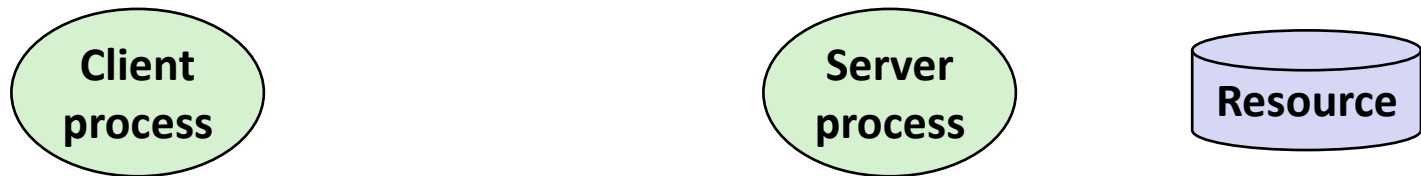


# **Network Programming:**

## ***Overview: From hardware to software***

# A Client-Server Transaction

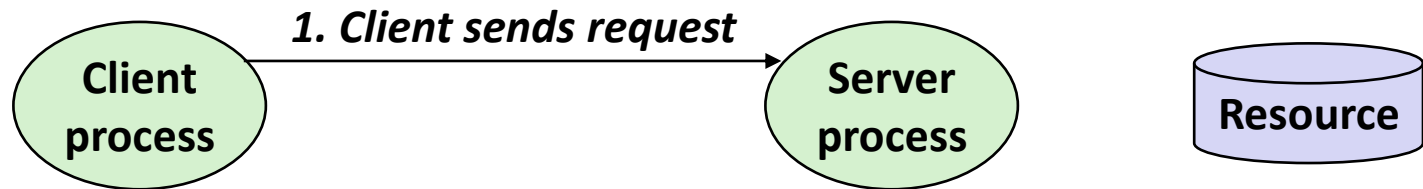
- Most network applications are based on the client-server model:
  - A **server** process and one or more **client** processes
  - Server manages some **resource**
  - Server provides **service** by manipulating resource for clients
  - Server activated by request from client (vending machine analogy)



*Note: clients and servers are processes running on hosts  
(can be the same or different hosts)*

# A Client-Server Transaction

- Most network applications are based on the client-server model:
  - A **server** process and one or more **client** processes
  - Server manages some **resource**
  - Server provides **service** by manipulating resource for clients
  - Server activated by request from client (vending machine analogy)

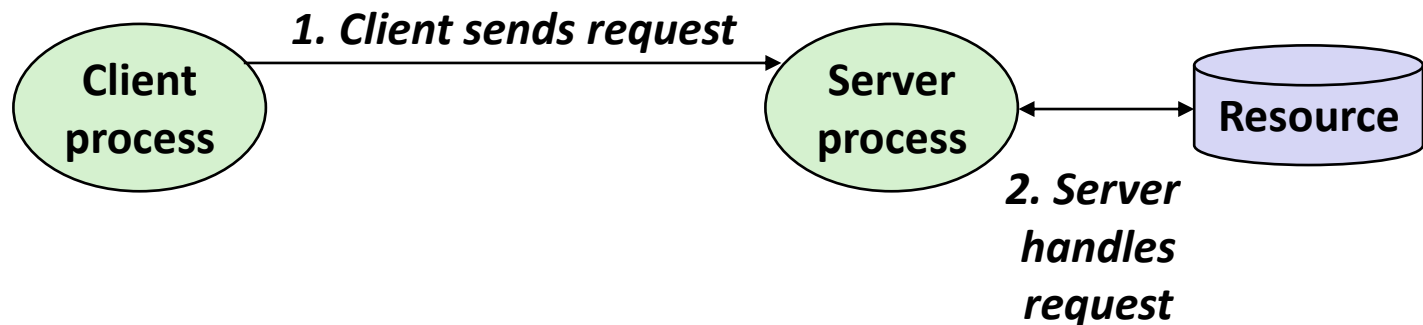


*Note: clients and servers are processes running on hosts  
(can be the same or different hosts)*

# A Client-Server Transaction

## ■ Most network applications are based on the client-server model:

- A **server** process and one or more **client** processes
- Server manages some **resource**
- Server provides **service** by manipulating resource for clients
- Server activated by request from client (vending machine analogy)

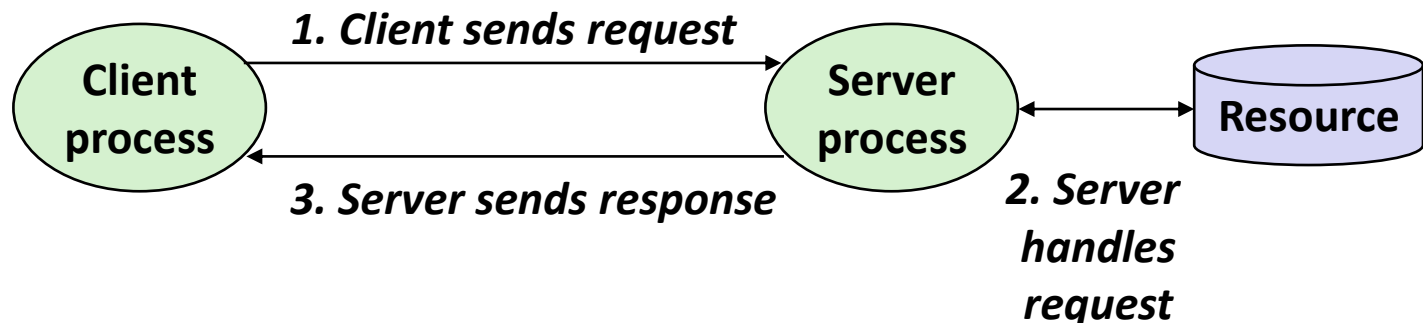


*Note: clients and servers are processes running on hosts  
(can be the same or different hosts)*

# A Client-Server Transaction

## ■ Most network applications are based on the client-server model:

- A **server** process and one or more **client** processes
- Server manages some **resource**
- Server provides **service** by manipulating resource for clients
- Server activated by request from client (vending machine analogy)

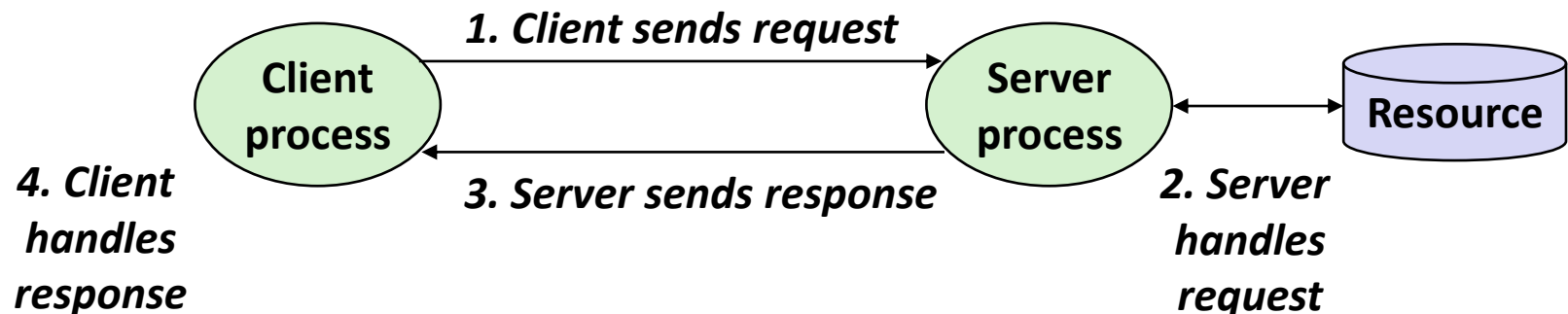


*Note: clients and servers are processes running on hosts  
(can be the same or different hosts)*

# A Client-Server Transaction

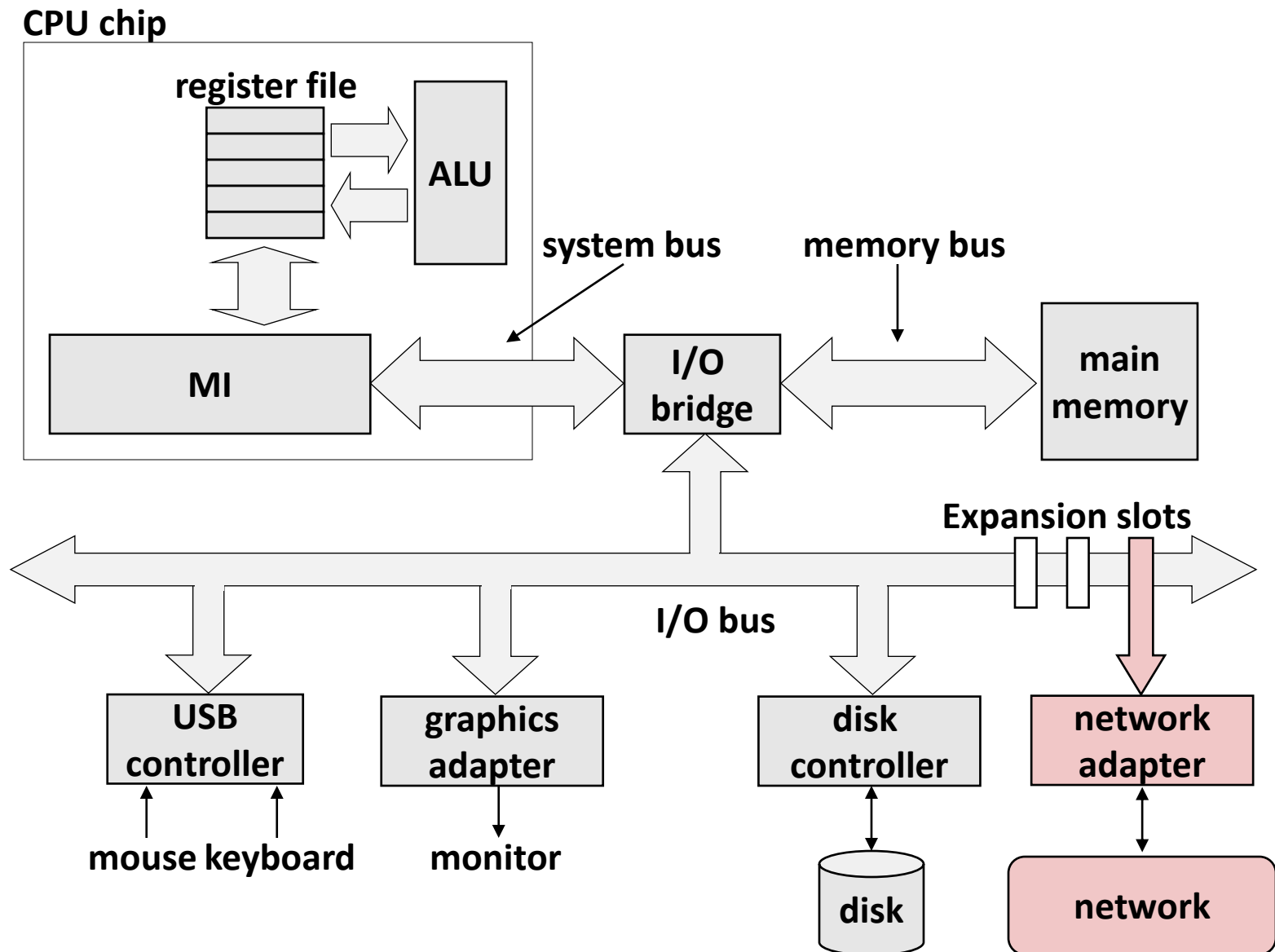
## ■ Most network applications are based on the client-server model:

- A **server** process and one or more **client** processes
- Server manages some **resource**
- Server provides **service** by manipulating resource for clients
- Server activated by request from client (vending machine analogy)



*Note: clients and servers are processes running on hosts  
(can be the same or different hosts)*

# Hardware Organization of a Network Host

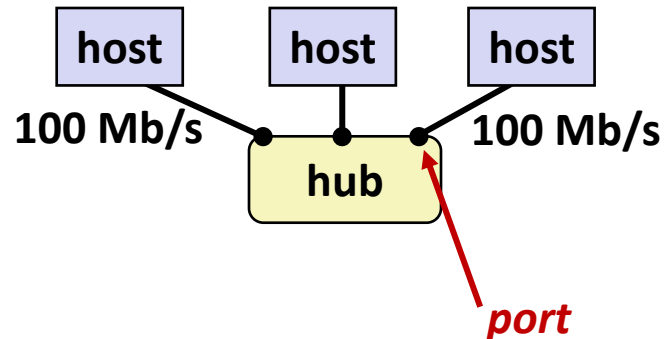


# Computer Networks

- A ***network*** is a hierarchical system of boxes and wires organized by geographical proximity
  - SAN (System Area Network) spans cluster or machine room
    - Switched Ethernet, Quadrics QSW, ...
  - LAN (Local Area Network) spans a building or campus
    - Ethernet (hubs) is most prominent example
  - WAN (Wide Area Network) spans country or world
    - Typically high-speed point-to-point phone lines
- An ***internetwork*** (*internet*) is an interconnected set of networks
  - The Global IP Internet (uppercase “I”) is the most famous example of an internet (lowercase “i”)
- Let's see how an internet is built from the ground up

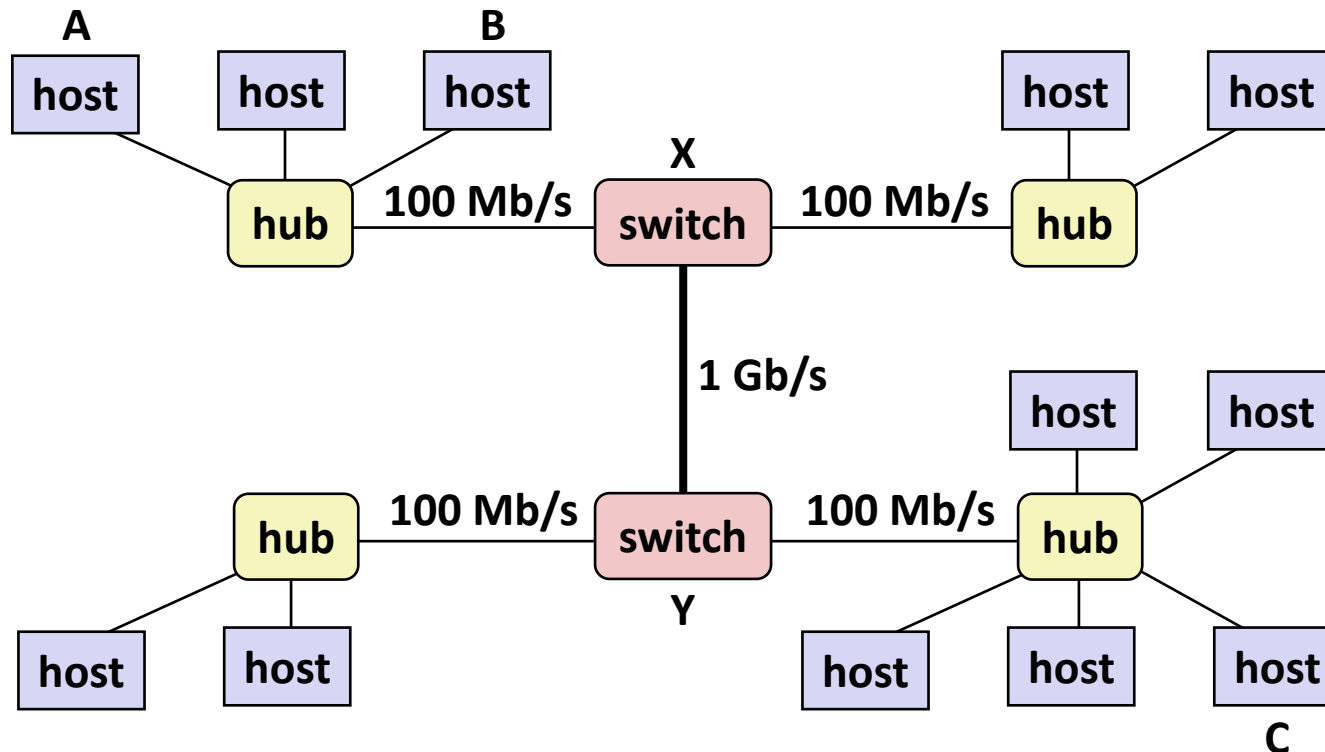


# Lowest Level: Ethernet Segment



- Ethernet segment consists of a collection of *hosts* connected by wires (twisted pairs) to a *hub*
- Spans room or floor in a building
- Operation
  - Each Ethernet adapter has unique 48-bit MAC (Media Access Control) address
    - E.g., 00:16:ea:e3:54:e6
  - Hosts send bits to any other host in chunks called *frames*
  - Hub slavishly copies each bit from each port to every other port
    - Every host sees every bit
    - Note: Hubs are on their way out. Switches (bridges) cheap enough to replace them

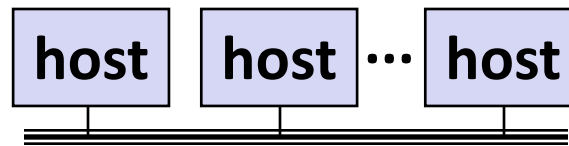
# Next Level: Switched Ethernet Segment



- Spans building or campus
- Switches learn which hosts (MAC@) are reachable from which ports and selectively copy frames from port to port

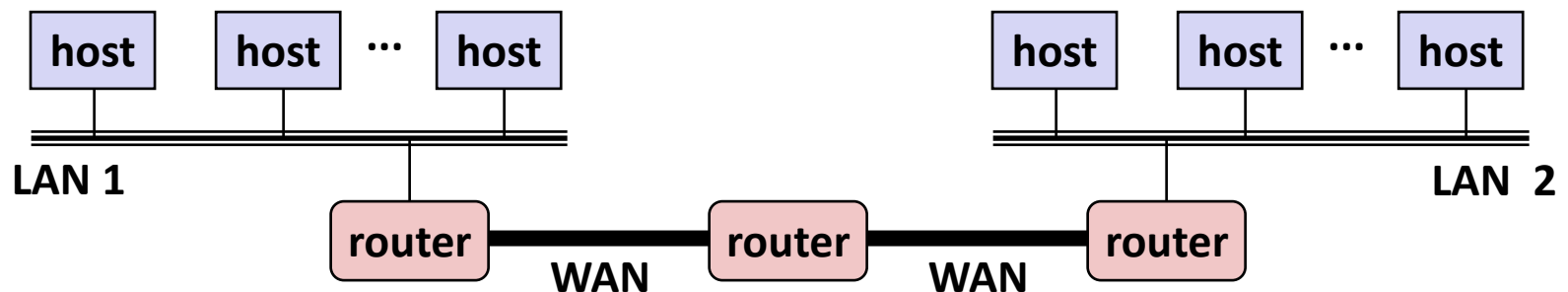
# Conceptual View of LANs

- For simplicity, hubs, switches, and wires are often shown as a collection of hosts attached to a single wire:



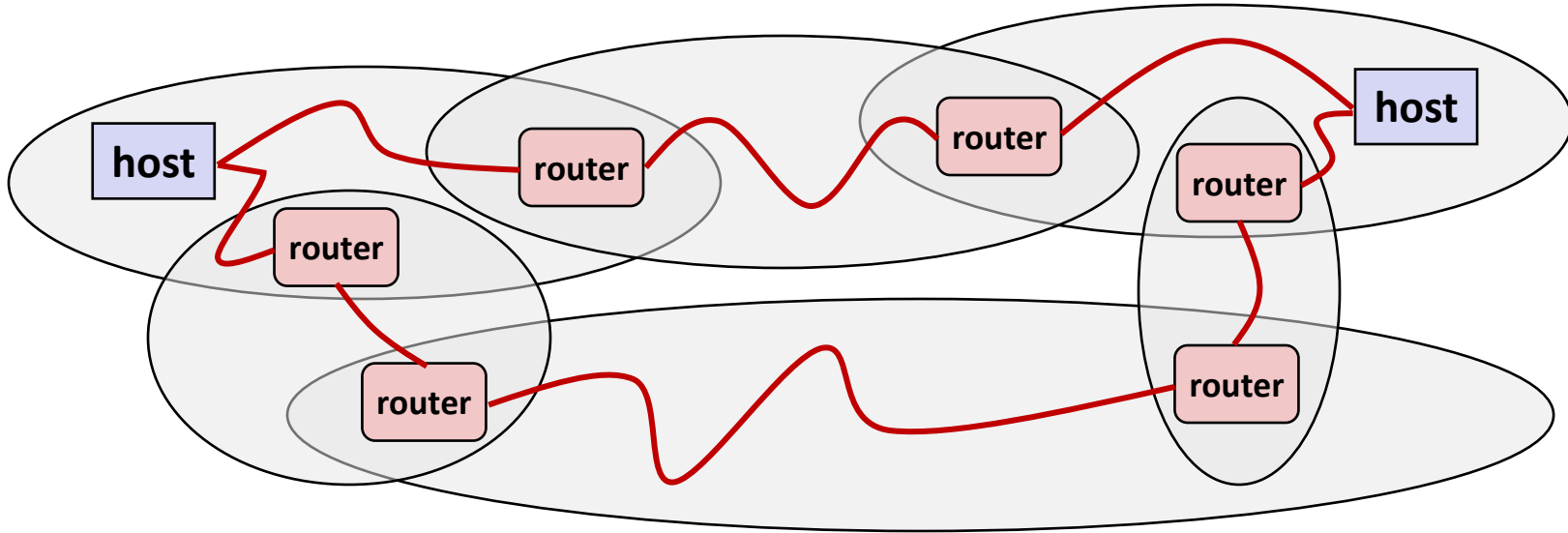
# Next Level: internets

- Multiple incompatible LANs can be physically connected by specialized computers called **routers**
- The connected networks are called an **internet** (lower case)



*LAN 1 and LAN 2 might be completely different, totally incompatible (e.g., Ethernet, Fibre Channel, 802.11\*, T1-links, DSL, ...)*

# Logical Structure of an internet



- **Ad hoc interconnection of networks**
  - No particular topology
  - Vastly different router & link capacities
- **Send packets from source to destination by hopping through networks**
  - Router forms bridge from one network to another
  - Different packets may take different routes

# The Notion of an internet Protocol

- How is it possible to send bits across incompatible LANs and WANs?
- Solution: *protocol* software running on each host and router
  - Protocol is a set of rules that governs how hosts and routers should cooperate when they transfer data from network to network.
  - Smooths out the differences between the different networks

# What Does an internet Protocol Do?

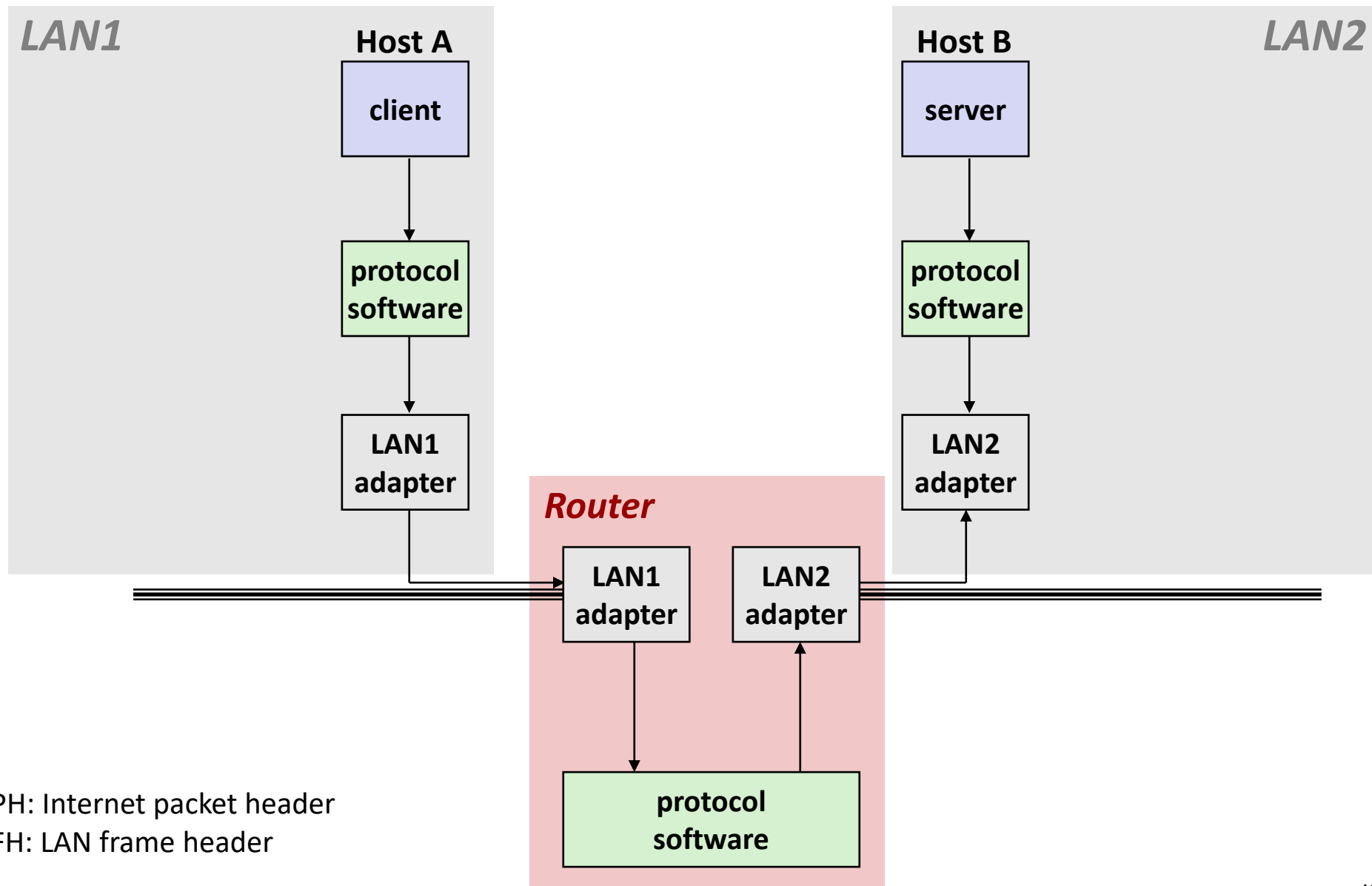
## ■ Provides a *naming scheme*

- An internet protocol defines a uniform format for *host addresses*
- Each host (and router) is assigned at least one of these internet addresses that uniquely identifies it

## ■ Provides a *delivery mechanism*

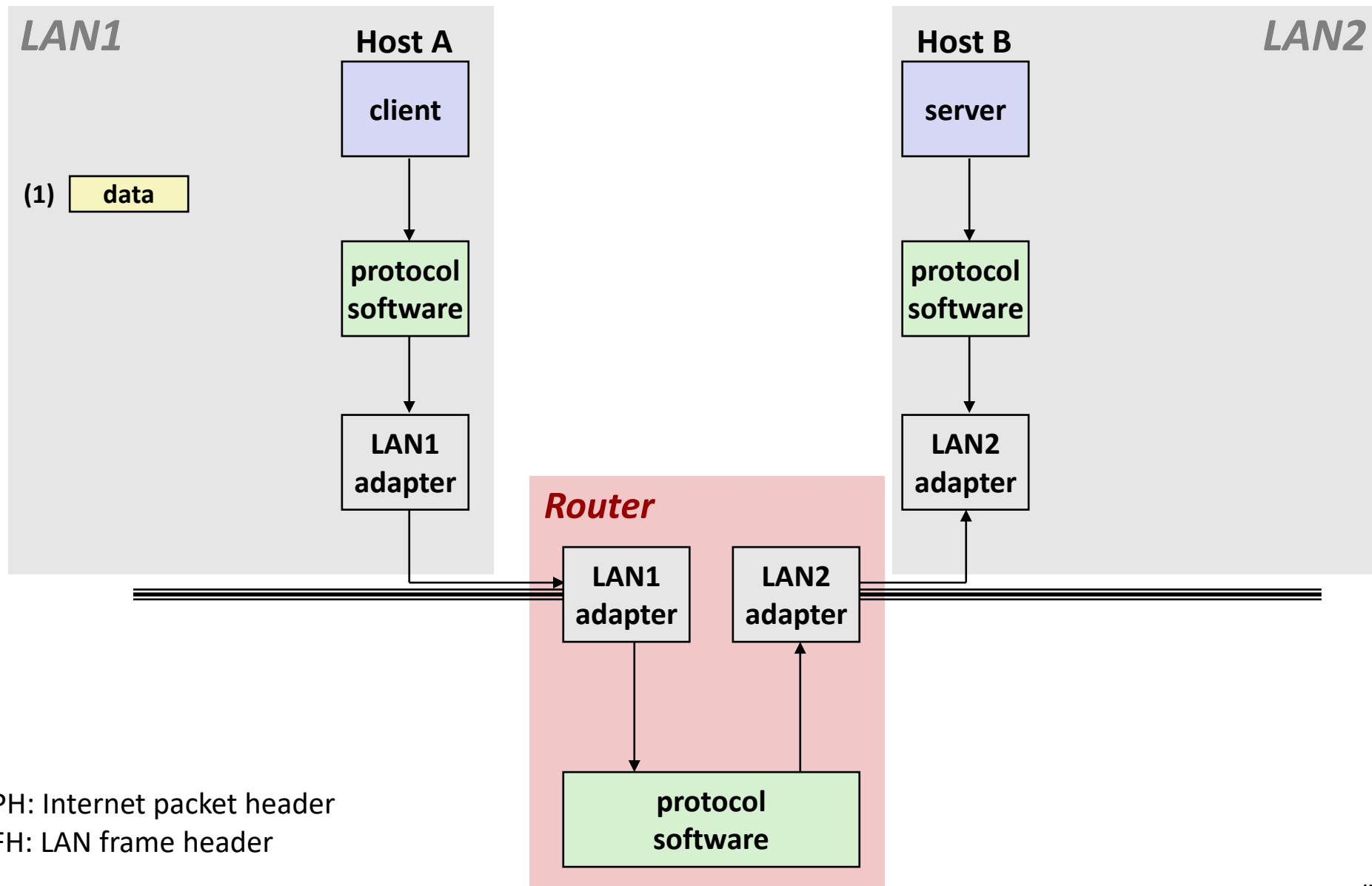
- An internet protocol defines a standard transfer unit (*packet*)
- Packet consists of *header* and *payload*
  - Header: contains info such as packet size, source and destination addresses
  - Payload: contains data bits sent from source host

# Transferring internet Data Via Encapsulation

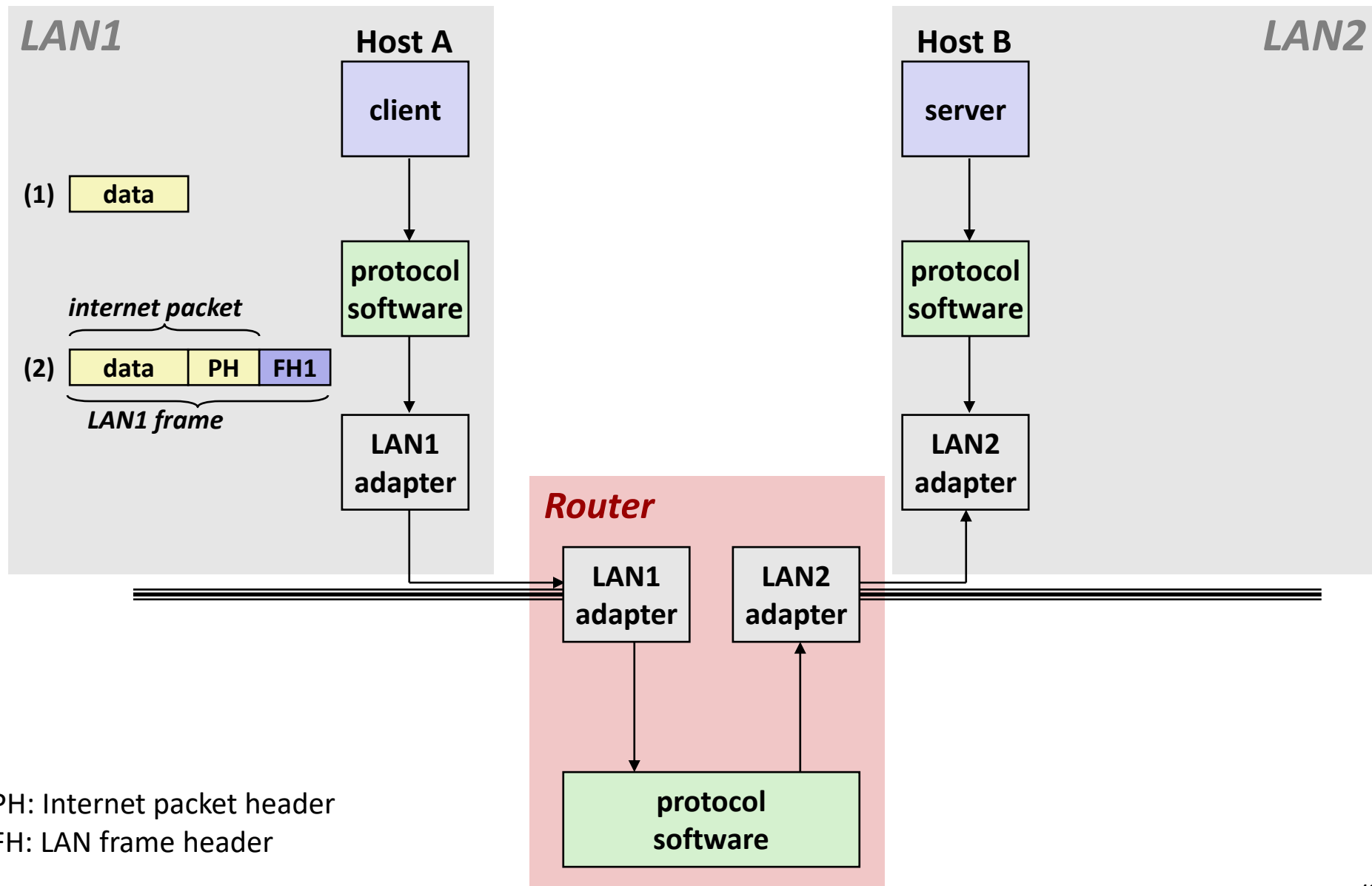




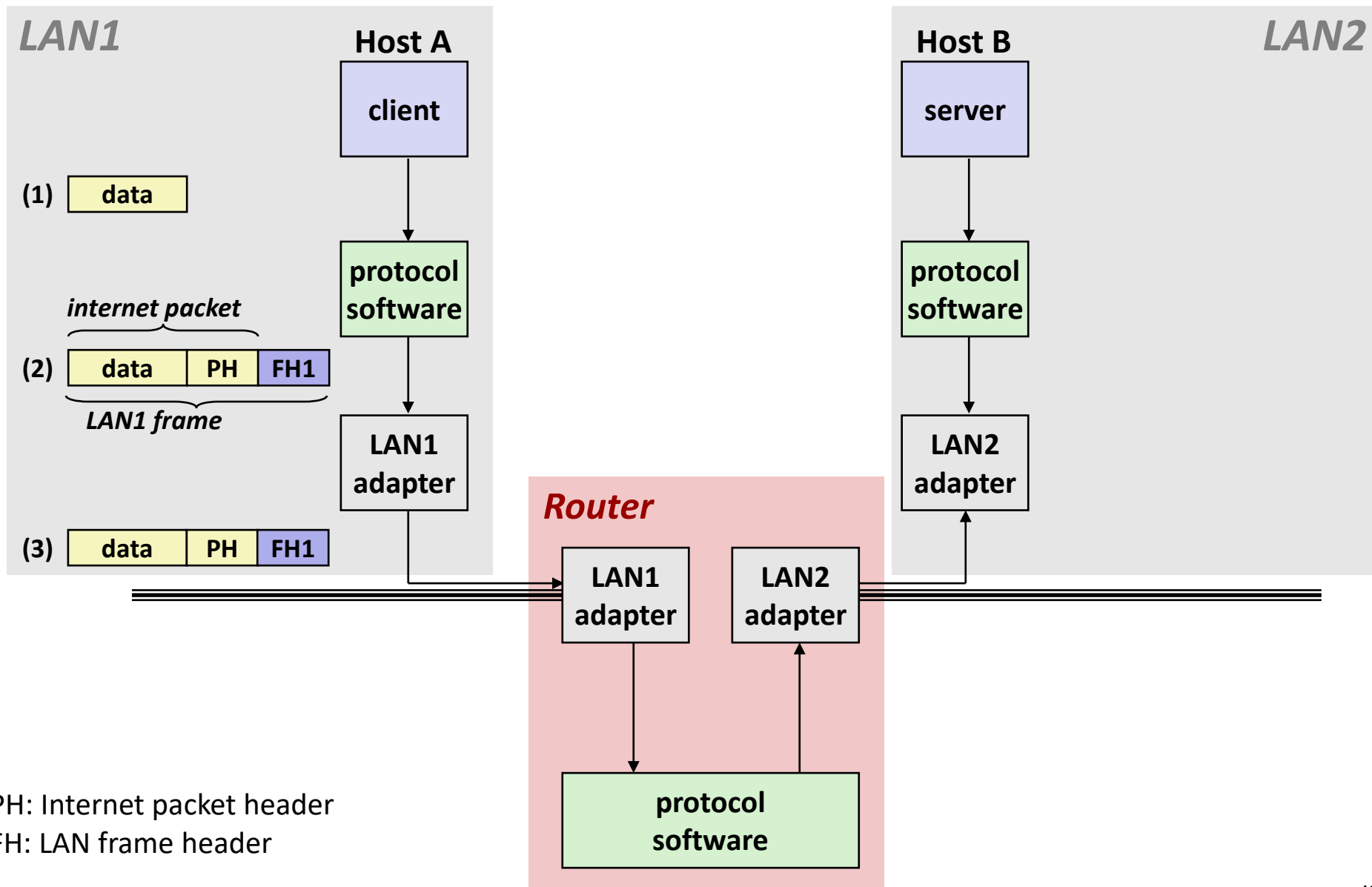
# Transferring internet Data Via Encapsulation



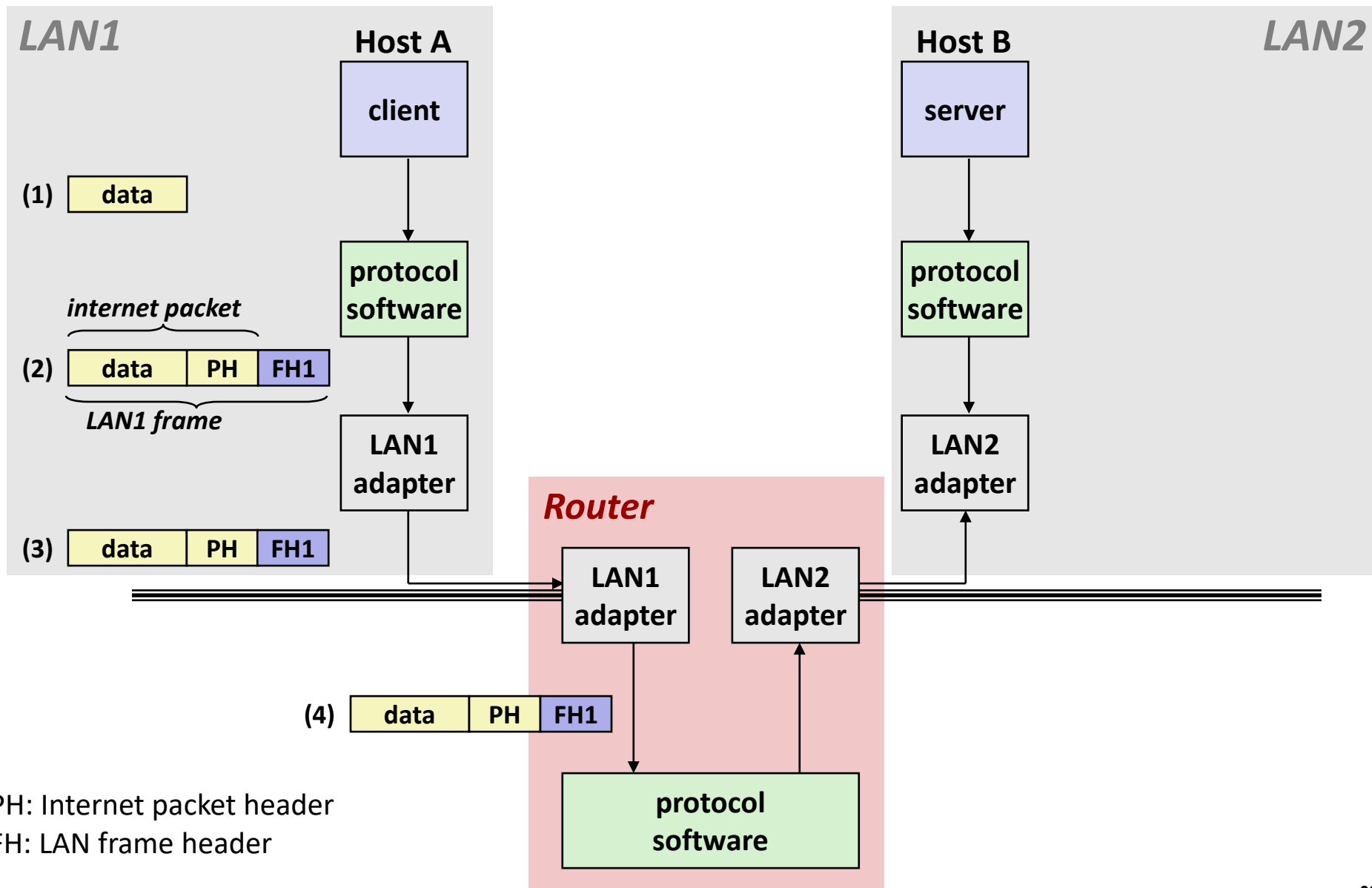
# Transferring internet Data Via Encapsulation



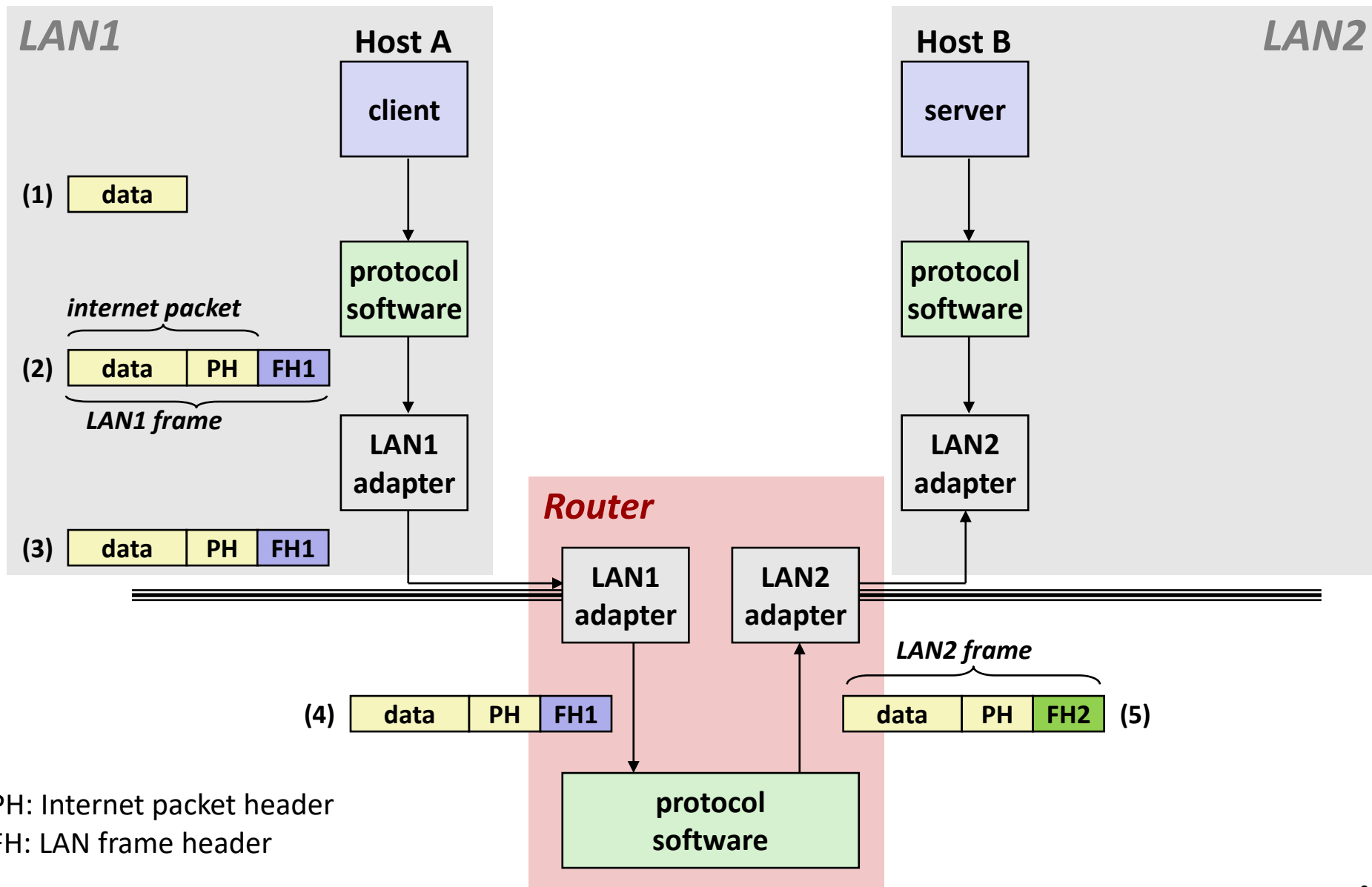
# Transferring internet Data Via Encapsulation



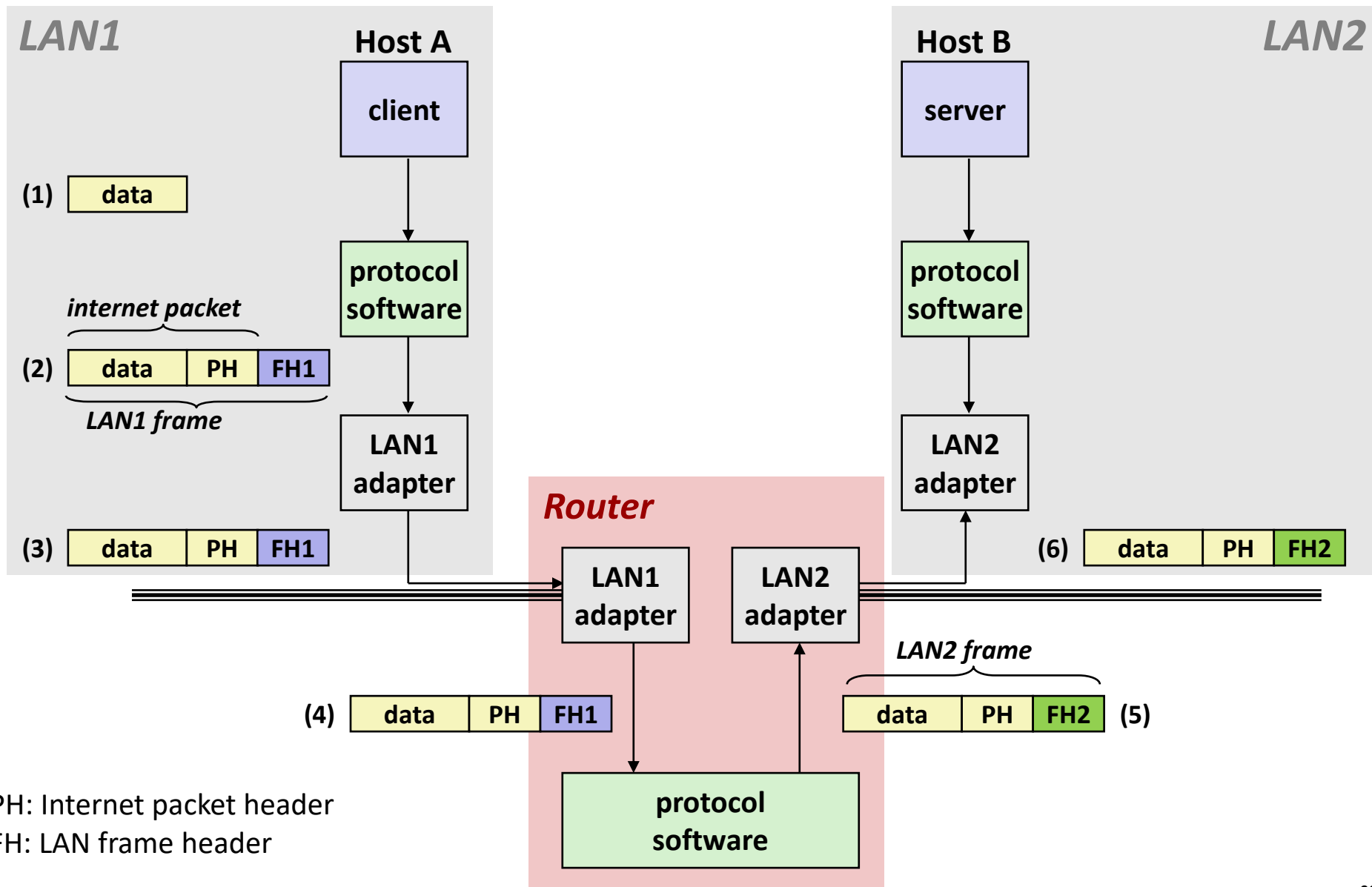
# Transferring internet Data Via Encapsulation



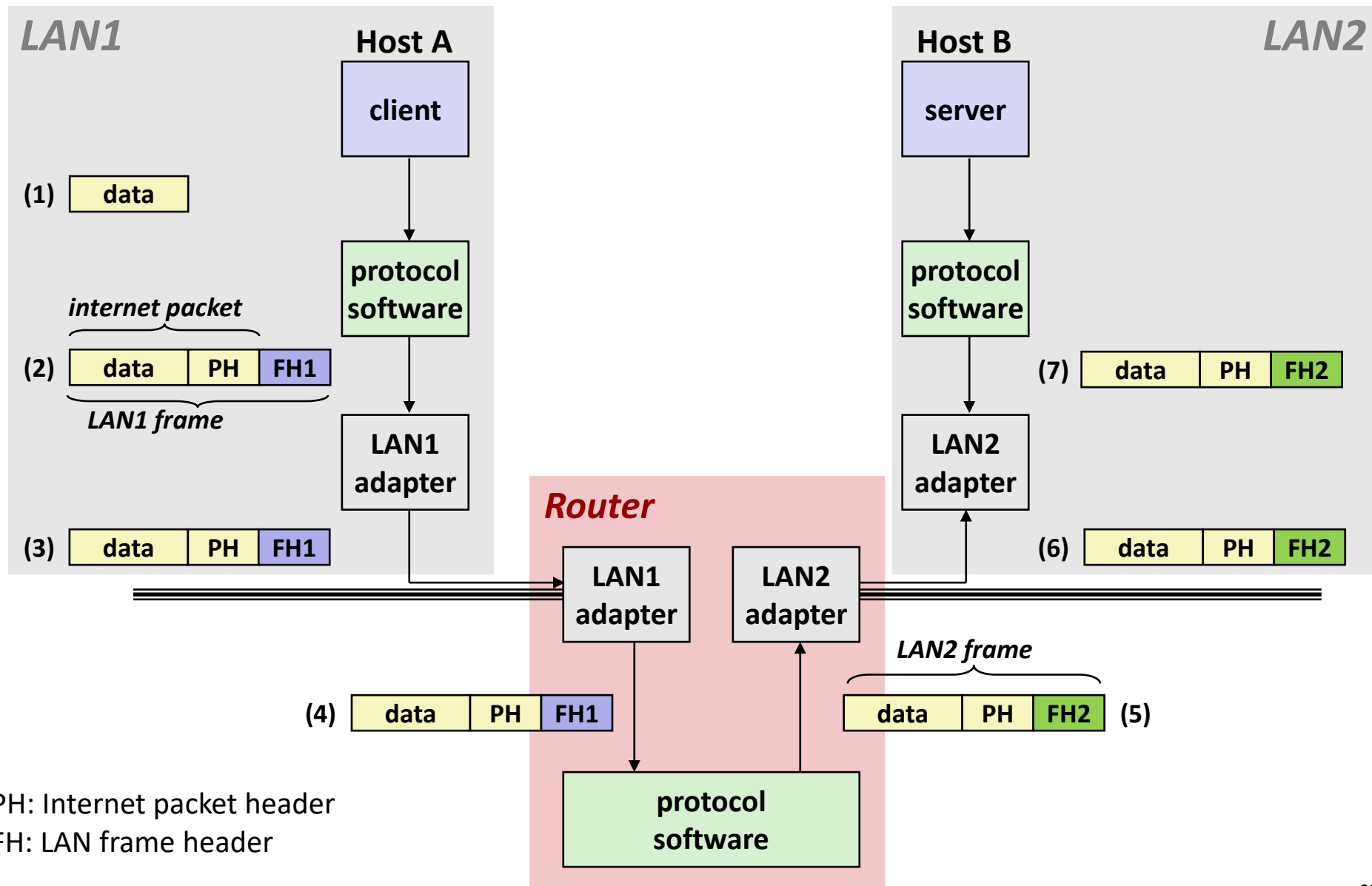
# Transferring internet Data Via Encapsulation



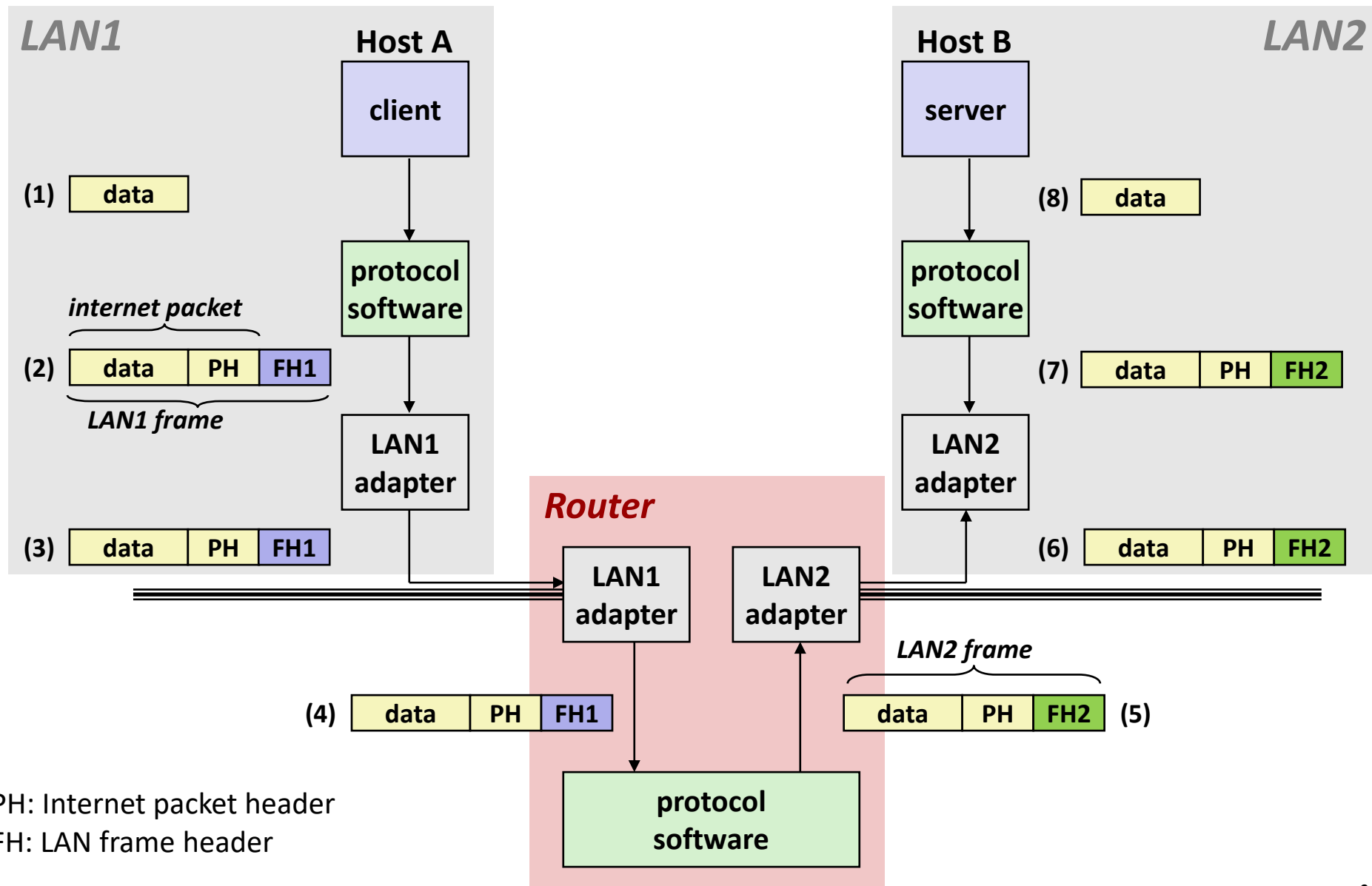
# Transferring internet Data Via Encapsulation



# Transferring internet Data Via Encapsulation



# Transferring internet Data Via Encapsulation





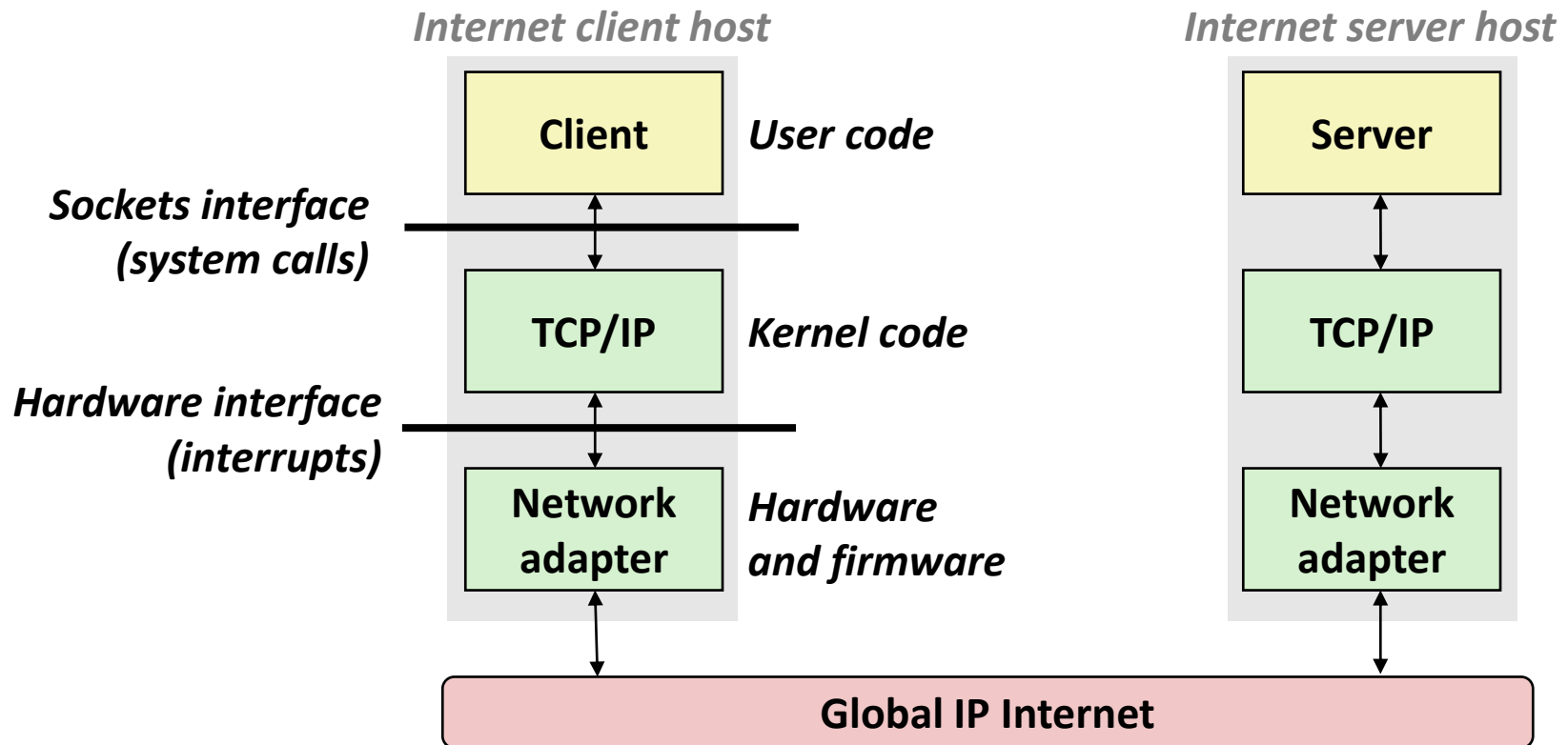
# Other Issues

- **We are glossing over a number of important questions:**
  - What if different networks have different maximum frame sizes? (segmentation)
  - How do routers know where to forward frames?
  - How are routers informed when the network topology changes?
  - What if packets get lost?
  
- **These (and other) questions are addressed by the area of systems known as *computer networking***

# Global IP Internet (upper case)

- Most famous example of an internet
- Based on the TCP/IP protocol family
  - IP (Internet Protocol) :
    - Provides *basic naming scheme* and unreliable *delivery capability* of packets from *host-to-host*
  - UDP (User Datagram Protocol)
    - Uses IP to provide *unreliable* datagram delivery from *process-to-process* without prior *connection*
  - TCP (Transmission Control Protocol)
    - Uses IP to provide *reliable* byte streams from *process-to-process* over *connections*
- Accessed via a mix of Unix file I/O and functions from the *sockets interface*

# Hardware and Software Organization of an Internet Application



# OSI (Open Systems Interconnection) model

- Conceptual model
- Loose connection with *Internet Protocol Suite* (“layering considered harmful” RFC 3439)

<i>n</i>	Name	Protocol Data Unit	Protocols (examples)
7	Application	Data	HTTP
6	Presentation		SSL, SSH, FTP
5	Session		TCP (sockets)
4	Transport	Segment Datagram	TCP UDP
3	Network	Packet	IP
2	Data link	Frame	<i>MAC-based</i> Ethernet, 802.11
1	Physical	Bit	

# A Programmer's View of the Internet

## 1. Hosts are mapped to a set of 32/128-bit *IP addresses*

- 140.192.193.60 (IPv4)
- 2a03:2880:f127:83:face:b00c:0:25de (IPv6)

## 2. The set of IP addresses is mapped to a set of identifiers called Internet *domain names*

- 140.192.193.60 is mapped to `www.cdm.depaul.edu`

## 3. A process on one Internet host can communicate with a process on another Internet host over a *connection*