

SNMP MIB-II를 이용한 인터넷 분석 파라미터계산 알고리즘에 관한 연구

안 성 진[†] · 정 진 옥^{††}

요 약

본 논문에서는 TCP/IP 프로토콜을 기반으로 하는 인터넷에서 SNMP의 MIB-II를 활용하여 분석 파라미터를 정의하고 이를 계산하는 알고리즘을 제안하고자 한다. TCP/IP 망의 사용자에게 적절한 QoS를 제공하기 위해서는 성능과 장애에 관련된 파라미터를 기반으로 한 망 관리 행위를 수행해야 한다. 이를 위해서 인터넷 관리 표준으로 정의된 MIB-II의 관리 정보를 기반으로 분석 파라미터를 정의하고 이를 계산하기 위한 알고리즘을 제시하고자 한다. MIB-II에서 system, interface, ip, snmp 그룹의 관리 변수를 Case 다이어그램에 따라 분석하여 선로 이용률, 에러 수신율, 인터페이스 패킷 송수신율, 인터페이스 패킷 송수신 손실률, 입출력 트래픽률, 방송형 송수신 트래픽 비율, 시스템 패킷 입출력률, 시스템 패킷 송수신 손실률, 시스템 자원 부하율, 패킷 전달률, 경로 설정 실패율, 관리 트래픽 이용률 등의 분석 파라미터와 계산 알고리즘을 제안한다. 분석 파라미터 계산 알고리즘에 대한 적용성을 실험하기 위해서 실존하는 라우터를 대상으로 분석 결과를 제시하고 진단하였다. 이러한 인터넷 분석 파라미터 계산 알고리즘은 망 관리자가 전체 TCP/IP 통신 네트워크를 진단하고 분석할 수 있는 자료로 활용될 수 있을 뿐만 아니라 인터넷 사용자에게 QoS를 제공할 수 있을 것으로 기대된다.

A Study on the Algorithms for Calculating Internet Analysis Parameters using SNMP MIB-II

Seong Jin Ahn[†] · Jin Wook Chung^{††}

ABSTRACT

In this paper, algorithms are proposed to define and calculate analysis parameters using SNMP MIB-II on TCP/IP based Internet. To provide a user on TCP/IP network with QoS it is required that management activities are based on performance and fault related parameters. Analysis parameters by MIB-II, which is Internet standard, are defined and algorithms to calculate these parameters are proposed. In MIB-II, system, interface, ip, and snmp groups are analyzed according to Case diagrams and then analysis parameters, which are line utilization, error rate in link, I/O packet rate of a interface, I/O packet loss rate of a interface, I/O traffic, I/O nonunicast packet rate, I/O packet rate of a system, I/O packet loss rate of a system, resource load rate of a system, packet forwarding rate, routing fault rate, and utilization of management traffic, and its calculation algorithms are provided. To show availability of these algorithms, some results are shown and analyzed about routers on real environments. These algorithms for calculating Internet analysis parameters can give effects to analysis and diagnosis of a TCP/IP network for a manager and also, it is expected to provide QoS for Internet users.

[†] 종신회원 : 성균관대학교 전기 전자 및 컴퓨터공학부 시간강사

^{††} 종신회원 : 성균관대학교 전기 전자 및 컴퓨터공학부 교수

논문접수 : 1998년 3월 30일, 심사완료 : 1998년 6월 9일

1. 서 론

TCP/IP 프로토콜을 사용하고 있는 컴퓨터와 통신망, 그리고 분산 시스템들이 양적인 측면 외에 복잡성 면에서도 증가하고 있다. 반면에 끊임없이 발전하고 다양화되는 하부 망 기술과 상위 응용에 대한 관리에 있어서 어려움을 겪고 있는 것도 사실이다. 더욱이, WWW(World Wide Web) 서비스의 등장과 이의 활성화로 인해 통신망에서의 트래픽 증가가 급격히 일어나고 있다.[1] 따라서, 인터넷 관리자는 대규모화되고 복잡해져 가는 통신 자원의 관리 문제와 사용자에 대한 QoS(Quality of Service) 제공 문제가 절실한 과제로 부각되고 있다.

일반적으로 통신 장비에 대한 관리를 수행하기 위해서 인터넷에서는 SNMP(Simple Network Management Protocol)를 사용하고 있다. SNMP는 관리 시스템과 피관리 시스템간에 관리 정보를 교환하기 위해서 사용되는 프로토콜이다. 이에 대한 개선은 현재 SNMPv2의 표준화로 보안성과 통신 행위에서 발생하는 중복 PDU(Protocol Data Unit)의 제거 등을 지원하고 있다.[2][3] SNMP에 의해서 교환되는 정보는 MIB(Management Information Base)이라는 관리 정보 집합을 사용하도록 하고 있다. 관리 정보 집합인 MIB은 해당 피관리 시스템에 대한 정보 수집의 결정적 단서가 되는 것으로 이를 기반으로 하여 모든 망 관리 시스템들이 개발되고 있다.[4] TCP/IP 프로토콜을 기반으로 하는 인터넷에서는 MIB-II(5)에 의해서 통신 장비에 대한 관리 행위를 수행하도록 요구하고 있다. 따라서, Win95를 탑재한 PC를 비롯하여 라우터, 스위치, 허브 등이 MIB-II(Win95는 MIB-I)를 제공하고 있다. 이와 같이 인터넷 관리란 관리 정보를 유통시키는 프로토콜에 해당하는 SNMP와 통신 장비에서 어떤 관리 정보를 갖고 있는가 하는 MIB에 대한 두 가지 관점을 갖는다.

인터넷 관리자가 수행해야 할 주요한 작업 중 하나는 망에 대한 분석을 수행하여 결과를 망 설계나 확장 등에 반영하는 것이다. 이는 사용자에게 QoS를 향상시키는 결과를 가져올 것이다. 이를 위해서는 인터넷에 대한 분석 파라미터를 정의하고 이를 구하기 위한 모델과 알고리즘이 제시되어야 한다. 관리 정보에 대한 확장으로는 대표적인 것이 RMON MIB이다. RMON MIB은 LAN의 분석을 위한 통계량과 관리 변수를 제공하

고 있으며, 그 외에 트래픽 분석을 위한 MIB(ATM 스위치를 위한 MIB 등이 정의되고 있다. 관리 정보의 수집과 계산 수식에 대해서는 논문(6)과 (7)에서 제시하고 있으나, 이를 구체적으로 처리하기 위한 알고리즘이 아직 제시되고 있지 않고 있다. 논문(8)에서는 수학적 모델을 도입하여 처리량과 응답 시간 등의 해석을 할 수 있는 모델을 각 프로토콜 별로 제시하고 있다. 그러나, 인터넷 표준 관리 정보를 기반으로 한 것은 아니다. 논문(9)에서는 응용 서비스 관리를 위한 모델과 분석을 수행하는 시스템을 구현하고 있다. ATM의 VPC/VCC 관리 기법은 논문(10)에서, 폴링 방식에 따른 특성 비교는 논문(11)에서 제시하고 있다. RMON MIB을 이용한 분석 수식은 논문(12)에서 제안하고 있다.

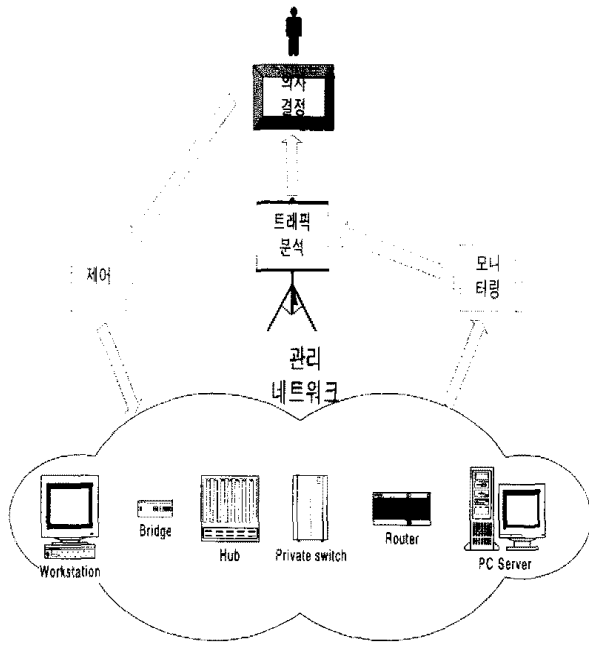
이 논문에서는 인터넷 관리를 위해 MIB-II 관리 변수에서 분석 파라미터를 정의하고 이를 계산하기 위한 모델과 알고리즘을 제안하고자 한다. 이는 망 관리 시스템에서 인터넷 QoS 관련 자료를 추출하여 망 운영/관리에 반영할 것으로 기대된다.

2. 분석 파라미터

현재 사용하는 망에 대한 현황을 파악하고 이를 개선시키기 위해서는 다음 그림 1과 같은 순환 구조의 관리 행위가 필요하다. 그림 1에서와 같이 망에 속한 피관리 시스템들로부터 관리 정보를 모니터링이란 행위로 수집하여 트래픽 분석을 시행하게 된다. 트래픽 분석은 통상 수집된 MIB-II 변수들에 대해 성능과 장애에 대한 분석을 의미한다. 분석을 통해 망의 자원에 대한 문제점과 현황을 파악하고 이를 토대로 의사 결정 과정을 수행하여 임의의 피관리 시스템에 제어를 가하게 된다. 이 과정에서 가장 문제가 되는 것은 어떤 관리 정보를 모니터링할 것인가 하는 것과 이를 어떻게 분석해야 하는가 하는 문제가 있다. MIB-II는 175개의 관리 변수를 갖고 있어서 이들 중 어떤 관리 변수를 추출해야 망 분석과 진단에 유용한 지를 선택하기가 곤란하다. 더구나, 관리 정보는 통상 정적인 상태 정보로서 이를 가공해야 의미 있는 정보가 된다. 따라서, 관리 정보의 분석을 수행하는 알고리즘이 필요하다.

2.1 유통 트래픽 분석

유통 트래픽 분석은 옥테트 트래픽 유통량을 기준으



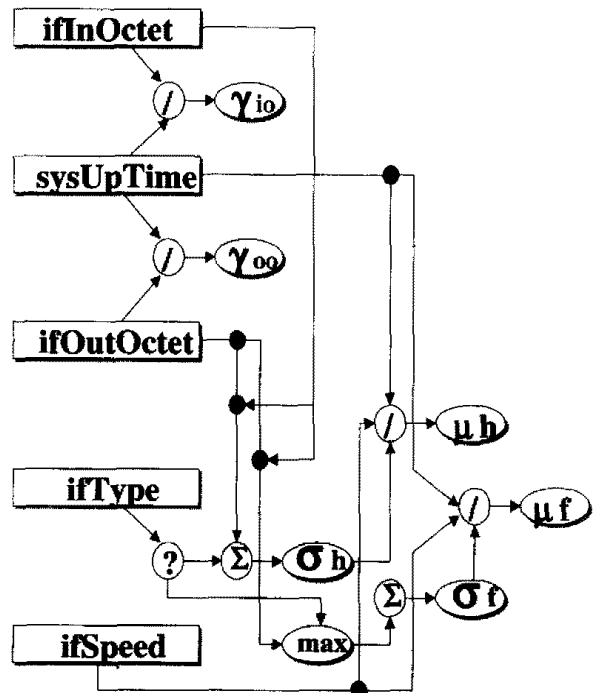
(그림 1) 망 분석 작업
(Fig. 1) Network Analysis Task

로 입출력 트래픽률과 선로 이용률을 계산하는 분석 영역이다. 이때, 필요한 MIB-II 변수는 ifInOctet, ifOutOctet, sysUpTime, ifType, ifSpeed로서 Case 다이어그램(2)[13]을 기반으로 제시되었으며 이들에 대한 계산 모델은 그림 2와 같다.

γ_{io} (입력 트래픽률)를 구하기 위해서는 ifInOctet와 sysUpTime 관리 정보를 추출하여야 하며, γ_{oo} (출력 트래픽률)를 구하기 위해서는 ifOutOctet와 sysUpTime 관리 정보를 구해야 한다. γ_{io} 와 γ_{oo} 를 사용하여 입출력 트래픽률을 비교하여 분석할 수도 있다. 선로 이용률은 ifType에 따라 단방향/반이중 전송 선로인 경우는 μ_h 를, 전이중 전송 선로인 경우는 μ_f 를 구해야 한다. 선로 이용률을 구하는 알고리즘에서는 ifSpeed 값을 가져와서 사용할 수 없다. 이는 통신 장비에서 사용하는 ifSpeed값의 정확성에 문제가 있기 때문이다. 통신 장비 관리자가 구성 정보로 ifSpeed값을 정확하게 입력하는 경우, 알고리즘에서 이 값을 사용할 수 있으나 현실적으로 선로의 현재 속도를 설정하지 않으므로 사용자로부터 입력을 받아서 수행해야 한다.

2.2 인터페이스 유통 패킷 분석

인터페이스 유통 패킷 분석은 인터페이스로 유출입 되는 패킷을 기준으로 에러 수신율, 인터페이스 패킷

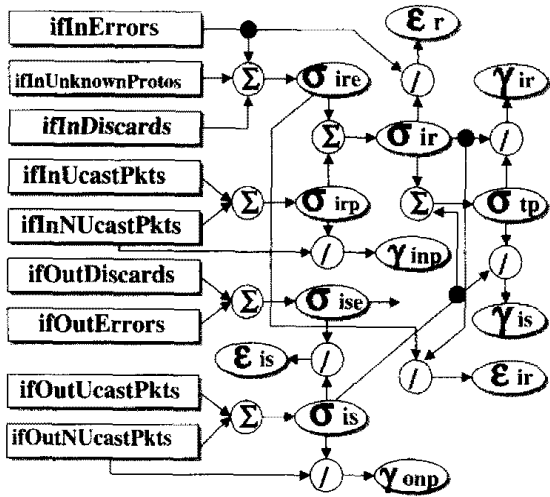


(그림 2) 유통 트래픽 분석 계산 모델
(Fig. 2) Calculation Model for I/O Traffic Analysis

송수신율, 인터페이스 패킷 송수신 손실률, 방송형 송수신 트래픽 비율 등을 계산하는 분석 영역이다. 이때, 필요한 MIB-II 변수는 ifInErrors, ifInUnknownProtos, ifInDiscards, ifInUcastPkts, ifInNUcastPkts, ifOutDiscards, ifOutErrors, ifOutUcastPkts, ifOutNUcastPkts로서 Case 다이어그램(2)[13]을 기반으로 제시되었으며 이들에 대한 계산 모델은 그림 3과 같다.

ϵ_r (에러 수신율)을 구하기 위해서는 입력되는 모든 패킷의 합인 σ_{ir} 에 대한 ifInErrors의 율을 구한다. ϵ_{is} (인터페이스 패킷 송신 손실률)은 σ_{is} (인터페이스 패킷 총 송신)에 대한 σ_{ise} (인터페이스 패킷 총 송신 손실)로 나타난다. 아울러, ϵ_{ir} (인터페이스 패킷 수신 손실률)은 σ_{ir} (인터페이스 패킷 총 수신)에 대한 σ_{ire} (인터페이스 패킷 총 수신 손실)로 나타난다. γ_{ir} (인터페이스 패킷 수신율)은 총 유통 패킷인 σ_{tp} 에 대한 σ_{ir} 로 표시된다. γ_{is} (인터페이스 패킷 송신율)은 총 유통 패킷인 σ_{tp} 에 대한 σ_{is} 로 표시된다. 방송형 트래픽은 멀티캐스트와 브로드캐스트 트래픽을 모두 의미한다. 이들에 대한 트래픽 비율을 구하기 위해서는 γ_{inp} (방송형 수신 트래픽 비율)과 γ_{onp} (방송형 송신 트래픽 비율)을 구해야 한다. 이들은 입력 비방송형 트래픽 및

입력 방송형 트래픽의 합에 대한 입력 방송형 트래픽 비율과 출력 비방송형 트래픽 및 출력 방송형 트래픽의 합에 대한 출력 방송형 트래픽 비율을 구하는 것으로 계산된다.



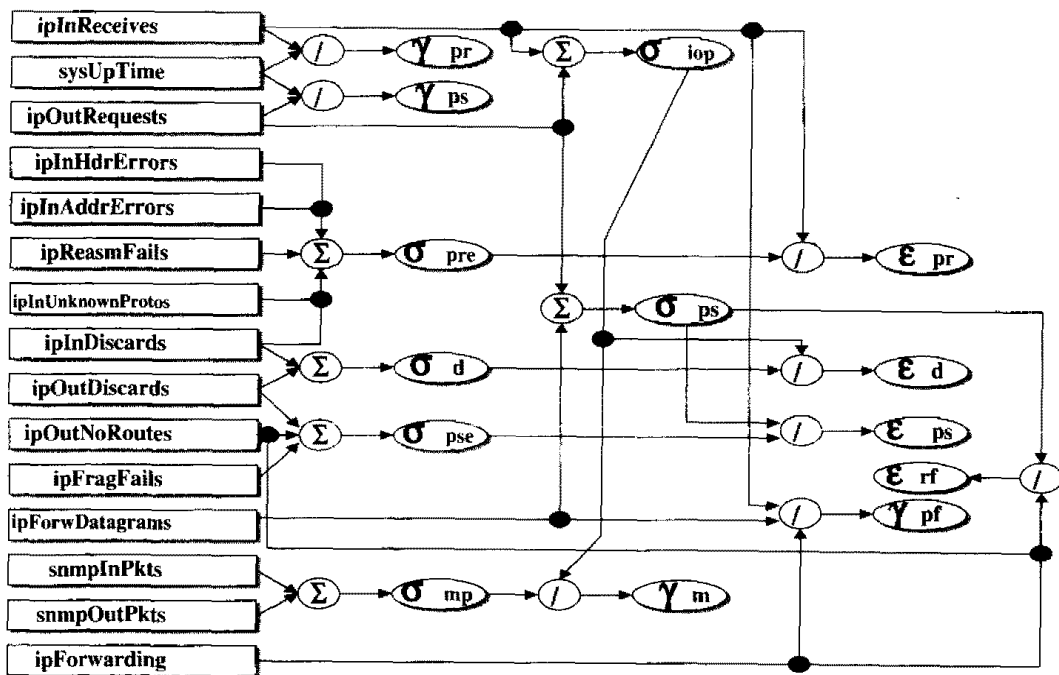
(그림 3) 인터페이스 유통 패킷 분석 계산 모델
(Fig. 3) Calculation Model for Interface I/O Packet Analysis

2.3 패킷 트래픽 및 관리 트래픽 분석

패킷 트래픽 및 관리 트래픽 분석은 IP 계층에서 유출입되는 패킷을 기준으로 시스템 패킷 입출력률, 시스

템 패킷 송수신 손실률, 시스템 자원 부하율, 패킷 전달률, 경로 설정 실패율, 관리 트래픽 이용률 등을 계산하는 분석 영역이다. 이때, 필요한 MIB-II 변수는 ipInReceives, sysUpTime, ipOutRequests, ipInHdrErrors, ipInAddrErrors, ipReasmFails, ipInUnknownProtos, ipInDiscards, ipOutDiscards, ipOutNoRoutes, ipFragFails, ipForwDatagrams, snmpInPkts, snmpOutPkts로서 Case 다이어그램 (2)[13]을 기반으로 제시되었으며 이들에 대한 계산 모델은 그림 4와 같다.

γ_{pr} (시스템 패킷 입력률)과 γ_{ps} (시스템 패킷 출력률)은 통신 장비에서 sysUpTime을 기반으로 하여 단위 시간당 유통되는 패킷의 율을 나타낸다. 이때, 발생하는 손실은 ϵ_{pr} (시스템 패킷 수신 손실률)과 ϵ_{ps} (시스템 패킷 송신 손실률)로 정의된다. 이들은 송수신시의 손실합을 나타내는 σ_{pre} 및 σ_{pse} 와 송수신 패킷 트래픽으로 계산된다. 시스템 자원 부하율(ϵ_d)과 경로 설정 장애율(ϵ_{ps})은 메모리 부족과 같은 자원의 부족 현상과 경로 설정 테이블 상에 존재하지 않는 패킷으로 발생하는 원인만을 표시하기 위해 사용된다. γ_{pr} (패킷 전달률)은 수신 패킷에 대한 전달 패킷의 율로 계산된다. γ_m (관리 트래픽 이용률)은 전체 패킷에 대한 SNMP 트래픽의 발생률로 나타낼 수 있다.



(그림 4) 패킷 트래픽 및 관리 트래픽 분석 계산 모델
(Fig. 4) Calculation Model for Packet and Management Traffic Analysis

3. 알고리즘

그림 2에서 그림 4까지 제시된 계산 모델을 기반으로 계산 알고리즘을 작성할 수 있다. 이때, 각 계산 모델에 따른 알고리즘이 정의될 수 있으므로 유동 트래픽, 인터페이스 유동 패킷, 패킷 트래픽 및 관리 트래픽 분야로 나누어서 제시되어야 한다.

3.1 유동 트래픽 분석 계산 알고리즘

(1) 선로 이용률

인터넷을 관리하기 위한 기본적인 행위가 바로 선로의 이용률 계산이다. 통신 장비 사이의 선로에 대한 이용률은 선로의 증속 계획 등과 같이 비용과 직결되어 있어서 망 운영자에게 가장 중요한 관심 사항중 하나이다. 선로 이용률을 계산하기 위해서는 선로의 형태를 알아야 한다. 선로의 형태는 단방향/반이중/전이중으로 구분된다. 단방향과 반이중의 경우는 송수신 트래픽의 합과 시간이 필요하며, 전이중의 경우는 송수신 트래픽을 비교하여 최대값과 시간으로 계산된다. 전이중 통신 회선의 경우, 송수신 트래픽이 동일시간에 동시에 발생할 수 있기 때문이다. 따라서, 알고리즘에서도 이를 ifType으로 비교하여 계산한다. 선로 이용률 계산을 위한 알고리즘은 다음과 같다.

단계 1) 인덱스 번호 추출

ipAdEntIfIndex를 이용하여 IP 주소에 대한 인터페이스 인덱스 번호 값 I_{if} 획득한다.

단계 2) 초기 MIB 변수 폴링

관련 MIB 변수 집합 $V\{ifInOctet, ifOutOctet, sysUpTime, ifSpeed, ifType\}$ 를 I_{if} 를 이용하여 폴링한다.

단계 3) 기준 값과 링크 형태 설정

μ_{max} 에 ifSpeed를 설정하고 L_{type} 에 ifType값을 설정한다.

단계 4) 비교 값 설정

B_{io} 에 ifInOctet, B_{oo} 에 ifOutOctet, 그리고 B_t 에 sysUpTime을 설정한다.

단계 5) 다음 MIB 변수 폴링

다음 폴링 변수 집합 $V\{ifInOctet, ifOutOctet, sysUpTime\}$ 을 I_{if} 를 이용하여 t_p 시간 간격으로 폴링한다.

단계 6) 링크 형태 비교

L_{type} 의 값이 단 방향 통신 링크 집합 S에 속하거나 반 이중 통신 링크 집합 D_h에 속하는 경우는 (단계 7)로 가서 σ_h 를 계산하고, 전 이중 통신 링크 집합 D_f에 속하는 경우는 (단계 8)로 가서 σ_f 를 계산한다.

단계 7) 단 방향/반 이중 통신 선로에 대한 바이트 양 누적

인터페이스에 유출입되는 바이트 양을 식 (1)에 따라 누적한다.

$$\begin{aligned} \delta_{hi}(i) &= ifInOctet(i) - B_{io} \\ \delta_{ho}(i) &= ifOutOctet(i) - B_{oo} \\ \sigma_h &= \sum_{i=1}^n (\delta_{hi}(i) + \delta_{ho}(i)) \end{aligned} \quad (1)$$

단계 8) 전 이중 통신 선로에 대한 바이트 양 누적
각 i번째 폴링에 대해 인터페이스에 유출입되는 바이트 양 중에서 최대치를 식 (2)에 따라 누적한다.

$$\begin{aligned} \delta_{fi}(i) &= ifInOctet(i) - B_{io} \\ \delta_{fo}(i) &= ifOutOctet(i) - B_{oo} \\ \sigma_f &= \sum_{i=1}^n \max(\delta_{fi}(i) + \delta_{fo}(i)) \end{aligned} \quad (2)$$

단계 9) 시간 변화량 누적

각 i번째 폴링에 대해 지나온 시간 값을 식 (3)에 따라 누적하고 종료 시에는 (단계 10)으로 가고 아니면, (단계 4)로 이동한다.

$$\begin{aligned} \delta_{tp}(i) &= sysUpTime - B_t \\ \sigma_t &= \sum_{i=1}^n \delta_{tp}(i) \end{aligned} \quad (3)$$

단계 10) 선로 이용률 계산

L_{type} 의 값이 S나 D_h에 속하는 경우는 식 (4)로 계산하고, D_f에 속하는 경우는 식 (5)를 계산한다.

$$\mu_h = \frac{\sigma_h \delta}{\sigma_t \mu_{max}} \quad (4)$$

$$\mu_f = \frac{\sigma_f \delta}{\sigma_t \mu_{max}} \quad (5)$$

(2) 입출력 트래픽률

입출력 트래픽률은 단위 시간당 인터페이스에서 입출력되는 트래픽의 비율이다. 전체 송신 혹은 수신 바이트 양과 sysUpTime을 추출하여 각각에 대한 바이트 트래픽률을 구하여 송수신을 비교할 수 있다. 이는 인터페이스 입출력 패킷 송수신률과 같이 인터페이스를 통해 송수신되는 패킷의 율을 나타내는 것과는 달리 실제 유출입되는 바이트 양을 표시하기 때문에 인터페이스 입출력 패킷 송수신률에 비해 더 명확하게 인터페이스의 송수신 양을 알 수 있다. 그리고, 송신과 수신 비율을 비교함으로써 각 피관리 시스템이 현재 외부로부터 받아오는 양이 많은지 제공하는 양이 많은지 판단할 수 있다. 이와 같은 자료는 ISP(Internet Service Provider)들에게 특히 중요하다. 이에 대한 알고리즘은 다음과 같다.

단계 1) 인덱스 번호 추출

ipAdEntIfIndex를 이용하여 IP 주소에 대한 인터페이스 인덱스 번호 값 I_{if} 획득한다.

단계 2) 초기 MIB 변수 폴링

관련 MIB 변수 집합 V(ifInOctet, ifOutOctet, sysUpTime)를 I_{if}을 이용하여 폴링한다.

단계 3) 비교값 설정

B_{io} = ifInOctet, B_{oo} = ifOutOctet, B_t = sysUpTime을 설정한다.

단계 4) 다음 MIB 변수 폴링

다음 폴링 변수 집합 V를 I_{if}를 이용하여 t_p 시간 간격으로 폴링한다.

단계 5) 변화량 누적

$$\begin{aligned} \delta_{if}(i) &= \text{ifInOctet} - B_{io} \\ \delta_{oo}(i) &= \text{ifOutOctet} - B_{oo} \\ \delta_{tp}(i) &= \text{sysUpTime} - B_t \end{aligned}$$

단계 6) 입출력 트래픽 율 계산

종료 시면 식 (6)과 식 (7)에 따라 계산을 수행하고 아니면 (단계 3)으로 간다.

$$\sigma_{io} = \sum_{i=1}^{n_t} \delta_{if}(i)$$

$$\sigma_{oo} = \sum_{i=1}^{n_t} \delta_{oo}(i)$$

$$\gamma_{io} = \frac{\sigma_{io}}{\sigma_t} \tag{6}$$

$$\gamma_{oo} = \frac{\sigma_{oo}}{\sigma_t} \tag{7}$$

3.2 인터페이스 유통 패킷 분석 계산 알고리즘

(1) 인터페이스 패킷 송수신율

인터페이스 패킷 송수신율은 인터페이스에 유출입되는 전체 패킷에 대한 송신/수신 패킷의 율을 나타내주는 파라미터이다. 이 분석 파라미터는 인터페이스의 관점에서 받아들인 패킷과 상위 계층에서 전송한 패킷으로 입출력률을 구한다. 관리자는 인터페이스 패킷 송수신율 파라미터를 통해서 한 인터페이스에 유출입되는 패킷의 비율을 알 수 있다. 아울러, 이를 가공하면, 단위 시간당 유출입되는 패킷을 계산할 수 있다. 이에 대한 알고리즘은 다음과 같다.

단계 1) 인덱스 번호 추출

ipAdEntIfIndex를 이용하여 IP 주소에 대한 인터페이스 인덱스 번호 값 I_{if} 획득한다.

단계 2) 초기 MIB 변수 폴링

관련 MIB 변수 집합 V(ifInErrors, ifInUnknownProtos, ifInDiscards, ifInUcastPkts, ifInNUcastPkts, ifOutUcastPkts, ifOutNUcastPkts)를 I_{if}을 이용하여 폴링한다.

단계 3) 비교값 설정

B_{ie} = ifInErrors, B_{iu} = ifInUnknownProtos, B_{id} = ifInDiscards, B_{iup} = ifInUcastPkts, B_{inp} = ifInNUcastPkts, B_{oup} = ifOutUcastPkts, B_{onp} = ifOutNUcastPkts를 설정한다.

단계 4) 다음 MIB 변수 폴링

다음 폴링 변수 집합 V를 I_{if}를 이용하여 t_p 시간 간격으로 폴링한다.

단계 5) 변화량 누적

$$\delta_{ie}(i) = \text{ifInErrors} - B_{ie}$$

$$\begin{aligned} \delta_{iu}(i) &= \text{ifInUnknownProtos} - B_{iu} \\ \delta_{iid}(i) &= \text{ifInDiscards} - B_{iid} \\ \delta_{iup}(i) &= \text{ifInUcastPkts} - B_{iup} \\ \delta_{inp}(i) &= \text{ifInNUcastPkts} - B_{inp} \\ \delta_{oup}(i) &= \text{ifOutUcastPkts} - B_{oup} \\ \delta_{onp}(i) &= \text{ifOutNUcastPkts} - B_{onp} \end{aligned}$$

단계 6) 인터페이스 패킷 송수신율 계산
중요할 경우는 식 (8)과 식(9)에 따라 계산을 수행하고 아니면 (단계 3)으로 간다.

$$\sigma_{is} = \sum_{i=1}^{n_s} (\delta_{onp}(i) + \delta_{oup}(i))$$

$$\begin{aligned} \sigma_{tp} = \sum_{i=1}^{n_s} & (\delta_{iie}(i) + \delta_{iu}(i) + \delta_{iid}(i) \\ & + \delta_{iup}(i) + \delta_{inp}(i) + \delta_{oup}(i) \\ & + \delta_{onp}(i)) \end{aligned}$$

$$\gamma_{ir} = \frac{\sigma_{ir}}{\sigma_{tp}} \quad (8)$$

$$\gamma_{is} = \frac{\sigma_{is}}{\sigma_{tp}} \quad (9)$$

(2) 방송형 송수신 트래픽 비율

방송형 송수신 트래픽 비율은 입출력 패킷 중에서 입출력 각각에 대한 방송형(Broadcast/Multicast) 패킷의 비율을 보여주는 분석 파라미터이다. 비방송형(Unicast) 트래픽은 수신 시스템과 송신 시스템 사이에 일대일로 전송이 되는 형태이지만 방송형 트래픽은 어떤 특정한 목적지에 도달하기 위해서 망 내의 불특정 다수의 시스템으로 패킷이 전송되고 각 시스템이 이 방송형 패킷을 받아서 상위 계층에 전달하는 형태의 트래픽이다. 그러므로, 관리자는 방송형 트래픽을 발생할 응용이 존재하지 않는데 이러한 트래픽이 있다면 이를 확인하여 망의 비효율성을 제거해야 한다. 이 파라미터를 구하는 계산 알고리즘은 다음과 같다.

단계 1) 인덱스 번호 추출

ipAdEntIfIndex를 이용하여 IP 주소에 대한 인터페이스 인덱스 번호 값 I_{if} 획득한다.

단계 2) 초기 MIB 변수 폴링

관련 MIB 변수 집합 V{ifInUcastPkts, ifInNUcastPkts, ifOutUcastPkts, ifOutNUcastPkts}를 I_{if}를 이용하여 폴링한다.

단계 3) 비교값 설정

B_{iup} = ifInUcastPkts, B_{inp} = ifInNUcastPkts, B_{oup} = ifOutUcastPkts, B_{onp} = ifOutNUcastPkts를 설정한다.

단계 4) 다음 MIB 변수 폴링

다음 폴링 변수 집합 V를 I_{if}를 이용하여 t_p 시간 간격으로 폴링한다.

단계 5) 변화량 누적

$$\begin{aligned} \delta_{iup}(i) &= \text{ifInUcastPkts} - B_{iup} \\ \delta_{inp}(i) &= \text{ifInNUcastPkts} - B_{inp} \\ \delta_{oup}(i) &= \text{ifOutUcastPkts} - B_{oup} \\ \delta_{onp}(i) &= \text{ifOutNUcastPkts} - B_{onp} \end{aligned}$$

단계 6) 방송형 송수신 트래픽 비율 계산

중요 시면 식 (10)과 식 (11)에 따라 계산을 수행하고 아니면 (단계 3)으로 간다.

$$\sigma_{irp} = \sum_{i=1}^{n_s} (\delta_{iup}(i) + \delta_{inp}(i))$$

$$\gamma_{inp} = \frac{\sum_{i=1}^{n_s} \delta_{inp}(i)}{\sigma_{irp}} \quad (10)$$

$$\gamma_{onp} = \frac{\sum_{i=1}^{n_s} \delta_{onp}(i)}{\sigma_{is}} \quad (11)$$

(3) 인터페이스 패킷 송수신 손실률

인터페이스 패킷 송수신 손실률은 인터페이스에서 손실되는 송신/수신 패킷의 율을 나타내주는 파라미터이다. 인터페이스 패킷 손실률은 전체 패킷과 손실 패킷의 비율로부터 구할 수 있다. 여기서 손실 패킷은 인터페이스에서 에러가 발견되어 버려지는 패킷과 에러가 발견되지 않았지만 버퍼의 부족과 같은 이유로 폐기되는 패킷, 상위 계층 프로토콜을 인식하지 못하여 제거되는 패킷을 의미한다. 이에 대한 계산 알고리즘은 다음과 같다.

단계 1) 인덱스 번호 추출

ipAdEntIfIndex를 이용하여 IP 주소에 대한 인터페이스 인덱스 번호 값 I_{if} 획득한다.

$$\varepsilon_{is} = \frac{\sigma_{ise}}{\sigma_{is}} \quad (13)$$

단계 2) 초기 MIB 변수 폴링

관련 MIB 변수 집합 V (ifInErrors, ifInUnknownProtos, ifInDiscards, ifInUcastPkts, ifInNUcastPkts, ifOutUcastPkts, ifOutNUcastPkts, ifOutDiscards, ifOutErrors)를 I_{if} 을 이용하여 폴링한다.

단계 3) 비교값 설정

$B_{iie} = \text{ifInErrors}$, $B_{iu} = \text{ifInUnknownProtos}$, $B_{iid} = \text{ifInDiscards}$, $B_{iup} = \text{ifInUcastPkts}$, $B_{inp} = \text{ifInNUcastPkts}$, $B_{oup} = \text{ifOutUcastPkts}$, $B_{onp} = \text{ifOutNUcastPkts}$, $B_{iod} = \text{ifOutDiscards}$, $B_{ioe} = \text{ifOutErrors}$ 를 설정한다.

단계 4) 다음 MIB 변수 폴링

다음 폴링 변수 집합 V 를 I_{if} 를 이용하여 t_p 시간 간격으로 폴링한다.

단계 5) 변화량 누적

$$\begin{aligned} \delta_{iie}(i) &= \text{ifInErrors} - B_{iie} \\ \delta_{iu}(i) &= \text{ifInUnknownProtos} - B_{iu} \\ \delta_{iid}(i) &= \text{ifInDiscards} - B_{iid} \\ \delta_{iup}(i) &= \text{ifInUcastPkts} - B_{iup} \\ \delta_{inp}(i) &= \text{ifInNUcastPkts} - B_{inp} \\ \delta_{oup}(i) &= \text{ifOutUcastPkts} - B_{oup} \\ \delta_{onp}(i) &= \text{ifOutNUcastPkts} - B_{onp} \\ \delta_{iod}(i) &= \text{ifOutDiscards} - B_{iod} \\ \delta_{ioe}(i) &= \text{ifOutErrors} - B_{ioe} \end{aligned}$$

단계 6) 인터페이스 패킷 송수신 손실을 계산

종료 시면 식 (12)과 식 (13)에 따라 계산을 수행하고 아니면 (단계 3)으로 간다.

$$\begin{aligned} \sigma_{ire} &= \sum_{i=1}^{n_s} (\delta_{iid}(i) + \delta_{iu}(i) + \delta_{iie}(i)) \\ \sigma_{ise} &= \sum_{i=1}^{n_s} (\delta_{ioe}(i) + \delta_{iod}(i)) \\ \varepsilon_{ir} &= \frac{\sigma_{ire}}{\sigma_{ir}} \quad (12) \end{aligned}$$

(4) 에러 수신율

에러 수신율은 원격지 시스템으로부터 유입되는 프레임 등의 에러에 의해서 상위 계층 프로토콜로 전송되지 못하는 수신 패킷의 수를 의미한다. 따라서, 전체 수신 패킷에 대한 에러 패킷의 수를 비율로 나타냄으로써 에러 수신율을 구할 수 있다. 이 파라미터는 선로의 품질을 평가하는 기준이 될 수 있으며 응답 시간 개선에 기여할 수 있다. 에러 수신율 ε_r 을 구하기 위한 알고리즘은 다음과 같다.

단계 1) 인덱스 번호 추출

ipAdEntIfIndex를 이용하여 IP 주소에 대한 인터페이스 인덱스 번호 값 I_{if} 획득한다.

단계 2) 초기 MIB 변수 폴링

관련 MIB 변수 집합 V (ifInErrors, ifInUnknownProtos, ifInDiscards, ifInUcastPkts, ifInNUcastPkts)를 I_{if} 을 이용하여 폴링한다.

단계 3) 비교값 설정

$B_{iie} = \text{ifInErrors}$, $B_{iu} = \text{ifInUnknownProtos}$, $B_{iid} = \text{ifInDiscards}$, $B_{iup} = \text{ifInUcastPkts}$, $B_{inp} = \text{ifInNUcastPkts}$ 를 설정한다.

단계 4) 다음 MIB 변수 폴링

다음 폴링 변수 집합 V 를 I_{if} 를 이용하여 t_p 시간 간격으로 폴링한다.

단계 5) 변화량 누적

$$\begin{aligned} \delta_{iie}(i) &= \text{ifInErrors} - B_{iie} \\ \delta_{iu}(i) &= \text{ifInUnknownProtos} - B_{iu} \\ \delta_{iid}(i) &= \text{ifInDiscards} - B_{iid} \\ \delta_{iup}(i) &= \text{ifInUcastPkts} - B_{iup} \\ \delta_{inp}(i) &= \text{ifInNUcastPkts} - B_{inp} \end{aligned}$$

단계 6) 에러 수신율 계산

종료할 경우는 식 (14)에 따라 계산을 수행하고 아니면 (단계 3)으로 간다.

$$\sigma_{is} = \sum_{i=1}^{n_s} (\delta_{inp}(i) + \delta_{imp}(i) + \delta_{inc}(i) + \delta_{in}(i) + \delta_{iid}(i))$$

$$\epsilon_{ir} = \frac{\sum_{i=1}^{n_s} \delta_{iic}(i)}{\sigma_{ir}} \quad (14)$$

3.3 패킷 트래픽 및 관리 트래픽 분석 계산 알고리즘

(1) 시스템 패킷 입출력률

시스템 패킷 입출력률은 단위 시간당 시스템(라우터, 스위치 등)에 입출력되는 IP 패킷 트래픽의 율이다. 이 파라미터를 통하여 관리자는 전체 인터페이스에서 입출력된 패킷 양에 대한 율을 파악할 수 있다. 인터페이스 패킷 입출력률은 전체 시스템에 입출력된 패킷이 아니라 한 인터페이스에 입출력된 패킷의 양을 보여주는 파라미터이다. 따라서, 인터페이스 패킷 송수신 율과는 다른 의미를 갖고 있다. 이에 대한 알고리즘은 다음과 같다.

단계 1) 초기 MIB 변수 폴링

관련 MIB 변수 집합 V(ipInReceives, ipOutRequests, sysUpTime)를 I_{if}를 이용하여 폴링한다.

단계 2) 비교값 설정

B_{pr} = ipInReceives, B_{por} = ipOutRequests, B_t = sysUpTime을 설정한다.

단계 3) 다음 MIB 변수 폴링

다음 폴링 변수 집합 V를 I_{if}를 이용하여 t_p 시간 간격으로 폴링한다.

단계 4) 변화량 누적

$$\delta_{pr}(i) = ipInReceives - B_{pr}$$

$$\delta_{por}(i) = ipOutRequest - B_{por}$$

$$\delta_{tp}(i) = sysUpTime - B_t$$

단계 5) 시스템 패킷 입출력 율 계산

종료할 경우는 식 (15)와 식 (16)에 따라 계산을 수행하고 아니면 (단계 2)로 간다.

$$\sigma_{pr} = \sum_{i=1}^{n_s} \delta_{pr}(i)$$

$$\sigma_{ps} = \sum_{i=1}^{n_s} \delta_{por}(i)$$

$$\gamma_{pr} = \frac{\sigma_{pr}}{\sigma_t} \quad (15)$$

$$\gamma_{ps} = \frac{\sigma_{ps}}{\sigma_t} \quad (16)$$

(2) 패킷 전달률

패킷 전달률은 수신 패킷에 대해서 경로 설정에 관여하는 패킷의 율을 나타내는 파라미터이다. 따라서, MIB-II 변수 중 ipForwarding이 1인 경우에만 측정 가능한 파라미터이다. ipForwarding값이 1이라는 것은 해당 엔티티가 라우터나 게이트웨이와 같은 기능을 수행한다는 의미이다. 이에 대한 알고리즘은 다음과 같다.

단계 1) 경로 설정 기능 동작 확인

ipForwarding 변수를 폴링하여 값이 1이면 (단계 2)로 가고 아니면 종료한다.

단계 2) 초기 MIB 변수 폴링

관련 MIB 변수 집합 V(ipInReceives, ipForwDatagrams)를 I_{if}를 이용하여 폴링한다.

단계 3) 비교값 설정

B_{pr} = ipInReceives, B_{pf} = ipForwDatagrams을 설정한다.

단계 4) 다음 MIB 변수 폴링

다음 폴링 변수 집합 V를 I_{if}를 이용하여 t_p 시간 간격으로 폴링한다.

단계 5) 변화량 누적

$$\delta_{pr}(i) = ipInReceives - B_{pr}$$

$$\delta_{pf}(i) = ipForwDatagrams - B_{pf}$$

단계 6) 패킷 전달률 계산

종료할 경우는 식 (17)에 따라 계산을 수행하고 아니면 (단계 3)으로 간다.

$$\gamma_{pf} = \frac{\sum_{i=1}^{n_s} \delta_{pf}(i)}{\sigma_{pr}} \quad (17)$$

(3) 시스템 패킷 송수신 손실률

시스템 패킷 송수신 손실률은 시스템의 IP 계층에서 송수신시 손실되는 패킷의 율을 나타내주는 파라미터이다. 이 파라미터는 특정 인터페이스에서 손실되는 패킷의 율을 나타내는 인터페이스 패킷 손실률과는 달리 전체 시스템(라우터, 스위치 등)에서 손실되는 패킷의 율을 나타내는 파라미터이다. 송신 시스템 패킷 손실률은 전체 패킷과 송신 손실 패킷(자원 부족, 경로 설정 에러, 단편화 실패 등)의 비율로 구할 수 있다. 수신 시스템 패킷 손실률은 전체 패킷과 수신 손실 패킷(헤더 에러, 주소 에러, 조립 에러, 자원 부족, 알려지지 않은 프로토콜 등)의 비율로 구할 수 있다.

단계 1) 초기 MIB 변수 풀링

관련 MIB 변수 집합 V (ipInReceives, ipInHdrErrors, ipInAddrErrors, ipReasmFails, ipInDiscards, ipInUnknownProtos, ipOutRequests, ipFragOks, ipFragCreates, ipOutDiscards, ipOutNoRoutes, ipFragFails, ipForwDatagrams, sysUpTime)를 I_{if} 을 이용하여 풀링한다.

단계 2) 비교값 설정

$B_{pr} = ipInReceives$, $B_{phe} = ipInHdrErrors$, $B_{pae} = ipInAddrErrors$, $B_{prf} = ipReasmFails$, $B_{pid} = ipInDiscards$, $B_{pup} = ipInUnknownProtos$, $B_{por} = ipOutRequests$, $B_{pfo} = ipFragOks$, $B_{pfc} = ipFragCreates$, $B_{pod} = ipOutDiscards$, $B_{pon} = ipOutNoRoutes$, $B_{pff} = ipFragFails$, $B_{pfd} = ipForwDatagrams$, $B_t = sysUpTime$ 을 설정한다.

단계 3) 다음 MIB 변수 풀링

다음 풀링 변수 집합 V 를 I_{if} 을 이용하여 t_{i+1} 시간 간격으로 풀링한다.

단계 4) 변화량 누적

$$\begin{aligned}\delta_{pr}(i) &= ipInReceives - B_{pr} \\ \delta_{phe}(i) &= ipInHdrErrors - B_{phe} \\ \delta_{pae}(i) &= ipInAddrErrors - B_{pae} \\ \delta_{prf}(i) &= ipReasmFails - B_{prf} \\ \delta_{pid}(i) &= ipInDiscards - B_{pid}\end{aligned}$$

$$\begin{aligned}\delta_{pup}(i) &= ipInUnknownProtos - B_{pup} \\ \delta_{por}(i) &= ipOutRequest - B_{por} \\ \delta_{pfo}(i) &= ipFragOks - B_{pfo} \\ \delta_{pfc}(i) &= ipFragCreates - B_{pfc} \\ \delta_{pod}(i) &= ipOutDiscards - B_{pod} \\ \delta_{pon}(i) &= ipOutNoRoutes - B_{pon} \\ \delta_{pff}(i) &= ipFragFails - B_{pff} \\ \delta_{pfd}(i) &= ipForwDatagrams - B_{pfd} \\ \delta_{tp}(i) &= sysUpTime - B_t\end{aligned}$$

단계 5) 시스템 패킷 송수신 손실률 계산

종료할 경우는 식 (18)과 식 (19)에 따라 계산을 수행하고 아니면 (단계 2)로 간다.

$$\begin{aligned}\sigma_{prc} &= \sum_{i=1}^{n_s} (\delta_{phe}(i) + \delta_{pae}(i) + \delta_{prf}(i) \\ &\quad + \delta_{pid}(i) + \delta_{pup}(i))\end{aligned}$$

$$\sigma_{psc} = \sum_{i=1}^{n_s} (\delta_{pod}(i) + \delta_{pon}(i) + \delta_{pff}(i))$$

$$\epsilon_{pr} = \frac{\sigma_{prc}}{\sigma_{pr}} \quad (18)$$

$$\epsilon_{ps} = \frac{\sigma_{psc}}{\sigma_{ps} + \sum_{i=1}^{n_s} \delta_{pf}(i)} \quad (19)$$

(4) 시스템 자원 부하율

시스템 자원 부하율은 시스템의 송수신 패킷에 대해서 발생하는 대기 패킷의 율로 나타낸다. 따라서, 송수신 패킷의 부하에 따라 메모리나 성능의 문제로 인해 발생하는 시스템의 부하를 측정하는 파라미터로 사용할 수 있다. 이에 대한 알고리즘은 다음과 같다.

단계 1) 초기 MIB 변수 풀링

관련 MIB 변수 집합 V (ipInReceives, ipOutRequests, ipFragOks, ipFragCreates, ipInDiscards, ipOutDiscards, ipRoutingDiscards, sysUpTime)를 I_{if} 을 이용하여 풀링한다.

단계 2) 비교값 설정

$B_{pr} = ipInReceives$, $B_{por} = ipOutRequests$, $B_{pfo} = ipFragOks$, $B_{pfc} = ipFragCreates$, $B_{pid} = ipInDiscards$, $B_{pod} = ipOutDiscards$, B_{rd}

· ipRoutingDiscards, B_{ir} = sysUpTime을 선정한다.

단계 3) 다음 MIB 변수 폴링

다음 폴링 변수 집합 V를 I_{ir}를 이용하여 t_p 시간 간격으로 폴링한다.

단계 4) 변화량 누적

$$\begin{aligned} \delta_{pr}(i) &= ipInReceives - B_{pr} \\ \delta_{por}(i) &= ipOutRequest - B_{por} \\ \delta_{pfo}(i) &= ipFragOks - B_{pfo} \\ \delta_{pfc}(i) &= ipFragCreates - B_{pfc} \\ \delta_{pid}(i) &= ipInDiscards - B_{pid} \\ \delta_{pod}(i) &= ipOutDiscards - B_{pod} \\ \delta_{rid}(i) &= ipRoutingDiscards - B_{rid} \\ \delta_{tr}(i) &= sysUpTime - B_t \end{aligned}$$

단계 5) 시스템 자원 부하율 계산

종료할 경우는 식 (20)에 따라 계산을 수행하고 아니면 (단계 2)로 간다.

$$\sigma_d = \sum_{i=1}^{n_s} (\delta_{pid}(i) + \delta_{pod}(i) + \delta_{rid}(i))$$

$$\epsilon_d = \frac{\sigma_d}{\sigma_{pr} + \sigma_{ps}} \quad (20)$$

(5) 경로 설정 실패율

경로 설정 실패율은 시스템에서 전송하려는 패킷 중에서 경로 설정 테이블에 없음으로 인해 경로 설정을 수행하지 못하는 패킷의 율을 나타내는 파라미터이다. 이는 경로 설정 기능을 수행하는 게이트웨이나 라우터에 대한 분석을 수행할 수 있는 파라미터이다. 이에 대한 알고리즘은 다음과 같다.

단계 1) 경로 설정 기능 동작 확인

ipForwarding 변수를 폴링하여 값이 1이면 (단계 2)로 가고 아니면 종료한다.

단계 2) 초기 MIB 변수 폴링

관련 MIB 변수 집합 V(ipForwDatagrams, ipOutRequests, ipOutNoRoutes)를 I_{ir}를 이용하여 폴링한다.

단계 3) 비교값 설정

B_{pr} = ipForwDatagrams, B_{por} = ipOutRequests, B_{pnr} = ipOutNoRoutes를 설정한다.

단계 4) 다음 MIB 변수 폴링

다음 폴링 변수 집합 V를 I_{ir}를 이용하여 t_p 시간 간격으로 폴링한다.

단계 5) 변화량 누적

$$\begin{aligned} \delta_{por}(i) &= ipOutRequest - B_{por} \\ \delta_{pr}(i) &= ipForwDatagrams - B_{pr} \\ \delta_{pnr}(i) &= ipOutNoRoutes - B_{pnr} \end{aligned}$$

단계 6) 경로 설정 실패율 계산

종료할 경우는 식 (21)에 따라 계산을 수행하고 아니면 (단계 3)으로 간다.

$$\epsilon_{rf} = \frac{\sum_{i=1}^{n_s} \delta_{pnr}(i)}{\sigma_{ps} + \sum_{i=1}^{n_s} \delta_{pr}(i)} \quad (21)$$

(6) 관리 트래픽 이용률

관리 트래픽 이용률은 전체 트래픽에 대한 SNMP 관리 트래픽의 이용률을 나타내주는 파라미터이다. 관리 트래픽의 발생은 관리 시스템에 설정된 폴링 주기와 밀접한 관계를 갖고 있다. 관리 트래픽은 사용자 트래픽의 입장에서 보면 부가적으로 발생한 트래픽이므로 이를 적정 수준으로 유지하는 것이 필요하다. 이에 대한 알고리즘은 다음과 같다.

단계 1) 초기 MIB 변수 폴링

관련 MIB 변수 집합 V(ipInReceives, ipOutRequests, ipFragOks, ipFragCreates, snmpInPkts, snmpOutPkts)를 I_{ir}를 이용하여 폴링한다.

단계 2) 비교값 설정

B_{pr} = ipInReceives, B_{por} = ipOutRequests, B_{pfo} = ipFragOks, B_{pfc} = ipFragCreates, B_{mrs} = snmpInPkts, B_{ms} = snmpOutPkts를 설정한다.

단계 3) 다음 MIB 변수 폴링

다음 폴링 변수 집합 Δ 를 이용하여 t_p 시간 간격으로 폴링한다.

단계 4) 변화량 누적

$$\begin{aligned} \delta_{pr}(i) &= ipInReceives - B_{pr} \\ \delta_{por}(i) &= ipOutRequest - B_{por} \\ \delta_{pfo}(i) &= ipFragOks - B_{pfo} \\ \delta_{pfc}(i) &= ipFragCreates - B_{pfc} \\ \delta_{mr}(i) &= snmpInPkts - B_{mr} \\ \delta_{ms}(i) &= snmpOutPkts - B_{ms} \end{aligned}$$

단계 5) 관리 트래픽 이용률 계산

중요할 경우는 식 (22)에 따라 계산을 수행하고 아니면 (단계 2)로 간다.

$$\gamma_m = \frac{\sum_{i=1}^{n_s} (\delta_{mr}(i) + \delta_{ms}(i))}{\sigma_{ps} + \sigma_{pr}} \quad (22)$$

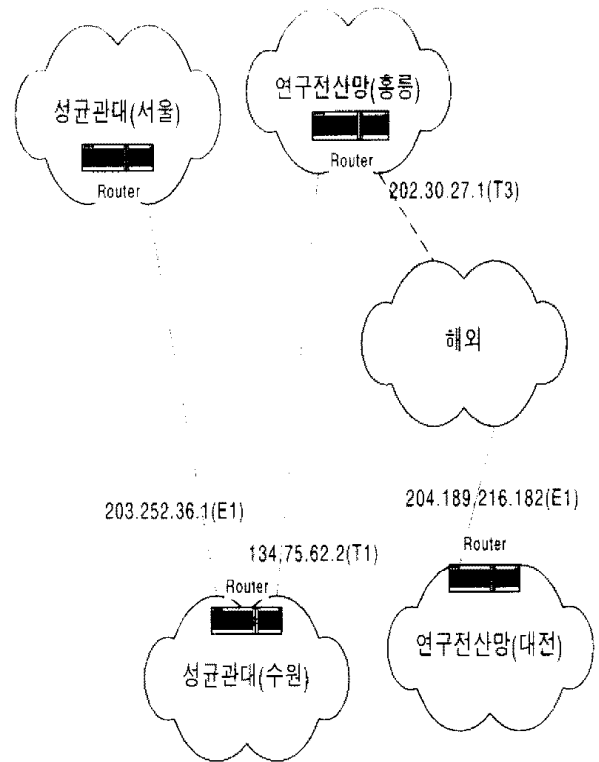
4. 실험 및 고찰

논문에서 제시한 유통 트래픽 분석 계산 알고리즘, 인터페이스 유통 패킷 분석 계산 알고리즘, 패킷 트래픽 및 관리 트래픽 분석 계산 알고리즘의 적용성을 실험하기 위하여 실제 존재하는 성균관대학교 내부 망 연결 라우터와 외부 인터넷 접속 라우터, 서울 지역 연구 전산망 해외 접속 라우터, 대전 지역 연구 전산망 해외 접속 라우터에 대해서 관리 정보를 수집하고 이에 대한 계산을 수행하였다. 제안한 알고리즘은 Case 다이어그램[2][13]에서 명시한 각 MIB 변수들 간의 관계를 규정하기 위한 목적으로 제안된 것으로 라우터와 같은 피관리 시스템에서 이를 이용하여 SNMP 에이전트가 개발되고 있다. 따라서, 실험 및 고찰에서 제시한 결과와 실측 데이터와 동일한 결과를 제시하게 된다. 모니터링과 분석에 사용된 전체 구성도는 그림 5와 같다.

그림 5에 제시된 라우터들을 주기적으로 모니터링하여 관리 정보를 수집하고 이를 제시한 알고리즘에 따라 계산을 수행하였다. 이때, 설정된 실험 환경은 다음과 같다.

[실험 환경]

- (1) 성균관대학교 내부망은 서울과 수원사이의 링크



(그림 5) 실험 대상 망 구성도
(Fig. 5) Network Environments for Experiments

이며 선로 속도는 E1 이다. 외부 인터넷 연결에 대한 속도도 T1이다. 서울 지역의 연구 전산망 가입자는 다른 기관을 통해서 해외 인터넷으로 접속한다. 이때, 연구 전산망과 다른 기관과의 선로 속도는 T3이다. 대전 지역의 연구 전산망은 E1 속도로 해외에 연결한다.

(2) 폴링 시간 간격 t_p 는 60초이다.

(3) Solaris 2.5의 crontab을 이용한 주기적인 폴링을 시행한다.

(4) 폴링 누적 데이터의 수집 기간은 98년 3월 10일 16시에서 98년 3월 16일 15시까지이다.

성균관대학교는 서울 캠퍼스와 수원 캠퍼스로 나뉘어 있으며, 이들은 E1속도의 선로로 라우터로 서로 연결되어 있다. 아울러, 인터넷 접속을 위해 연구 전산망에 가입되어 있는데 이 선로에 대한 속도는 T1이다. 이들 두 라우터에 대해서 제시한 알고리즘을 적용하면 표 1과 같은 결과를 얻을 수 있다. 서울-수원간의 선로는 이용률이 저조하여 당분간 증속의 필요성이 없음을 알 수 있다. 송신 트래픽이 수신 트래픽에 비해서 현저히 적은 것은 대부분의 트래픽이 서울에서 수원으로 향

하고 있다는 것을 보여주고 있으며, 이는 인터넷의 접속이 수원에서 이루어지므로 서울 캠퍼스의 호스트 트래픽은 수원으로 향하고 있기 때문이다. 이를 인터페이스 패킷 송수신을 파라미터에서 증명하고 있다. 그 외의 파라미터 값들은 그 값이 작은 것으로 보아 203.252.36.1 라우터는 성능과 장애 면에서 큰 문제가 없다. 인터넷 접속을 위한 선로는 이용률이 다소 높아서 추후 선로 증속이 있을 수 있음을 나타내고 있다. 134.75.62.2 라우터에서 관리 트래픽이 203.252.36.1 라우터에 비해서 많은 것은 성균관대학교의 망 관리 시스템이 이를 모니터링하고 있음을 증명한다. 여타 파라미터의 값으로 보아 장애나 성능 측면에서 상비 교체나 자원 증설 등을 수행할 필요가 없음을 알 수 있다.

연구 전산망의 서울 측 가입자는 다른 기관을 경우

<표 1> 성균관대학교 라우터의 분석 실험 결과
<Table 1> Analysis Results of Routers on SKKU

분석 파라미터	IP 주소			
	203.252.36.1		134.75.62.2	
선로 이용률	15.5424%		61.7431%	
입출력 트래픽률	송신	수신	송신	수신
	23.4907 %	76.5093 %	59.8874 %	40.1126 %
인터페이스 패킷 송수신율	송신	수신	송신	수신
	56.9868 %	43.0132 %	50.0983 %	49.9017 %
방송형 송수신 트래픽 비율	3.2422%		0.1758%	
인터페이스 패킷 송수신 손실률	0.0001%		0.1409%	
에러 수신율	0.0022%		0.0000%	
시스템 패킷 입출력률	송신	수신	송신	수신
	0.2654%	99.7346 %	1.3067%	98.6933 %
패킷 전달률	98.5418%		99.6715%	
시스템 패킷 송수신 손실률	송신	수신	송신	수신
	0.4811%	0.4818%	0.0005%	0.0019%
시스템 자원 부하율	0.0059%		0.0001%	
경로 설정 실패율	0.4882%		0.0005%	
관리 트래픽 이용률	0.0279%		0.0953%	

하이 해외 인터넷에 접속하게 된다. 따라서, 홍콩 소재 라우터(202.30.27.1)는 연구 전산망 가입자 측면에서 해외 접속 라우터 인터페이스이다. 이 라우터의 선로 이용률이 저조한 것으로 보아 선로 증속의 필요성은 없어 보인다. 그러나, IP 패킷의 손실이 많이 나타나는 것으로 보아 메모리의 확장이나 고성능 라우터의 도입이 필요한 것으로 기대된다. 입출력 트래픽 율이 심한 차이를 보이고 있지 않은 것은 타 ISP의 가입자가 연구 전산망의 가입자와 서로 통신하고 있음을 증명하고 있다. 문제가 될 만한 사항은 관리 트래픽이 상당히 많이 발생하고 있다는 것이다. 연구 전산망 관리 시스템에서 관리를 위한 폴링 주기의 개선이 필요한 것으로 기대된다.

대전 지역의 연구 전산망 가입자는 204.189.216.182

<표 2> 국외 인터넷 접속 라우터의 분석 실험 결과
<Table 2> Analysis Results of Routers for Foreign Access

분석 파라미터	IP 주소			
	202.30.27.1		204.189.216.182	
선로 이용률	31.3885%		93.2026%	
입출력 트래픽률	송신	수신	송신	수신
	44.1347 %	55.8653 %	27.2496 %	72.7504 %
인터페이스 패킷 송수신율	송신	수신	송신	수신
	51.6010 %	48.3990 %	72.5997 %	27.4003 %
방송형 송수신 트래픽 비율	0.0064%		0.0192%	
인터페이스 패킷 송수신 손실률	0.0008%		7.1922%	
에러 수신율	0.0000%		0.0001%	
시스템 패킷 입출력률	송신	수신	송신	수신
	2.7966%	97.2034 %	0.3731 %	99.6269 %
패킷 전달률	93.8700%		95.5741%	
시스템 패킷 송수신 손실률	송신	수신	송신	수신
	0.0240%	0.7197%	1.4481 %	4.2570 %
시스템 자원 부하율	0.0230%		0.0081%	
경로 설정 실패율	0.0255%		1.4910%	
관리 트래픽 이용률	1.2671%		0.0141%	

라우터를 통해서 해외 인터넷에 접속하게 된다. 따라서, 대전 지역 이남의 가입자는 이 라우터가 해외로 접속하는 인터페이스이다. 선로 이용률이 상당한 것으로 보아 증속의 필요성이 설실히 요구되고 있다. 아울러, 해외 정보에 대한 증속성이 입출력 트래픽을 파라미터에서 보여주고 있다. 높은 선로 이용률에 따라 손실되는 패킷의 수도 상당히 발생하고 있음을 시스템 패킷 송수신 손실률에서 보여주고 있다. 특히, 경로 설정 실패율이 높은 것으로 보아 이 부분에 대한 보강이 필요하다.

5. 결 론

망을 운영 보존하기 위한 관리자는 망 관리 시스템을 통하여 관리 행위를 수행한다. 그러므로, 관리 행위를 수행하기 위해서는 사용자의 QoS를 개선시키기 위한 파라미터들이 있어야 한다. 이러한 파라미터들은 관리자에게 망에 대한 실질적인 개선을 이루게 할 수 있다. 즉, 망 관리 시스템에 대한 활용 방안으로서 이들 파라미터를 이용할 수 있다. 따라서, 이 논문에서는 MIB-II의 변수를 이용하여 유동 트래픽 분석, 인터페이스 유동 패킷 분석, 패킷 트래픽 및 관리 트래픽 분석 등을 위한 파라미터를 제시하고 이를 실현하는 계산 알고리즘을 제안하였다. 알고리즘에서 제시되는 수식들은 Case 다이어그램을 기반으로 하여 제시되었다.

제시된 파라미터들은 선로 이용률, 입출력 트래픽률, 인터페이스 패킷 송수신율, 방송형 송수신 트래픽 비율, 인터페이스 패킷 송수신 손실률, 에러 수신율, 시스템 패킷 입출력률, 패킷 전달률, 시스템 패킷 송수신 손실률, 시스템 자원 부하율, 경로 설정 실패율, 관리 트래픽 이용률로 구성된다. 이들 파라미터를 계산하기 위한 알고리즘을 제시하고 실제로 운영되는 TCP/IP 통신망에 적용하여 그 효용성을 입증하였다. 실험에 적용된 통신망은 성균관대학교 내부망과 외부 인터넷 연결 라우터를 대상으로 가입자망의 특성을 살펴보았다. 해외 접속 라우터의 트래픽 특성과 관리 상황을 분석하기 위해서 연구 전산망 서울 지역 센터에서 다른 기관 접속 라우터와 연구 전산망 대전 지역 센터에서 해외로 연결되는 라우터의 인터페이스 상태를 분석하였다. 그 결과 사용자의 QoS를 향상시킬 수 있는 결과가 제시되었다. 이들 파라미터에서 제시하는 결과를 토대로 망의 재설계, 성능 개선, 선로 증속 등과 같은 관리 행위로 이어진다면, 사용자의 QoS는 상당히 개선될 것이다.

이 논문에서는, MIB-II를 기반으로 하여 인터넷에서 표준 망 관리 프로토콜 및 관리 정보로 지정된 SNMP와 MIB-II를 이용하여 사용자의 QoS를 개선시키기 위한 파라미터들을 제시하고 이를 실현하는 알고리즘을 제안하였다. 이는 인터넷의 트래픽 증가 추세로 볼 때, 통신 장비 및 선로 등과 같은 관리 대상에서 필수적으로 요구되는 망의 재설계, 장비 도입, 분석 및 진단 등을 위한 기본 자료로서 활용할 수 있다. 아울러, 트래픽 분석 시스템과 같은 분석 전문 망 관리 도구에서 반드시 필요한 알고리즘으로 사용될 것이다.

참 고 문 헌

- [1] John Blommers, Practical Planning for Network Growth, Prentice Hall, 1996.
- [2] William Stallings, SNMP, SNMPv2, and RMON : Practical Network Management, Addison-Wesley Publishing Company, 1996.
- [3] Uyless.Black, Network Management Standards, 2nd Ed. McGraw-Hill, 1995.
- [4] David Perkins, Evan McGinnis, Understanding SNMP MIBs, Prentice Hall, 1997.
- [5] K. McCloghrie, M. Rose, "Management Information Base for Network Management of TCP/IP-based Internets : MIB-II", RFC1213, March 1991.
- [6] Allan Leinwand, "Accomplishing Performance Management with SNMP", INET'93, CEA-1-CEA-5, 1993.
- [7] 산상철, 안성진, 정진욱, "Design and Implementation of SNMP-based Performance Parameter Extraction System", APNOMS, 1997.
- [8] Olga Havel, Ahmed Patel, "Design and Implementation of a Composite Performance Evaluation Model for Heterogeneous Network Management Applications", International Journal of Network Management, Vol. 5, No. 1, pp.25-46, 1995.
- [9] Jeong Soo Han, Seong Jin Ahn, Jin Wook Chung, "Web-based Performance Manager System for a Web Server", Network Operations and Management Symposium, pp.1-10.

1998

- [10] 송운집, 이길홍, 최용훈, 이준호, 장태원, 김석규, 이세용, 이상배, "SNMP를 이용한 ATM망의 VPC/VCC 관리 기법", 통신 학회 논문지 Vol. 22 No. 3, pp.580-592, 1997.
- [11] 최영수, 안성진, 정진욱, 최홍진, 변옥환, "TCP/IP 네트워크 관리에서 동시폴링과 순차폴링 트래픽의 특성 분석", 한국정보처리학회 '95년 추계 학술발표 논문집, pp.548-552, 1995.
- [12] 조강홍, 안성진, 정진욱, 박형우, "RMON MIB을 이용한 LAN 성능 및 장애 파라미터 추출에 관한 연구", 한국 정보 처리 학회 '97년 추계 학술 발표 논문집, pp.943-948, 1997.
- [13] J. Case, C. Patridge, "Case Diagrams: A First Step to Diagrammed Management Information Bases", ACM Computer Communication Review, March 1989.



안 성 진

1988년 성균관대학교 정보공학과 (공학사)
 1990년 성균관대학교대학원 정보공학과 (공학석사)
 1998년 8월 성균관대학교대학원 정보공학과 (공학박사)

1990년~1995년 시스템 공학 연구소 연구원
 1996년 8월 정보 통신 기술사 자격 취득
 관심 분야 : 망 관리, 분산 시스템, 고속 통신



정 진 욱

1974년 성균관대학교 전자공학과 (공학사)
 1979년 성균관대학교 전자공학과 (공학석사)
 1991년 서울대학교 계산통계학과 (공학박사)

1982년~1985년 한국과학기술연구소 실장
 1981년~1982년 Racal Milgo Co. 객원연구원
 1985년~현재 성균관대학교 정보공학과 교수
 관심분야 : 네트워크 관리, 네트워크 보안, 고속 및 무선 통신 프로토콜