

DOSA: An Architecture for Providing a Robust IP Telephony Service

C.R. Kalmanek, W.T. Marshall, P. P. Mishra, D.M. Nortz, K.K. Ramakrishnan

AT&T Labs., New Jersey

Abstract -- An increasing number of communication services are moving to an IP-based infrastructure. Packet telephony is probably the first important real-time service that must be supported well over an IP network. The use of IP presents a tremendous opportunity for service providers to exploit endpoint intelligence to offer creative new services going far beyond the current telephony service model. However, to support telephony, signaling protocols are needed that allow the service provider to offer network-layer service differentiation and, at the same time, to control access to both the enhanced network-layer quality of service as well as other services. This paper describes the Distributed Open Signaling Architecture (DOSA), which incorporates protocols that meet these needs. A key contribution of our work is a recognition of the need for coordination between call signaling, which controls access to telephony specific services, and resource management, which controls access to network-layer resources.

Index terms-- packet telephony, quality of service, signaling

I. INTRODUCTION

The world is evolving to use packet networks as the underlying framework for communications. These networks will allow multiple services to be provided to the user, including traditional best-effort data service as well as enhanced services like telephony. At the same time, improvements in silicon are reinforcing the trend towards increased functionality in endpoints. This combination of packet networking and intelligent endpoints has the potential to enable a rich set of new services by integrating multiple applications, media streams, and network services together, in addition to supporting traditional telephony.

To support real-time applications like telephony well, IP networks must provide different levels of service to different applications. To be comparable with today's telephone service, a packet telephony service must meet tight bounds on end-to-end packet delay and loss. In the absence of differentiated service, all applications using the network receive the "lowest common denominator" in terms of performance and quality. Over the past several years, experience with IP telephony has shown that this service quality does not meet user's needs for a robust telephony service. One alternative to differentiated service is to over-engineer end-to-end capacity throughout the network. However it is unlikely that this will be economical for use on a large scale. As a result, IP has been evolving to incorporate some form of quality of service [1][2][3] at the network layer.

To provide network-layer service differentiation, resource management protocols must allow the provider to control how and when resources are used, and to monitor usage. In addition, it is now accepted that support for differentiated services will require charging mechanisms that allow the provider to derive revenue from these

services. An IP telephony service also provides the user with additional capabilities, such as translation, privacy, conferencing, and messaging. These capabilities provide added value to the user, allowing the provider to derive revenue from the telephony service, above and beyond providing enhanced network-layer service.

We believe that *coordination* between call signaling and resource management protocols is key to achieving the balance between managing the enhanced service, giving users an incentive to use the enhanced service and generating revenue for the service provider. Call signaling protocols for the IP telephony environment [4][5][6] have been developed independent of resource management issues. Similarly, IP resource management protocols have not traditionally provided any hooks for service-specific authorization or control. This paper describes the Distributed Open Signaling Architecture (DOSA) for packet telephony, which addresses these issues in order to meet the needs of a telephony service. DOSA exploits endpoint intelligence (in keeping with the IP philosophy) while allowing the service provider to derive revenue from the service. While some of the details relate specifically to packet telephony, we postulate that the DOSA framework, in which application layer protocols coordinate with resource management to authorize use of enhanced QoS, is broadly applicable to providing enhanced services over IP.

Whenever a provider charges for services, user expectations need to be kept in mind. Telephony users expect that negotiation for network layer resources has completed before the phone rings. It would be unacceptable for a user to answer the phone only to find that resources are not available. Also, there are commonly understood semantics for charging in our society. For instance, residential users are not charged while the phone is ringing.

In the circuit switched telephone network, the intelligence for feature functionality is provided by network switches and other servers. Evolution of functionality has been slow. On the other hand, IP is based on intelligent endpoints that can participate in application and network layer protocols. Intelligent endpoints have the potential to enable tremendous innovation in the types of features and functionality available to the user. We believe this potential is especially compelling when the endpoint integrates many different services. The classic integration of voice, video, data, and other media is naturally supported by the endpoint. DOSA enables applications on intelligent endpoints to participate in managing calls and supports negotiation of new functionality among endpoints.

That said, there are important functions that rightfully belong in the network. The network service provider has the unique ability to manage and provide enhanced network layer quality of service. In addition, the network

service provider can offer services to the user based on its role as a trusted intermediary. For example, the provider plays a role in ensuring the integrity of routing and naming information, and the privacy of signaling and addressing information. The network service provider can manage customer profiles in the network to ensure that users receive consistent service independent of how and where they access the network. This ensures that users receive consistent service even when an endpoint is unavailable. For example, while endpoints may have the ability to forward calls, the network can forward calls to a voice mailbox if an endpoint is unavailable. Finally, there are many services that can be implemented more efficiently in the network than at the endpoint. For example, it may be more cost effective to implement conferencing bridging in a multi-point bridge than in every endpoint.

There has been considerable debate in the IP networking community about maintaining network layer state. In the telephone network, motivated by the limited functionality of the end terminal, state information associated with the connection and the “application” is kept in a switch. In comparison with network layer state, the state information associated with an application such as telephony is much more complex. Maintaining that state information increases the requirements on the reliability and availability of the call control servers, impacting cost and scalability. In contrast, we recognize that the endpoint has the greatest need and incentive to be concerned with the state of the calls in which it is participating. Thus, we believe that application-level state information should be kept at the endpoint, rather than in the network, whenever possible.

A key contribution of DOSA is a recognition of the need for coordination between call signaling, which controls access to telephony-specific services, and resource management, which controls access to network-layer resources. This coordination provides several critical functions: it ensures that users are authenticated and authorized before receiving access to enhanced QoS for the telephony service; it ensures that network resources are available end-to-end before ringing the destination phone; and it ensures that the use of resources is properly accounted for, consistent with the semantics of telephony where charging occurs only after the called party answers.

This paper presents the DOSA principles and architecture. This architecture is being further developed to support telephony over cable access networks, where DOSA is being realized as profiles of current Internet protocols. DOSA call signaling is implemented using a profile and extensions of the Session Initiation Protocol (SIP) [4], while the resource management and authorization functions are supported by profiles of RSVP [7] and the Common Open Policy Service (COPS) [8] respectively. The architecture supports integration with an application layer anonymizer that performs address translation, allowing IP telephony to support user’s

expectations of privacy. We also support network-based conference bridging for multi-party calls.

The rest of the paper is organized as follows. In Section II, we present the principles and application requirements that drive the signaling architecture design. In the next section, we describe the components of the Distributed Open Signaling Architecture (DOSA). Section IV presents a simple call flow. Section V details DOSA’s resource management mechanisms for meeting telephony’s QoS requirements. The next section examines issues related to scaling the service by exploiting endpoints to maintain state. In Section VII we show how privacy and anonymity may be provided. Finally, we conclude.

II. REQUIREMENTS AND DESIGN PRINCIPLES

Since a signaling architecture is developed for a specific family of applications, it is necessary to briefly describe the telephony service that DOSA was designed to support. We designed DOSA to support a first-line telephone service that is at least comparable to the circuit-switched telephone service offered today. There are important requirements in the areas of reliability, performance, scalability, and feature support. Performance requirements for the enhanced bearer channel and signaling include: 1) Low delay: end-to-end packet delay must be small enough that it does not interfere with normal voice conversations, typically on the order of 300 milliseconds round trip; 2) Low packet loss: packet loss must be small enough to not perceptibly impact voice quality or voice-band modem performance; 3) Short post-dial delay: the delay between the user dialing the last digit, and receiving positive confirmation of the call being setup, must be short enough that users do not perceive a difference from the post-dial delay they now experience in the circuit-switched network; 4) Short post-pickup delay: the delay between a user picking up a ringing phone and the voice path being cut through must be short enough that the “hello” is not clipped.

We believe that the service needs to support popular features such as Call Forwarding, Caller ID (and Caller ID Block), Call Waiting, and Three-Way Calling. The architecture must support usage recording, to enable billing for, and maintenance of, the service. The architecture must also allow the provider to meet regulatory requirements, such as law enforcement assistance, without the knowledge or cooperation of endpoints.

The design of DOSA was influenced by several key design principles that arise from the requirements and philosophy outlined above. The architecture needs to:

1. Support differentiated quality of service, while allowing the provider to derive revenues from its use.
2. Ensure network resources are available before inviting the end-users to participate in the call (e.g., by ringing the phone).

3. Allow, and even encourage, implementation of specific services and features in intelligent endpoints.
4. Enable the service provider to give a consistent view of basic services and features even when customer premise equipment is unavailable.
5. Allow the service provider to add value by acting as a trusted intermediary and offering services that truly belong in the network.
6. Ensure that the network is protected from fraud.
7. Enable maintenance of caller and callee privacy, including IP address privacy.
8. Enable a large-scale, cost-effective implementation.

III. DISTRIBUTED OPEN SIGNALING ARCHITECTURE

The Distributed Open Signaling Architecture (DOSA) follows the principles outlined in Sections I and II to support a robust telephony service. Figure 1 illustrates the key components in the architecture. We assume that there will be a broad range of *IP endpoints* (called CPE below) that support telephony, including personal computers, handheld devices, and adapters for conventional telephones. Endpoints may be connected to a local area network, which connects to the service provider via an access network. Typical access networks include point-to-point links and shared media networks, such as cable or wireless access networks. For instance, the customer's LAN may be connected to the access network via a cable modem. Access to the backbone network is controlled by trusted *edge routers*. Edge routers participate in resource reservation protocols, support policy-based admission control, and provide QoS to the voice and signaling flows. *DOSA proxies* provide subscriber authentication, perform call routing, implement service-specific admission control policies, and provide other service-specific functions such as telephony feature support. A *gate controller* is associated with each DOSA proxy. Gate controllers support a policy function to authorize access to enhanced quality of service. *Media servers* process media streams, e.g., to perform audio bridging, play terminating announcements, provide interactive voice response services, etc. Finally, *PSTN gateways* interface to the Public Switched Telephone Network.

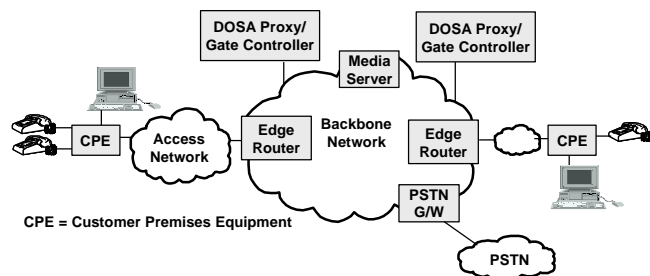


Figure 1: Architecture Components

Telephony endpoints are considered to be "clients" of the telephony service, and can implement a range of telephony features. They collect dialed digits, participate in signaling

and contain the service logic required for call setup and feature support. Endpoints also participate in end-to-end capability negotiation. However, endpoints are not trusted to provide accurate information to the network or to keep information that is received private.

The edge router receives resource management requests from endpoints, and is responsible for ensuring that packets are provided the QoS they are authorized to receive (e.g., through packet marking, or through queueing and scheduling the packets as a specific QoS assured flow). We introduce the concept of a "gate" in an edge router, which manages access to enhanced quality of service. The gate is a packet classifier and policer that ensures that only those IP flows that have been authorized by the gate controller are granted access to enhanced QoS in the access and backbone networks. Edge routers require authorization from a gate controller on a call-by-call basis (for the telephony service) before "opening" the gate and providing access to enhanced QoS for an end-to-end IP flow. Opening a gate involves an admission control check that is performed when a resource management request is received from the endpoint for an individual call, and it may involve resource reservation in the backbone for the call if necessary. The packet filter in the gate allows a flow of packets to receive enhanced QoS for a call from a specific IP source address and port number to a specific IP destination address and port number. Since edge routers know about the resource usage associated with individual IP flows, they generate the usage events that enable providers to charge users for service. In DOSA we set up a gate in advance of a resource management message. This allows the DOSA proxy and associated gate controller to be "stateless" and not maintain the state of calls that are in progress.

DOSA proxies implement a set of service-specific control functions to support the telephony service:

1. Authentication and authorization: Since services are only provided to authorized subscribers, DOSA proxies authenticate signaling messages and authorize requests for service on a call-by-call basis.
2. Name/number translation and call routing: DOSA proxies translate dialed E.164 numbers, or names, to a terminating IP address based on call routing logic.
3. Service-specific admission control: Admission control may be used to implement overload control mechanisms, e.g., to restrict the number of calls to a particular location. DOSA proxies may also provide precedence for particular calls (e.g., 911 calls).
4. Signaling and service feature support: While many service features are implemented by endpoints, a DOSA proxy also plays a role in feature support, e.g., to provide privacy or to ensure that calling features behave consistently even when an endpoint is down.

DOSA proxies are typically organized in domains: a proxy is responsible for a set of endpoints and the

associated edge routers. While endpoints are not trusted, there is a trust relationship between the edge router and its associated DOSA proxy/gate controller, since the gate controller plays a role as a policy server controlling when the edge router can provide enhanced QoS service. There is also a trust relationship among proxies.

The DOSA proxy is designed as a simple transaction server, so that failure of a DOSA proxy does not affect calls in progress. A domain will likely have both a primary and a secondary DOSA proxy. If the primary DOSA proxy fails, only calls in a transient state are affected. The endpoints involved in those calls will time out and retry. All active calls are unaffected, made possible because signaling proxies retain no call state for active calls. This design makes the proxy efficient and highly scalable, and reduces their reliability requirements.

DOSA supports inter-working with the circuit switched telephone network through PSTN gateways. A signaling gateway provides signaling inter-working between DOSA call signaling and conventional telephony signaling protocols such as ISUP/SS7, although the details are beyond the scope of this paper.

There are additional system elements that may be involved in providing the telephony service. For example, the DOSA proxy may interface with other servers that implement the authorization or translation functions. Similarly, three way calling may be supported using a conference bridge in the network.

IV. A SIMPLE CALL FLOW

In this section, we illustrate the functional interaction between the various elements in the DOSA architecture to set up a call from one packet telephony client to another.

An endpoint initiates a call by sending a request to a DOSA proxy. Each request is authenticated to ensure it originated from a valid subscriber. The DOSA proxy authorizes the request and performs service-specific admission control. Request authorization takes into account the customer's profile, including the services to which the customer has subscribed, the status of the subscriber's account, etc. For example, an endpoint may be constrained to make only local telephone calls, or service may be denied based on an unusual pattern of usage that suggests fraud.

The first steps of call setup are shown in Figure 2. When a user goes off-hook and dials a telephone number, the originating endpoint (CPE_O) collects the dialed digits and sends a SETUP message to the "originating" DOSA proxy (DP_O). DP_O verifies that CPE_O is a valid subscriber of the telephony service and determines whether the subscriber is authorized to place this call. DP_O then translates the dialed number into the address of a "terminating" DOSA proxy (DP_T) and forwards the SETUP message to it.

We assume that the originating and terminating DOSA proxies trust each other. DP_O may augment the SETUP message that it forwards with additional data, such as

charging information containing the account number of the caller. DP_T then translates the dialed number into the address of the terminating endpoint (CPE_T) and forwards the SETUP message to CPE_T to notify it about the incoming call.

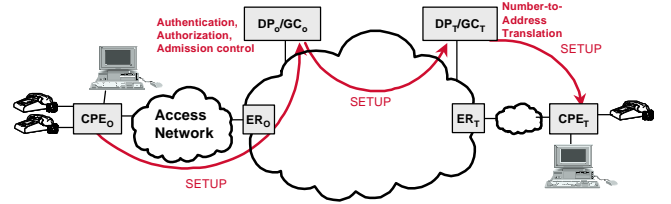


Figure 2: Call Setup - Part 1

In Figure 3, CPE_T sends a SETUP_ACK to DP_T. This SETUP_ACK contains a subset of the capabilities in the SETUP message that are acceptable to the terminating endpoint. DP_T forwards the SETUP_ACK to DP_O. DP_T in its role as a gate controller sends a GATE_SETUP message to the terminating edge router (ER_T), conveying policy information that allows ER_T to open a gate for the IP flow associated with this phone call. The GATE_SETUP message may also contain charging information such as the account number of the subscriber that will pay for the call. From the perspective of the edge router, this charging information is opaque data that is output in usage events for later interpretation by a rating and billing server. DP_O also in turn sends a GATE_SETUP message to the originating edge router (ER_O) to indicate that it can open a gate for the IP flow associated with the phone call. Finally, DP_O forwards SETUP_ACK to CPE_O. This SETUP_ACK contains the IP address and port number to use when communicating with CPE_T.

We note that the translation function illustrated in this portion of the call flow is one of the value-added services offered by the service provider. While it may be possible for the originating endpoint to obtain the IP address of the destination through some other means, the service provider has a role in maintaining the database associated with its subscribers and ensuring that calls are correctly routed to the desired destination.

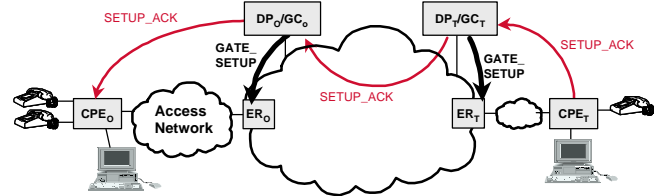


Figure 3: Call Setup - Part 2

Support for telephony requires that the service provider manage access to resources and that adequate capacity is available on an end-to-end basis. In many cases, such as cable or wireless networks, the access network is capacity constrained (at least in the upstream direction) and it is necessary to do resource management on a per-flow basis. However, there may be several alternatives in the

backbone, ranging from per-flow per-hop admission control to coarse-grained resource provisioning.

As a result, DOSA proposes a resource management architecture that partitions resource management into access and backbone “segments.” This allows distinct mechanisms to be used in each segment based on the resources that are available and the aggregation level that is appropriate to that segment. After the first phase of call signaling described above, both clients have completed capability negotiation and know what resources are needed end-to-end. Clients send resource management messages to the gate at the network edge, where the network makes an admission control decision based both on resource availability as well as on policy information associated with the gate. These messages may be interpreted hop-by-hop over the customer’s LAN and the access network. The edge router containing the gate then maps the resource management message to the resource management protocol that is used on the backbone (e.g., IETF’s Diffserv).

Segmented resource assignment is beneficial for several reasons. First, it allows for different bandwidth provisioning and signaling mechanisms for different network segments. Resource-poor segments of the network can maintain per-flow reservations and carefully manage resource usage, while backbone segments, which may have sufficient resources to manage resources more coarsely, do not need to keep per-flow state. Segmented resource management does not require a single end-to-end resource management protocol, which allows the network to cope with heterogeneity by performing mappings at segment boundaries. In addition, when the backbone does not use per-flow signaling (as with a Diffserv backbone), segmented resource management reduces the call setup time (minimizing post-dial delay) by avoiding the overhead of an end-to-end resource management exchange in addition to the end-to-end call signaling exchange.

The resource management exchange between the CPE and edge router reserves resources in *both* directions over the customer’s LAN and access networks. This has the additional benefit of minimizing the number of messages required for resource management in bandwidth-constrained access networks.

An alternative to segmented resource management would be to do resource management on an end-to-end basis, perhaps using RSVP, with the messages tunneled across the diffserv backbone. This approach requires RSVP to be supported end-to-end, potentially across multiple provider networks. Moreover, it potentially introduces an additional round-trip delay, adversely affecting both post-dial delay and voice cut-through time (post-pickup delay).

DOSA’s resource management protocols distinguish between two phases: Reserve and Commit. During the first phase, resources are reserved but are not yet made available to the endpoint. This ensures that resources are available before ringing the far-end telephone. The second phase, which commits resources, is initiated after ringing the far end telephone and after the called party picks up.

At this point, resources are made available to the endpoint, and recording is started so that the user can be billed for usage. The use of a two-phase resource management protocol is essential because of the unique requirements associated with human communication, such as telephony. In addition to ensuring that resources are available before ringing the phone, it also preserves the semantics of billing that users are accustomed to, whereby usage recording is not started until the called party picks up the phone. Backbone resources are reserved and, if necessary, allocated, in the first phase of the two-phase resource reservation protocol. If backbone resources have to be allocated, allocation in the first phase is important as it reduces post-pickup delay (this minimizes the likelihood of clipping the first few syllables of the conversation).

Figure 4 shows the first phase of the resource management protocol. CPE_O and CPE_T issue RESERVE messages to ER_O and ER_T respectively. ER_O and ER_T perform an admission control check for resource availability (initiating signaling for resource reservation in the backbone if necessary) and if successful send a RESERVE_ACK to the respective CPEs. The edge routers use the information from the Gate Setup to match the resource reservation request with the policy-based authorization provided by the gate controller.

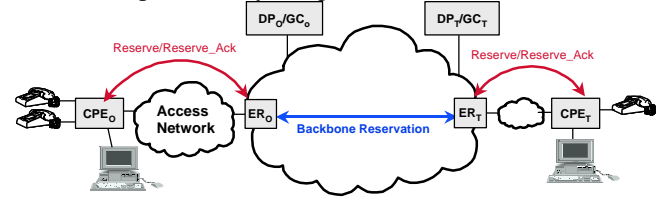


Figure 4: Resource Management - Part 1

Figure 5 shows the second phase of resource management and call setup. After determining that resources are available, CPE_O sends a RING message to CPE_T instructing it to start ringing the phone. CPE_T sends a RINGBACK message to CPE_O indicating both that resources are available and that the RING message was received. When the called party picks up the phone, CPE_T sends a CONNECT message to CPE_O and a COMMIT message to ER_T. CPE_O also sends a COMMIT message to ER_O when it receives the CONNECT message. The COMMIT messages cause resources to be allocated for the call in the access network. The arrival of the COMMIT messages at ER_T and ER_O causes them to *open* their respective gates and also start accounting for resource usage. To prevent some theft of service scenarios, the edge routers coordinate gate opening by exchanging messages. Gate coordination is further described in Section V.

The DOSA flows are designed to meet stringent call signaling performance requirements. We assume that signaling messages are given priority on the access and backbone networks, for example using the Diffserv Assured Forwarding service on the backbone. Post-dial delay until the caller hears remote ringback includes one round trip through the proxies for the initial SETUP

exchange, the time needed for resource reservation, and another round trip between the endpoints for the RING/RINGBACK exchange. Segmented resource reservation may require only a local round trip. Post-pickup delay to open the originating gate is only 1/2 round trip for the CONNECT message from CPE_T to CPE_O , plus the time needed for the (local) COMMIT exchange at CPE_O . This design can comfortably meet our signaling performance requirements.

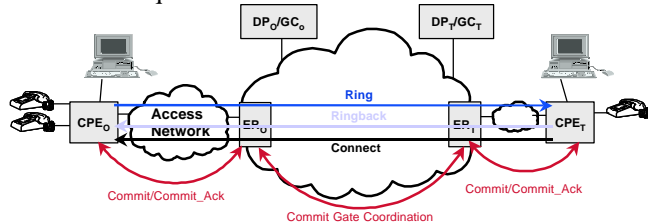


Figure 5: Resource Management - Part 2

DOSA's proxy function can be realized as an enhancement of a SIP proxy server [4]. A SIP proxy performs call routing and service feature support, and can operate as a stateless server. We hope to enhance the SIP protocol to provide coordination with resource management, support for privacy and charging information, thus meeting the requirements outlined in this paper. IPSEC [10] is used for endpoint authentication.

The SETUP message can be realized in SIP using a SIP INVITE message, but the semantics of INVITE need to be enhanced so that CPE_T does not ring the destination phone at this point, since resources are not yet available. One approach to achieving this is to add a "Stage1" header to the INVITE message instructing CPE_T not to ring the phone. The RING message of the DOSA architecture is a second SIP INVITE message without the "Stage1" header.

The DOSA resource management mechanisms are being implemented as a profile of the IETF RSVP protocol. RESERVE is implemented as a PATH message sent by the client towards the remote end-point. The edge router intercepts the PATH message. The edge router performs admission control and, in a capacity constrained access network, uses layer 2 mechanisms to reserve access resources. The edge router then responds with a RESV message to the client.

We have developed extensions to RSVP allowing bidirectional bandwidth reservations using a single PATH/RESV exchange. This is achieved by adding opaque objects to RSVP to convey the bi-directional bandwidth requirements. The RSVP extensions also support inter-operation with traditional RSVP resource management, thus accommodating arbitrary LAN topologies between the endpoint and the access network. Specifically, the edge router uses the Previous Hop (PHOP) field of the RSVP PATH message to determine if the endpoint is the previous hop of a bi-directional resource reservation request. If not, the edge router generates both a RESV response and a new PATH message, addressed to the endpoint. This allows us to cope with asymmetric

routing in the LAN. The COMMIT message exchange uses RSVP syntax, but is unicast by an endpoint to the edge router.

V. DETAILS OF DOSA'S RESOURCE MANAGEMENT MECHANISMS

In this section, we describe the details of how DOSA signaling is used to create a gate and control its operation in a manner that ensures only authorized calls have access to voice-grade QoS and allows the parameters of a gate to be modified over the duration of a call without involving the gate controller. DOSA provides detailed usage recording information and prevents theft of service.

In the DOSA model, a gate controller sets up a gate in the edge router after initial call signaling and authorization. The key parameters of a gate include:

- A packet classifier of the form $\langle source\ address, destination\ address, source\ port, destination\ port, protocol\ ID \rangle$ that identifies the IP flow associated with a gate;
- A set of *resource envelopes*, described below, that define the authorized, reserved and committed resources associated with this IP flow;
- An (optional) bound on the maximum duration of time for which a gate may be open;
- Service-specific opaque data used by the DOSA proxy or other servers, e.g., for billing.

Once a gate has been set up, an endpoint directly communicates with the edge router to request QoS for the flow associated with the gate. An edge router times-out state associated with a gate if it does not receive the initial resource management message from the endpoint within a pre-specified time interval. Denial-of-service attacks in which a client reserves an excessive amount of resources are prevented by enforcing an upper bound on the amount of resources that can be reserved by any endpoint.

In the context of policy-enabled networking [9], the gate controller may be viewed as a policy decision point (PDP), and the edge router as a policy enforcement point (PEP). The gate setup mechanism uses the COPS protocol [8], with the gate controller sending a "response" to the edge router based on an a priori, outstanding request from the edge router for policy information.

The relationship between different categories of resource – authorized, reserved, and committed – is shown in Figure 6. A set of resources is represented by an n -dimensional space (shown here only in 2-dimensions) where n is the number of parameters (e.g., bandwidth, burst size, delay, jitter, classifiers) needed to describe the required resources.

When a call is first established, the resource management protocols authorize the use of some maximum amount of resources, indicated by the outer oval, specifying the authorized resources. This is performed at Gate Setup time. When a client makes a reservation for a call, it reserves a certain amount of resources, which must not be greater than those for which it has been authorized. When the call is ready to proceed, the client commits to some

amount of resources, which must not be more than the reserved resources. In many cases, the committed and reserved resources will be equal. The committed resources represent resources that are currently in use by the active call, whereas reserved resources represent those that are tied up by the client and have been removed from the pool for admission control purposes, but which are not necessarily being used by the client.

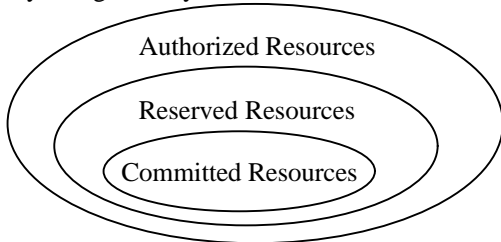


Figure 6: Authorized, Reserved and Committed Resources

The use of separate authorized, reserved and committed envelopes give endpoints the flexibility to renegotiate QoS with the network without requiring gate controller involvement. For example, endpoints might start communicating using a low-bit-rate audio coder. They can subsequently switch to a higher bit-rate coder or add a video stream, as long as the requested QoS is within the authorized envelope, and passes admission control.

Since the gate is in the bearer channel path and since it is directly involved in managing resources provided to an endpoint, resource usage accounting is done by the gate, as we describe in Section VI. It is also important to do usage accounting in the edge router to cope with endpoint failures. If an endpoint that is involved in an active call crashes, the edge router needs to be able to detect this and stop usage accounting for the call. This can be accomplished using soft state through a resource management refresh message, or by monitoring the flow of packets along the bearer channel path for continuous-media applications. In addition, since the gate retains state for flows that have been authorized by a service-specific gate controller, it is used to store service-specific information related to charging, such as the account number of the subscriber that will pay for the call.

Gate coordination ensures that the gates at the two edge routers open and close almost simultaneously (e.g., within a few hundred milliseconds of each other.) Since usage accounting at the edge routers is tied to the opening and closing of the gates, gate coordination protects against various theft of service scenarios.

We describe the gate coordination protocol for a simple two-party call, shown in Figure 7. Logically, for a gate to be opened after resources have been reserved, the edge router must receive *both* a COMMIT message from the client and a message from the remote edge router. The details are as follows. On receipt of a COMMIT message, the edge router opens the gate, sends a GATE_OPEN message to the remote edge router, sends a gate open event to a usage recording server, and starts a timer waiting to

receive a GATE_OPEN message from the remote edge router. If the edge router receives the GATE_OPEN message before the timeout, it sends a COMMIT_ACK to the client; otherwise, it closes the gate and sends a gate close event to the usage recording server. The GATE_OPEN message originated by an edge router also carries information about the resources that were allocated at that edge router as a result of the COMMIT. This allows both edge routers to ensure that the gate parameters are consistent; otherwise, the gates are closed.

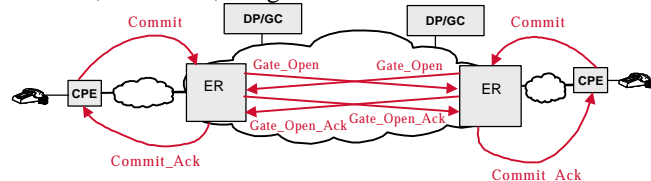


Figure 7: Gate Coordination

The edge router opens the gate as soon as it receives the COMMIT, in order to avoid the possibility of "clipping" the "Hello" when the far end user picks up the phone and starts talking. Since the terminating client sends the COMMIT message as soon as the user picks up, the gate open event starts accounting for resource usage at precisely the moment that resources are needed and used by the end user.

Gate coordination is also done at the time a gate is closed. Each edge router sends a GATE_CLOSE message to its peer edge router when it receives an explicit RELEASE (RSVP PATH_TEAR) message from the client (or when it detects that the client is no longer actively generating packets for the flow associated with a gate, e.g., using RSVP's soft-state mechanism). An edge router closes a gate and sends a gate close event when it receives either a RELEASE or a GATE_CLOSE message. This ensures that the gates associated with a call are closed almost simultaneously.

Gate coordination is needed to prevent fraud and theft of service in situations where a malfunctioning or malicious endpoint does not issue the expected signaling messages. Given the user's incentive not to be charged for service, it is essential that protocol mechanisms are robust against abuse. The mechanisms are designed deal with subtle cases, such as two "one-way" flows being set up to support a two-way call or the possibility that one endpoint commits more resources than expected if it is not the party that is paying for usage. Fraud of this type can only be prevented by synchronizing the operation of the gates.

Once a gate is set up, endpoints can communicate over the network with enhanced QoS. Several telephony features involve changing the endpoints of a call, for example when a call is transferred or forwarded, or with three-way calling. This requires the packet classifiers associated with a gate to be modified to reflect the address of the new endpoint. In addition, changing the end-points involved in a call may affect how the call is billed. As a

result, the packet classifiers must be changed by the gate controller.

Figure 8 illustrates a simplified view of how DOSA signaling achieves this for a call transfer. We refer to a service called “Transfer without Consultation” where the transferring party puts the originating party on hold, and signals a new destination for that call. The call originator hears another ringback tone before the new destination answers. Call transfer is one of several services that make use of a call signaling primitive which redirects a call. Initially, two gates were set up at the two edge routers adjacent to the called and calling parties. To initiate a transfer, one party (say, the called party) sends a REDIRECT message to its associated DOSA proxy (DP_T). This message contains state information [12], described in Section VI, that is used by DP_T to determine the identities of the originating DOSA proxy (DP_O) and the originating endpoint. This information needs to be passed from the endpoint to the DOSA proxy, because the DOSA proxies are stateless and retain no information about calls in progress. DP_T forwards the REDIRECT message to DP_O after validating that the endpoint issuing the call transfer is authorized to initiate a call transfer. DP_O sends the REDIRECT message to CPE_O, which sets up a call leg to the new destination. In parallel, DP_T releases the gate at the terminating ER.

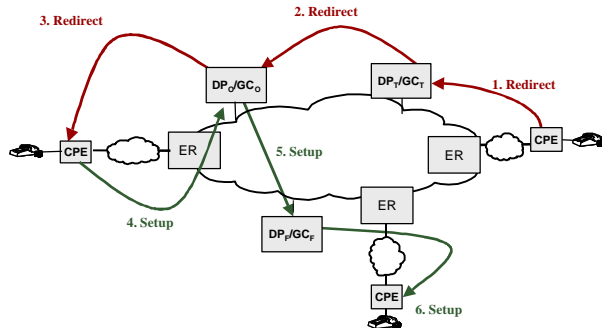


Figure 8: Call Transfer: Changing Packet Classifiers

Other examples of gate modification include cases such as three-way calling involving a network-based conference bridge. In three-way calling, the controlling party redirects the other parties to a conference bridge.

The DOSA proxy/gate controller is involved in modifying gate parameters for three reasons. One is that it needs to both authorize the use of the primitives and the use of network resources such as a bridge. It also needs to be an intermediary in handling REDIRECT messages in order to ensure that billing information is handled correctly and securely. This is described in more detail in Section VI. Finally, gate controllers must modify the gates to change the packet classifiers. The REDIRECT function is realized in SIP using the “Also” and “Replaces” headers that are part of the proposed SIP call control services [11].

DOSA recognizes that there may be a need to share resources across multiple calls, especially when resources are in short supply. In particular, when using the call-

waiting feature in telephony, the endpoint may be involved in only one conversation at a time. It is feasible in this case to share the network-layer resources (e.g., on the access link) between the two conversations. Therefore, DOSA allows a set of network layer resources (e.g., a bandwidth reservation) to be identified using an explicit *Resource_id*, and allows a gate to be associated with those resources. Resource management primitives allow the resources associated with a gate to be *shared* with another gate at the same edge router. This improves the efficiency with which resources in the access network are utilized.

Consider the call waiting service, where a user can switch back and forth between two calls. Clearly, only one of the calls needs to actively generate voice packets at any given time. Therefore, the reserved bandwidth can be shared on the network segments that are common to the end-to-end path of the two calls. DOSA facilitates this sharing by allowing an endpoint to reissue the RESERVE and COMMIT primitives to bind the resources to a new gate. This is shown in Figure 9.

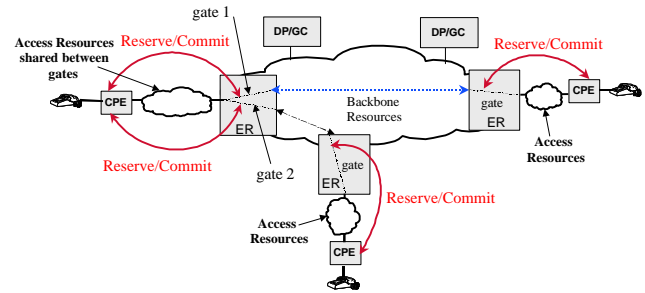


Figure 9: Sharing Resources Across Gates

When an endpoint reissues a RESERVE message with the same *Resource_id*, but with a new *gate_id*, it indicates to the edge router that this endpoint is willing to share resources for this new gate (Gate 2) with a previously created gate (Gate 1). As long as the QoS requested for the new gate can be satisfied with a bandwidth allocation equal to or less than the existing gate, no new bandwidth is reserved on the access network. However, bandwidth may need to be reserved in the backbone network depending on the end-to-end path taken by the new voice call. Access to the shared reservation occurs in a mutually exclusive manner: an endpoint has to reissue a COMMIT message to indicate to the edge router which flow is currently active. In the call waiting example, the endpoint sends a COMMIT message to the edge router to identify the currently active flow when the user switches between calls.

Our primary emphasis in DOSA has been a single service provider domain. For services spanning multiple providers, we envision bilateral agreements among the providers, with service level agreements at the boundaries. “Gates” might be implemented at provider boundaries, or, alternatively, admission control and policing at the boundaries might be done on aggregates to ensure that adequate capacity is available. The latter assumes a trust

relationship among providers. Settlements may be done via bilateral arrangements or via a trusted settlements authority, such as the AT&T Global Clearinghouse.

VI. ACHIEVING SCALE BY DISTRIBUTING STATE

In DOSA, once a call is set up between two endpoints, a DOSA proxy does not maintain state about active calls in the network. There are two major advantages to this design. First the reliability of the service does not depend on the reliability of an individual DOSA proxy. A DOSA proxy can fail without affecting active calls. Second, it removes many complex synchronization problems where two (or more) entities need to have simultaneously accurate information. Since interactions with the DOSA proxies are transactions without any residual state being retained at the proxy, it is not necessary for consecutive calls to be processed by the same DOSA proxy. Most calls in a transient state can be recovered and successfully established through a backup, secondary DOSA proxy using endpoint retransmission. We believe this design enables us to operate in large scale, cost effectively. It places the function of managing the state of a call where it belongs – at the endpoint. An existing call can only be affected by failures along the path or by failure of the endpoints: there are no unnecessary elements involved in a call.

Existing telephone network circuit switches must be designed to very high degrees of reliability since failure of the control processor can affect a large number of active calls. Similarly, with the Media Gateway Control Protocol (MGCP) [10], the telephony application is centralized in a *call agent* that is responsible for all telephony features and services. A call agent must either be available throughout the lifetime of a stable call or it must keep the state of all of its calls tightly synchronized with a backup call agent. This introduces complex recovery mechanisms into the design, resulting in a more costly, less scalable design. We note that, in the ITU Intelligent Network (IN) model for telephony, a switch maintains call state and can invoke service logic during an active call. In DOSA, we are able to implement most services, such as call forwarding no answer, at the endpoint, and we have found relatively few examples of post-answer services that need to be supported solely by a DOSA proxy without endpoint intervention, and none that require state to be maintained in the proxy.

There are services for which a DOSA proxy needs information about current or previous calls. Instead of keeping the state in the proxy, the proxy stores this information at the client, in an object known as a *state object*. State objects are in some ways like the cookies used by Web servers, which are also used to distribute state to the client for use later by the server. In DOSA, state objects are both encrypted and signed by the proxy to ensure the privacy and integrity of the information contained in the state object. The endpoint returns the information to the DOSA proxy when it is needed to implement specific features. The endpoint cannot interpret

the information in the state object and any attempt to tamper with it can be detected by the proxy.

A number of examples arise in the implementation of telephony features. When a user wants to return the last call (e.g., by dialing *69) the “call return” function is invoked. If the user had subscribed to the caller ID feature, the terminating endpoint could store information (phone number or IP address) associated with the last call. However, the user may not subscribe to the feature, or the originator of the previous call may have requested that this information be blocked in order to retain privacy. In this case, call return can be implemented, while keeping the caller’s identity private, by using a state object.

To do this, the DOSA proxy passes the originating caller’s phone number in a state object to the terminating endpoint. When the user at the terminating end wants to return the call it sends the state object in the SETUP message as a new dial-string type. The DOSA proxy decrypts the state object using the same key that it used for encryption, verifies the signature, and establishes the call.

State objects are also used to implement the “Call Trace” feature by which a user can report an obscene or harassing phone call to law enforcement agencies. The state object stored at the terminating endpoint contains identifying information for the last call it received. This information can be stored at the terminating endpoint for every call, even when the originating user’s identity must be kept private. When the user invokes Call Trace, the endpoint provides the state object to the DOSA proxy in a service-specific message, allowing the call to be traced. For three-way calling and call transfer services, the state object contains information about the remote gate that is used by the DOSA proxies to modify gate parameters.

The edge router maintains connection state at the “gate” for each authorized flow. The edge router is the logical place to store information that can be used to associated use of the resources with the telephony service. When a flow begins (the edge router receives the COMMIT from the endpoint), the edge router outputs a usage event indicating the time the flow began and parameters of a flow (e.g., bandwidth) along with the opaque customer data that the edge router received from the gate controller at gate setup. When a flow stops, the edge router outputs another usage event. These two events can be used to generate a complete call detail record for a simple call. In order to correlate usage events from the originating and terminating edge routers, we also include a billing identifier that is globally unique and guaranteed not to be reused for some time. Finally, for some telephony features, the billing model currently used in the PSTN involves split charging. For example, in call forwarding, the originator of the call pays for a call from itself to the dialed destination, while the dialed destination pays for a call from itself to the forwarded number. Since the edge router at the dialed destination is not involved in the call once it is forwarded, the billing information for both legs of the

call is stored in the gate at the edge router associated with the forwarded number.

By keeping billing information in the edge router, where resources are monitored and used, the DOSA proxies do not need to play a role in call termination. Messages to release resources are sent from an endpoint to the gate. During telephony feature invocation, the DOSA proxy manages the billing information stored in the edge routers. However, once the call has reached a stable state, DOSA proxies do not need to retain information about the call.

VII. PRIVACY

In the telephone network, the calling number and calling name delivery services provide the called party with caller information. However, users that want to keep this information private can subscribe to the calling identity delivery blocking service. In an IP environment since endpoints are not trusted, the network must ensure that private information is not sent to a remote endpoint. DOSA proxies can ensure that private information is not conveyed in signaling packets. In practice, IP addressing information may also provide the called party with information about the location or identity of the calling party. This section describes support for a network-based *anonymizer* service that can be used to keep IP addresses and end-to-end signaling exchanges private.

The anonymizer function is inserted in the bearer path between the originating and terminating endpoints. Each endpoint participating in a call sends its media and signaling packets to the anonymizer. The anonymizer maintains state and translates the IP addresses in those packets so that they can be routed to their destination. The desire for privacy can be requested in the initial SETUP message to the DOSA proxy, which locates an anonymizer, establishes the appropriate translations, and includes the IP address of the anonymizer in the SETUP/SETUP_ACK messages that are sent to the endpoints. The use of the anonymizer adds a level of indirection, hiding the IP addresses of each participant in the call from each other.

The anonymizer function can be implemented at an arbitrary location in the network. However, to optimize delay, we suggest using anonymizers at both ends of the call, where each anonymizer (possibly located close to an edge router) translates both source and destination addresses of packets that it forwards. The source and destination addresses in packets are from a "local" address space. Global addresses that convey location information are only used between the anonymizers.

VIII. SUMMARY

When the network is used for real-time applications such as telephony, it is essential to have differentiation at the IP layer for the flows associated with these applications. To achieve this, there is a need for access control to enhanced QoS. In this paper, we showed how application layer protocols, such as call signaling, can enable applications to negotiate with the network to obtain enhanced QoS in an authorized manner. The protocols in the Distributed Open

Signaling Architecture (DOSA) achieve this and provide the primitives for a provider to offer a robust service, while deriving revenue from the service and exploiting endpoint intelligence.

A key contribution of DOSA is its explicit coordination between resource management and call signaling. This allows the provider control over who accesses both feature functionality and network resources, enabling the provider to offer a service, rather than just transporting bits. DOSA's key features include:

- Encouraging and exploiting feature functions implemented in the end-system, in keeping with the Internet philosophy.
- Including resource management primitives to request access to enhanced QoS, while meeting user expectations of having the necessary resources before end-to-end human communication begins.
- Retaining state only where it is necessary thus enabling the service to scale.
- Allowing the service provider to offer services such as translation, privacy and support for customer profiles that provide value added to the customer
- Preventing fraud and theft of service, thus protecting the interests of the service provider.
- Allowing service providers to monitor and control usage, in order to manage the network and derive revenue from the service

In summary, we believe that there is a role for the network and service provider in an IP network, when supporting real-time services, while still accommodating intelligent endpoints. DOSA achieves these goals.

ACKNOWLEDGMENTS

The authors acknowledge the contributions of many individuals, especially those in the PacketCable DCS and DQoS Focus teams who have worked with us to realize the DOSA architecture and principles in industry standards.

REFERENCES

- [1] Bernet, Y. et al., "A Framework for Differentiated Services", Internet draft <draft-ietf-diffserv-framework-02.txt>, Feb 1999.
- [2] Wroclawski, J., "The Use of RSVP with Integrated Services", RFC 2210, Sept 1997.
- [3] Shenker, S., et al. *Specification of Guaranteed Quality of Service*, RFC 2212, September 1997.
- [4] Handley, M., H. Schulzrinne, E. Schooler and J. Rosenberg, SIP: Session Initiation Protocol, RFC 2543, Mar 1999.
- [5] Schulzrinne, H. and J. Rosenberg, "A Comparison of SIP and H.323 for Internet Telephony", Proceedings of NOSSDAV '98.
- [6] Cuervo, F., C. Huitema, K. Kelly, et al., "MEGACO Protocol", Internet draft <draft-ietf-megaco-protocol-04.txt>, Sep 1999.
- [7] Braden, R., et. al. *Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification*, RFC 2205, Sept 1997.
- [8] Boyle, J. et al., "The COPS (Common Open Policy Service) Protocol", <draft-ietf-rap-cops-08.txt>, Nov 1999.
- [9] Yavatkar, R. et al., "A Framework for Policy-based Admission Control", Internet draft <draft-ietf-rap-framework-03.txt>, May 1999.
- [10] Atkinson R., et al., "Security Architecture for the Internet Protocol," RFC1825, Aug 1995.
- [11] Schulzrinne, H. and J. Rosenberg, "SIP Call Control Services," Internet draft <draft-ietf-mmusic-sip-cc-01.txt>, Jun 1999.
- [12] Marshall, W. et.al., "SIP Extensions for Supporting Distributed Call State", Internet draft <draft-dcsgroup-sip-state-00.txt>, Oct 1999.