

OsmoGSMTester - Feature #2707

Add support for verifying content of log files after test is run

12/04/2017 04:30 PM - pespin

Status:	New	Start date:	12/04/2017
Priority:	Normal	Due date:	
Assignee:	osmo-gsm-tester	% Done:	0%
Category:			
Target version:			
Spec Reference:			
Description			
<p>We need to add some pieces to osmo-gsm-tester that lets us do some verifications on each object once the test is finished.</p> <p>We could have a generic verification phase after the test .py file has finished: During test run, when an object instance is requested by the test and allocated by suite.py, we can add that object to a list of objects to verify.</p> <p>When the test execution is passed, for each object we call obj.verify(), which will trigger an Exception if some verification fails.</p> <p>We can then have a generic API providing helpers to be used inside verify() of each object, as for each object we want to check different information and sources of information. This helpers can also be used by tests to verify specific requirements for that test which doesn't apply in general. Some verification sources I can think of:</p> <ul style="list-style-type: none">- log file generated by each object (process).- pcap file owned by each object (through its PcapRecorder object). <p>Log files:</p> <p>It would be nice to have an API which allows you to pass the log file, and then another file with strings/regexp to match. If one of those strings/regexp matches in the file, then a exception is triggered. We also want a similar API to do the same but to verify that it doesn't contain a specific string/regexp. The first one can be useful, for instance, to have a list of error strings which should never appear under normal circumstances and that we want to investigate in case they happen. That's the case for string "no space for uplink measurement" (see #2329). This way we can catch regressions quite quickly.</p> <p>Pcap files:</p> <p>It would be nice to have an API both for:</p> <ul style="list-style-type: none">- generic verification: In general, we may want to check that specific filter never appears or that a specific filter must appear. This can be easily implemented using tshark.- specific verification: A test may want to verify some information from the pcap file. For instance, that a certain kind of packet was received, or that a special error is triggered, or that at least N packets matching a filter are being sent.- wait() trigger: It can be used either internally or by tests to wait for specific events to happen before doing something. <p>I'd first start with the log file part, then once we are done, go for the pcap part.</p>			

History

#1 - 12/04/2017 05:05 PM - neels

In general an excellent idea.

My first impression is that we should not introduce an entirely new generic test phase, but rather have those "post-run" conditions as part of the tests, e.g. as last lines in the test script do things like

```
assert not re.search("no space for uplink measurement", bts.get_stderr())
```

get_stderr() is part of the process.py api, may need some plumbing to comfortably get those from the test script for the right Process instances.

There may also be some convenience API to do a couple of those verifications in a single call, but still would prefer to limit the amount of magic that happens implicitly, rather have explicit conditions only.

I'd place the pcap filtering in a separate issue and make it available in similar API to be called by individual test scripts.

(If you're really aiming for generic across-the-board conditions, we might also add some greps in the jenkins job, grepping the log files sent back to jenkins? We wouldn't be able to report in XML which specific tests failed though.)