TS-007
Reporting

Abstract: This document defines a standard for expressing test report data.

Authors: Haj Elfadil (Juniper), Marcus Hines (Google), Eric Miller (Spirent), Elemér Lelik (Ericsson), Dan Diolosa (Spirent)

Status: Release

Revision: 1

Revision date December 2015

Submission: ntaf TS-007

Revision History

Version	Date	By	Changes
1	2015-12-01	Technical Committee	First Release

Background and Motivation

The push for a standardized form of reporting test and automation results is gaining momentum. More and more service providers desire that equipment manufactures provide standardized test results to validate what they are selling is up to required standards. At the same time users of test equipment look for some standardization across the many types of data and report formats provided.

In order to communicate the test and automation results in a common language among the equipment vendors, the tools vendors, the service providers and the enterprises, NTAF reporting standard utilizes the well known industry JSON protocol as a format for the test result outputs.

Objective

The purpose for this specification is to provide the standardized way to generate the automation test output, and to provide the details about the requirement needed to generate the reporting data. The goal is to converge on a common way for systems, devices and tools to generate a test data output that is NTAF compliant.

Report Format

Overview

Reports are in JSON. It is possible to express the content defined by this document in other formats (such as XML or YAML) and such expressions may be defined in future documents. However this specification only recognizes JSON. The elements of the report correspond to JSON data structures, either objects or arrays. At the time of this document's publication, information about JSON structure and its formal definition are available at http://json.org.

The report itself is a JSON object. Other elements of the report are contained in this master "test report object". A test report object is a collection of data resulting from a test execution. The collected data is a set of elements which describe test setup, test execution, and results of test execution.

The test report can contain the following elements:

- Vendors
- Test Suite
- Test Cases
- Test Steps
- Topologies
- Parameters
- Logs
- Results

Elements of a report can contain other elements. For example, test suites and test cases can contain topologies. There is no restriction on the levels of nested elements.

Per standard JSON convention, the order of elements contained within other elements is arbitrary. Arrays are used when order is required, such as for listing test steps.

Because of its expansive nature and application in other areas, test topology is described in a separate document, TS-006. The test report and other related elements are described in the following sections.

Test Report

A test report is an object which can contain the following attributes. All but name are optional. Each attribute is given by name-value pairs which are members of the report object. These attributes contain overview data for a report. The following attributes are recognized by this specification.

In this table and those that follow the data type refers to how the data is interpreted. The element type is the JSON element type for the data. Suggested values, when provided, give NTAF recognized values for a attribute.

Name	Data Type	Element type	Description	Suggested Values
name	string	string	Name of report	
createdDate	date-time	string	The date (and possible time) the report was created.	
description	string	string	A human readable description of the report	
user	string	string	Person who created this report.	
plan	uri	string	Link to test plan object	

Vendors

The vendors element is a name-value pair whose name is “vendors” and whose value is an array of vendor objects. The ordering of the vendor objects has no significance. Each vendor object will usually correspond to a topology node, although this is not a strict requirement. Usually a vendor object refers to a single device, but it is possible for it to refer to an entire system when desired. Vendor objects can be test tools, devices under test (DUT), systems under test (SUT), or any other relevant device or system used in a test. The element is named “vendors” because the data is considered specific to setup of equipment from specific vendors.

The following attributes are recognized by this specification for use on vendor objects. All but name are optional.

Name	Data Type	Element Type	Description	Suggested Values
name	string	string	Name of device. Should be the same as the name used in associated topology.	
dutFeatures	string	array	Features of this node which were tested. Blank if this node is not a DUT.	
externalApi	string	string	Human readable description of external and third party APIs and libraries used.	
functionalClass	string	array	The functional classification of the node (i.e. the functions it primarily supports). A single node can provide multiple values.	Examples: router, ethernetSwitch, server,

				trafficGenerator. See Table 4 in TS-004 for complete list.
internalApi	string	string	Human readable description of internal APIs and libraries used.	
modelName	string	string	Model name of the node (human readable).	
physicalClass	string	string	The physical classification of the resource (i.e. its physical form). Each resource provides the single value which most closely matches.	Examples: chassis, handheld, rackMountable,, virtual. See Table 3 in TS-004 for complete list.
productFamily	string	string	Manufacturer supplied classification of the type of resource.	
protocols	string	array	Protocols enabled for this test	
testConfiguration	string	array	Human readable summary of configuration commands applied to vendor node for this test.	
versionSoftware	string	string	Identifies the version of the installed software of the resource (i.e. overall version).	
vendorName	string	string	Name of vendor	

Test Suite

The test suite element is a name-value pair whose name is “testSuites”. Its value is an array of test suite objects. Each test suite object describes a collection of test cases that were used in the test. The following attributes can be included on the test suite object. All but name are optional.

Name	Data Type	Element Type	Description	Suggested Values
name	string	string	Name of test suite	
description	string	string	A human readable description of the test suite.	

Test Cases

The test cases element is a name-value pair whose name is “testCases” and whose value is an array of test case objects. Each test case object describes one test case used in a test. The

ordering of the test case objects matches the order in which they were executed. Test case objects have the following test case attributes that are recognized by this specification. All but name are optional.

Name	Data Type	Element Type	Description	Suggested Values
name	string	string	Test case name	
uuid	string	string	Test case unique identifier	
purpose	string	string	Human readable description of the test case purpose	
author	string	string	Person who created test case	
description	string	string	Human readable detailed description of test case.	
protocols	string	array	Protocols enabled for this test	

In addition to the above attributes, test cases may have all the attributes listed below for test steps, except the test step name. In programming terms a test case may be thought of as a child class of a test step. Conceptually both test cases and test steps are very similar in that they both represent steps or stages of a test. A test case is distinguished as being a larger unit which has additional high level descriptive attributes as listed above.

Unlike most report objects the location of test case objects and test step objects is restricted. Test case objects must only be children of the main report object. Test step objects must only be children of test cases or other test steps.

Test Step

The test steps element is a name-value pair whose name is “testSteps” and whose value is an array of test step objects. Each test step object describes one test step. The ordering of the test step objects matches the order in which they were executed.Test step objects have the following test step attributes that are recognized by this specification. All step attributes except name are optional.

Name	Data Type	Element Type	Description	Suggested Values
name	string	string	Test step name	
startTime	date-time	string	Start time of test run	
endTime	date-time	string	End time of test run	
duration	float	number	Duration of test run in milliseconds.	
totalTime	string	string	Human readable duration.	

externalApi	string	string	Human readable description of external and third party APIs and libraries used.	
internalApi	string	string	Human readable description of internal APIs and libraries used.	
traffic	string	array	Human readable information about traffic used in this test. For example, L2-L7, percent of bandwidth.	

Topologies

Topologies are given in a name value pair named “topologies”. The value is an array which contains one or more topology items. Each topology item may be either a topology object or a string which describes a link to a file which contains a topology object. Topology objects are described in TS-006. Topologies may be associated with a report, test suite, or test case.

Parameters

Parameter objects are a name-value pair whose name is “parameters” and whose value is an object. Parameters are elements of the master parameters object, and thus are name-value pairs for which the value may be any type of data. Common values will be strings or numbers. However, values also can be objects or arrays. Thus nesting of parameter is supported to an arbitrary depth.

This standard places no definition nor restriction on what parameters and values are allowed. Such may be defined in later documents but for now users of this format are free to use whatever names and values they wish.

Logs

Log objects are a name-value pair whose name is “logs” and whose value is an array. Each element of the array is a string which points to the location of log files on a system. Because of its length, log content is not included in a report.

Logs contain detailed test run data. Whereas results contain overview data, logs contain detailed traces of test activity. Logs are used primarily for debugging or in-depth analysis of test runs. While traditionally logs files contain code traces, this specification makes no restriction on the content of such files so they might, for example, include tables, charts, pcap files, or databases.

This document does not restrict how or what produces logs. However, to help understanding it is noted that logs are often produced by the following.

- Test automation frameworks
- Device (or system) under test logging systems
- Other devices associated with a test
- Any scripts or other code run during the course of a test
- Test tool hardware systems

- Test tool control software.

Results

Results are expressed by the attributes in the table below. These result attributes may be added to reports, test suites, test cases, and test steps. All result attributes are optional.

Name	Data Type	Element Type	Description	Suggested Values
stepsFailed	integer	number	Number of steps that failed	
stepsPassed	integer	number	Number of steps that passed	
stepsSkipped	integer	number	Number of steps that were skipped	
stepsTaken	string	array	Human readable summary of each step executed.	
status	string	string	Overall result of associated test.	pass, fail, abort, error, blocked, unknown
testExpression	string	string	Mathematical (logical) test expression used for this test	
expectedResult	string	string	Expected result of test expression	
actualResult	string	string	Actual result of test expression	
executionIssues	string or object	array	High level messages or issues generated by test	

Schema

Reports can be validated against the following JSON schema. All use cases as well as this schema have been validated against JSONLint (http://jsonlint.com/) and JSON Schema Validator (http://jsonschemalint.com/draft4/).

```
{
  "$schema":"http://json-schema.org/draft-04/schema",
  "id":"http://ntaforum.org/2015/reporting",
  "title":"JSON Schema for NTAF reporting description",
  "description":"The JSON schema for NTAF reporting",
  "oneOf":[
    {
      "$ref":"#/definitions/TestReport"
    }
  ],
  "definitions":{
```

```
    "TestReport":{
      "type":"object",
      "allOf":[
        {
          "properties":{
            "name":{
              "type":"string",
              "description":"Name of report"
            },
            "createdDate":{
              "type":"string",
              "format":"date-time",
              "description":"The date (and possible time) the report was created."
            },
            "description":{
              "type":"string",
              "description":"A human readable description of the report"
            },
            "user":{
              "type":"string",
              "description":"Person who created this report."
            },
            "plan":{
              "type":"string",
              "format":"uri",
              "description":"Link to test plan object"
            }
          },
          "required":[
            "name"
          ]
        },
        {
          "$ref":"#/definitions/SubObjects"
        },
        {
          "$ref":"#/definitions/Result"
        }
      ]
    },
    "Vendor":{
      "type":"object",
      "allOf":[
        {
          "properties":{
            "name":{
              "type":"string",
              "description":"Name of device. Should be the same as the name used in associated
topology."
            },
            "dutFeatures":{
              "type":"array",
              "description":"Features of this node which were tested. Blank if this node is
not a DUT.",
              "items":{
                "type":"string"
              },
              "minItems":0,
              "uniqueItems":true
            },
            "externalApi":{
              "type":"string",
              "description":"Human readable description of external and third party APIs and
libraries used."
            },
            "functionalClass":{
```

```
              "type":"array",
              "description":"The functional classification of the resource (i.e. the functions
it primarily supports). A single resource can provide multiple values.",
              "items":{
                "$ref":"#/definitions/FunctionalClass"
              },
              "minItems":0,
              "uniqueItems":true
            },
            "internalApi":{
              "type":"string",
              "description":"Human readable description of internal APIs and libraries used."
            },
            "modelName":{
              "type":"string",
              "description":"Model name of the node (human readable)."
            },
            "physicalClass":{
              "$ref":"#/definitions/PhysicalClass",
              "description":"The physical classification of the resource (i.e. its physical
form). Each resource provides the single value which most closely matches."
            },
            "productFamily":{
              "type":"string",
              "description":"Manufacturer supplied classification of the type of resource."
            },
            "protocols":{
              "type":"array",
              "description":"Protocols enabled for this test",
              "items":{
                "type":"string"
              },
              "minItems":0,
              "uniqueItems":true
            },
            "testConfiguration":{
              "type":"array",
              "description":"Human readable summary of configuration commands applied to
vendor node for this test.",
              "items":{
                "type":"string"
              },
              "minItems":0
            },
            "versionSoftware":{
              "type":"string",
              "description":"Identifies the version of the installed software of the resource
(i.e. overall version)."
            },
            "vendorName":{
              "type":"string",
              "description":"Name of vendor"
            },
            "parameters":{
              "$ref":"#/definitions/Parameter"
            },
            "logs":{
              "type":"array",
              "items":{
                "$ref":"#/definitions/Log"
              },
              "minItems":0,
              "uniqueItems":true
            }
          },
          "required":[
```

```
        "name"
      ]
    },
    {
      "$ref":"#/definitions/SubObjects"
    }
  ]
},
"TestSuite":{
  "type":"object",
  "allOf":[
    {
      "properties":{
        "name":{
          "type":"string",
          "description":"Name of test suite"
        },
        "description":{
          "type":"string",
          "description":"A human readable description of the test suite."
        }
      },
      "required":[
        "name"
      ]
    },
    {
      "$ref":"#/definitions/SubObjects"
    },
    {
      "$ref":"#/definitions/Result"
    }
  ]
},
"TestCase":{
  "type":"object",
  "allOf":[
    {
      "properties":{
        "uuid":{
          "type":"string",
          "description":"Test case unique identifier"
        },
        "purpose":{
          "type":"string",
          "description":"Human readable description of the test case purpose"
        },
        "author":{
          "type":"string",
          "description":"Person who created test case"
        },
        "description":{
          "type":"string",
          "description":"Human readable detailed description of test case."
        },
        "protocols":{
          "type":"array",
          "description":"Protocols enabled for this test",
          "items":{
            "type":"string"
          },
          "minItems":0,
          "uniqueItems":true
        }
      },
      "required":[
```

```
            "name"
          ]
        },
        {
          "$ref":"#/definitions/TestStep"
        }
      ]
    },
    "TestStep":{
      "type":"object",
      "allOf":[
        {
          "properties":{
            "name":{
              "type":"string",
              "description":"Test name"
            },
            "startTime":{
              "type":"string",
              "format":"date-time",
              "description":"Start time of test run"
            },
            "endTime":{
              "type":"string",
              "format":"date-time",
              "description":"End time of test run"
            },
            "duration":{
              "type":"number",
              "description":"Duration of test run in milliseconds."
            },
            "totalTime":{
              "type":"string",
              "description":"Human readable duration."
            },
            "externalApi":{
              "type":"string",
              "description":"Human readable description of external and third party APIs and
libraries used."
            },
            "internalApi":{
              "type":"string",
              "description":"Human readable description of internal APIs and libraries used."
            },
            "traffic":{
              "type":"array",
              "description":"Human readable information about traffic used in this test. For
example, L2-L7, percent of bandwidth",
              "items":{
                "type":"string"
              },
              "minItems":0,
              "uniqueItems":true
            }
          },
          "required":[
            "name"
          ]
        },
        {
          "$ref":"#/definitions/SubObjects"
        },
        {
          "$ref":"#/definitions/Result"
        }
      ]
```

```
},
"Topology":{
  "oneOf":[
    {
      "type":"string",
      "format":"uri"
    },
    {
      "$ref":"http://ntaforum.org/2015/topology"
    }
  ]
},
"Parameter":{
  "type":"object",
  "properties":{

  },
  "additionalProperties":true
},
"Log":{
  "type":"string",
  "format":"uri"
},
"FunctionalClass":{
  "anyOf": [
    {"enum":[
      "accessPoint",
      "bladeSystem",
      "bridge",
      "ethernetSwitch",
      "forwardingDevice",
      "gateway",
      "hypervisor",
      "matrixSwitch",
      "monitor",
      "patchPanel",
      "power",
      "protocolEmulator",
      "printer",
      "router",
      "server",
      "storage",
      "testEquipment",
      "trafficGenerator",
      "transceiver"
    ]},
    {"type": "string"}
  ]
},
"PhysicalClass":{
  "anyOf": [
    {"enum":[
      "appliance",
      "bay",
      "blade",
      "card",
      "chassis",
      "component",
      "connector",
      "cpe",
      "desktop",
      "handheld",
      "laptop",
      "module",
      "patchPanel",
      "phone",
```

```
      "printer",
      "rackMountable",
      "screen",
      "server",
      "software",
      "tablet"
    ]},
    {"type": "string"}
  ]
},
"Status":{
  "anyOf": [
    {"enum":[
      "pass",
      "fail",
      "abort",
      "error",
      "blocked",
      "unknown"
    ]},
    {"type": "string"}
  ]
},
"Result":{
  "type":"object",
  "properties":{
    "stepsFailed":{
      "type":"number",
      "description":"Number of steps that failed"
    },
    "stepsPassed":{
      "type":"number",
      "description":"Number of steps that passed"
    },
    "stepsSkipped":{
      "type":"number",
      "description":"Number of steps that were skipped"
    },
    "stepsTaken":{
      "type":"array",
      "description":"Human readable summary of each step executed.",
      "items":{
        "type":"string"
      },
      "minItems":0
    },
    "status":{
      "type":"string",
      "description":"Overall result of associated test.",
      "$ref":"#/definitions/Status"
    },
    "testExpression":{
      "type":"string",
      "description":"Mathematical (logical) test expression used for this test"
    },
    "expectedResult":{
      "type":"string",
      "description":"Expected result of test expression"
    },
    "actualResult":{
      "type":"string",
      "description":"Actual result of test expression"
    },
    "executionIssues":{
      "type":"array",
      "description":"High level messages or issues generated by test",
```

```
      "items":{
        "$ref":"#/definitions/ExecutionIssue"
      },
      "minItems":0
    }
  }
},
"ExecutionIssue":{
  "oneOf":[
    {
      "type":"string"
    },
    {
      "type":"object",
      "properties":{

      },
      "additionalProperties":true
    }
  ]
},
"SubObjects":{
  "type":"object",
  "description":"Object properties to allow inclusion in other objects.",
  "properties":{
    "vendors":{
      "type":"array",
      "items":{
        "$ref":"#/definitions/Vendor"
      },
      "minItems":0
    },
    "testSuites":{
      "type":"array",
      "items":{
        "$ref":"#/definitions/TestSuite"
      },
      "minItems":0
    },
    "testCases":{
      "type":"array",
      "items":{
        "$ref":"#/definitions/TestCase"
      },
      "minItems":0
    },
    "topologies":{
      "type":"array",
      "items":{
        "$ref":"#/definitions/Topology"
      },
      "minItems":0
    },
    "parameters":{
      "$ref":"#/definitions/Parameter"
    },
    "logs":{
      "type":"array",
      "items":{
        "$ref":"#/definitions/Log"
      },
      "minItems":0,
      "uniqueItems":true
    }
  }
}
```

```
    }
}
```

Example Use Cases

BGP Test for Router

This BGP use case tests the BGP scaling capabilities for a router. Conceptually the type of network being tested looks like Figure 1 below. The router being tested (r0) should properly discover the reported external routes from the network on the left. Once discovered the networks on the right should properly send traffic to those locations.

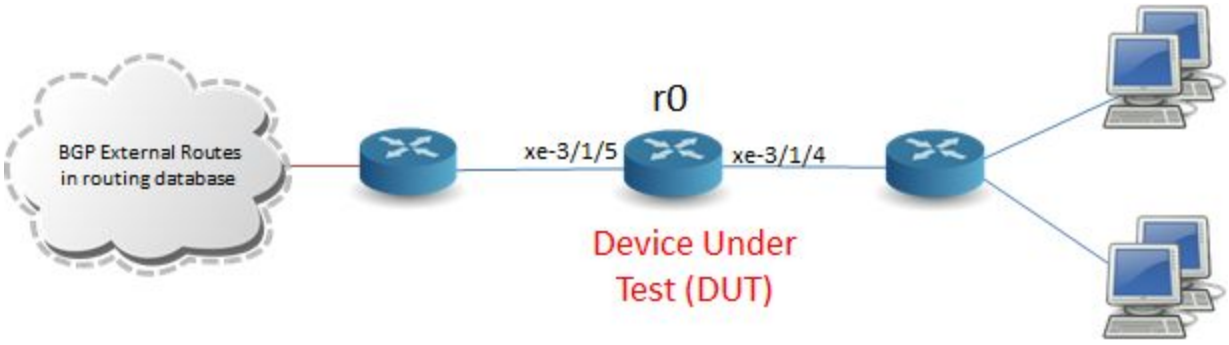


Figure 1: BGP Test Conceptual Network

To run the test a test tool (rt0) is used to emulate the parts of the network outside of the DUT. This is pictured below in Figure 2.

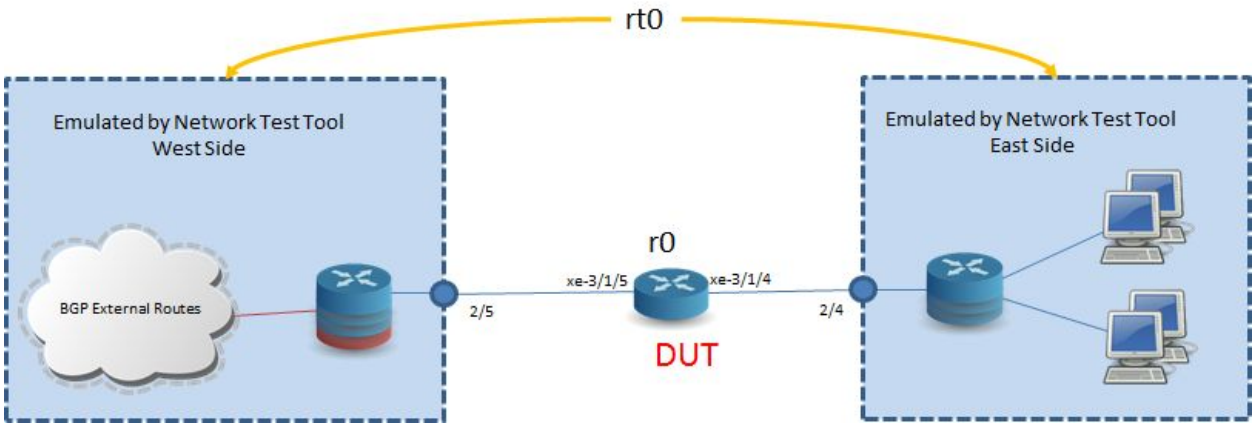


Figure 2: BGP Test Test Tool Emulated Network

From the emulated network above it is apparent that two test ports are needs on the test tool with each test port connected to a router interface. This leads to the hardware test topology pictured in Figure 3.

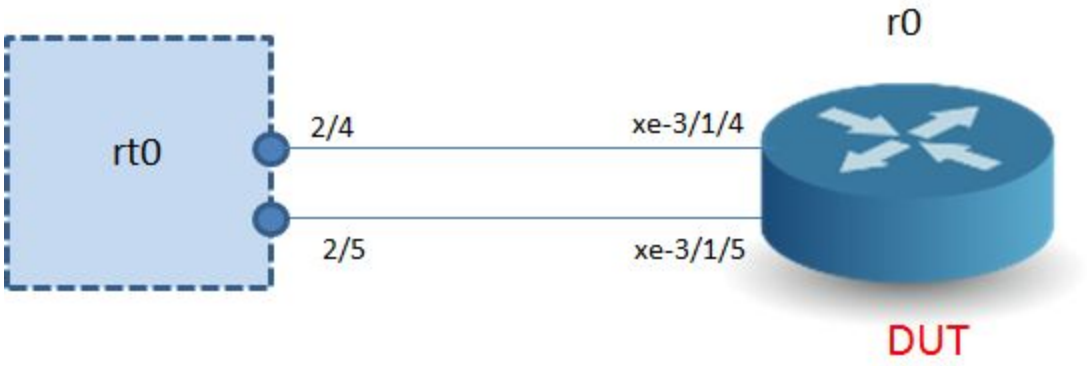


Figure 3: BGP Test Hardware Network

The following is the report associated with this test.


```
{
  "name":"BGP Scaling",
  "createdDate":"2014-11-11T18:25:43.511Z",
  "description":"Test BGP scaling for NEM Vendor router using Test Tool Vendor protocol
emulator.",
  "user":"Robin A. Tester",
  "plan":"https://nem.vendor.com/testPlans/router/v2.5/BGP/ScalingTestPlan.html",
  "topologies":[
    {
      "name":"topo1",
      "description":" Router and Router Tester ",
      "nodes":[
        {
          "name":"r0",
          "interfaces":[
            {
              "name":"xe3/1/4",
              "destination":"rt0:2/4"
            },
            {
              "name":"xe3/1/5",
              "destination":"rt0:2/5"
            }
          ],
          "modelName":"mx2020",
          "versionSoftware":"14.1"
        },
        {
          "name":"rt0",
          "interfaces":[
            {
              "name":"2/4",
              "destination":"r0:xe3/1/4"
            },
            {
              "name":"2/5",
              "destination":"r0:xe3/1/5"
            }
          ],
          "modelName":"Protocol Emulater 3000",
          "versionSoftware":"6.50"
        }
      ]
    }
  ],
```

```
"vendors":[
  {
    "name":"r0",
    "vendorName":"NEM Vendor",
    "parameters":{
      "global_params":"TC",
      "user_variables":{
        "commitforcethreshold":10000,
        "commitlogthreshold":300,
        "context":[
          "default:default"
        ],
        "daemon_logging":1,
        "duration":4,
        "if_unit_base":1,
        "jsoninspectformat":2,
        "jsonkeyformat":"lc,_",
        "jsonreformattopology":1,
        "jsonreportfileext":"ntafbgpreport.json",
        "jsonreportversion":"2.1",
        "jsonsectiondata":"Data",
        "jsonsectioninfo":"Vendors>RouterVendor",
        "jsonsectiontestcase":"Test Cases",
        "jsonsectiontopology":"Topologies>Topo1",
        "lockinterval":60,
        "locktimeout":7200,
        "r0_features":[
          "rip",
          "isis",
          "ospf",
          "multicast",
          "bgp",
          "gre",
          "gres",
          "vrrp",
          "ipv6",
          "gres",
          "vpn",
          "gmpls",
          "ccc",
          "l2.5vpn",
          "l2ckt",
          "l2vpn",
          "l3vpn",
          "ldp",
          "rsvp",
          "mpls",
          "tcc",
          "vpls"
        ],
        "r0_loop_if":"lo0.0",
        "r0_machine":"static",
        "r0_make":"RouterVendor",
        "r0_mgt_ip":"1.1.1.1/32",
        "r0_mgt_ip_addr":"1.1.1.1",
        "r0_mgt_ip_len":32,
        "r0_model":"MX2020",
        "r0_name":"??",
        "r0_name_isis":"??",
        "r0_os":"JunOS",
        "r0_r0_rt0_1_if":"xe3/1/4",
        "r0_r0_rt0_1_if_name":"xe3/1/4.0",
        "r0_r0_rt0_2_if":"xe3/1/5",
        "r0_r0_rt0_2_if_name":"xe3/1/5.0",
        "r0_x_os_version":"14.1",
        "rc_source_address":2,
```

```
        "rc_timeout":3,
        "rs_dictionary":1
      },
      "x_tags":{
        "x_cmd_log_level":"trace",
        "x_context":"nsr",
        "x_descr_long":"THIS TEST SETUP IS DESIGNED IS FOR NTAF TEST SAMPLE",
        "x_set_user_configs":[
          "&::TestConfig"
        ],
        "x_set_user_configs_append":[
          "&::TestConfig"
        ]
      }
    },
    "testConfiguration":[
      "04:11:23 (00:00) [INFO] _IniIfParams(): R0 'R0RT01'keywords: L1 ether xe [xe3/1/4]",
      "04:11:23 (00:00) [INFO] _IniIfParams(): R0 'R0RT02' keywords: L1 ether xe[xe3/1/5]",
      "04:13:20 (01:57) [INFO] TestConfig(): Setting myVar links for RT0"
    ]
  },
  {
    "name":"rt0",
    "vendorName":"Test Tool Vendor",
    "parameters":{
      "user_variables":{
        "machine":"static",
        "make":"TestVendor",
        "mgt_ip":"2.2.2.1/23",
        "mgt_ip_addr":"2.2.2.1",
        "mgt_ip_len":23,
        "mgt_ipv6":"abcd::2:2:2:1/128",
        "mgt_ipv6_addr":"abcd::2:2:2:1",
        "mgt_ipv6_len":128,
        "model":"rt",
        "r0_rt0_1_if":"2/4",
        "r0_rt0_1_if_name":"2/4",
        "r0_rt0_2_if":"2/5",
        "r0_rt0_2_if_name":"2/5",
        "x_os_version":"6.50",
        "x_tclserver":"2.2.2.2",
        "pppoa_open_rate":4,
        "pppoa_opening_window":4,
        "pppoa_retransmit_attempts":1,
        "pppoa_retransmit_timeout":5,
        "pppoe_open_rate":4,
        "pppoe_opening_window":4,
        "pppoe_retransmit_attempts":1,
        "pppoe_retransmit_timeout":5,
        "pppox_open_rate":4,
        "pppox_opening_window":4,
        "pppox_retransmit_attempts":1,
        "pppox_retransmit_timeout":5,
        "stream":3,
        "tolerance":5,
        "save_daemon_log":1
      }
    },
    "testConfiguration":[
      "04:13:20 (01:57) [INFO] TvNetConnect(): Initializing $t>{myVar}",
      "04:13:20 (01:57) [INFO] TestConfig(): Setting myVar links for RT0"
    ]
  }
],
"testSuites":[
  {
```

```
      "name":"BPG Scaling",
      "topologies":[
        "https://nem.vendor.com/topologies/testLab23.json"
      ],
      "description":"Test suite containing router BGP scaling test cases.",
      "parameters":{
        "testbed":"rack1"
      }
    }
  ],
  "testCases":[
    {
      "name":"BGP BGPScaling: 10000 Routes",
      "parameters":{
        "arguments":{
          "adv_route_ip":"1.1.1.1",
          "bgp_cfg":"BGP",
          "dut":"R0",
          "first_as":655555,
          "intf_rt_cfg":"R0:2:125:dual",
          "loopback_ip":"1.1.1.1/32",
          "loopback_ipv6":"ABCD::1:1:1:1/128",
          "num_active_routes":10000,
          "num_bgp_peers":100,
          "num_routes":10000,
          "testipv4net":"1.1.1.1/24",
          "testipv6net":"ABCD:0001:0001:0001:0001:0001/128",
          "testnetcount":100
        },
        "duration":685,
        "duration_unit":"seconds",
        "errors":1,
        "params":{

        },
        "name":"XTESTCASE001",
        "parameters":{
          "x_my_chap_num":6,
          "x_my_chapter_id":"1.1",
          "x_my_md5":"",
          "x_my_platform":"R0",
          "x_my_tc_num":1,
          "x_my_title":"Bgp TEST",
          "x_my_topology":"Topo1"
        },
        "platform":"MX2020",
        "polling_interval":1,
        "profile":{
          "x_args":{
            "bgp_cfg":"BGP_CONFIG",
            "intf_rt_cfg":"R0:2:125:dual"
          },
          "x_cmd":"&::tc_bgpTest",
          "x_descr":"BGP Testing BGP TO CREATE 10000 ROUTES WITH 100 PEERS",
          "x_descr_long":"Procedure:  Inject 10000 routes .  Check the route Number .  Clear
bgp session and close.",
          "x_is_run":true,
          "x_loop":1,
          "x_reset_commit":true,
          "x_type":"functional"
        }
      },
      "status":"fail",
      "testSteps":[
        {
          "name":"Step 1: Configuration Initialization",
```

```
          "duration":307,
          "testSteps":[
            {
              "name":"Step 1.1: Assigning some IP Addresses to the Test Network.",
              "duration":3,
              "executionIssues":[
                "04:15:59 (00:02) [INFO] tc_bgpTest(): Using Network Address for testing
between the peers"
              ],
              "status":"pass",
              "totalTime":"[00:18]",
              "endTime":"2014-11-11T18:25:43.511Z"
            },
            {
              "name":"Step 1.2: Assiging Ipv6 networks to the router and the Test Tool.",
              "duration":4,
              "executionIssues":[
                "04:16:03 (00:04) [INFO] tc_bgpTest(): Using Network Address for testing
between the peers"
              ],
              "status":"pass",
              "totalTime":"[00:20]",
              "endTime":"2014-11-11T18:25:43.511Z"
            },
            {
              "name":"Step 1.3: Configure the Router loopback",
              "duration":12,
              "executionIssues":[
                {
                  "04:16:06 Committing 4 line(s) of pending config commands on R0 (DUT)...":[
                    "set interfaces lo0 unit 0 family inet address 1.1.1.1/32 ",
                    "set interfaces lo0 unit 0 family inet6 address ABCD::1:1:1:1/128 ",
                    "set routingoptions autonomoussystem 255554",
                    "set routingoptions routerid 1.1.1.1"
                  ]
                }
              ],
              "status":"pass",
              "totalTime":"[00:24]",
              "endTime":"2014-11-11T18:25:43.511Z"
            },
            {
              "name":"Step 1.4: Configure Test Tool Interface.",
              "duration":264,
              "executionIssues":[
                "04:16:18 (00:03) [INFO] tc_bgpTest(): Configuring Test Tool interfaces"
              ],
              "status":"pass",
              "totalTime":"[00:36]",
              "endTime":"2014-11-11T18:25:43.511Z"
            },
            {
              "name":"Step 1.5: Configure BGP groups.",
              "duration":21,
              "executionIssues":[
                "04:20:44 (00:05) [INFO] configBgp(): Create BGP GROUPS DONE",
                {
                  "04:20:44 Committing 5 line(s) of pending config commands on R0 ...":[
                    "set policyoptions policystatement noroutesfilter term default then
reject",
                    "set protocols bgp group session1 type internal",
                    "set protocols bgp group session1 localaddress 1.1.1.1",
                    "set protocols bgp group session1 export noroutesfilter",
                    "set protocols bgp group session1 family inet any"
                  ]
                },
```

```
                {
                  "04:20:53 Committing 2 line(s) of pending config commands on R0 ...":[
                    "set protocols bgp group session1 neighbor 192.168.2.2",
                    "set protocols bgp group session1 neighbor 192.168.3.2"
                  ]
                }
              ],
              "status":"pass",
              "totalTime":"[05:00]",
              "endTime":"2014-11-11T18:25:43.511Z"
            }
          ],
          "status":"pass",
          "totalTime":"[00:14]",
          "endTime":"2014-11-11T18:25:43.511Z"
        },
        {
          "name":"Step 2: Test: Start BGP emulations",
          "duration":190,
          "testSteps":[
            {
              "name":"Step 2.1: Start BGP emulations for each group ",
              "duration":6,
              "executionIssues":[

              ],
              "status":"pass",
              "totalTime":"[05:24]",
              "endTime":"2014-11-11T18:25:43.511Z"
            },
            {
              "name":"Step 2.2: Verify BGP Neighbors.",
              "duration":182,
              "executionIssues":[
                {
                  "04:21:10 Verify(): Verify all bgp neighbors are up and all routes were
learned.":{
                    "duration":180,
                    "executionIssues":[
                      "04:21:11 (00:01) [INFO] Verify() at ntafbgptest. pm:109: Retry #1/180
(already past next 1 sec interval)",
                      "04:21:11 (00:01) [INFO] Verify() at ntafbgptest. pm:109: Retry #2/180
in 1 second...",
                      "04:21:12 (00:02) [INFO] Verify() at ntafbgptest.pm:109: Retry#3/180 in
1 second...",
                      "04:21:13 (00:03) [INFO] Verify() at ntafbgptest.pm:109: Retry#4/180 in
1 second...",
                      "04:21:14 (00:04) [INFO] Verify() at ntafbgptest.pm:109: Retry#5/180 in
1 second...",
                      "04:21:15 (00:05) [INFO] Verify() at ntafbgptest.pm:109: Retry#6/180 in
1 second...",
                      "04:21:16 (00:06) [INFO] Verify() at ntafbgptest.pm:109: Retry#7/180 in
1 second...",
                      "04:21:18 (00:08) [INFO] Verify() at ntafbgptest.pm:109: Retry#8/180
(already past next 1 sec interval)",
                      "04:21:18 (00:08) [INFO] Verify() at ntafbgptest.pm:109: Retry#9/180 in
1 second...",
                      "04:21:19 (00:09) [INFO] Verify() at ntafbgptest.pm:109: Retry#10/180 in
1 second...",
                      "04:21:20 (00:10) [INFO] Verify() at ntafbgptest.pm:109: Retry#11/180 in
1 second...",
                      "04:21:21 (00:11) [INFO] Verify() at ntafbgptest.pm:109: Retry#12/180 in
1 second...",
                      "04:21:22 (00:12) [INFO] Verify() at ntafbgptest.pm:109: Retry#13/180 in
1 second...",
                      "04:21:23 (00:13) [INFO] Verify() at ntafbgptest.pm:109: Retry#14/180 in
```

```
1 second...",
                    "04:21:24 (00:14) [INFO] Verify() at ntafbgptest.pm:109: Retry#15/180 in
1 second...",
                    "04:21:25 (00:15) [INFO] Verify() at ntafbgptest.pm:109: Retry#16/180 in
1 second...",
                    "04:21:27 (00:17) [INFO] Verify() at ntafbgptest.pm:109: Retry#17/180
(already past next 1 sec interval)",
                    "04:21:27 (00:17) [INFO] Verify() at ntafbgptest.pm:109: Retry#18/180 in
1 second...",
                    "04:21:28 (00:18) [INFO] Verify() at ntafbgptest.pm:109: Retry#19/180 in
1 second...",
                    "04:21:29 (00:19) [INFO] Verify() at ntafbgptest.pm:109: Retry#20/180 in
1 second...",
                    "04:21:30 (00:20) [INFO] Verify() at ntafbgptest.pm:109: Retry#21/180 in
1 second...",
                    "04:21:31 (00:21) [INFO] Verify() at ntafbgptest.pm:109: Retry#22/180 in
1 second...",
                    "04:21:32 (00:22) [INFO] Verify() at ntafbgptest.pm:109: Retry#23/180 in
1 second...",
                    "04:21:33 (00:23) [INFO] Verify() at ntafbgptest.pm:109: Retry#24/180 in
1 second...",
                    "04:21:34 (00:24) [INFO] Verify() at ntafbgptest.pm:109: Retry#25/180 in
1 second...",
                    "04:21:35 (00:25) [INFO] Verify() at ntafbgptest.pm:109: Retry#26/180 in
1 second...",
                    "04:21:37 (00:27) [INFO] Verify() at ntafbgptest.pm:109: Retry#27/180
(already past next 1 sec interval)",
                    "04:21:37 (00:27) [INFO] Verify() at ntafbgptest.pm:109: Retry#28/180 in
1 second...",
                    "04:21:38 (00:28) [INFO] Verify() at ntafbgptest.pm:109: Retry#29/180 in
1 second...",
                    "04:21:39 (00:29) [INFO] Verify() at ntafbgptest.pm:109: Retry#30/180 in
1 second...",
                    "04:21:40 (00:30) [INFO] Verify() at ntafbgptest.pm:109: Retry#31/180 in
1 second...",
                    "04:21:41 (00:31) [INFO] Verify() at ntafbgptest.pm:109: Retry#32/180 in
1 second...",
                    "04:21:42 (00:32) [INFO] Verify() at ntafbgptest.pm:109: Retry#33/180 in
1 second...",
                    "04:21:43 (00:33) [INFO] Verify() at ntafbgptest.pm:109: Retry#34/180 in
1 second...",
                    "04:21:45 (00:35) [INFO] Verify() at ntafbgptest.pm:109: Retry#35/180
(already past next 1 sec interval)",
                    "04:21:45 (00:35) [INFO] Verify() at ntafbgptest.pm:109: Retry#36/180 in
1 second...",
                    "04:21:46 (00:36) [INFO] Verify() at ntafbgptest.pm:109: Retry#37/180 in
1 second...",
                    "04:21:47 (00:37) [INFO] Verify() at ntafbgptest.pm:109: Retry#38/180 in
1 second...",
                    "04:21:48 (00:38) [INFO] Verify() at ntafbgptest.pm:109: Retry#39/180 in
1 second...",
                    "04:21:49 (00:39) [INFO] Verify() at ntafbgptest.pm:109: Retry#40/180 in
1 second...",
                    "04:21:50 (00:40) [INFO] Verify() at ntafbgptest.pm:109: Retry#41/180 in
1 second...",
                    "04:21:51 (00:41) [INFO] Verify() at ntafbgptest.pm:109: Retry#42/180 in
1 second...",
                    "04:21:53 (00:43) [INFO] Verify() at ntafbgptest.pm:109: Retry#43/180
(already past next 1 sec interval)",
                    "04:21:53 (00:43) [INFO] Verify() at ntafbgptest.pm:109: Retry#44/180 in
1 second...",
                    "04:21:54 (00:44) [INFO] Verify() at ntafbgptest.pm:109: Retry#45/180 in
1 second...",
                    "04:21:55 (00:45) [INFO] Verify() at ntafbgptest.pm:109: Retry#46/180 in
1 second...",
                    "04:21:56 (00:46) [INFO] Verify() at ntafbgptest.pm:109: Retry#47/180 in
```

```
1 second...",
                    "04:21:57 (00:47) [INFO] Verify() at ntafbgptest.pm:109: Retry#48/180 in
1 second...",
                    "04:21:58 (00:48) [INFO] Verify() at ntafbgptest.pm:109: Retry#49/180 in
1 second...",
                    "04:21:59 (00:49) [INFO] Verify() at ntafbgptest.pm:109: Retry#50/180 in
1 second...",
                    "04:22:01 (00:51) [INFO] Verify() at ntafbgptest.pm:109: Retry#51/180
(already past next 1 sec interval)",
                    "04:22:01 (00:51) [INFO] Verify() at ntafbgptest.pm:109: Retry#52/180 in
1 second...",
                    "04:22:02 (00:52) [INFO] Verify() at ntafbgptest.pm:109: Retry#53/180 in
1 second...",
                    "04:22:03 (00:53) [INFO] Verify() at ntafbgptest.pm:109: Retry#54/180 in
1 second...",
                    "04:22:04 (00:54) [INFO] Verify() at ntafbgptest.pm:109: Retry#55/180 in
1 second...",
                    "04:22:05 (00:55) [INFO] Verify() at ntafbgptest.pm:109: Retry#56/180 in
1 second...",
                    "04:22:06 (00:56) [INFO] Verify() at ntafbgptest.pm:109: Retry#57/180 in
1 second...",
                    "04:22:07 (00:57) [INFO] Verify() at ntafbgptest.pm:109: Retry#58/180 in
1 second...",
                    "04:22:08 (00:58) [INFO] Verify() at ntafbgptest.pm:109: Retry#59/180 in
1 second...",
                    "04:22:09 (00:59) [INFO] Verify() at ntafbgptest.pm:109: Retry#60/180 in
1 second...",
                    "04:22:10 (01:00) [INFO] Verify() at ntafbgptest.pm:109: Retry#61/180 in
1 second...",
                    "04:22:12 (01:02) [INFO] Verify() at ntafbgptest.pm:109: Retry#62/180
(already past next 1 sec interval)",
                    "04:22:12 (01:02) [INFO] Verify() at ntafbgptest.pm:109: Retry#63/180 in
1 second...",
                    "04:22:13 (01:03) [INFO] Verify() at ntafbgptest.pm:109: Retry#64/180 in
1 second...",
                    "04:22:14 (01:04) [INFO] Verify() at ntafbgptest.pm:109: Retry#65/180 in
1 second...",
                    "04:22:15 (01:05) [INFO] Verify() at ntafbgptest.pm:109: Retry#66/180 in
1 second...",
                    "04:22:16 (01:06) [INFO] Verify() at ntafbgptest.pm:109: Retry#67/180 in
1 second...",
                    "04:22:17 (01:07) [INFO] Verify() at ntafbgptest.pm:109: Retry#68/180 in
1 second...",
                    "04:22:18 (01:08) [INFO] Verify() at ntafbgptest.pm:109: Retry#69/180 in
1 second...",
                    "04:22:19 (01:09) [INFO] Verify() at ntafbgptest.pm:109: Retry#70/180 in
1 second...",
                    "04:22:20 (01:10) [INFO] Verify() at ntafbgptest.pm:109: Retry#71/180 in
1 second...",
                    "04:22:21 (01:11) [INFO] Verify() at ntafbgptest.pm:109: Retry#72/180 in
1 second...",
                    "04:22:22 (01:12) [INFO] Verify() at ntafbgptest.pm:109: Retry#73/180 in
1 second...",
                    "04:22:24 (01:14) [INFO] Verify() at ntafbgptest.pm:109: Retry#74/180
(already past next 1 sec interval)",
                    "04:22:24 (01:14) [INFO] Verify() at ntafbgptest.pm:109: Retry#75/180 in
1 second...",
                    "04:22:25 (01:15) [INFO] Verify() at ntafbgptest.pm:109: Retry#76/180 in
1 second...",
                    "04:22:26 (01:16) [INFO] Verify() at ntafbgptest.pm:109: Retry#77/180 in
1 second...",
                    "04:22:27 (01:17) [INFO] Verify() at ntafbgptest.pm:109: Retry#78/180 in
1 second...",
                    "04:22:28 (01:18) [INFO] Verify() at ntafbgptest.pm:109: Retry#79/180 in
1 second...",
                    "04:22:29 (01:19) [INFO] Verify() at ntafbgptest.pm:109: Retry#80/180 in
```

```
1 second...",
                      "04:22:31 (01:21) [INFO] Verify() at ntafbgptest.pm:109: Retry#81/180
(already past next 1 sec interval)",
                      "04:22:31 (01:21) [INFO] Verify() at ntafbgptest.pm:109: Retry#82/180 in
1 second...",
                      "04:22:32 (01:22) [INFO] Verify() at ntafbgptest.pm:109: Retry#83/180 in
1 second...",
                      "04:22:33 (01:23) [INFO] Verify() at ntafbgptest.pm:109: Retry#84/180 in
1 second...",
                      "04:22:34 (01:24) [INFO] Verify() at ntafbgptest.pm:109: Retry#85/180 in
1 second...",
                      "04:22:35 (01:25) [INFO] Verify() at ntafbgptest.pm:109: Retry#86/180 in
1 second...",
                      "04:22:36 (01:26) [INFO] Verify() at ntafbgptest.pm:109: Retry#87/180 in
1 second...",
                      "04:22:37 (01:27) [INFO] Verify() at ntafbgptest.pm:109: Retry#88/180 in
1 second...",
                      "04:22:38 (01:28) [INFO] Verify() at ntafbgptest.pm:109: Retry#89/180 in
1 second...",
                      "04:22:40 (01:30) [INFO] Verify() at ntafbgptest.pm:109: Retry#90/180
(already past next 1 sec interval)",
                      "04:22:40 (01:30) [INFO] Verify() at ntafbgptest.pm:109: Retry#91/180 in
1 second...",
                      "04:22:41 (01:31) [INFO] Verify() at ntafbgptest.pm:109: Retry#92/180 in
1 second...",
                      "04:22:42 (01:32) [INFO] Verify() at ntafbgptest.pm:109: Retry#93/180 in
1 second...",
                      "04:22:43 (01:33) [INFO] Verify() at ntafbgptest.pm:109: Retry#94/180 in
1 second...",
                      "04:22:44 (01:34) [INFO] Verify() at ntafbgptest.pm:109: Retry#95/180 in
1 second...",
                      "04:22:45 (01:35) [INFO] Verify() at ntafbgptest.pm:109: Retry#96/180 in
1 second...",
                      "04:22:46 (01:36) [INFO] Verify() at ntafbgptest.pm:109: Retry#97/180 in
1 second...",
                      "04:22:47 (01:37) [INFO] Verify() at ntafbgptest.pm:109: Retry#98/180 in
1 second...",
                      "04:22:49 (01:39) [INFO] Verify() at ntafbgptest.pm:109: Retry#99/180
(already past next 1 sec interval)",
                      "04:22:49 (01:39) [INFO] Verify() at ntafbgptest.pm:109: Retry#100/180
in 1 second...",
                      "04:22:50 (01:40) [INFO] Verify() at ntafbgptest.pm:109: Retry#101/180
in 1 second...",
                      "04:22:51 (01:41) [INFO] Verify() at ntafbgptest.pm:109: Retry#102/180
in 1 second...",
                      "04:22:52 (01:42) [INFO] Verify() at ntafbgptest.pm:109: Retry#103/180
in 1 second...",
                      "04:22:53 (01:43) [INFO] Verify() at ntafbgptest.pm:109: Retry#104/180
in 1 second...",
                      "04:22:54 (01:44) [INFO] Verify() at ntafbgptest.pm:109: Retry#105/180
in 1 second...",
                      "04:22:55 (01:45) [INFO] Verify() at ntafbgptest.pm:109: Retry#106/180
in 1 second...",
                      "04:22:56 (01:46) [INFO] Verify() at ntafbgptest.pm:109: Retry#107/180
in 1 second...",
                      "04:22:57 (01:47) [INFO] Verify() at ntafbgptest.pm:109: Retry#108/180
in 1 second...",
                      "04:22:59 (01:49) [INFO] Verify() at ntafbgptest.pm:109: Retry#109/180
in 1 second...",
                      "04:23:00 (01:50) [INFO] Verify() at ntafbgptest.pm:109: Retry#110/180
in 1 second...",
                      "04:23:01 (01:51) [INFO] Verify() at ntafbgptest.pm:109: Retry#111/180
in 1 second...",
                      "04:23:02 (01:52) [INFO] Verify() at ntafbgptest.pm:109: Retry#112/180
in 1 second...",
                      "04:23:03 (01:53) [INFO] Verify() at ntafbgptest.pm:109: Retry#113/180
```

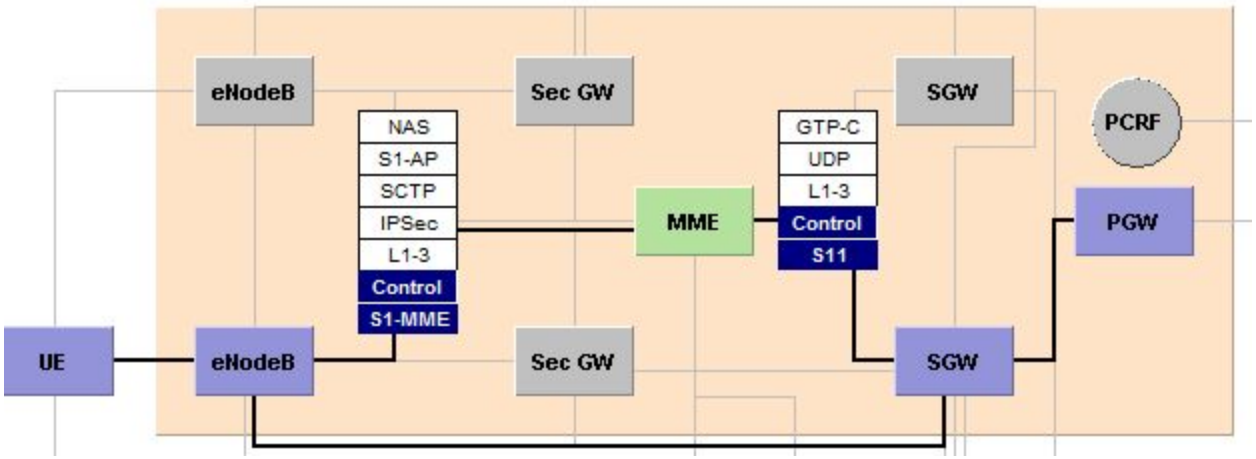
```
in 1 second...",
                    "04:23:04 (01:54) [INFO] Verify() at ntafbgptest.pm:109: Retry#114/180
in 1 second...",
                    "04:23:05 (01:55) [INFO] Verify() at ntafbgptest.pm:109: Retry#115/180
in 1 second...",
                    "04:23:06 (01:56) [INFO] Verify() at ntafbgptest.pm:109: Retry#116/180
in 1 second...",
                    "04:23:08 (01:58) [INFO] Verify() at ntafbgptest.pm:109: Retry#117/180
(already past next 1 sec interval)",
                    "04:23:08 (01:58) [INFO] Verify() at ntafbgptest.pm:109: Retry#118/180
in 1 second...",
                    "04:23:09 (01:59) [INFO] Verify() at ntafbgptest.pm:109: Retry#119/180
in 1 second...",
                    "04:23:10 (02:00) [INFO] Verify() at ntafbgptest.pm:109: Retry#120/180
in 1 second...",
                    "04:23:11 (02:01) [INFO] Verify() at ntafbgptest.pm:109: Retry#121/180
in 1 second...",
                    "04:23:12 (02:02) [INFO] Verify() at ntafbgptest.pm:109: Retry#122/180
in 1 second...",
                    "04:23:14 (02:04) [INFO] Verify() at ntafbgptest.pm:109: Retry#123/180
(already past next 1 sec interval)",
                    "04:23:14 (02:04) [INFO] Verify() at ntafbgptest.pm:109: Retry#124/180
in 1 second...",
                    "04:23:15 (02:05) [INFO] Verify() at ntafbgptest.pm:109: Retry#125/180
in 1 second...",
                    "04:23:16 (02:06) [INFO] Verify() at ntafbgptest.pm:109: Retry#126/180
in 1 second...",
                    "04:23:17 (02:07) [INFO] Verify() at ntafbgptest.pm:109: Retry#127/180
in 1 second...",
                    "04:23:18 (02:08) [INFO] Verify() at ntafbgptest.pm:109: Retry#128/180
in 1 second...",
                    "04:23:19 (02:09) [INFO] Verify() at ntafbgptest.pm:109: Retry#129/180
in 1 second...",
                    "04:23:20 (02:10) [INFO] Verify() at ntafbgptest.pm:109: Retry#130/180
in 1 second...",
                    "04:23:21 (02:11) [INFO] Verify() at ntafbgptest.pm:109: Retry#131/180
in 1 second...",
                    "04:23:22 (02:12) [INFO] Verify() at ntafbgptest.pm:109: Retry#132/180
in 1 second...",
                    "04:23:23 (02:13) [INFO] Verify() at ntafbgptest.pm:109: Retry#133/180
in 1 second...",
                    "04:23:25 (02:15) [INFO] Verify() at ntafbgptest.pm:109: Retry#134/180
(already past next 1 sec interval)",
                    "04:23:25 (02:15) [INFO] Verify() at ntafbgptest.pm:109: Retry#135/180
in 1 second...",
                    "04:23:26 (02:16) [INFO] Verify() at ntafbgptest.pm:109: Retry#136/180
in 1 second...",
                    "04:23:27 (02:17) [INFO] Verify() at ntafbgptest.pm:109: Retry#137/180
in 1 second...",
                    "04:23:28 (02:18) [INFO] Verify() at ntafbgptest.pm:109: Retry#138/180
in 1 second...",
                    "04:23:29 (02:19) [INFO] Verify() at ntafbgptest.pm:109: Retry#139/180
in 1 second...",
                    "04:23:30 (02:20) [INFO] Verify() at ntafbgptest.pm:109: Retry#140/180
in 1 second...",
                    "04:23:31 (02:21) [INFO] Verify() at ntafbgptest.pm:109: Retry#141/180
in 1 second...",
                    "04:23:32 (02:22) [INFO] Verify() at ntafbgptest.pm:109: Retry#142/180
in 1 second...",
                    "04:23:33 (02:23) [INFO] Verify() at ntafbgptest.pm:109: Retry#143/180
in 1 second...",
                    "04:23:35 (02:25) [INFO] Verify() at ntafbgptest.pm:109: Retry#144/180
(already past next 1 sec interval)",
                    "04:23:35 (02:25) [INFO] Verify() at ntafbgptest.pm:109: Retry#145/180
in 1 second...",
                    "04:23:36 (02:26) [INFO] Verify() at ntafbgptest.pm:109: Retry#146/180
```

```
in 1 second...",
                    "04:23:37 (02:27) [INFO] Verify() at ntafbgptest.pm:109: Retry#147/180
in 1 second...",
                    "04:23:38 (02:28) [INFO] Verify() at ntafbgptest.pm:109: Retry#148/180
in 1 second...",
                    "04:23:39 (02:29) [INFO] Verify() at ntafbgptest.pm:109: Retry#149/180
in 1 second...",
                    "04:23:40 (02:30) [INFO] Verify() at ntafbgptest.pm:109: Retry#150/180
in 1 second...",
                    "04:23:41 (02:31) [INFO] Verify() at ntafbgptest.pm:109: Retry#151/180
in 1 second...",
                    "04:23:42 (02:32) [INFO] Verify() at ntafbgptest.pm:109: Retry#152/180
in 1 second...",
                    "04:23:43 (02:33) [INFO] Verify() at ntafbgptest.pm:109: Retry#153/180
in 1 second...",
                    "04:23:45 (02:35) [INFO] Verify() at ntafbgptest.pm:109: Retry#154/180
(already past next 1 sec interval)",
                    "04:23:45 (02:35) [INFO] Verify() at ntafbgptest.pm:109: Retry#155/180
in 1 second...",
                    "04:23:46 (02:36) [INFO] Verify() at ntafbgptest.pm:109: Retry#156/180
in 1 second...",
                    "04:23:47 (02:37) [INFO] Verify() at ntafbgptest.pm:109: Retry#157/180
in 1 second...",
                    "04:23:48 (02:38) [INFO] Verify() at ntafbgptest.pm:109: Retry#158/180
in 1 second...",
                    "04:23:49 (02:39) [INFO] Verify() at ntafbgptest.pm:109: Retry#159/180
in 1 second...",
                    "04:23:51 (02:41) [INFO] Verify() at ntafbgptest.pm:109: Retry#160/180
(already past next 1 sec interval)",
                    "04:23:51 (02:41) [INFO] Verify() at ntafbgptest.pm:109: Retry#161/180
in 1 second...",
                    "04:23:52 (02:42) [INFO] Verify() at ntafbgptest.pm:109: Retry#162/180
in 1 second...",
                    "04:23:53 (02:43) [INFO] Verify() at ntafbgptest.pm:109: Retry#163/180
in 1 second...",
                    "04:23:54 (02:44) [INFO] Verify() at ntafbgptest.pm:109: Retry#164/180
in 1 second...",
                    "04:23:55 (02:45) [INFO] Verify() at ntafbgptest.pm:109: Retry#165/180
in 1 second...",
                    "04:23:56 (02:46) [INFO] Verify() at ntafbgptest.pm:109: Retry#166/180
in 1 second...",
                    "04:23:57 (02:47) [INFO] Verify() at ntafbgptest.pm:109: Retry#167/180
in 1 second...",
                    "04:23:58 (02:48) [INFO] Verify() at ntafbgptest.pm:109: Retry#168/180
in 1 second...",
                    "04:24:00 (02:50) [INFO] Verify() at ntafbgptest.pm:109: Retry#169/180
(already past next 1 sec interval)",
                    "04:24:00 (02:50) [INFO] Verify() at ntafbgptest.pm:109: Retry#170/180
in 1 second...",
                    "04:24:01 (02:51) [INFO] Verify() at ntafbgptest.pm:109: Retry#171/180
in 1 second...",
                    "04:24:02 (02:52) [INFO] Verify() at ntafbgptest.pm:109: Retry#172/180
in 1 second...",
                    "04:24:03 (02:53) [INFO] Verify() at ntafbgptest.pm:109: Retry#173/180
in 1 second...",
                    "04:24:04 (02:54) [INFO] Verify() at ntafbgptest.pm:109: Retry#174/180
in 1 second...",
                    "04:24:05 (02:55) [INFO] Verify() at ntafbgptest.pm:109: Retry#175/180
in 1 second...",
                    "04:24:06 (02:56) [INFO] Verify() at ntafbgptest.pm:109: Retry#176/180
in 1 second...",
                    "04:24:07 (02:57) [INFO] Verify() at ntafbgptest.pm:109: Retry#177/180
in 1 second...",
                    "04:24:08 (02:58) [INFO] Verify() at ntafbgptest.pm:109: Retry#178/180
in 1 second...",
                    "04:24:10 (03:00) [ERROR] NOK: Verify all bgp neighbors are up andall
```

```
routes were learned."
                    ],
                    "reference":"__ANON__() at ntafbgptest.pm:109",
                    "return":false,
                    "verify_attempts":179,
                    "verify_interval":1,
                    "verify_severity":"error"
                  }
                }
              ],
              "status":"fail",
              "totalTime":"[05:29]",
              "endTime":"2014-11-11T18:25:43.511Z"
            }
          ],
          "status":"fail",
          "totalTime":"[05:21]",
          "endTime":"2014-11-11T18:25:43.511Z"
        },
        {
          "name":"Step 3: Clean up Test Case",
          "duration":174,
          "executionIssues":[
            "04:24:13 (00:03) [INFO] Cleanup(): Inside the common clean up routine calledfrom
XTESTCASE001",
            "04:24:13 (00:03) [INFO] Cleanup(): Clean UP all Session..",
            "04:27:04 (02:54) [INFO] Clean_ConfigRouters()"
          ],
          "status":"pass",
          "totalTime":"[08:31]",
          "endTime":"2014-11-11T18:25:43.511Z"
        }
      ],
      "stepsFailed":2,
      "stepsPassed":8,
      "stepsSkipped":0,
      "stepsTaken":[
        "[00:00] Case 001: tc_bgpTest() started at 2014/11/11 04:15:39",
        "[00:14] Step 1: Config: Initialization",
        "[00:17] Step 1.1: Assigning some IP Addresses to the Test Network.",
        "[00:20] Step 1.2: Assiging Ipv6 networks to the router and the Test Tool.",
        "[00:24] Step 1.3: Configure the Router loopback,",
        "[00:36] Step 1.4: Configure Test Tool Interface.",
        "[05:00] Step 1.5: Configure BGP groups.",
        "[05:21] Step 2: Start BGP emulations",
        "[05:23] Step 2.1: Start BGP emulations for each group",
        "[05:29] Step 2.2: Verify BGP Neighbors.",
        "[08:31] NOK:Verify all bgp neighbors are up and all routes were learned.",
        "[08:31] Step 3: Clean up Test Case",
        "[11:25] Case 001: tc_bgpTest() ended with 1 error(s) in 11 minutes 25 seconds"
      ],
      "startTime":"2014-11-11T18:25:43.511Z",
      "endTime":"2014-11-11T18:25:43.511Z",
      "totalTime":"11 minutes 25 seconds"
    }
  ]
}
```


MME SMS Messaging

This use case tests the SMS feature of an MME in an LTE network. Conceptually the type of network being tested looks like Figure 1 below. The MME being tested (r0) should properly discover the emulated UEs and route the SMS message properly both to and from the emulated UEs.




```
{
  "name":"MME Nodal SMS Testing",
  "createdDate":"2015-03-05T15:36:10Z",
  "description":"Test MME SMS Messaging Performance",
  "user":"mmeuser",
  "testSuites":[
    {
      "name":"Test Session 1",
      "status":"pass",
      "executionIssues":["None"],
      "topologies":[
        {
          "name":"TestCase 1 Topology",
          "nodes":[
            {
              "name":"MME",
              "functionalClass":"MME",
              "interfaces":[
                {
                  "name":"S1-MME",
                  "ipAddress":"10.202.88.1/24",
                  "destination":"testvendor:enodeB",
                  "speed":"10g",
                  "category":"network"
                },
                {
                  "name":"S11",
                  "ipAddress":"10.202.88.1/24",
                  "destination":"testvendor:sgw",
                  "speed":"10g",
                  "category":"network"
                }
              ]
            },
            {
              "name":"testvendor",
              "functionalClass":"protocolEmulator,trafficGenerator",
              "interfaces":{
                "interface":[
                  {
```

```
              "name":"eNodeB",
              "destination":"MME:S1-MME",
              "ipAddress":"10.202.86.1/24",
              "category":"network"
            },
            {
              "name":"eNodeBUser",
              "destination":"sgwUser",
              "ipAddress":"10.202.86.2/24",
              "category":"network"
            },
            {
              "name":"sgw",
              "destination":"MME:S11",
              "type":"ethernet10g",
              "ipAddress":"10.202.86.3/24",
              "category":"network"
            },
            {
              "name":"sgwUser",
              "destination":"eNodeBUser",
              "type":"ethernet10g",
              "ipAddress":"10.202.86.4/24",
              "category":"network"
            }
          ]
        }
      }
    ]
  }
],
"vendors":[
  {
    "name": "MME",
    "vendorName":"XYZ",
    "testConfiguration":[
      "TBD"
    ]
  },
  {
    "name": "testvendor",
    "vendorName":"FooTest Company",
    "parameters":{
      "TasVersion":"13.5.0.5",
      "TsVersion":"13.5.0.7",
      "NodeId":"TBD",
      "toolId":"TBD",
      "host":"10.71.30.110",
      "username":"mmeuser",
      "password.masked":"true",
      "password":"abcdefgh"
    },
    "testConfiguration":[
      "See test suite parameters"
    ]
  }
],
```

```
"parameters":{
  "name":"MMENodal4.2.1.1.50",
  "id":29,
  "library":33,
  "user":"sms",
  "passfailstatus":"PASSED",
  "status":"COMPLETE",
  "reserve":false,
  "criteria":[
    {
      "type":"measurement",
      "operator":"EQ",
      "value":"20000",
      "ts":0,
      "tc":0,
      "tab":"Test Summary",
      "name":"Actual Session Connects"
    },
    {
      "type":"measurement",
      "operator":"EQ",
      "value":"70000",
      "ts":0,
      "tc":0,
      "tab":"SMS",
      "name":"SMS-DELIVER Packets Rcvd"
    }
  ],
  "reportOptions":{
    "format":0,
    "ts":-1,
    "tc":-1,
    "createDatabase":false,
    "l47Subtotal":false,
    "l47PerSession":false,
    "l3PerSession":false,
    "mobileAccessPerSession":false,
    "coreNodesPerSession":false,
    "tsReport":false
  },
  "tsGroups":[
    {
      "tsId":14,
      "testcases":[
        {
          "name":"MMENodal4.2.1.1.50",
          "type":"MME Nodal",
          "label":" [MME Nodal]",
          "parameters":{
            "ActiveEntryTime":"0",
            "AttachId":"0",
            "BearerCfgFileEn":"false",
            "BearerV4AddrPool":"88.0.0.1",
            "BearerV6AddrPool":"2020:8688::8800:1",
            "DataIpSecTunnelEnabler1":"false",
            "DataTraffic":"Disabled",
            "DedicatedsPerDefaultBearer":"0",
```

```
"DefaultBearers":"1",
"DetachMode":"1",
"DetachOption":"0",
"DetachType":"0x1",
"DisconnectRate":"1000.0",
"DisconnectType":"Mobile Node",
"EchoMessageTime":"0",
"EmbmsEn":"false",
"EmergencyPdnIndexEn":"false",
"EnableExternalData":"0",
"EnbAltTacEn":"false",
"EnbCellAccessMode":"2",
"EnbCfgFileEn":"false",
"EnbControlAddr":{
  "class":"TestNode",
  "forcedEthInterface":"",
  "innerVlanId":0,
  "ip":"10.202.86.1",
  "mac":"",
  "mtu":1500,
  "nextHop":"",
  "numLinksOrNodes":1,
  "numVlan":1,
  "phy":"eth2",
  "uniqueVlanAddr":false,
  "vlanDynamic":0,
  "vlanId":0,
  "vlanTagType":0
},
"EnbControlAddrErrInj":"0",
"EnbFailedThresholdPercent":"0",
"EnbId":"1",
"EnbIdType":"0",
"EnbIpsecEn":"false",
"EnbMcc":"000",
"EnbMmeIpsecEn":"false",
"EnbMnc":"000",
"EnbName":"enb.foofest.com",
"EnbNumLinksEn":"false",
"EnbNumSutsEn":"false",
"EnbSctpMultiHomeMultiSrcEn":"false",
"EnbSctpMultiHomedEn":"false",
"EnbSendS1ResetEn":"false",
"EnbStatusTransferMsgEn":"false",
"EnbTac":"000",
"EnbUpdateConfigEn":"false",
"EnbUserAddr":{
  "class":"TestNode",
  "forcedEthInterface":"",
  "innerVlanId":0,
  "ip":"10.202.86.2",
  "mac":"",
  "mtu":1500,
  "nextHop":"",
  "numLinksOrNodes":1,
  "numVlan":1,
  "phy":"eth2",
```

```
  "uniqueVlanAddr":false,
  "vlanDynamic":0,
  "vlanId":0,
  "vlanTagType":0
},
"EnbUserAddrEn":"true",
"EnbUserAddrErrInj":"0",
"Gtp2AmbrDownlink":"1",
"Gtp2AmbrUplink":"1",
"Gtp2ApnNumSpecifiedApns_0":"0",
"Gtp2ApnRestriction":"0",
"Gtp2ApnRetries_0":"0",
"Gtp2ApnSelectMode":"0",
"Gtp2ApnSpecified_0":{
  "class":"Array",
  "array":[

  ]
},
"Gtp2ApnTotalApns_0":"1",
"Gtp2Apn_0":"ssenoauth146",
"Gtp2CustomMsgEn":"false",
"Gtp2EchoTimeSec":"0",
"Gtp2ExtEn":"false",
"Gtp2GtpcTunnelEndptId":"1000000",
"Gtp2GtpuTunnelEndptId":"2000000",
"Gtp2Imei":"50502410121507",
"Gtp2Imsi":"505024101215074",
"Gtp2MsIsdnEn":"false",
"Gtp2PagingTime":"0",
"Gtp2QosArpPreemptCapEn_1":"false",
"Gtp2QosArpPreemptVulnEn_1":"false",
"Gtp2QosArpValue_1":"1",
"Gtp2QosClassId_1":"1",
"Gtp2QosDetail":"Summary",
"Gtp2QosGuarDownlink_1":"1",
"Gtp2QosGuarUplink_1":"1",
"Gtp2QosMaxDownlink_1":"1",
"Gtp2QosMaxUplink_1":"1",
"Gtp2RadioAccessType":"6",
"Gtp2Version":"8.4.0",
"HomeAddrType":"3",
"HomeAddrTypePerBearerEn":"false",
"IdleEntryTime":"0",
"Imei":"12345678901234",
"Imsi":"123456789012345",
"IncEnbIdEn":"true",
"IncTaiEn":"true",
"LogEnbStatusEn":"false",
"LteStartEnbRate":"1000.0",
"MmeCapabilityEn":"false",
"MmeSut":{
  "class":"Sut",
  "name":"Coast88 10.202.88.1"
},
"MnActivationDelay":"0",
"MsClassmark2A5_1":"1",
```

```
"MsClassmark2A5_2":"0",
"MsClassmark2A5_3":"0",
"MsClassmark2Classmark3":"0",
"MsClassmark2ControlledEarlyClsSendEn":"0",
"MsClassmark2FrequencyCapability":"0",
"MsClassmark2LcsVa":"0",
"MsClassmark2MtSmsPtToPt":"0",
"MsClassmark2NetInitMoCmConnRequest":"0",
"MsClassmark2PowerCapability":"0",
"MsClassmark2PseudoSync":"0",
"MsClassmark2RevLevel":"0",
"MsClassmark2Screening":"0",
"MsClassmark2SoLSA":"0",
"MsClassmark2Ucs2":"0",
"MsClassmark2Vbs":"0",
"MsClassmark2Vgcs":"0",
"MsMcc":"000",
"MsMmec":"0xFF",
"MsMmegi":"0xFFFF",
"MsMnc":"000",
"MsTac":"000",
"NasActRateAdjustInAttachFailureEn":"false",
"NasApnNumSpecifiedApns_0":"0",
"NasApnPcoSummary":"Summary",
"NasApnRetries_0":"0",
"NasApnRoundRobin_0":"false",
"NasApnSpecified_0":{
  "class":"Array",
  "array":[

  ]
},
"NasApnTotalApns_0":"1",
"NasApn_0":"ssenoauth146",
"NasAttachType":"2",
"NasCfgFileEn":"false",
"NasEnbGtpuSeqInS1uEn":"false",
"NasEsmInfoRequiredEn":"false",
"NasIncPcoOpt_1":"none",
"NasPcoAddEn_1":"false",
"NasPcoIpcpEn_1":"false",
"NasReattachWithGutiEn":"true",
"NasUpdateType":"0",
"NasUpdateTypeEn":"true",
"OpType":"OpVar",
"OpVar":"0x63BFA50EE6523365FF14C1F45F88737D",
"PagingDrx":"0",
"PagingResponseDelayTime":"0",
"PeriodicTauUpdateEn":"false",
"ResponseTime":"5000",
"RetryCnt":"5",
"RetryTime":"15",
"S1apCfgFileEn":"false",
"S1apCustomMsgEn":"false",
"S1apPagesToIgnore":"0",
"S1apVersion":"4",
"Sctp3SackRuleEn":"false",
```

```
"SctpChunkSize":"0",
"SctpDestPort":"2905",
"SctpHeartbeatInterval":"30",
"SctpHeartbeatMaxRetry":"5",
"SctpLinkRecoveryEn":"true",
"SctpRestartOnLinkRecoveryEn":"false",
"SctpRtoMax":"60000",
"SctpRtoMin":"1000",
"SctpSackThreshold":"2",
"SctpSrcPort":"2905",
"SctpWindowSize":"32000",
"SecretKey":"0x00",
"SecurityHeaderEmmEn":"true",
"SessionIntervalShape":"Fixed",
"SessionRetries":"true",
"Sessions":"20000",
"SetupTime":"2000",
"SgwControlAddr":{
  "class":"TestNode",
  "forcedEthInterface":"",
  "innerVlanId":0,
  "ip":"10.202.86.3",
  "mac":"",
  "mtu":1500,
  "nextHop":"",
  "numLinksOrNodes":1,
  "numVlan":1,
  "phy":"eth2",
  "uniqueVlanAddr":false,
  "vlanDynamic":0,
  "vlanId":0,
  "vlanTagType":0
},
"SgwControlAddrErrInj":"0",
"SgwEmulationEn":"true",
"SgwUserAddr":{
  "class":"TestNode",
  "forcedEthInterface":"",
  "innerVlanId":0,
  "ip":"10.202.86.4",
  "mac":"",
  "mtu":1500,
  "nextHop":"",
  "numLinksOrNodes":1,
  "numVlan":1,
  "phy":"eth2",
  "uniqueVlanAddr":false,
  "vlanDynamic":0,
  "vlanId":0,
  "vlanTagType":0
},
"SgwUserAddrErrInj":"0",
"SgwUserDualStackEn":"false",
"ShortMsgSvcEn":"true",
"SmApn":"ssenoauth146",
"SmsCpAckWaitTime":"10",
"SmsCpRetries":"2",
```

```
                "SmsDataCoding":"0",
                "SmsIncPriorityEn":"false",
                "SmsIncStatusReqEn":"false",
                "SmsMsgContent":"Take me out to the ball game",
                "SmsMsgCycleType":"Continuous",
                "SmsMsgInterval":"10000",
                "SmsMsgStartDelay":"30000",
                "SmsOrigAddrNum":"9729221234",
                "SmsOrigAddrPlan":"0001",
                "SmsOrigAddrType":"010",
                "SmsProtocolId":"0",
                "SmsRecipAddrNum":"2144694321",
                "SmsRecipAddrPlan":"0001",
                "SmsRecipAddrType":"010",
                "SmsRpAckWaitTime":"40",
                "SmsScAddrNum":"2144696000",
                "SmsScAddrPlan":"0001",
                "SmsScAddrType":"010",
                "SmsSendMsgEn":"true",
                "SmsTpReportWaitTime":"16",
                "SmsValidPeriod":"5",
                "StartRate":"1000.0",
                "SwVersion":"00",
                "SwapOrderOfShortMac":"false",
                "TauTime":"0",
                "TestActivity":"Capacity Test",
                "TestAlgorithmEn":"false",
                "TestType":"MME-NODAL",
                "UeCapEea1En":"true",
                "UeCapEea2En":"true",
                "UeCapEia1En":"true",
                "UeCapEia2En":"true",
                "UeCapInfoIndicationEn":"false",
                "UeContextReleaseReqCause":"0",
                "VendorVariant":"0",
                "Version":"2",
                "VolteEn":"false"
              }
            }
          ]
        }
      ],
      "portCaptureConfig":[
        {
          "tsId":14,
          "port":"eth2",
          "onStart":true,
          "options":""
        }
      ],
      "resultFilesList":[

"http://10.71.30.110/results/MME/03-05__04.33.14.PM__MMENodal4.2.1.1.50__RID-28.log.txt",
        "http://10.71.30.110/results/MME/03-05__04.33.14.PM__MMENodal4.2.1.1.50__RID-28.xls"
      ]
    },
    "logs":[
```

```
"http://10.71.30.110/results/MME/03-05__04.33.14.PM__MMENodal4.2.1.1.50__RID-28.log.txt",
        "http://10.71.30.110/results/MME/03-05__04.33.14.PM__MMENodal4.2.1.1.50__RID-28.xls"
      ]
    }
  ]
}
```

Extensions

Additional Content

Reports may contain additional attributes not listed in this standard. Such additional attributes must be named according to the principles laid out in Section 3 of TS-004 under "Additional Information". Briefly, attribute names must indicate attribute origin.

We recognize that tests often produce complex data such as tables, charts, pcap files, and databases. Consideration of such data is outside the scope of this document. Within the scope of this specification it is permitted to reference files with such data using log object elements. A more complete solution for these types of data can be addressed by future specifications.