Chapter 3

Methodology

As described in Chapter 1, the main contribution of this thesis work is a comprehensive study of a lightweight, scalable control architecture built on top of Diff-Serv packet level mechanisms to provide better end-to-end QoS support for latency-sensitive applications. This chapter outlines a set of research methodologies that we follow to carry out our analysis, specifically, how we formulate the problems, examine existing systems and evaluate our proposed solutions. Section 3.1 presents the general framework that guides our study. In Section 3.2, we discuss how we model the various workloads and their performance requirements that later drive the performance analysis of our architecture. In Section 3.3, we describe the evaluation methodology: we conducted simulation experiments using both real traces and generated traffic, with a range of parameters driven by different performance goals.

3.1 General Framework

Our research methodology is best summarized by Figure 3.1. We follow an iterative process that comprises the following three phases:

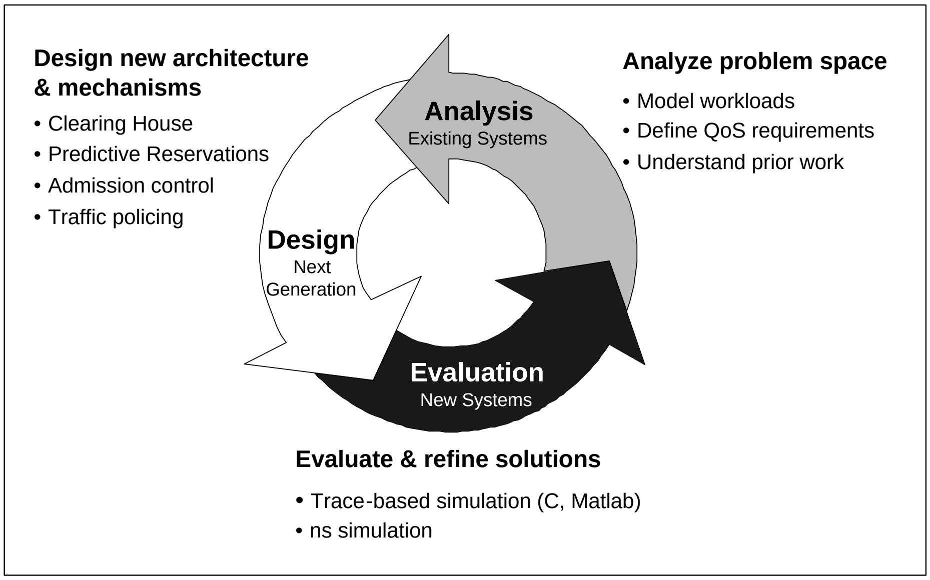

Figure 3.1: Iterative "Analysis, Design & Evaluation" phases.

- **Analysis**

  Before we know what constitutes a better QoS control architecture, we first need to model the characteristics of typical workloads and define their performance requirements. In particular, we focus on the latency-sensitive applications (LSAs) such as packet audio and video because this type of traffic require resource guarantees that are not supported by the current Internet. We will discuss mathematical models that capture the essence of real Internet workload. We also carry out a survey of prior work, as reported in Chapter 2, and determine the pros and cons of each approach.

- **Design**

  In our attempt to strike a balance between Diff-Serv, which provides differential treatment to traffic aggregates, and Int-Serv, which offers per-flow guarantees, we explore a new architecture called the Clearing House that combines features of the two previous approaches. In particular, we implement lightweight session-level control mechanisms such as resource reservations, admission control and traffic policing on top of a state-less Diff-Serv architecture. The initial design has been continuously refined based on discussions with researchers from two major Internet Service Providers that operate nationwide backbone networks in the United States.

- **Evaluation**

  The performance evaluation of our proposed architecture and algorithms are based on a combination of trace-based analysis, simulation experiments, and lab prototyping. We used simulations to examine scalability issues, determine the effect of various design parameters on system performance, and study the trade-offs involved at various operating points. The same set of experiments are often repeated for different "scenarios", where we vary the network topology, aggregate workload pattern, and individual source characteristics.

The next section provides detailed discussions on how we model packet audio applications that represent a typical LSA workload. Section 3.3 describes the general simulation settings but the details of each experiment will be presented in the subsequent three chapters where the corresponding algorithms and performance results are discussed.

3.2 Workload Modeling

In this dissertation, we consider two basic types of workload: data applications that can be sent as Best-effort traffic, and latency sensitive applications that require resource reservations and are sent as High-priority traffic. We have chosen Voice over IP (VoIP) as a representative workload of the latter, because interactive two-way conversations places a much more stringent delay requirements than other LSAs such as playback video. Section 3.2.1 presents the mathematical model that describes VoIP traffic. Section 3.2.2 documents the subjective testing and network measurements we carry out to quantify the performance requirement of VoIP in terms of network centric parameters such as delay and packet losses.

Besides VoIP, there are a wide variety of Internet audio applications that are also latency sensitive, including multimedia conferencing, distance learning, etc. To include these other applications in our analysis, we collected 70 packet audio traces from technical meetings, broadcasted lectures, and multimedia conferencing sessions. Section 3.2.3 documents the collection and analysis of these traces.

3.2.1 VoIP Traffic Model

VoIP refers to real-time delivery of packet voice across networks using the Internet protocols. The rapid growth of IP-based packet switched networks and the overall bandwidth efficiency of an integrated IP network make it an attractive candidate to transport voice connections. In fact, multiplexing data and voice results in a better bandwidth utilization than the traditional circuit-switched voice-or-nothing backbone in the PSTN (Public Switched Telephone Networks), which consists of over-engineered voice trunks. This justifies looking at VoIP as a workload for future Internet packet networks.

With silence suppression, each VoIP source can be modeled as an on-off Markov process. The alternating periods of activity and silence are exponentially distributed with average durations of $1/\beta$ and $1/\alpha$, respectively. An exponential variable X has the following density function:

$$f_X(x) = \begin{cases} ae^{-ax} & x > 0;\ a > 0 \\ 0, & \text{otherwise} \end{cases}$$

where $E[X] = 1/a$ and $var[X] = 1/a^2$.

The fraction of time that the voice source is "on" is $\frac{\alpha}{\alpha+\beta}$. We consider an average talk spurt of 30.83% and average silence period of 61.47% as recommended by the ITU-T specification [76] for conversational speech. In all our experiments, we set $1/\beta$ and $1/\alpha$ to be 1.004 s and 1.587 s, respectively. When the source is in the "on" state, fixed-size packets are generated at a constant interval. No packets are transmitted when the source is "off". The size of the packet and the rate at which the packets are sent depends on the corresponding voice codecs and compression schemes.

Let $X_i(t)$ be the instantaneous rate of voice connection i:

$$X_i(t) = \begin{cases} R & \text{when the source is active} \\ 0 & \text{when the source is silent} \end{cases} \tag{3.1}$$

where R is the voice bit rate (i.e., packet size/packet interval). The rate of transition from the state of transmitting "0 Kbps" to the state of "R Kbps" is λ while the reverse transition happens at the rate of μ.

Traditionally voice is Pulse Code Modulated (PCM) [77, 78] at 64 Kbps in the PSTN. PCM provides high quality reproduction of speech and comparable quality can be maintained with ADPCM [79]. Recent advances in compression technology have allowed highly compressed speech (16 Kbps and lower) that offer excellent voice quality in the absence of packet losses. In our experiments, we assume that the voice source generates constant bit rate (CBR) traffic of 80 Kbps when it is "active". [1] We use this on-off Markov process to generate VoIP traffic in our simulations (EXP1 model in Section 3.3).

3.2.2 VoIP Performance Requirements

High quality interactive voice imposes many performance requirements on the underlying transport network. For example, one way end-to-end delay should be less than 150 ms to preserve the quality of interactive communication. In a circuit switched network, propagation delay is the only significant component in the one way end-to-end delay. In addition, this delay is constant component during the entire call duration, and therefore can be easily controlled. VoIP architecture, on the other hand, introduces new delay compo-

[1] Assume 8 KHz, 8 bits/sample PCM codec was used with 20 ms frame per packet. With 12 byte RTP header, 8 byte UDP header and 20 byte IP header, the size of each voice packet = 20 (header) + 160 (data) = 200 bytes. The bandwidth required will be (200 x 8) bits/20 ms = 80 Kbps.

nents such as: coding/decoding delay, packetization delay, queuing delays at intermediate routers/switches, and jitter compensation delay introduced by playout buffers. The multiplexing of VoIP and data traffic on shared links also introduces packet losses caused by buffer overflow at congested nodes. Latency and packet losses have adverse impact on the perceived voice quality, and therefore need to be bounded.

Our goal is to show how high quality voice can be supported with maximum utilization of resources if the network resource is provisioned properly and distributed admission control is implemented. To achieve this, we need to quantify the performance requirements of VoIP, by mapping the human perceived voice quality to the more tangible network centric parameters: packet loss and packet delay. Proper resource provisioning techniques can then be applied to provide statistical guarantees such as upper-bound for delay or loss profile.

Delay

In this dissertation, we ignore the delay introduced by the playout buffer. We also assume that:

- the end-to-end propagation delay is relatively constant and can be easily estimated,
- the sender uses the same codec throughout the call duration, and
- the sampling rate and packet size is fixed at the beginning of each call.

Since we are interested in investigating the effect of bandwidth allocation on voice quality, we try to segregate the effects of application-level QoS mechanisms. We assume that no application-level congestion control or rate adaptation are deployed at the voice sources. The only highly variable delay component in our model is queuing delay that occurs due to the multiplexing of voice packets, as well as the integration of voice and data over a shared link.

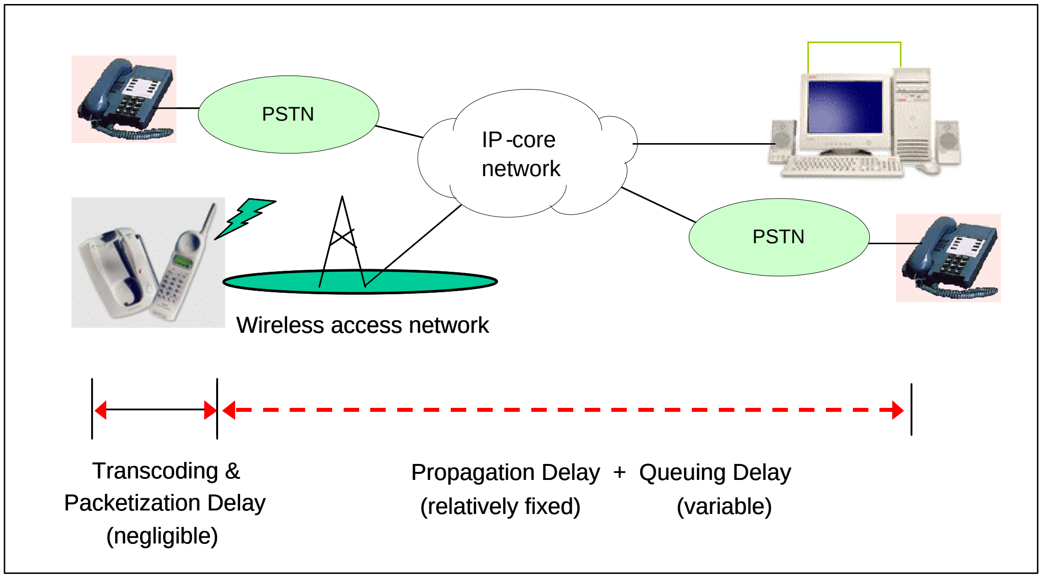

Figure 3.2: End-to-end delay components.

In our model, the end-to-end delay for VoIP are broken down to three components:

$$\text{packetization/transcoding} + \text{propagation} + \text{queuing delay},$$

as shown in Figure 3.2.

ITU-T Recommendation G. 114 [80] specifies that one-way transmission time for connections with adequately controlled echo should be in the 0-150 ms range to be acceptable for most user applications. We assume PCM transcoding introduces almost negligible delay if implemented in hardware (0.75 ms). The propagation delay is relatively constant and can be easily estimated. From [80], Public Land Mobile Systems contribute around 80 - 110 ms to one-way propagation time. Satellite systems introduce 12 ms at 1400 km altitude, and 110 ms at 14,000 km altitude. Optical fiber cable system contributes around 50-60 ms from coast to coast in the United States. Assuming it takes 100 ms propagation delay for voice packets to be transported across the United States, the total queuing delay should be kept within 50 ms (150ms - propagation delay). Since queuing delay is the only variable part in our model, we need to budget the per hop queuing delay. From about 50 traceroute[2] [81] experiments, we found out that there were typically around 8-12 hops between a machines on the west coast and the east coast. Assuming that queuing delay is almost the same for each hop, we require the per hop queuing delay to be *at most 5 ms* and use this upper-bound to choose appropriate buffer size.

Packet Loss

Packet losses can cause further distortion beyond the unavoidable loss of information introduced by speech encoding/decoding and therefore should be minimized. We consider packet losses that are caused by buffer overflows in routers as well as discarding of delayed packets in the receiver playout buffer (i.e., if packets arrive at the receiver after too long a delay and miss the playout time, these packets are discarded and therefore considered lost). The impact of packet loss on voice quality is dependent on the voice codec used.

In the Fall of 1998, we used Visual Audio Tool (vat) [82] to run a simple subjective test to *map the packet loss rate to perceived voice quality.* Vat is a multi-party audio conferencing tool enabled by IP-Multicast [83]. We consider the following case: PCM codec with silence suppression, 8 kHz sampling rate, 8 bits per sample (contributing to 64 Kbps when the source is active), and 20 ms of voice samples per packet.

Figure 3.3 shows the experimental setup for the subjective test. The sound files of three sentences (about 6 seconds each) from the movie, "A Few Good Men" were downloaded and converted to PCM format with 8 kHz sampling rate. Since these sound files are in WAV format, we used *sndrfmt* program to resample the voice at 8KHz and convert the format to PCM and saved as μ-law bytes. *sndrfmt*[3] is a sound utility program that

[2]Traceroute can be used to display the path taken by packets across network from one host to another host. This tool works by sending a series of UDP packets with different port numbers and TTL (Time To Live). A list of public servers that offer traceroute query service can be found at `http://www.traceroute.org/`.

[3]The complete "dpwelib" package that includes *sndrfmt* and many other utility programs can be downloaded from `http://www.icsi.berkeley.edu/~dpwe/dpwelib.html`.

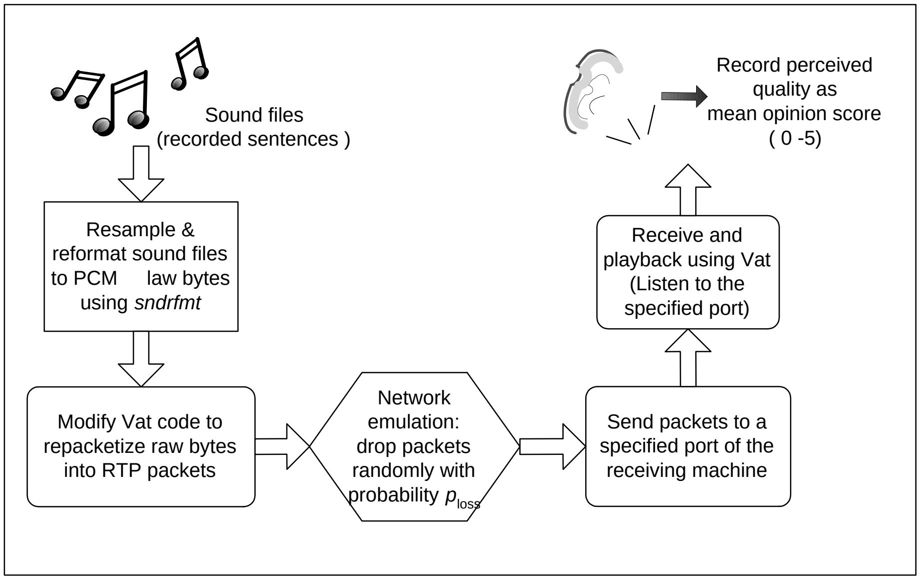

Figure 3.3: Experiment setup to carry out the subjective test that maps human perceived voice quality to different packet loss rates.

uses a library of audio hardware and sound-file access functions developed by Dan Ellis at the International Computer Science Institute (ICSI), Berkeley, CA. The voice samples were then packetized into RTP [5] packets with 12-byte RTP Header and sent through a simple network emulation that introduced random packet losses at different loss rates, p_{loss}. The packets are sent at every 20 ms interval. We ran vat at the receiving machine to listen to a specific port and playback the data. The perceived voice quality was scored on a numeric 0 to 5 scales with the following definitions: 5 = crystal clear, 4 = comprehensible but less clear; 3 = choppy speech; 2 = harder to comprehend sentences due to noise; 1 = can comprehend less than 50% of the sentence; 0 = gibberish noise. The same experiment was repeated for different p_{loss}, which was varied between 0 and 10%.

The result is plotted in Figure 3.4. Results show that the tolerable loss rates are within 1-2.5% and the speech becomes incomprehensible when more than 4% of the voice packets are lost. Note that packet voice using Forward Error Correction (FEC) [4] is more resilient to losses and therefore we would expect the curve to shift to the right in this case. On the other hand, the quality of voice connection using compressed speech is more

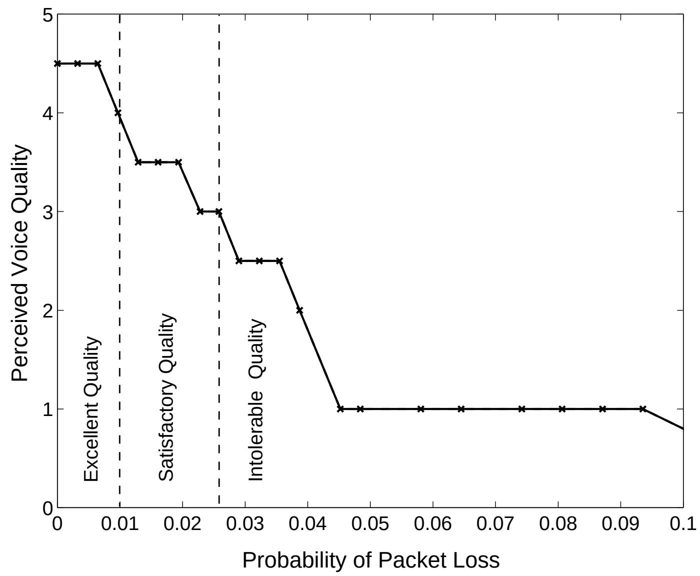

Figure 3.4: Subjective test results: how packet loss rate affect perceived voice quality.

sensitive to lost voice samples, and we expect the curve to shift to the left. The impact of packet loss on voice quality depends on the codec used, burstiness of losses, and frame sizes per packet, but this is out of scope of this project. For the rest of our analysis, we set the upper-bound packet loss rate to 1%, i.e., the QoS requirement is to send high-priority traffic from end-to-end with at most 1% loss rate.

3.2.3 Packet Audio Traces

To extend our analysis beyond VoIP, we collected 70 packet audio traces from a wide range of multimedia applications, including technical conference meetings, weekly lectures, technical demonstrations and social conversations. Based on these traces, we generate Internet workloads that have diverse characteristics to drive a subset of our simulation experiments.

These traces are classified into the following five categories according to their audio content:

Table 3.1: Summary of traffic traces.

Type	Number of Traces	Duration (minutes)	Voice Data (packets, MBytes)
Audience	32	(min) 1.26	616 pkt, 0.21 MB
		(max) 123.6	3747 pkt, 1.27 MB
Classroom lecture	11	(min) 4.4	6488 pkt, 2.21 MB
		(max) 71.8	100237 pkt, 34.1 MB
Conference call	26	(min) 0.5	528 pkt, 0.18 MB
		(max) 108.2	26819 pkt, 9.12 MB
Conversation	24	(min) 1.2	1553 pkt, 0.11 MB
		(max) 20.8	4287 pkt, 0.31 MB
Pre-recorded speech	1	6.6	9781 pkt, 3.33 MB

- Traces collected from speakers who were giving a lecture or leading a discussion are classified as *lecture in classroom type*, where the audience (other participants) may interrupt the speakers with questions, leading to occasionally long pauses in the speakers' voice stream.

- Traces that represent participants who remain silent most of the time except for occasional questions or technical discussions are classified as *audience type*.

- In a situation where the speakers were participating in a group discussion or multimedia conference calls, the traces may have longer silence periods such as time spent listening to other participants or looking at a shared media board. These traces are classified as *conference call type*.

- The traces from pre-recorded technical demonstrations and lecture are classified as *pre-recorded speech type*.

- Voice traces when two speakers were engaged in social conversations or technical discussions are classified as *conversation type*.

The traffic traces were generated by the following four sources and the breakdown of the traffic is tabulated in Table 3.1.

- **CSCW Electronic Classroom**

  58 traces were collected from a weekly Computer Science graduate-level class, Computer-Supported Cooperative Work (CSCW)[84] over 14 weeks in the Fall 1997. CSCW experimented with the idea of "Electronic Classroom" that was well-equipped with collaborative technology such as computers, video cameras, monitors, and a Xerox Live-Board. The class was held in a small, conference-style room. Some students would attend the course from their own office using remote collaboration tools (e.g., MASH tools like vic, vat and mb). 11 traces are classified as *classroom lecture*, 32 as *audience*, and 15 as *conference call*.

- **Research Groups' Multimedia Conferencing**

  11 traces were recorded from conference calls between professors, staff members, students and industrial sponsors of two research groups during January-September, 1998 and April-December 1999. All the traces are classified as *conference call*.

- **Pre-recorded Technical Demonstrations**

  We include in our analysis voice stream from pre-recorded technical demonstrations by graduate students, which we classified as *pre-recorded speech*.

- **CTS Test-bed with H.323 Gateway**

  24 traces were recorded from actual telephone conversations between students using the Computer Telephony Service (CTS) test-bed [85] from January-April 2000, where calls were made either from computer to computer, computer to normal PSTN phone or vice-versa via a H.323 Gateway.

All the participants in the CSCW class and multimedia conferencing communicate through three primary kinds of media: video, audio and shared white-board, using MASH [86] tools: vic, vat and "MediaBoard" (mb), respectively. These applications are launched on either Window-NT machines or Unix machines running Free-BSD. We are only interested in the voice packets recorded in these sessions/lecture.

Trace Processing

The voice traces were recorded according the MASH archive file formats [87, 88]. All data packets of one media type from a single source were stored in one file. Information such as the media type, the source identity, starting and ending time stamp were contained in the file header. The sender time stamp, receiver time stamp and sequence number of each packet were recorded. Voice packets were sent using RTP transmission format and 8 KHz 8 bits/sample PCM codec was used with 40 ms frame per packet. During the "talk" state, 340 bytes packets were generated every 40 ms (with 12 byte RTP, 8 byte UDP header and 320 bytes voice data).

We determined the talkspurt and silence periods by examining the interval between sender time stamps and locating gaps that were greater than 100 ms. Since the smallest meaningful element of speech, the phoneme, has an average size of 80-100 ms, we interpreted a pause smaller than 100 ms as a stop consonant or a minor break within the same talkspurt. We only ran statistical analysis on specific segments of the voice traces where actual conversations or lecture were in progress, and the rest of the traces were truncated. For example, a speaker sometimes had to restart his/her session because one of the tools (e.g., vic or "MediaBoard") failed to function. Although the voice packets were still recorded from the vat session, we truncated the packets recorded during the disruptions.

Section 3.3 will discuss how these traces are used in our simulation study.

3.3 Performance Evaluation

We have designed a new control architecture and resource provisioning mechanisms to deliver better QoS support to VoIP type workload outlined in previous section. The following three chapters (Chapter 4, 5, and 6) present the details of our proposed solutions and the design rationales behind them. To evaluate how well these mechanisms achieve our goals, we rely on a combination of simulation study and lab prototyping using both real-world and simulated topology. Besides network efficiency and end-to-end performance, we explore the architectural, scalability and practicality issues. It is also important to identify the degrees of freedom we have, e.g., the parameters that tune the several algorithms, and how they affect the trade-offs among contradicting performance goals, e.g., individual flow performance vs. overall network utilization.

3.3.1 Simulation Framework

Since it is infeasible to run large-scale experiments over actual wide-area networks, we resort to the following simulation experiments that capture the critical aspects of real-life Internet workloads and router technology:

Trace-based Simulation in C & Matlab

We developed a discrete-time event-driven C-simulator that implements the control logic of the Clearing House architecture and mechanisms. Two important inputs to the simulator are workload models and network topology. The arrival rate of the high-priority traffic is modeled as an independent Poisson process of intensity λ calls per second, and randomly pick from the pool of 70 traces (Section 3.2.3) to generate individual packet audio streams. We use the topology shown in Figure 3.5, which is an approximation of the AT&T WorldNet IP backbone as reported in [46].

With this simulator, we explored the efficiency and robustness of the CH-architecture in terms of resource utilization, call rejections and reservation setup time. The details of the experimental settings and simulation results are presented in Chapter 5. We also used Matlab to analyze the characteristics our proposed reservation scheme based on Gaussian traffic predictors.

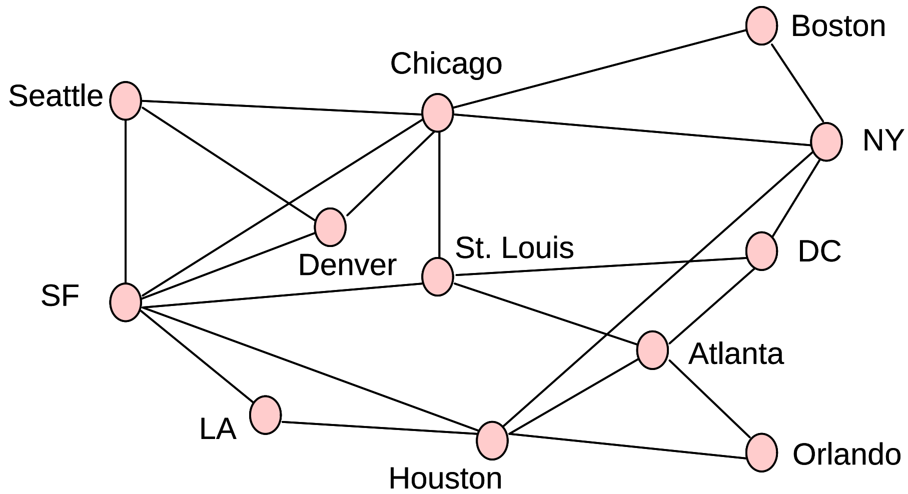

Figure 3.5: An example topology of a first-tier IP backbone network in the United States.

ns Simulation

To study the packet-level dynamics and further evaluate our system in extreme cases with diverse types of workload, we ran more experiments using the ns simulator [89]. We constructed an overlay network on top of ns-objects such as nodes and links (implemented in C++), and added session-level control at the tcl level. We added modules to generate and process control messages transmitted using the ns UDP/IP protocol stack.

To evaluate the robustness of our proposed mechanisms against the diversity of Internet workloads, we consider four kinds of traffic source models in our simulations: EXP1, EXP2, CBR and PARETO. Each of these models has its own distinct statistical properties and can be used to represent a variety of latency sensitive applications, as discussed in the following.

1. **EXP1** has exponential on and off times as described in Section 3.2.1 with an average of 1.004 s and 1.587 s, respectively. The peak transmission rate is 80 Kbps, and the average is approximately 31 Kbps. EXP1 can be used to model voice applications, e.g., VoIP and audio conferencing, which use silence suppression.

2. **EXP2** also has exponential on and off times, but with an average of 100 ms and 900 ms, respectively. The peak rate is increased to 310 Kbps while keeping the average rate the same as EXP1, leading to a burstier source. EXP2 generates the most bursty traffic among the four models that we consider and can be used to describe other workloads, such as video streams or multimedia conferencing applications, that have higher statistical variability than VoIP.

3. **CBR** is a constant bit rate source of 80 Kbps. Without silence suppression, packet voice streams can be represented using CBR model.

4. **PARETO** source has Pareto on and off times and has the same peak transmission rate (80 Kbps) as EXP1. A general Pareto density function is characterized by a shape parameter a and a scale parameter b:

$$f(x) = \frac{ab^a}{x^{a+1}} \text{ for } x \geq b.$$

   We set $a = 1.5$, and b is chosen such that the on and off times have the same average as EXP1 (1.004 s and 1.587 s, respectively). The aggregation of Pareto sources is known to exhibit long range dependencies [90, 91]. Hence, we use this model to describe interactive web applications that possess similar properties.

EXP1, EXP2 and CBR have exponential lifetimes with an average of 300s. The flow lifetimes of PARETO sources follow a log-normal distribution with average of 300 s.

We used ns to simulate different scenarios with these workloads to evaluate the admission control and malicious flow detection schemes. Results are documented in Chapter 6.

3.3.2 Lab Prototyping

We built a lab prototype of the Clearing House to evaluate certain performance metrics that could not be accurately quantified through simulations. One such metric is the overhead of implementing the various monitoring and policing mechanisms an edge router.

We extended the Click router [24] to support all the traffic policing and admission control functionalities of our architecture. Using this implementation, we measured the performance overhead incurred at an edge router, e.g., the degradation of system throughput. The current implementation works on Linux 2.2.16 and 2.2.17 kernels. Our architecture has been Operationally verified in our laboratory's test-bed. We will provide an overview of the implementation and performance measurements in Chapter 6.

3.4 Summary

This chapter gives an overview of our research methodology, including how we model the workloads of interest and how we evaluate the performance of our architecture through simulations and lab prototyping. The next three chapter document our technical contributions, design rationale and lessons learnt.