Click Installation

Installation of Click Modular Router

Bart Braem Johan Bergs

University of Antwerp
iMinds - MOSAIC Research Group

October 2014 - Click 2.0.1

Requirements

- Unix system: Linux works in userlevel and kernelspace, FreeBSD has a port (only userlevel works)
- A recent GNU C++ compiler $>=$ 3.0
- GNU Make
- For user-level Click: x86 or x86_64
- Not sure of the requirements? The ./configure script will tell you

Download and configure

- Version 2.0.1 is used: `http://www.read.cs.ucla.edu/click/click-2.0.1.tar.gz`
- For bleeding-edge development: git
- Untar Click in some directory:
  `tar xzf click-\clickversion.tar.gz`
- Configure it, disabling the kernel module and enable your own local elements:

```
cd click-\clickversion
./configure --disable-linuxmodule --enable-local
    --enable-etherswitch
```

Make and install

- Make Click (fast), inside the click-2.0.1 directory: `make -j2`
- Click will now run: `userlevel/click conf/test.click`
- Output should be five times

  ```
  ok:    40 | 45000028 00000000 401177c3 01000001
  02000002 13691369
  ```
- Do not run make install, every time you change elements you must do a make install, as root!

Calling handlers

Start click

```
click -p <port_nr> <click_script>
click -p 10000 somescript.click
```

Connect to click with telnet

```
telnet localhost 10000
quit
read list // displays all elements
```

See SocketHandler element (automatically added)

```
read <elementname>.<handlername>
read rt.table
write <elementname>.<handlername> <values>
write arptable.insert 00:50:BA:85:84:B1 10.0.1.2
```

Adding own elements

All elements are stored in /elements/ directory

- Yours should be put in elements/local
- Put the .hh and .cc files there

To make those elements available:

```
make elemlist
make
```

Notice new elements being compiled, solve any compilation problems and use your elements

A big thank you to Michael Voorhaen, one of the original authors of these slides.