Network-wide Protocol Monitoring (NPM): Use Cases

draft-chen-npm-use-cases-00

Huainan Chen (China Telecom)
Zhenqiang Li (China Mobile)
Feng Xu (Tencent)
Yunan Gu, Zhenbin Li (Huawei)

Mar. 24, 2019
**Control Plane Telemetry**

- **Management/control/data plane telemetry**
  - **Management plane telemetry**: network operational state retrieval and configuration management
  - **Control plane telemetry**: routing protocol monitoring and routing related data retrieval, e.g., topology, route policy, RIB...
  - **Data plane telemetry**: traffic performance measurement and traffic related data retrieval

- **Role of control plane telemetry**:
  - **Network troubleshooting**
    - 48% of the problems are based on protocol errors or misconfiguration impact both tracking of operational and provisioning
  - **Network planning**
    - No effective route policy/configuration validation approach, and lacks route-traffic correlation insight
    - Real time applications of 5G require real-time TE optimization, and accurate what-if simulation for network planning

---

**Hawei Internal Statistics: control protocol failures take up about 48% of all network issues.**

![Diagram showing network failure causes]

- 2016 Internal statistics: Network Failure Cause
  - Protocol neighbor set up failure: 20%
  - Protocol neighbor damping: 18%
  - Protocol table issue: 14%
  - Network table issue: 12%
  - Configuration issue: 10%
  - System/remote issue: 6%
  - Protocol error: 4%
  - Other: 7%

**Traditional troubleshooting methods**
- Time-consuming (hours)
- Labor-consuming
- Data acquisition difficult
- Require OAM experience for operators

**Issues, like route flapping, hard to localize**
Network-wide Protocol Monitoring (NPM) Framework

Data Source:
Topology, protocol PDU, RIB, route policy, statistics...

NPM problem space:
sufficient data type coverage, sufficient device coverage

Data Generation:
data encapsulation, data serialization, data subscription

NPM problem space:
data model definition, data process efficiency

Data Transportation:
BMP, gRPC, Netconf, new protocol?

NPM problem space:
Transportation protocol selection, exportation efficiency

Data Analysis:
Protocol troubleshooting, Policy validation, Traffic optimization, What-if simulation

NPM problem space:
data synchronization, data parse efficiency
Use case 1: ISIS Route Flapping

• Typical cause 1:
  • System ID conflict

• Typical cause 2:
  • IS-IS neighborship flapping: caused by interface flapping, BFD flapping, CPU high...

• Typical Case 3:
  • Route policy misconfiguration (e.g., multi-protocol import)

• Typical Case 4:
  • Abnormal LSP purges

<table>
<thead>
<tr>
<th>Causes</th>
<th>Conventional troubleshooting</th>
<th>Improvements with NPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>System ID conflict</td>
<td>• Manual check one by one</td>
<td>• Takes seconds</td>
</tr>
</tbody>
</table>
Use case 2: LSDB Synchronization Failure

• **Cause 1:** LSP not correctly advertised
  - It can be due to incorrect route export policy, or too many prefixes being advertised which exceeds the LSP/MTU threshold, and so on at Router A.

• **Cause 2:** LSP transmission error
  - IS-IS adjacency failure, e.g., link down/BFD down/authentication failure.

• **Cause 3:** LSP correctly received but incorrectly processed
  - The problem that happens at Router B can be faulty route import policy, or Router B being in Overload mode, or the hardware/software bugs.
Use case 3: Route Loop

- Conventional loop detection
  - Only post-event detection: TTL anomaly report or packet loss complain
  - Requires network-wide device-by-device check
- Improved with NPM:
  - Real-time and in-advance loop detection
  - Root cause analysis: correlated route change and policy record

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Route event</th>
<th>Route policy</th>
<th>Time stamp</th>
<th>Next hop</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.17.0.0/16</td>
<td>1</td>
<td>ISIS: Route-policy r1: permit/permit : cost 100</td>
<td>xx:xx:xx</td>
<td>192.168.2.2/24</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>RM: Route-policy r2: permit/deny : next-hop</td>
<td>xx:xx:xx</td>
<td>192.168.1.1/24</td>
<td>100</td>
</tr>
</tbody>
</table>
Use case 4: Tunnel Set Up Failure

- Root causes:
  - Configuration error, path computation error, link failure

- Gaps
  - Data not carried by RSVP-TE messages (PathErr/ResvErr, etc.)
    - IP address conflict
    - LSP establishment time out at head end node
    - RSVP-TE authentication failure

- Possible improvement with NPM:
  - Collection of LSP configurations, LSP states, link states and other reasons from devices along the LSP
Use case 5: Route Policy Validation

• Existing route policy validation:
  • Lacks the vision of how policy impacts the route attributes

• Route policy pre-check simulation:
  • Simulation based on device configurations: not 100% on-going network mirroring

• Possible improvements with NPM
  • Real-time track of how policy changes route attributes
  • Control plane snapshots as the simulation input: topology, protocol neighbor state, RIB... to improve the simulation accuracy
General Requirements from above use cases

1. A "tunnel" for the control plane data export:
   • Performance guarantee for: data modeling, encapsulation, serialization, exportation, transportation performance

2. Adequate protocol data collection:
   • The data type coverage:
     • Protocol PDUs (LSP, LSA, Hello, Open, Update...)
     • Network-wide RIBs
     • Route policies
     • Correlated policy and route attributes...
   • The network coverage:
     • Refers to the devices providing such information (network-wide)