

跨平台NFC软件栈的研究与架构设计

刘翔, 肖堃

(电子科技大学计算机科学与工程学院 成都 610054)

【摘要】 NFC软件栈是一种中间件, 是上层应用与底层NFC控制器硬件操作之间的纽带。其核心功能是实现近距离无线通信论坛(NFC Forum)定义的协议体系, 根据具体的NFC工作模式和应用场景建立不同的服务模型。NFC软件栈是多层次的复杂系统, 其实现方式与应用接口、操作系统服务、NFC控制器操作等技术细节密切相关。当前的NFC软件栈在多操作系统支持、多NFC控制器芯片支持以及第三方应用兼容性方面存在不足。为此, 该文提出了一种跨平台NFC软件栈架构, 通过NFC软件栈运行时环境提供抽象的、无差异的操作系统基础服务; 通过NFC控制器抽象层向上层屏蔽不同NFC控制器之间的差异; 通过NFC服务接口层来适应不同的操作系统服务组件或应用框架, 以实现第三方NFC应用的兼容。基于该架构实现的NFC软件栈可以适应多种软硬件平台, 其功能和性能均达到设计期望。

关键词 跨平台; 中间件; NFC软件栈; 软件架构

中图分类号 TP39

文献标志码 A

doi:10.3969/j.issn.1001-0548.2014.01.015

Cross Platform NFC Stack and its Architecture

LIU-Xiang and XIAO-Kun

(School of Computer Science and Engineering, University of Electronic Science and Technology of China Chengdu 610054)

Abstract Near field communication (NFC) stack is a kind of middleware. Its main task is to implement the function defined by NFC Forum's specifications and protocols and to construct different service models according to the business scenario and NFC operating mode. The present NFC stack can hardly not only meet the requirement of simultaneously support of different types of operation systems and different types of NFC controllers, but also keep compatible to the third party NFC applications. So this paper proposes a cross platform NFC stack architecture. In this architecture, NFC stack can adapt to different operation systems and different NFC controllers. It can also work with different operation system service components or different application frameworks, and third party NFC applications can be supported perfectly.

Key words cross platform; middleware; NFC stack; software architecture

近场通信(NFC)允许在数厘米距离之间^[1]通过射频信号自动识别目标并获取相关数据, 识别工作无须人工干预^[2]。通过NFC技术, 任意两个设备(如移动电话)之间不需要线缆连接, 就可以实现相互间的通信。与其他短距离无线通信技术相比, NFC更安全, 反应时间更短, 因此非常适合用做无线传输环境下的电子钱包技术^[3]。由于NFC与现有非接触智能卡技术兼容^[4], 目前已经得到越来越多的厂商支持并成为业界事实标准。除支付功能外, NFC技术还可以在各种设备间提供轻松、安全、迅速而自动的通信。如NFC可以帮助人们在不同的设备间传输文字、音乐、照片、视频等信息, 还可以购买新的信息内容^[5]。

随着技术的发展, NFC控制器芯片在体积减小的同时提供了更加丰富的功能, 目前通常在单一芯片上集成了感应式读卡器、感应式卡片和点对点的功能^[6], 甚至有些厂商还集成了安全芯片(secure element)。通过天线与NFC控制器芯片的组合或者天线、NFC控制器芯片再加安全芯片的组合, 就能够实现支付、传输等所有功能, 而成本却仅为1~2欧元。NFC方案所具有的集成度高、方案简单、成本低廉等特点, 为智能手机或智能终端厂商集成NFC功能创造了良好的条件。NFC技术本身也能为智能手机或智能终端消费者带来更多的便利、互动与安全性, 透过虚拟世界应用与真实环境的连接, 进一步提升了用户体验。

当然, 任何事物都具有两面性的。虽然NFC硬

收稿日期: 2012-09-12; 修回日期: 2013-02-19

基金项目: 工信部2011年物联网发展专项资金

作者简介: 刘翔(1979-), 男, 博士, 主要从事云计算、IDC相关方面的研究。

件的设计难度较小,但是操作系统和应用程序均通过NFC软件栈来使用NFC硬件能力。对于终端厂商而言,在当前的技术条件下确并没有适合的NFC软件栈可供选择。

首先,智能手机或智能终端操作系统不存在统一标准,并且出于产品差异化的需求,终端厂商会在不同的产品上选择不同的操作系统,如Android、Windows Phone 7、Windows Phone 8、Symbian、Meego等。当前的操作系统对NFC的支持情况并不理想。某些操作系统具有原生的NFC软件栈,但功能较弱,如Android FRI(forum reference implementation)仅支持NXP PN544和NXP PN531^[7];某些操作系统,如Windows Phone 7,甚至没有提供原生NFC软件栈。同时,不同操作系统在驱动模型、内存管理、时钟管理、同步机制等方面存在着极大的差异,操作系统的原生NFC软件栈自然无法适用于其他操作系统。

其次,出于产品生命周期、供货渠道和成本等因素的考虑,终端厂商会在不同产品甚至同一产品的不同批次上采用不同的NFC控制器芯片。因为竞争关系和知识产权等原因,NFC芯片厂商提供的NFC软件栈虽然可能会支持多种操作系统,但一般仅支持自家的芯片。如INSIDE Secure Open-NFC,虽然提出了硬件无关和可移植的概念^[8],但主要是针对INSIDE Microread开发的,没有对硬件抽象层进行设计^[9]。不同NFC芯片厂商提供的NFC控制器芯片在实现上存在差异,如芯片接口可能为I2C、SPI或SWP,在HCI层可能提供不同的general gate等等,因此对某个厂商的NFC软件栈进行修改以适应另一种NFC控制器并非易事。

最后,NFC技术的魅力需要通过丰富多彩的应用来体现。智能手机或智能终端是开放的平台,有着完善的应用开发生态体系。应用开发者通过SDK使用硬件能力而无需关心底层实现,对于NFC技术也是如此。目前应用市场上的海量应用,都是通过这种模式开发出来的。第三方公司提供的NFC软件栈,如Zenius Solutions NFC软件栈、Stollmann NFCStack+等,虽然能支持多种操作系统与多种NFC控制器,但都是封闭的平台,其目的在于推广自身的应用或服务,不兼容基于SDK开发的第三方NFC应用。

为此,本文就NFC软件栈与软硬件平台之间的关系进行分析,提出了一种跨平台NFC软件栈架构。

该架构从设计上保证了NFC软件栈对不同操作系统和不同NFC控制器的灵活适应,还能够保证对第三方NFC应用的良好兼容。基于该架构实现NFC软件栈可以极大地简化终端厂商的集成工作。

1 跨平台NFC软件栈总体架构

NFC技术是由非接触式射频识别(RFID)技术及互联互通技术整合演变而来的^[10]。为了推动NFC技术的发展和普及,业界创建了近距离无线通信论坛(NFC Forum)。NFC Forum是一个非营利性的标准组织,其使命是促进NFC技术的实施和标准化,确保设备和服务之间协同合作。目前,NFC Forum在全球拥有数百个成员,包括SONY、Phlips、LG、Motorola、NXP、NEC、Samsung、Intel等公司。

为了保证与RFID系统的兼容性,NFC技术在底层采用了ISO/IEC 14443系列协议和ISO/IEC 18092协议^[11],两个NFC设备之间或者NFC设备与NFC标签/NFC卡之间就是通过这些协议来实现近场通信功能的。NFC设备的工作原理如图1所示。由图1可知,NFC控制器在实现近场通信功能的同时还实现了与主机(主处理器之上运行的NFC软件栈)之间的通信功能。两者之间的通信在物理层采用I2C、SPI或UART等串行通信协议,在链路层采用ETSI TS 102613 Single Wire Protocol(SWP)规定的链路层协议,在网络层则遵循ETSI TS 102622 Host Controller Interface(HCI)协议。

NFC软件栈是一种中间件,在上层为应用开发提供支撑;在底层操作NFC控制器。其核心功能是实现近距离无线通信论坛定义的协议体系,根据具体的NFC工作模式和应用场景建立不同的服务模型。本文提出的跨平台NFC软件栈总体架构如图2所示。由图2可知,跨平台NFC软件栈由NFC软件栈运行时环境、NFC控制器抽象层、NFC服务模型层和NFC服务接口层4个组件构成。其中,NFC软件栈运行时环境向其他组件提供抽象的、无差异的操作系统基础服务;NFC控制器抽象层着力于解决对不同NFC控制器的适应问题,通过对NFC控制器的功能进行抽象来屏蔽不同NFC芯片厂家的实现细节;NFC服务模型层实现NFC Forum定义的协议体系,并根据具体NFC工作模式和应用场景建立不同的服务模型。NFC服务接口层对不同的操作系统服务组件或应用框架进行适应,以兼容基于SDK开发的第三方NFC应用。

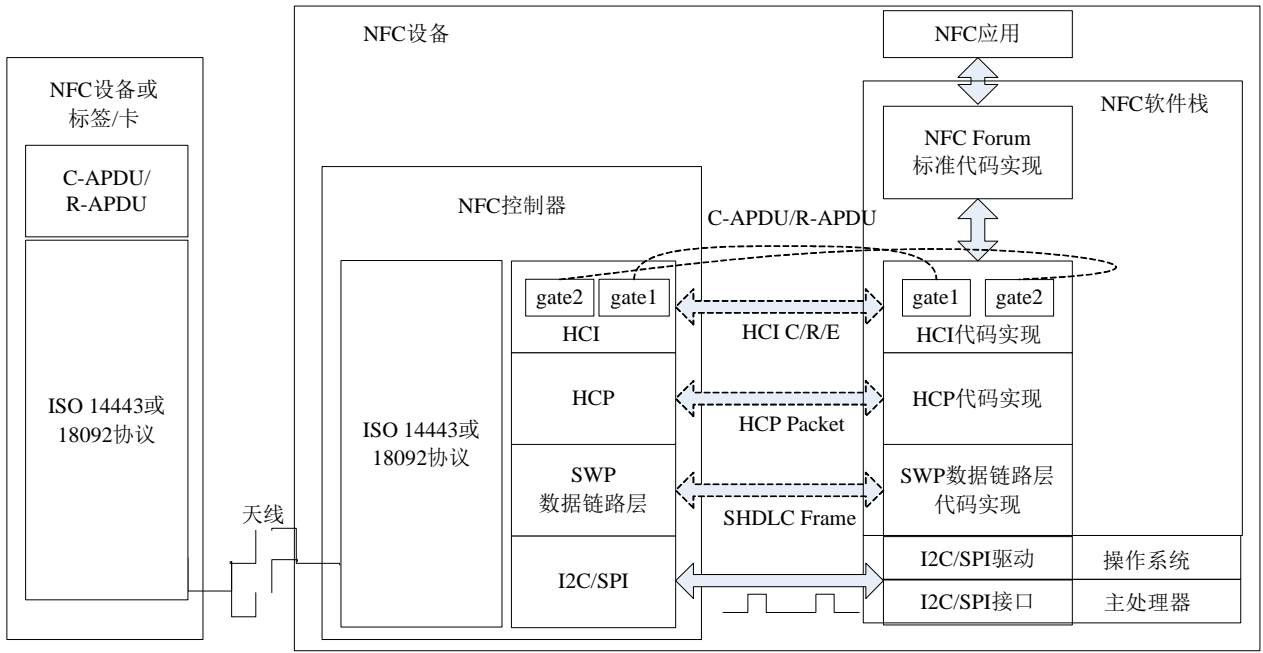


图1 NFC设备工作原理

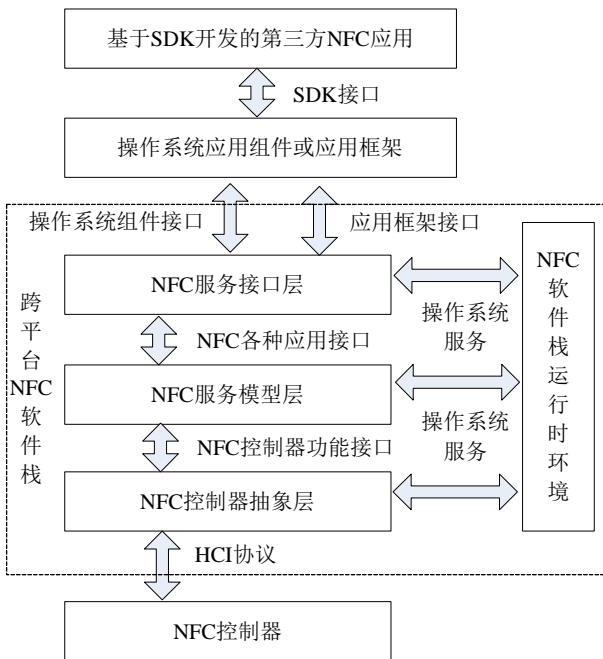


图2 跨平台NFC软件栈总体架构

2 跨平台性的实现

NFC软件栈是多层次的系统，既涉及底层的硬件操作，又涉及高层的应用接口；NFC软件栈更是复杂的系统，涉及到多线程甚至是多进程之间的交互。开发这样一个复杂的多层次系统需要用到很多的操作系统服务。

NFC控制器与主机之间通常是通过SPI、I2C或UART这样的串行总线连接的，NFC控制器抽象层就需要借助于串行总线设备驱动程序的能力来和NFC

控制器通信。

NFC软件栈的逻辑比较复杂，其组件都是多任务系统，并且会频繁地使用任务同步与通信机制。另外，实现NFC通信协议需要使用内存管理和定时器服务，为了辅助软件栈开发还需要使用操作系统的调试服务等。

为了适应不同操作系统之间的服务差异，通常的做法是在软件栈中设计移植层，然后针对不同的操作系统去实现移植层。如Open NFC在各组件中都有移植层，这样虽然能够支持不同的操作系统，但移植工作比较繁琐。为此，在跨平台NFC软件栈中设计了NFC软件栈运行时环境模块，通过该模块向软件栈中其他组件提供抽象的操作系统基础服务，包括任务控制、任务同步、任务通信、内存管理、时钟管理等，从而屏蔽不同操作系统之间的差异。如果需要适应不同的操作系统，仅需加载不同版本的“NFC软件栈运行时环境”即可。

由图1知，NFC控制器需要通过串行总线协议以及SWP、HCI等协议与主机进行通信。因为协议定义的是一般性的规则，不同的NFC控制器芯片在实现时可能存在差异，如数据链路层的帧封装格式可能不同、滑动窗口大小可能不一致等^[12]。

通过HCI协议定义的management gate，主机可以对NFC控制器与主机之间的通信进行管理；通过HCI协议定义的generic gate，主机可以操作NFC控制器进行近场通信，或者对NFC控制器的近场通信参

数进行配置。HCI协议并没有对generic gate的实现方式进行强制性的约定,这就给NFC控制器芯片厂商留下自主发挥的空间^[13]。

跨平台NFC软件栈通过NFC硬件抽象层来解决对不同NFC控制器芯片的适应问题。对于NFC控制器芯片在协议实现时的细节差异,如数据封装格式,

滑动窗口大小等,可以通过加载不同配置文件的方式来解决。对主流NFC控制器芯片的generic gate工作方式进行分析,同时结合HCI协议和ISO/IEC 14443系列协议的要求,抽象出NFC控制器的通用功能,从而向上层提供无差异的NFC控制器通用功能。NFC控制器抽象层的架构如图3所示。

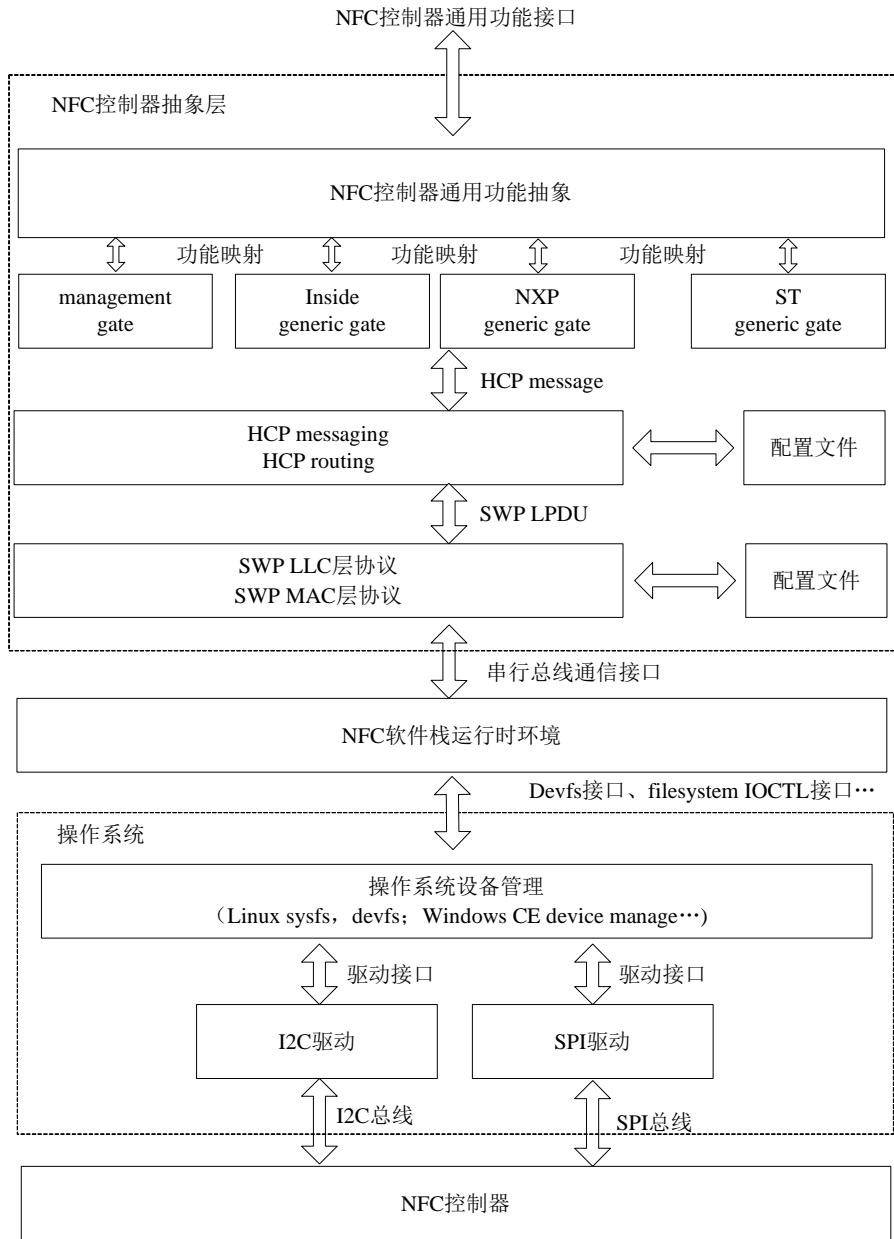


图3 NFC控制器抽象层架构

NFC服务模型层处于整个NFC软件栈的中心位置,与底层的NFC控制器、高层的应用接口或应用框架均没有关联。所需的操作系统基础服务也由NFC软件栈运行时环境提供,所以NFC服务模型层具有良好的跨平台性,其架构如图4所示。

NFC服务模型层提供了多种应用接口,如卡应用接口、NFC Forum注册协议接口、P2P数据传输应

用接口等,这些接口已经可以支撑NFC应用的开发。但是,在智能手机或智能终端的应用开发生态体系中,终端厂商需要按照操作系统服务组件或者应用框架的要求来实现相关接口,而不是简单地对外提供私有接口,这样应用开发者才能通过SDK使用NFC硬件能力。目前很多优秀的第三方应用已经运用NFC能力来实现新业务或改善用户体验,如果终

端不能兼容这些应用，无益会降低其产品的市场吸引力。因此，跨平台NFC软件栈通过NFC服务接口层来适应不同的操作系统应用组件或应用框架，以

确保应用开发者可以通过SDK使用NFC硬件能力，并兼容w基于SDK开发的第三方NFC应用。NFC服务接口层的架构如图5所示。

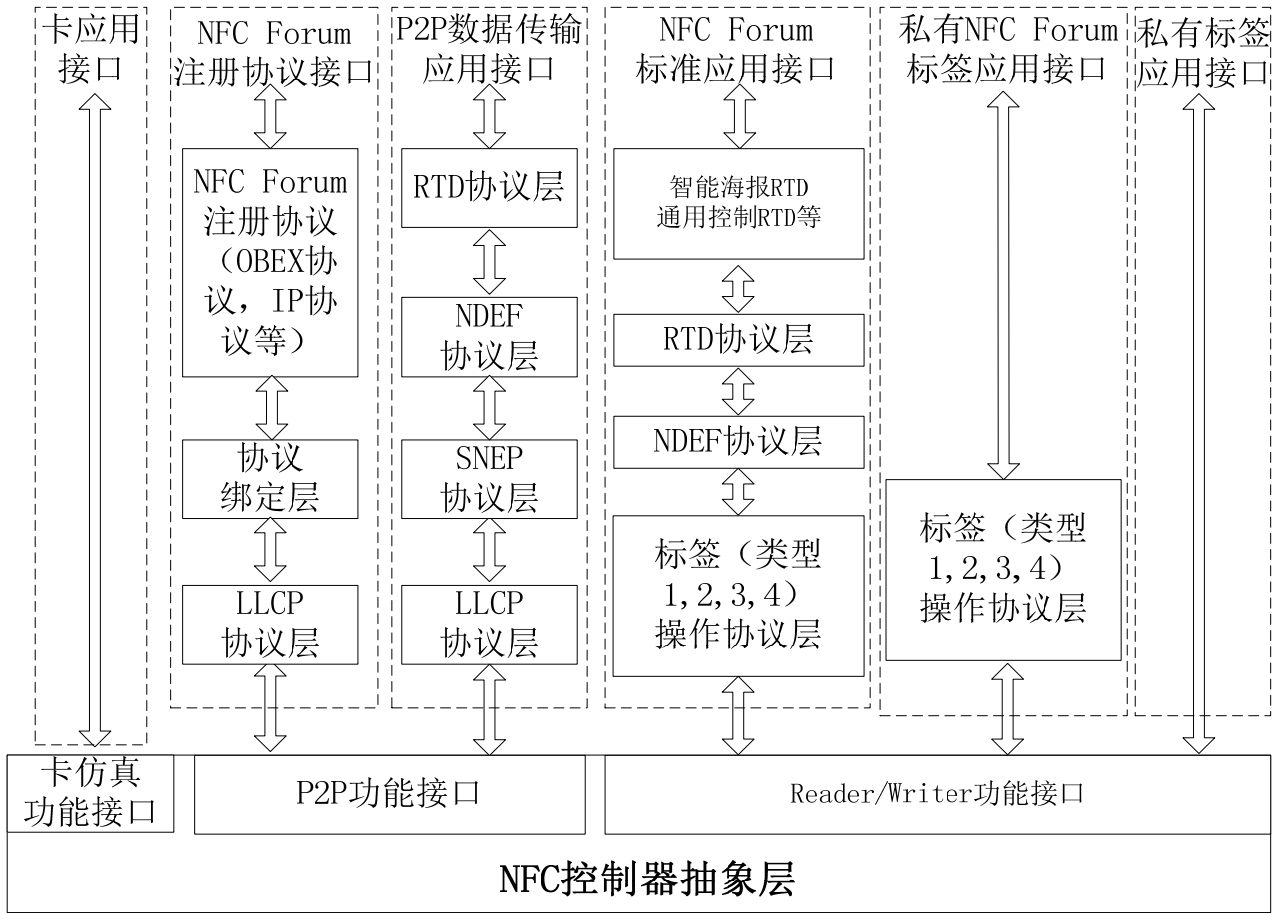


图4 NFC服务模型层架构

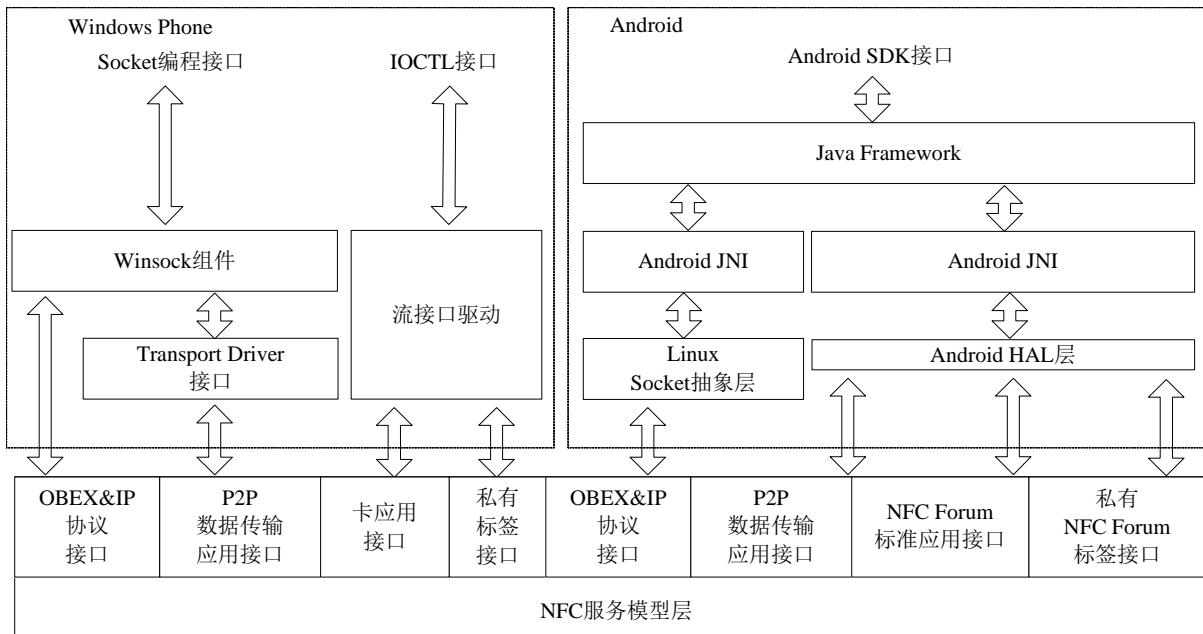


图5 NFC服务接口层架构

3 跨平台NFC软件栈的实现实例

为了验证文中提出的跨平台NFC软件栈架构的正确性与可行性,在不同的软硬件平台上对该架构进行了具体实现。选择的硬件平台分别集成了STMicroelectronics ST21NFCA、INSIDE Secure Microread和NXP Semiconductors PN544主流NFC控制器芯片;选择的操作系统包括Android 2.3.3、Android 4.0.4和Windows Phone 7.5。为了保证跨平台NFC软件栈在不同操作系统上实现源代码兼容,采用C语言作为编程语言。

NFC软件栈运行时环境的主要工作是对操作系统的基础服务接口进行封装,如任务(进程、线程)管理服务、时间(系统时间、定时器等)管理服务、内存(堆栈、堆)管理服务、文件管理服务、任务间同步机制(信号量、临界区等)、任务间通信机制(消息、管道等)及应用程序与设备驱动程序之间的接口等。

对于Android系统而言,因为跨平台NFC软件栈是通过C语言实现的,所以只能存在于本地框架之中,使用的基础服务是Linux的系统调用。以创建信号量的基础服务为例,Linux对应的系统调用是`sema_init`。其形式参数包括:指向`semaphore`结构的指针,指示信号量是由进程内线程共享,还是由进程之间共享的标志位,信号量的最大值。当该系统调用成功时返回0,错误时返回-1。

Windows Phone 7.5创建信号量的系统调用是`CreateSemaphore`,其形式参数包括指向信号量安全属性的指针(该参数在实际使用中直接被忽略)、信号量初始值、信号量最大值、信号量名字。当该系统调用成功时返回信号量句柄,错误时返回空指针。

在具体实现时,定义抽象数据结构如下:

```
typedef struct _PAL_Semaphore
{
    struct semaphore;
    uint32_t sema_handle;
}; PAL_Semaphore
```

通过`semaphore`结构和`sema_handle`指针保持与Linux、Windows Phone 7.5系统的兼容性。同时将创建信号量的操作抽象为`sys_create_semaphore`函数,其形式参数包括指向`PAL_Semaphore`结构的指针和信号量最大值。当该函数执行成功时返回0,错误时返回-1。

如前文所述,NFC硬件抽象层通过加载配置文件的方式来适应不同NFC控制器芯片在实现相关协

议时的差异。为了解决不同NFC控制器芯片设置了不同的`generic gate`的问题,在具体实现时抽象出一套标准的`generic gate`向NFC服务模型层提供NFC控制器通用功能接口。以卡仿真模式为例,针对底层采用的RF技术的不同,而设置对应的`card RF gate`与`card application gate`。

NFC服务模型层本身具有良好的跨平台性,遵循NFC Forum协议体系实现即可。在编程时注意需使用NFC软件栈运行时环境提供的抽象操作系统服务。

NFC服务模型层负责将NFC软件栈的能力整合到操作系统的SDK之中。以Windows Phone 7.5为例,实现时将基于NFC技术的OBEX、IP类服务以及P2P数据传输类服务与Winsoc组件进行集成,对应用提供标准的套接字接口、卡应用服务和私有tag服务则直接向应用提供标准Windows流设备接口。

4 结束语

针对终端厂商在将NFC功能集成到智能手机或智能终端产品时所面临的困难,本文提出了一种跨平台NFC软件栈架构,基于该架构设计的NFC软件栈可以灵活地适应不同的操作系统以及不同的NFC控制器芯片,降低终端厂商的开发难度,同时兼容基于SDK开发的第三方NFC应用。

在实际项目中,已经将基于该架构实现的NFC软件栈顺利地应用到了多种软硬件平台上。经过测试,该软件栈可以完整地支持NFC的不同操作模式,包括读卡器模式、卡仿真模式和P2P模式。

在读卡器模式下可以支持多种类型的非接卡和标签,如任何符合NFC Forum Type 1, 2, 3, 4 Tag 规范的标签、Inside PicoPass卡(2K, 32K)、NXP Mifare系列卡、Sony FeliCa卡、Innovision Topaz标签等等。在卡仿真模式下可以支持对ISO 14443 4-A卡、ISO 14443 4-B卡等的模拟。在P2P模式下可以支持ISO 18092/ECMA 340/NFCIP-1规范,支持LLCP规范中定义的有连接传输或者无连接传输,最大传输速率可达424 Kb/s,完全符合相关协议的性能要求。

参 考 文 献

- [1] 蒋华, 孙强. 近距离无线通信技术标准解析[J]. 信息技术与标准化, 2006(5): 26-30.
JIANG Hua, SUN Qiang. A consideration on near field communication technical standard[J]. Information Technology & Standardization, 2006(5): 26-30.
- [2] 游玉俊, 包志华, 曹平. ECMA-340标准介绍[J]. 电信工程技术与标准化, 2006(19): 35-38.

- YOU Yu-jun, BAO Zhi-hua, CAO Pin. Introduction of ECMA-340[J]. Telecom Engineering Technics and Standardization, 2006(19): 35-38.
- [3] 李小明. 基于NFC技术的电子钱包系统设计与安全性研究[D]. 北京: 北京邮电大学, 2007.
LI Xiao-peng. The design of e-wallet system based on nfc technology and investigation on its security[D]. Beijing: Beijing University of Posts and Telecommunications, 2007
- [4] 夏文栋, 林凯. 融合NFC的3G智能卡系统[J]. 计算机工程, 2011(37): 229-231.
XIA Wen-dong, LIN Kai. 3G smart card system fused on NFC[J]. Computer Engineering, 2011(37): 229-231
- [5] 王宇伟, 张辉. 基于手机的NFC应用研究[J]. 中国无线电, 2007(6): 3-8.
WANG Yu-wei, ZHANG Hui. Research on mobile NFC application[J]. China Radio, 2007(6): 3-8.
- [6] 蒋华, 徐晨, 袁红林. 基于NFC的移动商务模型研究[J]. 信息技术, 2006(30): 65-67.
JIANG Hua, XU Chen, YUAN Hong-lin. Model of M-Commerce based on near field communication[J]. Information Technology, 2006(30): 65-67.
- [7] 姚钢. NFC技术将经由Android平台走向普及[J]. 电子设计技术, 2011(3): 69-70.
YAO Gang. NFC technology become popular via android platform[J]. EDN China, 2011(3): 69-70.
- [8] Inside Secure. Open NFC core edition-porting guide [EB/OL]. [2012-06-12]. http://open-nfc.org/documents/SIS_NFC_0709-014%20Open%20NFC%20Core%20Edition%20-%20Porting%20Guide%20v1.9.chm.
- [9] Inside Secure. Open NFC-NFC HAL for Microread - porting guide [EB/OL]. [2012-06-12]. http://open-nfc.org/documents/SIS_NFC_1007-208%20Open%20NFC%20-%20NFC%20HAL%20for%20Microread%20-%20Porting%20Guide%20v0.6.chm.
- [10] 许海翔, 伏京生, 翟巍, 等. NFC应用的产业生态环境剖析[J]. 通信世界, 2007(48): 21-22.
XU Hai-xiang, FU Jing-shen, CUI Wei, et al. Research on NFC application industry ecosystem[J]. Communications World Weekly, 2007(48): 21-22.
- [11] NFC Forum. NFC technology overview[EB/OL]. [2012-06-14]. http://www.nfc-forum.org/events/oulu_spotlight/Technical_Architecture.pdf.
- [12] ETSI. Smart cards; UICC-contactless front-end (CLF) interface; Part 1: Physical and data link layer characteristics (Release 7)[EB/OL]. [2012-06-14]. http://www.etsi.org/deliver/etsi_ts/102600_102699/102613/07.07.00_60/ts_102613v070700p.pdf.
- [13] ETSI. Smart cards; UICC-Contactless Front-end (CLF) interface; Host Controller Interface (HCI) (Release 7) [EB/OL]. [2012-06-14]. http://www.etsi.org/deliver/etsi_ts/102600_102699/102622/07.00.00_60/ts_102622v070000p.pdf.

编辑 张俊