Jurnal Teknologi

Full Paper

A THRESHOLD-BASED CONTROLLER FOR MULTI-AGENT SYSTEMS

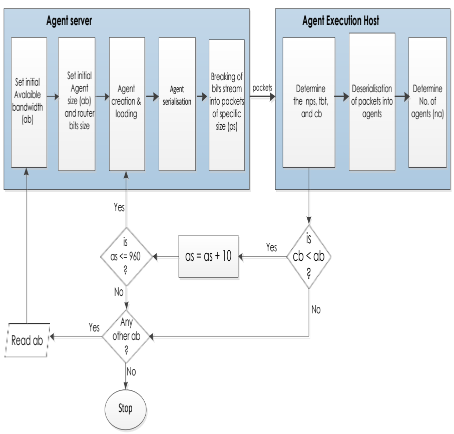
Olumide Simeon Ogunnusi, Shukor Abd Razak*, Abdul Hanan Abdullah

Department of Computer Science, Faculty of Computing, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia

Article history
Received
15 May 2015
Received in revised form
1 July 2015
Accepted
11 August 2015

*Corresponding author
shukorar@utm.my

Graphical abstract



Abstract

Monitoring and regulating the deployment of mobile agents to a network based on its available bandwidth is crucial to forestall the possibility of congestion and consequent network degradation. Our study has shown that only one experimental modelhas addressed the issue. Investigation into this model revealed its failure to honour some basic parameters necessary to yield efficient result. These parameters and network bandwidth determine the maximum deployable number of agents to a network. To achieve the set objective, a threshold-based controller is proposed to regulate the injection of mobile agents into the network relative to the available bandwidth, agent size and router traffic size. The result obtained shows that the proposed model is more accurate and reliable than the existing one.

Keywords: Threshold-based controller, multi-agent systems, serialized agent, network packet

1.0 INTRODUCTION

Multi-Agent Systems (MAS) uses the advantages of artificial intelligence (AI) and distributed computing to device a new way of solving problems [1]. It is a system in which several interacting and intelligent agents perceive and act upon an environment to achieve their own objectives (i.e. the mission of their owners). The basis of these systems arises from several disciplines, such as Sociology, Artificial Intelligence (AI), and Philosophy. The root of a multi-agent system is a special type of software entities called agents.

Olajubu, Ajayi [2] pioneered the research into the controller for regulating injection of agents into network. Up till now, they remain the only researchers ever delve into this research area. Our investigation of the work revealed its complexity of implementation and hence the challenges it posed to the developers of multi-agent system (MAS) based applications. This has motivated our further research into it which is focused at proposing a simple and implementable threshold-based multi-agent system controller with the consideration of all necessary parameters to overcome impending bottleneck. This is achieved with a simple and reusable Threshold-Based Controller (TBC) aimed at running in JADE environment with Java as the underlying programming language. In the proposed TBC, each available bandwidth considered is automatically set as the threshold value for the consumed bandwidth to mitigate network congestion. The rest of the paper is ordered as follows: Section 2 introduces the related work while Section 3 discusses the framework of the proposed controller. In Section 4, we compare the proposed controller and Fuzzy Logic Based Controller (FLC). Section 5 explains the implementation details with comparative analysis of results while Section 6 concludes the paper.

2.0 RELATED WORK

The only existing controller for regulating mobile agent injection into network in multi-agent system based application is discussed in this section. Fuzzy logic based controller (FLC) was proposed by [2] and not until now, no further research has been done in this domain either to improve on the performance of FLC or proposed another method for regulating injection of mobile agents into a network in multi-agent systems.

A number of multi-agent system based applications such as [3], [4], [5], [6], [7], [8], [9], [10], [11] and many others are notconscious of the need to consider the number of deployable agents to the network. The comparative study of the proposed controller with the existing controller models proposed by [2] and few others is depicted in Table 1, although, there are many other multi-agent system based applications which are not presented here. Section 3 discusses the framework of the proposed controller.

Table 1 Comparative study of related works with the proposed threshold-based controller model

Author	Proposed model	Consideration for deployable number of agents	Traffic volume		Function of agents
			Consideration for router traffic	Consideration for agent size	
Aye, Khin [3]	CA-CA interoperability	N	N	N	Agents are used for online users' certificate verification and validation.
Li and Mackaness [4]	Multi-Agent-based, SEmantic-driven (MASE) approach	N	N	N	Agents are used for analyzing epidemiological semantics.
Haghighi and Cliff [5]	Sensomax middleware	N	N	N	Agents are the primary means of communications amongst modules by acting as couriersmoving around the network.
Olajubu, Ajayi [2]	Fuzzy logic controller model	Y	N	N	Generic
Proposed	**Threshold-based controller model**	**Y**	**Y**	**Y**	**Generic**

3.0 FRAMEWORK OF THE PROPOSED THRESHOLD-BASED CONTROLLER

To accomplish the set objective, a threshold-based controller (TBC) is proposed to regulate the number of agents deployable to a network taking cognizance of the available bandwidth and network traffic (comprising agent size and router traffic size). The framework of the proposed TBC is shown in Figure 1.

After setting the agent size, initial available bandwidth, and router traffic size, the server creates, loads and serializes the agent and thereafter uses these parameters to determine the packet size. The packets are then transmitted to the agent execution host (EH), which determines the number of packets that arrive per second, consumed bandwidth, and the total transmitted bits. On arrival, the EH deserializes the packets into agents and determines the number of agents transmitted. The bandwidth consumed by the transmission is compared with the network bandwidth to determine the next action. If the bandwidth consumed is less than the network bandwidth, the agent size is increased by 10 bytes and the new agent size is checked against maximum size set (i.e. 70 bytes). If it is less than 70 bytes, another agent is deployed and the process begins. If the new agent size is greater than 70 bytes, another available bandwidth is considered and the process starts all over again.

The framework considers router traffic size, varying agent sizes, and available bandwidths. However, a number of assumptions are made:

a. Each packet encapsulates an agent.
b. To ascertain the efficient performance of the network, a transmission time of one second is assumed.
c. The router allows varying payload (packet) sizes.
d. All the agents created by the agent server are transmitted to the execution host.
e. The network has a mechanism for network re-distribution.

4.0 THE PROPOSED THRESHOLD-BASED MULTI-AGENTS CONTROLLER

Fig.3 shows the systematic procedure for the attainment of the method to determine the maximum number of agents that could be deployed to a network of specific bandwidth. The controller shows a piecemeal injection of agents into the network until the maximum number is reached. When an agent is created and loaded with all necessary security components, code and agent initialization data, the agent server serializes the agent by transforming it into bits stream. The router breaks the bits stream into packets and transmits them to the agent execution host, where it deserializes back to agent. Meanwhile, the transmitted packets contain the serialized agent and the traffic generated by the router, which are measured in bits. In this research, the router traffic size is taken to be 160 bits (20 bytes) being the minimum possible. This is chosen as a result of the usual small size of an agent. Here, it is assumedthat a transmitted packet encapsulates an agent such that a packet comprises an agent and the router traffic. The packet size and the number of packets transmitted per second are measured with the consumed bandwidth. The agent size is incremented and the process start again until it is above 70 bytes. At that juncture, the whole process starts again with another available bandwidth. For every available bandwidth considered, the controller provides the maximum deployable number of agents. The activities of the proposed controller follows the algorithm presented in Figure 3 with the notations defined in Table 2.

It has been established in [2] that the growth of network size without increase in available bandwidth does not imply increase in the number of agents to be deployed into a network and hence opined that the major factor that determines the number of task agents to be deployed into a network is the available bandwidth. However, this research has gone further to discover that other relevant factor is the network traffic comprising the serialized agent and the router traffic are as important as the available bandwidth in determining the number of agents to be deployed to a network. The concept of routing agents from the internal agent server to the agent execution host is shown in Figure 2.

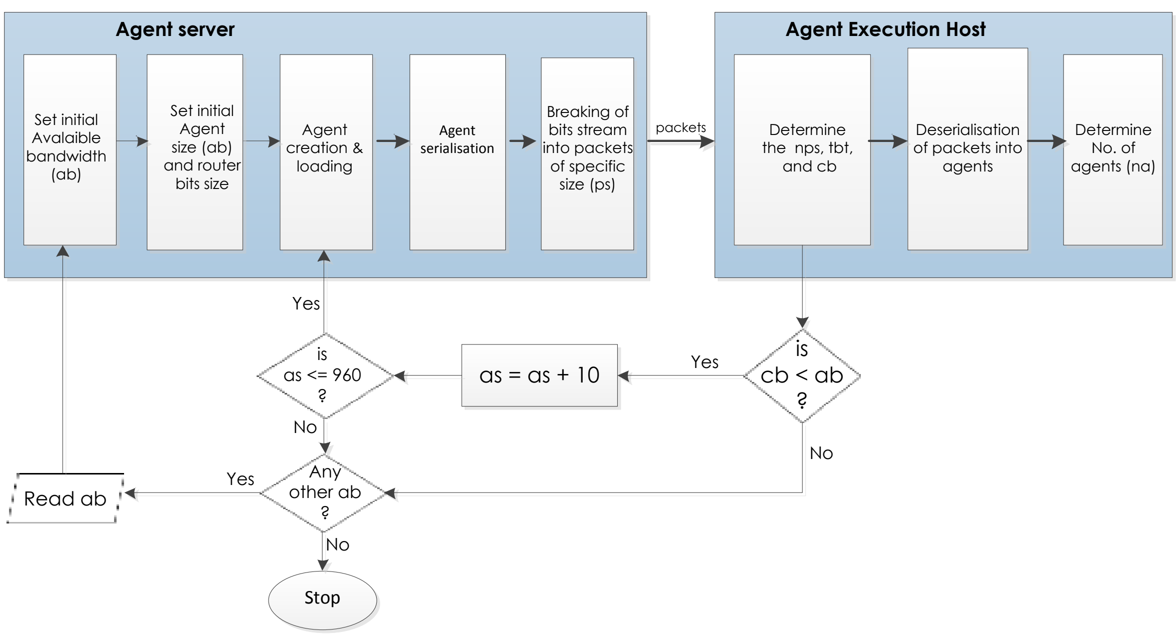


Figure 1 Proposed Threshold-Based Controller (TBC)framework

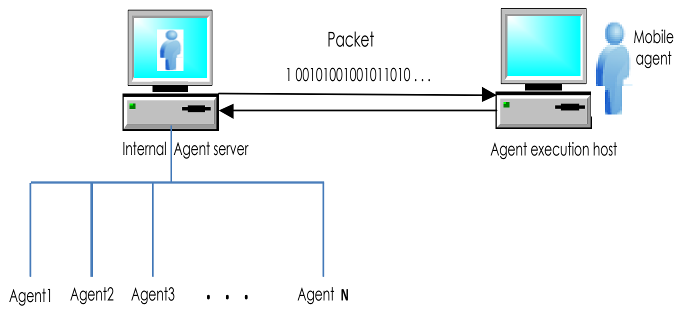


Figure 2 Agent routing concept

Table 2 Notations used in the algorithm

Notation	Meaning
AS	Agent server
EH	Execution host
nps	Number of packets transmitted per second
ps	Packet size
ab	Available bandwidth
ttb	Total transmittedbits
cb	Consumed bandwidth
as	Agent size

```
AS: Sets initial available bandwidth
AS: Sets initial agent size and router bit size
AS: Creates and load agent
AS: Serializes agent
AS: Determines the packet size (ps)
AS: Break bits stream into packets of fixed size
AS: Deploys packet into the network
AS: EH ← packets
EH: Determines the no. of packets per second
EH: Determines the total transmitted bits (ttb)
EH: Computes the consumed bandwidth (consumed
bandwidth (cb) ←ttb/ab)
EH: Deserializes packets into agents
EH: Determine no. of agents
EH: Comparescb and ab
EH: if cb<ab then
EH     increment agent size by 10 bytes
EH:    if as <= 70 bytes then
EH        goto 3
EH:    else consider another available bandwidth
EH:       goto 2
EH:    endif
EH: endif
EH: end
```

Figure 3 Algorithm of TBC

5.0 COMPARATIVE ANALYSIS OF FUZZY LOGIC BASED CONTROLLER WITH THE PROPOSED THRESHOLD-BASED CONTROLLER

The FLC model considered two inputs, that is, available bandwidth and network size and found available bandwidth as critical to the determination of the number of agents deployable to a network of a specific size. Olajubu, Ajayi [2] opined that the available bandwidth is more important than the network size since the network with large number of nodes and narrow bandwidth are apportioned few numbers of agents in order to overcome unnecessary burden on the narrow bandwidth. However, a network of same size but with larger bandwidth accommodates larger number of agents. Conversely, a small network with large bandwidth is faster even with large number of agents routing the network.

Generally, network traffic involves transmission of message and router traffic. In this study, the transmitted message is the serialized agent, while the protocol traffic is the bits stream generated by the transmission protocol for routing the serialized agent from the agent server to the agent execution host. The two traffics consume the available bandwidth. The proposed TBC has recognized and considered the combination of the traffics in the determination of the number of agents appropriate for a network of a specific bandwidth. Furthermore, we considered variation in agent sizes as an important factor too bearing in mind different agent sizes implemented by various agent systems.

6.0 IMPLEMENTATION DETAILS WITH COMPARATIVE ANALYSIS OF RESULTS

The experiment was run three times, one for each of the agent sizes considered. The results of the experiments were consistent as shown in Figures 4a to 4c for the three experiment runs. The graphs show the maximum number of agents that could be deployed into a network giving a specific bandwidth and agent size. From the graph, one can see that the behaviour of the proposed controller is changing as the agent sizes are changing for the varying network bandwidth. This is further explained in Subsection 6.2. On the contrary, the behaviour of FLC is not affected by the changing agent sizes.

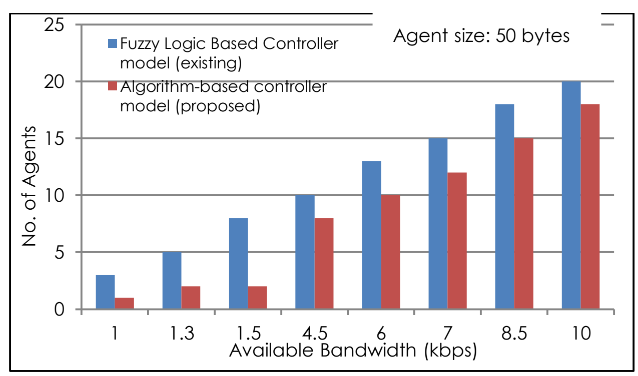


Figure 4a Experimental result 1

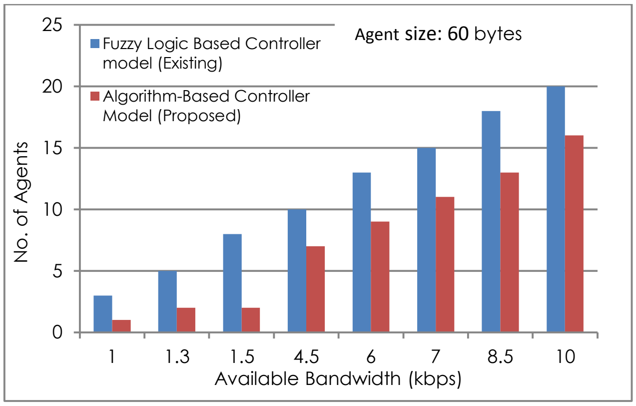


Figure 4b Experimental result 2

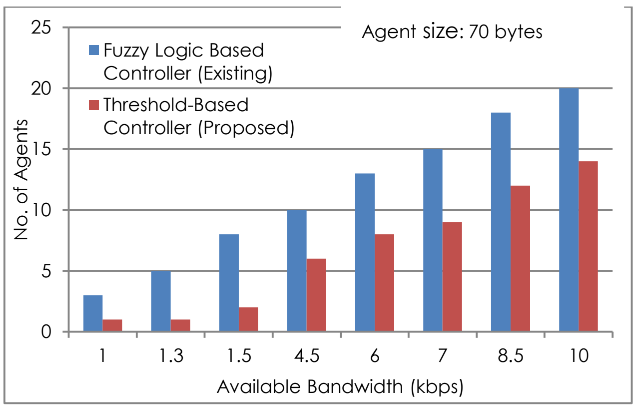


Figure 4c Experimental result 3

6.1 Implementation Details

We develop a java program for the creation and loading of agents, serialize and transmit them from the agent server to the agent execution host. We consider varying agent sizes in bytes(50, 60, 70). For each agent size considered, mobile agents are deployed and the maximum number of agents is obtained for each of the available bandwidths under consideration. To facilitate effective comparison, the bandwidths considered in Olajubu, Ajayi [2] were employed in the implementation.

In the experiment, each packet is made to encapsulate an agent with the router traffic size of 20 bytes. This protocol traffic size is chosen as a result of the small size of agents. It implies that for an agent size of 50 bytes, the packet size is 70 bytes. The packet size increases in the same proportion as the agent size. The packet is deployed into the network and several others until the maximum number is reached. For a given available bandwidth, the required number of packets are transmitted and arrived at the agent execution host in one second. The number of packets arriving at the agent execution host per second is tracked and the total number of bits transmitted is determined to calculate the consumed bandwidth. This process is repeated for all the available bandwidths considered in the experiment.

6.2 Result Analysis

The performance of FLC and the proposed TBC, on the basis of consideration of necessary parameters to reliably determine the maximum deployable number of agents, are analyzed in Section 6. This section verifies the performance of the proposed controller by implementation results. The proposed controller is implemented three times for each of the available bandwidths and similarly for each of the agent sizes considered. The maximum deployable number of agents is obtained in each case as shown in Table 3. The table reveals that no consideration is given to the agent size and protocol traffic size in FLC. Conversely, the proposed controller considered the influence of network traffic (comprising the serialized agent and router traffic) in the determination of accurate number of agents deployable to a network of specific bandwidth. In the experiment conducted, two parameters were fixed while the third was varied. For example, for agent size 50 byte and protocol traffic size 20 bytes, the available bandwidth was varied between 1kbps and 10kbps. Similarly, consider 1kbps available bandwidth and 20 bytes protocol traffic size, the agent sizes were varied between 50 byte and 70 bytes. Furthermore, consider 50 bytes and 70 bytes agent sizes of the proposed controller, maximum of 17 agents, of size 50 bytes, can be deployed into a network of 10kbps bandwidth while 13 agents, of size 70 bytes, can be deployed into the same network.

The consideration of agent size and router traffic size in the proposed controller is responsible for lower maximum deployable number of agents, but more accurate, reliable, and responsive to variation in available bandwidths and network traffic size, than that of FLC as shown in the Table 3. The major reason adduce to this are:

a. Router traffic part of the transmission is left out in FLC as a bandwidth consuming traffic.
b. FLC does not consider variation in agent sizes implemented in different agent systems.

For instance, in FLC, a network of 1kbps bandwidth can route a maximum of 3 agents (of no specific size) while in the proposed controller, the same network can only route a maximum of 1 agent of 50 bytes size. Similarly, a network of 1.3kbps bandwidth can route a maximum of 5 agents of no specific size with FLC while the same network can only route a maximum of 2 agents of 50 bytes size with the proposed controller. Comparatively, it can be inferred from Table 3 that network congestion is imminent in FLC as a result of its neglect of the traffic volume, which is determined by agent size and router traffic size, in the consideration of the number of deployable agents. The congestion subsequently could lead to collision, packet drop and low throughput during packets transmission.

Table 3 Experimental results

									Agent Size (byte)	Protocol Traffic Size (byte)
Number of runs	1	2	3	4	5	6	7	8		
Available Bandwidth (kbps)	1.0	1.3	1.5	4.5	6.0	7.0	8.5	10		
No. of agents (FLC)	3	5	8	10	13	15	18	20	Not applicable	Not applicable
No. of agents (the proposed controller)	1	2	2	8	10	12	15	17	50	20
	1	2	2	7	9	10	13	15	60	
	1	1	2	6	8	9	11	13	70	

7.0 CONCLUSIONAND FUTURE WORK

In this study, we have proposed a research in the field of multi-agent systems and distributed computing, which is part of the focus of this research for the next one year. The main objective is to develop a generic threshold-based controller for regulating number of agents in multi-agent system based applications. The focus is to effectively monitor and control the process so as to avert the possibility of congestion due to excessive deployment of agents. In the course of this study, we observed that most of the multi-agent system based application developers/researchers are yet to adopt a control mechanism that could monitor and regulate the deployment of agents into network.

We have presented a dynamic, implementable and reusable threshold-based multi-agents controller (TBC)to regulate and monitor injection of agents into network as a means to prevent network congestion due to excessive agent deployment with the consideration of agent fundamental routing procedures in computer networks. The performance of the proposed controller is justified by comparing it with the FLC model. Unlike FLC model, the proposed controller is responsive to the changing network bandwidth, agent size and router traffic size in regulating the number of deployable agents that could be routed in the network without compromising its performance. However, further research could be conducted in this field to explore the possibility of using any other control technique(s) that could yield better performance as the research efforts in this field are still very limited.

Acknowledgment

This research is supported by Universiti Teknologi Malaysia through grant: Q.J130000.2513.08H30 and Federal Polytechnic, Ado-Ekiti, Nigeria.

References

[1] Cavalcante, R. C., et al. 2012. A Survey of Security in Multi-Agent Systems. *Expert Systems with Applications*. 39(5): 4835-4846.

[2] Olajubu, E. A., O. Ajayi, and G. A. Aderounmu. 2011. A Fuzzy Logic Based Multi-Agents Controller. *Expert Systems with Applications*. 38(5): 4860-4865.

[3] Aye, N., *et al.* 2013. Multi-Domain Public Key Infrastructure for Information Security with Use of a Multi-Agent System. In *Intelligent Information and Database Systems*. Springer. 365-374.

[4] Li, S. and W. A. Mackaness. 2014. A Multi-Agent-Based, Semantic-Driven System for Decision Support in Epidemic Management. *Health Informatics Journal*. 1460458213517704.

[5] Haghighi, M. and D. Cliff. 2013. Multi-Agent Support for Multiple Concurrent Applications and Dynamic Data-Gathering in Wireless Sensor Networks. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013 Seventh International Conference on*. IEEE.

[6] Ohta, T., T. Hashimoto, and Y. Kakuda. 2011. Self-Organizing Real-Time Service Dissemination and Collection Using Mobile Agents for Mobile Ad Hoc Networks. In *Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW), 2011 14th IEEE International Symposium on*. IEEE.

[7] Hamza, S., B. Kazar Okba, and A. Y. Aicha Nabila. 2012. A Cloud Computing Approach Based on Mobile Agents for Web Service Discovery. In *Second International Conference on Innovative Computing Technology (INTECH), Casablanca*.

[8] Shayestegan, M. and M. H. Marhaban., A Braitenberg. 2012. Approach to Mobile Robot Navigation in Unknown Environments. In *Trends in Intelligent Robotics, Automation, and Manufacturing*. Springer. 75-93.

[9] Wang, M. and J. N. Liu. 2008. Fuzzy Logic-based Real-time Robot Navigation in Unknown Environment with Dead Ends. *Robotics and Autonomous Systems*. 56(7): 625-643.

[10] Motlagh, O., *et al.* 2012. An Expert Fuzzy Cognitive Map for Reactive Navigation of Mobile Robots. *Fuzzy Sets and Systems*. 201: 105-121.

[11] Wai, R.-J., C.-M. Liu, and Y.-W. Lin. 2011. Design of Switching Path-Planning Control for Obstacle Avoidance of Mobile Robot. *Journal of the Franklin Institute*. 348(4): 718-737.