A SURVEY ON ADMISSION-CONTROL SCHEMES AND SCHEDULING ALGORITHMS

Masaru Okuda, Murray State University

Abstract

There has been a sustained interest among researchers and network operators in providing quality of service (QoS) over the Internet. As an essential tool for supporting QoS, development of effective and scalable admission control is an important topic of research. Over the years, various admission-control schemes have been proposed that claim to scale well in a network environment where the network core is kept relatively simple and processing burdens are pushed to the edges of the network. This study surveyed selected admission-control schemes of this type. The contribution of this study is an introduction of new classifications of admission-control schemes, which is based on locations where the key admission-control mechanisms are implemented in a network. The survey of the literature was conducted in light of location-based classification of admission controls and details the workings of schemes, discusses their contributions, and identifies areas of further development.

Introduction

Admission control is a process through which a network node determines whether to accept a new flow request or deny it. It is a traffic management tool through which the load on a network is controlled. The admission decision is made based on several criteria: 1) the current and future availability of network resources, 2) the impact of admission decisions on the existing flows, and 3) the policy control implemented by the network administrator.

Admission control is essential when the network promises service guarantees or levels of service assurances. The goals of the admission control are to protect the performance objectives of the existing flows, deny any requests the network is unable to provide for, and accept as many new flows as the network can commit to.

A. Classical Admission Control

Admission control has been a topic of strong interest among researchers for many years. Research activities were particularly active when ATM standards were emerging. ATM employs a connection-oriented, hop-by-hop admission-control scheme as follows: A call is requested from a user to the network by means of signaling. The signaling message carries a profile of the requested call, referred to as a traffic descriptor, which details the characteristics of the generated traffic such as peak rate and delay requirement. Upon receiving the call request, the network node executes an admission test by examining the traffic descriptor against the current state of the node. If enough resource is available, the node admits the new call and forwards the request to the downstream node. The downstream node, in turn, executes an admission test and decides whether to admit the requested call or not. This process is repeated until the call request reaches the destination.

In order to make a sound admission decision, each node maintains the state of all calls established through the node. This information is updated every time a new call is added or an existing call is terminated. Due to a large amount of state information required at each node, concerns have been raised regarding resource usage efficiency and the scalability of such admission processes.

B. Integrated Services and RSVP

Successful deployment of ATM networks inspired researchers and engineers to build IP networks capable of QoS support, similar to that of ATM. Much of the knowledge and experience gained from ATM has been incorporated into the design of new IP networks. Integrated Services [1] and RSVP [2] are the outcomes of their effort and define the core specifications for QoS-enabled IP networks. The combination of IntServ and RSVP gave hope for the QoS support on IP networks. As with ATM, IntServ architecture aims to provide service guarantees through resource reservation. Through RSVP, it employs end-to-end signalling to communicate QoS parameters for the reservation of resources. However, on this type of architecture, each reservation of resource requires a state to be maintained at every node along the path of an end-to-end flow. It has been said that such architectures may not scale well due to heavy processing overhead and large memory consumption required to maintain those flow states. Considering the rate at which the size of Internet is growing and the number of hosts being added, the concentration of flows within the core routers can be a real issue and the management of individual flow states will become increasingly difficult. A mechanism that simplifies the operations of the network core is desired.

C. Differentiated Services Architecture

To remedy the scalability problem of IntServ with an RSVP approach, differentiated services [3] have been proposed. DiffServ achieves the scalability by relieving the network core from resource-intensive operations and placing the complexity at the edge routers. Specifically, classification and conditioning of packets is performed only at the edges of the network. DiffServ does not employ hop-by-hop signaling in order to avoid the maintenance of per-flow state swithin the core of the network. Instead, flows with similar profiles are aggregated at the edge routers so that the core routers only need to handle bundles of flows.

DiffServ supports class-of-service differentiations. In order to maintain the promised level of service, the amount of traffic accepted at each class, especially at higher levels of classes, must be limited. Otherwise, the Service Level Agreement between the user and the network will be violated [4]. Thus, there is a need for admission control. In order for the edge nodes to make sound admission decisions, they must receive feedback from other parts of the network. DiffServ specification makes no mention of how this is to be done.

D. Multiprotocol Label Switching

Multiprotocol Label Switching (MPLS) [5] is an evolving and expanding set of protocols developed by IETF. MPLS can be seen as a combination of different feature sets from ATM, IntServ, and DiffServ. This is achieved through the creation of a unidirectional signaled path, known as Label Switched Path. Label Switched Path is established by RSVP-based call control, known as RSVP-TE. MPLS aims to provide QoS-enabled transmission paths over the Internet. MPLS employs encapsulation of packets with short descriptors known as labels at the entry nodes of the MPLS network. The label determines which QoS class the packet belongs to and where it will be forwarded to. The same label will be placed on all packets that belong to the same QoS class and the same forwarding destination.

MPLS is gaining wide acceptance as a WAN protocol-of-choice and replacing Frame Relay and ATM-based WANs. It is used to transport Voice Over IP (VOIP) traffic and extend Ethernet LANs over the Internet. The strengths of MPLS include the seamless support of IP packets with QoS support, ability to operate through segments of network that do not support MPLS, and scalability afforded by the implementation of labels and simple operations at the core of the network. MPLS is protocol agonistic in that the payload of the labeled packets may be of any type, such as Ethernet frames or ATM cells. MPLS is designed specifically for those protocols that do not support QoS natively, such as IP. MPLS allows multiple layers of label encapsulations to allow tunneling through different administrative MPLS domains. Because MPLS uses RSVP-based call control, it inherits the same strengths and weaknesses of RSVP.

There is a need for admission control that scales well in an environment where core routers are kept relatively simple and processing burdens are pushed to the edges of the network. This study surveyed admission-control schemes recently proposed, all of which claim to offer some level of scalability. The remaining sections of this paper are organized as follows: Section 2 classifies the admission-control schemes and scheduling algorithms in several categories. Section 3 surveys the admission-control schemes being proposed in recent years and describes the goals, approaches, contributions, and shortcomings of each scheme; Section 4 concludes the survey.

Classifications of Admission Control and Scheduling Algorithms

This section describes the classifications of admission-control schemes and scheduling algorithms.

A. Parameter-Based vs. Measurement-Based Admission Control

Admission-control schemes are generally classified as either parameter-based or measurement-based approaches. In either case, users request service from the network by sending flow specifications. Flow specifications describe the nature of packet flows (e.g. peak rate) and requirements for packet handling within the network (e.g. loss rate). The network uses parameters specified in the flow specifications to compute how much resource it must set aside in order to support the requested flow. The admission decision will be made by comparing the required resource against what is available on a node.

The differences between the two approaches exist in the way the allocated resource on a node is being estimated. In parameter-based admission control, the node computes its reserved resource by keeping track of parameter values in flow specifications at each flow establishment and termination. With this approach, the amount of allocated resource is a discrete function and the network node knows exactly how much resource is used or reserved at any given time. The strength of this approach lies in its ability to provide hard guarantees to each flow being accepted. One of the shortcomings is that it does not use the resource efficiently. The worst-case scenario is typically used to compute the resource reservation requirements to assure hard guarantees. Once the resource is marked as reserved, it is no longer available for new flows that request guaranteed service.

Under the measurement-based approach, the resource consumed by existing flows on a network is estimated by measuring the actual traffic flow. It applies statistical principles to assess the current and very-near-future state of the network. Expressed by way of confidence level, it can predict a likelihood of being able to support a requested level of service based on the traffic pattern of the past. Using this information, a network node decides whether to admit a flow or reject it. This approach is shown to have much better utilization of network resources than the parameter-based one. However, measurement-based admission control does not provide hard guarantees.

The level of assurance this approach gives is based on the past history; the applicability of confidence level depends on whether the traffic pattern will remain similar to that of the past. Since the network is not immune to sudden changes in its environment (e.g. traffic pattern changes, link failures), the measurement-based approach may be effective only on stable networks. Another shortcoming of this approach is that it requires an accumulation of a long history. In order to yield a high utilization, the confidence interval at a given confidence level must be kept short. This requires many samples. Without a long history, admission decisions must be made with a very conservative view of the unused resources.

In recent years, admission-control schemes that are hybrid between parameter-based and measurement-based approaches have been investigated [4], [6], [12], [14]. They incorporate past history (i.e., measurements) to adjust the reserved bandwidth (i.e., parameters) of flows. Due to their duality, the strengths of either approach may mitigate the weaknesses of the other. Because of this unique property, a hybrid approach to admission control is gaining interest.

B. Stateful vs. Stateless Scheduling Algorithms

The manner in which the arriving packets are queued and processed at each network node, referred to as scheduling, can have a significant impact on the way the admission control is carried out. Scheduling algorithms are generally classified as stateful or stateless for the purpose of scalability discussion. Stateful algorithms require maintenance of individual flow state at every node along the path of a flow. Examples of stateful-scheduling algorithms include Fair Queueing [7], [8], Virtual Clock [9], and their variants such as Weighted Fair Queueing [8] and Jitter-Virtual Clock [4]. These algorithms have been developed for the support of guaranteed service as their primary objective. They give precise control over the treatment of individual flow and can provide bounds on bandwidth allocation and end-to-end delay. The major drawback of stateful schedulers is that they require maintenance of per flow QoS state of all flows at each network node. Due to the size of and the complexity involved in the management of QoS state information, the scalability of this approach has been challenged. When a stateful scheduler is deployed in a network, admission control makes use of individual QoS state maintained at each node and determines whether the node has sufficient resources to meet the demands of the newly-requested flow.

Stateless-scheduling algorithms, on the other hand, maintain no QoS state at any part of the network. FIFO and LIFO queueing are examples of stateless algorithms. Since it requires no state maintenance, it is scalable. However, it does not provide the control necessary to support various QoS requirements. The Internet, for the most part, is composed of network nodes supporting stateless-queueing algorithms.

In recent years, a new type of scheduler has been added to the above. It is called core-stateless scheduling. Core-stateless scheduling aims to provide a similar level of QoS control offered by stateful algorithms, yet tries to achieve network scalability comparable to one offered by stateless algorithms. In core-stateless algorithms, the edge nodes maintain QoS states of individual flows, but the core routers do not. The core routers may maintain aggregate-level information that assists in controlling flows, depending on the implementation. The elimination of individual flow states from the core routers is made possible by embedding the QoS states in each packet header. There have been some novel ideas proposed using this scheduling mechanism. Core-Stateless Fair Queueing [10], Core-Jitter VC [4], and Virtual Time Reference System [11] are examples of core-stateless algorithms. They are further explained later in the survey section of this paper.

C. Location Based Classification of Admission Control

A contribution of this study is the introduction of location-based classification of admission-control schemes. It is a new classification based on locations at which the key admission-control algorithms are applied. According to location-based classification, admission-control algorithms proposed in recent years are classified into the following five categories: admission control at 1) edge nodes (Edges), 2) central node (Central), 3) ingress node (Ingress), 4) egress node (Egress), and 5) end-user station (End-to-End). The taxonomy of admission-control schemes is given in Figure 1.

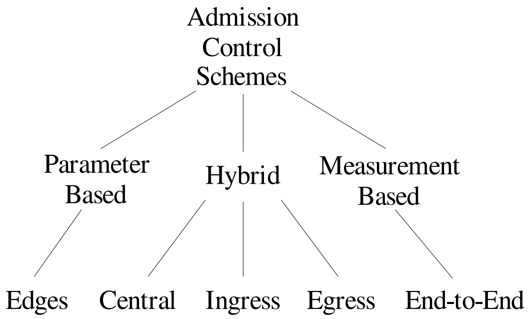


Figure 1. Taxonomy of admission-control schemes

Admission control at edge nodes (i.e., Edges) lessons the processing requirements of core routers through flow aggregation at network edges. Core routers process and maintain only the aggregate flow reservation information. Through aggregation, overhead reduction is made possible by fewer signaling-message exchanges and less call-state maintenance. Aggregation of RSVP [13] belongs to this category.

Admission control at central node (i.e., Central) employs a master server that performs admission-control functions on behalf of all routers in a network. By off-loading resource-intensive services, core routers become lightweight. Bandwidth Broker [12] uses this approach.

Admission control at ingress node (i.e., Ingriss) enables core routers to make admission decisions without needing to maintain individual flow states. Ingress node measures the rate of packet arrivals for each individual flow and inserts this information in each packet header. Core routers read this information and accumulate them per aggregate flow. Thus, the core routers only maintain aggregate flow states and are able to make admission decisions at an individual flow basis. Dynamic Packet Sate (DPS) [4] uses this approach.

Admission control at egress node (i.e., Egress) pushes the complexity to the egress routers so that no per-flow states need to be maintained in the core of the network. Egress routers construct profiles of flows by monitoring the packet arrivals and departures. By measuring delay experienced by each packet, egress routers estimate the dynamically changing network load. Based on this information, the egress routers make admission decisions. Egress admission control [14] belongs to this category.

Admission control at the end-user station (i.e., End-to-End) uses a form of in-band signaling to estimate the availability of network resources. The admission decision is typically made by end users, rather than the network. Prior to sending data traffic, an originating end user sends a stream of packets at a constant rate for a short period of time. The receiver measures the arrival pattern of probing packets and returns the summary statistics. Upon receiving the summary information, the sender decides whether the network is capable of carrying the requested load. Scalable Reservation Protocol [15] and others [16]–[18] belong to this category.

Survey of Admission-Control Schemes

In light of location-based classifications of admission-control schemes described above, this section surveys selected admission-control schemes.

A. Admission Control at Ingress Node

Dynamic Packet State (DPS) [4] is an ingress-node based admission-control scheme and employees a core-stateless scheduler. Its goal is to make admission decisions for new flows without maintaining individual flow states in the core of the network. DPS also aims to achieve end-to-end per-flow delay and bandwidth guarantees on a network, where only the edge routers perform per-flow management. To meet these goals, DPS uses a packet-header marking technique, where the ingress node encodes state information on the header of each packet. The core nodes apply control to packets according to their header markings. DPS proposes two innovative schemes, one in admission control and the other in scheduling. How these schemes work is described in subsequent sections.

DPS's admission-control scheme is comprised of two algorithms: 1) per-hop admission control and 2) aggregate reservation estimation. The former is parameter-based, while the latter is measurement-based. Each algorithm independently computes an estimated reserved bandwidth of aggregated flows. These estimates should be very close, if not the same. However, under certain conditions, deviations from the true reserved bandwidth are observed on each of the two algorithms in opposite directions. One algorithm estimates at a higher rate than the true reserved bandwidth and the other estimates at a lower rate. The first algorithm does not account for the duplicate reservation requests. This can lead to an under-utilization of a link due to inflated estimation of the reserved bandwidth.

The second algorithm does not include the effects of new calls being admitted in the middle of an estimate cycle. This results in estimating the reserved bandwidth at a lower rate than the actual rate. The results from these two algorithms are reconciled at the end of a fixed interval and arrive at one value that better reflects the true reserved bandwidth. The goal of admission control in DPS is to estimate a close upper bound on a reserved aggregate rate so that a deterministic guarantee can be made to those calls being accepted, while minimizing over-reservation. DPS proposes a scheduling algorithm that provides service guarantees at levels comparable to IntServ on DiffServ-like environments. This scheduling algorithm is called Core-Jitter Virtual Clock (Core-Jitter VC). It is a non-work conserving scheduling algorithm.

Core-Jitter Virtual Clock is a variant of Jitter Virtual Clock (Jitter VC). The primary difference between the two algorithms is that Core-Jitter VC is a core-stateless-based scheduler, while Jitter VC is a stateful scheduler. Core-Jitter VC provides the same delay guarantee as Jitter VC at an end-to-end path, but not at intermediate routers. Jitter VC has been proven to provide the same level of guarantee as Weighted Fair Queueing (WFQ) [19]. Thus, Core-Jitter VC also provides the same guarantee as WFQ at the end-to-end path.

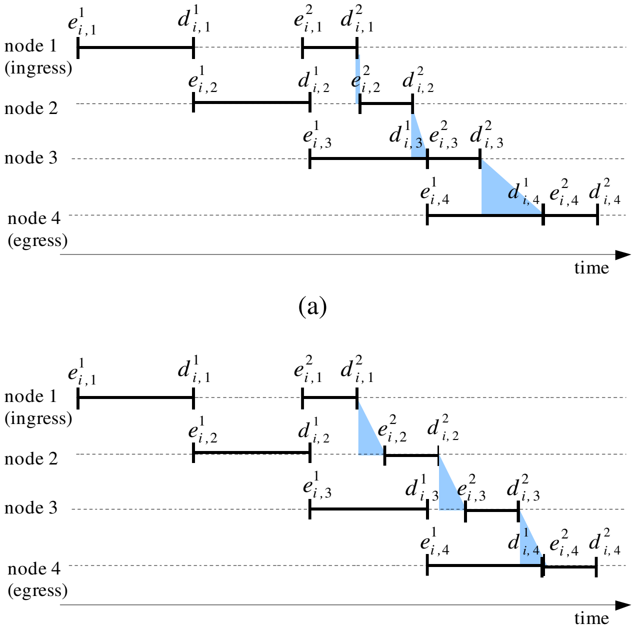
Jitter VC and Core-Jitter VC, are based on a packet-header marking and queueing architecture, where each router in a path of a flow reads and re-marks packet-header information for queueing and scheduling purposes. They employ a delay-jitter-rate-controller unit [20] for queueing purposes and a Virtual-Clock scheduler for scheduling purposes. A packet entering into a Jitter-VC router or a Core-Jitter-VC router will be held in a waiting room by the delay-jitter-rate-controller until it becomes eligible for transmission. Once the packet is released from the waiting room, Virtual-Clock scheduler services them in order of their earliest deadline. Each packet is given a deadline by which it must leave the Jitter-VC server or the Core-Jitter-VC server.

In order to better explain the workings of Core-Jitter VC, Jitter VC is described first. For the k^{th} packet of flow *i*, its eligible time $e^k_{i,j}$ and deadline $d^k_{i,j}$ at the j^{th} node on its path under the Jitter-VC algorithm are computed as follows:

$$e^1_{i,j} = a^1_{i,j} \quad e^k_{i,j} = \max(a^k_{i,j} + g^k_{i,j-1}, d^{k-1}_{i,j}) \quad i, j \geq 1,\ k > 1 \tag{4}$$

$$d^k_{i,j} = e^k_{i,j} + \frac{l^k_i}{r_i}, \quad i, j, k \geq 1 \tag{5}$$

where $a^k_{i,j}$ is the arrival time, l^k_i is the packet length, and $g^k_{i,j-1}$ is the amount of time between the packet's deadline and the actual departure time. At every packet departure at every router, this g value is computed and recorded in its packet header and read at the subsequent router. A sample time diagram of packets going through a series of Jitter-VC servers is depicted in Figure 2(a). The shaded area depicts delays experienced by the second packet at each node.



(a)

(b)

Figure 2. The time diagram of packets through (a) Jitter-VC servers and (b) Core-Jitter-VC servers

Jitter VC is a stateful service because, by equation (4), each router must maintain the deadline, $d^{k-1}_{i,j}$, of a previously received packet when it computes the eligibility time, $e^k_{i,j}$, of an arriving packet from the same flow.

Core-Jitter VC improves upon Jitter VC and makes the scheme stateless. It does so by removing the term $d^{k-1}_{i,j}$ from the equation (4) and introducing a new term, δ^k_i, a slack variable, instead, which holds the following property:

$$d^k_{i,j} + g^k_{i,j-1} + \delta^k_i \geq g^{k-1}_{i,j},\ j > 1 \tag{6}$$

With the above definition, the eligibility time of a packet at the j^{th} node can be computed as follows—compare it to equation (4):

$$e^k_{i,j} = a^k_{i,j} + g^{k-1}_{i,j-1} + \delta^k_i,\ \ j > 1 \tag{7}$$

The details of how the actual value for the slack variable δ^k_i is determined are given by Stoica and Zhang [4]. A sample time diagram of packets going through a series of Core-Jitter-VC servers is depicted in Figure 2(b). Observe that the slack time, δ, is a fixed value for all participating nodes in flow *i* for the k^{th} packet.

The strengths of DPS include its ability to guarantee bandwidth and delay bound-through Core-Jitter VC. The proposed admission-control algorithm is robust in the presence of network failures and partial reservation since the algorithm to estimate the reservation rate does not remember the past beyond the period, T_W . DPS's largest contribution is that it is the first of its kind to demonstrate that there is a way to provide a hard guarantee on bandwidth and delay requirements without maintaining individual flow states in the core of the network. DPS proposed a noble idea of inserting individual flow states in the header of packets so that the core nodes don't have to maintain them. DPS inspired others in developing new schemes based on this premise [11], [21].

While DPS offers guaranteed services without individual flow-state maintenance at core routers, the overall scalability gained from this architecture remains a question. Insertion and interpretation of state information in every data packet can be an expensive operation. Indeed, there is a concern that DPS may be transforming all data packets into control packets such that core nodes must pay extra attention to every packet they receive regardless of its type. Core-Jitter Virtual Clock scheduler requires both ingress and core nodes to monitor and alter the header of every data packet that travels through. For admission control, only the ingress router writes to the packet header, yet core routers must read and process every data packet. Since the control information is embedded in the data packet headers, all packets become essential to the healthy operation of the network. Considering the parameter-based QoS model, where only the control packets need extra attention from the routers, DPS's new approach could potentially add higher processing demands on network routers.

Another drawback of DPS's admission control is that it requires insertion of dummy packets at the ingress router when there is no data flow. Dummy packets must be injected in the network every time there is a gap between data packets, which is larger than the maximum inter-packet arrival time T_I. T_I is typically a small window compared to the period T_W used to compute the aggregate reserved rate. This type of approach works well for those applications that generate traffic at a constant bit rate and always terminate the reservation as soon as the transmission is over, such as telephony. DPS's admission-control scheme may not appeal strongly to other types of network applications. If the source is silent for an extended period of time, constant bit-rate dummy packets must be inserted into a network at $1 / T_I$ rate. This could result in wasted bandwidth because even the best-effort traffic cannot take full advantage of unused bandwidth.

B. Admission Control at Central Node

Bandwidth Broker (BB) [12] belongs to the centrally-controlled admission-control approach that aims to provide scalability in the network by off-loading the routers' control-plane functionalities to a master server known as a Bandwidth Broker. Bandwidth Broker maintains QoS-state information for all flows of every router within a designated domain. Network routers perform only the data-plane functionalities (i.e., packet forwarding), in addition to the exchange of QoS-related information of each flow with Bandwidth Broker.

Bandwidth-Broker architecture is built upon the Virtual Time Reference System (VTRS) [11]. It is classified as a core-stateless scheduling scheme, where the core routers in the network do not maintain individual flow states. VTRS is a framework on which guaranteed services can be offered in a network without mandating that a specific scheduling algorithm (e.g., Core-Jitter VC) be employed. It consists of three logical components: a packet state carried by packets, edge-traffic conditioning at the network edge, and a per-hop virtual-time reference and update mechanism at the core routers.

VTRS was inspired by the work presented in Dynamic Packet State (DPS) [4], where the core-stateless approach was first introduced. VTRS is an extension to DPS; however, VTRS has unique and significant contributions beyond what DPS proposed. First, it established generalized mathematical expressions that bound end-to-end delay and bandwidth requirement for the support of flows that travel through core-stateless routers. Second, framework defined in VTRS is generic enough that it not only expresses delay bounds and sustainable rates of a flow through core-stateless schedulers, but also through stateful schedulers (e.g., WFQ) as well as stateless (e.g., FIFO) schedulers. Third, the framework allows mixing of rate-based and delay-based schedulers in the path of a flow. Fourth, it introduced two new work-conserving core-stateless scheduling algorithms: Core Stateless Virtual Clock (CSVC), which is rate-based and Virtual Time Earliest Deadline First (VT-EDF), which is delay-based.

Consider the path of a flow j on a network, traversing h hops of routers. Suppose that q routers execute rate-based scheduling and $h - q$ routers employ delay-based schedulers. Packets entering the network will be shaped at the edge node and move through a series of core nodes. Delays that the packets experience will be at the shaper and at each core node. Then, the total delay of end-to-end path of flow j, d^j_{e2e} , is

$$d^j_{e2e} = d^j_{shaper} + d^j_{core} \tag{8}$$

For simplicity, we do not consider the delay experienced at the shaper in this paper. It suffices to say that d^{j}_{shaper} varies by the type of shaper being used and the maximum delay that can be bounded. Zhang gives an example of delay experienced at the shaper using a dual token-bucket regulator [12].

The delay experienced at core nodes is bounded by

$$d^{j}_{e2e} = q\frac{L^{j,\max}}{r^{j}} + (h-1q)d^{j} + \sum\nolimits_{i=1}^{h-1}\pi_i + \sum\nolimits_{i=1}^{h}\psi_i \qquad (9)$$

The term $q\dfrac{L^{j,\max}}{r^{j}}$ represents the delay experienced at rate-based routers and $(h-1q)d^{j}$ represents the delay observed at delay-based routers. $\sum\nolimits_{i=1}^{h-1}\pi_i$ is the total propagation delay and ψ_i is the error term of node i , which has the following property:

$$\hat{f}_i^{\,j,k} \le \hat{v}_i^{j,k} + \psi_i \qquad (10)$$

where $\hat{f}_i^{\,j,k}$ is the virtual finish time of packet *k* in flow *j* at node *i*. It means that the targeted packet-departure time in a virtual time line is the latest time the packet may leave the node and still meet the delay requirement. $\hat{v}_i^{j,k}$ is the actual finish time (i.e., actual packet departure time) of packet *k* in flow *j* at node *i*.

Having achieved the bandwidth and delay guarantees through VTRS on core-stateless network, the designers of VTRS enhanced its scalability further by moving the QoS-related control functions out of core routers to a master server known as a Bandwidth Broker. Bandwidth Broker is composed of three service components: policy control, QoS routing, and admission control. Policy control determines which hosts and applications are allowed to access the network. QoS routing selects a path that fulfills the requirements of a requested flow. Admission control determines the eligibility and feasibility of the requested flow by consulting the policy control and QoS routing control.

Bandwidth Broker (BB) suggests that by moving the admission-control function from the core routers to a central server, several positive outcomes can be expected. First, it further alleviates the core routers from burdensome processing and making them potentially more efficient. Second, service guarantees can be made for both per-flow and aggregate flows. Third, by decoupling the QoS-related functionalities of control plane from core routers, it may be possible to introduce new guaranteed services without requiring software or hardware upgrades at core routers. Fourth, it allows the execution of sophisticated and optimized admission control for the entire network, which might have been difficult under the hop-by-hop admission control. Fifth, the problem of inconsistent QoS states observed in the hop-by-hop reservation mechanism can be lessened. Sixth, through the physical separation of control- and data-plane functionalities, issues in control plane (e.g., scalability of Bandwidth Broker) can be dealt separately from the issues in data plane. Seventh, admission control can be performed at an entire path level, as opposed to a local level as done by the hop-by-hop approach, and could reduce the complexity of admission-control algorithms. Finally, BB addresses the effects of dynamic join and leave of individual flows to and from an aggregate flow and incorporates such effects into the admission-control algorithm.

There are several open issues with the design of Bandwidth Broker. While it addresses the core routers' scalability issues well, it does not elaborate much on the Bandwidth Broker's scalability issues. The amount of flow state information the Bandwidth Broker must manage could increase dramatically as the size of the network grows. There is a mention [12] that this problem can be alleviated by employing multiple Bandwidth Brokers in distributed fashion. This is contrary to one of the original motives of BB, where it tries to avoid the problem of inconsistent network view, which is often introduced by the distributed approach.

There also may be a potential delay incurred when the concentration of communications to and from the Bandwidth Broker becomes severe. Though it is convenient to have policy and QoS-routing information on-hand for admission decisions, performing all three tasks for the entire network can be demanding and it warrants a careful feasibility study. Finally, there is always a danger of a single point of failure, which results not only in an inability to make admission decisions, but also in loss of all QoS-related control-plane functionalities, which Bandwidth Broker provides.

C. Admission Control at Edge Nodes

Aggregation of RSVP reservations [13] belongs to the edges-based admission-control approach and it aims to provide scalability in the network core by aggregating reservation requests of individual flows at the edge nodes. Individual flows between the same pair of source and destination nodes can form an aggregate. It is an extension to RSVP specifications. The primary focus of this approach is on the reduction of RSVP message exchanges, which leads to conservation of memory and processing power at those locations where the volume of individual flows may be heaviest.

The scheme employs two techniques to achieve its goals: suppression and aggregation of reservation messages. Individual-flow RSVP requests are suppressed at the ingress node by altering their protocol ID. The subsequent nodes in the routing path will not see these packets as reservation messages, except at the egress node. When the packets reach the egress node, they will be restored to their original IDs. As the egress node alters the protocol ID of reservation packets for individual flow, it computes the total bandwidth requested from each flow. Once the requests reach a certain total bandwidth, the ingress node initiates an aggregation and sends an AGGREGATE PATH message to downstream nodes. Upon receiving this message, the egress node returns an AGGREGATE RESERVE message and nodes in the path commit the reserved resource for the aggregate flow. Each node in the downstream path marks an appropriate amount of resources for reservation.

RSVP Reservation Aggregation is a logical and natural extension to the existing RSVP. The main contribution is that RSVP will be able to signal AGGREGATE PATH and RESERVE messages and that the core routers need not maintain per-flow states any longer.

Anticipated resource savings can be large when the number of aggregated flows is substantially fewer than the individual flows. It will work well in an environment where there are many end stations that are networked together with a few edge routers, such as VPN. On the other hand, when the scheme is applied to a network, where the number of edge routers is large and the distribution of flows is evenly spread among all edges, resource savings may not be as large as the previously-described environment due to lack of concentrations. Service-provider networks typically belong to the latter type. Furthermore, when the number of aggregate flows increases to a substantial volume, they face similar problems to having many individual flows.

The scheme presented in RFC 3175 allows only those individual flows with the same source and destination pair to form an RSVP aggregate. This is different from DiffServ aggregation, where any flow can form an aggregate, regardless of addresses, so long as they are marked with the same DS Code Point.

RFC 3175 points out that frequent modifications to the bandwidth reservation of aggregate flows due to additions and terminations of individual flow can lead to a large number of reservation updates. This is contrary to the base assumption that fewer reservation messages are generated when individual flow requests are aggregated. On the other hand, infrequent updates to the reserved bandwidth of an aggregate flow can result in wasted bandwidth, since a large block of resources will need to be reserved to absorb temporal bandwidth fluctuations. Thus, there is a trade-off between scalability of the scheme and efficient use of bandwidth.

D. Admission Control at Egress Node

Egress Admission Control [14] performs data collection and admission decisions at the egress router. It processes reservation messages only at the network edge (egress router) and uses continual passive monitoring of a path to assess its available resources. It models the network as a black-box system, where a flow of packets arrives at one end of the box (ingress node), goes through the box (core nodes), and comes out at the other side of the box (egress node). All other flows on the network are modeled as interfering cross-traffic of the measured flow. Using this block-box model, Egress Admission Control aims to develop envelopes that accurately characterize the upper bounds on arrival and service processes through measurement at the egress node. A unique characteristic of these envelopes is that they implicitly include the effects of cross traffic that are not directly measured at the egress point and implicitly prevent other egress points from admitting flows beyond an acceptable range. By applying the extreme theory [22] to the measured envelopes, it estimates the end-to-end service availability of a certain traffic class. This estimate is used for making admission decisions.

Egress Admission Control constructs envelopes for arrival process and service process. All edge nodes are synchronized with each other using Network Time Protocol [23] and they time stamp every packet entering the network. When packets reach the egress node, the time stamp in the packet header is read and an arrival envelope, known as peak rate envelope [24], which captures the behavior of the peak rate of the arrival process, is constructed. The peak rate envelope is constantly updated at a short fixed interval. At a longer time scale, changes in the envelope are measured, expressed as variance, and used to compute the confidence interval of the peak-rate envelope.

The service envelope describes the behavior of the worst rate of service process. When packets arrive at an egress node, it examines each packet's header and computes the delay it experienced. Using this information, the egress node constructs the trace of maximum time required to service a certain number of bits, called minimum service envelope. The variance observed by the changes in the service envelop in a longer time scale is used to compute the confidence interval.

When a new flow is requested, using its declared peak rate and delay bound, adding it to the measured peak rate arrival envelope, the admission test will compare this value against the measured service envelop, taking into account variances, and determine if statistically enough bandwidth exists through the network.

Consider a black-box system that has a measured peak arrival envelope with mean $\bar{R}(t)$ and variance $\sigma^2(t)$. Assume it has a minimum service envelope with mean $\bar{S}(t)$ and variance $\psi^2(t)$. Suppose a new flow request arrives with the peak-rate envelope *r(t)*. Then, through the extreme theory [14], $\bar{R}(t)$ and $\bar{S}(t)$ are Gumbel distributed and the flow can be admitted with delay bound *D* at confidence level of *Φ(α)*, if

$$t\bar{R}(t)+tr(t)-\bar{S}(t+D)+a\sqrt{t^2a^2(t)+\psi^2(t+D)}<0$$
$$0\le t\le T \qquad (17)$$

$$\lim_{t\to\infty}\bar{R}(t)+r(t)\le\lim_{t\to\infty}\frac{S(t)}{t} \qquad (18)$$

Egress Admission Control has several noteworthy properties. First, since it employs a measurement-based algorithm, there is a potential for an efficient use of network bandwidth. Second, it does not require core nodes to process resource reservation messages or store any information associated with flows. Third, it does not assume or require any specific scheduling mechanism in the network and that multiple queueing disciplines can co-exist. Fourth, route pining, a key ingredient for deterministic service, is not fundamentally required. Fifth, egress routers can perform admission control on traffic aggregates and do not need to store or monitor per-flow traffic conditions.

While the approach is novel and elegant, the scheme is vulnerable to sudden traffic-pattern changes. Since the technique used to make admission decisions is based on statistical inference through measurements, the scheme will work best in an environment where the network is stable, the pattern of traffic is relatively unchanging, the amount of traffic added or subtracted at each flow admission or termination is much smaller than the overall traffic being carried, and the size of the network is large. On the other hand, if this scheme is applied to a network composed of few nodes with hap-hazard traffic patterns, it can result in an unpredictable and unacceptable outcome. Since the accuracy of this scheme is closely tied to the network condition, it would be difficult to establish contractual agreements between the user and the service provider. Furthermore, the scheme will not work on the very first flow on any given pair of edge nodes because it does not have past history to construct envelopes for admission tests. It will not hold up well when there are sudden changes in the traffic flow such as node and link failures. A lengthy convergence period may be observed after significant changes in the state of the network occur.

Since the scheme does not explicitly de-allocate the resources at flow termination, it is difficult for the network to distinguish whether a flow has been terminated or the source of a flow is being silent temporarily. Once a source becomes silent or sends traffic below the declared sustained rate for a period of time, it may need to re-initiate a flow request or send some type of control packet to restore its state. In order to do this, the source must maintain a timer and the timer must be set with some understanding of the behavior of the network. This could add further complexity not only on the network nodes, but also on the end systems. The scheme does not provide any graceful or intelligent way to drop packets when the load exceeds the anticipated limit. No provisioning for correcting the initial assessment of traffic in an explicit manner is given either.

It also implicitly assumes that the traffic sources always generate some packets for the duration of the reservation. If a flow is admitted at a certain peak rate but is silent for a long time, the scheme will admit other flows during the silent period, resulting in overbooking, and packet drops could be observed when the silent source restarts the traffic generation.

E. Admission Control at End-User Stations

There are several versions of end-to-end measurement-based schemes proposed thus far [15]–[18]. In this section, a system proposed by Karlsson and Ronngren [18] is reviewed.

The goal of this end-to-end measurement-admission scheme is to bound the loss probability of packets in high-priority flows. A host wishing to establish a low packet-loss flow probes the network prior to sending data. Information gathered through probing is used to make an admission decision at the source host. Probing is performed as follows: a source host transmits blocks of packets for a period of time at the peak rate of the flow it wishes to establish. Each packet contains information regarding the probing, such as probe duration and transmission rate. Upon expiration of the probe duration, the destination host returns a packet, which contains a measurement report such as the number of probing packets received. Based on the measurement report, the source host makes the admission decision.

The proposed service architecture employs simple queueing and scheduling mechanisms at each node. Data and probing packets belong to the controlled-load service and are allocated a certain portion of the link capacity. Within the controlled-load service, there are two partitions: high-priority queue and low-priority queue. Data packets are queued at the high-priority queue and always serviced prior to the low-priority queue packets. Probing packets are queued at the low-priority queue. All remaining packets belong to the best-effort traffic and are queued at the best-effort queue. This queue is serviced only when there are no packets in the controlled-load service.

The end-to-end measurement-based approach is by far the simplest of all the admission-control schemes surveyed in this study. The processing required by end system for the probing is light. The queueing and scheduling mechanisms necessary at each node are straightforward, and no flow state needs to be maintained in the network.

Due to its simplicity, the scheme is unable to provide sophisticated services. The proposed scheme can only give a statistical bound on the packet loss; delay is not considered by the admission control as it offers no guarantee since the source makes no requests to the network and the network makes no reservations for the probed flow. Bandwidth blocking could result in a highly contentious environment, where probing packets are generated at a high bandwidth rate compared to the remaining bandwidth of controlled-load service. Suppose there are multiple hosts wishing to establish sessions. Some hosts may request flows at a higher rate than the remaining bandwidth of controlled-load service, while others may wish to establish flows at a lower rate. Obviously, the attempts to establish flows at higher rates than the available bandwidth will not succeed. The slower rate flows should be accepted, so long as enough bandwidth remains in the controlled-load service. Under this type of condition, even the slower rate probing packets can be affected due to congestions in the controlled-load service and may not be able to receive the requested service.

In terms of bandwidth-use efficiency, it is desirable to keep the probing period as short as possible. However, a short probing period may not capture the average state of the network and may result in overbooking or under utilization. There is also an uncertainty in the probability of packet loss if this scheme were applied on a network without a route pinning, yet there was no mention of it in the literature reviewed.

Conclusion

There has been a sustained interest and effort in designing mechanisms to offer guaranteed services on IP networks. Admission control is one of the essential tools in supporting QoS. MPLS has received much attention as a promising transport architecture that could offer service differentiations in large-scale networks. The development of effective and scalable admission-control schemes and accompanying scheduling algorithms suitable for MPLS networks has become an important topic of research.

In this paper, the author introduced a location-based classification of admission-control schemes, a new taxonomy for admission-control schemes that compliments the traditional parameter-based and measurement-based admission-control categorization. In this new classification system, parameter-based and measurement-based admission-control schemes are further categorized into 1) edges-based, 2) centrally-controlled, 3) ingress-based, 4) egress-based, and 5) end-to-end-based. The author also surveyed various admission-control schemes in light of location-based classification systems in order to better understand their suitability for MPLS networks. The analysis summary is given in Table 1.

Table 1. Summary of analysis of admission-control schemes based on location classifications

Location	Admission	Scheduling	Pro / Con
Edges	Parameter	Stateful	Guarantee / Limited in scope
Central	Hybrid	Stateful, Stateless, Core-Stateless	Flexible scheduling / Single point of failure
Ingress	Hybrid	Core-Stateless	Guarantee / May not scale
Egress	Hybrid	Any	Mathematically modeled / Not proven to work
End-to-End	Measure-ment	Any	Simple / No guarantee

Each scheme asserts some level of scalability. Indeed, there are some novel ideas proposed and elegant approaches presented. Yet, every one of them has at least one significant shortcoming that prevents it from being deployed over the Internet.

Admission controls that are hybrid, whose control mechanisms are placed at an ingress node or central node, have characteristics that are more promising for future development than others. In subsequent studies, the author plans to design an admission-control scheme that builds upon those foundations, yet exceeds in its scalability when compared with those evaluated in this study.

References

[1] R. Braden, D. Clark, and S. Shenker, "RFC 1633: Integrated services in the Internet architecture: an overview," Jun. 1994.

[2] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, and S. Jamin, "RFC 2205: Resource ReSerVation Protocol (RSVP) — version 1 functional specification," Sep. 1997.

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "RFC 2475: An architecture for differentiated services," Dec. 1998.

[4] I. Stoica and H. Zhang, "Providing guaranteed services without per flow management," in *SIGCOMM*, 1999, pp. 81–94.

[5] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031 (Proposed Standard), Internet Engineering Task Force, Jan. 2001.

[6] J. Milbrandt, M. Menth, and J. Junker, "Improving experience-based admission control through traffic type awareness," *Journal of Networks*, vol. 2, no. 2, pp. 11–22, April 2007.

[7] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, pp. 344–357, June 1993.

[8] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," in *Proc. of ACM SIGCOMM '89*, 1989, pp. 3–12.

[9] L. Zhang, "Virtual clock: A new traffic control algorithm for packet switching networks," in *Proc. of SIGCOMM '90*, 1990, pp. 19–29.

[10] I. Stoica, S. Shenker, and H. Zhang, "Core-stateless fair queueing: Achieving approximately fair bandwidth allocations in high speed networks," in *SIGCOMM*, 1998, pp. 118–130.

[11] Z. Zhang, Z. Duan, and Y. Hou, "Virtual time reference system: A unifying scheduling framework for scalable support of guaranteed services," 2000.

[12] Z.-L. Zhang, "Decoupling qos control from core routers: A novel bandwidth broker architecture for scalable support of guaranteed services," in *SIGCOMM*, 2000, pp. 71–83.

[13] F. Baker, C. Iturralde, F. L. Faucheur, and B. Davie, "RFC 3175: Aggregation of rsvp for ipv4 and ipv6 reservations," Dec. 2001.

[14] C. Cetinkaya, V. Kanodia, and E. Knightly, "Scalable services via egress admission control," *IEEE Transaction on Multimedia*, vol. 3, no. 1, March 2001.

[15] W. Almesberger, T. Ferrari, and J. Y, "Srp: a scalable resource reservation protocol for the internet," 1998.

[16] F. Kelly, P. Key, and S. Zachary, "Distributed admission control," December 2000.

[17] G. Bianchi, F. Borgonovo, A. Capone, L. Fratta, and C. Petrioli, "Pcp-dv: An end-to-end admission control mechanism for ip telephony," in *Tyrrhenian IWDC 2001 Evolutionary Trends of the Internet*, Taormina, Italy, September 2001.

[18] V. Elek, G. Karlsson, and R. Ronngren, "Admission control based on end-to-end measurements," in *INFOCOM (2)*, 2000, pp. 623–630.

[19] R. L. Cruz, "Quality of service guarantees in virtual circuit switched networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, pp. 1048–1056, 1995.

[20] H. Zhang and D. Ferrari, "Rate-controlled service disciplines," 1994.

[21] R. Sivakumar, T. eun Kim, N. Venkitaraman, J.-R. Li, and V. Bharghavan, "Achieving per-flow weighted rate fairness in a core stateless network," in *International Conference on Distributed Computing Systems*, 2000, pp. 188–196.

[22] E. Castillo, *Extreme Value Theory in Engineering*. New York: Academic, 1988.

[23] D. L. Mills, "RFC 1305: Network time protocol (version 3) specification, implementation," Mar. 1992.

[24] J. Schlembach, A. Skoe, P. Yuan, and E. W. Knightly, "Design and implementation of scalable admission control," in *QoS-IP*, 2001, pp. 1–16.

Biography

MASARU OKUDA received the B.S. degree in Information System and Computer Science from Brigham Young University - Hawaii, Laie, HI, in 1989, and the M.S. degree in Telecommunications and the Ph.D. degree in Information Sciences from the University of Pittsburgh, Pittsburgh, PA, in 1996 and 2006 respectively. Currently, he is an assistant professor of Telecommunications Systems Management at Murray State University, Murray, KY. His teaching and research areas include computer and network security, US telecom policies, network protocol analysis, network architecture design, QoS enabled networks, peer-to-peer networks, and video distribution networks. Dr. Okuda may be reached at masaru.okuda@murraystate.edu.