QoS Provisioning Using A Clearing House Architecture

Chen-Nee Chuah, Lakshminarayanan Subramanian, Randy H. Katz and Anthony D. Joseph
{chuah, lakme, randy, adj}@cs.berkeley.edu
Department of Electrical Engineering and Computer Sciences, U.C. Berkeley

Abstract- **We have designed a Clearing House (CH) architecture that facilitates resource reservations over multiple network domains, and performs local admission control. Two key ideas employed in this design to make the CH scalable to a large user base are *hierarchy* and *aggregation*. In our model, we assume the network is composed of various basic routing domains which can be aggregated to form *logical domains*. This introduces a hierarchical tree of logical domains and a distributed CH architecture is associated with each logical domain to maintain the intra-domain aggregate reservations. The parent CH in the logical tree maintains the inter-domain reservation requests. Call setup time is reduced by performing advanced reservations based on statistical estimates of the call traffic across various links. We explore, with simulations, the efficiency of the CH-architecture in terms of resource utilization, call rejections and reservation setup time.**

Keywords-**Hierarchical Bandwidth Brokers, QoS Provisioning, Predictive Online Reservations**

I. Introduction

The unpredictable loss, delay and delay jitter in the conventional Internet can adversely impact the performance of real-time applications, such as audio and video conferencing. Such applications may need proper resource provisioning in the network to achieve acceptable end-to-end quality. There has been a significant research effort in changing the Internet architecture to one that can provide different service levels for specific quality of service (QoS) requirements. However, it remains an open question how to regulate the provisioning of resources or services to a particular group of users or hosts depending on the network conditions.

Integrated Services (Int-Serv) with RSVP signaling [1] introduces per-flow reservations in the network to provide per-flow QoS guarantees. This approach requires maintenance of individual flow states in the routers, and its signaling complexity grows with the number of users. Therefore, Int-Serv with RSVP may potentially become a bottleneck itself with negative impact on end-to-end performance. Differentiated Services (Diff-Serv) [2], on the other hand, relies on packet markers, policing functions at the edge routers, and different per-hop behaviors at core routers to provide coarse-grained QoS to aggregated traffic. Diff-Serv uses agents, known as bandwidth brokers (BB) [3], to negotiate service-level specifications (SLSs)[1] [4] between different autonomous systems, whereby SLSs describe the minimum expected level of service and volume of traffic that can be exchanged between two domains. Some kind of admission control is required to make sure that there are sufficient resources available to meet the SLSs. An initial evaluation of bandwidth broker signaling can be found in [5]. However, it remains unclear how a BB computes the amount of resources needed for a service type or how it sets up end-to-end resource reservations over multiple domains. We still need a better understanding of the inter-broker communications.

[1] A Service Level Specification (SLS) is a set of parameters and their values which together define the service offered to a traffic stream by a DS domain.

A. *Motivation*

The lack of a well-studied policy architecture to regulate resource provisioning in a scalable manner has motivated our design of a Clearing House (CH) as an alternative solution. The Clearing House attempts to provide higher QoS assurance levels and higher network utilization, as offered by stateful networks (e.g. Int-Serv), while maintaining the scalability and robustness found in stateless network architecture (e.g. Diff-Serv). Reference [6] has explored a possible implementation of such QoS architecture in SCORE network where each packet carries additional state information in its header (Dynamic Packet State).

The Clearing House has long been existent in the banking industry as an establishment where financial institutions adjust claims for cheque and bills, and settle mutual accounts with each other. Even in the context of the Internet, the concept of the Clearing House is not entirely new. In 1995, a consortium of leading California Internet Service Providers formed the Packet Clearing House (PCH) [7] to coordinate the efficient exchange of data traffic from one network to another. The PCH member agreement includes cost of membership, peering connections and routing policy. For example, PCH members may exchange traffic between networks without any settlement fees. However, the impact of PCH and its subsequent developments are unclear. Many architectural design issues involved in such an Internet Clearing House remain unexplored. On the other hand, increasing number of Internet companies are now offering on-line network resource brokerage by gathering guaranteed demand from the prospective customers and matching it with the sellers' capabilities. Examples include RateXChange's Real-Time Bandwidth Exchange (RTBX) [8], Arbinet Global Clearing Network's trading floor for minutes [9] and Priceline.com's future plan to offer time-block brokerage for domestic and international long-distance calls [10]. Such business models involve Clearing House mechanisms, which have not been studied carefully for the Internet scenario where bandwidth efficiency and QoS assurance are important.

B. *Scope and Layout*

We design the Clearing House as an inter-domain policy architecture that regulates the resource allocation to different groups of aggregated traffic. In our model, various basic domains (based on administrative or geographic boundaries) are aggregated to form *logical domains (LD)*, as shown in Fig. 1. These logical domains are then aggregated to form larger logical domains and so forth. This introduces a hierarchical tree of the LDs and a distributed CH architecture is associated with each LD. Individual CH-nodes can be thought of as agents that maintain aggregate reservations for all the links within the same domain at a particular hierarchical level. The reservations between neighboring domains are monitored by the parent CH-node. This hierarchical tree of CH-nodes form a "virtual overlay network" on top of existing wide-area network topology.

Although we present the CH as a general architecture, one specific example where CH will be useful is for IT managers to manage a WAN (wide-area network) that interconnects corporate offices, remote and mobile employees. Corporations have turned to Internet VPNs to deliver performance, security and manageability to their various sites scattered across the country. However, existing SLAs[2] [11] between service providers (ISPs) and customers have focused on backbone performance guarantees, and do not reflect the end-to-end performance of individual applications. In addition, some fraction of the traffic may traverse multiple routing domains that belong to different ISPs. IT managers still face the challenge of provisioning the total capacity (VPN tunnels) efficiently among the various types of traffic to meet application requirements such as latency and reliability characteristics. A CH-architecture can be deployed in this case to handle intra- and inter- domain resource allocation. For example, IT managers can treat each corporate site as a basic domain, and introduce a CH-node at each site to monitor the traffic flow, adapt resource allocation, and re-negotiate SLAs with the corresponding ISPs when necessary. Various sites can be aggregated to form a larger LD, or several LDs, depending on the layout of the corporate network. The CH-nodes associated with these LDs can coordinate the aggregate resource allocation between domains that reflect on end-to-end performance requirements.

The CH architecture can support two types of reservations: advanced and immediate reservations. An advanced reservation (AR) is time-limited and resources are allocated in advance based on statistical estimates of aggregate traffic over a particular link. We use advance reservations to reduce the call setup time, and the potential violation of QoS assurance if the traffic arrives before the resources are properly reserved. Such approach has been used for resource management in Virtual Private Networks (VPNs) as reported in [12]. Traffic statistics can be easily obtained by leveraging the existing traffic monitoring and measurement systems, through either third party organizations, e.g. MIDS Internet Weather Report (IWR) [13], Internet Traffic Report [14], or the ISPs themselves, e.g. Cable & Wireless USA [15] and AT&T IP Services [16]. We can also gather information from end nodes using software toolkit such as SPAND [17], which enables the networked applications to report the performance they perceive as they communicate with distant Internet hosts. Advance reservations only track the aggregate traffic pattern at a large time-scale (e.g., different hour of the day) and do not reflect the rapid fluctuations of local traffic volumes produced by end-users. Immediate reservations (IR), on the other hand, can be made on demand when the existing reservations become insufficient to accept the new admission requests. The local CH-nodes performs admission control to ensure that QoS assurance to the existing connections are not violated. For evaluation purposes, we only consider advance reservations in this paper.

[2] A service level agreement (SLA) is an explicit statement of the expectations and obligations that exist in a business relationship between two organizations: the service provider and the customer

The focus of this paper is on the architecture design of the Clearing House, its resource reservations and reservation request scheduling mechanisms. We evaluate, with simulations, the costs and benefits of the CH approach, e.g. the tradeoff between the reduction in setup time, call rejections and resource utilization by aggregating reservations. The rest of the paper is organized as follows. We discuss the related work in Section II. In Section III, we describe the design goals of the CH architecture and assumptions we make about the network. We introduce the Clearing House architecture in Section IV, with an overview of the hierarchical tree formation and the role of each component. Section V describes the advanced reservation strategies based on a Gaussian traffic predictor. We present the simulation framework in Section VI and performance evaluation of our design in Section VII and conclude the paper in Section VIII.

II. Related Work

The Internet2 QoS working group have been investigating the inter-broker signaling to automate the adaptive reservation scenario using an inter-domain Diff-Serv test-bed, Qbone [18]. However, the Bandwidth Brokers (BBs) are currently configured manually, and many design decisions remain open. Several BB implementations have been proposed and analyzed in [3], [5], [19], [20] as a scalable QoS provisioning mechanism over the Diff-Serv architecture. However, many of these proposals only consider peer-to-peer structure of BBs or Reservation Agents (i.e., flat rather than hierarchy). The reservations are performed locally between two neighboring domains without reflecting the traffic and network variation in other domains that lie in the end-to-end path between source and destination networks. In addition, these studies do not include advanced reservations.

Advanced reservations are analogous to the existing SLSs between two peering ISPs. The interaction between advance reservation and admission control for immediate reservation requests has been studied in [21], [22], whereby individual users specify the bandwidth requirement at the time of requests. We, on the other hand, use a traffic predictor to estimate the aggregate bandwidth demand without relying on how well individual flows keep to their bandwidth specifications. Reference [12] described a similar adaptive reservation scheme optimized for VPNs, and compared its performance to static provisioning using real traffic traces. However their work only considers a single ISP scenario. It is important to understand the performance of the traffic predictor in the context of the Clearing House where an under- or over- estimation of bandwidth requirement for aggregate traffic originating from one particular domain can affect the network utilization on links that are shared by other neighboring domains.

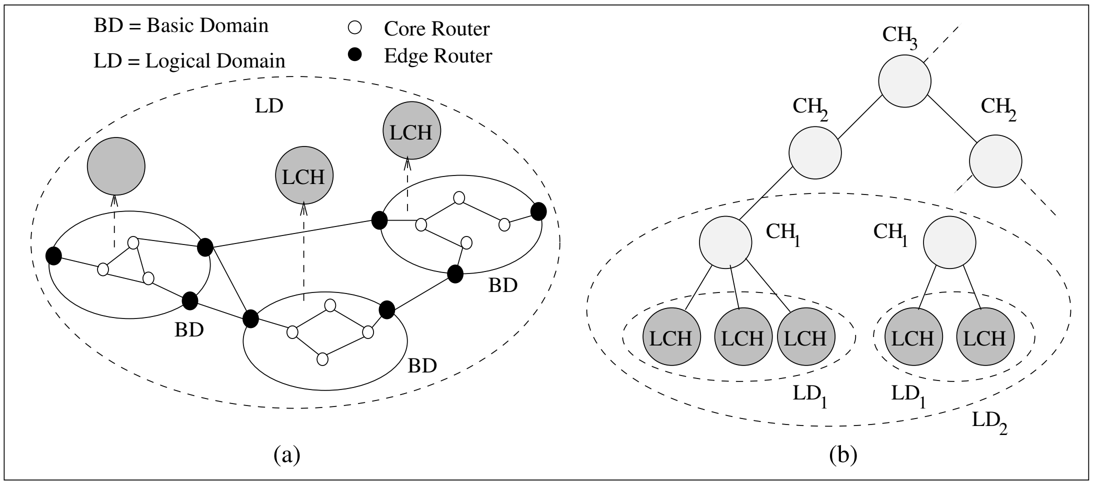

Fig. 1. (a) Local Clearing House (LCHs) associated with their basic domains that lie within a single logical domain (b) An hierarchical CH-tree with multiple levels of logical domains.

A new definition of QoS provisioning was defined based on mathematical economic models in [23]. The authors proposed a set of methodologies to compute the equilibrium prices based on the demands placed by the users, and the optimal allocation of buffer and link resources to each of the traffic classes. However, results in [23] were based on a single-node model that has multiple output links with an output buffer. Further studies are needed to investigate the applicability of this result to large networks, and develop market based mechanisms to admit and route sessions over multiple domains.

The concept of hierarchical databases has long been used in telephone network switching, and for user mobility management in the PCS network. In both cases, the sessions are circuit switched or connection oriented, and each session generates a constant bit rate (CBR) traffic. This paper explores a different problem space where all the sessions are connectionless, and individual flows can generate variable bit-rate traffic (due to compression), which allows statistical multiplexing at the packet level. The hierarchy of increasingly aggregated flows is common in the telephone network, but it is based on a fixed bit-interleaved digital multiplexing, as defined in the PDH standard [24], e.g. 24 telephone channels are carried at the T1 level (1.544 Mb/s). Each session is assigned a fixed time-slice of the resources. In this paper, the CH-architecture aggregates call requests and perform admission control decision in real-time based on the available bandwidth and network performance, leading to a constantly varying statistical multiplexing gain.

III. Design Goals and Assumptions

One of the basic design requirements of the Clearing House is to extend rather than modify the existing network architecture to minimize the development cost. The CH enhances the services and performance of the network by adding some functionality to the network access routers (or edge routers) and leveraging information from traffic monitoring devices. The basic goals that drive our design of the Clearing House are:

- **QoS Provisioning:** The CH attempts to provide an end-to-end coarse-grained QoS assurance by performing aggregate resource reservation along the path from source to destination host networks.
- **Scalability:** The CH has a hierarchical tree structure that can incrementally scale to support a large user base (i.e. large geographic regions and large volume of simultaneous calls). We strive to minimize the number of states maintained in each node of the CH and the backbone routers.
- **Efficient Network Utilization:** The CH attempts to optimize the overall throughput while preserving the QoS of admitted calls by performing admission control based on information of the entire network stored in the CH database, e.g. reservation status and available bandwidth of inter-domain links. The accuracy of this information depends on the time granularity at which database is being updated.
- **Secure Real-time Billing:** CH is a distributed database that can store the billing prices, quality and latency provided by various ISPs. It can inform ISPs and customers about the available bandwidth, bandwidth demand, and reservation costs. This aspect of CH has been explored in another paper [25].
- **Support for Multicast Operations and Mobility:** The CH infrastructure can be easily extended to support multicast operations by coordinating resource reservations and cost-sharing between the group members at different level of the multicast tree. The CH can also keep track of the dynamic path changes and modify resource reservations accordingly to support mobility. This is part of future work, and is out of the scope of this paper.

We focus mainly on the first three design goals in this paper. Specifically, we describe how the CH architecture establishes and negotiates aggregate resource reservations between neighboring domains in a hierarchical manner. We will not discuss how the reservation requests are translated to a specific traffic control agreement (TCA) that can be understood by the edge devices, or how these TCAs are delivered to the edge routers.

In designing the CH architecture, we make the following assumptions:

- The networks are capable of providing different service levels through a combination of packet marking, scheduling and queue management mechanisms. We assume network edge routers can verify whether the QoS assurance agreement is met by measuring the packet loss, average queuing delay, delay variance etc.
- Every routing domain has the capability to monitor and collect statistics of the incoming and outgoing traffic. We assume this information is trustable, and will be used by CH to negotiate resource reservations with neighboring domains.
- Control paths (e.g. reservation requests) and data paths are separated. We decouple call setup and resource reservation procedures to reduce the overall response time and increase the system throughput.

IV. Clearing House Architecture

In this section, we provide a complete description of the Clearing House and its various functionalities.

A. Hierarchical CH-Tree

First, we define several terms that we use in our discussions:

- A *basic domain* refers to a basic routing domain in the network. For example, a basic domain can be a small subset of backbone networks owned by a specific Internet Service Provider (ISP) which serves multiple host networks. We assume that the Internet can be divided into non-intersecting basic domains.
- A *logical domain (LD)* is a collection of adjacent basic domains that are clustered to form a larger domain, which may refer to geographic boundaries (e.g. states, or small countries) or for administrative reasons (e.g. campus, company etc). On the other hand, a big ISP backbone network can span across multiple domains.

The various logical domains can be clustered to form a larger logical domain. We can repeat the same process until we are left with one logical domain that represent the whole network. Together, these domains form a hierarchical tree, known as *CH-tree*. A distributed CH architecture is associated with every LD represented by a node in this tree. A CH-node at a particular level of the CH-tree maintains the reservation states of the LD, which is the union of all the sub-LDs whose states are maintained by its children CH-nodes. The actual number of CH-nodes in the distributed architecture will vary as a function of the size of the LD, and the level of the LD in the hierarchy. Mirror sites can be added to every CH-node to support fault tolerance and higher availability.

A CH in the hierarchy aggregates all inter-LD call requests to a particular domain and sends this aggregated request to the parent CH. In other words, all call requests between two LDs would be aggregated as a single request at a parent CH. Therefore, a CH of a LD that is a collection of K sub-LDs would contain $O(K^2)$ call requests. Typical values of K are around $10-50$. Only the CH at the local operators (at the leaf nodes of the CH-tree) maintain per-flow state information.

Although it is easy to extend the depth of the CH-tree to represent the whole network, this paper only considers the case of a two-level tree with one parent CH-node (a single logical domain) and multiple children nodes (basic domains). We quantify the performance of Clearing House and reservation strategies in this simple case.

B. Local and Global Clearing House

A CH of a basic domain is called a *Local Clearing House (LCH)* and all other CH-nodes up in the hierarchy are called the *Global Clearing House (GCH)*. For our initial design, we assume that the basic domains are non-overlapping to ensure that a user at a particular location has a unique LCH to contact for resource reservation or billing purposes. We concentrate on the case where there is only one GCH.

All service providers present in a domain can advertise the costs of reserving bandwidth on their links to the LCH. The service providers offer various prices based on the domain of the final destination (e.g., call Canada 7/9 cents/min) and the traffic load [26]. The LCH is responsible for the following set of operations:

- An LCH keeps track of the amount of existing reservations and the available bandwidth on all the links between edge routers within the same basic domain.
- Based on the statistics of the intra-domain traffic, an LCH performs advance resource reservations on the intra-domain links. It also makes local admission control decisions when a new call request arrives.
- An LCH monitors the aggregated incoming and outgoing traffic exchanged with other neighboring basic domains and uses these statistics to estimate the future bandwidth usage. The predicted bandwidth usage for inter-domain traffic, and the aggregate reservation state on inter-domain links are reported to the GCH at the parent level in the hierarchy.
- An LCH aggregates inter-domain call requests and forwards the aggregate reservation request to the parent GCH. If there are sufficient network resources on the end-to-end path, the LCH will receive acknowledgments from the GCH and the new calls will be admitted. Otherwise, the calls will be rejected.

A GCH, on the other hand, acts as the coordinator among the various basic domains and handles resource allocation for all inter-domain calls:

- A GCH keeps track of the links that run between children sub-domains and their corresponding reservation status and network performance such as latency, average queuing delay, and packet loss rate.
- Based on the traffic statistics collected from all the children-LCHs, a GCH estimates bandwidth usage on a particular inter-domain link and performs advance reservation accordingly (see Section V).
- A GCH aggregates call requests received from its children LCHs, and performs advance reservations for the inter-domain links that lie within its LD. If the reservation

request involves links that connect to neighboring LDs at the same level, the reservation request will be forwarded to the parent GCH, but this is not addressed in this paper. A GCH services reservation requests for aggregated traffic instead of individual calls.

C. Caching and RxW scheduling

We can employ two enhancements to improve the performance of the Clearing House, namely caching and RxW scheduling [27]. An LCH or GCH can cache intra-domain and inter-domain computed paths for previous reservation requests. This can reduce the service time of a reservation request at a CH. Since the number of logical domains maintained by a CH is small (10-50), a local cache can typically store all inter-domain paths. A local cache in a LCH can also store the price listings of various service providers to different destinations. RxW scheduling [27] is a very good algorithm for increasing the throughput of the CH. It schedules the aggregated call requests with the maximum value of $R \times W$, where R is the number of requests aggregated and W is the maximum waiting time of an aggregated request. This scheduling algorithm maximizes the throughput (number of call requests) serviced without unduly affecting the response time for call requests.

V. Resource Reservations and Traffic Predictor

A. Overview

This section describes the resource reservation and traffic monitoring mechanisms involved in the Clearing House infrastructure, which are critical for providing QoS in wide-area networks.

In many existing Diff-Serv proposals, bandwidth brokers negotiate the volume and the price of high-priority traffic to be exchanged between different domains through service level specifications (SLSs). However, the fluctuation of local traffic volumes produced by end-users has to be reflected in the SLSs between core networks. Fig. 1 shows a typical scenario that spans multiple basic domains. We assume each edge router (ER) or a third party prober can easily monitor the incoming and outgoing traffic on both the intra-domain links, and the links connecting to other neighboring domains. The LCH in each basic domain retrieves link properties (e.g. reservation status, link utilization, statistics on latency and packet loss) by querying ERs or probers seen in the topology map. This is not an unreasonable assumption because real-time report on Internet traffic statistics, and performance of major ISPs are currently available, and traffic monitoring architecture is in place in different parts of the network. As mentioned in Section I-B, we focus on advance reservations in this paper, whereby resources are reserved for aggregated traffic following a particular path in advance for a specific time period based on a traffic predictor.

B. Advanced Reservations

We assume that the ERs can measure mean, m, and variance, σ^2, of the aggregate priority traffic for different times of the day based on rates sampled during a specific measurement window, T_{mes}. ERs send regular updates to LCH, which uses these statistics to predict future bandwidth usage along a specific link.

Gaussian Predictor: When the number of individual flows gets large, the aggregate arrival rate tends to have a Gaussian distribution under *Central Limit Theorem* [28]. We estimate the required bandwidth as: $\hat{B} = m + \alpha\sigma$, where α is a QoS factor that controls the extent to which the bandwidth predictor accommodates variability in the samples. In the buffer-less case, the probability of packet loss is approximately $Q(\alpha)$, where $Q()$ is the complementary cumulative distribution of the standard Gaussian distribution.

An LCH uses the Gaussian predictor to set up advance reservations between different ERs within its own basic domain. Similarly, LCH keeps track of the mean and variance of aggregate traffic that flow into or out of the neighboring basic domains, and forwards this information to the parent GCH. The parent GCH uses a Gaussian predictor to estimate bandwidth usage between different children sub-domains, and establish advanced reservations between them. This process is repeated at different levels of the CH-tree and time-based advanced reservations are established on all the intra- and inter-domain links based on different sets of traffic predictors.

Internet data traffic exhibits burstiness at multiple time-scales. Therefore, a predictor ($\hat{B}$) based on a given sampling window can underestimate the bandwidth requirement that varies at a shorter time-scales, resulting in possible violation of QoS guarantees. One option to cope with the changing user requirements is to signal each change in flow activities at ER through the LCH to the core networks. However, this requires core networks to keep per-flow state information, and would lead to the same scaling problem that Int-Serv architecture faces. In our design, the reservation requests between ERs reflect aggregated changes, and only propagate to the nearest LCH. The regular updates of reservation status are decoupled from the actual reservation requests.

If the predicted bandwidth, $\hat{B}$, overestimates the actual bandwidth required, it results in inefficient resource utilization. Over provisioning an inter-domain link for aggregate traffic originating from a particular source domain may result in unnecessary call rejections for traffic flows coming from other domains. The performance of of $\hat{B}$ heavily depends on the measurement window, T_{mea}, and the time-scale at which the bandwidth demand varies. We explore these tradeoffs in our simulation study (Section VI).

C. Admission Control

Whenever a sender wants to make a call to a receiver, there should be sufficient resources along the particular path from the sender to the receiver. Since on-line resource reservation is very costly, the goal of our design is to minimize the amount of per-link reservation that needs to be made for a particular call. Based on the reservation status within a domain, a particular path is chosen such that the number of new per-link resource reservations is minimized. If the LCH fails to locate any links with sufficient resources reserved to complete

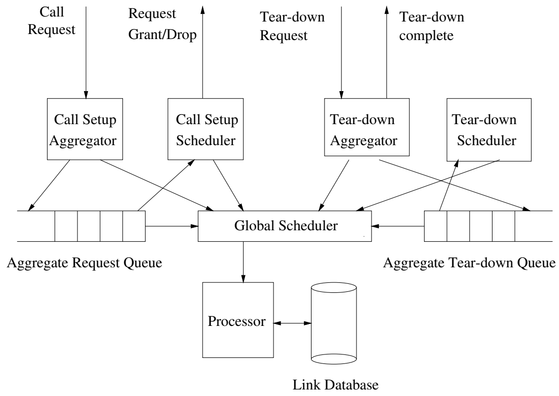

Fig. 2. Simulation Model

a chosen path, the ER will block the new call. The admission control decisions involve some trade-offs in the QoS assurance and the number of rejected calls.

VI. SIMULATIONS

A. Framework

We have developed a simulator that simulates the actions of a single-node Clearing House. The CH is treated as a database in which, the reservations along the various links in the topology are maintained. The database stores the propagation delay along the various links. The CH-node has the following simple structure:

```
typedef struct {
  Database *database;
  // Database of all links in topology
  IntexTable *itable;
  // Hash index for efficient database access
  PendingQueue *pq;
  // Pending queue of call requests
  TeardownQueue *tdq;
  // Pending queue of tear-down requests
  TeardownAggregator *tda;
  // Database of aggregated tear-downs
  AggregateRequests *arqueue;
  // Queue of aggregated requests
  Network_Stats *net;
  // Network Statistics
  Cache_Paths *cp;
  // Cache of Shortest Paths
}ClearingHouse;
```

There are four important components in our Clearing House simulator. As illustrated in Fig. 2, they are call setup aggregator, call setup scheduler, call tear-down aggregator and call tear-down scheduler. These four processes are scheduled by the global scheduler in a weighted round-robin fashion. RxW scheduling is employed by the call setup scheduler and the cache is used for storing previously computed shortest paths between different domains.

B. Network Topology

For our simulations, we use the topology shown in Fig. 3, which is an approximation of the AT&T Worldnet IP backbone as reported in [12]. Assume a corporate network has to interconnect different sites that reside in 12 important cities in the USA. Each site is represented by a basic domain, and they are interconnected by VPN tunnels with limited bandwidth. A Local Clearing House (LCH) is associated with each basic domain. Call requests are generated between various domains based on a weighted distribution and sent to the LCHs. The 12 domains are grouped to form one larger LD and a CH-node is introduced to service aggregate reservation requests and coordinate resource provisioning between multiple domains.

C. Workload Models

We use voice traffic as a workload to drive the initial evaluation of the Clearing House. The call arrival rate in each domain i is modeled as an independent Poisson process of intensity λ_i calls per second, and the call duration is exponentially distributed with a mean of $1/\mu$=120 s. We define the traffic load arriving at each LCH i as $\rho_i = \frac{\lambda}{\mu}$, where $i = 1, 2, \ldots, 12$. We assume silence suppression and model each voice source as an on-off Markov process. The alternating active ('on') and silence ('off') periods are exponentially distributed with average durations of 1.004 s and 1.587 s. We consider an average talk spurt of 38.53% and average silence period of 61.47% as recommended by the ITU-T specification for conversational speech [29]. We assume that the voice source generates CBR traffic of 80 Kbps[3] when 'on', and 0 Kbps when 'off'.

VII. PERFORMANCE EVALUATION

In this paper, we study the performance characteristics of a single Clearing House node and the prediction algorithm. These performance characteristics give us better insight into the policies that need to be adopted for building a complete Clearing House architecture.

A. Gaussian Predictor Characteristics

The first set of experiments explore the robustness of the Gaussian predictor with respect to traffic variability and measurement window, T_{mea}. We evaluate the bandwidth predictor using both simulated traffic and real voice traces.

In the first case, we simulate individual voice sources based on the on-off Markov model (VI-C) with a traffic load of ρ = 180 calls for a particular domain. We use a moving window of $\{1, 2, ..., 9, 10\}$ minutes for measurement and traffic predictions. Fig. 4 shows a sample path of the aggregate traffic, along with the predicted bandwidth usage, $\hat{B}$ for T_{mea} = 1 and 10 minutes. Note that the predictor with T_{mea} = 1 minute tracks the actual capacity requirement better than the 10-minute predictor. If we allocate bandwidth based on the maximum rate (80 Kbps), we need a total bandwidth of $N \times 80$ Kbps, where N is the number of flows. We define multiplexing gain as: $(N \times 80)/\hat{B}$. Advanced reservations based on 1-minute predictor achieves a multiplexing gain that ranges from 1.37 to 4.13 with mean of 1.62 and standard deviation of 0.318 when traffic load ρ = 180.

[3] Assume 8 KHz 8 bits/sample PCM codec was used with 20 s frame per packet. With 12 byte RTP header, 8 byte UDP header and 20 byte IP header, the size of each voice packet = 200 bytes. The bandwidth required will be $(200 \times 8)/20 = 80$ Kbps.

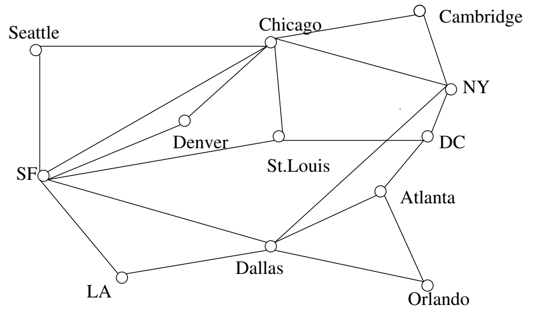

Fig. 3. Topology of the IP backbone with 12 basic domains

We repeat the experiments using voice traces collected from actual conversations between professors, students, and staff members during research group meetings and in a computer science graduate-level class [30]. The voice traces were recorded according to the MASH archive file formats [31]. The individual voice traces are aggregated according to a Poisson arrival model, and the sample path is plotted in Fig. 5, along with the bandwidth predicted using T_{mea} = 1 and 10 minutes. The 1-minute predictor still tracks the actual bandwidth usage closely, but the 10-minute predictor fails to keep up with the smaller time-scale fluctuation. In both cases (simulated and actual traffic), the probability of underprovisioning is less than 1% for all values of T_{mea} taken into consideration.

We measure the effectiveness of $\hat{B}$ in terms of the percentage of under-utilized capacity due to over-estimation of bandwidth requirement:

- $Overprovisioning = (\hat{B}_l - \sum_{ij} R_{ij})/\sum_{ij} R_{ij}$ where $\{ij\}$ are source-destination pairs that have traffic routed through link l, and R_{ij} is the actual traffic between source domain i and destination domain j routed through link l.

We run 100 simulations, each for 1 hour, using both simulated traffic and actual voice traces to evaluate the efficiency of the Gaussian predictor. The average $\%\ overprovisioning$ is plotted in Fig. 6 and Fig. 7 for the two cases. Observe that the amount of over-allocation increases with T_{mea}, as the predictor becomes less responsive to both upward and downward trends in the voice traffic. The $\%\ overprovisioning$ is higher when real traces are used, which implies that the actual voice traffic is more bursty than the traditional on-off Markov model for voice. We need smaller T_{mea}, i.e. 1-2 minutes, to track the fluctuation of the voice traces.

B. CH Node Characteristics

In our simulations, a single Clearing House node keeps track of the reservations along the various links in the topology given in Figure 3. The CH-node admits call requests between two domains and performs reservations on the various links and services the requests. The reservations are maintained as a back-end database which is constantly updated.

Given this setting, we test the performance of this node under various loads. We define the load as the number of reservation requests per second arriving at the CH node. A weight

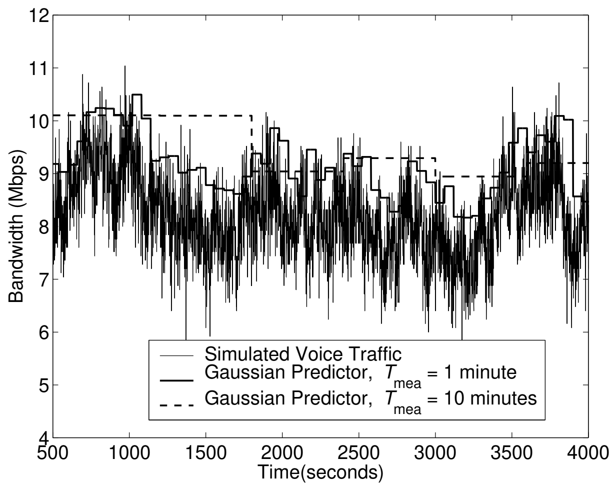

Fig. 4. Gaussian predictors for simulated voice traffic with measurement of 1 minute and 10 minutes, for ρ = 180.

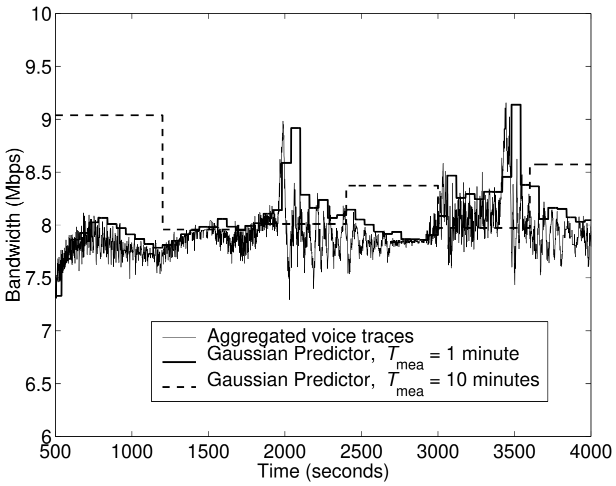

Fig. 5. Gaussian predictors for actual voice traces with measurement of 1 minute and 10 minutes.

proportional to the population of the city is associated with every node in the topology. The calling pattern is derived from a probabilistic model in which the probability associated with every node is proportional to the weight of the node.

We use three different scheduling policies to evaluate the CH-architecture. They are:

1. **RxW Scheduling:** RxW scheduling is an aggregate scheduling mechanism in which multiple requests of a particular type are aggregated into one request (Section IV-C).
2. **FIFO Scheduling:** This is the normal scheduling policy based on the first come first serve principle.
3. **Bounded FIFO:** Bounded FIFO refers to FIFO scheduling with bounded response time, i.e. requests with high waiting times are directly dropped.

We measure the throughput, mean response time, mean teardown time and the call blocking rate for varying loads. We also measure the fairness of the different scheduling policies in terms of the variability of individual response time.

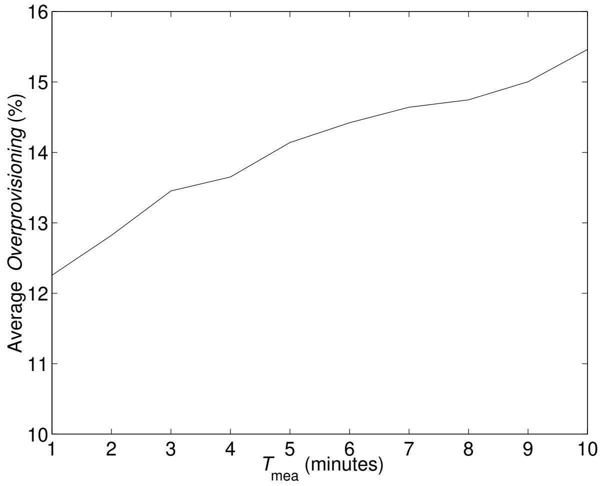

Fig. 6. Average $overprovisioning$ (in %) when reservations are made based on Gaussian predictors for simulated traffic at ρ=180.

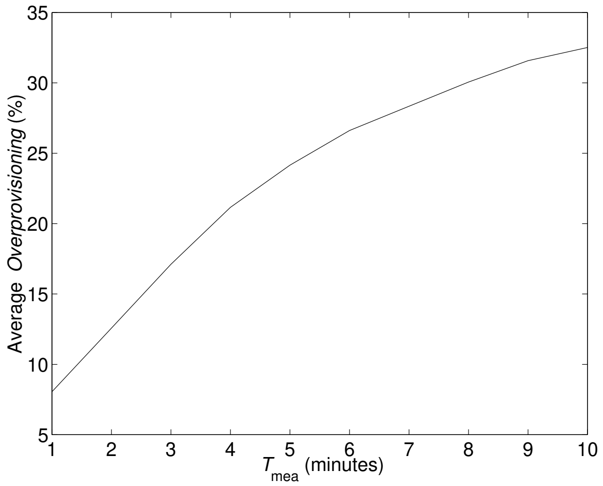

Fig. 7. Average $overprovisioning$ (in %) when reservations are made based on Gaussian predictors for aggregated voice traces.

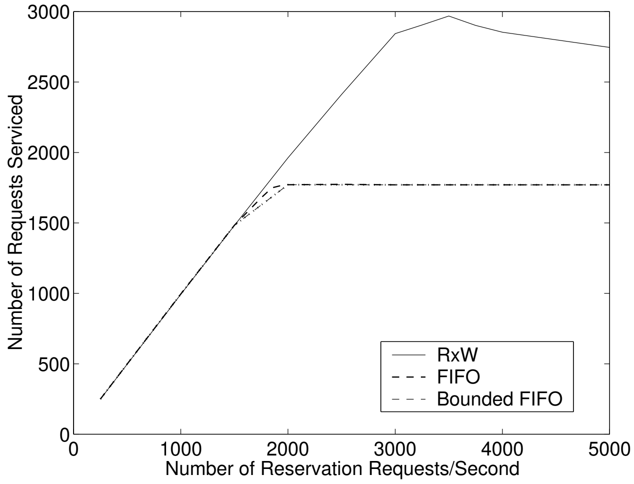

Fig. 8. Throughput of a Clearing House node as the traffic load is varied

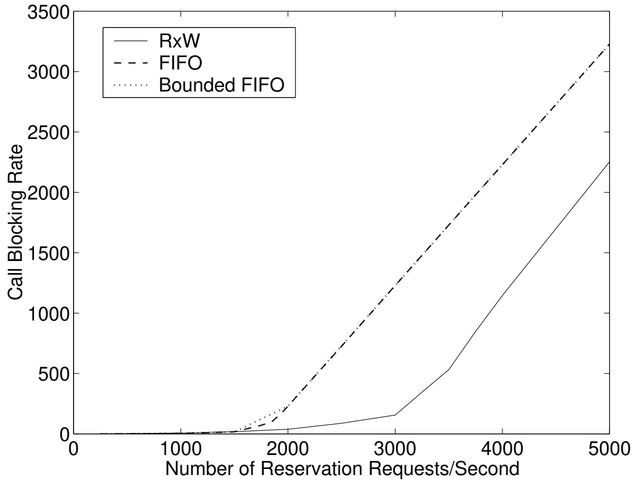

Fig. 9. Call Blocking Rate as the traffic load is varied

B.1 Throughput Characteristics

Throughput is measured as the number of calls serviced by the CH-node per second. From Fig. 8, we observe that the peak throughput obtained using RxW is 71% more than the peak obtained using FIFO. By introducing a bounded response time policy in FIFO scheduling, the throughput is unaltered. Using RxW scheduling, the CH can successfully take a load of 3500 calls/s while the CH can only take a load of 1850 calls/s using FIFO scheduling. The throughput of RxW drops once the load increases beyond 3750 calls/s.

B.2 Call-Blocking Characteristics

A call is "blocked" when the reservation request is dropped by the scheduler either due to insufficient resources or excessive load. The blocking rate obtained using RxW scheduling is much less than that of FIFO scheduling. The call blocking rate of FIFO scheduling is unaffected by the bounded response time constraint. The call blocking rate is negligible until a load of 1500 calls/s. For RxW scheduling, the call blocking rate is less than 10% until a load of 3200 calls/s. After a certain threshold, the blocking rate increases linearly with the load indicating a saturation point of throughput.

B.3 Response-time Characteristics

The response time is defined as the time taken to service a call request by the CH. In Fig. 10, we plot the mean response time as a function of the load for the three scheduling policies. The response time of bounded FIFO is always lower than 0.5 s and is much lesser than that of normal FIFO after a load of 2000 calls/s. The mean response time of RxW increases linearly after 2500 calls/s. After a load of 1850 calls/s, the mean response time of FIFO scheduling shoots up rapidly and is an order of magnitude larger than bounded FIFO or RxW scheduling.

B.4 Tear-down Characteristics

It is important to measure the time taken to tear down the reservations along a particular path. Fig. 11 shows some very

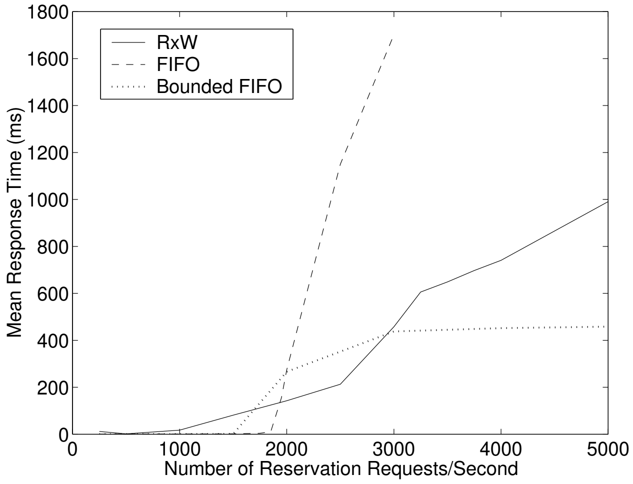

Fig. 10. Mean Response time as a function of traffic load.)

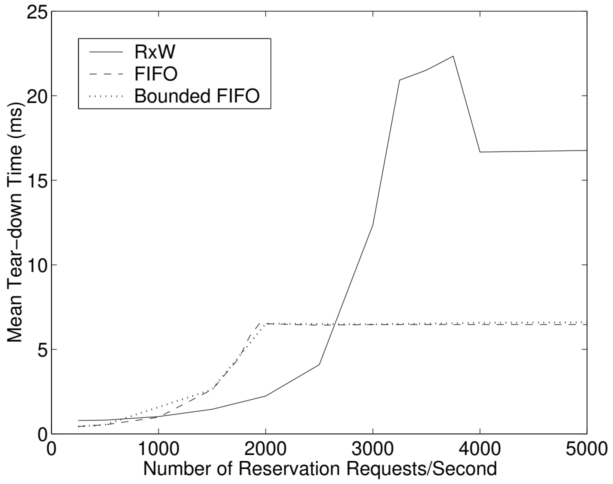

Fig. 11. Tear-down response time as a function of traffic load.

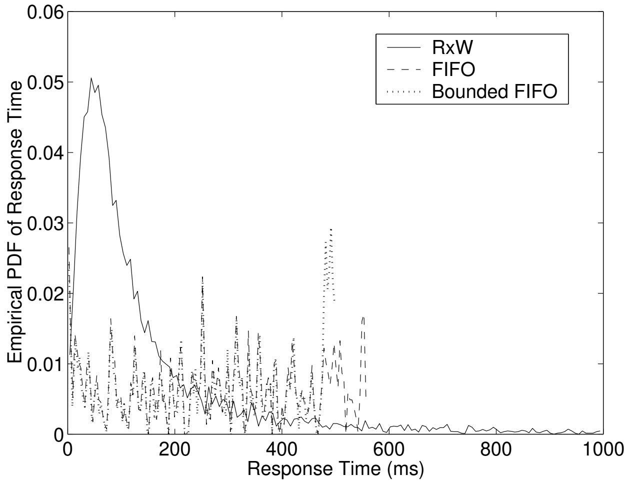

Fig. 12. Distribution of Response time at a load of 2000 requests per second.

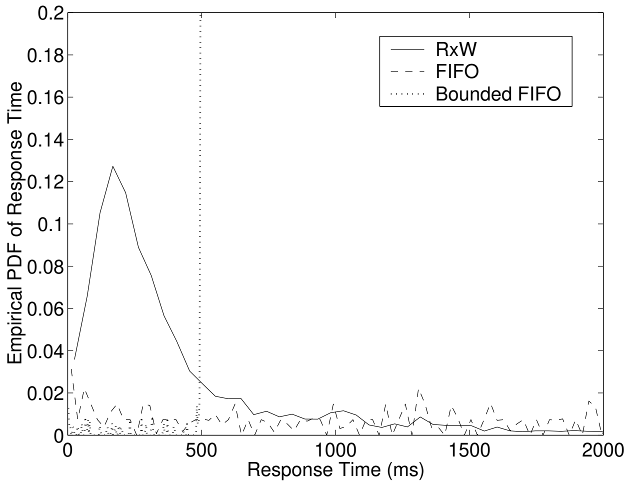

Fig. 13. Distribution of Response time at a load of 3000 requests per second

interesting properties of the tear-down response time. When the throughput of the system decreases at 3700 calls/s load, the tear-down response time drops by 5 ms for RxW scheduling and stabilizes at 16 ms. The mean tear-down time for RxW scheduling shows a steep increase after a load of 2500 calls/s while the mean tear-down time for FIFO policies stabilizes at 6ms beyond a load of 1850 calls/s. The number of tear-down requests is proportional to the throughput of the system at a specified load. Hence, when the throughput of FIFO and RxW scheduling stabilizes at 2950 calls/s and 1770 calls/s at loads of 3500 calls/s and 1850 calls/s, the mean-tear down time stabilizes.

B.5 Fairness of our Approach

We determine the fairness of our approach by studying the variations of response time. In a normal FIFO queue model, the response time of a call request would be proportional to the size of the queue. Since RxW scheduling tries to optimize the throughput of the CH, there might be a few requests which might have to wait for a long time before getting scheduled. Such requests will observe a high response time. In Fig. 12 and Fig. 13, we plot the variations of response time at two critical loads equal to 2000 calls/s and 3000 calls/s. At a load of 2000 calls/s, FIFO scheduling reaches its stable region and at 3000 calls/s, the throughput of RxW scheduling is close to its maximum and the response time of normal FIFO becomes a magnitude higher to 2 s. At a load of 2000 calls/s, a huge percentage ($> 80\%$) of requests in RxW scheduling have a response time in the range of 50-220 ms while the response time of FIFO is distributed over the range of 30-600 ms. A huge percentage ($>90\%$) of the requests in bounded FIFO have a response time between 450-500 ms at a load of 3000 calls/s. Bounded FIFO reaches a very high level of fairness for incoming requests. Even at a load of 3000 calls/s, the response time of FIFO is distributed equally between 30-2000 ms. A few percentage of the requests ($<2\%$) in RxW scheduling suffer due to loss of aggregation and have a high response time (>1 s).

C. Discussion

In summary, the basic strengths of the Clearing House approach are:

- The state that needs to be maintained by the entire CH architecture is shared between various CH-nodes. Every CH-node maintains only the state for its domain.
- The hierarchical model with aggregate reservations provides scalability of the architecture. Core routers do not need to maintain per-flow state information. The architecture supports easy insertion and deletion of the domains from the *CH-tree*. If a particular CH-node gets overloaded due to the growth of user-base, it is possible to split a logical domain (LD) into two or more sub-LDs, and create a new CH-node for the newly created LDs.
- The queue size of call requests in every GCH is bounded due to aggregation of call requests at the LCH. This architecture is optimized for making decisions based on locality. End-to-end resource reservations can be set up quickly through the CH architecture, and therefore reducing the call setup time.
- RxW scheduling increases the throughput of the system by servicing call requests efficiently while keeping the response time low for a large percentage of the requests.
- Bounded FIFO scheduling achieves a high degree of fairness for high loads. The response time of most of the requests serviced are restricted to a small range.
- Caching of inter-domain paths can enhance the performance of the system considerably.

VIII. Conclusions

We have presented the design of a Clearing House architecture that coordinates inter-domain reservations for aggregate traffic that transits multiple routing domains. The design maintains the scalability of the architecture by using a hierarchical CH-tree structure and aggregating reservation requests at multiple levels of the logical tree. We use a Gaussian predictor to estimate bandwidth usage and set up reservations in advance to reduce the overall reservation setup time. Our simulations demonstrate the effectiveness of the traffic predictor and the Clearing House design. Results show that the Gaussian predictor is robust if T_{mea} is smaller than the time-scale at which bandwidth demand varies, regardless of the number of flows being aggregated. We consider RxW scheduling for servicing reservation requests, and results show that high throughput can be obtained while maintaining reasonable response time of a particular reservation request. Our results also show the fairness of the Clearing House in terms of the response time experienced by individual requests.

Acknowledgments

The authors are grateful to Ramakrishna Gummadi, Helen J. Wang, Bhaskaran Raman, Ben Y. Zhao, and other ICEBERG members for their insightful feedback. Discussions with Borje Ohlman from Ericsson Lab and Julio Navas from Siemens Technology are enlightening and greatly appreciated.

References

[1] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, "ReSerVation Protocol (RSVP) version 1 functional specification," Internet RFC 2205, IETF Network Working Group, September 1997.
[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weise, "An architecture for differentiated services," Internet RFC 2475, IETF Network Working Group, December 1998.
[3] British Columbia Institute of Technology, "CA*net II differentiated services: bandwidth broker high level design," November 1998, http://www.merit.edu/working.groups/i2-qbone-bb/.
[4] D. Grossman, "New terminology for DiffServ," Internet Draft, draft-ietf-diffserv-new-terms-02.txt, November 1999.
[5] M. Günter and T. Braun, "Evaluation of bandwidth broker signaling," *Proc. of IEEE 7th International Conference on Network Protocols*, pp. 145-52, October 1999.
[6] I. Stoica and H. Zhang, "Providing guaranteed services without per flow management," *Proc. of ACM SICGOMM*, pp. 81-94, September 1999.
[7] California's Internet Interchange: Packet Clearing House, http://www.pch.net/, 1995.
[8] RateXChange, http://www.ratexchange.com/.
[9] Arbinet Global Clearing Network, http://www.arbinet.com/.
[10] Priceline.com, http://travel.priceline.com/infoctr/comingsoon/welcome.asp.
[11] D. Verma, *Supporting Service Level Agreements on IP Networks*, Macmillan Technical Publishing, 1999.
[12] N. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan and J. E. Van der Merwe, "A flexible model for resource management in virtual private networks," *Proc. of ACM SIGCOMM*, pp. 95-108, September 1999.
[13] Matrix Information and Directory Services Inc. (MIDS), http://www.mids.org/weather/.
[14] Internet Traffic Report, http://www.internettrafficreport.com/.
[15] Cable & Wireless USA Real Time Internet Traffic Statistics, http://traffic.cwusa.com/.
[16] AT&T IP Network Statistics, http://ipnetwork.bgtmo.ip.att.net/.
[17] S. Seshan, M. Stemm and R. H. Katz, "SPAND: shared passive network performance discovery," *Proc. of 1st Usenix Symposium on Internet Technologies and Systems*, pp. 135-46, December 1997, http://www.cs.berkeley.edu/ stemm/spand/index.html.
[18] Internet2 QoS Working Group: Qbone testbed, http://www.internet2.edu/qos/qbone/.
[19] B. Stiller, T. Braun, M. Günter and B. Plattner, "The CATI project: charging and accounting technology for the Internet," *Proc. of 4th European Conference: Multimedia Applications, Services and Techniques*, pp. 281-96, May 1999.
[20] R. Rajan, D. Verma, S. Kamat, E. Felstaine, and S. Herzog, "A policy framework for integrated and differentiated services in the Internet," *IEEE Network Magazine*, vol. 13, no. 5, pp. 36-41, September/October 1999.
[21] O. Schelen and S. Pink, "Resource sharing in advance reservation agents," *Journal of High Speed Networks: Special issue on Multimedia Networking*, vol 7, no. 3-4, pp. 213-28, 1998.
[22] O. Schelen and S. Pink, "Sharing resources through advance reservation agents," *Proc. of IFIP 5th International Workshop on Quality of Service*, pp. 265-76, May 1997.
[23] J. Sairamesh, D. F. Ferguson and Y. Yemini, "An approach to pricing, optimal allocation and quality of service provisioning in high-speed packet networks," *Proc. of IEEE INFOCOM*, vol.3, pp. 1111-19, April 1995.
[24] ITU-T Recommendation G.702, "Digital hierarchy bit rates," 1988.
[25] R. Gummadi and R. Katz, "A lightweight secure hot billing scheme for mobile networks using a clearing house," unpublished.
[26] Y. Paschalidis and J. N. Tsitsiklis, "Congestion-dependent pricing of network services," *Technical Report*, Systems Group, Department of Manufacturing Engineering, Boston University, 1998.
[27] D. Aksoy and M. Franklin, "Scheduling for large-scale on-demand data broadcasting," *Proc. of IEEE INFOCOM*, pp. 651-659, March 1998.
[28] H. Cramer, *Mathematical Methods of Statistics*, Princeton University Press, 1946.
[29] ITU-T Recommendation P.59, "Artificial conversational speech," 1993.
[30] C-N. Chuah, "Statistical analysis of packet voice traffic in Internet multimedia applications," unpublished.
[31] MASH Archive File Formats, http://mash.cs.berkeley.edu/mash/projects/archive/file-fmt.html.