Implementation Plan

Varain

Helsinki November 7, 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	31.10.2006	Initial version
1.1	3.11.2006	Updated version

Contents

1 Introduction

This document contains an implementation plan for Varain software project. The project aims to implement LRSVP protocol [Man06] on top of the KOM RSVP package [Kar]. There is already almost working implementation like this made by Adrian Bauffe [Bau05], and the project will focus on fixing and enhancing that implementation.

Due to project's type, this document has somewhat non-traditional structure. First of a very basic overview is given of the how Bauffe's LRSVP implementation on top of the KOM RSVP package looks like. After that the changes we think or knew are needed are listed.

2 Glossary

RSVP Resource Reservation Protocol. Protocol for reservering 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 Universitat Darmstädt.

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

ER-bit A bit in LRSVP package used to specify that explicit repair mechanism should be used to refresh connection path without waiting for timeouts.

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 reserveration by requesting the receiving and to respond with a path message.

PathRequestTear message A LRSVP message type, used for tearing down downstream reserverations.

3 LRSVP System Overview

3.1 Proxy

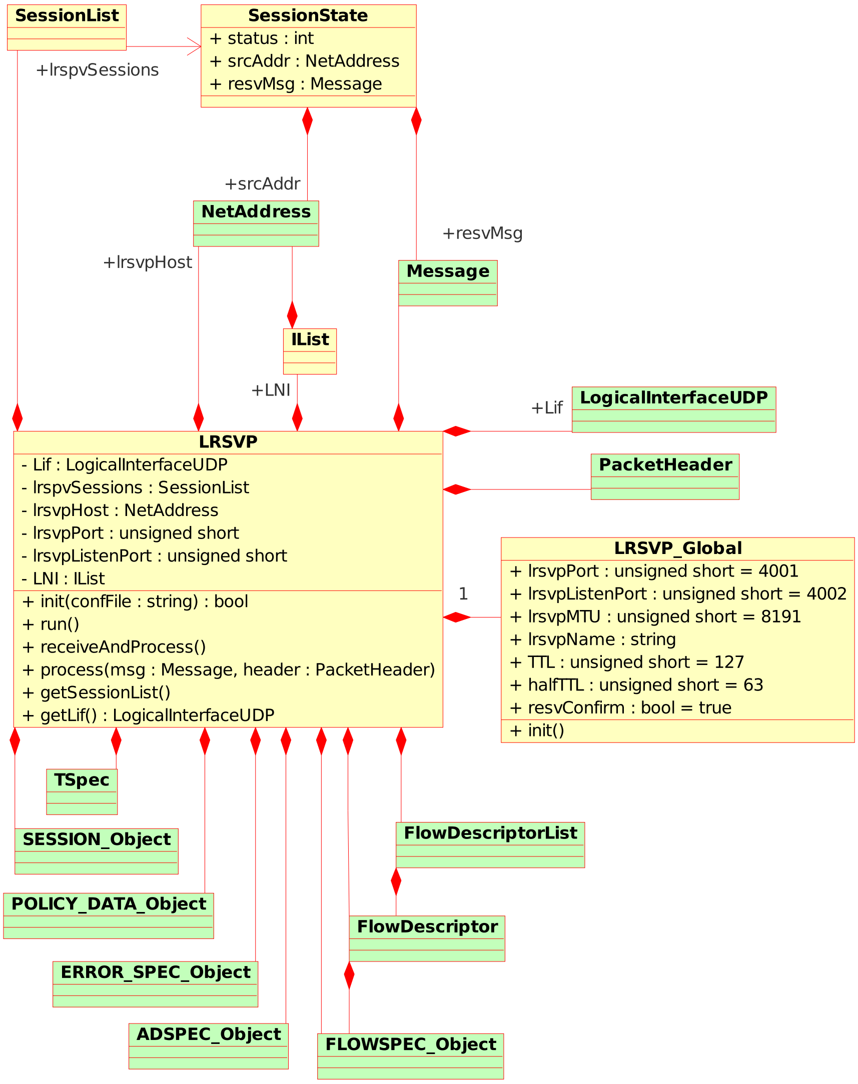

Figure 1: Class diagram of LRSVP proxy

LRSVP proxy is a program running on a computer between local and remote end hosts. It intercepts RSVP packets coming from local network interface that have LI-bit set, and responds to them.

As mentioned earlier, our plan is to fix Bauffe's LRSVP implementation. His LRSVP proxy is implemented on top of the KOM RSVP package. A class diagram of new classes made by Bauffe and their relation to old KOM RSVP classes is shown in Figure 1. Many of these existing classes have however been extended to support new message types of LRSVP etc.

The implementation in question works in conjuction with KOM RSVP daemon program. LRSVP proxy communicates with daemon process over UDP. When the proxy is initialized, certain configuration variables can be set, like the interfaces to be considered local. After initialization, the program will run in a loop listening for new network packets.

3.2 Tools

The LRSVP system made by Bauffe also comes with a tool made to work with LRSVP specific features. This tool is a program called lrsvpTest, that can be used for creating reservations using either LRSVP or plain RSVP. The program can be run either in sender or receiver mode, based on whether it is initializing the RSVP session.

4 Daemon and Proxy

4.1 Configuration

LRSVP proxy configuration file consists of a list of local interface descriptions. We will change the proxy to load the configuration file by default from /etc/komrsvp-proxy.conf, instead of file config.cfg used by Bauffe. This way the proxy configuration is more similar to configuration of the daemon, which also loads its configuration file from /etc folder. We also consider config.cfg a very nondescriptive filename. We will also change the proxy to read local interface IP addresses from command line parameters. When local interface information is provided as command line parameters the configuration file will be ignored.

4.2 Bugs and Issues

There are some bugs and some missing features in the current implementation. The most serious ones that affect the operation of the entire protocol are described below.

The current implementation has a problem with the routing of PathReq and PathReqTear messages. Intermediate routers don't pass the RSVP packets to the daemon for processing. We have found that this is because the generated IP packets don't include the router alert flag. The router alert flag is included in the all the original RSVP packet types but not in the new LRSVP packet types. The solution is simply to include the flag in all necessary packets.

The fast local repair mechanism is not implemented in the current system. The ER bit is included in all the necessary data structures and the tools support it but it doesn't affect the message processing the way it is meant to. Also the API already supports it.

Making a downstream reservation with the LI bit set works partially. In this scenario there is a proxy involved. The receiver sends a PathReq message, the proxy intercepts it and sends a Path message back to the receiver end. The receiver then sends the Resv message as usual. After that the proxy should send Path messages periodically to the receiver to keep the session alive. However, the current implementation seems to send Srefresh messages to both directions and no refreshing Path messages at all. We haven't designed a solution for this bug yet. It will require some studying of the source code but we expect that it will be easy to fix.

All other test scenarios described in Bauffe's documentation [Bau05] seem to work correctly. If we run into any other bugs that are not listed here we will attempt to fix them, too.

5 Tools

Bauffe made his own tool for testing LRSVP protocol called lrsvpTest. This little program seems to be mostly copy-pasted from KOM RSVP tools sendapi and recvapi. We decided to implement LRSVP extension to these two KOM RSVP programs so people can use these two programs for testing RSVP and LRSVP.

Modifications to these programs are quite small. We only have to add support for the new message types and the LI- and ER-bits. We also decided that if we have time we will sanitize the code in these small programs.

6 Build System

Bauffe's LRSVP implementation didn't integrate it's build system very well with the KOM RSVP package. We plan to move LRSVP changes to be part of the configure scripts. There will be two new switches for that script: --enable-lrsvp and --enable-lrsvp-proxy. First one is used to build LRSVP supported client tools etc. and second one enables building of the LRSVP proxy program.

According to Bauffe's documentation, proxy must be builded separately of the normal daemon, so you can only use one of the above switches at the same time. We will investigate whether this limitation can be removed.

References

Bau05 Bauffe, A., LRSVP extension to KOM RSVP, 2005.

Kar Karsten, M., KOM RSVP engine. URL `http://www.kom.tu-darmstadt.de/en/downloads/software/kom-rsvp-engine/`.

Man06 Manner, J. et all, Localized RSVP. URL `http://tools.ietf.org/wg/tsvwg/draft-manner-tsvwg-lrsvp-00.txt`. 2006.