Routing in packet switched networks using agents

Roland van der Put

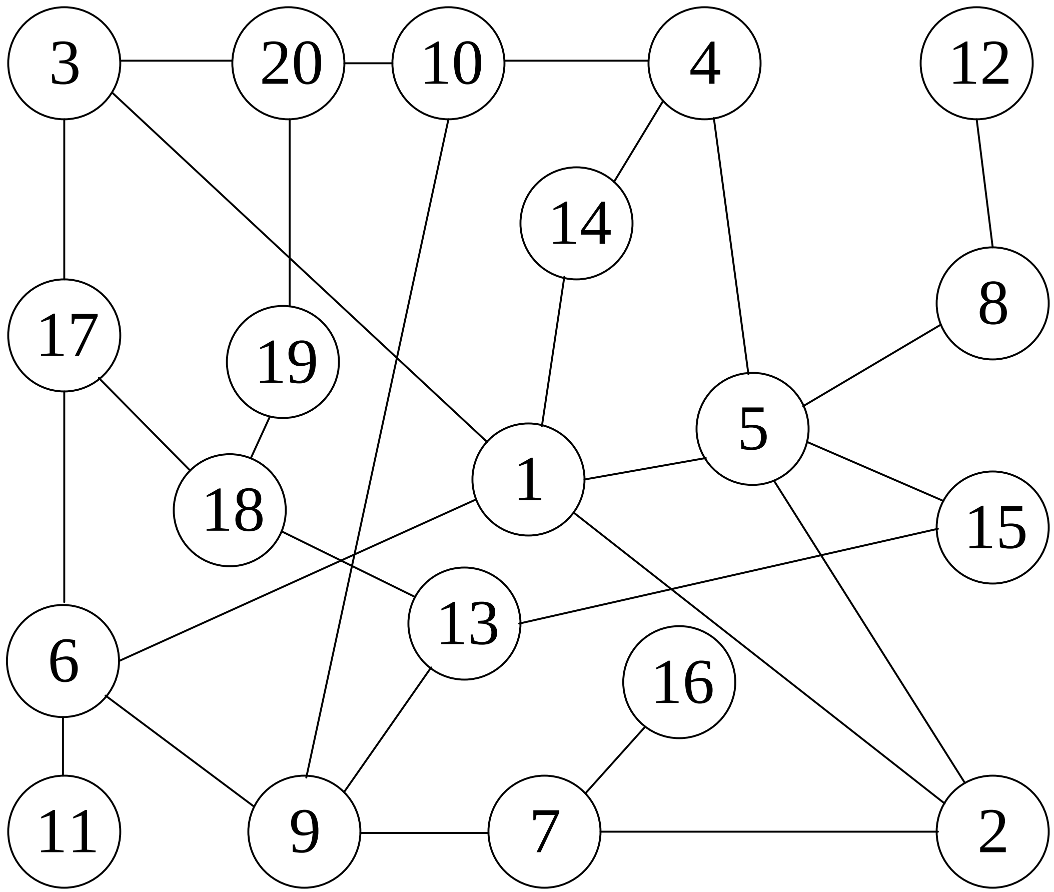
Léon Rothkrantz

TUDelft

Knowledge Based Systems Group

Zuidplantsoen 4
2628 BZ Delft, The Netherlands

L.J.M.Rothkrantz@cs.tudelft.nl


3
20
10
4
12
14
8
17
19
5
1
18
15
13
6
16
11
9
7
2

Circuit Switching Versus Packet Switching Networks

- **Circuit Switching Networks**

  On request a route from source to destination is initialized according to the routing tables in the network switching centers (or nodes) and a fixed number of resources is allocated for some time

- **Packet Switching Networks**

  A message is sent as a packet from one node to another, no fixed connection is made between source and destination

Applications of Artificial Life

- The symmetric and asymmetric travelling salesman problems.
- The sequential ordering problems.
- Quadratic assignment problems.
- Vehicle routing problems.
- Scheduling problems.
- Graph colouring problems.
- Partitioning problems.
- Real functions optimisation problems

Adaptive and Non-adaptive Routing

- **Adaptive Routing**

  The routes adapt automatically to changes in network load and network failures

- **Non-adaptive Routing**

  Routing decisions are made off-line and downloaded to the switching nodes of the network

Centralised and Decentralised Routing

- **Centralised Routing**

  There is a central controller in the network. The network nodes send periodically information about their states to the controller. According to this information the controller calculates better routes from every node to every other node in the network and sends the resulting routing tables to the nodes.

- **Decentralised Routing**

  Instead of one very complex program there may consist simple interacting entities.

The Network Model

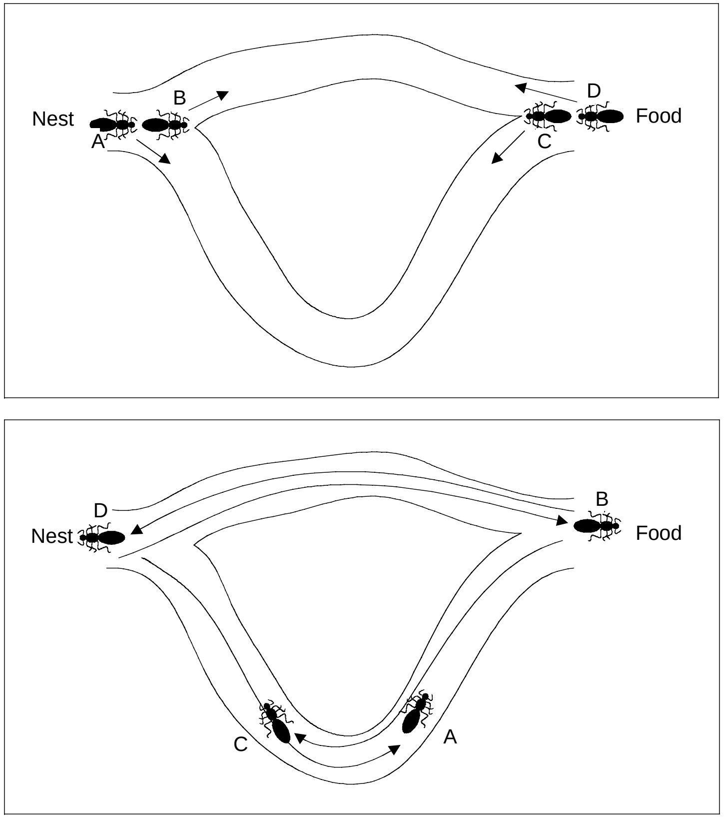
- The network model used is a graph of n nodes, each of which has several attributes:
  - a node identifier
  - a capacity; this is the number of simultaneous calls that the node can handle
  - a routing table with (n -1) entries, one for each node in the network. Each entry tells us which is the next node on the route to the destination node concerned
  - a probability of being the end node (either source or destination) of a call
  - a spare capacity. This is the percentage of the capacity that is still available on the node

Local Control Versus Global Control

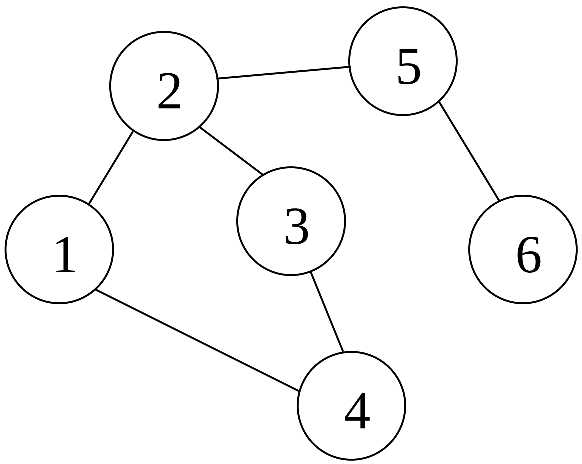
- Local knowledge and behaviour on micro level results in behaviour (and knowledge ?) on meso-, macro level
- Global supervision and local behaviour

Finding the Shortest Route

- Trail laying by ants
- Sensing pheromones
- Probability of choice of route depends on concentration of pheromones and on random choice
- Local interaction of ants results in higher level behaviour



Pheromone Trails and Probability Tables



Node 1	2	4
2	0.95	0.05
3	0.67	0.33
4	0.05	0.95
5	0.83	0.17
6	0.73	0.27

At every node a packet is directed to the route with the highest probability

TUDelft

Basic Principles for Ant-based Control Systems

- Ants are regularly launched with random destination on every part of the system
- Ants walks randomly according to probabilities in pheromone tables for their particular destination
- Ants update the probabilities in the pheromone table for the location they were launched from, by increasing the probability of selection of their previous location by subsequent ants
- The increase in these probabilities is a decreasing function of the age of the ant, and of the original probability

Basic Principles for Ant-based Control Systems *(continued)*

- This probability increase could also be a function of penalties or rewards the ant has gathered on its way
- The ants get delayed on parts of the system that are heavily used
- The ants could eventually be penalised or rewarded as a function of local system utilisation
- To avoid overtraining through freezing of pheromone trails, some noise can be added to the behaviour of the ants

Parameters of Ant-based Control Systems

- The speed of the ants
- The times at which ants get launched
- The probabilities are updated with decreasing function of age

$$\Delta p = \frac{d}{age} + c$$

- The delay in time steps is given to the ant is an increasing function of the spare capacity s of the node

$$delay = \left\lfloor a \cdot e^{-b \cdot s} \right\rfloor$$

Circuit and Packet Switched Networks and Algorithm

- Circuit Network
  - Cost function is spare capacity in nodes
  - Backward agent algorithm is implemented
- Packet Switched Network
  - Cost function is the trip time
  - Adapted backward agent algorithm is implemented
  - Forward/backward agent algorithm is implemented

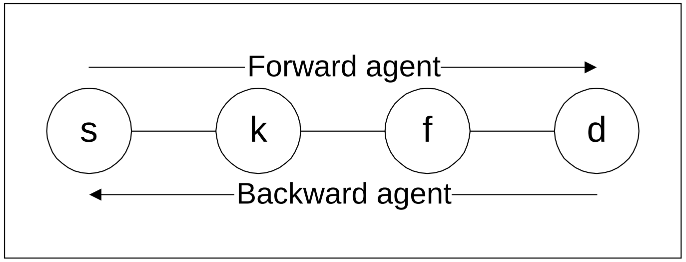
Forward and Backward Algorithm

- At regular intervals, from every network node s, a forward agent is launched with a random destination d:F_{sd}. This agent has a memory that is updated with a new information at every node k that it visits. The identifier k of the visited node and the time elapsed since its launch time to arrive at this node k is added to the memory. This results in a list of (k, t_k)-pairs in the memory of the agent.
- Each traveling agent selects the link to the next node using the probabilities in the probability table. Next nodes that have already been visited by this agent are filtered out in the agents copy and the new, temporary probability distribution is normalised to 1. Only this agent uses this new, temporary probability distribution. The probability table in this node is updated.
- If an agent is forced to return in an already visited node, the agent backtracks to the previous node. The duration of this cycle is subtracted from the trip times calculated after this cycle

Forward and Backward Algorithm *(continued)*

- When the destination node d is reached, the agent F_{sd} generates a backward agent: B_{ds}. The forward agent transfers all its memory to the backward agent and then it will destroy itself. During the transfer of the memory, all cycles in the agent's path are filtered out.
- The backward agent travels from the destination node d to the source node s along the same path as the forward agent, but in the opposite direction. It uses its memory to know the path, so it does not use the probability table.
- The backward agent (with previous node f) updates the probability table in the current node k. The probability p_{df} associated with node f and destination node is d is incremented and the probabilities p_{dx} associated with other next nodes x for the same destination node d are decremented.

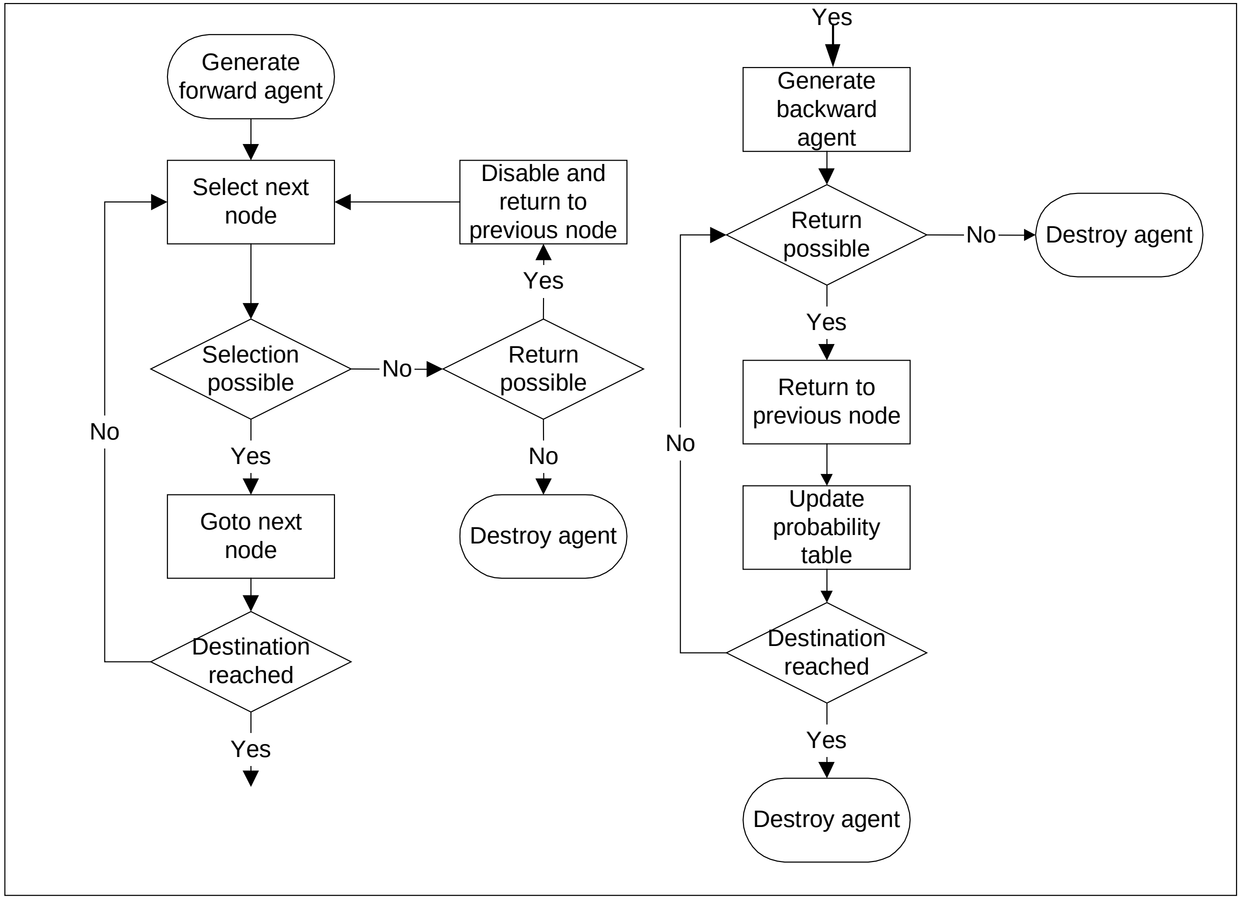
Path of Forward and Backward Agents



- Choice of route of packets depends on probability tables
- Agents are launched at each node at every time step with a random destination node
- Forward agents collect the data
- Backward agents update corresponding probability tables in the associated direction
- The algorithm uses the trip times as a cost function instead of spare capacity in nodes

TUDelft

Flow Chart of the Agent Algorithm



TUDelft

Updating the Probability Tabels

$$P_{new,\,f} = \frac{P_{old,\,f} + \Delta P}{1 + \Delta P}$$

$$P_{new,\,i} = \frac{P_{old,\,i}}{1 + \Delta P}, 1 \le i \le n, i \ne f$$

$$\Delta P = \frac{a}{t} + b$$

$P_{new,f}$ - the new probability

$P_{old,f}$ - the old probability

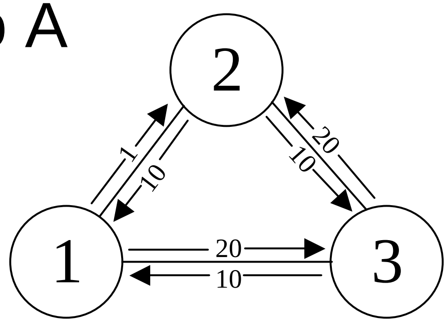
ΔP - the (probability) increase

Shortest paths

- The agents will bias towards the shortest (least expensive) paths because of the following three reasons:
  - shorter routes will be completed earlier than longer routes and thus attract other agents to their source nodes first
  - shorter routes involve fewer branches, so the number of agents will be larger than on longer routes with more branches
  - agents traveling shorter routes will be younger when they arrive. They will therefore influence the probability tables more than the older agents that traveled a longer route do

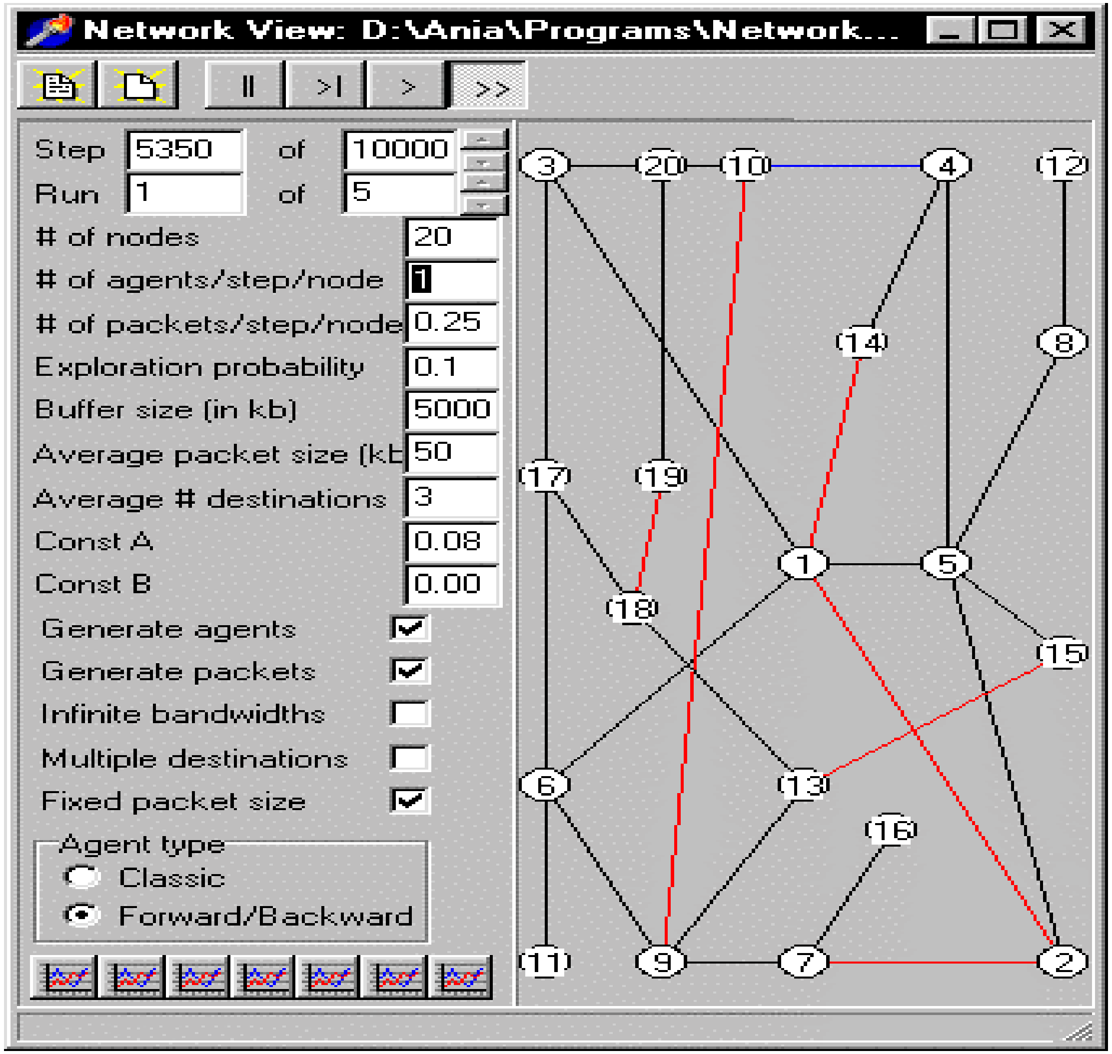
Asymmetrical networks

- A network is called asymmetrical when one or more links have a different capacity from A to B than from B to A



- With the forward/backward agent algorithm, the average age of agents remains about constant.
- With the classical agent algorithm there is an increasing average age of agents

Workbench and Simulation of Ant Behaviour

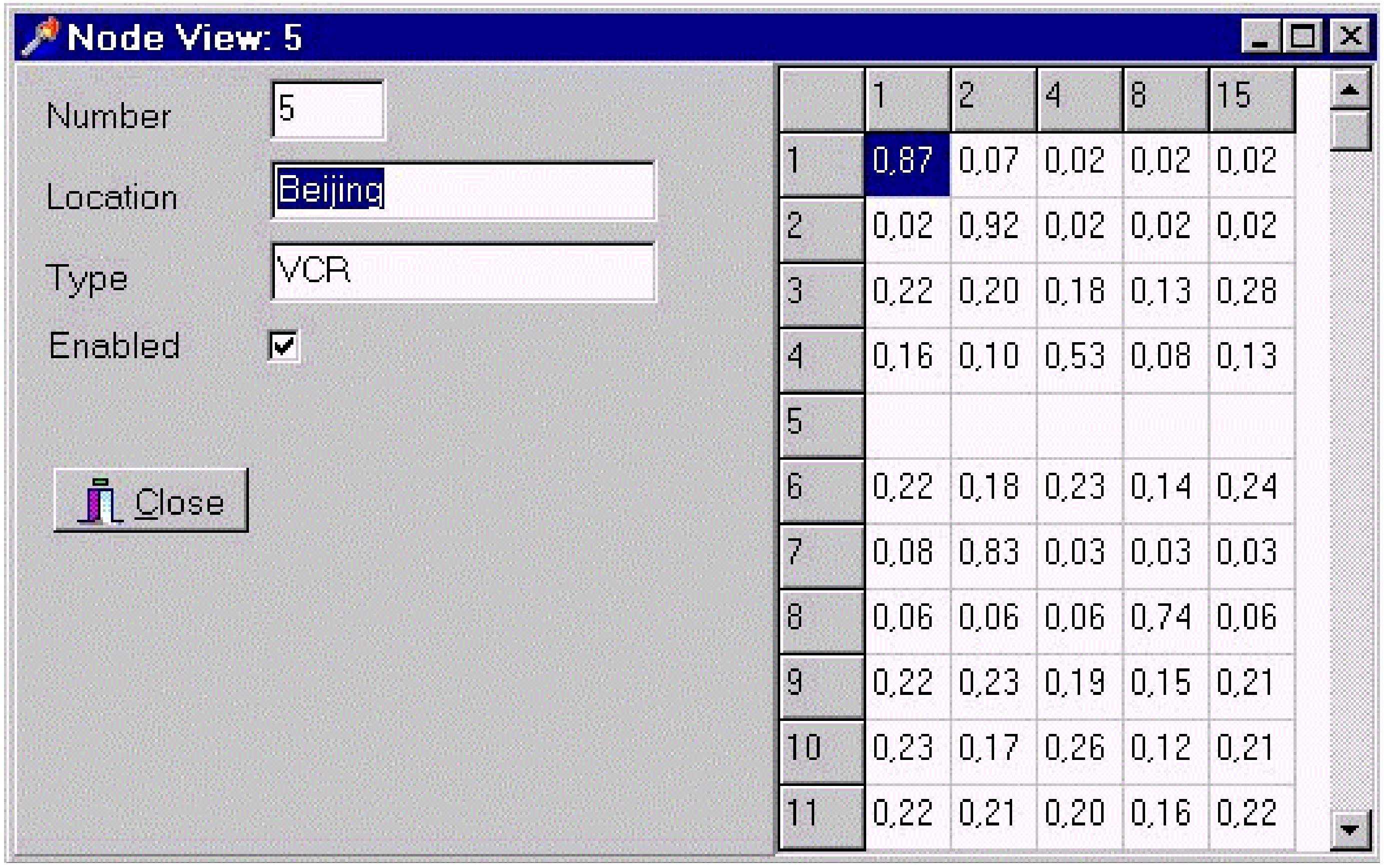


TUDelft

Network Topology

- The nodes know about each other by means of algorithm that broadcasts information about a node to its neighbouring nodes. The nodes receiving this information broadcast it to their neighbouring nodes and so on. A node does not broadcast information if it receives the information for the second time to prevent duplicates. After a few steps with this algorithm, all nodes in the network know about each other and the algorithm terminates.
- Each time a node is added or removed from network, a new broadcast is started. This broadcast sends the information about the addition or removal of a node to all other nodes in the network. This allows for dynamic changes in the network topology.

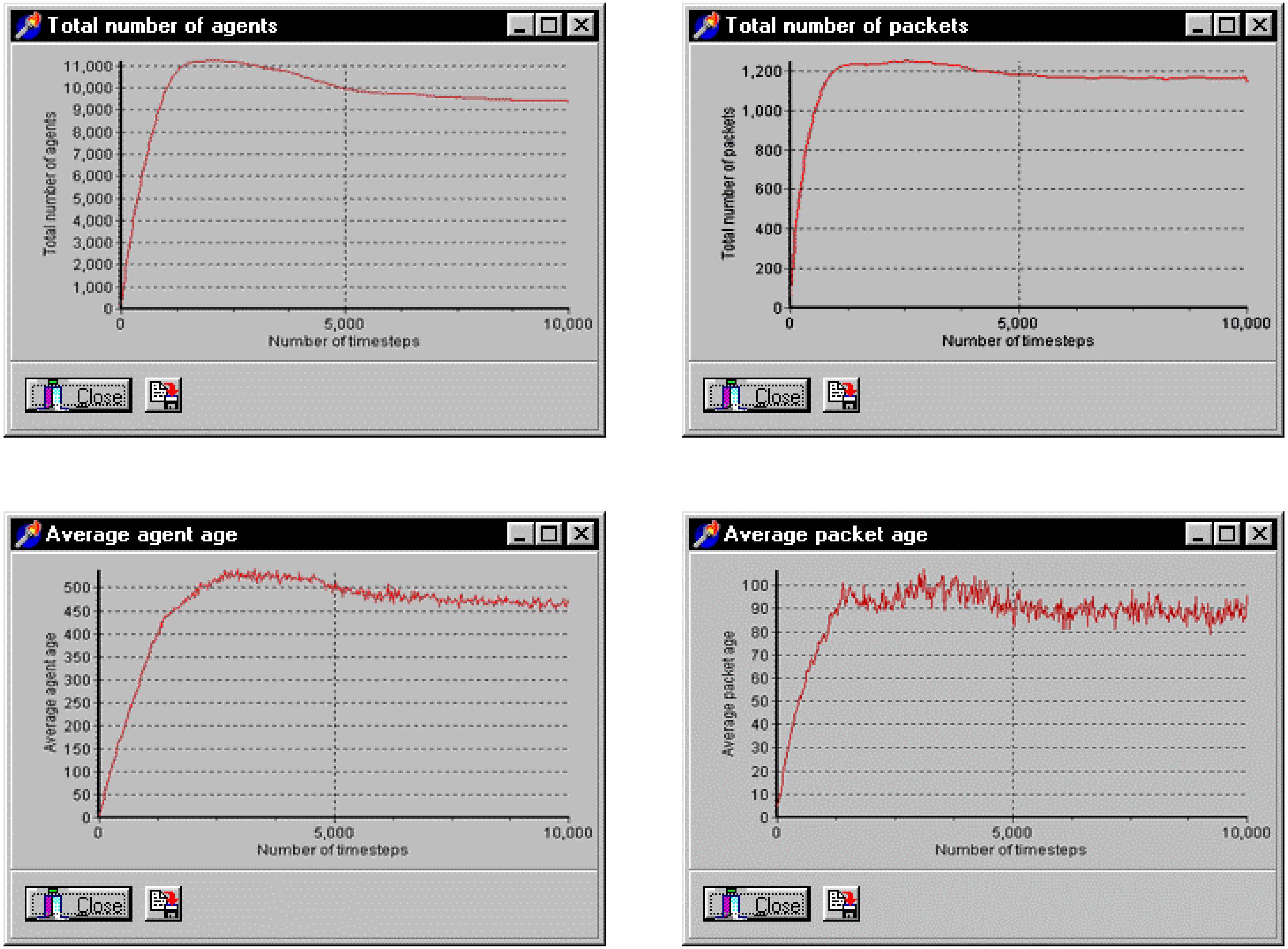
Detailed Node Information

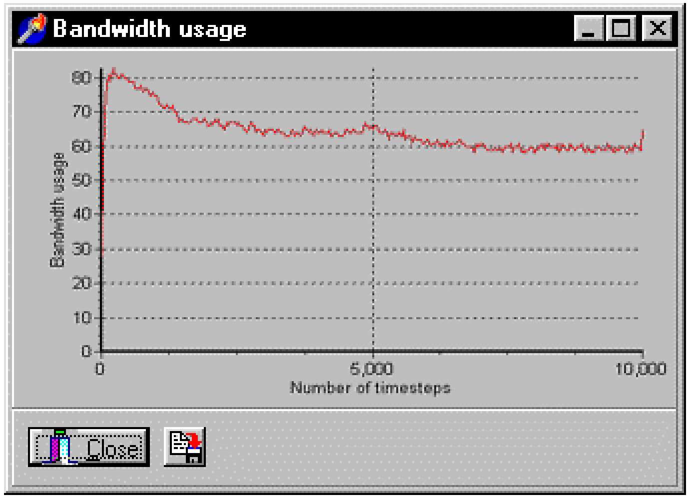


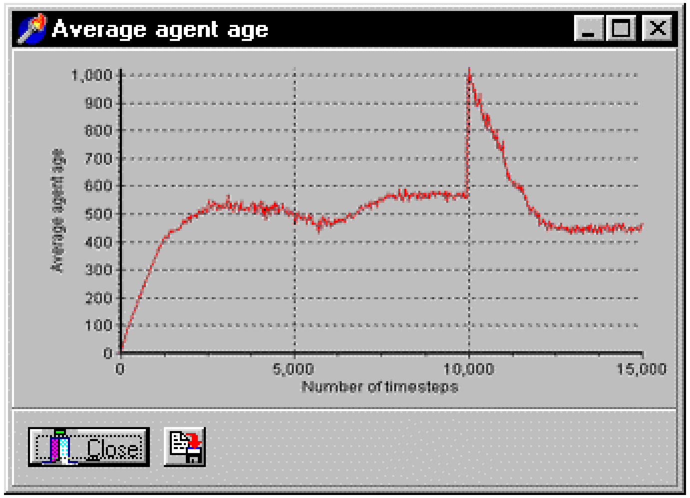
	1	2	4	8	15
1	0,87	0,07	0,02	0,02	0,02
2	0,02	0,92	0,02	0,02	0,02
3	0,22	0,20	0,18	0,13	0,28
4	0,16	0,10	0,53	0,08	0,13
5					
6	0,22	0,18	0,23	0,14	0,24
7	0,08	0,83	0,03	0,03	0,03
8	0,06	0,06	0,06	0,74	0,06
9	0,22	0,23	0,19	0,15	0,21
10	0,23	0,17	0,26	0,12	0,21
11	0,22	0,21	0,20	0,16	0,22

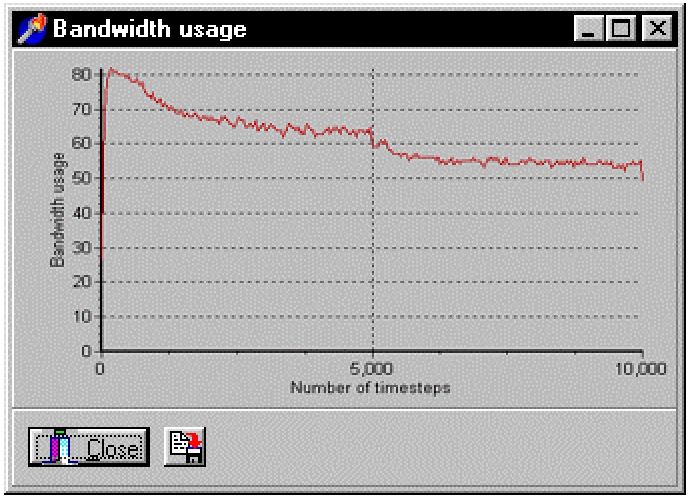
Parameters and Statistics

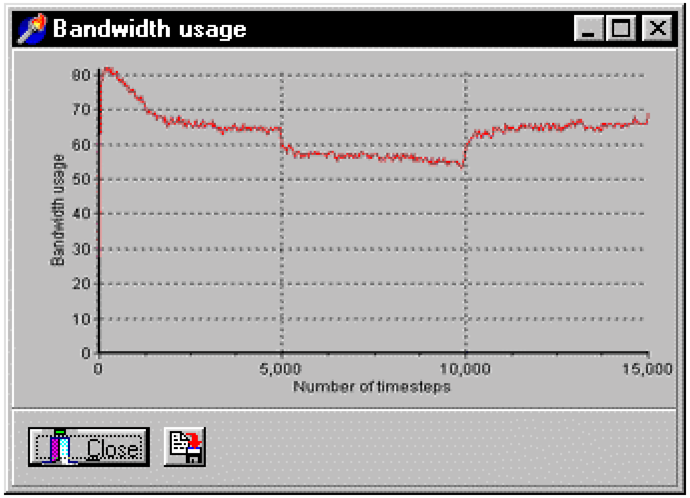
- Number of agents in the network.
- Number of packets in the network.
- Average age of agents.
- Average age of packets.
- Network throughput.
- Number of arrived packets.
- Number of packets lost.


Total number of agents
11,000
10,000
9,000
8,000
7,000
6,000
5,000
4,000
3,000
2,000
1,000
0
Total number of agents
0
5,000
10,000
Number of timesteps
Close
Total number of packets
1,200
1,000
800
600
400
200
0
Total number of packets
0
5,000
10,000
Number of timesteps
Close
Average agent age
500
450
400
350
300
250
200
150
100
50
0
Average agent age
0
5,000
10,000
Number of timesteps
Close
Average packet age
100
90
80
70
60
50
40
30
20
10
0
Average packet age
0
5,000
10,000
Number of timesteps
Close


Bandwidth usage
Bandwidth usage
Number of timesteps
Close


Average agent age
Average agent age
Number of timesteps
Close


Bandwidth usage
Bandwidth usage
Number of timesteps
Close


Bandwidth usage
Bandwidth usage
Number of timesteps
Close

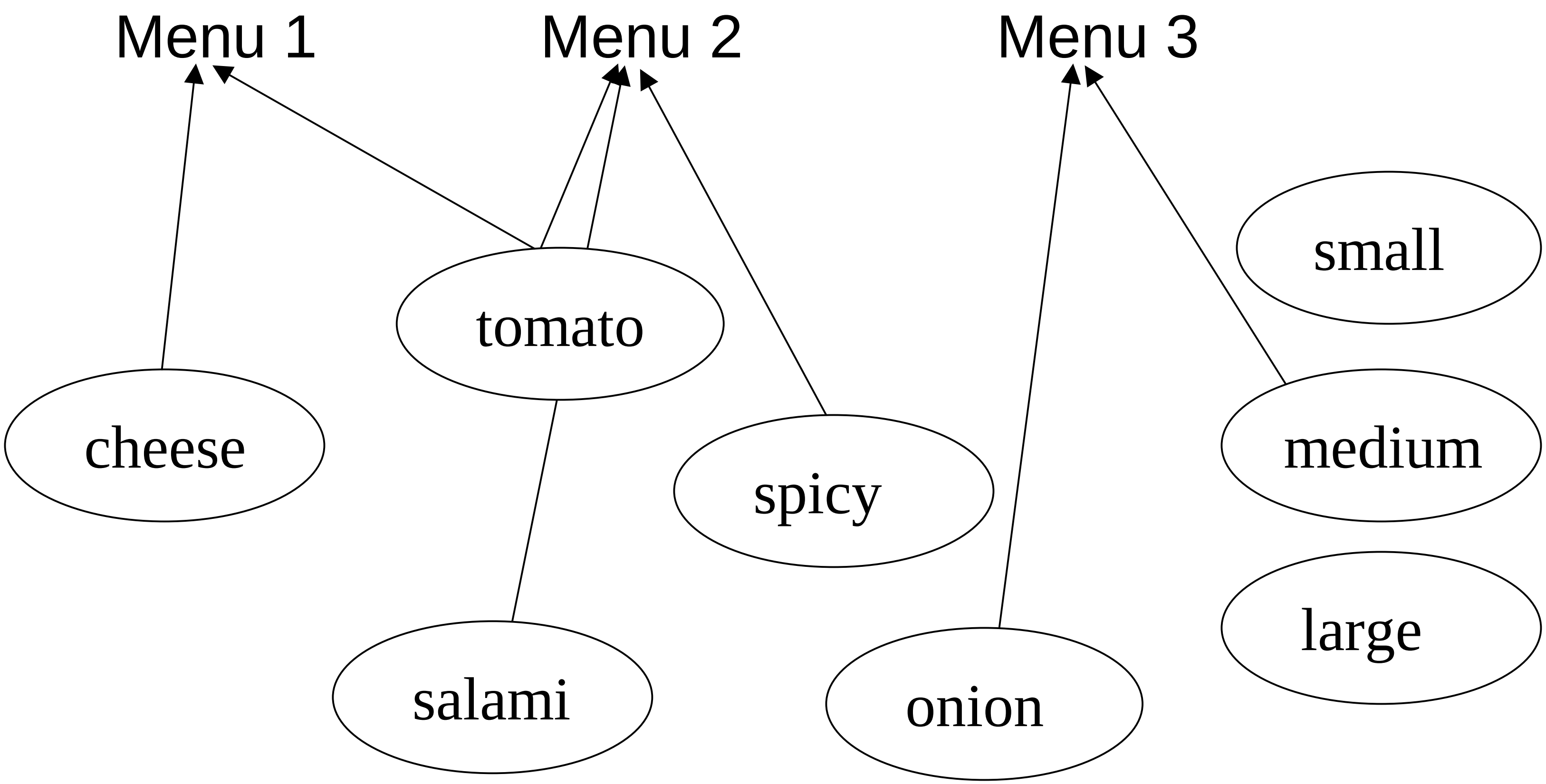
Conclusions

- The forward/backward agent algorithm proved to be very robust and efficient algorithm for routing in packet switched network
- The forward/backward agent algorithm performed better then the backward agent algorithm
- The forward/backward agent algorithm is capable of handling asymmetrical networks
- The forward/backward agent algorithm respond quickly to changes in the topology of the network

Future Work

- Improvements of the forward/backward agent algorithm
  - the agent will have a penalty related to the amount of buffer space in a visited node
  - the packets choose the nodes with the highest or the second highest probability
  - to balance the load more equally the algorithm could apply a threshold probability and choose one of the next node with a probability above this threshold

Semantic Network; Directed Graph



TUDelft

Frames

Menu 1
cheese
tomato
small
medium
large

Menu 2
tomato
spicy
small
medium
large

Menu 3
cheese 1
cheese 2
small
medium
large