International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 12, Issue 8 [August. 2023] PP: 75-81

Researching the Effect of RREQ Packaging Floating Against In AODV Protocol on MANET

Bui Quoc Tu
Faculty of Military and Sports, Air Force Officer School

ABSTRACT: *In AODV routing protocol on Manet network flooding attack increases route discovery time and communication waste. To perform the attack, the malicious node broadcasts a burst of packets as RREQ. A number of recent studies have been built and simulated to detect the RREQ packet flooding attack process based on the threshold value. In this paper, we build a malicious attack node that performs RREQ . packet flooding attack in the AODV routing protocol based on the NTU_attack framework . Simulation results on OMNeT++ show that parameters such as terminal latency, number of successful packet transfers and throughput change significantly when the malicious node has public behavior in the network.*
KEYWORDS: AODV , MANET , protocol , public attack .

Date of Submission: 10-08-2023 Date of acceptance: 25-08-2023

I. INTRODUCE

MANET is a type of mobile network with outstanding advantages in data communication: flexible infrastructure, mobility support, allowing better connectivity, ensuring stable handover between different networks, etc. is considered a convenient communication tool, occupies an important position and is expected to be very popular in the future. In MANET networks, nodes that work together to communicate should assume the function of routers, routing services provided at the network layer are the target of many types of denial of service (DoS [2]) attacks. , typically a flood attack [3]. The attack is done through the malicious node sending flooding system packets to nodes that do not exist in the network, or transmitting a large amount of useless data packets to cause network congestion. As a result, it creates a storm of broadcast packets on the network, increasing communication costs, and reducing responsiveness at each node because it has to handle unnecessary packets.

The routing protocol AODV (Ad-hoc On-Demand Distance Vector) [7] is built from two routing protocols DSDV and DSR. In this routing protocol, when nodes have a need to exchange information with each other, they will find a route to the most suitable destination , and it uses the traditional routing table to store routing information with each other. each entry for a destination address. This protocol does not require nodes to always maintain routes to destinations, since they are not always used. Only the endpoints of a connection have a suitable route to reach each other. To construct a route, AODV uses control packets to find and maintain connections, including: Routing request packets (RREQs), routing request response packets (RREPs), and routing request packets. HELLO and routing error (RERR) packets. AODV is the standard protocol that belongs to the group of protocols . As required , it is easy to believe that it is easy to perform attacks on this protocol , typically a packet attack . HELLO, RREQ package, and DATA package. [1].

1. Flood package HELLO

In the AODV protocol, a HELLO packet is periodically broadcast to announce the existence of a node something with a neighboring node. Hackers take advantage Due to this feature, HELLO packet flooding increases communication waste.

2. Flooding the RREQ . package

The RREQ route request packet in AODV is received by the node The source is used to perform route discovery when there is a need to exchange information with any destination node on the network. Attackers attack by over-broadcasting the RREQ packet, flooding the network with unnecessary traffic, affecting other nodes' route discovery, and increasing communication overhead.

3. Flooding of the DATA . package

The malicious node over-transmits data packets to any node on the network, affecting the bandwidth, processing capacity and causing congestion at some nodes participating in data routing.

II. BUILDING A SCENE OF SYMPTOMS OF FLOOD ASSIST

1. Platform NTU_attack framework

The NTU_Attack system is designed based on inheritance and development from the idea, architecture and code organization of the NETA framework and adheres to the following design principles:

- Do not modify the INET and OMNeT++ frameworks code;
- Modify as little as possible hacked modules to perform malicious behavior;
- Implement the principle of object-oriented programming in inheriting the existing components of OMNeT++ and INET overriding methods to perform system construction.

NTU-Attack aims to work compatible on OMNeT++ 5.6 and INET 4.x platforms, support to simulate attack scenarios on MANET, test new protocols, attack prevention and detection techniques .

NTU-ATTACK, a custom mobile wireless network attack simulation platform on OMNeT++, has been researched and built by author Mai Cuong Tho [3].

2. Build a node that performs a flood attack

Simulate flood attack behavior based on NTU_attack platform to design on MANET mobile wireless customized network. In order to do that, you must first build modules to perform the attack.

- Step 1: Build a controller module consisting of 2 components: Attack controller and controMesage

+ Build Attack controller: Name the folder named NTU_Attack_Controller to send messages that trigger the attack on the AODV routing protocol, flood attack in the AODV routing protocol, we have the information. For example: Attacking a RREQ packet, a HELLO packet, or a DATA packet, it is necessary to create logical variables to indicate whether the attack is performed or not, and also indicate in each attack the time to send the packets. RREQ, HELLO package, or DATA package. Therefore, the Attack controller menu is only responsible for sending messages that transmit parameters to trigger the attack and stop the attack.

+ Build AttackcontrolMessages: Generate FloodingAttack message located in controlMessages folder. For the behavior of sending attack packets, parameters such as: Attack time, attack type, message structure including 6 fields of information, corresponding to RREQ packet flooding, HELLO packet flooding and packet flooding DATA, these variables indicate the attack type and its duration.

- Step 2: Build AODV module to perform flood attack

To build the module simply copy the original AODV and modify a piece of code. Create a message and name it Flooding_AODV, this directory is the original AODV directory and has been edited, it has the function of launching a flood attack against the RREQ packet. By default the structure is exactly the same as the original AODV and added two methods:

The first method is responsible for processing messages from the Attack controller, when the Attack controller sends messages to the AODV protocol, extracts the messages and retrieves information such as: Decide whether to trigger the flood attack . or not, and how much time each flood attack takes. At the same time, we use the ScheduleAt OMNET function to automatically trigger sending that message after a preset period of time. In this simulation, I perform RREQ packet flooding attack, so after a preset period of time, it will automatically flood the packet.

```
/* floodNTU_ATTACKS: handle message from controller. */
void floodNTU_Aodv::handleMessageFromAttackController(cMessage *msg){
    Enter_Method("floodNTU_Aodv: handle message from attack controller");
    LOG << "floodNTU_Aodv: Received message: "<< msg->getFullName() << "\n";

    //  flooding attack
    if (not strcmp(msg->getFullName(), "floodingActivate")) {
            NTU_FloodingM *fmsg;
            fmsg = check_and_cast<NTU_FloodingM*>(msg);
            LOG << "--> Activating module floodNTU_Aodv for Flooding...\n";
            LOG << "    HelloFlooding ?: "<< fmsg->getIsHelloFlooding() << "\n";
            LOG << "    DataFlooding ?: "<< fmsg->getIsDataFlooding() << "\n";
            LOG << "    RredFlooding ?: "<< fmsg->getIsRreqFlooding() << "\n";
            floodingAttackIsActive =true;
            rreqFloodingTimer = new cMessage("rreqFloodingTimer");
            scheduleAt(simTime() + 5, rreqFloodingTimer);
        }
```

Figure 2. Handling messages from the Attack controller

+ The second method HandleMessagesWhenUp: When this protocol receives a message from the scheduler (from the timer that automatically generates the RREQ packet), if that message is a message generated by that scheduler, it will call the function with named handleRreqfloodingTimer, for this function need to handle

the following: create a fake address according to the principle of flooding attack (eg addr.set("1.2.3.4", broadcast RREQ packet to a destination address) bogus destination), call the built-in AODV createRREQ function and send the spoofed RREQ packet with a method named sendFakeRREQ, the structure of the method is unchanged from the original structure of the method. AODV.

```
// thomc flooding
void floodNTU_Aodv::handleRreqFloodingTimer()
{
    EV_INFO << "It is Flooding RREQ time" << endl;
    Ipv4Address addr;
    addr.set("1.2.3.4");
    L3Address destAddr;
    destAddr.set(addr);
    auto Frreq = createRREQ(destAddr);

    int rrqTTL =20;
    sendFakeRREQ(Frreq, addressType->getBroadcastAddress(), rrqTTL);
    scheduleAt(simTime() + rreqFlooding_Interval, rreqFloodingTimer);
}
```

Figure 3. handleRreqfloodingTimer . function

- Step 3: Build flood attack node

Name the node as NTU_flooding_Attacker, this node is built inherited on NTU_AdhocHost and uses the modified AODV protocol to floodNTU_AODV and needs to attach to this node an Attack controller (NTU_FloodingAttack built above) and process the the following parameters: active=true, starTime = 10s (activated after 10s), isRreqFlooding = true (only flood RREQ packets), RREQFlooding_interval = 0.07s (time to flood RREQ packets is 0.07 seconds).

```
module NTU_Attacker_Flooding extends NTU_AdhocHost
{
    submodules:
        aodv: floodNTU_Aodv {
            @display("p=872.3,211.9");
        }
        ntu_FloodingAttack: NTU_FloodingAttack {
            @display("p=898.3,74.1");
            active = true;
            startTime = 10s; //uniform( 10s,3s);
            isHelloFlooding = false;
            isRreqFlooding = true;
            isDataFlooding = false;
            Interval_RreqFlooing = 0.07;
        }
```

Figure 4. Building a flood attack node

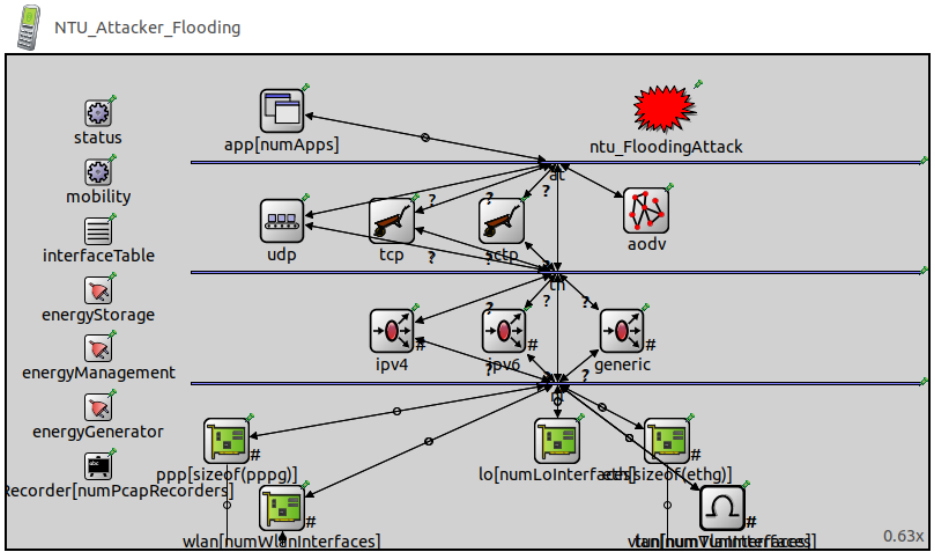


Figure 5. NTU_flooding_Attacker

III. ASSESSMENT OF RESULTS BY SIMULATION

1. Simulation parameters and evaluation criteria

Software The OMNet++ model is used to evaluate the effectiveness of the parameters when the malicious node has an attack of the AODV protocol . The network model is built with 20 nodes, including a sender node, a receiver node and 18 normal nodes, simulating a malicious node attack, the malicious node will be replaced by a normal node in the network. Operating in the range of 1000mx800m , network nodes are simulated in the following cases: stationary nodes , randomly moving with moving objects . change it to 2m/s, 8m/s, 15m / s , respectively , according to the random motion model .

Simulation protocol is AODV, simulation time 200s , broadcast area 250m , queue FIFO , has 10 UDP connections , CBR source , packet size 1000bytes , node standing at center location (1000 , 1000) and perform the attack behavior of the RREQ packet starting at 10 seconds , the first UDP transmitter starts at the 0th second . Details of simulation parameters are concatenated in Table 1.

Table 1. Simulation parameters on OMNeT++

Parameter	Value
Geographical area	1000mx800m
Simulation time	200s
Total number of network nodes	20 (1 malicious node)
Movement speed (m/s)	0, 2, 8, 15
Packet Size	1000(bytes)
Queue	FIFO
Routing protocol	AODV
Starts attack at seconds	ten
Distance	0.2

2. Simulation results

Run the simulation after completing the thesis, focusing on evaluating the following parameters:

- Evaluate the PDR parameter (successful packet transfer rate). With the PDR parameter will know the reliability of the routing protocol:

$$\text{PDR} = \frac{\text{Total number of packages received}}{\text{Total number of conversion packages}} * 100\%$$

- ETE parameter evaluation (terminal delay). With the ETE parameter, it will know the average value of the delay in the successful transmission of the packet.

- Evaluate parameters of Throughput (throughput). This is the amount of information sent per unit of time.

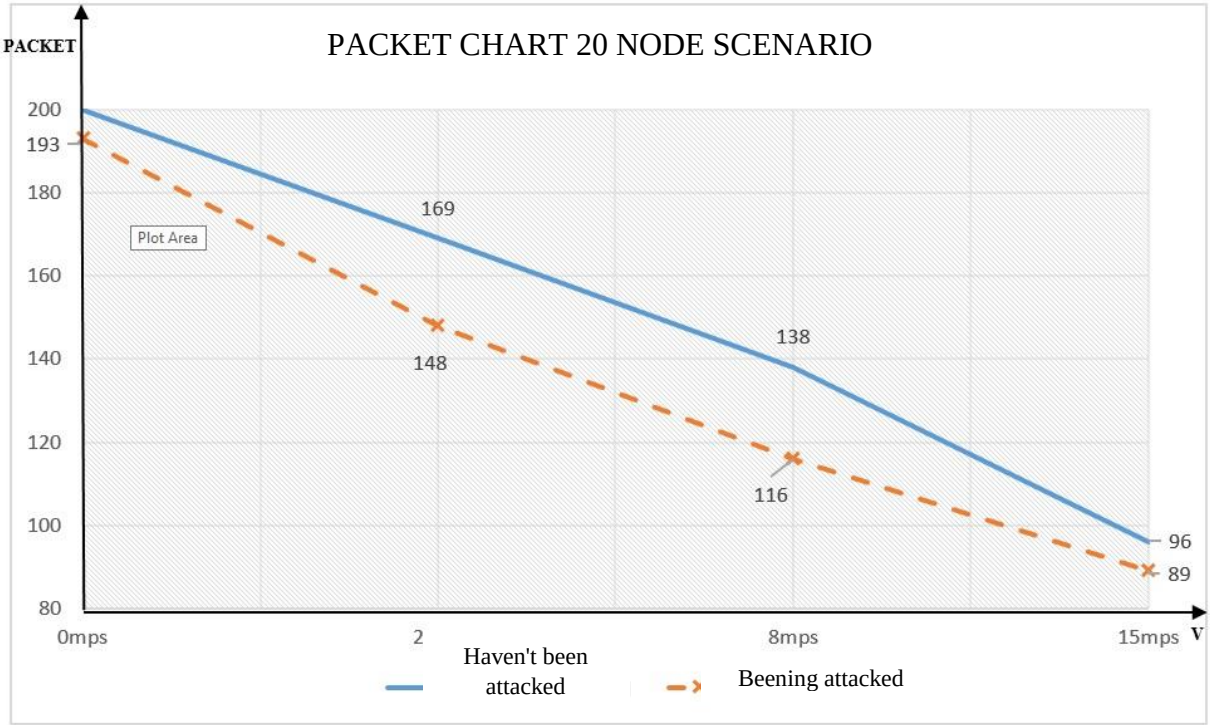
- The simulation results in a network environment where there are no malicious nodes attacking and having attack behavior are shown in Tables 2, 3, 4 and charts 1, 2, 3 show:

+ The successful packet transfer when there is no malicious behavior at 0m/s is 100%, and the successful rate of packet transfer is reduced when there is an attack of the malicious node to 96.5%. For the case of moving nodes, depending on the speed of movement, the faster the moving speed, the greater the number of packet drops, the more this process increases when there is a malicious node's attack. Details of the parameters are summarized in Table 2.

Table 2. Result of Packet Scenario Parameter Table 20 Node

Parameter	**Result parameters**			
Node receiver	0mps	2mps	8mps	15mps
Haven't been attacked	200 (100%)	169 (84.5%)	138 (69%)	96 (48%)
Being attacked	193 (96.5%)	148 (74%)	116 (58%)	89 (44.5%)

Chart 1. Parameter Packet script table 20 nodes

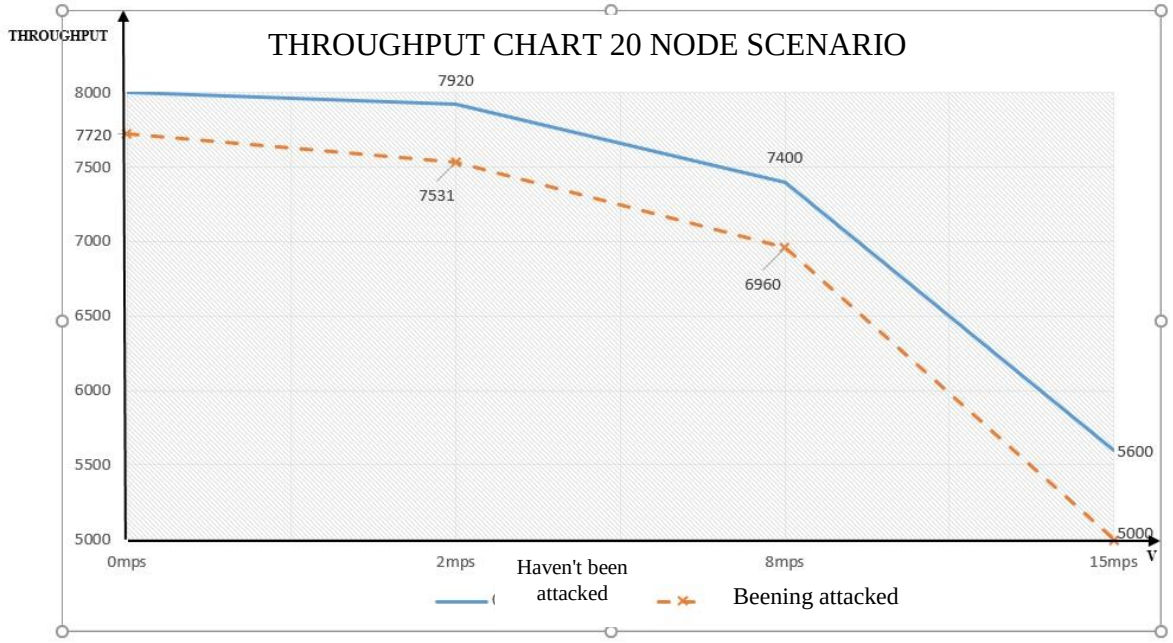


The throughput parameter (Throughput) is also reduced when there is attack behavior of malicious node at 0m / s and random motion with moving object . change it to 2m/s, 8m/ s , 15m/s , respectively , according to the random motion model .

Table 3. Throughput parameter results of the 20-node script table

Parameter	**Result parameters**			
Node receiver	0mps	2mps	8mps	15mps
Haven't been attacked	8000	7920	7400	5600
Being attacked	7720	7531	6960	5000

Chart 2. Throughput parameter table 20 node

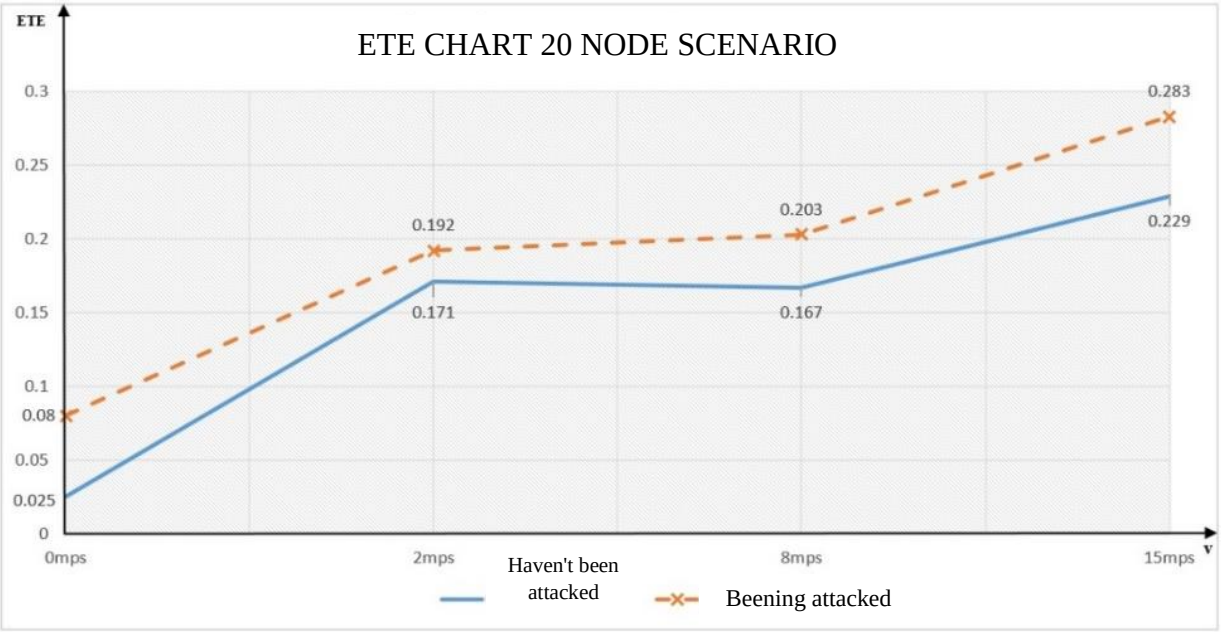


- to - end latency parameter is more inclined to increase when there is a malicious node's attack behavior at 0m/ s and random motion to the node . c move to n change it to 2m / s , 8m / s, respectively , according to the random motion model .

Table 4. Parameter Results Statistical Output Delay of 20 nodes . script

Parameter	**Result parameters**			
Node receiver	0mps	2mps	8mps	15mps
Haven't been attacked	0.009	0.066	0.182	0.305
Being attacked	0.011	0.074	0.182	0.197

Chart 3. 20 node script terminal delay parameter



IV. CONCLUDE

Thus , the paper presented building a malicious attack node that performs RREQ packet flooding attack . in the AODV routing protocol . Result _ _ _The results show that when attacked , the parameters have a significant change, especially the successful rate of packet transmission is low when there is a malicious node's attack .

In the future , we will continue to install several emulators to compare the success rate of packet transfers in the presence of malicious node attacks .

REFERENCES

[1]. Ngo The Hai Anh, "Evaluation of routing protocol security in MANET", master thesis of information technology, (2016).

[2]. Luong Thai Ngoc, Vo Thanh Tu, " Solutions to improve aodv protocol to reduce the harmful effects of flood attack on mannet", Science and technology magazine, volume 11 No. 1, (2018) .

[3]. Mai Cuong Tho, "NTU-ATTACK is a custom mobile wireless network attack simulation platform on OMNeT++", (2021).

[4]. E. Alotaibi and B. Mukherjee, "A survey on routing algorithms for wireless Ad-Hoc and mesh networks", Computer Networks, vol. 56, no. 2, pp. 940–965, 2012

[5]. T. Cholez, C. Henard, I. Chrisment, O. Festor, G. Doyen, and R. Khatoun, "A first approach to detect suspicious peers in the KAD P2P network", SAR-SSI Proceedings, pp. 1–8, 20

[6]. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules be tween sets of items in large databases. In Proceedings of the ACM SIG- MOD Conference on Management of Data, pages 207-216, 1993.

[7]. S. Desilva and RV Boppana, "Mitigating malicious control packet floods in ad hocnetworks," in IEEE Wireless Communications and Networking Conference, WCNC , 2005, vol. 4, pp. 2112–2117, (2005).

[8]. Mahmoud Abu Zant and Adwan Yasin, "Avoiding and Isolating Flooding Attack by Enhancing AODV MANET Protocol (AIF_AODV)", Research Article, Hindawi Security and Communication Networks Volume 2019, Article ID 8249108, 12 pages https://doi.org/10.1155/2019/8249108 , (2019).

[9]. Teklay Gebremichael, **"** Preventing Flooding Attack in MANETs using the reserved bits of AODVmessages", thesis masters of Science in Computer Engineering, (2014).