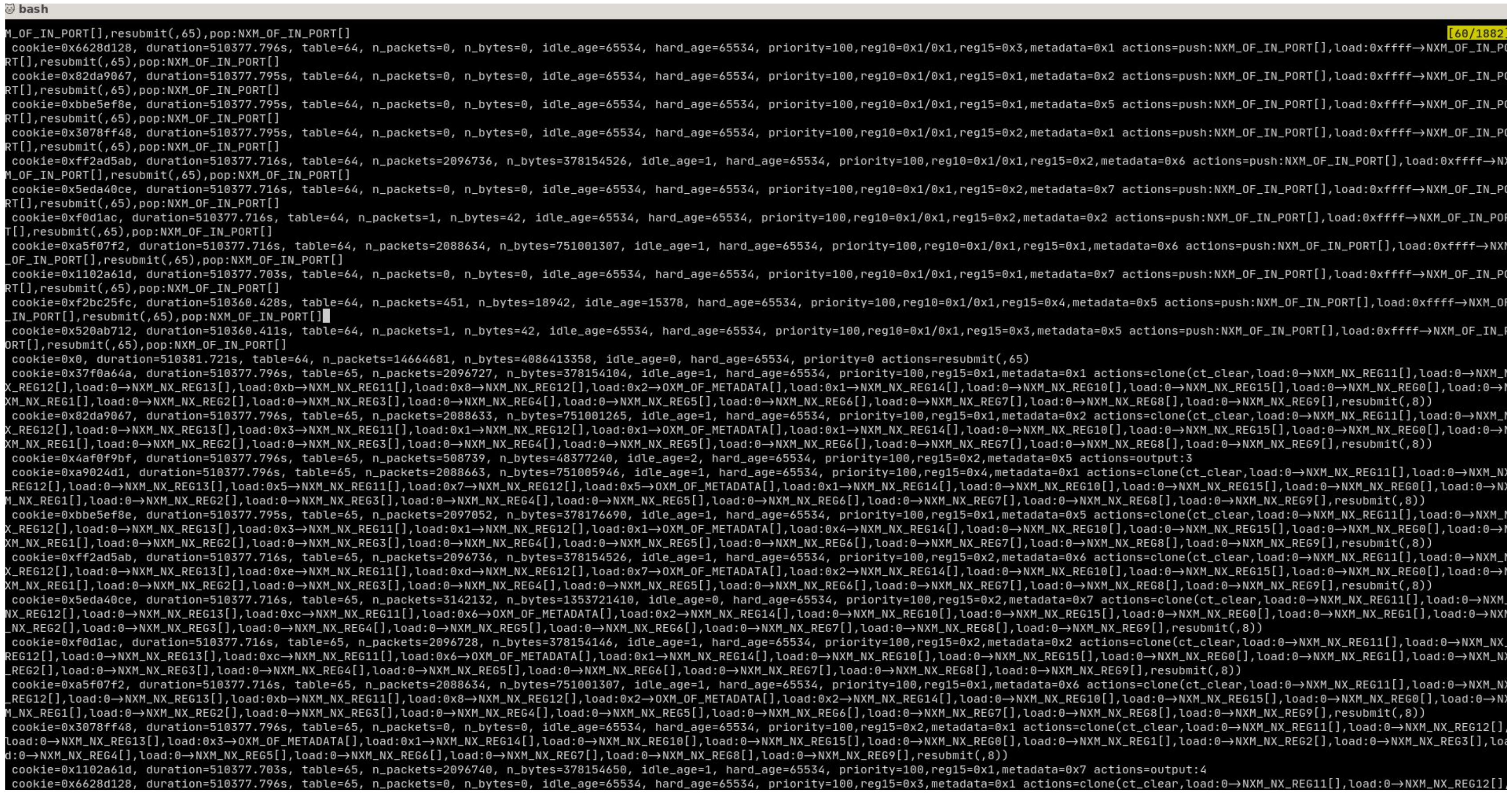
OVS Conference 2021: Improving online and offline flow debugging with ofparse and ovs-offline

Adrián Moreno

Salvatore Daniele

Openflow and datapath flows are difficult to “look at” and reason about

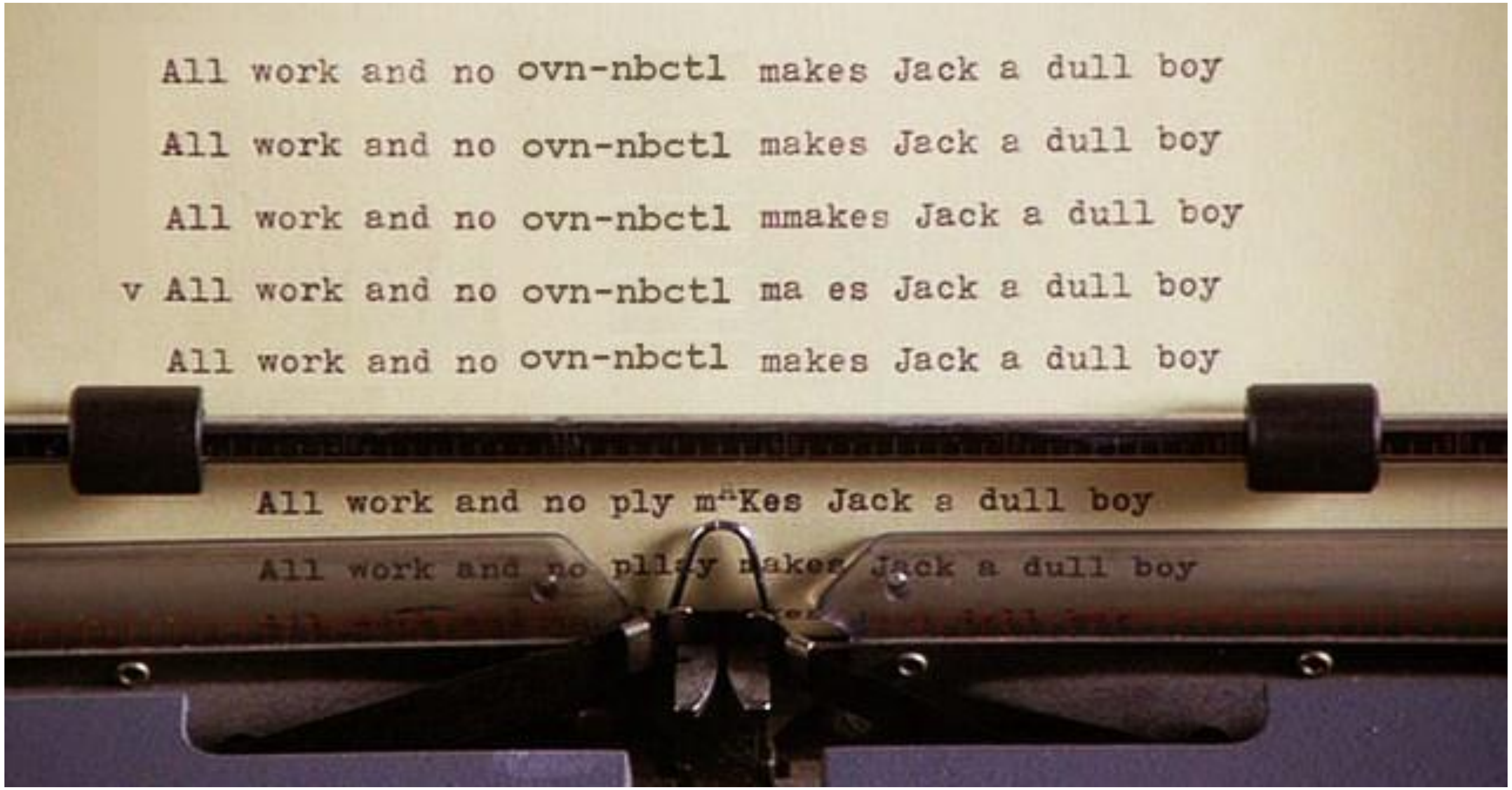


bash

```
M_OF_IN_PORT[],resubmit(,65),pop:NXM_OF_IN_PORT[]                                                                                                                                    [60/1882
 cookie=0x6628d128, duration=510377.796s, table=64, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534, priority=100,reg10=0x1/0x1,reg15=0x3,metadata=0x1 actions=push:NXM_OF_IN_PORT[],load:0xffff→NXM_OF_IN_PO
RT[],resubmit(,65),pop:NXM_OF_IN_PORT[]
 cookie=0x82da9067, duration=510377.795s, table=64, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534, priority=100,reg10=0x1/0x1,reg15=0x1,metadata=0x2 actions=push:NXM_OF_IN_PORT[],load:0xffff→NXM_OF_IN_PO
RT[],resubmit(,65),pop:NXM_OF_IN_PORT[]
 cookie=0xbbe5ef8e, duration=510377.795s, table=64, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534, priority=100,reg10=0x1/0x1,reg15=0x1,metadata=0x5 actions=push:NXM_OF_IN_PORT[],load:0xffff→NXM_OF_IN_PO
RT[],resubmit(,65),pop:NXM_OF_IN_PORT[]
 cookie=0x3078ff48, duration=510377.795s, table=64, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534, priority=100,reg10=0x1/0x1,reg15=0x2,metadata=0x1 actions=push:NXM_OF_IN_PORT[],load:0xffff→NXM_OF_IN_PO
RT[],resubmit(,65),pop:NXM_OF_IN_PORT[]
 cookie=0xff2ad5ab, duration=510377.716s, table=64, n_packets=2096736, n_bytes=378154526, idle_age=1, hard_age=65534, priority=100,reg10=0x1/0x1,reg15=0x2,metadata=0x6 actions=push:NXM_OF_IN_PORT[],load:0xffff→NX
M_OF_IN_PORT[],resubmit(,65),pop:NXM_OF_IN_PORT[]
 cookie=0x5eda40ce, duration=510377.716s, table=64, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534, priority=100,reg10=0x1/0x1,reg15=0x2,metadata=0x7 actions=push:NXM_OF_IN_PORT[],load:0xffff→NXM_OF_IN_PO
RT[],resubmit(,65),pop:NXM_OF_IN_PORT[]
 cookie=0xf0d1ac, duration=510377.716s, table=64, n_packets=1, n_bytes=42, idle_age=65534, hard_age=65534, priority=100,reg10=0x1/0x1,reg15=0x2,metadata=0x2 actions=push:NXM_OF_IN_PORT[],load:0xffff→NXM_OF_IN_POR
T[],resubmit(,65),pop:NXM_OF_IN_PORT[]
 cookie=0xa5f07f2, duration=510377.716s, table=64, n_packets=2088634, n_bytes=751001307, idle_age=1, hard_age=65534, priority=100,reg10=0x1/0x1,reg15=0x1,metadata=0x6 actions=push:NXM_OF_IN_PORT[],load:0xffff→NXM
_OF_IN_PORT[],resubmit(,65),pop:NXM_OF_IN_PORT[]
 cookie=0x1102a61d, duration=510377.703s, table=64, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534, priority=100,reg10=0x1/0x1,reg15=0x1,metadata=0x7 actions=push:NXM_OF_IN_PORT[],load:0xffff→NXM_OF_IN_PO
RT[],resubmit(,65),pop:NXM_OF_IN_PORT[]
 cookie=0xf2bc25fc, duration=510360.428s, table=64, n_packets=451, n_bytes=18942, idle_age=15378, hard_age=65534, priority=100,reg10=0x1/0x1,reg15=0x4,metadata=0x5 actions=push:NXM_OF_IN_PORT[],load:0xffff→NXM_OF
_IN_PORT[],resubmit(,65),pop:NXM_OF_IN_PORT[]
 cookie=0x520ab712, duration=510360.411s, table=64, n_packets=1, n_bytes=42, idle_age=65534, hard_age=65534, priority=100,reg10=0x1/0x1,reg15=0x3,metadata=0x5 actions=push:NXM_OF_IN_PORT[],load:0xffff→NXM_OF_IN_P
ORT[],resubmit(,65),pop:NXM_OF_IN_PORT[]
 cookie=0x0, duration=510381.721s, table=64, n_packets=14664681, n_bytes=4086413358, idle_age=0, hard_age=65534, priority=0 actions=resubmit(,65)
 cookie=0x37f0a64a, duration=510377.796s, table=65, n_packets=2096727, n_bytes=378154104, idle_age=1, hard_age=65534, priority=100,reg15=0x1,metadata=0x1 actions=clone(ct_clear,load:0→NXM_NX_REG11[],load:0→NXM_N
X_REG12[],load:0→NXM_NX_REG13[],load:0xb→NXM_NX_REG11[],load:0x8→NXM_NX_REG12[],load:0x2→OXM_OF_METADATA[],load:0x1→NXM_NX_REG14[],load:0→NXM_NX_REG10[],load:0→NXM_NX_REG15[],load:0→NXM_NX_REG0[],load:0→N
XM_NX_REG1[],load:0→NXM_NX_REG2[],load:0→NXM_NX_REG3[],load:0→NXM_NX_REG4[],load:0→NXM_NX_REG5[],load:0→NXM_NX_REG6[],load:0→NXM_NX_REG7[],load:0→NXM_NX_REG8[],load:0→NXM_NX_REG9[],resubmit(,8))
 cookie=0x82da9067, duration=510377.796s, table=65, n_packets=2088633, n_bytes=751001265, idle_age=1, hard_age=65534, priority=100,reg15=0x1,metadata=0x2 actions=clone(ct_clear,load:0→NXM_NX_REG11[],load:0→NXM_N
X_REG12[],load:0→NXM_NX_REG13[],load:0x3→NXM_NX_REG11[],load:0x1→NXM_NX_REG12[],load:0x1→OXM_OF_METADATA[],load:0x1→NXM_NX_REG14[],load:0→NXM_NX_REG10[],load:0→NXM_NX_REG15[],load:0→NXM_NX_REG0[],load:0→N
XM_NX_REG1[],load:0→NXM_NX_REG2[],load:0→NXM_NX_REG3[],load:0→NXM_NX_REG4[],load:0→NXM_NX_REG5[],load:0→NXM_NX_REG6[],load:0→NXM_NX_REG7[],load:0→NXM_NX_REG8[],load:0→NXM_NX_REG9[],resubmit(,8))
 cookie=0x4af0f9bf, duration=510377.796s, table=65, n_packets=508739, n_bytes=48377240, idle_age=2, hard_age=65534, priority=100,reg15=0x2,metadata=0x5 actions=output:3
 cookie=0xa9024d1, duration=510377.796s, table=65, n_packets=2088663, n_bytes=751005946, idle_age=1, hard_age=65534, priority=100,reg15=0x4,metadata=0x1 actions=clone(ct_clear,load:0→NXM_NX_REG11[],load:0→NXM_NX
_REG12[],load:0→NXM_NX_REG13[],load:0x5→NXM_NX_REG11[],load:0x7→NXM_NX_REG12[],load:0x5→OXM_OF_METADATA[],load:0x1→NXM_NX_REG14[],load:0→NXM_NX_REG10[],load:0→NXM_NX_REG15[],load:0→NXM_NX_REG0[],load:0→NX
M_NX_REG1[],load:0→NXM_NX_REG2[],load:0→NXM_NX_REG3[],load:0→NXM_NX_REG4[],load:0→NXM_NX_REG5[],load:0→NXM_NX_REG6[],load:0→NXM_NX_REG7[],load:0→NXM_NX_REG8[],load:0→NXM_NX_REG9[],resubmit(,8))
 cookie=0xbbe5ef8e, duration=510377.795s, table=65, n_packets=2097052, n_bytes=378176690, idle_age=1, hard_age=65534, priority=100,reg15=0x1,metadata=0x5 actions=clone(ct_clear,load:0→NXM_NX_REG11[],load:0→NXM_N
X_REG12[],load:0→NXM_NX_REG13[],load:0x3→NXM_NX_REG11[],load:0x1→NXM_NX_REG12[],load:0x1→OXM_OF_METADATA[],load:0x4→NXM_NX_REG14[],load:0→NXM_NX_REG10[],load:0→NXM_NX_REG15[],load:0→NXM_NX_REG0[],load:0→N
XM_NX_REG1[],load:0→NXM_NX_REG2[],load:0→NXM_NX_REG3[],load:0→NXM_NX_REG4[],load:0→NXM_NX_REG5[],load:0→NXM_NX_REG6[],load:0→NXM_NX_REG7[],load:0→NXM_NX_REG8[],load:0→NXM_NX_REG9[],resubmit(,8))
 cookie=0xff2ad5ab, duration=510377.716s, table=65, n_packets=2096736, n_bytes=378154526, idle_age=1, hard_age=65534, priority=100,reg15=0x2,metadata=0x6 actions=clone(ct_clear,load:0→NXM_NX_REG11[],load:0→NXM_N
X_REG12[],load:0→NXM_NX_REG13[],load:0xe→NXM_NX_REG11[],load:0xd→NXM_NX_REG12[],load:0x7→OXM_OF_METADATA[],load:0x2→NXM_NX_REG14[],load:0→NXM_NX_REG10[],load:0→NXM_NX_REG15[],load:0→NXM_NX_REG0[],load:0→N
XM_NX_REG1[],load:0→NXM_NX_REG2[],load:0→NXM_NX_REG3[],load:0→NXM_NX_REG4[],load:0→NXM_NX_REG5[],load:0→NXM_NX_REG6[],load:0→NXM_NX_REG7[],load:0→NXM_NX_REG8[],load:0→NXM_NX_REG9[],resubmit(,8))
 cookie=0x5eda40ce, duration=510377.716s, table=65, n_packets=3142132, n_bytes=1353721410, idle_age=0, hard_age=65534, priority=100,reg15=0x2,metadata=0x7 actions=clone(ct_clear,load:0→NXM_NX_REG11[],load:0→NXM_
NX_REG12[],load:0→NXM_NX_REG13[],load:0xc→NXM_NX_REG11[],load:0x6→OXM_OF_METADATA[],load:0x2→NXM_NX_REG14[],load:0→NXM_NX_REG10[],load:0→NXM_NX_REG15[],load:0→NXM_NX_REG0[],load:0→NXM_NX_REG1[],load:0→NXM
_NX_REG2[],load:0→NXM_NX_REG3[],load:0→NXM_NX_REG4[],load:0→NXM_NX_REG5[],load:0→NXM_NX_REG6[],load:0→NXM_NX_REG7[],load:0→NXM_NX_REG8[],load:0→NXM_NX_REG9[],resubmit(,8))
 cookie=0xf0d1ac, duration=510377.716s, table=65, n_packets=2096728, n_bytes=378154146, idle_age=1, hard_age=65534, priority=100,reg15=0x2,metadata=0x2 actions=clone(ct_clear,load:0→NXM_NX_REG11[],load:0→NXM_NX_
REG12[],load:0→NXM_NX_REG13[],load:0xc→NXM_NX_REG11[],load:0x6→OXM_OF_METADATA[],load:0x1→NXM_NX_REG14[],load:0→NXM_NX_REG10[],load:0→NXM_NX_REG15[],load:0→NXM_NX_REG0[],load:0→NXM_NX_REG1[],load:0→NXM_NX
_REG2[],load:0→NXM_NX_REG3[],load:0→NXM_NX_REG4[],load:0→NXM_NX_REG5[],load:0→NXM_NX_REG6[],load:0→NXM_NX_REG7[],load:0→NXM_NX_REG8[],load:0→NXM_NX_REG9[],resubmit(,8))
 cookie=0xa5f07f2, duration=510377.716s, table=65, n_packets=2088634, n_bytes=751001307, idle_age=1, hard_age=65534, priority=100,reg15=0x1,metadata=0x6 actions=clone(ct_clear,load:0→NXM_NX_REG11[],load:0→NXM_NX
_REG12[],load:0→NXM_NX_REG13[],load:0xb→NXM_NX_REG11[],load:0x8→NXM_NX_REG12[],load:0x2→OXM_OF_METADATA[],load:0x2→NXM_NX_REG14[],load:0→NXM_NX_REG10[],load:0→NXM_NX_REG15[],load:0→NXM_NX_REG0[],load:0→NX
M_NX_REG1[],load:0→NXM_NX_REG2[],load:0→NXM_NX_REG3[],load:0→NXM_NX_REG4[],load:0→NXM_NX_REG5[],load:0→NXM_NX_REG6[],load:0→NXM_NX_REG7[],load:0→NXM_NX_REG8[],load:0→NXM_NX_REG9[],resubmit(,8))
 cookie=0x3078ff48, duration=510377.796s, table=65, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534, priority=100,reg15=0x2,metadata=0x1 actions=clone(ct_clear,load:0→NXM_NX_REG11[],load:0→NXM_NX_REG12[],
load:0→NXM_NX_REG13[],load:0x3→OXM_OF_METADATA[],load:0x1→NXM_NX_REG14[],load:0→NXM_NX_REG10[],load:0→NXM_NX_REG15[],load:0→NXM_NX_REG0[],load:0→NXM_NX_REG1[],load:0→NXM_NX_REG2[],load:0→NXM_NX_REG3[],loa
d:0→NXM_NX_REG4[],load:0→NXM_NX_REG5[],load:0→NXM_NX_REG6[],load:0→NXM_NX_REG7[],load:0→NXM_NX_REG8[],load:0→NXM_NX_REG9[],resubmit(,8))
 cookie=0x1102a61d, duration=510377.703s, table=65, n_packets=2096740, n_bytes=378154650, idle_age=1, hard_age=65534, priority=100,reg15=0x1,metadata=0x7 actions=output:4
 cookie=0x6628d128, duration=510377.796s, table=65, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534, priority=100,reg15=0x3,metadata=0x1 actions=clone(ct_clear,load:0→NXM_NX_REG11[],load:0→NXM_NX_REG12[],
```

Openflow and datapath flows are difficult to “look at” and reason about

You’re favorite tools are often not available



All work and no ovn-nbctl makes Jack a dull boy

All work and no ovn-nbctl makes Jack a dull boy

All work and no ovn-nbctl mmakes Jack a dull boy

v All work and no ovn-nbctl ma es Jack a dull boy

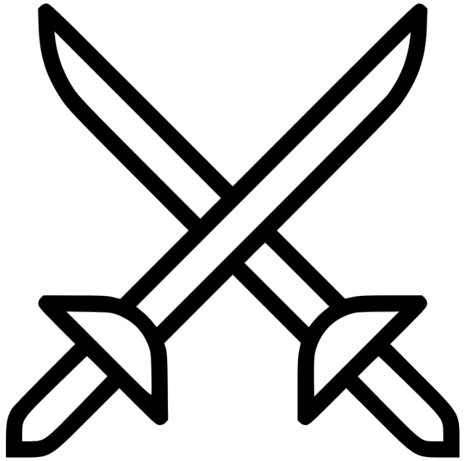
All work and no ovn-nbctl makes Jack a dull boy

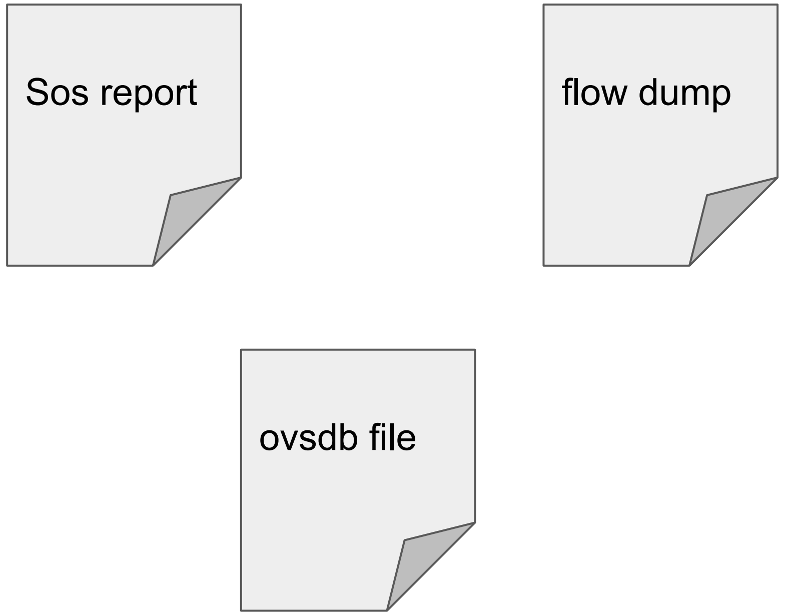
All work and no ply m"Kes Jack a dull boy

All work and no pllay makes Jack a dull boy

You’re favorite tools are often not available

- ovn-nbctl
- ovn-sbctl
- ovs-vsctl
- ovs-ofctl
- ovs-appctl





Tools to help out!

- ovs-ofparse: Parsing and Visualizing flows
  - Python library: v1 series posted
  - ovs-ofparse tool: rfc posted
- ovs-offline: create offline daemons based on logs

Parsing Flows

What, why and how?

- Create a python flow parsing library based on flow strings.
  - v1 series posted
- Create a python flow visualization tool for openflow and datapath flows
  - rfc posted
  - I'll demo it in a minute so bare with me :)

Feedback welcome!

Flow Parsing Library

Parsing flow strings

- Flows are mostly* simple Key->Value pairs split by ":", "=", "()", ...
  - *table=8, arp_spa=10.244.1.0/24, in_port(eth0),* ...
- Or Lists: *resubmit(,4)*
  - Which, given a bit of context: "**resubmit**([port],[table][,ct])"
  - Can be converted into Key->Value pairs
    - *resubmit={port="", table=4}*

*Need to be able to adapt to special cases

Flow Parsing Library

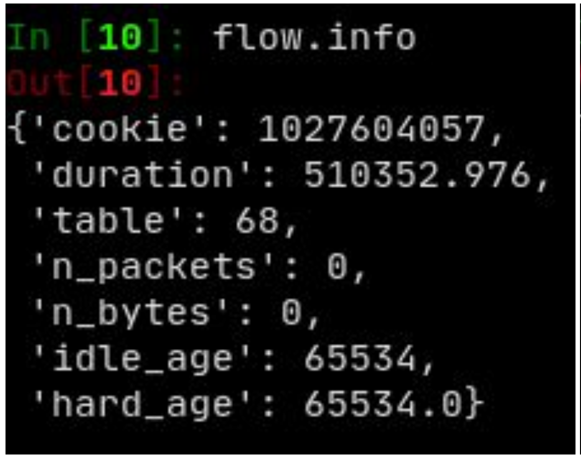
Anatomy of the parsing library

- KeyValue + Metadata (where in the string was found)
- KVParser + KVDecoders
  - For each key -> a function capable of decoding its value
- ListParser + ListDecoders
  - For each position -> a key name and a function capable of decoding its value
- Special types:
  - IntegerMask, IPMask, EthMask

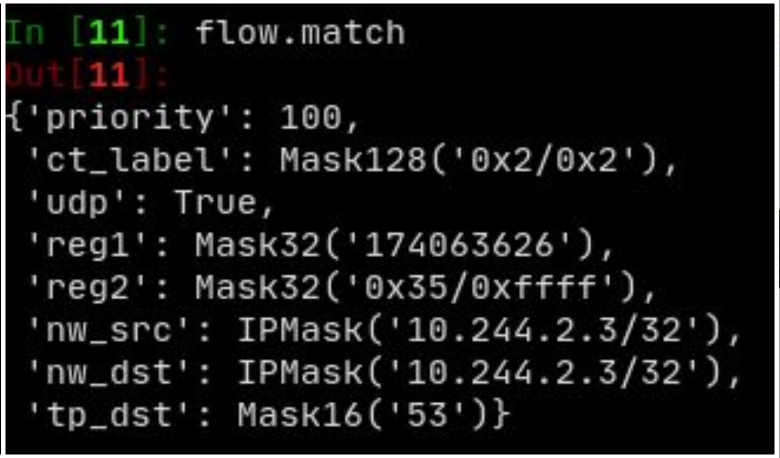
Flow Parsing Library

Example

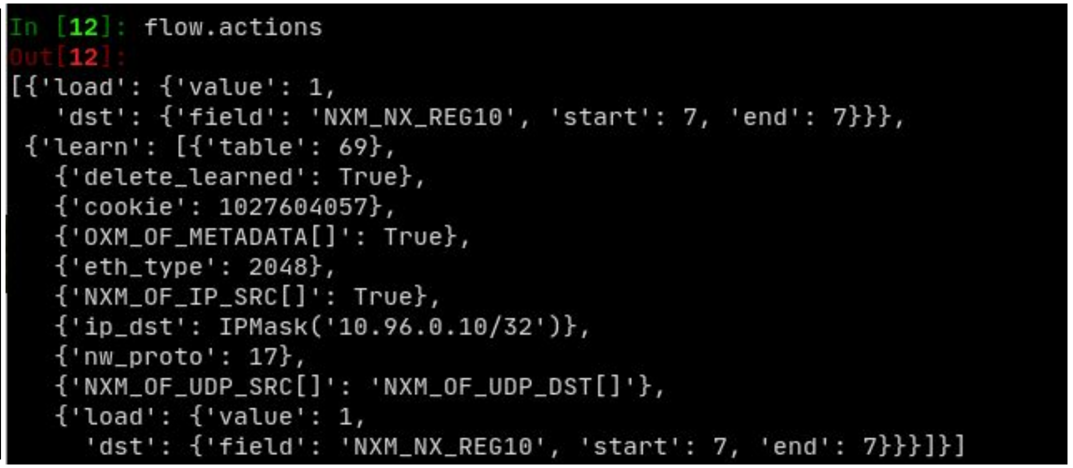
cookie=0x3d3ffe59, duration=510352.976s, table=68, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534, priority=100,ct_label=0x2/0x2,udp,reg1=0xa60000a,reg2=0x35/0xffff,nw_src=10.244.2.3,nw_dst=10.244.2.3,tp_dst=53 actions=load:0x1->NXM_NX_REG10[7],learn(table=69,delete_learned,cookie=0x3d3ffe59,OXM_OF_METADATA[],eth_type=0x800,NXM_OF_IP_SRC[],ip_dst=10.96.0.10,nw_proto=17,NXM_OF_UDP_SRC[]=NXM_OF_UDP_DST[],load:0x1->NXM_NX_REG10[7])



```
In [10]: flow.info
Out[10]:
{'cookie': 1027604057,
 'duration': 510352.976,
 'table': 68,
 'n_packets': 0,
 'n_bytes': 0,
 'idle_age': 65534,
 'hard_age': 65534.0}
```



```
In [11]: flow.match
Out[11]:
{'priority': 100,
 'ct_label': Mask128('0x2/0x2'),
 'udp': True,
 'reg1': Mask32('174063626'),
 'reg2': Mask32('0x35/0xffff'),
 'nw_src': IPMask('10.244.2.3/32'),
 'nw_dst': IPMask('10.244.2.3/32'),
 'tp_dst': Mask16('53')}
```



```
In [12]: flow.actions
Out[12]:
[{'load': {'value': 1,
   'dst': {'field': 'NXM_NX_REG10', 'start': 7, 'end': 7}}},
 {'learn': [{'table': 69},
   {'delete_learned': True},
   {'cookie': 1027604057},
   {'OXM_OF_METADATA[]': True},
   {'eth_type': 2048},
   {'NXM_OF_IP_SRC[]': True},
   {'ip_dst': IPMask('10.96.0.10/32')},
   {'nw_proto': 17},
   {'NXM_OF_UDP_SRC[]': 'NXM_OF_UDP_DST[]'},
   {'load': {'value': 1,
     'dst': {'field': 'NXM_NX_REG10', 'start': 7, 'end': 7}}}]}]
```

Flow Parsing Library

Filtering

[! | not] {key}[[.subkey]...] [COMPARISON {value}] [LOGICAL OPERATOR] ...

- Logical Operators
  - **not**, **!**
  - **and**, **&&**
  - **or**, **||**

- Comparisons
  - **=**, **>**, **<**
  - **~=** masking (value/mask in flow "contains" provided value)
  - (omit to check for existence of the key)

Flow Parsing Library

Filtering examples

table=8, n_packets=42, nw_dst=192.168.1.0/24,tcp actions=output

- "table>2": ✅
- "table = 10 or n_packets =42": ✅
- "n_packets > 0 && output": ✅
- "! tcp": ❌
- "nw_dst = 192.168.1.5": ❌
- "nw_dst ~= 192.168.1.5": ✅

ovs-ofparse

overview

ovs-ofparse [GLOBAL OPTIONS] openflow | datapath FORMAT [OPTIONS]

- Extensive styling
  - Define what color to use for each key-value
  - Highlight any key-value
  - Heat-map
- Filtering

- Multiple inputs
- Multiple output formats
  - console, json, html, logic...

ovs-ofparse

ovs-offline

- https://ovs-dbg.readthedocs.io/en/latest/ovs-offline.html
- Recreate an offline replica of a OVS or OVN daemon
- Pick up OVS/OVN information from:
  - Live-system
  - Sosreport
- Run ovs-vsctl, ovn-nbctl, ovn-sbctl, ovs-ofproto, even 'ovs-appctl ofproto/trace' as if you were in the live system
- Useful for post-mortem and customer issues

Usage

- Collect data
  - OpenFlow flow dumps
  - OpenFlow group dumps
  - TLV maps.
  - Databases
- ovs-offline start

Demo

- Openshift cluster
- Collect from sosreport
- Debugging with ovs-offline

Limitations of ovs-offline

- Ofproto/trace
  - Lack conntrack
  - No ovn-controller
- No Kernel
  - dummy system
  - datapath

Next Steps

- Suggestions?
  - https://github.com/amorenoz/ovs-dbg