DEPENDABILITY IN MOBILE COMPUTING SYSTEMS

Sarmistha Neogy

Department of Computer Science and Engineering

Jadavpur University

Kolkata, India

TUTORIAL, DEPEND 2010, JULY 18, 2010, VENICE

MY CO-RESEARCHERS

- *Chandreyee Chowdhury* – Jadavpur University, India
- *Suparna Biswas – West Bengal* University of Technology, India
- M. Tech students

TUTORIAL OUTLINE

- Definition and Attributes
- Mobile Computing Systems and its challenges
- Availability
  - Fault tolerance using Checkpointing and recovery
  - A few schemes

OUTLINE …

- Reliability
  - Estimation with fault tolerance
- Security
  - Approaches to secure checkpoints and authenticate mobile hosts
- Concluding remarks
- References

OBJECTIVE

- Mobile Computing Systems (MCS) is a much studied and analysed research area now a days.
- Focus of such research is basically on the aspects concerning location management, channel assignments and activities related to mobile telephony services, even integration of existing services with the vast computing that this system offers is often overlooked.

OBJECTIVE …

- Similarly, security of such a system is also considered but not in conjunction with others.
- To develop an integrated system all aspects of a system should be considered together.
- This view of the design of a system is the motivation of our research.
- MCSs are supposed to be logical extension of distributed systems but techniques meant for distributed systems fail to make impact in MCSs.

OBJECTIVE …

- Hence requirement for developing new design techniques for all aspects of MCSs grew.
- We found that MCS would not be complete without considering the challenges that it poses.
- This leads us to the factors of availability and reliability and overall security in a system.
- Thus the concept of developing a dependable mobile computing system …..

DEFINITION AND ATTRIBUTES

Definition

Dependability of any computing system may be defined as -

- the trustworthiness of a computing system which allows reliance to be justifiably placed on the service it delivers

Dependability includes the following attributes of a computing system:

Attributes

- *Availability*: readiness for correct service
- *Reliability*: continuity of correct service
- *Safety*: absence of catastrophic consequences on the user(s) and the environment
- *Security*: concurrent existence of availability for authorized users only

AVAILABILITY

- may be defined as a ratio of the expected value of the uptime of a system to the aggregate of the expected values of up and down time.

  Availability of a repairable system

- "the probability that the system is operating at a specified time t" [Barlow and Proschan, 1975]

AVAILABILITY contd.

- "a measure of the degree of a system which is in the operable and committable state at the start of mission when the mission is called for at an unknown random point in time" [Blanchard 1998]

RELIABILITY

- May be defined as the probability of failure-free software operation for a specified period of time in a specified environment
- usually defined in terms of a statistical measure

RELEVANT TERMS

- **Fault**: a defect in the software, e.g. a bug in the code which may cause a failure in the software/system
- **Failure**: a derivation of the observed behavior of a program/software from its expected/desired behavior

Reliability contd..

Evaluation of Reliability

- Reliability Estimation - applies statistical inference techniques to failure data
- Reliability Prediction - determines future software reliability

APPROACHES FOR ACHIEVING RELIABLE SOFTWARE SYSTEMS INCLUDE:

- Fault Prevention
- Fault Removal
- Fault Tolerance
- Fault/Failure Forecasting

SAFETY

- is the state of being safe,
  - the condition of being protected against
    - physical, social, spiritual, financial, political, emotional, occupational, psychological or other types or
    - consequences of failure, damage, error, accidents, harm or any other event

  which could be considered dangerous.

SAFETY …

- Safety is generally interpreted as implying a real and significant impact on risk of death, injury or damage to property.

SECURITY

- Basic components:
  - Confidentiality
  - Integrity
    - Data
    - Origin
  - Availability

SECURITY contd.

- Confidentiality
  - Concealment of information or resources
  - Also applies to the existence of data - sometimes more revealing than data itself

SECURITY contd.

- Integrity
  - Trustworthiness of data or resource
  - Generally means - preventing improper or unauthorized change
- Integrity mechanisms
  - Prevention of unauthorized access
  - Detection of threat

AVAILABILITY

- Ability to use desired information or resource
- An aspect of system design – closely related to other aspects: reliability, and security

MOBILE COMPUTING SYSTEMS AND ITS CHALLENGES

MOBILE COMPUTING SYSTEM (MCS)

- A new paradigm of computing: Wireless networking with mobility
- Users carrying portable devices have access to shared infrastructure
- Access is independent of physical location of user
- Portable devices are capable of wireless networking

Mobile computing systems contd…

- A Mobile Computing system may be described as consisting mobile hosts (MHs) that interact with the fixed network via mobile support stations (MSSs).
- Connection between an MSS and an MH is via wireless link.
- Each MSS can be thought of as the 'in-charge' of a *cell*.

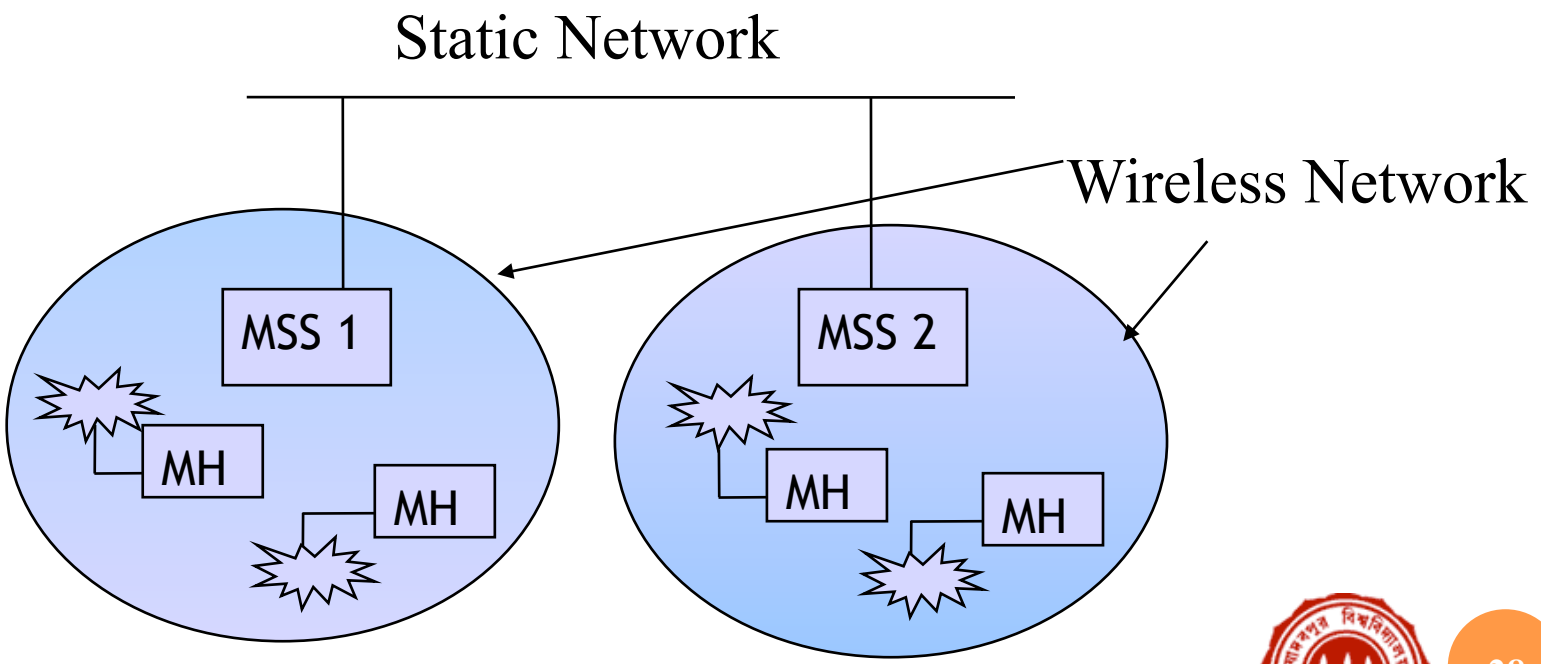
Mobile computing systems contd…

- Overall architecture of a wireless system is based on the concept of distributed system divided into physical cells.

- IN A NUTSHELL:

- MOBILE COMPUTING IS A FORM OF WIRELESS NETWORKING THAT PERMITS NETWORKED DEVICES TO BE MOVED FREELY YET REMAINING CONNECTED TO THE NETWORK AND PERFORMING COMPUTING.

MCS: a look



CHALLENGES IN DESIGNING MOBILE COMPUTING SYSTEMS

- Disconnection
- Low bandwidth
- Variable bandwidth
- Heterogeneous network
- Fading, noise, interference

Challenges contd….

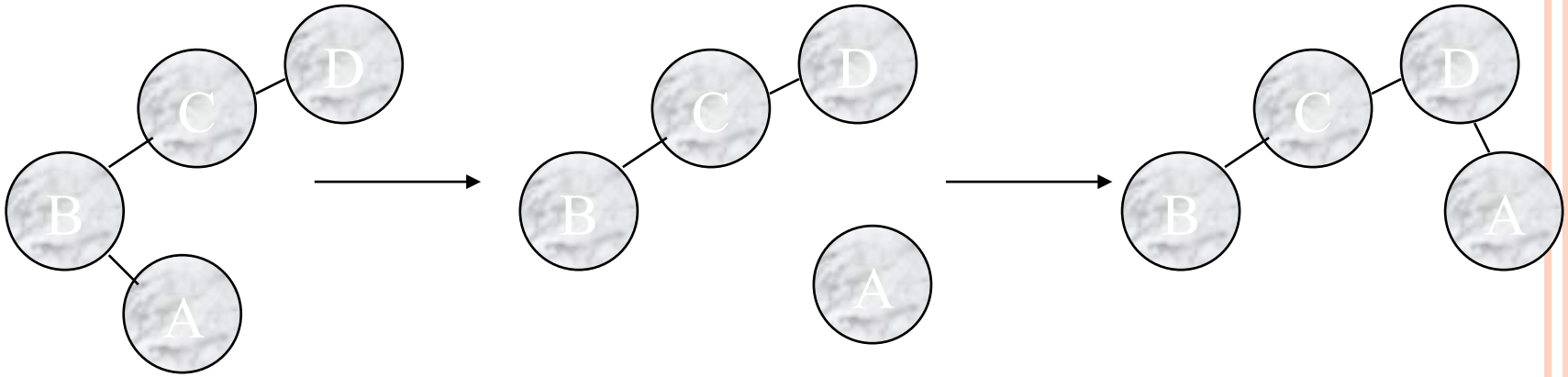
- Address migration
- Migrating locality
- Location dependence
- Management of transmission power
- Energy efficiency
- Security risks

DEPENDABILITY AND MOBILE COMPUTING SYSTEMS

- Designing MCS requires solutions to the challenges
- Designing dependable MCS requires not only solutions to the challenges mentioned but also integration of the different aspects of dependability
- This makes design of dependable MCS a far more challenging job

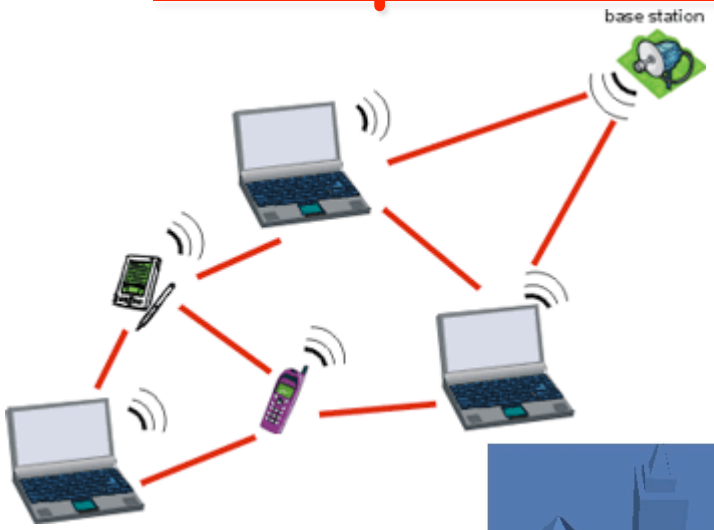
INTRODUCTION-MANET

- No pre-existing communication infrastructure
- Autonomous system of mobile routers (and associated hosts) connected by wireless links

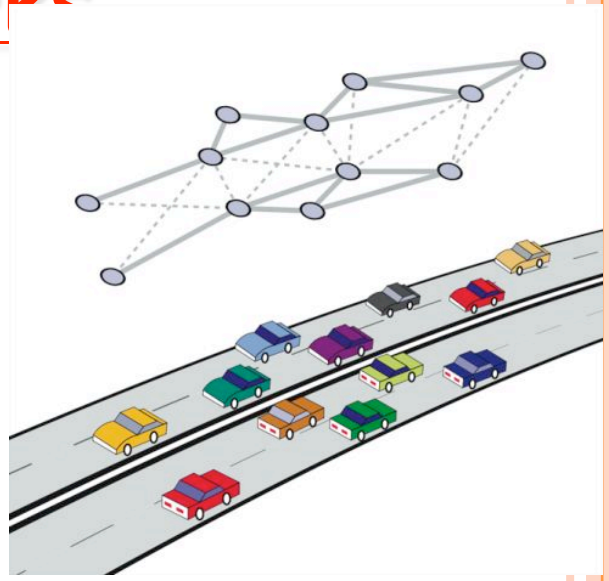


- Routers are free to move randomly and organize themselves arbitrarily
- Wireless topology may change rapidly and unpredictably

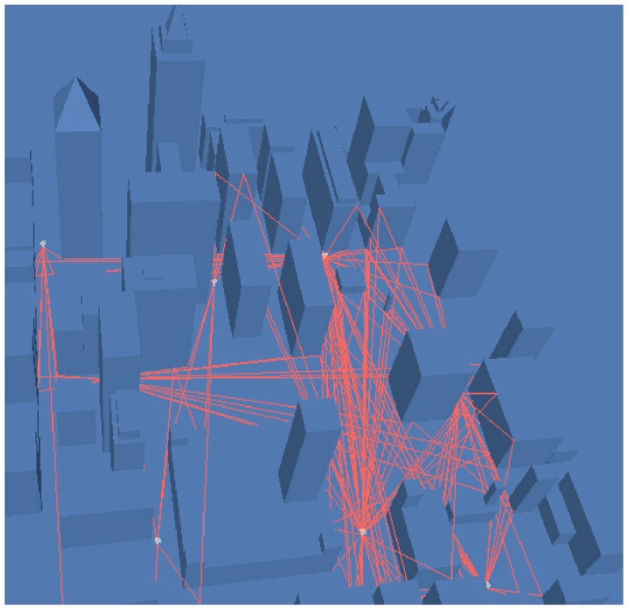
Example Ad hoc Networks



Mobile devices (laptop, PDAs)

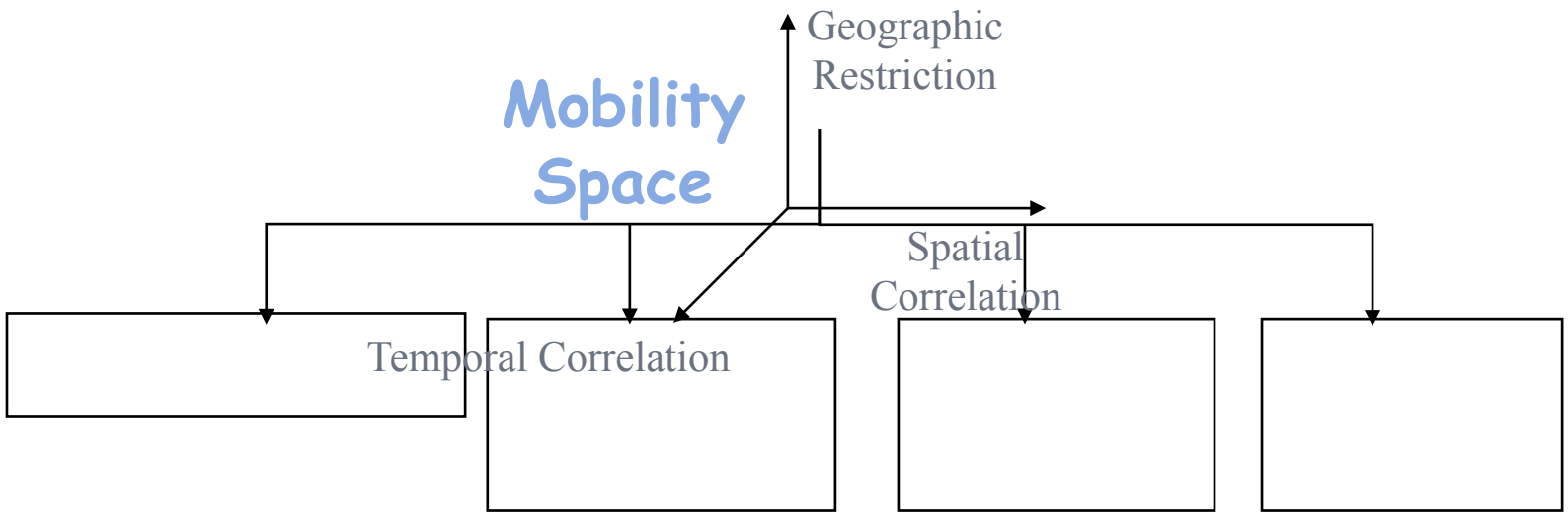


Vehicular Networks on Highways

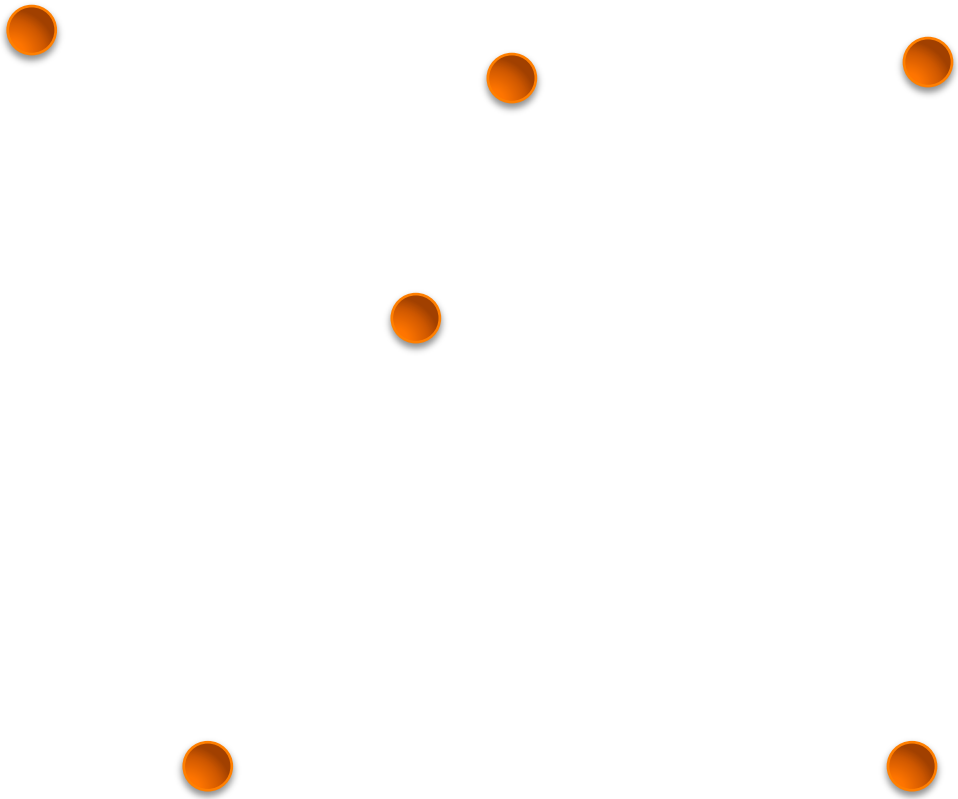


Hybrid urban ad hoc network (vehicular, pedestrian, hot spots,…)

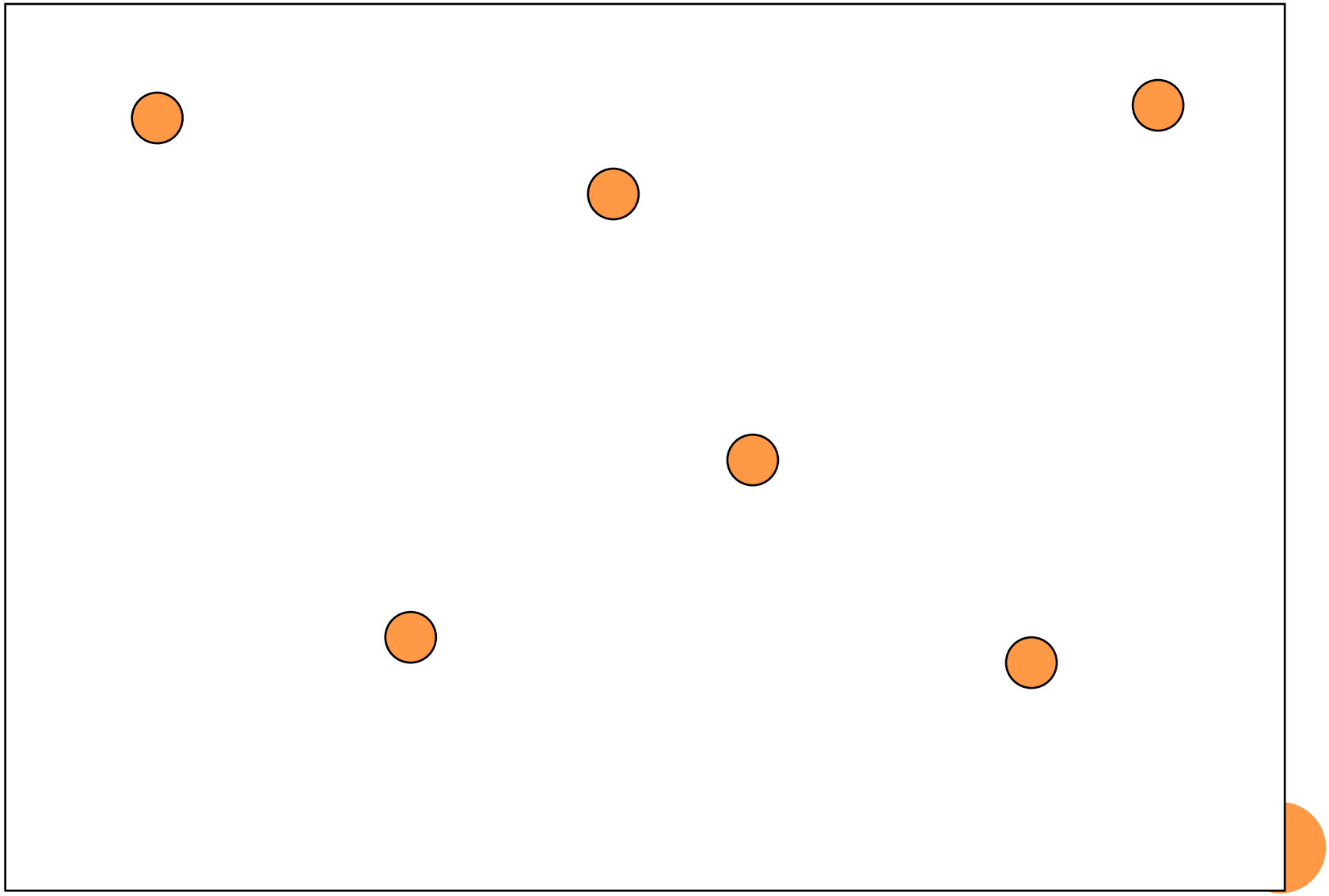
CLASSIFICATION OF MOBILITY MODELS



RANDOM WAY POINT



RANDOM WAY POINT (EXAMPLE)



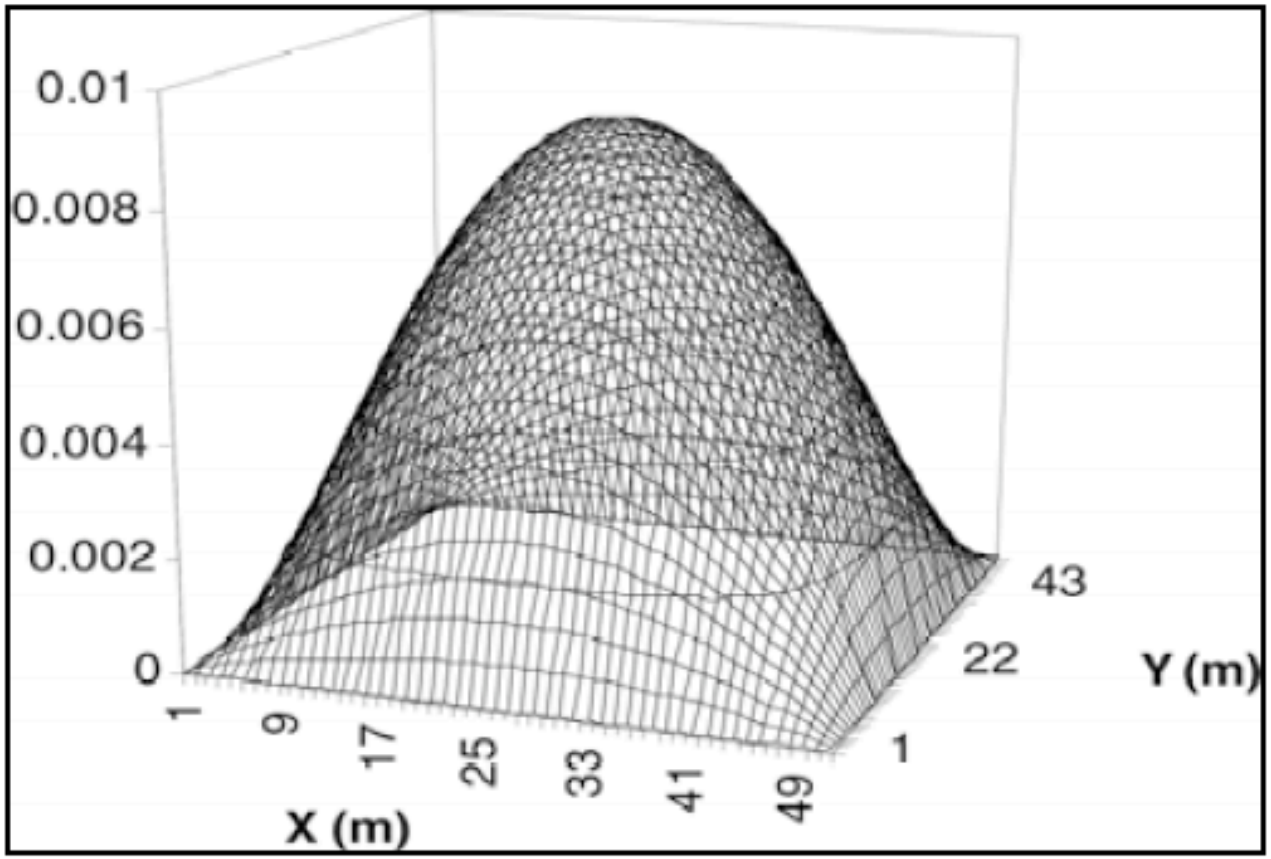
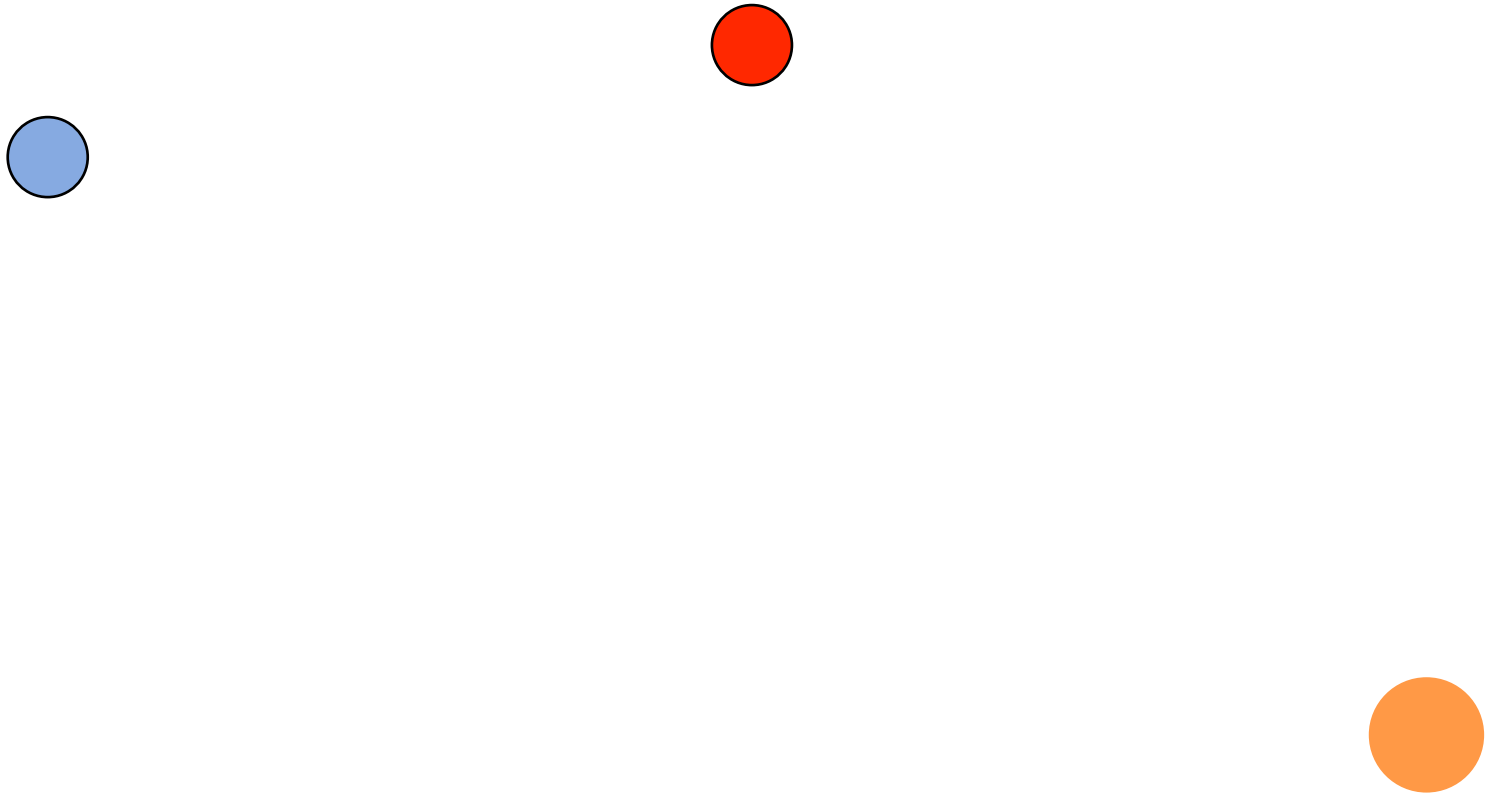


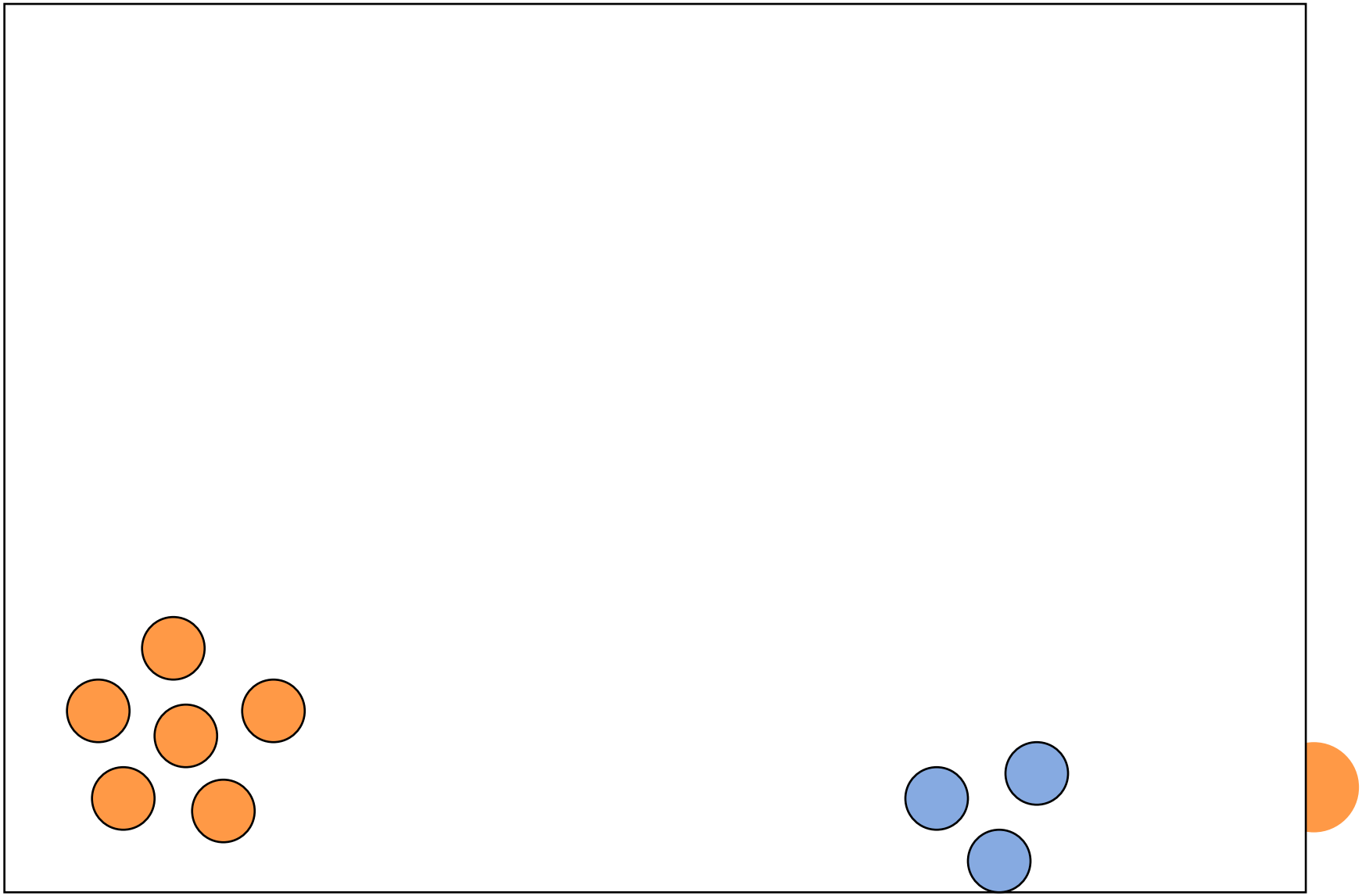
Figure 1-3. Node Spatial Distribution (Square Area)

RANDOM WALK

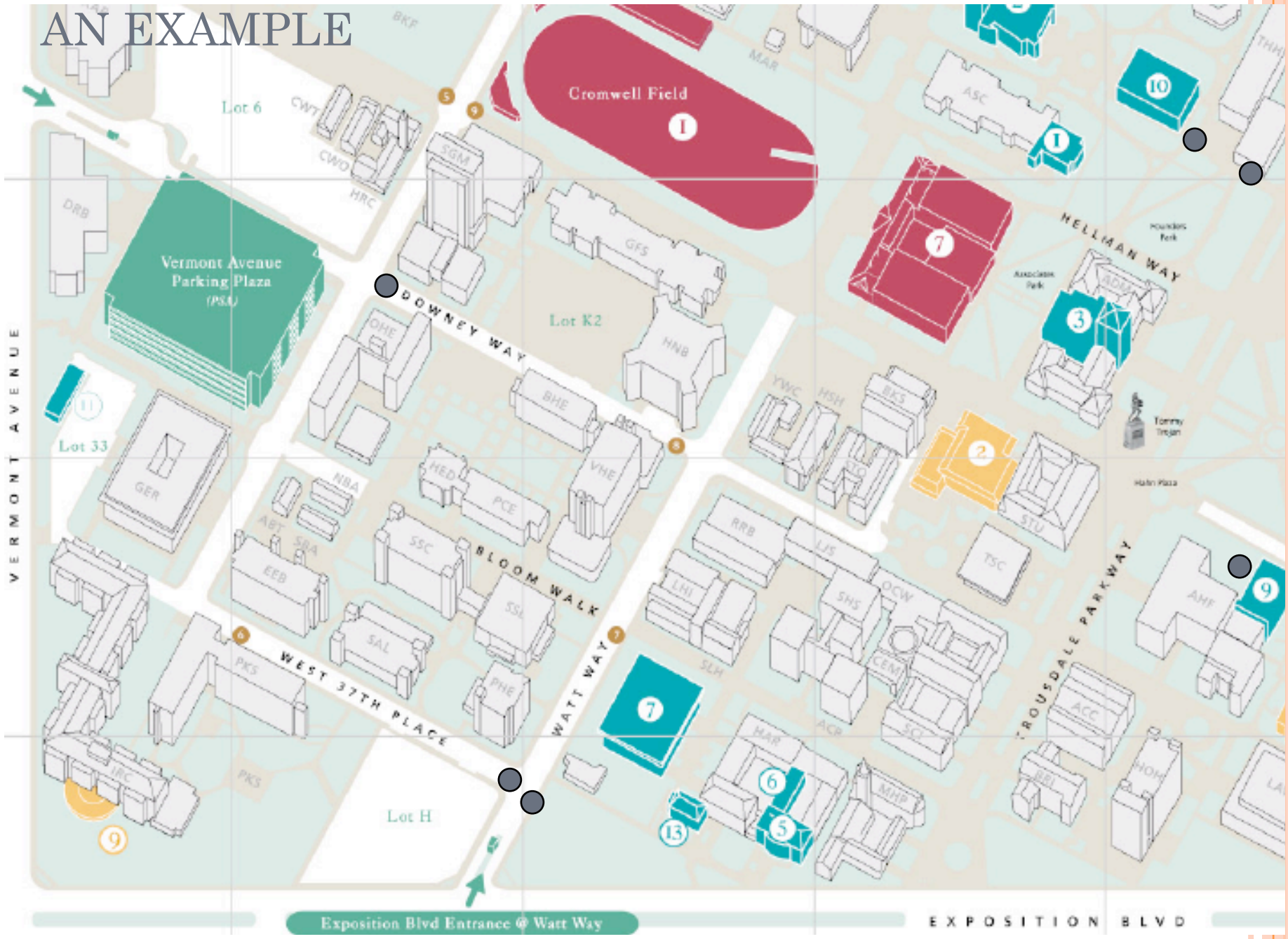


GROUP MOBILITY (SINGLE GROUP)

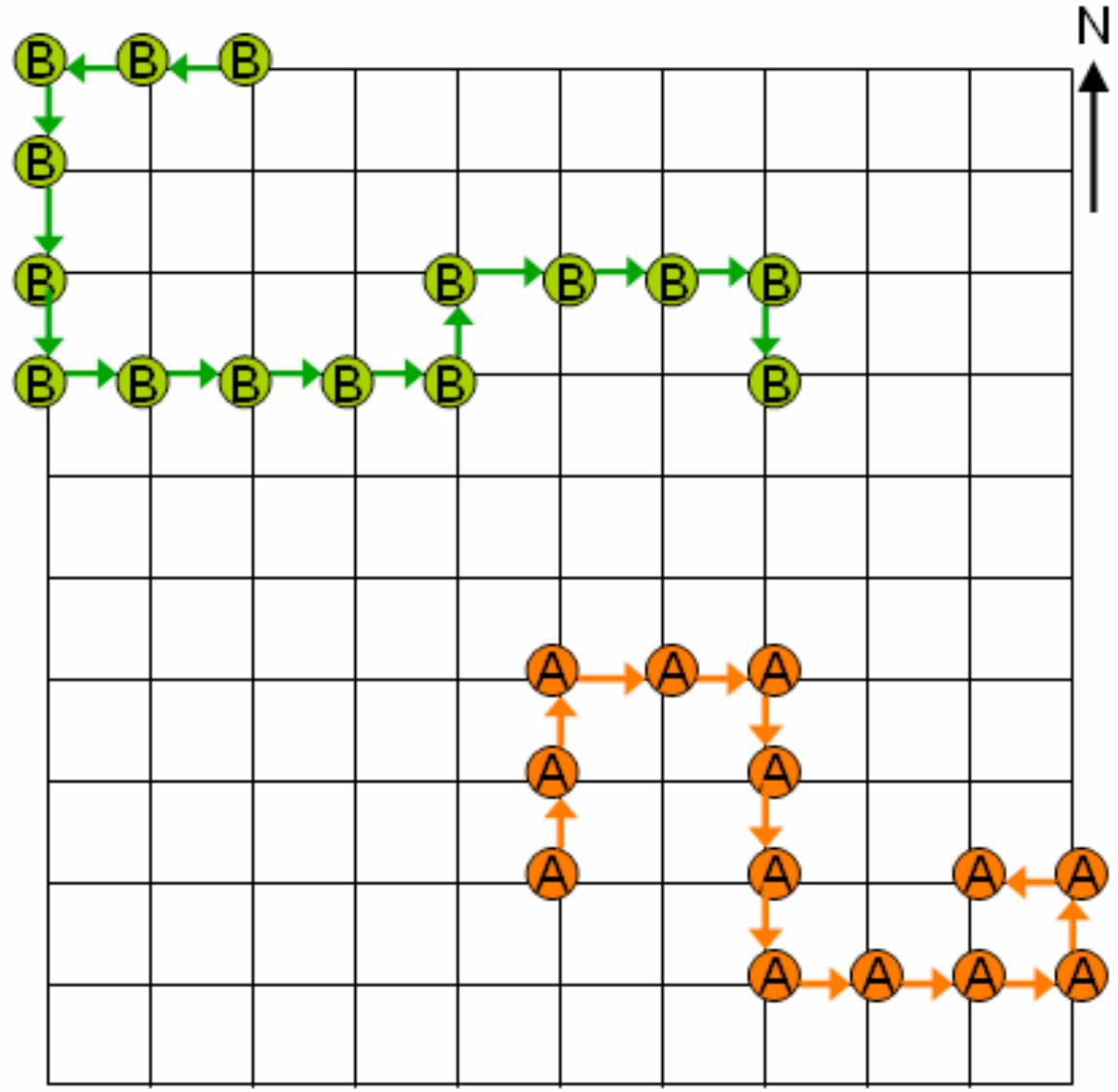
GROUP MOBILITY (MULTIPLE GROUPS)



AN EXAMPLE


Cromwell Field
Lot 6
Vermont Avenue Parking Plaza
(PSA)
Lot K2
Lot 33
Lot H
DOWNEY WAY
BLOOM WALK
WEST 37TH PLACE
WATT WAY
HELLMAN WAY
TROUSDALE PARKWAY
VERMONT AVENUE
EXPOSITION BLVD
Exposition Blvd Entrance @ Watt Way

MANHATTAN MOBILITY MODEL



COMPARISON

Table 1-1. The characteristics of mobility models used in IMPORTANT framework

	Temporal Dependency	Spatial Dependency	Geographic Restriction
Random Waypoint Model	No	No	No
Reference Point Group Model	No	Yes	No
Freeway Mobility Model	Yes	Yes	Yes
Manhattan Mobility Model	Yes	No	Yes

BUT WHY ?

- Some points to ponder:
  - MHs in MCSs are prone to frequent disconnections due to interference etc., mobility, lesser power and so on
  - Network traffic variation with time
  - Security threats like presence of bluetooth, for example

AND HOW ?

- Have to choose appropriate methods to incorporate *reliability*
- Reliability, in turn, ensures *availability*
- System is reliable if it is able to
  - *Detect and repair* or *tolerate* fault
  - Operate in a *secured and safe* state

WHAT TO DO?

- Characteristics of failure in MCS:
  - Transient
  - Independent
- Detection needs to be done
- Detection and repair not possible
- Requirement is to absorb the fault since it is temporary and let the system continue its execution

FAULT TOLERANCE

- Fault-tolerance is the property that enables a system to continue operating properly in the event of the failure of some of its components
- Fault-tolerance is required since systems are susceptible to failure and therefore ability to tolerate failures becomes a desirable property of such systems.

FAULT TOLERANCE ..

- Fault-tolerance is particularly sought-after in high-availability or life-critical systems.
- Fault-tolerance can be achieved by anticipating exceptional conditions and building the system to cope with them, and, in general, aiming for self-stabilization so that the system converges towards an er free state.

FAULT TOLERANCE ..

Fault-tolerance can be achieved by duplication in the following three ways:

- Replication: Providing multiple identical instances of the same system, directing tasks or requests to all of them in parallel, and choosing the correct result on the basis of a quorum

FAULT TOLERANCE ..

- Redundancy: Providing multiple identical instances of the same system and switching to one of the remaining instances in case of a failure (fall-back or backup)
- Diversity: Providing multiple different implementations of the same specification, and using them like replicated systems to cope with errors in a specific implementation.

FAULT TOLERANCE ..

- A redundant array of independent disks (RAID) is an example of a fault-tolerant storage device that uses redundancy.
- A machine with two replications of each element is termed *dual modular redundant* (DMR). The voting circuit can then only detect a mismatch and recovery relies on other methods.

DMR

One variant of DMR is pair-and-spare.

Two replicated elements operate in lockstep as a pair, with a voting circuit that detects any mismatch between their operations and outputs a signal indicating that there is an error.

Another pair operates exactly similarly.

A final circuit selects the output of the pair that does not proclaim that it is in error.

TMR

A machine with three replications of each element is termed triple modular redundant (TMR). The voting circuit can determine which replication is in error when a two-to-one vote is observed. In this case, the voting circuit can output the correct result, and discard the erroneous version.

Example :

Consider this:

P_1 produces x

P_2 produces x

P_3 produces y

$P_1 = P_2$ but $P_1 \neq P_3$ and $P_2 \neq P_3$

Hence P_1 and P_2 have produced identical result but not P_3.

Conclusion: P_3 has developed a fault

(Assumption: x is the correct result)

There is a BUT here !!!

RECOVERY

Recovery from errors in fault-tolerant systems can be characterised as :

- roll-forward
- roll-back

Roll-forward recovery takes the system to a state at a time and corrects it according to the desirable attainable state, to be able to move forward.

RECOVERY contd..

Roll-back recovery reverts the system state back to some earlier, correct version.

The technique of *checkpointing* may be utilized for roll-back recovery.

Roll-back recovery requires that the operations between the checkpoint and the detected erroneous state can be made idempotent.

RECOVERY contd..

- Depending on the type/characteristic of application and error, some systems make use of both roll-forward and roll-back recovery for different errors or different parts of one error.
- However, sometimes the consequences of a system failure may be catastrophic, or the cost of making it sufficiently reliable may be very high.

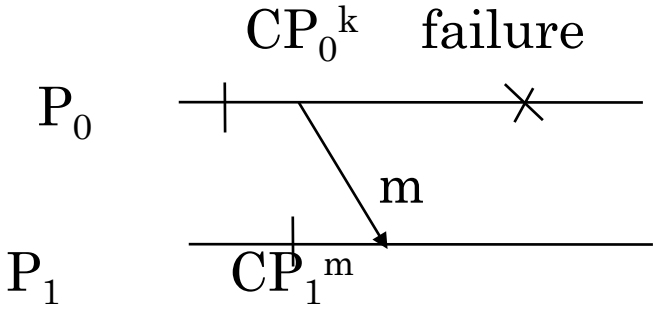
CHECKPOINTING AND *RECOVERY*

stable storage is periodically used to save process-states during failure-free execution. Each such saved state is called a *checkpoint*.

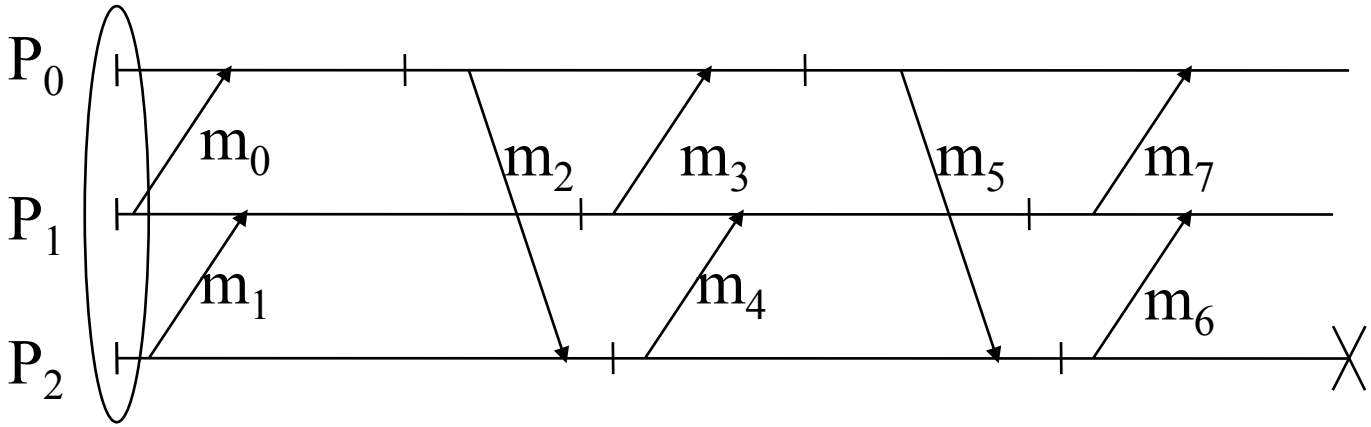
A consistent *global* checkpoint consists of checkpoints from each of the processes (also called *local* checkpoints) in the system.

CHECKPOINTING AND *RECOVERY..*

- In message-passing systems recovery gets complicated because messages induce inter-process dependencies



DOMINO EFFECT

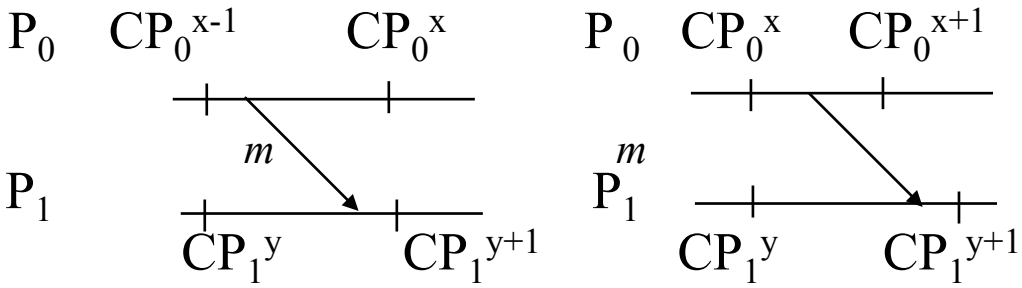


CHECKPOINT…

- A global checkpoint is *consistent* if and only if all its pairs of local checkpoints are consistent.
- Consistency is regarded as the scenario where if a sender 'S' sends a message 'm' *before* it has taken its i-th checkpoint, then message 'm' must be received by a receiver 'R' *before* the receiver has taken its i-th checkpoint.

CHECKPOINTING….

Figure below (a) shows a missing message *m* with respect to CP_0^x, CP_1^y whereas figure (b) shows an orphan message *m* with respect to CP_0^x and CP_1^{y+1}.



LOG BASED RECOVERY

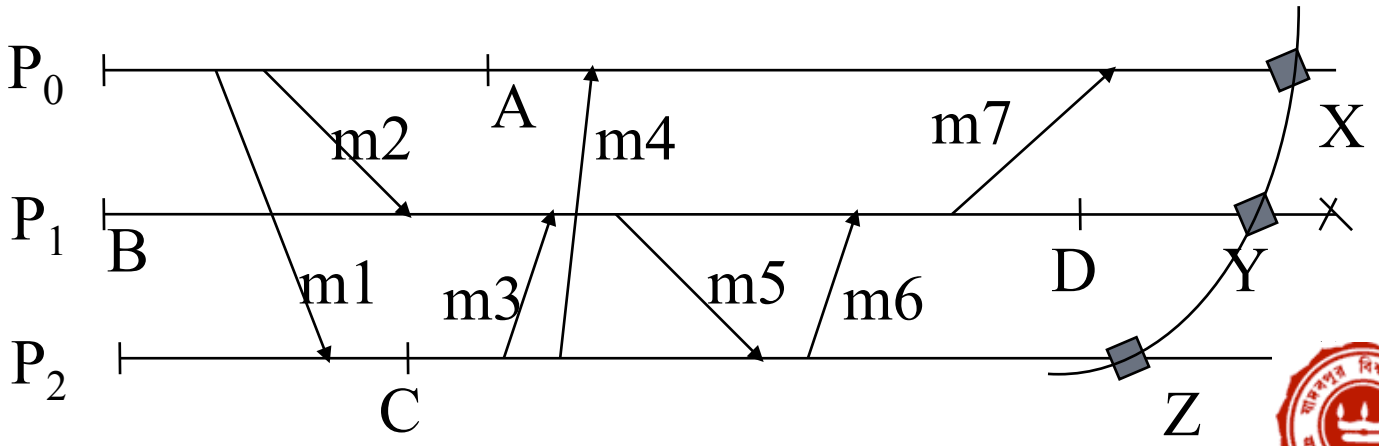
~Combines checkpointing with logging of nondeterministic events such that a process can deterministically recreate its pre-failure state even if that state has not been checkpointed.

~Relies on piecewise deterministic assumption.

~pessimistic logging, optimistic logging, causal logging.

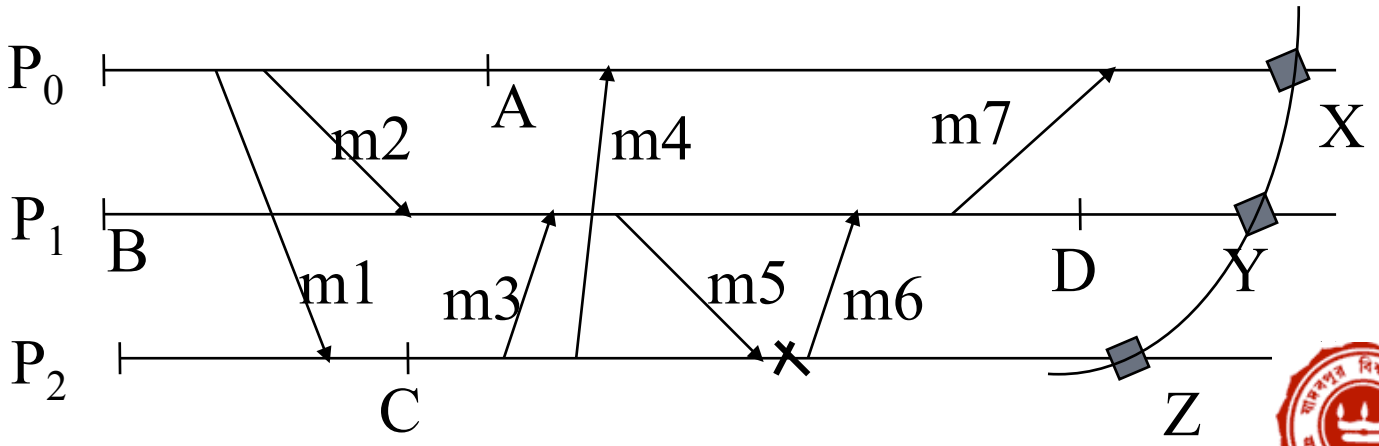
PESSIMISTIC LOGGING

- Assumption: A failure can occur after any nondeterministic event
- Determinant of each nondeterministic event is logged to stable storage before the event is allowed to affect computation.
- In addition checkpoints are also taken.



OPTIMISTIC LOGGING

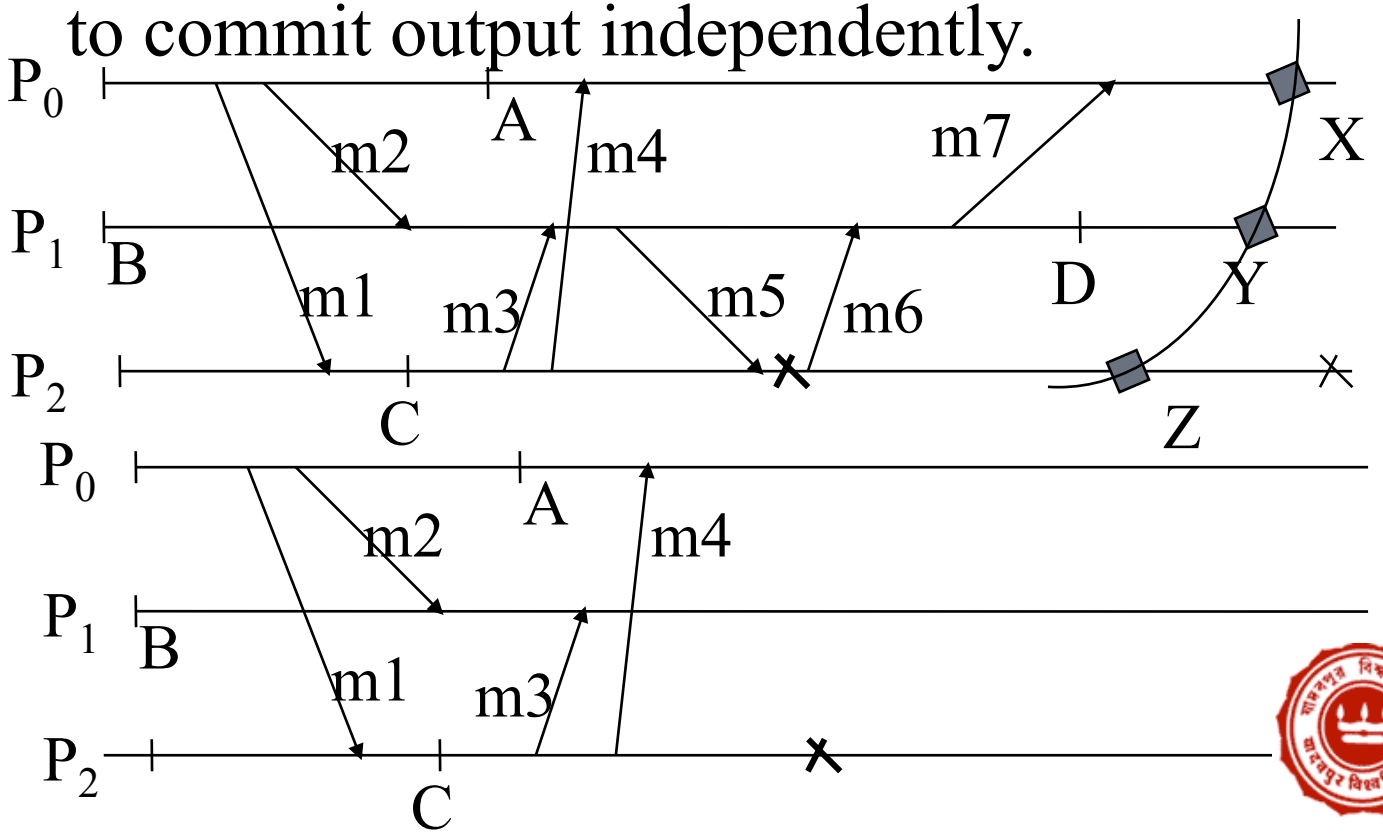
- Assumption: Logging will complete before a failure occurs.
- Determinant of each nondeterministic event is kept in volatile log which is periodically flushed to stable storage.
- In addition checkpoints are also taken.



CAUSAL LOGGING

– Like optimistic logging it avoids synchronous access to stable storage except during output commit.

– Like pessimistic logging it allows each process to commit output independently.



CHECKPOINTING ALGORITHMS MAY BELONG TO ANY OF THE FOLLOWING CATEGORY:

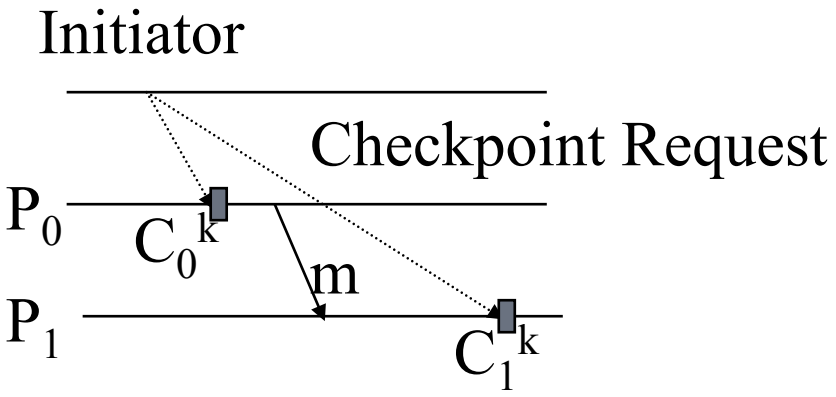
- Clock-based checkpointing protocols
  - Loosely synchronized clocks are used.
  - A process takes a checkpoint
  - Then waits for a period equal to the maximum deviation between the clocks and the maximum time to detect a failure in another process in the system

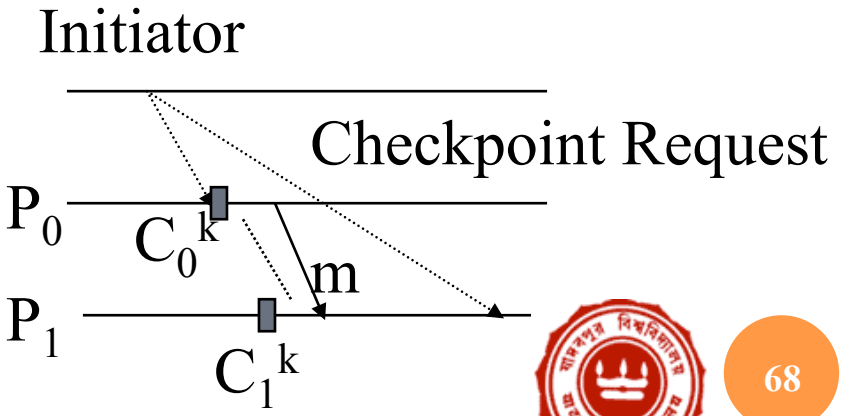
CHECKPOINTING PROTOCOLS WITHOUT CLOCKS

•Checkpoint initiator sends requests to all other processes in the system.

•It is of two types:

- Blocking type
- Non blocking type





TRADITIONAL CHECKPOINTING ALGORITHMS ARE INSUFFICIENT BECAUSE OF ISSUES LIKE:

~Low Bandwidth: Rollback recovery schemes requiring large number of message transfers or piggybacking large information require large bandwidth

~Power Consumption: Checkpointing at every initiation consumes power

ISSUES CONTINUED..

~ Limited Memory Space: Multiple checkpoints take up spaces

~ Handling Mobility:Locating a mobile host increases message complexity and communication delay.

A FEW SCHEMES

AN EFFICIENT COMMUNICATION-INDUCED CHECKPOINTING SCHEME

During normal operation whenever an application message is received a checkpoint is taken (*communication induced*). At the time of disconnection from an MSS, an MH takes a *local* checkpoint.

SCHEME ..

- The protocol employs one process periodically as the initiator (central) process to find out the *Globally Consistent Checkpoint*.
- The initiator collects information regarding latest (communication-induced or local) checkpoints of all the participating processes and finds out which processes need to take a forced checkpoint in order to maintain consistency.

TWO-TIER COORDINATED CHECKPOINTING ALGORITHM

- Sending Rule for MSS P:
- P first records state of each outgoing channels and then sends a marker along each of them before sending any message through it.

CHECKPOINTING ALGORITHM CONTD…

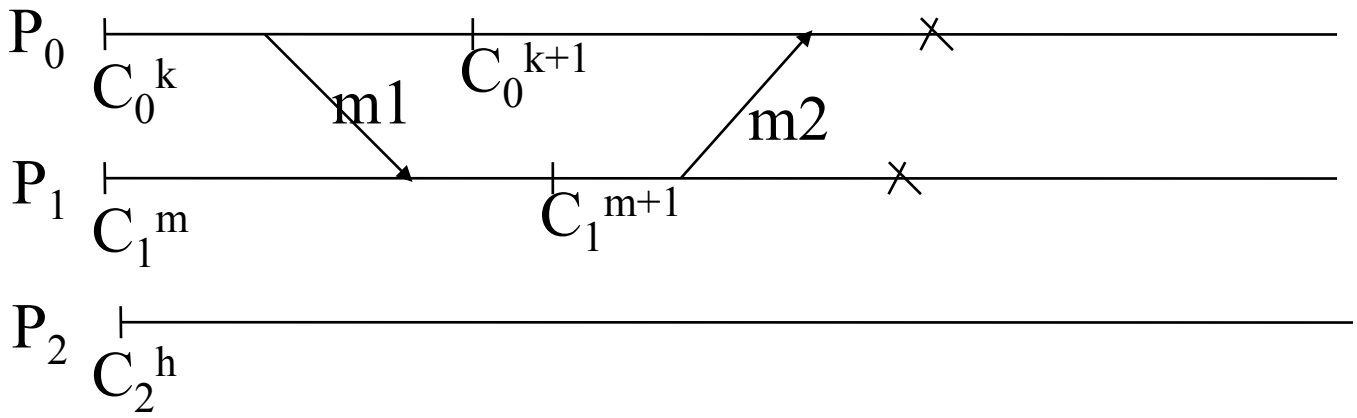
- Receiving Rule for MSS Q:
- On receiving a marker along a channel c:
- If MSS Q has not recorded its state,
- Records states of all MHs connected to it, Records state c as the empty sequence,
- Performs the Marker-Sending Rule.
- Otherwise, MSS Q records the states of channel c after the state of MH is recorded, until the marker is received through channel.

CHECKPOINTING PROTOCOL FOR MINIMAL SET OF NODES

- A checkpoint initiator (usually an MSS) that sends periodic checkpointing requests is assuumed.
- MHs (after receiving the checkpointing request) decide whether checkpoint is to be taken or not. (The decision may depend upon certain factors)

CHECKPOINTING PROTOCOL FOR MINIMAL SET OF NODES …

- MHs take checkpoints if needed.
- MSSs take checkpoints at every initiation regardless of others (i.e. MHs).
- Unacknowledged messages are logged at the MSSs.

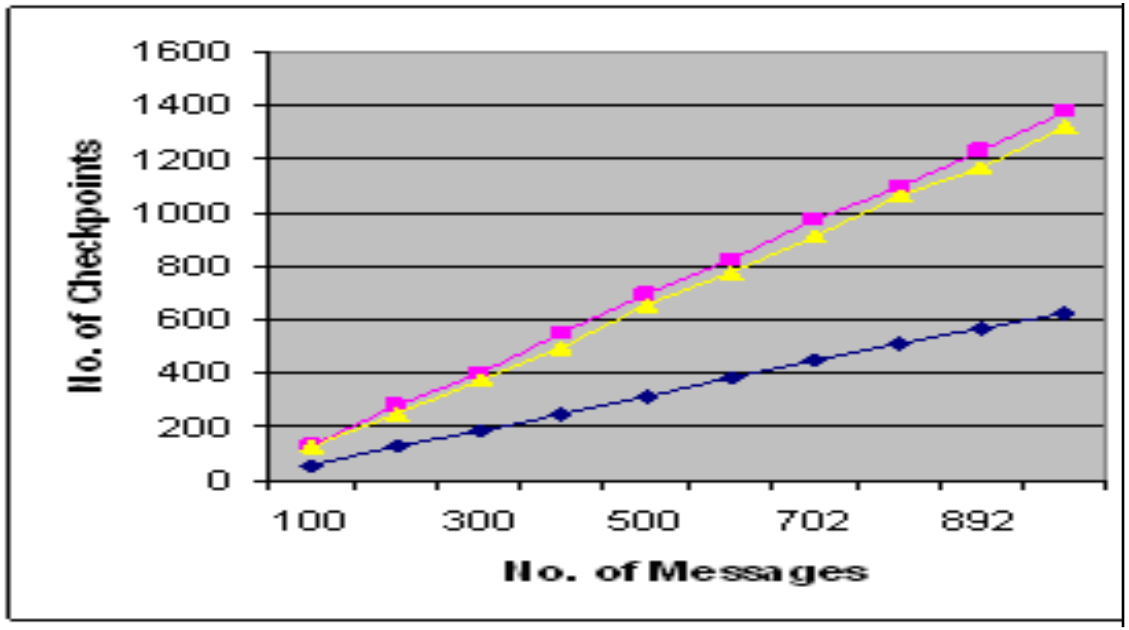


A new checkpoint is not needed

COMPARISON OF SCHEMES

	EFFICIENT COMMUNICATION-INDUCED CHECKPOINTING	TWO-TIER COORDINATED CHECKPOINTING ALGORITHM	Checkpointing Protocol for Minimal set of Nodes
LOW BANDWIDTH & LOW POWER CONSUMPTION	No information is needed to be piggybacked with application messages.	Messages are handled by the MSS instead of relaying messages directly to an MH.	Checkpoint requests are not sent to all MHs but to those that have communicated in the last checkpoint interval.
LIMITED MEMORY SPACE	MHs individual memory is flushed each time a GCC is taken.	Checkpoints and message logs are kept at MSSs.	Unacknowledged messages are logged at MSSs.

SIMULATION RESULTS



CHECKPOINTING IN MCS (AGAIN ..)

- The *home station (HS) of an MH*
  - An MSS through which an MH can communicate with the rest of the system
- If an MH moves to the cell of another base station, wireless channel to the old MSS is disconnected and a wireless channel in the new MSS is allocated
- Mobile IP is used as the underlying protocol for message transmission
- During disconnection interval only local events take place at MH

ASSUMPTIONs

- The MSSs are assumed to be fault-tolerant
- There is no shared memory or common clock among the nodes
- Communication and synchronization between the nodes is via message-passing only
- Checkpointing requests as well as computation messages from other MHs may be queued at the old MSS during this disconnection interval
- Fail-stop model of communication is assumed

OUR SCHEME

- The scheme proposes that a checkpoint initiator sends checkpointing requests from time to time to all MSSs only
  - An MSS finds out whether the MHs of which it is the HS needs to take checkpoint or not
  - Each MSS also maintains an account of the communication activities in the current checkpointing interval of the concerned MHs
  - An MSS forwards the checkpointing request only to those MHs (to which it is HS) if it finds that those particular MHs were active during the current checkpointing interval
  - Hence only a few selective MHs are able to take checkpoints after the checkpointing request reaches them
  - All MSSs also take checkpoints at every initiation

OUR SCHEME …

- Messages received are logged in the stable storage of the corresponding HS
  - Thus during recovery only the failed process needs to restart its computation from its last saved checkpoint while other processes can execute computation without any interruption
- Only unacknowledged messages are saved in the HS of the sender MH
- The recovering once-faulty process informs other processes (only the MSSs via its HS) that it is recovering
  - Receiving this message an MSS would start sending unacknowledged message (if any)

SALIENT FEATURES OF THIS SCHEME

- The problem of concurrent initiation does not arise as the MSSs take turn to act as the initiator
  - Hence initiator never becomes a bottleneck
- The coordinated checkpointing overhead is also minimized since the present scheme is not like the two-phase commit protocol
  - Hence bandwidth, power both are conserved
- Only the faulty process needs to recover leaving the others unaffected

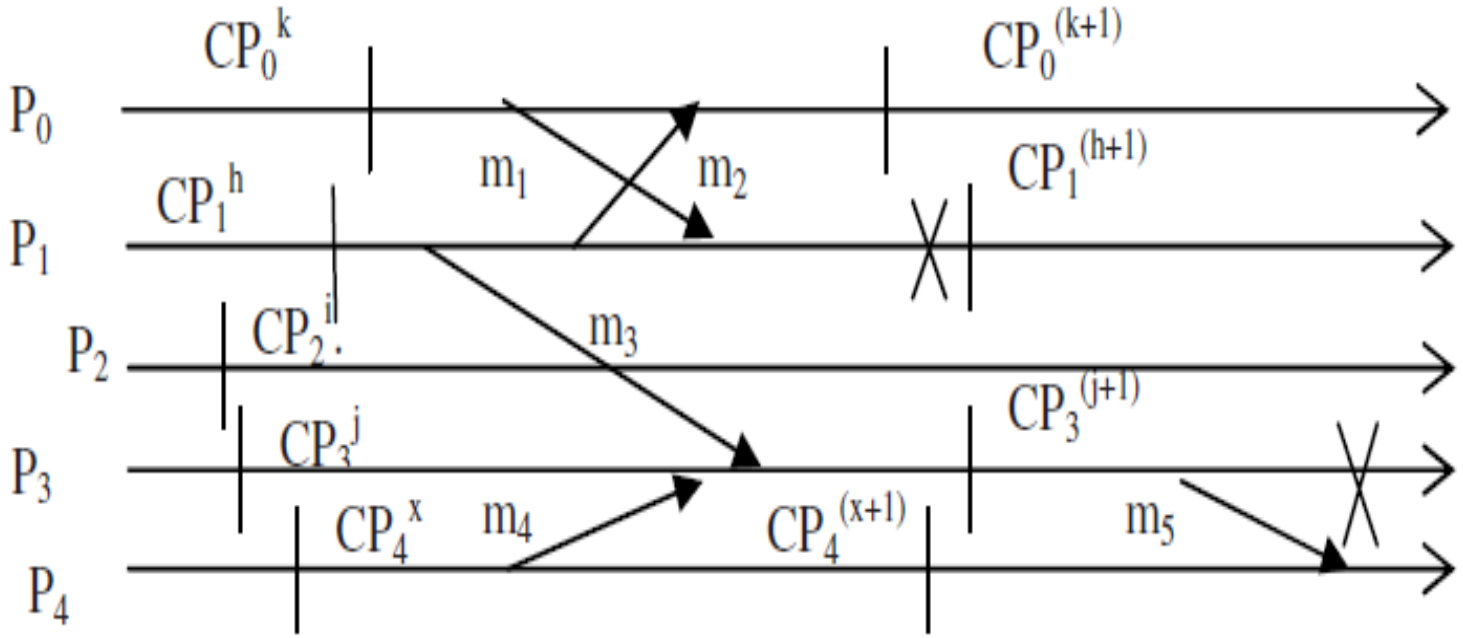
SALIENT FEATURES ….

- This scheme conserves energy and bandwidth since not all MHs need to take checkpoint
- Also this decision is taken by the HS of the MHs thereby relieving the MHs from executing an algorithm, thus saving battery power
- Memory constraint of mobile nodes is considered here and

SALIENT FEATURES ….

- the entire message logs required by the protocol are kept at the stable storage of the HS
  - This does not incur any extra overhead since the underlying network protocol (mobile IP) ensures that all communication is usually done via the HS.

AN EXAMPLE


CP_0^k
$CP_0^{(k+1)}$
P_0
m_1
m_2
CP_1^h
$CP_1^{(h+1)}$
P_1
CP_2^i
m_3
P_2
$CP_3^{(j+1)}$
CP_3^j
P_3
CP_4^x
m_4
$CP_4^{(x+1)}$
m_5
P_4

AN EXAMPLE ..

- MSS_1 is acting as the HS for MHs P_0, P_3 and P_4 and MSS_2 for the MHs P_1 and P_2
- When MSS_1 receives checkpointing request it first finds that
  - MHs P_0, P_3 and P_4 need to take their $(k+1)^{th}$, $(j+1)^{th}$ and $(x+1)^{th}$ checkpoints respectively since
    - P_0 has received m_2 and sent m_1,
    - P_3 has received m_3 and m_4 and
    - P_4 has sent m_4 in their respective last checkpointing intervals

AN EXAMPLE ..

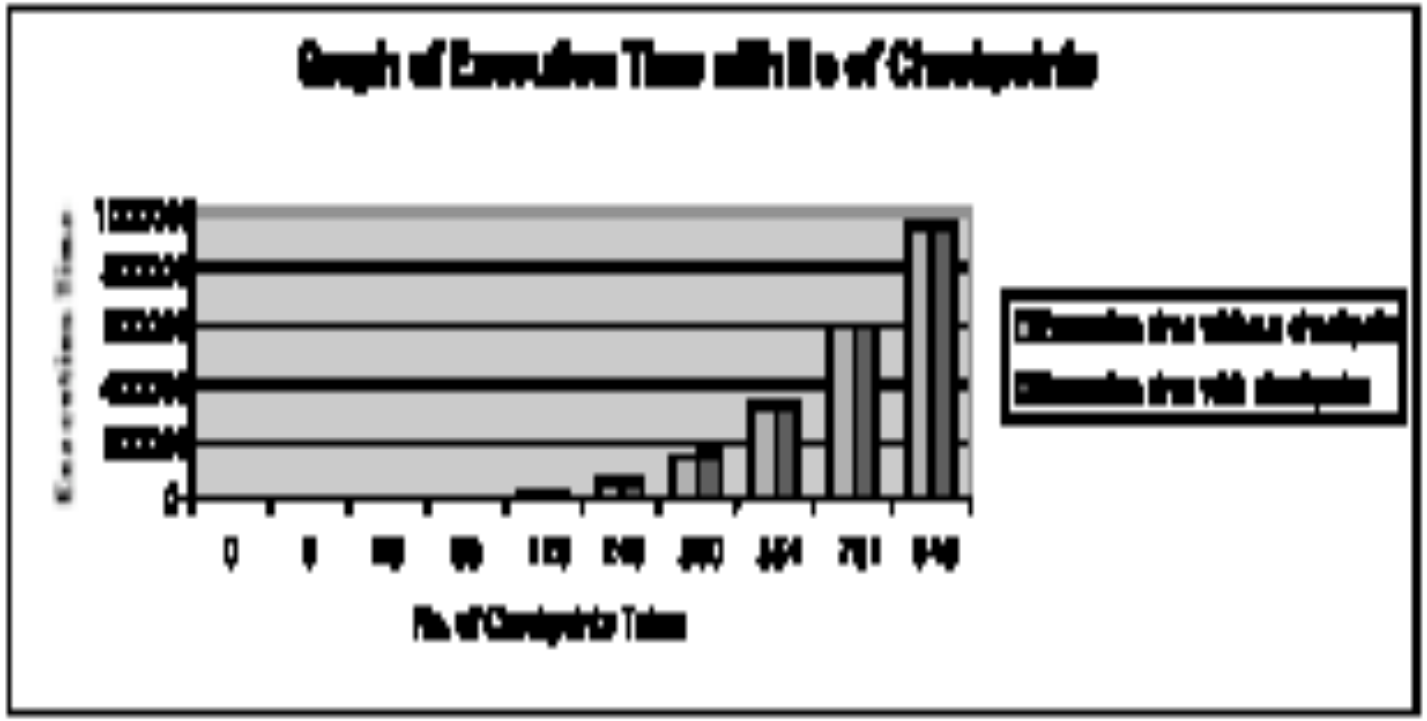
- MSS_2 finds that
  - only P_1 needs to take its $(h+1)^{th}$ checkpoint since
    - it has sent m2 and m_3 in its last checkpointing interval
  - But P_2 does not need to take any checkpoint since
    - it has not communicated since its last checkpoint CP_2^i
- When CP_0^{k+1} is created, CP_0^k is deleted and information

  related to m_1 and m_2 are transferred to the old log

RESULTS

- The checkpoint overhead is calculated to be the difference between the execution times with checkpoints and without checkpoints
- Hence our algorithm incurs very low checkpoint overhead

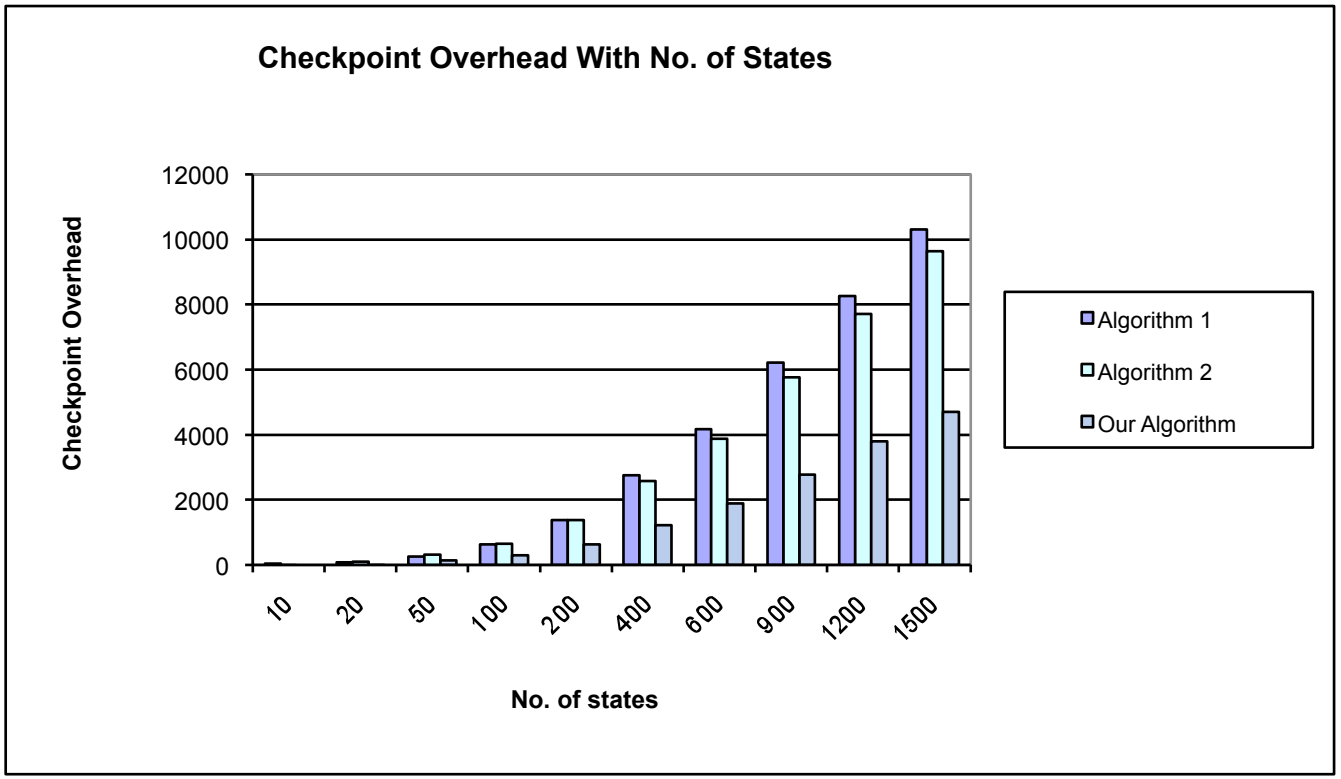
RESULTS



RESULTS ….

- Our algorithm is compared with two other approaches.
- From the beginning of algorithm execution an MH has undergone these states each of which may be
  - local computation,
  - sending a message to some other MH or
  - receiving a message from other MH

RESULTS …


Checkpoint Overhead With No. of States
Checkpoint Overhead
12000
10000
8000
6000
4000
2000
0
10
20
50
100
200
400
600
900
1200
1500
No. of states
Algorithm 1
Algorithm 2
Our Algorithm

TRIPLE MODULAR REDUNDANCY (TMR) IN WIRELESS NETWORKS

- The Triple Modular Redundant (TMR) [1] system is one well known method of achieving fault tolerance
  - When a minimum of three processors also known as replicas form a redundant group and perform replicated processing
- Input data is made available to all the replicas
  - They perform identical processing and distributed voting

THE IDEA

- A typical node in the proposed Wireless-Triple Modular Redundancy (WTMR) scheme consists of
  - Two MSSs and an MH connected to any one of the MSS,
  - each of which is also called replica
- Upon receiving input data, each replica checks to find if the data matches with each other
  - Thus input agreement is ensured
- Receiver of a message should be able to
  - Authenticate the validity of the received message and
  - Detect any possible corruption that the message might have suffered
- Digital signature can be used to counter these threats

THE IDEA ….

- Signatures on data rule out dependence on time-out and possibility of executing with two simultaneous faults because
  - If there is a stage during the computation where the result bears two signatures, this implies that
    - the result is endorsed by two replicas
  - In such cases a single result is enough for a replica to carry out its task
  - Implication
    - There may have been more than one fault in the TMR node and hence only one result has reached a replica

CHECKPOINTING IN WTMR

- Coordinated checkpointing approach is followed
- Each MSS in the system takes turn to act as coordinator
- It is sufficient that only any one replica (either MSS or MH) takes the checkpoint since computation is identical in all of them
  - MSSs are preferred
- Communication overhead is minimized in the sense that messages regarding checkpointing activity do not use wireless channels at all

CHECKPOINTING IN WTMR contd..

- If a disagreement is reached in voting at any point during execution
  - decision of recovery is taken among the replicas in the TMR node and
  - execution resumes from the last saved checkpoint
- The decision of recovery would have to reach all other TMR nodes in the system for maintaining consistency

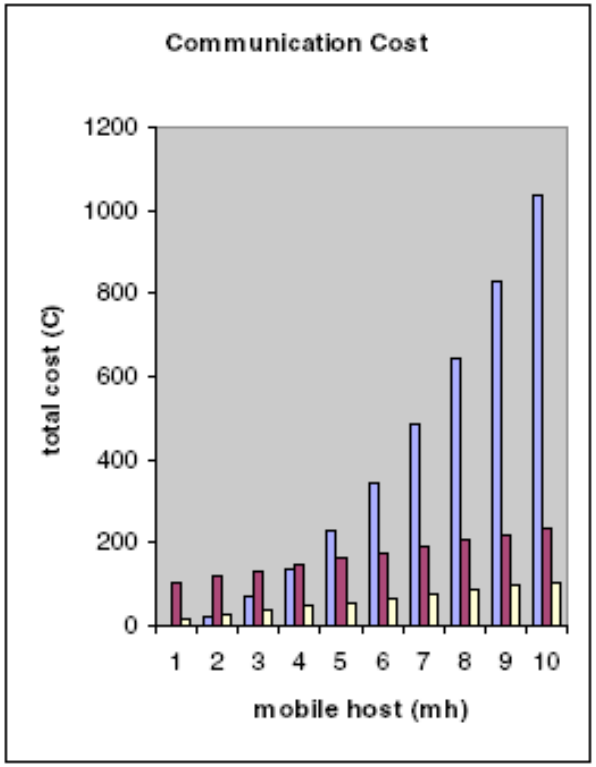
RESULTS

- C_w: cost of wireless networking
- C_f: cost of fixed networking
- mh : number of mobile hosts
- mss : number of mobile support stations
- m : mobility ratio (that is, the probability that an MH has moved from its own MSS to another MSS)

RESULTS …

- Approximate cost of
  - traditional coordinated checkpointing algorithms
    - $mh*(mh-1)*(2*C_w+C_f) + mh*(mh-1)*m*C_f$
- Approximate cost of
  - Proposed WTMR checkpointing algorithm is
    - $(mss-1)*C_f + (mss-1)*mh*m*C_f$
    - $(mss-1)*C_f$ during checkpointing
    - The additional cost incurred during recovery only would be
      - $(mss-1)*mh*m*C_f$

RESULTS …



- C_f = 1 unit of cost, C_w = 5 unit of cost, m= 0.5, mss= 10, and mh=10

- The column in violet colour shows cost of traditional coordinated checkpointing algorithms
- The column in brown colour shows cost of checkpointing algorithms of [3],
- The column in cream colour shows cost of present checkpointing algorithm

CHOOSING A SCHEME FOR FAULT TOLERANCE

- The discussed algorithms (and a host of others not discussed here) are able to meet the requirements of mobile computing system.
- The constraint of limited bandwidth is no more serious now a days and this is encouraging the developers to use coordinated checkpointing algorithms in mobile distributed systems.

RELIABILITY

SOFTWARE RELIABILITY

- Parameters:
- Average total number of failures:
  (with respect to a number of independent instantiations of an identical software)
- Failure intensity:
  Number of failures per time unit
- Mean Time To Failure (*MTTF*): $MTTF = \frac{1}{\lambda(t)}$
- t may denote elapsed execution time

IMPORTANCE OF SOFTWARE RELIABILITY

- In *safety-critical* systems, certain failures are fatal. This implies reliability has to be attained at whatever high levels and probably at very high costs (code redundancy, hardware redundancy, recovery blocks, *n* version programming…).

software reliablity ..

- In *non-safety-critical* systems a certain failure rate may usually be tolerable.
  - This is obviously subjective.
  - It is pretty hard to define a tolerable limit
  - Possibly this limit will vary from project to project

ESTIMATION WITH FAULT TOLERANCE

WHY RELIABILITY ESTIMATION ?

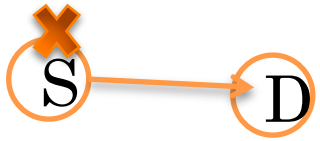
- The environmental conditions along with user mobility affects reliability of MCSs
- These issues are hindrances to analysing the reliability of mobile computing system

ESTIMATION ..

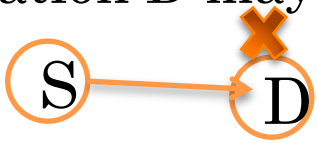
- The analysis is thus dependent on mobility modeling approaches describing effects of node mobility
- Distinguishing between temporary disconnection due to node movement and permanent disconnection due to hardware/software failure is a challenging issue
  - Often nodes become disconnected because of the high handoff rate
  - that the MSSs cannot tolerate

TYPES OF FAILURES CONSIDERED

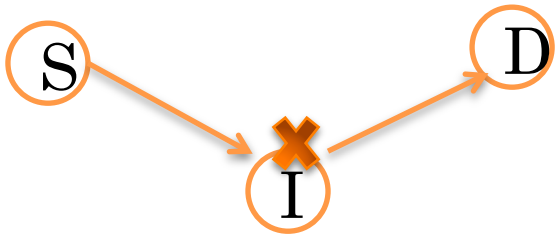
- In our work we have considered three kinds of Failure
  - Source S may fail



  - Destination D may fail



  - Intermediate node (say I) may fail



DEFINITIONS

- The probability of successful communication between S and D is called *two terminal reliability*
- This definition is extended to include all operating nodes in the network and can be termed as *all operating terminal reliability* (*AOTR*)

CONCEPTS

- The proposed method transforms existing reliability analysis methods and mobility modeling techniques in such a way that the parameters characterizing a mobile distributed system can be identified and the AOTR of the system may be estimated.

PROPOSED MODEL

- We use Monte Carlo based simulation method to determine AOTR and Two Terminal Reliability
- Smooth Random Mobility Model is used to recognize the effects of mobility
- If at t time instant the distance $d_{ij}(t)$ between MH_i and MSS_j is less than the cell radius τ then we say that MH_i is connected to the network via MSS_j

MODEL …

- Fault tolerance of the nodes is considered
- Reliability calculation considers both
  - node failure (Weibull distribution) and
  - link failure (due to node movement)
- The proposed model also distinguishes a failed link from a temporary disconnection using a timeout interval

MODEL …

- The reliability estimation procedure also includes the maximum velocity of MHs that is being supported by the MSSs to encounter the maximum handoff rate that can be supported by a MSS

MODEL ….

- Fault tolerance of the nodes is considered
  - A node fails according to Weibull distribution
  - We categorize node failure into two types
    - a recoverable fault
      - The process can be recovered using fault tolerance mechanism
    - A permanent fault
      - The process cannot be recovered during the runtime of the application
  - After a process fails a Poisson event decides when a recoverable or permanent fault occurs

MODEL ….

- Message transfer is simulated in a way that each node can send message to any other node
  - The receiver is selected randomly
  - The receiving MH_i receives the messages if
    - It is/remains connected within the end-to-end message transmission delay time [10] (=250ms approximately) and
    - if the MH has not failed according to Weibull distribution

MODEL ….

- The proposed model also distinguishes a failed link from a temporary disconnection
  - If an MH remains disconnected for less than a predefined connection tolerance limit then its previous association with MSS is not disrupted
- The maximum handoff rate supported by any MSS is fixed
  - If any MH moves too fast through the cells, it will be disconnected from the network even if the distance from its nearest MSS is well within the transmission range

TERMS

- Network Coverage

$$\frac{\sum_{i=1}^{|N|} \lambda_i(t)}{|N|}$$

  - Here $\lambda_i(t)$ represents connectivity of MH_i to the network at time t
  - N denotes the no. of MHs
- All Operating Terminal Reliability

$$AoTR = \frac{\sum_{q=1}^{|No.ofsim|} \prod_{i=1}^{|N|} check_i(t)}{Q}$$

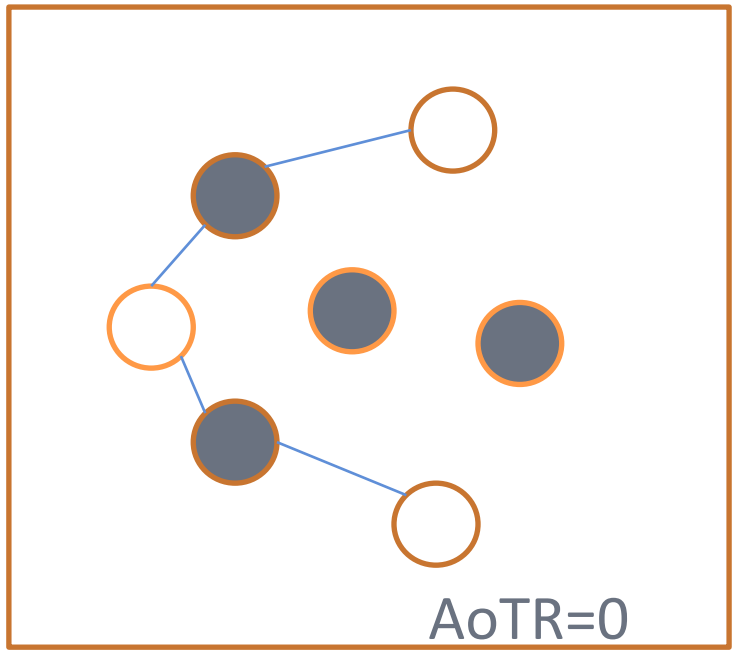
  - Here check is a variable that is set to 0 if the node has failed and is set to $\lambda_i(t)$ otherwise
  - Q represents the no. of simulation steps needed by Monte Carlo simulation

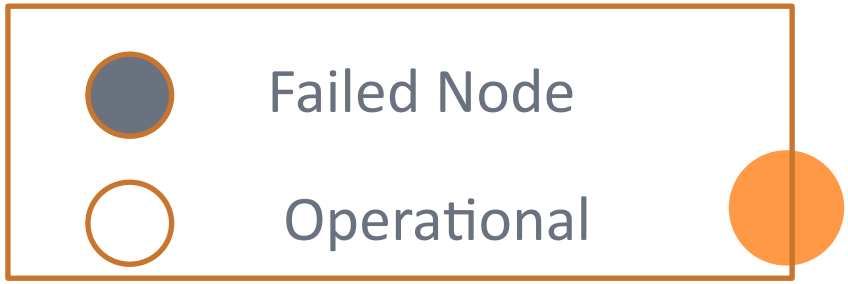
TERMS …

- Two Terminal Reliability
  - the probability of a successful path between a source MH_i to a destination MH_j

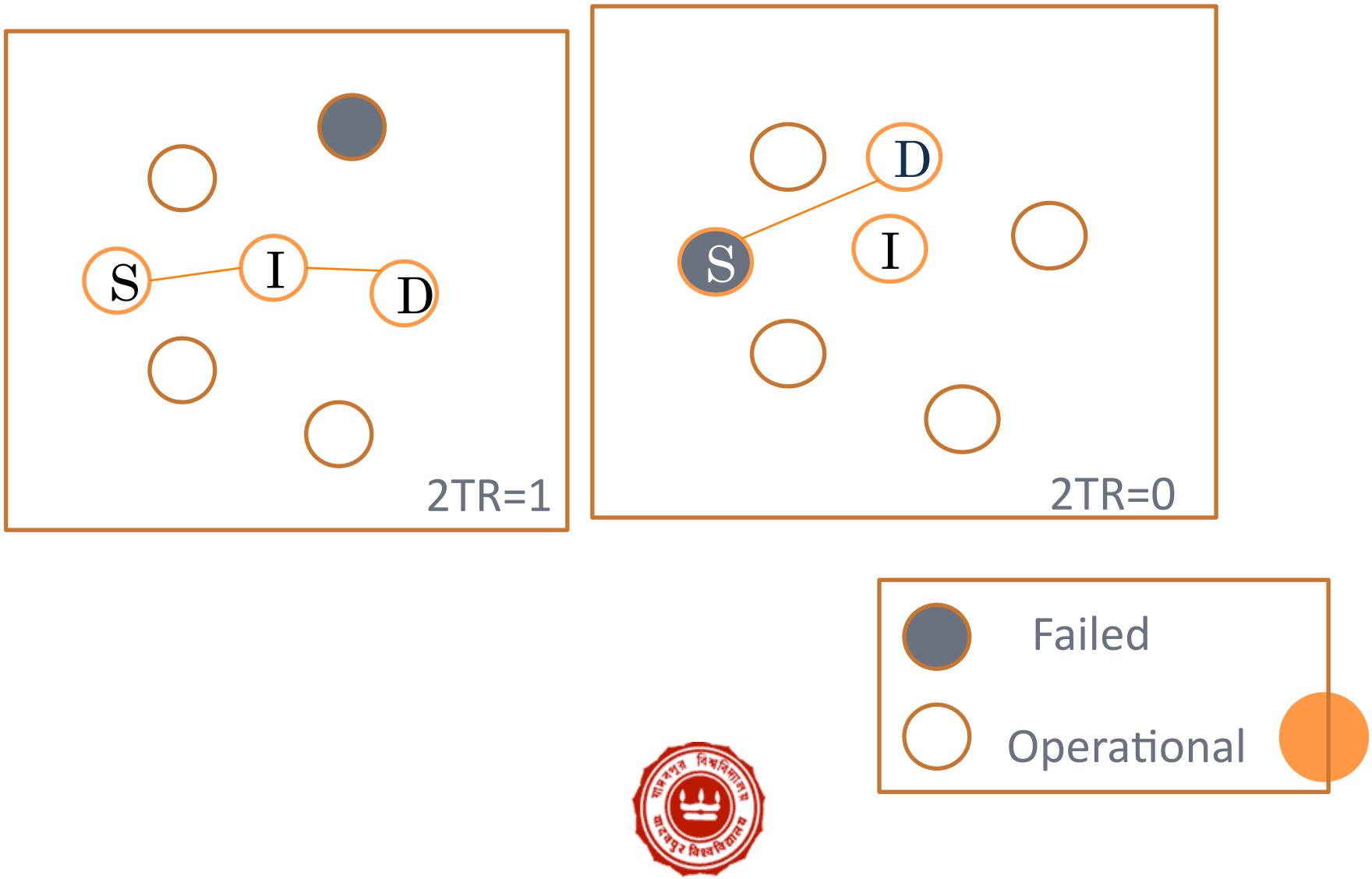
$$2TR_m = P(\Lambda_{|N|}(t) = 1)$$

All Operating Terminal Reliability (AoTR)





Two Terminal Reliability (2TR)



AN EXAMPLE

Time	x	y	λ(t)	Lt(t)
	MH_1			
t	1	5	1	MSS_1
t+Δt	1.09	5.05	1	NC
t+2Δt	1.18	5.05	1	NC
t+3Δt	1.27	5.02	0	NC
t+4Δt	1.36	5	0	NC
t+5Δt	1.44	5	0	NC
t+6Δt	1.54	5	0	NC
t+7Δt	1.63	5	0	NC
	MH_2			
t	0	4.00	1	MSS_1
t+Δt	0.09	4.00	1	MSS_1
t+2Δt	0.18	4.00	1	MSS_1
t+3Δt	0.27	4.00	1	MSS_1
t+4Δt	0.35	4.00	1	MSS_1
t+5Δt	0.44	4.00	1	MSS_1
t+6Δt	0.53	4.00	1	MSS_1
t+7Δt	0. 62	4.00	1	MSS_1
	MH_3			
t	2	2.2	1	MSS_1
t+Δt	2.09	2.2	1	MSS_0
t+2Δt	2.18	2.2	1	MSS_0
t+3Δt	2.27	2.2	1	MSS_0
t+4Δt	2.36	2.2	1	MSS_0
t+5Δt	2.45	2.2	1	MSS_0
t+6Δt	2.54	2.2	1	MSS_0
t+7Δt	2.63	2.2	1	MSS_0
	MH_4			
t	1	0	1	MSS_0
t+Δt	1.09	0.01	1	MSS_0
t+2Δt	1.18	0.02	1	MSS_0
t+3Δt	1.27	0.04	1	MSS_0
t+4Δt	1.35	0.05	1	MSS_0
t+5Δt	1.44	0.07	1	MSS_0

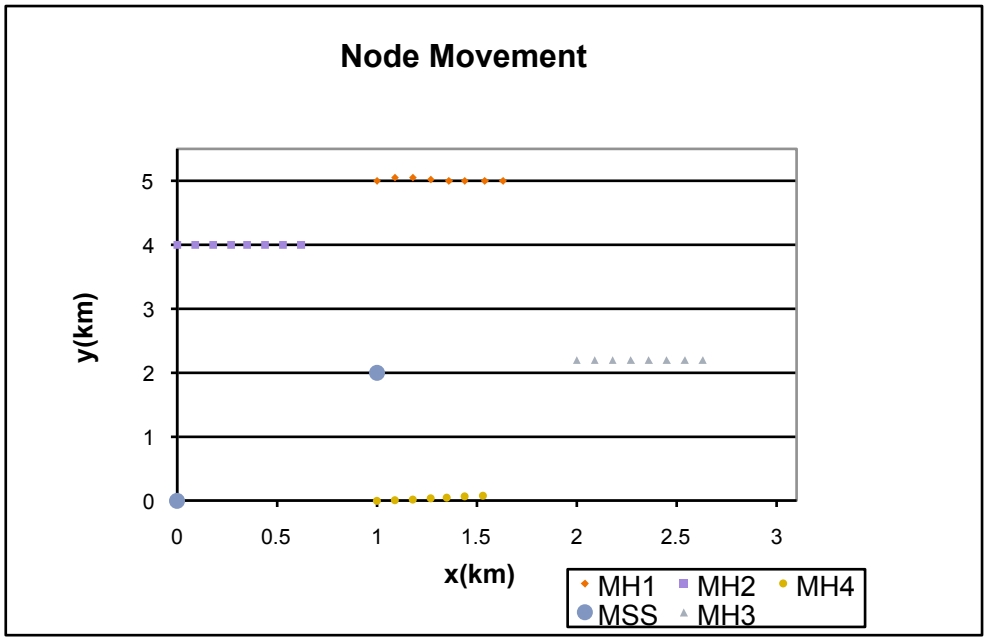
AN EXAMPLE

- Our simulation takes 4 MHs (N=4) situated at points given in the table
- There are two MSSs (M=2) at (0,0) and (1,2)
- The cell radius is taken as 3km
- The distance between MH_1 and MSS_0 is $\sqrt{(1-0)^2+(5-0)^2}$ that is more than 3 but distance from MSS_1 is less than 3 $\sqrt{(1-1)^2+(5-2)^2}$
  - So MH_1 is connected to MSS_1
- MH_3 was initially connected to MSS_0 but later on at $t=t+\Delta t$, it has moved to MSS_1 causing a handoff.

AN EXAMPLE ..

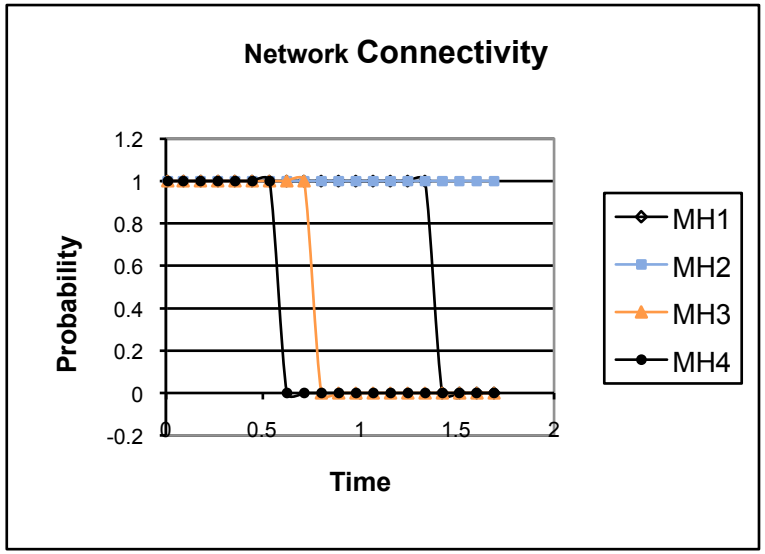
- The individual node reliability is described by Weibull shape parameter $\beta = 1.5$ and scale parameter $\theta = 1000$
- The transmission range $\tau = 3$km and $\sqrt{(1-0)^2 + (5-0)^2}$ $\sqrt{(1-1)^2 + (5-2)^2}$
- the maximum and minimum velocity with which a MH can move in the network is 30km/hr (as WiMAX can easiliy support such user mobility) and 0.1km/hr (for pedestrians) respectively
- Finally the network coverage and all operating terminal reliability are calculated

RESULTS



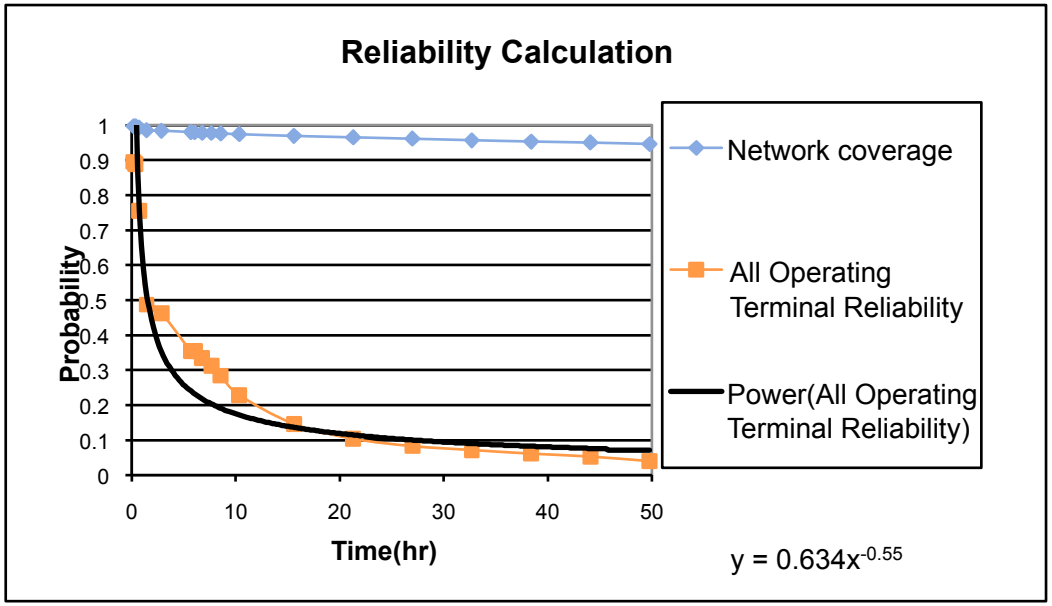
- Smooth movement of the MHs signifies the mobility model

RESULTS contd.



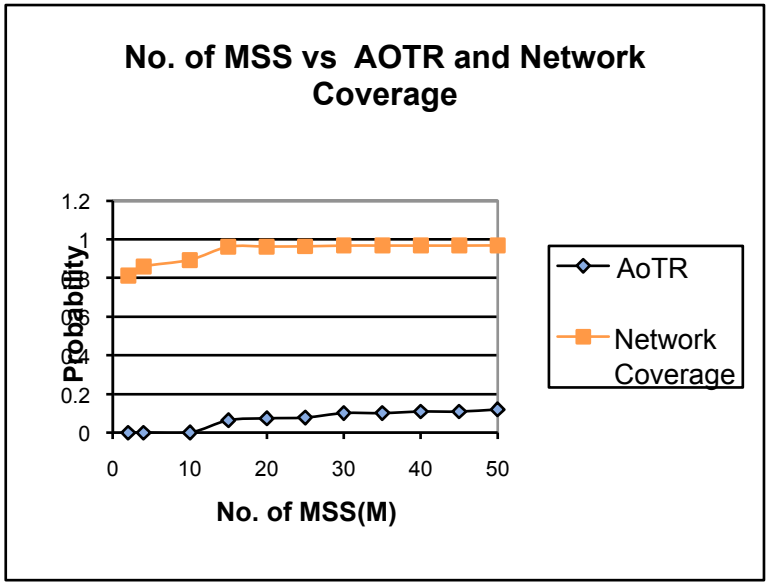
- As the movement is smooth, once an MH looses connectivity, there is very little probability of its regaining connectivity instantly

RESULTS contd.



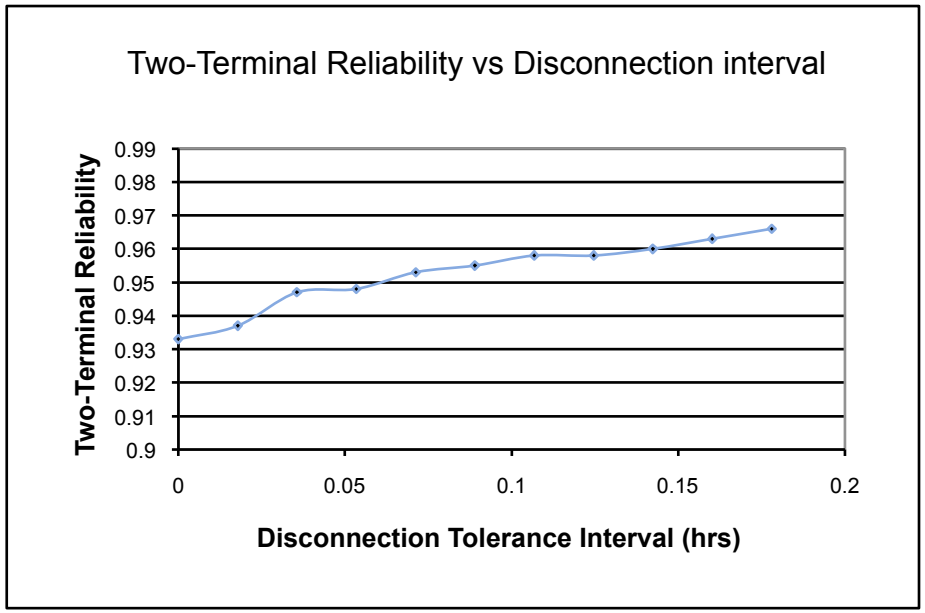
- The network coverage falls gradually over time as the node failure rate is expected to increase
- Time is inversely proportional to the square of AOTR

RESULTS contd.



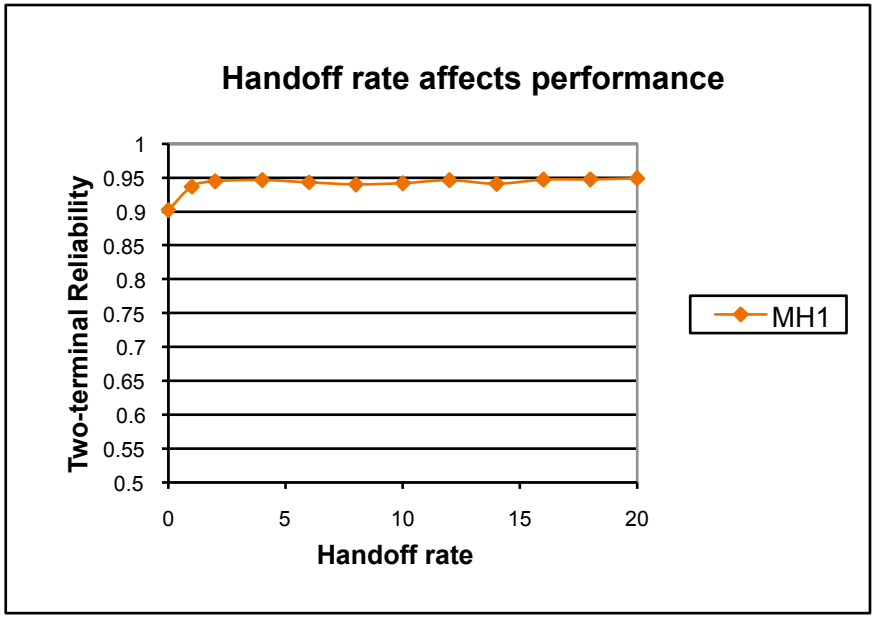
- Addition of new MHs does not affect AOTR but their positions relative to the MSSs do
- Thus introducing new MSSs improve AOTR well as the network coverage

RESULTS contd.



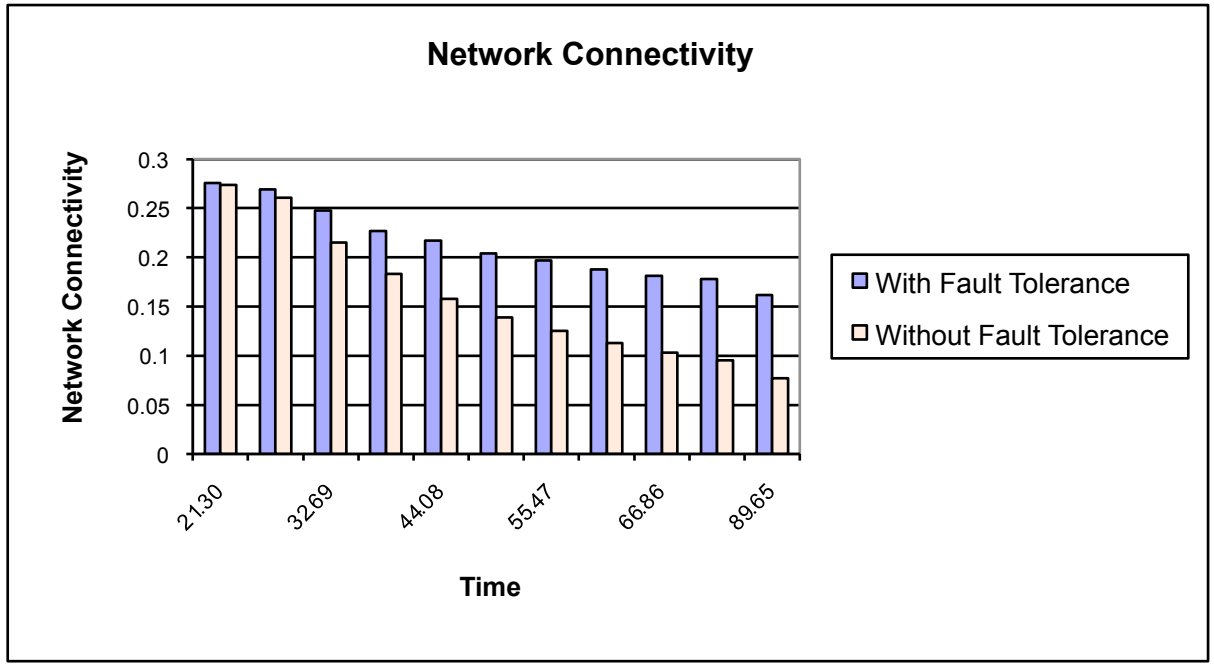
- Increasing disconnection interval improves system reliability estimate
- This in turn increases log of received messages kept at the MSSs
- Large disconnection intervals can cause time-outs in the higher layers.

RESULTS contd.



- If the network is designed to support increased or greater speed for moving nodes then the reliability estimate of the network improves with it
- But after a certain speed, saturation is reached when performance does not improve significantly with change in maximum supported speed V_{max}

RESULTS contd.



- The graph clearly shows that introducing fault tolerance results in better network connectivity and the performance gradually improves with time

SECURITY

SECURITY

- Threats are potential violation to security
- Broad categories of threat:
  - Disclosure
  - Deception
  - Disruption
  - Usurpation

SECURITY THREATS OF A NETWORK

- Masquerading
- Unauthorized use of resources
- Unauthorized disclosure and flow of information
- Unauthorized alteration of resources and information
- Repudiation of actions
- Unauthorized denial of service

SECURITY THREATS IN A MOBILE NETWORK

- Nuisance attack
- Impersonation attack
- Interception Attack
- Replay attack
- Parallel session attack

SECURITY IN DEPENDABLE MCS

- Two aspects:
  - Secure checkpointing as fault tolerance technique
    - Secure storage of checkpoints and related information (message logs etc.)
    - Integrity of checkpointing protocol
  - Authentication of mobile hosts for reliability

SECURITY IN DEPENDABLE MCS …

- Checkpoints may be stored in stable storage using:
  - Passwords for accessing
    - Drawback – MSS to forward the password to MH
  - Cryptography – encrypt the checkpoint and decrypt it when required
    - Drawback – requires generation and maintenance of keys depending upon the technique

SECURITY IN DEPENDABLE MCS …

- Integrity
  - Origin integrity - Control messages (checkpoint request or recovery etc.) may be *digitally* signed that requires:
    - generation of keys
    - and their maintenance

Signatures ensure authenticity of the messages.

All MSSs and MHs have to be aware of others' signatures (to be able to verify)

SECURITY IN DEPENDABLE MCS …

- Data integrity – checkpoints/message logs may be forwarded using message authentication technique.
  - Requires computation at sending end and recomputation at receiving end
  - Size of checkpoints/message logs increase (since original is appended with the hash code of the same)

SECURITY IN DEPENDABLE MCS …

- Encryption and decryption
  - Symmetric key cryptography – single key to be maintained per communication partner, key may be obtained from a central server (MSS specially designated for the purpose) or may be generated using Deffie-Hellman protocol. Same key is used for both encryption and decryption. Each MSS has to maintain a large no. of keys.

SECURITY IN DEPENDABLE MCS …

- Asymmetric key cryptography – a pair of keys – public and private to be maintained by each MSS and MH. Encryption is done by pubic key of recipient assuring that the message may be correctly read only by the recipient after successfully decrypting it by its private key.
- Cryptography therefore ensures concealment of information.
- Thus, stored encrypted checkpoints may be read by intended recipient processes only.

SECURITY IN DEPENDABLE MCS …

- Digital signature – asymmetric key cryptography is used, sender uses its private key to "sign" the message and the receiver uses sender's public key to decipher the message.
- RSA algorithm is generally used for both cryptography and digital signatures.

SECURITY IN DEPENDABLE MCS …

- Message authentication – origin integrity may be checked through digital signatures and data integrity may be checked through generating a "footprint" of the message (hash code using some one-way hash function) and appending it along with the original. Receiver recalculates the "footprint" of the received original message and checks it with the received "footprint".

SECURITY IN DEPENDABLE MCS …

- Authentication
  - is the act of establishing or confirming something (or someone) as *authentic*, that is, claims made by or about the subject are true

SECURITY IN DEPENDABLE MCS …

- Since MSSs are considered to be static, we assume (as of now) they do not pose any security threat to the system
- But MHs pose threat to the system since they can come and go at their own will !! So proper identification of MHs is required.
  - Hence the need to authenticate MHs.
  - Also, somebody must be powerful enough to authenticate the MHs.

SECURITY IN DEPENDABLE MCS …

- Authorization involves verifying that an authenticated subject has permission to perform certain operations or access specific resources. Authentication, therefore, must precede authorization.
- Thus, only authenticated MHs become authorized processes in the system.
- Generally MHs authenticate themselves with their respective HSs

SECURITY IN DEPENDABLE MCS …

- Authentication may be done using one or more of many techniques. May be: password verification, certificate possession, possession of token etc.
- Password – common practice since it is easy to use and implement. An MH needs to register itself (along with information and password/ passcode) with an MSS that is supposed to be its HS. During subsequent entries in a network the password may be checked with MSSs to find an appropriate match

SECURITY IN DEPENDABLE MCS …

- Drawback – since MH may join (after disconnection) a network in a foreign cell, passwords need to be checked with all. This generates unnecessary checking. If MH is able to retain information about its original HS, then unnecessary checking may be avoided. Otherwise when MH roams in the network without any disconnection, MIP retains necessary information.

- Certificate – MHs may obtain a certificate from an appropriate certificate issuing authority and show that during authentication.

SECURITY IN DEPENDABLE MCS …

- Drawback – any MSS may act as certificate issuing authority, which should be known to other MSSs. MHs joining network for the first time should get proper advertisements such that they get hold of certificates.

- Token – like possessing a certificate, however, with lesser formalities. MSS with whom MH is registered is issued a token
  - Drawback – MH has to retain the token

CONCLUDING REMARKS

Remarks

- Till now we have not come across a mobile computing system that can be termed as a purely dependable MCS.
- This tutorial therefore aims at providing an overall picture of what a truly dependable MCS would mean, look like and behave.
- Not all aspects of the system is fully designed (as of now), but since analysis is done and a few remains to be designed and tested, a dependable MCS is already on its way.

REFERENCES AND BIBLIOGRAPHY

REFERENCES:

[1] R. C. Gass and B. Gupta, “An Efficient Checkpointing Scheme for Mobile Computing Systems”, Computer Science Department of Southern Illinois University

[2] K. S. Byun and J.H. Kim, “ Two-Tier Coordinated Checkpointg Algorithm For Cellular Networks”, ICCIS 2001

[3] S. Neogy, “A Checkpointing Protocol for a Minimum set of Processes in Mobile Computing Systems”, IASTED International Conference on Parallel and Distributed Computing Systems (IASTED PDCS 2004)

REFERENCES:

[4] C. Chowdhury, S. Neogy,"A Consistent Checkpointing-Recovery Protocol for Minimal number of Nodes in Mobile Computing System", in the International Conference on High Performance Computing (HiPC 2007), pages-599-611, LNCS, 2007.

[5] S. Neogy, WTMR – A new Fault Tolerance Technique for Wireless and Mobile Computing Systems, *Proceedings of the 11th International Workshop on Future Trends of Distributed Computing Systems* (FTDCS 2007), ISBN: 0-7695-2810-4, USA, 2007, pp. 130 – 137

BIBLIOGRAPHY

1. C. Perkins , IP Mobility Support, ed., IETF RFC 2002(1996).
2. R. C. Gass and B. Gupta, An Efficient Checkpointing Scheme for Mobile Computing Systems, *Proceedings of ISCA 13th international conference on Computer applications in industry and engineering, Honolulu, November 2000, pp. 323–328.*
3. A. Acharya and B.R. Badrinath, Checkpointing Distributed Applications on Mobile Computers, Proc. Third Int'l Conf. Parallel and Distributed Information Systems, Sept. 1994.
4. B. R. Badrinath, A. Acharya and Tomas Imielinski, Designing Distributed Algorithms for Mobile Computing Networks, Computer Communications, Vol. 19, No. 4, 1996.
5. B. Yao, K. Ssu and W. K. Fuchs, Message Logging in Mobile Computing, Proc. of the 29th Int'l Symp. on Fault Tolerant Computing Systems,1999.
6. C.M. Lin and C. Dow, Efficient Checkpoint-Based Failure Recovery Techniques in Mobile Computing Systems, Journal of Information Science and Engineering, 17, 2001, 549-573.

BIBLIOGRAPHY ..

7. D. K. Pradhan, P. Krishna, N. H. Vaidya, Recoverable Mobile Environment: Design and Tradeoff Analysis,Proc. of The 26th Int'l Symp. on Fault Tolerant Computing Systems,1996.
8. D. L. Russell, State Restoration in Systems of Communicating Processes, IEEE Transactions on Software Engineering, SE-6(2):183-194, Mar. 1980.
9. G. Cao and M. Singhal, On Coordinated Checkpointing in Distributed Systems, IEEE Trans on Parallel and Distributed Systems, Vol. 9, No. 12, 1998.
10. G. Cao and M. Singhal, Mutable Checkpoints: A New Checkpointing Approach for Mobile Computing Systems, IEEE Transactions on Parallel and Distributed Systems, Vol.12, No.2, 2001.

BIBLIOGRAPHY …

11. L. Lamport, Time, Clocks and the Ordering of Events in a Distributed System, Communications of the ACM, Vol. 21, No. 7, pp. 558-565, July 1978.
12. P. Krishna, N. H Vaidya and D. K Pradhan, Recovery in Distributed Mobile Environments, IEEE Workshop on Advances in Parallel and Distributed Systems, 1993.
13. R. Prakash and M. Singhal, Low-Cost Checkpointing and Failure Recovery in Mobile Computing Systems, IEEE Transactions on Parallel and Distributed Systems, Vol.7, No. 10,1996.
14. T. Park, N.Woo and H.Y. Yeom, An Efficient Recovery Scheme for Mobile Computing Environment, Proc. of ICPADS 2001, 2001, 53-60.
15. Y. Morita and H. Higaki , Checkpoint-Recovery for Mobile Computing Systems, Proc. of Distributed Computing Systems Workshop, 2001, 479-484.

BIBLIOGRAPHY …

16. L. Alvisi,B. Hoppe and K. Marzullo, Nonblocking and Orphan-free Message Logging Protocols, Proc.of The Twenty-Third International Symposium on Fault-Tolerant Computing, 1993. FTCS-23. Digest of Papers, 145-154.

17. J. Ahn,S. Min and C. Hwang, A Causal Message Logging Protocol for Mobile Nodes in Mobile Computing System, Future Generation Computer Systems, Volume 20, Issue4, 2004, Pages: 663 - 686.

18. J.L. Cook and J.E. Ramirez-Marquez, "Mobility and reliability modeling for a mobile ad-hoc network", IIE Transactions, 1545-8830, Vol. 41, Issue 1, pp. 23 – 31, 2009.

19. X. Chen and M. R. Lyu, "Reliability analysis for various communication schemes in wireless CORBA", IEEE Transactions on Reliability. Vol. 54, No 2. pp. 232-242, 2005.

BIBLIOGRAPHY …

20. A. P. Snow, U. Varshney, and A. D. Malloy, “Reliability and survivability of wireless and mobile networks”, IEEE Computer, Vol. 33, No. 7, pp. 49–55, 2000.
21. Capt. Ş. Yaşar, “Algorithm design for reliability analysis in mobile communication networks”, in the Journal of Aeronautics and Space Technologies, Vol. 3, No. 1, pp. 29-39, 2007.
22. C. Bettstetter and C. Wagner, “The spatial node distribution of the Random Waypoint Mobility Model”, in the Proceedings of German Workshop on Mobile Ad Hoc Networks, 2002.
23. C. Bettstetter, “Smooth is better than sharp: a random mobility model for simulation of wireless networks”, in the Proceedings of the Fourth ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems, pp. 19-25, 2001.

BIBLIOGRAPHY …

24. H. AboElFotoh and C. J. Colbourn, “Computing 2-terminal reliability for radio-broadcast networks”, IEEE Transactions on Reliability, Vol. 38, No. 5, December, pp. 538-555, 1989.

25. Spirent Communications, “Critical elements of testing wireless mobile IP”, White paper, February, http:// spcprev.spirentcom.com/documents/1360.pdf , 2004.

26. C. Chowdhury and S. Neogy, “Checkpointing using mobile agents for mobile computing system”, in the International Journal of Recent Trends in Engineering, Issue. 1, Vol. 1, pp. 26-29, May 2009.

27. K. A. Shuaib, “A performance evaluation study of WIMAX using Qualnet”, in the Proceedings of the World Congress on Engineering 2009, Vol. I, WCE 2009.

BIBLIOGRAPHY …

28. Lyons, R.E. & Vanderkulk, W., The Use of Triple Modular Redundancyto Improve Computer Reliability, IBM Journal, 200-209, April 1962.
29. S. Neogy, P. K. Das, A Reliable Time-out-free Fault-Tolerant Architecture without Dynamic Reconfiguration, Proceedings of the 28th Annual Convention and Exhibition of IEEE India Council (IEEE ACE 2002), Kolkata, India, pp.200 – 203.
30. G. Cao and M. Singhal, "On Coordinated Checkpointing in Distributed Systems", IEEE Transactions On Parallel And Distributed Systems, Vol. 9, No. 12, 1998. Hou,
31. How, C.J.,&Shon,K.G., Incorporation of Optimal Time Outs Into Distributed Real-Time Load Sharing, IEEE Trans. on Computers, Vol.43, No.5, pp. 528-547, May 1994.

BIBLIOGRAPHY …

32. D. Manivannan and M. Singhal, “Quasi-Synchronous Checkpointing: Models, Characterization and Classification”, IEEE Transactions On Parallel And Distributed Systems, Vol. 10, No. 7, pp. 703-713, 1999.

33. M. Chandy and L. Lamport. “Distributed Snapshot: Determining global states of distributed systems”, ACM Transactions on Computer Systems, Vol. 3, No. 1, pp 63-75,1985.

34. R. Prakash and M. Singhal, “Low-Cost Checkpointing and failure Recovery in Mobile Computing Systems”, IEEE Transactions On Parallel And Distributed Systems, Vol. 7, No. 10, 1996.

BIBLIOGRAPHY …

35. G. Cao and M. Singhal, "Mutable Checkpoints: A New Checkpointing Approach For Mobile Computing Systems", IEEE Transactions On Parallel and Distributed Systems, Vol.12, No. 2, 2001.

36. T. Park, N. Woo and H.Y. Yeom, "An Efficient Recovery Scheme for Mobile Computing Environment", ICPADS 2001, pp. 53-60, 2001.

And many others ……..

Thank You