Resource Management and Differentiated Services

Lixia Zhang
UCLA Computer Science Department

with input from many others

May 1998
Network QoS support seen from 10,000 feet:

1. Define packet treatments at switches/routers
2. Control the amount of resources allocated to each treatment class
3. Sort packets into classes
What this talk is about

♦ How to provide scalable, robust, and manageable resource management
  ‣ Ongoing research/development effort
  ‣ incorporating others visions/ideas

What this talk is *not* about

♦ how many different traffic classes needed, or how to set the TOS field value
  ‣ the charters of IETF intserv & diff-serv Working Groups
What is pushing diff-serv effort

♦ A market need since yesterday: simple mechanisms that can be *quickly* and *incrementally* deployed to provide differentiated services
  ‣ no one wants to sell “bad” services
  ‣ everyone wants to sell varying levels of “good” services
♦ Various doubts on feasibility of intserv framework
  ‣ Complexity?
  ‣ Scalability?
  ‣ Quick deploy-ability?
How and how much do diff-serv & intserv differ?

♦ Where to start:
  - defining end-to-end services, vs.
  - defining packet treatments at individual components

A sidenote:
  - IP started by defining hop-by-hop forwarding, rather than end-to-end delivery service
How and how much diff-serv & intserv differ (II)

♦ How to control the amount of resources allocated
  ▸ end-to-end QoS support requires end-to-end signaling
  ▸ per-hop treatment can work with either static configuration or dynamic signaling

♦ How to sort packets into classes
  ▸ old RSVP way:
    • identify individual flows
    • map packets of each flow to proper traffic classes
  ▸ diff-serv: use TOS field as class ID
    • pre-classified somewhere
Network resource management

Emerging model:
- interconnects of administrative domains
- a priori bilateral agreement between neighboring domains
- each domain responsible for its internal resource management & usage
An analogy to global routing

- Hierarchical
  - needed for scaling
  - needed for administrative control
  - different granularity at different levels
- routes are *pre*-computed (or *pre*-configured)
- concatenation of hop-by-hop forwarding provides end-to-end data delivery
- routes dynamically adjustable
  - adapt to topology/policy changes
A proposed picture for Scalable QoS support

- Two-tier resource management
  - inter-administrative domains
  - intra-administrative domains

- Inter-domain: pre-negotiated neighboring relation
  - infeasible to set up business relation upon every new flow in real-time?

- Concatenation of bilateral agreement leads to end-to-end QoS delivery paths

- Amount of resources adjustable
  - adapt to demand/policy/topology changes
Resource manager: Bandwidth Broker (BB)

♦ A logical entity residing in each administrative domain
  ♦ Managing internal demands & resources according to the policy database (who can do what when)
  ♦ setting up & maintaining bilateral agreement with neighbor domains
    • bookkeeping how much traffic entering which border router & going out which border router

♦ today’s BB: network administrators & operators
  ♦ would like to automate over time

“A Two-bit differentiated services architecture for the Internet”
Nichols, Jacobson, Zhang
draft-nichols--diff-arch-00.txt, November 1997
An overall picture

“Keep complexity at edges, leave the core simple”
- peripheral domains may manage internal traffic and resources in any way they wish
- border-crossing packets carrying right TOS value and treated diffserv way

- ingress border routers policing
- (egress border routers shaping)
Some of the questions

1. How does a leaf domain BB know the total local demands for each egress border router?

2. How does a transit domain BB map its inter-domain commitment to internal resource allocation?

3. How much (& what) state must BB keep?

4. How much (& what) state must a router keep?
   - Router in leaf domains
   - Router in core networks
Choices for implementation

- adequate provisioning
  - eliminate Questions 2, 3, & 4

- manual configuration
  - not that different from static routing

- using some setup protocols
  - inter-domain: BB-to-BB
  - Intra-domain: RSVP as a ready candidate
BB is assumed to have adequate knowledge about internal demand; may readjust the allocation over time.

BB & BB instruct their edge devices how to shape/police.

Indicating additional configurations (shaping/policing) if the domain cannot solely rely on provisioning.
An example of using RSVP in a local domain

**BB** may pre-reserve adequate bandwidth with BB to avoid readjusting the inter-domain allocation everytime (the actions indicated by the dashed lines)
“Tunnel” RSVP messages between leaf domains

♦ Why “tunnel through”: do not want intermediate routers to see/act on end-to-end RSVP messages
♦ One way of doing it

♦ drawback
   ▶ assuming both ends using RSVP internally
   ▶ intra-domain signaling msgs crossing boundaries

“A Framework for End-to-End QoS Combining RSVP/Intserv and Differentiated Services”
draft-bernet-intdiff-00.txt
Intra-transit domain implementation

Choices of implementation

♦ provisioning
♦ manual configuration, or SNMP
♦ use an automatic setup protocol, such as RSVP

How to use RSVP in core networks

♦ border routers behave as sources & destinations for ingress and egress traffic
  ♦ similar idea discussed in PASTE draft
Here is a picture

A transit domain

Border routers

- Set up an RSVP session for each ingress flow
  - RSVP msgs tell each router along the way how much to reserve
  - Routers classify packets by TOS field
- Reservations follow routing changes automatically
Some of FAQ’s

♦ Is this sender or receiver driven?
  ▶ At BB level: yes to all, sender domain BB, receiver domain BB, possibly 3rd party BB

♦ How to handle allocation for multicast traffic?
  ▶ See above
    • details being worked out

♦ relation between the two levels of resource control?
  ▶ Inter-domain (BB) level:
    • independent from whether one does anything internally
  ▶ intra-domain: local decision
Summary: One possible picture for **diff-serv resource management**

- **two-level hierarchy**
  - inter-domain management
    - currently human
    - automate over time; BB as one proposal
  - intra-domain management: multiple possible choices
    - provisioning, manual-configuration, SNMP, RSVP

- **packet classification**
  - cross-domain traffic: classified by bits in TOS field
  - leaf domain: one’s own choice

- Work underway for a prototype implementation