

## ptype\_base하쉬표를 리용한 망층규약확장의 한가지 방법

김원철, 박은성

위대한 령도자 김정일동지께서는 다음과 같이 교시하시였다.

《프로그램을 개발하는데서 기본은 우리 식의 프로그램을 개발하는것입니다. 우리는 우리 식의 프로그램을 개발하는 방향으로 나가야 합니다.》(《김정일선집》 증보판 제21권 42페이지)

론문에서는 Linux핵심부의 망접근층에서 실현되는 파के트수신과정을 분석하고 핵심부에 새로운 망층규약을 확장하는 한가지 방법을 제안하였다.

망층규약확장방법은 간접처리비용이 높은 보안처리연산[1]과 방화벽과 같은 전용보안장치[2]들을 리용하지 않으면서도 전용망통신을 보장할수 있다.

컴퓨터망에서 보안기능실현과 전용통신실현은 일반적으로 소프트웨어적으로 실현되는 암호화기술과 하드웨어적으로 실현되는 망결합장치들을 리용하여 실현하고있다. 이러한 보안기능실현들은 간접처리비용이 크고 컴퓨터망구축의 물리적인 변경을 동반하게 된다.

원천공개프로젝트로 개발되고있는 Linux조작체계에 전용망층규약을 추가하면 전용망통신과 보안통신을 조작체계준위에서 실현하여 고속통신을 보장할수 있다.

우리는 Linux조작체계의 핵심부준위에서 실현되는 파케트수신과정을 분석하고 조작체계준위에서 전용망통신을 보장할수 있는 망층규약확장의 한가지 방법을 제안하였다.

### 1. 망접근층의 파케트수신

Linux핵심부의 망구동프로그램과 망장치에서 실현되는 파케트의 수신처리과정은 그림 1과 같다.

망기판에 파케트가 수신되면 net\_interrupt는 망구동프로그램의 net\_rx를 호출한다.

net\_rx는 새로운 소켓트완충기를 창조하고 파케트내용을 망기판에서 소켓트완충기로 복사한다. 다음 망장치(net/core/dev.c)의 netif\_rx는 소켓트완충기를 CPU의 입력대기렬에 보관하고 소프트웨어중단을 설정한 다음 하드웨어중단을 완료한다.

망장치의 소프트웨어중단처리함수 net\_rx\_action()은 일정작성기 do\_softirq()에 의하여 호출된다. net\_rx\_action()함수는 연속순환(for(;;){...})을 반복하면서 수신파케트들을 규약식별자에 따라 망층규약실체에 넘긴다.(그림 2)

먼저 \_\_skb\_dequeue는 대기렬에서 수신파케트를 보관하고있는 소켓트완충기를 꺼내서 netif\_receive\_skb함수에 의하여 분석한다. 이때 수신대기렬이 빌 때까지 또는 핵심부에서 설정된 한계시간까지 소켓트완충기들을 꺼내서 수신처리를 진행한다.

netif\_receive\_skb함수는 파케트의 목적지주소와 원천지주소, 규약식별자를 분석하고 정확한 자료연결층파케트이며 국부로 수신되는 파케트이면 deliver\_skb함수를 호출한다. 다음으로 호출되는 deliver\_skb함수는 수신파케트의 규약식별자에 대응한 망층규약실체로 소켓트완충기를 넘긴다.

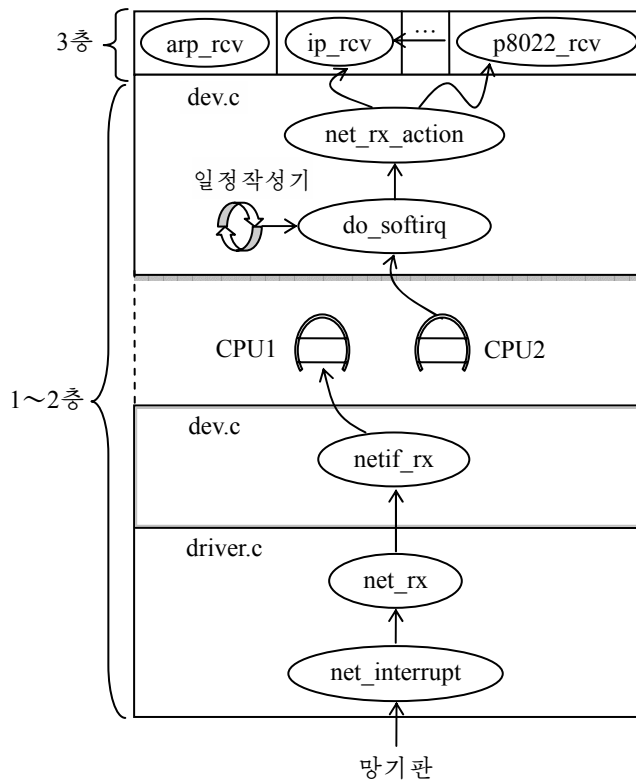


그림 1. Linux핵심부의 망구동프로그램과 망장치에서 실현되는 패킷의 수신처리과정

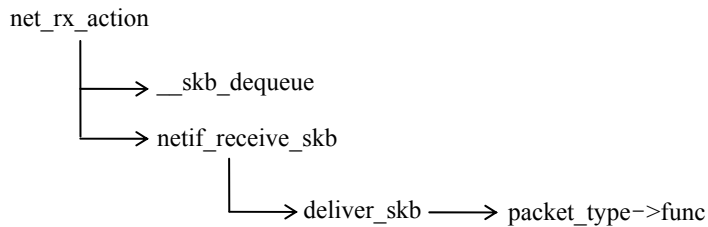


그림 2. 소프트웨어중단 net\_rx\_action의 처리과정

## 2. ptype\_base하쉬표를 리용한 망층규약확장방법

Linux핵심부에서는 ptype\_base[]하쉬표를 리용하여 망층규약들을 관리한다.(그림 3)

망장치에서 패킷을 수신하는 모든 망층규약들은 대역적인 배열 ptype\_base로 실현된 하쉬표에 등록된다.

하쉬표의 매개 입력점에 대응하는 packet\_type구조체는 매개 망층규약에 대하여 창조되며 핵심부에서 규약개체의 시작점위치를 지적한다.

```

struct packet_type {
    __be16 type;
    struct net_device *dev;
}
    
```

```

int (*func) (struct sk_buff *, struct net_device *,
             struct packet_type *, struct net_device *);
...
void *af_packet_priv;
struct list_head list;
};

```

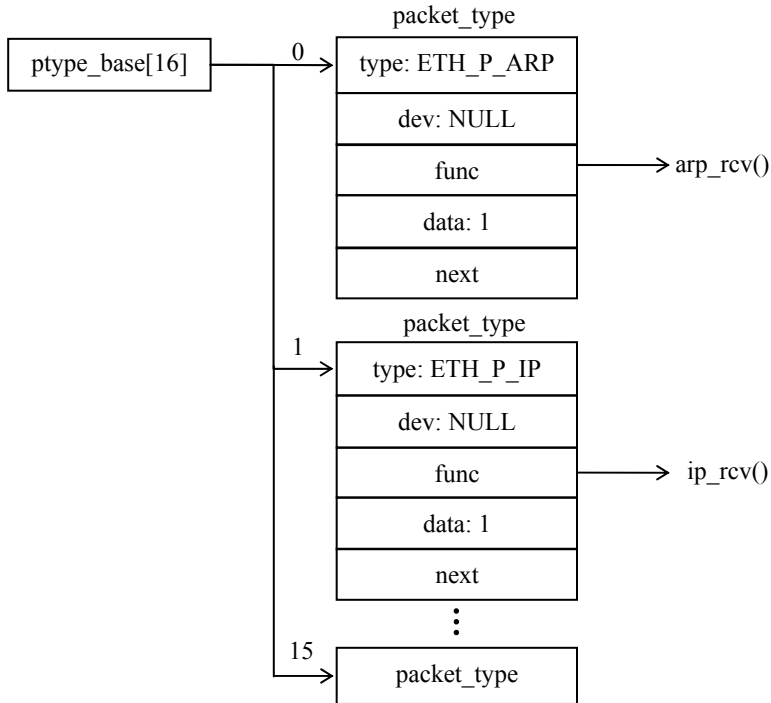


그림 3. Linux핵심부에서 망층규약관리

`packet_type`구조체에서 `type`는 자료연결층에서 망층규약들을 식별하는 식별자이며 `dev`는 망기관에 대응한 `net_device`구조체를 지적한다. `func`는 망층함수에 대한 지적자이며 `netif_receive_skb`는 수신된 소켓완충기를 대응한 `packet_type`구조체의 `func`로 지적한 망층함수에 전달한다.

다음 `ptype_base`하위표에서 `packet_type`구조체들을 추가하거나 삭제하기 위하여 핵심부는 `dev_add_pack`함수와 `dev_remove_pack()`함수를 제공한다.

Linux핵심부의 망층규약확장은 다음과 같다.

```

int gap_ip_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt);
static struct packet_type gap_ip_protocol =
{
    _constant_htons(GAP_PROTO_IP), NULL, gap_ip_rcv(),
    (void *) 1, NULL
};
int gap_ip_rcv(struct sk_buff *skb, struct net_device *dev, struct packet_type *pt)
{

```

```

        receive(skb)
        return process_state;
    }
    int init_module(void)
    {
        dev_add_pack(&gap_ip_protocol);
        return 0;
    }
    void cleanup_module(void)
    {
        dev_remove_pack(&gap_ip_protocol);
    }

```

핵심부모듈로 작성된 새로운 망층규약은 핵심부모듈의 추가과정에 type마당이 GAP\_PROTO\_IP인 packet\_type구조체를 ptype\_base하쉬표에 추가하며 모듈이 제거되면서 하쉬표에서 삭제된다.

핵심부모듈기동시에 추가되는 새로운 망층규약추가과정은 다음과 같다.

```

insert(gap_ip_protocol)
    h = hash(machdr->type)
    xpt = ptype_base[h]
    while(xpt != null)
        pt = pt(xpt)
        xpt = pt->next
    xpt = gap_ip_protocol

```

end

류형마당이 GAP\_PROTO\_IP인 경우 MAC프레임의 수신처리과정은 다음과 같다.

```

receive(skb)
    machdr = machdr(skb->data)
    h = hash(machdr->type)
    xpt = ptype_base[h]
    if(xpt == null)
        drop(skb)
        return -1
    else
        pt = pt(xpt)
        if(pt->type == machdr->type)
            gap_ip_rcv(skb)    //call pt->func
            return 0
        else
            hash_collision_process()
        endif
    endif
end
end

```

새로 추가한 규약의 수신처리 `gap_ip_rcv()`에서 유효자료의 보안을 위하여 진행한 암호화에 대한 복호화처리과정은 다음과 같다.

```
gap_ip_rcv(skb)
    payload, sign, ← extract(skb)
    if DecKs(sign) = H(payload || sign)
        skb = DecK(payload)
        return skb
    endif
end
```

여기서  $K$ 는 대칭암호화열쇠,  $K_s$ 는 서명확인열쇠이다.

새로 개발하여 추가한 망층규약은 핵심부에서 실행되기때문에 문맥절환이 적고 실행 성능이 높으며 망장치를 설정하여 지정된 망기관들사이 고정통신을 실현할수 있다.

## 맺 는 말

핵심부모듈에 의한 망층규약확장방법은 성능높은 망규약과 망기능을 우리 식 조작체계에 유연하게 추가할수 있으므로 망층규약확장과 보안기능추가에 효과적으로 리용될수 있다.

## 참 고 문 헌

- [1] William Stallings; Computer Security: Principles and Practices, Prentice Hall, 1~788, 2018.
- [2] Andrew Ginter; Proceedings of ICALEPCS, 1412, 2013.

주체109(2020)년 8월 5일 원고접수

## A Way to Add New Network Layer Protocol Using the Ptype\_base Hash Table in Linux Kernel

*Kim Won Chol, Pak Un Song*

This paper decodes the receiving process in the network interface layer and proposes a way to add new network layer protocol using the ptype\_base hash table in Linux kernel. This way can implement the private communication with low cost.

Keywords: Linux kernel, network layer protocol