doi:10.3772/j.issn.1002-0470.2015.02.004

基于 GMR-1 的卫星移动通信系统模拟信关站物理层软件的实现①

朱 哲②* 周 洁* 刘 洋* 高 镇③* 李 忻** 赵 明** 王 京**

(*天津大学电子信息工程学院 天津 300072)

(**清华大学信息科学与技术国家实验室 北京 100084)

摘 要 根据地球同步卫星移动通信接口标准 GMR-1 Release 1,使用软件无线电构架设计了卫星移动通信系统模拟信关站物理层软件,用于测试和验证 GMR-1 Release 1 卫星移动通信终端的物理层协议一致性。该软件综合考虑了处理效率和系统灵活性,采取用 Python 脚本调用 C 函数的工作机制实现 GMR-1 Release 1 的各种逻辑信道的收发功能,以满足多种测试需求。环回自测和终端联测证实了所开发系统的功能正确性和测试有效性。

关键词 卫星移动通信,模拟信关站,GMR-1 Release 1,软件无线电,Python

0 引 言

相比地面蜂窝移动通信,卫星移动通信具有覆盖范围广、不受地理条件和通信距离限制等优点,是地面移动通信的必要补充。卫星移动通信系统是利用高、中、低等不同轨道卫星实现区域乃至全球范围的移动通信业务,主要提供话音、数据等通信服务。根据卫星所处轨道不同可将卫星移动通信系统分为三类:低轨(low earth orbit, LEO)卫星系统,中轨(medium earth orbit, MEO)卫星系统,高轨(geostationary earth orbit, GEO)卫星系统。相比 LEO、MEO 卫星,GEO 卫星移动通信系统技术复杂度较低、建设周期短、成本低、见效快,具有较高的社会效益和经济价值。我国申请的移动通信卫星轨位即将到期,面临收回的压力,因此我国已确定优先发展基于 GEO 卫星的移动通信系统[1-6]。

GMR-1 是欧洲电信标准委员会(ETSI)制定的地球同步轨道移动无线接口(geostationary earth orbit mobile radio interface)规范,是为数不多的几个公开的卫星移动通信空口标准之一,对我国第一代卫星移动通信系统的建设具有较高的参考价值。2001 年,对应于地面蜂窝全球移动通信系统(GSM),ETSI 提出了卫星空口技术标准 GMR-1 R1(GMR-1 Release 1)[7],在此之后,分别对应于通用分组无线服务(GPRS)和 3G,又相继推出了 GMR-1 R2 和 GMR-1 3G。GMR-1 标准基于频分双工(FDD),采用时分多址接入/频分多址接入(TDMA/FDMA)模式。最新的 GMR-1 3G 发布于 2011 年,基于 3GPP R6 协议架构,支持与 3G 地面核心网连接,最高吞吐量可达 592kbps。基于 GMR-1 R1 标准的典型系统是 Thuraya 系统。作为区域性卫星移动通信系统,Thuraya 系统目前在轨 3 颗 GEO 卫星,覆盖全球 2/3 地区,向包括欧洲、非洲、中东以及中南亚在内的 140 个国家提供服务,到去年已经有 26 万用户数量。

为了深入理解 GMR-1 体制,为我国"十二五"863 项目"卫星移动通信系统关键技术研究"之子课题 7"第一代卫星移动通信系统终端关键技术研究"和子课题 8"第一代卫星移动通信系统信关站(gateway station, GS)与测试验证系统关键技术研究"提供参考,本文根据 GMR-1 R1 标准,设计了一套信关

① 863 计划(2012AA01A502)资助项目。

② 女,1989 年生,硕士;研究方向:卫星移动通信物理信道研究;E-mail: zhuzhe@tju.edu.cn

③ 通迅作者,E-mail: zgao@mail.tsinghua.edu.cn

(收稿日期:2014-08-22)

站物理层基带信号处理模拟设备。设计方案采用基于 PC 的“Python + C”软件无线电实现架构:使用 C 语言实现复杂度较高的关键算法,保证高效的数据处理;使用 Python 调用 C 模块的方式实现逻辑信道的发送和接收功能,保证系统开发的效率和灵活性。所研究的信关站模拟器可用于 GMR-1 终端的物理层一致性测试,并为上层协议一致性测试提供物理层基础。

1 GMR-1 R1 空口物理层简介

ETSI 制定的地球同步卫星移动通信标准(GMR-1 R1)兼容全球移动通信系统(GSM)标准,是少数几个公开的卫星移动通信系统空口标准之一。

1.1 GMR-1 R1 基本系统参数

GMR-1 R1 系统将有效带宽 34MHz 分成 1087 个载频,每个载频间隔为 31.25kHz,单个载频符号速率为 23.4ksps。GMR-1 R1 是时分复用系统,系统采用了相对简单的帧结构,以时分多址接入(TDMA)帧为单位,每帧 24 个时隙(40ms),一个复帧包含 16 个 TDMA 帧,一个超帧包含 4 个复帧(64 个 TDMA 帧),一个超高帧包含 4896 个超帧(313344 个 TDMA 帧)[8,9]。

1.2 信道类型及编码调制方式

GMR-1 R1 系统逻辑信道分为业务信道和控制信道[10,11]。

业务信道主要包括:

- TCH3:3 时隙突发,语音通信;
- TCH6:6 时隙突发,2.4/4.8kbps 的数据/传真传输;
- TCH9:9 时隙突发,2.4/4.8/9.6kbps 的数据/传真传输。

控制信道主要包括:

- FCCH:8 时隙突发,用于承载频率校正信息,由地面终端接收并进行频率校正;
- BCCH:8 时隙突发,用于向地面终端广播系统消息;
- PCH:8 时隙突发,单向下行链路,用于寻呼地面终端;
- RACH:8 时隙突发,单向上行链路,用于用户终端随机竞争接入;
- AGCH:8 时隙突发,单向下行链路,用于应答用户终端发出的 RACH,向地面终端分配 SDCCH 或 TCH 信道。
- BACH: 15 时隙突发,单向下行链路,用来提醒地面终端。当用户处于不利的位置和下行链路信号被严重遮蔽衰落时,对地面终端进行增强寻呼。

表 1 为以上各个信道调制和编码方式。

表 1 GMR-1 信道类型及调制编码方式

信道	信道编码	调制方式
TCH3	1/2 卷积	π/4-CQPSK
TCH6	1/2 卷积	π/4-CQPSK
TCH9	1/5 卷积	π/4-CQPSK
FCCH		双 Chirp
BCCH	1/2 卷积	π/4-CQPSK
PCH	1/2 卷积	π/4-CQPSK
RACH	1/4 卷积	π/4-CQPSK
AGCH	1/2 卷积	π/4-CQPSK
BACH	RS 编码	6PSK

2 物理层软件架构

基于 GMR-1 R1 的信关站模拟器可用于终端的物理层协议一致性测试。

为了便于对物理层协议进行充分测试,物理层协议采用“Python + C”的程序架构,其中 C 语言主要用于实现对运行性能要求较高的关键算法,然后通过 Python 脚本调用 C 模块的方法实现逻辑信道的发送和接收信号处理流程。每个信道收/发任务都是一个 Python 线程,独立启动和终结。为满足物理层协议对实时性的要求,采用 Linux 操作系统。

2.1 物理层软件的工作模式

模拟信关站的物理层软件有两种工作模式:脚本测试模式和高层联测模式。脚本测试模式下的收发逻辑信道调度由测试脚本实现,主要用于单个信道功能及简单信道组合的测试。测试脚本(测试用例)的选择是通过 Web Browser 界面完成的。另外,用户还可以通过 Web Browser 对设备的 RF 参数进

行设置和读取操作。高层联测模式下的收发逻辑信道调度由协议栈高层通过 L2 适配层指定,主要用于通信流程的测试。

2.2　物理层软件的设计思路与主要模块功能

2.2.1　软件系统组成及主要设计思路

图 1 描述了信关站模拟器的物理层基带信号处理软件功能结构。其中两条虚线之间为物理层软件的主体部分,主要包括测试脚本控制器、L2 适配器、收/发任务引擎、收发任务执行监视器、收/发任务和基带信号处理模块。所有这些模块都是 Python 线程,其中收/发任务线程由收/发任务引擎启动,其余线程都在系统初始化阶段即启动,并持续运行。软件系统主体部分的顶端通过 Socket 接口与 Web Server(脚本测试模式下)或者协议栈高层(高层联测模式下)相连,而底层则通过 PCIE 接口与 RF 处理板相连。

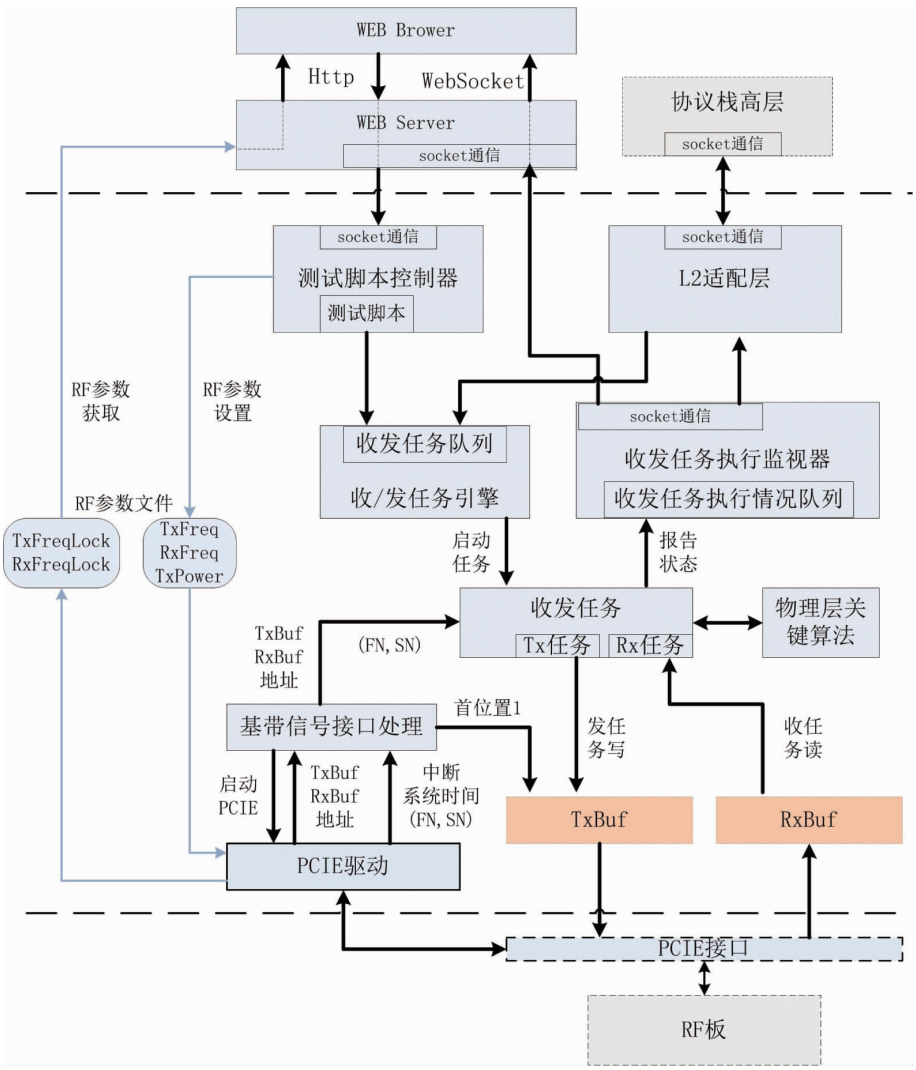

图 1　信关站模拟器功能结构图

在设计思路方面,为向上层提供统一的调用接口,并便于同时支持两种测试方式,上层调用与物理信道启动之间使用“任务队列”连接;同理,为提供统一的物理层向上层的数据传输方式,使用“收发任务执行情况队列”查询方式获取上报消息,并通过统一的 Socket 方式向上通报;为使软件设计与硬件系统解耦,收发任务通过两块物理内存(TxBuf 和 RxBuf)与 PCIE 接口相连接;为方便调试,软件主体及 Web Server 部分运行在工控机上,而 Web Browser 和协议栈高层可运行在单独的 PC 机上(不限操作系统),这样可以通过 Internet 网络连接物理层、上层与测试控制终端,从而简化测试系统的搭建。

2.2.2 主要模块功能

下面逐一介绍软件系统主体部分的 6 个模块以及收发存储区的设置。

(1) 测试脚本控制器和测试脚本

测试脚本由 Python 语言编写,用于指定何时(帧号 FN,时隙号 SN)进行对某个信道(包括信道名称、调制编码选项等)进行发送或接收处理。用户根据不同的测试用例分别编写测试脚本,使用 Web Browser 界面选择特定测试脚本,并通过 Web Socket 传递给工控机上的 Web Server。

脚本测试模式下,工控机上运行的测试脚本控制器将执行 Web Server 接收到的测试脚本,将其中指定的信道收发任务添加到“收/发任务队列”中。此外,测试脚本控制器还负责将用户通过 Web Browser 界面进行的 RF 参数设置写入到指定的文件中,并触发 PCIE 驱动读取相应文件,并将参数配置到 RF 板上。

(2) L2 适配模块

一方面,L2 适配模块解析上层原语,并将其中指定的信道收发任务添加到“收/发任务队列”中。另一方面,L2 适配模块负责将任务执行状态及接受信道解析出的信息比特传递给上层。

(3) 收发任务引擎

实时监控收/发任务队列,顺序读取队列中的任务项,并按照信道任务描述启动相应的 Python 线程。

(4) 收发任务与关键算法模块

每个任务都是一个 Python 线程,通过调用 C 语言实现的关键算法模块,独立完成某个信道的发送或者接受处理流程。

对于发送任务,主要是依次按照文献[9]描述的编码流程、文献[10]描述的调制方法和文献[8]描述的复用规则将信息比特生成指定突发,然后按照指定的时间及当前系统时间(FN,SN)将突发数据写到 TxBuf 中的对应位置上,并向“发任务执行情况队列”添加任务完成情况,如是否顺利生成突发、是否及时写到 TxBuf 中等。

对于收任务,首先需要根据任务指定的时间从 RxBuf 中的对应位置读取突发数据,然后进行同步、信道估计和均衡等工作,最后再按照发送突发形成的逆过程恢复信息比特,并向“收任务执行情况队列”添加任务完成情况,如循环冗余校验(CRC)是否正确、连续正确接收的帧数等。

(5) 收发任务执行监视器

该模块实时查询“收/发任务执行情况队列”中的状态项,并在“脚本测试模式”下将必要的状态信息通过 Socket 传回到 Web Browser 进行显示,或在“高层联测模式”下将高层所需信息反馈给 L2 适配层。

(6) 基带信号接口处理

该模块是软件系统与 PCIE 接口的桥梁。软件启动时,该模块将底层分配的内存从内核空间映射到用户空间,并将地址作为全局变量供每个收发任务查询。另外,该模块接收 PCIE 驱动送上来的中断,使用中断中包含的参数更新系统时间(FN,SN),并将其作为全局变量供收发任务查询。

(7) 收发缓存 TxBuf 和 RxBuf

软件系统通过收发缓存 TxBuf 和 RxBuf 与 PCIE 接口进行通信。TxBuf 和 RxBuf 的长度与一帧数据一致,I 路和 Q 路数据交错放置,存储区位置与系统时隙相对应。

硬件系统发通道根据系统时间周期性地通过 PCIE 接口读取 TxBuf,不管其中数据是否准备好。为了保证所生成的突发能够被 PCIE 接口正确读取,软件系统发任务需提前一段时间将数据写到 TxBuf 的相应位置。类似地,硬件系统收通道根据系统时间周期性地通过 PCIE 接口将 RF 板接收的数据写入 RxBuf。为了软件系统收任务读到是当前帧的新数据,收任务线程需在指定时间推后一段时间进行读操作。系统中,软件收/发任务与硬件读写缓存的时间差可按照具体情况进行调节。

3 物理层软件工作流程

本节将在脚本测试模式下,以“在(FN,SN) = (3,5)和(4,5)两个时刻发送 TCH3 突发”为例介绍软件系统的发送任务执行过程,并以“在(FN,SN) = (3,5)和(4,5)两个时刻接收 TCH3 突发”为例

介绍软件系统的接收任务执行过程。这两个例子中,系统时间更新间隔及收/发任务查询系统时间间隔均设置为 10 时隙,软件收/发任务与硬件读写缓存的时间差设置为 10 时隙至 20 时隙之间。

3.1　软件系统的发送任务执行过程

连续发送两个 TCH3 突发的过程如图 2 所示,具体步骤描述如下,其中每步的时序由最右侧的系统时间(FN,SN)表示。

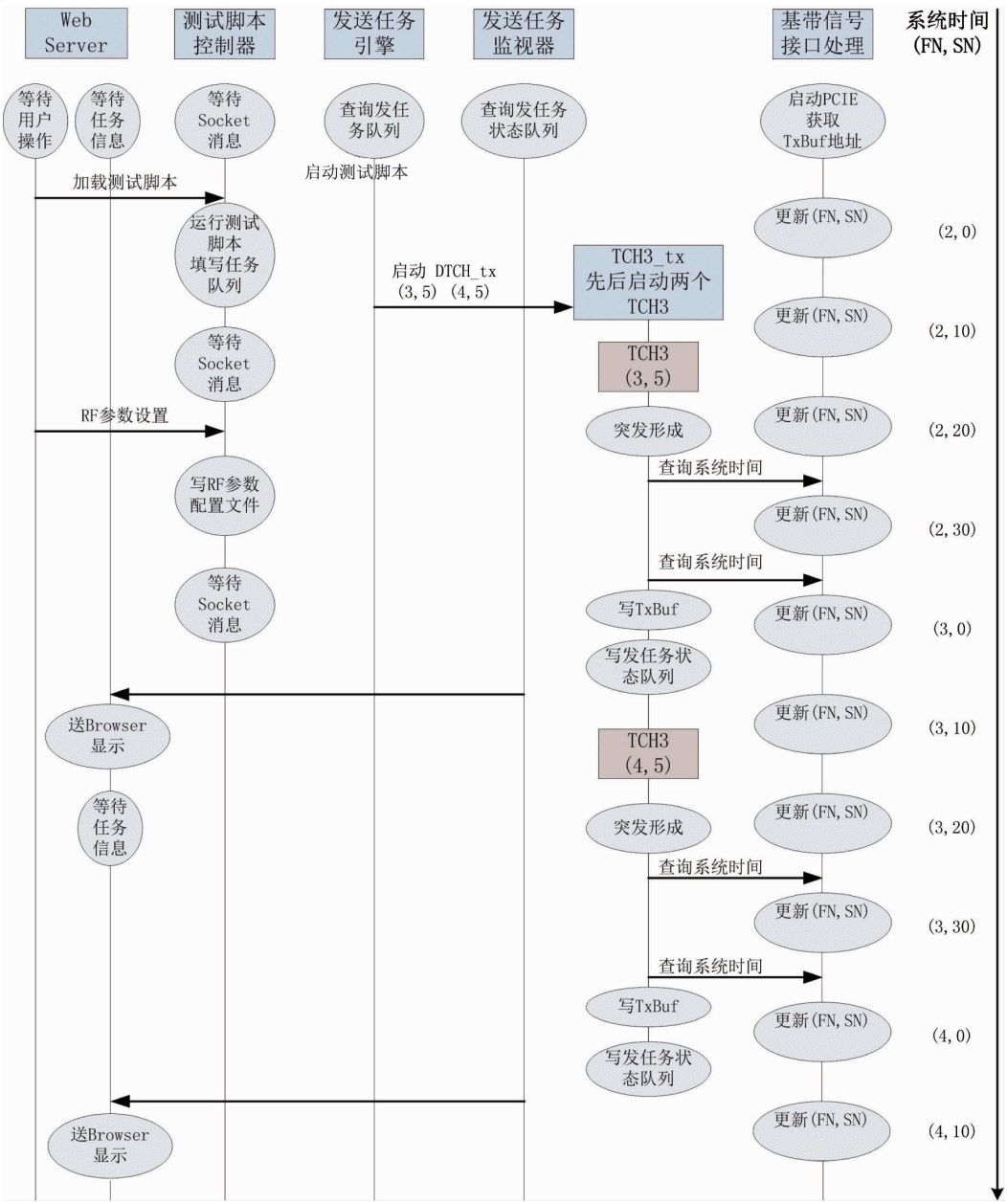

图 2　模拟信关站物理层软件工作流程图(发任务)

(1) 软件系统在(FN,SN) = (0,0)时刻启动,并开启了 Web Server、测试脚本控制器、基带信号接口处理、发送任务引擎和发送任务监视器 5 个 Python 线程。

(2) 基带信号接口处理模块启动后首先初始化 PCIE 接口,并获取 TxBuf 地址,然后每 10 时隙更新一次系统时间(FN,SN)。这里初始系统时间为(FN,SN) = (0,0)。

(3) 测试脚本控制器运行起来就开始等待 Web Browser 上传测试脚本。假设用户在(FN,SN) = (1,30)时刻上传测试脚本,Web Server 程序马上通过 Socket 通知测试脚本控制器,后者随即向发送任务

队列中添加一项任务,即在(3,5)开始时刻发送 TCH3 突发,连续发送两帧。

(4) 发送任务引擎一启动就开始循环查询发送任务队列,在(FN,SN) = (2,10)时刻发现有发送任务被添加,马上依据任务种类启动 TCH3 _ tx 线程,并依据发送时间序列先后调用两次 TCH3 突发形成程序。

(5) 第一个 TCH3 突发形成后开始每 10 个时隙查询一次系统时间,在(FN,SN) = (2,30)时刻判断出与预定的发送时间(3,5)时刻相差 15 时隙(介于 10 时隙至 20 时隙之间),于是开始将突发数据写入到 TxBuf 中的 5 ~ 7 时隙位置,完成后向发送任务状态队列中添加状态项"顺利完成任务"。

(6) TCH3 _ tx 线程随即再次调用 TCH3 突发形成程序,重复突发形成、时间查询、写 TxBuf 和报告完成状态的工作。发送任务完成后,TCH3 _ tx 线程结束。

(7) 发送任务监视器一起动就开始循环

查询发送任务状态队列,在(FN,SN) = (2,30)和(3,30)时刻发现队列中有状态添加后,马上将该消息通过 Socket 发送给 Web Server,后者将消息送 Web Browser 界面进行滚动显示。

3.2 软件系统的接收任务执行过程

系统连续接收两个 TCH3 突发的过程如图 3 所示。基本工作流程与发送任务类似,主要的不同之处是:TCH _ RX 线程被接收任务引擎启动后,马上开始周期查询系统时间,当在(FN,SN) = (3,20)时刻判断出当前时刻超出预定接收时间(3,5)的时间为 15 个时隙(在 10 ~ 20 个时隙区间内),马上读取 RxBuf 的 5 ~ 7 时隙位置。完成突发解码后将接收状态(CRC 校验结果)及解码数据一同添加到收任务状态队列。第一突发接收完成后,TCH _ RX 线程继续查询系统时间,重复上面两步接收第二个 TCH3 突发,并在完成两帧解码后结束进程。接收任务监视器查询到收任务状态队列中出现任务完成报告后,首先将信息发送给 Web Server 进行显示,之后将解码结果保存到文件中。

4 功能测试

由于所开发的信关站模拟系统主要是用于对终端芯片的物理层实现进行协议一致性测试,因此测试标准主要为是否可以实时正确解码终端发送的突发、所发出的突发是否可以由终端芯片正确解调,现阶段并不对底层算法的性能做过高要求。

4.1 环回自测

为了验证所开发的物理层信道收发功能实现是否正确,首先在工控机上对所开发的系统进行了射频环回自测。工控机采用 Intel8 核 I5-2400/3.10G 处理器,操作系统为 Ubuntu12.10。基本测试方式是将发送射频端口通过电缆连接到接收射频端口,通过测试脚本方式指定某个物理信道的发送和接收时间(帧号 + 时隙号)。测试结果表明,接收端可以正确恢复所有的发送数据。业务信道的收发处理时间如表 2 所示,控制信道的处理时间如表 3 所示。从表中可以看出,大部分信道的处理时间在 10ms 以内,最长的 RACH 接收为 30ms,为帧长的一半,因此所开发的系统可以支持实时处理各个逻辑信道的收发。

表 2 业务信道的收发处理时间

业务信息	发送处理时间(ms)	接收处理时间(ms)
TCH3	1.33	7.4
TCH6	4	12.1
TCH9	4.72	15.7

表 3 控制信道的处理时间

控制信道	时间(ms)
FCCH 发送	1.22
BCCH 发送	2.04
PCH 发送	2.06
AGCH 发送	2.06
BACH 发送	3.39
RACH 接收	30.2

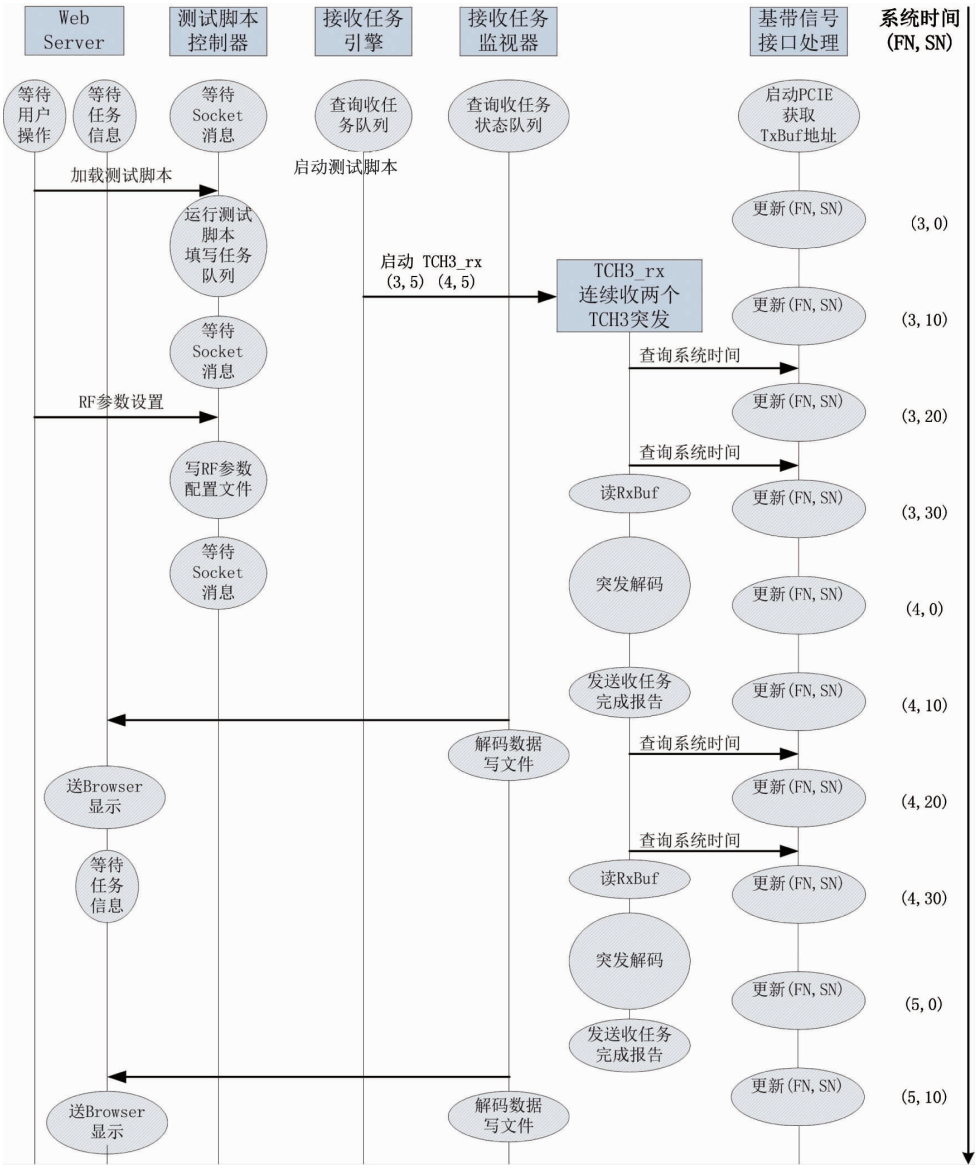

图 3　模拟信关站物理层软件工作流(收任务)

4.2　终端联测

进行以上环回测试之后,开展了对厂家开发的终端设备的协议一致性测试工作。测试系统的结构如图 4 所示。所开发的模拟信关站物理层设备通过网线与第三方开发的高层协议设备相连,共同组成完整的信关站协议处理系统。信关站物理层设备通过射频电缆(或者数字接口)与终端样机相连,并串接必要的衰减器。信关站物理层设备根据高层协议下发的原语指令发送或接收制定的物理信道,实现与终端样机的联通及测试。

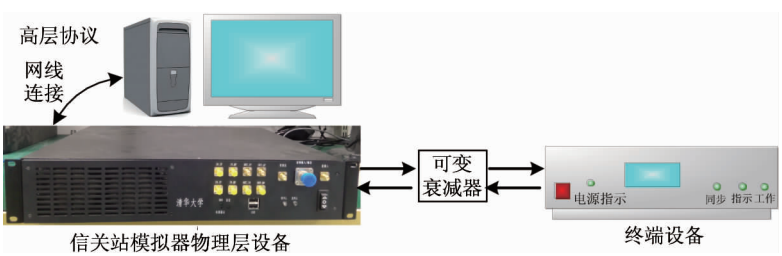

图 4　测试终端设备协议一致性的系统结构

借助以上测试系统,目前已经顺利完成了终端附着、随机接入、短信和语音通话的测试。这些结果不仅为终端样机的开发提供了参考,也证明所开发的信关站物理层设备的设计方法是可行的,所实现功能也是正确的。这为下一步开展同步、功控和越区切换等复杂流程打下了良好基础。

5 结论

本文介绍了基于"Python + C"软件无线电构架开发的 GMR-1 R1 模拟信关站物理层软件系统。该系统使用 C 语言实现关键算法模块,保证信道处理速度;通过 Python 调用 C 模块的方式实现信道处理流程,构建灵活的开发与调试架构。环回自测和终端联测结果都表明,所开发的模拟信关站的功能是正确的,可以满足 GMR-1 R1 标准的实时收发要求,能够支持终端设备协议一致性测试需求。接下来我们还将对该系统中的算法模块做进一步优化和性能评估。相信这套软件的开发将有力支持我国第一代卫星移动通信系统的终端基带芯片开发。

参考文献

[1] 张更新,甘仲民. 浅论我国卫星移动通信系统的发展思路和策略. 数字通信世界,2005,(7):24-27

[2] 何平江. 静止轨道卫星移动通信系统的设计考虑. 数字通信世界,2008,(8): 18-21

[3] 何家富,李广侠. 卫星移动通信的应用与发展. 见:第四届卫星通信新业务新技术学术年会会议论文集,2009. 17-24

[4] 张健. 发展我国卫星移动通信的问题及建议,见:第四届卫星通信新业务新技术学术年会会议论文集,2008. 60-65

[5] 张更新,甘仲民,李广侠. 对发展我国卫星移动通信的有关思考. 卫星与网络,2010,(5): 32-34

[6] 王京,赵明,晏坚. 关于发展卫星移动通信系统的一些思考. 见:第六届卫星通信新业务新技术学术年会论文集,2010. 241-244

[7] GMR-1 05.001GEO-Mobile Radio Interface Specification; part5: Radio interface physical layer specifications; Sub-part 1: Phtsical Later on the Radio Path:General Description;[S]2005.02

[8] GMR-1 05.002GEO-Mobile Radio Interface Specification; Radio interface physical layer specifications; Sub-part 2: Multiplexing and Multiple Access[S], 2002,04

[9] GMR-1 05.003GEO-Mobile Radio Interface Specification; Part 5: Radio interface physical layer specifications; Sub-part 3: Channel Coding.[S]2002.04

[10] GMR-1 05.004GEO-Mobile Radio Interface Specification; Part 5: Radio interface physical layer specifications; Sub-part 4: Channel Coding.[S]2005.02

[11] GMR-1 05.005GEO-Mobile Radio Interface Specification; part5: Radio interface physical layer specifications; Sub-part 5: RadioTransmission and Reception;[S]. 2005.02

Software design for physical layer of simulative gateway station of mobile satellite communication systems based on GMR-1

Zhu Zhe*, Zhou Jie*, Liu Yang*, Gao Zhen*, Li Xin**, Zhao Ming**, Wang Jing**

(*School of Electronic Information Engnineering, Tianjin University, Tianjin 300072)

(**Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084)

Abstract

According to the GMR-1 Release 1, a standard for geostationary earth orbit mobile radio interface, a software system for the physical layer of the simulative gateway station of mobile satellite communications was designed by using the frame of software radio to test and verify the consistence of the physical layer protocol of GMR-1 Release 1 based mobile satellite communication terminals. With the processing efficiency and flexibility considered, the software system adopts the mechanism of using the Python to transfer the C function to realize each GMR-1 Release-1 logic channel's functions of receiving and delivering to meet varieties of needs. The tests show the system's functional correctness and effectiveness.

Key words: mobile satellite communication, simulative gateway station, GMR-1 Release 1, software radio, Python