Netmap with Bro

Seth Hall
ICSI, Bro Project, Corelight

What is netmap?

- Mechanism for bypassing kernel
- Batched operations on packets
- Cross platform API
  - Linux as external module
  - Built into FreeBSD

Historical Problems

- **Difficulty in installing patched drivers for high performance**
  - Mostly solved by making netmap download and patch Intel drivers. Remaining drivers can be installed with kernel source.
- **Lack of load balancing mechanism.**
  - Welcome ‘lb’!
- **Bugs**
  - Many have been ironed out in the past year.

Supported Native Driver (without kernel source!)

- i40e
- ixgbevf
- ixgbe
- igb
- e1000e

Netmap “interface” names

```
* ifname  (netmap:foo or vale:foo) is the port name
*    a suffix can indicate the follwing:
*    ^    bind the host (sw) ring pair
*    *    bind host and NIC ring pairs (transparent)
*    -NN     bind individual NIC ring pair
*    {NN     bind master side of pipe NN
*    }NN     bind slave side of pipe NN
*    a suffix starting with / and the following flags,
*    in any order:
*    x    exclusive access
*    z    zero copy monitor
*    t    monitor tx side
*    r    monitor rx side
*    R    bind only RX ring(s)
*    T    bind only TX ring(s)
```

Netmap “interface” names

Using: https://github.com/luigirizzo/netmap-libpcap

```
Read from Netmap Pipe:
 # tcpdump –i netmap:eth1}0
Zero Copy Interface access:
 # tcpdump –i netmap:eth1/Rz
Connect to NIC ring:
 # tcpdump –i netmap:eth1–4
```

Bro configuration
node.cfg

First install the Bro netmap plugin from:
aux/plugins/netmap

```
[worker-1]
type=worker
host=localhost
interface=netmap::bro
lb_method=custom
lb_procs=3
```

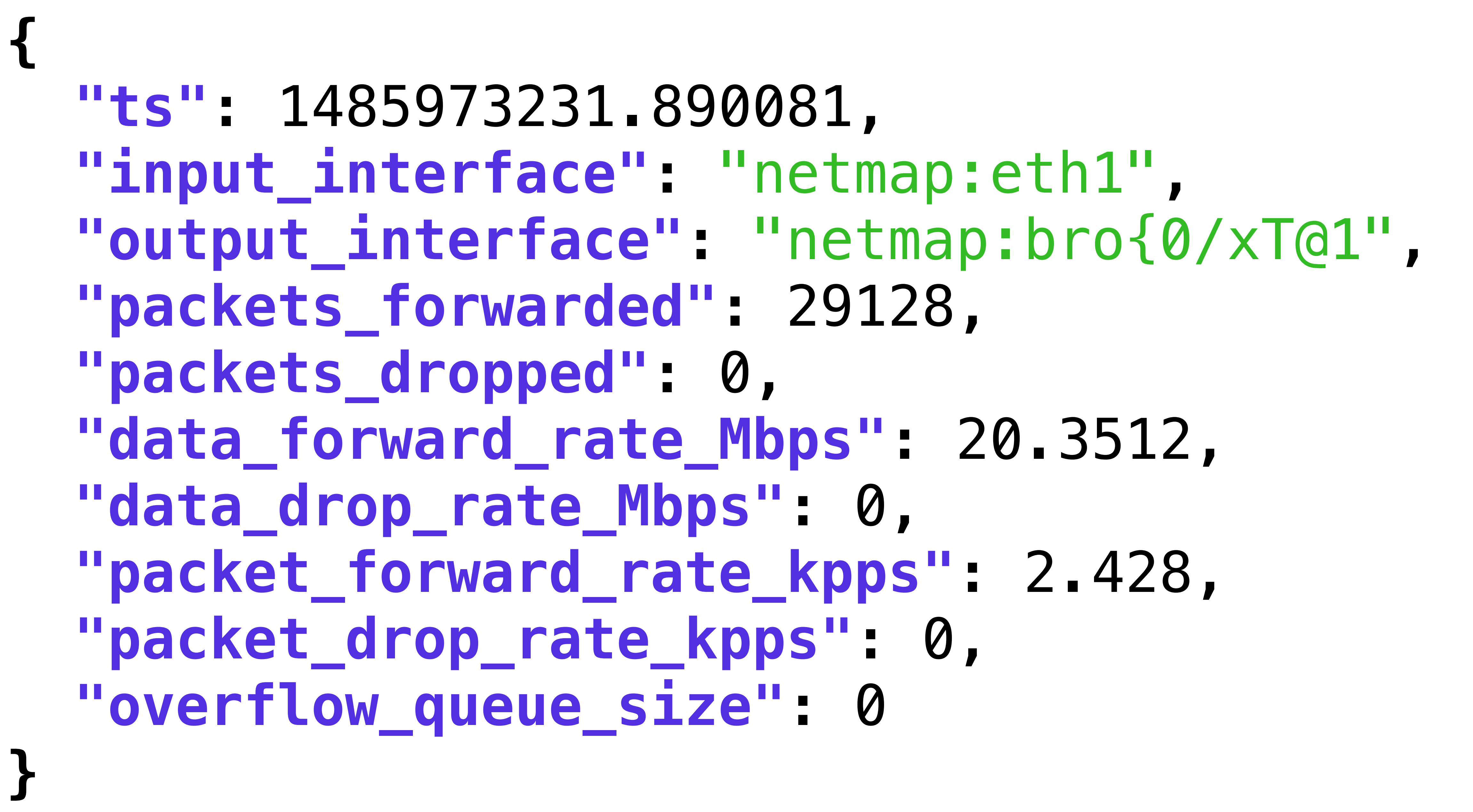
Notice double quotes in interface!
It means we’re using the Bro Netmap plugin.

Run lb

```
usage: lb [options]
where options are:
  -h               view help text
  -i iface         interface name (required)
  -p [prefix:]npipes add a new group of output pipes
  -B nbufs         number of extra buffers (default: 0)
  -b batch         batch size (default: 2048)
  -w seconds         wait for link up (default: 2)
  -s seconds       seconds between syslog stats messages
(default: 0)
  -o seconds       seconds between stdout stats messages
(default: 0)
```

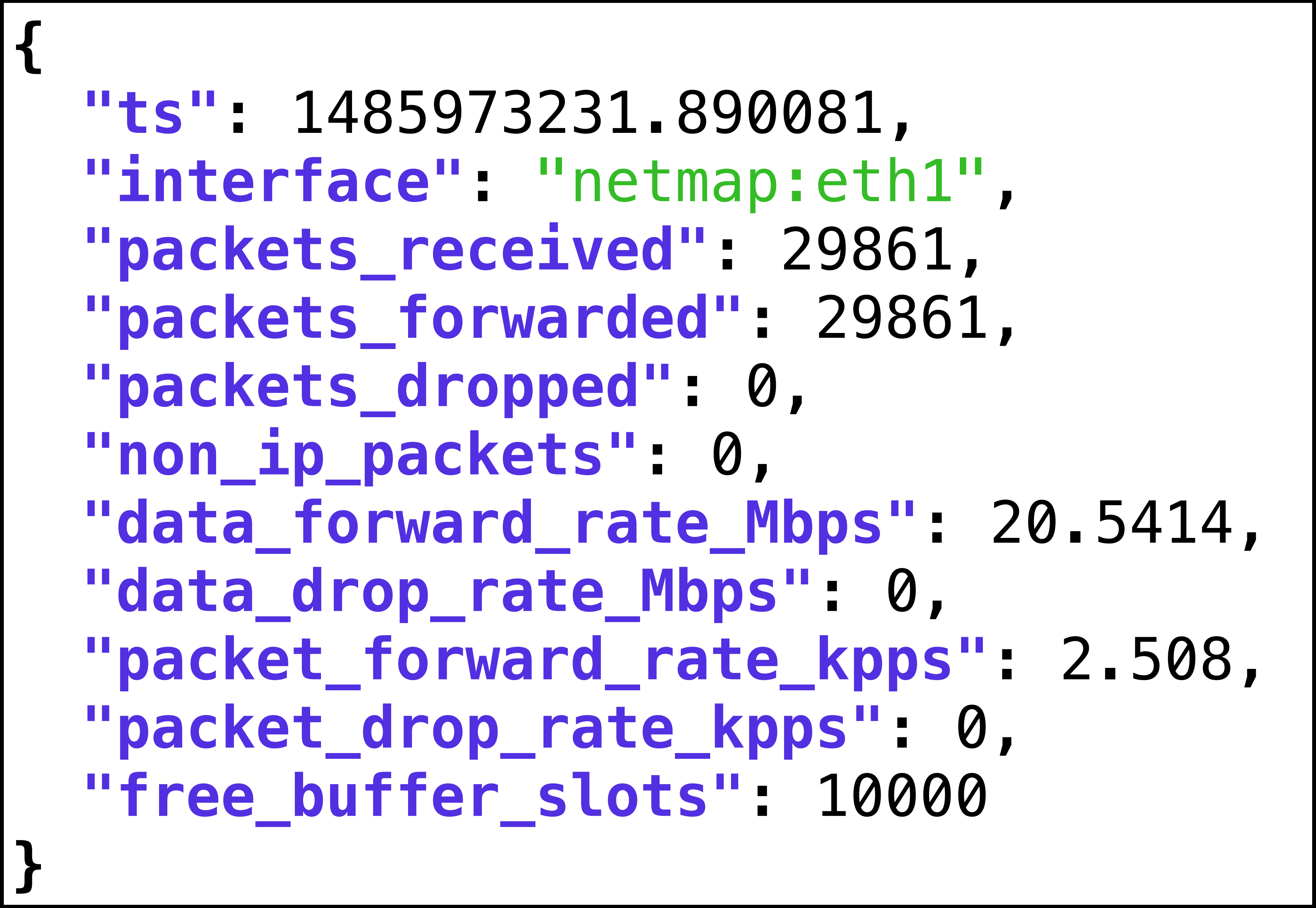
```
sudo lb -i eth1 -p bro:3 -B 10000 -o 1
```

Ring Stats!



```
{
  "ts": 1485973231.890081,
  "input_interface": "netmap:eth1",
  "output_interface": "netmap:bro{0/xT@1",
  "packets_forwarded": 29128,
  "packets_dropped": 0,
  "data_forward_rate_Mbps": 20.3512,
  "data_drop_rate_Mbps": 0,
  "packet_forward_rate_kpps": 2.428,
  "packet_drop_rate_kpps": 0,
  "overflow_queue_size": 0
}
```

Overall Stats!



```
{
  "ts": 1485973231.890081,
  "interface": "netmap:eth1",
  "packets_received": 29861,
  "packets_forwarded": 29861,
  "packets_dropped": 0,
  "non_ip_packets": 0,
  "data_forward_rate_Mbps": 20.5414,
  "data_drop_rate_Mbps": 0,
  "packet_forward_rate_kpps": 2.508,
  "packet_drop_rate_kpps": 0,
  "free_buffer_slots": 10000
}
```

Another lb example

```
lb –i eth1 –p bro:3 –p snort:3
```

Will give these the same packets….

```
tcpdump –i netmap:bro}1
```

```
tcpdump –i netmap:snort}1
```

Multigroup load balancing!

Resources and Links

- Main netmap: https://github.com/luigirizzo/netmap
- Netmap libpcap: https://github.com/luigirizzo/netmap-libpcap
- Updated lb: https://github.com/corelight/netmap/tree/corelight_updates
    - Changes here will be integrated back into the main netmap repo eventually, working on it now.

Thank you!

- Contact me:
  - seth@icir.org
  - seth@corelight.com
- twitter: @remor
  - Also: @Bro_IDS, @Corelight_Inc