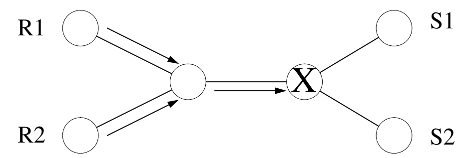
for D
Control

Lee Breslau

Why not do the obvious?



Returning an error and dropping the request limits users to those services available at all hops

Goals

Allow use of different services at different points along a path

Allow heterogeneity across users

Provide feedback to users (e.g., error messages)

No persistent error messages (when users are ‘well-behaved’)

Four Options

1. Return an error message and drop the request
2. Return an error message and forward the request
3. Reservation scripts
4. Advertise installed service

(1) Return an error and drop the request

Simple to implement

Receivers can only use services that are available end-to-end

(2) Return an error and forward the request

Simple to implement

Provides requested service where it is available and best-effort otherwise

Persistent error messages

(3) Reservation scripts

Allow receiver to specify router actions on admission control failures

Examples:

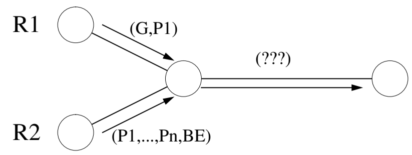
- "Try guaranteed, if that fails try predictive level 1, if that fails return an error"
- "Try each level of predictive service, if they all fail use best-effort"

Increases functionality by allowing users flexibility in expressing their service requests

Reservation Scripts (cont.)

Merging problems and complicated error processing

- how are scripts merged?



(3a) Constrained Scripts

Each user chooses 1 of 2 possible router actions on admission control failures:

- use the best available service and forward the request, or
- return an error

Service request contains flags indicating whether either of these 2 options was selected by a downstream receiver

- logically 'OR' the flags to merge requests

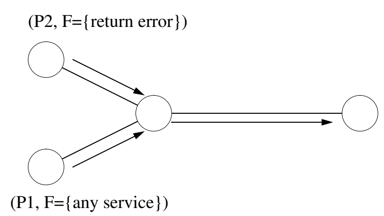
Constrained Scripts (cont.)

If the first flag is set, provide the best available service (real-time or otherwise) when requested service is not available, and forward the original request upstream

If the second flag is set, return an error when the requested service is not available (and turn off the flag if the request is forwarded upstream)

Constrained Scripts (cont.)

To ensure correct behavior, level of service must be associated with each flag



Reservation Scripts -- Extensions

Specify router behavior on merge failures

Extend set of actions that routers can specify

- use best available service
- use best available real-time service
- return an error

(4) Advertise installed service

No error messages on admission control failures

Router action: when requested service is unavailable install best available service and forward request upstream

Advertising messages include minimum level of service installed along a path

Advertise installed service (cont.)

No persistent error messages

Simple router processing

Accommodates user flexibility and heterogeneity

Depends on advertising

Summary

First 2 options are simple but limit functionality and lead to refreshed error messages

Reservation scripts extends functionality at the expense of additional complexity

Advertising installed service eliminates persistent error messages and complex merging/processing, but depends on advertising

Questions

How much functionality/flexibility to provide?

Allow composition of different real-time services end-to-end?

Do applications need to know the service they are receiving at every hop?

If so, best handled through individual error messages or a special mechanism (e.g., probe)?

If we opt for persistent error messages, are there ways to damp them?