

AF_XDP: semantics, queues, offloads, testing — what next?

BoF · Linux Plumbers Conference 2026

Goal and agenda

- **Goal:** agree on the top pain points, leave with owners? + next steps
- **Not:** design anything in the room
- **Gazillion slides in 45m**

Goal and agenda

1. 18 months in numbers
2. Semantics: copy vs ZC, multi-buffer, UMEM, CQ
3. Metadata, offloads, capability discovery
4. Queue ownership, RSS, netkit leasing, TX-only
5. Generic (skb) mode: keep, fix, or freeze?
6. UMEM as a page_pool memory provider (RFC)
7. KubeVirt: TX BPF hook, life without XSKMAP
8. Testing: xskxceiver -> drivers/net/hw, NIPA
9. Next steps + owners

18 months in numbers (2025-04 .. 2026-10)

GREW

ZC drivers gve DQ0 6.17 | idpf 6.18 | icssg-prueth 6.19 | fec 7.0
 netkit (via queue lease) 7.1
 posted, not merged: am65-cpsw, machb
TX meta launch time 6.15 (igc, stmmac)
 ZC honours per-desc XDP_TX_METADATA 7.3
RX meta bnxt hash 7.1 | gve ts 6.19 | i40e hash+vlan 7.4
infra queue mgmt API, memory providers, instance lock,
 netkit queue leasing

BROKE

72 core commits (net/xdp + uapi), 37 of them carry Fixes:
 2025-04..12: 6 fixes 2026-01..09: 31 fixes
Fixes: targets 13 -> 2023 multi-buffer commits
 10 -> TX metadata (6.8 / 6.15)
 7 -> fixes/refactors from this same window
xsk patches on list ~6/month --> ~32/month (from 2026-02)
Sashiko: 112 replies on 70 xsk patch subjects since 2026-04

Semantics: the bug storm, by theme

- A. CQ meaning entry posted before skb freed (shared UMEM);
 veth clobbered skb->cb; fix -> fix-of-fix -> +2
- B. mbuf TX invalid/oversized descs, half-built skbs, UAF,
 32-bit overflow (8 fixes 7.1-rc3, 4 in 7.2-rc6)
- C. ZC -> skb partial pkts on full RX ring; tailroom/truesize
- D. UMEM geometry shinfo in unreserved tailroom; desc overflow;
 MTU not re-validated after bind
- E. TX metadata 00B read (len < 16), double fetch from UMEM,
 launch_time @12 (32-bit) vs @16 => uAPI change 7.3

- “[is this code] getting sufficient networking eyes” — Jakub => xsk patches now go via net
- “we relied too much on contract with userspace WRT descriptor layout” — Maciej
- “generic xmit became utterly horrible” — Maciej
- “I hope that there’s a rewrite of this code coming up” — Jakub

Semantics: the copy vs ZC contract today

	ZC	copy (DRV+COPY / SKB)
CQ entry	"ownership return", NOT "sent" (7.2) -- both	
invalid/oversized	drained to EOP, all frames back via CQ -- both	
mbuf max frags	xdp_zc_max_segs: >1 only	MAX_SKB_FRAGS
	on i40e, ice, idpf	
TX meta flag	honoured per desc (7.3)	honoured
TX meta in batch	mlx5 MPWQE per batch (open)	n/a
TX csum	HW csum_start/offset	CHECKSUM_PARTIAL, no L3/L4 hdr set (open)?
launch time	HW (igc, stmmac)	skb_mstamp_ns, no qdisc
TX timestamp	HW	SW TAI in destructor
MTU after bind	not blocked (open)	
zc flag scope	per UMEM, should be per pool (open)	
skb_shared_info	in UMEM tailroom = user-writable memory	
32-bit	"I don't think we test 32 bit arches" -- Stan	

Semantics: questions for the room

- Normative “**AF_XDP contract**” doc? copy vs ZC, CQ ordering + ownership, mbuf limits, metadata rules, UMEM geometry
- **CQ = ownership return**: add oversized_descs stat? stmmac contradicts the doc; tell libxdp/apps
- **Copy-mode TX rewrite** (Stan’s 5 steps: peek all -> reserve CQ -> prealloc -> build -> xmit). Who? Fold into batch xmit?
- **shinfo in UMEM tailroom**: keep it? Headroom limits must be documented or exposed
- **32-bit**: support + test, or best effort?
- **Open bugs**: TX_BUSY consumer rewind (mbuf), MTU after bind, per-pool zc flag
- **AI reports** on pre-existing bugs: policy?

RX/TX metadata: who implements what (net-next)

RX kfuncs	hash	ts	vlan		TX (tmo)	TS	CSUM	LAUNCH
mlx5	Y	Y	Y		igc	Y	-	Y
mlx4	Y	Y	-		stmmac	Y	-*	Y
ice	Y	Y	Y	(SF: no)	mlx5	Y	Y	-
igc	Y	Y	-		idpf	-	Y	-
idpf	Y	Y	-	splitq	copy	SW	Y/SW	mstamp
i40e	Y	-	Y	7.4				
bnxt	Y	-	-	7.1	* stmmac inserts csum unconditionally (fix pending)			
gve (DQ)	-	Y	-					
stmmac	-	Y	-					
virtio_net	Y	-	-	hash rpt				
veth	Y	Y	Y	from skb	No TX meta: ice i40e ixgbe igb			
rx checksum	nobody			(not merged)	No ZC yet: fbnic			

Capability discovery: what's visible, what isn't

- **Can query:** xdp-features, xdp-zc-max-segs, xdp-rx-metadata-features, xsk-features, queue-get (xsk = empty nest, lease)
- **Can't:**
 - “op exists” != “works now” (HW ts off, idpf splitq, gve DQ -> -ENODATA)
 - rx csum, hash-type granularity, timestamp clock domain
 - data_meta support: ~19 drivers meta_valid=false, incl. ZC tsnep, fec, dpaa2, icssg
 - UMEM rules (headroom/tailroom/chunk, tx_metadata_len, SW csum) -> EINVAL at bind
 - copy-mode TX semantics, per-queue differences, “can this queue be leased?”
- **Inconsistent:** mlx5 zc_max_segs=1 despite rx-sg, ice SF ZC w/o xmo, stmmac csum w/o tx-checksum, mlx5 vs fbnic COMPLETE

Metadata + discovery: questions for the room

- **RX csum:** ship TODAY semantics (NONE/UNNECESSARY) now, invalidate/update kfuncs later?
- **Redirect hints:** pick one model (driver SET ops / stored struct / virtio_net_hdr-like / skb alloc in XDP) — or drop it
- **TX offload roadmap:** GSO/TSO/USO, VLAN insert in xsk_tx_metadata? Struct is extensible now. Users?
- **Extend netdev-genl:** per-queue + runtime metadata, data_meta, UMEM/metadata limits, copy-vs-ZC TX semantics, provider chunk rules
- Make xdp_rss_hash_type / xdp_checksum **UAPI**?
- Tell libxdp / xdp-tools about the **32-bit launch_time ABI change**
- Driver gaps: Intel TX meta, bnxt RX ts, fbnic ZC + xmo

Queue ownership today: KubeVirt on netkit (7.1)

host netns	pod netns (virt-launcher)
eth0 (phys)	nk0: netkit, single mode
ethtool -X ctx1 -> q15	lease rxq1 (= eth0 rxq15)
ntuple rule -> q15	<===
Cilium XDP prog on ALL of	QEMU af-xdp backend
eth0: redirect_map(xsks_map)-+-map-+>	bind(nk0, qid=1, ZC)
	pin+ -> eth0 ndo_bpf(
	mount XSK_POOL, q15)

RX: NIC q15 -> Cilium XDP -> XSKMAP -> xsk (validated via lease)
TX: netkit_xsk_wakeup -> eth0 ndo_xsk_wakeup -> ZC on eth0 txq15
no tc, no host BPF on egress

- **Why** (Daniel; Toke accepted): the VM is the barrier; Cilium shouldn't proxy xsk sockets; memory charged to the pod

Queue ownership: pain points

- **No single owner:** XDP prog (dev), xsk pool (rxq + txq), MP (rxq), lease (rxq), RSS/ntuple (userspace convention)
- **Leasing is RX-only;** ZC also takes the NIC TX queue (untracked)
- **Steering:** ethtool -L blocked, -X/-N unchecked — “the user can break the socket if they mess with the flow steering rules” (Mina)
- **Teardown order** + lock-order special cases (bond+XDP lockdep)

TX-only sockets: still bound to an RX queue id

- One qid for both directions; pool stored in `_rx[qid]` **and** `_tx[qid]`
- `XDP_SETUP_XSK_POOL` has no direction → drivers set up RX anyway
- Copy mode transmits on the **stack's** `txq[qid]` under the `xmit` lock
- `queue-create` is RX-only; only netkit implements `ndo_queue_create`
- `XSKMAP` rejects TX-only sockets (7.1), `libxdp` still attaches a redirect prog

2018 Bjorn: should TX-only be constrained by `#rx` queues?
Jakub: "...you'd have to allocate a separate queue"

2019 mlx5 "XSK queues beyond channels": `qid=-1` / "anonymous queue"
ideas; `uapi` flag NAKed; channel scheme merged 5.3, gone 6.1

2021 Magnus: helper so drivers know Rx/Tx -- never added

2025 Jason: "a standalone and clean tx queue ... only for xsk"

2026 Daniel: TX leasing PoC -- "enable `AF_XDP` TX-only support"
Jakub: "rx lease implies tx queue does not seem great"
=> asked for since 2018, never posted as a patch

TX-only: proposal for the room

- **TX type for queue-create / lease** (Daniel's PoC); NICs implement `ndo_queue_create` for TX => one op serves netkit TX leasing **and** dedicated xsk TX queues
- **bind() to a new TX queue** (`qid = -1 / flag`); report the real queue via `getsockopt` or `queue-get`
- **Direction in XDP_SETUP_XSK_POOL** -> no RX setup for TX-only
- **libxdp**: no redirect prog, no fill ring for TX-only sockets
- **Copy-mode TX-only?**

Generic (skb) mode: status

RX GSO/GRO skb	attach disables LRO/HW-GRO + SW GRO on that dev, but GSO skbs still arrive (bond/veth/tun) -> linearized (GFP_ATOMIC, <=64K); AF_XDP drops them unless XDP_USE_SG
RX csum	ip_summed not exposed; COMPLETE may go stale
RX hash/vlan/ts metadata kfuncs	lost; prog runs before skb_vlan_untag impossible: dev-bound progs rejected in SKB mode; veth already does xmo from the skb = precedent
RX mbuf	OK (frags + XDP_USE_SG)
TX csum	CHECKSUM_PARTIAL works, but skb->protocol and network header never set -> wrong/SW csum on IP_CSUM-only NICs, i40e/ice/ixgbe (open)
TX launch time	skb_mstamp_ns (6.15), qdisc bypassed
TX timestamp	SW TAI in destructor
TX GSO/TSO	none
TX mbuf	works (fixes in 7.2-rc6)

Generic mode: what the list said

- **Jason (copy-mode perf):** TX budget knob merged (6.17); batch xmit RFC v4: ixgbe 1.80 -> 3.39 Mpps (AF_PACKET 1.38) — stalled
 - Jakub: “I’m questioning optimizing the copy mode AF_XDP in the first place”
- **TX header bug:** two competing fixes (Petr Oros, Jason) — unresolved
- **Generic mem-type leak:** v1..v5, Jiayuan’s fix pending
- **Non-Ethernet generic XDP:** rejected — “Just attach the BPF in TC”
- **Jakub 2026-09-28:** drop GSO skbs in generic XDP + veth — “I don’t think generic XDP is worth the effort”
 - next: HW-GRO and XDP incompatible; Sashiko: guest TSO/USO via tun dropped even on XDP_PASS
- Toke: “so the feature set doesn’t diverge” Stan: “mostly test-fallback”

Generic mode: next steps — questions for the room

- **What is generic mode for?** test fallback | runs everywhere | AF_PACKET replacement — everything below depends on it
- **TX headers:** set `skb->protocol` + network header in `xsk_build_skb` or `__dev_direct_xmit`? Pick one, land it
- **RX hints:** skb-backed default xmo (veth precedent) -> allow dev-bound progs in SKB mode?
- **GSO RX:** drop (Jakub) vs frags with `XDP_USE_SG`? Fix the tun regression first
- **Batch xmit:** yes/no against Jakub's AF_PACKET bar
- Land the leak fix; document the generic contract; provider RFC: copy into provider buffers?

RFC: UMEM as a page_pool memory provider

- **Today:** every ZC driver keeps a 2nd RX path (xsk_buff_alloc, own refill, own XDP_PASS copy)
- **RFC:** UMEM chunks → net_iov (NET_IOV_XSK); per-queue provider via the queue mgmt API, like devmem / zcrx ⇒ one page_pool RX path

UMEM → xsk mp → page_pool(rxq N) → driver RX → XDP → XSKMAP → xsk
(same net_iov, no copy)

- **Status:** fbnic (+640/-122); bnxt working, not posted; aligned 4K chunks; xdp_buff 56 → 64 B; generic/cpumap can't deliver; XDP_PASS copies
- **Asks:** refill_done shape? split
PP_FLAG_ALLOW_UNREADABLE_NETMEM? +8 B xdp_buff? chunk constraints before bind?

RFC: feedback so far + interactions

- **Mina:** keep PP_FLAG_ALLOW_UNREADABLE_NETMEM semantics; add a new flag
- **Mina:** audit every “net_iov == unreadable” assumption:
xdp_build_skb_from_buff leak, skb_frag_address → netmem_address, skb_dump, bnxt/bnge need_head_pool, gve
- **Sashiko** replied on patches 03, 11, 13
- **Ask:** page_pool MP as the only ZC RX model long term? All new drivers use page pool.

KubeVirt asks (Daniel)

- **1. BPF on the AF_XDP socket's TX path** — policy / anti-spoof for VM traffic. In xsk core only, no driver changes?
- **2. Drop XSKMAP and the per-queue XDP prog** — the VM's leased queue should just deliver to its socket
- **Today:** ZC never builds an skb; copy mode uses `__dev_direct_xmit` (no qdisc, no tc); single-mode netkit has no copy mode
- **=> VM egress is invisible to host BPF**

TX hook in xsk core: where it could go [inf]

```
driver NAPI / wakeup
|-- xsk_tx_peek_desc()                mlx5 gve igc ixgbe stmmac nfp
|                                     icssg
|-- xsk_tx_peek_release_desc_batch() ice i40e igb virtio_net fec
|                                     dpaa2 otx2 tsnep
|                                     xskq_cons_read_desc_batch()
>>> HOOK: prog per desc (pool->netdev = netkit dev)
xsk_tx_commit_batch()
```

```
copy: __xsk_generic_xmit() -> xsk_build_skb()
      >>> HOOK on the built skb (data already copied)
      -> __dev_direct_xmit()
```

- **Prog:** BPF_PROG_TYPE_XDP + new attach type (like devmap progs); PASS/DROP, later punt
- **Attach:** bpf_link on the netkit dev (bpf_mprog), static key · **mbuf:** first-frag verdict covers the packet

TX hook: the hard parts

- **Drop vs CQ:** CQ is positional — dropped descs must still complete in order. Reuse reclaim (7.2), but sustained drops stall → pass/drop partition or reclaim-marker FIFO
- **TOCTOU:** ZC DMAs from UMEM that userspace can rewrite after PASS; copy mode is stable (except IFF_TX_SKB_NO_LINEAR)
- **Mitigations:**
 - copy-only (dead on netkit) · bounce slots in `xp_raw_get_*` · header-only bounce (`zc_max_segs ≥ 2`)
 - HW anti-spoof instead · narrow the threat model (does QEMU own UMEM?)
- **Ask:** threat model? is CQ order uAPI? bounce OK? attach point + verdicts? does isolation change Alexei's “filter only” view?

Life without XSKMAP: options

- **History:** XDP_ATTACH 2018, per-queue progs 2019 (“Queue is a low level primitive”), bpf_redirect_xsk + XDP_REDIRECT_XSK 2021
- **(0) Map-less redirect:** socket from rxq->pool — Daniel’s goal; no driver changes; host prog still needed
- **(i) Deliver-all:** provider queue without prog -> its socket; no ARP/DHCP punt
- **(ii) Per-rxq XDP prog** in netdev_rx_queue/qcfg; attach per (ifindex, queue); many drivers have per-ring prog pointers
- **Provider RFC:** queue ->UMEM binding becomes queue state; (i)/(ii) hang off it; needs lease interop

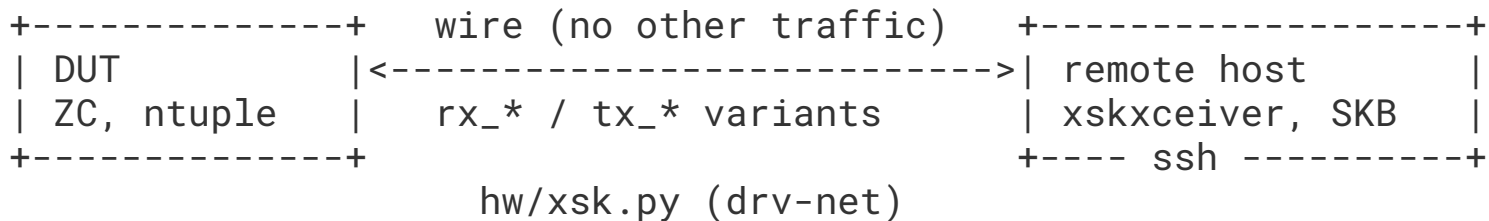
Testing today

xskxceiver / test_xsk.sh	veth; HW mode = single-NIC loopback, manual
test_progs -t xsk	BPF CI on veth (since 2025-10)
drv-net on NIPA HW runners (gve, fbnic qemu model, ...)	
xdp.py	native XDP, not AF_XDP
queues.py	check_xsk (bind visible in queue-get)
hw/xsk_reconfig.py	virtio-net deadlock repro
hw/xdp_metadata.py	rss_hash
xdp_hw_metadata	manual

- ⇒ **no AF_XDP data-path test runs on real HW in CI**
- veth missed: BIDIRECTIONAL broken on HW, ice NULL-deref, ZC/non-ZC quirks
- Jakub: stmmac/mlx5 metadata fixes wait for AF_XDP in drv-net

Maciej's series: xskxceiver → drivers/net/hw

- xskxceiver → selftests/net/lib/xsk; endpoints = processes + TCP control channel; one case per run



- Removes test_progs -t xsk and single-NIC loopback; tested on back-to-back ice
- Not on HW yet: BIDIRECTIONAL, SHARED_UMEM, STAT_*, XDP_DROP_HALF, ADJUST_TAIL, TEARDOWN, ...
- Review: TX loops forever if peer dies (CI hang); multi-frag IP/UDP lengths; net/bpf.mk skeleton rule; capability exchange?

Testing: next steps

- **Land hw/xsk.py.** NIPA must run test_xsk.sh **before** test_progs -t xsk is removed, or BPF CI loses coverage
- Per-case timeouts + peer-death handling — CI must not hang
- **HW runner owners:** gve (Google), fbnc (Meta), ice (Intel) — mlx5? bnxt? stmmac?
- **Per-driver coverage matrix:** ZC/copy, mbuf, TX meta, RX hints, need_wakeup, shared UMEM, teardown, XDP_PASS from ZC, ethtool -L/-G/MTU while bound, IPv6, stats, 32-bit
- Consistent oversized-desc handling + oversized_descs stat
- Long term: is xskxceiver the right base, or something new?

Next steps — owners (candidates, confirm in room)

topic	next step	who?
-----	-----	-----
semantics	contract doc; oversized_descs; 32-bit decision; open bug list	Maciej, Stan, Magnus
copy-mode TX metadata	rewrite (Stan's 5 steps); batch y/n csum TODAY doc + repost; redirect model	Jason, Maciej Lorenzo, Jesper, Stan
discovery	netdev-genl: UMEM limits, runtime/per-queue meta, data_meta, netkit mirroring	?
queues	TX leasing; map-less redirect; lease x provider interop; steering validation	Daniel, Bjorn
TX-only	TX queue-create/lease; bind new queue; direction in XSK_POOL setup; libxdp	Daniel, Bjorn, Magnus
generic	TX headers; GSO policy; skb xmo; leak	Jakub, Jason, Jiayuan
provider RFC KubeVirt	new PP flag; net_iov audit; bnxt; v2 threat model -> TX hook RFC; per-queue delivery on top of provider	Bjorn, Mina Daniel, Bjorn
testing	land hw/xsk.py; NIPA runners; matrix	Maciej, Jakub

Backup

Metadata: where the threads ended up

- **RX checksum kfunc** (Lorenzo, RFC 2025-09 .. v5 2026-07) — stalled
 - Jakub: COMPLETE goes stale once the prog writes; wants invalidate/update plan + HW tests
 - Stan: TODAY doc vs TOMORROW kfuncs; mlx5 hides COMPLETE under XDP, fbnet/gve don't
- **RX hints for redirected frames** (Jesper/Lorenzo) — stalled since 2025-08
 - Jakub: “another adhoc scheme”, xdp_frame as “para-skb”
- **Traits** abandoned → skb_meta dynptr (6.18) · **rx queue index kfunc** rejected twice
- **TX metadata**: race hardening 7.2, ABI fix 7.3; open: mlx5 MPWQE, stmmac
- **Nobody proposed**: AF_XDP TX GSO/TSO/USO, VLAN insert, RX hints without an XDP prog

Egress hook: prior art

- 2019 tun TX XDP (Bhole/Ahern) RFC dropped
- 2020 IFLA_XDP_EGRESS (Ahern) v1..v5 -> devmap progs (5.8)
 - Jakub: no use case, SG/GSO
 - Daniel: "AF_XDP's xmit would be more suited for something like that"
- 2022 PIFO / XDP queueing (Toke) stalled
- 2023 devtx (sdf) v1..v3 -- Jakub NACK -> xsk_tx_metadata (6.8)
 - Alexei: for AF_XDP "the bpf prog in the kernel is a filter only"
 - Magnus: AF_XDP was designed expecting an egress hook
- 2026 netkit single mode (Daniel, 7.1): future "XDP-layer BPF to filter egress / punt DHCP"
- 2024-26: no TX-hook RFC found on list