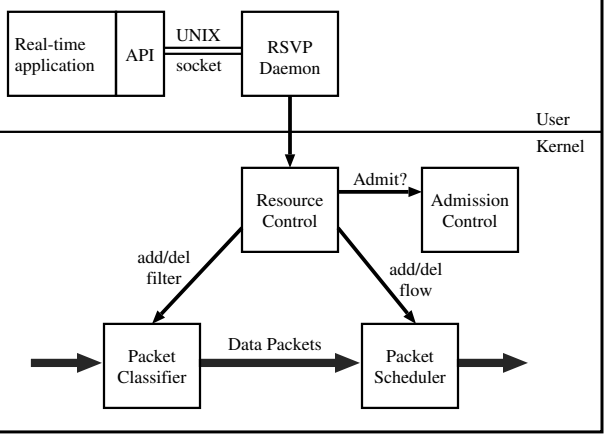
RSVP Implementation Status

Steven Berson
USC Information Sciences Institute

Overview of Implementation Status

- System Model
- Requirements
- What is implemented
- Testing environment
- What is not (yet) implemented



Requirements

- Requires IP Multicast (with modifications)
  - + query for multicast routes
  - + query kernel for list of VIFS
  - + send packets to particular interface, i.e. bypassing routing
- Additional kernel modification needed to pass RSVP protocol messages to daemon

What is implemented - I

- Working Prototype of RSVP for SunOS 4.1.x
  - + Runs as a user level daemon
  - + Same kernel/daemon runs on routers and hosts
- 3 Reservation Styles DF, FF, and WF
- Merging of Reservations
- Dynamic adaptation to changes in group membership, flowspecs
- Supports tunnels through non-RSVP routers

What is implemented - II

- API
  - + Register a sender
  - + Register a receiver
  - + Make reservations
  - + Asynchronous error delivery
- Tools
  - + rtap - tool for testing API
  - + mmcc - multimedia conferencing tool
  - + nv - video display tool

DARTnet Testing Environment

- dmap - program to display RSVP state of DARTnet routers
- tcpdump - tool to dump contents of RSVP packets
- RSVP meta-daemon - uses multicast to send commands to all RSVP sites; uses:
  - restart a new daemon
  - adjust the debug level
  - distribute a new version of daemon

What is not implemented

- Admission control
  - earlier version had partial implementation
  - a new partial version is forthcoming
- General flowspecs/filterspecs
  - IP address/port number now used
  - variable length (mask, value) pairs to be used
- Support for new ("prune-beta") multicast
- Teardown messages