**Software Requirement Specification**

**Project Overview:**

Wireless Sensor Networks (WSN) have been playing an important part in different remote event monitoring applications, particularly in hazardous regions and unfriendly situations. Communication in wireless sensor network can be achieved through the intermediate nodes which in turn forward the got information to another node until packet reach their desired destination. This can save energy and increase the lifetime of their battery.

The software corruption, hardware failure and non-favorable operating environment among different nodes in wireless sensor network can reduce the nodes functionality and affect the entire wireless sensor network operations. Node experiencing such a problem is called an infected node. Due to infected node packets cannot be forwarded to destination these packets become lost or stuck in the infected areas. This problem will increase the packet loss rate and energy consumption. The corrupted data in the packets results in false analyses and wrong decision making at end system. Hence a timely detection of the infected nodes and determine the alternative route to divert the traffic from infected area.

**Objectives :**

* To design the method that can get the stuck packets out of the infected regions.
* To design the method that can by-pass infected areas and reroute the incoming packets to uninfected regions.
* To minimize the effects of infected nodes because effect of trapping important packets inside an infected region could be massive.

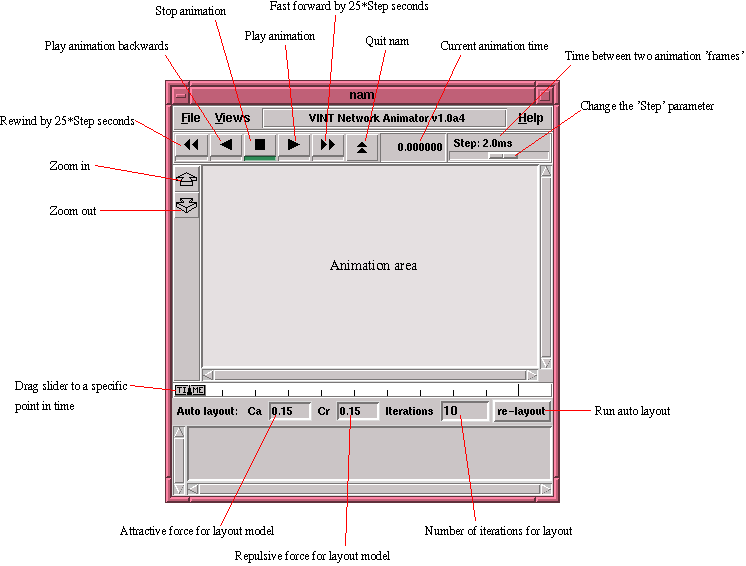
**External Interface Requirements:**

* **User Interfaces :** NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the

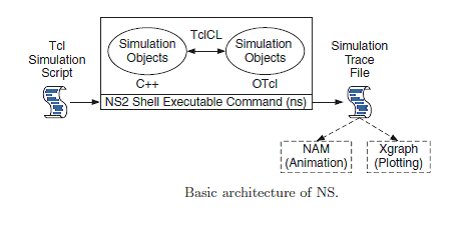
objects as well as scheduling discrete events. The C++ and the OTcl are linked together using TclCL.

NS2 uses OTcl to create and configure a network, and uses C++ to run simulation. All C++ codes need to be compiled and linked to create an executable file.

OTcl is used for configuration, setup, or one time simulation, or to run simulation existing NS2 module **C++** is used when we are dealing with a packet, or when we need to modify existing NS2 modules.



**GUI of NS2**



**b) Hardware Interfaces:**  64 bit processor, Intel core i5  
 1GB RAM

500GB Hard Disk

**c) Software Interfaces :** Ubuntu  
 NS-2 (network simulator 2)  
 C++

d) **Communication Interfaces:** The software communicates internally with the main network module and other modules.

**Functional Requirements:**

**Functional Requirement 1.1**

**Network Deployment :** First define the Network configuration parameters i.e., specify the number of nodes ,initial energy, MAC, propagation, Receiver power, sleep power, transmission power, Channel Type, Propagation or TwoRayGround i.e., radio-propagation model, network interface (Phy / Wireless Phy), MAC type(Mac/802\_11),interface queue type(CMUPriQueue), link layer type, antenna model (Antenna /OmniAntenna), maxpacket in fig, number of mobile nodes, X axis distance, Y axis distance Initial Energy, Initial energy in Joules. Then deploy all the nodes into the network with some moving velocity. The network stack for a mobile node consists of a link layer (LL), an ARP module connected to LL, an interface priority queue (IFq), a mac layer (MAC), a network interface (netIF), all connected to the channel. These network components are created and plumbed together in OTcl.

**Functional Requirement 1.2**

**Node Discovery Phase:** After the nodes are created then assign the node positions with the set destination of xvalue, yvalue and hdistance. Configuration of the nodes in the network by specifying values to network   
configuration parameters.

**Functional Requirement 1.3**

**Identification of infected nodes :**Detects the occurrence of infected nodes adapting a Fuzzy Data Clustering approach to identify anomalies based on the received data signals. The fuzzy clustering method is chosen as it provides an unsupervised and modular method for evaluating anomalous data over the different sensor nodes. A data-centric point of view as with fuzzy clustering is suitable when evaluating whether a node is infected or not, whether it is through a hardware malfunction, malware attack or software corruption.

**Functional Requirement 1.4**

**By-passed routing :** The information about infected nodes is then directly used for traffic diversion in the proposed BPR technique. The novelty of the BPR approach relies on the introduction of the simultaneous twin rolling balls technique that detects the next 1-hop neighbor faster than the existing GAR approach. Using this approach, the first node that hits any ball in any direction and is uninfected is assigned as the next hop. A further different way of getting the stuck packets out of infected regions is using the by-passing infected area the normal routing is started. Finally shortest path is identified to reach the destination.

**Functional Requirement 1.1**

**Performance Analysis :** In this mathematical operations are performed based on the above all the operations then results will be stored into the xgraphs.

**Software System Attributes:**

1. **Reliability:** We make use of ‘Fuzzy detection algorithm’ for identifying the infected area and ‘Twin rolling ball technique’ for by-passing the infected area. These are better and more reliable than compared to the  
   existing system.
2. **Availability:** The system is available to detect the attacks and re-route the packets as long as it is working without any failures. If a failure occurs then the system is restarted.
3. **Security:** There are no security issues with this system as there is no confidential content to be handled.
4. **Portability:** The software can be used on any operating system as there no OS specific code as long as that OS is supports NS2.
5. **Maintainability:** The Code is modularized and can be modified easily. It is written in such a way that new features can be implemented easily.
6. **Performance:** Performance of the software will be better than the existing system as new approach is used for solving the problem. The algorithms are optimized to provide the best performance.