Internet: Best effort service does not support real time applications well
Integrated services: extension to internet architecture to support integrated services
Like real time applications e.g., Packet voice, packet video, distributed simulation.
ISPN (Integrated Services Packet Network): network which supports integrated services.

Integrated services:
Adv: an end system can request a particular quality of service (QoS)
e.g., bounded end-to-end queuing delay, for a particular data flow.
Difficulty: providing QoS requires reservation of network resources in routers along the path(s)
of the flow as well as in the end hosts.

In order to provide a requested QoS, the nodes of an ISPN must perform
reservation setup,
admission control,
policy control,
packet scheduling,
and packet classification functions.

Figure 1 illustrates these functions in an ISPN router.
A reservation setup protocol is used to pass the QoS request originating in an end-system to each router along the data path, or in the case of multicasting, to each router along the branches of the delivery tree. In particular, **RSVP was designed to be the reservation setup protocol for an ISP**.

An RSVP reservation request is basically composed of

- **Flowspec** (defines the desired QoS)
- **Filter spec** (defines the subset of the data stream, i.e., the flow, that is to receive this QoS.)

### 2. Admission Control
At each node along the path, the RSVP process passes a QoS request (flowspec) to an admission control algorithm, to allocate the node and link resources necessary to satisfy the requested QoS. If admission control accepts the request, the necessary state is established for it; otherwise, an error message is sent.

### 3. Policy Control
Before a reservation can be established, the RSVP process must also consult policy control to ensure that the reservation is administratively permissible.

### 3 RSVP Project: Final Report

### 4. Packet Scheduler
Assuming that admission control and policy control both succeed, the RSVP program installs state in the local packet scheduler or other link-layer QoS mechanism, to provide the requested QoS. The packet scheduler multiplexes packets from different reserved flows onto the outgoing links, together with best-effort packets for which there are no reservations.

### 5. Packet Classifier
The RSVP process also installs state in a packet classifier, which sorts the data packets forming the new flow into the appropriate scheduling classes. The state required to select packets for a particular QoS reservation, known as a filter, is specified by the filter spec.

### 1.2 RSVP Architecture
RSVP was designed to provide

- robust,
- scalable,
- flexible,
- and heterogeneous resource reservation

These design requirements led to a number of basic architectural features:

1. a multipoint-to-multipoint communication model,
2. receiver-initiated reservations, (adv: easy scaling for large number of receivers)
3. cached (“soft”) state management in routers, and
4. separation of reservations from routing.

### 2. OVERVIEW OF THE RSVP PROTOCOL

#### Data Flow Definitions
Flow definition is divided into two parts, the session definition and the filter spec.
Session specifies the destination and the filter spec specifies the “rest”.

**RSVP Messages**
RSVP creates and maintains state by periodically sending control messages in both directions along the m-to-n data paths for a session. RSVP messages are sent as IP datagrams and are captured and processed in each node -- router or host -- along the path(s), to establish, modify, or refresh state.
There are two primary RSVP message types: **Resv** and **Path**.
**Resv** (Reservation request) messages travel upstream (opposite to data stream)
**Path** message is initiated by a sender and travels downstream, to create *path state* in each node.
**ResvTear** (Reservation Teardown) messages
**PathTear** (Path Teardown) messages
**ResvErr** (Reservation Error) messages
**PathErr** (Path Error) messages

**RSVP State**
contains information about Reservation State and Path State

**Distributed State Management**
RSVP is designed to create “soft” state, which times out if it is not refreshed soon enough. The only permanent state is in the end nodes. The algorithms for refreshing and timing out state are a fundamental aspect of RSVP.

### 3. RSVP Design Issues

**Data Selection**
The fundamental design decision was made that:
* Routing determines *which* packets are forwarded, but
* RSVP determines *how* they are forwarded,

**Heterogeneity**
RSVP supports
“downstream heterogeneity”: heterogeneous QoS requests from different receivers.
This requires merging flowspecs when **Resv** messages are forwarded,
“upstream heterogeneity”: senders with heterogeneous traffic flows

**Killer Reservations**
The “*killer reservation*” (KR) problem is a denial of quality-of-service that can result from merging two different flowspecs; thus, it is a result of heterogeneity.

**Non-RSV Regions**
To be useful, RSVP must be deployable in the real Internet. Deployment will be gradual, and at any time there will be paths and regions of the Internet that do not support RSVP. Therefore, RSVP was designed to work transparently through arbitrary non-RSVP-capable (*non-R*) routers.

**Liveliness and Local Repair**
Refresh messages make the RSVP state self-healing. If a route changes, for example, the next refresh will (try to) re-establish a reservation along the new path.

Source:
[http://www.isi.edu/rsvp/rsvp2.project.summary.html](http://www.isi.edu/rsvp/rsvp2.project.summary.html)
Some other references:

http://www.isi.edu/div7/rsvp/overview.html
http://citeseer.nj.nec.com/zhang93rsvp.html
http://www.ietf.org/rfc/rfc2205.txt