Packet Flows and Networking Functions

이동무선네트워크 연구실
홍 석 준

Link-Layer Packet Flows

- 링크 계층에서의 패킷의 흐름은 두 가지 경로가 있음.
- 프레임을 물리 계층에서 받으면 그것을 네트워크 계층으로 전달하고
- 전송시에는 네트워크 계층에서 받으면 물리 계층으로 전달한다.

Link-Layer Networking Functions

계층	함수	설명	디렉토리	파일
Data Link	net_rx_action() ->netif_receive_skb()	수신 인터럽트 호출에 의해 커널이 호출.	net/core	dev.c
Data Link	net_tx_action() ->dev_queue_xmit()	송신 인터럽트 호출에 의해 커널이 호출.	net/core	dev.c

Link-Layer Packet Flows(Rx)

- Network Interface에서 프레임을 받으면, 인터럽트 핸들러가 skbuff구조체에 dev_alloc_skb()을 통해 메모리로 할당하고, 구조체로 복사한다.
- net_rx_action()함수는 들어오는 프레임에 대한 처리를 위해 사용된다. 이 함수는 Interrupt와 polling방식을 혼합한 방식을 사용한다.
- 이 방식을 사용해서 커널은 전자의 프레임을 다루는 동안에 다른 들어오는 새로운 프레임을 인터럽트 없이 계속 핸들링할 수 있다.

Link-Layer Packet Flows (Rx)

- net_rx_action()함수는 poll()이라는 가상함수를 각 device에서 들어온 queue에서 dequeue하기위해 호출함.
- poll() 가상 함수에서는 프레임을 처리하기 위해 netif_receive_skb()함수를 호출함.
  - 여기서 가상 함수(virtual function)란 하나의 디바이스에서 특정 polling함수를 호출하는 일반적인 함수(c++의 virtual 함수가 아님)
  - 현재 커널 소스코드(linux-4.15.8버전)에서는 process_backlog 함수가 poll()함수를 위해서 사용되었음

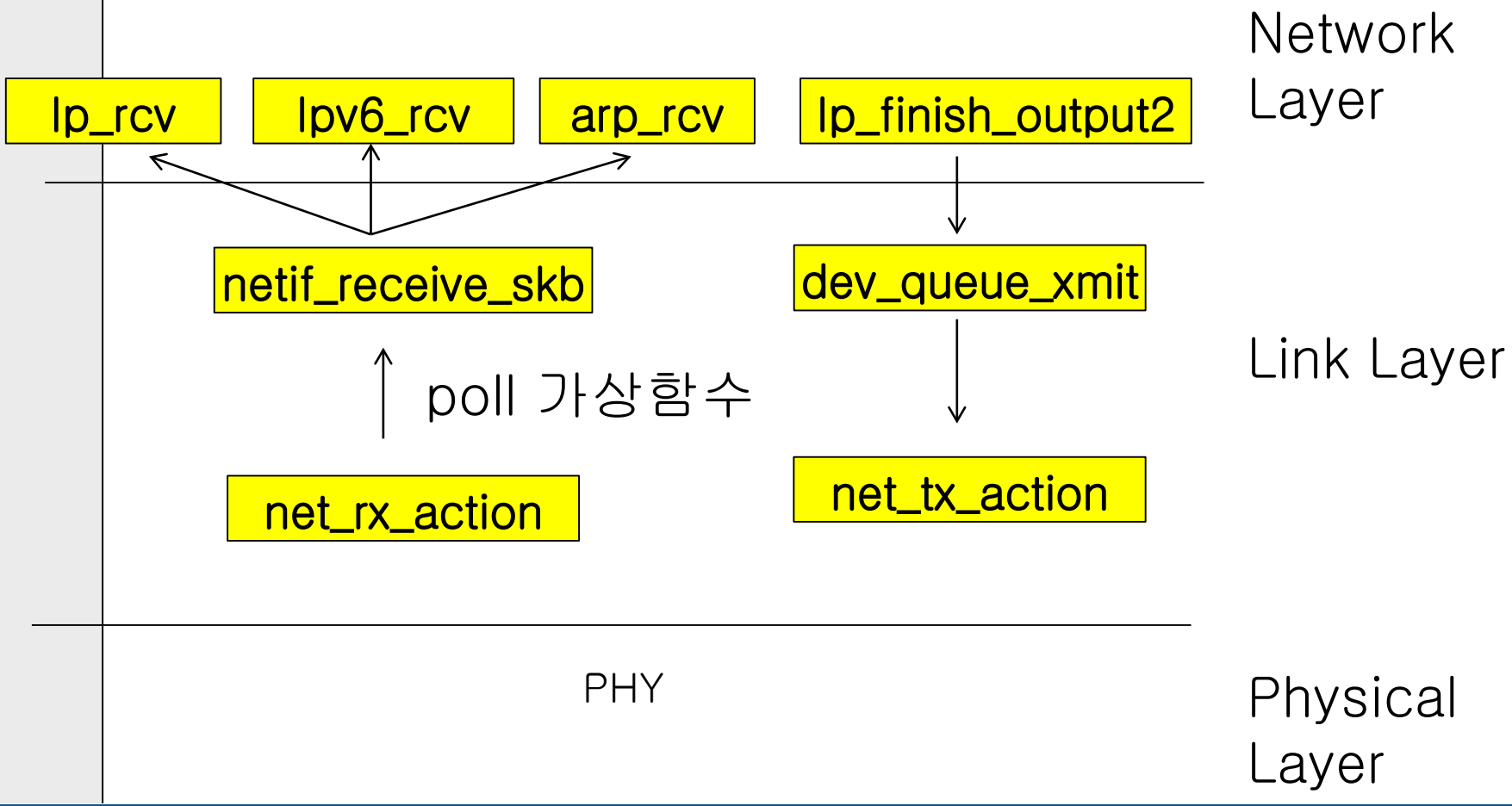
Link-Layer Packet Flows (Rx)

- net_rx_action()함수가 호출될때 L3 protocol type이 이미 인터럽트 핸들러에 의해 sk_buff에 의해 셋팅된다.
- 그러므로 netif_receive_skb()함수는 L3 protocol type을 알고 있고 그것을 protocol field와 연관시킴.
- 이것의 함수 포인터는 IPv4, IPv6, 그리고 ARP를 다루는 ip_rcv(), ip_ipv6(), arp_rcv() 함수를 가리킨다.

Link-Layer Packet Flows(Tx)

- 전송하는 패킷의 흐름은 수신하는 경로와 대칭적임.
- net_rx_action()과 대칭적으로 net_tx_action()함수가 있음.
- net_tx_action()함수는 두 가지 일을 수행.
  1. 전송되어지기 원하는 프레임이 실제로 전송되어졌는지를 보장하는 것.
  2. 전송이 끝난 이후에 sk_buff 구조체에 대해서 자원할당을 해제하는 것.

Link-Layer Packet Flows in Call Graphs



IP(Network)-Layer Packet Flows

- IP layer는 Link layer위에 그리고 Tranport Layer아래에 위치하고 있음.
- 따라서 계층적인 관점에서 인터페이스는 두 개의 인접한 레이어에 대해서 제공되어져야함.
- 이 계층에서는 수신된 패킷을 TCP, UDP, 그리고 raw IP 소켓 인터페이스를 포함하는 Transport 계층에 전달되어진다.
- 그리고 패킷을 전송하기 위해서는 반대의 과정이 이루어짐.

IP-Layer Networking Functions

계층	함수	설명	디렉토리	파일
Network	Ip_rcv(), ip_rcv_finish()	패킷을 수신하고, 라우팅 정보에 따라 패킷을 전달함.	net/ipv4/	ip_input.c
Network	Ip_local_deliver(), ip_finish_output()	패킷을 분할,재조립하여 전송함.	net/ipv4/	Ip_output.c

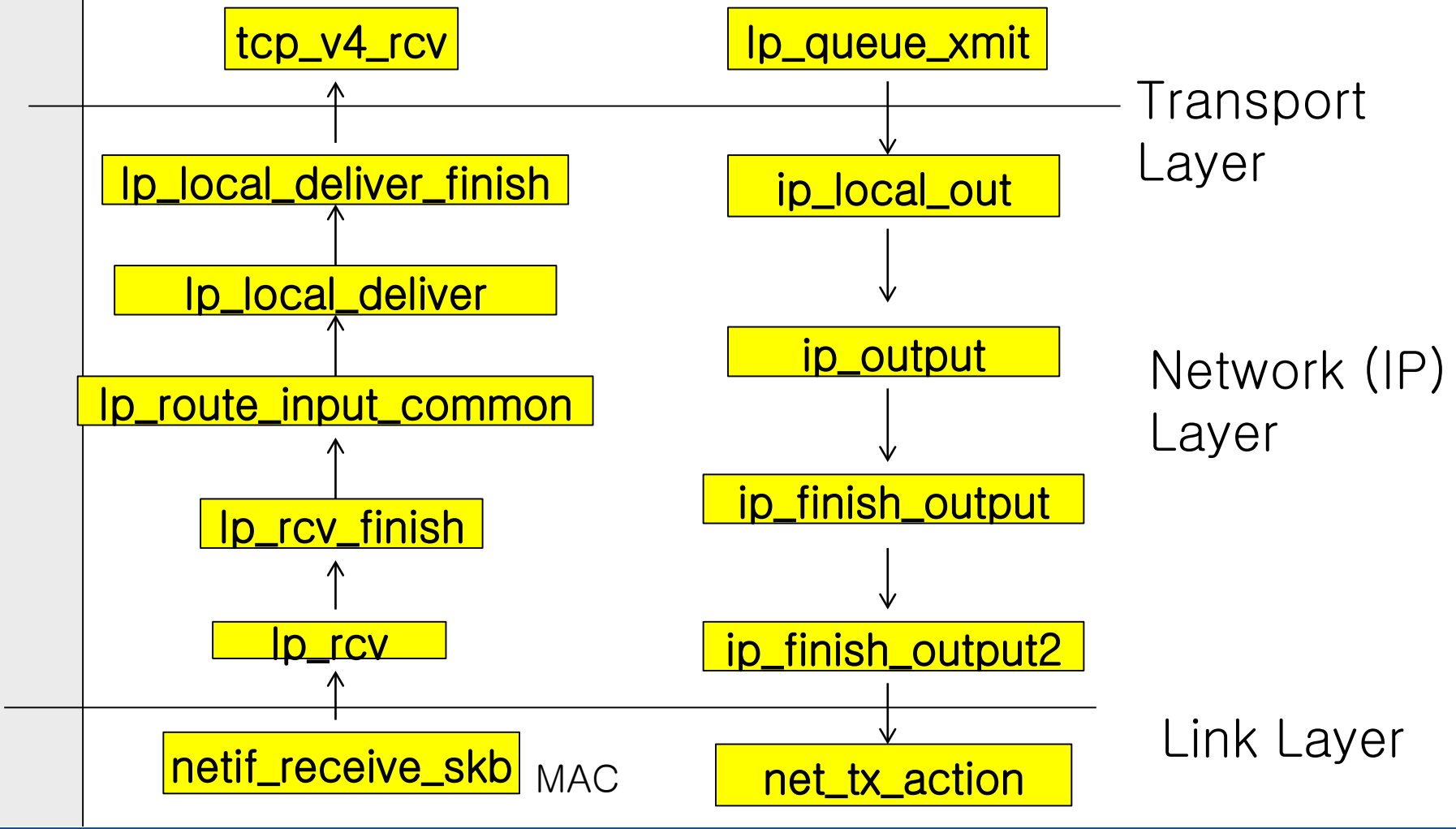
IP(Network)-Layer Packet Flows(Rx)

- sk_buff에 있는 Network-layer protocol type이 IP 프로토콜인 경우, ip_rcv()가 프로토콜 핸들러에 의해 호출됨.
- 만약, 이 패킷이 local_host를 위한 패킷인 경우 ip_local_deliver()가 호출되고, 그것은 패킷을 transport layer로 전송하기 위해 ip_local_deliver_finish()를 다시 호출함.
- 그리고 또한 상위 계층이 raw IP socket interface, UDP, 혹은 TCP이냐에 따라서 각각 raw_v4_input(), udp_v4_input(), 혹은 tcp_v4_rcv()로 정해짐.

IP(Network)-Layer Packet Flows(Tx)

- 전송 경로에 있어서는 상위 프로토콜은 IP 계층으로 보내기 위해서 패킷을 큐에 넣는다.
- ip_append_data(), ip_append_page(), or ip_queue_xmit() 들이 각각의 프로토콜에 따라 호출됨.(그림 참조)
- 그리고 이 모든 함수들은 dst_output()함수를 호출하고 그것들은 부분적으로 sk_buff에 등록된 skb->dst->output()이라는 함수를 호출한다.
- 그리고 네트워크 프로토콜이 IP프로토콜인 경우, 네트워크 계층의 핸들러인 ip_output()함수를 호출한다.
- 그리고 더 이상 분할이 필요하지 않은 경우 ip_finish_output2()가 호출되어서 link layer의 net_tx_action에 전달되어진다.

IP-Layer Packet Flows in Call Graphs



Transport-Layer Packet Flows

- IP 계층과 Application 계층에 대한 두 개의 인터페이스를 제공해야함.
- 수신 경로에 있어서는 하나의 패킷이 IP 계층에서 받아지면 응용 계층으로 전달되어지고,
- 송신 경로에 있어서는 하나의 패킷이 응용 계층으로부터 도착하면 IP 계층으로 전송되어짐.
- 거의 모든 패킷의 처리 흐름을 통해 포함되는 데이터 구조는 sk_buff와 sock 2가지가 있음.
- sk_buff는 앞장에서 설명하였고, sock 구조는 TCP를 수행하기 위한 대부분의 필수적인 변수를 가지고 있는 tcp_sock을 포함하고 있음.

Transport-Layer Packet Flows

계층	함수	설명	디렉토리	파일
Transport	Tcp_v4_rcv()	Tcp로 보내기 위해 Ip계층에서 호출함.	/net/ipv4	tcp_ipv4.c
Transport	Ip_queue_xmit()	sk_write_queue로부터 Ip계층으로 데이터를 전달하기 위해 호출함.	/net/ipv4	Ip_output.c

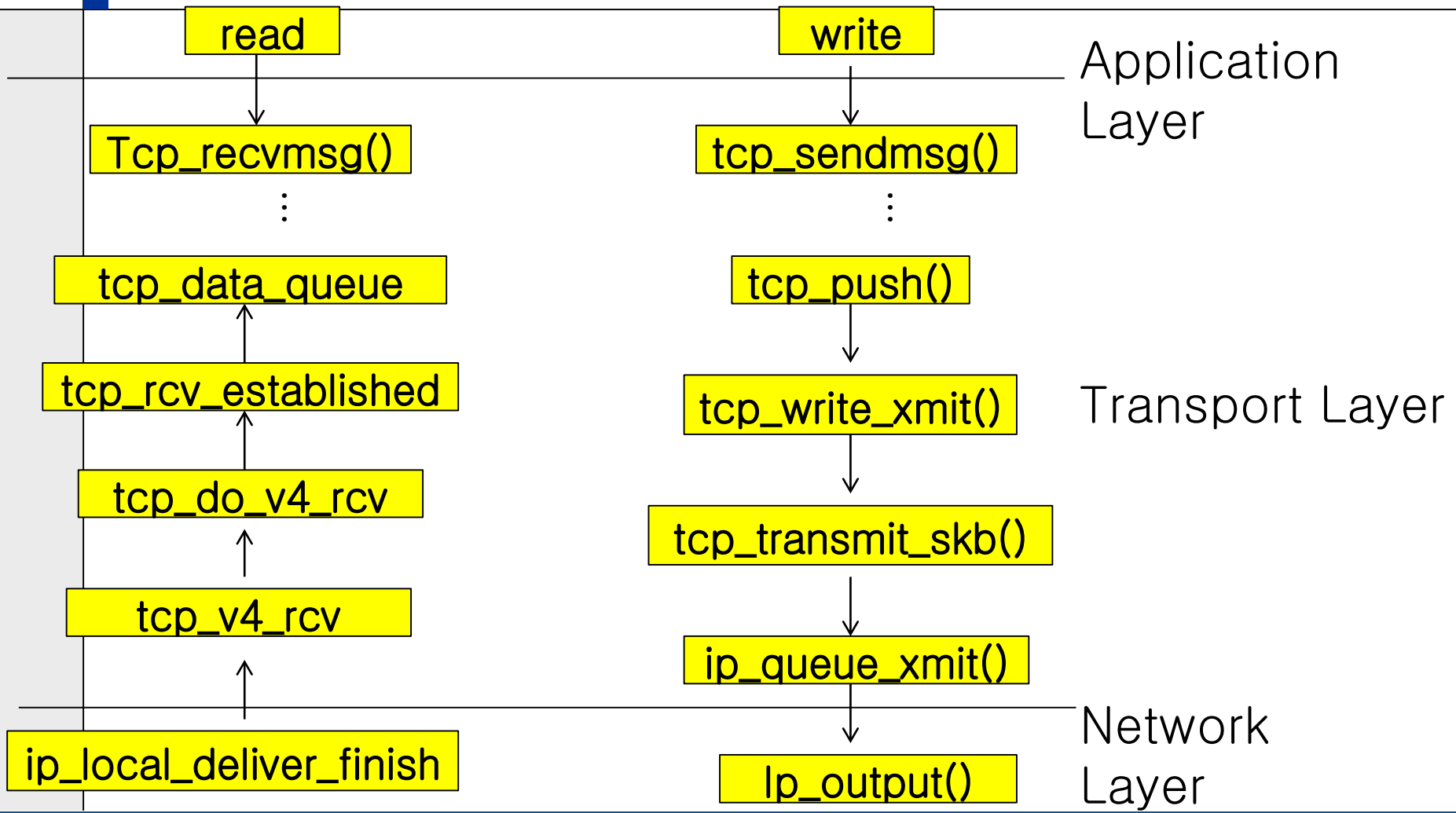
Transport-Layer Packet Flows(Rx)

- Raw_v4_input(), udp_rcv(), or tcp_v4_rcv() 세가지로 분류되어서 전송됨. (그림 참조)
- 이후 각각의 프로토콜은 패킷과 연관된 소켓 구조를 가져오기 위한 lookup함수를 가짐.
- Sock 구조체에 있는 정보로부터 전송 계층을 구별할 수 있음.
- 그런다음 수신된 패킷은 skb_queue_tail()함수를 호출에 의해 큐에 넣어짐.
- sk->sk_data_ready()함수에 의해서 이 흐름이 속해있는 응용은 수신을 위해 데이터를 사용가능한 것을 공지받음.
- 이후 응용은 read() 혹은 recvfrom() 을 통해서 sock 구조로부터 데이터를 가져올 수 있음.

Transport-Layer Packet Flows(Tx)

- 한 응용이 인터넷에 패킷을 전송할 것을 계획하면, 그것은 write()이나 sendto()를 호출하고 그것은 raw_sendmsg(), udp_sendmsg(), tcp_sendmsg()를 소켓이 생성된 프로토콜에 기초해서 호출하게 됨.
- 마침내 skb가 sk_write_queue에 넣어진다.
- TCP 소켓을 위해서는 tcp_sendmsg()와 skb_add_data()가 큐에서 skb를 제거하고 커널 공간의 메모리로 복사하기 위해 위해 사용된다.
- 마침내 sk_write_queue로부터 ip_queue_xmit()가 호출되어 ip_output()함수를 통해 IP Layer로 데이터를 전송된다.

Transport-Layer Packet Flows in Call Graphs



Reference

- **Computer Networks: An Open Source Approach**
  - **Ying-Dar Lin, Ren-Hung Hwang, Fred Baker**