Programme Name and Semester: MCA, 4th Semester
Course Name (Course Code): **Introduction to Blockchain Technology & Application**
Course Code: **[MCA403B]**
Academic Session: 2023-2024

Study Material-2

(Introduction to Blockchain Technology & Application [MCA403B])

Module: 2: Basic Distributed Computing

- Atomic Broadcast
- Consensus
- Byzantine Models of fault tolerance

In the context of distributed systems and computer networks, broadcasting refers to a communication method where a single sender transmits a message to all potential receivers or nodes in the system. The purpose of broadcasting is to disseminate information widely to a group of recipients or to the entire network. The term "broadcast" is often used to describe a one-to-all communication pattern.

There are different types of broadcasting mechanisms based on how the messages are delivered and who the intended recipients are. Here are three common types:

1. **Unicast:**
    - **Description:** In unicast, a message is sent from one sender to a specific recipient. It's a one-to-one communication.
    - **Use Case:** Unicast is suitable for point-to-point communication where a message is intended for a particular node or recipient.
    - **Advantages:**
        1. **Point-to-Point Communication:** Unicast is well-suited for one-to-one communication, making it efficient for scenarios where a message is intended for a specific recipient.
        2. **Addressing Specific Nodes:** Since unicast targets a specific recipient, it allows for precise targeting of messages to particular nodes in the network.
    - **Disadvantages:**

        1. **Scalability:** Unicast becomes less efficient as the number of recipients increases since a separate message must be sent to each recipient individually.

        2. **Bandwidth Usage:** Unicast can lead to higher bandwidth usage in scenarios where the same information needs to be delivered to multiple nodes individually.
2. **Multicast:**
    - **Description:** Multicast involves sending a message from one sender to multiple, but not necessarily all, recipients. It's a one-to-many communication.

- **Use Case:** Multicast is useful when a message needs to be delivered to a specific group of nodes, and not necessarily to all nodes in the system.
- **Advantages:**

1. **Efficient One-to-Many Communication:** Multicast allows a single sender to efficiently reach multiple recipients simultaneously, reducing network congestion compared to unicast.
2. **Resource Savings:** Multicast can save network resources and improve scalability by delivering a single copy of a message to multiple recipients instead of sending individual copies.

- **Disadvantages:**

1. **Limited Support:** Not all networks and devices may support multicast, and its use can be restricted in some network configurations.
2. **Security Concerns:** Multicast communication may pose security challenges, especially if not properly configured, as it could be susceptible to unauthorized access.

3. **Broadcast:**
   - **Description:** In broadcast, a message is sent from one sender to all potential recipients in the system. It's a one-to-all communication.
   - **Use Case:** Broadcast is applicable when a message needs to be disseminated to every node in the system.
   - **Advantages:**
     1. **Simplicity:** Broadcasting is a simple and straightforward way to communicate a message to all nodes in the network simultaneously.
     2. **No Addressing Overhead:** Broadcast eliminates the need for addressing each recipient individually, simplifying the communication process.
   - **Disadvantages:**
     1. **Network Congestion:** Broadcast can lead to network congestion, especially in large networks, as every node needs to process the broadcasted message.
     2. **Security Risks:** Broadcast messages are accessible to all nodes in the network, potentially raising security concerns, especially if sensitive information is transmitted.

Each type of broadcasting has its own set of advantages and use cases, and the choice depends on the requirements of the distributed system. It's worth noting that the terms unicast, multicast, and broadcast are often used in the context of network communication and distributed systems, but they can also be applied to various communication protocols and technologies. Additionally, within the context of consensus and atomic broadcast, the term "broadcast" may refer specifically to atomic broadcast, where there is a guarantee of the order in which messages are delivered to all nodes in the system.

In a broadcast communication, the sender does not need to address each individual recipient separately; instead, the message is implicitly directed to all nodes. The challenge in broadcasting is to ensure that the message reaches all intended recipients in a reliable and efficient manner. Broadcasting is commonly used in various networking protocols and distributed systems for tasks such as disseminating configuration updates, synchronization, or distributing events. It's worth noting that in certain contexts, such as consensus algorithms or atomic broadcast, the term "broadcast" may specifically refer to a more controlled and ordered dissemination of messages across a distributed system.

● Atomic Broadcast

Atomic Broadcast is a distributed computing concept that refers to the reliable broadcasting of messages across a network of nodes in such a way that all nodes agree on the order of the messages. The term "atomic" here does not refer to atomicity in the context of databases, but rather to the idea that the broadcast is an indivisible or atomic operation from the perspective of the participating nodes.

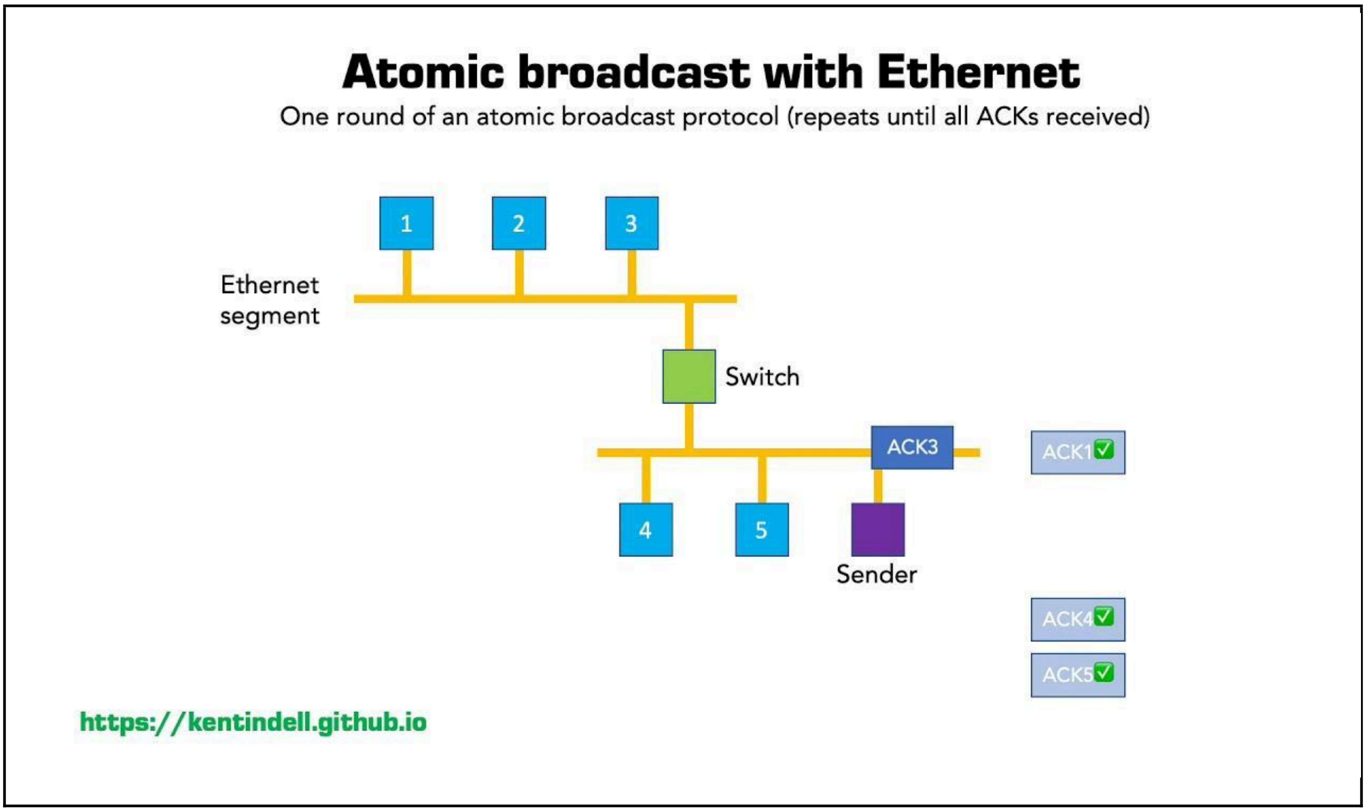
In an atomic broadcast system, if a node delivers a message, then all correct nodes in the system will deliver that message in the same order. This ensures a consensus on the order of messages and helps maintain the consistency of the distributed system.

Key properties of Atomic Broadcast include:

The following properties are usually required from an atomic broadcast protocol:

1. **Validity:** if a correct participant broadcasts a message, then all correct participants will eventually receive it.
2. **Uniform Agreement:** if one correct participant receives a message, then all correct participants will eventually receive that message.
3. **Uniform Integrity:** a message is received by each participant at most once, and only if it was previously broadcast.
4. **Uniform Total Order**: the messages are totally ordered in the mathematical sense; that is, if any correct participant receives message 1 first and message 2 second, then every other correct participant must receive message 1 before message 2.

Atomic Broadcast is a fundamental concept in distributed systems and is often a crucial component in designing fault-tolerant and reliable distributed applications where maintaining a consistent order of events is essential. It finds applications in various distributed systems, including consensus algorithms, distributed databases, and blockchain networks. Examples of consensus algorithms that implement atomic broadcast include Paxos and Raft.



- <u>**Types of Atomic Broadcast**</u>

There are several types of atomic broadcast protocols, each with its own approach to achieving the goal of ensuring that all nodes in a distributed system agree on the order of messages. Here are a few types of Atomic Broadcast protocols:

1. **Primary-Backup Replication:**
   - In this approach, one node is designated as the primary, and others are backups.
   - The primary broadcasts messages to all backups, and backups apply the messages in the order received from the primary.
   - If the primary fails, a new primary is elected from the backups.
2. **Consensus-Based Replication (e.g., Paxos):**
   - Paxos is a consensus algorithm that can be used for atomic broadcast.
   - It ensures that a majority of nodes agree on the order of messages, even if some nodes may fail or behave maliciously.
   - Paxos has different roles, including proposers, acceptors, and learners, and it involves multiple phases to reach consensus.
3. **View-Synchronous Replication:**
   - This approach organizes nodes into views, and each view has a primary node.

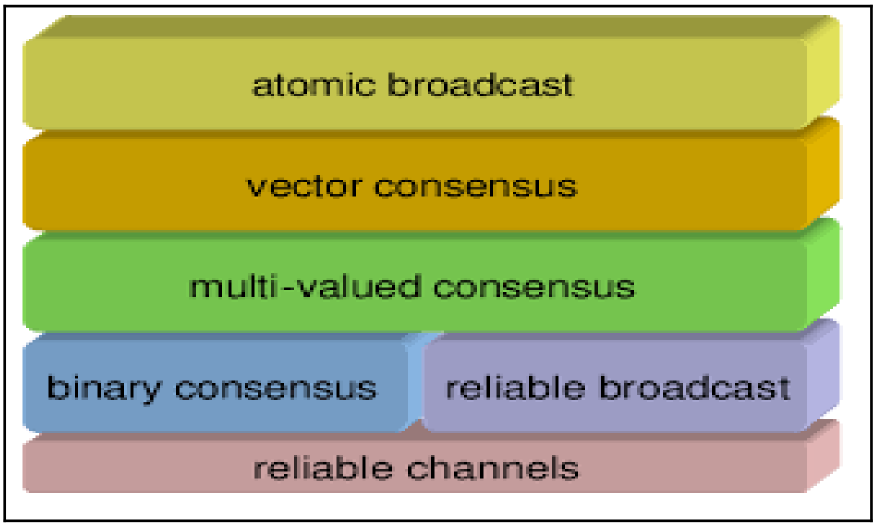
- The primary is responsible for broadcasting messages to all nodes in the view.
- View changes occur when there is a failure or when the primary becomes unreliable.

4. **Totally Ordered Multicast (TOM):**
   - TOM protocols ensure total order delivery of messages in a multicast setting.
   - They define a total order on all messages broadcast to the group.
   - Examples include Virtual Synchrony and Extended Virtual Synchrony protocols.
5. **Raft:**
   - Raft is a consensus algorithm designed for understandability and simplicity.
   - It ensures that all nodes agree on the order of messages through leader election, log replication, and commitment phases.
   - Raft is widely used in distributed systems for its ease of understanding and implementation.

These types of atomic broadcast protocols cater to different requirements and scenarios, such as fault tolerance, simplicity, or specific guarantees regarding the order of messages. The choice of a particular protocol depends on the specific needs and characteristics of the distributed system being designed or implemented.

• Mechanism of transformation from Consensus to Atomic Broadcast

Consensus and Atomic Broadcast are related concepts in distributed systems, but they serve different purposes. Consensus focuses on agreeing on a single value among a group of nodes, while Atomic Broadcast is concerned with agreeing on the order of messages. Achieving Atomic Broadcast typically involves reaching consensus on the delivery order of messages. Here's a basic mechanism of transforming from Consensus to Atomic Broadcast:



1. **Consensus Protocol:**
   - Start with a consensus protocol such as Paxos or Raft.
   - Consensus protocols ensure that a group of nodes reaches an agreement on a single value, even in the presence of failures or network partitions.
2. **Extended Protocol for Atomic Broadcast:**
   - Extend the consensus protocol to include the notion of delivering messages in a specific order.
   - Messages can represent commands or updates that need to be applied in the same order across all nodes.
3. **Log Replication:**
   - Introduce a replicated log that records the agreed-upon sequence of messages.
   - Each node in the system maintains a copy of this log, and the consensus protocol ensures that all nodes have the same entries in the same order.
4. **Message Broadcasting:**
   - Use the consensus protocol to agree on the order in which messages are added to the log.
   - Once the order is agreed upon, the consensus protocol ensures that each node receives the same set of messages in the same order.
5. **Applying Messages:**
   - After reaching consensus on the order of messages, each node applies the messages from the log in the agreed-upon order.
   - This ensures that the state of each node evolves consistently and that the effects of the messages are the same across all nodes.
6. **Ensuring Atomicity:**
   - Atomic Broadcast ensures that either all correct nodes receive a message in the same order or none of them receive it.
   - This property ensures atomicity in the delivery of messages.

By extending a consensus protocol to include the ordering of messages and introducing a replicated log, you create a mechanism that achieves Atomic Broadcast. The consensus protocol guarantees that all nodes agree on the order in which messages are delivered, providing a reliable and consistent approach to broadcasting messages across a distributed system.

Programme Name and Semester: MCA, 4th Semester
Course Name (Course Code): **Introduction to Blockchain Technology & Application**
Course Code: **[MCA403B]**
Academic Session: 2023-2024

Name of the Faculty
Designation and Department
Brainware University, Kolkata

Programme Name and Semester
Course Name (Course Code)
Class
Academic Session