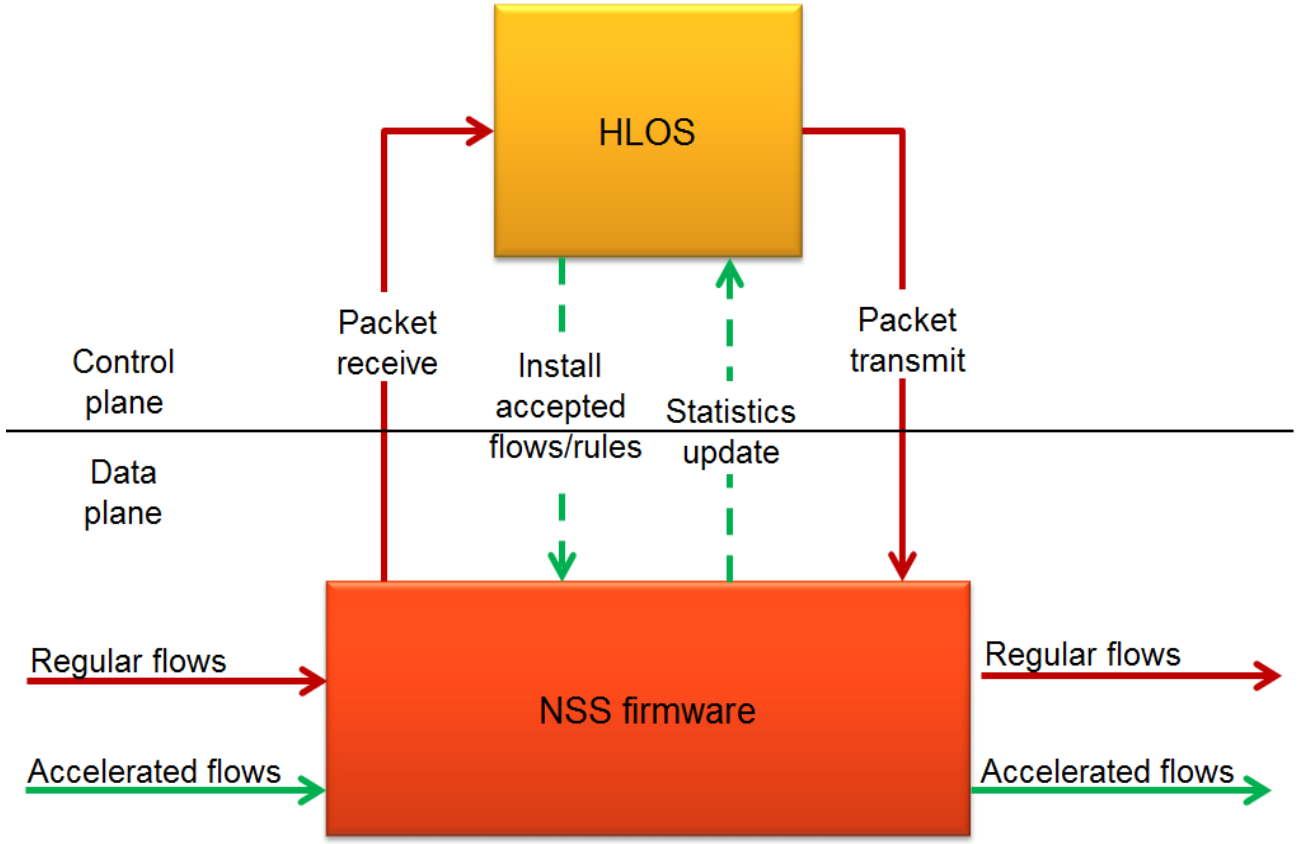
IPQ806x Hardware acceleration

Proceedings of netdev 0.1, Feb 14-17, 2015, Ottawa, On, Canada

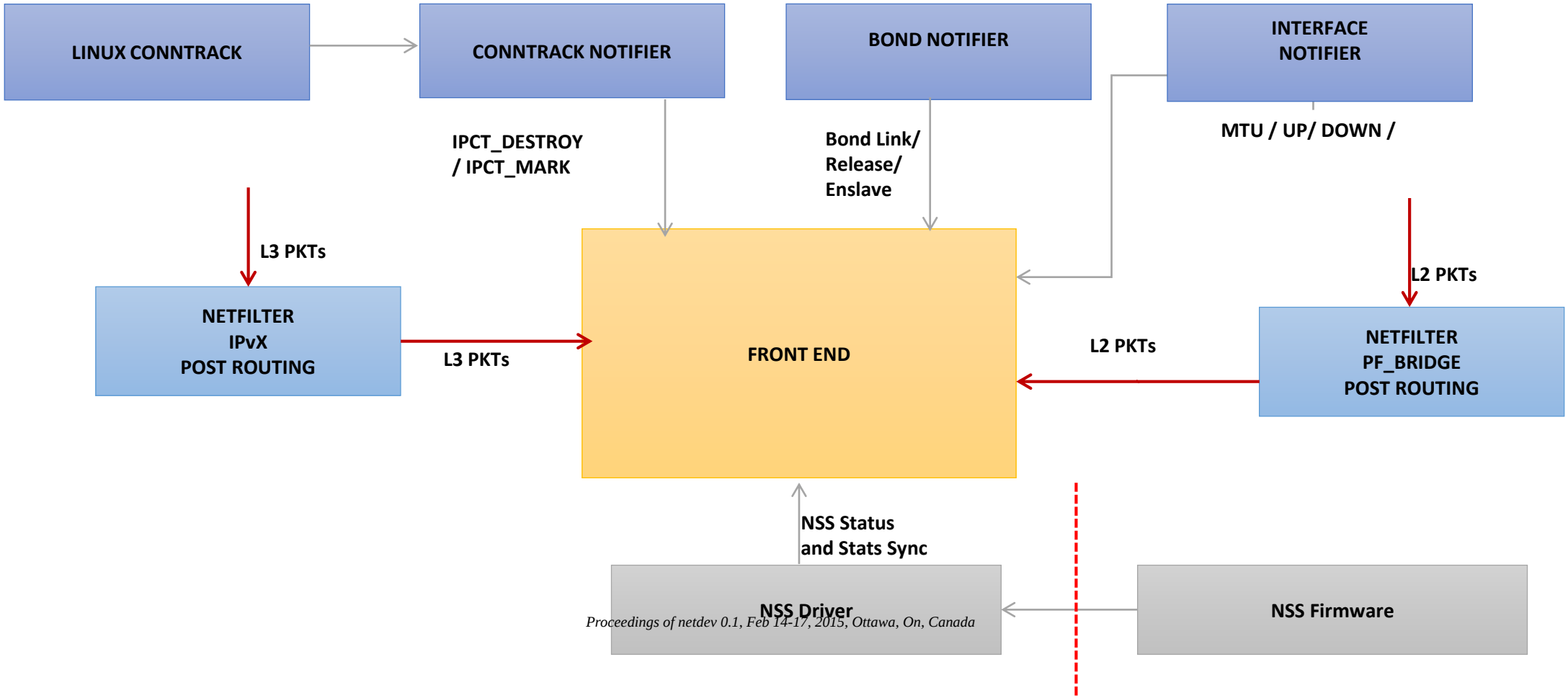
NSS acceleration model

- Features
  - Designed for Home Gateways (CPE)
  - **Flow detection based** “All-or-nothing” offload
  - Acceleration supports:
    - IPv4, IPv6, NAT, PPPoE, L2TP, VLAN, Qdisc
- Performance gain
  - Linux: 640k pps (bridged) – 220kpps (routed)
  - NSS: 7200k pps (bridged) – 7200kpps (routed)
  - 11x (bridged) – 32x (routed)
- Functional behavior
  - 0% cpu load seen in Linux
  - Keep Linux counters up to date
  - Does not require functional changes at an upper level (user space)

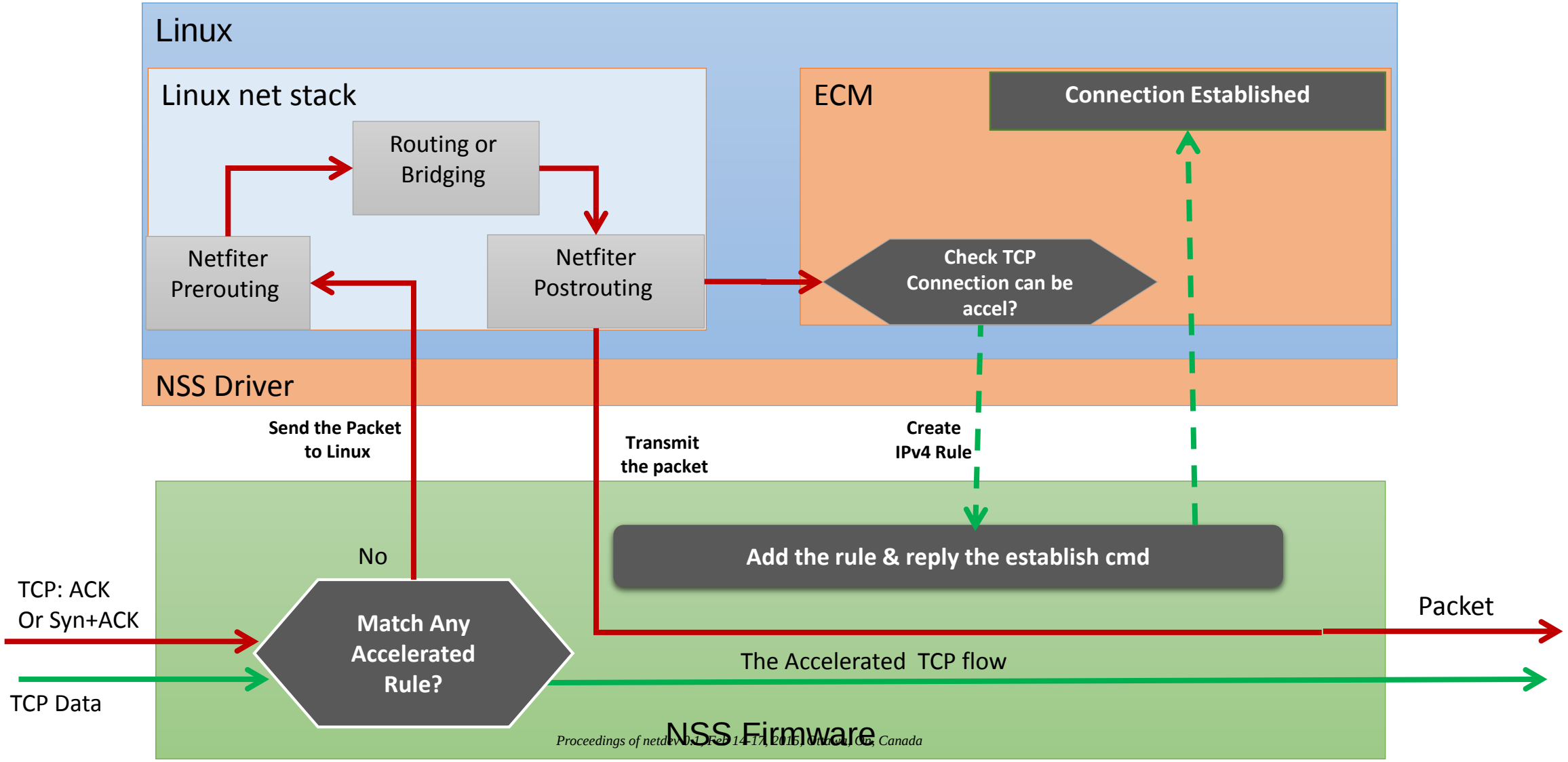


ECM Front End Inspect Packets and Events

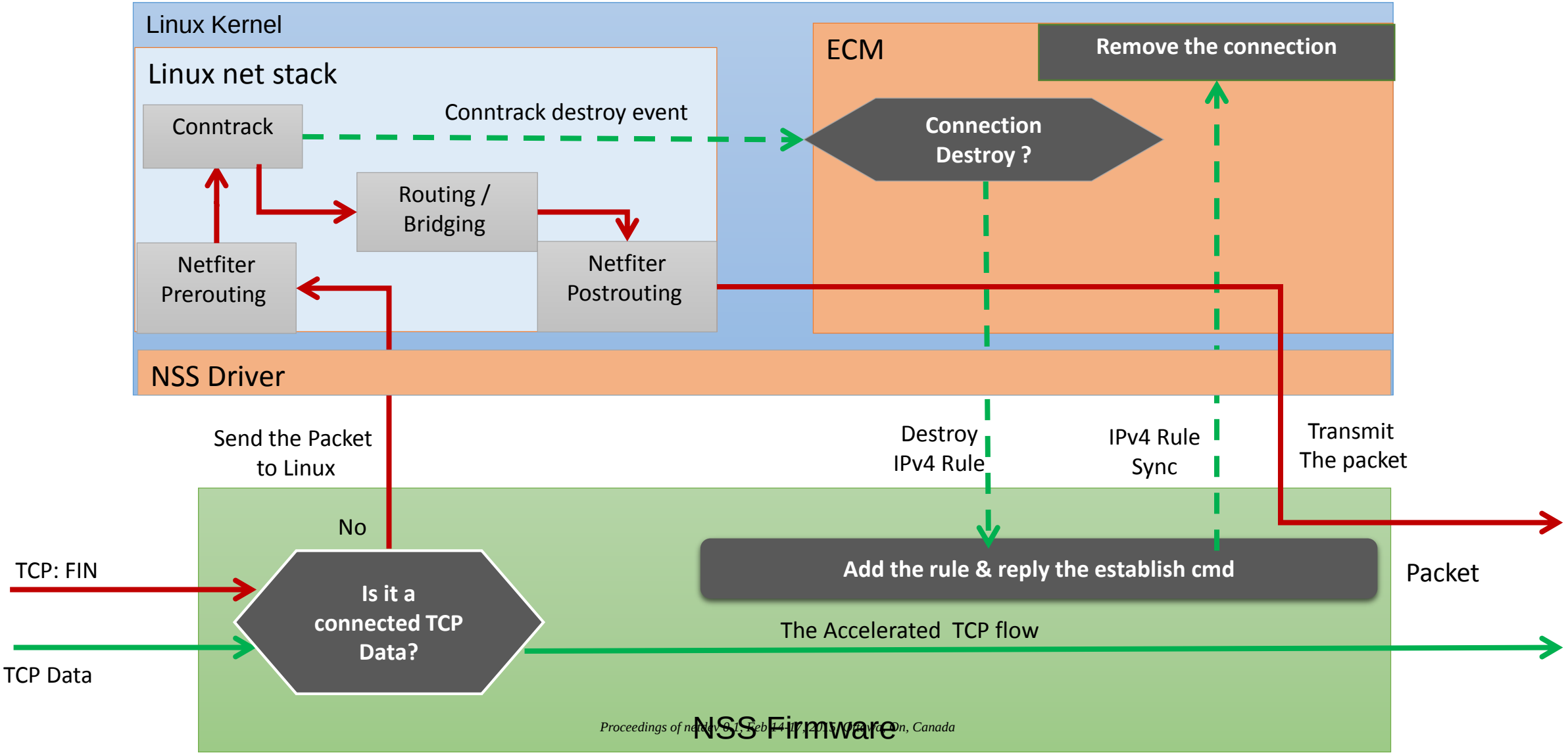
- Inspect all outgoing packets at POSTROUTING chain by registering post routing hooks.
- Inspect conntrack and device events to destroy and regenerate connection.
- Inspect NSS status and stats to update the connection state and statistics info in Linux and ECM DB.



Example of IPv4 TCP rule creation



Example of IPv4 TCP rule destroy



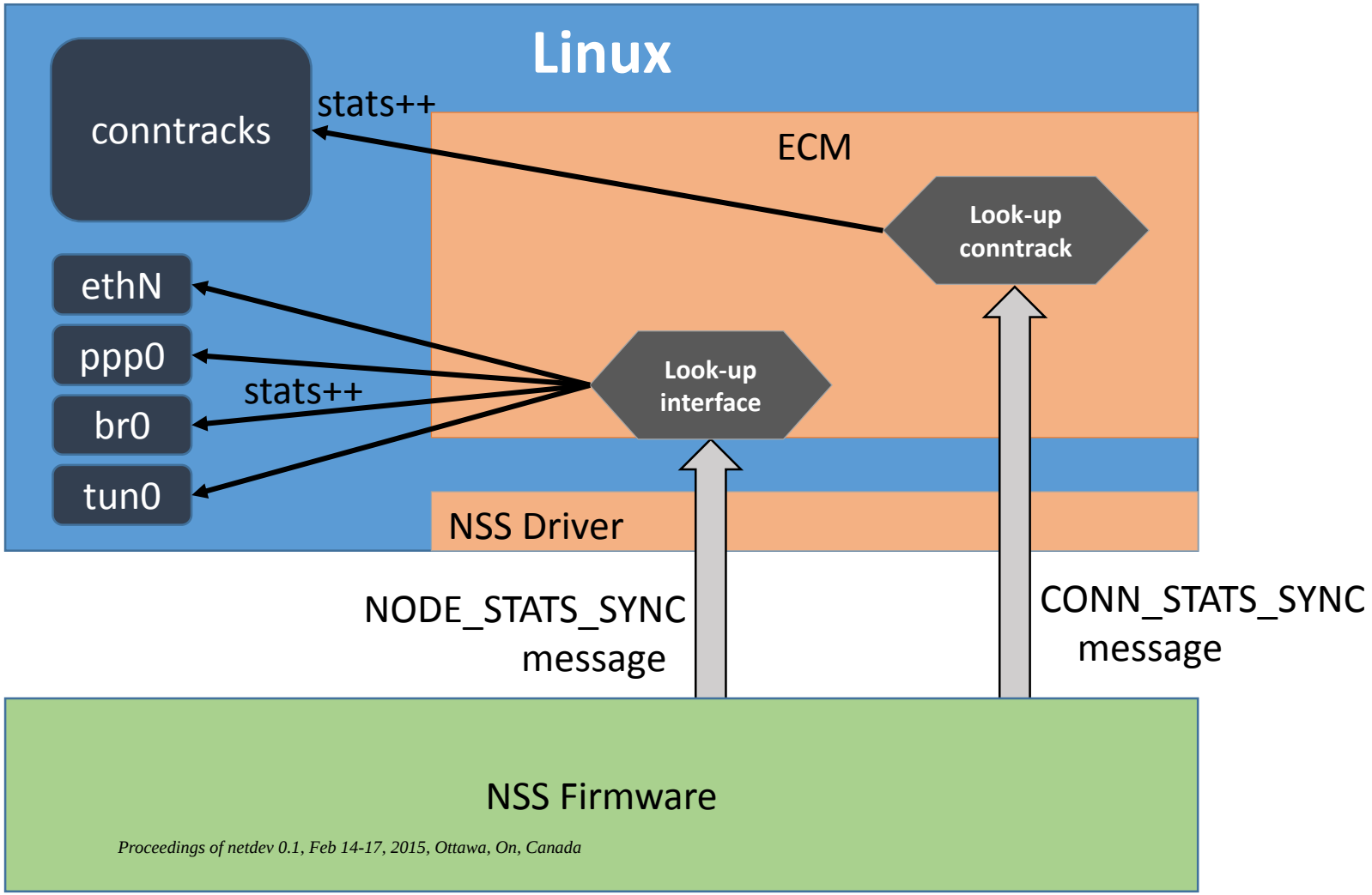
Example of IPv4 rule API

- Common messaging interfaces
  - Protocol type: IPv4, IPv6, PPP, LAG…
  - Rule type: CREATE, DESTROY, CONN_STAT_SYNC , NODE_STAT_SYNC…
  - Callback/args pointers: will be passed back in the FW ACK/NACK reply
- IPv4 rule create message structure example

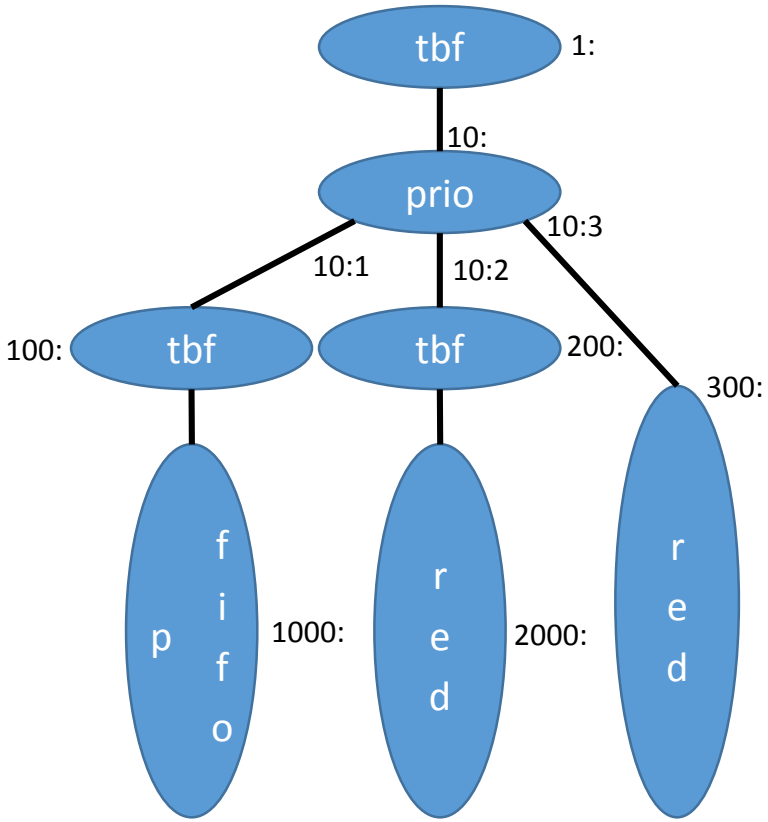
```
struct nss_ipv4_rule_create_msg {
        /* Request */
        uint16_t valid_flags;                           /* Indicate which of the parameters below is filled-in
                                                           Indirectly says which operation to be done on the flow */
        uint16_t rule_flags;                            /* Bit flags associated with the rule */
        struct nss_ipv4_5tuple tuple;                   /* src_ip, dst_ip, src_port, dst_port, proto */
        struct nss_ipv4_connection_rule conn_rule;      /* src_mac, dst_mac, src_iface, dst_iface, src_mtu, dst_mtu,
                                                           nat_src_ip, nat_dst_ip, nat_src_port, nat_dst_port */
        struct nss_ipv4_protocol_tcp_rule tcp_rule;     /* TCP related accleration parameters */
        struct nss_ipv4_pppoe_rule pppoe_rule;          /* flow_session_id, flow_remote_mac, ret_session_id, ret_remote_mac */
        struct nss_ipv4_qos_rule qos_rule;              /* flow_qos_tag, ret_qos_tag,  */
        struct nss_ipv4_dscp_rule dscp_rule;            /* flow_dscp, ret_dstp */
        struct nss_ipv4_vlan_rule vlan_primary_rule;    /* ingress_vlan_tag, egress_vlan_tag */
        struct nss_ipv4_vlan_rule vlan_secondary_rule;  /* ingress_vlan_tag, egress_vlan_tag – for QinQ */

        /* Response */
        uint32_t index;                                 /*Slot ID for cache stats to host OS */
};
```

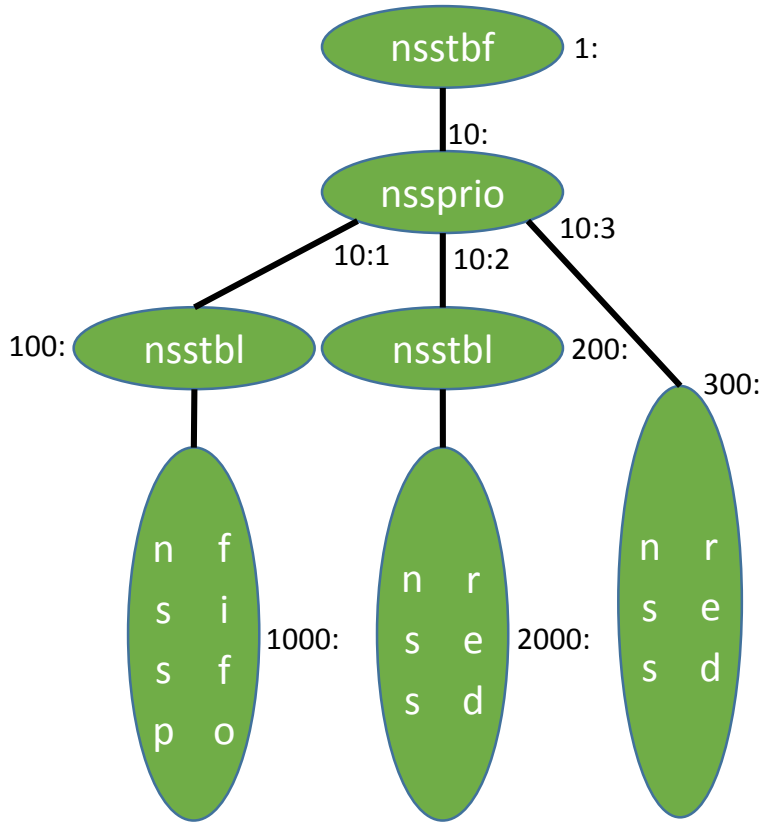
Interfaces & Connections statistics update

- Stats updates sent periodically from the Firmware
  - Per-interfaces stats (update net_devices)
  - Per-connections stats (update conntracks)
- Minor modifications to ppp/l2tp/ipsec… layers for iface look-up and stats update



Qdisc acceleration






```
# tc qdisc add dev eth0 root handle 1: tbf rate 1000Mbit burst 100k limit 100
# tc qdisc add dev eth0 parent 1: handle 10: prio bands 3
# tc qdisc add dev eth0 parent 10:1 handle 100: tbf rate 2Mbit burst 10k limit 100
# tc qdisc add dev eth0 parent 100: handle 1000: pfifo limit 100
# tc qdisc add dev eth0 parent 10:2 handle 200: tbf rate 40Mbit burst 30k limit 100
# tc qdisc add dev eth0 parent 200: handle 2000:  red limit 100k min 30k max 80k \
avpkt 1k burst 55 probability 0.20
# tc qdisc add dev eth0 parent 10:3 handle 300:  red limit 100k min 30k max 80k \
avpkt 1k burst 55 probability 0.30
```

```
# tc qdisc add dev eth0 root handle 1: nsstbl rate 1000Mbit burst 100k
# tc qdisc add dev eth0 parent 1: handle 10: nssprio bands 3
# tc qdisc add dev eth0 parent 10:1 handle 100: nsstbl rate 2Mbit burst 10k
# tc qdisc add dev eth0 parent 100: handle 1000: nsspfifo limit 100
# tc qdisc add dev eth0 parent 10:2 handle 200: nsstbl rate 40Mbit burst 30k
# tc qdisc add dev eth0 parent 200: handle 2000: nssred limit 100k min 30k max 80k \
avpkt 1k burst 55 probability 0.50
# tc qdisc add dev eth0 parent 10:3 handle 300: nssred limit 100k min 30k max 80k \
avpkt 1k burst 55 probability 0.50 set_default
```