

# CDN 성능 측정

임병희<sup>o</sup> 정재일 도현택  
한양대학교 전자통신전파공학과  
JSCOOL75@ihanyang.ac.kr<sup>o</sup> jijung@hanyang.ac.kr taegido@mnlab.hanyang.ac.kr

## CDN Performance Estimation

Byounghui Lim<sup>o</sup> Jaeil Jung Hyuntaek Do  
Dept. of Division of Electrical and Computer Engineering, Hanyang University

### 요 약

인터넷 콘텐츠와 애플리케이션에 대한 요구가 날로 급증하고 있는 상황에서 인터넷의 성능 즉, 접속 품질과 속도에 대한 만족도는 현저히 개선되지 않고 있다. 그 동안 인터넷의 성능향상을 위한 꾸준한 노력의 결과로 많은 기술들이 개발되었지만, 망 대역폭 용량을 확장하거나 새로운 기술을 적용하더라도 정체현상은 여전히 되풀이되고 있다. 이와 같은 상황에서 인터넷의 성능을 향상시킬 수 있는 방안 하나로 최근 CDN(Content Delivery Network)에 관심을 쏟고 있다.

본 논문에서는 이러한 CDN 서비스의 성능 측정을 위한 구조 설계에 관해 논하게 된다. 특히 CDN 기술을 이용한 웹 콘텐츠가 사용자의 요청에 신속하게 제공되고 있는지를 평가하기 위하여, 단말의 상위 계층에서 시스템 구조를 설계한다.

### 1. 서 론

CDN 서비스는 지능적이고 동적인 네트워크 인프라로써 인터넷 비즈니스를 하는 사업자에게 콘텐츠를 최대한 신속하게 제공하여 사용자의 요구를 충족시키고자 하는 새로운 네트워크 비즈니스이다. 일반적으로 콘텐츠는 CP(Content Provider)의 서버로부터 ISP의 백본 네트워크, IX(Internet Exchange), 가입자망 등 복잡한 경로를 거쳐 사용자에게 전달된다. 이 과정에서, 미들마일(Middle-Mile)이라 불리는 CP 서버로부터 사용자 측 노드에 이르는 구간이 인터넷 속도저하의 원인이 되고 있다. CDN은 콘텐츠가 미들 마일 구간을 경유하지 않고 인터넷 사용자의 가장 가까운 지점에서 콘텐츠를 전송하므로 속도 저하의 문제를 해결하게 된다. 근래에 들어 CDN이 인터넷 방송, 동영상 포털, 쇼핑몰 등 대량의 데이터를 전송해야 하는 인터넷 서비스의 구현에 있어 해결책으로 제시되고 있는 상황에서 진정한 의미에서 종단간(end-to-end)의 CDN 서비스의 품질을 보장할 수 있는 측정 시스템 개발이 필요하다. 본 논문에서는 proxy 방식을 이용한 CDN 성능을 측정한다.

2장에서는 CDN 콘텐츠 전달 방식에 대해 소개하고, 3장에서는 웹 성능 측정 방법 및 구조에 대해서 설명하고, 4장에서는 프락시 모듈의 구현에 대해서 자세히 알아보고, 5장에서 결론을 맺는다.

### 2. CDN 콘텐츠 전달 방식

일반적으로 우리가 컴퓨터의 브라우저를 통해서 특정한 제공사업자의 콘텐츠를 받아 보기 전까지는 다음과 같은 단계를 거치게 된다. 인터넷의 edge에 위치한 cache/streaming 서버들은 실제로 인터넷상의 중요한

전략적 위치 및 사용자들이 가입되어 있는 ISP들의 POP에 설치되는데, 물론 한 두 개가 아니라 곳곳에 다량의 CDN 서버들이 설치된다. 현재 일반적으로 사용되고 있는 기술은 사용자의 local DNS의 IP 주소를 기반으로 최적의 edge farm을 선정한다. 이 서버들은 일종의 캐시 서버로서 작동하며, 사용자들의 요구가 있을 때마다 콘텐츠 제공사업자의 웹 서버 대신 자체의 저장 장치에 저장되어 있는 콘텐츠를 전달해주는 역할을 하게 된다 [1].

인터넷 사용자들이 콘텐츠를 받아보는 과정은 다음과 같다.

[1단계] URL 입력

사용자가 브라우저상에서 방문하고자 하는 사이트의 URL을 입력하면 인터넷을 통해 콘텐츠 제공사업자의 웹 서버로부터 request가 전달된다.

[2단계]

HTML 전송 request를 받은 콘텐츠 제공사업자의 웹 서버는 CDN 서버를 가르키고 있는 embedded URL를 포함한 HTML을 사용자에게 전송한다. 즉 embedded URL이 가르키고 있는 주소들이 웹서버나 다른 서버의 주소가 아니라 바로 CDN 서버의 주소라는 점이다.

[3단계] CDN 서버에 embedded object 요청

사용자의 브라우저는 HTML에 포함된 embedded URL의 주소, 즉, CDN 서버로 object를 요청하게 된다.

[4단계] local CDN 서버로부터 콘텐츠 전달

CDN서버에 복제되어 있던 콘텐츠가 사용자의 웹브라우저로 전송된다.

3. 중단간 성능 측정 및 구조

3.1 Proxy 방식을 이용한 성능 측정 방식

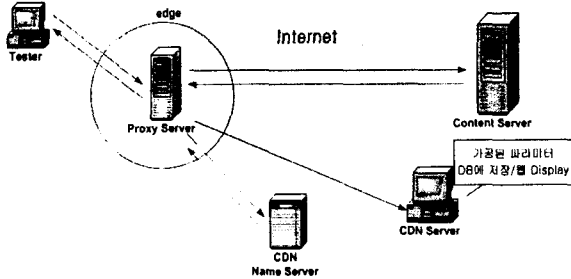


그림 1 웹 서버 성능 측정 구성도

그림 1에 tester와 proxy server는 CDN 측정을 위해 구현된 시스템이다. 여기서 proxy server는 캐쉬 기능을 하지 않으며, 단지 Web server와 인터페이스를 하며, Web server와 측정하는 기능을 한다. 그림 1에서 tester를 가동 후 웹 서버에 HTML 요청을 하게 되면 tester에서 캐시 삭제 과정을 거쳐 proxy server로 request를 전송한다. proxy server는 tester로부터 받은 request를 DNS Lookup 과정을 거쳐 웹 서버로 페이지를 요청하고, proxy server는 요청한 페이지를 웹 서버로부터 수신하게 된다[3][4]. 그 후 proxy server에서 받은 페이지는 tester로 전송한다. 이런 동작 과정을 통해 proxy server에서 웹 사이트의 성능 파라미터를 추출할 수 있다. proxy server에서 추출한 파라미터는 tester와 SQL 서버로 전송한다.

3.2 기능별 구성

본 논문에서 제시한 시스템 구조는 Proxy 방식의 CDN test 기능을 가진 모듈로서, server측의 프락시 프로그램과 클라이언트 측의 tester 프로그램을 이용한 구조이다. 클라이언트 연결 설정, 초기화 명령을 포함한 configuration module과 SQL database 연결, 프락시 설정, 버퍼 캐쉬 삭제, 웹사이트 성능 측정, 측정 결과를 SQL 서버로 전송하는 구조로 되어있다.

3.2.1 서버 모듈

구성 모듈은 클라이언트(tester)에게 측정된 결과를 전송하기 위한 준비를 하며 측정값이 나오는 대로 실시간으로 전송한다.

프락시 모듈은 측정 파라미터(측정 시간, Client IP, Site URL, Site IP, Object URL, Object IP, Object Name, Object Download time, DNS Lookup time, TCP connection Time, First Packet Download time, Object download time, Object Size, Object download speed, Object type)를 추출한다.

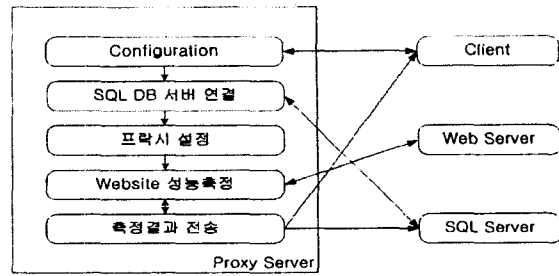


그림 2 Proxy Server의 기능별 구성도

3.2.2 클라이언트 모듈

그림 1에서 tester에 해당하는 부분으로 기존의 object를 삭제 하는 기능과 웹 브라우저 기능이 있으며 프락시 모듈에서 측정된 결과 값을 보여 주는 기능이 포함되어 있다.

4. 프락시 모듈의 동작 및 파라미터 추출 방법

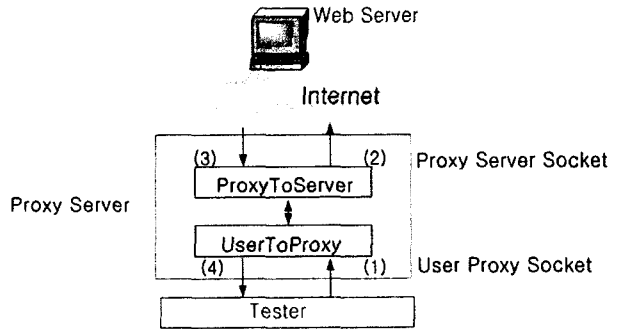


그림 3 프락시 모듈

4.1 프락시 모듈의 동작

그림 3은 그림 1의 proxy server의 내부를 보여주는 그림이다. 프록시 서버는 웹 서버(CDN 서버)와 클라이언트 사이에서 중계 역할을 하는 것이다. 그림 3과 같이 proxy server에서는 소켓을 생성하고 tester의 접속을 proxy server가 기다리게 된다. 이 모듈에는 ProxyToServer와 UserToProxy 모듈이있는데, 전자는 웹 서버와 데이터를 주고 받는 모듈이고 후자는 tester와 데이터를 주고 받는 모듈이다.

접속 요청(request)이 있으면 UserToProxy에 쓰레드가 생성되고 그 요청을 받아들인다. 이 request는 다시 ProxyToServer에 전달되게 되고 결국 이 요청은 웹 서버에 전달되게 된다. 서버의 응답(response)은 위에서 말한 방법의 역순으로 response(또는 data)가 전달되게 된다. 모든 request와 response가 우리가 구현한 proxy server를 통해 지나가기 때문에 파라미터의 값을 얻을 수 있다. 그림 4는 구현한 proxy server이다.

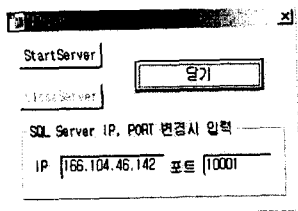


그림 4 Proxy Server

#### 4.2 파라미터 추출 방법

모든 request와 response가 우리가 구현한 proxy server를 통해 지나가기 때문에 이 proxy server를 통해 지나가는 packet의 시간차를 구함으로써 파라미터를 구할 수 있다[3~6].

DNS LookUp Time은 도메인 이름을 IP주소로 변환하게 되는데 이때 걸리는 시간이다. Initial Connection Time은 접속할 서버와 TCP connection을 연결 시점과 (그림3의 (1))SYN/ACK packet을 받는 시점(그림 3의 (4))의 시간차이다. 이것으로 서버와의 지연(delay) 정보를 파악하여 해당 네트워크에 장애 정도를 알 수 있다. First Packet Time은 TCP connection time을 끝내고 웹 서버의 HTML의 첫 패킷을 받을 때 걸리는 시간이다. 이 시간은 서버 지연(delay) 시간과 동일하기 때문에 서버 지연(Delay) 정보를 알 수 있다. Redirection Time은 original 서버로부터 HTTP 302 redirection packet을 받은 후 CDN 서버로 다시 연결한 후 first packet을 받을 때까지 걸리는 시간이다. 이것으로 CDN의 핵심 기능인 redirection 성능을 평가 할 수 있다. Base Page Download Time은 HTML의 마지막 패킷이 도달할 때까지의 시간이다. 이 페이지는 보통 수 킬로바이트가 전송되기 때문에 정적인 오브젝트 보다 다운로드 시간이 길다. 또한 single thread로 페이지를 다운로드 하기 때문에 전체 웹페이지 구성에 많은 영향을 줄 수 있다. Content Download Time은 HTML 코드로 이루어진 베이스 페이지가 다운로드 되면 브라우저는 HTML을 분석하여 embedded 컨텐츠 오브젝트(images, applets, or sound files)들을 받기 위해서 오브젝트 별로 각각 요청을 하는데 이때 웹 페이지를 이루고 있는 모든 오브젝트들을 다운로드하는데 걸리는 시간이다. Download Speed은 현재 인터넷의 트래픽 중 상당수가 대용량의 파일을 전송받는 것으로 인해 발생한다. 이를 측정하기 위해서 1MB 이상의 오브젝트를 받을 경우에만 평균 다운로드 속도를 측정함으로써 사용자 측면에서 직관적으로 느낄 수 있는 다운로드 속도를 bps단위의 수치로 제공한다.

위에서도 언급한 것과 같이 모든 데이터(request, response)는 우리가 구현한 proxy server를 경유해 지나가기 때문에 데이터들의 시간차를 구함으로써 다양한 파라미터를 구할 수 있다. 그림 5는 이러한 방식으로 추출

한 파라미터의 결과를 보여 주고 있다.

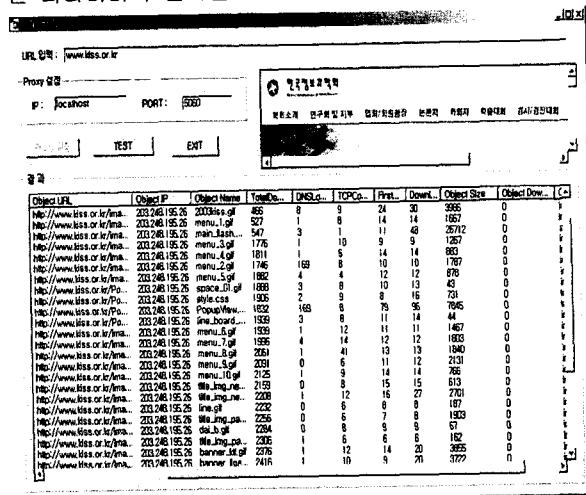


그림 5 Tester의 결과 화면

#### 5. 결론

기존에 웹 성능을 측정하는 툴은 웹상에서 운영이 되었기 때문에 측정하고자 하는 웹 서버와 클라이언트 간의 측정이 아닌 측정하고자 하는 웹 서버와 그 툴을 제공하는 웹과의 측정이다. 또한 제공하는 파라미터는 극히 적고, CDN 서버의 성능 측정은 제공하지 않는다.

하지만 본 논문에서는 웹 서버와 클라이언트 사이에 측정서버(그림 1의 proxy server)를 두어 original 서버와 CDN 서버의 성능을 측정할 수 있다. 일반 사용자는 proxy server를 local에 설치 후 tester를 실행하면, 웹 서버와 CDN 서버를 측정할 수 있다. 관리자는 proxy server를 측정하고자 하는 host에 설치 후 tester를 실행하면 서버(웹과 CDN)를 측정할 수 있다. 따라서 이 논문에서 구현한 방식을 사용함으로써 일반 사용자나 관리자 모두가 웹 서버 및 CDN 서버의 성능을 측정할 수 있으며, 이를 통해 서비스 제공자와 사용자와의 서비스 레벨 협정 계약(SLA) 보장의 근거로 사용할 수 있다.

#### 5. 참고 문헌

- [1] Scot Hull "Content Delivery Networks: Web Switching for Security, Availability, and Speed"
- [2] W. Richard Stevens, "TCP/IP Illustrated, Vol. 1 II III "
- [3] RFC 1945 "Hypertext Transfer Protocol - HTTP/1.0 "
- [4] RFC 2068 "Hypertext Transfer Protocol - HTTP/1.1 "
- [5] Jones, Ohlund, "Network Programming for Windows"
- [6] Dave Roberts, "Internet Winsock2"