BGP: Hydrogen: Release Plan

Contents

Introduction

BGP/PCEP protocol library is a project aiming to provide Java-based implementation of Border Gateway Protocol and Path Computation Element Protocol.

By enabling the Controller to utilize more standardized ways of talking to the underlying network, it can be deployed in a wider variety of scenarios.

BGP is the core protocol holding together the Internet in its current shape and form, which is extensible enough to provide functions well outside of its original purpose. With the advent of Link State and TE Information draft, BGP can be used to safely distribute IGP information to entities outside of usual IGP peering. The Controller is a prime example of a network entity which can benefit from this information, as it can directly use it as a source of topology information.

PCEP is a protocol originally designed for offloading optimal path computation in MPLS-TE networks from the head-end router to a dedicated accelerator (called Path Computation Element, or PCE) residing on the control plane. While the original specification called for a router-driven operation, the recently adopted Stateful PCE extension aligns the protocol for use by the Controller.

Release Deliverables

Name	Description
BGP-LS	• Definition of a topology model exposed by BGP/LS • Implementation of a BGP protocol library, supporting BGP-4, Graceful Restart, Multiprotocol extensions, BGP/LS, Communities, Extended Communities and 4-Byte AS numbers (NEW speaker only) • Implementation of a BGP RIB, allowing for redundant data sources, but limited to listener-only functionality
PCEP	• Definition of a network-wide path programming model exposed by PCEP • Implementation of a PCEP protocol library, supporting PCEP, Objective functions, Diffserv, Stateful extensions, GCO and PCE-initiated LSPs
Models	• Model of a PCEP message in YANG. • Model of a PCEP tunnel in YANG. • Model of a BGP-LS NLRI in YANG.

Release Milestones

Milestone	Offset 0 Date	Deliverables
M1	7/22/2013	Name: Release Plan; Status: Done; Description: Candidate Release Plan

<table>
<tr><td>M2</td><td>8/21/2013</td><td>
<table>
<tr><th>Name</th><th>Status</th><th>Description</th></tr>
<tr><td>Release Plan</td><td>Done</td><td>Final Release Plan</td></tr>
<tr><td>Migration of Framework to NETTY</td><td>Done</td><td rowspan="3">NETTY framework is an asynchronous event-driven network application framework
for rapid development of maintainable high performance protocol servers & clients. Our
client-server communication is mostly hand-written using java NIO directly. The code is
robust and not bug free (there is known issue when BGP client constantly receives messages,
the socket gets congested and does not send KeepAlive messages, therefore the connection
drops unexpectedly). Migration to NETTY was suggested to avoid such issues and maintain
less code on our side. First part of the migration is to switch the core of the client-server
communication to NETTY. This should not affect the protocols itself. After the core
communication is switched to NETTY, the protocols can be switched too.</td></tr>
<tr><td>Migration of BGP to NETTY</td><td>Bug44
Done</td></tr>
<tr><td>Migration of PCEP to NETTY</td><td>Bug43
Done</td></tr>
<tr><td>Write pcep-message.yang</td><td>Done</td><td rowspan="3">To integrate with MD-SAL, we need to write YANG models representing BGP-LS/PCEP
concepts. BGP: we need YANG model for BGP-LS NLRI. This model was already started and needs
to be finished. The last part for BGP is to integrate with YANG models provided by IETF
draft-clemm-yang-network-topo-00. PCEP: we need a model representing PCEP messages and PCEP
tunnels.</td></tr>
<tr><td>Finalize bgp-ls-nlri.yang</td><td>Done</td></tr>
<tr><td>Write pcep-tunnel.yang</td><td>Done</td></tr>
<tr><td>DTO Generation</td><td>Done</td><td>Wiring yangtools DTO generation into build system.</td></tr>
</table>
</td></tr>
<tr><td>M3</td><td>9/18/2013</td><td>
<table>
<tr><th>Name</th><th>Status</th><th>Description</th></tr>
<tr><td>Switch BGP to generated DTOs</td><td>Bug45
Done</td><td rowspan="2">In this stage of the project, the YANG models are finalized and the generation of the DTOs should be provided by Yang tools. Therefore it is the time and place to integrate
the generated DTOs with the code.</td></tr>
<tr><td>Integrate BGP RIB with MD-SAL</td><td>Bug46
Done</td></tr>
</table>
</td></tr>
<tr><td>M4</td><td>10/16/2013</td><td>
<table>
<tr><th>Name</th><th>Status</th><th>Description</th></tr>
<tr><td>API Freeze</td><td></td><td rowspan="2">Integrate PCEP with more general overlay model.</td></tr>
<tr><td>Switch PCEP to generated DTOs</td><td>Bug47
Done</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>M5</td><td>11/13/2013</td><td>
<table>
<tr><th>Name</th><th>Status</th><th>Description</th></tr>
<tr><td>Code Freeze</td><td></td><td rowspan="4">Writing end-to-end tests and user-facing documentation.</td></tr>
<tr><td>BGP tests</td><td>Bug49
Done</td></tr>
<tr><td>PCEP tests</td><td>Bug50</td></tr>
<tr><td>Documentation</td><td>Bug51</td></tr>
<tr><td>Implement BGP topology provider</td><td>Bug108
Done</td><td>After sourcing topology data from BGP/LS and discriminating it in bgp-rib-impl

component, we need to transform this data into a topology model.

The base topology models are defined in http://tools.ietf.org/html/draft-clemm-netmod-yang-network-topo-00 and already
imported into topology-api artifact. Evolve those models such that they are usable with MD-SAL and create a model-to-model transformation artifact, topology-provider-bgp, which will consume the Local RIB produced by bgp-rib-impl and will provide an L3 IGP topology view of that data.</td></tr>
<tr><td>End-to-end integration tests</td><td>Bug110</td><td>Design and implement test cases for the entire protocol stack. This involves a

mock PCEP and BGP speakers, who feed some information into a completely-assembled
MD-SAL container and then inquiring the data service to see if correctly-modeled
information is there and no anomalies are detected in the system.</td></tr>
<tr><td>Integrate PCEP with Tunnels</td><td>Bug48
Done</td><td>Integrate PCEP with more general overlay model.</td></tr>
</table>
</td></tr>
<tr><td>RC0</td><td>11/20/2013</td><td>
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>RC0</td><td>bugfixing</td></tr>
</table>
</td></tr>
<tr><td>RC1</td><td>11/27/2013</td><td>
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>RC1</td><td>bugfixing</td></tr>
</table>
</td></tr>
<tr><td>RC2</td><td>12/4/2013</td><td>
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Release Review</td><td>Release Review Description</td></tr>
</table>
</td></tr>
<tr><td>Formal Release</td><td>12/9/2013</td><td>
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>Deliverable Name</td><td>Deliverable Description</td></tr>
</table>
</td></tr>
</table>

Expected Dependencies on Other Projects

Depends On	Dependency Description	Needed By	Is in Other Project Release Plan
OpenDaylight Controller	Generate DTOs from provided YANG models.	M3	wiki.opendaylight.org/view/OpenDaylight_Controller:Release_Plan_2013

Compatibility with Previous Releases

Themes and Priorities

- Definition of a topology model exposed by BGP/LS
- Definition of a network-wide path programming model exposed by PCEP
- Implementation of a BGP protocol library, supporting BGP-4, Graceful Restart, Multiprotocol extensions, BGP/LS, Communities, Extended Communities and 4-Byte AS numbers (NEW speaker only)
- Implementation of a BGP RIB, allowing for redundant data sources, but limited to listener-only functionality
- Implementation of a PCEP protocol library, supporting PCEP, Objective functions, Diffserv, Stateful extensions, GCO and PCE-initiated LSPs

Versioning

0.1.0 - initial code drop
0.2.0 - project switched to NETTY
0.3.0 - integrated with generated DTOs

Other

Primary Setup Contact : Dana Kutenicsova (dkutenic@cisco.com)

CI Resource : Dana Kutenicsova (dkutenic@cisco.com)