

웹 응용 서비스 관리를 위한 성능 관리자 시스템의 설계 및 구현

한 정 수[†] · 안 성 진^{††} · 정 진 옥^{†††} · 박 형 우^{††††}

요 약

본 논문은 웹 서버의 트래픽 관리를 위하여 성능 파라미터를 정의하고 이를 분석하기 위한 관리자 시스템을 웹 기술과 JAVA 기술을 이용하여 구현함으로써 웹 서비스를 모니터링하고 웹 서버의 성능을 관리하고자 하였다. 현재 웹 상에서의 트래픽 관리가 웹 성장에 의한 웹 트래픽의 증대로 그 필요성이 대두됨에 따라, 웹 서비스 트래픽 관리와 웹 서버 성능에 대한 효율적인 관리가 필요하게 되었다. 이를 위해 네트워크 상의 임의의 호스트에 성능 관리자 시스템을 위한 클라이언트를 구현하여 사용자로 하여금 수집 요구와 분석 요구, 실시간 모니터링, 그리고 비교 분석 요구 등을 할 수 있는 인터페이스를 설계하고, 이러한 요구를 분석하는 서버 시스템을 구현하였다. 또한 기존의 MIB을 사용하여 각 분석 항목에 맞게 HTTP 메시지량과 바이트량 분석, 연결 수 분석, 입출력 트래픽 비교 분석 등과 같은 새로운 성능 파라미터를 정의하였다. 아울러, 웹 클라이언트-서버 시스템 간의 통신을 위해 새로운 메시지 형식을 정의하고 구현하였다. 따라서 웹 서버 시스템의 운영과 관리 시 성능에 따른 문제점을 발견하고 해결을 지원하는데 기여할 수 있다.

Design and Implementation of Performance Manager System for Web Application Service Management

Jeong Soo Han[†] · Seong Jin Ahn^{††} · Jin Wook Chung^{†††} · Hyoung Woo Park^{††††}

ABSTRACT

In this paper, we shows the implementation of Web based performance manager which analyze the traffic of a Web server to support the diagnostics of it. The manager monitors the HTTP traffic by polling and measures and presents is performance on demand. To enhance the adaptability of management interface Web based interfaces with JAVA is used. Recently, the need of traffic management on s Web has grown, because of increasing Web traffic. Therefore, the traffic management of Web service and the effective management of a Web server's performance are needed. We have designed interfaces with which is comprised of Collection-Request, Analysis-Request, Realtime-Monitoring, Comparison-Analysis on a client with Web Browser on a network, and implemented the server system that can analyze these requests. Also we have introduced some performance indicator by referring a Web related MIB. Also, we have designed and developed a message format for communication between the Web client and the server system.

† 준 회 원:성균관대학교 전기전자 및 컴퓨터공학부
 †† 경 회 원:성균관대학교 전기전자 및 컴퓨터공학부
 ††† 중신회원:성균관대학교 전기전자 및 컴퓨터공학부
 †††† 경 회 원:시스템공학연구소 고성능 전산망 개발실
 논문접수:1997년 6월 5일, 심사완료:1997년 11월 3일

1. 서 론

정보화 사회로 접어들면서 컴퓨터 통신 기술이 급속히 발전되었고 이에 발맞추어 컴퓨터 네트워크를 이용하는 사용자들의 요구 사항이 점점 복잡, 다양화되어 가는 추세에 있다. 또한 초고속 네트워크 환경에서 제공되는 대부분의 응용 서비스들은 대용량의 데이터와 실시간 처리를 요구하는 멀티미디어 서비스로 변해가고 있다. 이러한 초고속 정보 통신망 사용자의 다양한 서비스 욕구를 충족시키기 위해서는 통신 속도의 고속화 및 안정적인 하부 계층의 프로토콜 개발도 중요하지만 전송이나 응용 등의 상부 계층에서 응용 서비스를 대상으로 하는 새로운 관리 방식이 필요하게 된다.[1][2] 이는 단지 하부 네트워크의 고속화와 안정만으로는 사용자가 요구하는 모든 서비스를 원활하게 관리하고 지원할 수 없기 때문이다. 따라서 기존의 자원 관리 방식에서 벗어나 응용 서비스를 대상으로 하는 새로운 관리 기술이 요구되어진다. 이러한 응용 서비스 트래픽에 대한 관리는 서버와의 트래픽과 사용자간의 트래픽을 모니터링하여 서버와 사용자 소프트웨어의 품질 관리를 제공하기 위한 제반 활동을 의미한다[3]. 이를 위하여 각 응용 서비스별로 관리 항목을 설정하고 이들에 대한 정보를 수집하며 이를 토대로 하며 원활한 트래픽 흐름이 존재하도록 관련 파라미터를 조정할 수 있는 응용 서비스 트래픽 관리 플랫폼이 필요하다. 응용 서비스 관리 플랫폼을 각 사용자나 서버의 관리자가 트래픽의 흐름, 응답 시간등을 확인할 수 있는 설비를 제공하여야 한다. 아울러 그래픽 관리자 인터페이스를 제공하여 이를 사용하는 자가 운영, 관리를 원활히 하도록 지원되어야 한다. 이러한 응용 서비스 트래픽 관리의 필요성과 함께 현재 사용자 요구 사항을 전세계 네트워크를 통해 텍스트, 그림, 그래픽, 사운드, 영화와 같은 모든 종류의 멀티미디어 형식이 포함되어 있는 정보를 배포할 수 있는 시스템인 WWW(World Wide Web)와 이러한 기능을 담당하는 프로토콜이 하이퍼텍스트 전송 프로토콜(HyperText Transfer Protocol)의 고속의 성장으로 인해 발생하는 트래픽 양이 증가하게 되고, 그에 따라 웹 서버의 응용 서비스가 발생하는 트래픽 양 또한 놀라울 정도로 증가하게 되어 그 관리가 시급한 문제가 되고 있다.[4]

현재 연구 동향을 살펴보면 세계에 가장 널리 분포되어 있는 인터넷상의 Domain Name Server(DNS)의 트래픽 분석을 통한 DNS의 성능을 관리하고[5] Internet상에서 간단한 파일 전송 프로토콜인 FTP의 네트워크 트래픽 분석을 통해 FTP 효율을 관리하는 등의 연구가 진행되었다[6]. 이러한 연구 동향에 발맞추어 본 논문은 실제 데이터에 입각하여 웹 서버의 성능을 관리할 수 있는 체계적인 시스템을 구현하게 되었다. 또한 이러한 웹 서버의 성능 관리를 위한 시스템의 설계를 위해 다음과 같은 MIB를 정의하여야 하고 그 속에서 필요한 MIB 변수 값을 추출하여 웹 서버의 응용 서비스 트래픽 분석을 위한 분석 항목을 정의하였다. 첫째, 시스템 응용 MIB은 WWW 서버가 실행되었을 때의 시간이나 프로세스 이름, 지속 시간을 정의할 수 있고, 두번째 네트워크 서비스 모니터링 MIB은 WWW 서버의 모니터링을 위한 IP 주소나 입/출력 패킷 수를 정의할 수 있으며, 마지막으로 WWW 서버 MIB은 WWW 서버의 입/출력 바이트나 서버가 받은 요구 수, 보낸 응답 수를 정의할 수 있다.

구현 환경을 살펴보면, 크게 대상 머신, 운영체제, 사용 언어, 사용 프로토콜, 인터페이스로 나눌 수 있다. 먼저, 대상 머신은 Sun UltraSPARC 및 호환 기종에서 동작하도록 설계하였으며, 자바 애플릿의 실행은 시스템에 무관하게 실행되므로 자바 애플릿이 실행 가능한 웹 브라우저만 있으면 관리 정보 설정과 결과 뷰를 볼 수 있도록 구현한다. 다음은 운영체제로서 각각의 시스템은 Solaris 2.5의 라이브러리와 시스템 함수를 사용하며, 자바 API(Application Programming Interface)를 지원할 수 있는 플랫폼 상에서 구현하였다. 사용 언어로서는 통합 관리자 시스템은 자바 언어 JDK 1.0.2와 HTML 3.0을 사용하며, 자바 언어는 관리자 시스템을 위한 클라이언트와 관리자 시스템을 위한 서버인 화면 표시 제어기(Visualization Controller)의 구현에 사용되며, 초기 웹 서버 연결에는 HTML을 사용하도록 하였다. 다음 사용 프로토콜로는 웹 문서 전송 프로토콜로써 HTTP(HyperText Transfer Protocol)을 사용하며[7], 요구 전송 프로토콜과 뷰정보 전송 프로토콜로써 자바 소켓 응용 프로그래밍 인터페이스를 사용하며, 이를 기반으로 관리 응용 전송 프로토콜(MATP: Management Application Transfer Protocol)을 설계하여 구현하였다[8]. 마지막

으로 인터페이스 부분을 살펴보면 통합 관리자 시스템의 인터페이스는 웹 클라이언트와 인터페이스와 웹 응용 세스 트래픽 분석 시스템과의 인터페이스를 갖는다.

본 논문에서는 Web 관리를 위해 JAVA 기술을 사용하여 Web 서버를 Web으로 관리하였고, 기존의 MIB을 사용하여 새로운 성능 파라미터를 도출함으

로써 Web 서버 트래픽 관리 관리자 시스템을 구축하였다.

2. 웹 서버의 분석 항목

웹 응용 서비스 트래픽을 분석하기 위해 다음과 같이 3가지 형태의 MIB 변수 그룹을 사용할 수 있다.

〈표 2.1〉 웹 응용 서비스 트래픽 분석 시스템에서 사용되는 MIB 변수 설명
(Table 2.1) Performance Indicators of Web Sewer

분석 항목명	MIB 변수명	의미
HttpConnection	applAccumulatedInboundAssociatons	응용 초기화로 인한 응용 엔티티에 연관된 전체수(응답자)
HttpInPacket	applAccumulatedInboundPacket	서버의 입력 패킷 수
HttpOutPacket	applAccumulatedOutboundPacket	서버의 출력 패킷 수
HttpRequest	WWWSummaryInRequests	엔티티에서 수신된 요청의 전체 갯수
HttpResponse	WWWSummaryOutResponse	엔티티에서 생성된 응답의 전체 갯수
HttpInOctet	WWWSummaryInContentBytes	엔티티에서 수신된 데이터 바이트의 수
HttpOutOctet	WWWSummaryOutContentBytes	엔티티에서 생성된 데이터 바이트의 수
Sbyte	HttpInOctet, HttpOutOctet	엔티티에서 수신한 입력 바이트에 대한 출력 바이트의 수
SPkt	HttpInPacket, HttpOutPacket	엔티티에서 수신한 입력 패킷에 대한 출력 패킷의 수
SRR	HttpRequest, HttpResponse	엔티티에서 수신한 요청에 대한 응답의 수
SBP	HttpInOctet, HttpOutOctet, HttpInPacket, HttpOutPacket	엔티티에서 수신, 송신한 패킷에 대한 입/출력, 전체 바이트의 수
SBRR	HttpInOctet, HttpOutOctet, HttpRequest, HttpResponse	엔티티에서 수신한 요청에 대한 응답의 입/출력, 전체 바이트의 수
SPRR	HttpInPacket, HttpOacket, HttpRequest, HttpResponse	엔티티에서 수신한 요청에 대한 응답의 입/출력, 전체 패킷의 수
SC	HttpConnection, HttpRequest, HttpResponse, HttpInOctet, HttpOutOctet, HttpInPacket, HttpOutPacket	엔티티에 연결된 수에 대한 패킷, 바이트, 요청에 대한 응답의 수
TByte	HttpInoctet, HttpOutOctet	엔티티에서 수신한 입력 바이트에 대한 출력 바이트의 수
TPkt	HttpInPacket, HttpOutPacket	엔티티에서 수신한 입력 패킷에 대한 출력 패킷의 수
TRP	HttpRequest, HttpResponse	엔티티에서 수신한 요청에 대한 응답의 수
RByte	HttpConnection, HttpInOctet, HttpOutOctet	엔티티에서 연결된 수에 대한 입/출력, 전체 바이트의 수
RPkt	HttpConnection, HttpInPacket, HttpOutPacket	엔티티에 연결된 수에 대한 입/출력 패킷의 수
RRR	HttpConnection, HttpRequest, HttpResponse	엔티티에 연결된 수에 대한 요청당 응답의 수

1) 시스템 응용 MIB

WWW 서버가 실행되었을 때의 시간이나 프로세스 이름, 지속 시간 등을 정의할 수 있다.

(systApplElmtRunTimeStarted, sysAppleElmtRunName등)

2) 네트워크 서비스 모니터링(NSM) MIB

WWW 서버의 모니터링을 위한 IP 주소나 입/출력 패킷 수 등을 정의할 수 있다.

(aaplIP, applSysOS, applUptime, applOperStatus, applAvgInPacket, applAvgOutpacket등)

3) WWW 서버 MIB

WWW 서버의 입/출력 바이트나 서버가 받은 요구 수, 응답 수 등을 정의할 수 있다.

(WWWEntityName, WWWEntityContact, WWWSummaryInRequests, WWWSummaryOutResponse 등)

웹 응용 서비스 트래픽 분석에 사용되는 MIB 변수 및 설명은 위 <표 2.1>과 같다.[9]

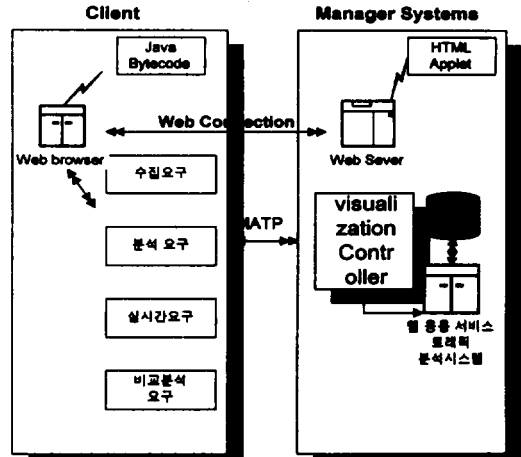
3. 시스템 구현

3.1 통합 성능 관리자 시스템 구현

3.1.1 기능

웹 서버 응용 서비스 트래픽 분석을 위한 통합 성능 관리자 시스템을 살펴보면 (그림 3.1)과 같이 나타낼 수 있다. 전체 구조도를 보면, 웹 서버 시스템과 클라이언트 부분으로 구성되는데, 초기에 웹 클라이언트에서 웹 서버에게 서비스를 요구하면, 웹 서버는 해당 HTML 문서를 제공하면서 초기 연결을 맺게 된다[7]. (그림 3.1)에서 보여지는 것처럼 서비스의 종류에는 수집 요구, 분석 요구, 실시간 분석, 비교 분석이 제공된다. 그러나 사용자에게 모든 서비스를 제공하지는 않고 권한이 인정된 사용자만이 사용이 가능하도록 수집 요구 시 사용자 인증에 필요한 사용자 이름과 패스워드를 입력하게 하였다. 이러한 수집 요구에 의해서만 수집된 데이터를 가지고 분석 요구를 행할 수 있기 때문이다. 그러나 실시간 분석 요구는 모

든 사용자에게 그 권한을 인정하여 자신이 풀링하고자 하는 IP 주소를 가지고 항상 실시간 적으로 데이터를 분석할 수 있도록 하였다. 이러한 모든 요구는 웹 서버 시스템에서 포함되어 있는 화면 표시 제어기와 연결되어 작동하는데 이러한 것은 웹 클라이언트 내의 구성 요소들과 새로이 정의된 메시지 형식으로 통신하여 웹 응용 서비스 트래픽 분석 시스템에서 수집한 데이터를 데이터베이스에서 읽어 들여 웹 클라이언트의 각 구성요소에게 전달하게 된다. 여기서 사용되어지는 메시지 형식(MATP:Management Application Transfer Protocol)은 새로이 정의되어 사용되어진다.

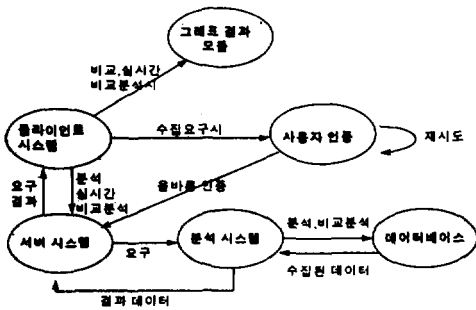


(그림 3.1) 통합 성능 관리자 시스템의 전체 구조도
(Fig. 3.1) Integrated performance manager system architecture

3.1.2 상태 천이도

본 논문에서 구현된 웹 응용 서비스 트래픽 분석을 위한 통합 성능 관리자 시스템의 내부 동작에 대한 상태 천이도를 살펴보면 (그림 3.2)와 같다. 초기의 클라이언트 시스템에서의 요구 사항은 서버 시스템으로 전달이 되어지고, 그 해당 요구에 맞게 트래픽 분석 시스템은 데이터베이스의 연관 동작에서 분석된 데이터를 클라이언트 시스템으로 전달이 되어진다. 이렇게 전달된 데이터들은 화면 출력을 위한 모듈로 전달되어 결과를 출력하게 된다. 특히 본 논문에서 구현된 시스템의 사용을 위한 사용자 인증은 사용자

가 수집 요구를 시도할 때 요구 되어진다. 물론 수집 요구 없이 분석 요구나 비교 분석은 할 수 없다. 다만 어느 사용자나 실시간 요구 분석을 하여 그때 그때의 값을 출력할 수 있다.



(그림 32) 통합 성능 관리자 시스템의 내부 동작 상태 천이도
(Fig. 32) State transition of integrated performance manager system

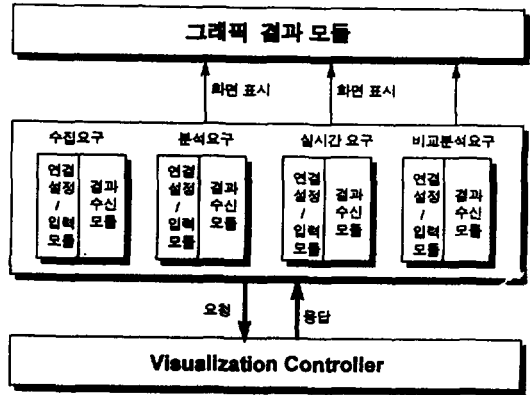
3. 성능 클라이언트 시스템

3.2.1 구조

성능 관리자 시스템을 위한 클라이언트의 구조는 (그림 3.3)과 같다.

먼저, 클라이언트상의 각 구성요소에서 이루어진 요구는 화면 표시 제어기를 위한 서버에게 정의된 메시지 형식으로 전달되어진다. 이렇게 전달된 요구 사항에 맞게 서버에서는 웹 응용 서비스 트래픽 분석 시스템과 새로운 메시지 형식으로 전달되어 관련 요구 사항에 맞는 데이터를 전달받게 된다. 이렇게 전달받은 데이터는 클라이언트에게 전달되어 그래픽을 위한 모듈로 그 결과가 전달되어 사용자에게 보여지게 된다. (그림 3.3)에서 보여주는 모듈을 살펴보면, 연결 설정/입력 모듈과 결과 수신 모듈로 나눌 수 있는데 먼저, 연결 설정/입력 모듈은 사용자가 수집 요구를 선택하였을 시 서버 시스템과의 연결을 설정하게 되고 그에 따라 필요한 파라미터를 위한 입력을 대기하게 되는 모듈이다. 결과 수신 모듈에서는 화면 표시 제어기에서 수신한 결과 데이터 및 요구 결과를 수신하는 모듈로서 이렇게 수신된 데이터를 그래픽 결과 모듈로 전달하게 된다.

그럼, 클라이언트 상의 각 구성요소들에 대해 살펴보면 다음과 같다.



(그림 3.3) 성능 관리자 시스템을 위한 클라이언트 시스템 전체 구조도
(Fig. 3.3) Architecture of the client system for performance manager system

1) 수집 요구

① 의미: 일정 기간 동안 <표 2.1>에서 정의한 성능 파라미터에 대한 값을 폴링하고자 할 때 사용된다. 폴링한 데이터는 클라이언트 시스템이 서버 시스템에게 요청할때까지 관리자 시스템 내의 데이터베이스에 저장하게 된다.

② 기능: 사용되는 파라미터로는 폴링하고자 하는 호스트 이름이나 IP 주소와 폴링하고자 하는 기간 등을 입력으로 한다. 또한 수집 요구 사용자는 그 권한을 검사하기 위한 사용자 인증 과정을 거쳐야 한다.

2) 분석 요구

① 의미: 수집 요구된 데이터에 대해 데이터의 분석 결과가 그래픽 형식으로 출력 되어진다.

② 기능: 사용되는 파라미터로는 수집 요구 시 폴링한 IP 주소와 분석을 요하는 기간을 입력한다.

3) 실시간 요구 분석

① 의미: 그때 그때 변하는 값을 실시간으로 결과를 분석한다.

② 기능: 폴링하고자 하는 IP 주소와 일정 기간 폴링 횟수를 입력한다.

4) 비교 분석 요구

① 의미: 누적된 데이터를 연관성 있는 분석 항목들

과의 비교를 위해 사용한다.

② 기능: 사용되는 파라미터로는 해당 IP 주소와 폴링 시간을 입력한다. 이렇게 입력된 파라미터를 사용하여 요약별과 시간대별로 분석할 수 있고, 해당 IP 주소와 일정한 폴링 횟수의 입력으로 실시간 분석을 할 수 있다.

각 분석 항목에 대해 살펴보면

가. 요약별: Sbyte(입력 바이트에 대한 출력 바이트), SPkt(입력 패킷에 대한 출력 패킷), SRR(요청에 대한 응답), SBP(패킷 당 바이트 수), SBRR(요청에 대한 응답 당 바이트 수), SPRR(요청에 대한 응답 당 패킷 수), SC(연결 당 바이트, 패킷 수)

나. 시간대별: Tbyte(입력 바이트에 대한 출력 바이트), TPkt(입력 패킷에 대한 출력 패킷 수), TRR(요청에 대한 응답 수)

다. 실시간별: Rbyte(연결 당 입/출력, 전체 바이트 수), RPkt(연결 당 입/출력, 전체 패킷 수), RRR(연결 당 요청에 대한 응답 수)

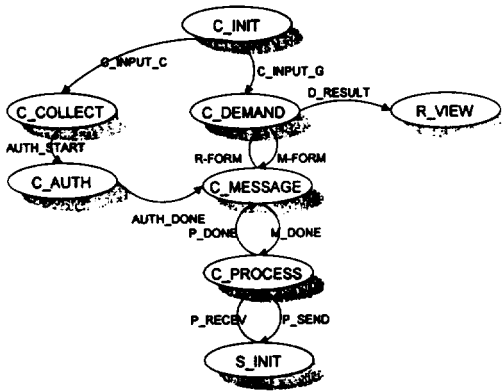
3.2.2 상태 흐름 및 동작

성능 관리자 시스템을 위한 클라이언트 시스템은

C_INIT, R_VIEW, C_COLLECT, C_DEMAND, C_AUTH, C_MESSAGE, C_PROCESS, S_INIT 등의 상태를 가진다. C_INIT 상태에서는 시스템의 초기 상태를 설정하고 입력 발생시 각 요구 사항에 맞는 상태로 천이하게 된다. 다시 말해서 C_COLLECT, C_DEMAND 상태로 천이하게 된다. 단, 여기서 C_DEMAND는 수집 요구를 제외한 분석 요구, 실시간 요구, 비교 분석 요구에 해당하는 상태를 지칭한다. 이렇게 천이된 상태에서 새로이 정의된 메시지를 형성하게 위해 C_MESSAGE 상태로 천이하게 된다. 단, C_COLLECT 상태에서는 사용자 인증을 하기 위해 C_AUTH 상태로 천이한 후, 인증이 올바르게 C_MESSAGE 상태로 천이하게 된다. 이렇게 형성된 메시지는 관리자 시스템을 위한 서버 시스템으로 전송하기 위한 C_PROCESS 상태로 천이를 한 후, 서버 시스템으로 전송하게 된다. 반대로 서버 시스템에서 전송된 결과 데이터는 반대의 순서로 처리되며 각 요구에 해당하는 결과를 보여주는 R_VIEW 상태로 천이하게 된다. 상태 천이에 따른 자세한 동작을 (그림 3.4)과 <표 3.1>와 같이 요약하였다.

<표 3.1> 성능 관리자 시스템을 위한 클라이언트 시스템의 상태 천이 테이블
 <Table 3.1> State transition table of client system for performance manager system

Current State	Event	Action	Next State
C_INIT	C_INPUT_C	시스템 초기상태에서 수집 요구를 위한 입력 시 수집 요구 화면 표시	C_COLLECT
	C_INPUT_G	시스템 초기상태에서 분석 요구, 실시간 요구, 비교 분석 요구를 위한 입력 시 각 요구 화면 표시	C_DEMAND
C_COLLECT	ACTH_START	수집 요구 시에 사용자 인증을 위한 작업	C_AUTH
C_AUTH	AUTH_DONE	올바른 사용자 인증 후 메시지 포맷 형성	C_MESSAGE
C_DEMAND	M_FORM	각 요구 사항 후 메시지 포맷 형성	C_MESSAGE
	D_RESULT	결과로 얻은 데이터를 그래프로 출력	R_VIEW
C_MESSAGE	M_DONE	메시지 포맷 형성 후 전송을 위해 대기	C_PROCESS
	R_FORM	결과로 받은 메시지에서 필요한 데이터 부분을 얻어냄	C_DEMAND
C_PROCESS	P_DONE	수신한 패킷에서 필요한 데이터 부분을 추출하기 위한 작업으로 전달	C_MESSAGE
	P_SEND	송신을 위해 형성된 메시지를 server에게 전달	S-INIT
R_VIEW		수신된 결과를 그래프로 형태로 출력	
S_INIT	P_RECEV	서버에서 클라이언트로 패킷을 전송	C_PROCESS

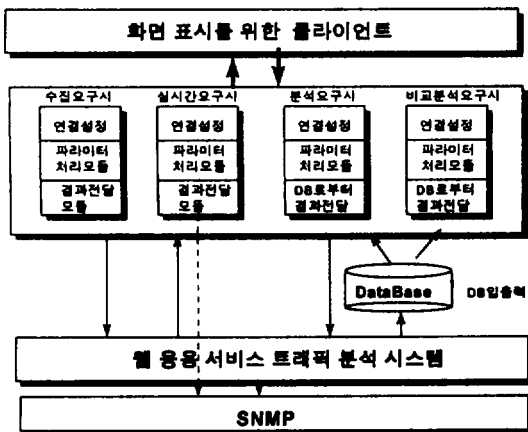


(그림 3.4) 성능 관리자 시스템을 위한 클라이언트 시스템의 상태 천이도
 (Fig. 3.4) State transition diagram of client system for Performance manager system

3.3 성능 서버 시스템

3.3.1 구조

웹 성능 관리자 시스템을 위한 서버 시스템의 전체 구조도는 (그림 3.5)와 같다. 클라이언트에서 받은 패킷을 분석하여 각 요구에 해당하는 모듈로 필요한 정보가 전달되어진다. 또한 각 모듈의 해당 요구에 맞게 웹 응용 서비스 트래픽 분석 시스템과 이에 정보를 저장하는 데이터베이스와 함께 결과를 생성하여 클라이언트에게로 전달하게 된다. 분석 요구 시나 비

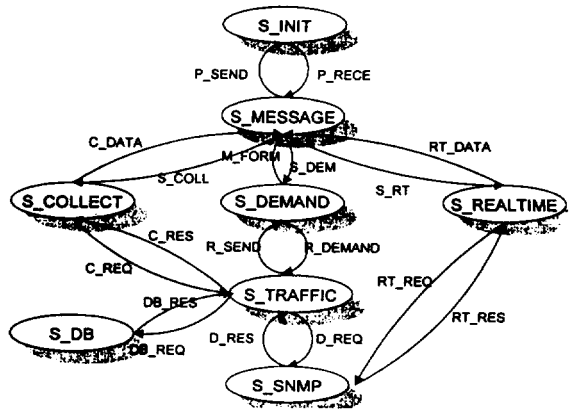


(그림 3.5) 성능 관리자 시스템을 위한 서버 시스템 전체 구조도
 (Fig. 3.5) Architecture of server system for performance manager system

교 분석 요구시에는 데이터베이스에서 결과를 전달 받아 처리를 하게 되지만 실시간 요구 시는 서버시스템에서 직접 SNMP를 호출하여 값을 전달 받게 된다.

3.3.2 상태 흐름 및 동작

성능 관리자 시스템을 위한 서버 시스템은 S_INIT, S_MESSAGE, S_DEMAND, S_COLLECT, S_REALTIME, S_TRAFFIC, S_DB, S_SNMP 등의 상태를 가진다. S_INIT 상태에서는 시스템의 초기상태를 설정하고 클라이언트에서 전달받은 패킷의 처리를 위해 다음 상태인 S_MESSAGE로 상태 천이를 하게 된다. S_MESSAGE 상태에서는 전달 받은 패킷을 분석하게 되고, 분석된 결과에 의해 각 요구에 맞는 상태로 천이하게 된다. 즉 S_COLLECT, S_DEMAND, S_REALTIME 등의 상태로 천이를 하게 된다. 먼저 S_COLLECT에서는 클라이언트에 의한 수집 요구에 대한 처리를 하며 트래픽 분석 시스템에게 해당 MIB 변수에 대한 값을 수집하도록 S_TRAFFIC 상태로 천이하게 된다. 다음은 S_DEMAND로서 클라이언트의 분석 요구와, 비교 분석 요구에 대한 처리가 행해지며 트래픽 분석 시스템에게 해당 MIB 변수에 대한 결과를 요청하게 된다. 마지막으로 S_REALTIME에서는 클라이언트에서 요청한 실시간 요구에 대한 처리가 행해지며 직접 SNMP를 호출하여 해당 MIB 변



(그림 3.6) 성능 관리자 시스템을 위한 서버 시스템의 상태 천이도
 (Fig. 3.6) State transition of server system for performance manager system

〈표 3.2〉 성능 관리자 시스템을 위한 서버 시스템의 상태 천이 테이블
 (Table 3.2) State transition table server system for performance manager system

Current State	Event	Acton	Next State
S_INIT	P_RECE	시스템 초기 설정 후, 클라이언트로 부터 수신한 패킷을 분석한다.	S_MESSAGE
S_MESSAGE	S_COLL	분석된 패킷에 의해 수집 요구를 위해 필요한 정보를 분석한다.	S_COLLECT
	S_DEM	분석된 패킷에 의해 분석, 비교분석 요구를 위해 필요한 정보를 분석한다.	S_DEMAND
	P_SEND	결과 정보를 클라이언트에게 전달하기 위한 작업을 한다.	S_INIT
	S_RT	분석된 패킷에 의해 실시간 요구를 위해 필요한 정보를 분석한다.	S_REALTIME
S_COLLECT	C_DATA	수집 요구의 결과에 대해 패킷을 형성한다.	S_MESSAGE
	C_REQ	수집 요구를 위해 트래픽 분석 시스템에게 요청한다.	S_TRAFFIC
S_DEMAND	M_FORM	분석, 비교분석, 요구의 결과에 대해 패킷을 형성한다.	S_MESSAGE
	R_DEMAND	분석, 비교분석 요구를 위해 트래픽 분석 시스템에게 요청한다.	S_TRAFFIC
S_REALTIME	RT_DATA	실시간 요구에 대한 결과를 패킷을 형성한다.	S_MESSAGE
	RT_REQ	실시간 분석을 위해 직접 데이터를 폴링한다.	S_SNMP
S_TRAFFIC	R_SEND	분석된 결과를 해당 요청에게 전달한다.	S_DEMAND
	D_REQ	분석을 위해 SNMP를 폴링한다.	S_SNMP
	DB_REQ	데이터베이스에게 분석한 정보를 저장하도록 한다.	S_SNMP
	C_RES	수집 요구시에 결과를 전달한다.	S_SNMP
S_DB	DB_RES	분석요구시 결과를 트래픽 분석 시스템에게 전달한다.	S_TRAFFIC
S_SNMP	D_RES	폴링한 데이터를 트래픽 분석 시스템에게 전달한다.	S_TRAFFIC

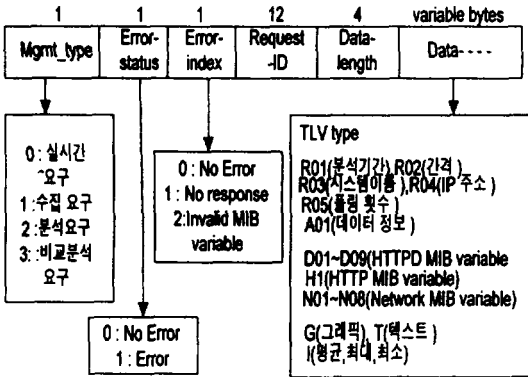
수 값을 요청하게 된다. S_TRAFFIC 상태에서는 해당 요구에 대해 SNMP를 호출하기 위한 S_SNMP 상태로 천이하게 되고, 또한 이렇게 받아들인 결과를 데이터베이스에 저장하는 S_DB 상태로 천이하게 된다. 다음은 S_SNMP 상태로서 트래픽 분석 시스템에서 해당 MIB에 대한 값을 폴링 하기 위한 상태이다. 마지막으로 S_DB는 폴링한 MIB 변수 값을 저장하기 위한 상태이다. 상태 천이에 따른 자세한 동작을 (그림 3.6)과 〈표 3.2〉와 같이 요약하였다.

3.4 클라이언트-서버간 통신

앞에서 설명하였듯이 웹 클라이언트와 웹 서버간의 연결 관계는 초기 웹 문서의 전송을 위한 웹 연결과 관리 정보를 전송하는 관리 응용 전송 프로토콜

(MATP)이 담당하게 된다.[10] 이러한 프로토콜에 대해 살펴보면 (그림 3.7)과 같다. 이러한 메시지 형식을 사용하여 웹 응용 서비스 트래픽 분석 시스템과 성능 관리자 시스템간의 인터페이스를 위한 메시지 형식을 정의할 수 있다. 따라서 성능 관리자 시스템의 클라이언트는 성능 관리자 시스템의 서버에게 트래픽 분석을 위한 요구를 할 수 있으며, 성능관리자 시스템의 서버 또한 같은 데이터 형식을 사용하여 성능 관리자 시스템의 클라이언트에게 응답을 하게 된다. (그림 3.7)에서 볼 수 있듯이, 데이터 형식은 TLV (Type, Length, Value) 형식을 따르게 된다. 각 인터페이스 요구 시 Mgmt_type 필드 부분에 그 형식에 맞는 값(실시간 요구:0, 수집 요구:1, 분석 요구:2, 비교 분석 요구:3)이 지정되고, Error-status와 Error-index

필드 부분에는 0으로 초기화하여 패킷을 형성하여 웹 서버에게 데이터를 전송하게 된다. 성능 관리자 시스템의 서버에서는 해당 인터페이스 요구에 맞는 값을 지정하고 Error-status와 Error-index 부분에는 값을 지정하게 된다.



(그림 3.7) 성능 관리자 시스템과 트래픽 분석기 사이의 메시지 형식 정의
(Fig. 3.7) Message format between performance manager system and traffic-analysis-system

웹 서버에서는 또한, Data 필드 부분에 해당 MIB에 대한 ID와 출력으로 사용될 방식(그래프와 텍스트)과 분석 요구와 비교 분석 요구에서 사용하게 될 최대, 최소, 평균에 대한 값을 지정하여 전송하게 된다. 또한 출력을 위해 필요한 파라미터를 위해 분석 기간, 간격, 시스템 이름, IP 주소, 플링 기간에 대한 형태가 사용되어진다. 데이터 정보를 위해 사용되는 형태 또한 볼 수 있다. 또한 전체 데이터 부분의 길이를 나타내는 Data-Length 필드 부분과 여러 웹 서버의 플링을 위해 사용될 Request-ID 필드 부분이 존재한다[10]

4. 사용자 인터페이스와 결과 화면

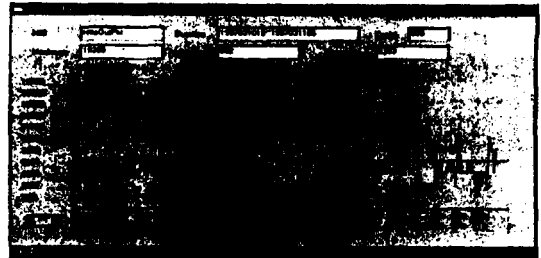
성능 관리자 시스템의 사용자 인터페이스와 결과 화면은 (그림 4.1), (그림 4.2), (그림 4.3)과 같다. 먼저, (그림 4.1)은 사용자의 수집 요구 시 사용되어질 사용자 인터페이스 부분으로서 사용자에게 의해 플링하고자 하는 IP 주소와 수집하고자 하는 기간을 입력하고

나서 Send 버튼을 입력하면 사용자 인증에 관한 화면이 출력되며 사용자의 올바른 인증 확인 후 서버 시스템에게 수집 요구를 요구할 수 있다.



(그림 4.1) 클라이언트 시스템의 수집 요구 사용자 인터페이스
(Fig. 4.1) User interface of collection-request

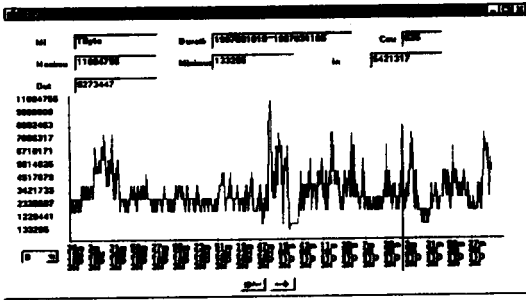
(그림 4.2)는 이런 수집 요구에 의해 수집된 분석 항목에 대해 분석 결과를 보여주는 결과 화면이다. 이러한 결과 화면은 사용자에게 의해 분석 요구 시 사용되어진다. 결과로 보여주는 항목으로는 분석 항목, 분석 기간, 전체 분석 갯수, 최대값, 최소값, 평균값을 볼 수 있다. 또한 x축은 분석 기간을 나타내는 월, 일, 시, 분을 나타내며 x축은 분석 항목의 값들을 나타낸다.



(그림 4.2) 클라이언트 시스템상에서 분석 요구 시 결과 화면
(Fig. 4.2) Result view of analysis-request

또한, (그림 4.3)은 사용자에게 의해 비교 분석 요구가 선택되어질 때 볼 수 있는 결과 화면이다. 특히 이러한 결과는 사용자가 시간대별로 보기를 원할 때 볼 수 있는 결과 화면이며, 결과 항목으로는 분석 항목명, 분석 기간, 전체 분석 갯수, 최대값, 최소값, 그리

고 각 항목에 대한 결과값을 볼 수 있다. 여기서는 입력 바이트에 대한 출력 바이트에 대한 결과를 볼 수 있다.



(그림 4.3) 클라이언트 시스템에서 비교 분석 요구 시 결과 화면(시간대별)
 (Fig. 4.3) Result view of comparison-analysis(summary-analysis)

5. 결 론

본 논문은 웹 응용 서비스를 모니터링하고 웹 서버의 성능을 관리하기 위해 HTTP 메시지량과 바이트량 분석, 연결 수 분석, 입출력 트래픽 비교 분석 등의 분석 항목을 도출하고 이를 위한 관리자 시스템을 구현하였다. 이를 위해 네트워크 상의 웹 서버에 성능 관리를 위해 서버와 클라이언트를 설계, 구현하였으며 사용자로 하여금 수집 요구와 분석 요구, 실시간 모니터링, 그리고 비교 분석 요구 등을 할 수 있는 인터페이스를 구현하였다. 수집 요구 시는 사용자에 의해 관리하고자 하는 분석 항목에 대해 일정한 분석 기간 동안 데이터를 수집하고자 할 때 사용되어지며, 분석 요구는 이렇게 수집된 데이터를 사용하여 분석 항목에 대해 분석을 요구하는 항목이며, 실시간 모니터링은 이러한 분석 항목을 실시간적으로 그때 그때 출력하고자 할 때 사용한다. 마지막으로 비교 분석 항목은 수집된 항목에 대해 서로 연관성 있는 항목을 비교 분석하고자 할 때 사용되는 인터페이스이다. 그리고 클라이언트에서 분석을 위해 요구되어지는 MIB 정의를 위해 현재 연구되어지고 있는 시스템 응용 MIB, 네트워크 서비스 모니터링 MIB, WWW 서버 MIB등을 사용하여 분석 항목을 설정하였고, 또한 서

버와 클라이언트 사이, 또한 서버와 웹 응용 서비스 트래픽 분석 시스템간의 메시지 형식을 새로이 정의하여 그들 간의 메시지 교환을 원활히 하도록 구현하였다. 따라서 본 논문에서 구현된 모델로서 웹 응용 서비스의 트래픽을 추출하고 그 흐름을 분석하여 최종 사용자에게 최적의 사용 환경을 구축할 수 있는 계기가 되었다. 특히 구현된 모델은 분산처리가 가능한 Web 및 자바 기술을 활용하여 시스템을 설계 및 구현하였으므로 다른 관리 시스템에 비해 사용 방법이 쉽고, 사용자 인터페이스가 웹 브라우저 이므로 분석 정보의 파악이 용이할 뿐만 아니라 분석 정보의 인지도가 높다는 장점이 있다. 그리고 관리자는 네트워크에 접속한 웹 브라우저만 있음 어디에서라도 이러한 분석 정보를 요구하거나 수집 요구를 할 수 있다.

웹 서버의 성능 상태를 모니터링하고 이를 분석하는 관리자 시스템을 개발함으로써, 웹 서버 관리의 효율성과 확장에 대한 설계를 지원할 수 있다.

참 고 문 헌

- [1] Jesper Nilausen, "Token Ring Network Management Performance Management", 1995.
- [2] Warren S.Meyers, "The Evolution of a Network-A Case Study of Network Development at John Wiley & Sons", 1995.
- [3] Addison-Wesly, "Network Managment Standards", Addison-Wesley Publishing Company, 1993.
- [4] 구자환 "네트워크 트래픽 분산 시스템의 통합 인터페이스 설계에 관한 연구", 1996.
- [5] Peter B.Danzig, Katia Obraczka, Anant Kumar, "An Analysis of Wide-Area Name Server Traffic", August, 1992.
- [6] Thoas Johannsen, Glenn Mansfield "A Study of FTP Traffic Estimation of the NonOptimality", December, 1994.
- [7] T.Berners-Lee R.Fielding, and H.Frystyk, "Hypertext Transfer Protocol-HTTP/1.0", 1996.
- [8] William Stalling, "SNMP, SNMP v2, and CMIP: The Practical Guide to Network Managment Standards",

[9] Alexander Clemm, "Adding Value to MIBs: A Relationship Layer for Management Platforms", 1995.

[10] Olga Havel and Ahmed Patel, "Design and Implementation of a Composite Performance Evaluation Model for Heterogeneous Network Management Applications", 1995.



한 정 수

1977년 성균관대학교 정보공학과 졸업(학사)

1997년~현재 성균관대학교 전기전자 및 컴퓨터공학부 대학원 석사과정

관심분야: 네트워크 관리



안 성 진

1988년 성균관대학교 정보공학과 졸업(학사)

1990년 성균관대학교 정보공학과 졸업(학사)

1990년 성균관대학교 대학원 정보공학과 졸업(석사)

1990년~1995년 시스템 공학 연구소 연구원

1995년~현재 성균관대학교 정보공학과 박사과정
관심분야: 네트워크 관리, 트래픽 분석, Unix 네트워킹



정 진 옥

1974년 성균관대학교 전자공학과 학사

1979년 성균관대학교 전자공학과 석사

1991년 서울대학교 계산통계학과 박사

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



박 형 우

1985년 서울시립대학교 전자공학과 졸업

1996년 성균관 산업과학 대학원 정보공학과 졸업(석사)

1997년~현재 성균관 대학원 박사과정(정보통신 전공)

1977년~현재 시스템공학연구소 고성능전산망 개발실장

관심분야: 고성능 컴퓨터 네트워크, 인터넷 서비스 관리, 전산망 보안