Linux Network Device Testing

An easy to use network driver and firmware testing framework

Lee Trager <ltrager@meta.com>

Meta

Motivation

- Needed to quickly and easily test firmware and out of tree Linux driver
- Framework required to work in a variety of environments
- Needed something very easy to both run and develop tests for
- Tests need to be easy to understand when debugging a failure

Inspiration

- https://git.kernel.org/pub/scm/fs/xfs/xfstests-dev.git/
- Originally developed for XFS but support has grown 11 different Linux filesystems
- Easy to run on any Linux system
- Integrated into automatic kernel filesystem patch testing
- Supports both generic tests applicable to all filesystems as well as filesystem specific tests
- Provides a number of helper functions to easily and consistently write tests
- Tests are validated using a return code and comparing STDOUT with a reference file
- Most tests are bash scripts

Philosophy

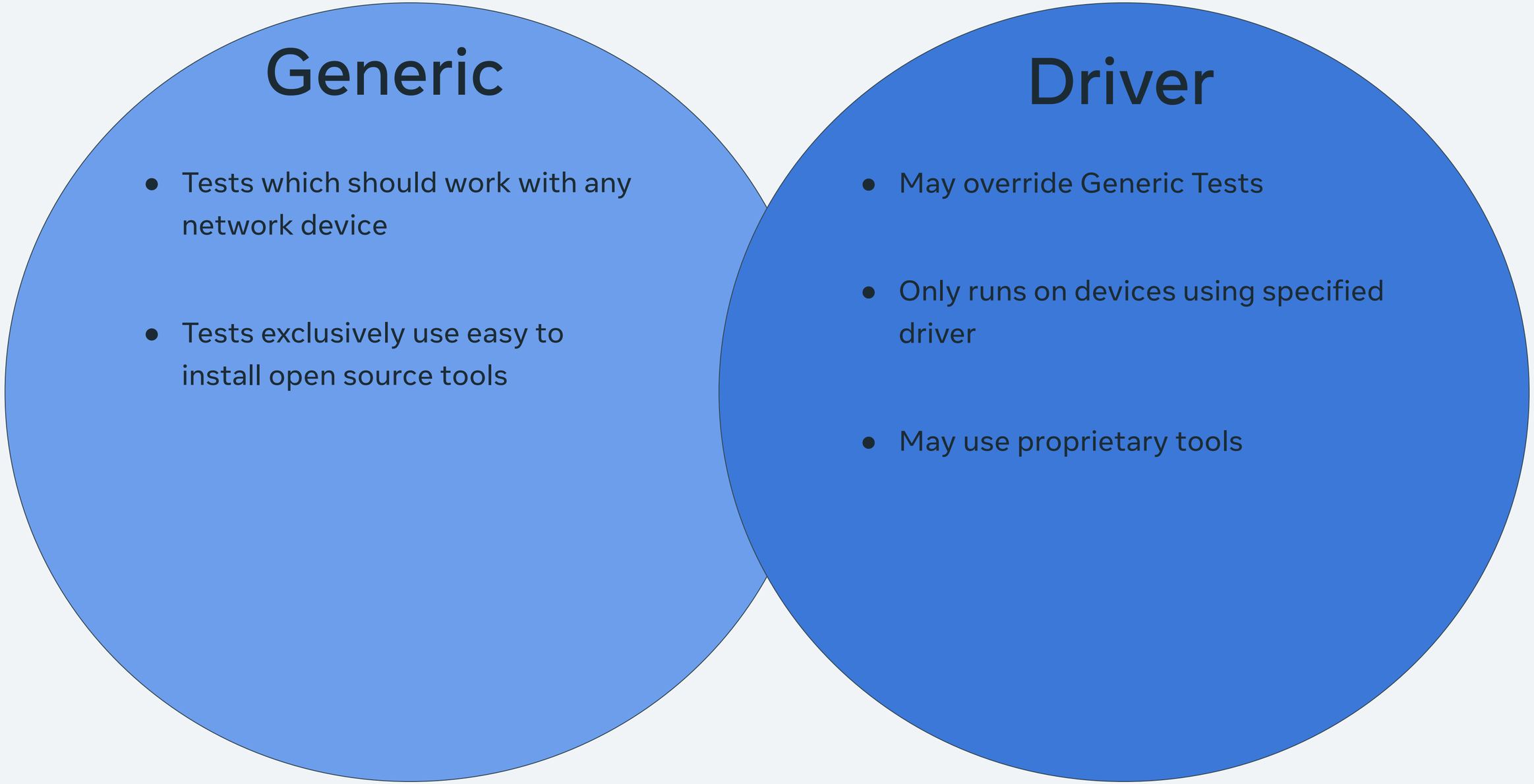
- No networking or environmental knowledge needed to execute
- Test suite can run in a standard Linux environment
- Tests are easy to understand by any Linux network device developer
- Tests are provided with basic information about what is being tested(NIC, MAC, PCI BUS, etc)
- LNDT provides test isolation, tests can modify link status or remove driver without having to worry about cleanup

Interoperability

Tests

- Can be any executable but Bash scripts are preferred
  - LNDT provides a Bash library for common functions
- LNDT passes test target and environmental information as environment variables
- Tests may optionally provide a configuration file to specify requirements
  - Link status
  - Remote resource such as iperf
  - Override expected test executable name
  - Required commands
  - Disable test
- Tests are validated by checking its return code
- Optionally tests can also have their output validated
  - On failure a patch is produced to clearly show differences
  - Test configuration may provide different expected output per PCI ID
- Tests may be skipped
  - Happens automatically when LNDT detects a lack of a required resource specified in test configuration
  - Tests may signal to LNDT that they skipped running by returning 42

Test Targets



Post Tests

- Post tests are rerun after every test
- Used to analyze ethtool statistics, system logs, and senor values
- Makes it easy to pinpoint which test caused an error

LNDT Flow

User

- All arguments are optional
- May specify
  - Target NIC, driver, MAC, PCI BUS
  - Out of tree driver path
  - Remote host for network tests(ping, iperf, etc)
  - Test filters
  - Verbosity
  - Log output

LNDT Setup

- Auto discovers all NICs
- Captures system information, NIC information, and link status
- Filters tests based on user input, available hardware, and environment

Test

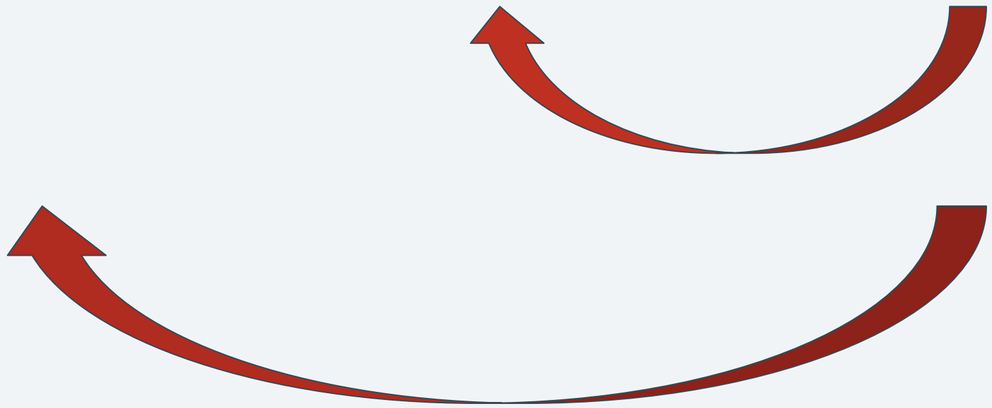
- Validate test can run in environment
- Optionally setup logging
- Run test

Reset

- Reload driver if needed
- Ensure link is back to initial state

Post Test

- Validate test can run in environment
- Optionally setup logging associated with parent test
- Run test



Demo

Future Work

- Open source
- More tests!
- Remote daemon to launch required services and return tcp dumps
- 100% unit test coverage
- Linux Self Test integration
- Integration into public test CI
- NetworkManager/networkd support

Questions and comments