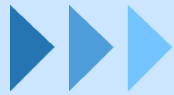


ToMaTo

Topology Management Tool

Dennis Schwerdel

University of Kaiserslautern, Germany
Department of Computer Science
Integrated Communication Systems ICSY
<http://www.icsy.de>



Introduction

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](#))

▶ **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.

Up and running in seconds



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

Sites worldwide



Create topologies that go around the globe, select freely from our various sites.

Packet capturing



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

Link emulation



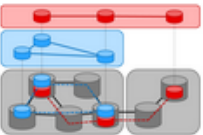
Configure different properties like latency, packet loss or bandwidth on each link.

Graphical console



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

Virtual networks



Use virtual switches or external networks to connect your topology.

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-configured 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

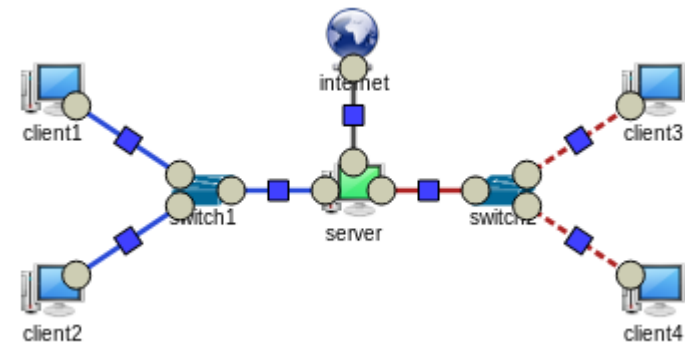
- Icons show element type
- Colored icons show virtualization technology
- Link color shows network segments
- Link style shows link attributes

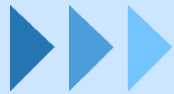
▶ Example

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

▶ Per Topology

- Accounting
- Permissions





- ▶ **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

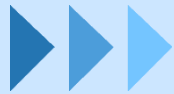
▶ Repy

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

```
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)
```

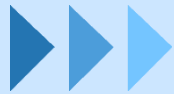
▶ Library

- 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

	KVM	OpenVZ	Repy scripts
# per node	~20	~100	~1000
any x86 OS	✓	✗	✗
Linux OS	✓	✓	✗
Kernel space	✓	✗	(✓)
Console support	✓	✓	✓
Mouse/Keyboard input	✓	✓	✗
Layer 2 connectivity	✓	✓	✓
Interface configuration	(✓)	✓	(✓)

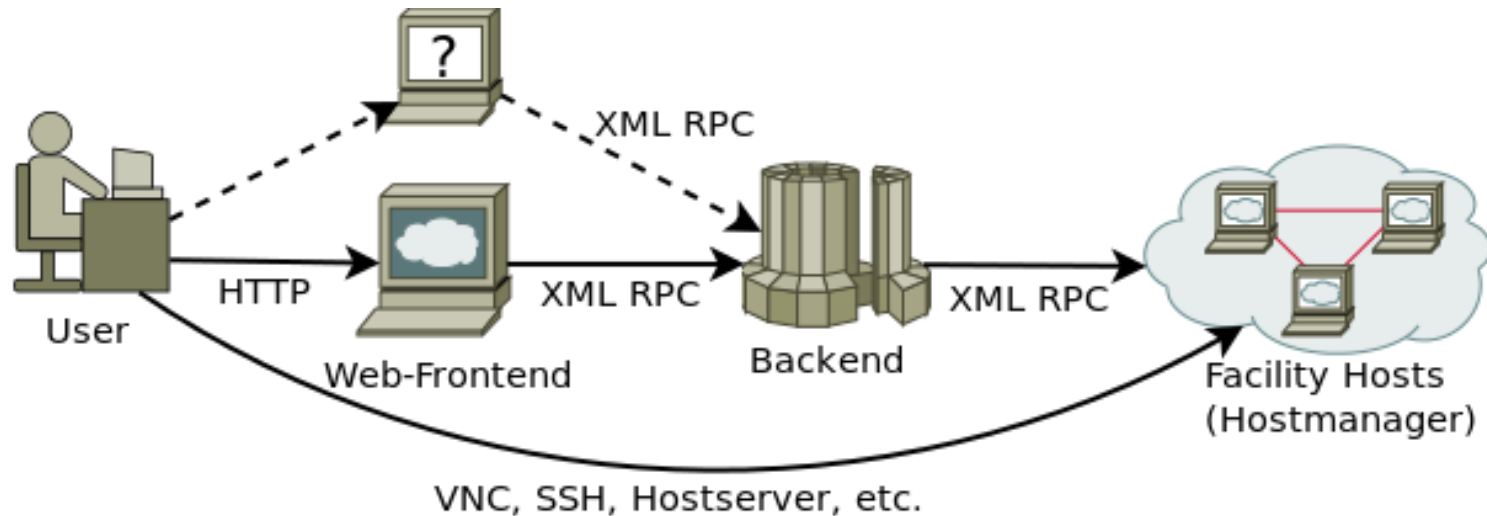


- ▶ **VPN: Tinc**
 - Full mesh VPN without server
 - Fully contained, virtual network
 - Cross-site layer 2 connectivity
 - Open endpoints allow federation

- ▶ **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

Home Devices Network Topology Options

Select & Move

Connect Delete

Start Stop

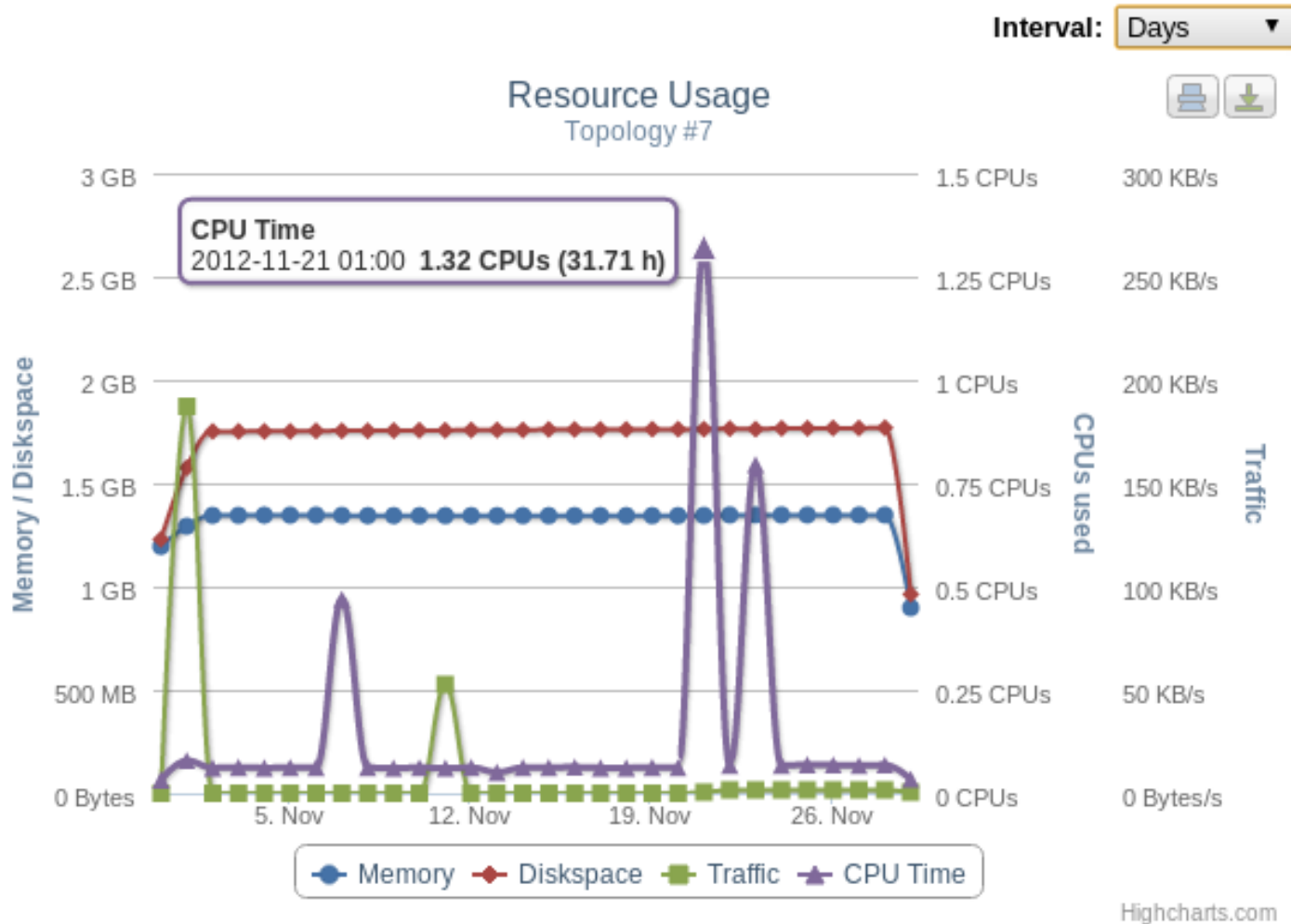
Prepare Destroy Delete

Debian 6.0 (OpenVZ) Debian 6.0 (KVM) Switch Internet

Modes Topology control Common elements

The screenshot displays a network topology editor interface. At the top, there are tabs for 'Home', 'Devices', 'Network', 'Topology', and 'Options'. Below the tabs are three main panels: 'Modes' containing 'Select & Move', 'Connect', and 'Delete'; 'Topology control' containing 'Start', 'Stop', 'Prepare', 'Destroy', and 'Delete'; and 'Common elements' containing 'Debian 6.0 (OpenVZ)', 'Debian 6.0 (KVM)', 'Switch', and 'Internet'. The main workspace shows a network diagram with various elements: 'sw1h5', 'openvz4', 'switch4', 'openvz8', 'sw1h1', 'openvz3', 'internet1', and 'switch3'. A context menu is open over the 'Element openvz2', listing actions: 'Connect', 'Start', 'Destroy', 'Console', 'Resource usage', 'Download image', 'Upload image', and 'Configure'.

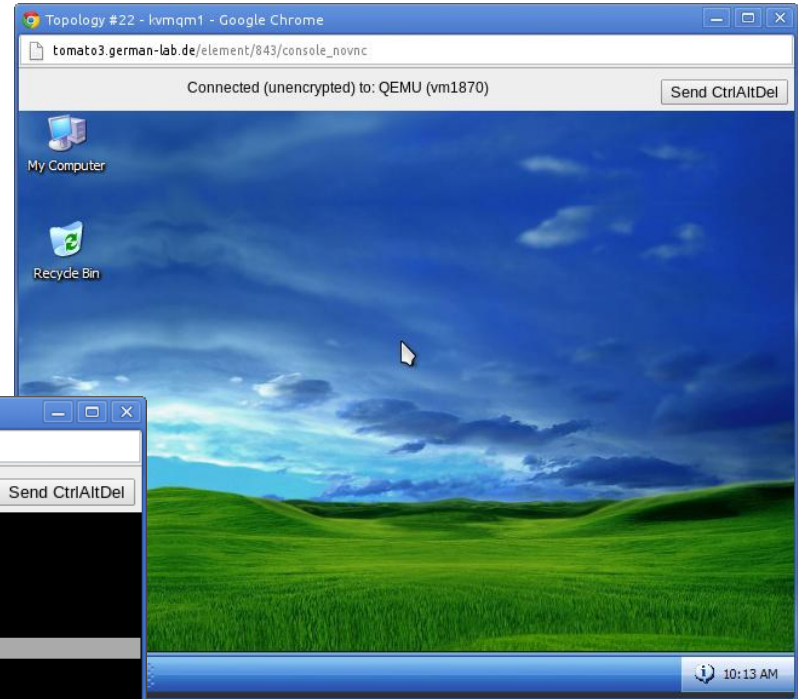
Webfrontend – Usage Statistics



Console Access

Multiple VNC options

- HTML 5
- Java applet
- Client software



Topology #22 - computer1 - Google Chrome
tomato3.german-lab.de/element/827/console_novnc
Connected (unencrypted) to: VNC Command Terminal Send CtrlAltDel

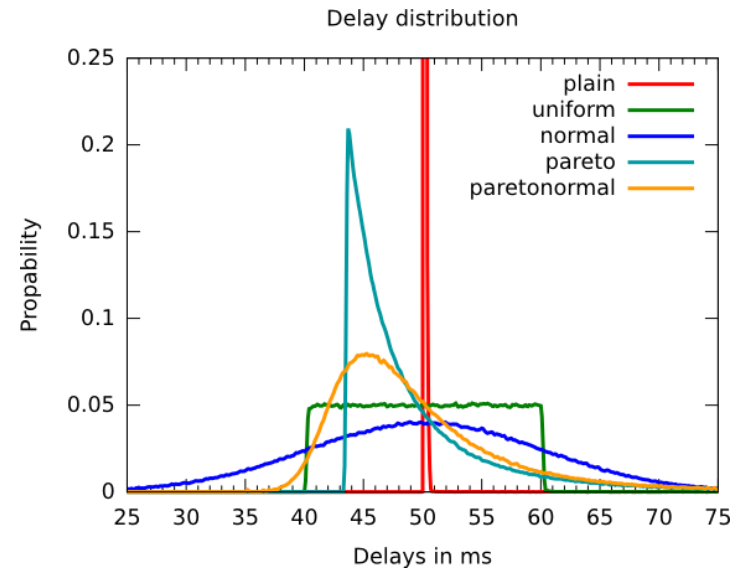
```
top - 11:16:52 up 0 min, 0 users, load average: 0.05, 0.01, 0.00
Tasks: 9 total, 1 running, 8 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0%us, 0.0%sy, 0.0%ni,100.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 524288k total, 9780k used, 514508k free, 0k buffers
Swap: 524288k total, 0k used, 524288k free, 6768k cached
```

PID	USER	PR	NI	UIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	2028	696	600	S	0.0	0.1	0:00.00	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd/1436
3	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khelper/1436
14	root	20	0	104	12	4	S	0.0	0.0	0:00.00	init-logger
205	root	20	0	27280	1340	972	S	0.0	0.3	0:00.00	rsyslogd
231	root	20	0	3780	772	604	S	0.0	0.1	0:00.00	cron
248	root	20	0	18648	624	440	S	0.0	0.1	0:00.00	vzctl
249	root	20	0	4480	1724	1404	S	0.0	0.3	0:00.00	bash
258	root	20	0	2328	1080	892	R	0.0	0.2	0:00.00	top

Link Emulation

▶ Properties

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



Link emulation

Enabled	<input checked="" type="checkbox"/>		
Direction	From openvz1.eth0 to tinc_vpn5.tinc_endpoint6	From tinc_vpn5.tinc_endpoint6 to openvz1.eth0	
Bandwidth	<input type="text" value="10000"/>	<input type="text" value="10000"/>	kbit/s
Delay	<input type="text" value="0"/>	<input type="text" value="0"/>	ms
Jitter	<input type="text" value="0"/>	<input type="text" value="0"/>	ms
Distribution	<input type="text" value="Uniform"/>	<input type="text" value="Uniform"/>	
Loss ratio	<input type="text" value="0"/>	<input type="text" value="0"/>	%
Duplication ratio	<input type="text" value="0"/>	<input type="text" value="0"/>	%
Corruption ratio	<input type="text" value="0"/>	<input type="text" value="0"/>	%

Packet Capturing

▶ Properties

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

Packet capturing

Enabled

Capture mode

For download ▼

Packet filter expression

▶ Cloudshark

- Online tool for packet analysis

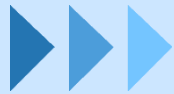
Current File: <http://blog.rootshell.be/wp-content/uploads/2010/06/cloudsharkdemo.pcap> download original (1.1 kb)

CloudShark

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.254.229	88.191.119.130	ICMP	Echo (ping) request
2	0.054183	88.191.119.130	192.168.254.229	ICMP	Echo (ping) reply
3	1.001388	192.168.254.229	88.191.119.130	ICMP	Echo (ping) request
4	1.054820	88.191.119.130	192.168.254.229	ICMP	Echo (ping) reply
5	2.002870	192.168.254.229	88.191.119.130	ICMP	Echo (ping) request
6	2.056771	88.191.119.130	192.168.254.229	ICMP	Echo (ping) reply
7	3.004017	192.168.254.229	88.191.119.130	ICMP	Echo (ping) request
8	3.058191	88.191.119.130	192.168.254.229	ICMP	Echo (ping) reply
9	4.005573	192.168.254.229	88.191.119.130	ICMP	Echo (ping) request
10	4.059409	88.191.119.130	192.168.254.229	ICMP	Echo (ping) reply

```
▶ Frame 1 (98 bytes on wire, 98 bytes captured)
▶ Ethernet II, Src: Dell_ad:4d:aa (00:21:70:ad:4d:aa), Dst: 3Com_17:0c:77 (00:04:76:17:0c:77)
▶ Internet Protocol, Src: 192.168.254.229 (192.168.254.229), Dst: 88.191.119.130 (88.191.119.130)
▼ Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0 ()
  Checksum: 0x8bdc [correct]
  Identifier: 0xf524
  Sequence number: 1 (0x0001)
  Data: 274F224C00000006A8F04000000000101121314151617...
  Length: 56

0000 00 04 76 17 0c 77 00 21 70 ad 4d aa 08 00 45 00  .v.v!p.M...E.
0010 00 54 00 00 40 00 40 01 aa d9 c0 a8 fe e5 58 bf  .T.e@.....X.
0020 77 82 00 00 00 00 00 00 00 00 00 00 00 00 00  w.
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .
```



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**

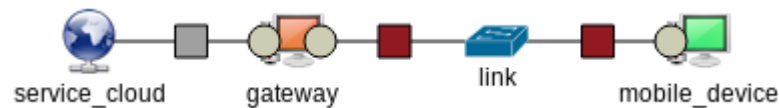
- Install software packages on VMs
- Run complete experiments
- Upload/Download data

▶ **Experiment lifecycle**

- Create executable archive
- Upload archive, run experiment via start script
- Download archive, contains results

► Scenario

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



► ToMaTo usage

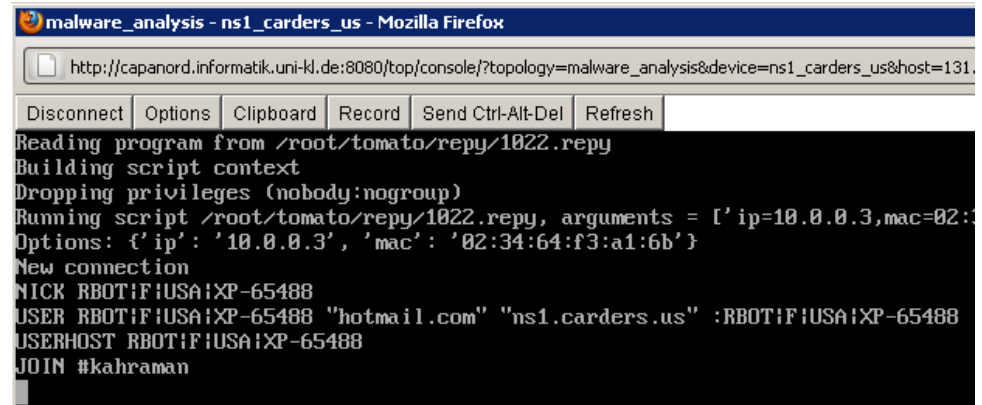
- Simple topology
- Special template with Android emulator
- Usage of link emulation



Use case: Malware Analysis

► Scenario

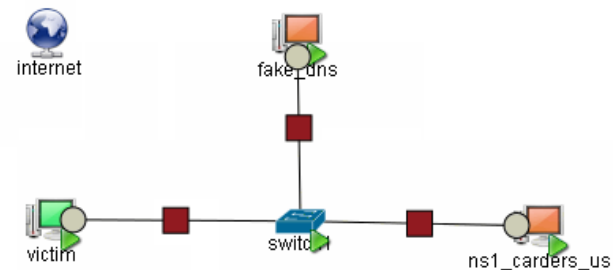
- Analysis of worm
- Focus on network behavior
- Fully contained topology

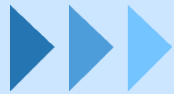


```
malware_analysis - ns1_cadders_us - Mozilla Firefox
http://capanord.informatik.uni-kl.de:8080/top/console/?topology=malware_analysis&device=ns1_cadders_us&host=131.
Disconnect Options Clipboard Record Send Ctrl-Alt-Del Refresh
Reading program from /root/tomato/repy/1022.repy
Building script context
Dropping privileges (nobody:nogroup)
Running script /root/tomato/repy/1022.repy, arguments = ['ip=10.0.0.3,mac=02:34:64:f3:a1:6b']
Options: {'ip': '10.0.0.3', 'mac': '02:34:64:f3:a1:6b'}
New connection
NICK RBOT!F!USA!XP-65488
USER RBOT!F!USA!XP-65488 "hotmail.com" "ns1.cadders.us" :RBOT!F!USA!XP-65488
USERHOST RBOT!F!USA!XP-65488
JOIN #kahraman
```

► ToMaTo usage

- Simple topology
- No connection to Internet
- Usage of packet capturing





Framework Comparison

	Planet-Lab	Emulab	Seattle	ToMaTo
Multiple sites			()	
Physical hardware access				
End-System virtualization		()		
Network virtualization				
Layer 2 access				
Link emulation				
Packet capturing		()		
High traffic (>100 Mbit/s)				
Resource profiles				
VNC control				

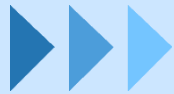
ToMaTo Deployment (July 2014)



University of Victoria

Technische Universität München
[**simula** . research laboratory]

NORTHWESTERN UNIVERSITY



▶ **Cooperations**

- GpENI
- NorNet
- GENI-Rack in Victoria

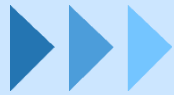
The logo for GpENI, featuring the letters 'GpENI' in a stylized font. The 'G' is large and bold, with 'pENI' in a smaller font to its right.The logo for NORNET, featuring the letters 'NORNET' in a bold, sans-serif font. The 'O' is replaced by a globe icon with a green signal wave emanating from it.

▶ **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)

Host count:	93
Average availability:	96.47 %
Average load:	32.84 %
Total CPUs:	1052
Total memory:	2.3 TB
Total disk space:	64.3 TB

Users:	47
Topologies:	118
Elements:	1619
Connections:	484
Virtual machines:	56 KVM, 405 OpenVZ, 27 Repy

► Websites

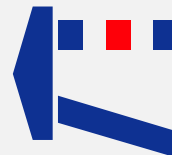
- ToMaTo project: <http://www.tomato-lab.org>
- ToMaTo German-Lab testbed: <http://master.tomato-lab.org>
- German-Lab: <http://www.german-lab.de>

► 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 Phouc Tran-Gia
Computer Networks, special issue on FI testbeds, tbp



TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN



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