일회용 마스터 키 개념을 이용한 다중 방화벽 동적 통과 메커니즘 연구

박형우† · 김상완†† · 이종숙††† · 장행진††††

요 약

그리드/클라우드 컴퓨팅 플랫폼에 외부 기관의 컴퓨팅 자원이 잠시 참여하려면 관련된 모든 방화벽 마다 필터링 규칙을 추가하여야 한다. 따라서 인터넷 응용 플랫폼이 점차 그리드/클라우드 환경으로 진화될수록 다중 방화벽을 동적으로 한 번에 통과하기 위한 연구 필요성이 증대한다. 본 논문에서는 방화벽들마다 필터링 규칙을 추가/삭제하기와 같은 기존의 네트워킹 자물쇠 관리 방식에서 일회용 네트워킹 마스터 키를 사용자가 생성하는 방식으로 전환하는 새로운 다중 방화벽 동적 통과 메커니즘을 제시한다. 여기서 마스터 키는 시스템에서 자동 생성되었던 IP 주소, 포트 번호, 시퀀스 번호 등을 일종의 일회용 패스워드 형태로 네트워킹 상대방과 사전 교환을 통한 후 서로 로컬 방화벽들을 상대방에게 열어주는 키로 활용한다. 따라서 제안된 메커니즘은 단대단 사용자 사이의 방화벽의 종류와 숫자에 관계없이 동적으로 방화벽을 통과할 수 있게 한다.

주제어 : 보안, 방화벽, 동적 통과, 그리드컴퓨팅, 클라우드컴퓨팅

Study on the mechanism for the dynamic traversing of multiple firewalls using the concept of one-time master key

Hyoungwoo Park† · Sangwan Kim†† · Jongsuk Ruth. Kim††† · Haengjin Jang††††

ABSTRACT

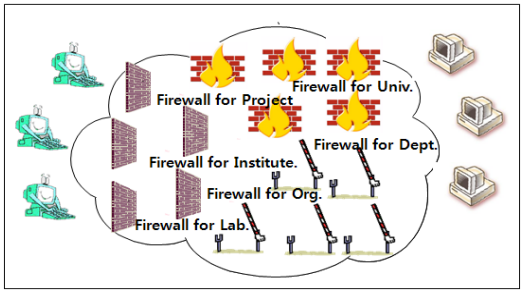
If an exterior computer wants to join the Grid/cloud computing platform for a while, all of the related firewalls' filtering rule should be immediately updated. As the platform of Internet application is gradually evolving into the Grid/Cloud environment, the R&D requirement for the dynamic traversing of the multiple firewalls by a single try is also increasing. In this paper, we introduce the new mechanism for the dynamic traversing of the multiple firewalls using the concept of the one-time master key that can dynamically unlock the tiers of firewalls simultaneously instead of the existed filtering rule based method like a lock management at each firewall. The proposed master keys are like one-time password, consisted of IP addresses, port numbers, and TCP's initial sequence numbers, and generated by end users not administrators. They're exchanged mutually in advance and used to make a hole at local-side firewalls for the other's packet incoming. Therefore, the proposed mechanism can function regardless of the number or type of firewalls.

Keywords : Security, Firewall, Dynamic traversing, Grid computing, Cloud computing

† 정 회 원: KISTI 책임연구원
†† 정 회 원: KISTI 선임연구원
††† 정 회 원: KISTI 책임연구원(교신저자)
†††† 정 회 원: KISTI 책임연구원(글로벌데이터 허브센터장)
논문접수: 2010년 9월 16일, 심사완료: 2010년 9월 20일
* 본 논문은 2010년 교육과학기술부의 대용량 실험 데이터 센터 구축 사업의 지원으로 수행되었음.

1. 서 론

전 세계적으로 15억 명 이상이 인터넷을 사용하면서 구글과 같이 유명한 응용 사이트의 사용자 유동성은 매우 높다. 유동성이 높은 서비스를 위하여 사용자의 수에 비례하여 서버를 동적으로 할당하는 효율적인 리소스 관리 기술들이 지속적으로 연구되고 있다. 대표적인 기술로는 그리드컴퓨팅[1][2],클라우드컴퓨팅[3][4]등이 있다. 이 기술들은 사용자의 요구에 따라서 지리적으로 분산된 컴퓨팅 자원들을 모아서 서비스하기 위하여 연구개발되고 있다. 그러나, 인터넷 보안의 중요성이 강조되면서 방화벽이 기관 단위, 부서 단위, 또는 응용 서버 단위로 구축되면서 서로 다른 보안 관리 도메인 내에 분산되어 있는 컴퓨팅 자원들의 활용이 어려워지고 있다. 현재는, 서로 다른 기관의 컴퓨팅 자원을 공동 활용하려면 많은 수의 방화벽의 포트를 항상 열어주어야 하는 데, 이는 또 다른 보안 취약성을 불러일으킨다. 그림 1은 다중 방화벽 모습을 보여준다



<그림 1> 다중 방화벽 모습

이러한 다중 방화벽 환경에서 그리드/클라우드 컴퓨팅 플랫폼에 외부의 컴퓨팅 자원을 잠시 활용하려면 다중 방화벽을 동적으로 통과할 수 있어야하고 그리고 통과 방법도 간단해져야 한다. 현재는 그리드/클라우드 컴퓨팅 자원들의 중간에 있는, 즉 통과하여야 할 방화벽 관리자들의 도움을 일일이 받아서 모든 방화벽 마다 관련 필터링 규칙을 업데이트 하여야 하는데 이를 실시간으로 처리하기가 쉽지 않다. 왜냐하면 방화벽의 종류가 다양하고 새로운 분산 컴퓨팅 기술들의 응용 또한 다양하기 때문에 이를 위하여 모든 방화벽 마다 관련 필터링 규칙을 추가하는 작업은 쉽지 않다. 방화벽 관리가 소프트웨어로 자동화되고 있지만 방화벽 관리 소프트웨어[5]가 여러 종류이고 방화벽 관리자를 위한 소프트웨어이기 때문에 사용자 측면에서 보면 다중 방화벽 동적 통과 환경이 개선되었다고 보기에는 미흡하다.

본 논문에서는 방화벽에서 네트워킹 자물쇠를 붙였다 뗐다하는 것과 같이 방화벽들마다 필터링 규칙을 추가/삭제하는 기존 방식에서 모든 방화벽들을 한 번에 열 수 있는 일회용 네트워킹 마스터 키를 사용자가 생성하여 사용할 수 있는 방식으로 전환하는 새로운 다중 방화벽 동적 통과 메커니즘을 제시한다. 여기서 마스터 키는 IP 주소, 포트 번호, 시퀀스 번호 등 TCP 네트워킹에 필요한 번호들로 구성된 일종의 일회용 패스워드와 같은 형태이다. 이들 번호들은 기존에는 시스템이 생성한 번호들이지만 본 논문에서는 외부 인증서버에서 이들 번호들을 발급할 수 있게 하여 방화벽 동적 통과를 원하는 사용자와 시스템 관리자가 동적 통과하고자 할 때 네트워킹 상대방과 사전 교환을 통하여 미리 알게 된 IP 주소, 포트번호 그리고 TCP 시컨스 번호로 클라이언트/서버의 IP 주소, 포트번호 그리고 시퀀스 번호를 매핑시킴으로써 서로 로컬 방화벽들을 상대방에게 열어주는 마스터 키로 활용케한다. 따라서 제안된 메커니즘은 단대단 사용자 사이의 방화벽의 종류와 숫자에 관계없이 동적으로 다중 방화벽들을 통과할 수 있어서 그리드/클라우드 컴퓨팅 플랫폼 참여 자원의 범위를 동적으로 관리 도메인 밖까지 쉽게 확장할 수 있다.

2. 관련연구

방화벽의 동적 통과를 위한 기술들의 접근 방식은 크게 3가지 방식으로 대별된다. 하나는 동적 통과 요청이 있을 때마다 방화벽의 규칙을 동적으로 자동 생성하는 방법을 연구하는 것이고 두 번째는 방화벽 안쪽에서 외부로 향하는 연결 요청은 방화벽이 기본적으로 통과시켜주는 방화벽의 특징을 이용하는 연구이다. 마지막 세번째는 방화벽이 사용자와 서버간의 인증을 중계하여 주는 것이다. 첫 번째 방식은 다중 방화벽 환경에서는 비효율적이다. 왜냐하면 여러 개의 방화벽을 동적 통과하기위해서는 매번 동적 통과 할 때마

다 방화벽 하나하나 통과 요청을 하기 때문이다 그리고 방화벽이 바뀌면 사용자는 이용법을 새롭게 학습하여야 하는 문제가 있다. 두 번째 방식은 방화벽 마다 외부에 중개자를 두어서 내부에서 외부로 연결 요청한 것처럼 중개하여야 하기 때문에 통과하여야 할 방화벽이 늘어날수록 연결하여야 할 중개자의 수도 늘어나는 문제가 있다. 그리고 최근의 방화벽들이 TCP의 시퀀스 번호와 three-way hand shaking 상태를 체크하면서 스카이프와 같은 UDP 응용으로만 활용이 제한받는 문제를 갖고있다. 세번째 방식은 다중 방화벽 환경에서는 방화벽수 증가로 방화벽에서의 인증 중계가 더욱 더 복잡해진다. 표 1은 기존 연구사례를 보여준다.

<표 1> 연구 사례

구분	사례
방화벽의 필터링 규칙을 동적으로 자동 추가하기 연구 그룹	· Dyna-Fire · CODO · DyNeF
방화벽 내부에서 연결을 시도한 것처럼 중개하여주기 연구 그룹	· GCB · UDP 홀 펀칭 · 게이트웨이/프록시
방화벽에서 사용자와 서버의 인증 중계하기	· TCP-AuthN

첫 번째의 연구방법인 방화벽의 필터링 규칙을 동적으로 추가하기에는 Dyna-Fire[6], CODO[7], DyNeF[8] 등이 있다.

Dyna-fire는 Portknocking을 이용하여 방화벽에게 통과 신청을 하는 시그널링 프로토콜을 사용한다. Portknocking은 일종의 knock sequence로 신청자의 IP 주소, 요구 자원의 이름 또는 포트 번호, 포트 개방 시간 그리고 사용자 식별자 등으로 구성된다. 외부에서의 연결 시도를 모니터링하다가 적법한 knock sequence를 발견하면 Dyna-fire는 새로운 필터링 규칙을 방화벽의 iptable에 추가한다. 그리고 iptable과 knock sequence의 목적지 호스트의 IP 주소와 포트 번호가 일치하면 knock sequence에 있는 요청 시간에 따른 허용된 시간 동안 연결을 허락한다. 초기에 열려진 포트는 없고 모든 사용자와 자원들에 대한 정보를 보유하는 데이터베이스가 필요하다

CODO는 Cooperative On-Demand Opening의 약자이며 방화벽에 CODO 에이전트를 설치한 후 방화벽에 포트 3개를 시그널링 메세지를 듣기 위하여 열어 놓는다. 하나는 내부 응용이 자신의 가용한 목적지를 등록하는 데에 사용하고, 다른 하나는 외부 응용이 내부 목적지에 연결을 요청하는 데에 사용한다. 제 3의 포트는 내부 응용이 외부 목적지와 연결을 요청할 때 사용한다. 모든 시그널링 메세지는 SSL을 사용한다. X.509 디지털 인증를 이용하여 상호 인증한다. 외부에서 요청한 내부의 목적지가 이미 메모리 테이블에 등록이 되어있고 모든 콘트롤들이 통과되었을 때에 CODO 에이전트가 새로운 필터링 규칙을 방화벽의 iptable에 추가한다.

DyNeF는 사용자로 부터 직접 요청을 받는 것이 아니고 사용자가 E-CAS (Extended Community Authorization Service)에 등록을 하면 방화벽 에이전트가 E-CAS로부터 수신된 정책에 따라서 동적 방화벽 구성을 실행한다.

두 번째 연구 방법인 외부 연결 요청을 방화벽 내부에서 먼저 연결을 시도한 것처럼 중개하여주는 기술 연구로서는 GCB[9], UDP 홀 펀칭[10], 게이트웨이/프록시[11]가 있다.

GCB는 Generic Connection Brokering의 약자로 버클리 소켓 시스템에서 클라이언트가 서버에게 연결 요청을 하지만 통상 서버가 방화벽 뒤에 있어서 서버는 외부 연결을 통상 수신할 수 없게 된다. GCB는 응용 소프트웨어와 버클리 소켓 사이에 중간 계층에 구현되며 응용이 알지 못하게 네트워크의 연결 방향을 바꾼다. 서버가 클라이언트에게 연결을 먼저 요청하는 것처럼 방화벽을 알게 한다. 이를 위해서 GCB와 방화벽안의 서버가 사전에 연결되어있게 한다. GCB를 방화벽 밖에 또한 두어서 클라이언트가 서버에 연결하고자 할 때 먼저 외부 GCB에게 방화벽 내부의 서버와 연결하고 싶다고 한다. 그러면 외부 GCB가 클라이언트에게 서버가 클라이언트에게 연결을 시도할 것을 알려준다. GCB는 서버에게 client에게 연결할 것을 요청한다. 서버는 직접 client에게 연결한다.

UDP 홀 펀칭은 방화벽이 통상적으로 외부 연결은 허용하는 것을 이용한 기술 연구이다. 클라이언트 A가 외부의 중계서버에 TCP 연결을 설정

한 후 클라이언트 B와 특정의 동적 UDP 포트로 연결하고 싶다는 정보를 중계서버에게 제공한다. 중계서버는 클라이언트 B에게 클라이언트 A의 IP 주소와 특정 동적 UDP 포트 정보를 알려준다. 클라이언트 B는 중계 서버에서 알려준 클라이언트 A의 IP 주소와 포트로 UDP 데이터그램을 보내면서 중계서버에게 자기가 만든 UDP 포트 정보를 보낸다. 중계 서버가 클라이언트 B의 IP 주소와 UDP 포트 정보를 클라이언트 A에게 전한다. 클라이언트 A 와 클라이언트 B 이를 이용하여 UDP 통신을 한다. UDP 데이터그램을 외부로 보낼 때 로컬 방화벽은 연결정보와 동적 액세스 규칙을 저장하여서 저장된 규칙에따라 통과된 데이터그램의 응답 데이터그램이 방화벽을 통과하는 것을 허용한다.

게이트웨이/프록시는 우리가 제일 많이 아는 방화벽 통과 방식으로 미리 타겟 응용 또는 서비스가 설정된 응용 계층의 게이트웨이나 프록시의 연결 정보를 사용자들이 알 수 있게 공지하면 사용자가 응용계층의 게이트웨이 및 프록시 연결 및 인증(로그인) 과정을 거쳐서 해당 서비스를 제공 받는다

세번째 방식인 TCP-AuthN[12]은 TCP three-way handshaking을 할 때 방화벽에서 TCP 헤더의 인증 필드를 사용하여 사용자가 보낸 X.509 인증서를 서버가 확인하는 것을 중계하여 주면서 적법한 사용자인지를 판단한다. 사용자와 서버 사이의 방화벽이 증가할수록 인증 절차가 복잡해질 수 있다

3. 제안 메커니즘 및 시스템 구조

3.1 제안 메커니즘

인터넷에서 보안의 중요성이 강조되면서 대부분의 대학이나 연구소에서는 보안 가능이 강화된 stateful firewall을 구축한다. stateful firewall에서는 송수신 IP 주소와 포트 번호 외에도 TCP 응용의 경우 송수신 시퀀스 번호와 TCP three-way hand shaking의 상태 변화도 체크하므로 다중 방화벽을 통과하기위해서는 현재로서는 방화벽 담당자들과 사전에 협의하여야 한다. 그리고 방화벽의 동적 통과를 위한 방화벽 필터링 규칙의 실시간 수정은 쉽지 않은 것이 또한 현실이다. 본 논문에서는 다중 방화벽 환경에서 동적 통과를 위한 사용자와 방화벽 관리자의 요구 사항들을 동시에 충족시키기 위하여 아래와 같은 기준을 충족시키는 메커니즘을 연구하였다.

① 공인을 받은 연결 방식이어야 하고 내부 보안 관리자와 외부 공인 인증기관의 실시간 모니터링을 지원할 수 있거나 사후 승인을 받을 수 있는 방법이어야 한다.

② 인터넷 응용 플랫폼이 그리드/클라우드 환경으로 진화되어도 end-to-end간 인터네트워킹은 양종단의 사용자 주도로 이루어져야 한다. 클라이언트 서버 모델에서 사용자가 수동적이고 단순한 기능을 수행하는 모습에서 능동적이고 지능적인 기능을 수행하는 모습으로 바뀌어야 한다.

③ 양쪽 단 사이에 위치한 방화벽의 수와 관계없이 일회의 시도로 연결설정이 이루어져야 한다.

④ 다중 방화벽 동적 통과 서비스를 수시로 이용하여도 보안 환경이 낮아지지 않게 일회 단위로 서비스되어야 한다.

⑤ 기존 애플리케이션을 전혀 수정할 필요가 없어야 한다.

상기와 같은 조건을 만족하는 방화벽 동적 통과 메커니즘은 결국은 미래 인터넷이나 미래 인터네트워킹 연구 측면과도 관계될 수 있다고 본다. 상기 조건을 고려하여 본 논문에서는 사용자, 응용 서버 담당자, 방화벽 관리자 그리고 마스터 키 관리 담당자가 참여하는 즉 4자가 서로 인증 및 확인할 수 있는 일회용 마스터 키를 사용하여 방화벽들을 동적으로 통과하는 메커니즘을 제안한다. 기본 개념은 송수자의 IP 주소, 포트 넘버, TCP 시퀀스번호, 그리고 연결 요청 패킷 발송 시간을 서로 미리 알게 한 후 TCP three-way hand shaking 프로토콜에 맞추어 연결을 하는 것처럼 첫 번째 연결 요청 패킷과 첫 번째 응답 패킷 두 패킷을 시간에 맞추어 보내는 것이다. 본 제안 메커니즘의 기본 개념이 기존의 두 번째 연구 방식인 외부에서의 연결 요청을 내부에서 외부로 연결 요청하는 것처럼 연결 방향을 전환시켜주는 방식과 유사하게 보이지만 전혀 다른 방식이다. 제안 메커니즘은 두 번째 방식에서 채택한 외부에서의 연결 요청을 내부에서의 연결 요

청으로 전환시켜주는 연결 중개자를 두지 않는다. 제안 메커니즘은 TCP 고유의 연결 프로토콜을 다중 방화벽 환경에서도 정상적으로 동작할 수 있도록 관련 정보의 공유에 기반을 둔다. 따라서 TCP는 물론 UDP적용에도 문제가 없다. 두 번째 연구 방식은 최근의 방화벽들이 기존의 stateless 에서 stateful 체크 기능을 지원하기 때문에, 즉 TCP의 시퀀스 번호와 three-way hand shaking 상태를 순서적으로 체크하기 때문에 TCP 응용에서는 적용할 수 없는 방식이다. 따라서 현재 두 번째 연구방식은 스카이프와 같은 stateless 응용 즉 UDP 응용에만 활용되고 있다.

본문에서 제안한 마스터 키는 사용자, 서버 관리자, 방화벽 담당자, 마스터키 생성 인증 담당자 4자간 합의에 의한 약속을 방화벽 동적 통과를 위한 trust의 기반으로 삼았기 때문에 방화벽 관리자의 선행 조치를 요청하지 않고도 방화벽을 동적 통과할 수 있다. 그러나 방화벽 담당자가 트래픽 모니터링에서 해당 트래픽이 인가된 트래픽인지 분석하는 것과 사용자 또는 서버 관리자에게 마스터 키 기반 방화벽 동적 통과를 수행하였는지에 대한 확인하는 일은 실시간으로 지원한다. 마스터 키의 형태는 그림 2와 같다.

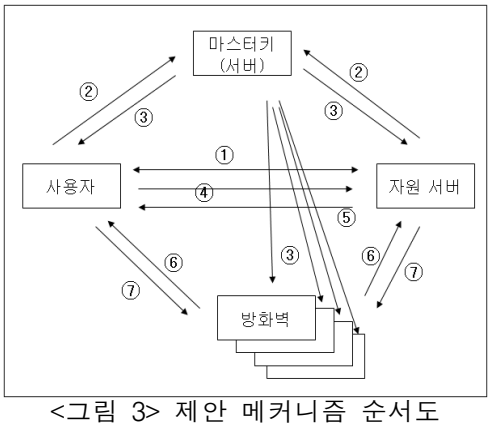
16비트	16비트
실제 소스 IP 주소	
실제 목적지 IP 주소	
설정 소스 IP 주소	
설정 목적지 IP 주소	
실제 소스 포트 번호	설정 소스 포트번호
실제 목적지 포트 번호	설정 목적지 포트번호
설정 소스 시퀀스 초기 번호	
설정 목적지 시퀀스 초기 번호	
설정 소스 전송시간-시간대 월 일 시간	
설정 소스전송시간- 분 초 밀리초	
설정 목적지 전송시간-시간대 월 일 시간	
설정 목적지 전송시간-분 초 밀리초	
사용자 1 ID	
사용자 2 ID	
사용자 3 ID (추가 가능)	

<그림 2> 마스터 키 형태

사용자와 서버 담당자는 자신이 사용하는 시스템의 환경에 맞추어서 마스터 키관리 담당자에게 이용 가능한 IP 주소 범위, 포트 번호 범위를 알려주어 방화벽 동적 통과를 위한 마스터 키 요청을 할 때 해당 범위 내에서 자동 생성하여 주도록 한다. 그리고 관련 정보는 마스터 키 관리 DB에 등록한다. 그림 3은 본 논문에서 제안한 메커니즘의 흐름도이다. 다중 방화벽 통과 메커니즘을 간략히 설명하면 다음과 같다.

① 임의의 시간에 사용자와 자원 서버 담당자는 사용자가 서버에 액세스하는 것을 합의한다.

② 합의된 이용 시작 시간과, 사용 서버에 대한 네트워킹 관련 정보를 마스터 키 서버에 보낸 후 해당 서버에 액세스하기위한 마스터 키를 할당하여 줄 것을 마스터키 관리자에게 요청한다.



<그림 3> 제안 메커니즘 순서도

③ 마스터 키 관리자는 사용자와 서버 담당자가 사전에 등록한 IP주소 범위, 포트번호 범위 내에서 랜덤 생성한 IP 주소와 포트번호가 포함된 마스터키를 생성한 후 사용자, 서버 담당자, 그리고 관련된 방화벽 담당자들 모두에게 생성된 마스터키를 보내준다.

④ 사용자와 서버 담당자는 수신된 마스터 키의 값을 이용하여 TCP의 초기 시퀀스 번호를 설정하는 등 네트워크 연결 환경을 구축한다. 그리고 사용자는 SYN 플래그가 설정된 첫 번째 TCP 연결 요청 패킷을 미리 알고 있는 서버의 IP주소와 포트 번호로 보낸다. TCP의 경우 SYN 플래그가 설정되어 있는 경우 로컬 방화벽은 내부에서 외부로 나가는 트래픽이기 때문에 통과할 수 있다. 하지만 상대방 방화벽에 도착해서는 상대방 방화벽에는 사전 작업이 되어 있지 않기 때문에 drop 된다. 그러나 사용자가 보낸 TCP 연결 요청 패킷은 로컬 방화벽들을 지나면서 로컬 방화벽들이 사용자가 보낸 연결 요청 패킷에 대응된 서버 쪽의 응답 패킷이 도착하기를 기다리는 상태로 만든다.

④ 서버에서는 클라이언트가 보낸 연결 패킷의 IP 주소, 포트 번호, TCP 시퀀스 번호, 그리고 보낸 시간을 이미 알고 있기 때문에 사용자가 보낸 패킷을 수신하지 않아도 서버에 있는 클라이언트 시뮬레이터를 통하여 클라이언트가 보낸 연결 요청 패킷을 내부에서 대신 서버로 보내주게 한다. 클라이언트 시뮬레이터는 단지 연결 초기 패킷 한 종류만 생성하는 시스템이며 패킷 수신은 할 수 없는 상태로 설정한다.

⑤ 서버는 정상적 패킷을 수신한 것으로 인식하여 클라이언트 쪽의 로컬 방화벽들이 기다리고 있는 첫 번째 응답 패킷을 생성하여 보낼 수 있다. 즉 SYN와 ACK 플래그가 설정되고 수신이 예정되었던 패킷의 시퀀스 번호에 대응된 ACK 번호를 가진 패킷을 생성하여 보낼 수 있다. 따라서 첫 번째 연결 요청 패킷과 첫 번째 응답 패킷으로 송수신자 사이의 방화벽들은 기다리고 있는 연결 상태의 값을 충족시킬 수 있기 때문에 정상적인 연결이 설정된다.

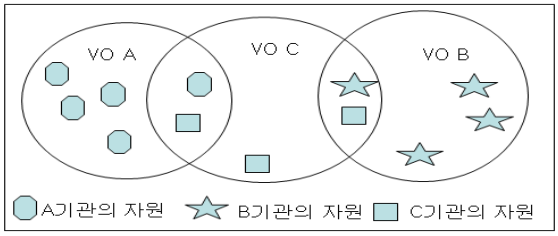
⑥ 방화벽 관리자들은 방화벽 통과 트래픽 분석 또는 보안 모니터링 과정을 통하여 사용자 또는 서버 관리자가 마스터 키를 사용하고 있는 지 또는 사용하였는 지를 확인한다.

⑦ 사용자(또는 서버 관리자)는 방화벽 관리자에게 마스터 키를 이용한 것 (또는 하고 있는 것)을 확인하여 준다.

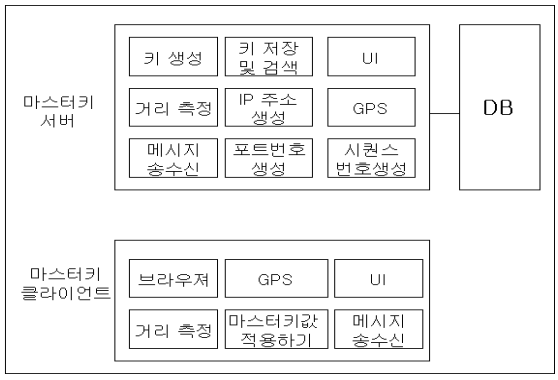
기존에 개별 시스템들이 생성하였던 IP 주소, 포트 번호, 그리고 TCP 초기 시퀀스 번호를 개별 시스템이 아닌 신뢰성이 높은 제 3자의 인증 시스템을 통하여 사용자가 직접 생성하고 제어할 수 있게 함으로써 비도를 높이면서도 다중 방화벽을 한번에 동적 통과할 수 있는 방법을 제시한다.

3.2 시스템 구조

제안 메커니즘을 위한 시스템의 구조는 기본적으로 그림 3과같이 그리드/클라우드 컴퓨팅을 위한 가상 조직(VO: Virtual Organization) 환경에서 서버 자원을 동적으로 공동 활용 환경을 구성하는 것을 가능하도록 하는 데에 목적을 두었다. 그림 4에서 마스터키 서버는 마스터키 관리자의 시스템에 구축되고 다른 사용자, 응용 서버 담당자, 그리고 방화벽 담당자의 시스템에는 마스터키 클라이언트를 구축한다. 마스터키 서버 시스템과 마스터키 클라이언트 시스템은 GPS를 이용하여 시간을 동기화 시킨다. 사용자와 응용 서버 담당자는 기종의 인터넷 도구들을 이용하여 서로 간의 시간 거리(RTT: Round Trip Time)를 측정하고 측정 시간을 마스터키 서버에 연결된 DB에 등록한다. 방화벽 담당자 정보도 DB에 등록한다.



<그림 3> 가상 조직 기반 자원 공유

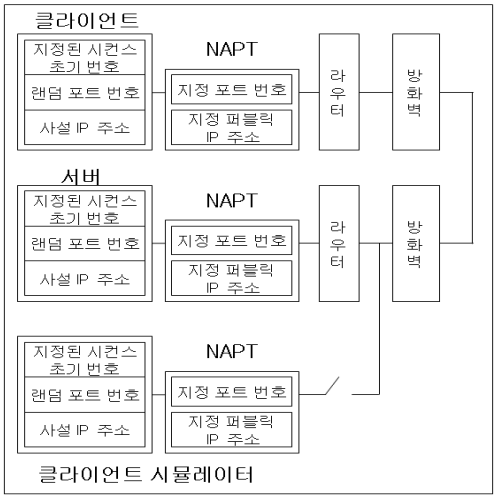


<그림 4> 시스템 구조도

4. 실 험

본 논문에서 제안한 메커니즘의 가능성을 확인하기위한 primitive한 실험을 하였다. 먼저 IP 주소와 포트 넘버가 고정 값을 갖게 하고 상대방에게 미리 통보하기위하여 그림 5와 같이 NAPT (Network Address and Port Translation)[13],[14]를 이용하였다. 양 쪽에 NAPT를 설치하여 사전에 협의된 IP 주소와 포트 넘버를 설정하였다. 왜냐하면 포트 번호는 일부 well-known 포트를 제외하고는 대부분이 시스템에서 임의의 숫자로 생성하기 때문이다. 따라서 실제 연결을 시도할 때는 알 수 없는 클라이언트의 송신 포트 넘버를 미리 약속한 번호로 만들어 보낼 수 있다. 서버 측의 포트넘버는 well- known port number를 이용할 수 있지만 포트 번호를 외부에 항상 열어두

는 것을 방화벽 담당자들이 선호하지 않는다. 초기 TCP 시퀀스 번호의 수정은 TCP 소스에서 랜덤 번호 생성 대신 정해진 초기 번호를 이용하도록 한다. 연결 시간은 cron 테이블을 이용해서 송수신 패킷 전송 시간을 맞췄다. 서버 측에 있는 클라이언트 시뮬레이터는 서버에서의 초기 응답 패킷의 수신 응답을 못하게 한 후 초기 연결 요청 Telnet 패킷만 1-2회 전송하고 네트워크 연결을 끊게 하였다.



<그림 5> 실험 구성도

5. 결 론

본 연구는 기존의 클라이언트/서버 기반 인터네트워킹 패러다임에서 그리드/클라우드 컴퓨팅이 보편화가 될 수 있는 미래 분산 컴퓨팅 환경을 위한 새로운 인터네트워킹 패러다임은 어떠한 모습이어야 하는 가를 보여줄 수 있는 연구 방향의 하나로 예상한다. 대부분의 컴퓨팅 자원들은 보안 문제로 몇 중의 방화벽 안에 위치해 있기를 강요받기도 하면서 동시에 이들 컴퓨팅 자원들은 유휴 시간에는 그리드/클라우드에 참여 하도록 요청도 받는다.

따라서 다중 방화벽 환경에서 자원을 가진 사용자들이 그리드/클라우드 컴퓨팅 인프라를 동적으로 참여하기 위해서는 기존의 인터네트워킹과 다른 방식의 도입이 필요하다. 즉 미래 인터네트워킹을 위하여 다중 방화벽 통과가 고려된 인터네트워킹 방식이 연구되어야 한다고 생각한다.

본 연구를 시작하게 된 동기는 그리드/클라우드 컴퓨팅 기술 연구를 국내 또는 해외의 외부 기관 파트너 연구자와 실험을 할 때 한 달 전부터 방화벽 관리자에게 방화벽에서 포트를 일부를 열어 달라고 요청하여야 하였다. 국내 연구자들의 의견을 들어보면 자신이 조금 수고해서 자신이 연구하고 싶을 때 실험을 할 수 있으면 약간의 수고는 얼마든지 할 수 있다고 하였다. 자신의 실험을 위해서 특히 해외랑 연구를 할 때 다른 기관의 연구원을 같이 밤새우게 하는 것이 제일 미안하다고 하였다. 이런 분들에게 조금이라도 도움을 주는 연구를 지속적으로 하려 한다.

참 고 문 헌

[1] Foster, I., Kesselman, C., Nick, J. M., &Tuecke, S. (2002). Grid services for distributed system integration, *IEEE Computer 35 (6)*, 37-46.

[2] Foster, I., &Kesselman, C. (1999). **The Grid: Blueprint for a New Computing Infrastructure.** Morgan- Kaufman.

[3] http://en.wikipedia.org/wiki/Cloud_computing

[4] http://download.boulder.ibm.com/ibmdl/pub/software/dw/wes/hipods/**Cloud_computing_wp_final_8Oct.pdf**

[5] dynfw **Dynamic Firewall Tools**, v1.0, http://www.gentoo.org/doc/en/articles/files/dynfw-1.0.1.tar.bz2.

[6] Green M. L., Gallo S. M., &Miller R. (2004). Grid-enabled virtual organization based dynamic firewall. *Proceedings. fifth IEEE/ACM International Workshop on Grid Computing*, 208-216

[7] Son S., Allcock B., &Livny M., (2005). CODO: Firewall Traversal by Cooperative On-Demand Opening, *14th IEEE Symposium on HPDC14.*

[8] Subramanian. N, Usha. R. E., &Bravi R. E. (2008). DyNeF: Host-Privilege-Based Dynamic Network Firewall for Grid Environment. *Proceedings of WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY Vol. 32*. 653-657

[9] http://www.cs.wisc.edu/condor/**gcb**/

[10] Gruenter, E., Meier M., Miederberger, R., &Petri, F. (2006). Dynamic Configuration of Firewalls Using UDP Hole Punching.

http://www.D-grid.de/fileadmin/user_upload/documents/DGI-FG3-5/**Dynamic_Configuration_ of_Firewalls.pdf**
[11] http://en.wikipedia.org/wiki/Proxy_server
[12] Wiebelitz, J., Kunz, C., Piger, S., &Grimm, C. (2009). TCP-AuthN: TCP Inline Authentication to Enhance Network Security in Grid Environments, *Proc. of ISPDC '09*
[13] Rosenberg, J. (2010). Interactive Connectivity (ICE): A protocol for network Address Translator (NAT) Traversal for Offer/ Answer Protocols. **RFC5245. (draft-ietf-mmusic-ice).**
[14] Rosenberg, J. (2010). Traversal Using Relays around NAT (TURN) Extensions for TCP allocations. **draft-ietf-behave-turn-tcp-07. txt**

이 종 숙

2001 Univ. of Canterbury (New Zealand) 컴퓨터공학(박사)
1992 ~ 1993 한국전자통신연구원 연구원
1999 ~ 2002 Univ. of Canterbury 연구원
2002 ~ 현재 한국과학기술정보연구원, 슈퍼컴퓨팅본부 책임연구원
2004 ~ 현재 인터넷정보학회 논문지 편집위원, 과학기술연합대학원대학교(UST) 그리드/슈퍼컴퓨팅전공부문, 부교수(겸임)
관심분야: 컴퓨터 시뮬레이션 기반 사이버 러닝, 그리드 및 분산 컴퓨팅, 컴퓨터 네트워크 및 트래픽 모델링
e-Mail: jsruthlee@kisti.re.kr

박 형 우

2000 성균관대학교 전기전자 컴퓨터공학과(박사)
1997 ~ 1999 한국전자통신연구원 (ETRI) 전산망 개발실장
2000 ~ 2005 한국과학기술정보연구원(KISTI) 그리드연구실장, 책임연구원
2006 ~ 현재 KISTI 슈퍼컴퓨팅본부 글로벌데이터허브센터 책임연구원
관심분야: 네트워크 그리드, 네트워크 가상화, 오버레이 네트워크
e-Mail: hwpark@kisti.re.kr

김 상 완

2001 포항공과대학교 컴퓨터공학과(석사)
2001 ~ 현재 한국과학기술정보연구원 슈퍼컴퓨팅본부 (선임연구원)
관심분야: 그리드 컴퓨팅, 클라우드 컴퓨팅
e-Mail: sangwan@kisti.re.kr

장 행 진

2005 전북대학교 컴퓨터공학박사
2002 ~ 2006 한국과학기술정보연구원 책임연구원, 국가그리드사업 팀장
2007 ~ 2008 정보통신 표준화협회, 국제 IT 표준화 위원
2008 ~ 그리드 비즈니스 협회 부의장
2008 한국과학기술정보연구원 사이버인프라 사업팀장
2009 한국과학기술정보연구원 대용량데이터센터구축팀장
2010 ~ 현재 한국과학기술정보연구원/슈퍼컴퓨팅본부/글로벌데이터 허브센터장
관심분야: 데이터 그리드, 슈퍼컴퓨팅, 분산 컴퓨팅, 그리드 미들웨어
e-Mail: hjjang@kisti.re.kr