



# Hardware GRO

**Michael Chan**



# Hardware LRO

- **Every vendor's LRO implementation is slightly different**
- **No guarantee to reconstruct the original packet stream**
  - Segments of different sizes may be merged
  - Segments with non-incrementing IP ID may be merged
  - Segments with different TCP timestamps may be merged
- **LRO has to be globally disabled when bridging/ip-forwarding is enabled**

# Hardware GRO

- **Hardware GRO first implemented in bnx2x about 4 years ago**
  - The concept originated from the bnx2x team
  - Today, it is supported by bnx2x, bnxt\_en, and qede
- **Basically, a stricter implementation of LRO**
  - All segments must be the same size (except possibly the last one)
  - Hardware returns MSS to the driver
  - Useful for hardware to return header offsets to setup SKB
  - IP ID must be incrementing unless DF is set
  - All TCP timestamps must be the same
- **Exact reverse of TSO**
  - We must be able to perform TSO on the Hardware GRO packet and recreate the same packet stream
- **Hardware GRO can co-exist with software GRO**

# Hardware GRO SKB

```
NAPI_GRO_CB(skb)->segs  
skb_shinfo(skb)->gso_size  
skb_shinfo(skb)->gso_type  
skb->network_header  
skb->transport_header  
th->check = pseudo checksum  
tcp_gro_complete(skb);
```

**Tunneled packets require additional SKB setup**

# Perf profile

## Hardware GRO:

0.15%	swapper	[bnxt_en]	[k] bnxt_rx_pages
0.06%	swapper	[bnxt_en]	[k] bnxt_poll
0.01%	swapper	[kernel.kallsyms]	[k] tcp_gro_receive
0.01%	swapper	[kernel.kallsyms]	[k] dev_gro_receive

----

## Software GRO:

1.31%	swapper	[kernel.kallsyms]	[k] tcp_gro_receive
1.10%	swapper	[bnxt_en]	[k] bnxt_rx_pkt
1.01%	swapper	[kernel.kallsyms]	[k] dev_gro_receive
0.85%	swapper	[kernel.kallsyms]	[k] inet_gro_receive
0.41%	swapper	[kernel.kallsyms]	[k] skb_gro_receive
0.35%	swapper	[bnxt_en]	[k] bnxt_alloc_rx_data
0.31%	swapper	[bnxt_en]	[k] bnxt_poll
0.15%	swapper	[kernel.kallsyms]	[k] napi_gro_receive

# NETIF\_F\_GRO

- **Controls both software GRO and hardware GRO**
- **bnx2x has a `disable_tpa` parameter to disable hardware GRO**
- **Add a new `NETIF_F_?` flag to differentiate the two?**
  - Will send patch to add `NETIF_F_GRO_HW`