Requirements Specification

Varain

Helsinki October 12, 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	4.10.2006	Initial version
1.1	6.10.2006	Updated chapter 7

Contents

1 Introduction

This document lists the different requirements set for the Varain software project. The general aim of the Varain project is to add support for the LRSVP protocol in the KOM RSVP software package.

In the beginning of the document the system's interest groups are discussed. After this all the use cases that the project aims to satisfy are listed and analysed. Based on these, the user requirements are specified. Latter part of the document concentrates more on the system rather than on the user. Overview of the system architechure is given as well as list of system requirements. In the last chapter system life cycle is analysed.

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 Universität Darmstadt.

3 Interest Groups

Interest groups of our system consists of four units as illustrated in figure 1. The remote end-host, the proxy server and routers have to support LRSVP. Local end-host sends the LRSVP requests. In every use case there can be (and usually is) many routers, but only one of each end-host type and only one proxy.

Administrator is a person who is resposible for confuguring and maintaining the system. There can be many administrators.

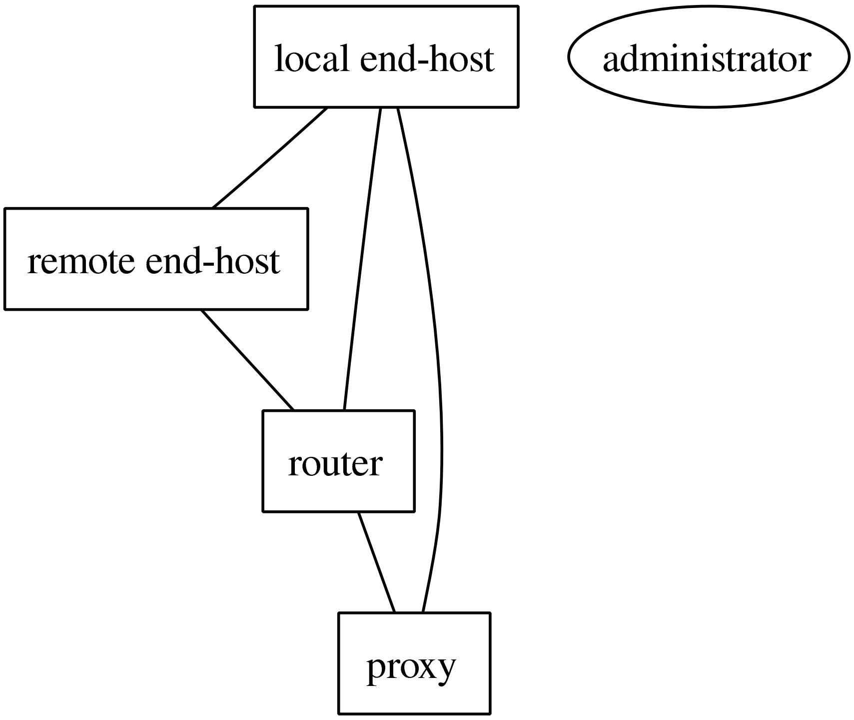

Figure 1: Interest groups.

4 Use Cases

4.1 Local Reservation

Summary:	Reserving entity wants to do a bandwith reservation in local environment.
Preconditions:	-
Triggers:	Reserving entity wants to limit reservation into local environment. This may happen e.g. when the sending end-point can't form an end-to-end RSVP-reservation, so as a fallback it creates a local reservation.
Basic course of events:	1) Reserver sets LI-bit to 1 for reservation requests. 2) Proxy works as the other end-point.
Alternative paths:	When an LRSVP-proxy is lacking from the routing path reservation fails and possibly falls back to RSVP.
Postconditions:	Local reservation has succeeded or failed.

4.2 Downstream Reservation

Summary:	Local end-point wants downstream bandwith to be reserved.
Preconditions:	The remote end-point understands LRSVP Path Request messages.
Triggers:	Local end-point feels the need for downstream reservation.
Basic course of events:	1) Local end-point sends a Path Request message to the remote end-point. 2) Remote end-point triggers a reservation.
Alternative paths:	-
Postconditions:	Reservation has been triggered by remote end-point.

4.3 Local downstream Reservation

Summary:	Local end-point wants a local downstream bandwith reservation.
Preconditions:	A local LRSVP-proxy exists on the routing path.
Triggers:	Local end-point feels the need for a local downstream reservation.
Basic course of events:	1) Reserver sends a Path Request message with LI-bit set to 1. 2) LRSVP-proxy triggers a reservation.
Alternative paths:	-
Postconditions:	Reservation has been triggered by remote end-point.

4.4 Downstream Reservation Tear

Summary:	Receiving end-point suggests tear down of a downstream reservation.
Preconditions:	A downstream reservation exists.
Triggers:	Local end-point feels downstream reservation is no longer needed.
Basic course of events:	1) Local end-point sends a Path Request Tear message to remote end-point. 2) Remote end-point may follow the advice and kill the reservation.
Alternative paths:	-
Postconditions:	A reservation tear has been suggested.

4.5 Updating Reservation

Summary:	Sending end-point asks routers on the path to refresh reservation path information.
Preconditions:	A previous reservation exists.
Triggers:	Sending end-point assumes (or finds out) the reservation path is broken.
Basic course of events:	1) Sending end-point sends a reservation update message with ER-bit set to 1.
Alternative paths:	-
Postconditions:	Reservation path is refreshed.

5 User Requirements

Description	*Priority*
The system shall provide a means for a host to reserve resources using RSVP on routers along a path to a foreign end host even if the foreign end host doesn't support RSVP.	High
The system shall be able to do a fast local repair when the end host's route to the foreign end host changes (in a mobile environment).	Low
The usage instructions of all KOM RSVP's executables shall be updated to include LRSVP-related options.	Medium

6 System Architecture

Our project is based on Adrien Bauffe's implementation of the LRSVP protocol. Bauffe made his implementation on top of the KOM RSVP package, which is an RSVP implementation made by Multimedia Communications Lab at Technische Universität Darmstadt. He didn't get all features working. Our project goal is to make an LRSVP daemon and a proxy using KOM RSVP and we have a possibility to use Bauffe's implementation.

The latest release of the KOM RSVP project is rather old (2004) but we are still able to compile and use it. The KOM RSVP project has been written in C++ and has a hybrid form of object orientation and procedural design. The whole project can be divided into smaller modules. The main parts that have to be modified in order to make the software LRSVP compatible are identified as RSVP daemon, RSVP API and RSVP Message. There are some other classes also to be edited but they need only minor modifications. There is also a completely new module called LRSVP proxy, which is working in Bauffe's implementation so we can utilize his work. More detailed information about KOM RSVP can be found in their technical documentation [Kar00].

7 System Requirements

Description	*Priority*
The system shall be able to run on the hardware that we are given for testing purposes (laptop with a 750 MHz Pentium III processor and 512 MB of RAM).	High
The system has to run on an operating system based on Linux from the 2.6 version family.	High
All the features supported by the vanilla KOM RSVP version 3.0f should also work with the LRSVP-enabled version.	Medium
The system shall support all the functionality related to the LI bit as described in [Man06].	High
The system shall support all the functionality related to the ER bit as described in [Man06].	Low
The system shall support the Path-Request message type as described in [Man06].	High
The system shall support the Path-Request Tear message type as described in [Man06].	High
The system shall provide an API for client applications to be able to make and cancel reservations.	High
The administrator shall be able to configure LRSVP.	High

8 System Life Cycle

The system can be later ported to operating systems other than Linux but initially that will be the only supported platform.

According to its developers KOM RSVP has been developed and tested in Linux 2.4, Solaris 2.6,7 and FreeBSD 4.x using gcc 2.95.3 or gcc 3.0 and it should compile and work on any Linux 2.X and FreeBSD 3.X/4.X platform. According to our tests KOM RSVP also compiles with the early versions of the gcc 3.4.X series but not with any newer ones.

The fact that newer versions of gcc can't be used to compile KOM RSVP might become a problem in the future. Fixing this would possibly require modifying a lot of the existing code in KOM RSVP so we will not get into that in this project.

FreeBSD and Solaris would probably be the first platforms where the LRSVP extension could be ported after the Linux version has been finished.

References

Kar00 Karsten, M., Design and implementation of RSVP based on object-relationships. Technical Report, 2000. URL `http://www.kom.e-technik.tu-darmstadt.de/publications/abstracts/Kar00-2.html`.

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