Three Phase Commit (3PC)

- Assumptions:
  - No network partitioning
  - At any point, at least one site must be up.
  - At most K sites (participants as well as coordinator) can fail
- Phase 1: Obtaining Preliminary Decision: Identical to 2PC Phase 1.
  - Every site is ready to commit if instructed to do so
- Phase 2 of 2PC is split into 2 phases, Phase 2 and Phase 3 of 3PC
  - In phase 2 coordinator makes a decision as in 2PC (called the pre-commit decision) and records it in multiple (at least K) sites
  - In phase 3, coordinator sends commit/abort message to all participating sites,
- Under 3PC, knowledge of pre-commit decision can be used to commit despite coordinator failure
  - Avoids blocking problem as long as < K sites fail
- Drawbacks:
  - higher overheads
  - assumptions may not be satisfied in practice

Three Phase Commit (3PC)

- Assumptions:
  - No network partitioning
  - At any point, at least one site must be up.
  - At most K sites (participants as well as coordinator) can fail
- Phase 1: Obtaining Preliminary Decision: Identical to 2PC Phase 1.
  - Every site is ready to commit if instructed to do so
  - Under 2 PC each site is obligated to wait for decision from coordinator
  - Under 3PC, knowledge of pre-commit decision can be used to commit despite coordinator failure.

3PC: Phase 2. Recording the Preliminary Decision

- Coordinator adds a decision record (<**abort** *T*> or < **precommit** *T*>) in its log and forces record to stable storage.
- Coordinator sends a message to each participant informing it of the decision
- Participant records decision in its log
- If abort decision reached then participant aborts locally
- If pre-commit decision reached then participant replies with <**acknowledge** *T*>

3PC: Phase 3. Recording Decision in the Database

- Executed only if decision in phase 2 was to precommit
- Coordinator collects acknowledgements. It sends <**commit** *T*> message to the participants as soon as it receives K acknowledgements.
- Coordinator adds the record <**commit** *T*> in its log and forces record to stable storage.
- Coordinator sends a message to each participant to <**commit** *T*>
- Participants take appropriate action locally

3PC: Handling Site Failure

- **Site Failure.** Upon recovery, a participating site examines its log and does the following:
  - Log contains <**commit** *T*> record: no action
  - Log contains <**abort** *T*> record: no action
  - Log contains <**ready** *T*> record, but no <**abort** *T*> or <**precommit** *T*> record: site consults Ci to determine the fate of *T*.
    - if Ci says *T* aborted, site executes **undo** (*T*) (and writes <**abort** *T*> record)
    - if Ci says T committed, site executes **redo** (*T*) (and writes < **commit** *T*> record)
    - if *c* says *T* committed, site resumes the protocol from receipt of **precommit** *T* message (thus recording <**precommit** *T*> in the log, and sending **acknowledge** *T* message sent to coordinator).

3PC: Handling Site Failure (Cont.)

- Log contains <**precommit** *T*> record, but no <**abort** *T*> or <**commit** *T*>: site consults Ci to determine the fate of *T*.
  - if *Ci* says *T* aborted, site executes **undo** (*T*)
  - if Ci says *T* committed, site executes **redo** (*T*)
  - if Ci says *T* still in precommit state, site resumes protocol at this point
- Log contains no <**ready** *T*> record for a transaction *T*: site executes **undo** (*T*) writes <**abort** *T*> record

Figure 19.02

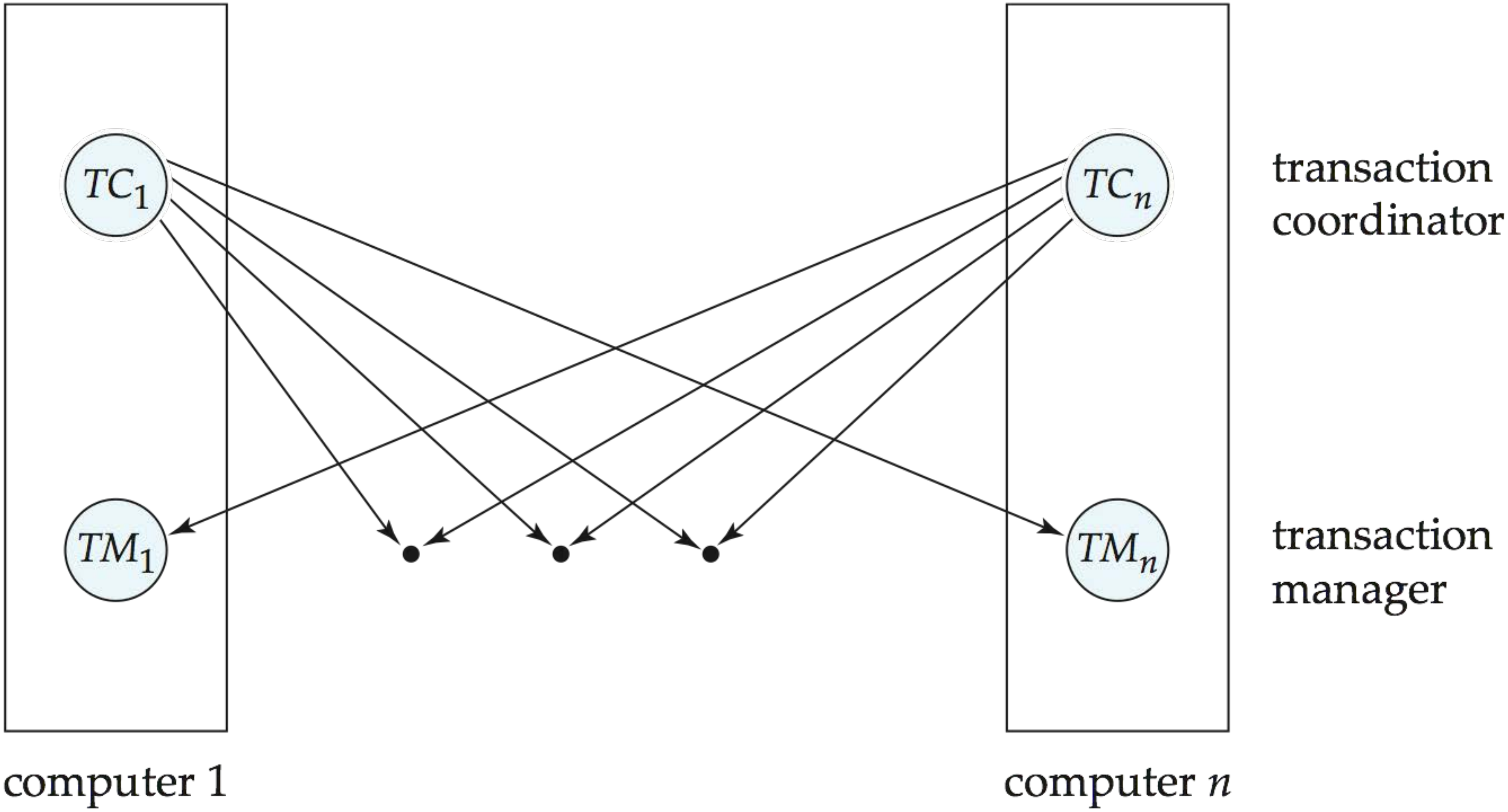

TC_1
TC_n
transaction coordinator
TM_1
TM_n
transaction manager
computer 1
computer *n*

Figure 19.03

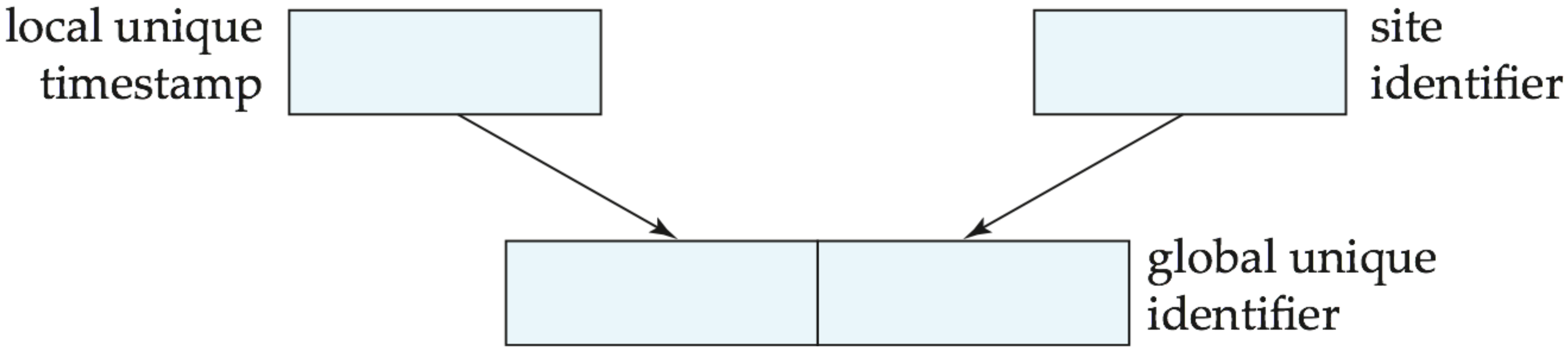

local unique timestamp
site identifier
global unique identifier

Figure 19.04

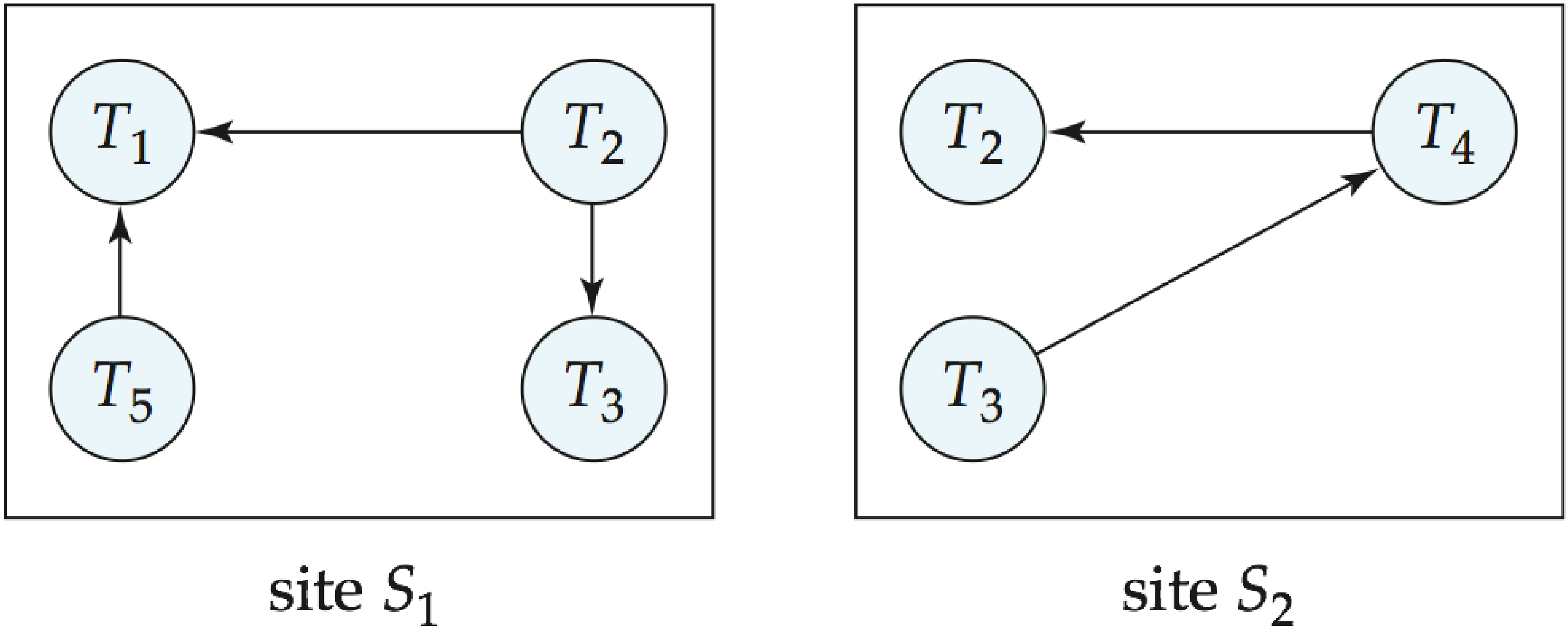


Figure 19.05

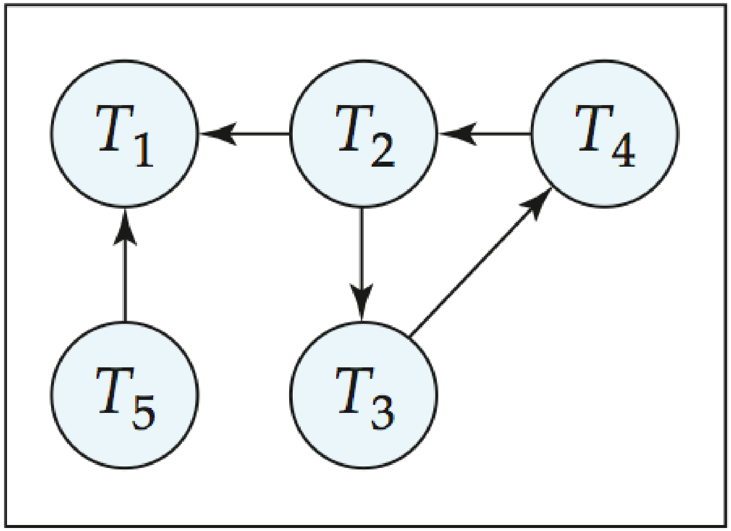


Figure 19.06

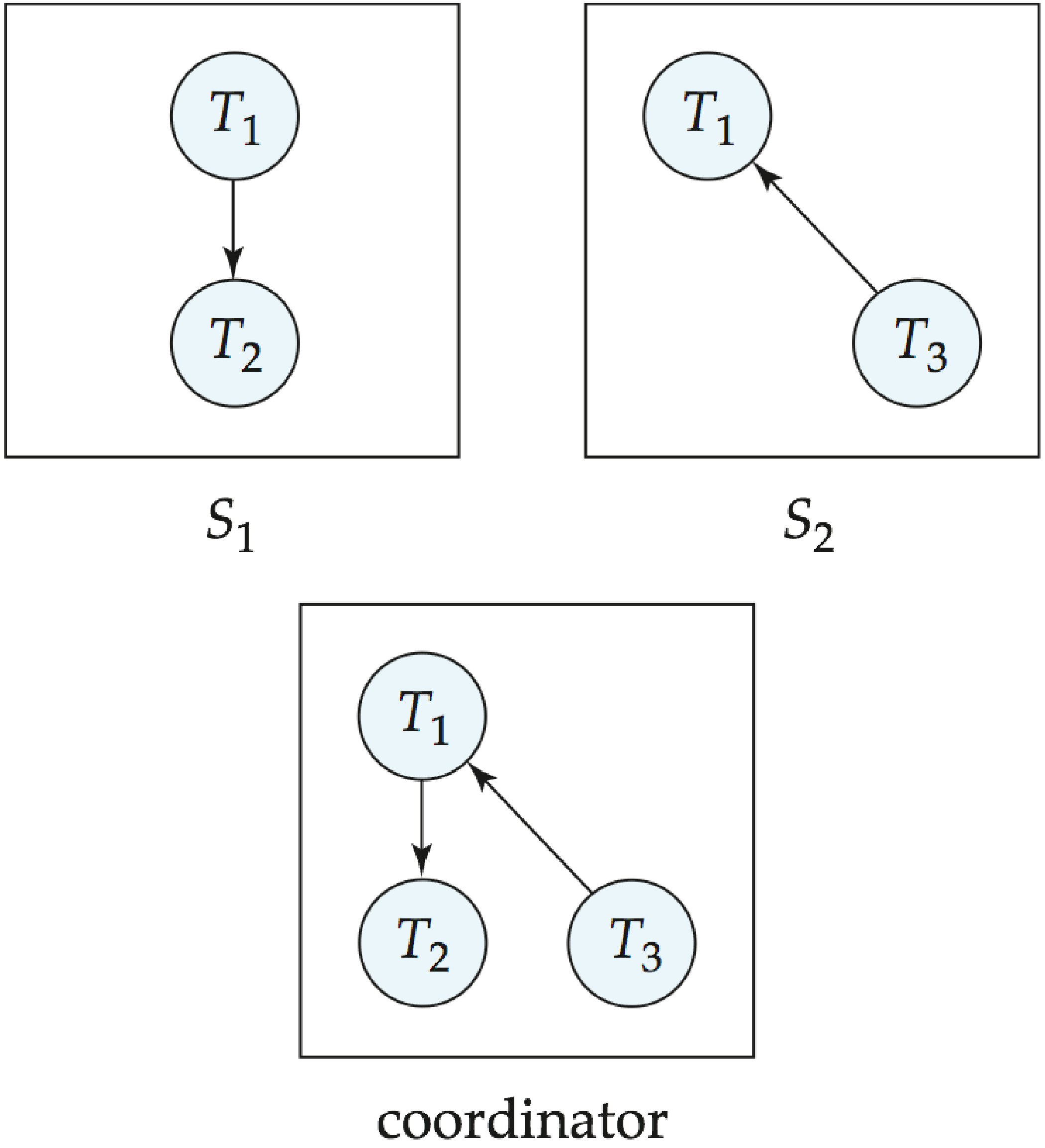
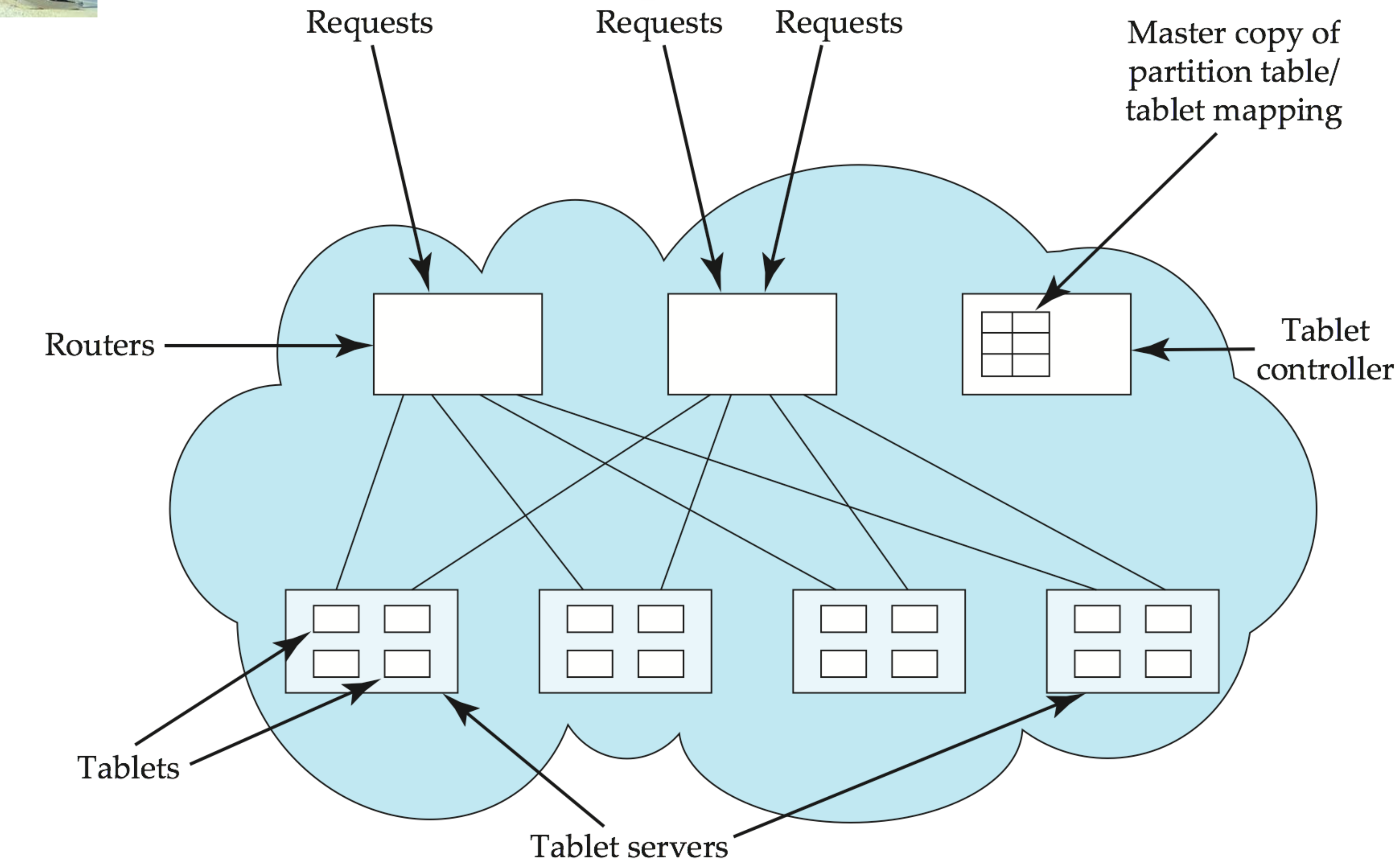


Figure 19.07


Requests
Requests
Requests
Master copy of partition table/ tablet mapping
Routers
Tablet controller
Tablets
Tablet servers