

콘텐츠 중심 네트워킹을 위한 효율적인 미디어 스트림 전송 기법

(An Efficient Media Stream Transmission Scheme for Content Centric Networking)

권 봉 용[†] 홍 충 선^{**}
(BongYong Kwon) (ChoongSeon Hong)

이 성 원^{**} 최 종 원^{***}
(Sungwon Lee) (Jongwon Choe)

요 약 최근 다양한 스마트 기기들이 개발되고 수많은 사용자가 스마트 기기로서 인터넷, 미디어, 게임 등을 사용함으로써 매년 인터넷 트래픽이 폭증하게 되었고, 그로인해 네트워크를 효율적으로 활용하기 위한 방법들이 연구되고 있다. 콘텐츠 중심 네트워킹(CCN)은 콘텐츠 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 기법을 미디어 스트리밍에 적용하는 방식과 패킷 손실에 대응하는 방안을 제안한다. 또한 제안한 방식이 기존 CCN 방식보다 Interest 패킷의 발생량과 패킷손실율의 측면에서 우수함을 보인다.

키워드: 콘텐츠 중심 네트워킹, CCN, 스트리밍, 미래인터넷

· 본 연구는 미래장조과학부 및 정보통신산업진흥원의 대학 IT연구센터 육성지원사업의 연구결과로 수행되었음(NIPA-2013-(H0301-13-4006))

· 이 논문은 2013 한국컴퓨터종합학술대회에서 '콘텐츠 중심 네트워킹을 위한 효율적인 전송 기법'의 제목으로 발표된 논문을 확장한 것임

[†] 학생회원 : 경희대학교 컴퓨터공학과
by1130@khu.ac.kr

^{**} 종신회원 : 경희대학교 컴퓨터공학과 교수
cshong@khu.ac.kr
(Corresponding author)
drsungwon@khu.ac.kr

^{***} 종신회원 : 숙명여자대학교 컴퓨터공학과 교수
choejn@sookmyung.ac.kr

논문접수 : 2013년 9월 13일

심사완료 : 2014년 1월 10일

Copyright©2014 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제20권 제3호(2014.3)

Abstract Recently, a variety of smart devices have been developed and a number of users uses smart devices for accessing Internet, multimedia contents and game contents. For this reason, the traffic on internet has been burst tremendously, thereby the methods for an efficient utilization of network are being studied in these days. Content-centric networking is a future internet architecture, with an innovative addressing system based on the name of the content and with each of content nodes caches all the transferred data to reduce the amount of overall traffic that will flow over the network and to ensure faster response time. In this paper, we propose techniques of media streaming where a client receives n data packets using single Interest packet. This proposal also deals with possible packet losses. The simulation result shows the superiority of the proposed method to the conventional CCN in terms of amount of generated Interest packet and packet loss rate.

Keywords: content centric networking, CCN, streaming, future internet

1. 서 론

현재의 인터넷은 1970년대에 만들어진 구조와 방식으로 동작하고 있다[1]. 그러나 인터넷은 시대적 요구를 수용하기에는 그 구조에 문제가 제기되고 있으며, 합부로 구조를 변경하기 어렵기 때문에 인터넷에서 문제가 발생하는 경우 부분적으로 해결책을 적용하는 방식으로 인터넷이 진화하고 있다.

또한 최근 다양한 스마트 기기들이 개발되고 수많은 사용자가 스마트 기기로서 인터넷, 미디어, 게임 등을 함으로써 매년 인터넷 트래픽이 폭증하게 되었고[2], 그로인해 네트워크 트래픽을 효율적으로 활용하기 위한 다양한 방법들이 부각되고 있으며, 인터넷에서 발생하는 문제들을 해결하기 위해 다양한 미래인터넷 연구들이 진행되고 있다[3,4].

콘텐츠 중심 네트워킹(CCN: Content Centric Networking)[5]은 콘텐츠 이름을 기반으로 주소 체계를 만들어 각 CCN 노드들이 전송된 데이터를 저장하여 네트워크상에서 발생하는 트래픽 양을 줄이고, 빠른 응답시간 또한 보장할 수 있는 미래인터넷 아키텍처이다.

본 논문에서는 CCN에서 1개의 Interest 패킷으로 n 개의 Data 패킷을 받는 기법을 미디어 스트리밍에 적용하여 보다 효율적으로 데이터를 받는 방법과 패킷 손실이 일어날 경우 이에 대한 해결 방안을 제시한다.

2. 관련 연구

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. 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 1에서 볼 수 있듯이 CCN 패킷은 Interest 패킷과 Data 패킷으로 나뉜다. Interest 패킷은 Content Name, Selector, Nonce 이렇게 3개의 필드를 가지고 있으며[6], 클라이언트가 원하는 콘텐츠를 요청할 때 사용된다. Data 패킷은 Content Name, Signature, Signed Info, Data 필드를 가지고 있고[7], 클라이언트의 요청에 대한 응답으로 사용된다.

그림 2는 CCN에서의 라우팅을 보여준다. 클라이언트 1에서 CCN 서버로 Interest 패킷을 전송하여 원하는 콘텐츠를 요청하면(a), 서버는 Interest 패킷의 역방향으로 Data 패킷을 클라이언트로 전송하며, 그 경로에 있

는 라우터들은 Data 패킷을 CS에 저장한다(b). 그 후 클라이언트2가 클라이언트1과 같은 콘텐츠를 요청을 한다면(c), CCN 라우터의 CS에 Data 패킷이 저장되어 있기 때문에, 클라이언트2는 보다 빠르게 Data 패킷을 받을 수 있다(d). 이러한 CCN의 특성으로 인해 동일한 데이터를 수많은 사용자 수만큼 반복 전송되는 방식의 현재의 인터넷보다 높은 효율성을 가지게 된다. CCN 기술은 서버가 데이터를 전송한다는 것에 제한을 두지 않고, 데이터를 가지고 있는 라우터도 데이터를 전송할 수 있게 함으로써, 빠른 데이터 전송과 데이터가 네트워크에서 반복되어 전송되는 횟수를 줄이는 장점을 가진다.

2.2 Streaming

스트리밍(Streaming)이란 인터넷에서 미디어 데이터를 전송하는 기술이다. 스트리밍은 ‘흐름’이라는 의미로 미디어 파일을 하나의 형태가 아닌 여러 개의 파일로 나누어 물 흐르듯이 연이어 보낸다. 동영상 파일의 경우 용량이 크기 때문에 한꺼번에 파일 전체를 보내는 것은 힘들기 때문에 이를 해결하기 위해 파일을 작은 조각으로 나누어 조금씩 (영상이 실제로 플레이되는 분량) 실시간으로 전송해주는 것이다. 네트워크의 상태에 맞추어 비교적 큰 스트리밍 파일의 경우 아주 작은 크기의 조각들로 나누어지며, 이 각각의 조각들은 뒤의 조각들과 이어질 수 있는 헤더정보를 가진 상태로 전송되고, 클라이언트에서는 이 조각들을 받으면서 동시에 동영상 재생된다.

스트리밍에는 다음과 같이 다양한 서비스가 존재한다.

- VoD 스트리밍(VoD Streaming) : Video on Demand Streaming으로 맞춤형영상정보 서비스, 주문형 비디오 조회 시스템이라고도 한다. 서버에서 프로그램을 일방적으로 수신하는 것이 아니라 촬영 및 편집을 거쳐 동영상 파일을 만든 후, 클라이언트의 요구에 따라 원하는 시간에 원하는 내용의 동영상을 제공하는 쌍방향 서비스이다.
- HTTP 라이브 스트리밍(HTTP Live Streaming) : 라이브 스트리밍을 위한 전통적인 프로토콜인 RTP/RTSP는 방화벽 환경에서 서비스가 원활하게 돌아가지 않는 등의 단점이 있는데 이를 보완하기 위해 Apple이 HTTP를 이용해 스트리밍 서비스가 개발하였으며, MPEG-2 TS 코덱을 사용하고 Adaptive Bitrate Streaming을 지원한다[8].
- 적응적 비트율 스트리밍(Adaptive Bitrate Streaming) : 네트워크의 상태(전송 속도, 대역폭 등)에 따라 가능한 정도의 화질을 가지는 동영상을 전송하는 기법이다. 대역폭이 낮아지거나 네트워크 혼잡도가 높아져서 전송효율이 떨어지면 낮은 화질의 콘텐츠가 전송된다[9].

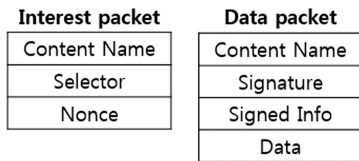


그림 1 CCN 아키텍처에서의 패킷
Fig. 1 CCN Architecture of the packet

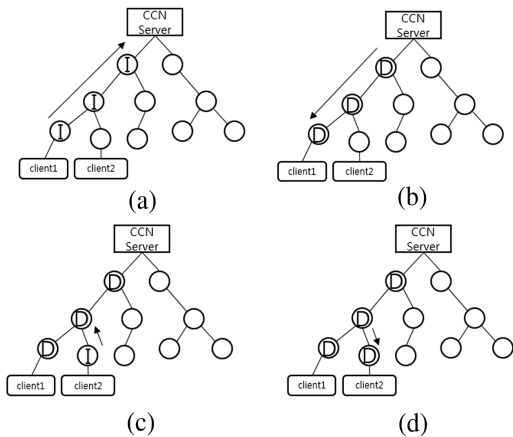


그림 2 CCN 라우팅
Fig. 2 CCN Routing

3. 제안사항

3.1 CCN with SInterest Packet

Basic CCN에서는 서버에서 보내는 Data 패킷의 비율을 조절하거나 원하지 않는 데이터를 전송해서 대역폭을 소비하는 것을 방지하는 등의 흐름 균형(flow balance)을 유지하기 위해 한 개의 Interest 패킷으로 한 개의 Data 패킷을 받지만[10], 본 논문에서는 Data 패킷을 받기 위해 각각의 Interest 패킷을 보내는 것 보다 효율적으로 하나의 Interest 패킷으로 n개의 Data 패킷을 받을 수 있는 SInterest 패킷과 패킷 손실이 발생할 경우 손실된 패킷만 재전송 요청하기 위해 전통적인 TCP에서 혼잡제어를 위해 사용하는 윈도우를 CCN에서 사용하고자 한다.

Alg1. 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
    
```

표 1 알고리즘 1 파라미터 설명
Table 1 Describes the parameters in Alg1

WinSize	Size of the window in which the client has
Win	the window in which the client has
k	Power of 2
WP	A pointer to the position of the window
MaxWP	Last pointer that have RCID in Win
LCID	Last received ChunkID
Int	Function that create new Interest packet
CID	ChunkID
RCID	Received ChunkID
n	Number of request data packet
SInt	Function that create new Interest packet for receive n data packet
LastT	Last time that sending Interest packet
InterestLifetime	Interest packet's life time

본 논문에서 제안하는 기법에서는 그림 1의 Interest 패킷에 몇 개의 Interest 패킷을 받을 지에 대한 필드를 추가하여, CCN 서버 혹은 라우터가 이 SInt(n)를 받을 경우 n개의 Interest 패킷을 연속적으로 보낼 수 있게 수정하였으며, 클라이언트가 n개의 패킷을 요청한 후에 받는 Data 패킷 중 패킷 손실이 있을 경우, 이미 받은 Data 패킷을 확인하고 손실된 패킷만을 재전송 요청하기 위한 윈도우를 추가하였다.

Alg1은 본 논문에서 제안하는 기법의 알고리즘이며, 표 1은 Alg1에서 사용되는 파라미터에 대한 설명이다. SInt 패킷을 전송하는 경우 하나의 Interest 패킷으로 WinSize 보다 작거나 같은 n개의 Data 패킷을 순차적으로 받을 수 있으며, 패킷 손실이 일어나지 않으면 Interest 패킷의 수는 표 2와 같이 1/n로 감소된다. 본 알고리즘에서는 WinSize를 10으로 고정하였으며, 한 번에 받을 수 있는 패킷의 최대치를 요청된 모든 패킷이 손실될 경우를 생각하여 WinSize와 같은 10으로 하였다. 만약 패킷손실이나 타임아웃이 발생하면 그 패킷에 대한 Data 패킷만 재전

표 2 Interest 패킷 수 비교
Table 2 Comparison of the number of Interest packet

Basic CCN	Modified CCN
$\frac{\text{Content size}}{\text{chunk size}}$	$\frac{\text{Content size}}{\text{chunk size} * n}$

송을 요청을 하게 되며 k를 0으로 만들어 요구하는 Data 패킷의 수를 1로 초기화한다. 또한, 아무 문제없이 Data 패킷을 받은 경우 요청하는 Data 패킷의 수는 2의 지수로 증가하게 되는데, 이 경우 요청하는 Data 패킷의 수는 클라이언트가 가지고 있는 윈도우의 최대크기로 한정을 하여 요청한 모든 패킷이 손실되는 경우를 방지한다. 또한 클라이언트가 Data 패킷을 받는 경우 현재 받은 Data 패킷의 시퀀스가 순서에 맞게 들어왔는지 확인을 하게 되며中间的 패킷이 손실되는 경우 클라이언트의 윈도우를 확인해서 패킷 손실이 일어난 패킷에 대해서만 재전송을 요청해 필요 없는 재전송이 일어나지 않게 한다.

그림 3은 클라이언트가 n이 4인 SInt 패킷을 전송하는 상황을 보여주며, 그림 4는 그림 4의 상황에서 패킷 손실이 일어날 경우의 윈도우 기능을 보여준다. 클라이언트는 SInt(4)를 보내서 CID가 0~3인 Data 패킷을 요청하는 Interest 패킷을 전송을 하고 서버는 이 패킷을 아무 문제없이 받은 후 CID가 0~3인 Data 패킷을 순차적으로 전송을 하지만 CID가 0,2인 패킷은 타임아웃이나 네트워크의 상태가 안 좋아져서 패킷 손실이 일어나 클라이언트는 CID가 1,3인 Data 패킷만을 받게 되었다. 이렇게 패킷 손실이 일어난 경우 클라이언트의 윈도우는 현재 받은 CID를 확인하여 받은 청크를 체크한 후, 못 받은 패킷 0,2에 대해 재전송을 요청한다. 클

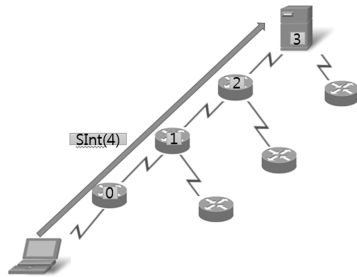


그림 3 SInt 패킷 전송
Fig. 3 Sending SInt Packet

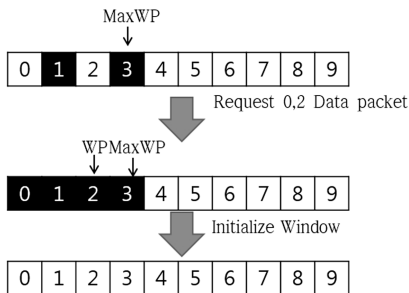


그림 4 윈도우 기능
Fig. 4 Window Function

라이언트는 이상 없이 요청한 모든 패킷을 받게 되면 윈도우를 초기화하고 다음 패킷 손실을 대비한다.

전통적인 CCN에서는 순차적으로 1개의 Interest 패킷으로 1개의 Data 패킷을 받기 때문에 어떤 패킷이 손실되었는지를 클라이언트에서 쉽게 파악할 수 있지만, 본문에서 제안하는 1개의 Interest 패킷으로 n개의 Data 패킷을 받는 기법의 경우 클라이언트가 n개의 패킷 중에 어떤 패킷이 손실되었는지 파악하기가 힘들기 때문에 이를 해결하기 위해, 전통적인 TCP 전송 기법에서 혼잡 제어를 위해 사용하는 윈도우 기법을 사용하여 패킷 손실이나 타임아웃이 발생으로 인해中间的 Data 패킷을 못 받는 경우 이미 받은 패킷을 제외한 못 받은 Data 패킷에 대해서만 재요청 한다.

3.1 CCN Streaming

CCN에서 스트리밍을 사용하는 경우 그림 5와 같이 CCN 서버에 미디어를 제어하는 파트가 들어가게 된다. 서버에서 CCN Repo는 콘텐츠를 저장하고 있는 CCN Repository, ccnd는 CCN에서 라우팅을 가능하게 하는 ccn demon이다. 클라이언트에서 A라는 동영상을 요청한 경우 서버는 A라는 동영상을 인코딩 한 후, Stream Segmenter를 이용해 일정한 시간 간격마다 입력받은 동영상 데이터를 분할해 파일을 만들고, 그 분할한 파일에 접근할 수 있는 메타데이터를 생성하는 일을 한다. 이때 생성되는 Distribution의 A.m3u8은 메타데이터, Block.ts 분할된 파일이다. 이렇게 생성된 메타데이터와 분할된 파일은 CCN Repo에 저장되고 클라이언트에 전송이 된다. 그 후, 요청한 Data 패킷을 받은 클라이언트는 Media Player로 동영상 스트리밍을 재생하게 된다.

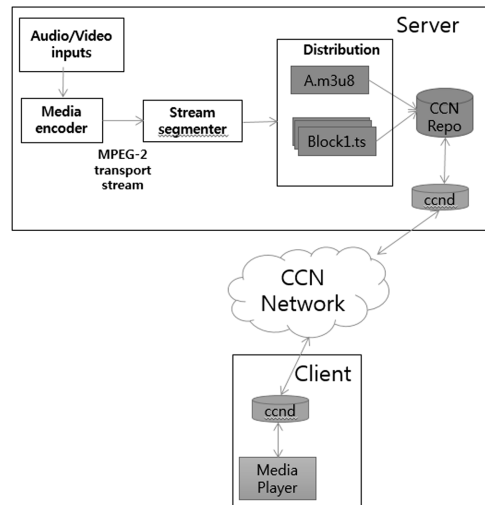


그림 5 CCN 스트리밍의 시스템 모델
Fig. 5 System Model of CCN Streaming

4. 성능 평가

본 논문에서는 VMware 9.0.2에 리눅스 우분투 12.10.x.3을 설치하고 ndnsim을 이용하여 테스트베드를 구축하고 VoD 스트리밍을 이용하여 테스트하였다. 그림 6은 본 논문에서 제안하는 기법의 성능평가를 위해 기존 CCN에서의 Interest 패킷 수와 비교하였다. 그림은 각 콘텐츠 크기에 대하여 전송된 Interest 패킷의 수를 나타낸 그림이며, 청크의 크기는 1KB 기본값을 사용하였다. 그림에서 보이는 것처럼, 기존의 CCN에서는 콘텐츠의 용량이 커질수록 Interest 패킷의 수가 급격히 증가하는 것을 볼 수 있으며 본 논문에서 제안하는 방식에서는 Interest 패킷의 수가 콘텐츠의 용량이 증가하더라도 기존의 CCN 보다는 상대적으로 적은 양의 Interest 패킷을 사용하는 것을 볼 수 있다.

그림 7은 본 논문에서 제안하는 기법을 가지고 실험을 했을 때 발생하는 패킷 손실을 보여준다. 패킷 손실의 경우 클라이언트의 윈도우에서 패킷 손실이 발생하면 카운트를 하였으며, 콘텐츠의 사이즈가 증가할수록 패킷 손실량이 증가함을 볼 수 있는데, SInterest 패킷으로 생성된 여러 개의 Data 패킷으로 인해 네트워크가 혼잡해져서 패킷 손실이 발생하게 되고, 이 손실된 패킷

들을 재전송을 통해 바로 처리하게 된다. 마찬가지로 SInterest 패킷의 n이 증가할수록 요청하는 Data 패킷이 많아질 것으로 예상되기 때문에 SInterest 패킷과 클라이언트의 윈도우의 최적화가 필요하다.

5. 결론 및 향후 연구 방향

본 논문에서는 CCN에서 1개의 Interest 패킷으로 n개의 Data 패킷을 받는 SInterest 패킷을 미디어 스트리밍에 적용하여 보다 효율적으로 데이터를 받는 방법과 패킷 손실이 일어날 경우 클라이언트에서 윈도우를 사용하여 손실된 패킷에 대해서만 재전송을 하는 기법에 대해 설명하였다.

패킷이 손실이 일어날 경우 윈도우가 손실된 패킷을 확인하여 재전송을 요청하지만, SInterest의 n이 증가하면 여러 개의 Data 패킷으로 인해 네트워크가 혼잡해져 패킷 손실이 증가될 것으로 예상되기 때문에, 향후 최적화된 윈도우 사이즈 및 어느 정도의 Data 패킷을 요청해야 패킷 손실이 적게 일어나는지를 찾아내고 이를 기반으로 본 논문의 알고리즘이 스트리밍 서비스의 효율성에 어느 정도 영향을 미치는지에 대한 연구를 할 계획이다.

References

- [1] Kyu-Bo Lim, Jung-Hwan Lee, Chuck Yoo "Efficient Cache Replacement Policy over CCN With SVC Video Streaming," *proc. of the KCC 2012*, vol.39, no.1(D), p.235, 2012. (in Korean)
- [2] Cisco Visual Networking Index: Forecast and Methodology, 2012-2017.
- [3] J.I. Kim, H.Y. Jung, W.G. Park, "Content Centric Networking Technology," *ETRI, Electronics and Telecommunications Trends*, vol.25, no.6, Dec. 2010. (in Korean)
- [4] Nakjung Choi, Jae-Hyun Hwang, "Adaptive video streaming in Content-centric network," *Journal of KICS*, vol.30, no.30, pp.73-79, Mar. 2013. (in Korean)
- [5] CCNx Home Page. [Online]. Available: <http://www.ccnx.org/>
- [6] CCNx Interest Message. [Online]. Available: <http://www.ccnx.org/releases/latest/doc/technical/InterestMessage.html>
- [7] CCNx Content Object. [Online]. Available: <http://www.ccnx.org/releases/latest/doc/technical/ContentObject.html>
- [8] HTTP Live Streaming in Wiki. [Online]. Available: http://en.wikipedia.org/wiki/HTTP_Live_Streaming
- [9] Adaptive bitrate streaming in Wiki. [Online]. Available: http://en.wikipedia.org/wiki/Adaptive_bitrate_streaming
- [10] CCNx Protocol. [Online]. Available: <http://www.ccnx.org/releases/latest/doc/technical/CCNxProtocol.html>

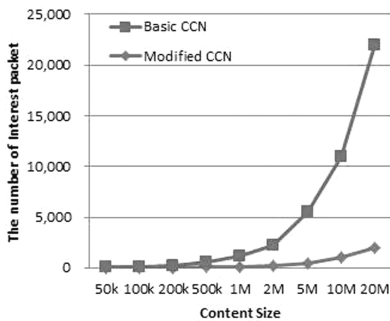


그림 6 전송된 Interest 패킷의 수

Fig. 6 The number of transmitted Interest packet

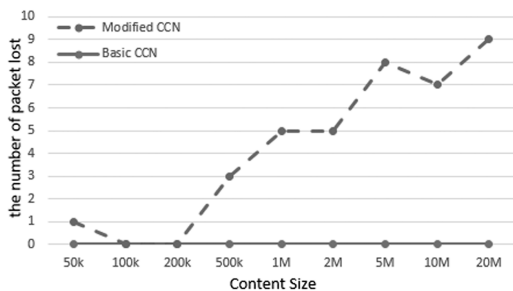


그림 7 손실된 패킷의 수

Fig. 7 The number of packet lost