CCNx 환경에서 동적 메모리 기반 효율적 콘텐츠 제공 기법

이진원○, 홍충선*
경희대학교 컴퓨터공학과
{notwonz, cshong}@khu.ac.kr

Dynamic Memory Based Efficient Content Provide Mechanism in CCNx

Jinwon Lee○, ChoongSeon Hong*
Department of Computer Science and Engineering, Kyung Hee University

요 약

현재 인터넷은 호스트 기반 통신으로 정보의 위치가 정보 자체를 나태는 구조로 구성되어 있다. 이러한 구성은 병목현상, 불필요한 반복전송 등과 같은 문제점을 야기할 수 있다. 이에 따라 현재 인터넷 모델의 문제점을 개선할 수 있는 콘텐츠 중심 네트워크(Content Centric Network)와 같은 콘텐츠 및 정보 기반의 네트워크가 등장하였다. 본 논문에서는 Content Centric Network의 구성 요소 중 콘텐츠를 저장하고 전송하는 콘텐츠 저장소(Content Store)에서 사용자가 요청한 콘텐츠를 검색 후 전송할 때 발생할 수 있는 지연의 원인을 파악하고, 이러한 지연을 최소화 할 수 있는 기법을 제안 및 평가한다.

1. 서 론

현재의 컴퓨터 네트워크 모델은 1970년대에 분산된 사용자들에게 서비스를 제공하기 위해 개발된 TCP/IP 기반의 호스트 중심 모델이다. 컴퓨터 네트워크에 관여하는 당시의 장비들은 이동이 어려웠고 사용자들이 요구하는 서비스가 단순하였기 때문에 TCP/IP 통신은 정보의 위치가 정보 자체를 나타내는 구조를 가지고 있었다. 하지만 이동 단말의 급격한 증가와 사용자들이 요구하는 콘텐츠가 매우 다양해짐에 따라 기존의 TCP/IP 통신은 병목현상과 불필요한 반복전송 등의 여러 가지 문제점들을 가지게 되었다[1]. 기존 네트워크 모델의 문제점을 해결하기 위해, 요구하는 정보의 위치가 아닌 정보 자체가 중요해지는 새로운 네트워크 패러다임의 필요성이 대두되었다. 이에 따라, 정보 중심 네트워크(Information Centric Network), 콘텐츠 중심 네트워크(Content Centric Network, 이하 CCN) 등의 모델이 등장하게 되었다. CCN은 2009년 Van Jacobson에 의해 제안되었으며[2] IP 대신에 Content Name을 이용해 통신하기 때문에 TCP/IP 통신에 비해 효율적인 통신을 가능하게 하는 특징을 가지고 있다[3].

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (B0101-16-0033, CCN 기반의 다차원 scalability를 활용한 5G 이동통신 기술 연구개발)
*Dr. CS Hong is the corresponding author

한편, 최근에 등장하는 대부분의 이동 단말은 인터넷을 지원하고 구입비용이 감소하여 네트워크에 대해 접근성이 향상되었다. 이에 따라 여러 가지 콘텐츠들을 요구하는 수요가 급진적으로 증가하였고 콘텐츠의 크기에 따른 지연 발생 문제의 해결책 개발을 위한 연구가 활발히 진행되고 있다. 본 논문에서는 CCNx 환경에서 동적 메모리 기법을 기반으로 콘텐츠 전송 시 효율적으로 전송할 수 있는 성능 개선 기법을 제안하며 구성은 다음과 같다. 2장은 CCN, CCNx, 동적 메모리 할당에 대해서 소개하고 3장에서는 기존 연구의 문제점 및 제안하는 알고리즘에 대해서 소개한다. 4장은 제안하는 알고리즘을 바탕으로 한 성능 테스트 수행 및 평가하며 5장은 본 논문의 결론으로 구성하였다.

2. 관련연구

2.1 Content Centric Network

[2]에서 언급한 CCN은 기존의 TCP/IP 통신과는 다른 네트워크 모델이다. TCP/IP 통신은 콘텐츠의 위치를 나타내는 IP Address를 사용하지만 CCN은 어떤 콘텐츠를 원하는지를 나타내기 위해 Content Name을 사용한다. 따라서 CCN Client는 Content Name을 사용하여 원하는 콘텐츠를 CCN Server의 Content Store(이하 CS)에서 찾기 때문에 효율적인 통신을 가능하게 한다.

CCN은 2종류의 Packet을 사용하는데 콘텐츠를 요청할

때 사용하는 Interest Packet과 요청에 응답하고 콘텐츠를 전달하기 위해 사용하는 Data Packet이 있다. 또한, CCN은 콘텐츠 제공, 콘텐츠 요청, 콘텐츠 전달의 역할을 하는 노드들로 나뉜다. 각 노드들은 콘텐츠를 저장하는 Cache인 CS, 콘텐츠를 요청하는 Interest Packet의 기록을 저장하는 Cache인 Pending Interest Table(이하 PIT), Interest Packet의 Forwarding을 위한 Cache인 Forwarding Information Base(이하 FIB)로 구성된다. CCN Client의 콘텐츠 요청을 빠르게 처리하기 위해서는 CS, PIT, FIB와 같은 Cache들의 처리속도가 매우 중요하다 [1].

2.2 CCNx

CCNx는 Palo Alto Research Center(PARC)의 CCN을 기반으로 한 유닉스 계열 운영체제에서 사용가능한 오픈 소스 프로젝트이다.

2.3 동적 메모리 할당

동적 메모리 할당은 컴퓨터 프로그래밍에서 실행되는 시간 동안 사용할 메모리의 공간을 할당하는 것을 말한다. 동적 메모리 할당은 해당 프로세스의 힙 영역에서 할당한다. 동적 할당을 한 프로세스가 계속해서 실행될 때 동적 할당 된 영역은 유지되므로 해당 프로그램의 힙 영역의 크기를 넘는 메모리 할당을 요구해서는 안 된다.

3. 기존 연구의 문제점 및 제안사항

CCNx에서 콘텐츠를 요청하기 위해 CCNx Client는 원하는 Content Name을 포함한 Interest Packet을 전송한다. Interest Packet을 수신한 CCNx Server는 Content Name을 바탕으로 해당 콘텐츠가 CS에 존재하는지를 판단한다. CS에 Interest Packet의 Content Name과 같은 콘텐츠가 존재하면 CCNx Server는 CCN Input Stream으로 해당 콘텐츠를 Byte 타입의 Buffer Stream으로 복사한다. Buffer Stream으로 복사한 콘텐츠를 다시 File Output Stream을 통해 CCNx Client에게 전달한다. 콘텐츠를 Buffer Stream을 통해 복사하는 과정에서 기존의 CCNx 프로젝트는 고정 크기의 사이즈로 해당 콘텐츠를 분할하여 복사한다. 하지만 이 방법은 콘텐츠의 크기가 증가할수록 복사하는 횟수가 증가하게 되어 전송 지연이 발생하는 문제점을 가지고 있다. 서론에서 언급한 것과 같이 현재의 인터넷 트래픽은 대용량 파일과 스트리밍 서비스의 전송이 대부분을 차지하고 있다. 따라서 기존의 콘텐츠 제공 방식은 대용량 콘텐츠의 전송 시 지연을 가져올 수 있으며 저사양의 장비에서 시도할수록 더 많은 지연을 가져올 수 있다.

본 논문에서는 CCNx 프로젝트에서 CS에 매칭된 콘텐츠를 Buffer Stream으로 복사하는 ccngetfile.java 소스코드에 제안하는 알고리즘을 적용한다. 제안하는 알고리즘은 네트워크 장비의 힙 메모리 영역을 초과하지 않는 범위 내에서 Buffer Stream의 크기를 지속적으로 동적 할당하여, 콘텐츠 전송 시 Buffer Stream의 최적의 사이즈를 찾는다. 제안하는 알고리즘의 의사코드(pseudocode)는 그림 1에 나타나 있다.

```
Proposed Algorithm
Random Large Integer(R) (2^2 + 1) to 2^3, Random Small Integer(S) 2^1 to 2^2
Byte Buffer Array(A), Size of Buffer(B)
Input. Content Request
Output. Matched Content
Method
    1: Declaration of A and B
    2: Set A as new array with B
    3: while ( End of Content)
    4: Generate R, S
    5:   Read Content with CCN Input Stream and A
    6:   Send the Content of A using File Output Stream
    7:   if (End of A is Not NULL) then
    8:     if (RB < Max of Heap Memory) then
    9:       B ← RB and Reset File Output Stream
   10:     else
   11:       Reset File Output Stream
   12:     end if
   13:   else
   14:     if (B/S > Min of default B) then
   15:       B ← B/S and Reset File Output Stream
   16:     else
   17:       Reset File Output Stream
   18:     end if
   19:   end if
   20: Set A as new array with B
   21: end while
```

그림 1. 제안하는 알고리즘의 의사코드

동적 할당을 위한 변수인 Random Large Integer와 Random Small Integer의 범위를 각각 (2^2+1)에서 2^3, 2^1에서 2^2로 설정한 것은 해당 장비에서 CCNx 프로세스만 실행되는 것이 아니기 때문에 기존의 크기보다 10배를 넘지 않는 범위 내에서 동적 할당하여 다른 프로세스들이 Heap 영역을 사용할 수 있도록 하기 위함이다. 또한, Buffer의 사이즈를 다시 줄이는 분기문(Statement)이 필요한 이유는 분기문을 통해 최대 사이즈와 가장 가까운 사이즈를 유지하여, 한 번 참조된 콘텐츠 및 데이터는 잠시 후에 또 참조될 가능성이 높다는 시간 지역성(Locality)의 원리를 적용하기 때문이다. 기존처럼 고정 크기 S_s로 전체 T_c 크기의 콘텐츠를 분할하여 복사한다

면 T_{basic} (기존 방식의 분할 시도 횟수)는 T_c / S_s를 만족하는 정수로 나타낼 수 있고 가변 크기 S_{di}을 사용하는 제안하는 알고리즘을 적용하면 $T_{proposed}$ (제안하는 방식의 분할 시도 횟수)는 $\sum_{i=1}^{n} S_{di} \geq Tc$ 을 만족하는 정수 n으로 나타낼 수 있다. T_{basic} 와 $T_{proposed}$ 의 차이는 대용량 파일 전송 시에 더욱 커지게 된다.

4. 성능평가

4.1 테스트 환경

Virtualization Tool	Oracle Virtual Machine 5.0.18
Operating System of Server	Ubuntu 12.04.5
Operating System of Client	Ubuntu 12.04.5
CCNx Version	CCNx-0.8.2
테스트 파일의 크기	100MB,400MB,600MB, 1000MB

표 1. 테스트 환경

표 1의 환경을 바탕으로 그림 2와 같은 테스트베드를 구성하였다.

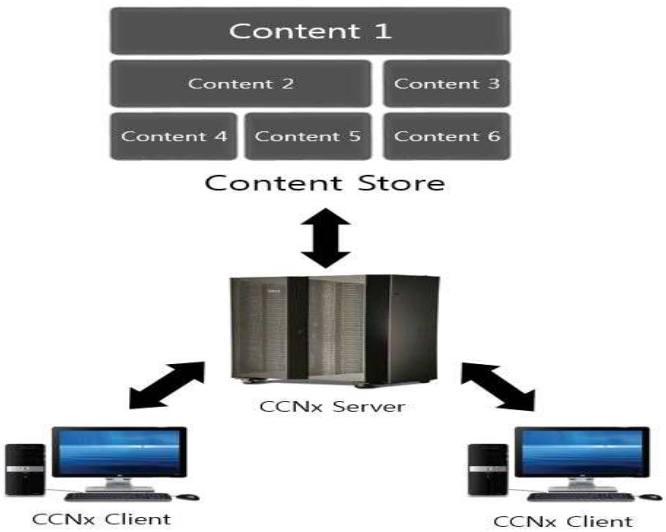

그림 2. 테스트베드

테스트 시나리오는 CCNx Server에서 100MB, 400MB, 600MB, 1GB의 테스트 파일을 CS에 저장하고 CCNx Client가 저장된 각각의 콘텐츠를 요청하는 순서로 진행된다. 평가는 기존의 방식과 제안하는 알고리즘이 적용된 방식으로 나눠 각각 용량별로 10 번씩 시도한 뒤 콘텐츠를 수신하는데 걸린 시간을 평균 내어 비교한다.

4.2 테스트 결과

그림 3에서 CCNx Server의 CS에서 콘텐츠를 동적 크기로 Buffer Stream에 복사하여 Client에게 전달하는 것이 고정 크기로 복사하는 기존의 방식보다 좋은 성능을 보였다. 제안하는 방식은 용량이 적은 파일의 경우에도 좋은 성능을 나타내지만 파일의 크기가 커질수록 성능의 차이는 더욱 커지는 것을 알 수 있다.

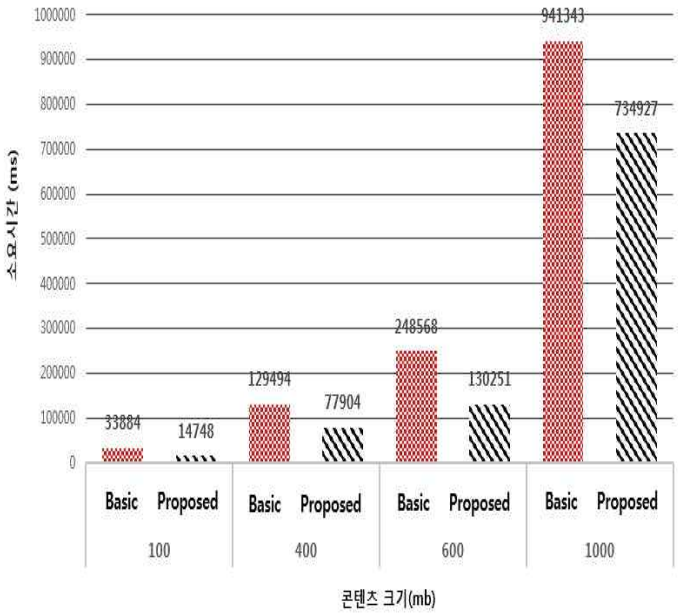

그림 3. 콘텐츠 크기에 따른 콘텐츠 수신 결과

5. 결론

본 논문은 기존 TCP/IP 네트워크의 문제점을 해결하기 위해 등장한 Content Centric Network 패러다임을 바탕으로 한 오픈소스 프로젝트인 CCNx 환경에서, 콘텐츠를 효율적으로 전송하기 위해 CS에 저장되어 있는 콘텐츠를 복사하는 Buffer Stream의 크기를 지속적으로 동적 할당하여 점진적으로 최적의 크기를 찾는 알고리즘을 제안하였으며 이를 통하여 Client가 콘텐츠를 요청할 때 기존의 방식보다 효율적으로 콘텐츠를 수신할 수 있게 하였다.

참 고 문 헌

[1] 최상일, “Content Centric Networking 개요,” pp.1-9, July 2013.

[2] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, Rebecca L. Braynard, “Networking Named Content,” ACM CoNEXT 2009, pp. 1-12, December 2009.

[3] Donghyun Son, Daejin Choi, Donghyun Kim, Youngbin Im, Jinyoung Han, “An Efficient Data Structure for Content-Centric Networks,” 한국통신학회 학술대회논문집, Vol.2013, No.6, pp. 81-82, 2013.