Available online at www.sciencedirect.com
ScienceDirect
Computer Communications 30 (2007) 943–956
computer communications
www.elsevier.com/locate/comcom

Experiences with node virtualization for scalable network emulation

Steffen Maier *, Daniel Herrscher, Kurt Rothermel

University of Stuttgart, Institute of Parallel and Distributed Systems (IPVS), Universitätsstr. 38, D-70569 Stuttgart, Germany

Available online 18 September 2006

Abstract

During the development of network protocols and distributed applications, their performance has to be analyzed in appropriate environments. Network emulation testbeds provide a synthetic, configurable network environment for comparative performance measurements of real implementations. Realistic scenarios have to consider hundreds of communicating nodes. Common network emulation approaches limit the number of nodes in a scenario to the number of computers in an emulation testbed. To overcome this limitation, we introduce a virtual node concept for network emulation. The key problem for node virtualization is a transparent, yet efficient separation of node resources. In this paper, we provide a brief survey of candidate node virtualization approaches to facilitate scalable network emulation. Based on the gathered insights, we propose a lightweight virtualization solution to achieve maximum scalability and discuss the main points regarding its implementation. We present extensive evaluations that show the scalability and transparency of our approach in both a traditional wired infrastructure-based, and in two wireless ad hoc network emulation scenarios. The measurements indicate that our solution can push the upper limit of emulation scenario sizes by a factor of 10–28. Given our emulation testbed consisting of 64 computers, this translates to possible scenario sizes of up to 1792 nodes. In addition to the evaluation of our virtualization approach, we discuss key concepts for controlling comprehensive emulation scenarios to support scalability of our system as a whole.

Keywords: Software performance evaluation; Network emulation; Mobile ad hoc network; Scalability; Virtual routing

1. Introduction

During the design and implementation of distributed applications and network protocols, it is essential to analyze the impact of various network environments on their performance. While mathematical analysis and simulations are commonly used in early design stages, measurements are used to confirm the theoretical results as soon as implementations become available. Such measurements usually compare the performance of one implementation in different network environments or of different implementations in the same network environment.

Comparative performance measurements in real environments are considered problematic for two reasons. First, especially in scenarios with mobile nodes and wireless networking, it is hard to obtain multiple comparable measurement runs. Secondly, resource requirements prohibit measurements in larger scenarios. Therefore, there is strong demand for synthetic network environments that can be parametrized in order to reproduce an original or fictitious network.

The process of introducing network properties that differ from the actual properties of the hardware in use is called *network emulation*. A *network emulation tool* is software capable of altering network traffic in a specified way. A facility consisting of a combination of flexible networking hardware and suitable emulation tools is called *network emulation testbed*. Network protocols and distributed applications subjected to performance measurements in a network emulation testbed are called *software under test*.

Comparative performance measurements for mobile computing scenarios, e.g. the evaluation of an ad hoc routing protocol, typically require large scenarios with

* Corresponding author. Tel.: +49 711 7816 226; fax: +49 711 7816 424.
E-mail addresses: maier@ipvs.uni-stuttgart.de (S. Maier), herrscher@ipvs.uni-stuttgart.de (D. Herrscher), rothermel@ipvs.uni-stuttgart.de (K. Rothermel).

0140-3664/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2006.08.018

hundreds of nodes. The analysis of new applications for traditional infrastructure-based networks, e.g. a large-scale location service, may also require a high number of nodes, since both the end systems and the intermediate systems of the underlying infrastructure have to be considered.

Common network emulation systems assume that one communicating node in an emulation scenario corresponds to one physical computer in an emulation testbed. This severely limits the scalability, since testbeds with the required number of hundreds of computers are typically not available.

However, a number of applications aiming at resource-poor devices, e.g. in mobile computing scenarios, only need a fraction of the resources that a testbed node can provide. Therefore, we propose to run several instances of the software under test on a single testbed node ("physical node," *pnode*) [1]. Each instance of the software under test has to be provided a separate execution environment ("virtual node," *vnode*). In this paper, we give a brief survey of candidate approaches for node virtualization. Based on these approaches, we present a transparent, yet lightweight and thus very scalable solution to node virtualization for network emulation testbeds. In addition to node virtualization in the system core, we also discuss key concepts that support scalability of our system as a whole. Our implementation supports scalable emulation not only of networks consisting of point to point links but also of shared media based networks such as mobile ad hoc networks and even arbitrary combinations for the emulation of hybrid networks.

The remainder of this paper is structured as follows. The Network Emulation Testbed, which we use as a basis for our scalable network emulation approach, is introduced in Section 2. In Section 3, we provide a brief survey of candidate node virtualization approaches. We choose one of the candidate approaches for our implementation, which we discuss in Section 4. In Section 5, we provide extensive measurements showing the scalability of our approach for two important kinds of scenarios: emulation of infrastructure-based networks and MANETs (mobile ad hoc networks). For the latter, we also present a comprehensive real life example for performance analysis of an ad hoc routing protocol. Furthermore, we discuss the achievable degree of transparency for the software under test. In Section 6, we address key concepts to fully support scalability of our system as a whole. We discuss related work in Section 7. Finally, we conclude the paper in Section 8.

2. Overview of the network emulation testbed

The Network Emulation Testbed (NET) [2] at the University of Stuttgart provides the basis for our scalable network emulation approach. It consists of 64 PC-nodes connected by a monolithic, programmable gigabit switch, and a separate administration network for setup and control (see Fig. 1). Using IEEE 802.1Q VLAN (virtual LAN) technology, the gigabit switch is able to create an

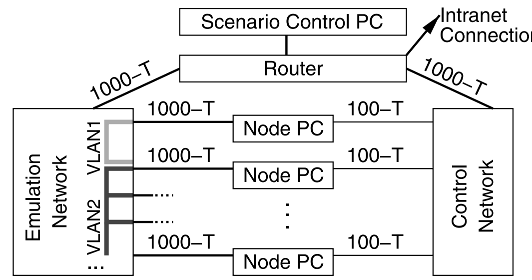

Fig. 1. The network emulation testbed.

arbitrary connection topology between the nodes. Each point-to-point link or shared media network segment in an emulation scenario, e.g. a WLAN (wireless LAN) channel, is mapped to a uniquely tagged VLAN.

On a testbed node, several tagged VLANs represent several virtual network interfaces, each of which is assigned a separate instance of our custom emulation tool. Each tool introduces the desired artificial network properties. It enables the configuration of arbitrary bandwidth limitations, delays, and frame error loss ratios. Additionally, to enable the realistic emulation of shared media networks, the effects of a MAC (media access control) layer can be reproduced. At the present time, this tool is capable of emulating IEEE 802.3 (Ethernet) [3]. We are currently completing an extension of the tool to allow the emulation of the ad hoc mode of IEEE 802.11 WLAN.

Our network emulation tool provides the service level abstraction of an unreliable datagram service to the software under test (see Fig. 2). This is the lowest possible emulation abstraction feasible to be implemented in software. The tool is implemented as a virtual network device driver, and therefore completely transparent to implementations on the network layer. As a result, the protocol stack including the network layer and all higher layers can be considered as software under test.

In order to control the distributed network emulation tools during an experiment, we pursue a hybrid architecture including a central scenario controller. The controller dynamically updates the parameters of the emulation tools. For MANET emulation, this includes changing connection quality and thus frame error rates between communicating nodes. The connection quality is automatically derived

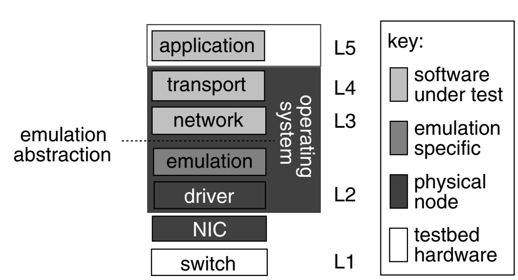

Fig. 2. Software under test and network emulation tools on a physical node in NET.