Towards ONOS-based SDN Monitoring using In-band Network Telemetry

APNOMS 2017

Nguyen Van Tu, Jonghwan Hyun, and James Won-Ki Hong
Distributed Processing & Network Management (DPNM) Lab
POSTECH, Pohang, Korea
Outline

❖ Network monitoring & related work
❖ In-band Network Telemetry (INT)
❖ IntMon - INT in ONOS
❖ Discussion
❖ Summary
Network monitoring

❖ Polling and sampling

▪ Polling interval
  • Not real-time
  • Coarse-grained
  • Require continuous polling from the monitoring probe

▪ Sampling
  • May miss important information (such as micro-burst)
Related work

❖ For traditional switches

- NetFlow: polling, aggregate flow information
- SFlow: packet sampling
Related work

❖ **For traditional switches**
  - NetFlow: polling, aggregate flow information
  - SFlow: packet sampling

❖ **For SDN - OpenFlow switches**
  - OpenNetMon: [N. L. M. van Adrichem et. al., NOMS 2014]
    - Adaptive polling
  - OpenSample: [J. Suh et.al., ICDCS 2014]
    - Sampling for detecting elephant flows
Related work

❖ For traditional switches
  ▪ NetFlow: polling, aggregate flow information
  ▪ SFlow: packet sampling

❖ For SDN - OpenFlow switches
  ▪ OpenNetMon: [N. L. M. van Adrichem et. al., NOMS 2014]
    • Adaptive polling
  ▪ OpenSample: [J. Suh et.al., ICDCS 2014]
    • Sampling for detecting elephant flows

❖ For programmable data plane switches
  ▪ In-band Network Telemetry
Definition

“A framework designed to allow the collection and reporting of network state, by the data plane, without requiring intervention or work by the control plane”

In-band Network Telemetry

Definition

“...A framework designed to allow the collection and reporting of network state, by the data plane, without requiring intervention or work by the control plane”

In-band Network Telemetry

Definition

“A framework designed to allow the collection and reporting of network state, by the data plane, without requiring intervention or work by the control plane”

In-band Network Telemetry

Definition

“A framework designed to allow the collection and reporting of network state, by the data plane, without requiring intervention or work by the control plane”

In-band Network Telemetry

Definition

“A framework designed to allow the collection and reporting of network state, by the data plane, without requiring intervention or work by the control plane”

In-band Network Telemetry

❖ Advantages

▪ Real-time, packet-level granularity, polling-free
▪ Complete view of network state in the flow’s path
In-band Network Telemetry

❖ Advantages
  ▪ Real-time, packet-level granularity, polling-free
  ▪ Complete view of network state in the flow’s path

❖ INT implementation
  ▪ Implemented in the data plane
    • NPU, FPGA
    • P4 supported hardware
IntMon - P4 and ONOS

❖ **P4 - programming protocol-independent packet processors**
  - Program **how** packets should be processed in the data path
  - **Match/action** approach
  - Allow programmable packet processing, custom packet format
IntMon - P4 and ONOS

❖ P4 - programming protocol-independent packet processors
  ▪ Program **how** packets should be processed in the data path
  ▪ **Match/action** approach
  ▪ Allow programmable packet processing, custom packet format

❖ P4 and ONOS
  ▪ P4 supported in ONOS
    • ONOS-BMv2 subsystem in ONOS 1.6
IntMon - INT packet format

❖ INT as TCP/UDP shim header
  ▪ INT Port: use a specific port for INT

```
<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source Port</td>
<td>INT Port</td>
<td>TCP/UDP</td>
<td></td>
</tr>
<tr>
<td>+++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other TCP/UDP fields</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ver Rep</td>
<td>c e</td>
<td>r r r</td>
<td></td>
</tr>
<tr>
<td>+++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Int Cnt</td>
<td>Max Hop Cnt</td>
<td>Total Hop Cnt</td>
<td></td>
</tr>
<tr>
<td>+++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Instruction Bitmap</td>
<td>Reserved</td>
<td>INT header</td>
<td></td>
</tr>
<tr>
<td>+++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INT len</td>
<td>Original Dest Port</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INT data Stack</td>
<td>INT per-switch data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++ +++++++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Original payload</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

O bit: indicates that INT pkt is sent to ONOS
INT len: length of the INT header + data
IntMon - INT data

❖ Per-switch information
  - Switch ID
  - Ingress port, egress port
  - Hop latency
  - Queue occupancy
  - Ingress timestamp
  - Queue congestion status
  - Egress port TX utilization

❖ Others are possible
IntMon - Architecture

- **IntMon switch**
  - **Parser**: parse Ethernet - IP - TCP/UDP - INT
  - **Forward**: forwarding table
  - **INT process**: add/remove INT data, send INT packets to ONOS
IntMon - Architecture

- **IntMon ONOS application**
  - **IntMon control**: which flows to monitor, which data to monitor
  - **IntMon collector, processor**: receive and process INT data
  - **Sw.json**: compiled from P4 INT code, then deployed to BMv2 switches
  - **Java Interpreter**: mapping ONOS rule - P4 rule
IntMon switch

- P4 language
- Example: if pkt is at final switch (sink), then restore original packet;
IntMon - Code

- **IntMon switch**
  - P4 language
  - Example: *if pkt is at final switch (sink), then restore original packet;*

```p4
action int_sink()
{
  remove_header(int_header);
  remove_header(int_val[0]);
  subtract_from_field(ipv4.ipv4Len, int_header.int_len);
...
}

table tb_int_sink {
  reads {
    l2e.sink: exact;
  }
  actions {
    int_sink;
  }
}

control process int_sink {
  apply (tb_int_sink);
}
```
IntMon controller

- if pkt is at final switch (sink), then restore original packet;

```java
private void installRuleIntSink(DeviceId did) {
    /* in table tb_int_sink, if i2e.sink flag value is 1, then do action int_sink*/

    ExtensionSelector extSelector = Bmv2ExtensionSelector.builder()
            .forConfiguration(INTMON_CONFIGURATION)
            .matchExact("i2e", "sink", 1)
            .build();

    ExtensionTreatment extTreatment = Bmv2ExtensionTreatment.builder()
            .forConfiguration(INTMON_CONFIGURATION)
            .setActionName("int_sink")
            .build();

    FlowRule rule = DefaultFlowRule.builder().forDevice(did).fromApp(appId)
            .withSelector(DefaultTrafficSelector.builder().extension(extSelector, did).build())
            .withTreatment(DefaultTrafficTreatment.builder().extension(extTreatment, did).build())
            .withPriority(FLOW_PRIORITY)
            .makePermanent()
            .forTable(tableMap.get("tb_int_sink"))
            .build();

    // install flow rule
    flowRuleService.applyFlowRules(rule);
}
IntMon - Interface

❖ Controlling interface
  ▪ Multiple flows with wildcard support
IntMon - Interface

- Monitoring interface

<table>
<thead>
<tr>
<th>FID</th>
<th>SRC ADDRESS</th>
<th>DST ADDRESS</th>
<th>MONITORING DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>10.0.0.1:38874</td>
<td>10.0.0.4:5001</td>
<td>SWITCH: Switch ID = 2, InPort ID = 1, Hop Latency = 20, SWITCH: Switch ID = 1, InPort ID = 1, Hop Latency = 20, SWITCH: Switch ID = 3, InPort ID = 0, Hop Latency = 37,</td>
</tr>
</tbody>
</table>
Discussion

❖ Problem: original INT sends all INT packets to the monitoring probe

- 1 packet in network ~ 1 INT packet sent to monitoring probe
  - Information duplication, high CPU cost
Discussion

❖ **Problem:** original INT sends all INT packets to the monitoring probe

- 1 packet in network ~ 1 INT packet sent to monitoring probe
  - Information duplication, high CPU cost

❖ **Solution**

- **Remove unnecessary information**
  - Only send INT packets to ONOS when the value exceeds a threshold (e.g., hop latency)

- **External Collector**
  - Multiple instances to share the load
  - Send the aggregated data to centre ONOS controller
Problem: High INT bandwidth overhead

- Every packets carry INT information through their path
Discussion

❖ **Problem: High INT bandwidth overhead**
  - Every packets carry INT information through their path

❖ **Solution**
  - Use **Sampling** for some specific purposes (e.g., elephant flow detection)
  - Need option to enable/disable sampling
Summary

❖ Network monitoring
  - Problems, related work
Summary

- **Network monitoring**
  - Problems, related work

- **In-band Network Telemetry**
  - A new method for real-time, fine-grained network monitoring
Summary

❖ **Network monitoring**
  ▪ Problems, related work

❖ **In-band Network Telemetry**
  ▪ A new method for real-time, fine-grained network monitoring

❖ **IntMon: INT monitoring in ONOS**
  ▪ IntMon packet format: INT data as TCP/UDP shim header
  ▪ IntMon P4 switch
  ▪ IntMon ONOS application
Summary

❖ **Network monitoring**
  - Problems, related work

❖ **In-band Network Telemetry**
  - A new method for real-time, fine-grained network monitoring

❖ **IntMon: INT monitoring in ONOS**
  - IntMon packet format: INT data as TCP/UDP shim header
  - IntMon P4 switch
  - IntMon ONOS application

❖ **Future work**
  - INT pre-processing in P4, external Collector
  - Sampling
Q&A