Enhanced AODV: Routing Protocol based on Packet Size for Wireless Mobile Networks

D. Mohammad Abdulla , G. Tejdeep Reddy , G. Mukesh Venkata Sai , S. Srihemanth , M. Rajesh
Department of Computer Science and Engineering, Amrita School of Computing,
Bengaluru, Amrita Vishwa Vidyapeetham, India .
BL.EN.U4AIE21044@bl.students.amrita.edu, BL.EN.U4AIE21048@bl.students.amrita.edu,
BL.EN.U4AIE21050@bl.students.amrita.edu, BL.EN.U4AIE21123@bl.students.amrita.edu, m_rajesh@blr.amrita.edu.

Abstract**-- a new Ad Hoc On-Demand Distance Vector (AODV) routing protocol within the context of the NS2 simulation environment. Improving wireless network routing efficiency is the goal of this work. The modification entails combining (or integrating) an existing packet size-based routing mechanism with the AODV protocol. Upon receiving a packet, The routing method determines the size of a packet in bytes upon reception, sending larger packets via the optimal route and smaller packets via the next best route. By allocating packets according to their size, this modification seeks to increase network traffic efficiency and, consequently, communication efficiency overall. The paper discusses In addition to presenting experimental findings and discussing the implementation of the modified protocol, the paper also simulates a wireless network with six mobile nodes (n0-n5) and highlights the potential benefits of the proposed adaptation over the standard AODV protocol for handling different packet sizes in ad hoc networks.***

Key Words— AODV protocol, Routing efficiency, Ad hoc networks, Packet size , Routing, mobile nodes, Wireless networks, NS2 , Routing protocols, Network traffic, Communication

I. INTRODUCTION

In wireless ad hoc networks, the Ad Hoc On-Demand Distance Vector (AODV) routing technique is crucial for creating effective communication routes between mobile nodes. In order to provide responsive and loop-free routing, the current AODV uses a routing algorithm based on hop count and sequence numbers. But as network traffic has changed, more sophisticated methods are needed to handle the range of packet sizes.

In this context, we aim to improve the current AODV protocol by integrating a packet size-based routing system. Although efficient, the current AODV is unable to optimize network traffic according to packet sizes. By dynamically routing packets based on their size, our innovation circumvents this barrier by sending larger packets along the optimal route and smaller packets via the next best paths.

This improvement is important because it shows how different packet sizes affect network efficiency in different ways. Higher throughput paths might be advantageous for larger packets, whilst lower throughput paths might be used for smaller packets. Our goal with AODV is to increase the overall communication efficiency in wireless networks by integrating packet size-based routing.

This paper presents The conceptual foundation and implementation specifics of our suggested AODV protocol modification are presented in this study. It also includes simulations carried out on a wireless network with six nodes (n0-n5) in order to verify the effectiveness and benefits of this modified routing strategy. The outcomes of these simulations are examined and contrasted with the state-of-the-art AODV routing, emphasizing the possible improvements provided by our modification in handling varying packet sizes in ad hoc networks.

Section II of this work shows works that are related to this field. Section III shows the implementation of the design and the results are shown in the Section IV. Section V provides conclusion of the project.

II. RELATED WORKS

[1] The AODV routing protocol in VANETs was modified by Afsana Ahamed and Hamid Vakilzadian for research purposes in order to lower packet loss and end-to-end latency. In order to reduce the amount of RREQ and RREP messages and eventually enhance performance, the study included direction parameters and implemented a two-step filtering method. After using the updated AODV in VANETs, the simulation results showed a decrease in packet loss, an increase in throughput, and an end-to-end delay.

[2] Using NS-2 simulation, Rani A. and Dave M. presented and assessed Modified AODV for Load Balancing. By altering the number of sources and queue length, they compared modified-AODV with conventional AODV to find the NRL, average latency, and average throughput. The simulation findings demonstrated that by monitoring the aggregate interface queue length, the adjustments were able to decrease normalized routing burden and increase average throughput. In high mobility networks, however, the AODV adjustments are more beneficial for traffic that is moderately laden. As with earlier research, this study did not address network performance during an attack.

[3] A study was proposed by Pedro M. Ruiz and Francisco J. Ros to create a new Manet unicast routing protocol in NS2. It is intended for people who are accustomed to NS2 simulations and wish to create custom protocols. Topics

such establishing packet types, handling routing tables, putting the routing agent into practice, and building the required files are all covered in the guide.

[4] In This paper We examined the design options for an AODV implementation in this article. First, we determined which unsupported events AODV needs in order to conduct routing. Next, we looked at the benefits and drawbacks of three different approaches to get this data. Our choice to combine minimal kernel modules with a user-space daemon was validated by this investigation. At last, we showcased the layout of numerous AODV implementations that are accessible to the public. We expect that the data in this work will help scholars better understand the trade-offs involved in developing ad hoc routing protocols. In addition, the features and design structure descriptions of each implementation let consumers determine which one best suits their requirements.

[5] The performance of the improved AODV's performance in various circumstances. Humaira Nishat et al. evaluated the RAODV protocol, a modified version of the AODV protocol. and the standard AODV was used to compare its performance. R-AODV improved AODV routing performance and reduced RREQ transmission by using a multi-reverse path. Based on NS-2 simulation results, RAODV outperforms AODV in terms of throughput, average delay, and packet delivery ratio by modifying nodes velocity. However, the scope of this examination was restricted to performance in the absence of an assault and during regular business hours.

[6] A study on an altered AODV routing algorithm for wireless sensor networks (WSNs) was proposed by MOHITANGURALA, MANJUBALA, AND SUPHVINDER SINGH BAMBER. The protocol's goal is to increase network lifetime by lowering delay and energy consumption.The protocol has the potential to increase performance in WSN applications, as the authors show by proposing a set of formulas and mathematical models to examine the energy consumption and remaining energy of nodes in the network.With its clear explanations and examples, the well-structured paper aids readers in understanding the topics and research methodologies.

[7] The assessment of DSDV, AODV, and DSR routing algorithms in mobile ad hoc networks is covered in "A study for Different Environments," which was proposed by Ahmed Muhi Shantaf, Sefer Kurnaz, and Alaa Hamid Mohammed. Packet Delivery Ratio (PDR), Average Throughput, and Average End-to-End Delay are its three main performance measures. The authors examine how mobility affects these protocols in two distinct contexts: various node density and geographical locations. The presentation and discussion of the simulation findings sheds light on how well the three methods operate in different scenarios.

[8] This paper Nisarg Gandhewar Thus we have evaluated the performance of very popular on demand routing protocol AODV, by means of various performance metrics such as PDR, end to end delay & packet loss, as well obtained simulation results by varying number of nodes in the network & found that there is non linear change in the values of these metrics also we realized working & control massages involved in AODV protocol.

[9] This paper evaluated the performance of AODV, DSDV and DSR, routing protocols for MANET using NS2 simulator. Comparison was based on variety of performances metrics, namely energy consumption. From results reports in section 3 we calculated that DSR protocol is the best in terms of packet delivering ratio than other protocols making it suitable for highly mobile random networks.

[10] This paper assesses and contrasts the effectiveness of Ad-hoc On-demand Distance Vector (AODV) and Optimized Link State Routing (OLSR), two widely used mobile ad-hoc network (MANET) routing techniques. The authors use parameters such as packet delivery ratio, delay, throughput, overhead, etc. to simulate different scenarios with varying node densities, mobilities, and network loads in NS-2. They discover that reactive AODV is more suited for use cases requiring high throughput, whereas proactive OLSR works better for applications that are sensitive to delays. The study expands on previous research on MANET protocol analysis by addressing additional parameters. Aspects of scalability and security are not examined, though. Developers can use the data to select the best protocol for their application's requirements.

[11] This paper examines and contrasts the effectiveness of three widely used routing protocols for vehicular ad hoc networks (VANETs): AODV, DSDV, and DSR. In NS-2, a 7-node simulation scenario is developed. Critical performance indicators such as packet delivery ratio, throughput, and end-to-end delay are assessed by the writers. Based on the traffic characteristics and topology provided, the reactive protocol AODV performs the best overall, according to the results. The work expands on previous research on VANET protocol analysis by using a thorough simulation-based assessment. But just one situation is taken into account. Experiments with other network sizes and mobility models can yield more information. The results will assist developers in choosing the best routing protocol for their VANET application needs.

[12] This paper extend the lifetime of networks in ad hoc networks, this study suggests modifying the AODV routing protocol. In addition to hop count, the routing measure takes into account node mobility, load, and remaining energy. Compared to normal AODV, the enhanced AODV increases network lifetime by an average of 35%, according to simulations conducted in NS-2. By changing the pause time and number of connections, the analysis is carried out. The main innovation is choosing routes with energy awareness to balance load and increase longevity. The paper offers a low-cost, high-impact method for enhancing AODV performance.

[13] This paper OLSR proactive routing protocol and the AODV reactive routing protocol are two Mobile Ad hoc Networks protocols that are realistically compared and their

performance analyzed in this study. This comparative analysis was carried out by adjusting the network load with a fixed transmission gate and by using a variety of measurement parameters, including throughput, packet delivery fraction, and end-to-end delay, depending on the size of the network. It is evident that AODV consumes less resources than OLRS and performs better in networks with static traffic. Furthermore, the OLSR protocol performs better in traffic networks with high densities and irregular traffic patterns. In fact, our scenario's simulation findings support the routing strategies' improved performance in terms of optimization, dependability, and quality of service.

III. DESIGN

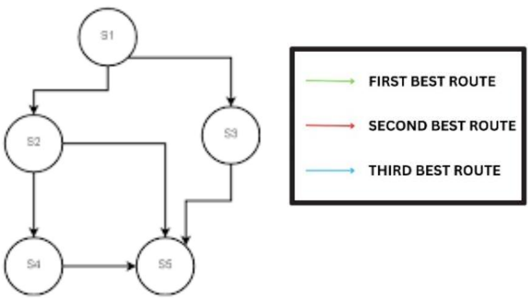


Fig-1 - Design

Now let us take a case. We have 5 mobile nodes namely(S1-S5).Now S1 has 3 pakets(namely a,b & c) and it wants to send those packets to n4.Now using distance vector we found green is the best route followed by red and then green. We have a threshold length of x.we found that packet length of a is less than x. b lies in between x and c.Now from our design it takes c and sends it through green path(best path).The next largest packet b is sent through red path.The smallest packet a is sent in the blue path.

I.

IV. IMPLEMENTATION

aodv.h : This is the header file where will be defined all necessary timers (if any) and routing agent which performs protocol's functionality.
aodv.cc : In this file are actually implemented all timers, routing agent and Tcl hooks.
aodv_packet.h: Here are declared all packets Acnproject protocol needs to exchange among nodes in the manet.
cmu-trace.cc : Routing table implementation

Changes made in '_pkt.h' file:

- If we see that for our protocol to route packets based on the length of packets, we have to include a variable named as **pkt_len_** in the pkt.h file.
    - Below are the changes made:

```
nsaddr_t pkt_src_;
u_int16_t pkt_len_;
u_int8_t pkt_seq_num_;
```

- Moreover, we have to include the pointer mentioning the reference for our pkt_len_ variable in the **cmu_trace.cc** file.
- Cmu_trace.cc file contains all the information and pattern to be printed in the trace file.
    - Below is the modification for that:

```
sprintf(pt_->buffer() + offset,
"-P aodv -Po %d -Ps %d -Pl %d",
ph->pkt_src(),
ph->pkt_seq_num(),
ph->pkt_len());
}
```

- This ensures that, our protocol is going to route the packets according to the length of packet (in bytes).

Changes made to aodv_packet.h

```
struct hdr_aodv {
        u_int8_t        ah_type;
        /*
        u_int8_t        ah_reserved[2];
        u_int8_t        ah_hopcount;
        */
            // Header access methods

        nsaddr_t pkt_src_;       //Node which originated this packet
        u_int16_t pkt_len_;    //Packet length (in bytes)
        u_int8_t pkt_seq_num_; //Packet Sequence Number


        inline nsaddr_t& pkt_src() { return pkt_src_;}
        inline u_int16_t& pkt_len() { return pkt_len_;}
        inline u_int8_t& pkt_seq_num() { return pkt_len_;}

        static int offset_; // required by PacketHeaderManager
        inline static int& offset() { return offset_; }
        inline static hdr_aodv* access(const Packet* p) {
                return (hdr_aodv*) p->access(offset_);
        }
};
```

Changes made to aodv.h

```
#define HELLO_INTERVAL          1               // 1000 ms
#define ALLOWED_HELLO_LOSS      3               // packets
#define BAD_LINK_LIFETIME       3               // 3000 ms
#define MaxHelloInterval        (1.25 * HELLO_INTERVAL)
#define MinHelloInterval        (0.75 * HELLO_INTERVAL)

#define JITTER                  (Random::uniform()*0.5)
```

```
void        forward(andv_rt_entry *rt, Packet *p, double delay);
void        sendHello(void);
void        sendRequest(nsaddr_t dst);

void        sendReply(nsaddr_t ipdst, u_int32_t hop_count,
                      nsaddr_t rpdst, u_int32_t rpseg,
                      u_int32_t lifetime, double timestamp);
void        sendError(Packet *p, bool jitter =true);
void        send_aodv_pkt(void);
```

Changes made to aodv.cc

```
/*
   send aodv
*/

void AODV::send_aodv_pkt(){
        Packet* p = allocpkt();
        struct hdr_cmn* ch = HDR_CMN(p);
        struct hdr_ip* ih = HDR_IP(p);
        struct hdr_aodv* ph = HDR_AODV(p);

        ph->pkt_src() = index;
        ph->pkt_seq_num() = seqno++;
        ch->ptype() = PT_AODV;
        ch->directional() = hdr_cmn::DOWN;
        ch->size() = IP_HDR_LEN + ph->pkt_len();
        ch->error() = 0;
        ch->next_hop() = IP_BROADCAST;
        ch->addr_type() = NS_AF_INET;;


        ih->saddr() = index;
        ih->daddr() = IP_BROADCAST;
        ih->sport() = RT_PORT;
        ih->dport() = RT_PORT;
        ih->ttl() = IP_DEF_TTL;

        Scheduler::instance().schedule(target_, p, JITTER);


}
```

Changes made to cmu-trace.cc

```
void
CMUTrace::format_aodv(Packet *p, int offset)
{
        struct hdr_aodv *ah = HDR_AODV(p);
        struct hdr_aodv_request *rq = HDR_AODV_REQUEST(p);
        struct hdr_aodv_reply *rp = HDR_AODV_REPLY(p);


        switch(ah->ah_type) {
        case AODVTYPE_RREQ:

                if (pt_->tagged()) {
                    sprintf(pt_->buffer() + offset,
                            "-aodv:t %x -aodv:h %d -aodv:b %d -aodv:d %d "
                            "-aodv:ds %d -aodv:s %d -aodv:ss %d "
                            "-aodv:c REQUEST ", "-aodv:O %d -aodv:s %d -aodv:l %d",
                            rq->rq_type,
                             rq->rq_hop_count,
                             rq->rq_bcast_id,
                             rq->rq_dst,
                             rq->rq_dst_seqno,
                             rq->rq_src,
                             rq->rq_src_seqno);
                } else if (newtrace_) {

                    sprintf(pt_->buffer() + offset,
                        "-P aodv -Pt 0x%x -Ph %d -Pb %d -Pd %d -Pds %d -Ps %d -Pss %d -Pc REQUEST "
                        rq->rq_type,
                         rq->rq_hop_count,
                         rq->rq_bcast_id,
                         rq->rq_dst,
                         rq->rq_dst_seqno,
                         rq->rq_src,
                         rq->rq_src_seqno);


                  } else {

                      sprintf(pt_->buffer() + offset,
                          "[0x%x %d %d [%d %d] [%d %d]] (REQUEST)",
                          rq->rq_type,
                           rq->rq_hop_count,
                           rq->rq_bcast_id,
                           rq->rq_dst,
                           rq->rq_dst_seqno,
                           rq->rq_src,
                           rq->rq_src_seqno);
                  }
```

TCL

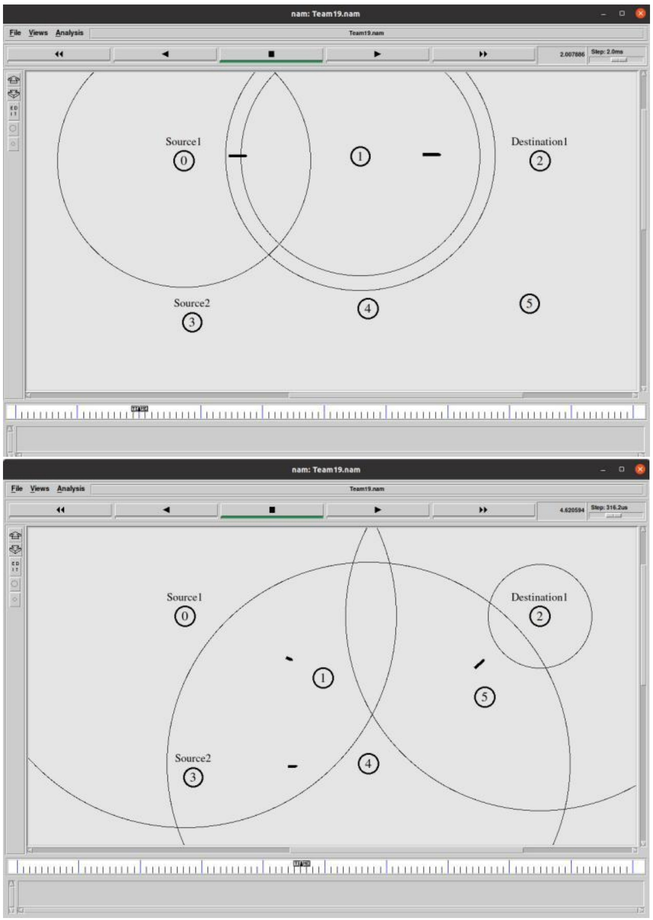
```
set val(chan)   Channel/WirelessChannel    ;# channel type
set val(prop)   Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)  Phy/WirelessPhy            ;# network interface type
set val(mac)    Mac/802_11                 ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue    ;# 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)      1028                      ;# X dimension of topography
set val(y)      760                      ;# Y dimension of topography
set val(stop)   10.0                           ;# time of simulation end

#==================================
#       Initialization
#==================================
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
set tracefile [open Team19.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open Team19.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel

#===================================
#     Mobile node parameter setup
#===================================
$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) \
                -channel       $chan \
                -topoInstance  $topo \
                -agentTrace    ON \
                -routerTrace   ON \
                -macTrace      ON \
                -movementTrace ON

#===================================
#        Nodes Definition
#===================================
# Create 6 nodes
set n0 [$ns node]
$n0 set X_ 562
$n0 set Y_ 618
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 743
$n1 set Y_ 623
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20
set n2 [$ns node]
$n2 set X_ 928
$n2 set Y_ 618
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20
set n3 [$ns node]
$n3 set X_ 570
$n3 set Y_ 452
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20
set n4 [$ns node]
$n4 set X_ 751
$n4 set Y_ 466
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20
```

```
set n5 [$ns node]
$n5 set X_ 917
$n5 set Y_ 471
$n5 set Z_ 0.0
$ns initial_node_pos $n5 20

#===================================
#        Generate movement
#===================================
$ns at 2 "$n1 setdest 690 530 30"
$ns at 2 "$n5 setdest 780 660 30"

#===================================
#        Agents Definition
#===================================
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n2 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500

#Setup a UDP connection
set udp3 [new Agent/UDP]
$ns attach-agent $n3 $udp3
set null5 [new Agent/Null]
$ns attach-agent $n2 $null5
$ns connect $udp3 $null5
$udp3 set packetSize_ 800


#===================================
#        Applications Definition
#===================================
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 10.0 "$ftp0 stop"

#Setup a CBR Application over UDP connection
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp3
$cbr1 set packetSize_ 800
$cbr1 set rate_ 1.0Mb
$cbr1 set random_ null
$ns at 4.0 "$cbr1 start"
$ns at 10.0 "$cbr1 stop"
#===================================
#        Labeling nodes
#===================================
$ns at 0.0 "$n0 label Source1"
$ns at 0.0 "$n3 label Source2SS"
$ns at 0.0 "$n2 label Destination1"

#===================================
#        Set destinations
#===================================
$ns at 100.0 "$n5 setdest 385.0 228.0 5.0"
$ns at 60.0 "$n2 setdest 200.0 20.0 5.0"
$ns at 30.0 "$n3 setdest 115.0 85.0 5.0"
$ns at 45.0 "$n1 setdest 375.0 80.0 5.0"
$ns at 89.0 "$n4 setdest 167.0 351.0 5.0"
$ns at 78.0 "$n0 setdest 50.0 359.0 5.0"

#===================================
#        Color change during movement
#===================================
$ns at 73.0 "$n2 delete-mark N2"
$ns at 73.0 "$n2 add-mark N2 pink circle"
$ns at 124.0 "$n3 delete-mark N11"
$ns at 124.0 "$n3 add-mark N11 purple circle"
$ns at 87.0 "$n4 delete-mark N26"
$ns at 87.0 "$n4 add-mark N26 yellow circle"
$ns at 92.0 "$n1 delete-mark N14"
$ns at 92.0 "$n1 add-mark N14 green circle"


#===================================
#        Termination
#===================================
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam Team19.nam &
    exit 0
}
for {set i 0} {$i < $val(nn) } { incr i } {
    $ns at $val(stop) "\$n$i reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

V. RESULTS

```
s 1.000000000 _0_ AGT  --- 0 tcp 40 [0 0 0 0] ------- [0:0 2:0 32 0] [0 0] 0 0
r 1.000000000 _0_ RTR  --- 0 tcp 40 [0 0 0 0] ------- [0:0 2:0 32 0] [0 0] 0 0
s 1.000000000 _0_ RTR  --- 0 AODV 48 [0 0 0 0] ------- [0:255 -1:255 30 0] [0x2 1 1 [2 0] [0 4]] (REQUEST)
s 1.000115000 _0_ MAC  --- 0 AODV 106 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0x2 1 1 [2 0] [0 4]] (REQUEST)
r 1.000963554 _3_ MAC  --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0x2 1 1 [2 0] [0 4]] (REQUEST)
r 1.000963604 _1_ MAC  --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0x2 1 1 [2 0] [0 4]] (REQUEST)
r 1.000963808 _4_ MAC  --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0x2 1 1 [2 0] [0 4]] (REQUEST)
r 1.000988554 _3_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0x2 1 1 [2 0] [0 4]] (REQUEST)
r 1.000988604 _1_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0x2 1 1 [2 0] [0 4]] (REQUEST)
r 1.000988808 _4_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0x2 1 1 [2 0] [0 4]] (REQUEST)
s 1.001869781 _3_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] ------- [3:255 -1:255 29 0] [0x2 2 1 [2 0] [0 4]] (REQUEST)
s 1.002104781 _3_ MAC  --- 0 AODV 106 [0 ffffffff 3 800] ------- [3:255 -1:255 29 0] [0x2 2 1 [2 0] [0 4]] (REQUEST)
r 1.002953335 _0_ MAC  --- 0 AODV 48 [0 ffffffff 3 800] ------- [3:255 -1:255 29 0] [0x2 2 1 [2 0] [0 4]] (REQUEST)
r 1.002953386 _4_ MAC  --- 0 AODV 48 [0 ffffffff 3 800] ------- [3:255 -1:255 29 0] [0x2 2 1 [2 0] [0 4]] (REQUEST)
r 1.002953592 _1_ MAC  --- 0 AODV 48 [0 ffffffff 3 800] ------- [3:255 -1:255 29 0] [0x2 2 1 [2 0] [0 4]] (REQUEST)
r 1.002978335 _0_ RTR  --- 0 AODV 48 [0 ffffffff 3 800] ------- [3:255 -1:255 29 0] [0x2 2 1 [2 0] [0 4]] (REQUEST)
r 1.002978386 _4_ RTR  --- 0 AODV 48 [0 ffffffff 3 800] ------- [3:255 -1:255 29 0] [0x2 2 1 [2 0] [0 4]] (REQUEST)
r 1.002978592 _1_ RTR  --- 0 AODV 48 [0 ffffffff 3 800] ------- [3:255 -1:255 29 0] [0x2 2 1 [2 0] [0 4]] (REQUEST)
s 1.005400337 _1_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] ------- [1:255 -1:255 29 0] [0x2 2 1 [2 0] [0 4]] (REQUEST)
s 1.005715337 _1_ MAC  --- 0 AODV 106 [0 ffffffff 1 800] ------- [1:255 -1:255 29 0] [0x2 2 1 [2 0] [0 4]] (REQUEST)
r 1.006563861 _4_ MAC  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 29 0] [0x2 2 1 [2 0] [0 4]] (REQUEST)
r 1.006563941 _0_ MAC  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 29 0] [0x2 2 1 [2 0] [0 4]] (REQUEST)
r 1.006563954 _2_ MAC  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 29 0] [0x2 2 1 [2 0] [0 4]] (REQUEST)
r 1.006564108 _5_ MAC  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 29 0] [0x2 2 1 [2 0] [0 4]] (REQUEST)
r 1.006564148 _3_ MAC  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 29 0] [0x2 2 1 [2 0] [0 4]] (REQUEST)
r 1.006588861 _4_ RTR  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 29 0] [0x2 2 1 [2 0] [0 4]] (REQUEST)
r 1.006588941 _0_ RTR  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 29 0] [0x2 2 1 [2 0] [0 4]] (REQUEST)
r 1.006588954 _2_ RTR  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 29 0] [0x2 2 1 [2 0] [0 4]] (REQUEST)
s 1.006588954 _2_ RTR  --- 0 AODV 44 [0 0 0 0] ------- [2:255 0:255 30 1] [0x4 1 [2 4] 10.000000] (REPLY)
r 1.006589108 _5_ RTR  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 29 0] [0x2 2 1 [2 0] [0 4]] (REQUEST)
r 1.006589148 _3_ RTR  --- 0 AODV 48 [0 ffffffff 1 800] ------- [1:255 -1:255 29 0] [0x2 2 1 [2 0] [0 4]] (REQUEST)
s 1.006883954 _2_ MAC  --- 0 ARP 86 [0 ffffffff 2 806] ------- [REQUEST 2/2 0/1]
r 1.007572446 _5_ MAC  --- 0 ARP 28 [0 ffffffff 2 806] ------- [REQUEST 2/2 0/1]
r 1.007572571 _1_ MAC  --- 0 ARP 28 [0 ffffffff 2 806] ------- [REQUEST 2/2 0/1]
r 1.007572732 _4_ MAC  --- 0 ARP 28 [0 ffffffff 2 806] ------- [REQUEST 2/2 0/1]
s 1.007722571 _1_ MAC  --- 0 RTS 44 [52e 2 1 0]
r 1.008075188 _2_ MAC  --- 0 RTS 44 [52e 2 1 0]
s 1.008085188 _2_ MAC  --- 0 CTS 38 [3f4 1 0 0]
r 1.008389805 _1_ MAC  --- 0 CTS 38 [3f4 1 0 0]
s 1.008399805 _1_ MAC  --- 0 ARP 86 [13a 2 1 806] ------- [REPLY 1/1 2/2]
s 1.008980363 _4_ RTR  --- 0 AODV 48 [0 ffffffff 0 800] ------- [4:255 -1:255 29 0] [0x2 2 1 [2 0] [0 4]] (REQUEST)
r 1.009088422 _2_ MAC  --- 0 ARP 28 [13a 2 1 806] ------- [REPLY 1/1 2/2]
s 1.009098422 _2_ MAC  --- 0 ACK 38 [0 1 0 0]
r 1.009403039 _1_ MAC  --- 0 ACK 38 [0 1 0 0]
s 1.009612422 _2_ MAC  --- 0 RTS 44 [5ae 1 2 0]
s 1.009679898 _5_ RTR  --- 0 AODV 48 [0 ffffffff 1 800] ------- [5:255 -1:255 28 0] [0x2 3 1 [2 0] [0 4]] (REQUEST)
r 1.009965039 _1_ MAC  --- 0 RTS 44 [5ae 1 2 0]
s 1.009975039 _1_ MAC  --- 0 CTS 38 [474 2 0 0]
r 1.010279656 _2_ MAC  --- 0 CTS 38 [474 2 0 0]
s 1.010289656 _2_ MAC  --- 0 AODV 102 [13a 1 2 800] ------- [2:255 0:255 30 1] [0x4 1 [2 4] 10.000000] (REPLY)
r 1.011106273 _1_ MAC  --- 0 AODV 44 [13a 1 2 800] ------- [2:255 0:255 30 1] [0x4 1 [2 4] 10.000000] (REPLY)
s 1.011116273 _1_ MAC  --- 0 ACK 38 [0 2 0 0]
```

Trace file



Output nam file

VI. Conclusion

In conclusion, This study developed a packet size-based routing mechanism, which significantly improved the AODV routing protocol. The change was made with the intention of maximizing wireless network communication efficiency by allocating packets according to their sizes.

After careful adjustments in the NS2 simulation environment and six mobile node simulations, our modified protocol demonstrated encouraging gains in handling various packet sizes. Although the study shows potential benefits in terms of throughput and communication efficiency, it also emphasizes the need for further improvement, practical deployment scenarios, enhanced security, and possible integration with future networking protocols. This improvement provides prospects for more effective and flexible communication in dynamic network situations by laying the groundwork for the advancement of routing systems in wireless networks.

VII. REFERENCES

[1] N. A. Ahmed, K. Saghar, R. Ahmed, and A. K. Kiani, ‘‘Achieving energy efficiency through Load Balancing: A comparison through formal verification of two WSN routing protocols,’’ in Proc. 13th Int. Bhurban Conf. Appl. Sci. Technol. (IBCAST), 2016.

[2] B. T, Rajkumar P. Sreedharan, Rajesh M, and M. Nithya, “Leader-follower co-ordination of multiple robots with obstacle avoidance”, International Conference On Smart Technologies For Smart Nation (SmartTechCon2017). Reva University, Bengaluru , 2017.

[3] P. Singh, M. Doja, and B. Alam, ‘‘Energy efficient self-organizing Load Balanced clustering scheme for heterogeneous WSN,’’ in Proc. Int. Conf.Energy Econ. Environ. (ICEEE), 2015.

[4] V. N. V. Aditya Sharma and Rajesh M, “Building a quadcopter: An approach for an Autonomous Quadcopter”, 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI). 2018.

[5] B. Sreevidya and Rajesh M, “False Data Injection Prevention in Wireless Sensor Networks using Node-level Trust Value Computation”, 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI). 2018.

[6] S. S. Chauhan and M. M. Gore, ‘‘Balancing energy consumption across network for maximizing lifetime in cluster-based wireless sensor network,’’ CSI Trans. ICT, vol. 3, nos. 2–4, pp. 83–90, Dec. 2015.

[7] S. R. Barath and C. K. S. Vijla, ‘‘Mitigating packet drop and energy consumption in wireless sensor networks using network level optimized relay election protocol,’’ Wireless Pers. Commun., vol. 94, no. 4, pp. 2783–2796, Jun. 2017.

[8] L. Wang and S. Olariu, "A Two-Zone Hybrid Routing Protocol for Mobile Ad Hoc Networks", IEEE transactions on Parallel and distributed systems, December 2004 (Vol. 15, No. 12, 2004, pp.1105-1116.

[9] C. Perkins, E. Belding-Royer, S. Das, Ad hoc OnDemand Distance Vector (AODV) Routing, RFC3561, July: 2003

[10] Xinsheng Wang, Qing Liu and Nan Xu, The EnergySaving Routing Protocol Based on AODV, Fourth International Conference on Natural Computation 2008

[11] Said Khelifa and Zoulikha Mekkakia Maaza, “An Energy Multi- path AODV Routing Protocol in Ad Hoc Mobile Networks”, IEEE 2010.

[12] Zhi Li and Yu-Kwong Kwok, "A New Multipath Routing Approach to Enhancing TCP Security in Ad Hoc Wireless Networks" in Proc. ICPPW 2005.

[13] Chonggun Kim, Elmurod Talipov, and Byoungchul Ahn, "A Reverse AODV Routing Protocol in Ad Hoc Mobile Networks" , LNCS 4097, pp. 522 531,2006.