Document No. ATPR246189

**Prepared For Hitachi High-Tech Solutions Corp**

Acceptance Test Plan and Results

**JPN-HITH246189**



Acceptance Test Plan and Results

Version 3.00

Sept. 02, 2017

Prepared For Hitachi High-Tech Solutions Corp

Wind River KK

Wind River Systems, Inc.

Ebisu Prime Square Tower

1-1-39 Hiroo, Shibuya-ku,

Tokyo 150-0012 Japan

Phone: 81-3-5778-6001 FAX: 81-3-5778-6004

[www.windriver.com](http://www.windriver.com)

Copyright

Unless otherwise agreed in writing, all copyright and intellectual property rights embodied in this document are and shall remain the property of Wind River Systems, Inc. This document is provided solely for the purposes of evaluating the work proposed and no other rights whatsoever to use the information herein are granted. The contents of this document may not be disclosed to any third party without the prior written consent of Wind River Systems, Inc.

Trademarks

Wind River, the Wind River logo, Tornado, and VxWorks are registered trademarks of Wind River Systems, Inc. Any third party trademarks referenced are the property of their respective owners. For further information regarding Wind River trademarks, please see <http://www.windriver.com/company/terms/trademark.html>.

| Revision History | | | | |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Date | Version | By | Description of Change |
| 23 July 2017 | 1.00 | Wind River | Official Release |
| 28 Aug 2017 | 2.00 | Wind River | Official Release |
| 02 Sept 2017 | 3.00 | Wind River | Official Release |

**Table of Contents**

1. Applicable Documents 5

2. Glossary of Terms 6

3. Introduction 7

3.1 Purpose and Scope 7

3.2 Overview 7

3.3 Acceptance Criteria Definition 7

3.3.1 Overview 7

3.3.1.1 Entrance Criteria 7

3.3.1.2 Exit Criteria 8

3.4 Acceptance 8

4. Open Issues 9

5. Test Summary 10

5.1 Test Environment 10

5.2 Test applications 11

5.3 Test Scenarios & Cases 14

6. Test Results 22

7. Test Logs 26

# Applicable Documents

The following documents are referenced within:

| No. | Document | Version | Scope |
| --- | --- | --- | --- |
| 1 | JPN\_HITH246189DS001R200J | 2.00 | Design Specification |
| 2 | JPN\_HITH246189TS300J | 3.00 | Test Specification |
| 3 | JPN\_HITH246189RN100J | 1.00 | Release Note |

# Glossary of Terms

The table below represents an alphabetical list of abbreviations, acronyms, and terminology commonly used in reference to this product.

| **Term** | **Definition** |
| --- | --- |
| IP | Internet Protocol |
| UDP | User Diagram Protocol |
| ATP | Acceptance Test Plan |
| BSP | Board Support Package |

# Introduction

## Purpose and Scope

The purpose of this document is to define the acceptance test plan and procedures for support to VxWorks 7 for broadcast addresses with all bits in host part set to zero. However 0.0.0.0 is out of scope.

An example is below:

IP address: 192.168.1.4

Netmask: 255.255.255.0

Broadcast address: 192.168.1.0

This document shall cover tests for broadcast using UDP/IP communication.

Broadcast communication support shall also work when VxWorks7 device is operating with multiple network interfaces with each interface connecting to a different IPv4 subnet.

## Overview

Acceptance tasks take place throughout the project life cycle and culminate in acceptance testing.

Acceptance Testing incorporates four phases:

1. Defining the acceptance criteria
2. Developing an acceptance test plan
3. Executing the acceptance test plan
4. Reaching acceptance based on the test results

## Acceptance Criteria Definition

### Overview

This section defines the acceptance criteria and provides a high-level checklist to enter and exit test execution.

#### Entrance Criteria

The entrance criteria detailed below must be completed prior to starting formal acceptance testing:

* Customer must approve this ATP.
* All software development must be completed (with agreed upon exceptions as outlined in known issues mentioned in this document).
* All software must be under configuration management and frozen prior to the acceptance test.
* All resources (human, hardware, or software) must be assigned and in place.

#### Exit Criteria

The Exit Criteria detailed below must be achieved to complete acceptance testing for the project:

All acceptance tests have been conducted and a pass rate of 100% (or mutual agreement on tests not passing reached, Explain why you cannot PASS technically) has been achieved.

The test report has been reviewed and accepted.

## Acceptance

Acceptance happens when all pre-defined exit criteria have been achieved. Upon completion of formal acceptance testing, the test lead and the Project Manager will approve the test results. The test result document will be provided to Customer for review and acceptance.

# Open Issues

Not applicable.

# Test Summary

## Test Environment

Two environments shall be used for test executions.

1. VxWorks simulator environment
2. Wind River owned target board environment

1) x86 PC

Ubuntu 14.04.5 LTS

Wind River Workbench 4

VxWorks7: vxsim simulator

LAN simulator – vxsimnetd

2)VxWorks 7

BSP: ARM: Cortex A9

Architecture: Cyclone V

Board: Cyclone V

BSP: Cyclone V BSP for VxWorks7

Ethernet Switch – 1

Test environment shall support vxWorks7 nodes with multiple interfaces.

It is difficult to construct old version of test environment. So, testing shall be executed between VxWorks7 environments as programs used are the same.

An example network topology is shown below.

VxWorks7 - 1

192.168.10.1

192.168.1.1

VxWorks7 - 3

192.168.10.3

192.168.1.3

VxWorks7 - 2

192.168.10.2

192.168.1.2

VxWorks7 - 4

192.168.10.4

192.168.1.4

Ethernet LAN

## Test applications

1. Broadcast client

This test application sends UDP broadcast packet to a broadcast address supplied as argument. Outline of program is given below.

Broadcast client is built as downloadable kernel module. To run broadcast client, follow procedure below:

STATUS bc

(

char \*bsaddr, int port /\* IP addr and port number \*/

)

{

int sockFd; /\* socket fd \*/

struct sockaddr\_in sendToAddr; /\* receiver's addresss \*/

int sendNum ;

int on;

struct sockaddr\_in bsAddr;

int bsAddrSize = sizeof (struct sockaddr\_in);

/\* Open UDP socket \*/

sockFd = socket(AF\_INET, SOCK\_DGRAM, 0);

if (sockFd == ERROR)

{

perror ("socket not opened ");

return (ERROR);

}

/\* Use the SO\_BROADCAST option when an application needs to broadcast data

\*/

on = 1; /\* turn ON SO\_BROADCAST option\*/

if (setsockopt (sockFd, SOL\_SOCKET, SO\_BROADCAST, (char \*) &on, sizeof(int)) == ERROR)

{

perror ("setsockopt failed ");

return (ERROR);

}

/\* zero out the sockaddr\_in structures and setup receivers' address \*/

bzero ((char \*) &sendToAddr, sizeof (struct sockaddr\_in));

sendToAddr.sin\_family = AF\_INET;

sendToAddr.sin\_port = htons (port);

sendToAddr.sin\_addr.s\_addr = inet\_addr (bsaddr);

printf("sendToAddr.sin\_addr.s\_addr %x\n",sendToAddr.sin\_addr.s\_addr);

FOREVER {

gets(usermsg);

if (strcmp(usermsg,"quit") == 0) break;

sprintf(buf,"msg%03d-%s",msgcount++,usermsg);

/\* send the broadcast message to other systems in the same network \*/

if ((sendNum = sendto (sockFd, buf, sizeof (buf), 0, (struct sockaddr \*) &sendToAddr,

sizeof (struct sockaddr\_in))) == ERROR) {

perror ("sendto broadcast failed ");

return (ERROR);

}

printf (">>> [%s:%d] %s\n",bsaddr,port,buf);

if (strcmp(usermsg,"stop bs") == 0) break;

if (recvfrom (sockFd, &buf, sizeof (buf), 0,

(struct sockaddr \*) &bsAddr,&bsAddrSize) == ERROR) {

perror ("recvfrom");

close (sockFd);

return (ERROR);

}

inet\_ntoa\_b (bsAddr.sin\_addr, bsAddrStr);

printf ("<<< [%s:%d] %s\n",bsAddrStr,ntohs(bsAddr.sin\_port), buf);

}

close (sockFd);

return(TRUE);

}

At VxWorks prompt,

->ld <bc.out

->bc(“192.168.10.0”,7001);

1. Broadcast server

This test application acts as UDP server that can accept packets with broadcast destination and echo back the message to the sender. Server takes an address as argument on which it shall listen for broadcast packets. Argument can be ADDR\_ANY (0.0.0.0) for server to listen to all interfaces. It listens to port 7001. Outline of program is given below.

STATUS bs(char \*bsaddr)

{

struct sockaddr\_in myAddr; /\* Server socket address \*/

struct sockaddr\_in clientAddr; /\* Socket address for client \*/

char clientRequest[128];

int sFd; /\* Server's socket file descriptor \*/

char inetAddr[INET\_ADDR\_LEN]; /\* Buffer for dot notation \*

\* internet addr of client \*/

/\* Size of socket address structure \*/

int sockAddrSize = sizeof (struct sockaddr\_in);

int msgcount = 1;

LOCAL char replyMsg[128];

/\* Build socket address \*/

bzero ((char \*) &myAddr, sockAddrSize);

myAddr.sin\_family = AF\_INET;

myAddr.sin\_port = htons (SERVER\_PORT\_NUM);

if (!bsaddr || !strcmp(bsaddr,"")) {

myAddr.sin\_addr.s\_addr = htonl (INADDR\_ANY);

}

else {

myAddr.sin\_addr.s\_addr = inet\_addr(bsaddr);

}

/\* Create socket \*/

if ((sFd = socket (AF\_INET, SOCK\_DGRAM, 0)) == ERROR) {

perror ("socket");

close (sFd);

return (ERROR);

}

/\* Bind socket to local address \*/

if (bind (sFd, (struct sockaddr \*) &myAddr, sockAddrSize) == ERROR) {

perror ("bind");

close (sFd);

return (ERROR);

}

printf("Broadcast server listening on port %d\n", ntohs(myAddr.sin\_port));

FOREVER {

/\* Read data from a socket and satisfy requests \*/

if (recvfrom (sFd, &clientRequest, sizeof (clientRequest), 0,

(struct sockaddr \*) &clientAddr,&sockAddrSize) == ERROR) {

perror ("recvfrom");

close (sFd);

return (ERROR);

}

/\* Convert internet address to dot notation for displaying \*/

inet\_ntoa\_b (clientAddr.sin\_addr, inetAddr);

printf ("<<< [%s:%d]: %s\n",

inetAddr, ntohs (clientAddr.sin\_port), clientRequest);

if (strstr(clientRequest,"-stop bs") != 0) break;

sprintf(replyMsg,"msg%03d-%s",msgcount++,"reply from bserver");

if (sendto (sFd, replyMsg, sizeof (replyMsg), 0,

(struct sockaddr \*) &clientAddr,sockAddrSize) == ERROR){

perror ("sendto");

close (sFd);

return (ERROR);

}

printf (">>> [%s:%d]: %s\n", inetAddr,

ntohs (clientAddr.sin\_port), replyMsg);

}

close (sFd); /\* Just in case. Should never get here. \*/

return TRUE;

}

Broadcast client is built as downloadable kernel module. To run server, follow procedure below:

At VxWorks prompt,

->ld <bs.out

->bs(“192.168.10.3”);

## Test Scenarios & Cases

Following table lists different test scenarios and test cases under each of them.

***Note: The test cases in the below table are applicable for test environment 1 and 2 (Ref section: 5.1 Test Environment).***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Client Configuration** | **Server Configuration** | **Network stack Action** | **Expected result** |
| 1 | SCN01 | 1server, 1client | 1 interface per vxsim | 255.255.255.0 |  |
|  | **TC0101** | Simnet0:192.168.10.5 | Simnet0(192.168.10.6) Listens to ADDR\_ANY | Client sends broadcast on192.168.10.0 | server receive OK |
|  | **TC0102** | Simnet0:192.168.10.5 | Simnet0(192.168.10.6) Listens to ADDR\_ANY | Client sends broadcast on192.168.10.255 | server receive OK |
|  | **TC0103** | Simnet0:192.168.1.5 | Simnet0(192.168.10.6) Listens to ADDR\_ANY | Client sends broadcast on192.168.1.0 | Server receive NG |
|  | **TC0104** | Simnet0:192.168.1.5 | Simnet0(192.168.10.6) Listens to ADDR\_ANY | Client sends broadcast on192.168.1.255 | Server receive NG |
| 2 | SCN02 | 1server, 2client | 1 interface per vxsim | 255.255.255.0 |  |
|  | **TC0201** | Client1::  Simnet0:192.168.10.5 Client2: simnet0:192.168.10.56 | Simnet0::192.168.10.6 Listens to ADDR\_ANY | a)Client1sends broadcast on192.168.10.0 b)Client2 sends broadcast on 192.168.10.0 | a) Server receive OK b) Server receive OK |
|  | **TC0202** | Client1::  Simnet0:192.168.10.5 Client2: simnet0:192.168.10.56 | Simnet0::192.168.10.6 Listens to ADDR\_ANY | a)Client1 sends broadcast on192.168.10.255 b)Client2 sends broadcast on 192.168.10.255 | a) Server receive OK b) Server receive OK |
|  | **TC0203** | Client1::  Simnet0:192.168.10.5 Client2: simnet0:192.168.20.6 | Simnet0::192.168.10.6 Listens to ADDR\_ANY | a)Client1 sends broadcast on192.168.10.0 b)Client2 sends broadcast on 192.168.20.0 | a) Server receive OK b) Server receive NG |
|  | **TC0204** | Client1::  Simnet0:192.168.10.5 Client2: simnet0:192.168.20.6 | Simnet0::192.168.10.6 Listens to ADDR\_ANY | a)Client1 sends broadcast on192.168.10.255 b)Client2 sends broadcast on 192.168.20.255 | a) Server receive OK b) Server receive NG |
| 3 | SCN03 | 2servers, 1client | 1 interface per vxsim | 255.255.255.0 |  |
|  | **TC0301** | Simnet0:192.168.10.5 | Server1:: Simnet0 192.168.10.6 Seerver2:: Simnet0 192.168.10.7 Servers listen on ADDR\_ANY | Client sends broadcast on192.168.10.0 | Server1 receive OK Server2 receive OK |
|  | **TC0302** | Simnet0:192.168.10.5 | Server1:: Simnet0 192.168.10.6 Seerver2:: Simnet0 192.168.10.7 Servers listen on ADDR\_ANY | Client sends broadcast on192.168.10.255 | Server1 receive OK Server2 receive OK |
|  | **TC0303** | Simnet0:192.168.1.5 | Server1:: Simnet0 192.168.10.6 Seerver2:: Simnet0 192.168.10.7 Servers listen on ADDR\_ANY | Client sends broadcast on192.168.1.0 | Server1 receive NG Server2 receive NG |
|  | **TC0304** | Simnet0:192.168.1.5 | Server1:: Simnet0 192.168.10.6 Seerver2:: Simnet0 192.168.10.7 Servers listen on ADDR\_ANY | Client sends broadcast on192.168.1.255 | Server1 receive NG Server2 receive NG |
| 4 | SCN04 | 2 servers, 2 clients | 1 interface per vxsim | 255.255.255.0 |  |
|  | **TC0401** | Client1::  Simnet0:192.168.10.4 Client2: simnet0:192.168.10.56 | Server1:: Simnet0 192.168.10.6 Seerver2:: Simnet0 192.168.10.7 Servers listen on ADDR\_ANY | Client1 and Client 2 send broadcast on192.168.10.0 | Server1 receive OK Server2 receive OK |
|  | **TC0402** | Client1::  Simnet0:192.168.10.4 Client2: simnet0:192.168.10.56 | Server1:: Simnet0 192.168.10.6 Seerver2:: Simnet0 192.168.10.7 Servers listen on ADDR\_ANY | Client1 and Client 2 send broadcast on192.168.10.255 | Server1 receive OK Server2 receive OK |
|  | **TC0403** | Client1::  Simnet0:192.168.1.4 Client2: simnet0:192.168.1.56 | Server1:: Simnet0 192.168.10.6 Seerver2:: Simnet0 192.168.10.7 Servers listen on ADDR\_ANY | Client1 and client2 send broadcast on 192.168.1.0 | Server1 receive NG Server2 receive NG |
|  | **TC0404** | Client1::  Simnet0:192.168.1.4 Client2: simnet0:192.168.1.56 | Server1:: Simnet0 192.168.10.6 Seerver2:: Simnet0 192.168.10.7 Servers listen on ADDR\_ANY | Client1 and client2 send broadcast on 192.168.1.255 | Server1 receive NG Server2 receive NG |
| 5 | SCN05 | 1server, 1client | 2 interface per vxsim | 255.255.255.0 |  |
|  | **TC0501** | Simnet0:192.168.10.5 Simnet1: 192.168.1.2 | Simnet0:192.168.10.6  Simnet1:192.168.1.3 Listens to 192.168.1.0 | Client sends broadcast on192.168.10.0 | Server receive NG |
|  | **TC0502** | Simnet0:192.168.10.5 Simnet1: 192.168.1.2 | Simnet0:192.168.10.6  Simnet1:192.168.1.3 Listens to 192.168.1.0 | Client sends broadcast on192.168.10.255 | Server receive NG |
|  | **TC0503** | Simnet0:192.168.10.5 Simnet1: 192.168.1.2 | Simnet0:192.168.10.6  Simnet1:192.168.1.3 Listens to 192.168.1.0 | Client sends broadcast on192.168.1.0 | server receive OK |
|  | **TC0504** | Simnet0:192.168.10.5 Simnet1: 192.168.1.2 | Simnet0:192.168.10.6  Simnet1:192.168.1.3 Listens to 192.168.1.255 | Client sends broadcast on192.168.1.255 | server receive OK |
| 6 | SCN06 | 1server, 2client | 2 interface per vxsim | 255.255.255.0 |  |
|  | **TC0601** | Client1::  Simnet0:192.168.10.4, simnet1:192.168.1.2 Client2: simnet0:192.168.10.5 simnet1:192.168.1.3 | Simnet0: 192.168.10.6 Simnet1:192.168.1.4 Listens to 192.168.10.0 | a)Client1sends broadcast on192.168.10.0 b)Client2 sends broadcast on 192.168.1.0 | a) Server receive OK b) Server receive NG |
|  | **TC0602** | Client1::  Simnet0:192.168.10.4, simnet1:192.168.1.2 Client2: simnet0:192.168.10.5 simnet1:192.168.1.3 | Simnet0::192.168.10.6 Simnet1::192.168.1.4 Listens to 192.168.10.255 | a)Client1 sends broadcast on192.168.10.255 b)Client2 sends broadcast on 192.168.1.255 | a) Server receive OK b) Server receive NG |
|  | **TC0603** | Client1::  Simnet0:192.168.10.4, simnet1:192.168.1.2 Client2: simnet0:192.168.10.5 simnet1:192.168.1.3 | Simnet0::192.168.10.6 Simnet1::192.168.1.4 Listens to 192.168.1.0 | a)Client1 sends broadcast on192.168.10.0 b)Client2 sends broadcast on 192.168.1.0 | a) server receive NG b) Server receive OK |
|  | **TC0604** | Client1::  Simnet0:192.168.10.4, simnet1:192.168.1.2 Client2: simnet0:192.168.10.5 simnet1:192.168.1.3 | Simnet0::192.168.10.6 Simnet1::192.168.1.4 Listens to 192.168.1.255 | a)Client1 sends broadcast on192.168.10.255 b)Client2 sends broadcast on 192.168.1.255 | a) server receive NG b) Server receive OK |
| 7 | SCN07 | 2servers, 1client | 2 interface per vxsim | 255.255.255.0 |  |
|  | **TC0701** | Simnet0:192.168.10.5 Simnet1: 192.168.1.2 | Server1:: Simnet0 192.168.10.6 Simet1192.168.1.3 Server2:: Simnet0 192.168.10.7 Simnet1 192.168.1.4  Servers listen on 192.168.10.0 | Client sends broadcast on192.168.10.0 | a)Server1receive OK b)Server2receive OK |
|  | **TC0702** | Simnet0:192.168.10.5 Simnet1: 192.168.1.2 | Server1:: Simnet0 192.168.10.6 Simet1192.168.1.3 Server2:: Simnet0 192.168.10.7 Simnet1 192.168.1.4  Server1 listen on 192.168.10.255 Server2 listens on 192.168.1.255 | Client sends broadcast on192.168.10.255 | a)Server1 receive OK b)Server2 receive NG |
|  | **TC0703** | Simnet0:192.168.10.5 Simnet1: 192.168.1.2 | Server1:: Simnet0 192.168.10.6 Simet1192.168.1.3 Seerver2:: Simnet0 192.168.10.7 Simnet1 192.168.1.4  Servers listen on 192.168.1.10 | Client sends broadcast on192.168.1.0 | a)Server1receive OK b)Server2receive OK |
|  | **TC0704** | Simnet0:192.168.10.5 Simnet1: 192.168.1.2 | Server1:: Simnet0 192.168.10.6 Simet1192.168.1.3 Seerver2:: Simnet0 192.168.10.7 Simnet1 192.168.1.4  Server1 listen on 192.168.10.255 Server2 listens on 192.168.1.255 | Client sends broadcast on192.168.1.255 | a)Server1receive NG b)Server2receive OK |
| 8 | SCN08 | 2 servers, 2 clients | 2 interface per vxsim | 255.255.255.0 |  |
|  | **TC0801** | client1::  Simnet0:192.168.10.4, simnet1:192.168.1.2 Client2: simnet0:192.168.10.5 simnet1:192.168.1.3 | Server1:: Simnet0 192.168.10.6 Simet1192.168.1.4 Server2:: Simnet0 192.168.10.7 Simnet1 192.168.1.5  Servers listen on 192.168.10.0 | Client1 and Client 2 send broadcast on192.168.10.0 | Server1 receive OK Server2 receive OK |
|  | **TC0802** | client1::  Simnet0:192.168.10.4, simnet1:192.168.1.2 Client2: simnet0:192.168.10.5 simnet1:192.168.1.3 | Server1:: Simnet0 192.168.10.6 Simet1192.168.1.4 Server2:: Simnet0 192.168.10.7 Simnet1 192.168.1.5  Server1 listen on 192.168.10.0 Server2 listens on 192.168.1.0 | Client1 and Client 2 send broadcast on192.168.10.0 | Server1 receive OK Server2 receive NG |
|  | **TC0803** | client1::  Simnet0:192.168.10.4, simnet1:192.168.1.2 Client2: simnet0:192.168.10.5 simnet1:192.168.1.3 | Server1:: Simnet0 192.168.10.6 Simet1192.168.1.4 Server2:: Simnet0 192.168.10.7 Simnet1 192.168.1.5  Servers listen on 192.168.1.0 | Both clients send broadcast on 192.168.1.0 | Server1 receive OK Server2 receive OK |
|  | **TC0804** | client1::  Simnet0:192.168.10.4, simnet1:192.168.1.2 Client2: simnet0:192.168.10.5 simnet1:192.168.1.3 | Server1:: Simnet0 192.168.10.6 Simet1192.168.1.4 Server2:: Simnet0 192.168.10.7 Simnet1 192.168.1.5  Server1 listen on 192.168.10.0 Server2 listens on 192.168.1.0 | Both clients send broadcast on 192.168.1.0 | Server1 receive NG Server2 receive OK |
| 9 | SCN09 | 1server, 1client | 2 interface per vxsim | 255.255.0.0 |  |
|  | **TC0901** | Simnet0:192.168.10.5 Simnet1: 10.10.1.2 | Simnet0:192.168.10.6 Simnet1:10.10.1.3 Listens to 10.10.0.0 | Client sends broadcast on192.168.0.0 | Server receive NG |
|  | **TC0902** | Simnet0:192.168.10.5 Simnet1:: 10.10.1.2 | Simnet0 : 192.168.10.6 Simnet1: 10.10.1.3 Listens to 10.10.255.255 | Client sends broadcast on192.168.255.255 | Server receive NG |
|  | **TC0903** | Simnet0:192.168.10.5 Simnet1: 10.10.1.2 | Simnet0: 192.168.10.6 Simnet1:10.10.1.3 Listens to 10.10.0.0 | Client sends broadcast on10.10.0.0 | server receive OK |
|  | **TC0904** | Simnet0:192.168.10.5 Simnet1::10.10.1.2 | Simnet0: 192.168.10.6 Simnet1:10.10.1.3 Listens to 10.10.255.255 | Client sends broadcast on 10.10.255.255 | server receive OK |
| 10 | SCN10 | 1server, 2client | 2 interface per vxsim | 255.255.0.0 |  |
|  | **TC1001** | Client1::  Simnet0:192.168.10.4, Simnet1:10.10.1.2 Client2: Simnet0:192.168.10.5 Simnet1:10.10.1.3 | Simnet0:192.168.10.6 Simnet1:10.10.1.4 Listens to 192.168.0.0 | a)Client1 sends broadcast on192.168.0.0 b)Client2 sends broadcast on 10.10.0.0 | a) Server receive OK b) Server receive NG |
|  | **TC1002** | Client1::  Simnet0:192.168.10.4, Simnet1:10.10.1.2 Client2: Simnet0:192.168.10.5 Simnet1:10.10.1.3 | Simnet0::192.168.10.6 Simnet1::10.10.1.4 Listens to 192.168.255.255 | a)Client1 sends broadcast on192.168.255.255 b)Client2 sends broadcast on 10.10.255.255 | a) Server receive OK b) Server receive NG |
|  | **TC1003** | Client1::  Simnet0:192.168.10.4, Simnet1:10.10.1.2 Client2: Simnet0:192.168.10.5 Simnet1:10.10.1.3 | Simnet0:192.168.10.6 Simnet1:10.10.1.4 Listens to 10.10.0.0 | a)Client1 sends broadcast on192.168.0.0 b)Client2 sends broadcast on 10.10.0.0 | a) Server receive NG b) Server receive OK |
|  | **TC1004** | Client1::  Simnet0:192.168.10.4, Simnet1:10.10.1.2 Client2: Simnet0:192.168.10.5 Simnet1:10.10.1.3 | Simnet0::192.168.10.6 Simnet1::10.10.1.4 Listens to 10.10.255.255 | a)Client1 sends broadcast on192.168.255.255 b)Client2 sends broadcast on 10.10.255.255 | a) Server receive NG b) Server receive OK |
| 11 | SCN11 | 2servers, 1client | 2 interface per vxsim | 255.255.0.0 |  |
|  | **TC1101** | Simnet0: 192.168.10.5 Simnet1: 10.10.1.2 | Server1:: Simnet0: 192.168.10.6 Simet1: 10.10.1.3 Server2:: Simnet0 : 192.168.10.7 Simnet1 : 10.10.1.4  Servers listen on 192.168.0.0 | Client sends broadcast on192.168.0.0 | a)Server1 receive OK b)Server2 receive OK |
|  | **TC1102** | Simnet0: 192.168.10.5 Simnet1: 10.10.1.2 | Server1:: Simnet0: 192.168.10.6 Simet1::10.10.1.3 Seerver2:: Simnet0: 192.168.10.7 Simnet1: 10.10.1.4  Server1 listen on 192.168.255.255 Server2 listens on 10.10.0.0 | Client sends broadcast on192.168.255.255 | a)Server1 receive OK b)Server2 receive NG |
|  | **TC01103** | Simnet0: 192.168.10.5 Simnet1: 10.10.1.2 | Server1:: Simnet0: 192.168.10.6 Simet1: 10.10.1.3 Server2:: Simnet0 : 192.168.10.7 Simnet1 : 10.10.1.4  Servers listen on 10.10.0.0 | Client sends broadcast on10.10.0.0 | a)Server1receive OK b)Server2receive OK |
|  | **TC1104** | Simnet0: 192.168.10.5 Simnet1: 10.10.1.2 | Server1:: Simnet0: 192.168.10.6 Simet1::10.10.1.3 Seerver2:: Simnet0: 192.168.10.7 Simnet1: 10.10.1.4  Server1 listen on 192.168.0.0 Server2 listens on 10.10.255.255 | Client sends broadcast on10.10.255.255 | a)Server1receive NG b)Server2receive OK |
| 12 | SCN12 | 2 servers, 2 clients | 2 interface per vxsim | 255.255.0.0 |  |
|  | **TC1201** | Client1::  Simnet0:192.168.10.4,  Simnet1::10.10.1.2 Client2: Simnet0:192.168.10.5 Simnet1::10.10.1.3 | Server1:: Simnet0:192.168.10.6 Simnet1:10.10.1.4  Server2:: Simnet0:: 192.168.10.7 Simnet1:: 10.10.1.5  Servers listen on 192.168.0.0 | Client1 and Client 2 send broadcast on192.168.0.0 | Server1 receive OK Server2 receive OK |
|  | **TC1202** | Client1::  Simnet0:192.168.10.4,  Simnet1::10.10.1.2 Client2: Simnet0:192.168.10.5 Simnet1::10.10.1.3 | Server1:: Simnet0:192.168.10.6 Simnet1:10.10.1.4  Server2:: Simnet0:: 192.168.10.7 Simnet1:: 10.10.1.5  Server1 listens on 192.168.0.0  Server2 listens on 10.10.0.0 | Client1 and Client 2 send broadcast on192.168.0.0 | Server1 receive OK Server2 receive NG |
|  | **TC1203** | Client1::  Simnet0:192.168.10.4,  Simnet1::10.10.1.2 Client2: Simnet0:192.168.10.5 Simnet1::10.10.1.3 | Server1:: Simnet0:192.168.10.6 Simnet1:10.10.1.4  Server2:: Simnet0:: 192.168.10.7 Simnet1:: 10.10.1.5  Servers listen on 10.10.0.0 | Both clients send broadcast on 10.10.0.0 | Server1 receive OK Server2 receive OK |
|  | **TC1204** | Client1::  Simnet0:192.168.10.4,  Simnet1::10.10.1.2 Client2: Simnet0:192.168.10.5 Simnet1::10.10.1.3 | Server1:: Simnet0:192.168.10.6 Simnet1:10.10.1.4  Server2:: Simnet0:: 192.168.10.7 Simnet1:: 10.10.1.5  Server1 listens on 192.168.0.0  Server2 listens on 10.10.0.0 | Both clients send broadcast on 10.10.0.0 | Server1 receive NG Server2 receive OK |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | **Client Configuration**  **(Client Runs Vxworks7 image without feature support)** | | **Server Configuration**  **(Server Runs Vxworks7 image with feature support)** | **Network stack Action** | **Expected result** |
| 13 | SCN13 | | 1server, 1client | 1 interface per vxsim | | 255.255.255.0 |  |
|  | **TC1301** | Simnet0:192.168.10.5 | | Simnet0(192.168.10.6) Listens to ADDR\_ANY | | Client sends broadcast on192.168.10.0 | Client fails to send broadcast |
|  | **TC1302** | Simnet0:192.168.10.5 | | Simnet0(192.168.10.6) Listens to ADDR\_ANY | | Client sends broadcast on192.168.10.255 | server receive OK |
|  | **TC1303** | Simnet0:192.168.1.5 | | Simnet0(192.168.10.6) Listens to ADDR\_ANY | | Client send s broadcast on192.168.1.0 | Client fails to send broadcast |
|  | **TC1304** | Simnet0:192.168.1.5 | | Simnet0(192.168.10.6) Listens to ADDR\_ANY | | Client sends broadcast on192.168.1.255 | Server receive OK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Client Configuration**  **(Running on Vxworks7 image WITH feature support)** | **Server Configuration**  **(Server running on vxworks7 image WITHOUT feature support)** | **Network stack Action** | **Expected result** |
| 14 | SCN14 | 1server, 2client | 1 interface per vxsim | 255.255.255.0 |  |
|  | **TC1401** | Simnet0:192.168.10.5 | Simnet0(192.168.10.6) Listens to ADDR\_ANY | Client sends broadcast on192.168.10.0 | server receive NG |
|  | **TC1402** | Simnet0:192.168.10.5 | Simnet0(192.168.10.6) Listens to ADDR\_ANY | Client sends broadcast on192.168.10.255 | server receive OK |
|  | **TC1403** | Simnet0:192.168.1.5 | Simnet0(192.168.10.6) Listens to ADDR\_ANY | Client sends broadcast on192.168.1.0 | Server receive NG |
|  | **TC1404** | Simnet0:192.168.1.5 | Simnet0(192.168.10.6) Listens to ADDR\_ANY | Client sends broadcast on192.168.1.255 | Server receive OK |

# Test Results

**6.1 Test Results**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sl No | Scenario  No | Test Case  ID | ResultVxsim  (PASS/FAIL) | Result\_Board  (PASS/FAIL) | Remarks |
| 1 | SCN01 | TC0101 | PASS |  |  |
| 2 |  | TC0102 | PASS |  |  |
| 3 |  | TC0103 | PASS |  |  |
| 4 |  | TC0104 | PASS |  |  |
| 5 | SCN02 | TC0201 | PASS |  |  |
| 6 |  | TC0202 | PASS |  |  |
| 7 |  | TC0203 | PASS |  |  |
| 8 |  | TC0204 | PASS |  |  |
| 9 | SCN03 | TC0301 | PASS |  |  |
| 10 |  | TC0302 | PASS |  |  |
| 11 |  | TC0303 | PASS |  |  |
| 12 |  | TC0304 | PASS |  |  |
| 13 | SCN04 | TC0401 | PASS |  |  |
| 14 |  | TC0402 | PASS |  |  |
| 15 |  | TC0403 | PASS |  |  |
| 16 |  | TC0404 | PASS |  |  |
| 17 | SCN05 | TC0501 | PASS |  |  |
| 18 |  | TC0502 | PASS |  |  |
| 19 |  | TC0503 | PASS |  |  |
| 20 |  | TC0504 | PASS |  |  |
| 21 | SCN06 | TC0601 | PASS |  |  |
| 22 |  | TC0602 | PASS |  |  |
| 23 |  | TC0603 | PASS |  |  |
| 24 |  | TC0604 | PASS |  |  |
| 25 | SCN07 | TC0701 | PASS |  |  |
| 26 |  | TC0702 | PASS |  |  |
| 27 |  | TC0703 | PASS |  |  |
| 28 |  | TC0704 | PASS |  |  |
| 29 | SCN08 | TC0801 |  |  |  |
| 30 |  | TC0802 |  |  |  |
| 31 |  | TC0803 |  |  |  |
| 32 |  | TC0804 |  |  |  |
| 33 | SCN09 | TC0901 |  |  |  |
| 34 |  | TC0902 |  |  |  |
| 35 |  | TC0903 |  |  |  |
| 36 |  | TC0904 |  |  |  |
| 37 | SCN10 | TC1001 |  |  |  |
| 38 |  | TC1002 |  |  |  |
| 39 |  | TC1003 |  |  |  |
| 40 |  | TC1004 |  |  |  |
| 41 | SCN11 | TC1101 |  |  |  |
| 42 |  | TC1102 |  |  |  |
| 43 |  | TC1103 |  |  |  |
| 44 |  | TC1104 |  |  |  |
| 45 | SCN12 | TC1201 |  |  |  |
| 46 |  | TC1202 |  |  |  |
| 47 |  | TC1203 |  |  |  |
| 48 |  | TC1204 |  |  |  |
| 49 | SCN13 | TC1301 |  |  |  |
| 50 |  | TC1302 |  |  |  |
| 51 |  | TC1303 |  |  |  |
| 52 |  | TC1304 |  |  |  |
| 53 | SCN14 | TC1401 |  |  |  |
| 54 |  | TC1402 |  |  |  |
| 55 |  | TC1403 |  |  |  |
| 56 |  | TC1404 |  |  |  |

**Supplement:**

1) Test results are described either PASS or FAIL.

2) Test environments are described in Remarks.

3) Customer target board has only one active internet interface.

# Test Logs

Refer to evidence files.