ToMaTo

Topology Management Tool

Dennis Schwerdel

University of Kaiserslautern, Germany
Department of Computer Science
Integrated Communication Systems ICSY
http://www.icsy.de
ToMaTo is a topology-oriented control framework for virtual networking experiments.

- **Control framework**
  - Like Planet-Lab, Emulab, ...
  - Developed in the German-Lab project
  - Open-Source project ([hosted on Github](https://github.com))

- **Virtual networking experiments**
  - Developed for networking experiments
  - E.g. networking research or software testing
  - All parts of the experiment setup are virtual

- **Topology-oriented**
  - Basic abstraction: Network topology
  - Each experiment has its own topology
  - Topologies contain connected elements

- **Advanced features**
  - Direct console access
  - Link emulation
  - Packet capturing
Overview

Intuitive editor
Create your virtual network topology in an intuitive web-based editor. No need to write config files or code.

Packet capturing
Have a look at what goes over the wire. Just activate packet capturing on a link and analyze the traffic in Wireshark.

Virtual networks
Use virtual switches or external networks to connect your topology.

Up and running in seconds
Even very complex topologies can be configured and started in a few seconds.

Link emulation
Configure different properties like latency, packet loss or bandwidth on each link.

Sites worldwide
Create topologies that go around the globe. Select freely from our various sites.

Graphical console
Control your virtual machines directly from a web-based console viewer. Watch your VM boot or control graphical user interfaces.

Different virtualization technologies
Select from full virtualization (KVM) and container virtualization (OpenVZ) for each topology component to save resources.

Pre-configured templates
Select from a wide range of pre-configures VM images spanning from Debian, Ubuntu and other Linux systems to Windows XP.

Integrated monitoring
Observe the resource usage of a whole topology or individual components.
Topology

- **Graphical representation**
  - Icons show element type
  - Colored icons show virtualization technology
  - Link color shows network segments
  - Link style shows link attributes

- **Example**
  - One central server
  - 4 clients, connected with 2 switches
  - Internet connected to server

- **Per Topology**
  - Accounting
  - Permissions
VM Elements

- **KVM**
  - Full virtualization
  - Integrated into Linux Kernel

- **OpenVZ**
  - Container virtualization
  - Added to Linux Kernel via patch

- **Scripts**
  - Programming language virtualization
  - Installed as software

- **Additional elements**
  - Easy to add more
  - Planned: VirtualBox, LXC
Repy scripts

- **Repy**
  - Restricted Python (Sandbox)
  - Technology from Seattle testbed
  - Modified for ToMaTo
  - Functions for receiving and sending raw ethernet packages

```python
packet = tuntap_read("eth0", timeout=None)
ethernet = ethernet_decode(packet)
echo("%s -> %s: %d bytes\n" % (ethernet.src, ethernet.dst, len(packet)))
tuntap_send("eth1", packet)
```

- **Library**
  - Basic protocols implemented: Ethernet, IPv4, TCP, UDP and ICMP
  - Even some higher protocols: DHCP and DNS
  - Examples for: NAT router, DHCP server, DNS server, Switch, ...
  - Can be extended within the language
## VM Elements - Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>KVM</th>
<th>OpenVZ</th>
<th>Repy scripts</th>
</tr>
</thead>
<tbody>
<tr>
<td># per node</td>
<td>~20</td>
<td>~100</td>
<td>~1000</td>
</tr>
<tr>
<td>any x86 OS</td>
<td>✔</td>
<td>☠️</td>
<td>☠️</td>
</tr>
<tr>
<td>Linux OS</td>
<td>✔</td>
<td>✔</td>
<td>☠️</td>
</tr>
<tr>
<td>Kernel space</td>
<td>✔</td>
<td>☠️</td>
<td>(✔)</td>
</tr>
<tr>
<td>Console support</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Mouse/Keyboard input</td>
<td>✔</td>
<td>✔</td>
<td>☠️</td>
</tr>
<tr>
<td>Layer 2 connectivity</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Interface configuration</td>
<td>(✔)</td>
<td>✔</td>
<td>(✔)</td>
</tr>
</tbody>
</table>
Network Elements

- **VPN: Tinc**
  - Full mesh VPN without server
  - Fully contained, virtual network
  - Cross-site layer 2 connectivity
  - Open endpoints allow federation

- **Tunnel: VTun**
  - Layer 2 tunnel over UDP
  - Open endpoints allow federation

- **External networks**
  - Bridge into local network segments
  - E.g. Internet or local research network
### Architecture

- **Hostmanager**
  - Hosts based on Proxmox VE
  - Controls one host,
  - Offers virtualization/network capabilities
  - Controls local topology elements

- **Frontend(s)**
  - Multiple frontends possible
  - Currently: Web-based, CLI

- **Backend**
  - Controls whole topologies
  - Distributes topologies over hosts
  - Applies stitching
  - Delegates management to hosts
  - Monitoring
  - User management
  - Provides XML-RPC interface
Webfrontend - Editor
Webfrontend – Usage Statistics

Resource Usage
Topology #7

CPU Time
2012-11-21 01:00 1.32 CPUs (31.71 h)

Interval: Days

Memory / Diskspace

CPUs used

Traffic

0 Bytes/s
0 Bytes
500 MB
1 GB
1.5 GB
2 GB
2.5 GB
3 GB
0.25 CPUs
0.5 CPUs
0.75 CPUs
1 CPUs
1.25 CPUs
1.5 CPUs
300 KB/s
250 KB/s
200 KB/s
150 KB/s
100 KB/s
50 KB/s
0 Bytes/s

Memory
Diskspace
Traffic
CPU Time

Highcharts.com
Multiple VNC options
- HTML 5
- Java applet
- Client software
**Properties**

- Bandwidth
- Latency
- Jitter
- Packet loss
- Corruption & duplication

**Link Emulation**

<table>
<thead>
<tr>
<th>Enabled</th>
</tr>
</thead>
</table>

**Direction**

| From openvz1.eth0 to tinc_vpn5.tinc_endpoint6 | From tinc_vpn5.tinc_endpoint6 to openvz1.eth0 |

| Bandwidth | 10000 | 10000 | kbit/s |
| Delay     | 0     | 0     | ms     |
| Jitter    | 0     | 0     | ms     |
| Distribution | Uniform ▼ | Uniform ▼ |
| Loss ratio | 0     | 0     | %      |
| Duplication ratio | 0     | 0     | %      |
| Corruption ratio | 0     | 0     | %      |
Packet Capturing

- **Properties**
  - Captures packages on the wire
  - Direct filtering
  - Format: Pcap, (compatible with Wireshark)
  - Two modes: Download, Live capture

- **Cloudshark**
  - Online tool for packet analysis
Executable Archives

- **Archive contents**
  - Software + dependencies
  - Start/Install script

- **Archive actions**
  - Upload: unpacks contents to a folder and runs start script
  - Download: packs folder into archive and transfers it to user
  - Status: displays the status of execution

- **Use cases**
  - Install software packages on VMs
  - Run complete experiments
  - Upload/Download data

- **Experiment lifecycle**
  - Create executable archive
  - Upload archive, run experiment via start script
  - Download archive, contains results
Use case: IGreen

Scenario
- Services for agriculture
- Support for mobile devices
- How does latency affect QoE?

ToMaTo usage
- Simple topology
- Special template with Android emulator
- Usage of link emulation
Use case: Malware Analysis

Scenario
- Analysis of worm
- Focus on network behavior
- Fully contained topology

ToMaTo usage
- Simple topology
- No connection to Internet
- Usage of packet capturing
<table>
<thead>
<tr>
<th>Feature</th>
<th>Planet-Lab</th>
<th>Emulab</th>
<th>Seattle</th>
<th>ToMaTo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple sites</td>
<td>✓</td>
<td>✗</td>
<td>(✓✓)</td>
<td>✓</td>
</tr>
<tr>
<td>Physical hardware access</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>End-System virtualization</td>
<td>✓</td>
<td>(✓✓)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Network virtualization</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Layer 2 access</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Link emulation</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Packet capturing</td>
<td>✗</td>
<td>(✓✓)</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>High traffic (&gt;100 Mbit/s)</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Resource profiles</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>VNC control</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>
ToMaTo Deployment (July 2014)
Cooperations
- GpENI
- NorNet
- GENI-Rack in Victoria

Additional hardware and links
- Multiple OpenFlow-Switches deployed at several sites
- Gigabit link to GENI connected to ToMaTo in Kaiserslautern
- Goal: Federation with European FIRE facilities

Other ToMaTo installations
- Used in Vietnam for research
- Possible installation in China
- Used in Hamburg-Harburg for teaching
## Some Statistics (July 2014)

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host count:</td>
<td>93</td>
</tr>
<tr>
<td>Average availability:</td>
<td>96.47 %</td>
</tr>
<tr>
<td>Average load:</td>
<td>32.84 %</td>
</tr>
<tr>
<td>Total CPUs:</td>
<td>1052</td>
</tr>
<tr>
<td>Total memory:</td>
<td>2.3 TB</td>
</tr>
<tr>
<td>Total disk space:</td>
<td>64.3 TB</td>
</tr>
<tr>
<td>Users:</td>
<td>47</td>
</tr>
<tr>
<td>Topologies:</td>
<td>118</td>
</tr>
<tr>
<td>Elements:</td>
<td>1619</td>
</tr>
<tr>
<td>Connections:</td>
<td>484</td>
</tr>
<tr>
<td>Virtual machines:</td>
<td>56 KVM, 405 OpenVZ, 27 Repy</td>
</tr>
</tbody>
</table>
More Information

- **Websites**
  - ToMaTo project: [http://www.tomato-lab.org](http://www.tomato-lab.org)
  - ToMaTo German-Lab testbed: [http://master.tomato-lab.org](http://master.tomato-lab.org)
  - German-Lab: [http://www.german-lab.de](http://www.german-lab.de)

- **Publications**
  - Future Internet research and experimentation: The G-Lab approach
    *Dennis Schwerdel, Bernd Reuther, Thomas Zinner, Paul Müller and Phuoc Tran-Gia*
    TBP, Computer Networks, Available online 8 January 2014, ISSN 1389-1286
  - ToMaTo a Virtual Research Environment for Large Scale Distributed Systems Research
    *Paul Müller, Dennis Schwerdel and Justin Cappos*
    TBP, PIK - Praxis der Informationsverarbeitung und Kommunikation, Available online 25 January 2014
  - ToMaTo - a network experimentation tool
    *Dennis Schwerdel, David Hock, Daniel Günther, Bernd Reuther, Paul Müller and Phuoc Tran-Gia*
    7th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom 2011), Shanghai, China, April 2011.
  - Future Internet Research and Experimentation: The G-Lab Approach
    *Dennis Schwerdel, Bernd Reuther, Thomas Zinner, Paul Müller and Phuoc Tran-Gia*
    Computer Networks, special issue on FI testbeds, tbp
Dennis Schwerdel
Integrated Communication Systems ICSY
University of Kaiserslautern, Germany
Department of Computer Science
P.O. Box 3049
D-67653 Kaiserslautern

Phone: +49 (0)631 205-26 43
Fax: +49 (0)631 205-30 56
E-Mail: schwerdel@informatik.uni-kl.de