SYN

Security Analysis of Serendipity

Sahil Chadha
Rituraj Satpute
Manish Choudhary
Girish Dhoble

Mobile Adhoc Networks

- Collection of autonomous nodes or terminals
- Communicate with each other by forming a multi-hop radio network
- Maintaining connectivity in a decentralized manner
- Bandwidth constrained wireless links
- Frequent Host movement
- Frequent Topology change

Issues with Ad Hoc

- Lack of a centralized entity
- Routing and Mobility Management
- Low Channel Bandwidth
- Hidden/Exposed station problem
- Network topology changes frequently and unpredictably

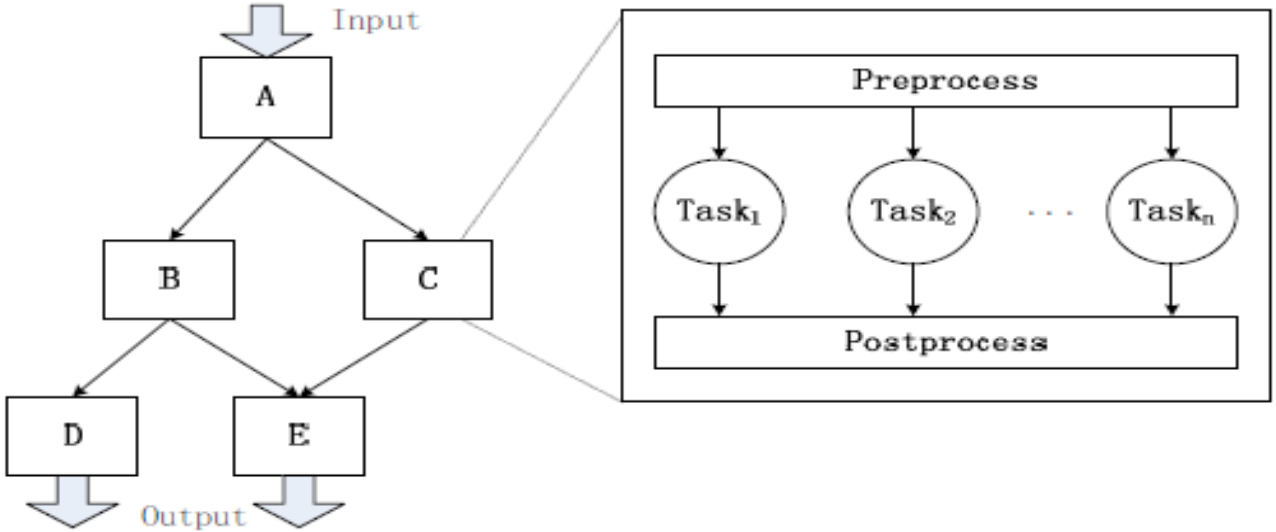
Issues with Ad Hoc

- Affected by higher loss rates, and can experience higher delays and jitter than fixed networks due to the wireless transmission
- Energy constrained nodes
- Physical security is limited due to the wireless transmission
- Lack of symmetric links

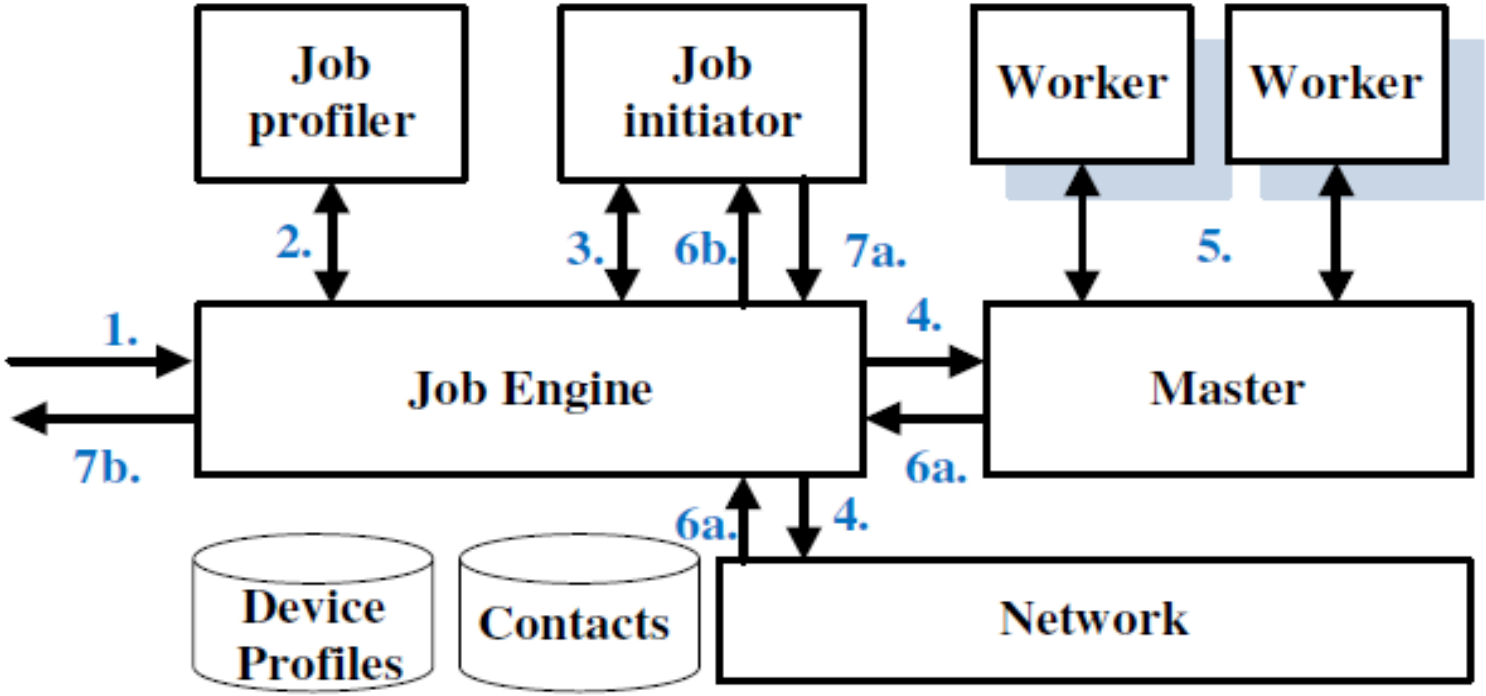
Serendipity

- Serendipity is a system that enables utilization of remote computational resources in a mobile ad hoc network.
- Job initiator divides job into multiple tasks.
- Tasks disseminated through the network using Water Filling algorithm.

Serendipity: Job Model (PNP Block)



Serendipity: High Level Architecture



Serendipity Continued

- Tasks can be pieces of code that execute on remote nodes
- Current version of Serendipity employs no task checking mechanism
- This makes nodes in Serendipity susceptible to a range of attacks
- These attacks could be:
  - General attacks on MANETs.
  - Attacks specific to Serendipity (exploiting the specific workings of Serendipity to mount an attack.)

Existing Attacks

- Attack on Confidentiality
  - No data encryption used
  - Any malicious node can intercept the communication between the sender and the receiver
  - Military scenario: the result would be catastrophic

- Attack on Authenticity and Integrity
  - No authentication scheme in current system
  - Attacker can impersonate as authentic initiator or add malicious content on the fly

Existing Attacks

- Attack on Availability
  - Malicious node can launch DoS attack
    - By continuous assignment of tasks to the nodes making them unavailable for other job initiators
    - Introducing malicious nodes in the network to take the jobs and never returning the results

Existing Attacks

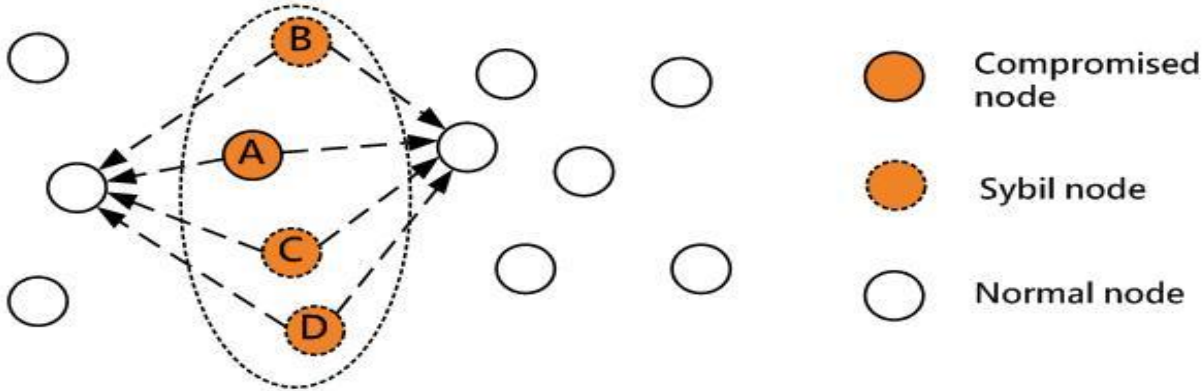
- Paper suggests the use of Reputation Based Systems
- But,

  Attacks on Reputation based systems

  - Whitewashing attack
  - Sybil attack
  - Impersonation and reputation theft
  - Denial of reputation
  - Attack on underlying network
  - Trust topology threats

Existing Attacks

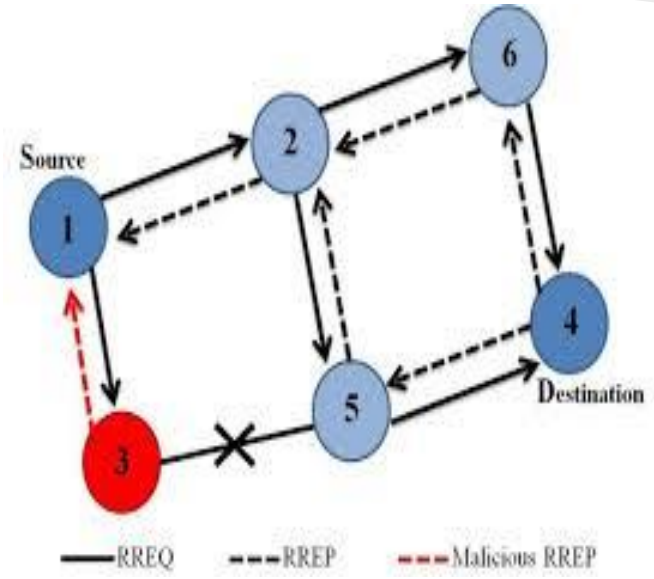
- Sybil Attack
  - Counterfeiting multiple identities with malicious intent
  - Results in submission of all the tasks to the attacker
  - This may hamper the computation and can also result in denial of service attack



Existing Attacks

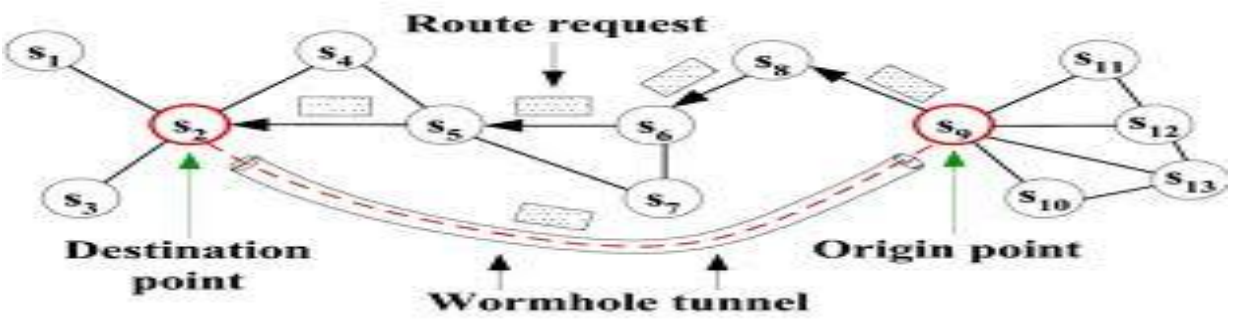
- Black Hole Attack
  - Attacker advertises itself as best route to destination

- May Result in
  - DOS
  - Man in the Middle attack



Existing Attacks

- Wormhole Attack
  - Attacker forward packets through a high quality out-of-band link and replays those packets at another location in the network
  - Affects routing and other decisions
  - May lead to packet drop, network disruption and even DoS



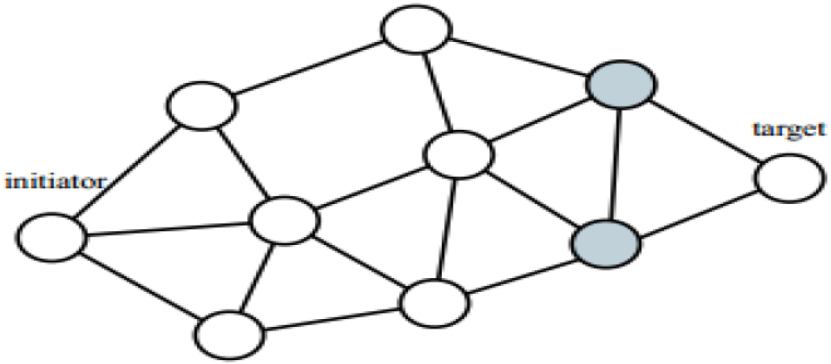
Existing Attacks

- Flooding Attack
  - Kind of denial-of-service attack
  - Malicious initiator can flood many fake tasks exhausting the computing power and resources of the node

- Grey Hole Attack
  - Extension of black hole attack but harder to detect
  - Behaviours and activities of malicious node are unpredictable
  - Attacker node behaves maliciously only for some nodes or for some period of time

Existing Attacks

- Rushing attack
  - Initiator node starts a route discovery for target node
  - If RREQ forwarded by attacker is first to reach neighbour of target then Route Discovery include a hop through attacker
  - All other RREQ will be discarded



Fixes for Existing Attacks

- Ensuring Confidentiality, Authenticity and Integrity
  - Use standard IND-CCA secure cryptographic schemes like Encrypt-then-MAC
    - IND-CPA Secure CBC$ for ensuring Confidentiality
    - SUF-CMA Secure HMAC for ensuring Authenticity and Integrity

- Ensuring Availability
  - Adding redundancy
  - Our Proposed solution

Fixes for Existing Attacks

- Preventing Sybil Attack
  - Centralized or Semi-Centralized Trusted Identity Management Authorities
  - Specific System Features Based Techniques, Sybil Guard etc.

- Preventing Black Hole Attack
  - Wait and Check replies from all the neighboring nodes to find a safe route

Fixes for Existing Attacks

- Preventing Wormhole Attack
  - Each node monitor behaviour of neighbouring nodes
  - Source send RREQ if it does not receive RREP within time add route to wormhole list

- Preventing Flooding Attack
  - Flooding Attack Prevention (FAP)- Use Trust Function

Fixes for Existing Attacks

- Preventing Rushing Attack
  - Secure neighbour detection (Two nodes detect each other as neighbors)
    - If they can communicate
    - If they are within maximum transmission range
  - Secure Route Delegation
    - Verify secure neighbour detection protocol executed successfully
  - Randomized Message Forwarding
    - Receiving node collects a number of RREQ
    - Forward a random RREQ.

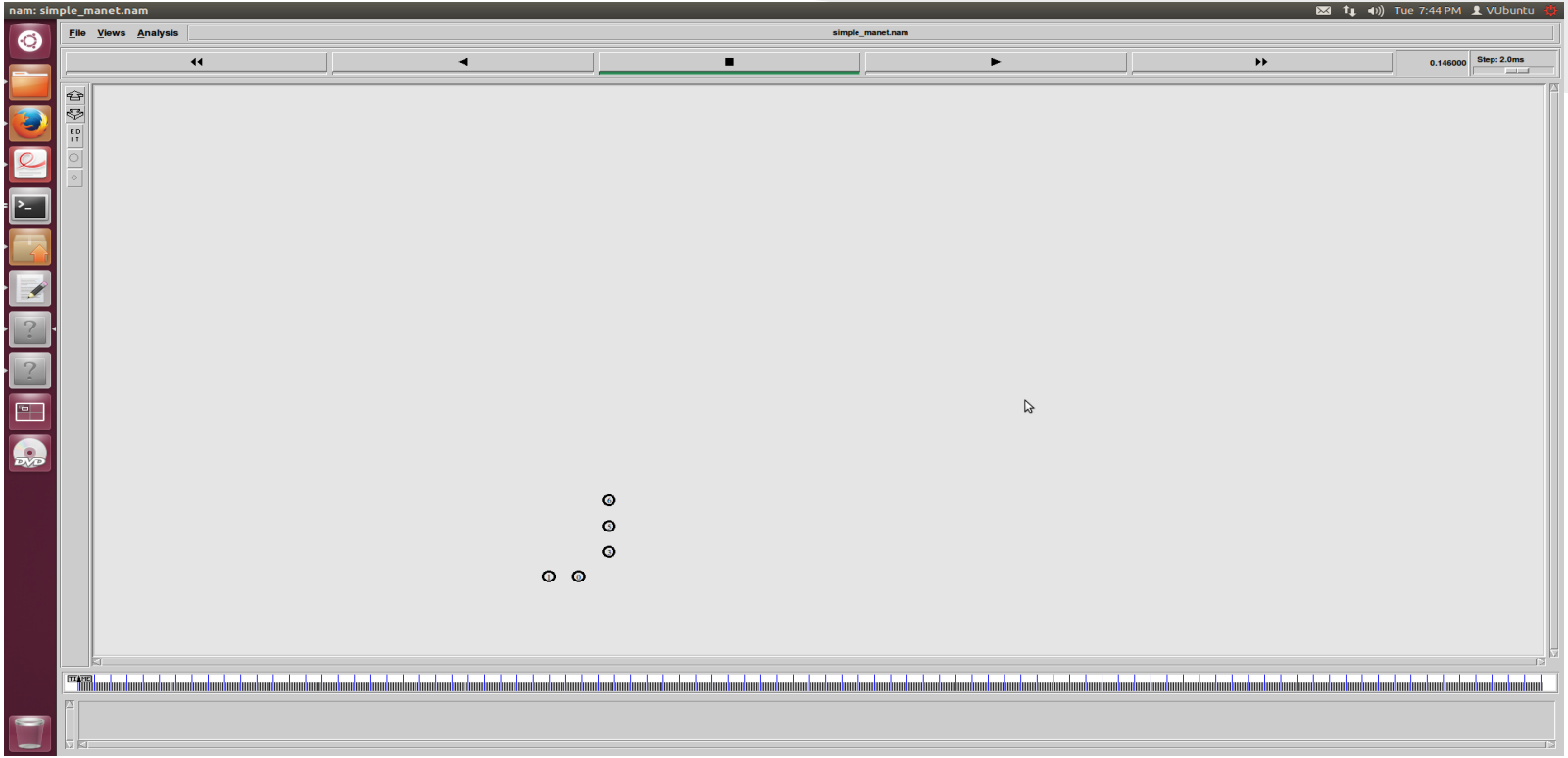
Attacks Specific to Serendipity

- Attacks involving malicious nodes:
  - Monopolize Serendipity by advertising itself as an all-powerful node.
  - Mount DOS attack on the job initiator (extension of above attack).
  - Mount DOS attack on Serendipity by refusing to return job results.
- Attacks involving malicious job initiator:
  - Malicious Code Injection
    - Retrieve Personal Information
    - Bombard with advertisement
    - Installing Backdoor to perform attacker-specific jobs
    - ROP attacks.
  - DOS attack mounted on the nodes by the job initiator.
- Man in the middle attacks between job initiator and nodes of the network.

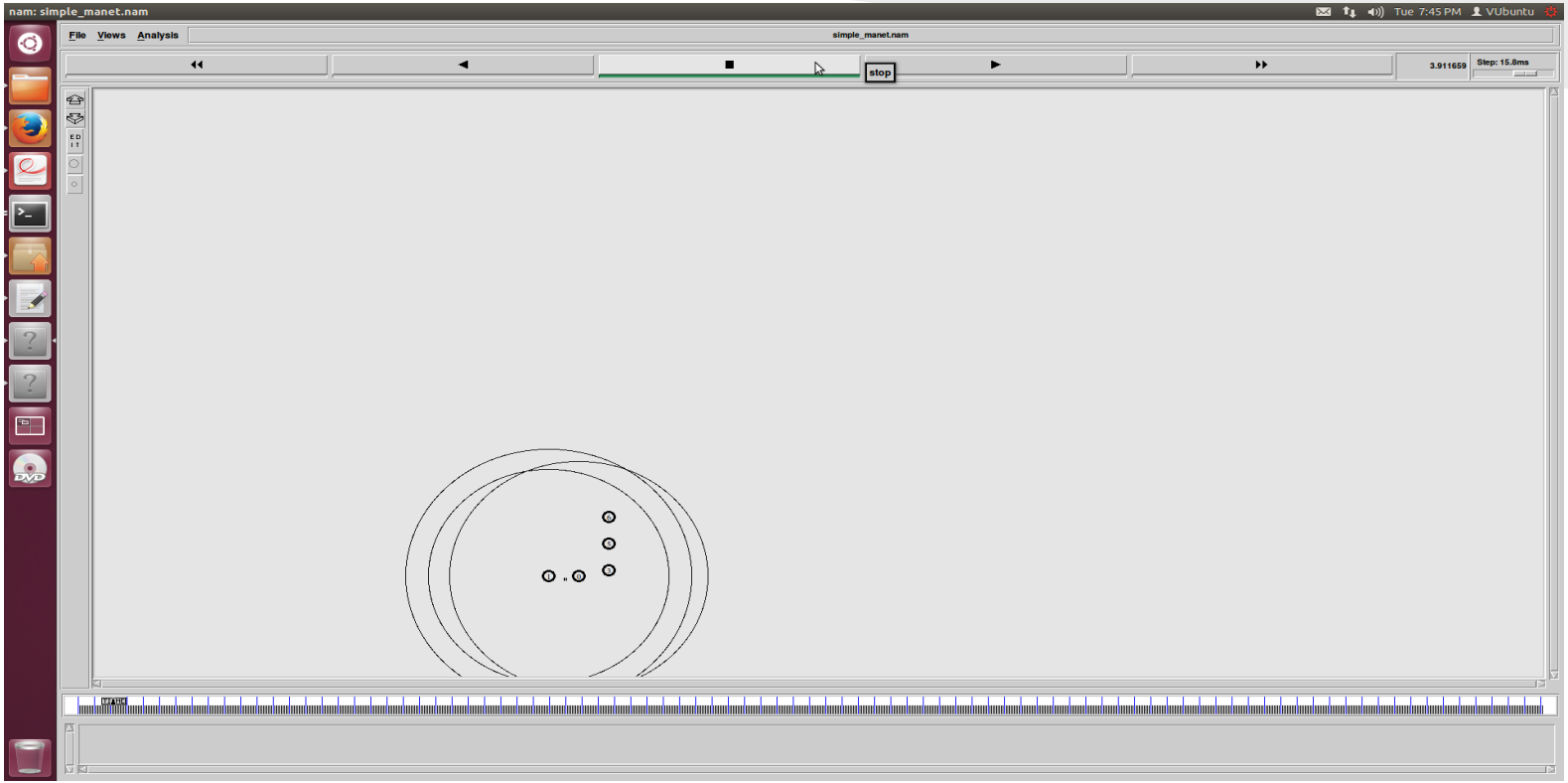
Demo of DOS Attack on Serendipity

- Simulated a DOS attack on a job initiator by a malicious node in NS 2.
- Malicious node constantly advertised itself with high availability of resources in the form of battery, CPU & lesser propagation delay.
- Job initiator ends up forwarding all tasks to the malicious node due to the greedy algorithm.

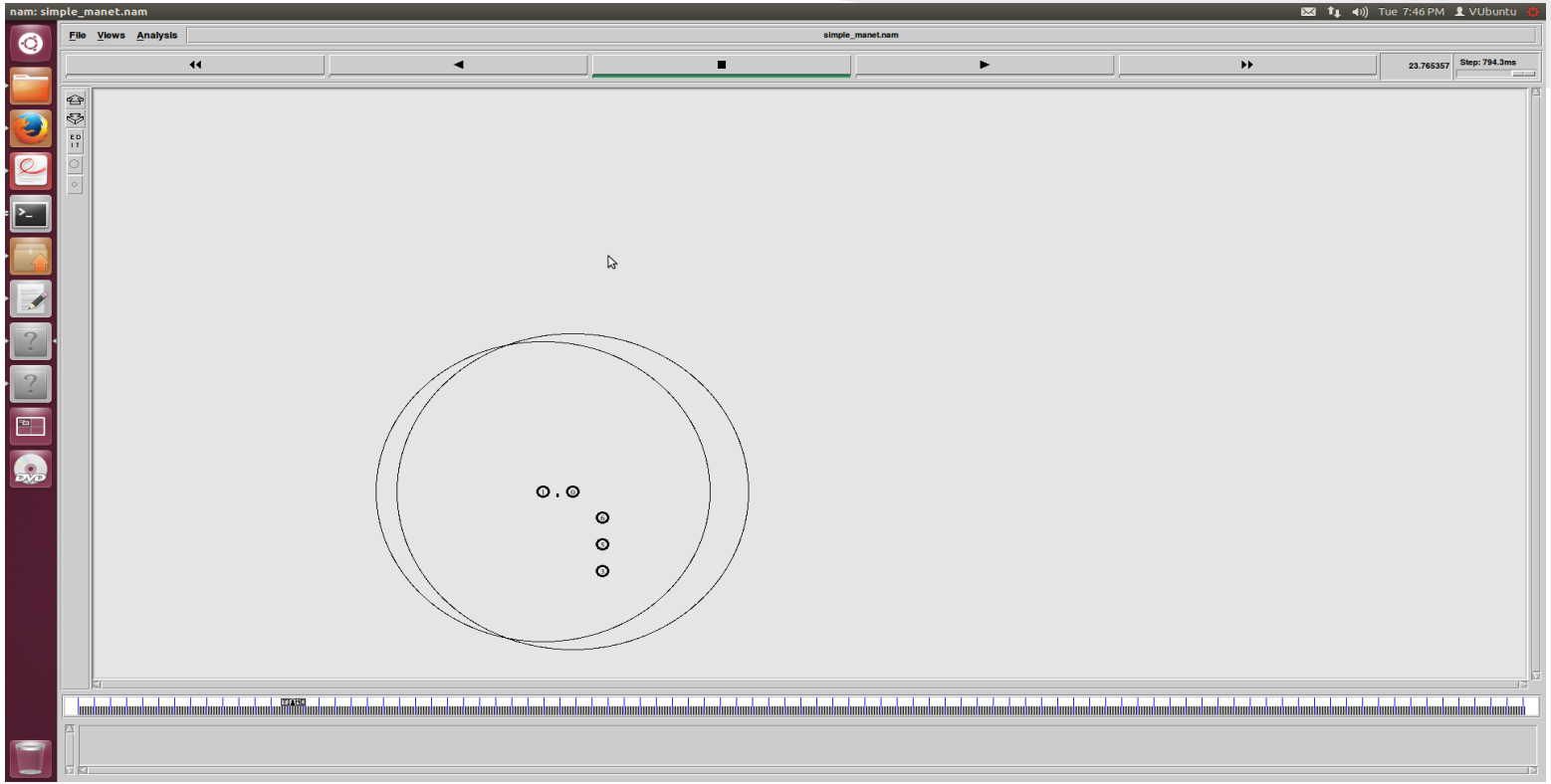
Initial Status: Job initiator tries to disseminate tasks based on node profiles.



Intermediate status: malicious node starts monopolizing the network



Final Status: Malicious node successfully monopolizes the network



Secure Serendipity - Building the Trust (Phase I)

- We present a solution ‘Secure Serendipity’, which is similar to route reflectors concept, to deal with the problem of malicious nodes by making some modifications to Serendipity:
  - Run Serendipity for some time t_o
  - During t_0, Job Initiator and participating worker nodes maintain trust matrix for each other
  - After t_o, all nodes share their matrix with each other.
  - m+k nodes with highest trust factor are selected out of which m random nodes are picked as trusted nodes.
  - Trusted nodes are responsible for the security of the network and can get their jobs done in return by passing jobs to other trusted nodes.
- Note that all the communication between the nodes happens in encrypted form.

Secure Serendipity - Job Execution (Phase II)

- Job initiator divides the task based on the predictive contacts of nearest trusted node
- Job initiator forwards the jobs to be allocated to the nearest trusted node that runs it partially in a sandbox and allocates if not found malicious.
- All worker nodes return results to the trusted nodes which ensure the authenticity and that the results are not malicious.
- Trusted nodes maintain hashes and results of the job verification to improve the efficiency.
- Trusted nodes ensure that jobs are not allocated to same node by maintaining a database and threshold.
- Trusted nodes can share databases and predictive contacts with each other
- Trusted node can forward job allocation task to other trusted nodes which would verify the job for malicious content

Secure Serendipity - Rebasing the Trust (Phase III)

- Trusted nodes can blacklist malicious nodes which would not be considered as a part of network in future
- All trusted and untrusted nodes maintain trust matrix for each other
- After fixed time period, trusted nodes are selected based on the updated trust matrix.
- Nodes can’t be selected as trusted thrice consecutively.
- New nodes without trust matrix can not be selected as trusted nodes.

Secure Serendipity - Limitations

- Degrades performance of Serendipity
- Lesser number of worker nodes
- Requires encrypted communication channel
- Tradeoff between security and efficiency
- Predictive contacts assumed

FIN