Test Plan

Varain

Helsinki November 28, 2006

Software Engineering Project

UNIVERSITY OF HELSINKI
Department of Computer Science

Course
581260 Software Engineering Project (6 cr)

Project Group
Kankaanniemi Marko
Lampila Pekka
Ruottu Toni
Viita Antti

Instructor
Yrjänäinen Sampo

Client
Manner Jukka

Project Masters
Taina Juha

Homepage
`http://www.cs.helsinki.fi/group/varain`

Change Log

Version	Date	Modifications
1.0	27.11.2006	First version

Contents

1 Introduction

This document contains the test plan for software project Varain. Configuration of our testing environment is discussed, and all the planned tests are described in detail. There is also a section about our plans for the testing schedule.

2 Glossary

RSVP Resource Reservation Protocol. Protocol for reserving resources in a IP network.

LRSVP Local Resource Reservation Protocol. Modification of the RSVP protocol, that provides the means to reserve local resources.

KOM RSVP RSVP implementation made by Multimedia Communications Lab at Technische Universität Darmstadt.

LI-bit A bit in LRSVP package used to specify whether the package should be handled by the LRSVP proxy.

Path message A RSVP message type, used to initialize a reservation path. Sent when upstream reservation is desired.

PathRequest message A LRSVP message type, used to initialize a downstream reservation by requesting the receiving and to respond with a path message.

3 Testing Environment

3.1 Computer network

All our testing is done in a test lab described here. The lab consists of four machines. Each of these machines is running Linux, kernel version `2.6.17-1.2174_FC5`. The computers form a network, illustrated in Figure 1, where varain2 and varain3 work as routers, forwarding packets from interface eth0 to eth1 or the other way around as needed.

3.2 Software versions

There are two different versions of the software binaries. They differ by configuration flags, but both versions have these common flags:

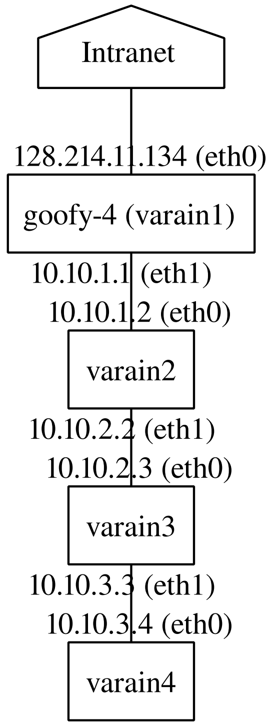

Figure 1: Test lab configuration.

```
--disable-mpls --disable-java --disable-java-api
--disable-refresh-reduction --disable-dns --disable-onepass
--enable-virt --enable-active-api
```

The software version we call a *proxy version*, has one additional configuration flag `--enable-lrsvp-proxy`. These binaries are used to run the LRSVP proxy program and the daemon that is associated with it.

Second software version has the additional flag `--enable-lrsvp`. This version is used for everything else, namely running `recvapi` and `sendapi` programs and running daemons on non-proxy computers. This version will be referred to as *daemon version*.

3.3 Configuration files

Whenever a RSVPD daemon is launched, it needs a configuration file. This file is the same, no matter if proxy version or daemon version of the software is used. The configuration files for computers varain2 and varain3 are following:

```
interface eth0 tc cbq 10000000 10 refresh 10000
interface eth1 tc cbq 10000000 10 refresh 10000
api 4000
```

On varain4, the line concerning interface eth1 is removed. In our tests varain1 won't run the RSVPD daemon.

The LRSVP proxy program uses a configuration file for the list of local interfaces. The configuration file used in these tests always only contains one line, and that line is included in the individual test cases.

3.4 Commandline Parameters

Programs `recvapi` and `sendapi` work based on the command-line parameters. Here you can see what each of these parameters means.

```
recvapi session-addr port [RSVPD-host] [RSVPD-port]
        [LI_Flag{on,off}] [PATH_REQ{on,off}] [src-address]
        [src-port] [rate in byte/s]

sendapi session-addr port [src-port] [rate in byte/s]
        [RSVPD-host] [RSVPD-port] [LI_Flag{on,off}]
```

4 Tests without Proxy

4.1 Upstream reservation without LI-bit set

This test checks for basic RSVP functionality, that is used to reserve upstream bandwidth.

Computer	*Software*	*Function*
varain2	daemon + recvapi	Remote end-host, accept the reservation request
varain4	daemon + sendapi	Local end-host, initialize the reservation request

Setting up varain2

Start daemon version of the RSVP daemon

Run: `recvapi 10.10.2.2 12765`

Setting up varain4

Start daemon version of the RSVP daemon

Run: `sendapi 10.10.2.2 12765 31337 500000 localhost 4000 off`

Computer varain2 has to be setup before varain4.

Result on a successful run is that a RSVP reservation from varain4 to varain2 is created and kept up until one of the programs is shut down.

4.2 Upstream reservation with LI-bit set

This test checks for the case, where a local upstream reservation is requested, but the remote end host is reached before the LRSVP proxy server.

Note that this test is identical to the previous one, except that the LI-bit is set. Since in this configuration there is no proxy between the two end hosts, the functionality should be the same.

Computer	*Software*	*Function*
varain2	daemon + recvapi	Remote end-host, accept the reservation request
varain4	daemon + sendapi	Local end-host, initialize the reservation request

Setting up varain2

> Start daemon version of the RSVP daemon
>
> Run: `recvapi 10.10.2.2 12765`

Setting up varain4

> Start daemon version of the RSVP daemon
>
> Run: `sendapi 10.10.2.2 12765 31337 500000 localhost 4000 on`

Computer varain2 has to be setup before varain4.

Result on a successful run is that a RSVP reservation from varain4 to varain2 is created and kept up until one of the programs is shut down.

4.3 Downstream reservation without LI-bit set

This test checks the new LRSVP downstream reservation system directly between two end hosts.

Computer	*Software*	*Function*
varain2	daemon + recvapi	Remote end-host, initializes the reservation request based on the path request message
varain4	daemon + recvapi	Local end-host, send a path request message

Setting up varain2

> Start daemon version of the RSVP daemon
>
> Run: `recvapi 10.10.3.4 12765`

Setting up varain4

Start daemon version of the RSVP daemon

Run: `recvapi 10.10.3.4 12765 localhost 4000 off on 10.10.2.2 31337`

Computer varain2 has to be setup before varain4.

Result on a successful run is that a RSVP reservation from varain2 to varain4 is created and kept up until one of the programs is shut down.

4.4 Downstream reservation with LI-bit set

This test checks for the case, where a local downstream reservation is requested, but the remote end host is reached before the LRSVP proxy server.

Note that this test is identical to the previous one, except that the LI-bit is set. Since in this configuration there is no proxy between the two end hosts, the functionality should be the same.

Computer	*Software*	*Function*
varain2	daemon + recvapi	Remote end-host, initializes the reservation request based on the path request message
varain4	daemon + recvapi	Local end-host, send a path request message

Setting up varain2

Start daemon version of the RSVP daemon

Run: `recvapi 10.10.3.4 12765`

Setting up varain4

Start daemon version of the RSVP daemon

Run: `recvapi 10.10.3.4 12765 localhost 4000 on on 10.10.2.2 31337`

Computer varain2 has to be setup before varain4.

Result on a successful run is that a RSVP reservation from varain2 to varain4 is created and kept up until one of the programs is shut down.

5 Tests with Proxy

5.1 Upstream reservation with LI-bit set

This test checks the RSVP upstream reservation works between the local end host and a proxy.

Computer	*Software*	*Function*
varain2	-	Remote end-host, do nothing
varain3	daemon + proxy	LRSVP proxy, accept the reservation request
varain4	daemon + sendapi	Local end-host, initialize the reservation request

Setting up varain2

Doesn't need any setup work, since it won't receive or send any RSVP messages

Setting up varain3

Start proxy version of the RSVP daemon

Set LRSVP proxy's configuration file to ”`LNI 10.10.3.3`”

Start LRSVP proxy

Setting up varain4

Start daemon version of the RSVP daemon

Run: `sendapi 10.10.2.2 12765 31337 500000 localhost 4000 on`

Computer varain3 has to be setup before varain4.

Result on a successful run is that a RSVP reservation from varain4 to varain3 is created and kept up until one of the programs is shut down.

5.2 Upstream reservation without LI-bit set

This test checks that normal RSVP upstream reservation works, even with a LRSVP proxy between the end hosts.

Note that this test is similar to the ”Upstream reservation without LI-bit set” in ”Tests without Proxy”, except that there is a LRSVP proxy between the end hosts. However since LI-bit is not set, the end results should be the same.

Computer	*Software*	*Function*
varain2	daemon + recvapi	Remote end-host, accept the reservation request
varain3	daemon + proxy	LRSVP proxy, pass messages along
varain4	daemon + sendapi	Local end-host, initialize the reservation request

Setting up varain2

Start daemon version of the RSVP daemon

Run: `recvapi 10.10.2.2 12765`

Setting up varain3

Start proxy version of the RSVP daemon

Set LRSVP proxy's configuration file to "`LNI 10.10.3.3`"

Start LRSVP proxy

Setting up varain4

Start daemon version of the RSVP daemon

Run: `sendapi 10.10.2.2 12765 31337 500000 localhost 4000 off`

Computers varain2 and varain3 have to be setup before varain4.

Result on a successful run is that a RSVP reservation from varain4 to varain2 is created and kept up until one of the programs is shut down.

5.3 Downstream reservation with LI-bit set

This test checks the new LRSVP downstream reservation system using a proxy.

Computer	*Software*	*Function*
varain2	-	Remote end-host, do nothing
varain3	daemon + proxy	LRSVP proxy, initializes the reservation request based on the path request message
varain4	daemon + recvapi	Local end-host, send a path request message

Setting up varain2

Doesn't need any setup work, since it won't receive or send any RSVP messages

Setting up varain3

Start proxy version of the RSVP daemon

Set LRSVP proxy's configuration file to "`LNI 10.10.3.3`"

Start LRSVP proxy

Setting up varain4

Start daemon version of the RSVP daemon

Run: `recvapi 10.10.3.4 12765 localhost 4000 on on 10.10.2.2 31337`

Result on a successful run is that a RSVP reservation from varain3 to varain4 is created and kept up until one of the programs is shut down.

5.4 Downstream reservation without LI-bit set

This test checks that the new LRSVP downstream reservation system works between end-hosts, even with a LRSVP proxy in the middle.

Note that this test is similar to the "Downstream reservation without LI-bit set" in "Tests without Proxy", except that there is a LRSVP proxy between the end hosts. However since LI-bit is not set, the end results should be the same.

Computer	*Software*	*Function*
varain2	daemon + recvapi	Remote end-host, initializes the reservation request based on the path request message
varain3	daemon + proxy	LRSVP proxy, pass messages along
varain4	daemon + recvapi	Local end-host, send a path request message

Setting up varain2

Start daemon version of the RSVP daemon

Run: `recvapi 10.10.3.4 12765`

Setting up varain3

Start proxy version of the RSVP daemon

Set LRSVP proxy's configuration file to "`LNI 10.10.3.3`"

Start LRSVP proxy

Setting up varain4

Start daemon version of the RSVP daemon

Run: `recvapi 10.10.3.4 12765 localhost 4000 off on 10.10.2.2 31337`

Result on a successful run is that a RSVP reservation from varain2 to varain4 is created and kept up until one of the programs is shut down.

5.5 Path Err with LRSVP Not Supported Error Spec Object

This test checks that the LRSVP proxy correctly rejects RSVP messages with LI-bit set coming from an interface that is not on the local interface list.

Computer	*Software*	*Function*
varain2	-	Remote end-host, do nothing
varain3	daemon + proxy	LRSVP proxy, deny the reservation request
varain4	daemon + sendapi	Local end-host, initialize the reservation request

Setting up varain2

Doesn't need any setup work, since it won't receive or send any RSVP messages

Setting up varain3

Start proxy version of the RSVP daemon

Set LRSVP proxy's configuration file to "`LNI 10.10.2.3`"

Start LRSVP proxy

Setting up varain4

Start daemon version of the RSVP daemon

Run: `sendapi 10.10.2.2 12765 31337 500000 localhost 4000 on`

Result on a successful run is that the LRSVP daemon sends a path error message with "LRSVP Not Supported" error spec object to varain4.

6 Testing schedule

Testing will be carried out right after this document has been released. In the case of further changes to the code, the tests will be ran again.