Simtools

Jose L. Muñoz, Oscar Esparza, Juanjo Alins, Jorge Mata
Telematics Engineering
Universitat Politècnica de Catalunya (UPC)

Outline

1 Simtools
Introduction
Install
Scenarios
Simctl
Troubleshooting & Customization
Capturing Traffic

Simtools

- Simtools is a tool for creating and running virtual environments of guests that use User Mode Linux (UML) kernels.
- UML is a type of paravirtualization to run Linux guests over an unmodified Linux kernel that serves as hypervisor.
- Simtools provides several preconfigured scenarios using a modified version of a tool called Virtual Network User Mode Linux (VNUML).
- VNUML allows us to easily define and run virtual networks using UML Kernels. In particular, VNUML provides us a language and a parser to create these scenarios.
- We will explain how to install simtools and how its main scripts and configuration files work.

Outline

1 Simtools
Introduction
Install
Scenarios
Simctl
Troubleshooting & Customization
Capturing Traffic

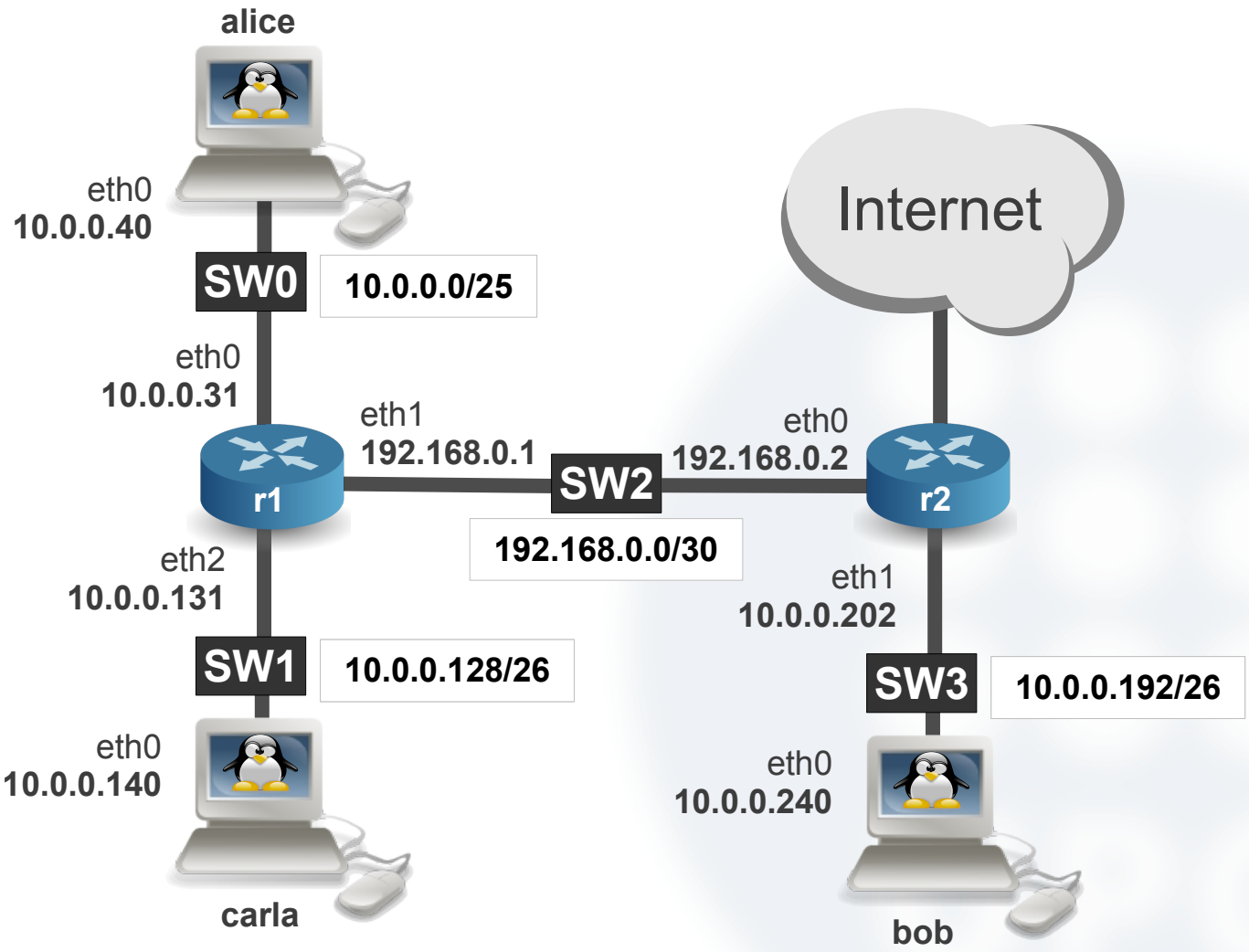
Install & Update Simtools

- The instructions to install the software are in **http://simtools.upc.edu**.
- The installation has been tested using the **32-bit version** and **64-bit version** of Ubuntu 12.04/14.04 and Debian 7.
- To run the system you can:
  A. Install our tools in your Linux (Ubuntu) system.
  B. Create a bootable USB with our system and boot from this device (using the BIOS boot menu).
  C. Use virtualbox and our OVA file.
  Important: for Virtualbox you have to activate the hardware virtualization in your BIOS to run our software fluently.
- Visit http://simtools.upc.edu for further information.

Outline

1 Simtools
Introduction
Install
Scenarios
Simctl
Troubleshooting & Customization
Capturing Traffic

Example Topology



Scenario's VNUML File


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE vnuml SYSTEM "/usr/share/xml/vnuml/vnuml.dtd">
<vnuml>
  <global>
    <version>1.8</version>
    <simulation_name>ip-routing-abc</simulation_name>
    <automac/>
    <vm_mgmt type="none" />
    <vm_defaults exec_mode="mconsole">
      <filesystem type="cow">/usr/share/vnuml/filesystems/root_fs_tutorial</filesystem>
      <kernel>/usr/share/vnuml/kernels/linux</kernel>
      <console id="0">pts</console>
    </vm_defaults>
  </global>
  <net name="Net0" mode="uml_switch" hub="yes" sock="/var/run/vnuml/Net0.ctl" />
  <net name="Net1" mode="uml_switch" hub="yes" sock="/var/run/vnuml/Net1.ctl" />
  <net name="Net2" mode="uml_switch" hub="yes" sock="/var/run/vnuml/Net2.ctl" />
  <net name="Net3" mode="uml_switch" hub="yes" sock="/var/run/vnuml/Net3.ctl" />
  <vm name="alice">
    <if id="0" net="Net0"></if>
  </vm>
  <vm name="r1">
    <console id="1">pts</console>
    <if id="0" net="Net0"></if>
    <if id="1" net="Net2"></if>
    <if id="2" net="Net1"></if>
    <forwarding type="ip" />
  </vm>
  ...
</vnuml>
```

Locating Scenarios

- `simctl` searches .vnuml files by default in **/usr/share/vnuml/scenarios**.
- Our scenarios installed by default are in /usr/share/vnuml/scenarios.
- You can use DIRPRACT for other directories:

```
phyhost$ export DIRPRACT=.
```

- With the previous command `simctl` will search scenarios in the current directory.

Outline

1 Simtools
Introduction
Install
Scenarios
Simctl
Troubleshooting & Customization
Capturing Traffic

Start a Scenario

- To start a particular scenario, you must execute `simctl` in the host with the name of the selected scenario and use the start option. Example:

```
phyhost$ simctl ip-routing-abc start
start
 name="tap0" mode="uml_switch" hub="yes" sock="/var/run/vnuml/Net0.ctl"
 name="tap1" mode="uml_switch" hub="yes" sock="/var/run/vnuml/Net1.ctl"
 name="tap2" mode="uml_switch" hub="yes" sock="/var/run/vnuml/Net2.ctl"
 name="tap3" mode="uml_switch" hub="yes" sock="/var/run/vnuml/Net3.ctl"

Set 'tap0' persistent and owned by uid 1000
Set 'tap1' persistent and owned by uid 1000
Set 'tap2' persistent and owned by uid 1000
Set 'tap3' persistent and owned by uid 1000
host> /usr/bin/touch /home/user/.vnuml/LOCK
....

Total time elapsed: 77 seconds
phyhost$
```

- Be patient because it might take some time to complete the starting process[1].
- Finally, the command ends indicating the time taken to start the scenario and we get the console prompt again.

[1] Starting an scenario might take up to several minutes.

Access to VMs I

- In the hypervisor you will have special interfaces to capture traffic between Virtual Machines (VMs) on each network.
- You can use the `vms` option to view the VMs of an scenario:

```
phyhost$ simctl ip-routing-abc vms
Virtual machines from ip-routing-abc:
num     vms       enabled tty's Id
  1     alice     0
  2     r1        0 1
  3     r2        0
  4     bob       0
  5     carla     0
```

- In this scenario, we can see that all the VMs have a single console (console 0) except **r1**, which has two consoles enabled (consoles 0 and 1).

Access to VMs II

- You can get a console of a VM with the `get` option:

```
phyhost$ simctl ip-routing-abc get
    alice    Running    --------
    r1       Running    --------
    r2       Running    --------
    bob      Running    --------
    carla    Running    --------
phyhost$ simctl ip-routing-abc get alice
phyhost$ simctl ip-routing-abc get r1
phyhost$ simctl ip-routing-abc get r1 1
phyhost$ simctl ip-routing-abc get r2 0
```

- You will observe that a terminal appears in your system each time you execute the `get` command for a VM.
- To login as root in a VM type:
  - **ENTER**.
  - Use the **root** user with **xxxx** as password.

Stop a Scenario

- When you wish to stop the simulation, you can type the following:

```
phyhost$ simctl ip-routing-abc stop
host> /usr/bin/touch /home/user/.vnuml/LOCK
host> /bin/rm -f ~/.vnuml/simulations/ip-routing-abc/vms/alice/status
...
---------- Waiting until UML extinction ----------
waiting on processes 6719 7482 8583 9351 10288...
15 seconds elapsed...
host> /bin/rm -f /home/user/.vnuml/networks/Net0.ctl.cter
....
Total time elapsed: 15 seconds
stop
 name="tap0" mode="uml_switch" hub="yes" sock="/var/run/vnuml/Net0.ctl"
 name="tap1" mode="uml_switch" hub="yes" sock="/var/run/vnuml/Net1.ctl"
 name="tap2" mode="uml_switch" hub="yes" sock="/var/run/vnuml/Net2.ctl"
 name="tap3" mode="uml_switch" hub="yes" sock="/var/run/vnuml/Net3.ctl"
Set 'tap0' nonpersistent
Set 'tap1' nonpersistent
Set 'tap2' nonpersistent
Set 'tap3' nonpersistent
phyhost$
```

Outline

Troubleshooting I

- A virtual machine might spent some time while booting. Then, it might appear a message telling us to retry, continue or abort. **Type always retry (r)**.
- If a simulation never starts or stops, to clear the system, type CRL+c and then:

```
phyhost$ simctl simulation_name stop
phyhost$ simctl forcestop
```

  Finally, reboot the physical host.
- You should **never run two different simulations at the same time.** Stop a simulation before running a new one.

Troubleshooting II

- If by mistake you start two simulations type:

```
phyhost$ simctl simulation_name1 stop
phyhost$ simctl simulation_name2 stop
phyhost$ simctl forcestop
```

- You should **never use the superuser ”root“ to execute `simctl`!!!**
- If by mistake you start a simulation with the root user, you must clear the system and start it again using your user:

```
phyhost$ sudo -s
phyhost# simctl simulation_name stop
phyhost# simctl forcestop
phyhost# exit
```

Finally, reboot the physical host.

Terminal Customization

- Copy the file `/usr/local/etc/simrc` to `~/.simrc`.
- With this file you can define to use `gnome-terminal` instead of `xterm` setting the TERM_TYPE variable:

```
# simrc: tunning of environment variables for simctl
# Definition of scenario files directory
DIRPRACT=/usr/share/vnuml/scenarios
# Definition of VNUML working directory
# (default to $HOME/.vnuml)
# VNUMLWORKDIR=/tmp/$USER
# Definiton of the user which launch the
# switch software
TAPUSER=$USER
# Change the default terminal type (xterm)
# values: (gnome | kde | local)
TERM_TYPE=gnome
# XTERM_FG=white
# XTERM_BG=black
```

- Then, with a `gnome-terminal` you can create a profile called **vnuml** to define colors, font sizes, etc.

Problems and Workarounds

- **Problem**. The UML Kernel is not aware of terminal size.
  - The consequence of this is that when you resize the terminal, the size of the terminal is not refreshed.
  - **Workaround**. Once in a virtual machine terminal and when the terminal size is modified, keypress Ctrl+a f.
- **Problem**. There is a lack of terminal scrolling.
  - We use `screen`, which has a scrollback history buffer for each virtual terminal of 100 lines.
  - **Workaround**. To enter screen into scrolback mode press Ctrl+a ESC.
    To exit scrollback mode press ESC.

Outline

1 Simtools
Introduction
Install
Scenarios
Simctl
Troubleshooting & Customization
Capturing Traffic

Capturing Traffic

- Enable your user for capturing traffic in promiscuous mode:

```
$ sudo chmod +s /usr/bin/dumpcap
```

- `simctl` automatically creates a `tap` interface in the **phyhost** for each virtual network.
- These interfaces are called `SimNet0`,`SimNet1`, etc.
- Simtools also provides a tool called `simtools-captap` to automatically start wireshark protocol analyzers in several `SimNetX` interfaces.
- The syntax is:

```
phyhost$ simtools-captap
phyhost$ simtools-captap -k  to killall wiresharks
phyhost$ simtools-captap -s 7 to start capturing from SimNet0 to SimNet7
phyhost$ simtools-captap -r 7 to restart capturing from SimNet0 to SimNet7
```