Performance Modeling and Analysis of an Autonomic Router

Marco Gribaudo
Dipartimento di Elettronica, Informatica e Bioingegneria
Politecnico di Milano
via Ponzio 51
20133, Milano, Italy

Lelio Campanile
Mauro Iacono
Michele Mastroianni
Dipartimento di Matematica e Fisica
Università degli Studi della Campania
"L. Vanvitelli"
viale Lincoln 5
81100 Caserta, Italy

KEYWORDS

Performance evaluation; autonomic networks; software defined networks; Petri nets.

ABSTRACT

Modern networking is moving towards exploitation of autonomic features into networks to reduce management effort and compensate the increasing complexity of network infrastructures, e.g. in large computing facilities such the data centers that support cloud services delivery. Autonomicity provides the possibility of reacting to anomalies in network traffic by recognizing them and applying administrator defined reactions without the need for human intervention, obtaining a quicker response and easier adaptation to network dynamics, and letting administrators focus on general system-wide policies, rather than on each component of the infrastructure.

The process of defining proper policies may benefit from adopting model-based design cycles, to get an estimation of their effects. In this paper we propose a model-based analysis approach of a simple autonomic router, using Stochastic Petri Nets, to evaluate the behavior of given policies designed to react to traffic workloads. The approach allows a detailed analysis of the dynamics of the policy and is suitable to be used in the preliminary phases of the design cycle for a Software Defined Networks compliant router control plane.

I. INTRODUCTION

Autonomic Networking (AN) and Software Defined Networks (SDN) are the most promising approaches to computer networks that show the potential for supporting the needs of larger, more complex, faster infrastructures. While AN puts its focus on management issues, SDN try and address the need for moving to software services all the control plane.

SDN advocate an architectural shift in network devices, proposing the implementation of as many network layers as possible in software, ideally keeping in hardware only the physical one. This approach leverages the availability of high performance and highly reliable computing hardware at affordable costs, and the possibility of remote installation and update of software on network devices. The general idea is using software layers to implement a flexible and reconfigurable control plane of a network, completely defined by the administrator according to the needs, to benefit of the natural distribution of network devices and of the resulting centralized control of the network.

AN aims to support the management of large networks by allowing network devices to autonomously react to network conditions, by locally applying proper strategies according to general policies, thus allowing network managers and administrators to focus on global issues, planning and system-level specifications. The use of AN may allow faster and better adaptation of the network to traffic dynamics, compensation of failures, self-management, faster reaction, but the complexity of network infrastructures may cause instability and unnecessary failures because of the interactions of autonomic devices driven by locally correct, globally wrong solutions.

In order to support the design of network management policies, a proper modeling approach is needed that may help predict the effect of design decisions in different traffic conditions. Such modeling approach should encompass both local issues, dealing with a single network device, and global issues, that is, the dynamics resulting from the interactions, at a system level, of all the autonomic reactions in each network device. The approach must be supported by a proper methodology, capable of capturing all aspects of the design and maintenance process of complex networks.

This paper, that extends [6], is a first step towards the development of a suitable design approach. In particular, we propose a Stochastic Petri Nets (SPN) based modeling approach to design and evaluate the adaptive policy of an autonomic router that has to balance traffic generated by two variable sources, so to grant at least part of the bandwidth to each source, by detecting a regime change in the traffic of each source and enacting a proper symmetric and a proper asymmetric strategy.

The paper is organized as follows: in Section II some related works are presented to introduce readers to the problem; in Section III the case study is defined, to-

Communications of the ECMS, Volume 33, Issue 1,
Proceedings, ©ECMS Mauro Iacono, Francesco Palmieri,
Marco Gribaudo, Massimo Ficco (Editors)
ISBN: 978-3-937436-65-4/978-3-937436-66-1(CD) ISSN 2522-2414

gether with the router policies defined to manage traffic and the performance model developed to study the behavior of an autonomic router; in Section IV the case studied is evaluated and results are presented; conclusions follow in the last Section.

II. RELATED WORKS

Problems and opportunities related to autonomic networks are well analyzed and documented in the literature. A presentation of the main approaches adopted to organize the architecture of autonomic networks is provided by [12], that also discusses their management and related services: the authors also propose a unifying approach to the problem, to balance more conservative and more innovative proposals. Performance evaluation aspects, including related qualitative and quantitative metrics and main indicators, are examined in [10], that proposes these indicators as a tool to perform comparisons and decision making with different organizations or designs of autonomic network infrastructures. A detailed framework for the design and organization of autonomic networks based on a set of abstraction tools is presented in [5]. Autonomic management of networks and network components is analyzed in [7].

Future developments and roadmaps have been presented both by market players, such as in [1], and academia, such as in [9] and in [8], and standardization entities, such as in [13]. Such developments include future applications, extensions of currently adopted management policies and technologies, important technical issues that may shape innovation in the field, future networks, impact of and on Software Defined Networks, strategic domains such as networking infrastructures to support Cloud Computing, information-centered networking management, autonomic communications and 5G technology and applications [11].

Security is of course a relevant aspect to be considered, as autonomicity is somehow a form of delegation, thus it may open the way to innovative threats and new classes of vulnerabilities: for an overview, we suggest the reader refer to [4].

III. MODELING AN AUTONOMIC ROUTER

To simplify the presentation, we focus on a single channel shared by two sources, as shown in Figure 1. We imagine that there are two possible sources, α_1 and α_2 that might be subject to traffic bursts. Both sources are modelled as Markov Modulated Poisson Processes (MMPPs), in which the state of a Markov Chain defines the traffic produced by a source in a particular moment in time.

The traffic generated by both sources, respectively at rates μ_1 and μ_2, is multiplexed over a single channel of capacity $\mu = \mu_1 + \mu_2$. The autonomic router γ, that handles both sources, has two different buffers for the two traffic flows considered, addressed respectively as β_1 and β_2. Such buffers are characterized by capacity N_{Q1} and N_{Q2}.

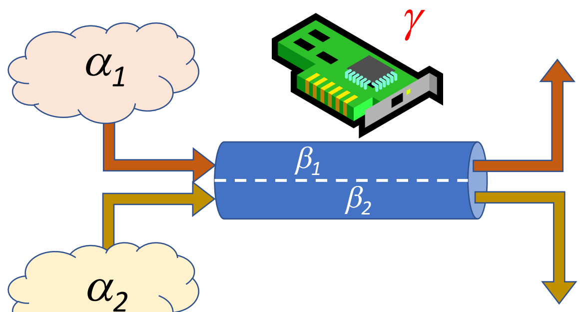


Fig. 1. The shared channel model

The autonomic router γ can offer, in different time instants, a variable service capacity to the two sources: we model its behavior with a state machine. In particular, we consider that the router alternates between three states: traffic is equally shared between the sources (state U), source α_1 is given more bandwidth (state P_1), or source α_2 is privileged (state P_2). Reconfiguration is driven by a set of rules that checks conditions over the current occupations of the two buffers β_1 and β_2: if a considered rule is verified by the queue sizes of the two sources for a given amount of time, a state switch is triggered. This reconfiguration requires a non-negligible amount of time: during this period, the router can only work at a fraction of its capacity, and both sources are affected by this event.

We model the systems presented in Figure 1 with the SPN shown in Figure 2.

A. THE SPN MODEL OF THE AUTONOMIC ROUTER

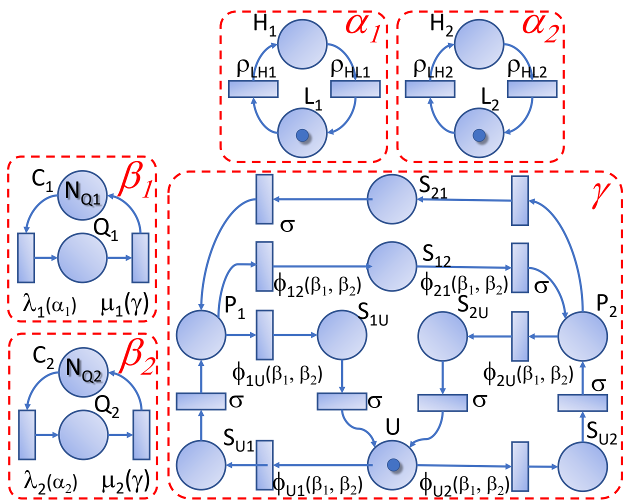


Fig. 2. The SPN model of the autonomic router

To simplify the presentation, both sources are modeled as a MMPP process with two states. Each source α_i (with $i = 1, 2$), is described by two places L_i and H_i that respectively identify a low and high traffic state. We consider the modulation processes defined by rates ρ_{LHi} and ρ_{HLi} that characterized respectively the transitions between low traffic to high traffic, and vice-versa.

The buffers of the two sources, β_1 and β_2, are modeled by places Q_1 and Q_2. New packets arrive at rate $\lambda_i(\alpha_i)$, depending on the state of modulation process of the considered source i, and are served at rate $\mu_i(\gamma)$, according to the state of the autonomic router modeled by sub-model γ (which will be described in the following). In this example, we consider service times to be exponentially distributed to easily fit in the SPN framework. Although being a simplifying assumption, exponential workloads have been measured in several interesting scenarios documented in the literature. The finite capacity of the buffers is guaranteed by places C_i, that are initially marked with N_{Qi} tokens to reflect the potentially different capacities of the two sub-systems.

The autonomic features of the router are modeled by the marked-graph SPN shown in sub-model γ. Places U, P_1 and P_2 model the states of the router: when one of such places is marked, the router is working in the corresponding state. Switching from state i to state j is performed at rate $\phi_{ij}(\beta_1, \beta_2)$ depending on the buffers occupations β_1 and β_2: the definition of these functions allows the implementation of different reconfiguration policies and will be discussed more in depth in Section III-B. Since switching takes a non-negligible amount of time, it is modeled by places S_{ij}, depending on the previous and next configuration state. To simplify the presentation, we assume switching time to be independent of the initial and final states and to be exponentially distributed with rate σ. Places of the considered SPN and the transition rates of the model are summarized respectively in tables I and II.

TABLE I: Places of the model

Place	Description
L_i	Low rate for source i
H_i	High rate for source i
Q_i	Packets in queue for source i
C_i	Capacity for source i
U	Uniform sharing of capacity
P_i	Priority to source i
S_{ij}	Reconfiguration from i to j

B. MODELING THE RECONFIGURATION POLICY

In each state of the autonomic router sub-model γ, reconfiguration is issued according to the rates given by $\phi_{ij}(\beta_1, \beta_2) = 1(\psi_{ij}(\beta_1, \beta_2))\theta$ where $1(\cdot)$ is the indicator function, $\psi_{ij}(\beta_1, \beta_2)$ is a predicate that defines the rule that triggers the state transition, and θ is a constant rate that defines the time a rule must be true before the switching process is initiated. In particular, as a simplifying assumption, we assume that each rule must be satisfied for an exponentially distributed amount of time, characterised by rate θ, before switching is triggered. Predicate $\psi_{ij}(\beta_1, \beta_2)$ can be used to model different autonomic routing reconfiguration policies. In this example, we decide to model the following policies: a symmetric policy, and a priority policy.

Case I: Symmetric policy

- Under normal (low) traffic, the channel is equally shared between the two sources.
- If a single source α_i is detected to be in high traffic, the router is reconfigured to privilege such source. Detection is performed by checking that the buffer occupancy of the considered source exceeds a high-level threshold χ_{Ti}, while the other is still under a low-level threshold χ_{Bj}.
- When the router privileges a source α_i, it returns to the equal share if its buffer occupancy goes below a low-level threshold χ_{Ti}, while the other remains below a high-level threshold χ_{Tj}.
- The router also returns to the equal share if its buffer occupancy of the other source goes above a high threshold χ_{Tj} while the privileged source remains above a low-level threshold χ_{Bi}. The rationale of this change is that, whenever both sources are in high traffic at the same time, it is fairer to equally share the channel between them, rather than privilege the one that becomes saturated first.
- Finally, from a state that privileges one source, the router can immediately jump to the states that privilege the other if there is an inversion of the bottleneck, that is if the buffer of the other source goes above a high-level threshold χ_{Tj}, while the privileged one goes below a low-level threshold χ_{Bi}

The proposed policy can be implemented in the following way. When the autonomic router is in the uniform channel sharing state U, the model switches to the state P_1 that privileges source α_1 if:

$$\psi_{U1}(\beta_1, \beta_2) \equiv (\#_{\beta_1}(Q_1) \geq \chi_{T1}) \wedge (\#_{\beta_2}(Q_2) < \chi_{B2})$$

where $\#_{\xi}(A)$ represents the marking of place A in sub-model ξ, and χ_{T1} and χ_{B2} are the two thresholds previously introduced, which can be used to tune the policy. Conversely, $\psi_{U2}(\beta_1, \beta_2) \equiv (\#_{\beta_2}(Q_2) \geq \chi_{T2}) \wedge (\#_{\beta_1}(Q_1) < \chi_{B1})$.

The autonomic router returns to the uniform sharing state U from the one that privileges source α_1, when:

$$\begin{aligned}\psi_{1U}(\beta_1, \beta_2) \equiv \Big(&(\#_{\beta_1}(Q_1) \\ &\leq \chi_{B1}) \wedge (\#_{\beta_2}(Q_2) \\ &< \chi_{T2}) \Big) \vee \Big((\#_{\beta_2}(Q_2) \\ &\geq \chi_{T2}) \wedge (\#_{\beta_1}(Q_1) \\ &> \chi_{B1}) \Big)\end{aligned}$$

In state P_1 the system can also switch to state P_2 if the traffic has an abrupt change between the two sources. This is identified by the following predicate:

$$\psi_{12}(\beta_1, \beta_2) \equiv (\#_{\beta_2}(Q_2) \geq \chi_{T2}) \wedge (\#_{\beta_1}(Q_1) \leq \chi_{B1})$$

TABLE II: Transition rates

Transition	Rate
ρ_{LHi}	Transition rate from Low to High traffic for source i
ρ_{HLi}	Transition rate from High to Low traffic for source i
$\lambda_i(\alpha_i)$	Arrival rates from source i
$\mu_i(\gamma)$	Service rate offered to source i
$\phi_{ij}(\beta_1, \beta_2)$	Controller transition rate from state i to state j
σ	Channel reconfiguration rate

Predicates that define the transition from state P_2 to either state U or P_1 can be defined in the same way.

Case II: Priority policy

In the second case, the first source has priority over the second one. In particular, the network reconfigures to privilege source β_2 only if β_1 is in low traffic. Whenever β_1 is detected to overflow buffers, the system immediately reconfigures to privilege that sources. This behavior is achieved with the following definitions of predicates $\psi_{ij}(\beta_1, \beta_2)$:

$$
\begin{aligned}
\psi_{U1}(\beta_1,\beta_2) &\equiv \#_{\beta_1}(Q_1) \geq \chi_{T1} \\
\psi_{U2}(\beta_1,\beta_2) &\equiv (\#_{\beta_2}(Q_2) \geq \chi_{T2}) \wedge (\#_{\beta_1}(Q_1) < \chi_{T1}) \\
\psi_{1U}(\beta_1,\beta_2) &\equiv (\#_{\beta_1}(Q_1) \leq \chi_{B1}) \wedge (\#_{\beta_2}(Q_2) < \chi_{T2}) \\
\psi_{12}(\beta_1,\beta_2) &\equiv (\#_{\beta_1}(Q_1) \leq \chi_{B1}) \wedge (\#_{\beta_2}(Q_2) \geq \chi_{T2}) \\
\psi_{2U}(\beta_1,\beta_2) &\equiv (\#_{\beta_2}(Q_2) \leq \chi_{B2}) \wedge (\#_{\beta_1}(Q_1) \leq \chi_{T1}) \\
\psi_{21}(\beta_1,\beta_2) &\equiv \#_{\beta_1}(Q_1) \geq \chi_{T1}
\end{aligned}
$$

IV. RESULTS

We have tested the proposed autonomic routing channel sharing policies on the configuration defined by the parameters presented in Table III. The notation $f(A)$ is used to denote that the corresponding function $f(\cdot)$ assumes the associated value when place A is marked: for example, $\lambda_1(L_1)$ denotes the arrival rate of packets from source α_1 when the modulating sub-model has a token in place L_1. To make the result realistic, we set the model parameter according to CISCO documentation [2]. We assume the buffer size equal to $N_{Qi} = 16$ packets. Accordingly, with a real-world network, we set a different rate for each parameter of the two sources. In particular, we set these parameters to an average of different real-world networks and routers observation, specifically, we consider both low-end and hi-end devices, analysed in various times of the day and under different network's loads. We set the low arrival rate, $\lambda_1(L_1)$ and $\lambda_2(L_2)$, when sources are normally used, to 5000 pkt/s, and we assign to the high arrival rates, $\lambda_1(H_1)$ and $\lambda_2(H_2)$, to 18000 pkt/s. Next, let us consider the low rate to high rate switching rate ρ_{LHn} and its dual, ρ_{HL1}: in this case, we use different parameters for the two queues to consider asymmetry in the system, as shown in Table III. The parameters that control the network reconfiguration and its network service speed. Network services are assumed to give a maximum throughput for the autonomic router corresponding to 20000 pkt/s. For state U, when the router equally divide the band between the queues, we define $\mu_1(U)$ and $\mu_2(U)$ to 10000 pkt/s, instead $\mu_1(P_1)$ and $\mu_2(P_2)$ equal to 16000 pkt/s and, finally, $\mu_2(P_1)$ and $\mu_1(P_2)$ equal to 4000 pkt/s. When the router reconfigures itself, we set $\mu_1(S_{ij})$ and $\mu_2(S_{ij})$ to 4000 pkt/s. The thresholds that trigger a change of state from up to low state and vice-versa for both the queues are defined as follows: χ_{B1} and χ_{B2} equal to 14 packets, χ_{T1} and χ_{T2} equal to 7 packets.

TABLE III: Parameters of the proposed SPN model

Parameter	Value	Parameter	Value
ρ_{LH1}	0.1 s^{-1}	ρ_{LH2}	0.05 s^{-1}
ρ_{HL1}	0.15 s^{-1}	ρ_{HL2}	0.1 s^{-1}
$\lambda_1(L_1)$	5000 pkt/s	$\lambda_2(L_2)$	5000 pkt/s
$\lambda_1(H_1)$	18000 pkt/s	$\lambda_2(H_2)$	18000 pkt/s
$\mu_1(U)$	10000 pkt/s	$\mu_2(U)$	10000 pkt/s
$\mu_1(P_1)$	16000 pkt/s	$\mu_2(P_1)$	4000 pkt/s
$\mu_1(P_2)$	4000 pkt/s	$\mu_2(P_2)$	16000 pkt/s
$\mu_1(S_{ij})$	4000 pkt/s	$\mu_2(S_{ij})$	4000 pkt/s
N_{Q1}	16 pkt	N_{Q2}	16 pkt
σ	0.01 s^{-1}	θ	0.1 s^{-1}
χ_{B1}	14 pkt	χ_{B2}	14 pkt
χ_{T1}	7 pkt	χ_{T2}	7 pkt

We start considering the steady-state distribution of model components, depicted in Figure 3. For components α_i and β_i, we show respectively the distribution of the number of tokens in places L_i and Q_i. For component γ, we show where the token is, using the following convention: $0 - U$, $1 - P_1$, $2 - P_2$, $3 \ldots 8 - S_{ij}$. Results for the two policies are shown in Figure 3. As can be seen, in both cases overflow conditions tend to be pretty severe, creating U-shape distribution for what concerns the queue lengths. The symmetric case tends to have the same distribution for both queues, while in the priority policy case the privileged source tends to have a higher probability of an empty system and a lower probability of full buffer as expected. It is interesting to see how, in the symmetric policy case, states P_1 and P_2 of the controller are almost equiprobable, while for the privileged case states U and P_2 occur very rarely.

Next, we consider the transient evolution of the average buffer occupation, for three different starting points: the empty system (*-emp*), the system when it switches to high traffic for source β_1 (*-H1*), and for source β_2 (*-H2*). The transient behavior for the second and third study has been studied by constructing a starting distribution that considers controller γ in state U, the corresponding modulating process α_1 or α_2 in state H_1 or H_2, and all the other components in steady-state. Results are shown in Figure 4. In the symmetric case, time scales are emphasized: curves corresponding to the empty system starts reaching the steady-state as the modulating process starts switching to high traffic states.

In particular, it is interesting to relate the evolution of the average buffer occupancy with the probability of having the controller γ in a given state, as shown in Figure 5.

The most important results come from the analysis of the loss rates, as shown in Figure 6, where the three possible initial states are considered during the transient evolution of the model. In the empty system case, losses occur only when some of the two sources switch in the high traffic state. When the system starts with one source in high traffic mode, the corresponding source experiences a high loss until the system reconfigures. When a symmetric policy is used, the other source is generally only marginally affected by the reconfiguration. In the priority policy, instead, reconfiguration always has a bad impact on source β_2, even when the system starts with β_2 in high traffic: this occurs because whenever β_1 becomes saturated, the other source is penalised. It is then mandatory to carefully study the effect of priority on the other class before using such a policy: the proposed methodology could then be used as a starting point for an optimization algorithm that selects the parameters to provide the best trade-off between priority and fairness between the users.

Even if it is based on a SPN, this model has been analyzed by means of the SIMTHESys solver for Hybrid Systems [3], here applied to a single-formalism totally discrete case, seen as a degenerate case of its application to a model without a continuous part. In this case, this solver has been exploited to leverage its capability of considering regime changes, originally designed to apply the effects of discrete state space transitions on continuous parts of a hybrid model, and here used to compute the initial states of the transient analysis when considering the transition to the high-level regime. Details on the internals and the implemented mathematical framework are here omitted for sake of space, as they are mainly focusing on the hybrid aspects that are not used here, and we recommend [3] to interested readers.

V. CONCLUSIONS

We presented in this paper a first step towards the design and evaluation of traffic management policies for autonomic routers. The adopted quantitative approach allows to get significant insights about the behavior of the target router in different traffic conditions, with the chance to evaluate and compare alternative configurations or different policies, since the very early phases of the design cycle, and without the need for building a detailed simulation of the hardware, keeping high the level of abstraction. Results proved the effectiveness of the approach, as they show realistic behaviors and interesting effects.

Future works include validation with specialized network simulators, such as NS3, and a comparison with measures taken in an experimental setup on a router, as well as the definition of a network system-level modeling approach and a proper design methodology to support management policy design in large networks and Cloud-oriented networks. Further goals include the definition of model-based SDN software definition.

References

[1] Autonomic networking configuration and deployment guide. https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/auto˙net/configuration/xe-16/an-auto-cfg-deploy.html. Technical report.

[2] Buffer Tuning for all Cisco Routers. https://www.cisco.com/c/en/us/support/docs/routers/10000-series-routers/15091-buffertuning.html.

[3] E. Barbierato, M. Gribaudo, and M. Iacono. Modeling hybrid systems in SIMTHESys. *El. Notes in Th. Computer Science*, 327:5 – 25, 2016.

[4] M. Barrere, R. Badonnel, and O. Festor. Vulnerability assessment in autonomic networks and services: A survey. *IEEE Communications Surveys and Tutorials*, 16(2):988–1004, 2014.

[5] G. Bouabene, C. S. Jelger, C. F. Tschudin, S. Schmid, A. Keller, and M. May. The Autonomic Network Architecture. *IEEE Journal on Selected Areas in Communications*, 28(1):4–14, 2009.

[6] L. Campanile, M. Gribaudo, M. Iacono, and M. Mastroianni. Quantitative modeling of the behaviour of an autonomic router. In *Valuetools 2019, Proceedings of the 12th EAI International Conference on Performance Evaluation Methodologies and Tools, March 13-15, 2019, Palma de Mallorca, Spain*, 2019.

[7] M. A. Khan, S. Peters, D. Sahinel, F. D. Pozo-Pardo, and X. T. Dang. Understanding autonomic network management: A look into the past, a solution for the future. *Computer Communications*, 122(January):93–117, 2018.

[8] D. Matsubara, T. Egawa, N. Nishinaga, V. Kafle, M. K. Shin, and A. Galis. Toward future networks: A viewpoint from ITU-T. *IEEE Communications Magazine*, 51(3):112–118, 2013.

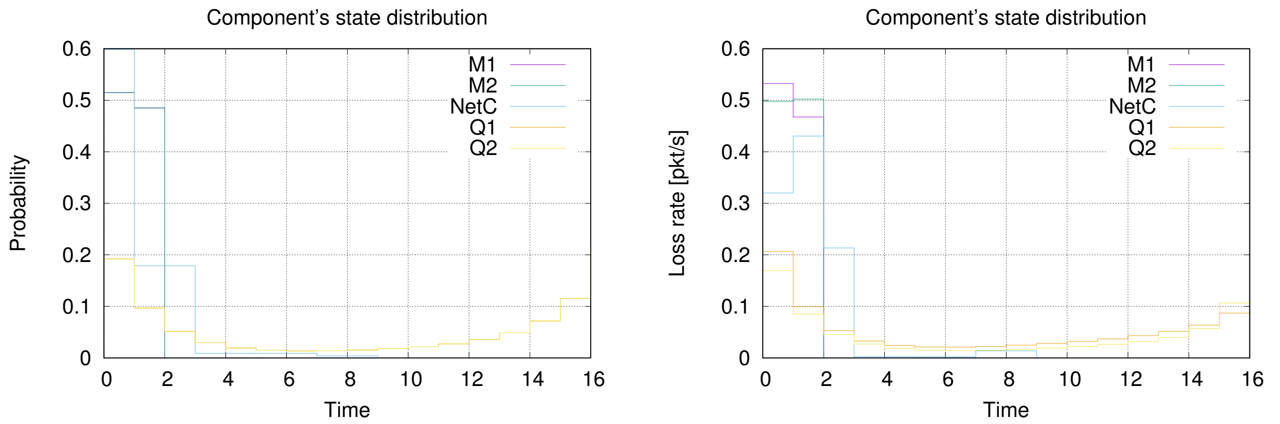
[9] D. Miorandi, L. Yamamoto, and F. De Pellegrini. A survey of evolutionary and embryogenic approaches to autonomic networking. *Computer Networks*, 54(6):944–959, 2010.

[10] Z. Movahedi, M. Ayari, R. Langar, and G. Pujolle. A survey of autonomic network architectures and evaluation criteria. *IEEE Communications Surveys and Tutorials*, 14(2):464–490, 2012.

[11] V. e. a. Nejkovic. *Big Data in 5G Distributed Applications*, volume 11400. 2019.

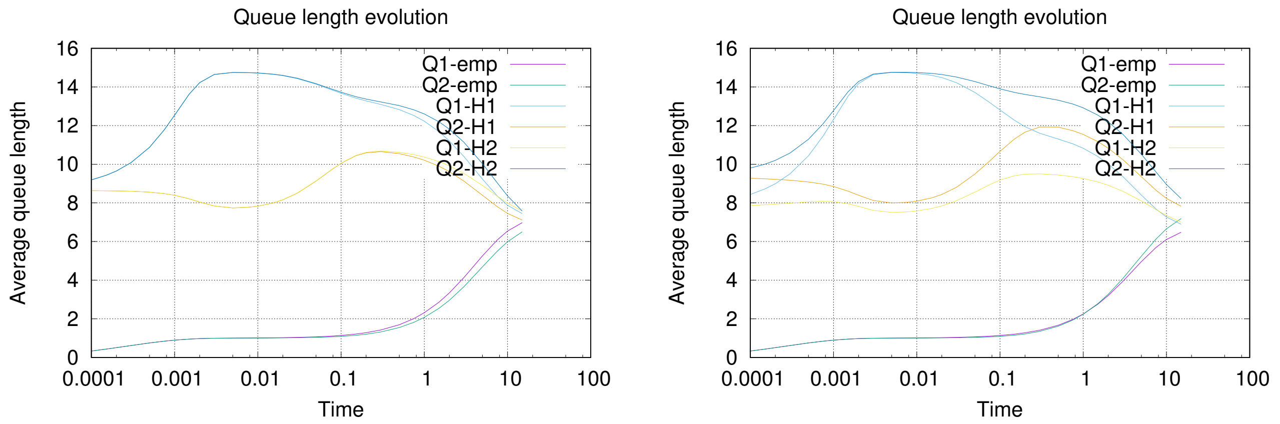
[12] I. G. Siqueira, L. B. Ruiz, and A. Loureiro. A survey of autonomic networking architectures: towards a Unified Management Framework . *Internation Journal of Network Management*, (October 2005):17–31, 2014.

[13] Z. Zhao, E. Schiller, E. Kalogeiton, T. Braun, B. Stiller, M. T. Garip, J. Joy, M. Gerla, N. Akhtar, and I. Matta. Autonomic communications in software-driven networks. *IEEE Journal on Selected Areas in Communications*, 35(11):2431–2445, 2017.



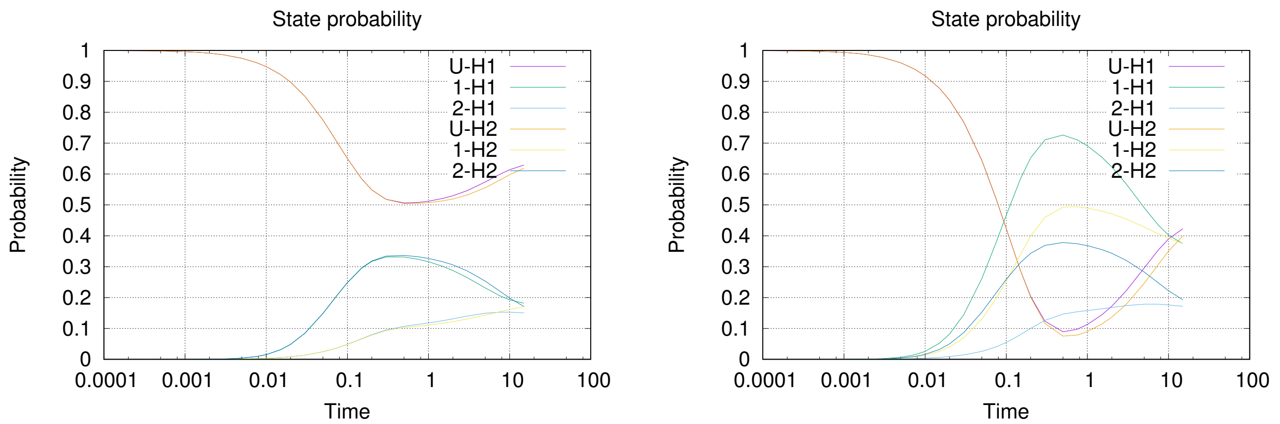
Policy I Policy II

Fig. 3. Probability distribution of the states of model components (M1 is 0 when α_1 is in state L_1, is 1 when α_1 is in H_1; M2 is 0 when α_2 is in state L_2, is 1 when α_2 is in H_2)



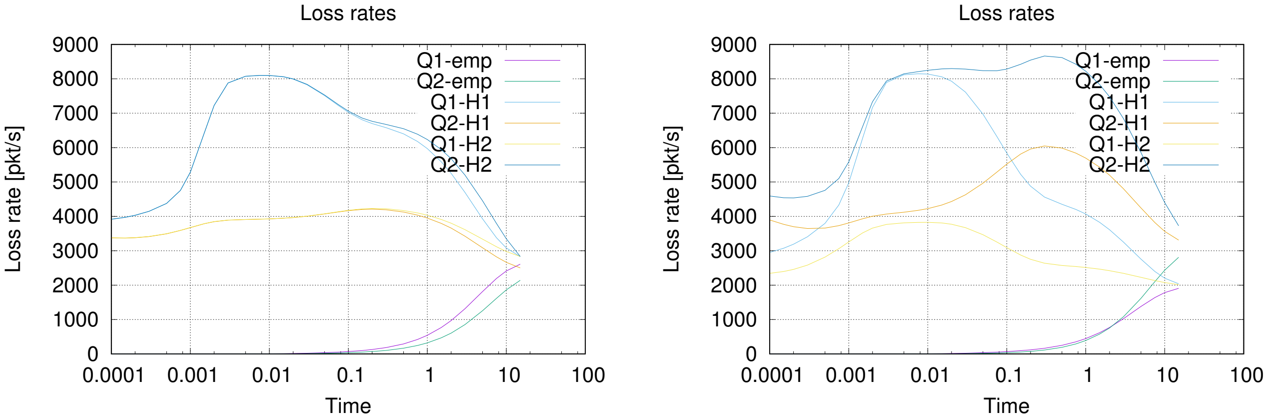
Policy I Policy II

Fig. 4. Queue length evolution



Policy I Policy II

Fig. 5. Autonomic router state distribution as a function of time



Policy I Policy II

Fig. 6. Loss rate as a function of time

AUTHOR BIOGRAPHIES

LELIO CAMPANILE is a PhD. student at Dipartimento di Matematica e Fisica, Università degli Studi della Campania "L. Vanvitelli", Caserta, Italy, where he has been a technician, a network administrator and an expert for many local and regional projects. He holds a M. Sc Degree in Computer Science. His email is lelio.campanile@unicampania.it.

MARCO GRIBAUDO is an associate professor at the Politecnico di Milano, Italy. He works in the performance evaluation group. His current research interests are multi-formalism modeling, queueing networks, mean-field analysis and spatial models. The main applications to which the previous methodologies are applied comes from cloud computing, multi-core architectures and wireless sensor networks. His email is marco.gribaudo@polimi.it.

MAURO IACONO is a tenured assistant Professor and Senior Researcher in Computing Systems at Dipartimento di Matematica e Fisica, Università degli Studi della Campania "L. Vanvitelli", Caserta, Italy. His research activity is mainly centered on the field of performance modeling of complex computer-based systems, with a special attention for multiformalism modeling techniques. His email is mauro.iacono@unicampania.it. More information about his activities is available at his website http://www.mauroiacono.com.

MICHELE MASTROIANNI is currently the Data Protection Officer of Università degli Studi della Campania "L. Vanvitelli", Caserta, Italy, and is also a research associate at Dipartimento di Matematica e Fisica of the same University. He holds a M. Sc. degree in Electrical Engineering and a Ph.D. degree in Management Engineering, and has been Network Manager at the same University, project leader and expert for many local, regional and national technical projects. He also teach Computer Science courses at University Vanvitelli. His email is michele.mastroianni@unicampania.it.