

CDN의 핵심 구성 기술들과 경향

한국과학기술원 최승락
(주)아라기술 양철웅 · 이중식

1. 소개

초기 인터넷 연구의 대부분은 네트워크 인프라 즉 데이터 링크, 네트워크, 트랜스포트 계층에 집중되었다. 그러나 웹, 스트리밍 등의 새로운 대표적 응용 분야가 발생하고, 이에 대한 수요가 기하급수적으로 증가함에 따라 인터넷 인프라는 한계에 다다랐다. 인터넷 인프라의 확충은 기본적으로 고비용을 요구하기 때문에 연구자들은 애플리케이션 계층에서 해결을 시도하였다. 이로 인해 나온 기술이 콘텐츠 네트워킹 기술이다.

기존 연구들이 네트워크 대역폭 증가, 전송의 효율화에 중점을 두었다면, 콘텐츠 네트워킹 기술은 네트워크에 지능을 부여한다. 사용자와 보다 인접한 위치에 콘텐츠를 준비하고, 사용자에게 가장 인접한 위치의 콘텐츠로 안내함으로써 보다 고품질의 콘텐츠 서비스가 가능하도록 한다.

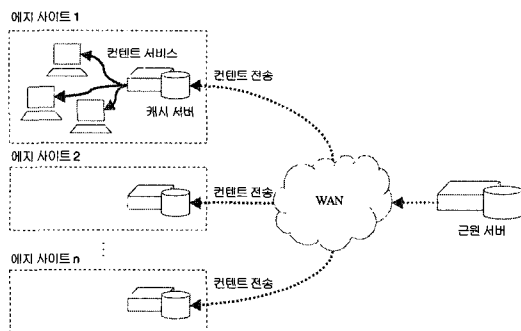


그림 1 CDN의 개념적 아키텍처

콘텐츠 네트워킹의 가장 대표적인 기술은 CDN (Content Delivery Network)이다. CDN은 콘텐츠를

사용자와 보다 인접한 위치로 이동시켜 네트워크 상에서의 콘텐츠 전송 비용을 줄인다(그림 1 참조). 뿐만 아니라, CDN은 지능적 전송과 체계적 관리를 부여함으로써 관리자가 한 지점에서 전체 콘텐츠 전송을 완벽히 제어할 수 있도록 한다. CDN을 통해 관리자는 사전에 콘텐츠를 에지(edge)로 이동시킬 수 있으며, 이동하는 시점, 방법, 상태를 관리 감독할 수 있다.

본 기사의 구성은 다음과 같다. 2장에서는 각 모듈에서 사용되는 기술들을 기술한다. 3장에서는 차기 CDN으로 대두되고 있는 eCDN과 P2P CDN에 대해 기술하고 4장에서 결론을 맺도록 하겠다.

2. CDN 구성 기술

CDN은 세 가지 핵심 모듈로 구성된다. 첫째, 클라이언트의 요청을 받아 콘텐츠를 서비스하는 캐시 서버이다. 웹 캐시[3], 스트리밍 캐시[4]들이 이에 해당한다. 둘째, 콘텐츠 전송 및 관리 모듈이다. 콘텐츠 전송 및 관리 모듈을 통해 관리자는 콘텐츠들을 여러 캐시 서버에 전송하고 시스템 동작 상태를 모니터링할 수 있다. 셋째, 요청 리다이렉터이다. 요청 리다이렉터는 클라이언트가 가장 인접한 서버로 접속할 수 있도록 한다. 요청 리다이렉터로 L4 스위치, GLB (Global Load Balancer)가 사용된다.

2.1 캐시 서버

캐시 서버는 클라이언트와 인접한 위치에 설치되어 클라이언트에게 콘텐츠를 서비스하는 모듈이다. 따라서 CDN의 클라이언트 요청 처리 능력 및 응답 시간은 캐시 서버의 성능에 좌우된다고 봐도 과언이 아니다. 캐시 서버는 근원 서버(origin server)로부터 콘텐츠를 전송 받아 근원 서버 대신 클라이언트에게

서비스한다.

웹 콘텐츠를 위해 사용되는 웹 캐시는 정적 콘텐츠 처리에 최적화되어 있어 웹 서버에 비해 우수한 성능을 보인다. 그러나 기존 웹 캐시는 동적 콘텐츠를 처리하지 못한다. 동적 콘텐츠를 캐싱하기 위한 제품이 나오고는 있으나 기존 시스템을 수정해야 하는 불편함이 있어 아직 보편화되어 있지 않은 실정이다.

멀티미디어 스트리밍 콘텐츠는 웹 콘텐츠보다 높은 네트워크 회선 대역폭을 요구하고 패킷 손실과 지연에 민감하기 때문에 보다 더욱 CDN의 지능형 네트워크를 필요로 한다. 동영상의 원활한 서비스를 위해서 한 대의 클라이언트에게 대략 200~700Kbps 정도의 대역폭은 보장이 되어야 한다. 클라이언트가 10대만 되도 2Mbps~7Mbps의 대역폭이 요구된다. 동시 접속 클라이언트 수가 증가할수록 네트워크 회선은 더더욱 한계에 다다른다.

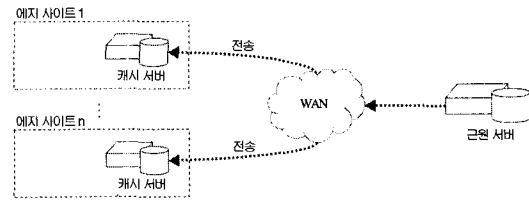
현재, 대부분의 캐시 업체들은 웹 캐시와 함께 스트리밍 캐시를 제공하고 있다. 웹 캐시와 통합되어 단일 장비에서 웹과 스트리밍 캐시가 동시에 가능한 제품이 있고 스트리밍 캐시 전용 장비로 판매되는 제품이 있다. 통합 장비는 관리적 측면에서 유리하다. 전용 장비는 성능이 우수하고 서비스 별 확장이 가능하다는 장점이 있다. 마이크로소프트사의 WMT(Windows Media Technology)[9], 리얼네트웍스사의 Helix[10], 애플사의 QuickTime[2]이 대표적인 스트리밍 기술들이고 많은 스트리밍 캐시 서버들이 이들 프로토콜을 지원하고 있다.

2.2 콘텐츠 전송

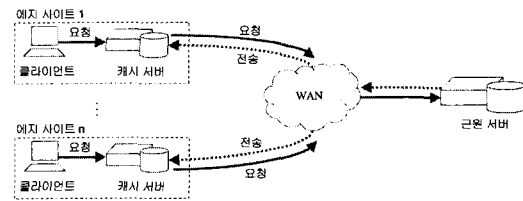
콘텐츠 전송 및 관리 모듈은 콘텐츠 전송과 시스템 설정 및 모니터링을 담당한다. 본 기사에서는 콘텐츠 전송 방법에 대해 살펴보고자 한다.

2.2.1 푸싱 vs. 캐싱

콘텐츠를 전송하는 방법에는 푸싱(pushing)과 캐싱(caching)이 있다(그림 2 참조). 푸싱은 관리자가 콘텐츠 전송을 유발시키는 경우이다. 관리자가 전송 대상과 전송 시점을 지정한다. 캐싱은 클라이언트에 의해 콘텐츠 전송이 발생하는 경우이다. 클라이언트가 캐시 서버에 특정 콘텐츠를 요청했을 때, 캐시 서버는 요청 중인 콘텐츠가 캐시에 저장되어 있는지 검색한다. 만약 요청 중인 콘텐츠가 없으면 캐시 서버는 근원 서버에게 콘텐츠 전송을 요청한다.



(a) 푸싱 전송 방법



(b) 캐싱 전송 방법

그림 2 콘텐츠 전송 방법

푸싱과 캐싱의 유용성은 콘텐츠 유형에 따라 달라진다. 웹 콘텐츠의 경우 기본적으로 캐싱 방법이 유용하다. 웹 콘텐츠는 개수가 매우 많기 때문에 일일이 관리자가 지정하여 전송 시키는 것은 매우 큰 불편함이 따른다. 따라서 관리자의 노력 없이 클라이언트의 요청에 의해 자동으로 콘텐츠를 전송 시키는 캐싱 방법이 유용하다. 캐싱 방법은 미리 콘텐츠를 캐시 서버에 전송 시켜 두지 않기 때문에 첫 번째 요청은 근원 서버에서 서비스 되고 따라서 서비스 품질이 낮아질 수도 있다. 그러나 웹 콘텐츠는 크기가 매우 작고 실시간 서비스를 요구하지 않기 때문에 다소간의 지연은 크게 문제가 되지 않는다.

물론 웹 콘텐츠의 크기가 매우 커서 캐싱 하는데 걸리는 시간이 길다면 푸싱을 통해 미리 캐시 서버에 전달 시키는 것이 적합하다. 미리 전달하지 않는다면 초기 많은 클라이언트의 요청이 근원 서버로 전달되고, 그 결과 근원 서버가 폭주하고 다운로드 시간이 오래 걸릴 수 있다. 일부 웹 캐시들은 오브젝트 스플리팅(object splitting) 기능을 통해 요청 중인 클라이언트가 아무리 많더라도 근원 서버와 하나의 연결만 생성하고, 그 연결에서 받은 패킷을 클라이언트에게 나누어 주는 방식을 사용한다. 이 방법을 사용한다면 미리 콘텐츠를 전송하지 않더라도 근원 서버가 폭주할 가능성은 없다.

푸싱은 캐시 일치성 유지를 위해 중요한 역할을 한다. 캐시 서버에 저장되어 있는 콘텐츠는 가급적

근원 서버에 있는 콘텐츠와 동일한 버전이어야 한다. 웹 캐시는 캐시 일치성 유지를 위해 TTL(Time To Live)을 사용한다[6]. 웹 캐시는 TTL 값에 지정된 기간이 만료되지 않는 한 근원 서버에게 콘텐츠가 변경되었는지 물어보지 않는다. 성능을 위해서는 가급적 TTL의 값을 크게 하여 웹 캐시가 근원 서버에게 빈번히 물어보지 않도록 해야 한다. 하지만 TTL의 값이 클수록 변경된 콘텐츠의 내용을 즉시 클라이언트에게 전달하기 어렵다. 이 경우 푸싱은 유용하다. 관리자는 변경된 콘텐츠를 캐시 서버로 즉시 전달함으로써 클라이언트는 변경된 콘텐츠를 서비스 받을 수 있다.

이상 웹 콘텐츠의 경우를 정리하면, 기본적으로 캐싱을 사용하는 것이 바람직하다. 그러나 크기가 큰 웹 콘텐츠와 캐시 일치성 유지를 위해 푸싱도 아울러 필요하다. 다음으로 스트리밍 콘텐츠의 경우를 살펴 보도록 하자.

스트리밍 콘텐츠는 실시간 재생을 필요로 한다. 따라서 패킷이 제 시간 내에 클라이언트까지 전달되어야 한다. 이를 위해서는 푸싱을 통해 미리 콘텐츠를 클라이언트와 인접한 캐시 서버에 저장 시켜두어야 한다. 그렇지 않다면 초기 요청들은 근원 서버에서 서비스 되고 따라서 QoS 보장이 불가능하다. 푸싱을 통해 미리 캐시 서버에 스트리밍 콘텐츠가 저장되면 초기 요청들은 인접한 캐시 서버에서 패킷 지연이나 분실 없이 서비스를 받을 수 있다.

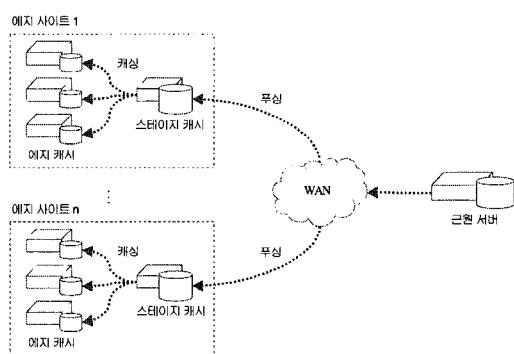


그림 3 푸싱과 캐싱의 혼용 방식

캐싱 방법 역시 스트리밍 콘텐츠에서 쓸모가 있다. 워크로드의 크기가 매우 커서 모든 콘텐츠를 캐시 서버에 저장시키는 것이 여의치 않을 때 캐싱은 매우 유용하다. 캐싱은 교체(replacement) 알고리즘에 의

해 인기 있는 일부 콘텐츠만 캐시 서버에 저장하므로 디스크 공간을 효율적으로 이용한다. 그림 3은 캐싱의 디스크 비용 절감 효과를 잘 보여 주고 있다. 에지 캐시 서버는 모든 콘텐츠를 저장하지 않고 인기 있는 일부 콘텐츠만 저장한다. 따라서 에지 캐시에 설치된 디스크 크기는 상대적으로 작다. 스테이지 캐시 서버는 모든 콘텐츠를 저장하고 있다. 대부분의 요청은 에지 캐시 서버에서 처리되고 에지 캐시 서버에 없는 콘텐츠에 대한 요청은 스테이지 캐시 서버에서 처리된다. 이러한 방식으로 모든 요청은 사이트 내에 설치된 캐시 서버들에서 처리되므로 QoS 보장이 가능하다. 동시에 에지 캐시 서버들의 디스크 크기는 작기 때문에 디스크 비용이 절감된다.

이상 스트리밍 콘텐츠의 경우를 정리하면, 초기 요청들의 QoS 보장을 위해 푸싱이 바람직하다. 또한, 워크로드의 매우 커서 디스크 비용이 문제가 될 때 캐싱 방법을 혼용하여 사용할 필요가 있다.

2.2.2 스플리팅

푸싱과 캐싱은 주문형 스트리밍(on-demand streaming)을 위해 사용되는 전송 방법이다. 라이브 스트리밍을 위해서는 스플리팅(splitting) 기술을 사용한다. 스플리팅 기술은 웹 캐시에서 설명한 오브젝트 스플리팅 기술과 유사한 기술로서 네트워크 트래픽 사용량을 획기적으로 절감한다. 그림 4에서 볼 수 있듯이 캐시 서버는 라이브 스트림 요청 개수에 상관없이 캐시 서버와 근원 서버간에 하나의 스트림 연결만을 생성한다. 캐시 서버는 하나의 스트림 연결에서 받은 데이터를 클라이언트에게 분배한다. 이러한 방식으로 클라이언트의 수에 관계 없이 캐시 서버와 근원 서버간의 트래픽은 매우 낮은 수준을 유지한다.

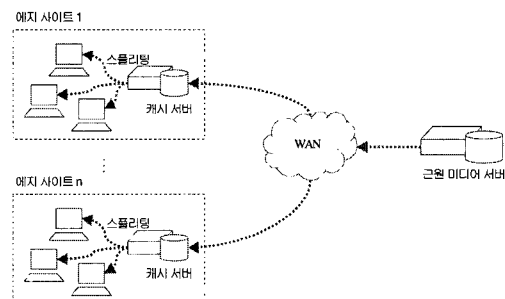


그림 4 라이브 스트리밍을 위한 스플리팅

2.2.3 전송 프로토콜

컨텐츠 전송을 위해 사용하는 프로토콜은 신속하고 안전한 전송을 보장해야 한다. 웹 컨텐츠 서비스를 위한 HTTP, 스트리밍 컨텐츠 서비스를 위한 MMS, RTSP 등의 프로토콜이 전송을 위해서도 사용된다. 그러나 이 프로토콜들은 전송보다는 서비스에 초점을 맞추어 설계되었기 때문에 전송 측면에서는 비효율적일 수 있다. 스트리밍 컨텐츠 전송 시에 이는 극명하게 나타난다. 스트리밍 프로토콜은 정해진 시간에 정해진 분량의 패킷을 전송한다. 따라서 네트워크 회선에 여유가 있다라도 이를 완전히 활용하지 않는다. 이러한 경우에 FTP와 같이 파일 전송을 위해 설계된 프로토콜을 사용하는 것이 바람직하다.

허가되지 않은 침입자에 의해 컨텐츠가 전송 중 유출되는 것을 막기 위해 보안이 확보되어야 한다. 보안이 필요한 경우에는 Secure FTP나 기타 보안이 가능한 전송 프로토콜을 사용한다.

2.3 요청 리다이렉터

요청 리다이렉터는 클라이언트가 가장 인접한 캐시 서버로 접속할 수 있도록 한다[1, 7]. 요청 리다이렉터는 두 가지 주요 기능을 제공한다. 하나는 클라이언트와 가장 인접한 캐시 서버를 선택하는 인접 서버 선택 기능과 다른 하나는 선택된 캐시 서버로 클라이언트의 요청을 보내는 요청 리다이렉션 기능이다. 우선 인접 서버 선택 기능에 대해 살펴 보기로 하자.

2.3.1 인접 서버 선택

인접 서버 선택을 위해서 사용되는 방법으로 정적 맵핑과 동적 맵핑이 있다. 정적 맵핑은 미리 정의된 룰에 따라 클라이언트가 접속할 캐시 서버를 결정한다. 가령, 10.0.1/24 네트워크는 A 캐시 서버로 접속하도록 하고 10.0.2/24 네트워크는 B 캐시 서버로 접속하도록 설정한다. 10.0.1.1의 IP 주소를 가지는 클라이언트가 요청을 하면 정적 맵핑에 의해 이 클라이언트를 위해서는 A 캐시 서버 선택된다.

동적 맵핑은 네트워크 상태를 측정하여 클라이언트가 접속할 캐시 서버를 결정한다. 네트워크 상태의 측정을 위해 홉 개수, 응답 시간, 패킷 분실율 등이 사용된다. 이러한 값들을 구하기 위해서 클라이언트와 캐시 서버들간에 측정을 위한 패킷이 오고 가야 한다.

정적 맵핑과 동적 맵핑은 장단점을 가지고 있다. 성능 면에서는 정적 맵핑이 유리하다. 동적 맵핑은

클라이언트와 캐시 서버간에 인접성 측정을 위한 패킷 전달이 필요 없다. 따라서 매우 신속한 인접성 계산이 가능하다. 그러나 관리 측면에서는 동적 맵핑이 유리하다. 가령, 특정 사이트의 네트워크 주소가 변경되거나 추가되었을 때 관리자는 이를 파악해야 하고 정적 맵핑 규칙을 갱신 해야 한다. 그러나 동적 맵핑에서는 이러한 작업이 필요 없다.

2.3.2 요청 리다이렉션

인접 서버 선택이 완료되었으면 다음으로 클라이언트가 선택된 캐시 서버로 접속하도록 클라이언트의 요청을 조정해야 한다. 가장 빈번히 사용되는 방식으로 DNS 리다이렉션이 있다. DNS 리다이렉션은 클라이언트의 DNS 질의에 대해 선택된 캐시 서버의 IP를 응답하는 방식으로 클라이언트가 선택된 캐시 서버로 접속하도록 한다. DNS 리다이렉션의 단점은 클라이언트의 정확한 위치를 알 수 없다는 것이다. 대부분의 경우 클라이언트는 TCP/IP 설정에 지정된 로컬(local) DNS에게 DNS 질의를 맡긴다. 따라서, 요청 리다이렉터는 실제 클라이언트가 아니라 로컬 DNS로부터 질의를 받기 때문에 요청 리다이렉터는 클라이언트의 위치를 알 방법이 없다. Id로 인해, 만약 클라이언트가 인접한 네임 서버를 이용하지 않는다면 DNS 방식의 요청 리다이렉션은 부정확한 결과를 초래한다.

이를 해결하기 위한 방법으로 웹 컨텐츠를 위한 HTTP 리다이렉션과 스트리밍 컨텐츠를 위한 메타파일 리다이렉션이 있다. HTTP 리다이렉션은 HTTP 프로토콜의 302 응답 코드[?]를 사용한다(그림 5-b 참조). 요청 리다이렉터는 웹 서버로 동작하여 클라이언트 요청 시 302 응답 코드와 함께 인접 서버의 주소를 반환한다. 클라이언트의 웹 브라우저는 302 응답 메시지에 포함된 인접 서버의 주소로 연결을 시도한다. 이 방법에서는 클라이언트가 바로 요청 리다이렉터에게 요청을 보내기 때문에 실제 클라이언트의 위치를 정확히 알 수 있다.

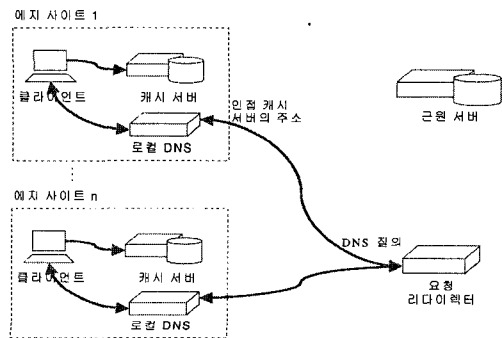
스트리밍 컨텐츠를 위해서는 메타파일 리다이렉션이 DNS 리다이렉션 문제의 해결책이 될 수 있다. 가령, 윈도우즈 미디어에서 사용되는 ASX 파일은 스트림 정보를 기술하는 메타파일로서 ASF 파일의 위치 정보를 포함하고 있다. 요청 리다이렉터는 클라이언트의 ASX 파일 요청 시 위치 정보에 인접 서버의 주소를 기입함으로써 클라이언트가 인접한 서버

로 접속하도록 한다. 메타파일 리다이렉션에서도 클라이언트가 바로 요청 리다이렉터에게 요청을 보내기 때문에 실제 클라이언트의 위치를 알 수 있다. 다만, 기존 시스템이 메타파일을 사용하고 있지 않다면 메타파일을 사용하도록 시스템의 변경이 필요하다.

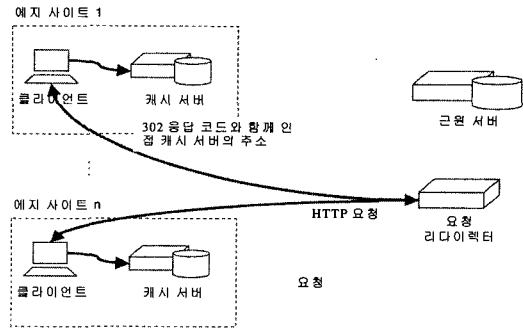
요청 리다이렉션의 또 다른 방식으로 L4 스위치를 사용하는 방식이 있다. L4 스위치를 각 에지 사이트마다 설치하고 근원 서버로 가는 패킷을 캐시 서버로 가로채는 방식이다. 이를 위해 목적 IP가 근원 서버이고 해당 애플리케이션의 포트를 가지는 패킷이 캐시 서버로 전달되도록 L4 스위치에 규칙을 추가해야 한다. 이 방식 역시 DNS 리다이렉션에서의 문제점을 해결한다. 하지만 근원 서버의 IP가 변경되면 모든 L4 스위치의 규칙을 수정해야 하는 불편함이 따른다. 또한 에지 사이트의 모든 클라이언트들이 단일 지점을 통과해야 한다는 제약이 따른다. 따라서 L4 스위치 리다이렉션은 기업 등의 폐쇄적 네트워크 환경에서 사용 가능하다. 지금까지 설명한 네 가지 리다이렉션 방법의 장단점을 정리하면 표 1과 같다.

표 1 요청 리다이렉션 방식 비교

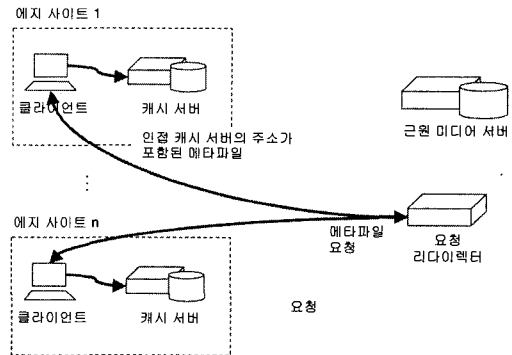
	DNS 리다이렉션	HTTP 리다이렉션	메타파일 리다이렉션	L4 스위치
클라이언트 위치 파악	불가능	가능	가능	N/A
지원 서비스	모든 서비스	웹	스트리밍	모든 서비스
관리 편의성	편리	편리	편리	불편. 모든 L4 스위치를 단일 인터페이스로 관리할 수 있는 통합 관리 도구를 통해 개선될 수 있음



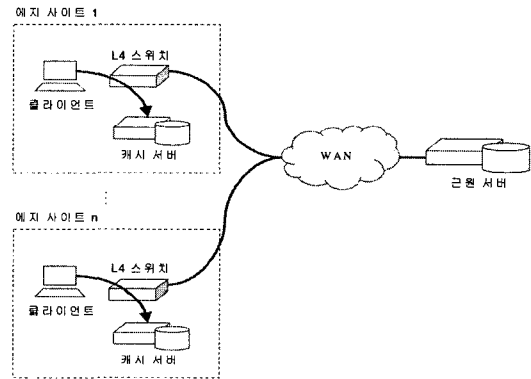
(a) DNS 리다이렉션



(b) HTTP 리다이렉션



(c) 메타파일 리다이렉션



(d) L4 스위치 리다이렉션

그림 5 요청 리다이렉션

3. CDN 경향

본 장에서는 차세대 CDN으로 각광받고 있는 eCDN과 P2P CDN에 대해 살펴해보도록 하겠다. eCDN은 기업 환경을 위한 CDN으로 CDN 솔루션 업체들이 새로운 목표 시장으로 정하고 공격적인 마케팅을 하고 있는 분야이다. P2P CDN은 P2P 기술을 CDN에 접목하여 콘텐츠 전송 및 서비스 시 클라이

인트 자원을 활용한다. CDN의 차기 기술로써 최근 관심이 집중되고 있다.

3.1 eCDN

네트워크는 기업 내에서 직원 및 부서들 간의 정보 공유 및 업무 연계의 효율을 위해 매우 중요한 역할을 담당해 왔다. 일반적으로 기업들은 보안을 위해 전용 네트워크 라인을 구축하거나, VPN 등을 통해 보안이 확보된 공중 라인을 임대한다. 어떤 경우이든 기업의 네트워크 라인은 비용 문제로 인해 대역폭이 매우 제한적이다. 지금까지는 네트워크를 통해 전달되는 대부분의 정보가 텍스트 형태였으므로 문제가 되지 않았다. 그러나 기업에서도 멀티미디어 콘텐츠의 수요가 급속히 증가함에 따라 기업 인트라넷은 한계에 다다르고 있다. 멀티미디어 콘텐츠를 요구하는 대표적 응용 분야로 e-러닝(e-learning), 사내 방송, 대규모 문서 전달이 있다.

상기 언급된 응용 분야들의 공통된 근본적 문제점은 네트워크 회선에서의 병목 현상이다. 이에 대한 솔루션으로 eCDN이 새롭게 각광을 받고 있다. eCDN은 기업의 멀티미디어 콘텐츠를 클라이언트와 인접한 위치로 미리 이동시킴으로써 고품질의 멀티미디어 콘텐츠 서비스가 네트워크 환경이 상대적으로 열악한 지사 혹은 해외 영업소에서도 가능하도록 한다.

eCDN을 위해서 캐시 서버가 반드시 지원해야 할 기능은 다양한 인증 프로토콜의 지원이다. 현재 사내 정보는 허가된 사용자만이 접근할 수 있도록 서버 측에 인증 규칙이 설정되어 있을 것이다. 캐시 서버도 이러한 규칙에 따라야만 한다. 잘 알려진 인증 프로토콜에는 RADIUS, NTLM 등이 있다. 사실 기업에서 운영중인 기존 인증 시스템들을 보면 잘 알려진 프로토콜을 사용하지 않는 경우가 많다. 이러한 경우에는 캐시 서버의 커스터마이제이션을 통해 해결해야 한다.

2.3.2절에서 언급한 요청 리다이렉션의 방법 중 DNS 리다이렉션은 eCDN에 적용되기가 매우 어렵다. 이유는 DNS 리다이렉션에서 요청 리다이렉터에게 요청을 보내는 호스트는 실제 클라이언트가 아닌 클라이언트에서 지정한 로컬 DNS이기 때문이다(그림 5-a 참조). 일반적으로 기업 내의 컴퓨터들은 로컬 DNS로 회사에서 지정한 하나의 DNS를 설정한

다. 따라서 모든 클라이언트의 요청은 하나의 로컬 DNS를 거쳐서 요청 리다이렉터에게 전달되기 때문에 인접 서버 선택이 무의미해진다.

이 문제를 해결하기 위해서 지사마다 별도의 DNS를 설치하고 이용하는 방법이 있다. 하지만 이 방법은 지사별로 DNS를 관리해야 하고 클라이언트들이 로컬 DNS 설정을 변경해야 하는 부담을 초래한다. 또 다른 방법으로 기존에 회사 내에서 운영 중이던 DNS를 요청 리다이렉터로 변경하는 방법이다. 물론 이를 위해서는 요청 리다이렉터가 기존 DNS 기능을 완벽하게 지원해야 한다. 하지만 이 방법도 DNS 서버를 관리하던 회사 내의 전산 팀이 과히 좋아할 만한 솔루션은 아니다.

3.2 P2P CDN

P2P CDN은 콘텐츠 전송과 서비스를 위해 P2P 기술을 응용한다. P2P CDN의 가장 큰 특징은 클라이언트 자원의 활용이다(그림 6 참조). 이로 인해 P2P CDN은 기존 CDN에 비해 두 가지의 장점을 제공한다. 첫째, 클라이언트가 캐시 서버가 아닌 인접한 클라이언트로부터 콘텐츠를 전송 받기 때문에 콘텐츠 제공자들은 서버 확장성 문제에서 해방될 수 있다. 둘째, 캐시 서버보다 더욱 인접한 클라이언트로부터 콘텐츠를 전송 받기 때문에 더욱 고품질의 콘텐츠 서비스가 가능하다.

P2P CDN은 콘텐츠가 클라이언트의 디스크에 저장되는 것을 허용하기 때문에 콘텐츠의 복제를 통제할 수 있는 DRM(Digital Right Management) 기술 [8]을 반드시 필요로 한다. DRM을 통해 콘텐츠 제공자는 클라이언트간의 콘텐츠 복제를 감독 관리 할 수 있다.

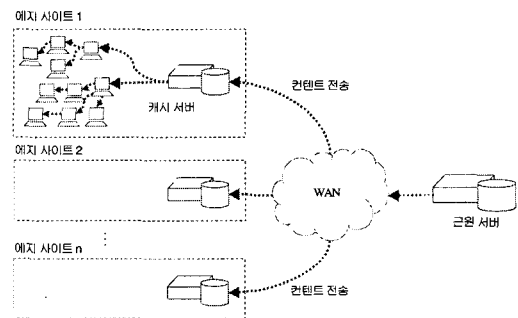


그림 6 P2P CDN 아키텍처

4. 결론

지금까지 CDN을 구성하는 핵심 기술들과 향후 경향들에 대해 살펴보았다. CDN은 시장과 기술 모두 오픈 되어 있는 분야이다. 미국의 경우 1999년부터 본격적인 시장을 형성하였고 국내의 경우 2000년부터 CDN 시장이 생기기 시작하였다. CDN은 아직도 버전 1.0의 기술일 뿐이다. 아직 어느 업체도 승자라고 얘기 할 수 없다. 시장이 확장됨에 따라 보다 새로운 응용들과 이에 대응하는 개념과 기술들이 나타날 것이다.

참고문헌

- [1] Yair Amir and Alec Peterson. Seamlessly Selecting the Best Copy from Internet-Wide Replicated Web Servers. In Proceedings of 12th International Symposium on Distributed Computing, Sep. 1998.
- [2] Apple Computer, Inc. QuickTime API Documentation.<http://developer.apple.com/techpubs/quicktime/qtdevdocs/RM/frameset.htm>
- [3] Greg Barish and Katia Obraczka. World Wide Web Caching: Trends and Techniques. IEEE Communications, Internet Technology Series, May 2000.
- [4] Ethendranath Bommaiah, Katherine Guo, Markus Hofmann, and Sanjoy Paul. Design and Implementation of a Caching System for Streaming Media over the Internet. In Proceedings of IEEE Real-Time Technology and Applications Symposium (RTAS), Washington D.C., USA, May 31-June 2, 2000.
- [5] J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol--HTTP/1.1(rfc2616).<http://www.w3c.org/Protocols/rfc2616/rfc2616.html>
- [6] James Gwertzman and Margo Seltzer. World-Wide Web Cache Consistency. In Proceedings of 1996 USENIX Technical Conference, pages 141-151, San Diego, CA, January 1996.
- [7] Jussi Kangasharju and Keith W. Ross. Locating Copies of Objects Using the Domain Name System. In Proceedings of 4th Web Caching Workshop, San Diego, CA, March 1999.
- [8] Microsoft Corporation. Digital Content Delivery Using WindowsMedia DRM: Benefit from it,http://www.microsoft.com/windows/windowsmedia/wm7/wmrm_business_value.pdf
- [9] Microsoft Corporation. Windows Media Overview.<http://www.microsoft.com/windows/windowsmedia/overview/default.asp>
- [10] RealNetworks. Helix Universal Platform: Content Networking.http://docs.real.com/docs/rm/howto/ContentNetworkingwithHelix_WP_7_21.pdf

최 승 락



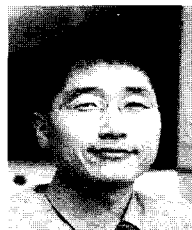
1993~1997 울산대학교 전자계산학과 학부 졸업
 1997~1999 한국과학기술원 전산학과 석사 졸업
 1999~현재 한국과학기술원 전산학과 박사 과정 재학 중
 E-mail:sichoi@observer.kaist.ac.kr

양 철 응



1989~1993 한국과학기술원 전산학과 학사 졸업
 1993~1995 한국과학기술원 전산학과 석사 졸업
 1995~현재 한국과학기술원 전산학과 박사 과정 재학 중
 2001~2002 (주)아라기술 수석연구원
 2002~현재 (주)아라기술 네트워크 연구소 연구소장
 E-mail:cwyang@aratech.co.kr

이 중 식



1985~1989 연세대학교 전산학과 학사 졸업
 1989~1991 연세대학교 전산학과 대학원 석사 졸업
 1991~1995 삼성전기 종합연구소 CAD/CAM실
 1996~1998 삼성전기 종합연구소 E-CIM팀
 1998 삼성그룹 비서실 인터넷T/F
 1999~2000 삼성전기 본부기획 인터넷 T/F(신규사업기획)

2000~2001 아라기술 영업&마케팅 총괄 이사
 2002~현재 아라기술 전략기획 이사
 E-mail:jslee@aratech.co.kr