

Introduction

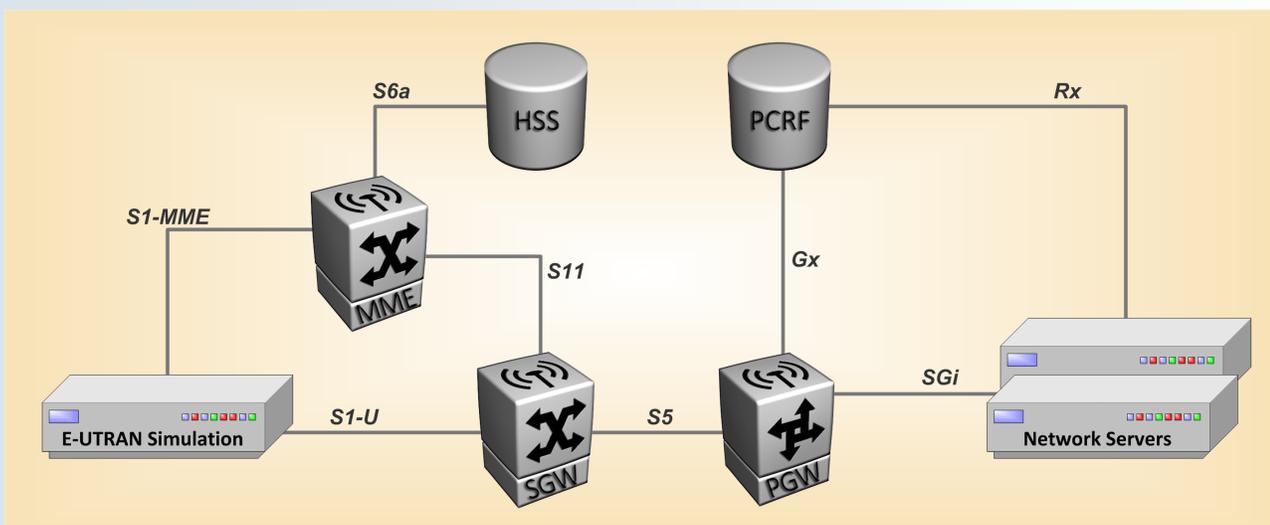
As the wireless industry moves toward a unified IP network that will carry both voice and data traffic, the Policy and Charging Rule Function (PCRF) will take on an increasingly important role in managing a service provider's network resources. The PCRF will be used to authorize a subscriber's bandwidth allocation based on multiple factors including the subscriber's past usage, the level of service a subscriber has purchased, and the amount of resources currently available in the network.

With the planned adoption of Voice over LTE (VoLTE), the PCRF's role in the service provider's network will become increasingly vital. It is therefore essential that the PCRF be fully tested to establish its behavior, performance, and capacity prior to deployment in a live network. Although we'll emphasize the importance of PCRF testing in this white paper, it's equally important to test all Network Elements (NE) in the network, therefore the testing principles and philosophies that we'll discuss can be applied to most any NE.

Two equally important yet uniquely different test methodologies are frequently employed for validating a NE prior to deployment in a live network. We'll refer to the first methodology as NE Testing in a Test Network and we'll refer to the second methodology as NE Testing in Isolation. Each methodology provides unique and valuable insight into the performance and behavior of a NE that is being evaluated and tested.

NE Testing in a Test Network

This first method (also referred to as end-to-end testing) usually consists of building a test network in a lab using all the same nodes that are deployed or planned for deployment in the service provider's live network. For obvious reasons the test network would generally contain a scaled down set of the hardware that the live network would contain. The goal of the test network would be to have an environment which would support doing all the same types of activities that would occur in the live network, but on a much smaller scale. When NEs in the core network are being tested using a test network, the test will usually consist of simulated User Equipment (UE) and Radio Access Network (RAN) communicating with simulated network servers. The following figure shows a graphical representation of this type of test network setup.

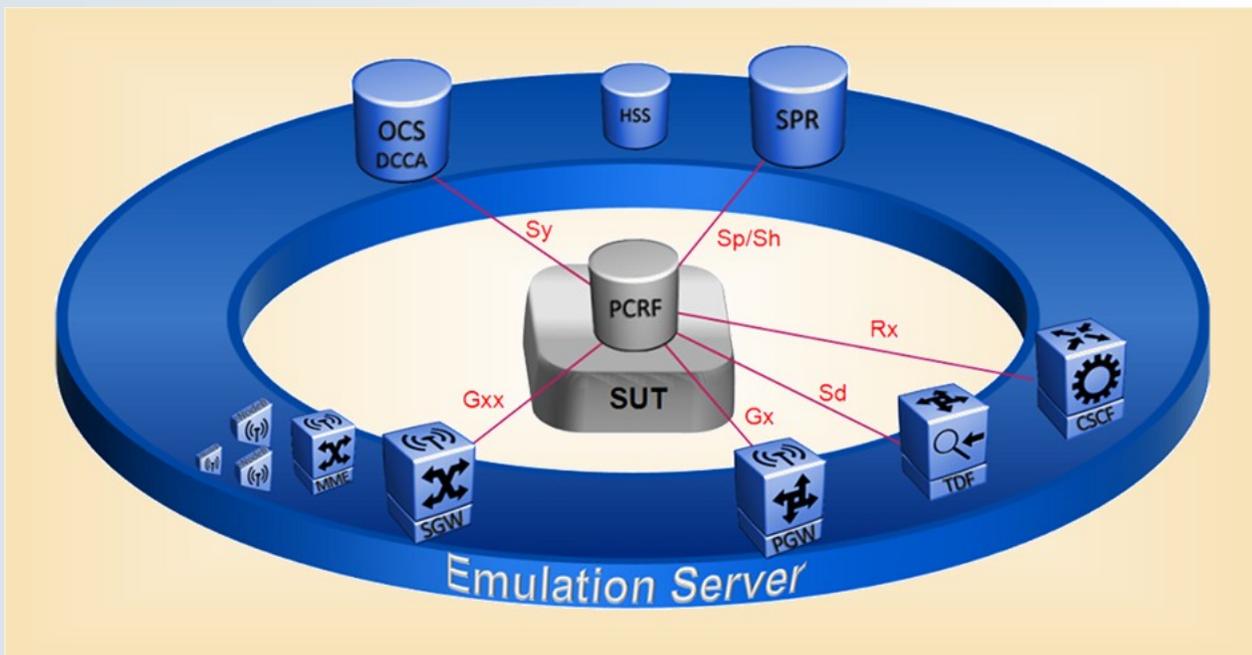


Testing in this manner is an absolutely essential part of any NE qualification plan and it provides some of the following key benefits:

- ◆ Ensures interoperability of equipment in the network.
- ◆ Provides a environment to evaluate the user's experience.

NE Testing in Isolation

Another widely used technique for testing an NE is to test the NE in isolation. Testing in isolation means that the NE is isolated from other NEs and tested solely with test equipment in a highly controlled environment. This test equipment is commonly referred to as an emulator or load generator because it emulates other NEs and it typically produces load on the NE under test. The following image shows an example of how the PCRF may be tested in isolation mode.



As you'll notice, it is possible for the test equipment to be attached to more than one application interface (i.e. Gx, Rx, or S6) on the NE under test. In addition to allowing each of the application interfaces to be tested, this also allows the test equipment to verify that actions generated on one application interface of the NE will result in the NE generating an appropriate action on another application interface with acceptable timing.

In this white paper we'll further explore some of the benefits that can be gained from testing the PCRF or other NEs in isolation. Our intention is not to suggest that this is a superior method of testing, but rather to highlight the benefits that can be realized from adding isolation testing to a qualification plan.

Benefits of Testing an NE in Isolation

Cost

PCRF performance is exponentially increasing to a capacity of tens of millions of subscribers with transaction rates in excess of 100,000/second. Building and maintaining a test network that can fully load a PCRF can be cost prohibitive. It would take numerous other NEs in addition to a significant amount of test equipment to build a test network that would be capable of fully loading the PCRF. When using an

emulator to test the PCRF in isolation, the PCRF can be fully loaded with much less equipment and in many cases a single emulation server would be adequate, which results in less capital being spent.

Direct Control

Using an emulator to test an NE like the PCRF allows the test operator to create a precise test that can produce the specific scenario that's desired. There's no need to modify the configuration of multiple NEs and UE simulators to get the desired results.

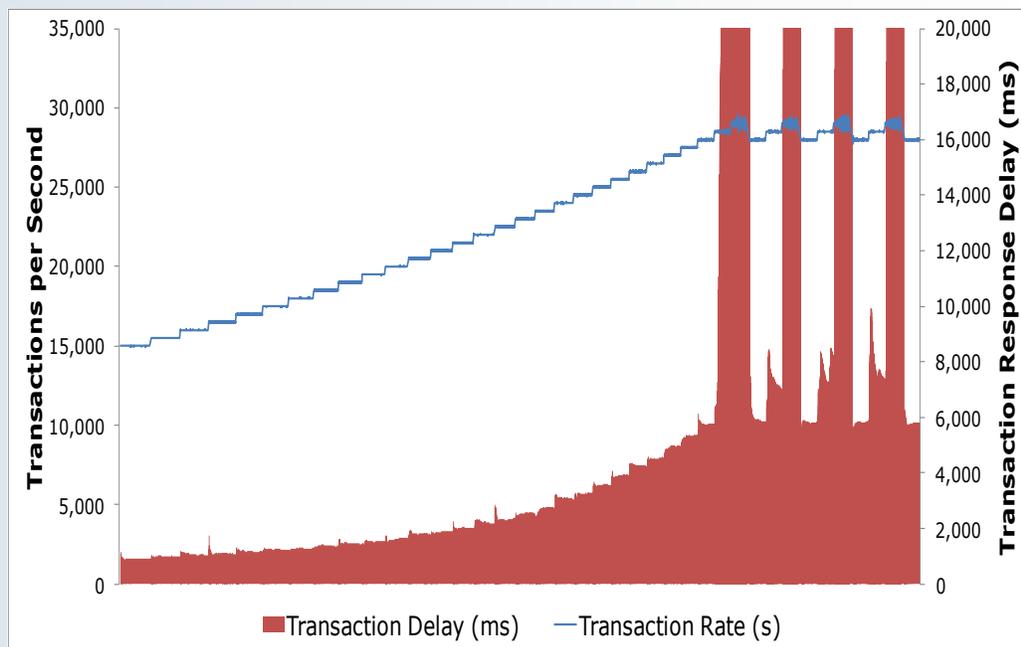
An emulator will also allow tests to be reproduced quickly and with a high degree of accuracy. An operator can create a library of previously executed tests with the knowledge of how those tests previously performed. This allows the operator to verify that future software loads have not degraded quality or performance.

Negative Testing

By using an emulator you'll be able to create scenarios that you would rarely encounter in a real-world. However, when these errors do occur, a NE should handle them gracefully. An emulator will provide the ability to create invalid messages or message sequences. For example, an emulator can generate invalid AVP values, insert or remove AVPs which would violate the protocol's specification, respond with errors, delay responses, and numerous other protocol violations.

Capacity and Performance

Determining the maximum transaction rate and subscriber capacity of a PCRF is vital to insure that it will hold up to the demands of a live network. An emulator will allow the operator to setup tests that will cause maximum capacity and load to be produced on the NE. Going even further, the operator will be able to exceed the capacity of the NE and verify that it can gracefully recover from overload conditions.



Test results determine the impact of transaction rate on response time

Easier to Determine Point of Failure

When using an end-to-end test network, finding the root cause of a failure can be difficult to isolate. This is partially due to the fact that every network element does its best to survive network problems

and absorb unsustainable transaction bursts. A network element's ability to survive error conditions or overload scenarios is a highly desirable characteristic in the live network, but when trying to validate individual NEs, you don't want other network elements to compensate for or to conceal its weaknesses. Each network element will typically have its own statistics collection capability which may or may not be consistent with other equipment that's in the network. An emulator will give you detailed statistics at each application interface. You will also be able to monitor statistics across application interfaces and correlate those statistics to a particular test activity.

What to Look for in a Policy Control Validation Solution?

When evaluating a test solution for a PCRF or other NE, make sure it has some, and preferably all, of the following features. They will provide you with the ability to push the NE to its limits and beyond. Pushing your NE to the limit will help you understand exactly what your NE is capable of and minimize unpleasant surprises once it's deployed in the live network.

◆ **Ability to reproduce events**

Using an emulator will allow the operator to quickly reproduce a network event or sequence of events. Attempting to recreate a scenario in a test network using real NEs could take countless hours of reconfiguration. Some NEs are not designed for quickly changing configurations and therefore it could take many man hours of work to change a configuration across all the NEs involved in a test scenario.

◆ **Results verification**

Additionally, the emulator should be able to verify that an action on one application interface will result in an appropriate action on another application interface. For example, you'll want to ensure that requests on the Rx interface will result in the appropriate rules being pushed to the PCEF across the Gx interface.

◆ **Ability to push a PCRF to its maximum capacity and performance**

As mentioned previously, today's PCRF devices are exponentially increasing their capacity toward handling tens of millions of simultaneously active subscribers while processing tens of thousands of transactions per second. A good network emulator should be able to achieve those capacities with a single piece of hardware rather than using multiple servers. Using a single piece of hardware also allows the emulator to host the entire subscriber pool rather than dividing subscribers into groups that must be separately configured and emulated across multiple nodes.

◆ **Ability to coordinate events across application interfaces**

To adequately test a PCRF, you need more control over the testing than just generating messages at some predetermined rate into each application interface. It is a requirement to synchronize messages on multiple application interfaces. For example when testing a PCRF, the emulated CSCF needs to wait until the emulated PCEF receives a successful IP-CAN Initialization answer before it generates its AAR message.

◆ **Ability to capture and store test detailed results**

A fundamental requirement of any test equipment is its ability to store results for off-line processing and analysis. Maintaining historical information from previous tests will allow you to establish a baseline for how well the NE performs. This can be useful for ensuring that future releases of the NE do not significantly degrade its performance or capacity.

The statistics that are captured by the emulator should have sufficient detail to allow correlation of information. Those statistics should also be synchronized so that statistics from one application interface can be correlated with statistics on another application interface for a given interval of time.

Historical information should also be maintained by the emulator to help build a complete picture of how the NE under test performed over the entire time that the test was active.

◆ **Ability to create both fixed and realistic interaction**

Many network issues are discovered by randomizing the behavior of the tests. Being able to set up tests that don't always generate the same sequence of events is an important component of any test plan. Look for an emulator that can create complex load profiles to accurately reproduce live network scenarios.

◆ **Ability to inject errors**

An emulator should be capable of injecting a variety of error scenarios that would be seen in the live network. Using an emulator will allow you to inject error messages and sequences into the NE and thereby verify that it can detect and recover from the error scenarios.

◆ **Turnkey Yet Flexible**

Look for an emulator which has knowledge about the applications that you're interested in testing. The emulator should be aware of the message contents and message sequences that are used in each of the applications you plan to use. This will allow you to get a quick start in your testing, but at times you'll still need to get involved in functional testing, therefore an emulator should also have the flexibility to customize virtually any aspect of the application protocol. This should include the ability to perform negative testing.

Also look for the ability to share subscriber data across application interfaces. Since you'll most likely be emulating multiple network elements that are simultaneously attached to the PCRF, those network elements should all be working with the same pool of subscribers. This can save countless hours of configuration.

◆ **Expansion Capability**

Find an emulator that will continue to meet your needs as your requirements change. An emulator that minimizes its dependency on custom hardware will be better positioned to keep up with [Moore's Law](#) over the long term.

Other Considerations

Although we will not spend any time discussing another methodology, it is worth noting that it is also possible to create a hybrid test environment in which a test network using real NEs is supplemented with emulators that emulate additional NEs. By using a combination of real and emulated NEs, the operator can create a baseline level of load while also using real network equipment for certain aspects of the test.

Summary

A key component of building a fast and reliable wireless network includes extensive testing of the NEs that make up the network. Multiple testing methodologies exist and using multiple methodologies will provide the best possible coverage. We've focused on the value of testing NEs in isolation, but it's vital to include multiple methodologies in any comprehensive test plan.

Selecting full featured test equipment will enhance test coverage and ensure the service provider that it is getting the needed performance, capacity, features, and stability in the equipment that it deploys in its network.



Developing Solutions, Inc.

1801 West Louisiana Street
Suite 200
McKinney, TX 75069

Sales & Information

877-233-8350
info@developingsolutions.com
www.developingsolutions.com