Architectural Design of External Autonomic Control For Implementing Reconfigurable Autonomic Networks

[*]Jyoti Bansal, [**]Mukesh Sharma
[*]*Student(M. Tech.), The Technological Institute of Textile & Sciences*
[**] *Associate Professor, The Technological Institute of Textile & Sciences*
[[*]jyoti.412,[**]drmukeshji]@gmail.com

Abstract

Autonomic networking is a concept to create self-managing networks to overcome the rapidly growing complexity of the networking and to enable further growth. Self healing is an important property of autonomic systems and for this purpose the network needs to be self-aware and self-managing in real time. Thus the main challenge lies in implementing the real time learning in the network and real time as well as synchronized communication among the network devices. This paper provides an architectural design of an external autonomic control system for implementing reconfigurable autonomic network using recurrent neural network for adaptive learning and timestamped messages for synchronization.

Key Words: Autonomic network, Reconfigurable, Neural Network, Recurrent.

1. Introduction

An autonomic network is one which exhibit self-* properties. These properties includes self-awareness, self-configuration, self-optimization, self-healing and self-protection. Thus an autonomic network can take care of itself and repair itself without any manual interference. Hence self-healing is one of the main characteristics of autonomic computing. It is concerned with ensuring effective recovery when a fault occurs. This means that any suggested solution must aim to successfully identify the fault and then repair it.

There are two different approaches to be used in the self management of systems and specifically with self-healing aspects; firstly; the software engineering approach where integrated expert system rules are used to manage the system, and secondly the adaptive learning method with the aid of neural network techniques. In this paper a new architecture is devised which takes the fusion of both the techniques and devise an optimal solution.

Here a network of similar mobile devices is taken and an external autonomic control is used to manage and repair the network. That control can manage the whole network which comes into its sensor range. Thus we can manage a big network by implanting these controls at proper distances.

2. Architecture

This is a block diagram which can be used in mobile device network controller to make an autonomic reconfigurable network. The function of this controller includes

1. Sensing arrival or removal of any device within its range,
2. Finding if network is working correctly or not, finding the faults and repairing them,
3. Communicating the configuration changes to all devices under it,
4. Establishing communication with other controllers,
5. To serve as a back bone network for creating a huge autonomic network.

The block diagram includes sensors, database, simulator, comparator, recurrent neural network, external communication unit and a control unit to manage communication among these.

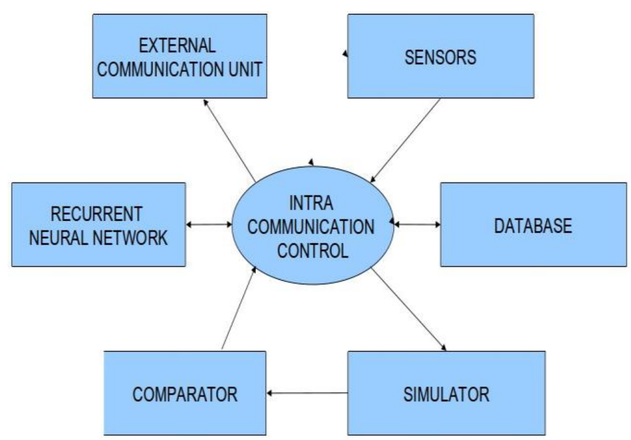

Figure 1. Block diagram of autonomous device

The functionality of each component is as follows:

2.1. Sensors

Sensors are devices which measures a physical quantity and converts it into the signals that can be read by an appropriate instrument. Thus here the different sensors can be combined to make a sensor unit that can sense any change in the current configuration. It will sense the arrival of and removal of any unit within the range of controller. This sensor unit on sensing any change will immediately send a notification to the control unit along with the details of the change. This unit will do this by interpreting the messages received by it. It will change the analog information in to the digital information with the help of A/D converters. Thus this unit will act as the input unit to whole system.

2.2. Database

Whatever input will be sent by the sensors will then be handed over to the database to make an entry into it. Here the database will also check if the same situation had arrived in the past as well. This can be done by matching all past entries with current entry and implementing the expert system rules. This will help as every entry will not only include the properties of network but also whether that state was stable or not and what were the measures taken. If any matching entry is found than it can tell if it can cause a problem and if the answer is yes it can also tell what measures were taken in the past to solve the problem. If no matching entry is found, that means such scenario has not happened in past thus it will document the new scenario in itself and wait for the solution that control will provide to it. Thus the controller will first try a software based solution.

2.3. Simulator

In case that no matching entry is found the inputs are directed to simulator which simulates the scenario and finds out the performance. The different simulators can be used according to different applications. But care must be taken that simulator must be strong enough to catch the actual scenario accurately so that it can provide a comparable performance.

2.4. Comparator

Now this performance is provided to comparator which checks if the performance is lower than the threshold value or higher than it. A comparator can be a simple one which checks only one parameter or a complex one which focuses on different parameters. But the comparator must be relevant to the application which uses it. The result will than be transferred to the control.

2.5. Recurrent neural network

Neural networks have been loosely analogized to biological systems such as those present in the brain or nervous system in which many neurons are interconnected by a complex network capable of carrying information. Neural networks may be simulated by digital systems such as computers, and are based on parallel architectures which are generally characterized by: simple processing elements, highly interconnected configuration of the elements, the elements exchanging many simple massages and preferably adaptive interconnection between the elements. Once a neural network is established, for example using hundreds or thousands of simple elements and interconnections, it can greatly benefit from a stored library of knowledge using learning rules. A neural network adapts by changing the weights according to interconnections by an amount proportional to the difference between a desired output and an actual output of some portion of the network. Such systems can be useful in situations where drawing on a knowledge base as well as learning in real-time can be used to make future decisions.

A neural network can be of feed forward type or feedback type. A feed forward network is one in which present inputs provide output. These have no memory. While feedback networks, also called as recurrent network are of a different type. A simple recurrent network has activation feedback which em-bodies short-term memory. A state layer is updated not only with the external input of the network but also with activation from the previous forward propagation. The feed-back is modified by a set of weights as to enable automatic adaptation through learning (e.g. back-propagation).

Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found.

Thus in this architecture when no entry is found in database and simulator shows a poor performance than the inputs are given to the neural network and a hardware based approach is taken. The neural network which is present in the controller is a recurrent neural network. Thus when inputs are got by it, it assign those to its node and do the computations. On initial computation it will not find the configurations appropriate for safe state, but due to feedback network, it will work till it reaches a solution which provides a safe state configuration of the network.

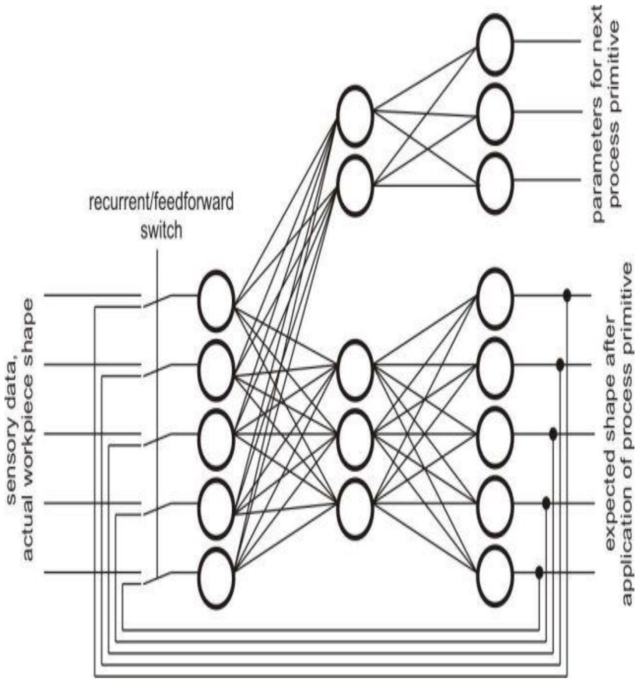

Figure 2. Architecture of recurrent neural network

Thus the recurrent network becomes the main point of controller which provides the autonomous properties to it. Now neural network will provide this solution to the controller which will communicate it to the devices present in the network. This configuration will remain saved in the recurrent network until it gets the new inputs thus controller can retrieve them again whenever it need. Controller will also save it in database for future use. When new inputs will be provided to the neural network, it will wash previous results. This is how recurrent neural network will work in this system.

2.6. External communication unit

This unit will take the commands from the control unit. Control unit, on the basis of inputs it got and the processing, will instruct the communication unit what changes has to be made in the network. The communication unit will then communicate this to the devices which will make changes in their configuration. The message to be communicated will be of following types:

1. "Safe" message: This message is sent to the devices to tell that the network is in safe state and thus no modification is needed.
2. "Modify" message: This message is sent to devices to tell that network needs some modification to come into stable state and to provide optimal performance. This message also contains the device by device instructions for modifications needed.
3. "Changes Noticed" message: Its a special message which control system broadcasts to neighboring controllers. This message is delivered when there is some large scale modification whose effects can not be mitigated within the network and they will effect other networks as well or when stable state can not be got even with the help of neural network, in that case a request is sent to others for help. This message contains the detailed network configuration.

2.7. Intra communication control

This unit can be said as the heart of this whole system. This is the unit which will make the whole system work by acting as a communicator/translator between them. Sensors will give input to it. This digital information will than be forwarded to the database. Database will notify it whether or not it has found a matching entry. In case a matching entry has been found it will transfer the to-do commands to communication unit based on database recommendations.

If no matching entry is found than it will send input to the simulator and after processing take the output from comparator. If comparator value is 1, that means performance is above threshold than it send the signal to communication unit to tell others that "All is well and no modification is needed." In this case communication unit will send "Safe" message to all devices within the range of controller.

If comparator results in 0, then it send the inputs to the recurrent neural network which finds the best possible solution and forward it to control unit. Control unit now give command to external communication unit to give modification messages. The contents of message is decided by this unit. The modification message includes the changes that are needed to be done. Alongside control unit also feed this all information into the database so that if similar situation arise in future then it can be dealt with.

In case a serious issue has arrived which can not be resolved within the local network then it commands communication unit to broadcast the "changes noticed" message to other controllers so that the spare resources of other networks can be used to find a solution of the current situation. Thus it can act as a life-saver for wider networks like internet.

The whole functioning of this block diagram can be depicted by this flow-chart in fig.3.

Figure 3. Flowchart

3. Implementation

The present concept involves the use of devices, which can be arranged in networks. The environment may be a physical environment of any size and dimension, such as a battlefield, building, or the human body, or a more abstract environment, such as a computer or a communication network. The controllers may also be aggregated to form a network of networks and provide scalability to the system.

The devices may be mobile, in the form of vehicles with sensors, or may be information agents. Two or more devices may communicate directly, or they may exchange information through the controller according to the specific use required in an application. This aspect can be enhanced by proper selection and formatting of the information that is communicated between the devices and controller. Proper protocols can be devised as well for successful completion of tasks. These measures will lower the need for bandwidth as well, thus proving quite efficient.

Communication may take place between various components of network over one of any suitable communication channels. Devices and controller may communicate over wireless radio frequency channels, optical communication method and other wireless and wire-connected communication networks, including any which are currently used for telephony or networking and other similar channels which may be developed or may become known to those skilled in the art of communication. The communication may further take advantage of auxiliary functions utilized in communication, such as data compression and encryption.

4. Usage

As mentioned above the above controller can be used in different scenarios ranging from a battlefield to the human body but this concept can be grossly useful in case of ad-hoc networks. This solution can be easily implemented in universities, at the places of seminars, in hotels etc. In fact a whole autonomous web can be designed using hierarchy of controllers.

In ad-hoc networks as the devices enter into the range of controller, it can get attached to its network, since the controller is stationary thus it will not be difficult to accommodate the controller on internet using traditional protocols while the controller can take care of mobile devices which is present in its range. This solution will provide an autonomous and speedy service to the devices at low band-width and also will not interfere in to the working of internet.

An another application where it can be used with equal efficiency is in traffic control. As in this scenario the moving vehicles can be viewed as the mobile devices and commands can be provided to them by controller according to the road conditions. It can help in eliminating road accidents to a great extent.

5. Conclusion

The present paper presents an architectural design of external autonomic control for implementing reconfigurable autonomic networks. The proposed architecture includes sensors, database system, Simulator, comparator, Recurrent neural network, external communication unit and intra communication control.

The Intra communication control provides the service as a communicator/translator among different system units. It receives input from sensor, provides input/ feedback to database, simulator, comparator and neural network, and based upon results guides external communication unit, “what message to broadcast”.

The functioning of purposed block diagram has been explained with the help of a flowchart. The application area and potential usages of this controller are elaborated in paper.

6. References

[1] Roy Sterritt, Dave Bustard, “Towards an Autonomic Computing Environment”. Proceedings of IEEE 1st International Workshop on Autonomic Computing Systems at 14th International conference on Database and Expert Systems Applications (DEXA'2003), Prague, Czech Republic September 1-5, 2003.

[2] M. Al-Zawi, D.Al-Jumeily, A.Hussain, A. Symons, A.Taleb-Bendiad ,”A Survey: Autonomic Computing”, Proceedings of the 1st International Conference on Digital communications and Computer Applications (DCCA2007), Jordan, pp 973-979, March 19-22, 2007.

[3] U.S. Provisional Application Serial No. 10/161,058 , filed Jun. 3,2002 , entitled RECONFIGURANLE AUTONOMOUS DEVICE NETWORKS.

[4] M. Mousa Al-Zawi, A.Hussain, D.Al-Jumeily, A.Taleb-Bendiab ,”Using Adaptive Neural Networks in Self-Healing Systems”, Proceedings of the 2nd International Conference on Developments in eSystems Engineering, 2009.

[5] David Garlan, and Bradley Schmerl, “Model-based Adaptation for Self-Healing Systems”, In ACM SIGSOFT Workshop on Self-Healing Systems (WOSS'02), Charleston, SC, 2002.

[6] David Garlan, An-Cheng et al., ”Rainbow: Architecture-Based Self-Adaptation with Reusable Infrastructure”, IEEE Computer Society Vol. 37(10), , pp 46-54, October,2004.

[7] Shang-Wen Cheng and Bridged Spitznagel, “Using Architectural Style as a Basis for System Self-Repair”, Proceedings of the 3rd Working IEEE/IFIP Conference on Software Architecture, Kluwer Academic Publishers, pp 45-59, 25-31 August 2002.

[8] S. Haykin, Neural Networks: A Comprehensive Foundation, New York Macmillan, 1994.

[9] T. Hagan, Demuth, and Beale, Neural Network Design, Canada, Thomson, 1996.