

## J·BOSN RTOS 개발하기⑧

# 매크로 커널-네트워크 서버

JBOSN RTOS에서 네트워크를 사용하려면 네트워크 서버를 반드시 연결시켜야 하며, 애플리케이션에서는 이 네트워크 서버의 도움으로 외부와 통신을 구현 사용할 수 있도록 되어 있다. 이러한 기능은 실시간시스템을 설계할 때 다량의 데이터를 송신, 수신할 수 있고, 이 데이터를 가공하는 시스템을 제공할 수 있으며, 제작된 시스템을 이용하여 대량의 데이터를 사용자에게 제공하거나 입력을 받아 사용자에게 제공하거나 임베디드 시스템의 제어 등을 처리할 수 있는 해결책을 제시한다.

글: 정병오 대표 / 아이보슨시스템즈 [www.jbosn.com](http://www.jbosn.com)

## 연재 차례

1. 나노커널(nano-kernel)
2. 시간서버(Time Server)

3. 시스템서버(System Server)
4. 동기화서버(Sync. Server)
5. 장치서버(Device Server)
6. 파일시스템서버(FileSystem Server)

7. 윈도우서버(Window Server)
8. 네트워크서버(Network Server)

JBOSN RTOS의 구조는 그림 1에 나타나 있다, 구조적인 측면에서 보면 그림 1에서 보이듯이 계층적이고 모듈화 되어 있는 각 모듈 중 JBOSN RTOS의 네트워크 통신에 관련된 기능을 제공하는 목적으로 제작된 것이 네트워크 서버(Network Server)이다. 운영체제의 측면에서 보면, 네트워크 서버는 네트워크를 통한 임베디드 단말장치와 외부와의 통신과 관련된 기능을 제공해주는 운영체제의 필수적인 부분이다.

JBOSN RTOS의 구조는 나노커널을 기본으로 하여 매크로 커널(시스템서버, 시간서버, 동기화서버)을 사용하면 순수한 운영체제의 핵심기능을 제공하여 준다. 매크로커널은 외부확장기능을 담당한다. 이곳에는 디바이스 관리와 통신을 위한 Device server, 블록디바이스의 데이터를 관리해 주는 파일시스템 서버인 File System Server, 네트워크 통신을 담당하고 있는 네트워크 서버, 그래픽과 윈도우 시스템을 제어하여 GUI를 제공하여 주는 윈도우 서버 등 4가지로 이루어져 있다.

네트워크 서버를 사용하여 외부와의 데이터 통신을 제공하고, 이를 이용하여 임베디드 제품에서 다량의 데이터를 송신/수신하고, 수신된 데이터를 이용하여 유용한 서비스를 제공하는 확장이다.

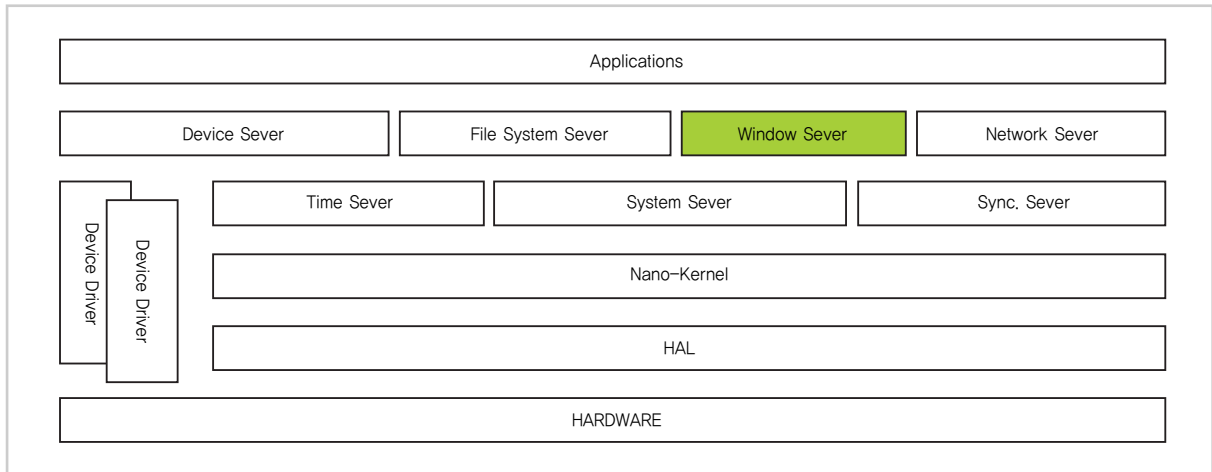


그림 1. JBOSN RTOS 구조

JBOSN RTOS에서 네트워크를 사용하려면 네트워크 서버를 반드시 연결시켜야 하며, 애플리케이션에서는 이 네트워크 서버의 도움으로 외부와 통신을 구현 사용할 수 있도록 되어 있다. 이러한 기능은 실시간시스템을 설계할 때 다량의 데이터를 송신, 수신할 수 있고, 이 데이터를 가공하는 시스템을 제공할 수 있으며, 제작된 시스템을 이용하여 다량의 데이터를 사용자에게 제공하거나 입력을 받아 사용자에게 제공하거나 임베디드 시스템의 제어 등을 처리할 수 있는 해결책을 제시한다.

### 네트워크 서버(Network Server)

이 모듈은 JBOSN RTOS의 사용자가 네트워크를 통하여 외부의 장치들과 통신을 할 수 있는 기능을 제공하는 역할을 담당하는 서비스를 제공해 주며 서버의 형태를 갖고 있다.

네트워크 서버는 IP 프로토콜을 기본으로 제작되었으며, UDP, TCP 통신 프로토콜과 직접 IP layer까지 접근할 수 있는 RAW 통신까지 제공하여 준다. IP layer는

IP 통신뿐만 아니라 제어를 위한 ARP, RARP, ICMP, IGMP 등을 제공하고 있다.

사용자는 소켓인터페이스를 통하여 네트워크 서버에 접근할 수 있도록 설계되어 있다. 애플리케이션에 속하는 다량의 프로토콜들도 소켓인터페이스를 통하여 네트워크서비스를 이용할 수 있다.

위와 같은 동작을 완성하기 위하여 네트워크 서버는 그림 2와 같은 구조를 가지고 있다.

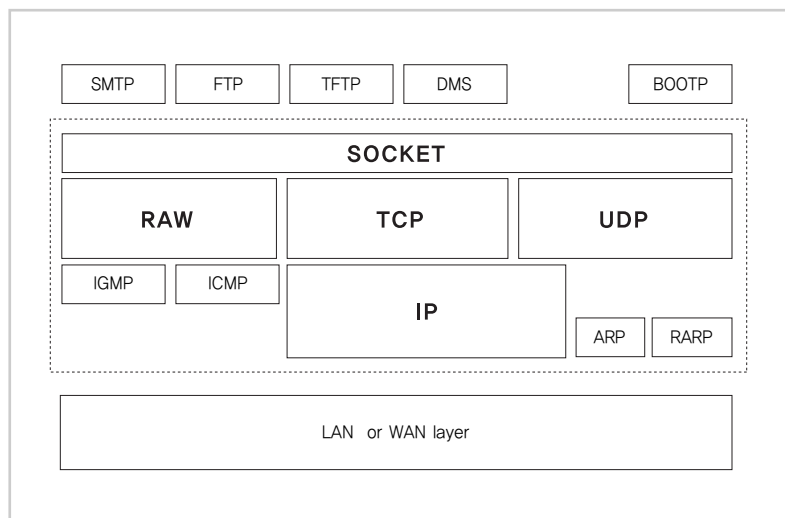


그림 2. 네트워크 서버 구성도

JBOSN RTOS는 네트워크 서버가 그림 2에서 보듯이 IP layer, RAW module, TCP module, UDP module과 사용자 인터페이스를 위한 Socket Interface를 포함하고 있다.

네트워크 서버는 네트워크 디바이스(NIC)들에 직접 IP packet을 전달하지 않고 네트워크 서버에서 전송할 데이터가 발생하면, NIC에서 IP packet을 네트워크 서버로부터 획득/송신하여 처리하도록 되어있다. 즉, 네트워크 서버는 실제 IP packet의 송수신에는 관여하지 않으며, IP packet의 생성/해석만을 담당한다. 이러한 구조는 네트워크 서버의 안정성을 향상시키며, 여러 개의 NIC 중 일부의 동작에 문제가 발생하더라도 네트워크 서버의 동작성에 영향을 미치지 못하게 한다. 결론적으로 매우 안정적인 네트워크 시스템을 구축할 수 있다.

다음은 네트워크 서버에서 제공하는 기능에 대하여 자세히 살펴보겠다.

### IP layer

IP 프로토콜은 신뢰할 수 없고 연결지향형이 아닌 프로토콜이다. 최선을 다해서 상대방에 전달되도록 노력을 할 뿐이다.

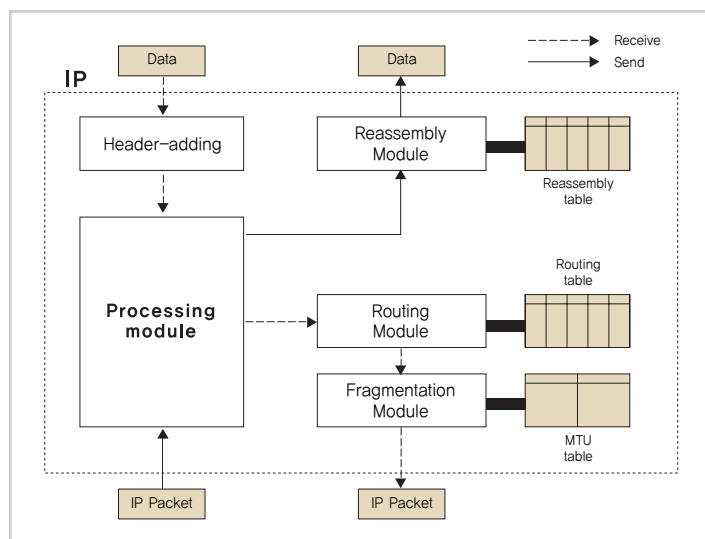


그림 3. IP layer 구성도

즉, 전달된 데이터가 완벽한지 에러검출도 하지 않으며, 전달된 경로를 추적하지도 않는다. 따라서 IP layer의 데이터는 신뢰성이 떨어지며, 실제 전달되었는지도 보장하지 않는다. 더군다나, IP packet들은 서로 간의 연결성 또는 의존성도 존재하지 않으며, 패킷단위로만 처리가 된다.

IP layer는 패킷을 가능한 범위 내에서 전달하는 것을 목적으로 설계가 되어 있어서 전달될 목적지와 도착한 패킷을 빨리 조합하여 상위의 레이어에 전달을 하여야 한다. IP layer의 구성요소들의 역할을 살펴보자.

#### Header-adding

이 모듈은 IP layer의 상위 레이어에서 데이터와 목적지 주소를 받는다. 주어진 데이터를 이용하여 IP 표준에 맞게 IP header를 구성하여 생성한 IP packet을 processing module에 전달한다. 이 과정에는 IP checksum을 계산하는 과정이 있는데, 산술적인 계산이 필요하여 다소 많은 CPU 자원을 사용한다.

#### Processing module

이 모듈은 IP layer의 핵심으로서 상위 레이어에서 내려오는 IP packet이나 하위 레이어에서 올라오는 IP packet을 동일한 알고리즘으로 해석/처리한다.

즉, IP packet이 어디에서 전달되었는지는 중요한 것이 아니고, IP packet의 목적지만을 보고 전달한다. 이 모듈의 출력은 다음과 같다.

목적지가 자신과 일치하여 IP packet을 상위 레이어로 올려 보내려면, reassembly module에 전달을 하고, 목적지가 자신이 아니어서 외부 IP 망으로 전달하려면, routing module로 전달을 한다.

부가적으로 이곳에서는 TTL(time-to-live)값을 감소시키는 역할을 한다. TTL이 '0' 보다 작거나 같으면 IP packet은 버려지게 되며, 에러메시지를 전송자에게 ICMP로 전달하게 된다.

## Routing module

이 모듈은 IP processing 모듈에서 외부로 전달될 IP packet을 입력받는다. 목적지를 보고, routing table을 참조하여 IP packet이 목적지로 최대한 빨리 전달될 수 있는 Network interface 장치와 다음에 전달될 주소를 찾아낸다. IP packet과 이 모든 정보를 fragmentation module에 전달한다.

## Fragmentation module

이 모듈은 routing module에서 전달된 IP packet을 MTU table의 값에 따라 적절한 단위로 쪼개어 Network interface 장치로 전송을 한다. 즉, 적절히 분리된 IP packet을 gateway 또는 router나 최종 목적지로 나누어서 보내게 된다.

이 모듈에서는 IP packet이 보내어질 인터페이스의 MTU(maximum transfer unit)값을 참조한다. IP packet이 MTU값보다 크면, 패킷의 데이터를 적절히 분리하고, IP header를 분리된 데이터에 붙여서 만든 IP packet들을 전송한다. 이 과정에서 ARP를 사용하여 보내어질 목적지의 정보를 얻어오기도 한다.

얻어진 정보를 이용하여 IP header를 재구성한다. 그리고 fragmentation이 발생하면 그에 해당하는 정보도 IP header에 추가한다. 사용자가 원한다면 추가적인 IP option을 덧붙일 수도 있다. 이 모듈도 IP packet의 옵션에 따라('D' option-do not fragment) fragmentation을 수행하지 못하고, IP packet이 버려지면, ICMP로 에러메시지를 전송자에게 전달한다. 이상의 모듈들은 IP packet을 전송하는데 필요한 것들이다.

## Reassembly module

이 모듈은 IP packet을 수신할 때 사용한다. 전달된 IP packet의 수신자가 자신일 경우에만 이곳으로 IP packet이 전달된다. 수신된 IP packet들은 fragmentation이 되었든, 안

되었든 이곳을 통과해야만 상위의 layer로 전달이 된다.

IP protocol은 연결형 프로토콜(connection oriented)이 아니므로, 전달되는 IP packet들이 분리된 순서대로 전달되는 것을 보장하지 않는다. 또한 분리된 서로 다른 IP packet들이 서로 섞여서 도착하기도 한다. 이런 상황에서 분리된 IP packet의 조각들을 받아서 완벽히 하나의 IP packet을 재구성하는 것이 이 모듈의 역할이다.

이 모듈에 도착한 모든 패킷들은 reassembly table에 삽입이 된다. Reassembly table에는 도착한 패킷들의 상태, 출발지 주소, ID, time-out 정보 그리고 데이터에 따라 배치된다. 이상의 조건들이 일치하거나 재조합이 가능한 패킷들은 서로 연결하여, 완벽히 하나의 IP packet으로 조합되는 것은 상위의 프로토콜로 전달이 된다. 그러나 일정시간이 지나도 재조합이 이루어 지지 않는 패킷들은 삭제가 된다.

추가적으로 빠른 패킷 처리를 위하여 fragmentation이 되지 않는 IP packet은 곧바로 상위 레이어로 전달이 된다. 물론 상위 모듈 중 IP packet에 지정된 프로토콜을 처리할 수 있는 상위 protocol layer로 전송이 된다.

이상은 IP layer에서 packet이 전달되는 모듈들을 설명하였다. 각 모듈들은 모두 IP packet data로 연결이 되어 있으며, 이 데이터들은 복제가 되지 않고, 주소 값만을 전달하여 빠른 응답속도를 낼 수 있도록 구현되었다. 즉, 상위에서 내려오거나 하위에서 올라오는 모든 IP packet은 복제되지 않고 곧바로 상위나 하위 layer로 전달이 된다.

## UDP layer

다음은 UDP layer에 대하여 살펴보자. UDP는 connectionless protocol로서 비교적 단순한 프로토콜에 속한다. IP protocol의 특성을 가장 잘 대변하고 있다고 해도 될 것이다. UDP packet을 전달하기 위하여 control block module, Input module, Output module 등으로 이루어져 있다. 각 모듈의 기능에 대하여 살펴보자.

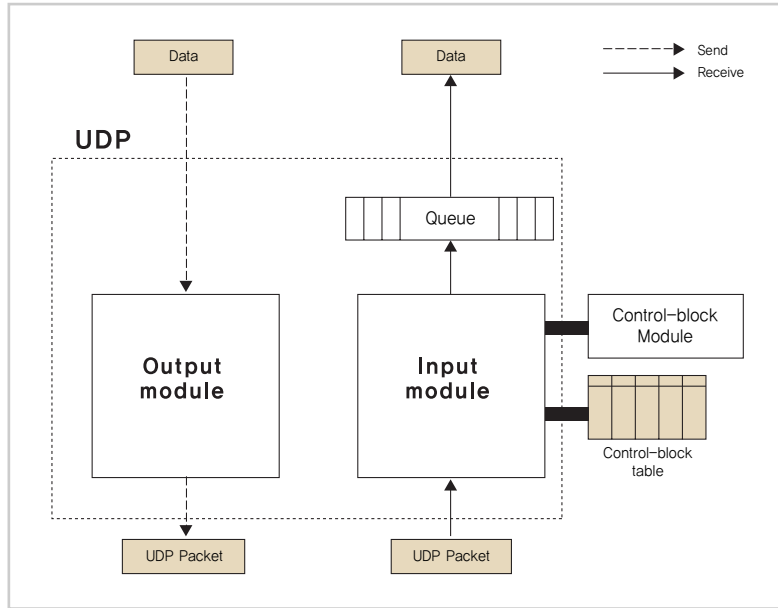


그림 4. UDP layer 구성도

이 모듈은 IP layer에서 올라오는 IP packet에서 전달되는 UDP port를 추출한다. 추출된 UDP port는 control-block module을 참조하여 전달되고자 하는 포트에 연결된 프로세스의 데이터 큐에 패킷을 전달한다.

애플리케이션에서 소켓인터페이스를 통하여 전달된 데이터를 사용하여 UDP packet을 생성하고 하단의 IP layer로 패킷을 전달한다. JBOSN RTOS에서 제공하는 network packet을 할당받아 UDP packet들을 생성한다.

### TCP layer

#### Output module

이 모듈은 애플리케이션에서 소켓인터페이스를 통하여 전달된 데이터를 사용하여 UDP packet을 생성하고 하단의 IP layer로 패킷을 전달한다. JBOSN RTOS에서 제공하는 network packet을 할당받아 UDP packet들을 생성한다.

#### Input module

Control-block table은 사용되고 있는 UDP 포트의 정보를 저장하고 있다. 해당하는 포트의 사용 여부와 사용하고 있는 thread, 그리고 연결된 데이터 큐에 대한 정보를 가지고 있다.

UDP 포트를 사용하려는 thread는 control-block module을 사용하여 control-block table을 제어한다. 즉, UDP 포트를 사용하려고, 포트번호를 요청하면, 사용되지 않는 포트를 넘겨주거나, 요청한 특정 포트가 사용되지 않으면 control-block table에서 포트를 할당하여 준다. 이 때 사용될 데이터 큐도 할당된다. 그 외에 필요한 정보를 control-block table에 삽입한다.

TCP protocol은 스트리밍 서버이며, state-transition diagram을 가지고 있는 연결지향형(connection oriented)의 매우 복잡한 프로토콜이다. 또한 흐름/에러 제어기능을 가지고 있다.

#### TCB table

TCP protocol은 connection-oriented 프로토콜이고, 매우 오랜 기간 동안 상호 연결되어 있다. 연결을 제어하기 위하여 매우 다양한 정보를 보관하고 있어야 한다. 이러한 모든 정보를 보관하고 있는 것이 TCB table이다. 동시에 여러 개의 연결이 유지되기 때문에 많은 수의 TCB가 존재한다. 이 구조체는 TCP 연결에 필요한 나 자신의 정보뿐만 아니라 연결된 상대방의 정보를 포함하여 현재의 연결 상태와 데이터의 이동에 필요한 버퍼와 상태 데이터까지 포함하고 있다.

#### Timers

TCP protocol은 많은 수의 Timer를 사용하고 있다. Retransmission Timer는 IP protocol의 특성 때문에 전송

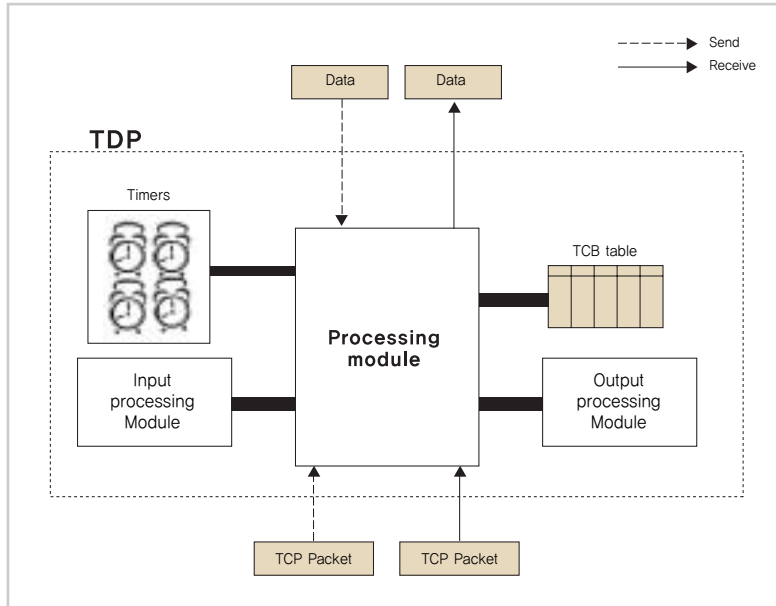


그림 5. TCP layer 구성도

중 손실된 IP packet을 재전송하기 위하여 사용한다. 전송된 TCP packet에 대한 Ack가 일정시간 후에도 도착하지 않으면 재전송을 시도한다. Ack가 도착하면 해당한 TCP packet에 연결된 Retransmission Timer는 멈추게 된다.

Persistence Timer는 TCP protocol의 한계를 극복하기 위하여 사용된다. TCP 통신은 동시에 보낼 수 있는 최대 데이터 크기를 정해놓는데 이것이 윈도우 크기이다. 그러나 Ack를 받지 못하고 지속적으로 데이터를 보내게 되면 할당된 윈도우 크기가 줄어서 나중에는 데이터를 보낼 수 없는 상황에 도착할 수도 있다.

즉, 윈도우의 크기가 0이 될 수 있다. 이 상황을 벗어나려면 Ack를 수신자 측에서 받아야 한다. 그러나 IP 통신의 특성 때문에 Ack가 송신자에게 전달되지 않을 수 있다.

결과적으로, 송신자와 수신자 모두 데이터를 주고받지 못하는 상태가 될 수 있다. 이러한 상황을 피하기 위하여 윈도우 크기가 0에 도달하려면, persistence timer가 동작한다. 일정시간이 지나면 송신자 쪽에서 probe라는 패킷을 지속적으로 전달하여 Ack의 재전송을 유도하여 이 상황을 벗어나려고

노력한다. JBOSN RTOS는 이러한 상황을 만들지 않기 위하여 다양한 알고리즘을 사용하여 상호 deadlock이 발생하는 것을 막고 있다.

Keepalive Timer는 연결은 되어 있지만, 데이터의 송/수신이 오랜 기간 동안 이루어지지 않는 경우 송/수신자가 상호간의 연결 상태를 확인하기 위하여 주고받는 Keepalive packet을 전송하는 시간을 정하여 준다.

즉, 정상적인 연결이 유지되고 있는지, 아니면 연결의 상대방에 문제가 발생하여 데이터 송/수신이 안 되는지 검사를 할 수 있다.

JBOSN RTOS는 embedded 제품에 사용되므로 상대방의 상태를 빠르게 감지하기 위하여 빈번히 keepalive packet을 전송하도록 되어 있으며, 해당하는 응답이 없을 경우 연결을 강제로 종료하고 있다.

Time-Wait Timer는 TCP 연결을 종료하는 경우에 사용된다. TCP 연결을 종료할 때, 실제로 연결이 송/수신자간에 합의가 이루어져 종료가 되었는지 알 수가 없다. 또한 종료되는 동안 패킷이 전송이 되어 전송되고 있는 중간일 수도 있다. 따라서 이 두 가지 경우를 방지하기 위하여 종료할 때, 일정시간(2MSL)동안 연결 상태를 유지하여, 중복된 FIN을 방지하고, 늦게 도착한 TCP packet을 제거한다.

#### Input module

이 모듈은 TCP 연결이 완성되었을 때(Established), 수신된 모든 TCP packet을 받아 필요한 정보를 뽑거나, 상태를 갱신한다. 필요할 경우 ACK를 전송하도록 main module에 명령할 수도 있고, 윈도우 크기를 조절하거나, 에러체크를 수행하며, TCP packet에 대한 모든 검사를 수행하고, 내부의 변수들을 갱신한다.

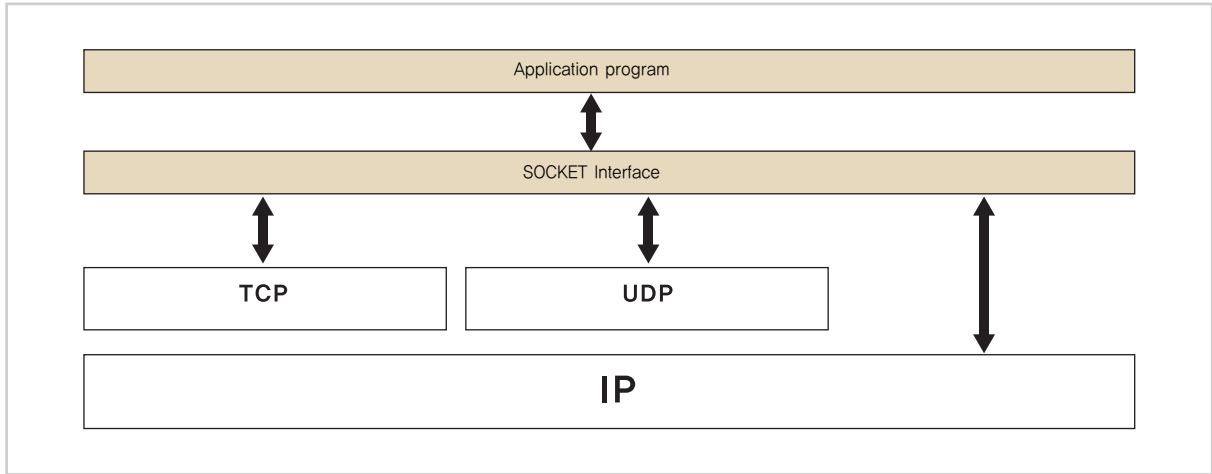


그림 6. Socket layer 구성도

**Output module**

이 모듈은 TCP 연결이 완성되었을 때, 송신된 데이터를 전송하기 위하여 필요한 작업을 하는 곳이다. 대표적으로 각종 Timer들을 제어하여 TCP packet의 원활한 흐름을 만들어 준다. 윈도우 크기를 적절히 제어하고, 재전송 및 연결 상태를 유지하기 위한 각종 노력을 담당하는 곳이다.

**Main module**

이 모듈은 상위 layer에서 전송되는 데이터나 IP layer에서 입력되는 TCP packet들을 받아들인다. 또한 각종 Timer들에서 발생하는 이벤트와 입력된 TCP packet데이터의 상태에 따라 동작을 결정한다.

TCP 연결의 모든 상태에 따라 입력된 TCP packet과 이벤트들을 처리하여야 하므로 매우 복잡한 모듈이다.

JBOSN RTOS는 11가지의 TCP 연결 상태를 기준으로 TCP packet의 입력과 Timer 이벤트와 Socket 인터페이스를 통한 사용자의 명령들을 모두 이벤트로 처리하여 이 모듈을 구성하였다.

입력된 이벤트의 처리 결과는 Socket

인터페이스를 통하여 사용자에게 전달되거나, Output module을 통하여 IP layer로 전송이 된다. 물론 내부 상태의 천이도 이루어진다.

**Socket Interface**

소켓이란 네트워크 통신을 하기 위한 소켓프로그램에서 사용하는 데이터 구조체이다. 소켓은 통신 family, 주소 그리고 해당포트로 정의된다. 소켓 인터페이스는 3가지의 종류로 나눌 수 있다. Stream socket, datagram socket, raw socket

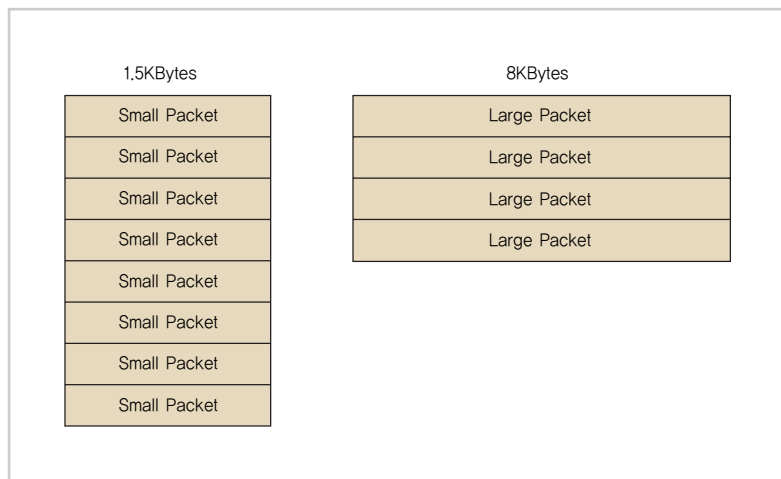


그림 7. Internal packet data structure


이다. JBOSN RTOS는 모든 것을 제공하고 있다.

Stream socket은 connection-oriented protocol인 TCP 통신을 사용할 수 있게 도와준다. TCP 통신은 네트워크를 통하여 연결된 streaming socket의 쌍을 사용한다. Datagram socket은 connectionless 형의 UDP 통신을 위하여 사용된다. UDP 통신도 마찬가지로 datagram socket의 쌍을 유지하고 있다. Raw socket은 TCP나 UDP를 통하지 않고 직접 IP layer에 접근하기 위하여 사용된다.

JBOSN RTOS는 표준적인 소켓인터페이스들(NwSocket, NwBind, NwConnect, NwListen, NwAccept, NwSendTo, NwRecvFrom, NwRead, NwWrite, NwClose)을 지원하여 준다. 비동기 소켓프로그램을 구현하기 위하여 NwSelect 함수도 제공한다.

## Internal Data Structure

JBOSN RTOS는 패킷처리의 속도를 높이기 위하여 통일된 데이터 패킷을 가지고 있다. 작은 패킷을 처리하기 위하여 small packet이라는 1.5Kbytes의 용량의 데이터를 처리할 수 있는 것과 8Kbytes까지 대용량의 데이터를 한꺼번에 처리할 수 있는 large packet이라는 것이다.

JBOSN RTOS의 최대 처리 패킷 크기는 8Kbyte로 제한이 된다. 네트워크 서버에서 처리되는 모든 데이터는 반드시 두 개의 packet 데이터 구조에 맞추어 전송이 되므로 매우 빠른 처리가 가능하며, 통일된 패킷관리가 가능하게 된다. 

» 연재끝

# Saving Your Time

전자부품 웹진은 제작비용, 및 유통비용, 판매가격이 상대적으로 인쇄책자보다 저렴하며 수정 및 재가공이 가능합니다. 또한 멀티미디어 기능이 있어 표현이 다양하고 광고효과가 뛰어난 장점을 지니고 있습니다.

» 인터넷에서 읽는 전자부품  
e-Magazine

02-2026-5700  
www.epnc.co.kr