Simulation Modelling Practice and Theory 31 (2013) 169–185

Contents lists available at SciVerse ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

On the effectiveness of Linux containers for network virtualization

G. Calarco *, M. Casoni

Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia, Via Vignolese 905, 41125 Modena, Italy

ARTICLE INFO

Article history:
Received 25 July 2012
Received in revised form 21 November 2012
Accepted 22 November 2012
Available online 23 December 2012

Keywords:
Network emulation
Virtual network
Software router
Linux container

ABSTRACT

This paper presents a novel approach to the study of multi-technological networks based on Linux containers and software emulators. We illustrate the architecture and implementation issues of a modular and flexible testbed (NetBoxIT) that supports the virtualization and the concurrent, real-time execution of several independent emulators on a single, multi-core hardware platform. Distinct virtual networks can be instantiated, and connected to synthesize heterogeneous networks configurations. NetBoxIT is also an open platform, which can be interfaced with external networks and nodes, enabling the evaluation of true users' applications and protocols. We examine its performance under different viewpoints (scalability, computational load, timing overheads, and realism) and we show how the proposed testbed architecture leads to a general-purpose, reliable, and economical tool for assessing multipart networks with respect to real-world applications. Moreover, we discuss which are the current and future technologies that can be introduced to reduce the testbed timing overheads and to further improve performance.

1. Introduction

Heterogeneous networks design is often a challenging task. In particular, when different technologies are required to inter-operate, it is hard to predict if a planned infrastructure can accomplish the expected operational performance. Things can become even trickier when the designer aims at foreseeing how the compound system could possibly be tuned to meet the peculiar requirements of a specific user application (e.g., a certain quality of service constraint in multimedia communications). In this sense, the a priori availability of a smart design tool would offer a strategic advantage, as it could help to converge toward an optimal network architecture from the very beginning. In particular, it would be attractive to have an assessment tool that enables the rapid comparison of different and alternative architectures. Such a tool should offer at least some fundamental characteristics: realism and repeatability, for a consistent comparison of distinct design choices; flexibility and scalability, to manage more and more intricate scenarios; and, possibly, be low-cost.

In this paper, we illustrate the software architecture and the implementation issues of a modular and scalable software testbed (NetBoxIT) which can be fruitfully utilized to simulate complex networks, possibly organized with different topologies and technologies. NetBoxIT is conceived to support the creation, and interconnection of several, coexisting virtual networks ("netboxes") on a single, multi-core hardware platform. In summary, using container-based virtualization techniques and a network simulation tool, a number of distinct netboxes (i.e., using different configurations each) can be run concurrently, to mimic the separate portions of a heterogeneous network. By means of internal bridges or external true-world equipment, these virtual networks can be interconnected to assemble the model of the overall network infrastructure under investigation. Fig. 1 depicts the conceptual scheme of a netbox.

* Corresponding author. Tel.: +39 3385657879.
E-mail address: giorgio.calarco@unimore.it (G. Calarco).

1569-190X/$ - see front matter
http://dx.doi.org/10.1016/j.simpat.2012.11.007

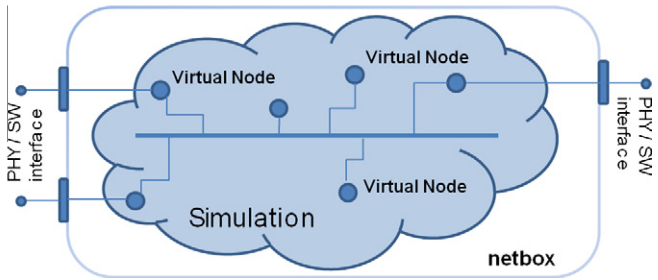

Fig. 1. The abstract scheme of a netbox: a whole network is emulated within an insulation container and interfaced to external entities or other modules by physical or software interfaces.

The proposed testbed is designed with a novel architecture and it aims at offering a number of advantages. First, it is modular and extensible, to let the designer arrange complex network scenarios by the combination of simple building blocks. Virtualization helps creating a first level of insulation among netboxes: the overall computing resources (i.e. mainly, the group of available CPU-cores and the memory address space) can be split and decoupled among them, so that their processing co-interference is limited. To reinforce encapsulation further, the network emulation logic (i.e., the operational parameters of any simulated network) are entirely enclosed inside each corresponding netbox (i.e., no network emulation processing is executed out of their boundaries). In this manner, netboxes are both computationally and logically self-contained, and can be instantiated and employed as if they were hardware emulation devices. Moreover, thanks to the chosen emulator, NetBoxIT can offer a fairly realistic modeling of several network standards, and supports real-time data handling, with negligible timing overheads against the represented true network (i.e., data traverse the virtual networks with the same latencies that would occur in reality). Also, NetBoxIT aims at being an open platform, which can be transparently interfaced with external equipment and nodes, to further increase the realism of simulations and verify networks behavior against true-world applications. Finally, it is low-cost, being based on PC-class hardware and open-source, "off-the-shelf" software only.

The paper is organized as follows: in Section 2, we give an overview of the pursued design guidelines in comparison with existing tools used in networks design nowadays. Section 3 briefly sketches the existing virtualization technologies and how these can be employed to assemble virtual networks. In Section 4 we bring to light the hardware and software NetBoxIT components, the motivations that led to their choice and the key issues we have taken into account during their deployment. In Section 5 we focus on experimental trials, with the aim at proving that the testbed is feasible for heterogeneous networks emulation. Finally, we draw some conclusions and report our current and future work.

2. Networks design: methodologies and trends

A wide variety of techniques can be employed for networks design, ranging from pure mathematical models to real-world testbeds. Mathematical queuing modeling is the most abstract methodology, but is usually proficient only for qualitative evaluations and can lead to unmanageable levels of complexity when applied to specific situations (e.g., wireless mesh networks). Physical testbeds, that use true devices, are unquestionably realistic, but frequently impracticable for multi-technological networks study, in particular for wide-ranging topologies, due to the time and costs for their implementation. Hardware emulators are based on special-purpose equipment: they are fairly reliable to reproduce the behavior of a certain network link and can be cascaded to reproduce a certain heterogeneous network. These commercial devices, however, are typically close to modifications by the researcher and often expensive. Simulation tools are indeed a more popular choice: they are adaptable, quick and cheap to deploy, and repeatable, as well as they frequently offer reliable and faithful networks modeling. Nevertheless, the simulation of complex networks usually requires a vast amount of computing resources; moreover, pure simulations cannot be validated against true-world applications, due to their lack of interoperability with existing applications and systems. For these reasons, in the recent past software network emulation has gained a lot of interest in the networking research community. The key idea behind it is to follow a hybrid approach, where a software simulator (executed using general-purpose hardware) can be mixed (using true network interfaces) with real network components and applications, with the aim of increasing the fidelity of results and to perform the validation of a planned infrastructure against real traffic. Similarly to pure simulators, software emulators are fast to configure and low-cost, and their realism depends on the network modeling accuracy. However, a fundamental distinction is that an emulator cannot run in a virtual simulated time, to coexist with real network entities. Therefore, it should have enough computing resources to respect real-time constraints when processing incoming data: if true applications exchange traffic through an emulated topology, it is necessary to guarantee that the emulator does not introduce any extrinsic delay (or, at least, that it is limited and measurable). There are therefore some critical aspects, mainly related to the choice of the underlying hardware and software platform.