Extending the Internet Service Model

Integrated Services Working Group

Extending the Service Model

We need to address three basic issues:

1. What end-to-end data delivery services should the network provide?
2. What do switches need to do in order to deliver those services?
3. How do flows request the desired service from the network?

What end-to-end data delivery services should the network provide?

A Proposed Service Model:

- ASAP, Guaranteed, Predictive

Put in larger context:

- Consider other possible services
- All consistent with basic architecture

Application/Service Taxonomy

There are three fundamental attributes to network service:

- Bandwidth
- Delay
- Loss

Postpone the issue of loss for a few slides.

Consider bandwidth and delay as being either controlled or uncontrolled.

Classify applications by whether they need bandwidth and/or delay controlled.

	Uncontrolled Delay	Controlled Delay
Uncontrolled Bandwidth		
Controlled Bandwidth		

Fill in table:

- Identify motivating applications
- Propose appropriate service

Uncontrolled Delay and Uncontrolled Bandwidth

Elastic Applications:

- Can gracefully adapt to variable per-packet delays
- Can gracefully adapt to variable bandwidth
- e.g., traditional data applications

Proposed Service: ASAP (best-effort)

- Multiple classes of ASAP service
- Not subject to admission control
- To be defined: fairness, congestion notification, etc.

Uncontrolled Delay and Controlled Bandwidth

Time-Constrained Bulk Transfers:

- Can gracefully adapt to variable per-packet delays
- Can gracefully adapt to variable bandwidth, but need an assured minimal amount of bandwidth

Proposed Service: Assured ASAP

- Top level ASAP service with an assured minimal available bandwidth rate
- Subject to admission control
- To be defined: fairness, congestion notification, etc.

Controlled Delay and Controlled Bandwidth

Standard Real-Time Applications:

- Need bound on per-packet delays
  - Nonadaptive: setting playback point
  - Adaptive: *a priori* estimation of delays
- Have an intrinsic bandwidth requirement
- Either tolerant or intolerant of occasional violations of delay bounds

Proposed Services: Predictive & Guaranteed

- Guaranteed: worst-case upper bound
- Predictive: fairly reliable upper bound
- Both give speed-of-light lower bound
- Subject to admission control
  - Flow specifies traffic characteristics

Controlled Delay and Uncontrolled Bandwidth

Variable Quality Real-Time Applications:

- Need bound on per-packet delays
- Can function at different throughput rates
- Assumed tolerant of occasional violations of delay bounds

Proposed Service: Variable Bandwidth Predictive

- Fairly reliable upper bound on delay
- Variable bandwidth with assured minimum
- Subject to admission control
- Predictive is a special case of VBP

Design Issues for VBP

Bandwidth Variation:

- Relatively static traffic hierarchy
- Network drops layers it cannot carry
- Drop preference field indicates which layers network should drop first
- ID calls these “expendable” packets

Traffic Specification:

- Does the flow specify (and the network police) the traffic characteristics of each layer?

Dropping Algorithm:

- Drop at uniform drop preference level, or at a per-flow level, or in a more ad hoc fashion?
- What is the fairness criterion for dropping?

To be discussed next time....

Controlled Delay Services

Separate specifications for traffic and quality of service

Traffic Specification: (TSpec)

- Token bucket rate and depth
- Policed by network at edge
  (and at source merge points)
- More complicated for VBP....

Quality of Service Specification: (QSpec)

- Type of assurance (P or G)
- Min (sol) and max end-to-end delay

Dropping

Preemptable Packets:

- In any service class
- Can be dropped when service commitments are in danger of being violated
- Dropping is assumed to be rare and sporadic

Expendable Packets:

- Only in VBP service
- Steady-state dropping is to be expected

Service Taxonomy

	Uncontrolled Delay	Controlled Delay
Uncontrolled Bandwidth	ASAP	VBP
Controlled Bandwidth	Assured ASAP	Predictive & Guaranteed

Not as different as it looks:

- Diagonal elements are what we have concentrated on
- Off-diagonal elements are not entirely new

Where is Link-Sharing?

Assured ASAP is a per-flow link-share

Institutional Link-Sharing:

- Not part of application visible service model
  - network management interface
  - not an end-to-end service
- Will be part of router requirements
- Need more discussion of goals and policies

Discussion Items

Two Questions:

1. Is there consensus that we need not include any other services?
2. Is there consensus that we should throw something out?

Outline

1. What end-to-end data delivery services should network provide?
2. **What do switches need to do in order to deliver those services?**
   - **Specification of per-hop services**
   - **Subnets**
3. How do flows request the desired service from the network?
4. Other Issues

Specifying the Per-Hop Service

Will focus only on P and G service

- ASAP and AASAP trivial to add
- VBP too new, but also is based on P

Two Issues:

- Characterizing the required behavior of a switch (router requirements)
- Deriving the end-to-end quality of service from the per-hop behaviors

Specifying the Per-Hop Service

Guaranteed:

- Fluid service rate r (for a particular flow)
- Link error parameters A_i and B_i
  - $\frac{A_i}{r} + B_i$ bounds deviation from fluid model
  - Fluid model: idealized WFQ algorithm
- Admission control algorithm based on worst-case analysis

Parekh's result turns this into end-to-end delay bound

- $\frac{b}{r} + \sum_i \left(\frac{A_i}{r} + B_i + L_i \right)$

Specifying the Per-Hop Service

Predictive:

- Fixed set of globally labeled predictive classes
- Several local levels of predictive service
- A monotonic mapping from the global classes to the local levels
- Each local level is associated with a fairly reliable delay bound
- Admission control algorithm based on predictions

End-to-End delay bound is sum of per-hop delay bounds

- Controversial
- Discussed at some later time....

Router Requirements

These per-hop service descriptions will form the basis of the future router requirements specification.

Special Challenge:

- Validating predictive admission control algorithms

Subnets

Should look like ordinary router to network:

- Supporting IP means supporting the service model
- Supporting the service model means conforming to the router requirements

Special Issues:

- Mapping the IP service model onto the subnet service model (e.g., IP over ATM)
- Subnet-specific QoS's
  - Wireless: corruption
  - Others?
- Legacy subnets and other nonconforming network elements

Nonconforming Subnets

Given a specific service request, a router at the entrance to a nonconforming subnet can specify the degree of *reliability* of that QoS being met by the subnet.

This reliability measure:

- is merely a guess
- allows flows to evaluate, at least crudely, if the service is worthwhile
- e.g., a ranking from 1-10, and *Don't Know*

Goal is to allow incremental deployment of QoS-capable network elements

Outline

1. What services should network provide?
2. What do switches need to do in order to deliver those services?
3. **How do flows request the desired service from the network?**
   - **Reservation interface**
4. Other Issues

Up to now, completely uniform

- everybody needs to comply
- single service model
- single router requirement

How you request need not be uniform

- can depend on sender or receiver
- each mechanism needs an interface
- need not be same

Working with RSVP paradigm

- Talk about receiver initiated interface
- Fully expect that there can be a sender-initiated interface

I am <u>not</u> talking about the API

- low-level interface
- carried by RSVP to each switch

Basic Problem

Local Behavior vs. Global Requirements

- QoS is an End-to-End Concept
- Reservation is a Per-Link Concept

How do we ensure that the set of per-link reservations meets the end-to-end service needs of a receiver?

Who does the mapping between the global requirements and the local service?

Receiver-Oriented Reservation Interface

Two approaches:

- Receiver specifies the desired end-to-end QoS and the network figures out what per-hop service to deliver
  - *Global to Local mapping by network*
- Receiver specifies the desired per-hop service after the network has told it the mapping between that request and the resulting end-to-end QoS
  - *Global to Local mapping by receiver*

We are pursuing the second approach

- Come to Thursday's RSVP WG meeting

Reservation Interface

Basic Form:

- minimal acceptable reliability
- type-of-assurance (G, P)
  - G: G-rate
  - P: P-class
- Traffic specification
  - Token bucket rate and size

<u>Bells and Whistles: to be considered later....</u>

Requests for a range of services:

- G-rate range $[r_{min}, r_{max}]$
- Predictive class range $[P_{min}, P_{max}]$

Nonuniform Predictive Classes:

- Uniform predictive reservation an artifact of the particular mechanism
- We can do better

Reservation styles:

- RSVP introduced the notion of reservation styles to describe how resource reservations should be aggregated
- Must be incorporated into the reservation interface
- What do the specifications mean for multi-source flows?

Outline

1. What services should network provide?
2. What do switches need to do in order to deliver those services?
3. How do flows request the desired service from the network?
4. **Other Issues**
   - **Implications of multicast**
   - **Unresolved Issues**
   - **Discussion**

Multicast

Two opposing goals:

- Use shared multicast tree
- Accommodate heterogeneous QoS requests from receivers

Key Question:

- When can two service requests be merged into a shared tree?

Answer:

- Merging can occur as long as neither receiver suffers as a result

Formalization: (per-hop comparisons)

- Define a partial ordering of the end-to-end services
  - based on universal perception of quality
- Define a consistent partial ordering of the per-hop services
  - i.e., the local to global mapping must be monotonic with respect to partial orders
- Require that admission control respects per-hop ordering relations
- Require that the installed per-hop reservations along a receiver's data path must be greater than or equal to those that would result from that receiver's own reservation request

Ordering Relations

Natural End-to-End Orderings:

- TSpec's are ordered by rate and bucket size
- P ordered by upper delay bound
- G ordered by upper delay bound

Resulting Per-Hop Orderings:

- P: ordered by class
- G: ordered by rate

Unresolved Issue:

- Is G $\geq$ P (for same bound)?
  - Actual delays are larger under G

Potential Problem

If P and G are not comparable:

- Mcast tree is either all P or all G
- Severely limits the degree of heterogeneity

Possible Solutions:

- Impose the ordering $G \geq P$
  - mechanistic issues
  - local vs. nonlocal bounds
- Explicitly hybrid reservations
- Multiply scheduled packets
- Any suggestions?

Unresolved Issues

Service Model Formulation:

- What should we add?
- What should we leave out?

Service Design:

- VBP
- Fairness and congestion notification in ASAP service
- Institutional link sharing

Detailed Design Issues:

- Fancier reservation interface
- Reservation styles
- Multi-Source flows
- Number and meaning of global predictive classes
- Designing predictive admission control
- Validating predictive admission control
- Local/E2E mapping for predictive
- Partial orders and hybrid reservations
- Explicit buffer allocation
- Subnet requirements (are they different?)
- Nonconformance *reliability* computation
- .....