MMUSIC Working Group
Internet Draft
Document: <draft-dcsgroup-mmusic-arch-00.txt>
Category: Informational

W. Marshall
K. Ramakrishnan
AT&T

E. Miller
G. Russell
CableLabs

B. Beser
M. Mannette
K. Steinbrenner
3Com

D. Oran
Cisco

J. Pickens
Com21

P. Lalwaney
J. Fellows
General Instrument

D. Evans
Lucent Cable

K. Kelly
NetSpeak

F. Andreasen
Telcordia

June, 1999

Architectural Considerations for Providing Carrier Class Telephony Services Utilizing SIP-based Distributed Call Control Mechanisms

Status of this Memo

The distribution of this memo is unlimited. It is filed as <draft-dcsgroup-mmusic-arch-00.txt>, and expires December 31, 1999. Please send comments to the authors.

1. Abstract

This document provides an overview of a SIP-based Distributed Call Signaling (DCS) architecture to support carrier class packet-based voice, video, and other real time multimedia services. Companion documents [3,4,5,6,7] address a specific set of SIP 2.0 protocol extensions and usage rules that are necessary to implement the DCS architecture in an interoperable fashion.

The DCS architecture takes advantage of endpoint intelligence in supporting telephony services without sacrificing the network's ability to provide value through mechanisms such as resource management, lookup of directory information and translation databases, routing services, security protection, and privacy enforcement. At the same time, the architecture provides flexibility to allow evolution in the services that may be provided by endpoints and the network.

DCS also takes into account the need to manage access to network resources and account for resource usage. The SIP usage rules defined in the accompanying IDs specifically address the coordination between Distributed Call Signaling and dynamic quality of service control mechanisms for managing resources over the access network. In addition, the DCS architecture defines the interaction needed between network provided call controllers, known as a "DCS-proxy" (also called Gate Controller in this draft on occasion) for supporting these services.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [1].

3. Introduction

This document provides an overview of a SIP-based Distributed Call Signaling (DCS) architecture to support carrier class packet-based voice, video, and other real time multimedia services. The DCS architecture and the corresponding SIP protocol enhancements

(described in companion documents) are being developed as part of the cable industry's PacketCable initiative, managed out of CableLabs (see www.cablelabs.com). PacketCable is defining a series of interface specifications that will enable vendors to develop interoperable products for providing internet telephony and other multimedia services over DOCSIS-enabled cable data networks.
The DCS architecture described herein has its roots in the DOSA work performed by AT&T Laboratories ["Distributed Open Signaling Architecture"; Kalmanek, Marshall, Mishra, Nortz, Ramakrishnan, et al.; October, 1998]. AT&T contributed DOSA to PacketCable in March, 1999 as a framework proposal for DCS. Earlier exploratory work by AT&T and a select group of vendors had demonstrated that the DOSA concepts could be expressed in terms of SIP signaling messages. Since the AT&T submission, a relatively large group of vendors have cooperated in an intensive effort to develop the DCS architecture and SIP protocol extensions described here and in the accompanying protocol drafts. Although DCS was originally designed with cable access networks in mind, the SIP signaling enhancements have general applicability to carrier class VOIP services running over QoS enabled IP networks.

The authors are submitting this draft to the IETF in order to provide general information regarding the DCS architecture and to convey the motivation behind the SIP enhancements recommended in the accompanying protocol drafts. We believe that incorporation of the concepts and mechanisms described in this set of drafts by the IETF into the SIP standard will significantly enhance SIP's ability to function as a carrier-class signaling protocol. Such an enhancement to SIP would undoubtedly aid in its widespread acceptance and deployment.

3.1 Background and Motivation

The design of the Distributed Call Signaling (DCS) architecture recognizes the trend towards use of packet networks as the underlying framework for communications. These networks will provide a broad range of services, including traditional best-effort data service as well as enhanced, value-added services, such as telephony. At the same time, improvements in silicon will reinforce the trend towards increased functionality in endpoints. These intelligent endpoints will take advantage of the widespread availability of packet networks to enable a rich set of applications and services for users.

However, when the network is used for real-time telephony applications, it is essential to have service differentiation at the IP layer. The ability to control and monitor usage is needed for the provider to be able to provide service differentiation and to derive revenue from the enhanced services. At the same time, the availability of best effort communications and the migration of functionality to the endpoints pose a challenge to the provider to find incentives for users to use or pay for enhanced services.

We see three key functions that a provider can offer, as incentives to use enhanced services. First, the network service provider has the unique ability to manage and provide network layer quality of service. When users depend on the quality of the service, as with telephony, there is a strong incentive to use the enhanced service, rather than a best effort service. Second, the network service provider can play an important role as a trusted intermediary. This includes ensuring the integrity of call routing, as well as ensuring both the accuracy and the privacy of information that is exchanged. The service provider can also add value by ensuring that services are provided consistently and reliably, even when an endpoint is unavailable. Finally, there are a number of services that may be offered more efficiently by the network service provider rather than in endpoints. For example, conference bridging may be more cost effective to implement in a multi-point bridge rather than in every endpoint attached to the network.

A key contribution of the DCS architecture 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 is designed to meet the user expectations and human factors associated with telephony.   For example, the called party should not be alerted until the resources necessary to complete the call are available. If resources were not available when the called party picked up, the user would experience a call defect.   In addition, users expect to be charged for service only after the called party answers the phone. As a result, usage accounting starts only after the called party picks up. Coordination between call signaling and resource management is also needed to prevent fraud and theft of service. The coordination between DCS and Dynamic QoS protocols ensures that users are authenticated and authorized before receiving access to the enhanced QoS associated with the telephony service.

It is important to be able to deploy a residential telephone service at very large scale, cost-effectively.   To achieve this, DCS minimizes the messaging overhead on network call servers, and does not require these servers to maintain call state for active calls. Once a call is established, call state is maintained only where it is needed: at the endpoints that are involved in the call, and at the ERs in the bearer path that are providing differentiated service to the media flow. This allows the network call servers to scale to support more users, and imposes less stringent reliability requirements on those servers.

DCS is also designed so that calling users receive consistent service even when a called endpoint is unavailable. For example, when an endpoint is unavailable service logic in a network call server can forward telephone calls to a voice mailbox.

3.2 Requirements And Design Principles

In this section, we briefly describe the application requirements that led to a set of DCS signaling design principles. In its most basic implementation, DCS supports a residential telephone service comparable to the local telephone services offered today. In addition to the commonly used service features that need to be supported, there are important requirements in the areas of reliability, performance, and scalability that influence the signaling architecture. Supporting an IP telephony service comparable to the telephony service offered today requires enhanced bearer channel and signaling performance, including:

@ Low delay - end-to-end packet delay must be small enough that it does not interfere with normal voice conversations. The ITU recommends no greater than 300 ms roundtrip delay for telephony service.

@ Low packet loss - packet loss must be small enough to not perceptibly impede voice quality or performance of fax and voice band modems.

@ Short post-dial delay - the delay between the user dialing the last digit and receiving positive confirmation from the network must be short enough that users do not perceive a difference with post-dial delay in the circuit switched network or believe that the network has failed.

@ 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 so that the "hello" is not clipped.

We identify a number of key design principles that arise from the requirements and philosophy outlined above.

1. Providing differentiated network-layer quality of service is essential, while allowing the provider to derive revenues from the use of such differentiated services.

2. The architecture should allow, and even encourage, implementation of services and features in the intelligent endpoints, where economically feasible, while still retaining value in the network and network-based services

3. The architecture must ensure that the network is protected from fraud and theft of service. The service provider must be able to authenticate users requesting service and ensure that only those authorized to receive a particular service be able to obtain it.

4. The architecture must enable the service provider to add value by supporting the functions of a trusted intermediary. This includes protecting the privacy of calling and called party information, and ensuring the accuracy of the information that is provided in messages from the network.

5. The architecture must enable the service provider to give a consistent view of basic services and features even when customer premise equipment is unavailable, and allow users to take advantage of functionality that is provided in the network, when it is cost-effective and easy to use.

6. The architecture must be implementable, cost-effectively, at very large scale.

3.3 Distributed Call Signaling Architecture

The Distributed Call Signaling Architecture follows the principles outlined above to support a robust telephony service. Figure 1 introduces the key components in the architecture.

The architecture assumes a broad range of DCS-compliant endpoints that provide telephony service to the user including Media Terminal Adapters (MTAs) that may be integrated with a Cable Modem or is a standalone device, as well as other endpoints such as personal computers. The access network interfaces to an IP backbone through a system we refer to as the Edge Router (ER). The ER is the first trusted element within the provider's network and is considered to be the edge of the network for providing access to differentiated quality of service. The ER performs resource management, acts as a policy enforcement point and as a source of billing information.

DCS-proxies (GCs) process call signaling messages and support number translation, call routing, feature support and admission control. In the context of SIP, a DCS-proxy is a SIP proxy that is involved in processing and forwarding of SIP requests. GCs act as trusted decision points for controlling when resources are committed to particular users. Media servers represent network-based components that operate on media flows to support the service. Media servers perform audio bridging, play terminating announcements, provide interactive voice response services, etc.   Finally, PSTN gateways interface to the Public Switched Telephone Network.

```
                 +-----+
                 | MTA |                MTA: media terminal adapter
                 +-----+
                    |                   ER: Edge Router
    Access Network  |
                    |
                 +----+
                 | ER |
                 +----+
                    |    +-----------+
                    |----| DCS Proxy |
                    |    +-----------+
                    |
                    |    +------------+
  Backbone Network  |----|Media Server|
                    |    +------------+
                    |
                    |    +------------+
                    |----|PSTN Gateway|
                    |    +------------+
                 +----+
                 | ER |
                 +----+
                    |
    Access Network  |
                    |
                 +-----+
                 | MTA |
                 +-----+
```

Figure 1: A Simple view of Components of an IP Telephony Architecture used in a HFC Cable Environment.

Telephony endpoints are considered to be "clients" of the telephony service. Consistent with the design principles, the architecture allows a range of services to be implemented by intelligent endpoints. They collect dialed digits, participate in signaling and contain the service logic required for basic 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, except when it is in the endpoint's best interests to do so.

Access to network resources on a differentiated basis is likely to be controlled by the service provider. The ER receives resource management requests from endpoints, and is responsible for ensuring that packets are provided the QoS they are authorized to receive (either through packet marking, or through routing and queueing the packets as a specific QoS assured flow). The ER requires authorization from a network entity (on a call-by-call basis for the telephony service) before providing access to enhanced QoS for an end-to-end IP flow. The obvious point where this policy and control function resides is the DCS-proxy (also called gate-controller,

because of this responsibility for managing access to enhanced QoS). Thus, the ER is able to ensure that enhanced QoS is only provided for end-to-end flows that have been authorized and for which usage accounting is being done. Since the ER knows about the resource usage associated with individual IP flows, it generates the usage events that allow a user to be charged for service.

We introduce the concept of a "gate" in the ER, 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 DCS-proxy are granted access to enhanced QoS in the access and backbone networks. Gates are "opened" selectively for a flow. For the telephony service, they are opened for individual calls. 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 network 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.

The DCS-proxy, in addition to implementing many of the call control functions, is responsible for the policy decision regarding whether the gate should be opened. DCS sets up a gate in advance of a resource management message. This allows the policy function, which is at the DCS-proxy, to be "stateless" in that it does not need to know the state of calls that are already in progress.

DCS-proxies are typically organized in domains. A DCS-proxy is responsible for a set of endpoints and the associated ERs. While endpoints are not trusted, there is a trust relationship between the ER and its associated DCS-proxy, since the DCS-proxy plays a role as a policy server controlling when the ER can provide enhanced QoS service. There is also a trust relationship among DCS-proxies. Details of the security model and mechanisms are work in progress.

The DCS-proxy is designed as a simple transaction server, so that failure of a DCS-proxy does not affect calls in progress. A domain will likely have both a primary and a secondary DCS-proxy. If the primary DCS-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. This is possible because the DCS-proxy retains no call state for stable calls. We believe this design makes the DCS-proxy efficient and highly scalable, and keeps the reliability requirements manageable.

DCS supports inter-working with the circuit switched telephone network through PSTN gateways. A PSTN gateway may be realized as a combination of a media controller, media gateway, and a signaling gateway. A media gateway acts as the IP peer of an endpoint for media packets, converting between the data format used over the IP network and the PCM format required for transmission over the PSTN.

The signaling gateway acts as the IP peer of an endpoint for signaling packets, providing signaling inter-working between DCS and conventional telephony signaling protocols such as ISUP/SS7. A media gateway control protocol is used to control the operation of the media gateway from the signaling gateway.

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

3.4 Basic Call Flow

Figure 2 presents a high-level overview of a basic MTA-to-MTA call flow in DCS. Each MTA is associated with a DCS-proxy, which acts as a SIP proxy. When a user goes off-hook and dials a telephone number, the originating MTA (MTA-o) collects the dialed digits and sends the initial call message, called an INVITE in SIP, to the "originating" DCS-proxy (DP-o). DP-o verifies that MTA-o is a valid subscriber of the telephony service (using authentication information in the INVITE message) and determines whether this subscriber is authorized to place this call. DP-o then translates the dialed number into the address of a "terminating" DCS-proxy (DP-t) and forwards the INVITE message to it.

We assume that the originating and terminating DCS-proxies trust each other. DP-o augments the INVITE message that it forwards with additional information, such as billing information containing the account number of the caller. DP-t then translates the dialed number into the address of the terminating MTA (MTA-t) and forwards the INVITE message to MTA to notify it about the incoming call.

The initial INVITE message invokes call feature handling at the destination MTA, such as call-forwarding. Assuming that the call is not forwarded, MTA-t negotiates the coding style and bandwidth requirements for the media streams. The 200 OK response to the initial INVITE is forwarded back through the DCS-proxies.

```
MTA-o      ER-o       DP-o       DP-t       ER-t      MTA-t
```

```
    | Invite   |          |          |          |          |
    |(no-ring) |          | Invite   |          |          |
    |----------|--------->|(no-ring) |          |          |
    |          |          |--------->|          |          |
    |          |          |          |  Invite  |          |
    |          |          |          |(no-ring) |          |
    |          |          |          |----------|--------->|
    |          |          |          |          |  200 OK  |
    |          |          |          |<---------|----------|
    |          |          |          |          |          |
    |          |  Gate    |  200 OK  |Gate Setup|          |
    |          |  Setup   |<---------|--------->|          |
    |          |<---------|          |          |          |
    |   200 OK |          |          |          |          |
    |<---------|----------|          |          |          |
    |   ACK    |          |          |          |          |
    |----------|----------|----------|----------|--------->|
    |          |          |  Reserve |          |          |
    |          |          | Resources|          |          |
    |<---------|----------|----------|----------|--------->|
    |  Invite  |          |          |          |          |
    |----------|--------- |----------|----------|--------->|
    |          |          |          |          | 180 Ring |
    |<---------|----------|----------|----------|----------|
    |          |          |          |          |          | User
    |          |          |          |          |  200 OK  | Answers
    |<---------|----------|----------|----------|----------|
    |   ACK    |          |          |          |          |
    |----------|----------|----------|----------|--------->|
    |          |          |  Active  |          |          |
    |          |          |   Call   |          |          |
    |<---------|--------- |----------|----------|--------->|
    |          |          |          |          |          | User
    |          |          |          |          |   BYE    | Hangs Up
    |<---------|----------|----------|----------|----------|
    |          |          |          |          |  200 OK  |
    |----------|----------|----------|----------|--------->|
```

Figure 2: A Basic Call Flow, including Resource Management functions

In the figure, MTA-t sends a 200 OK message to DP-t. The 200 OK contains a subset of the capabilities in the INVITE message that are acceptable to MTA-t. DP-t sends a GATE-SETUP message to the terminating ER (ER-t), conveying policy instructions allowing ER-t to open a gate for the IP flow associated with this phone call. The GATE_SETUP message contains billing information containing the account number of the subscriber that will pay for the call.

DP-t forwards the 200 OK to DP-o. DP-o sends a GATE-SETUP message to the originating ER (ER-o) to indicate that it can open a gate for the IP flow associated with the phone call. Finally, DP-o forwards 200 OK to MTA-o. The initial INVITE request and 200 OK response contain a SIP Contact header to indicate the IP address of the remote MTA to be used for subsequent end-to-end SIP signaling exchanges. MTA-o acknowledges the 200 OK directly to MTA-t.

Once the initial INVITE/200 OK exchange has completed, both MTAs reserve the resources that will be needed for the media streams. Once MTA-o has successfully made its reservation, it sends a second INVITE message to MTA-t containing a command to ring the destination telephone. If MTA-t successfully reserved the resources needed for the call, it responds with a 180 Ringing to indicate that the phone is ringing, and that the calling party should be given a ringback call progress tone. When the called party answers, by going off-hook, MTA-t sends a 200 OK final response, which MTA-o acknowledges. At this point the resources that were previously reserved are committed to this conversation, and the call is "cut through."

Either party can terminate the call. An MTA that detects an on-hook sends a SIP BYE message to the remote MTA, which is acknowledged.

4. Resource Management

DCS'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 committed and made available to the endpoint, and recording is started so that the user can be billed for usage. The use of a two-phase 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 allocated in the first phase of the two-phase resource reservation protocol. This is important in order to limit the impact of backbone resource management on post-pickup delay (this minimizes the likelihood of clipping the first few syllables of the conversation).

5.
Distributed Call State

In order to provide enhanced services to millions of endpoints, we need an architecture that can be implemented cost-effectively at very large scale. Just as we enable flexibility by exploiting intelligence at the endpoints, services can be provided in a scaleable manner by storing the state associated with applications at the endpoints, rather than in network servers. Especially with telephony, endpoints are directly involved in handling calls and therefore need to maintain and use call state. In contrast, while network servers may need to be involved when setting up a call to gain access to enhanced QoS, there is no fundamental need for those servers to be involved throughout the lifetime of the call.

Maintaining state for every call at network servers, while achievable, increases the reliability requirements and load on the servers. The less state kept in the network, the better.

As a result, the DCS-proxy in DCS are designed to be stateless transaction servers. When a DCS-proxy processes a service request from an endpoint, it maintains state until the transaction is complete, but does not maintain any per-call 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 DCS-proxy. A DCS-proxy can fail without affecting calls that are currently in progress. Second, it removes many complex synchronization problems where two (or more) entities need to have simultaneously accurate information. Since interactions with the DCS-proxies are simple stateless transactions, it is not necessary for consecutive calls to be processed by the same DCS-proxy. DCS-proxy crashes affect only the transient calls (the number of calls in the process of being set up), and not stable conversations. Further, it is likely that most calls in a transient state can be recovered and successfully established through a backup or spare DCS-proxy using endpoint retransmission, with no explicit synchronization protocol required between the DCS-proxies. We believe this design principle will enable us to operate in very large scale, cost effectively. Furthermore 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.

We note that there are many services that involve the use of servers or proxy endpoints that communicate directly with clients. Since these endpoints are directly involved in providing service, it is necessary and appropriate for them to maintain state. Examples of proxy endpoints include application layer firewalls, caching servers, transcoders, network-based conference bridges, interactive voice response systems, and PSTN gateways. The DCS architecture models these as end-points, that maintain appropriate call state.

We now turn to the mechanisms that allow us to avoid state in the DCS-proxies. The DCS-proxy stores state information about an endpoint's calls at that endpoint in an object called a DCS-State (which is like a cookie). DCS-state cookies are both encrypted and signed by the DCS-proxy to ensure the privacy and the integrity of the information contained in the cookie. When needed, the endpoint provides the DCS-State to the DCS-proxy, which can use the information to provide additional functionality. Because the DCS-State contains information encrypted by the DCS-proxy, the information it contains is trusted by the network even though the endpoint itself is not trusted. In addition, the DCS-proxy stores service-specific opaque data associated with a call at the edge router. Since charging for telephony services is tightly tied to the use of resources, this information is best stored at the edge router, where knowledge of resource usage exists.

The endpoint returns the information to the DCS-proxy when it is needed to implement specific features. The endpoint cannot interpret the information in the cookie and any attempt to tamper with it can be detected by the DCS-proxy. There are a number of examples that arise in the implementation of telephony features. For example, when a user at an endpoint wants to return the last call (e.g., by dialing *69 on a traditional telephone) the "call return" function is invoked. If the user had subscribed to the caller ID service feature, the terminating endpoint could store the information (phone number or IP address) associated with the last call. However, it may be the case that the user does 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 DCS-state cookie.

6. DCS Proxy - DCS Proxy Communications

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

@ Authentication and authorization: Since services are only provided to authorized subscribers, DCS-proxies authenticate signaling messages and authorize requests for service on a call-by-call basis.

@ Name/number translation and call routing: DCS-proxies translate dialed E.164 numbers, or names, to a terminating IP address based on call routing logic to support a wide range of call features.

@ Service-specific admission control: DCS-proxies can implement a broad range of admission control policies for the telephony service. For example, DCS-proxies may provide precedence for particular calls (e.g., 911 calls). Admission control may also be used to implement overload control mechanisms, e.g. to restrict the number of calls to a particular location or to restrict the frequency of call setup to avoid signaling overload.

@ Signaling and service feature support: While many service features are implemented by endpoints, the DCS-proxy also plays a role in feature support. DCS signaling provides a set of service primitives to end-points that are mediated by the DCS-proxy. The DCS-proxy is involved in implementing service features that depend on the privacy of calling information, e.g., caller-ID blocking. It also plays a role in supporting service features that require users to receive a consistent view of feature operation even when an endpoint is down. For example, while an endpoint may normally participate in call forwarding, the DCS-proxy can control call forwarding on behalf of an endpoint when the endpoint is down.

End-points MTA-o and MTA-t communicate through the DCS-Proxies DP-o and DP-t, as shown in Figure 2. The interface of concern in this

section is the one between the DCS-Proxies DP-o and DP-t. In contrast to a true stateless SIP proxy, the DCS-Proxy maintains transaction state. During the interval that a call is being setup, a DCS-Proxy keeps state related to a request until a response is received.

For each call made to a phone number, DP-o may need to perform the functions needed for Local Number Portability (LNP). If a LNP database lookup is performed and the resulting dialed string is modified, DP-o must modify the Request-URI to include the result of the LNP lookup. When using a two-stage call setup, as described in [3], the information provided in the INVITE(no-ring), the originating proxy DP-o generates and stores the DCS-State: header. This information is intended to be sent to endpoint MTA-o and included with the final response that is returned to MTA-o. The originating DCS-Proxy, DP-o, may then use the call state information provided to it in the DCS-State: header to manipulate call-legs when requested by MTA-o.

As with conventional SIP proxies, DP-o adds its address to the top of the Via: header list with a branch=1 field when forwarding the request (e.g., INVITE(no-ring)). In addition, to support billing functions for a carrier, DP-o appends opaque pieces of information called the Billing-Info: and Billing-ID:. In addition, to support the resource management functions (such as manipulating Gates for resource management in concert with call-leg manipulation), a Gate-Location: header is included. This allows for the subsequent generation of requests for access network QoS by the end-points.

We also depend on originating DCS-Proxy, DP-o to be responsible for manipulating call legs. For instance, when a call is being forwarded, information about the new destination that the call is being forwarded to is provided by DP-t to DP-o. The new INVITE is then issued from DP-o. The information exchanged between the DCS-proxies enables such a function to be performed.

7. Privacy

Many conventional telephony systems have the ability to provide information about the identity of the calling party to the called party before the latter accepts the call (such a capability is typically termed "Caller-ID"). Systems that support Caller-ID usually provide a mechanism that allows the calling party to instruct the network to refrain from delivering this information to the destination.

In order for an IP-based network to provide a caller with a similar capability, a new SIP header is needed to signal the desire for anonymity to the network elements that would otherwise provide the caller's identity to the destination party. If a caller desires to remain anonymous, several additional changes to standard SIP are necessary.

In the cable IP network, the triplet {From:, To:, Call-ID:} is used to identify a call leg. Because call state information is pushed to the edge of the network, this information must be delivered to the destination endpoint.

The SIP From: header normally contains information that identifies the caller. In order to hide the identity of the caller, the From: header information is encrypted with the originating endpoint's key. The destination endpoint does not possess the key to decrypt the From: information.

Normally, the SIP Call-ID: header also contains information about the caller. In the DCS architecture, to support privacy in cable-based IP telephony networks the value of Call-ID: header is a cryptographic hash string that contains no information about the user.

Since all the normally-available mechanisms for passing information about the caller are no longer available, a new SIP header is used to pass the caller's identity to the destination if privacy has not been requested. This header is introduced that contains the information that would normally be present in the From: header; the network passes it to the destination endpoint only if the caller has not requested anonymity.

In addition to the usual privacy elements provided by telephone systems, IP-based systems must implement methods of hiding the source IP address from the destination if the caller requires privacy. The entire address must be obscured, since even a few address bits may provide partial location information. Likewise, IP addresses of the destination should not be revealed to the caller, in order to maintain privacy of transfer destinations.

IP addresses typically appear in the Contact: header; they also appear in SDP descriptions contained in SIP messages. These must all be protected. The mechanism we choose is an application-level anonymizer that inspects the SIP call signaling messages and replaces any identifying information contained therein in a consistent manner and in such a way that when the messages are delivered to the destination endpoint any identifying information has been replaced with fields that obscure the identity of the party seeking privacy.

This mechanism does not require any modification to the call signaling initiated by the endpoints: the application-level anonymizer performs these functions silently within the network.

8. Security Considerations

Detailed security considerations related to this architecture will be addressed in a future companion draft.

9. References

1. Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

3. "Integration of Resource Management and Call Signaling for IP Telephony", Internet Draft: <draft-dcsgroup-mmusic-resource-00.txt>, June 1999.

4. "SIP Extensions for Caller Privacy", Internet Draft: <draft-dcsgroup-mmusic-privacy-00.txt>, June 1999.

5. "SIP proxy-to-proxy extensions for supporting Distributed Call State", Internet Draft: <draft-dcsgroup-mmusic-proxy-proxy-00.txt>, June 1999.

6.

"SIP extensions for supporting Distributed Call State", Internet Draft: <draft-dcsgroup-mmusic-state-00.txt>, June 1999.

7. "Integration of Call Authorization and Call Signaling for IP Telephony", Internet Draft: <draft-dcsgroup-mmusic-call-auth-00.txt>, June 1999.

10. Acknowledgments

The Distributed Call Signaling work in the PacketCable project is the work of a large number of people, representing many different companies. The authors would like to recognize and thank the following for their assistance: John Wheeler, Motorola; David Boardman, Daniel Paul, Arris Interactive; Bill Blum, Jon Fellows, Jay Strater, Jeff Ollis, Clive Holborow, General Instruments; Doug Newlin, Guido Schuster, Ikhlaq Sidhu, 3Com; Jiri Matousek, Bay Networks; Farzi Khazai, Nortel; John Chapman, Bill Guckle, Cisco; and Chuck Kalmanek, Doug Nortz, John Lawser, James Cheng, and Partho Mishra, AT&T.

11. Author's Addresses

Bill Marshall
AT&T
Florham Park, NJ 07932
Email: wtm@research.att.com

K. K. Ramakrishnan
AT&T
Florham Park, NJ 07932
Email: kkrama@research.att.com

Ed Miller
CableLabs
Louisville, CO 80027

Email: E.Miller@Cablelabs.com

Glenn Russell
CableLabs
Louisville, CO 80027
Email: G.Russell@Cablelabs.com

Burcak Beser
3Com
Rolling Meadows, IL 60008
Email: Burcak_Beser@3com.com

Mike Mannette
3Com
Rolling Meadows, IL 60008
Email: Michael_Mannette@3com.com

Kurt Steinbrenner
3Com
Rolling Meadows, IL 60008
Email: Kurt_Steinbrenner@3com.com

Dave Oran
Cisco
Acton, MA 01720
Email: oran@cisco.com

John Pickens
Com21
San Jose, CA
Email: jpickens@com21.com

Poornima Lalwaney
General Instrument
San Diego, CA 92121
Email: plalwaney@gi.com

Jon Fellows
General Instrument
San Diego, CA 92121
Email: jfellows@gi.com

Doc Evans
Lucent Cable Communications
Westminster, CO 30120
Email: n7dr@lucent.com

Keith Kelly
NetSpeak
Boca Raton, FL 33587
Email: keith@netspeak.com

Flemming Andreasen

Telcordia
Piscataway, NJ
Email: fandreas@telcordia.com

Full Copyright Statement

Expiration Date This memo is filed as <draft-dcsgroup-mmmusic-arch-00.txt>, and expires December 31, 1999.