Abstract

This paper deals with the simulation of a multi-agent system based on the Fragmented Mobile Agent Network model. The model consists of agent teams performing remote software management operations and network elements that connect processing nodes and allow agent mobility. A case study considering a scenario in which multi-operation teamwork agents install new software in a network is included. An analysis of simulation results based on operation execution in a simulated large-scale network with different fragment sizes, network sizes and node/link capabilities is elaborated.

Network Simulation in a Fragmented Mobile Agent Network

Mario Kusek, Gordan Jezic, Kresimir Jurasovic and Vjekoslav Sinkovic

{mario.kusek, gordan.jezic, kresimir.jurasovic, vjekoslav.sinkovic}@fer.hr

University of Zagreb, Croatia

Faculty of Electrical Engineering and Computing

Model of a Mobile Agent Network (MAN)

MAN is represented by a triple $\{A, S, N\}$ were:

- A represents a Multi Agent System $A=\{agent_1, agent_2, \dots, agent_k, \dots, agent_{na}\}$
- $agent_k$ is a mobile agent described by $agent_k=\{name_k, adress_k, task_k\}$
- $task_k$ represents agent functionality $task_k=\{s_1, s_2, \dots, s_i, \dots, s_p\}$
- A set of nodes is represented by S $S=\{S_1, S_2, \dots, S_i, \dots, S_m\}$
- Si represents a set of operation that can be executed at the node $S_i=\{S_1, S_2, \dots, S_j, \dots, S_{nsi}\}$
- N represents a network described by $N=\{S, E\}$
- A set of edges e_{ij} connecting nodes S_i and S_j is represented by E

Model of a Fragmented MAN (F-MAN)

F-MAN is represented by a triple $\{SA, FS, N\}$ were:

- SA represents a Multi Agent System organized as a team and divided into subteams $SA=\{SA_1, SA_2, \dots, SA_i, \dots, SA_n\}$
- FS represents a set of network fragments on which subteams perform service $FS=\{f_1, f_2, \dots, f_i, \dots, f_n\}$
- Each fragment f_i includes a set of processing nodes $f_i=\{S_{i1}, S_{i3}, \dots, S_{iq}\}$
- N represents a network described by $N=\{S, E\}$

Subteam organization

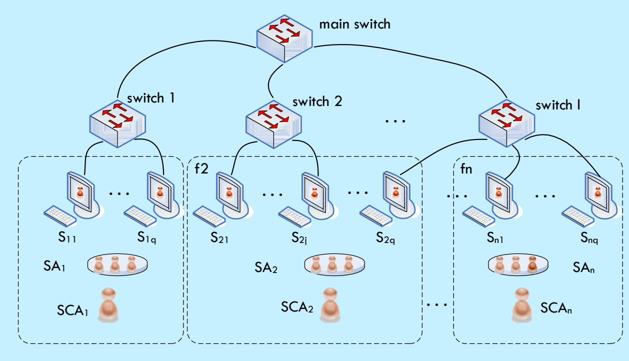
Subteam consists of:

- One subteam coordination agent (SCA)
- Subteam member agents
- Subteam coordination plan (SCP)

Subteams are coordinated by a management agent (MA). MA is responsible for defining fragments and divides initial request into fragment requests. Fragment requests are sent to the SCAs of each fragment. SCAs then coordinate operation execution according to the SCP.

Link entity

Represent a full-duplex physical links which connect nodes and switches in the network. Each link is limited by its network capacity which causes delay when sending data over the link. A link can be divided into two components: a queue (TQ_i) and a service station (P_i). The queue is used to store processing requests that cannot be processed at that particular time since the service station is already processing some other request. In the network model, a processing request is data regarding the agent sent during the process of agent migration or the content of an ACL message. The service station represents an Ethernet card used to send data through the network.

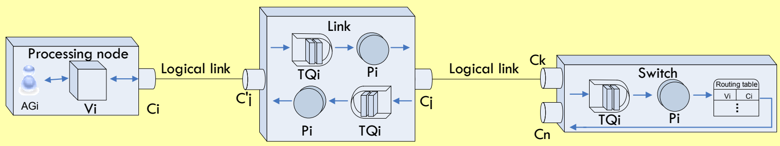


Switch entity

The switch entity represents a network switch used to transfer data between fragments. The switch is composed of three components: a queue, a service station and delivery logic. The queue and the service station are modelled using the same principles as for the link entity. The only difference is that the switch entity's service station has a deterministic service time. The delivery logic component was introduced since a request needs to be sent to the corresponding outgoing connector after processing. It contains a routing table with a list of hosts and the connectors leading to them. The routing table is updated every time data is received from a host not present in the table. The delivery logic is placed after the service station element.

Processing node entity

Processing node (S_i) represents a network node from the F--MAN model. It contains two elements: a network host (V_i) and an agent node (AG_i). The network host offers communication functions to the agent node. The agent node represents the agent platform running on the processing node.



MAN Simulator is a tool used to simulate elements from the F-MAN model. It enables simulation of different coordination and cooperation strategies used by agents while performing operations on remote nodes. In order to improve accuracy of the MAN simulator we introduced several network components: Switch, Link and the Processing node. This was necessary because network introduces delays caused by limited network capacity and the processing times of network elements. In some cases, these delays can have a significant influence on the results. Introduction of network elements also enabled the creation of complex network structures.

RMS overview

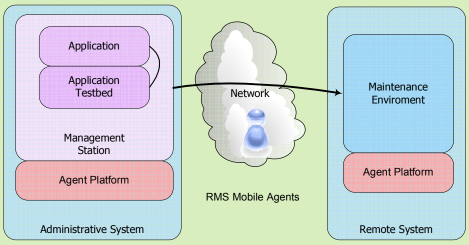
RMS is an agent-based framework, conforming to the MAN model, used to control remote locations. It supports software delivery to remote systems and operations used for remote installation/un-installation, starting/stopping, tracing, maintaining several versions of software, selective or parallel execution of two versions, and version replacement.

RMS architecture

RMS consists of a management station and of remote systems distributed over the network.

The management station is responsible for software delivery to the remote systems and for performing remote operations on them. All tasks are defined by the user thought the management station GUI.

The Maintenance Environment is an application-independent part of RMS, pre-installed on the target remote system(s) in order to enable maintenance actions. The Maintenance Environment is responsible for communication with the management station. Its main tasks include enabling remote operations and storing data about installed software.



In order to validate the results obtained by the simulator we decided to compare results from the simulator with the results from an actual agents system. The agent system which was used is RMS since it conformss with the MAN model. Three distinct analysis were performed. In the first analysis total execution times were compared as the number of switches and fragments increase as well as network bandwidth. Results show that the number of switches does not have much influence on total execution time, increasing network bandwidth significantly decreases the time. High total execution time is a characteristic of scenarios with small and large number of fragments due to parallel execution. In the second analysis compares minimal total execution time as the number of switches and the node delay increases as well as network bandwidth. The analysis show that the only factor that affects minimal total execution time is node delay. The last analysis determinates the optimal number of fragments as the node delay and the number of switches increase. The analysis was performed for two distinct network bandwidths. The analysis show that the optimal number of fragments ranges from 10-15.