성형 VPN 구조에서의 주문형 터널 생성 메커니즘
(On-Demand Tunnel Creation Mechanism in Star VPN Topology)

변 해 선 † 이 미 정 ††
(Haesun Byun) (Meejeong Lee)

요 약 성형(Star) VPN(Virtual Private Network) 구조에서는 통신하는 두 CPE(Customer Premise Equipment) VPN GW(Gateway) 간 발생하는 트래픽이 항상 Center VPN GW를 거쳐서 전송되므로 비효율적인 트래픽 전송이 이루어진다. 또한 Center VPN GW에서의 패킷 프로세싱으로 인한 과부하도 발생한다. 이를 해결하기 방안으로 IPSec(IP Security)의 IKE(Internet Key Exchange) 메커니즘을 이용하여 통신하는 두 CPE VPN GW 간 직접터널을 설립할 수 있으나 이 경우에는 터널 설립 및 관리가 복잡하고 오버헤드가 크다. 이에 본 논문에서는 통신하는 CPE VPN GW 간에 자동적으로 직접터널을 설립할 수 있게 하는 SVOT(Star VPN On-demand Tunnel) 방안을 제안한다. SVOT 방안에서는 CPE VPN GW가 트래픽 모니터링 정보를 기반으로 직접터널을 설립할 것인지를 판단한다. CPE VPN GW는 Center VPN GW로부터 터널 설립에 필요한 제반정보들을 제공받아 상대 CPE VPN GW와 직접터널을 설립한다. 시뮬레이션을 통해 제안하는 방안에 대하여 성능을 조사하였고, 이와 함께 기본적으로 Center VPN GW를 통하여 모든 트래픽이 전송되는 성형 VPN 구조, 모든 CPE VPN GW간 풀-메시(Full-mesh)로 터널 연결 정보를 유지하고 있는 풀-메시 VPN 구조와 성능을 비교하였다. 시뮬레이션 결과, 제안하는 SVOT 방안이 기본적인 성형 VPN 구조에 비해 확장성과 트래픽 전송효율성, Center VPN GW의 과부하를 방지하는 측면에서 우수한 성능을 보이면서 종단간 지연 및 처리율에 있어서는 풀-메시 VPN 구조와 거의 비슷한 성능을 보임을 확인할 수 있었다.

키워드 : VPN(가상사설망), 터널 생성, 직접터널

Abstract In the star VPN (Virtual Private Network) topology, the traffic between the communicating two CPE(Customer Premise Equipment) VPN GW(Gateway)s may be inefficiently transferred. Also, the Center VPN GW may experience the overload due to excessive packet processing overhead. As a solution to this problem, a direct tunnel can be established between the communicating two CPE VPN GWs using the IKE (Internet Key Exchange) mechanism of IPSec(IP Security). In this case, however, the tunnel establishment and management may be complicated. In this paper, we propose a mechanism called 'SVOT (Star VPN On-demand Tunnel)', which automatically establishes a direct tunnel between the communicating CPE VPN GWs based on demand. In the SVOT scheme, CPE VPN GWs determine whether it will establish a direct tunnel or not depending on the traffic information monitored. CPE VPN GW requests the information that is necessary to establishes a direct tunnel to the Center VPN GW. Through a simulation, we investigate the performance of the scheme performs better than the SVST scheme with respect to scalability, traffic efficiency and overhead of Center VPN GW, while it shows similar performance to the FVST with respect to end-to-end delay and throughput.

Key words : VPN, tunnel establishment, direct tunnel

1. 서 론

인터넷을 기반으로 하는 기업 활동이 증대됨에 따라 제한된 LAN(Local Area Network)의 구성에서 벗어나 멀리 떨어진 본·지점 간의 네트워크 연결을 위해 VPN(Virtual Private Network)을 구축하는 기업이 늘어나고 있다. VPN은 인터넷과 같은 공중망을 이용하여 구성되는 가상 사설망으로 특정 그룹의 사용자에게 보

· 본 연구는 대학 IT연구센터 육성·지원사업의 연구결과로 수행되었음
† 학생회원 : 이화여자대학교 컴퓨터학과
ladybhs@ewhain.net
†† 정 회 원 : 이화여자대학교 컴퓨터학과 교수
lmj@ewha.ac.kr
논문접수 : 2004년 7월 15일
심사완료 : 2005년 4월 1일

안 및 서비스 품질 지원을 제공하는 것을 목적으로 한다[1].

현재 운용중인 대부분의 VPN은 모든 CPE(Customer Premise Equipment) VPN GW(Gateway)들이 Center VPN GW와 논리적인 연결을 유지하고 있는 성형(Star) VPN 구조를 취하고 있다. 성형 VPN 구조에서는 각각의 CPE VPN GW와 Center VPN GW간 직접 터널이 설립되어 있으며 이들 간에는 터널링을 이용하여 패킷을 전달한다. CPE VPN GW는 고객의 사이트로부터 패킷을 받으면 외부(Outer) 헤더의 목적지 주소를 Center VPN GW의 주소로 하여 패킷을 인캡슐레이션(Encapsulation)한 후 미리 설립되어 있는 터널을 이용하여 전송한다. Center VPN GW는 CPE VPN GW로부터 받은 패킷을 디캡슐레이션(Decapsulation) 한 후 내부(Inner) 헤더의 목적지 CPE VPN GW의 주소를 알아내고 다시 그 주소를 외부 헤더의 목적지로 하여 패킷을 인캡슐레이션하여 전송한다. 이러한 성형 VPN 구조는 새로운 사이트를 추가하고자 할 때 Center VPN GW와만 연결하면 되므로 확장이 매우 간편하고, 특정 사이트에서 문제가 발생 시 위치를 파악하여 문제를 해결하기가 매우 간편하다는 장점을 가지고 있다. 그러나 고객의 사이트에서 발생되는 모든 트래픽들이 항상 Center VPN GW를 거쳐서 목적지 CPE VPN GW로 전송된다. 이러한 패킷 전달 방법은 대용량의 멀티미디어 서비스의 이용이 많은 기업이나 수많은 지점을 갖고 있는 대규모 기업에서 Center VPN GW에서의 오버헤드를 증가시키는 원인이 된다.

한편, 이와 같은 성형 구조의 문제를 피하기 위해 모든 CPE VPN GW 쌍 간에 직접터널을 설립하는 메쉬 구조를 사용한다면 트래픽의 효율적 전송이 가능하고 Center VPN GW에서의 병목현상을 피할 수 있지만 각 CPE VPN GW의 구성 복잡성이 문제가 된다. CPE VPN GW 간 직접터널은 IPSec[2]의 IKE(Internet Key Exchange)[3]라는 표준 키 교환 메커니즘을 이용하여 설립되어질 수 있다. 그러나 터널 설립 절차를 수행하기 이전에 통신하고자 하는 상대 CPE VPN GW의 IP 주소, 요구되는 보안 등급 등 터널 설정에 필요한 기본적인 정보를 관리자가 직접 설정해 주어야 한다[4]. 현재 VPN GW의 근간을 이루는 ADSL 기반의 VPN GW들은 DHCP(Dynamic Host Configuration Protocol)와 같은 동적 IP 환경에서 운용되므로 GW의 기본 환경설정의 잦은 업데이트가 발생한다. 이러한 환경에서 관리자가 터널 설정에 필요한 기본적인 정보를 매번 수동으로 입력해야 하는 것은 관리를 복잡하게 하며 오버헤드를 증가시키는 요인이 된다[5]. 대규모의 VPN 망을 구성하고자 할 경우 이와 같은 관리 오버헤드는 VPN 운용에 심각한 문제점이 될 수 있으므로 현실적으로 운용되는 망에서는 Center VPN GW를 거치는 기본적인 구성을 그대로 사용하고 있는 실정이다. 따라서 CPE VPN GW간 직접터널 설립을 위한 관리 오버헤드를 최소화하면서 Center VPN GW에서의 오버헤드를 줄일 수 있는 방안이 연구되어야 한다.

대규모의 VPN 망 구성 시 관리 오버헤드를 줄이기 위한 연구로 정책 기반 IPSec 매니지먼트에 대한 방안들이 제안되었다[4,6,7]. 서로 다른 보안정책을 요구하는 IP 패킷에 적절한 보안정책을 적용하기 위해서는 각 CPE VPN GW에 보안정책을 프로비전(Provision)해야 하는데, 이를 위해 이들 방안에서는 정책 서버를 두고 정책서버가 CPE VPN GW의 보안정책 및 자원구성을 수행하도록 제안하였다[4,6], 본 논문에서 제안하는 방안도 자동적인 CPE VPN GW 구성을 위해 Center VPN GW에 이를 수행하는 서버를 도입한다는 측면에서 이들 기존 연구와 유사한 입장을 취하고 있으나, 이들 방안과는 달리 사용자 트래픽 플로우 발생에 따라 온디맨드(On-demand)로 터널을 자동생성하고 해제하는 방안을 추가함으로써 CPE VPN GW에서 유지하는 터널을 최소화할 수 있도록 하였다.

또 다른 기존 연구로 여러 도메인들을 거쳐 터널이 설립되어질 때 정확한 정책 룰을 적용하여 중복되어지는 터널의 수를 최소화함으로써 관리자가 수동적으로 네트워크 장비에 보안 및 정책 관련 정보를 설정해야 하는 오버헤드 줄이는 방안도 제안되었다[7]. 이 방안은 터널 중첩을 최소화함으로써 두 VPN GW간 혹은 정책서버와 VPN GW간의 보안 정책 설정상의 관리 오버헤드 및 중첩터널로 인한 전송 오버헤드를 줄이는데 초점을 두고 있으며, 제안하는 방안에서의 주요 이슈인 CPE VPN GW 간 직접터널의 설립을 위한 자동화에 대해서는 다루지 않고 있다.

본 논문에서는 성형 VPN 구조에서 통신하고자 하는 CPE VPN GW 간 직접터널을 설립하고자 할 때 사용자 트래픽 발생에 따라 자동으로 동적 터널 설립이 가능하게 하는 SVOT(Star VPN On-demand Tunnel) 방안을 제안한다. SVOT 방안에서는 주문형 터널 생성(On-demand Tunnel Creation) 메커니즘을 이용하여 터널을 설립하며[8] 관리자의 수동적인 설정을 거치지 않고 자동적으로 터널 설립을 위한 제반 기능들을 수행함으로써 네트워크 관리를 용이하게 한다. 또한 통신하는 CPE VPN GW 간 발생하는 트래픽이 Center VPN GW를 거치지 않고 직접터널을 통하여 전송되므로 효율적인 트래픽 전송이 이루어지며 Center VPN GW에서의 패킷 프로세싱에 대한 과부하도 줄일 수 있다.

시뮬레이션을 통해 기본적인 성형 VPN 구조에서의

터널 설립 메커니즘(이하 본 논문에서는 SVST(Star VPN Static Tunnel)로 부르기로 함), 풀-메시 VPN 구조에서의 터널 설립 메커니즘(이하 본 논문에서는 FVST(Full-mesh VPN Static Tunnel)로 부르기로 함)과 함께 성능을 조사·비교하였다. 시뮬레이션 결과를 통하여 제안하는 SVOT 방안이 SVST 방안에 비해 확장성, 트래픽 전송효율성 및 Center VPN GW의 과부하를 방지하는 측면에서 우수한 성능을 보이면서 종단간 지연 및 처리율에 있어서는 FVST 방안과 거의 비슷한 성능을 보임을 확인할 수 있었다.

본 논문의 구성은 다음과 같다. 1장의 서론에 이어서 2장에서는 제안하는 SVOT 방안에 대하여 살펴보고, 3장에서는 SVOT 방안의 성능 평가를 위해 수행한 시뮬레이션 및 그 결과에 대하여 살펴본다. 마지막으로 4장에서는 결론과 향후 연구에 대하여 기술한다.

2. 제안하는 방안

본 장에서는 성형 VPN 구조에서 CPE VPN GW간 직접터널을 설립하기 위하여 주문형 터널 생성(On-demand Tunnel Creation) 메커니즘을 이용하는 SVOT 방안에 대하여 설명한다.

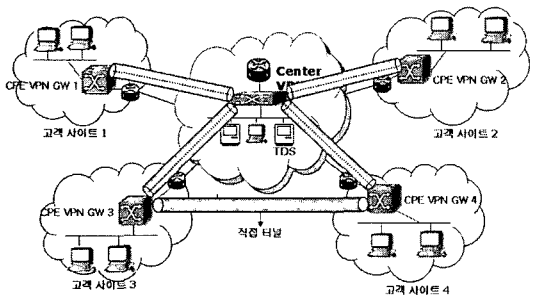


그림 1 성형 VPN 구조의 환경에서 CPE VPN GW 간 직접터널 설립

그림 1은 모든 CPE VPN GW들이 Center VPN GW와 논리적으로 연결되어 있는 성형 VPN 구조의 환경에서 고객 VPN 사이트(Site) 3의 CPE VPN GW 3과 고객 VPN 사이트 4의 CPE VPN GW 4 간에 직접 터널이 설립된 형태를 보이고 있다. 기본적인 성형 VPN 구조인 SVST 방안에서는 모든 CPE VPN GW가 Center VPN GW와만 터널을 설립한다. 따라서 CPE VPN GW에서 나가는 패킷들은 항상 Center VPN GW를 통하여 목적지 CPE VPN GW에게 전달된다. 그러나 제안하는 SVOT 방안에서는 SVST 방안에서 유지하고 있는 터널 이외에 필요에 따라 그림 1의 CPE VPN GW 3과 CPE VPN GW 4처럼 CPE VPN GW 간 직접터널을 설립하고, 이와 같은 직접 터널이 있는 경우에는 CPE VPN GW에서 나가는 패킷들은 Center VPN GW를 거치지 않고 직접 목적지 CPE VPN GW에게 전달된다.

SVOT 방안에서는 CPE VPN GW들이 VPN 망 내에서 터널을 설립함에 있어서 유기적으로 동작할 수 있도록 TDS(Tunnel Directory Service) 서버를 이용한다. TDS 서버는 Center VPN GW와 직접 연결되어 있으며 CPE VPN GW들의 터널 설립에 필요한 제반 정보들을 유지하고 있다. CPE VPN GW는 상대 CPE VPN GW와 직접터널의 설립이 필요할 시에 이를 판단하여 TDS 서버에게 터널 설립을 위한 제반 정보들을 요청한다. TDS 서버는 통신하고자 하는 두 CPE VPN GW들의 터널 설립에 필요한 정보들을 찾아 두 CPE VPN GW에게 각각 전송해준다. 터널 설립에 필요한 제반 정보들을 받은 CPE VPN GW들은 그 둘 간의 터널 설립에 대한 정보를 교환한 후에 터널을 설립한다.

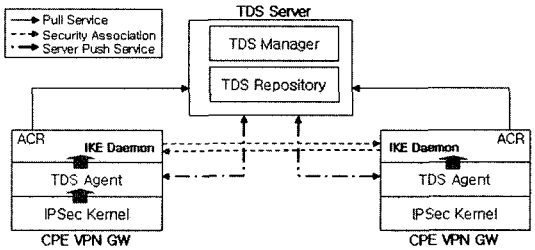


그림 2 주문형 터널 생성 메커니즘 및 VPN 시스템 구성요소들의 메시지 전달 과정

그림 2는 제하는 SVOT 방안에서 주문형 터널 생성 메커니즘 및 VPN 시스템 구성요소들 간의 메시지 전달 과정을 보여주고 있다. SVOT 방안에서 사용되는 VPN 시스템 구성요소들은 다음과 같다. CPE VPN GW는 IPSec 커널 모듈과 TDS Agent, ACR(Auto Configuration Registration) 등으로 구성되어 있으며, TDS 서버는 TDS Manager와 TDS Repository로 구성되어 있다. IPSec 커널 모듈은 직접터널을 설립할 것인지를 판단하여 주문형(On-demand) 터널을 생성하도록 지시하는 모듈이다. TDS Agent는 TDS Manager에게 터널 설립에 필요한 제반 정보를 요청하거나 TDS Manager로부터 제반 정보를 제공받는 에이전트이다. ACR은 CPE VPN GW의 제반 정보가 변경되었을 때 이를 TDS Manager에게 알려주는 작업을 수행한다. TDS Manager는 CPE VPN GW들간 터널 설립에 필요한 정보를 TDS Repository에서 찾아 CPE VPN GW에게 알려주는 역할을 수행한다. 또한 ACR로부터 받은 CPE VPN GW 변경 정보를 TDS Repository에

저장하는 역할도 담당한다.

주문형(On-demand) 터널 생성 메커니즘의 동작과정 및 각 구성요소들 간의 상호 메시지 전달과정은 다음과 같다. CPE VPN GW의 IPSec 커널은 고객의 사이트로부터 패킷이 도착하면 상대 CPE VPN GW와 직접터널이 설립되어 있는지 확인한다. 직접터널이 설립되어 있는 경우에는 해당 터널의 SA(Security Association)를 이용하여 상대 CPE VPN GW에게 패킷을 전송하고, 직접 터널이 없는 경우에는 Center VPN GW와 기존의 설립되어 있던 터널을 이용하여 패킷을 전달 한 후, 상대 CPE VPN GW와 직접터널을 설립할 것인지를 판단한다. IPSec 커널이 직접터널을 설립하기로 결정을 했다면 TDS Agent로 하여금 직접터널 설립을 위한 제반 작업을 수행하게 한다. CPE VPN GW의 TDS Agent는 IPSec 커널로부터 새로운 터널을 설립하라는 요청을 받았을 때 TDS 서버의 TDS Manager에게 Query 메시지를 보낸다. 이 메시지는 통신하고자 하는 상대 CPE VPN GW와의 터널 설립에 필요한 목적지 CPE VPN GW 주소 및 제반 정보에 대하여 요청하는 메시지이다. TDS Manager는 TDS Repository에서 해당 CPE VPN GW의 정보를 찾아 TDS Agent에게 전송한다. 이때 전송되는 정보는 요청한 CPE VPN GW의 IP 주소, 서브넷 정보, SA을 맺는데 필요한 부가정보 등이다. 또한 TDS Manager는 직접터널을 맺을 목적지 CPE VPN GW에게 Tunnel Notification 메시지를 보냄으로써 두 개의 CPE VPN GW가 유기적으로 동작할 수 있도록 한다. TDS Manager로부터 정보를 받은 두 CPE VPN GW의 TDS Agent들은 IKE 메커니즘이 효과적으로 동작할 수 있도록 제반 환경을 자율적으로 구성한다. IKE가 활성화되고 두 CPE VPN GW간 터널 생성에 필요한 자료의 교환이 이루어지면 직접터널이 설립된다. 한편, ACR은 자신의 IP 주소, 서브넷 주소 등 제반 설정 정보가 변한 것을 감지하게 되면 이를 TDS 서버의 TDS Manager에게 등록하여 자신의 네트워크 정보가 변경되었음을 알린다. TDS Manager는 CPE VPN GW의 ACR 모듈로부터 받은 변경사항을 TDS Repository에 업데이트한다.

CPE VPN GW는 다른 CPE VPN GW와 직접터널을 맺고 있으면 그와 관련된 터널 정보를 유지해야 한다. 이때 CPE VPN GW에서 유지하는 터널의 정보는 확장성과 자원할당 및 네트워크 성능에 직접적으로 영향을 미칠 수 있기 때문에 가급적 최소화되어야 한다. 따라서 CPE VPN GW간 직접 터널은 적절하게 설립되고 해제되어야 한다. 즉, 통신하는 두 CPE VPN GW 간 발생하는 플로우의 특성 및 네트워크의 특성을 반영하여 터널을 설립하고 해제함으로써 불필요한 터널 설립으로 터널 오버헤드를 증가시킨다거나 직접 터널이 필요함에도 불구하고 터널이 설립되지 않아 비효율적인 트래픽 전송이 발생하는 것을 최소화하여야 한다.

이를 위해 제안하는 방안에서는 각 CPE VPN GW가 다른 CPE VPN GW와의 통신을 모니터링 하도록 한다. CPE VPN GW는 VPN 사이트로부터 트래픽이 발생하면 일단 Center VPN GW와 설립된 터널을 이용하여 트래픽을 전송하면서 일정 시간동안 들어오는 트래픽의 양을 살펴보고 그 트래픽양이 터널 설립 판단 기준인 Decision_value 이상이면 목적지 CPE VPN GW와 직접터널을 설립한다. CPE VPN GW가 들어오는 트래픽의 양을 살펴보는 시간(이후 Monitering_Time으로 표시)은 트래픽 플로우 지속시간의 통계치에 근거하여 책정되어야 하는데, 제안하는 방안에서는 단순하게 평균 플로우 지속시간의 1/n로 설정하도록 한다. 한편, 이미 설립되어 있는 터널은 일정기간(이후 Expire_Time으로 표시) 해당 터널로 전송되는 패킷이 없는 경우 해제되도록 하였다. 이를 위해 각 CPE VPN GW는 현재 자신이 설정하고 있는 모든 직접터널에 대하여 마지막으로 패킷을 전송한 이후 경과한 시간이 얼마인지를 모니터링 하여야 한다. 그리고 이 직접터널에 대하여 마지막으로 패킷을 전송한 이후 터널 해제 기준 값인 Expire_Time이 지나면 자동적으로 터널정보를 삭제한다. 이때 Expire_Time은 터널에서 전송되는 트래픽의 패킷 도착 간격 분포를 기반으로 적절하게 책정되어야 한다. Expire_Time을 너무 길게 주면 필요하지 않은 터널 정보를 유지해야 하므로 CPE VPN GW에서의 오버헤드가 커지게 된다. 반대로 Expire_Time을 너무 짧게 주면 실제로 계속 필요한 터널을 해제하게 되어 빈번하게 새로운 터널설립을 해야 하기 때문에 프로세싱 오버헤드가 발생할 수 있다.

3. 성능평가

제안하는 방안의 성능평가를 위해 캘리포니아 버클리 대학에서 개발된 NS-2(Network Simulator-2)[9]를 이용하여 시뮬레이션을 수행하였다. 본 시뮬레이션을 위해 NS-2에서 제공하는 기본적인 노드(Node)의 기능을 상속받은 CPE VPN GW와 Center VPN GW 역할을 수행하는 노드를 각각 만들고, Center VPN GW에는 TDS Manager를 CPE VPN GW에는 TDS Agent를 노드에 연결하였다. 또한 Center VPN GW와 CPE VPN GW에는 터널링 기능을 수행할 수 있도록 인캡슐레이션과 디캡슐레이션 프로시저를 추가하여 시뮬레이션을 수행하였다.

시뮬레이션에서 사용된 네트워크는 그림 3에서와 같이 한 개의 Center VPN GW와 20 개의CPE VPN

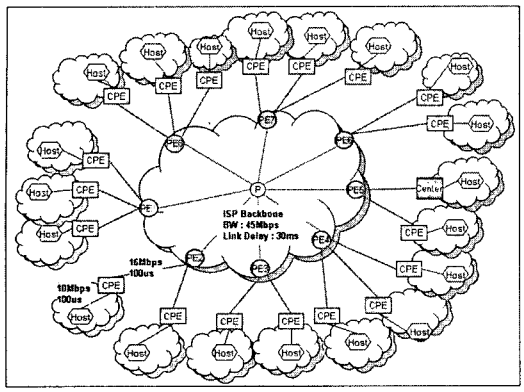


그림 3 시뮬레이션 토폴로지

GW로 구성된다. 이는 중소기업의 VPN 규모 정도에 해당하는데, 본 실험에서 사용한 시뮬레이션 툴인 NS의 제약으로 인해 이와 같이 비교적 규모가 작은 VPN을 대상으로 실험하였다. CPE VPN GW는 각각의 실험에 따라 5~20개가 활성화되도록 하였다. 고객의 VPN 사이트에서의 대역폭은 LAN을 가정한 10Mbps로 할당하였으며, CPE VPN GW와 PE(Provider Edge device) 간의 대역폭은 ISP(Internet Service Provider)에서 제공하는 VPN LAN 사이트 접속속도를 가정한 16Mbps, 백본망에서의 대역폭은 45Mbps로 가정하였다. 백본망에서의 링크 지연시간은 30ms, PE와 CPE VPN GW 간 링크 지연시간과 고객 VPN 사이트 내에서의 링크 지연시간은 100us로 가정하였다. 시뮬레이션에서 사용한 트래픽 전송은 UDP를 이용한 CBR(Constant Bit Rate)을 가정하였으며, 플로우의 전송률은 0.5Mbps~2Mbps로 변화시켜보았다. 각 고객 VPN 사이트에서의 호스트(Host)가 발생시키는 플로우(Flow)의 발생간격은 시뮬레이션 시간동안 포아송(Poisson) 분포로 발생하며, 플로우 지속시간은 1~30초 내에서 유니폼(Uniform) 분포로 지속된다.

본 시뮬레이션에서는 임의의 CPE VPN GW에서 트래픽 발생이 되었을 때 목적지 CPE VPN GW로의 직접터널 설립여부를 결정하기 위해 패킷 도착율을 검사하는 시간인 Monitering_Time을 시뮬레이션에서 사용한 플로우 모델의 평균 플로우 지속시간인 15초의 1/10인 1.5초로 가정하였다. 그리고 이 Monitering_Time 동안 Decision_Value가 시뮬레이션에서 사용한 플로우 모델에서 최소 전송률을 가지는 플로우의 전송률인 0.5Mbps 이상이면 송신 CPE VPN GW가 수신 CPE VPN GW로의 직접터널을 설립하도록 하였다. 이와 같이 터널 설립 기준을 가정함으로써 본 시뮬레이션에서는 플로우 지속시간이 1.5초 미만인 짧은 플로우를 제외한 모든 플로우에 대해 CPE VPN GW 간 직접터널이 생성된다. 한편, 터널 해제를 위한 Expire_Time 값은 0.1초로 가정하였다. 본 시뮬레이션의 트래픽 플로우 모델에서는 CBR 플로우를 가정하였고 플로우의 최저 전송률을 0.5Mbps로 가정하였기 때문에 플로우가 지속되는 상황에서의 최대 패킷 도착 간격은 0.008초이다. 따라서 링크지연 및 과부하로 인한 지연시간을 포함하더라도 0.1초간 패킷 도착이 없다면 그 터널에서 진행 중인 플로우가 없음이 확실하다. 따라서 본 시뮬레이션에서는 임의의 CPE VPN GW가 설립한 직접터널에서 0.1초 이상 패킷 도착이 없다면 그 직접터널이 사용되지 않는다고 판단하고 터널을 해제한다. 실제 네트워크에서의 트래픽 발생에 적합한 Expire_Time을 결정하기 위해서는 터널에서 전송되는 트래픽에 대한 통계적 분석이 이루어져야 하며 이에 대한 효율적인 결정방안 자체가 새로운 연구주제가 될 수 있다.

본 시뮬레이션에서는 CPE VPN GW의 수, 하나의 사이트에서 발생하는 플로우의 수, 플로우별 패킷 전송률을 변화시켜보면서 VPN의 확장성[5], 트래픽 전송에 대한 효율성, Center VPN GW에서의 오버헤드를 측정하였다. 또한 모든 실험에서 CPE VPN GW 쌍에 플로우가 유니폼(Uniform) 분포 형태로 발생하도록 한 Even 형태와 특정 쌍에 플로우 발생이 빈번하도록 한 Uneven 형태를 시뮬레이션 하였다. Uneven 형태는 전체 플로우 발생 횟수 중 40%가 전체 CPE VPN GW 쌍(Pair)의 10%의 비율로 발생하도록 하였다. 실험에서 사용된 모델의 CPE VPN GW의 수와 하나의 사이트에서 발생하는 플로우 수, 호스트에서의 패킷 전송률은 표 1과 같다.

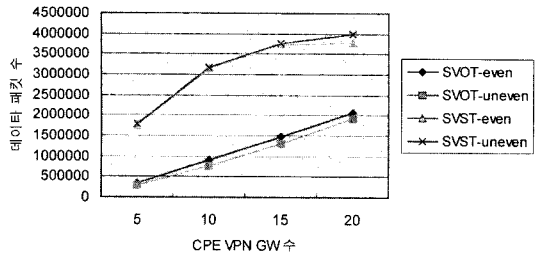
3.1 Center VPN GW를 통해 전달되어지는 데이타 패킷의 수

Center VPN GW는 패킷을 터널링 하기 위해 인캡슐레이션/디캡슐레이션 작업을 수행한다. 따라서 Center VPN GW를 지나는 패킷이 많을수록 패킷 프로세싱 과부하가 증가하게 된다. 그림 4는 Center VPN GW의 패킷 프로세싱 오버헤드를 측정하기 위하여 Center VPN GW를 지나는 패킷의 수를 구한 결과이다.

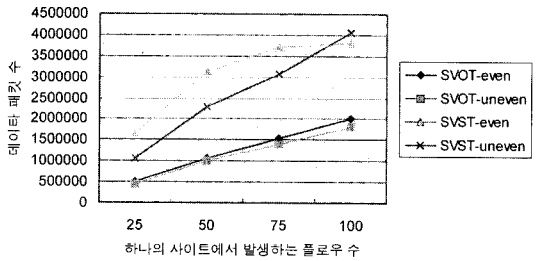
그림 4에서 볼 수 있듯이, 세 모델에서 모두 SVST 방안이 SVOT 방안보다 Center VPN GW를 지나는 패킷의 수가 약 1~5배 정도 많았다. SVST 방안의 경

표 1 실험에서 사용된 파라미터

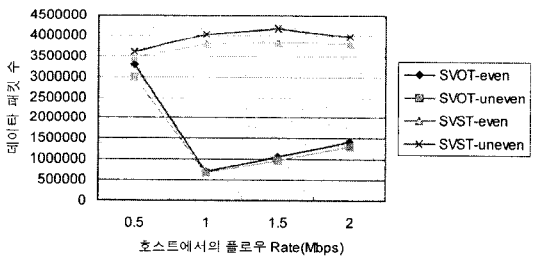
시뮬레이션 모델명	CPE VPN GW의 수(개)	하나의 사이트에서 발생하는 플로우 수(개)	호스트에서의 패킷 전송률(Mbps)
A-모델	5, 10, 15, 20	100	0.5 ~ 2
B-모델	20	25, 50, 75, 100	0.5 ~ 2
C-모델	20	100	0.5, 1, 1.5, 2



A-모델



B-모델



C-모델

그림 4 Center VPN GW에서의 데이타 패킷의 오버헤드

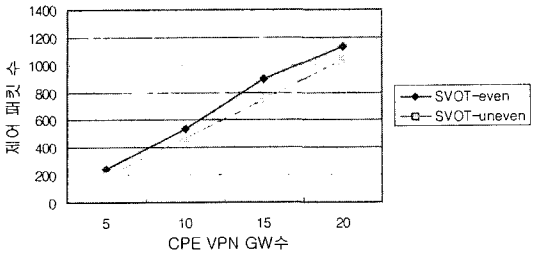
우 모든 패킷이 Center VPN GW를 통해서만 전달되기 때문에 Center VPN GW에서의 패킷 프로세싱 오버헤드가 SVOT 방안보다 더 큼을 알 수 있다.

그림 4의 A-모델의 SVST 방안의 경우 Center VPN GW를 지나는 패킷의 수가 Even 형태와 Uneven 형태에서 거의 비슷한 반면, B-모델의 경우 Even 형태가 UnEven 형태보다 많았다. A-모델은 CPE VPN GW 수를 변화시켜보면서 성능을 측정하는 모델이다. A-모델에서 CPE VPN GW 수의 증가는 CPE VPN GW 쌍의 수를 증가시킨다. 따라서 CPE VPN GW의 증가로 인해 네트워크에 주입되는 트래픽이 증가하여도 여러 CPE VPN GW 쌍 간에 플로우가 분산되기 때문에 UnEven 형태로 트래픽을 발생시켜도 네트워크 일부분에 대한 트래픽 집중 문제가 B 모델만큼 심각하지 않다. 그러나 B-모델에서는 CPE VPN GW 수가 일정하고 그에 따른 CPE VPN GW 쌍의 수도 일정한데 비해 네트워크에 주입되는 트래픽의 증가로 특정 CPE VPN GW 쌍에 많은 플로우가 집중되며 이로 인한 링크의 혼잡과 패킷 손실이 많이 발생한다. SVST에서는 전달되는 모든 패킷이 Center VPN GW를 지나야 하므로 손실이 많은 Uneven 형태의 트래픽 발생의 경우 Even 형태 경우보다 Center VPN GW를 지나는 패킷의 수가 훨씬 적었다.

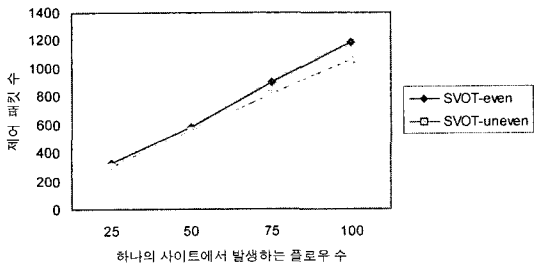
한편, 그림 4의 C-모델에서 보면, 플로우별 패킷 전송률이 0.5Mbps일 때는 제안하는 SVOT 방안에서 Center VPN GW를 지나는 데이타 패킷의 수가 SVST의 경우와 거의 유사하다. 본 실험에서는 CPE VPN GW에서 터널 설립 기준 값인 Decision_Value를 0.5Mbps로 가정하였으므로 호스트에서의 플로우 전송률이 0.5Mbps인 경우에는 CPE VPN GW에서 받은 패킷 도착률이 Decision_Value에 미처 도달하지 못해서 직접 터널이 설립되지 않고 대부분의 패킷들이 Center VPN GW를 통해 전달되기 때문이다. 호스트에서 발생하는 패킷 전송률이 1Mbps 이상인 경우에는 통신하는 CPE VPN GW 간 직접터널이 설립되어 트래픽이 전송되므로 직접터널 설립 전까지 받은 패킷들만 Center VPN GW를 통해 전달된다.

3.2 터널 설립을 위해 CPE VPN GW에서의 제어 메시지 발생 수

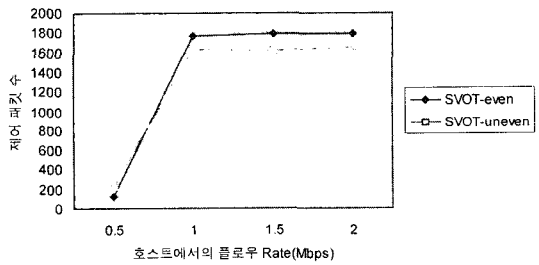
SVST 방안에서는 VPN 초기구성 시에만 Center VPN GW와 CPE VPN GW 간 터널 설립 정보 교환이 발생하지만, SVOT 방안에서는 통신하고자 하는 CPE VPN GW간 직접터널이 필요할 때 Center VPN GW의 TDS 서버에게 상대 CPE VPN GW의 제반정보를 요구하는 제어 메시지가 발생한다. 그림 5는 CPE VPN GW가 Center VPN GW의 TDS 서버에게 보내는 제어 메시지의 수를 Even 형태와 Uneven 형태 각각에 대해 구한 결과이다. 그림 5에서 보는 바와 같이, 세 가지 모델 모두에서 Uneven 형태가 Even 형태보다 제어 패킷의 수가 더 적었다. Uneven 분산 형태는 특정 CPE VPN GW 쌍 간 플로우가 집중적으로 발생하는 형태인데, 통신하고자 하는 CPE VPN GW 쌍 간 이미 직접 터널이 생성되어 있다면 터널 요청을 위한 제어 패킷을 보내지 않는다. 실제 네트워크에서는 통신하는 노드간에 일률적으로 트래픽이 발생하는 것이 아니라 특정 링크에 트래픽이 집중되는 Uneven 형태의 플로우가 발생하는 경우가 많다. 그런데 이와 같은 환경에서 SVOT 방안의 제어 메시지의 발생 오버헤드는 감소하게 되고, 일단 설립된 직접 터널의 활용은 높기 때문에 실제 네트워크에서 매우 유용한 방안으로 볼 수



A-모델

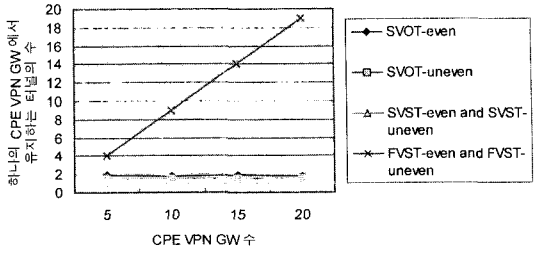


B-모델

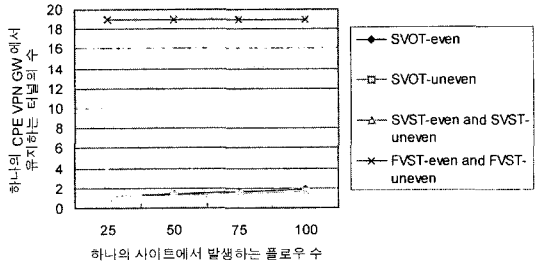


C-모델

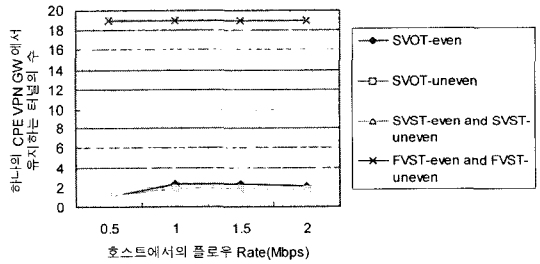
그림 5 CPE VPN GW에서의 제어 메시지 발생 오버헤드



A-모델



B-모델



C-모델

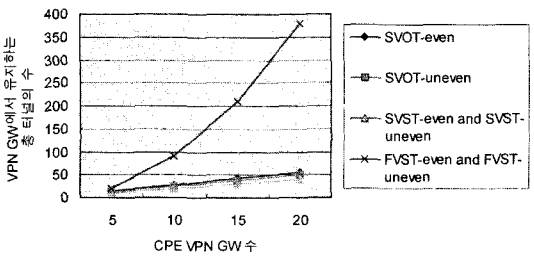
그림 6 하나의 CPE VPN GW가 유지하고 있는 터널 오버헤드

있다.

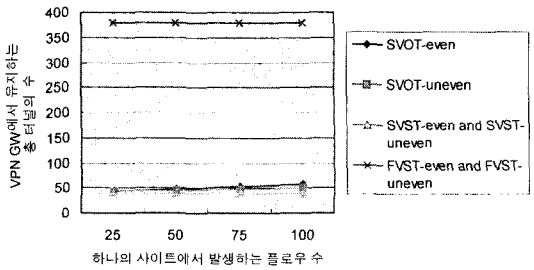
3.3 VPN GW에서 유지해야 하는 터널의 수

그림 6은 확장성 측면에서 CPE VPN GW에서 터널 정보를 유지하기 위한 오버헤드를 측정하기 위하여 하나의 CPE VPN GW에서 유지하는 터널의 수를 Even 형태와 Uneven 형태 각각에 대해 측정한 결과이다. 그림 6에서 보다시피, SVST 방안은 CPE VPN GW가 Center VPN GW와 하나의 터널을 유지하면 되므로 하나의 CPE VPN GW에서 유지해야 하는 터널 수는 CPE VPN GW 수가 늘어나는 것과 관계없이 항상 1이다. FVST 방안은 각각의 CPE VPN GW 간에 독립적인 터널을 유지하기 때문에 하나의 CPE VPN GW가 유지해야 하는 터널의 수는 CPE VPN GW 수 - 1에 해당한다. SVOT 방안은 기본적으로는 SVST 방안에서와 같이 Center VPN GW와 터널을 유지하면서 필요에 따라 동적으로 다른 CPE VPN GW와 직접 터널을 설립 및 해제한다. CPE VPN GW가 유지해야 하는 터널 오버헤드는 SVST 방안보다 약간 더 높으나 그 차이가 크지 않음을 알 수 있다.

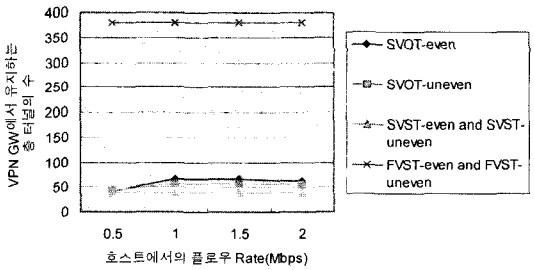
그림 7은 확장성 측면에서 전체 VPN GW에서 유지하는 터널 오버헤드를 측정하기 위하여 Center VPN GW와 모든 CPE VPN GW가 유지하는 총 터널의 수를 Even 형태와 Uneven 형태 각각에 대해 측정한 결과이다. SVST 방안은 Center VPN GW와 각각의 CPE VPN GW 간 양방향으로 터널을 유지하기 때문에 전체 VPN GW에서 유지하는 터널의 수는 CPE VPN GW 수 * 2 이다. FVST 방안은 모든 CPE VPN GW 간 터널을 유지해야 하므로 CPE VPN GW 수 * (CPE VPN GW 수 - 1)이다. 따라서 CPE VPN GW가 늘어



A-모델

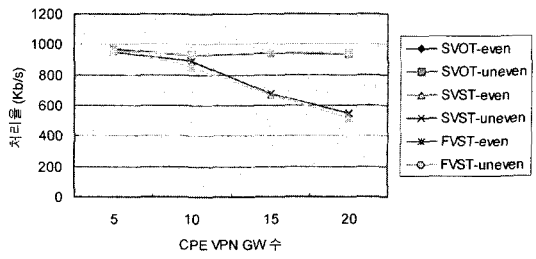


B-모델

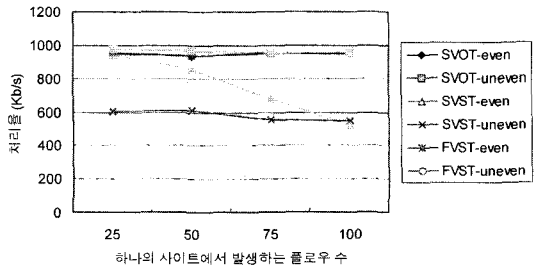


C-모델

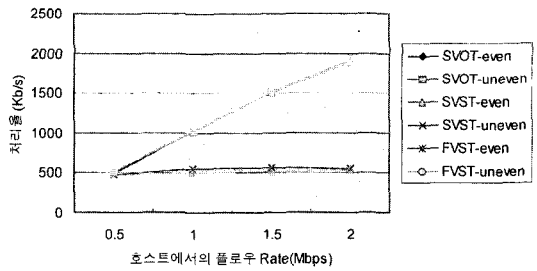
그림 7 전체 VPN GW가 유지하고 있는 터널의 오버헤드



A-모델



B-모델



C-모델

그림 8 패킷 처리율

날수록 모든 CPE VPN GW가 유지해야 하는 터널의 수는 그림 7의 A-모델에서와 같이 기하급수적으로 증가한다. SVOT 방안은 SVST 방안에서와 같이 기본적으로 유지해야 하는 터널의 수는 같으며 필요에 따라 동적으로 직접터널을 생성했다가 정해진 시간동안 터널이 사용되지 않으면 해제하기 때문에 유지해야 하는 터널의 오버헤드는 SVST 방안보다 약간 높은 정도이다.

3.4 Throughput

그림 8은 CPE VPN GW의 수, 하나의 사이트에서 발생하는 플로우 수, 호스트에서의 플로우 패킷 전송률을 변화시켜보면서 세 가지 방안의 패킷 처리율을 구한 결과이다. 표 1에서와 같이 호스트에서의 플로우 전송률을 0.5~2Mbps로 주었을 때, 그림 8의 A-모델과 B-모델의 SVOT 방안과 FVST 방안은 일정한 패킷 처리율을 보인 반면, SVST 방법은 패킷 처리율이 점점 떨어졌다. SVOT 방안과 FVST 방안은 네트워크에 주입되는 트래픽이 많아지더라도 CPE VPN GW 간 직접 터널을 이용하여 패킷이 전송되기 때문에 트래픽이 분산되어 전송되지만 SVST 방안은 항상 Center VPN GW를 거쳐서 전송되기 때문에 네트워크에 주입되는 트래픽이 많아지면 트래픽 혼잡과 패킷 손실이 발생하여 패킷 처리율이 떨어진다. B-모델의 SVST 방안에서 Uneven 형태의 경우는 일정한 처리율을 보이지만, 다른 실험에 비해 훨씬 낮은 처리율을 보이고 있다. 이는 호스트에서 발생하는 플로우 수가 25개인 경우에도 이미 Uneven 형태의 트래픽이 집중되는 링크에 혼잡과 패킷 손실이 발생하였기 때문이다. 이를 통해 앞서 그림 4의 B-모델에서 보았던 SVST 방안의 Uneven 형태가 Even 형태보다 Center VPN GW를 통과한 패킷의 수가 적었던 결과를 뒷받침해 주는 것을 알 수 있다.

C-모델의 경우 SVOT 방안과 FVST 방안에서의 처리율은 각 플로우별 패킷 전송률에 대응하는 성능을 보인 반면 SVST 방법에서는 플로우별 패킷 전송율을 최대 2Mbps까지 증가시켜도 약 0.6Mbps 정도의 처리율을 보였다. 그 이유는 링크의 혼잡으로 인한 것이다. 예를 들어, 한 사이트에서 16개의 호스트가 평균 1Mbps의 전송률로 동시에 통신한다면 본 시뮬레이션에서 가정한 CPE VPN GW과 백본망의 PE 사이의 링크 대역폭인 16Mbps를 모두 사용한 상태가 되므로 트래픽 혼잡이 발생한다.

3.5 End-to-End Delay

그림 9는 CPE VPN GW 수, 하나의 사이트에서 발생하는 플로우 수와 플로우별 패킷 전송률을 변화시켜 보면서 종단간 평균 지연시간을 구한 결과이다.

그림 9에서 보는 바와 같이 FVST 방안에서는 0.05ms의 지연시간을 보인 반면 SVOT 방안은 더 높은 지연시간을 보였다. SVOT 방안에서는 CPE VPN GW 간 직접터널 설립 전까지 들어오는 패킷은 Center VPN GW를 거쳐서 전송되기 때문이다. SVST 방안은 항상 패킷들이 Center VPN GW를 지나가기 때문에 FVST 방안과 SVOT 방안보다 훨씬 높은 지연시간을 보이고 있다.

그림 9의 C-모델의 경우, FVST 방안은 플로우별 패킷 전송률에 관계없이 일정한 종단간 지연시간을 보인 반면 SVOT 방안은 패킷 전송률이 0.5Mbps인 경우에는 직접터널 생성이 설립될 가능성이 낮아서 높은 지연시간을 보이다가 1Mbps 이상의 패킷 전송률에는 항상 직접터널이 생기므로 FVST 방안보다 약간 높은 지연시간을 보이고 있다. SVST 방안은 패킷 전송률에 관계없이 항상 높은 지연시간을 보이고 있다.

4. 결 론

본 논문에서는 성형 VPN 구조에서 CPE VPN GW 간 직접터널을 설립하기 위한 방법으로써 주문형 터널 생성(On-demand Tunnel Creation) 메커니즘을 이용한 SVOT 방안을 제안하였다. 제안하는 SVOT 방안은 관리자의 수동적인 설정을 거치지 않고 자동적으로 터널 설립을 위한 제반 기능들을 수행함으로써 네트워크 관리를 용이하게 한다.

시뮬레이션을 통해 제안하는 SVOT 방안이 SVST 방안에 비해 확장성, 트래픽 전송 효율성, Center VPN GW의 과부하를 방지하는 측면에서 우수한 성능을 보임을 알 수 있었다. 또한, 종단간 지연시간과 처리율에 있어서는 FVST 방안과 거의 비슷한 성능을 보임을 확인할 수 있었다.

본 연구를 실제 네트워크에 적용했을 때 기존의 성형

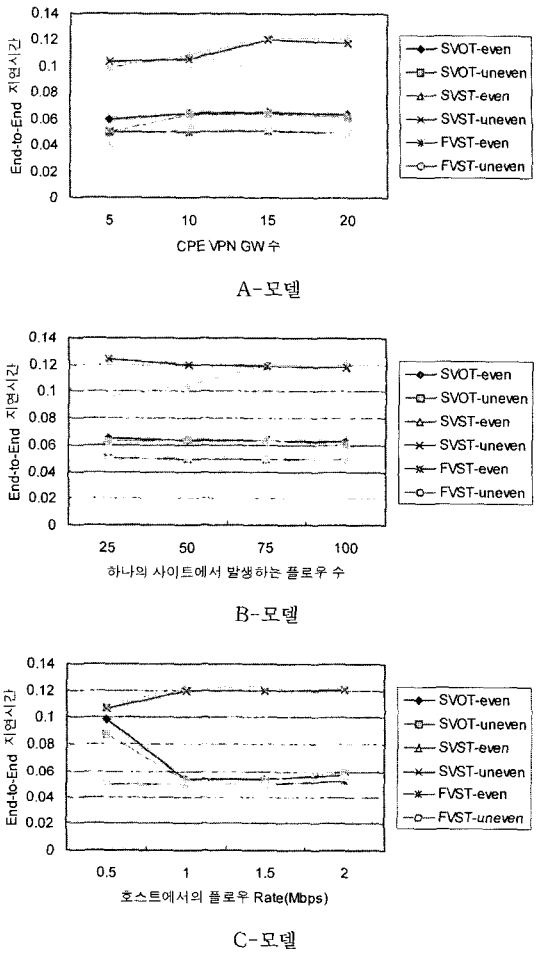


그림 9 종단간 지연시간

VPN 구조로 VPN을 구축한 기업들이 새로운 형태로 VPN 구조로 변경하는 것 없이 풀-메시 구조의 VPN 효과를 낼 수 있으므로 새로운 VPN 구축 비용을 절감할 수 있는 장점을 갖게 된다. 향후 다양한 응용 트래픽에 대하여 실제적인 트래픽 트레이스를 수집하여 실험함으로써 터널에서 전송되는 트래픽 특성에 적합하게 터널을 설립하고 해제 시점을 결정하는 방법에 대하여 좀 더 구체적으로 연구하고자 한다.

참 고 문 헌

[1] B. Gleeson, A. Lin, J. Heinanen, G. Armitage A. Malis, "A Framework for IP Based Virtual Private Networks," RFC 2764, Informational, 2000.

[2] R. Atkinson, "Security Architecture for the Internet Protocol," RFC 1825, Standards, 1995.

[3] D. Harkins, D. Carrel, "The Internet Key

Exchange(IKE)," RFC 2409, Standards, 1998.
[4] Man Li, "Policy-Based IPsec Management," IEEE Network, Vol. 17. no.6, pp. 36~43, 2003.
[5] Jeremy De Clercq, Olivier Paridaens, "Scalability implications of virtual private networks," IEEE Communications Magazine, no. 5, pp. 151-157, 2002.
[6] Dinesh C. Verma, "Simplifying Network Administration Using Policy-Based Management," IEEE Network, vol. 16, no. 2, pp. 20~26, 2002.
[7] Y. Yang, C. U. Martel, S. F. Wu, "On Building the Minimum Number of Tunnels:An Ordered-Split approach to managem IPSec/VPN policies," NOMS-IEEE/IFIP Network Operations and Management Symposium, vol. 9, no. 1, pp. 277-290, 2004.
[8] Lisa Phifer, "Speeding Deployment from the Center: eTunnels VPN-on-Demand," http://www.isp planet.com/technology/etunnels1.html, 2000.
[9] http://www.isi.edu/nsnam/ns/index.html.

변 해 선
정보과학회논문지 : 정보통신
제 32 권 제 2 호 참조

이 미 정
정보과학회논문지 : 정보통신
제 32 권 제 1 호 참조