



(12) 发明专利

(10) 授权公告号 CN 102291797 B

(45) 授权公告日 2014. 02. 26

(21) 申请号 201110185322. 4

CN 101262652 A, 2008. 09. 10, 全文.

(22) 申请日 2011. 07. 04

WO 2005/079001 A1, 2005. 08. 25, 全文.

(73) 专利权人 东南大学

审查员 张文明

地址 211189 江苏省南京市江宁开发区东南
大学路 2 号

(72) 发明人 曹秀英 龚挺 李喆

(74) 专利代理机构 南京天翼专利代理有限责任
公司 32112

代理人 朱戈胜

(51) Int. Cl.

H04L 29/12(2006. 01)

H04W 40/24(2009. 01)

H04W 88/14(2009. 01)

(56) 对比文件

CN 101335671 A, 2008. 12. 31, 说明书第 3 页
第 5 行至第 6 页最后 1 行、附图 1-4.

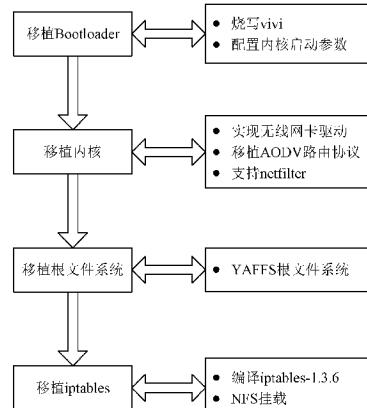
权利要求书2页 说明书5页 附图1页

(54) 发明名称

具有路由功能的无线终端设计方法

(57) 摘要

一种能够实现数据无线收发的具有路由功能的无线终端设计方法。它是在 ARM Linux 平台上搭建一个无线自组织网络硬件终端，利用 Samsung s3c2410 处理器连接支持 802.11a/b/g 的无线网卡，再加上 SDRAM、Nand flash、调试串口以及电源这些模块组成了无线终端的硬件框架。在此基础上移植了 Bootloader、Linux 内核、文件系统和 AODV 路由协议。为了更方便地演示路由协议功能，同时移植了 iptables。整个具有路由功能的无线终端设备在 802.11a/b/g 的基础上实现了路由跳转的演示和自组织功能，同时将硬件电路板进行封装，形成具有路由功能的无线终端设备。



1. 一种具有路由功能的无线终端设计方法,其特征在于,包括步骤:

步骤 1) 在 ARM Linux 平台上搭建一个无线自组织网络硬件终端,利用处理器连接支持 802.11a/b/g 的无线网卡,再加上存储接口、调试串口以及电源,构成了无线终端的硬件框架;

步骤 2) 在所述硬件框架的基础上移植 Bootloader、Linux 内核、文件系统、无线网卡驱动和 AODV 路由协议;

步骤 3) 同时移植了 iptables;

步骤 4) 最后将硬件电路板进行封装,形成具有路由功能的无线终端;

所述步骤 2) 中包括以下步骤:

21) 进行 Bootloader 移植:将 Bootloader 的镜像文件通过 JTAG 口烧写到无线终端的 Nandflash 中;

22) 进行引导参数 Boot Parameters 的设置:在 Boot parameters 分区存放可设置参数;

23) 进行 linux 内核的移植:在 Makefile 文件中修改编译工具为交叉编译工具,修改 MTD 设备分区,使得 Linux 内核可以挂载 NAND Flash 上的文件系统,在默认配置文件的基础上增加对网络文件系统 NFS 和 YAFFS 文件系统的支持;

24) 进行根文件系统移植:根文件系统选择 YAFFS 文件系统,使用 NFS 启动系统,然后使用 bs 命令对 flash 进行格式化,并安装根文件系统;

25) 移植无线网卡驱动:移植无线网卡驱动,使无线网络具有自组织的特性;

26) 移植 iptables:Linux 下支持 netfilter 机制的配置工具就是 iptables,它也就相当于一个应用程序,可以对 netfilter 进行配置;

27) 移植 AODV 路由协议:修改 Makefile 并进行编译,在当前目录下生成 aodvd 文件,在 lnx 目录中生成 kaodv. ko 文件,然后将 aodvd 和 kaodv. ko 拷贝到无线终端中。

2. 根据权利要求 1 所述的具有路由功能的无线终端设计方法,其特征在于,所述步骤 27) 中,

a、移植 AODV:Linux 的内核配置要求支持 netfilter;把 advanced router 选项也编译进 Linux 内核,使得节点当路由器使用;

在 Makefile 中指定交叉编译工具和 Linux 内核源码路径,先进行 Make 编译,在当前目录下生成 aodvd 的可执行文件,在 lnx 目录中生成 kaodv. ko 模块;然后将 aodvd、kaodv. ko 下载到无线终端;

b、移植 iptables:Linux 下支持 netfilter 机制的配置工具就是 iptables,它可以对 netfilter 进行配置;所以要实现 netfilter (iptables) 就要从使 Linux 内核支持 netfilter 和实现用户层 iptables 配置命令这两方面来着手。

3. 根据权利要求 1 所述的具有路由功能的无线终端设计方法,其特征在于,所述步骤 2) 中,AODV 路由协议是使用 AODV 路由算法建立路由,更新 IP 路由表并维护之,步骤是重新编译 Linux 内核添加 netfilter 网络支持:

利用 Linux 内核可加载模块 kaodv. ko 完成转发功能,数据分组发往网络接口前,在决定数据发送的路由时,如果 Linux 内核路由表中存在与数据分组目的地址匹配的路由,数据分组就被投递到相应的网络接口;如果 Linux 内核路由表中不存在与数据分组目的地址

匹配的路由，就会由回调函数处理，送往用户空间的数据分组缓冲区进行排队，同时用户空间进程 aodvd 启动路由查找功能；若找到与被缓存的数据分组报头目的地址匹配的路由，就将缓存在用户空间的数据分组重新注入 Linux 内核；如果没有找到与被缓存的数据分组报头目的地址匹配的路由，则缓存的数据分组将被丢弃，且释放它使用的内存空间；

所述由回调函数处理处理是，回调函数检查每条路由的使用时间，将路由表的使用状况发往用户空间，将 Linux 内核路由表的使用状况告知用户态进程 aodvd；aodvd 根据该使用状况重置路由缓冲表的定时器，同时 aodvd 删除 Linux 内核路由表中过时的路由条目或添加新的路由。

4. 根据权利要求 1 所述的具有路由功能的无线终端设计方法，其特征在于，所述步骤 25) 中，无线网络具有自组织的特性具有网络过滤机制：假如三个节点无线终端依次是 A, B 和 C；在 A 的 Linux 内核中启动网络过滤把从 C 节点收到的数据包全部滤掉，同样，在 C 的 Linux 内核中启动网络过滤把从 A 节点收到的数据包全部滤掉，也即 A 和 C 不能相互通信；然后如果在 A, B, C 节点中都启动了路由转发功能，就可以实现 A 通过 B 两跳后和 C 通信。

具有路由功能的无线终端设计方法

技术领域

[0001] 本发明涉及一种具有路由功能的无线终端设计,尤其是实现了无线自组织和路由转发功能,所属技术领域为无线通信领域。

背景技术

[0002] 无线局域网有 802.11a/b/g 三种主流的标准。802.11b 规范工作在 2.4GHz 通信频带,数据速率超过了 10M bps 的临界限度,物理层采用高速直接序列扩频技术。802.11 的第二个分支为 802.11a,802.11a 工作在 5.2G Hz 频带,并被指定高达 54Mbps 的数据速率。而 802.11g 可以在与 802.11b 网络兼容的情况下,最高提供与 802.11a 标准相同的 54Mbps 连接速率。

[0003] 目前市售的无线网卡只是单纯地支持 Infrastructure 模式,很少有支持 AdHoc 模式的产品。即使支持,也不是标准的 Ad Hoc 模式,而是简单的点对点 Ad Hoc 模式,不能实现路由转发。

[0004] 由于 Ad Hoc 网络拓扑结构动态变化,如何在移动中保持通信成为一个重要的研究方向,现阶段已经提出了许多路由算法,本设计使用被 IETF 的 MANET 工作组认为是最好的路由协议之一的 AODV 路由协议。AODV 协议采用逐跳的方式转发分组,路由表中记录了到目的节点的下一跳,不需要在报文中携带完整的路由信息,减少了数据分组额外开销,提高了网络带宽利用率。并且采用源序列号和目的序列号标识链路的新旧程度,避免了环路的产生,并且每次都使用最新的路由。此外,AODV 协议还采用了扩展环路搜索技术,通过设置路由请求信息中生存时间 TTL(Time to Live) 的值,避免了路由请求所带来的全网广播。AODV 协议通过每个节点周期性广播 HELLO 消息提供与相邻节点的相互连接信息,检测链路状态。鉴于 AODV 路由协议功能强大而又便于实现的特点,故成为嵌入式系统下移植 Ad Hoc 协议的首选。

[0005] 目前媒介接入控制层和物理层协议一般使用网络设备硬件或固件来实现,所以一旦做成产品就不能对其再进行开发

发明内容

[0006] 为了解决现有技术中存在的上述问题,本方法提出一种用 net80211 和 mac80211 来实现软件 MAC 的方法。本发明要解决的主要技术问题是完成具有路由功能的无线终端设计,使得所设计出来的无线终端能够具有 AODV 路由协议的功能和数据转发的功能,并且可以演示其路由转发功能。具体技术方案如下:

[0007] 一种具有路由功能的无线终端设计方法,包括步骤:

[0008] 步骤 1) 在 ARM Linux 平台上搭建一个无线自组织网络硬件终端,利用处理器连接支持 802.11a/b/g 的无线网卡,再加上存储接口、调试串口以及电源,构成了无线终端的硬件框架;

[0009] 步骤 2) 在所述硬件框架的基础上移植 Bootloader、Linux 内核、文件系统、无线网

卡驱动和 AODV 路由协议；

[0010] 步骤 3) 同时移植了 iptables；

[0011] 步骤 4) 最后将硬件电路板进行封装,形成具有路由功能的无线终端。

[0012] 所述步骤 2) 中包括以下步骤：

[0013] 21) 进行 Bootloader 移植 : 将 Bootloader 的镜像文件通过 JTAG 口烧写到无线终端的 Nandflash 中；

[0014] 22) 进行引导参数 Boot Parameters 的设置 : 在 Boot parameters 分区存放可设置参数；

[0015] 23) 进行 linux 内核的移植 : 在 Makefile 中修改编译工具为交叉编译工具,修改 MTD 设备分区,使得 Linux 内核可以挂载 NAND Flash 上的文件系统,在默认配置文件的基础上增加对网络文件系统 NFS 和 YAFFS 文件系统的支持；

[0016] 24) 进行根文件系统移植 : 根文件系统选择 YAFFS 文件系统, 使用 NFS 启动系统, 然后使用 bs 命令对 flash 进行格式化, 并安装根文件系统；

[0017] 25) 移植无线网卡驱动 : 移植无线网卡驱动, 使无线网络具有自组织的特性；

[0018] 26) 移植 iptables : Linux 下支持 netfilter 机制的配置工具就是 iptables, 它也就相当于一个应用程序, 可以对 netfilter 进行配置；

[0019] 27) 移植 AODV 路由协议 : 修改 Makefile 并进行编译, 编译过后将在当前目录下生成 aodvd 的可执行文件, 在 lnx 目录中将生成 kaodv. ko 模块, 然后将 aodvd 和 kaodv. ko 拷贝到无线终端中。

[0020] 所述步骤 27) 中,

[0021] a、移植 AODV :Linux 的内核配置要求支持 netfilter ; 把 advanced router 选项也编译进 Linux 内核, 使得节点当路由器使用；

[0022] 在 Makefile 中指定交叉编译工具和 Linux 内核源码路径先进行 Make 编译, 在当前目录下生成 aodvd 的可执行文件, 在 lnx 目录中生成 kaodv. ko 模块 ; 然后将 aodvd、kaodv. ko 下载到无线终端；

[0023] b、移植 iptables ;Linux 下支持 netfilter 机制的配置工具就是 iptables, 它可以对 netfilter 进行配置 ; 所以要实现 netfilter(iptables) 就要从使 Linux 内核支持 netfilter 和实现用户层 iptables 配置命令这两方面来着手。

[0024] 所述步骤 2) 中,AODV 路由协议是使用 AODV 路由算法建立路由, 更新 IP 路由表并维护之, 步骤是重新编译 Linux 内核添加 netfilter 网络支持 :

[0025] 利用 Linux 内核可加载模块 kaodv. ko 完成转发功能, 数据分组发往网络接口前, 在决定数据发送的路由时, 如果 Linux 内核路由表中存在与数据分组目的地址匹配的路由, 数据分组就被投递到相应的网络接口 ; 如果 Linux 内核路由表中不存在与数据分组目的地址匹配的路由, 就会由回调函数处理, 送往用户空间的数据分组缓冲区进行排队, 同时用户空间进程 aodvd 启动路由查找功能 ; 若找到与被缓存的数据分组报头目的地址匹配的路由, 就将缓存在用户空间的数据分组重新注入 Linux 内核 ; 如果没有找到与被缓存的数据分组报头目的地址匹配的路由, 则缓存的数据分组将被丢弃, 且释放它使用的内存空间；

[0026] 所述由回调函数处理处理是, 回调函数检查每条路由的使用时间, 将路由表的使

用状况发往用户空间,将 Linux 内核路由表的使用状况告知用户态进程 aodvd ;aodvd 根据该使用状况重置路由缓冲表的定时器,同时 aodvd 删减 Linux 内核路由表中过时的路由条目或添加新的路由。

[0027] 所述步骤 25) 中,无线网络具有自组织的特性具有网络过滤机制:,假如三个节点无线终端依次是 A, B 和 C ;在 A 的 Linux 内核中启动网络过滤把从 C 节点收到的数据包全部滤掉,同样,在 C 的 Linux 内核中启动网络过滤把从 A 节点收到的数据包全部滤掉,也即 A 和 C 不能相互通信;然后如果在 A,B,C 节点中都启动了路由转发功能,就可以实现 A 通过 B 两跳后和 C 通信。

[0028] 本技术方案中,

[0029] Makefile 文件描述了 Linux 内核源代码中文件之间的关系,提供更新每个文件的命令;

[0030] netfilter 是 Linux 内核的扩展,可以实现防火墙、报文加密、报文分类等工作;

[0031] iptables 是与最新的 2.6.x 版本 Linux 内核集成的 IP 信息包过滤系统,这里用来配置 netfilter;

[0032] AODV 路由协议的用户态模块文件。

[0033] 有益效果:本发明所设计的无线终端能够使用 802.11a/b/g 进行数据的收发,并且具有 AODV 路由协议的功能和路由转发的功能,可以在室内短距环境下进行路由转发功能的演示。

附图说明

[0034] 图 1 是系统设计过程示意图。

具体实施方式

[0035] 本实施方式公开了具有路由功能的无线终端设计方法,包括以下步骤:

[0036] 一种具有路由功能的无线终端设计方法,它是在 ARM Linux 平台上搭建一个无线自组织网络硬件终端,利用 Samsung s3c2410 处理器连接支持 802.11a/b/g 的无线网卡,再加上 SDRAM、Nand flash、调试串口以及电源这些模块组成了无线终端的硬件框架。在此基础上移植了 Bootloader、Linux 内核、文件系统和 AODV 路由协议。为了更方便地演示路由协议功能,同时移植了 iptables。整个具有路由功能的无线终端设备在 802.11a/b/g 的基础上实现了路由跳转的演示和自组织功能,同时将硬件电路板进行封装,形成具有路由功能的无线终端设备。

[0037] 所述移植包括:

[0038] a. 进行 Bootloader 移植,所移植的 Bootloader 选用韩国 MIZI 公司为 SAMSUNG 的 ARM 架构 CPU 专门设计的 ViVi,

[0039] b. 进行 Boot Parameters 的设置,Boot parameters 分区存放一些可设置的参数,比如 IP 地址、要传给 Linux 内核的命令行参数等,

[0040] c. 进行 linux-2.6.17 内核的移植,在 Makefile 中修改编译工具为交叉编译工具,修改 MTD 设备分区,使得 Linux 内核可以挂载 NAND Flash 上的文件系统,在默认配置文件的基础上增加对网络文件系统 NFS 和 YAFFS 文件系统的支持,

[0041] d. 进行根文件系统移植, 根文件系统选择 YAFFS 文件系统, 使用 NFS 启动系统, 然后使用 bs 命令对 flash 进行格式化, 并安装根文件系统,

[0042] e. 移植无线网卡驱动, 单纯移植无线网卡驱动较简单, 但不是目的, 我们最终要使无线网络具有自组织的特性, 因为要移植 AODV 路由协议, 所以 Linux 内核应支持 netfilter 机制,

[0043] f. 移植 iptables, Linux 下支持 netfilter 机制的配置工具就是 iptables, 它也就相当于一个应用程序, 可以对 netfilter 进行配置 (包过滤规则, NAT 等等),

[0044] g. 移植 AODV 路由协议, 修改 Makefile 并进行编译, 编译过后将在当前目录下生成 aodvd 的可执行文件, 在 Inx 目录中将生成 kaodv. ko 模块, 然后将 aodvd kaodv. ko 拷贝到无线终端中。

[0045] Boot Loader 是在 Linux 操作系统内核运行之前运行的一段小程序。通过这段小程序, 可以初始化硬件设备、建立内存空间的映射图, 从而将系统的软硬件环境带到一个合适的状态, 以便为最终调用 Linux 操作系统内核准备好正确的环境。

[0046] 本方法中使用了网络过滤机制, 假如三个节点依次是 A, B, C。在 A 的 Linux 内核中启动网络过滤把从 C 节点收到的数据包全部滤掉, 同样, 在 C 的 Linux 内核中启动网络过滤把从 A 节点收到的数据包全部滤掉, 也即 A 和 C 不能相互通信。如果在 A, B, C 节点中都启动了路由转发功能, 这样就可以实现 A 通过 B 两跳后和 C 通信。

[0047] 为了实现 netfilter (iptables), 设计方法从两方面来着手:

[0048] (1) Linux 内核支持 netfilter;

[0049] (2) 用户层的 iptables 配置命令, 在宿主机上进入 Linux 内核目录, 配置所需的 Linux 内核模块, 选择 Linux 内核中支持 netfilter 的选项及其相关选项, 下载 iptables 工具包, 进入下载目录, 进行编译生成可执行文件, 设计方案选择 iptables-1.3.6 进行静态编译, 将生成的 iptables 可执行文件复制到 NFS 文件系统下并进行挂载, 重新启动终端后, 就可以用 iptables 命令了。

[0050] 通过修改 Linux 内核配置和修改 Makefile 文件来实现 AODV 路由协议的编译, 选中 Linux 内核选项 :IP Userspace queueing via NETLINK (OBSOLETE), 修改 Makefile 中 Linux 内核版本号和 Linux 内核源码路径, 接着进行编译, 编译过后将在当前目录下生成 aodvd 的可执行文件, 在 Inx 目录中将生成 kaodv. ko 模块, 然后将 aodvd 和 kaodv. ko 两个文件拷贝到目标板中, 上述路由功能模块 aodvd 使用一定的路由协议算法来计算和维护 Linux 内核路由表, 这是在用户空间执行的, 而转发功能模块 kaodv. ko 在 Linux 内核中实现, 所以在转发功能模块保持不变的情况下, 可以通过修改路由功能模块, 来达到使用其它路由协议或者改进当前路由协议的目的。

[0051] 路由功能模块在用户空间运行, 负责和其它网络节点进行信息交流, 使用 AODV 路由算法建立路由更新 IP 路由表并维护之。具体流程是重新编译 Linux 内核添加 netfilter 网络支持利用 Linux 内核可加载模块 kaodv. ko 完成转发功能, 数据分组发往网络接口前, 在决定其路由时, 如果 Linux 内核路由表中存在与数据分组目的地址匹配的路由, 数据分组就被投递到相应的网络接口; 若不存在, 就会由回调函数处理, 送往用户空间的数据分组缓冲区进行排队, 同时用户空间进程 aodvd 启动路由查找功能, 若查找到与被缓存的数据分组报头目的地址匹配的路由, 就将缓存在用户空间的数据分组重新注入 Linux 内核; 如

果路由未被发现，则缓存的数据分组将被丢弃，且释放它使用的内存空间。回调函数检查每条路由的使用时间，将路由表的使用状况发往用户空间，将 Linux 内核路由表的使用状况告知用户态进程 aodvd。aodvd 据此重置路由缓冲表的定时器，同时 aodvd 删除 Linux 内核路由表中过时的路由条目或添加新的路由。

[0052] 首先把 aodv-uu-0.9.5 移植到 ARM Linux 中，Linux 内核配置要求支持 netfilter，用来在几个特殊的地方抓包，把 advanced router 选项也编译进 Linux 内核，使得节点当路由器使用，这样就可以中转数据包了。在 Makefile 中指定交叉编译工具和 Linux 内核源码路径。Make 编译后将在当前目录下生成 aodvd 的可执行文件，在 lnx 目录中生成 kaodv.ko 模块，然后将 aodvd、kaodv.ko 下载到无线终端。

[0053] 接下来移植 iptables，移植 iptables 的目的是为了测试，因为在室内短距环境下进行节点间多跳路由的测试是比较难实现的。Linux 下支持 netfilter 机制的配置工具就是 iptables，它也就相当于一个应用程序，可以对 netfilter 进行配置（包括过滤规则，NAT 等等）。所以要实现 netfilter(iptables) 就要从使 Linux 内核支持 netfilter 和实现用户层 iptables 配置命令这两方面来着手。

[0054] (1) 编译 Linux 内核，支持 netfilter

[0055] 在宿主机上进入 Linux 内核目录，配置所需的 Linux 内核模块：

[0056] make menuconfig

[0057] 选中如下 Linux 内核选项：

[0058] General setup--->

[0059] [*] Sysctl support(在 ROMFS 文件系统中 /proc/sys/net/ipv4/ 出现 ip_forward)

[0060] Networking options--->

[0061] [*] Network packet filtering(replaces ipchains)

[0062] IP :Netfilter Configuration---> (全部选择即可)

[0063] 这样 Linux 内核就支持了 netfilter。接下来只需编译并生成 Linux 内核镜像文件并烧写到嵌入式系统即可。如果烧写后重起成功进入 Linux，则说明新的支持 netfilter 的 Linux 内核已经正常运行。

[0064] (2) 编译生成 iptables 命令

[0065] 下载 iptables 工具包，进入下载目录，进行编译生成可执行文件，编译方法如下：

[0066] 选用 iptables-1.3.6

[0067] cd/\$DIR/iptables-1.3.6

[0068] vim Makefile

[0069] line14 :NO_SHARED_LIBS = 1(静态链接编译生成可执行文件)

[0070] make

[0071] 执行上述步骤之后便可生成 iptables 可执行文件，把生成的 iptables 可执行文件进行 NFS 挂载，设置参数后重新启动无线终端，就可以用 iptables 命令了。

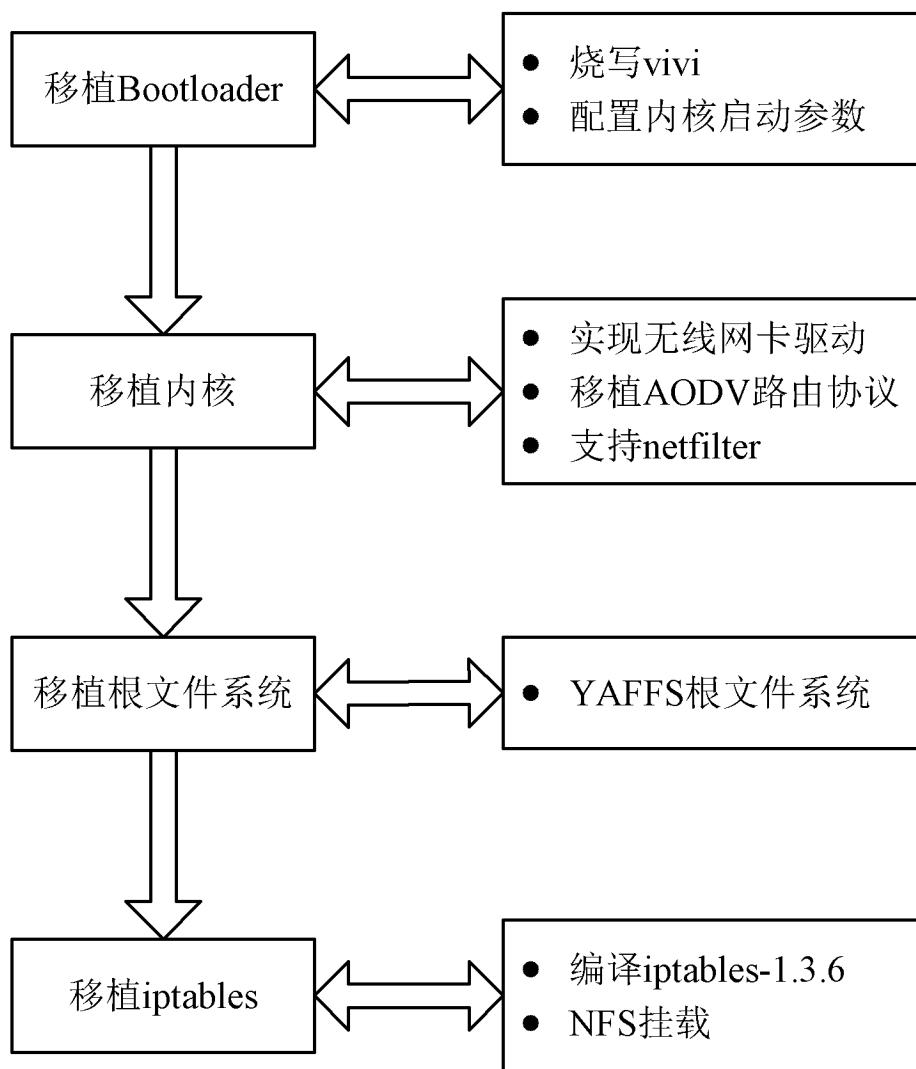


图 1