NAME

rsvptrace − Resource Reservations diagnostic utility

SYNOPSIS

rsvptrace [**−h** *hops*] [**−H**] [**−L** *Last-hop*] [**−m** *path-mtu*] [**−s** *timeout*] [**−U|T**] [**−p** *proto#*] *session sender*

DESCRIPTION

Rsvptrace is an RSVP reservation diagnostic tool. It collects a trace of reservation state information for a given session, along the reverse path from a given last-hop node towards a given sender. Each RSVP hop along the reverse path adds its response to the request, and forwards the request to the appropriate previous hop. Upon receiving replies, *rsvptrace* prints out the responses, with any errors found.

OPTIONS

−h Specifies the maximum number of rsvp hops that this diagnostic trace can span, after which a reply will be returned, and diagnosis terminated. The default is 10 hops.

−H

Specifies that the reply is not sent back directly to the requester, but hop-by-hop (HBH) to the Last-hop. The Last-hop node then sends the replies to the requester. The default is to send the reply directly to the requester. This option is useful when the path being diagnosed traverses a firewall.

−m Specifies maximum MTU of the diagnostic packet, in bytes. The default MTU is 1400.

−s Specifies the retransmit timeout period in seconds for the first diagnostic request.

−L Specifies the Last-hop node, as a host name or IP address. Diagnostics commence from this rsvp hop. If this parameter is omitted, the current node is used as the Last-hop.

−U|T Specifies the Session IP protocol to be UDP or TCP, rspectively. The default is UDP.

−p Specifies the Session IP protocol using the numeric IP protocol ID. If multiple U/T/p options are given on the command line, the last option will override the others.

−n Disolays only numeric/dotted-decimal IP addresses, avoiding all reverse name lookups on IP addresses.

session Specifies the RSVP session to be traced, in the format *addr/port*. This is a required parameter.

sender addr/port

Specifies the RSVP sender towards which diagnostic trace will be done, in the format *addr/port*. This is a required parameter.

How it works

A trace starts when a trace request is sent to the Last-hop node, (if the -L parameter is specified) or to the current node. The Last-hop node must implement RSVP. Each node will forward the trace request to the previous RSVP hop corresponding to that session and sender. Before forwarding, each node will add its local reservation state pertaining to that sessio to the trace. When the sender is reached or the maximum scope of the diagnostic is attained, a reply is sent back to the requester.

The completed trace request will normally be unicast directly back to the requesting node, as a trace reply. However, the -H parameter will force a hop-by-hop (HBH) reply mode: the reply is sent hop-by-hop from the replying node back to the Last hop node, and thence to the requesting node. In this case, the route for the reply is created as the diagnostic request travels towards the sender, with each RSVP node adding an entry to a ROUTE object in request message. The HBH reply mode is intended for tracing through a firewall.

An error encountered during the trace may terminate the trace early and force a reply. For example, there may be no PATH state for that sender on a node. Another more complicated situation occurs when the request packet grows too large for the MTU towards the previous hop. The diagnostic packet is then fragmented at a response boundary and one reply is sent back. A trimmed request is then forwarded upstream.

If a request does not receive a reply, *rsvptrace* times out and resends upto three times. The timeout increases exponentially each time. The default timeout value for the first request is 10 seconds.

Rsvptrace collects the replies and prints out the responses out in order.

For further details, please read the reference Internet Draft.

SAMPLE OUTPUT

We present a sample output of a trace below. The session is 224.10.10.10/25000. The sender is 10.0.0.1/8888. There is an FF style reservation in place for the sender in this session.

% rsvptrace 224.10.10.10/25000 10.0.0.1/8888

Rsvptrace from 40.0.0.1 to 10.0.0.1/8888 for session 224.10.10.10/25000
Querying reverse path...
Timeout in 10 seconds...
Reply from 10.0.0.1, 4 hop(s), 4 response(s)

1) 40.0.0.1 [arr time: 14:12:59.478]
[IN->30.0.0.2, OI<-40.0.0.1, Phop->30.0.0.1, timer: 30, DTTL: 0]
[Gen T=[100K(40K) InfB/s 0 65.5K]]

2) 30.0.0.1 [arr time: 14:18:55.866]
[IN->20.0.0.2, OI<-30.0.0.1, Phop->20.0.0.1, timer: 30, DTTL: 64]
[Gen T=[100K(40K) InfB/s 0 65.5K]] [CL T=[20K(5K) InfB/s 0 65.5K]] FF
{10.0.0.1:8888}

3) 20.0.0.1 [arr time: 14:18:55.989]
[IN->10.0.0.2, OI<-20.0.0.1, Phop->10.0.0.1, timer: 30, DTTL: 64]
[Gen T=[100K(40K) InfB/s 0 65.5K]] [CL T=[20K(5K) InfB/s 0 65.5K]] FF
{10.0.0.1:8888}

4) 10.0.0.1 [arr time: 14:18:56.027]
[IN->local, OI<-10.0.0.1, Phop->N/A, timer: 30, DTTL: 64]
[Gen T=[100K(40K) InfB/s 0 65.5K]] [CL T=[20K(5K) InfB/s 0 65.5K]] FF
{10.0.0.1:8888}

The responses are numbered starting at the last hop and ending at the node that sent the reply. (in this case the sender itself). Here is a brief explanation of the output.

1) 40.0.0.1 [arr time: 14:12:59.478]
The first line indicates the specific node at which this response was recorded and the arrival time of the request.

[IN->30.0.0.2, OI<-40.0.0.1, Phop->30.0.0.1, timer: 30, DTTL: 0]
The second line lists the incoming interface for that *sender* on the specific *session*, the outgoing interface being traced, and the previous hop. It also gives the value for the refresh timeout being used at that RSVP node. The DTTL value is the number of non-RSVP routing hops between the downstream RSVP router and the RSVP router that added this response.

[Gen T=[60(80) InfB/s 30 80]], [CL T=[1(2) InfB/s 3 4]] FF
The third line lists the Effective Tspec, the flowspec, and the style of the reservation in place, if any, on the outgoing interface. If the reservation is a merged reservation this will be indicated with the M symbol. If there is no reservation, only the Tspec of the sender will be listed.

{10.0.0.1:8888}
The fourth line lists all the senders sharing this reservation. If the reservation is shared, multiple senders may be listed on successive lines.

SEE ALSO

rtap(8), rstat(8), rsvpeep(8), rsvpd(8)

References

L. Zhang and A. Terzis, "RSVP Diagnostic Messages", Internet Draft - draft-ietf-rsvp-diagnostic-msgs-03.txt, April 1997.

Implementation differences with the specification

This release does not strictly implement the internet draft specification for RSVP diagnostic messages. The response address (where the reply is sent to) in the specification is an IP address, identifying the target system. The response address in this implementation includes the ephemeral UDP port of the trace program. The final hop RSVP router (from where the reply is sent) will send the UDP diagnostic reply to the waiting trace program. The advantages are that muliple traces can be simultaneously run from the same system and this is independent of whether Rsvpd is running on the system or not.

BUGS