Border Crossings: Evolving Interdomain QoS in a Heterogeneous World

ITU-T Workshop on QoS

Kathleen Nichols
nichols@pollere.com

June 14, 2006

Talk Overview

1. Where we've been

2. Where are we now?

   (a) Domain Resource Managers
   (b) Small number of classes
   (c) "SS7-style" interactions
   (d) Emerging work on how to define and compute metrics across multiple domains

3. Getting to E2E

4. What should next steps be and how should they be taken?

Beginnings of E2E Differentiated Services QoS

Differentiated (or “differential”) Services started in the IRTF E2E WG. Clark and Jacobson each had ideas on how to provide a range of services from simple mechanisms.

Van’s ideas were captured in a November 1997 internet draft, now available as RFC 2638. The pdf version has Dave Clark’s slide response with a proposal very close to where the subsequent DiffServ WG ended up.

RFC 2638 laid out a Bandwidth Broker approach to service within and across domains. Features include a mixed use of (unspecified) signaling and an approach where some allocations are “ask” and others “don’t ask.”

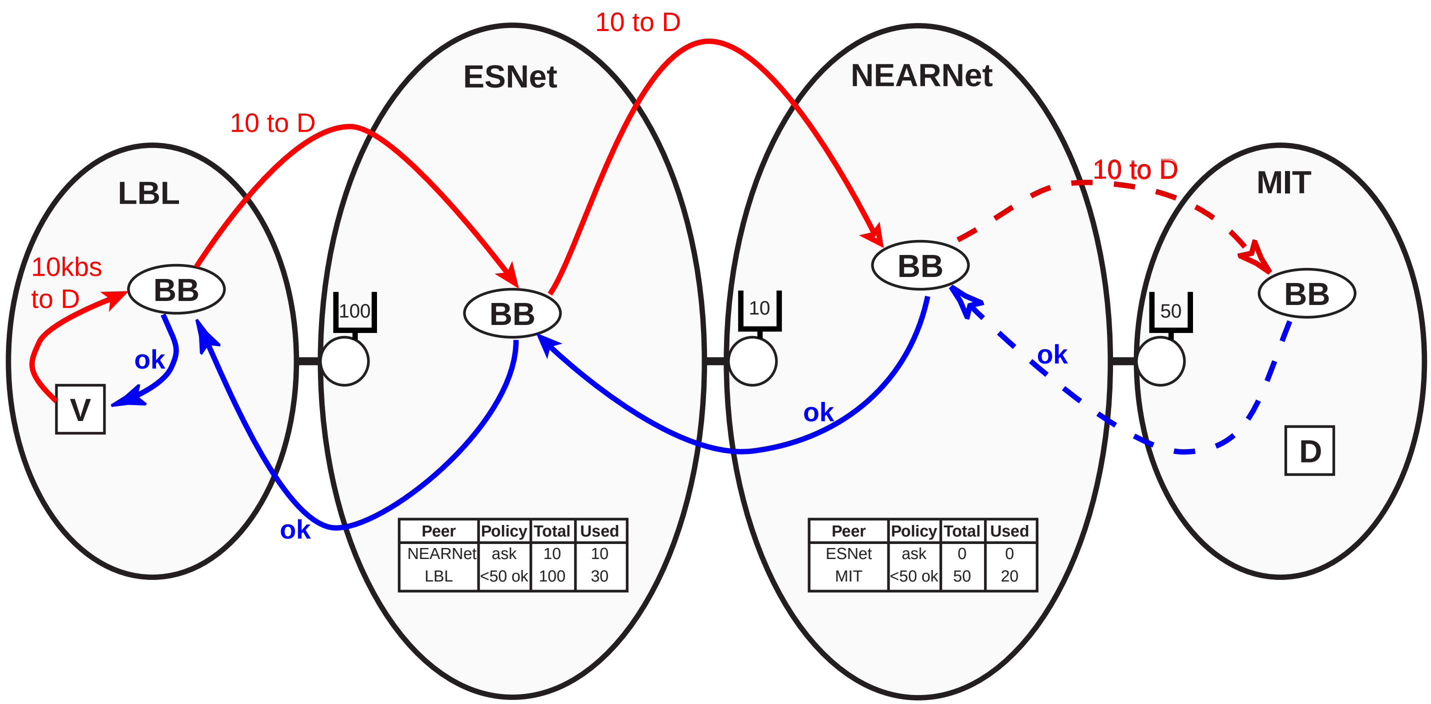
In the early Bandwidth Broker architecture, signaled messages pass from domain to domain, vaguely SS7-like.

RFCs 2474/5 lay out the basics of IETF DiffServ with a single domain focus, but the interdomain ideas were there from the start. I gave DiffServ talks in 1999 (including one at an ETSI workshop) including discussions of how to connect domains.

(Go back even farther to Paul Baran’s ideas and the TOS wording in RFC 791.)

E2E DifS: A Picture from the Past

This picture appeared in the pdf version of RFC2638 and shows a combination of approaches to deliver E2E differentiated services.



V asks LBL's BB for a specific level of service. The request is passed along between BBs. May or may not require a signal to MIT.

E2E DifS: State of the Art

- Clear emergence of **Domain Resource Managers** (of various flavors), **"SS7-like" messaging** (employing SIP/XML/SOAP/NSIS), and **a small number of useful classes** or per-domain behaviors (PDBs, RFC 3086) used to deliver services (ITU-T Y.1541). Not that much change since the mid-90's Bandwidth Broker architecture, but today, "the devil is in the details."

- Working those details are EuQoS, IPsphere Forum, Multiservice Forum's Bandwidth Manager Architecture, Alliance for Telecommunications Industry Solutions' NGN E2E QoS contribution at Global Standards Collaboration-10 (strongly VoIP solution based), Packetcable server-based architecture (also strongly VoIP)

- EuQoS (and MSF) provides an excellent basis. Quibble over how much path-orientedness is needed and the "PHB continuity along the AS path" notion, and some implementation choices. Also would like more work on making the framework interoperate with other signaling or no signaling. Not a disagreement with the fundamental EuQoS framework, just aspects of the current prototype direction (in my opinion).

Products and Prototypes: Domain-based Resource Managers

- Commercial products: Juniper's SDX, Operax Bandwidth Manager
- Specifications include the MSForum's bandwidth manager. ("The MSF bandwidth manager functionality is an example of an off-path or path-decoupled topology aware admission control system framework")
- Multiple domain prototype: EuQoS
- Field prototype: Telcordia's Bandwidth Broker for mobile networks
- Laboratory prototype: Lockheed Martin (draft-nichols-dcpel-strawman-arch and what I'm working on)

Rough Consensus: A Small Number of Useful Classes

- **RFC 3086** defines Per-Domain Behaviors, a specification matched to domain-oriented control (e.g. for SLS). PDBs are the behavior experienced by a particular set of packets as they cross a Diffserv cloud and specify both the forwarding path treatment and the edge rules for its traffic aggregate.

- ITU-T Recommendation **Y.1541**: "the number of classes must be small for implementations to scale in global networks" and specifies six QoS classes that would have global visibility. **Appendix VI** of Y.1541 notes that the IP QoS classes should be mapped to PDBs. ITU-T Amendment 1 maps Y.1541 QoS classes to diffserv PHBs, but this is a mistake:

Service is behavior on a domain (PDBs), not through a router (PHBs).

- Per-domain specifications must account for metrics under aggregation and subject to edge rules, unlike the PHB of a single node.

- Suggest map Class 5 to Lower Effort (RFC3662), Class 4 to Best Effort (RFC3086), map Classes 2-4 to Assured Forwarding PDB (expired internet draft) or (in progress) Controlled Effort PDB. Map Class 0 or Dedicated Bandwidth (DBW) to Virtual Wire PDB (CCR paper). (seeking PDB collaborators)

Needed: Composable Metrics for Interdomain Differentiated Services

- Y.1541 specifies desired/required bounds on performance for each QoS class, but need a way to **construct** a service that **delivers** these bounds across each domain (the intent of a PDB spec) and, further, a way to compose those individual domain's metrics as packets flow across domains.

- Specification of metrics and their properties with **aggregation** may emerge with deployments like the EuQoS project.

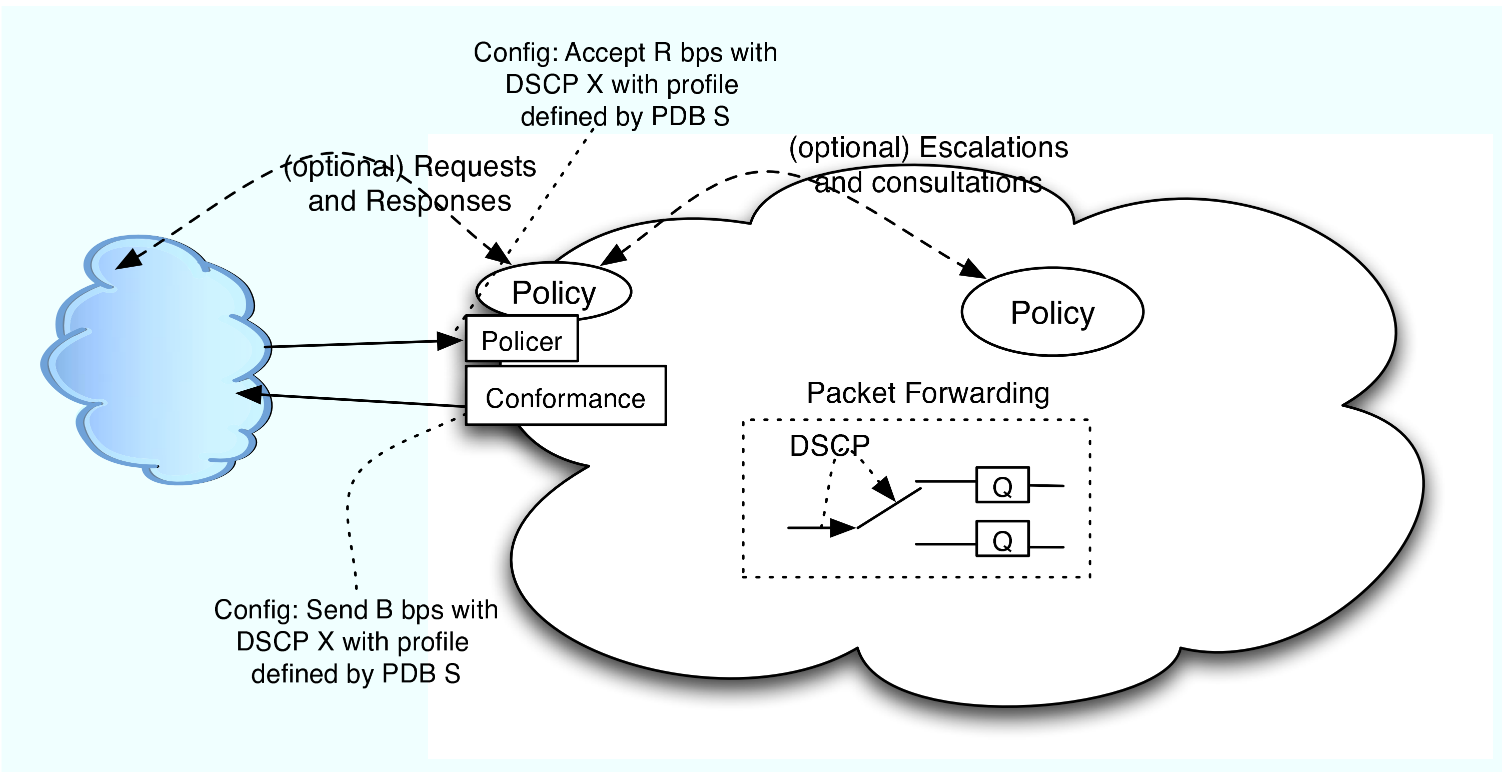
- We pointed out Virtual Wire PDBs can statistically bound delays much more tightly than the worst case and have been watching for tight bounds that apply to a practical architecture. Seems to be some work on that emerging. In the future, it will be necessary to work out the composite statistical bounds of multiple domains.

- Experience combined with new theoretical results may lead to tight bounds that are virtually "perfect" in deployment. (i.e. statistical bound never exceeded?)

Getting to E2E DifS

As individual network domains may be implemented quite differently, E2E Solutions should mix control approaches smoothly. Requests and responses are optional (i.e. signaling not required) and should be able to be done in some small number of different ways.

Also want to decouple the requestor from the service (as for SS7).



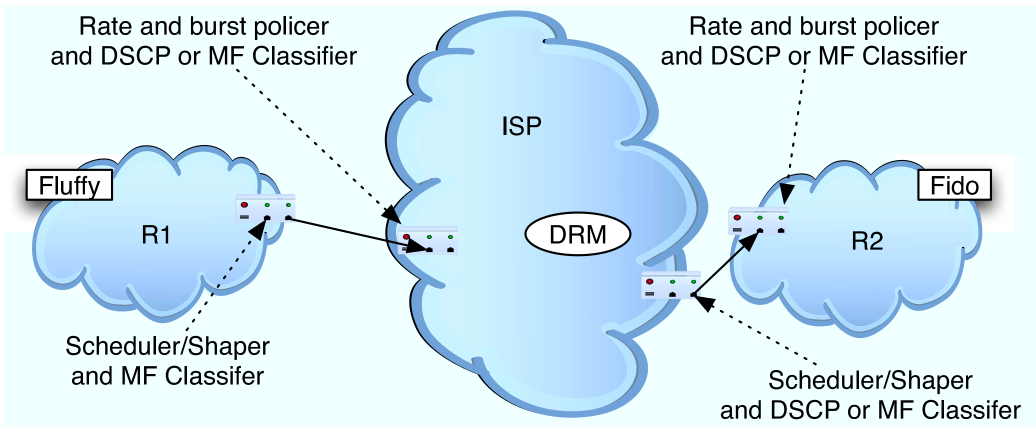
Some Words about my Examples

- Every application of QoS will have its own idiosyncracies. These examples are more motivational than recipe.
- Intent is to make solutions no more complex than necessary and show some situations I think should be handled in E2E solutions
- In these examples, resources (bandwidth) are **provisioned** for a specific desired QoS, **allocation** parcels out the provisioned resources according to policy, and **resource control** ensures allocation is not violated.
- Allocation and resource control may be static, dynamic or some combination
- Use "domain-based resource manager" (DRM) as a generic designator.
- A domain's resource manager may be centralized, distributed, hierarchical.

Provisioned Service Levels

A link between two networks is configured for allocations of jointly accepted PDBs. Allocations each domain agrees to accept must be supportable to "anywhere" or to a list of destinations that is specified to the sending domain at the particular agreed service level. Part of SLA.

Example: to support a desired number of telephony users use Erlang formulas to get bandwidth. This bandwidth of the PDB gets allocated in an SLA annotated with other information (e.g. outages).

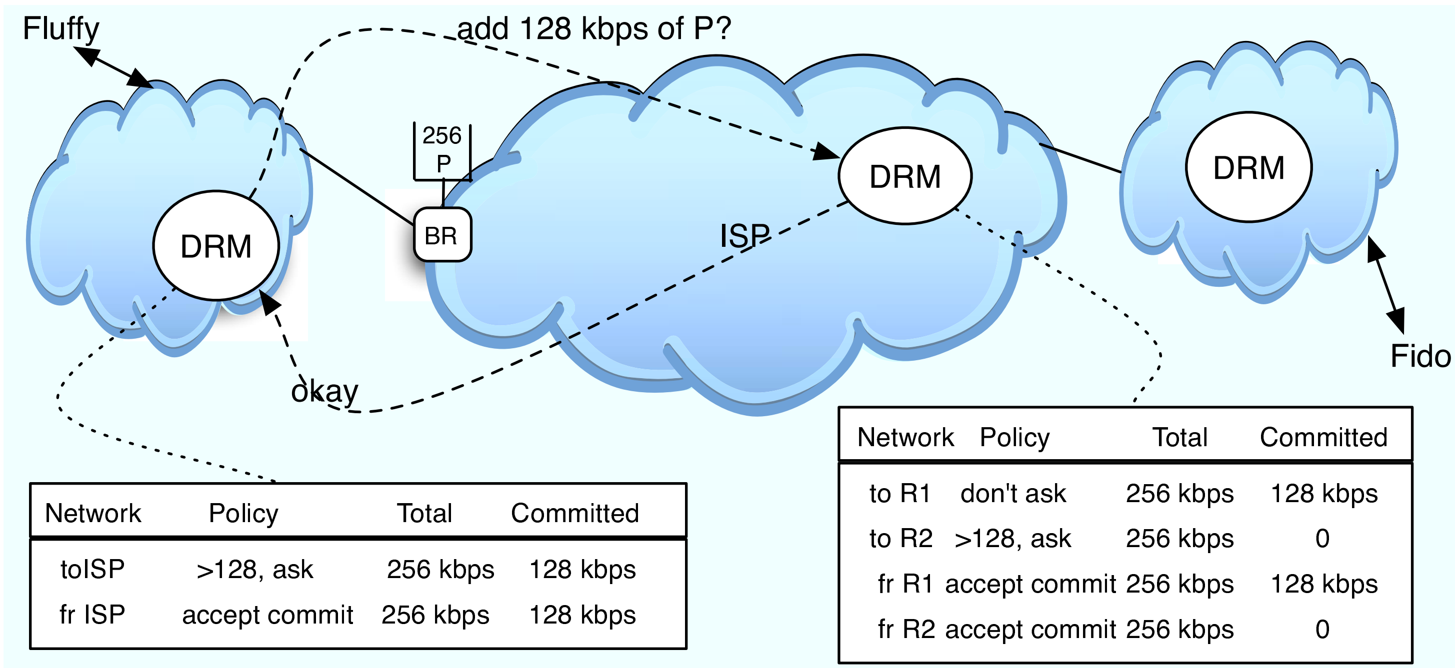


Agree upon DSCP and policer settings for downstream network. Police to this.

Downstream domains reserve a portion of their PDB for use by the upstream domain's traffic. To hide which DSCP using internally, necessary to remark the packets on ingress.

Advance Provisioned but Signal before Use

Useful where resources are overallocated or network changes might occur. In latter case, “don’t ask” amount could be minimum expected to be available. Communication between DRMs in attached networks.



Network	Policy	Total	Committed
toISP	>128, ask	256 kbps	128 kbps
fr ISP	accept commit	256 kbps	128 kbps

Network	Policy	Total	Committed
to R1	don't ask	256 kbps	128 kbps
to R2	>128, ask	256 kbps	0
fr R1	accept commit	256 kbps	128 kbps
fr R2	accept commit	256 kbps	0

Ex: R1’s SLA with ISP has a static allocation of **P** traffic with rules on when requests must be made. R1’s Border Router can handle up to 256 kbps of **P**, but only 128 kbps is committed. Fluffy’s request causes R1 to ask ISP for permission to use the remaining 128 kbps. R1 aims a request at DRM.ISP for 128 kbps of type **P** traffic from 10am till noon. The destination (R2) *may* be given. R1 waits for DRM.ISP to return a reply.

(cont'd): Advance Provisioned but Signal before Use

- DRM.ISP consults its policies with regards to the requestor, R1, and the two session end-points, R1 and R2. R2 has a policy not to be asked about commitments below 128 kbps, so DRM.ISP increases the committed amount to 128 kbps, increases its committed amount from R1 to 256 kbps and returns an okay to R1.

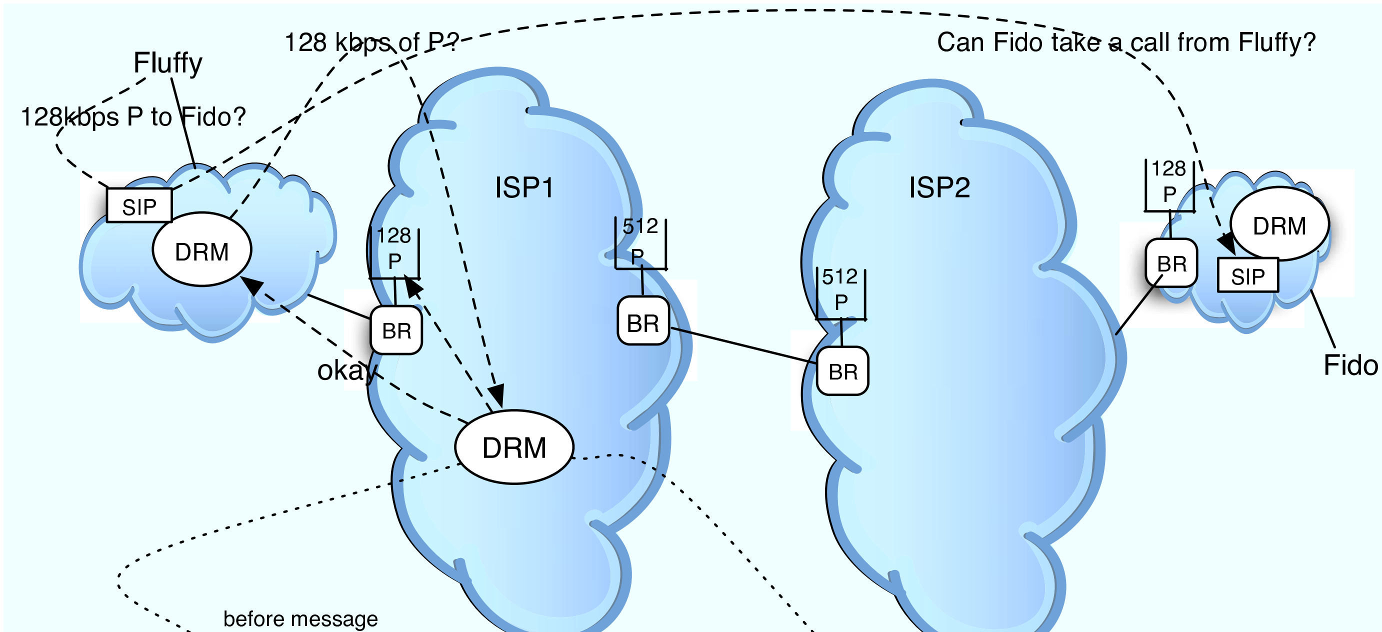
- The policy "accept commit" means that the network is configured to accept whatever the committed amount is. Here the Border Router policer for **P** traffic from R1 is preconfigured at its 256kbps maximum which means DRM.ISP does not need to change configuration. Alternatively, set the policer for only the current committed amount, with a floor of the "don't ask" amount, then DRM.ISP must reconfigure the BR policer with each signaling transaction.

- Sessions could be configured in both directions at once or they might be done with independent messages. "Ask" commitments should have a limited lifetime and/or time out if not refreshed.

- This covers the resource set up for a session that requires special service. Call control messages, if any, are carried by ISP transparently, the call information being instantiated at each end network, R1 and R2.

Achieving E2E Service

Requests pass from each domain to the next until either final *okay* is received or a *no. (no* might return information about which domain or direction said *no*.)



Network	Policy	Total	Committed
to R1	don't ask	128 kbps	0 kbps
to ISP2	send total	512 kbps	256 kbps
fr R1	by request	128 kbps	0 kbps
fr ISP2	accept total	512 kbps	256 kbps

Network	Policy	Total	Committed
to R1	don't ask	128 kbps	128 kbps
to ISP2	send total	512 kbps	384 kbps
fr R1	by request	128 kbps	128 kbps
fr ISP2	accept total	512 kbps	256 kbps

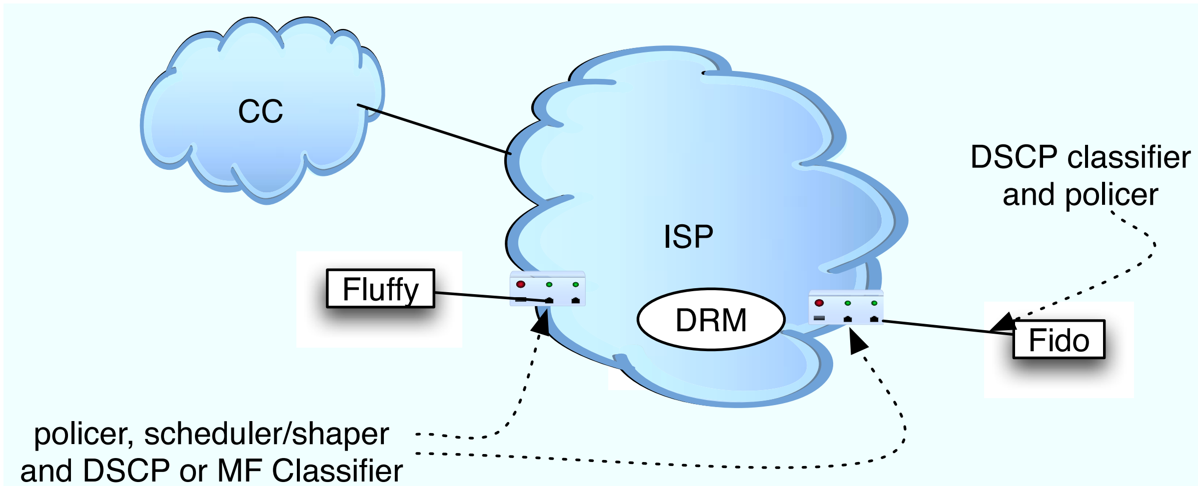
Ex: Fluffy in R1 calls Fido in R2. No ISP1-ISP2 signaling just an agreement to mutually accept 512kbps of P-marked traffic conforming to a specific profile (e.g., the 512 kbps is burst-limited by agreement one packet per millisecond). R2 polices its incoming traffic from ISP2 to 128kbps by agreement. R1 and R2 are using SIP.

After Fluffy initiates the call

- SIP.R1 gets the session invite and sends a request for the 128kbps necessary to DRM.R1.
- DRM.R1 determines that Fluffy is authorized and that the 128 kbps can be carried to R1's border. DRM.R1 asks DRM.ISP1 for 128kbps of type P to the destination address for Fido.
- DRM.ISP1 determines that the destination address is one that is routed through ISP2, notes that there are sufficient uncommitted resources both on the ISP1-ISP2 link and on the path between R1's ingress and the egress to ISP2, so returns an okay to DRM.R1 as well as configuring the Border Router where R1 is attached to police for 128kbps of P from R1 (possibly also including destination). DRM.ISP1 updates its entries.
- DRM.R1 lets SIP.R1 know that the quality level for the call has been set up (as far as it can) and may configure its own Border and Edge Routers at this time. (Steps 1-3 may include a bidirectional reservation of 128 kbps or that direction could be handled during the response from Fido in R2.)
- SIP.R1 sends a message to SIP.R2 asking if Fido can/will accept a call from Fluffy. This is sent as an ordinary data message.
- SIP.R2 checks with DRM.R2 for allocation of P type traffic for this session. DRM.R2 may reject the session for either policy or resource reasons. DRM.R2 has only a static agreement with ISP2, so no further signaling is required to check QoS availability at R2. If okay, DRM.R2 puts a hold on the resource while it signals Fido. If Fido accepts the call, the resource is committed and an okay is returned to SIP.R1.
- SIP.R1 messages DRM.R1 to commit all resources and the call starts (in one direction) or set up begins in the other direction with DRM.R1 signalling for incoming allocation now.

Third Party Set Up

Third party control of sessions can be useful under such circumstances as when attached devices have few capabilities or when central control of a multicast session is required.



CC sends a message to DRM.ISP requesting 128kbps VW from fido to fluffy starting now (or some future time) for 1 hour.

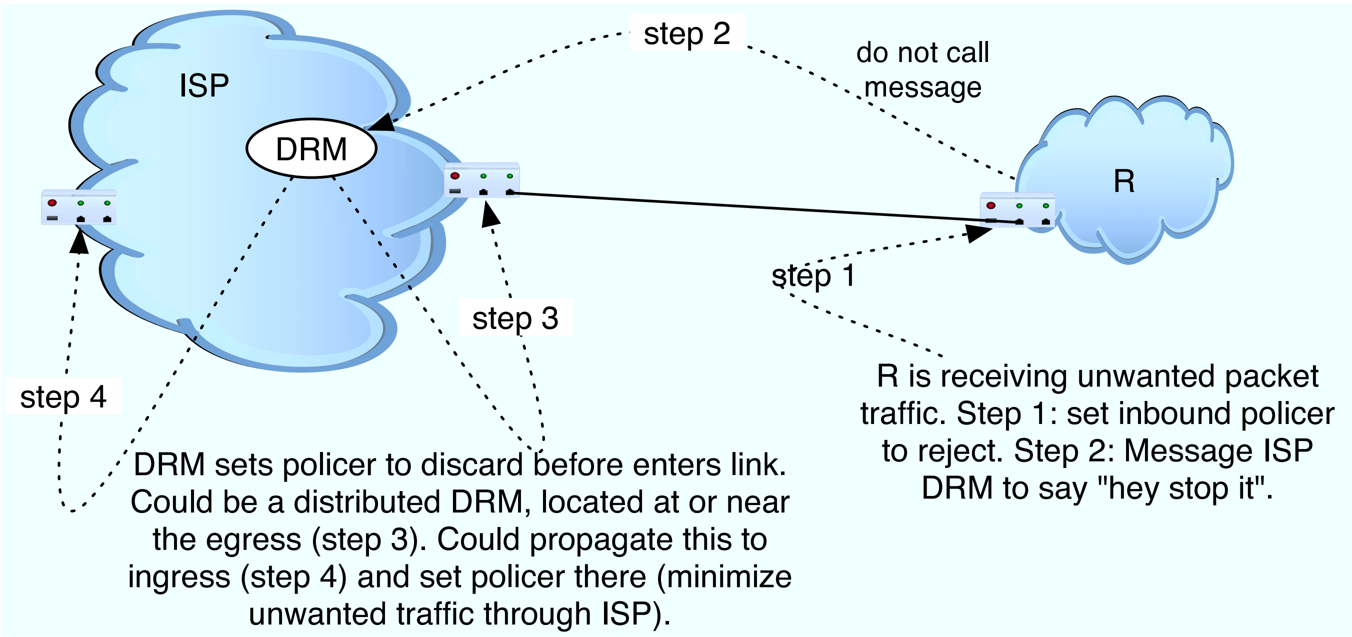
DRM.ISP checks allocations, sets up policers at edges to admit the traffic, ensures links to participants are capable of handling the traffic. *The **service** continues whether fido and fluffy are both available or not*.

The allocation might be more sophisticated, with sessions to CC included.

Fido (or fluffy) can be preconfigured to only accept packets from fluffy or CC.

To Stop Receiving Unwanted Traffic

Putting control of traffic at ends - for multicast, when there are no subscribers, branches are pruned, want to give similar control to ends through messaging if possible



What is the future path?

- Many bodies are working on architectures and specifications for the elements of domain-based DiffServ resource control.
- Some vendors are supplying elements of a domain-based resource manager.
- Interdomain issues are just beginning to be addressed, EuQoS project is notably active here.
- These approaches are promising, though would like to make sure they remain flexible for a heterogeneous environment, creating a framework that embraces a range of network types.

Where should the future be shaped and specified?

- Bodies active in specifying architectures and specifications for control and managment of DiffServ QoS include ITU-T, IPsphere Forum, MS Forum, Packetcable, DSL Forum. There is some coordination.
- Will the specifications converge? Which of the specifications should be ascendant?
- Currently, IETF is notably not a party in the provision of E2E DifS QoS. IETF does protocols (NSIS is notable here) and "boxes" but not architecture. IETF has also stayed out of operator agreement area. Is there a useful role for the IETF in unifying this area or are there already more than enough organizations making standards in this area? (Note also that the drafts related to control of QoS at the IETF differ from the approaches being taken everywhere else: path-oriented, hop-by-hop, a dazzling array of "classes".)
- Discussion is welcome, both here and at dcpel@ietf.org