POLICY-BASED SYNCHRONIZATION OF PER-CLASS RESOURCES BETWEEN ROUTERS IN A DATA NETWORK

[01] The present invention relates to communication networks and, in particular, providing an enhanced quality of service (QoS) to selected traffic flows within a network.

[02] For network service providers, a key consideration in network design and management is the appropriate allocation of access capacity and network resources between traffic originating from network service customers and traffic originating from outside the service provider's network (e.g., from the Internet). This consideration is particularly significant with respect to the traffic of network customers whose subscription includes a Service Level Agreement (SLA) requiring the network service provider to provide a minimum communication bandwidth or to guarantee a particular Quality of Service (QoS) for certain flows. Such service offerings require the network service provider to implement a network architecture and protocol that achieve a specified QoS and that enforce admission control to ensure sufficient access capacity and network resources are available for customers.

[03] In Internet Protocol (IP) networks, a straightforward approach to achieving QoS and implementing admission control comparable to that of connection-oriented network services, such as voice or Asynchronous Transfer Mode (ATM), is to emulate the same hop-by-hop switching paradigm of signaling resource reservations for the flow of IP packets requiring QoS. In fact, the IP signaling standard developed by the Internet Engineering Task Force (IETF) for Integrated Services (Intserv or IS) adopts precisely this approach. As described in IETF RFC 1633 [R. Branden et al., "Integrated Services in the Internet Architecture: an Overview" June 1994], Intserv is a per-flow IP QoS architecture that enables applications to choose among multiple, controlled levels of delivery service for their data packets. To support this capability, Intserv permits an application at a transmitter of a packet flow to use the well-known Resource ReSerVation Protocol (RSVP) defined by IETF RFC 2205 [R.

Branden et al., "Resource ReSerVAtion Protocol (RSVP) – Version 1 Functional Specification" Sept. 1997] to initiate a flow that receives enhanced QoS from network elements along the path to a receiver of the packet flow.

[04] RSVP is a QoS signaling protocol on the control plane of network devices that is utilized to request resources for a simplex flows (i.e., RSVP requests resources for a unidirectional flow). RSVP does not have routing functions, but is instead designed to operate with unicast and multicast routing protocols to ensure QoS for those packets that are forwarded in accordance with routing (i.e., RSVP consults the forwarding table (as populated by routing) in order to decide the downstream interface on which policy and admission control for QoS are applied).

[05] **Figure 1** is a block diagram of an Intserv nodal processing model that utilizes RSVP to achieve QoS in accordance with RFC 2205. As illustrated, a transmitting host **100** executes an application **104**, which transmits data (e.g., video distribution or voice-over-IP (VoIP)) that requires a higher QoS than the "best effort" QoS generally accorded Internet traffic. Between transmitting host **100** and a receiving host **118** are coupled one or more additional nodes, such as router **102**, which implements a routing process **116**.

[06] In the control plane, each network node includes an RSVP process **106** that supports inter-node communication of RSVP messages, a policy control block **108** that determines if a user has administrative permission to make a resource reservation for an enhanced QoS flow, and an admission control block **110** that determines whether or not the node has sufficient outgoing bandwidth to supply the requested QoS. In the data plane, each node further includes a packet classifier **112**, which identifies packets of a flow and determines the QoS class for each packet, and a packet scheduler **114**, which actually achieves the QoS required for each flow in accordance with the packet classification performed by packet classifier **112**.

[07] To initiate an RSVP session, application **104** transmits a PATH message, which is sequentially passed to the RSVP process **106** at each node between transmitting host **100** and receiving host **118**. Although transmitting host **100** initiates the RSVP session, receiving host **118** is responsible for requesting a specified QoS for the session by sending a RESV message containing a QoS request to each network node along the reverse path between receiving host **118** and transmitting host **100**. In response to the receipt of the RESV message, each RSVP process **106** passes the reservation request to its local policy control module **108** and admission control block **110**. As noted above, policy control block **108** determines whether the user has administrative permission to make the reservation, and admission control block **110** determines whether the node has sufficient available resources (i.e., downstream link bandwidth) to supply the requested QoS. If both checks succeed at all nodes between transmitting host **100** and receiving host **118**, each RSVP process **106** sets parameters in the local packet classifier **112** and packet scheduler **114** to obtain the desired QoS, and RSVP process **106** at transmitting host **100** notifies application **104** that the requested QoS has been granted. If, on the other hand, either check fails at any node in the path, RSVP process **106** at transmitting host **100** returns an error notification to the application **104**.

[08] Although conceptually very simple, Intserv QoS provisioning has limited scalability because of the computationally intensive RSVP processing that is required at each network node. In particular, RSVP requires per-flow RSVP signaling, per-flow classification, per-flow policing/shaping, per-flow resource management, and the periodic refreshing of the soft state information per flow. Consequently, the processing required by Intserv RSVP signaling is comparable to that of telephone or ATM signaling and requires a high performance (i.e., expensive) processor component within each IP router to handle the extensive processing required by such signaling.

[09] In recognition of the scalability and other problems associated with implementing IP QoS utilizing conventional Intserv RSVP signaling, the IETF promulgated the Differentiated

Services (Diffserv or DS) protocol defined in RFC 2475. Diffserv is an IP QoS architecture that achieves scalability by conveying an aggregate traffic classification within a DS field (e.g., the IPv4 Type of Service (TOS) byte or IPv6 traffic class byte) of each IP-layer packet header. The first six bits of the DS field encode a Diffserv Code Point (DSCP) that requests a specific class of service or Per Hop Behavior (PHB) for the packet at each node along its path within a Diffserv domain.

[10] In a Diffserv domain, network resources are allocated to packet flows in accordance with service provisioning policies, which govern DSCP marking and traffic conditioning upon entry to the Diffserv domain and traffic forwarding within the Diffserv domain. The marking and conditioning operations need be implemented only at Diffserv network boundaries. Thus, rather than requiring end-to-end signaling between the transmitter and receiver to establish a flow having a specified QoS, Diffserv enables an ingress boundary router to provide the QoS to aggregated flows simply by examining and/or marking each IP packet's header.

[11] As described in IETF RFC 2998 [Y. Bernet et al., "A Framework for Integrated Services Operation over Diffserv Networks" Nov. 2000] and as illustrated in **Figure 2**, Integrated Services can be implemented over a Differentiated Services domain. In the network model illustrated in **Figure 2**, edge routers (ERs) **120, 128** connect Integrated Services-aware customer LANs (not shown) to boundary routers (BRs) **122, 126** of a Diffserv network **124**. To reflect a unidirectional traffic flow from LAN-TX (transmitting) to LAN-RX (receiving), edge router **120** and boundary router **122** are labeled ER-TX and BR-TX, respectively, at the transmitter or ingress side, and edge router **128** and boundary router **126** are labeled ER-RX and BR-RX, respectively, at the receiver or egress side.

[12] Viewed logically, each of routers **120, 122, 126** and **128** has control and data planes, which are respectively depicted in the upper and lower halves of each router. The data plane includes all of the conventional hardware components in the forwarding path of the router

(e.g., interface cards and switching fabric), and the control plane includes control hardware (e.g., a control processor) and control software (e.g., routing, signaling and protocol stacks) that support and direct the operation of the data plane.

[13] In the data plane, packets are marked by data plane **120b** of ER-TX **120** with the appropriate DSCP (e.g., based upon the Intserv 5-tuple of source address, destination address, protocol id, source port and destination port) and forwarded to Diffserv network **124**. The packets are then solely Diffserv forwarded across Diffserv network **124** to data plane **128b** of ER-RX **128**. In the control plane, each of edge routers **120, 128** and boundary routers **122, 126** has a control plane that performs Intserv (IS) processing by reference to policies implemented in policy decision points (PDPs) **130a, 130b**. In ER-TX **120**, control plane **120a** performs Intserv per-flow classification and per-flow policing. In boundary routers **122** and **126**, the Intserv interfaces facing edge routers **120, 128** manage RSVP signaling, perform Intserv policy and admission control functions, and maintain per-flow state with path state blocks and reservation state blocks. Control plane **128a** of ER-RX **128** performs Intserv per-flow shaping before outgoing packets are forwarded to LAN-RX.

[14] As discussed above, before sending a traffic flow, a transmitting host in LAN-TX initiates a RSVP PATH message. When the receiving host in LAN-RX receives the PATH message, the receiving host returns a RESV message along the reverse data path to request reservation of resources to provide the desired QoS. After receiving the RESV message, each intermediate router having an Intserv control plane performs admission control for only its downstream link. Thus, ER-RX **128** performs admission control for LAN-RX, BR-RX **126** performs admission control for the link between itself and ER-RX **128**, BR-TX **122** performs admission control for the path across Diffserv network **124** to BR-RX **126**, and ER-TX **120** performs admission control for the link between itself and BR-TX **122**. The RSVP admission control process verifies resource availability on each link and accordingly adjusts the remaining resource count for the link.

[15] Although Intserv per-flow admission control is performed on the control plane, the actual delivery of QoS for a traffic flow is accomplished on the data plane. ER-TX **120** performs Intserv operations (i.e., per-flow classification, per-flow policing, and per-flow DSCP marking) on data packets received at its Intserv input interface (IS IN). At the Diffserv output interface (DS OUT) of ER-TX **120**, data packets are identified and class-based queued based on only their DSCP values. BR-TX **122** then performs per-class policing for each customer at its input interface (DS IN) and class-based queuing at its output interface (DS OUT). At BR-RX **126**, no operation is performed at the input interface (DS IN), and class-based queuing and optionally per-class shaping are performed for each customer port at the output interface. ER-RX **128** forwards packets received at its input interface (DS IN) and may perform per-flow scheduling or shaping at its Intserv output interface (IS OUT).

[16] Although the Diffserv standard improves upon Intserv's scalability by replacing Intserv's processing-intensive signaling in the Diffserv domain with a simple class-based processing, implementation of the Diffserv protocol introduces a different problem. In particular, because Diffserv allows host marking of the service class, a Diffserv network customer link (e.g., the outgoing link of BR-RX **126**) can experience a Denial of Service (DoS) attack if a number of hosts send packets to that link with the DS field set to a high priority, as discussed in detail in Application Serial No. 10/023,331 cross-referenced above.

[17] Furthermore, despite some improvements in scalability within the Diffserv domain, Intserv admission control utilizing RSVP still requires per-flow state installation, per-flow state refreshment, per-flow traffic management and resource reservation on each edge and boundary router of a service provider's networks. Because boundary routers process thousands of traffic flows as network aggregation points, many vendors' boundary routers cannot install flow state for such a large number of flows. As a result, RSVP per-flow admission control has been rarely implemented and supported by router vendors. Thus,

conventional Intserv per-flow admission control using RSVP remains undesirable due to its lack of scalability.

[18] The present invention addresses the foregoing and additional shortcomings in the prior art by introducing an improved method, apparatus and system for performing admission control.

[19] In accordance with one embodiment of the invention, the policy server includes processing resources, a communication interface in communication with the processing resources, and data storage that stores a configuration manager executable by the processing resources. The configuration manager configures data handling queues of the upstream router to provide a selected bandwidth to one or more of a plurality of service classes of data flows. In addition, the configuration manager transmits to the downstream router one or more virtual pool capacities, each corresponding to a bandwidth at the upstream router for one or more associated service classes among the plurality of service classes. In one embodiment, the configuration manager configures the data handling queues on the upstream router only in response to acknowledgment that one or more virtual pool capacities transmitted to the downstream router were successfully installed.

[20] Additional objects, features, and advantages of the present invention will become apparent from the following detailed written description.

[21] The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself however, as well as a preferred mode of use, further objects and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

[22] **Figure 1** depicts a conventional Integrated Services (Intserv) nodal processing model in which per-flow QoS is achieved utilizing RSVP signaling in accordance with RFC 2205;

[23] **Figure 2** illustrates a conventional network model in which Integrated Services (Intserv) are implemented over a Differentiated Services (Diffserv) domain in accordance with RFC 2998;

[24] **Figure 3** is a high-level network model that, in accordance with a preferred embodiment of the present invention, implements Intserv over a Diffserv domain while eliminating Intserv processing in the boundary routers of the Diffserv domain;

[25] **Figure 4** illustrates one method by which the receiving edge router of a traffic flow can be identified within the network model of **Figure 3**;

[26] **Figure 5** is a more detailed block diagram of a transmitting edge router in accordance with a preferred embodiment of the present invention;

[27] **Figure 6** is a more detailed block diagram of a receiving boundary router and receiving edge router in accordance with a preferred embodiment of the present invention;

[28] **Figure 7** is a block diagram of an exemplary server computer system that may be utilized to implement a Policy Decision Point (PDP) in accordance with a preferred embodiment of the present invention;

[29] **Figure 8A** depicts a preferred method of installing policies on a receiving boundary router and receiving edge router during service initialization;

[30] **Figure 8B** illustrates a preferred method of installing policies on a receiving boundary router and receiving edge router in response to a service update; and

[31] **Figure 8C** depicts a preferred method of policy synchronization following a direct service update to a receiving boundary router.

I. Network Model Overview

[32] With reference again to the figures and, in particular, with reference to **Figure 3**, there is depicted a high level block diagram of an scalable network model that provides enhanced QoS to selected traffic by implementing edge-based Intserv over a Diffserv domain in accordance with the present invention. Specifically, as described in detail below, the illustrated network model improves network scalability by eliminating Intserv per-flow admission control from network devices in the Diffserv domain using a mechanism that maps per-flow bandwidth requirements to class-based resource pools for resource reservation and management. For ease of understanding, **Figure 3** employs the same receiver/transmitter and data plane/control plane notation utilized in **Figure 2**.

[33] In **Figure 3**, Integrated Services-aware LAN-TX and LAN–RX, which may each contain one or more hosts, are connected to customer premises equipment (CPE) edge routers (ERs) **150, 158**. Edge routers **150, 158** are in turn coupled by access networks (e.g., L2 access networks) to boundary routers (BRs) **152, 156** of Diffserv network **124**. The network service provider configures routers **150, 152, 156** and **158** and installs admission control and other policies on **150, 152, 156** and **158** utilizing one or more PDPs **160**.

[34] Utilizing this configuration, the network model of **Figure 3** supports unidirectional traffic flow from transmitting hosts in LAN-TX to receiving hosts in LAN-RX. As is typical, such communication is preferably conducted utilizing a layered protocol architecture in which each protocol layer is independent of the higher layer and lower layer protocols. In one preferred embodiment, communication employs the well-known Internet Protocol (IP) at the network level, which corresponds to Layer 3 of the ISO/OSI (International Organization for Standardization/Open Systems Interconnect) reference model. Above the network layer,

communication may employ TCP (Transmission Control Protocol) or UDP (User Datagram Protocol) in the transfer layer corresponding to Layer 4 of the OSI/ISO reference model.

[35] Above the transfer layer, communication may employ any of a number of different protocols, as determined in part by the required QoS and other requirements of a flow. For example, the International Telecommunication Union (ITU) H.323 protocol and the IETF Session Initiation Protocol (SIP) are commonly utilized to provide signaling for voice, video, multimedia and other types of enhanced QoS sessions over an IP network. As an end-to-end protocol, SIP advantageously permits the end nodes with the capability to control call processing utilizing various call features (e.g., Find-me/Follow-me).

[36] In contrast to the prior art network model illustrated in **Figure 2**, which requires an Intserv control plane that performs Intserv processing in at least each edge and Diffserv boundary router, the network model illustrated in **Figure 2** employs Intserv processing only at the extreme edge of the network, that is, on network-managed CPE edge routers **150, 158**. Thus, for the illustrated unidirectional packet flow, edge routers **150, 158** perform Intserv admission control utilizing RSVP signaling to provide enhanced QoS for a flow sent from LAN-TX to LAN-RX. Because edge routers **150, 158** perform Intserv admission control for Diffserv network **154** (and assuming that Diffserv network **154** has been well traffic engineered), there is no need to implement any additional admission control for Diffserv network **154**. Consequently, in accordance with the present invention, none of the routers in Diffserv network **154**, including boundary routers **152, 156** and unillustrated core routers, is required to have an Intserv control plane, as indicated at reference numerals **152a** and **156a**. Consequently, boundary routers **152** and **156** can be significantly simplified to promote enhanced scalability of the service provider network.

[37] To achieve this advantageous simplification in boundary routers **152, 156**, the network model of **Figure 3** implements modifications to the conventional Intserv RSVP signaling model, which, as described above, always performs symmetric processing at each

node to perform admission control for the downstream link. In the network model illustrated in **Figure 3**, the RSVP RESV message returned by the receiving host is processed only by the Intserv control planes **150a, 158a** of edge routers **150, 158**, which verify the availability of the requested resources and adjust resource counts accordingly. In particular, Intserv control plane **150a** of ER-TX **150** performs downstream admission control for the link between itself and BR-TX **152**. Intserv control plane **158a** of ER-RX **158**, however, performs admission control not only for its downstream link (i.e., LAN-RX), but also for the upstream link itself and BR-RX **156** because boundary routers **152, 156** are not RSVP-aware.

[38] Although conceptually elegant, this network model shown in **Figure 3** has a number of non-trivial challenges that must be addressed in order to obtain operative network implementations. For example, because conventional Intserv RSVP signaling is symmetrical at each node, no conventional mechanism is provided to inform ER-RX **156** that it is the "receiving" edge router and must therefore perform admission control for its upstream link. In addition, conventional Intserv RSVP signaling does not provide ER-RX **156** with any information regarding the resource capacity and resource availability of the upstream link for which admission control must be performed. Moreover, RFC 2998 (and the art generally) does not provide any guidance regarding how to implement Diffserv/Intserv interworking at ER-TX **150** and, in particular, does not disclose how to map Intserv classes to Diffserv classes. Preferred solutions to these and other issues concerning an implementation of the network model shown in **Figure 3** are described in detail below.

II. Receiving Edge Router Identification

[39] Referring now to **Figure 4**, there is depicted one preferred method by which an edge router, such as ER-RX **158**, can determine that it is the receiving edge router. In the depicted operating scenario, each of the customer LANs, edge routers **150, 158** and boundary routers **152, 156** has a different IP address, and the customer LANs coupled to ER-RX **158** are each assigned an IP address that is a subnet of the IP address assigned to ER-RX **158**.

[40] As noted above, a transmitting host in LAN-TX initiates an enhanced QoS session with a receiving host in LAN-RX by transmitting an RSVP PATH message. Based upon the destination address (DestAddress) specified in the PATH message, which in the illustrated example is a.b.p.d, the PATH message is routed to across Diffserv network **154** to LAN-RX. In response to the PATH message, the receiving host transmits an RSVP RESV message containing a SESSION object that specifies the destination address. Upon receipt of the RESV message, the RSVP process in Intserv control plane **158a** of ER-RX **158** can determine whether ER-RX **158** is the receiving edge router by comparing the destination address with the IP subnet address of each attached customer LANs. If and only if the destination address falls into one of its attached customer subnets, ER-RX **158** "knows" it is the receiving edge router for the traffic flow. For example, when ER-RX **158** receives a RESV message having a SESSION object containing destination address a.b.p.d, ER-RX **158** knows that it is the receiving edge router since the IP address of LAN-RX (i.e., a.b.p.d) is an IP subnet address of a.b.p.0/24. ER-RX **158** therefore performs Intserv admission control for its upstream link for the enhanced QoS flow.

[41] Although this method of identifying the receiving edge router has the advantage of simplicity, it requires that each destination address specify a subnet of the receiving edge router's IP address. In implementations in which this restriction is not desirable, alternative methods of identifying the receiving edge router may be employed. For example, as described below in detail with respect to **Figure 6**, the receiving edge router may alternatively be identified through an Edge Point Identification table configured on edge routers **150, 158** by PDPs **160**. These policy data structures specify one or more ranges of IP addresses for which a router is the receiving edge router.

III. Resource Management

[42] To track resource availability (including the resource availability utilized to perform upstream admission control), each Intserv-aware edge router maintains a separate or shared virtual pool in its control plane for each Intserv class, where each virtual pool represents the resource availability for the associated Intserv class(es) on a link for which the router performs admission control. Whenever an edge router receives an RSVP RESV message, the edge router performs admission control on the link by checking the requested bandwidth against the appropriate virtual pool to determine resource availability in the requested Intserv class. If the virtual pool indicates the requested bandwidth is less than the available bandwidth, the reservation request is approved and the reservable resources of virtual pool are reduced by the amount of reserved bandwidth. If, however, the requested bandwidth exceeds the virtual pool's available bandwidth the QoS request is denied.

[43] Interworking between the Intserv admission control and Diffserv data plane functions is achieved by association of the virtual pools utilized to perform Intserv admission control with the logical queues employed by Diffserv to deliver class-based QoS on the data plane. In particular, each Intserv class is uniquely associated with one and only one Diffserv logical queue. However, like the virtual pools utilized to perform Intserv admission control, a separate logical queue can be implemented for each of one or more Intserv classes, and one or more logical queues may be implemented as shared queues that are associated with multiple Intserv classes.

[44] Table I below summarizes the possible combinations of logical queues and virtual pools that may be implemented within the boundary and edge routers of a service provider network.

Table I

Logical Queue	Virtual pool	
	Separate	Shared
Separate	Case 1	Not Applicable
Shared	Case 3	Case 2

[45] As shown in Table I, three cases are possible: separate virtual pools with separate logical queues, shared virtual pools with shared logical queues, and separate virtual pools with shared logical queues. The case of a virtual pool shared by multiple Intserv classes is not applicable to an implementation having separate logical queues for each Intserv class, since no virtual pool information would be available on an individual class basis. Importantly, boundary and edge routers in the same network may be configured to concurrently implement different cases, as long as marking is correctly performed.

[46] With reference now to **Figures 5** and **6**, there are depicted more detailed block diagrams of edge and boundary routers of the network model of **Figure 3** in which traffic in each Intserv service class is assigned a separate virtual pool in the control plane and separate logical queue in the data plane in accordance with Case 1 of Table I. Referring first to **Figure 5**, a more detailed block diagram of ER-TX **150** is depicted. As noted above, ER-TX **150** has an Intserv control plane **150a**, which manages RSVP signaling and implements Intserv policy and admission control, and a data plane **150b**, which provides the link level delivery of Diffserv class-based QoS. Control plane **150a** includes an RSVP process **180**, an admission control block **182** having associated virtual pools **184**, a policy control block **188**, an IS-DS interworking function (IWF) configuration block **186**, and a Policy Configuration Interface (PCI) **190** through which ER-TX **150** communicates policy information with PDP **160a**. Data plane **150b** has an input port **200**, a forwarding function **208**, and an output port **210** having a number of queues **212** that each corresponds to a Diffserv class.

[47] As described above, RSVP process **180** in control plane **150a** handles RSVP signaling (e.g., PATH and RESV messages) utilized to reserve (and release) resources for enhanced QoS flows. In response to receiving a RESV message requesting resources for an enhanced QoS flow, RSVP process **180** interrogates admission control block **182** and policy control block **188** to verify that the requestor has administrative permission to establish the QoS flow and that the downstream interface has sufficient available resources to support the requested QoS. In addition to determining administrative permission, policy control block **188** can execute additional policies, such as authentication based on certificates or signatures, management of bandwidth distribution among the authorized requestors, and preemption of allocated resources for a pending, higher-priority flow.

[48] In the illustrated embodiment, each supported Intserv class (e.g., Guaranteed Service (GS) and Controlled Load (CL)) has a separate virtual pool **184a, 184b**. Admission control block **182** monitors the availability of resources on the downstream link for each Intserv class using virtual resource pools **184**. Thus, admission control block **182** grants reservation requests when sufficient available bandwidth is available in the virtual pool associated with the requested Intserv class and otherwise denies the reservation request. Admission control block **182** reduces the available resources in a virtual pool by the amount requested by each successful reservation, and increases the reservable resources in a virtual pool by the amount of resources freed upon termination of a flow. Importantly, the number of virtual pools, the bandwidth allocated to each virtual pool **184**, and the mapping between the virtual pools and Diffserv classes are not fixed, but are instead expressed as policies that are installed at ER-TX **150** (and other network elements) by a PDP **160**. Utilizing Common Open Policy Service (COPS) or other protocol, such policies may be pushed onto network elements by PDP **160** or pulled from PDP **160** by a network element, for example, in response to receipt of an RSVP RESV message.

[49] PDP **160a** configures the mapping between Intserv classes and Diffserv classes (and DSCPs) on IS-DS IWF configuration block **186** (e.g., GS to DSCP 100011, CL to DSCP 010011). IS-DS IWF configuration block **186** may also receive configurations from RSVP process **180**. Based upon these configurations, IS-DS IWF configuration block **186** dynamically provisions a packet classifier **202**, policer **204**, and marker **206** on input port **200** for each Intserv flow. (In some implementations, packet classifier **202**, policer **204**, and marker **206** may be implemented as a single integrated module, such as a Field Programmable Gate Array (FPGA) or Application Specific Integrated Circuit (ASIC).)

[50] In accordance with this provisioning, packets within each Intserv flow, whose service class is indicated by an Intserv 5-tuple, are classified and marked by packet classifier **202** and marker **206** with the appropriate DSCP of the aggregate Diffserv class (e.g., with one of the 16 code points (Pool 2 xxxx11) reserved for experimental or local use). In this manner, Intserv flows having enhanced QoS are aggregated into preferential Diffserv classes. Because the embodiment shown in **Figure 5** reflects Case 1 from Table I, a separate logical queue **212** is provided on port **210** for each supported Intserv class (GS and CL) in addition to the logical queues assigned to other Diffserv classes (e.g., the Expedited Forwarding (EF), Assured Forwarding (AF) and default Best Effort (BE) classes). Scheduler **214** then provides the appropriate QoS to the packets within each logical queue **212** by scheduling packet transmission from logical queues **212** in accordance with scheduler weights assigned to each logical queue **212** by PDP **160a**.

[51] Because the illustrated embodiment of ER-TX **150** is managed by the network service provider, ER-TX **150** can be trusted by the network service provider to correctly mark packets with DSCPs so that no "theft" of QoS occurs. In alternative embodiments in which ER-TX is not managed by the network service provider, PDP server **160a** may provide the Diffserv classification policies to BR-TX **152** instead of ER-TX **150**. It should also be noted

that core routers of Diffserv network **154** need not implement separate Diffserv queues for Intserv flows, even if separate queues are implemented on edge and boundary routers.

[52] Referring now to **Figure 6**, there are illustrated more detailed block diagrams of BR-RX **156** and ER-RX **158** in accordance with a preferred implementation of Case 1 of Table I. As noted above, BR-RX **156** and ER-RX **158** have respective control planes **156a, 158a** and data planes **156b, 158b**. Control plane **158a** of ER-RX **158** is an enhanced Intserv control plane including a PCI **190**, an RSVP process **180** having associated admission and policy control blocks **182** and **188**, and an edge point identification table **252** and upstream virtual pools **250** by which admission control block **182** performs upstream admission control. BR-RX **156a**, by contrast, has no Intserv control plane, but instead includes only a PCI **190** through which the components of data plane **156b** are configured by PDP **160b**.

[53] Within control plane **158a** of ER-RX **158**, PDP **160b** installs policies by which local policy control **188** determines which customers having administrative permission to request resource reservations for enhanced QoS flows. In addition, PDP **160b** installs an edge point identification table **252** that specifies one or more ranges of destination IP addresses for which ER-RX **158** is the receiving edge router. Thus, upon receipt of a RESV message requesting an enhanced QoS flow for which the customer is granted administrative permission by policy control **188**, admission control **182** interrogates edge point identification table **252** to determine if ER-RX **158** is the receiving edge router for the requested flow. If not, ER-RX **158** performs only conventional downstream admission control. However, if edge point identification table **252** indicates that ER-RX **158** is the receiving edge router for the requested flow, admission control block **182** performs upstream admission control by reference to the upstream virtual pool capacities allocated by PDP **160b** to each Intserv class within virtual pools **250**. As described generally above, each virtual pool **250a, 250b** is utilized by admission control block **182** to ascertain the availability of sufficient bandwidth for a requested flow of a particular Intserv class on the upstream link

between ER-RX **158** and BR-TX **152**. As indicated at reference numeral **252**, PDP **160b** obtains periodic or solicited feedback regarding virtual pool usage on ER-RX **158** and dynamically coordinates any operator-initiated adjustments to the capacities of the virtual pools with updates to the logical queue(s) and scheduler weight(s) implemented in the data plane to ensure that the Intserv bandwidth actually utilized is less than the operator-specified capacity.

[54] Referring now to the data plane, data plane **158b** of ER-RX **158** may be implemented with conventional classification, forwarding and Intserv queuing, the details of which are omitted to avoid obscuring the present invention. Data plane **156b** of BR-RX **156** includes an input port **220** having a classifier **222**, an output port **240** having a plurality of Diffserv physical queues **242** and a scheduler **244**, and a forwarding function **230** that switches packets from the input port to the appropriate physical queues **242** on output port **240** in accordance with the classification performed by classifier **222**. As indicated, classifier **222** and physical queues **242** are configured by PDP **160b** in a coordinated manner to reflect the configuration of upstream Intserv virtual pools on control plane **158a** of ER-RX **158**. In particular, in the illustrated embodiment, classifier **222** is configured to identify packets belonging to the separate Diffserv classes into which Intserv traffic are aggregated, such the packets in each Diffserv class representing an Intserv traffic type are forwarded to separate physical queues **242** for Intserv GS and CL classes on output port **240**. PDP **160b** also configures the scheduling weight scheduler **244** gives each of queues **242**. In addition, PDP **160** coordinates the sum of the virtual pool capacities on ER-RX **158** with the resource pool capacity dictated by queue capacities and weights in data plane **156b** of BR-RX **156** to ensure that the virtual pool capacity does not exceed the actual resource pool capacity. Thus, in essence, ER-RX performs upstream admission control as a proxy for BR-RX.

[55] Mapping different Intserv classes to separate virtual pools and Diffserv queues as shown in **Figures 5** and **6** permits better traffic management than mapping all Intserv classes

to a single Diffserv queue. By preserving the distinction between Intserv classes over the Diffserv network in this manner, different traffic types (e.g., VoIP, VideoIP and file transfer) can be provided optimal handling, and enterprise resource planning is simplified. However, as noted above, some or all routers in a service provider network may alternatively be implemented in accordance with Cases 2 and 3. To implement Case 2 instead of Case 1, ER-TX **150** and ER-RX **158** are configured with a single shared virtual pool for multiple Intserv classes, and ER-TX **150** and BR-RX **156** are configured with a single shared logical queue for the multiple Intserv classes. Alternatively, to implement Case III, ER-TX **150** and ER-RX **158** are configured with separate virtual pools, and ER-TX **150** and BR-RX **156** are each configured with a single shared queue for multiple Intserv classes.

[56] It should be noted that no flow-specific network configuration of control plane **152a** or data plane **152b** of BR-TX **152** is required in order to provide enhanced QoS to particular flows. This is because the admission control provided by downstream ER-RX **158** ensures that the downstream link of BR-TX **152** has sufficient bandwidth to support each admitted enhanced QoS flow, and the mapping of Intserv flows to particular Diffserv classes ensures that data plane **152b** achieves the requested QoS.

IV. PDP

[57] With reference now to **Figure 7**, there is depicted a high level block diagram of a server computer system that may be employed as a PDP **160** in accordance with a preferred embodiment of the present invention. PDP **160** includes one or more processors **262** coupled by an interconnect **264** to a storage subsystem **268**, which may comprise random access memory (RAM), read only memory (ROM), magnetic disk, optical disk and/or other storage technology. Storage subsystem **268** provides storage for data (e.g., tables **280-290**) and instructions (e.g. configuration manager **292**) processed by processor(s) **262** to configure network elements and to install and determine network policies. Also coupled to interconnect **264** may be one or more input devices (e.g., a keyboard and/or graphical

pointing device) **270** and one or more output devices (e.g., a display) **272**, as well as a communication interface **274** through which computer system **260** may communicate with network devices, such as routers **150**, **152**, **156** and **160**.

[58] To configure and install policies on routers **150**, **156**, **160** in the manner described above, each PDP **160** preferably implements a number of Policy Rule Class (PRC) tables within storage subsystem **268**. In one preferred embodiment, these PRC tables include at least an Admission Control Virtual Pool Table **280**, Intserv Capacity Table **282**, Intserv-to-Diffserv Interworking Function Table **284**, Edge Point Identification Table **286**, Pool Usage Feedback Table **288**, and Boundary Resource Pool Table **290**.

[59] Admission Control Virtual Pool Table **280** determines the capacities of the virtual pools on edge routers **150**, **158** that are utilized to perform admission control for various Intserv classes. In Admission Control Virtual Pool Table **280**, the sum of the capacities assigned to the virtual pools associated with all Intserv classes is set to be less than the data plane queue capacity of the associated boundary router to ensure that the requested QoS of each admitted flow can be achieved in the data plane. The table further specifies whether the admission control will accept reservations and the logical interface name of the boundary router associated an edge router. In an exemplary embodiment, Admission Control Virtual Pool Table **280** may be defined as follows:

AdmCtlVirtualPoolTable

Logical Interface Name

Description: This SNMP string identifies the logical interface associated with the AdmCtlVirtualPool entry.

Object Type: SNMP string

Direction

Description: This attribute indicates the relationship of the traffic stream to the interface as either (1) inbound or (2) outbound. This attribute is used in combination with the BoundaryLogicalInterfaceName to differentiate ER-RX virtual resource pools and ER-TX virtual resource pools. An ER-RX upstream virtual resource pool has an inbound Direction and non-

resource pools and ER-TX virtual resource pools. An ER-RX upstream virtual resource pool has an inbound Direction and non-empty BoundaryLogicalInterfaceName. An ER-TX downstream virtual resource pool has an outbound Direction and a non-empty BoundaryLogicalInterfaceName attribute. An ER-RX downstream virtual resource pool has an outbound Direction and an empty BoundaryLogicalInterfaceName attribute.

IntSrvClass

Description: This bit string indicates the Intserv class or classes that have resources allocated by admission control from this virtual pool.

Object Type: bits

Controlled Load Service (1)
Guaranteed Services (2)
Null Service (3)
Other (4)

VirtualPoolMaxAbsRate

Description: the maximum absolute rate in kilobits that this pool may allocate to Intserv sessions defined by the AdmCtlIntSrvClass. The sum of ER-RX upstream virtual resource pools is not to exceed the ResourcePoolMaxAbsRate for the associated BoundaryInterfaceName.

Object Type: Unsigned 32

BoundaryLogicalInterfaceName

Description: identifies the adjacent boundary router and resource pool that governs the capacity of the local virtual pool defined by this entry. An empty attribute signifies that the VirtualPoolMaxAbsRate is governed by a local ResourcePoolMaxAbsRate defined for the LogicalInterfaceName of this entry. A non-empty attribute indicates that a remote virtual pool capacity defined for this BoundaryLogicalInterfaceName governs the value of the VirtualPoolMaxAbsRate of this entry.

Object Type: SNMP string

AcceptReservations

Description: This value indicates whether Admission Control will attempt to process RSVP RESV requests. A value of 0 indicates that reservations are not to be processed. A value of 1 indicates reservations are to be processed.

Object Type: Unsigned 32

[60] Intserv Capacity Table **282** defines the data plane data rate capacity allocated to Intserv classes in terms of both Diffserv queue weights and shaper parameters. These rate capacities are also associated by the table with one or more edge router virtual pools. This Policy Rule Class, according to one preferred embodiment, is contained in the Differentiated Services Policy Information Base (PIB).

[61] Intserv-to-Diffserv IWF Table **284** defines the attributes used for interworking between the RSVP process in the control plane and Diffserv in the data plane. These attributes are used by classifier **202**, policer **204**, and marker **206** on input port **200** of ER-TX **150** to classify, police and mark Intserv traffic flows so that Diffserv achieves the appropriate QoS for each flow. In addition, the table specifies the specific scheduler instance to be used for flows having particular Intserv classes. An exemplary embodiment of Intserv-to-Diffserv IWF Table **284** is as follows:

Intserv-to-Diffserv Interworking Function Table

IwfPrid
Description: This is the unique identifier of the PktIwfTable entry.
Object Type: Instance ID (unsigned 32)

IwfIntSrvClass
Description: The value of the Intserv Class associated with the attributes of this specific interworking function entry. (It must have a corresponding bit set in AdmCtlIntSrvClass)
Object Type: unsigned 32

IwfDSCP
Description: The value of the DSCP to assign the data steam for the session with the Intserv class type matching the value of PktIwfIntSrvClass.
Object Type: integer value 0 - 63

IwfOutOfProfile
Description: This value indicates the policing behavior when the data stream is out of profile. The profile can be defined by the associated MeterTableEntry. A value of 1 indicates out-of-profile packets are to be dropped. A value of 2 indicates out-of-profile packets are to be remarked with the DSCP defined in IwfRemarkValue.
Object Type: Unsigned 32

IwfRemarkValue
Description: The value of the DSCP to remark an out-of-profile packet. This value is only used if the IwfOutOfProfile is set to 2.
Object Type: Unsigned 32 value 0-63

IwfSchedulerPrid
Description: The value of the instance ID of the specific scheduler to be used by data streams of the sessions with an Intserv class matching the value of attribute IwfIntSrvClass.
Object Type: Unsigned 32

[62] Edge Point Identification Table **286** defines a range or ranges of addresses for which an edge router is a receiving edge router. This information may be configured on PDP **160** initially or may be learned locally. Admission control block **182** on ER-RX **158** performs upstream admission control for reservation requests that specify a destination address within the RSVP SESSION Object that falls within one of these address ranges. The values for a particular edge router may be pushed down by PDP **160** to the local Edge Point Identification

Table **252** utilizing COPS or other policy protocol. According to one embodiment, Edge Point Identification Table **286** may be defined as follows:

End Point Identification Table

ReceiverDomainPrid
Description: unique identifier of an entry of this policy rule class
Object Type: Instance ID, a 32 bit unsigned integer.
ReceiverAddrType
Description: The enumeration value that specifies the address type as defined in RFC 2851 [M. Daniele et al., "Textual Conventions for Internet Network Addresses" June 2000]
Object Type: INET Address Type as defined by RFC 2851
ReceiverAddr
Description: The IP address for the Session Object Destination Address to match
Object Type: INET Address as defined by RFC 2851
ReceiverAddrMask
Description: the length of the mask for matching the INET Address
Object Type: unsigned 32

[63] Pool Usage Feedback Table **288** contains entries that specify the current resources consumed by Intserv flows. This PRC table, which is used by PDP **160** to determine when to complete provisioning an operator-initiated capacity update, may in an exemplary embodiment be defined as follows:

Pool Usage Feedback Table

Usage Feedback Prid
Description: unique identifier of the Virtual Pool Usage Feedback entry.
Object Type: Instance Id. (unsigned 32)
PoolPrid
Description: value of the instance ID of the specific AdmCtlVirtualPool entry that usage is describing.
Object Type : Unsigned 32

ResourceAbsRateInUse
Description: current total value of the Intserv resources in use.

[64] Boundary Resource Pool Table **290** defines the total rate capacity that may be assigned by PDP **160** to the various admission control virtual pools associated with a given egress boundary router (BR-RX). This PRC table may be defined in an exemplary embodiment as follows:

Boundary Resource Pool Table

BoundaryResourcePool TableBoundaryResourcePoolPrid
Description: unique identifier of the Virtual Pool Usage Feedback entry
Object Type: Instance Id. (unsigned 32)
BoundaryLogical Interface Name
Description: identifies the adjacent boundary router and resource pool that governs that capacity of the local virtual pools associated with this entry in the AdmissionCtlVirtualPool Table
Object Type: SNMP string
ResourcePoolMaxAbsRate
Description: maximum absolute rate in kilobits that may be allocated to IntServ sessions defined by the AdmCtlIntSrvClass. The sum of ER-RX upstream virtual pools is not to exceed the ResourcePoolMaxAbsRate for the associated BoundaryInterfaceName.
Object Type: Unsigned 32

V. Network Configuration

[65] With reference now to **Figures 8A-8C**, a number of network diagrams are depicted, which together illustrate preferred techniques by which PDP **160b** configures and installs policies on BR-RX **156** and ER-RX **158**. The illustrated functions may be implemented, for example, through the execution by PDP **160** of configuration manager software **292**. In each figure, it is assumed that communication between PDP **160b** and routers **156**, **158** is conducted utilizing COPS, although it should be understood that other protocols may be

employed.

[66] **Figure 8A** specifically illustrates PDP **160b** synchronizing virtual pool capacities on ER-RX **158** with Diffserv logical queue bandwidths on BR-RX **152** during service initialization. As indicated at reference numeral **300** of **Figure 8A**, a Network Management System (NMS) may initiate the configuration of Intserv capacity for a customer, for example, during service initialization. In response, PDP **160b** pushes the configuration of Intserv virtual pool capacities onto each network-managed edge router (of which only ER-RX **158** is shown) that is downstream of a boundary router of Diffserv network **154**. For example, in the depicted embodiment, PDP **160b** pushes the virtual pool capacity for each Intserv class supported by LP1 at interface 1.m.n.b/30 onto ER-RX **158** with a message allocating 10 megabits to the Intserv GS class and 25 megabits to the Intserv CL class. If the configuration is successfully installed on ER-RX **158**, ER-RX **158** replies with an acknowledgement (ACK) message, as shown at reference numeral **304**. PDP **160b**, as indicated at reference numeral **306**, then pushes the corresponding configuration of Diffserv queue(s) and scheduler weight(s) onto BR-RX **156**. BR-RX **156** also returns an ACK **308** to PDP **160b** if the configuration is successfully installed.

[67] If ER-RX **158** fails to install the virtual pool capacities pushed down by PDP **160b**, ER-RX **158** returns a negative acknowledgement (NACK) to PDP **160b**. PDP **160b** accordingly sends a warning message to a network operator, such as "Fail to configure Integrated Services virtual pool on ER XX!" Similarly, if the queue(s) and scheduler weight(s) cannot be installed on BR-RX **156**, BR-RX **156** returns an NACK to PDP **160b**. In response, PDP **160b** transmits a message to ER-RX **158** to release the configuration of the virtual pools and may also send a warning message to a network operator stating: "Fail to configure Queue and Scheduler on BR XX!"

[68] It should be noted that PDP **160b** may not directly communicate with network elements, such as BR-RX **156** and ER-RX **158**, but may instead communicate through other network elements. For example, messages between PDP **160b** and BR-RX **156** may be communicated through ER-RX **158**.

[69] Attention is now turned to a scenario in which a service update (i.e., an increase or decrease in subscribed Intserv capacity) is performed for an existing network service customer. Increasing or decreasing the BR-RX capacity when the currently reserved bandwidth is below the new subscribed capacity is a straightforward process because the new capacity can accommodate all ongoing customer traffic, meaning no service impact will be observed. However, decreasing the BR-RX capacity when the currently reserved bandwidth is greater than the newly requested capacity requires coordination among PDP **160b**, BR-RX **156**, and ER-RX **158**, as described below with respect to **Figure 8B**.

[70] In **Figure 8B**, the NMS may initiate the reconfiguration of Intserv capacity for an existing network service customer, as depicted at reference numeral **320**. As shown at reference numeral **322**, PDP **160b** installs the new virtual pool capacity value(s) on ER-RX **158**. Admission control block **182** of ER-RX **158** compares each new virtual pool capacity value with the amount of resources currently reserved within each virtual pool. If the new virtual pool capacity value(s) are greater than the amount of resources currently reserved from each virtual pool, admission control block **182** of ER-RX **158** overwrites the virtual pool capacity value(s) with the new value(s) and immediately sends an ACK **324** to PDP **160b**. However, if the new virtual pool capacity value(s) are less than the amount of currently reserved resources, admission control block **182** of ER-RX **158** saves the new capacity value(s) without overwriting the old ones. Admission control block **182** of ER-RX **158** accepts no new reservations from a virtual pool to which an update is to be performed until the amount of reserved resources falls below the new virtual pool capacity. Once the reserved resources fall below the new virtual pool capacity, admission control block **182** of

ER-RX **158** overwrites the old virtual pool capacity value(s) with the new value(s), and acknowledges acceptance of the new virtual pool capacity value(s) by sending an ACK **324** to PDP **160b**.

[71] PDP **160b** defers installation of new scheduler weight(s) on BR-RX **156** until PDP **160b** receives ACK **324** from ER-RX **158**. In response to ACK **324**, PDP **160b** pushes queue configuration(s) and scheduler weight(s) onto BR-RX **156**, as illustrated at reference numeral **326**. After successful installation of the new queue configuration(s) and scheduler weight(s), BR-RX **156** returns an ACK **328** to PDP **160b**.

[72] In an alternative embodiment, PDP **160b** determines when to perform a virtual pool capacity update instead of ER-RX **158**. In this embodiment, PDP **160b** solicits reports of or programs periodic unsolicited reporting by ER-RX **158** of the currently reserved Intserv bandwidth. If the currently reserved bandwidth is greater than the new capacity specified by the NMS, PDP **160b** pushes a policy to ER-RX **158** to stop accepting new reservations until the reserved bandwidth is below the new capacity. To further reduce the amount of messaging, PDP **160b** may push a policy on ER-RX **158** that instructs ER-RX **158** to send a single unsolicited report to PDP **160b** only after the reserved bandwidth is less than the new capacity. In response to a message from ER-RX **158** indicating that the currently reserved Intserv bandwidth is less than the new virtual pool capacity, PDP **160b** pushes the new Intserv virtual pool policy onto ER-RX **158** and pushes the corresponding new scheduler queues and weights to BR-RX **156** in the manner described above.

[73] If PDP **160b** fails to successfully update either ER-RX **158** or BR-RX **156**, PDP **160b** may roll back to the old virtual pool capacities and queue and scheduler weight configuration. Additionally, PDP **160b** may send warning messages to the network operator to describe the reason of the failure (e.g., "Failure to configure the updated Integrated

Services virtual pool capacity on ER XX!" or "Failure to configure the updated scheduler weight on BR XX!").

[74] To prevent a PDP (e.g., PDP server **160b**) from becoming a single point of failure, a backup PDP may be utilized for one or more primary PDPs. In the event that a primary PDP fails, the Intserv service control may be switched to the backup PDP, and each ER-RX controlled by the primary PDP may report its current reservation state to the backup PDP. However, each ER-RX should stop accepting new reservations until the switch to the backup PDP is completed. After the primary PDP is restored, the backup PDP first synchronizes state with the primary PDP and then informs each ER-RX to switch back to the primary PDP. After switching back to the primary PDP, each ER-RX synchronizes its reservation state with the primary PDP.

[75] In the event of a failed ER or BR, IP routing and RSVP refresh messages are used to discover a new route and reroute flows around the failed ER or BR. Upon successful rerouting, PDP **160b** may push a policy to the corresponding BR-RX **156** to release the Diffserv queues allocated to Intserv traffic for the failed ER-RX or push policies to all downstream ER-RXs of a failed BR-RX to release the configured virtual pool(s) for the failed BR-RX.

[76] Referring now to **Figure 8C**, there is illustrated an exemplary scenario in which an NMS or network service provider operator directly alters the configuration of queue(s) and scheduler weight(s) on BR-RX **156**. In response to the update, BR-RX **156** notifies PDP **160b** of the changes. If not contained in the notification, PDP **160b** pulls the configuration update from BR-RX **156**, as indicated at reference numeral **342**, and then, as depicted at reference numeral **344**, pushes the new configuration of virtual pool capacities onto all affected ER-RX(s) (of which only ER-RX **158** is shown).

VI. Conclusion

[77] As has been described, the present invention provides a scalable IP network model that provides end-to-end QoS for selected flows by implementing edge-based Intserv over a Diffserv domain. The network model supports a number of functions, including per-flow admission control utilizing Intserv RSVP processing only at the CPE edge routers, receiving edge router identification, upstream admission control at the receiving edge router, pool-based resource management, and synchronization of bandwidth usage information between the receiving boundary router and receiving edge router by policy management. Despite introducing additional functionality, the network model of the present invention is consistent with existing Intserv, COPS and Diffserv models, and the Diffserv policy provisioning model using policy and management information bases. The network model of the present invention advantageously enhances scalability while maintaining a standardized architecture and can therefore be readily adopted for implementation.

[78] While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example only, and not limitation. Thus, the breadth and scope of the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents. For example, although the present invention has been primarily discussed with respect to implementations employing Resource Reservation Protocol (RSVP) and Internet Protocol (IP), it should be appreciated the present invention has applicability to other communication protocols, including Session Initiation Protocol (SIP) and ITU H.323, which may be used to perform admission control by the selective admission or denial of an enhanced QoS flow based upon policy and available resources. Moreover, although the present invention has been described with respect to various hardware elements that perform various functions in order to achieve end-to-end QoS for selected network flows, it should be understood that such functions can be realized through the execution of program code embodied in a computer-readable medium. The term

"computer-readable medium" as used herein refers to any medium that participates in providing instructions to a data processing system for execution. Such a medium may take many forms, including but not limited to non-volatile media, volatile media, and transmission media.

CLAIMS

What is claimed is:

1. A policy server for use in a data network including an upstream router and a downstream router, said upstream router including one or more data handling queues, said policy server comprising:

processing resources;

a communication interface in communication with said processing resources; and

data storage that stores a configuration manager executable by said processing resources, wherein said configuration manager configures the one or more data handling queues of the upstream router to provide a selected bandwidth to one or more of a plurality of service classes of data flows, and wherein said configuration manager transmits to the downstream router one or more virtual pool capacities each corresponding to a bandwidth at the upstream router for one or more associated service classes among said plurality of service classes.

2. The policy server of Claim 1, wherein said configuration manager transmits said virtual pool capacities to said downstream server prior to configuration of said data handling queues on said upstream server.

3. The policy server of Claim 2, wherein said configuration manager configures said data handling queues on said upstream server only in response to acknowledgment that the one or more virtual pool capacities transmitted to said downstream server were successfully installed.

4. The policy server of Claim 1, wherein said configuration manager configures the one or more data handling queues of the upstream router and transmits to the downstream router the one or more virtual pool capacities in response to receipt of a service initialization message.

5. The policy server of Claim 1, wherein said configuration manager configures the one or more data handling queues of the upstream router by specifying (1) a mapping between said one or more service classes and said one or more data handling queues and (2) a scheduler weight for each of the one or more data handling queues.

6. The policy server of Claim 1, wherein said configuration manager transmits a virtual pool capacity to said downstream router in response to a determination that the virtual pool capacity is greater than a currently reserved bandwidth at said upstream router.

7. The policy server of Claim 1, wherein said configuration manager outputs an alert message in response to a failure to install said one or more virtual pool capacities on said downstream router or configure said one or more data handling queues of the upstream router.

8. The policy server of Claim 1, wherein said configuration manager, responsive to receipt of an indication that said upstream router has been reconfigured, automatically transmits one or more new virtual pool capacities to said downstream router to resynchronize said one or more virtual pool capacities with a new configuration of said one or more data handling queues.

9. A data network, including:

a policy server in accordance with Claim 1; and

at least the upstream router and the downstream router coupled for data communication.

10. A program product for use by a policy server in managing a data network including an upstream router and a downstream router, said upstream router including one or more data handling queues, said program product comprising:

a computer-usable medium; and

a configuration manager within said computer-usable medium that configures the one or more data handling queues of the upstream router to provide a selected bandwidth to one or more of a plurality of service classes of data flows and transmits to the downstream router one or more virtual pool capacities each corresponding to a bandwidth at the upstream router for one or more associated service classes among said plurality of service classes.

11. The program product of Claim 10, wherein said configuration manager transmits said virtual pool capacities to said downstream server prior to configuration of said data handling queues on said upstream server.

12. The program product of Claim 11, wherein said configuration manager configures said data handling queues on said upstream server only in response to acknowledgment that the one or more virtual pool capacities transmitted to said downstream server were successfully installed.

13. The program product of Claim 10, wherein said configuration manager configures the one or more data handling queues of the upstream router and transmits to the downstream router the one or more virtual pool capacities in response to receipt of a service initialization message.

14. The program product of Claim 10, wherein said configuration manager configures the one or more data handling queues of the upstream router by specifying (1) a mapping between said one or more service classes and said one or more data handling queues and (2) a scheduler weight for each of the one or more data handling queues.

15. The program product of Claim 10, wherein said configuration manager transmits a virtual pool capacity to said downstream router in response to a determination that the virtual pool capacity is greater than a currently reserved bandwidth at said upstream router.

16. The program product of Claim 10, wherein said configuration manager outputs an alert message in response to a failure to install said one or more virtual pool capacities on said downstream router or configure said one or more data handling queues of the upstream router.

17. The program product of Claim 10, wherein said configuration manager, responsive to receipt of an indication that said upstream router has been reconfigured, automatically transmits one or more new virtual pool capacities to said downstream router to resynchronize said one or more virtual pool capacities with a new configuration of said one or more data handling queues.

18. A method of managing a data network including a upstream router and a downstream router, said upstream router including one or more data handling queues, said method comprising:

a policy server configuring the one or more data handling queues of the upstream router to provide a selected bandwidth to one or more of a plurality of service classes of data flows; and

the policy server transmitting to the downstream router one or more virtual pool capacities each corresponding to a bandwidth at the upstream router for one or more associated service classes among said plurality of service classes.

19. The method of Claim 18, wherein the policy server transmits said virtual pool capacities to said downstream router prior to configuration of said data handling queues on said upstream router.

20. The method of Claim 19, wherein the policy server configures said data handling queues on said upstream router only in response to acknowledgment that the one or more virtual pool capacities transmitted to said downstream server were successfully installed.

21. The method of Claim 18, wherein the policy server configures the one or more data handling queues of the upstream router and transmits to the downstream router the one or more virtual pool capacities in response to receipt of a service initialization message.

22. The method of Claim 18, wherein configuring the one or more data handling queues of the upstream router comprises specifying (1) a mapping between said one or more service classes and said one or more data handling queues and (2) a scheduler weight for each of the one or more data handling queues.

23. The method of Claim 18, wherein said transmitting comprises the policy server transmitting a virtual pool capacity to said downstream router in response to a determination that the virtual pool capacity is greater than a currently reserved bandwidth at said upstream router.

24. The method of Claim 18, and further comprising the policy server outputting an alert message in response to a failure to install said one or more virtual pool capacities on said downstream router or configure said one or more data handling queues of the upstream router.

25. The method of Claim 18, and further comprising:

in response to receipt of an indication that said upstream router has been reconfigured, the policy server automatically transmitting one or more new virtual pool capacities to said downstream router to resynchronize said one or more virtual pool capacities with a new configuration of said one or more data handling queues.

26. A router, comprising:

a data plane having an input port connectable to an upstream link and an output port connectable to a downstream link; and

a control plane including:

a virtual pool having a capacity corresponding to a resource capacity of an upstream router coupled to the upstream link;

a policy control interface through which a policy server installs the capacity of the virtual pool; and

an admission control function that, responsive to a request to reserve resources for a flow from said input port to said output port through said data plane, performs admission control for the upstream link by reference to resource availability within said virtual pool.

27. The router of Claim 26, wherein said control plane transmits an acknowledgement to the policy server in response to successful installation of the virtual pool capacity.

28. The router of Claim 26, wherein said control plane transmits a notification to the policy server to indicate that a reserved bandwidth of one or more flows admitted by said admission control function is less than a new uninstalled capacity of the virtual pool.

29. A router, comprising:

a data plane having an input port connectable to an upstream link and an output port connectable to a downstream link , said data plane including a plurality of data handling queues; and

a control plane including a policy control interface through which a policy server configures which of the data handling queues is assigned to handle each of a plurality of types of Integrated Services flows.

30. The router of Claim 29, wherein said control plane transmits an acknowledgement to the policy server in response to successful configuration of the plurality of data handling queues.

31. The router of Claim 29, wherein said control plane transmits a notification to the policy server to indicate that the configuration of the plurality of data handling queues has been modified.

32. The router of Claim 29, wherein said plurality of data handling queues are configured such that at least two types of Integrated Services flows are assigned to different ones of said plurality of data handling queues.

33. The router of Claim 29, wherein data plane further includes a scheduler that schedules transmission of packets from said plurality of data handling queues, and wherein said policy control interfaces further configures said router by providing the scheduler a scheduler weight for each of the plurality of data handling queues.