

# Hardware protocol stack에서 free buffer size 결정 방법

## The decision method of freebuffer size in hardware protocol stack

문춘경, 김영근

Moon Choon-Kyoung, Kim Young-Keun

**Abstract** - Hardware implemented ring buffer systems and methods are presented for the effective management of the ring buffer in TCP/IP communication. The layer interface of the ring buffer systems transfer free buffer and used buffer size information to the TCP/IP stack upper or low layer. The pointer updation interface calculates a temporary pointer from the data size which is needed by the present pointer of the ring buffer and upper or lower layer. The pointer manager of the ring buffer systems is responsible for saving the present pointer of the ring buffer, updating the ring buffer pointer to the new pointer, calculating the free buffer size and used buffer size of the ring buffer, and transferring the information to the upper layer. The ring buffer systems help the TCP/IP layer and TCP/IP upper or lower layer to decide the sending or receiving data size effectively. The delay of transferring data can be lowered by the ring buffer system.

**Key Words** : Hardware TCP/IP, ring buffer

### 1. 장 서론

인터넷 보안 카메라 휴대용 단말기 등의 분야에서 상용화된 ASIC 형태의 hardware TCP/IP가 사용되고 있으며 기존 소프트웨어 방식의 단점을 보완하여 고속 패킷 처리가 가능하다. hardware TCP는 고성능 플랫폼에서 동작하는 소프트웨어 방식 보다 뛰어난 성능을 발휘할 수 있다.[1] 사용자로부터 필요한 정보를 받아 네트워크를 통해 정보를 전달하기 위해서는 데이터의 임시 저장 공간인 버퍼(Buffer)가 반드시 필요한데, 이러한 버퍼의 크기는 물리적으로 무한대가 될 수 없기 때문에 소프트웨어적인 기법으로 메모리 버퍼(M-buffer)를 구현하여 버퍼의 크기 제약을 없애고 있다. 그러나, TCP/IP 스택을 하드웨어적으로 처리할 경우에는 버퍼의 크기를 가변적으로 설정하기가 매우 어렵다. 따라서, 이러한 문제를 해결하기 위해 링 버퍼(Ring Buffer)를 이용하는데, 링 버퍼를 이용하기 위해서는 버퍼 포인터의 관리와 링버퍼의 프리 버퍼 사이즈(free buffer size) 또는 사용중인 버퍼 사이즈(using buffer size)를 판단하고 알려주는 것이 중요하다.

### 2. 소프트웨어 기법의 Hardware TCP/IP 적용 문제점

데이터를 송신하는 경우에 링 버퍼에 송신할 데이터를 쓰기

위해 링 버퍼의 프리 버퍼 사이즈를 결정하여 TCP 계층에 알려주는 프로세스는 일단 TCP 계층에서 원하는 양의 데이터 패킷을 전송하기 전에 FBS(Free Buffer Size) 함수를 호출하고, FBS 함수는 현재의 프리 버퍼 사이즈를 계산하여 송신해야 할 데이터 사이즈와 비교하여 프리 버퍼 사이즈가 송신 데이터 사이즈보다 커지는 시점에서 버퍼에 송신 데이터를 쓰고 버퍼 매니저의 소정 영역의 Send Bit를 1로 설정한다.

이와 같이, TCP/IP 프로토콜 스택이 링 버퍼에 데이터를 쓰거나 읽기 위해서는 그 전에 반드시 링 버퍼의 프리 버퍼 사이즈 또는 사용중인 버퍼 사이즈의 계산을 요청해야 하고, 송수신할 데이터 사이즈가 계산된 프리 버퍼 사이즈 또는 사용중인 버퍼 사이즈보다 클 경우에는 프리 버퍼 사이즈 또는 사용중인 버퍼 사이즈가 송수신 데이터 사이즈보다 커질때까지 기다려야 하므로 데이터 전송 지연이 발생한다는 문제점이 있었다.

또한, 송수신할 데이터를 버퍼에 쓰거나 읽은 다음에 바로 버퍼 포인터를 업데이트하기 때문에 TCP/IP 스택에서 송수신 데이터를 취소해야 하는 등의 사정이 발생할 경우에 업데이트된 버퍼 포인터를 복구하기가 매우 어렵다는 문제점이 있었다.

따라서, TCP/IP 통신을 위해 하드웨어적으로 링 버퍼 관리를 구현함에 있어서, 데이터를 링 버퍼에 쓰거나 읽으려고 하는 TCP 계층과 IP 계층과 TCP/IP 스택의 상위 계층 모듈이 링 버퍼의 프리 버퍼 사이즈 또는 사용중인 버퍼 사이즈를 바로 알 수 있도록 하여 적절하게 송수신 데이터 사이즈를 조절하게 함으로써 데이터 전송 지연을 방지하는 것이다.

또한, TCP/IP 통신을 위해 하드웨어적으로 링 버퍼 관리를 구현함에 있어서, TCP/IP 스택에서 포인터 업데이트 신호가 있기 전까지는 버퍼 포인터의 업데이트를 보류시킴으로써 효율적인 버퍼 포인터 관리를 실현하는 것이다.

저자 소개

\* 正會員 : 三星電子 디지털 미디어 研究所 研究員

\*\* 正會員 : 三星電子 디지털 미디어 研究所

### 3. 장 Freebuffersize관리 시스템의 구조와 원리

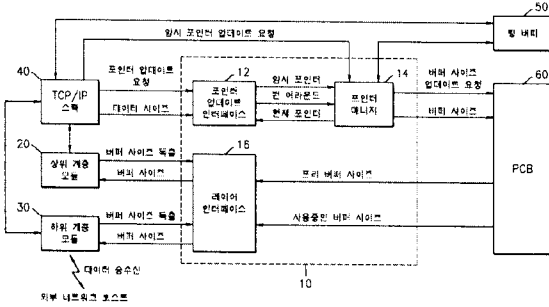


그림 1 Ring Buffer 관리 시스템 구조

그림 1에는 링 버퍼 관리 시스템이다. 링 버퍼 관리 시스템은 포인터 업데이트, 인터페이스 포인터 매니저, 및 레이어 인터페이스로 이루어져 있으며, 하드웨어 스택으로 구현되어 있다. 링 버퍼는 데이터의 읽기 지점을 나타내는 라이트(write) 포인터와 데이터의 쓰기 지점을 나타내는 리드(read) 포인터를 갖는다.

포인터 업데이트 인터페이스는 TCP/IP 스택의 포인터 매니저로부터 포인터 업데이트 요청, 링 버퍼에 쓰거나 읽을 데이터 사이즈 및 링 버퍼의 현재 포인터를 수신하여 임시 포인터를 생성하여 포인터 매니저로 전달한다.

임시 포인터는 링 버퍼의 현재 포인터와 TCP/IP 스택에서 전달한 데이터 사이즈의 합에 의해 결정되며, 임시 포인터는 나중에 TCP/IP 스택으로부터 임시 포인터 업데이트 요청을 수신한 경우에 비로소 새로운 포인터로 업데이트된다. 그 이유는, TCP/IP 스택에서 TCP/IP 패킷을 처리하는 과정에서 어떤 이유에서든지 송수신 데이터의 처리를 중지하거나 취소해야 할 사정이 발생한 경우에 미리 버퍼 포인터를 업데이트해 버리면 이를 복구하기가 어렵기 때문이다.

또한, 포인터 업데이트 인터페이스는 임시 포인터를 계산할 때, 현재의 라이트 포인터와 데이터 사이즈의 합이 링 버퍼의 최상위 포인터를 초과하는지를 판단하여, 턴 어라운드 비트(TAB)를 생성하고 이를 포인터 매니저로 전달한다.

그림 3에는 임시 포인터의 계산과 턴 어라운드 비트(TAB)의 생성을 설명하기 위한 개념도가 도시되어 있다.

링 버퍼는 최상위 포인터 m과 최하위 포인터 n사이의 저장 공간을 가지며, Pw는 데이터 송신시에 링 버퍼에 데이터를 쓰기(write) 시작할 지점을 나타내며 Pr은 데이터 수신시에 링 버퍼로부터 데이터를 읽기(read) 시작할 지점을 나타낸다.

그림 2의 (a)는 링 버퍼에 데이터를 쓰기 위해 TCP/IP 스택에서 Ds의 데이터 사이즈를 포인터 업데이트 인터페이스에 요구함으로써 임시 Pw가 최상위 포인터 m을 초과해 버릴 경우를 설명한다. 물론, 송신 데이터의 크기가 최상위 포인터 m을 초과하지 않은 경우에는 임시 Pw는 Pw + Ds로 결정된다.

즉, TCP/IP 스택에서 요구한 데이터 사이즈가 현재의 라이트 포인터로부터 링 버퍼의 최상위 포인터를 초과하는 경우에는 임시 Pw는 Pw + Ds + n - m으로 결정되며, 턴 어라운드 비트(TAB)는 1로 설정된다. 이 때, 임시 Pw는 현재의 리드 포인터 Pr를 초과할 수 없는데 이것은 구조상 상위 계층 모듈 또는

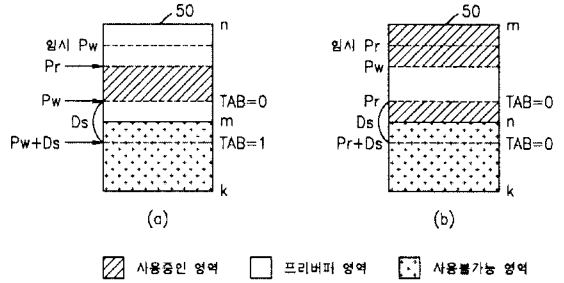


그림 2 Pointer 갱신 과정

하위 계층 모듈이 레이어 인터페이스로부터 링 버퍼에 쓰거나 읽을 데이터의 크기를 거의 실시간으로 항상 확인할 수 있기 때문에 가능한 것이다.

그림 2의 (b)의 경우에도 그림 2의 (a)와 동일하게 임시 Pr를 계산할 수 있다. 따라서, 현재의 포인터를 Pp라 하고 임시 포인터를 Pn이라고 하면 다음의 [식 1]에 의해 임시 포인터 Pn이 결정된다.

$$[식 1]$$

$$Pn = Pp + Ds : (Pp + Ds) \leq m$$

$$Pn = Pp + Ds + n - m : (Pp + Ds) > m$$

포인터 업데이트 인터페이스에서 계산된 임시 포인터와 턴 어라운드 비트를 전달받은 포인터 매니저는 TCP/IP 스택으로부터 임시 포인터 업데이트 요청이 있는 경우에 링 버퍼의 현재 포인터를 임시 포인터로 업데이트한다. 또한, 포인터 매니저는 링 버퍼의 프리 버퍼 사이즈와 현재 사용중인 버퍼 사이즈를 계산하여 버퍼 사이즈 업데이트 요청과 버퍼 사이즈를 프로토콜 제어 블록(PCB)으로 출력한다. 프로토콜 제어 블록(PCB)은 포인터 매니저의 요청에 따라 링 버퍼의 버퍼 사이즈를 업데이트하고 업데이트된 버퍼 사이즈를 레이어 인터페이스로 전달한다.

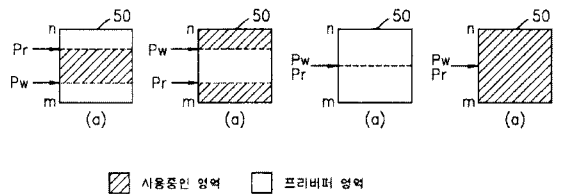


그림 3 free size 계산

그림 3에는 프리 버퍼 사이즈 계산을 설명하기 위한 개념도도 도시되어 있다. 링 버퍼는 최상위 포인터 m과 최하위 포인터 n사이의 저장 공간을 가지며, Pw는 데이터 송신시에 링 버퍼에 데이터를 쓰기(write) 시작할 지점을 나타내며 Pr은 데이터 수신시에 링 버퍼로부터 데이터를 읽기(read) 시작할 지점을 나타낸다. 이 때, 프리 버퍼 사이즈를 FBS라고 하면, FBS는 다음의 [식 2]에 의해 결정된다.

[식 2]

$$FBS = Pr - Pw : Pr > Pw$$

$$FBS = m - n - Pw + Pr : Pr < Pw$$

$$FBS = 0 : Pr = Pw \text{ 이고, } TAB = 1$$

$$FBS = m - n : Pr = Pw \text{ 이고, } TAB = 0$$

[식 2]에서 턴 어라운드 비트(TAB)는 Pw가 m을 초과하는 순간에 1로 설정되며 링 버퍼가 모두 비워질 경우에 0으로 설정된다. 따라서, 링 버퍼의 초기에는 TAB가 0으로 설정되어 있다가, Pw가 m을 초과하는 순간에 1로 설정된다.

한편, TCP/IP 스택은 상위 계층 모듈과 하위 계층 모듈로부터 링 버퍼에 쓰거나 읽을 데이터의 사이즈를 전달받아 링 버퍼의 포인터를 업데이트하도록 포인터 매니저에게 요청하며, 일반적으로 TCP/IP 통신에서 상위 계층 모듈에서 전달받은 데이터를 TCP 패킷 및 IP 패킷으로 변환하여 하위 계층 모듈을 통해 네트워크로 출력하며, 하위 계층 모듈에서 수신한 IP 패킷에서 TCP 패킷을 분리하고 상위 계층 모듈에게 전달하는 역할을 한다. 최근에는, TCP/IP가 소프트웨어적으로 처리될 때의 순차적 처리의 단점을 보완하고 시스템 운영 체제(OS)의 부담을 줄이면서 데이터의 병렬 처리를 구현하기 위해 하드웨어적으로 구현하는 TCP/IP 하드웨어 스택이 개발되고 있는 추세다.

그림 4에는 링 버퍼 관리 방법의 흐름도가 도시되어 있다. 데이터 사이즈 수신 단계에서는 포인터 업데이트 인터페이스가 TCP/IP 스택으로부터 링 버퍼에 쓰거나 읽을 데이터 사이즈를 수신한다. 임시 포인터 계산 단계에서는 데이터 사이즈와 링 버퍼의 현재 포인터를 바탕으로 임시 포인터를 계산한다. 또한, 데이터 사이즈와 현재 포인터의 합이 링 버퍼의 최상위 포인터를 초과하는 경우에 턴 어라운드 비트를 1로 설정하고 이를 추가로 출력한다. 이 때, 현재 포인터는 포인터 매니저에서 [식 1]을 바탕으로 계산하여 포인터 업데이트 인터페이스로 출력한다. 링버퍼 포인터 업데이트 단계에서는 포인터 매니저가 임시 포인터로 링 버퍼의 포인터를 업데이트하며, 이 때 포인터 업데이트는 TCP/IP 스택(40)으로부터 임시 포인터 업데이트 신호를 전달받은 경우에 수행한다.

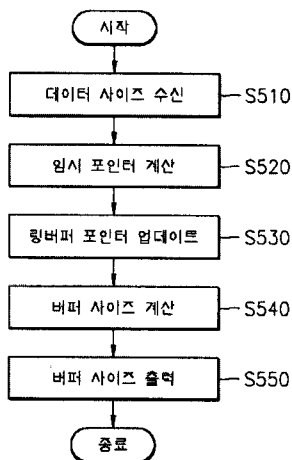


그림 4 링 버퍼 관리 흐름

버퍼 사이즈 계산 단계에서는 포인터 매니저가 [식 2]를 바

탕으로 링 버퍼의 프리 버퍼 사이즈와 사용중인 버퍼 사이즈를 계산하여 버퍼 사이즈 출력 단계에서 버퍼 사이즈를 출력한다. 따라서, 상위 계층 모듈과 하위 계층 모듈은 버퍼 사이즈 출력 단계에서 출력된 버퍼 사이즈를 바탕으로 링 버퍼에 쓰거나 읽을 데이터 사이즈를 적절하게 결정할 수 있다.

#### 4. 장 결론

데이터를 링 버퍼에 쓰거나 읽으려고 하는 TCP/IP 스택의 상하위 계층 모듈이 링 버퍼의 프리 버퍼 사이즈 또는 사용중인 버퍼 사이즈를 바로 알 수 있도록 하여 적절하게 송수신 데이터 사이즈를 조절하게 함으로써 데이터 전송 지연을 방지 할 수 있다.

또한, TCP/IP 스택에서 포인터 업데이트 신호가 있기 전까지는 버퍼 포인터의 업데이트를 보류시킴으로써 효율적인 버퍼 포인터 관리를 실현하는 효과가 있다.

#### 참 고 문 헌

- [1] Kyohong Jin, Jungtae Lee, "TCP/IP Hardware Implementation for the High Speed Networks", APSITT97, pp9.3.1-9.3.5, 1997.03
- [2] D.Comer, D.Stevens, "Internetworking with TCP/IP volume II," p200, prentice hall 1999