

Article

# Agent-Based Model for Automaticity Management of Traffic Flows across the Network

Karina Raya-Díaz <sup>1,\*</sup> , Carelia Gaxiola-Pacheco <sup>1</sup>, Manuel Castañón-Puga <sup>1</sup> , Luis E. Palafox <sup>1</sup>,  
Juan R. Castro <sup>1</sup> and Dora-Luz Flores <sup>2</sup> 

<sup>1</sup> Facultad de Ciencias Químicas e Ingeniería, Universidad Autónoma de Baja California, Calzada Universidad 14418, Tijuana 22390, Mexico; cgaxiola@uabc.edu.mx (C.G.-P.); puga@uabc.edu.mx (M.C.-P.); lepalafox@uabc.edu.mx (L.E.-P.); jrcastror@uabc.edu.mx (J.R.C.)

<sup>2</sup> Facultad de Ingeniería, Arquitectura y Diseño, Universidad Autónoma de Baja California, Carretera Transpeninsular Ensenada, Tijuana 3917, Colonia Playitas, Ensenada, Baja California, Mexico; dflores@uabc.edu.mx

\* Correspondence: karina.raya@uabc.edu.mx; Tel.: +52-664-979-7500 (ext. 54303)

Received: 1 July 2017; Accepted: 4 September 2017; Published: 9 September 2017

**Abstract:** This paper presents an agent-based model that performs the management of traffic flows in a network with the purpose of observing in a simulation of distinctive congestion scenarios how the automation of the monitoring task improves the network performance. The model implements a decision-making algorithm to determine the path that the data flows will follow to reach their destination, according to the results of the negotiation between the agents. In addition, we explain how the behavior of the network is affected by its topology. The aim of this paper is to propose an agent-based model that simplifies the management of the traffic flows in a communications network towards the automaticity of the system.

**Keywords:** agent-based modeling; complex network; multi-agent system; network management

## 1. Introduction

The communication networks are complex systems that support a wide variety of applications, many of them in real time; this has led to a change in the network management scheme. Network administrators must deploy multiple tools to obtain the visibility and control necessary to operate today's complex networks, some of these tools are: software agents, active networks and policy languages [1].

The integration of intelligent agents into the management system facilitates the evaluation of the parameters that determine the network performance. An agent has the capacity to detect changes in the environment, and respond in accordance, in order to isolate or anticipate failures making the system proactive; thus, the direct intervention of the network administrator becomes unnecessary in the situations that the agents can handle.

Agent-based modeling is a powerful tool for complex system simulation. A computer network can be described as a complex system because it is dynamic, has nonlinear interactions, and its components exhibit unpredictable reactions that results in behavior patterns [2,3]. In the proposed model, when a micro behavior, as a congestion, emerges in a network node, the agent needs to make a decision according to its preferences.

The topology of a network contributes valuable information to analyze its structure. Network managing considering the complex networks approach allows the identification and classification of the nodes using the metrics of centrality and clustering degree, among others.

Autonomic network management is a complex task and consists in the automaticity of the system to protect, configure, optimize and heal itself without human intervention. In networks, the property

of automaticity allows self-management, given a set of policies from the administrator, which depicts the agent's high-level goals [4]. The proposed model applies a distributed management to avoid attacks over the central nodes in the network.

This paper is structured as follows: in Section 2, autonomic networking management is defined, in Section 3, network metrics based on graph theory principles are presented. Next, in Section 4, we introduce the architecture of the agent-based model and the algorithm proposed is explained. In Section 5, the simulation experiment and the results obtained are presented. Finally, concluding remarks are given in Section 6.

## 2. Autonomic Networking Management

Today, people are using networks to communicate, and networks have become bigger and more complex. The network management is a complicated task that goes beyond technical support and device configuration. To simplify the network management, the automation of the management tasks are critical. The International Business Machines Corporation (IBM) has proposed five levels according to the degree of automaticity in the network management [4,5].

- Level 1: Refers to the continuous monitoring of the elements of the system by means of software tools handled by people who provide manual support in the event of a failure.
- Level 2: Consists of a set of monitoring applications that intelligently collect information to reduce the workload of the network administrator, and is known as the management level.
- Level 3: This level can recognize certain behavior patterns of the network in addition to suggesting actions to be taken by the support staff; this level is also called predictive.
- Level 4: Uses level 3 tools but attempts to minimize human intervention by helping with Service Level Agreements (SLAs); this level is known as adaptive.
- Level 5: Business policies and objectives dynamically manage the system at this level. Its disadvantage is that the automaticity is very closed only based on certain rules, and this level is recognized as autonomous.

The proposed agent-based model is a step forward to reach the fifth level of automaticity, by the integration of intelligent agents that determine the path that will follow the data flows when a congested node situation emerges. The rules used for these agents rely on the preferences of each network node.

## 3. Measures of Centrality in Networks

A network is a set of nodes and links [6]. Modeling a network helps to determine the structural issues relevant at a given time. The topology of the network can be illustrated applying an adjacency matrix or adjacency list. The decision of which one has to be used depends on the analysis type.

An adjacency matrix is a mathematical representation of a network, which allows the analysis of the clustering relation between the nodes. When search algorithms are used to enumerate components, the disadvantage of this matrix is the inefficient use of memory, but this is not a problem when a particular element is accessed [7,8]. Different techniques for network interpretation and visualization are used nowadays; one of these techniques is based on clustering algorithms, in which the use of an adjacency matrix is more convenient.

One of the measures that is applied to adjacency matrix is the centrality, which captures the hierarchy of the network nodes [9]. The awareness of the node relevance is crucial to manage any network because this will lead to the determination of the main nodes. The nodes are classified by their betweenness centrality. Eigenvalue centrality is an extension of degree centrality [8], and the topology of the network has an effect on both measures. The use of a node classification based

on the centrality allows the network manager to realize the importance of the nodes. A definition of eigenvalue centrality of a node  $i$  is shown in Equation (1):

$$x'_i = \sum_j A_{ij}x_j, \quad (1)$$

where  $x'_i$  is defined by the sum of the  $i$ 's the centrality of  $i$  neighbors, and  $A_{ij}$  is an element of the adjacency matrix. The interpretation of eigenvalues tells if a node is growing or shrinking according to the amount of neighbors. In the model, this measure is used to identify the hierarchy of each node. Links allow communication to reach a remote node. Here, the betweenness centrality of a node has a relevant role. Clustering coefficient  $C_i$  describes how extended a node is, Equation (2) shows how to calculate it, where Equation (3) is the degree of the node  $k(i)$  and  $N(i)$  is the amount of neighbors of the node  $i$  [2,10]:

$$C_i = \frac{k_i(k_i - 1)}{2}, \quad (2)$$

$$k(i) = |N(i)|. \quad (3)$$

In a heterogeneous network, the nodes with higher clustering degree are called hubs. In [11], it is affirmed that high degree nodes play a very significant role because the packets are routed through these hubs, and, in a social network, the people highly connected distribute information better than isolated people [12].

Figure 1 shows a representation of hub nodes in a network where the size of the nodes is proportional to its degree, defined as  $k(i) = |N(i)|$ , where  $N(i)$  is the amount of the neighbors of a node  $i$  [10], using this metric, a node hierarchy is obtained. The proposed agent-based model detects clusters of nodes when the flow preferences are assigned.

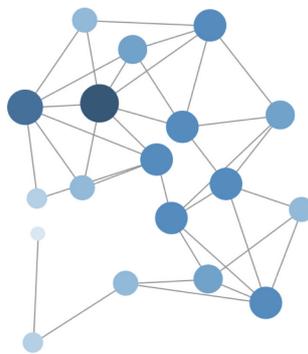


Figure 1. Example of network with representation of nodes by degree.

#### 4. Architecture of the Agent-Based Model

An agent-based model is useful to analyze an approach for decision-making under conditions of deep uncertainty because it has the ability to connect heterogeneous micro and macro behaviors [13]. In a network model, the micro behaviors represent the interactions of different types of flows (data, video and voice), each one gives information for understanding the network behavior. The flow analysis is critical to the network availability and performance [14]. Because of this, the proposed agent-based model incorporates into the nodes a set of collaborative agents to collect the information requested by the decision-making level.

#### 4.1. Multi-Agent Architecture for Management of Network Flows

The architecture of the proposed multi-agent model is shown in Figure 2. The Multi-Agent Architecture for Management of Network Flows, TELEKA, is organized by hierarchy using three levels: Network, Control and Negotiation.

- **Network:** In this level, all nodes are set up with the type of flows that will transmit and their destination. In addition, the agents of each node are initializing with its hierarchy obtained by Equation (1), and these preferences are assigned by the network administrator depending on their policies, and their utilities are set to zero. The agents collect information related to congestion status by sensing the current state of the nodes. When a congestion is recognized by exceeding a threshold, an ECG flag changed their value to one, and this will trigger the activation of the negotiation level.
- **Control:** Here, the agents detect and classify the flows that arrive at each node. After this, the agents provide the data to the Negotiation level, which holds the decision-making module.
- **Negotiation:** The algorithm SEHA (Social Election with Hidden Authorities) is triggered in this level when a congestion situation emerges. As a result, a set of actions to be performed by the lower-level agents is selected, achieving with this the optimization of the network status.

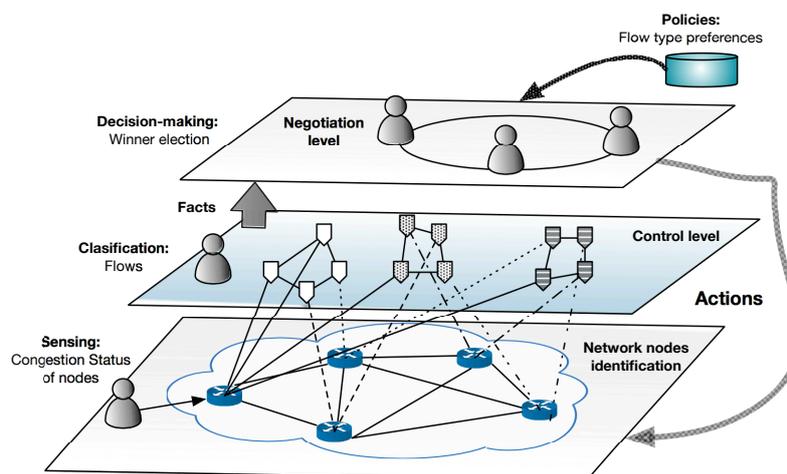


Figure 2. Multi-Agent Architecture for Management of Network Flows.

#### 4.2. SEHA Social Election with Hidden Authorities Algorithm

According to the social choice theory, the group decision-making process in a multi-agent system should consider that each agent will have their preferences and will vote to choose a winner [15]. The proposed model considers a set of agents  $A = \{Ag_1, \dots, Ag_n\}$  in a network environment composed by  $n$  number of nodes and  $k$  links with different costs  $c$ .

The tuple that represents an agent is  $Ag_i = \langle \Omega, \beta, h \rangle$ , where  $\Omega$  is a set of preferences' relations  $\Omega = \{ \omega_1 \succ \omega_2 \succ \omega_3 \}$ , they are arranged according to the flow type to transmit and its priority  $\beta$ . Both parameters  $\Omega$  and  $\beta$  are configured by the network administrator during the initialization process. The hierarchy of a node  $h$  is calculated by the centrality of each node among the network using the Equation (1).

The proposed algorithm is activated when a congestion emerges in a node, as is described in Algorithm 1. The SEHA algorithm returns a winner  $\omega$  flow type (data, video or voice) after a voting process between the neighbor's nodes. The winner flow will have the higher priority in the queue of the congested node. The next step is transmitting the flow through one of its neighboring nodes using the shortest path with minimum cost  $c$  to its destination, the agent of the selected node obtains an utility  $u$  of +1. The rest of the non-preferred flows in the queue of the congested node picks randomly

a neighboring node to continue their paths, but the utility  $u$  of the agent is set to  $-1$  because they are not transmitting the preferred flows.

---

**Algorithm 1** SEHA Social Election with Hidden Authorities Algorithm

---

```

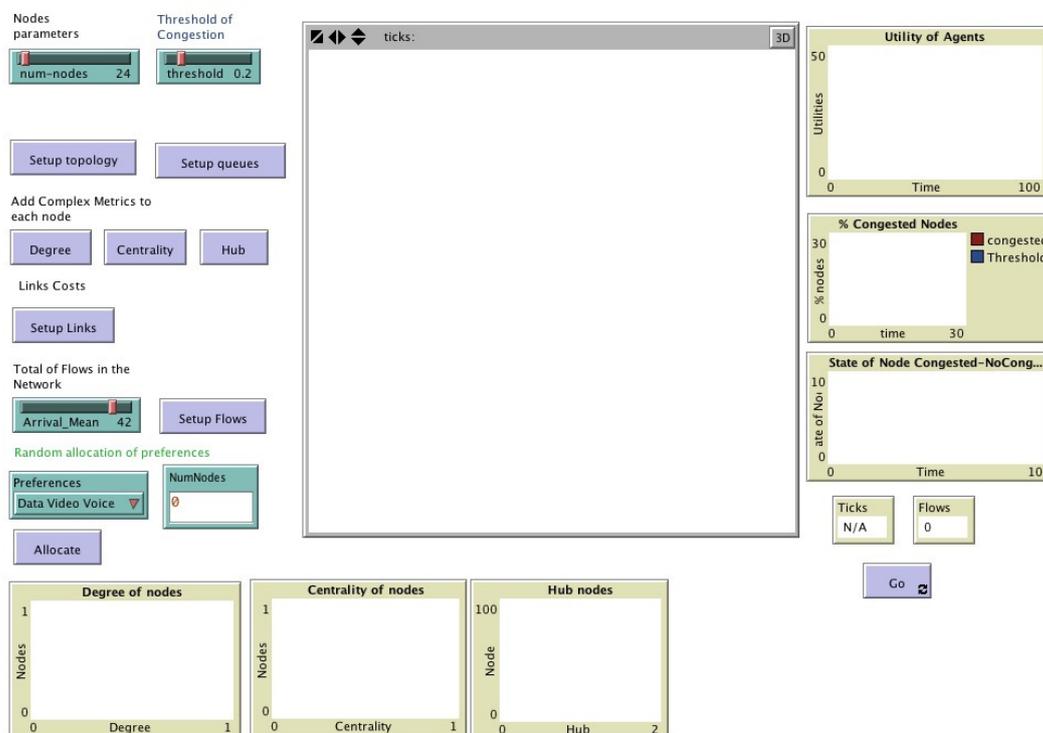
function FINDWINNER(  $\Omega, \beta, h$ )
    set choices [ $\omega$  with max  $\beta$ ] of my_neighbors
    set choices modes of choices
    if choices length > 1
        set preferred_choice [ $\omega$  with max  $\beta$ ] of my_neighbors with max  $h$ 
    else
        set preferred_choice choices
    return preferred_choice
end function
    
```

---

**5. Simulated Scenarios and Results**

The proposed model analyses the behavior of a network when a congestion situation emerges. In particular, this model aims to enable the implementation of SEHA ( Social Election with Hidden Authorities Algorithm) algorithm to automatically manage the traffic flows across the network. The model validates the TELEKA architecture, which was illustrated in Figure 2.

Figure 3 presents the user interface in Netlogo of the model, and also shows the nodes' parameters as number, degree, centrality, queues threshold of congestion, and topology of the network. The environment where the agents coexist is a network in which the interconnection of the nodes and links are generated by scale-free distribution [16–18].



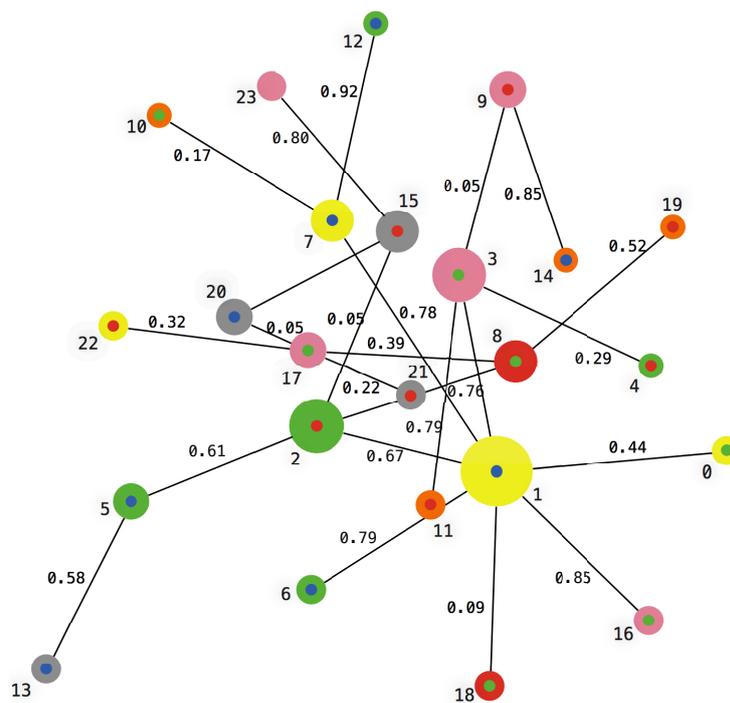
**Figure 3.** Netlogo interface of the model.

The experiment has the initialization values listed in Table 1, the hierarchy  $h$ , which corresponds to the values of centrality of each node, are calculated using the Equation (1), and the values of the preferences  $\omega$  of each node are listed. Link costs are listed in Table A1 in Appendix A.

**Table 1.** Initialization parameters in the network.

Node	h	Preferences			Node	h	Preferences		
		$\omega_1$	$\omega_2$	$\omega_3$			$\omega_1$	$\omega_2$	$\omega_3$
0	0.34	Voice	Video	Data	12	0.15	Data	Voice	Video
1	1.00	Voice	Video	Data	13	0.079	Video	Data	Voice
2	0.604	Data	Voice	Video	14	0.069	Voice	Data	Video
3	0.533	Data	Video	Voice	15	0.272	Video	Data	Voice
4	0.181	Data	Voice	Video	16	0.34	Data	Video	Voice
5	0.232	Data	Voice	Video	17	0.104	Data	Video	Voice
6	0.34	Data	Voice	Video	18	0.34	Video	Voice	Data
7	0.442	Voice	Video	Data	19	0.092	Voice	Data	Video
8	0.272	Video	Voice	Data	20	0.104	Video	Data	Voice
9	0.204	Data	Video	Voice	21	0.035	Video	Data	Voice
10	0.15	Voice	Data	Video	22	0.035	Voice	Video	Data
11	0.181	Voice	Data	Video	23	0.092	Data	Video	Voice

In Figure 4, the color of each node represents its preferences (see Table A2 in Appendix A). The sizes of the nodes are proportional to their degree  $k$ . Finally, the dots inside of each node represents the flows waiting to be transmitted to their destination nodes.



**Figure 4.** Topology with scale free distribution.

In Netlogo, the next process to activate after the setup of the world is called  $G_0$ , and this is illustrated in Figure 5. The flow diagram represents the actions that are executed within a tick.

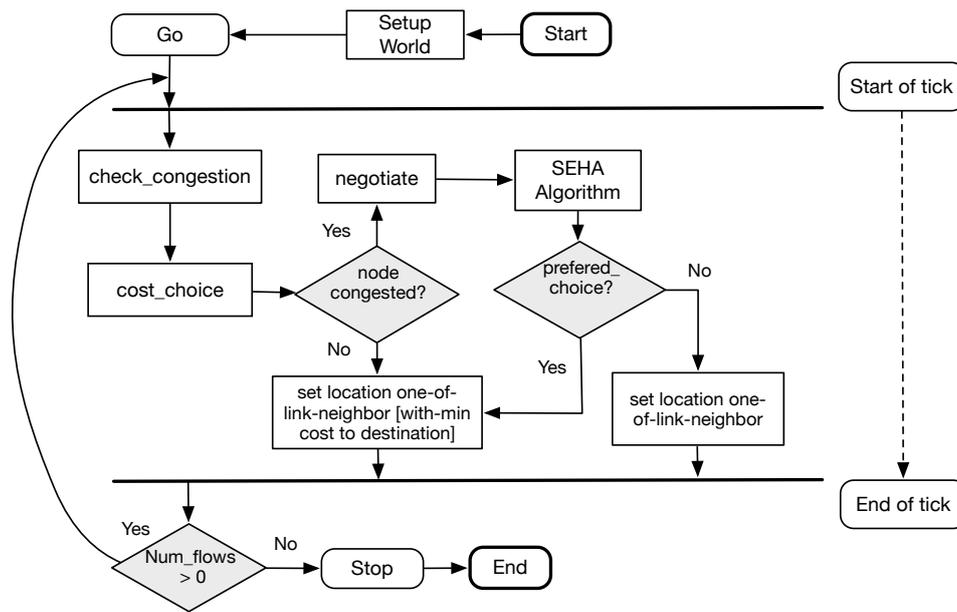


Figure 5. Flowchart of Go process.

### 5.1. Analysis of Results

Four different scenarios were simulated to analyze the results of the Netlogo model. Two topologies were used: the first topology considers a scale-free distribution and the second one considers a grade distribution. With each topology, the Algorithm 1 was applied, creating the first two scenarios, and it was not applied creating the two remaining scenarios.

### 5.2. Simulation Using the Topology with Scale Free Distribution

Figure 6 illustrates the histograms obtained from the measures of centrality and the classification of all nodes as hubs or not hubs. The histogram of degree allows the identification of the nodes with the higher degrees as those which are important to consider when a congestion starts.

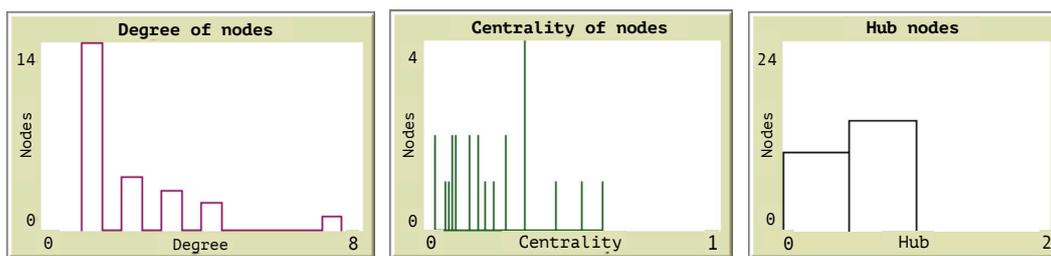
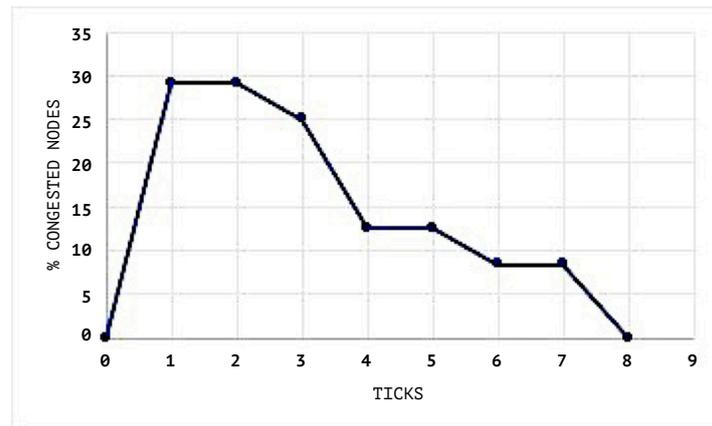
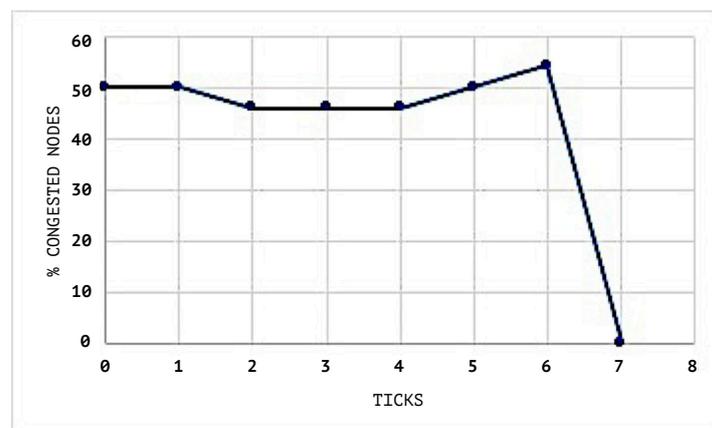


Figure 6. Histogram of complex metrics obtained from the model.

The network during the simulation maintains a maximum congestion limit of 30%. The congestion threshold of each node queue is set to 20%. This means that a value above this will change the status of the node to congested. Figure 7 shows the congestion status of the network when the SEHA Algorithm 1 was applied and Figure 8 shows the congestion status of the network without using a SEHA Algorithm 1.

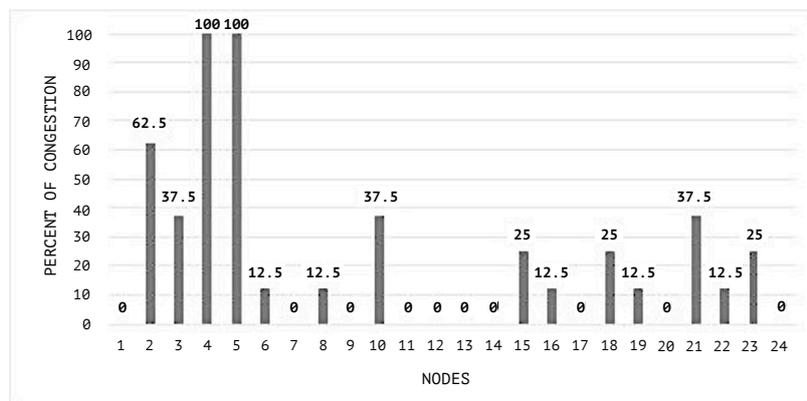


**Figure 7.** Congestion status of the network over the simulation applying the SEHA (Social Election with Hidden Authorities Algorithm).

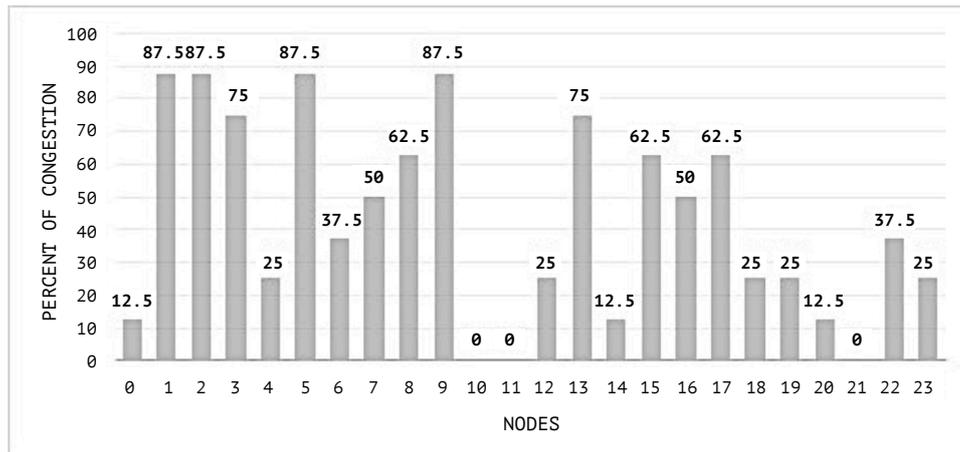


**Figure 8.** Congestion status of the network in simulation using Random distribution.

In Figures 9 and 10, the congestion averages by node are shown, and it can be noticed that the congested nodes amount are higher than those in the scenarios where SEHA Algorithm 1 was applied.

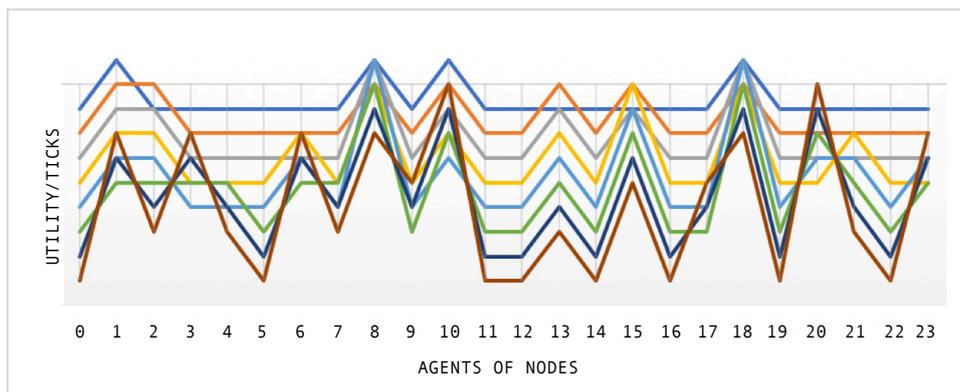


**Figure 9.** Congestion average by node during the simulation using the SEHA (Social Election with Hidden Authorities Algorithm).



**Figure 10.** Congestion average by node during the simulation with Random distribution.

After each agent negotiates, they obtain a utility reward, and Figure 11 shows the utility of all agents during the simulation time using the SEHA algorithm. During the simulation with a random flow distribution, the agents receive no utility. Both models complete the distribution of 42 flows through the network to its destination node at the eighth tick.



**Figure 11.** Utilities of each agent when the SEHA (Social Election with Hidden Authorities Algorithm) is applied.

The values obtained to generate this graph are shown in Table A3 of Appendix A.

### 5.3. Simulation Using Topology with Degree Distribution

The following simulation was configured with different topology, using the degree of six as a parameter to generate it. Figure 12 shows the view of the network in the simulator.

Figure 13 shows the complex metrics of degree, centrality and a classification of nodes in hubs or no hubs obtained by the model when the simulation was initialized.

The results obtained by the second simulation are illustrated in Figures 14 and 15. Compared with the results shown in Figure 14, the higher congested status of the network was less than 26% using the SEHA Algorithm 1.

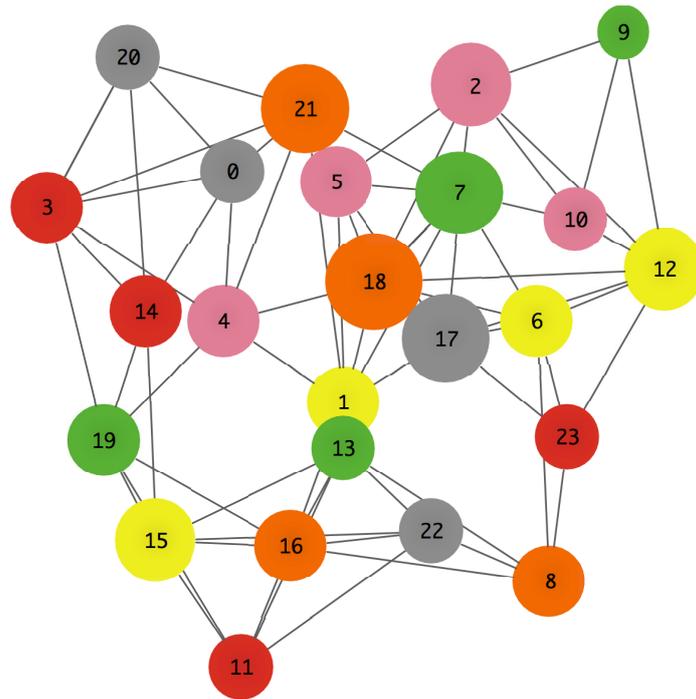


Figure 12. Topology with degree distribution each color represents the type of preferences.

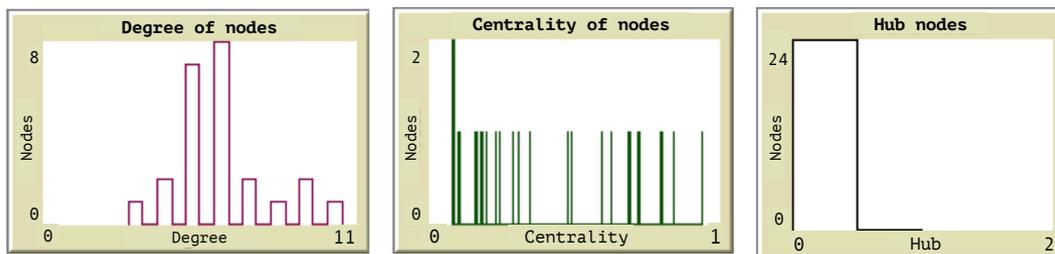


Figure 13. Histograms of the complex metrics obtained.

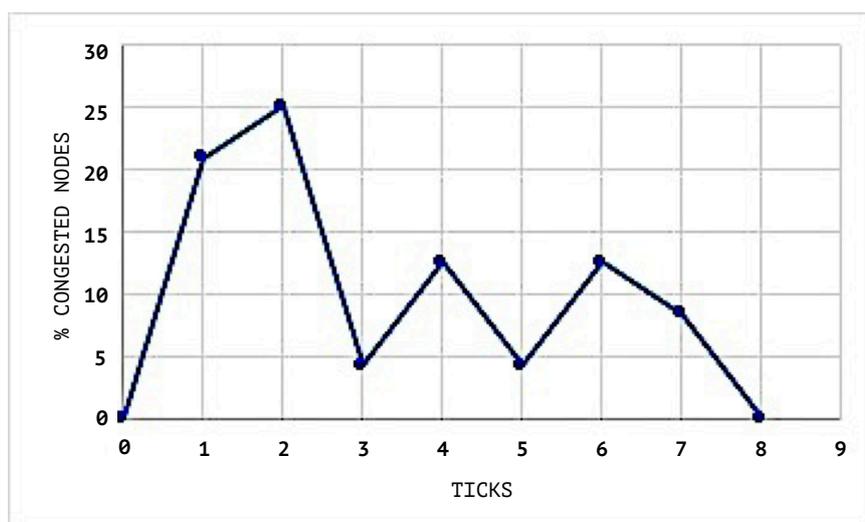
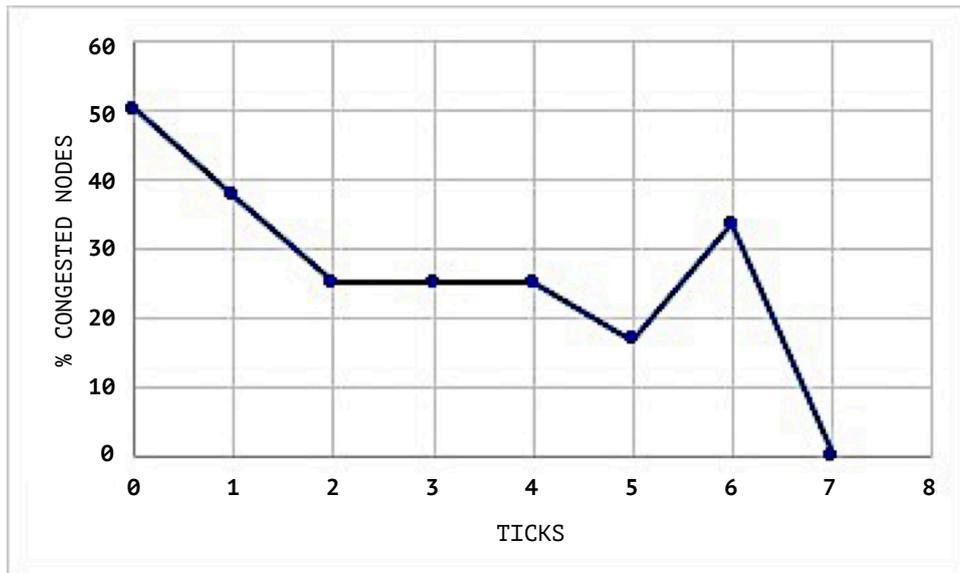


Figure 14. Average of congestion status of the network in the simulation applying the SEHA (Social Election with Hidden Authorities Algorithm).

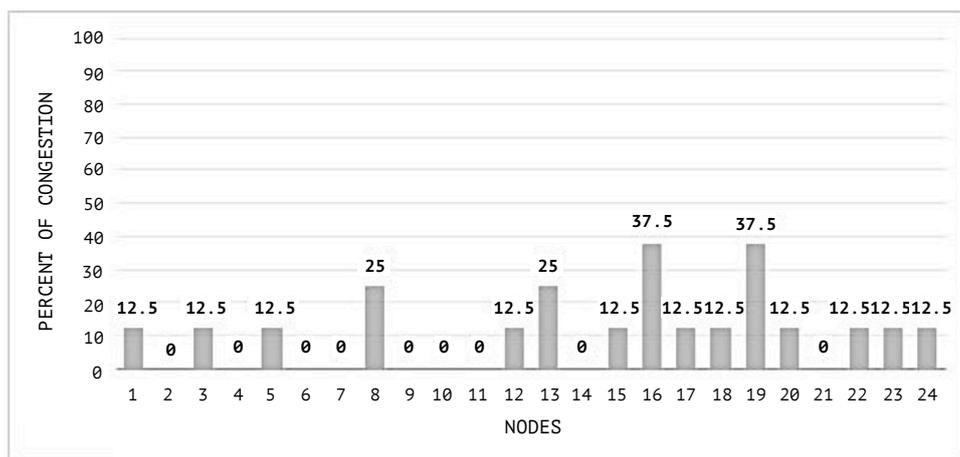


**Figure 15.** Average of congestion in the network during simulation using random distribution.

Detailed information of the congestion average by node during both simulations is shown in Figures 16 and 17.

According to the illustrations in Figures 16 and 17, the impact of congestion on the nodes of the network is lower using the SEHA (Social Election with Hidden Authorities Algorithm) than when a random distribution is used.

Finally, in Figure 18, the utilities of agents during the simulation are shown. The utilities obtained by the agents represent the most visited nodes when a congestion emerges using the SEHA Algorithm. Random distribution does not obtain utilities because there is no negotiation between agents to determine the new hop of the flow. For better description of the result values, see Table A4 in Appendix A.



**Figure 16.** Congestion average by node during the simulation using the SEHA (Social Election with Hidden Authorities Algorithm).

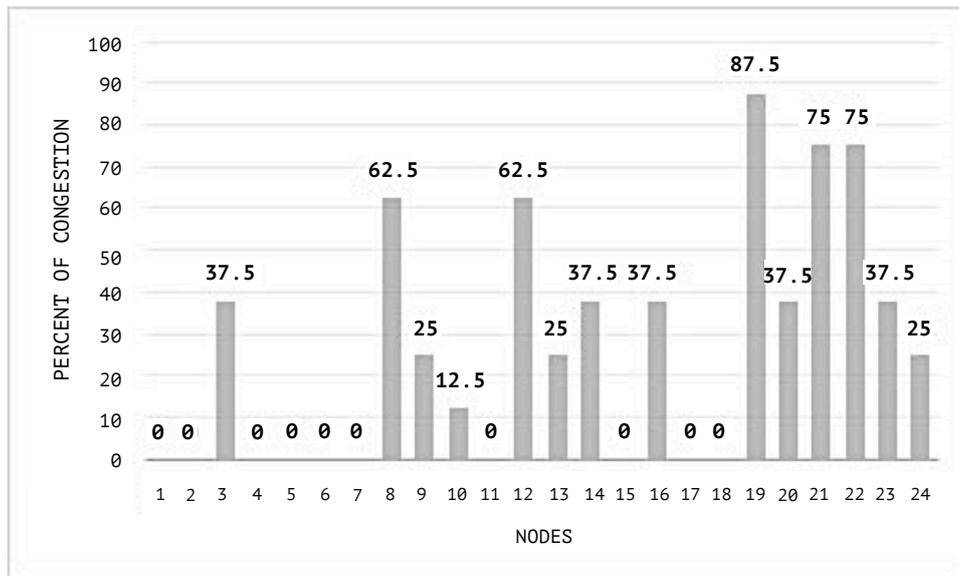


Figure 17. Congestion average by node during the simulation with Random distribution.

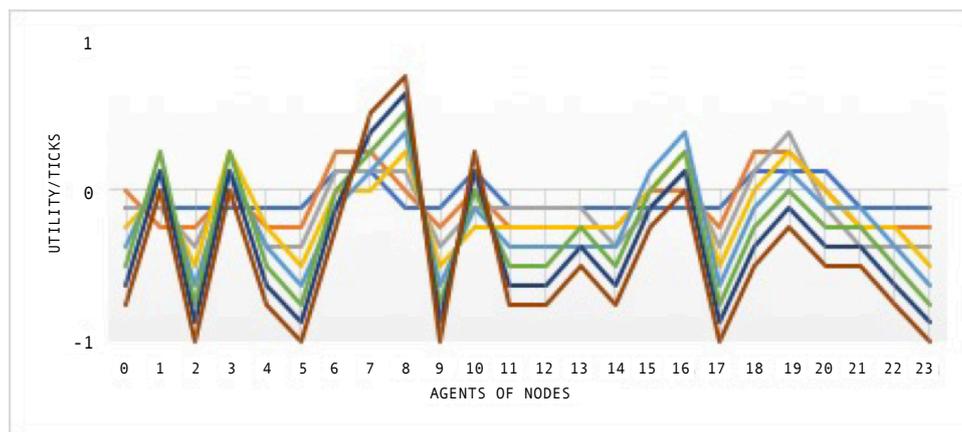


Figure 18. Utilities of each agent when applying the SEHA (Social Election with Hidden Authorities Algorithm).

#### 5.4. Comparison of Results

In Table 2, the outcome of 600 simulations are shown. The scenarios that were used for these simulations were obtained by means of the link cost variation and a random initial distribution of the flows. The comparison was made between the scenarios where the SEHA (Social Election with Hidden Authorities Algorithm) was applied, in contrast with those in which the flows distribution was performed in a random manner.

As described 300 additional simulations were generated to verify the behavior of the SEHA (Social Election with Hidden Authorities Algorithm), varying the topology, node preferences and link costs. In addition, 300 simulations were generated by configuring BehaviorSpace following the same procedure, but the flow distribution was performed in a random manner, instead of using the SEHA (Social Election with Hidden Authorities Algorithm). The outcome using NetLogo Behavior Space with the initial values specified in Figure 19 is illustrated in Table 2.

The use of agent-based models to evaluate the performance of an algorithm is a suitable tool. The results shows that, although the variation in the costs greatly influences the random distribution, the affectation is minor when the SEHA (Social Election with Hidden Authorities Algorithm) is applied.

In Figure 20, the behavior of both simulated scenarios are shown, where it is observed that the outcome variation is smaller when the SEHA (Social Election with Hidden Authorities Algorithm) algorithm is applied than when it is not.

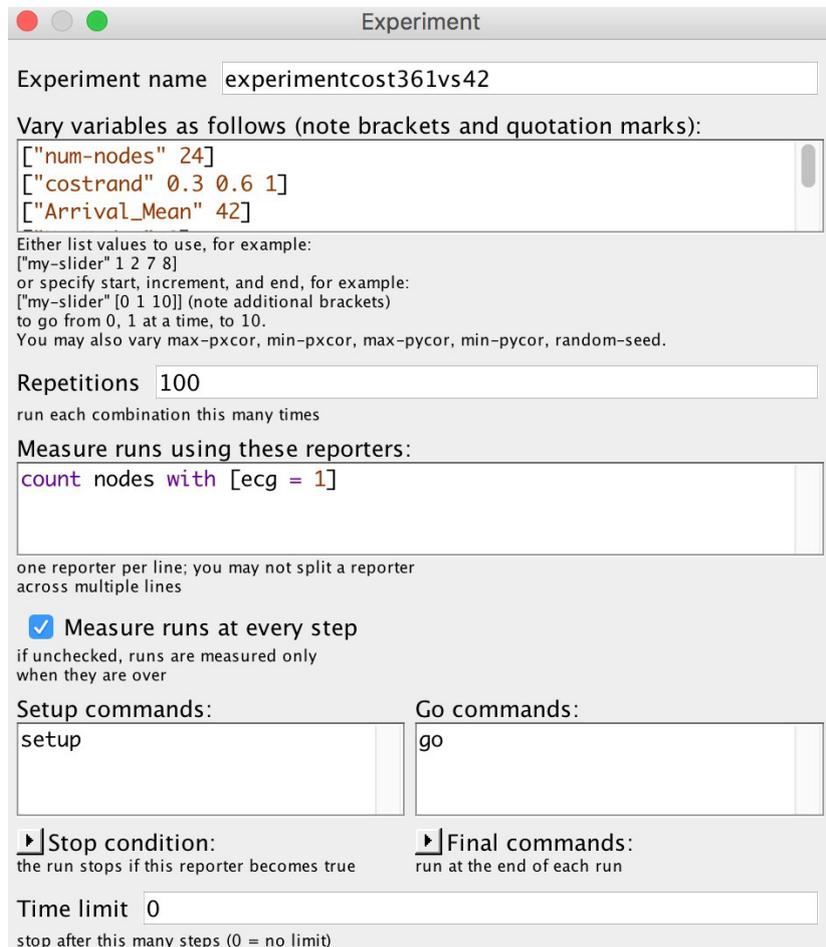
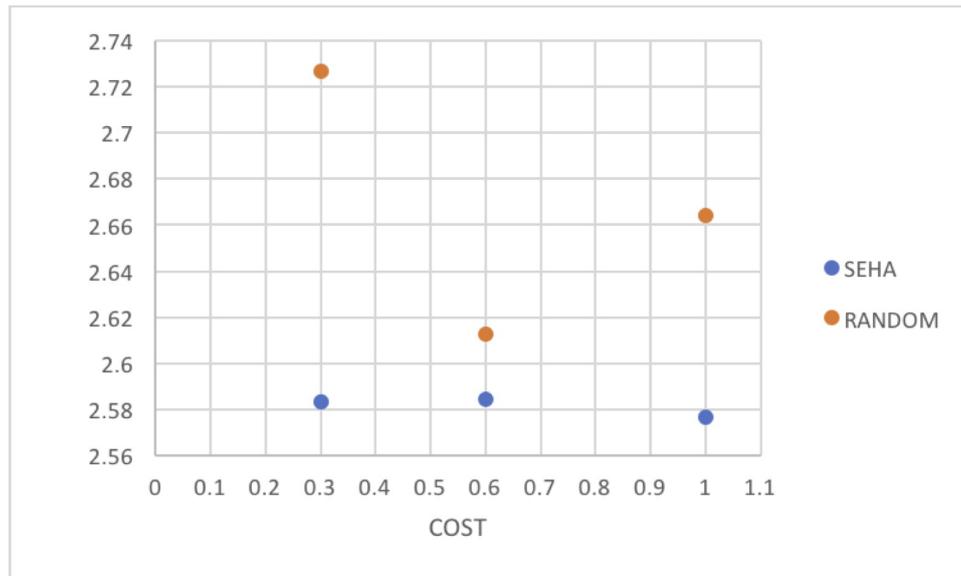


Figure 19. Initialization of Behavior Space of Netlogo.

Table 2. Results obtained after 600 simulations with Behavior Space of NetLogo.

Number of Runs	Flows Mean Arrival Rate	Cost	Social Election with Hidden Authorities Algorithm		Random Distribution	
			Mean of Congested Nodes	Standard Deviation	Mean of Congested Nodes	Standard Deviation
100	42	0.3	2.583	0.447	2.726	0.550
100	42	0.6	2.584	0.402	2.612	0.531
100	42	1.0	2.576	0.446	2.664	0.554



**Figure 20.** Comparison of performance using the SEHA (Social Election with Hidden Authorities Algorithm) and Random distribution of the flows.

## 6. Future Work

The integration of an agent that allows the automatic adaptation of network policies would bring us closer to the goal of automating the network-management tasks. This algorithm can be integrated into systems that require self-organization considering the connectivity between its components.

The proposed algorithm can be applied in a social network, where it is possible to identify those members who have a high hierarchy given their number of connections, in order to prioritize the messages that come from high hierarchy members.

## 7. Conclusions

The agent-based model allows the analysis of the results obtained to identify the patterns of behavior of a network. In addition, the model provides information about how the congestion of a network impacts the transmission of flows when classifying the nodes by their centrality.

Using the model to obtain the measures of complex networks of degree clustering and centrality by considering the network topology is a way to identify the hierarchy of each node. The identification of the nodes with greater centrality is one of the critical parameters of the SEHA Algorithm 1, which allows establishing a tiebreaker and a better flow distribution.

Flow transmission in the network are optimized according to the preferences of the community to which the congested node belongs, which exempts the network administrator from resolving the congestion failure by selecting manually the type of flow that must be transmitted according to its priority.

Considering that complex network metrics in a self-management system improves the network performance by decreasing congestion, this algorithm mainly considers the nodes with greater hierarchy in the network to transmit its flows because they have higher priority. The integration of intelligent algorithms that allows the autonomous negotiation of the resources improves the network performance.

**Acknowledgments:** This work was financially supported by CONACYT (Consejo Nacional de Ciencia y Tecnología) under the MYDCI (Mestría y Doctorado en Ciencias e Ingeniería) program at the Autonomous University of Baja California by the contract #101358.

**Author Contributions:** The corresponding authors Karina Raya-Díaz and Carelia Gaxiola-Pacheco proposed the research and drafted the manuscript. Karina Raya-Díaz was responsible for the development of the agent-based

model. Manuel Castañón-Puga and Juan R. Castro gave helpful advice on the paper preparation and contributed towards experimental evaluation. Luis E. Palafox were involved in the network analysis and Dora-Luz Flores gave recommendations for the policies rules.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

The cost of all links are random values, in the range of zero to one, and are listed in Table A1.

The preference values of each node are shown in Table A2, and these values are referenced in the Figure 4 of the Simulated Experiment and Results section. In Table A2, the flow type are Data = 0, Voice = 1 and Video = 2, and the order has an impact on the flow priority. The maximum size of queue in each node is set to 10 and the threshold set in the simulation are 20%. This means that, over this value, the bit of congestion ECG will be set to one, so a negotiation starts.

Table A4 provides the list of the agents' utilities in the second simulation using the SEHA algorithm and a network topology based on a degree distribution.

**Table A1.** Bidirectional links' costs.

Cost	Link	
	End1	End2
0.442	node 1	node 0
0.759	node 1	node 3
0.788	node 1	node 6
0.778	node 1	node 7
0.848	node 1	node 16
0.668	node 2	node 1
0.293	node 4	node 3
0.609	node 5	node 2
0.577	node 5	node 13
0.165	node 7	node 10
0.217	node 8	node 2
0.392	node 8	node 17
0.517	node 8	node 19
0.045	node 9	node 3
0.850	node 9	node 14
0.790	node 11	node 3
0.924	node 12	node 7
0.053	node 15	node 2
0.322	node 17	node 22
0.089	node 18	node 1
0.678	node 20	node 15
0.364	node 21	node 20
0.801	node 23	node 15

**Table A2.** Initialization values of nodes.

Source Node	Color	Preferences	Num. Flows	Type Flows	Destination Node
0	yellow	1 2 0	1	1	21
1	yellow	1 2 0	4	1,0,2,0	10,21,7,9
2	green	0 1 2	1	2	14
3	pink	0 2 1	2	1,1	22,6
4	green	0 1 2	3	1,2,2	22,3,21
5	green	0 1 2	5	1,2,2,0,0	8,7,21,15,7
6	green	0 1 2	3	1,0,0	22,10,21
7	yellow	1 2 0	1	0	6
8	red	2 1 0	1	1	20
9	pink	0 2 1	1	2	7
10	orange	1 0 2	1	1	17
11	orange	1 0 2	2	0,2	19,5
12	green	0 1 2	2	1,0	1,2
13	gray	2 0 1	2	1,0	21,8
14	orange	1 0 2	3	1,2,0	23,23,1
15	gray	2 0 1	1	2	20
16	pink	0 2 1	1	1	1
17	pink	0 2 1	4	1,2,0,1	11,20,7,4
18	red	2 1 0	2	2,1	23,20
19	orange	1 0 2	2	2,2	12,10
20	gray	2 0 1	2	1,0	5,19
21	gray	2 0 1	2	2,2	8,19
22	yellow	1 2 0	3	2,1,2	13,8,3
23	pink	0 2 1	0	-	-

**Table A3.** Agents' utilities after negotiation.

Ticks/Agent in Node	1	2	3	4	5	6	7	8
node 0	-1	-1	-1	-1	-1	-1	-1	-1
node 1	1	-1	-1	-1	-1	-1	1	1
node 2	-1	1	-1	-1	-1	-1	-1	-1
node 3	-1	-1	-1	-1	-1	1	1	1
node 4	-1	-1	-1	-1	-1	1	-1	-1
node 5	-1	-1	-1	-1	-1	-1	-1	-1
node 6	-1	-1	-1	1	-1	-1	1	1
node 7	-1	-1	-1	-1	-1	1	-1	-1
node 8	1	-1	1	-1	1	-1	-1	-1
node 9	-1	-1	-1	-1	-1	-1	1	1
node 10	1	-1	-1	-1	-1	1	1	1
node 11	-1	-1	-1	-1	-1	-1	-1	-1
node 12	-1	-1	-1	-1	-1	-1	-1	-1
node 13	-1	1	-1	-1	-1	-1	-1	-1
node 14	-1	-1	-1	-1	-1	-1	-1	-1
node 15	-1	1	-1	1	-1	-1	-1	-1
node 16	-1	-1	-1	-1	-1	-1	-1	-1
node 17	-1	-1	-1	-1	-1	-1	1	1
node 18	1	-1	1	-1	1	-1	-1	-1
node 19	-1	-1	-1	-1	-1	-1	-1	-1
node 20	-1	-1	-1	-1	1	1	1	1
node 21	-1	-1	-1	1	-1	-1	-1	-1
node 22	-1	-1	-1	-1	-1	-1	-1	-1
node 23	-1	-1	-1	-1	1	-1	1	1

Table A4. Agents' utilities.

Ticks /Agent in Node	1	2	3	4	5	6	7	8
node 0	-1	1	-1	-1	-1	-1	-1	-1
node 1	-1	-1	1	1	1	1	-1	-1
node 2	-1	-1	-1	-1	-1	-1	-1	-1
node 3	-1	1	1	1	-1	1	-1	-1
node 4	-1	-1	-1	1	-1	-1	-1	-1
node 5	-1	-1	-1	-1	-1	-1	-1	-1
node 6	1	1	-1	-1	-1	1	-1	-1
node 7	1	1	-1	-1	1	1	1	1
node 8	-1	1	1	1	1	1	1	1
node 9	-1	-1	-1	-1	-1	-1	-1	-1
node 10	1	-1	-1	-1	1	1	1	1
node 11	-1	-1	-1	-1	-1	-1	-1	-1
node 12	-1	-1	1	-1	-1	-1	-1	-1
node 13	-1	-1	1	-1	-1	1	-1	-1
node 14	-1	-1	-1	1	-1	-1	-1	-1
node 15	-1	1	-1	1	1	-1	-1	-1
node 16	-1	1	1	1	1	-1	-1	-1
node 17	-1	-1	-1	-1	-1	-1	-1	-1
node 18	1	1	-1	-1	-1	-1	-1	-1
node 19	1	1	1	-1	-1	-1	-1	-1
node 20	1	-1	-1	1	-1	-1	-1	-1
node 21	-1	-1	-1	1	1	-1	-1	-1
node 22	-1	-1	-1	1	-1	-1	-1	-1
node 23	-1	-1	-1	-1	-1	-1	-1	-1

## References

1. Samaan, N.; Karmouch, A. Towards autonomic network management: An analysis of current and future research directions. *IEEE Commun. Surv. Tutor.* **2009**, *11*, 22–36.
2. Mitchell, M. *Complexity: A Guided Tour*; Oxford University Press, Inc.: New York, NY, USA, 2009.
3. Toroczkai, Z. Complex networks the challenge of interaction topology. *Los Alamos Sci.* **2005**, *29*, 94–109.
4. Movahedi, Z.; Ayari, M.; Langar, R.; Pujolle, G. A survey of autonomic network architectures and evaluation criteria. *IEEE Commun. Surv. Tutor.* **2012**, *14*, 464–490.
5. International Business Machines Corporation. *An Architectural Blueprint for Autonomic Computing*; IBM Corporation: Hawthorne, NY, USA, 2005.
6. Kalton, M.C.; Mobus, G.E. *Principles of Systems Science. Understanding Complex Systems*; Springer: New York, NY, USA, 2015.
7. Sayama, H. *Introduction to the Modeling and Analysis of Complex Systems*; Milne Library: Geneseo, NY, USA, 2015.
8. Newman, M. *Networks: An Introduction*; Oxford University Press, Inc.: New York, NY, USA, 2010.
9. Freeman, L.C. Centrality in social networks conceptual clarification. *Soc. Netw.* **1978**, *1*, 215–239.
10. Pavlopoulos, G.A.; Secrier, M.; Moschopoulos, C.N.; Soldatos, T.G.; Kossida, S.; Aerts, J.; Schneider, R.; Bagos, P.G. Using graph theory to analyze biological networks. *BioData Min.* **2011**, *4*, doi:10.1186/1756-0381-4-10.
11. Lada, A.A.; Rajan, M.L.; Huberman, B.A. Local Search in Unstructured Networks. In *Handbook of Graphs and Networks*; Wiley-VCH: Weinheim, Germany, 2003.
12. Gladwell, M. *The Tipping Point: How Little Things Can Make a Big Difference*; Little Brown: Boston, MA, USA, 2000.
13. Lempert, R. Agent-based modeling as organizational and public policy simulators. *Proc. Natl. Acad. Sci. USA* **2002**, *99* (Suppl. S3), 7195–7196.
14. *Introduction to Cisco IOS NetFlow—a technical overview*. Available online: [http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6555/ps6601/prod\\_white\\_paper0900aecd80406232.html](http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6555/ps6601/prod_white_paper0900aecd80406232.html) (accessed on 1 September 2017).
15. Wooldridge, M. *An Introduction to MultiAgent Systems*; Wiley Publishing: Hoboken, NJ, USA, 2009.

16. Barabási, A.L. *Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life*; Basic Books: New York, NY, USA, 2014.
17. Barabási, A.L. *Network Science*; Cambridge University Press: Cambridge, UK, 2017.
18. Weiler, R.; Engelbrecht, J.; Cadmus, J. The New Sciences of Networks & Complexity: A Short Introduction. *Cadmus* **2013**, *2*, 131–141.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).