NIPA CI

How can it be even more helpful?

Matthieu Baerts (NGI0)

Netdev/NIPA CI

- Thanks Jakub for having initiated and continued the project!
- 2020-2023: Static analytic tests only
- 2024: Functional tests: kselftests, KUnit 🎉
  - It helped to increase the trust in the modifications that have been made, no?

Info

- Patchwork: for maintainers
  - It might not be clear that developers are not supposed to look: write it somewhere?
  - Developers might be able to find themselves what to fix, before reviewers, no?
  - That’s why there are no notifications in case of errors

- Functional tests: series are not individually tested
  - All pending patches are tested together (if they passed the build)
  - Needed due to the number of patches, and the time to run the tests
  - Does that work well?

Next steps

- Drivers should hopefully be tested soon™
- Anything else?
    - (Some small improvements listed here)
    - (more ideas in the next slides)

Existing tests not validated?

- Are there some subsubsystems that are not validated or only lightly covered?
- What about the kselftests that are unstable and currently ignored?
  - Need for an external repo to manage them? Not just Jakub / Stanislav + tracking?
  - Only some subtests are failing, not everything?
- How can we get them tested?
- What about external test suites?
  - e.g. Packetdrill tests are integrating selftests now
  - Possibility to have external runners, e.g. to validate MPTCP Packetdrill tests

In case of failures

- If no HW is required, they can be reproduced locally:
    - 'virtme-ng' can help here, but is the environment easily reproducible?
    - e.g. NIPA is using Amazon Linux with some tools that are built manually but should be up-to-date, while BPF is using Ubuntu ~~20.04~~ 24.04 images as a base, with some old tools.
    - Maybe the environment can be available in public container images? Is there a need?

- With MPTCP, easy to have the same environment (but less complex):

```
$ cd [kernel source code]
$ docker run -v "${PWD}:${PWD}:rw" -w "${PWD}" --privileged --rm -it \
    --pull always mptcp/mptcp-upstream-virtme-docker:latest \
    auto-normal
```

CI in other (network) subsystems?

- How to have a similar infrastructure to automatically validate patches?
  - NIPA is currently home-built, and specific to the Netdev case.
  - Is there a need?

- MPTCP subsystem is having a similar infrastructure, but still, it is not simple:
  - Applying patches from the ML: no public services, a server is needed
  - Running tests somewhere (GitHub Actions with KVM support), publishing results and logs somewhere else, tracking regressions (GitHub Pages), etc. and automating all of that, taking into account issues with Patchwork, security issues, etc.
  - Maybe some resources can be shared here if there is a need?

Stable versions

- What about supporting stable versions?
  - KSelftests should be supporting all previous versions: some CIs are running the last stable version of the kselftests suite on all supported stable kernel versions[1].

- This is a hard task to do and maintain:
  - Should there be an exception for network kselftests?
  - Or should this be enforced?
  - (Also, not sure many people (or their companies) care about stable versions :))

[1]: with QEmu, but without KVM acceleration…

Anything else?

- New tests: moving more code to lib.sh?