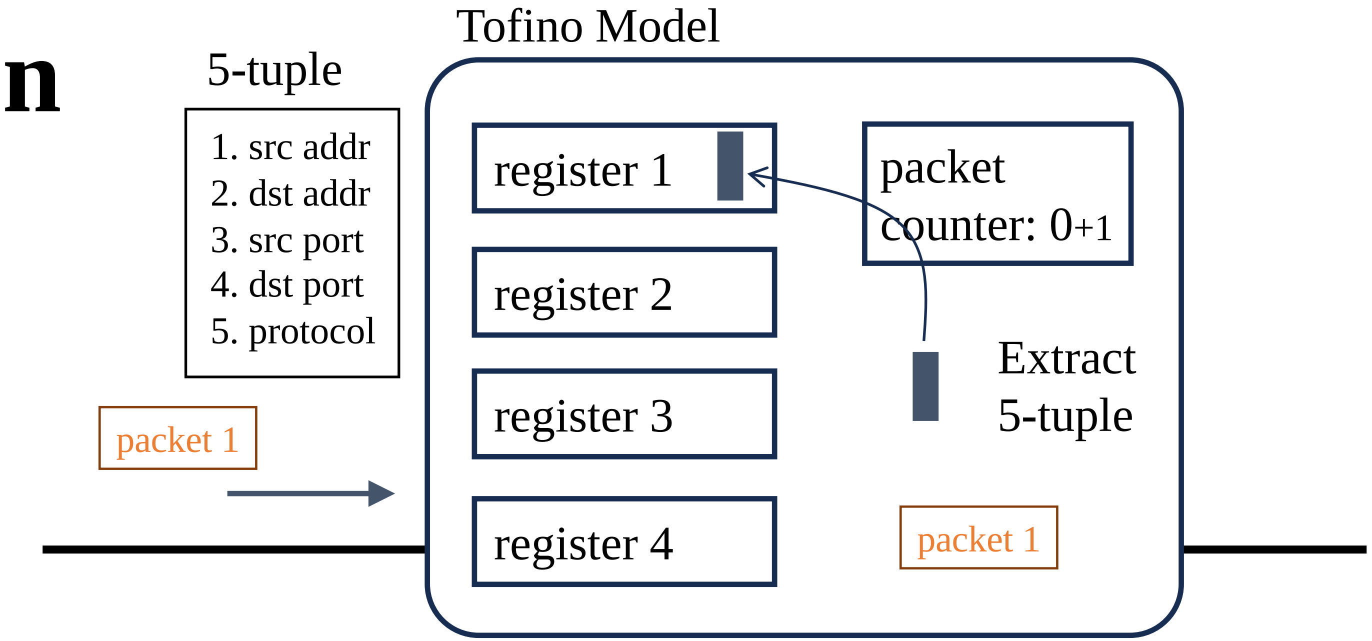
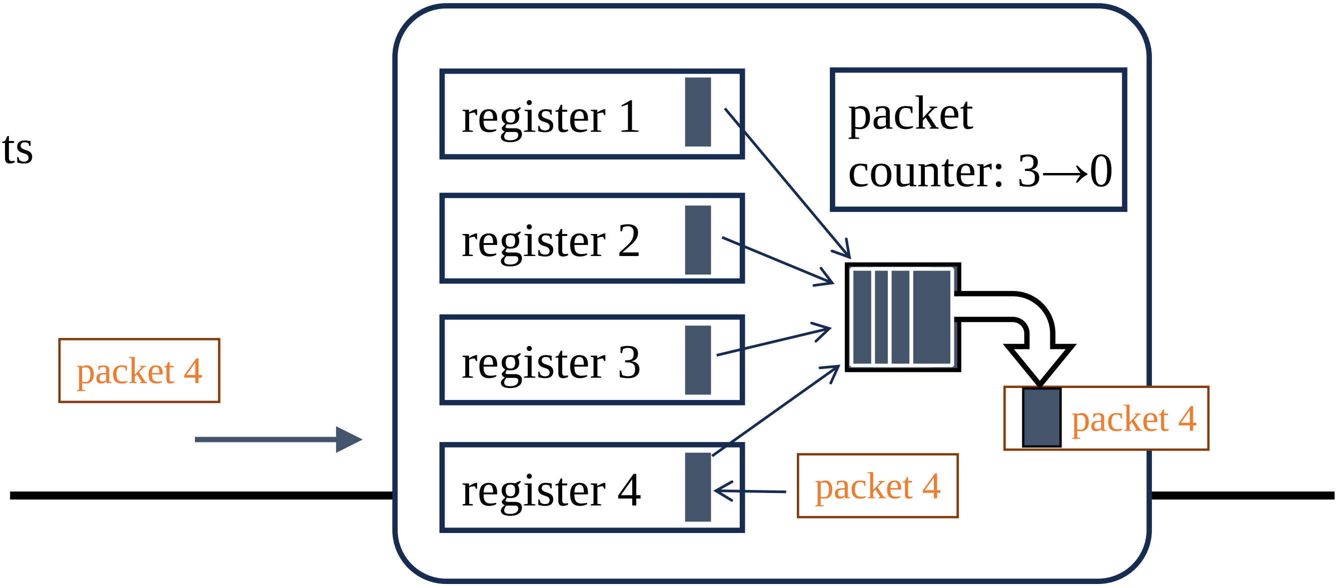
Overall implementation

- **Used registers to save some information**
  - ➢ Prepared 5 registers
    - ✓ To save 5-tuple of each packet
    - ✓ To count the current packet number

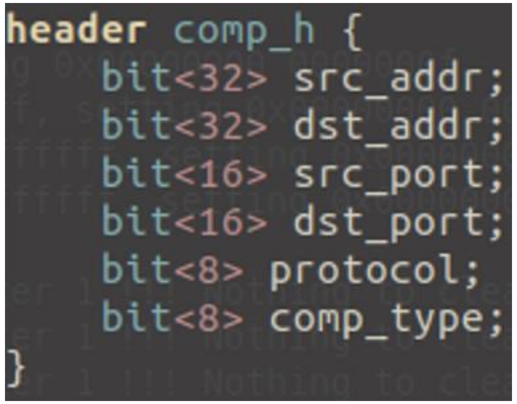
- **Behavior when a packet arrives**
  - ➢ Check the packet counter
  - ➢ Increment the packet counter
  - ➢ Save 5-tuple in the corresponding register number
  - ➢ If the packet counter exceeds the number of packets to be compressed,
    - ✓ Read all registers and get all 5-tuple
    - ✓ Insert into the #4 packet
    - ✓ Reset the packet counter





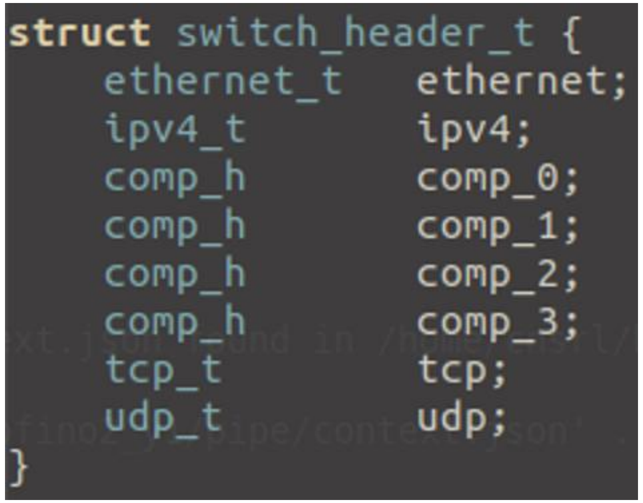
Define comp header

Define header

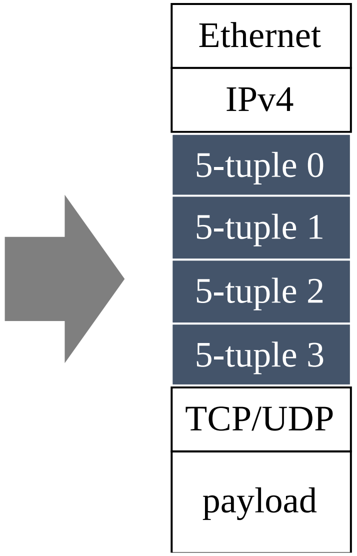


```
header comp_h {
    bit<32> src_addr;
    bit<32> dst_addr;
    bit<16> src_port;
    bit<16> dst_port;
    bit<8> protocol;
    bit<8> comp_type;
}
```

Add header

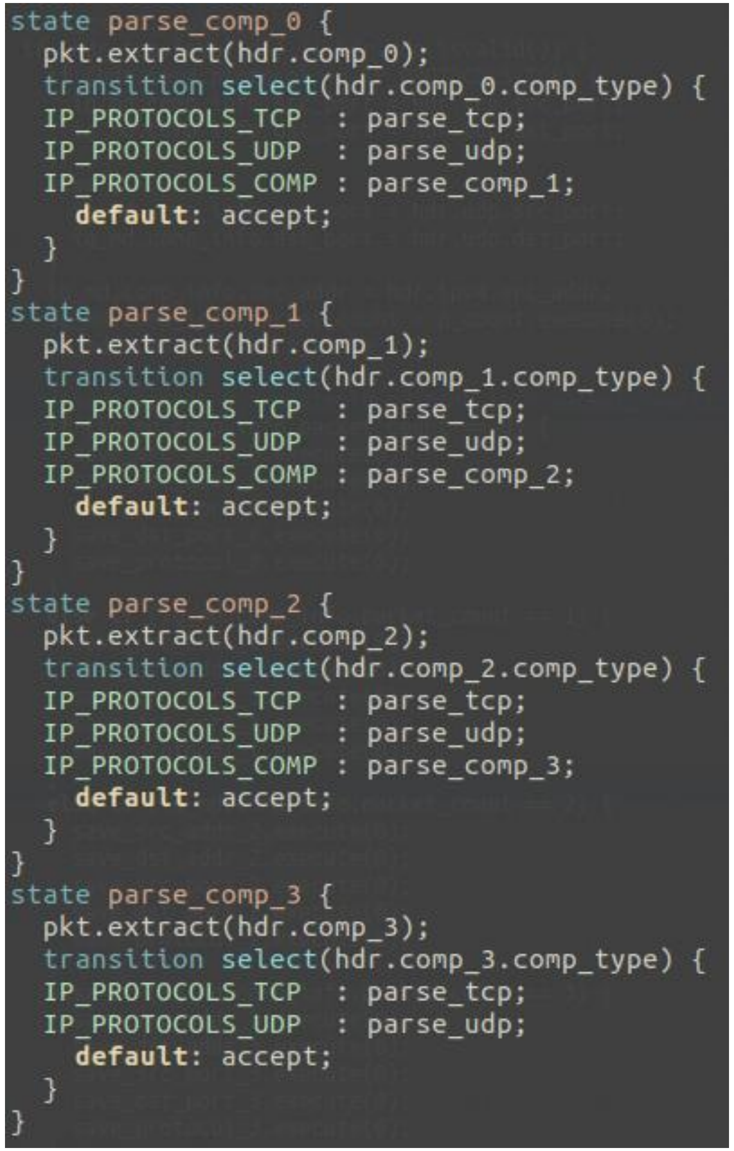


```
struct switch_header_t {
    ethernet_t    ethernet;
    ipv4_t        ipv4;
    comp_h        comp_0;
    comp_h        comp_1;
    comp_h        comp_2;
    comp_h        comp_3;
    tcp_t         tcp;
    udp_t         udp;
}
```

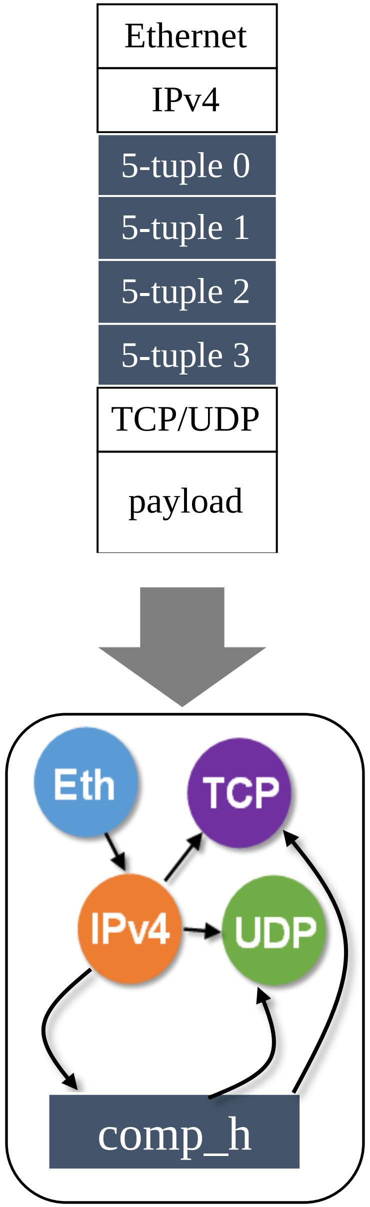


Define parser

Define how to process the header



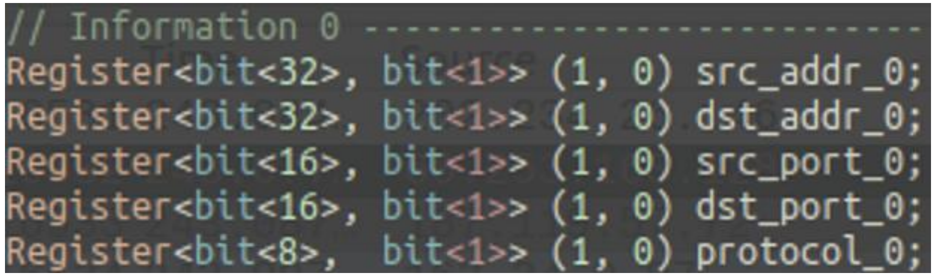
```
state parse_comp_0 {
  pkt.extract(hdr.comp_0);
  transition select(hdr.comp_0.comp_type) {
  IP_PROTOCOLS_TCP  : parse_tcp;
  IP_PROTOCOLS_UDP  : parse_udp;
  IP_PROTOCOLS_COMP : parse_comp_1;
    default: accept;
  }
}
state parse_comp_1 {
  pkt.extract(hdr.comp_1);
  transition select(hdr.comp_1.comp_type) {
  IP_PROTOCOLS_TCP  : parse_tcp;
  IP_PROTOCOLS_UDP  : parse_udp;
  IP_PROTOCOLS_COMP : parse_comp_2;
    default: accept;
  }
}
state parse_comp_2 {
  pkt.extract(hdr.comp_2);
  transition select(hdr.comp_2.comp_type) {
  IP_PROTOCOLS_TCP  : parse_tcp;
  IP_PROTOCOLS_UDP  : parse_udp;
  IP_PROTOCOLS_COMP : parse_comp_3;
    default: accept;
  }
}
state parse_comp_3 {
  pkt.extract(hdr.comp_3);
  transition select(hdr.comp_3.comp_type) {
  IP_PROTOCOLS_TCP  : parse_tcp;
  IP_PROTOCOLS_UDP  : parse_udp;
    default: accept;
  }
}
```



Passes as metadata

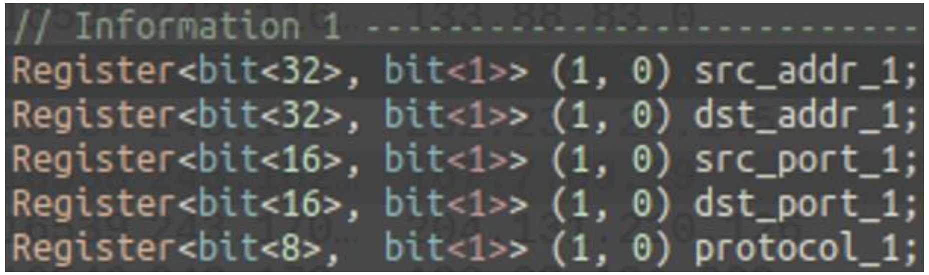
Define register

tuple information 0



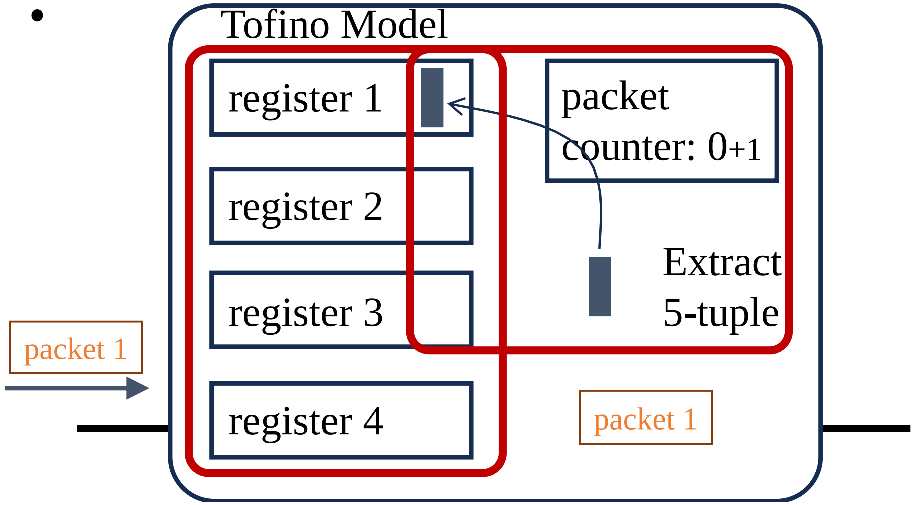
```
// Information 0 ---------------------------
Register<bit<32>, bit<1>> (1, 0) src_addr_0;
Register<bit<32>, bit<1>> (1, 0) dst_addr_0;
Register<bit<16>, bit<1>> (1, 0) src_port_0;
Register<bit<16>, bit<1>> (1, 0) dst_port_0;
Register<bit<8>,  bit<1>> (1, 0) protocol_0;
```

tuple information 1



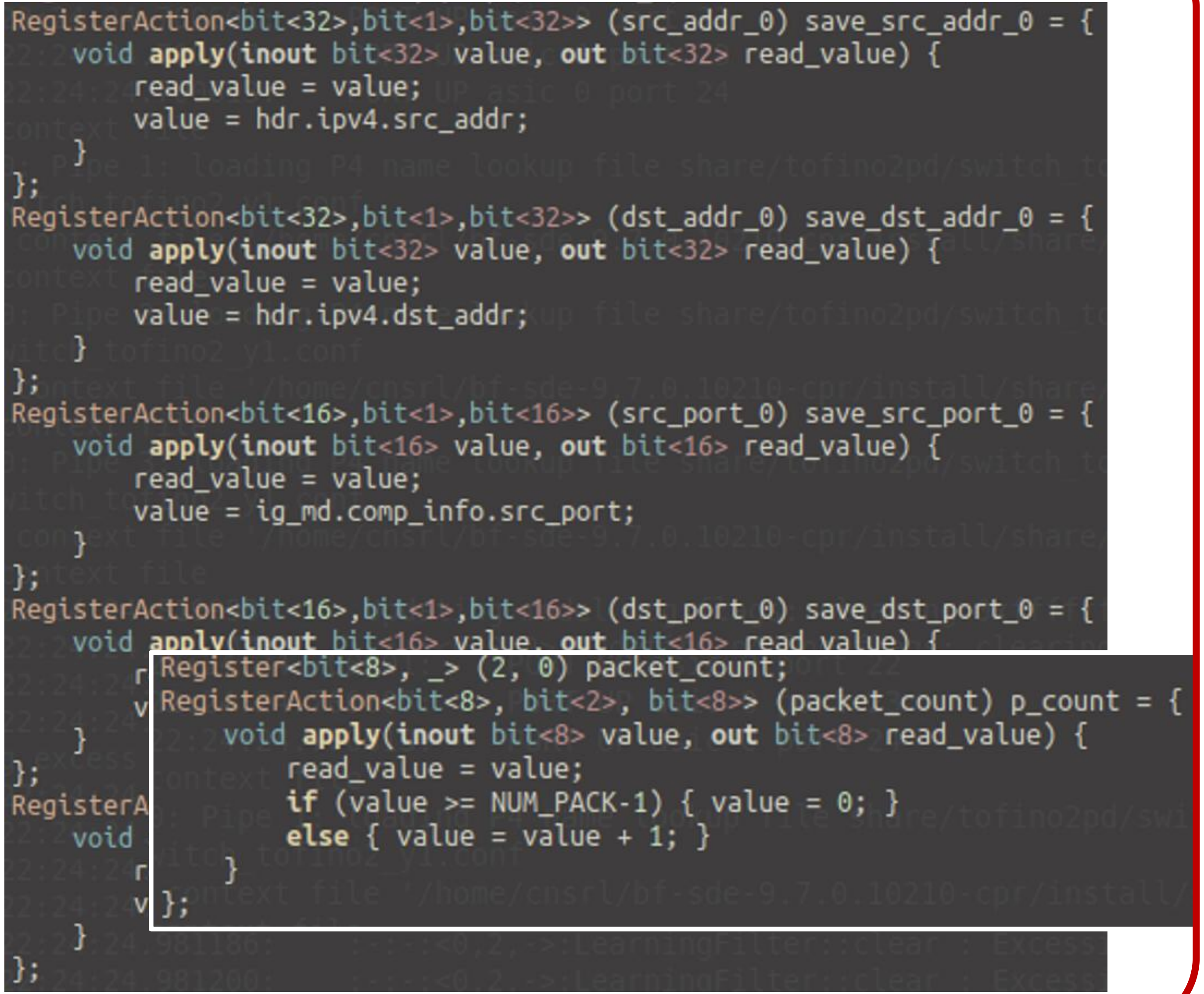
```
// Information 1 ---------------------------
Register<bit<32>, bit<1>> (1, 0) src_addr_1;
Register<bit<32>, bit<1>> (1, 0) dst_addr_1;
Register<bit<16>, bit<1>> (1, 0) src_port_1;
Register<bit<16>, bit<1>> (1, 0) dst_port_1;
Register<bit<8>,  bit<1>> (1, 0) protocol_1;
```

⋮



Define register action

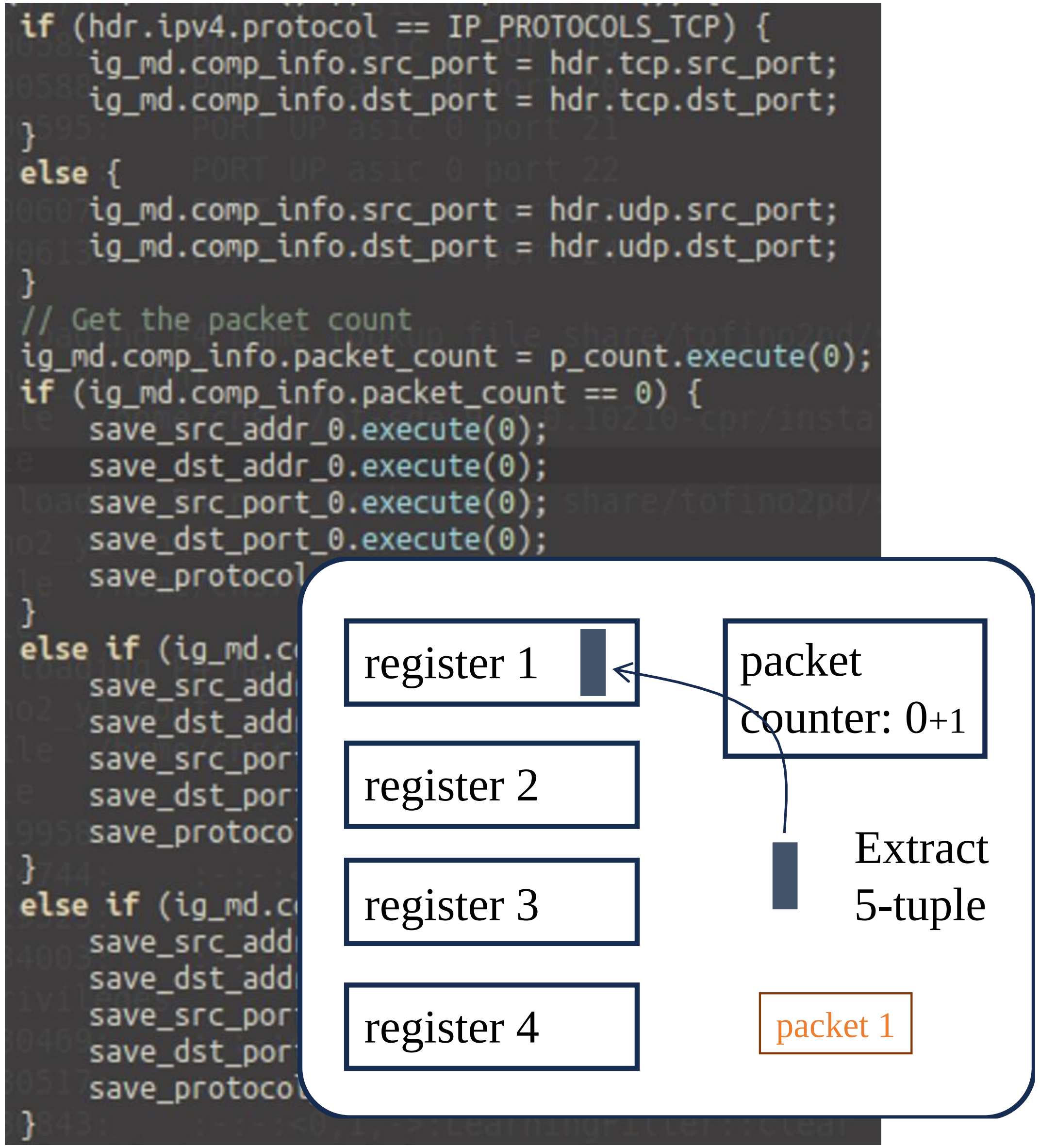
action of tuple information 0



```
RegisterAction<bit<32>,bit<1>,bit<32>> (src_addr_0) save_src_addr_0 = {
    void apply(inout bit<32> value, out bit<32> read_value) {
        read_value = value;
        value = hdr.ipv4.src_addr;
    }
};
RegisterAction<bit<32>,bit<1>,bit<32>> (dst_addr_0) save_dst_addr_0 = {
    void apply(inout bit<32> value, out bit<32> read_value) {
        read_value = value;
        value = hdr.ipv4.dst_addr;
    }
};
RegisterAction<bit<16>,bit<1>,bit<16>> (src_port_0) save_src_port_0 = {
    void apply(inout bit<16> value, out bit<16> read_value) {
        read_value = value;
        value = ig_md.comp_info.src_port;
    }
};
RegisterAction<bit<16>,bit<1>,bit<16>> (dst_port_0) save_dst_port_0 = {
    void apply(inout bit<16> value, out bit<16> read_value) {
        r
        v
    }
};
RegisterA
    void
        r
        v
    }
};
```

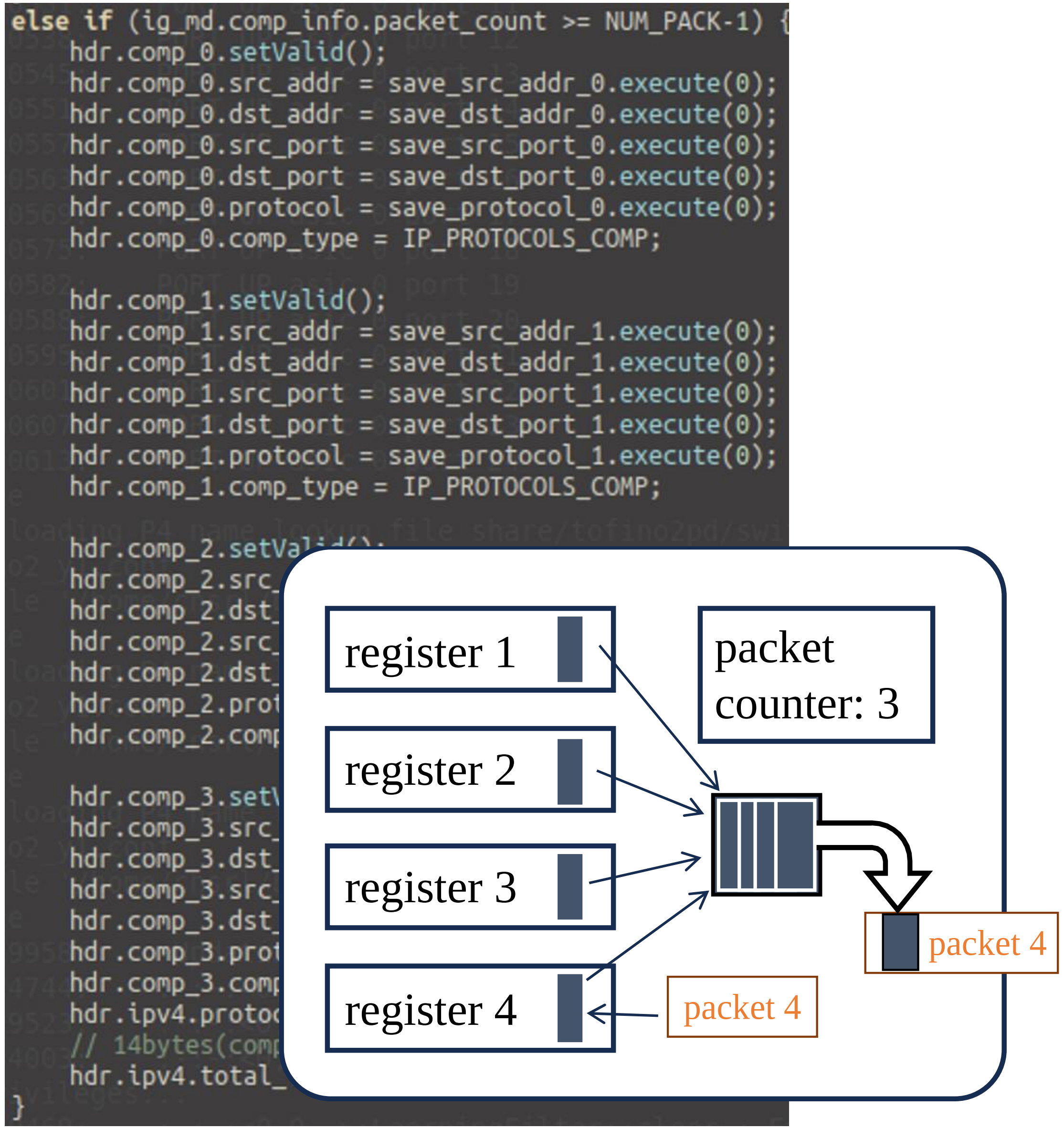
```
Register<bit<8>, _> (2, 0) packet_count;
RegisterAction<bit<8>, bit<2>, bit<8>> (packet_count) p_count = {
    void apply(inout bit<8> value, out bit<8> read_value) {
        read_value = value;
        if (value >= NUM_PACK-1) { value = 0; }
        else { value = value + 1; }
    }
};
```

Save 5-tuple by executing RegisterAction



```
if (hdr.ipv4.protocol == IP_PROTOCOLS_TCP) {
    ig_md.comp_info.src_port = hdr.tcp.src_port;
    ig_md.comp_info.dst_port = hdr.tcp.dst_port;
}
else {
    ig_md.comp_info.src_port = hdr.udp.src_port;
    ig_md.comp_info.dst_port = hdr.udp.dst_port;
}
// Get the packet count
ig_md.comp_info.packet_count = p_count.execute(0);
if (ig_md.comp_info.packet_count == 0) {
    save_src_addr_0.execute(0);
    save_dst_addr_0.execute(0);
    save_src_port_0.execute(0);
    save_dst_port_0.execute(0);
    save_protocol
}
else if (ig_md.c
    save_src_addr
    save_dst_addr
    save_src_port
    save_dst_port
    save_protoco
}
else if (ig_md.c
    save_src_addr
    save_dst_addr
    save_src_port
    save_dst_port
    save_protoco
}
```

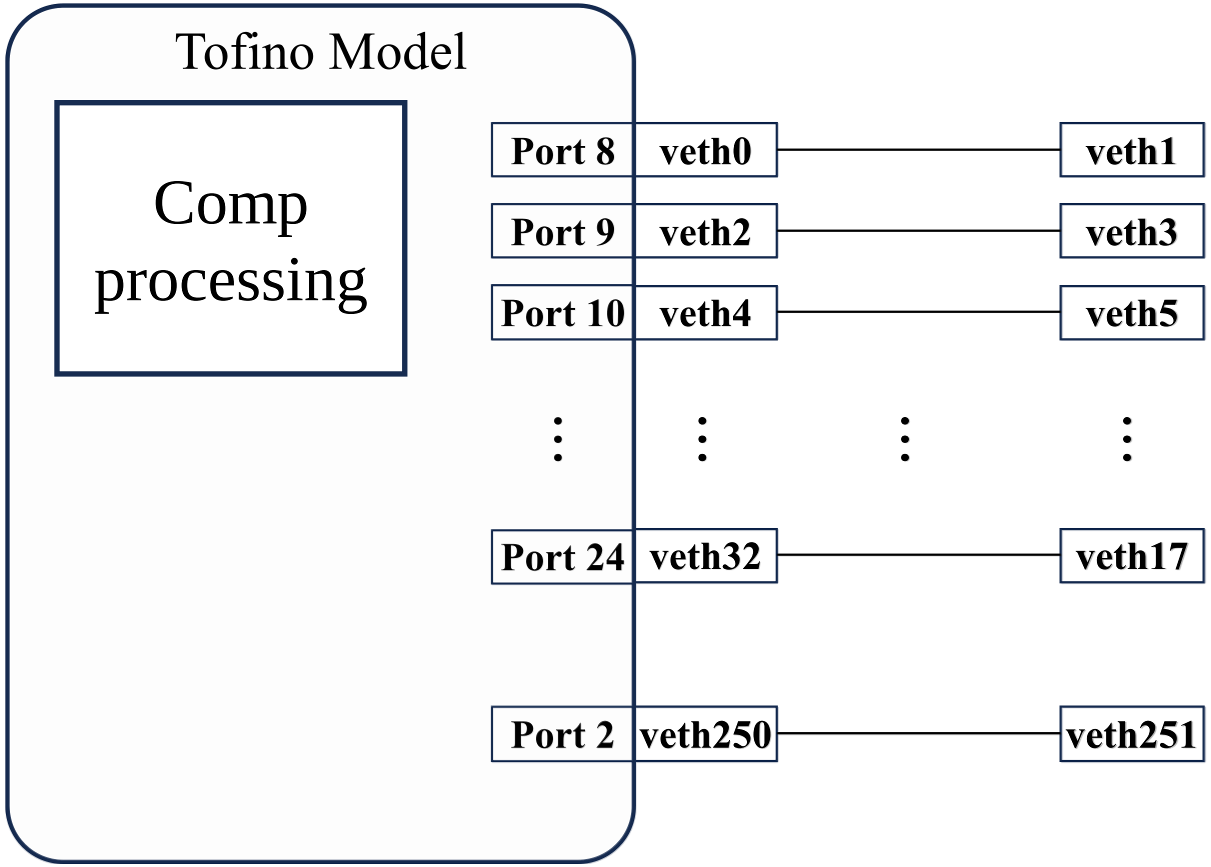
Compress the comp headers into a packet

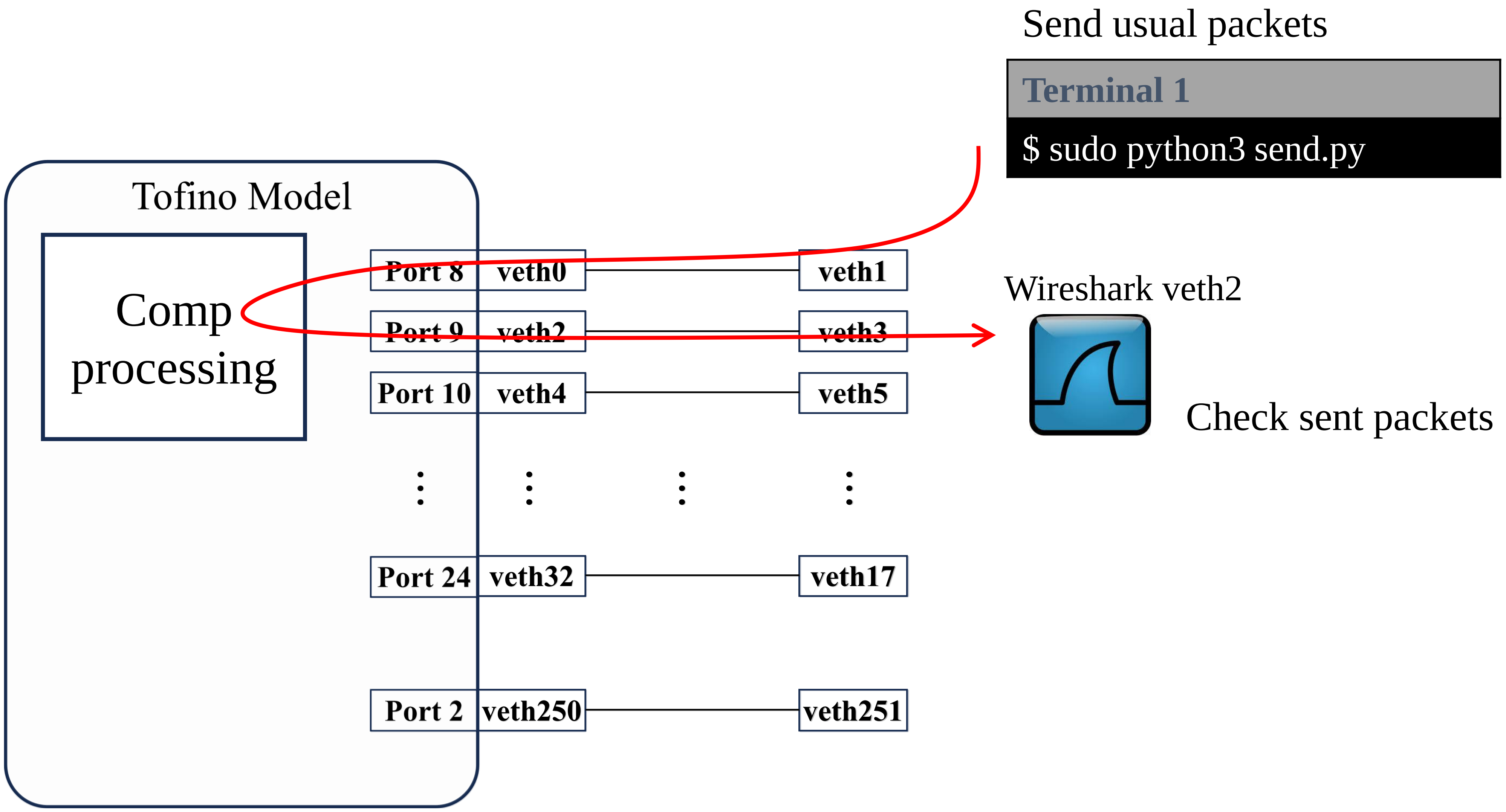


```
else if (ig_md.comp_info.packet_count >= NUM_PACK-1) {
    hdr.comp_0.setValid();
    hdr.comp_0.src_addr = save_src_addr_0.execute(0);
    hdr.comp_0.dst_addr = save_dst_addr_0.execute(0);
    hdr.comp_0.src_port = save_src_port_0.execute(0);
    hdr.comp_0.dst_port = save_dst_port_0.execute(0);
    hdr.comp_0.protocol = save_protocol_0.execute(0);
    hdr.comp_0.comp_type = IP_PROTOCOLS_COMP;

    hdr.comp_1.setValid();
    hdr.comp_1.src_addr = save_src_addr_1.execute(0);
    hdr.comp_1.dst_addr = save_dst_addr_1.execute(0);
    hdr.comp_1.src_port = save_src_port_1.execute(0);
    hdr.comp_1.dst_port = save_dst_port_1.execute(0);
    hdr.comp_1.protocol = save_protocol_1.execute(0);
    hdr.comp_1.comp_type = IP_PROTOCOLS_COMP;

    hdr.comp_2.setValid();
    hdr.comp_2.src
    hdr.comp_2.dst
    hdr.comp_2.src
    hdr.comp_2.dst
    hdr.comp_2.prot
    hdr.comp_2.comp

    hdr.comp_3.set
    hdr.comp_3.src
    hdr.comp_3.dst
    hdr.comp_3.src
    hdr.comp_3.dst
    hdr.comp_3.prot
    hdr.comp_3.comp
    hdr.ipv4.protoc
    // 14bytes(comp
    hdr.ipv4.total_
}
```

Model

- **Used the Tofino model and driver to check the “compression header program”**
  - ➢ Create the environment below:





Send usual packets

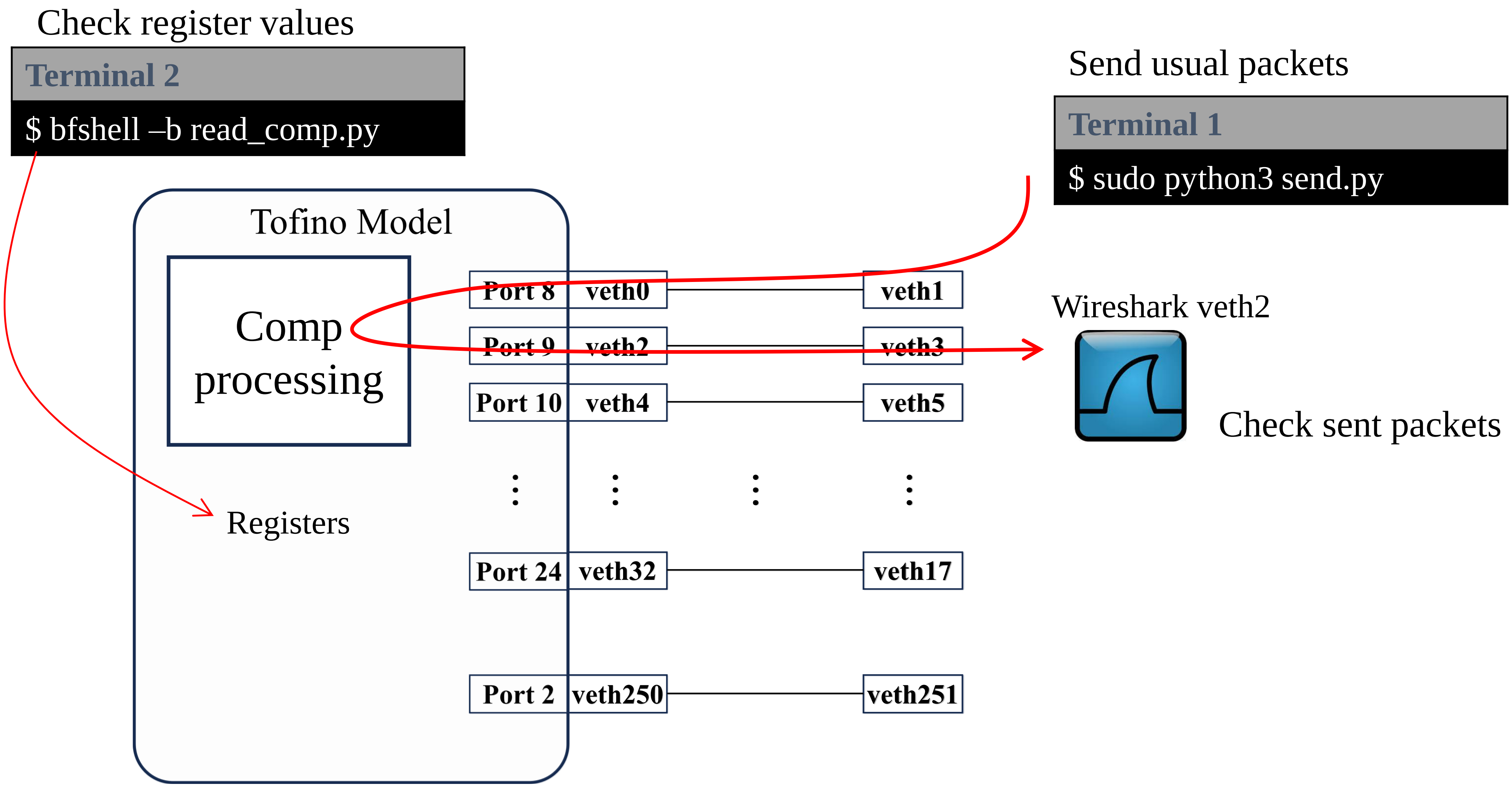
Terminal 1

```
$ sudo python3 send.py
```

Wireshark veth2

Check sent packets

Port 8	veth0	veth1
Port 9	veth2	veth3
Port 10	veth4	veth5
⋮	⋮	⋮
Port 24	veth32	veth17
Port 2	veth250	veth251



Check register values

Terminal 2

```
$ bfshell –b read_comp.py
```

Send usual packets

Terminal 1

```
$ sudo python3 send.py
```

Tofino Model

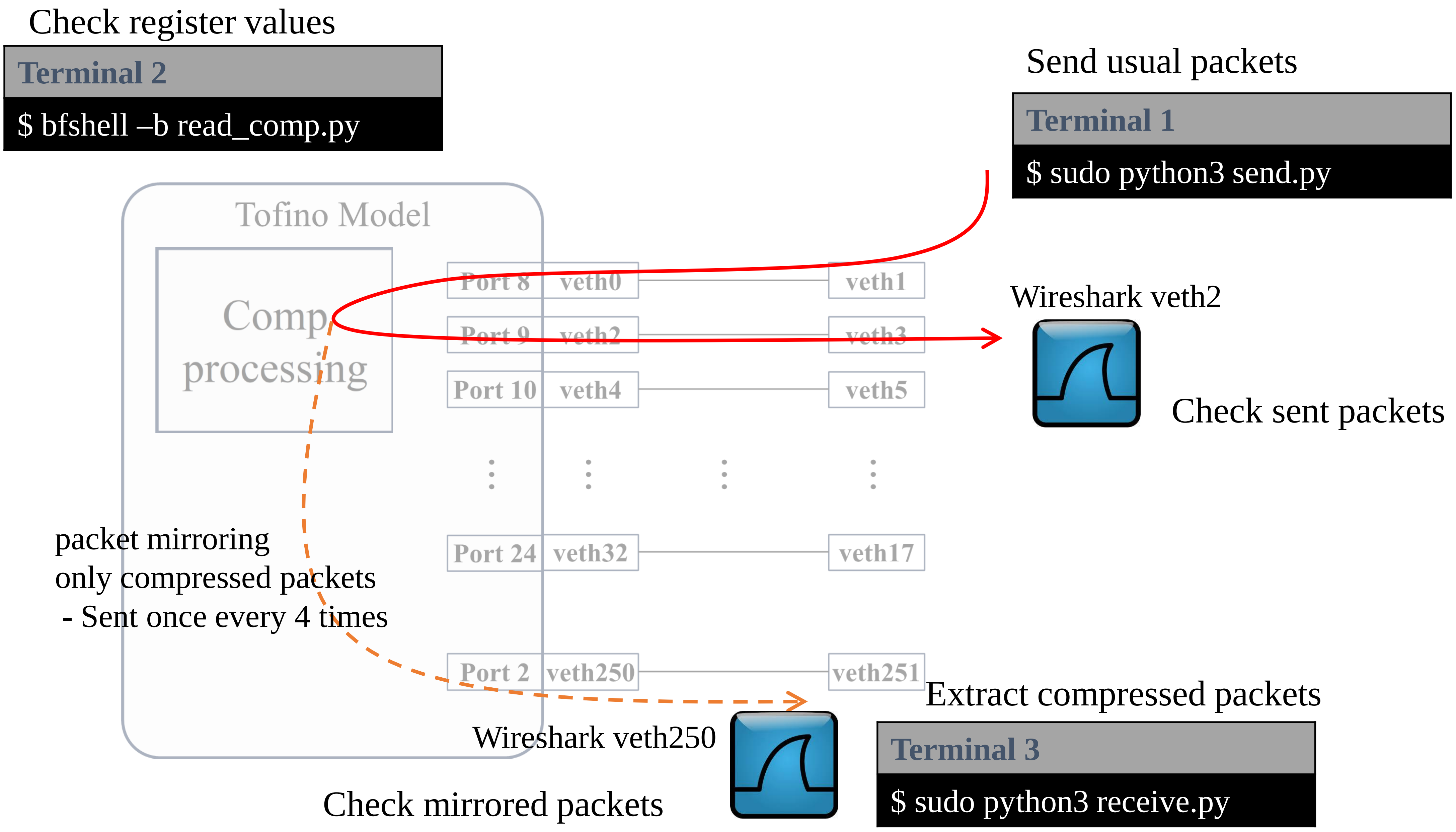
Comp processing

Registers

Port 8	veth0	veth1
Port 9	veth2	veth3
Port 10	veth4	veth5
⋮	⋮	⋮
Port 24	veth32	veth17
Port 2	veth250	veth251

Wireshark veth2

Check sent packets



Check register values

```
Terminal 2
$ bfshell –b read_comp.py
```

Send usual packets

```
Terminal 1
$ sudo python3 send.py
```

Wireshark veth2

Check sent packets

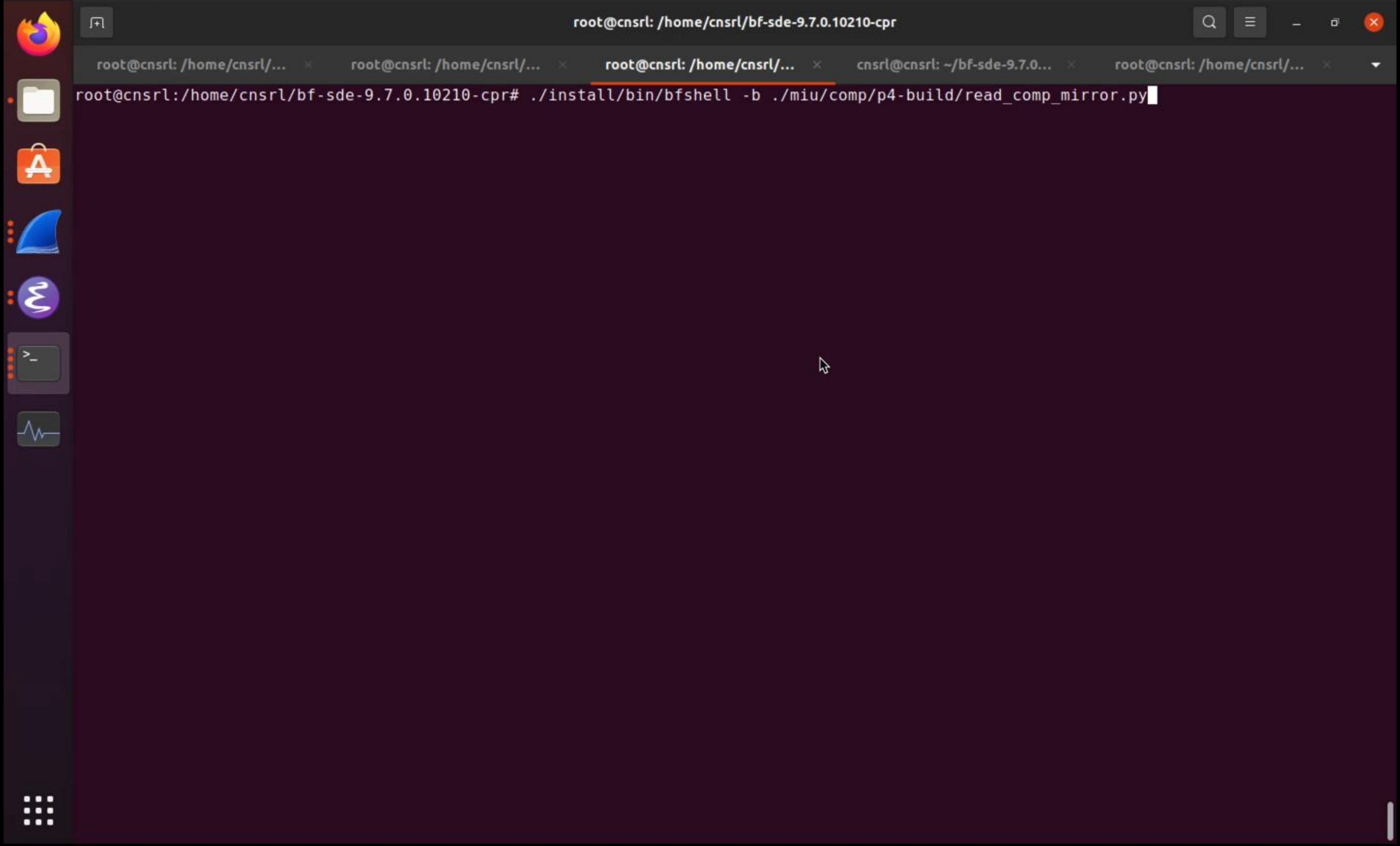
packet mirroring
only compressed packets
 - Sent once every 4 times

Wireshark veth250

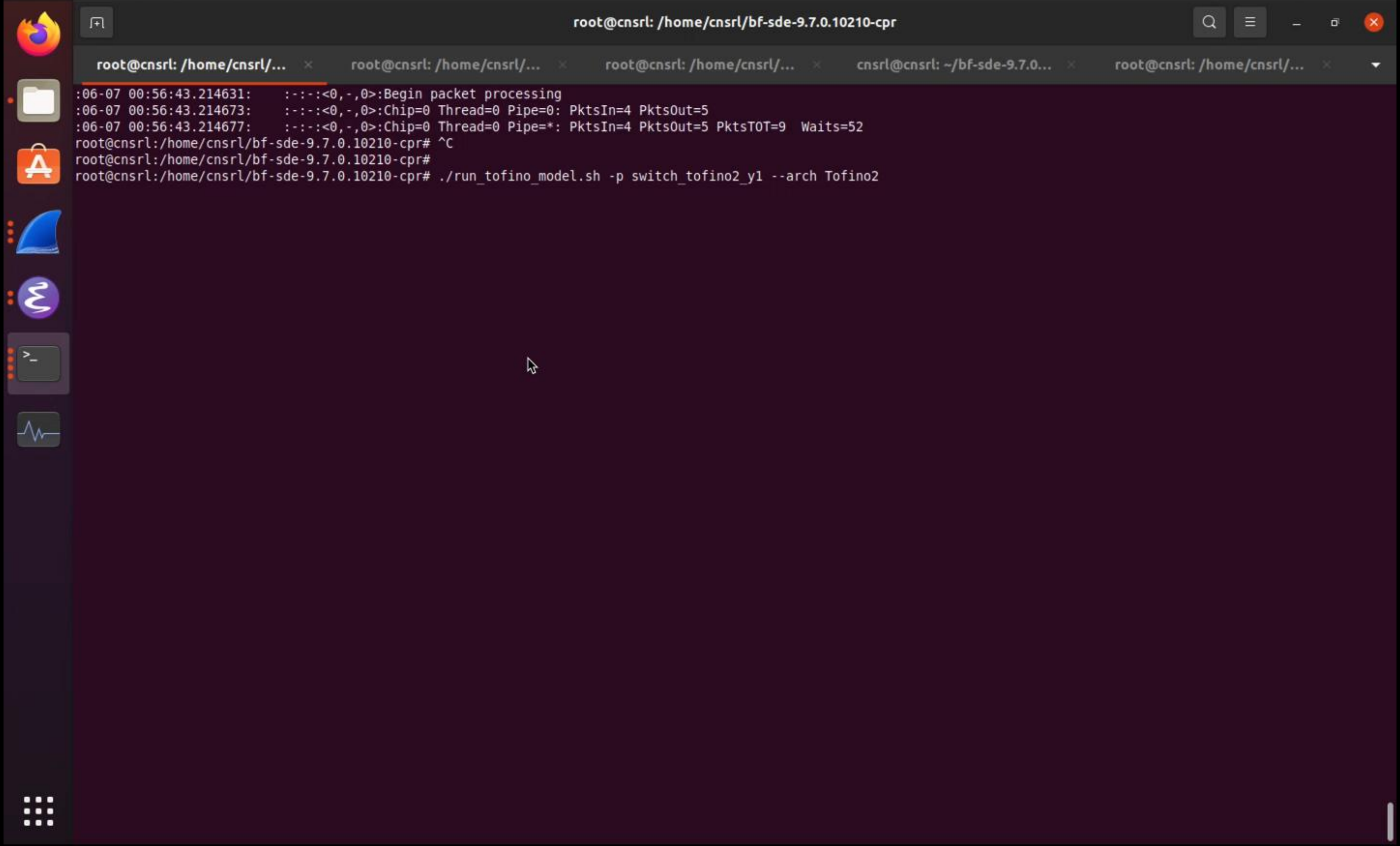
Check mirrored packets

Extract compressed packets

```
Terminal 3
$ sudo python3 receive.py
```

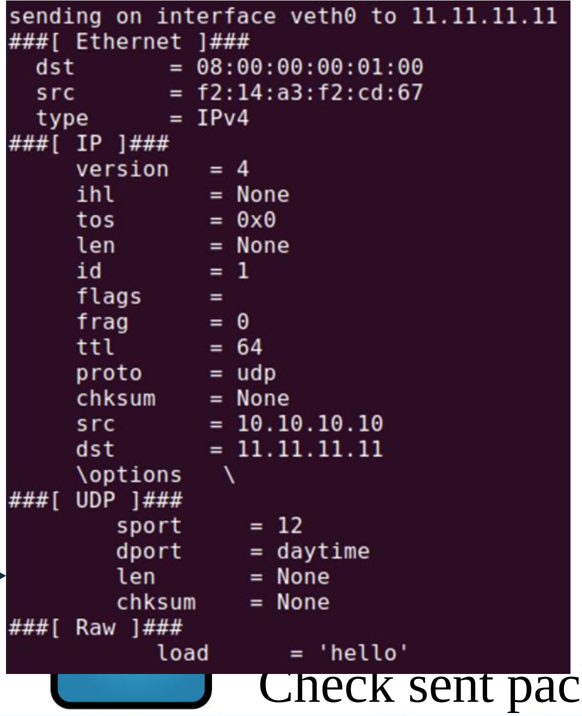

root@cnsrl: /home/cnsrl/bf-sde-9.7.0.10210-cpr
root@cnsrl: /home/cnsrl/...
root@cnsrl: /home/cnsrl/...
root@cnsrl: /home/cnsrl/...
cnsrl@cnsrl: ~/bf-sde-9.7.0...
root@cnsrl: /home/cnsrl/...

```
root@cnsrl:/home/cnsrl/bf-sde-9.7.0.10210-cpr# ./install/bin/bfshell -b ./miu/comp/p4-build/read_comp_mirror.py
```


root@cnsrl: /home/cnsrl/bf-sde-9.7.0.10210-cpr
root@cnsrl: /home/cnsrl/...
root@cnsrl: /home/cnsrl/...
root@cnsrl: /home/cnsrl/...
cnsrl@cnsrl: ~/bf-sde-9.7.0...
root@cnsrl: /home/cnsrl/...

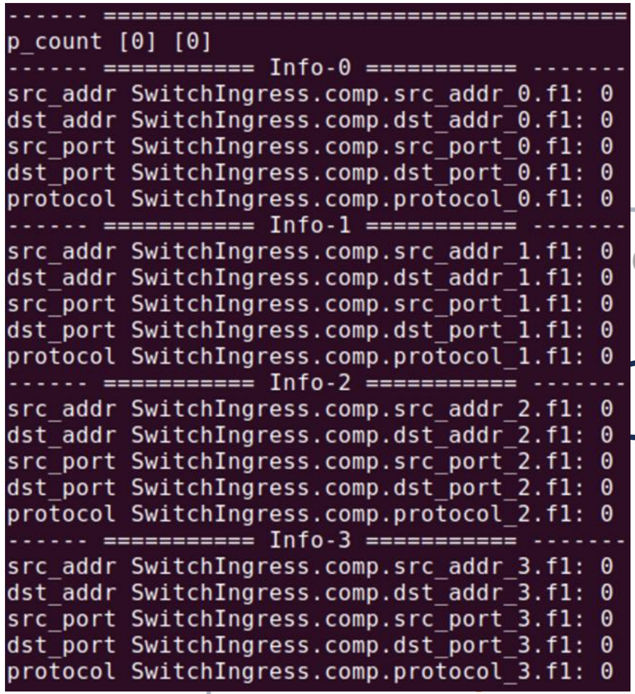
```
:06-07 00:56:43.214631:    :-:-:<0,-,0>:Begin packet processing
:06-07 00:56:43.214673:    :-:-:<0,-,0>:Chip=0 Thread=0 Pipe=0: PktsIn=4 PktsOut=5
:06-07 00:56:43.214677:    :-:-:<0,-,0>:Chip=0 Thread=0 Pipe=*: PktsIn=4 PktsOut=5 PktsTOT=9  Waits=52
root@cnsrl:/home/cnsrl/bf-sde-9.7.0.10210-cpr# ^C
root@cnsrl:/home/cnsrl/bf-sde-9.7.0.10210-cpr#
root@cnsrl:/home/cnsrl/bf-sde-9.7.0.10210-cpr# ./run_tofino_model.sh -p switch_tofino2_y1 --arch Tofino2
```

Send usual packets



```
sending on interface veth0 to 11.11.11.11
###[ Ethernet ]###
  dst       = 08:00:00:00:01:00
  src       = f2:14:a3:f2:cd:67
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = None
     tos       = 0x0
     len       = None
     id        = 1
     flags     =
     frag      = 0
     ttl       = 64
     proto     = udp
     chksum    = None
     src       = 10.10.10.10
     dst       = 11.11.11.11
     \options   \
###[ UDP ]###
        sport     = 12
        dport     = daytime
        len       = None
        chksum    = None
###[ Raw ]###
           load      = 'hello'
```

Check register values



```
------ =======================================
p_count [0] [0]
------ =========== Info-0 =========== -------
src_addr SwitchIngress.comp.src_addr_0.f1: 0
dst_addr SwitchIngress.comp.dst_addr_0.f1: 0
src_port SwitchIngress.comp.src_port_0.f1: 0
dst_port SwitchIngress.comp.dst_port_0.f1: 0
protocol SwitchIngress.comp.protocol_0.f1: 0
------ =========== Info-1 =========== -------
src_addr SwitchIngress.comp.src_addr_1.f1: 0
dst_addr SwitchIngress.comp.dst_addr_1.f1: 0
src_port SwitchIngress.comp.src_port_1.f1: 0
dst_port SwitchIngress.comp.dst_port_1.f1: 0
protocol SwitchIngress.comp.protocol_1.f1: 0
------ =========== Info-2 =========== -------
src_addr SwitchIngress.comp.src_addr_2.f1: 0
dst_addr SwitchIngress.comp.dst_addr_2.f1: 0
src_port SwitchIngress.comp.src_port_2.f1: 0
dst_port SwitchIngress.comp.dst_port_2.f1: 0
protocol SwitchIngress.comp.protocol_2.f1: 0
------ =========== Info-3 =========== -------
src_addr SwitchIngress.comp.src_addr_3.f1: 0
dst_addr SwitchIngress.comp.dst_addr_3.f1: 0
src_port SwitchIngress.comp.src_port_3.f1: 0
dst_port SwitchIngress.comp.dst_port_3.f1: 0
protocol SwitchIngress.comp.protocol_3.f1: 0
```

el

Port 8 veth0 veth1

Port 9 veth2 veth3

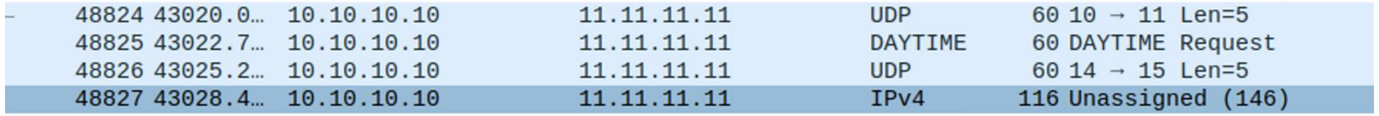
Port 10 veth4 veth5

⋮ ⋮

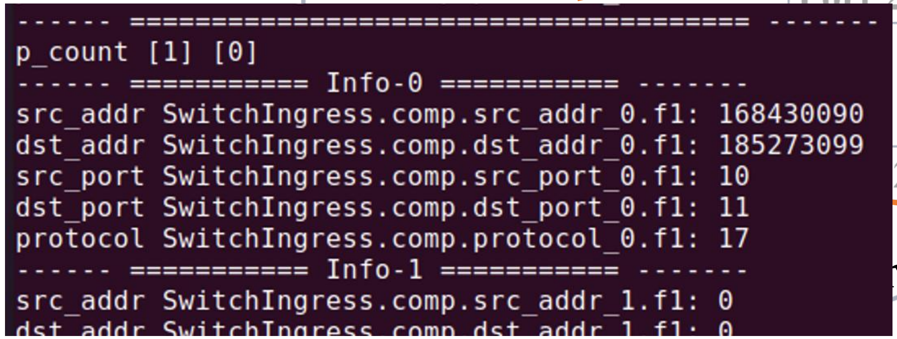
Port 24 veth32 veth17

2 veth250 veth251

Check sent packets



```
48824 43020.0…  10.10.10.10    11.11.11.11    UDP        60 10 → 11 Len=5
48825 43022.7…  10.10.10.10    11.11.11.11    DAYTIME    60 DAYTIME Request
48826 43025.2…  10.10.10.10    11.11.11.11    UDP        60 14 → 15 Len=5
48827 43028.4…  10.10.10.10    11.11.11.11    IPv4      116 Unassigned (146)
```



```
------ ===================================== -------
p_count [1] [0]
------ =========== Info-0 =========== -------
src_addr SwitchIngress.comp.src_addr_0.f1: 168430090
dst_addr SwitchIngress.comp.dst_addr_0.f1: 185273099
src_port SwitchIngress.comp.src_port_0.f1: 10
dst_port SwitchIngress.comp.dst_port_0.f1: 11
protocol SwitchIngress.comp.protocol_0.f1: 17
------ =========== Info-1 =========== -------
src_addr SwitchIngress.comp.src_addr_1.f1: 0
dst_addr SwitchIngress.comp.dst_addr_1.f1: 0
```

eshark veth250

Check mirrored packets

Extract compressed packets

Terminal 3

```
$ sudo python3 receive.py
```