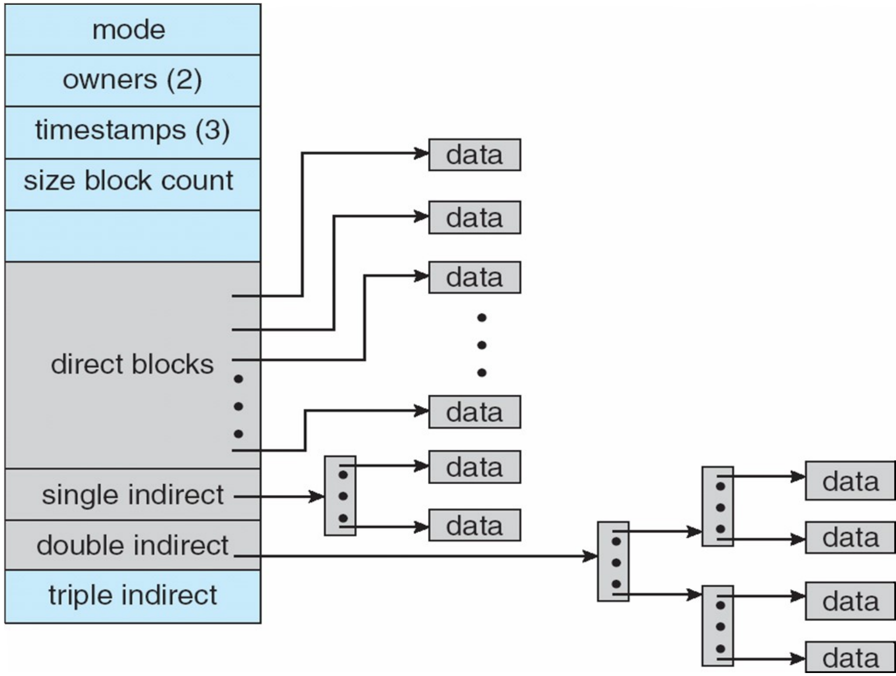
Filesystem

Disclaimer: some slides are adopted from book authors' slides with permission

Recap: Ext2

- Inode
  - 12 blocks are directly mapped, 1 indirect pointer. 1 double indirect pointer, 1 triple indirect pointer
- Maximum file size?
  - $min(((b/4)^3+(b/4)^2+b/4+12)*b, (2^{32}-1)*512)$
  - 1K block size
    - 1K * (12 + 256 + 256^2 + 256^3) = 16GB
  - 2K block size
    - 2K * (12 + 512+ 512^2 + 512^3) = 256G
  - 4K block size
    - (2^32-1)*512 = 2TB



Journaling

- What happens if you lost power while updating to the filesystem?
  - Example
    - Create many files in a directory
    - System crashed while updating the directory entry
    - All new files are now “lost”
  - Recovery (fsck)
    - May not be possible
    - Even if it is possible to a certain degree, it may take very long time

Journaling

- Idea
  - First, write a log (journal) that describes all changes to the filesystem, then update the actual filesystem sometime later
- Procedure
  - Begin transaction
  - Write changes to the log (**journal**)
  - End transaction (**commit**)
  - At some point (**checkpoint**), synchronize the log with the filesystem

KU THE UNIVERSITY OF KANSAS

Recovery in Journaling Filesystems

- Check logs since the last checkpoint
- If a transaction log was committed, apply the changes to the filesystem
- If a transaction log was not committed, simply ignore the transaction

THE UNIVERSITY OF KANSAS

Types of Journaling

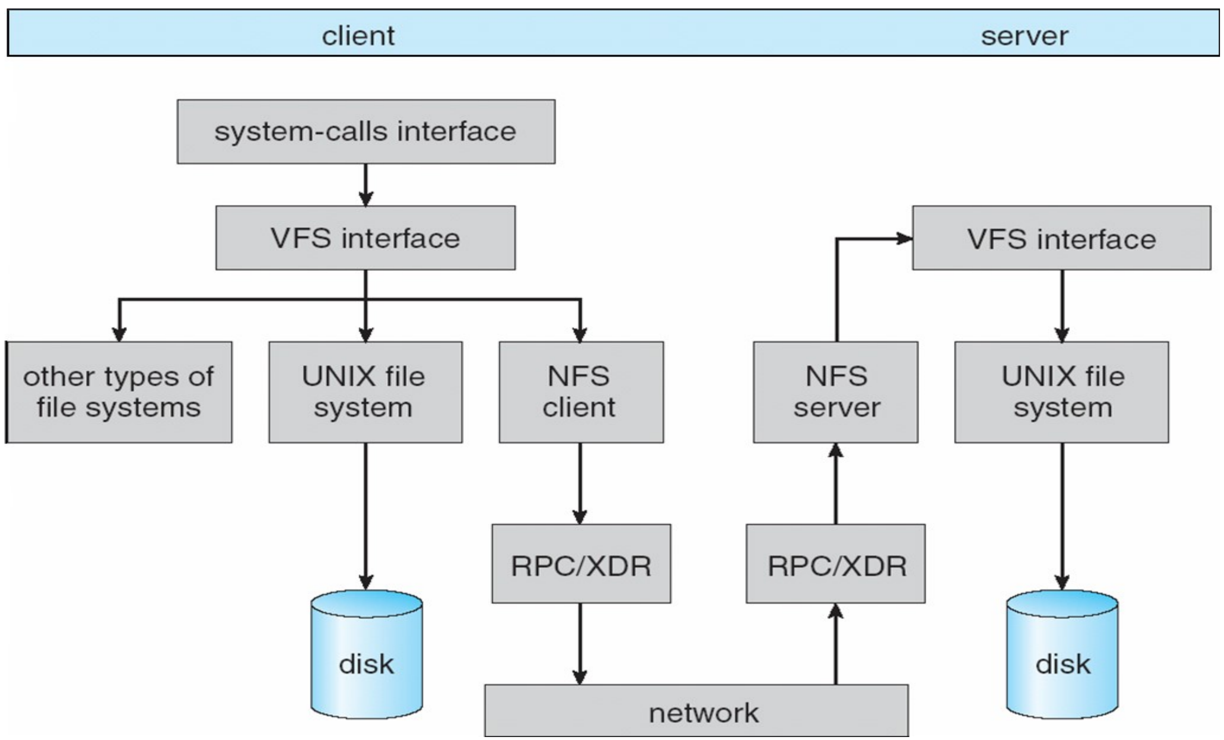
- Full journaling
  - All data & metadata are written twice
- Metadata journaling
  - Only write metadata of a file to the journal

Ext3 Filesystem

- Ext3 = Ext2 + Journaling
- Journal is stored in a special file
- Supported journaling modes
  - Write-back (metadata journaling)
  - Ordered (metadata journaling)
    - Data blocks are written to disk first
    - Metadata is written to journal
  - Data (full journaling)
    - Data and metadata are written to journal

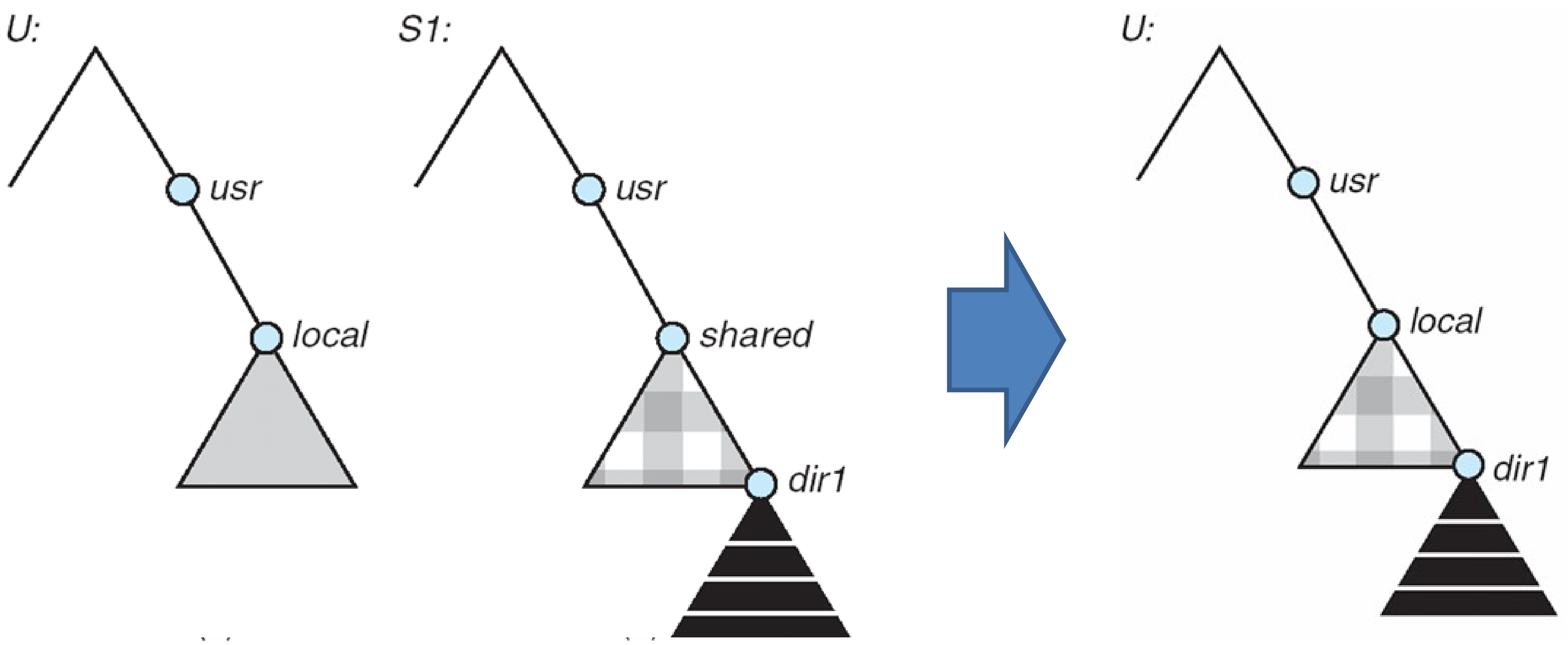
Network File System (NFS)

- Developed in mid 80s by Sun Microsystems
- RPC based server/client architecture
- Attach a remote filesystem as part of a local filesystem



NFS Mounting Example

- Mount S1:/usr/share /usr/local



NFS vs. Dropbox

- NFS
  - All data is stored in a remote server
  - Client doesn’t have any data on its local storage
  - Network failure → no access to data
- Dropbox
  - Client store data in its own local storage
  - Differences between the server and the client are exchanges to synchronize
  - Network failure → still can work on local data. Changes are synchronized when the network is recovered
- Which approach do you like more and why?

Summary

- I/O mechanisms
- Disk
- Disk allocation methods
- Directory
- Caching
- Virtual File System
- FAT and Ext2 filesystem
- Journaling
- Network filesystem (NFS)

KU THE UNIVERSITY OF KANSAS

Distributed Systems

Roadmap

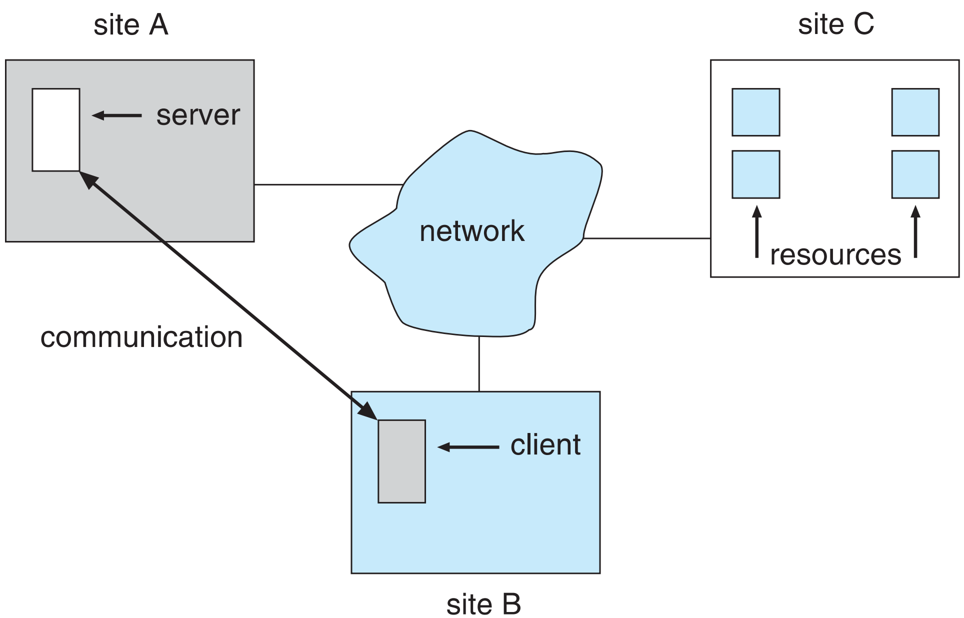
- CPU management
- Memory management
- Disk management
- **Distributed System**
- Protection & Security
- Virtual machine

Today

- Distributed systems overview
- Basic network concepts
- TCP/IP protocol
- Sending/Receiving a packet in Linux

Distributed Systems

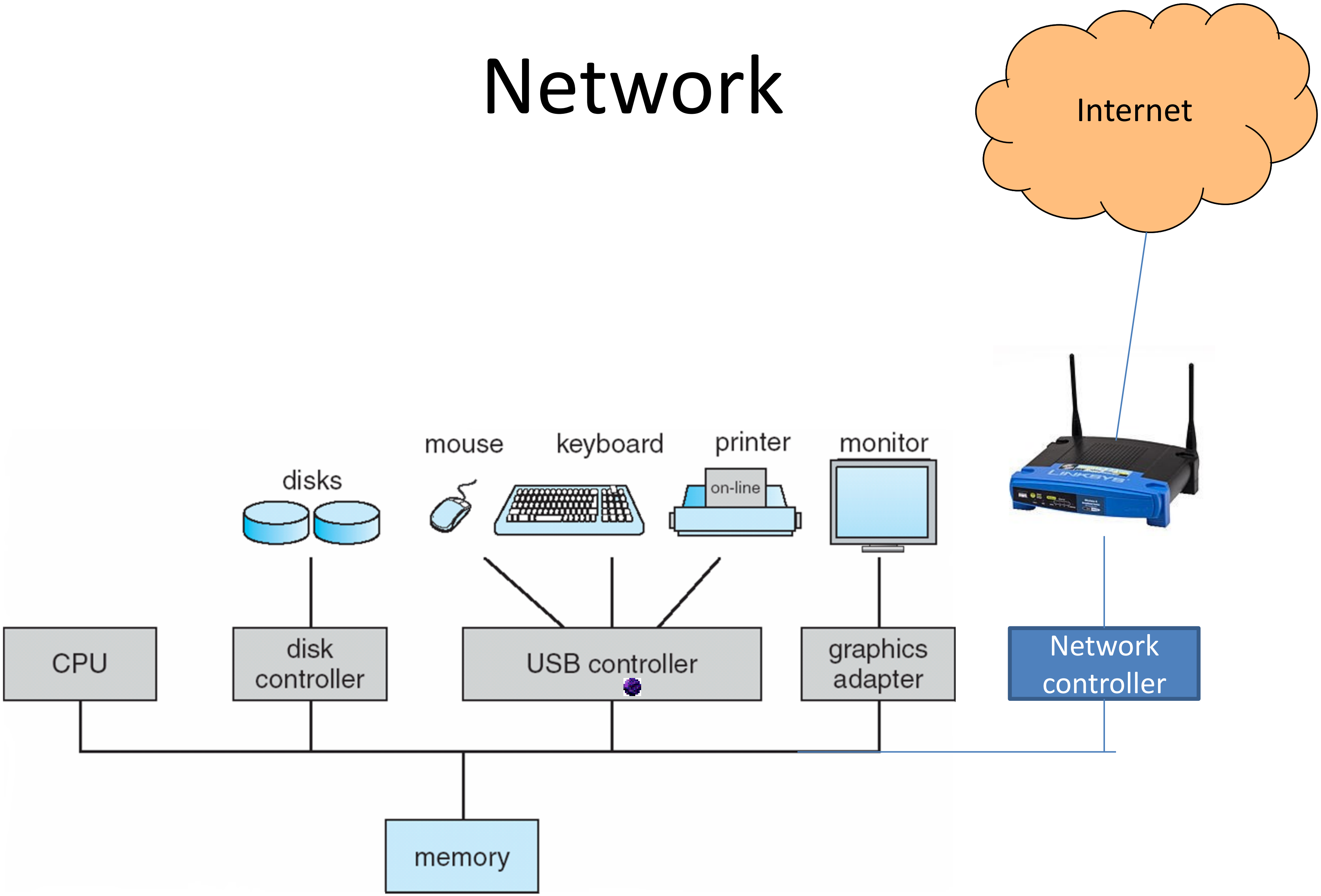
- A collection of connected computers



Why Distributed Computing?

- Resource sharing
  - Sharing and printing files at remote sites
  - Processing information in a distributed database
  - Using remote specialized hardware devices
- Performance
  - More computers → more performance
- Reliability
  - Detect and recover from site failure, function transfer, reintegrate failed site

Network

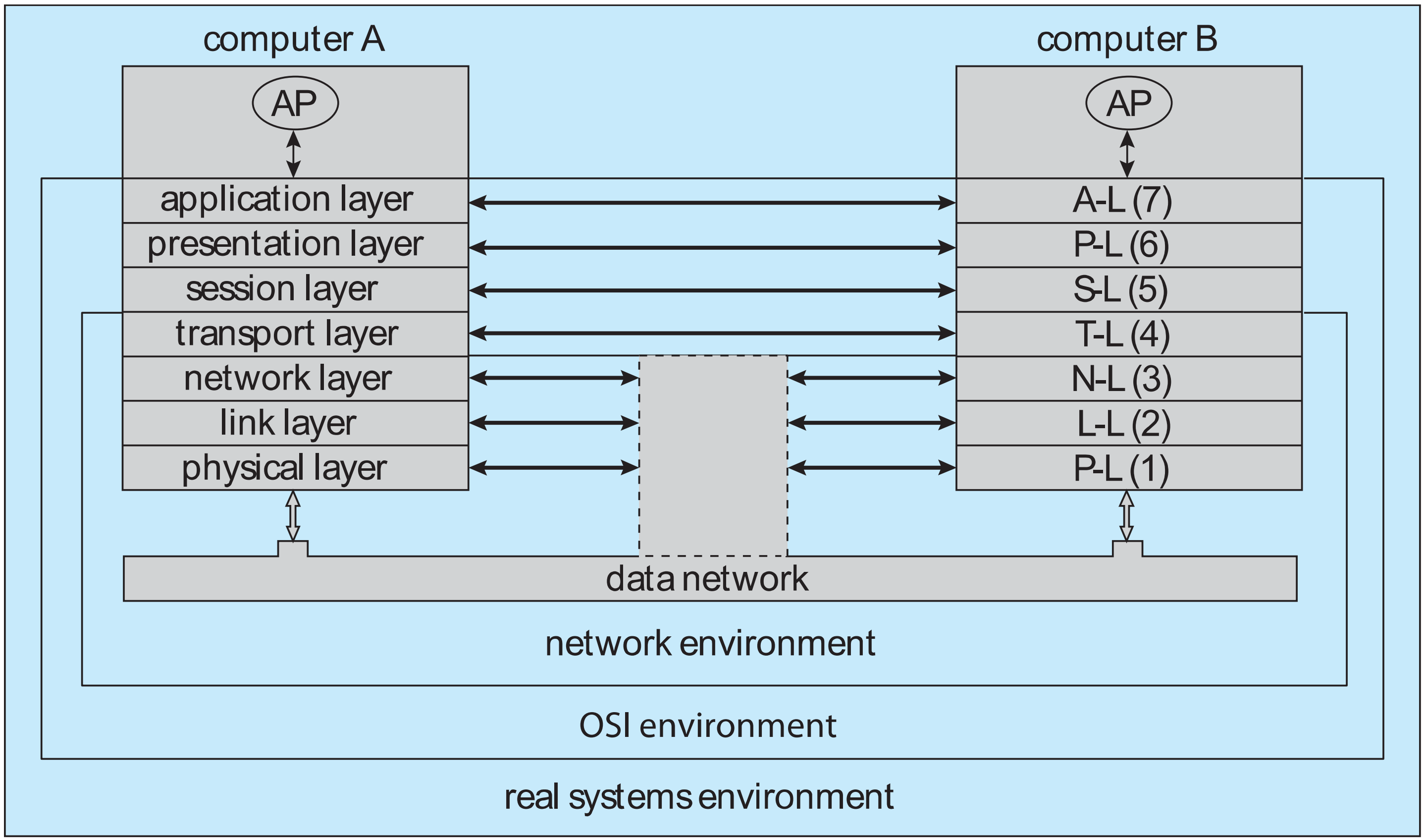

Internet
mouse
keyboard
printer
monitor
disks
on-line
CPU
disk controller
USB controller
graphics adapter
Network controller
memory

Terminologies

- Network
  - Physical medium of data transfer among multiple computers (e.g., Ethernet, CDMA,…)
- Packet
  - A unit of transfer in the network
- Protocol
  - A contract on how to transfer and receive data among the computers in the network

Communication Protocol

- Layered architecture



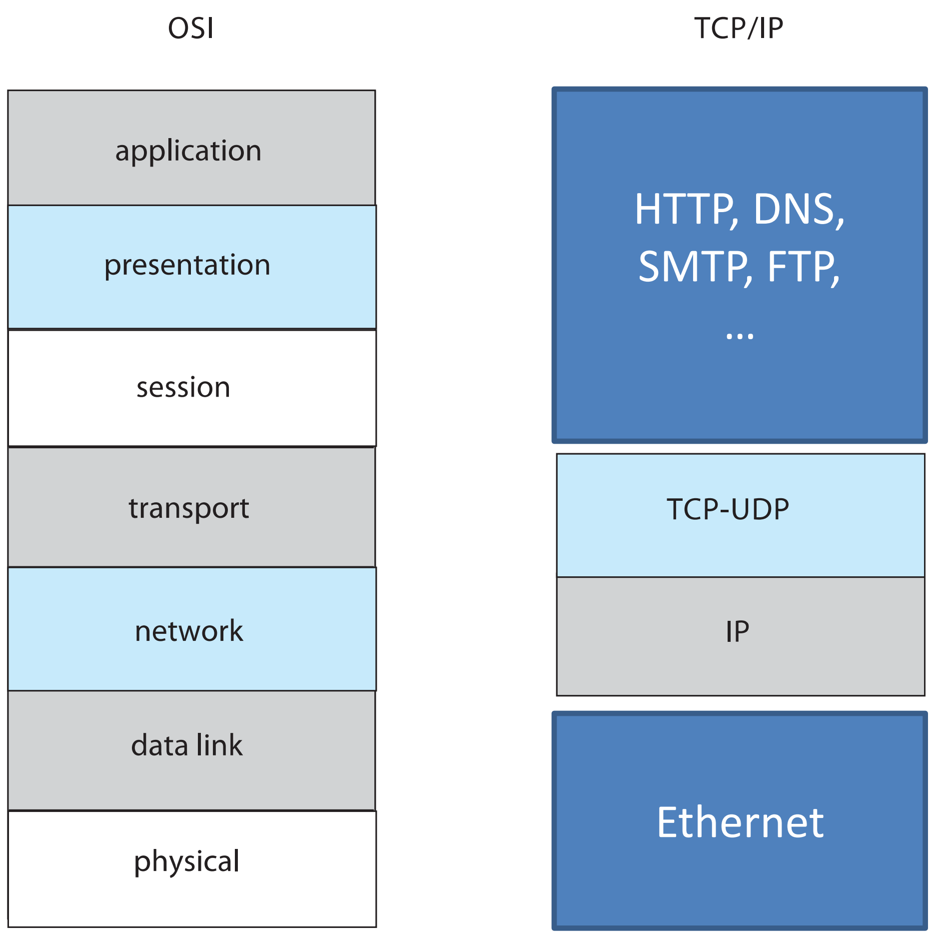
OSI 7 Layer communication model

OSI Layers

1. Physical - electrical details of the physical transmission of a bit stream
2. Data-link - reliable data delivery on the physical medium
3. Network - addressing, routing, and delivery of packets
4. Transport – reliable delivery over the network
5. Session – session management among applications
6. Presentation – data representation, encryption
7. Application – application specific

- Pros and Cons
  - Pros: separation of concerns
  - Cons: overhead, duplication

TCP/IP Protocol Layers



A Packet

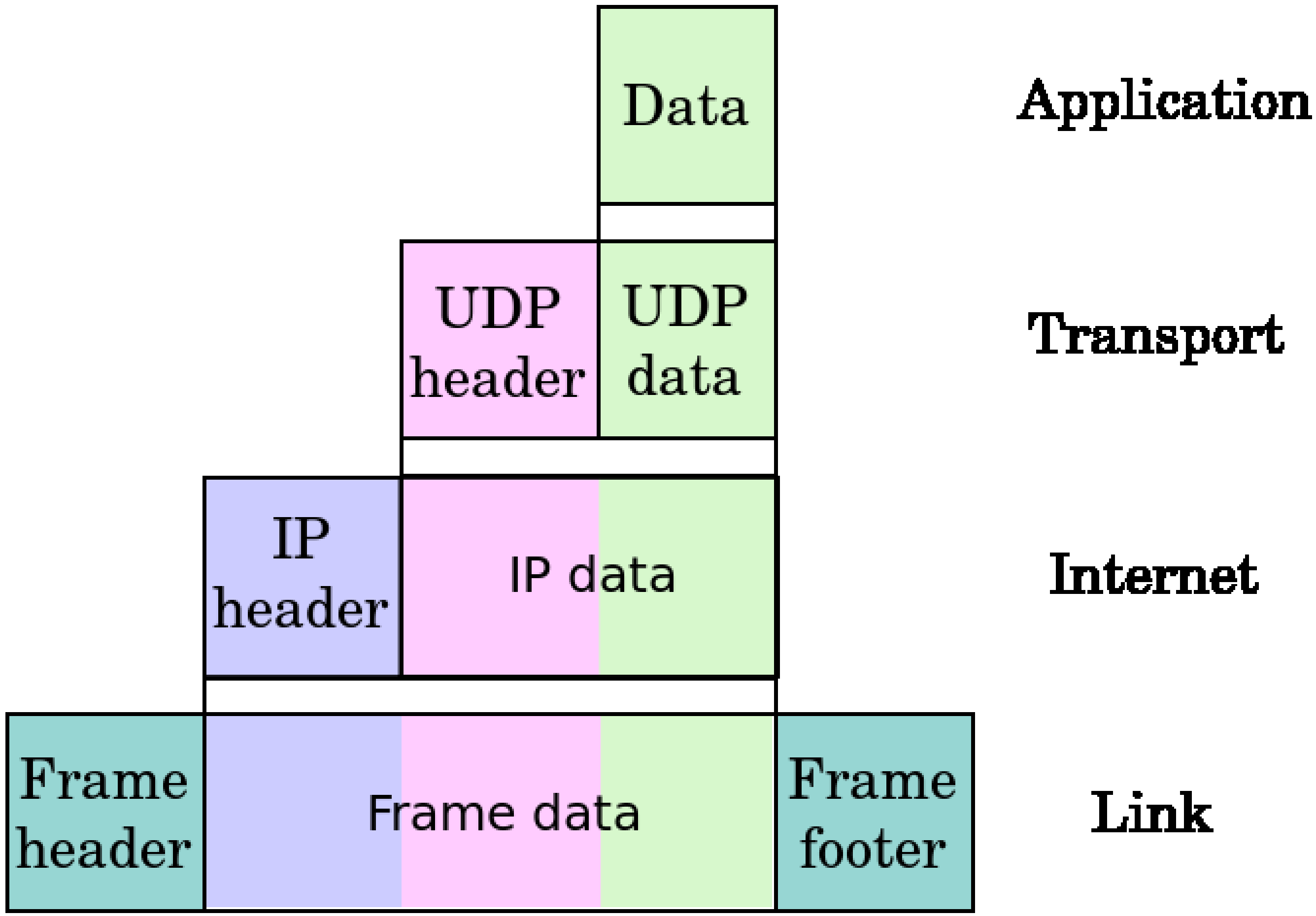


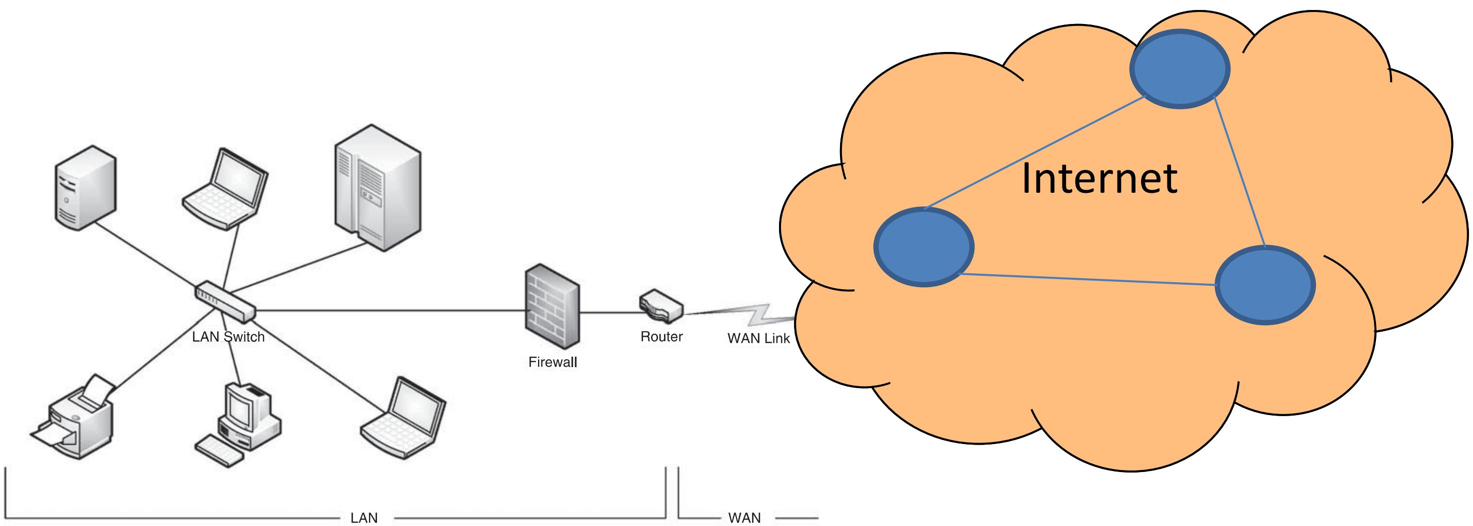
Image source: http://en.wikipedia.org/wiki/Internet_protocol_suite

An Ethernet Frame

bytes		
7	preamble—start of packet	each byte pattern 10101010
1	start of frame delimiter	pattern 10101011
2 or 6	destination address	Ethernet address or broadcast
2 or 6	source address	Ethernet address
2	length of data section	length in bytes
0–1500	data	message data
0–46	pad (optional)	message must be > 63 bytes long
4	frame checksum	for error detection

Internet Protocol (IP)

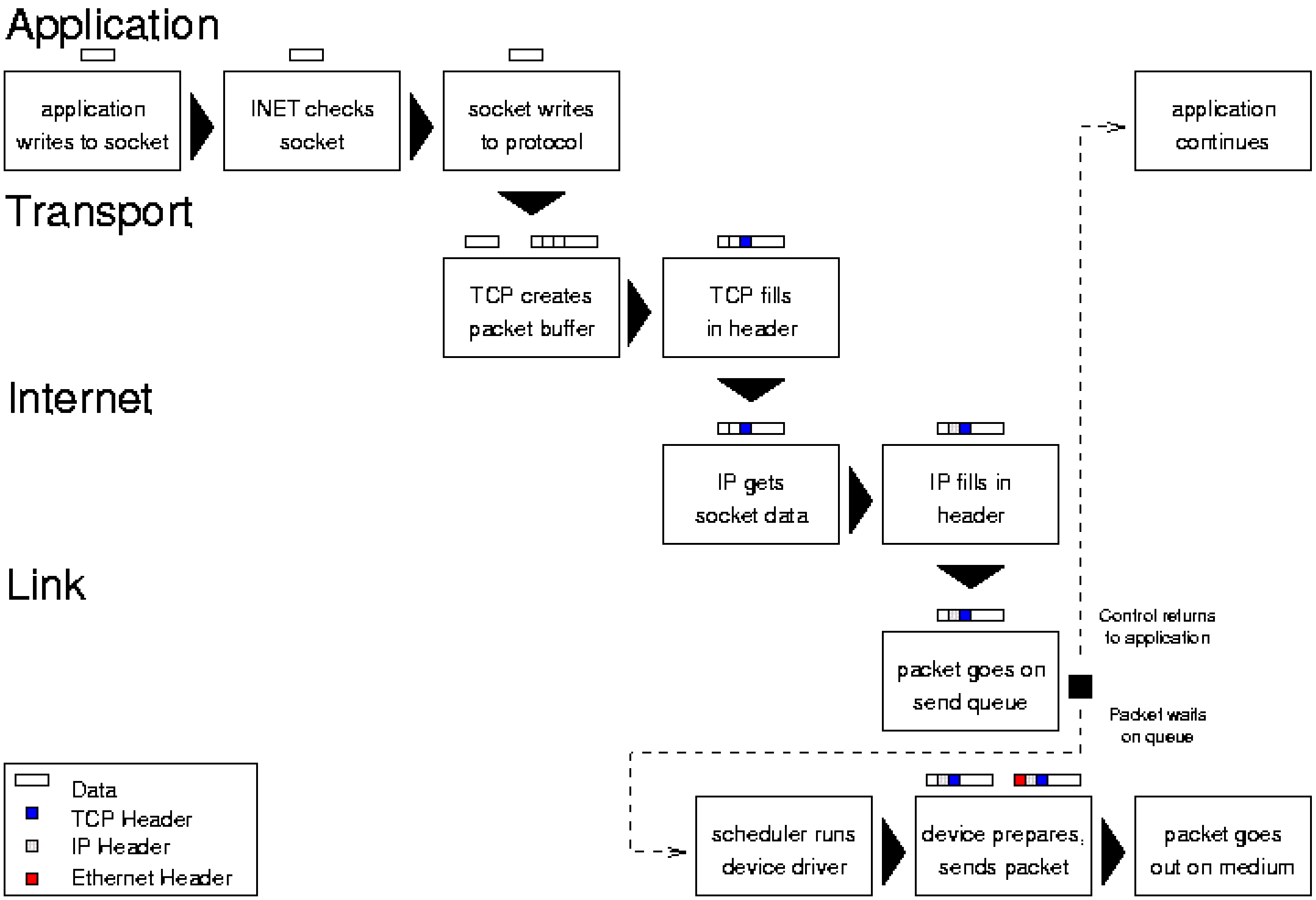
- Addressing
  - 32 bit (4 bytes) address: e.g., 129.237.123.1
- Routing
  - Forwarding packets through routers to reach their destination



Domain Name System (DNS)

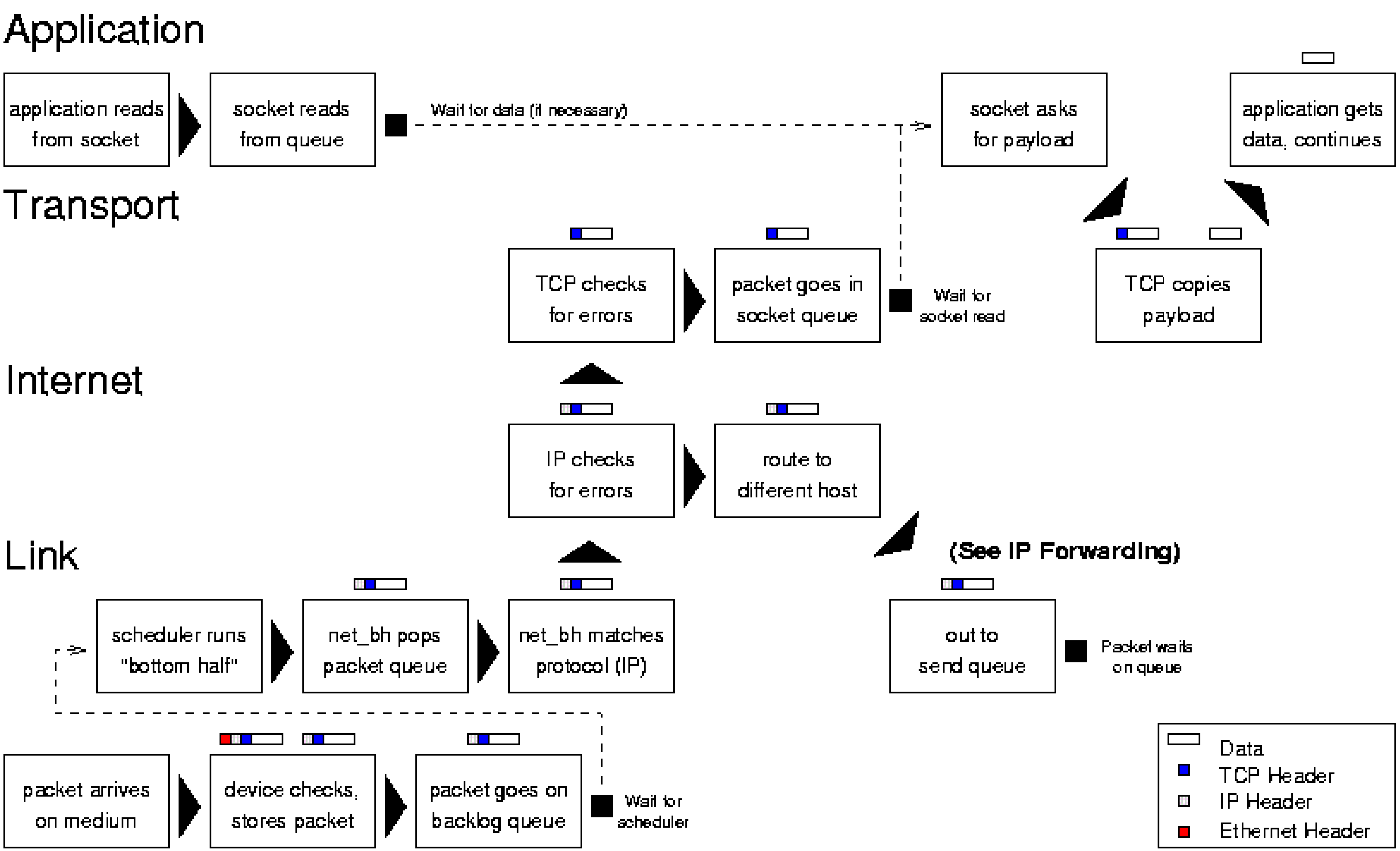
- Domain name
  - Human readable internet address: e.g., www.ku.edu
- How to map domain names to IP addresses?
  - www.ku.edu → 129.237.11.182
  - www.google.com → may vary depending on your location, server load, etc.
- Domain Name System
  - A distributed database of domain name, IP addr.

Sending a Packet



Source: G. Herrin, Linux IP Networking: A Guide to the Implementation and Modification of the Linux Protocol Stack, 2000

Receiving a Packet



Source: G. Herrin, Linux IP Networking: A Guide to the Implementation and Modification of the Linux Protocol Stack, 2000