a testbed for distributed systems research based on the topology oriented infrastructure: 

Tomato

tomato-lab.org

The 3rd International NorNet Users Workshop (NNUW-3)
August 28, 2015
Simula Research Laboratory
Fornebu/Norway
BMBF project 2008 – 2012
Closing the loop between research and real-world experiments
Provide an experimental facility for studies on architectures, mechanisms, protocols and applications towards Future Internet
Research projects as well as the experimental facility are controlled by the same community
Investigate interdependency of theoretical studies and prototype development
Project G-Lab: The studies

- Architecture and Composition of Services (ASP 6)
- Routing and Address Schemes (ASP 2)
- Wireless Networks and Mobility (ASP 3)
- Quality of Service and Security (ASP 5)
- Monitoring and Management Concepts (ASP 4)
- Architecture of the Future Internet (ASP 1)
- Coordination and Consolidation (ASP 0)

Experimental Facility

Other projects
Project G-Lab: The Lab

- Full control over the resources
  - Reservation of single resource should be possible
  - Elimination of side effects
  - Testing scalability

- Exclusive resource reservation
  - Testing QoS / QoE
  - Decentralized Resources can be independently used
  - Tests on the lower layers of the network without affecting the “life” network

- Extended functionality
  - New technologies (Wireless, Sensor,…)
  - Interfaces to other testbeds (GENI, PlanetLab Japan, WinLab,…)
  - OpenFlow setup

TUB  TU Berlin
TUD  TU Darmstadt
TUKL  TU Kaiserslautern
TUM  TU München
UKA  University Karlsruhe KIT
UWUE  University Wurzburg
Kind of experiments

- **Access layer experiments**
  - Consider lower layers and hardware
    - Example: Mobile handover
  - Requirements
    - Hardware access
    - Custom operating systems (Realtime)
    - Heterogeneous access technologies (3G, Wifi, etc.)

- **Network layer experiments**
  - Focus on TCP/IP suite
    - Example: IPv6 extensions, TCP substitutes, new architectures
  - Requirements
    - Deep OS access (modified kernels, etc.)
    - Small but complex topologies, link emulation

- **Algorithm/Protocol experiments**
  - Work on top of network layer
    - Example: P2P-Networks
  - Requirements
    - Huge but simple topologies
    - Link emulation
    - No hardware or OS access

- **Legacy software experiments**
  - Considers legacy software
    - „Legacy software“ refers to any widespread software with undocumented or unpublished behavior
    - Example: Skype and Windows
  - Requirements
    - Special environments, custom operating systems
    - Small but complex topologies
    - Link emulation and external packet capturing
- How many tests/experiments can I run in parallel?

- **emulab** at University of Utah offers systems which can be used only for one experiment in time

- **Seattle** as a Python dialect can run >1000 experiments at a time

- **Tomato** offers a mixture depending on the used virtualization technology
MaTo - A network experimentation tool

MaTo „Topology Management Tool“

Topology contains

- **Elements** producing, processing and consuming data
  - Virtual machines that can run software (KVM and OpenVZ)
  - Scripted elements written in Python
  - Networking elements: switches (VPN-based and SDN), external networks
- **Connections** forwarding data
  - Packet capturing
  - Link emulation

1: http://www.tomato-lab.org
**Tomato – Editor and Features**

- **Graphical editor**
  - Creating topologies by drag/drop
  - Connects topologies
  - Resource usage per topology

- **Topologies**
  - Colored icons show virtualization technology (KVM, OpenVZ, …)
  - Linux and Windows OS
  - Link style shows link attributes
  - Complex topologies (multi-homing)

- **On link basis properties**
  - Bandwidth, Latency, Jitter, …
  - Packet capturing (Cloudshark, Wireshark)

- **Console access**
  - HTML5, Java applet
Dynamic deployment of topologies/experiments

- One can create multiple independent complex topologies.
- For each host one can choose different OS (Windows or Linux) and on which site this host should run.
- Each topology is self contained and not accessible from the open Internet.
  - To do so an external network (the Internet) must explicitly connected to the topology.
  - Access to the hosts only by console access.
- After finishing the design of the topology/experiment one can deploy it to the designated sites.
- Then one can run the topology/experiment, control the resource usage of the whole topology and the link properties.
- After finishing the experiment one can destroy the topology and release the hosts.
The Architecture

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

Backend
- Controls whole topologies
- Distributes topologies over hosts
- Applies stitching
- Monitoring
- XML/RPC interface
- User management

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

Software-defined Testbed (APIs)

Virtualized resources

Fixed resources

Paul Mueller, University of Kaiserslautern
The architecture

Software-defined Testbed

Virtualized Resources

Fixed Resources

Dynamic Resources
Application Areas

Access layer experiments
- Requirements
  - Hardware access
  - Custom operating systems (Realtime)
  - Heterogeneous access technologies (3G, Wifi, etc.)
  - Needs specialized testbeds therefore NO support by Tomato
  - But federation possible

Network layer experiments
- Requirements
  - Deep OS access (modified kernels, etc.)
  - Small but complex topologies, link emulation
  - Tomato offers
    - Full kernel access via KVM
    - Complex topologies
    - Link emulation
    - Packet capturing (for analysis)
    - Easy setup of topologies

Algorithm/Protocol experiments
- Requirements
  - Huge but simple topologies
  - Link emulation
  - No hardware or OS access
  - Tomato offers
    - Lightweight virtualization with OpenVZ
    - Link emulation
    - Federation with other testbeds via Internet

Legacy software experiments
- Requirements
  - Special environments, custom operating systems
  - Small but complex topologies
  - Link emulation and external packet capturing
  - Tomato offers
    - Custom operating systems with KVM (Windows)
    - Access to external service via Internet connector
    - Packet capturing independent of guest OS

Paul Mueller, University of Kaiserslautern
Scaling up **TOMaTo**

- **Single-host deployment**
  - Hostmanager, Backend and Web-Frontend can run on the same host
  - Easy for local tests

- **Isolated multi-host setups**
  - Running multiple hosts with a single backend and web-frontend on user premises
  - Isolated infrastructure for SMEs

- **Federated setups**
  - **TOMaTo** hosts can be used by multiple backends
  - The **TOMaTo** community consists of over 100 hosts at several sites

- **Testbed on demand**
  - Dynamically allocate cloud resources for experiments
  - Current research effort
    - Master thesis on allocating resources from CloudLab for
    - Master theses on allocating resources from AWS
    - Bachelor thesis on dynamic host (VMs) allocation
Worldwide sites

- Victoria/CA
- Wisconsin
- Kansas
- Chicago
- Urbana-Champaign
- Rutgers
- Clemson
- Madrid
- NorNet
- G-Lab

Partner institutions:
- tomato-lab.org
- Northwestern University
- KU Kansas University
- University of Utah
- University of Florida
- UCL Norway
- University of Arizona
- University of California, San Diego
- TUI Munich
- University of Munich
- University of Keele
- University of Edinburgh
- University of Dundee
- University of Cambridge
- University of Oxford
The G-Lab approach combined the responsibility between projects and experimental facility

Current testbeds (like PlanetLab, …) does not fulfill the requirements of the projects (especially non CS)

A new testbed software was developed

- Tomato was the answer to the project requirements
- Tomato is independent of the underlying physical infrastructure

Sustainability

- Old G-Lab hardware will fade out and will be replaced by hardware required and offered from new projects
- Tomato is used in classrooms and new projects
  - Teaching: Hamburg, Kaiserslautern, Darmstadt, …
  - Projects: MAKI TU Darmstadt (SFB)

Screencast: [http://tomato-lab.org/screencasts/basic/](http://tomato-lab.org/screencasts/basic/)
Questions?
Literature


- Paul Müller, Dennis Schwerdel and Justin Cappos, *ToMaTo a Virtual Research Environment for Large Scale Distributed Systems Research*, PIK - Praxis der Informationsverarbeitung und Kommunikation, 2014.

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