

Bryan's NS-2 Dynamic Source Routing FAQ

This faq is provided "as is", I take no responsibility for anything anyone does using information contained here. Use of any information is at your own risk. The accuracy of the assertions made here is not guaranteed, why, it might all well be fictional. USE AT YOUR OWN RISK.

I decided to write this faq because of what I viewed as the lack of easily available information on the DSR implementation in the [network simulator \(ns-2\)](#). For my Masters in Telecommunications at the University of Limerick, I developed an algorithm to incorporate congestion awareness as part of the route selection process. Doing this I spent a considerable amount of time reading through the various classes in DSR, what their purpose is, how they relate and how to alter their functionality.

To read a paper on this work see the bottom of the page.

This faq mainly deals with programming issues related to the Dynamic Source Routing protocol within ns2.26 (from what I can see DSR in ns2.27, and ns2.28 is identical), much of the information would hold for previous versions of ns and maybe future versions.

Contact Info :

If you want to contact me, I can be reached at ---- [bjhogan\[at\] fastmail\[dot\] fm](mailto:bjhogan@fastmail.fm)

Please include "DSR" in the subject line.

Further information [about me](#) is located at the bottom of this page.

Corrections and additional information are very welcome. **But I am not actively answering questions nor updating this document.**

A big thank you to Miha for making this page finally look a bit nicer and save me the trouble of having to run my scripts.

Last updated February 3rd 2010

1. [What happens to a packet when it is received by DSR?](#)
2. [What happens to the packet if it is for this node?](#)
3. [How do I add a packet type of my own?](#)
4. [How do I get a node to maliciously alter/drop packets?](#)
5. [How do I get one in ten nodes to maliciously alter/drop packets?](#)
6. [How do I intentionally drop one in ten packets?](#)
7. [Where are the available routes stored?](#)
8. [How are routes added?](#)
9. [How is a route selected?](#)
10. [How can these caches be purged?](#)
11. [Mobicache is in cache used in DSR...but where is it that defined?](#)
12. [I want to use an alternate metric for route selection?](#)

13. Can DSR be altered to use "multipath"?
14. I want to make the source node and the destination node have two or more communication paths so I can select one as the primary and the other as a backup. I've setup these two routes (by node placement), but DSR only uses one of them?
15. Does DSR support multicast?
16. Where is the route of a packet stored?
17. How can the srpacket be examined/altered?
18. How can the route of a packet be viewed?
19. Dumping the route from the first few packets works, but not with subsequent packets?
20. How is an srpacket constructed from a packet of type Packet?
21. Does the route in the packet header have to be used?
22. How can the route be altered?
23. How do I tell what type of packet it is?
24. How can I tell what node code is currently executing on?
25. Where are packets queued before delivery to the mac?
26. What is the cmu-priqueue class?
27. How are packets added to the queue?
28. How do I get the length of the queue?
29. How does the following piece of code work?
30. How is a packet created?
31. How is a packet sent?
32. How is a broadcast packet sent?
33. How is flow state turned off?
34. What is the next hop for a packet?
35. How can a piece of code be called repeatedly by a timer?
36. Can I tell the c code how many nodes are in the simulation?
37. I get a Segmentation fault when I run a DSR script?
38. How does a node tell how many neighbours it has?
39. How do I turn off promiscuous mode?
40. How do I debug my code?
41. Why is there no information about the MAC in the trace files?
42. What application gives a nice visual representation of classes and how they relate?
43. How do I add my own information to the trace files?
44. How do I analyse trace files?
45. I've made changes to somefile, but when I run the simulation nothing has changed?
46. What is the status of DSR?

1. What happens to a packet when it is received by DSR?

It first arrives at

`recv(Packet* packet, Handler*)`

it is checked to see if it has a source route, if no source route is present, it may be a broadcast packet. If so, it is checked to see if it is an outgoing or incoming broadcast packet

If the packet was not a broadcast, it must have been an outgoing packet and a route is found for it.

If the packet did have a source route, the destination of the packet is checked against the node's net_id and the broadcast address. If the destination matches either of these the packet is sent to

```
handlePacketReceipt(p);
for further processing
```

if the destination does not match either address, the packet must be a route request, route error(if snooping is enabled), a packet to be forwarded or an invalid packet. It is processed accordingly.

2. What happens to the packet if it is for this node?

The packet will be dealt with by

```
handlePacketReceipt(p);
```

here it is checked to see if it is a route reply, route request, route error...

it is in this method that you can add all calls to your own methods to processing coming packets.

A route reply packet is recognised by

```
if (srh->route_reply()) {
    acceptRouteReply(p); //and handled by this
}
```

srh->route_reply() is a call to a method in hdr_sr.h

which checks if an int in a struct is set or not.

3. How do I add a packet type of my own?

You can add your own structures to identify that particular packets have certain properties, this is done in hdr_sr.h

For example if you added a

```
struct encryption_state {
    int enc_valid_; /*encryption header is valid? */
    int enc_type_; /*type of encryption used */
};
```

along with this structure you would have

```
struct encryption_state enc_state_;
```

.

```

.
.
inline int& encryption_valid() {return enc_state_.enc_valid_; }
inline int& encryption_type() {return enc_state_.enc_type_; }

```

the above both check and set the values in the structure.

4. **How do I get a node to maliciously alter/drop packets?**

Take a look at the questions one and two above.

Add in the criteria at the appropriate point. e.g. if you want to alter all packets to be forwarded, insert your code before the call to `handleForwarding(p)`;

Or place your code within the `handleForwarding` method.

5. **How do I get one in ten nodes to maliciously alter/drop packets?**

Use a simple modulus command on the node id.

To get the node id see question **How can I tell what node code is currently executing on?**

6. **How do I intentionally drop one in ten packets?**

See the ns-2 manual chapter on Error Models.

7. **Where are the available routes stored?**

in `primary_cache` and `secondary_cache` in `mobicache.cc`

8. **How are routes added?**

Routes can be added as the result of receiving a route reply (after sending a route request) or by overhearing a route used (or routing information) in a packet destined for another node.

Routes discovered from a route reply are added to the primary cache.

```

void MobiCache::addRoute(const Path& route, Time t, const ID& who_from) {
.
.
    (void) primary_cache->addRoute(rt, prefix_len);
.
}

```

Routes discovered by overhearing a packet are added to the secondary cache.

```

void MobiCache::noticeRouteUsed(const Path& p, Time t, const ID& who_from) {
.
.
    (void) secondary_cache->addRoute(stub, prefix_len);
.
}

```

the above methods call

```
Path* Cache::addRoute(Path & path, int &common_prefix_len) { ... }
```

9. How is a route selected?

searchRoute(..) selects routes from the cache that which lead to the requested destination.

searchRoute(..) is called by findRoute(..)

Each time searchRoute(..) finds a route to the destination, findRoute(..) checks if it is the shortest known route found so far.

searchRoute(..) continues checking every route in the cache and findRoute(..) keeps a record of the shortest one.

the code in findRoute(..) that does this is :

```
while (primary_cache->searchRoute(dest, len, path, index)) {
    min_cache = 2;
    if (len < min_length) {
        min_length = len;
        route = path;
    }
    index++;
}
```

len is the lenght of the route just found

path is the route just found by seachRoute(..)

route is the route that was passed to findRoute(..) by the caller.

index is the location in the route cache that will be checked next

the secondary cache is then checked in a similar manner.

10. How can these caches be purged?

This may not be the best way, but it should work –

```
delete primary_cache;
delete secondary_cache;
```

```
primary_cache = new Cache("primary", 30, this);
secondary_cache = new Cache("secondary", 64, this);
```

11. Mobicache is in cache used in DSR...but where is it that defined?

The code for the other caches is not compiled, this leaves mobicache as the only available cache. See the following lines in `#ifdef DSR_LINKCACHE` and `#ifdef DSR_SIMPLECACHE`.

12. I want to use an alternate metric for route selection?

The way that DSR selects a route for a particular packet is controlled by the `mobicache.cc` file, as explained above.

Make alterations to this file.

13. Can DSR be altered to use "multipath"?

See the previous question.

14. I want to make the source node and the destination node have two or more communication paths so I can select one as the primary and the other as a backup. I've setup these two routes (by node placement), but DSR only uses one of them?

So it could happen that in your network DSR is aware of a route between node X and node Y which uses the route

X - A - B - C - D - E - F - Y

and on visually or other inspection you see that it's a better route

X - K - J - Y

but unless this route has been discovered (through a route discovery or by overhearing its use) by X or Y, they cannot know about it.

What might help make this clear is to use static nodes in an arrangement in which two routes between a source/destination pair exists, a long route and a short route.

However, create the topology so that the short route will not work until after the longer route has been established and is functioning. You could do this by leaving a gap in the short route and then moving the a node to fill the gap.

DSR will continue to use the long route because it is not aware of the short route.

15. Does DSR support multicast?

As of draft-ietf-manet-dsr-10 (the latest document I know of), it does not.

16. Where is the route of a packet stored?

seems to be in `srpacket.h`

```
struct SRPacket {
    .
    .
    Path route;
    .
}
```

17. How can the srpacket be examined/alterd?

There are various methods in the `hdr_sr.h` and `path.cc/h` files that allow you to examine and alter everything about the packet.

18. How can the route of a packet be viewed?

where `p` is an `SRPacket` use - `p.route.dump()`

19. Dumping the route from the first few packets works, but not with subsequent packets?

Check for the following -

```
static const bool dsragent_enable_flowstate = true;
```

If it's there, you are using flow state, this does not use headers in packets once a route has been established. See the draft for details.

20. How is an srpacket constructed from a packet of type Packet?

```
hdr_sr *srh = hdr_sr::access(packet);
SRPacket p(packet, srh);
```

21. Does the route in the packet header have to be used?

as far as I know, yes. But you can alter the route or remove it, see below

22. How can the route be altered?

for an `SRPacket p` do the following

```
p.route = new_route; // whatever route you want to use
p.route.resetIterator(); // may not need to do this, sets cur_index = 0
cmh->size() -= srh->size();
p.route.fillSR(srh);
```

23. How do I tell what type of packet it is?

```
cmh->ptype() // returns a string
```

24. How can I tell what node code is currently executing on?

use `net_id.dump()`, this is the code used by the trace mechanism

25. Where are packets queued before delivery to the mac?

see next question

26. What is the cmu-priqueue class?

This is the class in which packets are placed, according to their priority before being delivered to the MAC. There are four priority queue, but there is no reason that you couldn't add more. The higher the priority of the queue, the sooner a packet will be sent from that queue

27. How are packets added to the queue?

Packets are added using the following or similar statements, they can be found in `dsragent.cc`

```
Scheduler::instance().schedule(ll, p, 0.0);
```

28. How do I get the length of the queue?

There are four queues for the different kinds of packet. see dsr-priqueue.cc/h
 To get the length of any of these queues use -

```
queue_length = prq_snd_[queue_number].ifq_len;
```

29. How does the following piece of code work?

This piece of code is in dsragent.cc

```
Packet *r, *nr, *queue1 = 0, *queue2 = 0;
// pkts to be recycled
while((r = ifq->prq_get_nexthop(to_id.getNSAddr_t())) {
  r->next_ = queue1;
  queue1 = r;
```

my understanding is that

*r, *nr, *queue1 and *queue2 are pointers to some packets

which link to each other by the packet->next_

together, these packets make up the queue of packets in a particular queue

30. How is a packet created?

```
SRPacket p;
p.src = net_id;
p.pkt = allocpkt();
```

```
hdr_sr *srh = hdr_sr::access(p.pkt);
hdr_ip *iph = hdr_ip::access(p.pkt);
hdr_cmn *cmnh = hdr_cmn::access(p.pkt);
```

31. How is a packet sent?

```
Scheduler::instance().schedule(ll, p.pkt, 0.0);
```

This schedules the packet p.pkt to be send in 0.0 seconds to the link layer output.

32. How is a broadcast packet sent?

do something like -

```
SRPacket p;
p.src = net_id;
p.pkt = allocpkt();
```

```
hdr_sr *srh = hdr_sr::access(p.pkt);
hdr_ip *iph = hdr_ip::access(p.pkt);
hdr_cmn *cmnh = hdr_cmn::access(p.pkt);
```

```
iph->daddr() = IP_BROADCAST; iph->dport() = RT_PORT;
iph->saddr() = Address::instance().create_ipaddr(net_id.getNSAddr_t(), RT_PORT);
iph->sport() = RT_PORT;
iph->tll() = 1;
```

```
cmnh->ptype() = PT_DSR;
cmnh->size() = size_ + IP_HDR_LEN; // add in IP header
```

```
srh->init();
```

```
Scheduler::instance().schedule(ll, p.pkt, 0);
```

```
p.pkt = NULL;
```

33. How is flow state turned off?

```
static const bool dsragent_enable_flowstate = false;
```

34. What is the next hop for a packet?

```
srh->get_next_addr() or cmh->next_hop()
```

35. How can a piece of code be called repeatedly by a timer?

the easiest way is to use `SendBufferTimer::expire(Event *)` method. Add your call to this and it will be executed approximately every 1/3 seconds.

36. Can I tell the c code how many nodes are in the simulation?

```
int num_nodes = God::instance()->nodes();
```

37. I get a Segmentation fault when I run a DSR script?

Change

```
set opt Queue/DropTail/PriQueue
```

to

```
set opt CMUPriQueue
```

38. How does a node tell how many neighbours it has?

From the cache, a node is able to guess what neighbours it has. Iterate through the paths in the cache, taking the first node in each path to be your immediate neighbour.

Problem is that some or many may be out of date.

39. How do I turn off promiscuous mode?

In dsragent.cc

Set

```
bool dsragent_use_tap = false;
```

```
// should we listen to a promiscuous tap?
```

40. How do I debug my code?

I found this useful: <http://heather.cs.ucdavis.edu/~matloff/debug.html>

41. Why is there no information about the MAC in the trace files?

```
$ns_node-config -adhocRouting $val(adhocRouting) \
  -llType $val(ll) \
```

```

-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-phyType $val(netif) \
-channelType $val(chan) \
-topoInstance $topo \
-agentTrace ON \
-routerTrace OFF \
-macTrace ON
^^^

```

setting -macTrace ON

42. **What application gives a nice visual representation of classes and how they relate?**

Source navigator from Redhat and Source Insight are pretty good.

43. **How do I add my own information to the trace files?**

Simply add your own trace command typically something like -

```
trace("X %.5f_%s_%s",Scheduler::instance().clock(),net_id.dump(), your.info);
```

where X identifies the type of event, then the time and node id are added to the trace and finally **your.info**

44. **How do I analyse trace files?**

Awk or perl scripts are the most common. But any programming language can be used.

If programming is not your thing(!), try trace graph <http://www.tracegraph.com>

45. **I've made changes to somefile, but when I run the simulation nothing has changed?**

Delete *.o from the ~ns/dsr directory and dsr-priqueue.o from the ~ns/queue directory.

Then recompile...

46. **What is the status of DSR?**

The DSR specification has been submitted to the IESG for publication as Experimental RFC.

About me:

Name : Bryan Hogan

I completed a Masters of Engineering in Telecommunications at the University of Limerick

Email : [bjhogan\[at\] fastmail\[dot\] fm](mailto:bjhogan@fastmail.fm)

Corrections and additional information are very welcome. **But I am not actively answering questions nor updating this document.**

Areas of Interest : Ad hoc networks, routing, 802.11, TCP/IP, Unix/Linux

Topics covered in thesis :

- Existing congestion control mechanisms
- Affects of wireless environment on congestion
- Dynamic Source Routing protocol
- Ad hoc networking congestion avoidance algorithm

Links:

[The Rice University Monarch Project](#)

Publications:

[Congestion Avoidance in Source Routed Ad Hoc Networks](#) 13th IST Mobile and Wireless Communications Summit, Lyon, 2004

Other Interests : Scuba diving, free diving, swimming, yoga, travelling, the cold war