# The Deployment of Features in Internet Telephony

**By**
**Dongyang Zhang, B.Eng**

**A thesis submitted to**

**the Faculty of Graduate Studies and Research**

**in partial fulfilment of**

**the requirements for the degree of**

**Master of Applied Science**

**in Electrical Engineering**

**Ottawa-Carleton Institute for Electrical and Computer Engineering**

**Faculty of Engineering**

**Department of Systems and Computer Engineering**

**Carleton University**

**Ottawa, Canada, K1S 5B6**

**January 10, 2002**

**The undersigned recommend to**

**the Faculty of Graduate Studies and Research**

**acceptance of the thesis**

# "The Deployment of Features in Internet Telephony"

**submitted by Dongyang Zhang, B.Eng**

**in partial fulfillment of the requirements for**

**the degree of Master of Applied Science in Electrical Engineering**

______________________________

**Department Chair**

______________________________

**Thesis Supervisor**

**Carleton University**

**January 10, 2002**

# Acknowledgements

I would like to express my deepest gratitude to my great supervisor, Professor Bernard Pagurek, for his constant guidance, patience, inspiration, advice and encouragement throughout the research work and the preparation of this thesis. His constant striving for creative and rigorous research influenced many aspects of my life. I benefited greatly from his profound knowledge and experience in the field.

I am grateful to all my colleagues in the Network Management Lab, for their help. I would like to thank AT&T for providing the thesis project and financial assistance. I would also like to thank V. M. Bell for her help.

Finally, I would like to thank my family for their support, understanding and love during my study and research.

# Abstract

Internet telephony enhances existing telephone features and creates a number of new features with the integration of Internet services. Basic features can be divided into client features and network features according to their characteristics. Distributed Feature Composition (DFC) is a new architecture to describe telecommunication features. Session Initiation Protocol (SIP) and the Call Processing Language (CPL) presented by the Internet Engineering Task Force (IETF) can be regarded as a service architecture of Internet Telephony. This thesis discusses the deployment of features based on the SIP protocol and the DFC architecture respectively, which is mainly concerned with pure signalling messages. In the SIP protocol, it is more desirable to deploy most client features in end devices if end users are online. In the DFC architecture, it is better to place most client features inside the network. However, network features must be provided inside the network. The modularized DFC architecture introduces signalling overhead although it brings many advantages. Based on the DFC architecture, this thesis also sketches out the design of a voice mail feature. In addition, aside from pure signalling messages, other issues regarding the deployment of features will be presented.

# Acronyms

| | |
|---|---|
| **3WC** | Three Way Calling |
| **AD** | AutoDial |
| **CF** | Call Forwarding |
| **CFB** | Call Forwarding on Busy |
| **CFNA** | Call Forwarding on No Answer |
| **CFND** | Call Forwarding on No Device |
| **CFU** | Call Forwarding unconditionally |
| **CO** | Camp-on |
| **CR** | Call Return |
| **CRD** | Call Re-routing Distribution |
| **CW** | Call Waiting |
| **DFC** | Distributed Feature Composition |
| **ECLIPSE** | Extended Communications Layered on IP Synthesis Environment |
| **FB** | Feature Box |
| **FSM** | Finite State Machines |
| **GAP** | Call Gapping |
| **HTTP** | HyperText Transfer Protocol |
| **ICS** | Incoming Call Screening |
| **IETF** | Internet Engineering Task Force |
| **IMCW** | Instant message call waiting |
| **IN** | Intelligent Networks |
| **IP** | Internet Protocol |
| **ITU-T** | International Telecommunications Union – Telecommunications Standards Sector |

| | |
|---|---|
| **LI** | Line Interface |
| **LNR** | Last Number Redial |
| **MAS** | Mass Calling |
| **ML** | Mail List |
| **OCS** | Originating Call Screening |
| **PSC** | Personal Speed Calling |
| **PSTN** | Public Switched Telephone Network |
| **QUE** | Call Queueing |
| **SCF** | Selective Call Forwarding |
| **SCFB** | Selective Call Forwarding on Busy |
| **SCFNA** | Selective Call Forwarding on No Answer |
| **SDP** | Session Description Protocol |
| **SFB** | Simple Feature Box |
| **SIP** | Session Initiation Protocol |
| **SMB** | Spontaneous Message on Busy |
| **TI** | Trunk Interface |
| **TRA** | Call Transfer |
| **UA** | User Agents |
| **UAC** | User Agent Client |
| **UAS** | User Agent Server |
| **UML** | Unified Modelling Language |
| **URI** | Uniform Resource Identifiers |
| **VD** | Voice Dialling |
| **VM** | Voice Mail |
| **VPN** | Virtual Private Network |
| **WUC** | Wake-Up Call |

# Table of Contents

# List of Figures

# Chapter 1 Introduction

## 1.1 Background

Internet telephony refers to real-time voice or multimedia communications which are transported via the Internet. In a relatively short period of time, Internet telephony has experienced significant growth. Although there are a number of technical challenges, such as packet loss, delay and jitter, which affect the quality of voice over the Internet, Internet telephony has become especially attractive due to its low cost and high flexibility of service.

Currently, there are two main sets of standards for Internet telephony [14]. The first standard is the Session Initiation Protocol (SIP) provided by the Internet Engineering Task Force (IETF), the group that standardises protocols used on the Internet. The second is the H.323 suite of protocols developed by the International Telecommunications Union - Telecommunications Standards Sector (ITU-T). Both standards provide mechanisms for customer call establishment and teardown, customer call control and support for advanced services. Compared with H.323, the SIP protocol has more advantages in terms of flexibility in adding new advanced services and relative ease of implementation and debugging [13]. The SIP protocol and the Call Processing Language (CPL) [22] presented by the IETF can be regarded as a service architecture of Internet Telephony.

However, neither H.323 nor SIP clearly addresses feature interactions. Feature interactions refer to situations in which instances of same features or different service features affect each other [21]. Distributed Feature Composition (DFC) is a new architecture to describe telecommunication services, designed for feature modularity, feature composition, and separation of service and transmission layers [17]. The DFC architecture uses a precedence of features to avoid bad feature interactions and preserve good feature interactions. The DFC architecture is currently implemented on an IP platform to decide whether it can function as a service architecture for next-generation networks or not [8]. Therefore, the DFC architecture also can be regarded as a modularized service architecture of Internet telephony.

## 1.2 Motivation and Objective

A significant benefit of Internet telephony is the enabling of many new services [26] which are supported by all of the DFC architecture, SIP and H.323. In addition, Internet telephony also offers flexibility in moving intelligence from the network to end devices so that most of advanced services or features can be provided inside the network or in end devices. However, none of the service architectures of Internet telephony address feature deployment in detail. Where should features reside in Internet telephony? Exploring the deployment of features in Internet telephony is the main motivation of this thesis.

The overall objective of this thesis is to investigate features in current telecommunications, to explore the SIP protocol and the DFC architecture, and to discuss the placement of features in Internet Telephony based on the SIP protocol and the DFC architecture respectively.

## 1.3 Thesis Contribution

In this thesis, features are under review at the beginning of the project. Internet Telephony introduces a number of new services integrating voice, multimedia and data applications. Both the SIP protocol and the DFC architecture are also studied in order to compare the deployment of features in the different service architectures of Internet Telephony. Based on these two different service architectures, the deployment of some features, such as call waiting, call forwarding and voice mail, are analysed, mainly concerning with signalling messages, including messages for media controls.

Since the VM feature involves media controls and interacts with other features, and since it can be deployed in an end device or inside the network, this thesis sketches out the design of a voice mail feature which is based on the DFC architecture.

In addition, other factors beyond signalling messages also affect the deployment of features. In the real world, there are many issues related to the deployment of features, including the characteristics of the end device.

## 1.4 Thesis Organization

The remainder of the thesis is organized as follows. The second chapter briefly outlines features in telecommunication networks. This chapter also divides basic features into network features and client features according to their characteristics. The third chapter briefly describes the SIP protocol and its basic functionality. In the fourth chapter, the DFC service architecture and its implementation are described. In the fifth chapter, the deployment of network features is discussed first, followed by the deployment of client features. The discussion of client features mainly concerns signalling messages. In order to explore where client features should be deployed in Internet Telephony, this chapter also provides criteria for comparing client features placed in end devices with client features placed inside the network. The sixth chapter designs a voice mail feature based on the DFC service architecture. The seventh chapter describes other issues related to the placement of features, beyond signalling messages. Finally, the eighth chapter summarises the thesis and provides suggestions for related future work.

# Chapter 2 Features in Internet Telephony

Internet telephony enables a wealth of new feature possibilities. Features in traditional telephone networks, such as call waiting, can be enhanced through integration with Internet services. In the following sections, the features of traditional telephone networks and Internet telephony will be discussed briefly.

## 2.1 Services and Features

The words "services" and "features" are used in several different contexts. A service is offered by an administration to its customers in order to satisfy a specific telecommunication requirement [7]. It is defined as a stand-alone commercial offering, characterized by one or more core service features, and can be optionally enhanced by other service features [3]. ITU-T defines a service feature as the smallest part of a service that can be perceived by the service user [5]. Supplementary services modify or supplement a basis telecommunication service [6]. Supplementary services must be offered together with a basic telecommunication service and the same supplementary service may be common to a number of telecommunication services. A feature is a unit of one or more telecommunications or telecommunications management-based capabilities which a network provides to a user [2].

In today's telecommunication systems, "features" and "services" are roughly synonymous although "services" carry the connotation of being bigger [8]. In this thesis, both "feature" and "service" refer to the increments of functionality that are constantly being added to telecommunication networks.

## 2.2 Features in Traditional Telephone Networks

This section describes features found in traditional telephone networks. ITU-T published its accumulated descriptions of features in Annex B of Q.1211: Introduction to Intelligent Network Capability Set 1 [3]. In its follow-up document Q.1221: Introduction to Intelligent Network Capability Set 2 [4], ITU-T also addresses a large number of new features, including wireless services, multimedia services and service management services. Since these service/feature descriptions were compiled from disparate sources, the document acknowledges that they may be self- and mutually inconsistent.

Q.1211 divides services into two categories: "services" and "service features". (Both services and service features are referred to as features in this thesis.) A service is what an Intelligent Network (IN) vendor would actually wish to provide to customers. A service feature is a specific aspect of a service that can also be used in conjunction with other services or service features as part of a commercial offering. It is either a core part of a service or an optional part offered as an enhancement to a service [3]. So service features are low-level building blocks used to construct services.

Here are some examples of service features listed in the Q.1211 [3].

Call Queueing (QUE): allows customer calls which would otherwise be declared busy to be placed in a queue and connected as soon as the free condition is detected. Upon entering the queue, the caller hears an initial announcement informing the caller the customer call will be answered when a line is available.

Call Transfer (TRA): allows a subscriber to place a customer call on hold and transfer the customer call to another location.

Call Waiting (CW): allows a subscriber to receive a notification that another party is trying to reach his number while he is busy talking to another calling party.

According to the Q.1211, some particular service features are fundamental to some services, which do not make sense as commercial offerings to the service subscriber in the absence of these particular service features. But some service features are not core. For example, without a particular service feature, the name of the service would still make sense as a commercial offering to the service subscriber. Therefore, this particular service feature can be regarded as an optional enhancement to the service.

Some examples of services in the Q.1211 are listed as follows [3]:

Call Re-routing Distribution (CRD): permits the subscriber to have any incoming calls which encounter a triggering condition (busy, specified number or rings, queue overload or call limiter) be rerouted according to a predefined choice. The calls may be re-routed to another destination number (including pager or vocal box), re-routed on a standard or customized announcement, or queued.

Selective Call Forwarding on Busy/Don't Answer (SCF-BY/DA) allows the called user to forward particular pre-selected customer calls if the called user is busy or does not answer within X seconds or Y rings. The customer calls will be pre-selected based upon an SCF-BY/DA list.

Selective Call Forwarding (SCF) permits the user to have his incoming customer calls addressed to another number, no matter what the called party line status is, if the calling line identity is included in, or excluded from, a screening list.

In addition, telephony companies also develop their features in order to provide better services for their customers. For instance, Incoming Call Screening (ICS), Call Return (CR), Camp-on (CO) and Three-Way Calling (3WC) are features provided by telephony companies. The ICS feature allows a called party to automatically reject customer calls from certain callers. The CR feature lets users know the number of their last caller even when they cannot answer the phone. The CO feature allows a caller who reaches a busy destination to continue to re-try that destination periodically until the line becomes free. The 3WC feature allows the subscriber to add additional parties to an existing customer call.

Appendix B of this thesis lists the descriptions of features including services/features published by the Q.1211. Although these descriptions are compiled from various sources, including papers, books, ITU publications and companies' web pages, this thesis tries to make them consistent.

Features are designed and used by many telephony companies. Currently, there are no standard specifications for features. One feature may have different names. For example, Call Forwarding Unconditionally (CFU) allows its subscribers to forward their incoming customer calls to another phone number, no matter what the status of the called party line. Call Forwarding (CF) listed in the Q.1211 also allows the user to have his incoming customer calls addressed to another number, no matter what the called party line status may be. It is clear that the CFU feature and the CF feature refer to the same feature.

## 2.3 New Features in Internet Telephony

Internet telephony transports real-time media, such as voice or multimedia, over the Internet. Internet services can easily be combined with voice services to create a number of new services, which are not possible using traditional telephone networks. Web-based call centres and enhanced teleconferencing using shared whiteboard and shared applications are some of the new features which have been developed in the Internet environment [16].

Internet telephony enhances traditional features with the integration of Internet services, such as e-mail, web presence, instant messaging and directory services. For example, Instant Message Call Waiting (IMCW) notifies the called party of an instant message instead of a call waiting notification (a call waiting tone) when the called party is busy and receives a new call. Users can also have their calls connected to their web pages, listing different destinations. Customer calls could be recorded by standard Internet multimedia tools instead of being sent to a separate voice-mail system. Customer calls can be transferred or refused automatically during meetings using time-of-day routing interfaced with a Web calendar.

Internet telephony also creates new features. An e-mail Mail List (ML) can make many outgoing customer calls, either sequentially or in parallel, to send the same e-mail to every member of its list [31]. The presence feature is similar to instant messaging, but uses voice communications instead of text messages. Using the presence feature, users can see whether their buddy is online and may make a customer call to their buddy if he is online. Web-based call centres allow users browsing the Internet to initiate a customer call from an organisation's Web site to its call centre [16]. With Web-based call centres, an Internet surfer does not need to stop browsing and can speak to a call centre staff member, and then be given further information. The caller selection feature allows a call initiator to choose whom to talk to in the case where multiple parties answer a call.

One of the interesting new features in Internet telephony is "Home Phone", which is used in a standard residential phone service [28]. When someone calls a particular number, all the phones in the home ring. After a user picks up one of phones, all other phones stop ringing. A user can pick up from any other telephone in home and join an existing customer call. On the other hand, there can be multiple lines in the home so that a user on other lines can initiate a new customer call, while one or more are currently in progress. All the users involved in a single customer call are essentially involved in a multiparty conference, and are thus able to hear each other.

In addition, traditional features - here referring to features in traditional telephony networks - have some differences within the Internet environment. For example, Call Waiting (CW) is one of the traditional features: it sends the called party a notification that another party is trying to reach his number when the line of the called party is busy. However, on the Internet, "busy" is not "the line in use". In fact, there is no limit to the number of "lines" that can be present at end devices, in which users place and receive calls; the only limit is the number of simultaneous media streams. Due to the different "busy" concept, the traditional CW feature does not make sense in the Internet environment [30].

## 2.4 The Classification of Features

Features in telecommunications can be roughly divided into two groups: basic features and bundle features (as shown in Figure 2.1 below). A bundle feature is a package of basic features which allows telephony companies to provide better service to their customers, based on marketing considerations. For example, Bell Canada provides an "Accessibility Bundle" package that includes the CF feature, the CR feature, the CW feature and the 3WC feature.

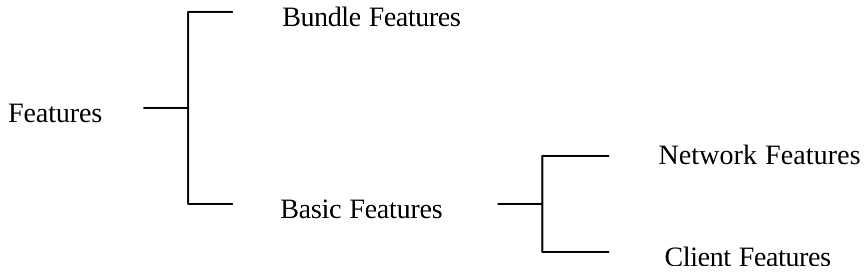


**Figure 2. 1 The Classification of Features**

A basic feature is one unit to provide base capabilities to a user. Basic features can be roughly divided into network features and client features.

Network features, such as 800 numbers and 900 numbers services, require the support of the network and usually involve users' location and the distribution of calls. For example, 800 numbers for toll-free services and 900 numbers for information services can be regarded as network features. They are geographically portable since they are mapped into routable phone numbers. Routable phone numbers and billing information are provided by the network. The user cannot identity the routable phone number without the support of the network.

Call gapping (GAP), which allows the service provider to restrict the number of calls to a served user to prevent congestion of the network [3], can also be regarded as another example of a network feature. A large number of users may call the same destination address simultaneously and this feature can be used to ensure that its servers and signalling network are not overloaded. The GAP feature requires the support of the network to prevent congestion.

Client features do not need the support of the network. They are usually heavily dependent on the state of the end device and on the specific content of media streams. They need some information which is available at the end devices. For example, the CW feature, which can be regarded as a client feature, must know the state of an end device. If an end device is busy, this feature will insert a call waiting tone into the media path. The CW feature will be performed if and only if the end device is in the busy state. Originating Call Screening (OCS) allows subscribers to authorize outgoing customer calls through the use of a screening list, which may be managed by subscribers. The OCS feature and the ICS feature are also examples of client features.

Most of the features in today's telecommunication systems belong to client features. Therefore, this thesis will primarily focus on client features.

## 2.5 Feature Interactions

Feature interactions refer to situations in which instances of the same features or different service features affect each other [21]. Bad feature interactions occur when several features operating simultaneously interact in such a way that they interfere with the desired operation of some features. This feature interaction problem has been well known for a long time. It has had a major effect on the Public Switched Telephone Network (PSTN), switches and IN. In 1988, Bowen et al [12] presented a framework for this problem and extensive research has been done in this field since then. However, the feature interaction problem has not been fully solved in existing telecommunication networks.

Two examples from traditional telephone networks illustrate the feature interaction problem. User A subscribes to the OCS feature, which blocks customer calls at an originating party based on an address to which a call is placed, and does not want customer calls to user C to be put through. If user A makes a customer call to user B who subscribes to the CFU feature with user C as the destination, the CFU feature will forward user A's call to user C unconditionally. In this case, user A may be confused.

Another example illustrates busy treatment features. A user may subscribe to several busy treatment features, such as the CW feature, Call Forwarding on Busy (CFB) feature and Voice Mail (VM) feature. The CW feature will send its subscriber a call waiting tone if its subscriber is busy and another party is trying to reach his number; the CFB feature will forward incoming calls to another address if its subscriber is busy; the VM will automatically take messages when its subscriber is unavailable or on another customer call. What happens if this user is busy?

Although Internet telephony is different to traditional telephone networks in many ways, the feature interaction problem still exists. As more and more sophisticated features are created and deployed in the Internet environment, the feature interaction problem becomes worse [8]. In addition, Internet telephony also introduces new types of feature interactions, probably combined with traditional telecommunication features.

The following example [11] illustrates the feature interaction problem of the traditional VM feature with Internet access. Today, many people access the Internet via a dial-up modem in their home. The telephony company' s VM feature will take callers' messages if its subscriber is on a call, is on Internet via a dial-up modem, or does not answer a call after a certain period of time. The VM feature will modify the dial tone on the subscriber' s phone if a message is left in the subscriber' s voice mail box. If the subscriber picks up the phone, he will know that a message has been left. However, the dial tone will not return to normal if the subscriber does not retrieve the message from his voice mail box. When the dial tone is in this special mode, the subscriber cannot connect to the Internet via a dial-up modem. Once he has retrieved the message in his voice mail box and the dial tone is returned to normal, the dial-up modem will work and he can access the Internet.

Some feature interactions are undesirable, like the examples above. But some feature interactions are desirable and necessary [33]. For example, there are many features dealing with busy signals, such as the CW feature, the CFB feature and the VM feature. If these features are properly integrated, they will treat a busy signal gracefully. Each feature can respond to a busy signal by priority and an enabling condition.

In Internet telephony, although new types of feature interactions are being introduced, a large amount of the work being done in traditional telephone networks for feature interactions will be useful in preventing or resolving bad feature interactions and preserving good feature interactions in Internet telephony.

In addition, feature interactions should be considered when deploying features in Internet telephony, which will be discussed in detail in section 7.2.

# Chapter 3 Session Initiation Protocol

Session Initiation Protocol (SIP), an application layer control protocol, is a principal standard that the IETF proposed for Internet telephony. It is used to establish, modify and destroy multimedia sessions or calls between one or more endpoints over the Internet. This chapter will briefly introduce the SIP protocol and its basic functionality.

## 3.1 The SIP Protocol Overview

The SIP Protocol is a textual protocol for establishing, changing, and terminating real-time calls or multimedia sessions over an IP network. It is similar to HyperText Transfer Protocol (HTTP) and is based on a client-server model, with requests sent by clients and responses returned by servers.

The SIP protocol is used in Internet telephony functions by mapping each function to one or more transactions. A SIP transaction occurs between a client and a server. A SIP transaction consists of a single request issued by a client, and one or more responses returned by one or more servers. In a typical SIP session, a SIP message originating at a user agent (UA), which usually resides at an end device, traverses one or more SIP network servers and then reaches one or more SIP user agents. However, SIP user agents can also communicate directly with each other.

In order to make a customer call or a session, SIP users must be identified. A SIP URL, which is an email-like address, “user@host”, is used to identify SIP users. The “user” portion of the SIP URL can be a user name or a telephone number, and the “host” portion can be a domain name or a numeric network address.

The SIP protocol can use both the TCP protocol and the UDP protocol as its transport protocols. For the UDP protocol, the SIP protocol provides its own mechanisms for reliability. The use of the UDP protocol provides for faster operation and better scalability than that of the TCP protocol. Besides the use of the TCP protocol and the UDP protocol on the Internet, the SIP protocol can also be used with ATM AAL5, IPX, frame relay or X.25 [27]. It makes minimal assumptions about the underlying transport and network layer protocols.

## 3.2 SIP Components

There are two components in a SIP system: user agents and network servers.

A user agent is a SIP-enabled end device. It uses a direction or an input for an end user and acts as an agent on its behalf to set up and tear down media sessions with other user agents. A user agent contains a User Agent Client (UAC) and a User Agent Server (UAS). The UAC client initiates a SIP request while the UAS server responds to a received SIP request.

In addition to UAS, the SIP protocol also provides network servers that accept SIP requests and respond to them. There are different logical types of entities in the SIP protocol: proxy servers, redirect servers and registrars.

A proxy server receives a request, makes a determination about the next server, and forwards the request to the next hop server, possibly after modifying some header fields. A proxy server typically accesses a database or a location server in determining the next hop. The interface between the proxy server and the location server is not defined by the SIP protocol. A SIP proxy server does not know whether the next hop to receive the request is another proxy server, a redirect server or a UAS server. A SIP request may traverse many servers on its way from the UAC client to the UAS server. Responses to the request always travel along the same sets of servers that the request followed, but in the reverse order.

A SIP proxy server can fork an incoming request and send it in parallel to multiple servers simultaneously. It is possible that each of the multiple servers generates its own response. The SIP protocol allows these responses to be merged and passed back upstream to the UAC client. After the UAC client receives multiple acceptances, the caller has multiple choices, depending on his system software. For example, the caller only accepts one of the responses and rejects the others.

A SIP proxy server can be stateful or stateless in a SIP transaction. A stateful server maintains a state of a SIP transaction. It keeps track of the requests and responses received in the past, and uses that information to process future requests and responses. A stateless proxy server receives a request, forwards it to the next hop and then forgets everything. A stateless SIP proxy server processes each SIP request or response based on its message contents. It does not store any request or response it has sent or received. A stateful server can transfer to a stateless server during a transaction. In general, larger and central SIP servers can be stateless while small and localised SIP servers can be stateful.

Unlike a proxy server, a SIP redirect server receives a request and tells the request's sender the next hop server which the sender can contact directly. It does this by responding to a request using a redirect response, which contains the address of the next hop. Like a proxy server, a redirect server also accesses a database or a location server to look up SIP users. After that, the location information is sent back to the sender in a redirect response.

In the SIP protocol, registrars keep track of users within their assigned network domains. They accept register requests, which are used to register SIP users' addresses. A registrar is typically co-located with a proxy server or a redirect server.

## 3.3 The Basic Operation

If a SIP user makes a customer call, a caller must find an appropriate server and then send a SIP request. The most important and common SIP operation is to invite a new participant to a customer call.

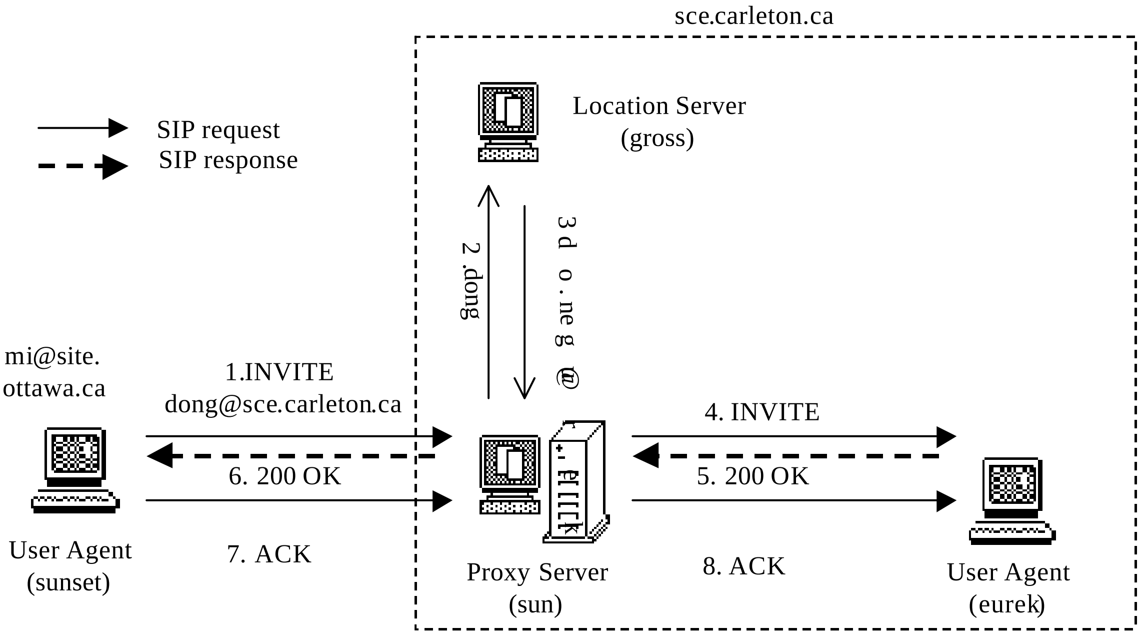


**Figure 3. 1 The SIP Invitation Working in Proxy Mode**

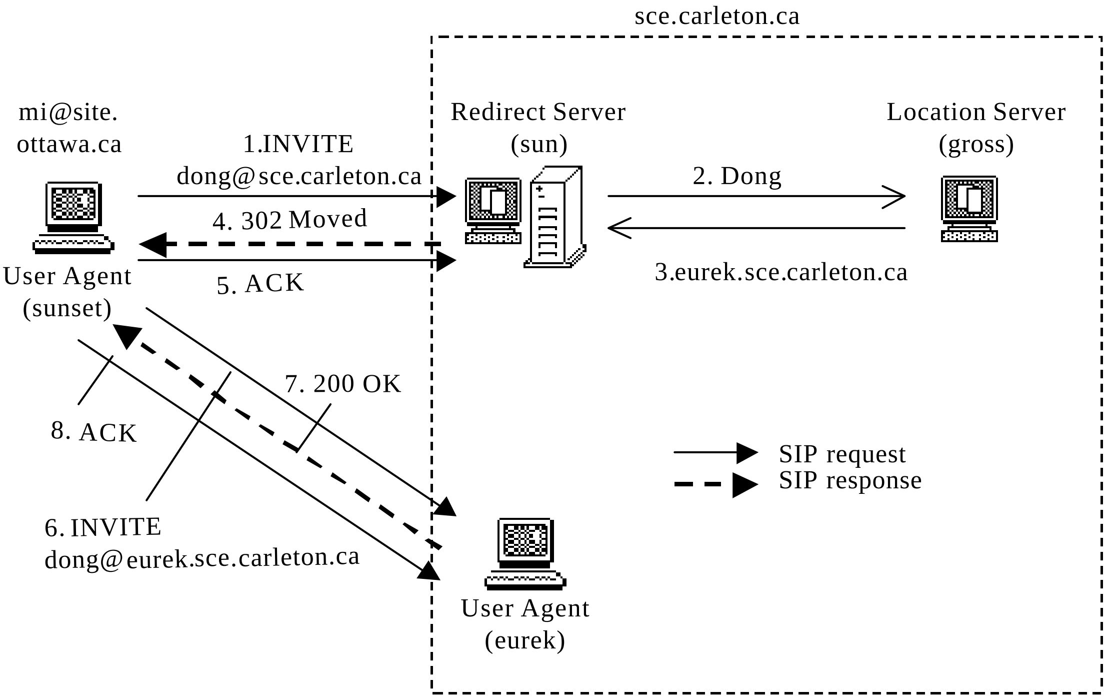


**Figure 3. 2 The SIP Invitation Working in Redirect Mode**

When making a customer call, a caller will first obtain a callee's address in the form of a SIP URL. If the host portion of the SIP URL is a domain name, the caller will try to translate the domain name to an IP address where a server may be found.

Once the caller gets the IP address of the callee's server, he can use the TCP protocol or the UDP protocol to send a SIP INVITE request to this server. However, this server may not be the server on which the callee is actually located. In that case, this server may forward the INVITE request to the current location of the callee (in proxy mode) or tell the caller to contact the next hop server directly (in redirect mode). Figure 3.1 and Figure 3.2 show a SIP invitation in proxy mode and in redirect mode.

After a UAS server is contacted, it will send the caller a response that includes a Status Code and a Reason Phrase. A Status-Code is a 3-digit integer result code that indicates the outcome of the attempt to understand and satisfy the request. A Reason-Phrase is intended to give a short textual description of a Status-Code.

## 3.4 SIP Messages

A SIP message is a request issued by a client or a response returned by a server. The SIP protocol reuses much of the syntax and semantics of the HTTP protocol, including its response code architectures and many message headers. Each of the SIP request messages and response messages contains a start-line, one or more header fields, an empty line indicating the end of the header fields and an optional message-body.

### 3.4.1 Request Messages

The SIP protocol defines six methods for its request messages. The six methods are INVITE, ACK, OPTIONS, BYE, CANCEL and REGISTER. INVITE is used to invite a user or service to participate in a session. The header fields for an INVITE request contain the addresses of the caller and the callee, the subject of the customer call, customer call priority, and other fields. The request body contains a description of the session.

```
INVITE sip:dong@sce.carletonca.ca SIP/2.0
 Via: SIP/2.0/UDP here.com: 5060
 From: Lucy <sip:lucy@here.com>
 To: Dongyang Zhang <sip:dong@sce.carleton.ca>
 Call-ID: 12345@here.com
 CSeq: 1 INVITE
 Subject: SIP will be discussed
 Contact:  Lucy <sip:lucy@here.com>
 Content-Type: application/sdp
 Content-Length: <appropriate value>

 v=0
 o=user1 53655765 2353687637 IN IP4 128.3.4.5
 c=IN IP4 224.2.0.1/127
 t=0 0
  m=audio 3456 RTP/AVP 0
```

**Figure 3. 3 An Example of a SIP INVITE**

A typical SIP INVITE is shown in Figure 3.3 [15]. The first line of the INVITE message is a request line that includes a method token, a request Uniform Resource Identifiers (URI) which is a SIP URL or a general URI, and a protocol version. “Via” traces the path of the invitation. “From” indicates the sender of the request and “To” indicates the recipient of the request. “Call-ID” identifies a particular invitation or all registrations of a particular SIP user. “Cseq” contains a request method and a single decimal sequence number that is unique to one Call-ID. “Subject” provides a summary of a customer call. “Contact” provides a URL where the user can be reached for further communications. “Content-Type” indicates the media type of the message body. “Content-Length” shows the size of the message body.

In a SIP INVITE message, if the Call-ID is new, this customer call is new. If the Call-ID is not new, and the originator of the request is already in the customer call, the message is either a duplicate, known by the “Cseq” field, or contains an update of the existing customer call. An update is usually executed silently by the server, without informing the user, as the user has already accepted the customer call. If the Call-ID is not new, but the originator of the request is not in the customer call, this is a new party being added to the existing customer call.

The ACK method confirms that the caller has received a final response to an INVITE request. The BYE method, which can be issued by a caller or a callee, is used to destroy a customer call. The OPTIONS method is used to find out the capability of a potential caller, but does not set up or tear down a customer call between a client and a server.

The CANCEL method is used to cancel a pending request with the same Call-ID, To, From and Cseq header values, but does nothing to a completed call. If the caller has already answered the call and receives a CANCEL request, the CANCEL request will have no effect.

A user uses a REGISTER request to register his address with a SIP server. A user agent may send a REGISTER request to a local server on start-up via a multicast address, “sip.mcast.net”. The REGISTER request should not be forwarded beyond the scope of an administrative system. The local server can build up a database of translations for those SIP users in its domain. A user agent may also be configured with the address of a registrar sever to which it sends a REGISTER request on start-up. The SIP protocol also supports a third-party registration.

Since SIP messages can be transmitted over unreliable transport protocols, the SIP protocol has to take care of reliability on its own. It defines two reliability mechanisms: one for INVITE requests and one for all other requests. Since locating the callee and waiting for a human to answer may take several seconds, an INVITE request usually cannot be answered immediately. A final response to an INVITE message is usually delayed. Thus, the client will retransmit the INVITE request until a provisional response arrives, and the server will retransmit the response until it receives an ACK request to confirm that the client has received a final response. For other requests, only the client retransmits these requests until a final response arrives.

### 3.4.2 Response Messages

After receiving and interpreting a request message, a recipient responds with a SIP response message. A SIP response message has a response statue code and a reason phrase. An example of responses is shown in Figure 3.4 [15].

The first line of a response message is a status line. The status line consists of a protocol version, a numeric status code and its associated textual phrase.

```
SIP/2.0 200 OK
Via:      SIP/2.0/UDP sce.carleton.ca: 5060
Via:      SIP/2.0/UDP here.com: 5060
From:   Lucy <sip:lucy@here.com>
To:       Dongyang Zhang <sip:dong@sce.carleton.ca>;tag=113448
Call-ID:  12345@here.com
CSeq:     1 INVITE
Content-Type: application/sdp
Content-Length: <appropriate value>
```

**Figure 3. 4 An Example of a SIP Response**

The Status Code is intended to be used by automata and the Reason Phrase is intended for the human user. The first digit of the Status Code defines the class of response. In SIP 2.0, there are six classes, shown in Figure 3.5 [15]. Other fields in a response message are similar to these in a request message. In general, the header fields of SIP messages are similar to the header fields of HTTP messages in both syntax and semantics.

1xx: Informational – request received, continuing to process the request;
2xx: Success -- the action was successfully received, understood, and accepted;
3xx: Redirection -- further action needs to be taken in order to complete the request;
4xx: Client Error -- the request contains bad syntax or cannot be fulfilled at this server;
5xx: Server Error -- the server failed to fulfil an apparently valid request;
6xx: Global Failure -- the request cannot be fulfilled at any server.

**Figure 3. 5 Six Classes of SIP Responses**

## 3.5 The SIP Basic Customer Call Set-up and Tear Down

Figure 3.6 shows a SIP basic customer call set-up and tear down (the sequence diagram is referred to as the call flow in the SIP protocol). User A makes a customer call to user B who is attached to the same proxy server. The customer call terminates when user A disconnects by initiating a BYE message.

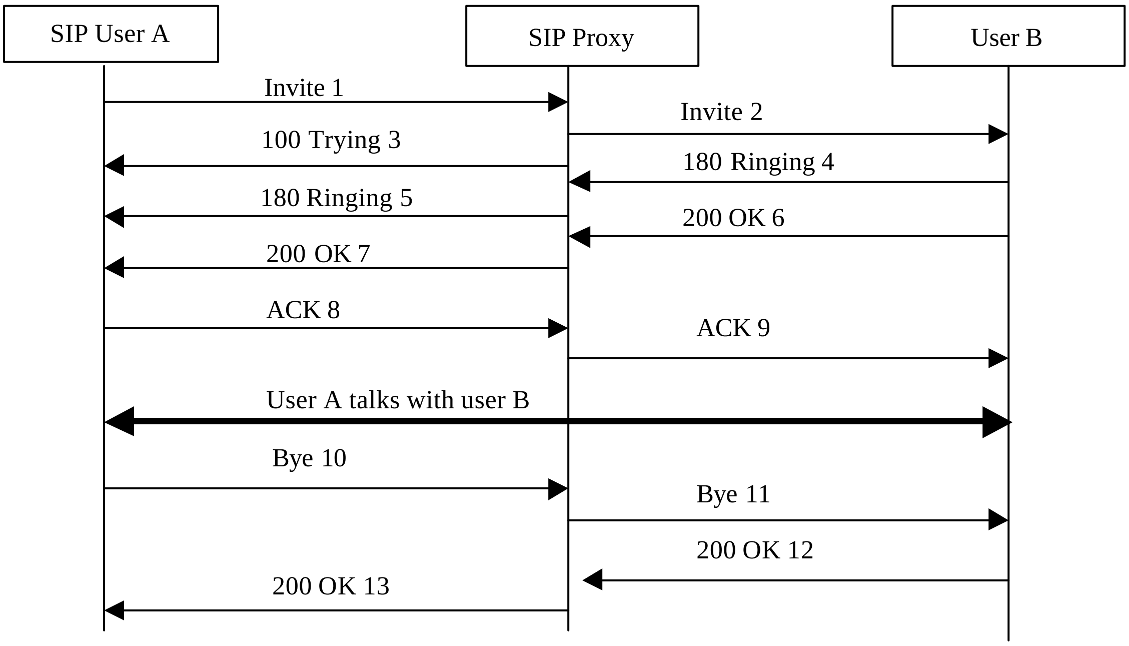


**Figure 3. 6 The Call Flow of A Basic Customer Call in the SIP Protocol**

First, user A sends an INVITE message to a SIP proxy server. The SIP proxy server responds with a TRY message and forwards the INVITE message to the target address, user B. User B responds to the INVITE message with a RING message and the proxy server forwards the RING message to user A. If user B picks up the phone, its UA will send an OK message that is forwarded to user A. After user B receives an ACK message issued by user A, a customer call between user A and user B is established.

Either user A or user B can destroy the customer call by sending a BYE message. The user who receives the BYE message will respond with an OK message.

# Chapter 4 Distributed Feature Composition

Distributed Feature Composition (DFC) is a new architecture used to describe telecommunication services. It was designed specially for feature modularity, feature composition and analysis of feature interactions. It treats features as independent components: boxes. In the DFC architecture, each customer call is processed by dynamically building a configuration of components and featureless internal calls. It applies the pipe-filter style in which featureless internal calls behave like pipes and feature boxes (FBs) behave like filters. The chapter will briefly describe the DFC architecture and its implementation.

## 4.1 DFC Components

DFC components are shown in Figure 4.1 [9]. Line interface (LI) boxes are interfaces to telecommunication devices, and trunk interface (TI) boxes are interfaces to trunks connecting other networks. Feature boxes (FBs) are implementations of features.

Boxes, including interface boxes and feature boxes, are the voice-processing components of the DFC system. Interface boxes include trunk interfaces, line interfaces and resource interfaces. Resource interfaces perform control-intensive media processing, such as recording, playing, mixing, monitoring and pattern recognition. Feature boxes are divided into two large categories: bound feature boxes and free feature boxes. A bound feature box is a unique, persistent, addressable unit while a free feature box is an anonymous, interchangeable copy of its type with no persistence outside its tenure in a usage.

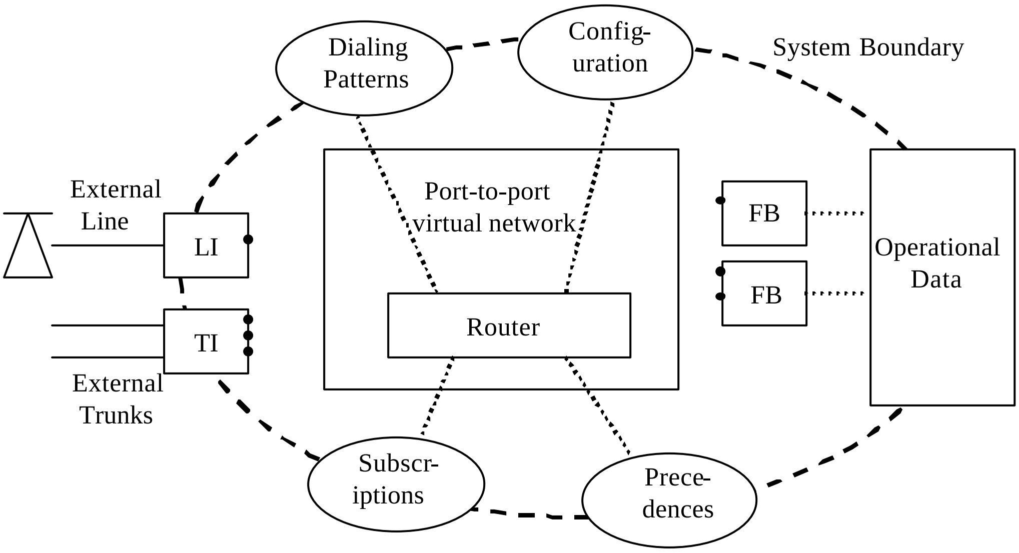


**Figure 4. 1 Components of the DFC System**

Each box has one or more ports which are used to place and receive featureless internal calls. Each port of a box can be a callee port, a caller port or a dual port. A callee port receives an internal call, a caller port places an internal call, and a dual port can play either role. The port-to-port virtual network carries internal calls. A trunk interface can have any number of DFC ports; and a resource interface can have one or more DFC ports depending on its size and the capabilities of the resource itself [32]. A feature box can also have one or more DFC ports according to the characteristics of the feature.

The router of the port-to-port virtual network routes internal calls from one box to another, so the router needs data on feature subscriptions, feature precedences, dialling patterns and normal configuration data. Feature subscriptions record compulsory features, such as Emergency Break-In (EBI), and customers' subscribed features. The EBI allows emergency services officials to break into an existing conversation. Feature precedences describe the order of the features, playing an important role in the DFC architecture. Configuration data record the set of existing boxes and the addresses of the interface boxes and feature boxes in the DFC system.

Feature boxes use global operational data, as shown in Figure 4.1. Access to operational data is currently partitioned by features, customers or both [19]. For example, the OCS feature will retrieve a subscriber's screening list from its operational data.

## 4.2 Internal calls

An internal call is a featureless connection between two ports on two different boxes. Each internal call begins with a set-up phase. In a set-up phase, an initiating dual port or a caller port sends a SETUP message to a DFC router. The DFC router determines the next box and forwards the SETUP message to that box. If the receiving box has no idle port to receive an internal call, the attempt to make a call fails.

Otherwise, that box might accept the call and send a message back to the initiating port. In some cases, the box might reject the call even if it has an idle callee port or dual port. For example, if a box' s idle port is reserved for calls with specific conditions, which the incoming call does not meet, this call will be rejected.

Any participating port can send a TEARDOWN message to terminate the call (teardown phase). Between the set-up phase and the teardown phase, the existing call has a two-way signal channel. The call can have any number of media channels [32], which can be initiated from any participating port. Each media channel can carry any medium and must be opened and closed explicitly by signals in the signalling channel. During a call, any participating port can send status messages to other ports along the signalling channel.

## 4.3 Usages

A usage that is dynamically created in response to an external service request is an assembly of boxes and internal calls. In a usage, feature boxes are selected by a DFC router in three zones: a source zone, a dialled zone and a target zone. Feature boxes in the source zone are applied to all calls made by the source caller; feature boxes in the target zone are applied to all calls directed to the target callee; and feature boxes in the dialled zone are applied according to the string dialled by a caller. The order of three zones is always the source zone, the dialled zone and the target zone (shown in Figure 4.2). The three zones correspond roughly to three obvious sub-chains in the construction of a usage.

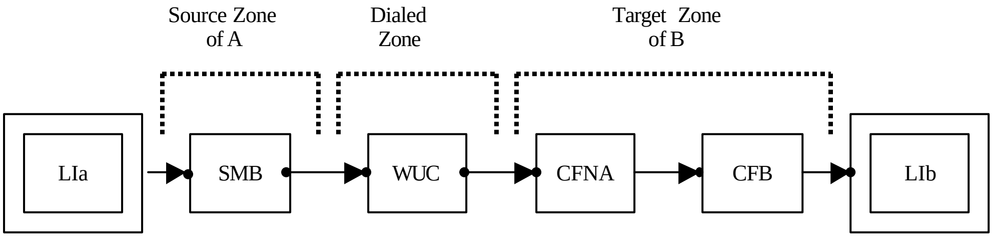


**Figure 4. 2 A Linear Usage**

A usage is analogous to an assembly of pipes and filters with pipes as featureless internal calls and filters as feature boxes. It has all the advantages of a pipe-and-filter architectural style, including that feature boxes are largely independent, can be composed freely in many different combinations, and the set of feature boxes is easily expanded.

A usage can be linear or non-linear. Figure 4.2 above shows an example of a linear usage. All the feature boxes in Figure 4.2 are free feature boxes which are freely interchangeable. When the DFC router must incorporate a free feature box into a usage, any idle box of the appropriate type will be chosen. In this usage, Spontaneous Message on Busy (SMB) is a feature box in the source zone. The SMB feature allows a caller to leave a message if the callee line is busy. It will deliver the stored caller's messages to the callee later. Wake-Up Call (WUC) service is a feature box in the dialled zone. The WUC feature allows subscribers to set up a wakeup call at a particular time. The Call Forwarding On No Answer (CFNA) feature and the CFB feature are feature boxes in the target zone. The CFNA feature allows the user to have any incoming calls sent to another number if the callee does not answer within X seconds or Y rings.

In this usage, the SMB feature box places an outgoing internal when it receives an incoming internal call. If the outgoing call receives a busy signal, the SMB feature box tears down the outgoing call and makes an announcement on an outgoing media channel to invite the caller to leave a message. It then monitors the incoming media channel for an acceptance signal. If the caller accepts the offer, the SMB feature box will record messages from the incoming media channel. The SMB feature box then tears down this call. Thereafter, the SMB feature box will awake periodically and attempt to deliver the stored messages by calling the original target. Eventually it should connect to the target and play the recorded messages, thus completing its task.

The WUC feature box is a free feature box in the dialled zone. When it receives an incoming internal call, the WUC feature box will collect the caller's address and desired wakeup-time and store them. Then it destroys the call and waits until the wake-up time is indicated on the real-time clock to which it has access. The WUC feature box is not connected to any other box during the waiting period, but it is occupied and is not available to service another wake-up call. At the right time, the WUC box will initiate a wake-up call to its subscriber.

As shown in Figure 4.2, the CFNA feature box receives an incoming internal call at one port, and then it first makes an outgoing internal call at another port. If the outgoing call is answered, or if the CFNA feature box is turned off, the CFNA feature box will go into a transparent state, in which the CFNA feature box connects the media channels at its two ports internally and makes itself unobservable to other boxes in the usage.

If the outcome of the outgoing call is that nobody picks up the phone and the CFNA feature box is activated, the CFNA feature box tears down the first outgoing call and makes another outgoing call to the forwarded address. The CFNA feature box then enters a transparent state.

The CFB feature box is similar to the CFNA feature box except that it is activated by a busy signal. If the outcome of the first outgoing internal call is busy, the CFB feature box will tear down the first outgoing call, place the second outgoing call and then go into a transparent state.

Of all the feature boxes shown in Figure 4.2, the SMB feature box and CFB feature box are the most interesting in that they are both busy treatment features. Their precedence impacts greatly on how features interact. If the CFB feature box is enabled, it will absorb and respond to a busy signal. Otherwise, it will forward the busy signal to another of the feature boxes discussed earlier. Therefore, only if the CFB feature box is not enabled, the SMB feature box can receive and respond to the busy signal. For busy treatment features, a higher priority feature box should be placed closer to the source of their trigger signal.

In addition to a linear usage, an example of a non-linear usage is shown in Figure 4.3. In this usage, the CW feature is implemented by a bound box, which is bound to a particular line interface. A bound feature is not interchangeable. When the DFC router must incorporate a bound feature in a usage, only the box of the appropriate type bound to that particular relevant address will be chosen. User C, who subscribes to a CW feature, makes a successful customer call to user A. This customer call goes through the CW feature box transparently. User B then tries to call user C when user C is talking with user A. The CW feature box will accept an internal call generated by user B's attempt on its third port, signal back to user B's LI and insert a call-waiting tone on the media path. Afterwards, it monitors the media path for a FLASH signal. Each time it recognizes a FLASH signal, it will switch its internal media path to connect user A and user C or connect user C and user B.

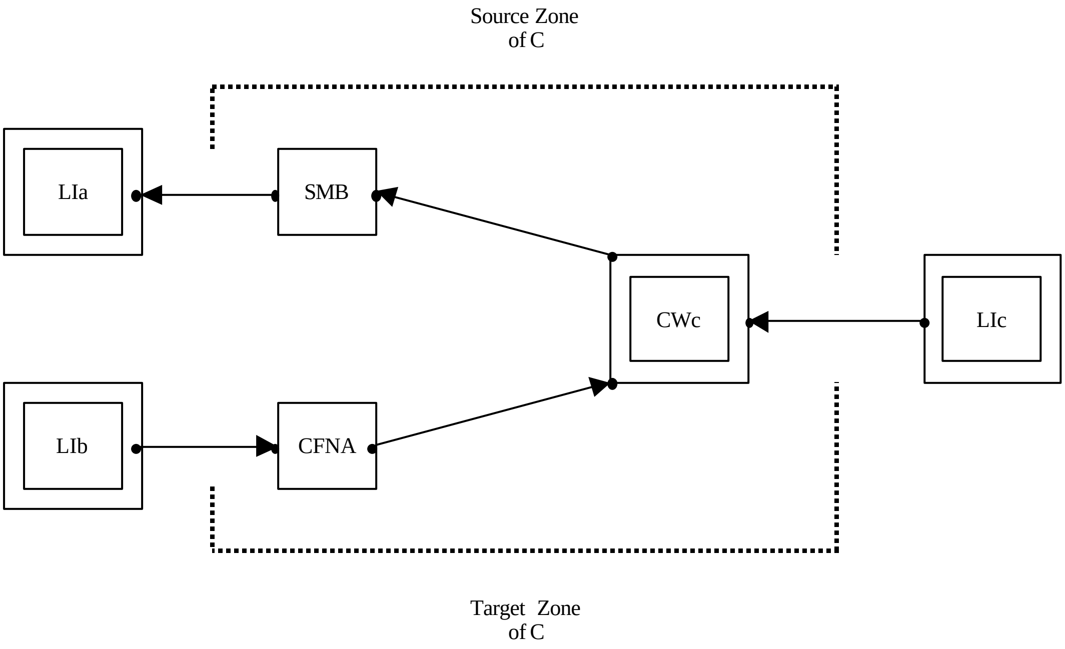


**Figure 4. 3 A Non-linear Usage**

Figure 4.3 also shows the difference between a usage and a customer call. In this non-linear usage, there are two customer calls, one made by user C and one made by user B. These two customer calls are joined together to construct a non-linear usage.

## 4.4 Routing

A DFC router performs a routing function using the DFC routing algorithm. The DFC router routes internal calls from one box to another based on fields of a SETUP message, which include a source field, a dialled field, a target field, a command field and a route field.

In the example shown in Figure 4.2, the LI of user A initiates a SETUP message, which includes a source field containing the address of user A, a dialled field containing the dialled string, a command field containing a NEW, an empty target field and an empty routing list. When the DFC router receives the SETUP message, it first assigns the target field from the dialled string. It then computes a new routing list for the usage based on the command NEW and inserts the routing list into the route field. The route field contains a box type and a zone type. In Figure 4.2, user A subscribes to the SMB feature, so the first pair of the route list is (SMB, source). The dialled string matches the triggering pattern of the WUC feature in the dialled zone so that the next pair on the route list is (WUC, dialled). User B subscribes to the CFNA feature and the CFB feature, thus the last pairs of the router list are (CFNA, target) and (CFB, target). The DFC router uses a feature precedence to control the order of the features in the route list.

The DFC router will find a box to route the internal call after it finishes modifying the SETUP message. It will remove the head pair (SMB, source) from the route list and route the internal call to an arbitrary box of that type, as the SMB feature box is a free feature box. Each feature box shown in Figure 4.2 does not change the routing list initially. Thus, each feature box will ensure that the command field is “CONTINUE” and copy all other fields of the SETUP message from the incoming call to the outgoing call. The SETUP message for the outgoing call is sent to the DFC router. The DFC router will not re-compute anything in the route list due to the “CONTINUE” command field, but will route the SETUP message to the box in the route list. In the last internal call of the usage, the route field is empty and the DFC router routes the internal call to the LI box of the target.

In the usage shown in Figure 4.2, the CFNA feature box will place the second outgoing call if no one answers its first outgoing call. In the second outgoing call, the CFNA box modifies the value of the target field in the SETUP message to the forwarded address, sets the command field to "UPDATE" and copies all other fields from the incoming call to the outgoing call. After the DFC router receives the SETUP message containing the UPDATE command, it will re-compute a new route list in a new target zone, remove the remnants of the target zone of user B, and insert the new route list in the SETUP message (shown in Figure 4.4). The behaviour of the CFB feature box is similar to that of the CFNA feature box, except that the CFB feature places its second outgoing call upon receiving a busy signal.

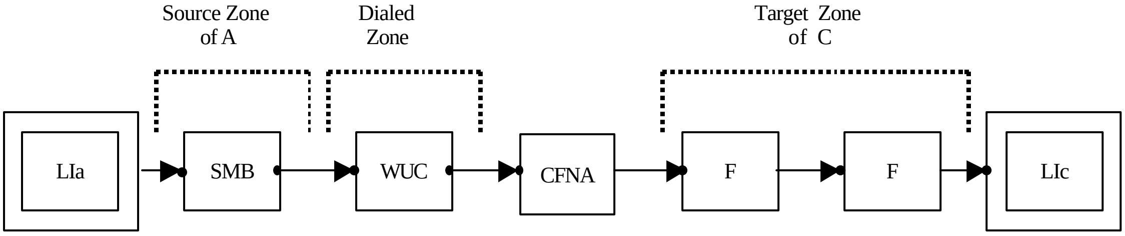


**Figure 4. 4 The Modified Route List in a Usage**

## 4.5 Signalling/Media Separation

The DFC architecture fully separates the paths carrying media from signalling paths. In the DFC architecture, a call can have any number (including zero) of bi-directional channels carrying media. Within a call, a media channel can be opened and closed any numbers of times.

Each channel has an identifier and two channel terminations at each port of a call. A DFC box port can be external or internal. Normal ports of the DFC architecture are internal ports. External ports refer to the place where a line, trunk or resource joins an interface box. Each interface box has at least one external port. Like ports, channels also can be external or internal. An external channel connects an external port and an end device, resource, or another network. An internal channel connects two normal box ports, each one in different boxes.

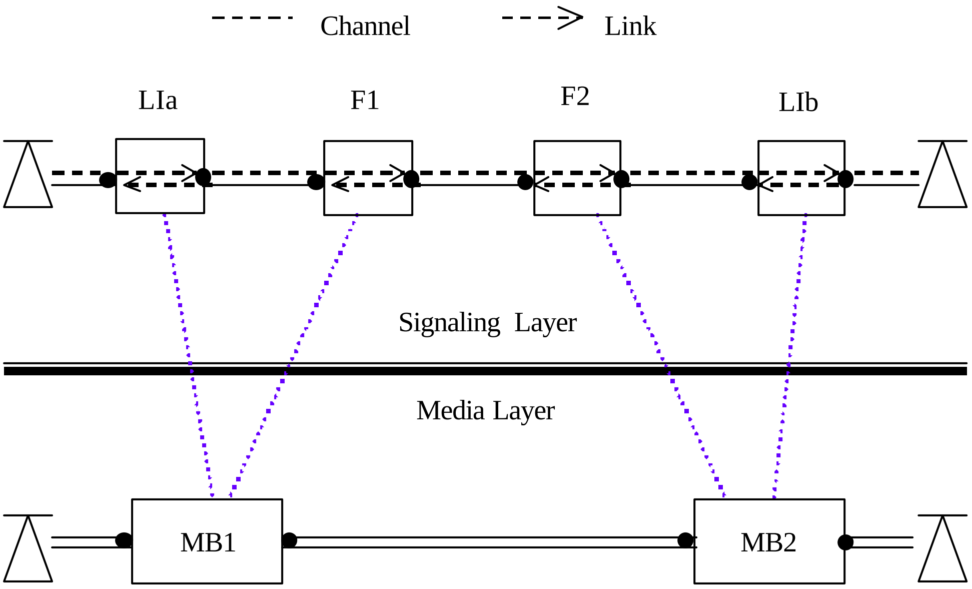


**Figure 4. 5 The Signalling Layer and the Media Layer in the DFC Architecture**

The DFC architecture also introduces links which refer to the media-processing states of a box. Each link is a unidirectional media connection between two channel terminations, which must have distinct ports on the same box [18]. Shown in Figure 4.5, a dash line in the signalling layer is a channel and a dashed arrow in the signalling layer is a link.

Shown in Figure 4.5, an Mbox in the medial layer is a software component, performing the function of a media switch. Mports refer to the ports of the media switch. An Mcall is a bi-directional media connection between Mports of different Mboxes. Each box in the signalling layer is given the address of one Mbox in that medium when it is created. It therefore knows the name of its assigned Mbox and it can communicate directly with the Mbox by sending commands. The dotted lines in Figure 4.5 show signalling connections between boxes in the signalling layer and their assigned Mboxes.

Each Mbox also includes a controller and a model. The controller receives commands from boxes in the signalling layer, maintains and uses the model, controls Mcalls, and responds to commands. A model is a data representation of the media processing states of all the boxes in the signalling layer, which this Mbox has been assigned to.

## 4.6 The ECLIPSE Project

The Extended Communications Layered on IP Synthesis Environment (ECLIPSE) currently implements the DFC service architecture on an IP platform. The main purpose of the ECLIPSE project is to determine whether the DFC architecture can be used as the service architecture of next-generation networks, which will be packet-switched networks.

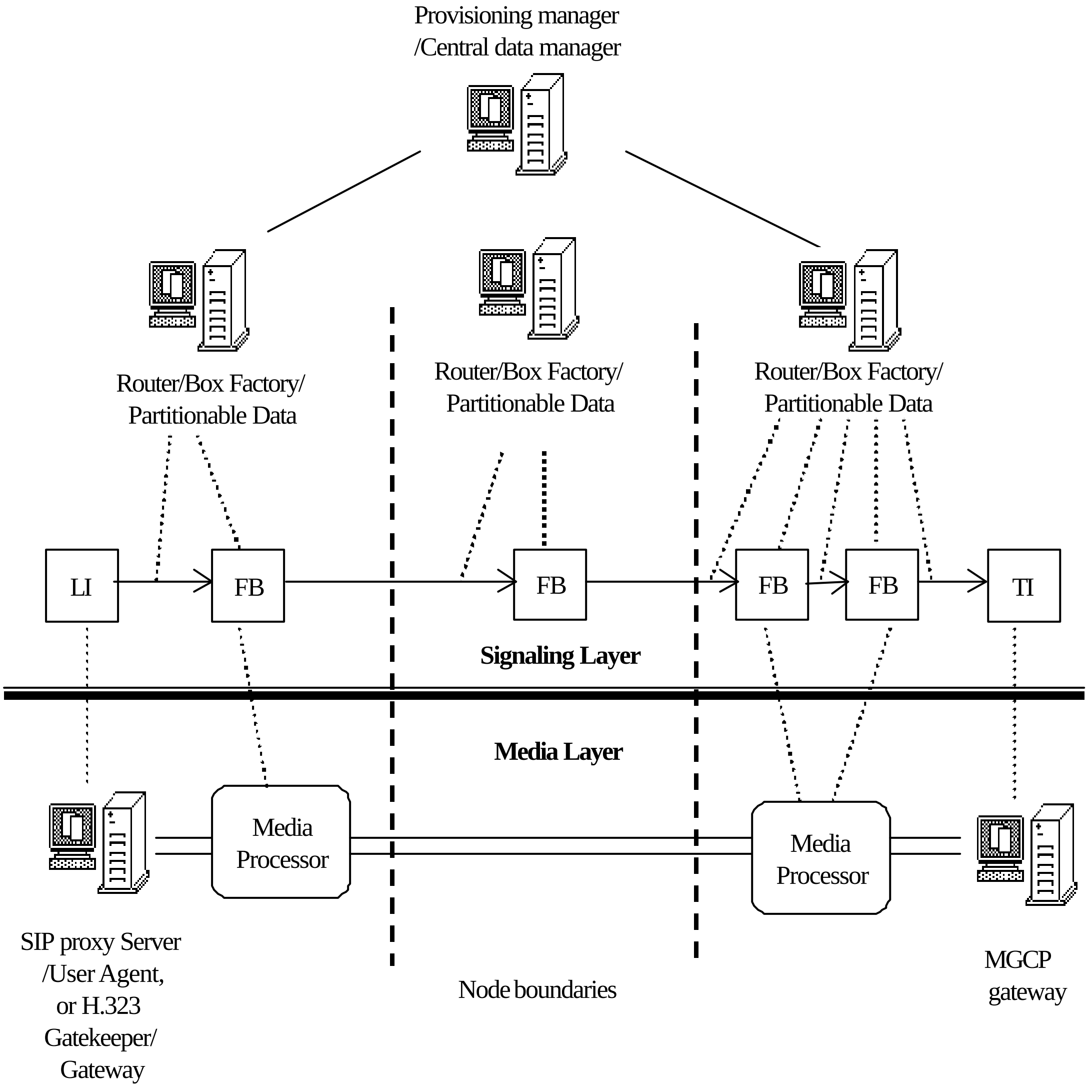


**Figure 4. 6 A DFC usage running on an ECLIPSE network**

In order to separate the signalling path from paths carrying media, the ECLIPSE project has introduced two major sub-systems: the signalling sub-system and the media sub-system. The signalling sub-system provides routing, signalling, control and global data while the media sub-system provides transmission and media processing below the signalling sub-system. Figure 4.6 shows a DFC usage running on an ECLIPSE network. Node boundaries indicate how software objects are distributed.

In the ECLIPSE project, a provisioning manager manages the operational data of features and configures router nodes. The DFC global data is organized as relations. The provisioning manager component distributes data to the DFC routers which need it. A centralized data manager currently handles the remaining relations.

Each ECLIPSE router can run independently to support a subset of customers. It not only computes and updates logical route lists of customers' calls, but also routes internal calls from one box to another required box. Local routers can work together to form a distributed implementation of DFC routers.

Within each router, there is an associated box factory which initiates new feature boxes as needed. Each feature box runs as an independent thread communicating with a Java implementation of port-to-port signalling through unbounded queues. The LI boxes and the TI boxes are persistent in the ECLIPSE implementation. All feature boxes can only access appropriate data partitions as determined by the ECLIPSE implementation.

In the ECLIPSE implementation, the network connects to its end users through an LI box or a TI box. End users can be devices that use DFC signalling natively. They can also be existing devices and application programs which implement new software to interface between DFC signalling and the protocols of existing endpoints. Some important interface boxes are implemented to handle H.323 and SIP. For a SIP endpoint, the SIP interface box behaves like a proxy server and a user agent.

## 4.7 A Basic Usage Set-up and Tear Down in the DFC Architecture

A basic usage refers to a usage connecting two end users who do not subscribe to any feature. A basic usage set-up and tear down is shown in Figure 4.7.

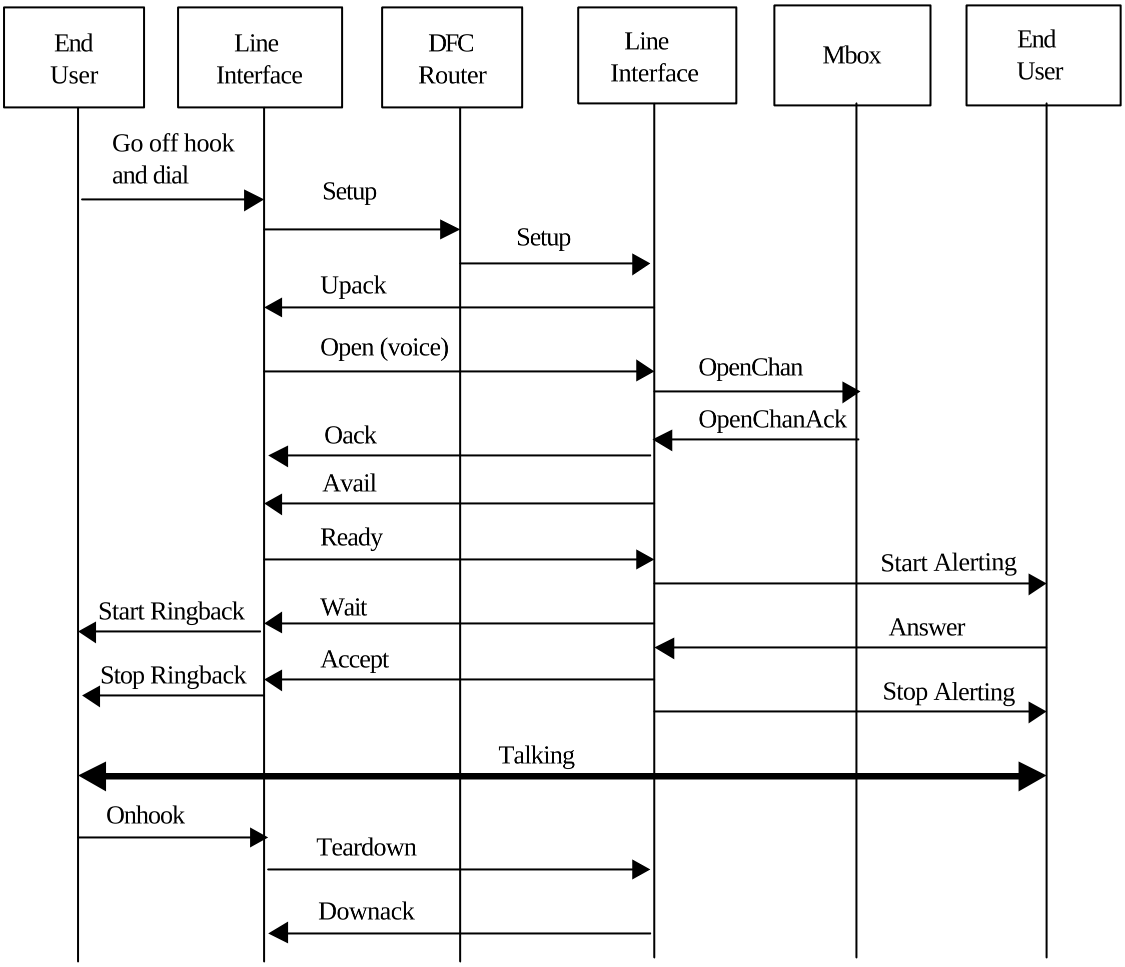


**Figure 4. 7 Sequence Diagram of A Basic Usage in the DFC Architecture**

When a caller wants to make a customer call, he will go offhook and enter a target address for the callee. The caller’ s LI box will generate a SETUP message, which is sent to a DFC router. The DFC router will compute the route list of the usage, insert the router list into the route field, and assign a callee address to the target field. Because there is an empty route list in the basic usage, the DFC router will forward the SETUP message to the callee’ s LI box. Upon receiving the SETUP message, the callee’ s LI box will send a UPACK message to the caller’ s LI box. Once the caller’ s LI box has received the UPACK message, it will send the callee’ s LI box an OPEN (voice) message, which is used to open a media channel. The callee’ s LI box will send its assigned Mbox an OPENCHAN command. The Mbox will response with an OPENCHANACK message. Then the callee’ s LI box will send an OACK message to the caller’ s LI box. If the callee’ s LI box can accept the incoming call, it will send an AVAIL message to the caller’ s LI box indicating that the usage has successfully reached an LI box and the LI box is ready for the usage. On receiving a READY message issued by the caller’ s LI box, the callee’ s LI box will alert the callee's end device. If the callee picks up his phone, the callee’ s LI box will send an ACCEPT message to the caller’ s LI. Thus, the caller and the callee set up a basic usage.

After the caller and the callee finish their call, either of them can generate a TEARDOWN message to destroy the usage. On receiving a TEARDOWN message, the LI box will send a DOWNACK message to the LI box which initiates it. The LI interface of the caller will send a command to its assigned Mbox to close the channel, which is not shown in Figure 4.7.

# Chapter 5 The Deployment of Features

In Internet telephony, as most end devices have the capacity to store and execute features, some of the features which are placed within the traditional telephone network - such as call waiting - can be deployed in such end devices. However, there are still some features that must be provided by the network.

In this chapter, some sample features are analysed, based on the SIP protocol and the DFC architecture respectively. The chapter also compares the SIP protocol with the DFC architecture. Since the current DFC implementation introduces much signalling overhead, it needs to be improved.

## 5.1 Overview

In traditional telephone networks, end devices can generate only a small set of signalling events and tones. They cannot receive or process sophisticated in-band signalling. These end devices are generally referred to as "dumb" end devices. When an end user presses a button or lifts a handset, a traditional phone will send a message to a gateway which then performs the corresponding function. A traditional phone does not know whether the button pressed is for entering calling card information, invoking services, dialling a subscriber number or answering prompts for an interactive voice response system. The phone does not know the state of a call. In addition, dumb end devices in traditional telephone networks have very limited storage. Therefore, in traditional telephone networks, most features have to be placed within the telephone network.

In contrast, most end devices in Internet telephony have the capacity to receive and process sophisticated signalling. They may have a notion of session states, such as a session connection state. After an end user makes or destroys a call, the intelligent end device will know whether the call is ringing, being answered, transferred, or terminated. This service awareness makes these end devices "intelligent". Intelligent end devices can usually execute programs that are placed on end devices or a remote server. Features that depend heavily on the state of an end device can be placed in these intelligent end devices.

Because of service-aware end devices, Internet telephony can move intelligence from the network to the end devices. This has prompted a great deal of discussion concerning the deployment of features [24, 30]. One view is to place all intelligence in end devices. This view is too narrow. Currently, end devices are usually not connected to the network permanently and may only have temporary IP addresses. They are likely to be powered off every so often. When end devices are turned off or do not connect to Internet, most features should still be available. For example, if an end user who subscribes to a VM feature turns off his end device, the voice mail system should continue to take callers' message. If an end user who subscribes to a CFNA feature is offline, his incoming calls should still be forwarded to another address. If the VM feature and the CFNA feature are placed in an end device and the end device is powered off, these features cannot be provided. Although it is reasonable to foresee that end devices may be permanently connected to the Internet in the future, some features which require the support of the network cannot be placed in end devices. Thus, placing all the features in end devices is not feasible in practice.

Another view is to place all the features inside the network. Here, “inside the network” means network nodes that handle the application-level control of routing signalling messages or are separated nodes containing feature programs and data with respect to the service offering. Unlike end devices, network nodes are permanently connected to the network. They usually have ample computational power and storage capacities. In the SIP protocol, network nodes can be network servers: proxy servers or redirect servers. In the DFC architecture, network nodes can be DFC routers. Especially, network nodes might be separated service nodes [25]. Separated service nodes isolate the services' programmability in computer nodes and make customer programmability more feasible. Furthermore, services can be deployed by third parties, rather than just by facilities-based carriers. However, placing all intelligence within the network ignores intelligent end devices. It uses more network resources, such as memory and storage, and it also usually requires payment for resource allocation. Some end users may prefer to place some features in their end devices based on some considerations, such as payment.

The third option is to place some of the features inside the network and some in intelligent end devices. There are several advantages to this. First, some features, such as 800 numbers and large-scale conferences, cannot be implemented in end devices because they require the support of the network. Second, there is the need to keep a state of a session inside the network while an end device is not connected or accessible. For example, such a state is necessary for call dispatching, such as the CFNA feature and the VM feature. Third, placing the features which depend heavily on the state of end devices in intelligent end devices is helpful for load balancing, since it does not place a burden on the network nodes. A fourth reason is that end users may prefer placing some features, such as the VM feature, in their intelligent end devices in order to reduce costs.

Moreover, features can be deployed dynamically in Internet telephony. A further option is to deploy features dynamically where they are needed at run time. This view provides a flexible way to deploy features within a distributed environment. Since a large number of features are enabled in Internet telephony, one end user may subscribe to many features. Storing and executing all subscribed features in the intelligent end device, such as a wireless device, are not feasible due to limitations of the end device. In this case, feature deployment can be determined dynamically at call time. In the DFC architecture, feature boxes can be deployed dynamically in intelligent end devices. After a usage, which is a dynamic assembly of boxes and internal calls, is constructed, feature boxes can be deployed dynamically. With this approach, upgrading features is much easier as service-implementation programs are usually stored in relatively few network nodes, rather than a larger number of endpoints.

Overall, in Internet telephony, powerful end devices offer flexibility in the deployment of features. Some features can be placed in intelligent end devices. Which features should be placed in intelligent end devices and which features can only be provided inside the network?

## 5.2 The Deployment of Network Features

Since network features require the support of the network, they must be provided inside the network. If they are implemented in end devices, they may not perform their functions very well or at all.

The 800 number service can be regarded as an example of a network feature. It allows for a commercial subscriber to pay for all incoming calls. As mentioned before, 800 numbers are virtual phone numbers and are used to point to real routable phone numbers. When a user dials an 800 number, the local exchange will try to find out which carrier handles the number and forwards the customer call to that carrier. The carrier examines a "toll free" record, maps the toll-free number to a real routable number and then responds with routing and billing information. The real phone number cannot be identified without the support of the network. Thus, the 800 number services must be provided within the network.

The GAP feature also can be regarded as a network feature. The network can use this feature to ensure that its servers and signalling network are not overloaded. The GAP feature cannot prevent congestion of the network if it resides in end devices. It is clear that the GAP feature needs the support of network and should be implemented within the network.

The third example of a network feature is Mass Calling (MAS). The MAS feature involves instantaneous and high-volume traffic that is routed to one or multiple destinations. Calls can be routed to these destination numbers based on various conditions, such as geographical location or time of day. This feature allows for the processing of huge numbers of incoming calls generated by broadcasts or games with the support of the network. Unlike the GAP feature that prevents overloading calls from going through, the MAS feature processes a large number of calls in "early" in a distributed manner for load balancing. The MAS feature cannot reside in an end device as it would not be able to process huge numbers of calls in the distributed manner. Therefore, this feature can only be provided within the network.

Network features usually take into account the location of users and the distribution of calls, and therefore need the support of the network. They can only be provided inside the network.

## 5.3 Comparing Client Features Residing in End Devices with Those Residing Inside the Network

Client features depend heavily on the state of end devices. They can be provided within the network or in intelligent end devices. Discussion on the deployment of client features is based on the DFC architecture and the SIP protocol respectively. It is assumed that end devices are intelligent, support the programming language(s) implementing the DFC architecture or the SIP protocol, and have enough resources to store and execute feature-implementation programs.

For simplification, SIP users are assumed to attach to the same proxy server and DFC users are assumed to attach to the same DFC router. If SIP users attach to different proxy servers or redirect servers, or DFC users attach to different DFC routers, a similar analysis can be used to discuss the deployment of client features. A similar analysis also can be used to discuss the deployment of client features for a pair of users such that the caller and the callee are in two different networks (i.e. a SIP user and a PSTN user).

In the DFC architecture, an LI box is an interface to an external line and translates between external protocols and internal DFC protocols. The LI interface boxes are assumed to reside in the end devices.

Client features can be provided inside the network or in end devices. Is it better to provide client features inside the network or in end devices? For discussion purposes, comparison criteria are presented as follows:

1. The number of signalling messages: Activating features usually requires some signalling messages. Since fewer signalling messages usually cost less network bandwidth and reduce signal-transmission delay, the smaller number of messages is more desirable. If deploying client features in the end devices means fewer messages than deploying them inside the network, deploying them in the end devices is better. Otherwise, deploying them within the network is more desirable.

2. The cost to the resources of network nodes: Executing features in network nodes (such as DFC routers, SIP network servers or separated service nodes) costs their CPUs and memories, and increases their load. Storing features in network nodes also

costs storage. A network node usually has ample computational power and storage capacity, but it is shared by a number of end users. Therefore, cost to network nodes' resources, such as CPUs, memories and storage, should be reduced if possible. Lower cost to the resources network nodes is more desirable.

In some cases, there is a trade-off between the number of signalling messages and the cost to the resources of the network nodes. Features that are placed inside the network or in end devices may have fewer messages but may use up more of the network nodes’ resources. In this situation, using fewer messages is more desirable in that reducing transmission delay usually plays a more important role in Internet telephony.

Based on these assumptions and comparison criteria, the deployment of client features is discussed in the following sections. The discussion and conclusions on the deployment mainly concern with pure signalling messages. Other issues regarding the deployment of features will be described in Chapter Seven.

## 5.4 The Deployment of the AD Feature, the LNR Feature, the OCS Feature and the PSC Feature

Auto Dial (AD) allows a subscriber to dial the telephone number (address) that he/she calls regularly with the push of one button. Last Number Redial (LNR) allows a subscriber to redial the most recent target address without entering the dialled telephone number (address). Personal Speed Calling (PSC) allows a subscriber to store some telephone numbers (addresses). These speed call numbers are programmed and used by individuals at their own telephones.

The AD feature, the LNR feature and the PSC feature help callers produce target addresses if they attempt to make a customer call. The OCS feature blocks calls at the originating party (the caller) based on the address to which a customer call is placed. Since none of these features – AD, LNR, OCS and PSC – require the support of network nodes, they can be placed in intelligent end devices. Furthermore, the AD feature, the LNR feature and the PSC feature, which only store the callee's addresses, can be provided even in some dumb end devices. In Internet telephony, these features can also be provided inside the network.

### 5.4.1 The Deployment of the AD Feature, the LNR Feature, the OCS Feature and the PSC Feature in the DFC Architecture

The OCS feature will check its screening list when its subscriber makes a customer call. The LNR feature, the AD feature and the PSC feature are simple features that store target addresses. If their subscribers make a customer call, the LI box will place a DFC internal call. Since there has been no target address in this internal call, the target field in the SETUP message is null. Only source information is present so that only the features in the source zone appear in the route field.

In the DFC architecture, one approach is that these features are implemented as free feature boxes, which are executed inside the network, in the source zone. If a user, who subscribes to an AD feature, an LNR feature, or a PSC feature, makes a customer call, the DFC router will route a SETUP message to an AD feature box, an LNR feature box, or a PSC feature box. In order to obtain a target address for the usage, these feature boxes have to retrieve their operational data. Once they get a target address, a null entry in the target field of the SETUP message is replaced by the new target address. The DFC router will compute the FBs in the target zone of the usage.

If a user, who subscribes to an OCS feature, makes a customer call, a SETUP message will be routed to an OCS feature box. The OCS feature box will check whether the target address of this usage is on the screening list of its operational data. If yes, this usage will be terminated. Otherwise, the usage will go through the OCS feature box transparently.

Figure 5.1 shows the sequence diagram for setting up a usage, which contains an OCS feature box, an AD feature box, a PSC feature box or an LNR feature box. After these feature boxes receive an OPEN message, they will send an OPENCHAN command and an OPEN2LINK command to their assigned Mboxes. The OPEN2LINK command is used to open two links between the two channel terminations, one in each direction. The callee' s LI interface will send an OPENCHAN command to its assigned Mbox if it receives an OPEN message. The Mboxes will respond with an OPENCHANACK message or an OPEN2LINKACK message. (The DFC architecture does not impose an ordering constraint on some messages, such as the order of an OACK message and an AVAIL message [32].) If the callee picks up his phone, its LI interface will issue an ACCEPT message. Once the LI interface of the callee has received the ACCEPT message, the caller and the callee can talk with each other.

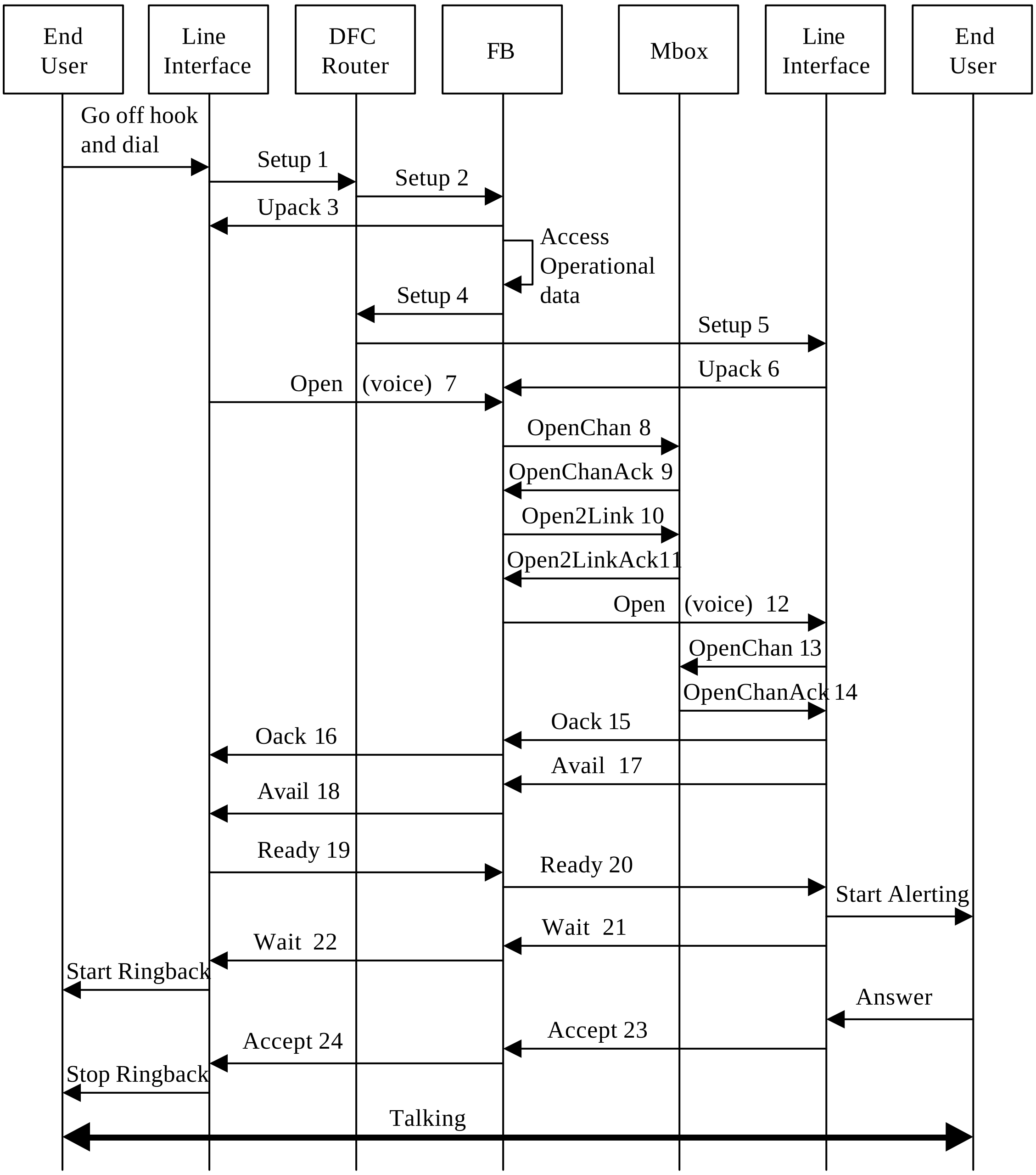


**Figure 5. 1 Sequence Diagram for Setting Up a Usage Containing one Feature Box In the DFC Architecture**

Figure 5.2 shows the sequence diagram for tearing down a usage containing one feature box. In the teardown phase, an LNR feature box, an OCS feature box, an AD or a PSC feature box will send a CLOSE2LINK message and a CLOSECHAN message to its assigned Mbox. The callee’ s LI interface will send a CLOSECHAN message to its assigned Mbox. The Mbox will respond with a CLOSE2LINKACK message or a

CLOSECHANACK message. The feature box will then send a DESTROYED message to the DFC router. Thirty-five messages are needed to set up and tear down a usage that contains one feature box. If a usage contains more feature boxes, more messages will be required.

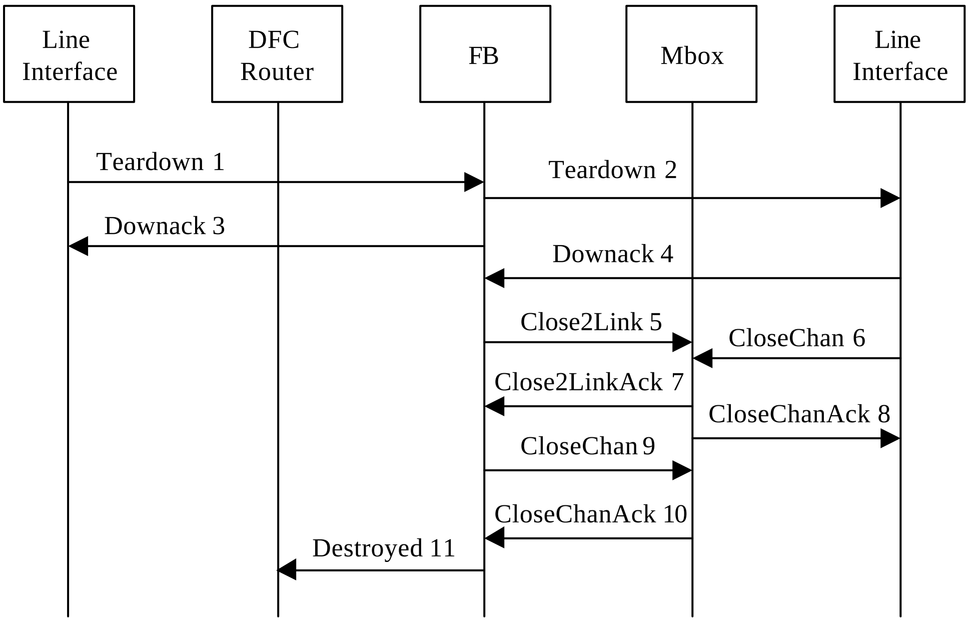


**Figure 5. 2 The Sequence Diagram For Tearing Down a Usage Containing one Feature Box in the DFC Architecture**

However, in the current DFC implementation, some messages, such as the second message (the SETUP message) in Figure 5.1, are sent and received in the local host if FBs are executed in the DFC router. In fact, if a feature box is executed in the DFC router, if the LI interfaces reside in the end devices, and if Mboxes reside in the DFC router (placing Mboxes in other network nodes introduces more signalling overhead), twenty-four messages will be transferred over the network to set up and tear down a usage containing this feature box (For the OCS feature box, the target address is not in the screening list.).

In the DFC architecture, another approach implements the LNR, AD, PSC and

OCS features as free feature boxes running in intelligent end devices. The location of a feature box will not affect the sequence diagram of the usage containing this feature box. Thirty-five messages are also required, in this approach. However, only twenty-six messages are transferred over the network if these feature boxes are executed in the end devices and the LI interfaces reside in the end devices (for the OCS feature box, the target address is not in the screening list.). Compared with the second approach, the first requires fewer messages over the network. (For the OCS feature box, the number of messages that are transmitted over the network will be affected by the location of the OCS feature box if the target address is in the screening list.)

In addition, some feature boxes have to write, modify or read their operational data. Since operational data are stored inside the network in the current ECLIPSE implementation, running these feature boxes in the end devices introduces communication overhead.

If the DFC implementation allows the option to store operational data in end devices, the feature boxes that are executed in the end devices can read, write or modify their operational data locally. The third approach allows operational data to be placed in the end devices and feature boxes to be executed in the end devices. This approach reduces overhead, compared with the second approach discussed above.

In the current ECLIPSE implementation, each feature box has to send commands to its assigned Mbox although it does not involve media controls. Mboxes should be available when the end devices are offline. Therefore, it is not feasible to place Mboxes in the end devices. Even if the LNR feature box, the AD feature box, the OCS feature box and the PSC feature box do not concern the media paths, the commands will have to be sent to the remote Mboxes, which introduces a great deal of overhead.

The fourth approach introduces a new kind of feature box: a simple feature box (SFB), which only deals with the features concerning the signalling layer. These feature boxes will not send any command to the Mboxes. If the OCS, LNR, PSC and AD features are implemented as SFBs, placing them in the end devices offers the benefits of modularity and reduces costs to the DFC routers, but does not require more signalling messages that are transmitted over the network. In this case, it is desirable to place them in the end devices.

The fifth approach is to integrate these features fully with LI interface boxes. For the LNR feature, the AD feature and the PSC feature, LI interface boxes can retrieve a target address stored in their local end devices easily, without the participation of the network. For the OCS feature, the LI boxes can determine whether the target address is on the screening list of the OCS feature, which is stored in the end devices. There is no the need for additional signalling messages to provide services with this approach. The disadvantage is that this approach loses the benefits of modularity.

In all the approaches discussed above, the fifth – in which the AD feature, the LNR feature, the OCS feature and the PSC feature are fully integrated with the LI interface boxes – is the least complicated with regard to providing services, but loses the modularity.

All of the other approaches have in common the fact that these features are implemented as modularized feature boxes. In the DFC architecture, implementing these features as feature boxes requires additional signalling messages–communication overhead is therefore introduced. In the ECLIPSE implementation, placing these FBs inside the network will save signalling messages over the network. Thus, it is better to deploy the LNR feature box, the AD feature box, the OCS feature box and the PSC feature box inside the network. As far as the DFC architecture is concerned, deploying feature boxes inside the network means feature boxes are executed inside the network.

However, it is not efficient to place the LNR feature box, the AD feature box, the OCS feature box and the PSC feature box inside the network. It seems that the current DFC implementation needs to be improved. This will be discussed in section 5.11.

If the OCS feature, the LNR feature, the PSC feature and the AD feature are implemented as SFBs, and if the operational data can be stored in the end devices, it will be better to place these feature boxes in the end devices.

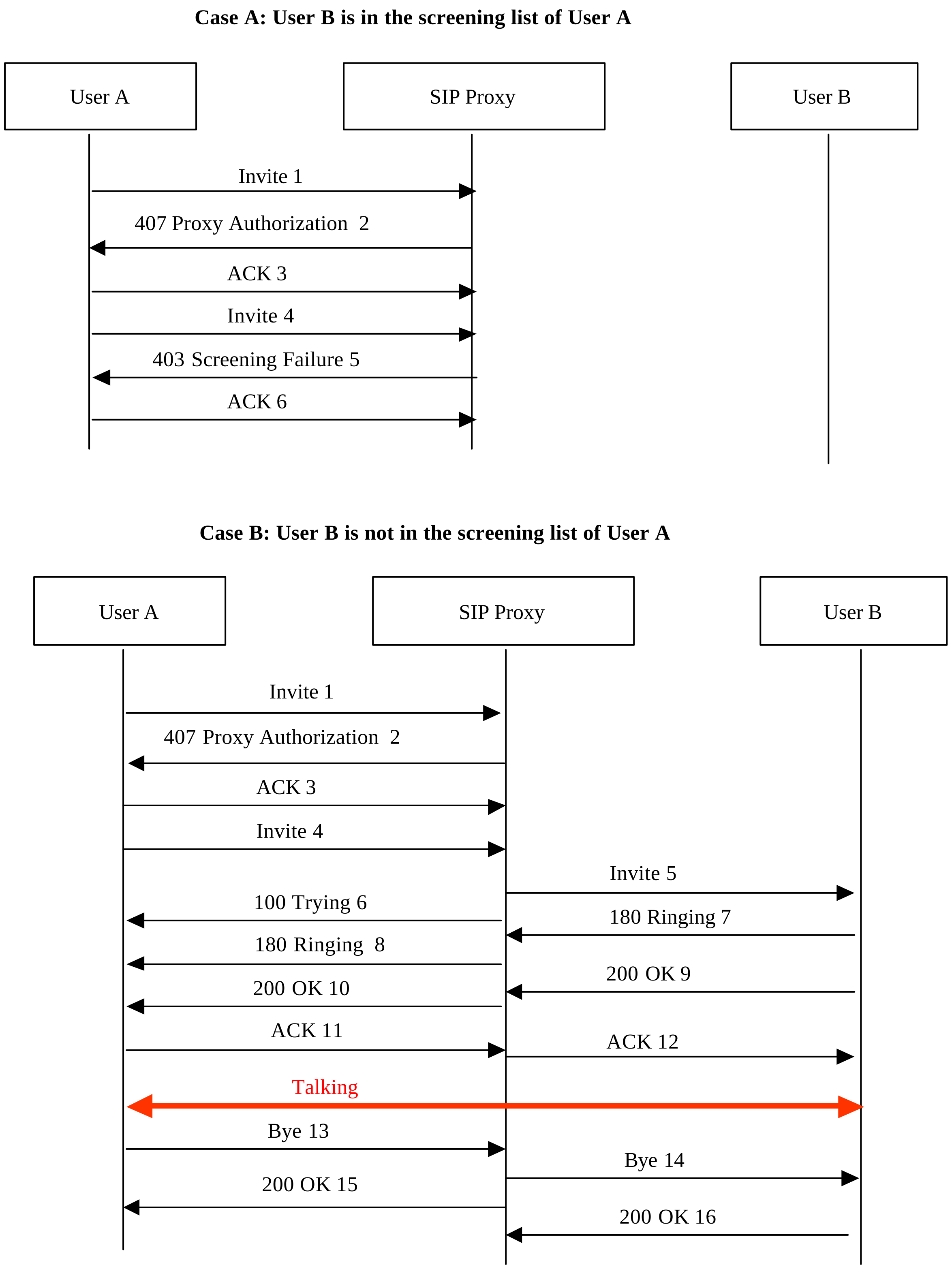
### 5.4.2 The Deployment of the AD Feature, the LNR Feature, the OCS Feature and the PSC Feature in the SIP protocol

In the SIP protocol, the LNR feature, the AD feature and the PSC feature can be placed in the end devices. Since the features that are placed in the end devices can recognize a target address, an INVITE message will include the target address in "TO" header field. Thus, if end users only subscribe to one of the features placed in the end devices, their calls will be basic calls, as shown in Figure 3.6 (page 26).

If the LNR feature, the AD feature and the PSC feature are placed inside the network, the proxy server has to retrieve the target address after receiving the INVITE message from the caller. This increases load on the proxy server and costs more of the proxy server' s resources. Since the proxy server usually processes the requests from a large number of clients, load on the proxy servers should be reduced if possible. Placing these features in the proxy server cannot save messages. Thus, in the SIP protocol, it is more desirable to place the LNR feature, the AD feature and the PSC feature in the end devices.

Should the OCS feature reside in the end devices, events related to the OCS feature, such as checking its screening list, can be completed in the end devices without sending an INVITE request to a proxy server. If the callee is in the screening list, the customer call will be terminated. Otherwise, the INVITE message will be forwarded to the callee. This customer call will be a basic call (Figure 3.6, page 26).

If the OCS feature resides inside the network, a proxy server will first require a subscriber to identity himself. Some messages related to authorizations [20], such as "407 Proxy Authorization", will be transmitted between the subscriber and the proxy server. (If the OCS feature resides in the end devices, authorizations for end users are optional.) The call flow of the OCS feature placed inside the network is shown in Figure 5.3. If the callee is in the screening list, the customer call will be terminated by the proxy sever. Otherwise, the proxy server will forward the incoming INVITE message to the callee.


Case A: User B is in the screening list of User A

Case B: User B is not in the screening list of User A

**Figure 5. 3 The Call Flow of the OCS Feature Residing inside the SIP-enabled Network**

If the OCS feature is placed in a separated service node, there is a need for additional messages between the proxy server and the service node to carry the information required to activate and execute features. These additional messages are not shown in Figure 5.3.

Table 5.1 shows the minimal number of messages related to locations of the OCS feature. Placing the OCS feature in the end devices not only saves messages, but also reduces the cost to the proxy server. Therefore, in the SIP protocol, it is better to place the OCS feature in the end devices.

**Table 5. 1 The Minimal Number of Messages Related to Locations of the OCS Feature in the SIP protocol**

| The Callee<br>The OCS Location | Not in the screening list | In the screening list |
|---|---|---|
| In an end device | 13 | 0 |
| Inside the network | 16 | 6 |

## 5.5 The Deployment of Call Forwarding

There are several features related to call forwarding. For example, the CFU feature, the CFB feature, the CFNA feature, Call Forwarding No Device (CFND), Selective Call Forwarding (SCF), Selective Call Forwarding on Busy (SCFB) and Selective Call Forwarding on No Answer (SCFNA) are all features related to call forwarding. The SCF feature allows its subscriber to forward particular pre-selected calls to another address no matter what the status of the called-party line is. The SCFB feature allows its subscriber to have particular pre-selected calls forwarded to another address if the called party is busy. The SCFNA feature allows its subscriber to forward certain pre-selected calls to another address if the callee does not answer within X seconds or Y rings.

The CFND feature redirects the incoming customer call to another address if no device is registered for the destination address given. Since it clearly needs the support of the network, it can be regarded as a network feature and must be provided inside the network.

The CFU feature, the CFNA feature, the SCF feature, and the SCFNA feature should still be available inside the network when end users are offline.

### 5.5.1 The Deployment of Call Forwarding in the DFC Architecture

In the DFC architecture, features related to call forwarding can be implemented as free FBs in target zones. Although they (except the CFND feature) can be integrated with the LI interfaces, the benefits of modular feature boxes are lost due to the integration. Therefore the discussion below only concerns their implementation as FBs.

**(1) The Deployment of the CFU Feature and the SCF Feature**

Figure 5.4 shows a CFU feature box in the DFC architecture. User A tries to make a customer call to user B, who subscribes to the CFU feature. The DFC router routes a

SETUP message to the CFU feature box (F4 in Figure 5.4) in the target zone of user B. The CFU feature box changes the target address in the SETUP message before user B is reached. This change requires reselection of the target user' features and reconstruction of the unused part in the routing list. The CFU feature box sets the "command" field to UPDATE. UPDATE causes the DFC router to remove the remnants in user B' s target zone from the route list and replace them with a newly computed target-zone route list for user C (shown in the lower part of Figure 5.4). The DFC router routes the updated SETUP message to the feature boxes in the target zone of the new address.

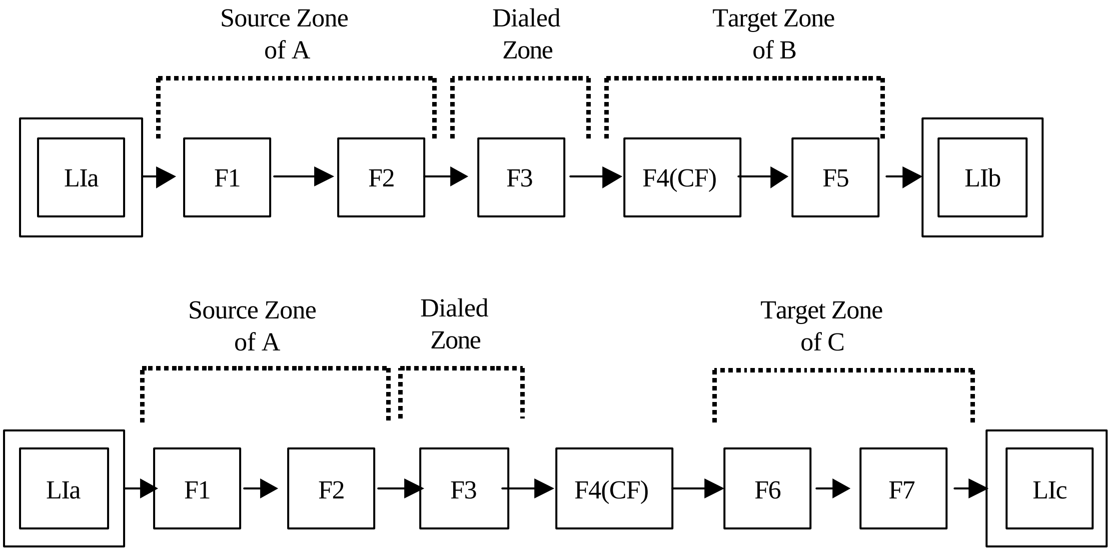


**Figure 5. 4 The CF Feature in the DFC Architecture**

The difference between the CFU feature and the SCF feature is that the SCF feature only forwards the calls from pre-selected callers to a new address. The pre-selected list is stored as operational data of the SCF feature box. Thus, the SCF feature box will have to access its operational data to check whether the caller is on its pre-selected list or not after it receives a SETUP message. If yes, the usage will pass through the SCF feature box transparently. Otherwise, the SCF feature box will change the target address before the original target is reached.

In the DFC architecture, the SCF feature box and the CFU feature box will still be in the chain of boxes during a usage although this usage does not reach the original target user who subscribes to the SCF feature or the CFU feature (shown in Figure 5.4). If the SCF feature box or the CFU feature box reside in an end device and the end device is turned off or not connected to the DFC system during a usage, this usage may be affected because it may need messages going through the SCF feature box or the CFU feature box. Thus, the SCF feature box and the CFU feature box have to be provided inside the network.

### (2) The Deployment of the CFB Feature, the CFNA Feature, the SCFB Feature and the SCFNA Feature

In Figure 5.4, user A tries to make a customer call to user B, who subscribes to the CFNA (F4) feature. The CFNA feature box appears in the target zone of user B. It monitors the messages coming back to it on its original outgoing call directed to user B after it places its first outgoing internal call. If the altering message arrived at the CFNA feature box at least X seconds ago, and if there has been no subsequent signal [8], the CFNA feature box will tear down this first outgoing internal call and place a second outgoing internal call whose target address is the forwarded address C. This is shown in the lower portion of Figure 5.4.

The SCFNA feature is similar to the CFNA feature except that the SCFNA feature only forwards calls from pre-selected callers, which is operational data of the SCFNA feature box. The behaviour of the CFB feature box is also similar to that of the CFNA feature box. The difference between them is that the CFB feature box responds to a busy signal while the CFNA feature box responds to a no-answer condition. The SCFB feature and the CFB feature are also similar except that the SCFB feature only forwards calls from pre-selected callers to a new address.

Like the SCF feature box and the CFU feature box in the DFC architecture, the CFNA feature box, the CFB feature box, the SCFNA feature box or the SCFB feature box will appear in the chain of boxes during a usage although the original target user who subscribes to these features is not reached (shown in the lower portion of Figure 5.4). If the CFNA feature box, the CFB feature box, the SCFNA feature box or the SCFB feature box resides in the end devices, and these are turned off or not connected to the DFC system during the usage, this usage will be affected. Therefore, the CFNA feature box, the CFB feature box, the SCFNA feature box and the SCFB feature box have to be provided inside the network.

## 5.5.2 The Deployment of Call Forwarding in the SIP protocol

The following section will examine the deployment of features related to call forwarding in the SIP protocol.

### (1) The Deployment of the CFU Feature and the SCF Feature

In Figure 5.5, user A attempts to make a customer call to user B, who subscribes to the CFU feature. User A's UAC sends an INVITE request message to a SIP proxy server. If the CFU feature resides inside the SIP proxy server, the INVITE message will be modified in the proxy server and the proxy server will forward the modified INVITE message to the new target address [20]. As shown in Figure 5.5, the number of total messages to set up a customer call is nine, if user B subscribes to the CFU feature that is placed in the SIP proxy server.

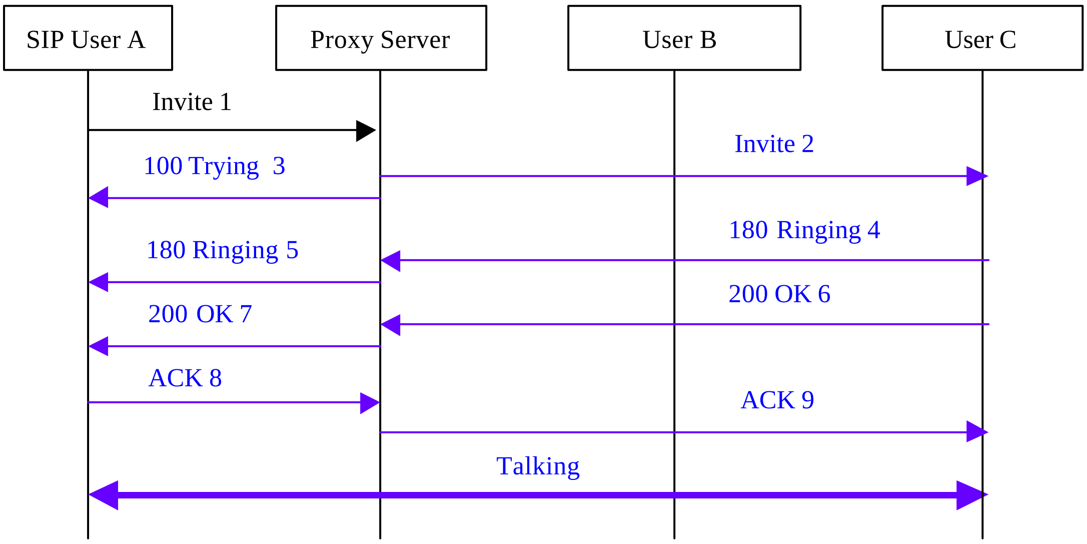


**Figure 5. 5 The Call Flow of the CFU Feature Residing in the SIP Proxy Server**

If the CFU feature is placed in a separate service node, two additional messages are required. One message is a service request, which the proxy server sends to the service node upon receiving the INVITE message. After the CFU feature is executed, the service node will send the proxy server a message that contains the new target address. In this situation, eleven messages are required to establish a customer call.

Should the CFU feature be placed in an end device (Figure 5.6), upon receiving the INVITE message the SIP server will forward it to the UAS of user B. Since user B subscribes to the CFU feature, user B responds with a "302 Moved" message, containing the new target address. The SIP server will forward the INVITE message to the new target address. As shown in Figure 5.6, the number of messages to set up a call will be twelve if the CFU feature resides in an end device.

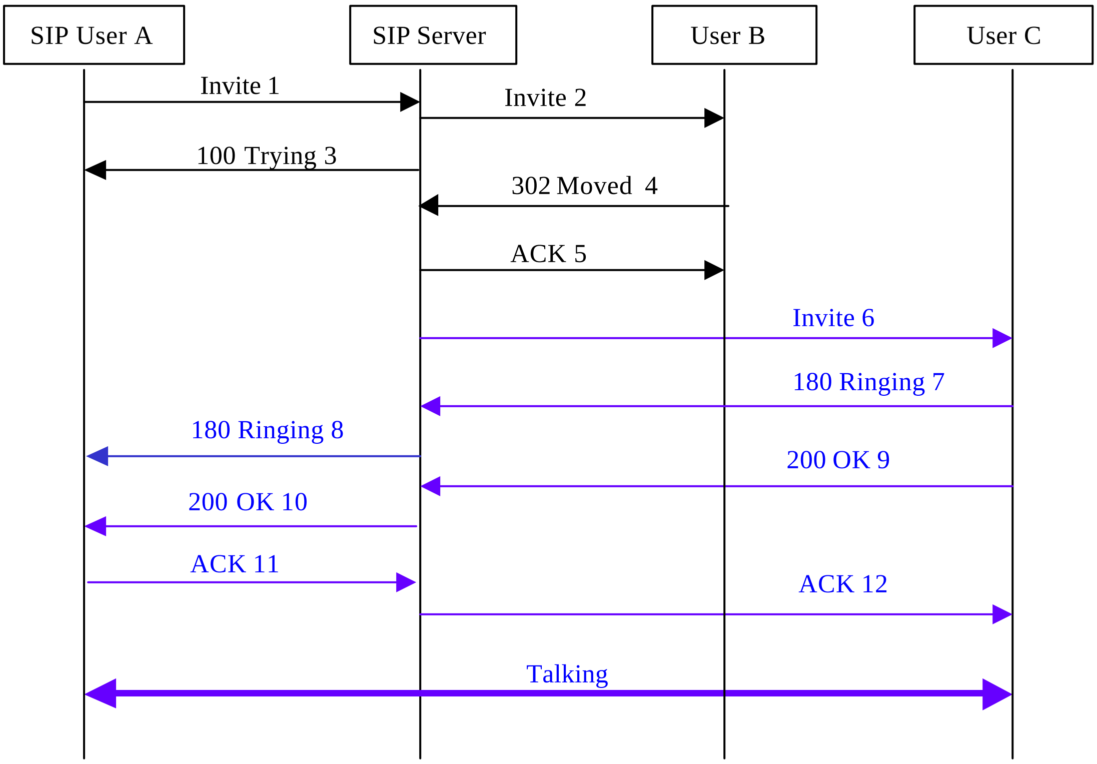


**Figure 5. 6 The Call Flow of the CFU Feature Residing in an End Device**

It is clear that placing the CFU feature inside the network saves the number of messages. Therefore, it is better to place the CFU feature inside the network.

The SCF feature is similar to the CFU feature except that the SCF feature only forwards calls from pre-selected callers to a new address. If the caller is not in the selected list, the call will be directed to the original callee. This call will be a basic call if neither user B or user A subscribes to any other feature. Otherwise, the call will be redirected to the new address. The call flow of the SCF feature placed in an end device or in the proxy server is shown in Figure 5.5 and Figure 5.6 respectively if the caller is in the selected list.

Like the CFU feature, placing the SCF feature inside the network has fewer messages if the caller is on the selected list. Should the caller not be in the list, placing the SCF feature inside the network will not need more signalling messages. Therefore, it is better to place the SCF feature inside the network. In addition, the CFU feature and the SCF feature placed in the network will still be available even if their subscribers are offline. However, placing them inside the network does cost more network node resources.

### (2) The Deployment of the CFB Feature, the CFNA Feature, the SCFB Feature and the SCFNA Feature

If user A attempts to make a customer call to user B, who subscribes to the CFB, CFNA, SCFB or the SCFNA feature, if user B is idle and answers the call, and if neither user A nor user B subscribes to any other feature, the customer call will be a basic customer call.

If user B subscribes to the CFB feature and is busy, the customer call will be forwarded to another address (user C). If the CFB feature resides in end devices and user B is busy, user B will send the proxy server a "302 Moved" response that contains a new target address. The proxy server will forward the INVITE message to the new address. Figure 5.6 also shows the call flow of the CFB feature placed in the end devices.

Since in Internet telephony only an end user knows exactly whether he is busy or not, the proxy server is aware that end users are busy only upon receiving a busy signal issued by an end user. If the CFB feature resides in the proxy server and user B is busy, the UAS of user B will send the proxy server a busy signal. After receiving the busy signal, the proxy server will look up its busy treatments. Then the INVITE message will be modified by the CFB feature. The proxy server will forward the modified INVITE message to the new address. The call flow shown in Figure 5.6 will be that of the CFNA feature placed in the proxy server if the MOVED message (the fourth message) is replaced by the BUSY message. If the CFB feature is placed in a separated service node, more messages are needed between the service node and the proxy server.

Compared with placing the CFB feature inside the network, placing the CFB feature in an end device reduces the cost to network node resources, but does not need more signalling messages. Thus, in the SIP protocol, it is better to deploy the CFB feature in the end devices.

The SCFB feature also forwards the incoming customer if the called party is busy, but it will check whether the caller is on its selected list or not. Like the CFB feature, in the SIP protocol, placing the SCFB feature in the end devices is better in that it reduces the cost to network node resources and does not increase the number of messages.

Unlike the CFB feature, the CFNA feature will redirect the incoming customer call if the called party does not answer it in X seconds. The call flow of the CFNA feature that is placed inside the network is similar to Figure 5.6. The call flow of the CFNA feature that is placed inside the network is shown in Figure 5.7. This call flow does not contain the messages between the service node and the proxy server if the CFNA is placed in the separated service node. Placing the CFNA feature in the end device not only saves signalling messages, but also reduces the cost to network node resources.

Therefore, in the SIP protocol, it is better to place the CFNA feature in the end device.

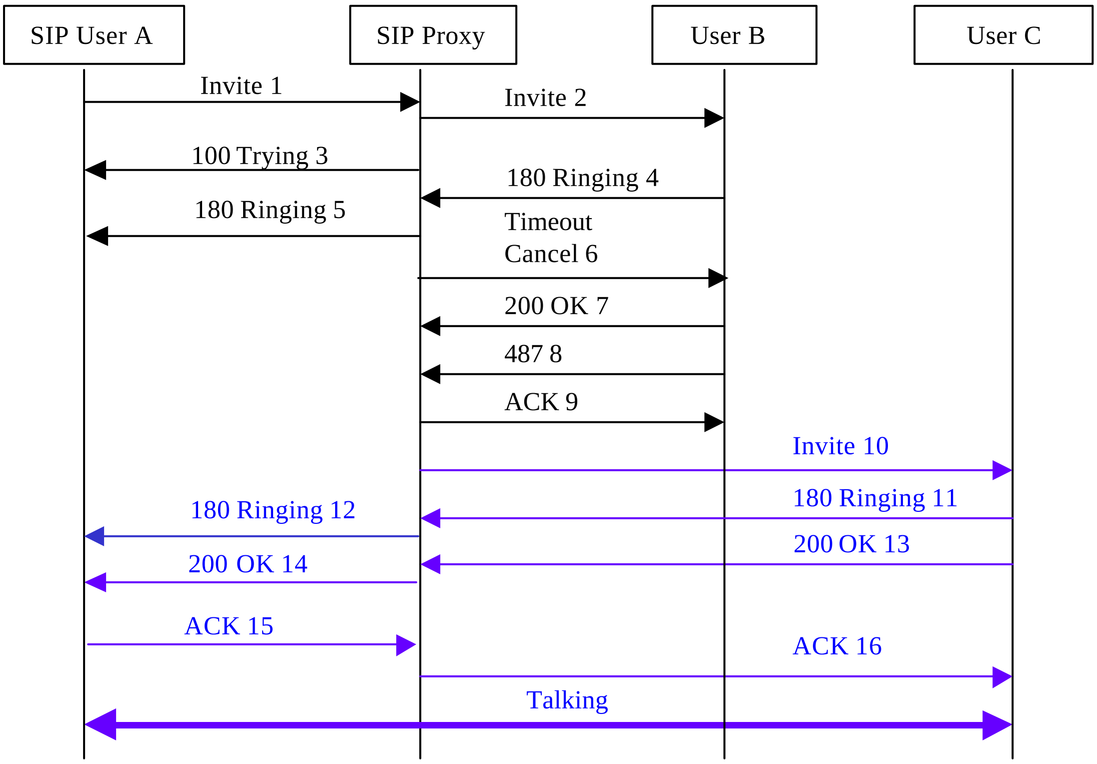


**Figure 5. 7 The Call Flow of the CFNA Feature Residing inside the SIP-enabled Network**

The SCFNA feature is similar to the CFNA feature. The difference is that the SCFNA feature only forwards the incoming customer calls from callers in its selective list. It is also better to place the SCFNA feature in the end devices due to fewer signalling messages and reduced cost to network node resources.

The above discussion shows that the deployment of the CFNA feature, the SCFB feature, the SCFNA feature and the CFB feature is similar. Placing them in the end devices reduces the cost to network node resources, such as CPUs and memories, but does not require more messages. Thus, in the SIP protocol, it is better to place them in the end devices.

## 5.6 The Deployment of the CW Feature

The CW feature is one of the most important features of traditional telephone networks. It sends its subscriber a notification that another party is trying to reach his number while he is busy. The CW feature depends heavily on the state of the end device. It can be deployed in intelligent end devices. Since the network is aware of the state of the end device when it receives a busy signal which is sent by the end user, it can also be deployed inside the network.

### 5.6.1 The Deployment of the CW Feature in the DFC Architecture

In the DFC architecture, the CW feature is implemented as a bound feature (Figure 4.3, page 35) that is bound to a particular line interface. User C, who subscribes to the CW feature, places a customer call to user A. This usage goes through the CW feature box transparently. The sequence diagram in Figure 5.1 also shows the set-up for the first customer call which contains a CW feature box if the FB does not access to the operational data. Twenty-four messages are required to set up the first customer call.

User B then tries to call user C while user C is talking with user A. Upon receiving the second SETUP message, the CW feature box accepts the second internal call, and signals back to user B that the customer call has reached a state of alert. The CW feature box inserts a call waiting tone on the media path and then monitors this media path for a in-band FLASH signal. If the CW feature box catches a FLASH signal, it will send its assigned Mbox a CLOSE2LINK message, which closes its internal link connecting user A with user C, and an OPEN2LINK message that opens its internal link connecting user B with user C. Figure 5.8 shows the sequence diagram for the set up of the second customer call using the CW feature box, which requires nineteen messages.

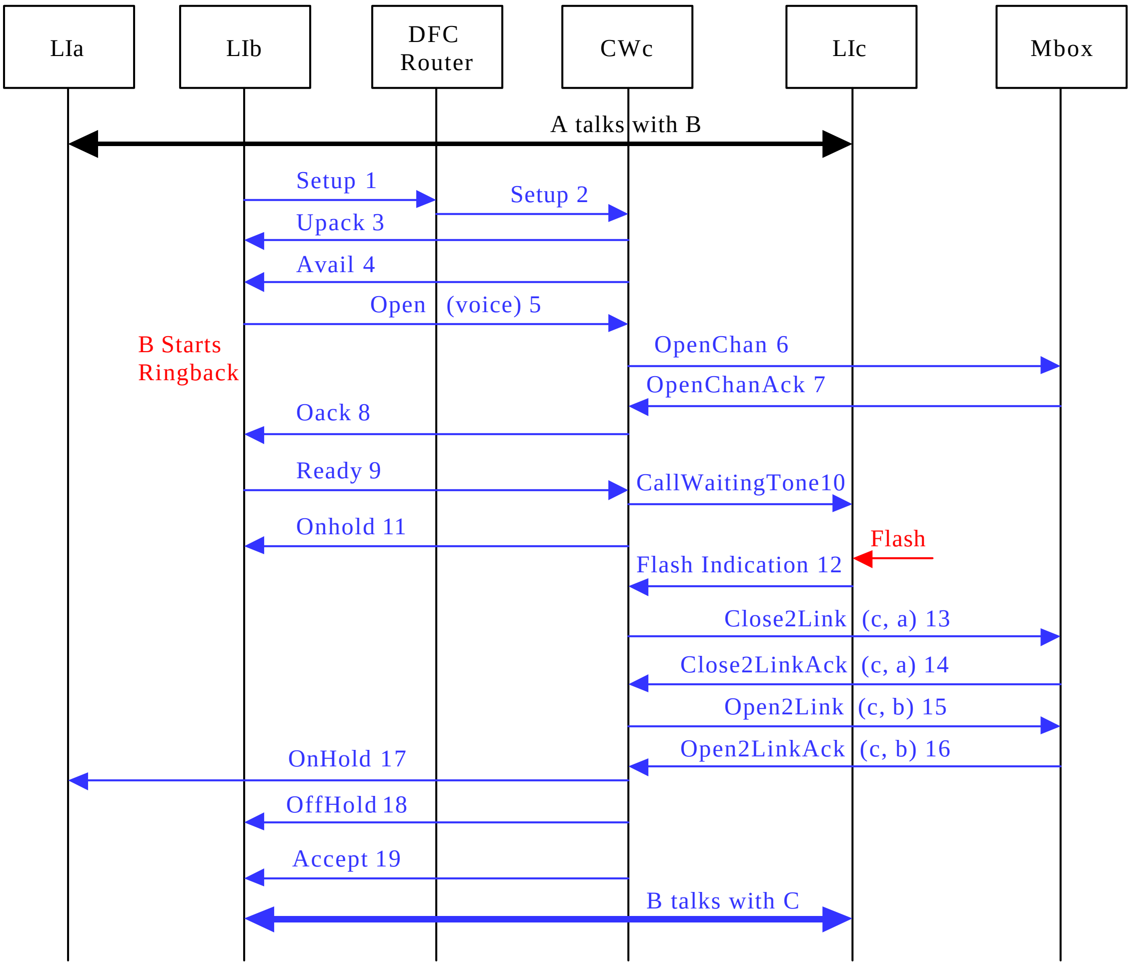


**Figure 5. 8 The Sequence Diagram for Setting Up the Second Customer Call Using the CW Feature Box**

As mentioned before, in the current ECLIPSE implementation, some messages are sent and received in the local host. In fact, if the CW feature box and its assigned Mbox are deployed in the DFC router, thirty messages will be transferred over the network to set up this non-linear usage that includes two customer calls. Should the CW feature box be deployed in the end devices, thirty-four messages will be transferred over the network.

During this non-linear usage, each time the CW feature box recognizes a FLASH signal, five additional messages are needed to switch the media path. One message is used to indicate a FLASH signal, and the other four messages, a CLOSE2LINK message, a CLOSE2LINKACK message, an OPEN2LINK message and an OPEN2LINKACK message, are used to switch the CW feature box's media path. If the CW feature box and its assigned Mbox are placed in one network node, one message indicating a FLASH signal will be transmitted over the network. Should the CW feature box be placed in the end devices, four messages to switch the media path will be transferred over the network. In the teardown phase, placing the CW feature inside the network also requires fewer messages to be transmitted over the network in the current DFC implementation.

Therefore, placing the CW feature box inside the network requires fewer messages to be transmitted over the network. It can be concluded that in the current DFC implementation, it is better to place the CW feature box inside the network. However, placing the CW feature box inside the network uses more network node resources.

### 5.6.2 The Deployment of the CW Feature in the SIP protocol

Figure 5.9 shows the call flow of the CW feature residing in an end device. In Figure 5.9, while user C, who subscribes to the CW feature, is talking with user A, user

B attempts to call user C. The INVITE message issued by the UA of user B will be forwarded to the UA of user B by the SIP proxy server. Due to the end device’ s busy state, the CW feature that is placed in the end device generates a call waiting tone to indicate a new incoming call. It then monitors the media stream for a FLASH signal. If the CW feature recognizes a FLASH signal, it will send the SIP proxy server an INVITE message with c field=0 of Session Description Protocol (SDP), which puts a user on hold. The SIP proxy server will forward this INVITE message to user A. User A will temporarily stop the media streams on receiving the INVITE message. Afterwards, user B will establish a session with user C. As shown in Figure 5.9, the number of messages used to set up this call will be fifteen if the CW feature resides in the end device.

Figure 5.10 shows the call flow of the CW feature placed in the proxy server. The SIP proxy server will receive a "486 Busy" response after it forwards the INVITE message issued by the UA of user B to the UA of user C. The CW feature is evoked for the busy treatment. A call waiting tone indication is then sent to user B and a "182 Queue" response is sent to user C. Afterwards, the CW feature will monitor a FLASH signal. If the CW feature catches a FLASH signal sent by user C, it will send user A an INVITE message with SDP's c field=0, which makes user A stop the media stream temporarily. User B will establish a session with user C once a "200 OK" message is received. Should the CW feature be provided in a separate service node, more messages will be required between the proxy server and the service node.

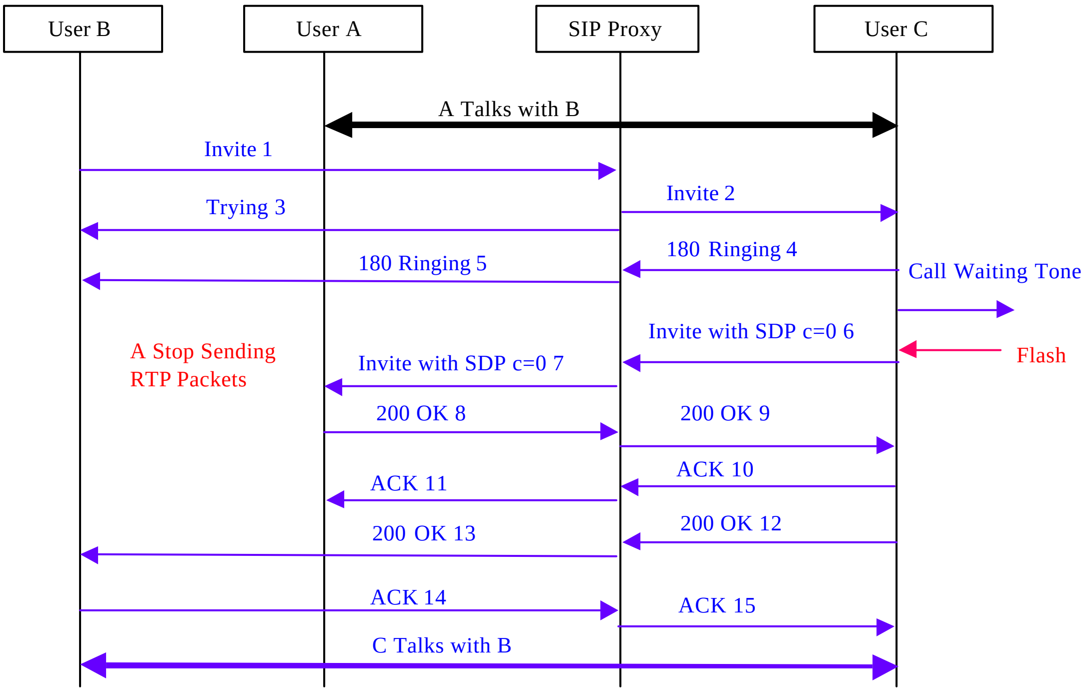


**Figure 5. 9 The Call Flow of the CW Feature Residing in an End Device**

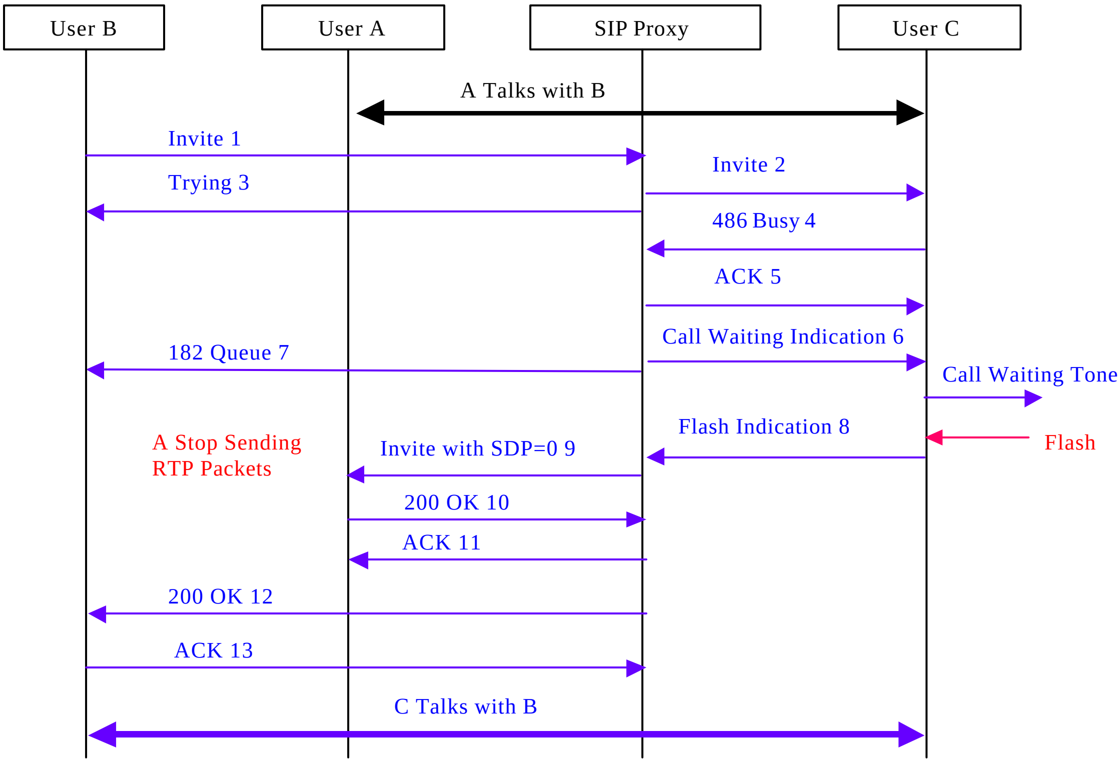


**Figure 5. 10 The Call Flow of the CW Feature Residing inside the SIP-enabled Network**

Comparing Figure 5.9 with Figure 5.10, it is clear that placing the CW feature in an end device does not need more messages, but costs fewer network node resources.

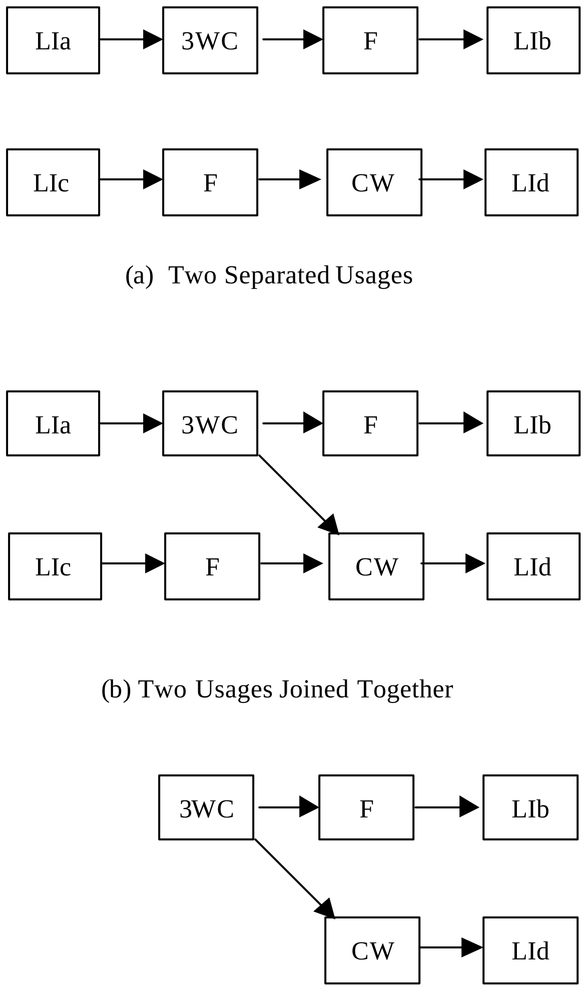
In addition, placing the CW feature inside the network also requires signalling messages for indicating a FLASH signal, which are not defined in the current SIP protocol. Should the CW feature be placed in the end devices, events related to the CW feature can be completed in the end devices without having to introduce new header fields or new messages. Thus, in the SIP protocol, it is better to deploy the CW feature in the end devices.

## 5.7 The Deployment of the 3WC Feature

The 3WC feature adds an additional party to an existing customer call.

### 5.7.1 The Deployment of the 3WC Feature in the DFC Architecture

In the DFC architecture, the 3WC feature can be implemented as a bound feature box. User A, who subscribes to the 3WC, first makes a customer call to user B. This usage goes through the 3WC feature box transparently. If user A tries to call user C when he is talking with user B, the 3WC feature box will place a new internal call that is directed to user C in its third port. The 3WC feature box will mix media streams of user A and user C.



(a) Two Separated Usages

(b) Two Usages Joined Together

(c ) The Continuing Use of the 3WC feature box

**Figure 5. 11 The 3WC Feature in the DFC Architecture**

A usage configuration [17] including the 3WC feature box is shown in Figure 5.11. It is assumed that User A subscribes to the 3WC feature and user D subscribes to the CW feature. In Figure 5.11 a, two clearly distinct usages are shown: the LI interface of user A is connected to the LI interface of user B in one usage; and the LI interface of user C is connected to the LI interface of user D in another usage. These two usages will be joined together (shown in Figure 5.11 b) if user A makes another customer call to user D using his 3WC feature. If both user A and user C drop out, user B and user D can continue to talk to each other, as long as the 3WC feature box that was originally introduced in the usage of user A and user B is still used (shown in Figure 5.11 c).

If the 3WC feature box is placed in an end device and the end device is turned off or is not connected to the Internet during the usage, the usage shown in Figure 5.11 (c) may be affected and not maintained. Therefore, the 3WC feature has to be placed inside the network.

### 5.7.2 The Deployment of the 3WC Feature in the SIP protocol

It is assumed that User B subscribes to a 3WC feature. User A first makes a customer call to user B, then user B invites user C to join the existing call. The scenario for the 3WC feature is as follows [20]:

Firstly, user A sends user B an INVITE message to establish a customer call between user A and user B. The customer call between user A and user B is a normal SIP customer call. User B then sends user C an INVITE message to establish another customer call between user B and user C. This customer call will also be a normal SIP customer call.

If the 3WC feature is placed in the end device, user B can mix the media streams to send the media originating at user A to user C and send media originating at user C to user A. If user B drops out of the customer call, the entire call will be terminated.

If the 3WC feature is placed within the network, the SIP proxy server or a separate service node will mix the media streams of user A, user B and user C. The media streams therefore have to pass the proxy server or the service node, which introduces overhead for the media streams. In addition, placing the 3WC feature inside the network increases the cost to the resources of the network nodes, but does not save the number of messages.

Therefore, in the SIP protocol, placing the 3WC feature in the end devices is more desirable than inside the network.

## 5.8 The Deployment of the VM Feature

The VM feature will automatically take the caller's messages when its subscriber is unavailable or on another call. It is widely used in traditional telephone networks. In Internet telephony, the VM feature should be provided inside the network if its subscribers are offline. Otherwise, the VM feature can be provided inside the network, and it can also be provided in an end device like a traditional answer machine. In addition, the VM feature can be deployed dynamically where it is needed at run time.

### 5.8.1 The Deployment of the VM Feature in the DFC Architecture

In the DFC architecture, the VM feature can be implemented as more than one feature box [34]. These feature boxes have to write, read or modify their operational data in order to perform their functions. In the current ECLIPSE implementation, operational data is stored inside the network. These feature boxes placed in the end device have to access to its operational data over the network. On the other hand, the VM feature has to send the remote assigned Mbox commands to control the media path even if it is executed and retrieves messages stored in the end devic es. Placing the VM feature in the end device usually require more messages that are transmitted over the network, so it is better to place the VM feature inside the network if it is implemented as feature boxes.

If the VM feature resides in an end device, retrieving messages stored in the end devices is likely to be completed in the end devices without the participation of the network. However, this approach may lose the benefits of modularity in the current ECLIPSE implementation.

In addition, the VM feature has to be provided inside the network and messages should be stored inside the network when subscribers are offline. The VM feature and its deployment in the DFC architecture will be discussed in the next chapter in more detail.

### 5.8.2 The Deployment of the VM Feature in the SIP protocol

The VM feature stores messages and plays them back. It is better to separate the VM system from the SIP proxy server if the VM feature is placed inside the network, since the main function of a SIP proxy server is to handle requests from a number of end users. This separation not only allows for the provision of more storage for voice messages, but also helps in building scaleable systems [29].

### (1) Placing the VM Feature inside the SIP-enabled Network

User A attempts to call user B who subscribes to a VM feature. After the SIP proxy server receives an INVITE message sent by the UA of user A, the SIP proxy server will fork the INVITE message and proxy the customer call to user B and a VM system. If user B picks up the phone, the branch to the VM system will be cancelled [29]. The call between user A and user B will be a normal SIP call.

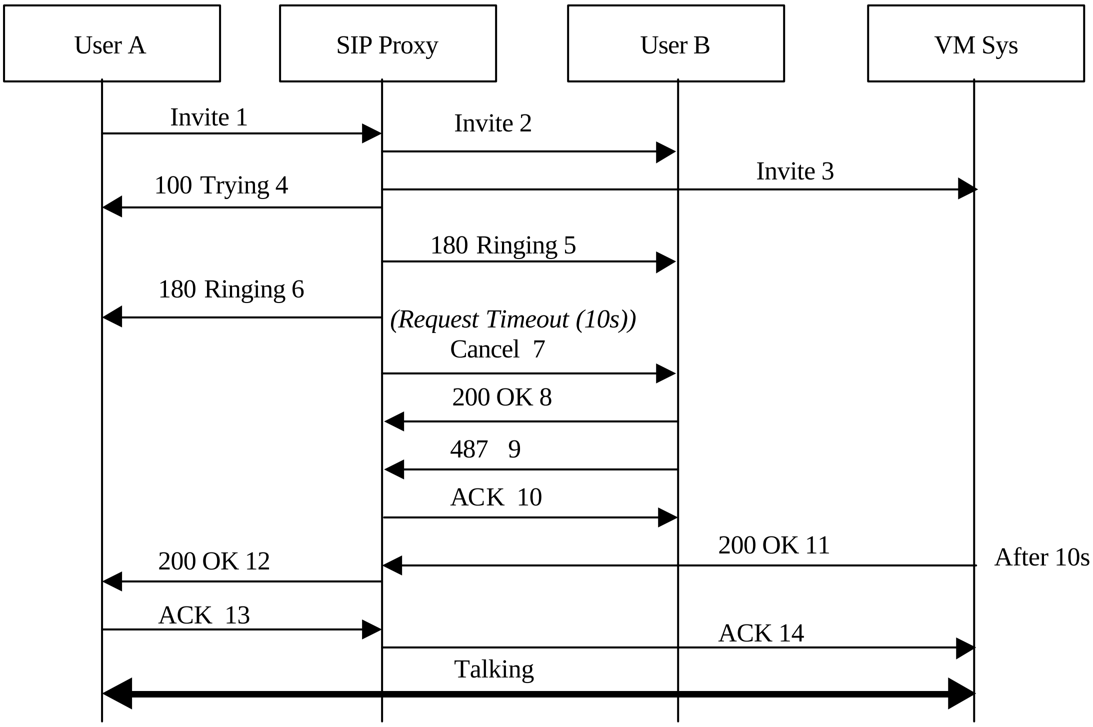


**Figure 5. 12 The Call Flow of the VM Feature residing inside the SIP-enabled Network (The VM feature takes the caller's message when nobody answers the call)**

If user B does not pick up the phone (Figure 5.12) within X seconds (for example, 10 seconds), the VM system will wait for 10s and then accept the customer call for sending a "200 OK" response to user A. If user B is busy, the VM system will also accept the customer call. After the VM system receives an ACK message issued by the UA of user A, the VM system will play back a greeting message stored in the VM feature, prompting user A to leave his message. Once user A has finished recording, he hangs up, triggering a SIP BYE request. The session between the VM system and user A will be terminated.

It is possible that the SIP proxy server and the VM system have different clocks, which brings a different length of time. If the timer of the VM system is earlier, the VM system will be activated due to timeout. But the branch to the user B is not cancelled. In this case, if user B picks up his phone, the VM system should provide an option for the users to stop the VM system and continue talking in a normal call [29].

Another approach to forward an incoming call to the VM system is that the proxy server redirects an incoming call to the VM system if user B is busy or does not pick up the phone within 10 seconds. After the session between user A and the VM system is established, the VM system will play back a greeting message and user A can then leave a message in user B's mailbox.

If the VM system takes a new message and if its subscriber is online, it will notify the subscriber of the arrival of the new message. The subscriber can send a request to retrieve his message.

One way to retrieve a message that is stored inside the network is that its subscriber makes a special customer call whose target address is the VM system. After setting up a session between user B and the VM system, user B can retrieve messages stored in his mailbox. User B can also delete messages or download them into his end device.

In the VM system, end users usually leave or modify some greeting message in the mailbox. One way to record or modify a greeting message is fairly similar to that of retrieving messages. As when retrieving messages, user B makes a special customer call whose target address is the VM system. After setting up a session between user B and the VM system, user B can leave or modify the greeting message in his mailbox.

### (2) Placing the VM Feature in an End Device

Some end users may prefer to place the VM feature in their end device. In this situation, the VM feature looks like an answer machine. If user B is busy or does not pick up his phone in 10 seconds, the UA of user B will forward an incoming call to the VM system that is placed in his local end device. The VM system will send a "200 OK" response to user A. After the VM system receives an "ACK" message sent by the UA of user A, the session is established between user A and the VM system. The VM system plays back the greeting message which prompts for user A to leave a message. Figure 5.13 below shows the call flow of the VM system placed in an end device.

Comparing Figure 5.12 with Figure 5.13, it is evident that placing the VM system in an end device will save the number of required messages for recording callers' messages.

If the VM system resides in an end device, retrieving messages and recording greeting messages can be completed at the end device without the participation of the network. End users can retrieve messages and record greeting messages even if they are offline. If the VM system is placed inside the network, retrieving messages and recording greeting messages will require more messages to establish a session between end users and the VM system.

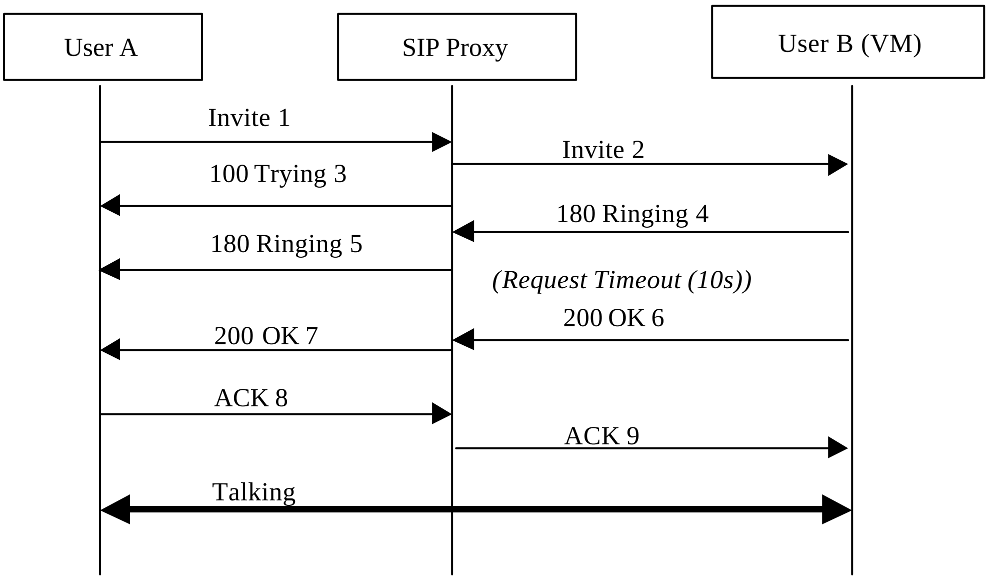


**Figure 5. 13 The Call Flow of the VM Feature residing in an End Device (The VM feature takes the caller' s message when nobody answers the call)**

In the SIP protocol, comparing the VM system placed in an end device with that residing inside the network, it is more desirable to place the VM system in an end device. But the VM system residing inside the network is available no matter whether end users are offline or not.

## 5.9 Summary of the Deployment of Features in the DFC Architecture and in the SIP Protocol

The study of feature deployment is based on the DFC architecture and the SIP protocol respectively. In these two service architectures, network features should be provided inside the network since they require the support of the network. Since these two service architectures have different characteristics, the deployment of client features in these two service architectures is significantly different. In the SIP protocol, it is better to deploy most client features in end devices if end users are online. It is more desirable to place most client features inside the network in the DFC architecture.

**Table 5. 2 A Summary of the Deployment of Features in the Current DFC Implementation and in the SIP Protocol**

| Abbreviation | The SIP protocol | | The DFC Architecture | |
|---|---|---|---|---|
| | **Inside the Network** | **In an End Device** | **Inside the Network** | **In an End Device** |
| 3WC | | Preferable | Required | |
| AD | | Preferable | Preferable | |
| CFND | Required | | Required | |
| CFNA | | Preferable | Required | |
| CFB | | Preferable | Required | |
| CFU | Preferable | | Required | |
| CW | | Preferable | Preferable | |
| IMCW | | Preferable | Preferable | |
| LNR | | Preferable | Preferable | |
| OCS | | Preferable | Preferable | |
| PSC | | Preferable | Preferable | |
| SCF | Preferable | | Required | |
| SCFB | | Preferable | Required | |
| SCFNA | | Preferable | Required | |
| VM | | Preferable | Preferable | |

Note:

"Preferable" means that it is better to place this feature in this location (the end device or inside the network).

"Required" means that this feature should be placed in this location (in the end device or inside the network).

Table 5.2 gives a summary of the deployment of client features based on the SIP protocol and the DFC architecture respectively, which mainly concerns with pure signalling messages. In the current DFC implementation, it is assumed that client features are implemented as feature boxes. Since end devices will be permanently connected to the Internet in the long term, end devices are assumed to be online.

However, since Internet telephony will still contain a large number of dumb end devices connected via gateways, basic versions of most features have to be available inside the SIP-enabled or DFC-enabled network. Some client features, such as the VM feature, should be provided inside the network if end users are offline.

## 5.10 Comparison of the DFC Architecture and the SIP Protocol

The SIP protocol with the CPL language and the DFC architecture can be regarded as two service architectures of Internet telephony. Each one has its advantages and disadvantages, which are discussed below.

The DFC architecture has many advantages [17, 30]. One of its chief advantages is that interactions between feature boxes are limited by a strict sequence. It also offers benefits in feature specification, analysis and implementation, since features are treated as independent feature boxes. Furthermore, each independent feature box can be implemented by third parties, which are dedicated only to providing services with no intention of providing actual media transport.

The DFC architecture makes it easy to reuse the existing feature implementations. However, this modularized architecture introduces signalling overhead. In the DFC architecture, a SETUP message is used to construct a usage. The DFC router routes a SETUP message from one box to another. An INVITE message in the SIP protocol is used to establish a session, but there is no need for an INVITE message to go through features. Comparing Figure 5.14 with Figure 5.15, the DFC architecture requires more signalling messages to establish a customer call than the SIP protocol.

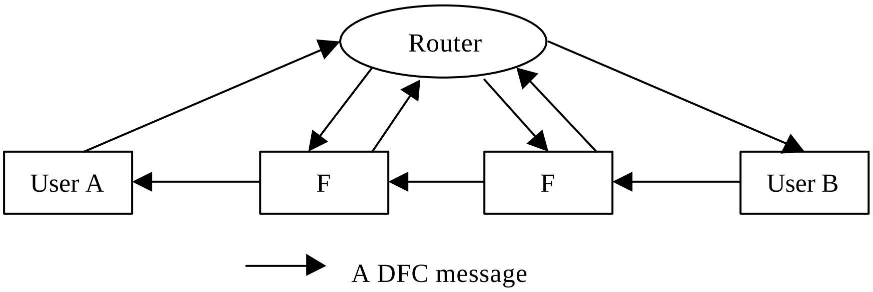


**Figure 5. 14 Establishing a Usage in the DFC Architecture**

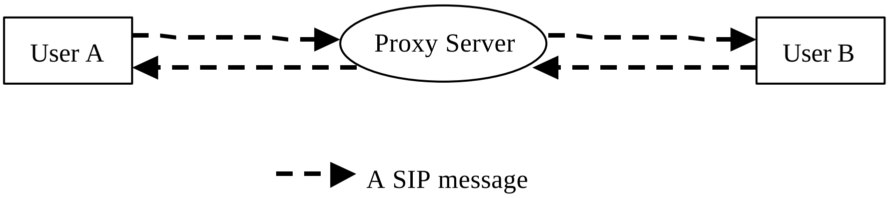


**Figure 5. 15 Establishing a Customer Call in the SIP protocol**

If feature boxes are placed in the DFC router, which is currently used in the ECLIPSE implementation, the transfer of SETUP messages from one feature box to another takes place in the local host. However, deploying feature boxes in the end devices usually significantly lengthens the signal paths for setting up a usage.

During a usage, even if all the feature boxes are in a transparent state, signalling messages have to go through these feature boxes. Because a signalling message may cause a transaction from the transparent state to another state. For example, the CW feature box monitors its media path for a FLASH signal when it is in a transparent state. However, signalling overhead is introduced.

Separating the signalling layer from the media layer also introduces signalling overhead. A feature box should send commands to its assigned Mbox for media controls (shown in Figure 5.1 and Figure 5.2). Since it is not feasible to place Mboxes in the end devices, FBs residing in end devices have to send remote Mboxes commands to control the media path. Signalling overhead will significantly increase when FBs reside in the end devices.

Feature boxes are assembled dynamically for each usage in the DFC architecture. This scheme has both advantages and disadvantages. The advantage is that resources are occupied only when they are needed. The disadvantage is that the dynamic assembly scheme is difficult to apply for the features that are inherently static. For instance, since the Presence feature is independent to any specific usage and should be on all the time, it is difficult to deploy in the DFC dynamic feature assembly scheme.

In Internet telephony, it will likely be important to have mechanisms that simplify the creation of features. Feature boxes are implemented as Finite State Machines (FSM). Compared with the DFC architecture, the CPL language accompanying the SIP protocol provides a simpler way to implement features.

The SIP protocol makes more use of the strong signal processing abilities of intelligent end devices. In the SIP protocol, it is better to deploy most client features in end devices. However, the SIP protocol does not provide a concrete scheme to deal with feature interactions, which, it is believed, will become worse in Internet telephony.

In addition, since the SIP protocol is mainly used to establish and tear down a session, it does not define header fields or messages to carry information during a session, which are required for some features to be executed. For instance, if the CW feature resides inside the network, the UA of an end user must send a message that indicates a FLASH signal. Upon receiving a FLASH signal, the CW feature can place an existing customer call on hold and take another customer call. Therefore, in the SIP protocol, new header fields or new messages must be introduced in order to execute some features.

Generally speaking, the SIP protocol is more distributed while the DFC architecture is more centralized, in which most work is done inside the network.

## 5.11 Limitations of the Current DFC Implementation

In the current ECLIPSE implementation, since operational data is stored inside the network, some feature boxes have to write, read or modify their operational data over the network if they are executed in end devices. This introduces communication overhead. This overhead can be eliminated if the DFC implementation allows operational data to be stored in the end devices.

In addition, each feature box must send commands to its assigned Mbox for media controls. In fact, since some simple feature boxes, such as the LNR feature and the OCS feature, only deal with the signalling layer, commands for media controls are not essential. If the FBs that do not concern with the media layer do not send commands to their assigned Mboxes, this overhead will be reduced.

Overall, the current DFC implementation needs to be improved as it introduces a great deal of signalling overhead, especially when some feature boxes are placed in the end devices. Here, several approaches are proposed to improve the current DFC implementation.

(1) The DFC implementation should allow operational data to be stored in end devices. A backup file of operational data can be stored inside the network. Thus, feature boxes executed in end devices can read, write or modify their operational data locally without having to send the request over the network.

(2) The SFB feature box should be introduced to handle features such as the AD feature and the OCS feature, which are only concerned with the signalling layer. For example, if the OCS feature is implemented as an SFB feature box, and if its operational data is stored in the end devices, it can be fully executed in the end devices without the participation of the network. Placing it in an end device reduces costs to DFC routers, but does not require more messages that are transmitted over the network. This approach, in which the SFB feature box does not involve media controls, provides an optimized solution and improves the system performance.

(3) End devices should be allowed to control the logical end-to-end media path, in which two media terminals are in the same host. A special Mbox should be introduced to handle this special situation. In the current implementation, the VM feature residing

in the end devices has to send the remote Mbox commands for retrieving messages stored in end devices. This is not efficient. Using this approach, the VM subscriber can easily retrieve messages stored in his end device locally if the VM feature is executed in end devices.

However, the proposed approaches will bring multiple versions of implementations. For instance, since a feature box that is executed in an end device can write or modify its operational data locally, and a feature box that is executed inside the network can access its operational data stored inside the network, their codes will be different. Thus, multiple versions of implementations should be provided.

# Chapter 6 Design of Voice Mail in The DFC Architecture

In the DFC architecture, since the VM feature involves signalling messages for storing operational data and controlling the media path, since it interacts with other features, and since it can be deployed in an end device or inside the network, its design is quite challenging. This chapter sketches out the design of the VM feature based on the DFC architecture.

The proposed VM feature consists of three feature boxes: a Receive feature box, a Retrieve feature box and a RecordGreet feature box. All three feature boxes are implemented as free feature boxes residing in the target zones. The desirable order of some feature boxes for busy treatments is also presented in this chapter.

## 6.1 The VM Feature in Internet Telephony

In traditional telephone networks, end devices – such as traditional phones, answer machines and faxes – are usually permanently connected to the network. Although end devices such as PCs may be permanently connected to Internet in the future, at present they may be turned off or only connected to the Internet every so often. The VM feature should be appropriate to this characteristic of Internet telephony.

The VM feature should be provided inside the network when its subscribers are offline. Otherwise, the VM system cannot take callers' messages. The VM system that resides inside the network can record callers' messages and store them inside the network. Once subscribers connect to the DFC system, the VM system will send them a message indicating that they have messages to be retrieved. If subscribers send a "retrieve messages" request, the VM system will retrieve messages and play them back. The VM system can also delete stored messages if the subscribers send a "delete messages" request. In addition, subscribers can download messages to their intelligent end devices if they wish.

As mentioned before, some end users may prefer to place the VM feature in their intelligent end devices. As far as the DFC architecture is concerned, placing the VM feature in an intelligent end device means that feature boxes related to the VM feature are executed in the end device. If voice messages are stored in the end devices, it is possible to retrieve and delete messages without the participation of the network. In addition, placing the VM system in an end device may also reduce a user's cost for subscribing to a service.

Furthermore, subscribers can choose where the VM system should reside if they are online. This is more convenient for subscribers than an answering machine and the voice mail in traditional telephone networks. Since Internet telephony will continue to use a number of "dumb" end devices connected via gateways, the VM system is placed inside the network by default.

## 6.2 The Architecture of the VM Feature

In the DFC architecture, the VM feature can be implemented by three FBs: the Receive feature box, the Retrieve feature box and the RecordGreet feature box. All three FBs are free feature boxes residing in target zones. The Receive FB plays back a greeting message and takes callers' messages if its subscriber does not answer a call or is busy. The Retrieve FB plays back messages stored in the VM system if its subscriber sends a request to retrieve messages. The RecordGreet FB records or modifies a subscriber' s greeting messages. All three FBs can be provided in end devices or inside the network.

### 6.2.1 The Receive Feature Box

The Receive FB is a free feature box in a target zone. As shown in Figure 6.1, an end user A makes a customer call to another end user, B, who subscribes to the VM system. Thus, the DFC router will route the SETUP message issued by the LI box of user A to the Receive FB that is assumed to be the last FB in this usage. If user B is busy or does not answer the call, the Receive FB will play back a greeting message and record user A's message. Otherwise, the usage will pass through the Receive FB transparently.

The Receive FB includes four ports (shown in the lower section of Figure 6.1). Port a is a callee port that receives an internal call; port b, port c and port d are caller ports that make internal calls to another box. Port c is connected to a PlayRI interface, which is responsible for playing greeting messages, and port d is connected to a RecordRI interface, which is responsible for recording messages. Both the PlayRI interface box and the RecordRI interface box are connected to a storage server that stores messages.

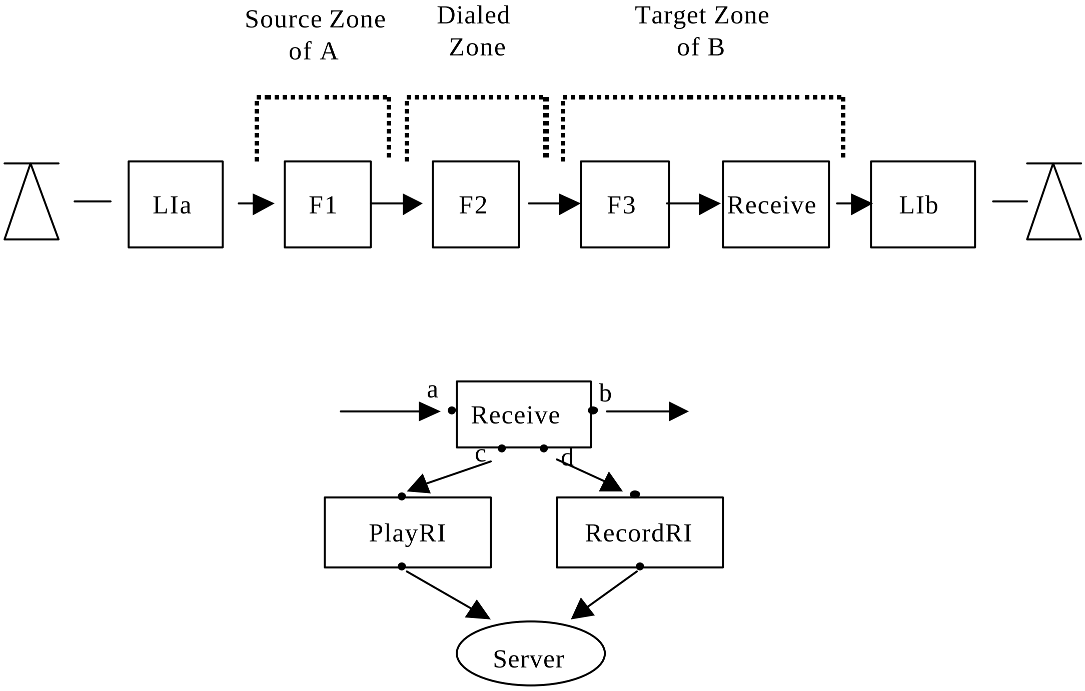


**Figure 6. 1 The Receive Feature Box**

If a Receive FB receives an internal call, port b of the Receive FB will initiate an outgoing internal call directed to the LI box of user B. If user B is offline, busy or does not pick up the phone within X seconds (for example 10 seconds), the first outgoing internal call will be terminated and port c will make another outgoing internal call to the PlayRI interface. The PlayRI interface will play a greeting message, such as "Please leave your message after the beep", if this internal call succeeds. After that, port d of the Receive FB will place an internal call to the RecordRI interface and user A can leave his message in user B's mailbox.

The Receive feature box uses four port aliases: subscriber(sub), connectCaller(conn), player(play) and recorder(rec). Each can refer to port a, port b, port c or port d.

The Receive FB must establish a media path to record a voice message. Figure 6.2 shows the signalling layer and the media layer as they relate to the Receive FB. User A' s end device is connected to the MB1 Mbox, and user B' s end device and the VM storage server are connected to MB2 Mbox. It is assumed that user A' s LI box and the F feature box are given the address of the MB1 Mbox while the Receive FB, user B' s LI box, the PlayRI interface and the RecordRI interface are given the address of the MB2 Mbox. The interface boxes and feature boxes will send commands to their Mboxes for opening and closing channels and links.

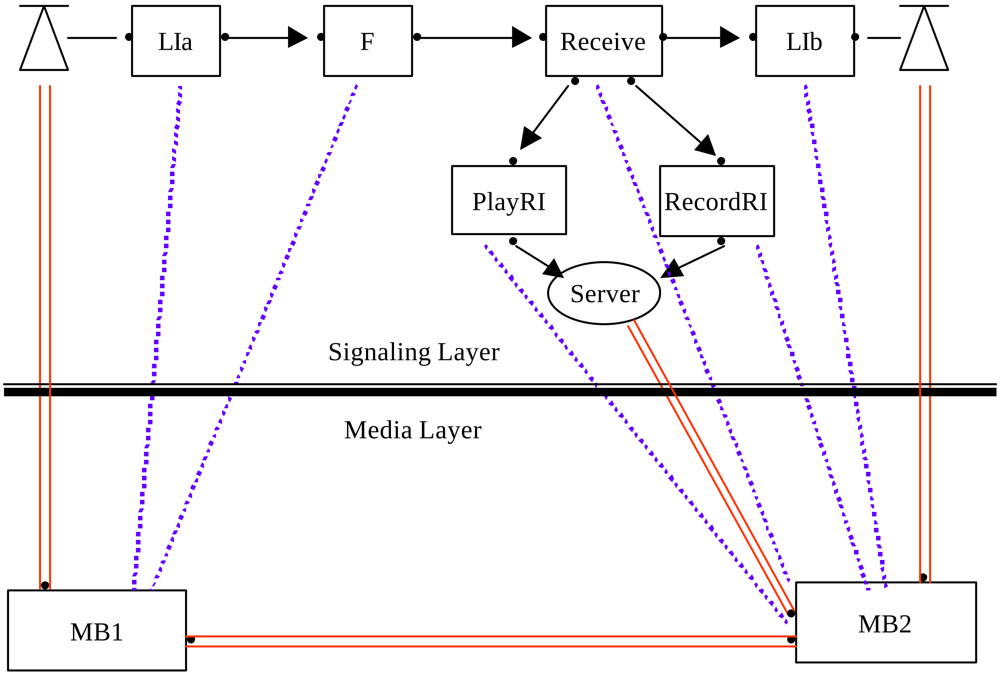


**Figure 6. 2 The Signalling layer and the Media Layer in Relation to the Receive Feature Box**

## 6.2.2 The Retrieve Feature Box

The Retrieve FB is designed for playing back messages stored in the mailbox. It can be implemented as a free feature box residing in the target zone of the PlayRI interface. Retrieving messages can be regarded as a special usage, the target address of which is the PlayRI interface.

The Retrieve FB has one callee port and one caller port (shown in the lower section of Figure 6.3). The caller port of the Retrieve FB will place an outgoing internal call to the PlayRI interface after the Retrieve feature box receives an internal call to retrieve messages.

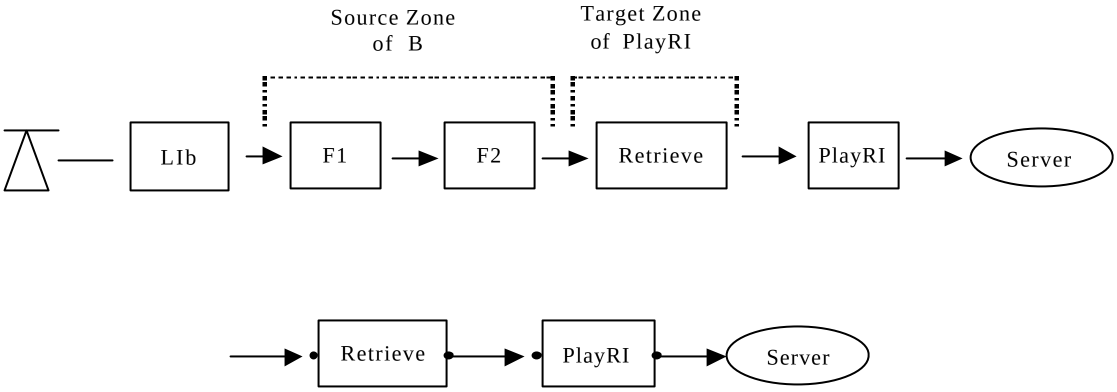


**Figure 6. 3 The Retrieve Feature Box**

Like the Receive FB, the Retrieve FB also must establish a media path to play back stored messages. It will send its assigned Mbox commands the media controls.

## 6.2.3 The RecordGreet Feature Box

The RecordGreet FB is designed to record or modify greeting messages. It can be implemented as a free feature box residing in the target zone of the RecordRI interface. Recording or modifying greeting messages can also be regarded as a special usage whose target address is the address of the RecordRI interface.

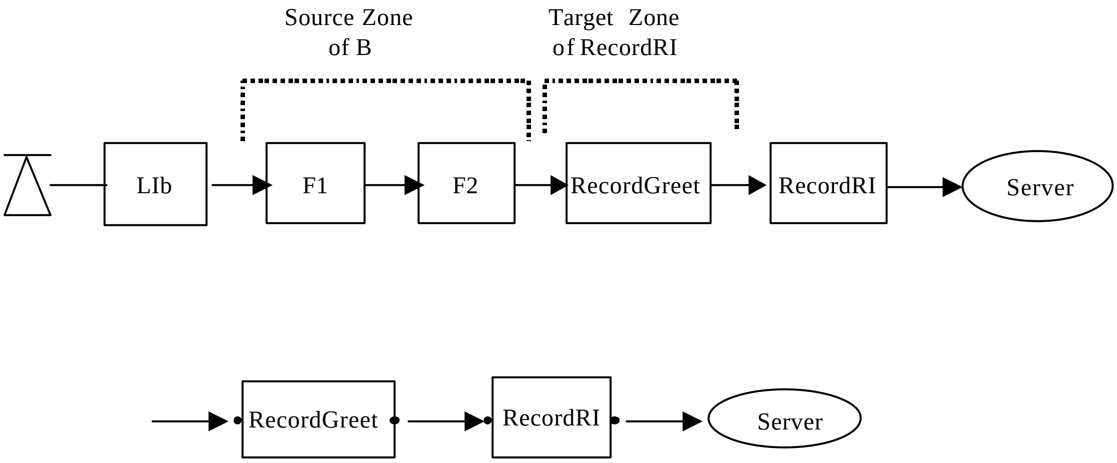


**Figure 6. 4 The RecordGreet Feature Box**

The RecordGreet FB has one caller port and one callee port (shown in the lower portion of Figure 6.4 above). The callee port receives an internal call and the caller port places an outgoing internal call to the RecordRI interface.

Like the Retrieve FB and Receive FB, the RecordGreet FB also sends its assigned Mbox some commands for media controls since it must establish a media path to record or modify greeting messages.

### 6.2.4 The Storage Server

The VM system usually stores a large number of messages. Although messages can be stored in a DFC router, this uses up more DFC router storage space. Storing messages in a separate storage server separates media processing – such as storing, recording and retrieving messages – from the DFC router, whose main function is to route internal calls from one box to another. This separation can reduce load on the DFC router and improve system performance. This separation may also be helpful in building a scaleable system.

Messages can also be stored in a special kind of storage server, end devices. In this case, messages may be retrieved without the participation of the network. It is possible for subscribers to retrieve their messages even if they do not connect to the DFC system. However, messages have to be stored in the storage server that is permanently connected to the DFC system when their subscribers are offline.

Every VM subscriber has his own mailbox on the VM system storage server, which can be identified by the customer. In a mailbox, messages can be stored in multiple "folders". For example, a Greeting folder is used to store greeting messages; a New folder is used to store messages that have not been retrieved; a Retrieved folder is used to store messages that have already been retrieved; and a Deleted folder is used to store deleted messages that subscribers may try to restore later. Messages in the Deleted folder can be emptied by the VM system periodically, such as every 20 days, or can be deleted permanently by subscribers.

In a folder, a message is stored with its recording time, its recording date and its "from" address. A message key, which consists of the VM subscriber information, a message folder and an offset in a message folder, can uniquely identify a message. Using this message key managed by a relational database, a message can be found in the storage server quickly and easily.

Message keys are shared by the Receive FB, Retrieve FB and RecordGreet FB. The RecordGreet FB generates a greeting message key once the subscriber begins to record a greeting message. The Receive FB uses this greeting message key as its operational data to find the greeting message and plays it back. After that, it generates another message key, which is the Retrieve FB's operational data, and records the caller's message. The Retrieve FB uses its operational data to retrieve the message and play it back.

## 6.3 The VM Feature Behaviour

The VM system not only takes a caller's message if the subscriber is busy or does not answer the call, but also plays back or deletes stored messages if the subscriber sends a request to the VM system to retrieve or delete the messages intended for him. In addition, greeting messages can be modified in the VM system. A subscriber has an option to download the messages stored in the network to his end device, or vice versa.

### 6.3.1 Taking Messages

User A attempts to make a customer call to user B, who is subscribed to the VM system. The Receive FB in the VM system will get a message key from its operational data, play back a greeting message, generate a new message key, and take the caller's message if user B is busy, or is offline, or does not answer the call. Four scenarios are discussed below.

(1) The VM system subscriber, user B, is offline.

In this scenario, the Receive FB has to be placed within the network in order to take the caller's message. Messages will be stored in the storage server, which is permanently connected to the DFC system.

A sequence diagram (Figure 6.5) shows signals relating to the Receive FB for the scenario that occurs when its subscriber is offline. In the sequence diagram, a “port:message” notation means that the message is received by this port and a “message(port)” notation means that the message is sent by this port. These two notations are used only for the Receive FB to set up and tear down internal calls.

After the Receive FB receives a SETUP message generated by user A's call attempt, its port conn will send his LI interface a UPACK message and it will place an outgoing internal call (SETUP1), which is directed to user B’s LI interface. Its port sub then monitors the messages coming back to it on its original outgoing call. On receiving a UPACK message, user A's LI interface will send the Receive FB an OPEN message.

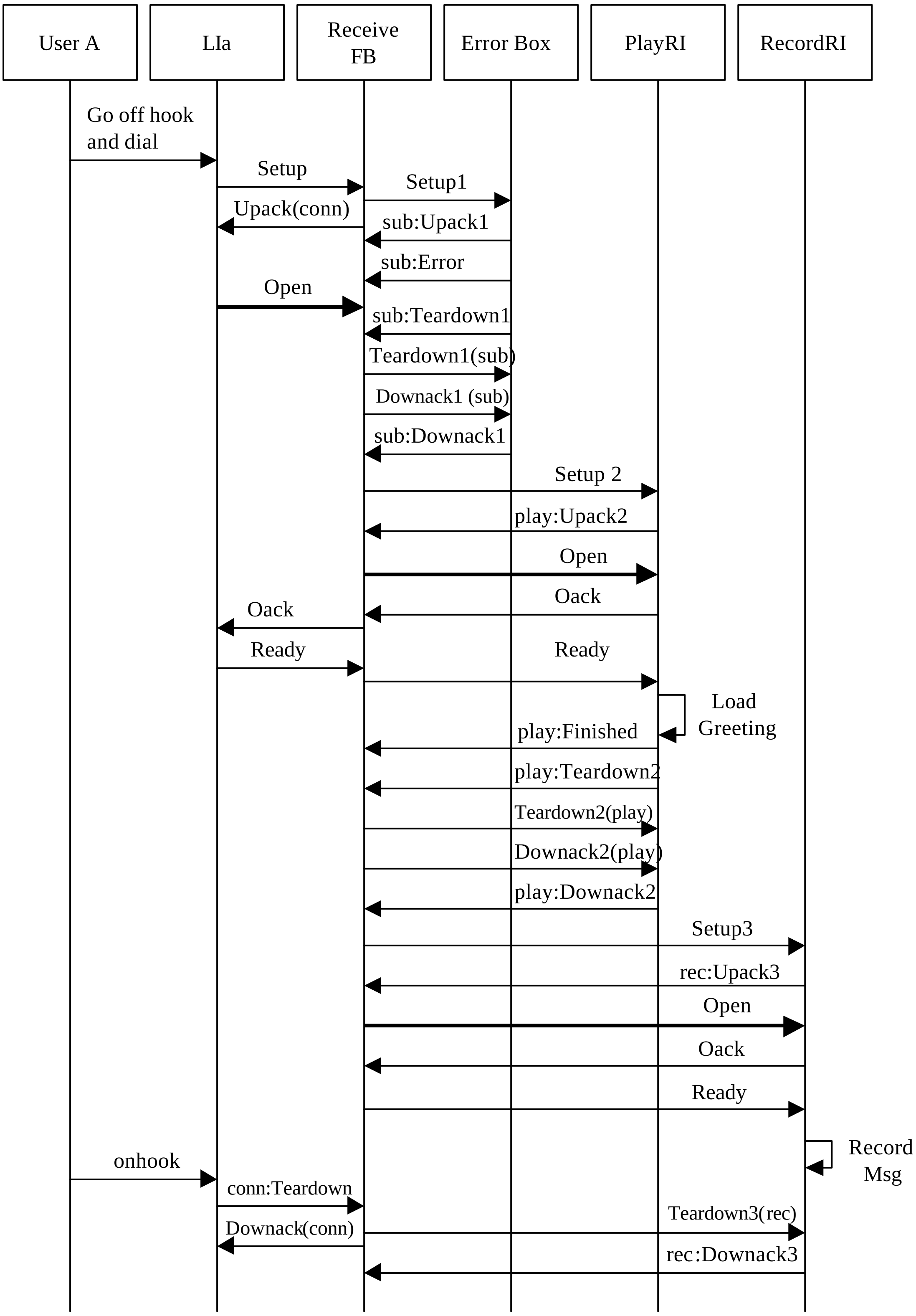


**Figure 6. 5 The Sequence Diagram for Message-Taking in the VM Feature when The Callee is Offline**

The router will try to find user B’ s LI interface after receiving the SETUP message. If user B’ s LI interface is not found, the DFC router will create an ErrorBox that is like a free feature box. The ErrorBox then sends a UPACK message (UPACK1) and an ERROR message to the Receive FB’ s port sub, indicating that the subscriber is offline. Afterwards, both the Receive FB (port sub) and the ErrorBox attempt to teardown this call between them (TEARDOWN1), as both know that it is no longer any use.

The Receive FB’ s port play then places a second internal call (SETUP2) to the PlayRI interface to play back a greeting message. If this internal call succeeds, a media path will be established between the caller's LI box and the PlayRI interface. Once the PlayRI interface has finished playing back the greeting message, the Receive FB’ s port play will receive a FINISHED message. The second internal call will be terminated (TEARDOWN2) by both the Receive FB (port play) and the PlayRI interface.

The Receive FB’ s port rec then places a third internal call (SETUP3) in order to record the caller's messages. If the RecordRI interface accepts the third internal call, the caller can leave a message for the callee. Once the callee hangs up his phone, the Receive feature box will tear down its internal call (TEARDOWN3).

(2) The subscriber is online.

In this case, it is assumed that the subscriber’ s LI box is placed in the end device. The subscriber can choose where the Receive FB resides and where messages are stored. The Receive FB can be placed inside the network or in an end device and messages can be stored in an end device or the storage server that is permanently connected to the network. In both situations, the behaviour of the Receive FB is similar.

If user A makes a customer call, the DFC router will route the SETUP message to the Receive feature box. The Receive feature box will place its first outgoing internal call (SETUP1) directed to the subscriber, user B, as before.

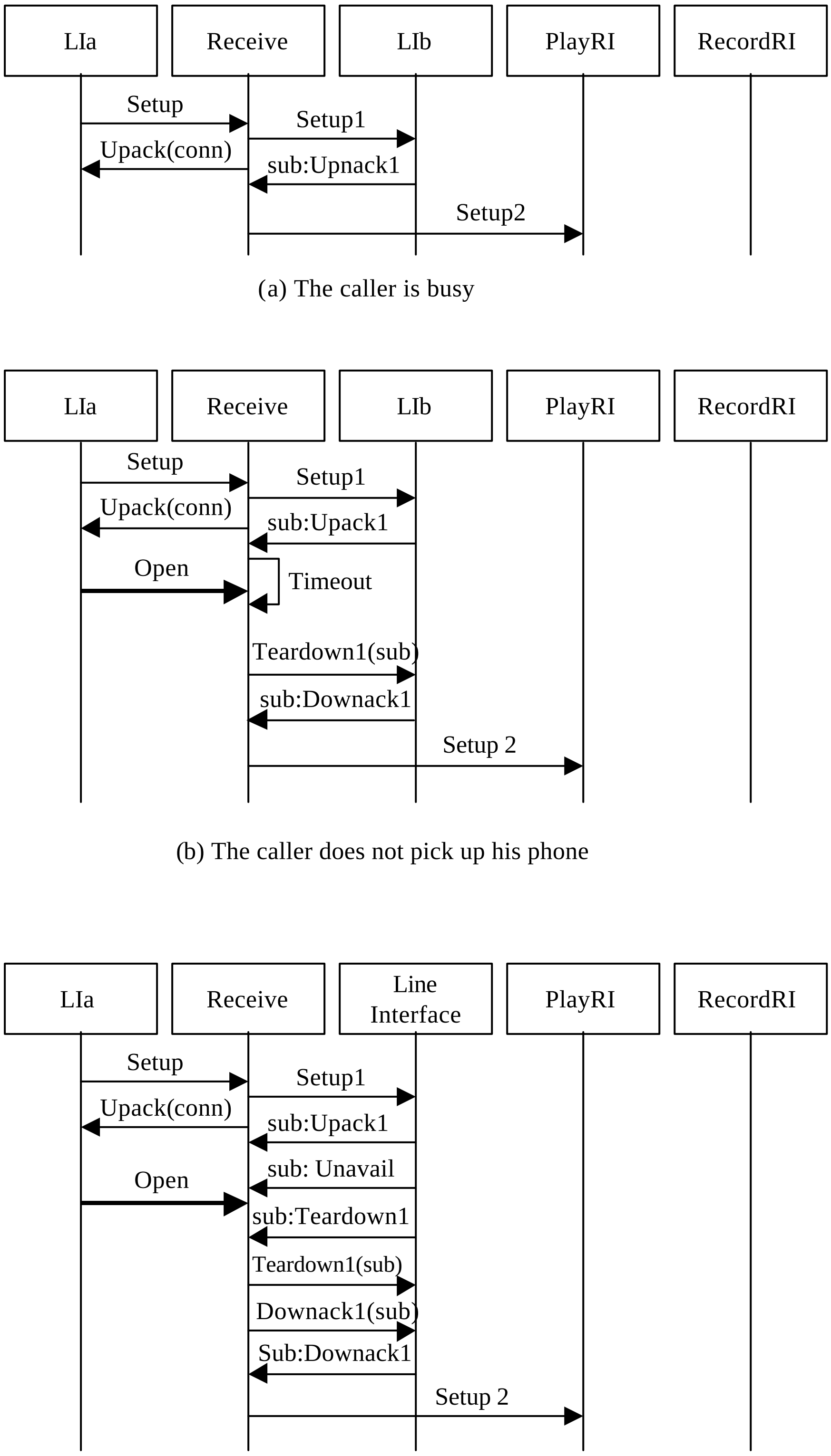
Should the subscriber be idle, his LI box will send a UPACK message (UPACK1) to the Receive FB' s port sub, which enables a timer. If the subscriber picks up his phone to answer the call, the usage will go through the Receive FB transparently and the timer will be cancelled.

If user B is online, there are three scenarios in which the Receive FB will take the caller' messages. These are shown in Figure 6.6.

(a) User B is busy. In this scenario, all the ports of user B' s LI box are occupied with other calls, so his LI box will send an UPNACK message (UPNACK1) to the Receive FB' s port sub, indicating that the first internal call set-up has failed (Figure 6.6 a).

(b) User B is online, but does not pick up the phone within X seconds. In this scenario, the Receive FB tears down its first outgoing internal call (TEARDOWN1) due to time-out (Figure 6.6 b).

(c) User B' s line or device is temporarily unavailable. In this scenario, user B' s LI box will generate an UNAVAIL message, which causes the Receive FB to teardown its internal call with the subscriber' s LI box (Figure 6.6 c).



(a) The caller is busy

(b) The caller does not pick up his phone

(c ) The line or device is temporarily unavailable

**Figure 6. 6 Three Scenarios for the Receive Feature Box when the Callee is Online**

Afterwards, the Receive FB places the second outgoing internal call (SETUP2 in Figure 6.5) to the PlayRI interface for playing back greeting messages, and then places the third internal call (SETUP3 in Figure 6.5) to the RecordRI interface for recording the caller's messages.

### 6.3.2 Playing Back Messages

The Receive FB will send a notification to the subscriber to indicate the arrival of new messages, once the subscriber has connected to the DFC system. The subscriber can send a request to retrieve messages if he wants to. Once he has authenticated himself, the Retrieve FB will play back un-retrieved messages stored in his mailbox. The Retrieve FB retrieves message keys from its operational data. There are two cases to consider:

(1) Messages are stored in the storage server that is permanently connected to the DFC system.

In order to retrieve the messages stored inside the network, a subscriber has to set up a special usage whose target address is the PlayRI interface that, it is assumed, is placed in the storage server. After the Retrieve FB in the usage receives a SETUP message, it will place an outgoing call directed to the PlayRI interface. If this call succeeds, a media path will be established between the LI box of the subscriber and the PlayRI interface. Messages stored in the New folder will be played one by one once the

Receive FB has obtained its operational data.

(2) Messages are stored in local end devices.

The signals to retrieve messages from the local host will be the same as those used to retrieve messages stored within the network. In the current DFC implementation, the Retrieve feature box must send its assigned Mbox commends to control the media path. Therefore the subscriber must be online even if he retrieves messages stored in the local host.

In addition, since operational data is stored within the network in the current ECLIPSE implementation, the Retrieve feature box sends the remote node a request to obtain its operational data. If the Retrieve FB is fully integrated with the LI boxes and resides in the end device, message retrieval can be completed in the end device without the participation of the network.

### 6.3.3 Deleting Messages

Subscribers may want to delete messages stored in the storage server. They can make a special customer call whose target address is the PlayRI interface associated with the storage server. Once the usage has been formed successfully, the subscriber can send a "DELETE" command to the Retrieve FB that is included in the usage. If the "DELETE" command is NULL in the message key field, the Retrieve FB will try to obtain information on all messages stored in the storage server and send it to the subscriber. The subscriber can determine the messages that are to be deleted and send their keys to the Receive FB. If the "DELETE" command contains one or more message keys, which identifies messages to be deleted, the Retrieve FB will delete its operational data associated with these message keys. Then it sends the storage server a command which causes the storage server to move these messages from the Retrieved folder or New folder to the Deleted folder. Subscribers can also delete messages permanently in the Deleted folder, or restore them.

Subscribers can also delete a message after retrieving it. In this case, they can send a "delete" command to the Retrieve FB to delete this message.

In addition, retrieving messages that are stored in end devices can be completed in the end devices if the Retrieve FB is fully integrated with the LI boxes and resides in the end devices.

### 6.3.4 Recording Greeting Messages

When an end user subscribes to a VM system, he will usually record a greeting message. He may modify this greeting message later. Recording or modifying greeting messages is performed by the RecordGreet FB. In the DFC architecture, greeting messages can be stored in an end device, or the storage server which is permanently connected to the network.

(1) Storing greeting messages in the storage server which is permanently

connected to the network.

In order to record or modify a greeting message stored within the network, a special usage, whose target address is the RecordRI interface, is formed. Once the usage has succeeded, a media path will be established between the subscriber's LI and the RecordRI interface. The subscriber can therefore store his greeting message on the network. If the subscriber records a new greeting message, the RecordGreet FB will generate a new message key for this message. If the subscriber modifies a greeting message that is stored on the storage server, the old message will be overwritten by a new greeting message with the same message key.

(2) Storing greeting messages in end devices.

If the RecordGreet FB is integrated with the LI boxes, recording a greeting message can be completed at the end device without the participation of the network. Otherwise, in the current DFC implementation, a subscriber must connect to the DFC system to record greeting messages stored at his local end device.

In addition, subscribers can record or modify greeting messages stored both in end devices and inside the network if they connect to the DFC system.

## 6.4 The FSM of the Receive Feature Box

In the current ECLIPSE implementation, the FSM is developed to simplify the design, implementation, and analysis of feature boxes. It is described as the ECLIPSE Statecharts Language [10]. Inspired by Unified Modelling Language (UML) Statecharts, the ECLIPSE statecharts language was developed by supporting a smooth transition from design to implementation and by supporting automated semantic analysis. The ECLIPSE statecharts language, whose semantics differ somewhat to the UML statecharts, is customized for box programming in ECLIPSE and provides explicit support for DFC concepts such as ports, messages and transparent states.

Figure 6.7 shows the high level FSM of the Receive feature box. In fact, each state, aside from the START state, contains some sub-states nested in this state. In the TRANSPARENT state, there is a nested state, the MEDIA_STATE state, which is responsible for media control. The MEDIA_STATE state is also decomposed into several nested states. Some signalling messages relating to media channels, such as OPEN, READY and OACK in Figure 6.5, are handled in these nested states. A detailed design of nested states in the Receive feature box is beyond the scope of this thesis.

The high level FSM of the Receive feature box shown in Figure 6.7 is specified in the ECLIPSE statecharts language. This statechart only shows the states required to provide the basic functions of the Receive feature box. Transitions are labelled in the form of event [guard]/action, where each label component is optional. Events are message receive operations on a port. Guards are arbitrary Boolean expressions. Actions are arbitrary, which often include send operations on ports. The "port!message" notation is used to send while the "port?message" notation is used is used to receive.

The initial state of the Receive FB is the START state. When the caller makes a customer call to the subscriber, the Receive feature box will receive a SETUP message, which results in a transition from the START state to the LINKCALLER state. During this transition, the Receive FB prepares a SETUP signal for its first outgoing call (SETUP1 in Figure 6.5), copies all the SETUP messages from its incoming call, and assigns port a (Figure 6.1) to conn and port b (Figure 6.1) to sub.

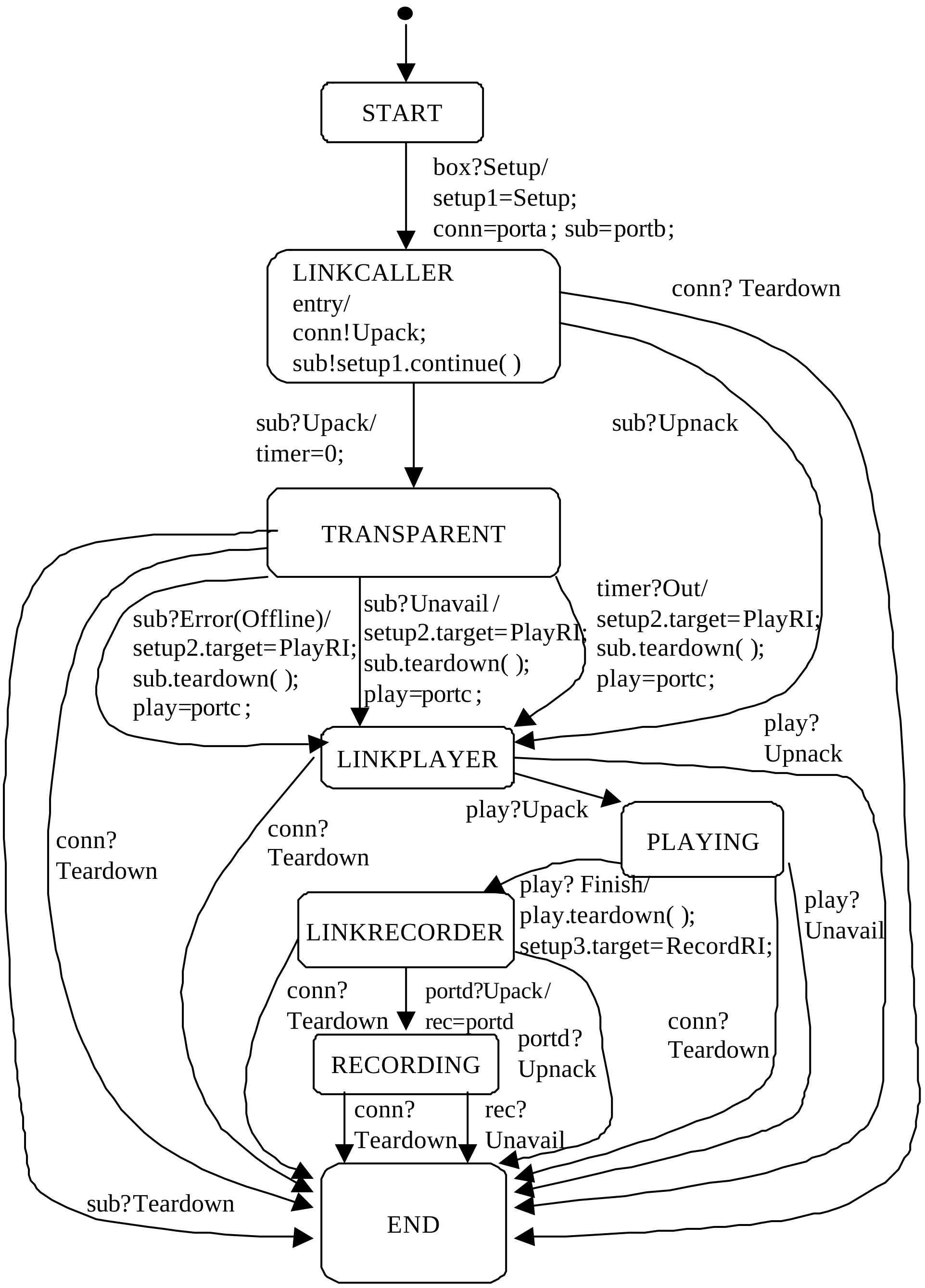


**Figure 6. 7 The High Level State Diagram of the Receive FB**

In the LINKCALLER state, the Receive FB sends a SETUP message (SETUP1 in Figure 6.5 and Figure 6.6), making sure that the command field in SETUP message is "continue". Its port conn signals a UPACK message back to the sender. If the subscriber is busy, an UPNACK event (Figure 6.6 a) occurs and a transition is taken to the LINKPLAYER state. Should the subscriber accept the first outgoing call, a UPACK event (UPACK1 in Figure 6.5) occurs, a timer is enabled and the FSM transitions to the TRANSPARENT state.

In the TRANSPARENT state, the timer will be cancelled if the callee picks up his phone. (The sequence diagrams in Figure 6.5 and Figure 6.6 do not show this scenario). The arrival of an UNAVAIL message indicating that the line or device is temporarily unavailable (Figure 6.6 c), or the arrival of an ERROR message indicating that the subscriber is offline (Figure 6.5) will cause a transition to the LINKPLAYER state. The same transition will also be made due to a timer event (Figure 6.6 b) if the subscriber does not pick up his phone. In the transition from the TRANSPARENT state to the LINKPLAYER state, three actions are performed: port sub  (port b) tears down its outgoing call (TEARDOWN1 in Figure 6.5), play (port) is assigned to port c, and the target address of the second outgoing call (SETUP2 in Figure 6.5) is assigned to the address of the PlayRI interface. In the TRANSPARENT state, the TEARDOWN event made by the caller or the subscriber causes a transition to the END state.

In the LINKPLAYER state, the Receive FB sends a SETUP message to the PlayRI interface (SETUP2 in Figure 6.5). If the second call is rejected, an UPNACK message event occurs and a transition is made to the END state. Should the second outgoing call to the PlayRI interface be successful, the UPACK event (UPACK2 in Figure 6.5) occurs and the FSM of the Receive FB transitions to the PLAYING state, in which a greeting message is played back.

In the PLAYING state, if the PlayRI is temporarily unavailable, port play of the Receive FB will receive a UNAVAIL message. The arrival of the UNAVAIL message causes a transition to the END state. Once message playback is completed, a FINISH event (the FINISH message in Figure 6.5) occurs and the statecharts move into the LINKRECORDER state. During this transition, port play tears down its outgoing call (TEARDOWN2 in Figure 6.5), and the target address of the third outgoing call (SETUP3 in Figure 6.5) is assigned to the address of the RecordRI interface.

When the Receive FB receives an UPNACK message indicating that its third outgoing call (SETUP3) has been rejected, or a TEARDOWN message made by the caller, the FSM transitions from the LINKRECORDER state to the END state. The arrival of the UPACK message (UPACK3 in Figure 6.5) causes a transition from the LINKRECORDER state to the RECORDING state, in which rec (port) is assigned to port d. Once in the RECORDING state, the Receive FB takes the caller' s message. The arrival of a UNAVAIL message, indicating that the RecordRI interface is temporarily unavailable, causes a transition to the END state.

In all states except the START state and the END state, if the caller tears down his customer call, a TEARDOWN event occurs and a transition is made to the END state.

Since the sequence diagram in Figure 6.5 is one scenario of the Receive feature box, some events in Figure 6.7 such as several UPNACK events cannot be shown in this sequence diagram.

## 6.5 The Deployment of Feature Boxes Relating to the VM Feature

As mentioned previously, the feature boxes relating to the VM feature can be provided in the end devices or inside the network.

In the current ECLIPSE implementation, since operational data is stored inside the network, the Retrieve FB and the RecordGreet FB placed in the end devices have to write or modify their operational data over the network. On the other hand, a feature box residing in the end device has to send commands to the remote assigned Mbox for media controls. Placing these feature boxes in end devices usually requires more signalling messages that are transferred over the network, so it is better to place these feature boxes inside the network in the current ECLIPSE implementation.

Even if the current ECLIPSE implementation allows operational data to be stored in the end device, the Retrieve feature box residing in the end device has to send commands to the remote Mbox to retrieve messages stored in the end device. This is not efficient. If the operational data of the feature boxes relating to the VM system could be stored in the end devices, and if the logical end-to-end media path, where two media terminals are in the same host, could be opened and closed by the local host, retrieving messages stored in the end device will be more efficient. However, this would mean significant changes to the current code and structure and results in multiple versions of the feature boxes' implementations.

If messages are stored in the end devices, one approach is to integrate the Retrieve FB and the RecordGreet FB with the LI boxes. In this case, retrieving and recording messages stored in the end devices can be completed in the end devices without the participation of the network. Although this approach loses the benefits of modularity, it saves signalling messages. This approach might provide an optimized solution under the current implementation.

However, the Receive FB cannot be integrated with the LI boxes. In the VM system, the Receive feature box will be activated if the subscriber is busy. If the Receive FB is integrated with the LI boxes, the Receive FB will be the nearest feature box to the LI box. Other busy treatment features, such as the CW feature and the CFB feature, cannot be enabled in that the Receive FB absorbs a busy signal. This is not desirable. Integrating the Receive feature box with the LI boxes will result in "bad" feature interactions.

In the current ECLIPSE implementation, if the VM feature is implemented as separated feature boxes, it is better to deploy them inside the network. Placing them in end devices results in more signalling overhead. Integrating the Retrieve feature box and the RecordGreet feature box with the LI boxes provides an optimized solution if messages are stored in the end devices. However, the Receive feature box has to be placed within the network and messages should be stored on the network when subscribers are offline.

## 6.6 Some Feature Interactions of Voice Mail

The VM feature is one of the busy treatment features. The following discussion focuses on some feature interactions related to the busy treatment features. There are many busy treatment features. The VM feature consists of three feature boxes, in which the Receive FB handles a busy signal. Both the CW feature and the CFB feature are also busy treatment features. If an end user B subscribes to three busy treatment features, the VM feature, the CW feature and the CFB feature, what is desirable behaviour?

In the DFC architecture, a precedence of feature boxes governs the order in which features can occur. This order has important effects on how features interact. Feature boxes close to the source of their triggering signal have a higher priority. What is the reasonable order of these three busy treatment FBs?

The reasonable order is the Receive feature box, the CFB feature box and the CW feature box as shown in Figure 6.8. If the CW feature box is activated, it is better for the CW feature box to supersede the CFB feature box and Receive FB. Since the CW feature box cannot accept more than two incoming calls, these calls can be directed to a new address by the CFB feature box. If the line of the new address is busy or no one answers these calls within X seconds, the Receive FB is activated to take caller's messages.

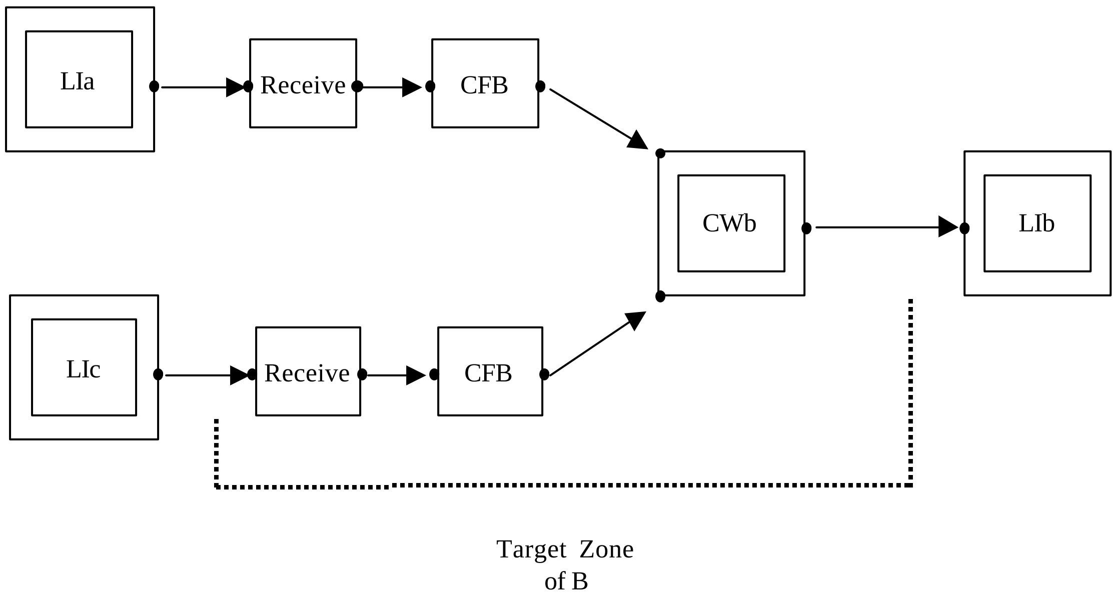


**Figure 6. 8 The Reasonable Order of Some Busy Treatment Feature Boxes**

Firstly, user A makes a customer call to user B, who is presumably connected to the DFC system and is idle. User B picks up his phone and talks with user A. In this customer call, the Receive FB, the CFB feature and the CW feature are in a transparent state.

While user B is talking with user A, user C also tries to call user B. The CW feature accepts the SETUP message generated by user C’ s LI box at its third port. It inserts a call waiting tone in the media path, and then monitors the media path for a FLASH signal. The arrival of the FLASH signal causes the CW feature box to switch its internal media path. If, while talking with user A, user B does not accept the second customer call initiated by user C, the CW feature box cannot receive a FLASH signal. However, the CFB feature box cannot be activated as it has not received an UPNACK message. The Receive FB will be activated because of time-out. User C can leave a message in user B's mailbox.

Perhaps another user, D, will try to call user B a little later. In this situation, the CW feature box has no idle port to accept another internal call. It will signal an UPNACK message back to the previous box (the CFB feature box). After receiving the UPNACK message, the CFB feature box is activated and places another internal call to the forwarded address. If this call succeeds, the Receive FB does nothing and is still in a transparent state. However, if the outcome of this internal call is busy or no one answers the call within X seconds, the Receive FB will be evoked. After playing back a greeting message, it can take user D's message.

Figure 6.9 shows another order of these FBs. In this order, since the Receive FB will absorb and respond to a busy signal, the CFB feature box will never be activated. This order is undesirable and brings a bad feature interaction.

As shown in Figure 6.8, if the Receive FB, the CW feature box and the CFB feature box are properly ordered, the busy situation is handled gracefully.

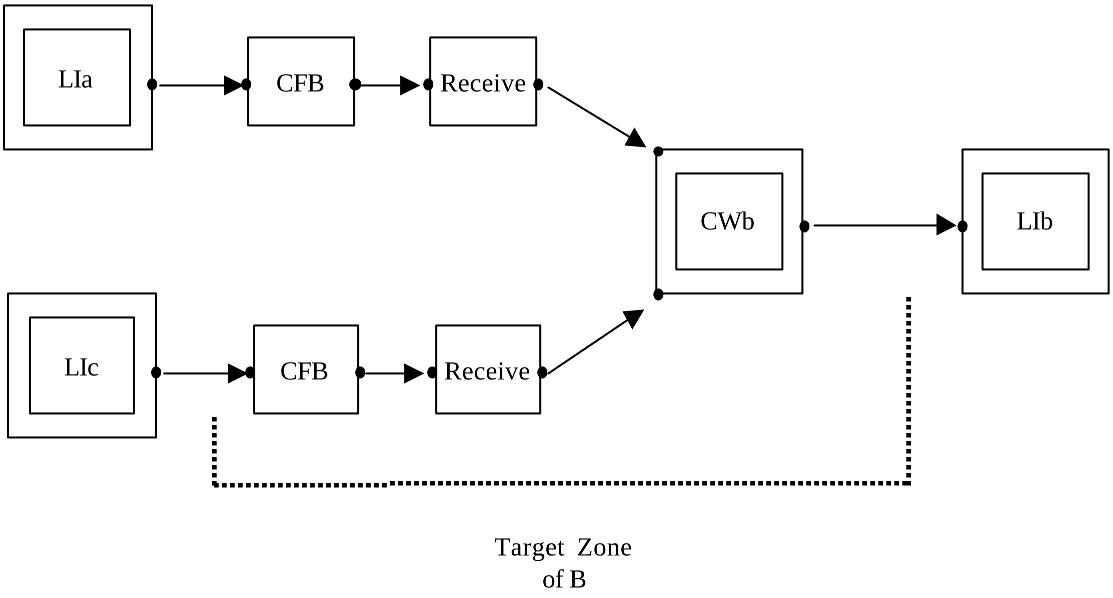


**Figure 6. 9 The Undesirable Order of Some Busy Treatment Feature Boxes**

## 6.7 Variations of the VM Feature

In the Internet environment, there are some variations of the VM feature when integrated with Internet data services. Firstly, the VM feature should be able to record multimedia messages in Internet telephony. It should not only store voice messages, but should also store text and video messages. In a multimedia system, more things can happen during a customer call [34]. For example, a caller tries to open channels for two distinct media. If a callee only accepts one media, the Receive FB may provide an opportunity to store a message in the rejected medium.

In the Internet environment, the VM feature can send its subscribers an e-mail or instant message to announce the arrival of a new message. It can also send subscribers un-retrieved messages as e-mail attachments. Subscribers can leave some messages in their mailboxes, which can be sent to their target addresses by the VM system later. A call might be reclaimed by a subscriber who picks up the phone and answers the call while a message is been recording. The proposed architecture of the VM system has good extendibility to support its variations.

# Chapter 7 Other Issues Regarding the Deployment of Features

The above discussions on feature deployment have primarily revolved around pure signalling messages. There are many factors, such as end devices, users' preference and security, which have an important effect on the placement of features. Issues other than technology also guide the decisions regarding feature deployment in networks. These will be discussed in this chapter.

## 7.1 End Devices

Internet telephony supports various end devices, including intelligent end devices and dumb end devices connected via gateways. As mentioned previously, with regard to dumb end devices, most client features have to be provided inside the SIP-enabled or DFC-enabled network since dumb devices do not know the state of a call and have very limited storage.

It is possible to place client features in intelligent end devices since intelligent end devices can hold the full states of a call or a session. Some “intelligent” devices, such as PCs, have powerful functions, large storage capacity and enough memory to store and execute client features. Therefore, client features can be placed in these intelligent end devices. However, some intelligent end devices, such as wireless devices, may only have small amounts of storage and limited computational capacities to store and run client features, some of which may be very complicated. In this case, basic versions of most client features have to be provided inside the network, although deploying them in the end device may be preferable.

Although client features can be deployed in intelligent end devices or inside the network, most may have to be placed inside the SIP-enabled or DFC-enabled network due to a large number of dumb end devices connected via gateways. End device characteristics should be considered in feature deployment.

## 7.2 The Issue of Feature Interactions

In Internet telephony, features can be provided and deployed by third parties who may have no knowledge of feature interactions, or may not consider feature interactions thoroughly, either through ignorance or expediency. Moreover, as the SIP protocol makes greater use of the strong signal processing abilities of intelligent end devices, end users can easily create and deploy features. If such features are placed inside the network, the service provider can probably track which feature causes undesirable interactions, and solve them. However, this is hard to ascertain without the co-operation of end users if such features have been placed in the end devices.

For instance, interactions occur if several features are programmed to deal with time-out. Multiple expiration timers may cause bad feature interactions. An end user creates the VM feature and places it in his end device, and also subscribes to the CFNA feature resided inside the network. The VM feature takes the caller's message if a timeout occurs. The CFNA feature redirects the call once the expiration has elapsed. If the VM feature and the CFNA feature have different timers, which feature is executed first depends on its timer. It is possible that the CFNA feature will never be executed. Although this kind of feature interaction also arises when features are placed within the network, it can be relatively easily solved by the service provider.

Furthermore, the SIP protocol provides end-to-end connectivity so the signalling server cannot force calls to be placed through it. As an end device can be programmed to communicate directly with the remote party, bypassing local administrative controls entirely, the OCS feature that is placed in the network cannot be enabled. In this case, it has to reside in the end device.

In the DFC architecture, if the Receive FB is integrated with the LI boxes, it will absorb a busy signal. Thus, other busy treatment features cannot be enabled. The order of the FBs has an important effect on how features interact. This order is currently determined by the DFC system. If features are implemented as independent FBs, the order of the FBs will not be affected no matter where the FBs are executed. However, if the DFC architecture allows the order of the FBs to be determined by third parties, third parties may bring the improper order which cause undesirable feature interactions.

In addition, in the DFC architecture, some of the feature boxes of client features, such as the CFU feature box and the 3WC feature box, should be executed inside the network. As mentioned previously, since they are still in the chain of boxes although the usage does not reach the subscriber, the existing usage may be affected if they are executed in the end device.

Overall, deploying features in end devices complicates feature interactions. Since traditional telephony networks assume that end devices are dumb, feature interactions rarely occur between end devices and the network. But this kind of feature interaction is more likely to occur in the Internet environment, which is still an open issue [30]. Further research on feature interactions is beyond the scope of this thesis.

## 7.3 Feature-implementation Delivery

In deploying client features, the method of delivering executable features across the network, if these features are not executed locally, must be examined.

“Download “ or “upload” is one of the most common technologies used to deliver the feature-implementation programs from/to network nodes. Downloading or uploading a small amount of feature codes over the network is feasible. However, downloading or uploading a large number of feature codes from/to end devices uses more bandwidth and increases the signal-transmission delay.

In the current DFC architecture implementation, once a customer call has been made, the DFC router will create a usage, which is a dynamic configuration of feature boxes and internal calls. Downloading feature boxes to the end device definitely increases call set-up delay, especially for a large number of feature codes. (In 1998, AT&T claimed a call set-up time in the PSTN networks of less than two seconds for toll calls, and 2.5 seconds for calls requiring database lookup [1]. Call set-up delay of Internet telephony is expected to be less than that of PSTN networks in that Internet telephony signalling uses the same high-speed backbone as that used for data, while most PSTN systems are still connected by 64kb/s links. In most cases, end users hope that the call set-up delay can be reduced if possible.) For example, Voice Dialling (VD) enables a user to make calls though simple voice commands. Since this feature depends heavily on complicated voice recognition technologies, more code may be required to implement it. Delivering this executable code to intelligent end devices may introduce more delays and cost more bandwidth. In this case, it is better not to deliver executable code for this feature.

In addition, some end users may subscribe to many features (twenty features, for example) implemented by the service provider in that Internet telephony provides more features than traditional telephone networks. It is better to place these subscribed features, such as the VM feature, in the end devices; however, delivering all of them may cause large delays. Some end devices, such as wireless devices, may not have enough resources to execute all of the subscribed features. In those cases, some subscribed features may have to be executed inside the network.

## 7.4 Security

Security is one of the important factors which affects the deployment of features in Internet telephony. In the open and distributed Internet environment, there are varying levels of trust between network nodes and end users. If the level of trust is low, very specific, structured information should be passed from network nodes to end users, and a very narrowly defined set of controls should be exposed to end users from network servers. This restricts the set of features or services that are executed in the end devices, but provides a greater level of security. Network nodes can be sure that end users cannot perform malicious operations, or cause the nodes themselves to crash.

Low levels of trust also cause end users to place some features in their end devices in order to protect their privacy. For instance, if voice messages are stored in an end device, only persons who can access the end device can retrieve the stored messages. Therefore, some end users will place the VM feature and store messages in their end devices for security reasons.

However, features that are placed in end devices can be accessed by anyone who has an opportunity to do so. For example, if the VM system is placed in an end device, unauthorised users can access it to retrieve and delete stored messages. Placing the CFU feature in an end device also allows unauthorised users to change the forwarded number. Of course, authentication, which proves one's identification to someone else, can be used to limit unauthorised accesses. Only after authenticating themselves can end users access the features which reside in end devices. This means that additional code must be introduced to handle authentication.

In addition, end users might download some executable client features across the network. Firstly, downloading traffic should be allowed to traverse firewalls, which are security mechanisms to protect applications. Secondly, in order to prevent information from being revealed during "downloading", it may have to encrypt some features inside the network and decrypt them upon the receipt of features in end devices. This allows network nodes to disguise data so that an intruder can gain no information from the intercepted data. The receiver, of course, must be able to recover the original data from the disguised data. However, this requires additional codes to encrypt features and decrypt them.

## 7.5 Implementation of Features

Some features can be implemented in several ways due to their particular characteristics. The approach to implementing these features may affect their deployment. It will be better to place a feature within the network if this feature is implemented in one way, but this feature may have to be provided in an end device if it is implemented in another way.

In the DFC architecture, some client features are so simple that there is no need to implement them as modularized feature boxes. For example, since the AD feature only stores a caller address, it can be provided by end devices, even most dumb end devices. It can be fully integrated with the LI boxes and reside completely in end devices. However, if the AD feature is implemented as a feature box, the AD feature box will have to retrieve a target address from its operational data, which is stored inside the network in the current ECLIPSE implementation. In this case, it is better to place the AD feature in the network. The implementation of features plays an important role in their deployment.

The deployment of features usually also affects their implementation. For instance, in the DFC architecture, since the modularized feature boxes bring so many advantages, the OCS feature should be implemented as a feature box if it is placed in the network. Should the OCS feature be placed in an end device, it is better to integrate the OCS feature with the LI interfaces in the current ECLIPSE implementation, since a modularized feature box introduces much signalling overhead and lengthens the signal paths.

## 7.6 Network Service Architectures

The service architecture of the network also affects feature deployment. As mentioned before, both the SIP protocol with the CPL language and the DFC architecture can be regarded as two service architectures of Internet telephony. In the two service architectures, network features have to be provided inside the network. In the SIP protocol, it is better to place most client features in intelligent end devices, but in the current DFC architecture, it is more desirable to deploy them inside the network when they are implemented as feature boxes.

For instance, in the SIP protocol, it is better to place the OCS feature in the end device. However, in the current DFC implementation, it is better to deploy the OCS feature inside the network if the OCS feature is implemented as a feature box. The deployment of client features relies heavily on the service architecture of Internet telephony.

## 7.7 Users' Preferences

Internet telephony allows end users to express their preferences, which includes the location of features. Although it is better to place some client features either in end devices or in the network, end users may, by themselves, choose the locations where their subscribed client features reside, whether in end devices or in the network. For example, a user who subscribes to the PSC feature may choose to store telephone numbers inside the network or in end devices. Users' preferences may be decided by the payment of subscriptions for features, service qualities, convenience and other factors.

The VM feature is one of the most widely used features in telecommunication today. Some end users would like to use an answer machine, which can be considered as a VM feature residing in an end device. Users can retrieve and delete messages stored in end devices without sending requests to the network.

However, some end users would like to subscribe to the VM feature residing in the network. In the traditional telephone network, the advantage of the VM feature residing in the network is that the VM feature can record the caller's messages even when the subscriber is busy. Since messages are stored inside the network, the subscriber has to send requests to retrieve and delete messages over the network.

In addition, in the public telephone network, the subscriber usually has to make a monthly payment for the VM feature subscription. Compared to an answering machine, the VM feature residing in the network may cost more, but provides better service for customers. However, some users may prefer answering machines to keep costs down.

Network features should be provided inside the network as they require the support of the network. Although end users can choose how to deploy client features by themselves, they cannot place network features in end devices according to their preference.

## 7.8 Factors Beyond Technology

There are some factors besides technology that affect the deployment of features [24]. For example, the requirements of the U.S federal law enforcement agencies, under the Communications Assistance for Law Enforcement Act, may cause carriers to enforce calls to traverse a carrier-provided server, as without this server it may be difficult to get a "pen register" recording that shows a suspect' s call patterns.

In addition, telephone companies derive significant revenue from services/features subscriptions. These services/features can be "free" in peer-to-peer protocols, such as the H.323 protocols and the SIP protocol, since these peer-to-peer protocols not only allow the use of servers (proxy and redirect servers in the SIP protocol), but also enable two end devices to connect directly without the help of a third party. However, the Media Gateway Control Protocol (MGCP) requires end devices to be controlled by a network server. In the DFC architecture, a customer call also requires the support of a DFC router (a network server). Telephone companies will try to retain control over "free" services/features in peer-to-peer protocols in order to avoid losing funds.

All of these factors show how hard it is to make a blanket statement about feature deployment. Factors beyond technology often direct the decision for deploying features. However, service providers and carriers should design flexible feature placement to meet future demand.

# Chapter 8 Conclusions and Future Work

In this thesis, the deployment of features is discussed based on the SIP protocol and the DFC architecture respectively, which mainly concerns pure signalling messages. The thesis also proposes a voice mail system based on the DFC architecture. Beyond pure signalling messages, this thesis also presents other issues regarding the deployment of features. This chapter makes some conclusions concerning the issues discussed in this thesis and presents some suggestions for future research.

## 8.1 Conclusions

With the integration of Internet services, Internet telephony enhances existing features and creates a number of new features. Basic features can be divided into client features and network features according to their characteristics. Using the SIP protocol and the DFC architecture as examples, the deployment of features is discussed in this thesis.

In Internet telephony, network features should be provided inside the network in that they need the support of the network. Client feature can be provided inside the network or in intelligent end devices. The thesis compares the client features residing in end devices with those residing inside the network by measuring the number of signalling messages and the cost to resources of network nodes.

The discussion of the deployment of client features assumes that end devices are intelligent and have enough resources to store and execute feature-implementation programs. It is mainly concerned with signalling messages. In the SIP protocol, it is more desirable to place most client features in intelligent end devices. In the DFC architecture, it is better to deploy most client features inside the network. However, since Internet telephony will continue to use a large number of dumb devices connected via gateways, basic versions of most features have to be provided in the SIP-enabled or DFC-enabled network.

In the DFC architecture, the modularized architecture introduces signalling overhead although it brings many advantages. Separating the signalling layer from the media layer also introduces signalling overhead.

The current ECLIPSE implementation needs to be improved as it introduces a great deal of signalling overhead. If the DFC implementation allows feature boxes to be executed in end devices, it will be better to allow the operational data to be stored in end devices, which reduces the communication overhead. If an SFB feature box is introduced to deal with the feature boxes that concern only the signalling layer, signalling overhead will be reduced, especially when they are executed in an end device. In addition, end devices should be allowed to control the logical end-to-end media path, in which two media terminals are in the same host.

Compared to the DFC architecture, the SIP protocol makes more use of the strong signal processing abilities of intelligent end devices. However, the SIP protocol does not provide a concrete scheme to handle feature interactions. The SIP protocol does not define header fields or messages to carry information during a session. They have to be introduced in order to execute some features. Generally speaking, the SIP protocol is more distributed while the DFC architecture is more centralized, in which most work is done inside the network.

The thesis also sketches out a voice mail feature based on the DFC architecture. In principle, the proposed voice mail feature can be deployed inside the network or in end devices. This feature consists of three feature boxes (FBs): the Receive FB, the Retrieve FB and the RecordGreet FB. All three FBs are implemented as free feature boxes in target zones. The desirable order of FBs related to some busy treatment FBs is the Receive FB, the CFB feature box and the CW feature box, in which the CW feature box is the nearest FB to the LI boxes.

In addition, this thesis explores other issues beyond signalling messages which are related to the deployment of features. The characteristics of the end devices, delivery of feature code, users' preference, security, the characteristics and implementation of features, network service architectures and feature interactions all should be considered in the deployment of features. Considerations beyond technology also play an important role in placing features.

## 8.2 Future Work

The current DFC implementation needs to be improved. This can be done by allowing operational data to be stored in end devices, by introducing a new kind of feature box to deal with the features that are only concerned with the signalling layer, and by allowing end devices to control the logical end-to-end media path in which two media terminals are in the same host. These approaches need to be implemented in the future. Furthermore, further research is needed to improve the current DFC implementation.

This thesis only discusses the deployment of some sample features. More research is needed to study the deployment of more features and other Internet-based services in Internet telephony.

This thesis sketches out the design of the VM feature. Implementation issues related to the VM feature have been left to be addressed in future research.

The design and implementation of more features – such as home phone feature and click to dial feature (new interesting features in Internet telephony) – in the DFC architecture and in the SIP protocol should also be examined further.

Feature interactions also require further research. In the DFC architecture, the precedence of feature boxes plays an important role on how features interact. More investigation is needed to determine the desired precedence of feature boxes.

# Appendix A  Descriptions of Features

### 1. 8xx Toll-Free Service

This feature allows a commercial subscriber to pay for all incoming calls.

### 2. 900 Information Service

This feature allows callers to pay back part of the call cost to the called party, considered as an added value service provider.

### 3. Abbreviated Dialling

This feature allows the definition of short (e.g., two digit) digit sequences to represent the actual dialling digit sequence for a public or private numbering scheme.

### 4. Account Code Detailed Billing

This feature separate business calls from personal calls by dial codes. At the end of every month, the bill will identify every outgoing call by its code.

### 5. Alternative Transfer

This feature makes use of a "make-before-break" approach. The party initiating the transfer asks the other original call party to call the transfer destination.

### 6. Anonymous Call Rejection

This feature allows a user to reject anonymous calls.

### 7. Attendant

This feature allows Virtual Private Network (VPN) users to access an attendant (operator) position within the VPN for providing VPN service information (e.g, VPN numbers) by dialling a special access code.

### 8. Authentication

This feature allows verification that a user is allowed to access certain options in the network.

### 9. Authorization code

This feature allows a user (typically in a VPN) to override the restrictions placed on the system from which calls are made.

### 10. Auto Busy Redial

This feature allows subscribers to automatically have their telephone redial a busy outside number up to a certain number of attempts.

### 11. Auto-Dial

This feature allows subscribers to dial a number that they call often with the push of one button.

### 12. Automated Attendant

This feature offers 24-hour coverage in answering incoming calls and directing them with a menu of dialling options - no need to worry about breaks, lunch, and after hour coverage.

### 13. Automatic Alternative Billing

This feature allows a user to call another user and ask him to receive the call at his expense.

### 14. Automatic Call Back

This feature allows the called party to automatically call back the calling party of the last call directed to the called party.

### 15. Automatic Calling Distribution

This feature queues calls to wait for the next available agent. Thus, incoming calls are distributed evenly, maximising productivity, and inbound calls are handled orderly and efficiently.

### 16. Automatic Hold Recall

This feature emits a warning tone after a call has been left on hold for a specific period.

### 17. Battery Backup and Power Failure Transfer

This feature keeps the system fully functional in the event of commercial AC power failure.

### 18. Call Answer

This feature automatically takes the caller's messages when the called party is unavailable or on another call.

### 19. Call Answer Plus Notification

This feature automatically takes the caller's messages and send its subscriber a notification when the called party is unavailable or on another call.

### 20. Call Answer plus Pager Notification

This feature automatically takes the caller's messages and sends its subscriber a pager notification when the called party is unavailable or on another call.

### 21. Call Blocking

This feature allows users to block some incoming calls.

### 22. Call Display

This feature can display the caller's name and telephone number.

### 23. Call Distribution

This feature allows a subscriber to have incoming calls routed to different destinations, according to an allocation law which may be real-time managed by the subscriber.

### 24. Call Forwarding

This feature allows the user to have his incoming calls sent to another number no matter what the called party line status may be.

### 25. Call Forwarding No Device

This feature forwards the incoming call to another number if no such device is registered for the destination address given.

### 26. Call Forwarding Remote Access

This feature allows users to turn the Call Forwarding feature "on" or "off" even when they are away from home

### 27. Call Forwarding No Answer

This feature allows the user to have his incoming calls addressed to another number if the called party does not answer within X seconds or Y rings.

### 28. Call Forwarding On Busy

This feature allows the user to have his incoming calls addressed to another number if the called party is busy.

### 29. Call Forwarding Unconditionally

This feature allows the user to have his incoming calls addressed to another number unconditionally.

### 30. Call Gapping

This feature allows the service provider to restrict the number of calls to a served user to prevent congestion of the network.

### 31. Call Hold

This feature allows a subscriber to place a call on hold while taking another call or performing another task

### 32. Call Hold with Announcement

This feature allows a subscriber to place a call on hold with an option to play music or customised announcements to the party on hold.

### 33. Call Hunt

This feature provides consistent coverage in the subscriber's group without forwarding his phone when he leaves the desk. With a Call Hunt group, an incoming call bounces to another phone in his group if it isn't answered by the dialled number.

### 34. Call Limiter

This feature allows a served user to specify the maximum number of simultaneous calls to a served user' s destination. If the destination is busy, the call may be routed to an alternative destination.

### 35. Call Logging

This feature allows for a record to be prepared each time that a call is received to a specified telephone number.

### 36. Call Me Card

This feature allows calls made using the card to be billed to you, not the cardholder.

### 37. Call Name Display

This feature displays the caller's name.

### 38. Call Number Display

This feature displays the caller's telephone number.

### 39. Call Park

This feature allows users to place a call on hold so another party can pick it up at another telephone.

### 40. Call Pickup

This feature enables a user to associate a call request to an already alerting call. The alerting call awaits answer while the user originating call pick-up signals to the network a desire to connect to the alerting call. The network then connects the call parties.

### 41. Call Prompter

This feature greets the caller with a recorded message

### 42. Call Queuing

This feature allows calls that would otherwise be declared busy to be placed in a queue and connected as soon as a free condition is detected. Upon entering the queue, the caller hears an initial announcement informing the caller the call will be answered when a line is available.

### 43. Call Re-routing Distribution

This feature permits the subscriber to have any incoming calls encountering a triggering condition (busy, specified number of rings, queue overload or call limiter) re-routed according to a predefined choice. The calls may be re-routed to another destination number (including pager or vocal box), re-routed on a standard or customised announcement, or queued.

### 44. Call Return

This feature lets users know the number of their last caller even when they cannot answer the phone

### 45. Call Trace

This feature allows users to automatically trace the last incoming call they received. A successful trace receives a success message.

### 46. Call Waiting

This feature allows a subscriber to receive a notification that another party is trying to reach his number while he is talking to another calling party.

### 47. Caller ID Blocking

This feature allows a subscriber to block incoming calls made by a specific ID.

### 48. Caller ID Display

This feature can display the caller's ID.

### 49. Caller Selection

This feature allows the call initiator to choose whom to talk to in the case where multiple parties answer a call.

### 50. Calling Card

This feature allows subscribers to place calls from any normal access interface to any destination number and have the cost of those calls charged to the account specified by this card number.

### 51. Calling Name Delivery

This feature gives to the network operator the capability to display/announce the name of the calling party to the calling name delivery user (the called party) prior to answer, thus allowing this user to screen or distinctively answer the call.

### 52. Calling Number Delivery

This service feature gives the network operator the ability to display/announce the number of the calling party to the calling number delivery user (the called party) prior to answer, thus allowing this user to screen or distinctively answer the call.

### 53. Camp-On

This feature allows a caller who reaches a busy destination to continue to re-try that destination periodically until the line becomes free.

### 54. Cellular Bill

This feature bills the users for airtime on a cellular phone.

### 55. Charge Call

This feature allows a caller to be automatically charged on a different telephone number.

### 56. Click to Dial

Click to Dial is based on Web browsing technologies and allows a subscriber to initiate telephone calls by clicking on a Web page icon or button.

### 57. Client Billing Allocation --Lawyer's Office

When a call comes in, the calling address is correlated with the corresponding client, and the client's name, address and the time of the call logged. If no corresponding client is found, the call is forwarded to the lawyer' secretary.

### 58. Closed User Group

This feature allows the user to be a member of a set of VPN users who are normally authorised to make and receive calls only within the group.

### 59. Completion of Calls to Busy Subscriber

This feature allows a calling user encountering a busy destination to be informed when the busy destination becomes free, without having to make a new call attempt.

### 60. Conference Calling

This feature allows the connection of multiple parties in a single connection. The number of parties connected simultaneously will vary, based on bridging requirements.

### 61. Conference Calling Add-on

This service allows the user to reserve a conference resource for making a multi-party call, indicating the date, time, and conference duration. Once the conference is active, the user controls the conference, and may add, drop, isolate, reattach or split parties.

### 62. Conference Calling Meet-me

This feature allows the user to reserve a conference resource for making a multi-party call, indicating the date, time, and conference duration. At a particular time, each participant in the conference has to dial a special number which has been attached to the booked conference, in order to access the conference bridge.

### 63. Consultation Calling

This feature allows a subscriber to place a call on hold, in order to initiate a new call for consultation.

### 64. Credit Card Calling

This feature allows subscribers to place calls from any normal access interface to any destination number and have the cost of those calls charged to the account specified by the CCC number.

### 65. Customer Profile Management

This feature allows the subscriber to real-time manage his service profile, i.e. terminating destinations, announcements to be played, call distribution, and so on.

### 66. Customer Recorded Announcement

This feature allows a call to be completed to a (customised) terminating announcement instead of a subscriber line. The served user may define different announcements for unsuccessful call completions due to different reasons (e.g. caller outside business hours, all lines are busy).

### 67. Customised Ringing

This feature allows the subscriber to allocate a distinctive ringing to a list of calling parties.

### 68. Data Port Adapter

This feature allows the user to connect a modern or facsimile machine.

### 69. Destination Call Routing

This feature allows customers to specify the routing of their calls to destinations according to the time of day, day of the week, etc.; the area of call origination; the calling line identity of customer; the service attributes held against the customer; priority (e.g. from input of a PIN); the charge rates applicable for destination; and proportional routing of traffic.

### 70. Detail Billing

This feature provides users with a complete listing of all calls placed and received during a monthly billing cycle.

### 71. Device Mobility

This feature concerns the physical location of the device.

### 72. Digital DATA to GO

This feature turns the digital mobile phone into a wireless modern.

### 73. Directory Link

When a subscriber dials a directory assistance number, this feature tries to recognize the number by directory assistance and offers to connect the customer to that number directly.

### 74. Distinctive Ringing

This feature means the phone can be programmed to use a different ring tone or melody, depending on who is calling.

### 75. Do not Disturb

This feature allows the user to press some buttons that prevent intercom calls from reaching end devices and silences the bell.

### 76. Easy Voice

This feature allows a person to speak a phase, such as “call mother”, which is recognized and used as a key to a customized directory.

### 77. Email Access

This feature enables the user to send and receive e-mail message on the screenphone.

### 78. Email Mailing List

This feature makes many outgoing calls, either sequentially or in parallel, to send the same email to every member of a list.

### 79. Email Relay

This feature allows the user to forward a copy of his e-mail directly to his digital mobile phone.

### 80. Emergency Break-In

This feature allows emergency service officials to break into an existing conversion.

### 81. Emergency Call Services

Emergency calls are handled specially by the network. An end user cannot hang up his emergency call and only an emergency operator can hang up before the line is cleared. The emergency call may need to insert some information, such as verifiable subscriber street address data, and will be routed to an appropriate emergency service point, such as a local fire department.

### 82. Follow-me Diversion

With this feature, a user may register for incoming calls to any terminal access.

### 83. Group Calling

This feature can make a multi-party call from mobile terminals.

### 84. Group Ringing

This feature makes an incoming call ring more than one phone. The phone that is first picked up is connected to the calling party. The remaining phones stop ringing.

### 85. Home Phone

This feature is used in a standard residential phone service. When someone calls a particular number, all phones in the home ring. After a user picks up one of phones, all the other phones stop ringing. A user can pick up from any other telephone in home and join an existing call. On the other hand, there may be multiple lines in the home so that a user on another line can initiate a new call, while one or more are currently in progress. All users involved in a single call are all essentially involved in a multiparty conference, and are thus able to hear each other.

### 86. IDC with Forwarding

This feature can identity a call and forward this call to another number.

### 87. Identity A Call

This feature allows the user to identify a call.

### 88. IN Teen Line

This feature restricts outgoing calls based on time of day.

### 89. Incoming Call Screening

This feature allows a called party to reject calls from certain callers automatically.

### 90. Info Service

This feature delivers some information – such as sport or stocks – to mobile phones.

### 91. Information Address

A company advertises a general "information" address for prospective customers. When a call comes in to this address, customers can get information on the company.

### 92. Instant Message Call Waiting

This feature notifies the called party via an instant message instead of a call waiting notification (a call waiting tone) when the called party is busy and receives a new call.

### 93. Integrated Message Centre

This feature routes all unanswered calls into one mailbox although the user has two separate phone numbers for mobile and home/office. Therefore, a user with multiple phone numbers can store his voicemail and faxes in one convenient place.

### 94. Intelligent User Location

When a call comes in, the list of locations where the user has registered should be consulted. Depending on the type of call, the call should ring at an appropriate subset of the registered locations, depending on information in the registry. If the user picks up from more than one phone, the pick-up should be reported back separately to the calling party.

### 95. Intelligent User Location with Media Knowledge

When a call comes in, the call should be proxied to the phone that the user has registered, the media capabilities of which best match those specified in the call request. If the user does not pick up from that phone within X rings, the call should be proxied to the other phones which the user has registered, sequentially, in order of decreasing closeness of match.

### 96. Intercept Recording

This feature notifies callers of the called party's new phone number when it has been changed.

### 97. Last Number Redial

This feature redials the phone number of the last outgoing call.

### 98. Least Cost Routing

This feature directs each outgoing call using the least costly line or long distance carrier connected to the system, reducing the cost of long distance calls.

### 99. Location Hiding

This feature hides the caller's location.

### 100.Malicious Call Identification

This feature allows the subscriber to control the logging (making a record) of calls received that are of a malicious nature.

### 101. Mass Calling

This feature involves instantaneous, high-volume traffic which is routed to one or multiple destinations. Calls can be routed to these destination numbers based on various conditions, such as the geographical location or time of day.

### 102.Meet-me Conference

This feature allows the user to reserve a conference resource for a multi-party call. At a specified date and time, each participant in the conference has to dial a designated number in order to have access to the conference.

### 103.Message Center

This feature takes and stores a caller's messages when the mobile user is on another call, away from the mobile, out of the coverage area or when the mobile is turned off.

### 104.Message Centre Express

This feature takes messages when a user is away from his mobile phone or the phone is turned off.

### 105.Message Centre Fax

This feature allows users to receive, print and forwards faxes to/from any facsimile machine in his mailbox. Users can also retrieve faxes stored in his mailbox.

### 106.Message Centre Paging

This feature signals and displays the user's mobile number across his pager to let a user know that he has a message/fax waiting in his mailbox.

### 107.Message Delivery Services

After a caller has recorded his messages in a virtual temporary mailbox, this feature will then attempt to deliver the messages by calling the callee's phone number.

### 108.Mobile Browser

This feature allows the user to turn the digital mobile terminal into a Web browser.

## 109.Multiple Appearance Directory Number

With a multiple appearance directory, the subscriber's number can ring at numerous locations anywhere on the same PABX.

## 110.Multi-way calling

This feature allows the user to establish multiple, simultaneous telephone calls with other parties.

## 111.Music On Hold

This feature allows a subscriber to place a call on hold with music played to the party on hold.

## 112.Off-net Access

This feature allows a VPN user to access his or her VPN from any non-VPN phone in the PSTN by using a personal identification number (PIN).

## 113.Off-net Calling

This feature allows the user to call outside the VPN network.

## 114.One Number

This feature allows a subscriber with two or more terminating lines in any number of locations to have a single telephone number. This allows businesses to advertise just one telephone number throughout their market area and to maintain their operations in different locations to maximize efficiency. The subscriber can specify which calls are to be terminated on which terminating lines are based in the area in which the calls originate.

## 115.Operator-Assisted Transfer

In this feature, the transferring user wants to confer with the transfer recipient to confirm that the transfer of the caller is acceptable.

### 116.Origin Dependent Routing

This feature enables the subscriber to accept or reject a call, and in case of acceptance, to route this call, according to the calling party geographical location. This service feature allows the served user to specify the destination installations according to the geographical area from which the call was originated.

### 117.Originating Call Screening

This feature allows the subscriber to authorise outgoing calls through the user of a screening list. This list may be managed by the subscriber. The user may override the restriction by giving a PIN.

### 118.Originating User Prompter

This feature allows a served user to provide an announcement which will request the caller to enter a digit or series of digits via a DTMF phone or generator. The collected digits will provide additional information that can be used for direct routing or as a security check during call processing.

### 119.Per Second Billing

This feature bills calls by the second.

### 120. Personal Mobility

This feature concerns the movement of a person from the vicinity of one device to the vicinity of another.

### 121. Personal Numbering

This feature supports a Universal Personal Telecommunication (UPT) number that uniquely identifies each UPT user and is used by the caller to reach that UPT user.

### 122. Personal Speed Calling

This feature allows a subscriber to store some telephone numbers. These speed call numbers are programmed and used by individuals at their own telephones.

### 123. Premium Charging

This feature allows for part of the cost of a call to be paid back to the called party.

### 124. Premium Rate

This service allows the pay-back of part of the call cost to the called party, considered as an added value service provider.

### 125. Privacy

This feature prevents other users from picking up a line in use.

### 126. Private Numbering Plan

This feature allows the subscriber to maintain a numbering plan within his private network, which is separate from the public numbering plan.

### 127. Remote Maintenance

A central maintenance centre calls into the system through a modem for remote trouble diagnosis.

### 128. Request Forking

This feature allows an Internet telephony proxy server P to attempt to locate a user by forwarding a request to multiple destinations, A and B. The call will be connected to the first destination to pick up, and call attempt to the others will be cancelled.

### 129. Reverse Charging

This feature allows the service subscriber (e.g. freephone) to accept received calls at its expense and be charged for the entire cost of the call.

### 130. Security Screening

This capability allows security screening to be performed in the network before an end-user gains access to the subscriber’ s network, systems, or applications. Access code abuse detection is a capability which will generate a report on the invalid access attempts: how many, over what time period, by whom, and from where.

### 131. Selective Call Acceptance and Rejection

This feature allows users to receive only the calls they really want to get and intercepts those calls they don't want.

### 132. Selective Call Forwarding

This feature permits the user to have his incoming calls addressed to another number, no matter what the called party line status is, if the calling line identity is included in, or excluded from, a screening list. The user’ s originating service is unaffected.

### 133. Selective Call Forwarding on Busy

This feature permits the  user to have his incoming calls addressed to another number, if the called user does not answer within X seconds or Y rings and the calling line identity is included in, or excluded from, a screening list. The user’ s originating service is unaffected.

### 134. Selective Call Forwarding on NO Answer Busy

This feature permits the user to have his incoming calls addressed to another number, if the called user is busy and the calling line identity is included in, or excluded from, a screening list. The user’ s originating service is unaffected.

### 135. Spontaneous Message on Busy

This feature allows the caller to leave a message in a mailbox if the callee line is busy. This feature will deliver the caller's stored message to the callee later.

### 136. Split Billing

This feature allows for the separation of charges for a specific call, the calling and called party each being charged for one part of the call.

### 137. Station Restriction

This feature provides different classes of service for restricting long-distance calls.

### 138. Televoting

Televoting enables subscribers to survey public opinion using the telephone network. Persons wishing to respond to an opinion poll can call advertised televoting numbers to register their votes. The charging is at the discretion of the service subscriber.

### 139. Terminating Call Screening

This feature allows the user to screen calls based on the terminating telephone number dialled.

### 140. Text Messaging

This feature allows the user to receive and view text message across the display screen of the phone.

### 141. Three Way Calling

This feature allows the subscriber to add additional parties to an existing call

### 142. Time Dependent Routing

This feature allows the served user to apply different call treatments based on the time of day, day of week, day of year, holiday, etc.

### 143. Toll Restriction

This feature limits toll calls made by designated users to only those toll calls that are necessary for them to carry out their job responsibilities.

### 144. Universal Access Number

This feature allows a subscriber with several terminating lines in any number of locations or zones to be reached with a unique directory number. The subscriber may specify which incoming calls are to be routed to which terminating lines, based upon the area in which the call originated.

### 145. Universal Personal Telecommunications

This feature provides personal mobility by enabling a user to initiate any type of service and receive any type of call on the basis of a unique and personal network independent number, across multiple networks, at any user-network access (fixed, movable or mobile), irrespective of geographic location, limited only by terminal and network capabilities.

### 146. User-defined Routing

This capability allows the subscriber to specify how outgoing calls, from the subscriber's location, should be routed, either through private, public, or virtual facilities or a mix of facilities, according to the subscriber's routing preferences list. These lists will typically apply to individual lines or to several lines at the subscriber's location.

### 147. Virtual Private Network

This service permits the subscriber to build a private network by using the public network resources. The subscriber's lines, connected on different network switches, constitute a virtual PABX, including a Number of PABX capabilities, such as private numbering plans (PNP), call transfer, call hold, and so on.

### 148. Visual Call Waiting

This feature allows a subscriber to receive a notification that another party is trying to reach his number and displays the name and number of this party on his screen-phone while he is talking to another calling party.

### 149. Visual Caller ID

This feature allows a subscriber to display the name and number of this party on his screen-phone.

### 150.Voice Dialling

This feature enables the user to make calls though simple voice commands

### 151. Voice Mail

This feature automatically takes the caller's messages when the called party is unavailable or on another call.

### 152. Wake-up Call Service

This feature allows the subscriber to set a wake-up call for a desired wake-up time.

# Appendix B References

1. AT&T, “AT&T sets the industry standard for network reliability”, Mar.1998, http://www.att.com/network/standrd.html.
2. Bellcore, “Advanced Intelligent Network (AIN) Release 1, Switching Systems Generic Requirements”, Bellcore Technical Advisory TANWT-001123, May 1991.
3. International Telecommunication Union, "General recommendations on telephone switching and signalling intelligent network: Introduction to intelligent network capability set 1", Recommendation Q.1211, Telecommunication Standardization of ITU, Switzerland, Mar. 1993.
4. International Telecommunication Union, "Introduction to intelligent network capability set 2", Recommendation Q.1221, Telecommunication Standardization of ITU, Switzerland, Sept. 1997.
5. International Telecommunication Union, “Principles of Intelligent Architecture”, Recommendation Q.1201, Telecommunication Standardization of ITU, Geneva, 1992.
6. International Telecommunication Union, “Principles of Telecommunication Services Supported by an ISDN and the Means to Describe them”, Recommendation I.210, Telecommunication Standardization of ITU, Geneva, 1989
7. International Telecommunication Union, “Vocabulary of Terms for ISDNs”,

Recommendation I.112, Telecommunication Standardization of ITU, Geneva, 1989

8. Greg Bond, Eric Cheung, Michael Jackson, Hal Prudy, Chris Ramming and Pamela Zave, "Investigations of A New Service Architecture for The Next Generation of Telecommunications", http://www.research.att.com/~pamela/dfc. html.

9. Gregory Bond, Eric Cheung, Andrew Forrest, Michael Jackson, Hal Purdy, Chris Ramming, and Pamela Zave, "DFC as the basis for ECLIPSE, an IP communications software platform", Proc.of the IP Telecom Services Workshop 2000, Atlanta, U.S.A, pp19-25, Sept. 2000.

10. Gregory Bond, Franjo Ivancic, Nils Klarlund, Richard Trefler, "ECLIPSE Feature Logic Analysis", http://www.fokus.gmd.de/events/iptel2001/pg/final_program/.

11. Lynne Blair and Jianxiong Pang, "Feature Interactions-Life Beyond Traditional Telephony", Feature Interactions in Telecommunications and Software Systems VI., IOS Press, pp83-93, Amsterdam, 2000.

12. T.F Bowen, F.S. Dworack, C.H.Chow, N. Griffeth, G.E. Herman and Y.J Lin, "The Feature Interaction Problem in Telecommunications Systems", Proc. Seventh Int'l Conf, Software Eng. for Telecommunications Switching System, London, pp59-62, July1989

13. Ismail Dalgic and Hanlin Fang, "Comparison of H.323 and SIP for IP Telephony Signalling", Proc. Of Photonics East, Boston, Massachusetts, Sept. 1999.

14. Roch H. Glitho, “Advanced Services Architectures for Internet Telephony: A critical Overview”, IEEE Network, vol.14, no.4, pp38-44, July-August 2000

15. M.Handley, H. Schularinne, E. Schooler and J. Roseberg, "SIP: Session Initiation

Protocol", Request for Comments 2543, Internet Engineering Task Force, Mar. 1999.

16. Mahbub Hassan, Alfandika Nayandoro and Mohammed Atiquzzaman, “Internet Telephony: Services, Technical Challenges, and Products”, IEEE Communications Magazine, vol.38, no.4, pp96-103, April 2000.
17. Michael Jackson and Pamela Zave, "Distributed Feature Composition: A Virtual Architecture for Telecommunication Services", IEEE Transactions on Software Engineering, vol.24, no.10, pp813-847, Oct.1998
18. Michael Jackson and Pamela Zave, "Media implementation in ECLIPSE I: Principles of signalling/media separation", AT&T Technical Memorandum number: HA1640000-000808-9TM
19. Michael Jackson and Pamela Zave, "The DFC Manual", May 2000, http://www. research.att.com/~pamela/dfc.html
20. Alan Johnston, Robert Sparks, Chris Cunningham, Steve Donovan and Kevin Summers, "SIP Service Example", Internet Draft, Internet Engineering Task Force, June 2001, work in progress
21. Dirk O. Keck and Paul J. Kuehn, “The Feature and Service Interaction Problem in Telecommunications Systems: A Survey”, in IEEE Transactions on Software Engineering, vol.24, no.10, 1998
22. Jonathan Lennox and Henning Schulzrinne, "Call Processing Language Framework and Requirement", Request for Comments 2824, Internet Engineering Task Force, May 2000.
23. Jonathan Lennox and Henning Schulzrinne, "Feature Interaction in Internet Telephony," in Proc. of Feature Interaction in Telecommunications and Software

Systems VI, Glasgow, United Kingdom, May 2000.

24. Christos A. Polyzois, K. Hal Purdy, Pine-Fai Yang, David Shrader, Henry Sinnreich, Fancois Menard and Henning Schulzrinne, “From POTS to PANS: A Commentary on the Evolution to Internet Telephony”, IEEE Network, vol.13, no.3, pp58-64, May-July 1999

25. Jonathan Rosenberg, Jonathan Lennox and Henning Schulzrinne, “Programming Internet Telephony service”, IEEE Network, vol.13, no.3, pp42–49, May/June 1999.

26. Jonathan D. Rosenberg and Richard Shockey, "The Session Initiation Protocol (SIP): A Key Component for Internet Telephony", Computer Telephony, vol. 8, issue 6, pp124-139, June 2000

27. Henning Schulzrinne and Jonathan Rosenberg, "Signalling for Internet Telephony", in International Conference on Network Protocols (ICNP), Austin, Texas, Oct. 1998.

28. Henning Schulzrinne and Jonathan Rosenberg, "The Session Initiation Protocol: Providing Advanced Telephony Services Across the Internet," Bell Labs Technical Journal, vol. 3, no. 4, pp. 144-160, Oct.-Dec.1998.

29. Kundan Singh and Henning Schulzrinne, "Unified messaging using SIP and RTSP", in Proceedings of the IP Telecom Services Workshop 2000, pp27-33, Atlanta, U.S. A, Sep.2000.

30. Xiaotao Wu and Henning Schulzrinne, "Where Should Services Reside in Internet Telephony Systems?", in Proceedings of the IP Telecom Services Workshop 2000, pp39-44, Atlanta, U.S.A, Sep.2000.

31. Pamela Zave and Michael Jackson, "DFC Modifications I: Routing Extensions",

AT&T Technical Memorandum HA1640000-990715-10TM.

32. Pamela Zave and Michael Jackson, "DFC Modifications II: Protocol Extension", AT&T Technical Memorandum HA1640000-991119-16.

33. Pamela Zave and Michael Jackson, "New feature interactions in mobile and multimedia telecommunication services", in Feature Interactions in Telecommunications and Software Systems VI, pages 51-66, IOS Press, 2000.

34. Pamela Zave, "An Architecture for Three Challenging Features", http: //www .research.att.com/~pamela/dfc.html.