

How to Write Your Own NIC Driver (and why)

Our experience writing 10G/100G drivers
for Snabb in Lua (without NDAs)

Luke Gorrie, Asumu Takikawa

FOSDEM, 3 Feb 2018



snabb



igalia

Why write device drivers?

- Fear of eternal damnation
- Pursuit of destiny
- Lust for power

Driver Heaven & Hell

In "Driver Heaven":

- Spec is 20 pages
- Driver is 500 lines
- Lots of drivers on Github

In "Driver hell":

- Spec is 1k pages and secret
- Driver is 50KLOC + 500KLOC deps
- Nobody understands but vendors
- Feature creep, binary blobs...

Road to hell:

- Use HW without public docs
- Use drivers you don't understand
- Depend on vendors for everything

Road to heaven:

- Insist on hardware docs
- Read and understand
- Write drivers together
- Engage vendors
- Seek out kindred spirits

```
Terminal - lugano-1:luke:~
File Edit View Terminal Tabs Help
lugano-1:luke:~ x lugano-4:luke:~ x Untitled x

Transmissions (last 1 sec):
apps report:
07:00.0 GPTC (Good TX packets) 14,880,844 GPRC (Good RX packets) 14,880,848
03:00.0 GPTC (Good TX packets) 14,880,560 GPRC (Good RX packets) 14,880,560
09:00.1 GPTC (Good TX packets) 14,880,849 GPRC (Good RX packets) 14,880,849
09:00.0 GPTC (Good TX packets) 14,880,832 GPRC (Good RX packets) 14,880,831
05:00.1 GPTC (Good TX packets) 14,880,602 GPRC (Good RX packets) 14,880,603
05:00.0 GPTC (Good TX packets) 14,880,604 GPRC (Good RX packets) 14,880,604
03:00.1 GPTC (Good TX packets) 14,880,543 GPRC (Good RX packets) 14,880,531
01:00.0 GPTC (Good TX packets) 14,880,495 GPRC (Good RX packets) 14,880,496
07:00.1 GPTC (Good TX packets) 14,880,832 GPRC (Good RX packets) 14,880,833
01:00.1 GPTC (Good TX packets) 14,880,515 GPRC (Good RX packets) 14,880,513

Transmissions (last 1 sec):
apps report:
88:00.0 GPTC (Good TX packets) 14,880,880 GPRC (Good RX packets) 14,880,869
84:00.0 GPTC (Good TX packets) 14,880,505 GPRC (Good RX packets) 14,880,505
8a:00.1 GPTC (Good TX packets) 14,880,500 GPRC (Good RX packets) 14,880,488
8a:00.0 GPTC (Good TX packets) 14,880,513 GPRC (Good RX packets) 14,880,500
86:00.1 GPTC (Good TX packets) 14,880,818 GPRC (Good RX packets) 14,880,817
86:00.0 GPTC (Good TX packets) 14,880,816 GPRC (Good RX packets) 14,880,811
84:00.1 GPTC (Good TX packets) 14,880,476 GPRC (Good RX packets) 14,880,468
82:00.0 GPTC (Good TX packets) 14,880,538 GPRC (Good RX packets) 14,880,538
88:00.1 GPTC (Good TX packets) 14,880,852 GPRC (Good RX packets) 14,880,852
82:00.1 GPTC (Good TX packets) 14,880,527 GPRC (Good RX packets) 14,880,525

1 [|||||||100.0%] 7 [ 0.0%] 13 [|||||||100.0%] 19 [ 0.0%]
2 [ 0.0%] 8 [ 0.7%] 14 [ 0.0%] 20 [ 0.0%]
3 [ 0.0%] 9 [ 0.0%] 15 [ 0.0%] 21 [ 0.0%]
4 [ 0.0%] 10 [ 0.0%] 16 [ 0.0%] 22 [ 0.0%]
5 [ 0.0%] 11 [ 0.0%] 17 [ 0.0%] 23 [ 0.0%]
6 [ 0.7%] 12 [ 0.0%] 18 [ 0.0%] 24 [ 0.0%]
Mem[||||||| 8.24G/31.4G] Tasks: 32, 9 thr; 3 running
Swp[ 0K/0K] Load average: 1.94 1.74 1.30
Uptime: 01:51:44

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
9655 root 20 0 339M 12008 4060 R 100.0 0.0 1:23.50 ./snabb packetblaster synth -S
9703 root 20 0 339M 12100 3936 R 99.6 0.0 1:22.62 ./snabb packetblaster synth -S
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit
[0] 0:sudo* 1:bash- 2:htop "grindelwald" 13:24 17-Apr-16
```

Lust for power

- Packetblaster
- Firehose
- Sidespy

Part 2: How

Luke gave you the *why*
(why **driver heaven**)

This part is about *how*
(how Snabb's drivers work)

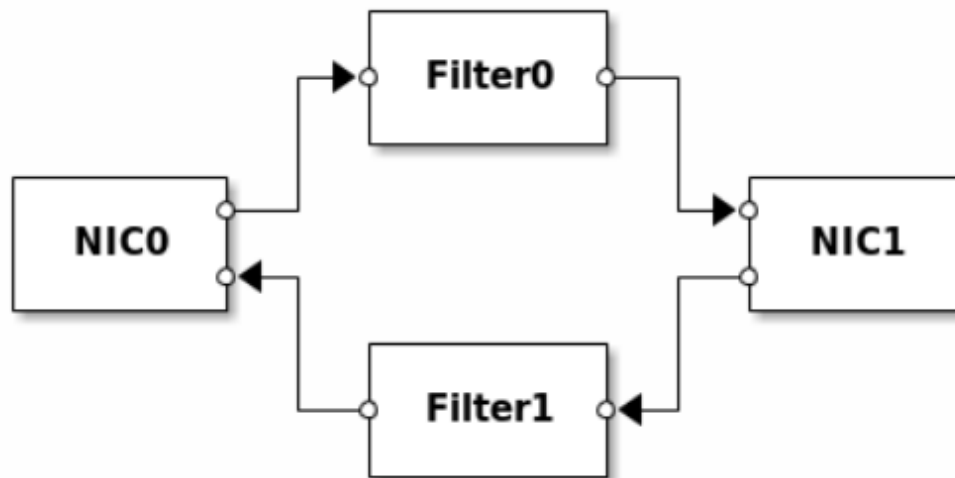
This part just gives a **flavor**
of implementation

Big picture: **1,485 LOC** of Lua
code is pretty high-level

LuaJIT - easy to understand +
abstractions with low cost

Receiving packets

Snabb driver is an **app**
(like everything else)



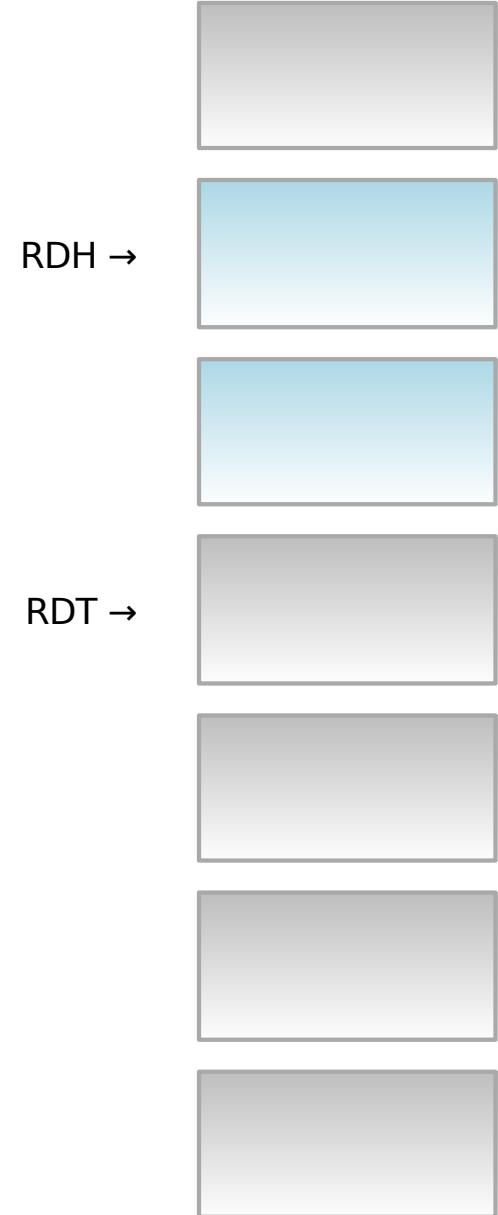
App = obj with some methods
new, push, pull

Let's consider `pull` (Rx)

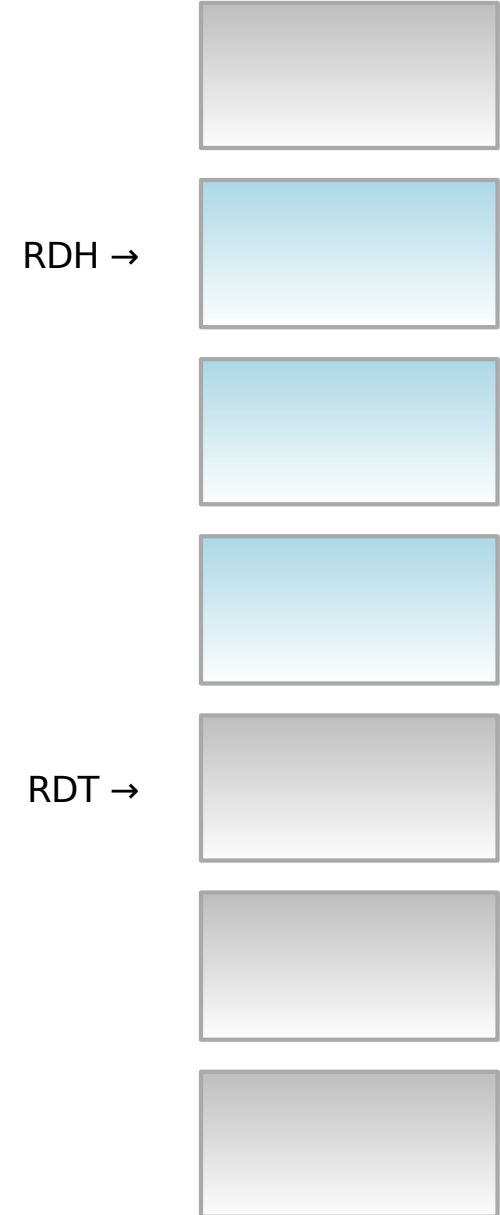
Driver maintains **ring buffer**

NIC **DMA**s pkts to ring

Driver sets base addr register
Maintains tail reg (**RDT**)



Driver sets base addr register
Maintains tail reg (**RDT**)



How to access registers

```
-- self          => driver object
-- self.r        => registers object
-- self.r.RDT    => tail register
self.r.RDT()

-- increment (or loop) tail reg after reading pkt
self.r.RDT(band(self.r.RDT() + 1, ring_size - 1))
```

Ring buffer representation

Address (to memory allocated by driver)				
VLAN tag	Errors	Status	Checksum	Length


```
rxdesc_t = ffi.typeof([[
    struct {
        uint64_t address;
        uint16_t length, cksum;
        uint8_t status, errors;
        uint16_t vlan;
    } __attribute__((packed))
]])
```

-- allocate driver's ring buffer

```
local buffer_size = sizeof(rxdesc_t) * ring_size
self.rxdesc = memory.dma_alloc(buffer_size)
```

```
function Driver:pull ()
    self:sync_receive() -- sync driver & HW pointers

    for i = 1, engine.pull_npackets do
        -- check ptrs for pkt availability
        if not self:can_receive() then break end

        -- move pkt from rx ring to Snabb
        local pkt = self:receive()
        link.transmit(self.output.tx, pkt)
    end

    -- alloc new buffers for ring
    self:add_receive_buffers()
end
```

```
-- method that fetches next packet to read
function Driver:receive()
    -- copy of tail register
    local tail = self.r.RDT()

    -- get length from ring buffer
    -- get actual packet from array of packets
    local len = self.rxdesc[tail]
    local p    = self.rxpackets[tail]
    p.length = len

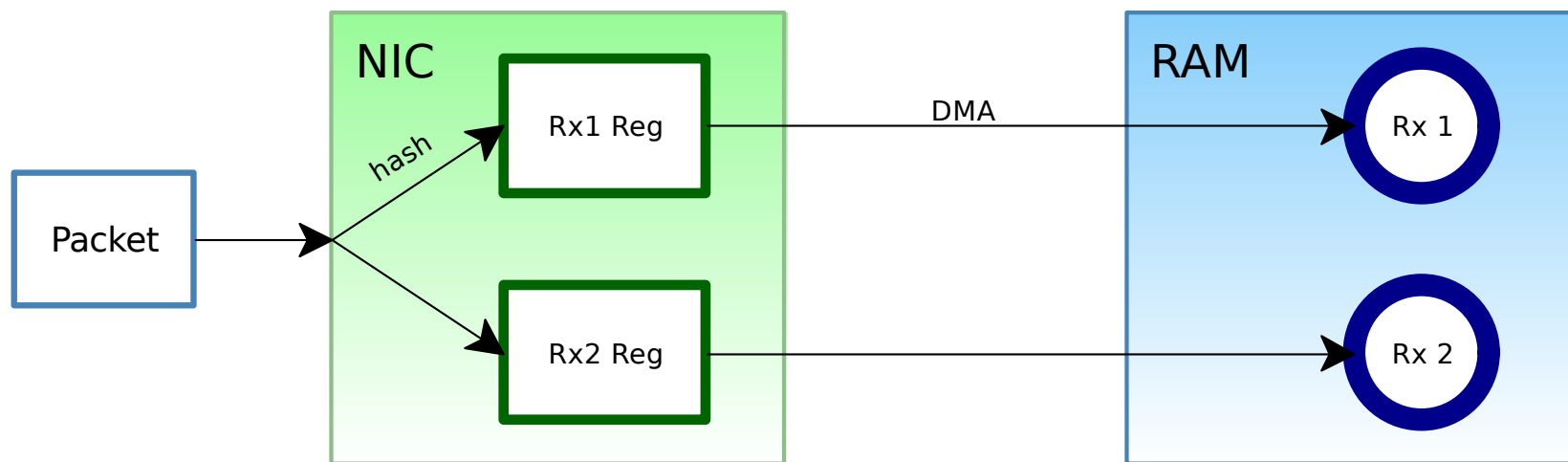
    -- delete packet from array, increment tail
    self.rxpackets[tail] = nil
    self.r.RDT(band(tail + 1, ring_size - 1))

    return p
end
```

Recent & future work

RSS + VMDq support
Scales to multiple cores

Hash flows to distribute to
separate queues



Future: XL710 (40G) support?

Thanks!

<https://github.com/snabbco/snabb>



snabb



igalia