SDCDS: 지연시간을 개선한 디지털콘텐트 전송 시스템

나 윤 지[†] · 고 일 석[††]

요 약

중앙집중 구조의 멀티미디어 디지털콘텐트 서비스에서는 서버의 과부하 문제와 네트워크 트래픽의 급격한 증가 문제가 발생한다. 최근에는 이러한 문제점을 해결하기 위한 디지털콘텐트 전송기술로 CDN에 대한 연구가 활발히 진행되고 있다. 본 연구에서는 디지털콘텐트의 전송 및 관리 시스템에서, 보안성능과 처리지연시간을 개선한 시스템인 SDCDS(Secure Digital Content Delivery System)을 설계하였다. SDCDS에서는 CDN에서의 보안성과 이에 따른 처리 지연시간 개선을 목적으로 하고 있다. 이를 위해서는 보안방식과 캐싱 방식을 CDN의 구조적 특성을 고려하여 설계하여야 한다. SDCDS에서는 공개키 기반 보안 방식을 CDN의 구조적 특성을 반영하도록 설계하였고, 암호화된 DC와 일반 DC의 그룹별 캐싱 방식의 향상된 캐싱 성능을 통해 처리 지연시간을 개선하였다. 또한 실험을 통해 제안 시스템의 성능을 평가하였다.

SDCDS: A Secure Digital Content Delivery System with Improved Latency time

Yun Ji Na[†] · Il Seok Ko[††]

ABSTRACT

Generally, the overloaded server problem and the rapidly increased network traffic problem are happened in a center concentrated multimedia digital content service. Recently, a study about the CDN which is a digital content transmission technology to solve these problems are performed actively. In this study, we proposed the SDCDS which improved a process latency time and a security performance on a digital content delivery and management. The goal of the SDCDS is the digital content security and the improvement of the processing time. For that, we have to design the security and the caching method considering the architecture characteristics of the CDN. In the SDCDS, the public key encryption method is designed by considering the architecture characteristics of CDN. And we improved the processing latency time by improved the caching method which uses the grouped caching method on the encrypted DC and the general DC. And in the experiment, we verify the performance of the proposed system.

키워드 : 디지털콘텐트(Digital Content), CDN, DC 보안(Security Of DC)

1. 서 론

콘텐트 네트워킹 기술은 사용자와 보다 인접한 위치에 콘텐트를 준비하고, 사용자에게 가장 인접한 위치의 콘텐트로 안내함으로써 보다 고품질의 콘텐트 서비스가 가능하도록 한다. 콘텐트 네트워킹의 가장 대표적인 기술은 CDN(Content Delivery Network)이다[1, 2]. 지금까지 많은 연구들이 콘텐트 전송의 성능을 향상시키기 위해 대역폭 증가, 전송의 효율화에 중점을 두었지만, 콘텐트 CDN은 네트워킹 기술을 통해 지능적으로 콘텐트 전송의 성능을 향상시킨 것이다. CDN은 콘텐트를 사용자와 보다 인접한 위치로 이동시켜 네트워크 상에서의 콘텐트 전송비용을 줄이며, 체계적 관리를 통해 관리자가 한 지점에서 전체 콘텐트 전송을 제어할 수 있도록 한다.

또한 인터넷은 보안을 고려하지 않은 전송매체이기 때문에 보안 문제가 발생한다. 일반적으로 웹에서 디지털콘텐트(DC: Digital Content) 전송의 안전성을 확보하기 위해 평문을 암호문으로 변환하는 암호화 과정을 통해 전송한다. 이 과정에서, 암호화로 인해 추가적인 부담이 발생하게 되고 암호화된 DC의 전송으로 인한 네트워크 트래픽 증가가 발생하게 된다. 또한 암호화된 DC의 실행을 위해서는 별도의 복호화 과정이 필요하게 되어 클라이언트의 요청 처리에 대한 지연시간을 증가시키게 된다. 처리지연은 네트워크의 물리적인 환경과 암호화 및 암호화된 파일의 전송과 이의 복호화 과정에서 발생한다[3, 4].

따라서 CDN에서 전송되는 DC의 보안성을 유지하기 위해서는 지연시간이 발생하게 된다. CDN은 대규모의 DC의 전송에 활용하기 위한 기술이기 때문에 이러한 지연은 CDN의 성능을 감소시키게 된다.

† 정 회 원 : 호남대학교 인터넷소프트웨어학과 교수
†† 정 회 원 : 충북과학대학 전자상거래과 교수
논문접수 : 2004년 9월 7일, 심사완료 : 2004년 11월 26일

본 연구에서는 DC의 전송 및 관리 시스템에서, 보안성능과 처리지연시간을 개선한 시스템인 SDCDS(Secure Digital Content Delivery System)을 설계하였다. SDCDS에서는 CDN에서의 보안성과 이에 따른 처리 지연시간 개선을 목적으로 하고 있다. 이를 위해서 보안기술과 캐싱 기술[5,6,7]을 CDN의 구조적 특성을 고려하여 설계하여야한다. SDCDS에서는 공개키 기반 보안 방식을 CDN의 구조적 특성을 반영하도록 설계하였고, 암호화된 DC와 일반 DC의 그룹별 캐싱을 통해 캐싱 성능 향상을 통해 처리 지연시간을 개선하였다. 또한 실험을 통해 제안 시스템의 성능을 평가하였다. 실험결과 보안성을 요구하는 DC의 전송이 많을수록 지연시간의 감소를 통한 성능의 향상을 기대할 수 있었다.

2. DC 보안 기술

인터넷에서 웹을 이용하여 클라이언트와 서버사이의 신뢰성과 안전성을 확보하기 위해서는 웹 보안 프로토콜과 평문 데이터를 암호화 및 복호화하는 알고리즘이 필요하다[8]. 웹 보안 프로토콜에는 응용 계층에서 메시지 전체를 암호화하여 전송하는 방식과, 메시지에서 몸체 일부분만을 암호화하여 전송하는 방식이 있다. 웹클라이언트와 서버 사이의 신뢰성과 안전성을 확보하기 위해서는 클라이언트와 서버 각각을 인증서버(CA 서버)를 통하여 인증 받는 절차가 필요하다. 이때 사용되는 데이터의 집합이 인증서이다.

DC의 보호를 위한 암호화 방법은 DES, SEED 알고리즘과 같은 대칭키 방식과 RSA(Rivest-Shamir-Adelman) 알고리즘과 같은 비대칭키 방식이 있다. 이들 기존에 개발된 알고리즘들은 DC의 보호를 위한 각종 요건을 갖추고 있으며, 각종 시스템에 다양하게 응용되고 있다. 대칭키 콘텐트의 암복호화만 가능하여 키값의 교환을 위해서는 Diffie-Hellan (DH)와 같은 별도의 키 분배 기법이 필요하게된다. 제안 시스템에서는 키 분배와 암/복호화 및 인증이 가능한 RSA 공개키 기법을 사용한다.

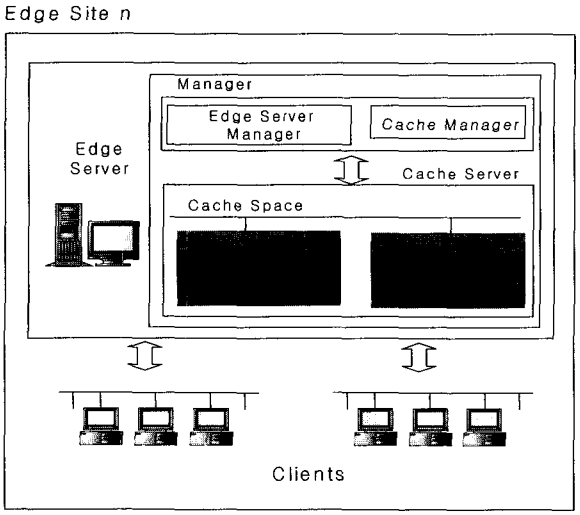
공개키 암호방식은 암호화할 때 사용하는 키(공개키(public key))와 복호화할 때 사용하는 키(비밀키(private key) 또는 개인키)가 달라서 공개키는 공개하고 비밀키만 안전하게 유지하는 방식이다[9, 11]. 공개키 기반의 웹 보안 시스템에서는 클라이언트와 웹서버의 인증이 종료된 후에 암호화 채널을 형성하여 실질적으로 데이터를 주고받을 때는 공개키와 비밀키를 이용한다. 공개키 개념이 발표된 후 많은 알고리즘이 발표되었지만, 현재 가장 안전성을 인증 받고 있는 것은 RSA와 Diffe-Hellman 알고리즘 등이 있다. 웹 상용 보안시스템의 경우 RSA 공개키 알고리즘을 이용하고 있다. RSA 방식은 계산에 소요되는 시간이 대칭키 방식에 비해 오래 걸리지만 공개키 방식은 대칭키 방식에 비해 암호화 과정에서 사용하는 키의 안전한 분배가 용이하여 상용 시스템에서 많이 사용되고 있다[10].

3. 시스템 설계

시스템은 DC를 제공하는 근원서버와 클라이언트 그룹과 캐시서버로 구성된 에지사이트로 구성되어 있다. 근원서버와 에지사이트간의 DC의 전송방식은 객체의 요청에 의해 근원서버에서 전송받는 캐싱기법을 사용한다. 에지사이트는 (그림 1)과 같이 다수의 클라이언트와 에지서버로 구성된다. 에지서버는 에지서버관리자와 캐시서버 및 캐시서버관리자로 구성되어 있다.

3.1 키 분배와 인증절차

에지서버는 클라이언트가 캐시 목록에서 원하는 DC를 찾을 수 없을 경우 근원서버에서 해당 DC를 전송받아야 한다. 이 경우 근원서버와 에지서버간에는 암호화 데이터를 주고받기 전에 CA(Certificate Authority) 서버를 통해 인증서를 발급 받아야한다. (그림 2)는 키 분배 및 DC의 전송 절차를



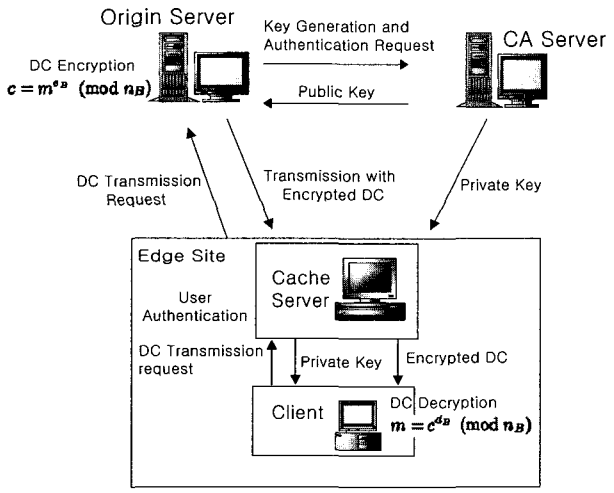
(그림 1) Edge Site 아키텍처

나타낸 것이다.

암호화와 관련된 특허출원 중에서 RSA 방식과 직·간접적으로 관련된 것이 60% 이상을 차지하고 있어, 대부분의 상용시스템에서의 RSA 방식을 활용하고 있다. 마찬가지로 본 연구에서 사용하는 암호화는 RSA 방식을 기반으로 하고 있다. SDCDS는 RSA 방식을 기반으로 다음과 같이 키 분배 및 인증 절차를 수행한다.

먼저, 근원서버는 CA 서버에 인증서 발급을 요청한다. CA 서버는 인증 요청서를 근원서버와 에지사이트(에지사이트를 구성하는 에지서버)로 전송한다. 근원서버와 에지사이트는 자신의 키 쌍을 생성하고, 인증 요청서를 작성한다.

인증서는 일련번호, 발행자, 발행일, 만기일, 소유자, 소유자공개키, 발행자서명과 같은 인증에 필요한 구조를 포함하고 있다. 인증서 형식 중 발행자와 발행자 서명은 CA와 관련된 필드이며, 이 필드는 인증서의 교환에서 인증서 안의 키가 인증서의 소유자의 것임을 CA가 보장함을 의미한다. 각 인증서는 소유자의 공개키를 포함한다. 그러므로 그 공개키는 인증서의 소유자에게 전달할 데이터의 암호화에 사용할 수 있다. 또한 인증서는 발행한 CA의 디지털 서명도 포함한다. 따라서 인증서가 수정되지 않았다는 것과 그 안에 저장된 정보의 신뢰성을 보증하는 것이다.



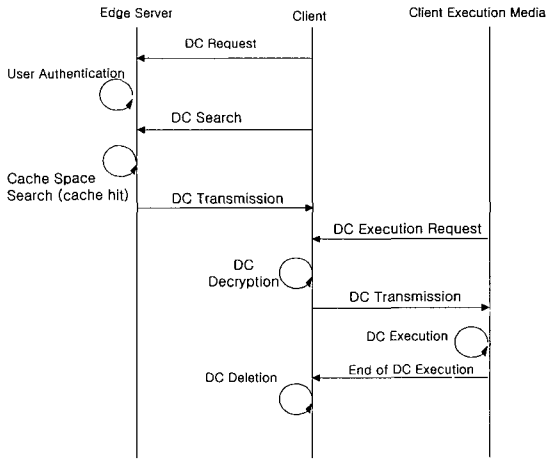
c: encrypted message
m: plain message(Content m(0<m<n))
e_B, n_B: public key
d_B, n_B: private key

(그림 2) 키 분배와 DC 전송 절차

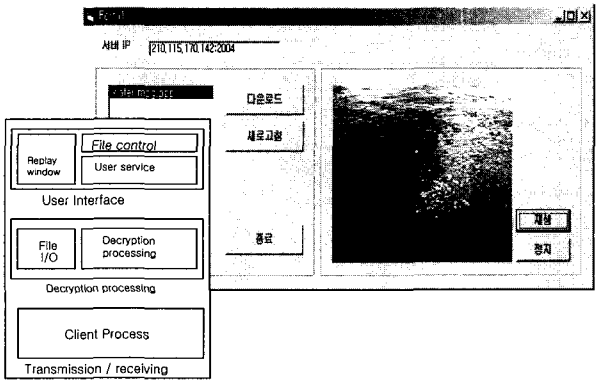
다음으로 근원서버와 에지사이트는 자신의 공개키와 인증요청서를 CA 서버로 전송한다. CA 서버는 수신된 인증 요청서를 확인하여 공개키를 포함한 인증서 발급한다. CA 서버는 근원서버와 에지사이트의 인증 요청서 정보와 인증서를 DB에 저장한다. 다음으로 CA 서버는 인증서를 근원서버와 에지사이트에 전송한다. 근원서버와 에지사이트는 CA 서버로부터 수신된 자신의 인증서를 비밀키와 함께 저장한다. 다음으로 에지사이트는 CA 서버로부터 수신된 비밀키를 자신의 에지사이트에서 인증된 클라이언트에게만 전송하고 클라이언트는 이 비밀키를 별도로 저장한다.

(그림 2)의 방식에서 각 클라이언트는 에지사이트의 인증을 통해 캐시서버에 저장된 암호화된 DC를 전송받을 수 있게 되어, 별도의 보안절차를 가질 수 있다. 에지사이트에 소속된 클라이언트는 동일한 비밀키를 사용하는 그룹이 되며, 에지사이트의 클라이언트 중에서도 암호화된 DC의 사용 권한에 대한 인증을 받지 못한 경우 비밀키의 전송을 받지 못하며, 암호화된 DC에 대한 권한을 가질 수 없게 된다. 결국 CDN의 에지사이트가 DC의 보안을 위한 키관리 및 인증 기능을 가지게 된다. 이것은 대규모의 DC 분배에서 각각의 클라이언트에 대한 관리가 중앙에서 이루어지는 시스템의 경우, 보안성의 확보를 위해 CA 서버 및 근원서버에 발생하는 부담을 줄일 수 있게 한다.

또한 에지사이트에서는 해당 사이트의 클라이언트와 비밀키 방식의 암/복호화를 통해 안전성을 확보하게 된다. 비밀키 방식은 공개키 방식에 비해 복잡성과 부담을 줄일 수 있지만, 키 분배를 위한 별도의 처리를 요구하게 된다. 하지만 제안시스템에서는 에지서버가 CA서버를 통해 이미 분배받은 비밀키를 클라이언트에서 이용하기 때문에 별도의 키 분배에 대한 부담을 줄일 수 있다.



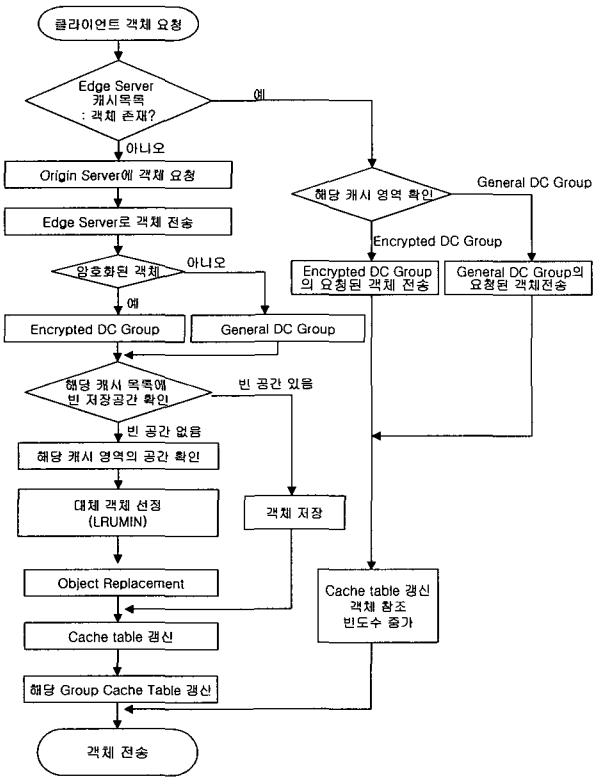
(그림 3) 에지서버에서 암호화된 DC의 전송과 실행 과정



(그림 4) Client Execution Media에서 파일의 수행

3.2 에지서버에서 DC 전송과 실행 과정

(그림 3)은 에지서버에서 클라이언트로 DC의 전송과 실행 과정을 나타낸 것이다. 클라이언트의 DC를 전송 요청한다. 시스템 자체에서 사용자 인증을 수행한다. 캐시 목록에서 DC 검색한다. 캐시적중(cache hit)일 때 해당 DC를 전송하게 된다. 전송 완료 후 클라이언트에서 전송받은 DC의 복호화 수행한다. 키 값에 의한 사용자 인증 절차를 수행한다. 복호화된 DC를 실행한다. 실행은 클라이언트 실행 미디어(Client Execution Media)에 의해 이루어지며, 실행을 완료 후에는 클라이언트에서 DC를 삭제한다. (그림 4)는 Client Execution Media에서 복호화된 mpeg 파일이 실행된 것을 보인 것이다.



(그림 5) 캐싱 기법

3.3 에지 서버에서의 캐싱

CDN의 구성에서 캐시서버는 근원서버와 클라이언트 사이의 에지사이트 내에 존재하게 된다. 캐시서버는 근원서버 대신 클라이언트에 직접 DC 객체를 서비스하는 모듈이다. 따라서 CDN의 처리 성능은 캐시 성능에 좌우된다고 할 수 있다. 캐시서버는 DC의 저장을 위해 저장공간과 관리테이블을 캐시관리자를 통해 유지하고 있다. 궁극적으로 캐시서버의 저장공간에는 클라이언트에서 요청되는 모든 DC 객체를 저장하고 있는 것이 최상의 서비스를 제공할 수 있지만, 물리적인 저장공간의 한계로 인해 모든 DC 객체를 저장할 수는 없다. 따라서 새로운 객체의 저장을 위해서는 기존의 객체를 캐시저장공간에서 삭제하고 이 영역에 새로운 객체를 저장하여야 한다. 이를 대체(Replacement)기법이라 한다. 캐시서버의 캐싱 능력은 결국 이 대체기법의 효율성에 좌우된다.

제안시스템에서 캐시가 저장하고 관리해야할 자료에는 암호화된 콘텐츠와 일반 자료가 있다. 따라서 캐시 테이블은 영역별 관리 구조를 가지고 있다. 전체 캐시관리를 위한 테이블과 암호화된 DC의 저장과 관리를 위한 테이블, 일반 DC의 저장과 관리를 위한 테이블로 구성되어 있다.

(그림 5)는 SDCDS의 캐싱 기법을 나타낸 것이다. 클라이언트의 객체 요청이 들어오면 캐시 관리자는 해당 객체그룹의 목록에서 존재 여부를 확인한다. 이때 캐시의 저장 영역에 해당 객체가 있을 경우 사용자가 요구한 객체를 클라이언트로 전송하게 된다. 이때 해당 객체는 캐시 테이블에서 참조빈도수가 1 증가한다. 캐시 목록에서 요청된 객체를 찾을 수 없을 경우 근원서버에게 객체 전송을 요청하여 전송받는다. 전송 받은 DC는 객체의 암호화 정보에 따라 암호화된 DC 그룹과 일반 DC 그룹으로 분류되고 캐시 관리자는 해당 그룹의 캐시 영역에 이 객체를 저장할 공간이 있는지를 확인한다.

저장할 공간이 있을 경우 객체를 해당 캐시 영역에 저장하고 캐시 목록을 갱신하고, 없을 경우 해당 그룹의 캐시 영역에 대해 캐시 대체 기법을 이용하여 객체를 대체한 후 캐시 목록을 갱신한다. 이때 사용할 수 있는 웹 캐시 대체 기법은 다양하지만 여기에서는 LRUMIN을 사용한다.

4. 실험 및 분석

CDN을 통한 DC의 전송에서 중요한 이슈는 보안성과 객체의 처리에 소요되는 처리지연시간을 개선하는 것이다. 따라서 제안 시스템의 효용성은 보안성에 관련된 분석과 CDN에서 발생할 수 있는 지연요인에 대한 분석을 통해 입증될 수 있다.

4.1 DC 안전성과 키 분배 방식 분석

비밀키는 공개키에 의해 암호화된 DC를 복호화하는데만 사용된다. 그러므로 근원서버에서는 중앙의 관리자(CA)로부터 수신자의 공개키를 찾음 다음, 그 공개키를 사용하여 보내는 DC를 암호화한다. 수신자는 그것을 받아서, 자신의 비밀키로 복호화한다. 근원서버에서는 각 에지사이트에 대한 키 값으로 DC를 암호화하게 되므로, 각 클라이언트의 공개키로 DC 각각을 암호화하는 부담을 줄일 수 있다. 또한 DC 사용그룹인 에지사이트의 사용자는 각 에지사이트 내에서만 인증되므로, 사용자 인증이 빨라지고 편리해진다. 또한, 제안 시스템에서는 DC의 네트워크 트래픽으로 인한 영향이 감소되고, 에지사이트의 캐시에서 서비스되는 DC의 영향을 받게 된다. 따라서 클라이언트에서의 복호화 시간이 감소되어 실행속도를 향상시킬 수 있다.

또한 근원서버에서는 각 에지사이트에 대한 키 값으로 DC를 암호화하게 되므로, 각 클라이언트의 공개키로 DC 각각을 암호화하는 부담을 줄일 수 있다. 또한 에지사이트에

서는 RSA 공개키 방식으로 암호화된 DC를 복호화한 후, 비밀키 방식으로 암호화하여 별도의 캐시 공간에 관리한다. 비밀키 방식은 공개키 방식보다 지연 요인이 감소할 수 있다는 장점이 있다. 에지사이트 내에서는 공개키 방식보다 복잡성과 부담을 감소시킨 비밀키 방식을 이용하므로 각 클라이언트는 공개키 방식보다 복호화에 따른 지연요인을 줄일 수 있고, 각 클라이언트 각각이 별도의 비밀키를 CA를 통해 분배받아야하는 부담을 없앨 수 있다.

4.2 캐싱 및 처리 지연시간 분석

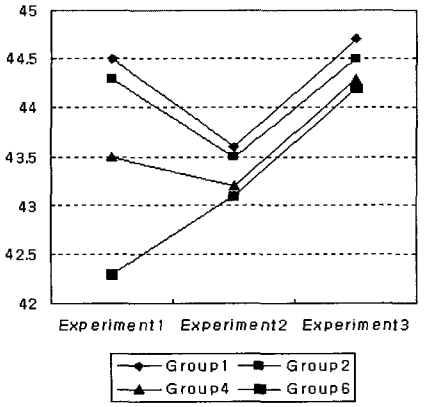
CDN에서의 처리지연시간 LT(Latency Time)는 클라이언트가 객체를 요청한 후 응답을 받아 실행할 때까지 걸리는 시간이다. 클라이언트의 객체 요구에 대한 응답시간은 다음과 같은 지연 요소가 발생하게 된다.

① $TDT_{client\_to\_cache}$: 클라이언트에서 캐시서버간의 객체 요구에 대한 전송 지연시간
② DLT_{Edge} : 에지서버에서 공개키 방식으로 암호화되어 전송된 DC를 복호화하는데 걸리는 지연시간
③ ELT_{Edge} : 에지서버에서 복호화된 DC를 비밀키 방식으로 암호화하는데 걸리는 지연시간
④ DDT_{cache} : 캐시서버의 cache hit 또는 cache miss의 판별 지연시간
⑤ SDT_{cache} : 캐시저장 영역에서 분할 영역별로 저장된 객체의 캐시 검색 지연시간
⑥ $TDT_{cache\_to\_client}$: 캐시서버에서 클라이언트로 객체 전송 지연시간
⑦ $TDT_{Edge\_to\_origin}$: 에지서버에서 근원서버로 객체 요구 전송 지연시간
⑧ $TDT_{origin\_to\_Edge}$: 근원서버에서 에지서버로 객체 전송 지연시간
⑨ RDT_{cache} : 캐시서버에서 객체 교체 지연시간
⑩ UT_{cache} : 캐시리스트 갱신 지연시간

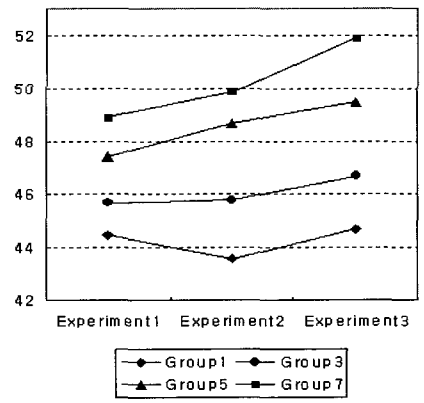
상기의 10가지 지연 요인에서는 에지서버와 캐시서버 사이의 지연은 고려하지 않은 것이다. 이것은 에지서버와 캐시서버사이의 지연은 거의 시스템 전체의 지연에 영향을 미치지 않기 때문이다.

응답시간에는 cache hit의 지연시간 LT_{ch}(Latency Time of cache hit)와 cache miss의 지연시간 LT_{cm}(Latency Time of cache miss)가 있다. 클라이언트가 요청한 객체가 캐시에 존재할 경우의 지연시간 LTch = DLT_{Edge} + ELT_{Edge} + $TDT_{client\_to\_cache}$ + DDT_{cache} + SDT_{cache} + $TDT_{Edge\_to\_client}$ + UT_{cache} 와 같다. 이것은 CDN을 사용하지 않은 콘텐트 서비스에서 발생하게되는 클라이언트의 해당 URL에 대한 객체 요청 전송지연시간과 해당 URL에서 클라이언트로 객체 전송 지연시간에 비해 지연시간을 줄일 수 있게 한다. 하지만 cache miss가 발생할 경우 지연시간 LTcm = DLT_{Edge} + ELT_{Edge} + $TDT_{client\_to\_cache}$ + DDT_{cache} + SDT_{cache} + $TDT_{Edge\_to\_origin}$ + $TDT_{origin\_to\_Edge}$ + RDT_{cache} + $TDT_{Edge\_to\_client}$ + UT_{cache} 와 같다.

지연시간이 네트워크의 물리적인 영향을 가장 많이 받으며, CDN을 통한 서비스에서는 웹 캐싱을 통해 네트워크의 영향을 줄일 수 있게 한다. 또한 캐시영역에서 암호화된 DC의 캐시영역의 비율은 제안시스템을 이용하는 서비스의 성격에 따라 달라진다. 높은 보안성 요구하는 서비스의 경우 암호화된 DC객체가 전체 객체에서 차지하는 비율이 높으므로 암호화된 DC의 캐시영역 비율이 높아질 것이다. 또한 일반적인 CDN의 경우 중요도가 높은 일부의 DC만을 암/복호화를 통해 안정성을 확보하면 되므로, 이 비율을 줄일 수 있게 된다.



(그림 6) 객체적중률(General DC Object)



(그림 7) 객체적중률(Encrypted DC Object)

(그림 6)과 (그림 7)은 객체적중률에 대한 실험 결과이며 x축은 3회에 걸친 실험을 나타내고 y축은 객체적중률을 나타낸다. 실험1은 클라이언트에서 요구하는 DC 중에 5%의 DC가 암호화된 경우에 대한 실험이다. 실험1의 그룹1은 단일영역으로 구성된 캐시에 대해 암호화된 DC와 일반 DC에 대해 구분이 없이 처리한 경우의 암호화된 객체의 적중률을 나타낸 것이고, 실험1의 그룹2는 단일영역으로 구성된 캐시에 대해 암호화된 DC와 일반 DC에 대해 구분이 없이 처리한 경우의 일반객체의 적중률을 나타낸 것이다. 실험1의 그

룹3은 1 : 9 분할 영역으로 구성된 캐시에 대해 암호화된 DC의 적중률을 나타낸 것이다. 실험1의 그룹5는 1:9 분할 영역으로 구성된 캐시에 대해 일반 DC의 적중률을 나타낸 것이다. 실험1의 그룹6은 2 : 8 분할 영역으로 구성된 캐시에 대해 암호화된 DC의 적중률을 나타낸 것이다. 실험1의 그룹7은 2 : 8 분할 영역으로 구성된 캐시에 대해 일반 DC의 적중률을 나타낸 것이다. 이와 동일한 방법으로 실험2는 클라이언트에서 요구하는 DC 중에 10%의 DC가 암호화된 경우에 대한 실험이며, 실험3은 클라이언트에서 요구하는 DC 중에 20%의 DC가 암호화된 경우에 대한 실험이다. 실험결과를 분석하면 다음과 같은 결론을 얻을 수 있다.

첫째, 단일 영역 캐싱을 CDN에 활용할 경우 암호화된 DC의 비율에 관계없이 객체적중률은 큰 변화가 없다. 따라서 이 경우 캐싱의 대체기법의 성능만이 캐싱 능력을 향상시킬 수 있다.

둘째, 분할 영역의 경우 그룹2, 4, 6과 같이 암호화된 DC의 처리를 위한 캐시영역 비율이 증가할수록 일반 DC객체의 적중률은 5%이내로 감소하게 된다. 하지만 그룹3, 4, 5의 경우를 살펴보면 암호화된 DC에 대해서 10% 정도의 객체 적중률 향상을 보인다. 또한 전체 DC 중에서 암호화된 DC의 비율이 높아질수록 성능 향상 비율이 더욱 높아짐을 알 수 있다. 따라서 CDN 상에서 발생하는 지연요소 중에서 가장 큰 비중을 차지하는 캐싱의 능력을 향상시킴으로서 SDCDS의 처리 성능을 향상시킬 수 있게 된다.

5. 결 론

본 연구에서는 DC의 전송 및 관리 기법에서 DC의 보안성을 유지하면서도 처리지연시간을 개선한 시스템인 SDCDS를 설계하였다. 실험결과 보안성을 요구하는 콘텐트의 전송이 많을수록 지연시간의 감소를 통한 성능의 향상을 기대할 수 있었다. 제안시스템의 추가적인 성능 향상을 위해서는 지연 시간 개선에 대한 추가적인 실험이 필요하다. 논문에서는 CDN의 처리지연 시간에 미치는 영향 요소 중에서 캐싱 능력이 가장 중요하며, 이에 따라 캐싱 능력의 개선을 실험을 통해 분석하였다. 이는 에지사이트가 구조적으로 트래픽의 요인을 적게 받는 환경일 때를 가정한 것이다. 하지만 에지사이트의 구조가 트래픽의 영향을 받은 경우 캐싱 능력의 향상과 트래픽의 영향을 동시에 고려하여야 한다. 따라서 이에 대한 추가적인 연구가 필요하다.

참 고 문 헌

[1] 반효경, "CDN을 위한 웹 캐싱 방법", 정보과학회, 제20권 제9호, pp.12-19, 2002.

[2] 최승락, 양철웅, 이중석, "CDN의 핵심 구성 기술들과 경향", 정보과학회, 제20권 제9호, pp.5-11, 2002.

[3] Spctral Lines, "Talking About Digital Copyright," IEEE Spectrum, Vol.38 Issue;6, pp.9, June 2001.

[4] Thorwkrth N. J., Horvatic P., Weis R., Jian zhap, "Security methods for MP3 music delivery," Signals, Systems and Computers 2000, Conference Record of the Thirty-Fourth Asilomar Conference, Vol.2, pp.1831-1835. 2000.

[5] 고일석, 임춘성, 나윤지, "ACASH: 웹 객체의 이질성과 참조특성 기반의 적응형 웹 캐싱 기법", 한국정보과학회논문지, 제31권 제3호, pp.305-313, 2004.

[6] H. Bahn, S. Noh, S. L. Min, and K. Koh, "Efficient Replacement of Nonuniform Objects in Web Caches," IEEE Computer, Vol.35, No.6, pp.65-73, June 2002.

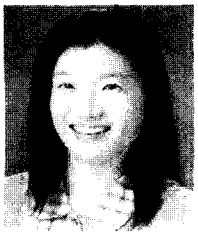
[7] L. Rizzo, L. Vicisano, "Replacement Polices for a Proxy Cache," IEEE/ACM Transaction of Networking, Vol.8, No.2, pp.158-170, 2000.

[8] R. Rivest, A. Shamir and L. Adelman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," Communications of the ACM, Vol.21, No.2, pp.120-126, 1978.

[9] ITU-T Rec. X.509, Information technology-Open Systems Interconnection - The Dictionary : Public-key and attribute certificate framework, March 2000.

[10] C. Serrao, J. Marques, T. Baker, M. Balestri, P. Kudumakis, "Protecting Digital Music Delivery and Consumption Using the OCCAMM Project Framework," Second International Conference on WEB Delivering of Music(WEDELMUSIC'02), Darmstadt, Germany, December 09-11, 2002.

[11] W. Diffie and M. E. Hellman, "New Directions in Cryptography," IEEE Transaction on information theory, Vol.1T-22, No.6, November, 1976.



나 윤 지

e-mail : yjna2967@korea.com
충북대학교 컴퓨터공학(공학박사)
미) NYIT Communication ART 전공
충북대학교 컴퓨터공학(공학석사)
경북대학교 생명공학(이학사)
현재 호남대학교 인터넷소프트웨어학과 전임강사
관심분야 : 멀티미디어, 정보보안, 지능형웹기반시스템

고 일 석

e-mail : isko@ctech.ac.kr
연세대 컴퓨터산업시스템공학(박사수료)
미) USIU 경영학과(MBA)
경북대학교 컴퓨터공학(공학석사)
경북대학교 컴퓨터공학(공학사)
현재 충북과학대학 전자상거래과 조교수
관심분야 : 지능형웹시스템, 정보보안, 게임기술, 기술평가