

배전설비를 위한 임베디드 무선랜 기술 개발

(Development of the Embedded Wireless LAN Technology for Power Utility Equipments)

손수국 · 우종정*

(Su-Goog Shon · Jong-Jung Woo)

요 약

본 논문에서는 배전설비 중 주상 변압기에 설치되어 운전되는 기존의 각종 제어설비를 무선 네트워크를 경유하여 제어하도록 하는 임베디드 무선랜 제어기술에 대하여 기술한다. 임베디드 무선랜 제어기는 Prism 2.5 칩셋을 포함한 PCMCIA 타입의 무선랜 네트워크 카드와 8비트 마이크로프로세서 Atmega128을 주요 구성요소로 이루어져 있다. 네트워크를 통한 제어를 위하여 임베디드 무선랜 제어기에는 IP, TCP, UDP, ICMP 같은 TCP/IP 스택과 기존 제어기를 인터페이스 하기 위한 Telnet, X/Z 모뎀의 개발내용을 포함된다. TCP/IP 네트워크와 모뎀을 통한 직렬 통신 사이의 데이터 구조와 흐름제어 차이에 대한 해결 방법과 램(RAM) 사용량을 최소화하기 위한 알고리즘을 제안한다. 끝으로 개발된 프로토콜들의 동작을 측정하기 위하여 DU[®], CommViewFi[®] 등이 사용된다. 개발결과 3,381 바이트 램(RAM)만을 사용하여 호스트 간에 상호 호환성을 잃지 않고 임베디드 무선랜 제어기가 동작된다.

Abstract

This paper describes the development of an embedded wireless LAN controller which can be in parallel operated with an existing utility controller. The embedded controller mainly consists of Prism[®] 2.5 chip set and Atmega 128 microcontroller. In order to communicate over the network, the controller including TCP/IP stack (IP, TCP, UDP, and ICMP), telnet, and X/Z modem has been developed. For a specific application, we have proposed an special method to convert data structure between TCP/IP and X/Z modem and a data buffer algorithm to minimize the RAM memory usage. Finally, the correctness and performance of the protocols are tested and verified using CommView[®] and DU[®]. The development is satisfactorily operated only for 3,381 bytes of RAM usage without sacrificing interoperability between hosts.

Key Words : Embedded controller, WLAN, Power utility, TCP/IP

1. 서 론

배전/변전 설비에는 FTU, 자동부하 절체 스위치, 릴레이, 스위치, 차단기, 리클로우저 등의 다양한 장치들이 존재하고 있으며, 이들은 배전자동화, 부하제어, 정전구간 최소화, 변전 자동화 업무 등에 사용되

* 교신저자 : 성신여자대학교 컴퓨터정보학부 교수

Tel : 031-220-2615, Fax : 031-229-8281

E-mail : sshon@suwon.ac.kr

접수일자 : 2006년 9월 26일

1차심사 : 2006년 10월 9일

심사완료 : 2006년 10월 20일

고 있다.

배전 설비 중 FTU(Feeder Terminal Unit) 장비는 상고장, 접지고장, 정상과 순간 고장 전류 간의 구분 등의 기능을 통한 고장구간 최소화, 빠른 사고 회복을 위하여 사용되는 중요한 배전자동화 설비 중의 하나이다. 배전계통의 특정지점에서 선로의 개폐 상태정보에 대한 모니터링, 전기 값 측정, 부하제어 스위치의 제어 등과 같은 배전자동화를 위한 중요한 기능을 수행하는 장치이다. 그 밖에도 장비의 관리를 위하여 고장 파형과 작동 횟수 등에 대한 기록기능을 갖고 있으며, 장비의 각종 동작 설정을 위한 설정 파일을 갖고 있다. 장비의 펌웨어를 업/다운 할 수 있는 기능도 포함되어 있다.

그러나 저장 기록된 데이터, 설정 파일, 펌웨어 업데이트 기능 등은 SCADA 등의 기능을 통하여 제어되지 않는다. 예를 들어 고장파형 데이터를 FTU 장비에서 다운로드 하여 분석하고자 한다면, 전신주에 올라가 노트북 같은 터미널을 통하여 다운로드 받아와야 한다. 펌웨어의 업그레이드, 장비 설정 값의 변경 경우에도 역시 전신주에 올라가 작업을 수행하여야 한다.

이러한 FTU는 정상적 동작 시에는 관리목적, 또는 배전계통 고장 발생 같은 상황 시에는 문제점의 분석과 해결을 위하여 관리자로부터 접속되고 관리되고 있다. 그런데 이러한 기능은 SCADA에 의하여 지원되지 않으므로 전신주로의 승주가 이루어져야 달성될 수 있다. 그런데 계통 고장 발생 등은 주로 비가 오고 바람이 부는 날씨, 또는 눈이 오고 바람이 강하게 부는 영하의 추운 날씨에 자주 발생한다. 이러한 상황에서 관리자가 승주하는 것은 생명의 위협을 느끼게 하며 또한 기피하는 업무가 되어 가고 있다.

이처럼 배전계통에 존재하는 설비들의 관리를 위한 개선사항이 필요하다. 본 논문에서는 이러한 상황에서 적용될 수 있는 배전설비에 대한 새로운 관리 기술을 제안하고 개발한다. 네트워크를 통하여 FTU 장비의 관리에 필요한 기록 데이터, 펌웨어, 설정파일 등을 원격으로 가능하도록 한다. 전신주에 설치되어 운전되고 있으므로 지상에서 관리가 가능하도록 무선랜 IEEE 802.11b를 통신 수단으로 사용

한다.

기존 FTU 제어기에는 관리 목적을 위하여 직렬 통신포트(RS-232)와 X/Z 모뎀이 동작되고 있다. 이는 개발에서 주요한 고려사항이 된다. 그리고 무선랜 제어장치에 대한 사용의 편리성, 네트워크 기반 운전 가능성, 그리고 무선모듈의 보안성이 고려된다.

일반적인 무선 RF 모듈은 데이터 전송거리의 한계, 주변상황에 따른 데이터 전송 오류의 발생, 예를 들어서 자동차 또는 사람의 주행에 따른 RF 전자파의 반사와 간섭으로 데이터 전송에 에러가 발생하는 문제점이 있다. 또한 일반 무선 RF 모듈은 시스템에 대한 외부인 침입에 대한 보안의 취약점이 있다. 이 밖에도 블루투스, 적외선 통신방식 등이 검토되었으나, 차후 제품의 유지 및 네트워크로의 확장성 등 면에서 적용하기에 문제점이 있다.

본 논문에서는 앞의 고려사항을 참고로 FTU같은 배전설비를 네트워크를 통하여 관리가 가능하도록 하는 시스템을 제안한다. 기존 배전설비의 제어기는 본 논문에서 제안한 IEEE 802.11b 임베디드 무선랜 제어기와 인터페이스 된다. 개발되는 무선랜 제어기는 주변의 AP(Access Point)와 인프라 또는 임의(ad-hoc) 네트워크를 구성하여 원격으로 연결될 수 있다. 결국 배전설비 제어기 내부의 각종 기록 데이터 또는 펌웨어가 네트워크를 통하여 원격에서 관리될 수 있다.

2. 임베디드 무선랜 제어기의 제안

FTU는 그림 1과 같이 전신주에 설치되어 있다. FTU는 배전자동화의 단말 장치로서 동작하는데, 내부에 내장된 마이크로프로세서에 의하여 각종의 제어기능을 수행하는 장치이다. 이러한 FTU 관리 업무를 위하여 외부와의 인터페이스인 UART 직렬 통신 포트가 있다[1]. 이 직렬 포트를 통하여 프로세서 펌웨어, FTU에 대한 설정파일, 고장 파형 등에 대한 업/다운로드를 할 수 있도록 한다. 관리자는 직렬통신 포트에 접속하기 위하여 승주를 하여야 하는 불편함이 있다.

배전설비를 위한 임베디드 무선랜 기술 개발

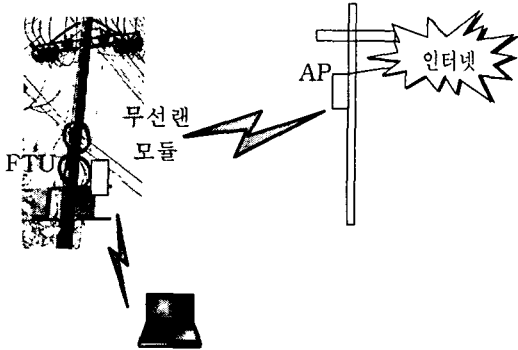


그림 1. 제안된 임베디드 무선랜 시스템
Fig 1. Proposed embedded WLAN system

그림 1에는 본 논문에서 제시한 배전설비의 관리를 위한 무선랜 기반의 시스템을 제시하고 있다. 임베디드 무선랜 제어기는 UART 포트를 통하여 FTU와 연결되며, 또한 무선 네트워크를 통하여 인터넷에 연결이 가능하다. 즉 무선네트워크를 통하여 임베디드 무선랜 제어기, 직렬통신, 및 FTU 제어기로의 접속이 이루어질 수 있다.

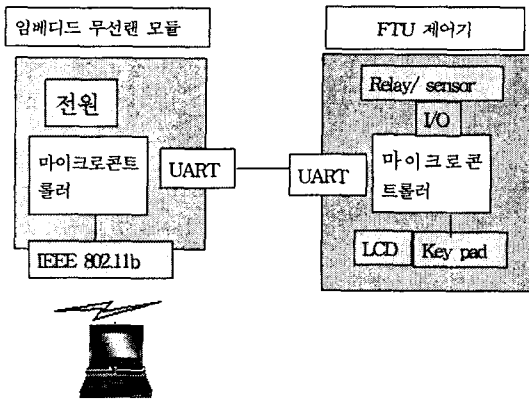


그림 2. 임베디드 무선랜 구성
Fig. 2. Embedded WLAN system

임베디드 무선랜 모듈 시스템 하드웨어 구성은 그림 2와 같다. 마이크로컨트롤러부, 통신부, 및 전원부로 구성된다. 마이크로컨트롤러 부에서는 입출력 신호처리, 프로그램 및 데이터의 보관, 및 시스템의 중앙관리 업무를 수행한다. 통신부는 네트워크와 데이터통신을 수행하는 기능을, 전원부는 시스템에 필

요한 에너지를 공급한다. 임베디드 무선랜 제어기에서 통신부는 인터넷 접속 및 FTU의 UART 통신 인터페이스 기능을 갖는다.

무선랜 제어기의 소프트웨어 개발에는 무선랜카드 제어 드라이버, TCP(Transmission Control Protocol), UDP(User Datagram Protocol), ARP(Address Resolution Protocol), IP(Internet Protocol) 등의 프로토콜 스택이 포함된다.

임베디드 무선랜 기술은 32비트 암(ARM : Advanced RISC Machine)을 기반으로 리눅스 또는 윈도우 같은 상업용 운영체제를 포팅하여 네트워크 기능을 구현하는 방법이 일반적이나, 이러한 기술을 본 논문에 적용할 때는 전력소모 및 가격 경쟁력에서 문제가 된다[2]. 본 논문에서는 운영체제를 사용하지 않으면서 8비트 마이크로컨트롤러에 간략화된 TCP/IP 스택을 구현하는 시스템을 개발한다. 무선랜 제어기의 중요 칩셋은 Conexant사의 Prism 2.5 칩셋을 사용한다[3].

제안된 임베디드 무선랜 제어기를 개발하는데 있어 중요한 개발사항은 제한된 메모리를 갖는 8비트 마이크로컨트롤러에 구현하는 것이다. 물론 이렇게 구현된 프로토콜은 인터넷의 다른 호스트와 상호통신에 문제가 없어야 한다. 호환성의 지원을 위한 국제표준이 있는데, RFC 1122[4]는 호스트에서의 연결계층, 네트워크계층, 및 전달계층에 대한 기준을, 그리고 RFC1123는 응용계층에서의 요구조건을 규정하고 있다.

인터넷 프로토콜 구현의 역사를 조사하면 버클리 BSD TCP/IP로부터 인터넷 프로토콜이 파생되어 많이 발전하여 왔다. 완벽한 TCP/IP 구현을 위하여 보통 수백 KB의 프로그램메모리 ROM과 수백 KB의 데이터 메모리 RAM이 필요하다. 하지만 수십 KB의 ROM과 수KB 바이트 RAM을 갖는 마이크로컨트롤러에 TCP/IP의 전체기능을 구현하는 것은 어려운 일이다.

임베디드 시스템을 위하여 소형으로 TCP/IP를 구현한 몇 예가 있다. 8비트에서 32비트 RISC 아키텍처까지 다양한 목표시스템이 있다. 8비트 마이크로컨트롤러를 기반으로 프로토콜을 구현한 예로서 uIP[5-6]이 있는데, Atmega128 AVR에서 5,164B의

프로그램 ROM을 사용하여 구현하였다. IPic[7]은 성냥 끝 크기의 칩에 서버를 구현하였으며, Jeremy Bentham의 PICmicro 스택[8]과 Atmel TCP/IP 스택[9]도 구현 예들이다. 이들은 간략화된 소형의 TCP/IP를 또는 전체 기능의 TCP/IP를 구현하였다. 32비트 인텔 x386 CPU을 사용하여 인터넷 기반 RF 태그리더를 개발한 경험과[10], 8비트 마이크로컨트롤러를 사용하여 운영체제 없이 인터넷기반 보일러 시스템을 개발한 연구결과도 있다[11].

3. 임베디드 TCP/IP 프로토콜 구현

3.1 개발환경

마이크로프로세서로서 ATmega128L을 사용한다. 마이크로프로세서에 대한 프로그램 개발환경은 컴파일러로 CodeVision Wizard AVR V1.23.9b이 사용된다. 클럭 주파수는 4.9512[MHz], 소형의 메모리 모델, 128[KB] 플래시 메모리, 내부 SRAM 크기는 4[KB] 및 데이터 스택 크기는 1,024[B]이다. 프로그램 다운로드에는 JTAG을 사용하여 이루어진다. 무선 랜은 IEEE 802.11b과 호환하는 Prism 칩셋 모듈을 사용한다.

3.2 제안된 임베디드 무선랜 데이터구조

일반적인 TCP/IP 프로토콜 스택은 물리계층에서 응용프로그램 계층까지 다수의 계층으로 구성된다 [12]. 개별 계층은 주변계층과의 인터페이스를 위하여 입출력 데이터 버퍼를 갖는 것으로 모델링이 된다. 상하 계층은 그 버퍼에 데이터 읽기/쓰기를 수행한다.

본 연구에서는 RAM 메모리가 제약된 상황에서 TCP/IP 프로토콜을 구현하는 방안을 제시한다. 스택 및 버퍼를 위한 데이터 구조를 4KB의 한도 내에서 진행하여야 한다. 이더넷에서 사용하는 데이터 프레임 크기가 1,526[B]인 것을 고려하면, 무선 측 송수신 버퍼에 각각 1,526[B], 직렬통신 측 송수신 버퍼에 각각 1,526[B]를 설정하면 최소한 6,104[B]가 필요하다. 이밖에도 스택을 위한 메모리를 고려한다

면 4[KB]의 한도 내에서 구현할 수 없다.

그림 3은 제안된 데이터 구조를 나타낸다. 마이크로컨트롤러의 RAM 메모리가 다수의 버퍼로 구성되어 사용된다. 무선 모듈(Prism 2.5 칩셋)과의 인터페이스를 위하여 240[B] 크기의 송수신 버퍼(WTxBuff, WRxBuff)를 각각 할당하고 있다. 전력 설비 제어기와 직렬통신 인터페이스를 위하여 제어기 측으로의 송신을 위한 400[B] 버퍼(STxBuff), 그리고 수신처리를 위한 1,526[B] 버퍼(SRxBuff)를 할당하고 있다. 제어기로부터의 수신된 데이터를 처리하는 부분에 640[B] 크기의 중간버퍼(Inter_Queue)를 두고 있는데, 이는 수신된 데이터를 무선 랜으로 송신할 때 만약에 오류가 발생할 수 있는데 데이터가 무사히 전달될 때까지 저장하기 위한 중간 버퍼로 사용된다. 일종의 오류제어를 위하여 사용되는 버퍼이다.

전기설비 제어기 측으로부터 직렬통신 채널을 통하여 데이터가 입력(심지어 1바이트)될 때마다 무선 랜으로 전송하는 것은 데이터 프레임의 오버헤드에 따른 처리율(throughput)을 감소시키게 된다. 따라서 수신버퍼(SRxBuff)에 어느 정도 크기의 데이터가 저장될 때까지 일정시간(약 200[ms]) 기다리도록 구현한다. RAM 사용을 줄이기 위하여 제안된 송신 버퍼(WTxBuff)로부터 무선랜 Prism 칩 내부 버퍼 RAM에 데이터를 업로드하는 드라이버 알고리즘을 개발한다.

또한 RAM 사용을 줄이기 위하여 Prism 칩 내부 버퍼 RAM으로부터 조금씩 데이터 프레임을 다운로드하는 알고리즘을 개발한다. Prism칩[13]의 내부 메모리 RAM에 저장된 1,500[B] 정도의 이더넷 프레임을 최대 240[B]씩 수신버퍼(WRxBuff)로 다운로드하여 처리하는 특별한 알고리즘이 필요하다. Prism 내부 RAM에 저장된 프레임으로부터 수신된 데이터에 대하여 TCP/IP 프로토콜을 직접 처리한다. 이 수신데이터는 처리된 후 직렬통신 채널의 버퍼(STxBuff)에 이동된 후 직렬통신 포트에 전송된다.

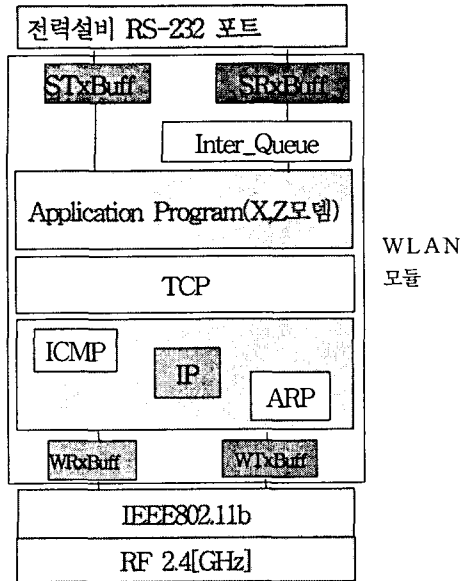


그림 3. 제안된 프로토콜 및 버퍼 구조
Fig. 3. Proposed protocol and buffer structure

직렬통신 속도는 9,600[bps], 무선랜 속도는 1, 2, 5.5, 11[Mbps] 속도로 양방향의 통신채널에 속도차이에 의한 흐름제어 문제가 발생할 수 있다.

직렬통신 측에서 데이터가 입력되면 수신버퍼(SRxBuff)에 저장한다. 이 크기는 데이터 프레임 1개를 저장할 수 있는 용량이다. 버퍼의 용량이 지정된 한계치 수준이상으로 차게 되면 전기설비 제어기에 더 이상 데이터를 전송하지 않도록 CTS(Clear To Send) 신호를 보내어 흐름제어를 수행한다. 반면에 전기설비로의 데이터 전송 경우는 특별한 흐름제어 기술을 사용하고 있지 않다. 개발한 임베디드 무선랜 장비보다 전기설비 측의 데이터 처리 속도가 빨라서 흐름제어를 위한 RTS(Request To Send)를 사용할 필요가 없다고 가정한다. 즉 버퍼(STxBuff)에 데이터가 존재하는 한 언제든지 데이터를 송신할 수 있다.

이러한 설계를 통하여 버퍼와 데이터 스택을 포함한 전체 RAM 사용량은 3,381[B]로서 4[KB]의 RAM 한도 내에서 달성하였다. 또한 흐름제어와 오류제어를 구현하였다.

3.3 임베디드 TCP/IP 프로토콜의 구현

TCP/IP의 전체 기능을 구현하는 것은 역시 RAM 소비의 문제를 야기한다. TCP/IP 스택 중에서 어떤 부분을 생략하여 간략화 할 것인가를 판단하여야 한다. 그럼에도 불구하고 임베디드 무선랜 제어기는 다른 네트워크의 호스트와 호환성을 달성하여야 한다.

TCP/IP 프로토콜에 관한 국제표준 RFC 1122 또는 RFC 1123에 따르면 일부 기능의 생략으로도 호스트간의 호환성이 이루어진다. 임베디드 시스템 같은 메모리 제약 상황에서는 표준규격의 일부기능을 생략하여야 하는 경우가 존재한다. 본 임베디드 제어기에서는 몇 가지를 생략하였다. 예로서 긴급하게 데이터를 처리하는 기능, 슬라이딩 윈도우를 통한 흐름제어, 및 IP 패킷의 조각 및 조립기능이 없다[14].

개발된 IP 계층에서는 수신패킷의 IP 헤더정보에 대한 판단과 ARP, ICMP, UDP, TCP 프로토콜 사이에 역 다중화(De-multiplex)를 한다. 수신 IP 패킷에 대하여 목적지와 자신의 IP 주소에 대한 부합여부를 검사하고, IP 헤더에 대하여 체크섬을 한다. 그러나 IP 패킷 조각내기 및 조립(Reassembly)은 구현하지 않았다. 이것은 임베디드 제어기와 같은 경우에는 패킷 조각내기가 필요하지 않기 때문이다.

ARP 프로토콜은 32비트 IP 주소와 48비트 MAC 주소사이에 변환작업을 한다. 클라이언트로 부터의 ARP 요청에 대하여 응답(Reply)을 하는 Echo-Reply 기능을 구현하였다. 그리고 ICMP 체크섬도 구현하였다[15]. 재전송(Redirect), 시간표시(Timestamp) 같은 에러처리 등의 다른 기능들은 생략하였다.

전달계층은 네트워크상의 일대일 연결설정 및 양방향 데이터 전달기능을 수행한다. 두 지점간의 양방향 데이터 전송을 위하여, 두 지점은 지속적으로 데이터를 추적하여 데이터의 생략, 중복 등을 검출한다.

TCP는 유한상태기계(Finite State Machine)로서 어떤 이벤트가 발생할 때까지 그 상태를 유지하도록 동작한다. 개발 시스템에서는 상태정보를 RAM에

저장한다. 수신된 패킷은 체크섬을 확인 한 후에 출발지/목적지의 포트/IP주소를 검사한다.

흐름제어에 보통 슬라이딩 윈도우(Sliding Window) 방법이 사용되는데, 본 연구에서는 간단한 윈도우 방법을 사용하였다. 논문에서는 1개의 TCP 조각에 대하여만 응답한다(acknowledge). 그래도 호스트간의 호환성은 유지된다. 데이터 전송 중에 에러가 발생하면 재전송하여야 하는데 본 연구에서는 고려하지 않았다. 네트워크 성능향상을 위하여 매번 연결설정마다 데이터의 왕복여행시간(RTT)를 계산하기도 하는데 본 논문에서는 이 기능을 생략하였다. 혼잡제어는 동시 다수 접속에 대한 제어를 위하여 필요한데 연결마다 1개의 TCP 조각을 가정하였기 때문에 생략되었다.

3.4 텔넷 및 X/Z 모뎀 인터페이스

FTU에 대한 데이터 전송은 RS-232 포트를 통하여 파일전송 기반으로 이루어진다. 파일전송은 텍스트 기반이 아닌 이진파일 형태로 이루어져 있다. 따라서 임베디드 무선랜 모듈에서 직렬통신포트에 이진파일 전송을 지원하는 X/Z 모뎀[16]이 되어야 한다. 이진파일 전달 프로토콜로서 시조격인 X 모뎀이 있고 뒤를 이어 Z, Y 모뎀 등이 개발되었다. 본 논문에서는 텔넷 기반으로 두 지점이 연결된 위에서 X/Z 모뎀이 이루어지도록 되어 있다.

MS사의 Windows 운영체제에서 사용되는 터미널 에뮬레이터 프로그램인 하이퍼터미널에서는 X모뎀 프로토콜이 그림 4와 같은 동작을 수행한다. X 모뎀에서 이진파일 전달 프로토콜에 대하여 설명한다. 한 번에 1바이트 혹은 1라인(line) 단위로 전달하지 않고 블록(block)의 문자 묶음으로 전송된다. 블록의 단위는 128바이트 혹은 1024바이트가 대표적이다. 서버는 송신할 파일로부터 블록 단위 만 크롬 읽은 후 송신하고, 클라이언트는 이를 수신하여 파일에 기입함으로써 파일 전달 기능이 수행된다. 송신 블록은 수신 클라이언트에 의하여 ACK(Acknowledge)된다. 신뢰성 있는 전달 기능을 제공하기 위해 전송과 관련된 제어 정보를 상호 교환 한다. 제어 정보에는 블록의 정상적인 수신, 오류 발생, 송신의 종

료 등을 알리는 정보가 포함된다. 이 밖에도 Z 모뎀 파일전송 프로토콜에서는 P2P(Point-to-Point) 직렬 파일전송 시에 세션 초기화 다이얼로그에서 중단된 파일전송이 재기되도록 하는 속성을 갖는다.

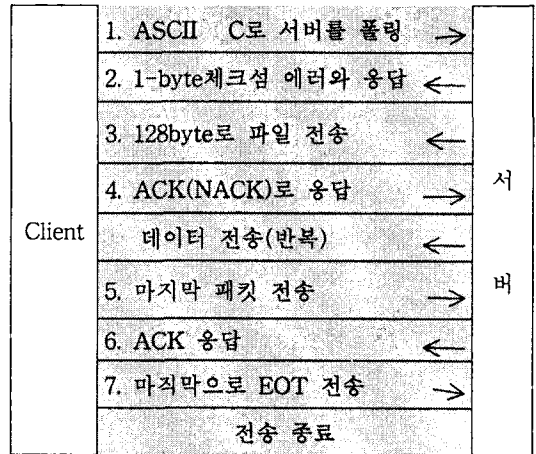


그림 4. X 모뎀 프로토콜 동작 흐름
Fig. 4. X modem protocol operation flow

텔넷은 다른 컴퓨터에 로그인할 수 있게 하는 클라이언트/서버 기반의 프로토콜로서 TCP/IP 기반에서 동작한다. 서버와 클라이언트 사이에서 23번 포트에 있는 소켓을 통하여 데이터를 송수신할 수 있도록 한다. 그리고 서버와 클라이언트는 다양한 연결설정 옵션에 의하여 제어된다. 연결 옵션 상태를 설정하기 위해서는 "IAC, <요구의 형태>, <옵션>"처럼, ASCII 코드 255번인 IAC(Interpret as command) 문자와 요구의 형태를 나타내는 명령어 코드 다음에 보내며, 응답 또한 같은 방식이다.

텔넷의 전체 프로토콜에는 다양한 화면제어 코드 및 가상터미널 제어코드들이 있다. 본 연구에서 구현하는 임베디드 시스템에서는 CRT 같은 디바이스가 없으므로 화면제어에 대한 필요성이 없다. 가상터미널 제어코드만 필요하며, 제어에 대한 요구의 형태명령어로서 4가지가 있다. 텔넷에서 사용되는 가장 기본적인 명령어 코드들인 WILL이나 WONT는 옵션을 요구하는 쪽이 그 옵션을 사용 또는 사용하지 않겠다는 뜻이며, DO나 DONT는 상대방에 옵션을 사용 또는 사용하지 말라고 요구하는 것이다.

배전설비를 위한 임베디드 무선랜 기술 개발

한 쪽에서 요구를 하면 상대방은 응답을 해야 한다. 이러한 요구와 응답의 조합으로 텔넷 프로토콜이 동작한다.

본 논문에서는 텔넷의 대부분 기능을 제거하고 최소한으로 데이터 전송만이 가능하도록 기본 기능만을 구현하였다. 최소한의 연결기능과 클라이언트/서버로서의 동작, 그리고 이진파일에 IAC (Hex 0xFF)와 동일한 제어코드 값이 포함되었을 때 이를 변환 처리하는 알고리즘을 사용하였다.

임베디드 모듈과 PC가 텔넷(Telnet) 프로토콜을 통하여 연결되고, 이후에 이진파일 전송이 이루어진다. 즉 임시 또는 인프라 네트워크의 무선랜 접속을 통하여 연결이 이루어진다[17]. 이후 TCP/IP 처리된 후 텔넷을 통하여 연결이 이루어진다. 그 이후 직렬통신의 X/Z 모뎀을 통하여 이진파일이 전달된다. 그리고 직렬통신 포트로부터 수신된 데이터는 TCP/IP 처리된 후 텔넷을 통하여 인터넷에 전달된다.

4. 실험 결과

개발 결과 무선랜 드라이버, ARP, ICMP, IP, UDP, TCP, Telnet, 및 X/Z 모뎀 인터페이스를 포함한 전체 소스는 8,207 라인으로 구성된다. 크기를 최적화하여 컴파일 했을 때 데이터 스택은 384B를 사용하며, EEPROM 66B로서 1.6[%], 프로그램은 15KB로 12.2[%], 및 RAM은 3,391B를 사용하였다. 무선 랜과 직렬통신 포트에 대한 설정정보를 저장하기 위하여 EEPROM이 사용되는데 SSID, WEP, 직렬통신 속도, 데이터 형식 등과 같은 데이터가 저장된다.

보통 TCP/IP 프로토콜에서 데이터 처리는 CRC (Cyclical Redundancy Check) 계산, 버퍼간의 데이터 복사와 이동, 그리고 문맥전환에서 주로 영향을 받는다[18]. TCP의 QoS 향상을 위하여 네트워크 성능 분석이 활용된다[19]. TCP/IP의 성능은 주로 재전송 시간 초과(Retransmission Time Out: RTO)에 의하여 주로 영향을 받는다. 그래서 네트워크 성능 향상을 위하여 패킷의 왕복여행시간(Round Trip Time: RTT)을 감소하도록 하여야 한다. 개발된 프로토콜에 대한 RTT 테스트를 위하여 핑(ping)을 사

용하였다. TCP는 네트워크 트래픽을 관찰하여 RTT를 조정하여 성능을 향상하도록 노력하고 있기 때문에 RTT 값은 프로토콜 성능예측에 중요하게 사용되는 값이다.

테스트를 위하여 그림 5와 같은 시스템이 사용되었다. PC A는 Intel Celeron(TM) 1.2[GHz] CPU에 윈도우 2000이 동작된다. 노트북 PC B는 Intel Pentium 4, 1.8[GHz] CPU 사양에 윈도우 XP가 동작한다. AP는 ReeNet의 802.11[b/g] 무선라우터 Router 모델 RW-IPG500R을 사용하였다. 무선랜 모듈의 무선랜 카드는 Airgate 2000C로서 11[Mbps] WLAN CF 형식을 사용하였다.

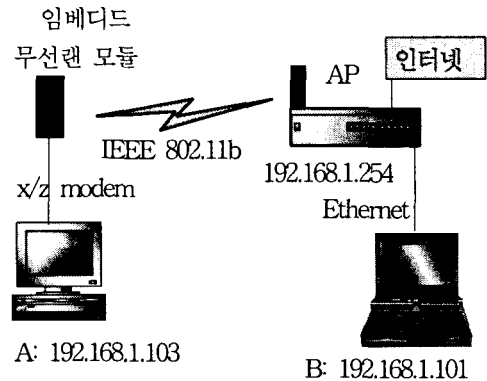


그림 5. 테스트 시스템 구성
Fig. 5. Test system setup

처리율에 대한 실험을 위하여 네트워크 트래픽 분석기 DU[®] meter[20]를 사용하였다. 노드 A와 B 간에 테스트베드용 637[KB] 크기 파일을 하이퍼터미널 윈도우소켓(winsoc) 연결을 통하여 전송하면서 DU[®]로 측정을 하였다. 최대와 평균 처리율은 1.6[KB/sec]과 1.0[KB/sec]로 되었다. 그림 6은 DU[®]를 사용한 측정중의 화면을 저장한 것이다.

A와 B간의 핑(ping) 테스트를 위하여 32[B]의 패킷을 28회 전송하여 왕복지연시간(RTT)으로 최소 11[ms], 최대 12[ms], 평균 11[ms]을 얻었다. 다음 무선랜 모듈과 B를 Ad-hoc 모드로 연결하고 핑(ping) 테스트를 실시하여 RTT로 최소 11[ms], 최대 12[m], 평균 11[ms]처럼 이전 테스트와 동일한 결과를 얻었다. 다음 B와 AP간의 유선을 통한 RTT

테스트 결과는 1[ms] 이하로 매우 작은 값을 얻었다. 핑(ping) 테스트 결과 상대적으로 느리지만 TCP, ICMP 프로토콜이 정상적으로 작동함을 알 수 있다. 이는 마이크로컨트롤러가 패킷 데이터 처리를 위하여 RAM을 접근하기 때문이다.

핑(Ping)은 TCP/IP가 올바르게 설정되었다는 것을 판단하는데 유용하게 사용되는데, 이것은 IP가 자신에게 패킷을 라우팅할 수 있다는 것을 의미한다.

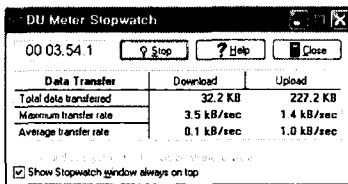


그림 6. DU[®]를 통한 처리율 측정
Fig. 6. Throughput measurement via DU

그림 7은 임베디드 무선랜 장비의 동작 상태를 모니터링 한 것으로 SSID, IP, 사용채널, WEP, 및 직렬통신 프로토콜 설정 등에 대한 데이터가 나타나 있다.

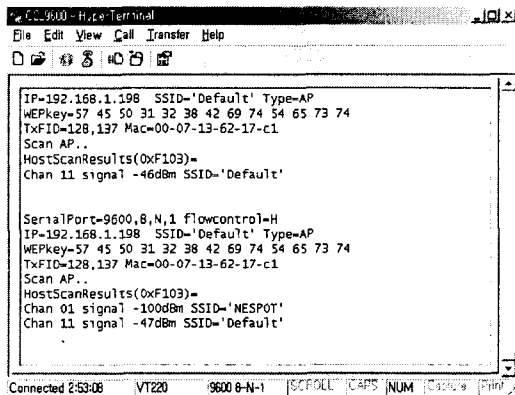


그림 7. 임베디드 무선랜 모듈 동작의 감시
Fig. 7. Monitoring for embedded WLAN module

프로토콜의 성능 분석에 이어 동작의 올바름(Correctness)에 대한 판단을 위하여 CommViewFi[®] 소프트웨어를 사용하였다[21].

끝으로 그림 8은 개발된 임베디드 무선랜 모듈을 보여주고 있다.

5. 결 론

본 연구에서는 주상에 설치되어 운전되는 기존의 FTU에 적합한 임베디드 무선랜 제어기를 개발하였다.

가격 면에서 경쟁력 있도록 하기 위하여 Atmega 128과 같이 RAM 용량이 4[KB]로 제한된 8비트 마이크로컨트롤러에서 TCP/IP 프로토콜 스택을 구현하는 방안과 기존 배전설비에 갖고 있는 X/Z 모뎀과의 호환성을 유지하기 위한 Telnet 기반에서 동작하는 간략화된 임베디드 X/Z 모뎀을 구현하는 기술을 개발하였다. 이러한 제한된 시스템에서 흐름제어, 오류제어를 위한 데이터 구조를 제안하였고, TCP/IP의 전체 스택과 기능을 개발하지 않고 호환성을 유지하는 방안을 찾아 이를 구현하였다.

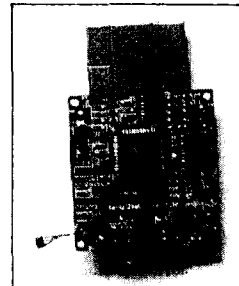


그림 8. 개발 제품
Fig. 8. Final product

그리고 DU와 CommViewFi라는 네트워크 성능 및 프로토콜 분석기를 통하여, 구현된 임베디드 TCP/IP 스택과 텔넷, X/Z 모뎀 인터페이스 프로토콜이 성능은 다소 떨어지지만 프로토콜의 올바름에는 아무 문제가 없는 것으로 분석하였다.

끝으로 개발된 TCP/IP 프로토콜 및 모뎀 기술이 하이퍼터미널의 기능으로 실험 및 검증중을 하였으나 실제 FTU 설비에 설치되어 검증되는 추가적인 연구를 하는 것이 필요하다고 판단한다.

References

- [1] P&C Technologies <http://www.pnctech.co.kr/english/home.htm>
- [2] J. Woo, S. Sugoog, "Embedded System Programming", HongNeung Pub., 2006.
- [3] Conexant Systems Inc, <http://www.conexant.com/>.
- [4] R. Braden, Requirements for internet hosts - communication layers. RFC 1122, Internet Engineering Task Force, October 1989.
- [5] Adam Dunkels, "Full TCP/IP for 8-bit Architecture", In Proceedings of the First International Conference on Mobile Systems, Applications and Services (MobiSys), San Francisco, May 2003.
- [6] A. Dunkels. uIP - a TCP/IP stack for 8- and 16-bit microcontrollers. Available from dunkels.com/adam/uip/, 2004.
- [7] H. Shrikumar. "iPic - a match head sized web server", Available from www-ccs.cs.umass.edu/~shri/iPic.html, 2004.
- [8] J. Benthams, TCP/IP Lean: Web servers for embedded systems. CMP Books, October 2002.
- [9] Atmel Corporation, Embedded web server. AVR 460, Available from www.atmel.com, January 2001.
- [10] S. Shon, "KOSEF Research Final Report", KOSEF, 2003.
- [11] Sugoog SHon, "Development of gas boiler control technology by using Web browser", KEE Journal, 2004.
- [12] Behrouz A. Forouzan, "TCP/IP Protocol Suite", McGraw-Hill International, page 297-299, 2000.
- [13] Intersil, "Prism Driver Programmer Manual", 2003.
- [14] David D. Clark, John Romkey, Van Jacobson, Howard Salwen, "An Analysis of TCP Processing Overhead", IEEE Communications Magazine, Pages 23 - 29, June 1989.
- [15] A. Rijssinghani, Computation of the internet checksum via incremental update. RFC 1624, Internet Engineering Task Force, May 1994.
- [16] MicroSoft "HyperTerminal Zmodem File Transfer", http://msdn.microsoft.com/library/default.asp?url=/library/en-us/randz/protocol/hyperterminal__zmodem__file__transfer.asp.
- [17] Jim Geier, "Wireless LANs", 2nd edition, SAMS, 2002.
- [18] J. Kay and J. Pasquale, The importance of non-data touching processing overheads in TCP/IP, In Proceedings of the ACM SIGCOMM '93 Symposium, pages 259-268, September 1993.
- [19] Craig Partridge, Jim Hughes, Jonathan Stone, "Performance of Checksums and CRCs over Real Data", In Proceedings of ACM SIGCOMM '95 (Jan '95). Pages 68 76, 1995.
- [20] Hagel Technologies Ltd, <http://www.dumeter.com>.
- [21] TamoSoft inc. CommView[®], <http://www.tamos.com/products /commview/>, 2004.

◆ 저자소개 ◆

손수국 (孫守國)

1959년 7월 15일생. 1982년 서울대학교 전기공학과 졸업. 1984년 서울대학원 전기공학과 졸업(석사). 1996년 미국 University of Texas at Austin 공학박사. 현재 수원대학교 정보통신공학과 조교수.

우종정 (禹宗楨)

1957년 9월 4일생. 1982년 경북대학교 전자공학과 졸업. 1990년 미국 University of Texas at Austin 공학석사. 1993년 미국 University of Texas at Austin 공학박사. 현재 성신여자대학교 컴퓨터정보학부 교수.