ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 3, pp: (320-324), Month: July - September 2015, Available at: www.researchpublish.com

Network Load Balancing Using Ant Colony Optimization

Mr. Ujwal Namdeo Abhonkar[1], Mr. Swapnil Mohan Phalak[2],
Mrs. Pooja Ujwal Abhonkar[3]

[1,3]Lecturer in Computer Engineering Department
[2]Lecturer in Information Technology Department
[1,2]Sandip Polytechnic, Nashik, [3]LGG Polytechnic, Nashik, India

Abstract: **Ants first evolved around 120 million years ago, take form in over 11,400 different species and are considered one of the most successful insects due to their highly organized colonies, sometimes consisting of millions of ants. One particular notability of ants is their ability to create "ant streets". Long, bi-directional lanes of single file pathways in which they navigate landscapes in order to reach a destination in optimal time. These ever-changing networks are made possible by the use of pheromones which guide them using a shortest path mechanism. This technique allows an adaptive routing system which is updated should a more optimal path be found or an obstruction placed across an existing pathway. Computer scientists began researching the behavior of ants in the early 1990's to discover new routing algorithms. The result of these studies is Ant Colony Optimization (ACO) and in the case of well implemented ACO techniques, optimal performance is comparative to existing top-performing routing algorithms. This article details how ACO can be used to dynamically route traffic efficiently. An efficient routing algorithm will minimize the number of nodes that a call will need to connect to in order to be completed thus; minimizing network load and increasing reliability. An implementation of ANTNet based on Marco Dorigo & Thomas stutzle has been designed and through this a number of visually aided test were produced to compare the genetic algorithm to a non-generic algorithm. The report will final conclude with a summary of how the algorithm perform and how it could be further optimized.**

Keywords: **Algorithm, Ant Colony Optimization, Networks, Routing, Traffic.**

I. INTRODUCTION

Electronic communication networks can be categorized as either circuit-switched or packet-switched. Circuit-switched networks rely on a dedicated connection from source to destination which is made once at start-up and remains constant until the tear-down of the connection. An example of a circuit switched network would be British Telecoms telephone network. Packet-switched networks work quite differently however and all data to be transmitted is divided into segments and sent as data-packet. Data-packets can arrive out of order in a packets-switched network with a variety of paths taken through different nodes in order to get to their destination. The internet and office local area networks are both good examples of packet-switched networks.

A number of techniques can be employed to optimize the flow of traffic around a network. Such techniques include flow and congestion control where nodes action packet acknowledgements from destination nodes to either ramp-up or decrease packet transmission speed. The area of interest in this report concentrates on the idea of network routing and routing tables. These tables hold information used by a routing algorithm to make a local forwarding decision for the packet on the next node it will visit in order to reach its final destination.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 3, pp: (320-324), Month: July - September 2015, Available at: www.researchpublish.com

One of the issues with network routing (especially in very large networks such as the internet) is adaptability. Not only can traffic be unpredictably high but the structure of a network can change as old nodes are removed and new nodes added. This perhaps makes it almost impossible to find a combination of constant parameters to route a network optimally.

II. LITERATURE SURVEY

A. Routing algorithms:

Packet-switched networks dynamically guide packets to their destination via routing tables stored in a link state and are selected via a link state algorithm. The Link state algorithm works by giving every node in the network a connectivity graph of the network. This graph depicts which nodes are directly connected. Values are stored for connected nodes in map which represent the shortest path to other nodes. One such link state algorithm used in network routing is Dijkstras algorithm. When a path between two nodes is found, its weight is updated in the table. Should a shorter path be found the new optimal weight will be updated to the table replacing the old value; the algorithm allows traffic to be routed around the network whilst connecting to the least number of nodes as possible. The system works but doesn't take into account influx of traffic and load balancing.

III. METHODOLOGY

A. Introducing ANTNet:

By replacing Dijkstras algorithm with a generic algorithm, paths taken by calls could be scored by how short of a path they took, that way if they were queued on a busy network they would perform badly. Consequently other paths would score relative and be chosen. This would work in real time and allow the routing system to adapt as packets are transmitted ANTNet uses virtual pheromone tables much like when an ant follows a path dropping pheromones to re-enforce it. The quicker the ants move down a path the more throughput of ants thus; a greater concentration of pheromones. In the same way pheromone tables in ANTNet allow fast routes to score a higher chance of being selected whilst the less optimal route scores a low chance of being selected.

The idea behind ANTNet is when a call is placed an Ant will traverse across the network using a link-state deterministic algorithm. Every node holds a pheromone table for all other nodes of the network. Each pheromone table holds a list of table entries containing all the connected nodes of the current node.

B. The Algorithm:

To begin with, each possible path has an even likelihood of being chosen. An ant is placed on a network of 4 nodes with the source node of 1 and destination node 2. A chance mechanism is invoked and a path is chosen.

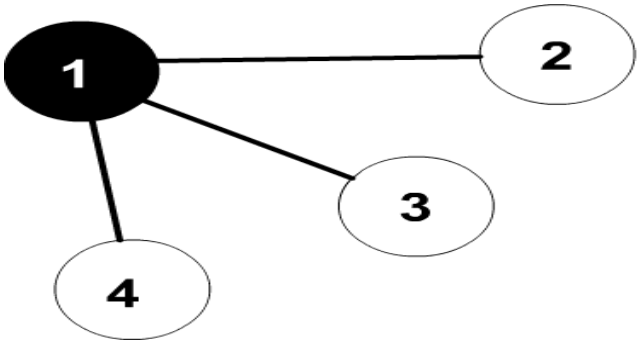

Figure I - The network graph

Table I - Pheromone table for node 1

Next Node	% chance
2	33.33333%
3	33.33333%
4	33.33333%

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 3, pp: (320-324), Month: July - September 2015, Available at: www.researchpublish.com

In this case node 2 has been selected [figure II] and the ant arrives at its source destination.

The ant then moves and updates the pheromone tables for the visited nodes with higher (and more mathematically biased) value. This would be calculated for figure II and table II in the following way:

- Node 2 was the final destination
- It took 1 hop to get to its destination
- Divide 1 (hop) by 100 : 100%
- Add 100 to the probability value of node 2 (currently 33.3333) : 133.3333
- Add the values of the other nodes to 133.3333 (133.3333+ 33.3333 + 33.3333) : 200 (approximately)
- Calculate the ratio : ratio = 100/200 0.5
- Set the probability of the node to its current value multiplied by the ratio
- Node 2: 133.3333 * ratio (0.5) = 66.6666%
- Node 3: 33.3333 * ratio (0.5) = 16.6666%
- Node 4: 33.3333 * ratio (0.5) = 16.6666%
- Node 2 (66.6666%) + Node 3 (16.6666%) + Node 4 (16.6666%) = 99.9999%

The system isn't 100% accurate as the total will never add up to exactly 100% but it will be close enough to allow accuracy within the level required. The following diagram depicts how the path and pheromone table after the update has taken place

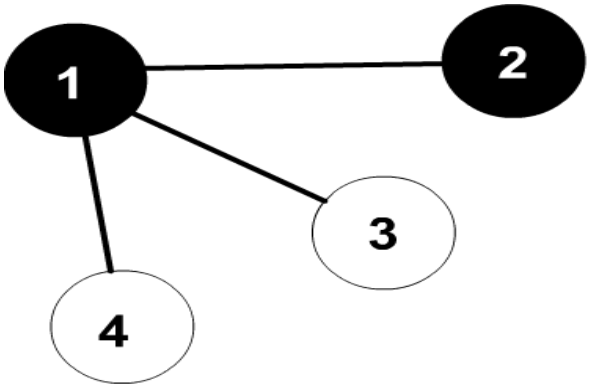

Figure II

Table II

Next Node	% chance
2	66.6666%
3	16.6666%
4	16.6666%

C. The Program:

For the purpose of this program a bi-directional, un-weighted topological network consisting of 30 nodes has been created and closely resembles the British Synchronous Digital Hierarchy (SDH) network. After a basic number of parameters have been set the simulation is run. Firstly all pheromone tables are defaulted to equal weights and then calls are generated and placed on the network. Initially the routes chosen are random. If a call cannot connect to a node it is forced to wait and the wait counter is enumerated to reflect the quantum (in timer ticks). Once a node has reached its destination node it will work its way backwards altering the local nodes pheromone table as it traverses. The shorter the route taken

the greater increase in probability given to its table entry in the pheromone table. This happens repeatedly until the weight of the fastest node is shifted such that slower routes have a very low probability of being chosen.

IV. SIMULATION RESULTS

The following tests will illustrate how the ANTNET algorithm effects the routing of traffic. These tests will show effectiveness of the algorithm against the system running without ANTNET. Since it is possible to switch nodes on and off, a number of test comparisons will be done to show how ANTNET can improve the routing of a network when paths are no longer valid and new routes have to be chosen.

These tests have been run with the following parameters

- ANTNet On
- Simulation Speed –1,000 tick p/s
- Total calls to make – 5000
- Maximum concurrent calls – 60
- Node capacity – 35
- Call duration – 170 The length (in ticks) of a call
- Reduce I/O – bypasses the network visualization to increase simulation speed
- Return on connection – returns the node immediately to source after connection

ANTNet VS non- ANTNet:

The first test contains two simulations.

- Simulation 1 (Orange) – ANTNet algorithm OFF
- Simulation 2 (Blue) – ANTNet ON

From this simulation it is clear that even by the first 500 calls completed, ANTNet has reduced the average number of hops by approximately 1.5 nodes. This is made more apparent by the end of the simulation where the best paths are made more biased as a choice and are re-enforced as the optimal route, resulting in ANTnet improving network performance by almost 3.5 hops. To view the algorithm from a different perspective the following graph depicts the system running with the ANTNet algorithm off and then activated on the 2,000 th call. This can be identified by a label and follows with a decline of average hops by almost 2.

Loop elimination:

Before an Ant returns back to its source node, an optimization technique of loop elimination can be invoked. The problem with loops is that they can receive several times the amount of pheromone than they should, leading to the problem of self re-enforcing loops. From this test, loop elimination has reduced the average number of hops by 1 node with a much more stable adaptation. This would mean that when alternative paths must be chosen, the loop elimination algorithm responds much faster that the regular implementation.

Adaptivity:

It is important to simulate how the network adapts when nodes are removed from the network. Static routing tables may hold the shortest path but they don't necessarily take into account network traffic and nodes that are offline. Three simulations have been run on the program to display how the system adapts compared to a non adaptive algorithm.

V. CONCLUSION

With the prevalence of dynamism in the web and network domains, Our approach have become a suitable option offering the advantages of low cost, increased availability, fault tolerance and flexibility. Achieving judicious work load balance in a network is a challenging task, as the network exhibits extreme dynamism in structure and load. Therefore an innovative

approach is required to facilitate a rational distribution of load among all the available network resources thus enabling full utilization of the accessible resources. Load balancing in networks has to be performed autonomously. An appropriate load balancing mechanism reaps the benefits of improved productivity, simplified management of large scale systems, and enhances the scale-out options.

REFERENCES

[1] Appleby, S., & Steward, S. (1994). Mobile software agents for control in telecommunications Networks. In BT Technology Journal, Vol. 12, No.2.

[2] AntNet: A Mobile Agents Approach to Adaptive Routing Gianni Di Caro and Marco Dorigo

[3] Ant-based load balancing in telecommunications networks Ruud Schoonderwoerd1,2, Owen Holland3, Janet Bruten1 and Leon Rothkrantz

[4] AntNet: A Mobile Agents Approach to Adaptive Routing Gianni Di Caro and Marco Dorigo

[5] Data Networks Bertsekas, D., & Gallager, R. (1992).