



(12)发明专利

(10)授权公告号 CN 103916405 B

(45)授权公告日 2017.02.22

(21)申请号 201410168823.5

(22)申请日 2014.04.25

(65)同一申请的已公布的文献号

申请公布号 CN 103916405 A

(43)申请公布日 2014.07.09

(73)专利权人 厦门享游网络科技有限公司

地址 361000 福建省厦门市思明区东浦路
22号二楼M17室

(72)发明人 兰劲

(74)专利代理机构 厦门市精诚新创知识产权代
理有限公司 35218

代理人 方惠春

(51)Int.Cl.

H04L 29/06(2006.01)

H04L 29/08(2006.01)

(56)对比文件

US 2012005369 A1, 2012.01.05,

US 2008276297 A1, 2008.11.06,

CN 103546487 A, 2014.01.29,

CN 103532972 A, 2014.01.22,

审查员 谭美玲

权利要求书1页 说明书5页

(54)发明名称

一种IOS上针对App的TCP/UDP流量引导的方法

(57)摘要

本发明公开一种IOS上针对App的TCP/UDP流量引导的方法,首先拦截原生Socket的各个函数,然后将APP的TCP/UDP流量引到远端的代理服务器,其中,远端的代理服务器主要转发APP发过来的数据包,再将数据包发送到APP请求的源IP地址。本发明揭示了IOS上在App所有调用情况下都能够引导且只引导感兴趣的TCP&UDP流量的方法,从而解决了IOS上使用VPN只能进行全局代理、使用标准的Socks5代理在使用原生Socket的情况下无效的问题。

1. 一种IOS上针对App的TCP/UDP流量引导的方法,包括以下步骤:

步骤1:建立本地代理服务;具体包括以下内容:获取各个Socket函数的地址,将该函数的地址替换为构造的代理函数中的函数地址;然后创建本地代理服务线程;

步骤2:构造连接函数集,该连接函数集实现跟原生Socket的各个函数的相同的功能,该连接函数集至少包括:创建与指定目的地的连接的函数my_connect、用于向一个已经连接的Socket发送数据的函数my_send、用于接收一个已经连接的Socket发送过来的数据的函数my_receive、向一指定目的地发送数据的函数my_sendto、接收指定目的地发送过来的数据的函数my_receivefrom;

步骤3:将产生网络访问的应用软件记为APP,查找APP在内存当中的镜像头部,找到创建与指定目的地的连接的函数,将该函数的地址替换步骤2中构造的连接函数集的地址,并保存旧地址ori_connect;

步骤4:循环步骤3,依次替换连接函数集的所有函数的地址,并保存各旧地址;

步骤5:检测APP的协议是TCP还是UDP,如果是TCP,则选择方案一或者方案二;如果是UDP,则选择方案二将流量引导到代理服务器:

方案一:

过程1:APP向服务器发送TCP连接请求,调用my_connect函数;

过程2:my_connect函数通过其函数内设置的目的IP地址和端口,与本地代理服务建立通信连接;同时,本地代理服务与目的服务器建立通信连接;

过程3:APP通过my_send函数发送TCP数据;my_send函数将TCP数据发送到本地代理服务的代理服务器;

过程4:本地代理服务解析接收到的TCP数据,将TCP数据转发给APP实际要发送的目的服务器;目的服务器接收到TCP数据后进行处理,返回结果数据,并把结果数据发送至本地代理服务;

过程5:本地代理服务接收目的服务器发送的结果数据,并将结果数据转发至APP;

过程6:APP调用my_receive函数接收结果数据;my_receive函数接收本地代理服务发送过来的数据;

方案二:

过程1:APP向服务器发送UDP数据,调用my_sendto函数发送UDP数据;

过程2:my_sendto函数通过其函数内设置的目的IP地址和端口,与本地代理服务建立通信连接,并将UDP数据发送至本地代理服务;

过程3:同时,本地代理服务与目的服务器建立通信连接;本地代理服务将UDP数据转发至目的服务器;目的服务器接收到UDP数据后进行处理,返回结果数据,并把结果数据发送至本地代理服务;

过程5:本地代理服务接收目的服务器发送的结果数据,并将结果数据转发至APP;

过程6:APP调用my_receivefrom函数接收结果数据;my_receivefrom函数接收本地代理服务发送过来的数据。

一种IOS上针对App的TCP/UDP流量引导的方法

技术领域

[0001] 本发明涉及IOS上TCP和UDP流量引导,尤其作用范围是App而非全局的。

背景技术

[0002] 目前的数据传输协议一般包括TCP(Transmission Control Protocol,传输控制协议)和UDP(User Data Protocol,用户数据报协议)。TCP是一种传输控制协议,提供的是面向连接、可靠的字节流服务。当客户和服务端彼此交换数据前,必须先在双方之间建立一个TCP连接,之后才能传输数据。TCP提供超时重发,丢弃重复数据,检验数据,流量控制等功能,保证数据能从一端传到另一端。UDP是一种用户数据报协议,是一个简单的面向数据报的运输层协议。UDP不提供可靠性,它只是把应用程序传给IP层的数据报发送出去,但是并不能保证它们能到达目的地。由于UDP在传输数据报前不用在客户和服务端之间建立一个连接,且没有超时重发等机制,故而传输速度很快。

[0003] 目前IOS上能够对TCP及UDP都进行流量引导的只有Socks5和VPN。其中,VPN代理只能够在系统级别上,对所有的流量都进行引导,不能只引导感兴趣的流量。而使用IOS的API设置Socks5代理,只能够在使用IOS较上层的API才能够使Socks5代理生效,使用原生Socket的TCP/UDP访问没办法使用当前设置的Socks代理。

发明内容

[0004] 因此,针对上述的问题,本发明提出一种IOS上针对App的TCP/UDP流量引导的方法,解决IOS上使用VPN只能进行全局代理、使用标准的Socks5代理在使用原生Socket的情况下无效的问题。

[0005] 为了解决上述技术问题,本发明所采用的技术方案是,提供一种IOS上针对APP(产生网络流量的应用软件)的TCP/UDP流量引导的方法,首先拦截原生Socket的各个函数,然后将APP的TCP/UDP流量引到远端的代理服务器,其中,远端的代理服务器主要转发APP发过来的数据包,再将数据包发送到APP请求的源IP地址。

[0006] 具体的,本发明的一种IOS上针对App的TCP/UDP流量引导的方法包括拦截原生Socket的各个函数以及将APP的TCP/UDP流量引到远端的代理服务器的步骤,具体由以下步骤组成:

[0007] 步骤1:建立本地代理服务;具体包括以下内容:获取各个Socket函数(_connect、_send、_receive等函数)的地址,将该函数的地址替换构造代理函数中的函数地址(替换原生Socket函数connect、send、receive等函数为自定义的my_connect、my_send、my_receive函数);然后创建本地代理服务线程即可。

[0008] 步骤2:构造连接函数集,该连接函数集实现跟原生Socket的各个函数的相同的功能,该连接函数集至少包括:创建与指定目的地的连接的函数my_connect、用于向一个已经连接的Socket发送数据的函数my_send、用于接收一个已经连接的Socket发送过来的数据的函数my_receive、向一指定目的地发送数据(适用于发送未建立连接的UDP数据报)的函

数my_sendto、接收指定目的地发送过来的数据(适用于接收未建立连接的UDP数据报)的函数my_receivefrom。

[0009] 步骤3:将产生网络访问的应用软件记为APP,查找APP在内存当中的镜像头部,找到创建与指定目的地的连接的函数(函数名字为“_connect”,函数功能:创建与指定目的地的连接),将该函数的地址替换步骤2中构造的连接函数集的地址(建立连接时的目标IP和端口)。具体做法是找到函数名字为“_connect”(函数功能:创建与指定目的地的连接)的地址,将此地址替换为自己实现的函数my_connect的地址,并保存旧地址ori_connect。

[0010] 步骤4:循环步骤3,依次替换连接函数集的所有函数的地址(也即步骤2中实现的与各原生Socket各个函数相同功能的函数)的地址,并保存旧地址。

[0011] 步骤5:检测APP的协议是TCP还是UDP,如果是TCP,则选择方案一或者方案二;如果是UDP,则选择方案二将流量引导到代理服务器。

[0012] 方案一:

[0013] 采用HTTPS代理,向代理服务器发送一个HTTPS连接请求的数据包,包含真实的目的IP地址和端口(“Connect IP:PORTHTTP/1.1\r\n\r\n”,这里需要HTTP代理服务器并支持HTTPS代理),然后等待代理服务端返回数据,如果代理服务器与真实服务器连接成功则会接收到连接成功的响应数据包(“200Connectionestablished”),my_connect返回成功(返回值为0),表示连接已经建立,否则返回失败。

[0014] 方案二:

[0015] 采用socks代理,socks的方案在移动网络下比较差,协商过程太复杂,容易超时。因此,使用优化过的socks代理,将协商过程简化。

[0016] 具体的,所述方案一包括如下过程:

[0017] 过程1: APP向服务器发送TCP连接请求,调用my_connect函数(调用_connect函数会自动自定向到my_connect函数);

[0018] 过程2:my_connect函数通过其函数内设置的目的IP地址和端口(替换原来服务器的目的IP地址和端口),与本地代理服务建立通信连接;同时,本地代理服务与目的服务器(APP实际要发送的服务器)建立通信连接;

[0019] 过程3: APP通过my_send函数(调用_send函数会自动自定向到my_send函数)发送TCP数据;my_send函数(使用本地代理服务的目的IP地址和端口替换服务器的目的IP地址和端口),将TCP数据发送到本地代理服务的代理服务器;

[0020] 过程4:本地代理服务解析接收到的TCP数据,将TCP数据转发给APP实际要发送的目的服务器;目的服务器接收到TCP数据后进行处理,返回结果数据,并把结果数据发送至本地代理服务;

[0021] 过程5:本地代理服务接收目的服务器发送的结果数据,并将结果数据转发至APP;

[0022] 过程6: APP调用my_receive函数(调用_receive函数会自动自定向到my_receive函数)接收结果数据;my_receive函数(使用本地代理服务的目的IP地址和端口替换服务器的目的IP地址和端口)接收本地代理服务发送过来的数据,并进行相应的处理(例如显示)。

[0023] 具体的,所述方案二包括如下过程,

[0024] 过程1: APP向服务器发送UDP数据,调用my_sendto函数(调用_sendto函数会自动自定向到my_sendto函数)发送UDP数据;

[0025] 过程2:my_sendto函数通过其函数内设置的的目的IP地址和端口(替换原来服务器的目的IP地址和端口),与本地代理服务建立通信连接,并将UDP数据发送至本地代理服务;

[0026] 过程3:同时,本地代理服务与目的服务器(APP实际要发送的服务器)建立通信连接;本地代理服务将UDP数据转发至目的服务器;目的服务器接收到UDP数据后进行处理,返回结果数据,并把结果数据发送至本地代理服务;

[0027] 过程5:本地代理服务接收目的服务器发送的结果数据,并将结果数据转发至APP;

[0028] 过程6: APP调用my_receivefrom函数(调用_receivefrom函数会自动自定向到my_receivefrom函数)接收结果数据;my_receivefrom函数(使用本地代理服务的目的IP地址和端口替换服务器的目的IP地址和端口)接收本地代理服务发送过来的数据,并进行相应的处理(例如显示)。

[0029] 本发明揭示了IOS上在App所有调用情况下都能够引导且只引导感兴趣的TCP&UDP流量的方法,从而解决了IOS上使用VPN只能进行全局代理、使用标准的Socks5代理在使用原生Socket的情况下无效的问题。

具体实施方式

[0030] 现结合具体实施方式对本发明进一步说明。

[0031] 作为一个具体的实例,本发明的一种IOS上针对App的TCP/UDP流量引导的方法,可以封装一个基于IOS平台的SDK,供IOS的App调用。该SDK具备如下两大功能:(1)拦截原生Socket的各个函数。(2)将流量TCP/UDP引导到代理服务器。

[0032] 其中,拦截原生Socket的各个函数包括如下步骤:

[0033] 步骤1:建立本地代理服务;具体包括以下内容:函数获取各个Socket函数_connect、_send、_receive等函数的地址,将该函数的地址替换构造代理函数中的函数地址(替换原生Socket函数connect、send、receive等函数为自定义的my_connect、my_send、my_receive函数);然后创建本地代理服务线程;

[0034] 步骤2:构造连接函数集,该连接函数集实现跟原生Socket的各个函数的相同的功能,该连接函数集包括:函数my_connect(功能:创建与指定目的地的连接),my_send(功能:用于向一个已经连接的Socket发送数据),my_receive(功能:用于接收一个已经连接的Socket发送过来的数据),my_sendto(功能:向一指定目的地发送数据,适用于发送未建立连接的UDP数据报),my_receivefrom(功能:接收指定目的地发送过来的数据,适用于接收未建立连接的UDP数据报)等;

[0035] 步骤3:将产生HTTP流量的应用软件记为APP,查找APP在内存当中的镜像头部,找到创建与指定目的地的连接的函数(函数名字为“_connect”,函数功能:创建与指定目的地的连接),将该函数的地址替换步骤2中构造的连接函数集的地址(建立连接时的目标IP和端口)。具体做法是找到函数名字为“_connect”(函数功能:创建与指定目的地的连接)的地址,将此地址替换为自己实现的函数my_connect的地址,并保存旧地址ori_connect;

[0036] 步骤4:循环步骤3,依次替换连接函数集的所有函数的地址(也即步骤2中实现的与各原生Socket各个函数相同功能的函数)的地址,并保存旧地址。

[0037] 将流量TCP/UDP引导到代理服务器,包括将流量TCP引导到代理服务器的步骤以及将流量UDP引导到代理服务器的步骤。

[0038] 1、对于将流量TCP引导到代理服务器的步骤,包括如下内容:

[0039] 步骤1:在my_connect函数中,替换要连接的地址为步骤1中构造的代理函数的地址;

[0040] 步骤2:调用旧地址ori_connect(my_connect替换了连接的地址(IP和端口)之后,会调用原生的ori_connect与代理服务器建立连接),连接成功后,有两种转发数据的代理方案:

[0041] 方案一:同代理HTTPS,向代理服务器发送一个HTTPS连接请求的数据包,包含真实的IP地址和端口(“ConnectIP:PORTHTTP/1.1\r\n\r\n”,这里需要HTTP代理服务器并支持HTTPS代理),然后等待代理服务端返回数据,如果代理服务器与真实服务器连接成功则会接收到连接成功的响应数据包(“200Connectionestablished”),my_connect返回成功(返回值为0),表示连接已经建立,否则返回失败。

[0042] 方案二:socks代理,socks的方案在移动网络下比较差,协商过程太复杂,容易超时。除非使用优化过的socks代理,将协商过程简化。

[0043] TCP数据的引导可以采用方案一和方案二,在程序实现APP时指定其中一种方式,没有特别的情况。UDP数据的引导只能采用方案二socks代理。

[0044] 2、对于将流量UDP引导到代理服务器的步骤,包括如下内容:

[0045] 在发送数据包的功能中(my_send,my_sendto等等),使用Socks5代理即可。UDP代理只能使用Socks5了。

[0046] 在系统运行过程中,IOS可先检测APP的协议是TCP还是UDP,如果是TCP,则选择方案一或者方案二;如果是UDP,则选择方案二将流量引导到代理服务器。

[0047] 在实际应用过程中,所述方案一具体包括如下过程:

[0048] 过程1:APP向服务器发送TCP连接请求,调用my_connect函数(调用_connect函数会自动自定向到my_connect函数);

[0049] 过程2:my_connect函数通过其函数内设置的目的IP地址和端口(替换原来服务器的目的IP地址和端口),与本地代理服务建立通信连接;同时,本地代理服务与目的服务器(APP实际要发送的服务器)建立通信连接;

[0050] 过程3:APP通过my_send函数(调用_send函数会自动自定向到my_send函数)发送TCP数据;my_send函数(使用本地代理服务的目的IP地址和端口替换服务器的目的IP地址和端口),将TCP数据发送到本地代理服务的代理服务器;

[0051] 过程4:本地代理服务解析接收到的TCP数据,将TCP数据转发给APP实际要发送的目的服务器;目的服务器接收到TCP数据后进行处理,返回结果数据,并把结果数据发送至本地代理服务;

[0052] 过程5:本地代理服务接收目的服务器发送的结果数据,并将结果数据转发至APP;

[0053] 过程6:APP调用my_receive函数(调用_receive函数会自动自定向到my_receive函数)接收结果数据;my_receive函数(使用本地代理服务的目的IP地址和端口替换服务器的目的IP地址和端口)接收本地代理服务发送过来的数据,并进行相应的处理(例如显示)。

[0054] 所述方案二包括如下过程,

[0055] 过程1:APP向服务器发送UDP数据,调用my_sendto函数(调用_sendto函数会自动自定向到my_sendto函数)发送UDP数据;

[0056] 过程2:my_sendto函数通过其函数内设置的的目的IP地址和端口(替换原来服务器的目的IP地址和端口),与本地代理服务建立通信连接,并将UDP数据发送至本地代理服务;

[0057] 过程3:同时,本地代理服务与目的服务器(APP实际要发送的服务器)建立通信连接;本地代理服务将UDP数据转发至目的服务器;目的服务器接收到UDP数据后进行处理,返回结果数据,并把结果数据发送至本地代理服务;

[0058] 过程5:本地代理服务接收目的服务器发送的结果数据,并将结果数据转发至APP;

[0059] 过程6:APP调用my_receivefrom函数(调用_receivefrom函数会自动自定向到my_receivefrom函数)接收结果数据;my_receivefrom函数(使用本地代理服务的目的IP地址和端口替换服务器的目的IP地址和端口)接收本地代理服务发送过来的数据,并进行相应的处理(例如显示)。

[0060] 尽管结合优选实施方案具体展示和介绍了本发明,但所属领域的技术人员应该明白,在不脱离所附权利要求书所限定的本实用新型的精神和范围内,在形式上和细节上可以对本发明做出各种变化,均为本发明的保护范围。