AIMS 2014

Labs

Information for Participants

Labs Co-Chairs

Jeroen Famaey jeroen.famaey@intec.ugent.be
Petr Velan velan@ics.muni.cz

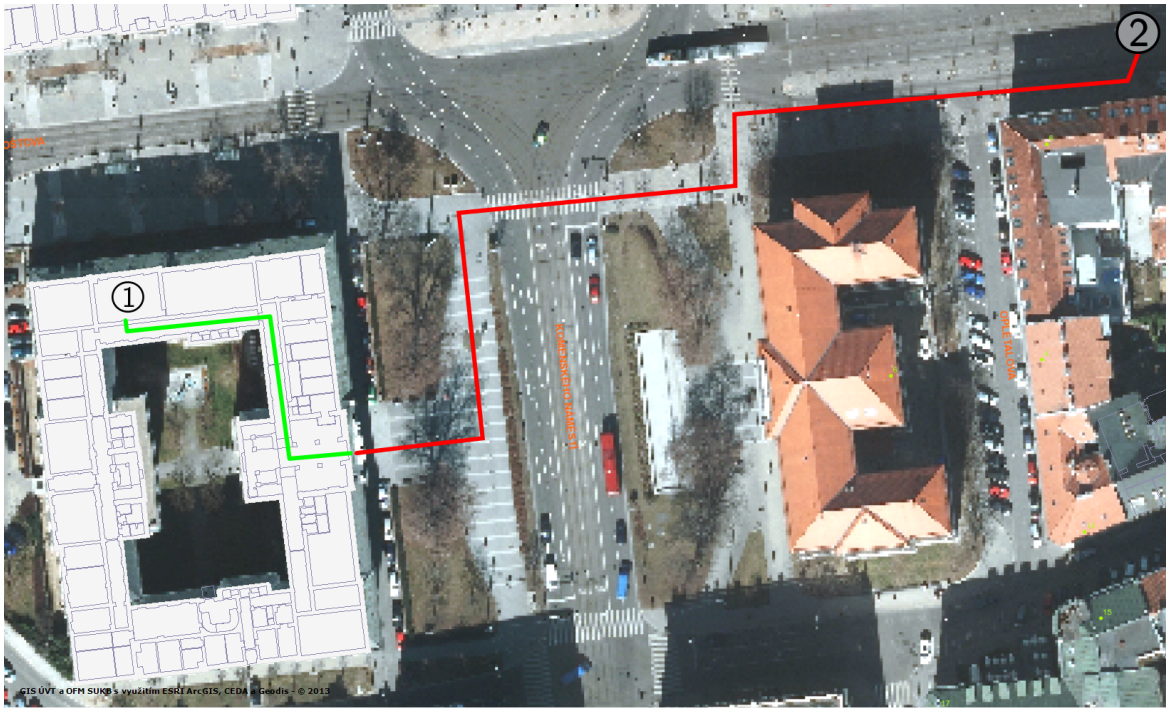
June 30 - July 3, 2014, Brno, Czech Republic

Contents

General Information

Labs Venue

Labs sessions are held in University Computer Center in Study room 3 (Door no. 054) on the 1st floor on Komenského nám. 220/2. The map illustrates route from Česká tram stop to the building and route on the 1st floor of the building.



① Labs Venue - 1st Floor ② Česká Tram Stop

Labs computers

There are 32 personal computers with Windows 7 OS available for the labs participants. Guest accounts will be available for these computers.

Wireless connection

The labs room is well covered with eduroam[1] network, which can be used to connect mobile devices and notebooks.

Contact

If you have any questions regarding the lab sessions, please contact the labs co-chairs. For questions related to individual labs contact the instructors.

[1] https://www.eduroam.org/

Lab 1: Fast Network Simulation Setup

Date: Tuesday 1. 7. 2014
Instructor: Lorenzo Saino <l.saino@ucl.ac.uk>

Arguably, one of the most cumbersome tasks required to run a network experiment is the setup of a complete scenario and its implementation in the target simulator or emulator. This process includes selecting an appropriate topology, provision nodes and links with all required parameters and, finally, configure traffic sources or generate traffic matrices.

Executing all these tasks manually is both time-consuming and error-prone. The Fast Network Simulation Setup (FNSS) toolchain addresses this problem by allowing users to generate even complex experiment scenarios with few lines of Python code and deploy them in the preferred target simulator. FNSS currently supports ns-2, ns-3, mininet as well as custom-built C++, Java and Python simulators.

The lab is divided in three parts. In the first part, participants will be familiarized with various models and datasets of networks topologies. They will also learn the most commonly used models to assign link capacities, delays and buffer sizes and how to synthetically generate realistic traffic matrices. The second part will provide an overview of the FNSS toolchain. Participants will learn the how install and configure it and will be walked through its main features. Finally, in the third part, participants will learn through live coding examples how to easily generate complex simulation scenarios and deploy them on a number of different simulators or emulators.

Session outline

- Modelling networks and traffic
  - Network topology models and datasets
  - Assignment of link capacities, delays, weights and buffer sizes
  - Modelling traffic matrices
- Overview of FNSS
  - Installation and configuration
  - Architecture and features overview
- Live coding examples
  - Create complete simulation scenarios
  - Deploy topologies on mininet platform
  - Deploy topologies and traffic matrices on custom-built simulators

Required software

- Download and install VirtualBox (https://www.virtualbox.org/)
- Download virtual machine image with necessary software from http://fnss.github.io/ (the image will be made available 6 weeks before the start of the conference)

Lab 2: Deploying OpenFlow experiments on the Virtual Wall testbed

Date: Wednesday 2. 7. 2014
Instructors: Niels Bouten <nbouten@intec.ugent.be>
Maxim Claeys <maxim.claeys@intec.ugent.be>
Jeroen Famaey <jeroen.famaey@intec.ugent.be>

Software-defined networking (SDN) greatly increases network management flexibility by decoupling decision making (i.e., control plane) from traffic forwarding (i.e., data plane) in network equipment. This enables network control to become directly programmable, and allows intelligent software components to dynamically reconfigure the network based on service requirements and network conditions. OpenFlow is without a doubt the most widely known implementation of the SDN concept. It is a protocol which structures the communication between the network's data and control plane and provides granular traffic control.

The goal of this hands-on tutorial is to familiarize the participant with the concept of SDN in general and with OpenFlow in particular. We will explore OpenFlow's capabilities to dynamically reroute traffic, guarantee bandwidth, and differentiate flows. Participants will be given the opportunity to apply their acquired knowledge by setting up an OpenFlow-based experiment that guarantees the Quality of Service requirements of a networked video application. The experiment will be run in a live network setting, facilitated by the Virtual Wall testbed.

The Virtual Wall is a testbed facility for setting up large-scale network topologies. The Virtual Wall nodes can be assigned different functionalities and organised in arbitrary network topologies on the fly. As such, it is a generic experimental environment for advanced network, distributed software and service evaluation, and supports scalability research. The facility has been made available to the research community through different FP7 FIRE projects. The tutorial will provide a brief theoretical introduction about the Virtual Wall's capabilities in preparation of the hands-on part.

Session outline

- SDN and OpenFlow (60 min)
  - General introduction to the SDN concept
  - The OpenFlow protocol and architecture
  - Routing and differentiated-service capabilities of OpenFlow
- The Virtual Wall testbed (20 min)
  - General introduction to large-scale network testbeds and Emulab
  - Overview of the Virtual Wall's functionality
- Hands-on OpenFlow experiment (90 min)
  - Configuring a network topology
  - Installing and configuring OpenFlow
  - Running a QoS-differentiation experiment

Required software

- Download and install the latest release of Java 7 JRE from http://www.java.com. Note that Java 8 will not work.

- Test if your Java 7 browser plugin is working correctly by navigating your browser to http://www.java.com/verify. Note that Chrome does not support Java 7 on Mac OS. It is advised to use Safari or Firefox instead in this case.

Lab 3: Cybernetic Proving Ground: A Cloud-based Security Research Testbed

Date: Thursday 3. 7. 2014
Instructors: Jakub Čegan <cegan@ics.muni.cz>
Martin Vizváry <vizvary@ics.muni.cz>
Michal Procházka <michalp@ics.muni.cz>

Cyber attacks have become ubiquitous and in order to face current threats it is important to understand them. However, studying these attacks in a real environment is not often viable. Therefore, it is necessary to find other methods of examining the nature of the attacks. This tutorial will present Cybernetic Proving Ground (CPG) that is being developed at Masaryk University. The CPG is a cloud based framework that allows users to instantiate and run miscellaneous security and forensic scenarios.

The CPG provides a generic way to simulate and study a wide range of cyber attacks. It facilitates an establishment of isolated virtual environments that researchers can use to pursue controlled analysis of the attacks. Using virtualization and clouds, we managed to provide an environment, where it is possible to configure any common network configuration. Therefore, we are able to fulfill needs of many types of security scenarios. The user can use the CPG to set up isolated environments very quickly without the necessity of knowing details about network configuration or deploying auxiliary services such as a monitoring infrastructure.

The tutorial is divided in three parts. In the first part of the tutorial, participants will learn how to access the CPG infrastructure and how to configure a scenario. The second part of the tutorial will focus on running a security scenario. The participants will take part in the scenario as each of them will have a machine to control. An overall status of the CPG scenario will be monitored in the course of the simulation. We will show how to use CPG to easily generate network scenarios, deploy them to simulate and evaluate experiments in a large cloud-based environment.

Session outline

- Cybernetic Proving Ground demonstration (60 min.)
  - CPG overview - presentation
  - Preparing DDoS attack scenario infrastructure - presentation
  - DDoS attack scenario - live demo
  - Q&A
- Penetration testing scenario - hands-on demo (120 min.)
  - Scenario infrastructure deployment
  - Penetration testing (3 levels)
  - Scenario evaluation

Required software

- Web browser
- SSH client (Linux console, Windows PuTTY)