Title of Experiment: Introduction to Wireless Sensor Network and its Simulation Tools

Objective:

An NS2 script written in Tcl for creating a basic network topology and running a simple simulation. A simple point-to-point network with two nodes connected by a link.

Introduction:

Components/Software:

WSN component-

- Nodes
- Routers
- Gateway

Types of Simulation-

- Monte Carlo
- Trace-Driven
- Discrete-Event

Simulation tools-

- NS2(Network Simulator-2)
- OMNet++
- J-Sim(Java Sim)
- Avrora

Theory:

Wireless Sensor network- WSN is an ad hoc based wireless network that is deployed in many wireless sensors that is used to monitor any system, physical or environment conditions. WSNs consist of nodes, routers, and a gateway.

Components Of WSNs-

Nodes- A node is an electronic device that is attached to a network and can create, receiving or transmitting information over a communication channel. There are two types of nodes:

1. sensor nodes: Also known as mote consist of an individual node from a sensor network that can perform a desired action such as gathering, processing or communicating information with other connected nodes in a network.
2. **actuator nodes:** An actuator node is a system, which converts electrical control signals to physical actions, and constitutes the mechanism by which an agent acts upon the physical environment.

Routers- These are used to extend communication range or circumvent an obstacle.

Gateway- It is the device that allows the management of the network and aggregates the information received from the nodes to send real time or near real time data to a user platform.

Simulator- It is used to develop and test protocols of WSNs in the initial stage of these designs. It is time saving, efficient and economical.

Use of Simulation in WSN- WSNs simulation is important for WSNs development. It allows users to isolate different factors by tuning configurable parameters. There are two key aspects in WSNs simulators these are:

- The correctness of simulation models.
- The suitability of a particular tool to implement the model.

Types of Simulation for WSN-

- Monte Carlo- A Monte Carlo simulation is a mathematical technique that simulates the range of possible outcomes for an uncertain event. The general steps to a Monte Carlo simulation are as follows:

1. Determine the mathematical model.
2. Choose the variable to simulate.
3. Run repeated simulation.
4. Visualize the results on histogram.

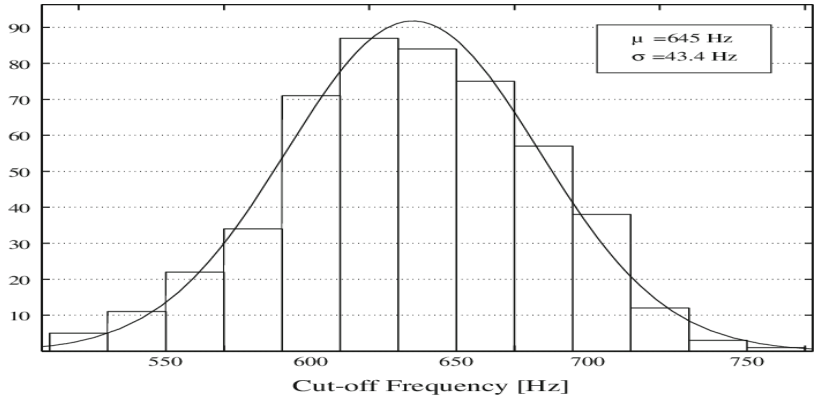

Fig 1: Histogram representation of Monte Carlo simulation

- Trace Driven- This kind of simulation is commonly used in real system. It provides more accurate workload; this detail information allows users to deeply study the simulation model. Usually input values in this simulation constantly unchanged.

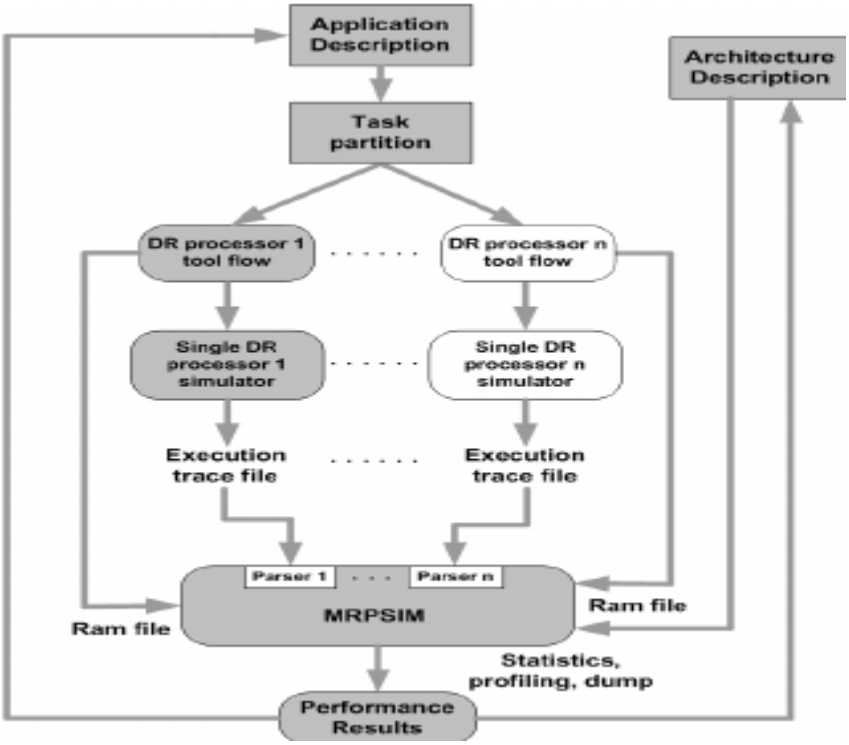

Fig 2: Schematic diagram of Trace Driven simulation

- Discrete Event- This kind of simulation widely used in WSNs, because it can easily simulate lots of jobs running on different sensor nodes. This simulation includes input routines, output routines, initial routines and trace routines. Debugger breakpoints are provided in discrete event simulation; thus, the users can check the code step by step without disrupting the program operation.

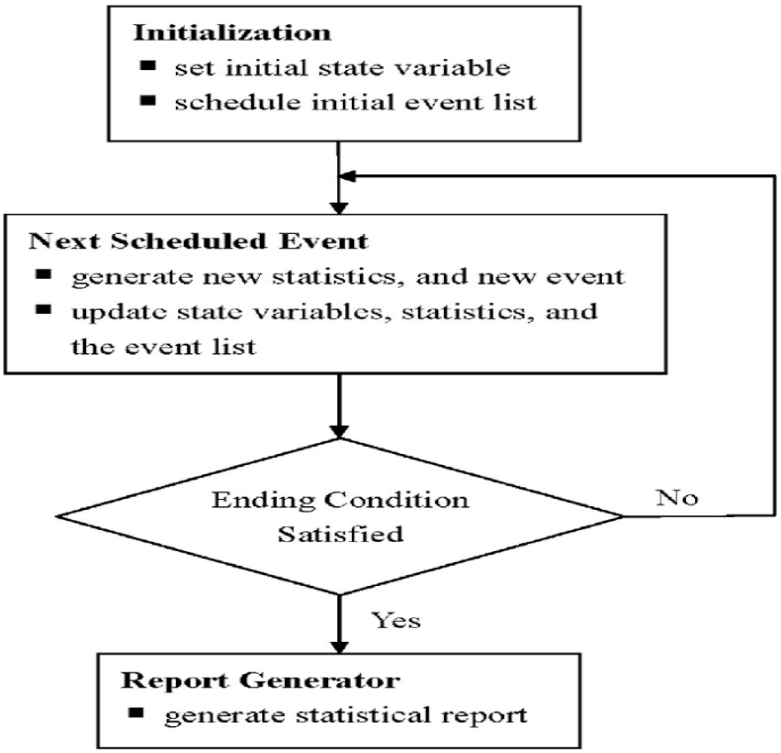

Fig 3: Schematic diagram of Discrete Event simulation

Simulation tools-

- NS2: NS-2 is the abbreviation of Network simulator version two, which first been developed by 1989 using as the REAL network simulator. Now, NS-2 is supported by Defence Advanced Research Projects Agency and National Science Foundation. NS-2 is a discrete event network simulator built in Object-Oriented extension of Tool Command Language and C++. People can run NS-2 simulator on Linux Operating Systems or on Cygwin, which is a Unix-like environment and command-line interface running on Windows. NS-2 is a popular non-specific network simulator can used in both wire and wireless area. This simulator is open source and provides online document.

Merits:

1. Support a considerable range of protocols in all layers.
2. The open-source model saves the cost of simulation, and online documents allow the users easily to modify and improve the codes.

Limitations:

1. Tool Command Language is somewhat difficulty to understand and write.
2. NS-2 is more complex and time-consuming than other simulators to model a desired job.
3. NS-2 provides a poor graphical support, no Graphical User Interface (GUI)
4. Due to the continuing changing the code base, the result may not be consistent, or contains bugs.

- OMNET++: *OMNET++ is a discrete event network simulator built in C++*. This simulator supports module programming model. Users can run OMNeT++ simulator on Linux Operating Systems, Unix-like system and Windows. OMNeT++ is a popular non-specific network simulator, which can be used in both wire and wireless area. Most of frameworks and simulation models in OMNeT++ are open sources.

Merits:

1. OMNeT++ provides a powerful GUI
2. OMNeT++ can support MAC protocols as well as some localized protocols in WSN.
3. OMNeT++ can simulate power consumption problems in WSNs.

Limitations:

1. The number of available protocols is not larger enough.
2. May have high probability report bugs.

- J-SIM(Java Sim): J-Sim is a discrete event network simulator built in Java. This simulator provides GUI library which facilities users to model or compile the Mathematical Modelling Language written to J-Sim models. J-Sim can simulate real - time processes.

Merits:

1. J-Sim has good reusability and interchangeability, which facilities easily simulation.
2. J-Sim contains large number of protocols; this simulator can also support data diffusions, routings and localization simulations in WSNs by detail models in the protocols of J-Sim.
3. J-Sim provides a GUI library, which can help users to trace and debug programs.

Limitations:

1. The execution time is much longer.
2. J-Sim was not originally designed to simulate WSNs.

- Avrora- Avrora is a simulator specifically designed for WSNs built in Java. Avrora can also simulate AVR based microcontroller MICA2 sensor nodes. It also supports energy consumption solution. It provides open sources and online documents.

Merits:

1. Avrora is an instruction level simulator.
2. Codes in Avrora run instruction by instruction, which provides faster speed and better scalability.
3. Can save much more execution time with similar accuracy.
4. It provides much flexibility.
5. Can simulate different programming code projects.

Limitations:

1. Avrora doesn't provide GUI.
2. Can not simulate network management algorithms
3. It doesn't provide network communication tools.

Methods and Procedures:

Flow chart: Table for - Simulation tools

NS-2	simulator	GUI No	Open source and online document	General simulator	Discrete event simulation	1.cannot simulate more than 100 nodes. 2 cannot simulate problems of the bandwidth or the power consumption in WSNs
OMNeT++	simulator	GUI Yes	Non-commercial and commercial licences	General simulator	Discrete event simulation	1.can support MAC protocols and some localized protocols in WSN 2. simulate power consumptions and channel controls 3. limited available protocols
J-Sim	simulator	GUI Yes	Open source and online document	General simulator	Discrete event simulation	1. can simulate large number of sensor nodes, around 500 2. can simulate radio channels and

						power consumptions 3. its execution time is much longer
Avrora	simulator	GUI Yes	Open source and online document	Specifically designed for WSNs	Discrete event simulation	1. can support thousands of nodes simulation 2.can save much more execution time

Methodology: NS2 Architecture -

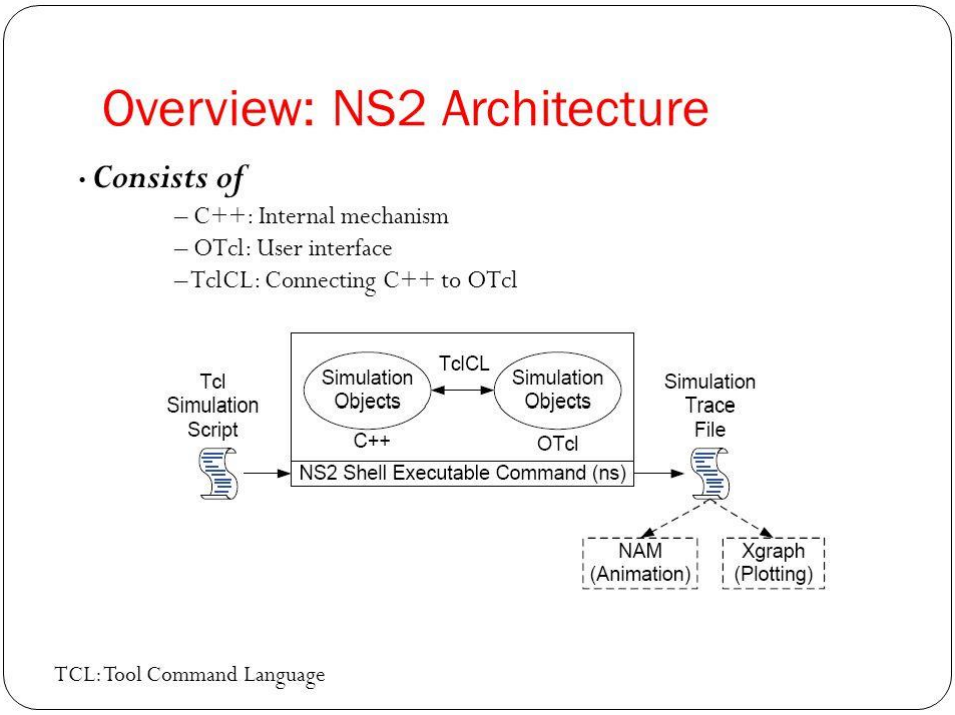

Fig : NS2 Architecture

How to Run NS2 Program in LINUX-

Step1: Source code to main file- Store the main source file with the filename without any whitespaces and stored in the folder name without any whitespaces.

Step2: Open the terminal and verify the installed package- Initially, next open the terminal by press Ctrl+Alt+T buttons or search from the installed software list.

Step3: Change the location- Change the location by using cd command.

Step4: Execute the main file- By using the NS2 to perform the process of execute / run the file by using command sudo./ns main.tcl.

Step5: Get the simulation: Select trace file in NS2. By using this process to execute the network simulator to get the simulation successfully.

Observation Data or Graphs:

Objective- An NS2 script written in Tcl for creating a basic network topology and running a simple simulation. A simple point-to-point network with two nodes connected by a link.

Algorithm-

1. Create a simulator object.
2. Define a setting option for wireless channel.
3. Create trace file and name file.
4. Setup topography object and nodes.
5. Provide initial location of mobile nodes.
6. Setup a UDP connection between nodes.
7. Printing the window size.

Calculation and Analysis:

Program-

```
# Create a new simulation object

set ns [new Simulator]

# Create two nodes

set node1 [$ns node]

set node2 [$ns node]

# Create a link between the two nodes

set link [$ns duplex-link $node1 $node2 1Mbps 10ms DropTail]
```

```
# Set up traffic generation

set udp [new Agent/UDP]

$ns attach-agent $node1 $udp

set cbr [new Application/Traffic/CBR]

$cbr attach-agent $udp

$cbr set interval_ 1.0

$cbr set random_ false

$cbr set packetSize_ 1000

# Connect traffic generator to the destination node

$ns connect $udp $link

# Set up tracing

set tracefile [open out.tr w]

$ns trace-all $tracefile

# Define simulation time

set simulation_time 5.0

# Run the simulation

$ns at $simulation_time "$ns namtrace-all-wireless $tracefile"

$ns at $simulation_time "finish"

# Start the simulation

$ns run
```

Fig: Result of NS2 Simulation

Results: The simulation runs for 5 seconds and traces the simulation events to the “out.tr” file. After the simulation, the “nametrace-all-wireless" function is called to generate a trace file that can be visualized using the Network Animator tool (nam).

Conclusion: The objective was to observe basic network behaviour and performance metrics using a simple traffic generation model. The simulation demonstrated the key steps involved in setting up and running a basic simulation in NS2. This experiment showcased the basic capabilities of NS2 for network simulation and provided insights into the mechanics of setting up and executing a simulation scenario. As we move forward, more advanced experiments can leverage NS2’s features to delve deeper into the complexities of networking protocols and behaviours.

Questions:

- What is wireless sensor network?
- What are types of Simulation tools?
- What is the range of wireless sensor nodes?
- What is the difference between NS2 and OMNeT++ simulation tool?
- How to run NS2 program on LINUX?