네트워크 트래픽 감시를 통한 동적인 트래픽 부하 분산에 관한 연구

(A study of dynamic traffic load distribution through network traffic monitoring)

김희진, 박상근, 최덕재
전남대학교 정보통신연구소
{hjin, psk}@tyranno.chonnam.ac.kr, dchoi@chonnam.ac.kr

요 약

최근 인터넷 사용자의 증가와 다양한 응용 서비스들의 등장으로 인한 트래픽의 폭증은 전송 지연 및 패킷 손실 등의 현저한 망 성능 저하의 문제를 야기시키고 있다. 문제 해결을 위해 대부분의 ISP는 단순한 대역폭의 확장이나 새로운 라우터의 도입 등으로 망을 확장하거나 Monitoring Tool을 이용하여 수동적인 트래픽 흐름을 조작하는 방법을 사용하고 있는 실정이다. 그러나 이러한 방법으로는 손실을 최소화시키는 시간 내에 망 전체의 포괄적인 관리 정보를 바탕으로 한 효율적인 트래픽 분산을 하기에는 많은 어려움이 존재한다. 본 논문에서는 현재 인터넷 라우팅 프로토콜로 가장 많이 이용되고 있는 OSPF 기반에서 SNMP 관리 메커니즘을 이용한 동적 트래픽 분산 방법을 제안하였다. 발견된 과부하 링크의 트래픽 부하 분산을 위해 제반 규칙을 정의하고, 이러한 규칙에 의해 선택된 링크의 Metric MIB값을 OSPF의 ECMP 라우팅 기능에 맞게 조절함으로써 자동화된 동적 트래픽의 부하 분산을 가능하게 하였다.

1. 서론

1.1 연구배경 및 목적

인터넷 사용자의 급속한 증가는 인터넷 트래픽을 현저히 증가시켰다. 이는 웹의 성장과 네트워크 기술의 발전으로 인해 인터넷이 대중화됨에 기인한 것이다. 인터넷 트래픽의 증가는 비단 사용자의 증가 뿐 아니라 인터넷을 통한 응용서비스의 유형 변화에서도 그 원인을 찾을 수 있다. 이러한 트래픽의 급속한 증가와 이로 인한 망의 성능 저하라는 문제가 발생하였을 때, 관리적 측면에서 망 성능의 저하에 대처하기 위해 이에 대한 원인 분석 및 개선을 위한 망의 성능 감시 및 제어가 요구되고 있다. 그러므로 전송의 지연이나 패킷 손실 등의 망 성능 저하를 사전에 방지하기 위해서는 망의 부하량 측정과 더불어 능동적인 대책을 세워야 한다.

망의 성능 저하를 해결하기 위해 단순히 대역폭을 확장하는 것뿐만 아니라 라우터 수의 증가 및 라우터 성능의 향상 등 다양한 노력을 시도하고 있다. 그러나 이러한 노력은 망의 성능 향상에 크게 영향을 줄 수 있으나 많은 비용으로 인해 적용에 어려움이 있다. 따라서 현재 대부분의 ISP(Internet Service Provider)들은 트래픽의 부하가 많은 곳의 트래픽을 부하량이 낮은 다른 경로로 분산시켜서 망의 성능 저하를 막는 접근 방식을 인터넷상의 트래픽 관리에 사용하고 있다[6].

기존의 트래픽 관리 방법은 트래픽 조정을 위한 노력으로 Monitoring Tool을 사용하여 망의 트래픽을 관찰한 뒤, 과부하인 곳이 발견되면 관리자가 라우터의 콘솔을 통해 라우터 정보의 수동적인 조작으로 트래픽을 조작하는 방법을 이용하고 있다. 이러한 트래픽의 조작을 위해 직접 트래픽의 경로를 변경하여 트래픽을 우회시키거나 Metric 값에 대해 명령어를 통한 값의 변경으로 과부하 트래픽을 분산시키도록 하고 있다[5]. 그러나 기존의 방법은 관리자가 수동적인 조작에 의해 망 트래픽을 직접 조작하기 위해 망 전체에 대해

본 연구는 1999년도 두뇌한국21 사업 핵심 분야에 의하여 지원되었음.

포괄적으로 망 정보를 인지하고 조치를 취해야 하므로 관리자의 부담을 가중시키고, 트래픽 분산 작업을 콘솔을 통한 명령어 입력에 의한 수작업으로 행해야 하므로 비효율적이다. 뿐만 아니라 급변하는 망의 변화에 대처하기 위해서는 기존의 수동적인 조절에 의존하여 망 트래픽을 관리하는 데에 따른 어려움도 있다.

이러한 단점들을 극복하기 위해서 본 논문에서는 현재의 트래픽 상태를 웹 기반의 망관리 시스템(Network Management System)이 감시하여 트래픽의 과부하가 일어나는 것을 자동으로 감지한 후 이에 대한 적합한 조치를 SNMP를 이용한 망관리 시스템을 통해 해결하도록 한다. 이러한 망관리 시스템을 이용한 동적인 트래픽 부하 분산으로 망의 성능 관리를 위한 관리자의 부담을 덜어주고 급변하는 망의 상황을 제어할 수 있도록 한다.

본 논문에서는 연구범위를 현재 IGP(Interior Gateway Protocol)로 많이 이용되고 있는 OSPF 기반의 망으로 제한하도록 하고, QoS와 같은 기능을 제공하지 않는 일반적인 망으로 적용범위를 정하도록 한다.

1.2 OSPF(Open Shortest Path First)

1.2.1 OSPF의 소개

OSPF(Open Shortest Path First)는 IETF에서 1988년에 정의한 IP 라우팅 프로토콜로 IGP(Interior Gateway Protocol)로서 주로 이용되며 경로 결정시 Link-State Algorithm을 사용한다.

OSPF의 주된 특징은 다음과 같다[5].

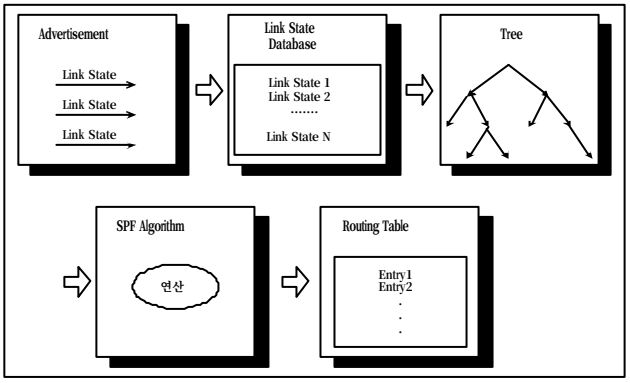
- 네트워크를 영역(Area)으로 구분하여 이것을 계층구조로 구성함으로써 많은 라우팅 정보의 교환으로 인한 라우터의 성능 저하를 예방하고 대역을 절약할 수 있다.
- 링크의 변화가 감지되면 그 링크에 대한 정보만을 즉시 모든 라우터에게 전달하므로 수렴(convergence)이 매우 빨리 이루어지고 해당 링크에 대한 정보만 전달되므로 전체 라우팅 정보를 전달할 때 대역이 많이 소모되는 문제점을 피할 수 있다.
- ECMP(Equal Cost MultiPath) 라우팅을 지원하여 여러 개의 경로를 효율적으로 활용할 수 있다.
- Subnet mask를 라우터에 연결시킬 수 있고 가변길이(variable length) subnet masking을 하게 해준다. 이 특징은 IP subnet addressing model을 확장시키고, IP 주소의 공간을 효과적으로 관리할 수 있게 해준다.
- OSPF는 TOS(Type Of Service)를 제공한다. OSPF는 IP TOS에 따라 서로 다른 경로를 제공하여 줄 수 있다.
- 라우팅 정보 전달시 IP multicast 방법을 이용한다.
- 라우팅 정보 교환시 인증을 지원한다.

본 논문에서는 OSPF의 특징 중 ECMP 라우팅 기능을 이용하도록 한다. ECMP 라우팅은 동일한 Cost를 갖는 경로들에 대해 트래픽을 분산하여 전송할 수 있도록 하는 메커니즘이다. OSPF의 장점인 ECMP 라우팅을 이용하여 과부하인 경로의 Cost를 분산 가능 경로의 Cost와 동일하게 조정하여 Multipath를 만들어 트래픽을 분산시키도록 한다.

1.2.2 Link State와 LSA

OSPF를 운영하고 있는 라우터는 자신에게 접속된 네트워크에 대한 정보를 다른 라우터에게 전달하는데 이 정보를 Link State라고 한다. 이는 OSPF의 라우팅 정책과 밀접한 관련이 있다.

그림 1은 Link State를 기반으로 경로 설정하는 방법을 보여준다. 라우터의 Link State 변경시 Link State packet을 받은 각 라우터들은 그들의 모든 인터페이스에 정보를 flooding하고 자신의 Link State database에 이 정보를 복사한다. 또한 목적지까지의 Cost를 계산하기 위하여 Link State database에서 SPT(Shortest Path Tree)를 형성하고 SPF (Shortest Path First) 알고리즘을 이용하여 목적지까지 가장 적은 Cost로 갈 수 있는 경로를 Routing Table에 등록한다.



(그림 1) OSPF에서의 경로 결정

Link State는 LSA(Link State Advertisement) 메시지로 전달되는데 Link State에 변화가 있을 경우 그와 관련된 LSA를 모든 라우터에게 즉시 전달한다. 라우터가 LSA를 전달받으면 해당 정보가 자신의 데이터베이스에 없을 경우 이것을 자신의 데이터베이스에 등록하고 즉시 그 LSA를 다른 라우터들에게 전달한다. 이러한 LSA는 OSPF에 참여하는 모든 라우터에게 전달된다.

1.2.3 Metric

경로결정 영향요소에는 Bandwidth, Delay, Reliability, Load, MTU, Hop Count 등이 있는데 이러한 것을 Metric Factor라고 한다. 그리고 이들을 근간으로 하여 해당 경로 결정을 위한 Cost 값을 구하며 경로 선택시 Cost가 가장 낮은 경로를 선택하여 라우팅 테이블에 반영한다.

Cost = function (metric factors) [5]

<표 1> 각 라우팅 프로토콜에 따른 Metric Factor

	Metric Factor
RIP	Hop Count
OSPF	Bandwidth
IGRP, EIGRP	Bandwidth, Delay, Reliability, Load, MTU
BGP	Weight, Local preference, AS-path, MED, prefix 길이. (Cost를 계산하는 과정이 RIP, IGRP, OSPF, EIGRP 등과 전혀 다름)

표 1은 각 라우팅 프로토콜에 따른 Metric Factor를 보여준다. 각각의 라우팅 프로토콜에 따라 Metric Factor는 다르게 적용되는데, 그 중 OSPF에서 사용되는 Metric Factor로는 Delay, Throughput, Reliability, Bandwidth 등이 있으나, 기본적으로 Bandwidth만을 이용한다. 앞에서 언급한 바와 같이 OSPF의 특징 중 TOS 라우팅에 관한 사항이 포함되어 있으나 이는 라우터에서 TOS 기능이 활성화되었을 때만 이용이 가능하다. 그렇지만 현재의 라우터에서는 TOS 라우팅이 지원되지 않아 실제로 OSPF에서 통용되는 Metric Factor는 링크의 Bandwidth이다. 현재 OSPF에서 Metric Factor인 Bandwidth를 이용하여 경로의 Cost를 계산하는 방법은 다음과 같다.

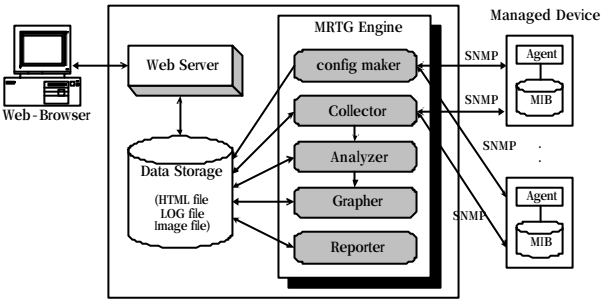
Link's Metric value(Cost) = 100,000,000/bandwidth
Cost = SUM(Link's Metric value) [5]

위의 식에 의하면 10M의 Bandwidth를 가지는 Ethernet은 10, 100M의 Bandwidth를 가지는 FDDI는 1의 Metric 값을 갖게 된다.

1.3 Monitoring Tool

1.3.1 MRTG를 이용한 Monitoring 시스템 구성도

트래픽 감시를 위해 본 논문에서 사용하게 될 Monitoring Tool은 MRTG(Multi Router Traffic Grapher)이다. MRTG는 SNMP를 기반으로 네트워크 링크상의 트래픽 부하를 감시하는 도구이다. SNMP MIB을 이용하여 라우터의 인터페이스별 트래픽 양을 측정하여 그 변화를 이미지로 만들어 그래프를 통해 웹 브라우저로 보여준다. 현재 많은 ISP들이 망의 트래픽을 감시하기 위한 Monitoring Tool로서 이용하고 있다.



(그림 2) MRTG를 이용한 Monitoring 시스템 구성도

그림 2는 MRTG를 이용한 Monitoring 시스템의 구성을 보여주고 있다. MRTG는 망관리 프로토콜인 SNMP를 지원하는 장치들을 관리대상으로 삼는데, 망내 라우터, 허브, 그리고 SNMP agent가 설치된 장비들에 대해 SNMP MIB을 이용하여 감시를 한다. 이를 위해서 먼저 Config Maker를 이용하여 관리대상의 구성 정보를 가져온 후, 정기적으로 해당 장비로부터 MIB 값을 가져와서 이를 Log file에 기록한다. 이러한 원 데이터(raw data)에 대해 트래픽 부하를 Analyzer가 Grapher를 이용하여 이미지 파일로 표현하여 웹에서 접근 가능하도록 웹 문서로 만들어 준다. 또한 Data storage에 모은 정보를 기반으로 망의 트래픽 부하에 대해 Reporter가 보고서를 웹 상에서 볼 수 있도록 웹

문서로 제공하여 준다.

1.3.2 MRTG의 확장

MRTG는 SNMP를 이용하여 수집된 정보에 대해 데이터베이스를 사용하지 않고 Log file로 저장하여 현재 트래픽 양을 그래프 이미지로 웹상에서 보여주는 기능을 하는 감시도구로서, 설정성과 확장성이 우수하여 대부분의 UNIX platform이나 Windows NT 기반의 시스템에서 이용할 수 있다. 이러한 기본적인 기능 이외에 여러 부수적인 기능들이 추가, 확장되었는데 이러한 기능은 MRTG의 contrib 디렉토리에 추가되어 있다. 다음은 contrib에 추가된 기능들 중 본 논문을 위한 감시 시스템에 적용 가능한 기능들이다.

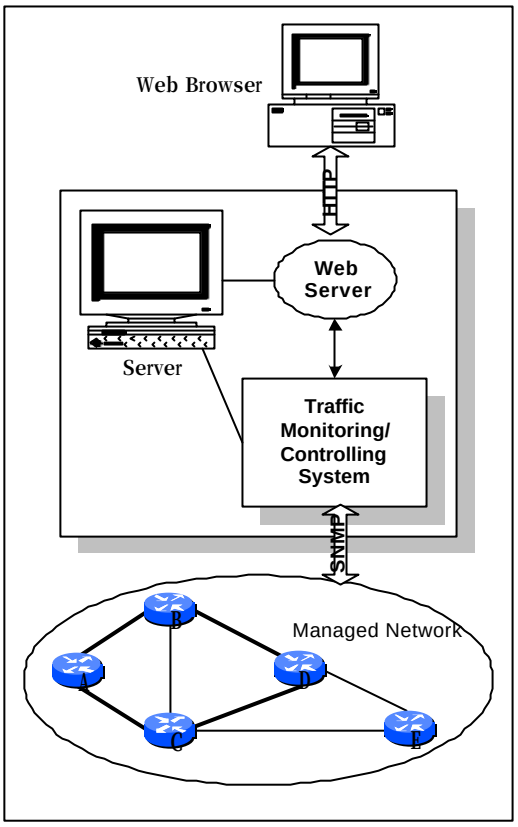
- get-active : 링크의 활용율이 높은 인터페이스부터 순서대로 정렬하여 표 형태로 웹 문서를 만들어 준다. 뿐만 아니라 get-active2에서는 미리 설정된 Threshold을 기준으로 그 값을 넘는 인터페이스를 체크하여 알려준다. 이러한 기능을 이용하여 링크 활용율을 알아본 후 Threshold 값을 넘는 링크를 알려줌으로써 조정이 필요한 링크를 찾아낸다.
- rdlog2 : 망의 구성도를 그래픽 툴로 작성한 뒤 작성된 그림 파일 상에서 링크의 활용율을 색깔로써 구분을 해준다. 전체적인 망 구성도에 링크의 활용율이 표시되므로 관리자가 망 상태를 확인하기 용이하게 해준다.

2. 트래픽 감시를 위한 네트워크의 구성

2.1 Monitoring Network의 모델링

트래픽 분산에 앞서 수행되어야 할 트래픽의 감시에 있어서 관리자는 관리하기 위한 망의 범주를 설정하도록 한다. 그리하여 관리할 망에 대해서 인터넷에서의 표준적인 망관리 프로토콜인 SNMP를 이용하여 망의 부하를 감시한다.

감시를 통해 발견된 트래픽의 부하에 대해서는 OSPF 망의 특성에 맞게 동적인 제어를 하여 트래픽 부하 분산을 하도록 한다. 이러한 트래픽 감시나 트래픽 분산의 결과 등은 관리자가 웹 브라우저를 통해 접근 가능하도록 한다. 그림 3은 이에 따른 시스템 모델이다. 그림과 같이 관리자는 웹을 통해 망을 관리할 수 있으며 관리대상에 해당하는 것은 관리자가 관리할 망 내에서 SNMP가 지원이 되는 라우터와 허브가 된다. Traffic Monitoring/Controlling System은 이러한 관리대상이 되는 망 구성요소에 대해 SNMP MIB 값을 이용한 감시 및 제어를 수행하게 된다.



(그림 3) 시스템 구성도

2.2 Monitoring Tool을 이용한 트래픽 감시

트래픽 감시를 위한 도구로써 본 논문에서는 MRTG(Multi Router Traffic Grapher)를 이용하였다. MRTG는 SNMP를 이용하여 라우터의 인터페이스별 트래픽의 양, 에러율, CPU 활용율 등을 그래프로 표현해주는 웹 기반의 Monitoring Tool이다. 이러한 툴을 이용하여 관리망 요소에서 MIB 값을 가져와 log 파일에 기록하고, log 파일에 기록된 데이터를 기반으로 그래프를 그려 HTML 문서로 보여준다. 측정된 인터페이스별 트래픽의 양에 기반한 링크의 이용율을 전체 망 구성도에 나타내어 관리자는 트래픽 부하의 정도를 이미지를 통해 한눈에 알아볼 수 있게 된다.

이러한 MRTG의 Monitoring 기능을 이용하여 망의 트래픽 상태를 감시하다가 링크의 이용율이 미리 정해진 Threshold 값을 넘게 되면 해당하는

인터페이스의 정보를 Controlling Module로 넘겨주게 된다.

3. 동적 트래픽 부하 분산

3.1 트래픽 부하 분산을 위한 자원의 정의와 분산 요건

트래픽 부하의 분산을 위해 관리 망을 구성하는 링크들을 감시하여 그 링크들의 과부하 여부를 판단한 후 과부하인 링크가 발생되면 분산을 행하도록 한다. 각 링크의 트래픽 부하 정도를 판단하기 위해서는 적용할 기준이 필요한데, 이를 위해서 Delay, Round Trip time, Link Utilization과 같은 요소들이 지표로 이용될 수 있다. 본 논문에서는 MRTG에서 SNMP MIB 값을 이용하여 측정할 수 있는 링크의 활용율을 트래픽 부하 측정의 지표로 삼았다. 링크의 활용율은 링크의 대역폭에 대한 트래픽의 양으로써 나타낼 수 있는데, 이 두 가지 모두 MIB값을 기반으로 추출할 수 있다.

링크의 활용율에 대해 각 링크의 특성에 따른 Threshold 값을 정하여 과부하를 판정하도록 하는데 정해진 Threshold 값을 넘는 링크에 대해서는 과부하로 판정을 내리고 분산을 시키도록 한다. 이 때 Threshold 값을 하나만 설정해 놓으면 Threshold 값 부근에서 링크 이용율이 진동할 경우 이에 대한 부하 분산 작업을 수시로 하게 된다. 그러므로 Threshold는 Gauge 타입으로 정의하며, Low Threshold와 High Threshold 값을 따로 두어 링크의 이용율이 Low Threshold를 넘어설 때는 무시하고 이용율이 High Threshold를 넘는 경우 과부하 판정을 내린다. 또한 과부하 링크의 이용율이 High Threshold를 내려갈 때는 무시하고 Low Threshold 값보다 작아지면 과부하 링크에 대한 Metric 복원 작업을 수행한다. 이 때 Threshold 값은 링크 종류에 따라 다르게 결정되는데 과부하를 판정하기 위해 보편적으로 사용되는 High Threshold 값은 FDDI, Ethernet, Serial link의 세가지로 구분하여 다음과 같이 정하도록 하고 있다.

- FDDI : 동일한 FDDI에 연결되어 있는 각 라우터 포트의 Total In/Out byte를 계산하여 결정한다. 일반적으로 FDDI는 Full-Duplex 방식으로 전송하므로 계산된 최대 In/Out bytes 중에서 높은 쪽의 사용율을 기준으로 약 90% 이상이 될 때 해당 네트워크는 부하가 있다고 본다. (Threshold = 90%)
- Ethernet : Broadcast 방식의 전송을 하므로, In/Out 트래픽을 합하여서 최대 사용율을 결정한다. 일반적으로 최대 사용율이 40% 이상이거나, Error Rate가 1% 이상이면 해당 Ethernet Segment에 대한 실질적인 원인분석이 필요하다고 본다. (Threshold = 40%)
- Serial Link : T1과 같은 Serial Link는 Full-Duplex 전송방식을 사용하므로, 해당 포트의 In/Out 트래픽 사용율 중 높은 쪽을 기준으로 최대 사용율을 결정한다. 일반적으로 최대 사용율이 90% 이상이면 부하가 있다고 본다. 특히, Serial-link에서 error율이 1% 이상이고, 주기적으로 발생할 경우는 Line 불량, 공중망을 통한 연결접점수 과다, 장비노후 등의 물리적인 원인으로 Error rate가 발생할 수도 있다. (Threshold = 90%)

3.2 네트워크 라우팅 정책에 맞는 부하 분산 규칙(Rule)의 정의

트래픽 조절을 동적으로 하기 위해서는 적용할 분산 규칙을 미리 정의해 두어야 한다. 분산을 위해 본 논문에서는 OSPF의 ECMP(Equal Cost MultiPath) 라우팅 기능을 이용하도록 한다. 과부하인 링크가 발생되면 해당 링크를 경유하는 트래픽 중 정의한 규칙에 맞는 것을 선택하여 해당 트래픽의 경로에 대해 과부하 링크를 거치지 않는 대체 경로로 트래픽이 분산되도록 한다. 즉 과부하 링크를 경유하는 경로와 이에 대한 대체 경로의 Cost 값을 동일하게 조정함으로써 OSPF의 ECMP 라우팅 기능에 의한 분산 메커니즘을 따른다. 이를 위해 과부하 링크의 트래픽 중 분산시킬 트래픽의 경로 선택과 선택한 경로의 Cost 값 조절을 위한 Metric 조정의 두 가지 단계로 규칙을 정의한다.

3.2.1 분산시킬 경로 선택

분산시킬 경로 선정을 위해 고려할 수 있는 사항으로 경로의 Cost, 활용 여유율을 들 수 있다. Cost 값은 Bandwidth에 기반하여 결정되므로 분산될 경로 선택시 Bandwidth를 고려한 결과를 가질 수 있다. 또한 활용 여유율의 고려로 망 자원의 사용을 분배할 수 있게 된다.

표 2는 과부하 링크의 부하를 분산시킬 경로를 선택하는 규칙이다.

<표 2> 분산 경로 결정 단계

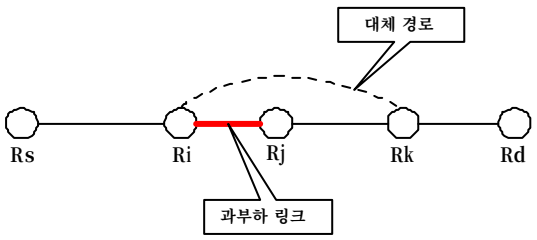
· 분산시킬 경로 선택

① 과부하 링크의 양 끝에 있는 두 개의 라우터에 대한 각각의 SPT(Shortest Path Tree) 생성.
② 생성된 SPT 각각에 대해 과부하 링크를 경유하는 경로 추출
③ 과부하 링크의 양 끝에 있는 두 개의 라우터에 대해 과부하 링크를 제거했을 때의 SPT' 생성
④ 생성된 SPT'으로부터 ①, ②에서 구한 과부하 링크를 경유하는 경로에 대한 대체 경로 추출
⑤ 과부하 링크를 경유하는 경로와 대체 경로의 Cost 계산
⑥ 대체 경로의 활용 여유율 계산
⑦ 과부하 링크를 경유하는 경로와 대체경로의 Cost 값이 같지 않은 것 중 대체 경로의 활용 여유율이 가장 많은 것을 분산될 경로로 선정.
⑧ 택한 결과가 여러 개가 나오는 경우 과부하 링크를 경유하는 경로와 대체 경로의 Cost 값의 차이가 적은 것을 분산시킬 경로로 택한다.

과부하 링크가 발견되면 우선 해당 링크를 경유하는 트래픽 중 정의한 규칙에 맞는 것을 골라 다른 경로를 이용하여 분산시켜야 하므로 이를 위해 과부하 링크를 경유하는 트래픽들을 조사해야 한다.

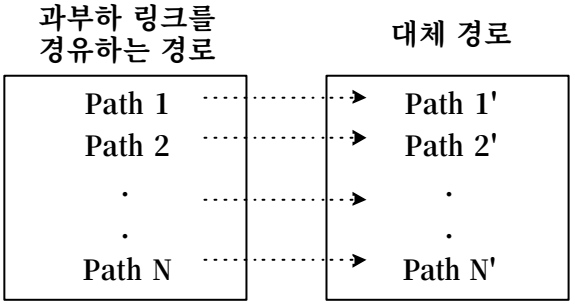
관리할 망이 OSPF 기반의 망이므로 OSPF에서 경로 결정을 위해 적용되는 Dijkstra 알고리즘을 이용하는 SPF(Shortest Path First) 방식에 근거하여 규칙 ①에서 SPT(Shortest Path Tree)를 구하고 규칙 ②에서 이 SPT를 기반으로 과부하 링크를 경유하는 트래픽을 추출한다. 이 때 추출할 트래픽은 반드시 과부하인 링크의 양 끝 노드를 지나므로 이 두 노드에서 SPT를 구하도록 한다. SPT를 통해 추출되는 경로는 다음과 같은 두가지 종류가 있다.

i) 과부하링크의 양 끝 노드 중 어느 하나에서 생성된 트래픽의 경로
   양 끝 노드에서 생성된 모든 트래픽의 경로를 추출할 수 있으므로 이에 대한 대체 경로의 모든 경우를 고려할 수 있다.
ii) 과부하 링크를 지나는 트래픽 경로의 하부경로(Subpath)



(그림 4) Rs에서 Rd로 가는 경로의 하부경로인 Ri에서 Rk로 가는 대체경로

그림 4와 같이 Rs에서 Rd로 가는 경로의 하부경로인 Ri에서 Rk로 가는 경로가 과부하 링크를 경유하는 경우, Ri의 SPT로부터 Rs에서 Rd로 가는 경로의 하부경로인 Ri에서 Rk로 가는 경로를 추출할 수 있고 이에 대한 대체 경로를 찾아 과부하 링크를 우회할 수 있다. 이러한 경우 과부하 링크를 만나는 지점에서 부분적 대체 경로를 찾아서 과부하 링크의 부하를 분산할 수 있게 된다.



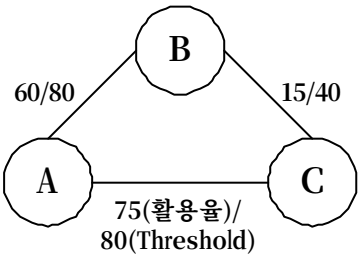
(그림 5) 과부하 링크를 경유하는 경로와 이에 대한 대체 경로의 대응

그 다음 규칙에서는 ①, ②에서 구한 과부하 링크를 지나는 경로들에 대해 과부하 링크를 경유하지 않고 전송할 수 있는 대체 경로를 구하도록 한다. 이러한 대체 경로는 트래픽을 분산시키기 위한 경로로 이용될 경로이다. 대체 경로를 구하기 위해서는 과부하 링크를 제거한 경우의 망 토폴로지에서, 앞서 구한 과부하 링크를 경유하는 경로의 양 끝단을 지나가는 경로를 찾으면 된다. 과부하 링크를 제거한 망 토폴로지에 대해 ③에서

와 같이 과부하 링크의 양 끝단의 두 노드에 대한 SPT’을 구하고 이를 기반으로 ④에서는 과부하 링크를 경유하는 경로에 대한 대체 경로를 추출한다. 이러한 경로와 대체 경로는 그림 5와 같이 대응되는 관계를 가진다.

앞에서 구한 과부하 링크를 경유하는 경로와 이에 대한 대체 경로를 이용하여 부하 분산을 하기 위해, 분산할 경로 선정에 지표로 삼을 경로의 Cost 값을 규칙 ⑤에서 계산한다. OSPF의 소개에서 언급한 바와 같이 경로의 Cost 값은 경로를 구성하는 각 링크의 Metric 값에 대한 합계로써 계산할 수 있다.

분산될 경로 선정에 경로의 Cost 값과 함께 고려될 수 있는 또 다른 지표는 경로의 활용 여유율이다. 트래픽 분산시 망 자원 사용의 분배를 위해 트래픽을 활용 여유율이 많이 남아있는 대체 경로로 분산하도록 한다. 대체 경로의 활용 여유율을 구하기 위해서는 대체 경로를 구성하는 링크의 활용 여유율을 먼저 계산해야 한다.



(그림 6) 경로의 활용 여유율 계산 예

경로의 활용 여유율을 구하는 예를 그림 6의 상황에서 보면, 그림 6과 같이 활용율과 Threshold가 결정되었을 때, 링크의 활용 여유율은 Threshold와 링크의 활용율의 차이로 결정하고 경로 A-B-C의 활용 여유율은 경로 A-B-C를 구성하는 링크인 A-B의 여유율 20과 B-C의 여유율 25를 비교하여 최소값인 20을 해당 경로의 활용 여유율로 정한다. 이와 같은 방법으로 대체 경로의 활용 여유율을 각각 계산하여 이를 분산될 경로 선정에 이용하도록 한다.

규칙 ⑥에서는 위의 예에서 적용했던 방법과 같이 각 링크의 활용 여유율을 Threshold와 링크의 활용율의 차이로 계산한 뒤, 경로를 구성하는 링크들의 활용 여유율 중 최소의 값을 가지는 경로의 활용 여유율로 정하도록 한다.

앞서 구한 과부하 링크를 경유하는 경로와 이에 대한 대체 경로, 그리고 이러한 경로들 각각에 대한 Cost 값, 대체 경로의 활용 여유율을 기반으로 하여 규칙 ⑦과 ⑧에서는 분산할 경로를 선정하도록 한다. 규칙 ⑦에서는 가장 많은 활용 여유율을 가진 대체 경로를 분산될 경로로 선택하도록 한다. 활용 여유율이 많은 곳으로 분산함으로써 자원의 효율적인 이용과 연쇄적인 과부하 발생의 방지를 할 수 있다.

그러나 규칙 ⑦을 적용할 때 과부하 링크를 경유하는 경로와 이에 대한 대체 경로의 Cost 값이 같은 것은 미리 선택에서 제외하도록 한다. Cost 값이 같은 경로는 OSPF의 ECMP 라우팅 메커니즘에 의해 이미 트래픽이 분산되어 전송되고 있으므로 이러한 경우는 선택하지 않도록 미리 제외한다.

규칙 ⑦의 결과가 여러 개가 나올 경우 미리 구한 Cost 값을 고려하여 분산될 경로를 선정하도록 한다. 규칙 ⑧에서는 규칙 ⑦의 적용 결과 중에서 경로와 대체 경로의 Cost 값의 차이가 작은 것을 분산시킬 경로로 선택한다. Cost 값은 Bandwidth에 기반하여 계산된 값이므로 Cost 값의 차이는 Bandwidth의 차이와 비례하게 된다. 그러므로 Cost 값의 차이에 따라 분산 경로를 결정하는 것은 Cost 값의 변경으로 트래픽을 분산시키는 메커니즘에서 Cost 값의 변경 폭이 크면 Bandwidth의 큰 차이로 인해 기존의 큰 Bandwidth로 가던 트래픽이 작은 Bandwidth로 분산되어 비효율적인 결과를 가지게 된다. 그러므로 Cost 값의 차이가 작은 경로를 선택하여 분산을 시켜 Bandwidth의 차이를 작게 한다.

이상과 같은 규칙에 따라 분산시킬 경로의 선택시 과부하 링크를 경유하는 경로와 이에 대한 대체 경로가 일대다(one-to-many)의 대응이 되는 경우, 즉 Cost 값이 같은 대체 경로가 여러 개인 경우가 발생할 수 있다. 이 경로가 분산시킬 경로로 선택되었을 경우 해당 대체 경로 모두로 분산을 시키도록 과부하 링크를 경유하는 경로와 대체 경로의 Cost 값 차이만큼 서로 다른 대체 경로 모두에 대해 Metric 조정 규칙에 따라 동일한 방식으로 조정을 한다.

이와 반대의 경우, 즉 과부하 링크를 경유하는 경로와 이에 대한 대체 경로가 다대일(many-to-one)의 대응이 되는 경우, 즉 양 끝단의 노드가 같은 과부하 링크를 지나는 경로가 하나의 대체 경로로 대응되는 경우가 발생할 수 있다. 이러한 경우 또한 과부하 링크를 경유하는 경로와 대체 경로의 Cost 값 차이만큼을 Metric 조정 규칙에 따라

동일한 방식으로 조정을 한다.

<표 3> Metric 조정 단계

· **Metric 조정**
① 선택된 과부하 링크를 경유하는 경로와 이에 대한 대체 경로와의 Cost 값의 차이 계산 ② 대체 경로의 구성 링크 중 활용 여유율이 가장 큰 링크의 Metric 값을 ①의 결과만큼 감소 ②' ②에서 Metric을 감소시킬 링크의 Metric이 1인 경우, 과부하 링크의 Metric 값을 ①의 결과만큼 증가

3.2.2 Metric 조정

분산시킬 경로 선택 규칙에 따라 선택된 경로를 이용하여 트래픽을 분산하기 위해서는 과부하 링크를 경유하는 경로와 이에 대한 대체 경로의 Cost 값을 동일하도록 조정하도록 해야 한다. 경로의 Cost 값은 경로를 구성하는 링크의 Metric의 값의 합계로써 결정되므로, 경로의 Cost를 조절하기 위해서 표 3과 같은 규칙에 따라 링크의 Metric 값을 조정하도록 한다.

트래픽을 분산할 분산 경로의 Cost 값을 같게 조정하도록 하기 위해 10우선 규칙 ①에서는 선택된 과부하 링크를 경유하는 경로와 이에 대한 대체 경로와의 Cost 값의 차이를 계산하도록 한다.

규칙 ①의 결과를 기반으로 규칙 ②에서는 대체 경로의 Cost 값을 하향 조정하여 두 경로의 Cost 값을 같도록 하는데, 경로의 Cost 값은 경로를 구성하는 링크의 Metric 값의 합으로 구하므로 경로를 구성하는 링크의 Metric 값을 조정하여 경로의 Cost 값을 조정하도록 한다. 대체 경로를 구성하는 링크들 중에서 가장 활용 여유율이 큰 링크의 Metric 값을 ①의 결과만큼 감소한다. 이로써 활용 여유율이 큰 링크로 트래픽을 유도하게 된다.

기본적으로 Metric 값은 음수가 아닌 정수의 값을 가진다. 그러나 규칙 ②에서와 같이 Metric 값을 감소하는 방법 사용시 조정 결과의 Metric 값이 0이나 음수가 나올 수가 있는데, Metric 값이 0인 경우는 직접 연결된 네트워크에 대한 것이고 음수는 Metric 값에 적합치 않으므로 이러한 경우를 피하기 위해 조정할 링크의 Metric 값이 1이면 규칙 ②를 실행하지 않고 규칙 ②'를 실행하도록 한다.

규칙 ②'에서는 과부하 링크를 지나는 경로의 Cost 값을 상향 조정하여 두 경로의 Cost 값을 같게 하는 방법을 취한다. 이 때 과부하 링크를 지나는 경로를 구성하는 링크 중 가장 링크의 활용 여유율이 작은 것은 과부하 링크가 되므로 과부하 링크의 Metric 값을 규칙 ①의 결과만큼 증가시키도록 한다. 이로써 과부하 링크의 트래픽을 다른 곳으로 분산·우회시킨다.

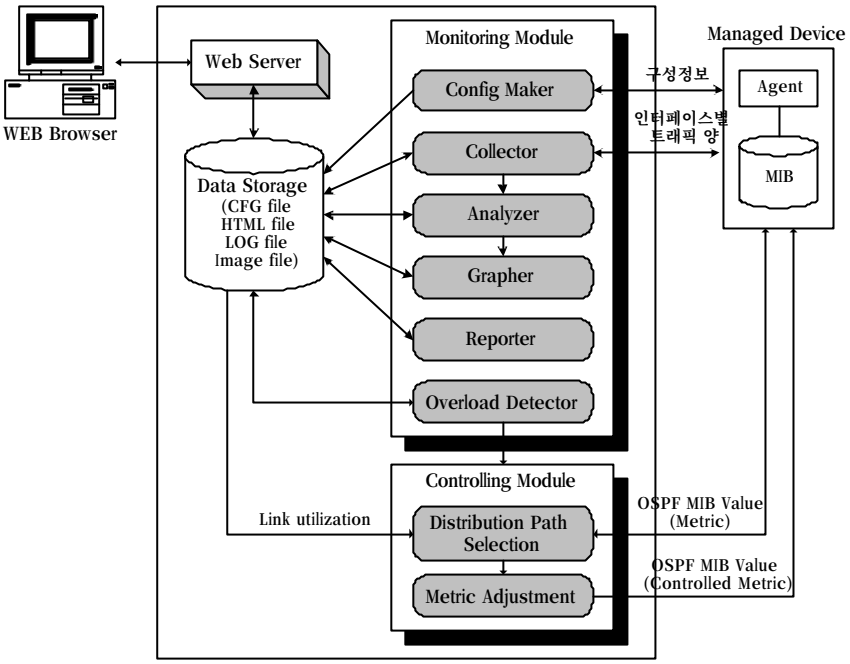
그러나 여기서 한가지 사항을 더 고려해 봐야 한다. 분산될 경로 선택시 그 결과가 여러 개인 경우가 발생할 수 있다. 이러한 경우, 대응되는 결과 각각에 대해 Metric 조정 규칙을 적용하도록 하는데 만약 규칙 ②를 실행하지 않고 규칙 ②'를 실행하도록 하는 경우가 발생하면 그러한 대응은 조정에서 제외한다. 왜냐하면 규칙 ②를 적용하는 경우는 대체 경로를 조정하여 Cost 값을 같게 만들어주지만 규칙 ②'는 과부하 링크의 Metric 값을 직접 바꿔주게 되어 모든 대응 결과에 영향을 주게 된다. 그러므로 분산될 경로 선택 결과가 여러 개인 경우는 마찰을 피하기 위해 Metric 조정 규칙 ②를 실행하는 것만을 조정하도록 한다.

3.3 SNMP를 이용한 자동화된 트래픽의 부하 분산

부하 분산 규칙에 근거하여 Metric을 조정할 링크가 선정되면 해당 링크에 대해서 RFC 1850의 OSPFv2 MIB을 이용하여 Metric 값을 조정하도록 한다. 표 4는 OSPF MIB에서 Metric 값을 조정할 수 있는 ospfIfMetricTable의 내용을 보여주고 있다. 이 중 인터페이스의 Metric 값을 나타내는 ospfIfMetricValue의 값을 조정함으로써 해당 Metric의 값을 변경할 수 있다. ospfIfMetricValue의 기본값은 "100,000,000/ifSpeed"이다.

<표 4> OSPF MIB에서의 ospfIfMetricTable

ospfIfMetricTable
- ospfIfMetricEntry
- ospfIfMetricIpAddress
- ospfIfMetricAddressLessIf
- ospfIfMetricTOS
- ***ospfIfMetricValue***
- ospfIfMetricStatus

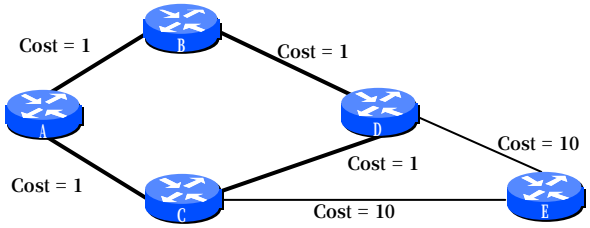


(그림 7) 트래픽 부하 분산을 위한 시스템 구성도

그림 7은 SNMP를 이용하여 부하 분산을 할 시스템의 간략한 구성도이다. 발견된 과부하 링크의 부하분산을 위해 Controlling module에서 링크들의 Metric 값을 알아내기 위해 라우터의 각 인터페이스들에 대해 OSPF MIB에서 ospfIfMetricValue를 가져온 후 규칙에 따라 부하 분산에 관련된 링크의 Metric을 조정하기 위해 ospfIfMetricValue에 조정 값을 SNMP의 SET request 메시지를 사용하여 변경하도록 한다.

4. 트래픽 조정의 실례

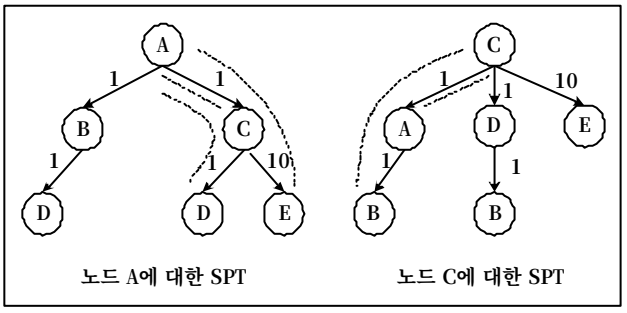
관리 망에서 트래픽의 과부하에 대한 처리 상황을 다음의 예로 보자.



(그림 8) Network Example

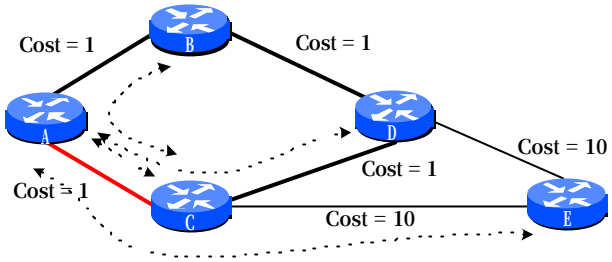
그림 8의 예와 같이 Cost 값이 결정되었을 때, 각각의 노드는 SPT(Shortest Path Tree)를 가지게 된다. 이것은 각각의 노드에서 목적지까지의 경로를 찾는 근거가 되며 각 링크의 Cost 값을 기반으로 만들게 된다.

만약 그림 8의 예에서 A-C 사이의 링크에서 과부하가 발생한 경우, 분산될 경로를 찾아내는 규칙 ①에 따라 먼저 A-C 링크를 지나는 트래픽 플로우를 알아내기 위해 A와 C를 root로 하는 SPT를 생성한다. 이는 그림 9와 같다.



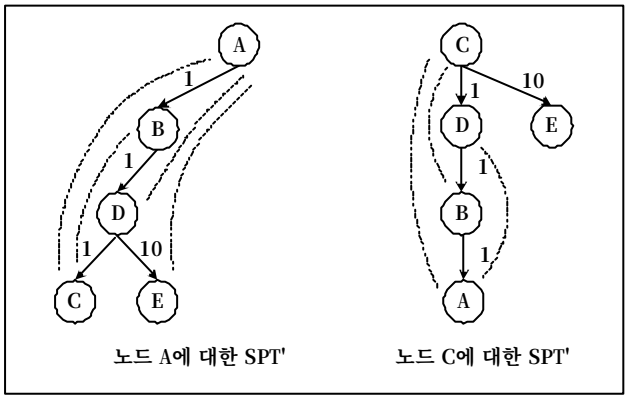
(그림 9) A와 C의 SPT(Shortest Path Tree)

그림 9에서 점선으로 표시한 것은 과부하 링크인 A-C를 지나는 경로를 표시한 것이다. 이것을 기반으로 A-C 링크를 지나는 트래픽 플로우의 경로를 추출하면 중복된 경로를 제외하고 A-C, A-C-D, A-C-E, C-A-B의 4가지의 경로가 생기게 된다. SPT를 기반으로 찾아낸 A-C사이의 트래픽 플로우를 네트워크 그래프에 점선으로 표현한 것이 그림 10이다.

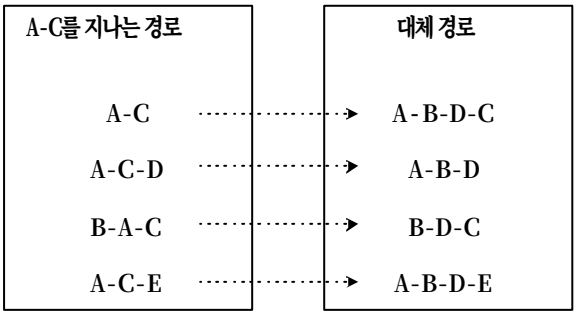


(그림 10) A-C 링크를 지나는 트래픽 플로우

찾아낸 경로 각각에 대해 과부하 링크를 지나지 않고 트래픽 플로우를 전송시킬 대체 경로를 찾기 위해 그림 11과 같이 과부하 링크를 제외한 망에서의 A와 C의 SPT'를 생성하고 이에 따라 대체 경로를 추출한다. 이러한 방법으로 찾아낸 경로와 대체 경로는 그림 12와 같다.

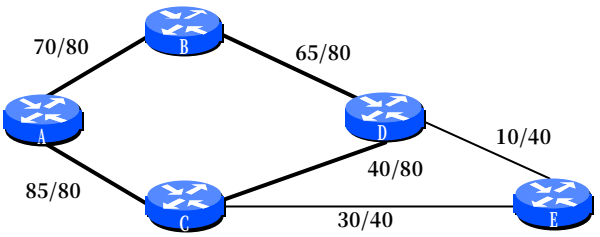


(그림 11) 대체 경로 추출을 위한 SPT' 생성



(그림 12) 경로와 대체 경로의 대응 결과

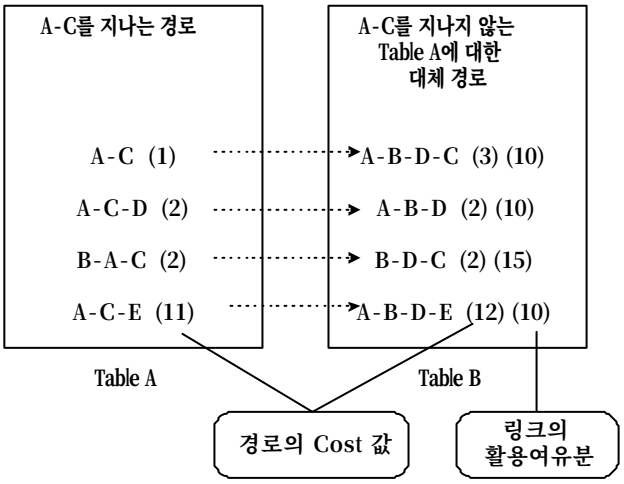
경로와 대체 경로를 찾아낸 후 각각의 Cost 값을 계산하고, 대체 경로 각각에 대해 링크의 여유율을 구한다. 링크의 여유율은 링크의 활용율과 Threshold 값의 차로써 구하는데 예로 든 네트워크의 링크의 활용율과 Threshold와 이를 기반으로 산출한 대체 경로의 활용 여유율을 나타낸 것이 그림 13이다.



경로의 활용 여유율
A-B-D-C : 10
A-B-D : 10
B-D-C : 15
A-C-E : 10

(그림 13) 각 링크의 활용율/Threshold과 활용 여유율

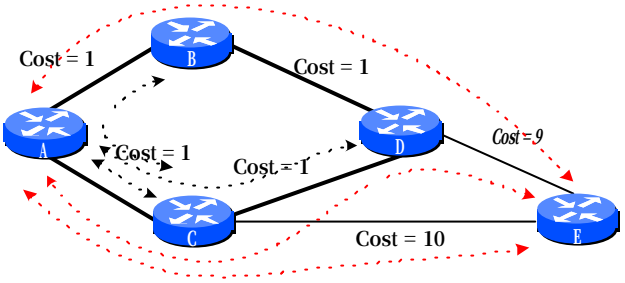
그림 14는 규칙에 따라 구해진 각 경로의 Cost 값과 대체 경로의 활용 여유율을 나타낸 테이블이다. 이 중 경로와 대체 경로의 Cost값이 같은 쌍인 A-C-D, A-B-D와 B-A-C, B-D-C를 제외하고 나머지 대응 결과들 중에서 대체 경로의 활용 여유율이 가장 많은 경로는 여유율이 10인 A-B-D-C, A-B-D-E가 된다. 이 중에서 경로와의 Cost 값의 차가 가장 적은 것은 A-B-D-E이므로 이것을 분산시킬 경로로 선정하도록 한다.



(그림 14) 분산될 경로 찾기

분산시킬 경로 선택 규칙에 따라 선택된 경로로 트래픽을 분산시키기 위해 A-C-E의 트래픽을 A-B-D-E에 분산할 수 있도록 두 경로의 Cost를

동일하게 맞춰 주도록 한다. 이를 위해 Metric 조정 규칙에 따라 A-B-D-E의 Cost를 경로의 Cost와의 차이인 1만큼 감소시킨다. 이 때 대체 경로의 구성 링크의 활용 여유율을 따져서 가장 링크 활용 여유율이 가장 많은 D-E의 Metric을 1만큼 감소시킨다. 이로써 트래픽은 아래의 그림 15와 같이 분산된다.



(그림 15) 분산된 결과

부하분산 시스템에 의해 A-C의 트래픽 플로우중 A-C-E를 경유하는 트래픽 플로우는 A-C-E와 A-B-D-E, A-C-D-E로 분산되어 기존에 A-C 링크를 이용하여 가던 A-C-E의 트래픽 중 1/3이 감소하게 된다.

5. Metric 값의 복원

조정한 Metric 값은 링크가 과부하 상태일 때 이를 해결하기 위한 방법으로 변경한 것이다. 그러므로 과부하 상태가 해소되면 원래의 초기 Metric 값으로 복원해 줘야 한다. 초기 Metric 값은 Metric 값 조정 이전에 미리 따로 저장해두고 과부하 상태의 회복 시에 이 값으로 Metric 값을 복원해 준다.

이를 위해 Metric 값의 회복시점은 Low Threshold로 알아내도록 한다. 과부하 판정시 링크의 이용율이 High Threshold 값을 넘는지의 여부로 판단하는 것과 같이 회복시점은 링크의 이용율을 Low threshold와 비교하여 판정한다. 링크의 이용율이 Low Threshold 값을 넘을 때나 과부하 링크의 이용율이 High Threshold 값보다 작아질 때는 이를 알리지 않고, 과부하 링크의 이용율이 Low Threshold보다 작아지거나 링크 이용율이 High Threshold보다 커지면 과부하 혹은 과부하 해소 상황을 시스템에 알려주도록 한다. High Threshold 보다 링크 이용율이 커지면 이는 과부하 상태로 인지하고 이에 따라 분산 메커니즘으로 대응하고, 링크 이용율이 Low Threshold보다 작아지면 이를 회복시기로 인지하고 Metric 값의 복원을 하도록 한다. 이와 같이 Threshold 값을 두개로 설정함으로써 하나의 Threshold로 판정시 일어날 수 있는 진동을 방지할 수 있다.

Low Threshold 또한 High Threshold와 같이 링크의 이용율로서 나타내며 이는 Monitoring Tool에서 과부하 링크의 이용율을 감시하다가 이것이 Low Threshold 값보다 작아지면 조정 링크의 Metric을 원래의 값으로 SNMP 메시지를 이용하여 복원해 주도록 한다. 앞서 Metric 조정시 사용한 OSPFv2의 MIB에 정의되어있는 ospfIfMetricValue를 이용하여 Metric 값을 복원시킨다.

6. 결론

인터넷상의 트래픽 양의 증가로 인해 기존의 인터넷망의 성능에 대한 효율적인 관리가 요구됨에 따라 망 성능의 저하에 대처하기 위해 이에 대한 원인 분석 및 개선을 위한 망의 성능 감시 및 제어가 필요하게 되었다. 망 성능이 급격히 저하되어 전송의 지연이나 패킷 손실이 일어나는 것을 방지하기 위해서 현재 대부분의 ISP들은 트래픽의 부하가 많은 곳의 트래픽을 부하량이 낮은 다른 경로로 분산시켜서 망의 성능 저하를 막는 접근 방식을 인터넷상의 트래픽 관리에 사용하고 있다.

그러나 기존의 방법은 관리자의 측면에서 대단히 비효율적이므로 이러한 단점들을 극복하기 위해서 본 논문에서는 현재의 트래픽 상태를 웹 기반의 망관리 시스템이 감시하여 트래픽의 과부하가 일어나는 것을 자동으로 감지한 후 이에 대한 적합한 조치를 SNMP를 이용하여 OSPF MIB에서 Metric 조절로 트래픽을 분산시키도록 하였다. 이러한 동적인 트래픽 부하 분산으로 망의 트래픽 관리를 위한 관리자의 부담을 덜어주고 급변하는 망 내의 트래픽 과부하 상황을 동적으로 제어할 수 있도록 하였다.

향후 연구 과제로서 OSPF OMP(Optimized Multipath)에 따른 트래픽 균등 분할에 대해 연구하여 전체 네트워크의 부하 분산을 하기 위한 효율적인 방안과 효과에 관한 연구와 함께 본 논문에서 제안하는 방식의 트래픽 부하 분산 시스템을 발전시켜 나가도록 하며 성능평가를 위한 실험이 병행되어야 할 것이다.

[참 고 문 헌]

[1] Douglas E. Comer, "Internetworking with TCP/IP", Prentice Hall, 1995

[2] W. Richard Stevens, "TCP/IP Illustrated, Volume 1", Addison-Wesley, 1994

[3] William Stallings, "SNMP, SNMPv2, SNMPv3, and RMON 1 and 2", Addison-Wesley, 1996

[4] Andrew S. Tanenbaum, "Computer Networks", Prentice Hall, 1996

[5] Thomas M. Thomas Ⅱ, "OSPF Network Design Solutions", Cisco Press, 1998

[6] 장우현, "BGP에서 Traffic 제어", APAN-KR & WIO6 워크샵, 1999

[7] J. Won-Ki Hong, S. U. Park, Y. M. Kang and J. T. Park, "Enterprise Network Traffic Monitoring, Analysis and Reporting Using Web Technology", Accepted to appear in the Journal of Network and Systems Management, Plenum Press, 2000.

[8] IETF RFC 2328, "OSPF Version 2", April 1998

[9] IETF RFC 1850, "OSPF Version 2 Management Information Base", Nov. 1995

[10] IETF Internet-Draft, "OSPF Optimized Multipath (OSPF-OMP)", Feb. 1999

[11] http://inw.webpd.co.kr/cisco.html

[12] http://www.ietf.org/html.charters/ospf-charter.html

[13] http://www.whchang.com/

[14] http://myhome.shinbiro.com/~son6971/sub/sub33/ospf.htm

[15] http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/mrtg.html

[16] http://amazon.postech.ac.kr/MRTG

김 희 진

1997.8 전남대학교 전산학 학사

1998 – 현재 전남대학교 전산통계학 석사과정

관심분야 : 네트워크 관리, 라우팅, 인터넷 관리

박 상 근

1996.8 전남대학교 전산학 학사

1999.8 전남대학교 전산통계학 석사

2000 – 현재 전남대학교 전산학 박사과정

관심분야 : 네트워크 관리, 서비스 관리, TMN, WBEM

최 덕 재

1982 서울대학교 컴퓨터공학 학사

1984 한국과학기술원 전산학 석사

1995 Univ. of Missouri-Kansas 컴퓨터 망 박사

1984-1996 전남대학교 전자계산소 운영부장

1997-1998 전자통신연구원 통신시스템 연구단 위탁연구원

1996-1998 전남대학교 정보화 기획단 상임연구원

1996. 5 - 현재 전남대학교 전산학과 부교수

1998. 9 - 현재 개방형컴퓨터 통신연구회 Application 분과위원장

관심분야 : 네트워크 관리, 서비스 관리, TMN, 차세대 인터넷