NetSim®

Software: NetSim Standard v15.0, Visual Studio 2022

Blackhole and Sinkhole Attacks in AODV

Project Information

Reference Package

https://github.com/NetSim-TETCOS/Sinkhole_Blackhole_attack_in_AODV_v15.0/archive/refs/heads/main.zip

NetSim Setup Reference

https://support.tetcos.com/en/support/solutions/articles/14000128666-downloading-and-setting-up-netsim-file-exchange-projects

Pause-Time XML Requirement

Before opening the scenarios, replace the installed `mobility_models.xml` file with the file provided in the GitHub reference package. Copy it to:

```
C:\Program Files\NetSim\<edition_v15_0>\Docs\UI_xml\ExternalFiles\
```

For example, the Standard edition destination is:

```
C:\Program Files\NetSim\Standard_v15_0\Docs\UI_xml\ExternalFiles\
mobility_models.xml
```

The supplied XML sets the Random Way Point pause-time limits to 0–100 s, allowing the 0, 20, 40, 60, and 80 s values used in this study. Keep a backup of the installed file and restart NetSim after replacement so that the revised property limits are loaded.

Contents

1 Introduction

Mobile ad hoc networks depend on cooperation among nodes during route discovery and packet forwarding. This makes the routing process vulnerable to malicious control messages, especially in AODV where a forged route reply can attract traffic before a valid route is used. Once traffic is redirected, the attacker can either relay the packets through a manipulated path or discard them, which changes end-to-end packet delivery.

This study compares three 50-node AODV scenarios in NetSim: a normal case, a sinkhole attack case, and a blackhole attack case. The same mobility setting, traffic pattern, radio configuration, and pause-time sweep are used in all three cases so that the effect of the malicious route advertisement can be observed directly. In sinkhole mode, the attacker pulls traffic toward itself and forwards packets through the altered route state. In blackhole mode, the attacker pulls traffic and drops the captured packets.

Why this experiment matters

Reference [1] shows that malicious AODV behavior can reduce routing performance by interfering with route establishment and packet forwarding. The NetSim scenarios in this document reproduce that comparison pattern by placing normal, sinkhole, and blackhole behavior under the same N50 setup and then measuring packet delivery ratio over different pause times.

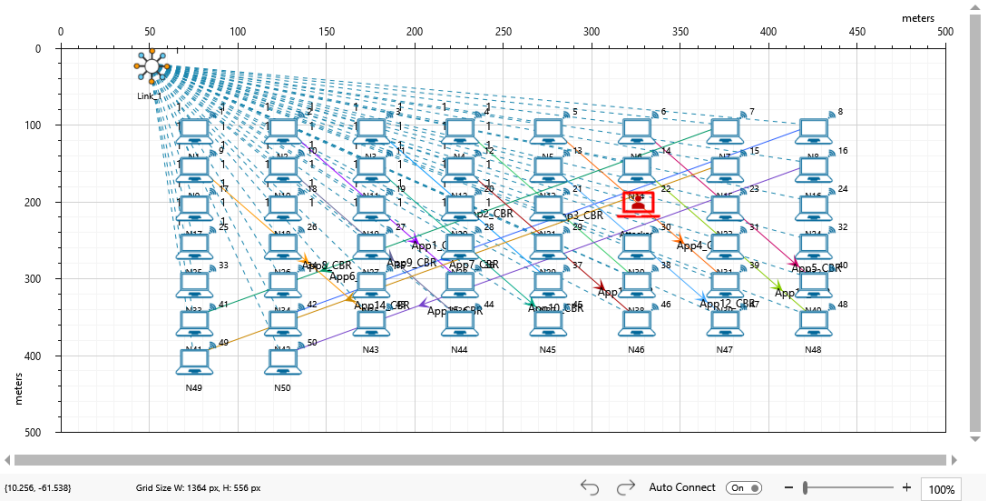

Figure 1-1: *N50 attacker scenario in NetSim with device 22 configured as the malicious node*

Figure 1-1 shows the NetSim scenario layout used for the attack analysis. The configuration contains 50 mobile wireless nodes in a 500 m × 500 m area, with device 22 assigned as the malicious node. The same N50 layout, traffic pattern, and mobility settings are used for the normal, blackhole, and sinkhole cases so that the effect of the malicious AODV route advertisement can be compared under the same simulation conditions. The traffic configuration uses 15 UDP unicast CBR applications, each generated at 16 kbps with 512-byte packets, which gives an aggregate offered load of 240 kbps. The application flows start at 5 s and continue through 19 s, with source-destination pairs ranging from 2→36 to 16→50.

2 Implementation in AODV

- In AODV, the source broadcasts the `RREQ` packet during route discovery.
- The destination, upon receiving the `RREQ` packet, replies with an `RREP` packet containing the route to reach the destination.
- Intermediate nodes can also send `RREP` packets to the source if they have a route to the destination in their route cache.
- A malicious node can exploit this behavior by adding a fake route entry into its route cache, with hop count 0 so the route appears attractive.
- In this implementation, the forged route is inserted inside `fn_NetSim_AODV_MaliciousRouteAddToCache()` when the malicious node receives an `RREQ`. After that, the normal AODV `RREQ` processing continues and uses the forged cache entry.
- The source node accepts this reply before the valid route becomes useful.
- All network traffic is then attracted toward the attacker, which can either forward it as a sinkhole relay or drop it as a blackhole.

A file named `malicious.c` is added to the AODV project. It contains the following functions:

- `fn_NetSim_AODV_MaliciousNode();` Identifies whether the current device is malicious and enables malicious behavior.
- `fn_NetSim_AODV_MaliciousRouteAddToCache();` Inserts the forged route entry used to generate the false `RREP`.
- `fn_NetSim_AODV_MaliciousHandleDataForward();` Applies the post-capture behavior. In blackhole mode it drops the packet locally; in sinkhole mode it leaves the packet in the AODV forwarding path.
- `is_sinkhole_forwarded_packet_to_be_ignored();` Checks whether a neighboring relay still has the malicious node as the next hop to the destination and should ignore the forwarded packet.
- `fn_NetSim_AODV_SinkholeIgnoreForwardedPacket();` Applies the sinkhole ignore action by marking the packet as dropped, writing the packet trace, and freeing the packet.

In the evaluated configuration, `fn_NetSim_AODV_MaliciousNode()` marks device 22 as the malicious node through `MALICIOUS_NODE1`. The attack mode is selected by the `ATTACK_MODE` constant in `malicious.c`. For the blackhole configuration discussed here, `ATTACK_MODE` is set to `ATTACK_BLACKHOLE`. Additional malicious nodes can be added by extending the device-ID check in the same function.

2.1 What the Malicious Node Changes

- It injects a false route reply so the source selects the attacker during route discovery.
- It uses hop count 0 to make the forged route look better than the valid route.
- It switches between sinkhole and blackhole behavior through the attack mode setting.

2.2 Sinkhole Versus Blackhole

The IEEE paper treats both sinkhole and blackhole as route disruption attacks. Both send false routing information and pull traffic toward the attacker. The difference is what happens after traffic is captured.

- In a **sinkhole** attack, the malicious node advertises an attractive route and forwards the packet again. A separate helper then checks whether the next relay still has the malicious node as its next hop to the destination. If so, the forwarded packet is ignored and logged as a sinkhole drop.
- In a **blackhole** attack, the malicious node attracts traffic and then drops it outright inside `fn_NetSim_AODV_MaliciousHandleDataForward()`.

This NetSim implementation contains both attack modes in the same source file. The attack behavior is selected through `ATTACK_MODE` in `malicious.c`, and the code must be rebuilt before running the corresponding scenarios.

2.3 Steps to Simulate

1. Open the source code in Visual Studio by navigating to "Your work" on the NetSim home screen. Select "Open, Reset or Compare" and click "Open Code".
2. Expand the AODV project, open `malicious.c`, set the malicious node ID, and select either `ATTACK_SINKHOLE` or `ATTACK_BLACKHOLE`.

Attack configuration settings

Inside `malicious.c`, the two settings that control the attack experiment are:

- `#define MALICIOUS_NODE1 22` selects device 22 as the attacker.
- `static const AODV_ATTACK_MODE ATTACK_MODE = ...` selects `ATTACK_BLACKHOLE` or `ATTACK_SINKHOLE`. In the blackhole configuration described here, it is set to `ATTACK_BLACKHOLE`.

3. Right-click on the AODV project and rebuild it.

Build step

After changing the malicious node ID or attack mode in `malicious.c`, rebuild the AODV project in Visual Studio so the updated `libAODV.dll` is copied into the project `bin` folder.

4. Upon rebuilding, `libAODV.dll` gets updated in the corresponding `bin` folder of the project.

3 N50 Scenario Set

To compare the behavior reported in Reference [1], this study uses three 50-node scenario families for pause times 0, 20, 40, 60, and 80 seconds:

- `N50-P{0,20,40,60,80} - Normal`
- `N50-P{0,20,40,60,80} - Blackhole`
- `N50-P{0,20,40,60,80} - Sinkhole`

These scenarios use the same topology, mobility, wireless settings, traffic pattern, and simulation duration so that the packet delivery changes come from the routing behavior under comparison.

Table 3-1: *Shared N50 scenario properties*

Property	Value
Node count	50 wireless nodes
Simulation area	500 m × 500 m grid
Mobility model	Random Way Point
Node speed	10 m/s
Pause times	0, 20, 40, 60, 80 s
Wireless standard	IEEE 802.11b
Channel model	Pathloss only, Range Based
Wireless range	100 m
Routing protocol	AODV
Transport/Application	UDP unicast CBR
Number of CBR applications	15
Traffic rate per application	16 kbps
Packet size	512 bytes
Inter-arrival time	256000 μs
Application start times	5, 6, 7, . . . , 19 s
Application end time	100000 ms
Simulation time	100 s
Packet trace	Enabled

Table 3-2: *Case-specific settings used in the N50 comparison*

Case	Configuration used in this study
Normal	Same 50-node topology and traffic set, used as the non-malicious reference for the pause-time sweep.
Blackhole	`MALICIOUS_NODE1 = 22` in `malicious.c`; `ATTACK_MODE = ATTACK_BLACKHOLE`; the malicious node sends a forged RREP with hop count 0 and drops captured packets.
Sinkhole	`MALICIOUS_NODE1 = 22` in `malicious.c`; `ATTACK_MODE = ATTACK_SINKHOLE`; the malicious node sends the same forged RREP and forwards traffic as a malicious relay, after which downstream forwarding can still fail because the next hop remains pointed to the attacker.

Why the three-case setup is useful

- The normal case provides the reference obtained with the same N50 topology and pause-time sweep.
- The blackhole case isolates the effect of route capture followed by packet drop.
- The sinkhole case isolates the effect of route capture followed by malicious relaying.
- Because the shared settings are fixed, the PDR differences are easier to interpret against pause time.

4 Results and Discussion

The N50 comparison was evaluated using packet delivery ratio (PDR), computed from the NetSim `Application_Metrics` table as:

$$\text{PDR} = \frac{\text{Total packets received}}{\text{Total packets generated}} \times 100$$

Table 4-1: *PDR versus pause time for the N50 scenario set*

Pause time (s)	Normal PDR (%)	Blackhole PDR (%)	Sinkhole PDR (%)
0	26.5247	24.1045	19.2062
20	95.9148	60.9487	25.7115
40	98.0058	63.5818	35.8761
60	99.2643	64.2207	63.1171
80	99.0900	66.9313	49.6031

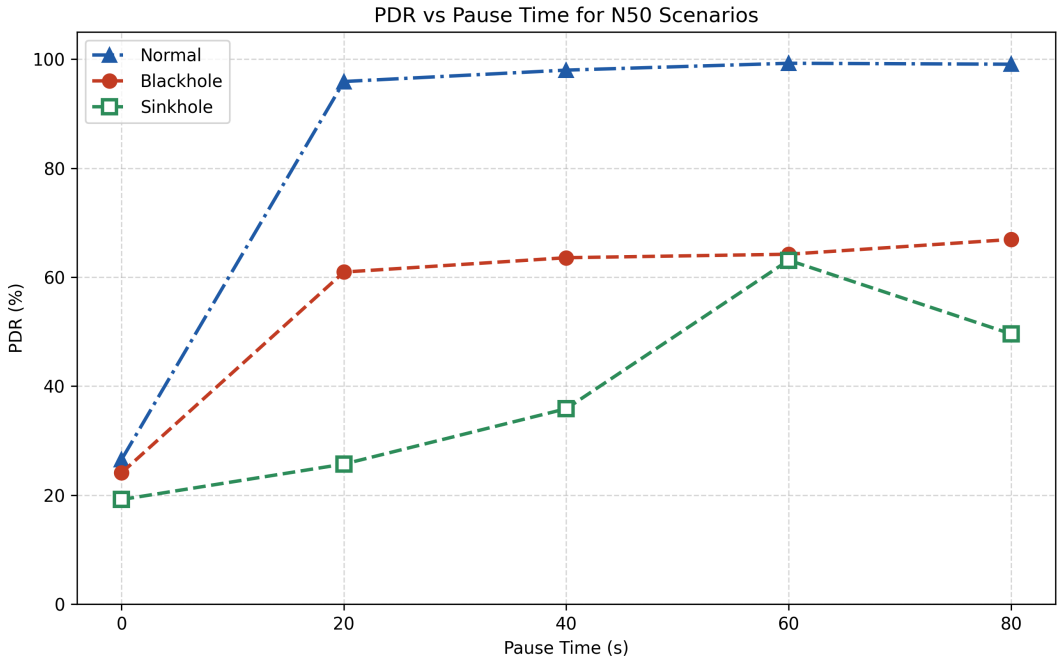

Figure 4-1: *PDR versus pause time for N50 normal, blackhole, and sinkhole scenarios*

1. The normal scenario gives the highest PDR at every pause-time value in the N50 results. The best normal result appears at 60 s pause time, where the PDR reaches 99.2643%.

2. The sinkhole scenario gives the lowest PDR at every pause-time value in the N50 results. The largest gap appears at 20 s pause time, where normal reaches 95.9148%, blackhole reaches 60.9487%, and sinkhole remains at 25.7115%.

3. The blackhole scenario stays between normal and sinkhole for the full pause-time sweep. This shows that forged route capture followed by packet drop reduces delivery strongly, but not as much as the sinkhole behavior observed in the N50 simulations.

4. Both attack scenarios improve as pause time increases from 0 s to 60 s, which indicates that lower mobility helps packet delivery even when the route has been manipulated.

5. At 80 s pause time, blackhole reaches 66.9313% while sinkhole falls to 49.6031%. The two attack curves therefore do not overlap in the N50 result set.

6. Overall, the graph shows that the normal scenario preserves the best end-to-end delivery, while the sinkhole case causes the strongest packet-delivery reduction across the pause-time sweep.

4.1 Throughput

Throughput was also taken from the `Application_Metrics` table after converting the reported values from Mbps to kbps. The per-application throughput tables for the normal, blackhole, and sinkhole cases are moved to Appendix A. Table 4-2 and Figure 4-2 summarize the aggregate totals across the same pause-time sweep.

Table 4-2: *Aggregate throughput versus pause time for the N50 scenario set*

Pause time (s)	Normal (kbps)	Blackhole (kbps)	Sinkhole (kbps)
0	62.7375	56.6360	45.1102
20	230.5133	145.8333	60.3931
40	235.6062	151.7846	84.2352
60	238.6756	153.5188	149.1640
80	238.1354	159.7905	116.3396

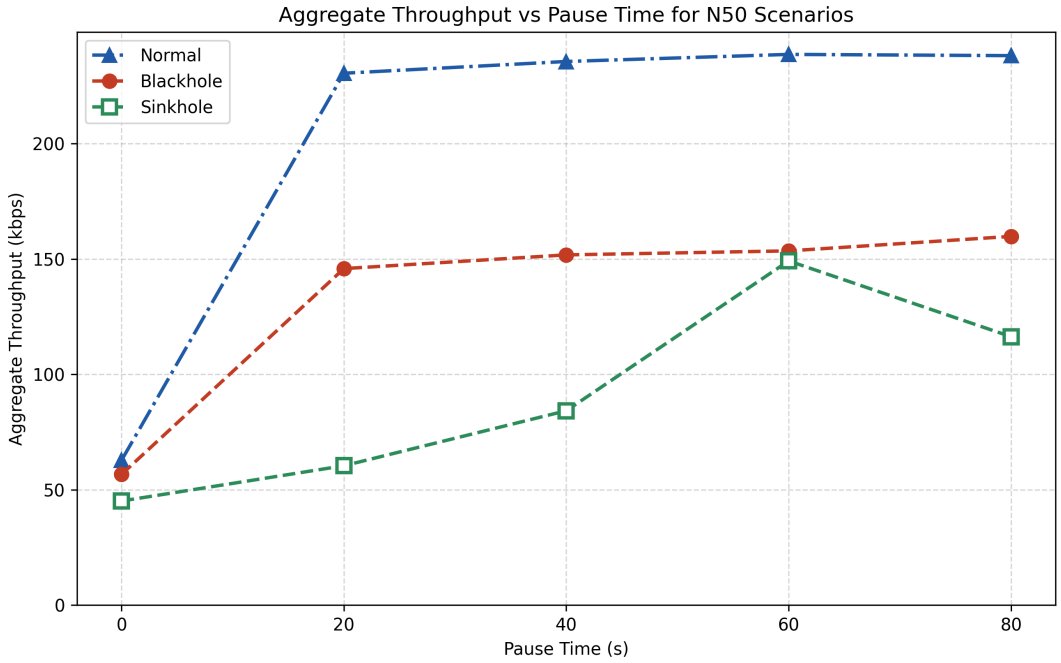

Figure 4-2: *Aggregate throughput versus pause time for N50 normal, blackhole, and sinkhole scenarios*

The throughput plot follows the same ordering seen in the PDR results. The normal case reaches the highest aggregate throughput from 20 s onward, blackhole stays in the middle, and sinkhole remains lowest except near 60 s where it rises close to blackhole. At 80 s pause time, normal reaches 238.1354 kbps, blackhole reaches 159.7905 kbps, and sinkhole reaches 116.3396 kbps.

4.2 Packet-Trace Loss Analysis

The following observations come from `Packet Trace.csv` and are separate from the PDR, throughput, and mean-PDR plots.

- In the blackhole runs, dropped CBR packets terminate at the malicious node itself. In the 20 s and 80 s pause-time blackhole traces, every dropped CBR row has `RECEIVER_ID = NODE-22`.
- In the sinkhole runs, the malicious node does not drop the captured traffic locally. The later dropped rows appear with `TRANSMITTER_ID = NODE-22`, which shows that node 22 forwards the traffic and the loss occurs after that forwarding step.

- The sinkhole trace therefore shows a downstream-loss pattern, while the blackhole trace shows a local-drop pattern at the attacker.
- The same sinkhole traces also contain many collision events, which indicates more contention around the forged forwarding path.

To reduce dependence on a single random realization, each N50 scenario family was also evaluated over five seed settings and summarized using the sample mean PDR with a 95% confidence interval. The resulting pause-time means are listed in Table 4-3 and plotted with confidence bands in Figure 4-3.

Table 4-3: *Mean PDR versus pause time for the N50 scenario set over five seeds*

Pause time (s)	Normal mean PDR (%)	Blackhole mean PDR (%)	Sinkhole mean PDR (%)
0	21.4482	16.6041	22.2653
20	87.6786	51.9032	39.8296
40	98.3737	53.0184	48.8364
60	99.5005	53.1888	58.6254
80	99.0397	54.4705	59.5547

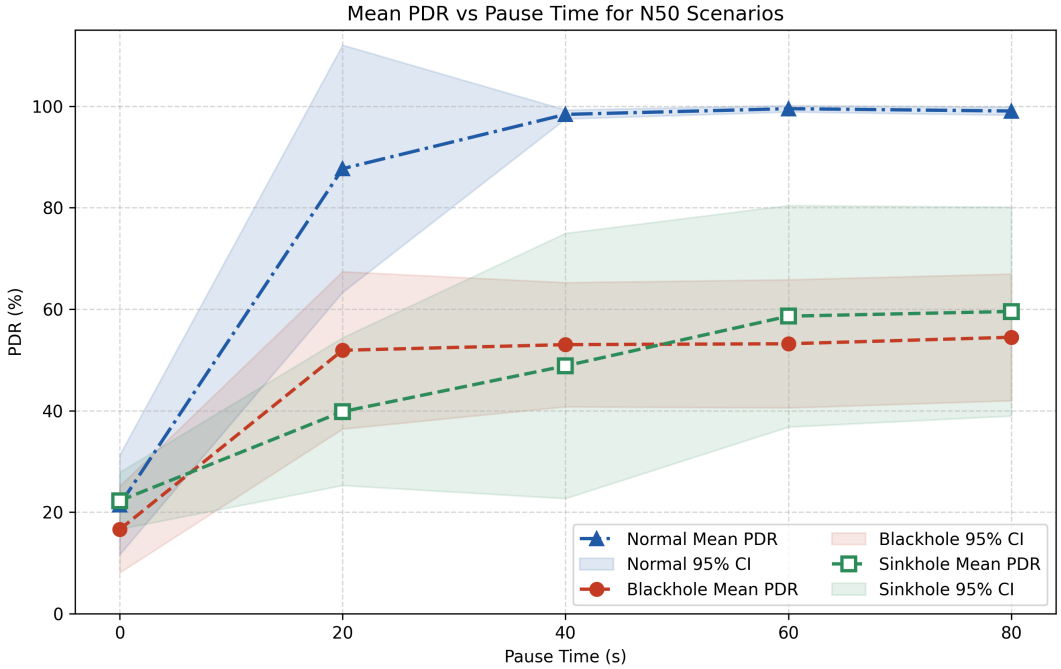

Figure 4-3: *Mean PDR versus pause time for N50 normal, blackhole, and sinkhole scenarios with 95% confidence bands*

The mean-PDR plot summarizes the same pause-time sweep over five seed settings and adds a 95% confidence band for each case.

- At 0 s pause time, all three cases begin with low mean PDR. The confidence bands are wider

in this high-mobility setting, which shows larger seed-to-seed variation during continuous movement.

- At 20 s pause time, the normal case rises to 87.6786%, while blackhole reaches 51.9032% and sinkhole reaches 39.8296%. The separation between the non-malicious and malicious cases is clear at this point in the sweep.
- At 40 s pause time, the normal case reaches 98.3737% and its confidence band becomes narrow. Blackhole remains at 53.0184% and sinkhole remains at 48.8364%, so both attack cases stay far below the baseline.
- At 60 s pause time, the normal case reaches the highest mean PDR in the five-seed summary, at 99.5005%. Sinkhole improves to 58.6254% and exceeds blackhole at 53.1888%, but both remain well below the normal case.
- At 80 s pause time, the normal case stays high at 99.0397%. Blackhole reaches 54.4705% and sinkhole reaches 59.5547%, which shows that lower mobility helps the malicious cases, but does not close the gap to the baseline.
- Across the full sweep, the normal case gives the highest mean PDR. The blackhole and sinkhole curves stay below the baseline, and the sinkhole confidence bands remain wider at several pause-time values, which shows stronger seed-to-seed variation in the forwarding-disruption pattern.

Table 4-4: *How to interpret the N50 comparison*

Observation	Interpretation
Normal mean PDR remains highest	Across the five-seed summary, the normal case gives the highest mean PDR at every pause-time value, rising from 21.4482% at 0 s to 99.5005% at 60 s.
Blackhole mean PDR remains below the normal baseline	The blackhole case improves from 16.6041% at 0 s to 54.4705% at 80 s, but it stays well below the corresponding normal mean throughout the pause-time sweep.
Sinkhole shows stronger variability than blackhole	The sinkhole mean rises from 22.2653% at 0 s to 59.5547% at 80 s, while the wider confidence bands indicate larger seed-to-seed variation in the forwarding-disruption pattern.
Higher pause time improves mean delivery	All three cases show higher mean PDR as pause time increases from 0 s to 60 s, which indicates that lower mobility reduces route disruption and helps end-to-end delivery.

Connection to the IEEE paper

Reference [1] reports that malicious AODV routing attacks degrade routing performance because the attacker interferes with route formation and packet forwarding. The N50 scenario set in this document follows that comparison style by using a normal baseline and two malicious variants over the same pause-time sweep. The NetSim results show the same

forged-route capture mechanism in both attacks, with the sinkhole case producing the lowest PDR and the normal case producing the highest PDR.

5 Exercise

The following steps can be used to extend the N50 comparison:

1. Open `AODV/malicious.c` and change `ATTACK_MODE` between `ATTACK_BLACKHOLE` and `ATTACK_SINKHOLE`.
2. Rebuild the AODV project so `libAODV.dll` is refreshed in the project `bin` folder.
3. Re-run the same N50 scenarios and compare packet trace, application metrics, and PDR against the normal baseline.
4. Repeat the same method for other node counts or radio ranges if you want a wider comparison against the paper.
5. Other AODV routing attacks discussed in Reference [1] can also be implemented and analyzed in NetSim using the same baseline, traffic settings, pause-time sweep, and performance metrics.

6 References

[1] F. A. K. Humaira Ehsan, "Malicious AODV Implementation and Analysis of Routing Attacks in MANETs," 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications.

A Per-Application Throughput Tables

Table A-1: *Per-application throughput for the N50 normal scenario set (kbps)*

Application	0 s	20 s	40 s	60 s	80 s
App1	5.8206	15.8235	15.7373	16.0391	16.0822
App2	7.0591	15.5561	16.0354	16.0354	16.0354
App3	6.6945	15.6793	15.5472	16.0317	16.0757
App4	4.4077	15.3600	16.0724	16.1169	16.0724
App5	4.9962	14.8086	15.0787	15.5288	16.0689
App6	2.3666	15.2007	16.0199	15.5193	16.0199
App7	2.8074	14.5891	15.5556	15.5096	15.5096
App8	3.1185	15.4996	15.4531	16.0116	15.4531
App9	10.9698	15.5366	15.4895	15.4424	15.4895
App10	3.7626	15.7172	15.7172	16.0506	16.1459
App11	4.4333	16.0949	16.0949	16.0467	16.0467
App12	2.2430	15.2625	15.5063	16.0914	16.0914
App13	1.4311	16.0879	16.0879	16.0879	16.1866
App14	0.1998	14.8855	15.5348	16.1342	15.4349
App15	2.4273	14.4118	15.6760	16.0300	15.4232
Aggregate	**62.7375**	**230.5133**	**235.6062**	**238.6756**	**238.1354**

Table A-2: *Per-application throughput for the N50 blackhole scenario set (kbps)*

Application	0 s	20 s	40 s	60 s	80 s
App1	3.7942	15.5648	15.8235	16.0822	16.0822
App2	3.3117	15.5125	16.0354	16.0790	16.1226
App3	7.0909	15.5472	15.2389	16.0317	16.0317
App4	8.7263	0.0445	0.0445	0.0445	0.0445
App5	10.1725	5.3563	12.4230	9.3623	15.9789
App6	0.0000	0.0000	0.0000	0.0000	0.0000
App7	0.0000	0.0000	0.0000	0.0000	0.0000
App8	5.6320	15.4996	16.1047	16.0582	16.1047
App9	10.2635	15.5366	15.4424	15.5366	15.5836
App10	4.2389	15.7172	15.8125	16.0506	16.0506
App11	1.1083	16.1431	16.1431	16.1431	15.6130
App12	1.9017	14.8236	12.6781	16.0427	16.1402
App13	0.3454	16.0879	16.0385	16.0879	16.0385
App14	0.0000	0.0000	0.0000	0.0000	0.0000
App15	0.0506	0.0000	0.0000	0.0000	0.0000
Aggregate	**56.6360**	**145.8333**	**151.7846**	**153.5188**	**159.7905**

Table A-3: *Per-application throughput for the N50 sinkhole scenario set (kbps)*

Application	0 s	20 s	40 s	60 s	80 s
App1	5.1308	7.4159	12.8485	16.1253	12.6760
App2	7.0591	7.2769	10.2836	15.5125	12.2008
App3	4.9328	7.8397	8.5884	12.9486	16.3400
App4	0.3562	4.0515	6.0550	14.2915	9.4831
App5	8.1470	3.0607	6.2115	15.5738	16.2940
App6	0.0000	0.9102	0.7737	1.1833	0.7282
App7	0.1381	0.5983	0.6443	2.3932	1.7028
App8	7.5404	11.2640	12.1018	20.1542	13.7774
App9	6.5442	7.7212	9.0865	15.6778	11.0168
App10	0.0000	5.1438	9.1446	12.0499	11.9546
App11	1.6866	3.2768	5.2043	8.9148	6.0235
App12	0.1950	0.5851	0.5364	0.5364	0.5364
App13	0.0493	0.4935	0.4441	0.4935	0.6909
App14	0.5495	0.2498	1.0989	11.4888	1.4486
App15	2.7812	0.5057	1.2136	1.8204	1.4665
Aggregate	**45.1102**	**60.3931**	**84.2352**	**149.1640**	**116.3396**