

## Cellular Network Infrastructure - Bug #3268

### execute TTCN3 test suites against "latest" feeds

05/15/2018 08:01 PM - laforge

<b>Status:</b> Resolved	<b>Start date:</b> 05/15/2018
<b>Priority:</b> High	<b>Due date:</b>
<b>Assignee:</b> osmith	<b>% Done:</b> 100%
<b>Category:</b>	
<b>Target version:</b>	
<b>Spec Reference:</b>	
<b>Description</b>	
<p>It would be good to run our test suites not only against current master builds, but also against the "latest" packages. This way we can nicely show the amount of bugs fixed (and/or regressions pending) between the latest tagged version[s] and the current master.</p> <p>In I6a564206dd81743deb1eb27eca7081bc333d7434 of docker-playground.git I already introduced Dockerfiles:</p> <pre>osmo-bsc-latest osmo-bts-latest osmo-ggsn-latest osmo-hlr-latest osmo-hnbgw-latest osmo-mgw-latest osmo-msc-latest osmo-sgsn-latest osmo-sip-latest osmo-stp-latest</pre> <p>What's missing is mainly</p> <ul style="list-style-type: none"><li>• generalization of the "jenkins.sh" scripts to either test master or latests</li><li>• migration of ttcn3 jenkins jobs to JJB, so we can template them and duplicate them</li></ul> <p>Potential problems: The tests use hard-coded IP addresses, and docker cannot have two networks with overlapping addresses, even if there's no routing between them :( We would then have to make sure that no ttcn3-bts-test-master and ttcn3-bts-test-latest jobs are running in parallel on the same build slave. A possible work-around would be to have separate build slaves for master/latest. Using different IP addresses (172.18.9.0/24 for ttcn3-bts-test-master and 172.17.9.0/24 for ttcn3-bts-test-latest) is of course desirable, but this would require different osmo-*.cfg and different TTCN-3 testsuite config files, etc :(((</p>	
<b>Related issues:</b>	
Related to Cellular Modem Information - Feature #3658: Create a "osmocom-bb-h...	<b>New</b> 10/16/2018
Related to OsmoPCU - Bug #2890: OsmoPCU TTCN-3 test suite not executed by jen...	<b>Resolved</b> 01/27/2018
Related to Cellular Network Infrastructure - Bug #3767: Most ttcn3-*test-lat...	<b>Resolved</b> 01/24/2019

### History

#1 - 05/30/2018 02:21 PM - laforge

- Tags set to TTCN3

#2 - 06/23/2018 06:18 PM - laforge

- Priority changed from Normal to High

#3 - 09/30/2018 08:13 AM - laforge

- Assignee changed from lynxis to osmith

#4 - 10/10/2018 09:04 AM - osmith

- Status changed from New to In Progress

#5 - 10/10/2018 09:21 AM - osmith

Potential problems: The tests use hard-coded IP addresses, and docker cannot have two networks with overlapping addresses, even if there's no routing between them :(

We could do it like gitlab CI: each docker instance is isolated inside a virtual machine. Then we don't need different IPs when running "latest" and "master". And as additional benefit, the config files could be used without any changes when running the TTCN3 testsuite *without* Docker (when we use 127.0.0.1 as the IPs).

[laforge](#): do you want that? If so, I could make a follow-up issue for that.

A possible work-around would be to have separate build slaves for master/latest

I'll go with that workaround for now.

#### #6 - 10/10/2018 09:50 AM - laforge

On Wed, Oct 10, 2018 at 09:21:50AM +0000, [redmine@lists.osmocom.org](mailto:redmine@lists.osmocom.org) wrote:

Potential problems: The tests use hard-coded IP addresses, and docker cannot have two networks with overlapping addresses, even if there's no routing between them

We could do it like gitlab CI: each docker instance is isolated inside a virtual machine. Then we don't need different IPs when running "latest" and "master".

how is this done exactly? do you hav a pointe explaining the setup?

And as additional benefit, the config files could be used without any changes when running the TTCN3 testsuite *without* Docker.

I don't see how that would work, sorry. I think there is too much complexity in there.

A possible work-around would be to have separate build slaves for master/latest

I'll go with that workaround for now.

sounds reasonable.

#### #7 - 10/10/2018 12:31 PM - osmith

Potential problems: The tests use hard-coded IP addresses, and docker cannot have two networks with overlapping addresses, even if there's no routing between them :(

We could do it like gitlab CI: each docker instance is isolated inside a virtual machine. Then we don't need different IPs when running "latest" and "master".

how is this done exactly? do you hav a pointe explaining the setup?

"Projects hosted in GitLab can have CI tasks defined in their .gitlab-ci.yml files. These tasks are performed by runners which are essentially virtual machines which run your builds in Docker containers. These machines can run any of your builds that are compatible with Docker."

(Source: <https://about.gitlab.com/2016/04/05/shared-runners/>)

"Shared runners" are the ones that can be used for free on gitlab.com (vs. running your own gitlab instance). These runners can have different "executors" (environments to run the docker container in, SSH, shell, VirtualBox, ...), and from the config linked in the article above, they are using the "docker+machine" executors.

"The Docker Machine is a special version of the Docker executor with support for auto-scaling. It works like the normal Docker executor but with build hosts created on demand by Docker Machine."

(Source: <https://docs.gitlab.com/runner/executors/README.html#docker-machine>)

Docker Machine is described here, and it seems to be possible to make it use VirtualBox to spawn new VMs on a Linux host:

<https://docs.docker.com/machine/overview/>

With that being said, using "Docker Machine" might be a bit over-engineered for our use case. We could also run a Qemu VM with an image we created before, connect to it via SSH, execute the test, and kill it afterwards. I think this can be done in a ~100 line python script.

#### #8 - 10/10/2018 02:06 PM - osmith

To make writing the new JJB YML files easier, I've dumped the existing Jenkins jobs with [jenkins-job-wrecker](#) and converted them to YML. I will still need to rewrite them from scratch, but that should make it easier to see the what is common/different between the ttcn3-\* jobs.

#### #9 - 10/10/2018 02:42 PM - osmith

- % Done changed from 0 to 10

WIP branches in [docker-playground.git](#) and [osmo-ci.git](#) are named `osmith/ttcn3-latest`.

In order to avoid conflicting with the existing Jenkins jobs (`ttcn3-msc-test` etc.), I'm naming the generated jobs `ttcn3-test-*`.

#### #10 - 10/11/2018 12:20 PM - osmith

The [TTCN3 tab](#) over at Jenkins does not only list TTNC3 testsuites. I've thought about how to generalize this and came up with giving the new jobs a different naming scheme:

`testsuite-{container-suffix}-{testsuite-name}`

`testsuite-name` is the name of the folder in [docker-playground.git](#), without "-test" at the end.

examples:

- `testsuite-master-ttcn3-mgw`
- `testsuite-latest-ttcn3-mgw`
- `testsuite-master-m3ua`
- `testsuite-latest-m3ua`

#### #11 - 10/11/2018 01:14 PM - osmith

- % Done changed from 10 to 50

Status update:

I've introduced a new `CONTAINER_SUFFIX` variable in all `jenkins.sh` files of the testsuites. By default it is set to "master", and the Jenkins job can override it to "latest".

Besides that, there is a new `docker_images_require()` function in `jenkins-common.sh` ([docker-playground.git](#)). It accepts docker image names (`ttcn3-sip-test`, `debian-jessie-build`, ...) and builds all of them which do not exist. That function gets called in all `jenkins.sh` files now. This has the following advantages:

- all images will be present before the testsuite starts (or the script will abort there, before running `network_create`; right now it just runs the testsuite even if images are missing)
- we can use the `$CONTAINER_SUFFIX` variable in the required image names
- this list can easily be maintained (vs. the top level makefile, which is not really maintained and it's easy to make mistakes there like depending on an image name that is not a target in the makefile, which will then always appear to be built because the folder exists)
- it is not necessary to run an additional command (`make ...`) before running `jenkins.sh`, as it will build the images. this simplifies the jenkins jobs (right now each jenkins job is calling `make` on the images that need to be built for its job before running `jenkins.sh`)

So far one short `testsuite.yml` seems to be able to generate all the `testsuite-*` jenkins jobs :)

On a related note, I've had a discussion in the local hackerspace about the overlapping network addresses problem. Someone said, it should be possible to use overlapping network addresses if we configure the network bridges ourselves and do not let docker manage them, just tell docker to use them. Something like that, I have not looked into it more. Maybe this is helpful.

#### #12 - 10/11/2018 06:37 PM - zecke

Can't you use: <https://wiki.jenkins.io/display/JENKINS/Build+Blocker+Plugin?>

From the online help:

Block build if certain jobs are running

Help for feature: Block build if certain jobs are running

Enable the build blocker to prevent this job to run while one of the here configured other jobs are running. The blocked jobs stays in the queue until all blocking jobs are not running.

**#13 - 10/12/2018 06:40 AM - laforge**

On Thu, Oct 11, 2018 at 12:20:23PM +0000, osmith [REDMINE] wrote:

The [TTCN3 tab](#) over at Jenkins does not only list TTNC3 testsuites. I've thought about how to generalize this and came up with giving the new jobs a different naming scheme:

Please don't rename the existing jobs, even if it ends up being inconsistent. This will

- break all hyperlinks/bookmarks/...
- break the history (i.e. if we look at the new jobs output, we don't have the historical data)
- consume more disk space as we will have two jobs and their build history/artefacts in parallel for some time and our jenkins master for economic reasons is running **just** on a ~250 GB SSD.

It's fine to give new names to the new jobs, but please keep the old ones as-is.

Also, regarding the "latest" build jobs: It seems their test results analyzer is broken. The build are successful as per 'console output', but the junit XML is not imported

Examples:

[https://jenkins.osmocom.org/jenkins/job/testsuite-latest-ttcn3-mgw/test\\_results\\_analyzer/](https://jenkins.osmocom.org/jenkins/job/testsuite-latest-ttcn3-mgw/test_results_analyzer/)  
[https://jenkins.osmocom.org/jenkins/job/testsuite-master-ttcn3-bsc/test\\_results\\_analyzer/](https://jenkins.osmocom.org/jenkins/job/testsuite-master-ttcn3-bsc/test_results_analyzer/)

**#14 - 10/12/2018 06:40 AM - laforge**

On Thu, Oct 11, 2018 at 01:14:05PM +0000, osmith [REDMINE] wrote:

On a related note, I've had a discussion in the local hackerspace about the overlapping network addresses problem. Someone said, it should be possible to use overlapping network addresses if we configure the network bridges ourselves and do not let docker manage them, just tell docker to use them. Something like that, I have not looked into it more. Maybe this is helpful.

Thanks, I'd actually prefer to get upstream fix their issues. Linux network namespaces exist to provide isolation, and hence there's nothing that prevents you from having the same internal addresses/subnets in different namespaces. The restriction docker imposes is an undue constraint on the capabilities of the Linux kernel network stack.

It may be as simple as disabling a single line inside docker source code itself, I haven't yet had the time to look at it in more detail. Might be worth to at least report it as an issue with them.

**#15 - 10/12/2018 01:14 PM - osmith**

Can't you use: <https://wiki.jenkins.io/display/JENKINS/Build+Blocker+Plugin?>

Good idea, implemented it that way now. Thank you!

Please don't rename the existing jobs, even if it ends up being inconsistent.

Okay. I'll delete the new ones, and change the naming scheme of the JJB config to match the existing ones. It would be possible to overwrite the existing (manually created) jobs that way, but I will comment that out until the "latest" jobs have been tested properly.

Also, regarding the "latest" build jobs: It seems their test results analyzer is broken. The build are successful as per 'console output', but the junit XML is not imported

I did not have the test result analyzer configured in the jenkins job builder config at that point. Later I did, and the test result analyzer became functional (I ran other jobs for that though).

**#16 - 10/12/2018 03:30 PM - osmith**

- % Done changed from 50 to 70

The naming scheme has been adjusted, and I've kicked off all builds at least once. Some run through, but quite a few will need further tweaks. I can look into this more on Monday.

- nplab-m3ua-test-latest (OK)

- nplab-sua-test-latest (OK)
- ttcn3-bsc-test-latest (FAIL)
  - all tests fail (why?)
- ttcn3-bsc-test-sccplite-latest (FAIL)
  - runs forever (sudo waits for password entry in ttcn3-tcpdump-start.sh in osmo-ttcn3-hacks.git?)
- ttcn3-bts-test-latest (FAIL)
  - missing docker container: "osmocom-bb-host-latest"
  - > osmocom-bb-host is not built for latest in OBS
- ttcn3-ggsn-test-latest (OK)
- ttcn3-hlr-test-latest (FAIL)
  - all tests fail
  - Could not connect IPA socket from "" port -1 to "172.18.10.20" port 4222; check your configuration
- ttcn3-mgw-test-latest (OK)
- ttcn3-msc-test-latest (FAIL)
  - all tests fail (why?)
- ttcn3-sgsn-test-latest
  - all tests fail (why?)
- ttcn3-sip-test-latest
  - all tests fail (why?)

#### #17 - 10/15/2018 11:41 AM - osmith

I've added ^ttcn3-bsc-test.\* etc. as blocking jobs regex (for "Block build if certain jobs are running") to all **existing** jobs (ttcn3-bsc-test in this example).

#### #18 - 10/15/2018 02:20 PM - osmith

Debugging the failing jobs, one after another.

ttcn3-bts-test-latest: changed to depend on osmocom-bb-host-master, even if we are testing against latest. Because there is no "latest" tag for osmocom-bb (so we can't build a osmocom-bb-host-latest docker container).

ttcn3-bsc-test-sccplite-latest: figuring out why this is stuck and runs forever. Unless assumed earlier, this is not related to sudo - all processes are running as root inside the Docker container anyway. I can reproduce this on my laptop, and it only happens when running -latest, not for -master. I am certain that it is not related to ttcn3-tcpdump-start.sh now, because it says that this script was executed successfully before it hangs:

```
MC2> MTC@0280f2461f40: Starting external command `../ttcn3-tcpdump-start.sh
BSC_Tests.TC_ctrl_msc_connection_status'.
Waiting for tcpdump to start... 0
MTC@0280f2461f40: External command `../ttcn3-tcpdump-start.sh
BSC_Tests.TC_ctrl_msc_connection_status' was executed successfully (exit status: 0).
MTC@0280f2461f40: Test case TC_ctrl_msc_connection_status started.
```

Right now I'm installing a few debugging utils (pstree, strace, ...) in my local debian-stretch-titan container, so I can enter the docker container while it hangs and figure out what is hanging exactly.

#### #19 - 10/15/2018 02:37 PM - osmith

it is hanging at the BSC\_Tests processes (there are four of them). Might be a race condition? strace says that they are all stuck at the epoll\_wait syscall.

#### #20 - 10/15/2018 08:30 PM - laforge

Hi Oliver,

On Mon, Oct 15, 2018 at 02:20:29PM +0000, osmith [REDMINE] wrote:

ttcn3-bts-test-latest: changed to depend on osmocom-bb-host-master, even if we are testing against latest. Because there is no "latest" tag for osmocom-bb (so we can't build a osmocom-bb-host-latest docker container).

Not nice, but ok as an interim solution for now.

ttcn3-bsc-test-sccplite-latest: figuring out why this is stuck and runs forever. Unless assumed earlier, this is not related to sudo - all processes are running as root inside the Docker container anyway. I can reproduce this on my laptop, and it only happens when running -latest, not for -master. I am certain that it is not related to ttcn3-tcpdump-start.sh now, because it says that this script was executed successfully before it hangs:

please simply disable this job for the time being and work on other topics meanwhile.

Right now I'm installing a few debugging utils (pstree, strace, ...) in my local debian-stretch-titan container, so I can enter the docker container

while it hangs and figure out what is hanging exactly.

The problem is likely that one of the tests is executing without a proper guard timer, i.e. it's waiting indefinitely for some event to occur which then never happens. The log files and possibly pcap file should indicate where exactly. I guess it's ok to put this aside until somebody with more experience can have a look and meanwhile work on other stuff.

**#21 - 10/16/2018 09:50 AM - osmith**

- % Done changed from 70 to 90

please simply disable this job for the time being and work on other topics meanwhile.

Done. I've polished the patches to finish this up and submitted them:

<https://gerrit.osmocom.org/#/c/docker-playground/+/11364> osmo-\* -latest: s/nightly/latest/g in Dockerfile  
<https://gerrit.osmocom.org/#/c/docker-playground/+/11365> jenkins-common.sh: add docker\_images\_require()  
<https://gerrit.osmocom.org/#/c/docker-playground/+/11366> jenkins.sh: new IMAGE\_SUFFIX environment variable  
<https://gerrit.osmocom.org/#/c/docker-playground/+/11367> symlinks: ttcn3-bsc-test-sccplite  
<https://gerrit.osmocom.org/#/c/docker-playground/+/11368> symlinks: nplab-m3ua-test, nplab-sua-test  
<https://gerrit.osmocom.org/#/c/docker-playground/+/11369> Remove top-level Makefile  
<https://gerrit.osmocom.org/#/c/osmo-ci/+/11370> jobs: testsuite.yml for all ttcn3/nplab jobs

**#22 - 10/16/2018 02:11 PM - osmith**

- Related to Feature #3658: Create a "osmocom-bb-host-latest" docker container added

**#23 - 11/05/2018 09:55 AM - osmith**

- Status changed from In Progress to Resolved

- % Done changed from 90 to 100

All TTCN3 related jenkins jobs (latest and master) are now generated with:

<https://git.osmocom.org/osmo-ci/tree/jobs/ttcn3-testsuites.yml>

**#24 - 11/05/2018 02:02 PM - msuraev**

- Related to Bug #2890: OsmoPCU TTCN-3 test suite not executed by jenkins added

**#25 - 01/24/2019 05:58 PM - osmith**

- Related to Bug #3767: Most ttcn3-\* -test-latest jenkins jobs are failing added