Introduction to Click and Project I

ECE544 Communication Networks II

Sumit Maheshwari

02/01/2019

Includes teaching material from Bart Braem, Michael Voorhaen and previous TA Francesco Bronzino
What is a ‘Software Router’?

• Router in a PC
• Program that receives, processes and forwards packets to the next node
Click Modular Router

- Extensible toolkit for writing packet processors
- PhD thesis Dr. Eddie Kohler (MIT) Architecture centered on elements:
  - Small building blocks
  - Perform simple operations e.g. decrease TTL
  - Add/remove components e.g. connect queues
- The whole Click router itself is a **Linux kernel module**!
  - Directed graphs of elements
  - Elements are written in C++
Why Click?

• Modular framework
  • Packets are intercepted with Click Toolkit
  • e.g. FromDevice(eth0) -> Queue() -> ToDevice(eth1);

• Building blocks:
  • Provides a number of prewritten routing elements like Queues, Shapers, packet senders etc.
  • You can write your own elements!

• High level programming interface
  • Hides Kernel intricacies

• Easy to install, modify and use
Click Routers: Main Concepts

• Elements
• Ports
• Packets
• Configuration
• More...
Click Routers: Main Concepts

- Router: **Elements** connected by **edges**
- Output ports to input ports
- Describes possible **packet flows** through directed graphs

![Diagram](image-url)
Elements

• Most important user-visible abstraction in Click
• Elements (they are C++ classes)
• Element instances: C++ objects
Elements

- Input port(s): Interface where packets arrive, triangles
- Output port(s): Interface where packets leave
- Inside: packet processing!

![Diagram](attachment:diagram.png)
Elements

• Lots of available elements to start with

• Write your own elements with the provided interfaces:
  • Receive configuration values
  • Receive packets
  • More later..

• List of Elements:
Ports

• Push port:
  • Filled square or triangle
  • Source initiates packet transfer: event based packet flow

• Pull port:
  • Empty square or triangle
  • Destination initiates packet transfer: Used with polling, scheduling etc.

• Agnostic port:
  • Square-in-square or triangle-in-triangle
  • Becomes push or pull (inner square/triangle filled or empty)
Ports

- FromDevice: receive packet p
- Null: push(p) => return
- Null: enqueue p
- ToDevice: ready to transmit

- pull(): pull(p) => return p
- send p
Push-Pull Violations

• Push port
  • has to be connected to push or agnostic port
  • Conversion from push to pull with push-to-pull element
  • E.g. queue

• Pull port
  • Has to be connected to pull or agnostic port
  • Conversion from pull to push with pull-to-push element
  • E.g. unqueue
Project 1 Goals

• Get familiar with Click’s environment
• Get familiar with our virtualized environment
• Practice with existing elements and create your first running configuration
• Use existing interfaces to communicate between multiple click instances
Playground 1(A): Payload to the Router

• Create a new packet with payload “hello”
• Print its content to terminal
• Drop the packet
• Hints:
  • Only use of “existing” elements
  • Minimum elements in the configuration are “three”
  • Click provides a collection of “source” elements
Playground 1(B): Print Me

• Run test2.click configuration (inside click folder) using:
  $userlevel/click conf/test2.click
• You will observe that it does not print anything!
• Modify test2.click to achieve following:
  • Change the data field to following:
    This name is: Your Name
  • Limit total packets in the router to 5.
  • Print output after each element.
  • Print your each output in a readable format!

• Hints:
  • Only use of “existing” elements
General Suggestions

• Get comfortable with the VM as soon as possible
  • You do not want to get stuck at the last minute without even having the chance to do the exercises

• These exercises are just a Click warm up
  • There is a lot more to learn
  • Some will be covered in the next project
  • Read more about click:
    • https://dspace.mit.edu/bitstream/handle/1721.1/86585/48118278-MIT.pdf
  • Use elements from:
Submission Info

• Due:
  • Playground1: Feb 8th *(before class)*

• Technical questions: use the mailing list. It is better for all of us.
  • comnet2@winlab.rutgers.edu
  • Apply at: https://groups.google.com/a/winlab.rutgers.edu/forum/#!forum/comnet2

• Submission instructions:
  • Submit a single archive (zip or tar.gz) to sumitm@winlab.rutgers.edu with subject “ECE544 P1PG1”
  • Include in the archive 1 folder named “PG1”. It should contain only the click configuration file(s).
  • If you want to include additional information, write a README file.
  • *Do not include the whole click resources!*