# The Click Modular Router

Ashutosh Mittal (E-mail: amittal@kth.se)

## I. Paper summary

THIS paper introduces *Click*, a flexible, modular software architecture for creating routers. The code for the software has been written in C++. Click modules are called elements, each of which are basically packet processing units. Elements have Input and output ports which can be either push, pull or both. Element behavior is determined by the elment class. Click includes a simple language for configuring a topology and has a preprocessor that catches intuitively incorrect configurations. To build a router configuration, the user chooses a collection of elements and connects them into a directed graph. The graphs edges called connections, represent possible paths for packet handoff. To extend a configuration, the user can write new elements or compose existing elements in new ways, much as UNIX allows one to build complex applications directly or by composing simpler ones using pipes. Click packets are moved from element to element via virtual function call and are not copied. Click works on copy-on-write principle.

It can forward packets 10% slower than Linux kernel alone. Slowdown is due to packet handoff to each element, ie. more the elements, slower it gets. It has a 73,000 (64-byte) packets IP forwarding rate which is faster than some cheap commercial routers.

## II. Significant Contributions

- Routers are now expected to do much more than just route packets, like firewall, prioritize, classify for differentiated services arch, tunnel, filter, NAT etc. Thus, the need for this flexibility is quite vital for the Internet where the conventional routers are not flexible enough. Click makes it very easy for the users to implement an element and write his own architecture for the router. Also, its very easy to program the modules for click by writing a C++ object. This ease gives the user flexibility to configure routers as per the need.
- Unlike many other software reviewed in this course which have been built on Python, Click is built on C++. This enhances the performance and optimization of the application.
- Generally in other software, packets might be pushed to interfaces busy sending and the router may loses the ability to affect that packet later. However, Click has a generic mechanism called packet upstream notification that enables upstream pull elements to be notified when a packet is ready. This is enabled by click's support for flow based router context.

## III. Unresolved Issues

- It is difficult to share things not oriented to packet flow such as router tables. Clicks reliance on packet flow as an organizational principle means that small elements are not appropriate for all problems. Particularly, large elements are required when control or data flow doesnt match the flow of packets: the control flow required to process a protocol like 802.1d is too complex to split into elements.
- Queues in Click are explicit rather than implicit within each element. This allows configurations like single queue feeding multiple interfaces. However, it doesn't say about how to schedule CPU time among competing push and pull paths. It uses Linux default scheduling and a work list, a list to which Click can defer processing that gets processed once every 8 packets, to schedule processing.
- Clicks modularity sometimes impose performance costs in two ways: the overhead of passing packets between elements, and the overhead of unnecessarily general element code.