Research article

Journal of Smart Systems Research (JOINSSR) 6(1), 1-10, 2025

Recieved: 11-Oct -2024 Accepted: 8-May-2025

https://doi.org/10.58769/joinssr.1565413

SAKARYA UNIVERSITY OF APPLIED SCIENCES

Comparative Study of Wireless Network Protocols using Ns-2 Network Simulator

Ahmad SABAH [1], Ahmet ZENGİN [2]*

[1] Computer and Information Engineering, Sakarya University, Türkiye, ahmad.khoshnaw@gmail.com

[2] Computer and Information Engineering, Sakarya University, Türkiye, azengin@sakarya.edu.tr

ABSTRACT

Many factors affect the quality of wireless networks, but one of the most important is the movement between the nodes connected in one network. This movement leads to the presence of large packet loss as well as the impact on the energy of each node. Thus, it generally affects the network performance; several familiar routing protocols, such as AODV, DSR, and DSDV, are used to communicate between all nodes in the wireless network. This paper presents a comparison of performance and study of nodes in a single network based on Interactive Protocols (AODV) to study the metrics such as transmission rate, packet delivery rate, the average delay of Using the NS-2, determining the network, packet loss rate, and residual power at each node in the network simulator. The simulation results showed that increasing the traffic of the nodes increases the rate of packet loss and increases the remaining energy in the nodes in which there was a greater loss of packets, thus reducing the rate of packet delivery between the nodes, so the more the nodes have less movement, the higher the packet delivery rate and the lower the packet loss rate.

Keywords: AODV; Packet Loss; NS-2; Residual Energy; Performance

***Corresponding Author's e-mail:** azengin@sakarya.edu.tr

Cite as: Sabah, A. & Zengin, A. (2025). Comparative Study of Wireless Network Protocols using Ns-2 Network Simulator, *Journal of Smart Systems Research*, 6(1), 1-10. https://doi.org/10.58769/joinssr.1565413

1. Introduction

Unsuccessful packets break down networks, create bottlenecks, and interrupt throughput and bandwidth [1]. Packet loss could also be costly. You'll have to spend a lot of money on extra IT equipment and bandwidth to handle the lag if you don't do everything possible to reduce packet loss in your system [2].

Consider packets as automobiles speeding along a freeway across your network. Sometimes there can be traffic on the road, for example, during rush hour or after lunch when all the employees of a large company return to their desks. The incident occurred when the 4-lane highway narrowed to two lanes, and many vehicles tried to merge. Some automobiles were unable to merge and were late for their appointments.

Packet loss and highway traffic are both common occurrences. Networks are manageable and impregnable, and they have constraints in terms of space [3] [4].

Packets must wait for delivery when network traffic reaches its maximum capacity. Unfortunately, packets are the first to remain when the network tries to capture traffic, and the connection can only handle a certain amount of traffic [5]. Fortunately, most software nowadays can automatically resend lost packets or slow down transfer speeds to allow each packet a chance to make it through [6].

In this study, we developed a wireless model scenario using the mobile wireless model presented in NS-2, got the trace result and, calculated the performance using the trace file, then displayed the results using the Gnuplot tool [7] to examine each result of the performance plot chart. There are two parts to this section.

In the first subsection, we defined the GOD, received, and sent delivery ratios and drooped packets and outlined a model scenario for establishing and operating a wireless network simulation with six nodes. In the second subsection, we started to calculate the performance of our model by using the awk scripts. Then, we started plotting the results as a chart using the Gnuplot tool, which helped us plot the results as line points and scatter charts. We can use the awk scripts to calculate the throughput performance (average or instant), residual energy, and packet delivery ratio.

2. Methodology

2.1 Network Simulator-NS-2

NS-2 is a C++-based object-oriented simulator with a front-end OTcl interpreter [8]. The emulator supports the C++ class structure (the assembly hierarchy in this document) and the OTcl interpreter class hierarchy (the translation hierarchy). The two hierarchies are similar, and from the user's perspective, there is a direct relationship between groups in the translation hierarchy and groups in the compilation hierarchy [9].

The TclObject class is at the very top of the hierarchy. The interpreter allows users to construct new simulator objects created within the interpreter and closely mirrored by a similar object in the compiled hierarchy. The TclClass class defines methods that automatically create the interpreted class hierarchy. Methods defined in the class TclObject are used to mirror user-created objects.

Other hierarchies exist in C++ code and OTcl scripts, but they are not replicated in the same way that TclObject is.

Many different organizations contribute to the structure and components of NS-2, and their strongest asset is in their rules. The number of models and joints in NS-3 is significantly less than in NS-2. Since

NS-3 is still in development, community participation is needed to add to and improve the supporting code[10].

3. Creating a wireless scenario model

We constructed a wireless situation scenario with six nodes. The topology comprises six mobile nodes (node0, node1, node2, node3, node4 and node5).

In this model, mobile nodes move in an area whose boundaries are defined as 500m x 500m.

The nodes start at the same point on the boundary. They then approach each other in the first simulation run. Then they clump in the middle; node2 and node3 connect using UPD protocol, and other nodes connect using TCP in the middle of the simulation. The node0 sends the packets to node 5. Packets are exchanged between nodes when they come within earshot of each other.

As they move away, packets are dropped, and the dropped packets increase from time nine until time 27 in the simulation scenario.

The Wi-Fi simulation begins by writing a TCL script named wireless.TCL, The Link Layer (LL), the Interface Queue (IfQ), the MAC layer, wireless channel nodes, and different network additives make up a mobile node.

```
#initialize the variables
set val(chan)   Channel/WirelessChannel    ;#Channel Type
set val(prop)   Propagation/TwoRayGround   ;# radio-
propagation model
set val(netif)  Phy/WirelessPhy  ;# network interface type
WAVELAN DSSS 2.4GHz
set val(mac)    Mac/802_11                 ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue    ;# interface
queue type
set val(ll)     LL                         ;# link layer
type
set val(ant)    Antenna/OmniAntenna        ;# antenna
model
set val(ifqlen) 50                         ;# max packet
in ifq
set val(nn)     6                          ;# number of
mobilenodes
set val(rp)     AODV                       ;# routing
protocol
set val(x)      500                    ;# in metres
set val(y)      500                    ;# in metres
#Adhoc OnDemand Distance Vector


#GOD Creation - General Operations Director
create-god $val(nn)
set channel1 [new $val(chan)]
set channel2 [new $val(chan)]
set channel3 [new $val(chan)]

#configure the node
$ns node-config -adhocRouting $val(rp) \
        -llType $val(ll) \
        -macType $val(mac) \
        -ifqType $val(ifq) \
        -ifqLen $val(ifqlen) \
        -antType $val(ant) \
        -propType $val(prop) \
        -phyType $val(netif) \
        -energyModel "EnergyModel" \
        -initialEnergy 100.0 \
        -txPower 0.9 \
        -rxPower 0.5 \
        -idlePower 0.45 \
        -sleepPower 0.05 \
        -topoInstance $topo \
```

Figure 1: Developed NS-2 Script

At the start of a wireless simulation, we want to outline the kind for every of those network components. Additionally, we want to outline different parameters such as the sort of antenna, the radio-propagation model, the ad-hoc routing protocol utilized by cellular nodes, etc. We start our script wireless.tcl with a listing of those special parameters defined above; we can take the stairs under to symbolize the wireless network scenario:

- ✓ Initialize variables.
- ✓ Create a Simulator object.
- ✓ Create Tracing and animation files.
- ✓ Creation of topography.
- ✓ Creation of General Operations Director (GOD).
- ✓ Build the nodes.
- ✓ Create a link.
- ✓ Set the position of the nodes.
- ✓ Set mobility.
- ✓ Create TCP and UDP Traffic.
- ✓ At last, run the simulation.

Figure 1 is our network simulation code and briefly clarifies each step mentioned.

4. Result and performance

4.1 The Files of Results

After running the simulation model by the command (> ns wireless.tcl), we got two file results; the first one is wireless.tr, which performs the trace file that we used to calculate the performance of our network model, as shown in Figure 2 the trace file we get after running the model.

The other file we get after running the tcl file is the NAM, representing our model's visualization. It shows us the whole scenario of how the nodes connect and move together, and we can see the send and response between the nodes. We can also see the dropping packet, as shown in Figure 2. Screenshots for our model NAM show the dropped packet (see Figure 3) and the node connection (see Figure 4).

4.2 Calculating and plotting performance

In our simulation, we focused on determining network model performance by writing scripts to extract the relevant data from the trace file (wireless.tr) and then plotting the data using the Gnuplot tool to calculate the following:

1. Residual energy of a node.
2. Mean throughput.
3. Instantaneous throughput.
4. Packet delivery ratio.

4.3 Average throughput

We got the result in Figure 5 using the awk script to calculate the average throughput we prepared for this aim.

Our simulation starts at time one and ends at time 29. The total time for the simulation is 30 seconds. The total number of packets received is 3939, with an average throughput of 1.09 kbps (see Table 1).

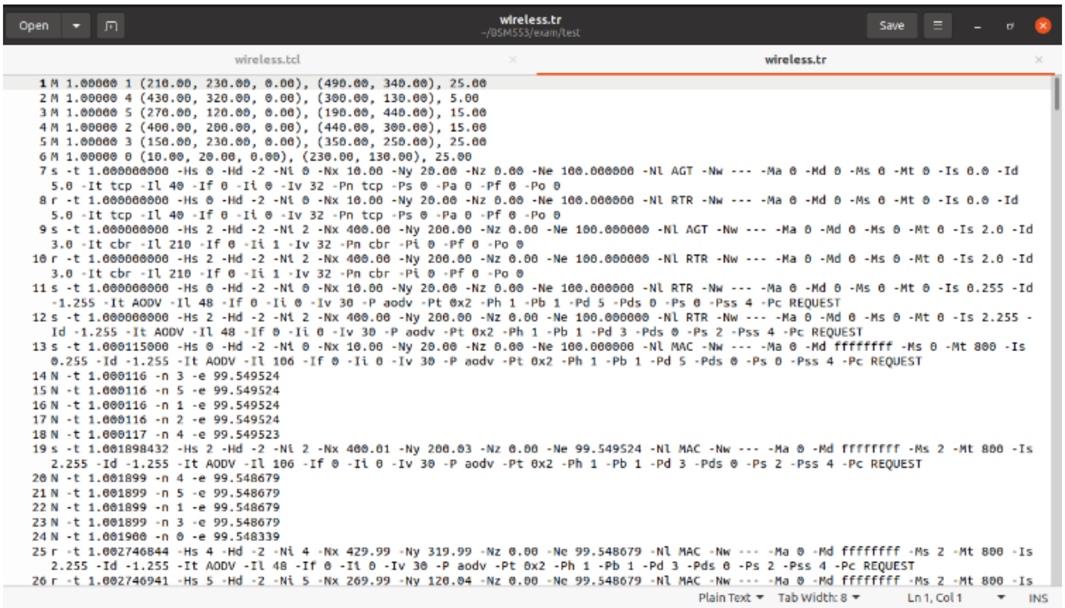

Figure 2: Trace file output (wireless.tr)

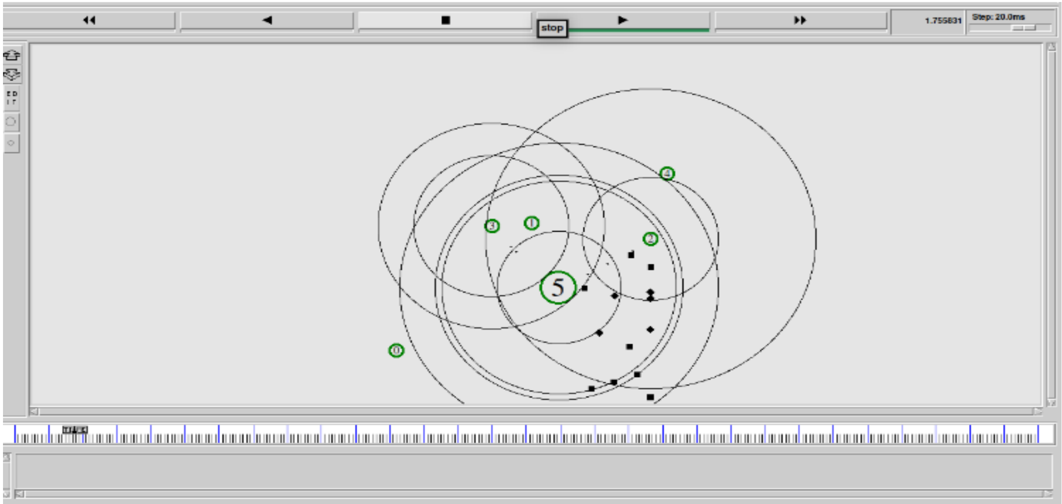

Figure 3. Nodes Motion and Drop packets.

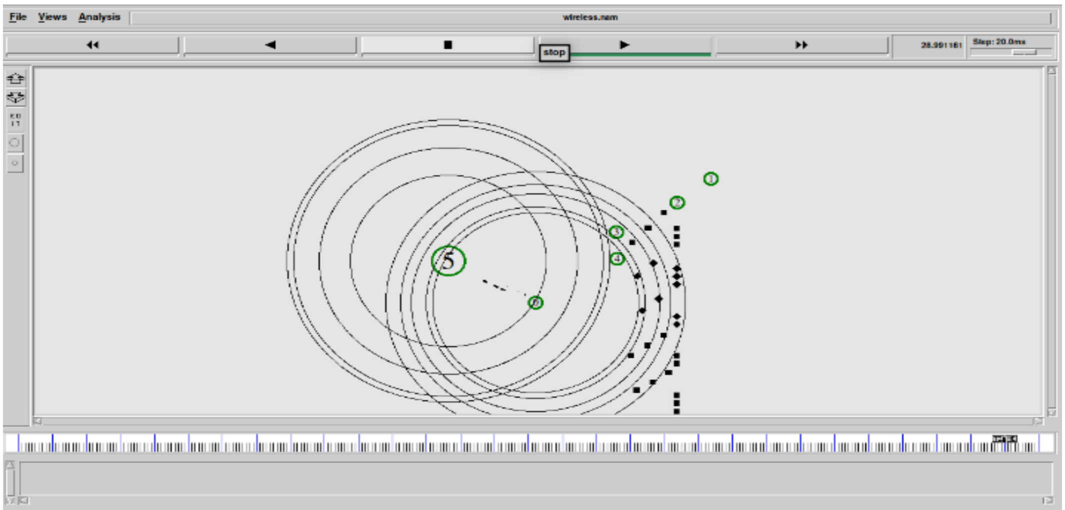

Figure 4. Node0 and Node5 Connection

Table 1: Average throughput

Function	Value
Start Time	1
Stop Time	29
Received Packets	3939
The throughput in kbps is	1.086713

4.4 Instant throughput

Using the awk script (instant_throu.awk), we got the throughput by time for the simulation, and by using Gnuplot, we can plot this data as a line chart, as shown in Figure 5, the throughput line chart of the model. As shown in Figure 5, the throughput increases at nine seconds until 27, then reaches the highest throughput rate between 27 and 29 through the simulation scenario. The x-axis is the time, and the y-axis is the throughput.

4.5 Packet delivery ratio:

Using the awk script to calculate the packet delivery ratio, we obtained the sent and received packets with the dropped packets, which are lost packets [2]. The details in Table 2 show the result of the script.

Table 2: Packet delivery ratio results

Variable	Value
Sent packets	9615
Received packets	3939
Forwarded packets	2493
Dropped packets	5735
Packet Delivery Ratio	0.409672

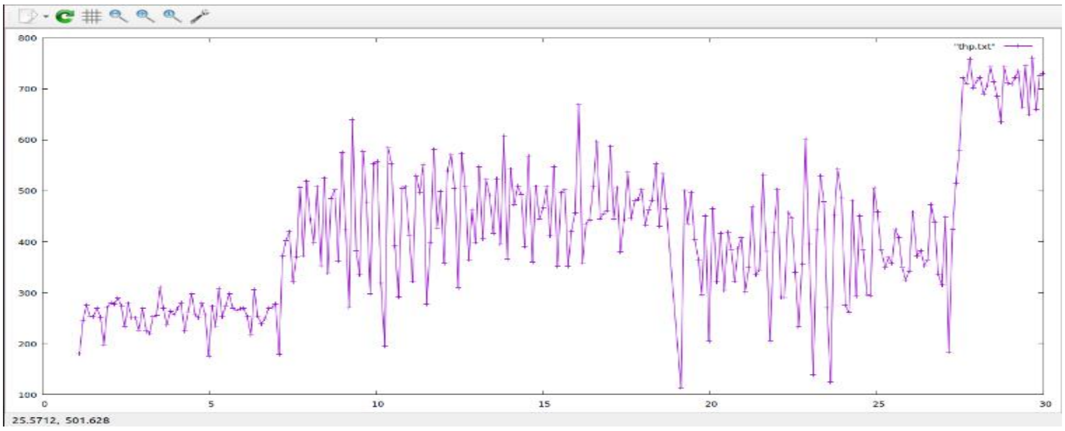

Figure 5. Throughput per time

Our model shows a 41% delivery ratio with 59% loss packets, 9615 packets sent, and 3939 packets received. The lost packets are 5735 packets. Thus, packets were in nodes (0,2,3 and 5). The graph in Figure 6 shows us that the lost packets were extremely high according to the sent and received packets.

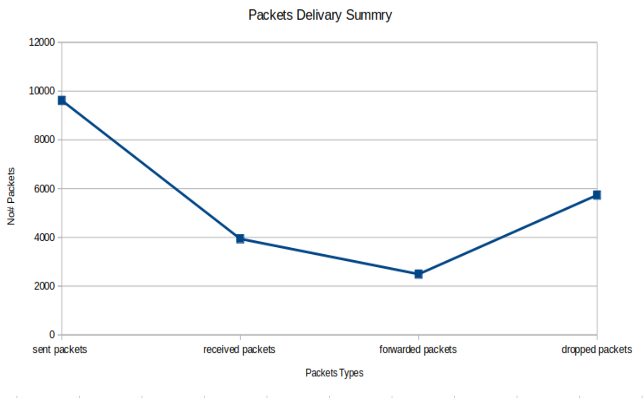

Figure 6. Delivery and dropped packets.

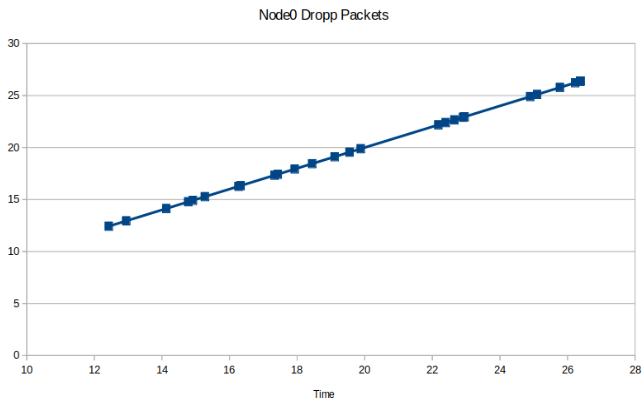

Figure 7. Node0 dropped packets by time.

In Figure 6, we can have a lot of dropped packets since the network nodes send 9615 packets and only 3939 packets are received; this affects the total dropped packets to be increased to 5735 dropped packets and only 2493 forward packets are transferred.

The graphs Figure 7-10 indicate the dropped packets in (Node0, Node2, Node3, and Node5, and the total dropped packets) in all nodes together) Through the simulation time, we can see that the number of dropped packets increases with the increase in simulation time. The highest dropped packet was node2, as shown in Figure 8.

Figure 7 shows the dropped packets in Node0, representing the node with fewer dropped packets than the others.

Figure 8 shows the first node of two nodes with a maximum dropped packets in our scenario, node2, and this node performs the highest drop packet rate among all nodes in our simulation scenario.

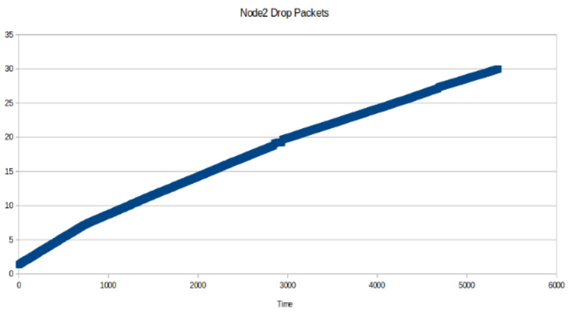

Figure 8. Node2 dropped packets by time.

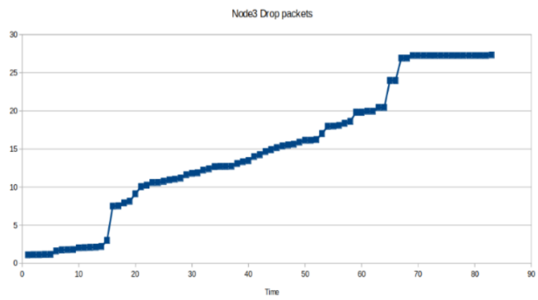

Figure 9. Node3 dropped packets by time.

Figure 9 depicts the rate of packet drops for node 3, which was moving in the middle of the simulated scenario and had a decent effect on a mobile node with this rate of packet loss.

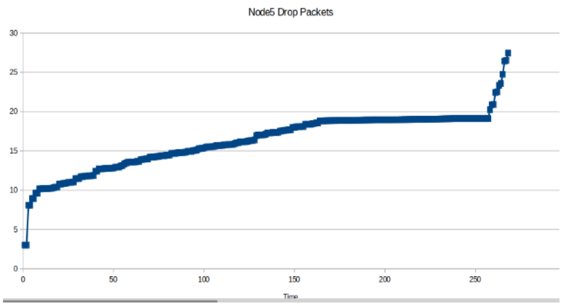

Figure 10. Node5 dropped packets by time.

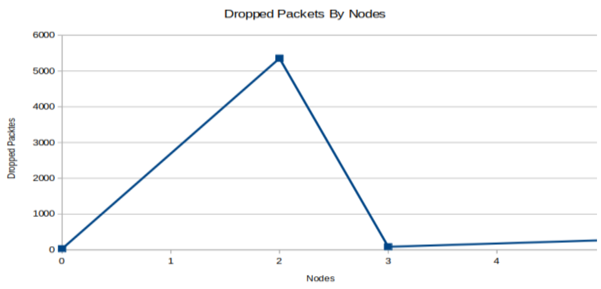

Figure 11. Dropped packets by node.

Figure 10 depicts the rate of missed packets for node 5, which serves as the study's focal point for the simulation scenario, as it moves away from and draws closer to the rest of the nodes.

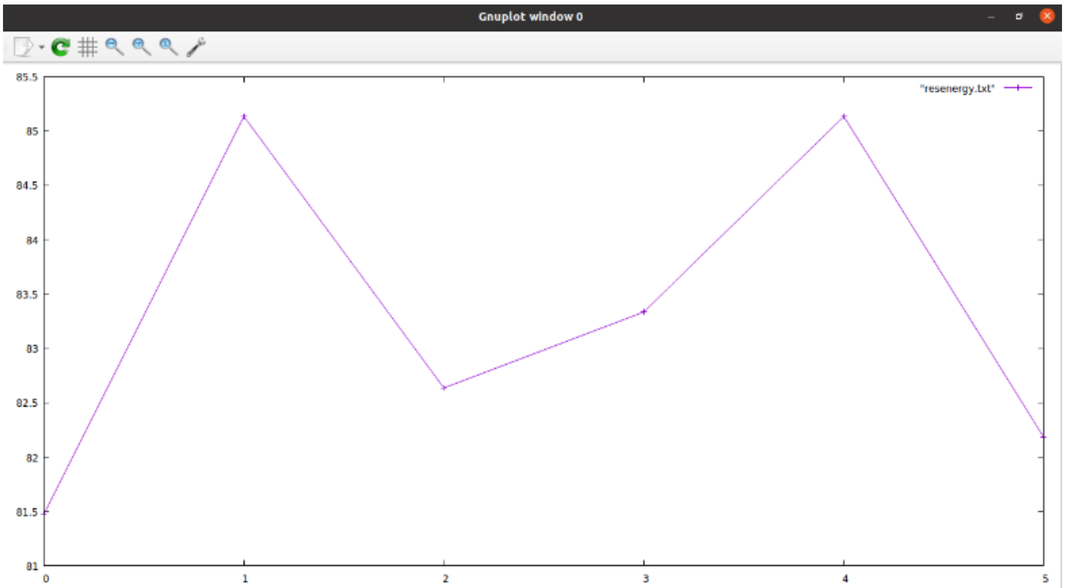

Figure 12. The residual energy of a node

Figure 11 shows the total rate of dropped packets for each node in our simulation scenario. Node 2 has the greatest rate of dropped packets, more than 5000, which means it performs 95 percent of the drooped packets in the entire scenario time, with 5735 total drooped packets in our simulation scenario.

4.6 Residual energy of a node

The utility model represents the energy level of network nodes [11]. The initial value of the power model set on a node is the node's power level at the beginning of the simulation. İlkEnerji_ is the name of energy. The different "energy" in the simulation represents the energy level at each moment. The initialEnergy_ value is given as an input parameter. Every packet delivered and received causes a node to lose a certain amount of energy. As a result, the initial Energy_ value in a node decreases. The current energy value in a node after receiving or transmitting routing packets is known as residual energy. Data transmission between nodes is established using a UDP agent and CBR traffic. The residual energy of the node is determined by accessing the inbuilt variable "energy" in the findEnergy function at various points, and the script (resenergy.awk) yields the following result. From Figure 12, we can see that node1 and node4 have the highest residual energy compared to the other nodes, and node2 has the lowest residual energy.

5. Discussion and Conclusions

In this study, NS-2 is used to simulate a wireless scenario. TCP and UPD protocols are used, and FTP and CBR applications are attached to the protocols. The loss packets and total sent and received packets with the instant and average throughput are calculated residual energy for each node are calculated, dropped packets and thruput and residual energy were plotted using Gnuplot tool and excel, packet delivery ratio was 41% with 59% loss packet, Network Congestion causing more drop packets. When network traffic reaches its maximum, the packet must wait to be sent. Unfortunately, packets are first thrown when the network tries to capture traffic, and the connection can only do so much. Fortunately, most software today can automatically resend lost packets or slow down the transmission rate to give each packet a chance to be received.

6. Declarations

6.1 Study Limitations

Data (preliminary results of the system, limited number of volunteer data).

6.2 Acknowledgements

The authors thank researchers and students for their contributions to Sakarya University.

6.3 Funding source

There is no funding source.

6.4 Competing Interests

There is no conflict of interest in this study.

6.5 Authors' Contributions

All authors: developing ideas for the research, planning the materials and methods to reach the results, taking responsibility for the experiments, organizing and reporting the data, taking responsibility for the explanation and presentation of the results, taking responsibility for the literature review during the research, taking responsibility for the creation of the entire manuscript, reworking not only in terms of spelling and grammar but also intellectual content or other contributions.

References

[1] A. A. K. Mohammad, and A. M. Abdul, “Optimised reactive resource aware routing for wireless infrastructure-less networks,” *International Journal of Computer Applications in Technology,* vol. 72, no. 2, pp. 131-138, 2023.

[2] J. Bienik *et al.*, “Impact of Packet Loss Rate on Quality of Compressed High Resolution Videos,” *Sensors,* vol. 23, no. 5, pp. 2744, 2023.

[3] C. A. G. d. Silva, and E. L. d. Santos, “A Compensation Model for Packet Loss Using Kalman Filter in Wireless Network Control Systems,” *Energies,* vol. 16, no. 8, pp. 3329, 2023.

[4] P. Kumar *et al.*, “CyberGuard: A Fortified Message Authentication Protocol with Digital Signatures in NS-2 Simulation Environment,” *Scalable Computing: Practice and Experience,* vol. 26, no. 1, pp. 114-122, 2025.

[5] S. Jahanbakhsh Gudakahriz *et al.*, “Link life time and energy-aware stable routing for MANETs,” *International Journal of Nonlinear Analysis and Applications*, 2025.

[6] A. Mhmood, and A. Zengin, “Performance Evaluation of MANET Routing Protocols AODV and DSDV Using NS2 Simulator,” *Sakarya University Journal of Computer and Information Sciences,* vol. 4, no. 1, pp. 1-10, 2021.

[7] M. Casquilho *et al.*, "Acceptance sampling in Quality Control: from theory to the Web: In PHP, Python, and gnuplot." pp. 1-6.

[8] B. Zoradia, and G. Indumati, “Comparison Of Software Defined Networking With Traditional Networking Using Ns2 SimulatoR,” *International Journal on Information Technologies & Security,* vol. 15, no. 3, 2023.

[9] Q. Monnet, and L. Mokdad, "Chapter 28 - DoS detection in WSNs: Energy-efficient designs and modeling tools for choosing monitoring nodes," *Modeling and Simulation of Computer Networks and Systems*, M. S. Obaidat, P. Nicopolitidis and F. Zarai, eds., pp. 795-840, Boston: Morgan Kaufmann, 2015.

[10] "Ns2 versus Ns3 Comparison between Ns2 and Ns3.."

[11] J. U. Kim *et al.*, “A Simple but Accurate Estimation of Residual Energy for Reliable WSN Applications,” *International Journal of Distributed Sensor Networks,* vol. 11, no. 8, pp. 107627, 2015.