

Kernel Networking Unit Tests

Since Android 5.0, proper operation of the Android networking stack on Linux kernels requires a number of commits that were upstreamed relatively recently or have not yet made it upstream. It is not easy to manually verify the required kernel functionality or track the missing commits, so the Android team is sharing the tests it uses to ensure the kernel behaves as expected.

Why run the tests?

These tests exist for three main reasons:

1. The exact version of the Linux kernel used on a device is typically device-specific, and it is difficult to know whether any kernel will work properly without running the tests.
2. Forward-porting and back-porting the kernel patches to different kernel versions or different device trees may introduce subtle issues that can be impossible to spot without running the tests. For example, during development the initial versions of certain devices had UID routing patches forward-ported from android-3.4 instead of cherry-picked from android-3.10, and did not behave correctly.
3. New networking features may require new kernel functionality or kernel bug fixes.

If the tests do not pass, the device's network stack will behave incorrectly, causing user-visible connectivity bugs such as falling off Wi-Fi networks. The device will likely also fail Android Compatibility Test Suite (CTS) tests.

Using the tests

The tests use [User-Mode Linux](http://user-mode-linux.sourceforge.net/) (<http://user-mode-linux.sourceforge.net/>) to boot the kernel as a process on a Linux host machine. See [Establishing a Build Environment](https://source.android.com/source/initializing.html) (<https://source.android.com/source/initializing.html>) for suitable operating system versions. The unit test framework boots the kernel with an appropriate disk image and runs the tests from the host file system. The tests are written in Python 2.x and use TAP interfaces to exercise kernel behaviour and the socket API.

Compiling the kernel for ARCH=um

For the tests to run, the kernel must compile for ARCH=um SUBARCH=x86_64. This is a supported architecture upstream and in the common Android kernel trees (e.g., android-3.10, android-3.18). But sometimes device kernels do not compile in this mode because device trees contain device-specific or hardware-specific code in common files (e.g., sys/exit.c).

In many cases, it's sufficient to ensure that hardware-specific code is behind an `#ifdef`. Typically this should be an `#ifdef` on a configuration option that controls the specific feature relevant to the code. If there is no such configuration option, put hardware-specific code inside `#ifndef CONFIG_UML` blocks.

In general, fixing this should be the responsibility of the kernel tree provider (e.g., chipset or SoC vendor). We're working with OEMs and vendors to ensure that current and future kernels will compile for ARCH=um SUBARCH=x86_64 without requiring any changes.

Running the tests

The tests are at [kernel/tests/net/test](https://android.googlesource.com/kernel/tests/+/master/net/test) (<https://android.googlesource.com/kernel/tests/+/master/net/test>). It is recommended that the tests **be run from AOSP master** because they are the most up-to-date; in some cases, kernel features that are necessary for proper operation in a given Android release do not yet have full test coverage in the given release. For information on how to run the tests, see the [kernel network test README file](https://android.googlesource.com/kernel/tests/+/master/net/test/README) (<https://android.googlesource.com/kernel/tests/+/master/net/test/README>). Basically, from the top of your kernel tree, run:

```
ANDROID_TREE/kernel/tests/net/test/run_net_test.sh all_tests.sh
```

Passing the tests

The kernel network test Python source files contain comments that specify kernel commits that are known to be required to pass the tests. The tests should pass in the common kernel trees - at least the android-3.10 and android-3.18 branches in the [kernel/common](https://android-review.googlesource.com/#/q/project:kernel/common) (<https://android-review.googlesource.com/#/q/project:kernel/common>) project in AOSP. Therefore, passing the tests on a kernel tree that's derived

from 3.10 or 3.18 should mostly be a matter of cherry-picking the patches from these trees.

Contributing

Reporting issues

Please report any issues with the kernel network tests in the [Android issue tracker](https://code.google.com/p/android/issues/entry?template=Developer%20bug%20report) (<https://code.google.com/p/android/issues/entry?template=Developer%20bug%20report>) with the [Component-Networking](https://code.google.com/p/android/issues/list?q=label%3AComponent-Networking) (<https://code.google.com/p/android/issues/list?q=label%3AComponent-Networking>) label.

Documenting commits and adding tests

Please report issues as described above, and if possible upload a change to fix the issue, if:

- The tests do not pass on the common kernel trees
- You find a necessary commit that is not mentioned in the source comments,
- Getting the tests to pass on upstream kernels requires major changes
- You believe that the tests are overspecified, or the test fail on future kernels
- You'd like to add more tests or more coverage to existing tests.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](http://creativecommons.org/licenses/by/3.0/) (<http://creativecommons.org/licenses/by/3.0/>), and code samples are licensed under the [Apache 2.0 License](http://www.apache.org/licenses/LICENSE-2.0) (<http://www.apache.org/licenses/LICENSE-2.0>). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (<https://developers.google.com/terms/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated August 8, 2017.