Fall '19 COSE322-00

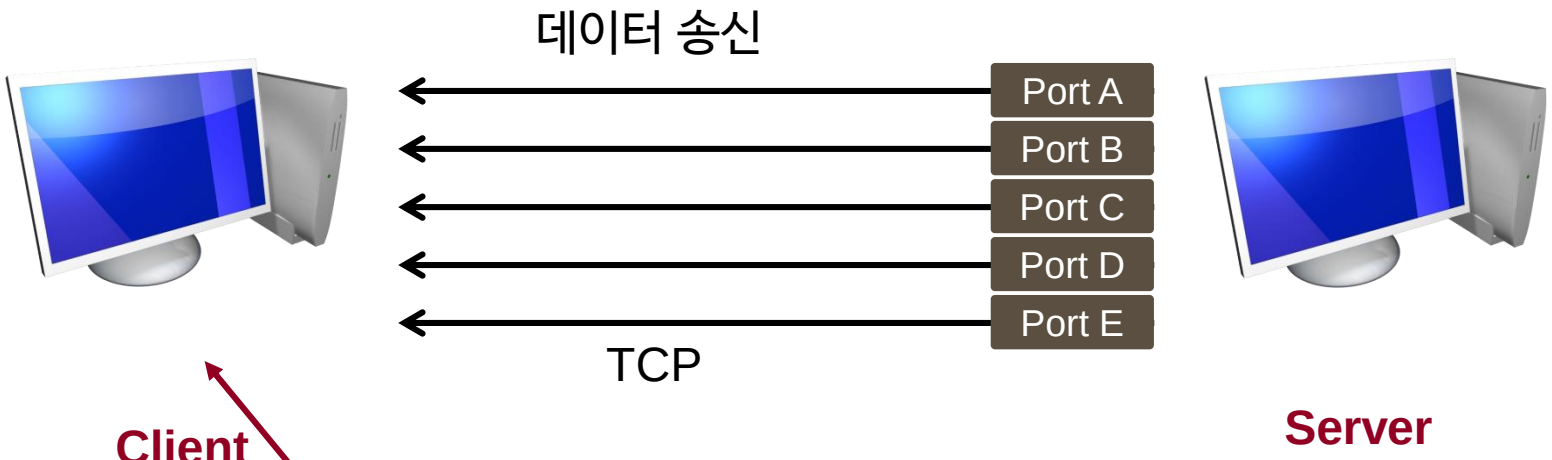
System Programming

Practice 07. [2차과제] Packet Forwarding

2019. 11. 21.

Contents

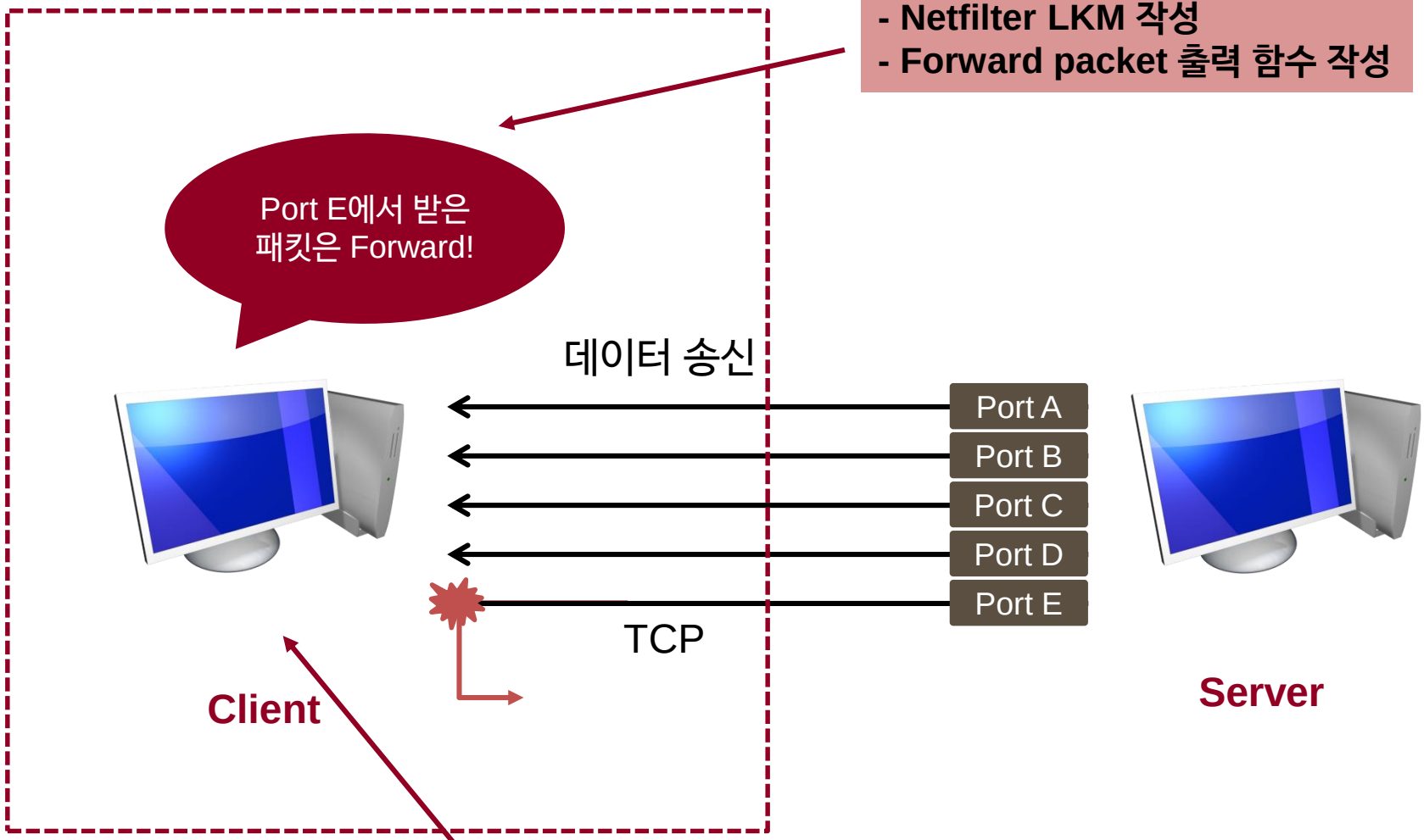
과제개요



패킷 수신하는 Application은
과제2(warmup)에서 작성한 코드를 사용

과제개요

- Netfilter LKM 작성
- Forward packet 출력 함수 작성



패킷 수신하는 Application은
과제2(warmup)에서 작성한 코드를 사용

과제개요

- **2차 과제 : Netfilter를 이용한 Packet forwarding 모듈 구현 및 커널 네트워킹 분석**
- **목표**
  - Packet forwarding을 위한 Netfilter 등록 **커널 모듈 작성**
    - Hook point와 콜백함수 등록에 대한 메커니즘을 이해한다
    - Netfilter의 동작 메커니즘을 이해한다.
    - IP Layer에서의 packet forward 동작을 이해한다
  - Packet filtering **수행 및 결과 분석**
    - 특정 포트에서 패킷이 수신되고, 해당 패킷을 forwarding하여, packet forwading이 실행됨을 보인다
  - IP Layer 네트워크 **구현 분석**
    - Ip_rcv 함수부터 ip_output함수까지 분석
    - 과제 수행시 커널 내에서 패킷이 포워딩되는 루틴을 코드 레벨에서 분석

Callback 함수

❖ 일반 함수호출과 Callback 함수 호출의 차이

- 일반 함수호출 : 내가 함수를 호출하는 것
- Callback 함수호출 : 내 함수가 호출되는 것

❖ Callback 함수

- 일반적으로 운영체제가 특정한 이벤트 발생 시 호출할 함수를 지정하도록 할 수 있다.
- 이 때 호출되는 함수를 Callback 함수라고 한다.
- Callback 함수가 실행되고 나면 컨텍스트의 제어권한은 원래의 프로시저로 돌아간다.

Hooking

❖ 후킹

- 소프트웨어 내에서 함수호출, 메시지, 이벤트 등을 중간에 바꾸거나 가로채는 방법
- 간섭된 함수 호출, 이벤트, 메시지와 관련된 코드를 hook라고 한다.

❖ Hooking point

- 후킹을 할 수 있도록 사전에 정의된 지점
- 우리가 사용하려는 넷필터도 일종의 후킹 포인트임

마감기한 및 제출방법

❖ **마감기한 : 2019년 12월 19일 (목) 23:59**

❖ **제출방법**

- 블랙보드 제출 : 소스코드, 결과파일, 보고서파일
  - 압축파일 이름 : sp2_Group번호
  - 보고서는 pdf 파일로 제출
- Hard Copy 제출 : 보고서
  - 운영체제연구실 (우정정보관 308호) 앞의 박스에 제출

❖ **Freeday 사용여부 및 사용기간을 반드시 보고서 상단에 표기**

- Freeday 사용하지 않는 경우에도 [0일 사용] 표기
- 이름, 학번, 제출일자, Freeday 사용 일수 기재가 채점 항목에 포함되어 있음

주요 진행 과정

❖ 진행과정

- Packet forwarding 을 위한 netfilter 커널 모듈 작성
- Packet forwarding 수행 및 결과 분석
- IP Layer 네트워크 구현 분석

❖ 유의사항

- server-side 프로그램은 2차 Warmup 과제에서 블랙보드를 통해 제공한 VM 이미지 파일 사용

진행과정

❖ Packet forwarding을 위한 커널 모듈 작성

- 커널 상에서 동작하는 kernel module을 작성한다.
- 모듈 초기화
  - 넷필터의 후킹 포인트 중 한 곳에 패킷 Forward 함수를 등록
    - 이 때 우선순위는 **가장 높게** 설정한다.
    - 왜 해당 후킹 포인트를 사용하는지 보고서에 기재
  - 넷필터의 후킹 포인트 중 '다른 두 곳'에 Forward한 패킷을 확인하는 함수를 등록
    - 출력 함수는 자유롭게 사용하여 로그파일로 기록
- 패킷 Forward 함수
  - 특정 포트에서 들어온 패킷을 forward하는 역할을 수행한다.

진행과정

❖ 패킷 포워딩 실험 수행 및 분석

- Routing table 정보를 추가
  - Routing table에 add명령어를 통해 포워딩을 위한 규칙 추가
- 들어온 패킷의 정보를 (Protocol;Sport;Dport;SIP;DIP)로 출력
  - 해당 정보 앞에 **PRE_ROUTING** packet 문자열을 추가하여 포워딩 패킷과 구분해준다.
  - 서버의 33333 포트에서 온 패킷을 Forwarding 대상으로 지정한다.
  - 출력 후 Forwarding 대상이 되는 패킷의 Sport, Dport를 7777로 바꾼다.
- 포워딩 패킷의 정보를 마찬가지로 출력
  - 해당 정보 앞에 **FORWARD** packet, **POST_ROUTING** packet 문자열을 추가하여 구분해준다.
- 이 과정에서 NF_INET_PRE_ROUTING, NF_INET_FORWARD, NF_INET_POST_ROUTING 에서 hooking 되었는지 확인한다.
- 패킷 헤더의 어떤 정보를 바꿨고 왜 바꿨는지 ‘반드시’ 서술

진행과정

❖ **IP Layer 네트워크 구현 분석**

- ip_rcv() ~ ip_output() 까지 포워딩 함수 루틴을 분석한다.
- 해당 내용은 별도의 구현이 아닌 **보고서에 작성**한다.

Packet Forwarding 네트워크 구현 분석 (1/2)

- 리눅스 4.4 버전을 기준으로 함 (LXR 활용 가능)
- ip_rcv()
  - 시작지점 (net/ipv4/ip_input.c)

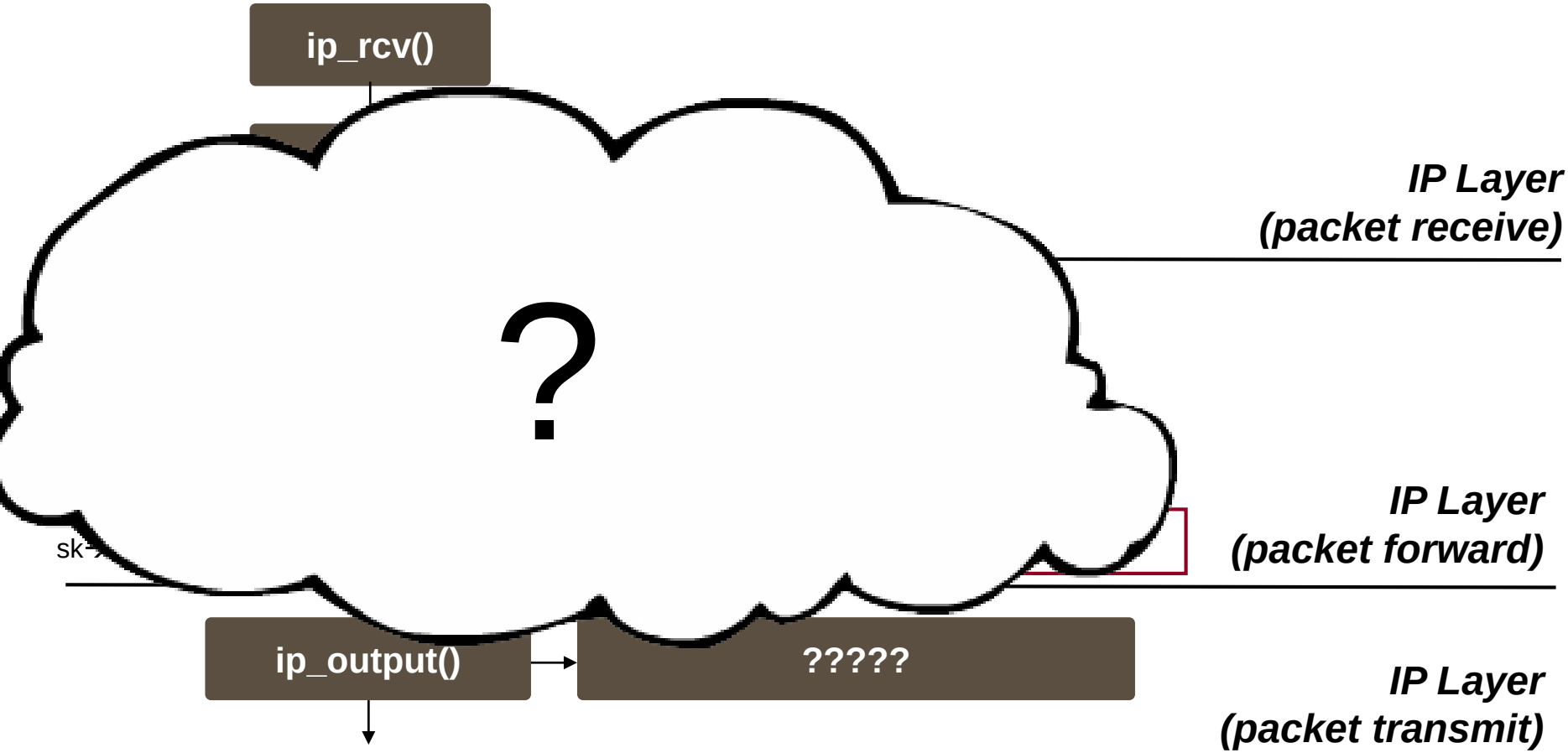
```
int ip_rcv(struct sk_buff *skb, struct net_device *dev,
struct packet_type *pt, struct net_device *orig_dev)
```

  - 종료지점 (net/ipv4/ip_output.c)

```
int ip_output(struct net *net, struct sock *sk, struct
sk_buff *skb)
```

Packet Forwarding

❖ Function calls in Kernel layer



채점기준 (80점 기준)

- **소스코드 (15점)**
  - LKM(Loadable Kernel Module) 소스파일, Makefile, .ko 파일 (15점)
- **결과파일 (5점)**
  - Application이 생성한 수신 패킷에 대한 결과 로그파일 (5점)

채점기준

❖ **보고서 (60점)**

- 조원 이름, 학번, 제출일자, Freeday 사용 일수 기재 (5점)
- Netfilter 및 Hooking에 대한 설명 (15점)
- 커널레벨 네트워킹 코드 분석 (10점)
- 작성한 소스코드에 대한 설명 (15점)
  - 패킷 헤더의 어떤 정보를 바꿨고 왜 바꿨는지 '반드시' 서술
- 실험 방법에 대한 설명 및 로그파일 결과 분석 (10점)
- 과제 수행 시의 Trouble과 Troubleshooting 과정 (5점)

비고

❖ **프리데이를 초과하여 과제를 늦게 제출한 경우**

- 지각제출 1일당 과목 전체 점수에서 1점씩 감점됨

❖ **2차과제 마감일 1주일 이후부터는 과제제출이 불가함**

Appendix

Routing Table

❖ 개념

- 라우팅 서브시스템의 핵심
- 인입 패킷이 로컬 호스트로 갈지 다른 곳으로 포워딩 돼야할지 결정 필요
- 다른 곳으로 포워딩돼야 하는 경우 다른 곳으로 정확히 전달하기 위한 정보 필요
- 이 정보를 담고 있는 것이 ‘라우팅 테이블’
- 잦은 사용 -> 캐시 (지난 실습 슬라이드에 언급)

Routing Table

❖ 사용법

– Ubuntu(Linux)기준으로 ‘route’ 명령어 입력 시 라우팅 테이블 출력

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
default	10.0.2.2	0.0.0.0	UG	100	0	0	enp0s3
10.0.2.0	*	255.255.255.0	U	100	0	0	enp0s3
100.1.1.0	*	255.255.255.0	U	100	0	0	enp0s8
link-local	*	255.255.0.0	U	1000	0	0	enp0s8

- Destination: 목적지 주소
- Gateway: 게이트웨이 주소(외부 네트워크 연결 목적)
- Genmask: 목적지 주소의 넷마스크
- Flags: 해당 경로 정보. (ex. U는 경로가 살아있음, G는 Gateway 경로)
- Metric: 목적지까지 거리
- Ref: 경로 참조 횟수
- Use 경로 탐색 횟수
- Iface: 해당 목적지 주소를 가진 패킷이 사용할 Interface

Routing Table

❖ 사용법

- 라우팅 테이블의 엔트리는 인터페이스에 할당한 IP에 대해선 자동으로 잡힘
- 라우팅 테이블 엔트리를 임의로 추가, 삭제할 수 있음.
  - Route add, del –net 명령어 이용
  - 구글링을 통해 다양한 예시 확인 가능
- 예시 (route add)

```
root@oslab-VirtualBox:/home/oslab# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         10.0.2.2        0.0.0.0         UG    100    0        0 enp0s3
10.0.2.0        *               255.255.255.0   U     100    0        0 enp0s3
link-local      *               255.255.0.0     U     1000   0        0 enp0s8
```

  - route add -net 100.1.1.0 netmask 255.255.255.0 dev enp0s8
  - route add -net 111.1.1.0 netmask 255.255.255.0 dev enp0s8

```
root@oslab-VirtualBox:/home/oslab# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         10.0.2.2        0.0.0.0         UG    100    0        0 enp0s3
10.0.2.0        *               255.255.255.0   U     100    0        0 enp0s3
100.1.1.0       *               255.255.255.0   U     0      0        0 enp0s8
111.1.1.0       *               255.255.255.0   U     0      0        0 enp0s8
link-local      *               255.255.0.0     U     1000   0        0 enp0s8
```

Q&A