International Journal of Innovative Computing, Information and Control
Volume **10**, Number **6**, December **2014**
ICIC International ©2014 ISSN 1349-4198
pp. 2143–2156

SCALABILITY ISSUES IN NETWORK SELF-MANAGEMENT: A PARTITIONING APPROACH TOWARDS SCALABLE AUTONOMIC MANAGEMENT COMPUTATION

Romildo Martins da Silva Bezerra[1] and Joberto Sergio Barbosa Martins[2]

[1]Computer Science Department
Federal Institute of Bahia
Rua Emidio dos Santos s/n Barbalho 40301-015, Salvador, Bahia, Brazil
romildo@ifba.edu.br

[2]Engineering and Architecture Department
Salvador University
Alameda das Espatodias 915, Caminho das Arvores, CEP 41820-460, Salvador, Bahia, Brazil
joberto@unifacs.br

Received September 2013; revised January 2014

Abstract. *Network characteristics such as service diversity, large network size and technology heterogeneity are currently inducing network management to become a highly complex task for administrators. In this context, Autonomic Management Systems (AM Ss) are being investigated as possible approaches capable of dealing with this inherent complexity. In effect, it is expected that new autonomic systems and solutions could, among other functions, both configure and optimize network resources while keeping network performance characteristics in a scalable way. This paper addresses, fundamentally, the scalability problem faced by current AMS implementations. The proposed solution consists in a generic partitioning method that improves AMS capability to execute on-the-fly autonomic management computation and, as such, establishes a viable path towards on-the-fly autonomic management systems. By on-the-fly autonomic management systems it is meant an AMS that computes autonomic solutions in more realistic time frame. It is argued that the network partitioning strategy proposed can be applied to manage the management computation scalability problem associated with complex networks while considering a practical and realistic set of requirements and parameters specified by the network administrator such as QoS parameters (SLAs) and execution time. This paper addresses initially the set of basic network management require-ments considered by the partitioning method, details the partitioning method and, finally, evaluates the scalability issues considering a self-managed framework (AMS) in diverse scalable scenarios. Results points to the feasibility of the partitioning method in scenarios with different traffic matrix allocation and, beyond that, under new topologies resulting from failures.*

Keywords: Autonomic systems, Network management, Scalability, Performability

1. **Introduction and Motivation.** Autonomic computing principles can be used in network management contexts since these systems are typically complex [2, 5], unpredictable and large-scale [8, 12]. One of the goals of applying the autonomic computing approach in network management is to efficiently achieve a more effective management solution and, at the same time, to reduce human intervention [7, 12] since the inherent network management complexity makes human intervention an important failure point [14].

AMSs (Autonomic Management Systems), in general, look at the provision of full-range (complete) management solutions, like new network states and/or decision-support management information for network managers. In this context, AMSs with autonomic

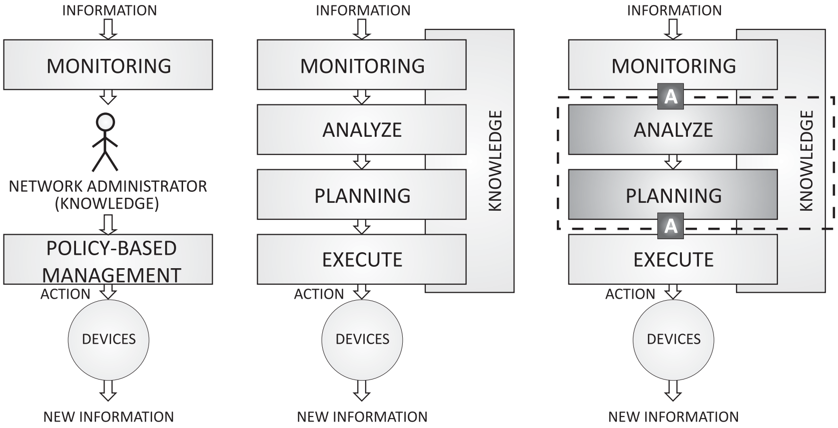


Figure 1. Administrator expertise replacement in analysis and planning phases

characteristics are considered relevant in diverse application scenarios like network reconfiguration, network optimization and network self-healing, among other possibilities. A fundamental challenge faced by AMSs is to compute the required management solution (either reconfiguration, optimization or self-healing ones) with an acceptable execution time for real networks. The size of the network (number of nodes and topology) and the volume of management parameters involved in the computation represent an important problem of scalability for current AMSs aiming to behave in an autonomous way. In this same context, the ideal AMS would be the one that would be able to scale in relation to network size and management complexity while keeping Performability[1].

The basic motivation of the proposed network partitioning solution presented in this paper is, in brief, to investigate alternatives to improve the AMS capability to scale in relation to network size and management complexity in the specific scenario of network reconfiguration with Quality of Service (QoS) requirements.

In the Quality of Service (QoS) scenario, in brief, it is required to administer parameters such as delay, loss, bandwidth and jitter and to define the paths for network flows (traffic matrix allocation). The computational complexity for path definition (routing) and traffic allocation (flow allocation) is directly associated with the number of nodes and links existing in the network (topology), making resource management intrinsically more complex in large domains. A second relevant dependence that exists for AMS to compute a management solution like reconfiguration is the maintenance of Quality of Service agreements (SLAs: Service Level Agreements) defined by applications. As such, an AMS considering a QoS implementation scenario has a two fold scalability implementation problem. It is composed by the computation of a new network flow allocation solution (traffic matrix allocation) while preserving QoS guarantees (SLAs and user profiles) defined by the applications and/or managers.

Autonomic Management Systems (AMSs) providing reconfiguration solutions while preserving Quality of Service (QoS) guarantees have to do with a number of factors. Parameters involved in the computation includes the number of flows (streams), the number of equipment (routers, switches, others) in the network topology, the traffic distribution (traffic matrix), the dynamicity of the traffic matrix (flow change, flow insertion, flow cancelation, flow preemption), among other possibilities. The motivation of the partitioning method and challenge involved is then to compute new network settings in a specific QoS

---

[1] Performability is the property of a system such that it delivers the performance required by the service specification, as described by Quality of Service (QoS) measures and SLA (Service Level Agreements) [21].

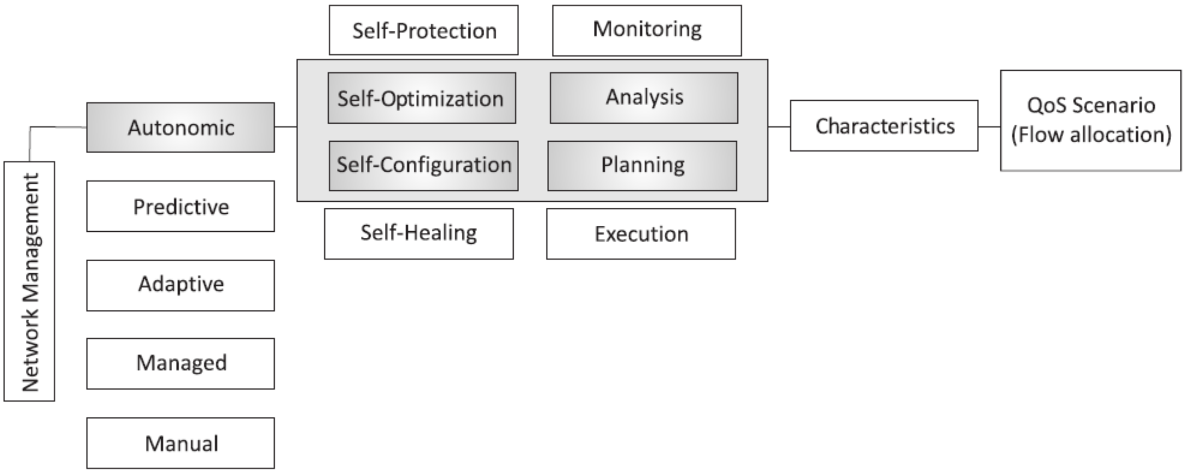


Figure 2. Characteristics overview

management scenario and, concomitantly, to support on-the-fly solutions which could be acceptable in real networks. By acceptable, we mean AMS ability to react faster than a human being and more efficiently than existing management systems.

The strategy developed is implemented in analysis and planning phases of an AMS to improve a dynamic and on-the-fly approach to the decision-making process towards the effective administrator expertise replacement (Figure 1). In general, when considering the management cycle (MAPE), the focus is placed on analysis and planning steps strongly coupled with the network knowledge involved. This Characteristics Overview is presented in Figure 2.

In practical terms, the autonomy will be used in the allocation of flows – in large areas – targeting the tradeoff between time allocation (Performability) and quality of solution found. A detailed overview can be seen in Figure 2.

2. **AMS Scalability and Approaches.** Fundamentally, the problem of scalability should be considered by current AMSs (Autonomic Management Solutions). Scalability, in this context, means the AMS ability to operate in networks with different quantities of devices and/or traffic volume. The variability of traffic volume and number of equipment managed can be an aspect considered by an AMS dynamically for a specific network.

The basic issue is to provide an autonomic solution in an acceptable response time and, at the same time, to maximize the Quality of Service currently offered by the network. This dual AMS objective is an actual research challenge addressed by network management autonomy with distinct approaches.

[10, 11] point to a solution for reducing network utilization considerably by using a Distributed Interactive Simulation Applications, called Dead Reckoning mechanism. However, this technique often ignores available contextual information that might be influential to the state of an entity, sacrificing remote predictive accuracy in favor of low computational complexity. A novel extension of Dead Reckoning based on ANFIS systems is also suggested to increase the network availability and to keep the required QoS for applications.

[15] is applied a probabilistic graphical model to characterize statistical dependence between physical layer failures and the network layer traffic. The objective is to analyze the scalability for large networks under different topologies. The focus in this research is on the self-scalability for regular topologies under uniform deterministic traffic with independent and dependent link failures, with and without protection.

3. A Network Partitioning Strategy towards AMS Scalability.

As previously discussed, large networks are highly complex to manage, mainly due to the conjunction of two factors: the dynamic nature of routing (traffic matrix allocation, topology changes, and others) and the size of such networks [16]. Autonomic Management Systems (AMSs) should be able to work with networks having different cardinalities (nodes and links) considering topological or traffic constraints.

In summary, the proposed partitioning strategy consists in performing some sorts of network segmentation (divide and conquer principle) towards a more efficient computation of autonomic network management solutions for large domains. In other words, a partitioning method is applied to the target network with a defined methodology and sequence of steps that, fundamentally, computes new network state (settings) in accordance with a set of overall requirements defined by the administrator (manager).

A network partitioning strategy is proposed to deal with the AMS scalability computation problem. In order to facilitate the partitioning strategy presentation, we consider a specific situation in an AMS framework (Figure 1). We will assume that there is an autonomic management framework and, in this framework, a decision plane is the entity that computes new network settings with autonomic characteristics [2]. As such, the partitioning strategy is effectively allocated (implemented) in the decision plane and we will consider, in order to facilitate the discussion, this instance as a Network Partitioning Computing Engine (NPCE). In terms of the network partitioning strategy, the NPCE initially receives a set of basic parameters (inputs): the network topology, the traffic matrix and the network requirements (response time and QoS). The NPCE (partitioning strategy) essentially computes the best tradeoff between the expected response time for finding a new network state and the Quality of Service requirements defined (Figure 3). Once the best partitioning option is found by the NPCE, the traffic matrix is reallocated in order to redistribute the flows over the network, configuring a new network state.

The essential aspect of the partitioning strategy is, firstly, that the traffic distribution for the network will have two new traffic components: the intra-domain and the inter-domain traffic. The intra-domain traffic can be understood as the amount of traffic (flows) which remains inside the boundaries of a newly computed network partition (new network segmented domain). The inter-domain traffic corresponds to the amount of traffic (flows) which will necessarily cross new partition boundaries in order to preserve its previous origin-to-destination flow allocation. The second important aspect of the partitioning strategy is that the computation of traffic matrix reallocation can now potentially be executed in parallel for all intra-domain traffic and, beyond that, this computation will require less computational resources since the number of parameters involved is potentially reduced. As such, there is a potential benefit in order to achieve a reduction in the computation time required, as previously indicated, by the AMSs.

The essential aspects of the partitioning strategy are, in brief, the potential capability to reduce the computation time for the definition of a new network state based on the ability to segment the network and its actual traffic among partitions and to reallocate the intra-domain and inter-domain traffic. This aspect will be considered in the following sections firstly in terms of its analytical concepts and subsequently showing additional details of its effective implementation.

3.1. Network partitioning basic concepts and discussion.

Let V be a set of vertices (network nodes) and E a set of edges (network links). A computer network can be defined as an undirected connected graph $G = (V, E)$ such that $E \subseteq |V^2|$.

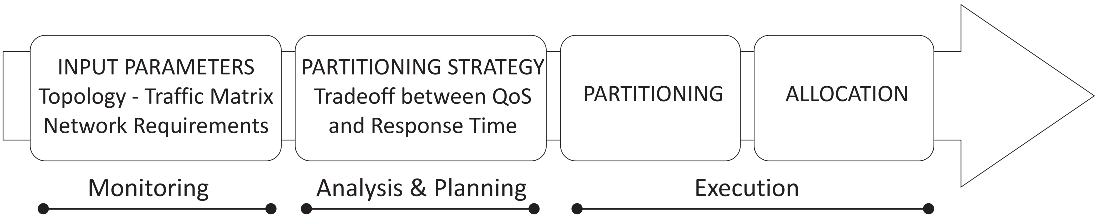


FIGURE 3. Network partitioning strategy overview

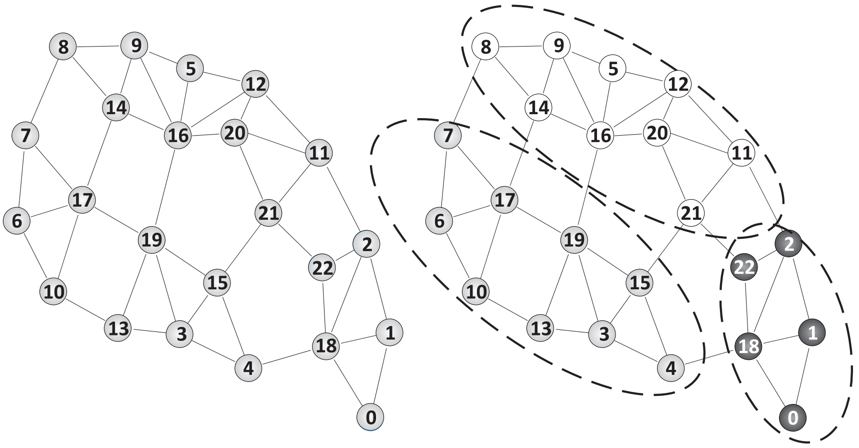


FIGURE 4. Network with 23 nodes and partitioning alternative

Based on that, the elements of E can be seen as tuples with two element of V^2. In this paper we use the following notations:

- The order of a graph G is $|V|$ (vertex number) and $|V| = n$. In Figure 4, $n = 23$;
- The number of edges is denoted by $||G||$;
- The vertex degree is the number of edges that connects to it; and
- The graph density is the ratio between the edges number and the possible edges number ($||G||/||K_n||$).

A simple partitioning example of network ($n = 23$) with 3 partitions ($d = 3$) and resulting interdomain traffic is exhibited in Figure 4.

3.2. **The partitioning algorithm and other implementation aspects of the partitioning strategy.** The proposed partitioning strategy uses a partitioning algorithm based on a network density parameter (Section 3.3) and divides a graph (network topology) into disjoint domains (partitions) so that flow/traffic allocation can be computed concomitantly to all partitions and independently by the adopted allocation algorithm. The network traffic resulting from the partitioning is either intra-domain (source and destination nodes belonging to the same partition) or inter-domain (source and destination nodes not belonging to the same partition). Figure 4 illustrates a network represented by a graph with $n = (A, B, C, D, E, F, G, H, I, K, J, L, M, N, O, P, Q)$ nodes and six traffic generators/receivers located in nodes A, B, C, D, E and F. The input traffic matrix is represented by a 6×6 square matrix (six traffic generators/receivers). Two possible partitionings A and B are illustrated, where the number of domains/partitions are respectively, two and three ($A = p_{A1}, p_{A2}$ and $B = p_{B1}, p_{B2}, p_{B3}$), represented as follows:

- $p_{A1} = (A, B, G, H, M)$
- $p_{A2} = (C, D, E, F, I, J, L, N, O, P, Q)$

---

[2] Edges of type (e_i, e_i) will not be considered and (e_i, e_j) is equal to (e_j, e_i).

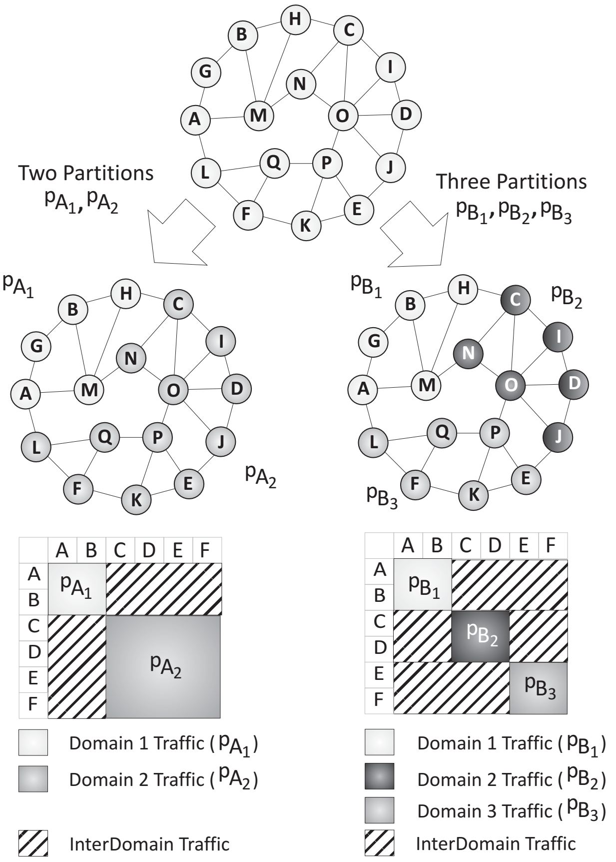


Figure 5. Partitioning and intra/inter-domains relationship

- $p_{B1} = (A, B, G, H, M)$
- $p_{B2} = (C, D, I, J, O)$
- $p_{B3} = (E, F, K, L, P, Q)$

This simple example illustrates that the intra-domain and inter-domain traffic for A and B solutions are different as perceived in Figure 5.

As such, the partitioning strategy implemented by the NPCE (Network Partitioning Computing Engine) has to go beyond the straightforward utilization of a network segmentation algorithm and, effectively, is structured in a sequence of steps as follows:

1. The NPCE runs initially the partitioning algorithm (Section 3.3) generating a set of possible network partitions ($d \geq 2$) of different sizes, granularity and inter-domain and intra-domain traffic patterns.
2. The NPCE evaluates the allocation algorithm cost for the input traffic matrix (current network traffic profile) for each obtained partition independently (Section 3.4). The inter-domain traffic cost is calculated without partitioning (global view).
3. The NPCE evaluates the best tradeoff between response time and Quality of Service (QoS) requirements as defined by the network administrator (Section 3.5). The result expected, as previously argued, is the reduction of the overall computational time required for a new network state definition (configuration) while preserving Quality of Service requirements. The following sections describe each specific step associated with the NPCE partitioning implementation (NPCE operation phases).

3.3. The partitioning algorithm – NPCE operation – Phase 01.

The input parameter for the partitioning algorithm is a topology graph G and, in general, finding an exact solution to a graph partition is considered an NP-complete problem, making it difficult for huge network structures. Many areas such as social networks, web graph, ecosystems and others, use graph partitioning algorithms to find common properties or relationships between vertices according with their specific requirements.

Considering the network management scenario adopted (Quality of Service), the overall partitioning principle is, basically, to divide network nodes (G graph) into subgraphs with the best possible density. When looking for higher density subgraphs, one obtains a set of vertices where the relationship between them is stronger, i.e., the number of edges in the subgraph is substantial. Being so, this corresponds to a computer network partition/domain with a greater number of paths between a set of vertices with, potentially, a higher number of allocable routing paths (LSPs) in this partition/domain. It is important to mention that when edges (links) do not belong to any sub-domain they correspond to links for inter-domain communication.

The edgebetweenness [18] is the algorithm adopted for network partitioning. A number of algorithms that find communities from large network domains exist in the literature and were evaluated considering the NPCE partitioning strategy [2]: Edgebetweenness [18]; Eigenvector [17]; Fastgreedy [3]; and tem Walktrap [19].

The criteria adopted considering the partitioning strategy were execution time, density and standard derivation of sub-domains size [2]. The edgebetweenness algorithm showed the best behavior and was the one chosen as the basic algorithms adopted to generate subgraphs with high average density for NPCE operation (Phase 01).

Thus, in this NPCE phase a set of partitions with different cardinalities is generated creating a set of partition groups derived from the basic topology. Each of these partition groups will be identified in the following text as a new network Partitioned Topology (NPT).

The process starts with an NPT with two partitions ($p = 2$) and proceeds until finding an NPT with internal partition(s) with less than two nodes ($p = p_{\max}$). This restriction is adopted in order to assure that there will be always intra-domain traffic for all partitions belonging to all NPTs.

3.4. **Traffic allocation and cost evaluation – NPCE operation – Phase 02.** The input parameters for the traffic allocation and cost evaluation phase are:

- The set of NPTs having network partitions with the highest possible density determined by the partitioning algorithm; and
- The actual traffic matrix assigned (allocated) for the current network graph topology (G the network topology before the partitioning).

In brief, the main objectives of this operation phase are:

- Firstly, to identify (map) the set of flows currently allocated to the network which are confined to a specific partition/domain (intra-domain traffic) for all NPTs;
- Secondly, to compute the complexity cost associated with the available routing protocols used for flow allocation and, optionally, choose autonomically the most efficient one in relation to the various NPTs computed; and
- Finally, to effectively compute paths (routing) costs considering flows allocation and identify relative costs for the allocation algorithm chosen considering intra-domain and inter-domain traffic for all NPTs.

In the first step, per partition/domain traffic mapping is executed. In effect, for each partition p ($2 < p < p_{\max}$) and for all NPTs the traffic percentage for each partition/domain is calculated as a function of the total traffic. Following that, the number of flows per domain and inter-domain traffic is computed for each partition/domain.

The path computation for flows requires a routing algorithm for flow allocation. In effect, any routing algorithm can be chosen in this step (Dijkstra [6], Bellman [1], Jonhson [13], and others) since the partitioning strategy, can be applied consistently and independently by the routing algorithm chosen by the network administrator.

Optionally, this step may evaluate the best option among any set of routing algorithms available. In brief, this step can evaluate the complexity[3] cost resulting from using a specific algorithm for the set of NPTs and corresponding partitions/domains. In practical terms this scenario could be helpful when an administrator is considering among a set of routing protocols for a new network state which is being computed. In terms of providing an effective implementation for the NPCE, the option adopted was the Dijkstra link state shortest path algorithm.

The last step of phase two computes cost for path allocation considering all flows (traffic) per partition/domain for all NPTs. It also determines the relative allocation cost (cost with partitioning/cost without partitioning) for the previous path computation considering the set of partitions/domains in NPTs. In effect, δ represents the sum of the intra-domain (C_{intra}) and inter-domain (C_{inter}) allocation costs for any partition p ($2 < p < n$) divided by the cost for flow allocation without partitioning (C) in a specific NPT. Obviously, the same routing algorithm (Dijkstra or others) is used for both cost computations.

3.5. **Tradeoff evaluation for NPT choice – NPCE operation – Phase 03.**

The input parameters for the tradeoff evaluation performed by the NPCE aiming to choose a new network state (NPT establishment) are:

- The set of generated NPTs; and
- The computed cost allocation (routing) for all NPTs considering the allocation of all traffic/flows (LSPs).

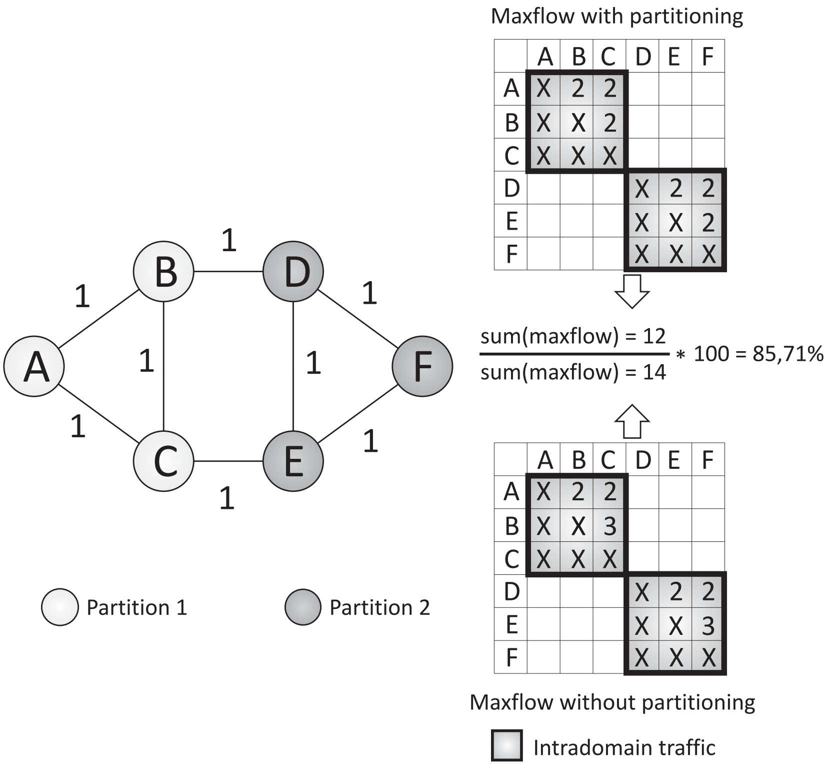
The computed allocation cost corresponds effectively to an externally visible execution time and the target of the partitioning strategy is exactly to get a significant execution time reduction considering, for instance, AMS solutions. As a first approach, a new network state (NPT) can be determined based exclusively on the smallest allocation cost just computed. However, there is an important problem with this straightforward approach: the network utilization (link utilization) might be severely compromised. When considered independently of other criteria, the reduction in execution time is improved by maximizing the number of partitions for NPTs and, in this case, it may generate a concentration of traffic in the NPT chosen. The result is that it may cause a higher concentration of flows in intra-domain edges, increasing the average bandwidth of intra-domain edges. In brief, the execution time is reduced but other performability parameters of the network may become compromised.

Thus, the partitioning strategy must include, at least, an additional tradeoff parameter that attempts to improve network utilization and, as such, guarantees a minimum level of performability for the network. The new input parameter defined by the partitioning strategy is the maximum flow loss. The maximum flow loss corresponds to the ratio of the sum of maximum flows[4] with and without the use of partitioning for all nodes (vertices) that belong to the same partition/domain. A simple illustrative example can be observed in Figure 6.

---

[3] The complexity cost for a routing algorithm for each partition/domain can be determined based on the number of vertex $|V|$ and the number of edges $||E||$ in a given graph G (network).

[4] The maximum flow between two vertices in a weighted graph is calculated according to Goldbergs Algorithm $O(|v|^3)$ [9].



Maxflow with partitioning

	A	B	C	D	E	F
A	X	2	2			
B	X	X	2			
C	X	X	X			
D				X	2	2
E				X	X	2
F				X	X	X

$$\frac{\text{sum(maxflow)} = 12}{\text{sum(maxflow)} = 14} * 100 = 85{,}71\%$$

	A	B	C	D	E	F
A	X	2	2			
B	X	X	3			
C	X	X	X			
D				X	2	2
E				X	X	3
F				X	X	X

Maxflow without partitioning

▢ Intradomain traffic

Figure 6. Maximum flow loss – Illustration

Thus, the sum of maximum flow corresponds to the sum of all maximum flows for every pair of vertices $(v1, v2) \in V$, calculated individually. The representation can be performed using a square matrix $V \times V$ and the sum corresponds to the sum of all cells of this matrix (Figure 6).

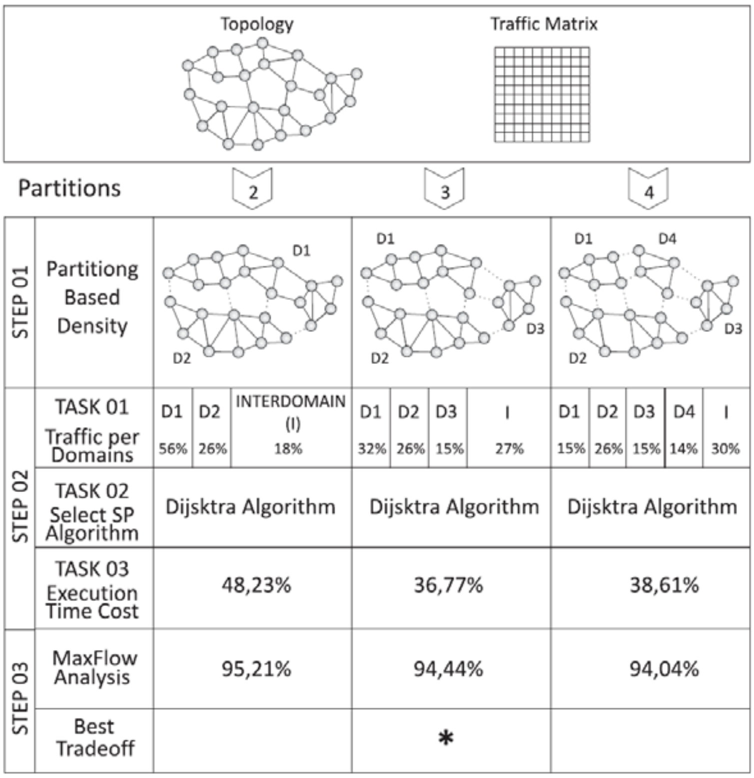
It is observed that the maximum flow between B and C nodes of partition 1 is three ($|f(B, C)| = 3$) without the use of partitioning. When partitioning is used, this value is reduced to two ($|f(B, C)| = 2$), since the paths $B \to D \to E \to C$ and $B \to D \to F \to E \to C$ cannot be used anymore since D, E and F nodes belong to partition two. The values of $|f(B, C)|$ with and without partitioning is illustrated in row two and column three of the arrays (Figure 6).

This reduction in the maximum flow does not occur for all pairs of nodes belonging to the partition/domain. For (A, B) and (A, C) (partition 1) and (D, F) and (E, F) pairs (partition 2) the maximum flow is equal to two ($|f| = 2$) with or without partitioning. Note that it is impossible to be greater than two since the A and F vertex have degree of two ($|d| = 2$).

In brief, this step computes maximum flows between nodes belonging to each partition/domain considering the partitioning and without partitioning. The main goal is to evaluate network capacity to maintain routing alternatives in the advent of the network partitioning. In operational terms the sequence of actions executed for choosing a new network state is as follows:

- The maximum flow percentage is calculated for all partitions/domains (NPTs); and
- The NPCE chooses among all partition/domain possibilities the best tradeoff between execution time and maximum flow loss.

In practical terms, the maximum flow loss is an input parameter for the NPTE defined by the administrator. The parameter definition may require an expertise or may be derived from a set of simulations (using the NPTE) where the administrator may star



		2	3	4
STEP 01	Partitiong Based Density			
STEP 02	TASK 01 Traffic per Domains	D1 56% D2 26% INTERDOMAIN (I) 18%	D1 32% D2 26% D3 15% I 27%	D1 15% D2 26% D3 15% D4 14% I 30%
	TASK 02 Select SP Algorithm	Dijsktra Algorithm	Dijsktra Algorithm	Dijsktra Algorithm
	TASK 03 Execution Time Cost	48,23%	36,77%	38,61%
STEP 03	MaxFlow Analysis	95,21%	94,44%	94,04%
	Best Tradeoff		*	

Figure 7. Partitioning algorithm example from two, three and four partitions

with a seed value and evaluates for diverse values their effect over the new NPT (network state) computed.

In terms of the quality of service scenario, the maximum flow loss parameter and the guarantee of all flows allocation in the selected NPT corresponds to the maintenance of basic QoS guarantees for the applications, minimally, in terms of applications bandwidth allocation.

Figure 7 illustrates all phases of the NPCE operation considering the tradeoff parameters defined by the administrator as follows: Maximum flow loss[5] (90%); and Computed cost allocation[6] (40%).

3.6. **Partitioning strategy results.** The partitioning strategy results are presented through a set of simulations[7] using four distinct scenarios. The purpose is to explore the potential applicability of the proposed solution and, at the same time, to validate it.

The distinct simulation scenarios primarily test scalability (stress test) and the execution time reduction for networks flow allocation in NPTs (T) in relation to the non-partitioned network (T). Another aspect evaluated is the impact of the partitioning strategy on the network performability requirements. This parameter corresponds to the quality threshold defined by the administrator and, fundamentally, guarantees all flow allocations (LSPs) for the chosen NPT while preserving network link utilization below managed defined limits.

---

[5] Indicate the minimum acceptable value for the maximum flow loss.

[6] Indicate the minimum acceptable gain in terms of the execution time (computed allocated cost = 40% represents a 60% reduction in execution time).

[7] Simulations were evaluated using R software [20] with igraph package [4].

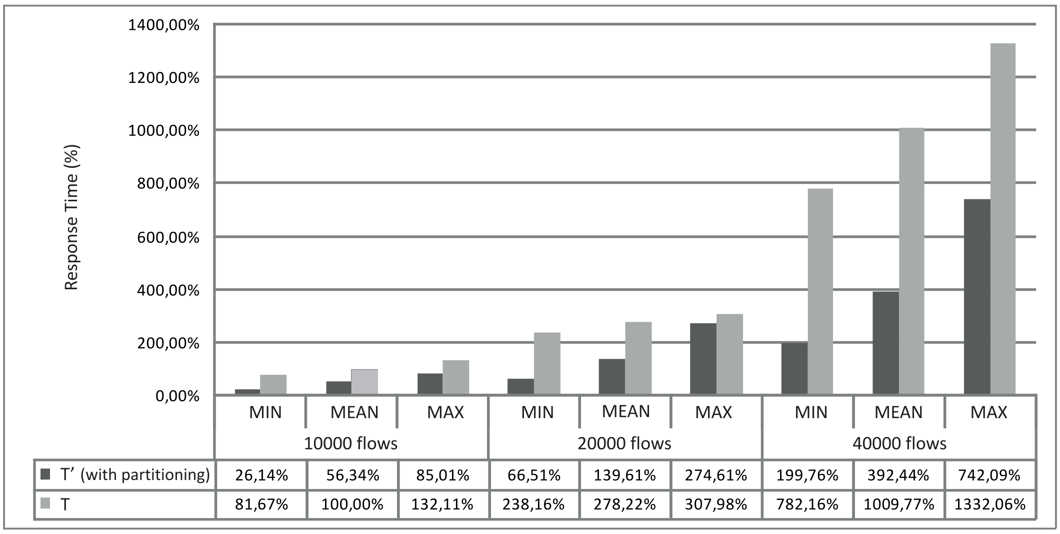
The routing algorithm (flow allocation) used to flows allocation in all simulations is Dijkstra algorithm. The network graphs used (G) were generated randomly by considering the following assumptions:

- All graphs should be connected;
- The minimum degree of all vertices of the graph must be greater than or equal to two. The goal is to avoid trivial routing; and
- All graphs should be simple graphs[8].

3.7. **Simulation scenario 01 – Response time behavior with flow scaling.** This first simulation scenario evaluates the behavior of the partitioning strategy in relation to its response time (time to define a new network state NPT) considering that the network is stressed with a significant traffic increase (number of flows scales). The initial scenario is composed by 25 nodes with 10.000 network flows (traffic). The network traffic is then aumented by 200% (20.000 flows) and by 400% (40.000 flows) respectively. The response time obtained with the partitioning strategy (T) adopted is then computed. The simulation parameters used are:

- 6 different topologies with 25 nodes ($v = 25$);
- For each network, six (6) different traffic profiles; and
- Number of flows: 10.000, 20.000 and 40.000 flows (representing approximately 2.3 GB, 4.7 GB and 9.5 GB of traffic).

In this run, 108 simulations 36 for each amount of flows configured were executed and the results are presented as average percentage execution time according to the initial scenario (25 nodes and 10.000 network flows) without partitioning (T) (Figure 8).



	10000 flows MIN	10000 flows MEAN	10000 flows MAX	20000 flows MIN	20000 flows MEAN	20000 flows MAX	40000 flows MIN	40000 flows MEAN	40000 flows MAX
■ T' (with partitioning)	26,14%	56,34%	85,01%	66,51%	139,61%	274,61%	199,76%	392,44%	742,09%
▨ T	81,67%	100,00%	132,11%	238,16%	278,22%	307,98%	782,16%	1009,77%	1332,06%

Figure 8. Simulation scenario 01 increasing flows

The parameters minimum (MIN), average (MEAN) and maximum (MAX) are values of T and T calculated 36 scenarios with 10.000, 20.000 and 40.000 flows (Figure 8). The results of execution time (percentage) are calculated as the ratio between the execution time for each scenario and the initial scenario $T_{mean_{25}v_{1}0.000f}$ (25 nodes with 10.000 network flows). Considering 10.000 flows the average value with partitioning strategy (T) is 56,34% of the average without partitioning (T).

---

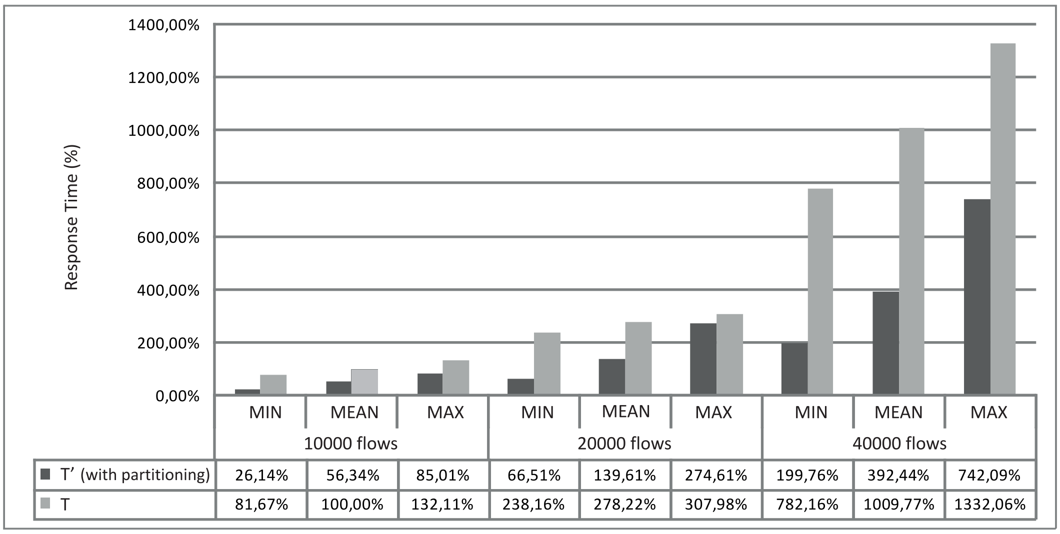
[8] By definition, a simple graph is an undirected graph that has no loops and no more than one edge between any two different vertices.

In the first simulation scenario it is possible to observe a better performance of the partitioning strategy (T) in relation to response time without partitioning (T) for the values of 10.000, 20.000 and 40.000 flows. Note also that the maximum execution time using the partitioning (T) is less than the average execution time of the solution without partitioning (T) to 10.000 ($T = 85,01\% < T_{mean} = 100,00\%$), 20.000 ($T_{\max} = 274,61\% < T_{mean} = 278,22\%$) and 40.000 ($T = 742,09\% < T_{mean} = 1009,77\%$) flows (Figure 8).

3.8. **Simulation scenario 02 – Response time behavior with node scaling.** In this simulation scenario, the number of network nodes is increased (scales) while keeping a fixed number of network flows (traffic). In this case, the objective is to evaluate the impact of the partitioning strategy in relation to the increasing network complexity resulting in having a significant augmentation in the number of network devices (routers, openflow switches, others). The parameters adopted in this simulation run are:

- 10.000 flows ($f = 10.000$);
- 6 different topologies with 25 nodes ($v = 25$), 50 nodes ($v = 50$) and 100 nodes ($v = 100$) each; and
- 3 different traffic profiles for each topologies set.

In this second scenario, the scalability validation is performed according to the network cardinality. The flows allocation time minimum (MIN, average (MEAN) and maximum (MAX) with the use of partitioning strategy (T) are compared with the same parameters without the use of the strategy (T). The results for 25, 50 and 100 nodes are observed in Figure 9. The average allocation time (MEAN) with the partitioning (T) was 56,34%, 58,51%, 53,07% in relation to the non-partitioned network (T). This significant reduction is result of NPT determination (presented in Section 2).

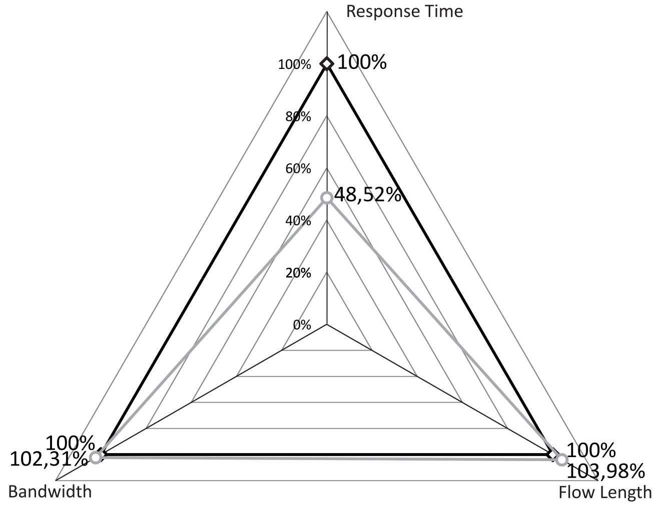


	10000 flows			20000 flows			40000 flows		
	MIN	MEAN	MAX	MIN	MEAN	MAX	MIN	MEAN	MAX
■ T' (with partitioning)	26,14%	56,34%	85,01%	66,51%	139,61%	274,61%	199,76%	392,44%	742,09%
■ T	81,67%	100,00%	132,11%	238,16%	278,22%	307,98%	782,16%	1009,77%	1332,06%

Figure 9. Simulation scenario 02 increasing nodes

3.9. **Simulation scenario 03 – Performability behavior of the partitioning strategy – Quality of Service requirements.** In the previous simulation runs the main result observed is that the partitioning strategy effectively reduces the computation time require to find a new network state even with an increasing computational demand (greater network computational complexity) for stressed scenarios created by augmenting both traffic demands and devices number in the network.

The next simulations scenario focuses on the partitioning strategy behavior in relation to two Quality of Service (QoS) parameters:



	Response Time	Flow Length	Bandwidth
◇ T	100%	100%	100%
○ T'	48,52%	103,98%	102,31%

Figure 10. Simulation scenario 03 performability behavior in relation to QoS parameters

- Average bandwidth utilization; and
- Path length

In short, the parameters used in the simulation run were:

- 10.000 flows ($f = 10.000$);
- 6 different topologies with 25 nodes ($v = 25$), 50 nodes ($v = 50$) and 100 nodes ($v = 100$) each; and
- 6 different traffic profiles for each topologies set.

On average, both bandwidth utilization and paths length had a similar performance showing an increase of 2.40% and 1.35% respectively with the partitioning strategy and, at the same time, presenting a significant response time reduction (46.49%) (Figure 10).

In brief, the result achieved shows that the QoS requirements are effectively guaranteed with the partitioning. The main technical consideration supporting this result is the higher concentration of paths within the domain belonging to the chosen NPT. Furthermore, the partitioning tradeoff also considers the maximum flow loss parameter as a measure that enforces intra-domain traffic.

4. **Final Considerations.** Autonomic management systems (AMSs) should be able to work with autonomic characteristics in networks with different cardinalities (nodes and links) and without topological or traffic constraints. To compute new states for computer networks considering, for instance, a Quality of Service scenario, requires considerable computer resources and, typically, represents a problem for autonomic systems which intend to compute and apply management solutions on-the-fly. In brief, AMS should deal with all inherent scalability issues associated with current computer networks: high number of routers and huge amount of flows, among others.

The partitioning strategy proposed and evaluated in this paper is an attempt to deal with the indicated scalability issue in network self-management context. The proposed network partitioning strategy demonstrated to be a feasible approach to deal with the scalability problem while considering a set of management requirements specified by the network administrators. This strategy has proven its capability to reduce the execution time to compute new network states considering the best tradeoff between the expected response time and parameters (bandwidth, path) reflecting the expected QoS guarantees.

The following steps in this research will focus in applying the inherent NPCE ability (capability) to compute new network states efficiently to other management areas like survivability and resilience schemes. In these cases, the computation of new states is an absolute must and, in brief, it is required the allocation of the actual traffic matrix over a subset of the previous operational network. NPCE implementation in a campus or industrial network is also being considered as complementary evaluation.

REFERENCES

[1] R. Bellman, On a routing problem, *Quarterly of Applied Mathematics*, vol.16, pp.87-90, 1958.

[2] R. Martins, S. Bezerra and J. S. B. Martins, Network partitioning and self-sizing methods for QoS management with autonomic characteristics, *APNOMS*, pp.151-160, 2009.

[3] A. Clauset, M. E. J. Newman and C. Moore, Finding community structure in very large networks, *Physical Review E*, vol.70, 2004.

[4] G. Csardi, *Igraph Manual*, 2010.

[5] M. K. Denko, L. T. Yang and Y. Zhang, *Autonomic Computing and Networking*, Springer Publishing Company, 2009.

[6] E. W. Dijkstra, A note on two problems in connexion with graphs, *Numerische Mathematik*, vol.1, no.1, pp.269-271, 1959.

[7] S. Dobson, S. Denazis, A. Fernndez, D. Gaïti, E. Gelenbe, F. Massacci, P. Nixon, F. Saffre, N. Schmidt and F. Zambonelli, A survey of autonomic communications, *ACM Trans. Auton. Adapt. Syst.*, vol.1, no.2, pp.223-259, 2006.

[8] E. Gelenbe, Steps toward self-aware networks, *Commun. ACM*, vol.52, no.7, pp.66-75, 2009.

[9] A. V. Goldberg and R. E. Tarjan, A new approach to the maximum flow problem, *Proc. of ACM Symposium on Theory of Computing*, New York, NY, USA, pp.136-146, 1986.

[10] A. Hakiri, P. Berthou and T. Gayraud, Survey study of the qos management in distributed interactive simulation through dead reckoning algorithms, *CoRR*, 2010.

[11] A. Hakiria, P. Berthoua, A. Gokhalec, D. Schmidtc and G. Thierrya, *Supporting End-to-end Scalability and Real-time Event Dissemination in the OMG Data Distribution Service over Wide Area Networks*, http://www.dre.vanderbilt.edu/ schmidt/PDF/elsarticle-journal.pdf, 2013.

[12] B. Jennings, S. van der Meer, S. Balasubramaniam, D. Botvich, M. O. Foghlu, W. Donnelly and J. Strassner, Towards autonomic management of communications networks, *Communications Magazine, IEEE*, vol.45, no.10, pp.112-121, 2007.

[13] D. B. Johnson, Efficient algorithms for shortest paths in sparse networks, *J. ACM*, vol.24, no.1, pp.1-13, 1977.

[14] Z. Jrad, F. Krief, L. Dehni and Y. Bennani, Artificial intelligence techniques in the dynamic negotiation of QoS: A user interface for the internet new generation, in *Autonomic Networking*, D. Gaïti, G. Pujolle, E. S. Al-Shaer, K. L. Calvert, S. A. Dobson, G. Leduc and O. Martikainen (eds.), Springer, 2006.

[15] G. Liu and C. Ji, Scalability of network-failure resilience: Analysis using multi-layer probabilistic graphical models, *IEEE/ACM Transactions on Networking*, vol.17, no.1, pp.319-331, 2009.

[16] R. Mortier, R. Isaacs and P. Barham, Anemone: Using end-systems as a rich network management platform, *Applications, Technologies, Architectures, and Protocols for Computer Communication*, pp.203-204, 2005.

[17] M. E. J. Newman, Finding community structure in networks using the eigenvectors of matrices, *Physical Review E*, vol.74, 2006.

[18] M. E. J. Newman and M. Girvan, Finding and evaluating community structure in networks, *Physical Review E*, vol.69, 2004.

[19] P. Pons and M. Latapy, Computing communities in large networks using random walks, *Journal of Graph Algorithms and Applications*, vol.10, no.2, pp.191-218, 2006.

[20] R Development Core Team, R: A language and environment for statistical computing, *R Foundation for Statistical Computing*, Vienna, Austria, 2009.

[21] J. P. G. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller and P. Smith, Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines, *Comput. Netw.*, vol.54, no.1245-1265, 2010.