Comparing the Performance of Message Delivery Methods for Mobile Agents

Andrei Olaru, Dragoș Petrescu, Adina Magda Florea

University Politehnica of Bucharest

07.10.20

- Context
- Challenge
- Architecture
- Results
- Conclusion

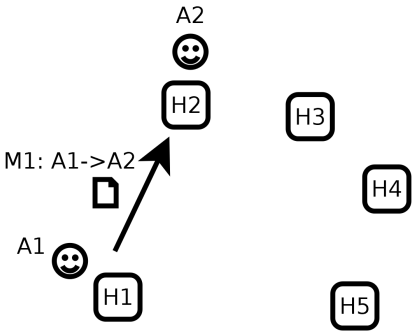
Comparing the Performance of Message Delivery Methods for Mobile Agents

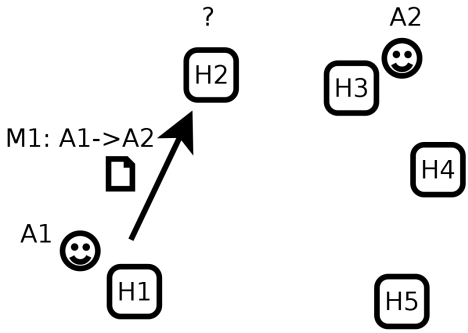
overview

. Andrei Olaru, Dragoș Petrescu, Adina Magda Florea

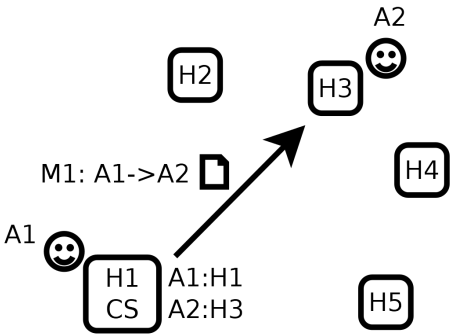
Mobile agents allow for an agentification model in which mobile code is part of autonomous agents which move between hosts (or nodes).

Applications include High-Performance Computing, fog computing, smart cities, and others, where moving computation units can be modeled as agents.

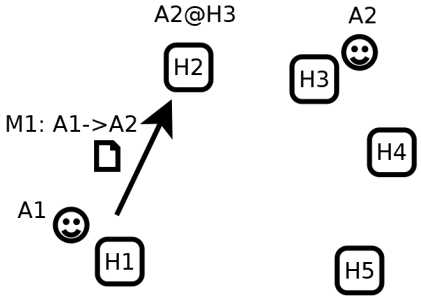

A2
H2
H3
M1: A1->A2
H4
A1
H1
H5



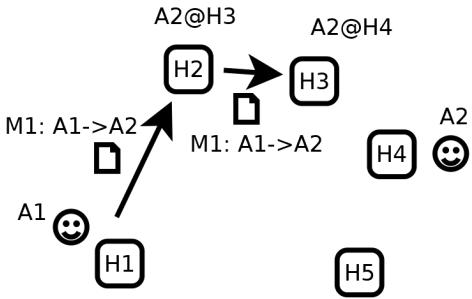
⇒ implementation of various *message delivery models* (or *protocols*)



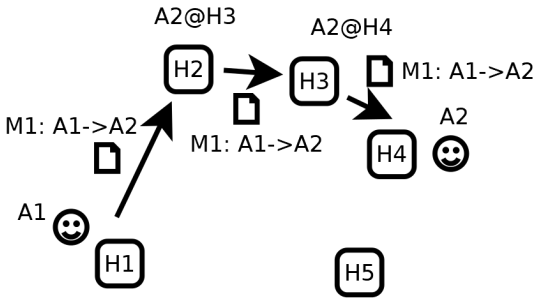
Central Server Scheme



Forwarding Proxy



Forwarding Proxy



Forwarding Proxy

Many message delivery models have been proposed:

- *centralized* or partially centralized: Central Server Scheme; Home Server Scheme; hierarchical solutions based on domains [Wojciechowski, 2001] and regions [Yousuf and Hammo, 2012] ;
- *blackboard* solutions [Cabri et al., 2000, Choi et al., 2006] ;
- *forwarding proxy* solutions, including the Shadow Protocol and region-based solutions [Baumann and Rothermel, 1998, Di Stefano and Santoro, 2002] ;
- combinations of forwarding proxies and location servers [Jingyang et al., 2003, Cao et al., 2005, Roman et al., 2018]

Many message delivery models have been proposed:

- *centralized* or partially centralized: Central Server Scheme; Home Server Scheme; hierarchical solutions based on domains [Wojciechowski, 2001] and regions [Yousuf and Hammo, 2012] ;
- *blackboard* solutions [Cabri et al., 2000, Choi et al., 2006] ;
- *forwarding proxy* solutions, including the Shadow Protocol and region-based solutions [Baumann and Rothermel, 1998, Di Stefano and Santoro, 2002] ;
- combinations of forwarding proxies and location servers [Jingyang et al., 2003, Cao et al., 2005, Roman et al., 2018]

Which one is the *best* message delivery model, in each specific situation?

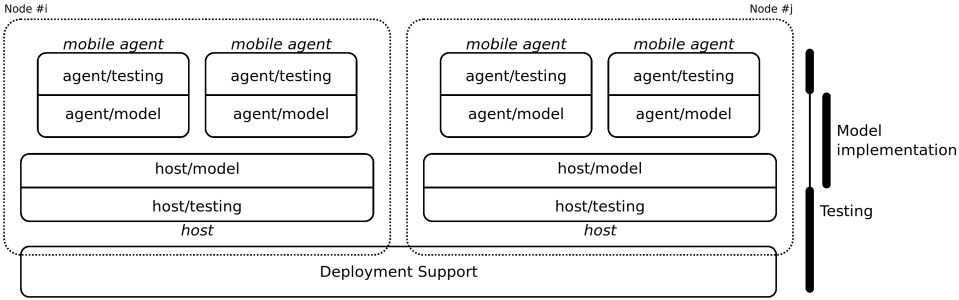
Challenge

· Our goal was to build a framework for the comparison of message delivery protocols in different difficult scenarios.

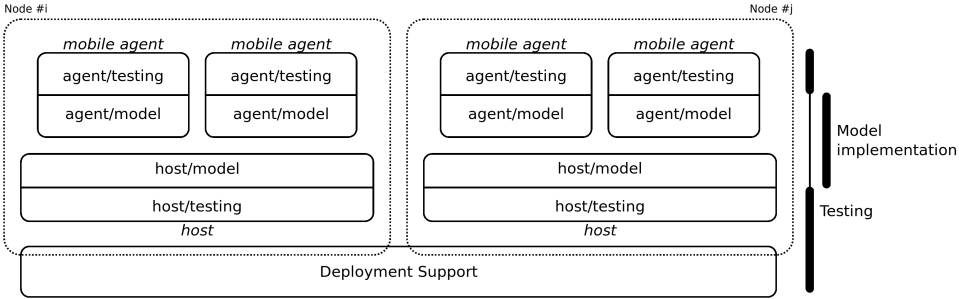
· Our contribution is twofold:

- the comparison framework
- a comparison between well-established message delivery protocols

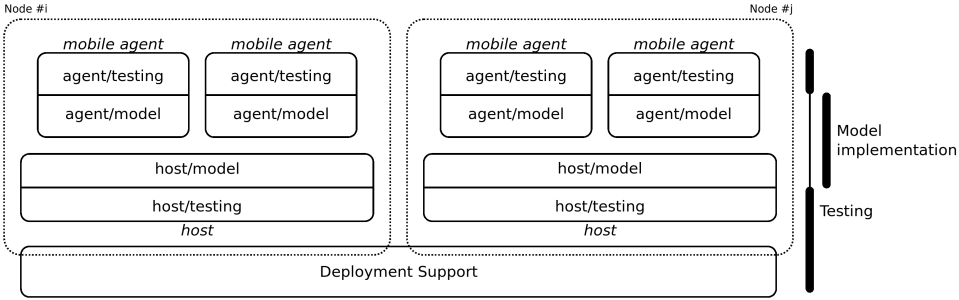
- the framework should separate message delivery model implementation from scenario generation and experimentation, such that:
  - the implementation of the message delivery model can be changed easily, without modifying the experimentation part
  - the implementation of a specific message delivery model can be used in a multi-agent application deployed in real life, exactly how it was used in the framework.
- scenarios can be generated to simulate difficult situations:
  - there are many messages
  - there are many agents who move around the system
  - agents move around the system very quickly
- the results of experiments can be analyzed quantitatively



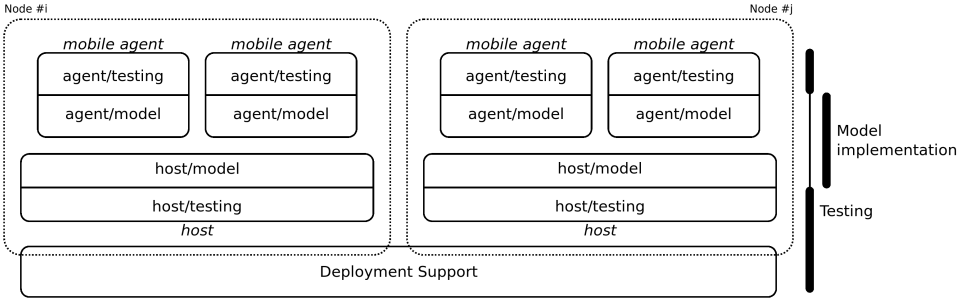
deployment support – modelling and simulating the underlying infrastructure which allows communication between hosts, complete with network topology and communication latencies.



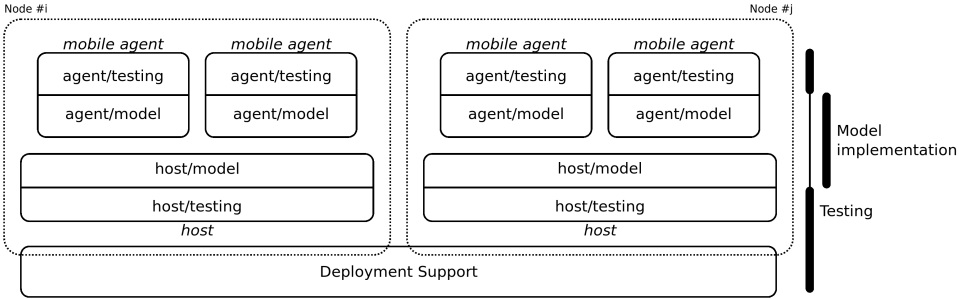
agent/model – the model-specific implementation that is bound to a mobile agent.



agent/model – the model-specific implementation that is bound to a mobile agent.
host/model – the model-specific implementation that is bound to a host.



agent/testing – the framework-related implementation that is bound to a mobile agent, sending and receiving messages according to a given scenario.



agent/testing – the framework-related implementation that is bound to a mobile agent, sending and receiving messages according to a given scenario.
host/testing – the framework-related implementation that is bound to a host (e.g. for packing/unpacking moving mobile agents).

	Agent/testing	Agent/model	Host/model	Host/testing	Deployment support
	1. A new message M is generated according to the scenario				
	2. M is passed to Agent/model				
Host A		3. M is prepared according to the delivery model			
			4. M is assigned a next hop		
				5. M is sent to the next hop	
					6. M travels through network
				7. M reaches next hop	
Host B			8. M is is assigned new next hop		
				9. M is sent to the next hop	
					10. M travels through network
				11. M reaches next hop	
			12. M passed to target agent/model		
Host C		13. M passed to target agent/testing			
	14. M reaches destination and measurements are updated				

	Agent/testing	Agent/model	Host/model	Host/testing	Deployment support
	1. A new message M is generated according to the scenario				
	2. M is passed to Agent/model				
Host A		3. M is prepared according to the delivery model			
			4. M is assigned a next hop		
				5. M is sent to the next hop	
					6. M travels through network
				7. M reaches next hop	
Host B			8. M is is assigned new next hop		
				9. M is sent to the next hop	
					10. M travels through network
				11. M reaches next hop	
			12. M passed to target agent/model		
Host C		13. M passed to target agent/testing			
	14. M reaches destination and measurements are updated				

	Agent/testing	Agent/model	Host/model	Host/testing	Deployment support
	1. A new message M is generated according to the scenario				
	2. M is passed to Agent/model				
Host A		3. M is prepared according to the delivery model			
			4. M is assigned a next hop		
				5. M is sent to the next hop	
					6. M travels through network
				7. M reaches next hop	
Host B			8. M is is assigned new next hop		
				9. M is sent to the next hop	
					10. M travels through network
				11. M reaches next hop	
			12. M passed to target agent/model		
Host C		13. M passed to target agent/testing			
	14. M reaches destination and measurements are updated				

	Agent/testing	Agent/model	Host/model	Host/testing	Deployment support
	1. A new message M is generated according to the scenario				
	2. M is passed to Agent/model				
Host A		3. M is prepared according to the delivery model			
			4. M is assigned a next hop		
				5. M is sent to the next hop	
					6. M travels through network
				7. M reaches next hop	
Host B			8. M is is assigned new next hop		
				9. M is sent to the next hop	
					10. M travels through network
				11. M reaches next hop	
			12. M passed to target agent/model		
Host C		13. M passed to target agent/testing			
	14. M reaches destination and measurements are updated				

	Agent/testing	Agent/model	Host/model	Host/testing	Deployment support
	1. A new message M is generated according to the scenario				
	2. M is passed to Agent/model				
Host A		3. M is prepared according to the delivery model			
			4. M is assigned a next hop		
				5. M is sent to the next hop	
					6. M travels through network
				7. M reaches next hop	
Host B			8. M is is assigned new next hop		
				9. M is sent to the next hop	
					10. M travels through network
				11. M reaches next hop	
			12. M passed to target agent/model		
Host C		13. M passed to target agent/testing			
	14. M reaches destination and measurements are updated				

	Agent/testing	Agent/model	Host/model	Host/testing	Deployment support
	1. A new message M is generated according to the scenario				
	2. M is passed to Agent/model				
Host A		3. M is prepared according to the delivery model			
			4. M is assigned a next hop		
				5. M is sent to the next hop	
					6. M travels through network
				7. M reaches next hop	
Host B			8. M is is assigned new next hop		
				9. M is sent to the next hop	
					10. M travels through network
				11. M reaches next hop	
			12. M passed to target agent/model		
Host C		13. M passed to target agent/testing			
	14. M reaches destination and measurements are updated				

	Agent/testing	Agent/model	Host/model	Host/testing	Deployment support
	1. A new message M is generated according to the scenario				
	2. M is passed to Agent/model				
Host A		3. M is prepared according to the delivery model			
			4. M is assigned a next hop		
				5. M is sent to the next hop	
					6. M travels through network
				7. M reaches next hop	
Host B			8. M is is assigned new next hop		
				9. M is sent to the next hop	
					10. M travels through network
				11. M reaches next hop	
			12. M passed to target agent/model		
Host C		13. M passed to target agent/testing			
	14. M reaches destination and measurements are updated				

	Agent/testing	Agent/model	Host/model	Host/testing	Deployment support
	1. A new message M is generated according to the scenario				
	2. M is passed to Agent/model				
Host A		3. M is prepared according to the delivery model			
			4. M is assigned a next hop		
				5. M is sent to the next hop	
					6. M travels through network
				7. M reaches next hop	
Host B			8. M is is assigned new next hop		
				9. M is sent to the next hop	
					10. M travels through network
				11. M reaches next hop	
			12. M passed to target agent/model		
Host C		13. M passed to target agent/testing			
	14. M reaches destination and measurements are updated				

	Agent/testing	Agent/model	Host/model	Host/testing	Deployment support
	1. A new message M is generated according to the scenario				
	2. M is passed to Agent/model				
Host A		3. M is prepared according to the delivery model			
			4. M is assigned a next hop		
				5. M is sent to the next hop	
					6. M travels through network
				7. M reaches next hop	
Host B			8. M is is assigned new next hop		
				9. M is sent to the next hop	
					10. M travels through network
				11. M reaches next hop	
			12. M passed to target agent/model		
Host C		13. M passed to target agent/testing			
	14. M reaches destination and measurements are updated				

	Agent/testing	Agent/model	Host/model	Host/testing	Deployment support
	1. A new message M is generated according to the scenario				
	2. M is passed to Agent/model				
Host A		3. M is prepared according to the delivery model			
			4. M is assigned a next hop		
				5. M is sent to the next hop	
					6. M travels through network
				7. M reaches next hop	
Host B			8. M is is assigned new next hop		
				9. M is sent to the next hop	
					10. M travels through network
				11. M reaches next hop	
			12. M passed to target agent/model		
Host C		13. M passed to target agent/testing			
	14. M reaches destination and measurements are updated				

	Agent/testing	Agent/model	Host/model	Host/testing	Deployment support
	1. A new message M is generated according to the scenario				
	2. M is passed to Agent/model				
Host A		3. M is prepared according to the delivery model			
			4. M is assigned a next hop		
				5. M is sent to the next hop	
					6. M travels through network
				7. M reaches next hop	
Host B			8. M is is assigned new next hop		
				9. M is sent to the next hop	
					10. M travels through network
				11. M reaches next hop	
			12. M passed to target agent/model		
Host C		13. M passed to target agent/testing			
	14. M reaches destination and measurements are updated				

	Agent/testing	Agent/model	Host/model	Host/testing	Deployment support
	1. A new message M is generated according to the scenario				
	2. M is passed to Agent/model				
Host A		3. M is prepared according to the delivery model			
			4. M is assigned a next hop		
				5. M is sent to the next hop	
					6. M travels through network
				7. M reaches next hop	
Host B			8. M is is assigned new next hop		
				9. M is sent to the next hop	
					10. M travels through network
				11. M reaches next hop	
			12. M passed to target agent/model		
Host C		13. M passed to target agent/testing			
	14. M reaches destination and measurements are updated				

	Agent/testing	Agent/model	Host/model	Host/testing	Deployment support
	1. A new message M is generated according to the scenario				
	2. M is passed to Agent/model				
Host A		3. M is prepared according to the delivery model			
			4. M is assigned a next hop		
				5. M is sent to the next hop	
					6. M travels through network
				7. M reaches next hop	
Host B			8. M is is assigned new next hop		
				9. M is sent to the next hop	
					10. M travels through network
				11. M reaches next hop	
			12. M passed to target agent/model		
Host C		13. M passed to target agent/testing			
	14. M reaches destination and measurements are updated				

	Agent/testing	Agent/model	Host/model	Host/testing	Deployment support
	1. A new message M is generated according to the scenario				
	2. M is passed to Agent/model				
Host A		3. M is prepared according to the delivery model			
			4. M is assigned a next hop		
				5. M is sent to the next hop	
					6. M travels through network
				7. M reaches next hop	
Host B			8. M is is assigned new next hop		
				9. M is sent to the next hop	
					10. M travels through network
				11. M reaches next hop	
			12. M passed to target agent/model		
Host C		13. M passed to target agent/testing			
	14. M reaches destination and measurements are updated				

The scenarios are generated with the ability of setting:

- the number of nodes and the underlying network topology;
- the number of agents in the scenario;
- the probability of an agent migrating to another node in a given time unit (between 1 in 1000 and 1 in 10);
- the probability of an agent sending a message to another agent in a given time unit (between 1 in 100 and 1);
- the "CPU" power of hosts, specifying how many messages a host can process in a time unit - a number of 1 to 50 messages processed in every time unit;

Several quantitative results are returned by a scenario simulation:

$$\textit{Delivery rate} = \frac{\textit{number of messages which have been delivered}}{\textit{total number of sent messages}}$$

Several quantitative results are returned by a scenario simulation:

$$\textit{Delivery rate} = \frac{\textit{number of messages which have been delivered}}{\textit{total number of sent messages}}$$

$$\textit{Mean Delivery Time} = \frac{\sum_{i}^{message} \textit{message delivery time for message } i}{\textit{total number of messages}}$$

Several quantitative results are returned by a scenario simulation:

$$\textit{Delivery rate} = \frac{\textit{number of messages which have been delivered}}{\textit{total number of sent messages}}$$

$$\textit{Mean Delivery Time} = \frac{\sum_i^{message} \textit{message delivery time for message } i}{\textit{total number of messages}}$$

$$\textit{Network Load} = \frac{\sum_i^{steps} \textit{number of messages in transit at time unit } i}{\textit{number of steps}}$$

Several quantitative results are returned by a scenario simulation:

$$\textit{Delivery rate} = \frac{\textit{number of messages which have been delivered}}{\textit{total number of sent messages}}$$

$$\textit{Mean Delivery Time} = \frac{\sum_i^{\textit{message}} \textit{message delivery time for message } i}{\textit{total number of messages}}$$

$$\textit{Network Load} = \frac{\sum_i^{\textit{steps}} \textit{number of messages in transit at time unit } i}{\textit{number of steps}}$$

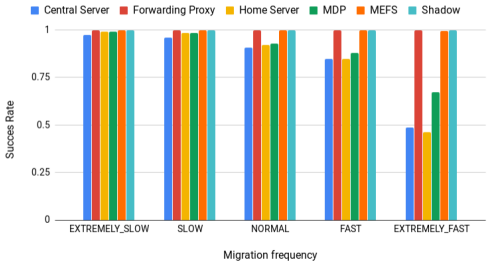
$$\textit{Wasted Time} = \frac{\sum_{\textit{failed messages}} \textit{time spent in transit}}{\textit{number of failed messages}}$$

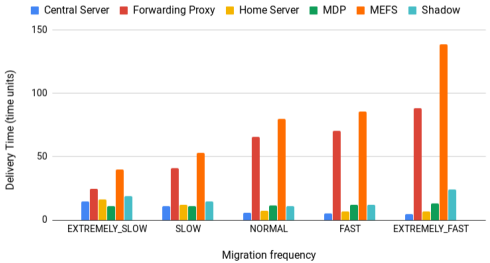
Experimental Results

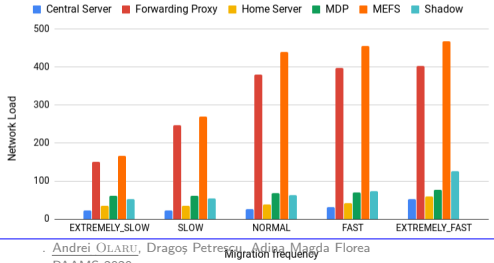
Several well-established protocols have been implemented:

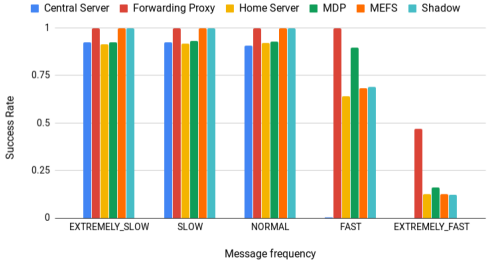
- Central Scheme(CS) – one host is informed of the location of all agents;
- Home Server Scheme (HSS) – each agent is assigned to a specific host, its *home server*, which knows where the agent is;
- Forwarding Proxy (FP) – an agent leaves a proxy on the host from which it moves; the proxy relays messages to the next hop;
- Shadow Protocol – combines HSS and FP; home server updated periodically;
- Message Efficient Forwarding Schema (MEFS) – combines CS and FP, with periodic updates to CS;
- Message Delivery Protocol (MDP) – creates a hierarchical topology for routing messages;
- Blackboard – agents need to visit the host where the blackboard is located in order to get their messages;
- Reliable Asynchronous Message Delivery Protocol (RAMDP) – groups messages in regions, each region with its own blackboard;

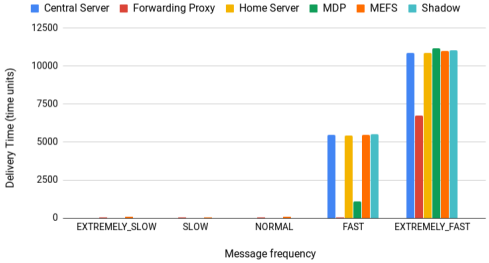
Experimental Results

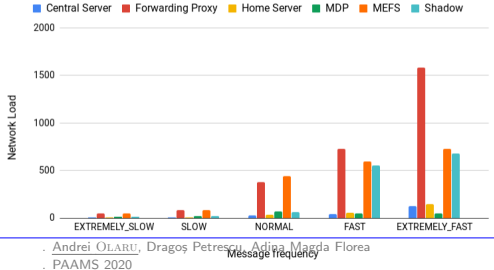







Central Server
Forwarding Proxy
Home Server
MDP
MEFS
Shadow
Success Rate
1
0.75
0.5
0.25
0
EXTREMELY_SLOW
SLOW
NORMAL
FAST
EXTREMELY_FAST
Message frequency


Central Server
Forwarding Proxy
Home Server
MDP
MEFS
Shadow
Delivery Time (time units)
12500
10000
7500
5000
2500
0
EXTREMELY_SLOW
SLOW
NORMAL
FAST
EXTREMELY_FAST
Message frequency


Central Server
Forwarding Proxy
Home Server
MDP
MEFS
Shadow
Network Load
2000
1500
1000
500
0
EXTREMELY_SLOW
SLOW
NORMAL
FAST
EXTREMELY_FAST
Message frequency

Conclusion & Future Work

- We have developed a framework for the comparison of message delivery protocols in mobile multi-agent systems.
- A user of the framework is able to set-up experimental scenarios with the required properties and analyze how various protocols handle the situation.
- A user is able to use an already implemented model or implement a new model and test it against existing ones.

- Faulty network and hosts will be modelled in order to check the robustness of the message delivery.
- Temporal distributions for scenario parameters will be modelled, analyzing the connection between the position in the distribution and the current value for qualitative measures.

Thank You!

Any Questions?

Baumann, J. and Rothermel, K. (1998).
The shadow approach: An orphan detection protocol for mobile agents.
Personal and Ubiquitous Computing, 2(2):100–108.

Cabri, G., Leonardi, L., and Zambonelli, F. (2000).
Mobile-agent coordination models for internet applications.
Computer, 33(2):82–89.

Cao, J., Xu, W., Chan, A. T., and Li, J. (2005).
A reliable multicast protocol for mailbox-based mobile agent communications.
In Proceedings Autonomous Decentralized Systems, 2005. ISADS 2005., pages 74–81. IEEE.

Choi, S., Kim, H., Byun, E., Hwang, C., and Baik, M. (2006).
Reliable asynchronous message delivery for mobile agents.
IEEE Internet Computing, 10(6):16–25.

Di Stefano, A. and Santoro, C. (2002).
Locating mobile agents in a wide distributed environment.
IEEE transactions on parallel and distributed systems, 13(8):844–864.

Jingyang, Z., Zhiyong, J., and Daoxu, C. (2003).
Designing reliable communication protocols for mobile agents.
In 23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings., pages 484–487. IEEE.

Roman, R., López, J., and Mambo, M. (2018).
Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges.
Future Generation Comp. Syst., 78:680–698.

Wojciechowski, P. T. (2001).
Algorithms for location-independent communication between mobile agents.
In Proc. of AISB'01 Symposium on Software Mobility and Adaptive Behaviour.

Yousuf, A. Y. and Hammo, A. (2012).
Developing a new mechanism for locating and managing mobile agents.
J. Eng. Sci. Technol, 7(5):614–622.

Thank You!

Any Questions?