한국철도학회논문집
제12권 제1호 2009년 | pp. 45-54

전력 IT용 D-TRS 접속을 위한 게이트웨이 플랫폼

Gateway Platform for Connecting D-TRS in Electric Power IT

송병권† · 정태의*
Byung-Kwen Song · Tae-Eui Jeong

Abstract TETRA(Terrestrial Trunked Radio) is the standard of D-TRS(Digital Trunked Radio System) developed by the ETSI(European Telecommunications Standards Institute). Currently, TETRA has been selected as Electric Power IT Wireless Backbone Network. Traditional distribution automation system has used the wireless network of Mobile Service Provider using CDMA(Code-Division Multiple Access) and WCDMA(Wideband CDMA) modem. Therefore, the development of gateway platform is indispensable according to the change to TETRA network. In this paper, we propose an embedded Linux-based Gateway platform for transmitting Industrial Electric Power IT protocol using TETRA network.

Keywords : TETRA, TETRA Gateway, D-TRS

요 지 TETRA는 유럽 전자 통신 표준기구인 ETSI에서 개발한 D-TRS(Digital Trunked Radio System) 표준이다. 현재 전력 IT 무선기간망으로 TETRA가 선정되었다. 기존 배전 자동화 시스템은 CDMA(Code-Division Multiple Access), WCDMA(Wideband CDMA) 모뎀을 이용한 이동통신사 무선 네트워크를 이용하고 있다. 따라서 TETRA 네트워크로 변경에 따른 게이트웨이가 필요하다. 본 논문은 TETRA 네트워크를 이용한 산업용 전력 IT 프로토콜을 전송하기 위한 임베디드 리눅스 기반 게이트웨이 플랫폼을 제안한다.

주 요 어 : 테트라, 테트라 게이트웨이, 디지털 TRS

1. 서 론

전력 IT 분야에서 사용되고 있는 다양한 프로토콜에는 전기, 가스의 원격검침에 사용되고 있는 DLMS (Device Language Message Specification) 프로토콜, DNP 3.0 (Distributed Network Protocol 3.0), IEC61850, Modbus 등이 있다. 현재, 해당 프로토콜 데이터들의 전송은 CDMA (Code Division Multiple Access) 모뎀이나 WCDMA 모뎀을 이용한 이동통신사 무선 네트워크를 이용하고 있다. 이러한 이동통신사의 무선 네트워크 자원의 이용에 매년 막대한 자원 사용비용을 지불하고 있다.

따라서 현재 전력 IT 무선 기간망으로 선정된 TETRA (Terrestrial Trunked Radio) 네트워크를 이용한 데이터 전송방식으로 전환한다면 해당 사용료에 대한 비용 절감효과를 볼 수 있다[1].

TETRA는 ETSI(European Telecommunications Standards Institute)에서 개발한 유럽 이동 무선 통신 기반의 표준으로 경찰, 소방 무선과 같은 공공 기관, 공공 단체의 이동 무선 통신의 필요에 따라 음성 및 데이터 통신 서비스를 제공할 목적으로 개발된 것으로 디지털 트렁크 무선 기술을 기반으로 한 이동 무선기술로 평상시 고유의 업무를 수행하다가 재난재해 발생시, 통합지휘 무선통신망의 기능을 수행하는 시스템을 말하는 것으로써 다음과 같은 여러 Air Interface를 지원한다[2,11]. 음성과 데이터 전송이 가능한 V+D(Voice Plus Data), 패킷 데이터 전송이 가능한 PDO(Packet Data Optimized), 그리고 두 단말 간에 직접통신을 할 수 있는 DMO(Direct Mode Operation)가 그것이다[3]. 이와 같은 TETRA 네트워크 서비스를 이용한 다양한 산업용 프로토콜 데이터 전송을 위한 게이트웨이 플랫폼을 제안하고자 한다.

† 책임저자 : 정회원, 서경대학교 정보통신공학과 교수
E-mail : bksong@skuniv.ac.kr
TEL : (02)940-7293
* 교신저자 : 서경대학교 컴퓨터과학과 교수

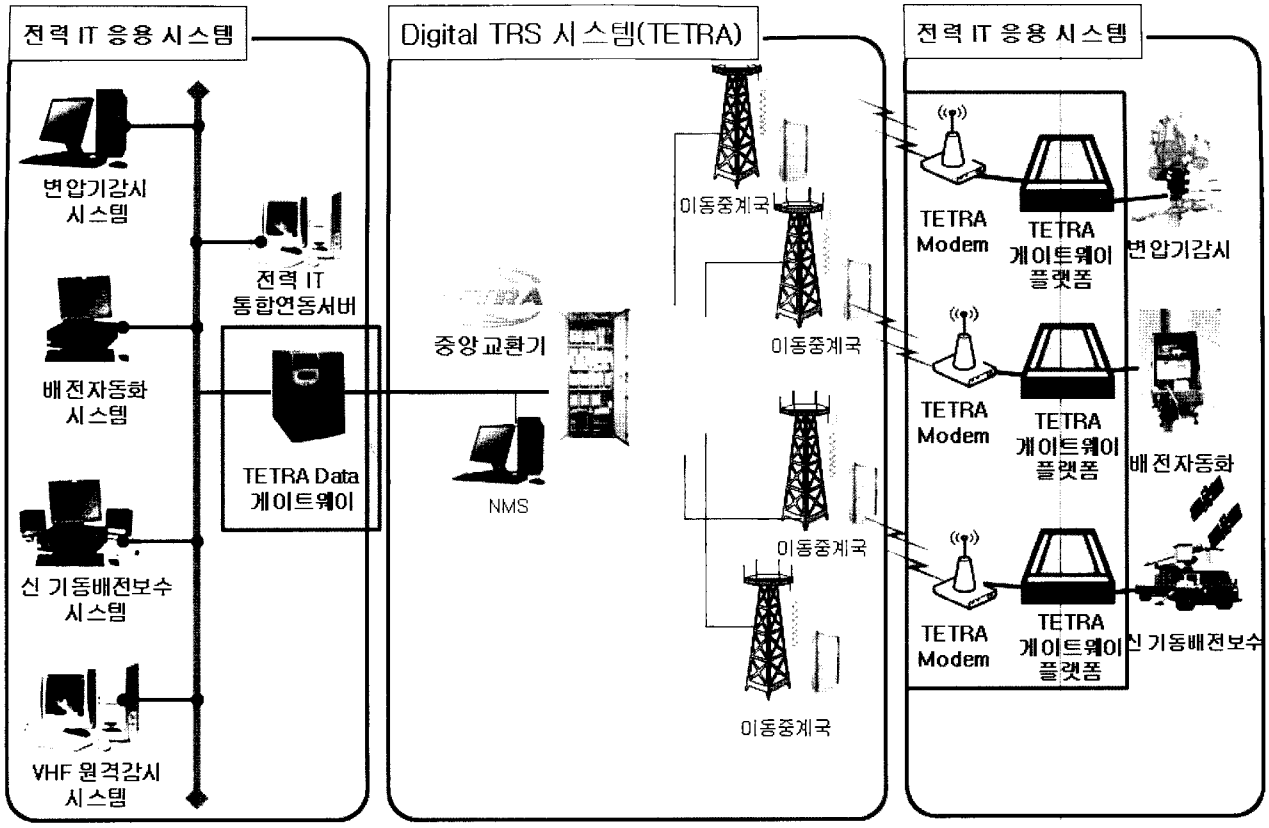

Fig. 1. Electric Power IT backbone network based on TETRA

2. 관련연구

독일 piciorgros사의 TRM-710.200T/Z - TETRA Master Station 모뎀은 TETRA 네트워크 상으로 Modbus, IEC60870-5-101 프로토콜 데이터를 보낼 수 있도록 해주는 제품이다. RS-232 또는 RS-422/RS-485로 전송받은 프로토콜 데이터를 TETRA 프레임으로 변환하여 TETRA Slave Radio 모뎀으로 전송한다. TETRA Slave Radio 모뎀은 전송받은 TETRA frame을 다시 변환하여 시리얼 포트로 다시 전송한다. 해당 제품들은 실제 독일에서 판매되고 있는 제품들로 TETRA 데이터 전송방식 중 SDS(Short Data Service)를 이용하여 전송하도록 구현되어있다. 또한 송수신 상태 및 현재 기기의 상태와 TETRA 모뎀의 전파 세기를 알 수 있는 간단한 형태의 LED 표시가 되어 있다[4].

핀란드의 FF-Automation 사의 AutoLog® TETRA-RTU 플랫폼은 TETRA 모뎀을 내장하여 TETRA SDS 서비스와 패킷 서비스를 이용한 데이터 전송을 지원하고 있다. PLC(Programmable logic controller)를 지원하여 목적지 프로그램에 독립적이고, 자동화된 구성이 가능하다. Modbus 프로토콜을 전송가능하고, 32x build-in PID 컨트롤러를 지원하며, RS-232C 포트를 이용한 HMI(Human Machine Interface)를 지원한다[10].

3. TETRA 기반 전력 IT 기간망 구조

Fig. 1은 한전 KDN(주)에서 구축한 전력 IT 기간망의 구조이다[1]. 디지털 TRS 시스템을 TETRA로 구축하여 변압기 감시, 배전 자동화, 신 기동배전보수 등의 전력 IT 응용시스템을 연동하여 사용하고 있다. 본 논문에서 구현할 플랫폼은 전력 IT 응용시스템 부분의 앞단에 위치하여 전력 IT 통신 데이터를 분석하여 TETRA Data 게이트웨이를 통해 목적지로 전달한다.

4. TETRA PEI(Peripheral Equipment Interface)

TETRA 네트워크를 이용한 데이터 전송은 TETRA 모뎀과 데이터를 전송하려는 장치와의 인터페이스가 필요하다. 그 관계를 ETSI 표준에서는 TETRA PEI로 정의해 놓았는데, Fig. 2는 TETRA PEI(Peripheral Equipment Interface)가 지원해주는 Layer를 나타낸 것이다. TETRA PEI는 패킷 데

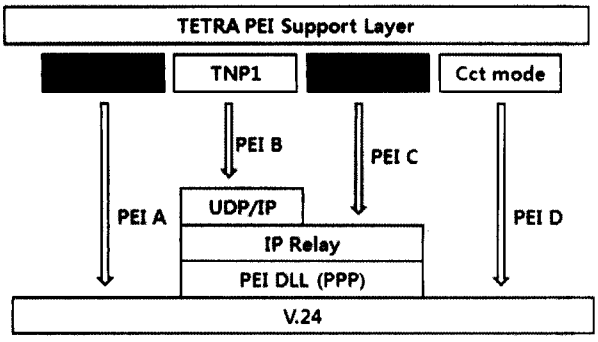

Fig. 2. TETRA PEI Support Layer

이터, 서킷 데이터, 그리고 단문데이터 3가지 형태로 사용할 수 있다. 각 전송 방식은 Fig. 2에서 나타난 것과 같이 AT command를 이용한 PEI A 방식인 단문 데이터 전송과 PEI C 방식인 IP(Internet Protocol)를 이용한 패킷 전송방식 그리고 TNP1(TETRA Network Protocol type 1)을 이용한 PEI B 방식 3가지가 있다[5].

AT command를 이용한 SDS 메시지 전송 방식은, SDS type 1, SDS type 2, SDS type 3, SDS type4, SDS-TL (SDS-Transport Header) type 4로 각기 구분은 최대 전송 가능한 데이터크기로 구분할 수 있다. 최대 2바이트의 데이터 전송이 가능한 SDS type 1, 4바이트의 데이터 전송이 가능한 SDS type 2, 8바이트의 데이터 전송이 가능한 SDS type 3, 최대 140바이트의 데이터 전송이 가능한 SDS type 4, 그리고 SDS-TL(Transport Layer)은 Transport Layer 컨트롤 기능을 삽입한 방식까지 5가지 방법이 있다. SDS-TL은 TETRA 네트워크를 관리 하는데 사용한다[5].

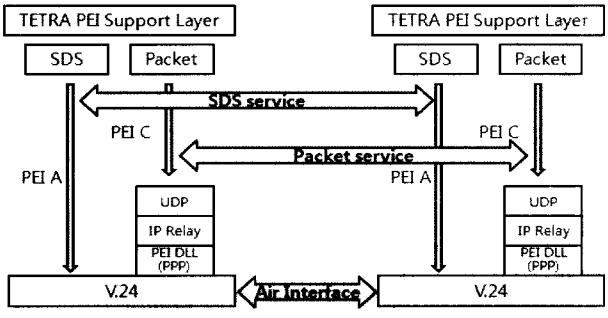

Fig 3. TETRA Network Service

본 논문에서 구현한 플랫폼은 PEI A 방식인 SDS 서비스와 PEI C 방식인 패킷 서비스 전송 방식을 이용한 데이터 전송을 Fig. 3과 같이 구현하였다. SDS 서비스를 이용한 데이터 전송 이후 송신한 SDS 형태의 데이터를 분석하여 상위응용계층으로 전송한다. SDS 서비스를 이용하기 위해서는 다음 절에서 설명한 AT command를 이용하여 SDS 전송 명령어인 "AT+CMGS"를 통해 전송한다. 패킷 서비스를 이용하기 위해서 "ATD*99#" command를 통해 PPP연결을 시도하여 IP를 획득 후 패킷 통신한다[5].

4.1 AT command

AT command는 모뎀 제어를 위해 사용하는 명령어로써 TETRA 모뎀을 제어하는데 사용하는 AT command는 ETSI 표준문서, ETSI EN 300 392-5에서 정의하고 있다. TETRA 모뎀은 AT command를 이용하여 3가지의 모드에 들어갈 수 있다. Fig. 4에서 보면 초기 AT command 모드에서 ATA(AT-Answer), CTCC(Command TETRA Call Connect), ATO(AT-OnLine) 등의 command로 AT circuit모드로 진입하게 되면 해당 상태에서는 AT command가 무시되고 TE(Terminal Equipment)에서 ESC(Escape) 신호를 보내거나 MT(Mobile Termination)와의 연결이 끊어질 경우 다시 AT command 모드로 돌아오게 된다[5]. AT command 모드에서 "ATD*99#" 명령을 주게 되면 TETRA 모뎀이 PPP(Point to Point Protocol) 연결을 시도하여 IP를 이용한 패킷 모드로 진입하게 된다. 패킷 모드 일 경우 AT command는 무시되고, 물리적 링크가 실패하거나 PPP 링크가 끝났을 경우, 다시 AT command 모드로 돌아가게 된다[5].

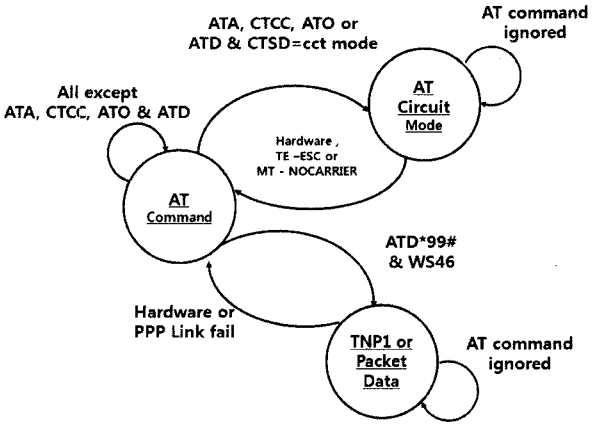

Fig. 4. AT command State Diagram

4.2 TETRA Inner 프로토콜

TETRA 네트워크 안에서만 사용될 프로토콜을 TETRA Inner 프로토콜이라고 정의하였다. TETRA Inner 프로토콜은 원본 프로토콜 데이터를 TETRA 네트워크로 전송 할 수 있도록 해주고, TETRA 모뎀 및 FRTU(Feeder Remote Terminal Unit)의 인터페이스 설정을 할 수 있게 해준다. Fig. 5에서 TETRA G/W와 TETRA 게이트웨이 플랫폼 사

이 구간이 Inner 프로토콜의 적용 범위이다. Inner 프로토콜의 세부 형식은 Table 1과 같다. 첫 번째 1바이트는 STX (Start of text)로써 아스키 코드표에서 사용하는 STX의 의미로 사용되지 않고 단지 Inner 프로토콜에서만 적용되는 의미로 항상 0×7E의 값을 가진다.

Table 1. TETRA Inner 프로토콜 프레임 Format

STX	LEN	Forward ID	TH	CMD	SQ	DATA Field
1byte	2byte	4byte	1byte	1byte	1byte	0~1490
0×7E	-	-	-	-	-	User Data

두 번째 LEN 필드는 2바이트의 값을 가지고, Forward ID로부터 Data field까지의 총 데이터 길이를 바이트 단위로 나타낸다.

Forward ID는 TETRA 네트워크 운영 간 필요한 필드로 총 4바이트가 사용되고, 평상시에는 목적지 TETRA 모뎀 주소가 들어가게 되지만, 망 운영상 타 권역에 존재하는 목적지로 전송할 때에는 중계지의 주소가 들어가게 된다.

TH필드는 Transport Header의 의미를 가지고 있고 분할 전송을 위한 필드이다.

CMD 필드는 Confirm, Group, Function 필드로 구성되어 있다. 각 필드의 조합으로 본 데이터의 목적을 알 수 있게 해주는 필드이다. 예를 들면, 0×42값을 가질 경우 TETRA G/W의 test의 목적을 가지는 내용이 된다.

SQ 필드는 CMD 필드가 0×12값일 경우 모뎀의 Signal Quality 값이 설정되어 응답\한다.

5. TETRA API 구현

TETRA 네트워크를 이용하여 전송 할 때 사용될 API를 C언어를 기반으로 구현하였다. 각 API는 크게 상태 확인용 함수, 데이터 분석 함수, 상태 제어용 함수, 데이터 생성 함수, 데이터 전송 함수로 구현 하였다. 상태 확인 함수는 본 플랫폼의 상태를 확인하거나, 전송 데이터 형식을 확인하는데 사용한다. 데이터 분석 함수는 전송받은 데이터의 Inner 프로토콜을 분석하여 적용하는데 사용한다. 상태 제어용 함수는 전송받은 데이터의 Inner 프로토콜상 제어 목적으로 사용될때 사용한다. 데이터 생성함수는 송신해야할 데이터를 Inner 프로토콜로 전송을 위해 사용하는 함수이다. 마지막으로 데이터 전송 함수는 데이터 전송 형식에 따라 데이터를 TETRA모뎀을 통해 전송하는 함수로 구현 하였다. 주요 함수의 원형은 Table 2와 같다.

Table 2. TETRA API

● **unsigned char analysis_inner_protocol(char *)**	
설명	Inner 프로토콜 분석 함수
사용법	include<TETRA_API.h> unsigned char analysis_inner_protocol(char *);
반환값	0x0 Inner프로토콜 포맷이 아님 0x11 설정된 값 read 0x12 TETRA 모뎀의 Signal Quality 값 read 0x13 GPS 데이터 read 0x21 설정값 write 0x31 TETRA 게이트웨이 플랫폼 Reset 0x32 모뎀 Reset 0x41 TETRA 게이트웨이 플랫폼 Test 0x42 TETRA G/W test 0x43 실제 프로토콜 데이터
● **int create_modem_read_thread(void)**	
설명	모뎀 read 스레드 생성
사용법	include<TETRA_API.h> int create_modem_read_thread(void);
반환값	1 = 생성 성공
Error	0 = 생성 실패
● **int create_frtu_read_thread(void)**	
설명	frtu read 스레드 생성
사용법	include<TETRA_API.h> int create_frtu_read_thread(void);
반환값	1 = 생성 성공
Error	0 = 생성 실패
● **void create_inner_protocol(char *, int cmd, int size_of_data)**	
설명	Inner 프로토콜 header 생성
사용법	include<TETRA_API.h> void create_inner_protocol(char *, int cmd, int size_of_data);
매개변수	char * : 생성될 Inner 프로토콜과 함께 전송될 데이터 cmd : 전송될 데이터의 목적 0x11 설정된 값 read 0x12 TETRA 모뎀의 Signal Quality 값 read 0x21 설정값 write 0x31 TETRA 게이트웨이 플랫폼의 Reset 0x32 모뎀 Reset 0x41 TETRA 게이트웨이 플랫폼 Test 0x42 실제 프로토콜 데이터 size_of_data : 전송될 데이터의 길이
● **int send_sds_message (char *, int sds_type)**	
설명	sds 메시지 전송 함수
사용법	include<TETRA_API.h> void create_tl_header(char *);
매개변수	char * : 전송될 메시지 sds_type : sds 전송규격 0 = sds type 1 1 = sds type 2 2 = sds type 3 3 = sds type 4 4 = sds type 4 - TL header
반환값	1 = 전송 성공
Error	0 = 전송 실패

각각의 API들은 매번 데이터 송수신 간에 호출되어 사용되어지며, 플랫폼 구동시 Table 2에 나타난 두 개의 read 스레드가 구동하며, 해당 세부 내용은 6.3절에서 나타내었다.

6. TETRA 게이트웨이 플랫폼 개발

앞 절에서 구현한 API를 이용하여 임베디드 보드 상에서 DNP 3.0 데이터를 TETRA 네트워크를 통해 전송하는 모델을 구현하였다.

DNP 3.0은 변전소 자동화 시스템에서 사용하는 산업계의 표준화 프로토콜이다. 기본적으로 Master/Slave의 구조를 가지고 있으며, 효율적인 네트워크 운영을 위해 OSI 7계층에서 변형한 3개의 Layer를 가지는 프로토콜 스택을 가지고 있다. 사용되는 분야로 전기, 석유, 가스 등 유틸리티 산업에서 광범위하게 사용되고 있는 프로토콜이다. 현재 한전KDN에서 전력자동화 시스템 개발에 적용중인 프로토콜로써, DNP 3.0 프로토콜을 TETRA 네트워크로 전송하는 테스트로 구현한 TETRA API 및 TETRA 게이트웨이 플랫폼을 테스트 하였다.

6.1 테스트 베드 구성

DNP 3.0 프로토콜을 TETRA 네트워크로 전송해보기 위해 다음과 같은 시스템과 네트워크를 구성하였다.

- FRTU 1대
- DNP 주장치 1대
- TETRA G/W system 1대
- TETRA 게이트웨이 플랫폼 모듈(Linux 2.6 kernel)
- TETRA 모뎀(EADS TMR880i, Teltronic MDT-400, Unimo 모뎀 MU-1000)

Master Server가 DNP 3.0 데이터를 생성하고 관리하는 DNP 주장치로 DNP Request 데이터를 TETRA Gateway로 전송하면 Inner 프로토콜로 인캡슐하여 TETRA 네트워크로 전송하고 이를 받은 TETRA 모뎀이 TETRA 게이트웨이 플랫폼으로 전송하면 원본 프로토콜 데이터를 추출하여 FRTU로 전송한다.

6.2 하드웨어 플랫폼

인텔사의 XScale PXA272코어, 128Mbyte의 SDRAM과 512Mbyte의 NAND flash, 10/100Mbps Ethernet을 지원하는 LAN91C111칩과 풀 펑션 UART를 지원하는 보드에 구현하였다. UART 및 Ethernet 포트를 이용하여 데이터를 송수신 할 수 있도록 구성하였다. 현재 TETRA 게이트웨이 플랫폼에 3개의 UART 포트를 이용하여 TETRA 모뎀과 FRTU, 그리고 모니터링을 위한 콘솔포트로 구성하였다. 앞으로 외부 I/O 포트를 이용하여 적외선 포트 통신을 위해 연구 중에 있다.

6.2.1 TETRA 모뎀

현재 구현한 TETRA 게이트웨이 플랫폼에서는 총 3대의 (EADS사의 TMR880i[7], Teltronic사의 MDT-400[8] 그리고 Unimo사의 MU-1000[9]) 모뎀을 가지고 테스트를 진행하였다. 각각의 모뎀마다 모뎀 초기화 명령어가 다르고, AT command 요청에 대한 응답 메시지의 형태가 달랐기 때문에 모뎀을 구별 할 수 있는 AT command인 "AT+GMI" 명령어를 통해서 각 모뎀의 응답 메시지를 분석하였다. EADS 모뎀의 경우 "AT+GMI" 요청에

"at+gmi\n\n\rEADS\n\r\n\rOK\n\r",

Teltronic 모뎀의 경우에는

"at+gmi\n\n\rTELTRONIC S.A.U.\n\r\n\rOK\n\r",

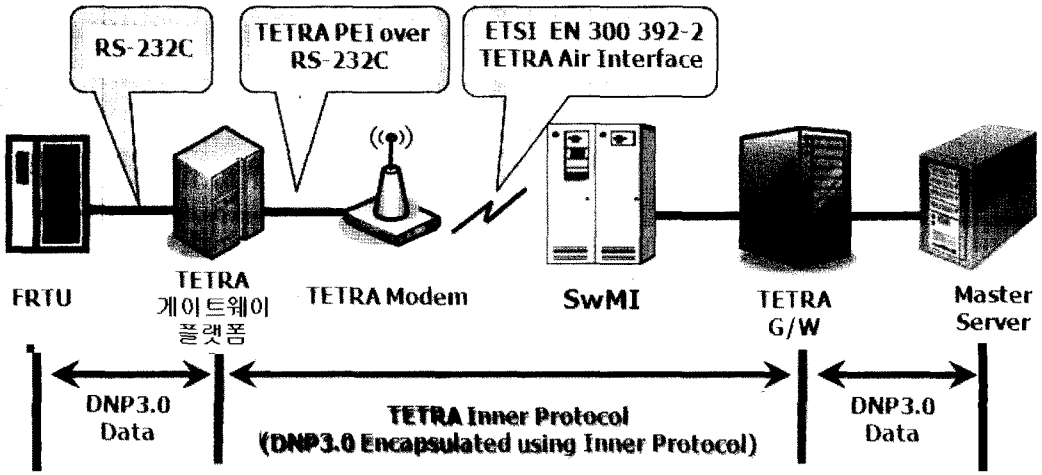

Fig. 5. Test Network structure

Unimo 모뎀의 경우에는
"at+gmi\n\n\r+GMI:\"UNIMO\"\n\r\n\rOK\n\r",

이런 응답 메시지를 보내왔다. 이와 같은 특징을 이용하여 TETRA 게이트웨이 플랫폼이 처음 기동될 때 연결된 모뎀 제조사를 확인하는 절차를 거쳐서 각 모뎀에 맞는 초기화를 할 수 있도록 구현하였다.

6.3 TETRA 게이트웨이 플랫폼 모듈 상태천이도

TETRA 게이트웨이 플랫폼에서 구현한 모듈을 상태천이도로 나타내면 다음 Fig. 6, Fig. 7과 같다. Fig. 6에서 1번 상태와 2번 상태는 각각 Fig. 7의 1번 상태와 2번 상태와 연결된다. TETRA 게이트웨이 플랫폼 모듈이 동작을 시작하면 Modem_Recv 스레드와 FRTU_Recv 스레드가 생성된다. Modem_Recv 스레드에서 TETRA 모뎀을 통해 전송받은 데이터는 Inner 프로토콜 Header 분석을 통해 Group 값이 4, Function 값이 3일 때 목적지로 전송한다. 그 외의 값일 경우 데이터 전송의 목적이 아닌 Control의 목적이므로 Fig. 7의 상태에 진입하여 SDS send thread나 패킷 thread 루틴을 수행하게 된다. FRTU_Recv 스레드에서 데이터를 받을 경우 전송 모드 확인 후 Fig. 7 상태로 진입하여 TETRA 모뎀을 통해 목적지로 전송한다.

6.4 메시지 송신 모듈 구조

송신 모듈을 5절에서 구현한 API와 C언어기반의 Pseudo code로 나타내었다.

```
//메인 송신 모듈
void FRTU_Recv(){
        create Buffer(data);
        if(read(data) == TRUE){
        create_inner_protocol( data, 0x43, sizeof( data ) );
        //Inner 프로토콜 생성 함수
                if(check_sds_tl( ))
                create_tl_header( char * );
                //SDS TL-Header 생성 함수
                if(s_type == SDS)
                send_sds_message( data, 4 );
                //SDS 메시지 전송함수
                if(s_type == PACKET)
                send_packet_message( data );
```

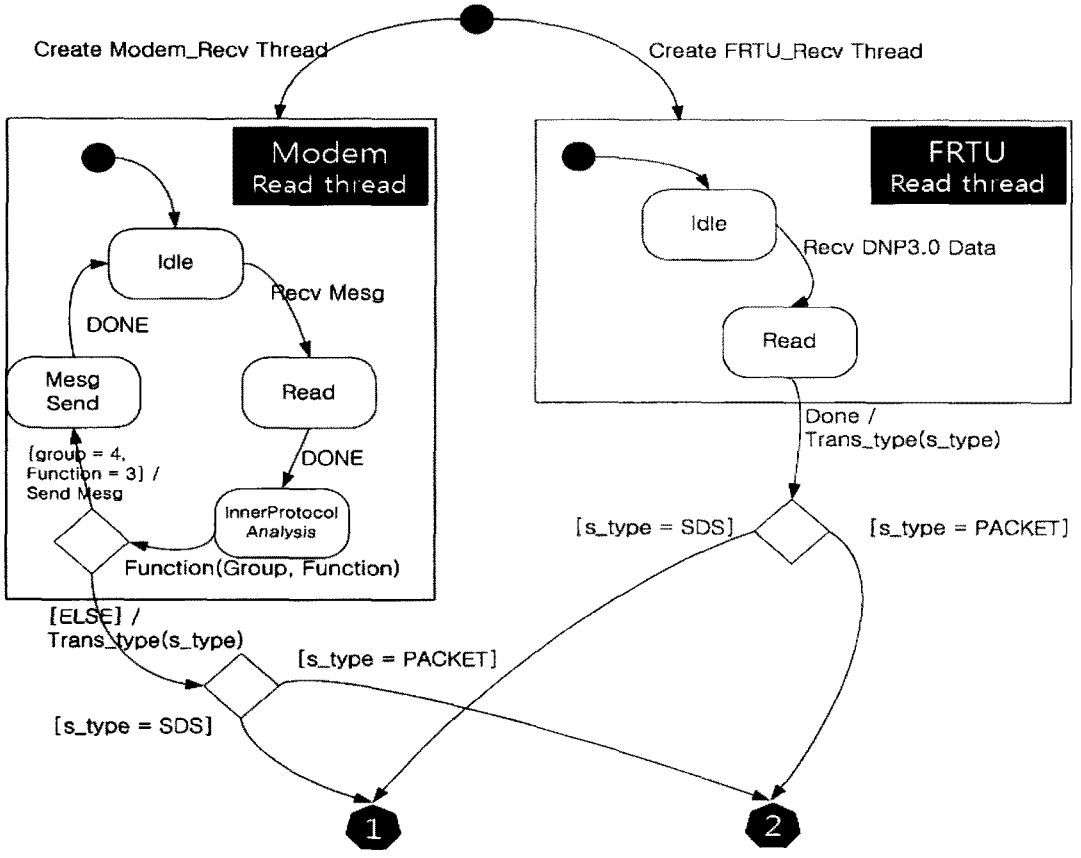

Fig. 6. TETRA Gateway status diagram(1)

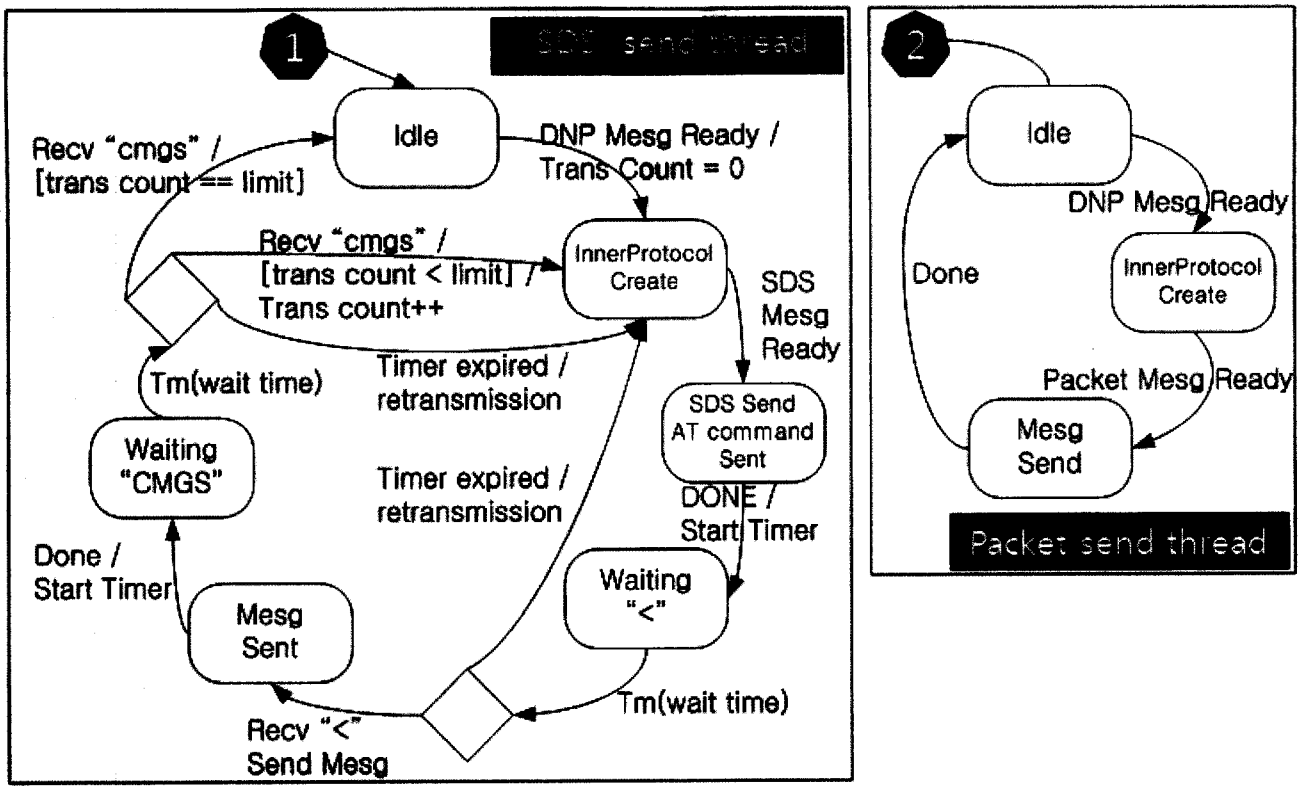

Fig. 7. TETRA Gateway status diagram(2)

```
            //패킷 메시지 전송함수
        }
        else wait for any data
}
```

FRTU_Recv는 스레드로 동작한다. 전송 받은 데이터를 송신 하기 위해 SDS-TL(Transport Layer)를 사용하는지 체크를 하여, 사용한다면 SDS-TL Header를 생성한다. 그다음 지금 사용하는 전송 모듈이 SDS방식을 사용하는지 Packet방식을 사용하는지 체크하여 각 방식에 맞는 API를 호출한다. 각각 호출한 모듈로 TETRA 모뎀을 통해 목적지로 전송한다.

6.5 메시지 수신 모듈 구조

다음은 모듈들을 C언어기반의 Pseudo code로 나타내었다.

```
//메인 모듈
void Modem_Recv(){
        create Buffer(data);
        if(read(data) == TRUE){
                analysis_modem_data( data );
                //모뎀에게 받은 데이터 분석 함수
                analysis_inner_protocol( data );
                //Inner 프로토콜 분석 함수
                if(group == 4 && function == 3)
                {       write( data );
                        //write to FRTU
                }
                else
                execute base on InnerProtocol
        }
        else wait for data
}
```

Modem_Recv는 스레드로 작동한다. TETRA 모뎀을 통해 전송받은 데이터는 AT command 형태로 되어 있다. 따라서 AT command와 실제 전송 받은 데이터를 분리한다. Inner 프로토콜 분석 루틴을 호출하여 Inner 프로토콜의 내용에 따라 상태확인, 제어 ,데이터 전송 등의 역할을 수행한다.

6.6 DNP3.0 over TETRA

TETRA 네트워크를 통해 전송되는 DNP3.0 프로토콜 데이터의 모니터링 모듈을 개발하여 TETRA 게이트웨이 플

랫폼 모듈에 적용하였다. 또한 Inner 프로토콜과 SDS-TL 헤더 분석 모듈도 개발하여 적용하였다.

6.6.1 DNP 3.0 Monitoring module

Fig. 8. DNP 3.0 Monitoring Module

DNP 3.0은 기본적으로 Master, Slave 구조를 가진다. 또한 OSI(Open System Interconnection) 7 Layer에서 변형된 EPA(Enhanced Performance Architecture)가 적용된 Data-link Layer, Transport Layer, Application Layer로 나뉘어있다. 각 Layer의 분석을 통해 현재 전송 중인 메시지의 형태를 알 수 있다[6]. 모니터링 모듈은 On/Off 가 가능하게 구현하였다. Fig. 8은 모니터링 모듈로 분석한 DNP 3.0 프로토콜 데이터의 내용이다.

6.6.2 SDS-TL Header and Inner 프로토콜 Monitoring module

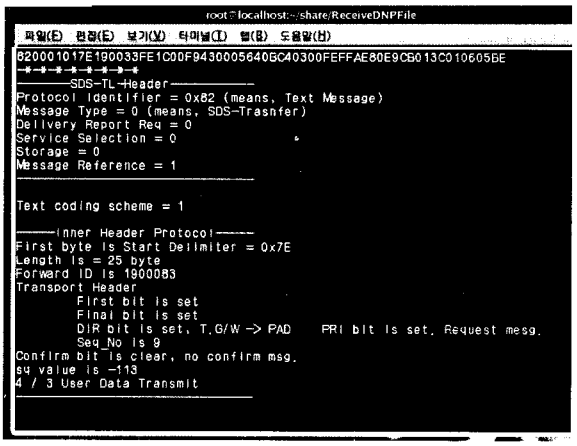

Fig. 9. SDS-TL Header and Inner Protocol Monitoring Module

TETRA 네트워크를 통한 송수신시 포함되는 SDS-TL header와 Inner 프로토콜에 대한 모니터링 모듈을 구현하여 TETRA 게이트웨이 플랫폼 모듈에 삽입하였다. Fig. 9는 각 모듈을 통해 현재 전송 중인 TETRA 메시지의 형태와 목적을 알 수 있다.

6.6.3 DNP3.0 over TETRA 전송 테스트

DNP 주장치 서버에서 송신된 DNP 데이터와 응답이 Fig. 10에서 나타나있다. 17:46:13에 송신된 Request에 2초 후 Response Message가 도착하였다. Fig. 11, Fig. 12는 TETRA 모뎀을 통해 데이터를 전송받은 화면을 구현한 임

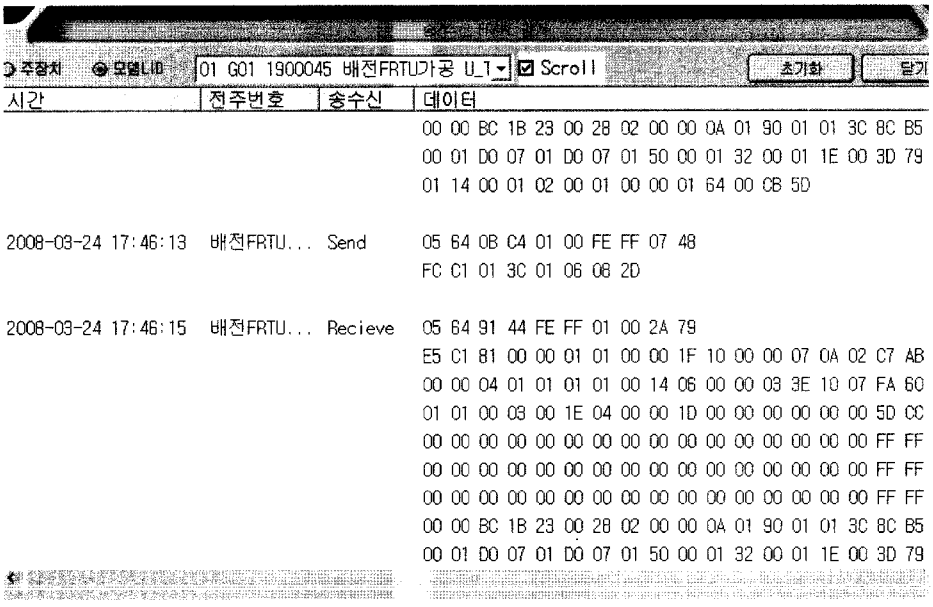

Fig. 10. DNP 3.0 Master Server

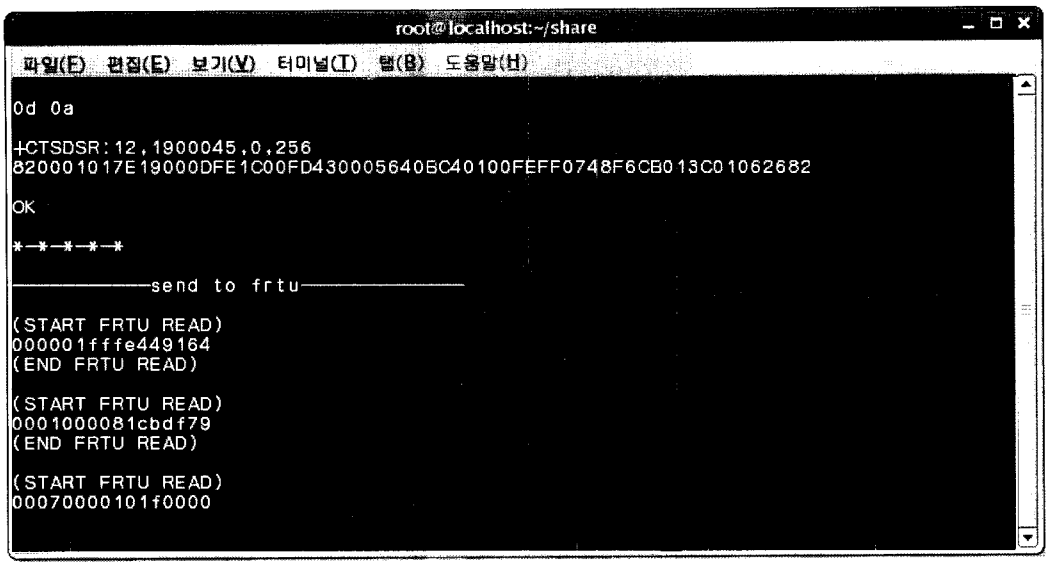

Fig. 11. Receive Data from FRTU(1)

베디드 리눅스 플랫폼의 콘솔 포트를 통해 출력한 화면을 캡쳐한 그림이다. Fig. 11에서 "+CTSDSR" 메시지로 시작하는 부분이 모뎀으로부터 받은 메시지 원본이다. TETRA 모뎀은 ASCII형태로 데이터를 전송하기 때문에 CR과 LF의 값을 확실하게 확인해 보기 위해서 Hexa 형태의 데이터를 한번 더 출력하였다. Fig. 11와 Fig. 12는 FRTU로부터

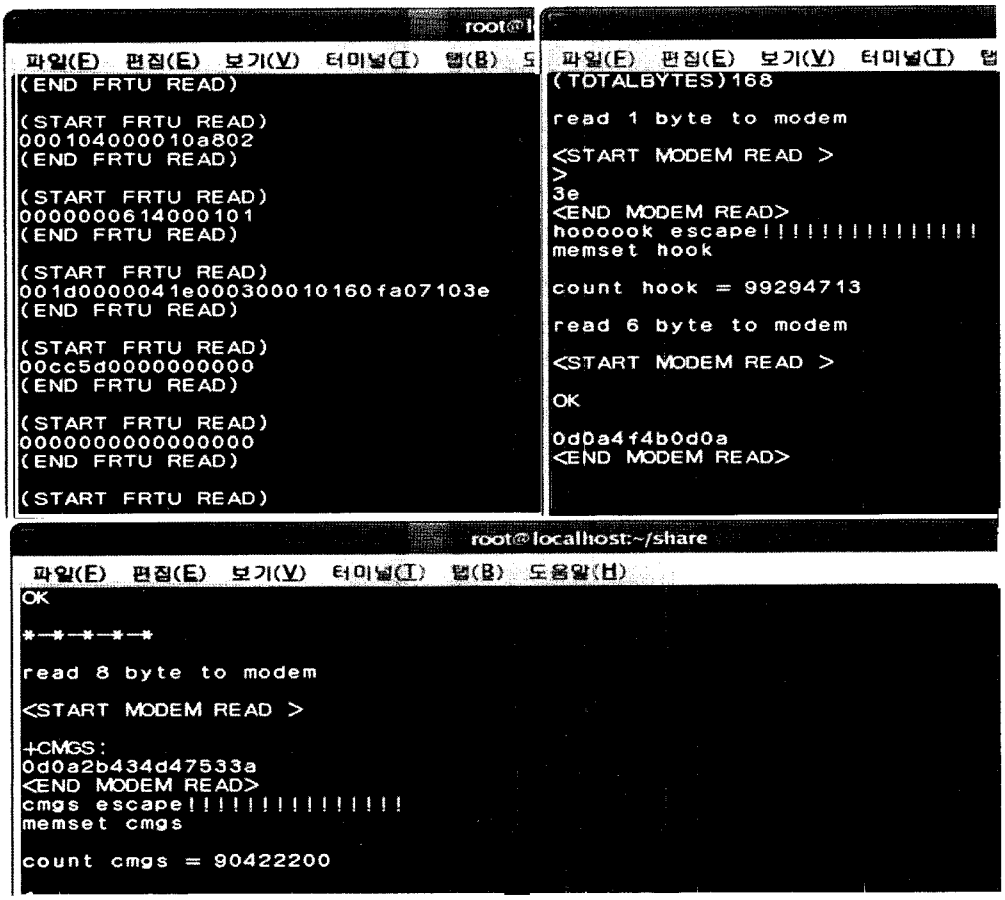

Fig 12. Receive Data from FRTU(2)

Response를 받아서 모뎀으로 전송하는 화면을 캡쳐하였다. FRTU로부터 Read 함수가 한번 호출 될 때마다 (START FRTU READ)메시지를 출력하였고, Read가 완료 되었을 때 읽은 총 바이트를 (TOTALBYTES)로 출력하였다. 데이터를 버퍼에 모아서 FRTU의 전송이 끝나면 해당 DNP 메시지를 위한 Inner 프로토콜을 생성하여 모뎀을 통해 전송한다. Fig. 12에서 모뎀을 통해 전송이 성공 했을 경우 모뎀으로부터 나오는 "+CMGS" 메시지를 확인할 수 있다.

7. 결론 및 향후 과제

전력 IT 기반 네트워크의 TETRA 네트워크로의 전환에 따라 앞으로 TETRA 네트워크를 이용한 다양한 산업용 프로토콜 데이터가 전송 될 것이다. 따라서 데이터 전송에 필요한 게이트웨이 플랫폼의 중요성이 대두 되고 있다. 그러나 기업에 따라 여러 가지 유형의 이기종 시스템들이 생김에 따라 이기종간의 연동에 문제 해결 역시 해결해야할 필수 문제이다.

본 논문에서 구현한 임베디드 리눅스 기반의 TETRA 게이트웨이 플랫폼은 자체적으로 구현한 TETRA API로 구현하여, TETRA 네트워크를 이용하여 산업용 전력 IT 프로토콜 중 하나인 DNP 3.0데이터를 전송하였다. 각각의 모듈에 따라 POSIX 기반의 스레드를 이용하여, POSIX를 지원하는 유닉스 기반의 OS에서는 문제없이 돌아가도록 구현하였다. 전송될 데이터를 받은 스레드에서 Inner 프로토콜을 생성하여 전송 모드에 따라 SDS 전송과 패킷 전송으로 나뉘어 TETRA 모뎀 제어를 통해 목적지로 전송된다. 구현한 TETRA API는 C언어로 구현였으며, Shared Library로 컴파일하여 다양한 목적으로 사용할 수 있다. 또한 EADS, Teltronic, Unimo 의 TETRA 모뎀을 기반으로 본 플랫폼을 테스트 하였다[7-9]. 각 모뎀 제조사 별 TETRA 네트워크로 전송시 명령어가 다르기 때문에 모뎀제조사 구별 루틴을 삽입하여 각각의 모뎀사 별로 전송하는 루틴을 구현하였다. 이와 같은 문제점을 해결해 줄 수 있는 표준화 작업이 필요하다.

본 논문에서 제안한 TETRA 게이트웨이 플랫폼 및 TETRA API은 데이터 전송이 필요한 서비스에 TETRA 네트워크와의 접목을 가능하게 해준다. 현재 철도 산업에 널리 사용되고 있는 센서를 이용한 다양한 관리 시스템에 적용되어 사용 할 경우 효과적으로 이용 할 수 있다. 예를 들면 기반 시설물 안전관리 네트워크나, 선로 유지관리 시스템 또는 지진 감시 시스템 등에서 각각의 센서들과 중앙 통제 시스템과의 통신에 TETRA 네트워크로가 사용된다면 국가 재난망으로 채택된 TETRA 무선 네트워크 자원의 효율적인 사용이 가능하고, 회선 임대에 사용되는 비용 절감 효과를 가져올 것이다.

참 고 문 헌

1. 전력자동화 디지털 TRS, http://www.kdn.com
2. Apostolis K. Salkintzis, Motorola "Evolving Public Safety Communication Systems by Integrating WLAN and TETRA Networks".
3. ETSI EN 300 392-2 "Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 2: Air Interface".
4. TRM-710.200,http://www.piciorgros.de/
5. ETSI EN 300 392-5 "Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 5: Peripheral Equipment Interface (PEI)".
6. DNP User Group, "Distributed Network Protocol DNP 3.0 BASIC 4 DOCUMENT SET".
7. EADS(TMR880i),http://www.eads.com/
8. Teltronic(MDT-400), http://www.teltronic.es/
9. Unimo(MU-1000),http://www.unimo.co.kr/
10. AutoLog® TETRA-RTU, http://www.ff-automation.com/
11. 오갑근, "국가통합지휘무선통신망 구축계획 및 현황", The Journal of the Korean Institute of Communication Sciences, Volume 23, p65-78, February 2006.
12. AT command, http://en.wikipedia.org/wiki/Hayes_command_set

접수일(2008년 6월 2일), 수정일(2008년 12월 9일),
게재확정일(2009년 2월 12일)