

netconf update

CI / NIPA topics

- Environment specific failures
 - NIPA CI occasionally fails for obscure reasons
 - Fedora vs other distros or even CPU / server related
 - Significant effort to identify the source of the problem
 - Long term we should support one-off runs in NIPA setup by known developers
- Patch churn
 - Settings (timeouts) are part of the kernel repo, they may need frequent updates
 - Wait and see...

CI / NIPA topics

- WIP by Matt to create container / system images with all the tools
 - Ease of recreating / running all selftests
 - Matching the version of tooling used in the NIPA CI
 - Using docker based on Fedora
 - Support both build / check and selftests run in vng
 - Some tools built from source: iproute2, ethtool, packetdrill
- Asks:
 - Provide chroot-able tar images
 - Make a single image that supports all ksft targets

CI / NIPA topics

- Driver tests
 - Problems in production like envs:
 - Expect root
 - Expect to run from DUT
 - Difficulties running existing tests
- For Python tests changing the execution model should be relatively painless

XDP and AF_XDP topics

- Unreadable memory vs XDP
 - Currently XDP enablement is per-interface
 - XDP is incompatible with zero-copy Rx and rejected by the kernel
 - What is the direction / expectation for XDP?
 - features should coexist, nobody had the need to write the code, yet
 - multi-buf XDP only, make the frag-access helpers fail on unreadable frags
 - per-queue XDP can also be supported via netdev NL

XDP and AF_XDP topics

- OOO completion support
 - IPU (idpf, gve) can complete Tx out of order
 - Should we expect drivers to linearize or extend the API?
 - Need to maintain support for both regardless of driver type
 - The immediate use of OOO Tx somewhat unclear
 - No fundamental objections raised

Zero-copy Rx, AF_XDP, latency

- Deployment and pain points
 - Lack of easy-to-consume high level API
 - Lack of best practices around configuration of clients and servers
 - No breakthroughs in the discussion, application-by-application
- Latency focus / busy polling
 - Historically we benchmarked throughput and RPS
 - Is there buy-in for potential tradeoffs in favor of latency / jitter
 - code complexity
 - sacrifice bandwidth

Timestamping and TCP Swift

- GVE gettimex64
 - Get host timestamps sandwiching PTP reads from host
 - Patches were merged but reverted by tglx
 - Need a way for VM to access precise network time (not system time) maintained by the hypervisor
 - PTP vclock, CLOCK_AUX, other “VM PTP” solutions do not match
 - Recommendation is to follow CLOCK_AUX / ptp_vmclock
- TCP Swift
 - need fast network <> system timestamp conversions
- Pacing hw offload
 - FQ parts mostly upstream
 - need optimizations to avoid timestamp conversion for ASAP
 - BQL interference

Fun with TCP upper stacking

- The stack potentially allows several upper layers on top of TCP: MPTCP, TLS, sockmap, ulp, etc.
- Most of them were not designed to co-exists with each-other:
 - Functional implications: AI and security researchers are very happy
 - Performance implications: overhead adds-up with compiler's knobs
- No magic bullet handy
 - May need to explicitly disable/prevent some/most/all dangerous permutations
 - No known/well defined “Incompatibility map”, explicit technical documentation may help
 -

PRBS

- PHY work and PRBS support
 - no clear shape of the API at this point
 - difficult tradeoffs between flexibility and “portability”
 - PHY experts not engaged in SW development
 - integrated NIC vendors not engaging
 - open question about PRBS being a networking vs PHY subsystem functionality

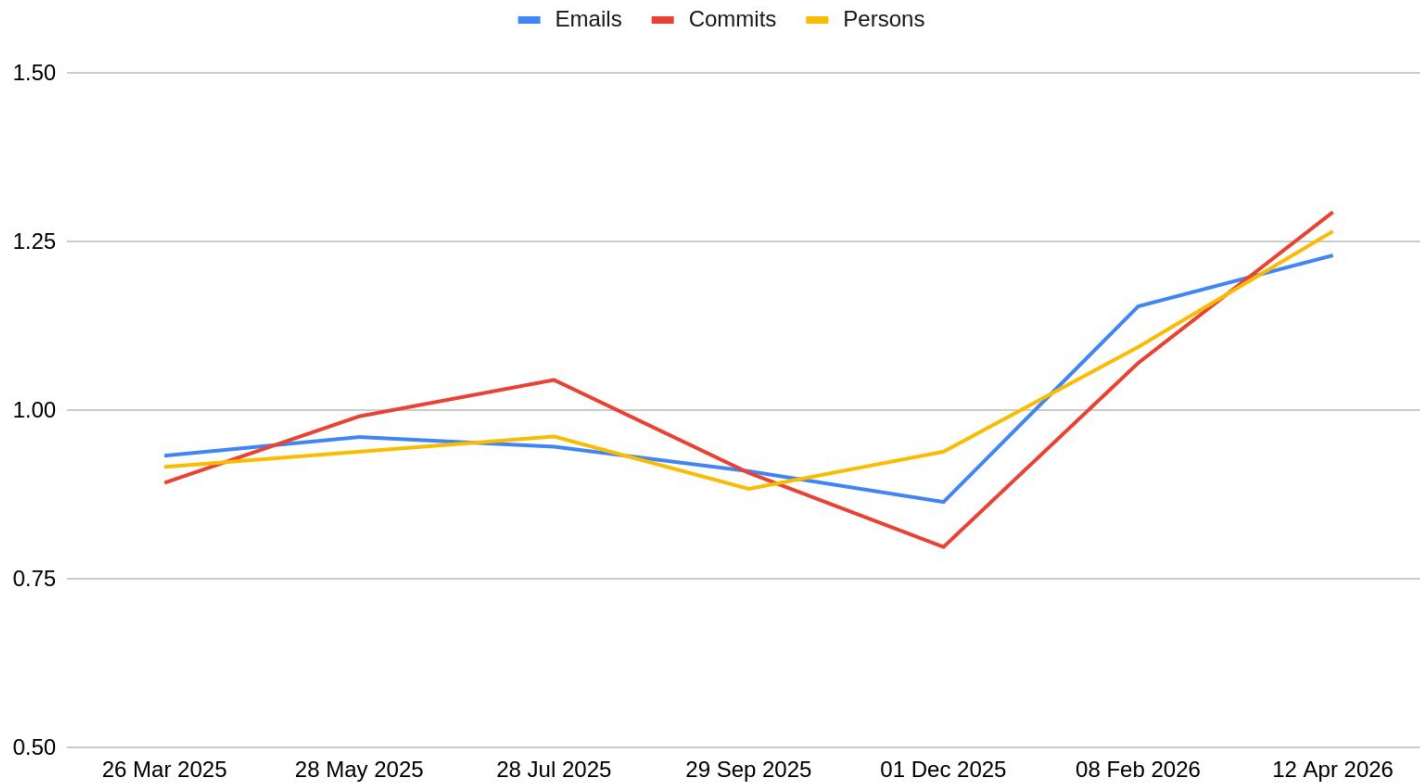
YNL use within kernel

- writing YAML specs for old families error prone, and vice versa, usually discovers pre-existing bugs like using different types in policy vs `nla_get_*`
- discussion about changing the way we declare and access attributes in the kernel

Topics covered in the main track

- BPF Timestamping 2.0: A Novel Kernel Latency Monitoring Framework (Jason, yesterday)
- skb->extensions and BPF annotations (Jakub S, tomorrow)

Development volume



Patch scoring

- Active reviewing (“reviewer score”)
 - individual
 - company
- Patches previously merged
- No failing checks
- Low revision count
- Outstanding comments