다중링크 VPN 부하균등 기술 개발

김정우, 손주영°
한국해양대학교 컴퓨터공학과

Development of a Multi-link VPN Load Balancing Technology

JungWoo Kim, JooYoung Son
Dept of Computer Engineering, Korea Maritime Univ.

요 약

현재 다중 링크VPN(Virtual Private Network) 장비들은 회선 장애(Fail-Off) 발생 시 회선 간의 전이(Transition) 기능만 가능할 뿐이고, 부하균등(Load Balancing)은 고려하지 않고 있다. 이럼으로써 다중 회선의 효율성이 크게 떨어지는 결과를 보인다. 본 논문에서는 부하균등을 하면서 회선장애 발생 시 전이되는 기술을 제안하였다.

1. 서론

VPN의 출현으로 비용을 절감 및 보안 증대의 효과를 얻고 있지만 향후 보다 강화된 보안 데이터와 멀티미디어 데이터 전송의 요구가 증가될 것이므로 VPN 회선 효율성의 최적화가 중요하다. 현재 단일 회선의 데이터 송수신으로 인한 데이터 손실률과 회선의 결함으로 인한 통신불능 가능성이 높아지게 되므로 이를 극복하기 위해서 다중 링크를 채택함과 동시에 다중 링크 VPN의 부하균등 기능이 필수적으로 요구된다. 따라서, 본 논문에서는 VPN 부하균등에 있어서의 문제점을 논의하고, 보안이 강화된 가상 전용선의 기능을 충실하게 하기 위한 다중 링크 VPN의 부하균등 기술을 제안한다.

2.관련연구

VPN은 공중망을 사용하면서도 사설망 효과를 얻을 수 있는 기술이다. VPN 서비스를 제공하기 위한 3가지 기본 사항은 터널링(Tunneling), 인증(Authentication), 암호화(Encryption)이다[1]. VPN 구성방법에는 Layer2 프로토콜을 사용하는 L2TP(Layer2 Tunneling Protocol)와 PPTP(Point to Point Tunneling Protocol)이 있으며, Layer3 프로토콜을 사용하는IPSec이 있다. 캡슐화는 Layer2기술은 IP와 IPX, Layer3기술은 IP를 사용하며, 패킷인증 및 암호화는 Layer2에는 없는 반면, Layer3에서는 AH헤더와 ESP헤더를 사용한다.[2, 3]

VPN 기술별 트래픽 부하 실험결과, PPTP는 Ping과 같이 OSI 하위계층에서 처리되는 패킷의 경우 가장 큰 트래픽 부하가 발생하였고, IPSec AH와 ESP를 상호 비교하면 AH가 ESP 보다 조금 많은 부하가 걸렸다[4].

또, 가장 큰 부하를 발생시키는 것은 L2TP/IPSEC이었는데 이것은 보안에 초점을 맞춘 것으로서 이중으로 터널을 구성하기 때문이다. 따라서, 보안을 중시하는 응용은 패킷의 무결성 보장을 위해 L2TP/IPSec기술을 채택한 VPN 사용을 권장하며, 시간 제약적인 실시간/멀티미디어를 중시하는 응용은 L2TP에 의한 VPN 사용을 권장한다.

계층 3 기반 라우팅 정보를 이용하는 부하 균등 기법은 크게 2 가지 방식이 있다.[11] 목적지 주소에 의거한 목적지별 부하 균등과 패킷 단위로 이루어지는 패킷별 부하 균등이다. 목적지별 부하 균등은 링크의 사용 빈도가 불균형을 이룰 수 있고, 패킷별 부하 균등에서는 패킷의 도착 순서가 뒤섞이는 현상이 발생할 수 있는 단점을 가진다. 현존하는 부하 균등 기법은 응용과 링크의 특성을 고려하지 않는 문제가 있다. 따라서 응용에 따른 동적인 링크 부하 균등 기법은 전체 링크들간의 부하 균등을 이루면서 응용의 특성을 고려하고, 즉각적인 링크 상황을 반영할 수 있는 점에서 유리한 기법이다.

3. 다중 링크 VPN 부하균등

3.1 다중 링크 VPN 인증 기법

그림 3.1과 같이 VPN 클라이언트에 두개의 회선을 설치한 후 다중 링크 VPN 부하균등 수행 시 A:주(Master), B:부(Slave)가 되고 회선 단절(Fail-Off)에 대비한 A와 B간의 상호 이전(transition)이 원활해야 한다.

그림 3.1 다중 링크 VPN 구조

그러나 현존하는 장비는 VPN 센터와 VPN 클라이언트간에 서로 1:1이라는 정형화된 구성으로 밖에 인식이 되지 않으므로 다중 링크를 채택한VPN에서의 부하균등

이 이루어지지 않는다.[5] 즉, 하나의 장비에 두개의 키 인증을 할 수 없는 것이 문제인 것이다.

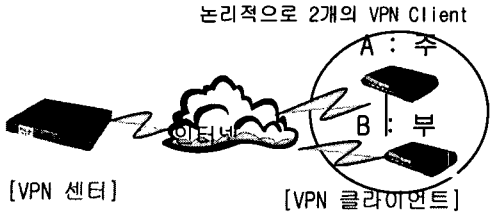

그림 3.2 다중 링크 VPN 부하균등 상세 구조

하나의 장비에 키 인증이 1:다[多]인 경우 현재 하드웨어적으로 해결하지 못하고 있는 VPN 클라이언트 장비의 한계를 극복하기 위하여 소프트웨어적인(논리적인) 방법을 병행하여 부하균등이 가능하게 한다. 이를 위해 그림 3.2와 같이 구성한 후 표 3.1과 같은 키 인증 작업을 실행한다.

표 3.1 다중 링크 VPN 키 인증 동작 순서

1. VPN 센터에서 클라이언트 정보 입력
2. 클라이언트에 키 값을 미리 할당
3. 클라이언트는 자신의 정보를 메모리에 할당하여 클라이언트2 개체를 복제
4. 소프트웨어 복제 시 일정 순서에 의한 값을 할당
5. 각 할당된 키 인증 값이 상호 일치할 때 각각의 클라이언트를 분리된 독립 개체로 인식하여, 다중 링크 VPN 통신이 가능

그림 3.2와 표 3.1을 바탕으로 그림 3.3과 같이 부하균등을 위한 인증 과정을 거쳐 VPN이 형성된다.[6,7]

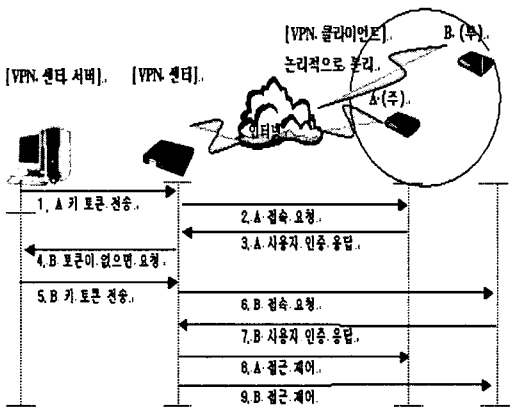

그림 3.3 부하균등 VPN을 위한 인증 과정

3.2 다중 링크 VPN 부하균등 알고리즘

VPN 클라이언트 사이에 부하균등을 위해 집합 분할(set partition) 알고리즘을 적용하였다. 패킷의 원활한 흐름과 VPN 클라이언트 주/부간의 부하 차이를 최소화할 수 있게 하여 링크 효율성을 증대 시킬 수 있다.

(1) 집합 분할 알고리즘의 개요

집합(Set) [n] = {1,2,~ n}에 대해 [n]의 부분집합(Subset)은 B0,B1,...,Bj 들로 나타낼 수 있으며, B0,B1,...,Bj 는 블록(Block)이라고 한다. 각각의 파티션 집합은 순차적으로 블록이 증가를 할 때, 하위의 블록에 포함된 원소(Element)는 포함이 되지 않는다. 이것을 Restricted Growth String이라고 하며, RG String은 다음과 같이 표현이 된다. String a[1,..n]이 있을 때, a[j]의 의미는 원소 j만큼 블록이 발생하게 됨을 뜻한다. RG String의 수식으로는 $a[i+1] < 1 + \max\{a[1],a[2],...,a[i]\}$로 표현이 된다.[8,9]

(2) 집합 분할 알고리즘의 적용

부하균등을 적용함에 있어 가장 중요한 것은 링크 사이에 패킷을 할당하는 기준이다. 기본적인 기준은 패킷을 발생시키는 응용에 기반한다. 이러한 부하균등 알고리즘을 주기적으로 적용하는 것을 원칙으로 한다. 이때 얼마나 자주 알고리즘을 적용해서 링크에 할당되는 패킷을 결정해야 VPN 라우터에서의 부하와 부하균등 정도 면에서 가장 최적한 것인지를 실험과 분석 과정을 거쳐 파악한다. 나아가 포트 또는 패킷별 부하균등과 같은 강제적 부하균등이 아니라 패킷의 시간대별 흐름을 분석한 후 패킷 및 포트와는 상관없이 회선에 가장 적합한 부하균등을 제공함으로써 부하균등 정책을 실시간적이며 유연성 있게 적용할 수 있게 한다.

실험에서 사용되는 기준값(port)은 다음과 같다.

[n] = {pop3, Domain, Http, SMTP, Ms-sql}

여기서 [n]에 포함된 각각의 값들은 시간대별 패킷 분석 결과 가장 많이 사용되어지는 값들로서 회선의 대부분을 차지하는 값들이다. 블록은 실험에서 부하균등을 위해 고려 중인 회선이 2 개이므로 2 블록을 사용하게 된다. 실험 결과의 성능은 각 시간별 집합 분할에 의해 얻은 파티션 값(원소로 지정된 포트로 전송되는 패킷의 양)간의 차이로써 구해지며, 각 결과 값의 분포도상 전체 표면량이 가장 적은 표면 분포값를 나타내는 것이 가장 성능이 가장 우수한 것이 된다.

이를 실험적으로 구현한 부하균등 VPN 라우터 시뮬레이터를 제작하였다.

(3)부하균등 VPN 라우터 시뮬레이터

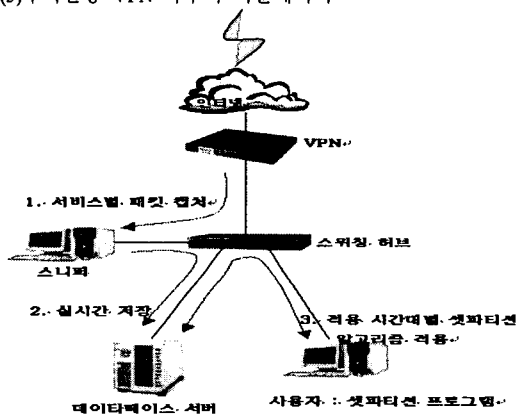

그림 3.4 부하균등 VPN 시뮬레이션 구조

그림 3.4와 같이 운영 환경은 윈도우 95 이상, 메모리 16M, 펜티엄133Mhz 이상이다. 데이터베이스 서버는 Ms-SQL을 사용한다. VPN은 하드웨어 VPN을 사용한다. 그림 3.5에 동작하고 있는 시뮬레이터가 나타나 있다.

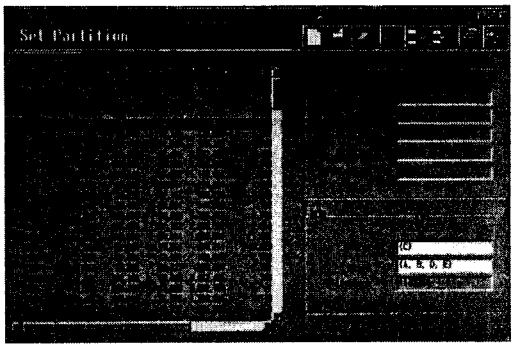

그림 3.5 부하균등 VPN 시뮬레이터 동작 모습

전체적인 동작 흐름은 다음과 같다. 첫째, VPN을 통과한 모든 패킷은 스위칭 허브를 지나게 되는데 이때 스니퍼가 이 패킷을 미러링하여 로그를 복사/분석한 후 서비스 종류와 크기를 파악한다. 둘째, 분석이 완료된 서비스는 실시간으로 데이터베이스 서버에 저장하며 마지막으로 집합 분할 VPN 시뮬레이터는 미리 설정된 시간 간격에 맞춰 주기적으로 데이터베이스 서버로부터 해당 로그 데이터를 가져온 후 각 링크에 포트를 할당하고, 링크에 할당된 패킷 양간의 차이값을 산출한다. 이때 얻은 차이값은 다시 해당 데이터베이스 결과 및 결과 이력 테이블에 저장된다

이러한 일련의 과정을 거쳐 생성된 데이터는 다음의 분석 과정을 거쳐 부하균등 알고리즘을 적용하기 위한 가장 최적의 주기시간을 얻었다.

3.3 시간대별 부하균등 시뮬레이션 결과

부하균등 VPN 시뮬레이션은 모두 15일동안 실행되었다. 각 하루 24시간동안 1분에서 60분까지 부하균등 적용 간격(주기)을 달리하여 실행하였다. 각각의 적용 간격(주기) 시간별로 계산된 결과의 차이값(Difference)들을 평균한 후 간격별로 표시하면 그림 3.6과 같다.

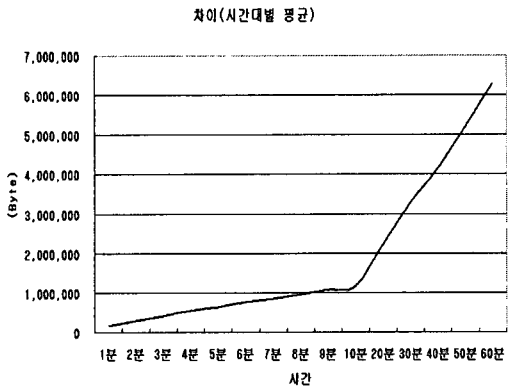

그림 3.6 적용 간격(주기)별 차이값 평균

그림 3.6에서 가장 작은 링크간의 부하차이를 나타내는 것은 적용 간격이 가장 짧은 1분임을 나타낸다. 따라서 부하균등 알고리즘 적용은 자주 할수록 좋은 결과를 내는 것임을 알 수 있다. 하지만 VPN 라우터에서 실시간 부하균등을 실행할 때 라우터내에서의 처리 비용이 증가 하게 되고, 소비 시간이 발생하게 된다. 따라서 부하균등 적용의 가장 적합한 시간을 찾아내기 위해서는 처리지연시간(Processing Delay)을 함께 고려하여야 한다. 이것은 다음과 같이 구할 수 있다.

명령문 수[Instructions]/컴퓨터처리능력[mips]

[명령문의 수]를 계산할 때 프로그램의 흐름에서 가능한 경우의 수를 모두 고려하여야 한다. 전체 프로그램의 흐름은 입력값 5가지를 받게 되면, 다음으로 집합이 부분집합으로 분할될 수 있는 각각의 경우에 대한 링크 부하 값을 계산한 후, 각 경우에 대해 비교 작업을 한다. 그런 후 가장 작은 차이값을 가지는 분할 결과에 따라 각각의 링크로 구분하여 할당된 패킷들을 전송하는 것이다. 본 실험에서 발생할 수 있는 부하균등 시 결과는 입력값이 5가지이고 분할되는 부분집합의 개수가 2이므로 총 15가지의 경우의 수를 가진다. 정리하면 가능한 분할 경우의 수가 15가지, 각각의 차이값에 대해 비교하는 경우의 수가 14가지, 끝으로 회선을 분리하기 위한 과정이 2가지이므로 총 31번의 비교 작업이 수행되어야 한다.[10]

따라서, 비교 동작을 하나의 명령어로 처리할 수 있다고 가정하고, VPN 라우터의 CPU 처리능력이 1 mips라고 할 때, 전체 1회 부하균등 알고리즘 실행 시간은 약 $31/10^6$ 초이며, 이것은 약 0.000031 초의 처리지연시간이 필요함을 나타낸다. 이것을 1분에서 60분 단위로 전체 24 시간동안 적용하였을 때,.누적되는 처리지연시간을 그림 3.7에 나타내었다. 이 그림에서는 앞서 3.6에서 구한 각 간격별로 구한 평균 차이값과 함께 표현하여 처리지연시간과의 상호 비교를 하였다.

적용시간대별 부하량을 구하기 위해서 다음과 같은 수식이 적용된다.

[처리부하(Byte)] = (전체 Byte량 x (처리지연시간 x 시간대별적용횟수))

전체 바이트량을 적용함으로써 적용시간대별 평균차이 값을 구하기 위해 사용된 데이터를 처리함에 있어 동일한 조건을 적용하여 적용시간대별 부하량을 얻기 위함이다. 이는 하루 24시간동안 적용시간대별 부하균등이 실행될 경우를 고려한 것이다. 따라서, 시간대별 적용을 할 경우 실행 되어져야 하는 횟수를 적용하면 그림 3.7과 같이 나타내어 진다.

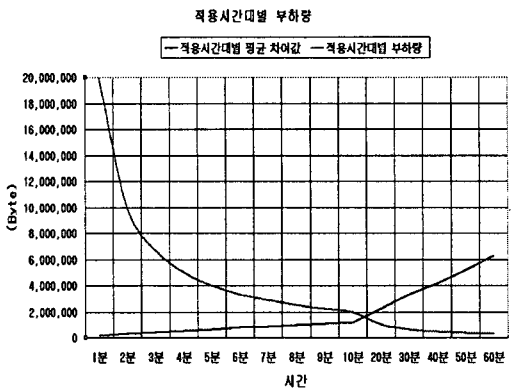

그림 3.7 적용 간격별 차이값과 처리지연시간

그림3.7에 따라 시간별/부하량에 따른 결과로써 10분이 부하균등을 시행함에 있어 가장 적절하며, 안정적임

을 알 수 있다. 하지만, 부하균등 수행 시 적용되는 포트의 수와 사용자 응용들의 타임아웃 시간을 고려하게 되면 적용시간대별 부하가 급감하는 4분~5분 이후가 가장 적절한 부하균등 시점이 될 것으로 예상된다. 이는 사용자 응용이 다양하지 못함으로 인해 발생할 수 있는 부하균등에 따른 링크간의 부하 차이값이 클 경우가 될 것이고, 반대로 다양한 응용들을 골고루 사용하는 경우는 10분 전후가 가장 적정할 것으로 예상된다. 따라서, 이런 시간대가 부하균등 시 VPN 장비의 부하를 최소화하면서, 원활한 부하균등을 수행할 수 있는 시간이 될 것이므로 부하균등 적용 시 사용그룹의 성향이 먼저 파악되어져야 할 것이다. 그리고, 여기서 고려되어야 하는 사항은 서비스의 포트별 경우의 수가 불 일정할 경우인데 이런 경우 적용시간대별 처리 지연시간의 등락 폭이 불 일정하게 될 것이고 이는 부하균등 정책을 수행함에 있어 적용시간 역시 달라질 수 있는 것을 의미한다.

5.결론 및 향후 연구과제

VPN 한 장비에 2개 회선 이상이 연결되게 되면 이중 투자 비용의 감소 및 안정성 있는 서비스가 가능 할 것이다. 이는 현재 VPN을 사용하고 있고, 앞으로 사용할 기업 및 인터넷 회선을 사용하는 대부분의 장소에 위와 같은 부하 균등 기능이 포함된 다중 링크 VPN이 적용될 것으로 예상된다.

향후 완벽한 QoS를 보장하기 위해서는 VPN은 부하균등이라는 과정을 거쳐야 할 것이며, 나아가 송/수신 시 패킷의 안정성도 높아져야 할 것이다. 본 연구 결과로써 VPN 한 장비에 멀티포인터 회선의 사용 시 부하균등 정책을 적용하여 시스템자원의 이중 낭비를 막고, 실 시간적인 부하균등 알고리즘을 제안하여 가장 효율적이며, 적응력을 높일 수 있는 방법 역시 제안하였다.

향후 연구 과제로는 부하균등 적용 시 패킷 손실을 최소화하는 방법, 그리고 응용별로 집합 분할 알고리즘에 의한 부하균등정책을 적용할 때 하나의 세션을 이루는 패킷들을 다른 링크로 할당했을 때 세션의 영속성을 확보하는 문제에 대하여 연구할 계획이다.

[참고문헌]

[1]"Understanding Virtual Private Network", pp.10~11, ADTRAN Inc, 2001

[2]김광호, 임채훈 "PPTP 와 L2TP 의 비교 분석", Cryptography & Network Security Center, Future Systems, Inc, Technical Report, p.3, pp.15-17, Sep. 25, 2000

[3] 오승희, 채기준, 남택용, 손승원, "다양한 트래픽을 이용한 VPN 프로토콜 성능 평가", pp.3-5, 정보처리학회 논문지 C 제 B-C 권 제 6 호, 2001. 12

[4] 박진형, 손주영, "VPN 기술별 트래픽 부하 비교", 멀티미디어 춘계학술 발표회, 2003

[5]"Network Load Balancing Technical Overview", pp.3-4, pp.12-13, Microsoft Windows2000 Server, White Paper, 2000

[6] Stamatis Karnouskos, Ingo Busse, Stefan Covaci, "Place Oriented Virtual Private Networks", pp.3-4, 33rd Hawaii International Conference on System Sciences, 2000

[7] Eli Herscovitz, "Secure Virtual Private Networks: The Future of Data Communications", pp.2-3, p.5, International Journal of network management, 1999

[8] P.C Chu and J.E.Beasley, "A genetic Algorithm for the set partitioning Problem", pp.1-2, The Management School Imperial College, april. 1995

[9] Zbigniew J.Czech, "Heuristic algorithms for solving the set-partitioning problem" p.1, Silesia Univ. of Technology, June. 1997

[10] Huican Ahu, Oscar H.Ibarra, "On Some Approximation Algorithms for the Set Partition Problem", p.2, pp.6-7, California Univ. USA

[11] Jeff Doyle, "Routing TCP/IP Vol.1, pp.109~112, Cisco Press, 1998.