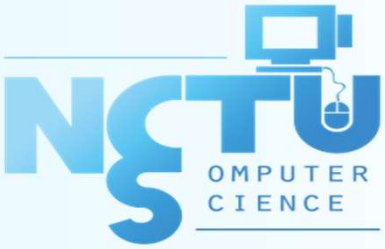


iptables and netfilter

Prof. Chien-Chao Tseng

曾 建 超 教授

Department of Computer Science
National Yang Ming Chiao Tung University

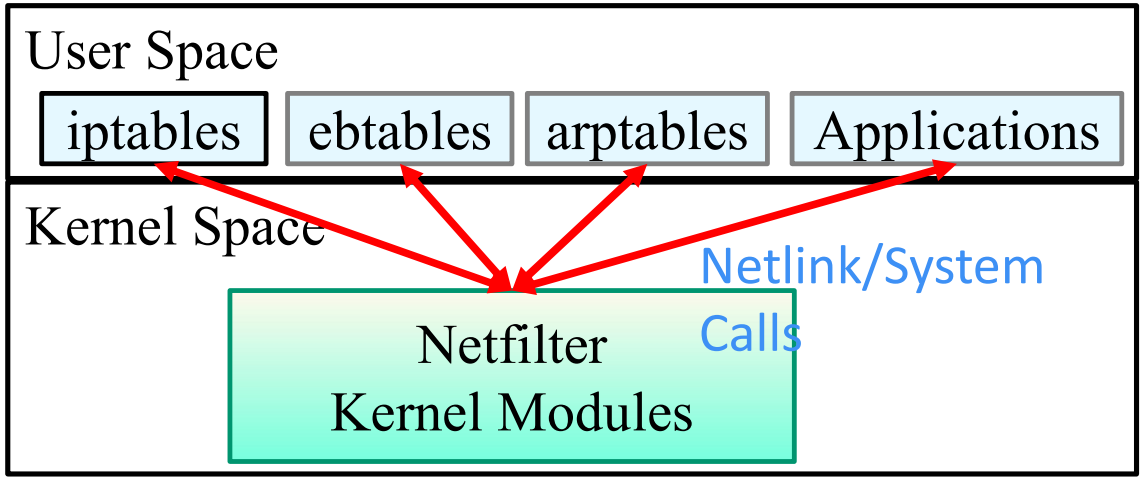
cctseng@cs.nctu.edu.tw

References: https://www.comparitech.com/blog/vpn-privacy/ipsec-vs-ssl-vpn/

Iptables Overview

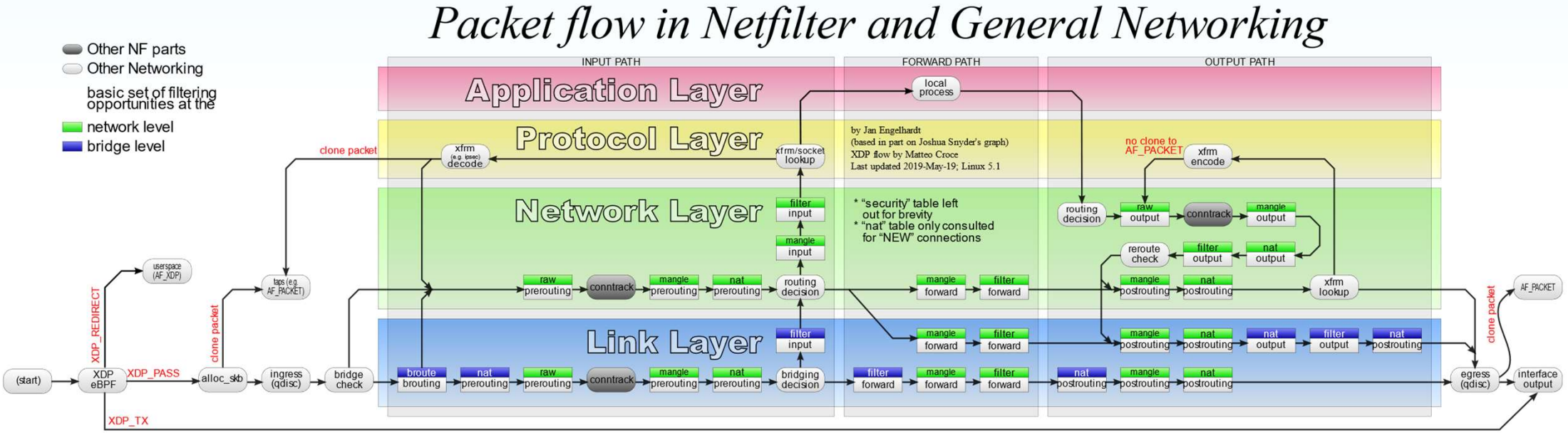
- A user-space utility program that allows a system administrator to configure the **tables** provided by the Linux kernel firewall, and the **chains** and **rules** it stores.
  - Linux kernel firewall implemented as different **Netfilter modules**
- **netfilter**

  a framework inside the Linux kernel that allows kernel modules to register **callback functions** at different locations (hooks) of the Linux network stack.
  - A **registered callback function** is called back for every packet that traverses the respective hook within the Linux network stack.



- ebtables: Ethernet Bridge Tables

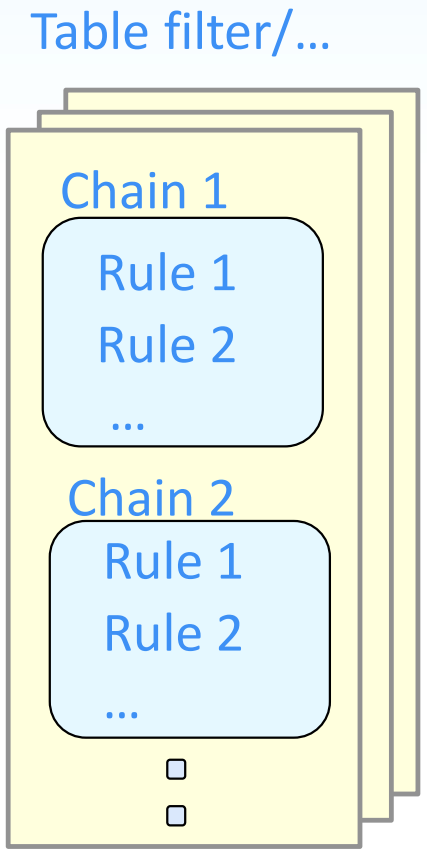
Netfilter Packet Flow



By Jan Engelhardt
File:Netfilter-packet-flow.svg - Wikimedia Commons

Components of iptables

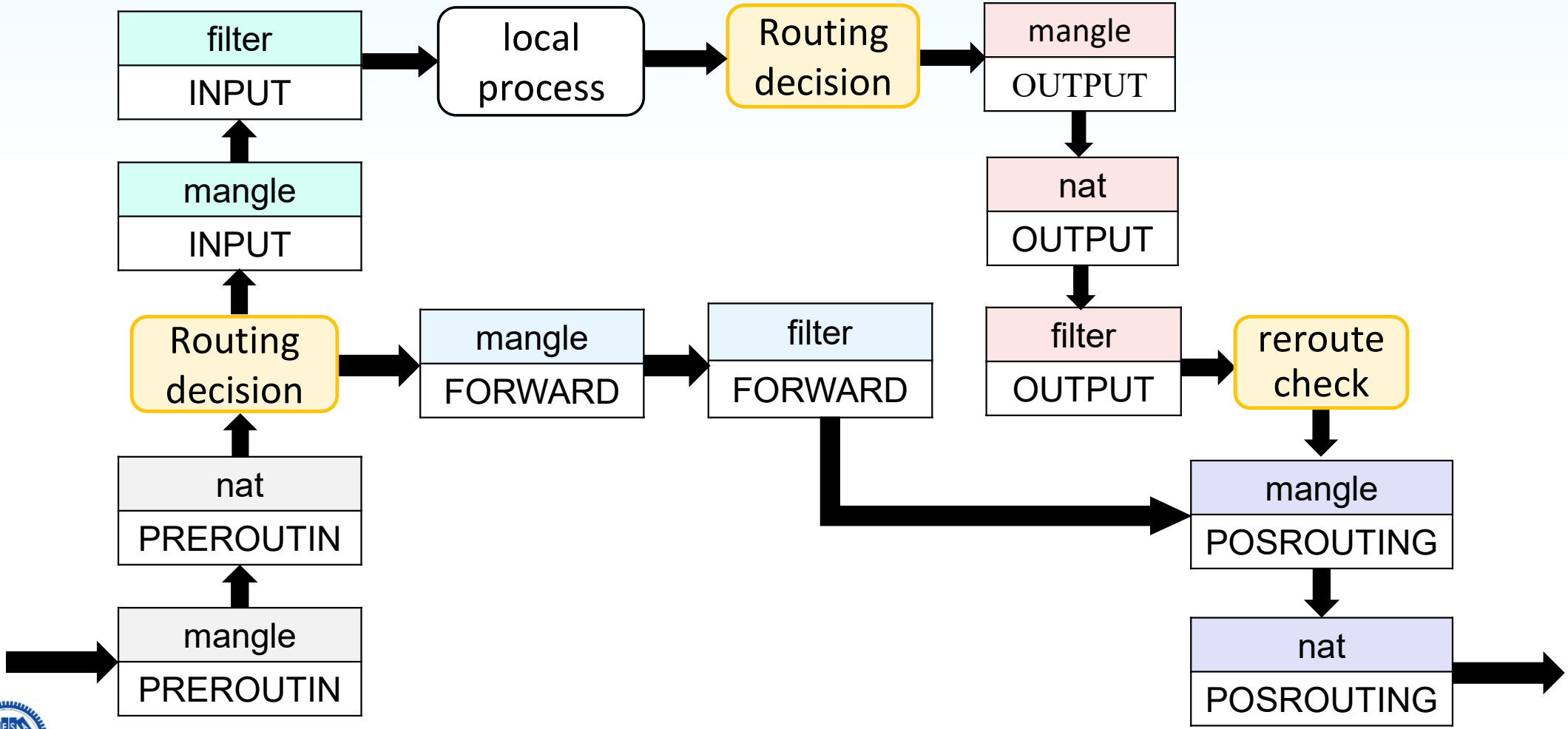
- **Tables:** files that join similar actions.
  - Contains a number of built-in chains or user-defined chains.
- **Chains:** a list of **rules** which can match a set of packets
  - When receives a packet, iptables finds the appropriate table;
  - Then apply the chain of **rules** on the packet until it finds a match.
- **Rules:** specifies what to do with a packet that matches.
  can block one type of packet, or
  forward another type of packet.
- **Targets:** a decision of what to do with a packet.
  - Typically, Accept, Drop, or Reject (which sends an error back to the sender)



Tables and Chains

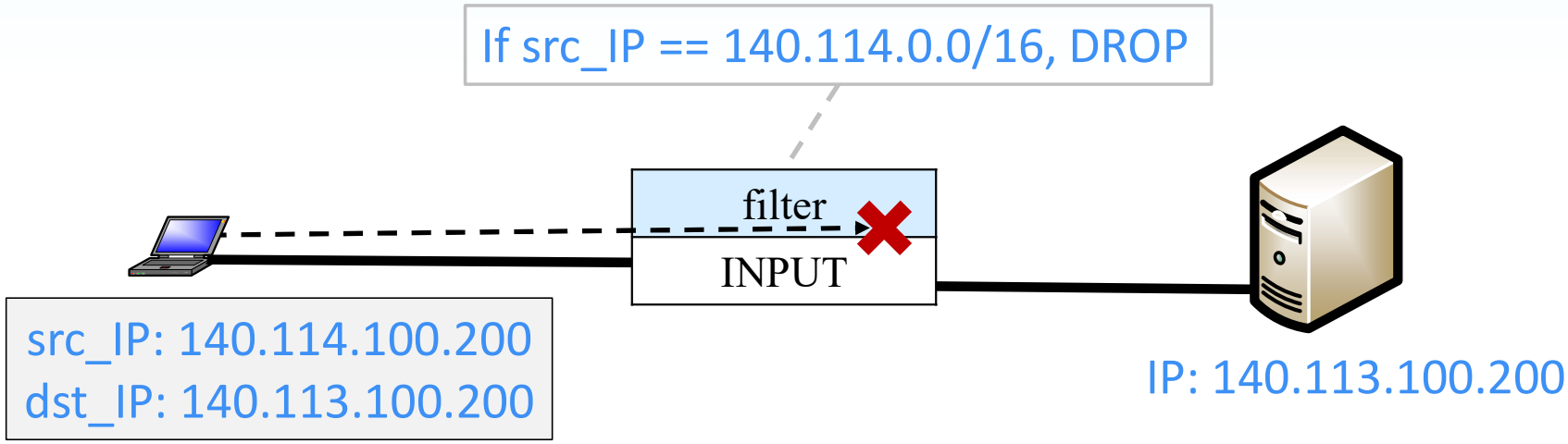
- Tables
  - filter: packet filtering, default table
  - nat: NAT operation
  - mangle: add tag on packet (for QoS or load distribution)
  - raw: mainly for exemptions from connection tracking
- **Five Predefined Chains** (mapping to the five available Netfilter hooks)
  - **PREROUTING**: for packets before a routing decision is made.
  - **INPUT**: for packets destined to local sockets
  - **FORWARD**: for packets being routed through the machine.
  - **OUTPUT**: for locally-generated packets.
  - **POSTROUTING**: for packets about to go out after Routing decision has been made.

Netfilter Network Layer Packet Flow



iptables Example

- `iptables -A INPUT -t filter -s 140.114.0.0/16 -j DROP`



Source NAT – MASQUERADE vs SNAT Rules

- Both MASQERADE and SNAT rules can perform source NAT
  - iptables nat MASQUERADE rule

```
iptables -t nat -A POSTROUTING -o eth2 -s 10.0.0.0/24 -j MASQUERADE
```

  - iptables nat SNAT rule

```
iptables -t nat -A POSTROUTING -s 10.0.0.0/24 -o eth2 -j SNAT --to-source 192.168.1.2
```

- **Differences**
- MASQUERADE does **NOT** require **--to-source** because it works with **dynamically** assigned IPs
- SNAT requires **--to-source** because it works **ONLY** with **static IPs**, that's why it
- MASQUERADE is slower than SNAT
  - Each time MASQUERADE rule gets hit by a packet, it has to check for the IP address to use.

Source: https://linuxhacks.org/what-is-ip-masquerade-and-how-to-rule-it-with-iptables/

- Hooking

  a programming technique for monitoring software behavior or extending functionality without altering the original code.

  – by intercepting function calls or messages or events passed between software components

  – Use them to trigger your own code.
- Linux Network hooks

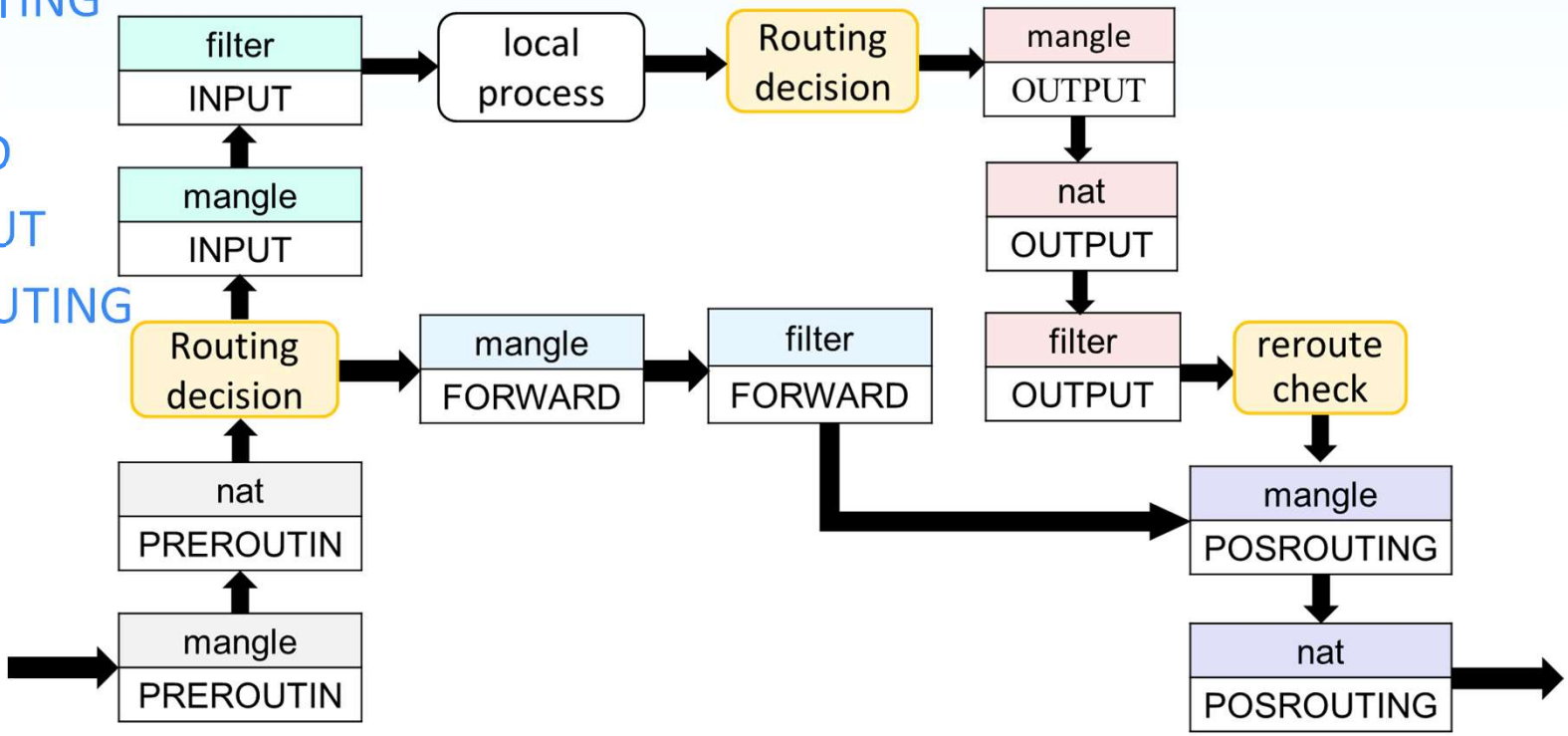
  – Some well-defined points in protocol stack
- Software modules can register callback functions for a specific hook

  – When protocol stack process a packet at the hooked point

  – The hook will trigger (call) the registered callback functions

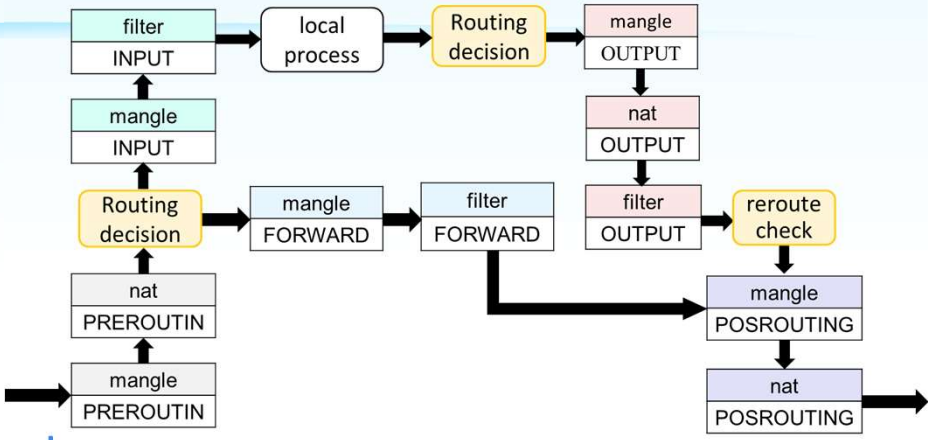
netfilter Hooks

- Five netfilter hooks for kernel module to register
  - NF_IP_PRE_ROUTING
  - NF_IP_LOCAL_IN
  - NF_IP_FORWARD
  - NF_IP_LOCAL_OUT
  - NF_IP_POST_ROUTING



Relation between iptables and netfilter

- Chains: represent sequences of executing rules
- Hooks: triggers at a specific well-defined points

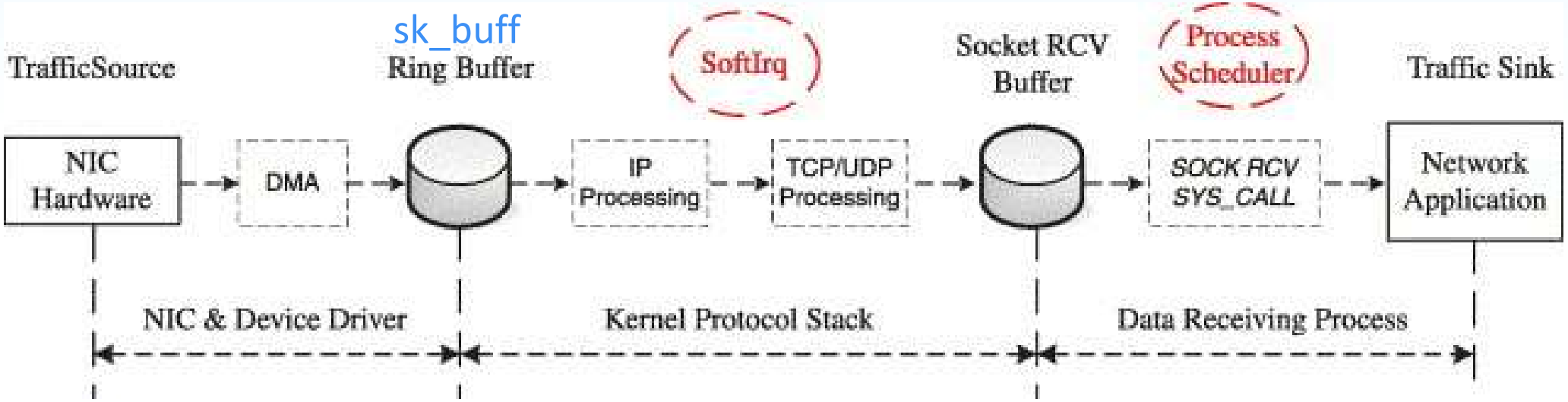


- Mapping between iptables chains and netfilter hooks

iptables chain	netfilter hook
PREROUTING	NF_IP_PRE_ROUTING
INPUT	NF_IP_LOCAL_IN
FORWARD	NF_IP_FORWARD
OUTPUT	NF_IP_LOCAL_OUT
POSTROUTING	NF_IP_POST_ROUTING

Linux Packet Receiving Process

- Process of incoming packet in Linux Networking Subsystem



- **sk_buff**: a common data structure in Linux kernel for all network-related queues and buffers
  - A large struct containing all the control information required for the packets (datagram, cell, whatever).

Source: Wenji Wu, Matt Crawford and Mark Bowden. "The performance analysis of linux networking – Packet receiving," Computer Communications, Elsevier, Volume 30, Issue 5, March 2007, Pages 1044-1057

https://www.sciencedirect.com/science/article/pii/S0140366406004221