Using Ants to Load Balance Telecom Networks

Ruud Schoonderwoerd, Owen Holland and Janet Brute

Presentation by Luke Ferderer and Ryan Bowering

What is a Telecom Network

- A graph
  - Node: Switching Station which calls are routed through
    - Each node has a different, limited capacity
  - Edge: Physical connections between switching stations
- Each node has a routing table
  - Look up table of next node in path to a destination

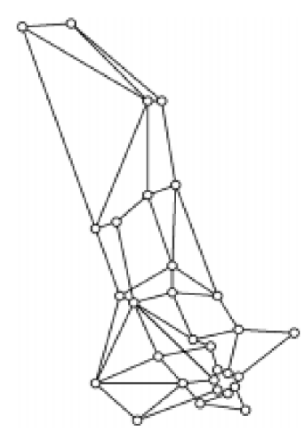


Figure 1: An example telecom network [1]

Table for node E	Next Node
To A	D
To B	B
To C	G
…	…

Figure 2: An example routing table

Load Balancing on a Telecom Network

- Calls are routed between switching stations to reach the destination
- When a call is routed to a full switching station, the call is dropped
- Static routing: precalculate optimal paths and don't update them
  - Doesn't adapt to changing conditions
- Dynamic routing: always calculate new paths based on current network conditions

Goal: Minimize dropped calls by distributing traffic across the network

Dynamic Routing with Mobile Agents

- Previous state of the art
- Mobile agents allow the network to dynamically adapt to changing loads
- 2 types of agents
  - Load balancing agents: Updates routing tables
  - Parent agents: Spawns load agents at congested locations
- Best path using Dijkstra's algorithm
  - Best path maximizes spare capacity of nodes in the path
- Robust through 3 principles
  - No direct inter-agent communication
  - Agents present in reasonably large numbers
  - Agents dynamically alter their task allocations and number

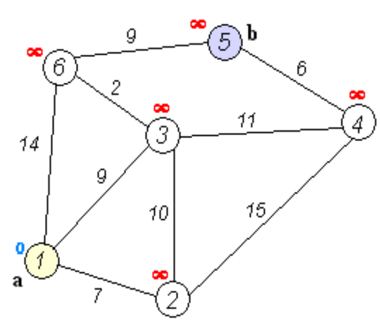


Figure 3: An animation of Dijkstra's algorithm [2]

Problems with Mobile Agents

- Circular routes
  - Load agents update every node along a route
  - Two agents from different source nodes might both update routes to the same node at the same time
  - The agents may have different data leading to different concurrent edits
- Very long routes
  - Maximizing spare capacity can lead to excessively long routes
  - These long routes take up extra network resources that cause congestion problems later

Ant-Like Mobile Agents

- Ants travel randomly throughout the network from a source node
- They record their age as they travel
- Younger ants took faster routes to a destination

Letting Ants Influence the Route

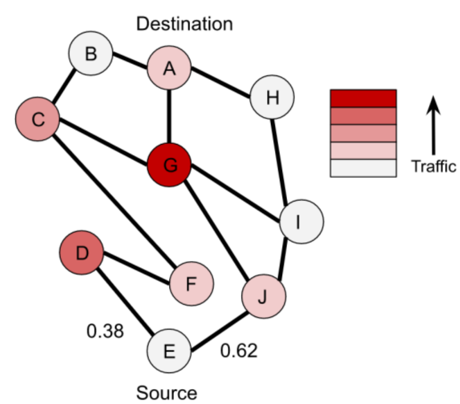
- Ants don't know how long it will take to reach a destination
  - Could retrace its path and alter it on the way back
  - Would require memory of the path and the time to traverse it
- Paths are influenced in reverse
  - It's cheap to store just the past node an ant visited
  - By reversing the source and destination, the ant knows how long the path is

At any node, the age of the ant represents the length of any path to the source from the node through the previous link

Finding the best route

- Each node stores the sum of all path lengths to a target node through each of its neighbors
  - Lower time is weighted toward higher sums
- Each path's sum is treated as the probability of choosing the path

Table for node E	Next Node	Probability
To A	D	0.38
To A	J	0.62
To B	…	…



Manually adding randomness

- Real ants can always stray from a heavily marked trail
  - This happens seemingly at random
- This behavior can be desirable
  - Ensures the whole environment is explored
- We can manually add a noise factor (f)
  - The ant will follow the route table with probability of 1-f
  - The ant will follow a random route with probability of f

Actually Influencing the Route Table

- Taking a max function is a simple way to determine the best path
  - Since we want to favor lower time, we take its reciprocal

$$\Delta p = \frac{0.08}{age} + 0.005$$

- We can add Δp to the sum to influence the path

$$p = p_{old} + \Delta p$$

- To use this as a probability, it must be normalised
  - The new entry becomes:

$$p = (p_{old} + \Delta p)/(1 + \Delta p)$$

  - The other entries become:

$$p = p_{old}/(1 + \Delta p)$$

Accounting for Congestion

- Can add an increasing function of the congestion of a node to the age
  - Very simple
- Can delay the ant for some increasing function of the congestion of a node
  - Will also increase the age
  - Will also delay the ant from updating future routes and increasing traffic
- Delay function is defined as:

$$delay = \left\lfloor 80 \cdot e^{-0.075 \cdot s} \right\rfloor$$

Quick Example

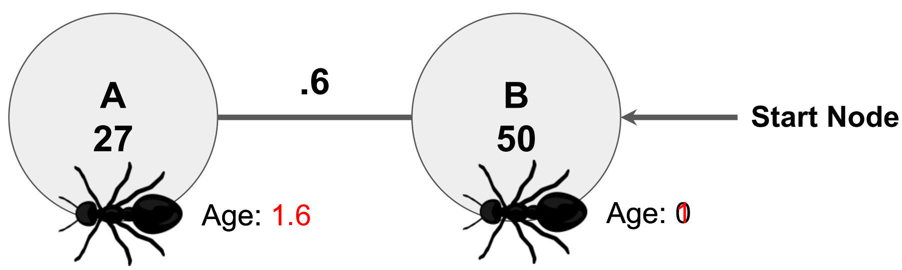


Table for node B	Next Node	Probability
To A	B	0.5

$$\Delta p = \frac{0.08}{1.6} + 0.005 = 0.055$$

$$p = \frac{0.4725+0.055}{1+0.055} = 0.5 \rfloor = 1$$

Worksheet Time

Equations:

$$delay = \left\lfloor 80 \cdot e^{-0.075 \cdot s} \right\rfloor$$

$$\Delta p = \frac{0.08}{age} + 0.005$$

$$p = (p_{old} + \Delta p)/(1 + \Delta p)$$

$$p = p_{old}/(1 + \Delta p)$$

Worksheet Solution

Routing Tables:

Node A	Next Node	Probabilities
To B	B	1.00
To B	C	0.00
To C	B	0.00
To C	C	1.00
To E	B	0.60 | 0.602 | 0.597
To E	C	0.40 | 0.398 | 0.403

Node B	Next Node	Probability
To A	A	1.00
To A	E	0.00
To C	A	0.67
To C	E	0.33
To E	A	0.10 | 0.098 |
To E	E	0.90 | 0.902 |

Node C	Next Node	Probability
To A	A	1.00
To A	E	0.00
To B	A	0.70
To B	E	0.30
To E	A	0.05 | | 0.049
To E	E	0.95 | | 0.951

Node E	Next Node	Probability
To A	B	0.35
To A	C	0.65
To B	B	1.00
To B	C	0.00
To C	B	0.00
To C	C	1.00

Now, route a call from A to E.
Choose path of highest probability at each node: A, B, E

Experimental Setup

Call Distributions

- Each node is assigned between .01 and .07 probability of being a source or destination in a call
  - Probabilities are normalised to all sum up to 1
- A random call sequence is generated over 15,000 time steps
  - An average of 1 call per time step is generated (Poisson distribution)
  - Calls last an average of 170 time steps (Exponential distribution)

The performance indicator is the proportion of calls that could not be placed on the network in a given interval

Results: Static Call Distributions

	Mean	Standard dev.
Fixed, shortest routes)	12.57%	2.16%
Original mobile agents	9.19%	0.78%
Improved mobile agents	4.22%	0.77%
Ants (0% noise)	1.79%	0.54%
Ants (5% noise)	1.99%	0.54%

TABLE 1. Results for unchanged call distributions

	Mean	Standard dev.
No improved mobile agents after 7500	6.43%	2.17%
No ants (0% noise) after 7500	2.11%	0.60%
No ants (5% noise) after 7500	2.48%	0.69%

TABLE 2. Results for unchanged call distributions, load balancing stopped

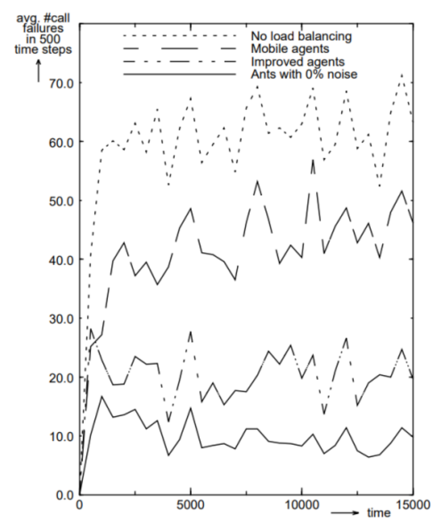


FIGURE 3. Temporal course of call failure rates for four load balancing techniques, with unchanging call probabilities.

Results: Dynamic Call Distributions

	Mean	Standard dev.
Fixed, shortest routes	12.53%	2.04%
Original mobile agents	9.24%	0.80%
Improved mobile agents	4.41%	0.85%
Ants (0% noise)	2.72%	1.24%
Ants (5% noise)	2.56%	1.05%

TABLE 3. Results for changed call distributions

	Mean	Standard dev.
No improved mobile agents after 7500	8.03%	2.88%
No ants (0% noise) after 7500	4.29%	2.06%
No ants (5% noise) after 7500	4.37%	2.27%

TABLE 4. Results for changed call distributions, load balancing stopped

Ants outperformed Mobile Agents in all scenarios.
Ants with noise were best with changing call distributions.

Sources

[1] Schoonderwoerd, R., Holland, O., and Bruten, J. 1997. Ant-like agents for load balancing in telecommunications networks. *Proceedings on the first international conference on Autonomous agents, 209-216.*

[2] By Ibmua - Work by uploader., Public Domain, https://commons.wikimedia.org/w/index.php?curid=6282617

Questions/Discussion