

효과적인 웹-캐싱 네트워크 구축전략에 관한 실증 연구

이 주 헌* · 조 병 룡**

An Empirical Study on the Construction Strategy of Web-caching Network

John Hearn Lee* · Byeong Ryong Cho**

Abstract

Despite the growth in Internet users, demand for multi-medial, large data files and resulting explosive growth in data traffic, there has been lack of investment in Middle-Mile, interconnection of various networks, resulting in bottleneck effect, which is acerbating. One strategy to overcome such network bottleneck is Content Delivery Network (CDN). CDN does not achieve efficient delivery of large file data through physical improvement/increase in network capacity, but by delivering large file contents, the cause of bottlenecks, from distributed servers. Since it is impracticable to physically improve networks capacity to accommodate the growth in Internet traffic, CDN, by strong CPs contents at cache servers deployed at major ISPs networks, is able to deliver requested contents to the requesting Web clients without the loss of data and long latency.

* 한국외국어대학교 경영학과 교수

** ㈜씨디네트웍스 개발팀 과장

1. 서 론

인터넷은 전 세계적인 컴퓨터 네트워크 시스템으로서, 이용자가 어떤 컴퓨터에 있는지에 상관 없이 그가 사용권한을 가지고 있다면 그 어떤 다른 컴퓨터에도 접속해서 정보를 얻을 수 있는 네트워크의 네트워크이다[22]. 이러한 인터넷은 월드-와이드-웹(이하 WWW)이 제공하는 사용 용이성과 편리성 때문에 오늘날의 인터넷 열풍을 몰고 오게 되었다[15].

세계적으로 인터넷에 접속된 호스트의 수로 볼 때 1990년 10월에 31만 여 대, 1997년 1월에는 1600만 여 대로 증가되었으며[7] 매년 50~200%의 급격한 증가율을 보이고 있다[10]. 우리나라에서도 2001년 6월말 현재, 7세 이상으로서 월평균 한번 이상 인터넷을 이용하는 사람은 모두 2,223만명(이용률 51.6%)으로 지난 2001년 3월말 2,093만명(48.6%)에서 3개월 동안 130만명(3.0%)이 증가하였다[21].

그런데, 이렇게 발달하는 인터넷과 더불어 인터넷을 통해 다양한 정보를 서로 공유 하고자 하는 사람들의 요구가 증가함에 따라 심각한 문제들이 발생하기 시작하였다. 즉, 한정된 네트워크의 대역폭과 이용자의 요구를 처리하는 능력 등이 기하급수적으로 증가하는 이용자의 수많은 요구에 부응하지 못하게 된 것이다.

WWW은 한 서버에서 제공한 자료를 인터넷에 널리 있는 많은 클라이언트들이 접속해서 가져가는 클라이언트-서버 모델을 기본으로 한다. 클라이언트들이 원하는 자료가 있을 때마다 일일이 서버에 연결을 요청하고 가져오는데, 서버는 이때마다 같은 자료를 반복해서 보내주게 된다. 그 결과 네트워크 상에서의 서버와 클라이언트들 상호 간의 거리 개념은 완전히 무시된 채 불필요하게 재전송을 거듭해서 네트워크 대역폭

을 낭비하고, 정보를 제공하는 서버는 요청하는 모든 클라이언트들에게 직접 답해야만 한다. 즉, WWW은 원천적으로 규모의 변화에 효율적으로 대처하지 못하는 모델 상의 한계를 가지고 있다[12]. 이와 같은 이유로 현재의 인터넷은 다음과 같은 문제점을 내포하고 있다.

첫째는 네트워크의 오버로드(overload)이다. 위에서 언급 한대로 수많은 클라이언트의 요청과 요청 대상이 되는 웹 문서 안에 포함된 대용량의 멀티미디어 데이터에 의해 트래픽량이 증가한다. 그러나 네트워크는 본질적인 대역폭의 제한으로 이러한 요구를 쫓아갈 수 없다.

둘째는 서버의 오버로드이다. 오늘날의 웹의 특징은 유명한 웹 페이지를 위주로 접속한다는 것이다. 따라서, 서버 이용이 증가하고 그에 따라 서버의 과부하를 유발하므로 추가적으로 하부 구조를 연구해야 한다.

셋째는 대기 시간(latency time)의 증가이다. 네트워크와 서버의 오버로드는 문서의 정보 검색 시간을 증가시킨다. 이러한 대기 시간의 증가는 자칫 이용자들을 지루하게 할 수 있고, 그렇게 되면 신속하게 정보를 제공할 수 있는 다른 사이트의 웹 페이지에 이용자를 빼앗길 수도 있다[5].

위에서 열거한 이러한 문제를 해결하는 가장 일반적인 방법은 이용자가 자주 요구하는 파일에 대해 복사본을 서버로부터 이용자와 가까운 지점으로 위치시키는 캐싱(caching)이다. 캐싱은 이용자들이 반복적으로 요구하는 파일의 복사본을 두어 제공함으로써 효과적으로 대처할 수 있다.

이렇게 함으로써 웹 이용자는 URL을 요청하고 난 후의 지연시간이 단축됨을 느끼고, 망 관리자는 통신량(traffic)이 감소됨을, 그리고 웹 서버는 오브젝트 요청 비율이 낮아짐을 알 수 있다.

이를 네트워크 상에 응용해서 만든 것이 네트

웹 캐시이다. 네트워크 캐시는 정적인 정보들을 대상으로 하는 미러링(mirroring) 보다는 동적인 성격이 강한 WWW 서비스에 적합한 방법이다.

그러나 인터넷 캐시는 낮은 캐시 적중률(hit ratio), 캐시 일관성(consistency) 유지의 어려움, 캐시 되는 파일 크기의 다양성 등의 이유로 기존에 연구되었던 캐시와는 많은 차이점이 있기 때문에 새로운 방향으로 연구되어야 한다.

웹에서의 캐싱은 여러 가지 형태로 사용된다. 첫 번째는 클라이언트 캐시로, 웹 브라우저 안에 구축한다. 캐싱의 두 번째 형태는 여러 이용자가 공유하면서 사용하는 것으로 망 안에 존재한다. 캐시는 다수의 클라이언트에서 다수의 서버 경로간에 있는 한 기계상에 위치한다. 이를 프락시(proxy) 캐시라고 하는데 다수의 클라이언트가 요청한 URL을 캐시한다[11, 17].

위에서 언급한 바와 같이 네트워크와 서버의 오버로드, 낮은 캐시 적중률, 캐시 일관성 유지의 어려움, 한정된 네트워크 대역폭 등과 같은 웹 환경의 한계 속에서 근래처럼 급격히 생겨나는 인터넷 방송국, 영화관, 쇼핑 사이트 등 대용량 실시간 멀티미디어 자료들이 홍수처럼 쏟아져 나오는 시대에서 현재의 네트워크 인프라와 캐싱 전략 만으로는 그 추세에 대응할 수 없다. 이에 본 연구는, HTTP 콘텐츠에서 멀티미디어 중심의 콘텐츠로 변화되는 시점에서 대용량 정보의 신속한 전송을 위해 기존의 캐싱 방법에 대해 보다 전략적인 접근과, 그 구현 해법을 제시하고자 한다.

2. 캐시 분석

2.1 캐싱(caching)과 미러링(mirroring)

캐싱은 본래 복사된 데이터를 느린 메인 메모

리에서 상대적으로 빠른 CPU로 이관 시킴으로써 처리속도와 성능향상을 위한 컴퓨터 아키텍처상의 하드웨어적 개념이다.

이러한 개념은 네트워크로 확산되어 프락시(proxy) 캐시의 발전을 가져왔다. 수 많은 이용자들이 웹 상에서 동일한 캐시를 사용한 이후로, 요청된 콘텐츠는 더욱 빨리 제공되었다. 더군다나 협력적인 캐시들은 다른 캐시들을 통해 미스(miss)된 콘텐츠들을 해석함으로써 WAN(Wide Area Network)의 네트워크 대역폭 소비를 감소시킬 수 있었다.

ICP(Internet Cache Protocol)는 콘텐츠를 제공하기 위한 가장 적당한 위치를 선택하여 정보를 수집하도록 사용되었는데, 이는 협력된 캐시 서버들의 집합에서 캐싱을 분산함에 따라 더욱 효율성을 증가시켰다[5].

미러링이란 특정 지역의 이용자들이 더욱 빠르게 액세스 할 수 있고, 원래 사이트에 집중되는 부하를 분산시키기 위해 한 사이트에 있는 파일의 일부 혹은 전부를 복사하여 상대적으로 여유 있는 다른 웹사이트에 저장, 운영하는 것이다. 이러한 미러링은 자신에게 요청되었던 모든 것에 대한 복사본을 유지하는 캐시나 프락시 서버와는 달리, 대체로 특정 원격 서버에 있는 전체 디렉토리나 파일들에 대해 이루어진다. 주로 크기가 크고 상대적으로 쉽게 변경되지 않는 정적인 성격을 가지면서 많은 이용자들의 요구를 받는 자료들이 미러링의 대상이 되며, 실제 인터넷 상에서 공개 소프트웨어나 문서의 배포 과정 중에서 흔히 발생할 수 있는 네트워크 부하나 서버 부하를 줄이는 데 이용된다.

그러나 WWW이 보급되고 폭발적인 성장을 하면서 몇 가지 근본적인 문제가 발생 하였다. 이 문제들은 WWW의 모델과 정보의 성격에서 비롯된 문제인데, WWW으로 제공되는 정보들

은 FTP로 제공되는 정보와는 달리 크기가 작은 경우가 많고 변경되는 횟수는 훨씬 잦다. 그리고 URL을 이용한 하이퍼텍스트/하이퍼미디어 모델을 따르고 있어서 서버에서 참조하고 있는 자료들이 동일한 서버가 아닌 다른 여러 서버에 저장되어 있는 경우가 많은데, 이 때 한 서버에 저장된 자료만을 미러링 하는 것은 미러링 원래 의미에 맞지 않고, 그렇다고 해서 다른 서버에 저장된 자료까지 가져 오자면 그 범위나 대상이 끝이 없는 등 결과적으로 미러링은 WWW에 적합하지 못하다[2].

또한 WWW으로 제공되는 정보의 특징은 크기가 작은 정보가 많고 변경되는 횟수가 잦은 것 외에도 다른 정보들에 비해 특별히 자주 요구되는 정보들이 많다.

이렇게 자주 요청되는 정보들을 각 이용자들에게 광역 네트워크를 통해 반복적으로 재전송하므로 네트워크 대역폭이 낭비되어 병목 현상이 일어나고 서버에 과부하가 걸리며 대기 시간이 길어진다.

이러한 문제를 해결하기 위한 대안으로서 인터넷 캐시는 캐시를 설치할 위치와 그 구성방법에 따라 다음과 같이 클라이언트 캐시, 서버 캐시, 프락시 캐시, 계층적 캐시로 구분할 수 있다[3].

2.2 클라이언트 캐시

콘텐츠를 캐시하는 한 가지 방법은 클라이언트 로컬에 캐싱하는 것이다. 요즘의 대부분의 웹 브라우저는 비지속성 캐싱과 지속성 캐싱을 제공한다[18]. 비지속성 캐싱은 최근에 사용된 파일이 브라우저가 실행되고 있는 동안 주 메모리에 저장한다. 지속성 캐싱은 현재 세션 이외에 다음 세션에서도 사용할 수 있도록 파일을 디스크에 저장한다. 요청된 파일이 웹 브라우저

의 캐시에 없으면 요청은 망을 통해 서버로 전달된다.

클라이언트의 캐시 적중률은 캐시의 크기를 늘림으로써 증가시킬 수 있다. 하지만, 현실적으로는 디스크 상에 한계가 있고 자칫 디스크의 낭비를 초래할 수도 있다.

2.3 서버 캐시

서버 캐시는 망 전송 거리를 감소 시키지는 않지만, 캐시 서버가 부하(load)를 공유함으로써 서버의 부하를 감소시키는 것이다. 망에만 과부하가 발생하는 것이 아니라 서버에도 과부하가 발생하기 때문에 이러한 서버 캐시를 이용하여 부하를 분산할 필요가 있다[4].

2.4 프락시 캐시

병목현상 해소를 통하여 네트워크 교통량을 줄이고 서버의 확장성을 향상시키는 효과적인 방안의 하나로써, 자주 요청되는 인기있는 콘텐츠를 이용자 노드의 가까운 곳에 캐싱하는 기법이 다[8].

프락시 캐시는 클라이언트와 서버간의 중계자 역할을 한다. 이용자의 모든 요청은 프락시 캐시에게 전송되어 프락시 캐시가 응답한다. 프락시 캐시는 다수의 클라이언트가 요청한 URL을 캐시한다. 프락시 캐시는 여러 이용자가 공유를 하기 때문에 캐시의 이용자가 비슷한 유형을 가질 때 적중률은 높아진다. 따라서 각 클라이언트에 캐시하는 것보다 망 내에 하나의 캐시를 공유하면 더 효과적으로 이용할 수 있다. 캐시에서 캐시 크기가 크면 더 효과적이라는 것은 명백하다. 만약에 한 이용자가 10MB의 클라이언트 캐시를 사용한다면 200명의 이용자들은 총 2GB의 디스크를 사용한다, 그런

데, 만약 이 2GB의 캐시를 모든 이용자가 공유한다면 캐시 적중률은 상당히 증가하고 효율적이 된다.

2.5 계층적 캐시

프락시 캐시는 다른 프락시 캐시의 URL을 캐시하기 위해 계층적인 프락시 캐시를 사용할 수 있다. 이와 같은 경우에 캐시는 첫 번째 단계 캐시, 두 번째 단계 캐시 등으로 구별하며 이를 계층적 캐시라 한다[6].

단일 프락시 서버를 사용하면 캐시는 수평(flat) 구조를 형성한다. 그러나 이용자의 수가 증가할 때 프락시 서버에도 과부하가 발생하여 확장성 문제가 발생한다.

계층적 캐시(hierarchical cache)는 여러 개의 프락시 서버를 트리 형태의 계층적 구조로 연결 시킴으로서 확장성을 갖게 한다.

계층적 캐시의 대표적인 것은 하비스트 캐시(harvest cache)이다. 하비스트 캐시는 이용자들이 정보 요구를 한 후의 대기 시간을 최소화하고 성능을 향상시키기 위해 여러 가지의 기능적인 특징과 다양한 선택 사항을 제공하고 있으며, 그 중에서도 가장 주목할만한 것은 캐시들이 네트워크를 통해 서로 가지고 있는 자료들을 공유하는 시스템을 구성할 수 있는 기능이다.

하비스트 캐시는 다른 캐시 소프트웨어에 비해 뛰어난 성능과 다양한 기능으로 많은 주목을 받았고, 이후 하비스트 캐시, 스쿼드 캐시 등 두 가지 방향으로 발전하게 되었다[15].

스쿼드(squid)는 클라이언트들을 대신해 FTP, Gopher, HTTP 콘텐츠들을 추출해 내는 프락시 서버로 기본적으로 유닉스 시스템을 위하여 개발되었다[16]. 스쿼드의 소스는 GPL(General Public License)에 의해 공개되었으며 현재 많은 개발자들에 의해 발전되고 있다.

3. 웹-캐싱 전략

이용자와 사이트의 급속한 증가에 따른 저장 서버와 네트워크의 교통량의 문제를 해결하기 위해서는 데이터의 저장과 전송에 필요한 저장 서버 및 네트워크 자원의 증설, 특히 백본 네트워크 자원의 증설이 필요하다.

그러나 단지 이들 만으로는 증가하는 데이터 요청 및 네트워크 교통량 처리에 효과적이지 못하다. 왜냐하면 현재의 추세 대로라면 인터넷 이용자와 사이트의 증가가 급속하게 이루어지기 때문에 자원을 추가한다고 하더라도 수요증가 속도를 따라잡을 수 없기 때문이다[13, 19]. 만약에 저장 공간이나 전송 대역폭이 무한하다면 중앙 서버에 모든 콘텐츠를 저장하고 서비스 하는 것이 가능하지만 현실적으로 이러한 환경은 존재하지 않는다[3].

결과적으로 아래에 열거한 문제들이 남아있다.

3.1 인터넷 성능에 대한 불만

웹사이트들은 콘텐츠의 전송 속도에 대해 네티즌들이 불만을 갖고 지루해 한다는 점을 익히 알고 있으며 이 같은 문제를 개선하려 노력한다.

온라인 업계에 관해 시장조사를 수행하는 조나 리서치(Zona Research)의 수석 경제학자 잭 스태프(Jack Staff)는 “궁극적으로 이용자들은 인터넷이 자신들의 TV 처럼 되기를 원한다”고 진단했다. 간편한 사용법과 다양한 콘텐츠, 그리고 콘텐츠의 신속한 전달을 원한다는 해석이다.

조나 리서치가 제시한 예에 따르면 온라인 기업들은 내려받기에 시간이 걸리는 페이지들로 인해 매년 30억 달러의 손실을 입을 수 있다고 한다.

3.2 미들마일(middle-mile)의 문제점 대두

트래픽은 매 4~6개월마다 2배로 증가하는데 반해 네트워크 인프라는 매 18개월마다 2배가 되므로 늘어나는 트래픽 수요를 충족시키지 못한다. 또한 늘어나는 트래픽 수요를 충족시키기 위해서는 천문학적인 투자비가 소요된다.

미들마일(middle-mile)이란 가입자망 노드에서 CP(Contents Provider) 웹서버에 이르는 구간을 말한다.

(그림 1)을 보면, 인터넷 속도가 느려지는 원인은 크게 컴퓨팅(computing), 액세스(access), 그리고 네트워크(network) 요인으로 나누어 볼 수 있다.[9] 1995년과 1999년을 비교해 보았을 때 컴퓨팅과 액세스는 비약적인 발전을 보였으나 네트워크 상의 문제는 여전히 존재함을 알 수 있다. 현재 추세로 나간다면 2003년에는 이러한 문제가 더욱 심화될 것이다.

3.3 현재의 캐싱 전략의 문제점

캐싱 전략(caching policy)이란 캐시 시스템 설치의 주된 목적, 다양한 측면을 가진 캐시의 성능이나 사용 권한을 조절하기 위해 운영자가

결정하고 실행하는 일련의 사항들이다[2].

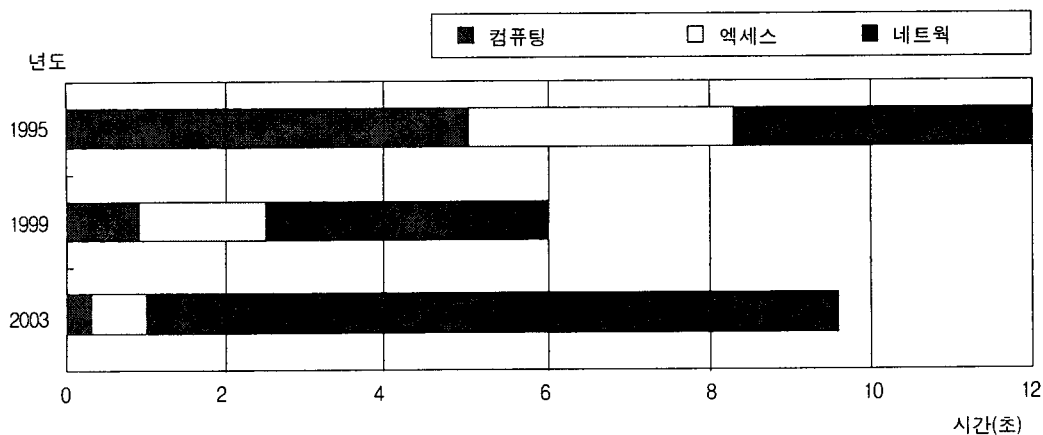
캐싱 전략은 캐시 서비스를 하는데 있어 성능 향상과 속도 개선 등과 직결되는 것으로 좀더 지능적인 캐시를 만들기 위한 발판이 된다. 따라서 주어진 환경에 적절하고 성능을 최적화 할 수 있는 캐싱 전략을 세우는 것이 중요하다.

점점 대용량 콘텐츠의 증가를 필요로 하는 조건 하에서 현재의 캐싱 전략은 다음의 두 가지 이유에서 적당치 못하다[5].

첫째, 대용량 파일들(예를 들어, 10 메가 이상들의 콘텐츠)은 네트워크상의 많은 대역폭과 높은 전송 시간 등으로 인해 현재의 전략으로서는 캐시되지 못한다.

둘째, 만약 대용량의 오디오, 비디오 그리고 응용 콘텐츠들이 작은 콘텐츠 파일들처럼 모든 프락시 서버에 저장된다면 복제와 같은 방법들은 그들을 저장하는데 많은 용량이 필요하기 때문에 많은 저장 비용이 필요할 것이다.

또한 제한된 용량의 캐시 경우에, 대용량 파일들은 많은 작고 유명한 콘텐츠들을 없애버림으로써 미스(Miss)가 나올 확률과 대기시간을 증가시킨다(<표 1> 참조).



(그림 1) 인터넷 속도 체증 요인

〈표 1〉 기존의 캐싱 전략

현재의 인터넷 실정	기존 캐싱전략의 방향
<ul style="list-style-type: none"> • 다양한 파일타입의 콘텐츠들이 존재함. • 콘텐츠 타입마다 가지고 있는 특징이 다양함. • 캐시가 설치된 네트워크의 대역폭과 이용자들의 요구 패턴 등 환경적 요인이 다양함. 	<ul style="list-style-type: none"> • 콘텐츠의 다양성을 고려하지 않고 모든 파일을 동일하게 취급하여 캐싱함. • 명중률, 일관성, 대기 시간과 같은 성능 평가 기준들 각각에 관련된 캐싱 전략이 체계적으로 정리되어 있지 않음. • 명중률 향상을 위한 삭제 전략이나 일관성 문제에 집중되어 있음.

3.4 소용량 콘텐츠 캐싱 전략

HTTP 는 전송되는 파일의 타입을 구분하기 위하여 MIME(Multipurpose Internet Mail Extensions)을 사용한다. MIME 에서는 파일의 타입을 7개의 콘텐츠 타입(content-type)과 각 콘텐츠 타입마다 여러 개의 서브 타입(sub-type)으로 나눈다.

텍스트 타입에는 html, htm, txt 와 같은 확장자를 가진 파일이 포함되고, 이미지 타입에는 gif, jpeg, tiff, xbm 등의 확장자를 가진 파일이 포함된다.

텍스트 타입과 이미지 타입의 파일로 대표되는 소용량 콘텐츠는 그 특성상 대용량 콘텐츠와 달리 변경이 잦고 크기가 작으므로 미리 캐시 서버에 콘텐츠를 미리 가져다 놓지 않는다. 따라서 캐시 서버에 콘텐츠가 존재하지 않음으로써 처음 한번은 Origin 서버에 직접 접속하여 콘텐츠를 전송 받지만 그 후로는 캐시 서버에 그 정보가 저장되어 있기 때문에 두 번째 요청부터는 캐시 서버에서 이용자가 요청한 콘텐츠를 서비스 한다.

3.5 대용량 콘텐츠 캐싱 전략

크기 가중 캐시 적중률은 이용자가 요청한 전체 바이트량에서 캐시에서 적중된 바이트량의 비율을 나타내는데[4], 캐시에 크기가 큰 파일

을 저장하지 않고 이용자가 많이 사용하는 크기가 작은 파일을 저장한다면 캐시 적중률은 상당히 증가할 것이다. 그러나 크기가 큰 파일이 캐시에서 적중되면 크기 가중 캐시 적중률이 많이 증가한다. 이것은 작은 파일들을 요청함으로써 망 통신량이 증가할 수도 있지만, 크기가 큰 파일을 몇 번만 액세스 하더라도 망 통신량이 많이 증가함을 의미한다.

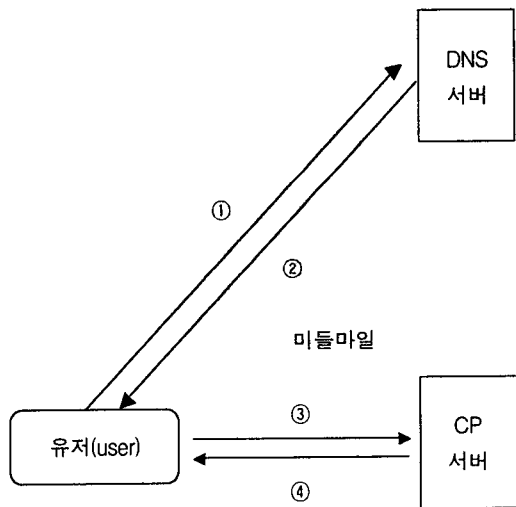
텍스트 파일의 크기는 다른 타입의 크기보다 작기 때문에 액세스 횟수가 많더라도 전체 전송 바이트량은 크기가 큰 다른 타입의 파일에 비해 적게 나타난다. 따라서 캐시 내에 텍스트와 이미지만을 저장하면 모든 파일을 저장할 때와 비교하여 캐시 적중률은 비슷하게 유지되더라도 크기 가중 캐시 적중률은 상당히 감소한다[4]. 즉, 이용자들이 많이 액세스 하지 않는 비디오, 오디오 등 다른 대용량 타입의 파일도 캐싱전략을 수립할 필요가 있다.

대용량 파일은 주로 위에서 언급한 미디어(스트리밍) 파일이나 다운로드(FTP) 파일이다. 이러한 파일들을 이용자의 요청 시 마다 Origin 서버에서 전송한다면 트래픽의 체증을 유발한다. 대용량 파일의 특성은 소용량 파일과는 달리 변경이 그다지 많이 이루어 지지 않는다. 이러한 대용량 파일들을 캐싱하지 않는다면, 유용한 네트워크 대역폭이 불필요한 재전송과 대기시간의 증가로 많이 소모되게 될 것이다. 하지만 그렇다고

해서 소용량 파일들과 함께 동일한 캐시 서버에서 관리한다면 한정된 공간을 대용량 파일들이 차지함으로써 그로 인해 소용량 파일들은 대부분 삭제되고 그것은 소용량 파일들을 캐싱 하는데 대해 미스(miss)가 발생할 확률과 응답시간이 늘어나는 등의 또 다른 문제를 발생시킨다. 따라서 소용량 콘텐츠와 대용량 콘텐츠를 서로 다른 캐시 서버를 두어 관리하고 각각의 독립된 캐시 서버에서 서비스 해야 한다.

4. 구현 전략

기존의 콘텐츠 다운로드 방식은 (그림 2)와 같다.



(그림 2) 기존의 콘텐츠 다운로드 방식

먼저 이용자가 URL(Uniform Resource Locator)을 입력하면 DNS(Domain Name System)로부터 CP(Content Provider) 서버의 IP(Internet Protocol) 주소를 지정받고 다시 해당 CP 서버에 콘텐츠를 요청하여 CP 서버로부터 콘텐츠를 제공받는다.

이같은 콘텐츠 다운로드 방식을 개선한 전략

적 CDN은 그 구축위치에 따라 크게 세 가지로 구분할 수 있다. 첫째는, IX(Internet eXchange) 형태로 ISP(Internet Service Provider)의 백본(backbone) 또는 IX 단에 서버를 구축하고 BGP(Border Gateway Protocol) 연동 등의 방식으로 네트워크를 구성하는 것이다.

둘째는, IDC(Internet Data Center)형태로 각 IDC 에 서버를 구축하고 특정 지역을 대상으로 네트워크를 구성한다.

셋째는, 에지(edge)형태로 각 ISP의 POP(Point Of Presence) 단에 다수의 서버를 구축하는 것이다.

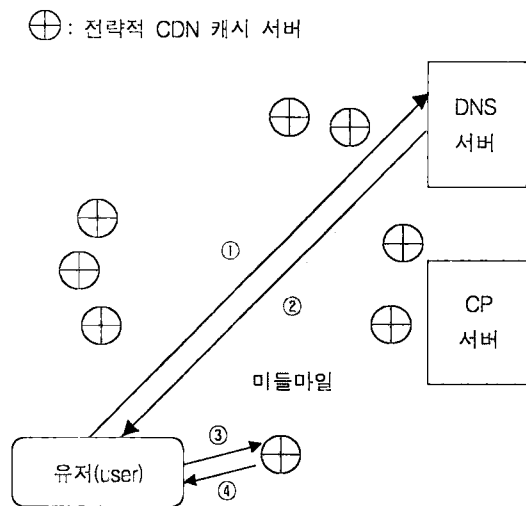
본 연구에서는 현재의 기술 수준과 정황 등을 고려하여 실현 가능한 두 번째 방법을 이용한 전략을 제시한다.

전략적 CDN 은 CP(Content Provider)의 서버에 보관된 콘텐츠를 전국 곳곳의 캐시서버로 복제함으로써 시작된다. 이때 CP 서버에서 콘텐츠가 변경되면 미리 CP 서버에 설치된 에이전트가 이를 감지하고 CDN 서버에 이 사실을 통보하여 변경된 콘텐츠를 인터넷의 각 지역에 설치된 캐시서버로 전송 및 갱신하고 모든 지역의 캐시서버를 동기화 한다.

그 후 이용자의 콘텐츠 요청이 있을 때 CP 본사의 서버가 아닌 해당 지역의 캐시서버로부터 콘텐츠를 전송하는데, 이 때 CP는 모든 콘텐츠 또는 일부 콘텐츠만을 전송하도록 선택할 수 있다. html 같은 텍스트는 트래픽에서 차지하는 비중이 미미하므로 전략적 CDN 대상에서 제외해도 무방하다. 즉, 일부 콘텐츠만을 전송할 경우 파일크기가 큰 콘텐츠는 가까운 지역의 캐시 서버로부터 전송하고, 텍스트 또는 작은 그래픽 등 파일 크기가 작은 콘텐츠는 CP 본사의 서버로부터 직접 전송하는 것이다.

(그림 3)을 보면, 먼저 이용자가 URL(Uni-

form Resource Locator)을 입력하면 DNS로부터 이용자에 근접한 전략적 CDN 캐시 서버의 IP 주소를 지정받아 지정 서버에 콘텐츠를 요청하여 가까운 캐시 서버로부터 콘텐츠를 제공 받는다. 이 그림에서처럼 중간 부분의 미들마일 구간을 기치지 않고 주변에서 지역적으로 가장 가까운 전략적 CDN 캐시 서버로부터 콘텐츠를 전송받기 때문에 빠른 전송 속도와 안정성 등을 보장 받을 수 있다.



(그림 3) 전략적 CDN을 이용한 콘텐츠 다운로드 방식

요약하면, ISP 각각에 전략적 CDN 캐시 서버를 설치하고 CP 들은 자신들의 인기있는 콘텐츠를 CDN 캐시서버 앞단에 존재하는 CDN 서버를 통하여 미리 각 캐시 서버에 가져다 놓음으로서, 유저 입장에서 특정한 콘텐츠를 서비스 받고자 할 때는 근처 ISP의 캐시 서버로부터 콘텐츠를 서비스 받으므로 Origin CP 웹 서버에 직접 접속하지 않고, 또한 그렇기 때문에 미들마일을 통과하지 않음으로서 네트워크 체증과 대기시간을 줄일 수 있다.

전략적 CDN의 구성은, 실제로 콘텐츠를 저

장하고 있으면서 인터넷 이용자의 요구를 받아들이는 캐시(cache) 서버와 미디어(media) 캐시 서버, FTP 캐시 서버 그리고 다수의 캐시 서버의 로드 밸런싱(load balancing)을 위한 네트워크 장비, 인터넷 이용자에게 가장 적합한 인터넷 서버로 인도하는 글로벌 서버 로드 밸런싱(Global Server Load Balancing)을 위한 네트워크 장비 등으로 이루어져 있다.

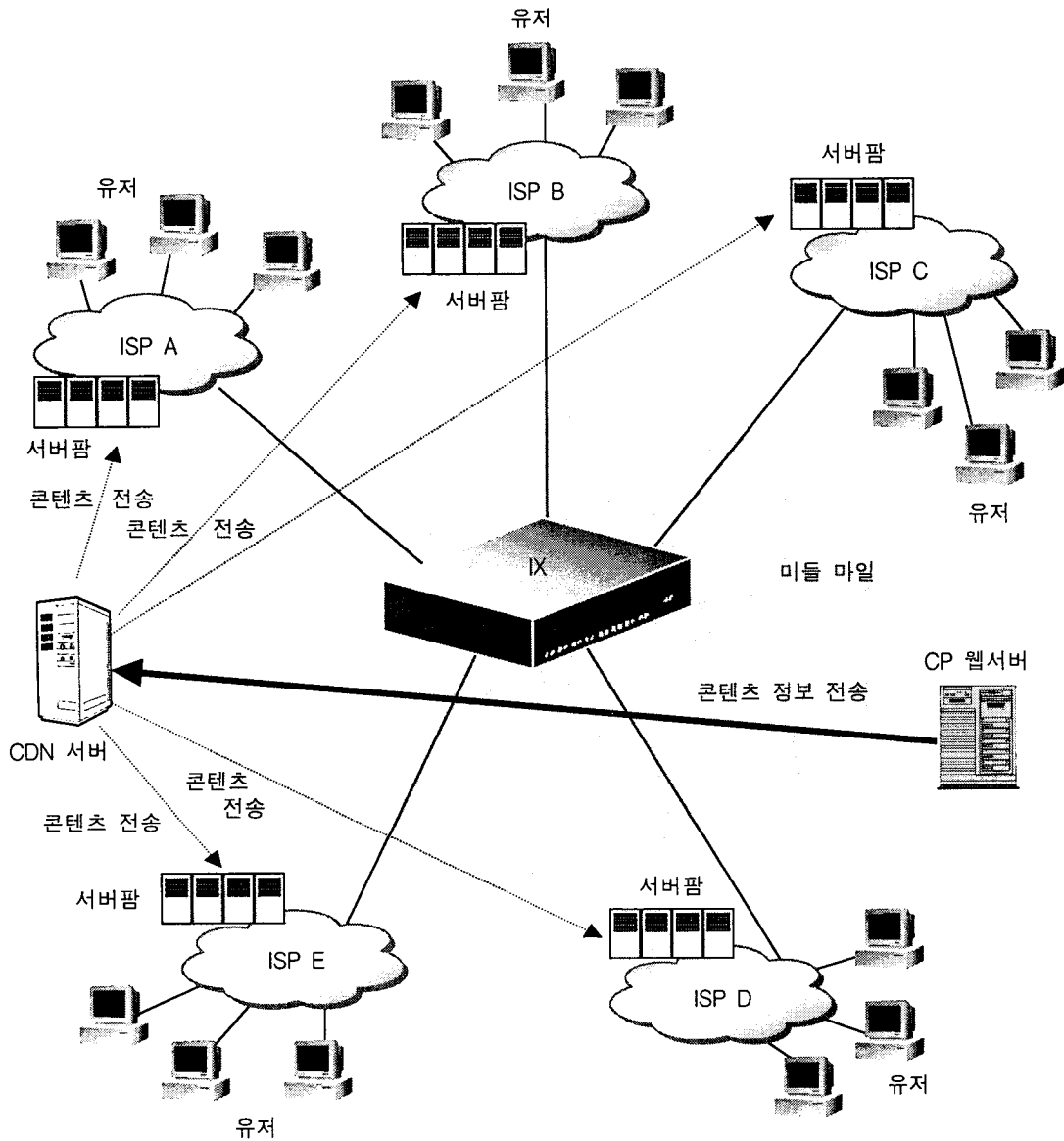
각각의 ISP에는 용도에 맞는 캐시 서버, 미디어 캐시 서버, FTP 캐시 서버 등이 하나의 서버군으로 구성되어 있으며 이러한 서버군을 이 연구에서는 서버팜(server farm)이라 칭한다.

(그림 4)를 보면 전략적 CDN 구성도가 잘 표현되어 있다. 각각의 ISP 에는 그 ISP에 가입한 유저(User)들이 존재한다. 그리고 각 ISP들은 IX(Internet eXchange)를 통해 서로의 콘텐츠를 전송 받는다.

이에 전략적 CDN은 품질을 보장 받을 수 없는 미들마일(middle-mile) 구간을 우회하여 콘텐츠를 주고 받음으로서 전송 속도 및 안정성을 추구함을 목적으로 한다.

전략적 CDN을 위해서는 먼저 각 ISP 에 서버팜(server farm)을 설치한다. 위에서 언급한 대로 서버팜은 캐시(cache) 서버와 미디어(media) 캐시 서버, FTP 캐시 서버들로 구성되어 있으며 하나의 서버팜 내의 트래픽 분배는 LB(Load Balancing) 기술이 담당하며 각 ISP에 설치되어 있는 서버팜 끼리의 트래픽 분배는 GSLB(Global Server Load Balancing) 기술이 담당한다.

각 ISP에 위치하고 있는 서버팜을 제어하고 관리하기 위해서 CDN 서버가 존재하는데, CP의 입장에서 자신이 접속한 ISP 외에 다른 ISP에 접속한 유저(user)에게 고품질의 콘텐츠를 제공하기 위해서 자신의 새로운 혹은 변경(수정)된 콘텐츠 정보를 CDN 서버로 전송한다. 그



(그림 4) 전략적 CDN 구성도

러면 CDN 서버는 그 정보를 각각의 서버팜에 전달 하여, 소용량 콘텐츠일 경우에는 전송된 경로를 기반으로 이미 캐시 서버에 존재하는 과거의(stale) 콘텐츠를들을 삭제토록 하여 이용자들이 CP의 웹서버에 콘텐츠를들을 요청 시 Origin 서버로부터 최신의 콘텐츠를를 제공받을 수 있게 하고(이 때 Origin 서버에서 최신의 콘텐츠를를 전송하면서 캐시 서버에 다시 새로운 콘텐츠가

갱신되어 저장됨으로써 두 번째 요청부터는 캐시 서버로부터 가져옴), 대용량 콘텐츠일 경우에는 각 서버팜의 미디어 캐시 서버 혹은 FTP 캐시 서버에 미리 콘텐츠를를 가져다 놓음으로써 각 서버팜에 존재하는 캐시 서버의 콘텐츠 동기화를 수행하게 된다.

따라서 어느 ISP 에 속한 유저라도 요청했던 콘텐츠를를 가까운 지역의 서버팜에 접속하여 제

공 받음으로서 Origin CP 서버와 동일한 내용의 콘텐츠를 신속하게 그리고 고품질로 서비스 받을 수 있다.

5. 구현 기술

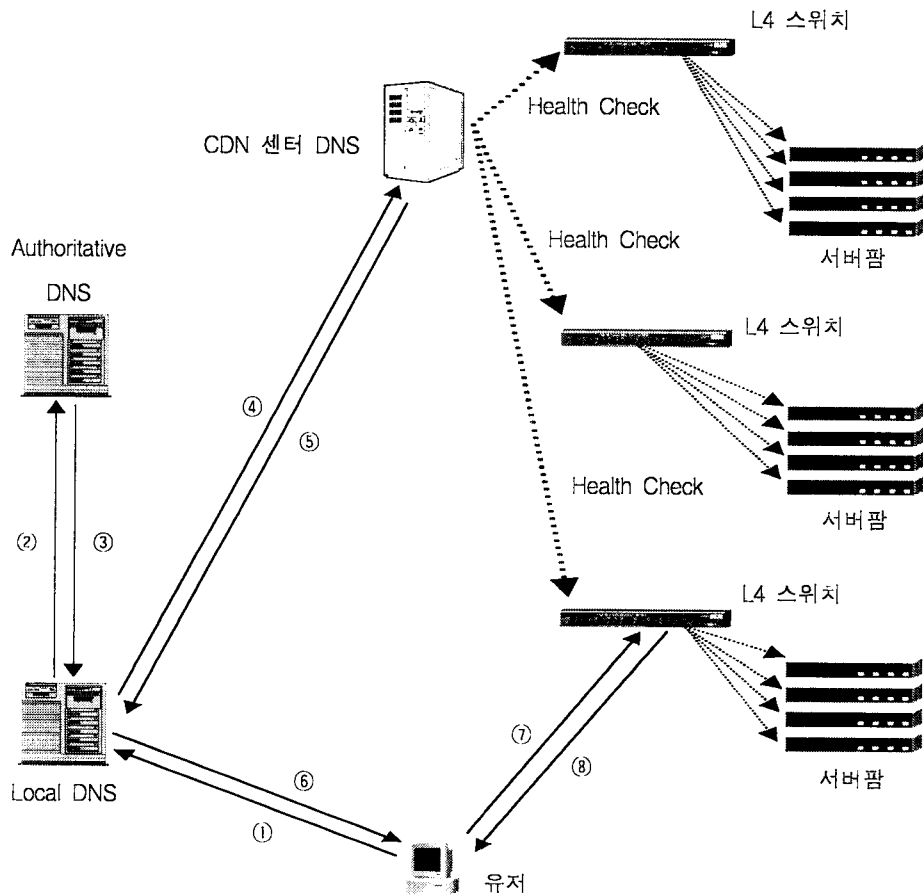
5.1 GSLB(Global Server Load Balancing)

GSLB(Global Server Load Balancing)는 CDN 서비스를 제공하기 위해 필요한 가장 중추적인 기술로서 인터넷 네트워크 여러 곳에 분산 배치되어 있는 많은 캐시서버 중에서 최종 사용자(end user)에게 최상의 서비스를 제공할 수 있는 캐시서버를 선정하여 서비스를 연결하는 기술이

다. 또한 GSLB는 최상의 위치에 있는 캐시서버에 장애가 발생했을 경우, 다음으로 상위 서비스를 할 수 있는 캐시서버로 우회 연결하여 장애를 해소한다.

이러한 GSLB 기술은 LDNS(Local DNS)를 이용하거나 HTTP 리다이렉션(redirection)을 이용하는 방법으로 구현할 수 있다.

(그림 5)에서는 ① URL http://www.hufs.ac.kr에 해당하는 서버를 알아내기 위해 LDNS에 www.hufs.ac.kr의 IP 주소를 요청한다. ② LDNS에 www.hufs.ac.kr에 대한 정보가 없으면, www.hufs.ac.kr의 정보를 관장하고 있는 hufs.ac.kr의 Authoritative DNS에 IP 주소를 요청한다.



(그림 5) GSLB(Global Server Load Balancing) 구성도

③ Authoritative DNS는 글로벌 서버 로드 밸런싱(global server load balancing)을 위해 www.hufs.ac.kr에 대한 정보 권한을 글로벌 서버 로드 밸런싱을 위한 DNS로 위임되어 있다는 사실을 LDNS(Local DNS)에 통보한다. ④ LDNS는 실제 www.hufs.ac.kr의 정보를 가지고 있는 CDN 센터의 글로벌 서버 로드 밸런싱을 위한 DNS로 IP 주소를 요청한다. ⑤ CDN 센터의 글로벌 서버 로드 밸런싱을 위한 DNS는 주기적으로 갱신되는 각 서버팜의 상태 정보와 DNS 쿼리를 요청한 LDNS의 IP를 분석하여 적합한 서버팜의 IP 주소를 결정하고, 글로벌 로드 밸런싱을 위한 DNS에서 LDNS로 결정된 IP 주소를 통보한다. ⑥ LDNS는 파악된 IP 주소를 요청한 유저에게 통보한다. ⑦ 유저의 브라우저는 www.hufs.ac.kr에 접속하기 위해 LDNS로부터 통보 받은 IP 주소의 서버로 www.hufs.ac.kr에 대한 콘텐츠를 요청한다. ⑧ 유저로부터 요청을 받은 CDN 서버가 유저에게 자신이 가지고 있는 www.hufs.ac.kr의 콘텐츠를 서비스 한다.

5.2 LB(Load Balancing)

LB(Load Balancing)는 요청되어지는 많은 작업들을 하나의 서버팜 내의 여러 대의 서버에 분배하는 기술로서 서버의 효율성을 최대화 할 뿐만 아니라 어플리케이션의 높은 가용성을 제공하는 솔루션이다. LB는 패킷의 Layer 3와 Layer 4의 헤더 정보, 즉 UDP나 TCP 포트 번호, TCP 세션의 끝과 시작을 나타내는 SYN/FIN 비트, 그리고 IP 소스/목적지 등과 같은 정보를 이용해서 세션을 인식하고 관리한다. 이러한 기술은 주로 Layer 4 단계를 지원하는 스위치(보통 L4 스위치)를 이용하여 구현된다.

서버팜 내의 서버는 서버간의 로드 밸런싱

(load balancing)을 통하여 효율성을 극대화 한다. 즉, 로드 밸런싱은 전국 또는 전 세계에 분산돼 있는 다수의 서버를 감시하고, 이용자의 요구에 가장 빠르게 응답할 수 있는 서버를 선택하게 하는 기술이다.

5.3 콘텐츠 동기화

전략적 CDN을 구축하는데 있어서 다음과 같이 여러 유형의 캐시 전략을 생각해 볼 수 있다. 첫째, 요청이 있을 때마다 Origin 웹 서버에 접속하는 것이다. 콘텐츠에 대한 요청이 있을 때마다 캐시 서버에 해당 콘텐츠의 존재유무 상관없이 웹 서버에 접속한다. 둘째, 시간이 지난 콘텐츠만을 대상으로 Origin 웹 서버에 접속하는 것이다. 유효 기간을 정해두고, 그 유효 기간이 지난 콘텐츠에 대한 요청에만 웹 서버에 접속한다. 셋째, 동일한 이름의 콘텐츠는 무조건 캐시서버에서 서비스 한다. 콘텐츠에 대한 요청시 동일한 이름의 콘텐츠가 캐시 서버에 존재하면 무조건 캐시 서버에서 서비스 하는 것이다.

첫 번째와 두 번째의 경우는 요청 시 마다 웹 서버에 직접 접속하여 가져오기 때문에 서비스 하는 콘텐츠가 늘 새로운 최신의 내용이라는 장점은 있으나 웹 서버 부하 증가와 성능 저하를 가져오게 된다. 따라서 이 경우라면 전략적 CDN의 존재 의미는 없다.

종래의 캐시 서버는 인터넷 이용자의 콘텐츠 전송 요청이 발생한 시점에 요청한 콘텐츠가 캐시 서버에 있으면 캐시 서버 제작자의 자체 알고리즘에 의해 콘텐츠 freshness(CP 웹 서버의 최종 콘텐츠가 캐시 서버의 콘텐츠와 동일한지의 여부)를 파악한 후 콘텐츠를 전송하는 수동적인 방식을 따른다. 따라서 인터넷 이용자에게

인지되는 시간과 최종 콘텐츠 인도까지 걸리는 시간이 늘어날 수 밖에 없다.

위 캐시 정책 중에서 콘텐츠 동기화는 세 번째 전략인 경우에 필요하다. 신속한 전송을 위해서는 콘텐츠 요청시 캐시 서버에 이미 저장되어 있는 기존의 콘텐츠를 신속히 제공해야 한다. 그렇지만 요청 받았던 콘텐츠가 캐시 서버에 존재한다고 해서 무작정 콘텐츠를 서비스한다면 이미 유효기간이 만료한(stale) 과거의 콘텐츠가 서비스 될 수도 있다. 따라서 최신의 콘텐츠를 전송받기 위해서는 웹 서버와 캐시 서버간의 콘텐츠를 조사하여 내용이 다른 경우 즉, 과거 내용(stale)의 콘텐츠는 요청이 오기 전에 미리 캐시 서버로부터 삭제해야 한다. 그럼으로써 실제 요청이 왔을 경우 캐시 서버에는 미스(miss)가 발생하면서 실제 Origin 웹 서버로 접속하여 새로운 콘텐츠를 제공받을 수 있다.

5.4 저장소(storage)

캐시 서버와 달리 미디어 파일, FTP 파일은 용량 규모가 상당히 크므로 가능한 한 Origin CP 웹 서버에 접근하지 않기 위해서, 인기 있는 콘텐츠를 미리 미디어 캐시 서버 혹은 FTP 캐시 서버에 가져다 놓아야 한다.

인기있는 콘텐츠를 판별하는 결정은 서비스를 제공하는 CP 측에서 실행해야 하며, 그러한 대용량의 콘텐츠를 미리 가져다 놓기 위해선 역시 그 파일들을 저장할 수 있는 대용량의 저장소(storage)가 존재해야 한다. 그리고 효율적인 저장소의 관리를 위해서는 미디어 캐시 서버, FTP 캐시 서버들이 각각의 저장소를 보유해야 하며, 그것이 가능하기 위해서는 SAN(Storage Area Network)과 같은 저장소 기술이 필요하다.

6. 성능 평가

캐시를 운영하는 데에는 크게 두 가지 목적이 있다. 하나는 비용 절감이고, 다른 하나는 성능 향상이다. 어느 쪽을 중점으로 두는지는 회선 사용에 대한 과금 제도나 회선의 대역폭, 운영자의 의도와 같은 주위 환경에 따라 많이 좌우된다[2].

일반적으로 기존의 캐시 성능은 그 명중률(hit ratio)로 말하는데, 이러한 맥락에서 이상적인 캐시는 자료를 저장할 수 있는 공간이 무한대인 것을 말하며 그 이유는 이 때의 명중률이 가장 높기 때문이다. 그러나 무한대의 자료 저장 공간은 현실적으로 불가능하고 실제 캐시들의 저장 공간은 한정되어 있을 수 밖에 없기 때문에, 이 한정된 저장 공간을 최대한 활용해서 명중률을 높이는 것이 캐시의 성능을 향상시키는 데에 있어서 중요하다. 캐시의 성능을 평가하는 기준은 명중률 이외에도 정보 요구를 처리하는데 걸리는 시간, 원본 자료와의 일관성이 있다. 그러나 이런 평가 대상들이 어느 하나를 최적화 한다고 해서 나머지 다른 것들도 같이 최적화 되는 것은 아니다.

본 연구에서는 현재까지 존재하는 캐시 자체의 성능평가가 아닌, 전략적 CDN 을 구축하고 그 전략의 네트워크 캐시 성능을 평가하기 위해서 실제 두 IDC(Internet Data Center)의 세 지역에 서버팜을 두고 그 각각에 미디어 캐시 서버, FTP 캐시 서버를 설치한 후 객관적인 속도 측정을 통해서 그 성능을 평가한다.

측정 대상 노드로서 ‘ㄱ’ 지역에 위치한 “A” IDC와 “B” IDC 그리고 “B” IDC의 ‘ㄴ’ 지역에 각각 서버팜을 구축하고 각 서버팜에 별도의 미디어 캐시 서버와 FTP 캐시 서버를 설치한 후, 평가 대상은 스트리밍 콘텐츠와 FTP 콘텐츠를 구분하여 성능을 측정하였다.

6.1 평가 환경 및 방법

<표 2>는 성능평가에 있어 설치된 하드웨어 사양과 프로그램, 설치 장소이다.

<표 2> 측정 대상 노드 및 하드웨어 사양

설치 장소	하드웨어(서버) 사양			설치 프로그램
	OS	CPU	Memory	
A IDC ('-' 지역)	Window 2000	Pentium 850*2	1 G	FTP, 미디어 서버
B IDC ('-' 지역)	Window 2000	Pentium 850*2	1 G	FTP, 미디어 서버
B IDC ('-' 지역)	Window 2000	Pentium 850*2	1 G	FTP, 미디어 서버

<표 3> 유저 접속 환경

설치 장소	측정 항목	PC 사양		미디어 플레이어 버전
		CPU	Memory	
'-' 지역 a	FTP	Pentium3 500	128 M	X
'-' 지역 a'	스트리밍	Pentium3 700	128 M	7.0
'-' 지역 b	FTP	Pentium3 750	128 M	X
'-' 지역 b'	스트리밍	Pentium3 500	128 M	6.4

<표 3>은 실제 전략적 CDN 네트워크에 접속하여 측정에 참여하였던 유저의 PC 사양과 프로그램 버전이고, <표 4>는 평가 방법이다.

<표 5>와 <표 6>은 스트리밍 콘텐츠 측정 목록과 FTP 콘텐츠 측정 목록을 나타낸다.

스트리밍 콘텐츠의 경우 CATV 가입자 환경의 미디어 플레이어 버전(7.0 버전) 문제로 측정 오류가 발생함으로써 스트리밍 1대, FTP 2대로서 평가가 이루어졌다.

<표 4> 평가 방법

	스트리밍 콘텐츠	FTP 콘텐츠
평가 방법	12분씩 3개 노드를 순차적으로 측정 (1 시간 마다)	3개 노드에서 순차적으로 계속 반복 측정

<표 5> 스트리밍 콘텐츠 측정 목록

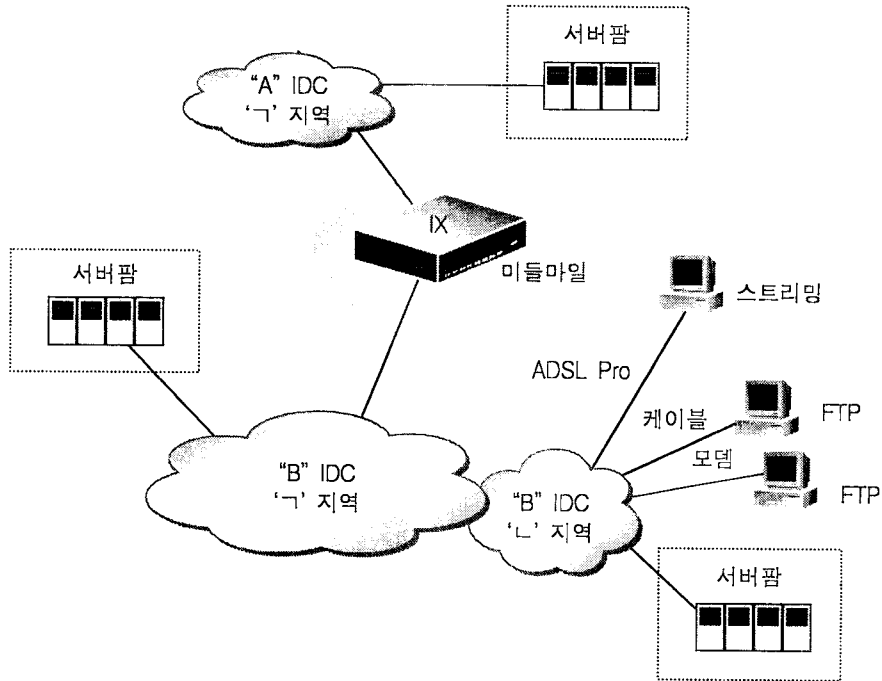
용량 및 설치 장소	스트리밍 콘텐츠
용량	500k 스트리밍, 12분
A IDC ('-' 지역)	mms://211.196.150.247/cdn_500.asf
B IDC ('-' 지역)	mms://211.202.0.44/cdn_500.asf
B IDC ('-' 지역)	mms://211.177.192.11/cdn_500.asf

<표 6> FTP 콘텐츠 측정 목록

용량 및 설치장소	FTP 콘텐츠
용량	48.6 Mbyte
A IDC ('-' 지역)	ftp://211.196.150.245/pub/cdn/cdn_client_2012.exe.
B IDC ('-' 지역)	ftp://211.202.0.43/pub/cdn/cdn_client_2012.exe
B IDC ('-' 지역)	ftp://211.177.192.11/pub/cdn/cdn_client_2012.exe

<표 7> 평가 기간 및 항목

	스트리밍 콘텐츠	FTP 콘텐츠
평가 기간	2001.9.7 15 : 00 ~ 2001.9.9 24 : 00	2001.9.7 00 : 00 ~ 2001.9.9 24 : 00
평가 항목	<ul style="list-style-type: none"> 전송 편차 - 하나의 스트리밍 콘텐츠를 반복 측정하여 대역폭에 대한 평균의 표준편차를 구함. 	<ul style="list-style-type: none"> 다운로드 시간 - 측정용 파일을 완전히 다운로드 받는데 소요되는 시간임. 다운로드 속도 - 측정용 파일의 파일 사이즈를 다운로드 시간으로 나눔.



(그림 6) 측정 구성도

<표 7>은 평가 기간 및 항목을 나타낸다.

6.2 측정 구성도

평가를 위한 측정 구성도는 (그림 6)과 같다. 그림에서처럼 평가에 참여한 유저들은 “B” IDC의 ‘L’ 지역 센터에 접속되어 있다.

6.3 스트리밍 콘텐츠 측정 결과

스트리밍 콘텐츠에 대한 평가는 12분짜리 콘텐츠에 대해 30초 간격으로 이루어졌다. 즉, 12분 분량의 스트리밍 콘텐츠 파일을 한 시간 마다 각각 30초의 구간으로(총 24구간) 나누어 측정하였다. 스트리밍 콘텐츠 품질 측정 결과 그래프의 횡축이 24구간인 것은 12분을 30 초씩 나눈 까닭이다(30 초 * 24 = 12분).

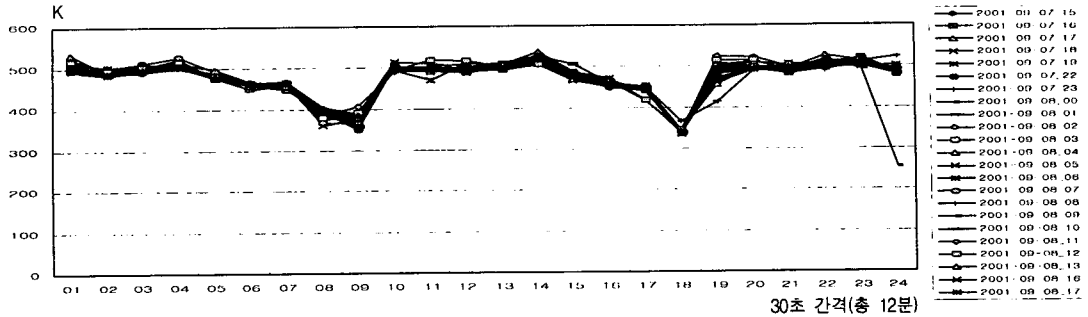
지금까지의 평가 결과를 분석해 보면, (그림

7)의 스트리밍 콘텐츠의 경우, ‘L’ 지역 가입자망(유저)과 “B” IDC ‘L’ 지역 센터는 매우 안정적인 전송 품질을 유지하였다. 하지만 (그림 8)의 경우 ‘L’ 지역 가입자망(유저)과 “B” IDC는 전송 품질이 비교적 양호하나 피크 타임 동안에는 전송 지연이 일부 발생하였다. 즉 트래픽이 집중되는 21시~1시 사이에 일시적으로 콘텐츠 전송 지연이 발생함을 알 수 있다.

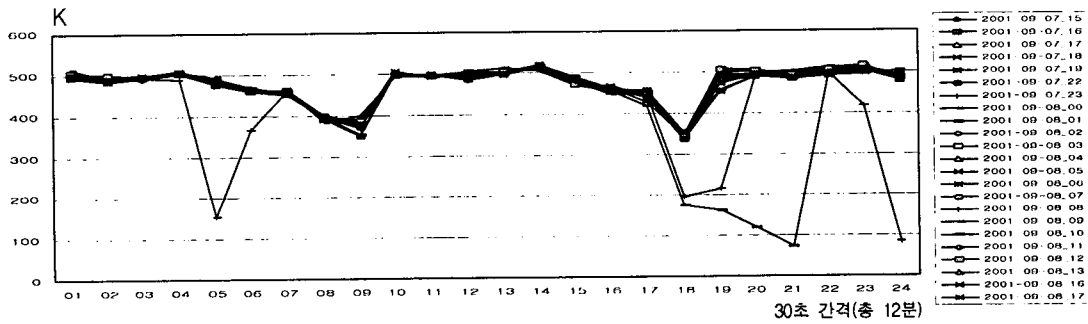
(그림 9)를 보면 ‘L’ 지역 가입자망(유저)과 “A” IDC의 전송 품질은 그 편차가 상당히 크다. 그래서 ‘L’ 지역의 “B” IDC 가입자의 경우에는, “A” IDC 내에 있는 스트리밍 콘텐츠를 전송받을 경우 안정적인 품질보장이 매우 곤란한 수준임을 알 수 있다.

6.4 FTP 콘텐츠 측정 결과

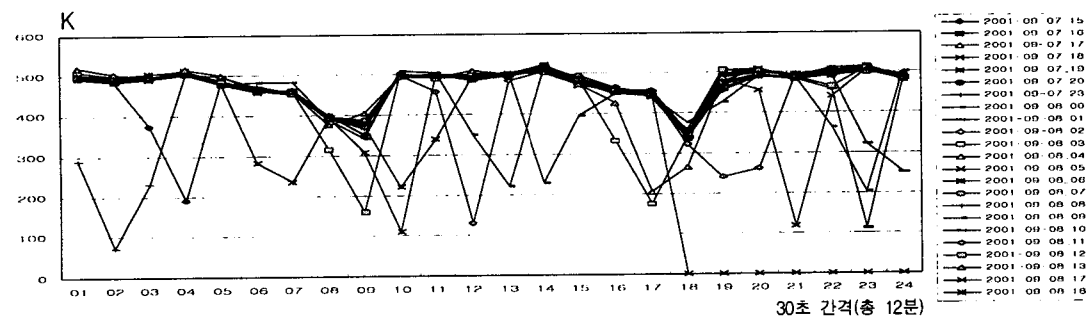
(그림 10)은 3일 동안 ADSL, CATV 다운로드



[그림 7] "B" IDC 'L' 지역의 스트리밍 콘텐츠 품질측정 결과



(그림 8) "B" IDC 'G' 지역의 스트리밍 콘텐츠 품질측정 결과

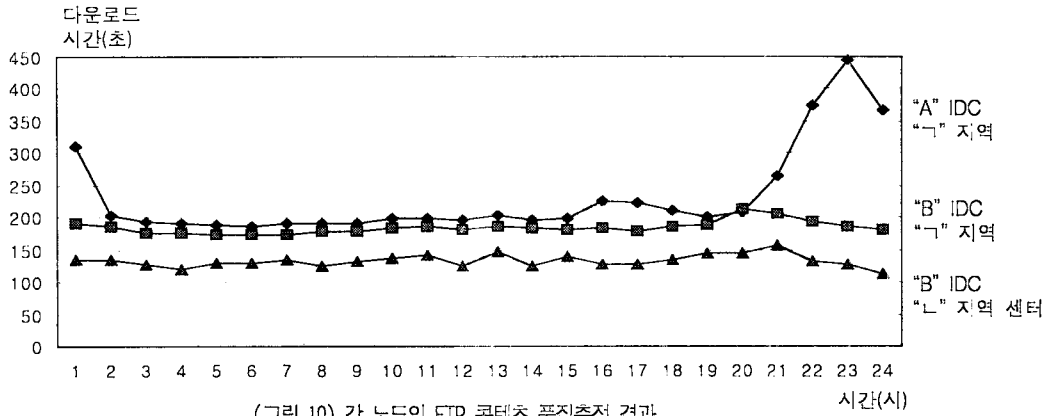


(그림 9) "A" IDC 'G' 지역의 스트리밍 콘텐츠 품질측정 결과

드 시간의 평균값으로서, FTP 콘텐츠의 다운로드 시간은 평균값으로 "B" IDC 'L' 지역 센터는 133초, 그리고 "B" IDC 'G' 지역은 184 초, 그리고 "A" IDC 'G' 지역은 231초로 나타났다. 즉, "B" IDC 'L' 지역 센터에 위치한 서버팜의 경우 다운로드 시간의 평균값은 "B" IDC 'G' 지역 보다 28%, 그리고 "A" IDC 'G' 지역 보다 43% 가 더 빠른 것으로 나타났다. 그래서 다운로드 시간과 속도는 "B" IDC 'L' 지역 센터 >

"B" IDC 'G' 지역 > "A" IDC 'G' 지역 순서였다.

한편, 피크 시간대인 21시~1시 사이에는 "B" IDC 'L' 지역 센터와 "A" IDC 간에 최고 290 초 정도의 다운로드 시간차가 발생했다. 즉, "B" IDC 'L' 지역 센터와 "B" IDC 'G' 지역의 서버팜은 안정적이고 일정한 다운로드 속도 대비를 보였으나 "A" IDC 'G' 지역의 서버팜의 경우 피크 시간대에 매우 불안정한 다운로드 속도



와 품질(최고 259초 편차)을 보였다.

6.5 평가분석 및 기대효과

이상의 분석 결과와 전략적 CDN 구현의 기대 효과를 다음의 <표 8>로 정리하였다.

7. 결 론

7.1 연구결과와 요약과 의미

미들마일(middle-mile)의 문제점으로 인터넷은 기본적으로 품질을 보장할 수 없다는 한계를 가지고 있다. 따라서 인터넷 네트워크의 QoS

(Quality of Service)가 보장되지 않는 한 디지털 콘텐츠 기반의 e-비즈니스는 활성화 될 수 없다.

영화, 그래픽, 공개자료파일 등의 콘텐츠를 제공하는 CP(Contents Provider)의 웹 서버는 일반적으로 한 개의 특정 ISP에 연결되어 있는데 반해서 콘텐츠를 요구하는 이용자는 다양한 ISP 네트워크에 존재하기 때문에 ISP간에 이동되는 트래픽은 CP와 이용자가 늘어남에 따라 기하 급수적으로 증가하게 된다.

이를 물리적인 망 증설만으로 해결하기는 현실적으로 어려우므로 주요 ISP 네트워크에 캐시 서버를 설치하여 CP의 콘텐츠를 각 ISP 네트워크

<표 8> 평가결과 분석 및 기대효과

평가 결과 분석	전략적 CDN 의 기대 효과
<ul style="list-style-type: none"> • 미들마일(middle-mile) 구간을 통과하는 "B" IDC 'L' 지역 가입자 망과 "A" IDC 'L' 지역과의 콘텐츠 전송 품질은 매우 불안정적임. • "B" IDC 'L' 지역센터와 "B" IDC 'L' 지역간은 전송 품질은 양호하나 트래픽 피크시에 스트리밍 콘텐츠의 품질 저하 현상이 부분적으로 발생함. • 일반 가정 또는 기업에 이르는 가입자망 구간은 xDSL, 케이블 모뎀 등으로 투자가 활발히 이루어지는 편임. • 기간망과 기간망 또는 기간망과 가입자망 사이를 연결하는 피어링(peering) 구간에 대해서는 상대적으로 투자가 이루어지지 않음. 	<ul style="list-style-type: none"> • 미들마일 구간을 우회함 전략적 CDN은 네트워크 체증과 불안정성을 유발하는 미들마일 구간을 우회함으로써 안정적 전송을 보장함. • 피크 시간대의 급격한 트래픽에 대한 품질 보호 - 평상시 내부 트래픽은 품질상에 문제가 없으나 이용자의 폭주로 인해 갑자기 증가하는 트래픽 피크시에 일부 발생할 수 있는 품질 저하 현상을 방지함. • 전송 품질 안정화 - 콘텐츠를 서비스하는 CP 등의 입장에서 볼 때 가입자 및 대용량 트래픽 증가에도 전송 품질이 안정화 됨. • 비용 절감 - 범 국가적 자원에서 전략적 CDN 구축하면 기타 인터넷 이용자들은 별도의 서버 추가나 네트워크 증설 등의 비용을 절감할 수 있음.

〈표 9〉 전략적 CDN의 장점

전략적 CDN 의 장점	세 부 설 명
빠른 속도	병목 현상과 잦은 데이터 손실이 발생하는 IX(Internet eXchange) 및 미들마일(middle-mile) 구간을 우회하여 이용자로부터 가장 가까운 지점에서 콘텐츠를 전송하여 빠른 속도를 보장함.
안 정 성	콘텐츠를 분산처리 함으로서 이용자 수에 상관없이 동일한 속도와 품질을 보장함.
성능 향상	트래픽 폭주로 인한 서버의 성능저하를 방지하여 안정적으로 콘텐츠를 서비스 받을 수 있음.
비용 절감	전략적 CDN 캐시 서버를 공유 함으로서, 서버/네트워크 투자비 및 운영 비용을 절감할 수 있음.
확 장 성	다수의 캐시 서버를 클러스터링 방식으로 연결 함으로서 시스템의 확장성이 뛰어나.
Origin 서버의 부하를 경감	Origin 서버에서는 전략적 CDN 캐시 서버로 콘텐츠를 전송하고 그 뒤로는 각각으로 분산되어 있는 캐시 서버로부터 콘텐츠를 서비스 함으로써 Origin 서버의 부하를 크게 줄일 수 있음.

마다 분산하여 저장하고 이용자의 요청 시 이용자의 인터넷 회선이 물려 있는 ISP 네트워크에서 바로 콘텐츠를 전송하여 데이터 손실과 속도 저하 등의 문제를 해결하는 것이 전략적 CDN 이다.

전략적 CDN은 인터넷 미들마일을 우회하여 콘텐츠를 전송함으로써 라우팅 홉(routing hop)이나 IX(Internet eXchange)에서 발생하는 패킷 손실(packet loss)을 차단하여 콘텐츠를 빠르고 안정적으로 전송함과 동시에 네트워크/서버의 투자비 및 운영비용을 절감하여 e-비즈니스의 활성화에 기여할 수 있다(〈표 9〉 참조).

Origin 서버에서는 전략적 CDN 캐시 서버로 콘텐츠를 전송하고 그 뒤로는 각각으로 분산되어 있는 캐시 서버로부터 콘텐츠를 서비스 함으로써 Origin 서버의 부하를 크게 줄일 수 있음.

7.2 연구의 한계와 향후 발전방향

전략적 CDN 은 인터넷 방송, 언론, 검색, 쇼핑몰 사이트 등과 같이 멀티미디어 콘텐츠에 적

합하다. 특히 동영상, 그래픽, 오디오 파일과 같은 파일 크기가 크거나 인기 있는 주문형 비디오(VOD) 등 접속빈도가 높은 콘텐츠 전송에서 큰 효과를 볼 수 있다.

한편, 실시간 변경 콘텐츠에 대해선 전략적 CDN의 적용이 어렵다. 다이나믹 콘텐츠 등은 실시간으로 내용이 바뀌어야 하기 때문에 캐시 할 수 있는 성격이 아니기 때문이다. 즉, 실시간 게임이나 채팅과 같은 콘텐츠들에 대해서는 전략적 CDN을 적용키 어렵다는 한계점을 갖는다.

전략적 CDN의 성공적인 적용을 위해서는 최종 이용자 근처에서 서비스를 대행하게 될 콘텐츠 캐싱을 위한 서비스 플랫폼의 구성과, 이용자 요청에 대해 최적의 경로를 찾아서 제공해줄 수 있는 기술(HTTP, IP, DNS Redirection)을 어떻게 구현할 것인가에 달려 있다.

또한 단순 HTML 문서와 이에 포함되어 있는 임베디드(embedded) 문서들(이미지, 스크립트, 바이너리 파일 등)과 어플리케이션 및 멀티미디어 데이터 스트리밍 등 다양한 콘텐츠를 지원할 수 있어야 하고, 새로운 형태의 콘텐츠들

이 지속적으로 개발되고 있기 때문에 전략적 CDN 역시 이를 지원할 수 있는 기술을 보유해야 한다.

참 고 문 헌

- [1] 김주용, "프락시 캐시 일관성 유지 기법 및 성능 분석", 서울대학교 전산학과 박사학위논문, 2000.
- [2] 김현진, "네트워크 캐시에서의 갱신전략에 관한 연구", 한국과학기술원 전산학과 석사학위논문, 1997.
- [3] 박용운, "광대역 네트워크에서 연속형 미디어 오브젝트 서비스를 위한 캐싱 정책", 부산대학교 전자계산학과 박사학위논문, 2001.
- [4] 박진석, "인터넷 캐시 성능향상을 위한 파일 타입별 캐싱", 중앙대학교 컴퓨터공학과 석사학위논문, 1997.
- [5] 정재연, "중복 메커니즘을 이용하여 오브젝트 크기에 확장 가능한 웹 캐시 시스템", 한국과학기술원 전산학과 석사학위논문, 1997.
- [6] Chankhunthod, A., P.B. Danzig, C. Neerdaels, M.F. Schwartz and K.J. Worrell, "A Hierarchical Internet Object Cache," Proceedings of the 1996 USENIX Technical Conference, January, 1996.
- [7] Gray, M., *Internet Statistics-Growth and usage of the Web and the Internet*, January 1996. <http://www.mit.edu/people/mkgray/growth>
- [8] Luotonen, A., and K. Altis, "World Wide Web Proxies," Computer Networks and ISDN Systems, First International Conference on WWW, 1994.
- [9] NCRI, *The World-Wide-Wait Status Report*, October 1999
- [10] Network Wizard, *Internet domain survey*, 1997.
- [11] Rizzo, L., Vicisano, L., "Replacement Policies for A Proxy Cache," Networking, IEEE/ACM Transaction on Volume : 82, April 2000. pp. 158-170.
- [12] Smith, N. G., "World-Wide Web Caching," SuperJANET user Support Workshop, September 1995, revised at August 1996
- [13] Wang, J., "A Survey of Web Caching Schemes for the internet," Cornell Network Research Group(C/NRG) Department of Computer Science, TR99-1747, May 12, 1999.
- [14] Wessels, D., "Web Caching," O'REILLY, 2001
- [15] Wessels, D., "Can WWW caches help save the Internet?," SCDzine, Vol.17, No.2, summer. 1996.
- [16] Wessels, D., *Intelligent caching for world-wide web objects*, Master's thesis, Washington State University, 1995.
- [17] Williams S. et. Al., "Removal Policies in Network Caches for World Wide Web Documents," Proceedings of ACM Sigcomm96, August 1996.
- [18] Wooster, Roland Peter, "Optimizing Response Time Rather than Hit Rates of WWW Proxy caches," Master Thesis, Virginia Polytechnic Institute and State University, 1995.
- [19] Zhang, Zhi-Li, Du, D.H.C., Dongli S, Yuewei Wang, "A Network-Conscious Approach to End-to-End Video Delivery over Wide Area Networks Using Proxy Servers," INFOCOM '98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE Volume : 2, 1998.
- [20] 씨디네트웍스(<http://www.cdnetworks.co.kr>)
- [21] 한국인터넷정보센터(<http://www.nic.or.kr>)
- [22] 텃즈코리아(<http://www.terms.co.kr>)

■ 저자소개



이 주 현

미국 University of Southern Mississippi 에서 전자계산학을 졸업한 후 Virginia Polytechnic Institute 에서 산업공학 석사학위를, Illinois Institute of

Technology에서 박사학위를 취득하였다. Bell Laboratories에서 5년간 정보통신분야 연구원으로 활동하다가 귀국하여 LG정보통신과 LG소프트웨어에서 사업본부장과 연구소장을 역임한 바 있다. 현재 한국외대 경영학과, 경영정보대학원 교수로 재직중이며, 한국CIO포럼 대표간사, 한국경영정보학회 학술담당 부회장, 그리고 한국외대 기업경영연구소장직을 맡고 있다. 관심 연구분야는 소프트웨어 공학, 프로젝트 관리, e-비즈니스 등이다.



조 병 룡

현재 (주)씨디네트웍스 개발팀 과장이며, 2002년도 2월 한국외국어대학교 경영정보대학원 응용전산학과 소프트웨어공학 전공, 이학석사 졸업 예정이다.