The Design of the Communication Protocols

NAME: Liu Shiyun
Number: G45472436

May 18, 2012

Abstract

This research article investigates how to design communication protocols in the field of computer network. In order to design a communication software, the protocol is the basic. A communications protocol is a system of digital message formats and rules for exchanging those messages in or between computing systems and in telecommunications. There are six protocol elements: Servive primitive and its timing, PDU and its exchange timing, protocol status, protocol events, protocol variables and protocol actions. This article takes SIP (Session Initiation Protocol) as an example. SIP is a signaling protocol for Internet conferencing, telephony, presence, events notification and instant messaging. It is an application layer protocol. In addition, MSC (Message Sequence Chart) and SDL（Specification and Description Language） are used to describe the protocol and SDL and TTCN Suite 4.5 software is choosen as a tool. This study would like to focus on the MSC diagrams. One purpose of doing this research is to introduce to the student that how to design a protocol and the other is to make readers know how the protocol works in detail and how to describe a protocol by using MSC diagram to help them understand the topic best.

1. Introduction

Scientific and technological development is the major mission of our world now. With the broadly usage of computer, people are becoming more and more depend on computer networks. Large amount of information and massages can be searched and exchanged in a while, so the core competence of the communication software is to help user search and exchange information quickly, exactly and reliably.

In this paper, how to design communication protocols is investigated in the field of computer network. Communication is an exchange of messages between two entities and a communications protocol is a system of digital message formats and rules for exchanging those messages in or between computing systems and in telecommunications. Protocol is the basic of the communication software. So understanding how the protocol works and how to design a protocol is very important.

SIP (Session Initiation Protocol) is choosen as an example to state the topic and SIP is a signaling protocol for Internet conferencing, telephony, presence, events notification and instant messaging. It is an application layer protocol.

One purpose of doing this research is to make the student know how to design a protocol and the other is to show how the protocol works and how to describe a protocol by using MSC diagram in order to help the readers understant well.

The major research questions addressed in this paper are:

1. How to design a protocol well?
2. How to describe a protocol by MSC?

2. Methodology

My work takes qualitative and quantitative research methods. Sampled RA introductions in the software engineering of computer science field are taken into consideration. The goal of this article is to find the patterns and features of English proses and help Chinese students master the English academic writing skills.

2.1. Protocol process

To develop a new protocol, some rules and specfications should be obeyed.

Here are the process of developping a new protocol.

(1) Analyse the protocol enviroments

(2) Design the functions of protocol

(3) Design protocol elements

(4) Create protocol documents

(5) Describe the protocol

(6) Test and verify the protocol.

This article focus on design and describe the protocol.

2.2. The instrument

The work takes SDL and TTCN Suite 4.5 software which runs on the Windows XP system. SDL and TTCN Suite 4.5 was developped by IBM and now update to 6.3 version.

MSC (Message Sequence Chart) and SDL（Specification and Description Language）are choosen to describe the protocols.

MSC is a trace language for the specification and description of the communication behaviour of system components and their environment by means of message interchange. SDL （Specification and Description Language）is a specification language targeted at the unambiguous specification and description of the behaviour of reactive and distributed

systems. And a specification of a system is the description of its required behaviour, while a description of a system is the description of its actual behaviour; that is its implementation.

Fig2.1. and Fig 2.2. show examples of MSC diagram and SDL diagram.

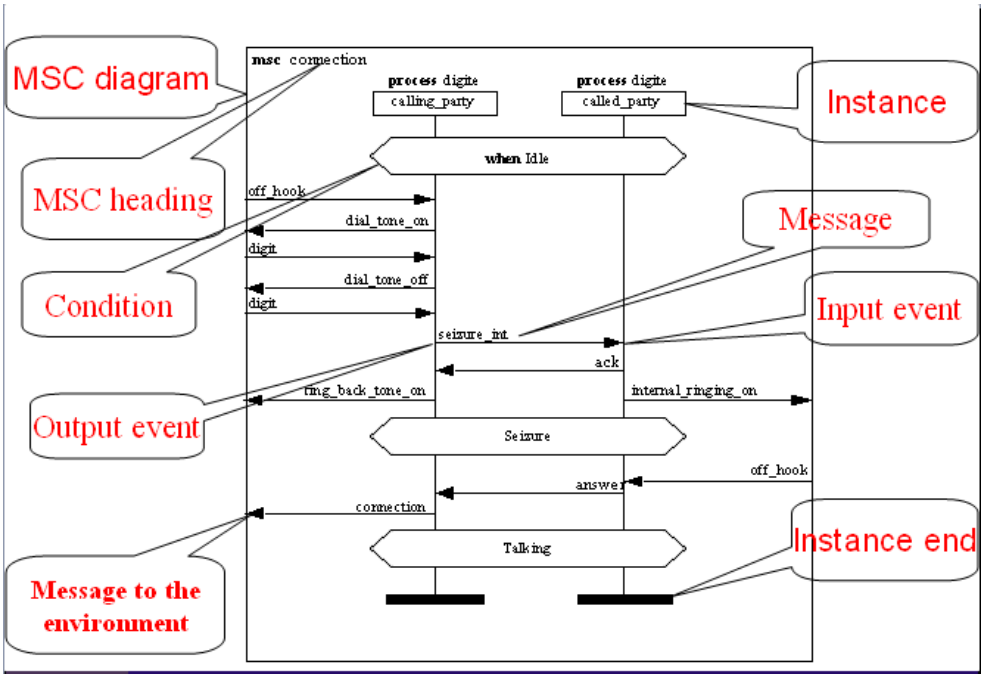

Fig 2.1. MSC Example Diagram

In the MSC diagram, communication entities operate their functions and exchange their messages between each other under some special conditions. The input events happens when receiving massages from others and output event occurs when sending massages to others. And if they finish their operations, the instance will be ended.

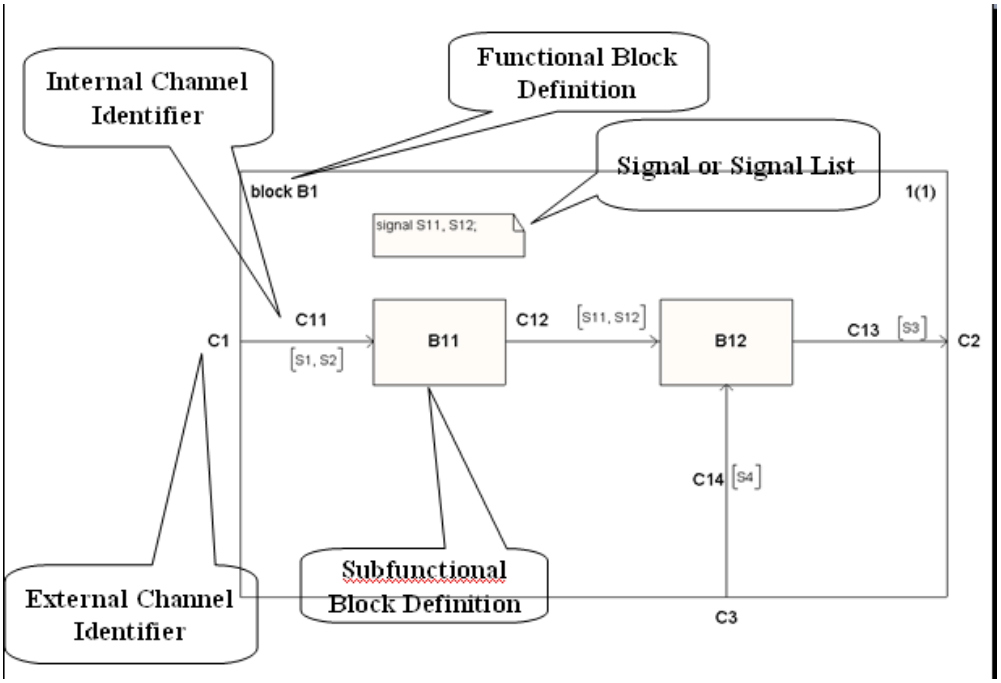

Fig 2.2. SDL Example Diagram

The SDL diagram definites functional blocks which can be seperated into many subfunctional blocks, signals or signal lists which transfer through the functional blocks or subfunctional blocks, and Internal Channel Identifier between subfunctional blocks and External Channel Identifier which come from other functional blocks

This study focus on the MSC diagrams.

There are also a lot of benefits of specification language: A well-defined set of concepts; unambiguous, clear, precise, and concise specifications; a basis for verifying specifications with respect to completeness and correctness; a basis for determining whether or not an implementation conforms to the specifications; basis for determining the consistency of specifications relative to each other; use of computer-based tools to create, maintain, verify, simulate and validate specifications; computer support for generating applications without the need of the traditional coding phase. (IBM Rational SDL Suite 6.3 Getting Started, 4/2009).

And SDL and TTCN Suite 4.5 software meets all of the benefits.

3. Results and discussion

This research chooses the Session Initiation Protocol (SIP) as an example to state how to design a communication protocol.

The Session Initiation Protocol (SIP) is an IETF-defined signaling protocol widely used [citation needed] for controlling communication sessions such as voice and video calls over Internet Protocol (IP). （Session Initiation Protocol - Wikipedia）

SIP is a signaling protocol for Internet conferencing, telephony, presence, events notification and instant messaging. It is an application layer protocol.

3.1 Analyze the communication entities and their functions

In RFC3261, SIP network structure contains four logical enities: User Agent, Proxy, Registrar, Redirect Server.

A SIP user agent (UA) is a logical network end-point used to create or receive SIP messages and thereby manage a SIP session. A SIP UA can perform the role of a User Agent Client (UAC), which sends SIP requests, and the User Agent Server (UAS), which receives the requests and returns a SIP response. These roles of UAC and UAS only last for the duration of a SIP transaction.(Wikipedia)

Proxy is an intermediary entity that acts as both a server (UAS) and a client (UAC) for the purpose of making requests on behalf of other clients.A proxy server primarily plays the role of routing, which means its job is to ensure that a request is sent to another entity "closer" to the targeted user. Proxies are also useful for enforcing policy (for example,

making sure a user is allowed to make a call). A proxy interprets, and, if necessary, rewrites specific parts of a request message before forwarding it.(Wikipedia)

Registrar is a server that accepts REGISTER requests and places the information it receives in those requests into the location service for the domain it handles which registers one or more IP addresses to a certain SIP URI, indicated by the sip: scheme, although other protocol schemes are possible (such as tel:). More than one user agent can register at the same URI, with the result that all registered user agents will receive a call to the SIP URI.(Wikipedia)

Redirect Server is a user agent server that generates 3xx (Redirection) responses to requests it receives, directing the client to contact an alternate set of URIs. The redirect server allows proxy servers to direct SIP session invitations to external domains.(Wikipedia)

But this project simplify the Session Initiation Protocol which only defines only three entities: Caller, Called and Register/Proxy server. Caller and Called are User agents and Register/Proxy server is a combination of Proxy, Registrar, Redirect Server.

Caller is an active end point. It creates a SIP call and invites others to make a session with it.

Called is a passive end point. It will receive a SIP call request if the Caller invites it and will make a decision of whether have a session with the Caller, if the status is idle.

Register/Proxy server is a sip server which is a media between caller and called. It deals with the registeration of user agent, looking for addresses of caller and called and forwarding messages between them.

3.2 Analyze the communication process and scenes

This communication process contains four stages: register stage, create stage, session stage and release stage. And all of the four stages have nomal and abnomal stuations.

3.2.1 Register stage

Before making a sesstion, both of the Caller and Called should register their addresses on the Register/Proxy server, so the server could find their addresses and provide service to them.

3.2.2 Create stage

After registering on the Register/Proxy server, Caller send an invite message with the Called ID to the server to make a request about ceating a connection with Called. Then the Called is ringing and if the Called is in the idle status, it can pick up the phone and return a message of the agreement.

3.2.3 Session stage

If the Called returns the message of agreement successfully, they can make a session and exchange information between each other. This system is a P2P system, so one user agent can only have one session at one time.

3.2.4 Release stage

Either Caller and Called can request to stop the session. Once the server receives the release message, it will help release the connection and stop the session.

3.3 Messages and timers define

Messages and timers need to be define in the protocol.

3.3.1 Messages definition

Register（UAID）： Register request (UAID is the address of the user agents)

200OK： Response of accepting the request

INVITE（UAID）： Caller invite Called to create a connection

180Ring：Called ring

ACK：Acknowledge

BYE：Release request

BusySignal：Called is busy

N0_DestinationAddress：Called hasn't been registered

NotPickUp：Called doesn't pick up the phone before time out

3.3.2 Timers definition

Register timer: 20s

Timer from Caller inviting to Called ringing: 30s

Waiting for Called picking up timer: 80s

Timer from Called ringing to picking up the phone: 60s

Timer from Called picking up the phone to returning acknowledge: 30s

Rlease timer: 30s

3.4 MSC diagrams of each scenes

To design a protocol, both of the normal and abnormal scenes need to be taken into consideration. And there are three entities: Caller, Register/Proxy Server and Called.

3.4.1 Normal scenes

（1）Register:

MSC normalRegister

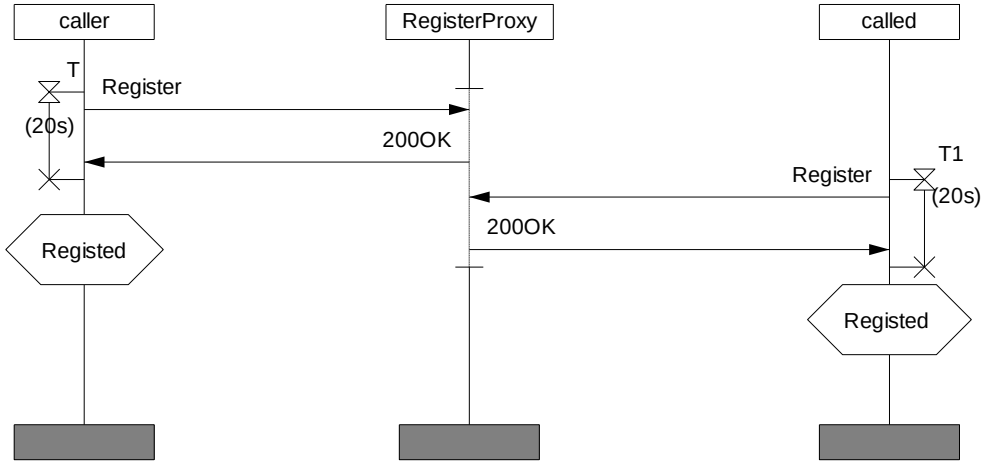

Fig3.1 Normal Register MSC Diagram

In Fig3.1, Caller and Called want to register their addresses on the Register/Proxy server. They send an Register(UAID) messges to the server and if it's successfully, the server will return an 200OK message to the user agent.

（2）Create:

MSC normalCreate

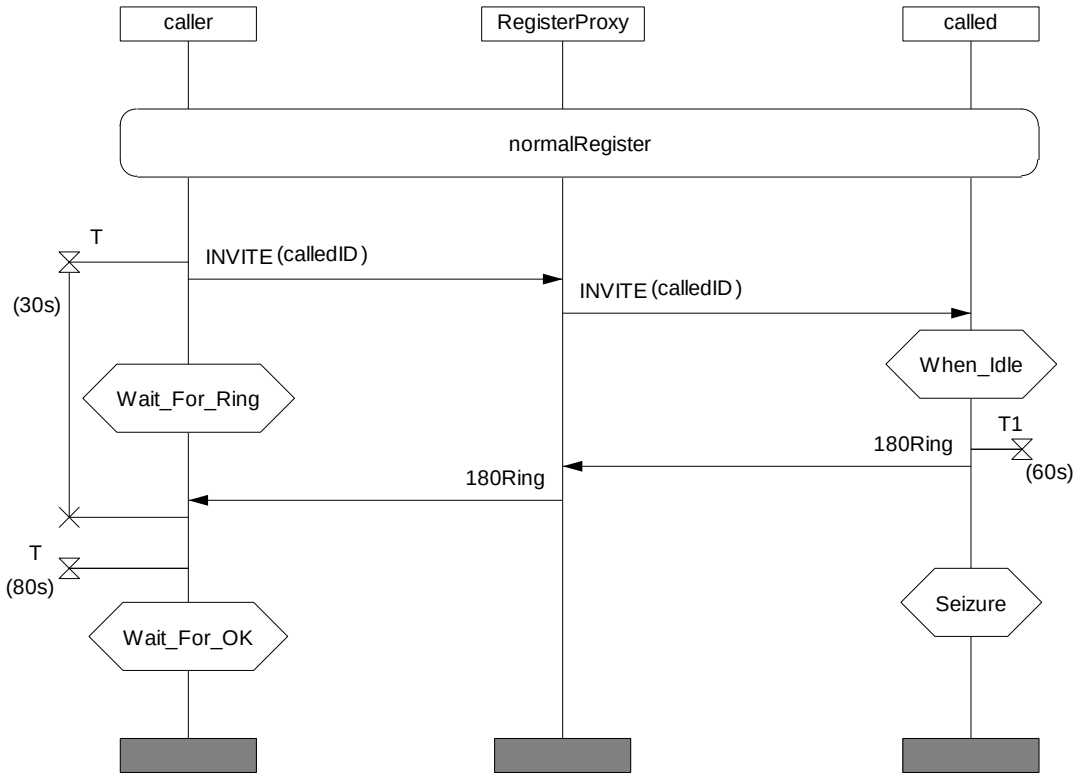

Fig3.2 Normal Create MSC Diagram

In Fig3.2, after registered, Caller wants to create a connection with Called, so it sends and Invite message with the Called address to the server. If the Called is in the idle status, it will ring and return a 180Ring message to make the Caller know the Called is idle.

（3）Session

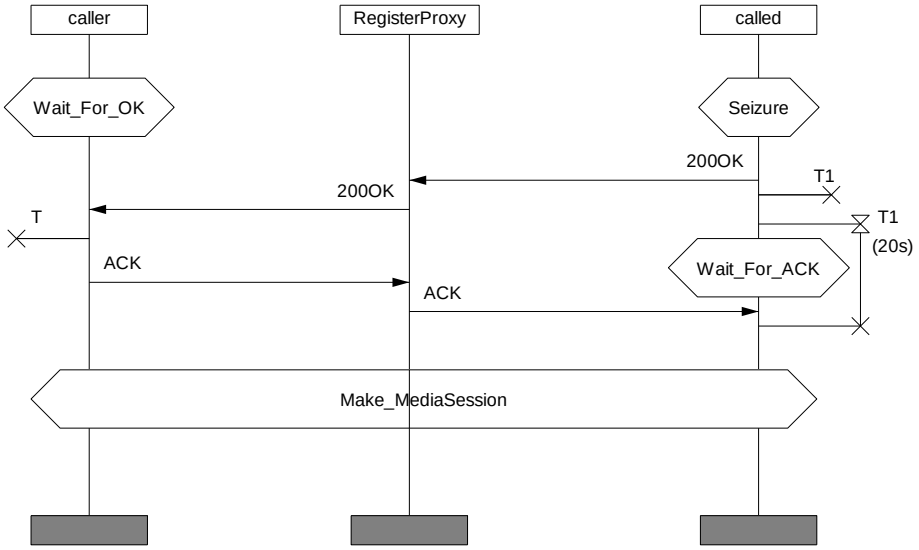

Fig3.3 Normal Session MSC Diagram

In Fig3.3, when ringing, Called can pick up the phone to agree with building a connection with Caller. If Called pick up the phone, a 200OK message will be returned. And after an ACK message coming back from the Caller, they can make a session with each other to exchange and transfer the messages and information.

（4）Release

MSC normalRelease

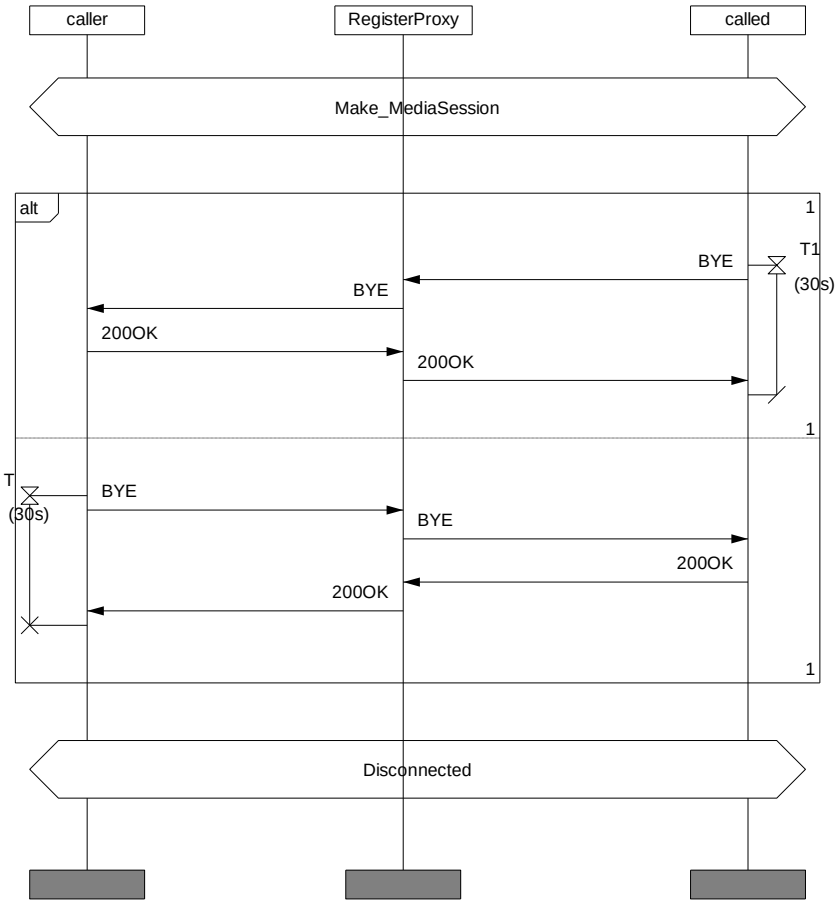

Fig3.4 Normal Release MSC Diagram

In Fig3.4, if the user agents finish their work during the session, they can request to release the connection and stop the session. Either the Caller or Called can send a BYE() message to the server and after the other one agreeing with the request, they will be a diconnected status now.

3.4.1 Abnormal scenes

（1）Register:

MSC registerFail

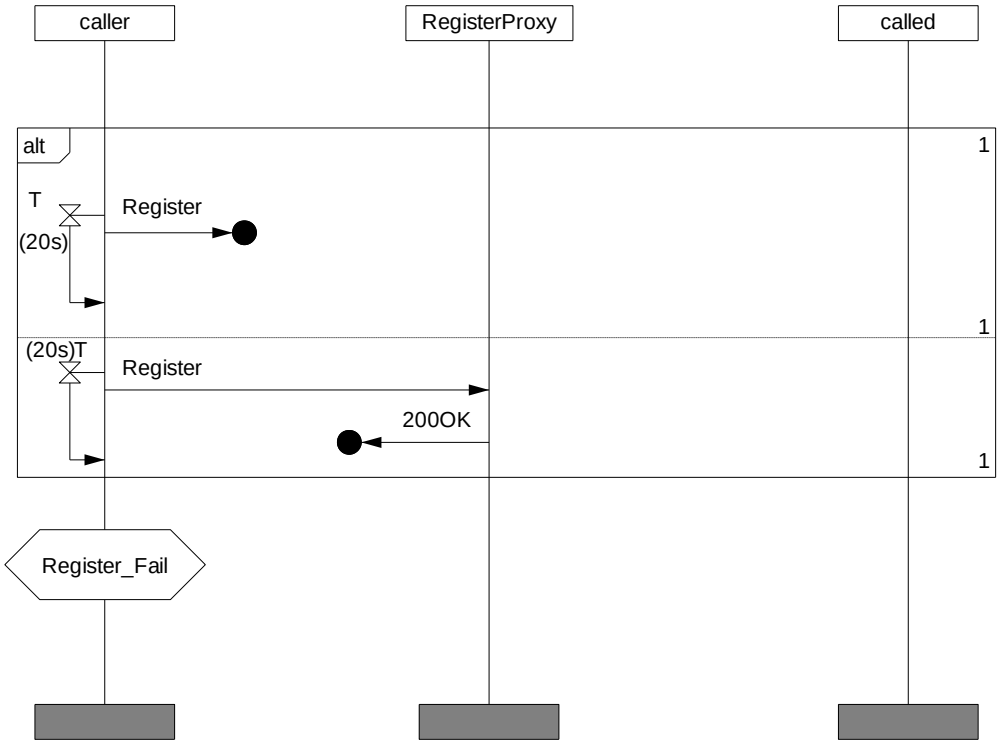

Fig3.5 Caller Registers Unsuccessfully MSC Diagram

When registering the address, there may be some mistakes, such as package lost. In Fig3.5, there are two abnormal scenes: one is the Register message is lost and the other is the 200OK response message is lost. So the register operation of the user agent is failed.

（2）Create:

There are many abnormal scenes occur when creating a connection between Caller and Called. Here is four situations:

<1> Called Busy

MSC calledBusy

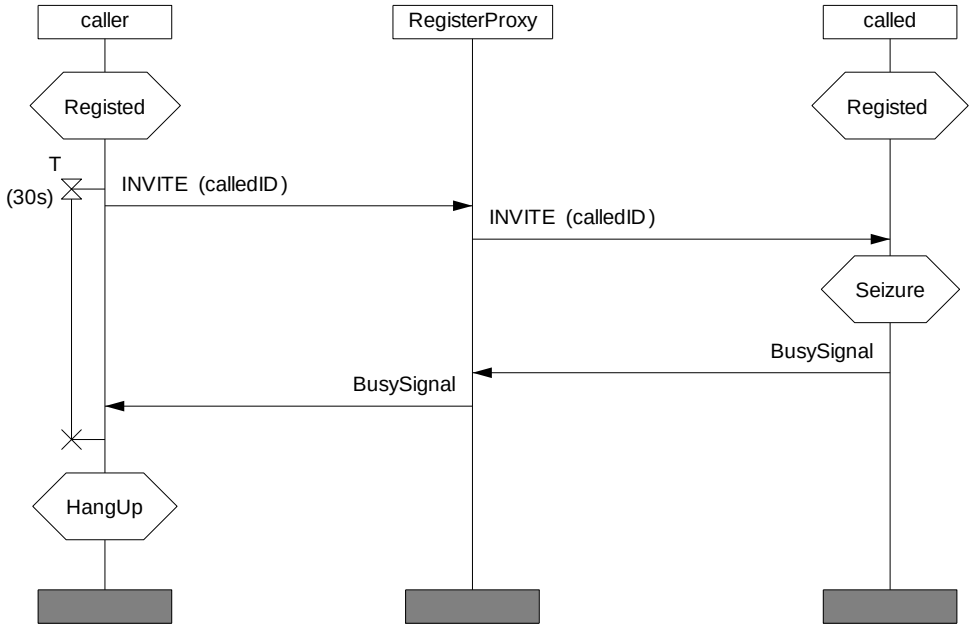

Fig3.6 Called Busy MSC Diagram

This system allows the user agent can only have one session at one time. In Fig3.6, when Caller invites Called to create a connection, but the Called have another one now which means Called it not in the idle status, it will return a BusySignal response to the server and the server forward it to the Caller, the creation of connection is failed.

<2> Called Unregistered

MSC calledUnregistered

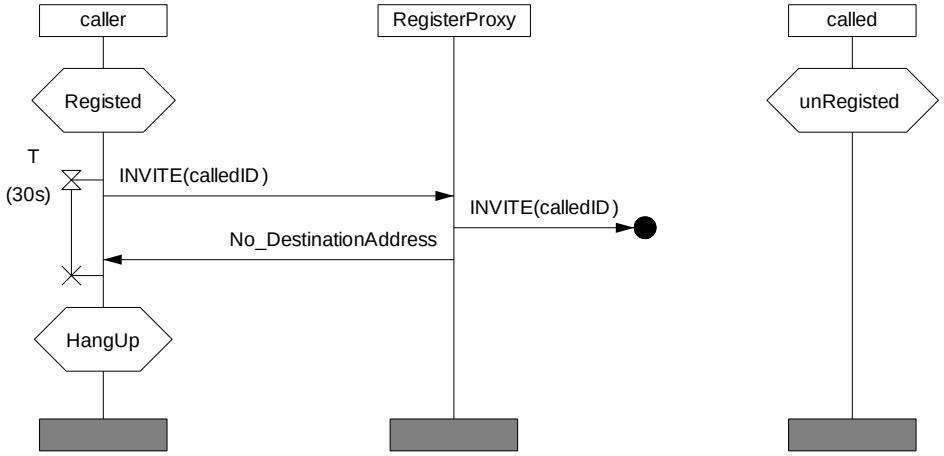

Fig3.7 Called Unregistered MSC Diagram

Before creating a connection between two user agents, both of the Caller and Called should be registered on the server. But if the Called has not been registered, the Register/Proxy server can not find the Called address and the creation of connection is failed (Look at Fig3.7).

<3> Called don't pick up the phone

MSC calledNotPickUp

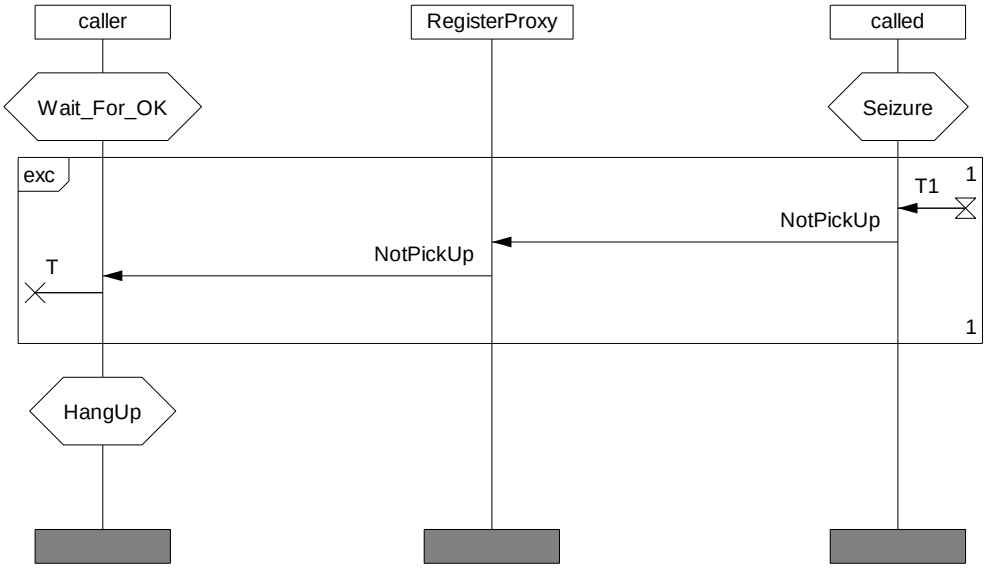

Fig3.8 Called Unpick up MSC Diagram

Before creating a connection between two user agents, both of the Caller and Called should be registered on the server. But if the Called has not been registered, the Register/Proxy server can not find the Called address and the creation of connection is failed.

<4> Caller hang up early

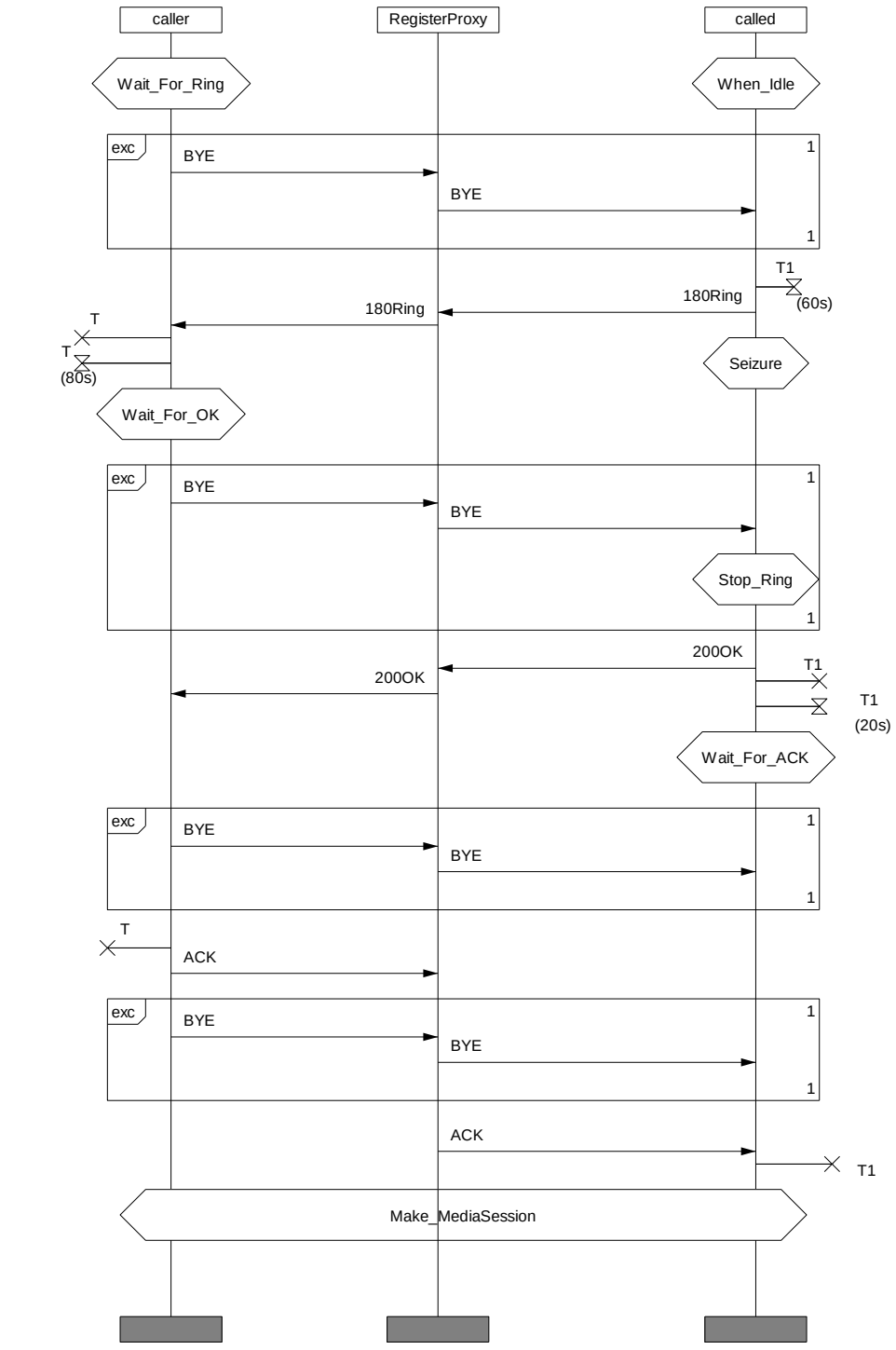

Fig3.9 Caller Hang Up Early MSC Diagram

If the Caller wants to build a connection with Called but then changes its mind immediately before creating connection successfully. In Fig3.9, the Caller sends an Invite massege to request for connection creation, but then hang up the phone before they build the connection. The creating connection operation is failed too.

3. Conclusion

This study talks about how to design a communication protocols and introduces MSC (Message Sequence Chart) and SDL（Specification and Description Language）which can help to describe the protocols and this project focuses on the MSC diagrams by using the SDL and TTCN Suite 4.5 software.

The results of this study is meanful, because it introduces to the student that how to design a protocol and make readers know more about how the protocol works in detail and how to describe a protocol by using MSC diagram to help them understand the topic best.

There is no doubt that something needs to be improved. There may be some other situations and exceptions during communication.

And the SDL can be used to design the functional blocks in the future.

Overall, this study can be conducive to make students know more about protocols design.

Referrence:

IBM Rational SDL Suite 6.3 and IBM Rational TTCN Suite 6.3 group of software developers, SDL Suite Getting Start, IBM Corporation, 2004.

Song Maoqiang, The Basic Design of Cummunication Software (the second version), Beijing University of Posts and Telecommunications Publisher, 2008.

Wikipedia, the free encyclopedia, Internet.