AORTA: Autonomic Network Control and Management System

Ali Tizghadam
School of Electrical and Computer Engineering
University of Toronto, Toronto, Canada
Email: ali.tizghadam@utoronto.ca

Alberto Leon-Garcia
School of Electrical and Computer Engineering
University of Toronto, Toronto, Canada
Email: alberto.leongarcia@utoronto.ca

***Abstract*—This paper reports on an autonomic network management architecture based on the concept of ”evolution”. A management methodology is developed which is relying on the ideas from evolutionary science, virtual networks, and autonomic networking. We argue that any communication network could be modeled as an evolved topology based on survivability and performance requirements. The evolution is in the direction of decreasing the chance of congestion and increasing the network robustness. We describe the architecture of our network management system in detail and tie it to the theory of evolution. We evaluate the ”betweenness centrality” of network topologies and build our robust routing algorithm to manage the transport of packets in the network based on it. This routing scheme is at the heart of out proposed network management system.**

***Index Terms*—Evolution, Robustness, Autonomic Networking, Graph-Theory, Virtual Networks, Routing Algorithms.**

I. Introduction

Traditional telecommunications service providers are undergoing a transition to a shared infrastructure in which multiple services and applications will be delivered by peer and server computers interconnected by IP networks. Automated service and network management are essential to creating and maintaining a flexible and agile service/application delivery infrastructure that also has much lower operations expense than existing systems.

In this paper we focus on the IP packet transport infrastructure and using the theory of evolution, we argue that the above requirements can be met by a self-management system based on autonomic computing and virtual network concepts. We provide the details of our view of management which is proposed as a whole self-organizing system called AORTA (AutonOmic netwoRk conTrol and mAnagement system). Our primary goal in the proposed architecture is to make the whole communication system ”robust” to possible changes in different parameters of the network due to uncertainties. There are three major type of changes affecting the performance of the network [1]: Network topology, Community of interest (active source-destination pairs), and Traffic Matrix. We call a system robust if it can resist against uncertainties which are the result of changes in topology, traffic or community of interest. To achieve the desired degree of robustness, we use Darwin’s theory to develop a management scheme to be able to survive under different circumstances and deliver desired performance as the secondary goal.

The rest of this paper is organized as follows. Our view of the network management is introduced in section II, followed by the details of our management system in section III. Section IV describes the routing scheme that is used in our management system is proposed. Simulation results and validation are provided in section V and the paper is concluded in section VI.

II. The Proposed Management Methodology

To build the conceptual idea of the management architecture, we are inspired by the concept of ”evolution” in nature. Evolutionary processes are good examples of self-organizing systems. Darwin’s seminal work on describing the long-term processes in life, and the theory of ”natural selection”, made us think about the management architecture as an evolutionary process that is being formed by natural selection on the way to ”survive” in regard to the environmental changes.

The main idea behind Darwin’s theory is to assign a ”survival value” to all the processes in the world. This value is a measure of resistance or robustness of a process or element to the changes in the nature. In other words, it shows how adaptable a system is to unwanted events. According to Darwin’s view, anything in this world is fighting to survive and all sort of changes happened in the life of species are to be able to survive in the world. On the other hand, from mathematical point of view, the processes in the life look like optimization problems. Darwin’s theory does not consider any ”final target” for the evolutionary changes in the nature, but one can see that the survival as the goal can lead to an implicit optimization problem.

We looked at the management as an evolutionary process in addition to one or more optimization problems. The first goal of any management system is to keep the system alive under unforeseen circumstances, which is the survivability feature. For our purpose, the main service is the ”packet transfer”. To have a safe packet transfer service, the first parameter to consider as the survival value is the connectivity of the graph. This gives us the first framework for the evolutionary process of the management system. Any kind of communication network is evolved or should evolve in a way to guarantee the connectivity to the extent possible.

The evolution could be real or ”virtual”. In a designed network that we are dealing with in communications, the real evolution is not happening before the appearance of the network, but as any designed network is built to achieve some specific goals and has some initial configuration to meet those goals, one can interpret the design process as the first step of the evolution which has happened some time in the past and its present form is a result of that evolutionary process (virtual evolution), and it will continue to evolve in real time to achieve its upcoming goals.

For the final target in a management scheme, one has an optimal situation and tries to build the whole management system to converge to this target by converting the management process to an optimization problem. This optimization is namely dealing with the real-time efficiency or performance of the whole network (short-term goal), while the survival value of the network should be maintained and improved constantly (the long-term loop). The main idea behind AORTA, is to design appropriate self-organizing management elements in order to build long-term and short-term control loops and to achieve the robustness and performance simultaneously. In fact robustness as we defined it in section I is considered as the most important factor for decision making in AORTA. This generalizes the idea of survivability to the robustness of the system against aforementioned system changes. In AORTA, the short-term and long-term control loops are designed to achieve robustness first and goodness or efficiency next.

We have used IBM's "Autonomic Computing" [2] framework as a reference model and built our "Autonomic Networking" architecture based on the above mentioned philosophy and the techniques adopted from autonomic computing (AC).

While AC is useful to describe the evolution of a network to the stable position and configuration, there is still a problem with large networks such as Internet. Having the information of the Internet topology and its other attributes is next to impossible. Further, Internet is grown out of efficiency. The performance is sacrificed to have flexibility, scalability and robustness. In a holistic view, this makes the management of large networks a challenging task that requires human resources in many situations. That can describe why there are an abundance of overlay networks on top of the real physical networks. Overlays and virtual networks are smaller in size and can be easily managed. It is possible to offer QoS-based services on top of overlay networks that improve the best-effort nature of the Internet.

In this research, we have used the concept of virtual networks (VN) and overlays as another tool to build AORTA.

A. Autonomic IP Transport Management

Several factors make the design of self-configuring networks under the current management structure challenging. Among them are: change propagation, configuration policy, and resource allocation.

Our proposed management system addresses the network automation problem by proposing an autonomic management system capable of self-management that can evolve to the best possible configuration to achieve robustness without human intervention. The overall conceptual architecture of the proposed autonomic management system is shown in Fig. 1. We introduce a hierarchical arrangement of two parts, a long-term (slow) and a short-term (fast) part. The short-term part reacts to the network changes in real-time and the 'slow' part takes actions on a longer time-horizon.

The long-term part develops the virtual evolution or real evolution part based on an initial knowledge-base that consists of the business policy as well as empirical results from previous experience about customer demand, and price elasticity. The network plan includes the translation of business policy into policies that are meaningful to the short-term part for use in the handling of customer requests. The plan also includes the synthesis of the SLA templates that will be offered to customers taking into account forecasted demand, resource requirements and price elasticity. All the planning parts are aiming at providing robustness with implementing the long loop to make constant evolution as well as immediate performance with short-term loop. Finally, the plan includes pre-partitioning of network resources to facilitate the handling of customer requests by the short-term part. For example, the plan may include pre-provisioned routes per each (ingress, egress) pair or more generally pre-provisioned VNs.

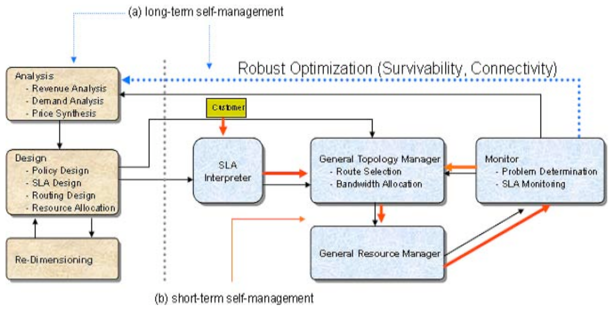


Fig. 1. Conceptual Architecture of AORTA

Because of the autonomic nature of the overall system, the short-term part needs to interact with the 'slow' part in carrying out self-healing, self-optimizing, and self-configuring functions (bringing robustness as the final result). This mainly occurs when un-predictable events take place, such as surges in demand or major failures in the network. In these situations the short-term part will respond to provide a fast real-time cure, but will act to provide a long-lasting cure by making a request for re-dimensioning to the 'slow' part. The interaction between slow and fast parts of the system could also be the result of detecting inefficiency in resource usage in the fast part. In this case a request for re-dimensioning is sent to the slow part to re-optimize the allocation of resources.

The short-term or 'fast' part of the system consists of four major building blocks that are driven by customer requests. As shown in Fig. 1, the 'SLA Interpreter' block is responsible for negotiating the SLA with the customer and for converting the SLA contract to an appropriate form understandable by a 'General Topology Manager' block. This latter block plans the route (or VN) and resource allocation based on the converted SLA, the already allocated resources, and current network demands. The results are delivered to the 'General Resource Manager' block which executes orders that allocate the appropriate amount of resources. The 'Monitoring' block continuously monitors the system to identify possible problems (e.g., SLA violations, failure alarms and so on). After filtering, it sends information to the 'General Topology Manager' to develop an immediate cure, and in parallel it may send a message to the 'Analysis' block of the 'slow' part to report an unpredictable event. If appropriate, the 'Analyze' block may initiate new network planning. In the next section we provide more details about our proposed system.

III. VN-based Autonomic Network Resource Management

A VN is the set of network resources that are dedicated in routers and transmission links to one such customer and can be viewed as a subset of the overall network ([3], [4]). The partitioning of network resources is made possible by abstracting the set of physical network resources into a set of Virtual Network Resources (VNR). The value of a VNR is specified by a resource quantity (e.g., equivalent bandwidth, protection bandwidth). The allocation of resources to VNs can then be specified in terms of these quantities.

In AORTA, any managed element is a VN and the management system tries to provide enough resources (according to the contracted SLA) to the clients in a self-organizing way. We now present a VN-based network resource management architecture which is compliant with the autonomic architecture illustrated in Fig. 1. The proposed system attempts to strike a balance between the flexibility of managing VNs separately and the complexity inherent in requiring a network element to interact with multiple managers and is composed of three different autonomic managers: autonomic system re-dimensioning (ASD), autonomic VN manager (AVNM), and autonomic resource manager (ARM), as illustrated in Fig. 2. By monitoring their target managed objects, the autonomic managers diagnose the object status. When a problem is determined, first, the manager tries to localize the problem and repair it by itself. Depending on the nature of the problem this will provide us with self-healing (in case of an error) or self-organizing (re-arrange the resources to attain optimum utilization) or self-configuring (adapt to the other changes). If it cannot handle the problem by itself, the high-level autonomic manager is involved (slow loop). The ASD is an off-line process if we are at the startup which accounts for virtual evolution or responsible for the long-term loop and is in charge of the business level network analysis and design in addition to the possible re-dimensioning on a long-term scale as in Fig. 1. The role of AVNM is to manage and control the VNs according to the customer SLAs and the service provider's policy. The role of the ARM, on the other hand, is to manage and control the VNRs. A single ARM is responsible for the control of a group of VNRs. Fig. 3 illustrates the detailed functional building blocks of the AVNM and ARM. The proposed architecture provided necessary blocks to build the control and management system, but one needs to develop appropriate algorithms for the fast and slow loops (for the long loop the algorithm is in fact an evolutionary process). In the following section we try to shed more light on it. To this end, we investigate the "general topology manager" block from AORTA (Fig. 1) and develop a fast algorithm for robust assignment of traffic demands to appropriate paths of the network to maximize the robustness as we defined it.

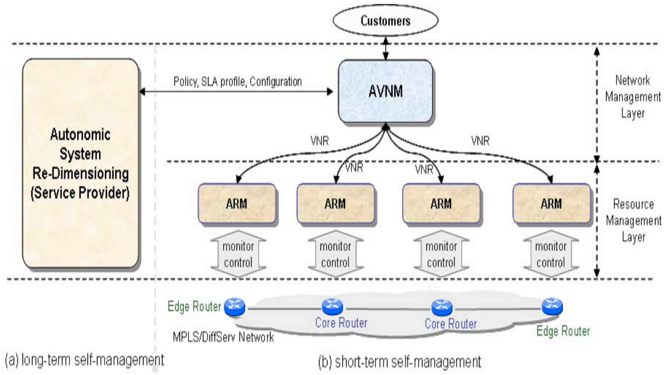


Fig. 2. Autonomic VN Management System Architecture

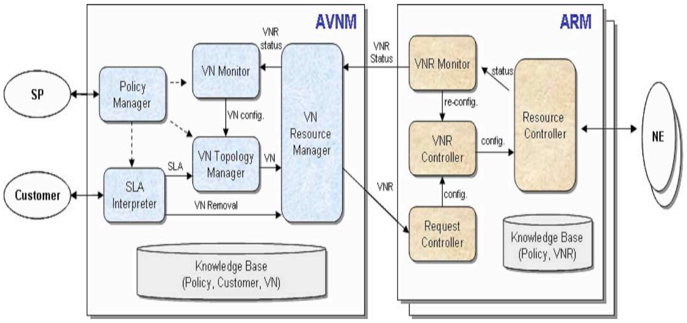


Fig. 3. Detailed Architecture of AVNM and ARM

IV. Design of a robust routing scheme

In this section we briefly outline an approach to the design of a robust routing algorithm which is realizing the "general VN topology manager" in AORTA. In [1] we proposed Path Criticality Routing (PCR) as a robust routing algorithm. In this paper we discuss a probabilistic version of PCR.

A. Path Criticality Routing (PCR)

In order to have a robust routing plan we need some metrics to robustness. We use concepts of graph theory to obtain useful metrics for our purpose. "Betweenness" of a link l for source destination pair s-d (b_{sld}) in probabilistic view is defined as the average number of times that a random walk starting at source s goes through link l before reaching at destination node d [5]. The total betweenness of link l would be the sum of all betweennesses for different source-destination pairs which are members of the community of interest ($b_l = \sum_s \sum_d b_{sld}$). We chose to have the link betweenness over "available capacity" as our main metric, and called it Link Criticality Index (LCI). LCI captures the effects that we would like to quantify. One can easily see that betweenness centrality captures the effect of load. The higher the link betweenness, the more the chance of congestion. This suggests link betweenness as the metric to quantify the criticality of that link. On the other hand, the available capacity has inverse effect on the congestion.

$$I(x) = 0 \quad if \quad x > 0 \quad otherwise \;\; 0$$

$$LCI(l) = \frac{b_l}{c_r(l)} \times \frac{1}{I(c_r(l) - \gamma_l)}$$

In above equations $I(x)$ is the indicator function, $LCI(l)$ is the total criticality of the link l, $c_r(l)$ is the available capacity of link l, and $\gamma(l)$ is the present demand on link l. The indicator function is added in the denominator to guarantee that if the demand is more than the available capacity of the link, the demand is not accepted in this link (the link criticality would be effectively infinite in this situation). Definition of the link criticality is clearly showing that the criticality of a link is increasing if more load is carried through this link.

When the LCI of all the links are known, the criticality of a path, Path Criticality Index (PCI), will be the maximum of the LCI of the links belonging to the path.

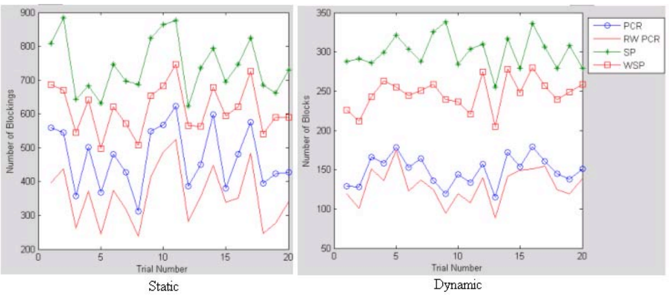
B. Random-Walk PCR Algorithm (RW-PCR)

The basic idea of our routing algorithm is to accommodate new requests for connections along the paths with low PCI. We label each and every link of the graph with its LCI as the cost (note that this cost is different than the weight set w_{ij}) and use Dijkstra's algorithm to obtain the shortest path(s) from a source s to a destination d using the assigned cost for the links. When a demand for source-destination pair $s-d$ arrives, the shortest path obtained in this way would be considered as a nominate to be assigned to the demand. A simple call-admission control is applied here by considering a threshold tr for the criticality of the path. If the PCI is more than this threshold, then the flow would be considered too risky for the network and be rejected (blocked), otherwise the path is used as the route and the demand flow is assigned to this path. The available bandwidth of all the links on this path is updated and the LCI's are also modified accordingly.

V. Evaluation

In order to investigate the effectiveness of our revised PCR algorithm, we ran a set of simulations on a network with 22 nodes and 45 full-duplex links. The link bandwidths were chosen to be 100 units. In the first experiment the requests for bandwidth arrive with Poisson distribution and stay for ever (no departures). In our tests the bandwidth requests for paths are taken to be uniformly distributed between 1 to 3 units. In Fig. 4 we show the number of rejected calls for the static case and we compare the performance to that of original PCR, RW-PCR, shortest path (SP), and widest shortest path (WSP). The test is performed 20 times and each time with 2000 path requests. We measured the number of blocked requests.

In another experiment we examined the behavior of the algorithms in the presence of dynamic traffic. Fig. 4 also shows the number of the path requests rejected in 20 experiments for the following dynamic scenario. Path requests arrive between each source-destination point (which is chosen at random) according to a Poisson process with an average rate λ, and the holding times are exponentially distributed with mean μ. We assume $\frac{\lambda}{\mu} = 1800$ in this experiment. We generate 7000 requests and measure the rejections or blocking for each one of the algorithms.



Static

Dynamic

Fig. 4. Applying PCR, RW-PCR, SP, and WSP to the Network

VI. Conclusion

In this paper We have developed a management architecture, what we call it AORTA, to cope with the management problems in a self-organized way. We gave the architectural details of the ingredients of AORTA and have shown that the conceptual architecture of AORTA contains a long-loop to account for the survivability and robustness of the network and a fast loop to watch for performance optimization. In second part we proposed our probabilistic routing algorithm which is the heart of the management system. The goal of this routing algorithm is to robustly manage the demands from any source to any destination and have the minimum service interruption in the core.

There are different venues for further investigation. first of all, we will explore the topologies which are providing the best connectivity while keeping the cost lower than a maximum. This is necessary to complete our picture of the long-loop in AORTA. The system is survivable only if the connectivity maintained in all situations. Another future work is to look at the capacity design problem from the evolutionary theory perspective whcih is our ultimate goal. To have a robust system we need to find the least biased network setting. In other words we need to maximize the entropy. Maximizing entropy gives the appropriate capacity of the network elements to maintain the highest degree of robustness.

References

[1] A. Tizghadam and A. Leon-Garcia. A robust routing plan to optimize throughput in core networks. *ITC20, Elsvier*, pages 117–128, 2007.

[2] IBM. An architectural blueprint for autonomic computing. April 2003.

[3] A. Leon-Garcia and L. Mason. irtual network resource management for next-generation networks. *IEEE Communications Magazine*, 41(7):102–109, July 2003.

[4] A. Jun and et al. Virtual network resources management: A divide-and-conquer approach for the control of future networks. *IEEE Globecom, Sydney Australia*, 2:1065–1070, November 1998.

[5] M. Newman. A measure of betweenness centrality based on random walks. *arXiv cond-mat/0309045.*, 2003.