

基于软件无线电的同步机制研究

段红亮^{1,2}, 刘浩¹, 杨志远¹, 任斌¹, 陈晓光¹

(1. 北京航天长征飞行器研究所, 北京 100076; 2. 西北工业大学 机电学院, 西安 710072)

摘要: 软件无线电 (software defined radio, SDR) 以其灵活性和通用性等优点, 成为了当代移动通信的一大研究热点; 然而主流的 SDR 原型平台仅实现了点对点通信, 由于未能解决时钟同步问题, 在多点间通信方面难以保证时序控制和多用户调度; 为搭建基于 SDR 平台的多点通信场景, 提出了一种基于软件的网同步实现机制, 并采用回程时延 (round trip delay, RTD) 算法获得软时钟同步; 实验表明在微软软件无线电原型平台 (Sora) 上能获得误差为纳秒级精度的软时钟源, 能够模拟 TDMA 接入方式下的多用户通信场景, 验证了软时钟同步机制的正确性与稳健性。

关键词: 软件无线电; 回程时延算法; 软时钟同步; TDMA

Research on Synchronization Mechanism Based on Software Defined Radio

Duan Hongliang^{1,2}, Liu Hao¹, Yang Zhiyuan¹, Ren Bin¹, Chen Xiaoguang¹

(1. Beijing Institute of Long March Vehicle, Beijing 100076, China

2. School of Mechanical Engineering, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: Software Defined Radio (SDR) has become a hotspot research of mobile communication because of its flexibility and versatility. However, the mainstream SDR prototype platform only realizes point-to-point communication. Because it fails to solve the problem of clock synchronization, it is difficult to ensure timing control and multi-user scheduling in the aspect of multi-point communication. In order to build a multi-point communication scenario based on SDR platform, a software network synchronization mechanism is proposed, and round-trip delay is used, RTD) algorithm obtains the soft clock synchronization. The experiments show that the soft clock source with nanosecond error can be obtained on the microsoft software radio prototype platform (Sora), which can simulate the multi-user communication scenario under TDMA access mode. In conclusion, the experiments verify the correctness and stability of the soft clock synchronization mechanism.

Keywords: software radio; round-trip delay algorithm; soft clock synchronization; TDMA

0 引言

当今无线通信领域内, 多种通信体系并存, 各种标准竞争激烈, 频谱资源紧张, 传统的以硬件为主、面向特定用途机制的不足日益突显等一系列问题制约着无线通信技术的发展, 而具备通用性、开放性和全面可编程性的软件无线电技术则适应了时代发展的需求^[1]。软件无线电由 J. Mitola 于 1992 年首次提出^[2], 全称为软件定义的无线电, 即 Software Defined Radio, SDR。其核心思想是: 构造一个具有开放性、标准化、模块化的通用硬件平台, 使 A/D 和 D/A 转换器尽可能靠近天线, 将其他各种功能, 如物理层和协议栈等都由软件来实现。理想的软件无线电结构如图 1 所示。

近年来 SDR 已经获得了迅猛的发展, 然而由于受到硬件性能和软件处理时延等诸多因素的限制, 如宽带天线、高速 A/D (D/A)、高吞吐量总线架构以及不同编程语言的实时性等, SDR 存在着多种问题^[3-4], 目前硬件的性能尚未达到理想 SDR 的要求, 因此, 如何在现有的有限软硬件性能的限制下, 尽可能实现 SDR 的功能成为了当今 SDR 通信

收稿日期: 2020-06-08; 修回日期: 2020-06-16。

作者简介: 段红亮 (1983-), 男, 河北邯郸人, 博士生, 工程师, 主要从事雷达电子对抗、机电一体化设计方向的研究。

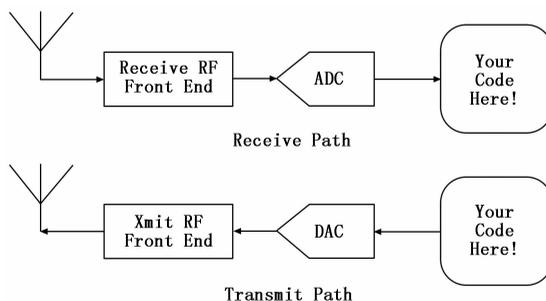


图 1 理想的软件无线电结构

原型平台的热点。

基于通用硬件架构的平台代表了 SDR 的发展趋势, 但是由于通用平台在处理速度、吞吐量及实时性方面的限制, 基于通用平台的 SDR 研究难以获得广泛的应用可行性。随着近几年来通用硬件性能的飞速提升和软件技术的发展, 微软亚洲研究院于 2009 年推出了基于普通 PC 和 Windows 操作系统的通用 SDR 平台——Sora, 它实现了 IEEE 802.11 a/b 协议, 采用一系列技术较好地解决了如何在通用平台和非实时操作系统中研究开发 SDR 的难题^[5]。Sora 平台实现了基于 CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) 接入机制的 802.11 a/b 协议, 并实现了点与点之间的通信^[5]。然而, 当采用多点构建多址

接入网络时, 由于这些接入点之间不存在统一的时钟, 无法实现网络同步, 因此难以模拟实时性的多址接入, 如 GSM, TD-LTE 等 TDMA 特性的接入系统。目前商用的无线通信系统中的网同步 (也称为时钟同步) 利用 GPS (Global Positioning System) 等作为基站同步时钟源, 这是 TDMA (如 GSM) 或是 CDMA (IS-95) 等系统的首选方案, 另外相邻基站间的时钟同步直接影响到越区切换的成功率, 也是 CDMA 系统中能够实现软切换的重要前提之一。

在此提出了一种基于通用硬件平台的软时钟同步实现机制, 相比于 GPS 等“硬”时钟同步, 该机制称之为软时钟同步, 它在现有 Sora 原型平台上实现了 TDMA 接入方式的模拟蜂窝通信。

1 通信系统中多种同步机制描述

在数字通信系统中, 同步包括载波同步, 码元同步, 群同步和网同步 4 种^[6]。载波同步和码元同步是解调时所必需的, 进一步是群同步 (或称为帧同步), 它是用来识别信息分组, 如一个完整的物理帧等, 如图 2 所示, 上述三种是点对点之间成功通信所必需取得的同步, 而网同步则描述了通信网中所有参与者的时钟关系, 称之为时钟同步, 使得在整个通信网内有一个统一的时间节拍标准, 它反映了基站和终端、多终端之间的协作关系, 是构成数字蜂窝网和多用户调度的基础。时钟同步又分为基站间的时钟同步和蜂窝小区内的时钟同步两种。在现有移动通信系统中, 相邻小区的基站之间必须取得时钟同步, 因为时钟同步是影响越区切换的重要因素之一。且对于那些 TDD 模式的蜂窝系统, 在每个蜂窝小区内, 每个终端必然也要与基站取得时钟同步, 否则无法正常接入系统, 例如基站会连续指示每个连接状态下的终端上行时间提前量, 即 TA (timing advanced) 补偿信号的空间传播延时从而在基站取得时钟同步^[7]。

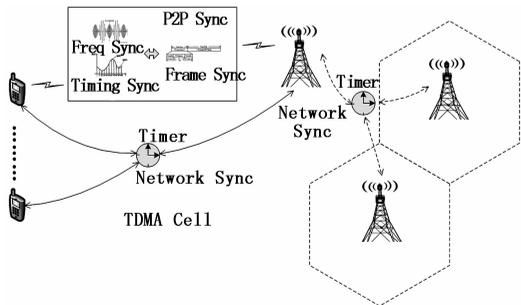


图 2 TDMA 蜂窝通信中的四种同步

基于通用硬件架构的 Sora 作为仿真平台 (如图 3 所示) 搭建仿真系统场景, 仿真场景中存在一个单独的基站 (BS), 以它的时钟作为蜂窝小区的基准时钟, 其他平台作为终端 (UE) 取得与基站的时钟同步。为简化仿真复杂度且不失一般性, 本文采用一基站、两终端的简单蜂窝场景, 设计帧长为 10 ms, 每帧包含有 10 个子帧, 每个子帧 1 ms,

#0 子帧基站用来发送一个特殊 MAC 帧 (软同步序列, software synchronization sequence, SSS), SSS 中包含了时间提前量 TA、帧号和上下行帧配置等信息, 具体场景和帧配置如图 3 所示。

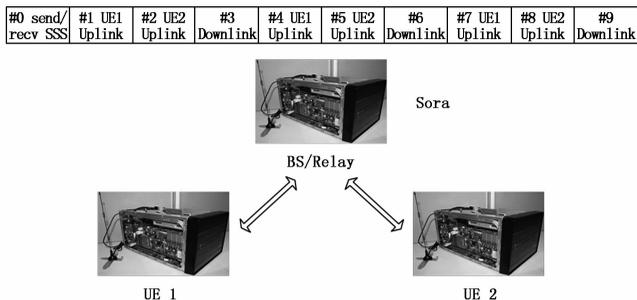


图 3 模拟 TDMA 蜂窝通信场景

2 软时钟同步方法研究

基于通用 CPU 平台的软件无线电原型平台中实现网同步必须具备两个条件: 一是要有一个实时高精度的时钟源; 二是每个时钟要取得同步。考虑到采用 GPS 的不足和 SDR 的可扩展性问题, 由软件生成一个时钟源来取代 GPS 等外接时钟源是较优的选择。

2.1 实时软时钟源的实现方法

相比于 GPS 或晶振等“硬”时钟源, 由软件生成的时钟源有以下三方面优势: 1) 时钟的频率在其精度允许范围内任意可调; 2) 程序处理软时钟中断的系统开销小于 I/O 硬中断; 3) 便于 SDR 的集成与交互。然而如何在非实时操作系统 (如 Windows) 中实现实时的、高精度的软时钟是现代操作系统的一大难题^[8]。intel 提供了一种直接提取 CPU 时钟运行周期的技术—RDTSC, 它可以达到 ns 级的时间精度^[9], 然而在多核的操作系统上该技术的精度不能保证, 原因是它每次读取的 CPU 逻辑核可能不相同, 而多个 CPU 核之间并不完全均衡, 时钟运行周期数一般不相等, 所以 RDTSC 指令读取的数据可能存在着较大的误差, 另外非实时操作系统中一旦将运行着 RDTSC 功能的线程调度出当前 CPU 核, 也就无法保证实时性, 例如 Windows 的调度时间片大小约为 10 ms^[8]。

文献 [5] 中提出了一种独占性线程技术 (exclusive thread), 该技术在多核 CPU 平台下能使某个线程始终独占一个 CPU 逻辑核 (目前 Intel CPU 可以在一个物理核中同时运行两个线程, 故称之为两个逻辑核), 保证它不会被操作系统调度出去, 从而保证了处理的实时性。这项技术充分利用了现有硬件平台的优势, 尤其是多核的优势, 一定程度上弥补了多任务操作系统在实时性方面不足, 在非实时的系统平台中同时实现了局部实时的操作响应。从 Windows 的任务管理器可以看到被某个独占性线程占用的 CPU 逻辑核的使用率始终保持在 100%, 如图 4 所示, 该 CPU 包含 8 个逻辑核, 核 3、5、7 分别被 3 个独占性线程占用。

为了保证多核系统中时钟的精度和实时性的要求, 本

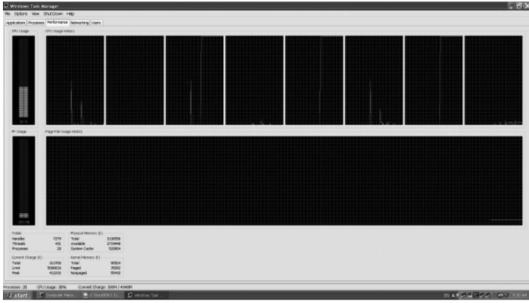


图 4 独占性线程的 CPU 占用率

文在 Windows 中创建一个独占性线程，在该线程在核 7 (图 4) 中始终在读取 CPU 的运行周期，克服了 RDTSC 指令在多核架构中的不足，从而实现了高精度的、实时的软时钟源，完全达到 (甚至超过) GPS 的商用授时精度 (误差 < 100 ns)。

2.2 软时钟同步机制——RTD 算法

近年来，许多学者对数字无线通信中的时钟同步做出了大量的有益探索与创新。如文献 [10] 针对采用 GPS 授时的网络同步提出了时钟差分的方法，N. Wu^[11] 等人利用差分 GPS 实现了 TDMA 网络中的时钟同步，C. E. Wheatley^[20] 对 CDMA2000 的 BS 间的时钟同步给出了一种时差传递测量技术，较好地达到了 BS 之间的时钟同步且有效降低了系统对 GPS 的依赖性，文献 [12] 对 [13] 所提出的技术进一步做了详细的仿真与证明。另外，文献 [14] 提出了一种基于 TDMA 自组织网络时钟的主从同步算法，而文献 [15] 则实现了自组织网络中时钟的互同步算法。然而，与 GSM、CDMA 等商用网络或自组织网络不同的是，基于非实时操作系统的 SDR 原型平台在信号处理及信令交互等方面有着自己独特的方面，例如采用复杂指令集 (CISC) 的通用处理器 (general-purpose processor, GPP) (如 Intel, AMD) 不能够精准地估计出指令的处理时延，另外，PC 通过 PCIe 连接前端 RF 还会存在一个指令的传输时延^[7] 等等，因此针对通用平台软件无线电，在如何取得软时钟同步方面，提出了一种回程时延 (round trip delay, RTD) 算法。

RTD 算法的核心思想是：基站每隔一定的时间间隔 (如 40 ms) 广播一个软同步序列 SSS，终端在接收到该帧并进行 PHY 层解码后递交到 MAC 层，根据 SSS 的发送时延、传播时延、接收时延和解码时延等确定出终端时钟的时间提前量 (time advanced, TA)，只要能确定终端时钟的 TA，那么就实现了终端与基站之间的时钟同步。因此如何确定 SSS 从基站到终端的总的时延是软时钟同步技术的关键，考虑基站只能测出发送时延，而终端仅能测出接收和解码时延，所以如果仅仅在任意一端均不能正确估计出总的时延大小。

RTD 算法的基本过程是：首先将 SSS 调制编码后放置 RF 缓存，BS 在一定的时间节点首先指示发送一个 SSS 给 UE，UE 在接收并在 PHY 成功解码 SSS 后，立即指示发送

一个回程 SSS 给 BS，BS 端再接收和解码该回程 SSS。RTD 算法如图 5 所示。

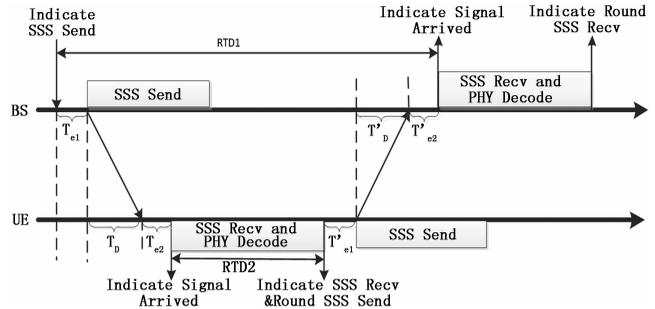


图 5 RTD 算法示意图

为了简化起见，上图中仅考虑 BS 与一个 UE 取得时钟同步的情况，水平横轴代表绝对时间轴。图中各参数的表示意义如下：

T_{e1}/T'_{e1} ：表示从软件指令指示 SSS 发送到 SSS 在 RF 端天线真正开始发送的时延，具体包含以下几因素：

- 1) 程序执行的时延，包括运行在 GPP 上程序的跳转、函数调用、代码编译和指令执行等；
- 2) 指令通过 PC 的通用外设接口的传输时延，如以太网卡、USB、PCIe、串/并口等；
- 3) RF 前端对指令的反应时延，如 DSP 或 FPGA 或专用 IC 从接收到执行指令的时延，一般来说，该时延在实时控制系统中也是存在的，一是它的量级非常小，二是它可以较为准确地估计出来。

T_D/T'_D ：从 BS 到 UE 的电磁信号路径传播时延；

T_{e2}/T'_{e2} ：表示从 RF 真正接收到信号到通知软件信号到来的软处理时延，具体因素与 T_{e1}/T'_{e1} 相似；

RTD1：表示在 BS 中从软件指示 SSS 发送到接收到回程信号到达指示的时间间隔；

RTD2：表示在 UE 端从接收到信号到来指示到成功解码 SSS 的时间间隔。

RTD 算法有两个前提假设：

- 1) 无论 BS 还是 UE，它们都采用完全相同的 SDR 原型平台，包括相同的硬件架构和软件架构，考虑到软处理时延具有较好地统计稳定性且时延比较时钟的要求精度小 2~3 个数量级 (参见本文仿真实验一)，那么工程上即用它们的统计均值来代替，由对称性有：

$$T_{e1} = T'_{e1}, T_{e2} = T'_{e2} \quad (1)$$

- 2) BS 与 UE 都是相对静止的，SSS 的来回路程是完全相同的，即：

$$T_D = T'_D \quad (2)$$

那么从图 5 中不难看出，BS 到 UE 的软时延大小为：

$$T_{e1} + T_D + T_{e2} + T'_{e1} + T'_D + T'_{e2} = RTD1 - RTD2 \quad (3)$$

联立以上三式，可以得到：

$$T_{e1} + T_D + T_{e2} = (RTD1 - RTD2) / 2 \quad (4)$$

UE 端若在“Indicate Signal Arrived”时刻启动自身的

软时钟, 那么它的时间提前量为:

$$TA = T_{e1} + T_D + T_{e2} \quad (5)$$

值得注意的是, RTD2 主要是由终端的接收和 PHY 解码时延构成, 时延较大且在不同系统中、不同负载下、不相同波动也较大, 另外可能在数据帧解码过程中也有很多不确定时延, 称为 Dead Time, 但并不影响 TA 的结果, 因为 TA 是 RTD1 与 RTD2 的差, 两者相减完全将这些不确定时延抵消。TA 是基于两个统计量的, 为了防止误差累积, 一般要每隔一定的时间间隔重新做一次软时钟同步, 在目前商用通信系统中也采用此种方法来消除累积误差。

3 实验结果与分析

在软件无线电平台上对软时钟精度、回程时延算法和软时钟同步机制进行了测试和验证。选取一台 Sora 作为中继, 另外两台作为终端, 硬件配置如表 1 所示。实验一中对参数 TA 进行了 1 000 次样本的抽样统计, 实验二是用示波器在射频端对软同步机制的验证。

表 1 Sora 平台配置

CPU	内存	操作系统	外接接口	物理层
intel Core i7 930	4GB	Windows XP sp3	PCIe X 8, 32 Gbps	802.11 a/b

实验一: 回程时延算法验证

物理层采用 802.11 a, 图 6 是 802.11a 协议中的 preamble 结构, 它在协议中由 10 个短训练序列和 2 个长训练序列组成, 完成信号检测、频率同步、码元同步, 自动增益控制、信道估计等, 在 RTD 算法中 T_D 参数包含了信号检测的时延, 所以参数 T_D 肯定大于 $5.6 \mu s$ (7 个短同步序列的时间长度)。

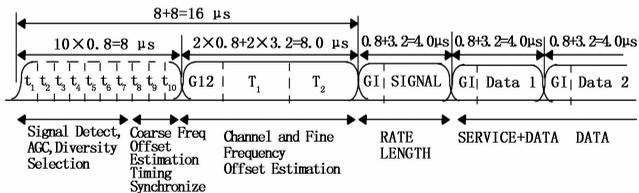


图 6 802.11a 的 preamble 结构

在室内测试两天线间的距离为 3 m (包含了两端屏蔽线的长度在内), 则传播时延 T_D 为:

$$T_D = 3/3 \times 10^8 = 10 \text{ ns} \quad (7)$$

所以传播时延与其他两个参数和 $T_{e1} + T_{e2}$ 相比, T_D 可以忽略不计。表 2 和图 7 分别是在 Sora 平台上所做的 1 000 次回程时延的统计表和样本分布直方图。

表 2 TA 测试值的统计结果

总次数	均值/ μs	方差/ μs^2	最大值/ μs	最小值/ μs
1000	16.08	1.54	19.07	11.45

通过统计表和直方图可以看到, 均值为 $16.08 \mu s$, 均方

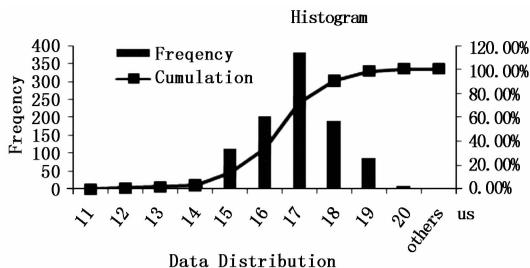


图 7 TA 统计结果的分布情况

差 (MSE) 仅为 $1.54 \mu s^2$, 最重要的是无论是最大统计值, 还是最小统计值, 与均值差的绝对值不超过 $5 \mu s$, 具备了良好地统计稳定性, 所以工程上完全可以采用均值作为 TA 的估计值来使用, $5 \mu s$ 的误差对于时隙长度为 3 ms, 在工程上是完全可以接受的。另外, 该实验也较好地证明了 RTD 算法在通用处理器 (GPP) 架构和非实时操作系统环境中的可行性和正确性。RTD 算法充分考虑了软件无线电平台的处理时延问题, 对于时延的估计既具备统计稳定性, 又能反映出软件无线电的软处理时延的瞬时波动性。算法的复杂度较小, 极易工程实现, 具备较高的工程实用性。

实验二: 软同步机制的验证

为了简便又不失一般性, 该实验中测试了一终端接入中继的情景, 保证在随机接入过程中没有冲突发生。当终端侦听到中继端广播的软同步序列后发起一个随机接入过程, 然后根据中继指示的 TA 值调整本地时钟, 使之与中继完成时钟同步, 直到 1 s 后再重新发起一次重同步过程。在这 1 s 的时间内, 中继每隔 10 ms 发送一个软同步序列, 终端也每隔 10 ms 发送一个软同步序列。也就是说, 中继端在 1 s 内发送了 100 个软同步序列, 而终端只发送了 99 个软同步序列 (相差的一个用于终端接入和时钟同步过程, 在此终端未发送任何序列)。

若终端与中继完成时钟同步后, 两端会在每隔 10 ms 相同的时间点 (时间上是重合的) 发送电磁信号, 两者的电磁信号在空间内相互干扰, 第三方测试点是不能够正确检测到两者任何一个的, 然而可以采用双通道的数字示波器同时测量两者的天线射频信号, 即一个高电平信号。双通道数字示波器的两个通道共用一个时间轴, 所以可以用来验证两个软同步序列是否在时间上是真正重合的, 如图 8 所示。

由示波器的测试结果可以看到两点: 一是中继端与终端的软同步序列在时间轴上是良好地重合的; 二是相邻的两个序列之间的时间间隔为 10 ms, 从另外一个层面也证明了软时钟的可靠性。

4 结束语

本文针对微软的 Sora 软件无线电平台提出一种软时钟同步机制, 首先用软件在 Windows 操作系统中实现了实时的高精度时钟源, 然后采用回程时延算法取得软时钟同步, 即实现了多台 Sora 构成的蜂窝通信场景的网同步。进一步而言, 软时钟同步机制有很好的灵活性, 它并依赖于 Sora

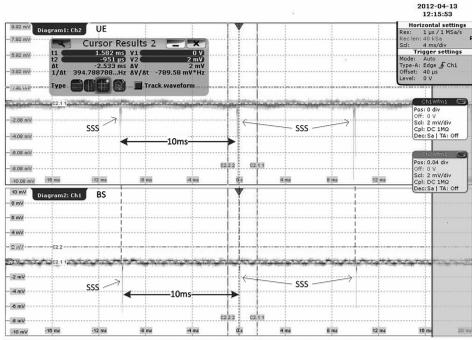


图 8 双通道示波器对软同步机制的测试

平台, 可以适用到其他基于通用硬件架构 (GPP) 和非实时操作系统的软件无线电平台中, 大大的拓宽了软件无线电的研究与应用范围。

参考文献:

[1] 肖维民, 许希斌, 朱健, 等. 软件无线电综述 [J]. 电子学报, 1998 (2): 65-70.

[2] Mitola J. The software radio architecture [J]. IEEE Communications Magazine, 2002, 33 (5): 26-38.

[3] 张健. 软件无线电的基本理论构架 [D]. 成都: 电子科技大学, 2000.

[4] 王珂. 软件无线电的关键技术及其应用 [J]. 四川通信技术, 2011 (1): 51-54.

[5] Tan K, Liu H, Zhang J, et al. Sora: High-performance software radio using general-purpose multi-core processors [J]. Communications of the ACM, 2011, 54 (1): 99-107.

(上接第 227 页)

[6] 张德华. 基于角色访问控制的授权管理系统设计和实现 [D]. 西安: 西安电子科技大学, 2010.

[7] 张宇一, 张保稳. 基于本体的 RBAC 建模及应用研究 [J]. 通信技术, 2017 (1): 102-108.

[8] 刘超, 赵建平, 穆星, 等. RBAC 权限管理在实验室管理系统中的应用 [J]. 电子技术, 2016 (1): 40-43, 37.

[9] 鞠宏军, 杜丽娟. RBAC 权限管理在教务管理系统中的应用与实现 [J]. 华北科技学院学报, 2014 (5): 70-73.

[10] 黄智, 张正风. 基于 RBAC 的校友管理系统权限管理研究 [J]. 软件导刊, 2015 (3): 128-129, 130.

[11] 李光文. 基于 RBAC 模型用户访问控制的研究与实现 [J]. 天津科技, 2014 (12): 15-17.

[12] 吴波, 王晶. 基于基本 RBAC 模型的权限管理框架的设计与实现 [J]. 计算机系统应用, 2011 (4): 50-54.

[13] 张晓李. 基于 MySQL 的小型数据库管理工具实现 [J]. 微处理机, 2015 (2): 48-52.

[14] 倪东英, 张晓丽. 基于 RBAC 的用户权限管理的设计与实现 [J]. 济南大学学报 (自然科学版), 2010 (2): 167-171.

[15] 李蔚, 王晶. 扩展 RBAC 访问控制模型在家庭开放平台中的应用 [J]. 电信科学, 2015, 31 (11): 21-28.

[16] 王志瑞, 顾问, 刘正涛. 基于 RBAC 模型的权限组件 [J]. 计算机系统应用, 2015 (3): 156-160.

[6] 樊昌信 曹丽娜. 通信原理 (第 6 版) [M]. 北京: 国防工业出版社, 2007.

[7] 3GPP TS 36.213. LTE-Advanced: Physical layer procedures [S]. V10.2.0, 2011,6.

[8] Andrew S. TANENBAUM. 现代操作系统 (英文版) (第 3 版) [M]. 北京: 机械工业出版社, 2009.

[9] Intel Corporation. IA-32 Intel architecture software developer's Manual: volume 2: Instruction Set reference [M]. Intel Corporation, 2002.

[10] Lindsey W C, Ghazvinian F, Haggmann W C, et al. Network synchronization [J]. Proceedings of the IEEE, 1985, 73 (10): 1445-1467.

[11] Wu N, Wang H, Kuang J M, et al. TDMA Network Time Synchronization and Navigation Using DGPS [A]. International Frequency Control Symposium & Exposition [C]. IEEE, 2007.

[12] Wheatley C, Wheatley C E. Self-synchronizing a CDMA cellular network [J]. Microwave Journal, 1999, 42 (5): 320-328.

[13] Mazzenga F, Vatalaro F, Wheatley C E I. Performance evaluation of a network synchronization technique for CDMA cellular communications [J]. IEEE Transactions on Wireless Communications, 2002, 1 (2): 322-332.

[14] Zhong X F, Mei S L, Wang Y Z, et al. Synchronization in TDMA Ad Hoc network [A]. IEEE Vehicular Technology Conference [C]. IEEE, 2005.

[15] 杨琦, 石江宏, 陈辉煌. 无线自组织网络分布式时隙同步算法 [J]. 系统工程与电子技术, 2011 (4): 178-183.

[17] Kuhn R D, et al. Adding Attributes to Role-Based Access Control [J]. Computer, 2010, 6 (6): 79-81.

[18] 张斌, 张宇. 基于属性和角色的访问控制模型 [J]. 计算机工程与设计, 2012 (10): 3807-3811.

[19] Chadwick D, Otenko A, Ball E. Role-based access control with X.509 attribute certificates [J]. IEEE internet computing, 2003, 2 (2): 62-69.

[20] 顾春华, 高远, 田秀霞. 基于角色访问控制约束的研究综述 [J]. 电脑知识与技术, 2017 (35): 24-27.

[21] 马晓普, 赵莉, 李瑞轩. 基于角色访问控制中的约束研究 [J]. 小型微型计算机系统, 2015 (9): 1982-1987.

[22] Sandhu R S, Coyne E J. Role-based access control models [J]. Computer, 1996, 2 (2): 38-47.

[23] 熊厚仁, 陈性元, 杜学绘, 等. 基于角色的访问控制模型安全性分析研究综述 [J]. 计算机应用研究, 2015 (11): 3201-3208.

[24] 冯俊, 王箭. 一种基于 T-RBAC 的访问控制改进模型 [J]. 计算机工程, 2012 (16): 138-141.

[25] Wang X Y, Sun J L, Yang X H, et al. Security violation detection for RBAC based interoperation in distributed environment [J]. IEICE transactions on information and systems, 2008, 5 (5): 1447-1456.