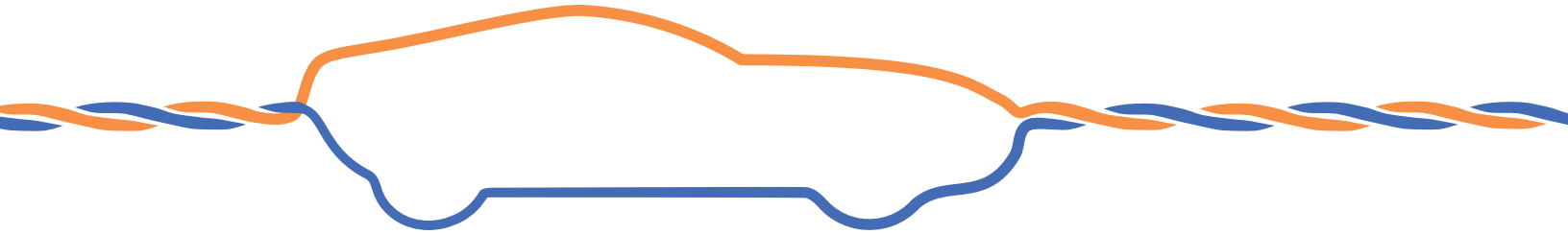
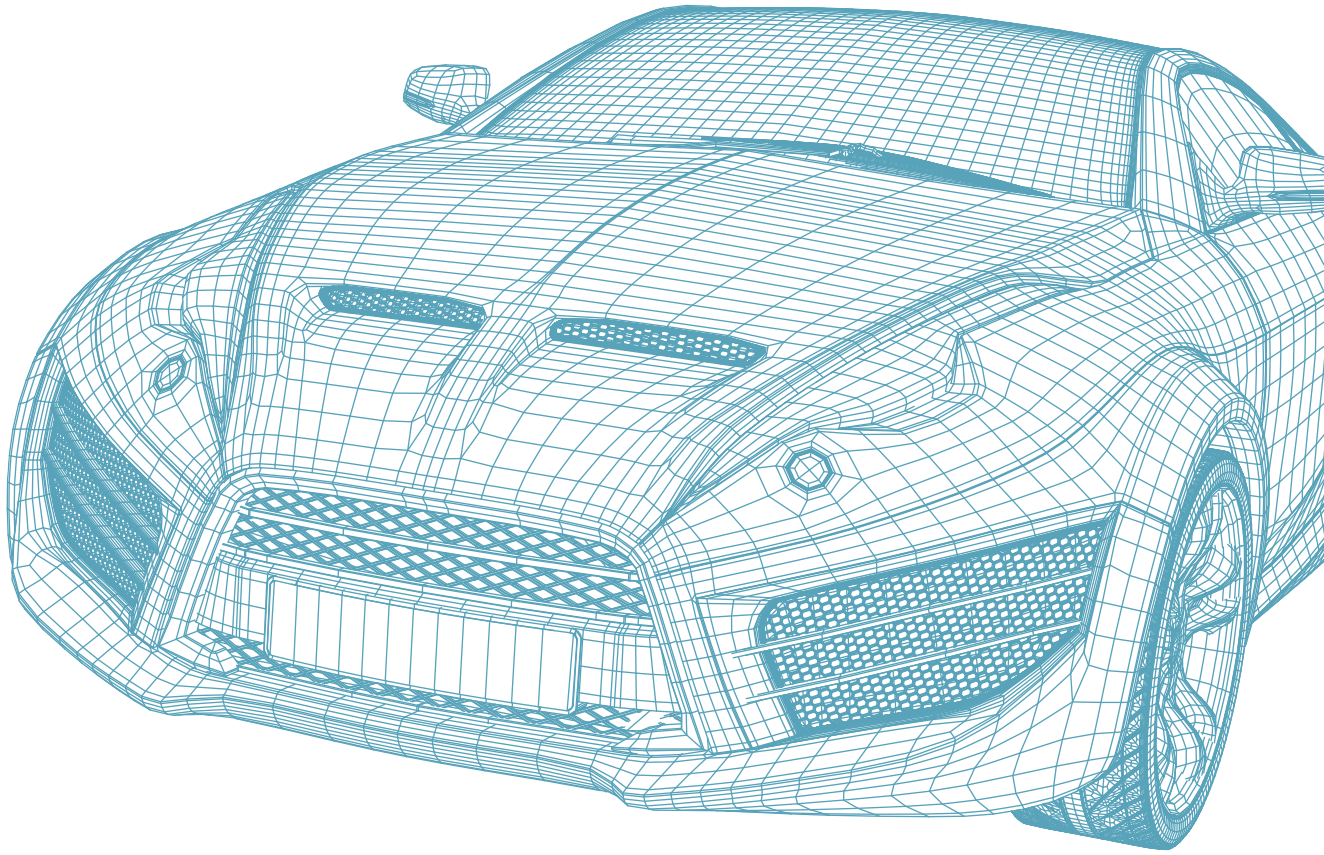
Playbook

spirent™
Promise. Assured.





Automotive Ethernet Playbook

OPEN Alliance SIG/TC8 Test Cases & Methodologies

Table of Contents

1. Automotive Ethernet and OPEN Alliance Automotive Ethernet Standards Overview....3
2. Conformance Testing....5
2.1. Conformance Testing: TTsuite-OPEN-IPv4....6
2.1.1. Objective....6
2.1.2. Test Topology....7
2.1.3. Test Suite Configuration....8
2.1.4. Expected Behavior....10
2.2. Conformance Testing: TTsuite-OPEN-ARP....11
2.2.1. Objective....11
2.2.2. Test Topology....12
2.2.3. Test Suite Configuration....14
2.2.4. Expected Behavior....16
2.3. Conformance Testing: TTsuite-OPEN-IPv4-AUTOConf....17
2.3.1. Objective....17
2.3.2. Test Topology....17
2.3.3. Test Suite Configuration....18
2.3.4. Expected Behavior....19
2.4. Conformance Testing: TTsuite-OPEN-DHCPv4....20
2.4.1. Objective....20
2.4.2. Test Topology....21
2.4.3. Test Suite Configuration....22
2.4.4. Expected Behavior....23
2.5. Conformance Testing: TTsuite-OPEN-TCP....24
2.5.1. Objective....24
2.5.2. Test Topology....26
2.5.3. Test Suite Configuration....26
2.5.4. Expected Behavior:....28
2.6. Conformance Testing: TTsuite-OPEN-UDP....30
2.6.1. Objective....30
2.6.2. Test Topology....30
2.6.3. Test Suite Configuration....31
2.6.4. Expected Behavior....32
2.7. Conformance Testing: TTsuite-OPEN-ICMPv4....33
2.7.1. Objective....33
2.7.2. Test Topology....34
2.7.3. Test Suite Configuration....34
2.7.4. Expected Behavior....35
3. Acronyms....36

1. Automotive Ethernet Overview

As autonomous/assisted driving and connected car are becoming a reality, there will be significant sustained growth in the extent and importance of Ethernet networking in vehicles. The industry is converging upon Single Pair Ethernet (BroadR-Reach®) as its preferred physical standard, and the benefits of 2-wire Ethernet cabling over conventional technologies are simply too great to ignore. Ethernet is a proven off-the-shelf technology that can scale to higher transmission rates as well as provide the necessary functionality required for Automotive application via Time-Sensitive Networking and general networking standards and protocols.

The cost and performance benefits—along with soaring future bandwidth requirements—make Ethernet too attractive for auto manufacturers to ignore. Furthermore, the OPEN Alliance (One Pair Ethernet) Special Interest Group (SIG) has effectively established BroadR-Reach as the de-facto physical layer standard for the industry, clearing the way for its widespread adoption. OABR (OPEN Alliance BroadR-Reach®) replaces the standard, 8-wire shielded twisted pair cabling with a smaller, 2-wire unshielded twisted pair. By simplifying to a single system, reducing expensive copper and duplicated network hardware, OABR, while providing larger bandwidth, is some 30% lighter than equivalent mainstream technologies, improving vehicle efficiency, and reducing installation costs by 80%. The new standard, commonly known as 100BASE-T1 (100Mbps) and 1000BASE-T1 (1Gbps), is becoming essential as ADAS (Advanced Driver Assistance Systems) and "connected car" functions escalate demand for complexity and bandwidth. There are already standard task groups working on 10Mbps and Multi-gig versions of 2-wire Ethernet to accommodate additional ranges of use cases. Moreover, increasing complexity in vehicle computing, infotainment and security will mean the next-generation of connected vehicles require software updates on a regular basis. Switching to BroadR-Reach from classic bus systems has the potential to prevent a bandwidth bottleneck, and transform the job from all-day garage procedure to something that can be carried out while the owner waits—or even to something that can be conducted over-the-air with no need to visit a garage.

The arrival of new systems and protocols requires new testing approaches, and will increase demands upon Automotive research and development engineers in terms of network expertise. What is more, Ethernet also presents the Automotive industry with an entirely new challenge that, unless taken seriously, could prove catastrophic for safety, reliability and brand reputation: network security. Potentially, making Ethernet the backbone of a car's network exposes critical performance and safety systems to the risk of attack—bringing Automotive into the frontline of the rapidly evolving world of IT security. As a member of the OPEN Alliance SIG, Spirent's network testing engineers have been working closely with the industry to develop tailored answers to the challenges the emerging technologies bring, ensuring tomorrow's vehicles take full advantage of the many benefits of Automotive Ethernet.

Test Requirements

Automotive Ethernet has different characteristics from existing bus systems, there remain a number of technologies that will need to be tested:

- Ethernet and Internet Protocol (Ethernet/IP)
- Applications, and their interaction with the new network
- Gateways and switches
- Protocols

Within each area, several important characteristics will affect the system's ability to carry out different functions. For example:

- **Availability and reliability testing** across a full range of likely scenarios, to ensure consistent, predictable performance of critical powertrain, chassis and body functions
- **Quality testing** for smooth handling of media files within infotainment systems
- **Latency testing** to guarantee timely operation of ADAS equipment
- **Load and scalability testing** for high-bandwidth applications like surround view cameras

Key Test Types

Three types of tests are likely to prove particularly useful:

- **Conformance testing** ensures protocols function correctly and meet approved standards. Within IT, such tests are now often taken as read, since the standards are so well established and understood that OEMs can rely on their vendors. In Automotive, however, OABR Ethernet is still new and developing. What is more, manufacturers assume overall responsibility for the function of the entire vehicle, and carry the reputational damage and recall costs if things go wrong, so will understandably want to test for themselves.
- **Negative testing** confirms how the system responds when it encounters errors, encounters unexpected or nonstandard signals, or no signal at all—while "fuzzing" rapidly tests every possible permutation close to the expected outcome. Exhaustive negative testing is particularly important in Automotive scenarios, where a vehicle must remain safe to use, whatever difficulties its network encounters.
- **Performance testing** checks how much load the system can bear, and what happens when this limit is exceeded. For example, when faced with a sudden surge in demand for network bandwidth, can a vehicle still identify, prioritize and deliver the most important messages, such as brake function?

In some cases, standard IT industry interfaces can be suitable for Automotive use. However, the requirement for hardware-in-the-loop testing, along with the ability to customize automated test runs, often means an Automotive-tailored solution is required.

OPEN Alliance Automotive Ethernet Standards Overview

This document provides an overview of the various OPEN Alliance Automotive Ethernet TC8 protocols and step-by-step instructions on how to test different use cases using Spirent TTworkbench.

You can find the overview of the technology on Open Alliance's webpage: http://www.opensig.org/about/about-open/

List of standards mentioned in this document:

Standard	Title
RFC 768	User Datagram Protocol
RFC 791	Internet Protocol
RFC 792	Internet Control Message Protocol
RFC 793	Transmission Control Protocol
RFC 826	Ethernet Address Resolution Protocol
RFC 894	Standard for the Transmission of IP Datagrams over Ethernet Networks
RFC 1122	Requirements for Internet Hosts - Communication Layers
RFC 1812	Requirement for IP Version 4 Routers
RFC 2131	Dynamic Host Configuration Protocol
RFC 2132	DHCP Options and BOOTP Vendor Extensions
RFC 2460	Internet Protocol, Version 6 (IPv6) Specification
RFC 3927	Dynamic Configuration of IPv4 Link-Local Addresses

2. Conformance Testing

Spirent OPEN Alliance SIG Conformance Test Suite Pack is an implementation of the OPEN Alliance TC8 Test Specification. It consists of different protocol conformance test suites for Automotive Ethernet ECU and Network tests running on Spirent C50 devices with both BroadR-Reach and RJ45 network interface cards. All test suites are prepared for full test automation and include frameworks for individual adaptation. Users can customize test scenarios, for instance to modify or exclude test stub activities, or to add negative testing, etc.

References

OPEN Alliance Automotive Ethernet ECU Test Specification TC8 ECU and Network Test, Test Specification ECU 2.0, July 18, 2017

Included Test Suites

- TTsuite-OPEN-IPv4
- TTsuite-OPEN-ARP
- TTsuite-OPEN-DHCP-Client
- TTsuite-OPEN-ICMPv4
- TTsuite-OPEN-IPv4-Autoconf
- TTsuite-OPEN-UDP
- TTsuite-OPEN-TCP
- TTsuite-OPEN-SOME/IP

For more information, visit:
https://www.spirent.com/Products/TTworkbench/TTsuites/OPEN-Alliance-SIG-Conformance

OPEN Alliance SIG Conformance First Steps User's Guide is available with any TTworkbench software and can be downloaded here.

2.1. Conformance Testing: TTsuite-OPEN-IPv4

2.1.1. Objective

The objective of this test suite is to verify the Internet Protocol Version 4 (IPv4) communication of the Device Under Test according to the following standards:

- RFC 791 - Internet Protocol, Darpa Internet Program Protocol Specification
- RFC 1122 - Requirements for Internet Hosts - Communication Layers
- RFC 894 - Standard for the Transmission of IP Datagrams over Ethernet Network

The representation of the IPv4 Header fields is shown in the figure below:

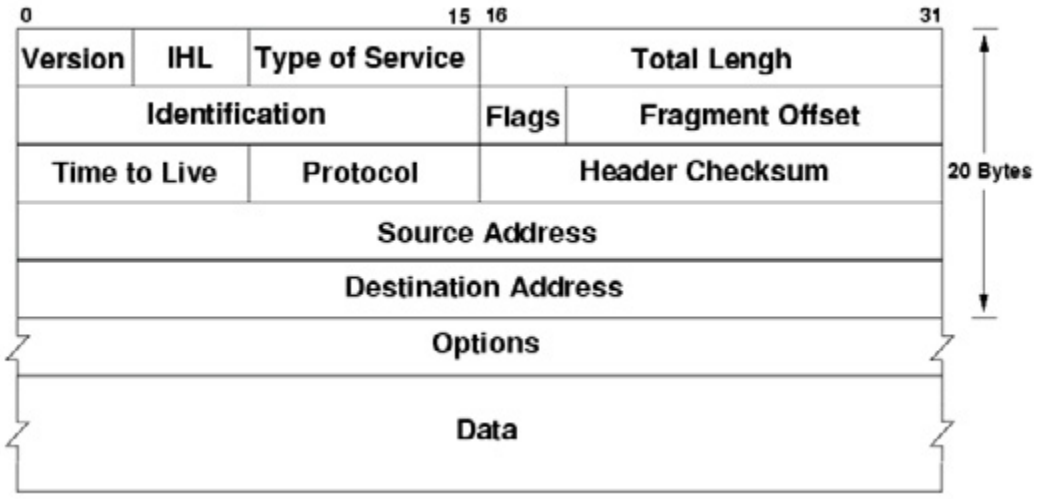


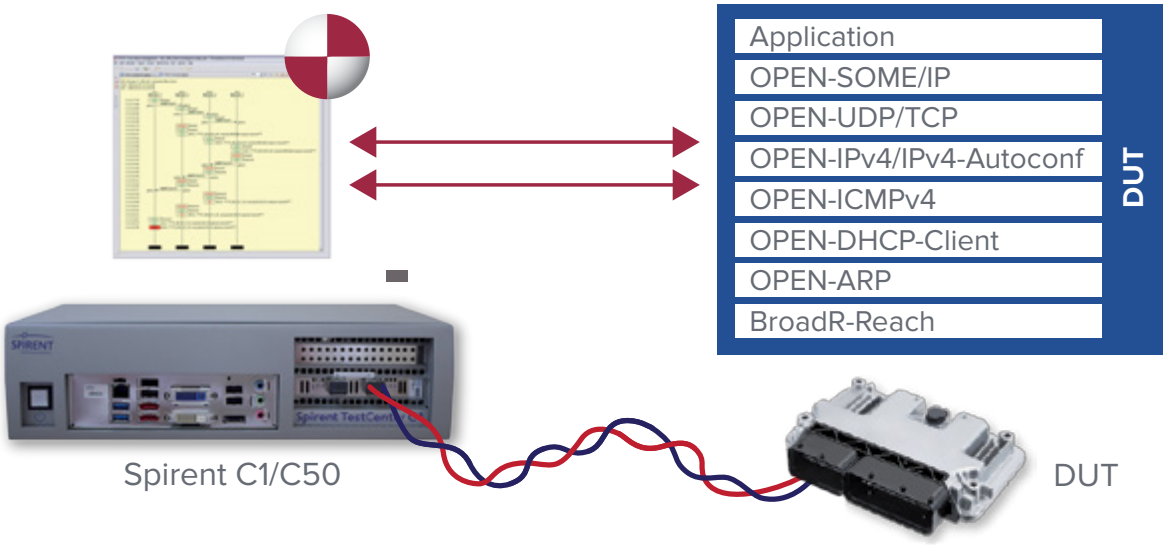
Figure 1. IPv4 Header.

TTsuite-OPEN-IPv4 defines 8 test categories and 51 test cases checking the conformance of various fields in the IPv4 header, the fragmentation and the reassembly of the fragmented IPv4 packets. The test case categories are listed below:

- IPv4 Header
- IPv4 Checksum
- IPv4 Time to Live
- IPv4 Version Number
- IPv4 Addressing
- IPv4 Options
- IPv4 Reassembly
- IPv4 Fragments

2.1.2. Test Topology

For step-by-step instructions on how to install and load the test campaign from the TTsuite-OPEN-IPv4 test suite please read: https://support.spirent.com/SpirentCSC/SC_KnowledgeView?id=DOC10947&origin=cscanswers

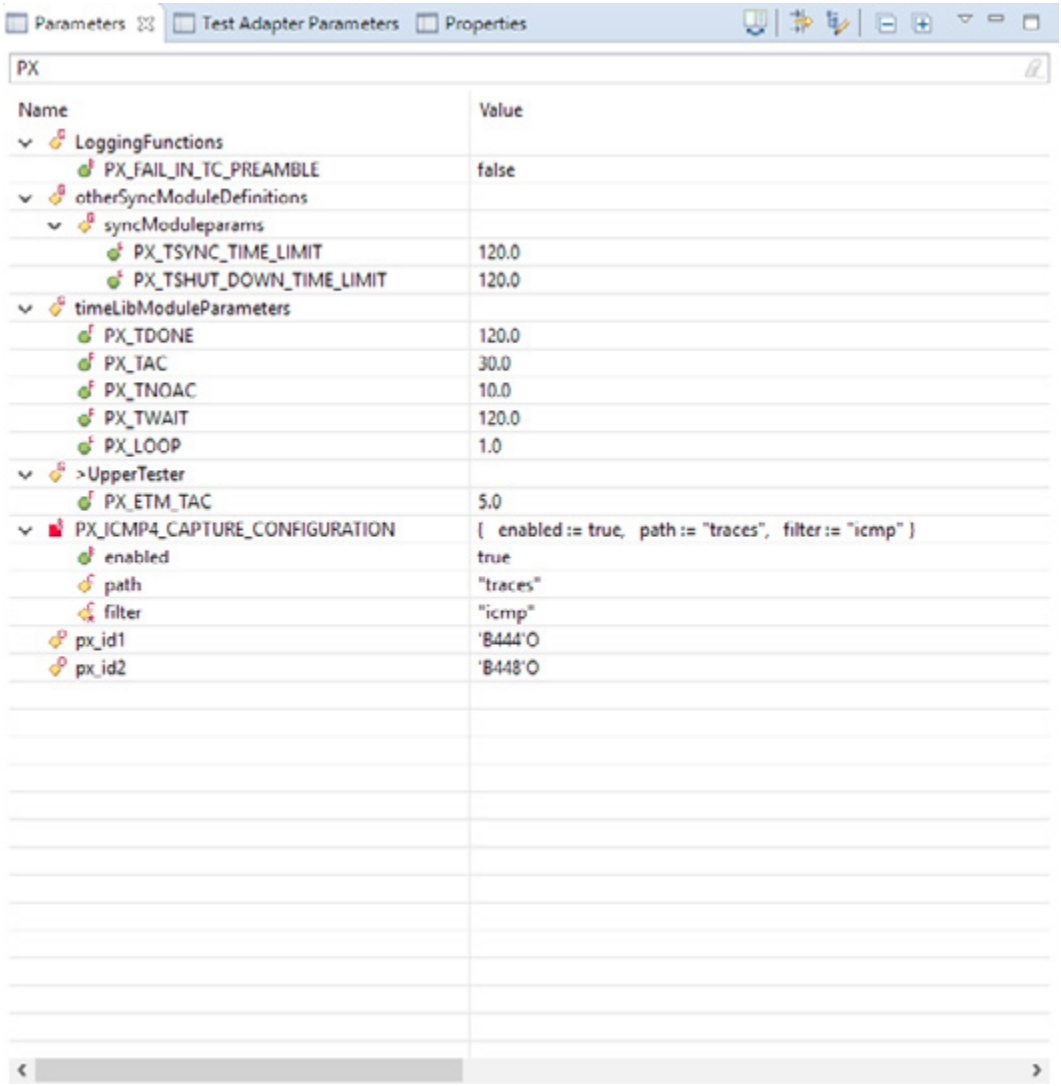


The test topology is very simple, just connect on the card where TTworkbench (TTwb) is active on the C50. For more details about TTworkbench and C50 please read https://support.spirent.com/SpirentCSC/SC_KnowledgeView?id=DOC10828 .

The general input parameters should be configured before running the test.

For more details on how to configure and run test cases from TTsuite-OPEN-IPv4 test suite, please read the help contents by going to the menu entry **Help -> Help Contents -> Using OPEN Alliance Automotive IPv4 Test Solution-> TTsuite-OPEN-IPv4**.

TIP: In general, the number of parameters in a test suite is big and it is useful to filter the displayed ones. By pressing the *"Show Used Module Parameters Based on Test Case Selection"* button (see below) the list is drastically reduced. Another way of filtering the parameters is by using the filter text. For example, if only the Device under Test (DUT) parameters are to be modified, one can type "DUT" in the filter box and further reduce the list of displayed parameters. The same approach can be used if only the Test System (TS) parameters should be displayed.



2.1.3. Test Suite Configuration

The test suite parameters can be configured in the *Parameters* view of the *TTCN-3 Execution Management* perspective.

2.1.3.1. Upper Tester Configuration

The Upper Tester configuration is like TCP, see chapter 2.5.3.1.

2.1.3.2. IP Interface Configuration

To configure the IP interface, just specify any information available about the connected network interfaces like seen below. It is possible to provide the interface name, its MAC and IP addresses.

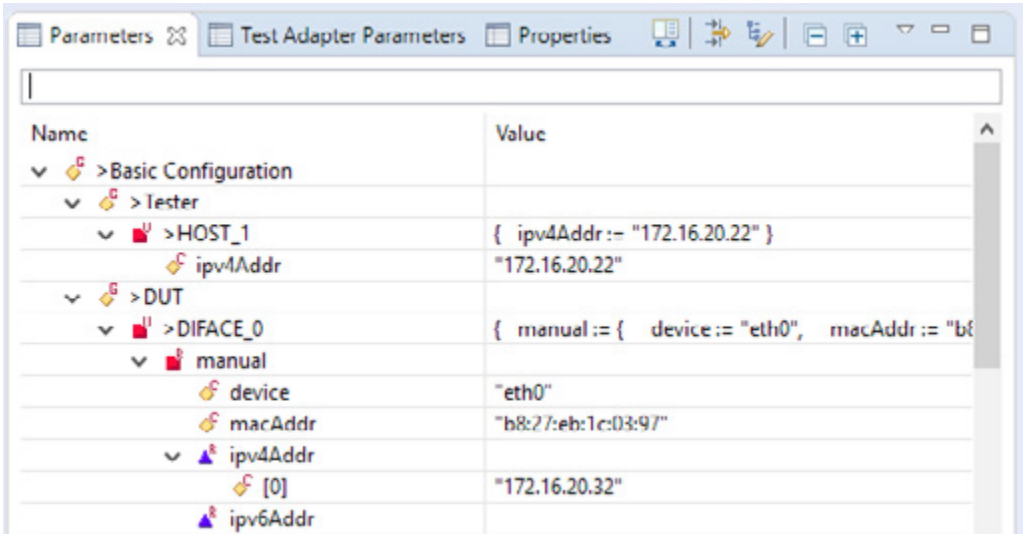


Figure 2. Network Interface Specification for the Test System (HOST_1) and the DUT (DIFACE_0).

2.1.3.3. Test Suite Specific Configuration

If a test case does not pass or the DUT has a custom IPv4 configuration, consider changing the IPv4 addresses listed below and run the test case again. There also exists an advanced configuration for specific IPv4 header values and timers.

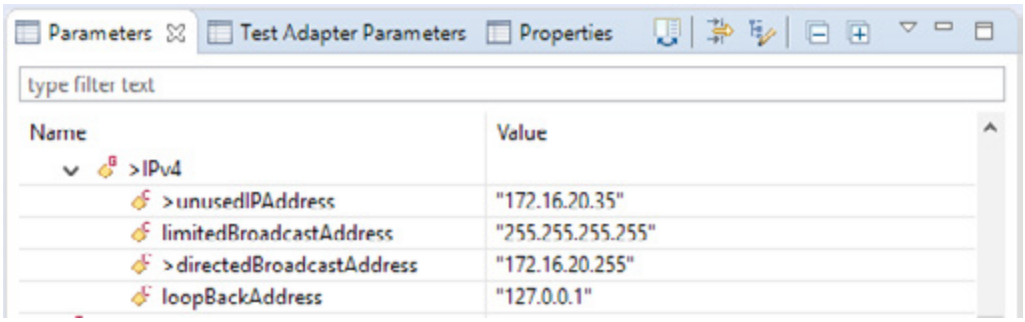


Figure 3. IPv4 Parameter Configuration.

2.1.3.4. Capture Configuration

To capture incoming and outgoing traffic, the parameter PX_CAPTURE_CONFIGURATION is provided. Enabling it with a custom filter as seen below stores a pcap file for further analysis in the specified location for each executed test case. By default, the configuration captures all IPv4 and ICMPv4 messages. Changing the filter has no effect on the actual test case execution.

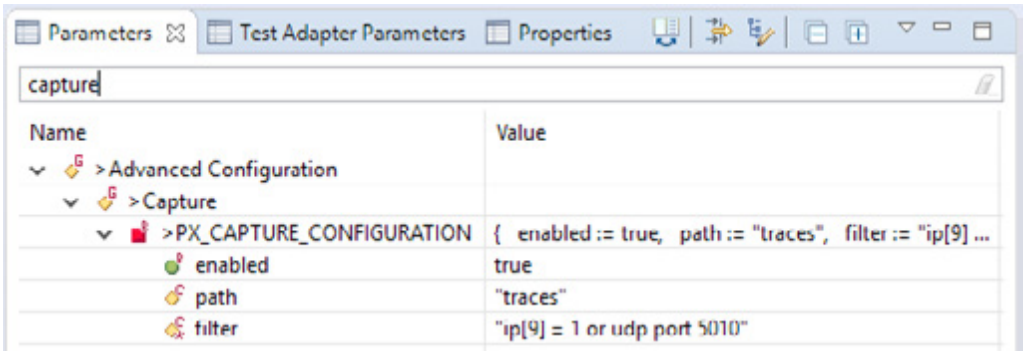


Figure 4. IPv4 Capture Configuration.

2.1.4. Expected Behavior

a. Every expected message is received

Once everything is configured, the test can be run by double clicking on it, or by pressing the run button. If the configuration was correct and if the DUT behaves correctly, messages exchanged between the TS and the DUT are seen in the graphical log and the final verdict is ***pass*** followed by a verdict reason, as shown below:

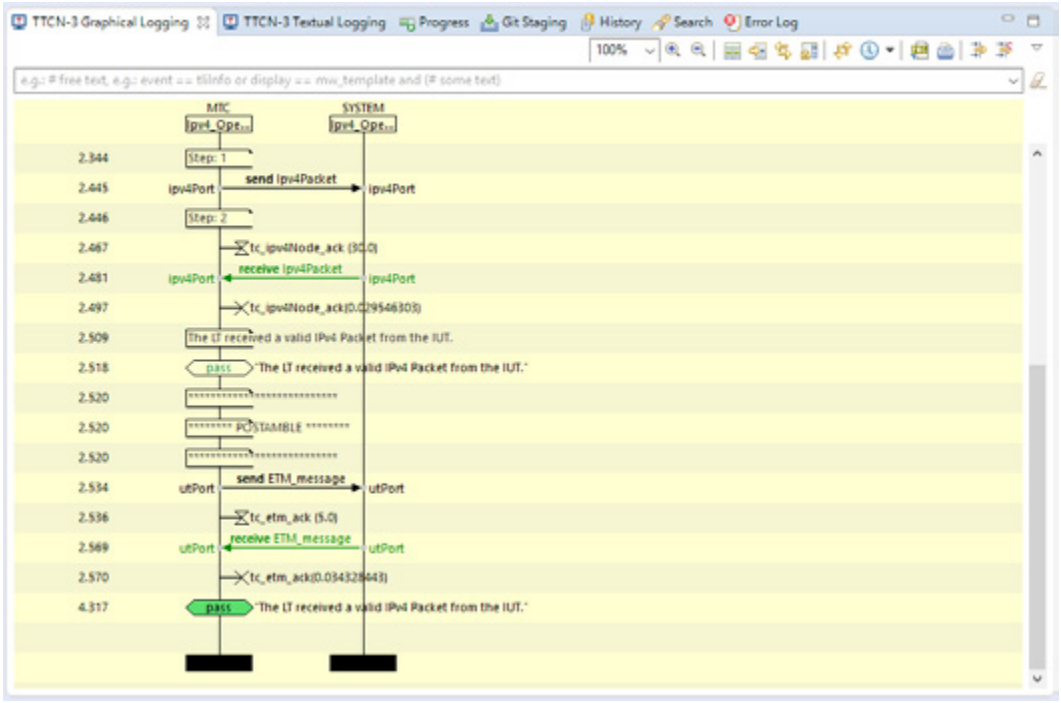


Figure 5. Successful IPv4 test case execution.

The user can scroll through the received messages and check them in detail if desired.

TIP: General information about the execution environment, how to execute test cases, how to read and interpret the graphical logging and much more can be found in the Help menu under: **Help -> Help Contents -> Spirent TTworkbench User's Guide -> Using TTworkbench TTman**

If something went wrong, the verdict will be ***fail***, also followed by a verdict reason. In the next section, the most common reasons for failure are documented.

b. No messages are received from the DUT

If no messages are received from the DUT there will be no "receive" events shown in the graphical log: no arrows originating from the system component with a text of "receive (…)". The verdict in such a case is ***fail*** and several reasons are listed (e.g. no IPv4 packets were received, etc.). In this case, please check the physical connection (Link LEDs should blink), the correct destination IPv4 and MAC address is used and check the Test Ethernet interface device's IP configuration (VLAN, IP-Subnet) by logging to the C1/C50/C100. Another reason for this failure could be the SUT is misconfigured. Execute the TTsuite-OPEN-ETM to validate if the Upper Tester is responding. Also check the VLAN, the IPv4 and MAC address of SUT.

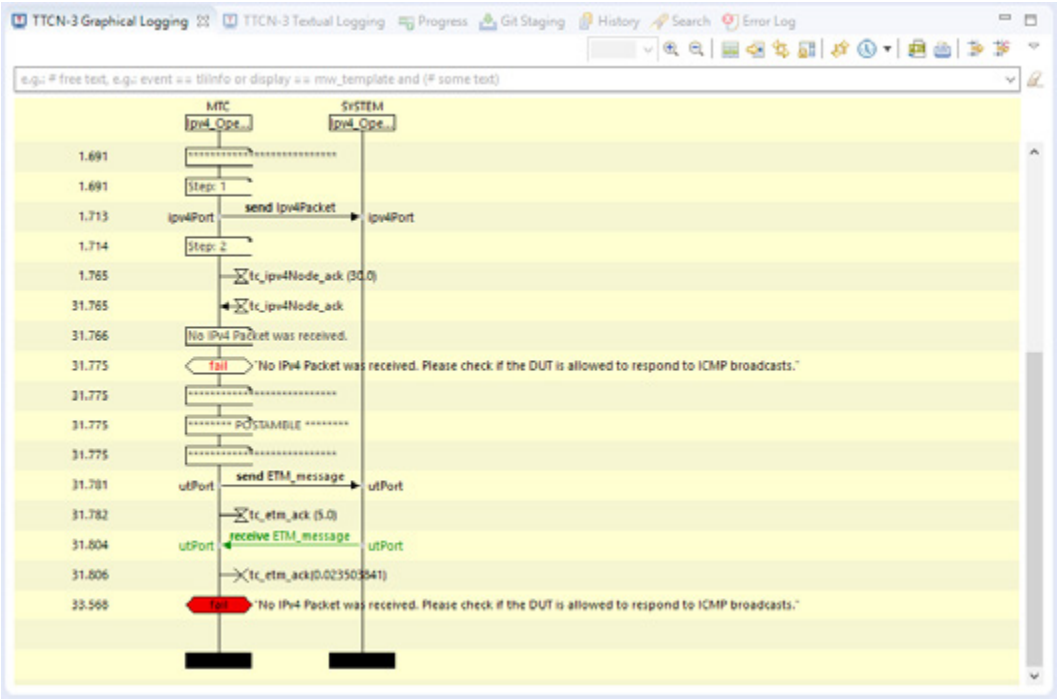


Figure 6. Failed IPv4 test case execution.

2.2. Conformance Testing: TTsuite-OPEN-ARP

2.2.1. Objective

The OPEN-ARP test suite verifies the DUT`s resolution of the Internet layer addresses into link-layer address using OPEN-ARP protocol according to the following standard:

- RFC826 - Ethernet Address Resolution Protocol: Converting Network Protocol Addresses

The representation of the ARP header fields is shown in the figure below:

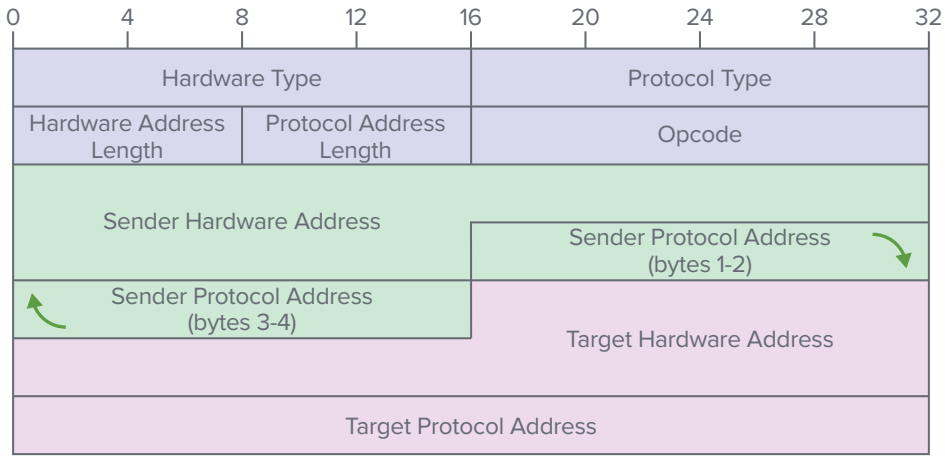


Figure 7. Address Resolution Protocol Header.

The test groups for OPEN-ARP are listed below:

- PacketGeneration – Packet Generation as defined in RFC 826: An Ethernet Address Resolution Protocol, Packet Generation
- PacketReception – Packet Reception as defined in RFC 826: An Ethernet Address Resolution Protocol, Packet Reception

2.2.2. Test Topology

DUT's interface has proper MAC address and connected to the tester interface.

If an IP interface is used, ARP packet exchange must be applicable between the tester and the DUT's interface.

The DUT must have only one IP interface.

For most of the test cases, the execution of Upper Tester commands like clearing the ARP cache, adding static address and sending ICMPv4 Echo Request is required to configure the DUT prior to or during ARP message exchange. As there exists no explicit Upper Tester implementation on the DUT, remote command execution is mandatory. Therefore, either automatic (via SSH if available) or manual remote command execution is eligible and can be configured via module parameters as seen below.

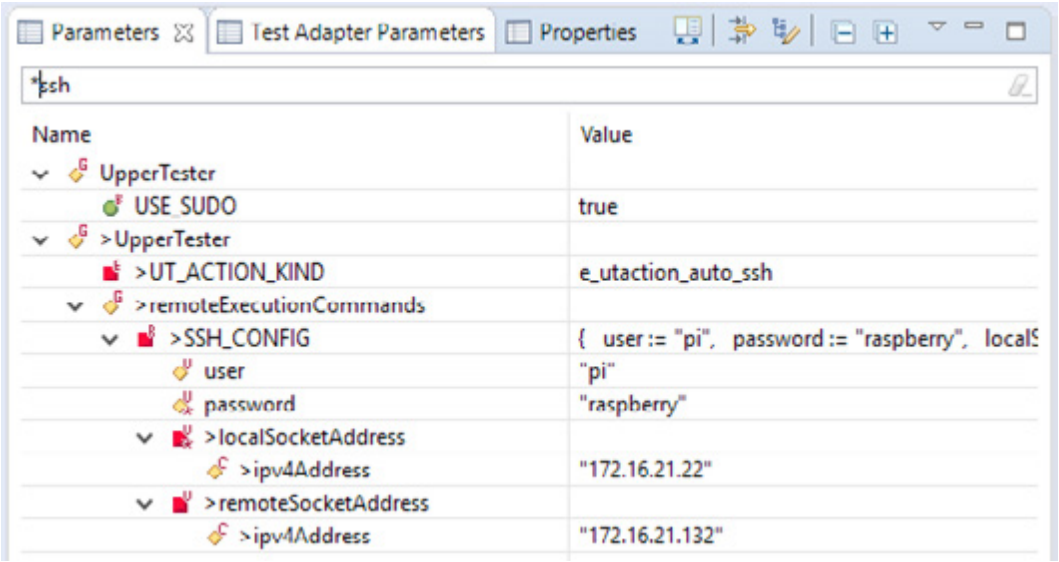


Figure 8. ARP Upper Tester configuration.

The Upper Tester command execution kind can be configured by changing parameter UT_ACTION_KIND.

If custom commands are required to trigger the expected behavior on the DUT, manual remote command execution must be chosen. The SSH configuration can then be ignored. To manually trigger the above listed commands, choose e_utaction_manual_gui as UT_ACTION_KIND and follow the instructions of a pop-up window during test execution (see Figure 6).

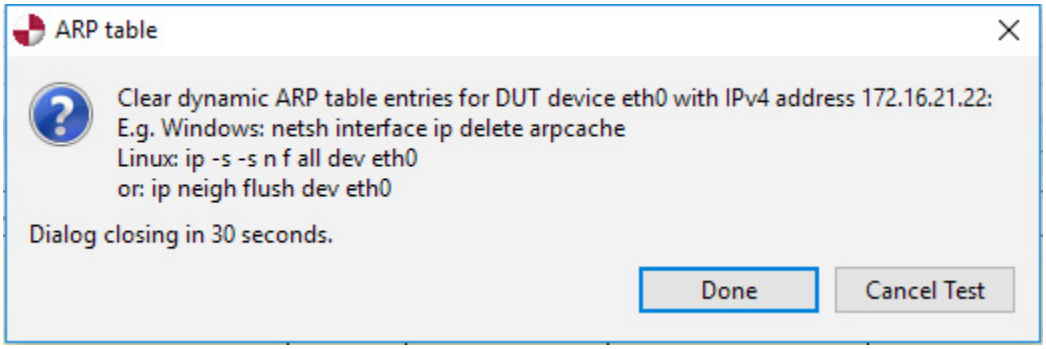


Figure 9. Action window to manually execute Upper Tester commands.

To configure the IP interface, just specify any information available about the connected network interfaces like seen below. It is possible to provide the interface name, its MAC and IP addresses.

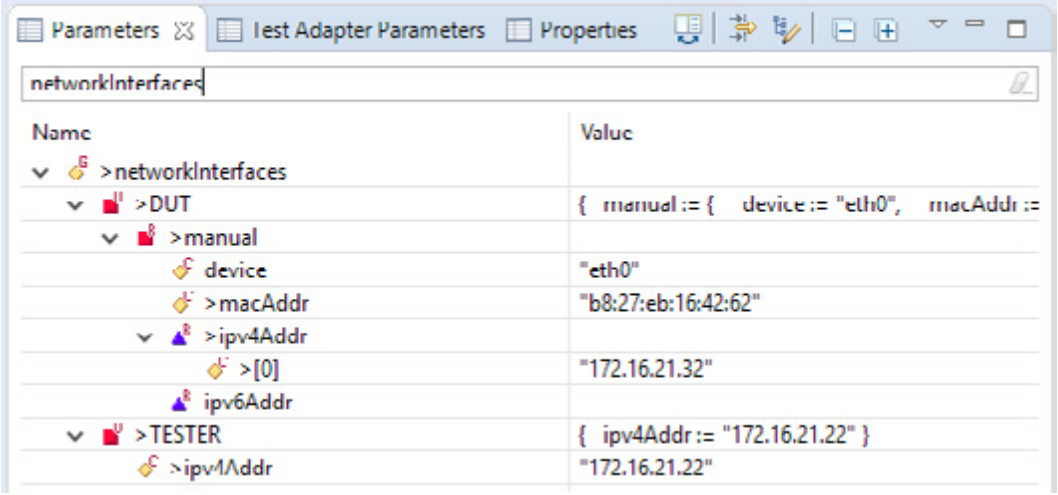


Figure 10. Network Interface Specification for the TS (TESTER) and DUT.

2.2.3. Test Suite Configuration

The test suite parameters can be configured in the *Parameters* view of the *TTCN-3 Execution Management* perspective.

2.2.3.1. Upper Tester Configuration

For most of the test cases, the execution of Upper Tester commands like clearing the ARP cache, adding static address and sending ICMPv4 Echo Request is required to configure the DUT prior to or during ARP message exchange. As there exists yet no explicit Upper Tester implementation on the DUT, remote command execution is mandatory. Therefore, either automatic (via SSH if available) or manual remote command execution is eligible and can be configured via module parameters as seen below.

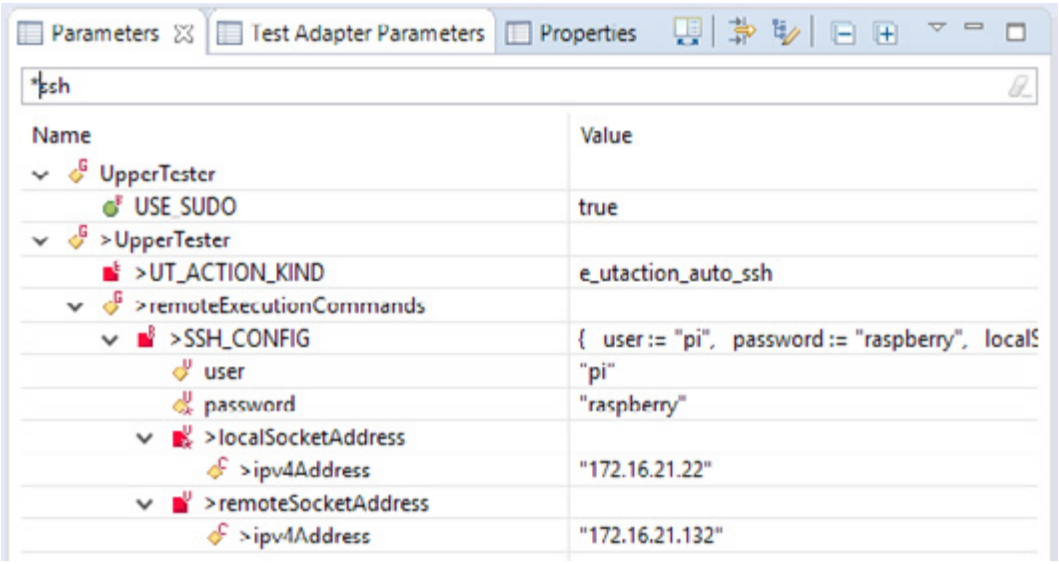


Figure 11. ARP Upper Tester Configuration.

The Upper Tester command execution kind can be configured by changing parameter UT_ACTION_KIND.

If custom commands are required to trigger the expected behavior on the DUT, manual remote command execution must be chosen. The SSH configuration can then be ignored. To manually trigger the above listed commands, choose e_utaction_manual_gui as UT_ACTION_KIND and follow the instructions of a pop-up window during test execution (see Figure 6).

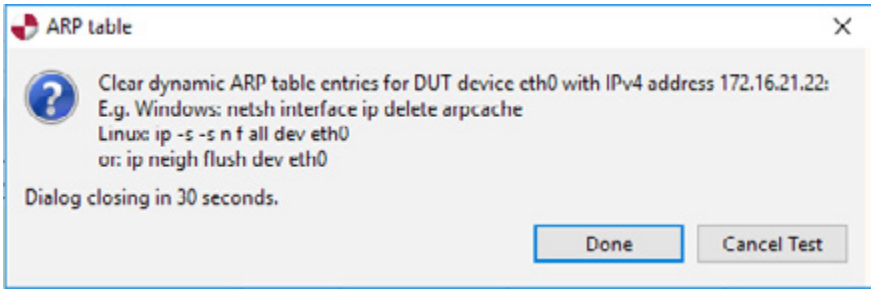


Figure 12. Action window to manually execute Upper Tester commands.

2.2.3.2. IP Interface Configuration

To configure the IP interface, just specify any information available about the connected network interfaces like seen below. It is possible to provide the interface name, its MAC and IP addresses.

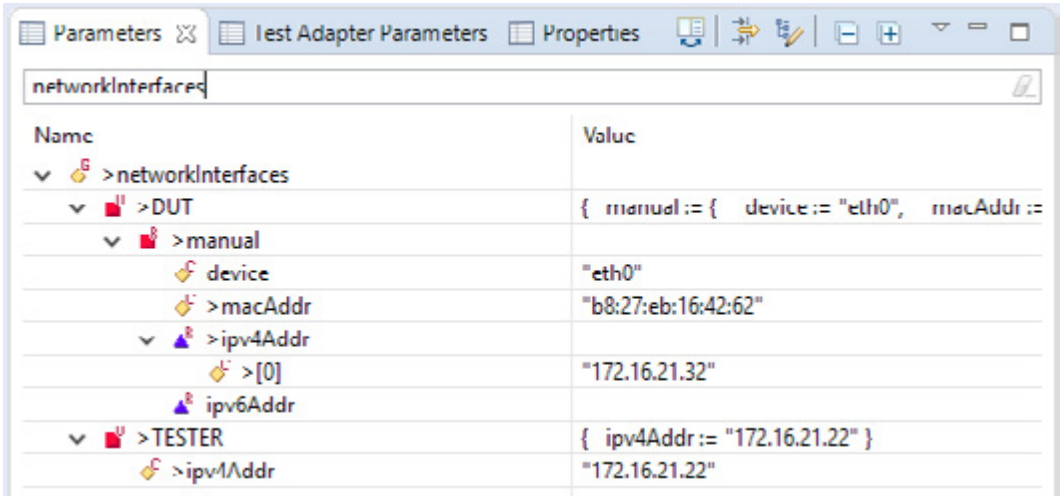


Figure 13. Network Interface Specification for the TS (TESTER) and DUT.

2.2.3.3. Test Suite Specific Configuration

If a test case does not pass, consider changing the timer values listed below and run the test case again.

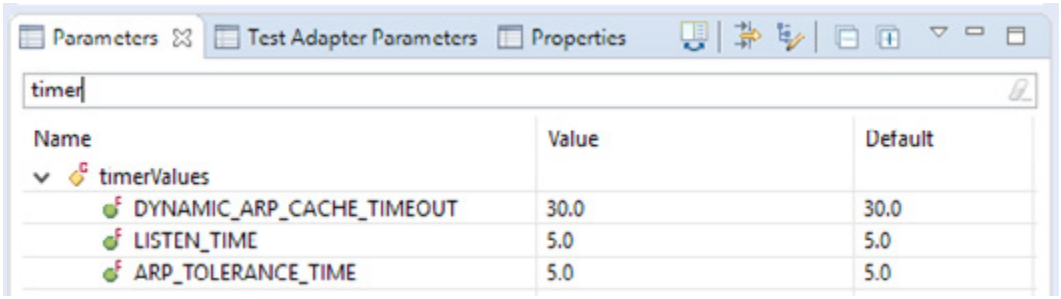


Figure 14. ARP timer values.

You may also have to change the default values for the parameters ARBIT_MAC and FIRST_UNUSED_IP_DIFACE_0 to suite your needs.

2.2.3.4. Capture Configuration

To capture incoming and outgoing traffic, the parameter PX_CAPTURE_CONFIGURATION is provided. Enabling it with a custom filter as seen below stores a pcap file for further analysis in the specified location for each executed test case. By default, the configuration captures all ARP messages. Changing the filter has no effect on the actual test case execution.

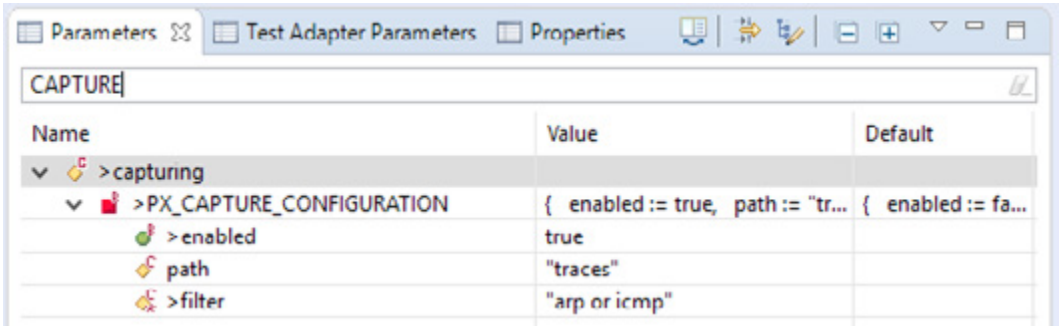


Figure 15. ARP capture configuration.

2.2.4. Expected Behavior

If the DUT`s network addresses and its ARP cache table is well-configured, execute the test cases by double clicking on the test case or clicking on the run button. The test case`s verdict should be ***pass*** and shown in green.

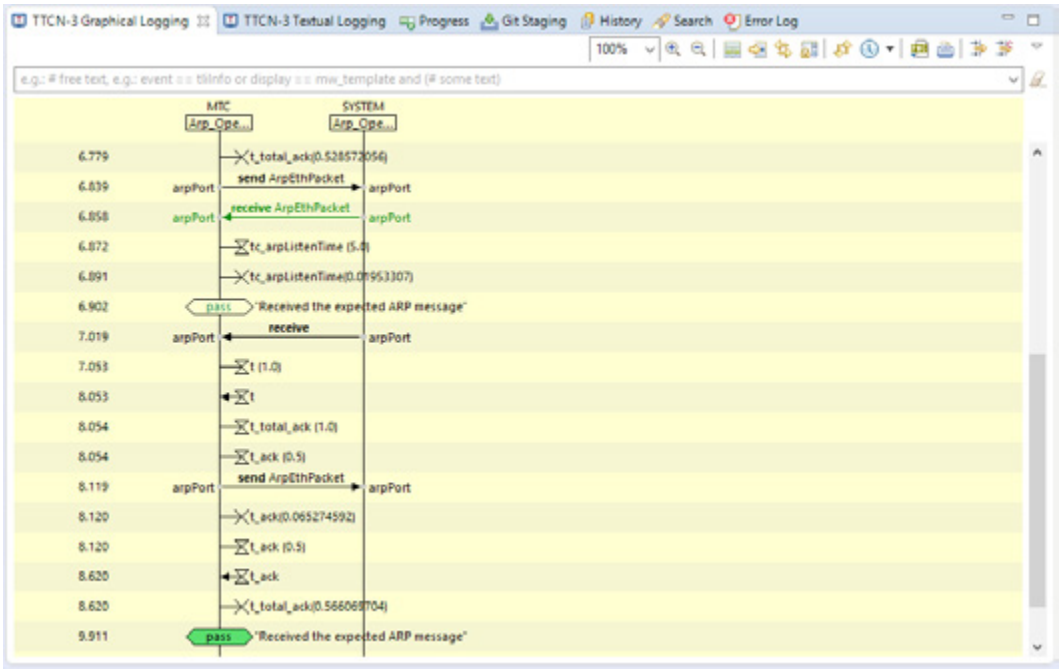


Figure 16. Successful ARP test case execution.

Sent and received ARP messages as well as the underlying Ethernet stack address can be analyzed in the Test Data View. In case any ARP header value does not match its expected template value, it is marked as red.

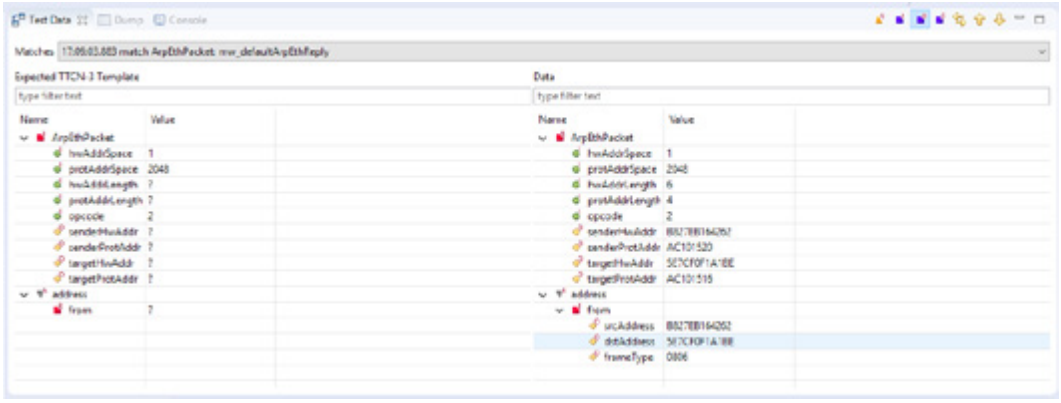


Figure 17. Example template match of a received ARP message.

2.3. Conformance Testing: TTsuite-OPEN-IPv4-AUTOConf

2.3.1. Objective

OPEN-TTsuite-IPv4AutoConf test suite verifies the DUT`s Dynamic IPv4 address Auto Configuration protocol according to the following standard:

- RFC 3927 - Dynamic Configuration of IPv4 Link-Local Addresses

2.3.2. Test Topology

The DUT should have an IP network interface which supports IPv4 AutoConf or DHCP. The IP addresses of all the emulated servers and their address pool should be obtainable from the tester.

The test groups for OPEN-IPv4-AutoConf are listed below:

- IPv4 AutoConf Introduction – When to configure a link local address as defined in RFC 3927: Dynamic Configuration of IP Link-Local Addresses
- IPv4 AutoConf Address Selection – Address Selection Defense and Delivery as defined in RFC 3927: Dynamic Configuration of IP Link-Local Addresses
- IPv4 AutoConf Announcing – Announcing an address as defined in RFC 3927: Dynamic Configuration of IP Link-Local Addresses
- IPv4 AutoConf Conflict – Conflict Detection and Defense as defined in RFC 3927: Dynamic Configuration of IP Link-Local Addresses
- IPv4 AutoConf Forwarding – Forwarding Rules as defined in RFC 3927: Dynamic Configuration of IP Link-Local Addresses
- IPv4 AutoConf Link Local Packets – Link Local Packets are Not Forwarded as defined in RFC 3927: Dynamic Configuration of IP Link-Local Addresses
- IPv4 AutoConf Routable Address – Interaction with Host with Routable Address as defined in RFC 3927: Dynamic Configuration of IP Link-Local Addresses
- IPv4 AutoConf Network Partitions – Healing of Network Partitions as defined in RFC 3927: Dynamic Configuration of IP Link-Local Addresses

The IP interface and Upper Tester configuration is similar to TTsuite-OPEN-ARP.

2.3.3. Test Suite Configuration

Test suite parameters can be configured in the *Parameters* view of the *TTCN-3 Execution Management* perspective.

2.3.3.1. Upper Tester Configuration

The Upper Tester configuration is like TTsuite-OPEN-ARP, see chapter 2.2.3.1. All test cases require a valid Upper Tester configuration for executing DHCP, ARP and ICMPv4 commands on the DUT.

2.3.3.2. IP Interface Configuration

To configure the IP interface, just specify any information available about the connected network interfaces. It is possible to provide the interface name, its MAC and IP addresses. The corresponding module parameters are HOST_1 for the test system and respectively DIFACE_0 for the DUT. For an example configuration, see chapter 2.2.3.2. The DUT needs to be preconfigured to have a link-local IPv4 address.

2.3.3.3. Test Suite Specific Configuration

Below you can see a sample configuration of this test suite. It combines all DHCP and ARP module parameters like timers, IPv4 addresses and MAC addresses. If a test case does not pass, consider changing these values and run the test case again.

Parameters | Test Adapter Parameters | Properties

type filter text

Name	Value	Default
UDP_SRC_PORT	67	67
UDP_DST_PORT	68	68
LISTEN_TIME	10.0	10.0
LEASE_TIME	15	15
RENEWAL_TIME	7	7
REBINDING_TIME	13	13
BROADCAST_IP	"255.255.255.255"	"255.255.255.255"
MAX_PACKET_SIZE	576	576
SERVER_1_IP_POOL	{ addresses := { { "172.1...	{ addresses := ...
addresses		
[0]		
[0]	"172.16.21.40"	
[1]	"172.16.21.41"	
[1]		
netmask	"255.255.255.0"	
SERVER_2_IP_POOL	{ addresses := { { "172.1...	{ addresses := ...
addresses		
[0]		
[0]	"172.16.22.40"	
[1]	"172.16.22.41"	
[1]		
netmask	"255.255.255.0"	
SERVER_1	"Server-1"	"Server-1"
HOST_MASK	"255.255.255.255"	"255.255.255.255"
ROUTABLE_IP_ADDR_1	"169.254.50.3"	"169.254.50.3"
ROUTABLE_IP_ADDR_2	"169.254.100.4"	"169.254.100.4"
DIFACE_0_IP_LINKLOCAL_ADDR	"169.254.50.4"	"169.254.50.4"
AIFACE_0_IP_LINKLOCAL_ADDR	"169.254.50.3"	"169.254.50.3"
ARBITRARY_IP_LINKLOCAL_ADDR	"169.254.10.10"	"169.254.10.10"
PARAM_PROCESS_TIME	10.0	10.0
PARAM_TOLERANCE_TIME	5.0	5.0
PARAM_LISTEN_TIME	15.0	15.0
MAC_ADDR_1	'001122334455'O	'001122334455'O
MAC_ADDR_2	'66778899AABB'O	'66778899AABB...
LINK_LOCAL_NET_ADDR	"169.254.0.0"	"169.254.0.0"
PROBE_MIN	1.0	1.0
PROBE_MAX	2.0	2.0
ANNOUNCE_WAIT	2.0	2.0
ANNOUNCE_INTERVAL	2.0	2.0
RATE_LIMIT_INTERVAL	60.0	60.0
DEFEND_INTERVAL	10.0	10.0
LOCAL_BROADCAST_IP	"169.254.255.255"	"169.254.255.255"

Figure 18. IPv4AutoConf Parameter Configuration.

2.3.3.4. Capture Configuration

To capture incoming and outgoing traffic, the parameter PX_CAPTURE_CONFIGURATION is provided. Enabling it with a custom filter as seen below stores a pcap file for further analysis in the specified location for each executed test case. By default, the configuration captures all DHCP, ARP and ICMPv4 messages. Changing the filter has no effect on the actual test case execution.

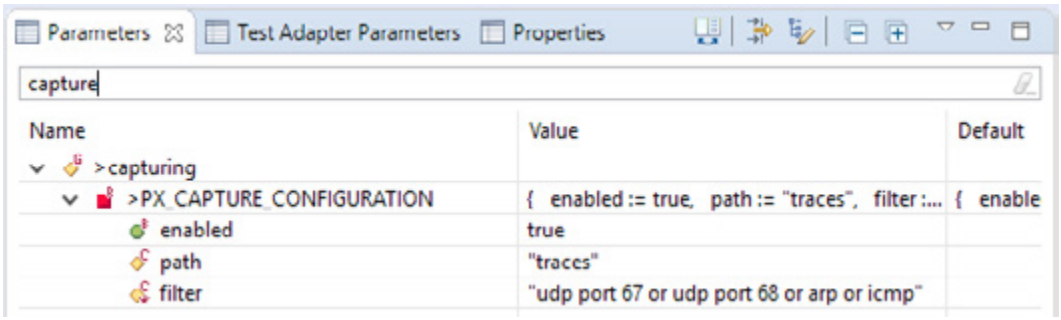


Figure 19. IPv4AutoConf Capture Configuration.

2.3.4. Expected Behavior

After a DHCP client is configured on the DUT's IP interface, execute the test cases by double clicking on the corresponding test case or click on the run button. The test verdict should be ***pass*** and shown in green.

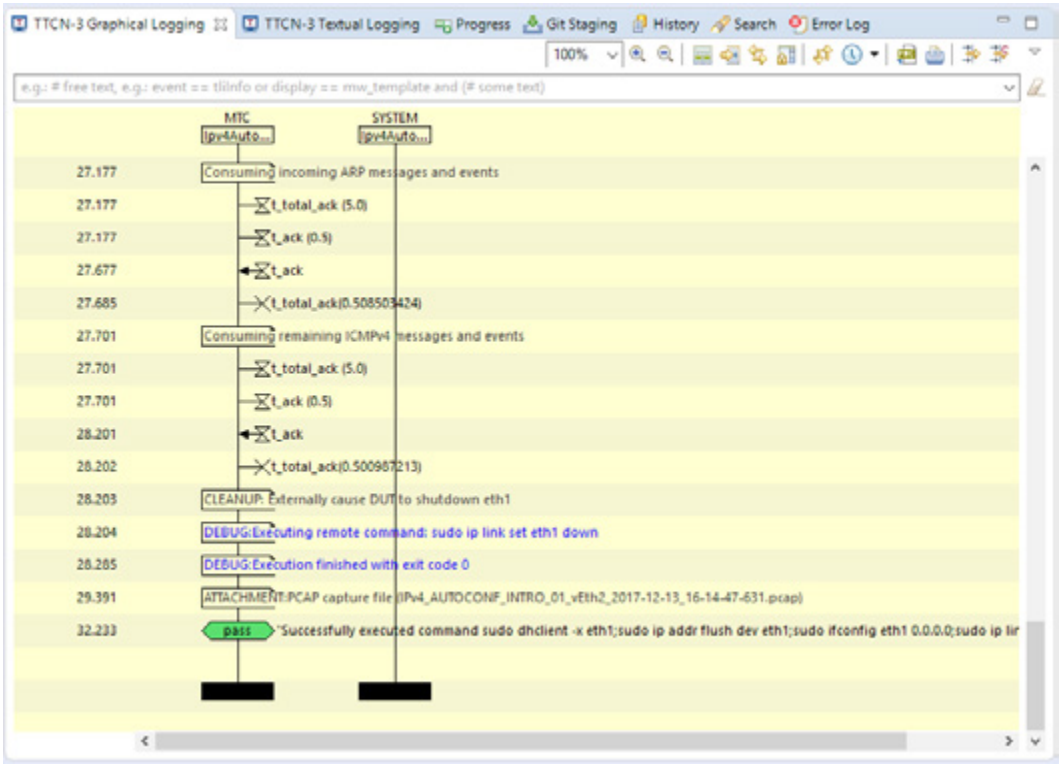


Figure 20. Successful IPv4 AutoConf test case execution.

2.4. Conformance Testing: TTsuite-OPEN-DHCPv4

2.4.1. Objective

OPEN-DHCPv4 test suite verifies the DUT's Dynamic Host Configuration Protocol Version 4 communication according to the following standards:

- RFC 2131 - Dynamic Host Configuration Protocol
- RFC 2132 - DHCP Options and BOOTP Vendor Extensions

The DHCPv4 header format is shown in the figure below:

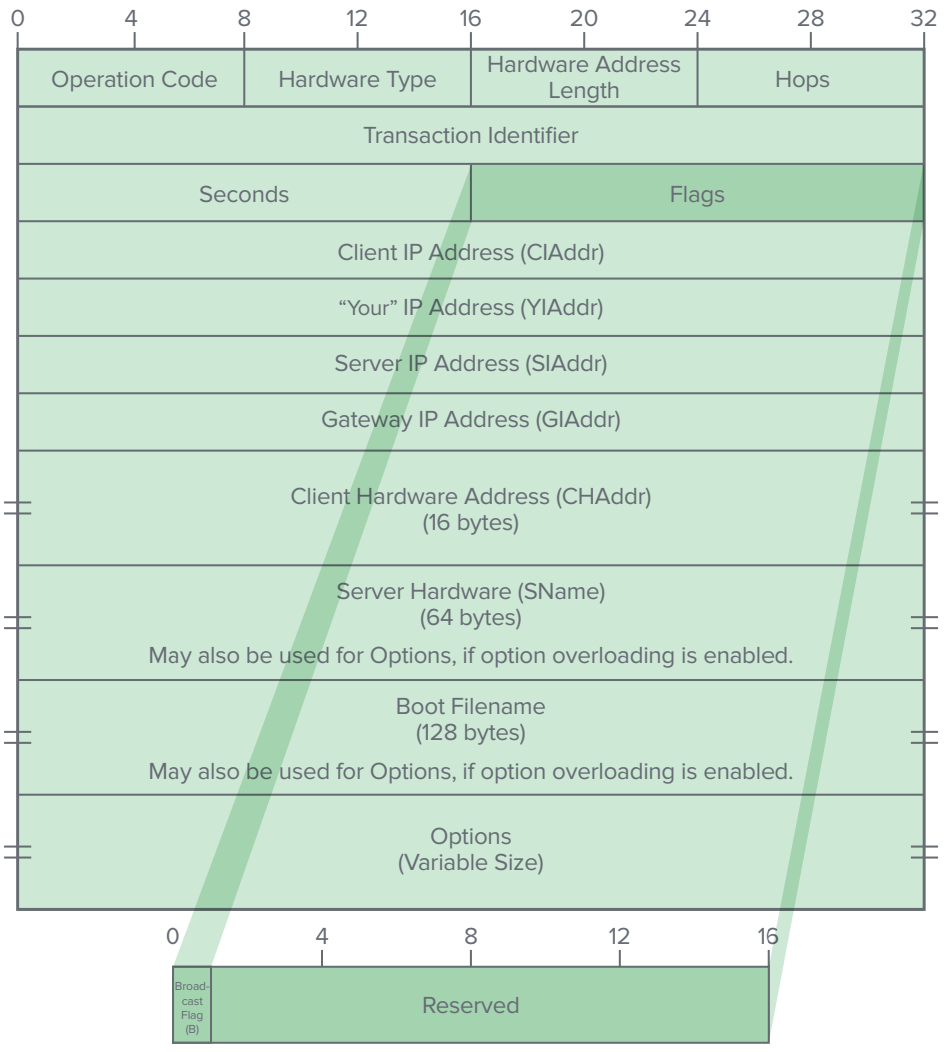


Figure 21. DHCPv4 Header.

The test groups for the OPEN-DHCPv4 test suite are listed below:

- DHCPv4 Client Summary - Design Goals and Protocol Summary as defined in RFC2131: Dynamic Host Configuration Protocol
- DHCPv4 The Client Protocol - The Client-Server Protocol as defined in RFC2131: Dynamic Host Configuration Protocol
- DHCPv4 Client Allocating - Client-server interaction, allocating a network address as defined in RFC2131: Dynamic Host Configuration Protocol
- DHCPv4 Client Reusing - Client-server interaction - reusing a previously allocated network address as defined in RFC2131: Dynamic Host Configuration Protocol
- DHCPv4 Client Parameters: Client parameters in DHCP as defined in RFC2131: Dynamic Host Configuration Protocol
- DHCPv4 Client Usage: DHCP Usage as defined in RFC2131: Dynamic Host Configuration Protocol
- DHCPv4 Client Constructing Messages - Constructing and sending DHCP messages as defined in RFC2131: Dynamic Host Configuration Protocol
- DHCPv4 Client Request: DHCPREQUEST message as defined in RFC2131: Dynamic Host Configuration Protocol
- DHCPv4 Client Initialization Allocation - Initialization and allocation of network address as defined in RFC2131: Dynamic Host Configuration Protocol
- DHCPv4 Client Initialization External - Initialization with an externally assigned network address
- DHCPv4 Client Reacquisition - Reacquisition and expiration message as defined in RFC2131: Dynamic Host Configuration Protocol

2.4.2. Test Topology

Depending on which topology is used from the 4 topologies, varied numbers of DUT interfaces and emulated DHCP servers are used. The requirements of the corresponding topologies are shown in the table below:

Topology No.	No. of DUT IP Interfaces	No. of Emulated Network Devices
Topology-1	1	1 Server
Topology-2	2	1 Server
Topology-3	1	2 Servers
Topology-4	1	1 Server 1 Static IP assigned non-DHCP Client

Topology-1: An emulated DHCP Server is connected to DUT`s IP interface.

Topology-2: An emulated DHCP Server is connected to DUT`s two different IP interfaces.

Topology-3: Two emulated DHCP server are connected to DUT`s IP interface via an emulated HUB.

Topology-4: An emulated DHCP Server and an emulated Non-DHCP Client with Static IP are connected to DUT`s IP interface via an emulated HUB.

The IP addresses of the emulated network devices and the IP address pool offered by the emulated DHCP servers must be obtainable. Unless the DHCP client must be pre-configured with a static IP address, no IP address should be pre-configured on the DUT's interface as the emulated DHCP server will eventually offer one from its configurable IP address pool.

The IP interface and Upper Tester configuration is similar to TTsuite-OPEN-ARP.

2.4.3. Test Suite Configuration

The test suite parameters can be configured in the *Parameters* view of the *TTCN-3 Execution Management* perspective.

2.4.3.1. Upper Tester Configuration

The Upper Tester configuration is like TTsuite-OPEN-ARP, see chapter 2.2.3.1. All test cases require a valid Upper Tester configuration for executing DHCP, ARP and ICMPv4 commands on the DUT.

2.4.3.2. IP Interface Configuration

To configure the IP interface, just specify any information available about the connected network interfaces. It is possible to provide the interface name, its MAC and IP addresses. The corresponding module parameters are TESTER_SERVER_1 and TESTER_SERVER_2 for the test system and respectively DUT_CLIENT_1 and DUT_CLIENT_2 for the DUT. For an example configuration, see chapter 2.2.3.2. To ensure correct behavior, verify that all DHCP client interfaces on the DUT do not have a preconfigured IPv4 address as the DHCP server emulated by the test system will assign an IPv4 address from its IPv4 address pool.

2.4.3.3. Test Suite Specific Configuration

Below you can see a sample configuration of this test suite. It contains all DHCP module parameters like timers, address pools and MAC addresses. If a test case does not pass, consider changing these values and run the test case again.

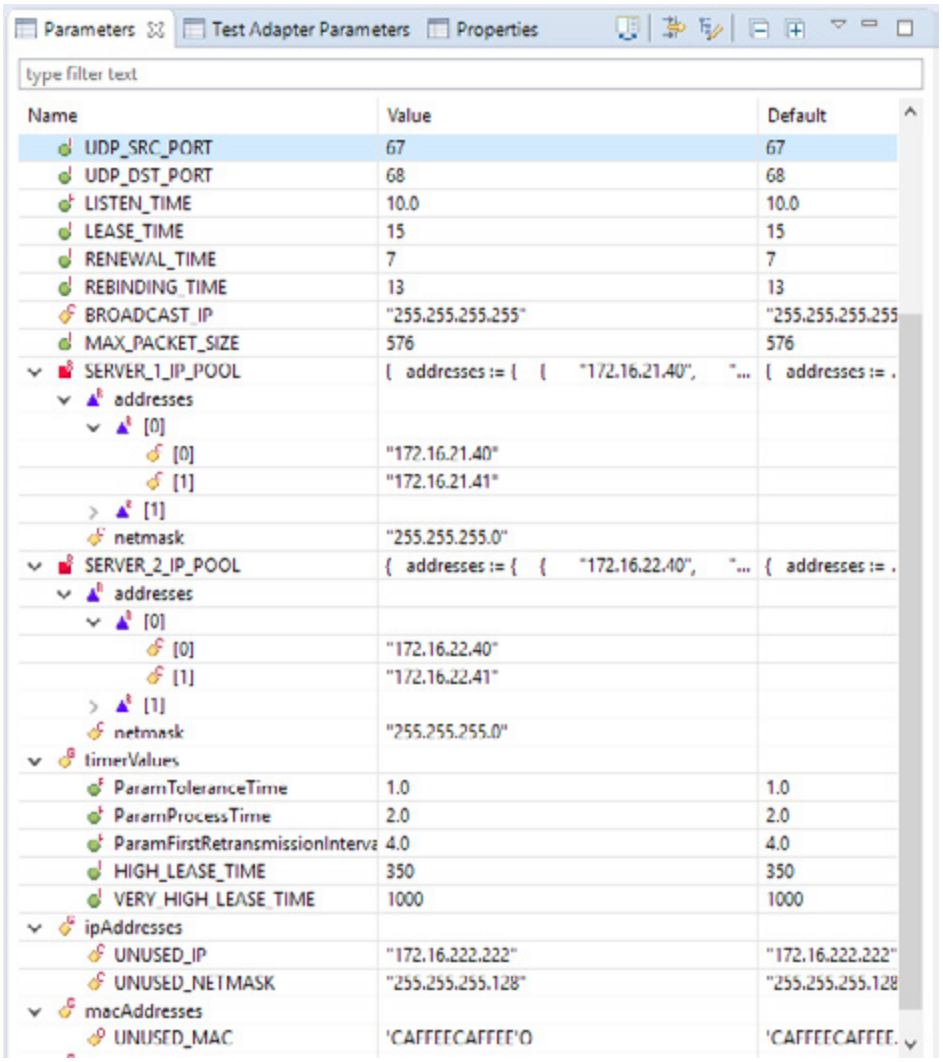


Figure 22. DHCP Parameter Configuration.

2.4.3.4. Capture Configuration

To capture incoming and outgoing traffic, the parameter PX_CAPTURE_CONFIGURATION is provided. Enabling it with a custom filter as seen below stores a pcap file for further analysis in the specified location for each executed test case. By default, the configuration captures all DHCP, ARP and ICMPv4 messages. Changing the filter has no effect on the actual test case execution.

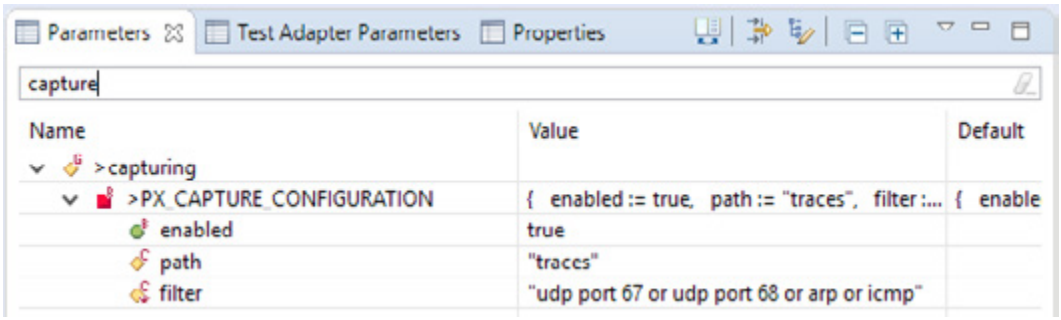


Figure 23. DHCP Capture Configuration.

2.4.4. Expected Behavior

After the DUT`s IP interfaces and the emulated network devices are configured, the test cases can be executed by double clicking on the corresponding test case or clicking on the run button. The test verdict should be ***pass*** and shown in green.

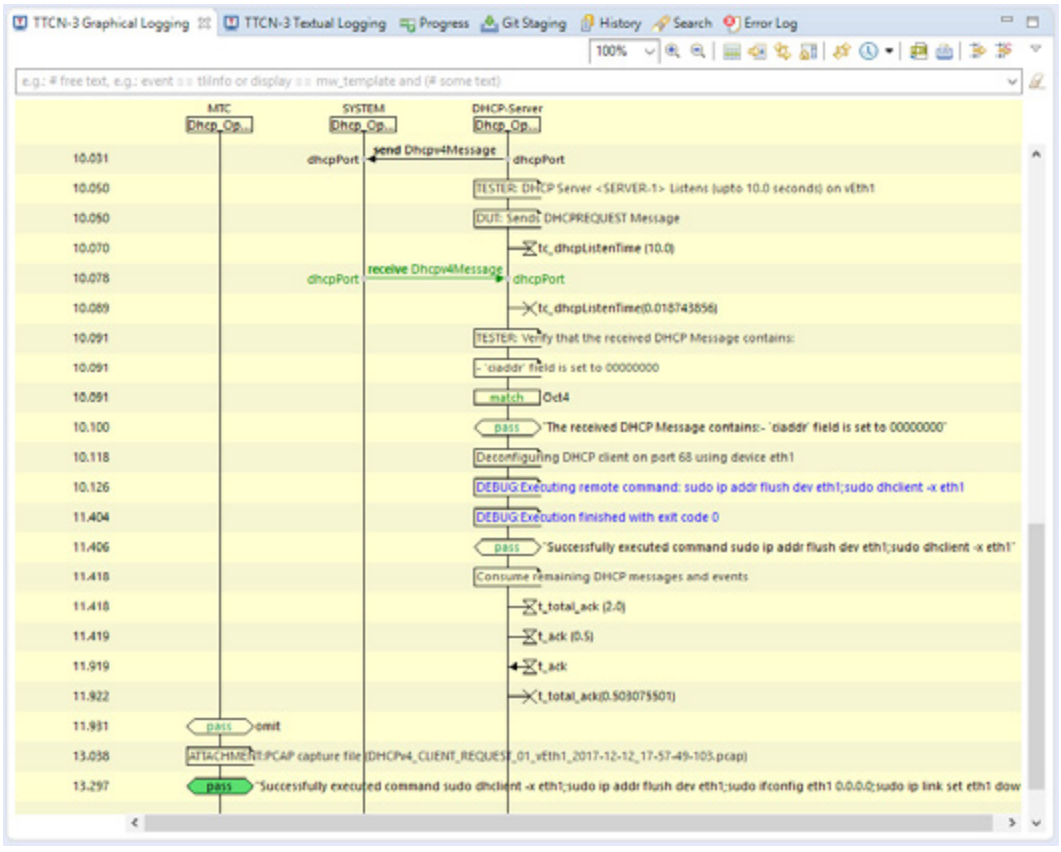


Figure 24. Successful DHCPv4 test case execution.

2.5. Conformance Testing: TTsuite-OPEN-TCP

2.5.1. Objective

The OPEN-TCP test suite verifies the TCP communication of DUT according to the following standards:

- RFC 793 - Transmission Control Protocol" of DARPA, Internet Program, Protocol Specification
- RFC 1122 - Requirement for Internet Hosts–Communication Layers
- RFC 2460 - Internet Protocol, Version 6 (IPv6) Specification

The representation of the TCP Header is shown in the figure below:

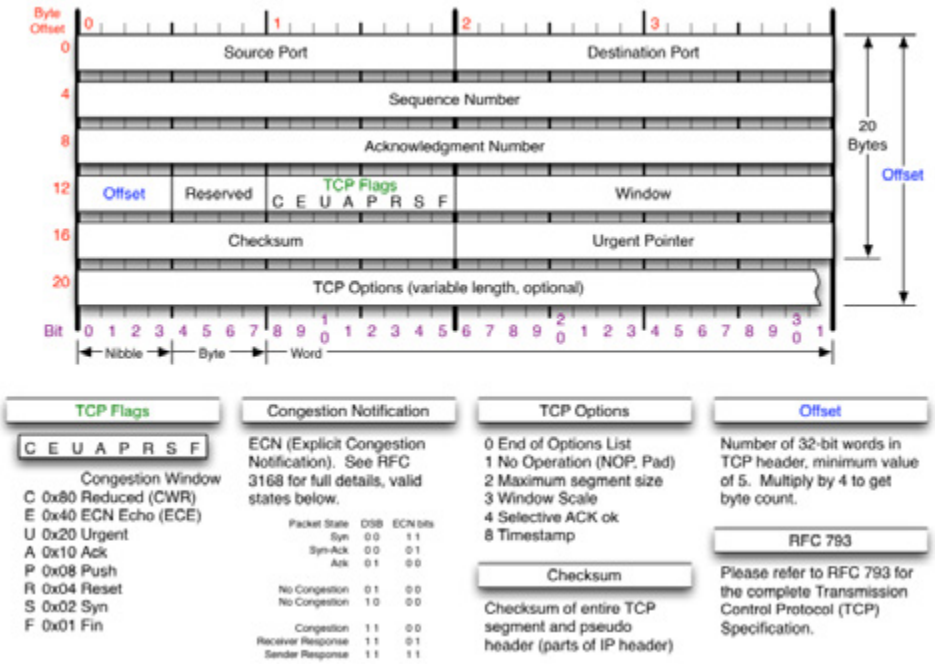


Figure 25. TCP Header.

The test groups for OPEN-TCP are listed below:

- Acknowledgment - Tests as defined by RFC 793
- Avoidance - SWS Avoidance Tests as defined by RFC 793
- Basics - Basic Functionality Tests as defined by RFC 793
- CallAbort - Abort Call Tests as defined by RFC 793
- CallClose - Close Call Tests as defined by RFC 793
- CallOpen - Open Call Tests as defined by RFC 793
- CallReceive - Receive Tests as defined by RFC 793
- CallSend - Send Tests as defined by RFC 793
- Checksum - Checksum Tests as defined by RFC 793
- Closing - Connection Closing Tests as defined by RFC 793
- ConnectionEstablish - Establish Connection Tests as defined by RFC 793
- KeepAlive - Keep-Alive Connection Tests as defined by RFC 793
- ControlFlags - Control Flags Tests as defined by RFC 793
- FastRetransmit - Fast retrasmissing Tests as defined by RFC 793
- FlagsInvalid - Invalid Flags Tests as defined by RFC 793
- FlagsProcessing - Processing Flags Tests as defined by RFC 793
- FlagsPush - Push Flag Tests as defined by RFC 793
- Header - TCP Header Tests as defined by RFC 793
- HostSpec - Host Related Tests as defined by RFC 793
- ImprovedWindowing - Window Size Change Tests as defined by RFC 793
- LayerActions - Layer Actions Tests as defined by RFC 793
- MSS_Options - MSS Options Tests as defined by RFC 793
- Nagle - Nagle Algorithm Tests as defined by RFC 793
- OutOfOrder - TCP segment ordering Tests as defined by RFC 793
- ProbingWindows - TCP receive window Tests as defined by RFC 793
- Retransmission - TCP segment retransmission Tests as defined by RFC 793
- Sequence - TCP segment Sequence Tests as defined by RFC 793
- Shutdown - Shutdown TCP Socket Tests as defined by RFC 793
- SlowstartCongestion - Slow start congestion Tests as defined by RFC 793
- Unacceptable - Unacceptable TCP segments Tests as defined by RFC 793
- UrgentPtr - Urgent Pointer Tests as defined by RFC 793
- WindowSize - Window Size Tests as defined by RFC 793

2.5.2. Test Topology

For most of the test cases, the test topology consists of one test interface connected to the DUT via Ethernet. The Upper Tester application runs on DUT using a predefined port.

2.5.3. Test Suite Configuration

The test suite parameters can be configured in the *Parameters* view of the *TTCN-3 Execution Management* perspective.

2.5.3.1. Upper Tester Configuration

For some test cases, another type of communication is needed with the IUT to trigger required behavior on the IUT, or to send different type of messages, or check the IUT's state and the received messages. This functionality is accomplished by using a separate UDP port.

Please contact our sales team "TTworkbench Sales" automotive.sales.support@spirent.com if you need an Upper Tester or integration support.

The following procedures can be executed:

- openTCPSocket(typeOfSocket): open two types of TCP socket on the IUT:
  - Passive: opens a socket with a receiving call.
  - Active: opens a socket with a sending call.
- openMultipleTCPSocket(typeOfSocket, numberOfSockets): open <numberOfSockets> TCP sockets of <typeOfSocket> on the IUT.
- type.closeTCPSocket: Close the TCP socket(s) on the IUT.

Confirmation: a confirmation message is sent via the Upper Tester channel in response to the tester`s procedure call. It can be a Boolean value only (for example success) or, a Boolean with some additional information depending on the required operation from the IUT.

To install the Upper Tester to the DUT execute the following steps:

1. Unpack the latest package tar xzvf utpackage_yyyymmdd.tgz
2. cd utpackage
3. sudo ./uninstall.sh
4. sudo ./install.sh

To make sure that everything is fine, type '!ps' command. The following screen should be displayed:

```
pi@ber-autopi1:~ $ !ps
ps -ef | grep UpperTester
root      3948     1  0 Sep11 ?        00:00:27 /etc/init.d/ut/UpperTesterIPv4 5001
root      3949     1  0 Sep11 ?        00:00:27 /etc/init.d/ut/UpperTesterIPv6 6001
pi       14461 14395  0 09:23 pts/1    00:00:00 grep --color=auto UpperTester
```

Figure 26. Command execution to check if Upper Tester is running.

2.5.3.2. IP Interface Configuration

To configure the IP interface, just specify any information available about the connected network interfaces. It is possible to provide the interface name, its MAC and IP addresses. The corresponding module parameters are TESTER for the test system and respectively DUT. For an example configuration, see chapter 2.2.3.2.

2.5.3.3. Test Suite Specific Configuration

If a test case does not pass or the DUT has a custom TCP configuration, consider changing the values listed below and run the test case again. Depending on the IP version of the test suite to execute, verify that the module parameter IP_VERSION matches the loaded test suite version to execute.

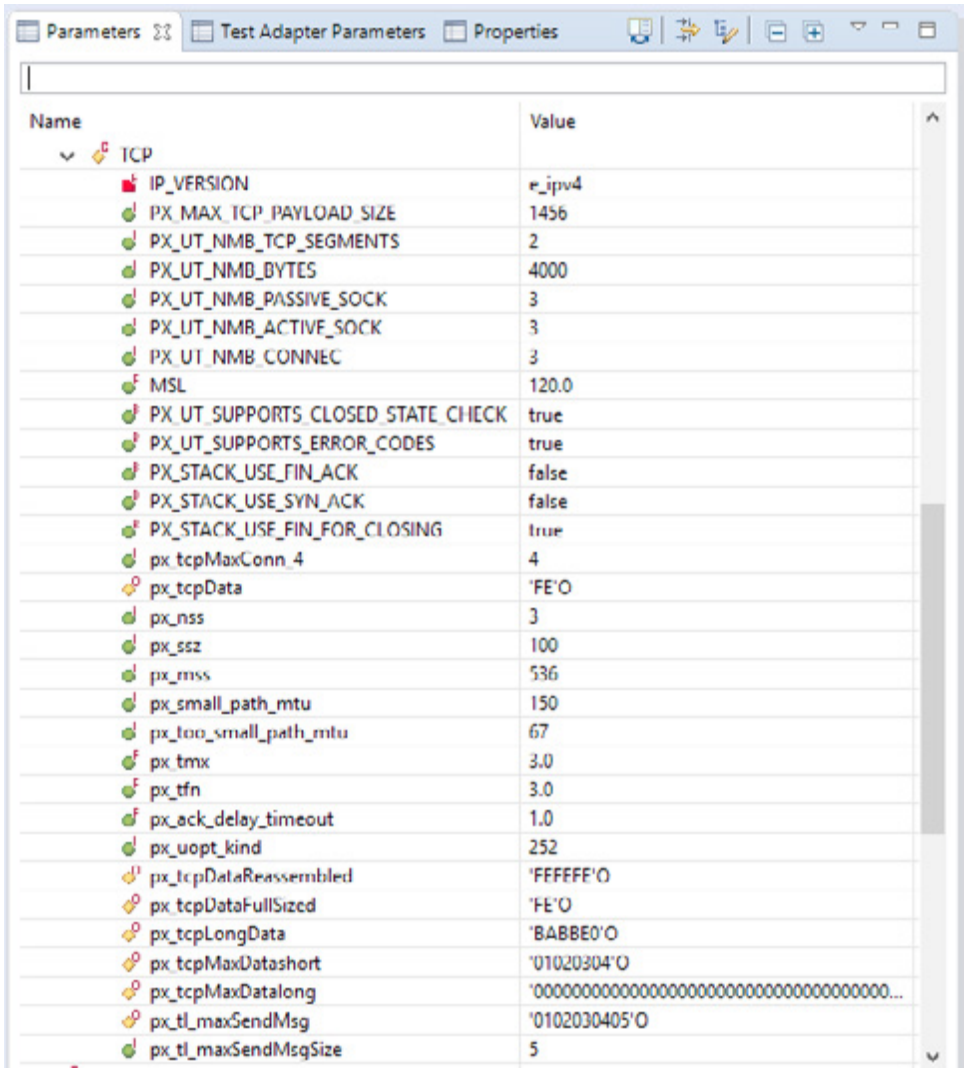


Figure 27. TCP Parameter Configuration.

2.5.3.4. Capture Configuration

To capture incoming and outgoing traffic, the parameter PX_CAPTURE_CONFIGURATION is provided. Enabling it with a custom filter as seen below stores a pcap file for further analysis in the specified location for each executed test case. By default, the configuration captures all TCP, ICMP and Upper Tester messages. Changing the filter has no effect on the actual test case execution. Depending on the used IP version and the TCP port range, consider to also update the capture filter.

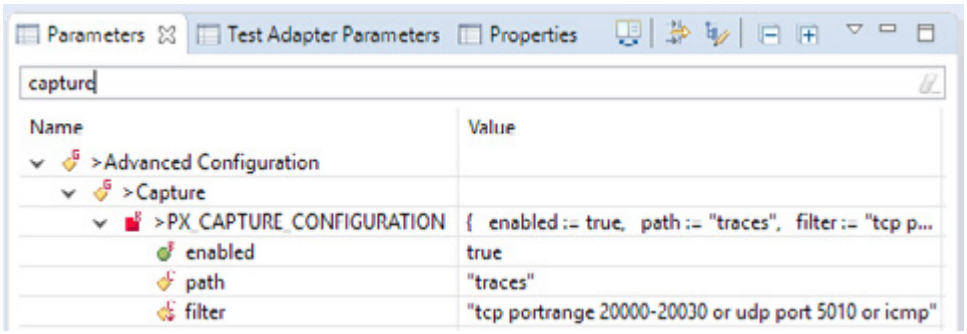


Figure 28. TCP Capture Configuration.

2.5.4. Expected Behavior:

After everything is configured just double click on the test case or push the run button to execute tests. At TCP State Machine Graph menu, the DUT's TCP states and the state changes can be monitored during the test execution. If you click on a different state, the message which triggered the state is shown at the TCP State history tab.

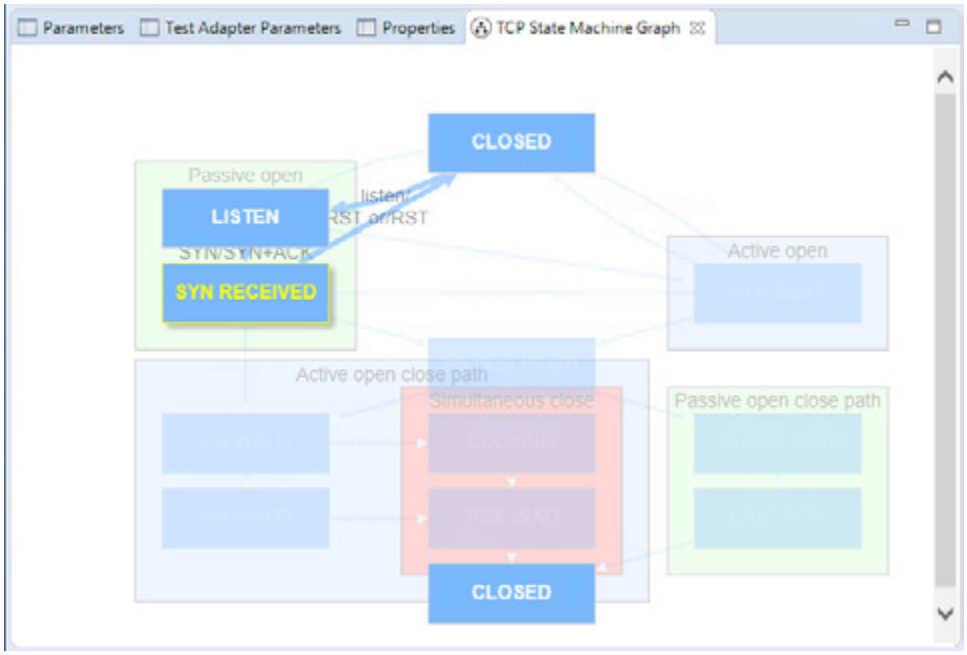
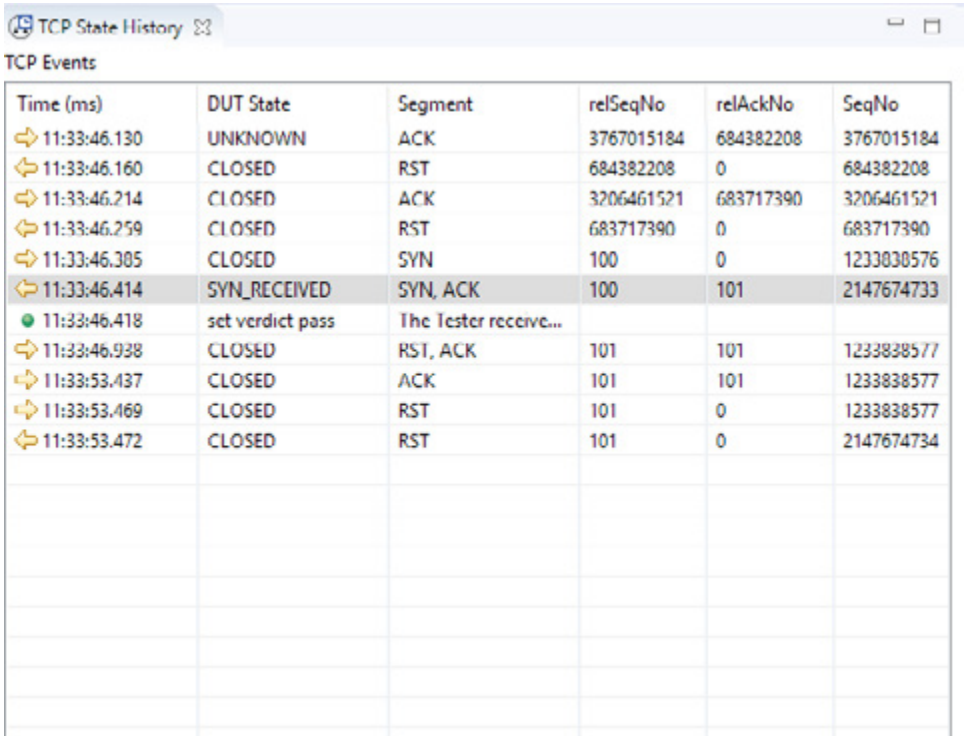


Figure 29. Sample TCP State Machine Graph.



Time (ms)	DUT State	Segment	relSeqNo	relAckNo	SeqNo
11:33:46.130	UNKNOWN	ACK	3767015184	684382208	3767015184
11:33:46.160	CLOSED	RST	684382208	0	684382208
11:33:46.214	CLOSED	ACK	3206461521	683717390	3206461521
11:33:46.259	CLOSED	RST	683717390	0	683717390
11:33:46.305	CLOSED	SYN	100	0	1233838576
11:33:46.414	SYN_RECEIVED	SYN, ACK	100	101	2147674733
11:33:46.418	set verdict pass	The Tester receive...			
11:33:46.938	CLOSED	RST, ACK	101	101	1233838577
11:33:53.437	CLOSED	ACK	101	101	1233838577
11:33:53.469	CLOSED	RST	101	0	1233838577
11:33:53.472	CLOSED	RST	101	0	2147674734

Figure 30. Live TCP State History for in- and outgoing TCP Events.

If everything was configured properly, the verdict will be ***pass*** and displayed in green color at the Graphical Logging tab.

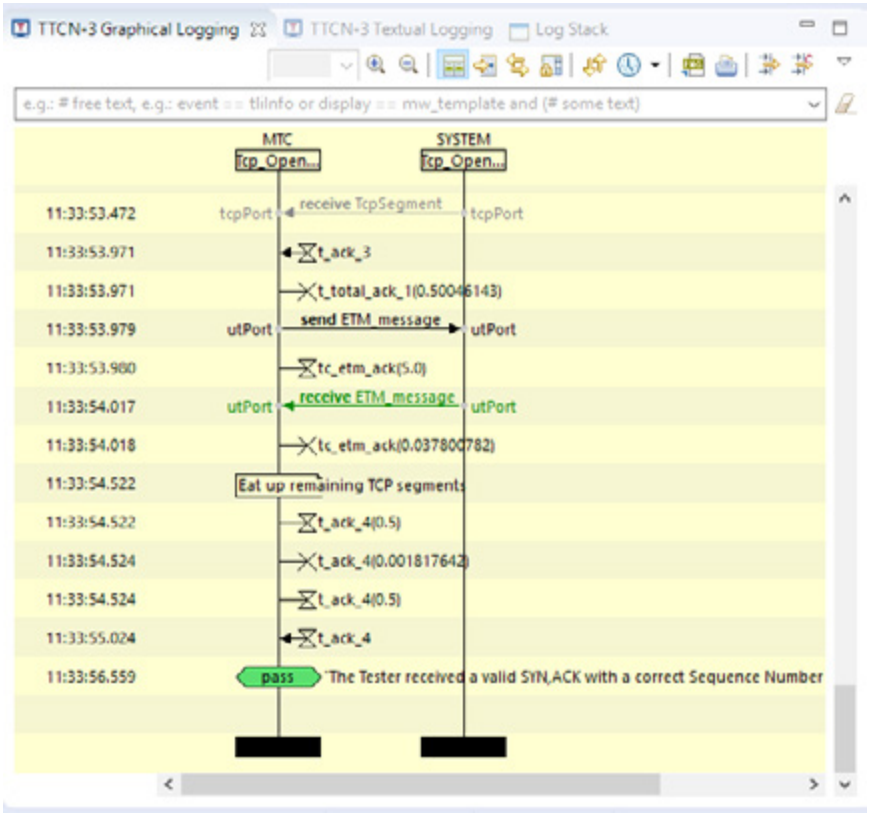


Figure 31. Successful TCP test case execution.

If a required message is not received by the IUT, the verdict of the test will be inconclusive and displayed in yellow.

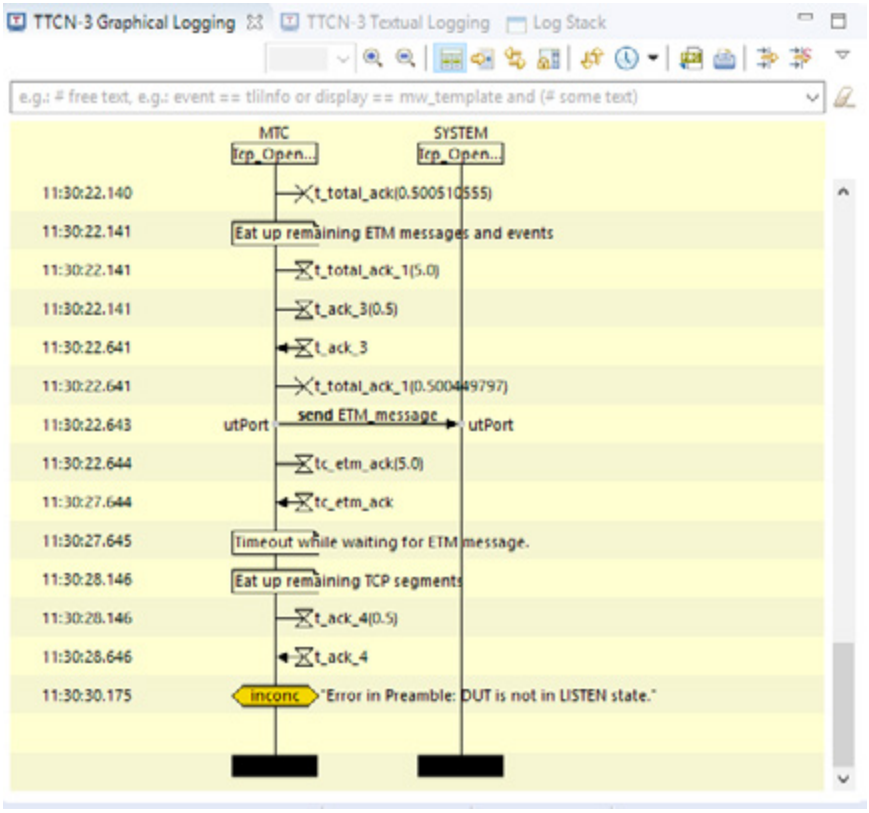


Figure 32. Inconclusive TCP test case execution.

2.6. Conformance Testing: TTsuite-OPEN-UDP

2.6.1. Objective

The objective of this test suite is to verify the User Datagram Protocol (UDP) communication of the Device Under Test according to the following standard:

- RFC 768 - User Datagram Protocol

The representation of the UDP Header fields is shown in the figure below:

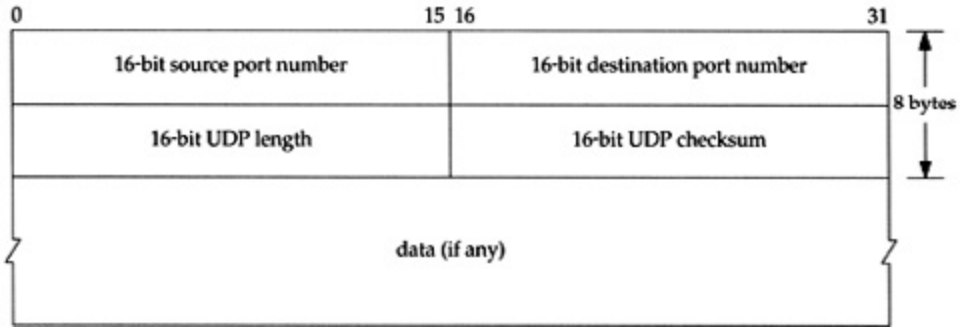


Figure 33. UDP Header.

OPEN-UDP defines 13 test categories and 44 test cases checking the conformance of the proper UDP communication. The test case categories are listed below:

- Messageformat
- DatagramLength
- Padding
- Fields
- UserInterface
- Introduction
- IPOption
- Multihoming
- InvalidAddresses
- AppInterface
- ICMPMessages
- ShotdownSocket

2.6.2. Test Topology

For most of the test cases, the test topology consists of one test interface connected to the DUT via Ethernet. The Upper Tester application runs on DUT using a predefined port. The DUT sends an indicator UDP datagram to the Upper Tester on DUT and it will be echoed back or discarded depending on the UDP header. In some test cases, a control channel is used for stubbing purposes.

2.6.3. Test Suite Configuration

The test suite parameters can be configured in the *Parameters* view of the *TTCN-3 Execution Management* perspective.

2.6.3.1. Upper Tester Configuration

The Upper Tester configuration is like TCP, see chapter 2.5.3.1.

2.6.3.2. IP Interface Configuration

To configure the IP interface, just specify any information available about the connected network interfaces. It is possible to provide the interface name, its MAC and IP addresses. The corresponding module parameters are HOST_1 for the test system and respectively DIFACE_0. For an example configuration, see chapter 2.2.3.2.

2.6.3.3. Test Suite Specific Configuration

If a test case does not pass or the DUT has a custom UDP configuration, consider changing the values listed below and run the test case again. Depending on the IP version of the test suite to execute, verify that the module parameter IP_VERSION matches the loaded test suite version to execute.

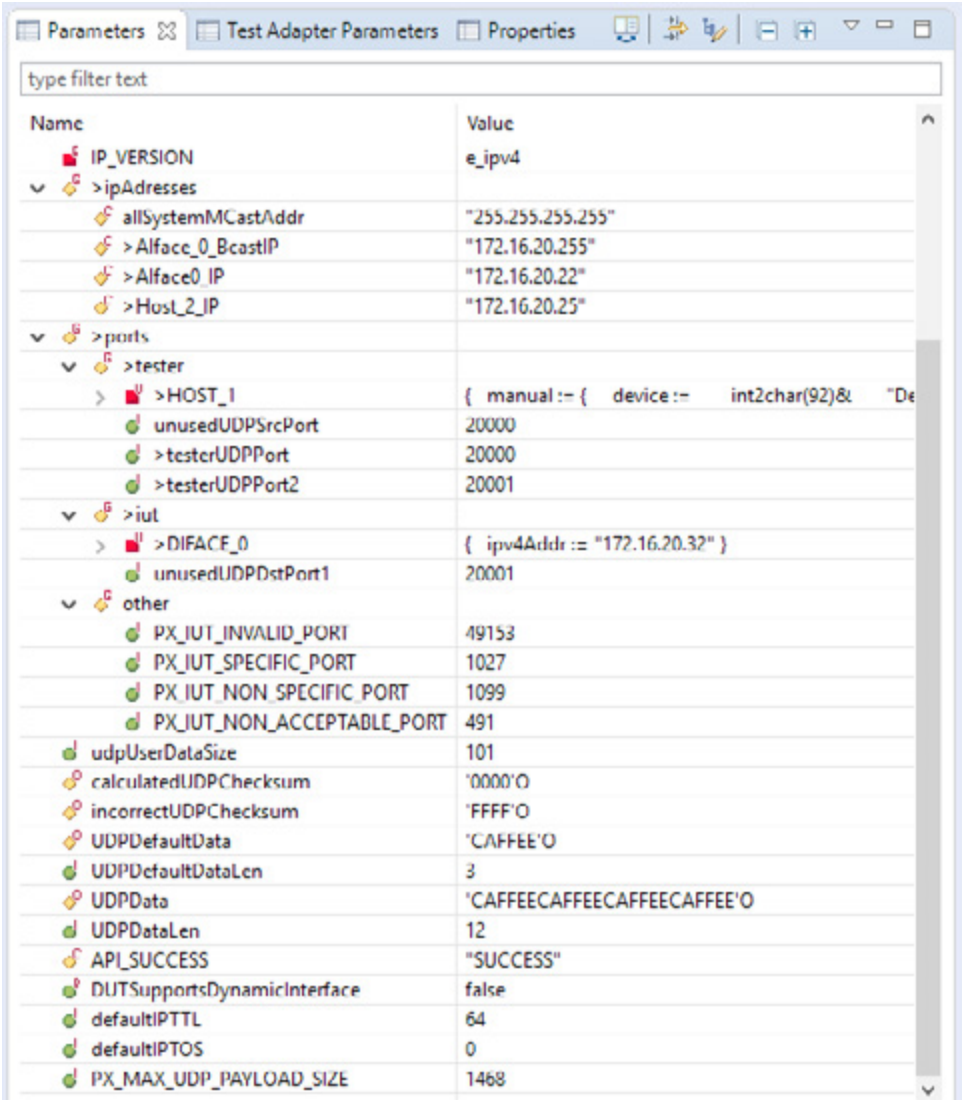


Figure 34. UDP Parameter Configuration.

2.6.3.4. Capture Configuration

To capture incoming and outgoing traffic, the parameter PX_CAPTURE_CONFIGURATION is provided. Enabling it with a custom filter as seen below stores a pcap file for further analysis in the specified location for each executed test case. By default, the configuration captures all UDP, ICMP and Upper Tester messages. Changing the filter has no effect on the actual test case execution. Depending on the used IP version and the local UDP port, consider to also update the capture filter.

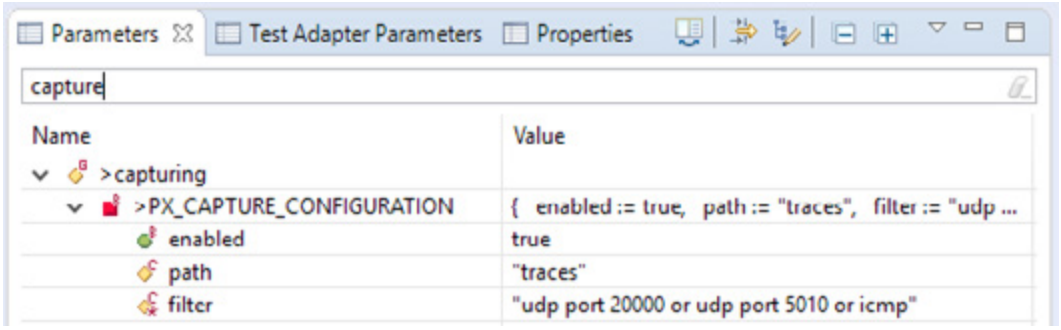


Figure 35. UDP Capture Configuration.

2.6.4. Expected Behavior

Once everything is configured, the test can be run by double clicking on it, or by pressing the run button. If the configuration was correct and if the DUT behaves correctly, messages exchanged between the TS and the DUT are seen in the graphical log and the final verdict is ***pass***, as shown below.

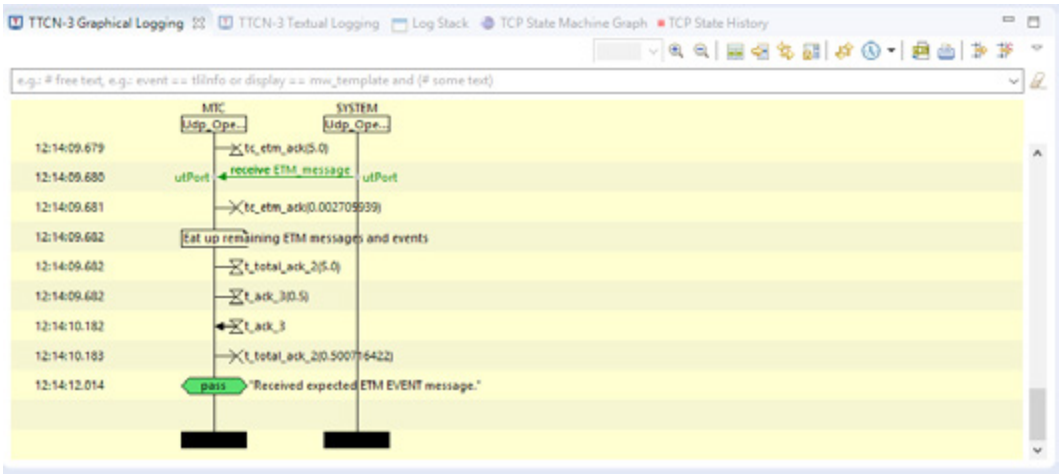


Figure 36. Successful UDP test case execution.

If a required message is not received by the IUT, the verdict of the test will be inconclusive and displayed in *yellow*.

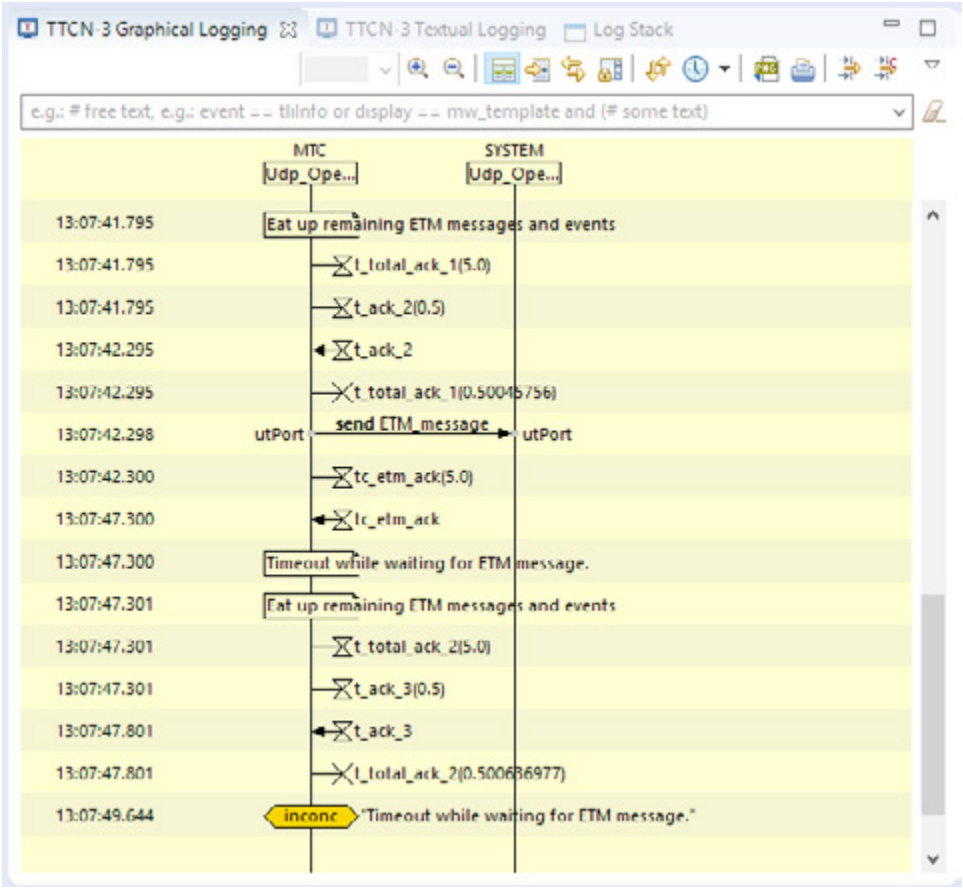


Figure 37. Inconclusive UDP test case execution.

2.7. Conformance Testing: TTsuite-OPEN-ICMPv4

2.7.1. Objective

The objective of this test suite is to verify the Internet Control Message Protocol Version 4 communication (ICMPv4) according to the following standards:

- RFC 792 - Internet Control Message Protocol
- RFC 1122 - Requirements for Internet Hosts—Communication Layers
- RFC 1812 - Requirements for IP Version 4 Routers

The representation of the ICMPv4 header is shown in the figure below:

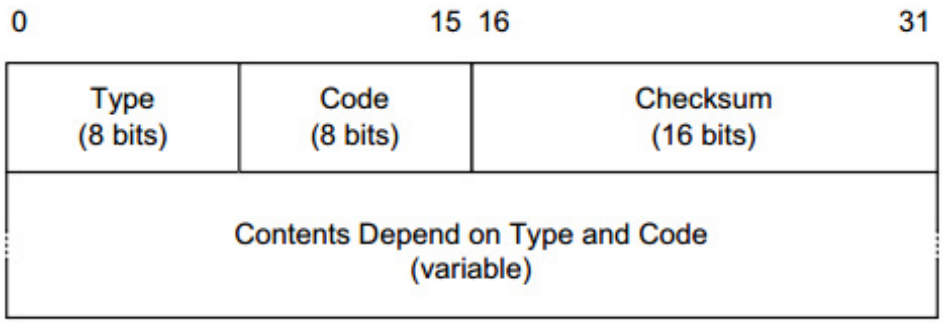


Figure 38. ICMPv4 Header.

There are several types of ICMPv4 request and reply messages covered by this test suite:

- Echo
- Timestamp
- Time Exceeded
- Information
- Parameter Problem
- Destination Unreachable

TTsuite-OPEN-ICMP defines two test categories and 21 test cases. The test categories are listed below:

- Error Handling
- ICMP Types

2.7.2. Test Topology

The test topology is the same as for TTsuite-OPEN-IPv4.

2.7.3. Test Suite Configuration

The test suite parameters can be configured in the *Parameters* view of the *TTCN-3 Execution Management* perspective.

2.7.3.1. Upper Tester Configuration

The Upper Tester configuration is like TTsuite-OPEN-ARP, see chapter 2.2.3.1. The configuration is only needed if you choose to use SSH and execute the test case ICMPv4_ERROR_01. In all other cases you may ignore the Upper Tester configuration.

2.7.3.2. IP Interface Configuration

To configure the IP interface, just specify any information available about the connected network interfaces. It is possible to provide the interface name, its MAC and IP addresses. The corresponding module parameters are TESTER for the test system and respectively DUT. For an example configuration, see chapter 2.2.3.2.

2.7.3.3. Test Suite Specific Configuration

If a test case does not pass, consider changing the timer values listed below and run the test case again. In some cases, the value of the fragment reassembly timeout may not be high enough for the SUT to discard fragments. For this reason, the actual fragment reassembly timeout can be looked up at /proc/sys/ net/ipv4/ipfrag_time (linux-based systems).

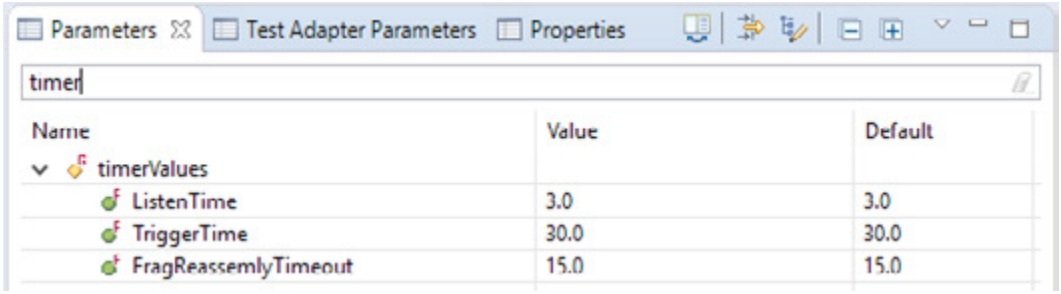


Figure 39. ICMPv4 Timer Values.

You may also have to change the default values for the parameters UnreachablePort and UnusedUDPPort to suite your needs.

2.7.3.4. Capture Configuration

To capture incoming and outgoing traffic, the parameter PX_CAPTURE_CONFIGURATION is provided. Enabling it with a custom filter as seen below stores a pcap file for further analysis in the specified location for each executed test case. By default, the configuration captures all ICMPv4 messages. Changing the filter has no effect on the actual test case execution.

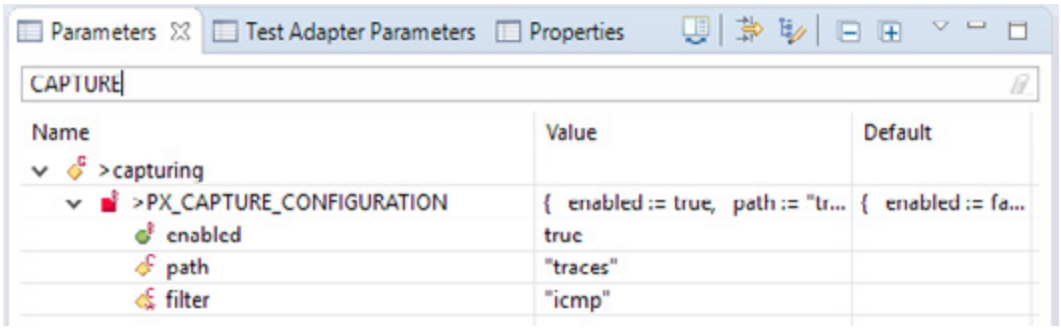


Figure 40. ICMPv4 Capture Configuration.

2.7.4. Expected Behavior

The tests should be run against the IPv4 stack. After giving IP and MAC address to the DUT and the tester as well, the tests can be executed with double click on the desired test case or push on the green play button. If everything is well-configured the verdict will be ***pass*** and displayed with green.

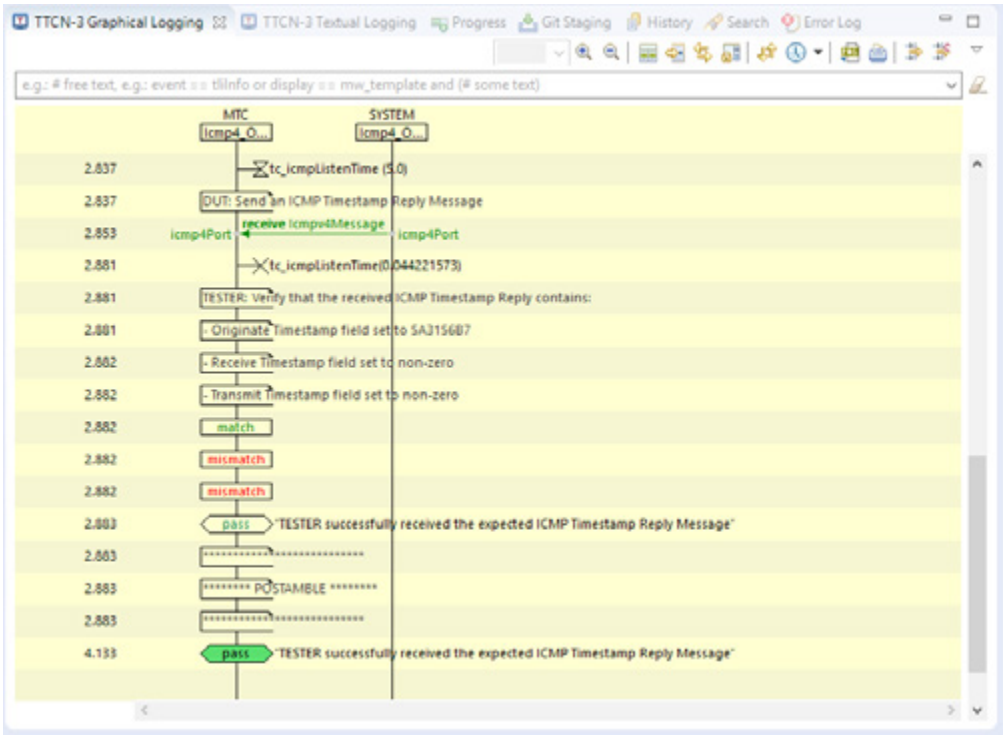


Figure 41. Successful ICMPv4 test case execution.

About Spirent Communications

Spirent Communications (LSE: SPT) is a global leader with deep expertise and decades of experience in testing, assurance, analytics and security, serving developers, service providers, and enterprise networks.

We help bring clarity to increasingly complex technological and business challenges.

Spirent's customers have made a promise to their customers to deliver superior performance. Spirent assures that those promises are fulfilled.

For more information, visit: www.spirent.com

3. Acronyms

ADAS	Advanced Driver Assistance Systems
DHCPv4	Dynamic Host Configuration Protocol Version 4
DUT	Device Under Test
ICMPv4	Internet Control Message Protocol Version 4
IPv4	Internet Protocol Version 4
IUT	Interface Under Test
MAC	Medium Access Control
OABR	OPEN Alliance BroadR-Reach
SUT	System Under Test
TCP	Transmission Control Protocol
TS	Test System
TSN	Time-Sensitive Networking
TTwb	TTworkbench
UDP	User Datagram Protocol

Contact Us

For more information, call your Spirent sales representative or visit us on the web at www.spirent.com/ContactSpirent.

www.spirent.com

© 2018 Spirent Communications, Inc. All of the company names and/or brand names and/or product names and/or logos referred to in this document, in particular the name "Spirent" and its logo device, are either registered trademarks or trademarks pending registration in accordance with relevant national laws. All rights reserved. Specifications subject to change without notice.

Americas 1-800-SPIRENT
+1-800-774-7368 | sales@spirent.com

US Government & Defense
info@spirentfederal.com | spirentfederal.com

Europe and the Middle East
+44 (0) 1293 767979 | emeainfo@spirent.com

Asia and the Pacific
+86-10-8518-2539 | salesasia@spirent.com

Rev C | 08/18