Negotiation Interface

Scott Shenker
Xerox PARC

IETF, San Jose
12/5/94

Negotiation Interface

- **This is how you invoke service**
  - **What the host says to the network**
  - **What the network says to the host**
  - **Host interface, not application interface**
- **Examples:**
  - **Host specifies level-of-service**
  - **Host specifies desired end-to-end delay/jitter**
  - **Host specifies level-of-service and network tells host the resulting end-to-end delay/jitter**
- **What kinds of negotiation interfaces can and should RSVP support?**
- **Comments**
  - **Question only interesting for delay-like services**
  - **Involves both service model and RSVP issues**
  - **This talk will not attempt to be fully general**

Current RSVP: One-Pass

Mechanism:

- **RESV messages carry service request through network from receiver to source(s)**
- **Reservations are established on each link in accordance with the service request**
- **Network notifies receiver of admission control rejections**

Functionality:

- **Host can specify desired level of service at each hop**
- **Host cannot feasibly specify end-to-end delay/jitter**
- **Host does not know the end-to-end delay/jitter that results from a given level of service**

Design Question: Is this enough?

- **Fine for services which have no quantified end-to-end behavior (e.g., priorities, level-of-effort predictive)**
- **Does not support Guaranteed service (and other delay bounded services) very well**
  - **Why offer delay bounds if hosts don't know what they are?**
- **Reduces to a service model question, not discussed here**
- **This talk will focus on possible alternatives if the answer is "not enough"**

Two-Pass Approaches

Mechanism:

- **Control messages go from receivers up to senders and then back down (or vice versa)**
- **Initial reservations are made on first pass, and then are relaxed on second pass**
- **Full Two-Pass: (e.g., Tenet)**
- **Merged Two-Pass: use RESV and PATH messages (not designed)**

Functionality:

- **Host can specify desired end-to-end delay/jitter in negotiation interface**
- **Network can tell host the end-to-end delay/jitter that results from a level-of-service request**

Problems:

- **Effort/performance tradeoffs are not visible**
- **Significant additional complexity**
- **Troubling design and architectural issues**
  - » **Major change in nature of soft-state**
  - » **Either RSVP knows about service, or service knows about RSVP**

One Pass with Advertising

Mechanism:

- **Each service posts "advertisements" at each router (part of service definition)**
- **PATH messages compose these advertisements along the path and give them to host**
  - » **sum of instantaneous delay/jitter**
  - » **sum of delay/jitter bounds**
  - » **"rough" delay/jitter bounds**

Functionality:

- **Host can determine the end-to-end service that will result from each level-of-service request**

Advantages

- **Effort/performance tradeoffs are visible**
- **Little additional complexity**

Problems:

- **Leads to overreserving for Wildcard Filters**
- **Requires PATH messages (routing later)**
- **Hard to reroute transparently**
- **Rough granularity**

Recommendations (Intserv and RSVP)

1. **One-Pass is "not enough" for now**
   - **market may decide against delay-bounded services later, but don't bake that decision into RSVP architecture now**
2. **Make "advertisement" a basic part of the service definition**
   - **"advertisements" are really a general characterization of service, which must be used by two-pass mechanisms as well as OPWA**
3. **Prototype OPWA**
   - **Prototype implementation**
   - **Provide QM that uses OPWA**
   - **Report back next time**
4. **Investigate two-pass approaches**
   - **Document why not the full two-pass approach**
   - **Design merged two-pass approach**
   - **Investigate architectural issues**