Introduction to Computer Networks

Linux Networking Stack & Infrastructure Services

CS640

https://pages.cs.wisc.edu/~mgliu/CS640/S26/index.html

Ming Liu
mgliu@cs.wisc.edu

Outline

- Last
  - In-Network Support for TCP
- Today
  - Linux Networking Stack
  - Infrastructure Services
- Announcements
  - Quiz 4 in-class on Thursday (04/23/2026)
  - Lab 4 due date 04/30/2026

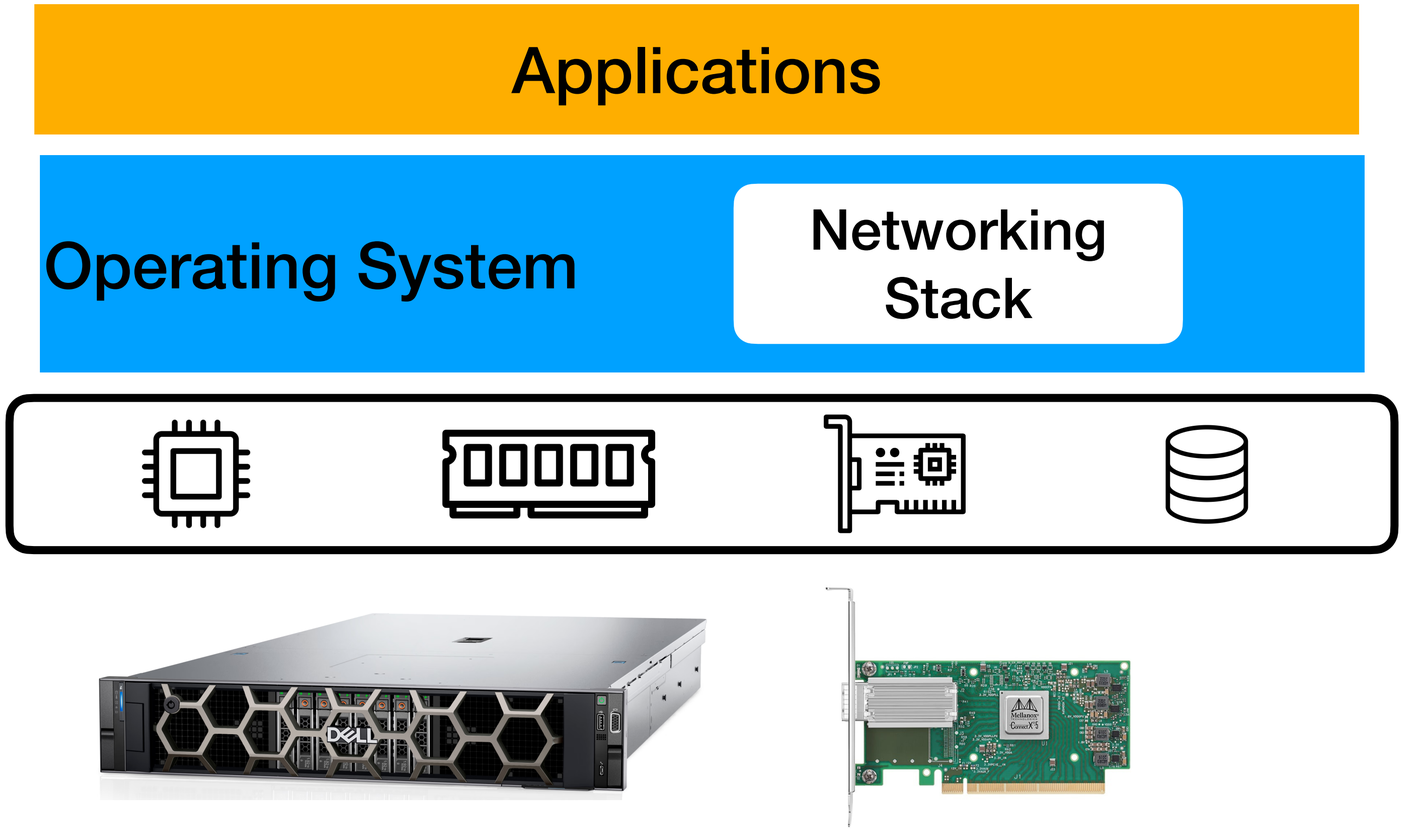
Recap

- Key Questions:
  - How can we leverage in-network intelligence to improve the efficiency of the transport protocol?

- Terminology
  - Active Queue Management (AQM)
  - Max-Min Fairness
  - Fair Queueing
  - Random Early Detection

The Host Networking Stack

- Virtualize the NIC and provide the communication services
  - Interact with the device driver and I/O subroutine

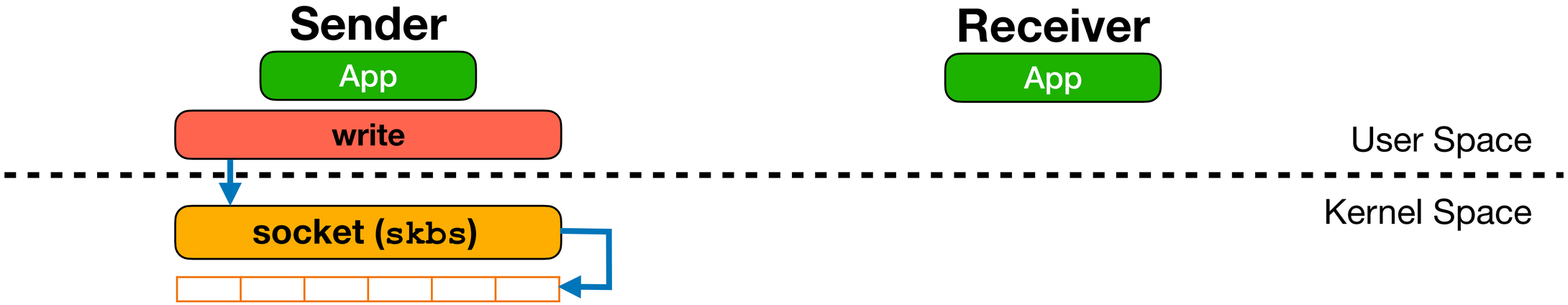


Linux Network Stack Data Path Walk-Through

Linux Network Stack Data Path Walk-Through

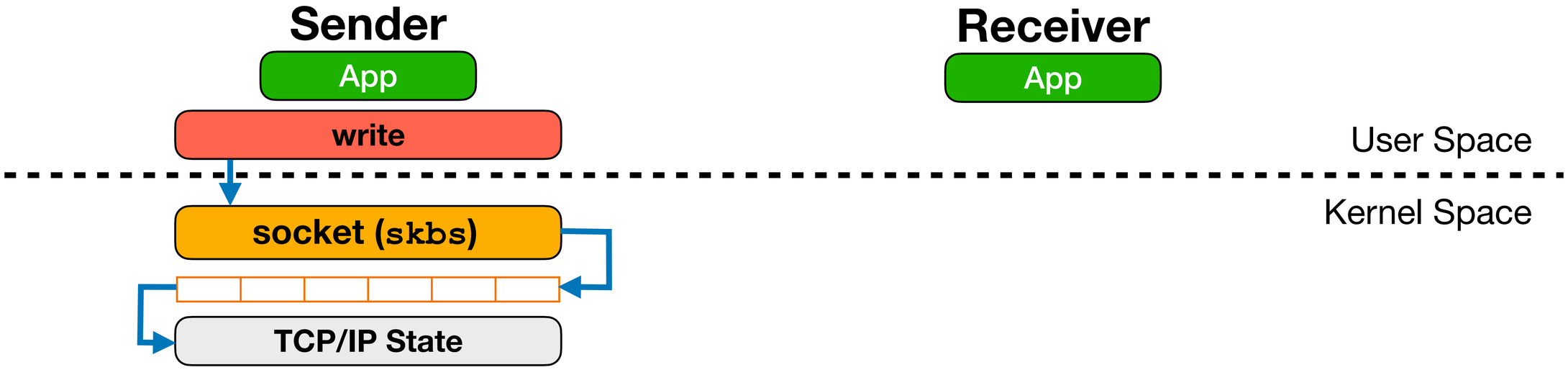
- Sender #1: Apps execute a `write` system call

Linux Network Stack Data Path Walk-Through



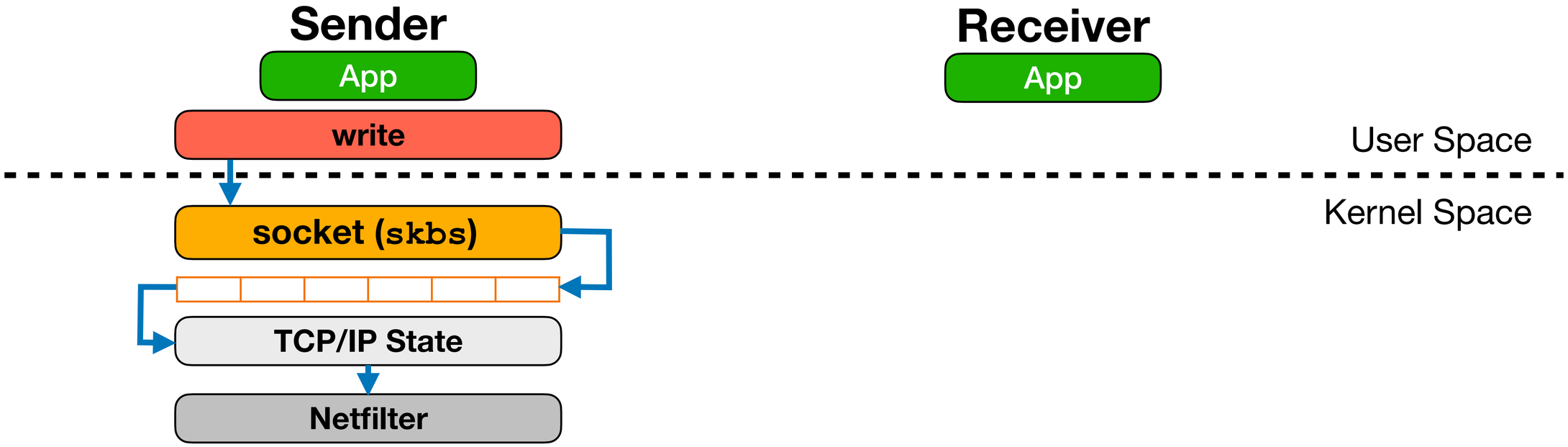
- Sender #2: The kernel initializes socket buffers

Linux Network Stack Data Path Walk-Through



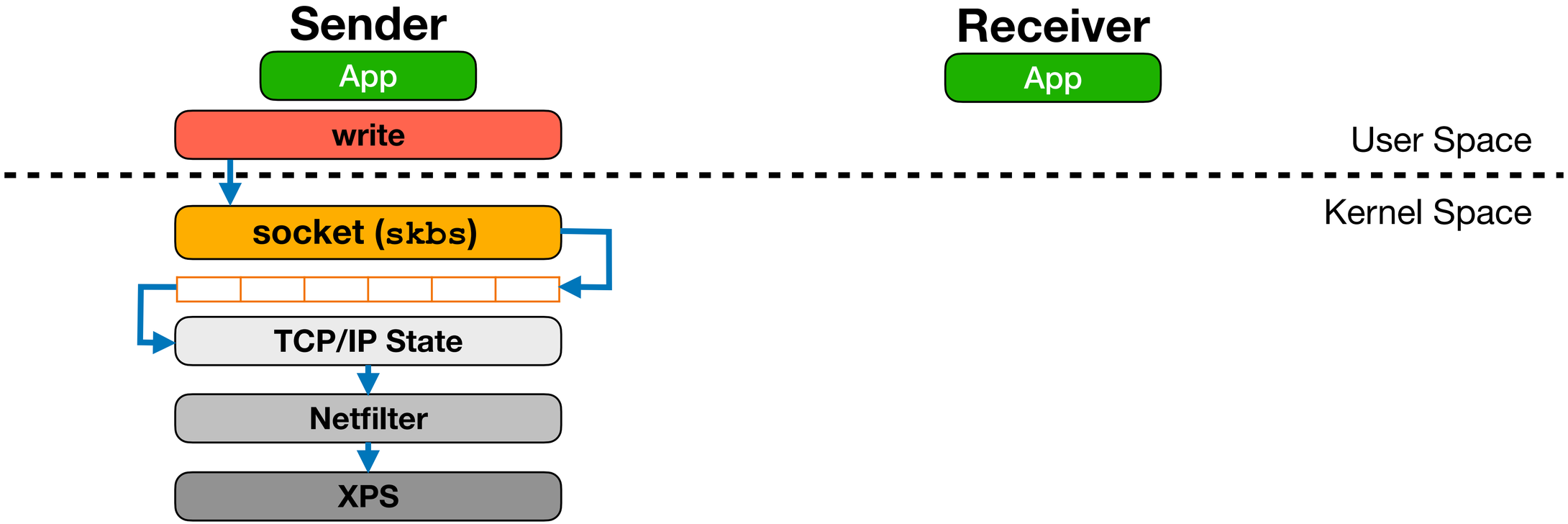
- Sender #3: `skbs` are processed by the TCP/IP layer

Linux Network Stack Data Path Walk-Through



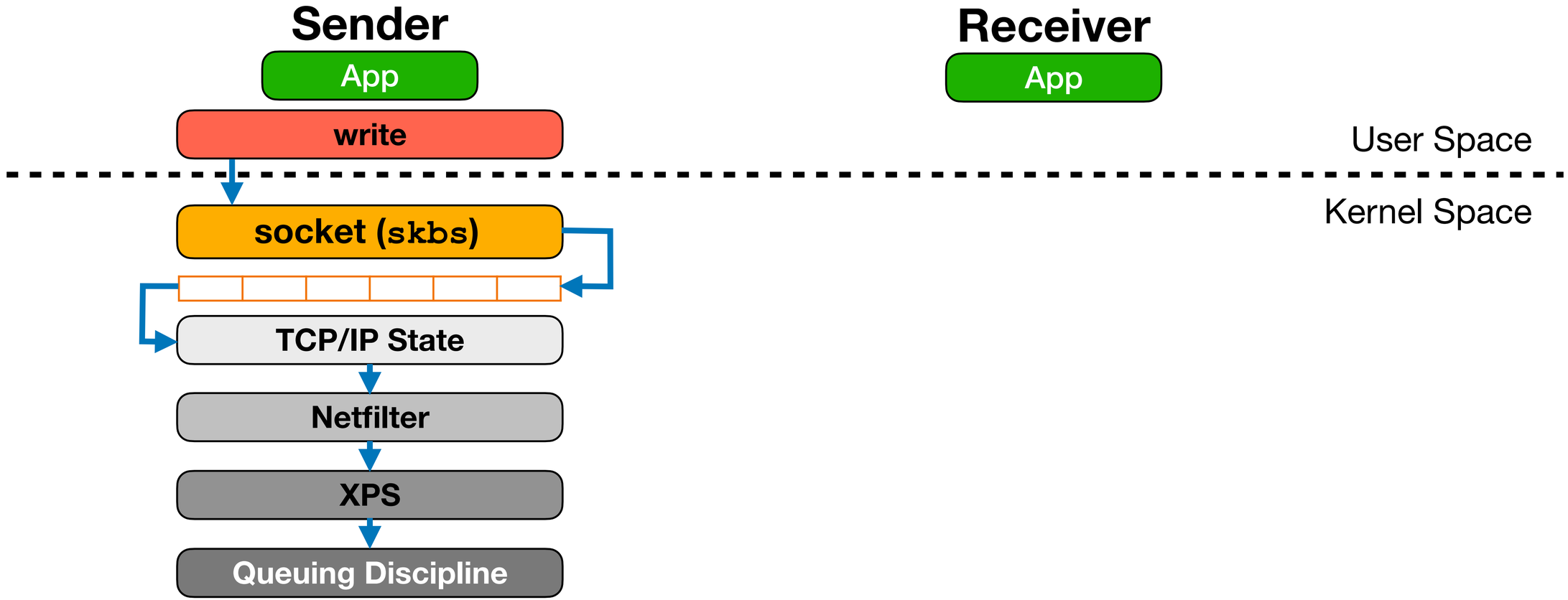
- Sender #4: Packets (`skbs`) are processed by customized networking functions
- Netfilter: a framework provided by the Linux kernel, allowing registering callback functions for packet handling

Linux Network Stack Data Path Walk-Through



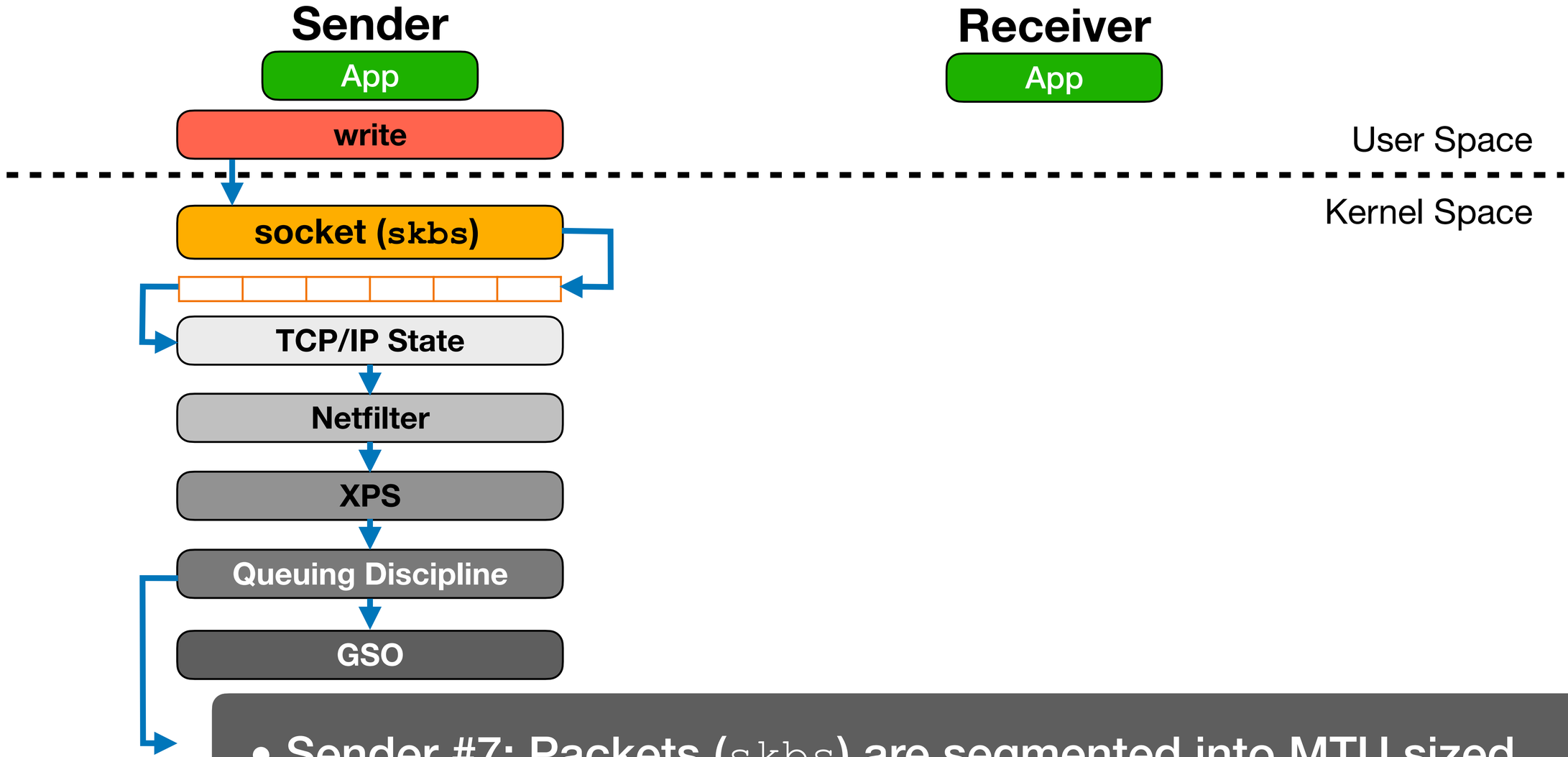
- Sender #5: Packets (`skbs`) are orchestrated by transmit-side traffic steering (XPS)
- Two approaches: using CPUs map or receive queue map

Linux Network Stack Data Path Walk-Through



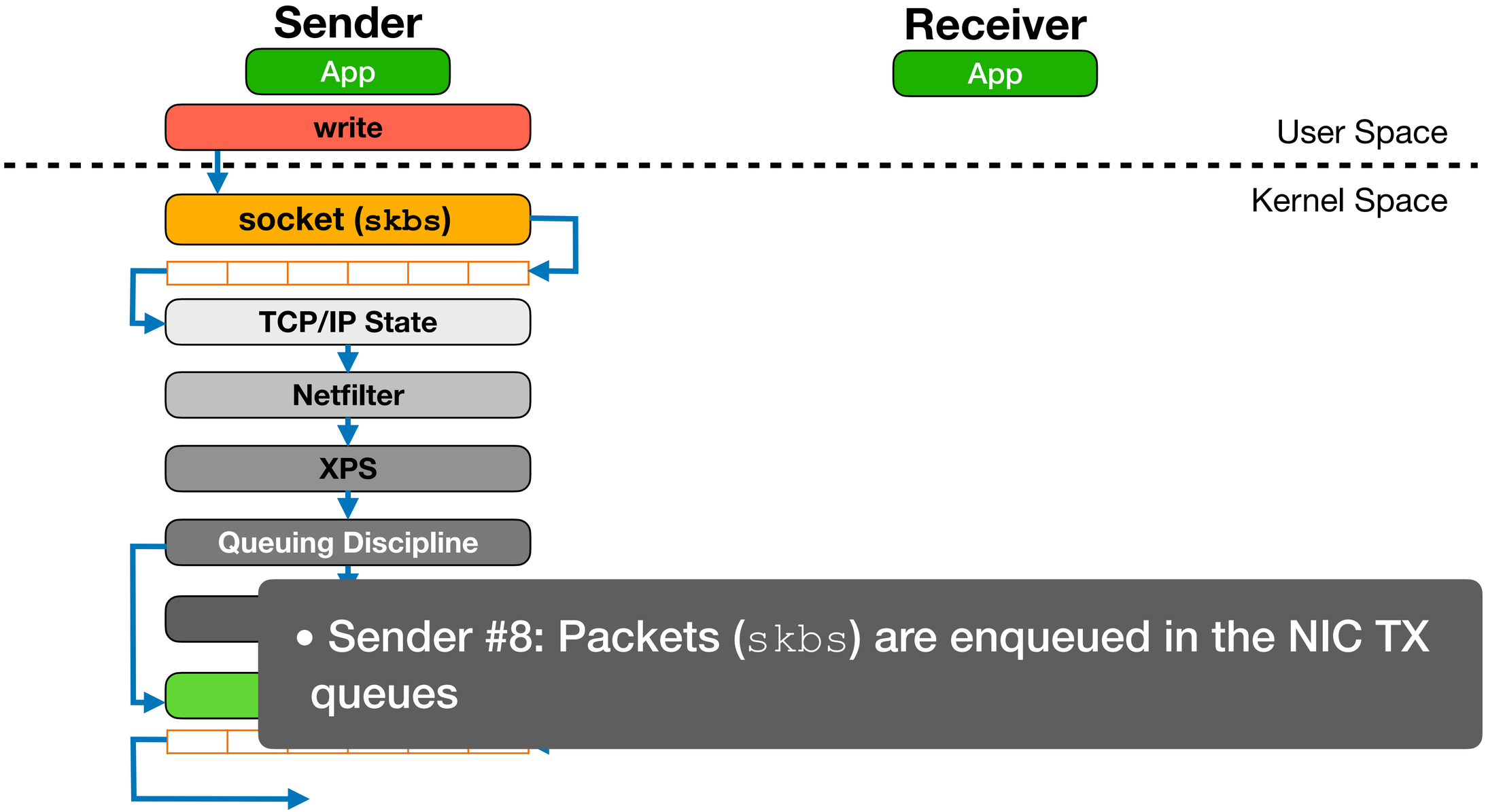
- Sender #6: Packets (`skbs`) are shaped via `qdisc`
- Rate limiting, FIFO, priority

Linux Network Stack Data Path Walk-Through

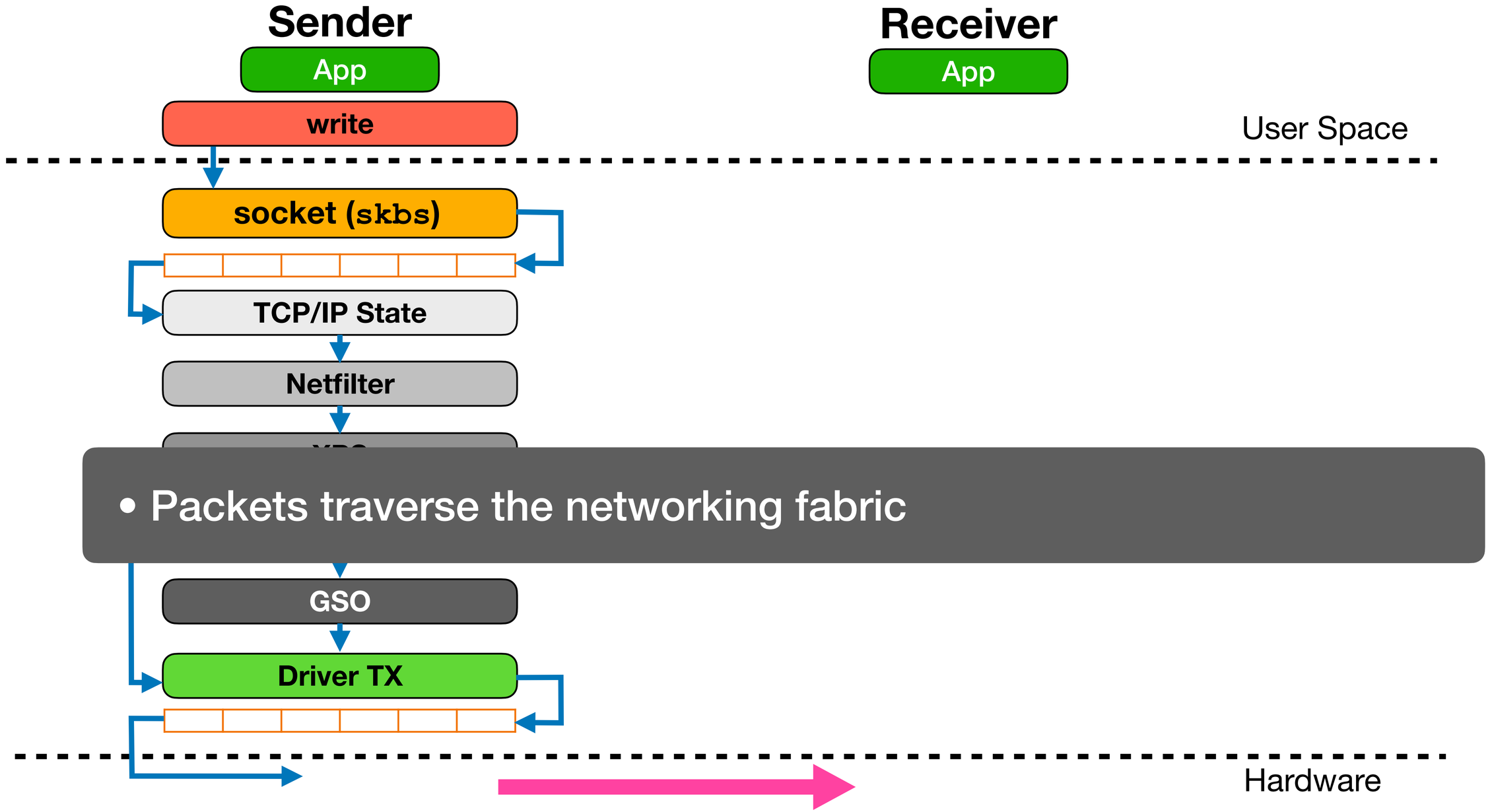


- Sender #7: Packets (`skbs`) are segmented into MTU sized chunks

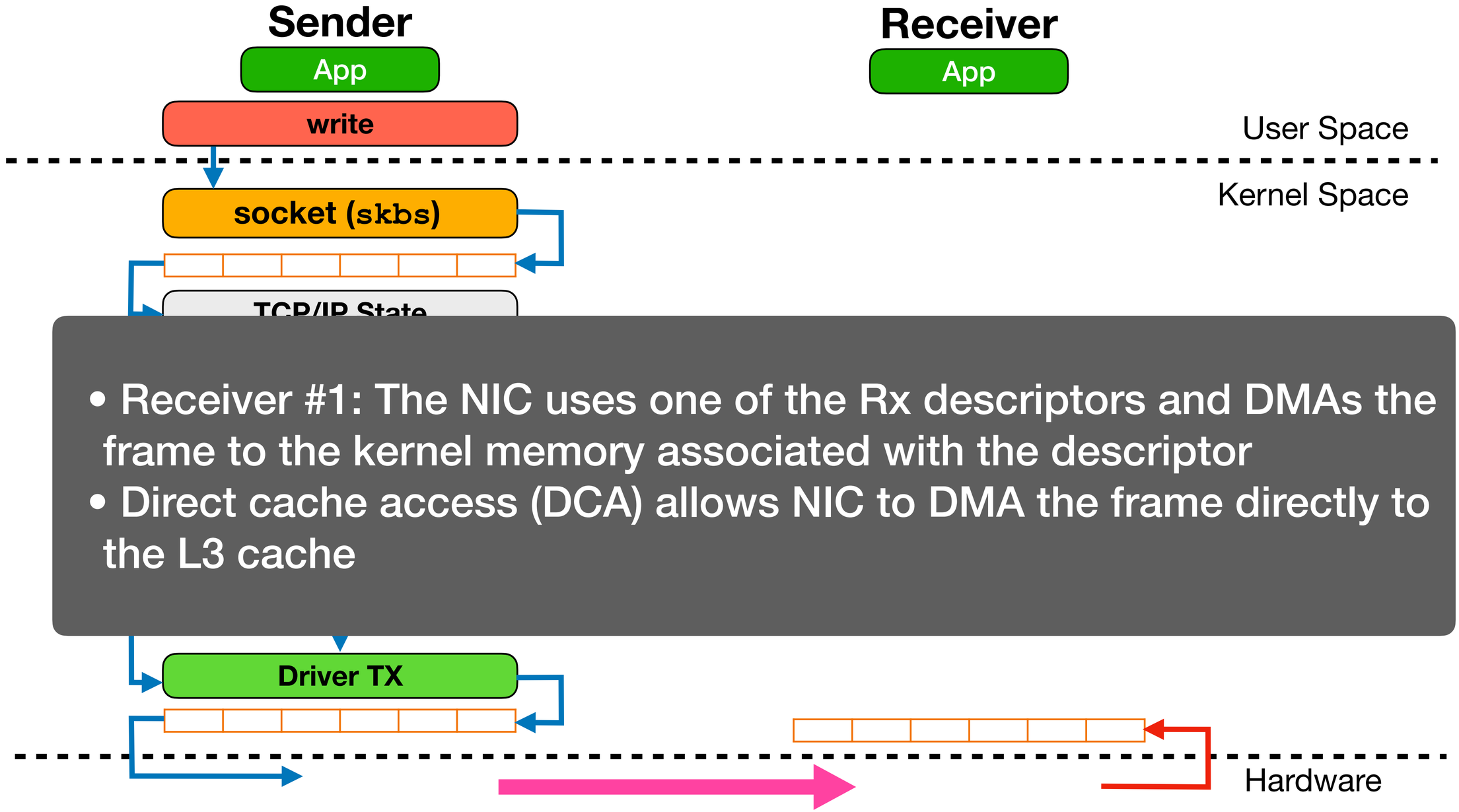
Linux Network Stack Data Path Walk-Through



Linux Network Stack Data Path Walk-Through

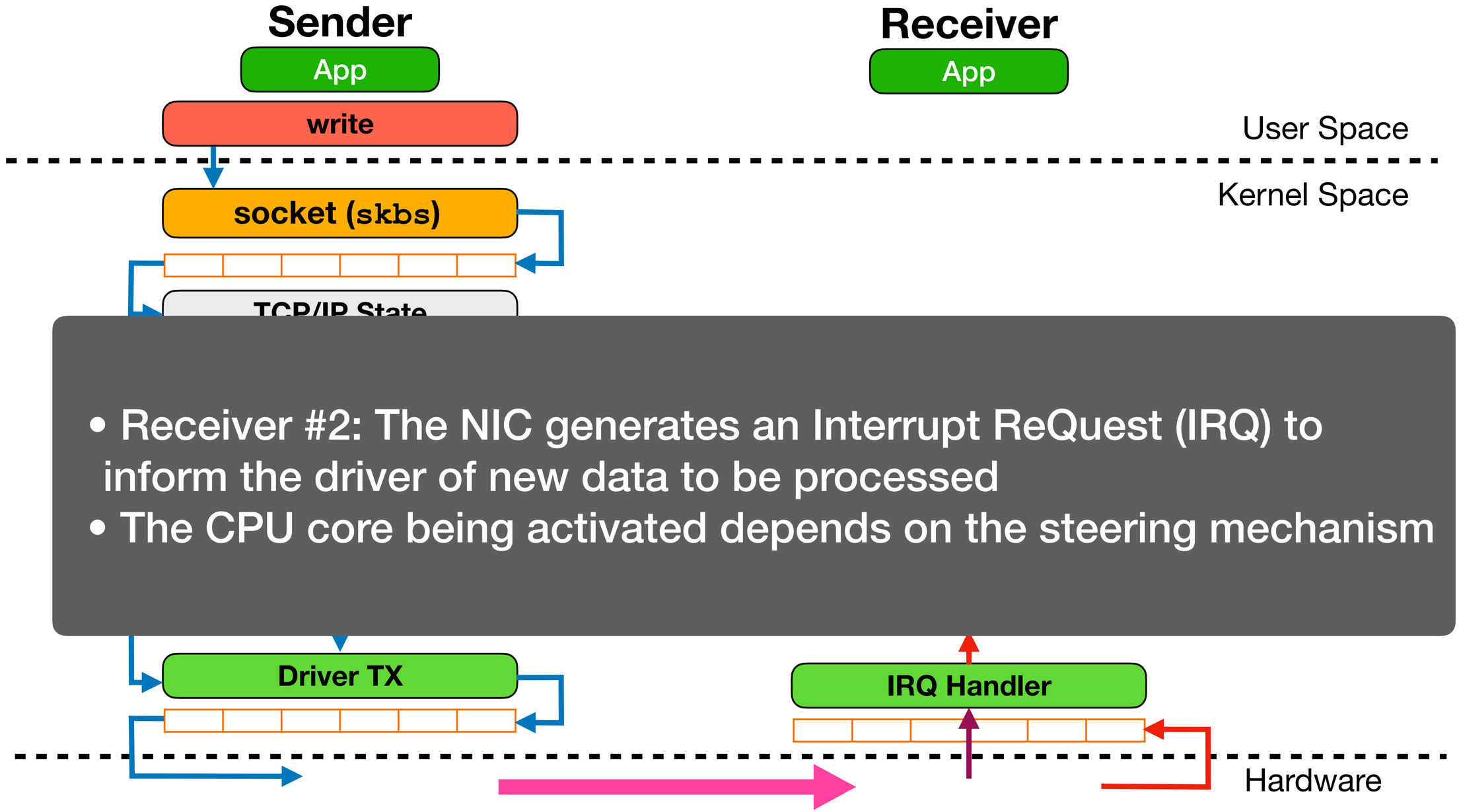


Linux Network Stack Data Path Walk-Through

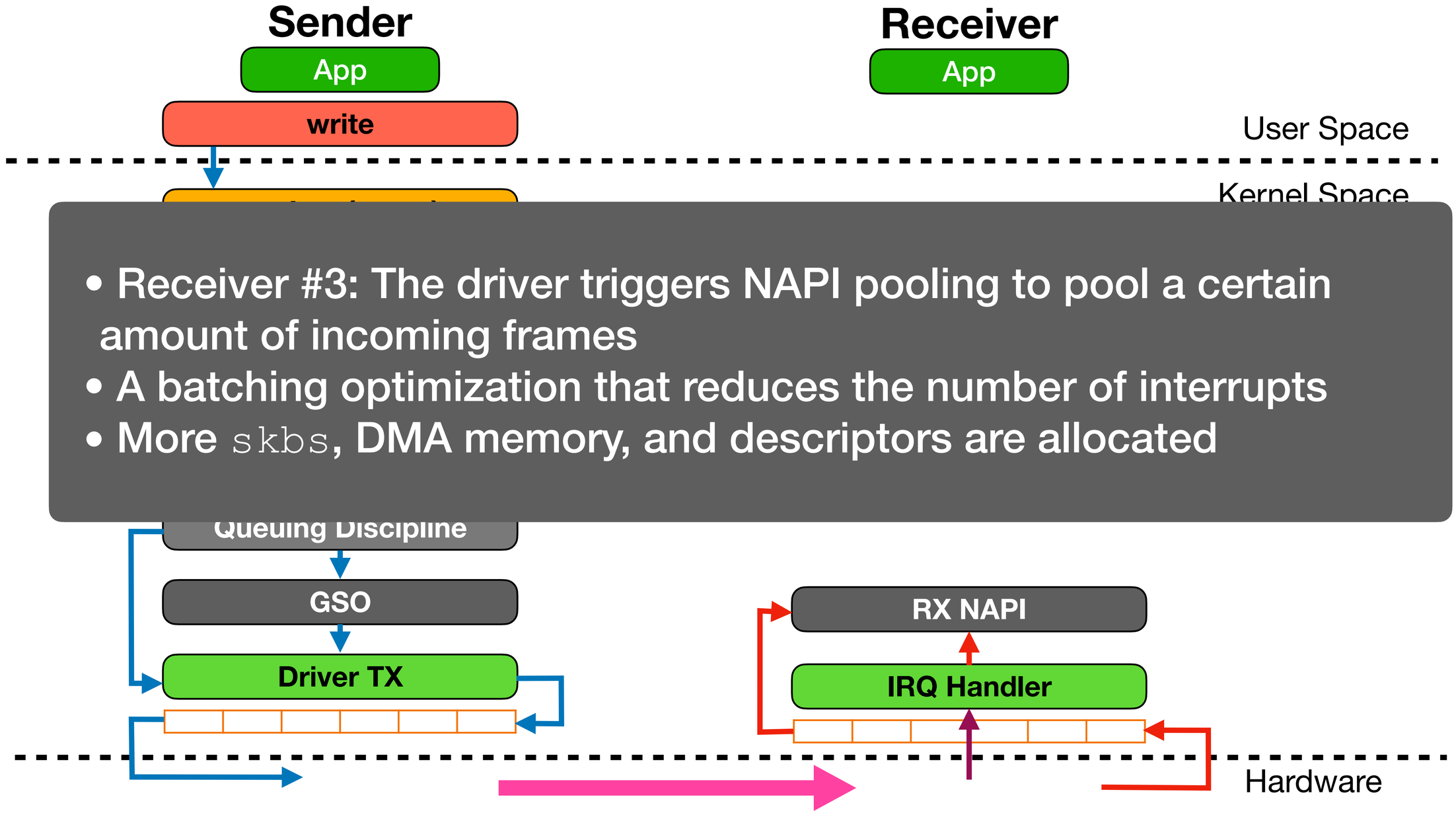


- Receiver #1: The NIC uses one of the Rx descriptors and DMAs the frame to the kernel memory associated with the descriptor
- Direct cache access (DCA) allows NIC to DMA the frame directly to the L3 cache

Linux Network Stack Data Path Walk-Through

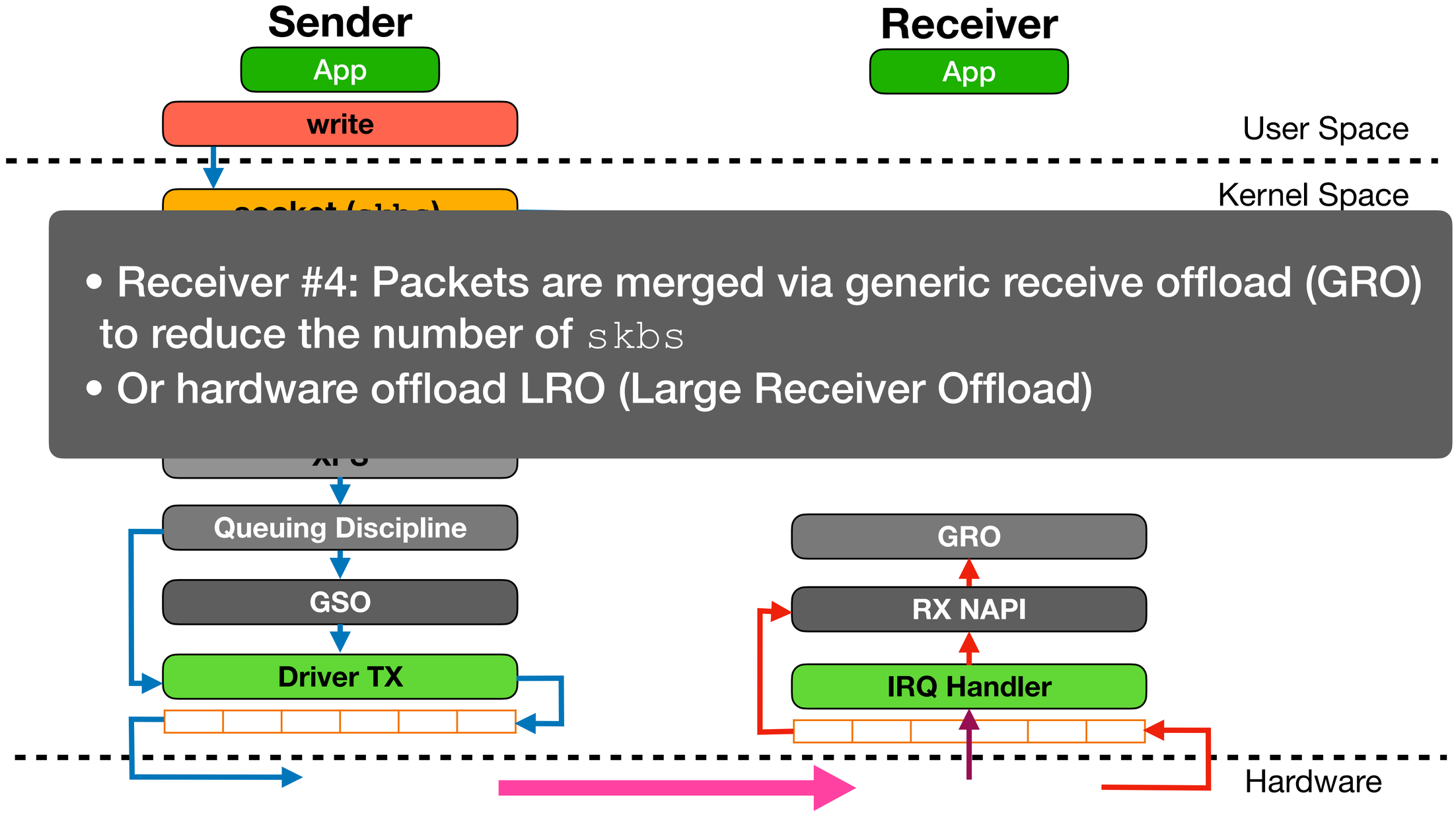


Linux Network Stack Data Path Walk-Through

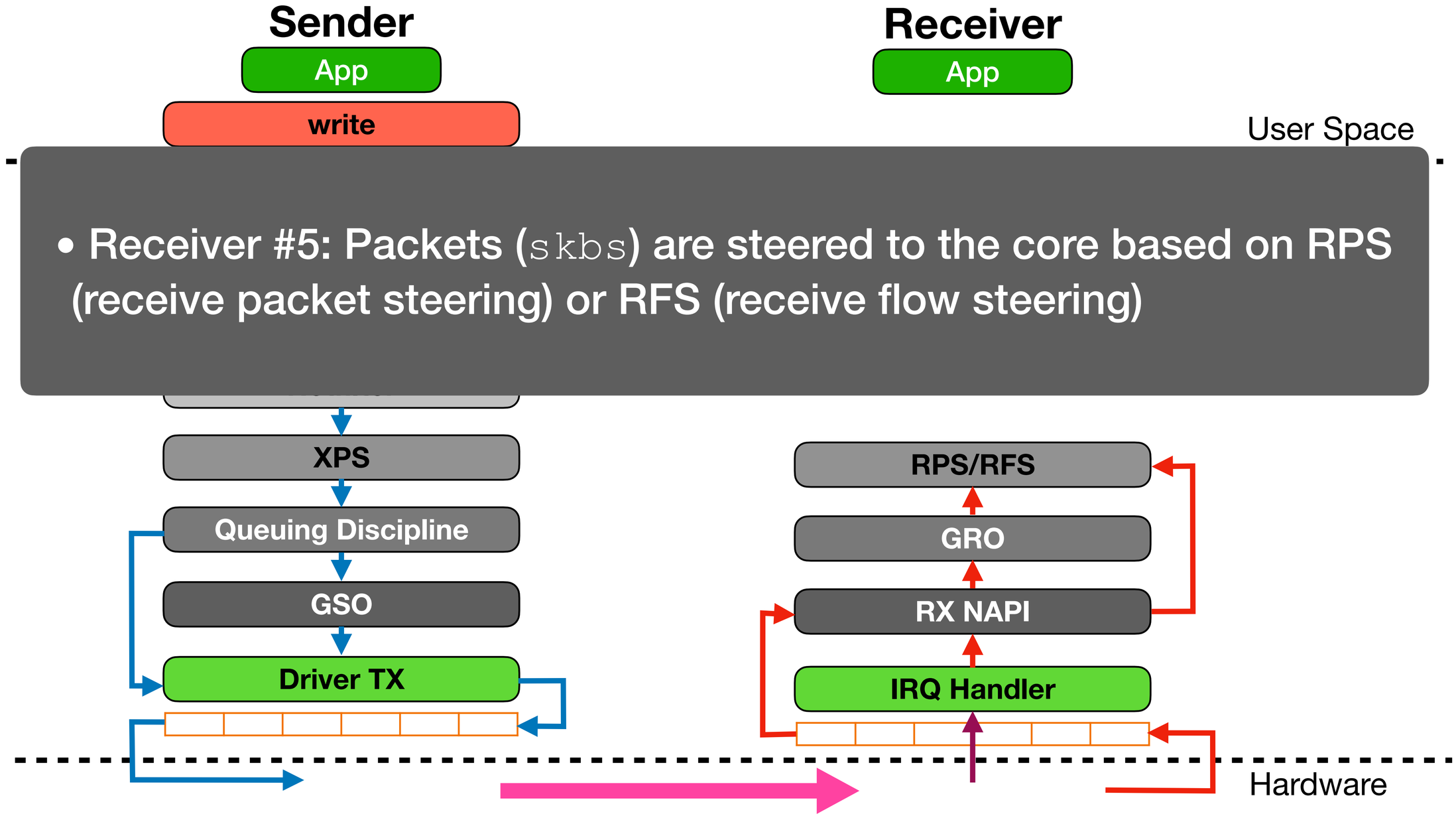


- Receiver #3: The driver triggers NAPI pooling to pool a certain amount of incoming frames
- A batching optimization that reduces the number of interrupts
- More `skbs`, DMA memory, and descriptors are allocated

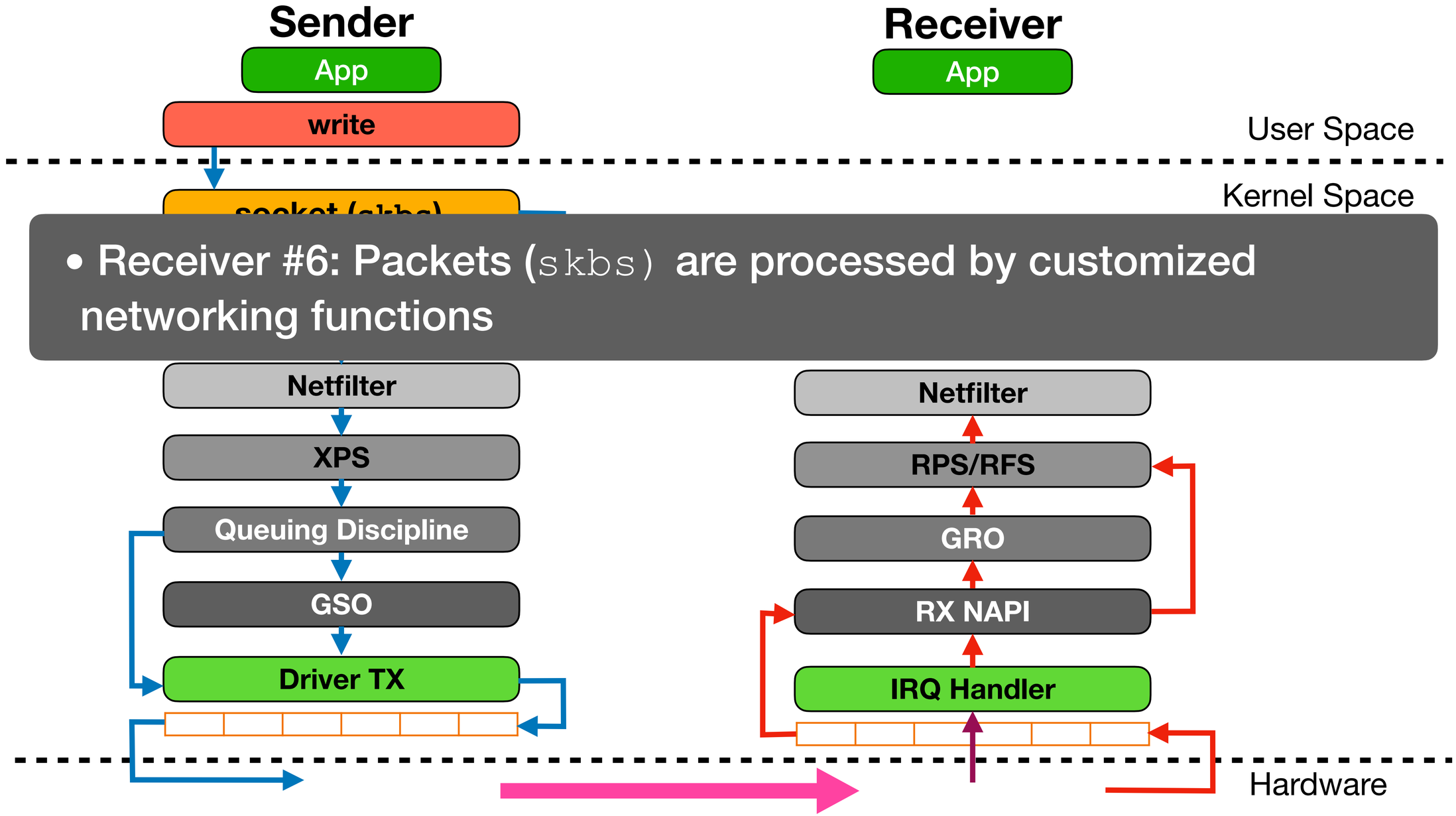
Linux Network Stack Data Path Walk-Through



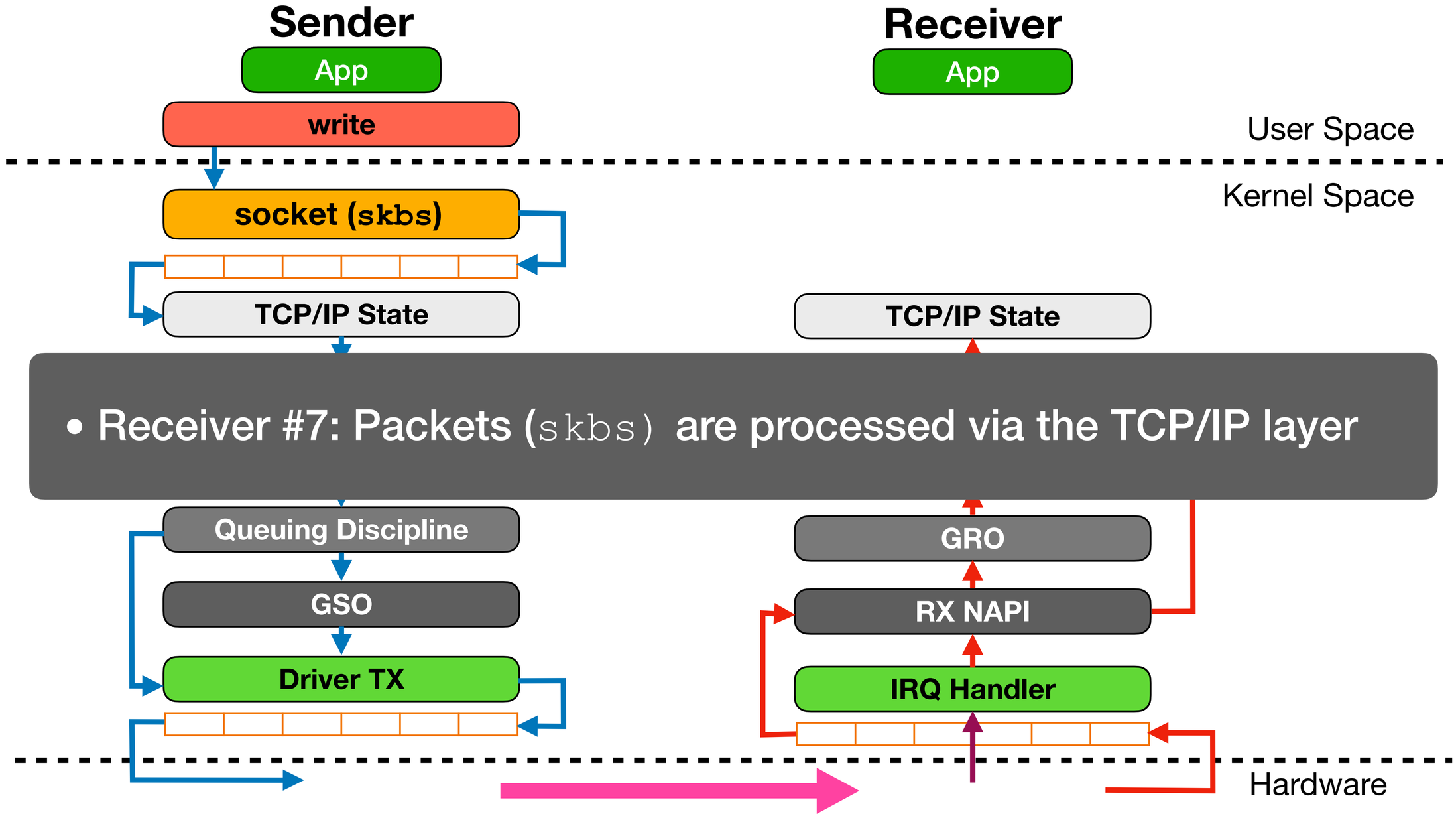
Linux Network Stack Data Path Walk-Through



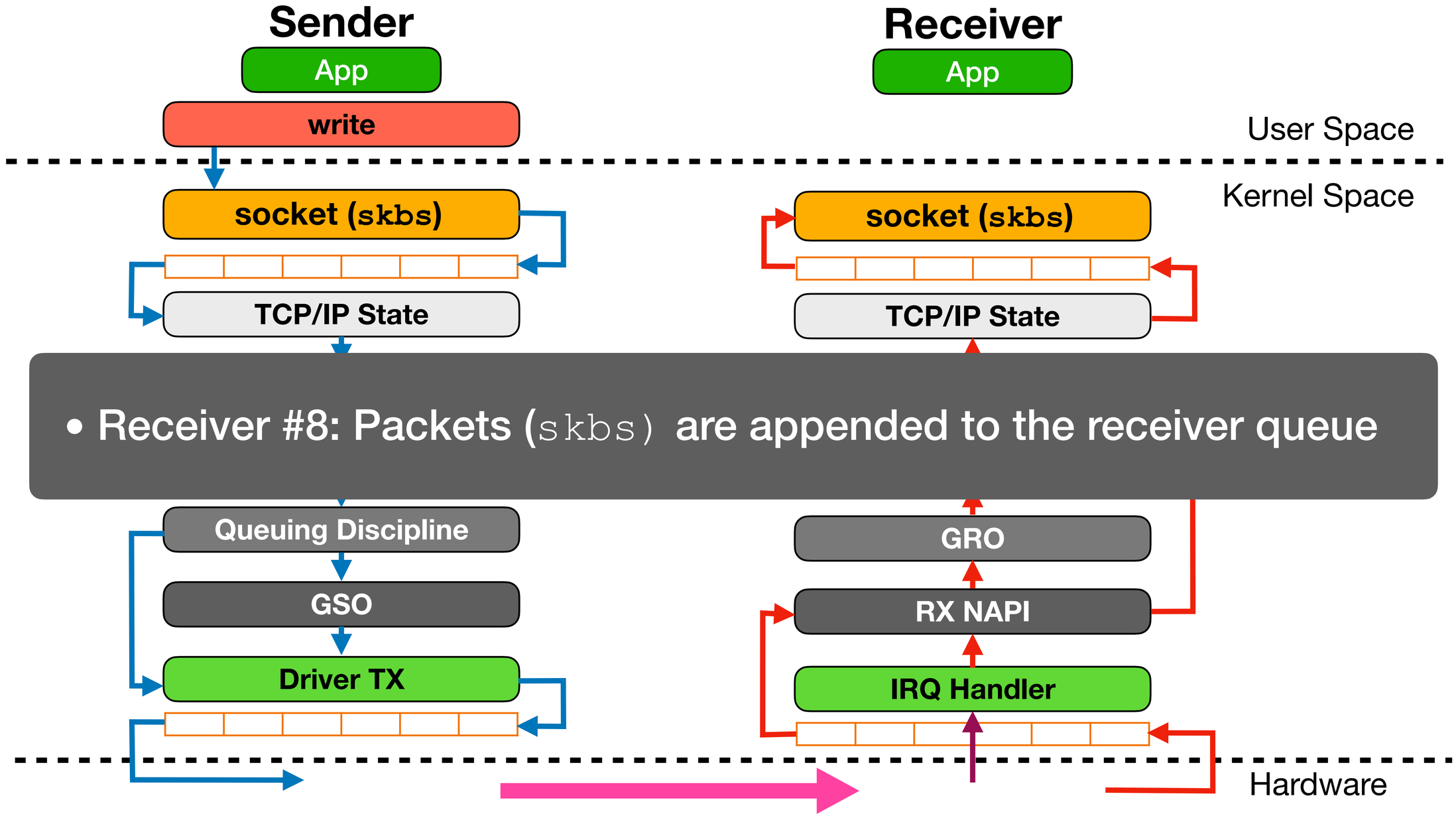
Linux Network Stack Data Path Walk-Through



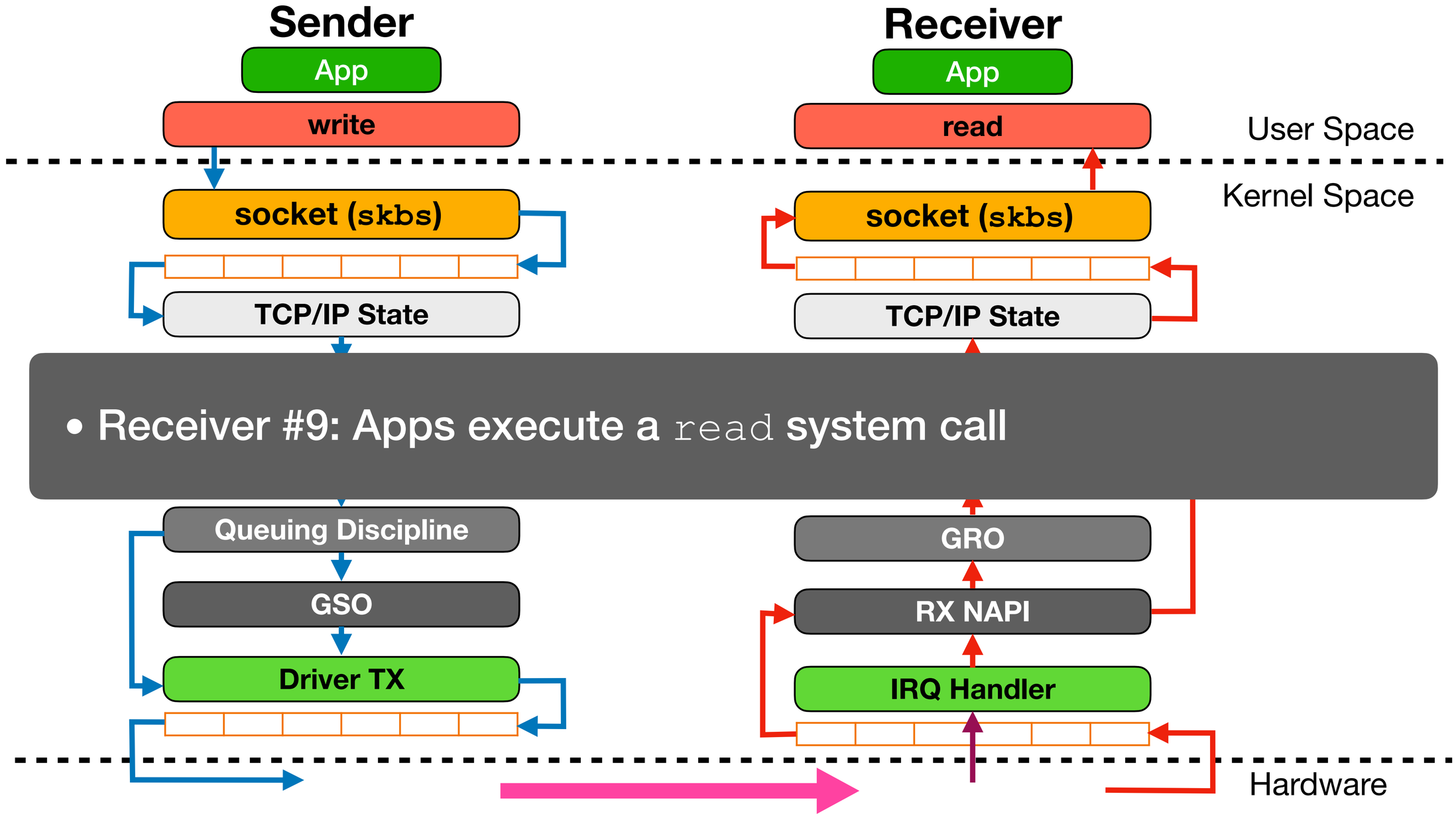
Linux Network Stack Data Path Walk-Through



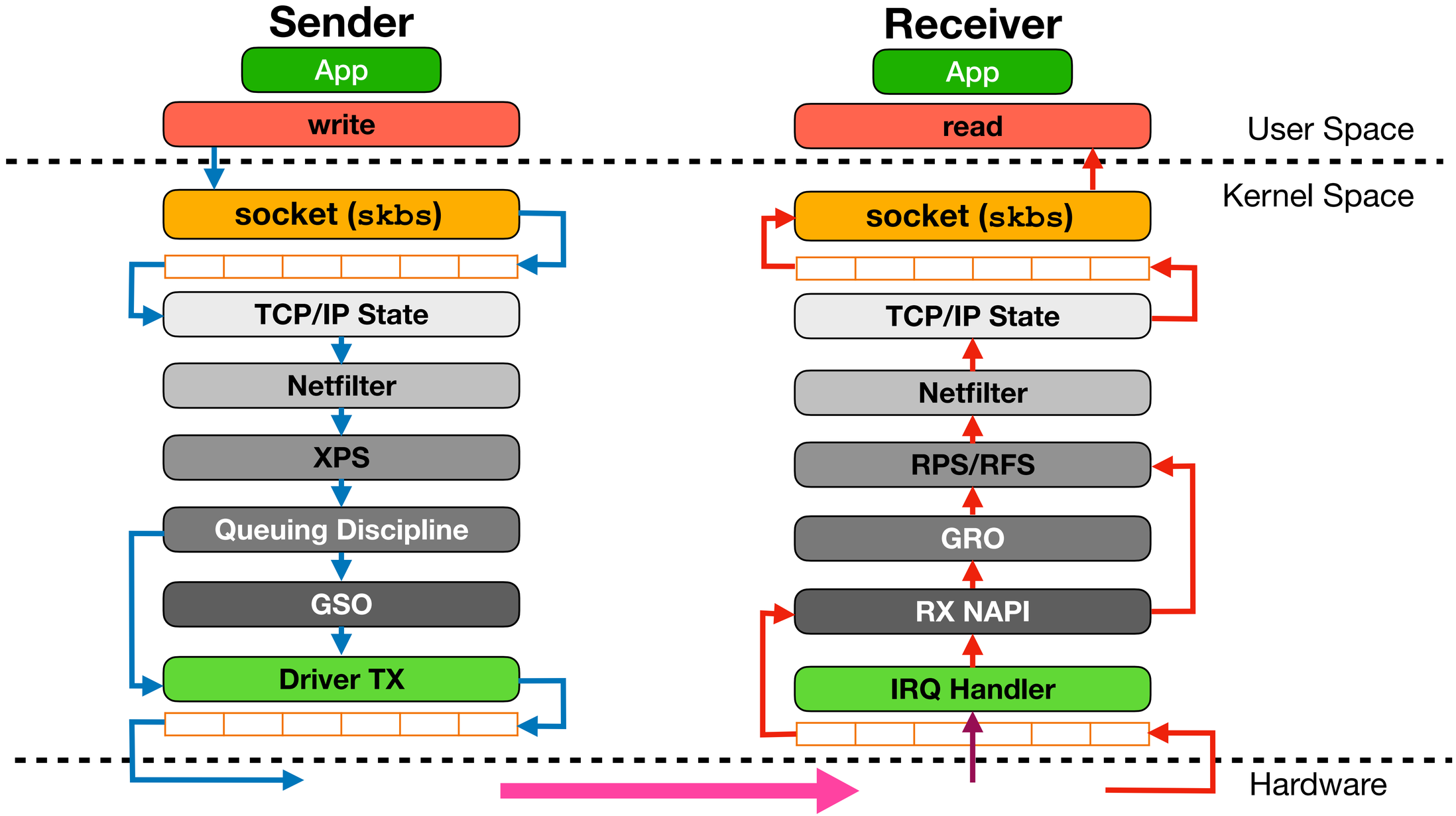
Linux Network Stack Data Path Walk-Through



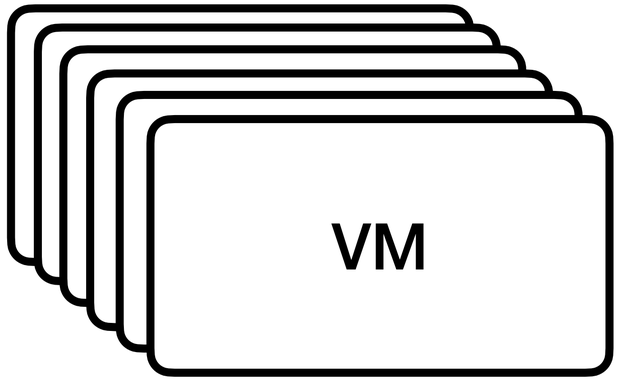
Linux Network Stack Data Path Walk-Through

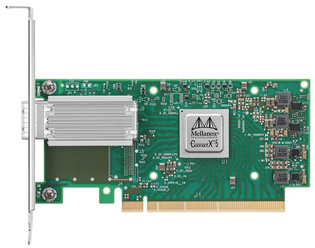


Linux Network Stack Data Path Walk-Through



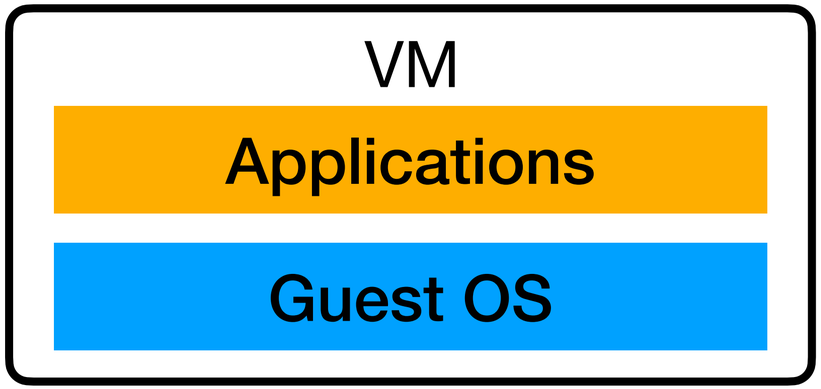
How network is virtualized?

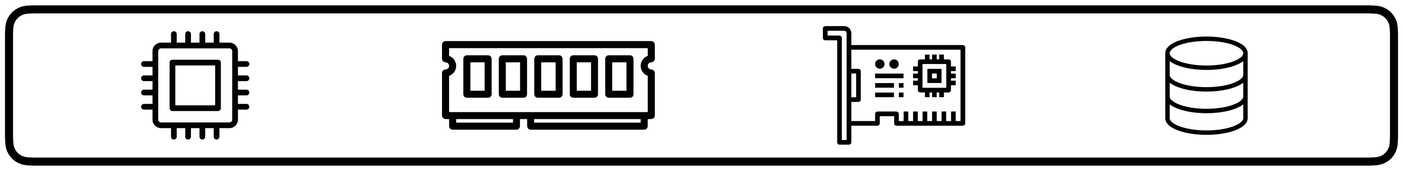




Virtual Machine Overview

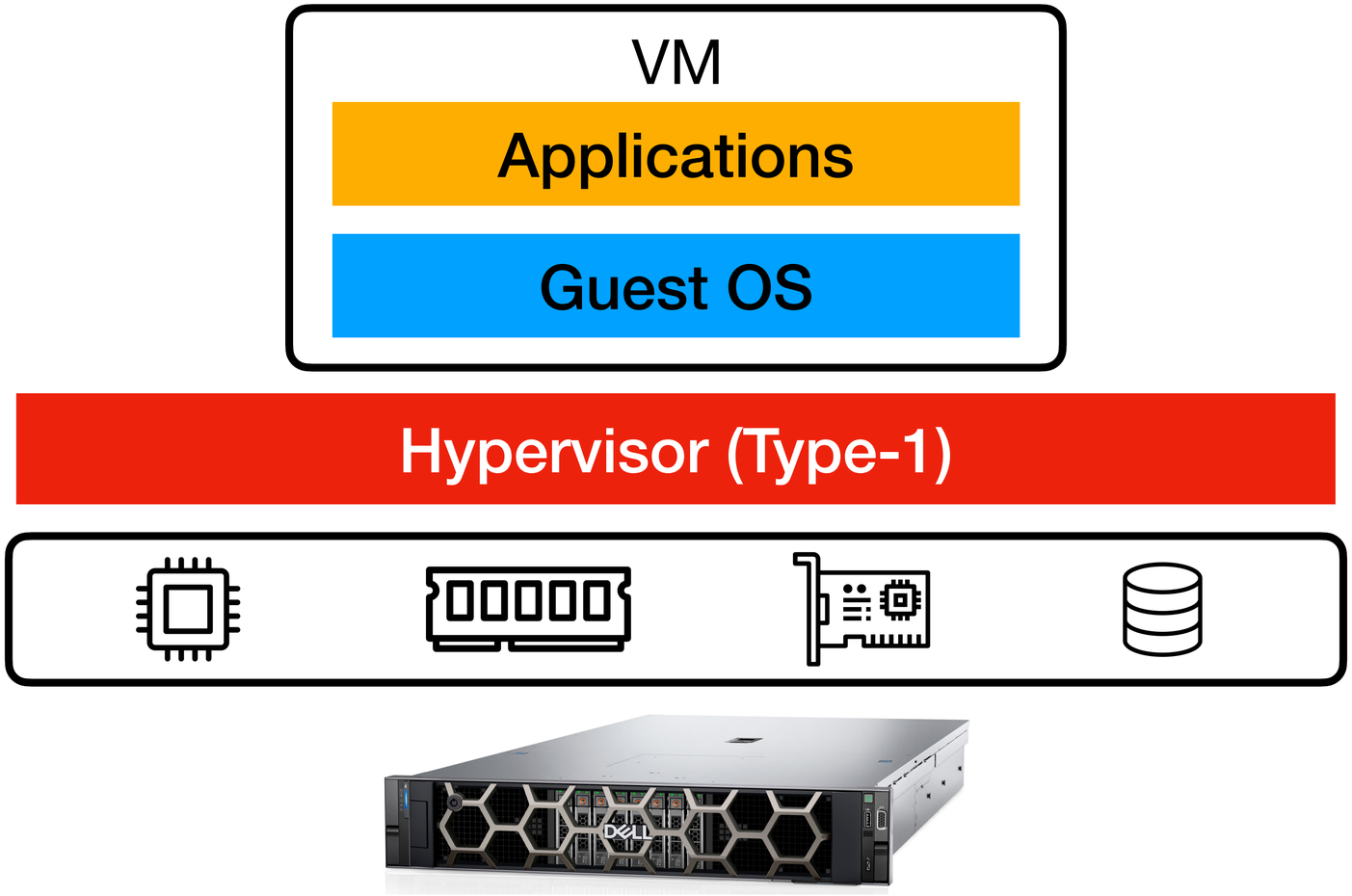
- A VM is an isolated computing environment
  - CPU, memory, network, and storage



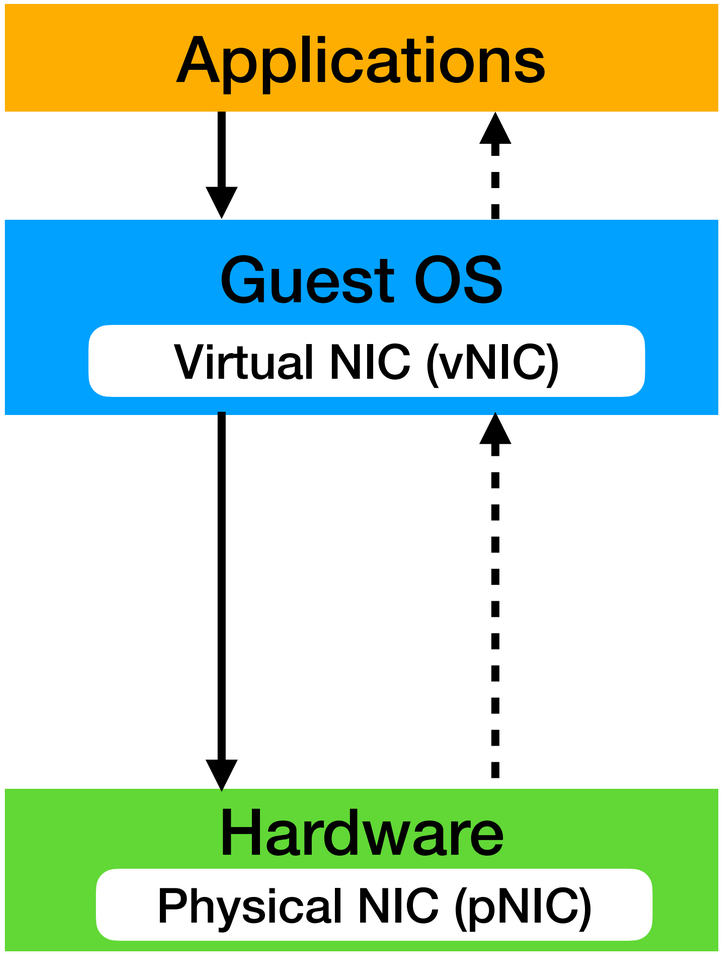


Virtual Machine Overview

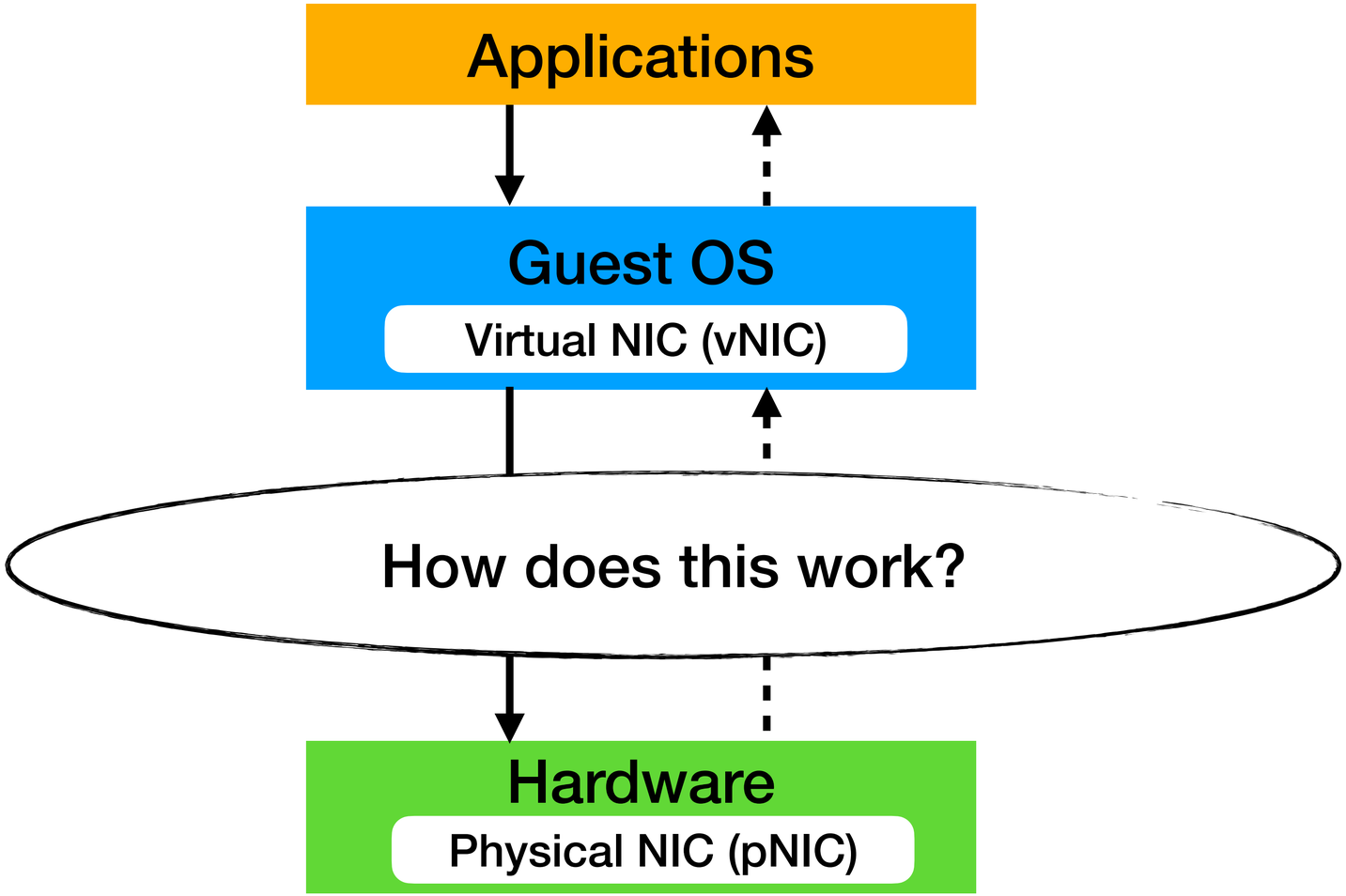
- A hypervisor is the privileged software for resource management.
  - E.g., KVM, Xen, Hyper-V, VMware ESXi, …



Set Up the Context



Set Up the Context



What is a vNIC?

- The network interface representation of a virtual machine
  - E.g., ifconfig

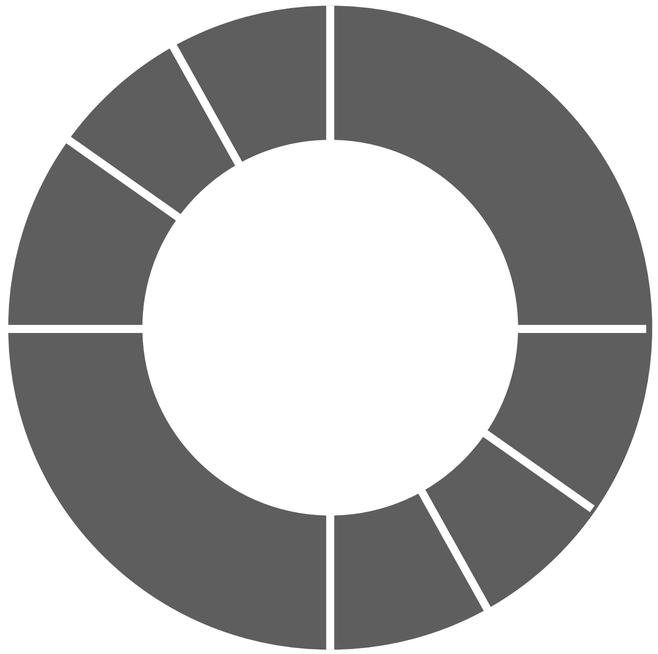
What is a vNIC?

- The network interface representation of a virtual machine
  - E.g., ifconfig

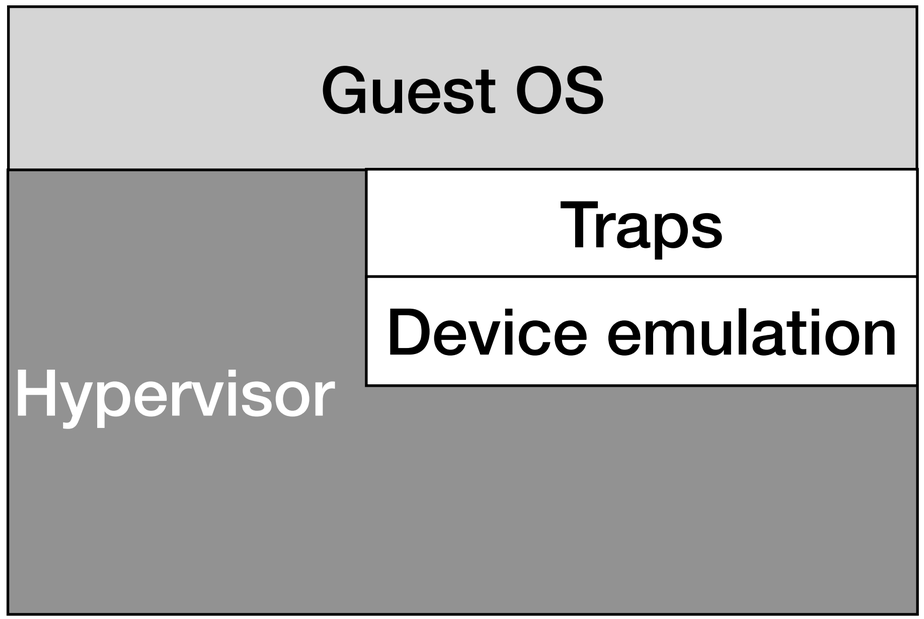
But what is “representation”?

What is a vNIC?

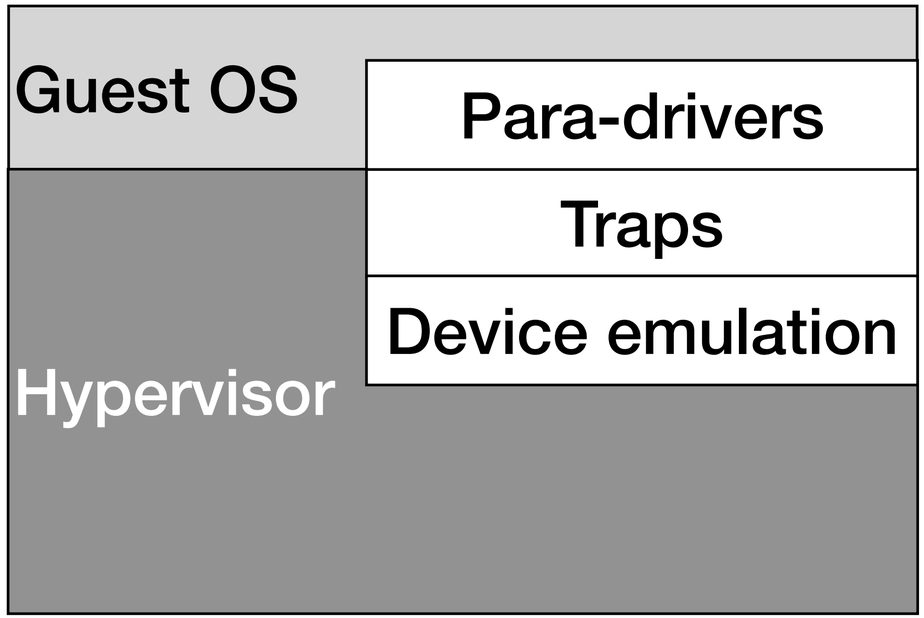
- The network interface representation of a virtual machine
  - E.g., ifconfig
- The communication buffer exposed by the virtual device



Full Virtualization v.s. Para-Virtualization



Full Virtualization



Para-Virtualization

Full Virtualization v.s. Para-Virtualization

```
void nic_write_buffer(char *buf, int size) {
  for (; size > 0; size--) {
    nic_poll_ready();
    outb(NIC_TX_BUF, *buf++);
  }
}
```

Full Virtualization

```
void nic_write_buffer(char *buf, int size) {
  vmm_write(NIC_TX_BUF, buf, size);
}
```

Para-Virtualization

Transport Layer Summary

Physical layer	A reliable (and efficient) bit delivery channel over a link

Transport Layer Summary

Link layer	A frame delivery channel between directly connected or switched hosts
Physical layer	A reliable (and efficient) bit delivery channel over a link

Transport Layer Summary

IP (Network) layer	A datagram delivery channel between hosts in any network under the best-effort service model
Link layer	A frame delivery channel between directly connected or switched hosts
Physical layer	A reliable (and efficient) bit delivery channel over a link

Transport Layer Summary

Transport layer	A data segment delivery channel between two processes in any network
IP (Network) layer	A datagram delivery channel between hosts in any network under the best-effort service model
Link layer	A frame delivery channel between directly connected or switched hosts
Physical layer	A reliable (and efficient) bit delivery channel over a link

Transport Layer Summary

Q1: How can we set up the process-to-process channel?
Q2: How can we multiplex concurrent channels over the physical link?
Q3: How can we control the transmission rate?
Q4: How can we achieve reliable delivery?
Q5: How can we share the in-network bandwidth resources?

Tran
laye

IP (N
layer

under the best-effort service model

Link layer

A frame delivery channel between directly connected or switched hosts

Physical layer

A reliable (and efficient) bit delivery channel over a link

Transport Layer Summary

Tran
laye

IP (N
laye

Link

Q1: How can we set up the process-to-process channel?
=> UDP: arbitrary communication
=> TCP: state machine-based connection management
Q2: How can we multiplex concurrent channels over the physical link?
=> UDP/TCP: destination port as a demultiplexing key
=> UDP/TCP: host service model (each flow owns a queue)
Q3: How can we control the transmission rate?
=> UDP: no control
=> TCP: congestion control + flow control

Physical layer

A reliable (and efficient) bit delivery channel over a link

Transport Layer Summary

Tran
laye

IP (N
laye

Q4: How can we achieve reliable delivery?
=> UDP: just checksum
=> TCP: sliding window + flow control
Q5: How can we share the in-network bandwidth resources?
=> UDP: no control
=> TCP: congestion control

Link layer

A frame delivery channel between directly connected or switched hosts

Physical layer

A reliable (and efficient) bit delivery channel over a link

What are infrastructure services used for?

What are infrastructure services used for?

Provide network-assisted functionalities

- #1: Domain Name System (DNS)
- #2: Simple Network Management Protocol (SNMP)

DNS Motivation: Naming Hosts

- So far, we have identified hosts using IP/MAC addresses
  - Hard for humans to remember these identifiers

- Goal: assign human-friendly names to hosts
  - But routing still needs IP addresses
  - Define and look up the mapping between a hostname and an IP address

Naive Approach

- Early Internet: a file mapping IP addresses to hostnames
  - Manually updated and copied to hosts on the Internet

- Problem: does not scale
  - Still useful for small-scaled local networks
  - Look at the /etc/hosts file on a CS department machine

Domain Name System (DNS)

- #1: Distributed name resolution system
  - Many name servers (NSs) are distributed throughout the Internet
  - E.g., ISPs, campus network, Enterprise network, etc.

- #2: Domain names (DNs) are hierarchical
  - A single NS doesn’t need to store the name for every host on the Internet

Domain Name System (Cont'd)

- #3: DNs can be mapped to IPv4/IPv6 addresses, and other DNs
  - Mapping can be changed or based on other factors (e.g., geo location)

- #4: Queries are issued to a sequence of NSs
  - Each knows about a different part of the DN hierarchy
  - Answers can be cached to avoid the overhead of frequent lookups

Domain Name Hierarchy

- DNs are processed from right to left
  - Period: separator
  - The rightmost name is the top of the hierarhcy
  - The leftmost name is at the bottom

- Example: cs.wisc.edu
  - Top of hierarhcy — edu (rightmost name)
  - Bottom of hierarchy — cs (leftmost name)

Domain Name Hierarchy — TLDs

- Top-level domain (TLD): rightmost name
  - Original TLDs were designed for the US: edu, com, gov, mil, org, net
  - Expanded to include TLDs for countries: uk, cn, etc.
  - Expanded to address the high demand for .com: .biz, .info, .tv, .ai, etc.
  - Recently expanded to include arbitrary TLDs
  - Lots of contention over who should have rights to a specific TLD

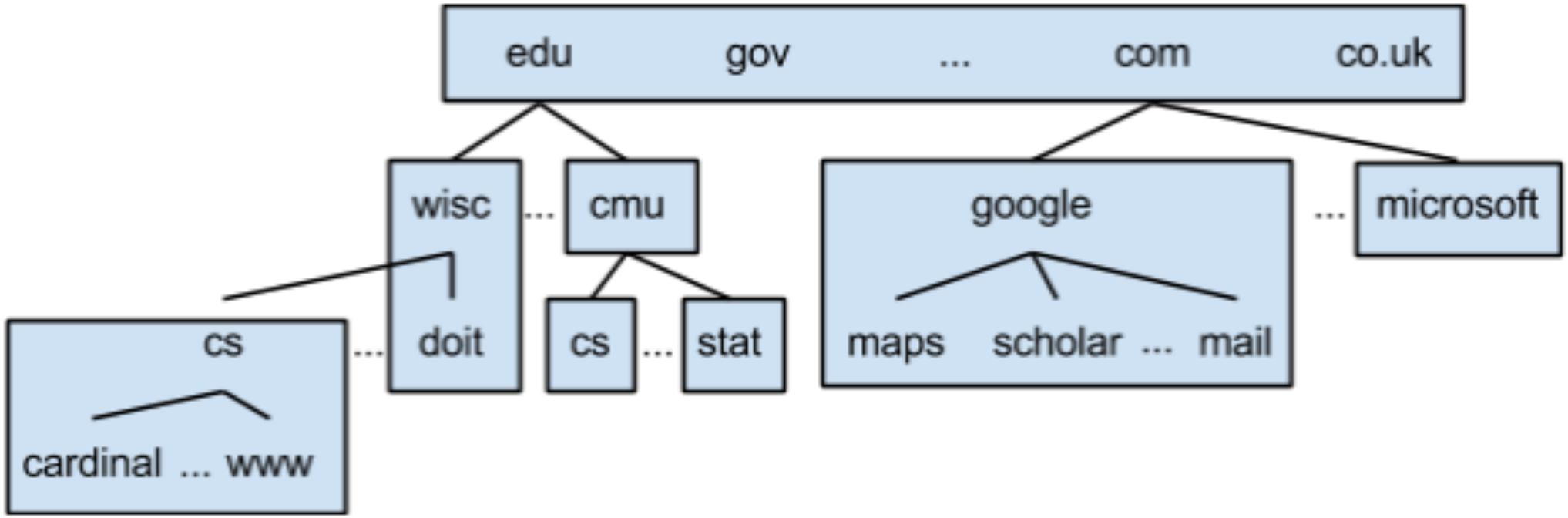
Domain Name Hierarchy — Subdomains

- Second-level domain (SLD): second from right

- DN consisting of 3+ names is often referred to as a subdomain
  - E.g., cs.wisc.edu is a subdomain of wisc.edu

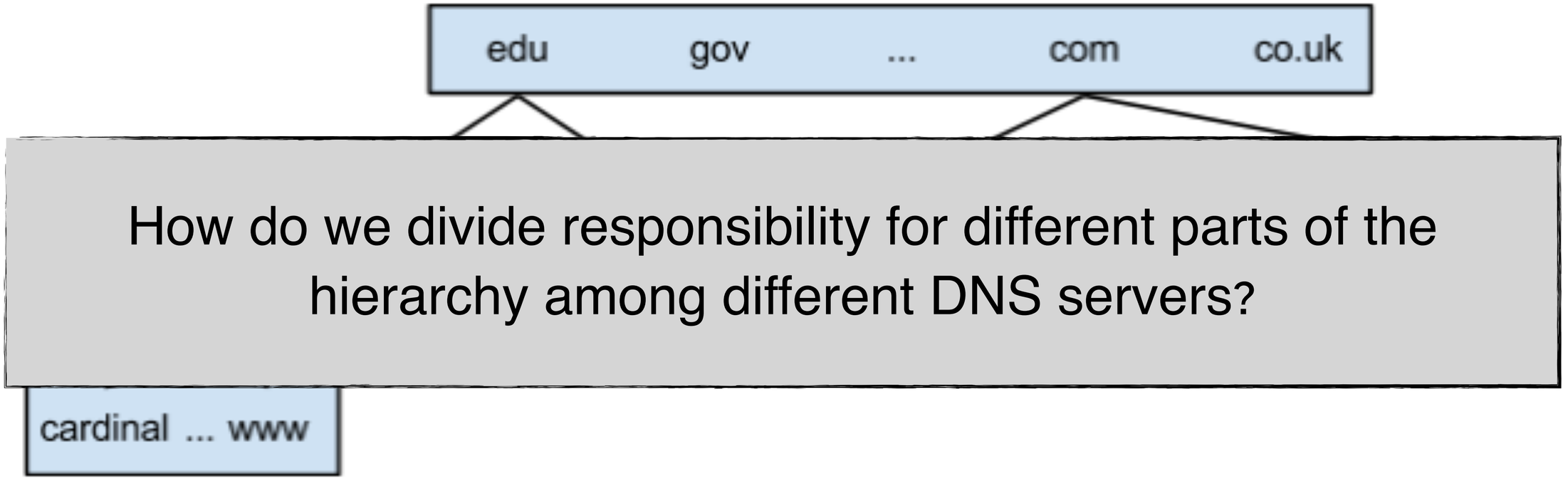
Domain Name Hierarchy — Example

- Complete hierarchy only exists conceptually
  - No single DNS server stores the entire hierarchy



Domain Name Hierarchy — Example

- Complete hierarchy only exists conceptually
  - No single DNS server stores the entire hierarchy

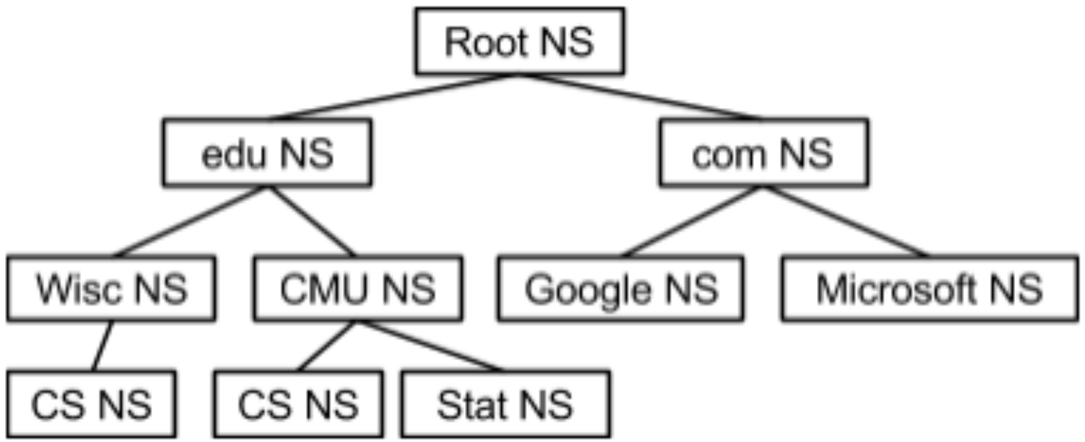


Zones

- A portion of the hierarhcy managed by an administrative entity
  - The Internet Corporation for Assigned Names and Numbers (ICANN) is responsible for the zone containing all TLDs
  - UW-Madison DoIT is responsible for the zone containing cs.wisc.edu and all subdomains, e.g., netlab-gateway.cs.wisc.edu, etc.
  - Google is responsible for the zone containing google.com and all subdomains, e.g., maps.google.com, scholar.google.com, etc.

Zones — Name Servers

- Two or more name servers (NSs) are responsible for each zone
  - Multiple NSs per zone to ensure availability in case of a failure
  - Each stores information for all domain names in the zone as records



DNS Records

- A record has: name, type, value, and TTL
  - Name: domain name
  - TTL: specify how long another DNS server can cache the record

- Type determines the “value” field
  - A: value = an IPv4 address
  - AAAA: value = an IPv6 address
  - NS: value = domain name for a DNS server responsible for the zone containing the DN
  - CNAME: value = another domain name for a particular host
  - MX: value = domain name for a mail server that accepts message for the DN

DNS Records Example: Root NS

- Root NS
  - edu, NS, a.eduservers.net
  - edu, NS, c.eduservers.net
  - a.eduservers.net, A, 192.5.6.30
  - c.eduservers.net, A, 192.26.92.30
  - …

DNS Records Example: edu NS

- edu NS
  - wisc.edu, NS, adns1.doit.wisc.edu
  - wisc.edu, NS, adns3.doit.wisc.edu
  - adns1.doit.wisc.edu, A, 144.92.9.21
  - adns3.doit.wisc.edu, A, 144.92.104.21
  - adns3.doit.wisc.edu, AAAA, 2607:f3888:aa53:3
  - cmu.edu, NS, NSAUTH1.net.cmu.edu
  - NSAUTH1.net.cmu.edu, A, 128.2.1.8
  - NSAUTH1.net.cmu.edu, AAAA, 2607:fb28::4

DNS Records Example: wisc NS

- wisc NS
  - cs.wisc.edu, NS, dns.cs.wisc.edu
  - cs.wisc.edu, NS, dns2.cs.wisc.edu
  - dns2.cs.wisc.edu, A, 128.105.2.10
  - dns2.cs.wisc.edu A, 128.105.6.12

DNS Records Example: cs NS

- cs NS
  - cardinal.cs.wisc.edu, A, 128.105.14.122
  - www.cs.wisc.edu, A, 128.105.7.31
  - cs.wisc.edu, MX, granite.cs.wisc.edu
  - cs.wisc.edu, MX, obsidian.cs.wisc.edu
  - granite.cs.wisc.edu, A, 128.105.6.24
  - obsidian.cs.wisc.edu, A, 128.105.6.123
  - netlab-gateway.cs.wisc.edu, A, 128.105.214.163

DNS Name Resolution Algorithm

DNS Name Resolution Algorithm

- Step #1: Clients contract the Local Name Server
  - Local NS is provided to the client by DHCP or set in its configuration
  - Local NS contacts the root name server

DNS Name Resolution Algorithm

- Step #1: Clients contract the Local Name Server
  - Local NS is provided to the client by DHCP or set in its configuration
  - Local NS contacts the root name server

- Step #2: Root NS provided the NS
  - A record for NS that can resolve TLD

DNS Name Resolution Algorithm

- Step #1: Clients contract the Local Name Server
  - Local NS is provided to the client by DHCP or set in its configuration
  - Local NS contacts the root name server

- Step #2: Root NS provided the NS
  - A record for NS that can resolve TLD

- Step #3: Local NS contacts the NS for TLD

DNS Name Resolution Algorithm (cont'd)

- Step #4: NS for TLD provides NS
  - A record for NS that can resolve SLD

DNS Name Resolution Algorithm (cont'd)

- Step #4: NS for TLD provides NS
  - A record for NS that can resolve SLD

- Step #5: Local NS contacts NS for SLD

DNS Name Resolution Algorithm (cont'd)

- Step #4: NS for TLD provides NS
  - A record for NS that can resolve SLD
- Step #5: Local NS contacts NS for SLD
- Step #6: NS for SLD provides a record for the domain or NS
  - A record for NS that can resolve the domain

DNS Name Resolution Algorithm (cont'd)

- Step #4: NS for TLD provides NS
  - A record for NS that can resolve SLD

- Step #5: Local NS contacts NS for SLD

Local DNS server will cache any records it receives

Name Resolution Example

- Resolve netlab-gateway.cs.wisc.edu?

Name Resolution Example

- Resolve netlab-gateway.cs.wisc.edu?

- Resolve mail.google.com
  - Google NS
  - mail.google.com, CNAME, googlemail.1.google.com
  - googlemail.1.google.com, A, 74.125.225.53
  - googlemail.1.google.com, A, 74.125.225.54

Name Resolution Discussion

- Interactions# with NSs depending on
  - #1: How many parts are there to the DN, e.g., wisc.edu v.s. cs.wisc.edu?
  - #2: How many levels are in the name hierarchy from the same zone, e.g., wisc.edu and doit.wisc.edu are in the same zone, while wisc.edu and cs.wisc.edu are not?
  - #3: Whether there are CNAMEs that require contacting a different NS?
  - #4: What records does the local NS have already cached?
  - #5: A single local name server is shared by many clients, i.e., leveraging benefits of caching and reusing DNS replies

Name Resolution Optimization

- #1: NS can return different sets of records for different queries
  - Use for load balancing or geo-based server selection

- #2: Load balancing
  - Assign a short TTL to records
  - Return different A records for each query for a DN
  - Cycle through A records in weighted round-robin order
  - Weight is based on the server load

Name Resolution Optimization

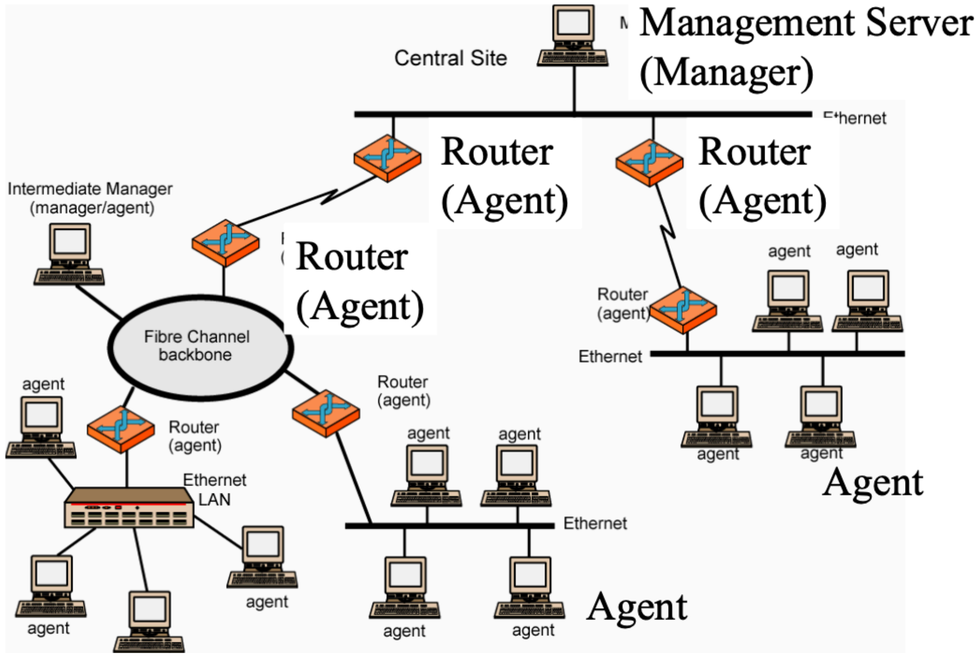
- #3: Geo-based server selection
  - Used by content distribution networks (CDNs)
  - CDN's NS is configured with the approximate geo-location of certain IP blocks
  - The address of the NS that issued the query is compared against IP blocks to determine the rough location of the client that issued the queries
  - Based on location, CDN's NS returns a CNAME record whose value is the DN for the server closest to a client

Name Resolution Optimization Example

- Example: google.com
  - Used by content distribution networks (CDNs)
  - #1: Local NS contacts root NS
  - #2: Root NS provides NS & A records for com NS
  - #3: Local NS contacts com NS
  - #4: Com NS provides NS & A record for Google NS, E.g., ns1.google.com, 216.239.32.10
  - #5: Local NS contacts Google NS
  - #6: Google NS looks at the source IP for the query and provides different addresses based on the estimated location of the source IP
    - From home: 64.15.120.52
    - From Milwaukee (AT&T DSL): 74.125.225.32
    - From Los Angeles: 74.125.239.161

DNS simplifies naming and addressing.

Why SNMP?


Central Site
Management Server
(Manager)
Router
(Agent)
Router
(Agent)
Intermediate Manager
(manager/agent)
Router
(Agent)
Fibre Channel
backbone
Router
(agent)
Router
(agent)
Router
(agent)
Ethernet
LAN
Ethernet
Ethernet
agent
Agent
Agent

Why SNMP?

Ticket #1: I cannot access the printer;
Ticket #2: I cannot access the file server;
Ticket #3: Why it took me 10mins to print a 1-page document?
Ticket #4: I sent an email to my colleague on another floor, which takes half an hour;

What is Network Management?

- #1: Configuration management
  - Keep track of device settings and how they function
- #2: Fault management
  - Dealing with problems and emergencies in the network
- #3: Performance management
  - How smoothly is the network running?
  - Can it handle the workload it currently has?

Network Management Goals

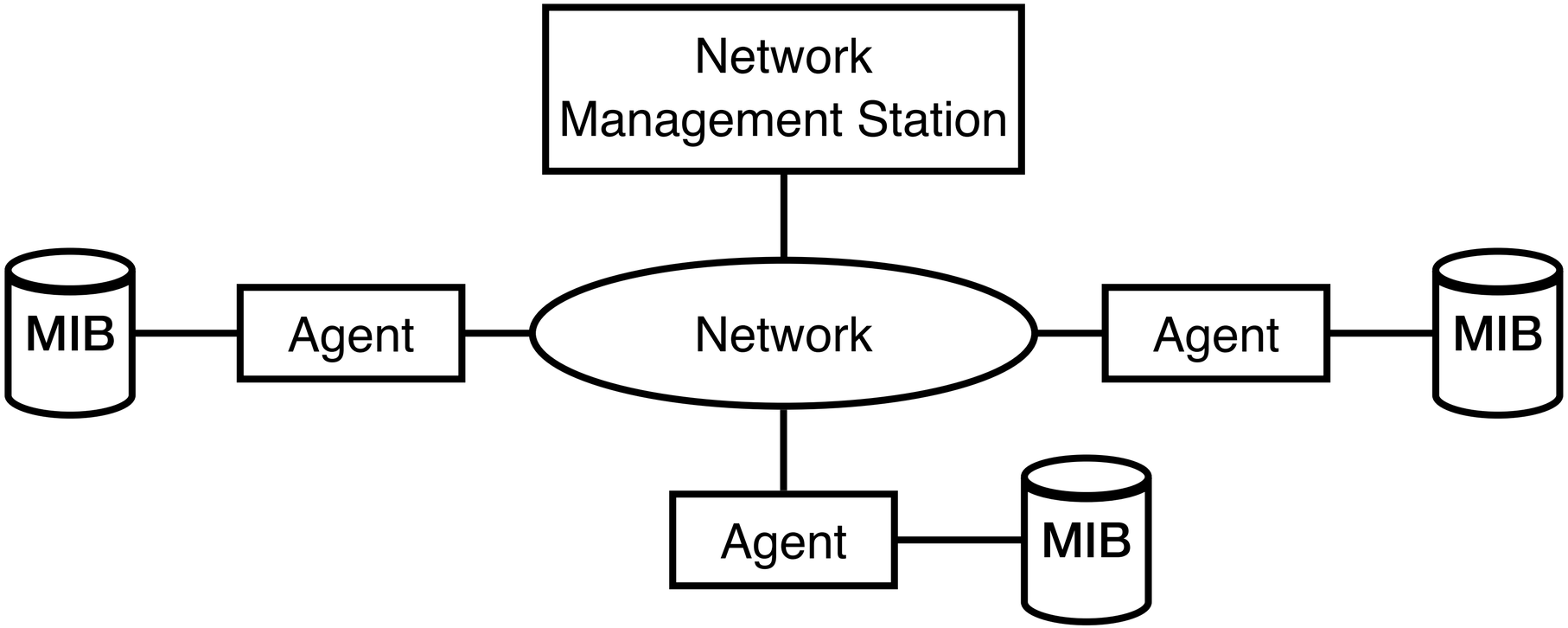
- The management interface must be
  - Standardized
  - Extensible
  - Portable

- The management mechanism must be
  - Inexpesive
  - Implemented as software only

Network Management Overview

- Management = Initialization, Monitoring, Control

- Manager, Agenits, and Management Information Base (MIB)



SNMP

- Based on the Simple Gateway Management Protocol (SGMP)
  - RFC 1028, Nov. 1987

- SNMP: Simple Network Management Protocol
  - A tool (protocol) that allows for remote and local management of items on the network, including servers, workstations, routers, switches, and other managed devices
  - RFC 1058, April 1988

SNMP Interface

- Only five commands

Command	Meaning
get-request	Fetch a value
get-next-request	Fetch the next value (in a tree)
get-response	Reply to fetch operation
set-request	Store a value
trap	An event

SNMP Advantages

#1: Standardized

#2: Universally supported

#3: Extendible

#4: Portable

#5: Allows distributed management access

#6: Light-weighted

Client Pull & Server Push

- SNMP is a “client pull” model
  - The management system (client) “pulls” data from the agent (server)

- SNMP is a “server pull” model
  - The agent (server) “pushes” out a trap message to a (client) management system

Ports & UDP

- SNMP uses UDP as the transport mechanism

- SNMP uses two well-known ports to operate
  - UDP Port 161: SNMP messages
  - UDP Port 152: SNMP trap messages

SNMP Details

- SNMP Protocol
  - Define the format of the message exchanged by management systems and agents
  - Specify the GET, GETNext, Set, and Trap operations
- Structure of Mangement Information (SMI)
  - Rules specifying the format used to define objects managed on the network that the SNMP protocol accesses
- Management Information Base (MIB)
  - A map of the hierarchical order of all managed objects and how they are accessed

Nodes

- Items in an SNMP Network are called nodes
  - There are different types
- Type #1: Managed nodes
  - Run an agent process that services requests from a managment node
- Type #2: Management nodes
  - A workstation running some network management & monitoring software
- Type #3: Nodes that are not manageable by SNMP
  - A node may not support SNMP but may be managed by SNMP through a proxy agent running on another machine

Community Names

- Community names are used to define where an SNMP message is destined for

- They mirror the same concept as a Windows or Unix domain
  - Set up your agents belong to certain communities
  - Set up your management applications to monitor and receive traps from certain community names

SNMP Agents

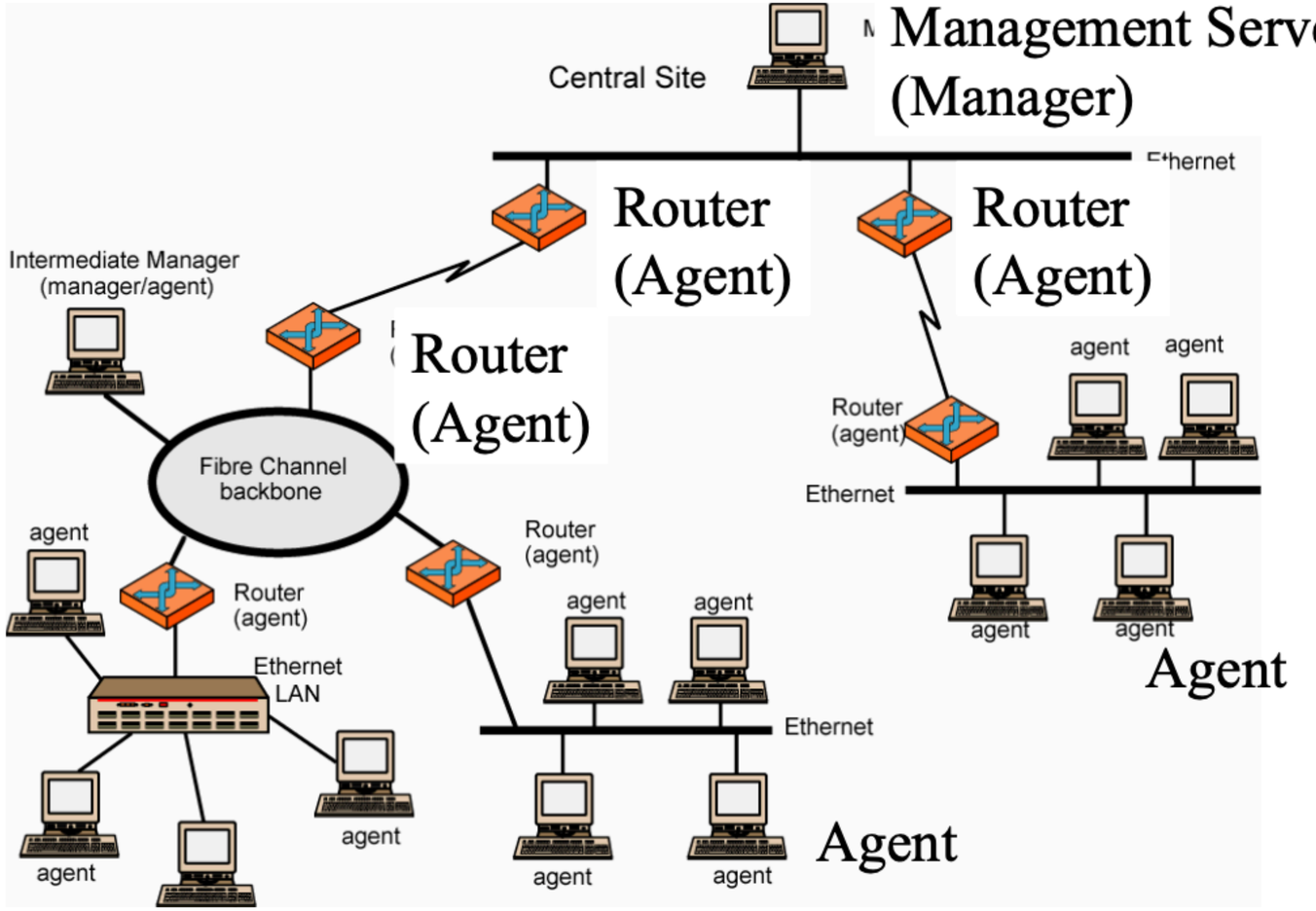
- Extensible agents
  - Open, modular design follows for adaptations to new management data and operational requirements

- Monolithic agents
  - Not extensible
  - Optimized for specific hardware platform and OS

SNMP Semantics

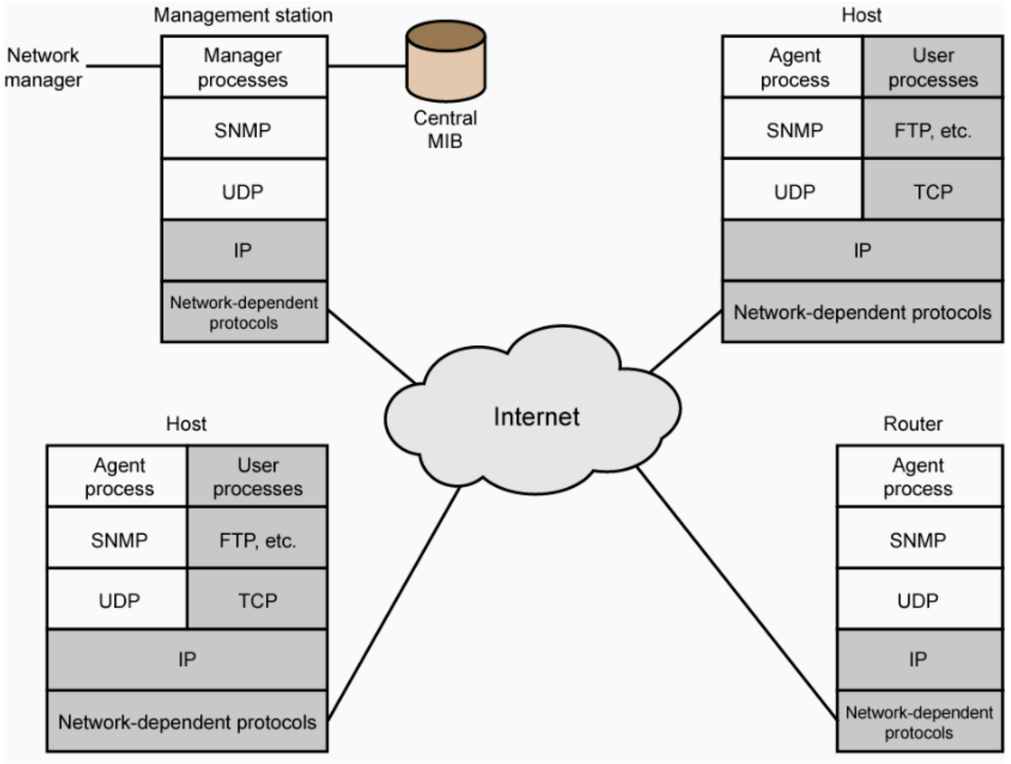
- Structure of Management Information (SMI)
  - Specify the format used for defining managed objects that are accessed via the SNMP protocol
- Abstract Syntax Notation One (ASN.1)
  - Used to define the format of SNMP messages and managed objects (MIB) modules using an unambiguous data description format
- Basic Encoding Rules (BER)
  - Used to encode the SNMP messages into a format suitable for transmission across a network

Example of Network Management

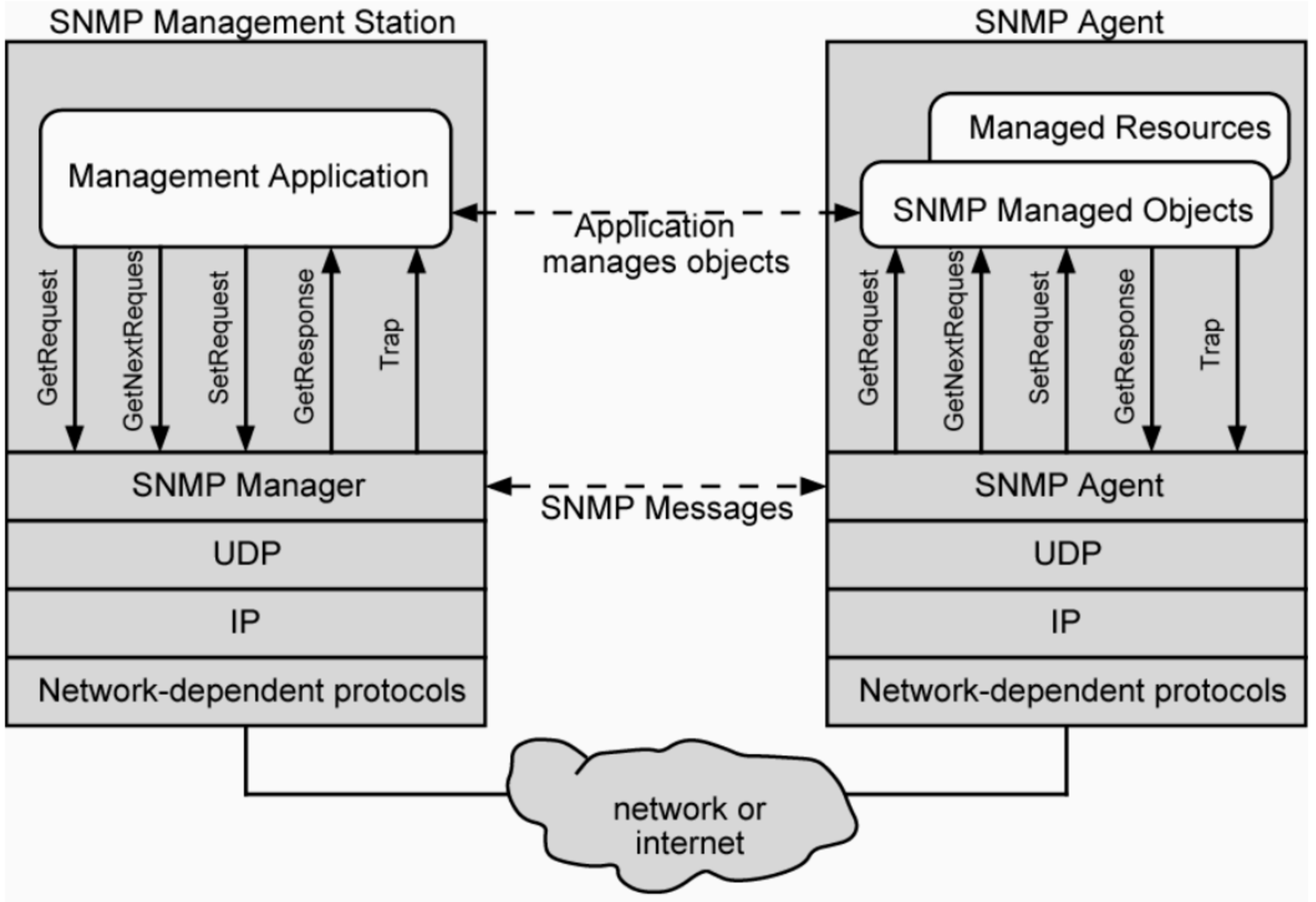


SNMPv1 Configuration

- Managers send requests to the UDP port 161
- Agents send traps to the UDP port 162



Role of SNMPv1



SNMPv2 (RFC 1441)

- Improved security: authentication and integrity using the Data Encryption Standard (DES)
- Information request => Multiple manager coordination
  - A locking mechanism prevents multiple managers from writing simultaneously
- Confirmation option for traps
  - Agents can ensure that the traps were received correctly
- New error codes
  - noSuchName, badValue, readOnly

SNMPv3 (RFC 2570)

- Security update of SNMPv2

- Authentication: message authentication code with a shared secret key

- Privacy: encryption using a shared secret key

- Access control: each manager can have a different set of read/write permission for various component of MIB

SNMP simplifies networking monitoring.

Infrastructure Service Application

- Distributed applications
  - Functionality:
  - System design goals: fault tolerance, scalability, and performance
- System deployment
  - The role of different participation nodes
- Data partition and layout
  - Data format and consistency requirements
- Execution
  - Communication requirement
  - Coordination logics

Summary

- Today
  - Linux Networking Stack
  - Infrastructure services

- Next lecture
  - Lab4 review
  - Quiz 4