

Qplus 실시간 운영체제를 위한 임베디드 웹 서버 구현

이재호^o 김선자 *김성조, 선동국, 심형용
 한국전자통신연구원 정보가전연구부, *중앙대학교 컴퓨터 공학과
 email: bigleap@etri.re.kr, sjkim@cau.ac.kr

Implementing Embedded Web Server on Qplus RTOS

Jae-Ho Lee^o, Sun-Ja Kim, *Sung-Jo Kim, Dong-Kuk Sun, Heung-Young Sim
 Information Appliance Technology Dept. ,ETRI, *Chung Ang University

요약

Qplus 실시간 운영체제는 네트워크 지원을 위해 범용 목적으로 사용되는 4.4 BSD 기반의 통신 프로토콜을 가지고 있으므로 메모리 요구량이나 동작 면에서 자원이 제약된 소형 임베디드 시스템에는 적합하지 않다. 본 연구에서는 인터넷 접속을 위한 필수 기능만을 포함하여 정보가전기기와 같은 소형 시스템에 적합한 마이크로 TCP/IP를 Qplus의 네트워크 모듈로 이식하고, 이를 기반으로 사용자가 웹 브라우저를 통해 실험 보드를 제어할 수 있도록 임베디드 웹 서버와 CGI를 구현하였다.

1. 서론

인터넷의 확산과 더불어 중대형 규모의 시스템은 물론, 집안에서 쓰이는 가전제품과 같이 비교적 규모가 작은 임베디드 시스템들도 유무선을 통해 인터넷에 접속되어 데이터를 송/수신하는 것이 보편화 되고 있다. 이러한 요구가 증가함에 따라 표 1 과 같이 많은 국내의 연구기관 및 기업에서 내장형 시스템용 RTOS 에서 TCP/IP 프로토콜을 지원하기 위해 많은 노력을 기울이고 있다.

표 1. 임베디드 시스템을 위한 네트워크 모듈들

코드명	제조사	특징(지원 CPU 및 기반 RTOS)
NextGenIP	NEXGEN	Many CPUs/RTOSes, BSD4.4 Interface, Polling/Realtime Mode
NETX	ExpressLogic	Many CPUs, ThredX
smxNet	MicroDigital	x86/PowerPC, smx
RTIP-32	OnTime	Win32/NT compatible, RTKernel-32/16/C
CMX-TCP/IP	CMX system	Many CPUs, CMX-RTX
twiP/uIP	SICS	X86, 6502 CPU, FreeBSD/LINUX compatible
TargetTCP	Blunk Microsystems	Motolara, TargetOS
TwikNet	KADAK	Many CPUs, ATM RTOS
Turbo Treck	Elmic Systems	Turbo Treck AMX
ezTCP	솔네 시스템	PPP, ethernet supports

본 논문에서 ETRI 에서 개발된 쓰레드 기반의 Qplus 실시간 운영체제에 사용된 기존 BSD 기반의 네트워크 모듈을 대신하여, 중앙대에서 개발된 코드 크기가 작고 소형 임베디드

시스템에서 필수적으로 요구되는 기능만을 갖춘 마이크로 TCP/IP 를 사용하였다. 또한, 이를 기반으로 인터넷을 통하여 원격으로 정보가전을 제어할 수 있도록 소형 웹 서버와 CGI 를 구현하였다. 논문의 2 장에서 Qplus 의 기존 네트워크 모듈 구조와 새롭게 이식된 경량의 마이크로 TCP/IP 모듈에 대하여 간략히 기술하고, 각각의 코드 크기 및 성능을 비교하였다. 3 장에서는 네트워크 응용계층으로서 임베디드 웹 서버와 CGI 를 구현하고, 인텔의 SA-1110 CPU 를 탑재한 실험 보드를 웹 페이지를 통해 제어함으로써 구현물의 동작을 검증하고, 마지막 장에서 향후 연구 과제를 논의하면서 결론을 맺는다.

2. 관련 연구

2.1 BSD 기반의 Qplus 의 기존 네트워크 모듈 분석

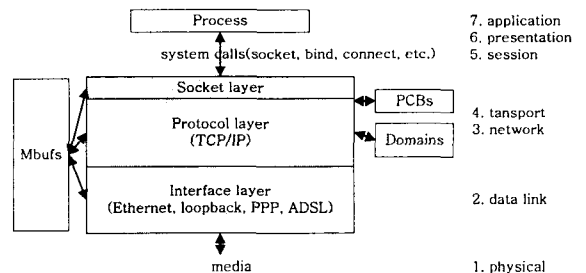


그림 1. Qplus 의 기존 네트워크 모듈

Qplus 의 네트워크 모듈은 그림 1 과 같이 4.4 BSD 코드를 기반으로 커널에 최적화 되도록 작성 되었으므로 일반적인

로 사용되는 대부분의 네트워크 기능을 포함하고 있다. 그러나, 이러한 구조가 정보대전 기기와 같은 소형 임베디드 시스템에 사용될 때에는 다음과 같은 단점을 갖는다.

- Full TCP/IP 지원: 대부분의 네트워킹 기능을 지원하므로 코드 사이즈가 크며, IP 단편화 개수에 제한을 두지 않아 재결합을 위한 메모리 요구량이 많고, 버퍼 관리 모듈과 재전송과 같이 복잡한 동작 메커니즘을 갖는다.

- mbuf 사용: 메모리를 관리를 위해 복잡한 mbuf 구조체를 사용하고 있으며, 이를 동적으로 할당 및 해제하기 위한 오버헤드가 존재한다.

- 실시간 지원 미흡: TCP/IP 프로토콜 자체가 실시간 서비스를 제공할 수 있는 프로토콜이 아니므로, UDP 기반의 RTP와 같은 실시간 지원 프로토콜이 추가적으로 요구된다.

그러므로, Qplus 운영체제가 실시간성을 지원하면서 소형 임베디드 시스템에 적합한 네트워킹 기능을 갖기 위해서는 기존 네트워크 모듈에서 불필요한 기능을 제거한 경량의 마이크로 TCP/IP가 필요하다.

2.2 마이크로 TCP/IP 네트워크 모듈 분석

실시간 운영체제에 적합하도록 구현된 마이크로 TCP/IP 프로토콜의 전체 구조는 그림 2와 같다. 수신된 프레임은 디바이스 드라이버에 의하여 읽혀지고, 이 프레임은 네트워크 매니저와 버퍼 관리자를 통하여 범용화되어 범용프레임에 저장된다. 범용프레임에 저장된 패킷은 네트워크 데몬에 의해 상위 계층으로 넘어간다. 한편, 상위 응용계층으로부터 전달된 패킷은 네트워크 데몬을 통하여 역시 범용화되어 범용화프레임에 저장되고, 네트워크 매니저를 통하여 네트워크 디바이스에게 전달된다.

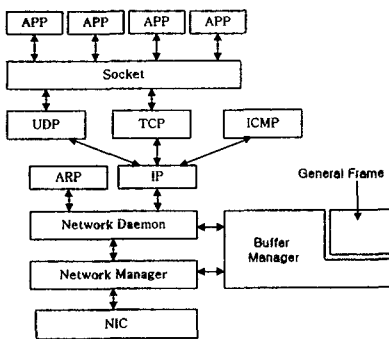


그림 2. 마이크로 TCP/IP의 구조

이러한 구조는 기존 BSD 기반의 네트워크 모듈구조에 비해 다음과 같은 부분이 개선되어 소형 임베디드 시스템에 사용되기에 적합하다.

- 재로 카피의 구현: 프로토콜 계층간의 데이터 복사는 전체 시스템의 성능을 저하시키고 메모리 낭비를 발생시키므로, 데이터 복사를 최소로 하기 위하여 모든 계층에서 접근할 수 있는 범용 프레임용 두어 각 프로토콜 단계에서 포인터를 이용하여 데이터를 전달한다.

- TCP/IP 데몬의 무한대기(Busy Waiting) 개선: 기존 방식은 네트워크 패킷을 처리하는 데몬들이 무한 루프를 돌면서 패킷의 송/수신 여부를 확인하여 처리하는 방식을 사용하고 있으나, 이는 시스템 성능 저하의 큰 원인이 될 수 있으므로, 세마포어를 이용한 시그널 방식을 사용하였다.

- IP 단편화 패킷 처리 개선: 2에서 6까지 조절이 가능한 단편화 기능을 구현하여, MTU 값이 1500 bytes 인 이더넷 환경에서 단편화가 최대 6번 발생하도록 허용하였다. 이는 최대 크기가 8192 bytes를 갖는 NFS 데이터그램을 처리할 수 있다.

- 소켓 API 구현: 기존 BSD 계열을 위해 작성된 응용 프로그램들과 호환성을 유지하고, 개발자들이 기존 방식과 큰 차이점 없이 코드를 작성할 수 있도록 소켓 API를 제공하였다.

2.3 코드 사이즈 및 성능 비교

그림 3은 기존의 방식과 마이크로 TCP/IP의 코드 사이즈와 성능을 비교한 것이다. 구현된 마이크로 TCP/IP의 코드 크기는 기존 BSD 기반의 네트워크 모듈에 비해 1/4 정도로 줄었으나, 성능 면에서 살펴볼 때, UDP 기반의 http 서비스를 통한 데이터 전송시간은 비슷한 결과를 보였다[5]. 한편, TCP 기반의 echo 서비스의 경우 전송되는 데이터 양이 하나의 mbuf로서 처리될 수 있는 108 bytes 이내 일 때에는 기존 방식에 비해 성능이 다소 떨어졌으나, 그 이상의 데이터를 전송할 때에는 대체적으로 우수한 성능을 보이므로 임베디드 시스템에 사용하기에 적합함을 알 수 있다[5].

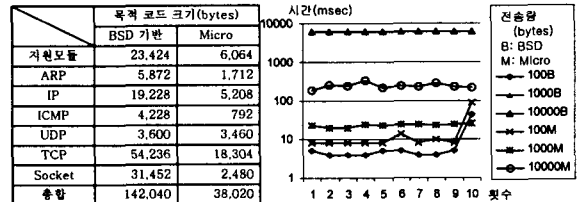


그림 3. 네트워크 코드 및 성능 비교

3. 임베디드 웹 서버 및 CGI 처리모듈 구현

3.1 임베디드 웹 서버의 특징

웹 브라우저를 통해 인터넷에 연결된 소형 임베디드 시스템을 제어하기 위해 마이크로 TCP/IP를 기반으로 간단한 웹 서버와 CGI를 구현하였으며, 그 특징은 다음과 같다.

- 소켓 API 사용: 소켓 API를 사용하여 수신 시에 불필요한 데이터 복사를 방지 하였고, 대용량의 이미지도 효율적으로 전송 및 재전송이 가능하다.

- 파일 시스템 지원: Qplus의 풀래시 파일 시스템을 사용하여 사용자에게 제공할 웹 콘텐츠를 저장한다.

- 소켓 GET 메소드 지원: 필수 기능만을 갖춘 임베디드 웹 서버이므로 전달인자 파싱이 POST 메소드에 비해 간단한 GET 메소드만 지원하도록 구현하였다.

- HTTP 헤더의 해석: GET 메소드만 지원하므로 HTTP 헤더

중 요청 URI 만 해석한다.

● 간단한 오류처리: 범용 목적이 아닌 소형 시스템을 위한 웹 서버이므로 캐쉬나 리다이렉트, 인증등과 관계된 오류는 고려하지 않고, 간단히 " HTTP 404 not found" 오류만 처리하여 코드를 간략히 하였다.

● 폼(Form) 인터페이스 지원: 일반적으로 CGI 와 같은 의미로 사용되며 웹 페이지로부터 데이터를 받아들이 서버쪽의 응용 프로그램을 구동시켜, 웹으로부터 받아들이 데이터의 처리를 가능하게 한다. 본 연구의 임베디드 웹 서버에서도 간단한 CGI 를 사용하여 원격에서 웹 페이지를 통해 임베디드 시스템의 동작을 변경시킬 수 있는 기능을 지원한다.

3.2 임베디드 웹 서버 및 CGI 의 내부 동작

그림 4 는 웹 서버 및 CGI 를 처리하는 과정을 나타낸 것으로서, TCP 모듈로부터 웹 서버가 사용하는 포트 번호로 수신된 요청 메시지를 HTTP 핸들러가 전달 받아 GET 메소드 일 경우 요청 파일을 풀래시 파일로부터 찾아서 전송해준다. CGI 요청이 전송된 경우, 웹 서버는 해당 CGI 태스크를 실행시키고 세마포어를 이용하여 태스크가 전달된 인자를 분석하여 작업을 마친 후 임시파일에 결과를 저장할 수 있도록 대기한다. 이후, 태스크가 세마포어에 시그널을 보내면 HTTP 핸들러는 임시파일에 저장된 결과를 TCP 모듈에 되돌려 줌으로써 CGI 요청에 대한 처리를 마친다.

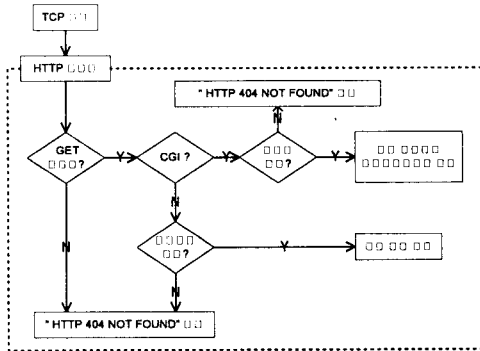


그림 4. 웹 서버 및 CGI 의 동작 과정

3.3 실험 환경 및 결과

실험 검증을 위해 마이크로 TCP/IP 네트워크 모듈을 가진 Qplus 커널과 임베디드 웹 서버가 이식된 하드웨어 사양은 표 2 와 같다.

[표 2] 실험용 하드웨어 사양

CPU	Intel StrongARM SA-1110	
OS	Qplus RTOS	
Memory	SDRAM	32M
	FLASH	16M
시리얼 인터페이스	Three 9-pin RS-232C	
이더넷	Cirrus Logic CS8900A	

구현된 웹 서버와 CGI 동작 테스트를 위해서, 그림 5 와 같이 인터넷 웹 브라우저를 통해 실험 보드에서 구동하는 웹

서버에 접속하고, 웹 페이지에 표시된 버튼을 통하여 CGI 요청을 보냄으로써 실험 보드에 장착된 LED 를 on 또는 off 시킬 수 있다. LED 는 SA-1110 CPU 에서 제공하는 GPIO 포트에 연결하였다. 여기서 동작하는 LED 가 집안의 가전기기를을 각각 연결하는 제어 라인이라고 가정할 때, 간단한 소프트웨어 만으로 집 밖에서도 인터넷 접속을 통하여 집안에서 동작하는 여러 기기들을 모니터링 하고 제어하는데 응용될 수 있음을 의미한다.

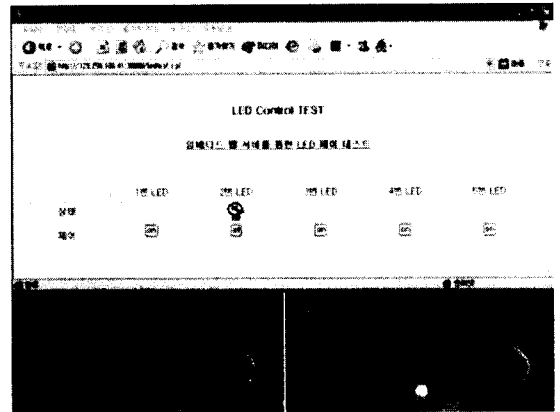


그림 5. 웹 서버 및 CGI 동작모습

4. 결론 및 향후 연구과제

본 연구는 실시간성을 필요로 하는 소형 임베디드 시스템에서 사용될 수 있는 작은 크기의 마이크로 TCP/IP 를 Qplus 실시간 운영체제에 이식하여, 비교적 자원이 제한된 소형 임베디드 시스템을 큰 오버헤드 없이 인터넷에 연결할 수 있도록 하였다. 뿐만 아니라 인터넷 웹 브라우저를 통해 원격으로 임베디드 시스템에 부착된 주변 하드웨어들을 제어할 수 있도록 소형 웹 서버와 CGI 핸들러를 구현하였다.

한편, 구현된 네트워크 프로토콜들은 IPv4 만 지원하고 있으나, 정보가전기기에서 이러한 서비스를 사용하기 위해서는 각각의 기기들에 할당해야 할 수많은 IP 가 필요하므로, IPv4 에 비해 풍부한 IP 자원을 가진 IPv6 를 지원하기 위한 연구도 추가로 요구될 것이다.

참고 문헌

- [1] ETRI, " 조립형 실시간 OS 개발", Dec., 2000
- [2] Intel, StrongARM SA-1100 Microprocessor Technical Reference Manual, December 1998
- [3] Jeremy Bentham, " TCP/IP Lean: Web Servers for Embedded Systems 2nd Edition", CMP Books, 2002
- [4] 심형용 외 3 인, " RTOS 를 위한 TCP/IP 프로토콜 스택의 구현", 정보처리학회 추계학술대회, 제 29 권 2 호, pp427-429
- [5] 김성조 외 5 인, " 인터넷 정보가전용 실시간 운영체제를 위한 네트워크 모듈에 관한 연구", 2002 년 ETRI 최종보고서