Enhancing the SDN Ecosystem – The Evolution of OpenFlow

Ben Mack-Crane / March 16, 2016
ONF Specification Area Director
You know you need to do something...

when your shiny new car begins to strain under the load.
What is OpenFlow’s Role in SDN?

Original OpenFlow:
• Separation of Control & Datapath
• Run-time control protocol
• Datapath model that provides flexibility to Controller
• Support some common packet headers (e.g., ETH, MPLS, IP)
What is OpenFlow’s Role in SDN?

Original OpenFlow:
• Separation of Control & Datapath
• Run-time control protocol
• Datapath model that provides flexibility to Controller
• Support some common packet headers (e.g., ETH, MPLS, IP)

Evolving OpenFlow:
• Separation of Control & Datapath
• Run-time control protocol
• Datapath models that provide flexibility to Controller
• Support any datapath headers, including new technologies
• Support programmable datapaths, e.g. P4 programed
• API and open source to speed programming
What concerns are we addressing?

What we cannot do…
(but would like to)

- Additional Events
- Encap/Decap
- Stateful Flows
- Delegate Control

What we have to do…
(but would prefer not to)

- Support ‘required’ technologies
- Continually add new technologies to OF Spec

How can we simplify or make OF easier?

- Eliminate Special Cases
- Application TTPs
- OpenFlow API
- Open Source

Additional Events
Encap/Decap
Stateful Flows
Delegate Control
Expanding and generalizing

<table>
<thead>
<tr>
<th>Add new capabilities</th>
<th>Explicitly define “built-in” behaviors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Events (timers, thresholds)</td>
<td>Header formats</td>
</tr>
<tr>
<td>Generic Encap/Decap</td>
<td>Parsing rules (Match prerequisites)</td>
</tr>
<tr>
<td>Flow State Tables</td>
<td>Flexible Action Set</td>
</tr>
<tr>
<td>Packet Generation</td>
<td>IP Frag/Reasm flag</td>
</tr>
<tr>
<td>Local Control Model</td>
<td></td>
</tr>
</tbody>
</table>
Restructuring the specification

Base Protocol Specification
(define datapath control protocol)
Match-Action (MAT) model
Control channel
Fault recovery

OF Spec

Table Type Patterns
(define datapath protocol details)
Match fields
Actions
Table entry types
Tables
Pipeline

Autonomous Functions
(define behaviors delegated to switch)
Type and parameters
Data model for internal state
Simplifying (tidying up)

**Action Set**
- Fixed actions
- Fixed order (in specification)

**Synchronized Tables**
- Built-in behavior
- “Magic” effects (unspecified)

**Required Switch Behavior**
- Required Match Fields/Actions
- Match field prerequisites, etc.

**Action Set Group**
- Any actions
- Programmable order

**Bundles and Macros**
- Explicit operations
- Programmable

**Application TTPs**
- App Match Fields/Actions
- App flow entry types/pipeline

Keep the datapath control model simple and programmable.
Opening the door to innovation

OpenFlow protocol provides:
• Basic, flexible datapath model
  – Mappable to implementations
• Open control model
• Robust control connectivity
• API and open source driver libs?

OpenFlow leaves to programmers:
• Header formats
• Protocol-specific actions
• Meter types
• Queue & scheduler types
• Composite behaviors
Thanks! Comments? Questions?

Ben Mack-Crane / ben-at-cor-sa-dot-com
ONF Specification Area Director