OFRewind: Enabling Record & Replay Troubleshooting for Networks

Andreas Wundsam • Dan Levin
Srini Seetharaman • Anja Feldmann

USENIX ATC 2011
classical switch
Quick OpenFlow 101

OpenFlow switch

OpenFlow Protocol

Controller

FLOW_MOD

PKT_IN

FLOW_MOD

entry

PKT_IN

entry

Packet Forwarding Table (TCAM)

Controller

CLI

Web-IF

SNMP
OpenFlow entry

(Figure from the Openflow Intro Presentation, N. McKeown)
Back to the topic of my talk: OFRewind!
Motivating use case

CPU Utilization of an OpenFlow switch

Nov-06-2009 to Nov-07-2009
No correlation!

Arrivals of PKT_IN msgs
No correlation!

Arrivals of FLOW_MOD msgs
No correlation!
Clueless...

- Switch is a **black box component**
- Can't inspect internal state, source code
- No analytical explanation for the behavior
- Message arrivals do not correlate with symptoms
- Existing interfaces (CLI, SNMP) too coarse grained
Troubleshooting networks is hard

huge, critical  black boxes  timing / races
A solution?

- Record
  - In production
- Replay
  - Reproduce at convenient location / pace
- Troubleshoot
Existing approaches

**Endhost Replay Debugging**

- Fully deterministic replay, via binary instrumentation / virtualization
- ✗ no black boxes
- ✗ scalability?

**TCPDump / TCPReplay et. al.**

- Capture/Replay events
- ✗ Single vantage point, no network wide view
- ✗ Scalability due to dataplane datarates
Existing approaches

Endhost Replay

- Fully deterministic replay, via binary instrumentation / virtualization
- ✗ no black boxes
- ✗ scalability?

TCPDump / TCPReplay

- Capture/Replay events
- ✗ Single vantage point, no network wide view
- ✗ Scalability due to dataplane datarates

Full recording of all events feasible?
However...

- Not all traffic is equal
  \((ctrl \text{ plane: } 1\% \text{ traffic, } 95-99\% \text{ bugs!})\)*

- Behavior of many network devices:

  Largely Deterministic w.r.t.

  Control Plane Network Events

* Altekar / Stoica, 2010
**Record**

- events + traffic
- **selective**: record important traffic (control)
- skip/aggregate less important traffic (data plane)

**Replay**

- reinject events + traffic
- "best effort replay"
- replay partial recordings
- reproduce problem at a chosen time / location

---

Go Network* Wide / Always On!

* controller domain
Replay Tweaking

Localize problems through:

- **Device mapping**
  - different devices / versions
  - investigate regressions / vendor implementation issues

- **Time dilation**
  - Scale time
  - investigate timing issues

- **Trace bisection**
  - iteratively replay subselected traffic
  - localize events that trigger failure
Goals

✓ **Record** a controller domain

✓ Scalable, selective, consistent

✓ Even with black boxes

✓ coordinated **Replay**

✓ Replay tweaking

✓ Localize problems
Non-Goals

- ✗ Root cause analysis
- ✗ Automatic configuration of what to record
- ✗ Fully deterministic replay
Introducing the tool
System design

2 components of 2 modules each:
OFRecord
OFReplay

OpenFlow controller

OFReplay
Typical Usage

• Deploy **Ofrecord** in production environment -> proxy to 'regular' controller

• **Always-on** OF messages, control plane, data plane summaries

• Alter selection rules as necessary

• Deploy **Ofreplay** in lab environment

• Localize bugs / validate bug fixes
Case studies

1. Debugging Black box components
   • CPU inflation in an OpenFlow switch

2. Debugging OpenFlow controllers
   • NOX problem

+ Others (see poster/paper)
Back to CPU inflation

- Replay and bisect the trace by message type
Back to CPU inflation

- Replay and bisect the trace by message type
- When replaying STATS_REQ msgs...

**Record** • **Replay**

STTS_REQ msgs reproduce the problem even though there is no correlation in arrival times
Debugging controllers: NOX problem

- Problem record: Messages initiated by one specific device don't reach NOX controller module
- Not reproducible at the lab
Debugging controllers: NOX problem

- Record at end user site
- Replay at lab towards NOX
- Use host-level debugging to analyze NOX behavior
Debugging controllers: NOX problem

- Trigger: specific source MAC address
  
  - NOX has an 'intelligent' MAC address parser that handles both binary and ASCII MAC addresses
  
  - '0x3a' is the ASCII representation of ':' and appeared in the binary form of this MAC :)

\[
00:26:55:da:3a:40 = 0x3a \text{ == } ':'
\]
Performance Evaluation

- **Record**: production environment
- OFRecord controller performance
- Impact on switch performance
- **Replay**: lab environment
- Timing accuracy
OFRecord controller performance

Median # Flows handled by different controllers (measured with cbench)
Impact on switch performance

- Single UDP packet flows created using hping
- sent to switches of two different vendors
- measure # flows successfully forwarded
- compare OFRecord vs. SimpleCtrl

![Graph showing performance impact]

- Vendor A saturates
- Vendor B breaks down
- OFRecord: limited switch performance penalty

Legend:
- o of–record (Vendor A)
- △ of–record–data (Vendor A)
- + of–simple (Vendor A)
- x of–record (Vendor B)
- ◇ of–record–data (Vendor B)
- ▼ of–simple (Vendor B)
## End-to-end performance

<table>
<thead>
<tr>
<th>Rate [Flows/s]</th>
<th>Drop %</th>
<th>sd (timing) [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0</td>
<td>4.5</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>15.6</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>21.1</td>
</tr>
<tr>
<td>50</td>
<td>0</td>
<td>23.4</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>10.9</td>
</tr>
<tr>
<td>200</td>
<td>0</td>
<td>13.9</td>
</tr>
<tr>
<td>400</td>
<td>19 %</td>
<td>15.8</td>
</tr>
</tbody>
</table>
Summary

New Primitives:

- Selective, consistent, multigranularity Network Recording
- Adaptive coordinated best-effort Network Replay

- reproduce problems at convenient time and place
- Combined in OfRewind, an Open-Flow based tool for Network Record & Replay
  http://www.openflow.org/wk/index.php/OFRewind
- Enables practical record and replay of network domains
Future work

• Scale to larger topology sizes, more complex networks
• Extend to production quality tool
  • Improve timing for very fast flow rates
• Automated regression tests through standard sets of traces
Thank you.
Summary

New Primitives:

- **reproduce** problems at convenient time and location
- Combined in **OfRewind**, an Open-Flow based tool for Network, Record & Replay
- Enables practical record and replay of network domains