# Implementation of QoS-Provisioning System for Voice over IP

Shengquan Wang[†], Zhibin Mai[†], Walt Magnussen[‡], Dong Xuan[*], and Wei Zhao[†]
[†] Department of Computer Science
Texas A&M University
College Station, TX 77843
{swang, zbmai, zhao}@cs.tamu.edu

[‡] Internet2 Working Groups
Texas A&M University
College Station, TX 77843
w-magnussen@tamu.edu

[*] Department of Computer and Information Science
The Ohio State University
Columbus, OH 43210
xuan@cis.ohio-state.edu

**Abstract**

*In this paper, we address issues related to implementing Voice-over-IP (VoIP) services in packet switching networks. VoIP has been identified as a critical real-time application in the network QoS research community and has been implemented in commercial products. To provide competent quality of service for voice over ip network as the traditional telephone, the call admission control (CAC) mechanism has to be introduced to prevent packet losing and over-queuing. Several well designed CAC mechanisms, such as the Site-Utilization-Based CAC and the link-utilization-based CAC mechanisms, are in place. However, the existing commercial VoIP systems have not been able to adequately apply and support these CAC mechanisms, and hence unable to provide QoS guarantees to voice in IP networks. We have designed and implemented a QoS-Provisioning system that can be seamlessly integrated to the existing VoIP system to overcome its weakness in offering QoS guarantees. As a result, our system has been realized at Internet2 Voice Over IP Testbed in Texas A&M University.*

**Submission Category: Implementation**

## 1 Introduction

In this paper, we address issues related to implementing Voice over IP (VoIP) services in packet switching networks. VoIP has been identified as a critical real-time application in the network QoS research community. Transmission of voice traffic has to meet stringent requirements on packet delay as it is an important factor that affects the quality of calls. The International Telecommunication Union (ITU) recommends that a one-way delay between $0 - 150$ ms is acceptable in Recommendation G.114 [4].

In the traditional telephony, there is a call admission control mechanism. That is, when the number of call attempts exceeds the capacity of links, the request for setting up new calls will be rejected, while all calls in progress continue unaffected. Most current IP networks have no call admission control and hence can only offer best-effort services. That is, new traffic may keep entering the network even beyond the network capacity limit, consequently making both the existing and the new flows suffer packet loss and/or significant delay. To prevent these occurrences and provide QoS guarantees, a call admission control (CAC) mechanism has to be introduced

in IP networks in order to ensure that sufficient resources are available to satisfy the requirements of both the new and the existing calls after the new call has been admitted.

Current VoIP systems have noticed the importance of call admission control to provide QoS guarantees. Several call admission control (CAC) mechanisms, such as the *Site-Utilization-Based Call Admission Control* (SU-CAC) and the *Link-Utilization-Based Call Admission Control* (LU-CAC), have been used in the current VoIP systems. However, none of the current VoIP systems can really provide QoS guarantees to voice in IP networks. The basic reason behind this is that none of them are able to well apply and support the CAC mechanisms. For example, the SU-CAC mechanism performs admission control based on the pre-allocated resource to the *sites* [1]. It demands an approach to do resource pre-allocation to the sites at the configuration time. Unfortunately, the current VoIP systems, such as the Cisco's VoIP system [8], have not been able to define such an approach. Resource pre-allocation in these systems is performed in an ad hoc fashion. Hence, no QoS can be guaranteed although the SU-CAC mechanism is applied. Another example is the case of the LU-CAC mechanism. With the LU-CAC mechanism, admission control is based on the utilization of the individual link bandwidth. This mechanism needs resource reservation on the individual links in the network. The current VoIP systems rely on the resource reservation protocols, such as RSVP, to do explicit resource reservation on all routers along the path of the traffic in the network. Such a resource reservation approach will introduce the significant overhead to the core-routers, and hence greatly comprise the overall network performance.

In this paper, we will discuss our work on design and implementation of a QoS-Provisioning system. The QoS-Provisioning system can be integrated seamlessly to the existing commercial VoIP systems to overcome their weakness in offering QoS guarantees. We have successfully realized our system in *Internet2 Voice Over IP Testbed* in Texas A&M University. During the process of realizing this system, we have to deal with the following challenging problems:

- *How to provide end-to-end delay guarantees:* The primary focus will be to design a delay analysis method that does not depend on the information about flow population as this kind of information is unavailable in the environment that uses SU-CAC and LU-CAC mechanisms. We adopt the utilization-based delay-guarantee technique that is not flow-population sensitive.

- *How to optimize the overall utilization of network resource:* This will have to be achieved while end-to-end delays are guaranteed. We use optimization modules that can provide a parameter setting that maximizes the utilizations of network resource.

- *How to seamlessly integrate the QoS-Provisioning system to current commercial VoIP systems:* The integration should be transparent to the current VoIP system, and the system should not introduce too much overhead. We will report the lesson learned in our effort to integrate the proposed QoS-Provisioning system with an existing VoIP system that are based on Cisco CallManagers and Gatekeepers.

We systematically evaluate our proposed QoS-Provisioning system in terms of admission delay and admission probability. Our data show that if a VoIP system is enhanced by our QoS-Provisioning system, the overall system can achieve high resource utilization while invoking relatively invisible overhead.

The rest of the paper is organized as follows: In Section 2 we will introduce the current VoIP systems. We will describe the architecture of our designed QoS-Provisioning system in Section 3. In Section 4 and Section 5, we will focus on Call Admission Control Module (CACA) – the main component of the QoS-Provisioning system and describe the algorithm and signaling processing in CACA. In Section 6, we will illustrate the performance with extensive experimental data. A summary of this paper will be given in Section 7.

[1] The site can represent a host or a network with different sizes.

## 2 Background

VoIP system is rapidly gaining acceptance. Currently, some of the leading vendors have made announcements about their strategies and product directions for the system. Some VoIP systems, such as Cisco's and Alcatel's VoIP systems have been put on the market. However, none of these systems can provide the end-to-end QoS guarantees to voice in IP networks. In the following, we would like to take Ciscos VoIP system as an example to briefly introduce the QoS architecture of the commercial VoIP systems, and illustrate why current VoIP systems cannot provide the QoS guarantees.

VoIP is a key part of Cisco's AVVID (Architecture for Voice, Video and Integrated Data) framework for multi-service networking [6]. The system aims to provide the certain degree of QoS to voice in IP networks. As we know, the QoS architecture includes two planes: *data plane* and *control plane*. The *data plane* is responsible for packet forwarding, while control plane is for resource management and admission control. Most effort in providing QoS guarantees for VoIP focuses on control plane, and no specific packet forwarding mechanism is defined in its data plane.

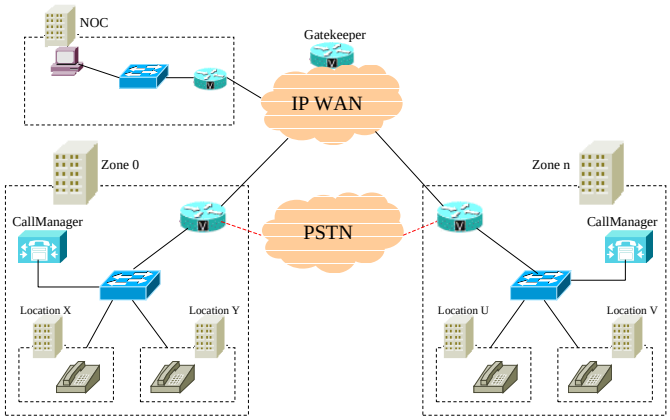



**Figure 1. An Illustration of a Typical Multi-site Control-Plane Architecture in Cisco VoIP System.**

Figure 1 illustrates a typical multi-site control-plane architecture of the system. CallManager (CM) [2] is the main component in the architecture. CallManager is a software-based call-processing component. It provides the overall framework for communication within a corporate enterprise environment. Gatekeeper (GK) is an optional component. Gatekeeper can provide services such as address translation and call admission control to the calls. It can be configured to work with CallManagers to do admission control. These two components communicate with each other by using the H.323 signaling protocol [2, 7].

CallManager as well as Gatekeeper performs admission control for calls inside or outside a corporate enterprise environment, aiming to provide a certain degree of QoS to voice in IP networks. To the call within a corporate enterprise environment, only the CallManager located in the enterprise environment is invoked to perform admission control. However, to the call cross through multiple corporate enterprise environments, not only CallManagers (both in the environment where the call is originated and in the one where the call is terminated), but also the related Gatekeeper(s) may be involved to do admission control.

Several call admission control mechanisms, such as the Site-Utilization-Based Call Admission Control (SU-CAC) and the Link-Utilization-Based Call Admission Control (LU-CAC), have been adopted by the current Cisco's VoIP system. The basic idea of SU-CAC is to do admission control based on the bandwidth which is *pre-allocated* to the *sites*. Bandwidth pre-allocation is performed at the configuration time (*i.e.*, at off-line). A new

[2] In this paper, we refer CallManager to a single CallManager or CallManager cluster

arrival call can be admitted if there is enough bandwidth left for the related site, otherwise the call will be rejected. In this strategy, the *site* can be a *location* to the CallManager or a *zone* to the Gatekeeper. A *location* defines a topological area connected to other areas by links with limited bandwidth registered to a CallManager. A *zone* is a collection of H.323 endpoints [3] that register to the same gatekeeper. The core of SU-CAC is how to do bandwidth pre-allocation to the sites. Bandwidth pre-allocation (or provisioning) determines the certainty of QoS that a VoIP system can provide to voice in IP networks. Unfortunately, so far, the Ciscos VoIP system does not define a proper way to do that. Currently, bandwidth pre-allocation is performed in a very ad-hoc manner. As a matter of fact, it is the reason why the end-to-end QoS guarantees cannot be achieved in the current Cisco's VoIP system.

The main advantage of SU-CAC is simple, and the admission control can be performed in a fully distributed fashion. It neither sends probes to test the availability of resources nor dispatches messages to make reservations. However, since the bandwidth has been pre-allocated to the sites at the configuration time, links cannot be fully shared by dynamic calls, and accordingly the high network resource [4] utilization cannot be achieved. The Link-Utilization-Based Call Admission Control (LU-CAC) aims to address this issue. The main idea of LU-CAC is to do admission control directly based on availability of the individual link bandwidth. With this mechanism, call multiplexing can be performed at the link level, hence the high network resource utilization can be obtained. The disadvantage of LU-CAC is its complexity. The current Cisco VoIP system has to rely on the resource reservation protocols, such as RSVP to do explicit resource reservation within the whole network. To achieve that, all the routers within the network should support resource reservation, which is not practical. Also in the current high speed network, there are potentially thousands of flows passing through the core-routers. The overhead of the core-routers within the network to support resource reservation is large. The overhead of resource reservation at the core-routers will compromise their main function, i.e., packet forwarding, which will degrade the whole network performance.

In summary, the current Cisco VoIP has given a basic framework to provide certain degree of QoS to voice in IP networks. The main reason it cannot provide the end-to-end guarantees is that the two admission control mechanisms, i.e., SU-CAC and LU-CAC, are not well applied and supported in the current VoIP system. In this study, we will design a QoS-provisioning system to enable the current system to well apply and support these two CAC mechanisms, aiming to provide the end-to-end guarantees for VoIP.

# 3 Architecture of QoS-Provisioning System

## 3.1 Design Rational

The goal of this study is to design and implement a practical, scalable and high efficient VoIP system that can provide the end-to-end QoS guarantees to voice in IP networks. Our main strategies to achieve this goal are listed as follows:

- We decide to accomplish our target system by enhancing the current Cisco VoIP system, rather than to build up a totally new VoIP system from scratch. We plan to enhance the current Cisco VoIP system by integrating a new QoS-provisioning system to enable both the SU-CAC and LU-CAC to be well utilized and supported. With this system, the overhead of resource reservation at the core routers will be pushed to the agents in the QoS-provisioning system, which overcomes the weakness of the current VoIP system in applying the LU-CAC mechanism. We will also address the issue that how to do resource allocation to well support the SU-CAC mechanism.

- We leverage our research results on absolute differentiated services in static-priority scheduling networks to provide *scalable* QoS guarantees to the VoIP system. In our previous research work, we derived a novel

[3] An H.323 endpoint can be a H.323 terminal, a gateway or a CallManager which represent a corporate enterprise environment.

[4] In this paper, the network resource is mainly referred to the link bandwidth.

flow-population-insensitive delay analysis formula that does not depend on the dynamic flow information. With this formula, in static-priority scheduling networks, the run-time overhead to do admission control is moved to the configuration time while sustaining scalability in the data plane. Recall that the Cisco VoIP has not defined a particular type of packet scheduler to be used in its data plane. This gives us room to select the proper scheduler for our purpose. We decide to use *Static Priority (SP)* scheduler in the data plane of the VoIP system. Particularly, all the voice traffic shares the same priority which is higher than the one used to transmit the non-voice traffic. In this way, our previous research results on absolute differentiated services in static-priority scheduling networks can be applied directly.

- We apply the linear programming approach to optimize the resource allocation in the control plane. As we know, the SU-CAC mechanism tends to under-utilize the network resource. Care must be taken to prevent wasting too much resource in applying this CAC mechanism. In this study, we will use the linear programming approach to optimize the resource utilization while still providing the end-to-end guarantees with SU-CAC mechanism.

In the following subsections, we will introduce the architecture of the QoS-Provisioning System and its components. Then, we will address how to conduct admission control and signaling processing within this architecture.

## 3.2 Architecture

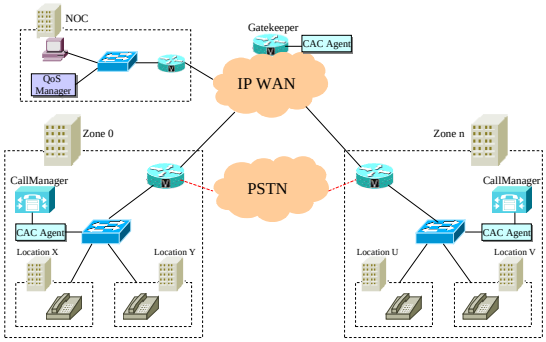



**Figure 2. The System Architecture.**

In this subsection, we describe the architecture of the QoS-Provisioning system. This system is to be integrated to the commercial VoIP system as shown in Figure 2. It consists of two kinds of components (See Figure 3): One is the central manager, called *QoS Manager* (QoSM); the other is distributed agents, called *Call Adminssion Control Agent* (CACA). The main functions of these two components are as follows:

- QoS Manager (QoSM) — The QoSM implements three basic functions: 1) Providing user interface to control and monitor the components, which are in the same QoS domain [5]. 2) Providing registration to the distributed agents and coordination among the distributed agents in the same QoS domain. 3) Cooperating with the peer QoSMs that belong to other QoS domains.

- Call Admission Control Agent (CACA) — The CAC Agent implements three main functions: 1) Doing deterministic or statistic delay analysis and obtaining the bandwidth utilization. 2) Performing admission control with specific CAC mechanism. 3) Processing call signaling.

[5]We define a QoS domain that covers one or multiple ASs. In each QoS domain, we can deploy one QoS Manager and multiple distributed agents, which are registerred to the QoS Manager.

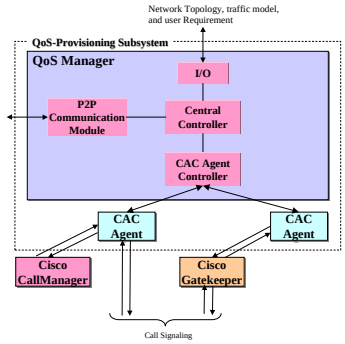



**Figure 3. The Components of the QoS-Provisioning System.**

As shown in Figure 3 and mentioned above, it is the CACA that does delay analysis, makes admission decision and intercepts call signaling, which are the key functions of the QoS-Provisioning system to provide QoS guarantee service for VoIP. Most of the challenging problems we encountered are in designing and implementing the CACA. Before going further to describe the CACA in detail, we would like to give a summary description about this agent. As shown in Figure 4, the CACA has four modules:

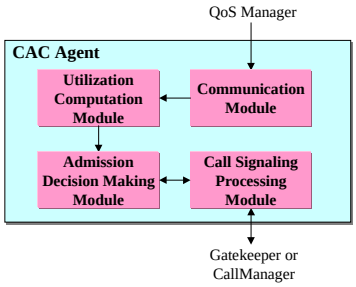



**Figure 4. The Architecture of Call Adminssion Control Agent (CACA).**

- *Communication Module:* It is used to communicate with QoS Manager. For example, the voice traffic model, network topology, and voice traffic deadline requirement can be downloaded from QoS Manager through the Communication Module.
- *Utilization Calculation Module:* It has two main functions. One of the functions is to compute the maximum link utilization. The other function is to optimize the utilization of resource to prevent wasting too much resource in applying SU-CAC mechanism.
- *Admission Decision Making Module:* It is used to make admission decision based on maximum link utilization and the incoming call information (i.e., bandwidth required and address of the caller/callee of the call etc.).
- *Call Signaling Processing Module:* It monitors and intercepts call setup signaling from Gatekeeper or CallManager, withdraws the useful message and passes it to *Admission Decision Making Module*, and executes call admission decision made by *Admission Decision Making Module*.

For the QoS-Provisioning system with SU-CAC mechanism, *Admission Decision Making Module* and *Call Signaling Processing Module* are optional. But *Utilization Calculation Module* must do bandwidth pre-allocation.

The pre-allocated bandwidth will be used as bandwidth limitation in SU-CAC mechanism, which has been in VoIP system.

In the next two sessions, we will describe in details the algorithms and signaling processing in CACA.

# 4 Algorithms in Call Admission Control Agent (CACA)

As we mentioned above, *Utilization Calculation Module* provides two main functions. One is to compute the maximum link utilization. With the maximum link utilization, the CAC mechanism (i.e., SU-CAC or LU-CAC) can be applied to make the link bandwidth usage below the maximum link utilization to provide end-to-end delay guarantees. The other is to optimize the utilization of resource for the SU-CAC mechanism. In this section, we will introduce two different algorithms that are used to achieve the two above functions.

## 4.1 Utilization Computation

*Utilization Computation Module* has a *utilization verfication* function as shown in Figure 5. Given the voice

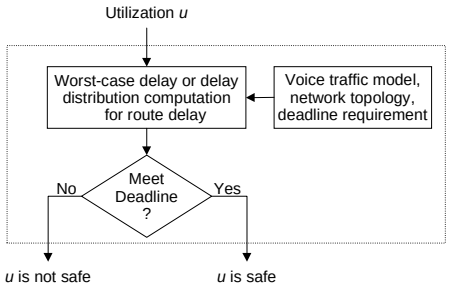



**Figure 5. Utilization Verification.**

traffic model, the network topology, and the voice traffic deadline requirement, for any input of link utilization $u$, we calculate the worst-case delay (deterministic case) or delay distribution (deterministic case) with our delay analysis methods. Then we can verify whether the utilization is safe or not to make end-to-end delay meet the deadline requirement. Using binary seaching method for utilization, we can obtain the maximum link utilization. As can be seen, the most challenging thing is to do delay analysis.

Genrally, there are two distinct types of delays suffered by a voice packet from source to destination: fixed and variable. Figure 6 identifies all the fixed and variable delay sources in the network:

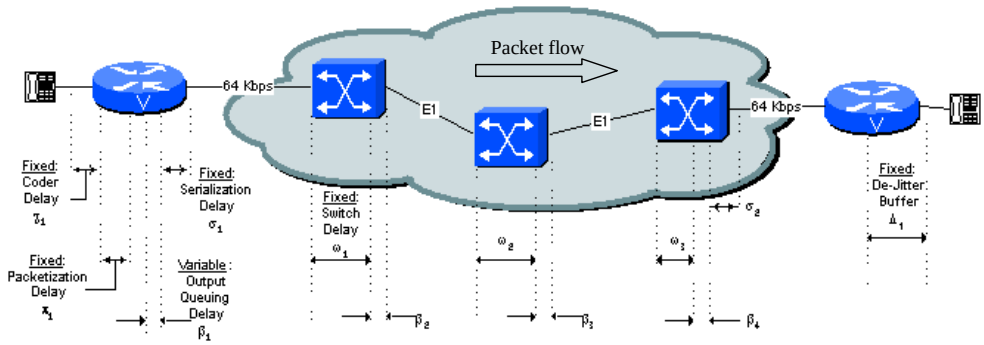



**Figure 6. Delay Sources.**

- Fixed delays cover: Coder (Processing) Delay ($\xi_n$); Packetization Delay ($\pi_n$); Serialization Delay ($\sigma_n$); Network Switching Delay ($\omega_n$); De-jitter Delay ($\Delta_n$).
- Variable delays arise from queuing delays ($\beta_n$) in the egress buffers. These buffers create variable delays, called jitter, across the network. Variable delays are handled via the de-jitter buffer at the receiving router/gateway.

Each source is described in detail in [3]. All fixed delays can be obtained by the well-known experimental data or by using suitable tools. However, it is difficult to obtain the variable delays. There are lots of research on queuing delay analysis [10, 9], either deterministic or statistical delay analysis method. In the following, we describe our deterministic delay analysis method.

Recall that we consider the VoIP system where static priority scheduling is used and voice traffic is assigned the highest priority. This scheduling does not provide flow separation, the local queuing delay at each output queue depends on detailed information (number and traffic characteristics) of other flows both at the output queue under consideration and at output queues upstream. Therefore, all the calls *currently* established in the network must be known in order to compute queuing delays. Delay formulas for this type of systems have been derived for a variety of scheduling algorithms. While such formulas could be used (at quite some expense) for flow establishment at system run time, they are not applicable for delay computation during configuration time, as they rely on information about flow population. In the absence of such information, the worst-case delays must be determined assuming a worst-case combination of flows.

Given the network topology (including link bandwidth information, potential path information), voice traffic pattern for each call (burst $\sigma$, average rate $\rho$), and link utilization $u_k$, we can bound worst-case delay as the following theorem [10]:

**Theorem 1** *The worst-case queuing delay $d_k$ suffered by any voice packet with highest priority at output link $k$ is bounded by*

$$\beta_k = \frac{c_k - 1}{c_k - u_k} u_k (\frac{\sigma}{\rho} + Y_k), \tag{1}$$

*for $c_k > 1$* [6]*, where*

$$c_k = \frac{\sum_j C_{j,k}}{C_k}, \tag{2}$$

$$Y_k = \max_{\mathcal{R} \in S_k} \sum_{s \in \mathcal{R}} \beta_s, \tag{3}$$

*$C_{j,k}$ and $C_k$ are the bandwidth of the $j$-th input link and the output link for Output queue $k$ repectively, and $S_k$ is the set of all sub-routes used by voice packet with highest priority* upstream from Server $k$.

Our delay analysis shows that under the given network topology and traffic model, the queuing delay at each output queue depends on link bandwidth utilization. By limiting the utilization of link bandwidth, the overall delay can be bounded.

Given the deadline requirement, for different link utilization $u_k$, with the delay analysis, *Utilization Computation Module* will verify whether the utilization is safe or not to make end-to-end delay meet the deadline requirement (fixed delays have been deducted). Using binary search method for utilization, the maximum link utilization can be obtained.

[6] If $c_k \leq 1$, then $d_k = 0$

### 4.2 Optimization of Bandwidth Utilization

In the QoS-Provisioning system with the SU-CAC mechanism, additional work must be done in the configuration time, *i.e.*, link bandwidth must be pre-allocated to sites. As we mentioned, SU-CAC mechanism tends to under-utilize the network resource while providing end-to-end delay guarantee. One of our objectives is to optimize the overall bandwidth utilization. As we mentioned, given the network topology and the limitation of link bandwidth allocated to voice traffic, to optimize the overall bandwidth utilization, an optimization problem can be defined as follows:

$$\text{Maximize} \quad \sum_{\mathcal{R}} B_{\mathcal{R}} \tag{4}$$

$$\text{Subject to} \quad \sum_{\mathcal{R} \in l} B_{\mathcal{R}} \leq B_l, \text{for each link } l; \tag{5}$$

$$B^0_{\mathcal{R}} \leq B_{\mathcal{R}} \leq B^1_{\mathcal{R}}, \text{for each route } \mathcal{R}; \tag{6}$$

where $B_l$ is the maximum bandwidth of link $l$ allocated to voice traffic (it is obtained in the above subsection), $\mathcal{R} \in l$ represents all routes among any pair of sites $\mathcal{R}$ goes through link $l$, $B_{\mathcal{R}}$ is the bandwidth for $\mathcal{R}$ allocated to voice traffic, and $B^0_{\mathcal{R}}$ and $\leq B^1_{\mathcal{R}}$ is the lower and upper bandwidth limit for $\mathcal{R}$ allocated to voice traffic.

In the above equations, (4) is the overall bandwidth utilization, (5) shows that link bandwidth pre-allocation is constrained by the link bandwidth limitation, and (6) is the user requirement for bandwidth pre-allocation to each pair of sites. This is a linear programming problem, which can be solved in polynomial time. The output, *i.e.*, the pre-allocated bandwidth, will be used as bandwidth limitation in the SU-CAC mechanism.

Once the pre-allocated bandwidth and the maximum bandwidth utilization are ready, they will be set as the overall bandwidth in the *bandwidth table* [7] in *Admission Decision Making Module* with the LU-CAC mechanism and the LU-CAC mechanism respectively. Based on the overall bandwidth and the bandwidth currently consumed, *Admission Decision Making Module* will make admission decision for the incoming call.

## 5 Signaling Processing and Admission Decision Making in CACA

In Section 4, we introduced two algorithms of Utilization Computation Module in CACA. In this section, we will describe two other modules in CACA: Signaling Processing Module and Admission Decision Making Module.

### 5.1 Signaling Processing in CACA

*Signaling Processing Module* is one of modules in CACA. It monitors and intercepts call setup signaling, withdraws useful messages (i.e. the bandwidth required by calls, the locations or addresses of callers and callees), passes them to admission decision making module, and executes call admission decision. Generally speaking, there are two approaches for this kind of module to intercept the call setup signaling and to execute the call admission decision:

- *Front-End approach:* In this approach, the call setup requests must *pass through* the agent before reaching the existing call admission decision unit (e.g., CallManager). The call setup responses must also *pass through* the agent before coming back to the call request endpoint. The agent can directly enforce its call admission decision to the call setup request by adding, modifying or dropping signaling between the endpoint and the existing admission decision unit. There are two basic methods to implement the agent in this approach: *proxy method* and *filter method*. To implement proxy method, the agent will be implemented

[7] It will be described in Subsection 5.2.

as a signaling proxy. The consistency of the call signaling between the end point and the existing call admission unit can be achieved. However, in most cases, it is complicated to implement a functional proxy, and the integration overhead can not be neglected since the integration is not transparent to the existing system. To implement the filter method, the agent only intercepts its interesting signaling and is easy to implement. Since the (filter) agent is treated as IP router or firewall by the existing system, the integration is transparent to the existing system. It may be difficult to achieve the call signaling consistency since there is not direct interaction between the (filter) agent and the existing system.

- *Back-End approach:* In this approach, the call setup requests and responses will be *forwarded* to the agent by the existing call admission decision unit (e.g., Gatekeeper). The agent will indirectly execute its call admission decision to the call setup request by negotiating with the existing call admission decision unit. The Back-End overcomes serveral problems of the Frond-End approach: 1) the implementation of the agent will not be very complicated since the existing system normally allows the agent to selectively receive and process the signaling; 2) the consistency of the signaling can be easily achieved since the agent directly interacts with call admission decision unit; 3) the integration overhead is little because only the existing admission control unit is aware of the agent. However, the Back-End approach requires the existing system to have the ability that the call setup requests and responses can be redirected to the external application, e.g., our Call Admission Control Agent (CACA), while the Front-End one does not. In most cases, the Back-End approach has more advantages than the Front-End approach.

Due to the different design and implementation methodology of Cisco Gatekeeper and Cisco CallManager, we adopt Back-End approach for Cisco Gatekpeeper and Front-End approach with filter method for Cisco CallManager in CACA. In the remaining of the section, we will describe in detail how the CACA, especially the Signaling Processing Module, works with the Cisco Gatekeeper and Cisco CallManager.

### 5.1.1 Signaling Processing Module for Cisco Gatekeeper

Cisco Gatekeeper is a built-in feature of Cisco IOS in some Cisco Router series (e.g., 2600, 3600 series) and is a lightweight H.323 gatekeeper. The RAS signaling that the Cisco Gatekeeper handles is H.323-compatible. Cisco Gatekeeper provides interface for external application servers to offload and supplement its features. The interaction between the Cisco Gatekeeper and the external application is completely transparent to the H.323 endpoint.

As shown in Figure 7, the Back-End approach is adopted for CACA to intercept the call signaling. The Signaling Processing Module handles the H.323 RAS signaling and communicates with Cisco IOS Gatekeeper. The communication between the Cisco IOS Gatekeeper and Signaling Processing Modules is based on Cisco‘s propriety protocol, Gatekeeper Transaction Message Protocol (GKTMP) [5]. GKTMP provides a set of ASCII RAS request/response messages between Cisco Gatekeeper and the external application over a TCP connection. There are two types of GKTMP messages:

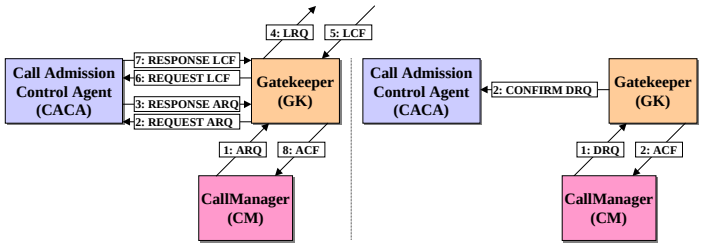
- *GKTMP RAS Messages:* It is used to exchange the contents RAS messages between the Cisco IOS Gatekeeper and the external application.

- *Trigger Registration Messages:* It is used by the external application to indicate to the Cisco Gatekeeper which RAS message should be forwarded.

If an external application is interested in receiving certain RAS messages, it must register this interest with the Cisco Gatekeeper.

In our implementation, the Signaling Processing Module is interested in receiving the following four RAS messsages from the Gatekeeper: Admission Request(ARQ), Location Confirm(LCF), Location Reject(LRJ) and

Disengage Request(DRQ). All of the four messages will be automatically registered to Cisco Gatekeeper once the CACA is up. Because of the space limit in this paper, we don't list out all the possibilities about how the Signaling Processing Module processes the RAS message. Figure 7(a) illustrates a successful call request procedure, which




(a) Call Request (b) Call Tearing-down

**Figure 7. An Illustation of a successful call request procedure in Call Adminssion Control Module for Gatekeeper.**

is described in detail in Figure 15 in the Appendix. Figure 7(b) illustrates a simple tearing down procedure, where CACA will update the status of network resource once receiving the message DRQ.

### 5.1.2 Signaling Processing Modules for Cisco CallManager

Cisco CallManager is comprehensive and heavyweight VoIP processing application, which runs on Windows 2000/NT platform. It can interact with endpoints using multiple protocols, e.g. Skinny Client Control Protocol(SCCP), H.323 and Session Initiation Protocol(SIP) etc. In this work, we implement SCCP, a popular sigaling protocol in Cisco VoIP system, in the CACA for CallManager.

To be best of our knowledge, Cisco CallManager does not provide interface for external application to supplement its call admission control mechanism as Gatekeeper does. In this case, only the Front-End approach can be adopted to intercept the Call Signaling of CallManager. As we mentioned above, there are two basic methods, proxy and filter, to implement Front-End approach. By the proxy method, the integration of the CACA to the current VoIP system is not transparent to the endpoints, the integration will affect thousands of the endpoints. The overhead and interruption caused by the integration to an operational environment is a realistic problem when using the proxy method. Considering that, we use the filter method in the current design and implementation of the CACA for CallManager. Figure 8 shows the basic idea of this method.

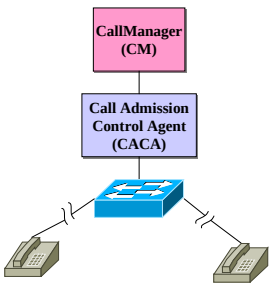



**Figure 8. An Illustation of Communication Protocol for Call Adminssion Control Module for CallManager.**

Since the basic idea and procedure of the signaling process in both CACA for CallManager and CACA for Gatekeeper is similar, we would like to highlight the difference in intercepting the SCCP using CACA for CallManager. The CallManager is unware of the CACA. It directly sends the implicit grant permission message (i.e. message "StartMediaTransmission") to the endpoints of the admitted call. However, in case the CACA makes a decision to deny the call because of the lack of available bandwidth, it should not let the message "StartMediaTransmission" be received by the endpoints. One of the approaches is to continue the message from the CallManager until the CallManager terminates the TCP connection after a finite timeout. There are two problems: 1) The caller does not get any indication whether the call is accepted or not. 2) The timeout is about 60seconds. To compensate the above two problems, CACA can explicitly indicate caller by sending the busy tone message, "StartToneMessage", to the endpoint and prevent the CallManager from sending a message to the endpoint by sending the call terminating message, "OnHookMessage", to the CallManager. However, additional messages from the Signaling Processing Module would interfere with the synchronization of the TCP connection between the endpoint and CallManager. To speedup the re-synchronization of the TCP connection and limit the impact on the CallManager, the Signaling Processing Module will send a TCP RESET packet to the endpoint in place of the CallManager.

## 5.2 Admission Decision Making in CACA

So far, we described the two algorithms in the *Utilization Calculation Module* and signaling processing in the *Call Signaling Processing Module*. We will continue to describe another important module in CACA: *Admission Decision Making Module*. The Admission Decision Making module supports both the SU-CAC mechanism [8] and the LU-CAC mechanism. In this subsection, we will describe the data structure and process of the two mechanisms in the Admission Decision Making module.

### 5.2.1 Admission Decision Making in SU-CAC Mechanisms

To support the SU-CAC mechanism, the Admission Decision Making module keeps neither the information about the overall bandwidth nor the available bandwidth for each individual link of the network. It takes a fixed amount of bandwidth for each pair of sites or a fixed total amount of bandwidth from/to a site, which is statically configured in *Bandwidth Table*. Note that the fixed bandwith is allocated by Utilization Computation Module in our QoS-provisioning system.

Table 1 shows an example of bandwidth table for pairs of sites. As each call is setup, the sites of source and destination can be known. If there is sufficient bandwidth left for the pair of sites, then the call is admitted and a certain amount of bandwidth is subtracted and will be returned to the pool when the call tears down. Otherwise, the call request is rejected.

| pair of sites | overall bandwidth | available bandwidth |
|---|---|---|
| $PairSites_1$ | 3.0 Mbps | 1.6 Mbps |
| $PairSites_2$ | 5.0 Mbps | 4.0 Mbps |
| $\vdots$ | $\vdots$ | $\vdots$ |
| $PairSites_R$ | 2.0 Mbps | 0.8 Mbps |

**Table 1. The bandwidth table in SU-CAC**

[8] Admission Decision Making module for the SU-CAC mechanism is an optional implementation in CACA since the SU-CAC mechanism in the current VoIP system can simply provide QoS guarantee by applying the result (i.e., resource allocation) from Utilization Computation Module.

### 5.2.2 Admission Decision Making in LU-CAC Mechanisms

To support the LU-CAC mechanism, the Admission Decision Making module has to keep the network topology information and the routing information. There are two tables in supporting this mechanism: *Bandwidth Table* and *Routing Table*. *Bandwidth Table* is used to keep the information of how much of the configured bandwidth on the links is currently consumed by voice traffic and how much link bandwidth is available for calls as shown in Table 2. Note that overall bandwidth in Table 2 is the maximun link utilization from the Utilization Computation module. The routing information can be found in the *Routing Table*.

| link | overall bandwidth | available bandwidth |
|---|---|---|
| $link_1$ | 40.0 Mbps | 21.0 Mbps |
| $link_2$ | 20.0 Mbps | 12.0 Mbps |
| $\vdots$ | $\vdots$ | $\vdots$ |
| $link_L$ | 35.0 Mbps | 15.0 Mbps |

**Table 2. The bandwidth table in LU-CAC**

Once a call request comes, each link along the call route will be checked to see if there is sufficient bandwidth left in the *Bandwidth Table*. First of all, the call route should be found in the *Routing Table* as shown in Table 3 with the source and destination of the call. If all links along the call route have sufficient bandwidth left, then CAC module will admit the call and decrease the available bandwidth of all call links by the requested bandwidth; otherwise, it will reject it. Once the call tears down, the bandwidth requested by the call will be returned to the pool for each link along the call route.

| source | destination | links |
|---|---|---|
| $src_1$ | $dst_1$ | $src_1 \rightarrow node_1^1 \rightarrow \cdots \rightarrow node_1^i \rightarrow \cdots \rightarrow dst_1$ |
| $src_2$ | $dst_2$ | $src_2 \rightarrow node_2^1 \rightarrow \cdots \rightarrow node_2^i \rightarrow \cdots \rightarrow dst_2$ |
| $\vdots$ | $\vdots$ | $\vdots$ |
| $src_R$ | $dst_R$ | $src_R \rightarrow node_R^1 \rightarrow \cdots \rightarrow node_R^i \rightarrow \cdots \rightarrow dst_R$ |

**Table 3. The routing table in LU-CAC (In the righ column, each arrow represent a link)**

# 6 Performance Evaluation

Two objectives are considered in this performance evaluation: 1) the introduced overhead to the admission; 2) the overall bandwidth utilization. Correspondingly, we choose two measurement metrics: *admission latency* and *admission probability*. *Admission latency* is used to measure the overhead of admission. *Admission probability* is the ratio of the number of admissions over the number of overall requests, which is a well-known metrix to measure the overall bandwidth utilization. Generally speaking, the higher the admission probability is, the higher the overall bandwidth utilization can be achieved.

## 6.1 Admission Latency

In this subsection, we run a suite of experiments to evaluate the *Admission latency* in two VoIP systems: 1) the one with our CACA; 2) the one without our designed CACA.

Due to the different design and implementation methodology of CACA for CallManager and Gatekeeper, we run two experiments for both of the cases. The experiments are run in the *Internet2 Voice Over IP Testbed* in Texas A&M University.

### 6.1.1 Call Admission Control Agent (CACA) for Cisco Gatekeeper

In the experiment, we tried 300 calls for each CAC mechanism. The call signaling crosses 2 Cisco CallMangers and 2 Cisco Gatekeepers from a Cisco IP phone in Texas A&M University to another IP phone in Indiana University.

To show the introduced overhead by our designed QoS-Provisioning system, we have two sets of data: local admission latency and round-trip admission latency.

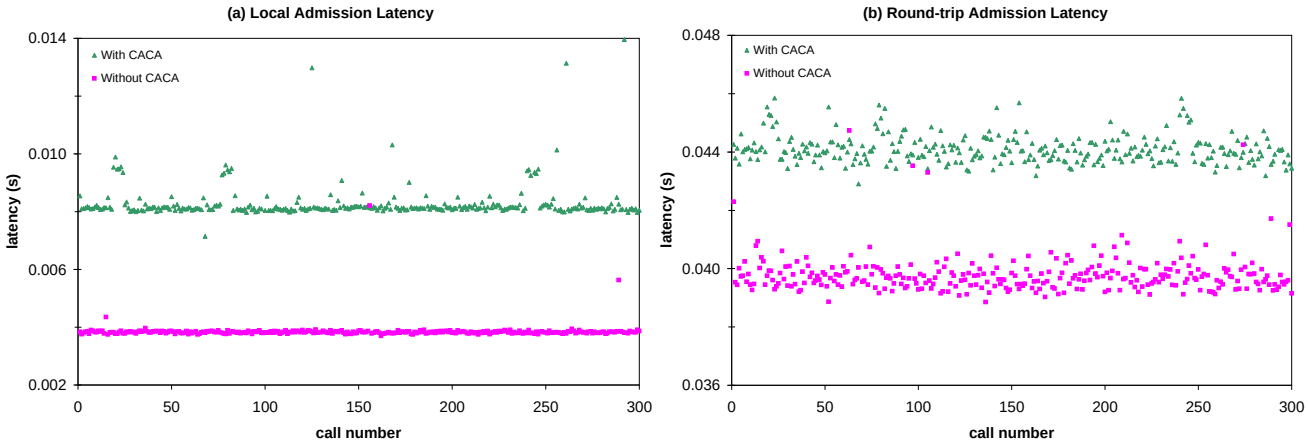



**Figure 9. The distribution of local and round-trip admission latency.**

| | local admission latency (ms) | | round-trip admission latency (ms) | |
|---|---|---|---|---|
| | mean value | standard deviation | mean value | standard deviation |
| with CACA | 8.286 | 0.863 | 44.302 | 2.665 |
| without CACA | 3.850 | 0.277 | 39.870 | 1.530 |

**Table 4. The mean value and standard deviation of latency distribution**

Figure 9(a) shows the distribution of local admission latency between receiving ARQ and sending out LRQ by Gatekeeper. Figure 9(b) shows the distribution of round-trip admission latency between receiving ARQ and sending ACF out by Gatekeeper. Table 4 gives us the summary of the distribution of admission latency for each case in term of the mean value and standard deviation.

The local admission latency excludes the network latency and the processing latency in the other side. It shows a more accurate latency introduced by our designed QoS-Provisioning system, which is shown by the standard deviation of the latency distribution in Table 4. The round-trip admission latency gives us the view of the overall admission latency.

The admission latency in the VoIP system with CACA is around 44.3 ms. The admission latency in the VoIP system without CACA is around 39.8 ms. With CACA, the introduced latency is about $8.3 - 3.9 = 4.4$ ms. The overall latency is very acceptable and the introduced latency is pretty small.

To measure the introduced latency, we measured the admission latency between receiving ARQ and sending out LRQ from the Gatekeeper, not the additional latency from the CACA directly. Here the additional admission latency includes not only the admission latency introduced in CACA, but also the additional latency in Gatekeeper caused by interaction between Gatekeeper and CACA, which can not be measured directly.

### 6.1.2 Call Admission Control Agent (CACA) for Cisco CallManager

In the experiment, we also tried 300 calls for each CAC mechanism. The call signaling crosses one Cisco CallManger between two Cisco IP phones in Texas A&M University.

To show the introduced overhead by our designed QoS-Provisioning system, we have two sets of data: local admission latency and round-trip admission latency.

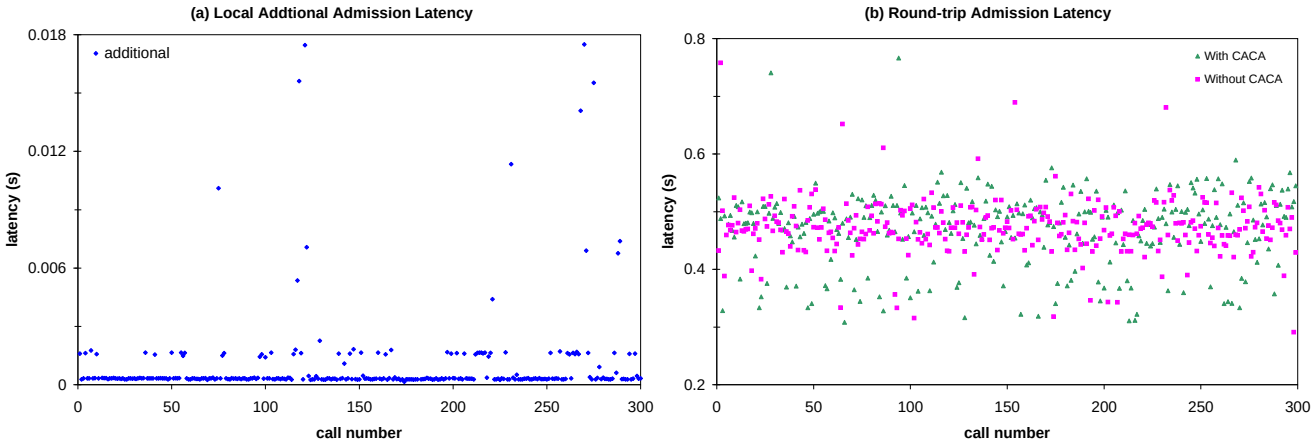



**Figure 10. The distribution of local additional and round-trip admission latency.**

| | latency (ms) | |
|---|---|---|
| | mean value | standard deviation |
| with CACA | 476.002 | 94.796 |
| without CACA | 479.367 | 92.114 |
| additional | 1.202 | 3.080 |

**Table 5. The mean value and standard deviation of latency distribution**

Figure 10(a) shows the distribution of local additional admission latency which is introduced by CACA in processing one call signaling message. Figure 10(b) shows the distribution of round-trip admission latency. Table 5 gives us the summary of the distribution of admission latency for each case in terms of the mean value and standard deviation.

The admission latency in the VoIP system with CACA is around 476.0 ms. The admission latency in the VoIP system without CACA is around 479.3 ms. With CACA, the introduced latency about 1.2 ms (*i.e.*, additional latency). The overall latency is very acceptable and the introduced latency is pretty small.[9]

[9] As can be seen in Table 5: 1) the average latency with CACA is less than the one without CACA. It is because the latency varies on the status of the network and VoIP system, and the additional latency introduced by CACA is a small proportion of the round-trip latency; 2) The average latency with CACA for CallManager is much larger than the one for Gatekeeper. It is because the SCCP signaling is more comprehensive than RAS signaling, and the call admission requires tens of round-trip SCCP messages between the CallManager and

## 6.2 Admission Probability

To make the data convincing, the measure of *admission probability* requires high volume of calls in VoIP system. However, it is not feasible or realistic to produce high volume of calls in VoIP system: Firstly, our designed QoS-Provisioning system is only deployed in *Internet2 Voice Over IP Testbed* in Texas A&M University, where simultaneous calls from lots of sites are not available; Secondly, even in the fully-deployed VoIP system, high volume of calls for experiment will affect the operation of VoIP heavily. *Admission probability* can only be measured by simulation. In this section, we run a suite of simulation to evaluate the *admission probability* for the LU-CAC mechanism and the SU-CAC mechanism respectively.

Traditionally, call arrivals follow a Poisson distribution and call lifetimes are exponentially distributed. This call mode can approximate the realistic call mode very well. In our simulation, we use this call mode to simulate calls by Mesquite CSIM 17 toolkits for simulation and modeling. In the simulation, overall requests for call establishment in the network form a Poisson process with rate $\lambda$, while call lifetimes are exponentially distributed with an average lifetime of $\mu = 180$ seconds for each call. All calls are duplex (bidirectional) and use G.711 codec, which has a fixed packet length of (160+40) bytes (RTP, UDP, IP headers and 2 voice frames) and a call flow rate of $80$ Kbps (including $64$ Kbps payload and other header).

Two different network topologies are chosen for the simulation: *Internet2 backbone network* and a *campus network*. Gatekeeper and CallManager are configed to perform call admission control in the *Internet2* environment and in the *campus* environment respectively.

### 6.2.1 Internet2 Backbone Network

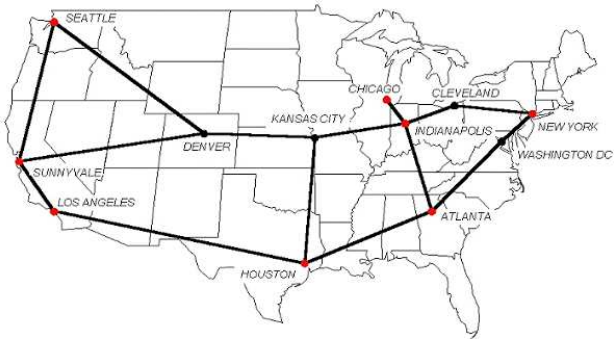



**Figure 11. The Abilene Network Topology (February 2002).**

Abilene is an advanced backbone network that supports the development and deployment of the new applications being developed within the Internet2 community. Figure 11 [1] shows the Abilene network topology (February 2002) used in our simulation. Currently, there are 12 core node routers, each located in a different geographical area. All backbone link are OC48 ($2.4$ Gbps) except that the link between SEATTLE and SUNNYYAL is OC12 ($622$ Mbps). The call route will be chosen uniformly randomly from the set of all pairs of core node routers. Suppose that the end-to-end deadline for queueing is $15$ ms. The maximum utilization is $0.209$, *i.e.*, under the condition that about $20.9\%$ link bandwidth is used for voice traffic, the end-to-end delay for any voice packet can meet the deadline requirement. Therefore, we choose this utilization for voice traffic. $\lambda$ changes from $10.0$ to $100.0$.

Figure 12 shows the admission probabilities for the voice call in the two CAC mechanism as a function of arrival rates. We find that the LU-CAC mechanism can achieve much higher admission probability than the SU-

endpoint for each call request.

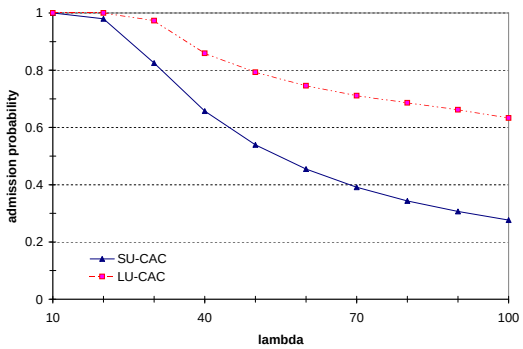



**Figure 12. The Admission Probability in Abilene Network.**

CAC, as we expected. As $\lambda = 30$, the difference is $97.3\% - 82.5\% = 14.8\%$; as $\lambda = 100$, the difference is $63.3\% - 27.7\% = 35.6\%$.

### 6.2.2 Campus Network

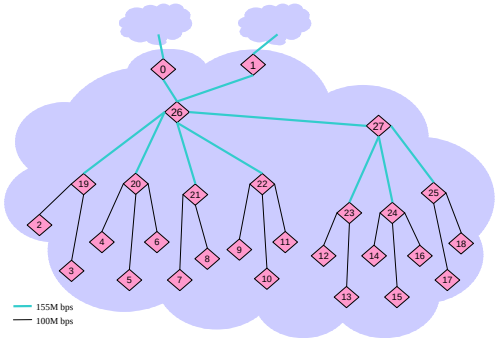



**Figure 13. A Campus Network Topology.**

Figure 13 shows the campus network topology used in our simulation. The link bandwidth is either $100$ Mbps or $150$ Mbps. The call route will be chosen uniformly randomly from the set of all pairs of sites $(0, 1, \ldots, 18)$. Suppose that the end-to-end deadline is $10$ ms for queueing. The maximum utilization is $0.151$. We choose this utilization for voice traffic. $\lambda$ changes from $1.0$ to $10.0$.

Figure 14 shows the admission probabilities for the voice call in the two CAC mechanism as a function of arrival rates. As can be seen, the admission probabilities in the two call admission control mechanism are different. We find that the LU-CAC mechanism can achieve much higher admission probability than the SU-CAC, as we expected. As $\lambda = 3$, the difference is $99.6\% - 64.2\% = 35.4\%$; as $\lambda = 100$, the difference is $66.9\% - 26.1\% = 40.8\%$.

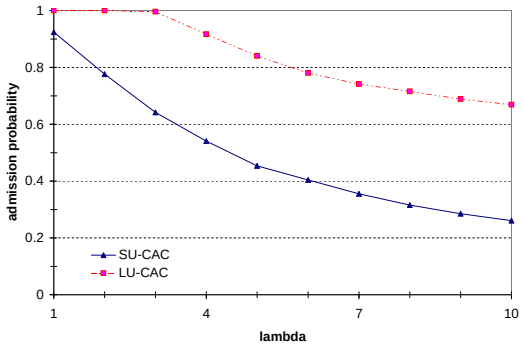



**Figure 14. The Admission Probability in Campus Network.**

# 7 Final Remarks

In this work, we designed and implemented a QoS-provisioning system that can be seemlessly integrated to the current Cisco VoIP system. This QoS-provisioning system has been successfully realized in *Internet2 Voice Over IP Testbed* in Texas A&M University.

The existing commercial VoIP systems only gives a basic framework to provide the certain degree of QoS. Several well designed CAC mechanisms, such as SU-CAC and LU-CAC mechanisms, are in place. We enhanced the current VoIP system by integrating a new QoS-provisioning system to make both two CAC mechanisms adequately utilized and supported. With this system, the overhead of resource reservation at the core routers will be pushed to the agents in the QoS-provisioning system, which overcomes the weakness of the current VoIP system in applying the LU-CAC mechanism. With this system, the resource utilization can be optimized while still providing the end-to-end guarantees with SU-CAC mechanism.

We leverage our research results on absolute differentiated services in static-priority scheduling networks to provide *scalable* QoS guarantees in the VoIP system. In our previous research work, we derived a novel flow-population-insensitive delay analysis formula that does not depend on the dynamic flow information. With this formula, in static-priority scheduling networks, the run-time overhead to do admission control is moved to the configuration time while sustaining scalability in the data plane. In this work, we realized the research results in the implementation of the QoS-Provisioning system that enhances the existing commercial VoIP system to provide QoS guarantees.

We systematically evaluated our proposed QoS-Provisioning system in terms of admission delay and admission probability. Our data show that if a VoIP system is enhanced by our QoS-Provisioning system, the overall system can achieve high resource utilization while invoking relatively invisible overhead.

# References

[1] Abilene network backbone. http://www.internet2.edu/abilene/html/maps.html.

[2] Registration, admission and status signaling (recommendation h.225/ras). International Telecommunication Union (ITU).

[3] Understanding delay in packet voice networks. http://www.cisco.com/warp/public/788/voip/delay-details.html.

[4] One-way transmission time (recommendation g.114). International Telecommunication Union (ITU), Febrary 1996.

[5] Gatekeeper external interface reference, version 3.1. Cisco Documentation, 2001.

[6] John Alexander and et. al. *Cisco CallManager Fundamentals*. Cisco Press, 2002.

[7] Daniel Collins. *Carrier Grade Voice over IP*. McGraw-Hill Professional Publishing, 2001.

[8] Jonathan Davidson, Tina Fox, and et. al. *Deploying Cisco Voice over IP Solutions*. Cisco Press, 2002.

[9] S. Wang, D. Xuan, R. Bettati, and W. Zhao. Differentiated services with statistical real-time guarantees in static-priority scheduling networks. In *Proceedings of IEEE Real-Time Systems Symposium*, December 2001.

[10] S. Wang, D. Xuan, R. Bettati, and W. Zhao. Providing absolute differentiated services for real-time applications in static-priority scheduling networks. In *Proceeding of IEEE International Conference on Computer Communications*, April 2001.

# Appendix

---

1. An CallManager as a H.323 gateway sends an ARQ message to Gatekeeper.
2. The Gatekeeper searches its trigger condition and find a match to the Call Admission Control Agent. It patches the message ARQ to message named “REQUEST ARQ” and sends it to the CACA .
3. The Call Signaling Module processes the “REQUEST ARQ”, e.g. recording down the bandwidth of the call, conference ID, address of the caller etc. and send back a “RESPONSE ARQ” to let Gatekeeper continue normal processing.
4. The Gatekeeper sends a LRQ to request address translation.
5. The Gatekeeper receives a message LCF, patches it to message “REQUEST LCF” and sends it to the CACA
6. The Call Signaling Module withdraws the address of the callee from the message “REQUEST LCF”. Plus the previous buffered information from message ARQ, it sends collected information to the Admission Decision Making Module.
7. The Admission Decision Making Module sends a confirmation to Call Signaling Module to accept the call if network resource is available. At the meantime, the Admission Decision Making Module updates the status of the network resource and the call.
8. The Call Signaling Module sends message “RESPONSE LCF” back to the Gatekeeper.
9. The Gatekeeper sends message ACF to grant call permission to the CallManager.

---

**Figure 15. A successful call request procedure in Call Adminssion Control Module for Gatekeeper.**