콘텐츠 중심 네트워킹을 위한 효율적인 전송 기법

권봉용○,홍충선,이성원
경희대학교
{by1130,cshong,drsungwon}@khu.ac.kr

A Efficient Transmission Support Scheme for Content Centric Networking

BongYong Kwon○,Choong Seon Hong,Sungwon Lee
Kyung Hee University

요 약

최근 인터넷 트래픽이 폭증하면서 네트워크를 효율적으로 활용하기 위한 방법들이 연구되고 있다. 컨텐츠 중심 네트워크(CCN)는 컨텐츠 이름을 기반으로 주소 체계를 만들며, 각 CCN 노드들이 전송된 데이터를 저장하여 네트워크상에서 발생하는 트래픽 양을 줄이고, 빠른 응답시간 또한 보장할 수 있는 미래인터넷 아키텍처이다. 본 논문에서는 CCN에서 하나의 Interest 패킷을 이용하여 n개의 data 패킷을 받는 기법에 대해 제안하고자 한다.

1. 서 론

현재의 인터넷은 1970년대에 만들어진 구조와 방식으로 동작하고 있으며[1], 함부로 구조를 변경하기가 어렵기 때문에 인터넷에서 문제가 발생하는 경우 부분적으로 해결책을 적용하는 방식으로 인터넷이 진화하고 있다. 또한 최근 인터넷 트래픽이 폭증하면서, 네트워크를 효율적으로 활용하기 위한 방법들이 부각되고 있으며, 인터넷에서 발생하는 문제들을 해결하기 위해 다양한 미래 인터넷 연구들이 진행되고 있다[2][3].

컨텐츠 중심 네트워크(CCN: Content Centric Networking)[4]는 컨텐츠 이름을 기반으로 주소 체계를 만들며, 각 CCN 노드들이 전송된 데이터를 저장하여 네트워크상에서 발생하는 트래픽 양을 줄이고, 빠른 응답시간 또한 보장할 수 있는 미래인터넷 아키텍처이다.

본 논문에서는 CCN에서 1개의 Interest packet으로 n개의 Data Packet을 받아 보다 효율적으로 데이터를 받는방법에 대해 설명한다.

2. 관련연구

2.1 Basic Content Centric Networking

CCN에서는 원하는 데이터를 식별하는 이름을 가진 Interest 패킷을 유저가 보냄으로써 시작된다. 라우터가 Interest 패킷을 받으면, Content Store(CS)의 리스트에서 Data를 확인한다. 만약 CS에서 Data를 찾으면 바로 클라이언트로 전송을 하지만, Data를 찾지 못하면 Pending Interest Table(PIT)에서 Data를 찾아 클라이언트로 전송을 한다. 마찬가지로 Data를 PIT에서 찾지 못하면, Forwarding Information Base(FIB)에서 그들의 이름을 찾음으로써 Interest 패킷을 포워딩하는 인터페이스를 기억하고, Interest 패킷이 요청된 데이터를 가진 노드에 도착할 때, Data 패킷은 Interest 패킷에 의해 만들어진 경로를 추적해서 그 역 경로로 보내진다.

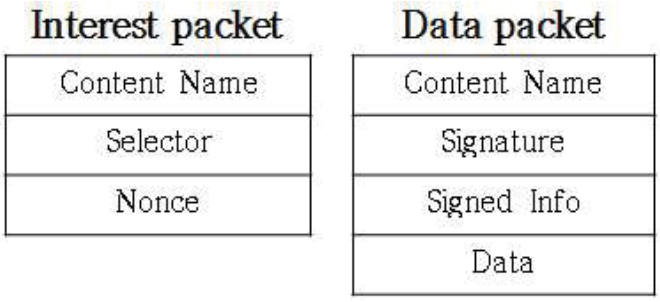

그림 1: CCN 아키텍처에서의 패킷

그림 1에서 볼 수 있듯이 CCN 패킷은 Interest 패킷과 Data 패킷으로 나뉜다. Interest 패킷은 Content Name, Selector, Nonce 이렇게 3개의 필드를 가지고 있으며[5], Data Packet은 Content Name, Signature, Signed Info, Data 필드를 가지고 있다[6].

3. 제안사항

3.1 Modified Content Centric Networking

Basic CCN에서는 서버에서 보내는 Data 패킷의 비율을 조절하거나 원하지 않는 데이터를 전송해서 대역폭을 소비하는 것을 방지하는 등의 flow balance를 유지하기 위해 한 개의 Interest 패킷으로 한 개의 Data 패킷을 받지만[7], 본 논문에서는 Data 패킷을 받기 위해 각각의 Interest 패킷을 보내는 것 보다 효율적으로 하나의

본 연구는 미래부가 지원한 2013년 정보통신•방송(ICT) 연구개발사업의 연구결과로 수행되었음. *Dr. CS Hong is the corresponding author.

Interest 패킷으로 n개의 data 패킷을 받을 수 있는 기법을 제안한다.

알고리즘1. Modified CCN Function

```
#Initial Value
WinSize=10
Win[WinSize]
k=0
WP=0
MaxWP=0
LCID=-1
#Send Interest packet
n=2k
if n>WinSize then
    n=WinSize
end if
send(SInt(n))
LastT=Now
k=k+1

#Receive Data packets
RCID=recvData()
if RCID == LCID+1 then
    LCID=RCID
else
    WP=RCID-LCID-1
    Win[WP]=RCID
    If WP > MaxWP then
        MaxWP=WP
    end if
    if RCID == LCID+1 and Win[1]-1 == RCID
then
        Win[0]=RCID
    end if
    for i=0, MaxWP do
        if Win[i] == LCID +1 then
            LCID= Win[i]
        end if
    end for
    if Win[MaxWP] == LCID or WP == WinSize
then
        WP=0
        MaxWP=0
    end if
end if


#Resend Interest Packet
if RCID != LCID+1 or LastT<Now-InterestLifetime
then
    k=0
    Resend(Int(LCID+1))
    LastT=Now
end if
```

본 논문에서 제안하는 기법에서는 그림 1의 Interest 패킷에 몇 개의 Interest 패킷을 받을 지에 대한 필드를 추가하고, 클라이언트는 n개의 패킷을 요청한 후에 받는 Data 패킷 중 패킷 손실이 있을 경우, 이미 받은 Data 패킷에 대한 정보를 가지고 있는 윈도우를 이용하여 Data 패킷을 받으면서 패킷손실이나 타임아웃이 발생하면 재전송을 요청한다.

알고리즘 1에서 WinSize는 클라이언트가 가지고 있는 윈도우 사이즈, Win은 클라이언트가 가지고 있는 윈도우, k는 2의 지수, WP는 윈도우의 위치를 가리키는 포인터, MaxWP는 Win에서 RCID를 가지고 있는 가장 마지막 포인터, LCID는 가장 최근에 받은 ChunkID, Int는 새로운 Interest 패킷 생성, CID는 ChunkID, RCID는 받은 ChunkID, n은 요청하고자 하는 data 패킷의 수, SInt는 n개의 data 패킷을 받기 위한 새로운 Interest 패킷 생성, LastT은 가장 최근 Interest 패킷을 전송한 시간, InterestLifetime은 Interest 패킷의 라이프타임이다.

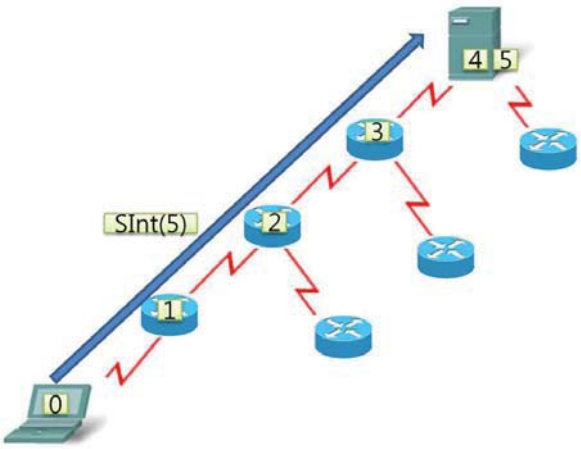

그림 2. SInt 패킷 전송

그림 2는 클라이언트가 SInt 패킷을 전송하는 상황을 보여준다. SInt 패킷을 전송하는 경우 하나의 Interest 패킷으로 WinSize보다 작거나 같은 n개의 data 패킷을 순차적으로 받을 수 있으며, 패킷손실이나 타임아웃이 발생하면 그 패킷에 대한 Data 패킷만 재전송을 요청하고 요구하는 Data 패킷의 수는 1로 변경된다. 또한, 아무 문제없이 Data 패킷을 받은 경우 요청하는 Data 패킷의 수는 증가한다. Data 패킷을 받는 경우 클라이언트는 현재 받은 Data 패킷의 시퀀스가 순서에 맞게 들어왔는지 확인을 하고 중간의 패킷이 손실되는 경우 윈도우에 저장을 해서 필요 없는 재전송이 일어나지 않게 한다.

그림 3은 클라이언트가 Data 패킷을 받는 상황을 보여준다. CID가 0~5인 Data 패킷을 요청한 후 클라이언트는 CID가 1,2,4,5인 Data 패킷을 받았지만, 0,3은 패킷 손실

이 발생하였다. 클라이언트는 패킷 0,3을 재요청해서 패킷을 받은 후 지금 까지 받은 윈도우 0~5를 초기화한다. 패킷 손실이나 타임아웃이 발생하여 중간의 Data 패킷을 못 받았더라도 윈도우를 사용하여 이미 받은 패킷에 대해 재전송을 요청하지 않으며, 못 받은 Data 패킷에 대해서만 재요청 한다.

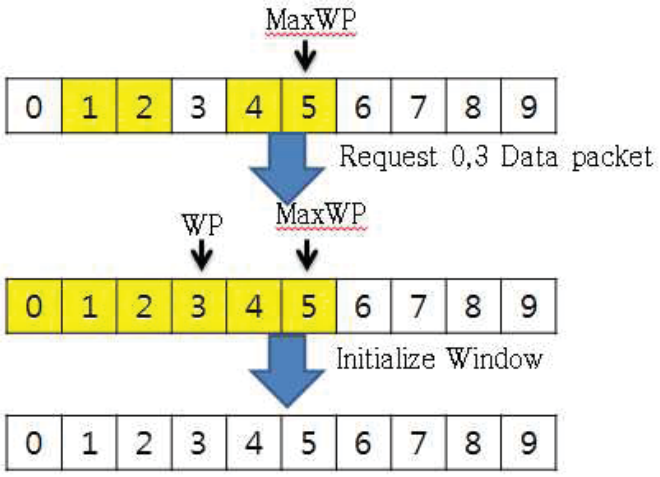

그림 3. Window Function

3. 성능평가

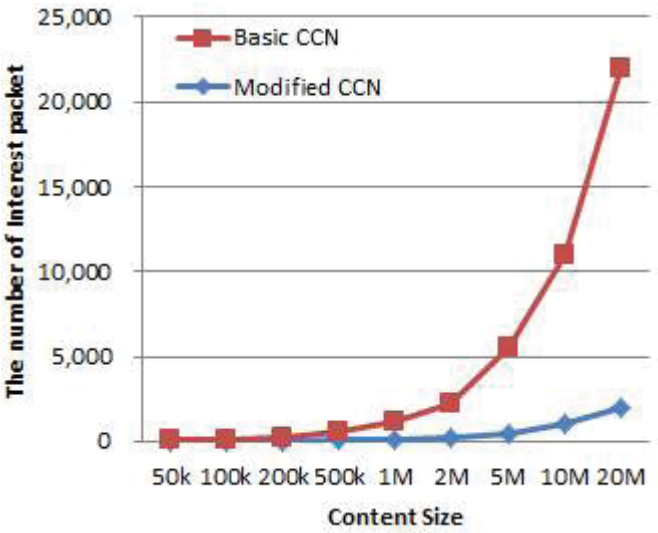

그림 4. The number of transmitted Interest packet

본 논문에서 제안하는 기법의 성능평가를 위해 기존 CCN에서의 Interest 패킷 수와 비교하였다. 그림 4는 각 컨텐츠 크기에 대하여 전송된 Interest 패킷의 수를 비교한 것이다. 청크의 크기는 1KB로 지정하였으며, 그림 4에서 보이는 것처럼, Interest 패킷의 수가 기존의 방식보다 감소하는 것을 볼 수 있다.

4. 결론 및 향후 연구 방향

본 논문에서는 CCN에서 하나의 Interest 패킷을 이용하여 n개의 Data 패킷을 받는 기법을 제안하였다. 향후 최적화된 윈도우 사이즈를 찾아내고 이를 기반으로 CCN Streaming에 적용을 해서 본 논문의 알고리즘이 스트리밍 서비스의 효율성에 어느 정도 영향을 미치는지에 대한 연구를 고려하고 있다.

5. 참고문헌

[1] 임규보,이정환,유혁 CCN 환경에서의 SVC 비디오 스트리밍을 위한 캐시 교체 정책에 관한 연구" 2012 한국컴퓨터종합학술대회 논문집 Vol39, No.1(D), 235p, 2012년

[2] 김정임,정희영,박우구 "콘텐츠 중심의 네트워크 기술" 전자통신동향분석 제25권, 제6호, 136p, 2010년 12월

[3] 최낙중,황재현 "콘텐츠 중심 네트워크에서의 적응적 비디오 스트리밍" 한국통신학회지, 30권, 30호, 73p, 2013년 3월

[4] CCNx Home Page. "http://www.ccnx.org/"

[5] CCNx Interest Message. "http://www.ccnx.org/releases/latest/doc/technical/InterestMessage.html"

[6] CCNx Content Object. "http://www.ccnx.org/releases/latest/doc/technical/ContentObject.html"

[7] CCNx Protocol. "http://www.ccnx.org/releases/latest/doc/technical/CCNxProtocol.html"