Simple Techniques for Enhancing the Robustness of DSR[*]

Po-Wah Yau and Chris J. Mitchell
Mobile VCE Research Group
Information Security Group
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
P.Yau@rhul.ac.uk, C.Mitchell@rhul.ac.uk

ABSTRACT

The Dynamic Source Routing protocol for ad hoc networks is one of the leading candidates for adoption by the IETF amongst the class of reactive routing protocols. The currently defined protocol assumes total node cooperation, which allows attacks by failed or malicious nodes whose behaviour violates the protocol rules. Failed nodes will typically fail to forward information; badly failed nodes will inadvertently introduce false routing messages; selfish nodes deliberately refuse to forward packets; finally, malicious nodes may disrupt the network in any possible way. There have been many proposals for securing DSR, most of which introduce major overheads and hence make the protocol less attractive. This paper takes a somewhat different approach and presents some simple techniques aimed at making the protocol more robust with minimal overhead

Keywords

Routing protocols, network security, mobile networks.

1. INTRODUCTION

Mobile ad hoc networks have existed since the 1950s, with the inception of the DARPA packet radio network [4]. Since then, like many other technologies, their use has been restricted to military scenarios, but recently the realisation of the potential for commercial use has grown. In parallel with this, research into the security of mobile ad hoc networks has recently mushroomed, resulting in the publication of many papers on the subject [1, 5, 6]. This particular paper concentrates on the proposed Dynamic Source Routing (DSR) protocol [2], and in particular on the version defined in the latest draft being considered by the Internet Engineering Task Force (IETF)[1] [3]. However, rather than introduce any new security mechanisms which will operate above or below the network layer, the ideas given are simple techniques which can be applied to the implementation of DSR to make it more robust against the inherent vulnerabilities of mobile ad hoc networks. Some of these vulnerabilities exist because of the assumption that there will be total cooperation amongst the mobile nodes.

Section 2 introduces the terminology used in this paper. Section 3 gives an overview of DSR routing. Section 4 describes a threat model for mobile ad hoc networks, used throughout the rest of the paper. Section 5 will focus on certain parts of the routing protocol, and describes what can be modified to eliminate specific vulnerabilities. Note that this paper is concerned only with the security of the routing protocol, and not with the many other security issues that can arise in mobile ad hoc networks.

2. TERMINOLOGY

The following terms are used in this document, but may be used differently elsewhere. A *node* is a device which has a network interface participating in routing in a mobile ad hoc network. It may or may not be mobile, and may also be part of another network. It is important to realise that a node can actually be a large network, or it could just be a single mobile device such as a mobile phone. An *originator node* is a node which originates a DSR data packet, intended for a certain *destination node*. A node is a *neighbour node* of another node if it is only one hop away and within direct transmission range. If the destination node is not a neighbour node of the originator node, the data packet will have to traverse a multi-hop route consisting of *intermediate nodes*. In a specific scenario, the *sending node* is the last node to have forwarded the data packet.

In DSR, a data packet can contain several headers of different types. *Route Request* packets are generated and sent whenever a node wishes to send packets to a certain destination node but does not have an up-to-date route. *Route Reply* packets are sent in response to Route Requests, by either an intermediate node or the destination node itself. *Route Errors* are generated and sent whenever a node fails to forward a packet because of a link break. Thus, Route Error messages contain details of the location of the link break. See Section 3 for details of how these DSR header options are used.

Most data packets in DSR contain a *Source route/Route record*. This is a sequential list of node addresses from the originator node to the destination node, and is used to deliver data packets from the listed originator node to the listed destination node. In the context of sending data packets along a route, the *Forward Path* is the (downstream) route from originator node to destination node. Conversely, the *Reverse Path* is the (upstream) route from the destination to originator node. Finally, so that DSR can interface

[*]The work reported in this paper has formed part of the Networks & Services area of the Core 2 Research Programme of the Virtual Centre of Excellence in Mobile & Personal Communications, Mobile VCE, www.mobilevce.com, whose funding support, including that of EPSRC, is gratefully acknowledged. Fully detailed technical reports on this research are available to Industrial Members of Mobile VCE

[1]`http://www.ietf.org`

with other routing protocols or, indeed, another DSR ad hoc network, allowing scalability, DSR identifies links with 'other' networks as *External links.* An external link is only permitted to be the last hop in a Source Route.

3. DSR PROTOCOL OVERVIEW

The DSR protocol is a distance-vector protocol for ad hoc networks [2], where all nodes participate in a distributed routing process. The ad hoc network consists of a collection of mobile nodes, where each node participating in DSR maintains a number of data structures fundamental to the running of DSR. This paper does not provide a complete description of DSR, but only describes the parts of the protocol which are necessary to understand the proposed modifications. Section 3.1 describes how data packets are routed through a DSR controlled mobile ad hoc network. Section 3.2 describes the route discovery cycle, before section 3.3 describes route maintenance.

Two key data structures that a node participating in DSR must correctly maintain, and that are relevant to this paper, are the route cache and the gratuitous Route Reply table. The route cache is where all route information is stored by a node. New information added to the route cache is extracted from both received DSR packets containing route control information, and from overheard packets being sent to other nodes. Route information is only removed from the route cache upon receiving a DSR packet with Route Error information. When new information is added, the node must check to see if there are any packets which can be immediately sent as a result of this new information. The gratuitous Route Reply table is used to record any gratuitous Route Replies sent by the node. This process is described in more detail in Section 3.2.

3.1 Processing DSR Data Packets

When a node wishes to send a data packet it will first search its route cache for a route to the destination IP address. If there is more than one route then one will need to be selected. If the destination is more than one hop away, a source route is added to the data packet containing the chosen route. The data packet is then transmitted to the first-hop node.

When a node receives a DSR data packet with a source route, the node will add the source route to its cache, subject to the Medium Access Control (MAC) layer protocol conditions[2].

If the node is the target of the data packet then it can pass the packet contents to the network layer for processing. If the node is not the target then it should perform route maintenance (see Section 3.3) to check if the specified next-hop in the source route is a reachable node. If the neighbour node is reachable, the node removes the first address in the data packet's source route and transmits the data packet to the next-hop.

3.2 Route Discovery

Route discovery is initiated when a node originates a data packet for transmission, and finds that it has no source route for the destination of the packet. The node will create and broadcast a Route Request. The Route Request will contain a unique identification value, the originator node's IP address, the destination node's IP address, and a source route. The initial source route only contains the originator node's IP address.

The Route Request will be discarded by nodes if it is a duplicate, or if the node's address is already in the source route (to prevent loops). If the receiving node is not the destination or does not know of a route to the requested destination, the node adds an entry into its recent Route Request cache and creates a copy of the Route Request, appending its own IP address to the source route, before broadcasting the new Route Request after a random *BroadcastJitter* delay.

Route Replies containing a source route to the requested destination node must be sent by the destination node, and intermediate nodes with a route to the destination stored in their route caches. Every Route Reply uses the same identification value as the corresponding Route Request so that it can be identified when it is received. When the originator node receives the Route Reply, it adds the source route to its route cache and then checks if it has any packets to send using the new information.

3.2.1 Automatic Route Shortening

An intermediate node may operate in promiscuous mode and overhear packets, in which case it should check to see if automatic route shortening is possible. This is possible if the node's own address appears in the unexpended portion of the source route [2, p152].

If this situation occurs, the addresses in the source route between that of the overheard node and the intermediate node itself are redundant and can be removed (see Figure 1). The intermediate node needs to check its gratuitous Route Reply table to see if it has received a gratuitous Route Reply from either the source node of the data packet, or the node which was overheard forwarding the packet. No further action should be taken if a reply has been received. Otherwise an entry is created in the gratuitous Route Reply table for this data packet, and a gratuitous Route Reply containing the shorter route should be sent to the originating node. If bi-directional communication is needed because of the MAC protocol, the gratuitous Route Reply's source route should be the reverse of the source route of the overheard data packet. A listening node may factor in other issues such as signal-noise ratio before determining that it can rationally be part of the shorter route.

3.3 Route maintenance

As mentioned previously, a node must perform route maintenance before forwarding a data packet. The preceding node is responsible for confirming that the next-hop node is reachable. There are three methods of acknowledging that the link is okay and these methods, in order of priority, are link layer acknowledgements, passive acknowledgements and network layer acknowledgements. This paper only proposes enhancements to the passive acknowledgement mechanism — see [2] for further information about the other two.

3.3.1 Passive Acknowledgements

Passive acknowledgments have second priority and are used when nodes' network interfaces can operate in promiscuous mode, with the majority of network links bi-directional. A node will send a packet to the next-hop and then listen for

[2]Whether or not the source route is added to the route cache will depend on whether the MAC layer protocol allows uni-directional links or not.

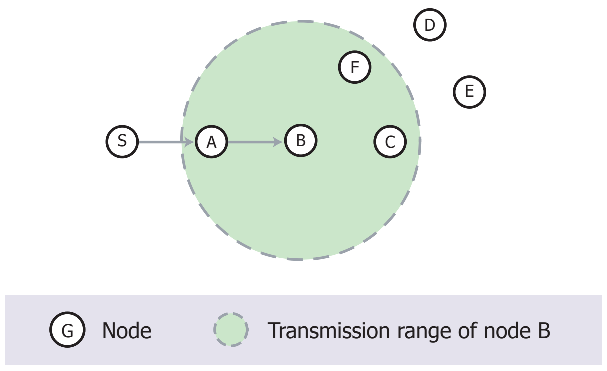

Figure 1: An example of how automatic route shortening can work. In the figure, the source route is SABCEFD. Node F overhears node B sending a packet with source route CEFD. Node F can now send a gratuitous Route Reply to S, telling it to use the shorter source route SABFD.

the next-hop to forward this packet. The node then checks that the data packets match, and that the source route of the forward packet is shorter than the source route of the packet that the node received. If both checks are positive then the next-hop has performed the passive acknowledgement. This cannot be used for the last hop of a data packet as there will be no transmission to use as a passive acknowledgement.

3.3.2 Processing Route Error data packets

When a node verifies that the next-hop for a data packet x is unreachable, the node needs to create a Route Error containing: a NODE_UNREACHABLE option, the IP address of the next-hop node that is unreachable, and the IP address of the source of the Route Error (i.e. the node's own address). The Route Error is then transmitted to every node which has sent a packet through a route which uses the broken link.

When a node receives a Route Error, it must delete all routes in its route cache which use the broken link, i.e. the link from the source of the Route Error to the unreachable node specified in the Route Error. Which data packets should be retransmitted is the decision of the upper layers, which may use another route in the route cache, or initiate a new route discovery procedure for the affected destination. However, there should be a limit on the number of times that a new route discovery should be used for that destination. Also, to increase the spread of the Route Error information, any subsequent Route Requests for the affected destination may be piggybacked with the Route Error header. This enables nodes to ignore Route Replies which contain the broken link.

3.3.3 Salvaging a packet

If the intermediate node has detected a link break but has an alternative route to the destination of the data packet, the node can salvage the packet up to an n number of times. [2, p151].

To salvage a data packet x, the intermediate node first sends a data packet containing a Route Error option back to the originator of x, as described above. The node should now copy the new alternative source route into the data packet being salvaged, and the salvage field should be incremented. The salvaged packet can then be forwarded and transmitted as in Section 3.1.

4. AD HOC NETWORK THREAT MODEL

The threat model used here distinguishes between external and internal attacks — see also [6, p.25]. External attacks are performed by unauthorised nodes or entities. These threats are likely to be more easily detected than threats from internal nodes. Internal attacks are conducted by internal nodes, i.e. authorised nodes within the ad hoc network, and are thus likely to be more difficult to detect, as they arise from trusted sources.

In the text below, 'correct' data packets and 'correct' procedures are simply those that adhere strictly to the DSR routing protocol being used. By contrast 'incorrect' data packets and 'incorrect' procedures are those which are in any way different to the format and behaviour as stated in the protocol. 'False' data packets are data packets that are of the correct protocol format, but contain false information.

4.1 External Threats

The main external threats posed to a mobile ad hoc network routing protocol are as follows:

- Unauthorised reading of routing information,
- Unauthorised modification of routing information,
- Preventing the routing protocol from functioning, and
- Masquerading as an authorised node.

The external threats exist because of the inherently limited physical security of mobile ad hoc networks. The wireless communications medium makes it easier to intercept communications and inject messages than in an equivalent wired network. Hence, an ad hoc network is very vulnerable to attacks where an external attacker masquerades as a trusted node to perform internal attacks (see below). The need for light-weight, highly mobile devices means that the physical security of the device itself may also be limited, allowing the device to be easily compromised.

4.2 Internal Threats

The threats posed by internal nodes are very serious, as internal nodes will have the necessary information to participate in the routing protocol. Internal nodes can misbehave in a variety of different ways; we identify four categories of misbehaviour, as follows:

- Failed nodes,
- Badly failed nodes,
- Selfish nodes, and
- Malicious nodes.

Failed nodes are simply those unable to perform a route operation; this could be for many reasons, including power failure and environmental events. The main issues with failed nodes of this type are failing to update data structures, or the failure to send or forward data packets, including those with DSR route header options. The threat of having failed nodes is most serious if failed nodes are needed as part of an emergency route, or form part of a secure route.

Badly failed nodes exhibit features of failed nodes such as not sending or forwarding data packets or route messages. In addition they can also send false routing messages, which are still correctly formatted, but which contain false information and are a threat to the integrity of the network. For example, false Route Requests for a node which does not exist may circulate in the ad hoc network using up valuable bandwidth, as no node can provide a suitable reply. Unnecessary Route Requests, for routes which badly failed nodes already have, might also be sent. False Route Replies in response to a true Route Request may result in false routes being set up and propagated through the network. False Route Error messages will cause working links to be marked as broken, potentially initiating a route maintenance procedure.

Selfish nodes exploit the routing protocol to their own advantage, e.g. to enhance performance or save resources. Selfish nodes are typified by their unwillingness to cooperate as the protocol requires whenever a personal cost is involved, and will exhibit the same behaviours as failed nodes, depending on which operations they decide not to perform. It is important to emphasise that, in this model, selfish nodes do not perform any action to compromise network integrity by actively introducing incorrect information.

Finally, malicious nodes deliberately disrupt the operation of the routing protocol in some way, i.e. denying network services to other nodes. Hence, they may display any of the behaviours shown by the other types of failed nodes. The impact of a malicious node's actions are greatly increased if it is the only link between groups of neighbouring nodes.

Note that two failed nodes within the same category may exhibit different degrees of failed node behaviour. For example, some nodes will be more selfish than others. Also, a node may demonstrate behaviours from more than one category — indeed, this may even be the typical case.

5. PROTOCOL MODIFICATIONS

Simple techniques are now described which can be used to enhance the robustness of DSR, alleviating certain attacks described in the threat model.

5.1 Source Route Verification

The currently defined version of the protocol only requires a node performing passive acknowledgements to check the source route header 'Segments Left' field, to passively verify that the data packet has been forwarded with the correct source route [3, p59]. Passive acknowledgement is achieved if the value of the 'Segments left' field, in the forwarded data packet, is smaller than the value in the original data packet which was sent to the neighbour. The obvious attack here is misdirection, where a malicious node can forward the data packet with a shorter but incorrect source route.

One solution to this would be to add a check of the source route in the packet being forwarded by the neighbour. Only if the source routes match (apart from the last address which will have been removed by the neighbour node performing the forwarding) is the passive acknowledgement deemed to have succeeded. Note that this check can only be performed using passive acknowledgements, and that the misdirection attack is still possible when using the other two acknowledgment methods (see Section 3.3).

5.2 Route Request Verification

Another means of source route verification is during route discovery, as also used in [5]. As no acknowledgement system is used, a badly failed node forwarding a Route Request can simply add spurious addresses to the source route without detection. A check by the neighbour nodes which receive this incorrect Route Request can be made, to see if the last address of the source route corresponds to the address of the node from which it was received, i.e. if the Route Request was received from node x then the recipient should check that the address of x is the last address in the source route.

However, what this approach will not detect is the scenario where a malicious or badly failed node adds different addresses, ending with its own address as the last address in the source route. The node will have thus injected a route which will divert traffic, before finally travelling to the intended destination. Indeed, a malicious node may make use of multiple network interfaces [3, p66] to create an undetectable loop. Therefore this technique is only useful for detecting badly failed nodes, who are more likely to add spurious addresses than malicious nodes.

5.3 Limiting Denial of Service Attacks that exploit Route Error Messages

Route Error messages are a potential source of denial of service attacks from malicious nodes. False Route Error messages can easily be propagated throughout the network, falsely indicating that certain links are broken. A method to partially mitigate this threat is to include a unique identifier with every data packet sent. Any Route Error message should then contain the unique identifier of the data packet which triggered the Route Error message. It is likely that a node which has sent the data packet will receive the corresponding Route Error message in the event of the link break, because it was part of the source route. Hence, any nodes which overhear or receive a Route Error message should check that it contains a unique identifier corresponding to a data packet it has forwarded, before accepting the Route Error message as true. A side-effect is that every node has to store the unique identifiers of each data packet it has forwarded, for a certain period of time.

Another possible check is whether or not the address of the node which has detected the break was actually part of the source route. This does impose the overhead that the source route used by every data packet forwarded must be stored, which may result in a serious loss of valuable storage space.

These techniques limit the scope of any false Route Error message attack. Malicious nodes can now only send false error messages using the unique identifier of a data packet that has already been sent, and then only the nodes which have already forwarded the data packet will believe the false Route Error message.

5.4 Improving Automatic Route Shortening

The mobile nature of devices in an ad hoc network leads to solutions with varying optimisations to cope with the fact that parts of routes may break at any time. The following scenario reduces the effectiveness of automatic route shortening (see Section 3.2). Consider the chain of events shown in Figure 2, where a node later in a source route moves closer, performs automatic route shortening, but then moves away.

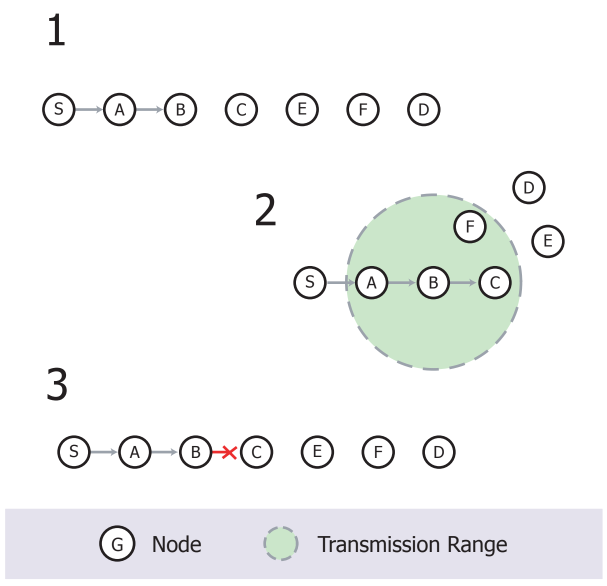

Figure 2: In diagram 1, node S originates and transmits a data packet with source route 'SABCEFD'. In diagram 2, when node B receives the data packet, nodes F and D move so that node F is now in transmission range of nodes B and C. Node B forwards the data packet to node C with source route 'CEFD'. Following the automatic route shortening procedure, node F overhears the data packet and sees that its address occurs later on in the source route. It then sends a gratuitous Route Reply to node S, if it has not done so before. Node S now has a shorter source route to node D, i.e. 'SABFD'. However, a problem arises if node F moves out of the transmission range of node B, since the route contained in the gratuitous reply is now false. Indeed, node F could move back to a position where the old route is now valid, as in diagram 3.

A possible solution is to maintain a table of received gratuitous Route Replies, and only accept a source route given in a gratuitous Route Reply if it has been received a certain number of times t say. The value of t can be made a dynamic variable which can be adjusted according to the mobility of the nodes in the network, where a highly changing network will need a higher value before a route can be accepted as stable.

5.5 Improving Salvaging

Recall that this is the process of forwarding a data packet along an alternative route, if the original source route is broken (Section 3.3). If the alternative route is also broken, then this optimisation actually becomes a burden on the node performing salvaging. This can be exploited by malicious nodes to perform a serious denial of service attack. A malicious node can inject a large number of false routes pointing to a destination which does not exist. The malicious node could then send a data packet addressed for the false destination, and let nodes who have stored the false routes in their route caches waste resources trying to salvage the packets using alternative routes.

To reduce this threat, a node should only salvage a packet using an alternative route which has used before. This means that a node must have successfully originated or forwarded a data packet along the route whilst receiving no Route Error message. Again, the number of times a route has been successfully used can be a dynamic variable which a node can adjust, depending on how stable a route the node wants, i.e. the higher the number, the more the node can trust that the route will not break while it is being used.

5.6 Other suggestions

One of the properties of ad hoc networks is their inherent redundancy, leading to multiple paths to certain destinations. The protocol specification does not specify how a node with multiple routes to the same destination should select which route to use. In order to increase the probability that a data packet will reach its destination, the node could send copies of the data packet along several routes, preferably along routes which do not use the same nodes. If IP addressing, and therefore IP headers, are not used, then some unique identification value will need to be included. This enables copies of data packets to be discarded. There will of course be a trade-off, as using multiple routes will result in more bandwidth being used.

DSR allows the possibility of creating network hierarchies [3, p17,37]. Not only does this make the protocol more scalable, but utilising this feature also helps enhance security in various ways. In the operation of DSR, external addresses (addresses of nodes not within the 'group') are only permitted to be the final address in a source route. This constrains internal route information within a group. Therefore any false or incorrect routing information is also contained within the group, and does not affect the rest of the network.

6. REFERENCES

[1] L. Buttyan and J. Hubaux. Report on a working session on security in wireless ad hoc networks. *ACM Mobile Computing and Communications Review*, 6(4), Oct 2002, to appear.

[2] D. Johnson, D. Maltz, and J. Broch. DSR — The dynamic source routing protocol for multihop wireless ad hoc networks. In C. Perkins, editor, *Ad Hoc Networking*, chapter 5, pages 139–172. Addison-Wesley, 2001.

[3] D. Johnson, D. Maltz, Y. Hu, and J. Jetcheva. The dynamic source routing protocol for mobile ad hoc networks — DSR. Internet Draft 8, Internet Engineering Task Force, Feb 2002.

[4] J. Jubin and J. Tornow. The DARPA packet radio network protocols. *Proceedings of the IEEE*, 75:21–32, 1987.

[5] P. Papadimitratos and Z. Haas. Secure routing for mobile ad hoc networks. In *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference, January 27-31, 2002, San Antonio, Texas USA*, pages 27–31. The Society of Computer Simulation International, 2002.

[6] L. Zhou and Z. Haas. Securing ad hoc networks. *IEEE Network*, 13(6):24–30, November 1999.