

Routing Performance Analysis

[[Trace](#) | [Routing Overhead](#) | [Packet Delivery Ratio](#) | [Route Length](#) | Back to [Network Simulator 2 for Wireless](#)]

A Typical DSR routing trace format:

s 606.210364161 _39_ RTR --- 1306 DSR 44 [13a a 27 800] ----- [39:255 8:255 255 8] 2 [0 0] [0 0 0 0->0] [1 1 8 39->10]

s: means send

606.210364161: time stamp

39: node id

RTR: means router message

1306: uid of this packet

DSR: DSR agent

44: size in the common header `hdr_cmn()`

[13a a 27 800] MAC detail: 13a: means the expected transmission time (note that packet size is large, 44 bytes, 314second?)

a: means the receiving node: 10

27: means the sending node is 39

800: IP header: 0x0800, (ETHERTYPE_ARP is 0x0806)

[39:255 8:255 255 8] IP detail: src address: IP 39 means 0.0.0.39

port 255

dst address: IP 8 means 0.0.0.8

port 255

TTL: 255

Next-hop: 8

2 [0 0] [0 0 0 0->0] [1 1 8 39->10] DSR detail:

2: `num_addrs()`

[0 0] route-request option, this is not a route request, the second 0 is labeled for sequence number

[0 0 0 0->0] route-reply option: [" route-reply?"

"Rreq seqno" "reply length" "dst of src route", "src of the src route"]

[1 1 8 39->10], 1: shows this is a route error

1: number of route errors

8: tp notify node 8.

39->10: link 39-10 is broken

Routing Overhead

How to define Routing Overhead? If all the routing packets no matter broadcasting or unicasting per -hop should be count once. There are some options:

1. The total number of routing packets, counted once per hop
2. The total number of routing bytes, counted once per hop
3. The # of routing packets, count with sequence number, this means end-to-end, not calculated by per-hop basis.

To calculate the number of DSR packets in method 1

```
$ cat out.tr | grep "DSR" | wc -l
```

The result is 3301 dsr packets for a 1000 seconds, 50 nodes 10 connections, 4pkt/sec, 512B size, 670X670 area, mobility speed is at most 20m/s and the average pause time is 600 seconds.

However, this is not true. because both send and rcv are included.

```
$cat out.tr |grep "^s.*DSR" | wc -l
```

shows only 514 DSR packets are sending.

Finally, use this awk code to count how many bytes are used.

```
BEGIN {pktno = 0; byte = 0;}
$1~/s/ && /DSR/ { pktno ++
                byte+=$8 }
```

It shows "544 27016". So only 544 packets and 27016 bytes sent.

With this method, the packet is only count once. But this may be wrong in some sense, because any forwarding packets are not calculate as overhead. The new awkcode should be:

```
BEGIN {pktno = 0; byte = 0;}
$1~/s/ && /DSR/ { pktno ++
                byte+=$8 }
$1~/f/ && /DSR/ { pktno ++
                byte+=$8 }
END { print ( pktno, byte) }
```

This shows: "806 packets 43696 bytes". Because, the concern is the time spent to send routing signaling, this is more accurate way to measure routing overhead.

However, all of the above methods are not fair to compare.

The way to count MAC transmissions is the only correct way to do that.

How many MAC packets are sent for routing purpose and how many MAC packets are sent for traffic.

```
BEGIN {dsrpktno = 0; dsrbyte = 0; cbrpktno = 0; cbrbyte = 0; }
$1~/s/ && /DSR/ && /MAC/ { dsrpktno ++ ;
                        dsrbyte+=$8 ;}
$1~/s/ && /cbr/ && /MAC/ { cbrpktno ++ ;
                        cbrbyte+=$8; }
```

```
END { print ( dsrpktno, dsrbyte , cbrpktno, cbrbyte) }
```

The result show 787 DSR MAC packets totaled as 83568 Bytes, but with data traffic as 4806 MAC packets of 2873836 Bytes.

Note that the scene used in this scenario (/mobility/scene/cbr-50-10-4-512 and scen-670-670....-1) has very short path, all connection are just one or 2 hops, and it is still to see that the routing overhead is huge in packet number comparison as $(787/(787+4806) = 14.07\%)$

Another definition is "**Normalized Routing Overhead**".

Normalized routing load is the number of routing packets transmitted per data packet sent to the destination. Also each forwarded packet is counted as one transmission. This metric is also highly correlated with the number of route changes occurred in the simulation.

The Perl script to calculate this is (thanks to Umut Akyol)

```
#!/usr/bin/perl

$src = out7;

my $rte_overhead = 0;
my $num_rte_pkt = 0;
my $num_data_pkt = 0;

open(INPUTFILE,$src) || die "Cannot open file\n";

$line = <INPUTFILE>;

while ($line ne "") {

@input = split(/ /,$line);

if ($input[0] =~ m/s/ && $input[3] =~ m/RTR/ && $input[7] =~ m/AODV/) {
#print "A routing packet was sent\n";
$num_rte_pkt = $num_rte_pkt + 1;
}

if ($input[0] =~ m/f/ && $input[3] =~ m/RTR/ && $input[7] =~ m/AODV/) {
#print "A routing packet was forwarded\n";
$num_rte_pkt = $num_rte_pkt + 1;
}

if ($input[0] =~ m/s/ && $input[3] =~ m/RTR/ && $input[7] =~ m/cbr/) {
#print "A data packet was sent\n";
$num_data_pkt = $num_data_pkt + 1;
}

if ($input[0] =~ m/f/ && $input[3] =~ m/RTR/ && $input[7] =~ m/cbr/) {
#print "A data packet was forwarded\n";
$num_data_pkt = $num_data_pkt + 1;
}

$line = <INPUTFILE>;
```

```

}
close(INPUTFILE);

if ($num_data_pkt == 0) {
print "No data sent!\n";
} else {
print "Routing overhead is ", $num_rte_pkt/$num_data_pkt, "\n";
}

```

Note that to calculate DATA packets we can both in layer 2 ("MAC") or layer 3 (RTR) from trace file. But if there are some cut-through or fast forward happened in the layer 2, the packet does not go to layer 3. Using RTR will yield wrong statistics.

Packet delivery ratio

The following **AWK** code to get all sending and receiving in Agent level is

```

BEGIN {counter1 = 0; counter2 = 0;}
$1~/s/ && /AGT/ { counter1 ++ }
$1~/r/ && /AGT/ { counter2 ++ }
END { print ( counter1, counter2) }

```

Then execute

```
$awk -f awkcode1 out.tr
```

it shows 2236 2213. Thus, most packets are received.

This is calculated as the total number of packets sent by AGT and received by AGT.

$2213/2236 = 98.97\%$

Average Hop Counts (Path length)

Instead of getting some information from DSR header about path length, we can use a method independent of Routing protocol (DSDV, DSR, AODV). this is to use the number of MAC transmission divided by Agent layer transmission.

So, for above simulation, the avg hop count is :

4806/2236= 2.14