tinyNBI: Distilling an API from Essential OpenFlow Abstractions

Jasson Casey, Andrew Sutton, Alex Sprintson

Texas A&M University
University of Akron
Flowgrammable.org
Where are we?

HTTP/S REST API
Server infrastructure, SDN libraries, data persistence
OpenFlow, NetConf, OVSDB
NetFlow, IPFIX, jFlow, sFlow
OpFlex
SNMP

DHCP, Authentication

bridging, multipath routing

North Bound Interface

South Bound Interface

Switches

Flowgrammable
Driving the Next SDN Generation
Where are we?

- DHCP
- Authentication
- HTTP/S REST API
  - bridging
  - multipath routing
- Plugin API
- Server infrastructure, SDN libraries, data persistence
- OpenFlow
- NetConf
- OVSDB
- NetFlow, IPFIX, jFlow, sFlow
- OpFlex
- SNMP
- North Bound Interface
- South Bound Interface
- Switches
North Bound Interface (NBI)

- Glue between controllers and applications
- API for writing OpenFlow applications

Diagram showing the interaction between switches, controllers, and applications with events such as FlowMod, PacketIn, FlowRemoved, PacketOut, and StatsReq.
Most production networks …

• are heterogeneous

• contain multiple vendors

• contain multiple device types

• operate varying versions of software
OpenFlow …

• has five versions in production

• has a new versions coming

• is not additive

• has a high degree of optionality
Most Features are Optional

<table>
<thead>
<tr>
<th>Feature</th>
<th>Match 1.0</th>
<th>Match 1.1</th>
<th>Match 1.2</th>
<th>Match 1.3</th>
<th>Match 1.4</th>
<th>Instruction 1.0</th>
<th>Instruction 1.1</th>
<th>Instruction 1.2</th>
<th>Instruction 1.3</th>
<th>Instruction 1.4</th>
<th>Action 1.0</th>
<th>Action 1.1</th>
<th>Action 1.2</th>
<th>Action 1.3</th>
<th>Action 1.4</th>
<th>Port 1.0</th>
<th>Port 1.1</th>
<th>Port 1.2</th>
<th>Port 1.3</th>
<th>Port 1.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
<td>70</td>
<td>80</td>
<td>90</td>
<td>100</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
<td>150</td>
<td>160</td>
<td>170</td>
<td>180</td>
<td>190</td>
<td>200</td>
</tr>
<tr>
<td>Required</td>
<td>0</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
<td>70</td>
<td>80</td>
<td>90</td>
<td>100</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
<td>150</td>
<td>160</td>
<td>170</td>
<td>180</td>
<td>190</td>
</tr>
</tbody>
</table>

Flowgrammable
Driving the Next SDN Generation
Writing OpenFlow applications …

• requires extensive capability detection

• requires extensive error handling

• is not possible without apriori knowledge

• is not for the average programmer
Introduce a tiny NBI

* Server distribution
* Application synchronization
* Topology discovery (LLDP)
* Network dependency management

* Version negotiation
* Echo state
* Barrier state
* Uniform datamodel
* Command translation

* Targets single switch
* Hybrid network behavior
* Maintenance behavior

---

Flowgrammable
Driving the Next SDN Generation
Introduce a tinyNBI that …

• has a simple sockets “like” interface

• abstracts away OpenFlow version details

• simplifies the capability detection

• supports cross language bindings
Read/Write from/to the Data Model

Minimal Control Plane

Data Plane

Switch

Datapath

Flow Table

Buffer

Flow

Match

Instruction

Port

Group

Action

Queue

Connection

0..1

1..*

0..1

0..1

0..*

0..*

0..*

0..*

1..*

1..*

1..*

1..*
Abstractions have …

• capabilities that are read only

• configurations that can be read or written

• statistics that are read only

• event generation: packet, port, flow
tinyNBI also introduces…

• an application lifecycle

• an allocation model for finite resources

• capability requirements statement

• non-native feature offload
Questions?

Minimal Control Plane

Switch

Data Plane

Datapath

Flow Table

Buffer

Flow

Match

Instruction

Port

Group

Action

Queue

Connection

0..1

1..*

0..1

0..*

1..*

0..1

1..*

0..*

1..*

0..1

1..*

0..1

1..*

0..1