The Effect of Routing under Local Information using a Social Insect Metaphor

Suihong Liang, *Student Member, IEEE*, A. Nur Zincir-Heywood, *Member, IEEE*,
Malcolm I. Heywood, *Member, IEEE*
Faculty of Computer Science, Dalhousie University, Halifax, NS, Canada

***Abstract*— Although adaptive and heuristic approaches perform well under idealized conditions to the packet network routing problem, such algorithms are also dependent on global information that is not available under real-world conditions. This work benchmarks routing under local information conditions using the AntNet algorithm and makes recommendations regarding future approaches.**

***Index terms*—Swarm Intelligence, Network Routing.**

I. Introduction

Networked information systems and telecommunication in general rely on a combination of routing strategies and protocols to ensure that information sent by a user is actually received at the desired remote location. In addition, the distributed nature of the problem means that multiple users can make requests simultaneously. This results in delayed response times, lost information or other reductions to the quality of service objectives on which users judge network service. Routing is the process used to determine how a packet gets from source to destination. Protocols are used to implement handshaking activities such as error checking and receiver acknowledgements. In this work we are interested in the routing problem on computer networks.

The routing problem has several properties, which make it particularly challenging. In particular, the problem is distributed in nature; hence a solution that assumes access to any form of global information is not desirable. The problem is also dynamic; hence a solution that is sufficient for presently experienced network conditions may well be inefficient under other loads experienced by the network. Moreover, the traffic experienced by networks is subject to widely varying load conditions, making it impossible to design for 'typical' network conditions.

Traditionally, routing strategies are implemented through the information contained in routing tables available at each node in the network. That is, a table details the next 'hop' a packet takes based on the overall destination of the packet. This should not be taken to imply that a routing table consists of an exhaustive list of all destinations – this is a form of global information. Instead the table consists of specific entries for the neighboring nodes and then a series of default paths for packets with any other destination. Application of a classical optimization technique to such a problem might take the form of first assessing the overall pattern of network traffic, and then defining the contents of each routing table such that congestion is minimized. Such an approach does not generally work in practice as it simply costs too much to collect the information on a regular basis, where regular updating is necessary in order to satisfy the dynamic nature of network utilization. We therefore see the generic objectives of a routing strategy to be both dynamically reconfigurable and be based on locally available information, whilst also satisfying the user quality of service objectives (i.e. a global objective).

Several approaches have been proposed for addressing these objectives including: active networking [1], social insect metaphors [2], [3], cognitive packet networks [4] and what might be loosely called other 'adaptive' techniques [5]. The latter typically involve using evolutionary or neural techniques to produce a 'routing controller' as opposed to a 'routing table' at each node, where the controller typically requires knowledge of the global connectivity to ensure a valid route. Both the social insect metaphor and the cognitive packet approach provide a methodology for routing, without such a constraint; by using the packets themselves to investigate and report network topology and performance.

All methods as currently implemented, however, suffer from one drawback or another. Cognitive packet networks and active networking algorithms attempt to provide routing programs at the packet level, hence achieving scalable run time efficiency becomes an issue. To date, implementations of 'adaptive' techniques and social insect metaphors have relied, at some point, on the availability of global information

The purpose of this work is to simulate an example of routing as based on the social insect metaphor [2] and investigate the performance of such a system under the local information constraints typically available in practice. That is to say, previous works have employed routing tables that list all possible destinations, where this is not the case in practice. Section II introduces the 'ant' based social insect metaphor, used to provide the routing strategy. Experiments and results are presented in section III, and conclusions are drawn in section IV.

II. Social Insect Metaphor to Routing

As indicated above, active networking [1] and cognitive packet [4] based approaches emphasize a per packet mechanism for routing. The aforementioned 'adaptive'

techniques [5] tend to emphasize adding 'intelligence' to the routers leaving the packets unchanged. A social insect metaphor provides a middle ground, in which the concepts of a routing table and data packets still exist, but in addition, intelligent packets – *ants* – are introduced that interact to keep the contents of the routing tables up to date. To do so, the operation of ant packets is modeled on observations made regarding the manner, in which worker ants use chemical trails as a method of indirect *stigmergic* communication. Specifically, ants are only capable of simple stochastic decisions influenced by the availability of previously laid stigmergic trails. The chemical denoting a stigmergic trail is subject to decay over time, and reinforcement proportional to the number of ants taking the same path. Trail building is naturally a bi-directional process, ants need to reach the food (destination) and make a successful return path, in order to significantly reinforce a stigmergic trail (Forward only routing has also been demonstrated [3]). Moreover, the faster the route, then the sooner the trail is reinforced. An ant on encountering multiple stigmergic trails will *probabilistically* choose the route with greatest stigmergic reinforcement. Naturally, this will correspond to the 'fastest' route to the food (destination). The probabilistic nature of the decision, however, means that ants are still able to investigate routes with a lower stigmergic trial.

This approach has proved to be a flexible framework for solving a range of problems including the traveling salesman problem [6] and the quadratic assignment problem [7]. The work reported here follows the 'AntNet' algorithm of Di Caro and Dorigo [2], and is informally summarized as follows,

- Each node in the network retains a record of packet destinations as seen on data packets passing through that node. This is used to periodically, but asynchronously, launch 'forward' ants with destinations stochastically sampled from the collected set of destinations;
- Once launched, a forward ant uses the routing table information to make probabilistic decisions regarding the next hop to take at each node. While moving, each forward ant collects time stamp and node identifier information, where this is later used to update the routing tables along the path followed;
- If a forward ant re-encounters a node previously visited before reaching the destination, it is killed (in other words, identification of a loop in the path);
- On successfully reaching the destination node, total trip time is estimated, and the forward ant converted into a backward ant;
- The backward ant returns to the source using exactly the same route as recorded by the forward ant. Instead of using the data packet queues, however, the backward ant uses a priority queue;
- At each node visited by the backward ant, the corresponding routing table entries are updated to reflect the relative performance of the path;
- When the backward ant reaches the source, it dies.

Simulation of the above AntNet scheme has been shown to provide a robust alternative to six standard routing algorithms – OSPF, SPF, BF, Q-R, P-QR and Daemon [2]. However, an assumption is made, which inadvertently implies the use of 'global' routing table information. That is to say, the definition of routing tables is such that it is assumed that every node has a unique location in the routing table, Table I. Thus, if there are a total of L neighboring nodes and K nodes in the entire network, then there are $L \times K$ entries in the table. In practice, this is never the case. Instead, routing tables consist of specific entries for the neighboring nodes and then a series of default paths for packets with any other destination – such as OSPF or BGP4 [8]. Table II, in which case there are only L×2 entries in the routing table, summarizes the 'local' routing table information format employed here, where every node sees only its neighbors, i.e. global information is not introduced.

TABLE I
ORIGINAL ROUTING TABLE AT ANY NETWORK NODE K ON THE JAPANESE NTTNET BACKBONE

	All Network Nodes (Possible Destinations)			
Outgoing Links (Each node has L neighboring links)	$P_{1,1}$	$P_{1,2}$	------	$P_{1,57}$
	$P_{2,1}$	$P_{2,2}$	------	$P_{2,57}$
	------	------	------	------
	$P_{L,1}$	$P_{L,2}$	------	$P_{L,57}$

TABLE II
PROPOSED ROUTING TABLE AT ANY NETWORK NODE K THE JAPANESE NTTNET BACKBONE

	Neighbor Node	If used for other nodes
Outgoing Links (Each node has L neighboring links)	$P_{k,1}$	$P_{k,d}$ $d \neq 1$
	$P_{k,2}$	$P_{k,d}$ $d \neq 2$
	------	------
	$P_{k,L}$	$P_{k,d}$ $d \neq L$

In the case of the "global table", as originally used in the AntNet algorithm, if the network configuration changes, then all nodes will be updated with additional rows and columns. Moreover, as forward ants propagate across the network the amount of information they need to 'carry' also increases (node identifier and time stamp).

Finally, the availability of globally synchronized time is also assumed.

In order to avoid the use of the aforementioned global information, the following methods are therefore employed,

- Routing tables only detail the neighboring nodes. Such a limitation therefore places greater emphasis on the learning capacity of the ant. This is particularly significant during step (2) of the ant forward pass (sub-section II-A). Tables 1 and 2 illustrate the difference in available information for a node in the commonly used Japanese benchmark backbone (NTTNet) routing problem, Figure 1;
- Each node has a buffer in which forward ants deposit time stamp and identifier for the previous node. As will become apparent in the following section, it is only the inter-node information, which is important;
- Time synchronization is treated as a protocol issue. That is to say, during low load conditions each node is responsible for letting neighboring nodes know what their current time clock is. Moreover, whenever interrupts to services are sustained, then the first step once a node returns to operation will be to reinitiate local time references.

In the following section, the details of the AntNet algorithm [2] are summarized in order to better understand the significance of the above modifications.

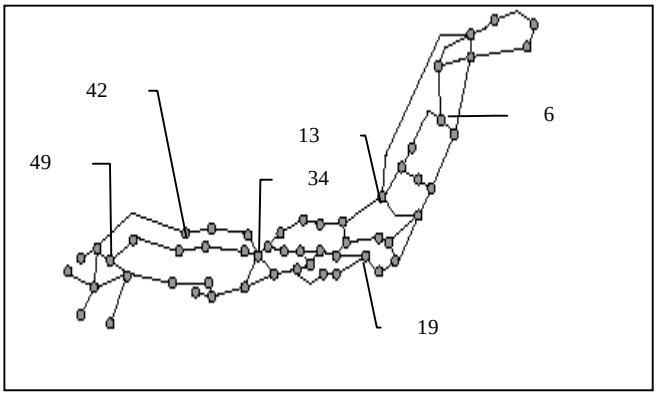

Fig 1. Japanese NTTnet backbone (55 nodes)

A. AntNet Algorithm

It is assumed that routing tables, T_k, exist at each node, k, in which a routing decision is made. Tables consist of 'n' rows, one row for each neighboring node/link. As far as a normal data packet is concerned, if the destination, d, from current node, k, is a neighbor then routing is deterministic ($P_{d,d} = 1$, where L = d). In all other cases, a route is selected based on the neighbor node probabilities.

1. New forward ants, F_{sd}, are created periodically, but independently of the other nodes, from source, s, to destination node, d, in proportion to the destination frequency of passing data packets. Forward ants travel the network using the same priority structures as data packets, hence are subject to the same delay profiles.
2. Next link in the forward ant route is selected stochastically, p'(j), in proportion to the routing table probabilities and length of the corresponding output queue.
$$p'(j) = \frac{p(j) + \alpha l_j}{1 + \alpha(|N_k| - 1)}$$
where $p(j)$ is the probability of selecting node j as the next hop; α weights the significance given to local queue length verses global routing information, $p(j)$; l_j is the queue length of destination 'j' normalized to the unit interval; and N_k is the number of links from node k.
3. On visiting a node different from the destination, a forward ant checks for a buffer with the same identifier as itself. If such a buffer exists the ant must be entering a cycle and dies. If this is not the case, then the ant saves the previously visited node identifier and time stamp at which the ant was serviced by the current node in a buffer with the forward ant's identifier. The total number of buffers at a node is managed by attaching "an age" to buffer space and allowing backward ants to free the corresponding buffer space.
4. When the current node is the destination, $k = d$, then the forward ant is converted into a backward ant, B_{ds}. The information recorded at the forward ant buffer is then used to retrace the route followed by the forward ant.
5. At each node visited by the backward ant, routing table probabilities are updated using the following rule,
IF (node was in the path of the ant)
THEN $p(i) = p(i) + r\{1 - p(i)\}$
ELSE $p(i) = p(i) + r\,P(i)$
where $r \in (0, 1]$ is the reinforcement factor central to expressing path quality (length), congestion and underlying network dynamics.

As indicated above, the reinforcement factor should be a factor of trip time and local statistical model of the node neighborhood. To this end [2] recommend the following relationship,

$$r = c_1\left(\frac{W_{best}}{t_{ant}}\right) + c_2\left\{\frac{I_{\sup} - I_{\inf}}{(I_{\sup} - I_{\inf}) + (T - I_{\inf})}\right\} \quad (1)$$

where W_{best} is the best case trip time to destination d over a suitable temporal horizon, W; t_{ant} is the actual trip time taken by the ant; $I_{inf} = W_{best}$; $I_{sup} = \mu_{kd} + W^{0.5}\{\sigma_{kd} / (1 - \gamma)\}$.

The estimates for mean, μ_{kd}, and variant, σ_{kd}, of the trip time are also made iteratively, using the trip time information thus,

$$\mu_d = \mu_d + \eta(o_{kd} - \mu_d)$$
$$(\rho_d)^2 = (\rho_d)^2 + \eta\{(o_{kd} - \mu_d)^2 - (\rho_d)^2\}$$

From the above algorithm, it is, therefore, apparent that ants are required to make decisions under more uncertainty than was previously the case. Moreover, the trip time information is updated incrementally based on the recorded trip duration between current node, *k*, and ultimate destination, *d*. This means that it is no longer necessary to carry all node and duration information as a 'stack' to the target duration as in the original model [2]. Only the previous step information is therefore necessary.

III. Simulations

To test the performance of the modified AntNet algorithm(local information only) we also simulated the original AntNet algorithm(global information). The two programs were written in C++ under UNIX environment.

The event driven simulation models the network as routers (nodes) and links. Every router has an incoming buffer, a memory space for processing packets, and an outgoing buffer for each link to its neighbor routers. A priority queue is used to store the events. In order to let the two programs have the same input, an event generator is used to generate the events, such as the new packets, or routers going down and up. The following is the parameters that are used in the simulation:

- The forward ants are launched every 300ms;
- The AntNet algorithm is given 5 seconds time to converge the initial routing tables, during this period, ants are the only packets traversing the network;
- Packets are generated by Poisson distribution (mean of 35ms).

Measurements used to compare the two algorithms are network throughput, and the queue length due to network failures. The network throughput is defined as the bytes (packets) that arrive at the destination routers per time step. The simulation length is 1250s, as the result, 1984840 packets are generated within 1250s. The following figures (2-11) are the result of our simulations, where all ***dark black*** lines indicate results for AntNet running with '***global*** routing' tables, Table I, whilst *gray lines* show the results for the AntNet running with '*local* routing' tables, Table II.

Five different scenarios are considered. In the first case, all routers remain operational for the duration of the simulation, figures 2 and 3, hence the base line conditions are identified. Scenario 2, figures 4 and 5, removes node 34, a degree 5 node that plays a predominant role in establishing network connectivity in the lower left of the Japanese backbone, figure 1. Two nodes (nodes 13 and 49) are removed in scenario 3, figures 6 and 7, and 3 nodes (nodes 6, 19 and 42) in scenario 4, figures 8 and 9. The last scenario removes 2 nodes (nodes 13 and 49), but at different times (13 at 300s, 49 at 500s), and lets them be repaired (both at 800s) before the simulation completes, figures 10 and 11.

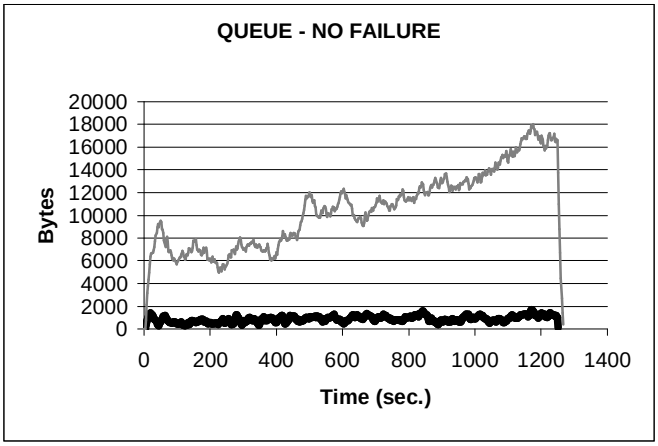

Fig 2. Queue Length - No network failure on the network

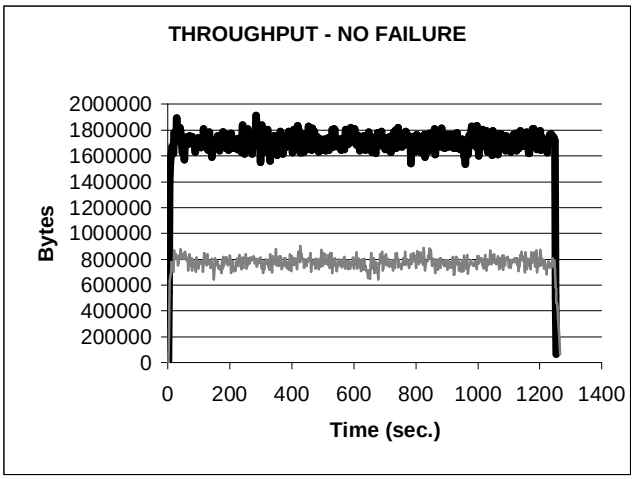

Fig 3. Throughput – No network failure on the network

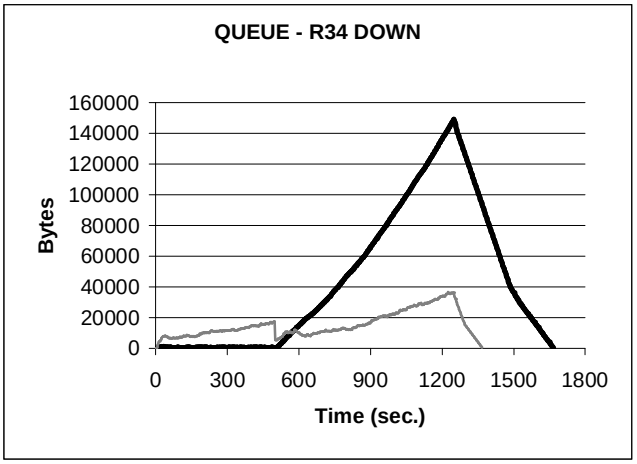

Fig 4. Queue Length – A critical router (degree 5) goes down at time 500 sec.

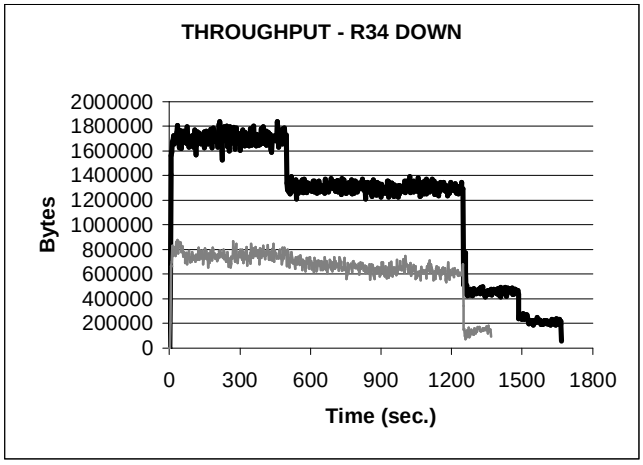

Fig 5. Throughput – A critical router (degree 5) goes down at time 500 sec

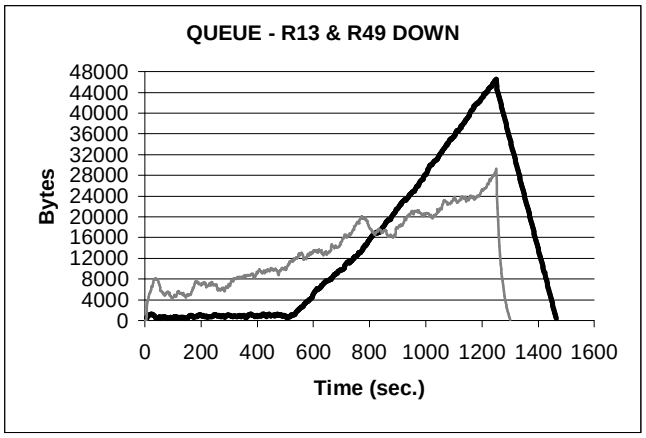

Fig 6. Queue Length – Two routers (one has degree 4, one has degree 3 respectively) go down at time 500sec.

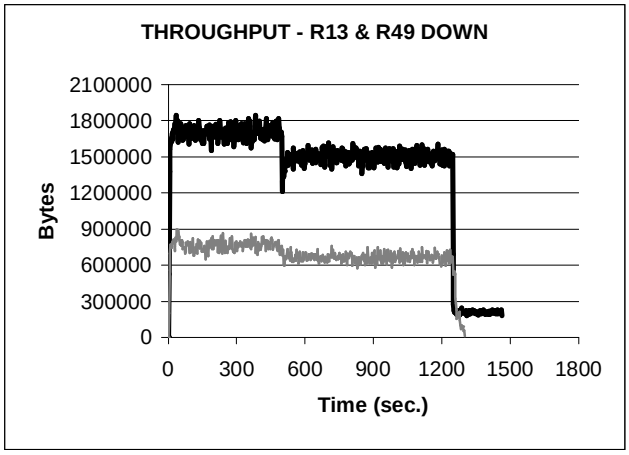

Fig 7. Throughput – Two routers (one has degree 4, one has degree 3 respectively) go down at time 500sec.

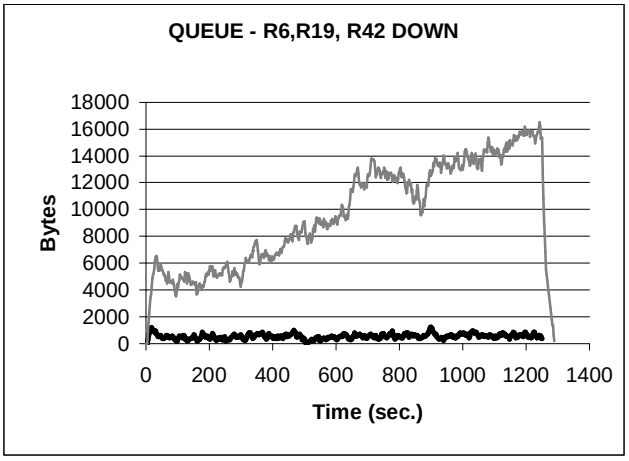

Fig 8. Queue Length – Three routers (they have degree 3, 3, 2 respectively) go down at time 500sec.

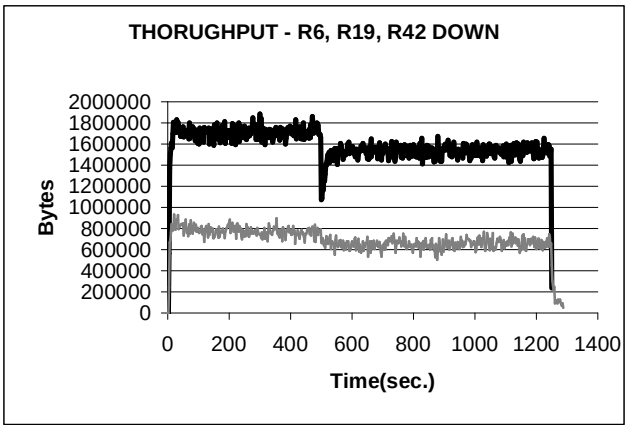

Fig 9. Throughput – Three routers (they have degree 3, 3, 2 respectively) go down at time 500sec.

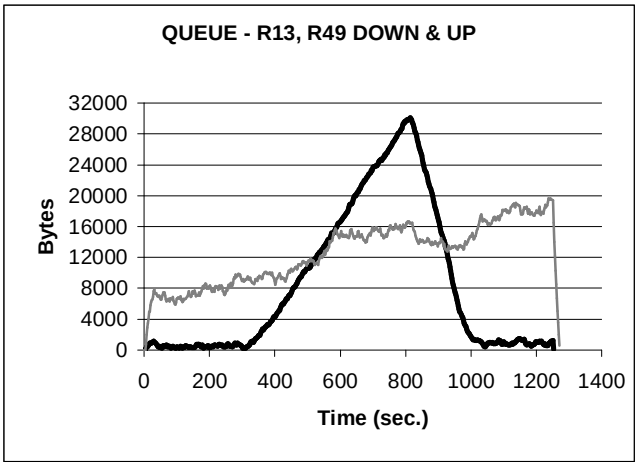

Fig 10. Queue Length – One of the two routers (same as in figures 4-5) go down at time 300sec; repaired at time 600sec. Second router goes down at time 500sec; repaired at time 800sec.

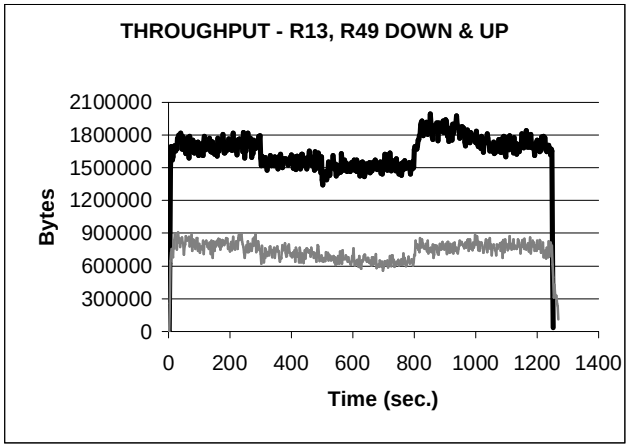

Fig 11. Throughput – One of the two routers (same as in figures 4-5) go down at time 300sec; repaired at time 600sec. Second router goes down at time 500sec; repaired at time 800sec.

From figures 2 and 3 it is readily apparent that the availability of global information makes a significant impact on the quality of the route. The global case effectively returning twice the throughput and an order of magnitude reduction in the queue length of the local information case. In the following 4 scenarios, the global case appears to be more sensitive, with worse queue lengths for all but the case of 3 nodes removed. However, this is deceptive as the number of lost packets under the local information AntNet is much higher, actually reducing queue lengths significantly! Thus, the AntNet algorithm under global information is much more desirable in all circumstances.

IV. Conclusion

The routing algorithms current used on the Internet are centralized, with some central routers, e.g., at the border gateways, taking the responsibility to redistribute routing information. This leaves the network engineer with the opportunity to tune router configurations according to their experience. Unfortunately, the network topology, and the traffic pattern are subject to continual change. This requires the network engineers to adjust the parameters for better performance. Distributed adaptive and intelligent routing algorithms have therefore become more and more important; in this way, the network can set up the routing tables itself, and adjust them automatically.

This paper characterizes performance from the AntNet routing algorithm under local information constraints. By doing so the aim of the authors is to demonstrate the importance of including this limitation when attempting to solve the general case of the packet routing problem (distributed dynamic information). Such a shortcoming is not unique to the AntNet algorithm, but a factor of heuristic and adaptive techniques the authors are currently aware of. In order to address these issues, increased intelligence will be necessary on the 'agent' side. In the specific case of the AntNet algorithm one possible solution would be to explicitly make the 'agent' responsible for learning the reinforcement in equation (1). Such an agent would require memory properties, e.g. recurrent interconnect of neural network models, and has indeed been initially proposed under the context of a different 'agent' routing framework (cognitive packet framework [3]). Such a solution would address the problem of letting the 'agents' see beyond the information retained in routing tables. However, under what conditions this is sufficient to ensure 'good' routes for the data packets is an open question.

The authors are currently working on approaches based on the co-evolution of routes by genetic algorithm based methods. In doing so, the objective is to provide a more explicit link between the routes identified by 'agents' and the information that data packets are routed against.

Acknowledgements

A. Nur Zincir-Heywood and Malcolm I. Heywood gratefully acknowledge the support of the Individual Research Grants from the Natural Sciences and Engineering Research Council of Canada.

Reference

[1] Tennenhouse D., Smith J., Sincoskie W., Wetherall D., Minden G., "A Survey of Active Network Research," IEEE Communications Magazine, 35(1), pp 80-86, Jan 1997.

[2] Di Caro G., Dorigo M., "AntNet: Distributed Stigmergetic Control for Communications Networks," Journal of Artificial Intelligence Research, 9, pp 317-365, 1998.

[3] Gelenbe E., Xu Z., Seref E., "Cognitive Packet Networks," Proceeding of 11th IEEE International Conference on Tools with Artificial Intelligence, pp 47-54, 1999.

[4] Corne D.W., Oates M.J., Smith G.D., Telecommunications Optimization: Heuristic and Adaptive Techniques. John Wiley & Sons, ISBN 0-471-98855-3, 2000.

[5] Dorigo M., Maniezzo V., Colorni A., "Ant System: Optimization by a Colony of Cooperating Agents," IEEE Transactions on Systems, Man and Cybernetics – B, 26(1) pp 29-41, Feb 1996.

[6] Heusse M., Snyers D., Guerin S., Kuntz P., "Adaptive Agent-driven Routing and Load Balancing in Communication Networks," Advances in Complex Systems, 1, pp 237-254, 1998.

[7] Maniezzo V., Colorni A., "The Ant System Applied to the Quadratic Assignment Problem," IEEE Transactions on Knowledge and Data Engineering, 11(5), pp 769-778, Sept/ Oct 1999.

[8] Halabi B., "Internet Routing Architectures", Cisco Systems Cisco Press, ISBN 1-56205-652-2, 1997.