율적인 통합 CCN 라우터 아키텍처를 개발할 계획이다.

ACKNOWLEDGMENT

본 연구는 중소기업청에서 지원하는 2012년도 산학연공동기술개발사업 (No.C0018176)의 연구수행으로 인한 결과물임을 밝힙니다.

참고 문헌

[1] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, R. Braynard, "Networking named content," Proc. ACM CoNEXT, 2009.
 [2] Won so, Ashok Narayanan, Dave Oran, Yaogong Wang, "Toward Fast NDN software Forwarding Lookup Engine Based on Hash Tables," Proc. ACM ANCS, 2012.