Blockchain Unit-3 Important Questions With Answers According to RGPV Exam

1. Permissioned Blockchain ⭐

Introduction

Permissioned blockchain is mainly used in companies, banks, hospitals and government systems where privacy and control are important.

Definition

A permissioned blockchain is a blockchain network in which only authorized and verified users can participate.

Why It Is Needed

Public blockchain is open for everyone, but enterprises need:

- Privacy
- Fast transactions
- Controlled access
- Known participants

Easy Explanation

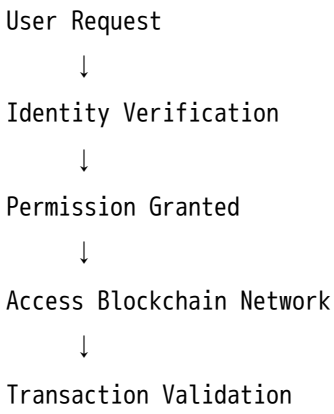
It works like a college ERP system. Only students, teachers and admin can login. Outsiders cannot access it.

Working

1. User requests to join network.
2. Authority checks identity.
3. Permission is given.

4. User can read/write/validate based on role.
5. Transactions are verified by selected nodes.

Diagram



Advantages

- High privacy
- Fast speed
- Better control
- Suitable for business
- Low energy consumption

Disadvantages

- Less decentralization
- Depends on trusted authority
- Limited transparency

Applications

- Banking
- Healthcare
- Supply chain

- Land records
- Certificate verification

Important Keywords

Authorized users, access control, privacy, enterprise blockchain, known participants

Conclusion

Permissioned blockchain is best for enterprises because it provides security, privacy and controlled access.

---

2. Paxos Algorithm ⭐

Introduction

Paxos is a consensus algorithm used in distributed systems to make all nodes agree on one value.

Definition

Paxos is a fault-tolerant consensus algorithm that helps distributed nodes agree even if some nodes fail.

Why It Is Needed

In distributed systems, many nodes work together. If one node fails, the system should still decide correctly.

Easy Explanation

Suppose 5 teachers need to select one exam date. Even if one teacher is absent, majority can decide one final date. This is like Paxos.

Main Roles

Role	Work
Proposer	Suggests value
Acceptor	Accepts/rejects proposal
Learner	Learns final accepted value

Working

1. Proposer sends proposal.
2. Acceptors receive proposal.
3. Majority acceptors approve it.
4. Accepted value becomes final.
5. Learners learn final decision.

Diagram

```
Proposer
   ↓
Proposal Sent
   ↓
Acceptors Vote
   ↓
Majority Accepted
   ↓
Final Decision
```

Advantages

- Fault tolerant
- Reliable
- Works even with node failure

Disadvantages

- Difficult to understand

- Complex implementation
- More communication required

Applications

- Distributed databases
- Permissioned blockchain
- Cloud systems

Important Keywords

Proposer, Acceptor, Learner, Majority, Fault tolerance

Conclusion

Paxos provides reliable agreement among distributed nodes and is important for enterprise blockchain consensus.

---

3. RAFT Consensus ⭐

Introduction

RAFT is a simple consensus algorithm designed to be easier than Paxos.

Definition

RAFT is a leader-based consensus algorithm where one leader manages log replication among nodes.

Why It Is Needed

It is needed to maintain the same data across multiple servers in a simple and understandable way.

Easy Explanation

RAFT works like a class monitor system. One monitor gives instructions, and all students follow.

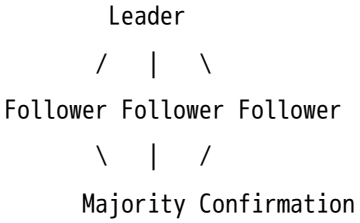
Main Roles

Role	Work
Leader	Handles requests
Follower	Follows leader
Candidate	Tries to become leader

Working

1. Nodes elect a leader.
2. Client sends request to leader.
3. Leader sends data/log to followers.
4. Followers copy the log.
5. Majority confirms.
6. Data becomes committed.

Diagram



Advantages

- Easy to understand
- Simple implementation
- Good for permissioned blockchain
- Reliable replication

Disadvantages

- Leader failure causes delay
- Not suitable for open public networks

Applications

- Distributed databases
- Enterprise blockchain
- Cluster management

Important Keywords

Leader, follower, candidate, log replication, majority

Conclusion

RAFT is a simple and effective consensus algorithm used in closed and permissioned systems.

---

4. Byzantine General Problem ⭐

Introduction

This problem explains how distributed systems can reach agreement when some participants may lie or behave maliciously.

Definition

The Byzantine General Problem is a problem of achieving agreement among distributed nodes when some nodes may be faulty or dishonest.

Why It Is Needed

Blockchain networks must work correctly even if some nodes send false information.

Easy Explanation

Imagine army generals surrounding a city. They must attack together. But some generals may send wrong messages. The loyal generals must still take the correct decision.

Diagram

```
General A → Attack
General B → Attack
General C → Retreat  (Dishonest)
        ↓
Need Correct Agreement
```

In Blockchain

Some nodes may:

- Send fake transactions
- Reject valid blocks
- Spread wrong information
- Try to attack consensus

Advantages of Studying This Problem

- Helps design secure consensus
- Handles malicious nodes
- Improves blockchain reliability

Important Keywords

Malicious node, faulty node, agreement, trust, distributed system

Conclusion

Byzantine General Problem is the base concept behind Byzantine Fault Tolerant blockchain systems.

---

5. Byzantine Fault Tolerant System (BFT) ⭐

Introduction

BFT systems continue working correctly even when some nodes are faulty or malicious.

Definition

A Byzantine Fault Tolerant system is a system that can reach correct agreement even if some nodes behave incorrectly.

Why It Is Needed

In blockchain, not every node can be trusted. BFT protects the system from dishonest nodes.

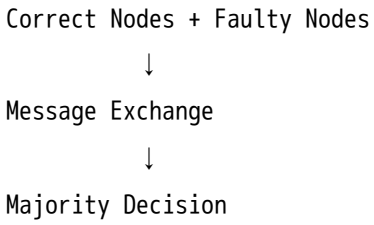
Easy Explanation

Even if some students in a group give wrong answers, the correct majority can still find the right answer.

Working

1. Nodes exchange messages.
2. Each node checks received information.
3. Faulty messages are ignored.
4. Majority decision is accepted.
5. System continues safely.

Diagram



```
          ↓
Correct Consensus
```

Advantages

- Handles malicious nodes
- High reliability
- Strong security
- Useful in permissioned blockchain

Disadvantages

- High communication overhead
- Complex design
- Slower in large networks

Applications

- Banking blockchain
- Hyperledger Fabric
- Enterprise systems
- Secure distributed databases

Important Keywords

Fault tolerance, malicious nodes, majority agreement, Byzantine failure

Conclusion

BFT is very important for secure enterprise blockchain where some nodes may fail or behave dishonestly.

---

6. State Machine Replication ⭐

Introduction

State Machine Replication keeps the same copy of data on all nodes.

Definition

State Machine Replication is a technique where multiple nodes execute the same operations in the same order to maintain identical state.

Why It Is Needed

It is needed for consistency. Every blockchain node should have the same ledger state.

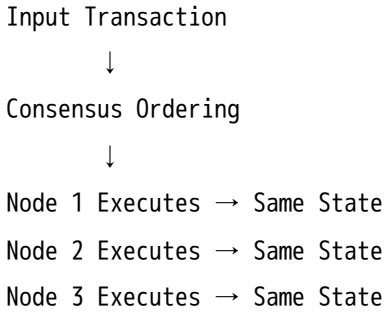
Easy Explanation

Suppose 3 notebooks have the same bank balance record. If ₹500 is added, all notebooks must update equally.

Working

1. Client sends request.
2. Request is ordered by consensus.
3. All nodes execute same operation.
4. All nodes reach same result.
5. System remains consistent.

Diagram



Advantages

- Data consistency
- Fault tolerance
- Reliability
- Useful for blockchain ledgers

Disadvantages

- Synchronization overhead
- Needs proper ordering
- Complex in large networks

Applications

- Blockchain ledger
- Banking systems
- Distributed databases

Important Keywords

Same state, same order, replication, consistency, fault tolerance

Conclusion

State Machine Replication ensures that all blockchain nodes maintain the same correct data.

---

7. Design Issues for Permissioned Blockchain ⭐

Introduction

Designing permissioned blockchain is not simple because enterprises need privacy, speed, security and control together.

Definition

Design issues are the important challenges and decisions faced while building a permissioned blockchain system.

Main Design Issues

Issue	Explanation
Access Control	Who can join network
Identity Management	Verifying user identity
Consensus Selection	Choosing Paxos, RAFT, BFT etc.
Privacy	Protecting sensitive data
Scalability	Handling more users
Performance	Fast transaction processing
Governance	Rules for managing network
Smart Contract Execution	Safe execution of contracts

Easy Explanation

A bank blockchain must decide:

- Who can access?
- Who can validate?
- How data is protected?
- What happens if node fails?

Diagram

```
Permissioned Blockchain Design
        ↓
Access + Privacy + Consensus + Scalability + Governance
```

Advantages of Proper Design

- Better security

- High speed
- Better privacy
- Easy management

Conclusion

Good design is necessary for successful enterprise blockchain implementation.

---

8. Consensus in Closed Environment ⭐

Introduction

Closed environment means only known and authorized users can participate.

Definition

Consensus in closed environment is the process where trusted and permissioned nodes agree on a common decision.

Why It Is Needed

Enterprise blockchain does not need heavy mining like Bitcoin because participants are known.

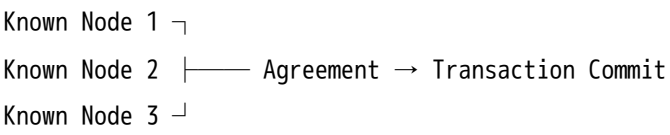
Easy Explanation

In a company meeting, only selected members vote. Since everyone is known, decision becomes faster.

Working

1. Authorized nodes participate.
2. Transaction proposal is made.
3. Nodes validate transaction.
4. Nodes vote or replicate logs.
5. Majority agreement is reached.
6. Transaction is committed.

Diagram



Consensus Models

- Paxos
- RAFT
- PBFT
- BFT algorithms

Advantages

- Faster than public blockchain
- Low energy consumption
- Better privacy
- Suitable for enterprise use

Disadvantages

- Less decentralization
- Needs trust in members

Important Keywords

Known nodes, closed network, permissioned consensus, fast agreement

Conclusion

Closed environment consensus is efficient and suitable for enterprise blockchain networks.

---

9. Lamport-Shostak-Pease BFT Algorithm ⭐

Introduction

This algorithm was designed to solve the Byzantine Generals Problem.

Definition

Lamport-Shostak-Pease BFT Algorithm is a Byzantine agreement algorithm that helps loyal nodes reach the same decision even if some nodes are traitors.

Why It Is Needed

It is needed when some nodes may send wrong or fake messages.

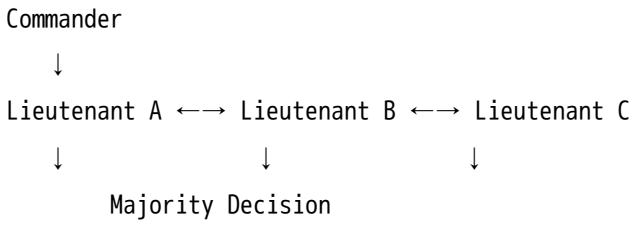
Easy Explanation

A commander sends order to lieutenants. Some commanders or lieutenants may lie. Still, all loyal lieutenants must decide the same action.

Working

1. Commander sends message to all lieutenants.
2. Lieutenants forward received messages to each other.
3. Each lieutenant collects messages.
4. Majority value is selected.
5. Loyal nodes take the same decision.

Diagram



Key Idea

If some nodes are faulty, majority voting helps correct nodes agree.

Advantages

- Solves Byzantine agreement
- Handles dishonest nodes
- Foundation for BFT systems

Disadvantages

- Many messages are exchanged
- High communication cost
- Complex for large networks

Applications

- Secure distributed systems
- Blockchain consensus
- Fault-tolerant networks

Important Keywords

Commander, lieutenants, traitor, majority, Byzantine agreement

Conclusion

Lamport-Shostak-Pease algorithm is a basic and important algorithm for Byzantine fault-tolerant consensus.

---

10. BFT over Asynchronous Systems ⭐

Introduction

Asynchronous systems are systems where message delivery time is not fixed.

Definition

BFT over asynchronous systems means achieving Byzantine fault tolerance when messages may be delayed or arrive at different times.

Why It Is Needed

Real-world networks are not perfect. Messages can be delayed due to internet speed, server load or network failure.

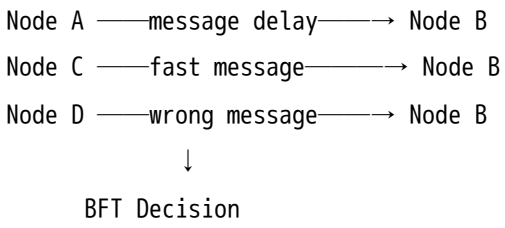
Easy Explanation

Suppose students are sending answers on WhatsApp, but messages arrive late. Still, the group must decide correctly.

Working

1. Nodes send messages.
2. Some messages may arrive late.
3. Nodes wait for enough valid messages.
4. Faulty messages are ignored.
5. Agreement is reached using BFT rules.

Diagram



Challenges

- No fixed message time
- Network delay
- Faulty nodes
- Message loss
- Complex agreement

Advantages

- Works in real-world networks
- Handles malicious nodes
- Provides strong reliability

Disadvantages

- Difficult implementation
- High message overhead
- Slower than simple consensus

Applications

- Enterprise blockchain
- Banking networks
- Distributed cloud systems
- Secure databases

Important Keywords

Asynchronous network, message delay, Byzantine fault, reliable consensus

Conclusion

BFT over asynchronous systems is important because real blockchain networks must work correctly even with delay and faulty nodes.

🔥 Most Important 7-Mark Questions

1. Explain permissioned blockchain with features.
2. Explain Paxos algorithm.
3. Explain RAFT consensus algorithm.
4. Explain Byzantine General Problem.
5. What is Byzantine Fault Tolerant system?
6. Explain State Machine Replication.
7. Explain design issues of permissioned blockchain.
8. Explain distributed consensus in closed environment.
9. Explain Lamport-Shostak-Pease BFT Algorithm.
10. Explain BFT over asynchronous systems.

⭐ Most Important 14-Mark Questions

1. Explain permissioned blockchain with model, design issues and use cases.
2. Explain Paxos and RAFT consensus algorithms with diagrams.
3. Explain Byzantine General Problem and Byzantine Fault Tolerant system.
4. Explain State Machine Replication and consensus in closed environment.
5. Explain Lamport-Shostak-Pease BFT Algorithm in detail.
6. Explain consensus models for permissioned blockchain.
7. Explain BFT over asynchronous systems with challenges.
8. Compare Paxos, RAFT and BFT consensus models.

🎯 PYQ-Based Expected Questions

Very Important

- Permissioned Blockchain
- Paxos Algorithm
- RAFT Consensus
- Byzantine General Problem
- BFT System

High Probability

- State Machine Replication
- Design Issues
- Consensus in Closed Environment

Medium Probability

- Lamport-Shostak-Pease Algorithm
- BFT over Asynchronous Systems

⚡ One-Night Revision Notes

```
Permissioned Blockchain = Only authorized users

Paxos = Proposal → Majority → Agreement

RAFT = Leader → Followers → Log Replication

Byzantine Problem = Some nodes may lie

BFT = System works even with faulty nodes

State Machine Replication = Same operation + Same order = Same state

Closed Consensus = Known nodes reach agreement

Lamport Algorithm = Commander + Lieutenants + Majority

Asynchronous BFT = Consensus with delayed messages
```

📌 Smart Study Plan

First Priority

Study these first:

1. Permissioned Blockchain
2. Paxos Algorithm
3. RAFT Consensus
4. Byzantine General Problem
5. BFT System

Second Priority

Study:

1. State Machine Replication
2. Design Issues
3. Consensus in Closed Environment

Last Priority

Revise:

1. Lamport-Shostak-Pease Algorithm
2. BFT over Asynchronous Systems

🧠 Memory Tricks

```
Paxos = Propose → Accept → Learn
RAFT = Leader → Followers
BFT = Bad nodes present, still system works
```

SMR = Same Method Repeated

Permissioned = Permission Required

Final exam answer pattern:

Definition → Need → Diagram → Working → Advantages → Applications → Conclusion