Netdev 0x16

XDP Workshop

Alexander Lobakin
Maciej Fijalkowski
Larysa Zaremba

Agenda

1. AF_XDP for virtio
   (Xuan Zhuo - Alibaba)
2. Dynamic pointers
   (Joanne Koong - Meta)
3. XDP Hints
4. AF_XDP multi-buffer
5. XDP_REDIRECT
   to various types of devices
   (Zhan Xue - Intel)

AF_XDP for virtio

(Xuan Zhuo - Alibaba)

Dynamic pointers

External presentation
(Joanne Koong - Meta)

XDP Hints

1. Why generic/common hints?
2. Hints handling in driver
   - if-else ladders vs “all or nothing”
3. How to configure hints?
   - ethtool vs program attachment
4. Hints visibility
   - UAPI vs BTF-only
   - Hints headers for userspace applications
5. Potential of kfunc/helpers
   - easily access hints fields
   - avoid data_meta altogether?

intel.

Why do we need generic hints?

- XDP_REDIRECT to veth
- cpumap
- source-independent programs

Hints handling in driver

```
if (!hints_enabled)
        return;

hints->rx_flags = XDP_META_RX_VID;
hints->rx_vid = vlan_id;

if (checksum_feature_enabled(netdev) &&
    nic_csum_ok(rx_desc))
        hints->rx_flags |= XDP_META_RX_CSUM_OK;

if (rss_feature_enabled(netdev)) {
        hints->rx_flags |= XDP_META_RX_RSS;
        hints->rss = nic_get_rx_hash(rx_desc);
}
```

```
hints->rx_flags = 0;

/* Can replace netdev with program/attachment */
if (vlan_hint_enabled(netdev)) {
        hints->rx_flags = XDP_META_RX_VID;
        hints->rx_vid = vlan_id;
}

if (checksum_hint_enabled(netdev) &&
    nic_csum_ok(rx_desc))
        hints->rx_flags |= XDP_META_RX_CSUM_OK;

if (rss_hint_enabled(netdev)) {
        hints->rx_flags |= XDP_META_RX_RSS;
        hints->rss = nic_get_rx_hash(rx_desc);
}
```

How to configure hints?

- Fill in all enabled offloads, so if you do need a checksum, just disable it through ethtool
- Separate hints configuration, but per-device
- Config hints per-program / per attachment / … – especially useful, if we want to have program per-context.
  Check automatically, what fields the program uses?

Hints visibility

Expose through UAPI

- Harder to change
- Extension still possible through adding fields at the top
- Easy to use in AF_XDP

Expose through BTF

- In-kernel hints structures would be easier to change
- AF_XDP needs to decode BTF information, hence speed penalty
- Maybe we can work out a solution for AF_XDP power users?

intel.

Generate hints headers for userspace

Open to debate:

1. Generate a header file from BTF with:
   - Selected XDP Hints structs
   - Their dependencies
   - Function **`int hints_verify(void)`**
2. In AF_XDP program, call **`hints_verify()`**
   - Find current versions of used types
   - Compare them to compiled types
   - Fail, if types do not match
   - Ignore changes, if they lead only to hints growing leftwards
3. Recompile, if program fails on the newer kernel

kfunc & bpf-helpers

Directly access metadata in descriptor with driver functions

- Possible to assemble custom metadata
- Even better with multiprog and AF_XDP

Discussion:

- Generic or driver-specific helpers?
- How should generic helpers be resolved?
- Pass rx_desc to XDP prog or make driver dig through stack?
- Must we drop the hints-to-skb conversion?

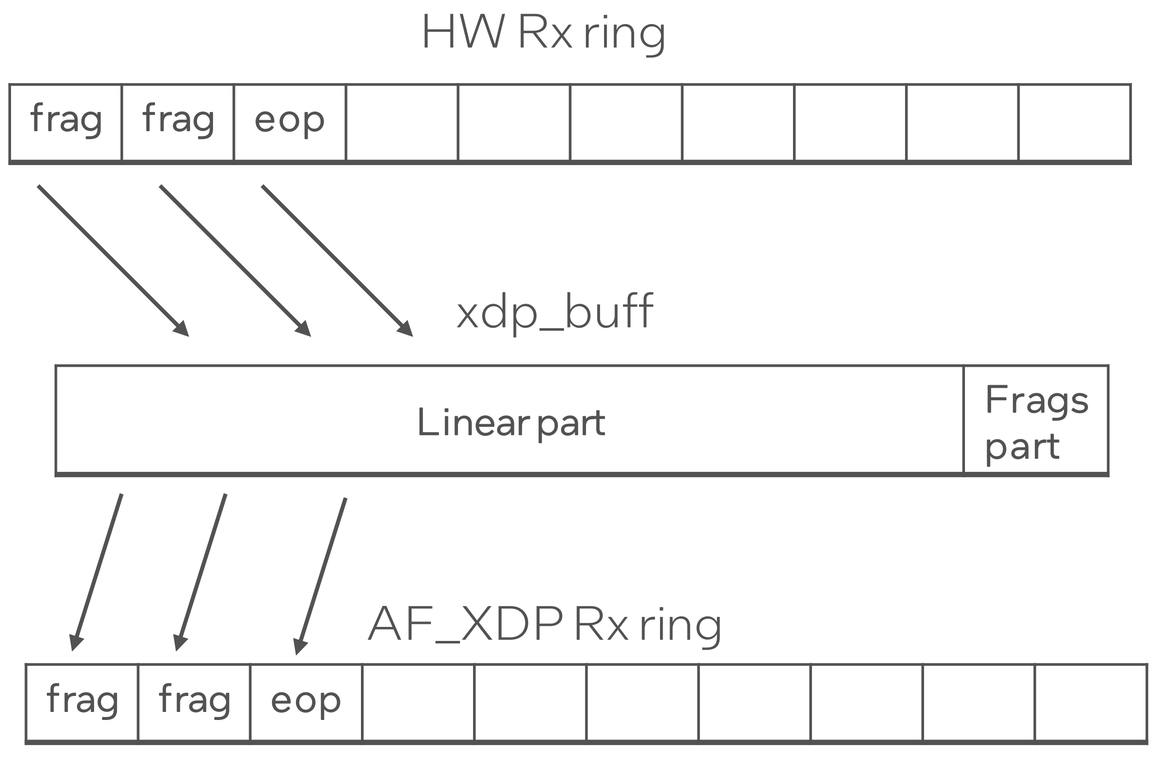
AF_XDP multi-buffer support

multi-buffer & jumbo frames

Magnus Karlsson
Martyna Szapar-Mudlaw
Tirthendu Sarkar
Maciej Fijalkowski

AF_XDP multi-buffer - Rx

- Standard XDP path that supports fragmented frames combines fragments to xdp_buff (linear + frags parts) before running XDP program
- For AF_XDP, we would like to push such operation up to the user space
  - Otherwise, performance will suffer
- How to address that?



```
struct xdp_desc {
                __u64 addr;
                __u32 len;
                __u32 options; // signal EOP here for AF_XDP rings
};
```

AF_XDP multi-buffer - Rx, contd.

- Option 1:
  - Special case XDP_REDIRECT in ZC drivers – build xdp_buff per single HW descriptor and assume XDP prog will give us XDP_REDIRECT as a verdict. If verdict is different and we are in the middle of a frame, rewind AF_XDP Rx ring state.
  - Propagate End-Of-Packet bit from HW descriptors via 'options' field from xdp_desc (or basically AF_XDP ring descriptors)
  - This unfortunately violates multi-buffer aware XDP progs that might use BPF multi-buffer helpers within
    - Forbid using these helpers for AF_XDP ?

AF_XDP multi-buffer - Rx, contd.

- Option 2:
  - Revive idea of removing XDP prog from AF_XDP data path (IOW AF_XDP direct receive) ?
    - https://lore.kernel.org/bpf/1570515415-45593-1-git-send-email-sridhar.samudrala@intel.com/
    - Quoted set received huge push back from community
    - Regardless of performance improvement, carrying XDP prog in here becomes a bit painful

Other ideas?

intel.

AF_XDP jumbo frames

- Unaligned mode, huge pages and chunk size set to 9k
  - This has been brought up many times on mailing lists
    - Matter of lifting chunk size restriction for unaligned mode?
- Aligned mode without forcing user to configure huge pages?
  - This would be desirable solution, but how to achieve it?
- Above is ZC, what about copy mode?
  - Matter of grabbing order 3 pages and DMA mapping it?
    - The higher order of page grabbed the less effective it might get
    - Probably better to stick with scattered way for copy mode
- Does jumbo frame support make multi-buffer support redundant for AF_XDP?
  - No. Way forward is to use single descriptor for ZC and multiple ones for copy mode when working with jumbo frames

XDP_REDIRECT
to various types of devices

(Zhan Xue - Intel)

intel®