

소비자의 처리량 증가를 위한 CCN 프로토콜

최원준*, Ramneek*, 석우진**

*과학기술연합대학원대학교, **한국과학기술정보연구원

*cwj@ust.ac.kr, *ramneek@ust.ac.kr, **wjseok@kisti.re.kr

Throughput improvement for Consumer using Modified CCN Protocol

Won Jun Choi*, Ramneek Sekhon*, Woo Jin Seok**

*Korea University of Science and Technology, **Koera Institute of Science and Technology Information.

요 약

차세대 미래 인터넷으로 관심을 모으고 있는 CCN은 ICN, CDN, NDN과 비슷한 개념으로 출발한다. 즉, 사용자가 관심을 가지는 데이터를 하나의 콘텐츠로 바라보고 네트워크를 한다. CCN은 이러한 콘텐츠에 대해 요청 패킷을 보내면 콘텐츠를 가지고 있는 노드에서는 해당 패킷을 보내는 방식이다. IP 기반의 네트워크에서와 마찬가지로 CCN에서도 한정된 네트워크 대역폭에서 소비자에게 전송되는 패킷의 전송 시간에 따른 처리량 향상은 주요 관심사 중의 하나이다. 본 논문에서는 CCN에서의 CCN-Helper 프로토콜을 사용한 소비자의 다운로드 시간 감소 방법을 제안하여 생산자에서 소비자로의 패킷 전송 처리량을 향상시키고자 한다.

1. 서론

최근 네트워크의 기술이 발전함에 따라 네트워크 IP 기반의 기술이 아닌 콘텐츠의 정보를 사용한 네트워크 기술이 이슈가 되고 있다. 그 대표적인 예로 ICN(Information Centric Network)[1], CDN(Content Delivery Network)[2], NDN(Named Data Network)[3] 등이 있으며, 이러한 기술들은 주로 목적지 기반 주소 네트워크가 아닌 콘텐츠 즉, 데이터를 요구하는 사용자에게 중심이 된 네트워크 구조를 이루고 있다. 콘텐츠 이름으로 네트워크를 실현하는 기술 중에 차세대 미래 인터넷으로 떠오르고 있는 기술 중의 하나인 CCN(Content Centric Network)은 유무선 네트워크에서 원하는 사용자에게 필요한 자료만 전송한다는 기본적인 생각을 현실화 시킨 기술이라고 볼 수 있다. CCN에서는 Interest 패킷과 Data 패킷이 쌍을 이루어 통신하게 된다. 콘텐츠 요청자가 Data를 받기 위한 Interest 패킷을 보내면 그에 해당하는 Data 패킷을 수신하게 되는 flow balance[7] 메커니즘으로 동작한다. 그렇기 때문에 점대점 플로우 제어를 할 필요가 없다.[8] 기존의 end to end 방식이 아닌 hop by hop 방식으로 동작하는 CCN은[5] 사용자가 Interest 패킷을 보내면 데이터를 가지고 있는 네트워크에 있는 노드에서 해당 Data 패킷을 전송하게 되고 중간 CCN 노드의 캐쉬를 거쳐 사용자에게 패킷이 전달되게 된다. 멀티캐스팅과 같은 다양한 포워딩 방식으로 패킷을 전달하는 특징을 가지고 있는 CCN은 데이터를 가지고 있는 노드가 작동하지 않더라도 다른 노드로 대체할 수 있는 flexibility를 가지고 있다.[5] 같은 콘텐츠

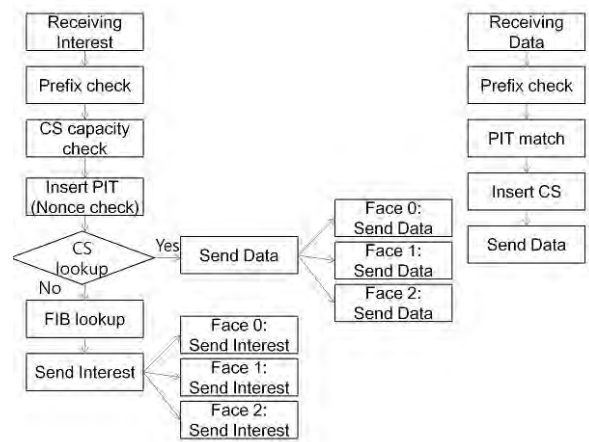
자료이라도 다른 노드에서 수신이 가능하기 때문에 PIT 경로를 변경하면서 유기적으로 데이터를 송수신할 수 있기 때문이다. 사용자가 요구하는 패킷의 크기가 커지게 되면 Interest 패킷 대 Data 패킷의 1:1 매칭 전송 방식이라도 데이터를 청크 단위로 나누어서 전송하게 된다.[4] CCN 노드는 캐쉬(Content Store), PIT(Pending Information Table), FIB(Forwarding Information Base)로 구성된다. 캐쉬는 수요자가 요청한 Interest 패킷에 대한 Data 패킷을 저장해 놓는 저장소이며, PIT 테이블은 수요자에게 요청한 Data를 전달해 주기 위한 매칭 테이블이다. FIB 테이블은 Interest 패킷을 어느 방향으로 전달할지를 결정하는 테이블이다. Interest 패킷이 CCN 노드에 전달되면, 먼저 prefix에 맞는지 확인하고 PIT 테이블에서 Nonce 레코드를 통해서 데이터 중복 여부를 체크한다. 만약 중복이 되지 않으면 캐쉬에서 자료를 찾아보고 있으면 Data 패킷을 전송한 후 PIT 테이블의 관련 기록은 삭제된다. 만약, 자료가 없으면 PIT 테이블에 기록하고 Interest 패킷을 보낼 전송 전략을 FIB 테이블을 통해서 결정한다. Data 패킷이 CCN 노드에 수신되면 prefix를 먼저 확인하고 PIT 테이블에서 Interest 기록과 매치되는지 확인한다. 만약, 매치가 되면 캐쉬에 정보를 기록하고 PIT 테이블의 정보를 확인하여 요청지 노드로 전송된다. 이와 같은 방식으로 동작하는 CCN은 Data 패킷을 요청할 때 Interest 패킷을 주변 CCN 노드에 broadcast 방식으로 전송하게 된다. 또한, 중간에 패킷 손실이 발생하게 되면 Interest 패킷을 다시 broadcast 방식으로 전송하게 되는데 동시에 패킷을 전송하는 것으로 인한 비효율적 전송이 발생할 수 있

고 중간 CCN 노드에서는 이러한 중복 패킷을 처리하는 비용이 발생하게 된다. 따라서, 소비자는 패킷을 받는 시간이 감소하게 되고 전체 패킷 다운로드 처리량이 감소하게 되는 원인이 된다. CCN 에서 소비자의 처리량을 향상시키기 위한 다양한 연구가 진행되었는데 그 예는 다음과 같다. 높은 처리량과 낮은 지연시간을 추구하여 CCN 패킷을 처리하는 방법을 제안한 연구 [6]가 있으며, 사용자에게 수신되는 지연 시간을 감소시키기 위하여 캐쉬 파이프라인을 조절한 연구[9]가 있다. 또한, 포워딩 전략, 캐쉬의 분리, 공평성, 오버로드 제어, 라우터 전략 등의 CCN 방안 연구[10], 캐쉬 PIT 를 혼잡 상황에서 다양하게 조절함으로써 작업 로드를 향상시킨 연구[11] 그리고 다운로드와 업링크의 혼잡을 최소화하여 패킷의 전송 시간을 최소화하여 처리량을 향상시킨 연구[12]가 있다. 본 논문에서는 중간 CCN 노드에서의 비효율적 패킷 전송을 완화하고 소비자에서의 패킷 다운로드 처리량을 증가시키기 위하여 중간 이웃 CCN 노드에서 관련된 패킷 전송을 도와주는 CCN-Helper 프로토콜을 제안하고자 한다.

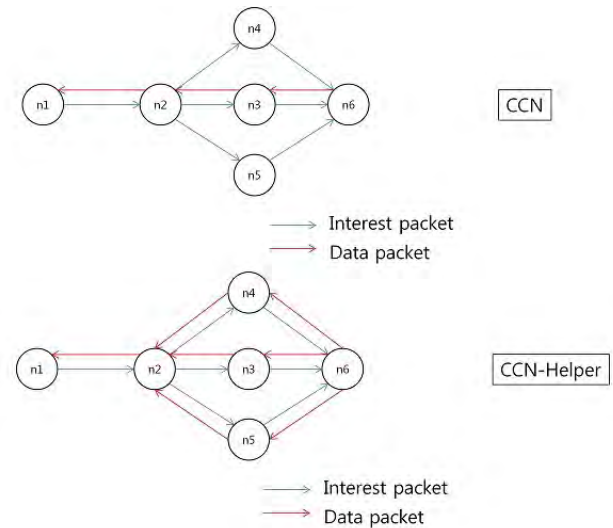
2. CCN-Helper 프로토콜

본 논문에서 제안하는 CCN-Helper 프로토콜은 말 그대로 주변의 CCN 노드에서 Data 청크 파일을 다운로드 하는데 도움을 주는 프로토콜이다. CCN 노드는 수요자가 요청하는 콘텐츠를 청크 단위로 전송하게 된다. 청크 단위의 패킷을 요청하기 위해서 수요자는 Interest 패킷을 네트워크에 전송하게 되고 네트워크에 있는 중간 CCN 노드에서는 Interest 패킷의 정보를 받아서 CS 에 관련 데이터가 없으면 다시 다음 CCN 노드로 포워딩을 한다. 이때, 중복 Interest 패킷일 경우와 dummy 파일이 CCN 노드에 수신될 경우에는 Prefix 확인 단계에서 걸러지게 된다. 하지만, 제안하는 방법은 중복 Interest 패킷이라도 요청하는 청크 데이터의 시퀀스가 다르게 되면 정보를 저장한다. 이렇게 하는 이유는 주변 CCN 노드에서도 같은 prefix 에 대해서 다운로드 처리에 도움을 주게 하기 위해서이다. Interest 패킷이 주변 CCN 노드에 전송되면 속도가 빠른 링크의 Interest 패킷이 먼저 다음 CCN 노드에 도착하게 되고 속도가 느린 링크에 속한 Interest 패킷은 CCN 노드에서 제외되어 처리된다. 기존 방법은 이렇게 네트워크의 속도가 가장 최적화된 링크로만 데이터를 전송한다. 하지만, 네트워크에 혼잡이 발생하게 되면 오히려 처리량의 저하를 불러올 수도 있다. 이럴 때에는 주변 CCN 노드에서 CS 에 여유용량이 허용된다면 해당 prefix 를 다운로드하는데 도움을 주도록 설정할 수 있다. 만약, 주변 CCN 노드에서 Data 패킷을 생산자로부터 다운로드 하는데 도움을 준다면 소비자가 수신하는 Data 패킷의 처리량은 증가하게 될 것이다. 그림. 1 은 CCN 노드에서 데이터 패킷을 수신하는 경우를 나타낸 것이다. CCN 노드에 Interest 패킷이 수신되면 prefix 와 CS capacity 를 확인하고 CS 에 관련 데이터가 있으면 주변 CCN 노드에 청크 데이터를 나누어서 전송하게 된다. 만약, 관련

데이터가 없다면 주변 CCN 노드에게 Interest 패킷을 전송한다. CCN 노드에서 Data 패킷이 수신되면 Interest 패킷에 명시된 reverse path 로 Data 패킷을 전송한다.



(그림. 1) CCN 노드에서 데이터 패킷 수신 경우



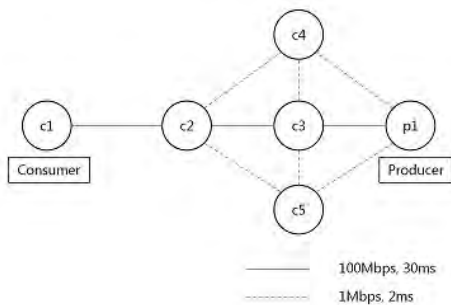
(그림. 2) CCN 과 CCN-Helper 의 비교

그림. 2 는 CCN 의 기본 forwarding 전략과 제안하는 CCN-Helper 의 전략을 비교한 것이다. CCN 에서는 주변 CCN 노드에 Interest 패킷을 broadcast 방식으로 전송한다. 만약, n3 CCN 노드에서 보낸 Interest 패킷이 n6 에 먼저 도착하여 Data 패킷을 전송하기 위한 reverse path 가 n6->n3->n2->n1 으로 설정되었다면 다음 청크 데이터도 같은 path 를 사용하게 될 것이다. 하지만, 네트워크 환경에 따라 링크가 불안정하게 되면 수요자가 받는 데이터의 수신 시간이 감소하여 처리량이 저하되는 원인이 될 것이다. CCN-Helper 에서는 Interest 패킷을 다음 CCN 노드에게 전송할 경우에 CCN 과 같은 방식으로 전송하게 되는데 생산자 노드 n6 에서 같은 prefix 에 대해서 예외 처리를 하지 않고 시퀀스가 다른 청크 데이터를 나누어서 분산하여 전송함으로써 n2 노드와 n1 이 단위 시간 당 받는 청크 데이터의 크기를 향상시켜 수요자 노드 n1 에서의 처

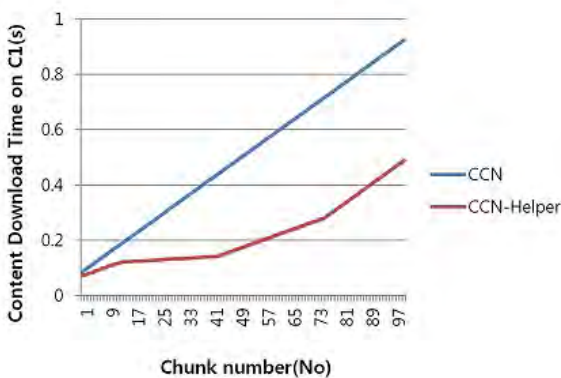
리량을 증가시킨다.

3. 성능 평가

본 논문에서 제안하는 CCN-Helper 방법론을 증명하기 위해 ndnSIM 2.0[13] 도구를 사용하여 테스트하였다. 실험에 사용된 혼잡 제어 토폴로지는 그림. 3 과 같으며, Interest 패킷에 대한 Data 패킷의 수신시간을 비교하기 위해 C1 요청자에서 Interest 패킷을 전송한다. 전송되는 패킷 크기는 100 개의 청크로 설정한다. 요청자와 송신자에서 전송되는 페이로드 크기는 1040 바이트로 설정하였다. 전송이 빠른 링크는 속도가 빠른 대신에 지연 시간이 길어서 오히려 성능 저하를 불러올 수 있도록 설정하였고 전송이 조금 느린 하지만 지연 시간이 짧은 링크의 경우를 설정하여 데이터 전송을 도와주는 링크로 설정하였다. 속도가 빠른 링크는 지연시간을 30ms 로 설정하였고 느린 링크에 대한 전송 Delay 는 2ms 로 설정한다. Bandwidth 는 전송이 빠른 링크의 경우에는 100Mbps 로 설정하고 느린 링크에는 1Mbps 로 설정한다. 캐쉬의 크기는 도착시간에 영향을 덜 주기 위해서 1000 으로 설정하고, 패킷 손실을 방지하기 위해 링크의 큐 크기는 500 으로 입력하여 테스트하였다.



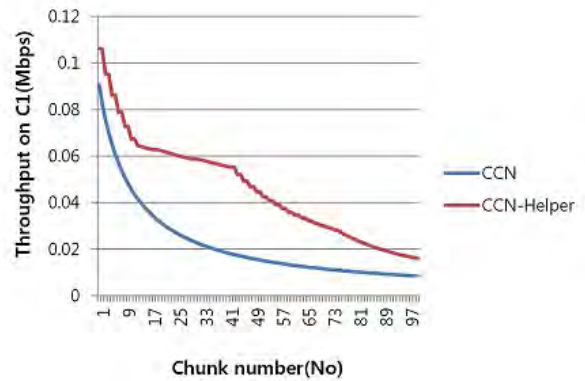
(그림. 3) 시뮬레이션 토폴로지



(그림. 4) C1 노드에서 다운로드 시간 비교

그림. 4 는 C1 노드에서 Interest 패킷에 대한 Data 패킷의 다운로드 시간을 그래프로 비교한 것이다. CCN의 도착 시간은 비례적으로 증가하는 반면, CCN-Helper 는 주변 노드의 도움을 받아 청크 데이터를 받기 때문에 도착시간이 거의 절반 정도 줄어든 것을 확인할 수 있다. N4 와 N5 의 중간 라우터에서 Data

패킷을 나누어서 전송함으로써 다운로드 시간을 줄일 수 있었다.



(그림. 5) C1 노드에서 처리량 비교

그림. 5 는 C1 노드에서 처리량을 비교한 것이다. 링크의 지연시간이 길기 때문에 처리량은 기본적으로 감소하는데 CCN 방법의 처리량은 지속적으로 감소하는 반면, CCN-Helper 는 처리량이 CCN 보다 조금씩 감소하는 것을 확인할 수 있다. 주변 노드의 도움을 받아서 패킷을 받게 되면 처리량을 증가시킬 수 있다는 사실을 확인하였다.

4. 결론

본 논문에서는 미래 인터넷 중의 하나로 이슈되고 있는 CCN 이론을 배경으로 CCN-Helper 프로토콜을 사용한 소비자의 다운로드 시간 감소 방법을 제안하여 생산자에서 소비자로의 패킷 전송 처리량을 향상시키고자 하였다. 네트워크의 링크 속도는 빠르지만 네트워크 환경이 좋지 않을 경우에 또는 속도가 느려서 다운로드 속도가 제대로 나오지 못하는 경우에 주변 CCN 노드에서 도움을 주어서 처리량을 향상시킬 수 있다. 이와 같은 CCN-Helper 프로토콜 방법을 통해서 수요자에게 전달되는 청크 데이터의 다운로드 속도를 증가시켜 처리량을 늘릴 수 있다. CCN 노드의 CS 용량에 따른 CCN-Helper 의 동작 방법은 다음 연구로 수행될 것이다.

ACKNOWLEDGMENT

본 연구는 KISTI 연구과제(사용자 기반 연구망 플랫폼 서비스 개발 및 적용: K-14-L01-C03-S03) 지원으로 수행되었습니다.

참고문헌

- [1] Van Jacobson, Diana K. Smetters, and James D. Thornton, "Networking Named Content," In proc. CoNEXT'09, pp. 1-12, New York, Dec. 2009.
- [2] Zhen Chen, Junwei Cao, and Zhenhua Guo, "A tentative comparison on CDN and NDN," Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference, pp. 2893-2898, San Diego, CA, Oct. 2014.
- [3] Lixia Zhang, Alexander Afanasyev, and Jeffrey Burke,

- “Named Data Networking,” in proc ACM SIGCOMM Computer Communication Review, Vol. 44, Issue 3, pp. 66-73, New York, Jul. 2014.
- [4] Stefano Salsano, Andrea Detti, and Matteo Cancellieri, “Transport-Layer Issues in Information Centric Networks,” in proc. ACM, pp. 19-24, NY, USA Aug. 2012.
- [5] Sara Oueslati, James Roberts, and Nada Sbihi, “Flow-aware traffic control for a content-centric network,” in proc. INFOCOM, pp. 2417-2425, Orlando, FL, Mar. 2012.
- [6] Ooka, A. Atat, S., and Inoue, K., Murata, M., “Design of a high-speed content-centric-networking router using content addressable memory,” Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on, pp.458-463, Apr. 2014.
- [7] Giovanna Carofiglio, Massimo Gallo, and Luca Muscariello, “ICP: Design and Evaluation of an Interest Control Protocol for Content-Centric Networking,” in proc. IEEE INFOCOM NOMEN, Orlando, USA, Mar. 2012.
- [8] Dojun Byun, Byoung-Joon Lee, and Myeong-Wuk Jang, “Adaptive flow control via Interest aggregation in CCN,” in proc. IEEE ICC, pp. 3738-3742, Budapest, Jun. 2013.
- [9] Sung-Hwa Lim, Young-Bae Ko, and Gue-Hwan Jung, Jaehoon Kim, Myeong-Wuk Jang, “Inter-Chunk Popularity-Based Edge-First Caching in Content-Centric Networking,” Communications Letters, IEEE , vol.18, no.8, pp.1331-1334, Aug. 2014.
- [10] Oueslati, S.; Roberts, J.; Sbihi, N., “Flow-aware traffic control for a content-centric network,” INFOCOM, 2012 Proceedings IEEE, vol., no., pp.2417-2425, 25-30 Mar. 2012.
- [11] Kazi, A.W., Badr, H., “PIT and Cache dynamics in CCN,” Global Communications Conference (GLOBECOM), 2013 IEEE, vol., no., pp.2120-2125, 9-13 Dec. 2013,
- [12] Dojun Byun, Byoung-Joon Lee, and Yongseok Park, “Adaptive interest adjustment in CCN flow control,” Global Communications Conference (GLOBECOM), 2014 IEEE, pp.1873-1877, 8-12 Dec. 2014.
- [13] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, “ndnSIM 2.0: A new version of the NDN simulator for NS-3,” NDN, Technical Report NDN-0028, 2015.