

摘 要

通信协议测试是通过实验的方法发现通信协议实现在功能和性能方面的错误。由于各国都在开展第三代移动通信系统（3G）的研究，通信协议一致性测试趋显其在理论和现实上的重要性。通信协议的一致性测试采用较多的是 FSM（Finite State Machine，有限状态机）模型，但该模型只能刻画协议的控制行为，不包括协议的数据行为，具有很大的局限性。3G 的测试文档对非接入层采用的一致性测试方法是分布式测试法，该方法要使用上、下两个测试器，因此需要上下测试器之间的协调过程。

本文的主要贡献在于：

- 运用 CEFSM（Communication Extended FSM，通信扩展有限状态机）模型，成功的对 TD-PRS 的 LLC（Logical Link Control，逻辑链路控制）模块进行了仿真。
- 同时用基于 CEFSM 模型的算法，实现了 LLC 模块的一致性测试，包括异步平衡模式的建立、释放，证实和非证实数据的传送，XID 的协商等。
- 采用远程测试法，对 3G 测试文档非接入层测试结构进行了简化。

采用本文所述方法，实现了对 TD-PRS 终端 2、3 层的协议仿真和一致性测试，纠正了协议开发中的错误。同时，该方法对其他通信协议的仿真和测试有重要的参考价值和实用意义。

关键词：

第三代移动通信系统，TD-PRS，LLC 模块，CEFSM 模型，协议一致性测试，测试方法

Abstract

The communication Protocol Test is the way to find out the errors of function and performance of communication protocol implementation. Until now, all countries are carrying through the research on the Third Generation Mobile Communication system (3G), followed by the discovery of importance in the study of communication protocol conformance test in theory and realization. Communication protocol conformance test mostly uses the FSM (Finite State Machines) to model real systems, but there is limitation in FSM that it can only depict the control behavior of the protocol, not include the data counterpart. In 3G-test document, distributed test method is used to test the conformance of non-access stratum. This kind of test needs two testers, the upper one and the lower one, implying the requirement of coordinate procedure between the two testers.

The main contributions made in this paper are summarized as follows:

- Successfully simulated the LLC (Logical Link Control) module of TD-PRS with CEFSM model.
- At the same time, conformance test of LLC module is realized using algorithm based on the CEFSM model, including the establishment and release of ABM (Asynchronous Balanced Mode), acknowledged and unacknowledged data transfer; XID negotiation, etc.
- The test structure of the non-access stratum in 3G-test document is simplified by means of remote test.

We complete the protocol simulation and conformance test in layer 2 and 3 of TD-PRS terminal using our proposed scheme in this paper, which rectifies the bugs of the protocol implementation. This scheme is also valuable for reference or utility in the simulation and test of other communication protocols.

Keywords: the third generation mobile communication system, TD-PRS, LLC module, CEFSM model, protocol conformance test, test method

第一章 绪论

1.1 论文背景

在通信系统之间，为了使来自不同厂家的系统能够成功地进行通信，必须有标准化的协议，它规定了通信系统上各部件之间进行通信时应遵守的规则。但标准化的协议并不能确保通信的成功。这是因为通信协议标准基本上是在 RFC(Request For Comment)中以自然语言描述，实现者对于协议的不同理解会导致不同的协议实现，有时甚至会是错误的实现。因此，我们需要一种有效方法对协议实现进行判别，这种方法便是“协议测试”(Protocol Test)。

测试是一个试图通过验证的方法找出错误的过程。随着通信技术的发展，特别是移动通信的迅猛发展，协议测试理论和技术的研究越来越重要，已经成为国际上通信工程研究的热点之一。协议测试理论是协议工程学的一个重要分支。协议测试是在软件测试的基础上发展起来的。根据对被测软件的控制观察方式，软件测试分为三种：白箱测试、黑箱测试和灰箱测试。协议测试是一种黑箱测试，它依据协议标准来控制、观察被测协议实现的外部行为，对被测协议实现进行测试。协议测试有三种类型的测试：一致性测试、互操作性测试和性能测试。一致性测试是基础，是通过观察具体实现在不同的环境和条件下的反应行为来验证协议实现与相应的协议标准是否一致。要保证不同的协议实现在实际网络中能成功的通讯，还需要检测某一协议实现与其它系统之间的交互过程是否正常，这是互操作性测试。另外还要对协议的性能进行测试，如健壮性(robustness)、吞吐量等。大致说来，这些工作都是用一种形式化描述语言来描述协议，并使用一种生成机制来自动生成对应该协议的测试集，这些描述语言包括 SDL、LOTOS、Estelle 等。当今，用户对可靠性的要求越来越高，一致性测试和互操作性测试变得尤为重要。因此，本人认为协议测试也是一个重要研究领域。

现在移动通信发展越来越快，各国都在研究第三代移动通信^{[1][2]}。基于 IP 技术的第四代移动通信(蜂窝宽带无线接入技术)也早已被提上日程，如 ITS、HAPS。因此对移动通信软件的测试也越来越关键。3G 测试分为三个阶段：物理层参数测试、互操作性测试、现场性能测试。测试方法主要有单元测试、验收测

试、互操作性测试、稳固性测试、系统测试、一致性测试、集成测试等。

OSI 协议一致性测试方法和结构规范是在 1987 年提出的, 后又在 1994 年由 Richard J. Linn and M. Ümit Uyar 撰写出版该书, 已经成功应用在互连网协议如 HTTP、SMTP (简单邮件传输协议) 以及通信网络 ATM/B-ISDN 协议等。SDL^[3] 主要应用在工业领域; LOTOS^[4] 主要用在复杂的分布式处理系统中进行设计、验证和证实; Estelle^[5] 主要用在实际生活系统中, 着重于实现, 而不是验证和设计。

OSTG (optimal test sequence generation, 最佳测试序列生成) 问题是协议测试的重要研究领域, 也就是找出一个长度最小的测试序列。AHO 等人在 1988 年提出一个叫 SUIO/RCPT (RURAL CHINESE POSTMAN TOUR) 算法, 该算法是在第一阶段给每个状态采用最小长度的 UIO 序列, 在第二阶段找出一个乡村邮递员旅行^[6]。SHEN 等人在 1989 年提出一个叫 MUIO/RCPT 的算法, 该算法是通过在第一阶段为每个状态采用多个最小长度 UIO 序列来达到对 SUIO/RCPT 算法的改进, 序列长度可减少 4-37%。这两种算法都是采用非重叠的方式来最小化测试序列长度。1993 年 RAYMOND 等人提出基于重叠的多个 UIO 生成测试序列的算法。

本课题主要对通信协议一致性测试方法进行研究及第三代移动通信标准 TD-SCDMA 的终端高层进行协议测试方法的研究和测试。

1.2 测试发展过程

测试一般来说都分为硬件测试和软件测试; 软件测试是使用为发现错误所选择的输入和状态的组合而执行代码的过程。测试按测试范围可以分为单元 (Unit) 测试、集成 (Integartion) 测试、系统 (System) 测试。单元测试的范围通常由一个相对小的可执行体组成; 集成测试的范围是软件和硬件单元的一个完整系统或子系统; 系统测试的范围是一个完全集成了的应用。测试就其应用领域又可以分为电子组装测试、芯片测试、声学测试及几何、力学、热工中使用的测试技术等, 其中发展最快的要算通信领域的测试技术。

目前在电子组装测试领域中使用的测试技术种类繁多, 常用的有手工视觉检查 (MVI)、在线测试 (ICT)、自动光学测试 (AOI)、自动 X 射线测试 (AXI)、功能测

试(FT)等。

1.2.1 通信测试仪表

通信领域测试发展较快的是通信测试仪表，通信测试仪表在通信产品研制、生产和应用的各个过程中，用于测试相关的通信器件、设备、系统、网络的特性，使之符合相关规范、标准的要求，以确保网络和业务运行的可靠和畅通。

通信测试仪表作为尖端的高技术产品，在很大程度上反映了一个国家的通信技术水平。世界通信强国都具有很强的通信测试仪表研发能力，也具有较大的应用规模。

随着光纤通信、移动通信和以 IP 为代表的网络技术和应用的飞速发展，其相应的测试技术也呈现高速发展的态势，光纤通信仪表、移动通信仪表和网络测试仪表成为当今通信测试仪表研制、开发和应用热点。

1. 光纤通信测试仪表

光纤通信的发展始于 80 年代初，是电信核心网和接入网的主要传输方式，也是有线电视网、计算机数据网主干段的主要传输方式。近年来，光纤通信干线传输容量越来越大，光纤接入向最终用户越来越接近，应用的覆盖面也越来越广。因此，不断有支持 SDH、DWDM、光放大器的新的光纤通信测试仪表面世，满足光纤通信研发、生产、检测等环节的测试要求。基本的光测试仪表，例如光源、光功率计等，在测试精度、操作简单性和智能化程度方面都得到了大幅度提高，非常适用于工程和维护测试的需要，有力地促进了光纤化发展。

2. 移动通信测试仪表

在移动通信测试领域，伴随着模拟移动通信的淘汰和数字移动通信的应用普及，数字化测试技术逐渐取代了模拟技术。GSM 系统和 CDMA 系统测试在移动通信的发展建设中产生了重要作用。由于移动通信发展迅速、潜力巨大，再过若干年，移动通信用户规模将超过固定网，因此移动通信测试仪表市场也随之扩大。另外移动通信技术新概念不断提出，从 2G 到 3G，以及中间的 2.5G 等，移动通信新仪表也是层出不穷，甚至相关技术是否应用和什么时候应用还没有确定，测试仪表的广告已先入为主，移动通信测试仪表市场的热度可见一斑。

3. 网络测试仪表

以 IP 为代表的网络技术的发展速度前所未有, IP 网络被认为是三网融合的最佳结合点, 用于 IP 网络互联的设备越来越多, 采用的技术也更加复杂。如何设计符合客户需求的网络、如何在设备选型时比较各厂家设备的性能、如何在网络实际运行之前判断网络是否符合设计的要求、如何在网络运行中发现网络性能瓶颈和故障原因等一系列问题就成为网络运营商和网络用户必须面对和解决的问题。所有这一切都需要 IP 网络测试仪器的应用和 IP 网络测试技术的支持。

通信测试仪表与通信技术、计算机及软件技术、网络技术、集成技术的发展息息相关。近些年来, 通信测试仪表的技术走向主要表现为计算机化、综合化、智能化和网络化。通信及通信测试仪表将相互促进, 共同快速发展。

1.2.2 移动通信测试技术

随着通信技术的发展, 发展迅速的移动通信测试技术具有如下新的特点:

1、从技术层面看: 移动通信技术经历了从模拟到数字技术的转变。从窄带的数字技术 (GSM、CDMA) 向宽带数字通信技术 IMT2000 的发展, 从单一语言通信向以数据通信为主的移动互联网、移动多媒体的发展。从硬件测试和低层协议 (1-3 层协议) 测试为主向以软件为核心和高层协议测试过渡。

2、从测试方式层面看: 移动通信设备制造商和网络运营商对测试仪器供应商的要求也正在经历从单台仪器-多仪表组合测试系统-自动 (程控) 测试诊断系统的过渡, 从离线测试-实时在线测试-实时测试诊断、校准系统的过渡。

3、从市场关系看: 移动通信测试方案供应商与移动通信设备制造商和系统运营商的关系也发生了性质上的变化。由纯粹的商业关系变成了一种合作伙伴关系, 即测试仪表/设备-测试系统-测试服务。

1.2.3 第三代移动通信测试系统

第三代移动通信 (3G) 的测试可以分为三个阶段: 第一阶段, 移动设备基本性能测试; 第二阶段, 移动设备/网络间互操作性测试; 第三阶段, 移动现场测试。

第一阶段, 物理层参数测试。移动设备的物理层性能参数。这一规范通过一系列空中接口测试来评估移动通信发射机和接收机的性能。这些测试通常是在利

用实验室设备搭建的有线测试环境中进行的。可初步归纳出以下几个关键测量项目：邻道功率比(Adjacent Channel Power Ratio, ACPR)、接收信号编码功率(Received Signal Code Power, RSCP)、干扰信号编码功率 (Interference Signal Code Power, ISCP)、块误码率 (Block Error Rate,BLER)、位误码率 (Bit Error Rate, BER)、接收信号强度指示 (Received Signal Strength Indicator, RSSI)、误差向量值 (Error Vector Magnitude, EVM)、灵敏度测试(Sensitivity)。

第二阶段，互操作性。完成第一阶段的测试后，移动设备要接受第二阶段的测试，这一阶段的测试是通过一系列移动基站信号发送和呼叫处理测试来评估移动设备的协议层性能。与第一阶段测试类似，第二阶段测试也是在有线环境中进行的。真正的第二阶段测试是在配备有主要通信体系设备制造商产品的互操作性 I/O 测试实验室中进行的，测试由一系列通过实际系统设备产品进行的试验构成。

第三阶段，现场性能测试。第一阶段和第二阶段的测试是在实验室中有线的环境下对移动设备的物理层和协议层性能进行验证，而第三阶段测试的目标则是验证移动设备在真实环境中在这两个层次上的性能。移动设备的制造商一般情况下要与购买其产品的服务商合作进行测试。这些测试在一组确定的行驶路径上进行，要用到一系列呼叫处理操作。

1.3 论文安排

本论文共分为五章，第一章介绍了开展通信协议测试研究的意义、测试的发展过程、本论文的主要工作和论文的组织结构。第二章介绍了协议工程学、形式化方法、协议测试理论、协议一致性测试的基本概念和方法、协议测试通常采用的数学模型（有限状态机理论）和基于 CEFSM 模型的测试例自动生成方法。第三章论述基于 EFSM 的开发工具 SDL 和采用 CEFSM 对 3G 双模终端 TD-PRS 的 LLC 的 SDL 设计实现。第四章介绍一致性测试生成工具 TTCN，及采用基于 CEFSM 测试例生成方法对 3G 双模终端的 TD-PRS 的 LLC 的测试。第五章总结全文，并探讨了进一步的研究工作。

第二章 通信协议测试理论与方法

2.1 形式化方法

通讯协议 (Communication Protocol) 就是用在通信系统中, 将各个不同部分连接起来的通信管理规则。为了使得来自不同厂家的系统能够成功地进行通信, 必须有标准化的协议。

协议工程 (Protocol Engineering) 的研究范围包括协议综合、协议说明、协议证实与验证、协议转换、性能分析、自动实现和协议测试 (Protocol Test)。在协议工程领域里, 协议综合是根据协议标准文本说明设计出正确的形式化协议说明的过程; 协议说明是指对协议标准文本的形式化说明; 协议证实与验证主要是对形式化协议说明的语法/语义正确性的证实和验证; 协议转换主要研究如何实现不同协议的进程之间的互操作问题; 性能分析是指根据已设计完成的协议说明来评价其所具有的特性; 自动实现是指由机器对形式化协议说明直接进行编译处理, 产生协议实现的主体部分; 协议测试是通过实验的方法发现协议实现在功能和性能方面的错误。

形式化方法可以分为两类: 数学逻辑法 (mathematical logic formalism) 和描述法 (description formalism)。数学逻辑法提供了形式化的语法, 严谨的数学语义和一组推导规则。有限状态机 FSM (Finite State Machine) 属于这种方法。描述法也有形式化的语法和数学语义。与逻辑法不同的是, 描述法不是为形式化操作而特别设计的, 所以它没有推导规则。这类方法的一个例子是抽象语法表示法 ASN.1 (Abstract Syntax Notation One)。它虽然具有形式化的语法, 但并没有推导规则。

2.2 一致性测试

2.2.1 一致性测试概念

一致性测试是一种“功能测试”, 它依据一个协议的描述对协议的某个实现

进行测试，判别一个协议的实现与所对应的协议标准是否相一致。一致性评价过程是评估某实现或系统与一个或多个 ITU-T 建议或国际标准相一致的所有一致性测试活动的全过程。它分三个阶段：测试准备、测试操作、测试执行。图 2.1 给出了协议一致性测试的过程框图^{[7][8]}。

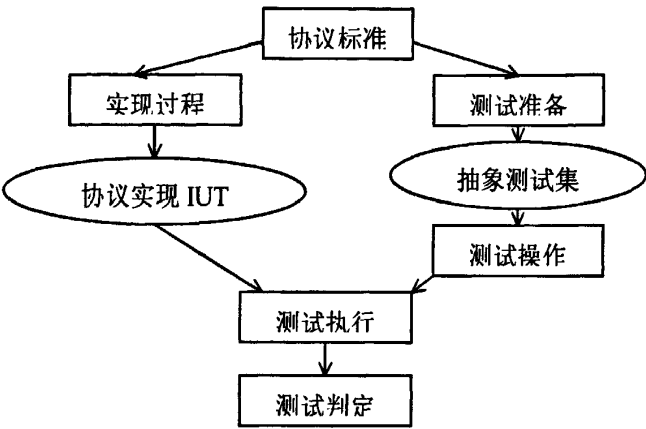


图2.1 协议一致性测试的过程框图

2.2.2 测试生成技术

测试生成即测试准备，是协议一致性测试的第一步。测试生成的目的是从协议描述中生成独立于协议实现的，用严格的测试表示语言描述的抽象测试集。测试集自动生成依赖于被测协议的形式化描述，其过程大致可以分为两步：首先，根据协议标准文本给出协议的形式化描述；第二步，采用一定的测试集自动生成算法产生出抽象测试集。测试集自动生成技术有很多算法，下面着重对基于有限状态机 FSM 的测试集生成方法进行介绍。

基于有限状态机 FSM 的测试集自动生成方法可以分为以下几种策略：变迁遍历、T-方法、U-方法、D 方法、W-方法^{[9][10]}等。

在变迁遍历方法中，经常使用中国邮递员算法或乡村中国邮递员算法来找到一条遍历路径。中国邮递员问题是由我国数学家管梅谷教授首先提出的。设邮递员从邮局出发，遍历他所管辖的每一条街道，将信件送到后返回邮局，要求路径最短。用图论的术语描述就是从一个强连通的有向图里找一个代价最小的路径，

该路径是包括每条边至少一次的回路。对于所有的有限状态机 FSM 都存在着邮递员遍历路径，此路径对 FSM 中的各个变迁至少访问一次，邮递员算法很容易在计算机上实现，变迁遍历法并不能检测出 FSM 的输入 / 输出序列中的所有错误。

T-方法假定图是强连通的；完全规定的，也就是任一个状态对所有的输入均有变迁；最小的，也就是不能再化简的图。该方法是对任一个随机输入，直到把所有的变迁遍历一次为止，然后用化简的方法把环去掉。例图 2.2 的有限状态机有五个状态：0、1、2、3、4，0 为初始状态，输入是 A、B，输出 0、1。T-方法的测试序列为：

B A B A B A A A A A A A B B
0 0 3 3 4 0 3 4 2 1 4 2 1 2

U-方法为每个状态产生一个唯一的输入/输出序列（UIO 序列），用来识别该状态，每个状态的 UIO 序列不唯一。图 2.2 的 FSM 的 UIO 序列如表 2-1 所示，λ 是空的输出。例如要测试从 Si 到 Sj，r 为复位，要先生成每个状态的 UIO，然后用 r，使状态机 FSM 到达初始状态 S0；找一个最短的从 S0 到 Si 的路径，连接上 Si 到 Sj 的变迁以及 Sj 的 UIO 序列，就是生成的测试序列。

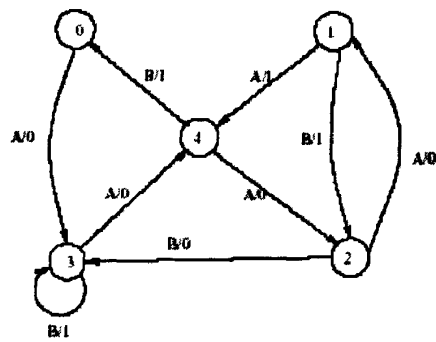


图2.2 U-方法、D-方法、W-方法举例

表 2-1 FSM 的 UIO 序列

状态	UIO
0	B/λ
1	A/1 A/1
2	B/0
3	B/1 B/1
4	A/1 A/0

图 2.2 的 FSM 的 U-方法的测试序列为：

rAAAAAArAAAABBrAAABBBBrAAAArAABB (2→1→2→3→4→0) 。

表 2-2 输入 DS- 序列 BB 后每个状态的输出

状态	输出
0	λ λ
1	10
2	01
3	11
4	1 λ

D-方法是用 DS-序列验证尾状态，此方法依据输出序列识别状态；DS -序列 (distinguishing sequence)是输入序列，对不同的开始状态，输出是不同的；因此可以用输出来判断当前所处的状态。图 2.2 的 FSM 的 DS-序列为 BB，对每个状态输入 BB 后的输出如表 2-2 所示。D-方法跟 U-方法有点象，不过是把末状态的 UIO 序列换为 DS-序列而已。图 2.2 的 FSM 的 D-方法的测试序列为：rAAAABBrAAAABBBBrAAABBBBrAABBrAABBB (2→1→2→3→4→0) 。

W-方法在识别变迁尾状态时使用了特征集 W，一个状态的特征集中往往会有多个特征序列。特征集是一些输入集合，用在不同的状态上，产生的最后输出是不相同的。图 2.2 的 FSM 的 W-集合为为{A, AA, B}，使用 W-集合后的 FSM 输出如表 2-3 所示。W-方法先构造 W-集合，跟 U-方法类似，不过在末状态用 W-集合换掉 UIO-序列。图 2.2 的 W-方法测试序列为：rAAAAAAABrAAAABAAABrAAABBAABrAAAAABrAABAAAB (2→1→2→3→4→0) 。

表 2-3 FSM 使用 W-集合后的每个状态输出

状态	A	AA	B
0	0	0	λ
1	1	1	1
2	0	1	0
3	0	1	1
4	1	0	1

四种方法中，U-方法、D-方法、W-方法的一致性测试错误覆盖比 T-方法的一致性测试好，因为前三者都对末状态和边测试，分别采用 UIO 序列、DS-序列和 W-序列；后面的 T-方法不对状态进行检查，只能检查协议机的哪些变迁存在，

哪些变迁不存在,并不能检测出 FSM 的输入/输出序列中的所有错误;与其他三种方法比较, T-方法的容错最差,不能检查状态错误,生成的测试序列最短; U-方法相对来讲较易于实现,而且它所生成的测试例长度较短;对有的状态机来说不一定有 DS-序列; W-方法最复杂,生成的测试例最长。唯一输入/输出序列(UIO)显然优于有区别的序列(DS-序列)和特征序列(W-序列),因此广泛采用 UIO 来进行测试序列的生成。

2.3 有限状态机理论

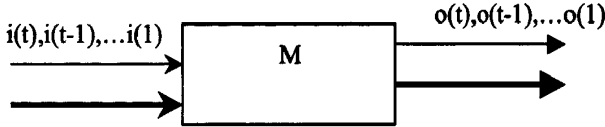


图2.3 信号转换器

考察上图 2.3 的信号转换器,假设机器 M 在时间 $t=t_0, t_1, t_2, \dots$ 工作,其中初始时间 t_0 及时间间隔 $\Delta t_i = t_i - t_{i-1}$ 都是任意的。为了简化起见,常把它们表示为 $t=0, 1, 2, \dots$ 。在 $t=0$ 时,将已知的初始条件作为初值,在以后每一瞬间 t ,通过输入信道,机器 M 收到一个输入信号 $i(t)$,它就产生一个输出信号 $o(t)$,通过输出信道输出。提供给机器的一串输入信号 $i(1), i(2), \dots, i(t)$ 称为一个激励,所产生的输出符号串称为 M 对此激励的响应。

所有输入符号组成的集合称为输入符号集,记为 I ,所有输出符号组成的集合称为输出符号集,记为 O 。现实世界中的机器都可以认为由有限个部件组成,且每个部件都有有限个状态。假设机器 M 由 N 个部件组成, $s^{(j)}(t)$ 表示第 j 个部件在瞬间 t 的状态,那么在时间 t , M 的状态是一个 N 有序组

$$\langle s^{(1)}(t), s^{(2)}(t), \dots, s^{(N)}(t) \rangle$$

故 M 总状态的最大可能数是所有部件状态数的积。若每一部件的状态数不超过 K ,则 M 的状态数不超过 K^N 。用 $s(t)$ 表示 M 在时间 t 的状态,即 $s(t) = \langle s^{(1)}(t), s^{(2)}(t), \dots, s^{(N)}(t) \rangle$,所有不同的 $s(t)$ 组成的集合称为状态集,记作 S 。机器 M 在 $t=0$ 时所处的状态称为初始状态,记作 S_0 ,即 $S_1 = S(0)$ 。机器 M 在接收到输入信号

$s(t)$ 时, 它将产生一个输出 $o(t)$, 同时, 状态 $s(t-1)$ 转换到 $s(t)$, $o(t)$ 和 $s(t)$ 不仅与 $i(t)$ 有关, 而且与机器在前一瞬间的状态 $s(t-1)$ 有关。因此存在一个函数 δ , 用它来描述机器的状态:

$$s(t)=\delta(s(t-1), i(t)), t \geq 1$$

δ 称为状态转换函数, 它的定义域是状态集 S 与输入符号集 I 的笛卡尔积, 值域是状态集合的子集, 即:

$$\delta: S \times I \rightarrow S$$

还有一个函数用它来描述机器的输出:

$$o(t)=\lambda(o(t-1), i(t)), t \geq 1$$

它称为输出函数, 定义域也是 $S \times I$, 值域是输出集合 O , 即:

$$\lambda: S \times I \rightarrow O$$

状态数有限的机器称为有限状态机 FSM (Finite State Machines) ^{[11][12][13]}, 描述状态机可以采用状态图。如果迁移函数 δ 和输出函数 λ 定义于所有的状态输入组合, 我们称这种有限状态机是完全描述的 (或完全定义的)。状态图是一个有向图, 其中每一个结点表示机器的一个状态, 每一个有向弧指出从一个状态到另一个状态的转换, 此有向弧上加以起始标记 i/o , 其中 i 、 o 分别表示相应的输入符号和输出符号, 对于初态 S_0 , 用一个指向它的箭头来标明, 如图 2.4。

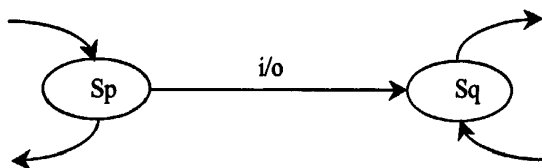


图2.4 状态图

根据应用的需要, 人们对有限状态机作不同的约束和扩展, 产生不同的状态机模型, 常用的为有限状态机模型 FSM、扩展有限状态机 EFSM (Extended FSM)、通信有限状态机 CFSM (Communicating Finite State Machines)、通信扩展有限状态机 CEFSM (Communicating Extended Finite State Machines) ^{[14][15][16][17]}。

一个有限状态机 FSM 是一个 6 元组。 $F = \langle I, O, S, S_0, \delta, \lambda \rangle$, 每个元素的含义说明如下:

▷ I 是输入符号集, 为一个有限集合;

- O 是输出符号集，为一个有限集合；
- S 是内部状态集，为一个有限集合；
- $S_0 \in S$ ，是初始状态；
- $\delta: S \times I \rightarrow S$ ，迁移函数（也叫做次态函数）；
- $\lambda: S \times I \rightarrow O$ ，输出函数。

迁移函数和输出函数一起刻画 FSM 的行为。

例如图 2.2 的 FSM，输入 $I=\{A, B\}$ ，输出 $O=\{0, 1\}$ ，状态 $S=\{0, 1, 2, 3, 4\}$ ，初始状态 $S_0=\{0\}$ ，状态跳变函数和输出函数分别如表 2-4 和表 2-5 所示。

表2-4 状态跳变函数 δ

末状态 起始状态	输入	
	A	B
0	3	-
1	4	2
2	1	3
3	4	3
4	2	0

表2-5 输出函数 λ

输出 起始状态	输入	
	A	B
0	0	-
1	1	1
2	0	0
3	0	1
4	0	1

基于 FSM 的测试序列生成方法一般分为两个阶段：

- 为每个状态确定和导出一些特定的输入/输出序列（I/O SEQUENCE），状态由 SI（State identification）来表示确定和区分。
- 把子测试序列连接起来，生成测试序列。

有限状态机 FSM 模型主要用在模拟时序电路、程序的语义分析、模式匹配、通信协议等方面。FSM 不能很好描述大的复杂系统，缺点在于两方面：不能很好的描述变量操作，不能模仿任意数值的转变能力，也就是它的缺点是 FSM 固有的顺序性和不分级的平面性。基于 FSM 模型的测试方法主要针对被测实现的

控制部分进行的，这种测试可以发现变迁错误和传输错误。但一个系统通常有两部分组成，一个是控制部分，还有一个是数据部分，采用 EFSM 模型能较好的实现对这类系统的描述。

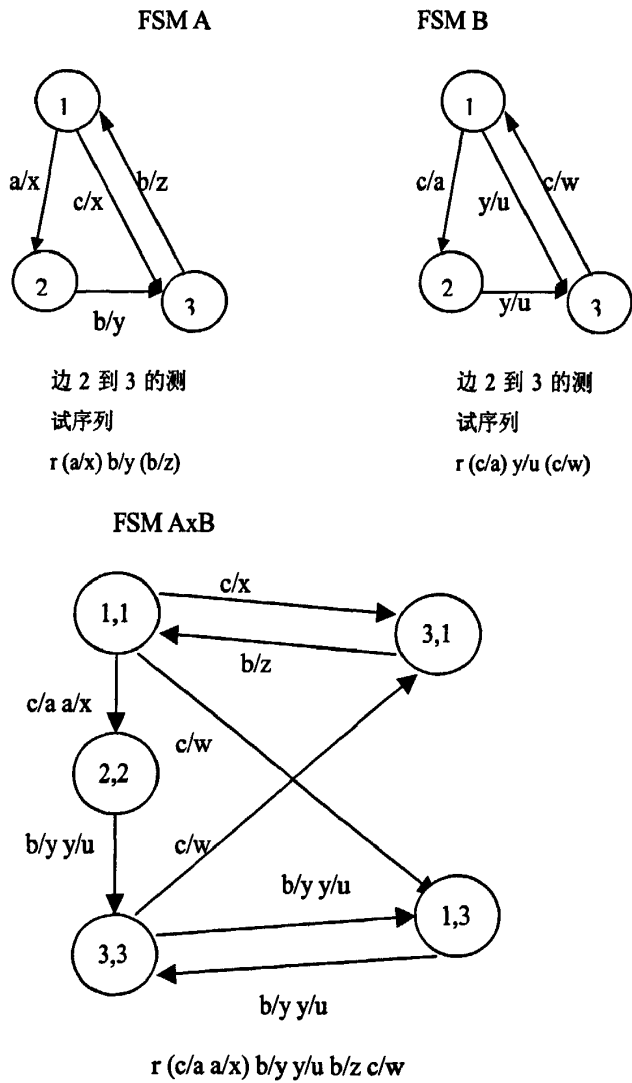


图2.5 计算[2,2]到[3,3]的测试序列

对控制流部分测试采用的传统方法是 T-方法、U-方法、D-方法、W-方法，这也是 FSM 模型采用的方法；对数据部分测试是采用数据流图。数据流图是一个有向图，结点表示程序的功能单元，边代表数据对象流。功能单元可以是一个状态、变迁、过程或程序。通过测试所有的输入变量的值来完全描画系统，测试系统的数据部分，但是要对输入变量的所有可能的取值进行测试是不现实的，通

常都是按照一定的条件来选择输入变量的一个子集进行测试,也就是只对一部分数据行为进行测试。对 EFSM 的测试同时要考虑数据流和控制流部分,因此要比单纯的对 FSM 的测试复杂,采用的方法有上下文相关依赖的唯一序列 CIUS (context independent unique sequence) (CIUS 比传统的控制特征序列 DS-序列、UIO 序列、W-序列的控制流覆盖好)或者把 EFSM 分为数据流图 DFG 和控制流图 CFG 来测试等。

FSM 一般是描述比较简单的系统, 有时候许多通信协议规范需要用通信 FSM 来描述, 因此就有了 CFSM。对 CFSM 的测试有两种方式, 一种方法是先作出状态机 A 和状态机 B 的测试例 T1、T2, 然后将 T1 和 T2 连接^[18], 例图 2.5。另一个方法是对状态机 A 和状态机 B 先做向量积, 生成新的状态机在用 U-方法来生成测试例。

EFSM 模型不能描述系统内部的依赖关系，有时候系统大的话，要考虑系统内部的通信；而 CFSM 没有考虑系统的数据行为部分，因此要描述这类系统，就要采用 CEFSM 模型。

2.4 基于 CEFSM 模型的测试方法

通信扩展有限状态机 CEFSM (Communicating Extended Finite State Machines) 是一个二元组, $\text{CEFSM} = \langle \text{EFSM}, \text{Channel} \rangle$:

- EFSM 是一个扩展有限状态机的集合;
- Channel 是信道的集合;

扩展有限状态机 EFSM 定义类似与上述 2.5 的定义。每个信道有三个元素，源设备、目的设备和可以通过该信道传送的信号列表。

对一个软件系统来说,测试例的自动生成是比较耗费时间和重要的。目前有两种生成测试例的方法:一个是基于规范的测试例生成方法,另外一个是基于代码的测试例生成方法。前者是根据规范,生成测试例;而后者必须在代码完成后再做测试例。当然基于规范的测试例生成方法具有很大的优越性,一是它可以独立于规范的实现;二是可以提前发现错误,代价低;三是它适于做系统级测试,而且可以在代码实现之前做出测试例。因此本文论述的 CEFSM 是基于规范的测试例生成技术。

基于规范的测试例自动生成技术的焦点在于采用的测试例自动生成模型, 有两类模型: 数据和行为。采用数据模型的是 TesterMaster 和 ATEG, 采用行为模型的是使用统一建模语言 UML 或者 SDL^{[19][20][21][22]}。这些模型都要求为生成测试例而建立新的模型, 这就增加了测试例生成的复杂度; 同时它们都要求覆盖所有的状态空间; 另外对实时系统不能很好的处理。

一般的对 CEFSM 的测试, 一种方法是把 CEFSM 分为几个单独的 EFSM, EFSM 还可以转变为 FSM, 然后分别进行测试。对 FSM 的测试, 是采用 U-方法, 首先把规范描述为 FSM; 然后把 FSM 转变为有向图, 状态代表图的顶点, 边代表变迁, 也就是输入/输出对, 从任意一个顶点 S_i 到另一个顶点 S_j 的测试 (也就是从一个状态 S_i 到另一个状态 S_j 的测试), 就是乡村邮递员问题。这种方法会遇到状态爆炸 (explosion) 问题, 因此有时候难以实现。另外一种方法是采用向量积的方式, 有两种向量积方式^[18], 一个是对状态机 A 和状态机 B 先做向量积, 生成新的状态机再用 U-方法来生成测试例; 另一个是先作出状态机 A 和状态机 B 的测试例 T1、T2, 然后做 T1 和 T2 连接。上述两种方法都要求为新的测试目的重新生成新的状态机模型, 都存在一定的局限性。

本文论述的 CEFSM^[16]不需要为生成测试例目的而重新建立模型, 它使用现有系统的模型, 包括数据和行为来生成测试例。使用 FSM 表示模型的行为和事件, FSM 的扩展变量表示模型数据, 交互信道表示模型的通信, 因此该方法是数据、行为和通信规范三者的结合。CEFSM 解决了分支覆盖问题, 包括行为判决覆盖和跟数据相关的决策覆盖。这种测试例生成方法的性能评价是用基于代码的分支覆盖语言 SDL, 而不是通常的 C、C++、JAVA 等编程语言。该方法采用假定的优先级和支配分析法来高效生成测试例, 经过一个测试遍历后重新修改各个节点的权值, 从而减少生成测试例的数目, 减少了维护测试例和执行测试例的代价, 重复该过程直到所有的节点至少遍历一次。

例图 2.6 要遍历结点 8, 就要经过结点 1、2、3、4 (该测试例为 T1), 也就是结点 8 被结点 1、2、4 支配; 只要对结点 8 做了测试, 结点 2 就被遍历过了, 所以说结点 8 的权值 (3) 高于结点 2 的权值 (1), 结点 8 的优先级高于结点 2。如果要测试经过结点 6 的遍历, 要经过结点 1、2 (该测试例为 T2), 结点 6 被结点 1、2 支配; 由于结点 1、2 已经在测试例 T1 中遍历过, 所以新增加的结点是

6, 因此在经过 T1 遍历后, 结点 6 和结点 3 有相同的权值 (1); 也就是在经过 T1 后, 覆盖结点 6 的测试例和覆盖结点 3 的测试例有相同的优先级。

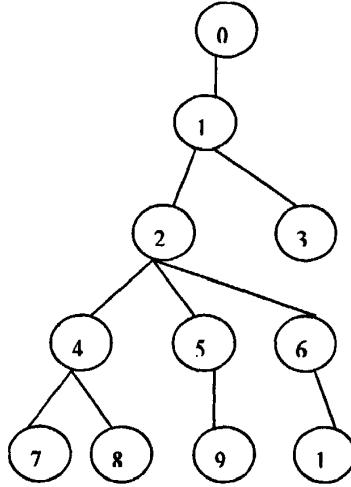


图2.6 优先级、支配、权值的修改举例

在按最大覆盖优先级较高的方法选择了测试例后, 采用后向跟踪策略把测试遍历映射到结点集, 如 T1 的结点是 4、2、1, 依次倒推; 然后再用前向检查的方法来验证该测试遍历, 前向验证有三种情况, 一是所需要的外部输入或者变量初始化; 二是需要的内部信号; 三是发向环境的外部信号。

CEFSM 有四个主要特征: 采用已经有的系统规范模型, 该模型包括数据和行为信息; 每生成一个测试例后重新计算所有未被覆盖结点的权值; 采用后向跟踪来构造遍历路径; 采用前向验证把遍历路径转变为测试例。具体的操作方法如下: 1. 采用 SDL 用形式化方法描述规范; 2. 生成流图; 3. 按优先级、支配的方法生成最大分支覆盖; 4. 每生成一个覆盖后重新计算所有没有被覆盖的结点的权值; 5. 采用后向跟踪的方法 (若结点不唯一, 则采用深度优先搜索算法) 构造变迁路径, 该路径覆盖了有最大权值的分支; 6. 重复 3、4、5 直到所有的结点至少被遍历一次; 7. 把变迁路径转变为测试例, 采用前向验证, 生成带有外部输入、输出和任务的测试步。

2.5 本章小结

本章主要介绍了协议一致性测试方法、测试例生成方法, 如变迁遍历、T-方法、U-方法、D-方法、W-方法; 同时对 FSM、CFSM、EFSM、CEFSM 进行了介绍, 接着重点介绍了基于 CEFSM 模型的测试例自动生成方法。

第三章 3G 双模手机中对 LLC 的仿真

3.1 规范描述语言 SDL

规范说明和描述语言SDL (Specification and Description Language) [23]作为国际标准化的正式语言, 被用来规范描述实时系统。作为SDL相关技术, Z.120建议——消息序列图(message sequence charts, MSC) [24]和Z.105建议——抽象语法符号(abstract syntax notation -one, ASN.1)和SDL形成一个完整的描述工具。SDL的特点是既可以用来说明 (Specification) 一个系统所应具有的行为 (Behavior), 也可以用来描述 (Description) 一个系统实际具有的行为。SDL把应用领域划分成两部分: SDL系统和环境。SDL系统是设计者关注的对象, 用SDL语言来说明和描述。系统以外的部分都被视为系统的环境, 系统和环境之间可以通过交换信号来通信, 保证系统能对外部事件作出反应。系统与环境的关系如图 3.1 所示。

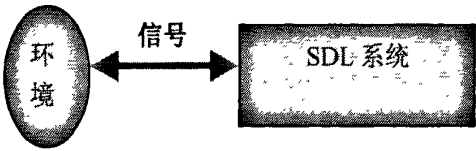


图3.1 系统与环境的关系

国际电联电信标准部ITU-T将SDL描述为“实时系统的特性描述语言”。绝大多数SDL概念有文本SDL/PR和图形SDL/GR两种表述形式。SDL使用扩展有限状态机EFSM (Extended Finite State Maching) 的概念来描述对象。SDL由系统 (SYSTEM)、块(BLOCK)、进程(PROCESS)、过程(PROCEDURE)组成, 采用从上至下逐层细化的方法。

SDL是开发通信软件常用的工具, 通信软件主要分为两大类: 电信软件和计算机网络软件。电信软件主要包括: 电话交换软件、移动通信软件、智能网软件等; 网络软件主要包括: 各种网络协议和网络应用软件。用SDL开发通信软件的过程如下: 首先是用SDL的图形方式设计出系统SDL图, 将其转化为C语言源代

码，或者CHILL源代码，嵌入到实时操作系统RTOS中，通过建立虚拟的模型用 TTCN仿真测试，以及实际的模型来验证测试，最后得到成熟、稳定的软件。SDL反复渐进的开发过程如图3.2所示^{[25][26]}。首先是使用例子构造规范MSC图，然后根据MSC图，创建SDL模型，通过仿真和验证运行SDL模型，生成MSC图，和规范MSC图比较，发现错误则修改SDL模型和添加新的例子到规范MSC图里，最后通过运行SDL模型，构造TTCN抽象测试集。其中仿真是交互的，可以观察输入输出信号；验证是采用状态空间搜索的方法，以发现死锁等。验证的状态搜索机制有位状态搜索、穷尽状态空间搜索、树搜索、随机步搜索、POWERWALK搜索五种策略。位状态搜索：采用的是穷尽从开始状态到终止状态的所有状态的搜索方法，测试例是有限的。穷尽状态搜索：搜索状态空间的每一个状态，以生成测试例。树搜索：按树深度搜索。随机步搜索：是随机的搜索状态，以生成测试例，用于回归测试。POWERWALK是树搜索、随机步搜索的结合，用于大的SDL系统。

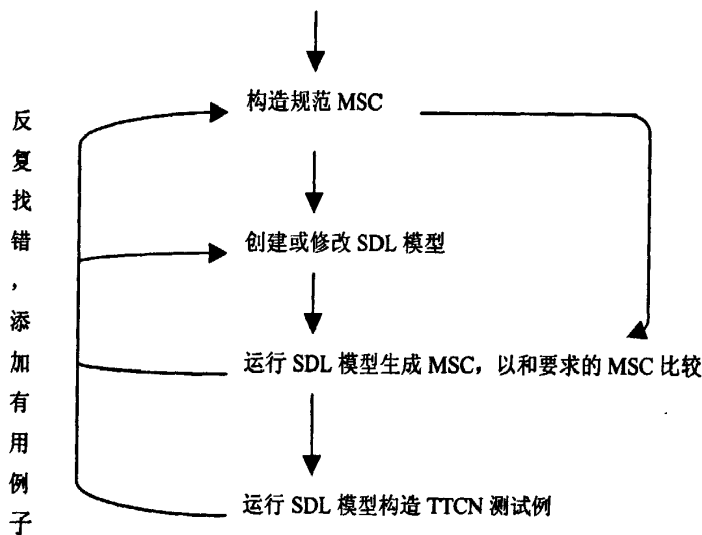


图3.2 SDL 反复、渐进开发过程

3.2 消息序列图 MSC

消息序列图 MSC 是 ITU-T Z.120 建议中给出的一种形式化语言，用来描述多个实体之间和实体与环境之间消息交互的顺序。在协议开发中，MSC 可以用来描述两个协议实体之间或多个协议实体之间协议数据单元(PDU)的交换时序。

MSC 有两种文法表示：文本文法（textual grammar）和图形文法（graphical grammar）。MSC 特点是：支持消息跟踪、图形化、形式化、结构化设计，可以与 TTCN 联合，支持测试例的自动生成。

MSC 图描述一个系统中若干组件之间的通信和组件与系统的外部环境之间的通信，系统组件在 MSC 图中以实例的形式出现，一张 MSC 图只描述系统的部分行为。MSC 图有两种：一种是简单的 MSC 图，描述实例之间的消息交互情况和部分系统行为；另一种是高级的 MSC 图(high-level msc diagram)即 HMSC，描述简单 MSC 图的组合情况，构成更复杂的 MSC 图。

在 MSC 图中，消息用来在两个实例之间或实例与环境之间交换信息；定时器可以用来监视某一事件（一般为等待接收消息）的到达。在通信系统中，经常有这样的情况，一个通信实体向另一个通信实体发出一条消息后，要等待对方的回应消息，如果在一定的时间内收不到回应消息，说明出现了异常，该通信实体要么重发消息，要么做异常处理，这时就要用定时器来进行描述。定时器包括两个操作和一个事件，即启动定时器、复位定时器和定时器超时。

3.3 3G 双模终端中用 CEFSM 模型对 LLC 的仿真

在第三代（3G, third Generation）移动通信终端的开发中，为了跟二代平滑过度，大多数的开发商采用了双模手机概念，我国是基于 3G 标准 TD-SCDMA 的，因此双模手机模式为 GSM 模式和 TD-SCDMA 模式。为了实现由第二代的 GSM 网络到具有我国自主知识产权的第三代移动通信系统 TD-SCDMA 的平滑过渡，引入 TSM (TD-SCDMA System for Mobile) 系统及其标准。

TSM 系统是在 GSM 核心网络支持下的 TD-SCDMA 系统。其标准基于 TD-SCDMA 标准，但又不同于 TD-SCDMA 标准。两者在物理层的基本结构上保持一致。而在高层，TSM 规范向 GSM 标准靠拢，TD-SCDMA 标准与 3GPP 保持一致。TSM 系统的核心思想就是在现有的 GSM 网络中，用 TD-SCDMA 技术来提供第三代移动通信业务，以便于实现 GSM 向 3G 的平滑过渡。一方面利用第三代移动通信的频谱来解决 GSM 系统容量不足，特别是在高密度用户区容量不足的问题，另一方面可以为用户提供第三代移动通信业务，即在初期高达 384kbps 的各种速率的数据业务。目前我校的移动通信研发中心主要是在做 TD-SCDMA 中的数据业务 GPRS（通用无

线分组业务)的开发,简称为 TD-PRS,因此严格来讲 TD-PRS 相当于是 2.5 代的产品,本节讲述采用协议工程学的方法,对 TD-PRS 中逻辑链路控制模块 LLC (Logical Link Control)的实现。

3.3.1 TD-PRS 概述

TD-PRS 定义了新的 TD-PRS 无线信道,这些信道的分配非常灵活,物理资源可以被每个激活的用户共享,上行链路和下行链路的信道安排可以独立进行。无线接口资源可以被话音和数据业务动态地共享,共享的方法取决于业务负荷和运营者的管理方法。不同的编码方案允许每个用户有不同的比特速率。TD-PRS 协议软件体系结构如图 3.3 所示^{[27][28][29]}。

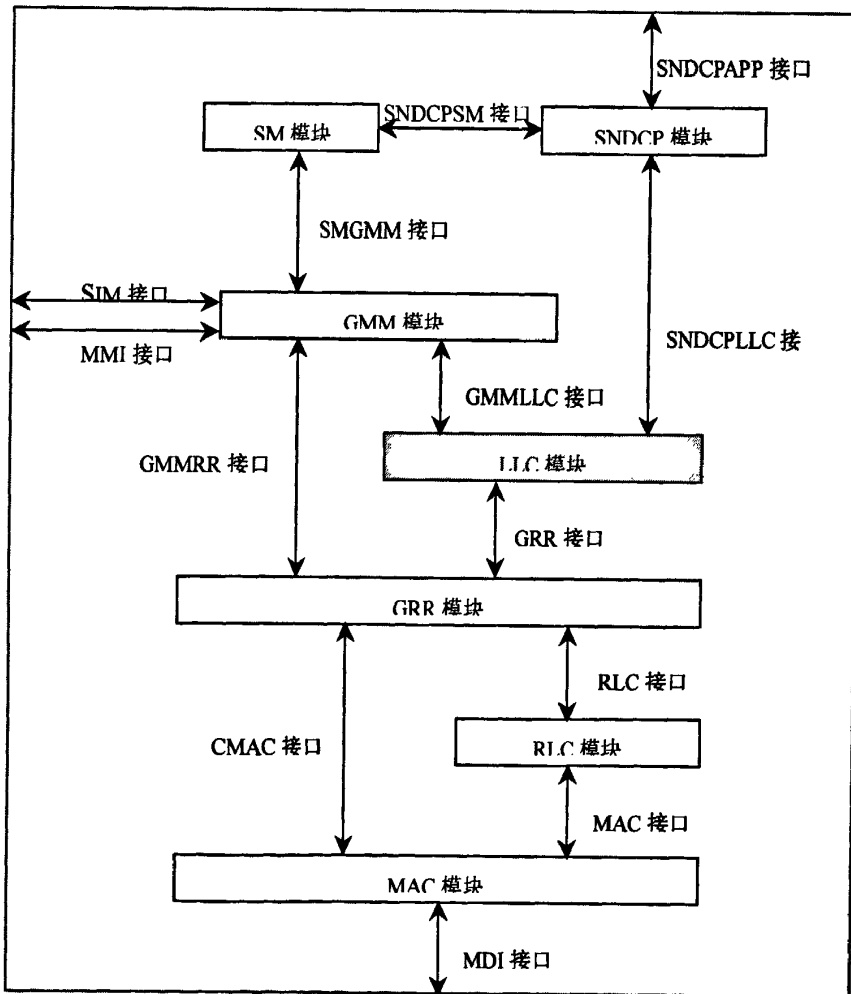


图3.3 TD-PRS 协议软件体系结构

TD-PRS 支持标准数据协议的应用,允许与 IP 网络和 X.25 网络互连互通。

它可以提供点对点和点对多点的业务,例如交通信息和国际铁路联盟的列车控制应用,它允许通过 TD-PRS 无线信道进行短消息业务的传输。

TD-PRS 可以支持间断、突发的数据传输,又能支持偶发性大数据量的传输,它支持不同服务质量 QoS (Quality Of Service) 的业务方式。能够快速启动分组传输,典型的延时为 0.5 到 1 秒。典型的计费方式可以基于传输的数据量。

TD-PRS 的安全功能和现有的 TD-SCDMA 安全功能相同。TD-PRS 业务支持节点 SGSN(Serving GPRS Support Node),根据现有 TD-SCDMA 的算法、密钥和准则来执行认证和加密,TD-PRS 为分组数据传输采用更为优化的加密算法过程。一个 TD-PRS 用户可以使用不能感知 TD-PRS 的用户识别模块 SIM 卡(Subscriber Identity Module)和能够感知 TD-PRS 的 SIM 卡接入 TD-PRS 业务。

为了发送和接收 TD-PRS 数据,用户设备 UE (User Equipment) 应该激活需要使用的分组数据地址,这种操作使得 UE 出现在相应的 GPRS 网关支持节点 GGSN (Gateway GPRS Support Node) 内,从而可以启动与外部数据网络的互联互通。

会话管理 SM(Session Management)是指 GPRS 移动台连接到外部数据网络的处理过程,主要功能是支持用户终端对 PDP (Packet Data Protocol) 移动场景的处理,包括 PDP 移动场景的激活、解除和修改,以及匿名接入时 PDP 移动场景的激活和解除。

子网汇聚层 SNDCP (SubNetwork Dependent Convergence Protocol) 主要功能是将网络层特性映射成低层网络特性。SNDCP 之上的协议集由网络协议组成,SNDCP 把来自不同源将要被发送的数据进行合路。它使 TD-PRS 可以引进新的网络层协议,并且该协议在 TD-PRS 传输,不会对 TD-PRS 带来任何改变,这样所有与传输网络层协议数据单元相关的功能都可以由 TD-PRS 实体以透明的方式执行。具体说来 SNDCP 的主要功能包括:若干分组数据协议(PDP)的合路;将网络协议数据单元(N-PDU)分解成逻辑链路控制协议数据单元(LLC-PDU),反过来将 LLC PDU 组装成 N-PDU;用户数据和协议控制信息的压缩和解压缩。

移动性管理 GMM (GPRS Mobility Management) 有附着、分离、清除、安全功能、位置管理、用户管理、类型标记等功能。安全功能包括用户认证、用户识别码保密、用户数据和 GMM/SM 信令保密、识别码验证。位置管理包括小区更新过程、路由区更新过程。

无线资源 GRR (GPRS Radio Resources) 的功能包括小区选择和重新选择、不连续接收、无线资源管理、对 GPRS 下行链路传递的寻呼、切换。

无线链路控制子层 RLC (Radio Link Control) 提供与无线解决方案有关的可靠的链路, 保证无线块在 TD-PRS 空中接口传输的正确性, 它将提供如下功能: 将 LLC PDU 进行分段, 并重新封装进 RLC 块中; 提供后向纠错 BEC (Backward Error Correct); 提供自动重发请求 ARQ (Automatic Repeat Request)。

媒体接入控制子层 MAC (Medium Access Control) 子层负责处理来自 RLC 和 GRR 层的数据流。它对上层提供不确认传输模式的服务。至 RLC 层的接口是通过逻辑信道服务节点, 它根据 GRR 层的要求指配无线资源并向高层报告测量结果。它将提供如下功能: 在上下行链路上, 提供对有效数据和控制信令的多路复用; 控制 GRR 层的业务量; 确定高层信令的路径及支持相应 MAC 信令连接的维护。

MDI 接口 (MCU-DSP interface, MCU-Microprocessor Control Unit) 是 HL1 与 LL1 的接口, HL1 是物理层接口的高层, LL1 是物理层接口的低层。所以把物理层接口分为两层 HL1、LL1, 是因为多任务同时运行在 DSP (Digital Signal Processing) 上实现比较困难; DSP 对 C 语言程序执行的效率不高; 与高层接口部分涉及大量的接口原语解析处理等工作, 浪费 DSP 资源。

采用封包和隧道方法可以在 UE 和外部数据网络之间透明的传输用户数据, 这种方法是指对用户数据分组, 插入特定的 TD-PRS 协议信息, 然后在 UE 和 GGSN 之间传输。

3.3.2 LLC 功能

LLC 相当于国际标准化组织 ISO-7 层参考模型中的 2 层 (数据链路层) 的功能, 其作用是在 UE 和 SGSN (Serving GPRS Support Node 支持 GPRS 的节点) 的层 3 的实体之间传送信息。LLC 支持同一 Uu 无线接口中的多个 UE 或者在一个 UE 中的多个 3 层实体之间的信息传送, 提供通过逻辑连接在 UE 和 SGSN 的对等实体之间的消息传送, 它的功能有:

- ▶ 提供用数据链路连接标识符 DLCI (Data Link Connection Identifier) 来区分的一个或多个逻辑连接;
- ▶ 对通过逻辑链路传送的帧的顺序控制;

- 对在逻辑链路上信息传送、信息格式和操作错误的检测；
- 对检测到的信息传送、信息格式和操作错误的恢复；
- 对不可恢复的错误的报告；
- 流量控制；
- 加密；

为了完成上述的功能，需要下层提供的服务是：LLC 层用来区分每个 LLC PDU (Protocol Data Unit) 协议数据单元的第一个和最后一个字节的 LLC PDU 定界符；在 UE 和 SGSN 之间传输 UE 的地址-TLLI (Temporary Logical Link Identifier)。

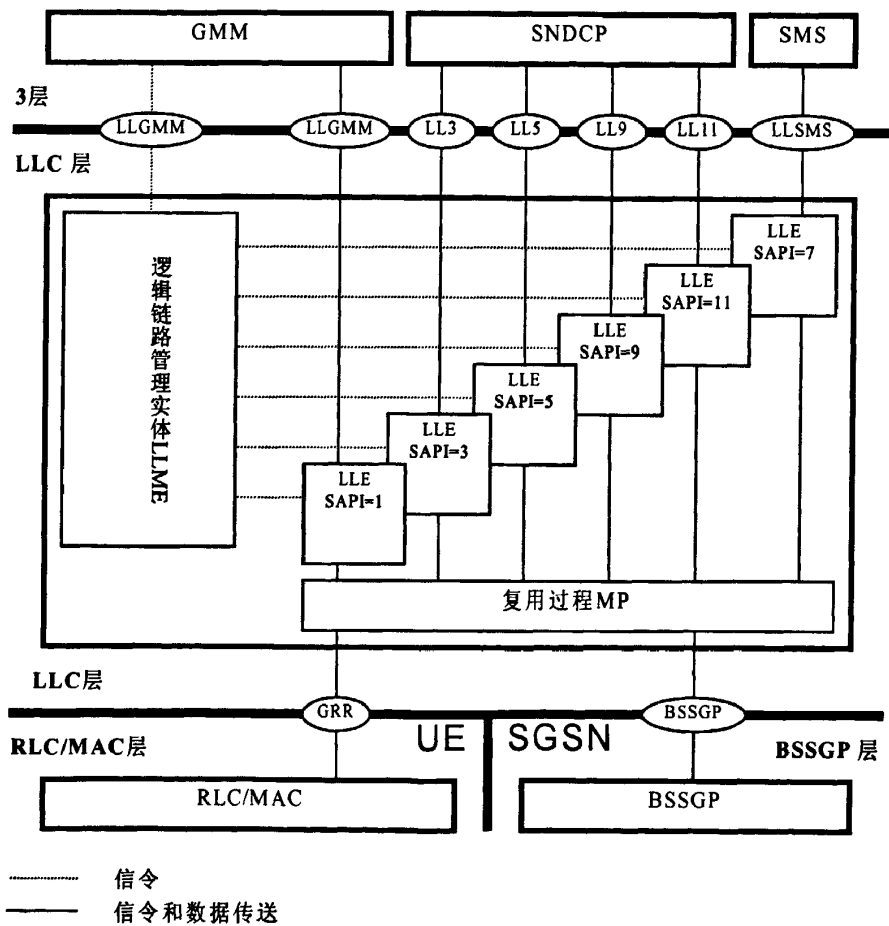


图3.4 LLC 功能结构

LLC 层的功能结构^[29]如图 3.4 所示，LLC 层处在无线链路控制 RLC 和基站子系统 GPRS 协议层 BSSGP (Base Station System GPRS Protocol) 之上，为 UE 和 SGSN 提供逻辑连接。在 LLC 层之上是子网依赖会聚层 SDCP，用来控制在 UE 和 SGSN

之间的网络协议数据单元 N-PDU (Network PDU) 的传输。

SAP (Service Access Point) 是服务访问点, 是 LLC 层为 3 层协议提供的逻辑连接。除了 SNDCP 层外, 还为 GPRS 的移动管理层 GMM 和短消息服务 SMS (Short Message Service) 提供连接。每个 LLC 层连接用 DLCI 唯一标识, 包括服务访问点标识符 SAPI 和 UE 的逻辑连接标识符 TLLI (Temporary Logical Link Identity)。每个 LLC 帧包括头、尾和信息字段。头和尾包括的信息有服务访问点标识符 SAPI、帧号和帧校验序列, 用以标识帧和提供可靠的传输。信息字段是变长的, 每帧的传送和重传都是由 LLC 控制的。

逻辑链路过程(Logical Link Procurement)包括多个逻辑链路实体 LLEs (Logical Link Entity), 每个逻辑链路实体控制每个连接的信息流。每个 TLLI 可以有多个逻辑链路实体, 每个 LLE 提供以下功能: 确认模式和非确认模式的信息传输、在异步平衡模式 ABM(Asynchronous Balanced Mode)下的流控制、帧错误检测。逻辑链路实体分析收到帧的控制字段并作出相应的应答和层与层之间的指示。另外 LLE 分析 LLC 层的服务原语以决定传送相应的命令和响应帧。每一个逻辑链路标识 DLCI 表示一个逻辑连接, 有一个逻辑链路实体与之对应。

在帧发送过程中, 复用过程(Multiplex Procedure)生成、插入帧校验序列字段, 执行帧加密功能, 提供在不同的逻辑链路实体之间的基于服务访问点的冲突解决。在帧接收过程中, 复用过程执行帧解密和检查帧校验序列字段。如果通过了 FCS (Frame Check Sequence) 的检验, 复用过程把该帧分发到相应的 DLCI 的逻辑实体。

逻辑链路管理实体 LLME (Logical Link Management Entity) 管理与每个连接有关的资源, 每个 TLLI 有一个逻辑链路管理实体。逻辑链路管理实体的功能是: 参数初始化、错误处理、连接的流量控制。

由 LLC 功能结构图看出, LLC 要跟 GMM、SNDCP、SMS、GRR 之间传送数据和控制信令, 因此不同的 SAPI 就相当于不同的实例, 共有 6 个实例, 采用不同的 QoS, 根据协议工程学方法, LLC 的协议实现使用 SDL 语言仿真, 采用扩展通信有限状态机 CEFSM 模型, 每个实例就是一个状态机, 不同的实例之间的通信采用队列相互交换信息。

3.3.3 LLC 的状态划分

LLC 支持两种方式的数据传送, 确认和非确认的方式, 确认模式需要建立异步平衡模式 ABM; 非确认模式数据可以在 ABM 下传送, 也可以在异步断连模式 ADM (Asynchronous Disconnected Mode) 下传送, 如短消息等, 即非证实的数据 UI (Unconfirmed Information) 和 GMM 控制信令的传输。因此为了确定 LLC 当前的操作模式, 需要相应的状态来标志当前 LLC 所在的工作方式, 即 ABM 和 ADM 两个主要状态是必须的。另外对 LLC 逻辑连接管理和定时器溢出等, 还需要其他的 5 个异常操作管理状态: 未分配 TLLI (TLLI_UNASSIGNED)、本地建立 (LOC_EST)、本地释放 (LOC_REL)、定时器超时恢复 (TIMER_REC)、远端建立 (REM_EST), 共有 7 个状态, LLC 状态图^[29]如图 3.5 所示。

在 ADM 模式下, 只能进行非证实数据的收和发, 已分配 TLLI 后就可以进入该模式。该模式可以接受 3 层和 LLC 的 XID (eXchange IDentification) 的协商, 即 U (Unnumbered format) 帧, 以及 3 层的 ABM 连接模式建立请求, 收到建立请求就进入 LOC-EST 状态。

在 ABM 模式下, 可以进行证实数据的传送, I (Information transfer format) 或者 I+S (Supervisory format) 帧的传送, 当然也可以进行 3 层和 LLC 的 XID 的协商; 同时也可以接受非证实数据的传送, 即 UI 帧收和发。同时还可以接受 3 层 GMM 的控制信令。当收到 LL_RELEASE_REQ 时就进入 LOC_REL 状态, 即本地释放的异常处理过程。ABM 方式可以有 4 个 SAPI 的证实数据收发, SAPI 为 3、5、7、9、11, 支持优先级控制, 以及对 I 及 I+S 的信息字段和 FCS 的加密。

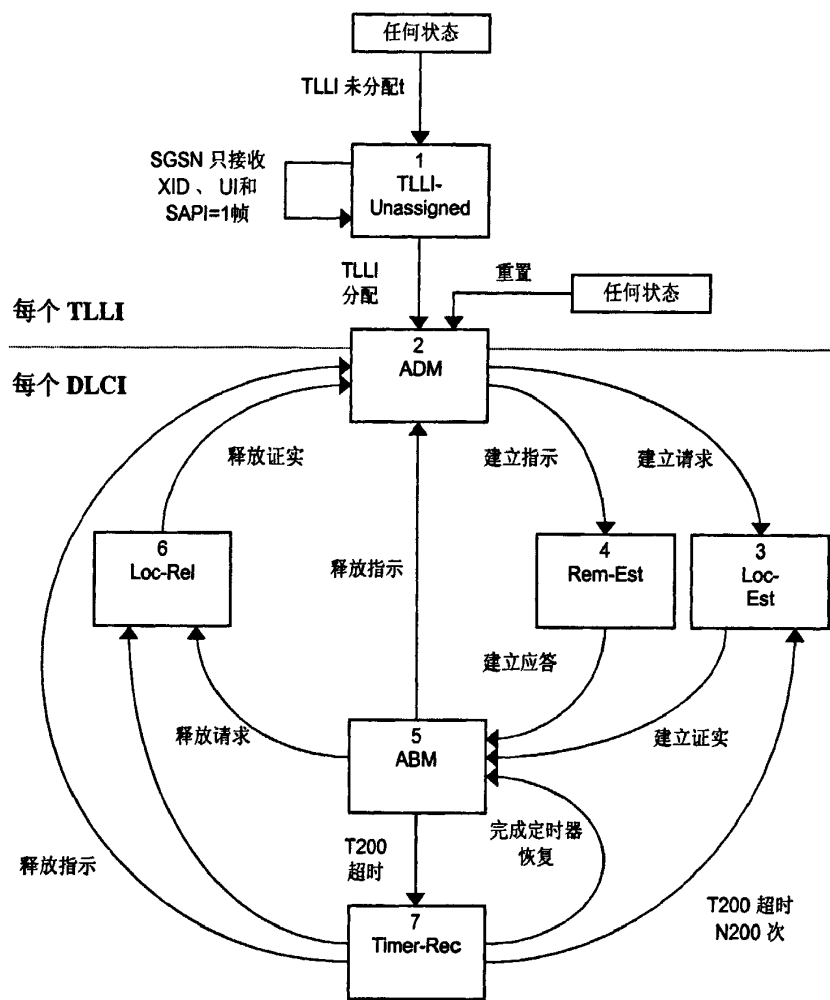


图3.5 LLC 状态转移图

由图 3.5 可以看出一个用户当前只有一个 TLLI，但可以有多个服务，即同时可以存在多个连接 DLCI，每一个 DLCI 由一个实例来管理，相当于一个状态机，都有虚线以下的状态跳变过程。每个状态下接收的信号如表 3-1 所示。

表3-1 LLC 各个状态下接收的信号

状态	信号
ABM	t_llc_T200, t_llc_T201, GRR_STATUS_IND, GRR_READY_TO_SEND, LLGMM_TRIGGER_REQ, LLGMM_RESUME_REQ, LLGMM_SUSPEND_REQ, LL_RELEASE_REQ, GRR_UNITDATA_IND, GRR_DATA_IND, LL_ESTABLISH_REQ, LL_UNITDATA_REQ, LL_DATA_REQ, LLGMM_ASSIGN_REQ
ADM	LLGMM_SUSPEND_REQ, GRR_STATUS_IND,

	LL_UNITDATA_REQ, LL_XID_REQ, LL_ESTABLISH_REQ, GRR_UNITDATA_IND, GRR_READY_TO_SEND, LLGMM_TRIGGER_REQ, LLGMM_RESUME_REQ, LLGMM_ASSIGN_REQ, GRR_DATA_IND, t_llc_T200, t_llc_T201
REM_EST	t_llc_T200, t_llc_T201, GRR_UNITDATA_IND, GRR_DATA_IND, LL_ESTABLISH_RSP, LL_XID_REQ, LL_UNITDATA_REQ, GRR_READY_TO_SEND
TLLI_UNASSIGNED	LLGMM_ASSIGN_REQ
LOC_EST	t_llc_T200, t_llc_T201, GRR_STATUS_IND, GRR_DATA_IND, LL_UNITDATA_REQ, GRR_READY_TO_SEND, LL_XID_REQ, GRR_UNITDATA_IND
LOC_REL	t_llc_T200, t_llc_T201, GRR_DATA_IND, GRR_STATUS_IND, GRR_DATA_IND, LL_UNITDATA_REQ, GRR_READY_TO_SEND, LL_XID_REQ, GRR_UNITDATA_IND
TIMER_REC	LL_RELEASE_REQ, GRR_READY_TO_SEND, GRR_DATA_IND, LL_UNITDATA_REQ, LL_XID_REQ, GRR_UNITDATA_IND

3.3.4 LLC CEFSM 模型的 SDL 实现

LLC 是 TD-PRS 的逻辑链路控制层，是为上层 GMM、SND CP、SMS 提供数据和信令的转发和传递功能，没有数据分段功能，处于上层 GMM、SND CP、SMS 和下层无线资源管理层 GRR 的中间。LLC 与其他 TD-PRS 模块的关系如图 3.3 所示。

采用 CEFSM 模型设计系统规范时，其设计思想是把系统分成多个 EFSM 模型来实现，多个 EFSM 组成总的 CEFSM 模型，不同的 EFSM 模型之间的通信可以采用先入先出消息队列（FIFO），也可以采用信道，也就是模型的定义可以根据实际的需要修改，本文两者都采用。对 LLC 的实现，可以根据 LLC 的功能，把 LLC 分为多个实例，共有 1、3、5、7、9、11 六个实例，因此要用六个 EFSM 模型来实现。每个 EFSM 由一个 SDL 的进程 process 来实现，总的 LLC 可以采用一个块（block）来实现。在具体软件实现时，由于实现的复杂度以及和其他模块的统筹考虑，决定所有的信号收发都通过一个中转模块 GSR 来实现，因此对 LLC 的具体实现，就可以只考虑单实例的情况。

由 LLC 的功能结构图可看出，LLC 跟其他模块 GMM、SND CP、SMS 的交互需要

多实例来实现，对多实例的管理和调度在 SDL 里用一个模块 GSR 专门来实现，以控制哪个实例来运行以及它们之间的消息交互，该模块实现了 LLME 的功能，剩下的 LLC 模块就相当于一个单实例，单实例通过一个状态机模型 EFSM 来实现，不同的模块之间的通信采用信道，同一个模块不同实例之间的通信采用消息队列。不同实例之间的通信都通过 GSR 来周转和管理，也就是总的 LLC 实现是采用 CEFSM 模型，但单个实例却是采用的 EFSM 模型，这样实现起来 LLC 的功能就比较单一，对 LLC 单实例实现的 SDL/PR 格式如图 3.6 所示。

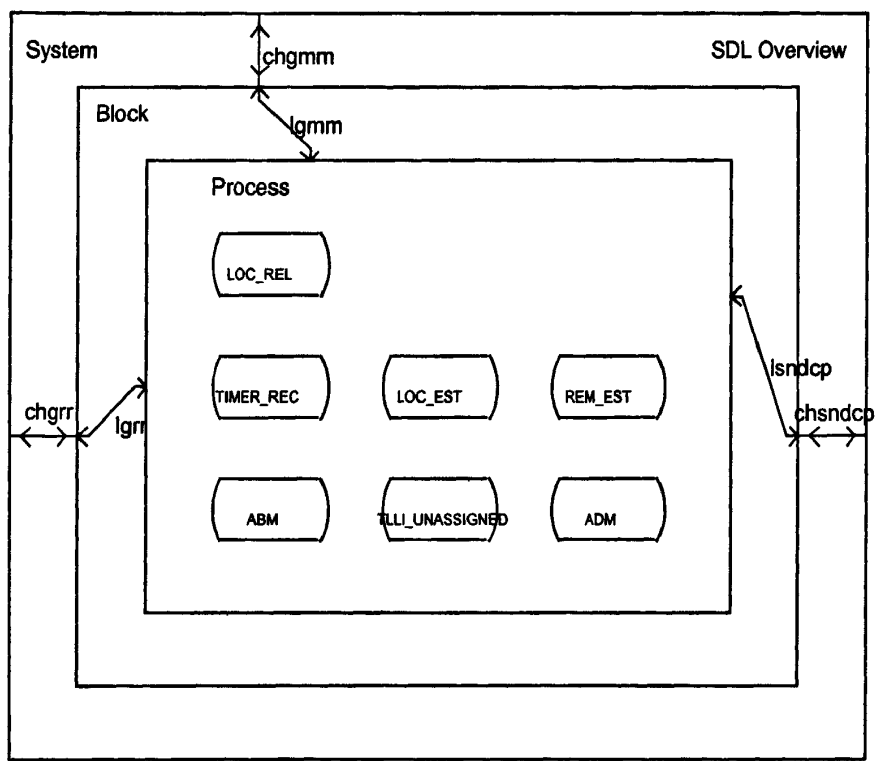


图3.6 LLC 单实例的 SDL 视图

系统 `td_prs_llc` 由一个块 `llc` 组成，该块包含一个进程 `llc`，相当于 CEFSM 的一个状态机，与 GRR 的信息传递是通过双向信道 `chgr`，与 GMM 的信令交互是通过信道 `chgmm`，与 SDCP 的信令和交互是通过 `chsndcp` 信道。

LLC 单实例的一个 EFSM 模型如表 3-2 所示，初试状态为 S_1 ，输入/条件信号为 10 个，任务/输出为 10 个，定义了正常情况下的 27 个变迁。

表3-2 LLC 单实例的 EFSM 模型

项目	取值
----	----

状态集	{<TLLI_UNASSIGNED>, <ADM>, <LOC_EST>, <REM_EST>, <ABM>, <LOC_REL>, <TIMER_REC>}
初始状态	= {S ₁ , S ₂ , S ₃ , S ₄ , S ₅ , S ₆ , S ₇ }
输入/条件集	{LLGMM_ASSIGN_REQ, LL_ESTABLISH_REQ, GRR_READY_TO_SEND, LL_RELEASE_REQ, GRR_UNITDATA_IND, GRR_DATA_IND, t _{llc_T200} , LL_UNITDATA_REQ, LL_DATA_REQ, LLGMM_UNITDATA_REQ } = {i ₁ , i ₂ , i ₃ , i ₄ , i ₅ , i ₆ , i ₇ , i ₈ , i ₉ , i ₁₀ }
任务/输出集	{LLGMM_UNITDATA_IND, LL_ESTABLISH_CNF, LL_RELEASE_CNF, LL_XID_IND, LL_DATA_IND, LL_UNITDATA_IND, LLGMM_UNITDATA_IND, GRR_DATA_REQ, GRR_UNITDATA_REQ, LLGMM_TRIGGER_IND, reset-timer, LLGMM_STATUS_IND } = {O ₁ , O ₂ , O ₃ , O ₄ , O ₅ , O ₆ , O ₇ , O ₈ , O ₉ , O ₁₀ , O ₁₁ , O ₁₂ }
变迁集合	{[S ₁ + i ₁ , S ₂], [S ₂ + i ₃ , O ₉ +S ₂], [S ₂ + i ₃ , O ₈ . O ₁₀ +S ₂], [S ₂ + i ₈ , S ₂], [S ₂ + i ₁₀ , S ₂], [S ₂ + i ₂ , S ₃], [S ₃ + i ₈ , S ₃], [S ₃ + i ₃ , O ₈ . O ₁₀ +S ₃], [S ₃ + i ₃ , O ₉ +S ₃], [S ₃ + i ₅ , O ₆ +S ₃], [S ₃ + i ₆ , S ₃], [S ₃ + i ₆ , O ₄ . O ₂ . O ₁₁ +S ₃], [S ₃ + i ₆ , O ₂ . O ₁₁ +S ₃], [S ₅ + i ₅ , O ₆ +S ₅], [S ₅ + i ₈ , S ₅], [S ₅ + i ₉ , S ₅], [S ₅ + i ₃ , O ₈ . O ₁₀ +S ₅], [S ₅ + i ₃ , O ₉ +S ₅], [S ₅ + i ₆ , O ₅ +S ₅], [S ₅ + i ₄ , S ₆], [S ₆ + i ₆ , O ₃ . O ₁₁ +S ₂], [S ₆ + i ₆ , O ₇ +S ₆], [S ₆ + i ₅ , O ₁₂ +S ₆], [S ₆ + i ₅ , O ₆ +S ₆], [S ₆ + i ₃ , O ₉ +S ₆], [S ₆ + i ₃ , O ₈ . O ₁₀ +S ₆], [S ₆ + i ₈ , S ₆]} = {t ₁ , t ₂ , t ₃ , t ₄ , t ₅ , t ₆ , t ₇ , t ₈ , t ₉ , t ₁₀ , t ₁₁ , t ₁₂ , t ₁₃ , t ₁₄ , t ₁₅ , t ₁₆ , t ₁₇ , t ₁₈ , t ₁₉ , t ₂₀ , t ₂₁ , t ₂₂ , t ₂₃ , t ₂₄ , t ₂₅ , t ₂₆ , t ₂₇ }

LLC 的基于 CEFSM 模型的 SDL 仿真图如图 3.7、3.8、3.9、3.10 所示，由于篇幅所限，本文对 LLC 标准规范进行了简化，没有考虑异常情况。B1-B27 是不同的分支，用来在第四章绘制 LLC 的流图。

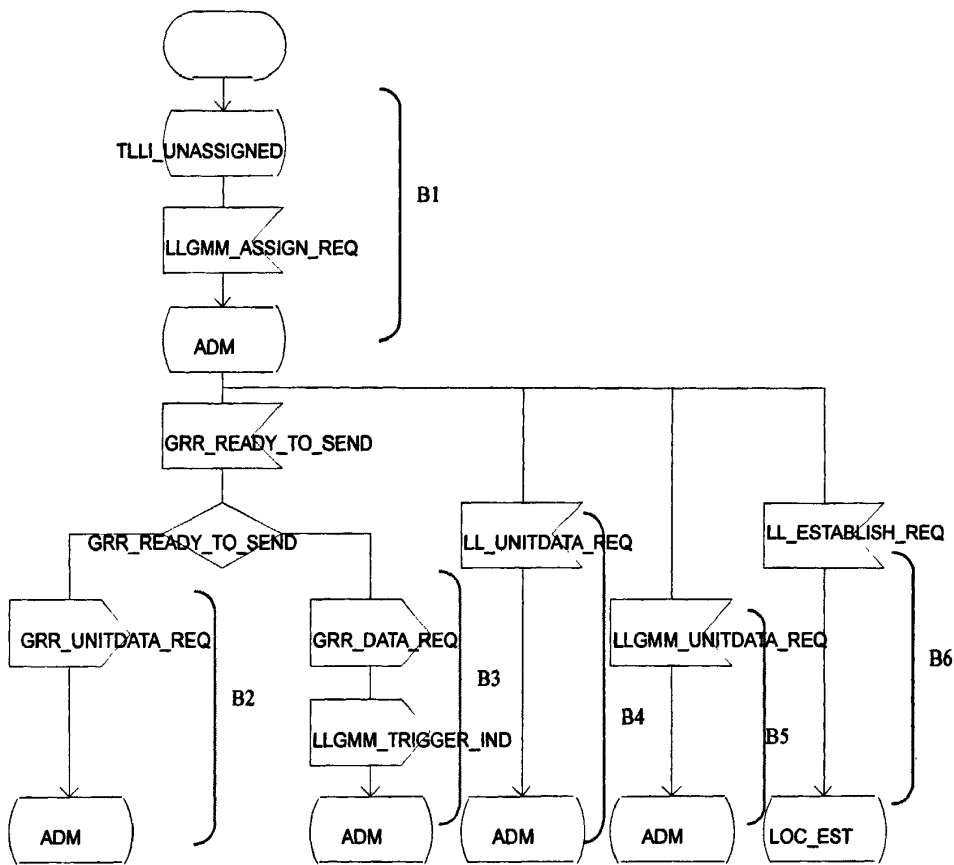


图3.7 LLC单实例的基于CEFSM模型的仿真图1

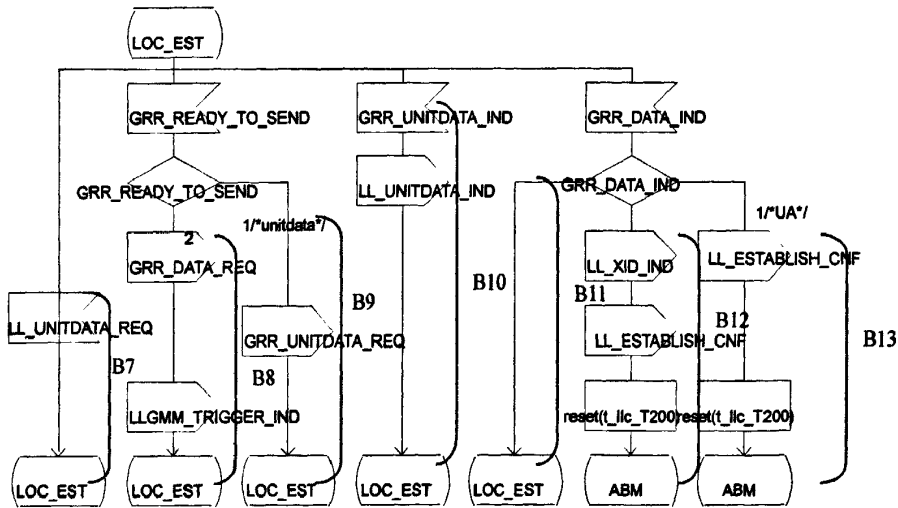


图3.8 LLC单实例的基于CEFSM模型的仿真图2

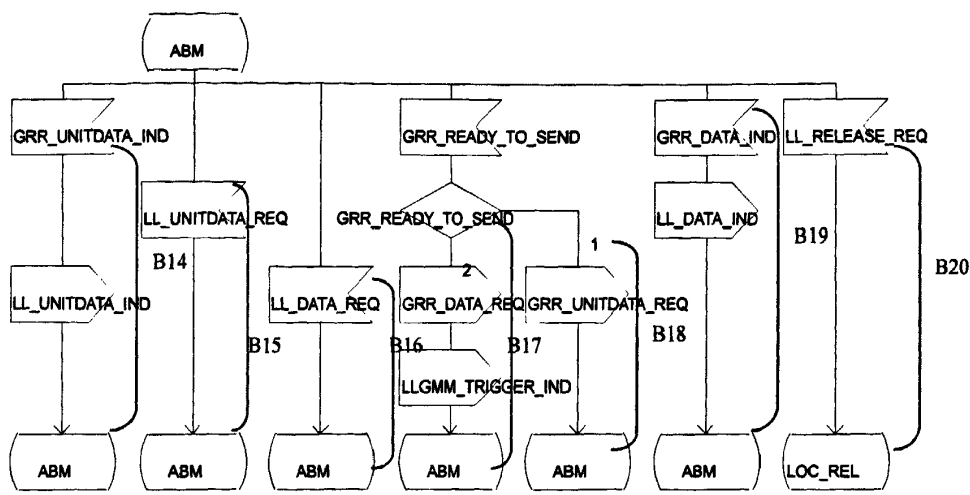


图3.9 LLC 单实例的基于 CEFSM 模型的仿真图 3

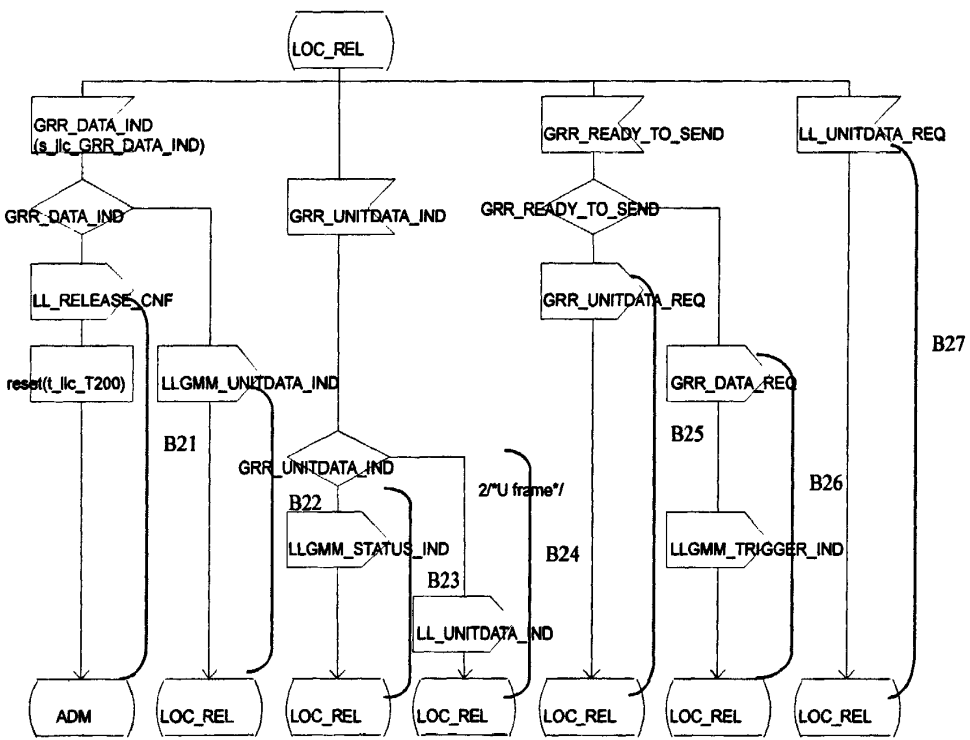
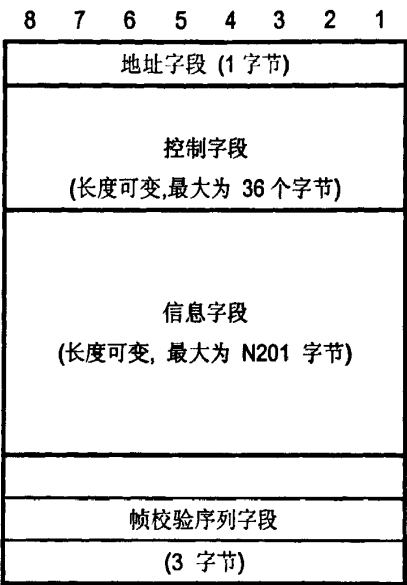


图3.10 LLC 单实例的基于 CEFSM 模型的仿真图 4

3.4 LLC 帧结构和优先级管理

3.4.1 LLC 帧结构

所有 LLC 对等层的信息交换都是按照一定的帧格式进行的，LLC 的帧结构如下：



帧头包括地址字段和控制字段，最小为 2 字节，最大为 37 字节。地址字段是一个字节，包括业务接入点 SAPI、协议标识符 PD、命令和响应指示 C/R。

控制字段由 1-3 个字节组成，有四种格式：I（信息帧）、UI（非证实的信息帧）、U（非编号格式帧）、S（监控帧）。主要是进行控制信息的组装，如指明传送方式（确认和非确认），数据是保护模式还是非保护模式，加密与否等。采用的地址字段格式如下：

控制域比特								字节
格式	8	7	6	5	4	3	2	
I 格式 (I+S)	0	A	X	N(S)				1
	N(S)			X	N(R)			2
	N(R)						S ₁ S ₂	3
S 格式	1	0	A	X	X	N(R)		1
	N(R)					S ₁	S ₂	2
UI 格式t	1	1	0	X	X	N(U)		1
	N(U)						E PM	2
U 格式	1	1	1	P/F	M ₄	M ₃	M ₂ M ₁	1

- A应答请求比特

M_n无编号功能比特

N(S)发送方发送序列号

P/F轮询比特，当作命令发布时；

PM保护模式比特

X空闲比特
- E加密功能比特

N(R)发送方接受序列号

N(U)发送方无确认序列号

终止比特，当作响应发布时

S_n监督功能比特

信息字段在控制字段之后，最大为 N201 字节，也可以没有该字段，若没有该字段则帧是监控帧或空帧。N201 是可以进行协商的一个参变量。

帧校验序列（FCS）字段包括 24 位的循环冗余校验（CRC）。CRC-24 用来检测帧头和信息字段的位错。FCS 字段包括通过 CRC 作用于头和信息字段计算出的值；CRC 的计算是在发送方加密消息之前，在接收方是在消息解密后进行。

3.4.2 优先级管理

一个 QoS 配置文件跟一个 PDP（Packet Data Protocol）有关，该属性被认为是多个数据传输属性的一个参数，它依照以下的属性定义所期望的服务质量：优先级、延迟级别、可靠性、总的吞吐量、平均吞吐量。根据上述属性可以定义多个可能的 QoS 配置文件，一个 PLMN（公共陆地移动网）只支持其中的一部分子集。

RLC/MAC 层支持四类无线优先级再加一个信令消息的级别。在上行接入的过程中，UE 指明采用的是四个优先级中的哪一个，以及上行接入的原因是传送用户数据或者信令，BSS 根据该优先级决定接入的无线优先级（Radio priority）和业务优先级（在有冲突的情况下的传送优先级）。在传送 MO SMS（主叫的短消息）采用的优先级的值由 SGSN 决定，并且由 SGSN 将该值通过附着（Attach accept）消息传送到 UE。用户数据传送的优先级由 SGSN 根据协商的 QoS 配置文件决定，通过 PDP 的激活过程和 PDP 的修改过程分发到 UE。

由于实现的复杂性，把 RLC/MAC 优先级放在 LLC 里实现。首先是数据和信令重传，看是否需要重传的，若有则按以下优先级规则来重传；若没有要重传的帧则发 I 帧，发 I 帧的优先判断也按下面的规则来发送；若上面两种情况都没有，则发 S 帧（如 ACK、RR 等）。优先级规则：首先根据 SAPI 来判断，SAPI=1 的信令先传送；若 SAPI 相同则按无线优先级；若无线优先级也相同，则看 QoS，根据 QoS 来判断哪个先传。

3.5 本章小结

本章概略的介绍了通信协议软件开发工具 SDL、MSC、通信协议软件开发过程；并采用协议工程学的方法对 TD-PRS 的逻辑链路控制 LLC 做了 SDL 仿真实现，采用模型是扩展通信有限状态机 CEFSM。

第四章 3G 双模手机中对 LLC 的测试

4.1 树表结合表示法 TTCN

TTCN (Tree and Tabular Combined Notation) 是 SDL 中用来定义测试规范的语言。国际标准化组织 ISO 推荐了一种专门描述测试集的半形式化的描述语言——树表结合表示法 (TTCN)。用以描述独立于任何实现的通用测试集。TTCN 具有两种格式：便于人类理解的 GR 图形格式和便于机器存储和处理的 MP 文本格式。

TTCN测试例配置由前导(Preamble)、测试体(Test Body)、后置(Postamble)三部分组成，前导是被测实现IUT从一个稳定状态转变到测试开始的状态；如果测试体的结束状态不唯一，则使用证实(Verification)过程来进行状态检查；同时后置则使被测实现IUT进入稳定状态。它们之间的关系^{[30][31][32][33]}如图4.1所示：

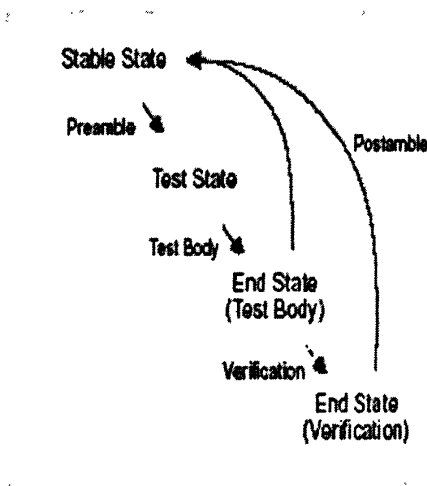


图4.1 TTCN 测试例测试机制

TTCN 描述的测试集具有标准性、通用性和可交换性等优点。用 TTCN 描述的标准抽象测试集由四部分组成：

- 测试集概述部分 (Suite Overview)
- 说明部分 (Declaration Part)
- 约束部分 (Constraints Part)

►动态描述部分 (Dynamic Part)

目前的最新版本是 TTCN-3(Testing and Test Control Notation), TTCN 有两种自动生成测试集的方法, TTCN LINK 和 TTCN AUTOLINK。

►TTCN LINK

该方法基于 SDL 规范来生成说明部分, 包括数据类型、ASP 或 PDU 类型、PCO 类型以及缺省的动态行为。数据类型转换为 TTCN 里用 ASN.1 方式描述的数据类型; ASP 或者 PDU 也转变为 ASN.1 描述的 ASP 或 PDU; 把信道转变为控制观察点 PCOs, 但是说明部分可能有错, 需要人工修改。尤其是测试例的约束部分和动态部分需要人工创建, 因此该方法大部分工作不是自动的。在创建测试例时, TTCN 和 SDL 规范是协同工作的, 二者关系如图 4.2 所示^[23], SDL 通过与环境相连的信道, 也就是 PCO 跟 TTCN 交换信息。一个 SDL 系统的状态空间是用一个连通的有向图来表示的, 用图的顶点表示系统状态, 边代表 SDL 系统的变迁。LINK 的测试例是用状态空间的搜索来产生, 状态空间搜索采用的算法是穷尽搜索和随机步两种算法。穷尽搜索是把每个状态都遍历一次为止, 因为要把每个状态保存在基本内存中, 因此其缺点是占用基本内存大, 适合于小型的 SDL 系统; 随机步搜索算法的输入参数是状态列表、最大遍历深度和重复的次数, 这三个参数中任一个满足都可以终止搜索, 所以占用内存小, 缺点是不能保证所有的状态空间都遍历完。

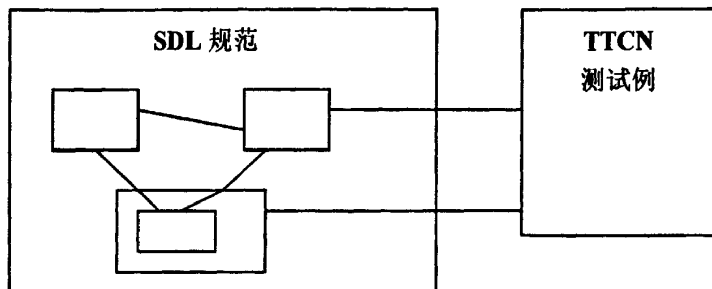


图 4.2 SDL 系统和 TTCN 关系

►TTCN AUTOLINK

用 TTCN AUTOLINK 生成 TTCN 测试集原理如图 4.3^{[34][35]}。该方法需要借助 SDL VALIDATOR 来生成, VALIDATOR 是基于状态空间搜索技术, 一个 SDL 系统的状态空间是用一个连通的有向图来表示的, 用图的顶点表示系统状态, 边

代表 SDL 系统的变迁。一个 SDL 系统的状态可能包含激活的进程实例、激活进程实例变量的取值、激活进程的控制流状态、激活过程、系统队列里的信号、激活的定时器等信息。根据配置文件的不同，SDL 系统的边可被转变为一个单独的任务、输入、输出。VALIDATOR 使用状态搜索来产生路径，路径的深度通过使用最大搜索深度、信道队列、SDL 事件的优先级、进程调度四种机制来控制，该路径被存为系统级别的 MSC 图，这些 MSC 图是 AUTOLINK 的输入。

在产生路径时，VALIDATOR 使用两种算法：一个是 POWER WALK，另外一个为 TREE WALK。POWER WALK 是穷尽搜索和随机步搜索的结合，需要定义三个参数，树深度、最大深度、继续深度，当搜索的树的深度达到树深度时，重新以新的状态作为根，来遍历状态空间，直到达到定义的最大深度为止。如果搜索深度达到最大深度，但还没有增加变迁覆盖，则使用继续深度来继续遍历。TREE WALK 使用的是确定的、启发式的算法，每当一个变迁增加树的深度，该变迁所到达的状态被添加到测试状态列表里，以便做为起始状态进行树遍历搜索。它生成的测试例比 POWER WALK 生成的测试例小、复杂，但是覆盖较大。

AUTOLINK 使用路径来生成测试例，该路径是一些输入/输出事件的集合，一条路径是 SDL 系统从一个状态到另一个状态的事件序列。AUTOLINK 把 SDL 系统里跟环境相连的信道转变为控制观察点 PCOs；对内部信号 AUTOLINK 不进行处理，AUTOLINK 只观察跟环境有关的信道上的信号，也就是跟环境相关的收发事件的名字被转换为约束的名字，采用 ASN.1 的 ASP 或者 PDU 来表示。AUTOLINK 可生成测试集的说明部分、约束部分和动态部分。测试例的概述部分由 ITEX 生成。

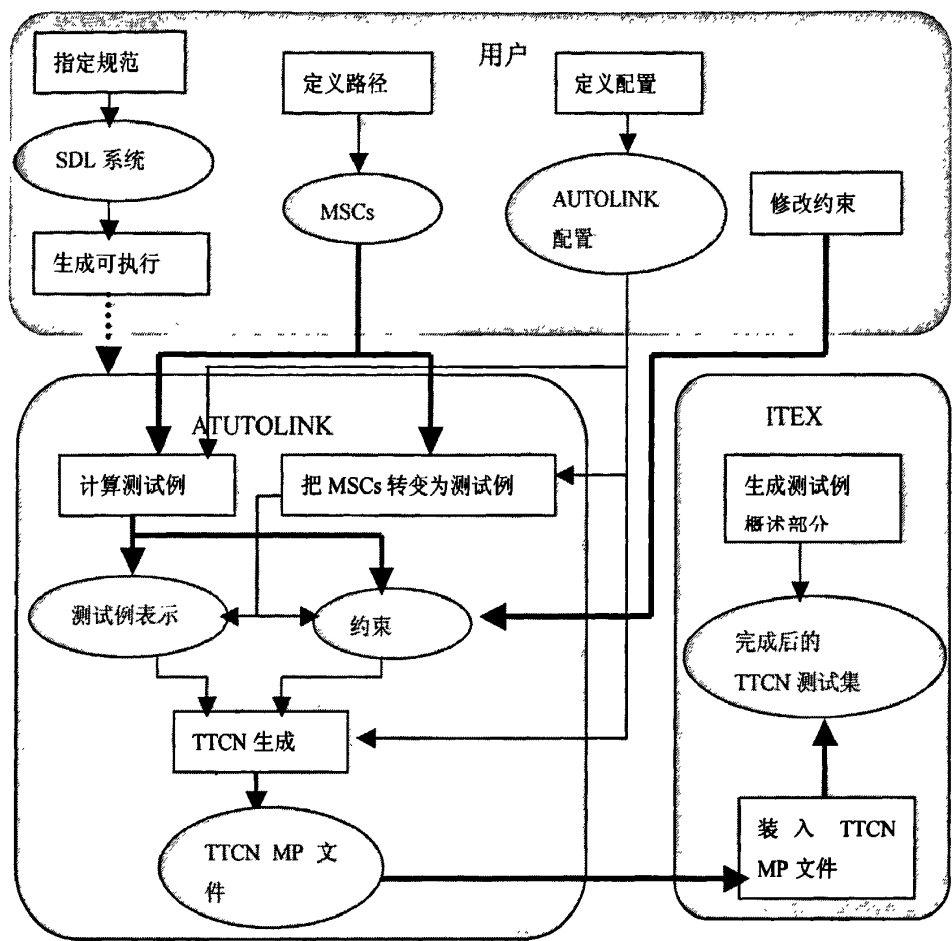


图 4.3 用 TTCNAUTOLINK 生成 TTCN 测试集

4.2 LLC 基于 CEFSM 模型的测试的实现

对 LLC 测试采用一致性测试方法，测试结构如图 4.4 所示^[36]，采用远程测试法。远程测试法是外部测试法的一种，不定义上测试器，也不需要显式的测试协调过程，只有一个下测试器。三个控制观察点 PCO: chgrr、chsndcp、chgmm，用来分别观察 LLC 与这三个模块间的信息交互。

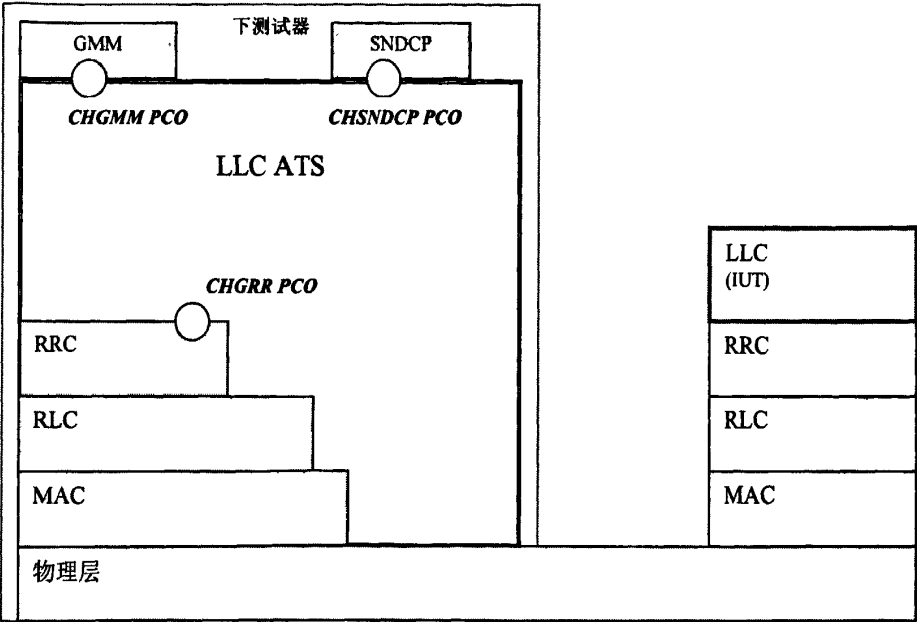
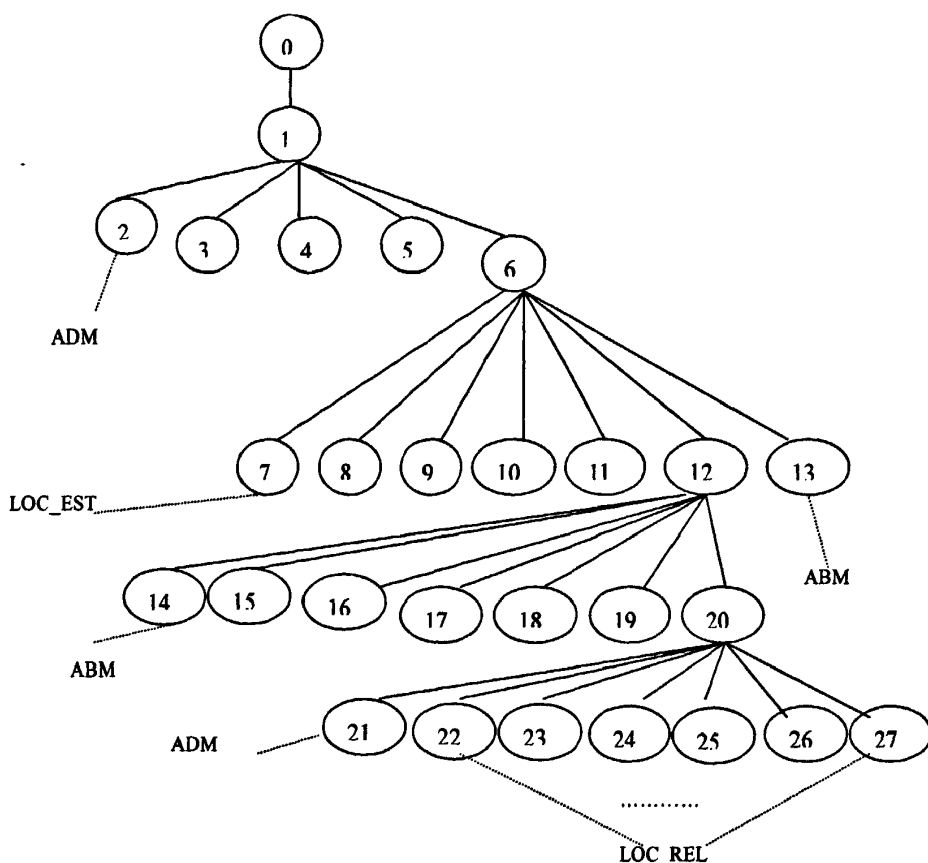


图4.4 LLC 远程测试法测试结构

4.2.1 LLC 基于 CEFSM 模型的测试算法描述

用 TTCN 对 LLC 进行测试，TTCN 正如其名称“树表结合表示法”所表明的，TTCN 包含两种表示方法：树型表示法，在测试例描述部分使用，用于表示测试事件之间的时序关系；表型表示法，在约束部分和声明部分使用，用于简化所有静态数据的表示，如数据类型，PDU 和 ASP 的格式，与测试事件相关联的测试判决等。要定义的 ASP 类型和约束的数量如第三章所列出的原语信号名字，采用的测试算法是第三章所描述的基于通信扩展有限状态机 CEFSM 模型的算法思想。也就是根据生成的 EFSM 的 SDL 图，绘制出流图，然后根据流图作出测试例。参照 SDL 图，把生成的测试例具体化，也就是需要收发的信号和状态跳变情况等，进行测试数据构造，以生成抽象测试集 ATS。对第三章 LLC 规范的 SDL 图生成的流图如图 4.5 所示。图中的结点 2、3、4、5 分支都进入相同状态 ADM，结点 6 到达 LOC_EST 状态，结点 7、8、9、10、11 都在 LOC_EST 状态，结点 12、13 分支都进入 ABM 状态，结点 14、15、16、17、18、19 也在 ABM 状态，结点 20 在 LOC_REL 状态，结点 21、22、23、24、25、26、27 状态如图 4.5 所示。



说明：1 在 TLLI_UNASSIGNED；2、3、4、5、21 在 ADM；6、7、8、9、10、11 在 LOC_EST；12、13、14、15、16、17、18、19 在 ABM；20、22、23、24、25、26、27 在 LOC_REL；

图4.5 LLC 单实例的基于 CEFSM 模型的流图

根据优先级和支配原则，首先选取最大权值的分支路径，也就是有最大分支覆盖的变迁路径，图中是 $0 \rightarrow 1 \rightarrow 6 \rightarrow 12 \rightarrow 20$ ，权值是 4，结点 21-27 权重相同，选取 21，所以第一个变迁路径为 $0 \rightarrow 1 \rightarrow 6 \rightarrow 12 \rightarrow 20 \rightarrow 21$ ，即变迁路径集合为 $\{0 \rightarrow 1 \rightarrow 6 \rightarrow 12 \rightarrow 20 \rightarrow 21\}$ ；重新计算余下结点的优先级，因为不管选结点 2-6 或者 7-11 或者 22-27，都只能增加一个新的结点，所以余下的结点优先级相同，权值也相同，可以随便选择。根据这个原则添加变迁路径到变迁集合里，结果为 $\{0 \rightarrow 1 \rightarrow 6 \rightarrow 12 \rightarrow 20 \rightarrow 21, 0 \rightarrow 1 \rightarrow 6 \rightarrow 12 \rightarrow 20 \rightarrow 22, 0 \rightarrow 1 \rightarrow 6 \rightarrow 12 \rightarrow 20 \rightarrow 23, 0 \rightarrow 1 \rightarrow 6 \rightarrow 12 \rightarrow 20 \rightarrow 24, 0 \rightarrow 1 \rightarrow 6 \rightarrow 12 \rightarrow 20 \rightarrow 25, 0 \rightarrow 1 \rightarrow 6 \rightarrow 12 \rightarrow 20 \rightarrow 26, 0 \rightarrow 1 \rightarrow 6 \rightarrow 12 \rightarrow 20 \rightarrow 27, 0 \rightarrow 1 \rightarrow 6 \rightarrow 12 \rightarrow 14, 0 \rightarrow 1 \rightarrow 6 \rightarrow 12 \rightarrow 15, 0 \rightarrow 1 \rightarrow 6 \rightarrow 12 \rightarrow 16, 0 \rightarrow 1 \rightarrow 6 \rightarrow 12 \rightarrow 17, 0 \rightarrow 1 \rightarrow 6 \rightarrow 12 \rightarrow 18, 0 \rightarrow 1 \rightarrow 6 \rightarrow 12 \rightarrow 19, 0 \rightarrow 1 \rightarrow 6 \rightarrow 7, 0 \rightarrow 1 \rightarrow 6 \rightarrow 8, 0 \rightarrow 1 \rightarrow 6 \rightarrow 9, 0 \rightarrow 1 \rightarrow 6 \rightarrow 10, 0 \rightarrow 1 \rightarrow 6 \rightarrow 11, 0 \rightarrow 1 \rightarrow 6 \rightarrow 13, 0 \rightarrow 1 \rightarrow 2, 0 \rightarrow 1 \rightarrow 3, 0 \rightarrow 1$

→4, 0→1→5}, 共 23 个测试例。

根据第三章的表 3-2 的状态集合、输入/条件集合、任务/输出集合以及相应的变迁集合, 采用前向验证的方法把变迁路径转变为测试例, 其中内部所需的信号在第三章 LLC 的 SDL 规范图中进行了描述。如测试例 0→1→5, 0 代表开始, 所以不用收发任何信号。1 是 B1 分支, 表示 TLLI 分配过程, 需要的外部信号是 LLGMM_ASSIGN_REQ, 起始状态是 TLLI_UNASSIGNED, 收到信号后变为 ADM (尾状态); 5 代表 B5 分支, 代表 GMM 信令的发送, 需要收到的外部信号是 LLGMM_UNITDATA_REQ, 起始状态是 ADM, 尾状态也是 ADM。所以 0→1→5 表示的测试例如下表 4-1 所示。需要两个信号 LLGMM_ASSIGN_REQ、LLGMM_UNITDATA_REQ, 它的类型定义和数据构造 (约束) 参照 4.3。

表4-1 基于 CEFSM 模型的测试生成算法

语句标号	行为描述	约束
1	Chgmm! LLGMM_ASSIGN_REQ	CLLGMM_ASSIGN_REQ
2	Chgmm! LLGMM_UNITDATA_REQ	CLLGMM_UNITDATA_REQ

4.2.2 LLC 基于 CEFSM 模型的测试实现

根据 3.3.4 中描述的变迁路径集合来做测试例, 包括两个测试步 TLLI 分配和 ABM 方式建立, 三个控制观察点 PCO。对 LLC 采用单元测试, 也就是一个模块的测试, 根据 LLC 功能, 它有两种数据传送方式: 证实和非证实; 相应的有 ADM 和 ABM 两种工作方式, 因此对 LLC 的单元测试主要是对这两种操作方式和数据传送方式的验证。有两个测试步过程: TLLI 分配过程和 ABM 建立过程。把 GRR、GMM、SND CP 都当成是环境, 只有一个下测试器, 采用远程测试法, 常用过程的测试例举例如下。

4.2.2.1 ABM 方式建立

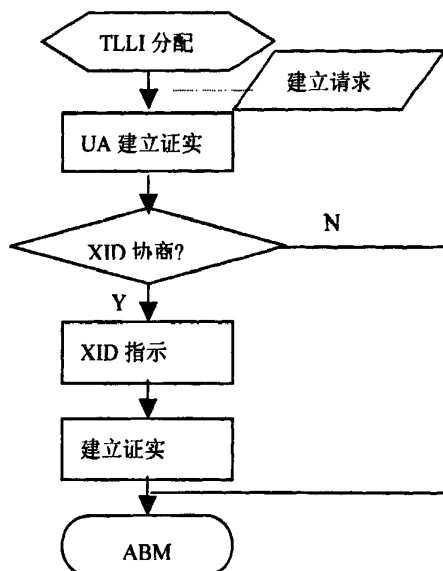


图4.6 ABM 方式建立的测试步过程

在 GMM 分配了 TLLI 后, LLC 进入 ADM 状态就可以接收三层发起的 ABM 建立请求, 就可以进行异步平衡模式的创建。在 LLC 模块里建立请求是通过 SABM (设置异步平衡模式的 U 帧) 来传送的, 等收到对端过来的正确的 UA (Unnumbered Acknowledgement) 帧后, ABM 即建立成功。SABM 可以带三层的 XID 协商参数, 也可以不带, 不带 XID 协商的 MSC 仿真图如附录一所示, TTCN 里的测试步过程如图 4.6 所示。

测试前导是 TLLI 分配过程, 首先是进行 TLLI 分配过程; 然后判断收到的信号的类型, 若是建立请求则在收到 GRR 准备好信号后进行建立请求的发送, 否则一直等待, 直到 GRR 准备好; 发完后等待接收对端的 UA 帧, 收到 UA 看是否有 3 层 XID 协商, 若有则向三层发协商指示信号。

4.2.2.2 TLLI 分配测试步

该测试步相当于是所有 LLC 测试例的测试前导 (preamble), 因为只有分配了 TLLI 后, LLC 进入 ADM 状态, 才能进行数据和话音传送。TLLI 分配后还可以由 GMM 根据需要改变。

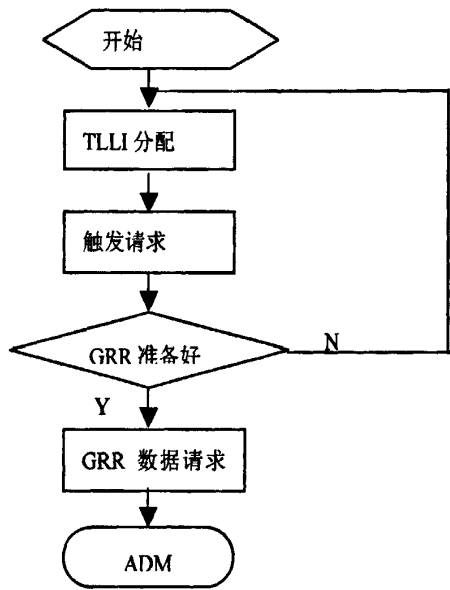


图4.7 TLLI 分配测试步过程

4.2.2.3 ABM 释放

测试前导是 SABM 的建立过程，进入 ABM 状态；首先是 GMM 的附着分离过程；然后接收到 SMDCP 的释放请求，则进行正常释放。

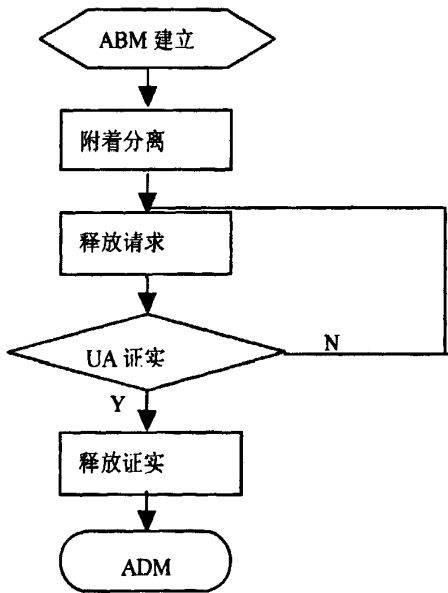


图4.8 ABM 正常释放过程

4.2.2.4 ABM 方式下 UI 数据传送

测试前导是 SABM 的建立过程，进入 ABM 状态；在该状态接收到 SNDCP 的 UI 帧数据请求，收到 GRR 准备后则进行数据发送请求，同时还可以接收对端过来的 UI 帧数据。测试的主要是 ABM 方式下的 UI 帧传送，在非 ABM 状态也可以进行 UI 帧数据传送，UI 帧的接收不需要证实，但可对接收重复与否进行判断，若是重复的帧则抛弃。MSC 如附录二所示。

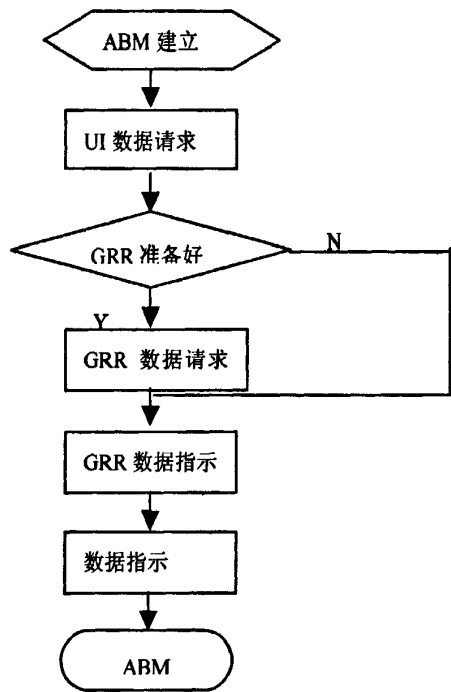


图4.9 ABM 方式下非证实数据的传送

4.2.2.5 ABM 方式下 I 数据传送

测试前导是 SABM 的建立过程，进入 ABM 状态；接收到 SNDCP 的 I 帧请求，则进行 I 帧数据传送，该数据需要对端的确认，也就是等收到对端的确认消息才表示数据正确的传送完成。I 帧的传送需要对端的证实，只有收到对端的确认帧，才表示传送成功；否则的话，若到一定的时间还未收到对端的确认帧，则进行重传，重传的规则根据滑动窗口来实现。滑动窗口采用的是返回 n-ARQ 重发机制。返回 n-ARQ 机制是接收和发送方都采用滑动窗口，滑动窗口大小为 n，LLC 缺省的窗口大小为 n=16 (SAPI=3)，模数 512，发送过程如下：

如果 $(V(S) - V(A)) \bmod 512 < n$

$N(R) := V(R)$; $N(S) := V(S)$;

$V(S) := (V(S) + 1) \bmod 512$;

如果接收到的帧的 $N(R)$ 满足 $(N(R) - V(A)) \bmod 512 \leq (V(S) - V(A))$

$V(A) =$ 接收到的帧 $N(R)$;

如果 $V(A) \neq V(S)$, 则选择一个满足 $(N(S) - V(A)) \bmod 512 < (V(S) - V(A)) \bmod 512$ 的 $N(S)$ 重传;

其中 $N(R)$ 是接收序列号, 表示将要接收的下一帧的发送序号; $N(S)$ 是发送序列号, 表示将要发送帧的序列号; $V(R)$ 是接收状态变量, 表示下一个希望接收的序号内 I 帧的标号; $V(S)$ 是发送状态变量, 表示下一个待发送的序号内 I 帧的编号; $V(A)$ 是应答状态变量, 表示发送窗口的第一帧, $V(A)-1$ 等于最后被应答的 I 帧的 $N(S)$ 。

接收 I 帧数据的过程:

如果接收到的帧的 $N(S)$ 等于 $V(R)$

$V(R) := (V(R) + 1) \bmod 512$;

否则 $V(R) := (\text{接收到的帧的 } N(S) + 1) \bmod 512$;

MSC 图如附录三所示。

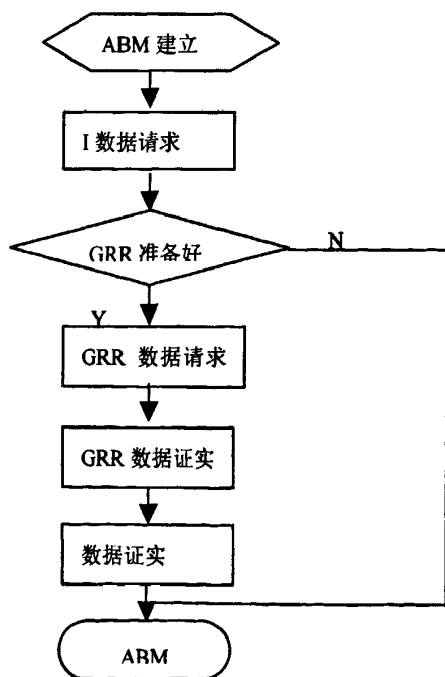


图4.10 ABM 方式下证实数据传送

4.2.2.6 ABM 方式下传送数据时收到信令情况

测试前导是 SABM 的建立过程，进入 ABM 状态；接收到 SNDCP 的 I 帧数据请求，则进行 I 帧数据发送，在没有收到对端的确认时又收到 GMM 的控制信令，则挂起当前的数据发送，进行信令传送；之后再用恢复过程进行 I 数据的传送。

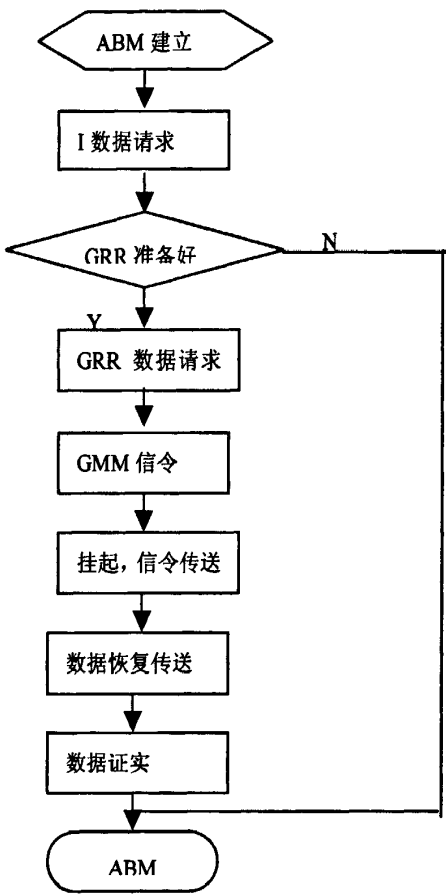


图4.11 ABM 方式下数据与信令的调度

4.2.2.7 重新分配 TLLI 过程的测试

TLLI 可以由 GMM 根据需要进行重新分配(如路由区更新后), 该过程即是测试 LLC 对 GMM 过程的支持，测试前导是 TLLI 分配。

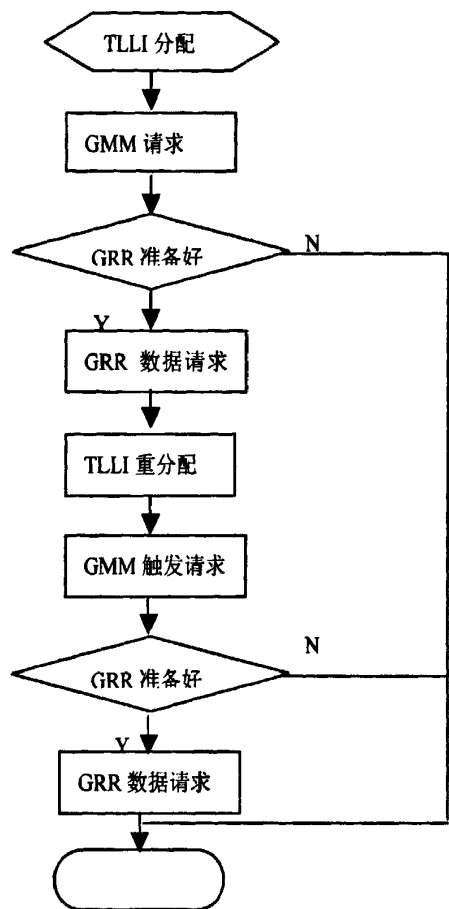


图4.12 TLLI 重分配

4.3 测试数据构造

SDL 和 TTCN 进行协仿真测试时，SDL 通过信道每收发一个信号，TTCN 都会通过 PCO 对 TTCN 里构造的测试数据和 SDL 的收发信号进行观察和比较，当二者一致时，测试例才可以往下继续运行；但一旦遇到不匹配时，TTCN 就会停在不匹配的信号，所以为了能够对要测试的系统进行成功测试，测试数据的构造是很重要的。下面对基本 LLC 测试过程数据构造进行介绍。

4.3.1 ABM 方式的建立

该方式下最重要的是建立请求的 SABM 帧和对等端过来的 UA 帧要匹配，而且 UA 帧必须是刚才所发的同一个 SABM 的 UA 帧，因为 LLC 存在多个实例，所以这两个数据的第二个字节一定要匹配。TTCN 对发出的信号不做检查，但对

收到的信号要和 SDL 里的数据相比较，所以由 LLC 模块发出的信号可以不用验证，数据都填为*，对 LLC 收到的信号数据一定要和 SDL 里匹配。信号一般有信号头和信号结构体，这里只列出了信号结构体如表 4-2。数据结构如下，其中 LL_PDU_LEN 为宏定义，是表示 LLC 信息字段长度。

表4-2 ABM 方式建立用的测试数据（无 XID 协商）

信号名	信号的数据内容
LLGMM_ASSIGN_REQ GMM→LLC	u32 oldTlli 123, u32 newTlli 155, u8 cipher '01'O, u8 gprsKc[8] '03'O,'00'O,'00'O,'00'O,'00'O,'00'O,'00'O,'00'O, v_Filleru8 '00'O/*填充字节*/ v_Filleru16 0
LL_ESTABLISH_REQ SNDTCP→LLC	u32 Tlli 155, /*为刚分配的 TLLI*/ u8 Sapi '03' O, u8 ReqXid[500] '03'O,'F7'O,'00'O../*SAPI=3, 'F7' O 代表是 SABM 帧，后面为 497 个字节的 '00' O*/
GRR_READY_TO_SEND GRR →LLC	u8 sapi '03' O, u8 Cvflag '01' O, /*???*/ u8 radioPriority '01' O, v_Filleru8 '00'O
GRR_DATA_REQ GRR←LLC	u32 TLLI *, u8 LL_PDU[LL_PDU_LEN] *, u16 llcLength *, u16 remainLength *, u8 sapi *, u8 cause *, t_Qos Qos_param *, u8 radio_priority *, u8 llcPduType *
LLGMM_TRIGGER_IND GMM←LLC	u32 tlli *
GRR_DATA_IND GRR →LLC	u32 TLLI 155, u16 length 155, u8 cause '05' O, u8 LL_PDU[LL_PDU_LEN] '03'O,'F7'O,'00'O../*SAPI=3, 'F6' O 代表是 UA 帧，后面为 497 个字节的 '00' O*/
LL_ESTABLISH_CNF SNDTCP←LLC	u32 Tlli *, u8 Sapi *, u16 N201_I *, u16 N201_U *,

	u16	negoXidLength *
	u32	dataPointer *

4.3.2 ABM 方式 I 数据传送

测试前导是 ABM 建立，因此前面的建立过程数据跟上述的相同，表 4-3 只列出 I 帧数据发送要用到的信号。

表4-3 ABM 方式下 I 数据发送

信号名	信号的数据内容
LL_DATA_REQ SND CP→LLC	u32 Tlli 155, u8 Sapi '03' O, u16 N_PDULength 34, u8 Reference '02' O, u8 QoS[4] '01'O,'02'O,'03'O,'04'O,, u8 radioPriority '02' O, u16 snPDULength 22, u8 SN_PDU[100] '03'O,'F6'O ,'01'O,'0E'O,'84'O,'10'O, '01'O,'02'O,'03'O,'04'O ,'88'O,'10'O.../*'F6'O的F表示要求对端证实*/
GRR_READY_TO_SEND GRR →LLC	u8 sapi '03' O, u8 Cvflag '01' O, /*???*/ u8 radioPriority '01' O, v_Filleru8 '00'O
GRR_DATA_REQ GRR←LLC	u32 TLLI *, u8 LL_PDU[LL_PDU_LEN] *, u16 llcLength *, u16 remainLength *, u8 sapi *, u8 cause *, t_Qos Qos_param *, u8 radio_priority *, u8 llcPduType *
LLGMM_TRIGGER_IND GMM←LLC	u32 tlli *
GRR_DATA_IND GRR →LLC	u32 TLLI 155, u16 length 155, u8 cause '01' O, u8 LL_PDU[LL_PDU_LEN] 05, 00, 01, 0E, 84, 10, 01, 02, 03, 04, 88, 10, 01, 02, 03, 04, 0C, 01, 02, 10, 0E, 14, 01, 02, 18, 00, 01, 1C, 00, 01, 20, 00, 01, 24, 01, 28,

	01, 2C, EE, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, /*后面为 '00' /
LL_DATA_IND SND CP←LLC	u32 Tlli 155, u8 Sapi '01'O, u16 snPDULength 400 u8 SN_PDU[100] 0E, 84, 10, 01, 02, 03, 04, 88, 10, 01, 02, 03, 04, 0C, 01, 02, 10, 0E, 14, 01, 02, 18, 00, 01, 1C, 00, 01, 20, 00, 01, 24, 01, 28, 01, 2C, EE, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, /* 后面为 '00' /

4.3.3 ABM 方式下 UI 数据传送

测试前导是 ABM 建立，因此前面的建立过程数据跟上述的相同。包括发送和接收 UI 帧数据，为了测试，测试前导后面先是 I 帧数据的发送，I 帧数据内容同上述，然后是 UI 帧数据的发送和接收，表 4-4 只列出 UI 帧数据发送和接收。

表4-4 ABM 方式下 UI 数据发送和接收

信号名	信号的数据内容
LL_UNITDATA_REQ SND CP→LLC	u32 Tlli 155, u8 Sapi '03' O, u16 N_PDULength 55, u8 Reference '02' O, u8 QoS[4] '01'O,'02'O,'03'O,'04'O,, u8 radioPriority '03' O, u8 cipher '03'O, u16 snPDULength 66, u8 SN_PDU[100] '03'O,'C0'O,'03'O,'C0'O,'01'O, '0E'O,'84'O,'10'O,'01'O,'02'O,'03'O,'04'O,'88'O,'10'O... /*F6'O 的 F 表示要求对端证实*/
GRR_READY_TO_SEND GRR →LLC	u8 sapi '03' O, u8 Cvflag '01' O, /*???*/ u8 radioPriority '01' O, v_Filleru8 '00'O
GRR_UNITDATA_REQ GRR←LLC	u32 TLLI *, u8 LL_PDU[LL_PDU_LEN] *, u16 llcLength *, u16 remainLength *, u8 sapi *, u8 cause *, t_Qos Qos_param *,

	u8 radio_priority *, u8 llcPduType *
GRR_UNITDATA_IND GRR →LLC	u32 TLLI 155, u16 length 155, u8 cause '01' O, u8 LL_PDU[LL_PDU_LEN] '03'O,'C0'O,'03'O,'C0'O,'01'O, '0E'O,'84'O,'10'O, '01'O,'02'O,'03'O,'04'O ,'88'O,'10'O /*后面为 '00' /
LL_UNITDATA_IND SND CP←LLC	u32 Tlli 155, u8 Sapi '01'O, u16 snPDULength 400 u8 SN_PDU[100] '03'O,'C0'O,, '01'O,'0E'O,'84'O,'10'O, '01'O,'02'O,'03'O,'04'O ,'88'O,'10'O /*后面为*/ u8 cipher 'FF'O

4.3.4 ABM 方式的释放

ABM 的释放前导是 ABM 方式的建立，然后是附着分离，之后是释放过程
如表 4-5。

表4-5 ABM 方式的释放

信号名	信号的数据内容
LLGMM_UNITDATA_REQ GMM→LLC	u8 sapi '01' O, u8 length 'FE' O, u32 tlli 155 u8 message[MAX_MESSAGE_LENGTH] '01'O,'F4'O,'FF'O, v_Filleru8 '00'O v_Filleru16 0
GRR_READY_TO_SEND GRR →LLC	u8 sapi '03' O, u8 Cvflag '01' O, /*???*/ u8 radioPriority '01' O, v_Filleru8 '00'O
GRR_DATA_REQ GRR←LLC	u32 TLLI *, u8 LL_PDU[LL_PDU_LEN] *, u16 llcLength *, u16 remainLength *, u8 sapi *, u8 cause *, t_Qos Qos_param *, u8 radio_priority *, u8 llcPduType *

LLGMM_TRIGGER_IND GMM←LLC	u32 tlli *
GRR_DATA_IND GRR →LLC	u32 TLLI 155, u16 length 155, u8 cause '03' O, u8 LL_PDU[LL_PDU_LEN] '03'O,'F4'O, 'FF'O,'FF'O,.../*后面为 '00' /
LLGMM_UNITDATA_IND GMM←LLC	u8 sapi '01'O, u8 length 'FE'O, u32 tlli 155, u8 message[MAX_MESSAGE_LENGTH] *, v_Filleru8 '00'O v_Filleru16 0
LL_RELEASE_REQ SNDGP→LLC	u32 Tlli 155, /*为刚分配的 TLLI*/ u8 Sapi '03' O, u8 Local '01'O
GRR_READY_TO_SEND GRR →LLC	u8 sapi '03' O, u8 Cvflag '01' O, /*???*/ u8 radioPriority '01' O, v_Filleru8 '00'O
GRR_DATA_REQ GRR←LLC	u32 TLLI *, u8 LL_PDU[LL_PDU_LEN] *, u16 llcLength *, u16 remainLength *, u8 sapi *, u8 cause *, t_Qos Qos_param *, u8 radio_priority *, u8 llcPduType *
LLGMM_TRIGGER_IND GMM←LLC	u32 tlli *
GRR_DATA_IND GRR →LLC	u32 TLLI 155, u16 length 155, u8 cause '05' O, u8 LL_PDU[LL_PDU_LEN] '03'O,'F6'O,'00'O,'FF'O,'FF'O../*SAPI=3, 'F6' O 代表是 UA 帧, 后面为 497 个字节的 '00' O*/
LL_RELEASE_CNF SNDGP←LLC	u32 Tlli *, u8 Sapi '03'O

4.4 协议测试结果分析

对 LLC 的测试，本文只列举了正常情况下的测试例，不包括异常情况。把

TTCN 测试得到的信号流图与按照规范设计的信号流图比较,发现在 TTCN 测试和仿真软件模块之间的信号交互是基本一致的。测试例中的数据与通过 TTCN 仿真的 MSC 图所产生的数据也基本是一致的。只有一些填充数据、动态数据和随机数据上不一致(例如信号头中的数据),这是因为不能在测试例中设定这些动态数据,而这属于软件设计的正常现象,可见仿真结果达到了预期的目的。由 TTCN 产生 MSC 图的信号和数据参见附录。

对 TLLI 分配、ABM 建立、ABM I 数据传送、ABM UI 数据传送、ABM 释放五个测试例分别命名为 A、B、C、D、E,它们一共可以发现 10 种类型的错误,分别如表 4-5 所示,采用 APFD^{[37][38][39]} (average of the percentage of faults detected)的方法来绘制纠错概率图,如图 4.13、4.14 所示,由图可见通过改变测试例的顺序,可以改变纠错的概率。图中是对平均检测错误概率 APFD 的计算,横轴为测试例的百分比,纵轴为发现错误的百分比,整个区域面积为 1,发现错误的概率为阴影部分的面积;也就是整个区域面积为 1,阴影部分的面积则为 APFD 的数值。

表4-6 测试例和可发现的错误

测试例	错误									
	1	2	3	4	5	6	7	8	9	10
A	×				×					
B	×				×	×	×			
C	×	×	×	×	×	×	×			
D					×					
E								×	×	×

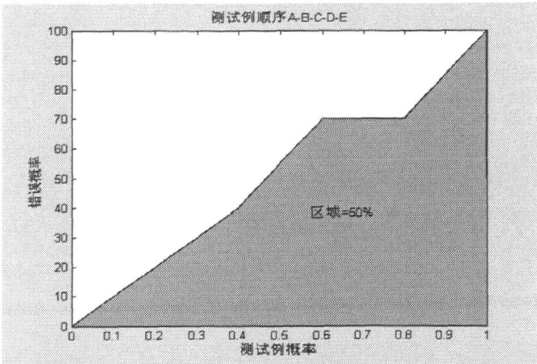


图4.13 测试例的平均纠错概率 APFD 为 50%

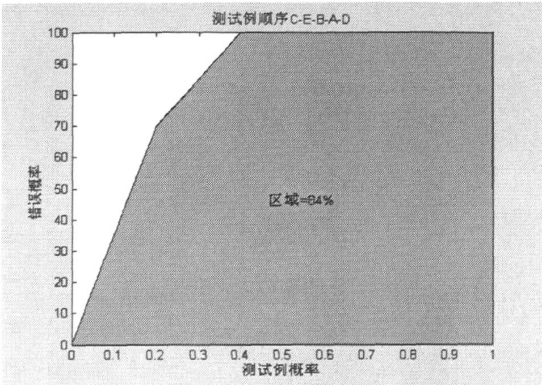


图4.14 测试例的平均纠错概率 APFD 为 84%

4.5 本章小结

本章主要介绍了 TTCN 的使用和对 TD-PRS 的 LLC 模块进行一致性测试方法，采用的是远程测试法，基于 CEFSM 的思想来生成测试例，着重对 LLC 的基本正常过程 SABM 的建立、释放、I 帧数据、UI 帧数据的传送，以及该模式下数据和信令的优先级调度管理进行了介绍。附录里给出了几个 TTCN 和 SDL 协仿真的 MSC 数据流图。

第五章 结束语

本文对协议一致性测试、协议测试采用的有限状态机模型和基于 CEFSM 模型的测试例自动生成方法进行了重点研究；采用远程测试法，对 3G 文档非接入层的测试结构进行了优化。运用本文的方法，完成了对 TD-PRS 终端的 2、3 层的协议仿真和一致性测试，发现了协议开发中的错误，缩小了协议开发的周期，为协议软件推向市场奠定了可靠的基础。同时，该方法对其它通信协议的仿真和测试有重要的参考价值和实用意义；同样也可以把该方法用于 4G 高层协议的仿真和测试上。

本文对通信协议测试方法进行了研究，通过对 TD-PRS 的 LLC 的实现和一致性测试发现，TTCN 具有容易实现，操作直观的特点，但 TTCN 对时间的支持不是很好，不能很好的实时跟踪实际通信系统，而且做测试例比较繁琐，虽然有 AUTOLINK，但对生成的测试例的约束部分还需进一步的修改；实际上采用多的还是用 TTCN 的 LINK，因此我认为下一步的研究工作是基于时间^[40]的重点考虑，对采用 CEFSM 模型的并发性系统开发出自动生成测试例的代码，以和 TTCN 结合对并发性实时通信系统进行协议测试。

致谢

首先我要衷心的感谢我的导师郑建宏教授，本论文的选题、研究和论文撰写各个阶段都凝聚着导师的心血。在三年的学习和科研工作中，郑老师渊博的学识，严谨的治学态度，以及一身作责的典范，都将使我受益终生，是他对我的悉心指导才使得本课题能顺利地完成。

其次，我要特别感谢重邮信科公司研发二部的李小文教授，自从我进实验室以来一直在李老师的组里。李老师在科研工作的具体环节上给了我更详尽、耐心地指导，本课题倾注李老师的大量心血，同时李老师的严谨治学态度、积极进取的工作作风和无私的奉献精神也一直激励着我。

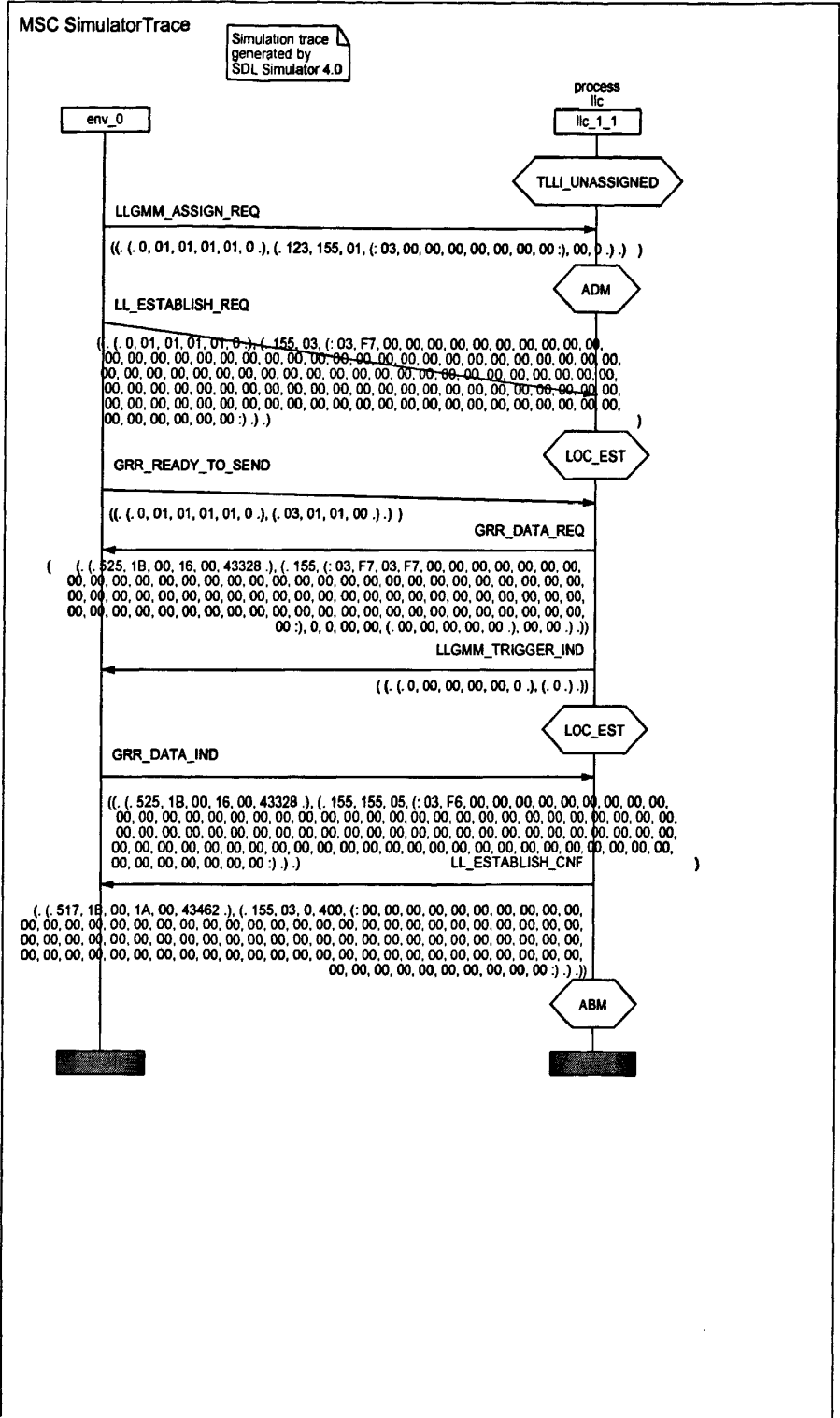
我还要感谢重邮信科公司研发二部的各位老师——申敏教授、陈发堂副教授、段红光副教授、黄俊伟副教授、陈贤亮、彭大芹和朱红专、何先军等老师，感谢他们给我的帮助和建议。同时感谢高层实验室各位同学和老师，他们给我提出了不少建议。

感谢为审阅本论文付出艰辛劳动的各位专家和教授。

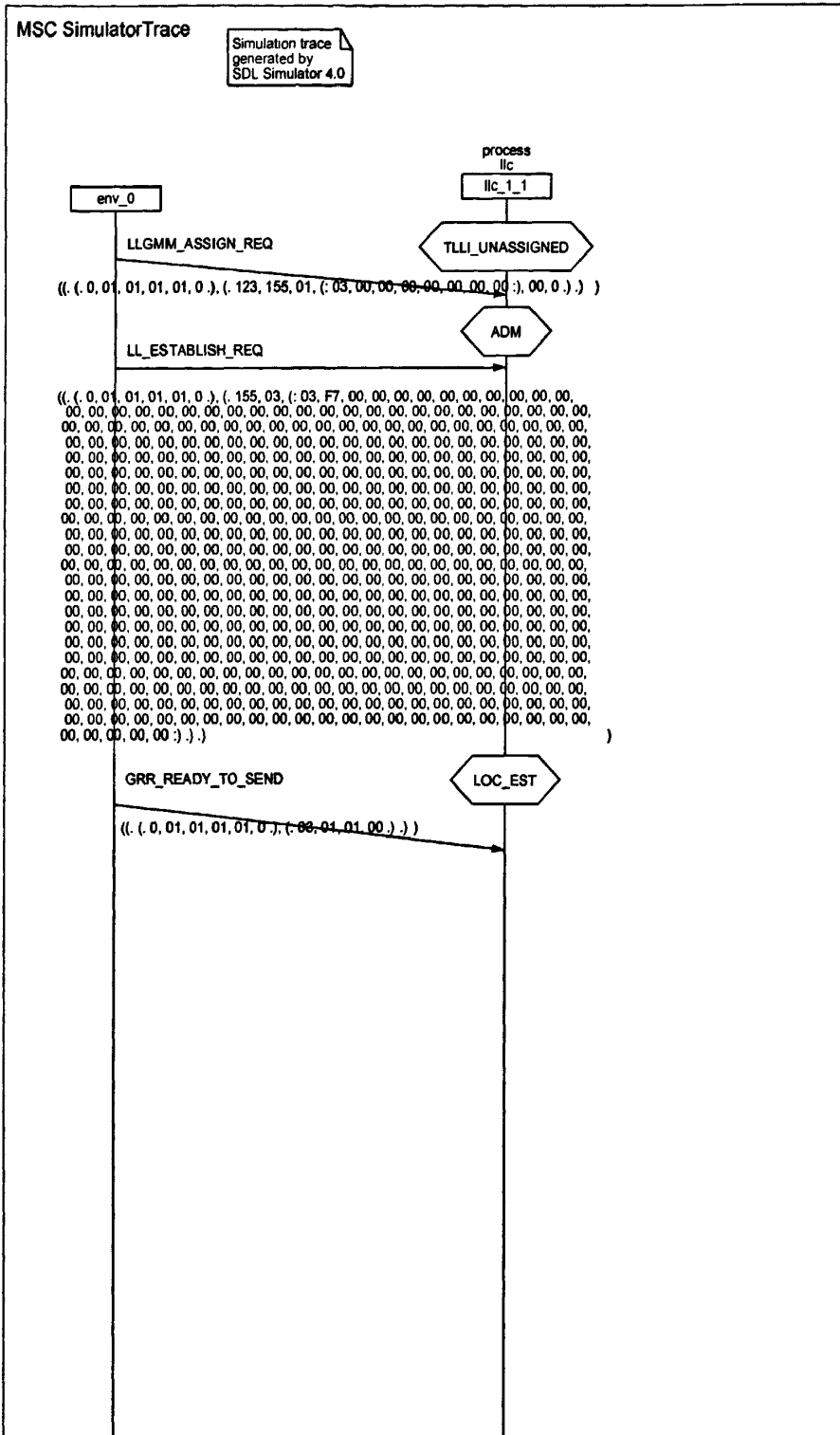
最后，我还要特别感谢我的父母和姐弟，他们一直以来在生活、学习和工作方面给予我无私的奉献、关心和支持

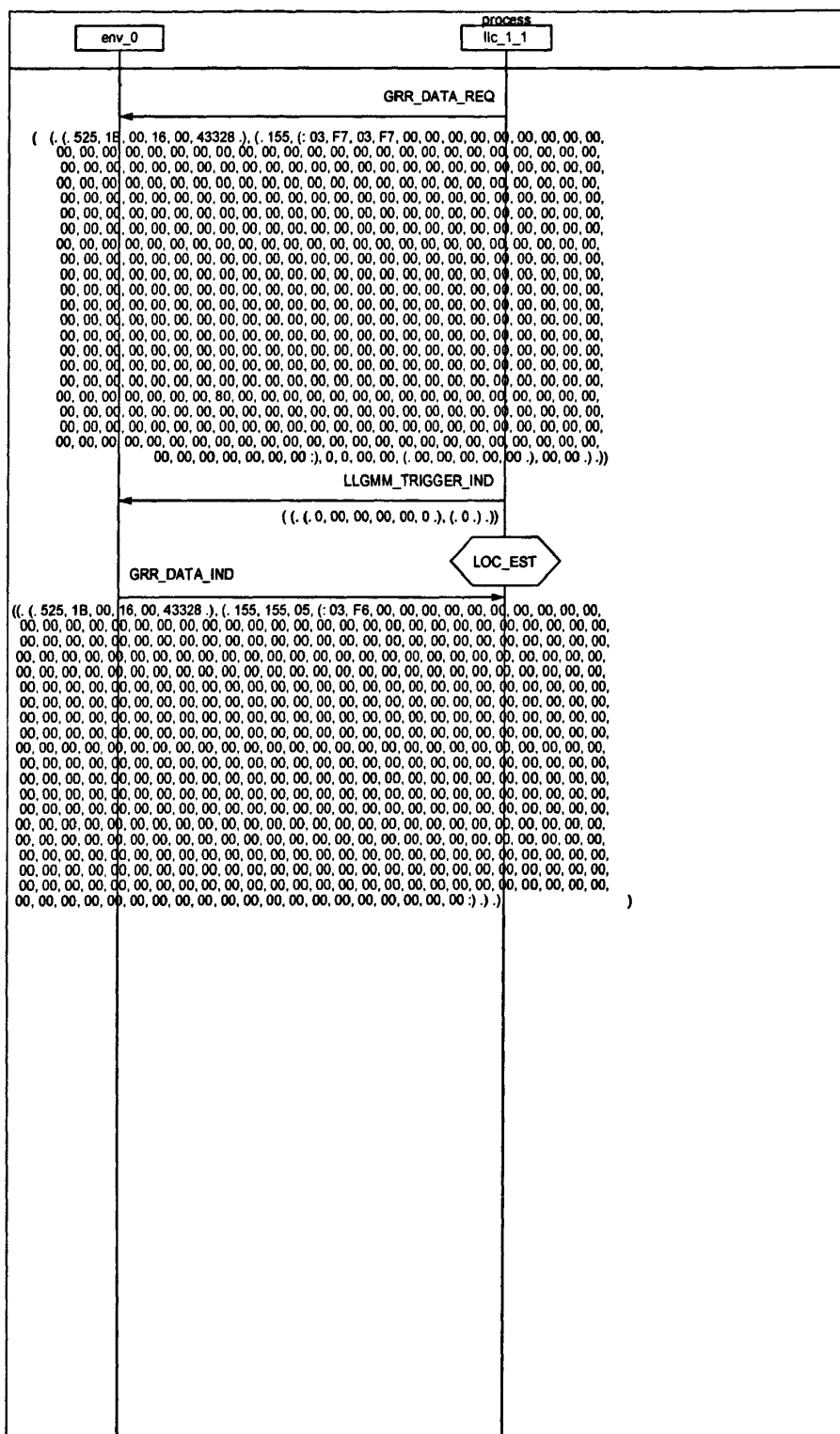
附录 LLC 测试的 MSC 图

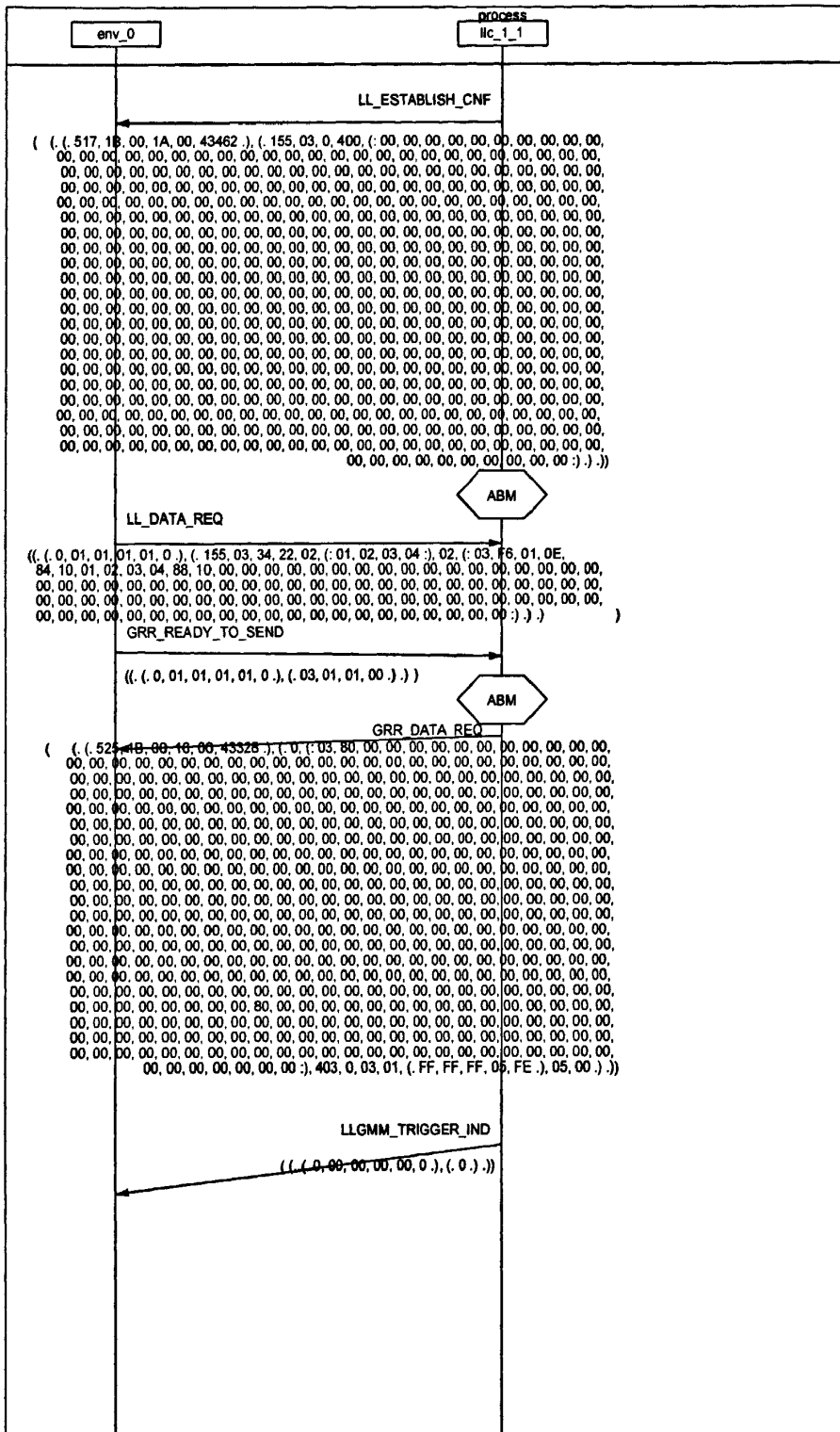
附录一 ABM 方式建立（没有 XID 协商情况）

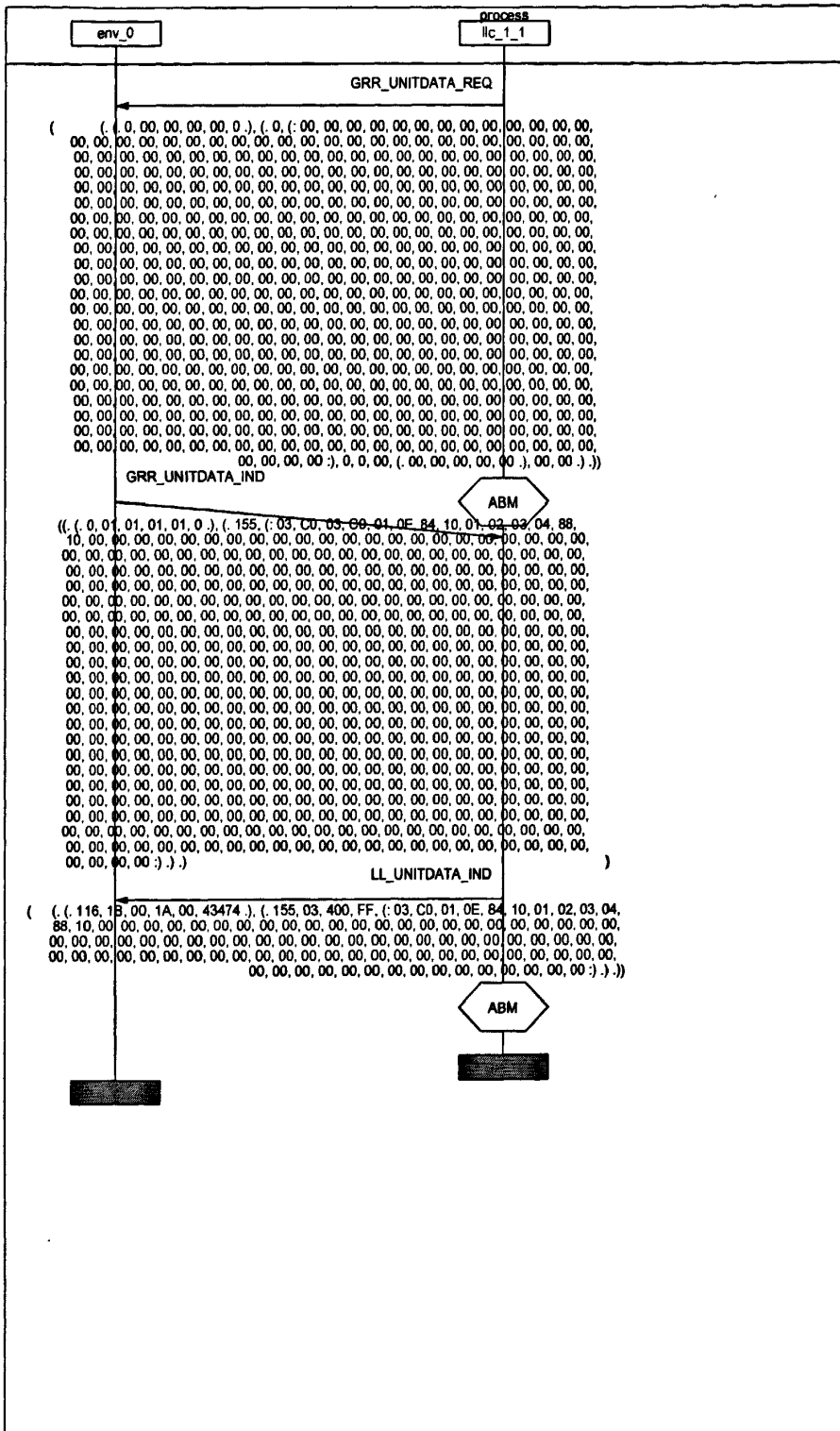


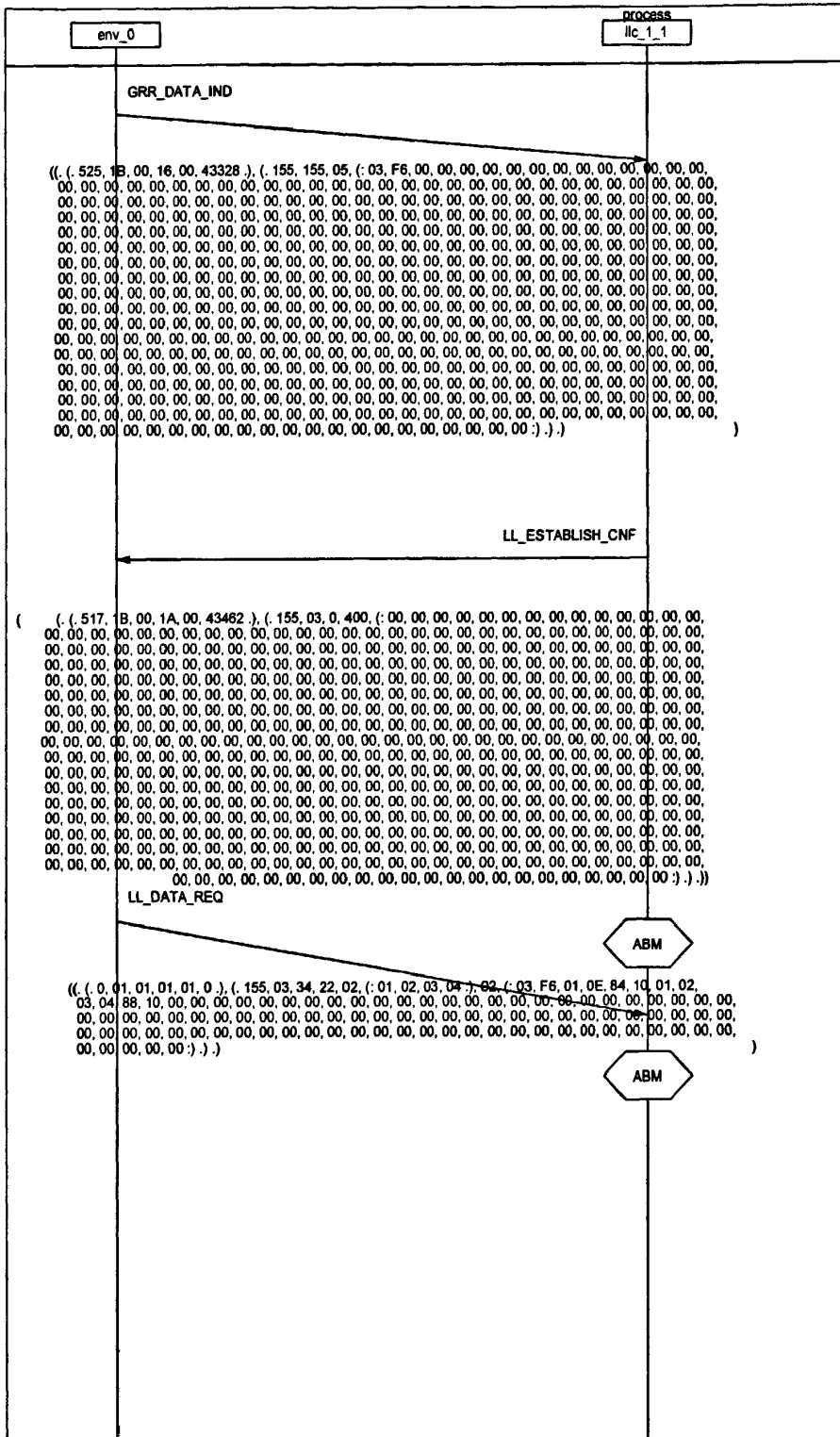
附录二 ABM 方式下 UI 数据传送

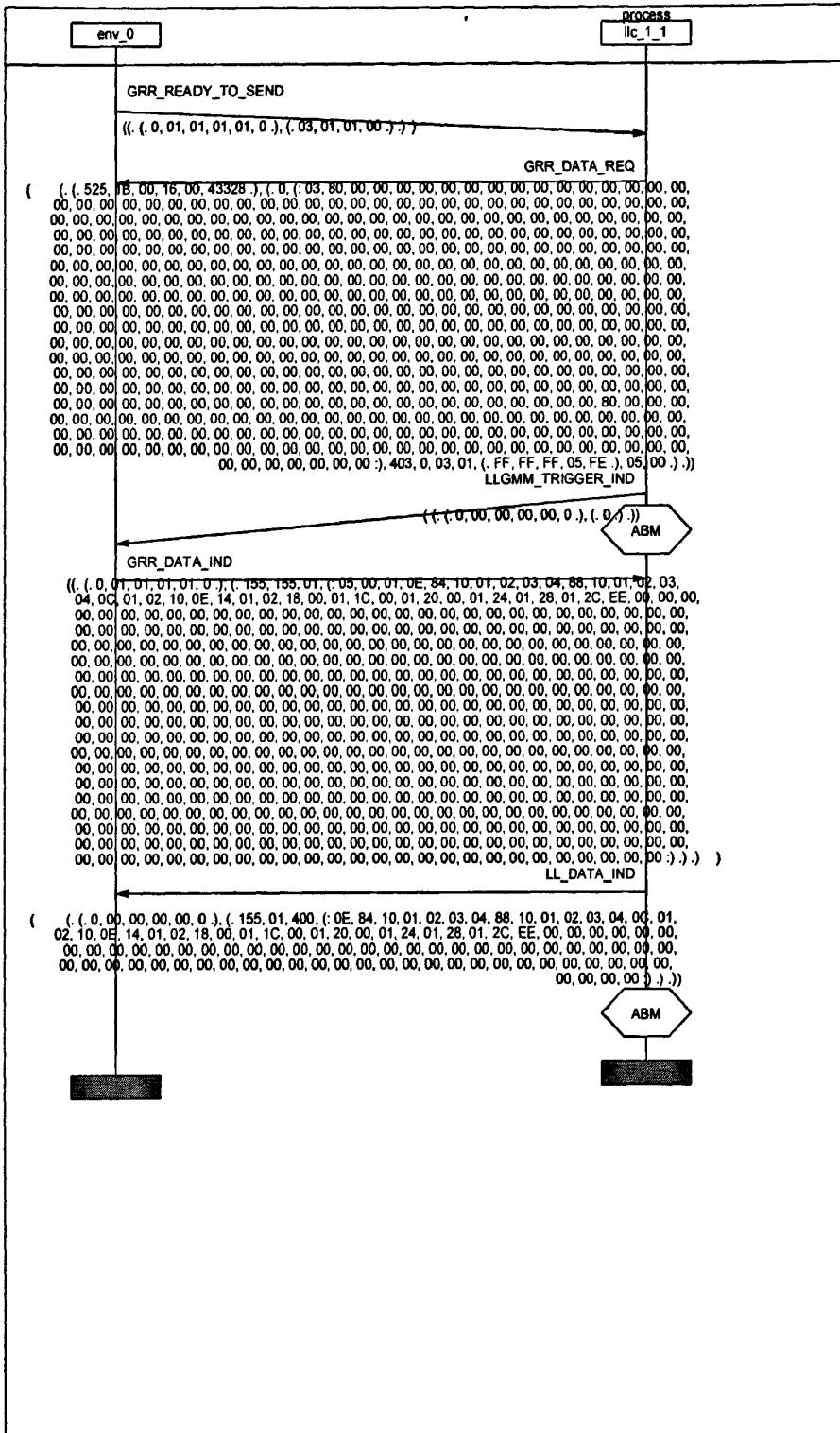












参考文献

- [1] 贾凤君, 杨明洪, 郑建宏, 第三代移动通信TD-PRS的逻辑链路控制层, 自然科学版, vol. 14, No. 3, 2002年9月
- [2] 杨明洪, 贾凤君, 杨济安, MPEG-4视频容错编码在3G移动多媒体终端的应用, 自然科学版, vol. 14, No. 4, 2002年12月
- [3] 贾凤君, 宋怀波, 郑建宏, SDL语言及其在TD-PRS终端开发中的应用, 自然科学版, vol. 15, No. 2, 2003年6月
- [4] Piyu Tripathy and Behcet Sarikaya, Test Generation from LOTOS Specifications, IEEE TRANSACTIONS ON COMPUTERS, VOL. 40, NO. 4, APRIL 1991
- [5] M. A. Feckoa, M. U. Uyurb, P. D. Amera, A. S. Sethia, T. Dzikc, R. Menellc, M. McMahonc, A success story of formal description techniques: Estelle specification and test generation for MIL-STD 188-220, Computer Communications 23 (2000) 1196 - 1213
- [6] Alfred V. Aho , Anton T. Dahbura , David Lee, M. Umit Uyar , An Optimization Technique for Protocol Conformance Test Generation Based on UIO Sequences and Rural Chinese Postman Tours, AT&T Bell Laboratories
- [7] 田军, IPv6 协议一致性测试研究及测试系统实现, 中国科学院博士学位研究生学位论文, 2001
- [8] KSHIRASAGAR NAIK, BEHCET SARIKAYA, Testing Communication Protocols, IEEE SOFTWARE, January 1992
- [9] DEEPINDER P. SIDHU, TING-KAU LEUNG, Formal Methods for Protocol Testing: A Detailed Study, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING. VOL. 15, NO. 4. APRIL 1989
- [10] Ana R. Cavallia, Jean-Philippe Favreaub, Marc Phalippouc, Formal Methods in Conformance Testing: Result and Perspectives, Institut National des Télécommunications, France Telecom-CNET LAA/SLC/EVP and the National Institute of Standards and Technology
- [11] Yoshiaki Kakuda, Hideki Yukitomo, Shinji Kusumoto, Tohru Kikuno , Localizing multiple faults in a protocol implementation, IEEE DESIGN & TEST COMPUTERS, IEEE Fall 1995
- [12] Raymond E. Miller, Sanjoy Paul, On the Generation of Minimal-Length Conformance Tets for Communication Protocols, IEEE/ACM TRANSACTIONS ON NETWORKING. VOL 1 NO. 1, FEBRUARY 1993
- [13] Susumu Fujiwara, Gregor v. Bochmann, Ferhat Kbendek, Mokhtar Amalou, and Abderrazak Ghedamsi, Test Selection Based on Finite State Models, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 17, NO. 6, JUNE 1991
- [14] Susumu Fujiwara, Gregor v. Bochmann, Ferhat Kbendek, Mokhtar Amalou, and Abderrazak Ghedamsi , Test Selection Based on Finite State Models, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 17, NO. 6, IEEE JUNE 1991 591
- [15] Chourouk Bourhfir, Rachida Dssouli and El Mostapha Aboulhamid, Automatic Test Generation for EFSM-based Systems, Université de Montréal, Dept. d' Informatique et

de Recherche Operationnelle, Montréal, H3C 3J7, Canada

- [16] J. Jenny Li, W. Eric Wong, Automatic Test Generation from Communicating Extended Finite State Machine (CEFSM)-Based Models, Proceedings of the Fifth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, IEEE 2002
- [17] Gang Luo, Anindya Das, and Gregor v. Bochmann, Software Testing Based on SDL Specifications with Save, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 20. NO. 1, IEEE JANUARY 1994
- [18] A Arun Kunnar, Conformance Testing of Protocols Represented as Communicating Finite State Machines, B.Tech Project Report, bachelor paper
- [19] C. Bourhfir, E. Aboulhamid, R. Dssouli, N. Rico, A test case generation approach for conformance testing of SDL systems, Nortel grant
- [20] David Lee, Mihalis Yannakakis, Testing Finite-State Machines: State Identification and Verification, IEEE TRANSACTIONS ON COMPUTERS, VOL. 43, NO. 3, MARCH 1994
- [21] Sebastian Elbaum, Alexey G. Malishevsky, Gregg Rothermel, Test Case Prioritization: A Family of Empirical Studies, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 28, NO. 2, FEBRUARY 2002
- [22] Eliane Martins, Selma B. Sabião, Ana Maria Ambrosio, ConData: a Tool for Automating Specification-based Test Case Generation for Communication Systems, Proceedings of the 33rd Hawaii International Conference on System Sciences - 2000, 2000 IEEE
- [23] Telelogic Tau 4.0 User's Manual - February 2000
- [24] Paul Baker, Jens Grabowski, Ekkart Rudolph, Ina Schieferdecker, A Message Sequence Chart-Profile for Graphical Test Specification, Development and Tracing - Graphical Presentation Format for TTCN-3
- [25] Robert L. Probert, Alan W. Williams, FAST FUNCTIONAL TEST GENERATION USING AN SDL MODEL
- [26] Lars Mats, Selection Criteria for Automated TTCN Test Case Generation from SDL
- [27] CWTS TSM Service description of the TD-SCDMA Packet Radio Service 03.60 V2.0.0, 2001-7
- [28] CWTS TSM Overall description of the TD-PRS radio interface 03.64 V2.0.0, 2001-7
- [29] ETSI TS 101 351 Mobile Station-Serving GPRS Support Node Logical Link Control Layer specification, GSM 04.64, 1999-08
- [30] European Telecommunications Standards Institute, Methods for Testing and Specification (MTS); Guide for the use of the second edition of TTCN, EG 201 148 V1.1.2 (1998-03)
- [31] Ruibing Hao and J&ping Wu, Toward Formal TTCN-based Test Execution, IEEE 1997
- [32] Zhen Ru Dai, Jens Grabowski, Helmut Neukirchen, TIMED TTCN-3 - A Real-time extension for TTCN-3, SIIM-TR-A-01-14, SIIM Technical Report

- [33] INA SCHIEFERDECKER , JENS GRABOWSKI, Conformance Testing with TTCN, Teletronikk 4, 2000
- [34] Beat Koch, Jens Grabowski, Dieter Hogrefe, Michael Schmitt, —Autolink —, A Tool for Automatic Test Generation from SDL Specifications
- [35] ETR/ATM-1002, Advanced Testing Methods (ATM); Tutorial on protocol conformance testing, Especially OSI standards and profiles
- [36] CWTs 3GPP, 3rd Generation Partnership Project; Technical Specification Group Terminal; User Equipment (UE) conformance specification; Part 3: Abstract Test Suites, TS 34.123-3, V1.0.3 2001-05
- [37] Yuri Chernak, Valley Forge Consulting , Validating and Improving Test-Case Effectiveness, IEEE SOFTWARE, January/February 2001
- [38] Sebastian Elbaum, Alexey G. Malishevsky, and Gregg Rothermel, Test Case Prioritization: A Family of Empirical Studies, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 28, NO. 2, FEBRUARY 2002
- [39] David Lee, Mihalis Yannakakis , OPTIMIZATION PROBLEMS FROM FEATURE TESTING OF COMMUNICATION PROTOCOLS, Proceedings of the 1996 International Conference on Network Protocols (ICNP '96), IEEE 1996
- [40] Abdeslam En-Nouaary, Rechida Dssouli, Ferhat Khendek, Timed Wp-Method: Testing Real-Time Ssystems, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, vol.28, NO 11, Novebember 2002

攻读硕士学位期间

参加的科研项目及发表的论文

1、 发表的论文

贾凤君, 杨明洪, 郑建宏, 《第三代移动通信 TD-PRS 的逻辑链路控制层》, 自然科学板, vol. 14, No.3, 2002 年 9 月;

杨明洪, 贾凤君, 杨济安, 《MEG-4 视频容错编码在 3G 移动多媒体终端的应用》, 自然科学板, vol. 14, No.4, 2002 年 12 月;

贾凤君, 宋怀波, 郑建宏, 《SDL 语言及其在 TD-PRS 终端开发中的应用》, 自然科学板, vol. 15, No.2, 2003 年 6 月;

2、 参加的科研项目

2001 年 2 月—2002 年 1 月, 北京大唐移动通信设备公司 3G 终端高层开发;

2002 年 3 月—2003 年 6 月, 重邮信科研发二部手机高层信令开发