

P2P이용한 임베디드 시스템 네트워크

윤찬호* · 김철영* · 김동현** ·

*조선대학교 컴퓨터공학과

** 청암대학 컴퓨터정보과

chosun74@stmail.chosun.ac.kr

Embedded System Network of P2P Using

Chan Ho Yun* · Chul Young Kim* · Dong Hyun Kim** ·

*Dept. of Computer Engineering, Chosun Univ.

** Dept. of Computer Information, Suncheon CheongamCollege.

요 약

정보화 사회로 진행되어 감에 따라 임베디드 시스템에 관한 연구는 날로 발전해 가고 있다. 개인과 개인간의 정보공유 모델인 P2P의 중앙서버가 필요 없이 다른 사용자들과 정보를 주고받을 수 있는 장점을 활용 하여 기존 임베디드 시스템의 각각의 센서들은 게이트웨이에 연결하여 서버와 클라이언트 구성이 되었다면 각 센서 끼리 통신이 가능하도록 설계하여 클라이언트 끼리 네트워크를 구성하고자 한다.

최적화 저전력 초소형 임베디드 웹서버를 제안하고 각각 센서(균열센서, 열감지 센서등등) 설계 하여 빌딩의 붕괴, 산재 등등의 예상치 못한 사고가 발생시 건물 정보를 수집하는 센서 상호간에 네트워크로 연결시키며 웹포팅 및 웹에서 하드웨어 제어 및 포팅 및 부트로더에서 하드웨어 테스트 과정을 걸쳐 최종으로 결과를 얻는다.

I. 서 론

임베디드 웹서버는 여러 가지 형태의 입출력 장치와 지능형 정보 기기 들을 인터넷에 손쉽게 접속할 수 있게 하는 초소형 웹 서버로서, 임베디드 웹 서버의 적용 영역은 가정, 사무실, 빌딩 및 공장 등의 전 영역에 걸쳐 있다. 지능형 정보 기기를 인터넷에 접속하여 원격지에서 기기를 감시 및 제어를 하기 위해서는 웹 콘텐츠의 제공을 주목적으로 하는 일반 서버 컴퓨터와는 다른 임베디드 운영체계를 이용한 전용화 된 서버가 필요하다.

본 논문에서는 개인과 개인간의 정보공유 모델인 P2P의 중앙서버가 필요 없이 다른 사용자들과 정보를 주고받을 수 있는 장점을 활용 하여 기존 임베디드 시스템의 각각의 센서들은 게이트웨이에 연결하여 서버와 클라이언트 구성이었다면 각 센서 끼리 통신이 가능하도록 설계하여 클라이언트 끼리 네트워크를 구성하고자 한다. 최적화 저전력 초소형 임베디드 웹서버를 제안하고 각각 센서(균열센서, 열감지 센서등등) 설계 하여 빌딩의 붕괴, 산재 등등의 예상치 못한 사고가 발생시 건물 정보를 수집하는 센서 상호간에 네트워크로 연결시키며 웹포팅 및 웹에서 하드웨어 제어 및 포팅 및 부트로더에서 하드웨어 테스트 과정을 걸쳐 최종으로 결과를 얻는다. 또한 실시간기능을 위하여 리눅스 기반위에서 동작하는 실시간 운영체제인 RTLinux를 포팅하여 실시간 운영체제상에서의 네트워킹 기능을 구현하여 최적화된 임베디드 웹서버를 제안하고

자 한다.

II. Peer to Peer

P2P(Peer to Peer) 컴퓨팅은 공동 파일[6] 서버에 전적으로 의존하지 않으면서 각 PC 간의 직접적인 리소스 교환[1]을 지원하는 Application 및 Network 솔루션으로 정의할 수 있다. 그러므로 모두 Client/Server 양쪽으로 활동할 수 있는 "Peer"가 되며 이는 다양한 신규 Application을 위한 기초가 될 뿐만 아니라, 기존 인프라스트럭처에서 상당한 로드를 덜어냄으로써 값비싸고 성능에 방해가 되는 업그레이드의 필요성을 줄일 수 있는 장점을 가지고 있다.

그림 1은 하이브리드 구조중 중앙 집중식 중앙 집중식, 중앙 집중식 분산 방식, 중앙 집중식 링 방식 등 다양한 방식중 중앙 집중식 분산 방식을 표현하였다. 이 네 가지 방식의 장단점을 이용해 더욱 향상된 기능을 제공한다.

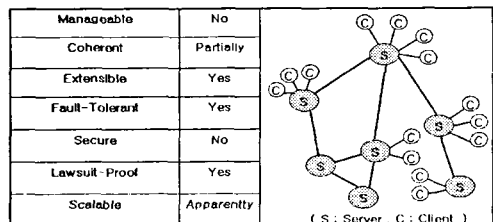


그림 1 하이브리드 시스템

그림 2은Hybrid P2P 방식의 서비스가 중앙관리 서버에 의해 검색 기능을 지원하는 것에 반해 Pure구조는[6] 서버가 존재하지 않는 방식으로 구현된 기술이다. 모든 Peer가 동등한 조건을 가지고 익명성이 보장된 형태로 자로나 리소스를 공유하는 구조이다

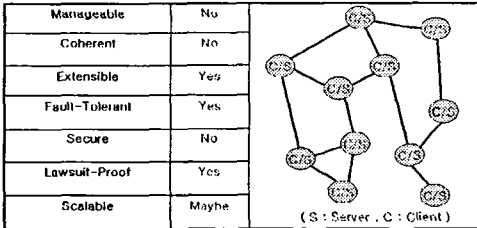


그림 2 순수 시스템

III. 실시간 웹서버 구현

2-1 실시간 시스템

실시간 시스템은 정해진 시간 내에 시스템이 결과를 출력하는 시스템을 말한다. 주어진 작업을 빨리 처리하는 것 보단 정해진 시간을 넘겨서는 안된다는 것이다.

임베디드 시스템이 실시간 적인 요소가 있기 때문에 실시간 시스템이라고 해도 과언이 아니다. 실시간 시스템은 결합 허용의 정도에 따라 세가지로 분류 될 수 있다. 우선 먼저, 경성 실시간에서는 모든 태스크의 조건을 충족시켜서 실행되어야 한다. 사소한 태스크 실행의 결함이 전체 시스템에 치명적인 결과를 초래할 수 있는 경우에 대한 시스템의 설계는 경성 실시간 시스템으로 해야한다. 두 번째로는 연성 실시간에서는 태스크의 조건을 만족시키지 못한다. 하더라도 치명적인 시스템 결함으로 이어지지 않고 종료시한을 넘겨서 수행된다고 하더라도 작업의 결과가 의미있는 경우를 말한다. 마지막으로 준경성 실시간은 경성과 연성의 중간 형태로서 종료시한을 넘겨서 작업을 마치는 경우는 무의미한 경우이고 그에 따른 시스템 결함도는 치명적이지 않은 경우를 말한다.

2-2 실시간 운영체제

임베디드 시스템이 가지는 특성 중 실시간 적인 요소를 충족하기 위해서 나온 운영체제가로 할 수 있다. 즉 실시간 운영체제는 임베디드 시스템의 근간이 되는 운영체제인 셈이다.

덧붙이면 임베디드 OS와 RTOS(Real Time OS)는 어느 정도 구별이 필요하다. 즉, 주어진 태스크를 달성하기 위한 목적뿐만 아니라 그 태스크의 종료시한을 만족 시키느냐 못 시키느냐의 문제에 중점을 둔 시스템이다. 또한 각 태스크의 응답시간을 실시간으로 처리 해주는 것이다. 그

렇지만 엄밀히 말한다면 멀티태스킹 환경에서 모든 작업을 즉각즉각 실시간으로 처리할 수는 없는 것이다. 각 태스크를 분할 처리하여 모든 작업을 실시간으로 처리하는 것처럼 보이는 것이다. 가장 중요한 종료시한을 넘기지 않도록 하기 위해서는 각 태스크에 대한 우선순위를 부여 해야 한다. 그래서 모든 태스크들의 우선순위에 따라서 작업이 스케줄되고 실행되어야 한다. 또 한 실행되는 중에도 더 높은 우선순위의 작업이 스케줄될 수 있기 때문에 선점이 가능해야 하는 점도 고려되어야 한다. 그림 1 에서는 리눅스를 이용한 일반적인 개발 환경을 표현 한것이다.

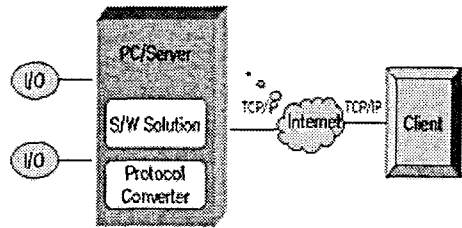


그림 3 Protocol Conversion Web Server

2-3 웹서버의 구조

대표적인 임베디드 웹서버의[2][3][4] 형태는 크게 네 가지 형태로 나눈다.

첫째, Protocol Conversion Web Server

I/O를 통하여 들어오는 정보를 중앙의 PC/Server에 저장한 상태에서 사용자가 접속 시에 정보를 제공하는 방식으로 Software에 의한 Web Server 형태이다.

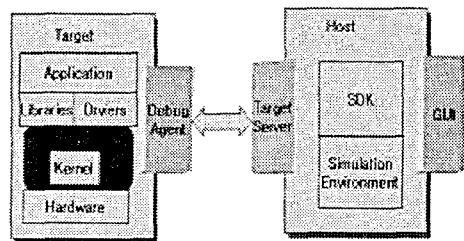


그림 4 임베디드 리눅스 개발 환경

둘째, Distributed Web Server

현재 대부분의 임베디드 웹서버들은 PC나 워크스테이션을 경유하여 네트워크 연결된다. 웹서버는 최소한의 요구되는 정보를 RS 232 통신 포트를 통해 PC에 보내고 PC는 웹서버의 정보를 수집하여 HTML페이지들을 전송한다. 이런 방식에서는 기기에 연산능력과 메모리 사이즈를 요구하지 않는다는 것이 장점이지만 PC를 사용함으로써 경제적, 공간적 단점을 가진다. 예로 간단한 센서의 정보를 위해 PC를 웹서버를 사용

한다면 비용도 비싸고 큰공간을 차지하게 되 비 효율적일 것이다. 이와 형태의 Web Server는 적은 정보와 분산형태의 제어를 필요하는 Home Networking 부분이나 빌딩내의 사무 기기 등의 감시 제어 분야에서 Machine Monitoring용으로 사용된다.

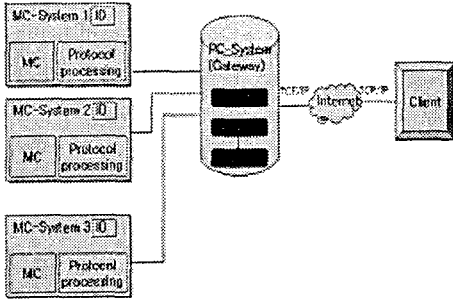


그림 4 Distributed Web Server

셋째, Protocol on Chip

TCP, UDP, IP, DHCP, ARP, ICMP와 같은 TCP/IP프로토콜 스택과 데이터 링크제어와 MAC 프로토콜과 같은 이더넷 프로토콜을 하나의 칩에 내장하는 형의 웹서버이다. TCP/IP 프로토콜 칩의 기간이 길어져 전체적인 개발 시간이 길어지지만, 소프트웨어로 구현하던 종래의 방법에 비해 사용자의 시스템에 이를 구현함으로써 인터넷을 빠르게 구현할 수 있게 해주어 소프트웨어 개발비용을 절감할 수 있다.

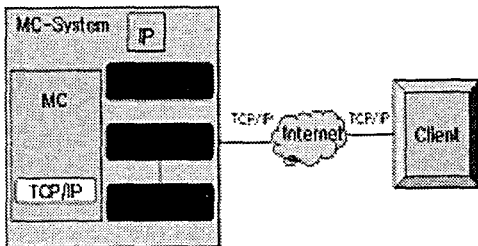


그림 5 Protocol on chip

넷째, Web Sever With/Without real time O/S

real time O/S 마이크로 초 이내의 응답의 정확도는 물론 신뢰성과 더불어 원하는 시간 안에 필요한 작업을 수행하여야 하는 시스템, 즉 항공·우주·군수 산업같이 세밀한 제어를 보장해야 하는 시스템이나 서버에 이용하는 운영체이다. 이러한 형태의 웹서버는 O/S의 제공으로 개발이 용이하다. 하지만 정밀성과 실시간성에서 우수하지만 고가의 라이센스 비용등 가격 면에서 소형의 저가형태의 제품생산에는 경쟁력이 약하다.

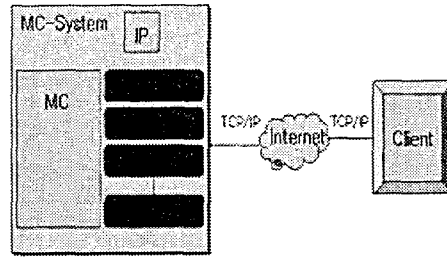


그림 6 Web Server with/without O/S

IV. 결 론

인터넷 통신 및 가정에서 쓰이는 ADSL등의 발달로 가정에서도 일반 가전제품에 대한 통신이 가능하게 된 임베디드 시스템에 대한 관심이 고조 되고 있는 가운데 최근 전자제품 등에 임베디드 프로세서를 장착하여 출시 하고 있다. 본 논문에서는 인터넷을 기반으로 원격으로 기기의 상태에 대한 정보를 접속된 시스템 관리자에게 제공하고, 관리자의 지시를 받아들이기 위한 임베디드 웹서버를 구축하였다. 인터넷을 통한 빌딩이나 장비 등의 원격제어를 할 수 있는 Web Server의 개발의 기반이 되었으며 이를 이용해 가전제품, 사무기기, 건물 경비 등의 모니터링 용으로 사용 될수 있다.

향후 연구 과제로 무선 인터넷의 발달로 핸드폰을 이용해 언제 어디서든 원하는 시간에 정보를 획득할수 있게되었다 기술적 측면에서 본 논문에서 구축된 임베디드 시스템을 무선네트워크, 블루투스, 등과 결합하여 무선원격 제어를 추가 연구가 필요하며. 건물의 위험 수위가 높아 지면 미리 저장해둔 건물내의 직원 핸드폰에 SMS 서비스를 이용하여 사고 위험을 문자로 알려주며 동시에 공개 방송, 게임 PC IP로 원격메시지를 전송 하여 빠른 시간에 대피 할 수 있도록 시스템을 개발하는 지속적인 연구가 필요하다.

참고문헌

- (1) Thomas Batt, Embedded Internet in Vielen Variationen, Elektronik <http://portolano.cswashington.edu/projects>.
- (2) protocol on chip <http://www.ubicom.com/products/processors> <http://www.connectone.com>
- (3) <http://www.sen.com> , 2002
- (4) imec, "RTLinux for StrongARM", <http://www.imec.be/rtlinux>
- (5) Michael Barr, "Programming Embedded Systems in C and C++", O'Reilly, 1999.
- (6) F. Balarin, L. Lavagno, P. Murthy, and A.

- S. Vincentelli, "Scheduling for Embedded Real Time Systems", IEEE, 1998.
- (7) Rick Lehrbaum, "Using Linux in Embedded and Real Time Systems",
<http://www.linuxdevices.com/articles/>
Feb 2000.
- (8) Real Time Magazine, "Comparison between VxWorks/x86 5.3.1, QNX 4.25 and pSOSytem/x86 2.2.6", April 1999.
- (9) Intel, "Intel StrongARM SA 1110 Microprocessor",
http://www.intel.com/design/pca/applications/processors/1110_brf.htm
- (10) PalmPalm Technology Inc. ,"Tynux Box II Hardware Manual