NDN에서 안정성 있는 데이터 배포를 위한 Lifetime 기반의 캐시 정책

최수호*, 조인휘*
*한양대학교 컴퓨터소프트웨어과
e-mail:aiek2612@hanyang.ac.kr
iwjoe@hanyang.ac.kr

A Lifetime-based Cache Policy for Stable Data Distribution in NDN

SU-HO CHOI*, IN-WHEE JOE.*
*Dept of Computer Software, Han-Yang University

요 약

CDN은 여러 서버에 데이터를 분산하여 캐시 한 뒤 빠르게 데이터를 배포하는 네트워크이다. 하지만 CDN은 서버 성능에 영향을 받고 데이터의 Lifetime이 얼마 없는 경우 중앙 서버 혹은 데이터 제작자와의 통신을 통해 데이터를 새로 받아야 한다. 반면에, NDN은 데이터를 라우터별로 캐시하여 트래픽을 분산시키고 딜레이를 줄였다. 하지만, 캐시 된 데이터의 Lifetime에 따라 데이터를 받지 못하는 문제가 있다. 본 논문에선 데이터 요청자가 Lifetime 내에 데이터를 못 받으면 요청자의 네트워크 상황과 Interest 패킷의 주기를 고려하여 캐시 된 데이터의 Lifetime을 증가시켜 안정성 있는 데이터 배포를 목표로 하였다. 캐시 정책이 적용된 시뮬레이션을 진행했으며, 얼마 남지 않은 Lifetime을 증가시켜 데이터 요청자가 데이터를 온전히 받을 수 있음을 확인하였다.

1. 서론

P2P(Peer-to-Peer) 네트워크는 여러 노드가 서로 클라이언트와 서버 역할을 동시에 행하며 데이터를 주고받는다. P2P의 통신 환경은 주로 비디오나 오디오와 같은 미디어 데이터나, 각종 디지털 형식의 파일 공유에 사용된다. 그 예로 FTP(File Transfer Protocol), Torrent 등이 있다.

그리고 P2P 네트워크는 피어 간의 안전한 데이터 전송을 위하여 TCP(Transmission Control Protocol) 전송계층 프로토콜을 사용한다. 그리고 IP를 통해서 대상을 특정 짓은 뒤, 연결을 시도하여 데이터를 주고받는다. 만약, 피어에 대한 정보에 IP가 없거나, IP가 변경되었지만 피어 정보에 업데이트가 안 된 경우에는 통신할 수 없다.

더욱이 피어 간의 링크가 단절된 경우, 해당 피어를 통해서는 데이터를 받을 수 없다. 이러면 링크가 복구되기를 기다리거나, 또는 해당 데이터를 가진 다른 피어를 계속해서 검색해야 하는 문제점이 있다.

CDN(Content Delivery Network)은 여러 서버에 데이터를 분산하여 캐시 한 뒤 사용하는 네트워크이다. 데이터를 분산하여 저장하기 때문에 딜레이가 감소하여 빠르게 데이터를 받을 수 있다. 하지만 서버의 성능이 낮은 경우 병목현상으로 딜레이가 높아져 데이터를 받는 속도가 감소한다. 또한, 캐시 된 데이터의 Lifetime이 얼마 없는 경우 중앙 서버 혹은 데이터 제작자와의 통신을 통해 데이터를 새로 받아야 하므로 트래픽이 유발된다. 더욱이, 중앙 서버나 데이터 제작자와의 링크가 단절된 경우 캐시된 데이터는 소멸할 수밖에 없다. 그리고 서버 단위 공격에 취약하다.

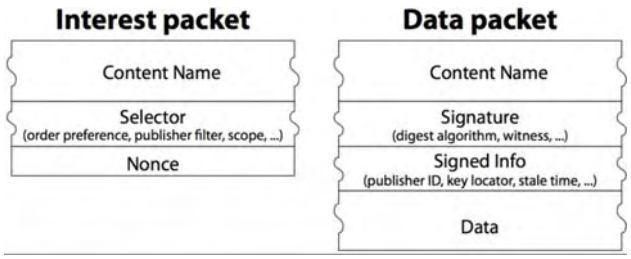
이러한 문제점들을 해결하기 위해 정보 중심 네트워크인 ICN(Information-Centric Network)[1]에 대한 연구로 NDN(Named-Data Network)[2][3][4]이 활발히 진행되고 있다. ICN에서는 IP 주소 체계가 아닌 데이터의 이름을 사용하여 통신한다. 그리고 데이터들을 캐시 하여 노드별로 존재하는 CS(Content Store)에 캐싱할 수 있다. 이에 따라 트래픽을 더욱더 분산시킬 수 있으며, 집중된 공격에 대비할 수 있다. 그리고 위에서 설명한 데이터 제작자와의 링크 단절이나 혼잡 등의 문제를 해결할 수 있다. 더욱이, 데이터 제작자의 위치가 노출되지 않고 바로 데이터를 전달하므로 보안성이 뛰어나다.

특정 데이터를 찾는 피어들은 Interest 패킷에 데이터 이름을 담아 라우팅을 한다. 해당 데이터를 가진 피어나 라우터에 존재하는 CS를 통해서 데이터를 반환받는 Hop-by-Hop 구조로 되어 있다. 요청한 피어에 데이터를 전달하는 동안, 지나가는 피어나 라우터에 데이터를 캐싱한다. 이후에 똑같은 이름을 가진 데이터를 찾을 때 데이터가 캐시 된 노드들을 통해서 데이터를 반환받을 수 있다.

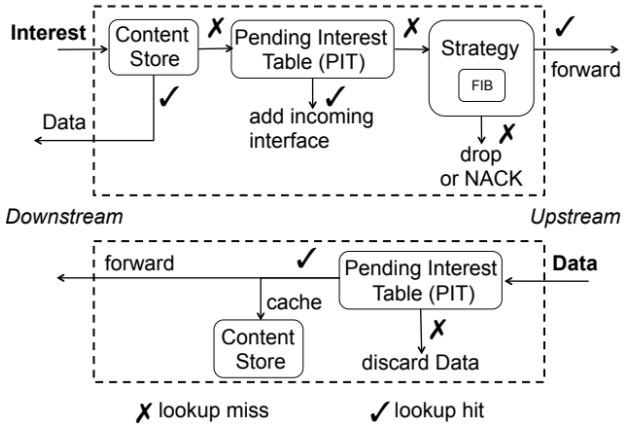
여러 피어가 같은 데이터를 찾는 경우, 인근의 라우터에 존재하는 PIT(Pending Interest Table)에 등록되고 하나의 Interest 패킷을 전달해 네트워크 오버헤드를 줄일

수 있다. (그림 1)과 (그림 2)는 각각 NDN의 패킷과 아키텍처 구조를 나타낸다.

이 논문의 구성은 다음과 같다. 2장은 네트워크 구성 및 제안하는 캐시정책을 소개한다. 3장에서 성능을 도출하고 평가한 뒤, 4장에서 결론을 맺는다.



(그림 1) NDN 패킷 구조[2]



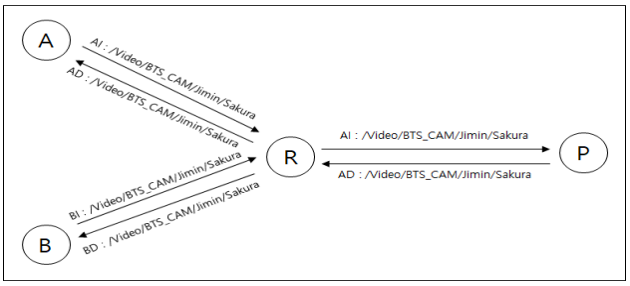
(그림 2) Forwarding Process at an NDN Node[4]

2. 제안하는 방법

본 장에서는 NDN 기반의 P2P 네트워크와 함께 제안하는 알고리즘을 소개한다.

각 데이터의 이름은 계층적으로 나누어져 있다. 본 논문에서는 *Sakura*라고 닉네임을 가진 데이터 제작자가 촬영한 *BTS CAM Video* 중 *Jimin* 이라는 이름을 가진 멤버의 영상을 찾도록 하였다. 데이터를 찾는 피어는 먼저 Interest 패킷 내에 *'Video/BTS_CAM/Jimin/Sakura'* 라는 이름을 삽입한 뒤, 라우터에 전달한다. 라우터들은 해당 Interest 패킷을 확인한 뒤, 자신의 CS에 해당 데이터가 존재하면 데이터를 반환하고, 그렇지 않다면 다음 노드를 향해 Interest 패킷을 전달한다. 이러한 행동을 반복하면 데이터를 가진 피어에게 Interest 패킷이 전달된다. 이후에, 해당 데이터를 가진 피어가 Interest 패킷당 Data를 Hop의 역순으로 전달하게 된다.

데이터를 요청한 피어에게 해당 데이터를 전달하면서 만나는 라우터의 CS에 데이터를 캐싱하게 된다. 그리고 해당 라우터와 연결된 피어가 똑같은 데이터를 요청할 경우, 데이터 제작자에게 패킷이 전달되지 않고 라우터에서 전달하므로 빠르게 데이터를 보급할 수 있다. 이러한 네트워크 구조는 (그림 3)과 같다.

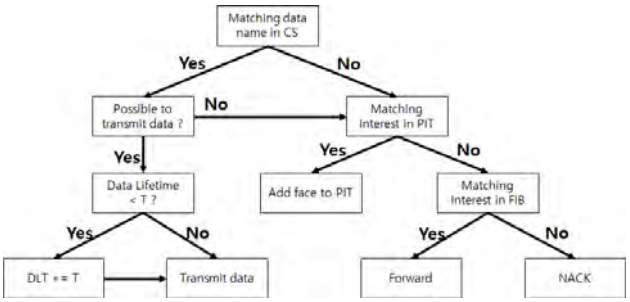


(그림 3) NDN 네트워크와 데이터 요청 및 반환

반면에, 데이터를 무작정 캐시할 경우, 각 노드의 저장소 문제가 유발된다. 이러한 문제를 해결하기 위해 데이터 Lifetime을 제작자가 설정할 수 있다. 하지만 데이터 Lifetime이 짧은 경우에는 데이터 배포에 문제가 생길 수 있다. 따라서 본 논문에서는 짧은 Lifetime을 가진 데이터를 찾는 피어들을 위해 캐시 된 데이터의 Lifetime을 추가로 늘리는 작업을 캐시 정책에 추가하였다. 수식을 통해 나온 시간보다 데이터의 Lifetime이 작으면 추가하는 정책이다. 수식은 다음과 같다.

$$L = Bandwidth \times Datalinklatency \quad (1)$$
$$P = Frequency \times Packetlossprobability \quad (2)$$
$$R = otherstime \quad (3)$$
$$T = Datasize \div L \div P + R \quad (4)$$

수식 (1)은 라우터와 연결된 데이터 링크의 처리량이다. 수식 (2)는 라우터가 받는 패킷의 수이다. 1초에 보낼 수 있는 Interest 패킷의 개수인 Frequency에 패킷 손실 확률을 곱하였다. 수식 (3)은 라우터 내에서 데이터를 찾는 시간이나 디스크 문제, 버퍼 사용 등으로 인한 추가적인 시간이다. 마지막으로 수식 (4)는 최종 시간이다. Data Size를 L로 나눈 뒤, P로 다시 나누었다. 그리고 R을 더하였다.



(그림 4) 캐시 정책

(그림 4)는 데이터 Lifetime을 추가시킨 캐시 정책이다. 최초, Interest 패킷을 받은 라우터는 자신의 CS에 찾고자 하는 데이터의 이름이 있는지 찾아본다. 데이터가 존재한

다면 라우터는 해당 데이터의 Lifetime을 확인하여 보낼 수 있는 상태인지 그렇지 않은지를 판단한다.

보낼 수 없는 상태라면 PIT에 이미 등록되어 있는지, 그렇지 않은지를 파악한다. PIT에 등록되어 있다면, 해당 face에 추가한다. PIT에 등록되어 있지 않다면, FIB에 Interest가 찾는 데이터를 가진 노드가 있는지 확인한다. FIB에 존재한다면, 해당 노드까지 적절한 Forwarding Strategy를 사용하여 Interest를 전달한다. FIB에 존재하지 않다면 NACK를 보내고 종료한다.

반면에, 데이터를 보낼 수 있는 상태라면 데이터의 Lifetime과 T를 비교한다. 데이터의 Lifetime이 T보다 작다면, T만큼 더해준 뒤 데이터를 전달한다. 데이터 Lifetime이 T만큼 늘어나서 데이터를 요청한 피어가 데이터를 온전히 받을 수 있다. 데이터의 Lifetime이 T보다 큰 경우에는 피어가 데이터를 받는 데 문제가 없으므로 바로 데이터를 전달해준다.

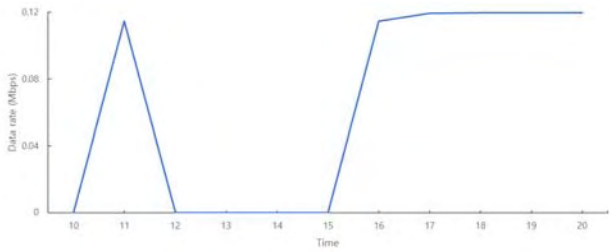
3. 성능 평가

본 장에서는 제안한 네트워크와 알고리즘을 토대로 성능 평가를 진행하였다. 네트워크 시뮬레이션은 NS-3[5]와 ndnSIM[6][7]을 사용하였다. 네트워크는 (그림 3)과 같이 구성하였고 거리는 400m 단위로 설정하였다. 시뮬레이션은 총 20초 동안 진행되었으며, A 노드는 라우터에 데이터 캐싱을 위해 10초 동안 실행한 뒤 종료하였다. B 노드는 10초부터 시뮬레이션 종료까지 실행되었다. 링크 단절은 12초부터 15초 사이로 지정했다. 사용된 데이터와 값은 (표 1)에 정리되어 있다.

(표 1) Simulation Parameters

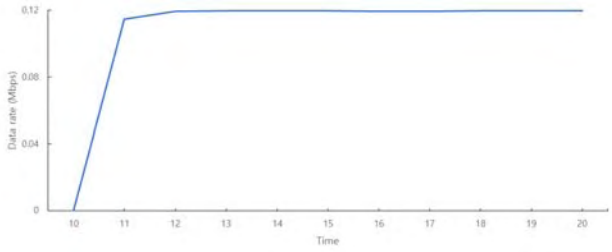
Parameter	Value
Payload size	1500 bytes
Data size	10 MB
Data link rate	100 Mbps
Data link delay	10 ms
Others time	5 ms
Interest frequency	10
Default data lifetime	11 s

(그림 5)는 제안된 알고리즘을 적용하기 전 B 노드의 처리량이다. 라우터 내 캐싱 된 데이터의 설정된 데이터 Lifetime이 11초이기 때문에 10초부터 실행된 B 노드는 라우터로부터 1초 동안 데이터를 받을 수 있게 된다. 15초 이후에는 데이터 제작자와의 링크가 복구되어 제작자로부터 데이터를 받기 시작했다.



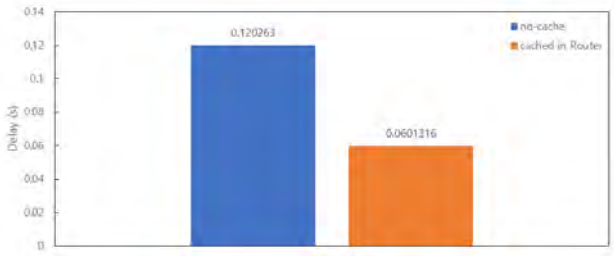
(그림 5) 캐시 정책 적용 전 B 노드의 처리량

반면에, (그림 6)은 제안된 알고리즘을 적용하였다. 수식(3)에 따라 도출된 값 T는 약 10초이다. B 노드의 첫 Interest를 받은 시점에서 남은 데이터 Lifetime은 1초이다. 따라서, 남은 데이터 Lifetime은 증가하게 되었고 시뮬레이션이 종료될 때까지 B 노드는 캐싱 된 데이터를 온전히 받을 수 있다.



(그림 6) 캐시 정책 적용 후 B 노드의 처리량

(그림 7)은 각 상황에 따른 평균 딜레이다. 데이터가 캐시되어 있지 않거나, Lifetime이 끝나서 데이터 제작자에게 데이터를 요청하는 경우 딜레이는 약 0.12초가 발생했다. 반면에, 가까운 라우터에 캐시 된 경우 딜레이는 약 0.06초가 발생했다. 따라서, 캐시 된 라우터를 통해서 데이터를 전송하는 것이 더 좋다.



(그림 7) 평균 딜레이

4. 결론

본 논문에서는 NDN 기반 P2P 네트워크에서 라우터에 캐시 된 데이터의 Lifetime 증가시킴으로써 링크 단절에도 안정성 있는 데이터 배포 방법을 알아보았다. 제안한 캐시 정책을 적용하여 시간이 얼마 남지 않은 Lifetime을 증가시켰고 그에 따라 라우터에 캐시 된 데이터를 요청자가 온전히 받을 수 있었으며, 낮은 딜레이를 유지하였다.

참고문헌

[1] Dannewitz C, “an information-centric design for the future internet. In: Proceedings of the 3rd GI/ITG KuVS workshop on the future internet”

[2] Named Data Networking project. https://named-data.net/

[3] Zhang, Lixia, et al. "Named data networking." ACM SIGCOMM Computer Communication Review 44.3 (2014).

[4] A.Afanasyev, J.Burke, T.Refaei, “A Brief Introduction to Named Data Networking”, MILCOM 2018

[5] NS-3: Network Simulator 3 https://www.nsnam.org/

[6] Afanasyev, Alexander, I. Moiseenko, and L. Zhang. "ndnSIM: NDN simulator for NS-3." University of California, Los Angeles, Tech. Rep 4 (2012).

[7] ndnSIM: NDN simulator for NS-3 http://ndnsim.net/current/