ABSTRACT

Resource Management and Use of Voting Protocols – An Application to Network Quality-of-Service
(December 2001)

Sai Ganesh Sitharaman

B.E., Bharathidasan University

Resource management techniques are used in wide range of fields where there are a limited set of resources are available to be utilized by many competing elements. Resource management (such as bandwidth allocation to specific resources in a congested network) techniques allocate and manage resources to applications that has a significant impact on the services quality provided by the resources and the efficiency realized by the system. In this paper, we discuss the network resource allocation and management used for ensuring Quality of Service (QoS) and fairness in today's network. Firstly, today's network has emerged from a simple point-to-point best effort delivery service to sophisticated distributed services with applications demanding services such as voice, video and multi-party conferencing. Secondly, as a simple point-to-point dedicated service is now replaced with multi-party packet-switched networks each demanding an equivalent service quality. Thirdly, as applications are increasingly making use of current network services and as their reliability and scalability increases, they place more demands towards future services.

In this paper, we discuss Resource allocation and management protocols such as RSVP [23], ST2 [8] for Integrated Services [6] and MPLS [17] that make use of existing traditional best-effort network but ELECT certain nodes in the ingress and egress networks elements to act as traffic filters, shaper and policy managers to schedule traffic.

In describing the various protocols used in ensuring service quality in networks, we find it interesting and meaningful to classify and describe the various mechanisms, techniques and algorithms used for resource management used in delivering service quality rather than describing the protocol operations.

This paper is organized as follows: Chapter 1 introduces the importance of resource management techniques in networks and the evolution of resource signaling in networks. Chapter 2 describes in detail how a traditional best-effort model attempted to provide resource management for buffering and bandwidth handling. Chapter 3 details resource management protocols used in various network layers and their design philosophy. Chapter 4 presents queuing disciplines and queue scheduling necessary for optimizing delays and bandwidth usage while storing and forwarding in routers. Chapter 5 discusses the

guaranteed services. We complete our discussion in Chapter 6, briefly mentioning the future trends. Chapter 7 presents the references made in this paper.

1. Introduction[1]

Resource management techniques allocate and manage resources to applications and these techniques have a significant impact on the services quality provided by the resources and the efficiency realized by the system. In this paper, we discuss the network resource allocation and management used for ensuring better quality-of-service (QoS) and fairness in today's network. Firstly, today's network has emerged from a simple point-to-point best effort delivery service to sophisticated distributed services with applications demanding services such as voice, video and multi-party conferencing. Secondly, as a simple point-to-point dedicated service is now replaced with multi-party packet-switched networks each demanding an equivalent service quality. Thirdly, as applications are increasingly making use of current network services and as their reliability and scalability increases, they place more demands towards future services.

First issue demands increased degree of communication and cooperation among existing network aggregating nodes (distribution routers) and distribution service nodes in the network. Second issue requires creation and management of various levels of service guarantees for better shaping and policing the traffic. RSVP Admission Policy [5] (RAP) IETF Working Group requires Policy Enforcement Point (PEP) and Policy Decision Point (PDP) to exchange policies using Common Open Policy Service Protocol (COPS) [11] and the policies specify global criterion algorithms to evaluate service provided. For instance, strict service guarantees are required for hard real-time network services such as packet voice and video streaming, multimedia conferencing, network gaming etc. Weak guarantees includes web services, network file system (NFS) etc. Third issue addresses the scalability and interoperability between several techniques used to manage network resources effectively. For instance, DiffServ [4] and IntServ [6] are techniques used for achieving service quality in networks demanding different resources guarantees.

In simplest terms, RSVP protocol carries requests for a specific flow of the service from the network for a particular type of application flow. RSVP enabled nodes (router/switch) communicate reservation information to Network Admission Control and Policy Control to query for availability in the network. In general, policy enforcing and decision making nodes and the node requesting for advanced services are not predetermined which makes it an interesting problem to discuss in this paper [11].

[1] Reference style sheet followed in this paper is that of IEEE/ACM Transactions on Networking.

A similar protocol ST2 [8] models the network with source as the root and Policy controllers as intermediate nodes. Receivers are added and deleted after initial stream setup and thus new receiver requires a new Connect message with source specifying the exact traffic characteristics.

1.1 Importance of Resource management in network quality-of-service

During the peak deployment and utilization time of IMP in the early Internet suggested that bandwidth utilization and delay characteristics vary due to the misbehavior of TCP retransmission algorithm previously known as retransmission-timeout (RSRE) [24]. The algorithm operates by recording the time and the initial sequence number when the segment is transmitted, then computing the elapsed time for the sequence number to be acknowledged. The retransmission timer is basically an estimation process in calculating the round trip time (RTT). RTT is an important heuristic estimation that approximately estimates the available bandwidth across the path in various networks [25]. The important factor here is that if the RTT is not well estimated, each of the nodes in the best effort network abuses the Internet system by successive retransmissions. A simple implementation such as the early TCP RTT estimation led to the successive efforts to analyze Internet resource utilization and better resource management techniques. This also proves our main first point in the first paragraph that Internet had to be moved from the best effort point-to-point to consideration of other potential resources usages by other hosts in a cooperative manner. A best effort technique with uses a sender-based, receiver-based and centralized router-based is described in section 2.0 [7].

Similarly, data networks were primarily forced to utilize the existing voice infrastructure when dial up computing services used packet services over a circuit switched network. But the trend is now reversed in which the data networks infrastructure is exploited to provide circuit switched quality voice-over-IP services. As today's IP network guarantees several features such as end-to-end voice-quality resource reservation, guaranteed network delay and throughput, VoIP services are extended to further constraint the system by advanced services such as fast directory lookup, or optimized route for better jitter control are being considered. This is an indication of the acceptance of traditional technology while there is a greater requirement for other advanced services. Figure 1.1 below indicates the various overheads involved in voice processing end-to-end from one phone set to another.

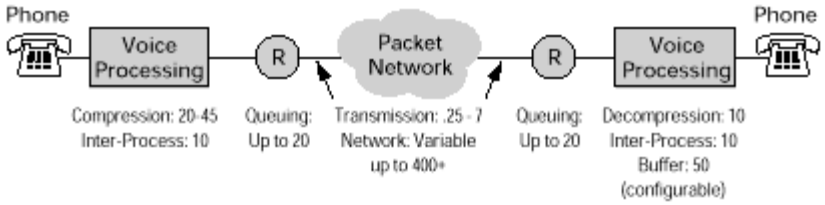

Fig. 1.1 End-to-End Voice processing overheads

In general, in the absence of resource allocation and management techniques, systems find an increased cost of operation due to over-provisioning or a degraded service for critical traffic.

Best Effort Resource Management

Given the importance for attaining better service quality for user in a cooperative manner, one of the techniques used is to optimize resource usages in networks. This may be a simple protocol implementation such as Additive Increase Multiplicative Decrease (AIMD) techniques in TCP [25]. During congestion avoidance stage, the network signals the transport endpoints that congestion is occurring and the sender policy is to apply multiplicative decrease: current window size:

$W_i = d\ W_{i-1}$ where $d < 1$.

Best effort IP network sincerely routes the datagram without any guarantee for delay or delivery to the to the destination. Datagrams may be lost, corrupted or delivered out of order during a network is overrun and congested.

We can see from above techniques that each of the distribution and routing services has buffer and bandwidth as their primary resources to be optimized. In the best-effort model, the network allocates bandwidth among all instantaneous users as best it can and attempts to serve all of them without making any explicit commitment or any other service quality. When congestion occurs, it is in the interest of the end sources and sinks to detect and slow down transmissions or receiving rate so that the sending rate is equal to the capacity at congestion point. However, this does not prevent misbehaving users from not detecting the congestion and overusing beyond its limit. For instance, unlike a TCP source that is accommodative, an UDP application may be transmitting large bursts of datagrams even during congestion [25].

While there are some techniques employed in protocol layering designs, the traditional best-effort IP networks does not achieve a peer-peer resource guarantees. Techniques such as congestion control mechanism are required to assume certain probability of loss and corruption of the underlying links. Transport applications can optimize their resource usage by network routing layer choosing alternative routing during congestion of traditional shortest network route [18]. Best-effort techniques require a high interaction between transport and network layer to perform better under failure conditions. Such interactions among various layers in an implementation can also be considered resource management protocols. A sample model that was studied included a detail specification of these parameters from transport to the network [21]. Every connection oriented transport session requests lower network layer the session's consistent packet size and requested bandwidth. Each network maintains the index of the session identifiers and the requests and calculates the requests from all such sessions from other available connections. Transport packet is encapsulated and forwarded using the network layer only when the request is guaranteed. A sample interaction diagram is presented in figure 1.2 below.

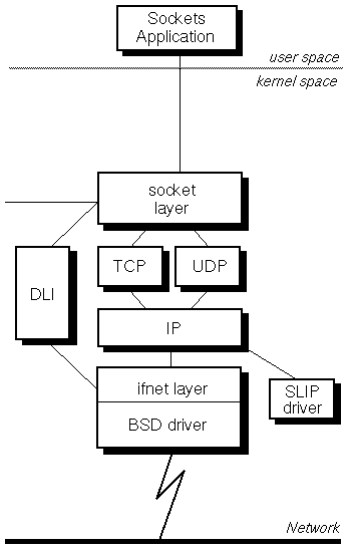

Fig. 1.2 Interaction between Transport and Network layers

This is a sender-based approach and does not scale well in the Internet due to the lack of global synchronization with other cooperating nodes. Each source manages its own resource locally while not estimating the overall bandwidth and buffer requirements used by other applications. Because of these, all transport services are treated with same priority.

1.2 Signaling Requirements in Resource Management

Several resource-signaling techniques are in use in the provision of network quality of service. This section describes some of the foremost important requirements in deploying a distributed framework for resource management. Later sections provide detailed analysis of various signaling protocols and their resource utilization techniques.

Several architectures exist to provide framework for applications to choose between delivery services that provide various traffic characteristics in the network. These are provided due to the following reasons:

- Nodes with various levels of responsibilities are required to provide interoperable resource reservation and management along the path from the host to distribution to core back to a distribution to the destination host again.
- Applications can communicate their resource requirements to the nodes along the path as well as for the network nodes to communicate between one another the resource requirements that must be provided for the particular traffic flows.
- An end-to-end treatment provided by network elements that conform to common standards and policy such as delay-bounded flow or policed services to ensure that the applications receive the same treatment as set in the policy.

IntServ [6] architecture provides two types of services that have different delivery characteristics – Guaranteed service [26] and Controlled Service [27]. Guaranteed service requires support of every service element along the path to achieve a bounded delay for an application. In general, the delay can be: fixed delay or queuing delay. Fixed delay occurs due to propagation and path setup from ingress to egress in the service-enabled network. Guaranteed service attempts to minimize the queuing delay in the network by varying the token bucket size *b*, at each service element and the application data request *R* [10]. If the application experiences a higher delay than expected, these parameters can be varied to achieve a reasonable delay lower delay bound.

RSVP [5] is a resource reservation protocol sent between the signaling elements to achieve the necessary services in the system. Figure 1.3 shows the traffic flow of the RSVP from the sender down to the receiver. Here, the RSVP signaling of resources is done along the complete route, making receiver responsible for the allocated resources. RSVP is a dynamic protocol that maintains soft state of resources along the path and sends periodic traffic characterization

parameters (Sender *Tspec*) that causes the receiver to modify reservation request. RSVP sender sends PATH messages downstream toward an RSVP receiver destination. Path messages store path information in each node in the traffic path. Each node maintains this path state characterization for the specified sender's flow indicated in sender's Tspec parameter. The receiver sending back an RESV message along the same path upstream to the sender indicates successful reservation. TENET [3] protocol suite also follows the same sequence of reservation and acknowledgement upstream to guarantee flow specifications.

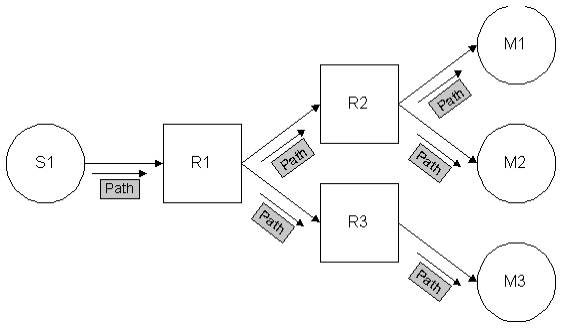

Fig. 1.3 Traffic flow of the RSVP Path Message

1.3 Working Groups in Resource Management

The following workgroups are actively involved in defining new resource management protocols and standards and studying the interoperability issues.

1.3.1 Resource Reservation Setup Protocol (IETF RSVP)

The primary purpose of this IETF working group is to evolve the RSVP specification and introduce it to the Internet. RSVP is independent of service model used and is interoperable among various applications that provide appropriate flow specifications such as traffic specification *Tspec* and flow specification *Rspec*.

1.3.2 Resource Allocation Protocol (IETF RAP)

Resource Allocation Protocol (RAP) is an extension of RSVP providing controlled and enforced access and usage policies. RAP specifications define Policy Enforcement Points (PEP) and Policy Decision Points (PDP) with a core RSVP policy ignorant set of nodes serving these two. Policies enforcements are done at the border nodes between autonomous systems. The workgroup's main objective is to define the policy framework, Common Open Policy Service (COPS) between PDP and PEP service elements and also define Policy Information Base (PIB) [11].

1.3.3 Multi-protocol Label Switching (IETF MPLS)

MPLS integrates a label-swapping framework for network routing layer. Each packet is attached a small label or a tag which is distributed across the MPLS cloud and this label is primarily used to make forwarding decisions without looking into any of the packet headers [17]. This technique along with a traffic oriented and resource-oriented objectives make it suitable for high performance label switching mechanism between layers 2 and 3 [2].

The primary functionality of the IETF workgroup is to provide a standardized label switching mechanism for various link-level technologies such as Packet-over-Sonet, Frame Relay, ATM and LAN technologies. The working groups also standardize the protocols required for exchanging labels (Label Distribution Protocol, LDP) within MPLS-enabled network and RSVP-TE signaling as well.

1.3.4 Audio Video Charter (IETF AVT)

The primary purpose of this IETF working group is to provide protocol specifications for real-time transmissions of audio and video traffic over traditional best effort IP network using unreliable UDP and multicast techniques. One of the emerging protocols used in Voice-over-IP is Real Time Protocol RTP and Real Time Control Protocol, RTCP. RTP provides an end-to-end transport function suitable for transmission of real-time audio or video traffic. RTP does not perform resource reservation nor guarantees service deliveries. RTP is accompanied by RTCP that performs these functionalities. RTCP rests as a transport level protocol performing congestion control and providing a feedback on quality of data distributions [18].

2. Best-Effort network Resource Management

As already seen in earlier sections, the best effort Internet architecture allows all packets to share the buffer and bandwidth resources equally. Link capacity is allocated locally and each sender is greedy to achieve as much throughput as possible. Best-effort eliminates reservation for resources across the path and all packets or flows are created equal. In terms of resource constraints, an upper limit for the delay and throughput are not guaranteed.

In this section, we describe several aspects of the best effort delivery services and some of the prevailing solutions. In dealing with resource constraining, we describe: Static priority tagging for multiple priority disciplines, Static tagging with a guaranteed bound on delay and throughput using existing resources, Policing and Checking constraint violations and finally we discuss techniques based on Receiver, Sender and a centralized router-based approaches [6]. Potential queuing and buffer handling for multimedia real-time traffic without any changes to existing infrastructure is an intriguing solution as well [9].

2.1 Dynamic Best-Effort Provisioning Techniques

The techniques described here are based on three varied approaches to using applications in real-time, non-real time for best effort delivery service: Allocation Capacity framework detailed in [7].

2.1.1 Allocated Capacity Framework

The goal of the Allocate Capacity framework [7] is to allocate bandwidth to the users in a controlled way during congestion control in a best effort service model. The framework defines a service allocation profile for each user and a mechanism is designed in the central routers to prefer those within the service allocation profiles.

While the major changes are made only in the distribution and central routers along the path in the network, the user applications and application resource reservation and reservation approaches remains the same (that is, best-effort).

Figure 1.4 describes the Allocated framework sender based approach. Here, host H1 has a sender-based profile, and is sending traffic to host H2. Routers G all have dropping algorithm D and M is the profile meter in the network.

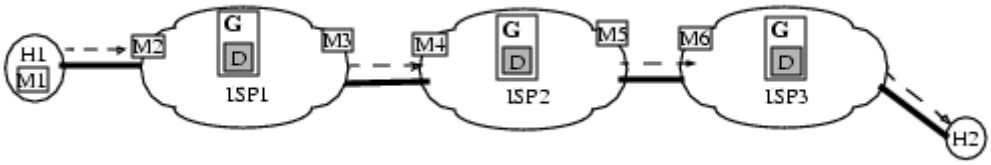

Fig. 1.4 Allocated Capacity Framework Sender-based

In the sender-based scheme, all the routers adopt a preferential dropping algorithm that is a function that checks whether the user traffic is properly defined within the limits of user allocation profile described earlier. A profile meter exists which tags packets at the edge of the network and tags either *In* or Out of their allocation profiles. As the traffic merges and aggregates in the center of the network, those which are *Out* are preferably dropped unless the available capacity is beyond the requested capacity.

The receiver-controlled scheme depends on Explicit Congestion Notification (ECN) and this bit is set when the TCP receiver copies the ECN bit into the acknowledgement packet indicating to the sender TCP to gracefully slow down upon receiving the bit. This technique also helps routers detect or set ECN bits during congestion and dropping those marked.

The advantage of this approach is user scalability in which new applications with variety of service guarantees can still operate over this framework due to the explicit service allocation profile specification per user. For greater level of commitments for these services, either static actions such as long term high-speed link bandwidth guarantees or dynamic policy decisions may be made such as policing and checking policies and using protocols like RSVP. The disadvantage is the modification of transport implementation at the distribution and central network that is proved to be costly in current Internet architecture.

2.1.2 ABE Best Effort Service Model

ABE service model [13] attempts to solve one of the major impediments of the real-time applications with a guaranteed bound on the end-to-end delay between applications. ABE service model does this by using the existing best-effort service delivery model and a simple Explicit Congestion Notification (ECN) mechanism in central routers [25]. Making no distinction or policing (such as the differentiated services) of existing traffic retains the operational simplicity of best-effort delivery model.

In ABE best effort service model, every packet is marked GREEN or BLUE based on the application requirements. GREEN packets are guaranteed a low delay at every router but with a compromise of being dropped during congestions. BLUE traffic receives as much throughput as a normal packet in best-effort legacy network would receive. The choice of coloring packets is application dependent based on the nature of application traffic and global traffic conditions. With this, neither of the packets is received to have better service quality as contrasted by differentiated or integrated services because these use the existing mechanisms without any requirement for additional reservation and profile maintenance [13].

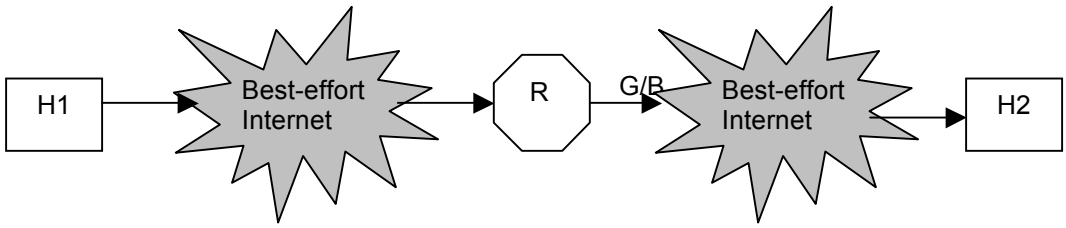

Fig. 1.5 ABE Best-Effort Coloring model using ECN

The operation of the ABE model is fairly simple and is depicted in figure 1.5 above. H1 and H2 are the end hosts in the systems and R is the router that adopts the ABE best effort service model. Each application uses a standard utility function *u(R, D)* for a given throughput R and end-to-end delay D. An simple example would be to have *u(R,D) = 0* for $R < R_0$ and *u(R,D)* is a decreasing function of delay D as $R \geq R_0$ where R_0 is the minimum rate after which the delay become an impediment. In these operating bounds, if the source decides to mark the traffic BLUE instead of GREEN, the service quality attained for all packets marked BLUE is remains same or better. Here transparency to BLUE packets from GREEN is maintained based on transport source rate θ and a probability of forwarding GREEN packet g. This factor g is adjusted based on the source rate and a factor based on whether throughput rate is lesser or same as BLUE traffic. Notice here that the GREEN packets are more likely to be dropped or is marked as ECN congestion bit. When this feature is enabled, the source of the traffic adjusts to its feedback without necessarily dropping the packets.

2.1.3 Best-Effort IP

In a much simpler model of service guarantee, IP Type-of-Service (TOS) field has been one of the most widely accepted in the community [1]. Using traditional IP for service differentiation provided an implementation that is not as costly as a flow differentiation (as in Differentiated Services) and resource path reservation (as in Integrated Services) but still does not requiring

any policy checking, computation or memory overhead nor is there a change in implementation of protocols.

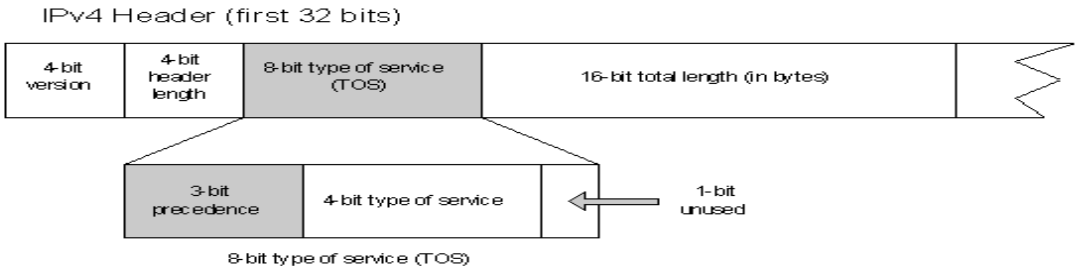

Fig. 1.6 IPv4 Type-of-Service (TOS) Field

RFC 1349 [1] defines the semantics of the 4-bit type of service field as follows:

Type-of-Service (TOS) [High to Low]			Precedence [Low to High]		
1000	-	Minimize delay	111	-	Network Control
0100	-	Maximize throughput	110	-	Inter-network Control
0010	-	Maximize reliability	101	-	CRITIC/ECP
0001	-	Minimize monetary cost	100	-	Flash Override
0000	-	Normal service	011	-	Flash
			010	-	Immediate
			001	-	Priority
			000	-	Routine

Fig. 7 IPv4 Type-of-Service (TOS) Field Definitions

Queuing packets based on these four bits may be made using two generic characteristics: Delay sensitivity and Drop preferences. The first two most-significant bits in the TOS field can effectively be used to map delays in intermediate routers along the path wherein a value of 11 represents the high delay sensitivity and 00 represents the lowest delay sensitivity. All routers use techniques such as class-based-queuing (CBQ) to classify traffic and service the high delay sensitive packets first before the low delay sensitive packets [10].

Similarly, the 3-bit precedence may be used to set “Drop preference” by applications. Drop preferences will be used during congestion or when congestion is imminent. When the network is not expecting or have not detected any congestion along the path, the delay preference is used. Drop preference may be used by router technique such as Random Early Detection (RED) with modified policy for dropping. Instead of marking with fair priority, RED may be modified to lookup the drop precedence priority in the incoming headers and mark those which are of highest precedence set by application. In implementing these techniques, it is assumed that the intermediate forwarding hosts ignore them and does not modify TOS and precedence fields whenever they are not used by them. Other delay and drop precedence policies are possible.

2.1.3 Multimedia Best-Effort Buffering and Resource Control

Several efforts are underway for multimedia bounded real-time applications using best effort delivery service. We investigate a sample technique describing how simple Forward Error Corrections (FEC) and Elastic delay-jitter buffering solution [12].

Applications requiring bounded real-time guarantee has to control various parameters: Packet delay, delay-jitter caused, Packet loss and out of arrivals and finally duplicate packet arrivals. The performance of a playback application such as this has two governing parameters: latency and fidelity. Apart from usage of multiplicative decrease of window in transport protocols like TCP, a multimedia application can ameliorate the problem by introducing sufficient redundancy into the data stream to enable the receiver to reconstruct the packet on the fly. A simple technique would be to replicate data stream and group them into multiple data streams with some form of identification of the streams. Receiver identifies duplicate streams and they may be discarded. A scheme like this is proved to increase the delay of the data streams by 1.5-2 times [12] but still is bounded delay and hence predictable. At least expected to reach the receiver if one of the packets is corrupted or lost.

Elastic queuing and queue monitoring techniques are useful in adjusting to variance in instantaneous busty arrival rates. Given an average buffer size m, if we know the buffer size changes by $\pm k$, over a period of time, the elastic buffer size will have a size of $m+k$. Any increase in the size of this buffer will only application latency in dealing with extra queuing overhead of the buffers. Accompanied with this elastic queue is a queue monitoring technique in which a count and a time threshold are associated with each position in the queue. A count for position k in the queue represents the duration (packet inter-arrival times) that the queue always contained at

least *k* elements. When a packet arrives at the receiver and the play-out queue currently contains *n* items, the counts for positions *1* through *n* are incremented. If the count for any queue position *k* between *1* and *n* exceeds its threshold, then the arriving packet is dropped. This indicates that the queue has contained at least *k* elements for sufficiently long time that we believe the current level of jitter is less than *k* sample inter-arrival times. By dropping a packet, the media sample acquisition is reduced [12].

Other novel architectures are provided for scaling existing routers to support differentiated services in best effort networks. One such approach is defined in [15].

3. Layered Resource Management

To study an end-to-end behavior of applications requiring service quality, it is required to classify the various resources in terms of their usages. For instance, a real-time multimedia application may classify its requirement into: Network buffer capacity, Bandwidth requirement, Bounded network delay and throughput, Operating System support for network delay-jitter buffer, Large storage support, Support for queuing policies and protocol standards and support for the specialized network interfaces cards that implement in-built link service techniques (such as LLRM).

Applications may require support for frame correction errors, compression and synchronization techniques. Typical applications may be layered under various contexts and a sample layering structure is shown in figure 1.7 below.

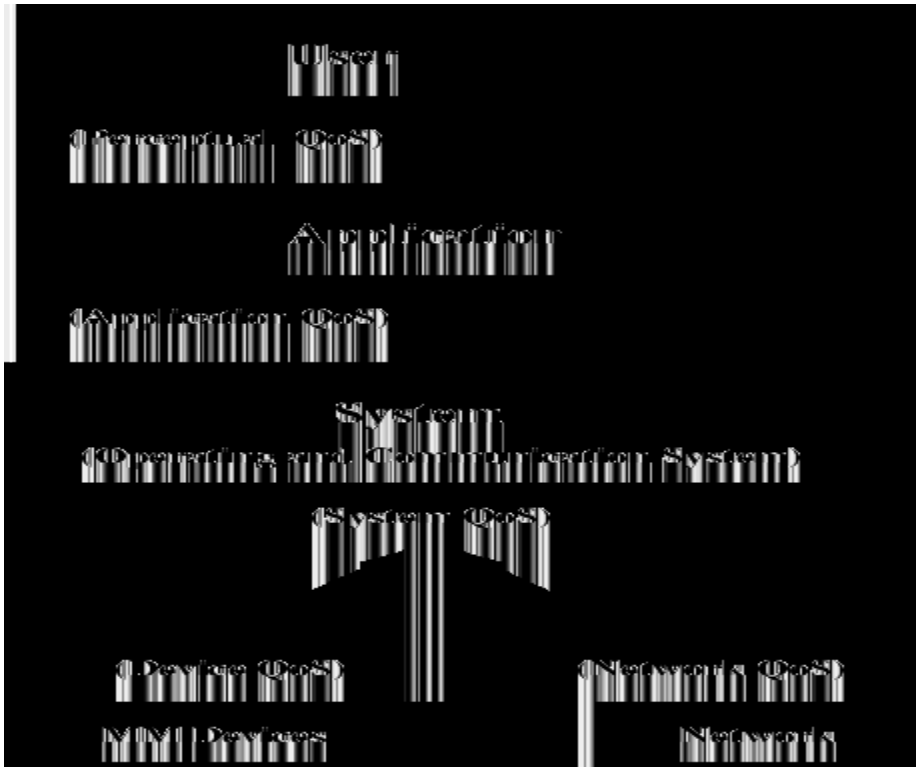

Fig. 1.7 Layered Model for Quality-of-Service

In the interest of our focus towards network services, we investigate the support available to study resource definitions and management techniques available at various network service

layers such as link layer, network and transport layers. At each layer, we describe the protocol functionality, the interfaces with higher layers in the system and method of resource allocation and management.

3.1 Link Level Resource Management Protocol (LLMR)

Link Level Resource Management Protocol (LLMR) [14] is a link level resource signaling protocol used to reserve resources in a shared medium, a bridge or a switched LAN infrastructure. The protocol interoperates with heterogeneous medium and is not dependent on any particular network level initiation procedure or protocols to reserve or update its local resource database. LLMR is known to operate on Ethernet, FDDI and Token Ring. The reservation setup for those segments will only differ with regard to link specific messages to be sent and the specific implementation of admission control algorithms.

LLMR uses a special value in its *ethertype* for carrying control information between end hosts and intermediate bridges. This makes it difficult to interoperate with existing infrastructure that does not support LLMR in the first place [14]. Details of the LLMR resource reservation and protocol operations are presented in [14]. Following sections refer to the same document.

3.1.1 LLMR Resources

LLMR uses only one message to perform reservation among many heterogeneous nodes in the network and uses only one query message to update and remain consistent with neighboring nodes and with other participating nodes. Each LLMR reserve message includes a detailed specification of the resources in the system. Following are presented to any nodes interested.

Tspec

Traffic Specification characterizing the data stream injected into the network. Determines exactly application traffic specification requirements which is determined by admission control algorithm implemented network-wide. Apart from application data stream, this may contain separate resource reservation for link control messages such as reserve, query and other control messages. *Tspec* parameter is added to the local rate regulator module when the admission is permitted and agreed upon. Examples of *Tspec* includes: Average Rate, Maximum burst size, Peak Rate.

Rspec

Identifies the receiver specification against each potential appropriate receiver identifiers. In a multicast mode, a single multicast server may be specified which is then distributed to many in the multicast network. Using unicast mode, several list of receivers may be specified accompanied with a common receiver flow specification. Each of these receiver identifiers is generally the MAC identifier of the network interface. Examples of *Rspec* includes: Reservation Delay, Slack Delay.

Similar reservation flow specifications are specified by network resource protocols such as RSVP. For instance, the RSVP RESV (reservation) message carries the *Rspec*, the receiver specification since RSVP is a receiver-oriented signaling mechanism. Each successful receiver reservation is acknowledged by a PATH message upstream along the same route as RESV message but carrying the exact traffic specification required in *Tspec* parameter [23].

3.1.2 LLMR Resource-Receiver Model

LLMR protocol is a sender-based resource reservation setup mechanism since it allows the sender to specify many receiver resource characteristics with a single message. Such a model also helps sender establish a valid end-to-end delay bounds across heterogeneous systems across multiple networks at least in the link layer.

In the link reservation receiver model, different receivers are specified using receiver identifiers that are their MAC identifiers generally. Each identifier has a corresponding *Rspec*, receiver traffic characteristic as set by application as defined above. For a multicast network, the receiver may contain the complete list of all the nodes in the network being requested or to a special receiver identifier of the multicast router.

3.1.3 LLMR Resource Reservation Model – Optimistic Approach

LLMR implementation maintains three local elements for admission control and reservation. A local database is maintained which contains the current *Rspec* for all the receivers along the path mentioned. This maintains the receiver identifier against the currently allocated flow specifications, the values of which available during previous successful requests. The local classifier and local rate regulators are used to act on the outgoing and incoming packet when admission control is successful during previous requests.

Before initiating reservation requests on the outgoing link to other nodes, the local node checks against the local reservation database to check if there was a previous successful reservation

sent along this receiver path. Since the database is updated periodically by query messages (the receiver query message can also be configured by control messages), this local lookup serves multiple purpose: to avoid further reservations requests if the reservation is larger than previous requests, and to verify if the current request may be satisfied by aggregating this flow with earlier flows along the same path.

A successful local lookup then initiates a reservation message request with *Tspec* definition and *Rspec* along with various receiver identifiers. Each node responds with an acknowledgement after a certain random time indicating whether this request may be satisfied. However, a special node, the arbiter, may immediately respond to reservation and query messages.

LLMR uses a distributed arbiter and queries other nodes in a distributed fashion. This mechanism is required to ensure that the loss of reservation message during congestion, does not cause the reservation message be completely discarded and ignored. However, if the set of nodes in a subnet fails (destination network, say), obviously this cannot guarantee the true reservation statistics along the path. Inconsistency in the data is also updated by periodic refresh messages.

3.1.4 LLMR Resource Reservation Model – Pessimistic Approach

LLMR implements another approach for robustness of reservation messages and also removes any potential inconsistency of data. In this pessimistic approach, each request is not updated to local database or forwarded to admission and rate control unless an acknowledgement is received from nodes in the nodes. Only after this acknowledgement that all nodes appropriately updates the *Tspec*.

3.1.5 LLMR Resource Reservation Protocol Messages

LLMR contains five control messages to reserve, maintain consistent reservation state on all the nodes on the segment. Each of these are described here without reference to message structure.

Resource Reservation Message

> Message for reservation, modification of flow and team down of resource reservations. This is sender based reservation scheme in which the traffic and receiver flow specifications for each receiver is specified (*service_id, Fspec, Tspec*, *Rspec*).

Reject Message

Rejection of reservation in response to resource reservation message. This may occur if there is inconsistency in the local database.

Query Message

Used by a node (and even other nodes in the segment) to learn the reservation state on a local segment as the query is responded by all nodes will report their reservation states.

Acknowledgement and Error messages

These are used to respond to reserve and query messages.

3.2 Subnet Bandwidth Manager: IEEE 802 style networks

Resource Reservation Protocol (RSVP) is a signaling mechanism that supports requests for network specific resources such as bandwidth. RSVP protocol however remains aloof of the underlying link layer technology supported and a technique such as LLMR described above requires explicit link layer and network layer management. Subnet Bandwidth Manager (SBM) is a signaling method and protocol for IEEE 802-based LAN based admission control for RSVP type of traffic flows [22]. SBM uses most of the RSVP messages such as RESV and PATH to bring up, tear down and update resource reservations [5].

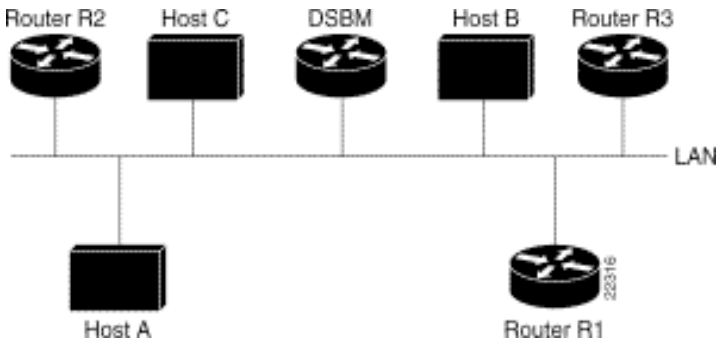

Fig. 1.8 DSBM Managed Segment

3.2.1 SBM and Network layer protocol RSVP

SBM-based admission control only limits RSVP-enabled traffic on the shared LAN. This means, a RSVP traffic and SBM traffic is indistinguishable by intermediate hosts. To allow RSVP-enabled traffic coexist with best-effort traffic in non-SBM compliant, the traffic is treated equivalent to integrated service admission control procedure. In general, some form of traffic

control and policing is expected in the LAN for the RSVP complaint nodes to be treated separately from the best-effort traffic.

All the SBM components utilize RSVP RESV and PATH messages to exchange traffic requirement exchange between them, that is, *Tspec* and *Rspec* (Traffic specifications and Receiver specifications). However, instead of the original destination used in the RSVP message, they are all forwarded to unique arbiter DSBM. DSBM in turn modifies the route and adds its own hop to the path message that is later used by the destination to route back to the sender. Unlike the RSVP, the DSBM may take decisions on the availability of the bandwidth on the segment in which it resides and thus may initiate or forward requests or may reject. As is obvious, both the L2 and L3 route identifiers are appended in the PATH message to ensure that DSBM looks up the appropriate identifier and other L3 uses this and treats as an RSVP message [5][23].

Details of SBM protocol operations and message definitions are described in [22]. Following sections refer the same document.

3.2.2 SBM Message Segments

DSBM

Designated SBM is a management unit that performs admission control and a protocol entity for many nodes to forward requests to it and process it locally within a segment. DSBM is aware of resource utilizations in the current segment.

DSBM client

Layer 3 entities that utilize admission control facility provided by DSBM in the shared SBM segment.

SBM

Layer 2 and layer 3 protocol entity that is protocol entity (sometimes called “device”) and capable of managing resources in the SBM segment. Each SBM segment has a single elected DSBM entity.

3.2.3 SBM Admission Control Procedure

Each segment is uniquely managed by a single DSBM managed entity that controls the admission control procedure for the entire segment. In cases of more than one SBM is attached to the segment, the DSBM takes the management control of other entities in the system as well.

Before a DSBM admission control procedure takes place, a distinguished member in the SBM segment and sub-segments is to be elected. This procedure happens when each SBM entity joins the network and waits for an announcement from DSBM. If none present at any moment, the SBM volunteers itself by sending a broadcast with its corresponding SBM priority identifier to the whole segment. When more than one candidate exists, the resolve is made using the relative SBM relative priority. For a tied priority values, the IP address is used to find the lexicographically highest IP address wins the tie.

Each DSBM initiates a link-negotiation procedure to collect the existing fraction of the resources in the segment managed. Link resources may here be bandwidth, link capacity etc. These resources are currently statically mapped using some static management procedure. Dynamic discovery may be performed but the DSBM is not currently capable of doing this.

DSBM client initiates a normal PATH message as it would in a normal RSVP method and instead of forwarding the RESV message to forward to the destination mentioned in *Rspec*, it is forwarded to the local DSBM on the LAN segment. DSBM managed entity just inserts its own hop identifier to the PATH message so the symmetric case would actually be forwarded to the DSBM entity before being forwarded to the actual source of the reservation status. DSBM thus adds the layer 2 and layer 3 hop objects in PHOP objects (PATH HOP identifier objects) [5]. After a successful reservation, the RSVP RESV message is forwarded to all of the hop entities in the original PATH message of which DSBM entity is one. The DSBM merges the request grant with its own local aggregation and forwards the request to the appropriate host or sub-segment. For the reservation path that contains more than one managed segment entity DSBM, each of these DSBM receive the hop objects setting up a PATH state at each DSBM. During a reservation grant state, each DSBM updates its intermediate DSBM routes before forwarding the next managed entity in the route.

3.2.4 SBM Traffic Class and User Priority

In order for the integrated layer 2 and layer 3 reservation techniques to be differentiated from best-effort traffic, some type of per-flow policing is done to ensure that the flows do not exceed the traffic capacity. However, IEEE 802.1 has a technique to provide a user priority set in the MAC outgoing packet with different traffic classes. Layer 3 traffic TCLASS objects used in the RSVP PATH message may use this user-priority. TCLASS objects are inserted by a layer 3 entity and is removed by the next layer 3 entity keeping a mapping between the user priority and the TCLASS. Alternatively, DSBM may itself add TCLASS object to the PATH message before

forwarding it to other SBM entities. Whenever the SBM entity receives a RESV message, it uses the TCLASS object and user priority to create traffic class locally to use for admission control locally on the segment. Following is the IETF format defined for the TCLASS object.

Only 3 bits in data contain the user priority value (PV). Whenever the DSBM receives a RESV message with TCLASS object out of its own specification of the traffic class, it basically lowers the traffic to the next service class available in the system. This allows only compatible traffic class definitions to exist between the SBM clients and DSBM in the segment.

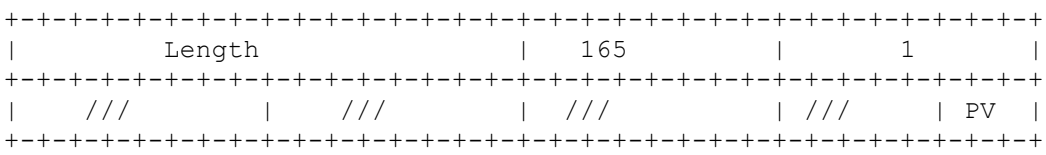

Fig. 1.9 TCLASS object definition in RSVP message

Whenever a SBM or DSBM merges two different RESV, the TCLASS object may or may not define the same object. If they are different, then an RESV_ERR message is generated to the latter or newer TCLASS object. If the objects are the same, it is merged and forwarded using the same message processing rules defined earlier.

3.3 Multi-protocol Label Switching (MPLS) – Traffic Engineering

MPLS integrates the label swapping paradigm with network layer routing to achieve improved network layer routing, routing services and achieving better traffic engineering by load balancing. MPLS uses traditional layer 3 routing services to define labels, define forwarding paths using the labels and a set of label distribution methods between participating Label Switch Routers (LSR) to exchange path information [2]. MPLS uses the predetermined labels and paths to destination corresponding to each label maintained in the Label Information Base (LIB). MPLS operates independent of underlying routing protocols and underlying link layer technologies. Route formation is done using distributed routing mechanisms such as OSPF or BGP that performs detection of loops. MPLS flows are aggregated and called traffic trunks with multiple trunks established between nodes with an associated priority between them that is used in load balancing. Each flow is a Forward Equivalent Class (FEC) that maps to an unique label with an aggregation of same type of flows [17]. With the predetermined support for such path and traffic engineering, MPLS supports network resource-signaling mechanism such as RSVP readily. The MPLS working group has also established constraint based label distribution protocol (CR-LDP) that supports implicit and explicit routing.

Figure 1.10 refers to the MPLS header structure and the various fields [17].

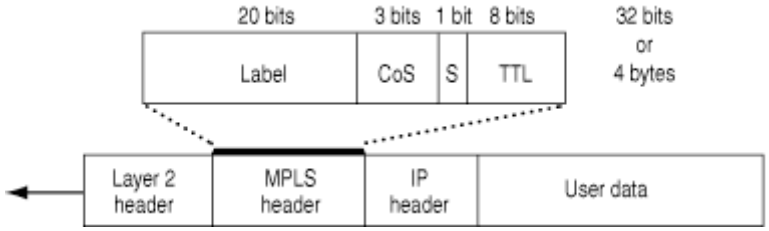

Fig. 1.10 32-bit MPLS Header

Label:	Value of the MPLS label
CoS:	Service class (used in scheduling)
S:	Support for Hierarchical Label Stack
TTL:	Time to Live

Label Switched Paths (LSP) are a sequence of labels at each and every node along the path from the source to the destination and are established prior to reservation (control-driven) or upon detection of flow of data (data-driven) [17].

Although MPLS recommends usage of RSVP and CR-LDP, it does not mandate usage of any label distribution protocol. Existing IGP such as BGP and OSPF may also be used to perform label distribution between heterogeneous networks[5][6]. A CR-LDP protocol use the routing layer support but performs explicit routing based on class of service requirements and thus is named “constrained”. Such “constraints” may include link bandwidth, delay, throughput and number of hops in calculating the best route based on QoS [17]. An important overall objective is to achieve better traffic performance using QoS routing techniques.

Figure 1.11 below demonstrates a label distribution signaling technique used in MPLS.

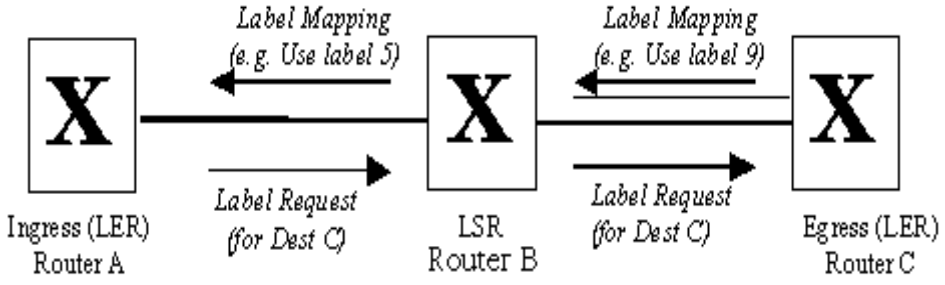

Fig. 1.11 Label Exchange signaling mechanism

3.3.1 MPLS Traffic Engineering for Resource Management

Traffic Engineering refers to the process of selecting the paths chosen by data traffic to balance the load in the network between switches, hosts and routers of heterogeneous networks. One of the key objectives of MPLS traffic engineering model being the resource optimization using load-balancing techniques [19]. Link bandwidth being the crucial resource in the network, the framework provides mechanism to traffic streams are efficiently mapped to available resources in the network by ensuring that the no path remains underutilized while other paths are congested [25]. In general, MPLS attempts to solve the problem by well-studied congestion avoidance techniques plus applying new load balancing algorithms. Congestion avoidance includes: Window rate control, queue management and choosing scheduling techniques. Traditional IGP or autonomous protocols such as OSPFv1, BGP perform route selection based on static parameters such as number of link hops (OSPFv2 is enabled to add certain other characteristics of intermediate routes such shortest path for each IP Type of Service ToS field). These routing mechanisms themselves suffered under several network congestion conditions due to heavy load that results in further instability in the ability to find the appropriate routes. The problem however persists even after deployment of the IGP routing solutions to MPLS but due to path resilience and constrained routing, alternative path may be chosen even though they may be non-optimize routes.

Following sections refer to the same reference [2].

To support better traffic optimization techniques, the following are some of the main objectives of an MPLS enabled domain. Each of these is addressed below:

- Mechanism to map incoming data packets to a particular FEC and an unique label that is distributed among LSR nodes along the path. Before labeling, each ingress Label Edge Router (LER) performs filtering and maps of the incoming traffic to appropriate service levels.
- Provision of Label Switched Path (LSP) for each possible destination along the path.
- Uses QoS routing schemes with constrained parameters such as link bandwidth, delay and capacity to establish “constrained” paths to all possible destinations. The Label Information Base (LIB) may be updated with newly added labels and they may be either added control-driven (initial management setup) or data-driven (as and when request data arrives).

3.3.2 MPLS Traffic Characteristics

Another main objective of the MPLS traffic-engineering model is to certain aspects of the traffic streams such as minimize delays, establishing bounds on delays and throughputs, and minimize packet losses. Yet traffic parameters may be used to capture certain characteristics: traffic peak rates, average rates and burst size.

MPLS defines traffic trunks and associates certain traffic properties with these “objects”. A Traffic trunk is an aggregate of traffic flows belonging to a same class and it allows inclusion of multi-traffic aggregates. Traffic trunks use Label Switched Paths (LSP) to switch for every destination in the MPLS domain.

Traffic trunks may be considered objects over which certain operations are valid. These include: Creation of a traffic class aggregation, Activating traffic class (routing to appropriate destination), Rerouting, Deactivating and Modifying traffic attributes and finally while inactive and finally Destruction of traffic class.

Table 1: Label Switch Path Attributes

LSP Attribute	**Attribute Description**
Maximum Bandwidth	Required maximum reservable bandwidth for specific data stream using this LSP path.
Color	Resource class used in the next label to map to appropriate FEC.
Path	A sequence of labeled routers with the first one being the ingress. Each path packet contains label stack in LIFO order of labels. Each sequence ends with egress node.
FEC	Forward Equivalency Class bound to the next label in the next hop entry in the label stack of path packet P.
LSP Setup Priority	Priority for resolving resource ties between LSPs.
Resilience	Choosing optimal path during failures.

For traffic engineering, the total number of LSP paths is bounded by the number of LSR routers in the network and the number of service classes supported. As per RFC, the number of trunks (or LSP's, or reservations) is $N * (N - 1) * C$, where C is the number of service classes, N is the number of MPLS-speaking routers. This is the N^2 problem we are talking about here. In case of

intra-domain, *N* is the number of border routers at backbone edge, and is in the range of *~100*. To setup a full-meshed network to support traffic engineering, the ISP needs indeed 10^3 or 10^4 trunks. The current RSVP/LDP should have no problem for intra-domain.

3.3.3 MPLS Resource Affinities in Traffic Management

MPLS traffic-engineering model allows specification of resource class and supports mapping of these resource classes against path calculation to any destination. The resource class may be established in the ingress LER entry point router in the MPLS-enabled domain and are distributed to other MPLS enabled nodes in the path. These resource classes are set of policies that are different and impose more constraint along the path.

Each traffic class is then associated with an affinity factor to indicate whether the resource class will be used in "constraining" traffic trunks. The constraint relation is whether to include or not include those resources in the traffic trunks along that path.

{ Resource class, Resource Affinity }, { Resource class, Resource Affinity },….

A constraint-based routing scheme uses to calculate explicit routes to destination that are subject to the appropriate resource-affinity constraints.

3.4 Resource Reservation Protocol (RSVP)

Resource Reservation setup Protocol is used by a host to better service quality for requests along the path of the flow. RSVP protocol is primarily designed for Integrated Services architecture but as shown by multiple publications, may be combined with differentiated services as well. Link sharing protocol LLMR uses RSVP style reservation mechanism in which the flow and traffic specifications are maintained very similar to RSVP. RSVP operates over IPv4 and IPv6 using the underlying routing protocols for control communications to the destination. RSVP design goals and protocol specifications are detailed under [5][23].

Following sections refer to the same reference [5] and [23].

3.4.1 RSVP and Protocol Features

RSVP treats senders and receivers as logically different entities even though the sender and receiver may be the same application residing in the host. RSVP provides a simplex reservation mechanism, to reserve resources in only one direction along the path. This is deliberately one of

the design goals as the protocol is receiver oriented. All resource allocations thus made receiver-responsible and receivers are responsible for resource initiation. This accommodates heterogeneous receivers to participate in a multicast session. This goal leads to flexibility and scalability of the protocol and also allows receivers to dynamically join group without explicit reservations requests.

RSVP makes each sender make an independent reservation request along the path to destination. While this approach may lead to several independent branched requests, they may share common link along a multicast tree. Reservation across shared links may not truly be utilized to the complete extent depending on the application. An multimedia lecturing type of session has only one source always initiating the data stream and the hence it is sufficient to have only that source identified and corresponding bandwidth reserved. RSVP permits such treatment by reserving resources along specific path to reflect the actual resource requirements along the path.

Whenever route changes occur, this must clearly be reflected across the complete RSVP path for current reservations as well as future ones. To enable this RSVP PATH message is sent from source during resource initiation along the route towards destination. A successful reservation leads to an acknowledgement along the same path back again to the source. In some circumstances, it is possible for reservation to fail because of the congestion or link failures in the established path agreed upon earlier. Due to the lack of QoS constraint-routing (as achieved by MPLS CR-LDP), the protocol makes use of light-weight refresh mechanism to maintain a “soft-state” along the path. During multicast to several receivers, it is possible that these query messages flood and lead to protocol overhead. It can be seen that a multicast group that has subscribed to several sources S requires at least S such query requests every refresh time period. To avoid such a protocol overhead, RSVP provides a receiver configurable parameters that may be appropriately set and returned during the initial PATH message to source. Each reservation message carries a *flowspec*, a reservation style, and a packet filter if the reservation uses a filtering mechanism (fixed or dynamic). Both PATH and RESV messages carry back a timeout value that the intermediate value uses for periodic refresh messages.

3.4.2 RSVP and Protocol Features

RSVP uses receiver-oriented reservation and makes receiver fully responsible for reservations along the path. This allows flexibility and scalability of the protocol while accommodating

heterogeneous receivers to reserve different amount of resources at different time based on their capacities. Some of the important features of this approach are:

- Receiver is aware of the capacity limitations and hence its capability for reservations. Receivers may indicate the sender of the necessary flow specifications or may acknowledge the sender's flow specifications with the RESV message.
- Allows requests to bundle flow specifications to multiple heterogeneous receivers in single requests and allow receivers to indicate the closest match.
- For multicast receivers, it is possible for senders to send to a single multicast group leader. Receivers may dynamically join or leave the group or switch to other sources without initiating new reservation requests.

3.4.3 RSVP Soft-state

RSVP keeps track of soft-states at all the intermediate nodes along the path and leaves the responsibilities of maintaining the actual reservations states to the source and receivers. This allows protocol to be more scalable as members join or leave the group. Soft-states are initiated at the intermediate nodes by the PATH message sent to from the source and RESV message from one of the receivers along the path. PATH and RESV messages are sent at specified timeout interval to periodically refresh the state information. Each source sends PATH message that contains the *flowspec* and the filtering flag F that it keeps track to remember the original source of the message. Filtering keeps track of the hops along the path from the source and is an extra state information maintained in intermediate nodes. Whether the filtering is specified or not, each PATH message has a table specifying the flow specification against the incoming link interface against an outgoing interface [23].

Each intermediate RSVP enabled nodes keep track of the following information:

- Reservations made along the path
- Source filter of this reservation request
- Reservation style and the associated filters for sources. Reservation styles concerns treatment of different senders within the same session.
- Sender of the resource reservation acknowledgement message (one of the receivers)

PATH and RESV messages are also sent to initiate change in requests or path information to the receivers. Intermediate nodes encountering this new message will forward requests only when a previous request from the corresponding source and receiver is active and if there are changes to those reservation requests (such as routing information changes). Also, the

intermediate node never forwards the request information on the same interface it received the change request [23].

3.4.4 RSVP Reservation Styles and Filtering

Resource reservation determines the type of packets that may be used by the entity in the reservation message. As we have already seen, the reservation is performed using a receiver-oriented mechanism. A packet filter allows selecting those packets that can use a resource and is set by the reserving entity (the receiver). Thus allowing the receiver to change the packet filter, without change the amount of resource, we may change the course of resource utilization.

In a multicast enabled network, an intermediate router uses the reservation styles (use of packet filtering) to aggregate the reservation requests from many receivers in the multicast group. RSVP enables three types of filters: *no-filter*, *fixed-filter* and *dynamic-filter*.

A no-filter specification allows any packet destined for the multicast application may use reserve the resources. A typical example is to have many audio sources and if all audio sources are required to be active at the same moment (audio conference), this reservation style may be used.

A fixed filtering mechanism uses a list of sources that overlap, the packets of whom only may use the reserved resources. Thus the receiver receives data only from the sources listed in the original reservation request.

One of the most important features of RSVP is the provision of dynamic-filter. While the resource reservation constraints the reservations to be made along the path, the filtering controls which packets may utilize the bandwidth. In *dynamic-filtering*, each receiver receives just enough bandwidth for maximum number of incoming streams it can handle at once. Thus a dynamic filter reservation style accommodates receivers to change its filter to different sources from time to time. In any case, the total amount of the dynamic filter reservations made over the link may be limited to the amount of bandwidth needed to forward all upstream data [23].

3.4.5 RSVP Relationship with LLRM protocol

LLMR requires network service protocol such as RSVP to initiate LLRM request over a particular bridged LAN. Reservation protocols first establish their requirements with the corresponding peers and initiate a special link level service identifiers for LLRM [27]. LLRM performs

reservation request and if successful, distributes all per-flow information to routers, bridges and switched LAN between sources and receiver(s). LLRM relies on the soft-state approach to disseminate state information like RSVP. Reservation model is sender based and is achieved segment-by-segment in reserving, tearing down and refreshing reservation states [14].

In this model, RSVP makes a reserve request to LLMP after it has received PATH and RESV messages for this flow. A successful RESV message contains (*Tspec*, *Rspec*) that is needed in the LLMP protocol requests.

RSVP uses three different sender selection and reservation styles: fixed-filter, shared-filter and wildcard-filter or no-filter. Fixed filter establishes reservation for explicit senders [23]. To map this to LLMP, each fixed reservation characteristics is mapped to the corresponding LLMR protocol. These include the service identifiers, flow specification, traffic specification and receiver list.

LLMR: *FF (service_id, Fspec, Tspec, Recv_R)*

For a shared filter, shared reservations are made only if the flows from different sources share the same outgoing interface. If same segments of link layer are used by multiple sources, then distinct reservations have to be made. A wildcard specification allows reservations to be made with all sources in the medium.

4. Queuing and Queuing Disciplines

Queuing is an act of storing packets or cells for subsequent processing. Queue management that includes algorithms for placing the packets into a queue and also scheduling which packet is to be serviced next are important criteria for guaranteeing service quality in a system. Routers need new queuing techniques to support differentiated services instead of traditional rate-based queuing techniques [10]. Queue implementation such as token-bucket need to have appropriate queue size to accommodate application requirements. Imposing a bigger queue sizes may introduce unacceptable amount of latency and round-trip times.

4.1 Class-based Queuing (CBQ)

Class-based queuing (CBQ) is a variation of priority queuing technique that has several classes of output queues defined. CBQ assigns multiple attributes to assigning the traffic to the class and assigns an average rate to each class as they become available. The preference in the output queue and the service is done by measurement in bytes and priority of each these differentiated traffic. A priority queuing technique like this node requires router to look each packet in detail to determine how and which output queue this packet must be placed [10]. Other computation overhead involves: packet reordering and filtering based on the particular Class of Service (CoS) [9].

CBQ-enabled routers process traffic in each queue until the byte count exceeds the configuration threshold or the current queue is empty. In either of the cases, the scheduling of queue processing is given to the next immediate queue in the priority list. Providing a threshold value and an associated scheduling algorithm prevents traffic in other lower priority (class) queues from starvation. However, due to the overhead involved in calculating the details of the packet and classifying to specialized classes, the CBQ does not scale well for large differentiated service classes [10].

4.1.1 CBQ and RSVP

CBQ is a queuing discipline existing at the routers differentiating packets into various service classes. RSVP protocol enables reservation setup along the path to the destination. It thus seems possible to compare these two techniques and to observe if one may coexist with other in an implementation.

If all the heterogeneous nodes along the path to the destination implements a static class based queuing technique with the policy for classes clearly differentiated, RSVP itself may not be required in the setup. For instance, for each of the service class preferred by RSVP flow specification, a corresponding static class queue may be created. With CBQ, each class then gets an “appropriate” bandwidth as provided by the dynamic scheduling policy [10].

On the other hand, a CBQ implementation along with a reservation protocol may enable better and efficient service compared to just a reservation protocol implementation. A flow within a higher priority may dynamically reserve resources using the RSVP protocol and within each of this flow, the queuing discipline may be FIFO. Other services such as guaranteed services may form a separate class of their own. In dealing with specialized services at each class, one of the main problems is dealing with admission control algorithms and queue classification filters. Increased complexity of queue classification leads to more computational overhead [10].

4.2 Weighted-Fair Queuing (WFQ)

Weighted fair queuing is a variation of fair queuing that allows different flows to be assigned different weights. A fair queuing technique maintains separate queue for each flow at the router and services them using round-robin scheme. Each flow is assigned a “fair” share in the router and those flows that “misbehave” the share are penalized. WFQ servicing algorithm attempts to provide predictable response times. The scheduling algorithm prevents larger flows from consuming network resources thus ensuring the other flows are not starved in the system. Intermediate routers in the system need to know the relative weights of such a classifications with other routers in the system. This may be achieved by network management static provisioning or by other dynamic provisioning and reservation protocols [9].

An example of flow based WFQ would be to have the source TCP, UDP addresses and port numbers based classifications and assigning system port numbers (<1024) a greater weight than other ports in the system. Another approach is to use IP precedence bits (Type of Service) to weight the individual flows – the higher the delay sensitivity precedence, the higher position it holds in the queue [9].

Like the fair and priority queuing techniques, WFQ need to look into the packet headers to perform packet reordering and packet filtering to place the incoming flows into appropriate queues. WFQ requires more overhead computation especially due to the fact that a FIFO is

maintained within a certain fair queue of a flow and scheduling is based on weight calculation. Static weight calculation technique may be applied in calculation of the weights for the flows [10].

5. Guaranteed Service Quality

In this section, we deal in detail the specific architecture of Integrated Services (IntServ) [6] and apply new resource reservation and management techniques. The Integrated Services architecture was originally designed by IETF to provide a set of extension to the best-effort traffic delivery model. The architecture requires clearly defining the services to be provided and mean of providing those services (such as RSVP for resource signaling and CBQ for queuing disciplines, admission control techniques and other end-end requirements). The architecture also requires most of the routers and application services to be enabled with the uniform integrated services. That is, these intermediate nodes be IS-capable (not just IS-aware). The following figure 1.12 gives and overview of the integrated services architecture of a router system.

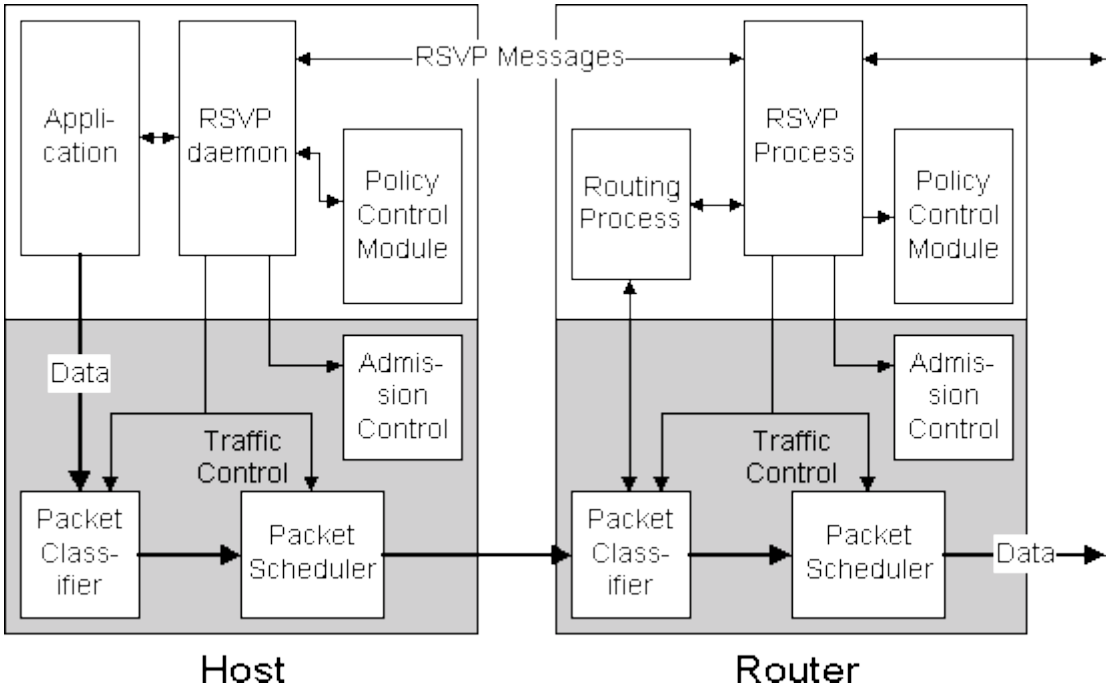

Fig. 1.12 Integrated Services Framework – Host and Router

One of the major assumptions of the Integrated Service model is that resources (such as bandwidth and buffers) are to be explicitly reserved for applications. The integrated services architecture introduced major change due to the fact that intermediate IS-capable systems now requires to have flow specific state information as opposed to just having them at the end hosts.

Flow specification may include details of the type of "conversations" between end hosts. Another significant model that Integrated Services propose is to have both real-time and non-real traffic characteristics in the framework. For real-time application guarantees, the latency is the cumulative sum of the transmission times and queuing hold times [19]. Since the real-time applications do not wait for late arrival of packets, this imposes an offset prior to processing [5].

5.1 Necessity of Controlled and Guaranteed Integrated Services

Integrated services propose two different architectures for real-time traffic – controlled load and guaranteed predictive services. Controlled load framework provides better than best-effort within the given environmental parameters and network conditions. The controlled load guarantees that the throughput achieved by the applications is high while not giving any upper bound on the latency introduced. Controlled load is provided to those flows for which the traffic conforms to the Tspec at the time of flow setup. Traffic that do not conform to the specifications are treated best effort. Guaranteed services, on the other hand provide a framework for delivering traffic for the applications with a bandwidth guarantee and upper bound on the delay.

Integrated services architectures exploit two main avenues where service quality provision is crucial. These include: time of delivery (whether real-time or elastic) and resource sharing mechanism using traffic flows [19]. Each of these is discussed below.

Latency is introduced in the network at each point in the transit along the path to the destination. The amount of latency introduced is cumulative and variable. Queuing delays are highly variable and a real-time application should provide for enough buffering for playback. The receiver buffers or smoothens out the jitter caused by the delay by buffering the received data for a period of time (equal to the offset delay) before playing out.

On the other hand, elastic applications do not require strict delay bound but require guarantee of delivery of packets. Most of the bulk transfer applications may be considered elastic as there is no requirement for them to be delay sensitive. For different elastic applications, it may be possible to differentiate classes of service based on the relative delay sensitiveness of applications. Elastic applications are based on adaptive flow control procedure in which the network parameters such as capacity and receiver capability are taken into consideration. Typical elastic applications are non-delay sensitive and hence error control may be done at a pace with no upper bound unlike real-time applications. Real-time applications are intolerant to

delays and do not wait for packets for late arrivals. Thus, the performance of an application may be measure in terms of two parameters: latency and fidelity [19].

5.2 Integrated Services Control and Characterization Parameters

Control parameters are used by the applications to provide information to the network related to the service quality control requests. Characterization parameters are used to discover or characterize quality of service management environment along the path. These parameters are used by applications requesting QoS or by other network elements along the path. Following are some the parameters and their utility [20]. Reference [20] talks about each of the characterization parameters in detail.

NON_IS_HOP

Provides information about the presence of nodes that do not implement control services along the data path. Also is used to indicate if the node is integrated-services aware. A flag is set if the node does not implement relevant service quality and it reflects a break in the traffic path of nodes that implement a service. For the nodes that are not IS-aware, this object does not exist so the implementation is left to the protocol that is setup or by other means.

NUMBER_OF_IS_HOPS

Number of IS-aware nodes that lie along the data path.

AVAILABLE_PATH_BANDWIDTH

Local parameter that provides an estimate of the bandwidth nodes available for traffic along the path. Values for this range from 1 byte per second to 40 terabytes per second.

MINIMUM_PATH_LATENCY

Latency in the forwarding path associated with the node. Includes propagation delay, packet processing and queuing delays. Cumulative sum of this parameters along the path gives the minimum latency from that node till the destination. Knowing both the minimum latencies allow the receiving application to compute jitter buffer requirements.

PATH_MTU

Maximum Transmission Unit (MTU) for the packets traversing the data path in bytes. The packet is assumed not fragmented.

TOKEN_BUCKET_TSPEC

Describes traffic parameters using a simple token-bucket filter. Specifies the token bucket specification with a peak rate p, minimum policed unit m, token rate r, and a maximum packet size M.

5.3 Integrated Services Resource Reservation and Sharing mechanism

To avoid different flows to unfairly utilize more than their fair share of network resources, integrated services require appropriate allocation of network resources on a flow-by-flow basis. Many flows may share the same link with aggregate traffic shared by various types of real-time and elastic traffic [5].

Several link-sharing mechanisms are introduced by the architecture: multi-entity link sharing, multi-protocol link sharing and multi-service link sharing. A link could be divided between a number of organizations, each of which would divide the resulting allocation among a number of protocols, each of which would be divided among a number of services.

5.4 Delivering Guaranteed Service Quality

Guaranteed service provides a framework for delivering traffic for applications with a bandwidth guarantee and bound on delays. The framework of guaranteed services asserts that the queuing delay is a function of token bucket depth and data rate of the application requests. A guaranteed service sender specifies traffic parameters *Tspec* and *Rspec*. The service uses TOKEN_BUCKET_TSPEC parameter to describe the data flow's traffic characteristics. Traffic characteristics are ensured that it is conforming to required classes or else is considered best effort service. Note that every node in the data path must implement the guaranteed service for this class of service to function [19].

Using the traffic characteristics Tspec and Rspec, the following basic traffic controlling is done in integrated services.

- Packet Scheduling. Scheduling mechanism that is a non-FIFO queuing implementation.

- Packet classifier. Maps each incoming packet to a specific class to be acted upon individually.
- Admission control. Whether request may be granted without affecting other established flows in the network.
- Resource Reservation. Integrated services require a resource reservation protocol like that of RSVP to setup flow state.

6. Conclusion

As we moved from the traditional best effort model to delivering end-to-end guaranteed services, we find an increased complexity in acquiring and managing global resource reservation state from other hosts. As the techniques evolved from best effort model, we were faced with three major issues: Group dynamics, Network dynamics and Traffic Dynamics. Group and Network dynamics deal with various distributed algorithms to coordinate among participating nodes in the service aware networks. As traffic dynamics evolve, newer congestion control and flow control techniques need be applied.

Although Resource Reservation Protocol (RSVP) provides a simple soft state maintenance, it could not handle route instability problem or link breaks. Route changes and failures are understood only after protocol timeout. This is due to the inherent lookup of local host routing table before forwarding reservation requests. These problems may be partially alleviated by defining a newer RSVP forward route protocol between interior routers that take critical decisions along the path; another technique is to build the route request database at each node and make decisions on the criticality of the node's position in making reservation requests by using some weights.

Resource reservation mechanisms are heading towards an end-host transparent model while maintaining consistency among heterogeneous networks. In future, reservations may tend towards maintaining advanced reservation models in which the there is no relation between resource reservation start time and resource usage time. This may be achieved by a global uniform resource manager singly maintaining requests and responses in an autonomous systems.

7. References

[1] P. Almquist, “Type of Service in Internet Protocol Suite”, Internet Draft, RFC 1349, Jul 1992.

[2] D. Awduche, J.Malcom, J.Agogbua, M.O’Dell and J.McManus, “Requirements for Traffic Engineering Over MPLS”, Internet Draft, RFC 2702, Sep 1999.

[3] A. Banerjea, D.Ferrari, B.Mah and M.Moran, "The Tenet real-time protocol suite: Design, implementation, and experiences", Technical Report TR-94-059, Department of Computer Science, University of California at Berkeley, 1994.

[4] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, “An Architecture for Differentiated Services”, Internet Draft, RFC 2475, Dec 1998.

[5] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, Resource Reservation Protocol (RSVP), Internet Draft, RFC 2205, Sep 1997.

[6] R. Braden, D. Clark and S. Shenker, Integrated Services in the Internet Architecture: an Overview, Internet Draft, RFC 1633, Jun 1994.

[7] D. Clark and W. Fang, “Explicit Allocation of Best-Effort Packet Delivery Service”, IEEE/ACM Trans. Networking. Vol 6, No. 4. Aug 1998.

[8] L. Delgrossi and L.Berger, “Internet Stream Protocol Version 2 (ST2)”, Internet Draft, RFC 1819, Aug 1995.

[9] P. Ferguson, G. Huston, Quality of Service: Delivering QoS on the Internet and in Corporate Networks, John Wiley and Sons, 1998.

[10] S. Floyd, and V. Jacobson, “Link Sharing and Resource Management Model for Packet Networks”, Vol. 3, No. 4, IEEE/ACM Transactions on Networking, Aug 1995.

[11] S. Herzog, “RSVP Extensions for Policy Control”, Internet Draft, RFC 2205, Jan 2000.

[12] T.Hudson, M.Weigle, K.Jeffay, R.Taylor, “Experiments in Best-effort multimedia networking for a disturbed virtual environment”, SPIE Proceedings Series, San Jose, Multimedia Computing and Networking, Vol.4312,pp 88-98, Jan 2001.

[13] P. Hurley., J. Le Boudec, P. Thiran and M.Kara, “ABE: Providing a Low-Delay Service within Best Effort,” IEEE Network Magazine, Vol. 15, No. 3, May, 2001.

[14] P. Kim, “LLRM: A Signaling Protocol for Reserving Resources in Bridged Networks”, In Proceedings of OPENSIG `96, Columbia University, New York, Oct 1996.

[15] V. Kumar, T. Lakshman, and D. Stiliadis, "Beyond best effort: Router architectures for the differentiated services of tommorrow's Internet," pp.152--164, Vol. 36, No. 5, IEEE Communications Magazine, May 1998.

[16] D. Mitzel, D. Estrin, S. Shenker, and L. Zhang. “An Architectural Comparison of ST-II and RSVP”, Proceedings of the IEEE INFOCOM '94 Conference on Computer Communications, Toronto, pp 716--725, IEEE Computer Society Press, 1994.

[17] E. Rosen, A. Vishwanathan, R. Callon, Multiprotocol Label Switching (MPLS) Architecture, Internet Draft, RFC 3031, Jan 2001.

[18] H. Schulzrine, S. Casner, R. Frederick, V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications”, Internet Draft, RFC 1889.

[19] S. Shenker, C. Partridge and R. Guerin, “Specification of Guaranteed Quality of Service”, Internet Draft, RFC 2212, Sep 1997.

[20] S. Shenker and J. Wroclawski, General Characterization Parameters for Integrated Service Network Elements, Internet Draft, RFC 2215, Sep 1997.

[21] X.Xiao, L.Ni, “Internet QoS: A Big Picture”, IEEE Network, p 8-18, Vol.13, No. 2, Mar, 1999.

[22] R.Yavatkar, D. Hoffman, Y. Bernet, F. Baker, M. Speer, “SBM (Subnet Bandwidth Manager): A Protocol for RSVP-based Admission Control over IEEE 802-style networks”, Internet Draft, RFC 2814, May 2000.

[23] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, ``RSVP: a new resource reservation protocol'', IEEE Network, pp. 8—18, Vol. 7, no. 5, Sep 1993.

[24] D.L. Mills, “Internet Delay Experiments”, Internet Draft, RFC 889, Dec 1983.

[25] R. Stevens, TCP/IP Illustrated, Vol. 1, Addision-Wesley Publ, 1994

[26] S. Shenker, C.Partridge and R.Guerin, “Specification of Guaranteed Quality of Service”, Internet Draft, RFC 2212, Sep 1997.

[27] J. Wroclawski, “Specification of the Controlled-Load Network Element Service”, Internet Draft, RFC 2211, Sep 1997.