

The Topology Management Tool - A Demonstration

Dennis Schwerdel, Bernd Reuther, Paul Mueller
Integrated Communication Systems Lab, University of Kaiserslautern, Germany
{schwerdel, reuther, pmueller}@informatik.uni-kl.de

I. INTRODUCTION

In the last years a lot of holistic research efforts investigate concepts and technologies for future networks. All of these research projects need ways to evaluate their ideas and results. Experimental facilities aim to provide a realistic environment for experiments using emulation techniques.

Distributed research projects often result in distributed research hardware like in the German-Lab project. Experimental facility software must be able to handle the restrictions and features of distributed resources.

Networking experiments often need increased control over the network environments. This includes configurable link characteristics and network topologies.

In the German-Lab project, the Topology Management Tool [1] (ToMaTo) has been developed as an experimental facility software to run networking experiments on.

II. ToMaTo

ToMaTo allows users to build networking topologies containing *devices* and *connectors*. Devices are active components like computers that run the software of the experiment and are the only sources and sinks of data. Connectors are network components that connect devices and transport their data exhibiting certain configurable characteristics.

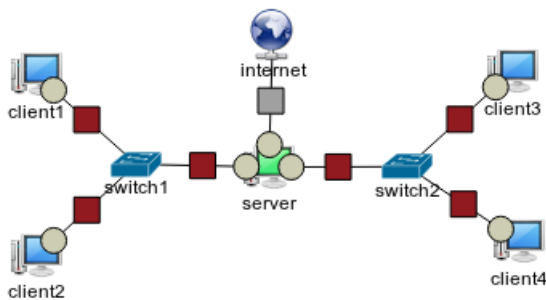


Figure 1. Example topology

ToMaTo currently supports two device types: OpenVZ¹ and KVM² virtual machines. OpenVZ is a container-based virtualization solution for Linux. OpenVZ virtual machines can run nearly all Linux systems and provide an environment similar to a physical machine from a user-space perspective.

¹OpenVZ: container-based virtualization technology (<http://wiki.openvz.org>)

²KVM: Kernel-based Virtual Machine (<http://www.linux-kvm.org>)

Since OpenVZ only runs a single kernel for all virtual machines some limitations arise. The kernel can not be modified, extended using modules or configured using *sysctl* from a virtual machine. Also the kernel version limits the guest systems to Linux operation systems that are compatible with a current kernel. Virtual network interfaces can be created on the OpenVZ host and exposed to the guest. Being a container-based virtualization, OpenVZ is very lightweight and offers flexible resource management (i.e. the external memory usage of a VM equals the usage inside the VM).

KVM offers a full virtualization including running one kernel per virtual machine and exposing emulated hardware to the VMs. Using KVM allows to run any x86-based operating system (even Windows and BSD) and to configure the operating system as needed. All hardware that is needed by the operating system including main board, hard disks and network interfaces is emulated by KVM. This offers maximal flexibility in choosing and configuring the VM but it also has a higher cost in terms of memory usage and performance reduction. ToMaTo offers both virtualization choices to the users so they can choose the optimal setups for their experiments. For both virtualization solutions ToMaTo offers pre-built virtual machines called templates. The users can choose between various Linux distributions in 32 and 64 bit architectures. For KVM also a pre-installed FreeBSD template is available. Users can also download and upload images of their virtual machines. This can be used for backup purposes, to prepare an experiment first before actually running it or to build images containing a custom operating system that is not in the template list.

To connect the devices, and thus form a network topology, ToMaTo offers different options the user can choose from. The simplest option is the connector type *external network*. This connector simply connects the network interface to an external network, most commonly the Internet. Network configuration is done automatically using DHCP. Using this connector, topologies can use external services, the user can access the exposed devices like servers over the Internet and even other testbed resources can be connected to ToMaTo topologies. The Internet connector does not allow any QoS guarantees and, due to technical reasons, no QoS limitations can be set on this connector type. Other connector types use the Tinc VPN³ which connects devices in private networks that are not connected to each other nor to the Internet. Users can choose between *hub*, *switch* and *router* semantics in

³Tinc VPN: Peer-to-peer VPN (<http://www.tinc-vpn.org>)

this private network. On connections using these connectors, network characteristics like packet loss, delay and bandwidth limitation can be emulated. Additionally users can capture network traffic at these connectors and download it as a *pcap* file to analyze or view the traffic using specialized tools like Wireshark⁴.

Figure 1 shows a topology with four client devices, one server device, two switch connectors and one Internet connector. ToMaTo features an easy-to-use graphical user interface for creating and configuring topologies as well as for accessing the devices. The user interface is web-based and thus is cross-platform and can be used without software installation.

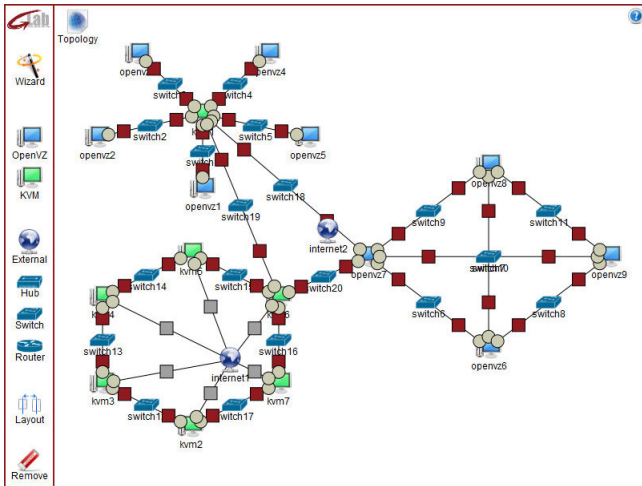


Figure 2. ToMaTo editor with complex topology

To help create complex topologies, ToMaTo features an automatic topology creator wizard. The basic idea of the wizard is to take as much manual work as possible from the user and to automate it. The wizard allows automatic creation of a previously configured number of identical hosts, to arrange these hosts in an ordered way inside the working area and to connect these nodes to each other in a predefined topology structure and with automatically assigned IP-addresses and host names. So far, the wizard can connect the hosts either in a star topology with a central switch or host, in a ring, or fully-meshed, i.e. each host connected to each of the other hosts. With the wizard, it is not only possible to create new hosts and connect them, but to connect (and auto-configure) already existing nodes in the network. Figure 2 illustrates the benefits of this functionality with a complex automatically generated topology.

With ToMaTo, users can choose between OpenVZ and KVM virtualization. This way users can get the level of access that is needed for their experiments and still use as few resources as possible. A modern cluster node can handle up to 250 OpenVZ devices and up to 50 KVM devices, both depending on device usage. The connector components only pose a very small overhead and can handle connections

with over 100 Mbps. ToMaTo hosts use an existing operating system as basis and only need small changes that have been bundled as software package. That means that support and security updates are available and do not have to be provided by the experimental facility administrators. As the ToMaTo back-end only controls the hosts and only contacts them when users change their topologies, the back-end can handle many host nodes making the solution very scalable. ToMaTo can be used to create experimental facilities with distributed hosts. Limitations in network emulation apply since the resulting link characteristics are a combination of real and emulated link properties. ToMaTo offers long-term link statistics so the users can plan their experiments accordingly.

III. DEMONSTRATION

The demonstration will show the basic usage of ToMaTo and its features. In a first step a simple topology will be created using the graphical editor. Then this topology will be used to demonstrate direct console access and the link emulation features. In a next step the traffic capturing will be enabled on one connection and the resulting capture files will be downloaded and analyzed in a graphical tool.

At the end of the demonstration, the automatic topology creator wizard will be shown and used to create a complex topology.

IV. ACKNOWLEDGEMENTS

This work has been done as part of the German Lab research project, funded by the German Federal Ministry of Education and Research (BMBF).

REFERENCES

- [1] Dennis Schwerdel, David Hock, Daniel Günther, Bernd Reuther, Paul Müller, and Phuoc Tran-Gia. ToMaTo - a network experimentation tool. In *7th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom 2011)*, Shanghai, China, April 2011.

⁴Wireshark: Capture analyzer (<http://www.wireshark.org>)