The Case for Services over Cascaded Networks

Anthony D. Joseph, B. R. Badrinath, and Randy H. Katz
EECS Department, CS Division
University of California, Berkeley
http://iceberg.cs.berkeley.edu/

Abstract

The widespread deployment of wired and wireless networks in both horizontal and vertical overlays presents new opportunities and challenges for developers of services. Users of these networks have multiple end devices, some of which are mobile, which complicates (and sometimes prevents) the transparent use of services across the networks. This paper presents an architecture for enabling the dynamic handoff of active services across multiple networks (cascaded networks). The architecture supports the rapid development of innovative new services and enables the use of service- and user-specific policies for transcoding and routing information between networks. These policies allow services to determine the "best" mode and route to use in reaching a particular user. The architecture described in this paper is being built on top of a campus-wide testbed consisting of GSM digital cellular, wireless and wired IP, and public switched telephone networks. An example application has been constructed to demonstrate the utility of the architecture.

1 Introduction

The proliferation of wired and wireless networks in both horizontal and vertical overlay deployments should enable better access to new and existing services. Users expect such services to operate across various networks. However, the use of services across such networks is often limited and difficult at best. For example, when a voice over IP user dials Enhanced 911[1] (E911), what location is passed to the emergency services operator — the closet containing the gateway? Furthermore, what happens when there are multiple potential paths through multiple networks (cascaded networks). For example, a user that is reachable through wired and wireless IP, cellular, telephone, and other networks.

The key problem of deploying services on cascaded networks is one of service handoff. Service handoff is the ability to pass an active service across network boundaries (*e.g.,* transferring an active telephone call from the Public Switched Telephone Network (PSTN) to a wireless IP phone). Service handoff also includes the ability to deploy and access a service in one network from another network.

Service handoff requires support for separating out the control (signaling, semantic information, and service metadata) and data information for each network, and propagating that information across each network boundary. Link-level gateways only provide a partial solution in that they perform low-level network-to-network bit transformations on the data, but they do not handle all of the metadata that is required by services [1, 8]. In the context of the Iceberg project (a joint project between the University of California, Berkeley and Ericsson), we are developing a solution to these problems that is based upon providing structured services that enable programmability over cascaded networks.

The Iceberg project is exploring architectures for combining voice and data services in third-generation (and beyond) digital cellular networks. These services are used by two types of entities: users (callers and callees) and agents acting on the behalf of one or more users.

1.1 Cascaded Networks

In the Iceberg project, we are exploring the issues associated with cascaded networks and collections of interconnected networks (see Figure 1 for an example of a cascaded network). Our research is based upon a diverse set of networks: IP, PSTN, digital cellular (GSM or CDMA), wireless IP, and 2-way pager networks [3, 4]. This set of networks covers a wide range of transport technologies, input/output interfaces, and user interfaces.

To test our ideas for the Iceberg architecture, we are constructing a testbed that will include all of these networks (see Section 2 for more information). In addition, we are also considering the issues associated with emerging networking

This work was supported in part by a grant from Ericsson, by the Defense Advanced Research Projects Agency under contract DAAB07-95-C-D154, by the State of California under the MICRO program, and by a grant from Intel. Computing and network resources for this research were provided in part through the National Science Foundation Research Infrastructure grant CDA-9401156. Badrinath was supported in part by the Defense Advanced Research Projects Agency under contracts DAAG55-97-1-0322 and DAAH04-95-1-0596.

[1] An emergency services operator in an Enhanced 911 system receives information about the caller's identity (the owner of the telephone line) and the location of the telephone line.

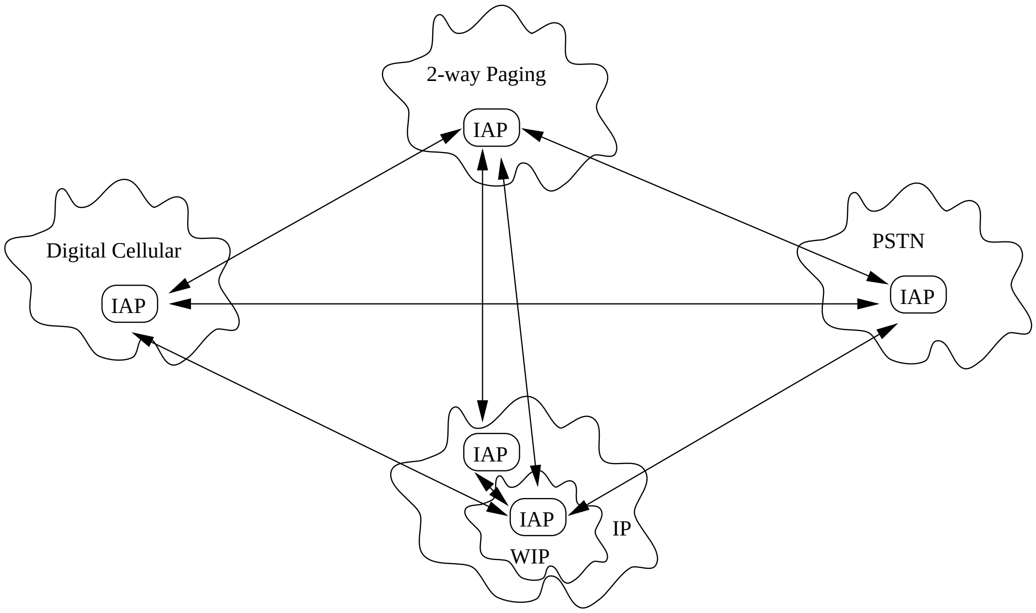

Figure 1: An example of a set of cascaded, interconnected networks. Iceberg Access Points (IAPs) provide gateway, routing, and security interconnections between the networks.

technologies, such as third generation digital cellular (UMTS / IMT2000) and Low-Earth Orbit satellites (LEOs).

1.2 Example Services

We envision many compelling services that could be offered over cascaded networks. For example, cross-network access to E911 services, Voice over IP (VoIP) access to speaker-independent Interactive Voice Response (IVR) systems, teleconferencing across multiple networks, and concierge or assistant services. In the following sections, we discuss some of the issues associated with supporting these services.

1.2.1 Cross-network access to E911 services

The primary benefits associated with E911 services are that the caller's telephone number and location are provided to emergency personnel. In the PSTN, identification and location information are determined by the user's "line," a customer information database that maps lines to users, and a geographic information database that maps lines to locations. This information is of critical importance when a user becomes incapacitated after contacting an E911 service. The information is also useful in dealing with false calls. Thus, at a minimum, cross-network access to E911 must include identification and location information.

As an example, consider a user who presses a button on their 2-way pager to send an emergency message to an E911 operator. It is a trivial function to convert the paging message from a text message into a speech message that can be played for an operator, but providing the location information is not nearly as easy or simple.

Using triangulation, the paging network could determine the user's location. However, there is no mechanism for passing this service-specific information through a paging gateway to the operator located in a telephony network.

A similar problem occurs if the emergency request is from a user using a VoIP gateway, either directly or indirectly? The location information isn't the location of the VoIP gateway. For a fixed machine, determining the location requires a geographic database that maps callers' IP addresses to locations. For a mobile machine, the location of a nearby wireless basestation may be sufficient. However, for a dialup users, it would be necessary to use caller identification and propagate the location based upon the dialup user's telephone number.

For each situation, the architecture needs to support the propagation of information across each network interface and, in some cases, multiple cascaded networks.

The solution in this example is to leverage local network support for locating callers and to provide an architecture that supports the passing of service-specific metadata between networks.

1.2.2 Voice over IP (VoIP) to Interactive Voice Response (IVR) service

This type of service is becoming very popular, as it allows any-to-any (multi-modal) access to information services and resources through a variety of end devices. However there are are a number of issues that occur when considering cascaded networks. For example, the IVR service may be located within the IP network or it may be in the PSTN network. The user may have a choice between multiple networks that can be used to reach an IVR service and a decision must be made as to what is the best network (or mode) to use to meet the IVR service's voice quality requirements.

For example, using a VoIP service in the wide-area may introduce an unacceptable amount of jitter into touch tones or speech, making high-level processing actions like touch tone detection or speech-to-text conversion difficult or impossible. Thus, a decision may have to be made as to how to best route a "call" to an IVR service or where to perform high-level actions, like detecting touch tones or performing speech recognition. Routing and action location decisions are service-specific and will differ from environment to environment with the result that there will be instances where the only way to achieve acceptable results will be to choose one network over another or to choose to perform an action close to or at the speech source. For example, in the GSM digital cellular network, touch tones are sent as control signals from a mobile handset to the network, rather than being sent through the lossy voice codec (which would render the touch tones useless).

1.2.3 Interactive Voice Response to room control service

As an initial proof-of-concept, we have built the following cross-network service. The service uses speaker-independent IVR to control A/V equipment in a room (*e.g.,* moving cameras, turning lights on and off, etc.).

The operation of the service is as follows (see Figure 2):

1. The user provides speech input to the system through one of the networks — currently, microphone, wired or wireless IP, and GSM.
2. A speech recognition operator converts the speech to text. This step requires the selection of a language and an N-gram. The language is speaker-dependent, while the N-gram is a function of the target service, in this case, room control.
3. A Natural Language Processing (NLP) operator converts the text into commands for the devices. This step also requires the selection of a grammar that is specific to the target services.
4. The command is delivered to the devices in the room and the response is returned to the user, in the form of a text-to-speech response.

The Iceberg architecture dynamically composes the path operators based upon the source and destination, thus the system transparently supports alternative input and output devices (*e.g.,* the rooms can be controlled from text-based devices — pagers, PDAs, or other interfaces). This dynamic composition, or Automatic Path Creation (APC) is possible because the interfaces of operators are strongly typed. Using dynamic composition, extending the system to support new devices or services is only a few hours of work.

In addition to the data transformations that are performed by the various steps, control information (metadata) is passed in *both* directions in parallel with the data. The metadata specifying that the destination is a room control operator is used to select the appropriate N-gram's by the speech recognition operator and to select the appropriate grammar by the NLP operator.

The Iceberg architecture makes the process of extending this service to other target services much simpler, for example, providing IVR-based access to web-based services (*e.g.,* news headlines, stock market symbol reports, weather reports, search, or driving directions) and other information repositories (*e.g.,* a user's personal information management repository). Some of these services require different transcoding steps from the ones in the room control service. However, the Iceberg architecture allows developers to rapidly add support for such services by simply specifying the interface to the service. We are in the process of building several of these services as additional example services.

1.3 Classes of Services

The example services in the preceeding sections can be generalized to three classes of services over cascaded networks: cross-implementation of existing services on new networks, mapping an existing service in one network onto another network, and totally new services.

1.3.1 Cross-implementation of existing services on new networks

In this instance, services that offer similar functionality in different networks are mapped onto one another. For example, directory services of some form exist in all networks: 411 (PSTN), LDAP and DNS (IP), HLR / VLR (digital cellular), and HA / FA (mobile IP [5]). These network-specific services would be used as a part of a universal name to network-specific name resolution process.

Another example service is voicemail (PSTN), E-mail (IP), and pages (pager networks and digital cellular's Short Message Service). These services offer similar messaging capabilities in different networks. Interoperability requires converting the information to the appropriate format (*e.g.,* performing speech-to-text to map voicemail to pages or e-mail; or performing text-to-speech to convert pages or e-mail to voicemail).

1.3.2 Mapping an existing service in one network onto another network

These are services that exist in one or more networks and need to be mapped onto a new network or instantiated in a new network using other existing services in that network.

For example, consider billing services in the PSTN and digital cellular networks. In the PSTN, billing is done on a "line" basis, while in the digital cellular environment, billing is based upon a Subscriber Identification Module (SIM) card. The challenge is to map these billing services onto networks like IP or wireless IP, where there are multiple authentication mechanisms (*e.g.,* PGP, X.509, Kerberos, etc.) for entities.

Another service that exists in several networks is E911 (PSTN and digital cellular networks). E911 calls have priority over all other traffic in these networks. That same degree of priority must be conveyed across gateways and interfaces between networks (*e.g.,* other calls or traffic must be dropped if necessary). Also, semantic information like the identity and location of the caller must be conveyed to emergency services operators.

1.3.3 Totally new services

In addition to existing services, cross-network access to services will allow the development of totally new services, like concierge or assistant services, location-dependent services, best-mode information routing services, and push services. These services may be on-demand services, for example, a service that allows a user to dial *HOTEL on a digital cellular phone and have a list of local hotels (customized to meet the user's requirements) and rates read back to them with the option of being connected to a local reservation desk.

The services may also be agent-based services (*e.g.,* an agent running in the infrastructure that monitors your location, perhaps by monitoring location update messages in a digital cellular network, and alerts you of changes in traffic conditions for your area). Other agent-based services customized news headline clipping, and stock market symbol tracking and alerting.

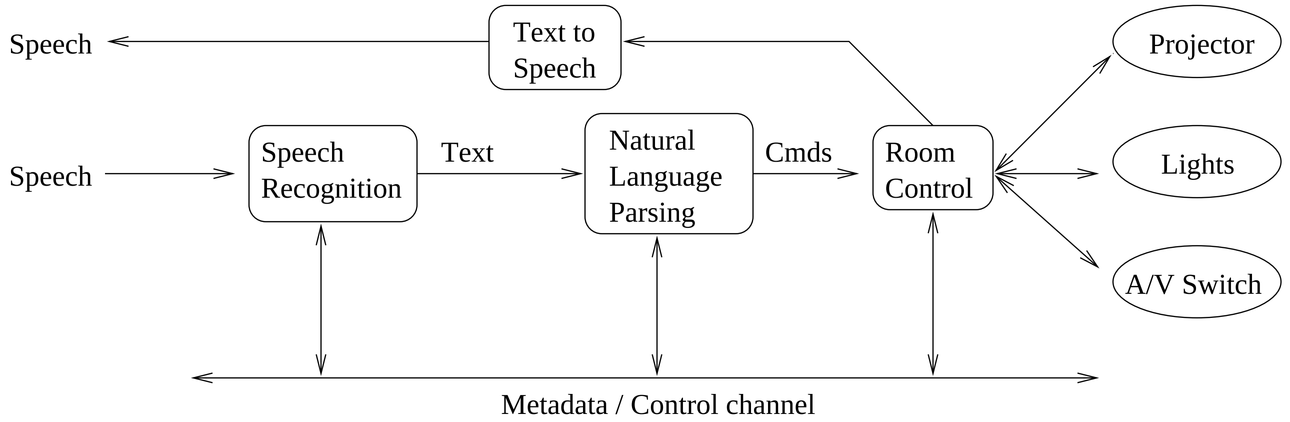

Figure 2: Interactive Voice Response to room control service. Each of the boxes represents an operator in the service, while the ovals are target devices. The Iceberg architecture uses the operator interface specification information passed along the metadata / control to dynamically construct the path from an input source to a destination device.

1.4 Paper organization

The rest of this paper is organized as follows: in Section 2 we discuss the Iceberg testbed and proof-of-concept application; in Section 3 we present the problems associated with providing services on cascaded networks; we detail the novel ideas in the Iceberg architecture in Section 4; in Section 5 we discuss related work; in Section 6 we define the design of the Iceberg architecture; finally, we summarize the paper in Section 7.

2 Iceberg Testbed and Applications

We are in the process of implementing the Iceberg architecture in the form of a campus-wide testbed and several services that will be used on a daily basis by the campus community. The usage of the testbed and the development of new services will provide us with an invaluable repository of access patterns and information about the effectiveness of the architecture.

2.1 Campus-wide Testbed

The Iceberg project will build upon several on-going research projects at the University of California, Berkeley, including the ProActive Infrastructure, Millennium, and Sim-Millennium projects. The ProActive Infrastructure project is building an architecture for providing parallel and distributed applications with a simple model for scalability and high availability, careful partitioning of persistent state, and dynamic location and composition of data transcoders. The Millennium and SimMillennium projects are campus-wide projects to provide scalable, reliable computing infrastructure.

Building upon this substantial infrastructure, the Iceberg project will undertake the following efforts: developing a cross-network services architecture; providing a toolkit for constructing such services; and building a campus-wide testbed and deploying real-world services.

The architecture will provide a means for specifying link-level transformations and service-level metadata interfaces. The toolkit will allow service developers to rapidly build services that can easily be deployed across networks and that support service handoff across networks. The campus-wide testbed will include wired and wireless telephony and IP networks that encompass the latest networking technologies, including early prototypes of digital cellular and high-density wireless LAN technologies. A significant portion of the testbed will be operational by early 1999.

In addition to being used in a research context, this project will include several services that will be used for educational purposes. The testbed's technology will be deployed throughout the UC Berkeley campus and in several classrooms. The toolkit will be used in conjunction with PDA hardware provided by IBM and our other industry partners to build collaborative learning services for student use (*e.g.,* instant messaging and access to web information, shared class repositories, and personal information management applications).

2.2 IVR Room Control Service

Using an early prototype of the Iceberg architecture and testbed, we have built a proof-of-concept service: the IVR room control service presented in the previous section (see Section 1.2.3). The service supports access from a variety of networks and enables the control of the A/V resources of several classrooms and conference rooms in our building. The service supports the Iceberg architecture's separation of metadata from data and it uses metadata to control the service's operation (see Section 3.2). XML descriptions are used to specify the interfaces to services and to help dynamically construct the path from requestor to requested service.

For example, the speech recognizer uses the metadata information provided by the room control operator to choose the appropriate N-gram (in this case, a customized room control N-gram). The service currently supports directly typed commands, and speech input from local microphones, the Mbone audio conferencing tool (VAT) over wired and wireless IP networks, and GSM networks. In the near future, it will support access from PSTN networks.

3 Problems with Services on Cascaded Networks

Cascaded networks introduce several significant problems that greatly complicate service handoff. In this section, we discuss how the Iceberg architecture addresses the following problems:

1. Multiple endpoints in multiple networks.
2. Propagation of metadata across / through cascaded networks.
3. Entity- or service-specific customized information collection, delivery, and dissemination.

3.1 Multiple Endpoints in Multiple Networks

Part of the complexity of cascaded networks is that users have multiple end devices (*e.g.,* cell phones, pagers, computers, laptops, PDAs, office and home telephones, etc.) and some of the devices have multiple network interfaces (*e.g.,* a PDA with cellular, pager, and wired network interfaces). The choice of device or interface may be a function of several parameters: connectivity (*i.e.,* users and devices are mobile, so connectivity and reachability changes dynamically), quality of service, cost, etc. The important consideration is that the user should be able to specify their desired choices for routing (see Section 6.2.2 for more information).

Mobility in cascaded networks takes one of two forms: terminal mobility and user / personal mobility.

- Terminal mobility takes two forms: intra- and inter-network mobility. Intra-network mobility is movement within a network (*e.g.,* handoff between wireless IP basestations or handover between cellular basestations). The underlying network handles, and in some cases, controls this type of movement. In some networks, the movement will be internally or externally visible, while in others, movements are hidden. Control over handoff and handover may reside with the mobile device (*i.e.,* mobile IP) or with the network (*i.e.,* GSM).

  The second form of terminal mobility is inter-network movement (*e.g.,* handoff between a cellular interface and a wireless IP interface). This type of handoff requires significant interoperation between the two networks to coordinate the handoff (more details later in Section 3.2).

- User / personal mobility involves users with multiple devices switching between the devices. Users will switch for a variety of reasons (*e.g.,* location, cost, quality of service, etc). Some of the movements will be within a network, like moving from one workstation to another or from one office phone to another, while others will be cross-network movements, like switching between a wireless IP phone and an office phone.

The Iceberg architecture uses indirection to hide both terminal and user / personal mobility. The architecture associates a globally unique, non-domain-specific Universal Name with each users or agents. Universal names along with an inter-domain naming protocol provide a mechanism for entities to refer to other entities in a non-domain-specific manner (discussed in more detail in Section 6.2.2).

3.2 Service Handoff Across Cascaded Networks

Service handoff requires support for passing metadata across one or more network boundaries to the target network at the time of a handoff. Such metadata is service- or network-specific and includes information like, the caller's identity and authentication rights, and billing or cost information. In addition, the information may include the target service being used, since this might affect where transcoding is performed or which transcoder is used, the requested QoS, or the path chosen through the network.

The key to supporting service handoff is to separate control information (metadata) from data and allow each to follow a separate logical path between the source and destinations. The physical paths may be the same or they may be physically distinct, as in the long-distance PSTN network, where call data is sent over a separate network from call set-up and control information.

The metadata may be passed across the same network interface as the data, or it may be passed to the new network through an alternate route. Once the metadata has been received by the new network, it is used to negotiate and optimize a path through that network, to setup the appropriate routing and transcoding, and to specify billing, authentication, and any other service- or network-specific information.

3.2.1 Service transformations

End-to-end cross-domain services raise the interesting question of where to perform data conversions, some of which may be very computationally intensive. Gateways are one place, however, as outlined earlier in the VoIP to E911 example, the choice of which gateway to use may affect the performance of later operations. In addition, the particular transcoding operation that is performed may also affect the performance of later operations.

Alternatively, computation may take place at computation servers within the network infrastructure (*e.g.,* at a network of workstations in an IP network). These large servers provide a scalable, highly available resource that can be allocated on-demand to support computation.

With cascaded services, there will often be cases where the service must negotiate with the underlying network for resources (*e.g.,* computational, call admission, QoS, etc.). Thus, it is important that services have control over where such conversions take place and what conversions are used.

For the same reasons, it is also important that services be given control whenever a service or user crosses a network boundary. At the boundary, the services can negotiate service-specific and network-specific attributes (*e.g.,* negotiating with the network's call admission policy or negotiating for a specific QoS).

The Iceberg architecture is designed to support both models of computation, including services that span both gateways and the infrastructure's computational resources, and to support the passing of metadata between operators.

3.2.2 Example: Cross-system service handoff

Consider a user talking on a cordless handset that is capable of wireless PSTN and wireless IP connectivity. As the user walks out of range of the PSTN cradle, the handset must negotiate with the wireless VoIP network to admit the call and for the appropriate bandwidth and QoS. Simultaneously, a PSTN core network component is contacted to bridge the call to the appropriate PSTN-to-wireless VoIP gateway, which then relays the call to the handset's wireless IP interface. When the user leaves the PSTN network, the PSTN link is dropped and the call seamlessly continues on the wireless IP network.

When the user re-enters PSTN coverage, the process is repeated in reverse and the call is restored to the handset's PSTN interface.

These actions require that metadata be passed to the appropriate gateways and resources in each network so that the call can be re-routed.

One significant aspect of cross-system service handoff is the decision-making process. The process may include network-based handover and mobile device-based handoff decisions. The key is that the overall process is controlled by users and services.

3.3 Customized Information Collection, Delivery and Dissemination

There are many cases where the infrastructure can be leveraged to route information in a dynamic, timely fashion. For example, a salesperson who has multiple end devices or end devices with multiple network interfaces could receive immediate notification when someone tries to contact them or the salesperson could have an service agent monitoring the availability of a particular product, etc. When triggered by the appropriate condition, the agent would then contact the salesperson using the best possible mode, as specified by the salesperson.

Another example is distributing the same information to multiple users on the same or different networks (*e.g.,* sending one message to users with cellular phones, pagers, voicemail, E-mail, etc.). Consider being able to type a message and have it delivered to a list of users, where the list not only contains e-mail addresses, but also contains cellular phone numbers, office numbers, pager numbers, voice mail numbers, etc. and the infrastructure transparently converts the message to the appropriate format as selected by the *recipient* of the message. This service simplifies and optimizes information delivery both for senders and recipients.

Sensor fusion is an example of information collection. Consider the problem of aggregating information from sensors in different networks (*e.g.,* a passive IR detector connected to an IP network and a video camera connected to a digital cellular network). The infrastructure must collect and correlate information from very different types of devices. This problem is similar to the problem of personal information management for a user (*e.g.,* voicemail, e-mail, pages, faxes, etc.), where the information has many different representations.

4 Novel Ideas in the Iceberg Architecture

There are several novel aspects to the Iceberg architecture that distinguish it from previous research in this area. The architecture provides support for the following features across cascaded networks: a carefully structured architecture for services, service transformations (not just bit transformations), and the ability to pass service- and entity-specific metadata across cascaded networks

4.1 Structured Architecture for Services Across Cascaded Networks

The Iceberg architecture addresses more than just one-to-one connections between networks [1, 2, 8]. The architecture includes the notion of cascaded networks, where there may be multiple paths between two networks, in addition to or in lieu of direct connections between the networks, and multiple places for services to execute within the networks. The architecture is not intended as a special-purpose or one-of architecture for interconnecting some existing networks, rather our intention is that the architecture is sufficiently broad enough that it can incorporate many existing and upcoming networks.

The traditional view of a network core is that the core services and resources should not be exposed to end users (or should be exposed only in a tightly controlled static manner). In the Iceberg architecture, we take the opposite view and provide secure, authenticated mechanisms for allowing service- and entity-specific policies to be injected into the network infrastructure.

Injecting code into the network is not something to be taken lightly, it requires support for secure execution and authentication of policies and entities. It also requires a careful structuring of the interfaces to resources the infrastructure. The Iceberg architecture gives entities control over the path that their data takes through the infrastructure and the resources that are used, but does not give them control over the flow of other entities' data or resources. The Iceberg architecture supports network-specific resource management, authentication, policies, billing, etc. to allow such policies to be controlled and enforced.

4.2 Not Just Bit Transformations, but Service Transformation

In addition to including more than just one-to-one network interconnections, the Iceberg architecture is also more than just bit-level transformations and simple network routing.

The Iceberg architecture addresses service transformations which involve decision making about: what bit-level transformations to perform, where to perform the transformations, high-level transformations to address deficiencies (*e.g.,* in the networks, network interfaces, and bit-level transformations), and how to route data across multiple networks, especially when there are multiple paths between networks.

Entities in the system are a key component of the architecture, especially the use of contextual information about them: who they are, what services they are using, where they are, and the capabilities or limitations of the available end devices and networks. This information directly affects the choice and placement of bit-level transformations.

Users may also have specific requirements (*e.g.,* cost, delivery latency, or QoS) for cross-network information. The network gateways and end devices may have a limited set of bit-level transformations, as such, it may be necessary to for end-devices, services, and network gateways to negotiate the particular bit-level transformation to use. Furthermore, the negotiation process may take place across or involve cascaded networks.

Routing is another example of a function that cannot simply be considered on a one-to-one basis, instead routing must be considered as a cascaded network function. The locally optimal choice of routing may not be the globally optimal function. For example, consider a call being placed between a user using VoIP on a computer in San Francisco and GSM user London. Routing the call over IP links might be optimal in the local area case, but could introduce excessive jitter and delay in the wide area case.

The Iceberg architecture is not trying to reinvent bit-level transformations or routing protocols. Instead, the architecture leverages off of existing services and management functionality in each network. Our concern is with the level of operation and scope of decision-making.

Many of networks are multi-modal, in that they can carry many different forms of information. This modality can be used as a benefit by cross-network coordination of where transformations should be located. However, reaching a global optimum in cascaded networks will require co-operation and communication between routing entities in multiple networks. Only in this manner, will it be possible to use high-level transformations, in addition to low-level transformations, to overcome the limitations of a given set of networks or their interfaces.

Also, networks contain many of the resource needed by services, although not in an abstract manner. For example, all networks have some form of naming and authentication. The Iceberg architecture unifies such mechanisms across the networks by providing an abstract naming and authentication mechanism that resolves to and relies upon network-specific mechanisms at the lowest level.

4.3 Propagation of Service- and Entity-Specific Metadata Across Cascaded Networks

As outlined earlier, service handoff requires the propagation of service- and entity-specific metadata across cascaded networks. The information must be propagated to the computational resources in each network that are involved in carrying the data associated with the particular service. The Iceberg architecture provides these propagation paths by providing logically or physically separate bi-directional metadata and data paths.

The Iceberg architecture provides infrastructure for specifying metadata in the form of XML descriptions. These descriptions are used during the creation or modification of a path (*e.g.,* during service handoff) between the requestor and requested service to choose which transcoders to use, where to place them, and how to route between them.

5 Related work

5.1 Domain Naming Service

The Internet's Domain Naming Service (DNS) by itself is not an acceptable naming protocol for the Iceberg architecture. DNS' name to address or information mappings are relatively static and limited amounts and types of information can be returned (*e.g.,* objects can't be returned).

DNS only provides timeouts for controlling the caching of mappings in servers and clients. Servers have no way of dynamically flushing or changing cached mappings. For example, implementing call forwarding using DNS would fail to work properly for an indeterminate amount of time after a user changed the forwarding mapping.

DNS supports one to many mappings, but does not specify a policy of which mapping a particular client should prefer or use. Instead the choice is left up to clients who typically have inconsistent policies.

Finally, DNS provides addresses, but does not specify the routes that should be used between hosts. Such information is managed by separate network routing protocols (*e.g.,* OSPF, RIP, etc).

5.2 Signaling System 7 (SS7)

Signaling System 7 (SS7) is the internetworking protocol used in telephone systems (*i.e.,* PSTN and digital cellular). SS7 provides a transport for building services with reliable message delivery and automatic failure recovery. SS7 enables telephone service providers to deploy many of the services discussed earlier in this paper. Such services, like dynamic call forwarding and routing, are being used as a basis for other services, like dynamic forwarding of calls to 800 numbers to local operators and local number portability, respectively.

The question is: should SS7 be used as the internetworking protocol for the Iceberg architecture? We believe the answer is no, for several reasons. SS7 is a very heavyweight solution and many services, like datagram delivery, do not require the overhead imposed by SS7. Also, SS7 was designed for closed networks and has limited security and authentication support. Security and authentication are strong requirements for an architecture that allows user code to execute in the network core.

The networking environment in Iceberg is very different from a traditional SS7 environment. The Iceberg architecture includes cascaded networks and routing and connectivity change on a very dynamic basis, both of these architectural aspects are outside the domain of SS7.

All of the negative aspects aside, SS7 has an important role in the Iceberg architecture. Interoperability between Iceberg and SS7 services and protocols is a requirement; as such, we will encapsulate SS7 services behind Iceberg services. This abstraction will serve to hide SS7's complexity from Iceberg developers.

5.3 Gateway Location Protocols

Gateway location (*e.g.,* for IP telephony [1, 6, 7]) is a requirement for the Iceberg architecture. However, our focus is on leveraging existing gateway location services as a component of a more abstract service discovery service.

5.4 Service Transformation

As with gateway location protocols, we intend to reuse existing solutions, but add a layer of indirection. Entities specify that they want a particular service between two cross-network endpoints and the Iceberg architecture handles the construction of the path by relying upon network-specific service location and transformation protocols and a language-independent means of specifying service interfaces.

6 Design of the Iceberg Architecture

The key principals in the Iceberg architecture are: entities and services. The task of the Iceberg architecture is to provide services between entities located in different or the same networks.

6.1 Entities

Entities are users, callers and callees, or agents acting on each other's behalf. Associated with each entity is a globally unique, non-domain-specific Universal Name that provides a mechanism for entities to refer to other entities in an abstract manner.

6.2 Services Architecture

Service information delivery requirements span a wide spectrum from real-time, interactive requirements (*e.g.,* a multimedia conversation), to information that can be delivered asynchronously. In addition, some information may have non-real-time delay bounds on delivery (*e.g.,* stock quotes). These delivery requirements are metadata that must be passed across network interfaces.

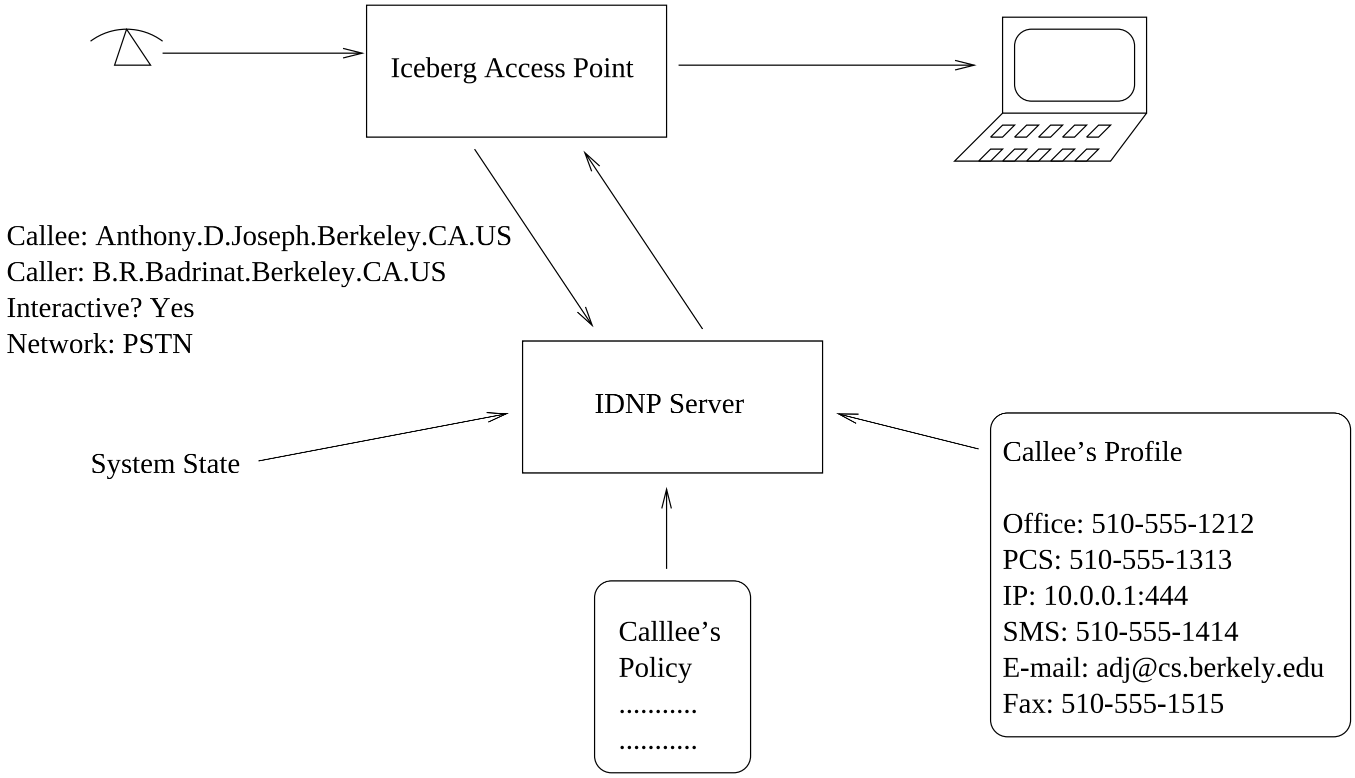

Figure 3: Iceberg Inter-Domain Naming Protocol example.

An interactive service may be based upon an information delivery mechanism that is connection-based or datagram-based. The underlying implementation is hidden from the user.

6.2.1 Service transformation

Iceberg Access Points (IAPs) are gateways located in each network that provide the interconnections between networks. To allow networks to connect to one another, the IAPs perform the link layer and bit transformations and they may also perform service-level transcoding. An IAP may be as simple as an H.323 gateway or it may be much more complicated, for example, providing metadata transport in addition to data transport.

The IAP may itself perform a data transcoding operation that has been requested by a service or it may choose to offload the operation to local computational resources in the network.

The IAP is usually the point where cross-domain (network-specific) signaling information is collected from an incoming network and potentially from the outgoing network. Such information includes authentication and identification information (*e.g.,* a PSTN caller's telephone number).

6.2.2 Cross-domain name resolution

An important requirement for service handoff is cross-domain name resolution, which allows users to provide a single globally unique name (Universal Name) and have it dynamically, resolved in any network to a domain-specific name using entity- or service-specific policies. Name resolution is handled by servers using the Iceberg Inter-Domain Naming Protocol (IDNP).

Iceberg Inter-Domain Naming Protocol (IDNP) servers map Universal Names to domain-specific names using three information elements: an entity profile that specifies the entity's domain-specific names, system state (for reachability information about an entity), and the entity's policy for mapping Universal Names to domain-specific names.

The entity profile is a list of domain-specific names for each of the end devices used by the entity. The entity profile changes on the order of weeks to months — basically when the entity acquires a new device.

System state or reachability information is domain-specific information about each domain-specific name. For example, for a PCS or digital cellular phone it might include cost, whether the phone is registered with a network (*i.e.,* reachable). While for a wireless LAN, it might include registration information and currently available QoS and bandwidth. System state usually varies on the order of minutes or hours, depending upon the entity's activities.

The entity's policy is the key difference between the Iceberg architecture and systems like the PSTN-based call routing approaches. The policy gives entities the ability to provide code that dynamically evaluates several variables and makes routing decisions. The policy is used once at call setup and again if system state information changes. Policies usually vary on the order of days to weeks.

Typical variables that are used in decision-making are: what is the caller's network, what is the cost for using each of the callee's available end devices, what QoS or bandwidth is available, who is the caller, is the service interactive or non-interactive, and any other entity- or service-specific information.

Entity-specific polices provide dynamic control over how an IDNP server maps Universal Names to specific domain-specific names. Figure 3 provides an example of a mapping.

Privacy is an important consideration; as such, the exposure of domain-specific names is under the control of entities. An entity's policy can specify that a domain-specific name should be returned to a caller or it can specify that the name should be hidden, in which case, the IDNP server informs the IAP that it should forward the call or message to the domain-specific name.

The IDNP executes on servers located in each network using portals provided by the network into the network's IDNP servers. For example, a special telephone number would be used in PSTN and digital cellular networks.

Information replication and caching are open questions in our design. Clearly, both are needed for scalability and reduced setup latency. We are experimenting with a number of different strategies. However, the maximum tolerable propagation delay is basically a function of the expected rate of change of information and the frequency of service connections being established to a particular universal name.

The rate of change for the information depends upon the information. Entity profiles change on the slowest time scale, followed by entity policies. System state or domain-specific reachability, however, may change very rapidly over the course of a few minutes or hours, as a function of user and terminal mobility.

IDNP servers cache information elements so that they can support low latency decision making. This means that the information may be replicated and cached at multiple IDNP servers and changes will need to be propagated to those servers in a timely and secure manner.

Wherever possible duplication of effort by IDNP will be avoided. In particular, name resolution will leverage off of each network's existing infrastructure. This means, for example, that the profile should contain a device's mobile IP home address rather than its current wireless IP address.

7 Summary

In this paper, we have described an architecture for providing services over cascaded networks. The need for service handoff across cascaded networks arises from the fact that users tend to carry multiple devices each of which may have multiple network interfaces; thus, the best network to reach a user may depend upon several factors.

The design of services over these networks requires a detailed consideration of the problems of interoperability over cascaded networks. We provide several examples of different classes of services over cascaded networks and discuss how the separation of control metadata from data helps with the construction of these services.

The Iceberg project is deploying a large-scale testbed consisting of GSM digital cellular, wired and wireless IP, and PSTN networks and we are beginning to experiment with cross-domain services. The Iceberg testbed and the services we develop on top of it will help validate the research, both through the demonstration of ease of service construction and deployment, and through the utility provided to students and the UC Berkeley community by such services and infrastructure.

References

[1] N. Anerousis et al. Architecture for signaling, directory services and transport for packet telephony. In *AT&T Labs - Research, NOSDAV 1998*, July 1998.

[2] Sandesh Goel, Partho Mishra, Huzur Saran, and Cormac Sreenan. Design and evaluation of a platform for mobile packet telephony. In *AT&T Labs - Research, NOSDAV 1998*, July 1998.

[3] David J. Goodman. *Wireless Personal Communication Systems*. Addison-Wesley Longman, Inc., Berkeley, CA, 1997.

[4] Rajan Kuruppillai, Mahi Dontamsetti, and Fil J. Cosentino. *Wireless PCS*. McGraw-Hill, San Francisco, CA, 1997.

[5] Charles E. Perkins. *Mobile IP: Design Principles and Practices*. Addison-Wesley Longman, Inc., Berkeley, CA, 1997.

[6] Jonathan Rosenberg and Henning Schulzrinne. A comparison of SIP and H.323 for internet telephony. In *Bell Laboratories, Lucent Technologies / Columbia University, NOSDAV 1998*, July 1998.

[7] Jonathan Rosenberg and Henning Schulzrinne. Internet telephony gateway location. In *Conference on Computer Communications (IEEE Infocom)*, San Francisco, California, March/April 1998.

[8] Henning Schulzrinne and Jonathan Rosenberg. Internet telephony: Architecture and protocols. *Computer Networks and ISDN Systems*, March/April 1998.