B.Tech. TCS-801/TIT-801

EIGHTH SEMESTER EXAMINATION, 2009-10

DISTRIBUTED SYSTEMS

Time: 3 *Hours* *Total Marks:* 100

Note: *(1) Attempt all questions.*

1. Answer any four parts of the following: (4 × 5 = 10)

Q.1. (a) How the resources sharing done in distributed system? Explain with an example.

Ans. With distributed system, it is easier for users to access remote resources and to share resources with other users. Example, printers, files, web pages etc.

- A distributed system should also make it easier for users to exchange information.
- Easier resource and data exchange could cause security problems a distributed system should deal with this problem.
- Resources in a distributed system are physically encapsulated within computers and can only be accessed from other computers by communication interface enabling the resources to be accessed and updated reliably and consistently. For example, users are concerned with sharing data in the form of a shared database or a set of web pages, not the disks and processors that those are implemented on. Similarly user think in terms of shared resources such as search engine or a currency converter, without regard for the server or server that provide there.

Another example, a search engine on the web provides a facility to users through out the world.

Q.1. (b) Discuss the limitation of distributed system.

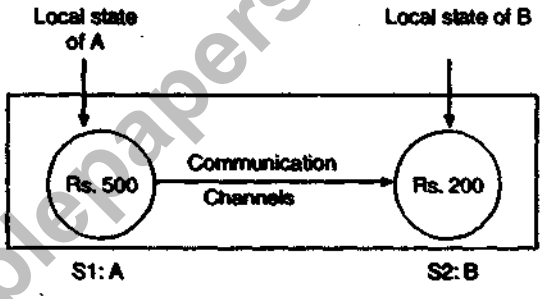
Ans. Global knowledge is not readly available, since it is impossible to collect upto date information of global state of distributed system.

Because of (limitation):

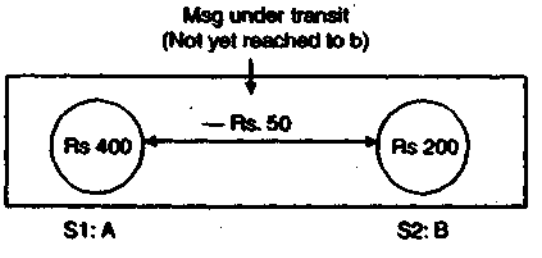
- Lack of global clock (common clock)
- Unprecedented message delay
- Non-existence of physically shared memory

Above limitation create difficulty in obtaining coherent global state.

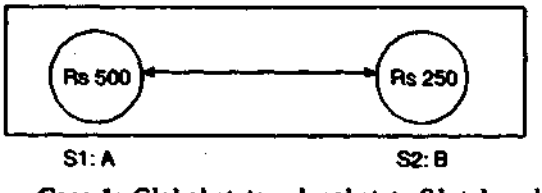
For example,



Case 1



Case 2



Case 1: Global state = local state S1 + Local state S2

= 450 + 200

= 650 = Rs 50 missing *i.e.*, in-coherent system

Case 2: Global state = Local A state + channel state + Local B state

Q.1. (c) What do you mean by Global state of the

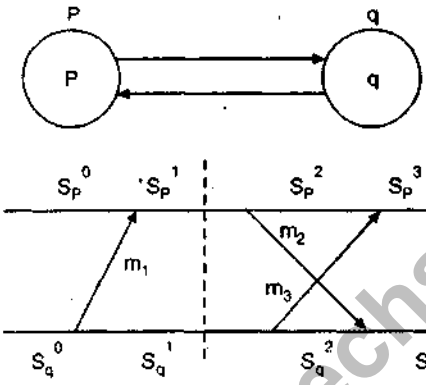
distributed system? Also explain the main features of consistent Global state.

Ans. Global State: "The global state of a distributed system is the set of local states of all individual processes involved in the computation plus the state of the communication channels."

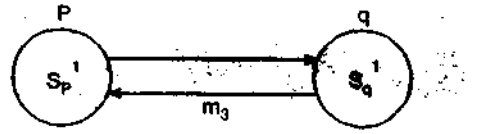
Consistent Global state: A global state of a distributed system is consistent if no transactions are in progress.

- A global state is consistent if it could have been observed by an external observer.
- If $e \rightarrow e'$ then it is never the care that e' is observed by the external observer and not .
- All fearible states are consistent.

An Example,



A Consistent State



Q.1. (*d*) Differentiate between Token based algorithm and non token based algorithm.

Ans. Token based algorithm: This is the first class of mutual exclusion (mE) algorithm. For distributed system.

In the algorithm, a site of sites possibly holding the token as opposed to all the sites.

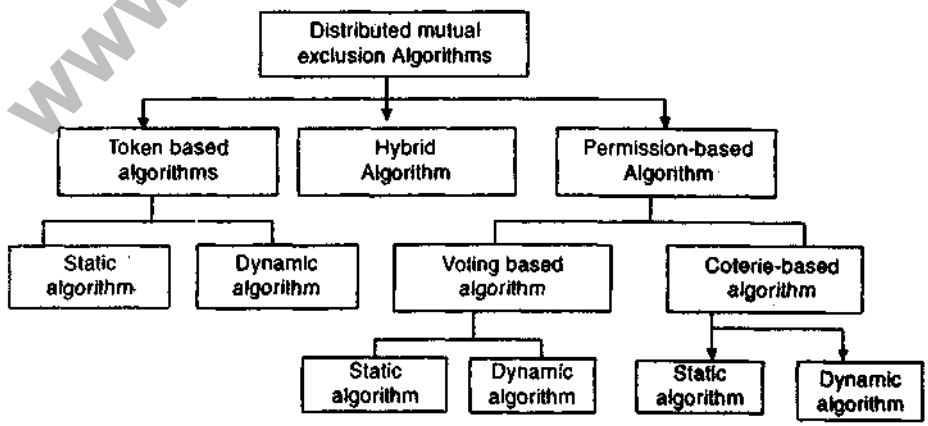
- A request set is used to record the identifiers of such a set of sites in the system.
- As the six of the request set is reduced, the number of the messages exchanged per critical section (cs) is reduced.

Non-token based Algorithm

- This is the second class of mutual exclusion (mE) algorithm for distributed system.
- Unlike token based algorithm, it uses Token "Privileged" message to arrived at mutual exclusion.
- The priveleged message is used like request set is used in the Token based algorithm.
- This algorithm is not as efficient as the token based algorithm.

Q.1. (*e*) Explain the classification of distributed mutual exclusion.

Ans. A number of DME algorithms have been proposed.



These are two classes of Mutual Exclusion:

- **Non token based** (uses message to arrive at ME)
- **Token based** (uses token "Privileged" message to arrive at ME)

Q.1. (f) Discuss the web challenges for implementing distributed system.

Ans. These are following:

1. Heterogeneity
2. Openness
3. Security
4. Stability
5. Failure handling
6. Concurrency
7. Transparency

1. Heterogeneity: We set up protocols to solve these heterogeneities.

- **Middle ware:** A software layer that provides a programming abstraction as well as marking the heterogeneity.
- **Mobile code:** Code that can be sent from one to another computer and run at destination.

2. Openness: Open DS can be constructed from heterogenous hardware and software.

3. Security: Security for information resources has three components: confidentiality, integrity, availability.

4. Scalability: A system is described as scalable if it remains effective when there is a significant increases in the number of resources and the number of users.

Failure Handling:

Techniques for dealing with failures

- Detecting failures
- Masking failures
- Tolerating failures
- Recovering from failures
- Redundancy

6. Concurrency: "There is a possibility that several clients will attempt to access a shared resources at the same time."

7. Transparency: Eight forms of transparency:

- Access transparency
- Location transparency
- Concurrency transparency
- Replication transparency
- Failure transparency
- Mobility transparency
- Performance transparency
- Scaling transparency

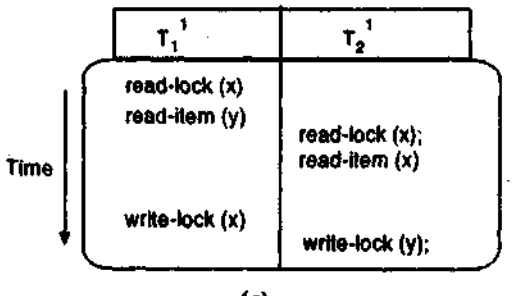
"Transparency is defined as the concealment from the user and application progammer of the separation of components in a distributed system, so that the system is provided as a whole rather than as a collection of independent components".

2. Answer any two parts of the following: **(2 × 10 = 20)**

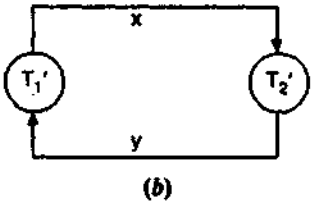
Q.2. (a) Define deadlocks. Differentiate between resource and communication Deadlocks. Discuss various deadlock handling strategies in detail.

Ans. Deadlocks occurs when each transaction T in a set of two or more transaction is waiting for some item that is locked by some other transaction T in the set. Hence, each transaction in the set is on a waiting queue, waiting for one of the other transaction in the set to release the lock on an item.

Example: A simple examples where the two transaction T_1' and T_2' are deadlocked in a partial schedule T_1' is on the waiting queue for X, which is locked by T_2'. While T_2' is on the waiting queue for y which is located by T_1'. Mean while, neither T_1' and T_2' nor other transaction can access items X and Y.



(a)



(b)

(*a*) A partial schedule of T_1' and T_2' that is in a state of deadlock.

(*b*) A wait for graph for the partial schedule *m*(*a*).

Resource and Communication Deadlock

- **Deadlock due to resource, here set of process are under resource deadlock:** It every process is waiting for resources held by process in a same set and it must receive all resource when process become unblock.
- **While deadlock due to communication:** Here set of process are said to be in communication deadlock if every process is waiting for message from other process, in the same set process initiate communication further only when it receives all the messages for which it is waiting for.

Deadlock handling strategies

1. Prevention
2. Avoidance
3. Detection

1. Prevention: Process begins its execution only when required resources is available and if the resource are not preempted before execution begins.

2. Avoidance: Resources are allocated only if resultant global system state is safe. Each site maintain its local state that require storage in form of memory which is an overhead.

3. Detection: It can be devided into 3 parts:

(*a*) Centralized D.D algorithm
Advantage: RAG and WAG is easy

(*b*) Distributed D.D algorithm
Advantage: Not susceptible to single point failure.

(*c*) **Hierachial D.D algorithm:** This is suitable for both centralized and distributed D.D.

Q.2. (*b*) Write short notes on following:

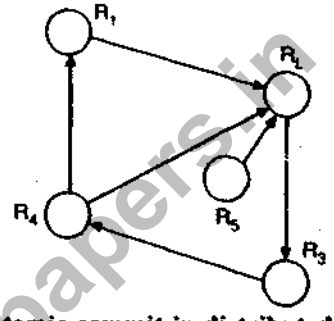
(*i*) Wait for graph

(*ii*) Atomic commit in distributed database systems.

Ans. (*i*) Wait for graph: A simple way to detect a state of 'deadlock' is for the system to construct and maintain a 'wait for graph'.

- Processes are represented as nodes and an edge from process P_i to P_j implies P_j is holding a resource that P_i needs and thus P_i is waiting for P_j to release its lock on the resource.
- A 'deadlock' occurs if the graph contains any cycles.
- A wait for graph scheme is applicable to a resource allocation system with "multiple instances" of each resource type.

Example:



(*ii*) Atomic commit in distributed database systems: Atomic commit in database system fulfil two of they key properties of ACID, (Automaticity and Consistency). Consistency is only achieved if each change in the atomic commits is consistent.

- The two-phase commit protocol (2PC) is a type of an 'atomic commitment' protocol.
- The two phase commit protocol requires a co-ordinator to maintain all the information needed to recover the original state of the database if something goes wrong. As the name indicates. These are two phase voting and commit.

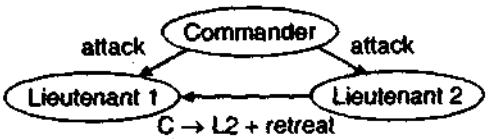
(*a*) During the 'Voting' phase each node writes the changes in the atomic commit to its own disk. The nodes then report their status to the co-ordinator.

(*b*) During the 'commit' phase the co-ordinator sends a commit message to each of the nodes to records in their individual logs. If any of the nodes reported, failure the co-ordinator will instead send a rollback message.

Q.2. (*c*) Explain Lamport - Shostak - Pease algorithm (Oral Message Algorithm) for 3 m + 1 or more processors where m is the number of faulty processors.

Ans. • Valid for oral messages

• No solution for processors < 3 M + 1



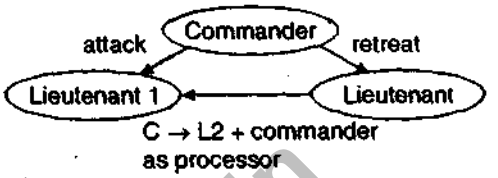
Assumptions: A1: Every message is delivered correctly.

A2: Receiver knows the sender

A3: Failure can be detected

Majority rule: 1. Choose the majority value, if exist else 'Retreat'.

2. If from an ordered set, choose the 'Median'.



Oral Messages:

Algo OM (O)

- Commander send his value to every lieutenant.
- Each lieutenant (L) use the value received from commander, or RETREAT if no value is received.

Algo OM (M), M > O

- Commander send his value to every lieutenant (v_i)
- Each lieutenant acts as commander for OM ($m - 1$) and sends v_i to the other n-2 lieutenant (or RETREAT)
- For each i, and each $j <> i$, let v_j be the value lieutenant i receives from lieutenant j in step (2) using OM ($m - 1$).
- Lieutenant i uses the value majority (V_1 ..., V_{n-1})
- Why $j <> i$? "True myself more than what others said I said."

Assures:

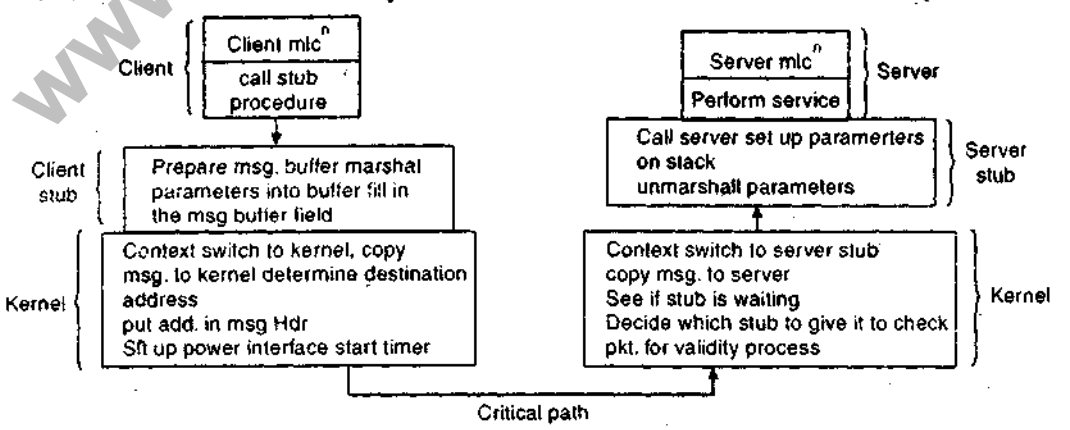
- Processors cannot interfere with communication as 3rd party.
- Can't send-take messages.
- Can't interfere by being silent.

3. Answer any two parts of the following: **(2 × 10 = 20)**

Q.3. (a) (i) What is the communication models proposed for the communication between the distributed objects?

Ans. Two models have been widely used to develop distributed applications for distributed system:

1. **Message passing method:** The method depends on the two basic concepts for message transfer and these are SEND and RECEIVE process. The two are the main concept for msg transfer among various sites of the distributed system and known as SEND and the RECEIVE primitives.



2. **Remote procedure call:** When a process on machine A calls a procedure located machine B, the calling process on A is suspended and execution at cared procedure takes place on B. Information can be transported from caller to callee in the parameters and can come back. It is known as RPC.

(*ii*) Explain following with an example:

(A) Remote object reference

(B) Remote interface

Ans. (A) Remote object reference: Other object can invoke the methods of a remote object if they have access to its remote object reference. Example, A remote object reference for B must be available to A as shown in figure.

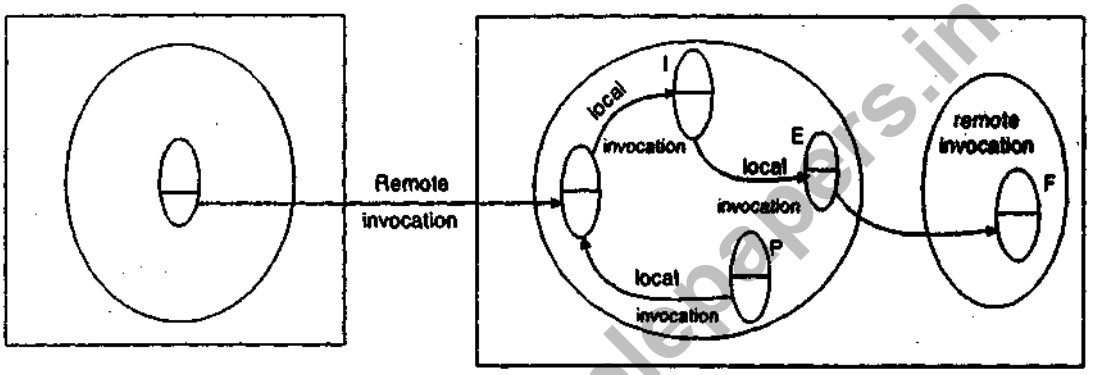


Fig. Remote and local method invocations

(B) Remote interface: Every remote object has a remote interface that specifies which of its methods can be involved remotely.

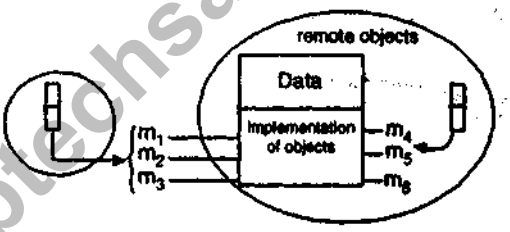


Fig. A remote objects and its remote interface

Q.3. (*b*) What are the public and private keys? List the key differences and issue in public keys cryptography and private key cartography?

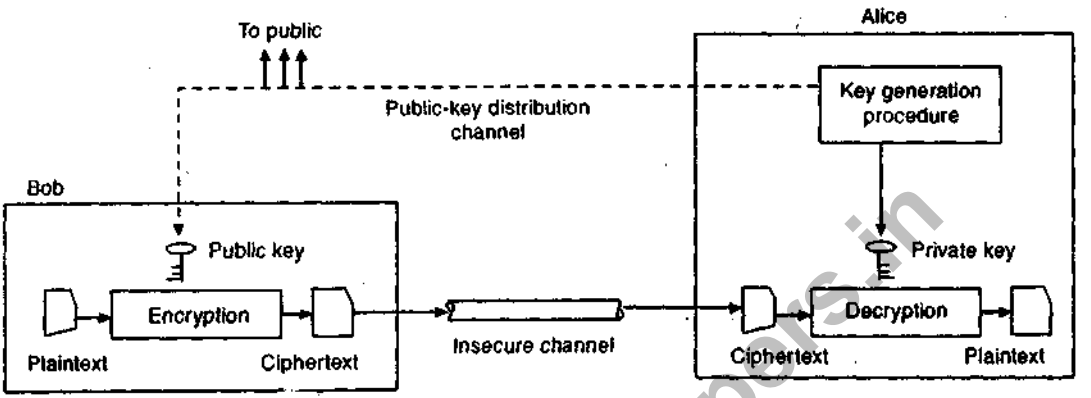
Ans. Public and Private Keys: In cryptographic system, there is a use of two key pairs – Public key and private key.

- A public key is known to everyone.
- A private key or key is known only to the recipient of the message.

Public Key Cryptography

- Public key encryption uses a pair of mathematically related keys. A message that is encrypted with the first key must be decrypted with the second key, and a message that is encrypted with the second key must be decrypted with the first key.
- Each participant in a public-key system has a pair of keys. The symmetric (private) key is kept secret. The other key is distributed to anyone who wants it, this key is public key.

- To send an encrypted massage to you, the sender encrypts the message by using your public key. When you receive the message you decrypt it by using your symmetric key to send a message to someone, you encrypt the message by using the recipients public key. The message can be decrypted with the recipients symmetric key only.

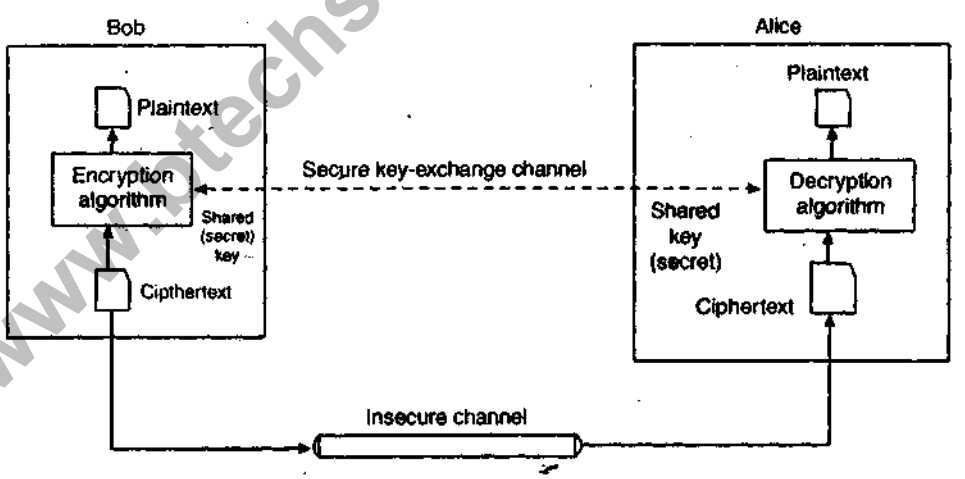


Private key encryption:

- Private key encryption systems use a single key that is shared between the sender and the receiver. Both must have the key, the sender encrypts the message by using the key, and the receiver decrypts the message with the same key. Both must keep the key private to keep their communication private.

Characteristics:

- Private key encryption requires a key for every pair of individuals who need to communicate privately. The necessary number of keys rises dramatically as the number of participant increases.
- The fact that keys must be shared between pairs of communicators means that the keys must somehow be distributed to the participants.



Q.3. (*c*) Write short notes on following:

(*i*) Architecture of distributed Event Notification

Ans. (*i*) Distributed event based system have two main characteristics:

Hetrogeneaus: When event notification are used as a means of communication between distributed objects.

Asynchronous: Notification are sent Asynchronausly by event generating objects to all the objects that have subscribed to them to prevent publishers needing to synchronize with subscribers.

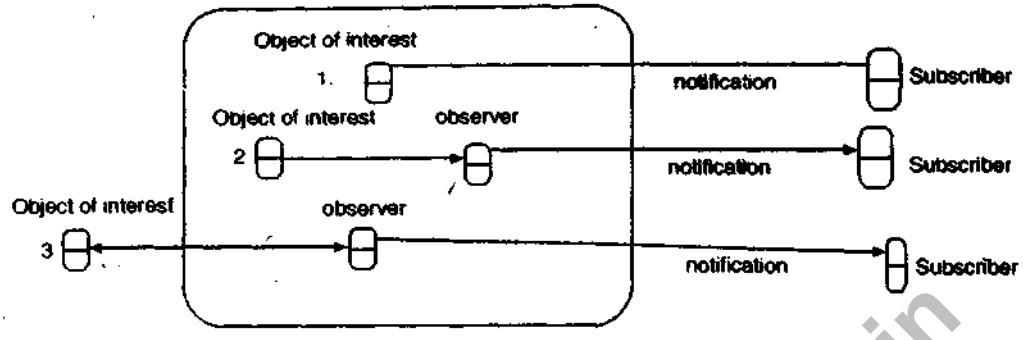


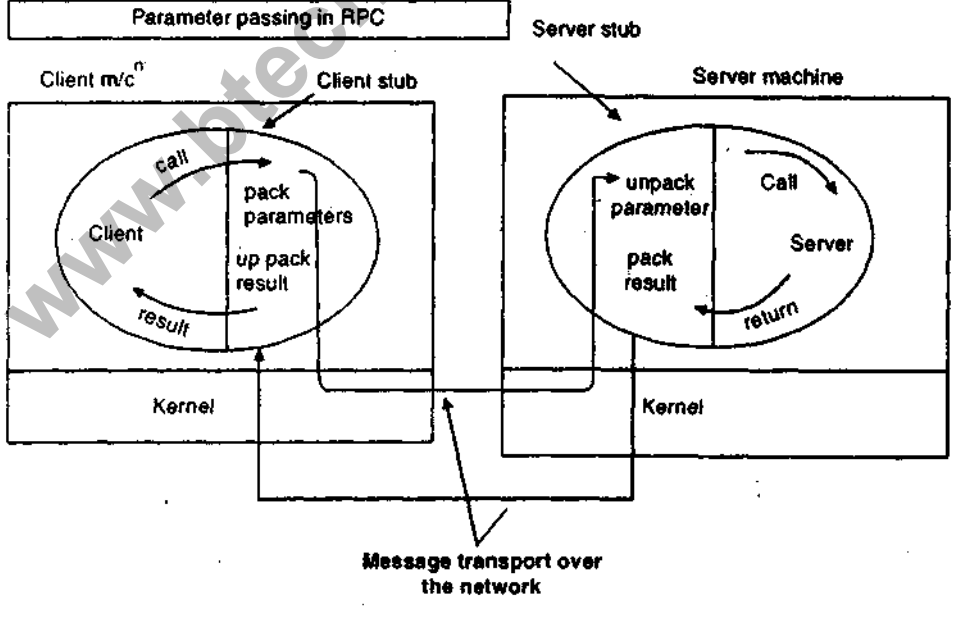
Fig. Architecture for distributed event notification

Different participating objects are –

1. The object of interest
2. Event
3. Notification
4. Subscriber
5. Observer objects
6. Publisher

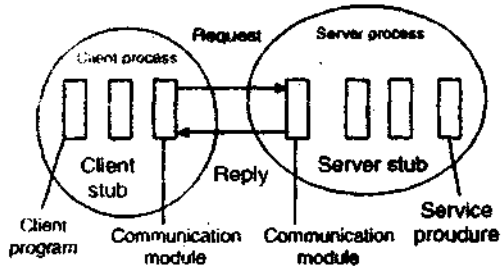
(ii) Remote procedure call

Ans. When a process on machine A calls a procedure located m/c^{n} B, the calling process on A suspended and execution of the called procedure takes place on B. Information can be transported from the caller to the callee in the parameters and can come back in the procedure result.



Each ellipse represents a single process, with the shaded portion being the stub.

Role of client and server stub in RPC



4. Answer any two parts of the following: **(2 × 10 = 20)**

Q.4. (*a*) Compare and contrast the methods of concurrency control for transactions. Explain the methods for concurrency control in distributed transactions.

Ans.
- Time stamp method is similar to two-phase locking in that both use pessimistic approaches in which conflicts between transactions are detected as each object is accessed.
- Time stamp ordering decide serialization order statically when transaction sort on the other hand two-phase locking decide the serialization order dynamically.
- Time stamp ordering is better then strict two phase locking for read only transaction.
- Two-phase locking is better when the operations in transactions are predominantly updated.
- Timestamp ordering aborts the transaction immediately, whereas locking makes the transaction wait.
- Time stamp ordering is deadlock free.
- When optimistic concurrency control is used all transactions when they allowed to commit or in forward validation transaction are aborted earlier.

This results in relatively efficient operation where there are few conflicts, but a substantial amount of work may have to be repeated when a transaction is aborted.

Methods for concurrency control is

1. **Locking:** In distributed transaction the locks on an object are held locally, when locking is used for concurrency control the objects remain locked and are unavailable for another transaction during the atomic commit protocol. Consider following interleaving of transaction T and U at servers X and Y.

T	U
Write (A) at X locks A	
	Write (b) at Y locks B
Read (B) at Y wait for U	
	Read (A) at X wait for T

2. **Time stamp ordering:** In distributed transactions, we require that each co-ordinator issue globally unique time stamp. A globally unique time stamp is issued to the client by the first coordinator accessed by a transaction.
3. **Optimistic concurrency control:** A distributed transaction is validated by a collection of independent servers, each of which validates transactions that access its own objects.

Q.4. (*b*) What do you mean by two phase Locking? How it is different from strict two phase Locking? Explain.

Ans. A transaction is said to follow the two-phase locking if all locking operation (Read-lock, write-lock) proceed the first unlock operation in the transaction. Such a transaction can be divided into two phases:

(*a*) Expanding or growing (first phase), during which new locks on items can be acquired but none can be released.

(*b*) Shrinking (second) Phase: During which existing locks can be released but no new locks can be acquired.

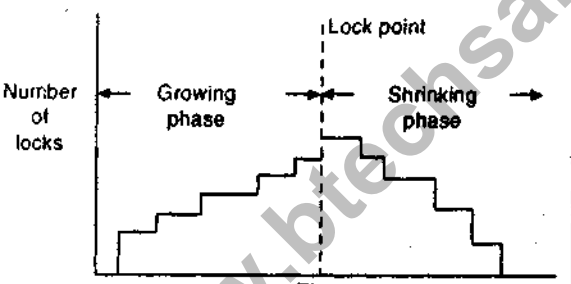
If lock conversion is allowed, then upgrading of locks must be done during expanding phase and downgrading of locks must be done in the shrinking phase. For example: Transaction T_1' and T_2' which are the same but follow the 2-phase locking they can produce a deadlock.

T_1'	T_2'
read_lock (Y);	read_lock (X);
read_item (Y);	read_item (X);
write_lock (X);	write_lock (Y);
unlock (Y);	unlock (X);
read-item (X);	read_item (Y);
$X_1 = X + Y$;	$Y_1 = X + Y$;
Write_item(X);	write_item (Y);
Unlock (X);	unlock (Y);

Strict two phase locking: is the most popular version of two phase locking which guarantees strict schedules. In this variation, a transaction T does not release any of its exclusive (write) lock and until after it commits or aborts.

Hence, no others transaction can read or write an item that is written by T unless T has committed, leading to a strict schedule for recoverability.

Strict two phase locking: is not deadlock-free. A more restrictive version is rigorous two phase locking



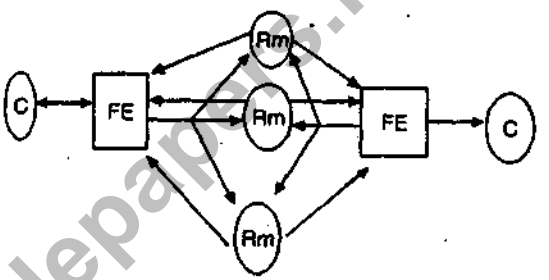
Two phase locking

Q.4. (c) Explain the following:

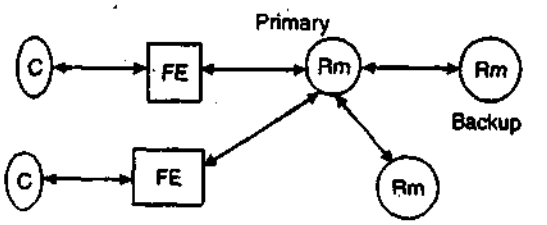
(i) Fault tolerant services.

Ans. (a) Fault tolerant using active replication services: In an active model of replication of fault tolerance, the replica managers are state machines that play equivalent roles and are organized as a group. Front ends multicart their request to the group of replica managers and all the replica managers process the request independently but identically and reply request on operation to be performed as follows:

1. **Request:** The front end attaches a unique identifier to the request.
2. **Co-ordination:** The every correct replica manager in the same order.
3. **Execution:** Every replica manager executes the request.
4. **Agreement:** No agreement phase is needed.
5. **Response:** Sends response to the front end.

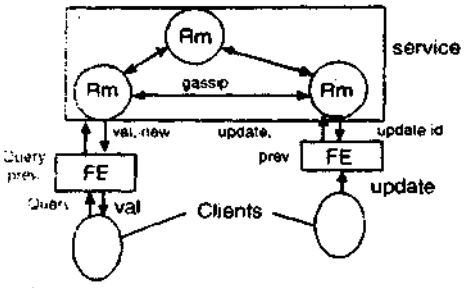


(b) Fault tolerance using passive (primary-backup) replication: In this model of replication for fault tolerance, there is at any one time a single primary manager and one or more secondary replica managers-backup or slaves.



(ii) Highly available services.

Ans. Our emphasis now is on giving clients access to the service-with reasonable response times-for as much of the time as possible, even if some results do not conform to sequential consistency. For example, user on the train at the beginning may be willing to core with temporary inconsistencies between copies of data such as diaries they can continue to work while disconnected and fix any problem later. Example, Gassip architecture for implementing highly available services.



5. Answer any two parts of the following:

(2 × 10 = 20)

Q.5. (*a*) Explain the term "routing". How routing problem can be classified? Also discuss the criterion for good routing algorithms.

Ans. A node in a computer network in general not connected directly to every other process by charmel. A node can send packets of information directly only to a subset of the nodes called the neighbours of the node.

Routing "is the term used to describe the decision procedure by which a node selects one (or sometimes more) of its neighbours to forward a packet on its way to an ultimate destination."

Classification: Two types of routing algorithms:

(*a*) Non-adaptive Routing Algorithms.

(*b*) Adaptive Routing Algorithms.

(*c*) Hierarchial Routing is used to make there Algorithms scale to large networks.

1. Non Adaptive:

Example: 1. Floding routing

2. Shortest path routing (Dijskrtra's shortest path Algorithm)

Criteria for good Routing Algorithm

1. **Correctness:** The Algorithm must deliver every packet offered to the network to its ultimate destination.
2. **Efficiency:** The algo. must bend packets through "good" paths, an algo. is called optimal if it uses the "best" paths.
3. **Complexity:** The algo. for the computation of the tables must use as few messages, time and storage as possible.
4. **Robustness:** In case of a topological change the Algorithm updates the routing tables in order to perform the routing function in the modified network.
5. **Adaptiveness:** The algorithm balances the load of channels and nodes by adapting the tables in order to avoid paths through channel.
6. **Fairness:** The algo. must provide service to every user in the same degree.

Q.5. (*b*) (*i*) What are traversal algorithms? Discuss the properties of this algorithm.

Ans. Traversal algorithm is a special case of "wave algorithm", in which all events of a wave are totally ordered by the causality relation and in which the last event occurs in the same process as the first event.

A traversal algorithm is an algorithm with the following three problems or properties:

1. In each computation there is one initiator, which starts the algorithm by sending out exactly one message.
2. A process, upon receipt of a message, either sends out the one message or decides.
3. The algorithm terminates in the initiator and when this happens, each process has sent a message at least once.

(*ii*) Explain Tarry's algorithm for traversing connected networks.

Ans. Tarry's Traversal Algorithm: For an arbitrary network.

- R_1: A process does not send the token twice over the same channel (in the same direction)
- R_2: A process sends the token to its father only if there is no other channel meeting the first condition.

Algo: Var used$_p$ [q]: boolean init false For each $q\varepsilon$ Neigh p ;

Father p: Process init udef

initiator, execute once:

begin father p: = choose $q\varepsilon$ Neigh p;

used p [q]: = true: send <to k> to q

end

all process, upon receipt of <to k> from q_o;

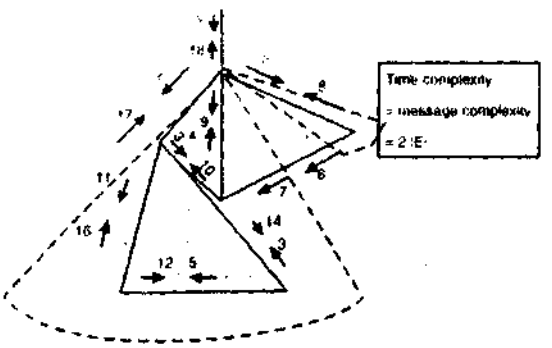
begin if father p = udef then father p: = q_o;

if $\forall\, q\varepsilon$ Neigh p: used p [q] then decide

else if $\exists\, q\varepsilon$ Neigh p: (q ≠ father p ^ ¬ used p[q])

then begin choose $q\varepsilon$ Neighp:/ {father} with ¬ used p [q] ;

usedp [q]: = true: send <to k> to q
end
else begin usedp [fatherp]: = true: send <to k> to fatherp end
end



Q.5. (c) Write short notes on following:

(i) CORBA services

(ii) Deadlock free packet switching

Ans. (i) CORBA service:

1. The naming service: The COBRA naming service is a sophisticated example of the binder it allows names to be bound to the remote object reference of COBRA objects with in naming contexts.

2. COBRA event service: The COBRA event service specification defines interfaces allowing objects of interest. Called suppliers to communicate as arguments or results of ordinary synchronous COBRA remote method invocation.

3. COBRA security service: It includes the following–

- Authentication of principles (users and servers)
- Access control to be applied on the COBRA
- Facilities of non-repudiation.

4. Trading service: In contrast to the naming service which allows COBRA objects to be located by name the trading service allows them to be located by name attribute, it is a directory service.

5. Transaction services and concurrent consis services: It allows distributed COBRA objects to participate in either flat or nested transaction.

6. Persistent object-services: The architecture of the POS allows for a set of data stores to be available-each persistent object has a persistent identifies.

(ii) Deadlock free packet switching

Ans. (ii) Message (packets) travelling through a packet switched communication network must be stored at each node before being forwarded to the next node on the path to their destination.

- Each node of the network reserves some buffer space for this purpose.
- As the amount of buffer space is finite in each node, situation may occur where no packet can be forwarded because all buffers in the next node are occupied.

There are two kinds of methods of avoiding store and forward deadlocks:

1. **Structured buffer pool:** Will identify for a node and a packet a specific buffer that must be taken if a packet is generated or received.
2. **Unstructured buffer pool:** Does not determine in which buffer it must be placed.

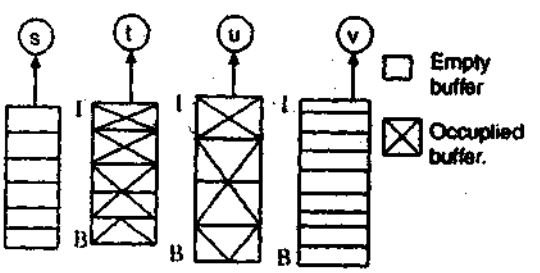


Fig. An example of store and forward deadlock