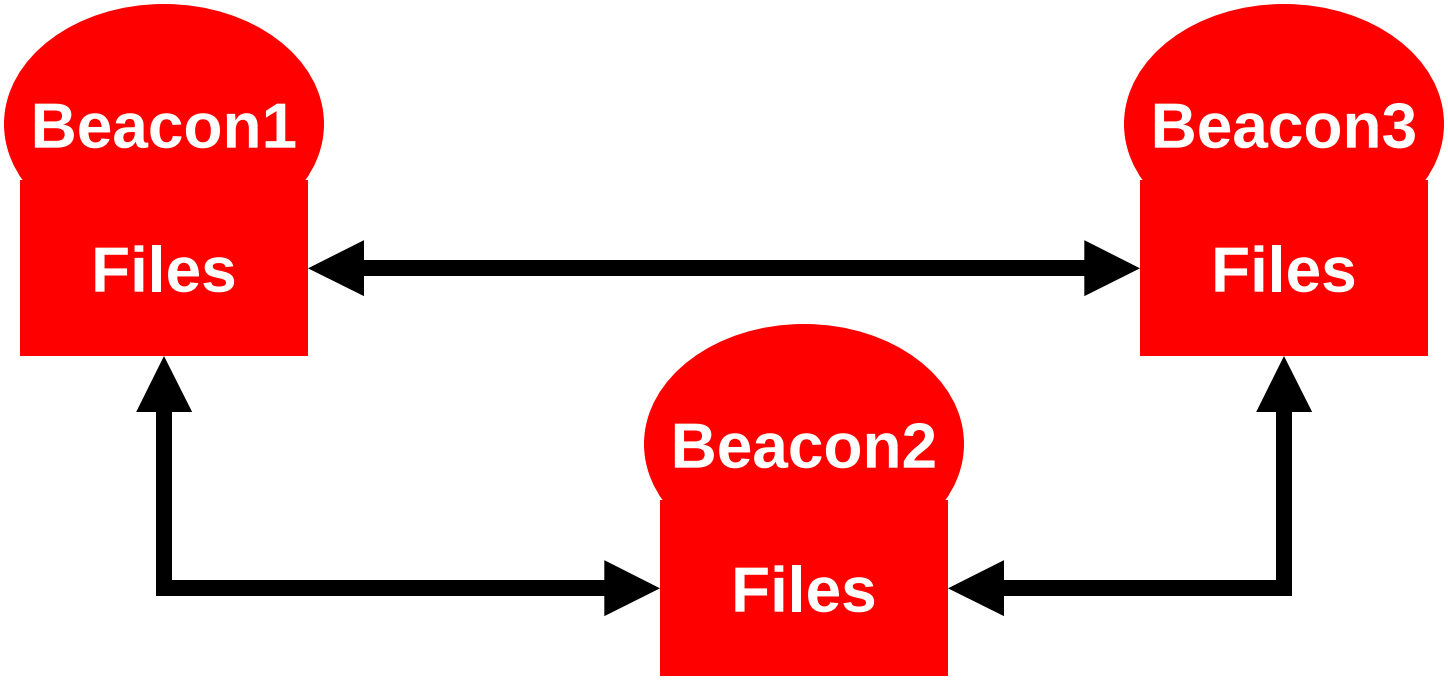
Announcements

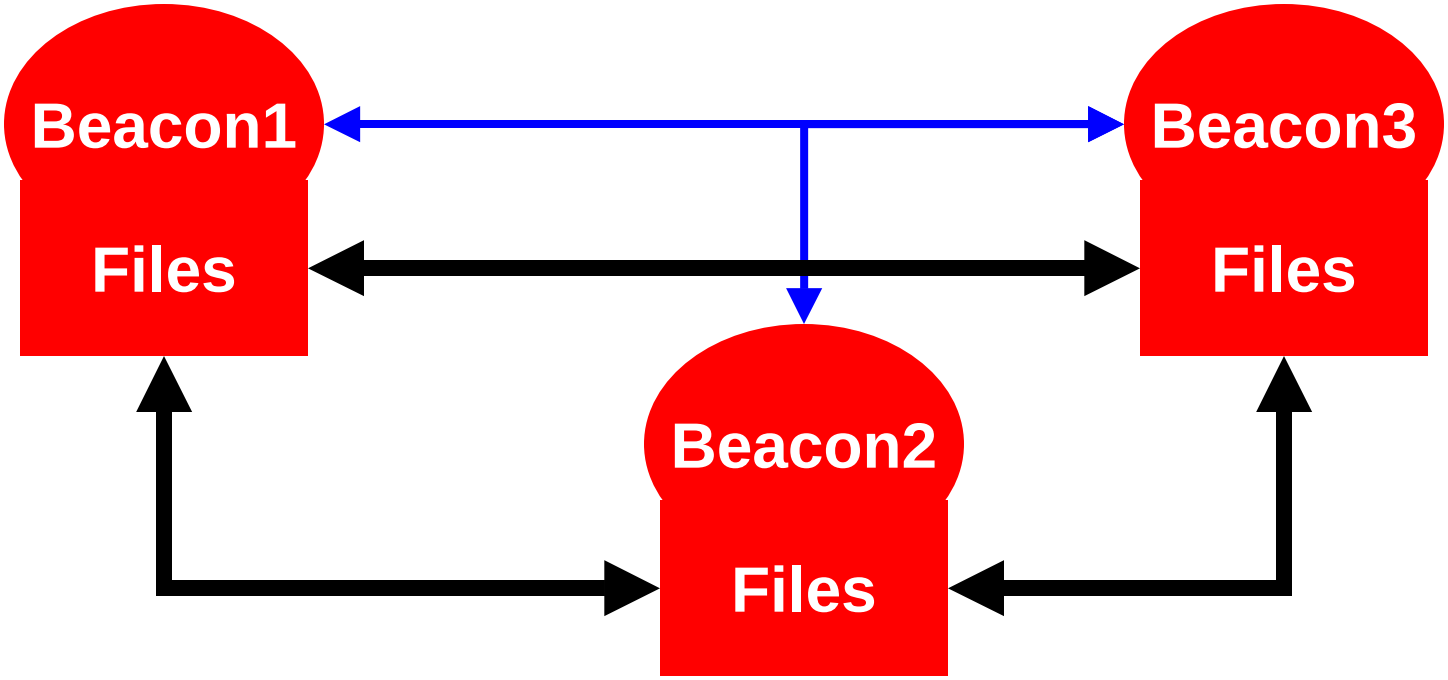
- HW 03

Home work 2



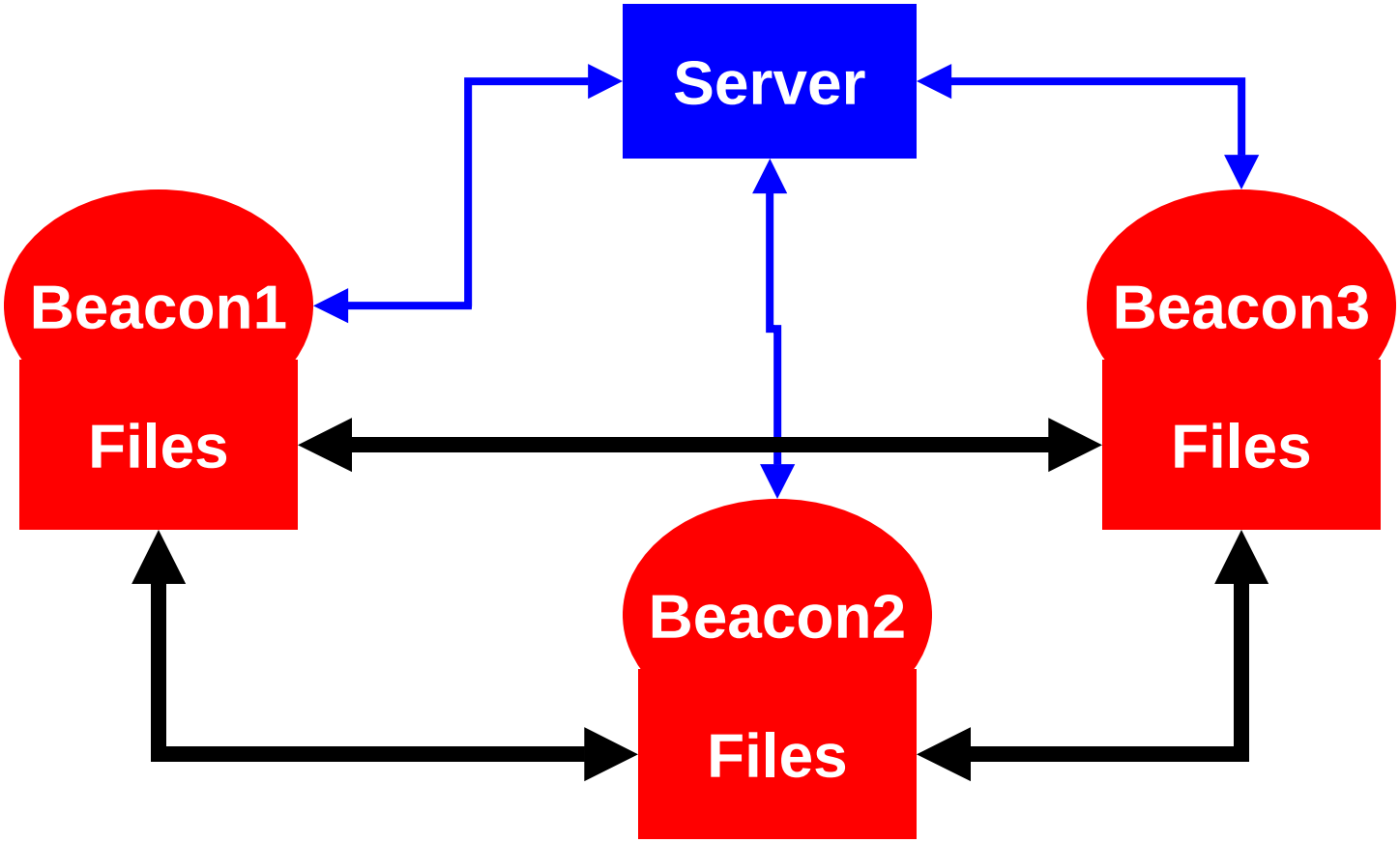
- Beacons provide file service for other beacons

Home work 2



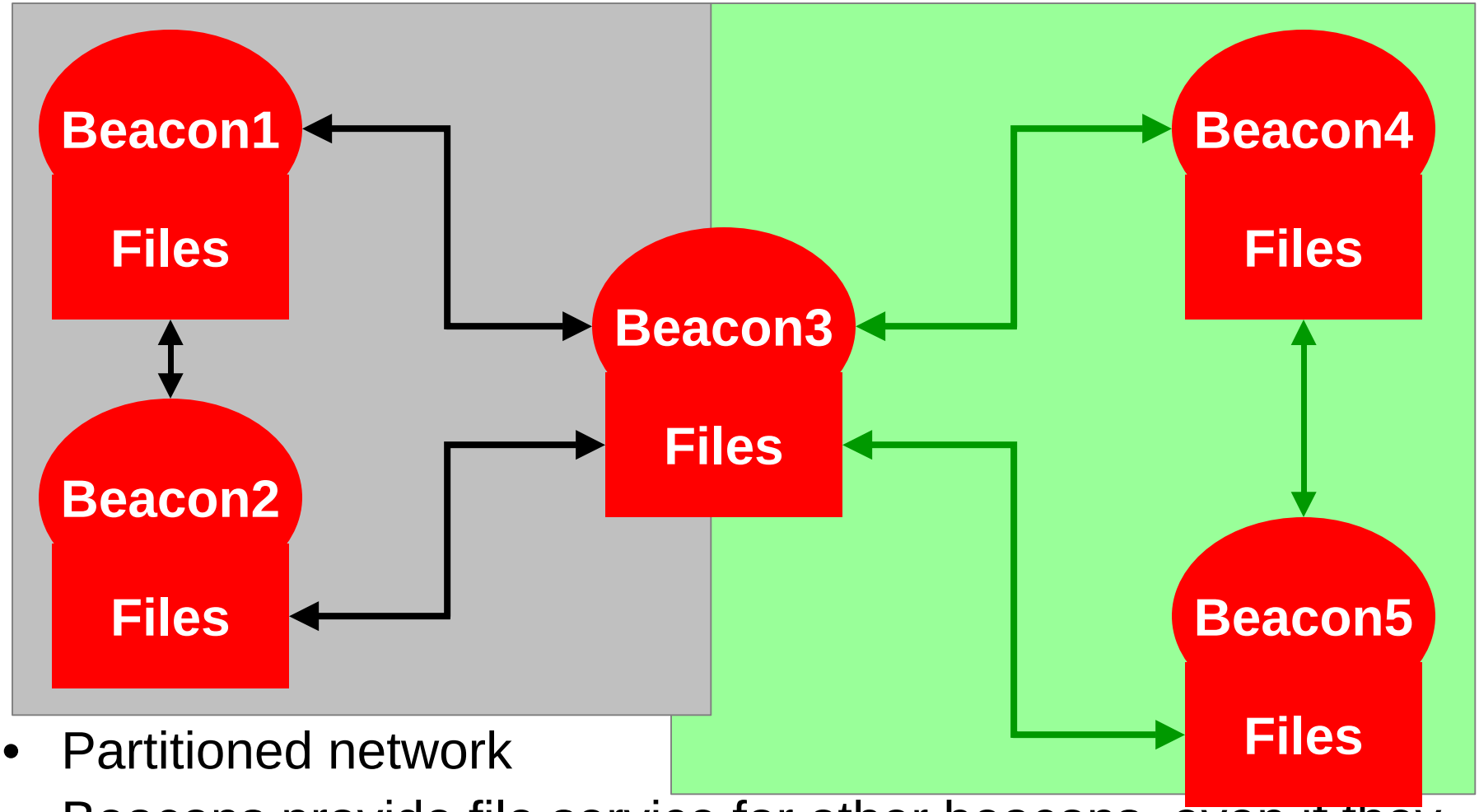
- Beacons provide file service for other beacons
- Used peer-to-peer to find other beacons

Home work 2



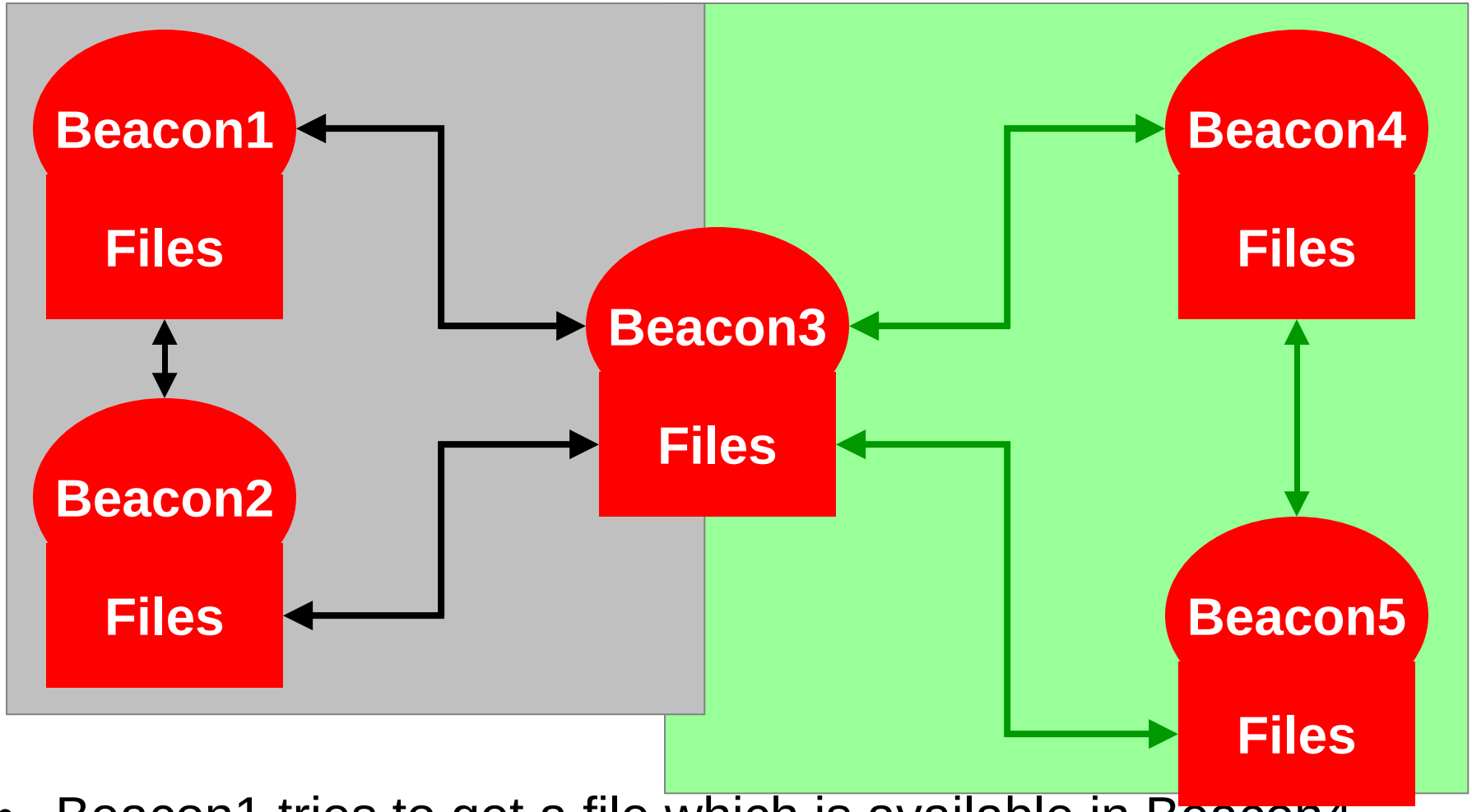
- Beacons provide file service for other beacons
- Used central server to find other beacons

Home work 3



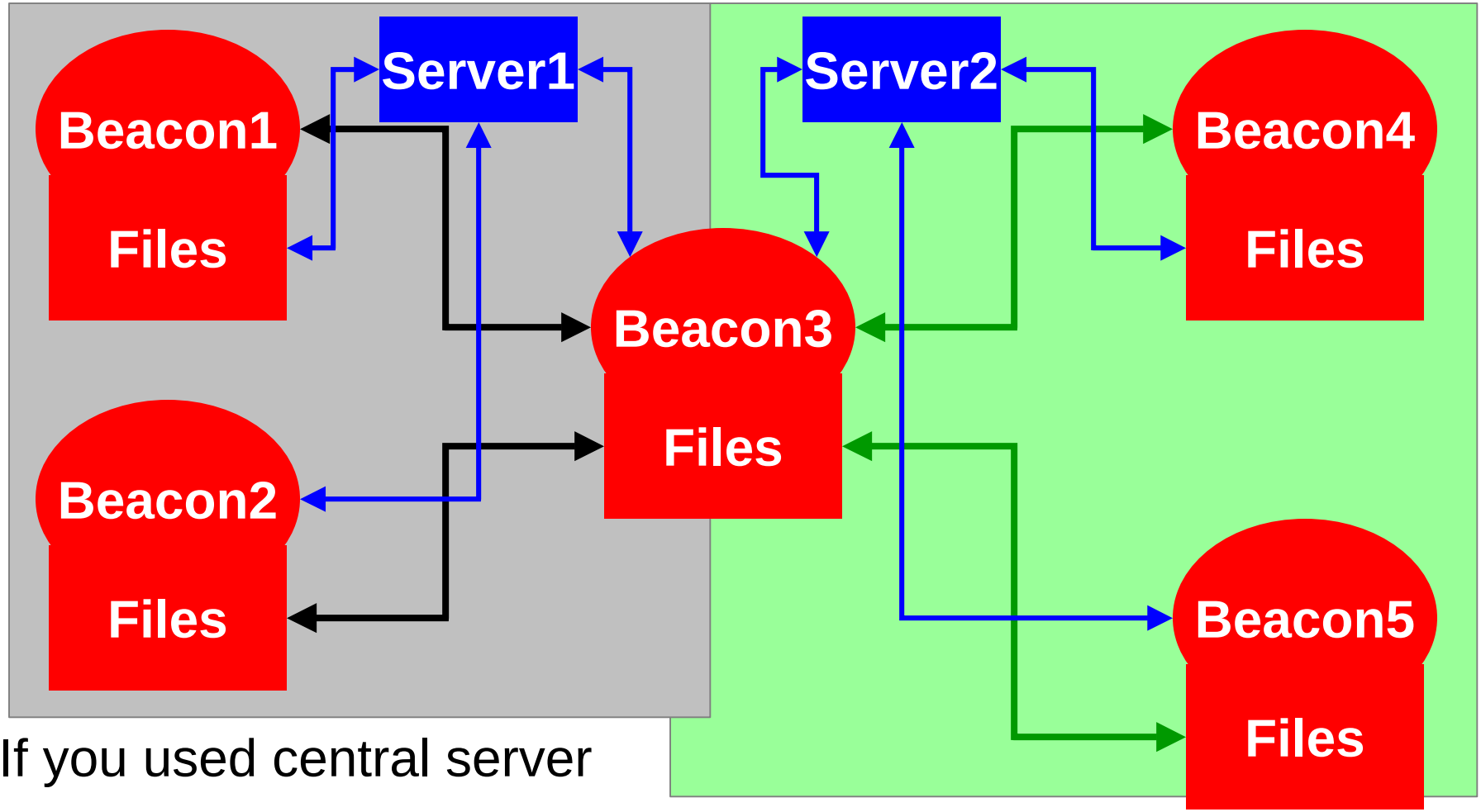
- Partitioned network
- Beacons provide file service for other beacons, even if they are not directly accessible

Home work 3



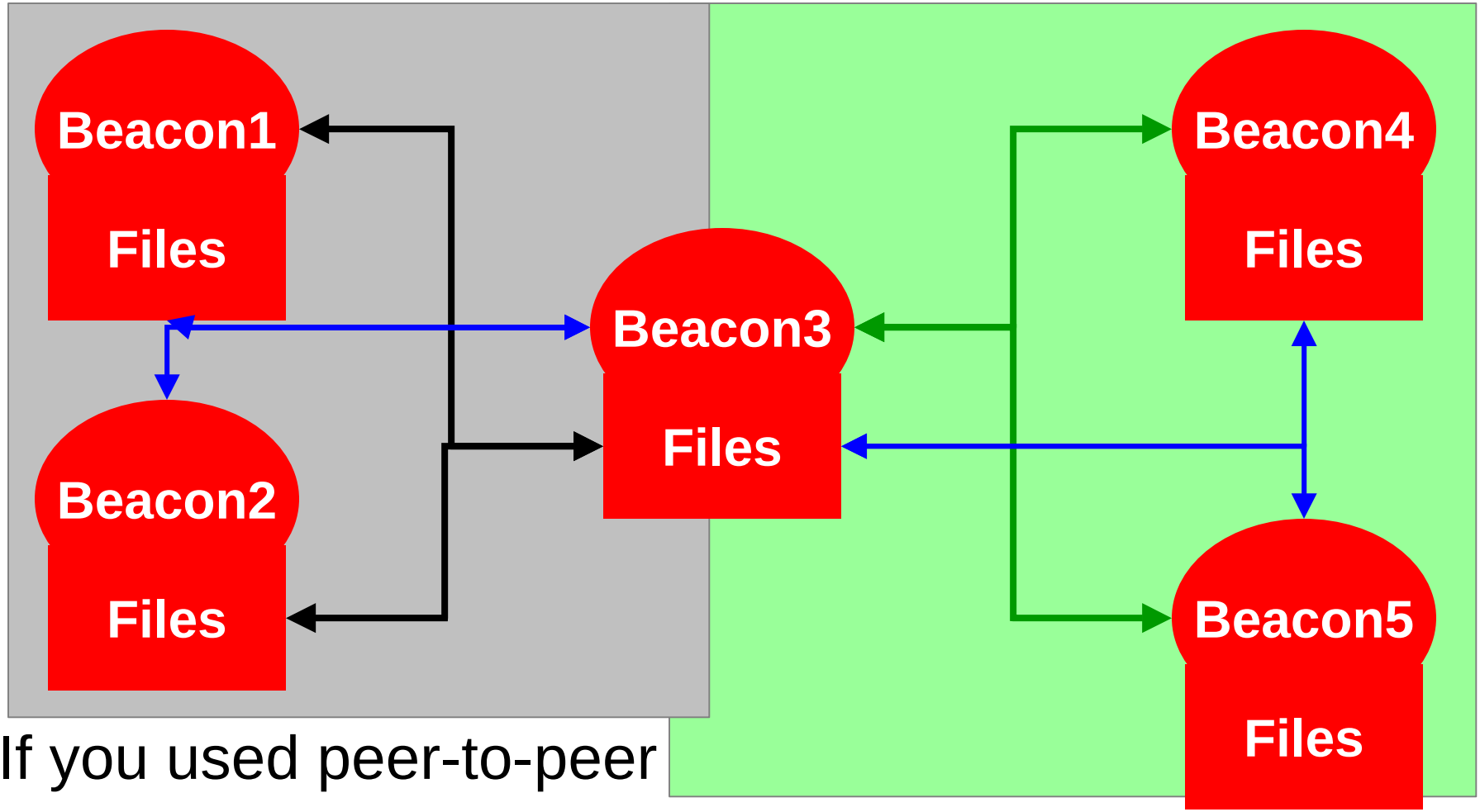
- Beacon1 tries to get a file which is available in Beacon4
- Beacon1 does not know that it is available in Beacon4

Home work 3 – Step 1



If you used central server to find other beacons, extend with multiple beacon servers

Home work 3 – Step 1

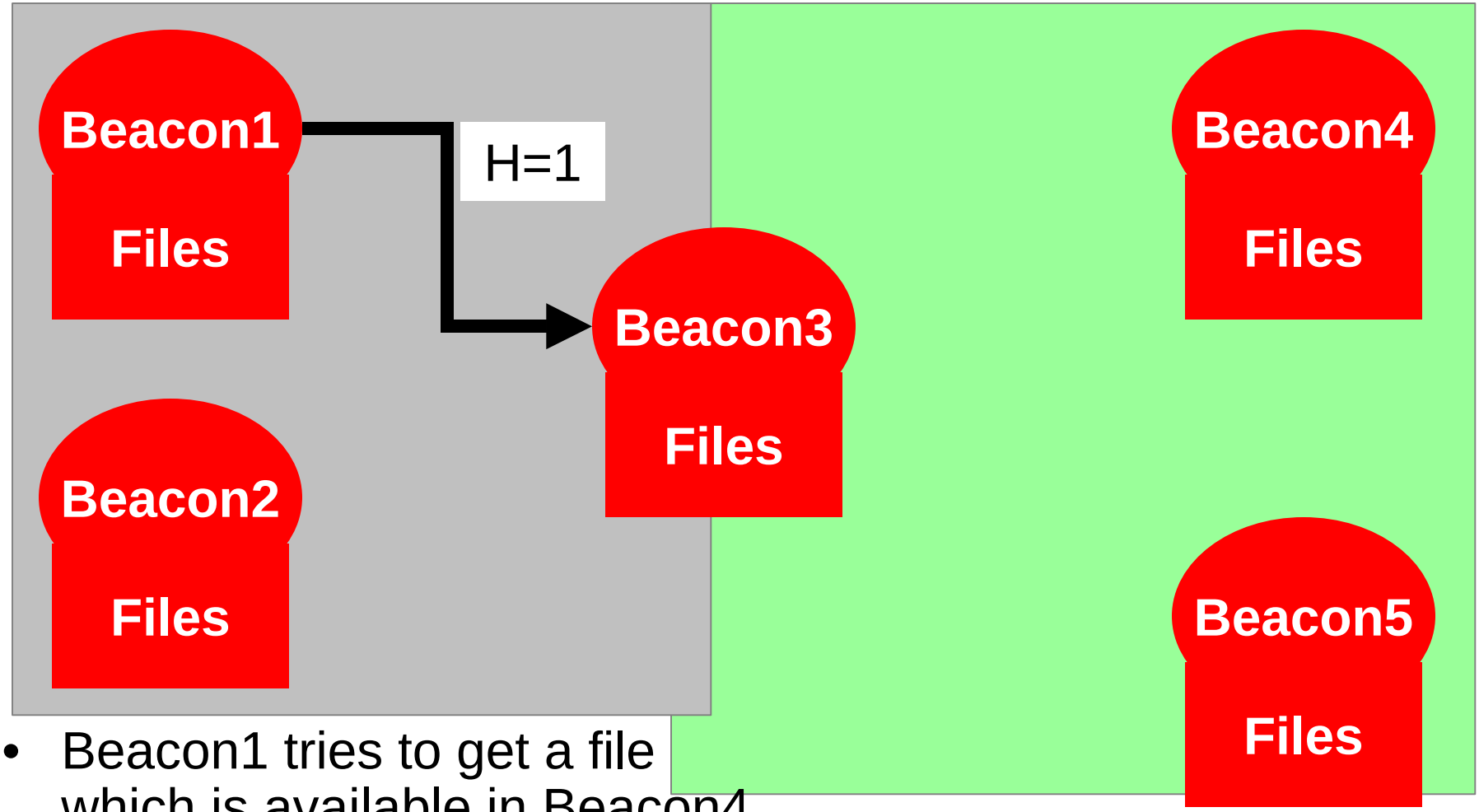


If you used peer-to-peer Networking, you should simulate partitioning

New Service

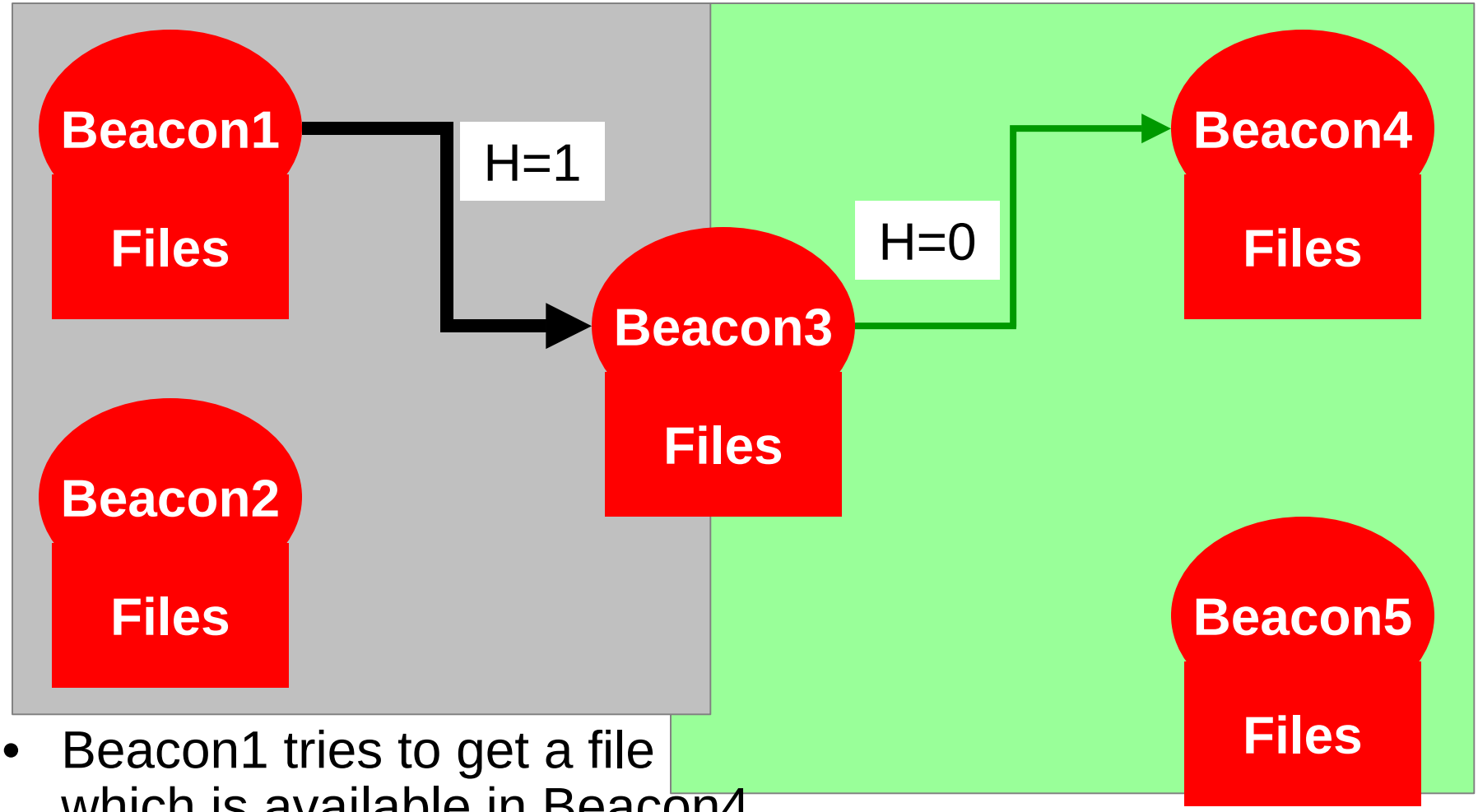
- Searchget(token, fileName, hopcount)
  - Search for fileName. You specify a upper-bound for the search using hopcount
  - You can use a traditional RPC based approach (Homework 2) or multiway RPC (described in ActiveNames paper)

Home work 3



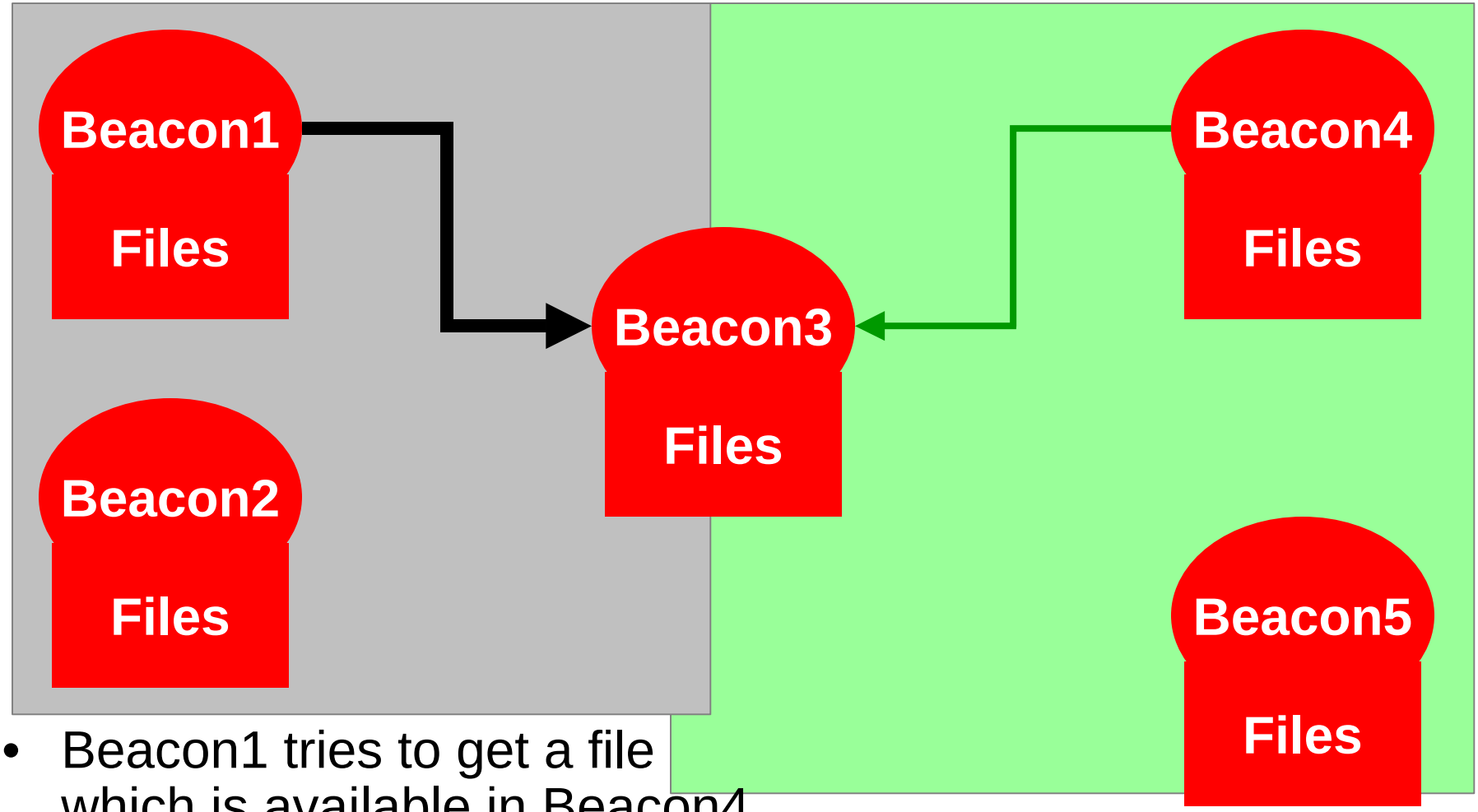
- Beacon1 tries to get a file which is available in Beacon4. Hopcount=2

Home work 3



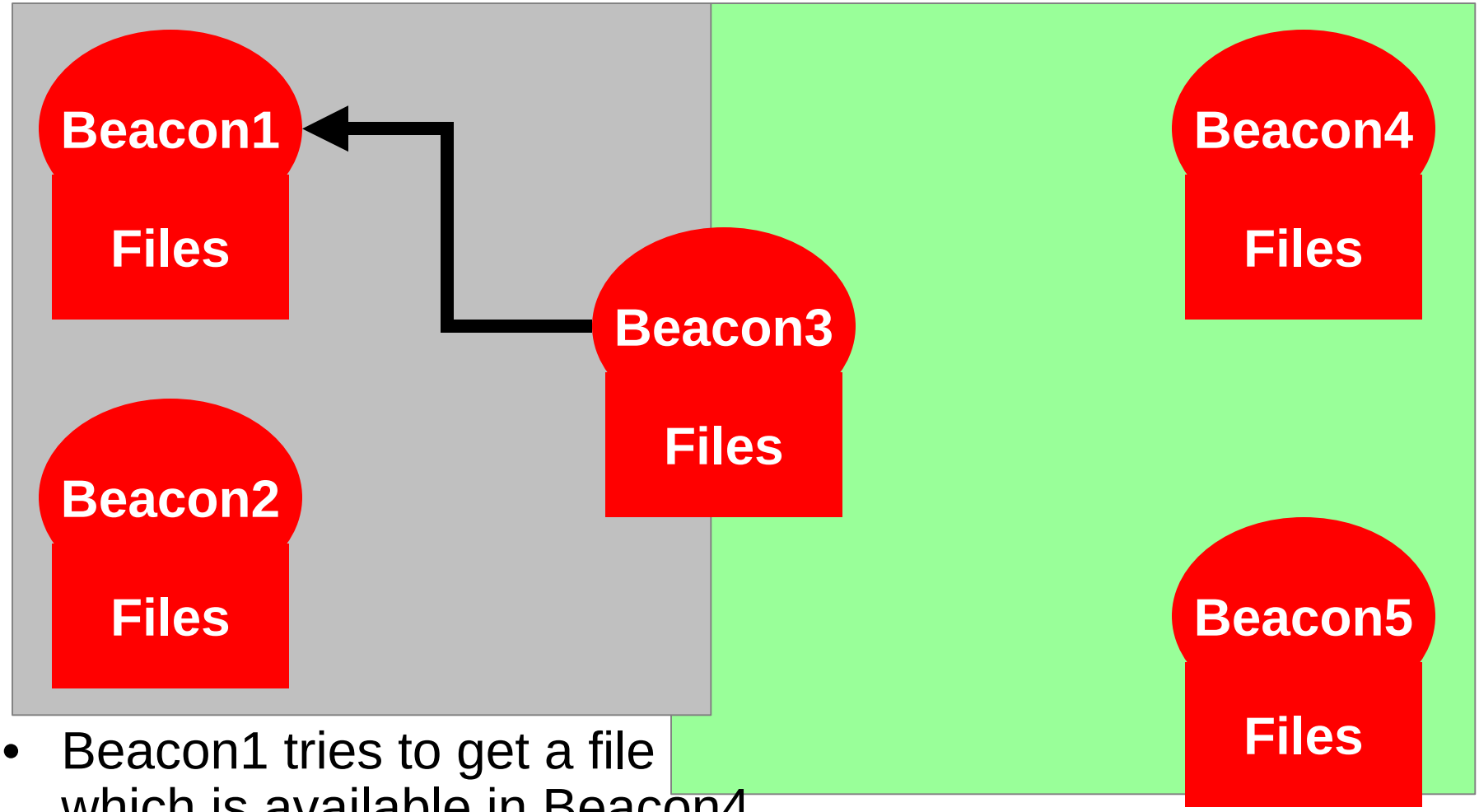
- Beacon1 tries to get a file which is available in Beacon4. Hopcount=2

Home work 3



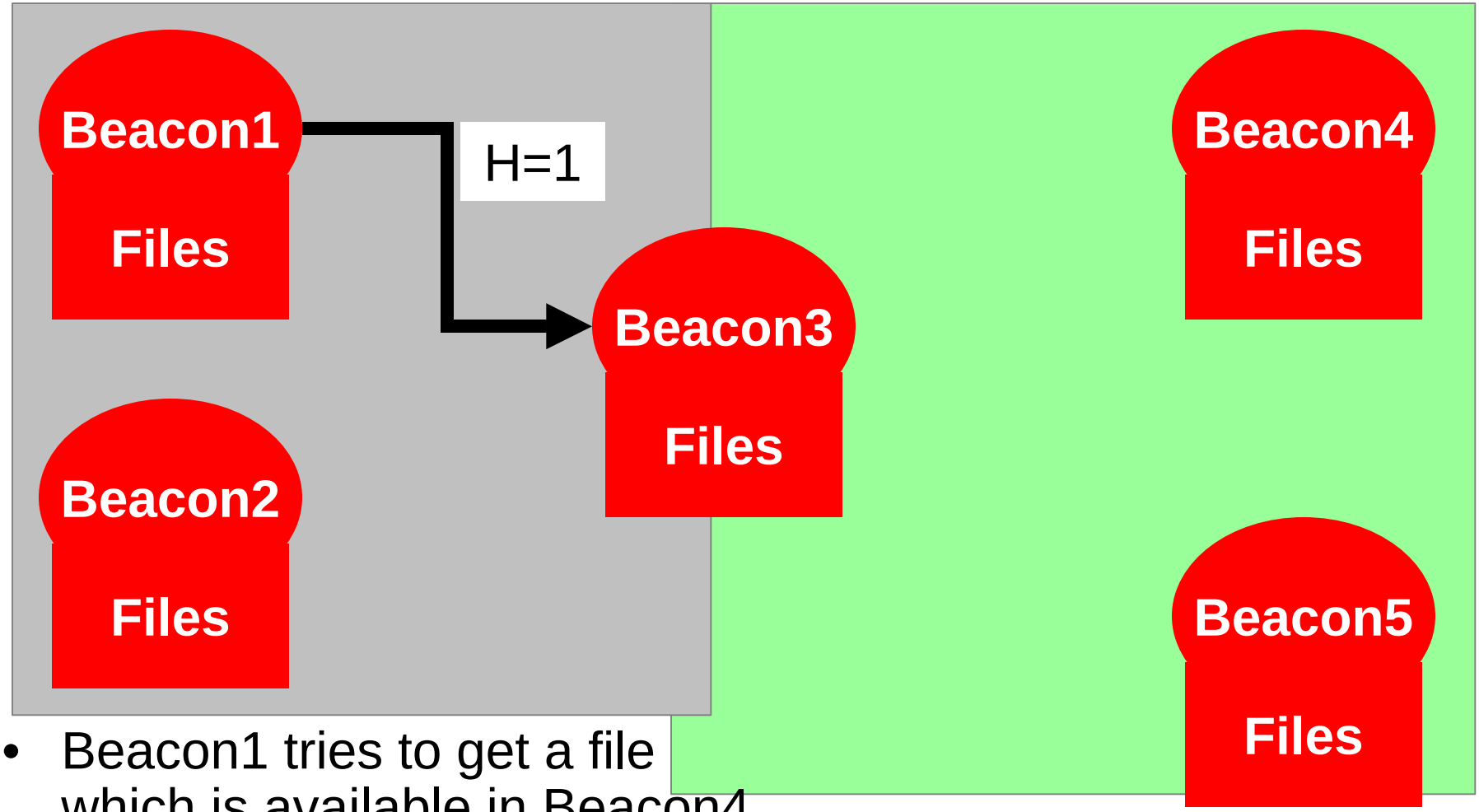
- Beacon1 tries to get a file which is available in Beacon4. Hopcount=2

Home work 3



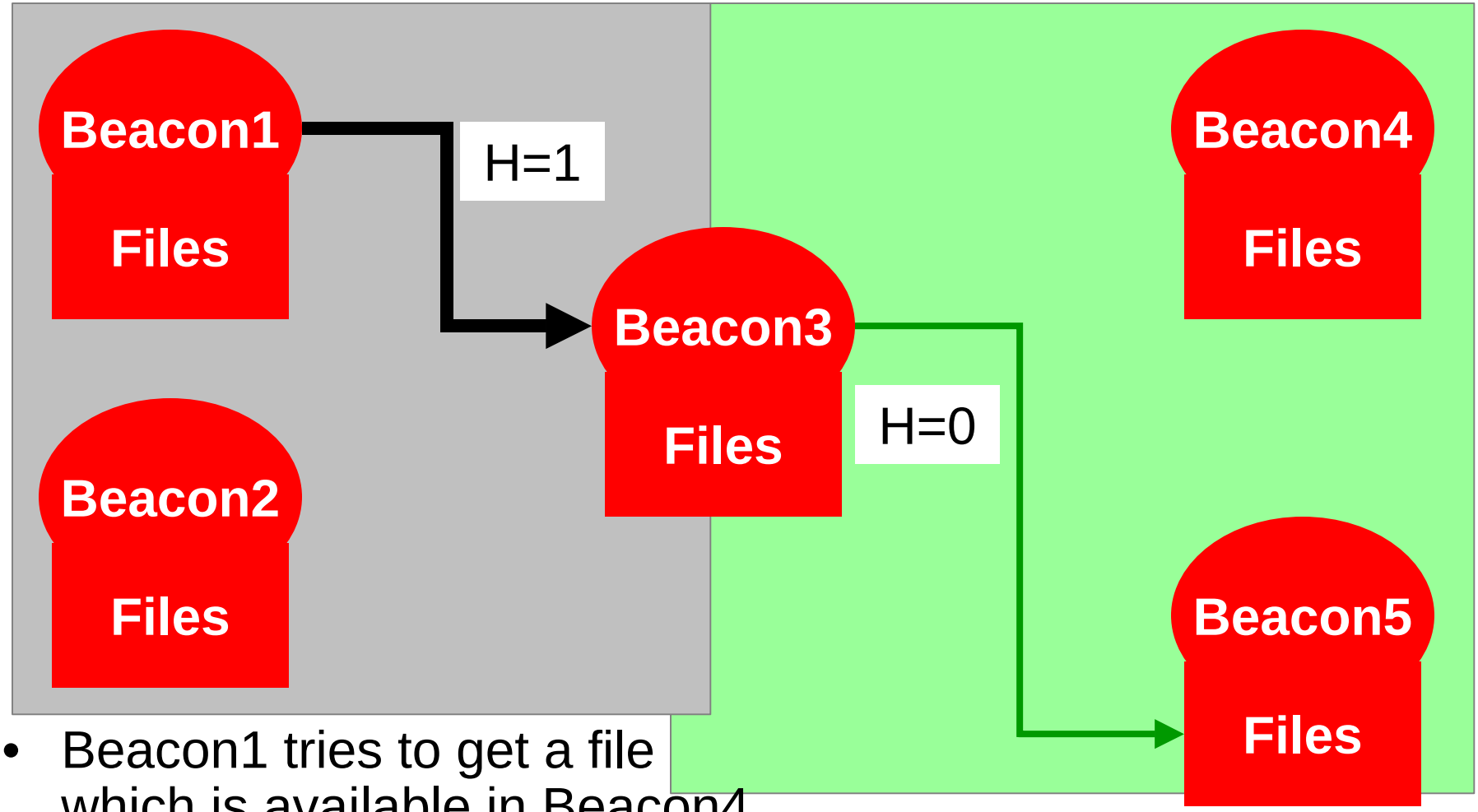
- Beacon1 tries to get a file which is available in Beacon4. Hopcount=2

Home work 3 – Alternative path



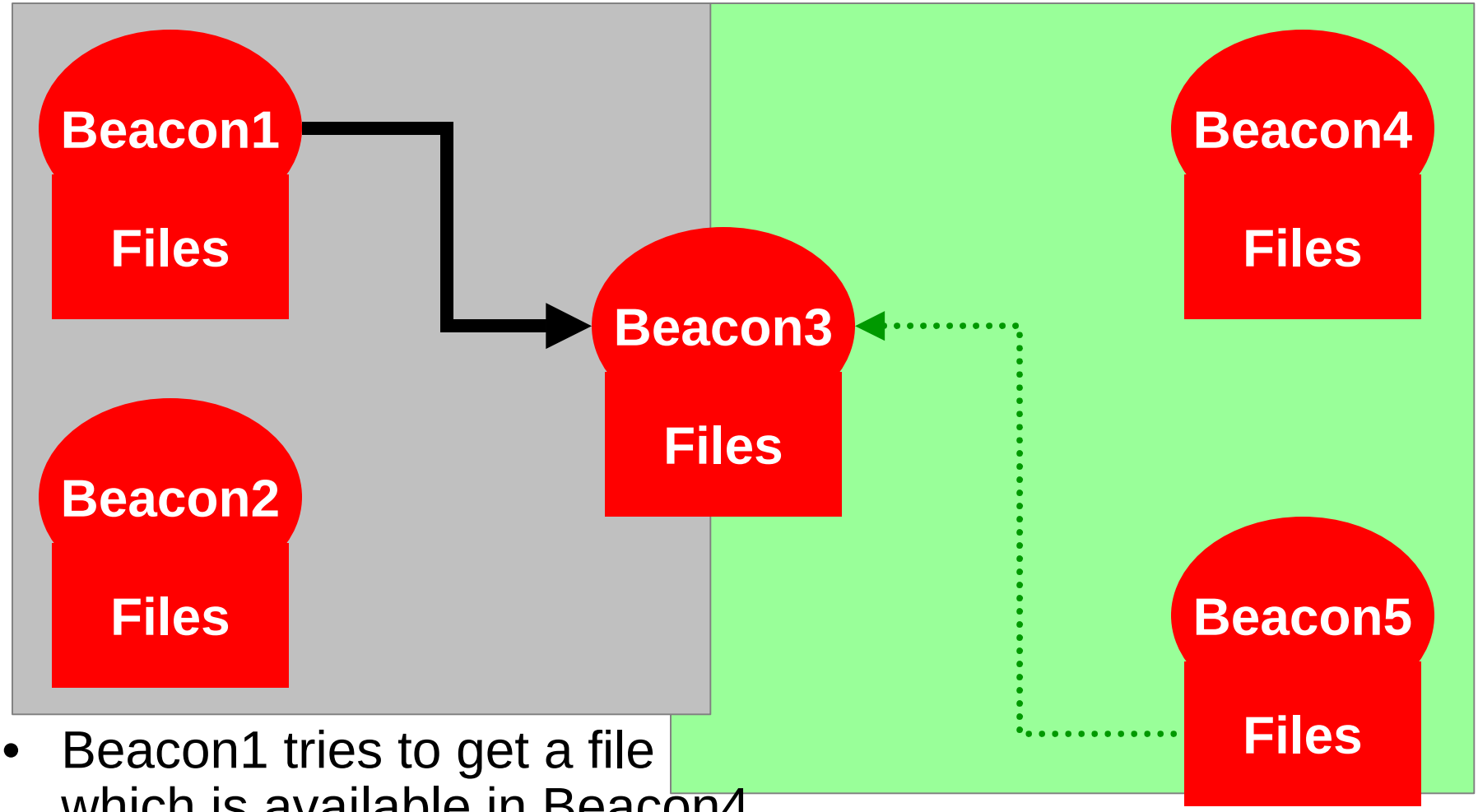
- Beacon1 tries to get a file which is available in Beacon4. Hopcount=2

Home work 3 – Alternative path



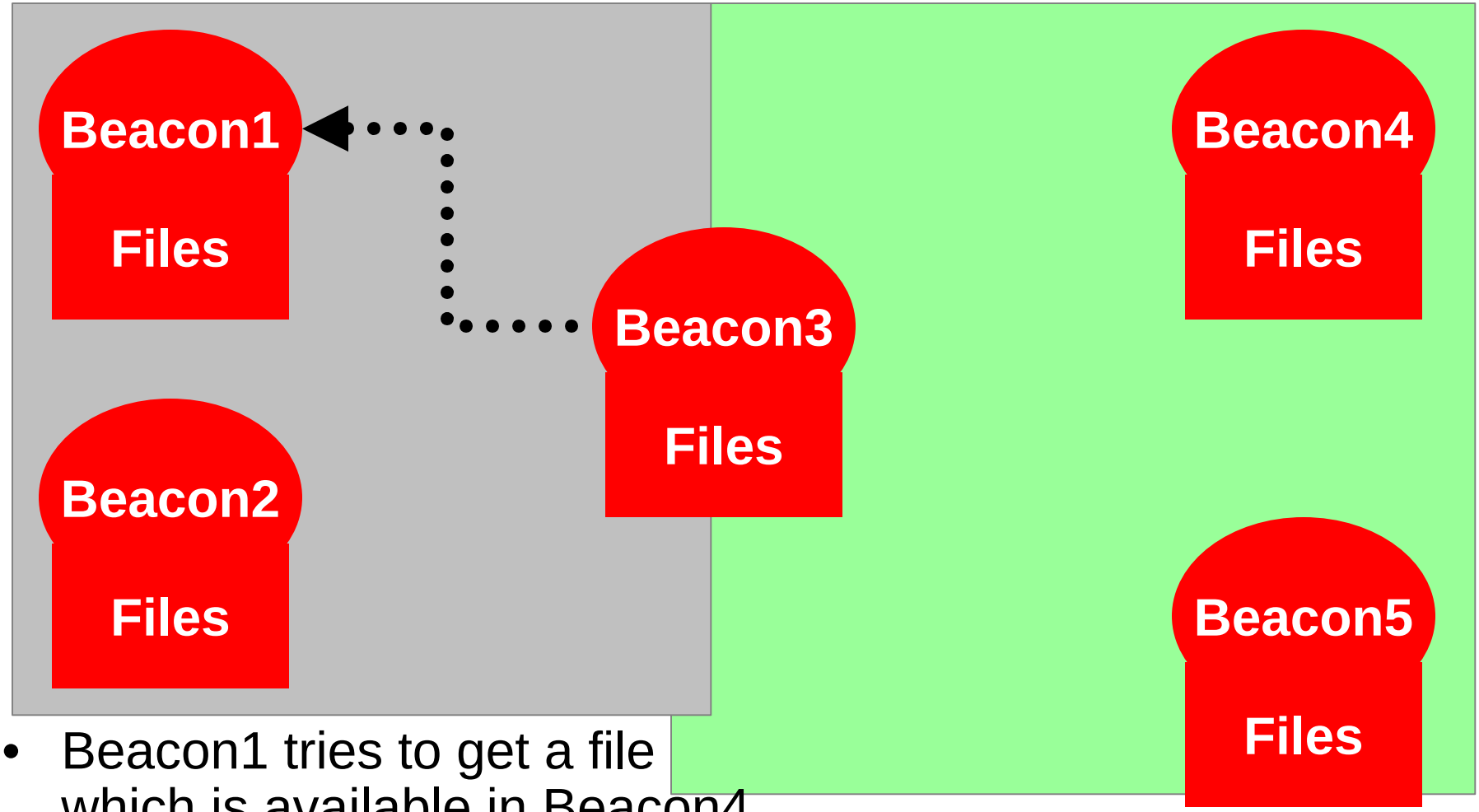
- Beacon1 tries to get a file which is available in Beacon4. Hopcount=2

Home work 3 – Alternative path



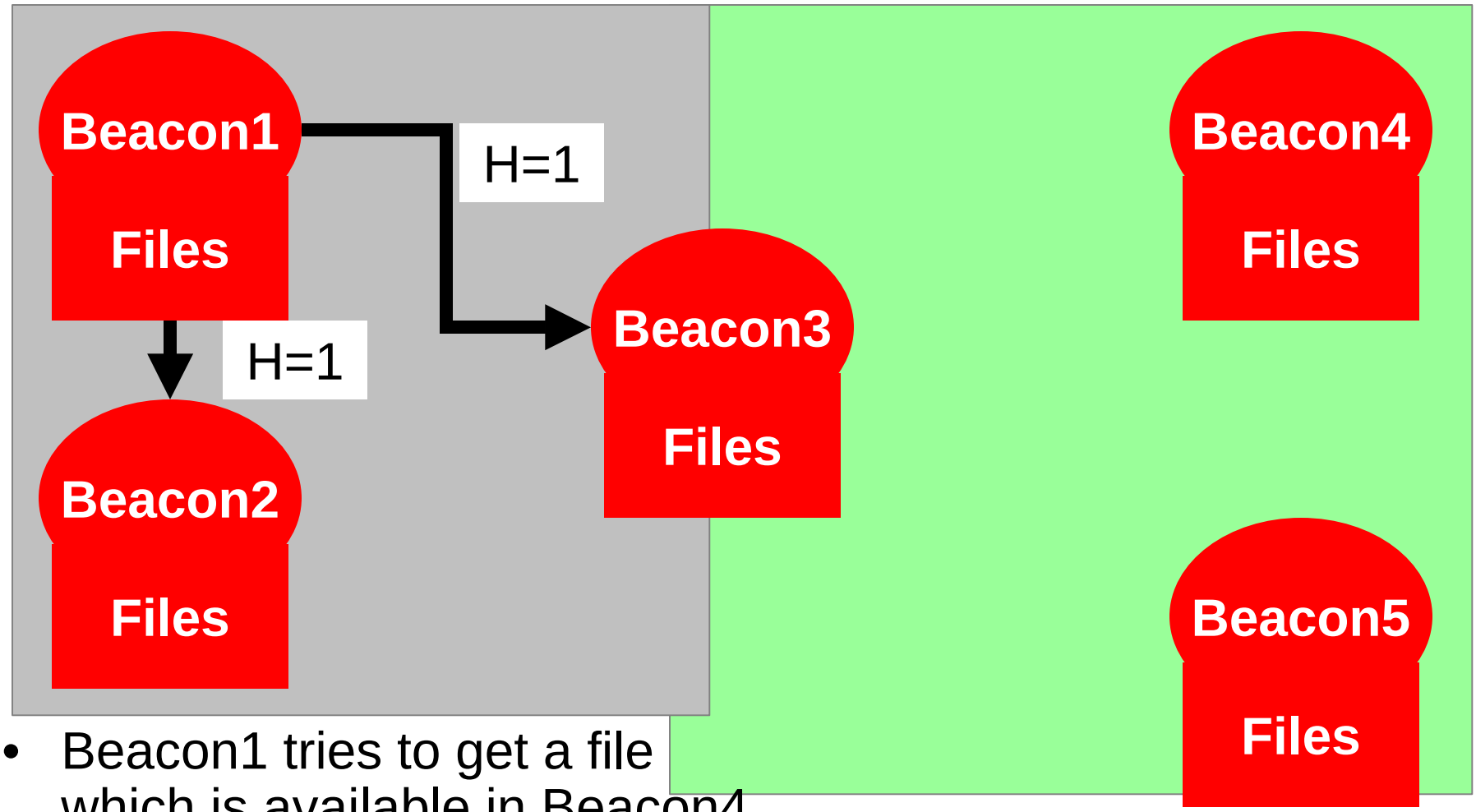
- Beacon1 tries to get a file which is available in Beacon4. Hopcount=2

Home work 3 – Alternative path



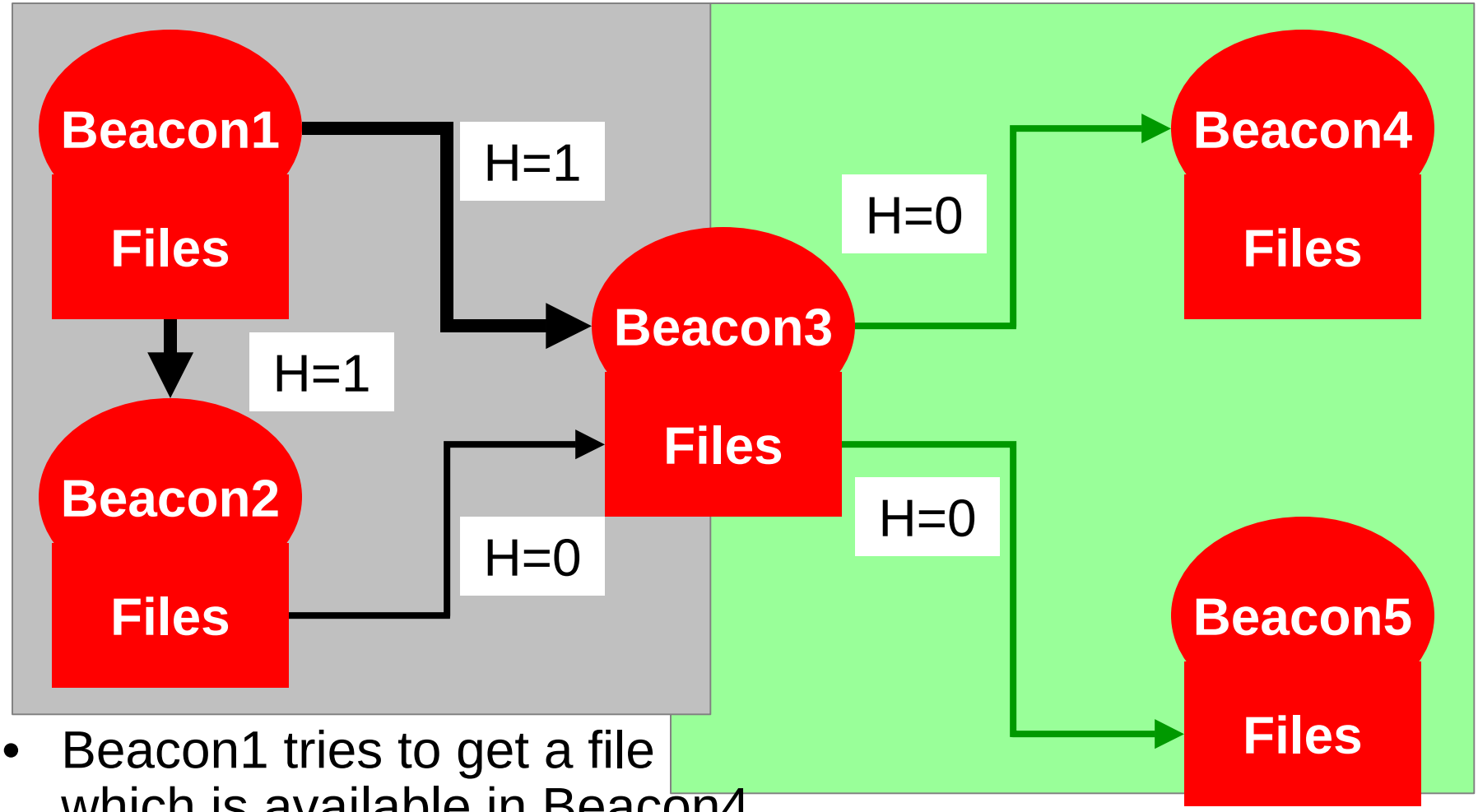
- Beacon1 tries to get a file which is available in Beacon4. Hopcount=2

Home work 3 – Flooding



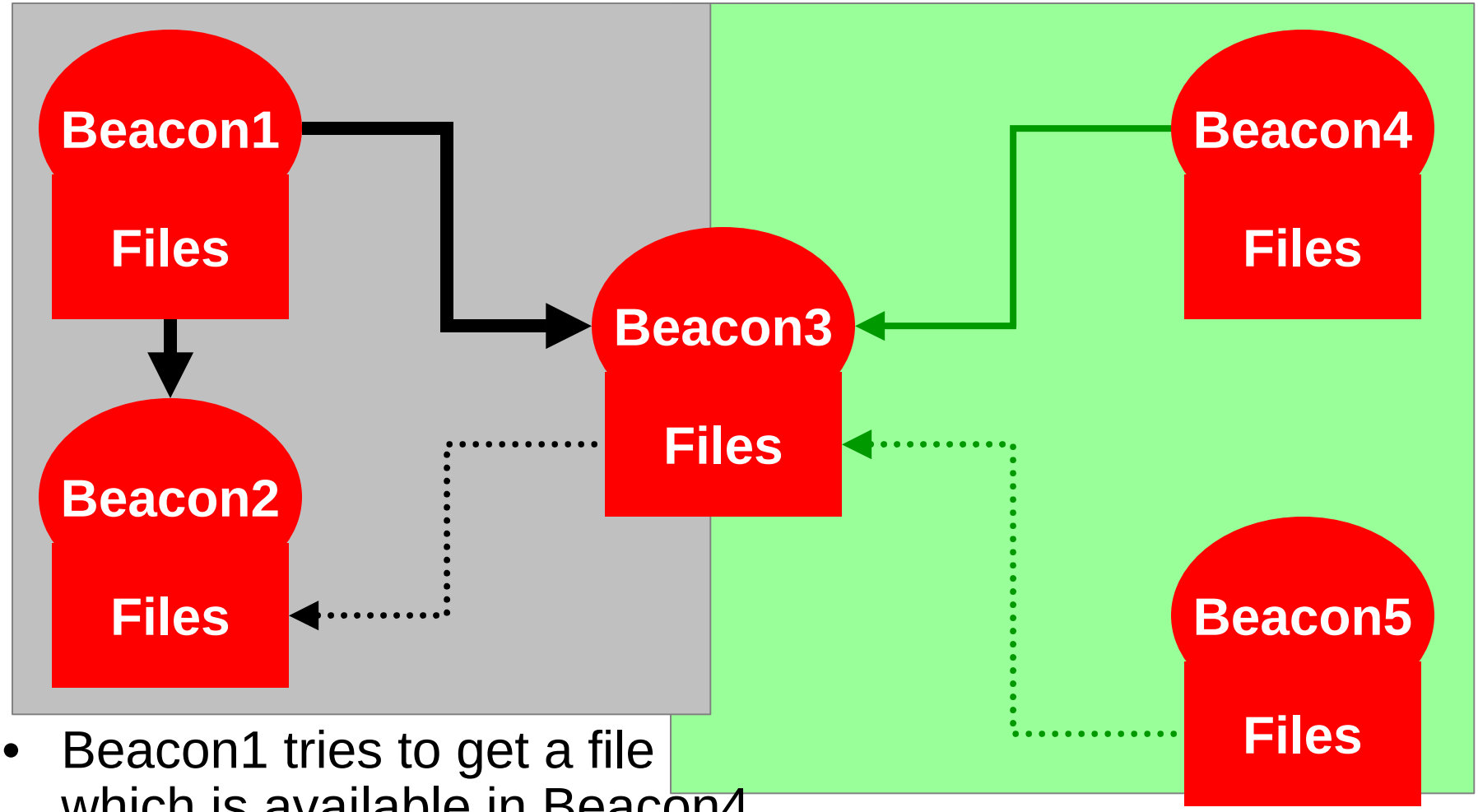
- Beacon1 tries to get a file which is available in Beacon4. Hopcount=2

Home work 3 – Flooding



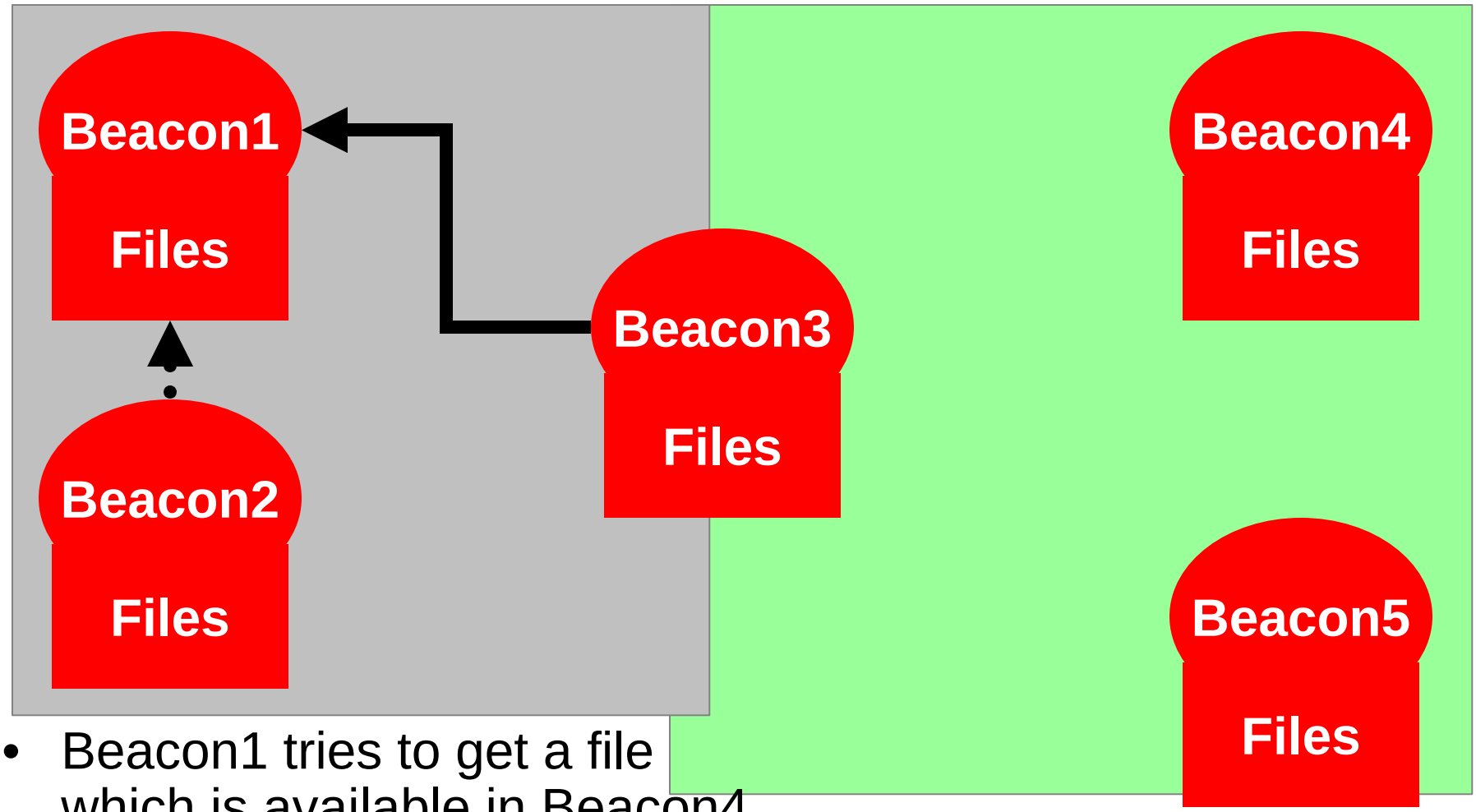
- Beacon1 tries to get a file which is available in Beacon4. Hopcount=2

Home work 3 – Flooding



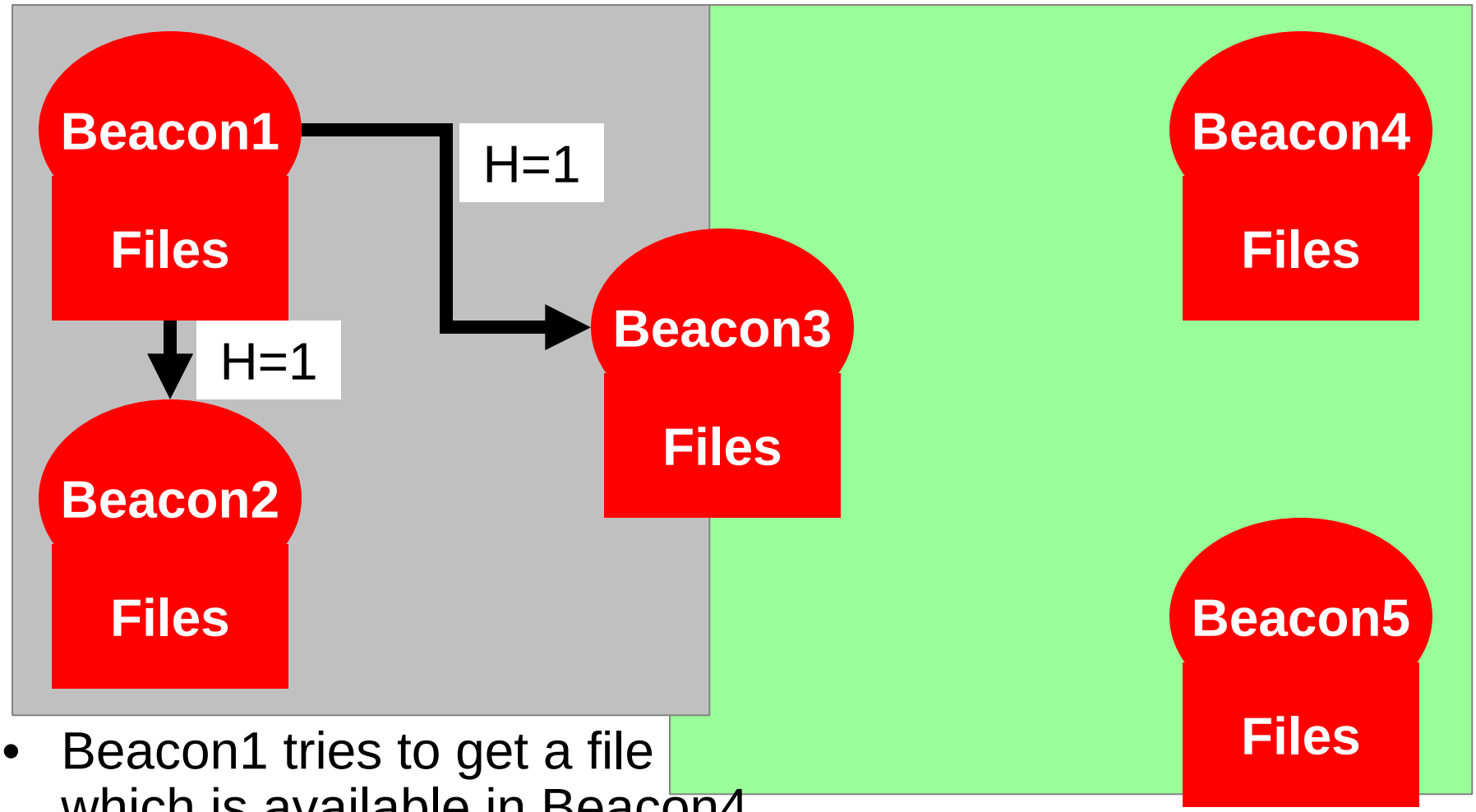
- Beacon1 tries to get a file which is available in Beacon4. Hopcount=2

Home work 3 – Flooding



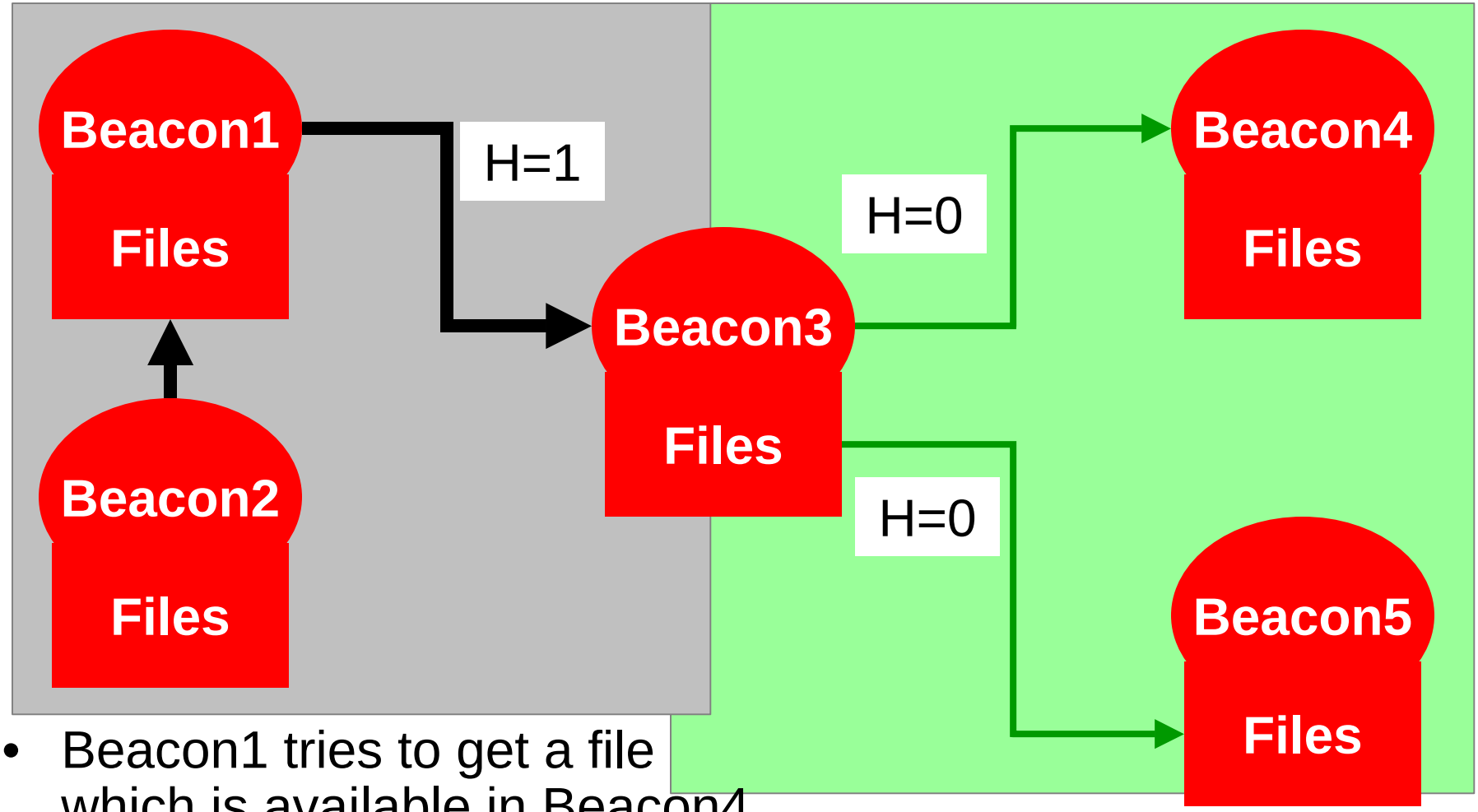
- Beacon1 tries to get a file which is available in Beacon4. Hopcount=2

Home work 3 – Flooding & Replication



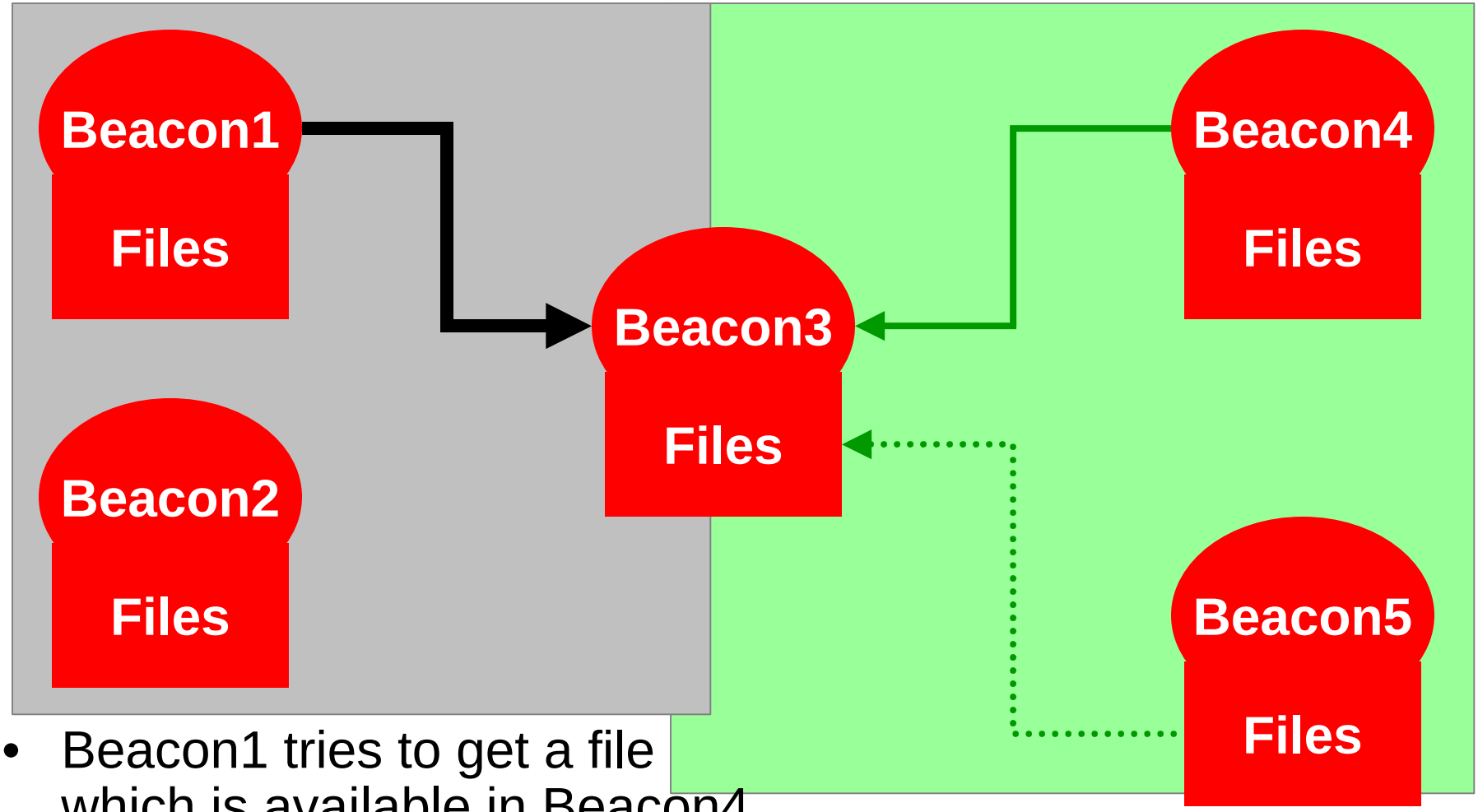
- Beacon1 tries to get a file which is available in Beacon4 and Beacon2. Hopcount=2

Home work 3 – Flooding



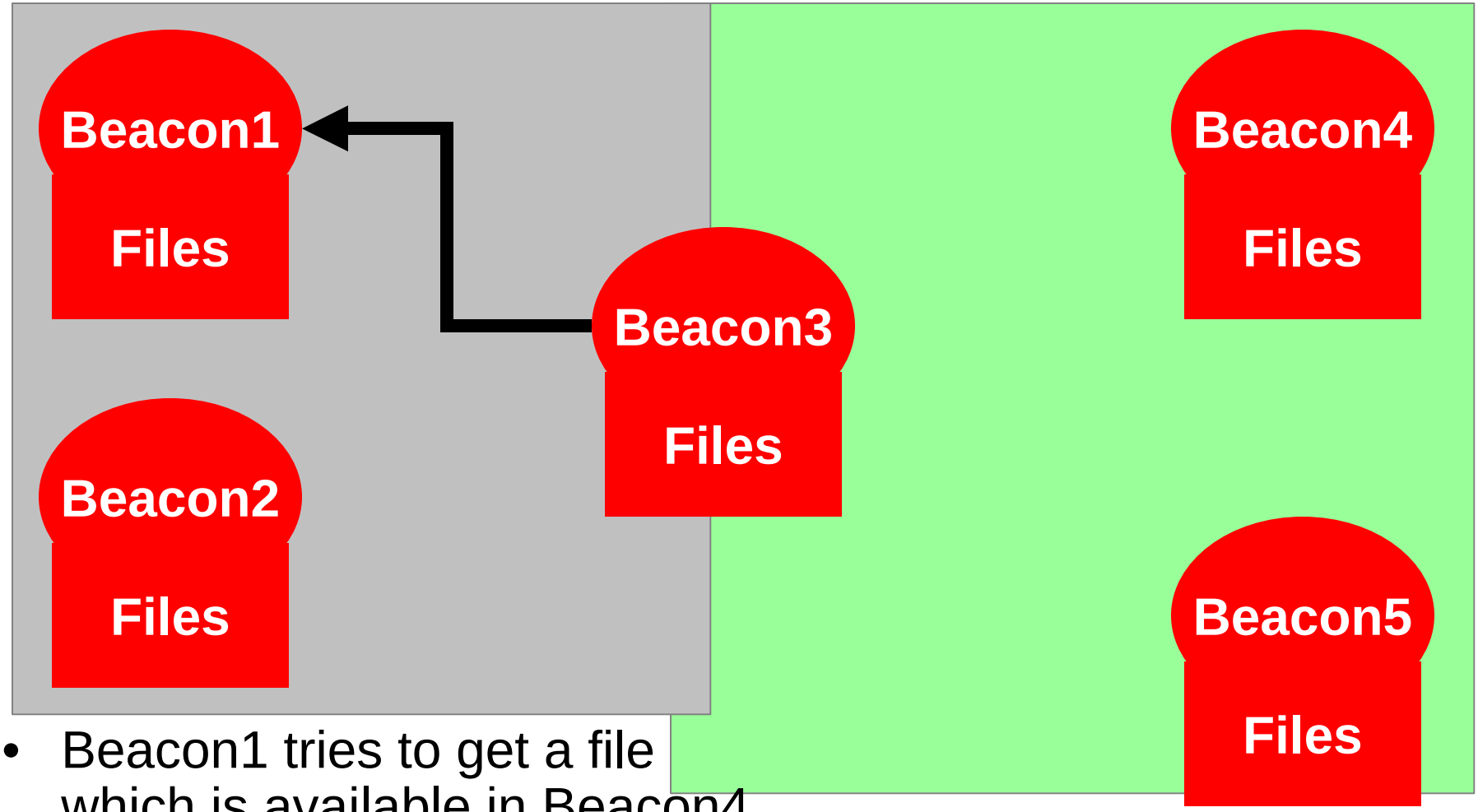
- Beacon1 tries to get a file which is available in Beacon4. Hopcount=2

Home work 3 – Flooding



- Beacon1 tries to get a file which is available in Beacon4. Hopcount=2

Home work 3 – Flooding



- Beacon1 tries to get a file which is available in Beacon4. Hopcount=2

New Service

- Searchget(token, fileName, hopcount)
  - Search for fileName. You specify a upper-bound for the search using hopcount
  - You can use a traditional RPC based approach (Homework 2) or multiway RPC (described in ActiveNames paper)
  - Can use flooding or sequential approach
  - Describe scalability, robustness etc.

Outline

- Replication in the Harp File System, Barbara Liskov, Sanjay Ghemawat, Robert Gruber, Paul Johnson, Liuba Shrira, Michael Williams, MIT
  - Barbara Liskov – first woman to receive Ph.D. in Computer Science in the US (1968, Stanford). Faculty at MIT since 1972

HARP – Highly Available Reliable Persistent

- Provides highly available, reliable storage for files
- Guarantees atomic file operations in spite of concurrency and failure

- Primary copy replication (Eager master)
  - Master server authoritative
  - Replicas – backup servers
  - Updates are sent to “enough” replicas to guarantee fail-safe behavior
- Log structured updates

HARP - Operating Environment

- Bunch of nodes connected by a network
- Fail-stop - Nodes fail by crashing (they don't limp along)
- Network may lose, duplicate messages, deliver messages out of order (but not corrupt messages)
- Loosely synchronized clocks
- Each server is equipped with a small UPS

Replication Method

- Primary-copy method
- One node acts as the primary – clients request from this primary
- Primary contacts backups
- Two-phase modifications
  - Phase 1:
    - Primary informs the backups about modification
    - Backups acknowledge receipt of modification
    - Primary commits and returns
  - Phase 2:
    - Primary informs backups of commit in background

Failover protocol

- Failover protocol – view change
- View change
  - failed node removed from service
  - Recovered node is put back into service
- Primary of new view maybe different from old view
- Clients requests sent to primary of new view
- 2n+1 servers for tolerating n failures

Failover – Contd.

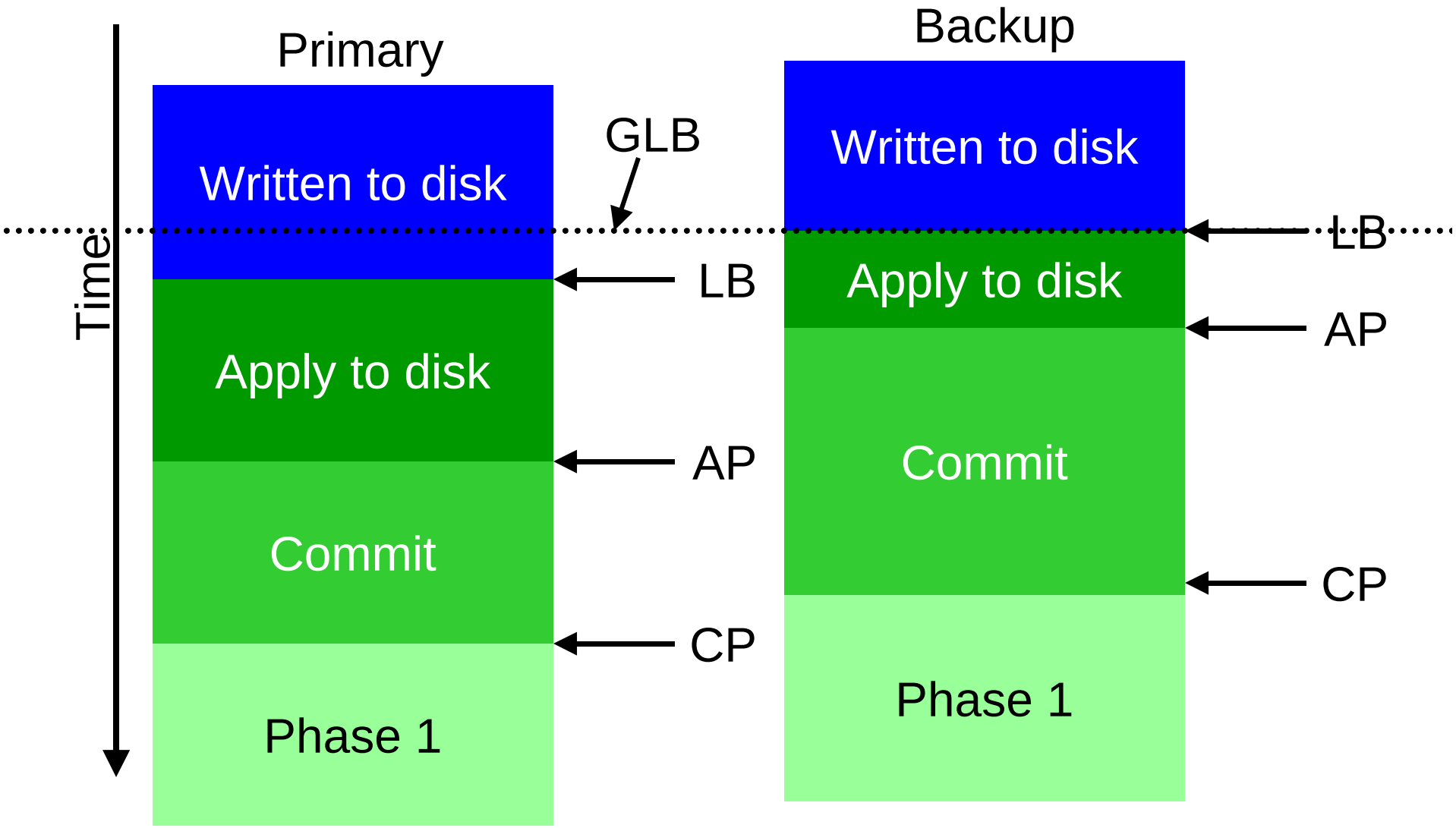
- Store n+1 copies
- N nodes witness view change to make sure only one view is selected

- Each replicate group manages one or more file systems
  - 1 designated primary
  - N designated backups
  - N designated witnesses

Normal case processing

- Primary maintains a log in volatile memory in which it records modification operations
- Some records are phase 1 while others are for operations that have committed
- Primary maintains a commit point (CP); index of the latest committed operation
- An apply process applies committed operations to the file system (AP) asynchronously
- Lower Bound pointer records operations that were written to disk
- Global LB is used to cleanup the logs

Volatile state of the modification logs



Normal case operations

- Operations that do not modify file system state (get file attributes – e.g. ls) entirely in primary. Performed on committed writes.
- Non-modifications can lead to problems in network partitions. Hence they use loosely synchronized clocks and restrict view change after a interval δ since the last message

View change

- Requirements
  - Correctness: Operations must appear to be executed in commit order
    - E.g. create file, list file, delete file expects a certain ordering
  - Reliability: Effects of committed operations must survive single failures and simultaneous failures
  - Availability:
  - Timelines: Fail-over must be done in a timely manner
- Details in paper
- Use event records to avoid problems with multiple applications of event records

Discussion