System Administrators Guide

Varain

Helsinki December 3, 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	1.12.2006	First version

Contents

1 Introduction

In this document we instruct the system administrator to compile, install and set up KOM RSVP with LRSVP support.

KOM RSVP is a server-side application which is meant to be operated by a skilled Unix administrator. Therefore these instructions are written for people who are already familiar with Unix.

2 Installation

You need to get the source package for KOM RSVP 3.0f [Kar] and a patch containing the LRSVP-related modifications mostly written by Adrien Bauffe and partly by the Varain project team. After that you need to patch KOM RSVP and compile it. The procedure is a little different depending on whether you are building KOM RSVP to run as an RSVP daemon or as an LRSVP proxy. The following two subsections provide the instructions for both.

2.1 LRSVP Proxy

1. Get the source package `rsvp-rel3.0f.tgz` and the patch file `lrsvp.diff`

2. Untar the package by running

```
tar zxvf rsvp-rel3.0f.tgz
```

3. Go to the directory `rsvp-rel3.0f/`

4. Patch the source code by executing the following command

```
patch -p1 < /path/to/lrsvp.diff
```

5. Check your GCC version. You need at least GCC version 3 but no later than 3.3.4 to compile KOM RSVP. If your default GCC version is not compatible, you can install, e.g., version 3.3.4 and set the environment variables CC and CXX to point to the compatible version.

For example:

```
export CC=/opt/gcc-3.3.4/bin/gcc
export CXX=/opt/gcc-3.3.4/bin/g++
```

6. Execute the following command

```
./configure --disable-mpls --disable-java --disable-java-api \
--disable-refresh-reduction --disable-dns --disable-onepass \
--enable-virt --enable-active-api --disable-lrsvp \
--enable-lrsvp-proxy
```

7. Execute the following command

```
make dep && make
```

If everything went well, you should now be able to see binaries in the `rsvp-rel3.0f/bin/` directory

2.2 Daemon

1. Get the source package `rsvp-rel3.0f.tgz` and the patch file `lrsvp.diff`
2. Untar the package by running

```
tar zxvf rsvp-rel3.0f.tgz
```

3. Go to the directory `rsvp-rel3.0f/`
4. Patch the source code by executing the following command

```
patch -p1 < /path/to/lrsvp.diff
```

5. Check your GCC version. You need at least GCC version 3 but no later than 3.3.4 to compile KOM RSVP. If your default GCC version is not compatible, you can install, e.g., version 3.3.4 and set the environment variables CC and CXX to point to the compatible version.

For example:

```
export CC=/opt/gcc-3.3.4/bin/gcc
export CXX=/opt/gcc-3.3.4/bin/g++
```

6. Execute the following command

```
./configure --disable-mpls --disable-java --disable-java-api \
--disable-refresh-reduction --disable-dns --disable-onepass \
--enable-virt --enable-active-api --enable-lrsvp
```

7. Execute the following command

```
make dep && make
```

If everything went well, you should now be able to see binaries in the `rsvp-rel3.0f/bin/` directory.

3 Configuration

3.1 RSVPD.conf syntax

```
'interface' <name>
        ['refresh' rate]
        ['virt' virt-addr virt-MTU]
        ['encap' local-port remote-addr remote-port {remote-port}
         ...]
        ['tc' 'none'|'cbq'|'hfsc'|'rate' bandwidth latency]
        ['disable']
        ['mpls']
        ['nompls']

'route' destination netmask gateway iface-name
'api' port
'timer' total-period-in-seconds number-of-slots resolution-in-usec
'threads' number
'session_hash' number
'api_hash' number
'mpls_all'
'nompls_all'
'peer' interface address <distance>

Units

rate -> milliseconds
latency -> microseconds
bandwidth -> bits/s
MTU -> bytes
total-period -> seconds
resolution -> microseconds
load-threshold -> [0,1] float
wtp-threshold -> marks per hour
interface -> ip address
distance -> hop count
all other parameters -> IP address resp. port numbers
```

3.2 RSVPD.conf example

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

4 Executables

4.1 RSPVD

First of all, before you can run RSVPD you have to do the CBQ initialization. Do this by running the following command, as root:

```
rsvp-rel3.0f/tools/cbqinit <device> <bandwidth-in-Mbits/s>
```

This only needs to be done once per boot.

By default, RSVPD will try to read its configuration from `/etc/RSVPD.conf`. Optionally you can specify another location on the command line. There are a few other command line options, too, as listed below. Syntax for the configuration file is specified in section 3.

```
usage: ./RSVPD [options]

Option list:
-h, -?                        print this help
-c configfile                 specifiy configuration file
-o output file                write logging output into file
-l loglevel,loglevel,...      enable given list of loglevels
-d loglevel,loglevel,...      start from 'all' and exclude
                              listed loglevels
for loglevels, choose from:
none,fatal,error,timer,ref,pc,tc,cbq,msg,api,packet,
session,sb,routing,rsrr,config,process,parser,select,
scheduler,hfsc,unused,unused,unused,unused,reduct,
mpls,ns2,short,all
```

4.2 lrsvp_proxy

You first need to create a configuration file `/etc/komrsvp-proxy.conf`. In this file you should specify the “local interfaces”. `lrsvp_proxy` will only support local clients whose requests come from interfaces that are listed there.

The IP addresses of the local interfaces are listed on separate lines preceded with the letters “`LNI `“. For example:

```
LNI 12.12.12.12
LNI 123.123.123.123
```

`lrsvp_proxy` takes no parameters.

4.3 recvapi

You can find several examples of how to use `recvapi` in the Test Plan.

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

4.4 sendapi

You can find several examples of how to use `sendapi` in the Test Plan.

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

References

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