Simulation of SLAWAN Routing Protocol in Mobile Ad-hoc Networks

[1]Rashmi Singh and [2]Sweta Verma

***Abstract*—This paper proposes a hybrid routing protocol which eliminates the delay before starting actual communication for most new connections of MANET's. W_AntNet [3] and SDLRLA [1] routing protocol are used together for decreasing route discovery latency in real time communications in high dynamic networks. This protocol is named as Secure Load Aware Wireless Ant Network (SLAWAN) routing protocol. This protocol monitors the congestion status of active routes and reconstructs the path when nodes of the route have their network buffer queue overloaded. Simulation results show that the SLAWAN hybrid technique proposed in this paper is able to achieve reduced end-to-end delay as compared to conventional ant-based and SDLRLA routing protocols. In addition SLAWAN provides high connectivity.**

***Index Terms*—Secure dynamic load routing load-aware (SDLRLA), Wireless Ant Network Routing (W_AntNet), Secure Load Aware Wireless Ant Network (SLAWAN), mobile agents.**

I. Introduction

Routing protocols in ad hoc networks [4] must manage frequent topology changes caused by node mobility and need to be bandwidth and power efficient. The conventional routing protocols for mobile wireless ad- hoc networks suffer from certain inherent shortcomings. The proactive routing schemes continuously update the routing tables of mobile nodes consuming large portion of the scarce network capacity for exchanging huge chunks of routing table data. The on-demand routing protocols on the other hand launch route discovery and require the actual communication to be delayed until the route is determined (found). This may not be suitable for real time data and multimedia communication applications. Ant-like mobile agents can be used for efficient routing in a network and discover the topology. However, they have certain drawbacks. The nodes depend solely on the ant agents to provide them routes to various destinations in the network. It may also happen that the nodes carrying ants suddenly get disconnected with the rest of the network due to their movement away from all other nodes, or they might go into sleep mode or simply turned off. In such situations, the amount of ants left for routing are reduced in the network which leads to ineffective routing.

This paper tries to overcome these shortcomings of ant routing and SDLRLA by combining them to develop a hybrid routing scheme. The paper is organized as follows; section II gives a brief idea on SDLRLA protocol, section III describes the proposed work SLAWAN routing protocol with simulation results in section IV and finally section V concludes the paper.

II. Secure Dynamic Load Routing Load-Aware (SDLRLA)

A. Overview

SDLRLA is based on basic DLAR functionality as described above. The assumptions made are the presence of only bi-directional links in the network and the existence of the so called MANET-IDs per node. The MANET-ID basically is a signed RSA key pair that prevents nodes from forging new identities. In order to handle MANET-IDs [19] and create signatures, nodes must be powerful enough to do asymmetric cryptographic operations in a limited number.

SDLRLA adds mechanisms to DLAR to

1) secure the integrity of a route
2) secure the freshness of a route
3) secure the authenticity of all nodes participating in a route
4) exchange of load information between nodes in a route.

Figure 1 shows the basic steps in a route discovery where node S searches a route to node D. In step 1, S creates a route request that contains the source *S*, the node *D*, a route request *ID* unique per source, a public Diffie-Hellmann key $DHPK_S$, a random nonce N_1, an initial source route SR{S}. Finally, S creates a signature *sigsks* using its private MANET-ID key. This signature protects all static information (*S*, *D*, *ID*, and *DHPKS*) from alteration. *D* can use this signature to verify, that *S* actually sent the route request. Before forwarding a route request (step 2), the intermediary node *K* appends itself to the source route and transforms the nonce *N*1 into *N*2 (see below). When the route request finally reaches *D*, the destination generates a route reply (step 3). This route reply contains all the static information from the route request, the source route, the signature from S and a new signature sig_{SKD}. This signature includes the source route and sig_{SKS} . So in the route reply phase (steps 3 & 4), alterations of the source route can be detected. Additionally, all elements from the route request are still protected. By verifying the signature sig_{SKD}, *S* can be sure that *D* created the reply. Finally, *D* adds its public Diffie-Hellmann key *DHPKD* and an encrypted, hashed key k_{SD} (a session key) to the route reply. To this basic procedure SDLRLA adds the load of the intermediate nodes as a new component, which results in a more secured and dynamic routing.

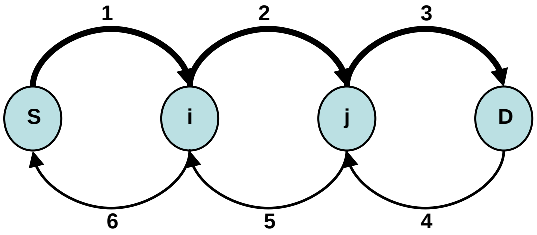
Manuscript received October 30, 2010; revised December 12, 2010.

[1]Research Scholar, UPTU , Lucknow,

[2]Associate Prof (IT), Galgotia's College Of Engg & Technology, Greater Noida

B. Detailed Work Explanation

In the proposed SDLRLA protocol, node S creates a LOAD REQUEST, similar to the previous one to node D; that contains the source *S*, the node *D*, a route request *ID* unique per source, a public Diffie-Hellmann key $DHPK_S$ and an empty node load information NL{} (Fig.1-step 1) with a signature *sigsks* using its private MANET-ID key. Since the REPLY path is decided by the destination node D, there is no need of a random nonce and a source route array. Node D chooses the reply path with the help of node-load information (NL). This is appended each time the LOAD REQUEST visits an intermediate node (see Fig.2-step 2&3). When nodes other than the destination receive a non-duplicate LOAD REQUEST, they build a route entry for the < source, destination > pair and record the previous hop to that entry (thus, backward learning). This previous node information is needed later to relay the REPLY packet back to the source of the route.



1. LREQ S D ID DHPK$_S$ NL{} sig$_{SKS}$
2. LREQ S D ID DHPK$_S$ NL{ Li } sig$_{SKS}$
3. LREQ S D ID DHPK$_S$ NL{ Li, Lj } sig$_{SKS}$
4, 5 & 6. RREP S D ID DHPK$_D$ sig$_{SKD}$

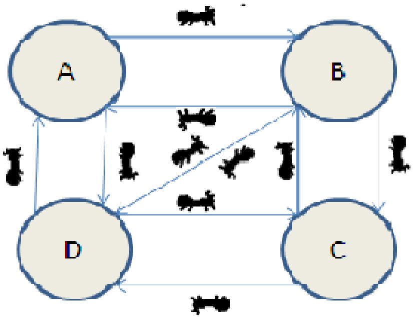
Fig.1 Example Network SDLRLA

In protocols such as AODV and DSR, intermediate node K sends a ROUTE REPLY to node S since it has a route to node D. To utilize the most up-to-date load information when selecting routes and to minimize the overlapped routes which cause congested bottlenecks, SDLRLA prohibits intermediate nodes from replying to LOAD REQUESTS. Figure 2 presents the steps involved in route discovery from node *S* to node *D* as per SDLRLA protocol. Intermediate nodes replying to ROUTE REQUESTS has an advantage of reducing the propagation of flooded packets, but causes congestion and a reply storm (i.e., too many nodes send ROUTE REPLIES at the same time resulting in collisions). After receiving the first ROUTE REQUEST packet, the destination waits for an appropriate amount of time to learn all possible routes, and selects the least loaded route to source node S. The node D then sends a REPLY packet to node S with its public Diffie-Hellmann key $DHPK_D$ and signature *sigskd.* During the active data session, intermediate nodes periodically piggyback their load information on data packets. Destination node can thus monitor the load status of the route. If the route is congested, a new and lightly loaded route is selected to replace the overloaded path. Routes are hence reconstructed dynamically in advance of congestion. The source, upon receiving LOAD REQUEST packets, selects the best route in the same manner as the destination. The source does not need to send a REPLY, and simply sends the next data packet using the newly discovered route.

III. SLAWAN ROUTING PROTOCOL

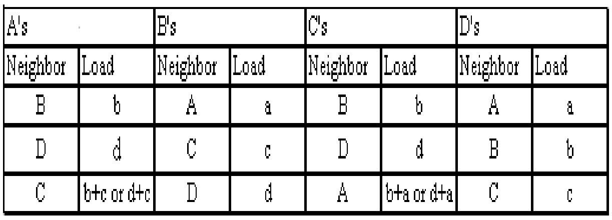
To overcome some of the inherent drawbacks of W_AntNet based routing and SDLRLA routing protocols the proposed SLAWAN technique forms a hybrid of both. The SLAWAN hybrid technique enhances the node connectivity and decreases the end-to-end delay and route discovery latency. In ant routing algorithms implemented so far there is no local connectivity maintenance as in SDLRLA. Hence when a route breaks or a less congested route is already available, the source still keeps on sending data packets to the same route unaware of the link breakage or availability of efficient routes. This leads to a large number of data packets being dropped due to congestion and link breakage. SLAWAN utilizes ants working independently and providing routes to the nodes as shown in figure 2. The nodes also have capability of launching on-demand route discovery to find routes to destinations for which they do not have a fresh enough route entry.

In this protocol every node builds a *Load Table* indicating the neighbor nodes with their respective loads [9] (see Table 1). At initial state the table remains empty. To initiate the route discovery, a node sends the forward ants *FANT* to its neighbors with its load information and load request *LREQ*. These ants are forwarded until they reach the destination or found an entry to the destination in an intermediate node's load table. As soon as an ant encounters destination a backward ant *BANT* is generated towards the source. The backward ant contains the public key *DHPKc* of the destination with its load information. During this process ants carry the load information of all the intermediate nodes and update their load tables with fresh entries.



Route discovery from A to C
FAnt A: LREQ A C ID NL{L$_A$} sig$_A$
FAnt B: LREQ A C ID NL{L$_A$,L$_B$} sig$_A$
BAnt C: LREP A C DHPKc ID NL{L$_A$,L$_B$,L$_C$} sig$_A$ sig$_C$

Fig.2. Example Network SLAWAN



A's		B's		C's		D's	
Neighbor	Load	Neighbor	Load	Neighbor	Load	Neighbor	Load
B	b	A	a	B	b	A	a
D	d	C	c	D	d	B	b
C	b+c or d+c	D	d	A	b+a or d+a	C	c

TABLE 1. EXAMPLE OF LOAD TABLE

The use of ants with SDLRLA increases the node connectivity (the number of destinations for which a node has unexpired routes), which in turn reduces the amount of route discoveries. Even if a node launches a LREQ (for a destination it does not have a fresh enough route) in this protocol, the probability of its receiving replies quickly (as compared to SDLRLA) from nearby nodes is high due to the increased connectivity of all the nodes resulting in reduced route discovery latency. Lastly, as ant agents update the routes continuously, a source node can switch from a longer (and stale) route to a newer and shorter route provided by the ants. This leads to a considerable decrease in the average end-to-end delay as compared to both SDLRLA and ant-based routing. Local connectivity in SLAWAN is maintained in a fashion similar to SDLRLA. Neighbor discovery is implemented same as used in W_AntNet algorithm where frequent HELLO broadcasts are used to maintain the neighbor load table. This table is used to select a randomly chosen next hop (avoiding the previously visited node) from the list of neighbors by the ant.

IV. Simulation Results

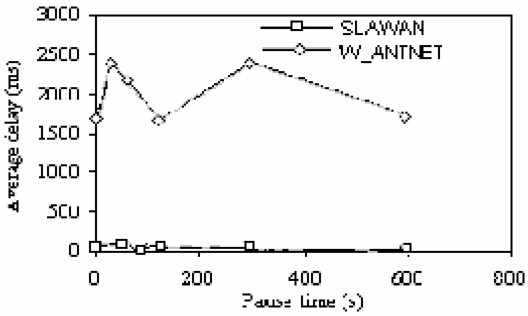


Fig. 3. Average end-to-end delay of routing data packets in W_AntNet and SLAWAN routing protocols

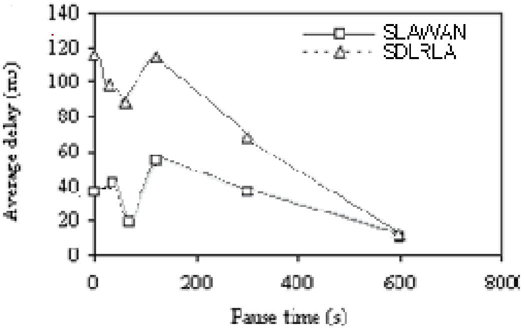


Fig. 4. Average end-to-end delay provided by SDLRLA and SLAWAN routing protocols.

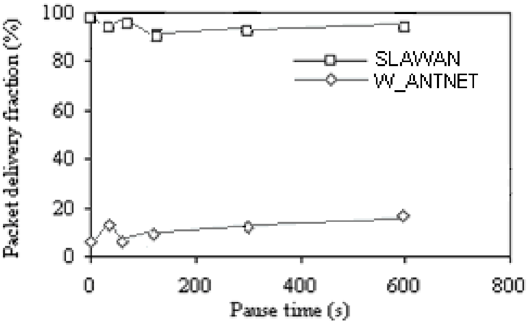


Fig. 5. Packet delivery fraction of W_AntNet and SLAWAN routing protocols.

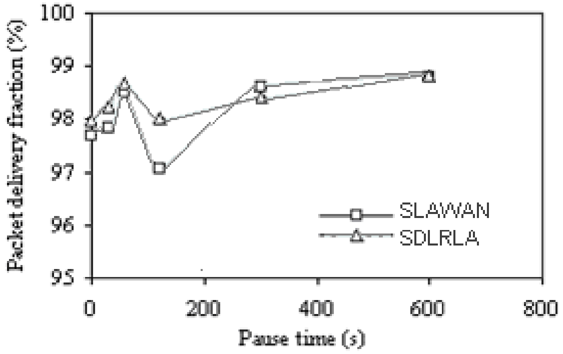


Fig. 6. Packet delivery fraction of SDLRLA and SLAWAN routing protocols.

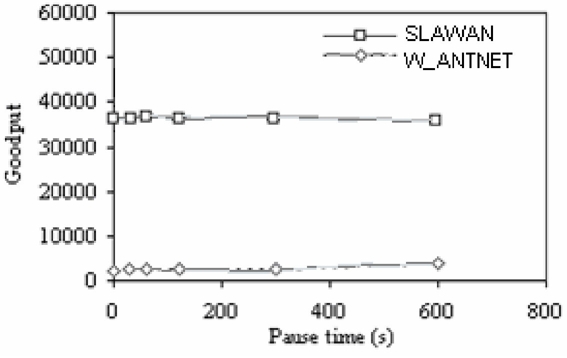


Fig. 7. Goodput of W_AntNet and SLAWAN routing protocols.

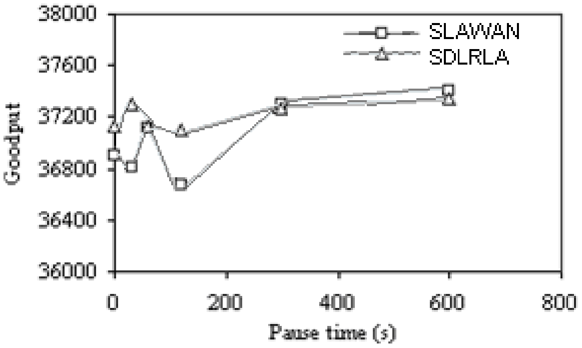


Fig. 8. Goodput of SDLRLA and SLAWAN routing protocols.

V. Conclusion

This paper tries to overcome the shortcomings of on-demand routing protocols like SDLRLA and W_AntNet based routing by combining them to enhance their capabilities and alleviate their weaknesses. SLAWAN protocol is able to provide reduced end- to-end delay and high connectivity as compared to W_AntNet. As a result of increased connectivity the number of route discoveries is reduced and also the route discovery latency. This makes SLAWAN routing protocol suitable for real time data and multimedia communication. The reduction in end-to end delay is achieved at the cost of extra processing of the ant messages and the slightly higher overhead occupying some network capacity. This however does not adversely affect the packet delivery fraction or the good put.

REFERENCES

[1] Rashmi Singh, Sweta Verma. Secure Dynamic Load Routing Load-Aware Protocol. In TICE, 2009.

[2] Rashmi Singh, Sweta Verma. Implementation of MLSRSA into SDLRLA. International Journal of Computer Network and Security (IJCNS), pages 78-85, 2009.

[3] FJ Arbona Bernat, "Simulation of Ant Routing Protocol for Ad-hoc networks in NS-2", Thesis project, Delft University of Technology, Delft, Netherlands, Nov 2006.

[4] E.M. Royer and C.-K. Toh, .A Review of Current Routing Protocols for Ad-Hoc Mobile Networks, *IEEE Personal Communications*, vol. 6, no. 2, April 1999, pp. 46-55

[5] H. Matsuo, K. Mori, "Accelerated Ants Routing in Dynamic Networks," *Second Intl. Conf. On Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, 2001.

[6] F. Kargl, "Sicherheit in Mobilen Ad Netzwerken," Ph.D. dissertation, University of Ulm, Ulm, Germany, 2003, also available as http://medien.informatik.uniulm.de/~frank/research/dissertation.pdf.

[7] S. S. Dhillon and P. Van Mieghem. Performance analysis of the AntNet algorithm. *Computer Networks*, in print, 2007.

[8] R. R Choudhary, S. Bhandhopadhyay, K.Paul, "A Distributed Mechanism for topology hoc Distributed Mechanism for topology hoc Networks Using Mobile Agents," *Proc. of First Annual Workshop on Mobile Ad Hoc Networking Computing* (MobiHOC'2000), 2000.

[9] Sung-Ju Lee, Mario Gerla, Dynamic Load-Aware Routing in Ad-hoc Networks. In Proceedings. of 3 rd IEEE Symposium on Application- specific Systems and Software Engineering Technology, 2000.