disk scheduling

Disk Performance Parameters

- To read or write, the disk head must be positioned at the desired track and at the beginning of the desired sector
- Seek time
  - Time it takes to position the head at the desired track
- Rotational delay or rotational latency
  - Time it takes for the beginning of the sector to reach the head

Timing of a Disk I/O Transfer

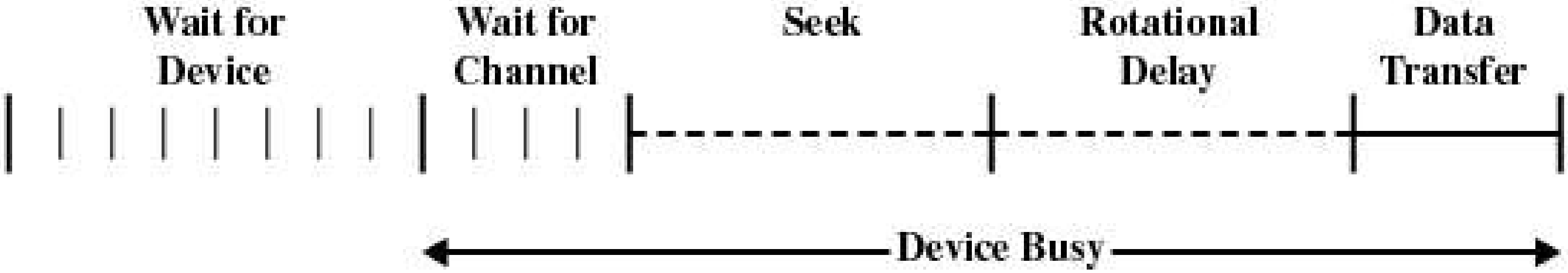


Figure 11.6 Timing of a Disk I/O Transfer

Disk Performance Parameters

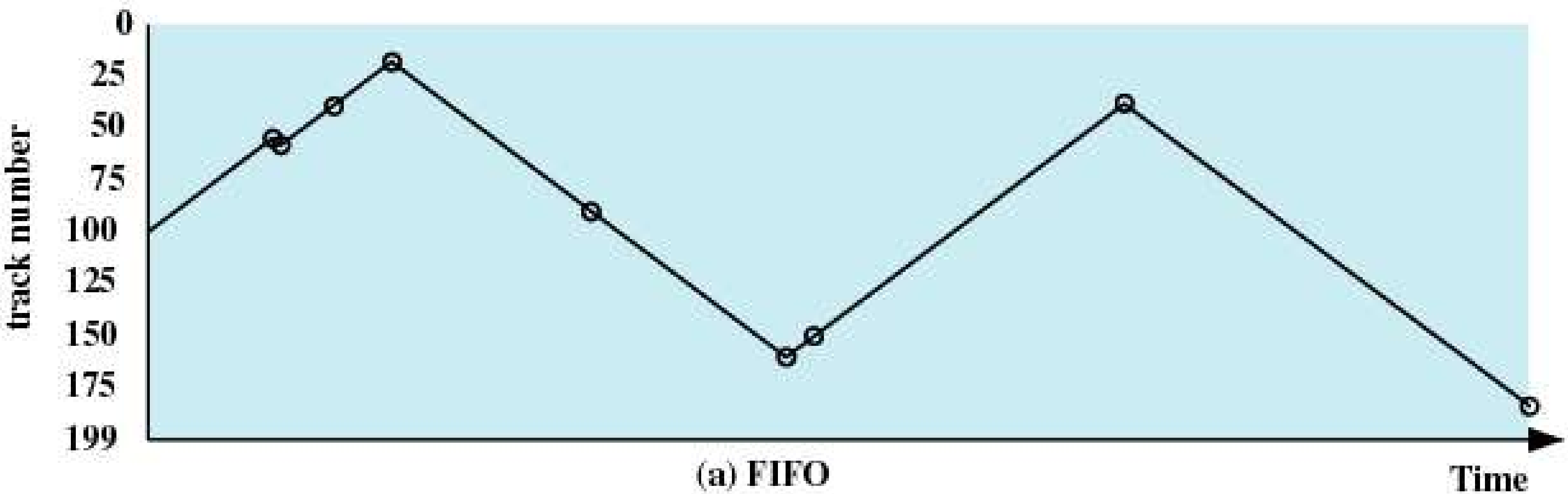
- Access time
  - Sum of seek time and rotational delay
  - The time it takes to get in position to read or write
- Data transfer occurs as the sector moves under the head

Disk Scheduling Policies

- Seek time is the reason for differences in performance
- For a single disk there will be a number of I/O requests
- If requests are selected randomly, we will poor performance

Disk Scheduling Policies

- First-in, first-out (FIFO)
  - Process request sequentially
  - Fair to all processes
  - if there are many processes it performs like random scheduling



(a) FIFO

Disk Scheduling Policies

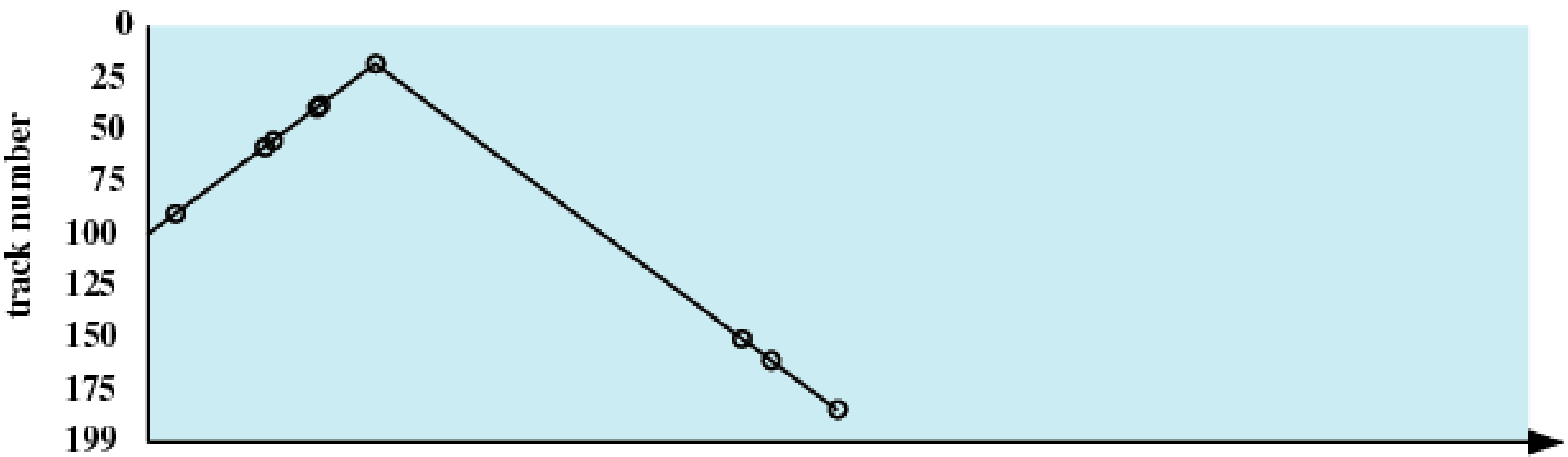
- Priority
  - Goal is not to optimize disk use but to meet other objectives
  - Short batch jobs may have higher priority
  - Provide good interactive response time

Disk Scheduling Policies

- Last-in, first-out
  - Good for transaction processing systems
    - The device is given to the most recent user so there should be little arm movement
  - Possibility of starvation since a job may never regain the head of the line

Disk Scheduling Policies

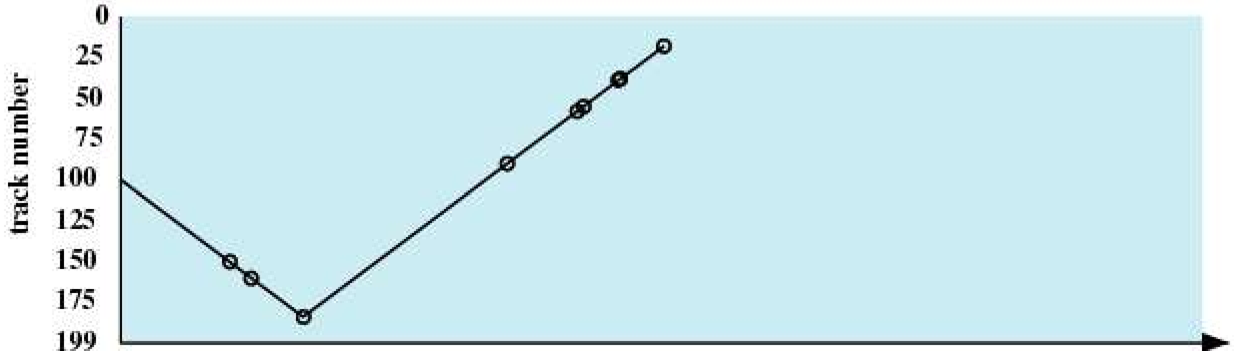
- Shortest Service Time First
  - Select the disk I/O request that requires the least movement of the disk arm from its current position
  - Always choose the minimum Seek time
  - Possibility of starvation



(b) SSTF

Disk Scheduling Policies

- SCAN (LOOK, ELEVATOR)
  - no starvation
  - Arm moves in one direction only, satisfying all outstanding requests until it reaches the last track in that direction
  - Direction is reversed



(c) SCAN

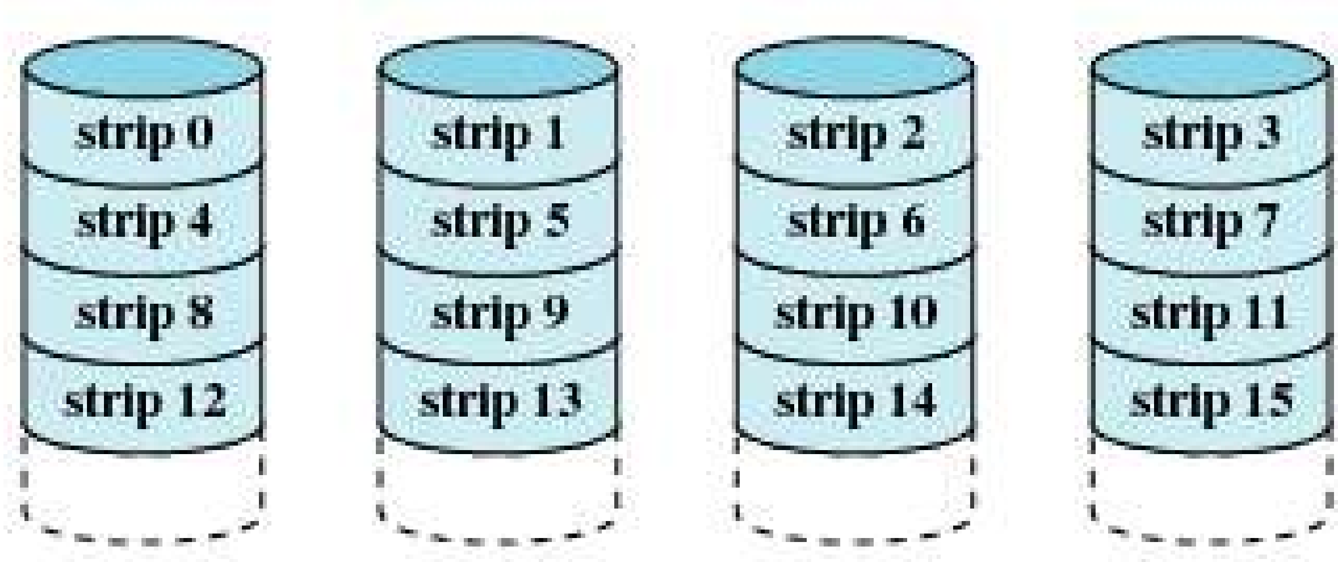
RAID

- Redundant Array of Independent Disks
- Set of physical disk drives viewed by the operating system as a single logical drive
- Data are distributed across the physical drives of an array
- Redundant disk capacity is used to store parity information

type of requests

- large I/O
  - big files (streaming, bulk)
  - usually of sectors stored contiguously
- High requests rates
  - very high frequency of requests for a very small amount of data
  - usually scattered through the disk

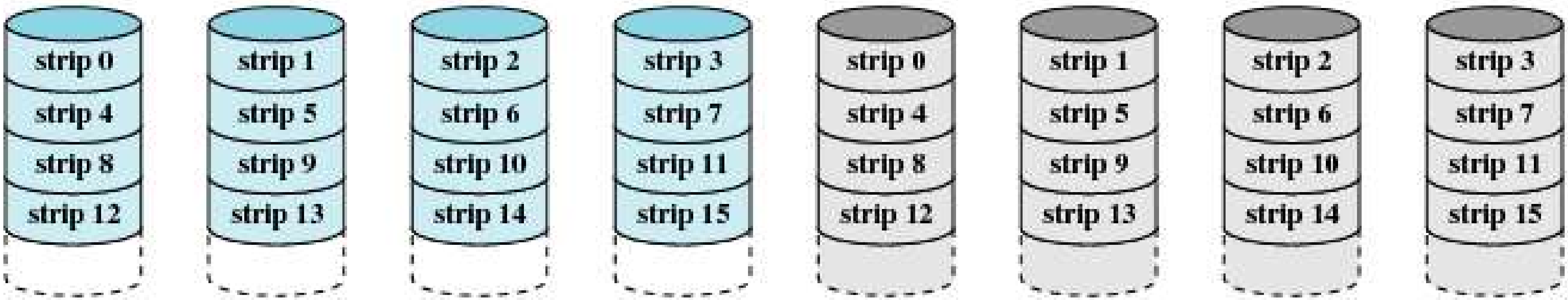
RAID 0 (non-redundant, striping)



(a) RAID 0 (non-redundant)

- availability: lower than single disk
- large I/O: very good
- high request rate: very good

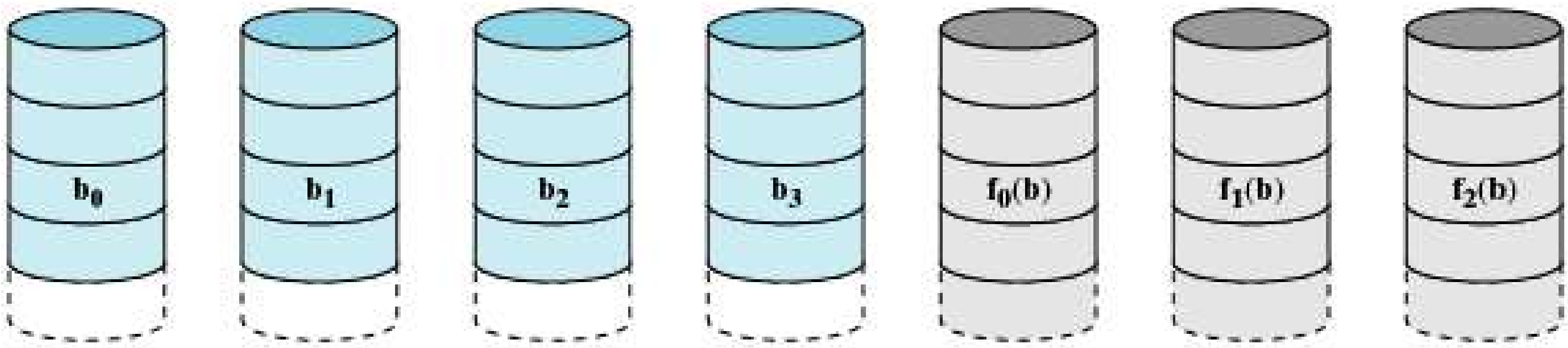
RAID 1 (mirrored)



(b) RAID 1 (mirrored)

- availability: high
- large I/O and high request rate: very good for read, like a single disk for write

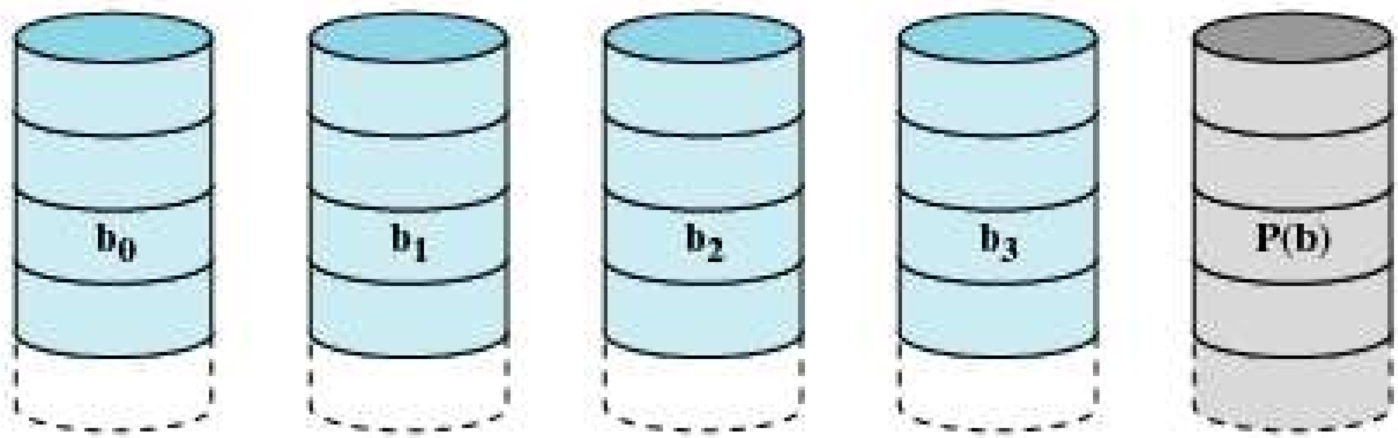
RAID 2 (redundancy through Hamming code)



(c) RAID 2 (redundancy through Hamming code)

- disks should be syncronized
- availability: high also for high bit error rate
- large I/O: best!!!
- high request rate: better than a single disk
- expensive!

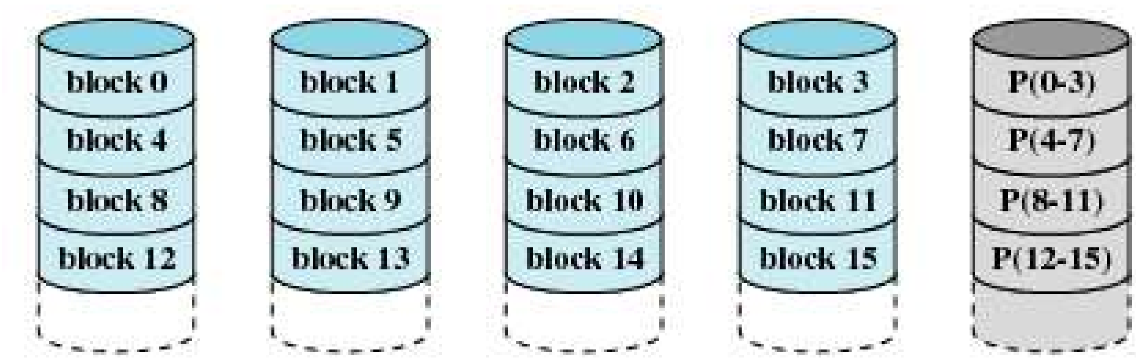
RAID 3 (bit-interleaved parity)



(d) RAID 3 (bit-interleaved parity)

- disks should be syncronized
- availability: high
- large I/O: best!!!
- high request rate: better than a single disk

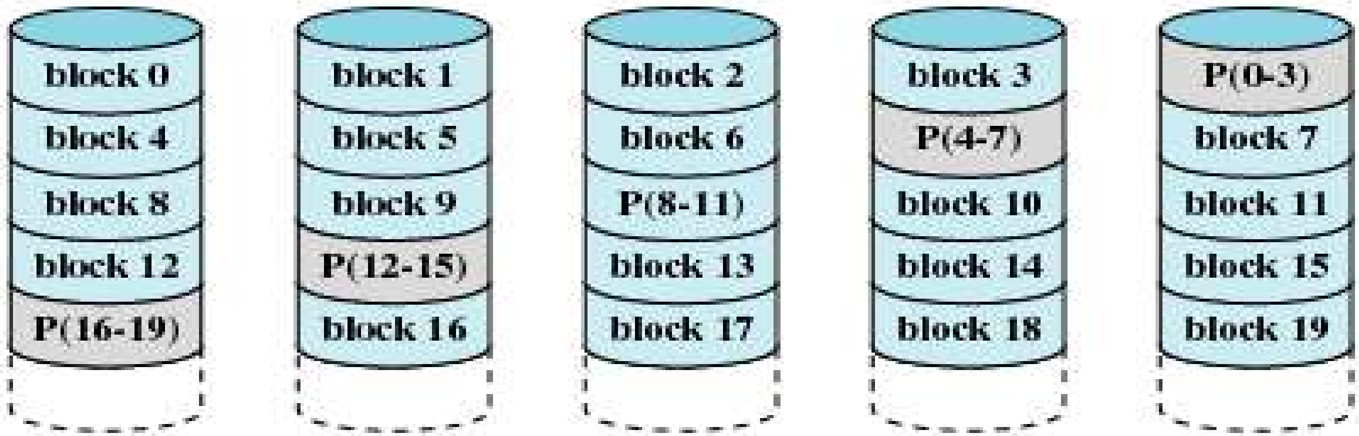
RAID 4 (block-level parity)



(e) RAID 4 (block-level parity)

- disks are independent
- availability: high
- **P is a bottlenek for write**
- large I/O: good, very bad for write
- high request rate: very good for read, very bad for write

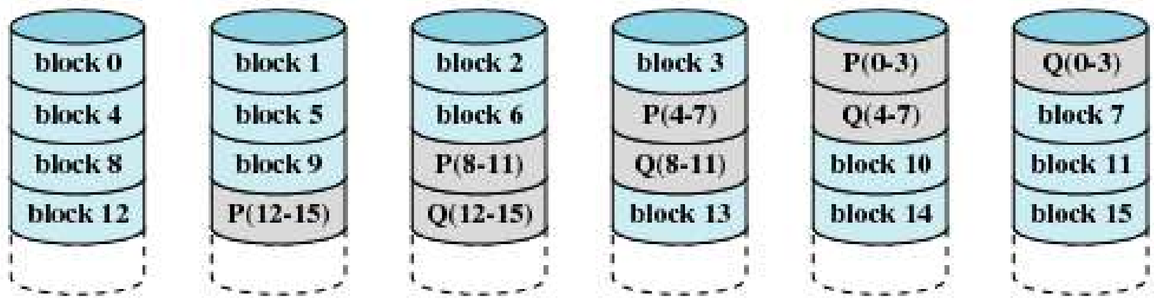
RAID 5 (block-level distributed parity)



(f) RAID 5 (block-level distributed parity)

- disks are independent
- availability: high
- large I/O: very good, bad for write (no bottlenek)
- high request rate: very good for read, bad for write

RAID 6 (dual redundancy)



(g) RAID 6 (dual redundancy)

- availability: highest
  - two disks may fail without data loss
- large I/O: very good for read, bad for write
- high request rate: very good for read, very bad for write