Reg. No.										

MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

(A constituent unit of MAHE, Manipal)

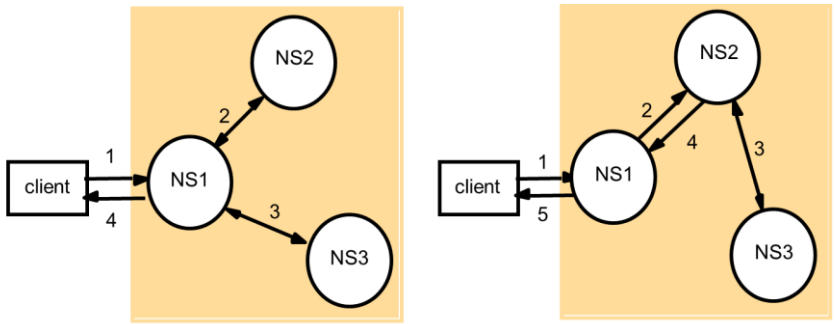
VI SEM. BTECH (INFORMATION TECHNOLOGY)
MID-TERM EXAMINATIONS, MARCH 2024
SUBJECT: DISTRIBUTED SYSTEMS [ICT 3254]
(20/03/2024)
SCHME

Time: 2 Hours MAX. MARKS: 30

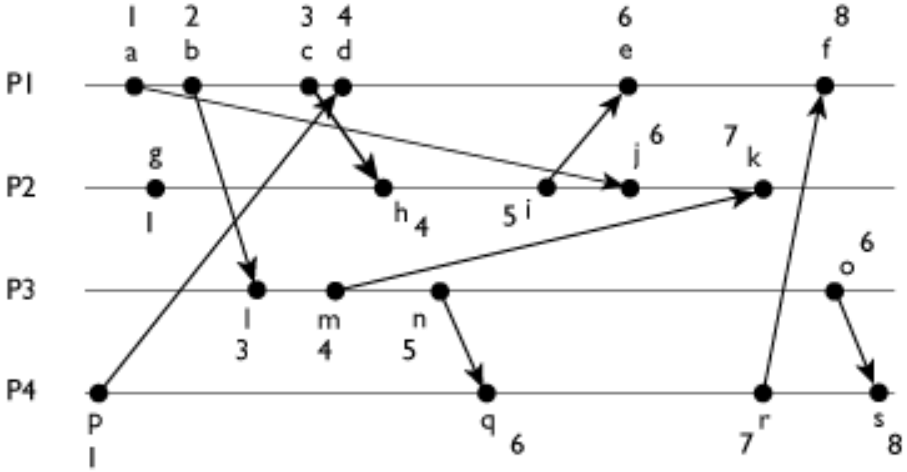
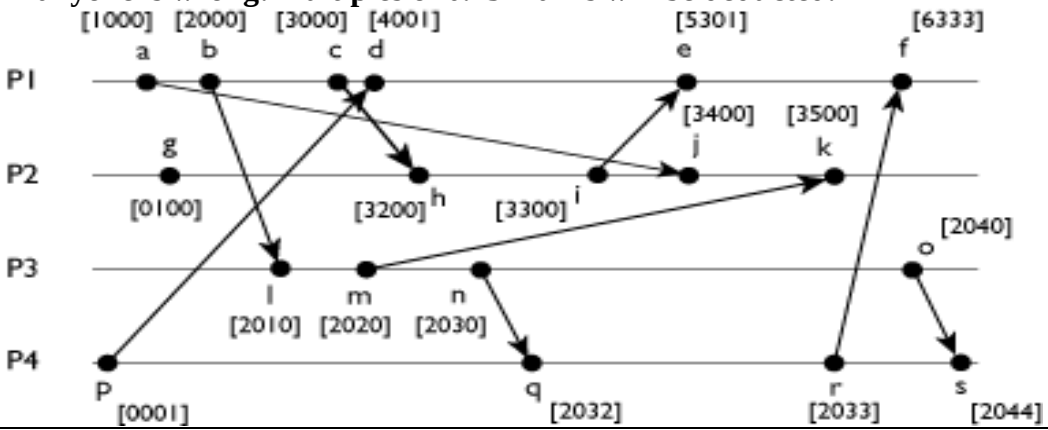
Q#	Question	M	CO	AHEP	BT
1	What is a characteristic feature of thin clients? • High processing power • Extensive local storage capacity • **Heavy reliance on centralized servers • Autonomous operation without network connectivity	0.5	1	1,2	2
2	A team of researchers is conducting fieldwork in a remote area with limited internet connectivity. They must collaborate on collecting data, analyzing findings, and drafting reports in real-time. Which of the following communication tools would be most effective for their situation? • **Cloud-based document collaboration platform • Social media platform for real-time messaging • Email clients with offline capabilities. • Video conferencing software for virtual meetings	0.5	1	1,2	2
3	Feature of a stateless distributed file system (DFS) includes • Persistent storage of file metadata on client machines • Client-server architecture with centralized control • **No dependency on maintaining session or state information • Heavy reliance on caching mechanisms for file access	0.5	3	1,2	2
4	The primary advantage of using NFS (Network File System) in a networked environment includes. • Enhanced security through encryption of transmitted files • Improved network performance by optimizing data transfer protocols. • **Simplified access to shared files across multiple networked devices • Reduced latency through the allocation of dedicated network resources	0.5	3	1,2	2
5	What is the minimum requirement for a class to be remotely accessible in RMI? • **It must implement the Remote interface • It must extend the Remote class • It must be annotated with @Remote • It must be marked as public and serializable	0.5	2	1,2	2
6	Which component is responsible for converting the message format received from the network back into a procedure call on the server? • **Skeleton • Stub • Middleware • Transport Layer	0.5	2	1,2	2

7	Which characteristic of a naming service ensures that names are unique and globally resolvable • **Consistency • Load Balancing • Availability • Scalability	**0.5**	**3**	**1,2**	**2**
8	What is the purpose of a DNS root server? • To store the DNS records for top-level domains (TLDs). • **To maintain the hierarchical structure of the DNS namespace • To resolve queries for local domain names within an organization • To provide authoritative responses for DNS queries	**0.5**	**3**	**1,2**	**2**
9	In the context of distributed systems, what is the significance of Lamport timestamps? • They represent the physical time at which events occur. • ** They provide a total order of events across distributed processes. • They ensure that processes have consistent views of the system state. • They allow processes to detect and resolve conflicts in concurrent operations.	**0.5**	**3**	**1,2**	**2**
10	Which statement accurately describes the behavior of logical clocks in the presence of network delays and message reordering? • **Logical clocks may incorrectly order events if network delays or message reordering occur. • Logical clocks automatically adjust for network delays to maintain global time consistency. • Logical clocks are resilient to message reordering but may require periodic synchronization. • Logical clocks synchronize with physical clocks to compensate for network delays and message reordering.	**0.5**	**3**	**1,2**	**2**
11	Evaluate the different call semantics associated with RPC. Discuss the trade-offs involved in choosing each semantics and provide examples of scenarios where each semantics is suitable. **Ans: the different call semantics: Maybe Semantics, At-least-once Semantics and At-most-once Semantics. (1-Marks)** **a) Maybe Semantics: (1-Marks)** Description: Maybe semantics in RPC imply that the remote procedure call may be executed once or not at all. It arises when no specific fault-tolerance measures are applied. Trade-offs: The main trade-off with maybe semantics is the uncertainty it introduces. Omission failures, where either the request or result message is lost, and crash failures, where the server fails before or after execution, are possible. However, this semantics can be acceptable for applications where occasional missed calls are tolerable. Example Scenario: A scenario where maybe semantics are suitable is in a distributed sensor network where occasional data readings may be missed due to network congestion or node failures. **b) At-least-once Semantics: (1-Marks)** Description: At-least-once semantics ensure that the remote procedure call is executed at least once, and the invoker receives either a result or an exception indicating failure.	**4**	**2**	**1,2,11**	**4**

	Trade-offs: The trade-off here is the potential for duplicated operations. Retransmitting the request message in case of failure can lead to multiple executions of the procedure, which may cause issues if the operation is not idempotent. Example Scenario: At-least-once semantics are suitable for tasks such as updating user profiles in a distributed social media platform, where ensuring that updates are eventually applied is crucial, even if they occur more than once. **c) At-most-once Semantics: (1-Marks)** Description: At-most-once semantics guarantee that the remote procedure call is executed exactly once, ensuring that the invoker receives either a result or an exception indicating failure. Trade-offs: The main trade-off here involves the potential for omission failures if the request or result message is lost. However, employing fault-tolerance measures effectively can mitigate this risk, ensuring that each RPC procedure is executed exactly once. Example Scenario: At-most-once semantics are suitable for financial transactions in a distributed banking system, where it's essential to ensure that each transaction is processed exactly once to maintain data consistency and integrity. In conclusion, the choice of call semantics in RPC systems involves trade-offs between reliability, efficiency, and complexity. Each semantics has its advantages and disadvantages, and the selection depends on the specific requirements and constraints of the application.				
12	You are the system administrator of a medium-sized company that recently migrated its IT infrastructure to a cloud-based environment. As part of the migration, you need to configure the DNS resolution settings to ensure smooth access to internal and external resources for all employees. Using server-controlled navigation, describe how you would configure the DNS resolution process in both recursive and non-recursive modes to meet the company's requirements. **Ans: Recursive: 2 Marks with explanation Non-recursive: 2 Marks with explanation** Configuring Recursive DNS Resolution: • Recursive DNS resolution involves the DNS server performing full resolution on behalf of the client, querying other DNS servers until it obtains the final answer. • Configure the DNS servers to allow recursive queries from internal clients. • Ensure that recursive queries are securely handled to prevent abuse and mitigate DNS-related attacks. • Implement caching mechanisms to improve performance for frequently accessed domains. Configuring Non-Recursive DNS Resolution: • Non-recursive DNS resolution involves the DNS server providing only the information it has in its local cache or authoritative zone files without performing further queries on behalf of the client. • Set up non-recursive DNS resolution for external queries to improve security and prevent unnecessary traffic to external DNS servers. • Define the specific domains or zones for which non-recursive queries are allowed, such as internal domains or trusted external domains.	**4**	**3**	**1,2,11**	**3**

	• Implement DNS filtering or security policies to ensure that non-recursive queries are restricted appropriately and do not expose the internal network to potential threats.  Non-recursive server-controlled Recursive server-controlled A name server NS1 communicates with other name servers on behalf of a client • Non-recursive/Recursive • Recursive type is suitable to environment where there are administrative domain prohibits • Under non-recursive server-controlled navigation, any name server may be chosen by the client. • This server communicates by multicast or iteratively with its peers in as it were a client. • Under recursive server-controlled navigation, the client once more contacts a single server. • If this server does not store the name, the server contacts a peer storing a (larger) prefix of the name, which in turn attempts to resolve it. • This procedure continues recursively until the name is resolved.				
13	A group of students is working on a collaborative project and must securely share files and documents. They all have laptops with Wi-Fi capabilities. How can the students be provided with a secure platform to share files and collaborate effectively without relying on central servers or specific user accounts? Discuss the technical challenges involved in implementing such a solution. Ans: **Explanation about Peer to Peer network →2M** **Any two technical challenges →1M** **Peer-to-Peer Network Setup:** Each student's laptop can act as a node in the P2P network, allowing them to directly connect and share files with each other. Encryption and Authentication: Implement end-to-end encryption to ensure that files are transmitted securely between peers. Each student's device can generate a public-private key pair, and authentication can be performed using these keys to verify the identity of the sender and receiver. File Indexing and Discovery: Develop a mechanism for indexing and discovering files shared by peers in the network. This could involve creating a distributed hash table (DHT) or implementing a decentralized search algorithm to locate files based on keywords or metadata. Data Replication and Redundancy: Ensure data replication and redundancy to mitigate the risk of data loss. Replicate files across multiple peers in the network to maintain availability even if some peers go offline or encounter failures.	**3**	**1**	**1,2,11**	**4**

	Conflict Resolution: Implement conflict resolution mechanisms to handle concurrent edits or updates to shared files. This could involve version control systems or consensus algorithms to ensure that changes made by different students are appropriately merged and synchronized. Firewall and Security Measures: Address security concerns such as firewall configurations and network security protocols to prevent unauthorized access and protect against external threats. **Challenges involved in implementing such a solution include:** Network Reliability: Ensuring reliable connectivity and data transfer over potentially unstable Wi-Fi networks, especially in environments with high network congestion or interference. Scalability: Designing the system to handle a growing number of peers and files while maintaining performance and efficiency. Data Consistency: Ensuring consistency and coherence of shared data across multiple peers, especially in scenarios where peers have different versions of the same file. Security Risks: Mitigating security risks such as data breaches, unauthorized access, and malware propagation within the P2P network. User Experience: Designing an intuitive user interface and experience that allows students to easily navigate and interact with the P2P file sharing system. By addressing these technical challenges and implementing appropriate solutions, the students can securely share files and collaborate effectively without relying on central servers or specific user accounts.				
14	A multinational corporation with globally dispersed offices requires a distributed file system to streamline collaboration and sharing among its diverse teams. Highlight three critical features the distributed file system must possess to support the corporation's operations effectively. Furthermore, it elucidates how each feature addresses a specific challenge commonly encountered in distributed environments. **Ans: 3 Features: 3M** Feature 1: Data Replication Explanation: Data replication is a critical feature that ensures copies of data are stored across multiple geographic locations in the distributed file system. This redundancy addresses the challenge of data availability and accessibility, particularly in distributed environments where network connectivity may be unreliable or where local hardware failures can occur. By replicating data across multiple sites, the distributed file system ensures that users have access to the required files and resources even if one location experiences downtime or network issues. Additionally, data replication enhances fault tolerance by providing backup copies of data, thereby reducing the risk of data loss due to hardware failures or disasters. Feature 2: Encryption and Security Explanation: Encryption and security mechanisms are essential features to safeguard data integrity and confidentiality in a distributed file system used by a multinational corporation. Encryption ensures that data transmitted over the network and stored on distributed nodes remains secure and protected from unauthorized access or interception. This addresses the challenge of data security and compliance with regulatory requirements, particularly when dealing with sensitive corporate information across diverse geographic locations. Additionally,	**3**	**3**	**1,2,11**	**3**

	robust authentication and access control mechanisms ensure that only authorized users can access and modify files, further enhancing security and mitigating the risk of data breaches or unauthorized access. Feature 3: Global Namespace and Unified Access Explanation: A global namespace and unified access provide a seamless user experience by presenting a unified view of the distributed file system across all geographic locations. This feature addresses the challenge of complexity and inconsistency in managing distributed data by providing users with a single, centralized interface for accessing and managing files regardless of their physical location. By abstracting the underlying complexities of distributed storage, a global namespace ensures that users can easily locate and access the files they need, irrespective of where they are stored geographically. This enhances collaboration and productivity among diverse teams within the multinational corporation by eliminating barriers to accessing and sharing files across different offices and locations.				
15	Consider processes P1, P2, P3, and P4 executing in a distributed system with Lamport timestamps. Write vector timestamps for all the events given in the figure below.  Ans: Vector Timestamps **All correct vector timestamps: 3 marks** **If anyone is wrong: multiples of 0.25 marks will be deducted.**  	**3**	**3**	**1,2,11**	**3**
16	An NTP server B receives server A's message at 16:34:23.480, bearing a timestamp of 16:34:13.430, and replies to it. A receives the message at 16:34:15.725, bearing B's timestamp, 16:34:25.7. Estimate the offset between B and A and the estimate's accuracy. Discuss the factors to consider when deciding which NTP server a client should synchronize its clock.	**3**	**3**	**1,2,11**	**3**

	Ans: Calculation of Offset: 1.5 marks Calculation of Accuracy: 1 marks Factors for synchronization: 0.5 marks To estimate the offset between server B and server A, we can calculate the difference between the timestamps provided by each server. Given: Server B's timestamp when receiving the message from server A: 16:34:13.430 Server A's timestamp when receiving the message from server B: 16:34:25.7 Offset between B and A = Server A's timestamp - Server B's timestamp = 16:34:25.7 - 16:34:13.430 = 12.270 seconds To estimate the accuracy of the offset, we can use the difference in timestamps when server B receives the message from server A and when server A receives the message from server B. Given: Server B's timestamp when receiving the message from server A: 16:34:23.480 Server A's timestamp when receiving the message from server B: 16:34:15.725 Difference in timestamps = Server A's timestamp - Server B's timestamp = 16:34:15.725 - 16:34:23.480 = -7.755 seconds The negative sign indicates that server A's timestamp is earlier than server B's timestamp, which suggests that the clocks of the two servers may not be perfectly synchronized. Therefore, the estimated offset between server B and server A is 12.270 seconds, and the accuracy of this estimate is approximately 7.755 seconds.				
17	How might the financial trading firm optimize the distribution of market data updates to multiple traders simultaneously, and what are the advantages of utilizing this mechanism for efficient broadcasting? To optimize the distribution of market data updates to multiple traders simultaneously, the financial trading firm could implement IP multicast. IP multicast allows the firm to broadcast market data updates efficiently to all interested traders within a multicast group. This mechanism offers several advantages**: (1-Marks)** Bandwidth Efficiency: With IP multicast, the firm can send a single copy of market data updates to the multicast group address, rather than individual copies to each trader. This reduces network bandwidth usage significantly, especially in scenarios with a large number of traders. Membership of group is dynamic: IP multicast scales well with the number of receivers. Regardless of the number of traders subscribed to the multicast group, the firm can efficiently distribute market data updates without experiencing significant performance degradation (scalability). **(2- Marks)**	**3**	**1**	**1,2,11**	**4**
18	Discuss how strict consistency ensures memory coherence and synchronization across multiple processors in a parallel computing environment. Ans: Strict consistency is a memory consistency model that ensures memory coherence and synchronization across multiple processors in a parallel computing environment. It guarantees that all processors in the system see the same consistent view of memory at any given time and that memory operations from different processors appear to occur in a globally agreed-upon order **1Mark**	**2**	**3**	**1,2,11**	**3**

	This ensures that updates made by one processor are immediately visible to all other processors, preventing inconsistencies and ensuring that the behavior of the parallel program is predictable. **1Mark**				