A Multi-agent System for Dynamic Network Reconfiguration

Presented by: Hoi Fung Lam

Based on the paper: Artur Maj, Jaroslaw Jurowicz, Jaroslaw Kożlak, Krzyszrof Cetnarowicz, A Multi-agent System for Dynamic Network Reconfiguration, CEEMAS 2003, LNAI 2691

Introduction

Problem: Increased traffic and more bandwidth-demanding applications converge at boundary routers, resulting in service degradation, caused by the router bottleneck.

Solution:

1. Buying more bandwidth.
2. Manage the resources already possessed effectively — dynamic network reconfiguration, which includes:
    - load optimization for making use of full potential;
    - automation of configuration and administration processes;
    - greater scalability;
    - autonomous reaction in case of failure or congestion.

Overview

Network management sytems (NMS)

- centralized — one management interface to the whole network
- weakly distributed hierarchical — agents installed on managed network device use a predefined command set
- strongly distributed hierarchical — decisions are made by an active and partly autonomous agent residing on a network device, which reduces the load on the management station
- cooperative — using intelligent agents, the agents' task is to choose and execute proper actions which will lead to the solution

Market-based methods

- resources are distributed
- the most profitable configuration is chosen

Existing Dynamic NMS

Algorithm

- agents circulating in the network
- gathering information about existing network
- entering the information in routing tables on the nodes

Mesaurement of algorithm efficiency

- percentage of nodes having valid route entries vs. the one of the hosts designed

Influencing parameter

- the next node's choice

Pricing, negotiation, auction, etc.

The Model — Network Architecture

Network hierarchy The network consist of hosts and lines, hosts are grouped in subnets, each subnet has one host appointed as a gateway to the other subnet

Hosts Data sources, client computers, routers

Ports and lines Hosts communicate by using lines, anchored in ports, each port has two buffers: IN and OUT, outgoing lines have parameters: capacity, price, length, maintenance cost and guaranteed capacity

Routing table Includes entries about adjacent hosts and some other hosts in the network

Destination address	Port	Time	Cost	Distance	Frequency

Packet Parameters: destination, money, history of a covered way and TTL

Network archiecture

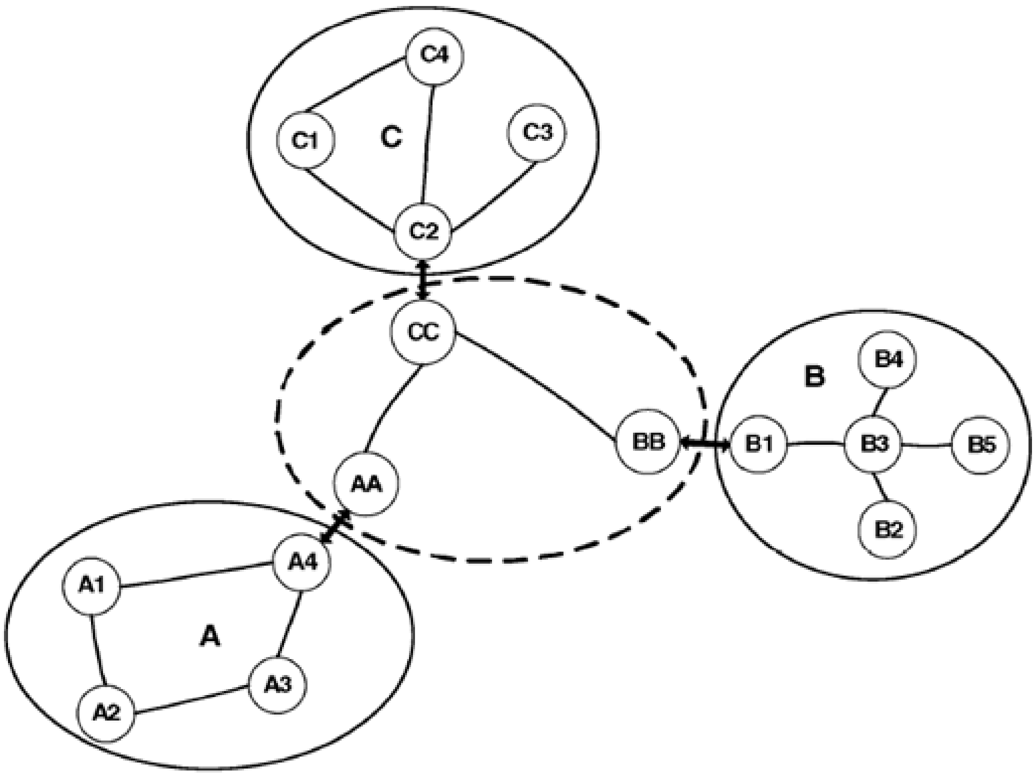

Figure 1: Network archiecture

The Model — Static Agents

Main agent Creating and updating the routing table, financial control, building new and deleting unprofitable lines, run once per simulation tour

Statistical agent Retrieve short and long term statistics of traffic, reside in each port and each host

Pricing agent Each port has its own resident pricing agent, which manupulates the price of the outgoing line

The Model — Mobile Agents

Seeking agent Find routes to a specific host in the network

Checking agent Verify the routing table entries, moves through the network accroding to the routing table and report any changes

Tracing agent Observe routes on which packets move, this agent is sent to the specific host

Marketing agent Seeks for nodes to which a new connection would be worth building

The agents work on behalf of their parent hosts, no direct information exhange, only modify routing and statistic tables of the parent hosts.

Network Simulator Algorithm

The simulation was done in iterations, in each round there are 3 phases:

1. **Packet processing**
   (a) New packets generated
   (b) Host takes incoming packets and send them
   (c) Main agents running, generates new mobile agents
2. **Mobile agent processing**
   (a) In each host, mobile agents move to the host's internal buffer
   (b) Each agent is activated
   (c) The agents were sent to chosen port or stay
   (d) Removal of agents which have finished their mission
3. **Packets and agents moves**
   (a) Each port sends the packets (w/ capacity limitation)
   (b) Sends all the agents to the desired port (w/o capacity limitation)

Experiments

Area of interest

- Network traffic management
- Network structure rebuilding

Experiment setup

- Network A: one-piece local subnet consisting of 32 hosts
- Network B: consist of 3 local networks

Network traffic management

- Assumption: No congestion would occur
- Areas to analyse:
  1. The time required for routing tables to reach the stable state
  2. The level accuracy of tables
- Measurement: Percentage of packets reaching the destination

Network	Table size	Time	Efficiency
A	8	430 rounds	99%
A	14[a]	instantly	99%
B	0	instantly	90%
B	3	instantly	98%
B	8 onwards	instantly	100%

[a] Only few key hosts require this big table size

Network structure rebuilding

- The experiment was carried only on network A
- Setup: bandwidth was 300, guaranteed bandwidth 10, routing table size 20

Money / packet	Time	Result
5000	500 rounds	traffic $<$ bandwidth, 4 new links appear on short and loaded routes
1000	2500 rounds	traffic reduced almost on all connections
6000	4500 rounds	topology almost identical to 5000, 500 rounds
10000	4500 rounds	9 new links were built, even on long distances

- Other experiment setup: bandwidth was 10, completely congested, other remains the same
- Result: number of new links much greater than in the previous experiment

Conclusions

- The system use microeconomics theory heavily to achieve an optimal balance between computational efficiency and flexibility
- Proper modeling of the real telecommunication network
- Possibility to change many parameter of the system configuration
- Huge computational complexity of the simulator
- Lack of graphical interface, necessity of parameter adjusting by hand

Appendix

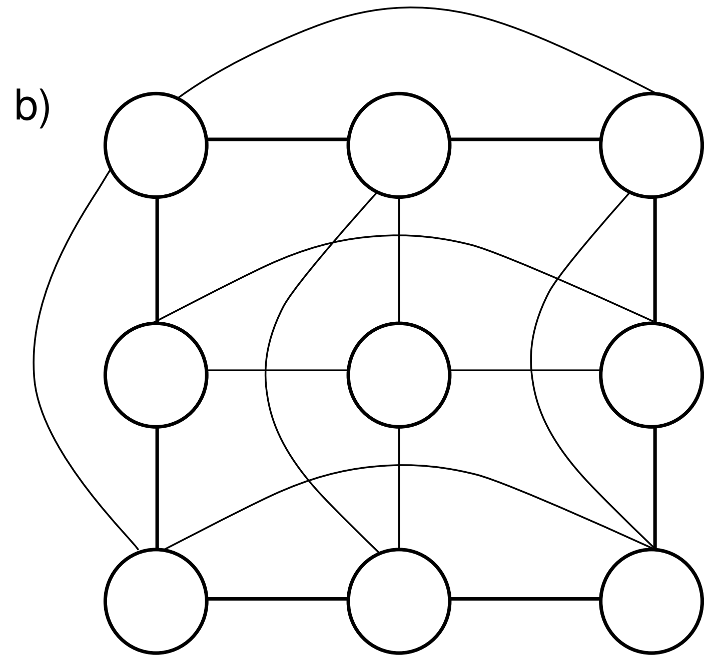

Figure 2: 2D Mesh with wraparound

Appendix

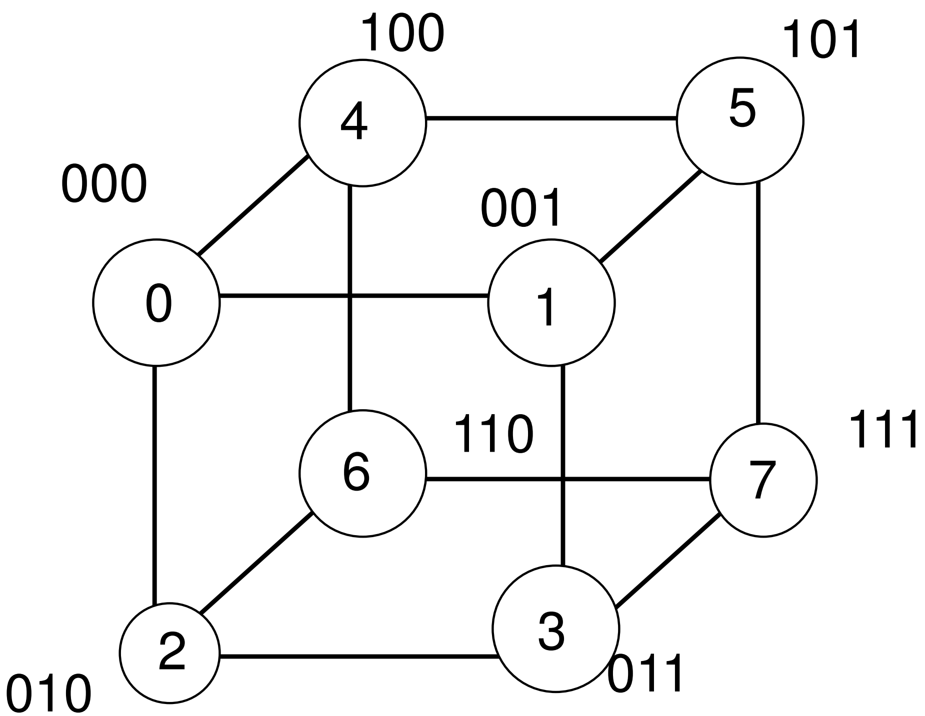

Figure 3: 3D Hypercube