Design and Implementation of QoS-Provisioning System for Voice over IP

Shengquan Wang, Zhibin Mai, *Student Member*, *IEEE*, Dong Xuan, *Member*, *IEEE*, and Wei Zhao, *Fellow*, *IEEE*

Abstract—In this paper, we address issues in 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 VoIP systems comparable to traditional PSTN systems, a call admission control (CAC) mechanism has to be introduced to prevent packet loss and over-queuing. Several well-designed CAC mechanisms, such as the Site-Utilization-based CAC and the Link-Utilization-based CAC mechanisms have been in place. However, the existing commercial VoIP systems have not been able to adequately apply and support these CAC mechanisms and, hence, have been unable to provide QoS guarantees to voice over IP networks. We have designed and implemented a QoS-provisioning system that can be seamlessly integrated with the existing VoIP systems to overcome their weakness in offering QoS guarantees. A practical implementation of our QoS-provisioning system has been realized.

Index Terms—Voice over IP, real-time, delay, admission control.

1 INTRODUCTION

Voice over IP (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 [1].

In the traditional telephony, there is a call admission control (CAC) 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 CAC and hence can only offer best-effort services. That is, new traffic may keep entering the network even beyond the network capacity limit, consequently causing both the existing and the new flows to suffer packet loss and/or significant delay. To prevent these occurrences and provide QoS guarantees, a CAC mechanism has to be introduced in IP networks in order to ensure that sufficient resources are available to provide QoS guarantees for both new and existing calls after a new call has been admitted.

Current VoIP systems have noticed the importance of call admission control to provide QoS guarantees. Several 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 over 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 preallocated resource to the *sites*, which represents a host or a network with different sizes. It demands an approach to do resource preallocation to the sites at the configuration time.[1] Unfortunately, the current VoIP systems, such as the Cisco's VoIP system [2], have not been able to define such an approach. Resource preallocation 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 individual links in a network. The current VoIP systems rely on 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 significant overhead to the core-routers and, hence, greatly comprise the overall network performance.

In this paper, we will discuss our work on the design and implementation of a QoS-provisioning system. The QoS-provisioning system can be integrated seamlessly into the existing commercial VoIP systems to overcome their weaknesses in offering QoS guarantees. Our QoS-provisioning system supports two types of QoS guarantees: Deterministic services and statistical services. *Deterministic services* support applications that have stringent performance requirements for a service without delay bound violations. While they provide a very simple service model to the application, deterministic services, by their very

- S. Wang and W. Zhao are with the Department of Computer Science, Texas A&M University, College Station, TX 77843. E-mail: {swang, zhao}@cs.tamu.edu.
- Z. Mai is with the Institute of Scientific Computing, Texas A&M University, College Station, TX 77843. E-mail: zbmai@tamu.edu.
- D. Xuan is with the Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210. E-mail: xuan@cse.ohio-state.edu.

Manuscript received 21 May 2004; revised 6 Jan. 2005; accepted 18 May 2005; published online 25 Jan. 2006.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-0126-0504.

1. The configuration time is the time phase when the system performs configuration of system parameters such as bandwidth utilization.

1045-9219/06/$20.00 © 2006 IEEE Published by the IEEE Computer Society

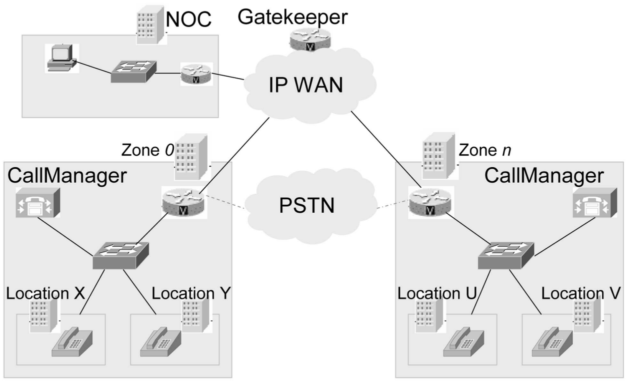

Fig. 1. An illustration of a typical multisite architecture in existing VoIP systems.

nature, tend to heavily over-commit resources because they reserve resources according to a worst-case scenario. *Statistical services*, on the other hand, exploit stochastic properties of traffic flows. They allow a predefined portion of packets to miss their deadlines, provide probabilistic performance guarantees and, therefore, significantly increase the efficiency of network usage by allowing increased statistical multiplexing of the underlying network resources. This comes at the expense of excessive packet delays.

We have successfully realized our system in the *Internet2 Voice Over IP Testbed* at Texas A&M University. 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 low overhead.

The rest of the paper is organized as follows: In Section 2, we will introduce a general architecture of VoIP systems. We will describe the design of a QoS-provisioning system in Section 3. In Section 4, we will focus on the design of Call Admission Control Agent (CACA), one of the main components in the QoS-provisioning system. We will discuss how to integrate our QoS-provisioning system with existing VoIP systems by Integration Component (IC) in Section 5. In Section 6, we will illustrate the performance of the system with extensive experimental data. The related work will be given in Section 7, followed by final remarks in Section 8.

2 A General Architecture of VoIP Systems

VoIP systems are rapidly gaining acceptance. Currently, some of the leading vendors have made announcements about their strategies and product directions for such systems. Some VoIP systems, such as Cisco's and Alcatel's VoIP systems, have been put on the market. The existing VoIP systems aim to provide a certain degree of QoS to voice over IP networks. However, none of these systems can provide end-to-end QoS guarantees. In the following, we introduce a general architecture of the existing commercial VoIP systems and illustrate why these VoIP systems cannot provide QoS guarantees.

Fig. 1 illustrates a typical multisite architecture of the VoIP system. CallManager and Gatekeeper are the main components in the architecture. CallManager provides the overall framework for communication within a corporate enterprise environment.[2] Gatekeeper can provide services such as address translation and call admission control to the calls. These two components communicate with each other by using the H.323 signaling protocol [3], [4]. In Fig. 1, 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.

CallManager as well as Gatekeeper performs admission control for calls between locations in a zone or calls between zones, aiming to provide a certain degree of QoS to voice over IP networks. To call within a zone, only the CallManager located in the enterprise environment is invoked to perform CAC. However, for a call traversing multiple zones, 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 perform CAC.

Utilization-based CAC is one type of CAC mechanisms. It makes an admission control decision based on a predefined utilization bound: For each call request, as long as the used resource utilization plus the requested resource utilization are not beyond the predefined resource utilization bound that is computed offline and set at the configuration time, the service guarantee can be provided. Two kinds of Utilization-based CAC mechanisms, *Site-Utilization-Based CAC* (SU-CAC) and *Link-Utilization-Based CAC* (LU-CAC), have been adopted by some VoIP systems.

The basic idea of SU-CAC is to perform CAC based on the bandwidth which is *preallocated* to the *sites*. In this strategy, the *site* can be a *location* to the CallManager or a *zone* to the Gatekeeper. Bandwidth preallocation to sites is performed at the configuration time (i.e., at offline). A new call can be admitted if there is enough bandwidth left for the related site, otherwise, the call will be rejected. The core of SU-CAC is how to do bandwidth preallocation to the sites. Bandwidth preallocation (or provisioning) determines the certainty of QoS that a VoIP system can provide to voice over IP networks. Unfortunately, so far, the existing VoIP systems do not define a proper way to do that. Currently, bandwidth preallocation is performed in an ad-hoc manner. As a matter of fact, this is the reason why the end-to-end QoS guarantees cannot be achieved in current VoIP systems. The main advantage of SU-CAC is that it is simple and it enables CAC to be performed in a distributed fashion. It neither sends probes to test the availability of resources nor dispatches messages to make reservations. However, since the bandwidth has been preallocated to the sites at the configuration time, links cannot be fully shared by dynamic calls and, accordingly, high network resource utilization cannot be achieved (in this paper, the network resource is mainly referred to the link bandwidth). The Link-Utilization-Based CAC (LU-CAC) aims to address this issue.

2. In this paper, we adopt Cisco CallManager term to refer to the component that manages calls within a corporate enterprise environment, which can be a proprietary product such as Cisco CallManger or a standard H.323 Gatekeeper.

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

The main idea of LU-CAC is to perform CAC directly based on availability of the individual link bandwidth. With this mechanism, call multiplexing can be performed at the link level, hence, high network resource utilization can be obtained. The disadvantage of LU-CAC is its complexity. Current VoIP systems have 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 current high speed networks, there are potentially thousands of flows passing through the core-routers. The overhead of the core-routers within the network to support resource reservation can be significantly 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.

3 Design of a QoS-Provisioning VoIP System

3.1 Design Rationale

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

- We build up our target system by enhancing the current existing VoIP systems, rather than building up a totally new VoIP system from scratch. The QoS architecture includes two planes: *data plane* and *control plane*. The *data plane* is responsible for packet scheduling and forwarding, while the *control plane* is for resource management and admission control. The simple utilization test in utilization-based CAC (both SU-CAC and LU-CAC) can render the *control plane* scalable. We plan to enhance current VoIP systems by integrating a new QoS-provisioning system to enable both 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 of performing resource allocation to better support the SU-CAC mechanism.
- We leverage our research results on absolute differentiated services in static-priority (SP) scheduling networks to provide *scalable* QoS guarantees to the VoIP system. The static-priority scheduler is *class-based*. Its overhead introduced to the *data plane* is small and independent from the flow population. The *data plane* is scalable with the static-priority scheduler. It is not the case for the guaranteed-rate schedulers, such as Weighted Fair Queuing (WFQ), which needs to maintain per flow information, hence making the *data plane* not scalable. However, the static-priority scheduler does not provide flow separation as the guaranteed-rate schedulers do. With the static-priority scheduler, the utilization-based admission control mechanism such as SU-CAC and LU-CAC cannot be directly applied to provide QoS guarantees. It cannot simply admit a new flow by checking the availability of utilization due to the flow interference with the static-priority scheduler. To account for such interference, the runtime overhead of admission control will be very large. In our previous research work, we derived a novel utilization-based delay analysis technique in static-priority scheduling networks. With this technique, the runtime overhead to do admission control is moved to the configuration time while sustaining scalability in the *data plane*. The existing VoIP systems do not specify 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 decided to use a static-priority 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 underutilize 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.

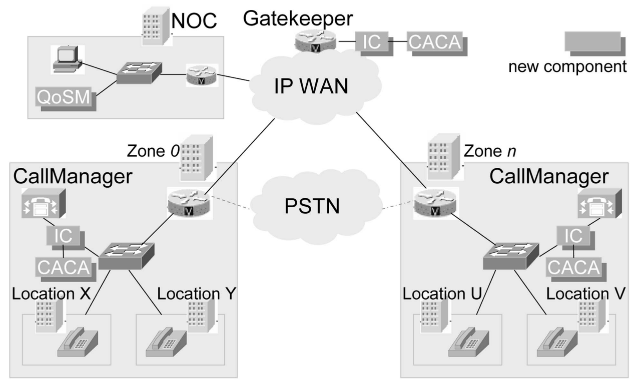

Fig. 2. The system architecture.

3.2 Architecture

Having discussed the design rationale of our QoS-provisioning system, in the following, we will introduce the architecture of this system and its components. Fig. 2 shows our QoS-provisioning system integrating with the commercial VoIP system. It consists of three kinds of components (see Fig. 3): *QoS Manager* (QoSM), *Call Admission Control Agent* (CACA), and *Integration Component* (IC). The main functions of these components are as follows:

- QoS Manager (QoSM)—The QoSM implements three basic functions: 1) It provides user interface to control and monitor the components, which are in

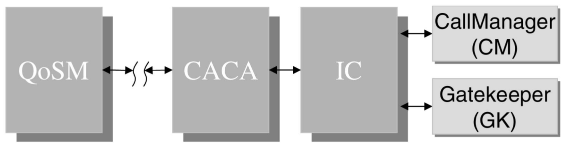

Fig. 3. The components of the QoS-provisioning system.

the same QoS domain.[4] 2) It provides registration to the distributed agents and coordination among the distributed agents in the same QoS domain. 3) It cooperates with the peer QoSMs that belong to other QoS domains.

- Call Admission Control Agent (CACA)—The CACA has two modules: 1) *Utilization Computation Module*, which performs deterministic or statistic delay analysis to obtain the maximum bandwidth utilization. 2) *Admission Decision Making Module*, which performs admission control with specific CAC mechanisms.
- Integration Component (IC)—The IC integrates CACA into existing VoIP systems and provides call signaling processing modules to monitor and intercept call setup signaling from Gatekeeper or CallManager, withdraws the useful message and passes it to CACA, and executes call admission decision made by CACA.

It is the CACA that does delay analysis and makes admission control, which are the main functions of the QoS-Provisioning system to provide QoS guarantees for VoIP. Most of the challenging problems we encountered are in designing and implementing CACA. We will devote Section 4 to this important component. Our QoS-provisioning system is not a stand-alone system. It should integrate with the existing VoIP system. Such integration is conducted through the Integration Component (IC) which will be discussed in Section 5.

4 Call Admission Control Agent (CACA)

The Call Admission Control Agent (CACA) is a key component in the QoS-provisioning system. It consists of two modules: *Utilization Computation Module* and *Admission Decision Making Module*. The utilization computation module performs delay analysis and computes the maximum bandwidth utilization. It usually runs at the configuration time. The computed utilization will be allocated to either links in the LU-CAC mechanism or to sites in the SU-CAC mechanism. At the runtime, the admission decision making module will make an admission decision for each incoming call request, based on the allocated bandwidth utilization (by the utilization computation module) and the currently consumed bandwidth. In the following, we discuss the details of these two modules.

4.1 Utilization Computation Module

The utilization computation module has two submodules: the *Link Utilization Computation Submodule* and the *Site Utilization Computation Submodule*. The first submodule is to compute the maximum link utilization for LU-CAC, while the second submodule is to compute the maximum site utilization for SU-CAC. The maximum link utilization is the maximum value of the link utilization under which the end-to-end delay can be guaranteed with LU-CAC. The maximum site utilization has a similar definition to SU-CAC. The above two submodules are closely related to each other. The maximum utilization allocated to sites is constrained by the maximum utilization allocated to the links since each pair of sites is connected by links. In the following, we will describe these submodules in detail.

4. We define a QoS domain that covers one or multiple ASes. In each QoS domain, we can deploy one QoS Manager and multiple distributed agents, which are registered to the QoS Manager.

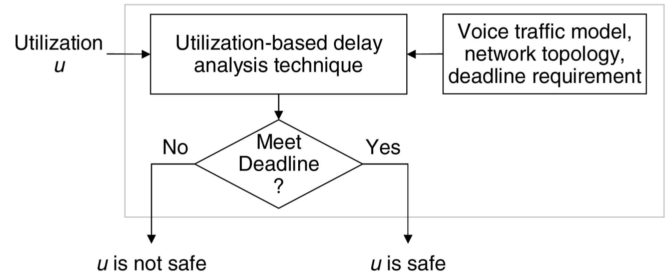

Fig. 4. The utilization verification procedure.

4.1.1 Link Utilization Computation Submodule

The main task of this submodule is to compute the maximum link utilization for LU-CAC by calling a procedure, named *the utilization verification procedure*. It is shown in Fig. 4.

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

Generally, there are two distinct types of delays suffered by a voice packet from source to destination: fixed and variable. Fixed delays include propagation delay, transmission delay, and so on. Variable delays arise from queuing delays in the output buffers. All fixed delays can be obtained by well-known experimental data or by using existing tools. However, it is difficult to obtain the variable delays. There is a significant amount of research on queuing delay analysis [5], [6] for both deterministic and statistical delay analysis.

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 the 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 system have been derived for a variety of scheduling algorithms. While such formulas could be used (at quite some expense) for flow establishment at the runtime, they are not applicable for delay computation during the configuration time, as they rely on information about flow population. In the absence of such information, the worst-case delays or deadline violation probabilities must be determined assuming a worst-case combination of flows.

We assume that the network topology is known in advance, which includes the potential end-to-end path information and link bandwidth information. Each voice traffic will be regulated by a leaky bucket with burst size σ and average rate ρ at the entrance of the network. Link k is of capacity C_k. The link bandwidth utilization allocated to voice traffic is assumed to be u_k at link k. Since the deadline requirement can be either deterministic or statistical, resulting in *deterministic services* and *statistical services*, we classify the delay analysis as *utilization-based deterministic delay analysis* and *utilization-based statistical delay analysis*, respectively.

Utilization-based Deterministic Delay Analysis: If the deadline requirement is deterministic, we can bound the worst-case queueing delay by the following theorem [5].

Theorem 1. *The worst-case queuing delay d_k suffered by any voice packet with the highest priority at the buffer of output link k is bounded by*[5]

$$d_k \leq \frac{c_k - 1}{c_k - u_k} u_k \left(\frac{\sigma}{\rho} + Y_k \right), \qquad (1)$$

where $c_k = (\sum_{j \in L_k} C_j)/C_k$, $Y_k = \max_{\mathcal{R} \in S_k} \sum_{s \in \mathcal{R}} d_s$, L_k is the set of the input links of output link k, and S_k is the set of all subroutes used by voice packets with the highest priority upstream from output link k.

The proof can be found in [5]. In (1), the value of Y_k, in turn, depends on the delays d_s's experienced at output link s other than k. Then, we have a circular dependency. Therefore, d_k can be determined iteratively. Furthermore, the end-to-end worst-case delay can be obtained, which only depends on the link utilization u_k, the parameters for voice traffic (burst size σ and average rate ρ), and the network topology. In some special cases, delay formulas independent of the network topology can be derived as shown in the Appendix.

Utilization-Based Statistical Delay Analysis: If the deadline requirement is probabilistic, we can bound delay violation probabilities as follows [6]:

Theorem 2. *In this case, d_k is a random variable and D_k is denoted as its deadline. The violation probability of delay for any voice packet with the highest priority suffered at the buffer of output link k is bounded by*

$$P\{d_k > D_k\} \leq \begin{cases} \frac{1}{\sqrt{2\pi}} \exp(-24 \frac{1-u_k}{{u_k}^2} \frac{D_k}{\frac{\sigma}{\rho}}), & u_k \geq \frac{D_k}{\frac{\sigma}{\rho}} \\ \frac{1}{\sqrt{2\pi}} \exp(-6 \frac{1-u_k}{{u_k}^3} (u_k + \frac{D_k}{\frac{\sigma}{\rho}})^2), & u_k < \frac{D_k}{\frac{\sigma}{\rho}}. \end{cases} \qquad (2)$$

The proof can be found in [6]. The end-to-end deadline violation probability can be bounded as

$$P\{d^{e2e} > \sum_{k \in \mathcal{R}} D_k\} \leq 1 - \prod_{k \in \mathcal{R}} (1 - P\{d_k > D_k\}), \qquad (3)$$

which only depends on the link utilization u_k, the parameters for voice traffic (the burst size σ and the average rate ρ), and the network topology.

Our utilization-based delay analysis techniques show that, under the given network topology and traffic model, the queuing delay or deadline violation probability at each output queue depends on link bandwidth utilization. By limiting the utilization of link bandwidth, the overall delay or deadline violation probability can be bounded. Given the deadline requirement, with the utilization-based delay analysis techniques, the maximum link utilization computation can obtain the maximum link utilization, which will be applied in the LU-CAC mechanism to perform admission control.

4.1.2 Site Utilization Computation Submodule

The main task of this submodule is to compute the maximum site utilization for SU-CAC. As we mentioned, the SU-CAC mechanism tends to underutilize the network resource while providing end-to-end delay guarantees. The objective in the maximum site utilization computation is to optimize the overall site bandwidth utilization. Our maximum link utilization computation will be based on the maximum link utilization computation and further splitting each maximum link utilization to pairs of sites that share this link. Given the network topology and the limitation of link bandwidth allocated to voice traffic, we can optimize the overall bandwidth utilization to sites, defined as follows:

$$\text{Maximize} \quad \sum_{\mathcal{R}} u_{\mathcal{R}} \qquad (4)$$

$$\text{Subject to} \quad \sum_{\mathcal{R} \in k} u_{\mathcal{R}} \leq u_k, \text{ for each link } k; \qquad (5)$$

$$u^0_{\mathcal{R}} \leq u_{\mathcal{R}} \leq u^1_{\mathcal{R}}, \text{ for each route } \mathcal{R}, \qquad (6)$$

where u_k is the maximum bandwidth of link k allocated to voice traffic (obtained in the above section), $\mathcal{R} \in k$ represents all routes among any pair of sites $\mathcal{R}$ going through link k, $u_{\mathcal{R}}$ is the bandwidth for $\mathcal{R}$ allocated to voice traffic, and $u^0_{\mathcal{R}}$ and $u^1_{\mathcal{R}}$ are the lower and upper bandwidth bounds for $\mathcal{R}$ allocated to voice traffic, respectively.

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

4.1.3 Discussions

The utilization computation module has three input parameters: 1) The network topology, which can be obtained by existing network management tools. 2) The worst-case delay bounds or deadline violation probability bounds, which are predefined according to the timing requirement of the voice traffic. 3) The burst size and the average rate for voice flow model, which are predefined according to the characteristics of the voice traffic. Except the network topology, all these parameters are fixed. When

5. If $c_k \leq 1$, then $d_k = 0$.

TABLE 1
The Routing Table in LU-CAS
(In the Right Column, Each Arrow Respresents a Link)

source	destination	links
src_1	dst_1	$src_1 \to \cdots \to node_1^i \to \cdots \to dst_1$
$\vdots$	$\vdots$	$\vdots$
src_R	dst_R	$src_R \to \cdots \to node_R^i \to \cdots \to dst_R$

the network topology changes, our system will recompute the link/site bandwidth utilization. Generally speaking, the network topology is relatively stable and predictable compared with the dynamic traffic. Even in the Internet, the study in [7] shows that the majority of the routing paths are stable. Therefore, the utilization recomputation will not happen frequently. We can further reduce the frequency of such recomputations by using our delay formula (7), given in Appendix. This formula depends only on the network diameter. A change in the network topology does not necessarily cause a change in the network diameter. With this formula, the utilization recomputation happens much less frequently.

Once the maximum link or site bandwidth utilizations are obtained and set in the LU-CAC mechanism and the SU-CAC mechanism, respectively, the *Admission Decision Making Module* will make the admission decision for the incoming call based on the overall bandwidth and the currently consumed bandwidth, which we discuss in the following.

4.2 Admission Decision Making Module

The Admission Decision Making module supports both the LU-CAC mechanism and the SU-CAC mechanism.[6] In this section, we will describe the data structure and the working process of the two mechanisms in the Admission Decision Making module.

4.2.1 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: the *Bandwidth Table* and *Routing Table*. The *Bandwidth Table* is used to keep the information about 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 3. Note that overall bandwidth in Table 3 is the maximum link utilization from the Utilization Computation module. The routing information can be found in the *Routing Table*.

6. The 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 guarantees by applying the result from the Utilization Computation Module.

TABLE 2
The Bandwidth Table in SU-CAC

pair of sites	overall bandwidth	available bandwidth
$PairSites_1$	3.0 Mbps	1.6 Mbps
$\vdots$	$\vdots$	$\vdots$
$PairSites_R$	2.0 Mbps	0.8 Mbps

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 with the source and destination of the call should be found in the *Routing Table* as shown in Table 1. If all links along the call route have sufficient bandwidth left, then the 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.

4.2.2 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 the *Bandwidth Table*. Note that the fixed bandwidth is allocated by the Utilization Computation Module in our QoS-provisioning system.

Table 2 shows an example of a 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.

5 Integration with Existing VoIP Systems

5.1 Methodology in Integration

The Call Admission Control Agent (CACA) is integrated into the existing VoIP systems through the Integration

TABLE 3
The Bandwidth Table in LU-CAC

link	overall bandwidth	available bandwidth
$link_1$	40.0 Mbps	21.0 Mbps
$\vdots$	$\vdots$	$\vdots$
$link_L$	35.0 Mbps	15.0 Mbps

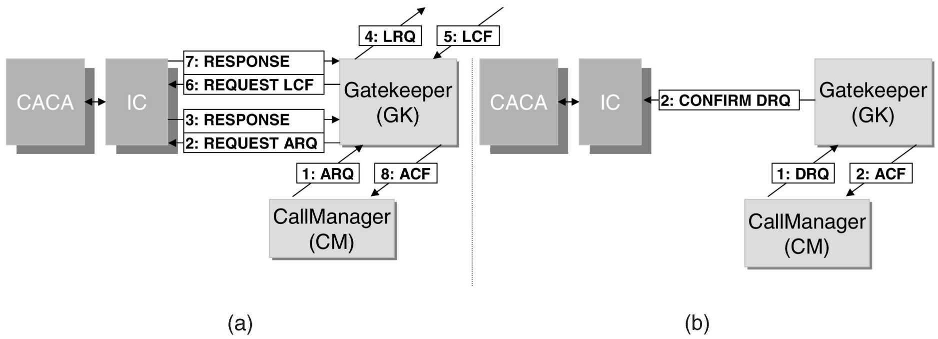

Fig. 5. An Illustration of a successful call request procedure in IC for Gatekeeper.

Component (IC). The Integration Component 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 the admission decision making module, and executes the call admission decision. Generally speaking, there are two approaches for this kind of component 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 as a signaling proxy. The consistency of the call signalings 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 cannot be neglected since the integration is not transparent to the existing system. To implement the filter method, the agent only intercepts its relevant 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 no 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 several 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 CACA, while the Front-End one does not. In most cases, the Back-End approach has more advantages than the Front-End approach.

5.2 Case Study

In this section, we would like to use the Cisco VoIP system to illustrate how our QoS provisioning subsystem can be integrated with the existing VoIP system. VoIP is a key part of Cisco's AVVID (Architecture for Voice, Video, and Integrated Data) framework for multiservice networking [8]. The architecture of the Cisco VoIP system is the same as the one shown in Fig. 1. It has two major components: Gatekeeper and CallManager. Due to the different design and implementation methodologies of these two components, we adopt the Back-End approach for Cisco Gatekeeper and the Front-End approach with the filter method for Cisco CallManager in the IC, respectively. In the remainder of the section, we will describe the details of how the integration component works with the Cisco Gatekeeper and Cisco CallManager.

5.2.1 Integration with 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 Fig. 5, the Back-End approach is adopted for the IC to intercept the call signaling between the CACA and the Gatekeeper. The IC handles the H.323 RAS signaling and communicates with the Cisco IOS Gatekeeper. The communication between the Cisco IOS Gatekeeper and the IC is based on Cisco's propriety protocol, Gatekeeper Transaction Message Protocol (GKTMP) [9]. 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: 1) *GKTMP RAS Messages:* It is used to exchange the contents of RAS messages between the Cisco IOS Gatekeeper and the external application. 2) *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

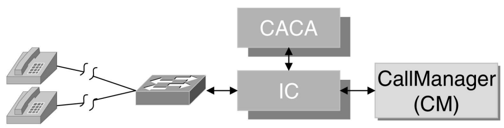

Fig. 6. An illustation of communication protocol in IC for CallManager.

in receiving certain RAS messages, it must register this interest with the Cisco Gatekeeper.

In our implementation, the IC is interested in receiving the following four RAS messages 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 functional. Due to space limitations, we do not list out all the possibilities of how the IC processes the RAS message. Fig. 5a illustrates a successful call request procedure. Fig. 5b illustrates a simple tearing down procedure, where the IC will update the status of network resource once receiving the message DRQ.

5.2.2 Integration with CallManager

Cisco CallManager is a comprehensive and heavyweight VoIP processing application, which runs on the 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 signaling protocol in Cisco VoIP system, in the IC for CallManager.

To the best of our knowledge, Cisco CallManager does not provide an interface for external applications to supplement its call admission control mechanism as a Gatekeeper does. In this case, only the Front-End approach can be adopted to intercept the Call Signaling of a CallManager. As we mentioned above, there are two basic methods, proxy and filter, to implement Front-End approach. By the proxy method, if 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 IC for CallManager. Fig. 6 shows the basic idea of this method.

Since the basic idea and the procedure of the signaling process in both the IC for CallManager and the IC for Gatekeeper are similar, we would like to highlight the difference in intercepting the SCCP using the IC for CallManager. The CallManager is unware of the IC. 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 dropping 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 60 seconds. To compensate for the above two problems, the IC 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 IC would interfere with the synchronization of the TCP connection between the endpoint and the CallManager. To speedup the resynchronization of the TCP connection, and limit the impact on the CallManager, the IC will send a TCP RESET packet to the endpoint in place of the CallManager.

6 Performance Evaluation

We consider two metrics in the performance evaluation: 1) the introduced overhead to the admission and 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 to the number of overall requests, which is a well-known metric to measure the overall bandwidth utilization. The higher is the admission probability, the higher is the overall bandwidth utilization achieved.

6.1 Admission Latency

In this section, we run a suite of experiments to evaluate the admission latency in two VoIP systems: 1) the one with our CACA and 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 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 two Cisco CallMangers and two 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.

Fig. 7a shows the distribution of local admission latency between receiving ARQ and sending out LRQ by Gatekeeper. Fig. 7b 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 terms of the mean value and standard deviation. The local admission latency excludes the network latency and the processing latency on 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.

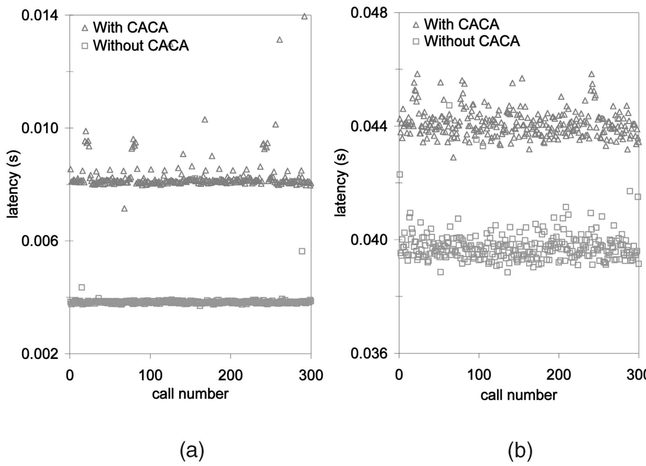

Fig. 7. The distribution of local and round-trip admission latency.

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 cannot be measured directly.

6.1.2 Call Admission Control Agent (CACA) for Cisco CallManager

In the experiment, we also initiated 300 calls for each CAC mechanism. The call signaling crosses one Cisco CallManager 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.

Fig. 8a shows the distribution of local additional admission latency which is introduced by CACA in processing one call signaling message. Fig. 8b 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 is about 1.2 ms (i.e., additional latency). The overall latency is acceptable and the introduced latency is quite small.[7]

7. 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 endpoint for each call request.

6.2 Admission Probability

To make the data convincing, the measure of admission probability requires a high volume of calls in the VoIP system. However, it is not feasible or realistic to produce a high volume of calls in the VoIP system: First, our designed QoS-provisioning system is only deployed in *Internet2 Voice Over IP Testbed* in Texas A&M University, where simultaneous calls from many sites are not available. Second, even in the fully-deployed VoIP system, a high volume of calls for the 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 19 toolkits [10] for simulation and modeling. In the simulation, overall requests for call establishment in the network form a Poisson process with rate λ, 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 two 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 configured to perform call admission control in the *Internet2* environment and in the *campus* environment, respectively.

6.2.1 Internet2 Backbone Network

Abilene is an advanced backbone network that supports the development and deployment of the new applications being developed within the Internet2 community. Fig. 9 [11] shows the core map of the Abilene network (September 2003) used in our simulation. There are 11 core node routers, each located in a different geographical area. All backbone links are either OC48 (2.4 Gbps) or OC192 (9.6 Gbps). 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 10 ms. The maximum utilization is 0.195 for deterministic service, i.e., under the condition that about 19.5 percent link bandwidth is used for voice traffic, the end-to-end delay for any voice packet can meet the deadline requirement. The maximum utilization is 0.307 for statistical service with deadline violation probability 10^{-6}, i.e., under the condition that about 30.7 percent link bandwidth is used for voice traffic, any voice packets may miss the deadline with small probability 10^{-6} at most. λ changes from 100.0 to 1000.0.

Fig. 10 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 a much higher admission probability than SU-CAC for both deterministic service and statistical service. Statistical service can achieve a much higher admission probability than deterministic service, as expected.

TABLE 4
The Mean Value and Standard Deviation of Latency Distribution

	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

6.2.2 Campus Network

Fig. 11 shows the campus network topology used in our simulation. The link bandwidth is either 100 Mbps or 155 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 0.208 for deterministic service, i.e., under the condition that about 20.8 percent link bandwidth is used for voice traffic, the end-to-end delay for any voice packet can meet the deadline requirement. The maximum utilization is 0.332 for statistical service with deadline violation probability 10^{-6}, i.e., under the condition that about 33.2 percent link bandwidth is used for voice traffic, any voice packets may miss the deadline with small probability 10^{-6} at most. λ changes from 1.0 to 10.0.

Fig. 12 shows the admission probabilities for the voice call in the two CAC mechanism as a function of arrival rates. The admission probabilities in the two call admission control mechanism are different. Similar to the observation made in Internet2 Backbone network, the LU-CAC mechanism can achieve much higher admission probability than SU-CAC for both deterministic service and statistical service. Statistical service can achieve a much higher admission probability than deterministic service, as expected.

7 RELATED WORK

Open standards to achieve interoperability between equipment of different vendors are critical to the success of VoIP. Most VoIP products implement one or more of the VoIP standards. Among well-known open VoIP standards are H.323 [12], SIP [15], MGCP [13], and Megaco/H.248 [14], where H.323 and SIP are peer-to-peer control-signaling protocols while MGCP and Megaco are master-slave control-signaling protocols. With the implementation of the existing VoIP standard signaling protocols in Call Signaling Processing Module, CACA of our system can interoperate with any existing VoIP products adopting open standards. Recall that, in the CACA architecture, the Call Signaling Processing Module provides generic call control APIs for the CAC Module. To support a new VoIP standard, only the Call Signaling Processing Module needs to be updated. The Utilization Calculation Module and CAC Module, which are the most complicated and error-prone modules in the CACA, can be intact. The modular design in CACA is very important for the QoS-provisioning system to be a reliable and time-to-market product.

In the past few years, VoIP has rapidly gained acceptance. Many leading vendors in the traditional voice and data industry have turned to provide VoIP solutions. In general, the VoIP vendors are classified into two camps in terms of the VoIP solutions. Traditional voice vendors, such as Nortel [16], Avaya [17], Alcatel [18], etc., with products based on circuit-switched technologies (Time Division Multiplexing), would like to adapt traditional circuit-switched voice solution to the VoIP solution. Meanwhile, data vendors, such as Cisco [2], and 3Com [19], would like to provide a pure VoIP solution, treating voice as just another data stream. To illustrate that our QoS-provisioning system can seamlessly integrate into any existing VoIP system, we would like to introduce one more VoIP system from Nortel Networks. In Nortel enterprise VoIP system,

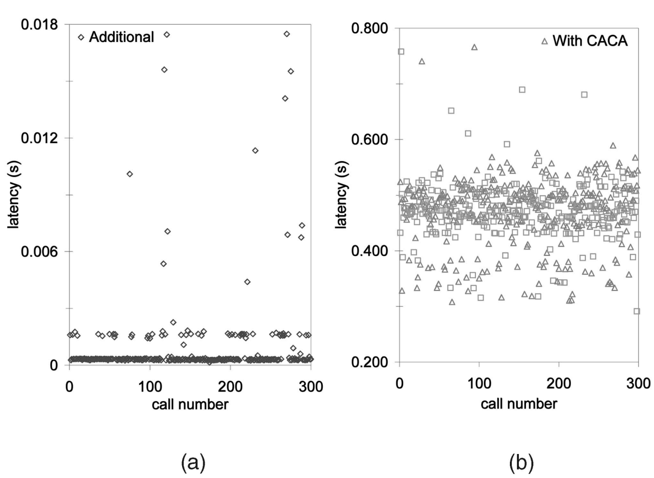

Fig. 8. The distribution of local and round-trip admission latency.

TABLE 5
The Mean Value and Standard Deviation of Latency Distribution

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

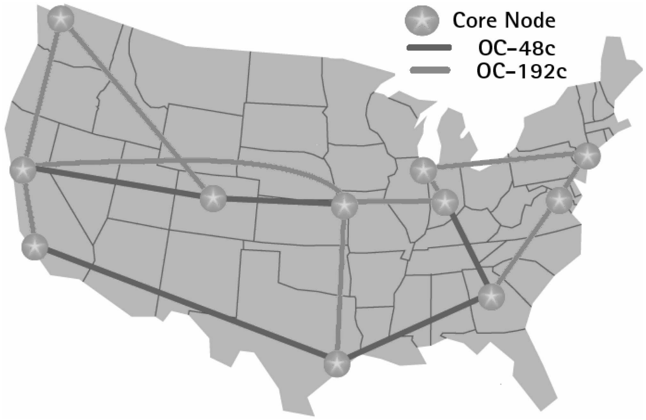

Fig. 9. The core map of Abilene network (September 2003).

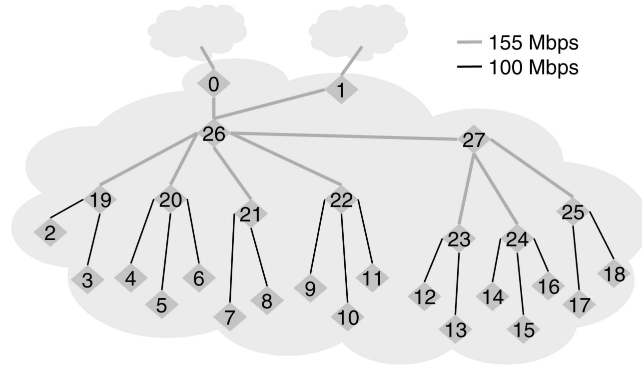

Fig. 11. A campus network topology.

called Succession Enterprise [16], [20], its core elements include call server, signaling server, media gateway. The call server is to provide call and connection management services for the IP network. The Signaling Server's H.323 Gateway provides an industry-standard H.323 signaling interface between Succession Enterprise systems across an enterprise WAN, or to H.323 gateways and PBXs that act as H.323 gateways. The Signaling Server also supports SIP standard. Nortel Succession Enterprise signaling server does not provide interface for external application to supplement its call admission control mechanism. In this case, the Front-End approach can be adopted to intercept the Call Signaling of the signaling server. As mentioned above, Succession Enterprise signaling server supports VoIP signaling open standards. With the implementation of VoIP signaling open standards in CACA, the CACA can be seamlessly integrated into Nortel enterprise VoIP system.

To provide end-to-end delay guarantees in VoIP systems, Call Admission Control (CAC) mechanisms have to be in place. CAC algorithms can be roughly grouped in two broad categories: 1) The *Measurement-based CAC* algorithm: It uses network measurement to estimate current load of existing traffic. It has no prior knowledge of the traffic statistics and makes admission decisions based on the current network state only [21], [22]. 2) The *Parameter-based CAC* algorithm: It uses the parameters of resource and service to decide whether the network can accommodate the new connection while providing the end-to-end delay guarantees. The parameters are used to compute a deterministic bound imposing that, in whichever traffic situation, the end-to-end delay guarantees are provided for all flows [5], [23], [24]. Utilization-based CAC (such as SU-CAC and LU-CAC used in this paper) belongs to this category, where the parameters are the requested bandwidth utilization for each new connection and the available bandwidth utilization in the resource [5], [23]. Through appropriate system (re)configuration steps, the delay guarantee test at run time is reduced to a simple utilization-based test: As long as the utilization of links along the path of a flow is not beyond a given bound, the performance guarantee of the end-to-end delay can be met. Utilization-based CAC renders the system scalable.

There has been some research work recently focusing on different issues in VoIP systems. In [25], the authors intended to determine the best method to implement VoIP in a large college campus, given vintage and overburdened PBX equipment. The plan was to study information available in current literature, analyze vendor offerings, compare real-life VoIP implementations, and make a recommendation. It focused on the existing systems and did not address how to extend them to provide QoS-provisioning features as this paper does. In [26], the authors assessed the ability of Internet backbones to support voice communication. They compared and combined results from various subjective testing studies and an approach to

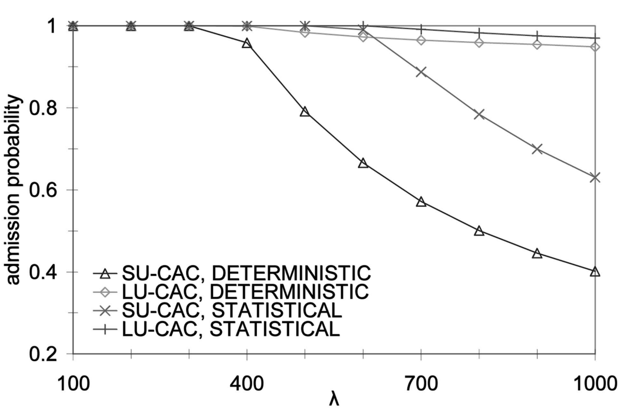

Fig. 10. The admission probability in Abilene network.

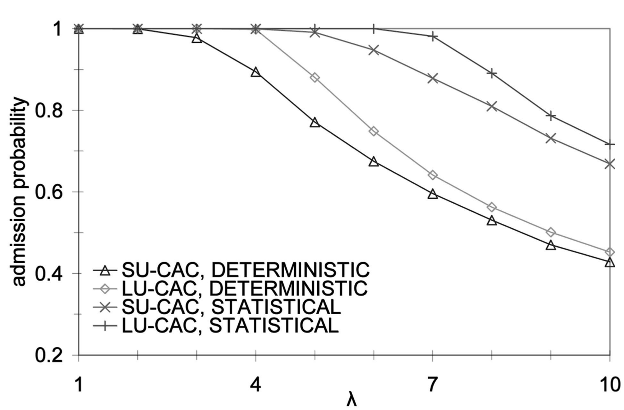

Fig. 12. The admission probability in the campus network.

assessing the quality of a call in terms of other relevant measures. This approach is a supplement to our QoS-provisioning system. A QoS adaptation module can be added to our system if desired. This module can use the measure approach in [26] to measure the resource availability in dynamic environments so as to adapt resource usage and quality requirements to achieve better and more flexible service to customers.

8 Final Remarks

In this work, we designed and implemented a QoS-provisioning system that can be seamlessly integrated into the current Cisco VoIP system. This QoS-provisioning system has been successfully realized in the *Internet2 Voice Over IP Testbed* in Texas A&M University. Generally speaking, a QoS-provisioning system has to have control or knowledge on the dynamics of the network, particularly on the traffic, in order to provide QoS guarantees. The practicability of a QoS-provisioning system relies on the degree of the dependency on such knowledge and control. Recall that our system adopts a static-priority scheduler. With such schedulers, low priority traffic has no impact on high priority traffic. In our system, voice traffic is assigned the highest priority so that knowledge of other highest priority traffic only is needed. Furthermore, our utilization delay analysis makes our system independent of the dynamic distribution of traffic which has the same priority as our voice traffic. To summarize, our QoS-provisioning system adopts several mechanisms which greatly reduce such dependencies and, hence, is practical.

The integration of our proposed QoS-provisioning system with the existing VoIP systems has practical applicability in different types of networks: 1) Closed networks where all traffic is under control. Our system can directly work in such networks. The examples of closed networks are enterprise networks and networks on ships, space shuttles, etc. 2) Semiopen networks, like Internet2, where all highest priority traffic can be known, although cannot be controlled. In our QoS provisioning system, voice traffic has the highest priority and low priority traffic has no impact on voice. Although VoIP is assigned the highest priority (the single real-time priority) in this paper, the theoretical results in our previous work [5], [6] support systems with some other applications which have a higher priority than VoIP applications. Our approach can still work by considering other higher priority traffic in the utilization computation. 3) Open networks, like the Internet, where traffic cannot be controlled and is difficult to be predicted. Our approach can provide statistical guarantees as long as the traffic in the Internet can be modeled. As we know, modeling Internet traffic is an open issue and is beyond the scope of this study. At the current stage, we can measure the Internet delay at the egress and ingress points of custom networks and dynamically change the bandwidth allocation at custom networks to compensate the fluctuation of the delay in the Internet. Certainly, such an approach cannot provide guaranteed services. However, if the Internet delay (backbone delay) is small (a general case), the end-to-end delay is predictable.

Appendix

This is the case, for example, in a network of identical link bandwidths and identical allocations of bandwidth to the real-time voice traffic on all links. In this case, we simplify the notation to let $C = C_k$, $u = u_k$, and $L = \max_k\{|L_k|\}$. We have an explicit delay formula as follows:

Corollary 1. *Let d be the maximum of worst-case delays suffered by all voice packets with highest priority across all buffers of output links in the network. If $u < \frac{1}{1+(h-2)(1-\frac{1}{L})}$, then*

$$d \leq \frac{1}{\frac{1}{r} - (h-1)} \frac{\sigma}{\rho}. \tag{7}$$

Therefore, the end-to-end delay d^{e2e} can be bounded by

$$d^{e2e} \leq \frac{h}{\frac{1}{r} - (h-1)} \frac{\sigma}{\rho}, \tag{8}$$

where $r = u\frac{L-1}{L-u}$ and h is the length of the longest flow route in the network. Note that this delay formula does not depend on the network topology except for the length h of the longest flow route.

Acknowledgments

This work was supported in part by the US National Science Foundation under Contracts 0081761, 0324988, and 0329181, by the Defense Advanced Research Projects Agency under Contract F30602-99-1-0531, and by Texas A&M University under its Telecommunication and Information Task Force Program. Dong Xuan's work was supported in part by the US National Sceince Foundation under contract ACI-0329155. The authors are grateful to the reviewers for their valuable comments that helped to improve the revised version. Part of this paper appeared in the *Proceedings of IEEE RTAS*, San Jose, CA, September 2002.

References

[1] "One-Way Transmission Time (Recommendation G. 114)," *Int'l Telecomm. Union (ITU)*, 1996.
[2] J. Davidson et al., *Deploying Cisco Voice over IP Solutions.* Cisco Press, 2002.
[3] "Registration, Admission and Status Signaling (Recommendation H. 225/RAS)," *Int'l Telecomm. Union (ITU)*, 1998.
[4] D. Collins, *Carrier Grade Voice over IP.* McGraw-Hill Professional Publishing, 2001.
[5] S. Wang, D. Xuan, R. Bettati, and W. Zhao, "Providing Absolute Differentiated Services for Real-Time Applications in Static-Priority Scheduling Networks," *IEEE/ACM Trans. Networking*, vol. 12, no. 2, pp. 326-339, 2004.
[6] S. Wang, D. Xuan, R. Bettati, and W. Zhao, "Differentiated Services with Statistical Real-Time Guarantees in Static-Priority Scheduling Networks," *Proc. IEEE Real-Time Systems Symp.*, Dec. 2001.
[7] V. Paxson, "End-to-End Routing Behavior in the Internet," *Proc. ACM SIGCOMM*, Aug. 1996.
[8] J. Alexander et al., *Cisco CallManager Fundamentals.* Cisco Press, 2002.
[9] "Gatekeeper External Interface Reference, Version 3.1," Cisco Documentation, 2001.
[10] "Mesquite csim 19," http://www.mesquite.com, 2004.
[11] "Abilene Network Backbone," http://abilene.internet2.edu, 2003.
[12] "Recommendation H.323," *Int'l Telecomm. Union (ITU)*, 1998.
[13] M. Arango, A. Dugan, I. Elliott, C. Huitema, and S. Pickett, "Media Gateway Control Protocol (MGCP) Version 1.0," IETF RFC 2705, 1999.

[14] F. Cuervo, N. Greene, A. Rayhan, C. Huitema, B. Rosen, and J. Segers, "Megaco Protocol Version 1.0," Megaco Protocol Version 1.0, 2000.
[15] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol v.2.0," IETF RFC 3261, 2002.
[16] "Nortel Networks: Voice over IP Solutions—Succession Enterprise," http://www.nortelnetworks.com, 2004.
[17] "Avaya IP Telephony Solutions," http://www.avaya.com, 2004.
[18] "Alcatel IP Telephony," http://www.alcatel.com, 2004.
[19] "3Com IP Telephony Solution," http://www.3com.com, 2004.
[20] "Delivering World-Class Telephony on a Distributed, IP-Based Architecture," Nortel Networks, Feb. 2001.
[21] S. Jamin, S. Shenker, and P. Danzig, "Comparison of Measurement-Based Admission Control Algorithms for Controller-Load Service," *Proc. IEEE Infocom,* Apr. 1997.
[22] D. Tse and M. Grossglauser, "Measurement-Based Call Admission control: Analysis and Simulation," *Proc. IEEE Infocom,* Apr. 1997.
[23] G. Agrawal, B. Chen, W. Zhao, and S. Davari, "Guaranteeing Synchronous Message Deadlines with the Timed Token Protocol," *Proc. IEEE Int'l Conf. Distributed Computing Systems,* June 1992.
[24] S. Chong and S. Li, "(σ, ρ)-Characterization Based Connection Control for Guaranteed Services in High Speed Networks," *Proc. IEEE Infocom,* Apr. 1995.
[25] S. Love, J. Nowalk, N. Weiss, and K. Then, "A Case Study in Campus VoIP Deployment: Hype vs. Reality," 2000.
[26] A.P. Markopoulou, F.A. Tobagi, and M.J. Karam, "Assessing the Quality of Voice Communications over Internet Backbones," *IEEE/ACM Trans. Networking,* vol. 11, no. 5, pp. 747-760, 2003.

Shengquan Wang received the BS degree in mathematics from Anhui Normal University (AHNU), China, in 1995, and the MS degree in applied mathematics from the Shanghai Jiao Tong University (SJTU), China. In 1998, he received the MS degree in mathematics from Texas A&M University. Currently, he is a graduate student in computer science at Texas A&M University. He was a graduate assistant in the Department of Applied Mathematics at SJTU. From 1998 to 2005, he was a graduate assistant and a research assistant in the Department of Mathematics and the Deparment of Computer Science at TAMU, respectively. His research interests include real-time computing and communication, and security in networks and distributed systems.

Zhibin Mai received the BS and MS degrees in electronic engineering from the Shanghai Jiao Tong University (SJTU), China, in 1992 and 1995. Currently, he is a research associate at the Institute for Scientific Computation, Texas A&M University (TAMU). He is also a PhD student in the Department of Computer Science at TAMU. He was on the faculty of electronic engineering at SJTU from 1995 to 1996. He worked as a system engineer and network specialist from 1996 to 2000 in Singapore. His research interests include real-time computing and communication, distributed computing systems, and high performance computing. He is a student member of the IEEE.

Dong Xuan received the BS and MS degrees in electronic engineering from Shanghai Jiao Tong University (SJTU), China, in 1990 and 1993, and the PhD degree in computer engineering from Texas A&M University in 2001. Currently, he is an assistant professor in the Department of Computer and Information Science, The Ohio State University. He was on the faculty of electronic engineering at SJTU from 1993 to 1997. In 1997, he worked as a visiting research scholar in the Department of Computer Science, City University of Hong Kong. From 1998 to 2001, he was a research assistant/associate in the Real-Time Systems Group of the Department of Computer Science, Texas A&M University. His research interests include real-time computing and communications, network security, and distributed systems. He is a member of the IEEE.

Wei Zhao completed his undergraduate program in physics at Shaanxi Normal University, Xian, China, in 1977. He received the MSc degree and PhD degree in computer and information science from the University of Massachusetts, Amherst, Massachusetts, in 1983 and 1986, respectively. He is currently the director of the computer and network systems division at the US National Science Foundation and an associate vice president for research at Texas A&M University. In 1990, he joined Texas A&M University where he has been a full professor in the Department of Computer Science since 1996. Between 1997 and 2001, he served as a department head. His current research interests includes secured real-time computing and communication, distributed operating systems, databases, and fault tolerant systems. His research group has been recognized by various awards and prizes and he is an inventor for two US patents and has published more than 200 papers in journals, conferences, and book chapters. Dr. Zhao was an editor of the *IEEE Transactions on Computers* between 1992 and 1996. He is currently on the editorial board of the *IEEE Transactions on Parallel and Distributed Systems*. He has been a program and general chair of many IEEE international conferences. He is a fellow of the IEEE.