(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 105744011 A
(43)申请公布日 2016.07.06

(21)申请号 201610049613.3

(22)申请日 2016.01.25

(71)申请人 中国科学技术大学
地址 230026 安徽省合肥市包河区金寨路96号

(72)发明人 凌强 张雷 徐骏

(74)专利代理机构 北京科迪生专利代理有限责任公司 11251
代理人 杨学明 顾炜

(51)Int.Cl.
H04L 29/12(2006.01)

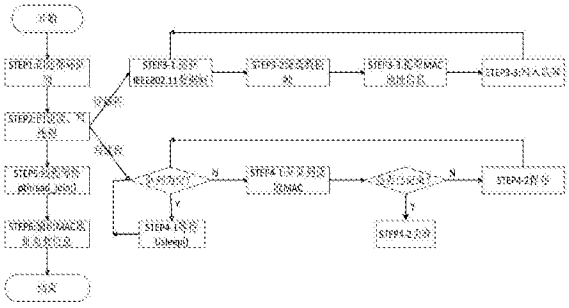
权利要求书1页 说明书4页 附图2页

(54)发明名称

一种基于Openwrt路由器的MAC地址扫描的方法

(57)摘要

本发明公开了一种基于Openwrt路由器的MAC地址扫描的方法，第一步骤，数据帧捕获处理，通过抓包获取IEEE802.11协议的原始无线数据帧，并通过分析、处理得到指定类型数据帧以及该帧中包含的设备MAC地址信息；第二步骤，MAC地址处理，通过设计循环队列，将首次出现的MAC地址保存到指定文件中，重复的MAC丢弃，从而实现MAC地址的不重复保存和统计；第三步骤，交叉编译，通过使用Openwrt系统对应的SDK对源码进行交叉编译，生成在路由器上可执行的文件。本发明该方法的实现不需要额外的设备支持，可以直接在基于Openwrt的路由器上执行，同时不影响路由器的正常工作；多线程和循环队列的方案设计，可以保证数据的快速处理。



CN 105744011 A

1.一种基于Openwrt路由器的MAC地址扫描的方法，其特征在于：该方法包括如下步骤：

第一步骤，数据帧捕获处理：通过抓包获取IEEE802.11协议的原始无线数据帧，并通过分析、处理得到指定类型数据帧以及该帧中包含的设备MAC地址信息；

第二步骤，MAC地址处理：通过设计循环队列，将首次出现的MAC地址保存到指定文件中，重复的MAC丢弃，从而实现MAC地址的不重复保存和统计；

第三步骤，交叉编译：通过使用Openwrt系统对应的SDK对源码进行交叉编译，生成在路由器上可执行的文件。

2.根据权利要求1所述的一种基于Openwrt路由器的MAC地址扫描的方法，其特征在于：第一步骤中通过Libpcap库实现对IEEE802.11数据帧的捕获，同时根据IEEE802.11数据帧格式建立相应的结构体，通过强制类型转换实现对数据帧头信息的提取，从而完成对数据帧的处理分析和相关MAC地址信息的获取。

3.根据权利要求1所述的一种基于Openwrt路由器的MAC地址扫描的方法，其特征在于：第二步骤中采用双线程设计，读进程负责完成IEEE802.11数据帧的捕获和MAC地址的提取，写进程负责完成MAC地址的处理和保存到指定文件；同时建立循环队列，由读进程负责在队尾进行写MAC地址操作，由写进程负责从队头读取已捕获到的MAC地址进行处理；在不需要加锁的情况下，实现读、写进程的同步工作，加快了数据的处理速度。

4.根据权利要求1所述的一种基于Openwrt路由器的MAC地址扫描的方法，其特征在于：第三步骤中使用Openwrt对应SDK对源码进行交叉编译，得到路由器下的可执行文件，通过在路由器端执行该文件，得到一定范围内MAC地址信息和开启WiFi的设备总数信息。

一种基于Openwrt路由器的MAC地址扫描的方法

技术领域

[0001] 本发明涉及智能路由器、嵌入式软件设计的技术领域，具体涉及一种基于Openwrt路由器的MAC地址扫描的方法。

背景技术

[0002] 智能路由器是可以智能化管理的路由器，除了具有普通无线路由器的功能之外，还具有独立的操作系统，用户可以安装各种应用，满足多样化的需求。为了充分利用路由器的功能，在不影响路由器正常使用的前提下，在Openwrt路由器上开发MAC地址的扫描功能，实现获取一定范围内所有开启了WiFi设备MAC地址信息的功能。考虑到手机设备的普及，利用设备MAC信息可以估算得到一定时间内本路由器覆盖范围内的WiFi设备的总数信息，实现了客流量数据的快速统计。由于MAC地址的收集和统计过程在路由器上完成，并且不需要用户的任何操作，十分方便快捷。同时由于MAC地址在设备端是唯一的，在一些简单的验证场合还可以作为验证凭证，简化验证步骤，为安装路由器的商家和WiFi的使用者提供便利。

[0003] 现有的抓包方案主要都是在windows和Linux平台下开发使用的，多数采用的是基于Libpcap函数库实现的。Libpcap是一个支持多种操作系统的数据包捕获函数库，可以是实现对网络数据的监听功能，因此被广泛研究和应用。文献【1】(赵新辉，李祥.捕获网络数据包的方法[J].计算机应用研究，2004，21(8):242-243.)介绍了使用原始套接字、Libpcap和Winpacp等多种方式实现网络数据抓包的基本原理。文献【2】(刘斌，代素环.基于Libpcap的数据包捕获机制的实现[J].农业网络信息，2008(9):62-63.)在Linux平台上设计实现了一个基于Libpcap的小型网络数据包捕获分析器，给出了具体使用方法和实现步骤。文献【3】(寇应展，杨素敏，陈利军，等.基于Libpcap网络数据包捕获技术的改进[J].军械工程学院学报，2011，23(3):49-51.)提出了Linux下基于Libpcap的数据包捕获的性能“瓶颈”问题，给出了基于零拷贝技术、内存映射技术的优化方案，并实验证明了其优化效果。除了应用于有线网之外，Libpcap还可以应用于无线网络环境下，文献【4】(刘敏，朱志祥.基于Linux的无线网络监听技术[J].西安邮电学院学报，2011，16(3):65-68.)提出了Linux平台下基于Libpcap的无线网络数据包捕获方案设计，文献【5】(袁泽宇，肖庆正.基于Libpcap的无线网络数据分解与重构技术研究[J].无线互联科技，2014(7):93-95.)则提出了无线网络下，基于Libpcap实现数据的分解和重构，使得Libpcap不再只是局限于抓包功能。在多种平台的应用上，文献【6】(曾敏，李峰.嵌入式数据包捕获器的设计与实现[J].计算机工程与设计，2009(7):1571-1573.)提出了移植方案，将Libpcap运用到了嵌入式平台上，使Libpcap的应用范围更加广泛了，同样也为我们的设计提供了借鉴。

发明内容

[0004] 本发明的目的为:1)本发明在Openwrt路由器上使用Libpcap实现IEEE802.11协议数据帧的捕获;2)本发明可以实现IEEE802.11数据帧帧头信息的提取和帧具体类型的判定;3)本发明针对一定范围内，已经开启了WiFi功能但是尚未连接路由器的设备，可以获取

其MAC地址;4)本发明设计双线程和循环队列,在Openwrt路由器上实现多线程的数据处理。

[0005] 本发明采用的技术方案为:1、一种基于Openwrt路由器的MAC地址扫描的方法,其特征在于:该方法包括如下步骤:

[0006] 第一步骤,数据帧捕获处理:通过抓包获取IEEE802.11协议的原始无线数据帧,并通过分析、处理得到指定类型数据帧以及该帧中包含的设备MAC地址信息;

[0007] 第二步骤,MAC地址处理:通过设计循环队列,将首次出现的MAC地址保存到指定文件中,从而实现MAC地址的不重复保存和统计;

[0008] 第三步骤,交叉编译:通过使用Openwrt系统对应的SDK对源码进行交叉编译,生成在路由器上可执行的文件。

[0009] 其中,第一步骤中通过Libpcap库实现对IEEE802.11数据帧的捕获,同时根据IEEE802.11数据帧格式建立相应的结构体,通过强制类型转换实现对数据帧头信息的提取,从而完成对数据帧的处理分析和相关MAC地址信息的获取。

[0010] 其中,第二步骤中采用双线程设计,读进程负责完成IEEE802.11数据帧的捕获和MAC地址的提取,写进程负责完成MAC地址的处理和保存到指定文件;同时建立循环队列,由读进程负责在队尾进行写MAC地址操作,由写进程负责从队头读取已捕获到的MAC地址进行处理。在不需要加锁的情况下,实现读、写进程的同步工作,加快了数据的处理速度。

[0011] 其中,第三步骤中使用Openwrt对应SDK对源码进行交叉编译,得到路由器下的可执行文件,通过在路由器端执行该文件,得到一定范围内MAC地址信息和开启WiFi的设备总数信息。

[0012] 本发明与现有技术相比的优点在于:

[0013] (1)、本发明该方法的实现不需要额外的设备支持,可以直接在基于Openwrt的路由器上执行,同时不影响路由器的正常工作;

[0014] (2)、本发明多线程和循环队列的方案设计,可以保证数据的快速处理;

[0015] (3)、本发明未连接路由器的智能设备在开启WiFi的情况下,也可以获取其MAC地址。通过剔除重复数据,从而实现通过设备MAC地址数量对一定范围内设备总数的统计。

附图说明

[0016] 图1为基于Openwrt路由器的MAC地址扫描程序流程图;

[0017] 图2为WiFi设备和热点AP的连接过程图;

[0018] 图3为数据帧过滤过程图;

[0019] 图4为双线程工作流程图。

具体实施方式

[0020] 下面结合附图以及具体实施例进一步说明本发明。

[0021] 本发明的技术方案分成三个部分,第一部分为数据帧捕获处理阶段,通过抓包获取IEEE802.11协议的原始无线数据帧,并通过分析、处理得到指定类型数据帧以及该帧中包含的设备MAC地址信息;第二部分为MAC地址处理阶段,通过设计循环队列,将首次出现的MAC地址保存到指定文件中,重复的MAC丢弃,从而实现MAC地址的不重复保存和统计;第三部分为交叉编译阶段,通过使用Openwrt系统对应的SDK对源码进行交叉编译,生成在路由

器上可执行的文件。

[0022] 1)无线数据帧捕获处理

[0023] 通过Libpcap库实现对IEEE802.11数据帧的捕获，同时根据IEEE802.11数据帧格式建立相应的结构体，通过强制类型转换实现对数据帧头信息的提取，从而完成对数据帧的处理分析和相关MAC地址信息的获取。

[0024] 2)MAC地址处理

[0025] 采用双线程设计，读进程负责完成IEEE802.11数据帧的捕获和MAC地址的提取，写进程负责完成MAC地址的处理和保存到指定文件；同时建立循环队列，由读进程负责在队尾进行写MAC地址操作，由写进程负责从队头读取已捕获到的MAC地址进行处理。在不需要加锁的情况下，实现读、写进程的同步工作，加快了数据的处理速度。

[0026] 3)交叉编译

[0027] 使用Openwrt对应SDK对源码进行交叉编译，得到路由器下的可执行文件。通过在路由器端执行该文件，得到一定范围内MAC地址信息和开启WiFi的设备总数信息。

[0028] 本发明的程序流程图如图1，就具体如下：

[0029] STEP1)：在主线程中创建循环队列，用于保存抓包得到的MAC地址信息。由于队列操作是尾进头出，可以保证双线程的并行操作。

[0030] STEP2)：创建两个线程，命名为读线程和写线程，双线程并行处理数据。

[0031] STEP3)：该线程为读线程，实现MAC的获取，具体工作流程如下：

[0032] STEP3-1)：基于Libpcap完成对IEEE802.11数据帧的捕获，通过分析帧头控制域的Type字段和Subtype字段，得到指定类型的数据帧，即Probe Request广播帧。

[0033] STEP3-2)：分析Probe Request帧，从数据帧指定位置提取出MAC地址。

[0034] STEP3-3)：将该MAC地址保存到循环队列中，供其他线程使用，并继续捕获IEEE802.11数据帧，循环STEP3的过程。

[0035] STEP4)：该线程为写线程，处理重复的MAC地址，完成MAC地址数据的保存。

[0036] STEP4-1)：在循环队列中有数据的情况下，读取数据；否则若队列为空，则等待一定时间后再进行读取。

[0037] STEP4-2)：将从队列中读取的MAC地址信息与以保存文件中的MAC地址进行比对，若该MAC已经保存，则丢弃；否则写入到文件末尾。然后继续读取队列，重复STEP4。

[0038] STEP5)：等待线程结束，回收线程资源。

[0039] STEP6)：从保存文件中统计MAC地址总数信息。

[0040] 在Openwrt路由器上，采用Libpcap实现对指定类型数据帧的捕获，并从相关数据帧中得到设备的MAC地址信息。通过与已保存的MAC进行比对，将已存在的MAC地址丢弃，将新出现的MAC地址写入文件保存。统计MAC地址即可以实现对一定范围内WiFi设备总数的估计。整个处理过程为了加快速度，采用多线程实现。

[0041] 1.WiFi的连接过程

[0042] 手机等WiFi设备为了获得附近范围内的热点信息，一般有两种方式：主动扫描和被动扫描。主动扫描就是WiFi设备(STA)在每个信道上主动发送Probe Request广播帧，向无线热点(AP)请求热点信息；被动扫描则不会发送数据包，而依靠接受AP发送的Beacon广播帧获得热点信息。因为被动扫描模式时，STA不发送数据，与AP没有交互，故无法获取MAC

地址。我们主要处理主动扫描的情况，这种情况比较普遍。

[0043] STA与AP建立连接过程如下图2所示：

[0044] STA以主动请求的方式在每个信道上发出Probe Request广播帧，请求获得有效范围无线网络热点(AP)的基本信息。当AP收到该Probe Request帧后，会回应一个Probe Response帧，该帧包含AP的SSID、MAC地址等相关信息，从而STA就获得了附近范围内所有wifi列表。STA接入某个AP之前，要先进行包括身份、认证方式、密钥等信息的交互，该过程是通过STA和AP彼此发送认证帧——Authentication实现；为了实现数据的收发，STA必须和AP进行关联，该过程是通过Association Request/Response的交互实现的。

[0045] 由于Probe Request帧中包含了设备端的MAC地址信息，同时又是广播帧可以被任何支持IEEE802.11协议的设备接收，故该帧就是我们的捕获和分析目标。

[0046] 2.数据帧的过滤和MAC地址提取

[0047] 在路由器端采用Libpcap对数据链路层的数据帧进行监听捕获。为了不影响路由器其他功能的正常工作，我们对路由器网卡开启镜像，将镜像设置为监听模式，这样，流经路由器的所有数据包都被复制了一份，交由镜像处理，为我们获取MAC信息打下了基础。

[0048] 由于数据众多，我们采用逐层过滤的方法进行数据的提取。过滤过程如下图3所示。

[0049] 首先，我们需要的Probe Request数据帧是广播帧，在使用Libpcap进行抓包时，我们使用其自带的过滤器，设置过滤条件为“Broadcast”，将大部分不需要的数据帧过滤掉。

[0050] 其次，利用数据封装协议格式特征准确的识别出Probe Request帧。在无线局域网内，数据通讯通常采用IEEE802.11协议进行数据包的封装，数据类型主要分成三种：数据帧，控制帧和管理帧，通过帧头控制域的Type字段来进行区分，同时，采用Subtype字段区分数据帧的具体类型。这里，我们需要的Probe Request帧是属于管理帧(Type＝00，SubType＝0100)。

[0051] 针对IEEE802.11协议，根据封装格式建立结构体radiotap_header和IEEE80211_header，对应数据头。提取Libpcap捕获数据帧的帧头信息，并与将值赋给结构体内变量，通过对Type和SubType的字段的匹配，实现对Probe Request帧的过滤提取，也实现了对数据帧中MAC地址的提取。

[0052] 3.MAC地址多线程处理

[0053] 为了加快数据的处理速度，采用多线程实现数据的处理，如下图4所示。

[0054] 建立两个线程A和B，线程A负责抓包、过滤和MAC地址的提取；线程B负责将A获得的MAC地址与已经保存的MAC进行比对，若已存在则丢弃，否则写入文件保存。考虑线程间数据的交互，建立循环队列，由A线程负责将获得的MAC地址写入队尾，B线程则从队头取MAC地址与已经保存的MAC地址进行比对操作。由于A，B线程对本队列的操作是同时进行的，并且是无锁的，所以操作效率比单线程方式获得了提升。

[0055] 由于一定范围内，开启了WiFi功能的设备的MAC地址都被扫描记录了下来，并且通过比对处理，重复捕获的MAC地址都没剔除了，在指定文件内保存了不重复的所有的MAC地址信息。同时，通过统计该MAC地址数量，实现对一定时间内的WiFi设备总数的估计。

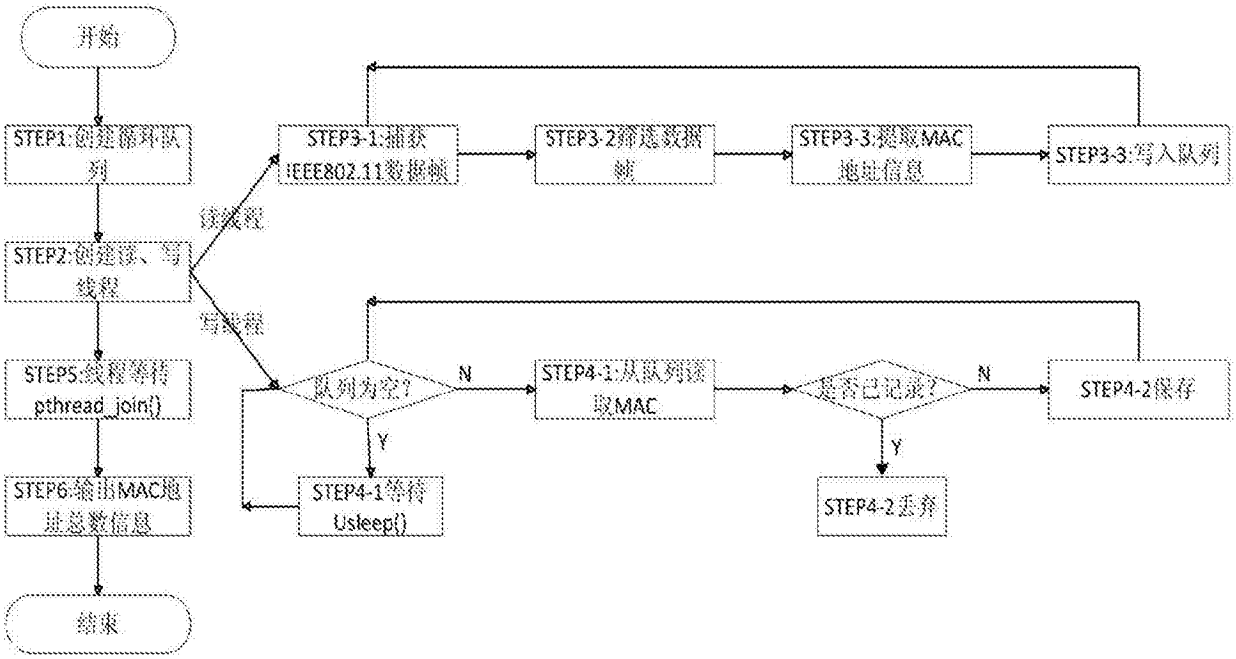

开始
STEP1:创建循环队列
STEP2:创建读、写线程
STEP3-1:捕获IEEE802.11数据帧
STEP3-2筛选数据帧
STEP3-3:提取MAC地址信息
STEP3-3:写入队列
读线程
写线程
STEP5:线程等待pthread_join()
队列为空?
N
STEP4-1:从队列读取MAC
是否已记录?
N
STEP4-2保存
Y
Y
STEP6:输出MAC地址总数信息
STEP4-1等待Usleep()
STEP4-2丢弃
结束

图1

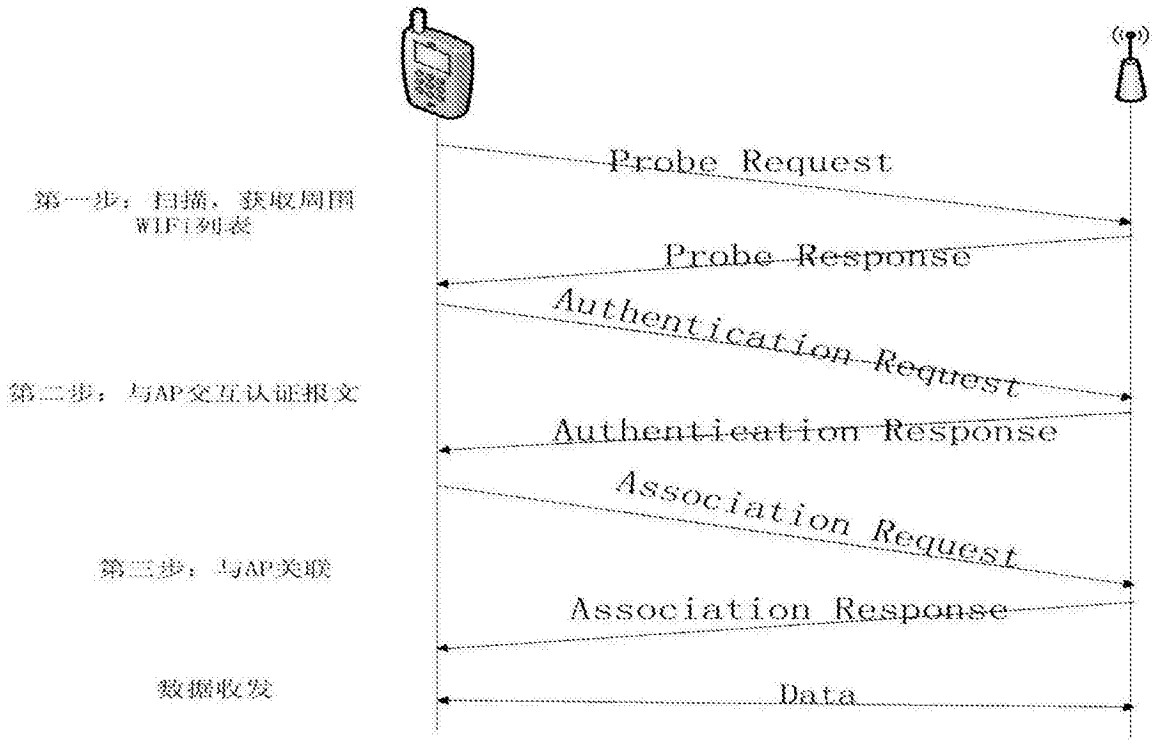

第一步：扫描，获取周围WIFI列表
第二步：与AP交互认证报文
第三步：与AP关联
数据收发
Probe Request
Probe Response
Authentication Request
Authentication Response
Association Request
Association Response
Data

图2

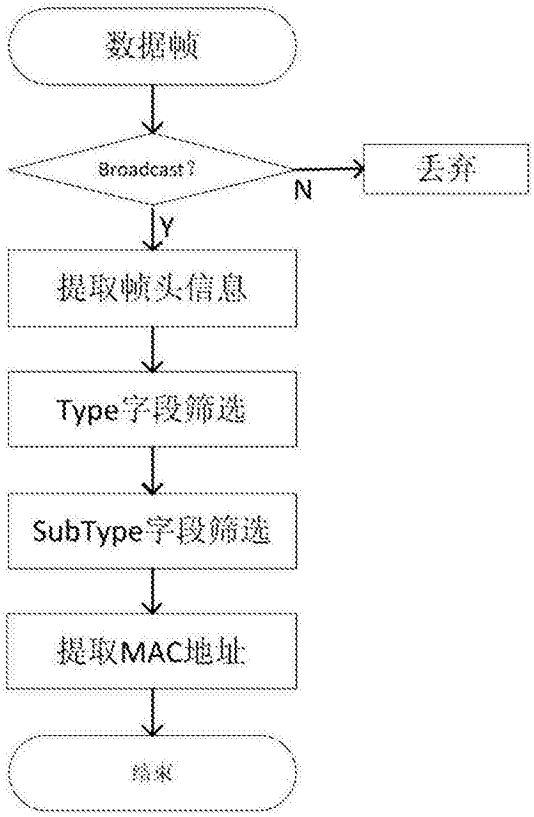

数据帧
Broadcast?
N
丢弃
Y
提取帧头信息
Type字段筛选
SubType字段筛选
提取MAC地址
结束

图3

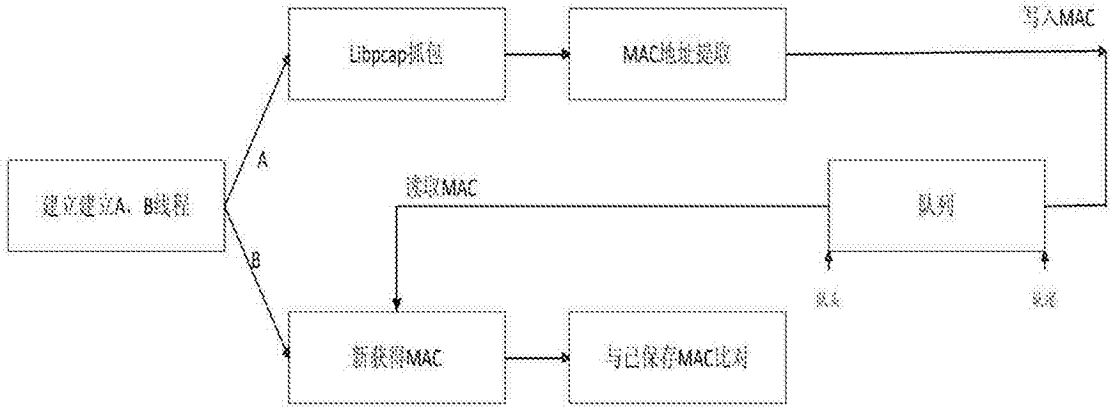

Libpcap抓包
MAC地址提取
写入MAC
建立建立A、B线程
A
B
读取MAC
队列
获得MAC
与已保存MAC比对

图4