LINC - OpenFlow Switch

Introduction

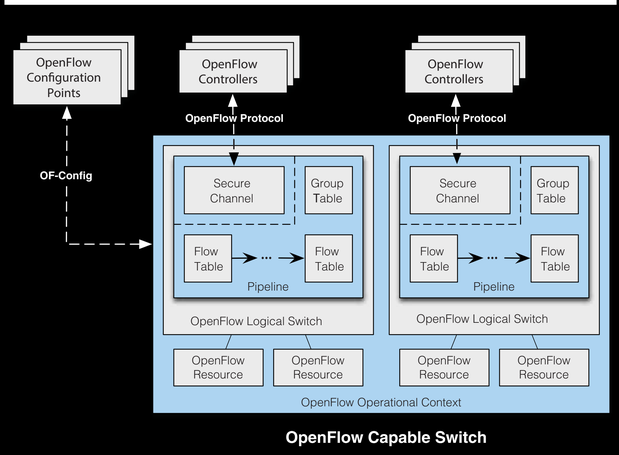
LINC (Link Is Not Closed) is a completely new open-source switching platform available through flowforwarding.org, a community promoting free open source Apache 2 license implementations based on OpenFlow specifications. LINC has been developed to serve as a reference implementation that includes the full feature set of OpenFlow specifications. The primary goal with developing LINC was to quickly and cost effectively develop and evaluate the OpenFlow 1.2 and 1.3.1 and OF-Config 1.1 versions of the specifications to run on COTS platform, and do it well ahead of similar projects that use conventional tools. For this reason we picked Erlang, a functional language with its roots in telecom.

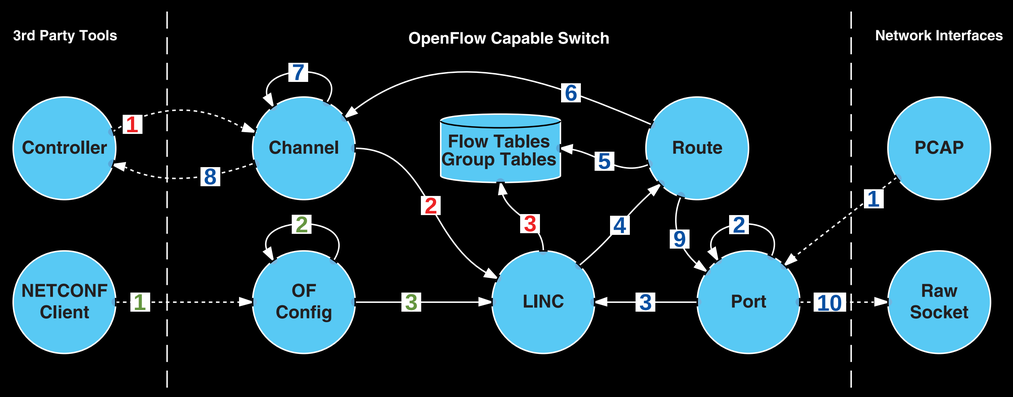
Architecture

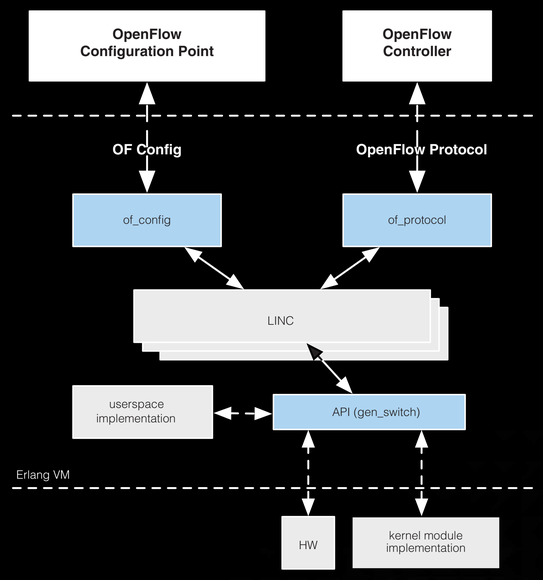
LINC is a software switch that fully implements OpenFlow 1.2, OpenFlow 1.3.1 and OF-Config 1.1 versions of the OpenFlow specifications. Its main software blocks: OpenFlow Capable Switch, OpenFlow protocol module and OF-Config module. These are designed to follow the Erlang OTP principles and are developed as separate applications where each application implements a certain well defined functionality, has clearly defined interface, dependencies and supervision tree. The OpenFlow protocol is implemented in the of_protocol library application. The functionality of the OpenFlow Capable Switch is implemented in the linc library. It receives OF-Config commands and executes them in the OpenFlow Operational Context. It handles one or more OpenFlow Logical Switches that consist of the channel component, replaceable back-ends and common logical switch logic. The channel component is a communication layer between the OpenFlow Logical Switch and the OpenFlow Controllers.

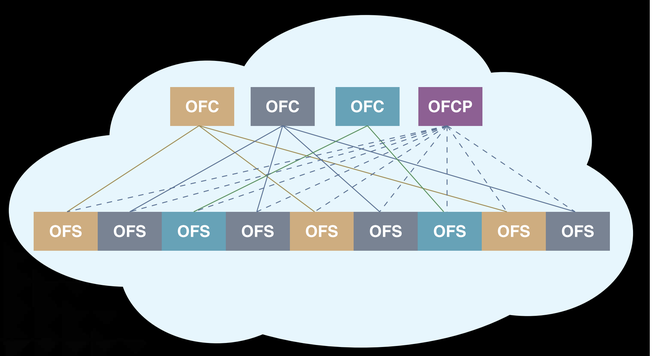
Conclusion

The development of LINC switch began in February of 2012 and thanks to the efficiency of development in Erlang, it was alpha-released in June and was one of the first OpenFlow 1.2 switches to be demonstrated at the InterOp, Tokyo. So far, the effort required to design, implement and support the LINC project has been less than 4000 man-hours. Now, that we have a full featured implementation as the foundation, we plan to work on its performance and uses cases. Since LINC backend has been currently implemented in user space, future plans include kernel space, multi-core and hardware implementations.









OFS - OpenFlow Switch
OFC - OpenFlow Controller
OFCP - OpenFlow Configuration Point

OF Config 1.1 interface to OpenFlow Configuration Points
1. NETCONF tool (e.g. Ryu) sends OF CONFIG message to set controller
2. of_config application parses XML message and transforms to Erlang term
3. update LINC state and initiate channel towards new controller

OpenFlow 1.2 and 1.3.1 interface to OpenFlow Controllers
1. controller sends packet_out message over OpenFlow channel
2. channel process transforms OF message to Erlang term and passes to LINC backend
3. LINC process updates ETS tables with flows

Erlang based implementation of LINC backend
1. packet from network interface received by Erlang process managing OF port
2. packet is decoded and transformed to Erlang term
3. Erlang term is forwarded to LINC backend
4. LINC Erlang backend spawns route process to process the packet
5. route process fetches flows from Flow/Group tables and applies them
6. route process sends packet_in message to channel process
7. packet_in message is encoded as OF message
8. OF packet_in message is sent to controller
9. OF packet_out is routed through flow tables
10. As a result of output action packet is sent to raw socket

Replaceable back-ends implement the actual logic for switching the packets. They manage flow tables, group table, ports etc. and reply to OpenFlow Protocol messages received from the Controller. Due to the use of a common API (gen_switch), LINC's Logical Switch can use any of the available backends. Common switch logic does not depend on the back-end. It handles switch configuration, manages the channel component and OpenFlow resources. OF-Config protocol handling is implemented by the of_config library application that handles parsing, validation and generates an interpretation of the OF-Config messages received from an OpenFlow Configuration Point.

References:

1. http://www.flowforwarding.org
2. http://www.erlang.org

FLOWFORWARDING.ORG

Erlang SOLUTIONS

Infoblox Inc., Santa Clara, CA.
Stuart Bailey (sbailey@infoblox.com)
Sandhya Narayan (snarayan@infoblox.com)

Erlang Solutions Ltd.
Krzysztof Rutka (krzysztof.rutka@erlang-solutions.com)
Konrad Kaplita (konrad.kaplita@erlang-solutions.com)