SDN: Experimenting with the control to forwarding plane interface

EWSDN’12 – Darmstadt

Evangelos Haleplidis (ehalep@ece.upatras.gr)
Jamal Hadi Salim (hadi@mojatatu.com)
Joel Halpern (joel.halpern@ericsson.com)
Spyros Denazis (sdena@upatras.gr)
Odysseas Koufopavlou (odysseas@ece.upatras.gr)
Intro

- **SDN**
  - A culmination of effort from the era of A&P Networks.
  - Requirement: Separate Forwarding & Control Plane.
    - Abstractions
    - API

- **Two protocols:**
  - ForCES
  - OpenFlow
Motivation

- Coexistence, Convergence, Assimilation?
  - Need for two?
- Unleash programmability for network service architectures.
  - How does OpenFlow & ForCES cope with new requirements?
ForCES

- FORwarding & Control Element Separation.
  - IETF
  - Suit of protocols
    - Protocol (RFC 5810)
    - SCTP-TML (RFC 5811)
    - Model (RFC 5812)
  - Abstraction of the Forwarding plane.
  - API to control and manage modeled devices.
**ForCES Model**

- Models FEs using LFBs.
- **LFB**
  - Model building block
  - Fine-grained* operations of the Forwarding Plane
  - LFB Topology creates FE.

![Diagram of LFBs and FEs](image-url)
ForCES Protocol

- Model agnostic.
- A CE can:
  - Associate with an FE
  - Configure one or more LFBs of an FE
  - Query one or more LFBs of an FE
  - Redirect in/out.
  - Subscribe to events to any LFB.
ForCES Protocol (2)

- **Mechanisms:**
  - Capability discovery
  - Transactions
  - Two phase commits
  - Batching/parallelization
  - High Availability and Failover
  - Command Pipelines
  - Heartbeat mechanism
OpenFlow

- Models in detail an OpenFlow switch.
- Protocol is tightly coupled with model
  - Model change => Protocol change
- Consists:
  - One or more Flow Tables.
  - Group Table
  - Action Set
  - Ports
<table>
<thead>
<tr>
<th></th>
<th>OpenFlow</th>
<th>ForCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match Fields / Counters.</td>
<td>LFB Components.</td>
<td>Special Values of Components / Implementation Specific.</td>
</tr>
<tr>
<td>Instructions / Action Set / Action List</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actions</td>
<td>LFBs</td>
<td></td>
</tr>
<tr>
<td>Pipeline.</td>
<td>LFB connectivity.</td>
<td></td>
</tr>
<tr>
<td>Static Model of switch.</td>
<td>No limitation.</td>
<td></td>
</tr>
<tr>
<td>Static Capability list.</td>
<td>No limitation.</td>
<td></td>
</tr>
</tbody>
</table>
## Protocol Similarities

<table>
<thead>
<tr>
<th>OpenFlow protocol</th>
<th>ForCES protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure channel – TLS</td>
<td>ForCES runs over IPSec</td>
</tr>
<tr>
<td>Feature discovery</td>
<td>LFB Capability discovery</td>
</tr>
<tr>
<td>Configuration / Modify-State / Read-</td>
<td>Configuration / Query</td>
</tr>
<tr>
<td>State</td>
<td></td>
</tr>
<tr>
<td>Packet Out / Packet In.</td>
<td>Packet Redirect</td>
</tr>
<tr>
<td>Barrier.</td>
<td>Transaction.</td>
</tr>
<tr>
<td>Flow-Removed / Port-Status / Error</td>
<td>Event Notification messages</td>
</tr>
<tr>
<td>Hello</td>
<td>Association messages</td>
</tr>
<tr>
<td>Echo</td>
<td>Heartbeat messages</td>
</tr>
<tr>
<td>Xid (Transaction id)</td>
<td>Correlator</td>
</tr>
</tbody>
</table>
## Protocol Differences

<table>
<thead>
<tr>
<th>OpenFlow protocol</th>
<th>ForCES protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experimenter</td>
<td>Not required.</td>
</tr>
<tr>
<td>Only error reporting</td>
<td>Acknowledge request mechanism</td>
</tr>
<tr>
<td>Not available</td>
<td>Message Batching</td>
</tr>
<tr>
<td>Static Events provided by protocol</td>
<td>Dynamic Event Subscription</td>
</tr>
<tr>
<td>Echo messages</td>
<td>Controlled heartbeat mechanism</td>
</tr>
<tr>
<td>Not available</td>
<td>Execution Mode Selection</td>
</tr>
<tr>
<td>Not available</td>
<td>Command pipelining</td>
</tr>
</tbody>
</table>
All well and good… but!

- Whither ForCES?
  - Industry (little adoption)
    - Disruptive business model.
    - Current known:
      - NTT Japan Implementation
      - Mojatatu’s Network OS
      - Verizon (recently published two drafts).
      - Ericsson
      - Huawei
  - Academia (few implementations)
    - No open source availability (YET!)
      - Zhejiang Gongshang University
      - University of Quebec (Montreal)
      - University of Patras
Good news everyone!

- ForCES-based Network Operating System SDK availability
  - Implemented by Mojatatu Networks (Jamal Hadi Salim - ForCES wg chair).
  - Mojatatu NetOS© SDK –write control applications and incorporate FE hardware.
  - Will be available this year.
  - Inquiries to sdk@mojatatu.com

- ForCES open source code availability. (internally announced – no link/date yet!)
  - Zhejiang Gongshang University
More good news - Tools!
Ah, yes the motivations!

ForCES Controller (CE)

ForCES Protocol

ForCES Wrapper

LFB description

Open vSwitch 1.4 + Extension
(OpenFlow v1.0)
Motivations – Current work
Use Case

- New services for Home Gateways.
- Split the Home Gateway
  - Forwarding in customer
  - Services in provider.
- Current issue: NAT
  - OpenFlow currently lacks TCP-flags matches.
Conclusions

- ForCES vs OpenFlow
- Defined a middleware for proof-of-concept of convergence.
- ForCES has solved issues that OpenFlow now tries to tackle.
- OpenFlow can take into advantage ForCES expertise and experience.
Backup Slide #1

OpenFlow Controller

ForCES Wrapper

LFB description

OpenFlow protocol

LFB