Paper

Swarm intelligence for network routing optimization

Peter Dempsey and Alfons Schuster

Abstract— This paper presents the results of a comparative study of network routing approaches. Recent advances in the field suggest that swarm intelligence may offer a robust, high quality solution. The overall aim of the study was to develop a framework to facilitate the empirical evaluation of a swarm intelligence routing approach compared to a conventional static and dynamic routing approach. This paper presents a framework for the simulation of computer networks, collection of performance statistics, generation and reuse of network topologies and traffic patterns.

Keywords— network routing, swarm intelligence, ant algorithms.

1. Introduction

Computer networks are handling larger amounts of traffic and experience tells us that they will continue to grow in size and demand for efficiency. Improvements in network hardware will offer performance improvements but need to rely upon intelligently designed policies and protocols to achieve optimality. Many of the problems that are encountered in the design of network communication policies have no easy or completely satisfactory solution. This is because they are often compromises between conflicting objectives. The problem that this paper investigates is known as network routing.

2. Routing and ant metaheuristic

Consider a network of devices, either hosts or routers, connected by point-to-point communication links. Any device may communicate with another by sending an addressed data packet to a neighbour who will then forward the packet on, eventually to the intended recipient. The decision of which outward link to send an incoming packet along is made by the network layer's routing algorithm. Routing is an important aspect of computer networks because it can greatly influence overall network performance; good routing can cause greater throughput or lower average delays, all other conditions being the same [6]. Routing is a difficult problem because it is a distributed multi-objective optimization problem. This has two important implications:

- Because the problem is distributed it is impossible for any one device to have an accurate picture of the overall problem state at the time when it must make decisions affecting the performance of the network.
- A good solution to the problem will be a compromise between conflicting requirements. For example, throughput is desirable but not to the extent that it will unfairly penalise some hosts.

Traditional approaches to network routing include static [13] routing algorithms and various dynamic routing approaches [4, 8]. Static routing algorithms compute the least costly paths through the network when the network is first booted using known information about the communication links used. The algorithm used in our simulation framework used Djikstra's shortest path algorithm calculating the distance using cost weightings for the communication links [13]. The obvious disadvantage of such an approach is that it is unable to adjust its policy to minimise the build-up of localised congestion in one area of the network. Also the network must implement another protocol to handle the failure of a communication link. The dynamic routing algorithm treated in this paper is a distance-vector algorithm similar to *Routing Information Protocol* (RIP) [2]. In this approach nodes (hosts or routers) periodically send a packet to each neighbour notifying them of the minimum number of packets that are queued along their best route to every other node in the network. When a node learns of a better route to a node it rewrites its routing table to begin using the new route. This gives dynamic routing algorithms the ability to direct traffic around congested areas to relieve congestion. This flexibility can backfire resulting in the situation where traffic is diverted between routes in constant oscillations that increase congestion in the local area. Invulnerability to this effect is an advantage of static routing algorithms over dynamic routing algorithms. Swarm intelligence routing approaches seem to offer the flexibility of distance-vector algorithms without the drawback of undamped traffic oscillations [6].

3. Ant metaheuristic

Swarm intelligence is a soft computing technique that has gained considerable attention in the research community over the last couple of years [1, 5, 6]. It was proposed for various tasks including the control of robot swarms, power saving in mobile networks and network routing, for example [6, 11].
The swarm intelligence approach we use is an ant routing algorithm. An important characteristic of swarm intelligence is its use of stigmergy. Stigmergy refers to a commu-

nication method that encodes information about the problem (and its solutions) on the environment of the problem. A good example of stigmergy in swarm intelligence is the ant metaheuristic. This refers to the method by which ant swarms find best paths. Information is encoded on the environment in the form of pheromone (scent chemicals) deposited on the ground as the ants walk over it. In the case illustrated in Fig. 1 ants continuously pour out of the nest toward the food. When a fork in the path at A is encountered ants choose either path with a probability based

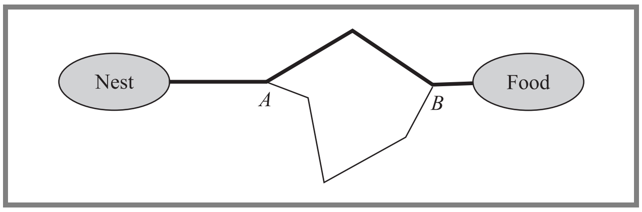

Fig. 1. An example of the shortest path-finding behaviour.

on the concentration of pheromone on that path. Initially unscented the swarm will divide evenly between the paths. The ants that took the shortest path will reach the food first and turn back toward the nest. When they reach the fork at B the shortest path will be more strongly scented because more ants will have emerged from it. The scent of the shortest path will be reinforced until the whole swarm converges on the shortest path.

3.1. Ant routing

The ant metaheuristic has lead to the creation of ant algorithms such as the AntNet system [6]. Ant routing algorithms generate ants (packets with random addresses) that traverse the network and collect timestamps as they pass through each node. The ants are routed by a stochastic process that is weighted by the goodness of a particular route [6]. When an ant reaches its destination it generates a backward ant which follows the same route as the original ant back to its source. As the backward ant travels through each node it updates the stigmergy table, which holds the goodness values for the different routes. An example of a stigmergy table typically produced in our study is illustrated in Table 1.

The table is consulted when sending an ant to the node whose address is the column name. The values represent the probabilities of using the node whose address is the row name as the next node on the path. The table is taken from node 0. Note how the sum of probabilities in each column is 1.

Each node also maintains a data structure that contains, for each other node in the network, the mean delay to that node (μ), the variance of observed delays (σ^2) and the number of observations (n) that contributed to μ and σ^2. This data structure is used when calculating the reinforcement to the goodness of a route.

First a raw measure (r) of the goodness of the reported time (t) is calculated as illustrated by Eq. (1):

$$r = \frac{t}{2\mu}, \tag{1}$$

with out-of-scale (> 1) values of r being saturated to 1.

If the mean is considered unstable ($\sigma^2/\mu \geq 0.25$) then the value for r is corrected as follows: for good values of r ($r < 0.5$) a value U is added to r, and for bad values of r ($r \geq 0.5$) U is subtracted from r, where:

$$U = 0.1^{\sigma^2/\mu}. \tag{2}$$

This rule mitigates the costly oscillatory behaviour that arises in dynamic routing algorithms by reducing the reinforcement effect if the mean delay to a node fluctuates.

If the mean is considered stable ($\sigma^2/\mu < 0.25$) then the value for r is corrected as follows: for good values of r ($r < 0.5$) a value S is subtracted from r, and for bad values of r ($r \geq 0.5$) S is added to r, where:

$$S = 0.1^{2\sigma^2/\mu}. \tag{3}$$

This rule amplifies the reinforcement effect for routes with stable mean delays, effectively increasing the learning rate when we trust observations and believe that it is safe to do so.

The corrected r is used to update the stigmergy table using the following rules.

For the probability (P_0) for the neighbour which the time relates to:

$$P_0' = P_0 + (1 - r)(1 - P_0). \tag{4}$$

For all other neighbour's probabilities yields:

$$P_n' = P_n + (1 - r)P_n. \tag{5}$$

The rule for updating the values in the stigmergy table simply ensures that the value being reinforced is increased in proportion to the goodness (r). The other values are decreased in proportion to r and their own relative magnitudes while keeping the sum of all the probabilities equal to 1.

Data packets are not routed stochastically, they are always sent to the neighbour with the greatest goodness value for the intended destination. When forward ants revisit a node the circuit that they have travelled in is cleared from their memory to avoid reinforcing circular routes. To attempt to provide a faster feedback mechanism backward ants have priority over all other packets. A common criticism of this system is that a faster yet feedback mechanism would be to design forward ants to update the routing tables of nodes with regard to the section of the trip that they already completed. To this we respond that an essential feature of the ant metaheuristic is that the reinforcement from poor routes must be delayed proportionally.

Table 1
A stigmergy table from node 0 (taken from simulation output)

Address	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	0.99993	$3.45 \cdot 10^{-5}$	0	0.00475	0.113151	0.22	0	0	0.03413
2	0	$7.48 \cdot 10^{-5}$	0.61347	0	0.00475	0.113151	0.195	$2.74 \cdot 10^{-5}$	0.436324	0.03413
3	0	0	0.23308	0.185908	0.00475	0.127658	0.195	0	0	0.8635
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0.13051	0	0.02975	0.113151	0.195	0.99997	0	0.03413
8	0	0	0.02291	0.814092	0.95599	0.532888	0.195	0	0.563676	0.03413
9	0	0	0	0	0	0	0	0	0	0

3.2. Traffic patterns and network topology

The traffic generator generates a random number of packets less than ten with random sizes below 1000 bytes. Self-addressed packets will not be generated. Traffic patterns are saved for reuse during simulations with different routing approaches. At the beginning of each simulation every node is preloaded with its traffic pattern after which no more traffic is added to the nodes.

The network generator included in the framework is a random network generator (as opposed to scale-free). In our study the network generator was typically used to generate a network of 10 nodes with a probability of any pair of nodes being connected at 0.35. A graph traversal function guarantees that only connected graphs proceed to simulation. The generator will not create self-to-self arcs. Topologies are saved for reuse during simulations with different routing approaches.

The choice to not use a scale-free network generator was an important one. For more than 40 years the study of networks was based on work by Paul Erdös and Alfréd Rényi. They suggested, in 1959, that networks could be described by nodes connected by randomly placed links. While their work revolutionised graph theory it has since been shown that scale-free networks are much more common. The physical structure of the Internet and the link structure of the world wide Web (WWW) have both been shown exhibit scale-free organisation [7]. The choice of generation method and implementation of a suitable algorithm is a considerable undertaking as can be seen in the recent work of Spencer and Sacks, for example [12]. For this study we have decided to simulate random networks only.

3.3. Characteristics of random networks

The random networks are those that are formed by the creation of links between randomly chosen pairs of nodes. Random networks are also known as exponential networks because the probability that a node is connected to k other nodes decreases exponentially as k increases. If the frequency of nodes is plotted against the number of links a normal distribution is evident as in Fig. 2.

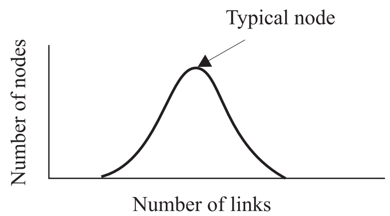

Fig. 2. Poisson distribution of node linkages in random networks.

One interesting point of random networks is that they are vulnerable to fragmentation by the removal of a number of randomly chosen nodes leaving the remaining nodes hopelessly separated from most of the rest of the network.

3.4. Characteristics of scale-free networks

Scale-free networks consist of clusters of nodes connected to a central hub that is connected to other hubs like it. They are characterised by power-law distribution of node linkages. This means that the probability of a node being connected to k other nodes is $1/k^n$. Scale-free networks all seem to have values of n between two and three. So for example, a node is four times as likely to have only half the number of links another node has. Figure 3 illustrates this behaviour.

In scale-free topologies the vast majority of nodes have roughly the average number of links, but a few "hubs" have thousands times the average number of links. When plotted on a double logarithmic scale the node linkage distribution is a straight line. This behaviour is illustrated in Fig. 4.

In contrast to random networks, scale-free networks are resilient to the removal of randomly chosen nodes to a high degree. As many as 80 percent of randomly selected nodes can fail and the remainder will still form a compact con-

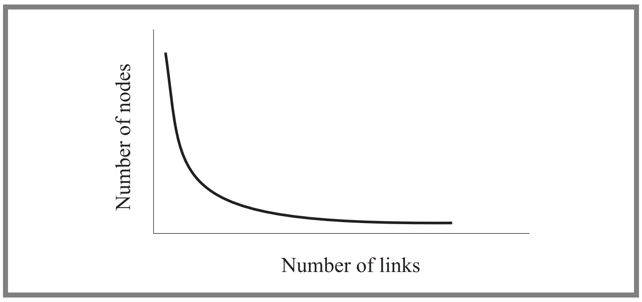

Fig. 3. Power-law distribution of node linkages in scale-free networks.

nected cluster [3]. However they are very sensitive to the removal of selected hubs. In fact, scale-free networks are only more robust to node removal than random networks if more than 95 percent of the removed nodes are chosen at random [10].

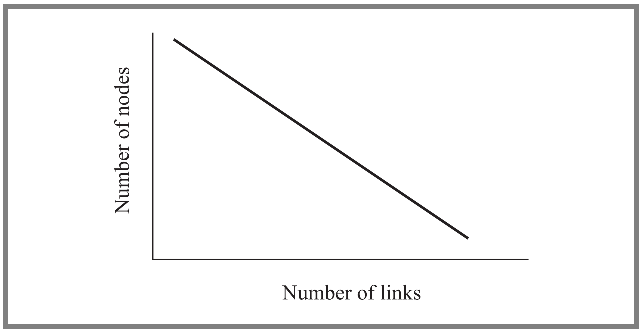

Fig. 4. Power-law distribution plotted on double logarithmic scale (scale-free networks).

The peculiar characteristics of scale-free networks are due to the way they are created. The Internet did not come into being as a set of randomly connected routers and hosts, nor do new nodes attach themselves to the network at random points. The Internet grew. When new users become connected there are reasons why they connect at specific points. The mechanisms which create scale-free networks are *growth* and *preferential attachment*. Growth implies that the organisations (nodes) that are oldest will have accumulated more links. Preferential attachment refers to the fact that the more links a node has the more attractive it is to be connected to. Other factors accentuate this effect; for example strategic positioning of service providers and users clustering around a preferred service.

4. Results

The results in this study are based on five simulation runs. Each run uses the same traffic patterns and topology for each of the three routing approaches discussed. In the case of critical events, the framework outputs event tags to delimited text files. These files were then analysed by importing into a spreadsheet. The results of the final analysis are illustrated in Table 2.

The values provided are mean values across the five simulations. Static routing offers the best throughput of the three approaches but dynamic routing offers shorter mean packet delays. This would suggest that the dynamic routing

Table 2
Summary of results from analysis of simulation output

Metric	Routing approach		
	static	dynamic	ant
Mean packet delay [ms]	1537.78	1457.00	3878.30
No. packets delivered	36	36	35.8
Time taken [ms]	6006	7362	17 388
Total data transfer [B]	17 537	17 537	17 537
Throughput [B/s]	2.92	2.39	1.03
Total busy time [ms]	31 455.60	34 446.60	61 104.40
Percentage utilisation	52.44	46.94	35.76

algorithm sacrifices raw efficiency for fairness (keeping the average delay of packets lower). In this case ant routing shows inferior performance when compared to the other two. We contend that this is due to a coarse system of adjusting the learning rate which would be improved with fine-tuning.

5. Summary and future work

We developed a framework for simulation of a computer network and implemented static, dynamic and ant routing algorithms. We collected 15 results sets in total from five simulations. Upon analysis we find static and distance-vector routing perform similarly. Our ant routing algorithm performs sub optimally but demonstrates the principle of stigmergetic communication successfully.

The poor performance of the ant algorithm will be investigated further with special consideration given to the learning rate adjustments. Swarm intelligence may yet not prove more efficient than traditional network routing algorithms but its ability to self-organise operating purely on local information may prove useful in *ad hoc* networks like the Bluetooth world, for example [9]. An investigation of other network topologies, using our simulation framework, such as scale-free networks would be useful. Also the work can be taken into new fields. For example, consider a network controlled by ants using stigmergy to communicate information on billing, virus infection, hardware failure, usage patterns, etc. One can also envision a genetic algorithm evolving the different ant variants and producing super-ants tailored to a cable company's own network [14]. We believe however, that great caution must be exercised in the application of fitness criteria to ants. Ants in a technical sense are computer viruses, it is not hard to imagine what harm could be done if they spread across networks or evolved to escape detection.

References

[1] "ANON – the swarm development group", 2004, http://wiki.swarm.org/

[2] B. Baccala, "Connected: an Internet encyclopedia", 2004, http://www.freesoft.org/CIE/Topics/90.htm

[3] A. Barabási and E. Bonabeau, "Scale-free networks," *Sci. Amer.*, vol. 288, no. 5, pp. 50–59, 2003.

[4] R. Bellman, *Dynamic Programming*. New Jersey: Princeton University Press, 1957.

[5] E. Bonabeau, M. Dorigo, and Z. Theraulaz, *Sarm Intelligence: From Natural to Artificial Systems*. New York: Oxford University Press, 1999.

[6] G. Di Caro and M. Dorigo, "AntNet: distributed stigmergetic control for communications networks", *J. Artif. Intel. Res.*, vol. 9, pp. 317–365, 1998.

[7] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of Internet topology", in *Conf. Appl., Technol., Architect. Protoc. Comput. Commun. SIGCOMM'99*, Cambridge, USA, 1999, pp. 251–262.

[8] L. Ford jr. and D. Fulkerson, *Flows in Networks*. Princeton, New Jersey: Princeton University Press, 1962.

[9] M. Heissenbuettel and T. Braun, "Ants-based routing in large scale mobile ad hoc networks". Kommunikation in verteilten Systemen KiVS'03, Leipzig, Germany, 2003.

[10] S. Park, A. Khrabrov, D. Pennock, S. Lawrence, C. Giles, and L. Ungar, "Static and dynamic analysis of the Internet's susceptibility to faults and attacks", in *22nd Ann. Joint Conf. IEEE Comput. Commun. Soc. Infocom'03*, San Francisco, USA, 2003.

[11] Ch.-Chu. Shen and C. Srisathapornphat, "Ant-based energy conservation for ad hoc networks", Tech. Rep. no. 2004-09, DEGAS Group, Dept. of CIS, University of Delaware, 2004.

[12] J. Spencer and L. Sacks, "Modelling IP network topologies by emulating network development processes", in *Int. Conf. Softw., Telecommun. Comput. Netw. Softcom'02*, Split, Croatia, 2002.

[13] A. S. Tannenbaum, *Computer Networks*. 4th ed. New Jersey: Prentice Hall, 2003.

[14] M. Ward, "There's an ant in my phone...", *New Sci.*, vol. 157, pp. 32–36, 1998.

Peter Dempsey holds a B.Eng. in software engineering from the University of Ulster at Jordanstown in Northern Ireland. His research interests are in telecommunication networks, swarm intelligence, and artificial intelligence. He has a year work experience in industry. He currently works as a language instructor in Japan.

e-mail: dempsey-p@ulster.ac.uk
University of Ulster
Shore Road, Newtownabbey, Co. Antrim, BT370QB
Northern Ireland

Alfons Schuster received the Ph.D. degree in computer science from the University of Ulster in Northern Ireland and the B.Sc. degree in applied physics form the University of Applied Sciences in Munich, Germany. He has several years experience working in industry. His research interests are in artificial intelligence, robotics and theories of computing.

e-mail: a.schuster@ulster.ac.uk
Faculty of Engineering
School of Computing and Mathematics
University of Ulster
Shore Road, Newtownabbey, Co. Antrim, BT370QB
Northern Ireland