NFS 기반의 임베디드 장비 모니터링 시스템 설계에 관한 연구1)

김수현*, 한지환*, 배지혜*, 배성환*, 조한신**, 박윤용*
*선문대학교 전자계산학과
**선문대학교 컴퓨터정보학과
e-mail:cyber@sunmoon.ac.kr

A Study on the Monitoring System for Embedded Device Based on the NFS

Soo-Hyun Kim*, Ji-Hwan Han*, Ji-Hye Bae*,
Sung-Hwan Bae*, Han-Shin Jo**, Yoon-Young Park*
*Dept of Computer Science, Sun Moon University
**Dept of Computer and Information Science

요 약

임베디드 시스템은 현재 가전, 단말, 제어, 통신등 다양한 분야에서 응용되고 있으며 컴퓨터 관련 기술의 발전과 더불어 응용분야는 더욱 넓어질 것이다. 이러한 임베디드 시스템의 관리나 테스트를 위해 모니터링 프로그램이 필요하다. 임베디드 시스템은 일반 PC 시스템과 달리 많은 제약을 가지고 있다. 본 논문에서는 자원적 제약을 가진 임베디드에 활용되어지는 NFS를 이용한 임베디드 장비 모니터링에 적합한 MONETA(distributed MONitoring for Embedded TArget system)를 설계, 구현하였다. MONETA에서 제시하는 구조는 모니터링을 위한 서버, 모니터링하는 대상인 클라이언트, 시스템 사용자의 3계층 구조의 분산 모니터링 시스템으로 각 장비별 CPU, Moemory, Disk, 트래픽량을 그래프로 나타도록 구현된다.

1. 서론

임베디드 시스템은 큰 시스템의 일부의 구성 요소로서 동작되거나 사람의 개입 없이 동작하도록 기대되는 특별한 목적으로 설계된 하드웨어를 제어하기 위해 설계된 하드웨어와 소프트웨어가 내장되어 있는 시스템이다. 협의의 임베디드 시스템은 마이크로프로세서 또는 마이크로 컨트롤러를 내장하여 원래 제작자가 의도했던 기능만을 수행하도록 제작된 장치를 말한다. 임베디드 소프트웨어는 PC용 소프트웨어와 달리 실시간성, 고신뢰성 및 저전력을 요구하는 각종 제어기기, 정보기기 및 센서 장비등의 마이크로 프로세서 위에 탑재되는 소프트웨어로서, 시스템 소프트웨어와 미들웨어 등으로 구성되며 실시간성과 고신뢰성을 요구하는 특성을 가진다.

본 논문에서는 위와 같은 임베디드 소프트웨어의 특성을 고려하여 소프트웨어의 개발과정 뿐만 아니고 시험 및 상용화 단계에서도 사용할 수 있는 모니터링 시스템을 개발하고자 한다. 모니터링은 디버깅, 테스팅 그리고 컴퓨터 프로그램들의 성능 평가를 지원하는 광범위한 기능을 가진다. 이러한 모니터링 시스템은 시스템의 상태 정보 또는 프로세스들의 정보를 동적으로 수집, 해석, 표시하기 때문에 임베디드 시스템을 관리 및 테스팅 하기 위해서 필요하다. 특히 분산 환경에서의 모니터링 시스템들은 프로세스간의 정보를 동적으로 수집해서 사용자가 원하는 유용한 형태로 표시하여야 하기 때문에 분산 환경에서 실행되는 임베디드 시스템에서의 모니터링 작업은 더욱 어려운 문제들을 가지고 있다[1,2].

분산 환경에서의 모니터링 시스템 구현의 문제점은 첫번째 네트워크를 통한 정보의 전송 지연으로 인한 모니터링 정보의 일관성 문제와 상태 정보들이 부정확한 순서로 전송되는 문제점이 발생된다. 이것은 실시간으로 모니터링 정보를 유지 관리하는 것을 어렵게 하고, 정확한 정보 전송을 위해서 클럭 동기화가 필요하다. 또한, 다양한 임베디드 장비들을 모니터링하기 하는 과정에서 발생되는 많은 모니터링

1) 본 논문은 정보통신부 기초기술연구지원사업(C1-2002-068-0-4)으로 수행되었음.

정보들은 관리자들을 어렵게 할 수 있으므로, 정보의 필터링과 분석 기능이 필요하다. 또 다른 문제점은 모니터링 작업을 수행하기 위해 모니터링 시스템 자체가 관찰되는 시스템과 동일한 자원을 사용하기 때문에 모니터 되는 타겟 시스템의 성능을 저하시키는 간섭현상이 발생한다는 것이다. 이러한 문제들을 해결하기 위해 모니터링 시스템은 모니터링 정보 생성, 처리, 전송과 표시에 관한 수많은 기능을 제공해야 한다. 본 논문에서는 위에서 서술한 문제점들을 해결할 수 있는 분산 환경에서 실행되는 임베디드 장비들의 모니터링 시스템 모델을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 모니터링 도구에 대해 알아보고, 3장에서는 임베디드 시스템을 위한 모니터링 시스템 모델을 제시한다. 4장에서는 임베디드 장비들을 모니터링하기 위한 모니터링 시스템 “MONETA”의 구조와 실제 구현에 대해 설명한다. 제 5장은 결론과 앞으로의 과제에 대하여 설명한다.

2. 모니터링 도구 사례 조사

본 장에서는 기존에 사용되고 있는 모니터링 도구의 특징에 대해 살펴본다. [5,6,7]

MRTG[8,9]는 Perl 언어와 C언어로 구성되어 있어, UNIX와 Windows NT에서 동작을 하는 장점이 있다. 이 그래프는 html 화일에 넣어지기 때문에 일반적인 웹브라우저에서 볼 수 있다. SNMP MIB정보를 사용하여 libpcap과 같은 방법으로 패킷 캡처를 하지 않으므로 패킷손실이 없이 정확한 정보를 제공해 준다. MRTG는 일간은 물론 주간, 월간, 연간 트래픽 체크가 가능하다. 로그 화일은 자동으로 정리되어 시간이 지나도 로그 화일의 크기는 증가하지 않는다. 네트워크의 트래픽뿐만이 아니라 어떠한 SNMP 변수에 대해서도 모니터링이 가능하다. 그러나 MRTG는 네트워크 트래픽의 총 양에 대한 정보를 제공해 줄 뿐 어떤 호스트에서 어느 정도의 트래픽을 발생시켰는지, 어떤 프로토콜이 사용되었는지 등의 정보를 제공해 주지 못하는 단점이 있다.

Ntop은 Dri Luca가 1998 년부터 오픈 프로젝트로 개발하고 있는 실시간 네트워크 트래픽 모니터링 및 분석하는 시스템이다. libpcap을 기본으로 하여 네트워크 사용량을 알려주는 유닉스 툴로, 윈도우에도 동작한다. Ntop은 웹기반에서 호스트 정보, 프로토콜 정보, 어플리케이션 정보 등을 분석해 준다. 호스트 중심으로 분석 정보를 제공하여 특정 호스트에서 나가고 들어오는 패킷의 양이라든가 특정 호스트에서 사용되는 프로토콜을 확인하기 편리하다. 데이터는 막대와 파이 그래프들이 프로토콜의 이용과 패킷 사이즈의 분포를 나타낼 수 있으며, 모니터링을 통해 얻어진 데이터는 스프레드시트에 사용될 수 있도록 로그 파일로 만들어 진다.

tcpdump는 캘리포니아 대학 버클리교 로렌스 버클리 연구소의 Van Jacobson씨 등에 의해 개발된 프로그램이다. 로컬한 네트워크 상에서 주고 받는 패킷을 덤프하여 표시할 수 있다. 덤프란 검사를 위해 데이터 등을 출력하여 표시한 것을 말한다. 덤프된 데이터에서 네트워크 상의 프로토콜 동작을 관찰할 수 있다. tcpdump 프로그램은 대부분의 UNIX 시스템에 내장되어 있다.

Ethereal는 유닉스나 윈도우에서 네트워크 프로토콜을 분석할 수 있는 무료 분석 툴로, tcpdump와 마찬가지로 네트워크를 통해 교환되고 있는 패킷을 캡처 할 수 있다. 캡처한 데이터는 GUI에 의해 하드웨어 주소, IP주소, 프로토콜 등의 리스트로서 표시할 수 있다. 또 필터를 사용함으로써 특정한 프로토콜에 관한 데이터만 표시할 수 있다. ethereal은 GUI를 이용한 툴로, 사용하기 쉽게 되어 있지만 X Window 상이 아니면 이용할 수 없다.

UniMon은 여러 네트워크 노드의 트래픽을 합쳐서 분석 할 수 있는 장점이 있지만 장기간의 네트워트 트래픽 분석에는 적당하지 않다. NNStat 여러 네트워크 포인트에서 트래픽을 합쳐서 분석 할 수 있으며 사용자가 지정하는 장기간의 네트워크 트래픽 분석도 지원한다. SunOS 4.0 NIT(Network Interface Tap)인터페이스를 사용하는 시스템에만 지원되는 단점이 있다. eWatch는 상용 프로그램으로 프로토콜 분석 및 모니터링 소프트웨어이다. 스니퍼 소프트웨어 기반의 네트워크 분석기이다. 네트워크를 통해 전송되는 데이터를 포착하는데 사용될 수 있으며, 스니퍼가 장착된 라우터는 발신지 및 수신지 주소 뿐 아니라 패킷 내의 데이터까지도 읽을 수 있다. 아래의 <표 1>은 위에서 소개한 프로그램들을 정리한 것이다.

<표 1. 모니터링 프로그램 비교>

	분석방식	분석범위	여러노드에서 패킷캡처	호스트 분석기능	웹기반
MRTG	일괄처리	현재, 매시, 매일, 매주, 매달	N	N	Y
ntop	일괄처리	현재상황, 매시	N	Y	Y
tcpdump	실시간	현재상황	N	Y	N
ethereal	실시간, 일괄처리	현재상황	N	Y	N
UniMon	실시간	현재상황	Y	Y	N
NNStat	일괄처리	지정된시간	Y	Y	N
ewatch	실시간	현재상황	N	Y	N
스니퍼	실시간	현재상황	N	Y	N

3. 임베디드 장비를 위한 모니터링 시스템 모델

본 장에서는 분산 환경에서의 실행되는 임베디드 장비들을 모니터링하기 위한 모니터링 시스템 모델에 관하여 설명하기로 한다. 일반적으로 임베디드 시스템에서는 단일 장비의 형태로 존재하는 경우보다는 유선 또는 무선으로 연결되어 있는 여러 종류의 내장형 장비들이 분산 시스템 구조를 이루면서 존재한다. 이와 같은 분산 환경에서는 모니터링을

위한 서버와 모니터링을 하는 대상인 클라이언트 그리고 시스템 사용자들로 구분된다. 일반적으로 클라이언트는 모니터링 대상인 임베디드 장비가 되고, 서버는 임베디드 장비 중에 하나이거나 별도의 서버로 구성되어 질 수 있다. 이와 같은 분산 시스템의 모니터링 구조를 3계층의 모니터링 구조라 한다.

본 논문에서는 NFS(Network File System) 기반의 모니터링 시스템 모델을 제안한다. NFS 는 2개 이상의 컴퓨터 시스템이 서로의 파일 시스템을 마치 동일한 하나의 파일 시스템을 사용하는 것처럼 보이게 해주는 기능을 제공하는 프로토콜이며, 내장형 시스템의 개발 환경으로 응용될 수 있다.

또한, 임베디드 시스템에서는 메모리 등의 자원이 충분하지 못하기 때문에 자원을 많이 사용하는 프로세스보다는 스레드를 사용하여 프로그램 하는 것이 적합하다. 따라서 본 논문에서는 임의의 임베디드 장비 내에 단일 프로세스가 존재하고 프로세스 내에 다양한 스레드를 사용하여 통신하는 구조의 분산 시스템을 가정하기로 한다.

본 논문에서는 N개의 노드로 구성된 분산 시스템 내에 M개의 통신하는 프로세스(P_1,P_2,.....Pm)들로 구성된 분산 시스템을 가정한다(단, n < m). 시스템의 변환이나 실행은 전역 상태의 순서들로 표현된다. 또한, 각 전역 상태들은 모든 노드의 프로세스들의 현재 상태 합으로 표시되고, 형식적으로 프로세스의 상태와 전역상태는 아래와 같이 정의된다.

[정의 1] 프로세스 상태(Process State). 프로세스 P_j는 메인 스레드를 포함하여 1개 이상의 스레드로 구성되어 있고, 프로세스 상태는 $<s_j, T_j>$ 투플로 표기된다. s_j는 프로세스 P_j의 메인 스레드의 현재 상태를 표시하고 T_j는 P_j의 스레드들의 현재 상태이다. 즉 $T_j=<t_{j0}, t_{j1}, t_{j2},\cdots\cdots t_{jn},>$. 여기서 t_{je}는 시스템에서 프로세스 P_j의 스레드 t_e의 상태를 표시한다. 단, i≠j 이다.[4]

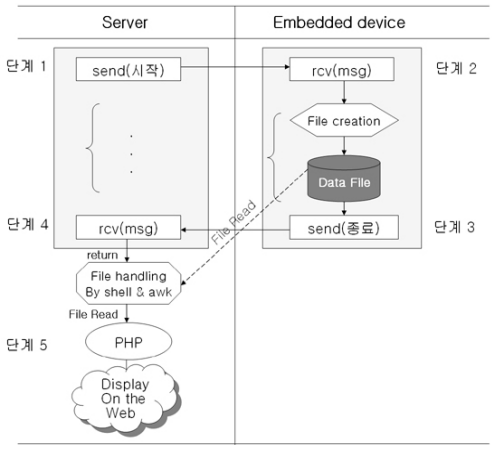
[정의 2] 스레드의 종류(The kinds of Thread). 임베디드 시스템내의 스레드는 메인스레드, 송신프럭시(send_proxy)와 수신프럭시(rcv_proxy) 스레드 그리고 *액션*스레드(*action*_thread)들로 구성되어 있다.

메인스레드는 시스템을 초기화 하고, 자신이 생성한 스레드에 관한 정보를 유지 관리하고, 이 정보들을 모니터링 서버에 전송하는 역할을 한다. 이 후에 일반적으로 유닉스 스레드들과 유사하게 메인 스레드들은 블록상태를 유지한다. rcv_proxy는 외부 프로세스로부터 메시지 또는 이벤트를 수신하여 해당 *action*_thread를 시작시키는 작업을 한다. *action*_thread들은 메인스레드에 의해 생성되어 블록상태로 존재하고, rcv_proxy에 의해 활성화(wake_up) 되어 관련된 모니터링 작업을 실행한 후 블록(block) 상태로 대기하게 된다. 따라서 *action*_thread들은 메인 스레드에 의해 생성되어 모니터링 이벤트를 대기하는 상태를 유지하게 된다. send_proxy는 프로세스 내의 스레드들이 외부의 프로세스에게 메시지를 전송하고자 하는 경우에 메시지 전송 작업을 대리하여 처리하는 역할을 한다.

메인 스레드는 자신의 프로세스 내에 각종 스레드들를 생성하고, rcv_proxy는 외부의 모니터링 이벤트를 수신하여 해당 *action*_thread들이 실행하도록 한다. *action*_thread 들은 미리 정의된 모니터링 작업을 수행하고, 수집된 정보를 NFS를 통해 서버의 파일에 저장하거나 send_proxy를 통해 서버에 전송하게 된다. 모든 스레드들은 정의된 작업을 완료한 후에는 새로운 이벤트에 의해 호출되어질 때까지 블록상태에서 대기하게 된다.

4. 구현

본 장에서는 임베디드 장비의 모니터링 시스템(MONETA : distributed **MON**itoring for **E**mbedded **TA**rget system)의 구현에 관하여 기술하였다. MONETA는 분산 환경에서 임베디드 장비들이 실행 중에 발생하는 각종 이벤트를 수집, 분석하여 인터넷 상에 도시하는 기능을 제공한다. 임베디드 장비인 목표시스템과 모니터링하는 서버시스템은 NFS로 연결되어 있고, 목표시스템에서 수집된 각종 이벤트 및 정보들을 NFS 상의 서버의 파일에 저장된다. (그림 1)은 모니터링 서버와 임베디드 장비 사이의 데이터 흐름을 도시한 것이다. 데이터흐름의 각 단계는 아래와 같다.



(그림 1)MONETA에서의 데이터 흐름도

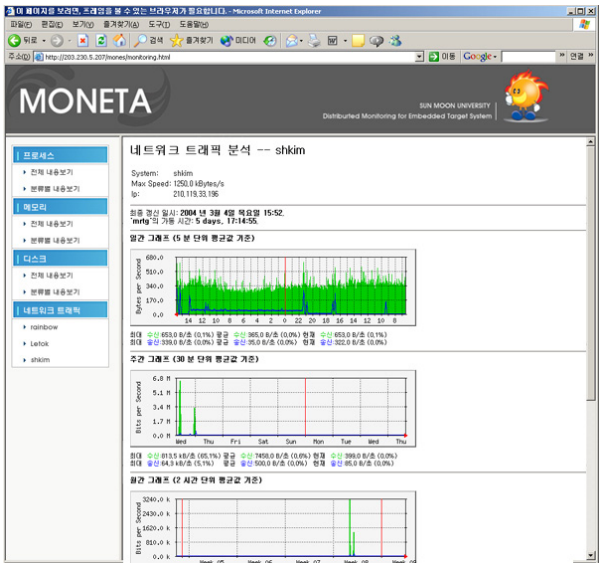
o 단계 1 : 서버는 send_proxy를 사용하여 임베디드 장비의 rcv_proxy에게 모니터링 정보를 수집하라는 제어 정보를 전송한다.

o 단계 2 : 임베디드 장비의 rcv_proxy는 제어 정보를 분석하여 관련된 이벤트의 상태 정보를 모니터링하고 수집된 정보를 NFS를 이용하여 서버의 파일에 저장한다.

o 단계 3 : 임베디드 장비의 send_proxy는 모니터링 작업의 종료를 서버에 알린다.

o 단계 4 : 서버의 rcv_proxy는 타켓 임베디드 장비로부터 종료 메시지를 수신하여 모니터링 정보들을 분석하고, 인터넷 상에서 표시하기 위한 형식으로 변환하여 저장한다.

o 단계 5 : 변환된 정보를 php를 이용하여 인터넷 상에 도시한다.



(그림 2) MONETA의 실행화면

(그림 2)는 MONETA의 실제 네트워크 트래픽을 모니터링한 실행화면으로 선택된 목록의 모니터링 정보를 보여준다. 크게 두개의 창으로 나누어져 있다. 왼쪽 창은 MONETA의 메뉴 화면으로 장비별 프로세스에 관한 정보, CPU, 메모리, 디스크 그리고 네트워크 트래픽에 관한 각종 정보들을 도시할 수 있다.오른쪽 창은 왼쪽 메뉴에서 선택한 장비의 시스템 이름, Max speeed, IP 정보와 함께 네트워크 트래픽의 총량을 분석하여 사용자들이 한눈에 볼 수 있도록 그래프로 나타낸다. 하루(5분단위 평균값), 일주일(30분단위 평균값), 한달(2시간단위 평균값), 일년(1일단위 평균값) 단위의 4개 그래프가 생성된다.

5. 결론

본 논문에서는 MONETA라는이름의 NFS기반 임베디드 장비의 모니터링 시스템의 설계 및 구현과정을 설명하였다. MONETA는 앞에서 설명한 프로세스, CPU, Memory, disk의 모니터링결과를 PHP를 이용하여 웹에서 출력하도록 설계되어 있다. 웹기반은 언제 어디서나 인터넷이 연결된 컴퓨터에서 시스템의 모니터링이 가능하다는 것 이외에도 사용자 프로그램의 배포 및 설치자 필요 없으며, 웹 서핑을 해본 사람이라면 누구나 친숙하게 사용할 수 있다는 장점이 있다.

모니터링 결과는 사용자가 한눈에 파악할 수 있도록 그래프화 하였고 일간, 주간, 월간, 년간의 그래프를 생성하여 장기간의 모니터링도 가능하다.

그러나 아직 시스템의 문제를 발견하고 해결하는 기능이 없으며 현재는 등록되어 있는 장비만의 모니터링이 가능하다. 앞으로 문제를 발견하고 해결하는 기능과 사용자가 추가되는 모니터링 장비를 직접 등록해주는 기능의 추가가 필요하다.

참고문헌

[1] J.Joyce etal. " Monitoring Distributed Systems," ACM Tran. Comm. Systems , Vol. 5, No. 2, May 1987, Pages 121-150.

[2] Masound Mansoun-Samani and Morris Sloman, "Monitoring Distributed Systems," IEEE Network Nov. 1993.

[3] Yoon-Young Park, Dong-Sun Lim, Sung-Hee Choi, Kyung-Oh Lee, Jung-Bae Lee, "A Study on the Monitoring System Models for the Embedded Systems", 2nd Asia Pacific International Symposium on Information Technology Jakarata, Indonesia

[4] Mohammad Zulkernine and Roudolph E. Seviora, "A Conpositional Approach to Monitoring Distributed Systems", Proceedings of the InternationalConference on Dependable Systems and Networks(DSN'02), 2002

[5] 권순선, 김재영, 홍원기, "웹 기반의 인터넷/인트라넷 네트워크 트래픽 모니터링 및 분석 시스템", KNOM Review 제 2권 제 1호, 1999년 4월, pp.1-10

[6] 주홍택, 홍원기 "내장형 웹 서버 기반의 네트워크 관리 구조 설계", Proc. of KNOM 2000 Conference, Taejeon, May, 2000, pp. 132-138.

[7] 홍순화, 김재영, 조범래, 홍원기, "분산 시스템 환경에서의 로드 밸런싱을 통한 웹기반 네트워크 트래픽 모니터링 및 분석", Proc. of KNOM 2001 Conference, Daejeon, Korea, May 24-25, 2001, pp. 198-205.

[8] MRTG Homepage, http://people.ee.ethz.ch/~oetiker/webtools/mrtg/

[9] superuserkorea Homepage, http://www.superuserkorea.co.kr