

andr流量测试方案1

对于andr2.2之前的系统版本，采用如下命令方式进行获取：

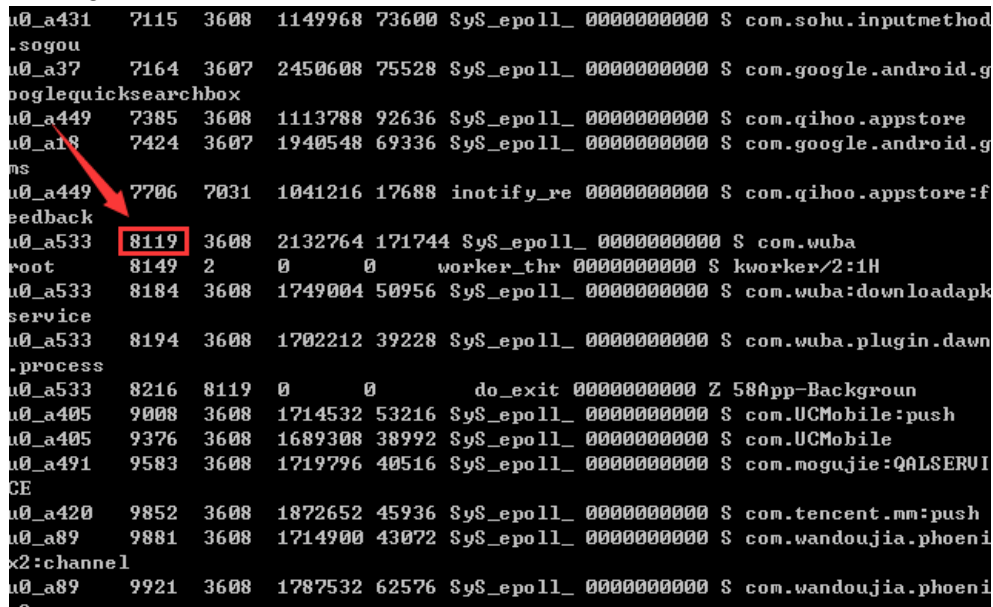
```
aadb devices
adb -s 设备id shell
cd proc
cd uid_stat
ls
cd uid
ls
cat tcp_rcv //查看该应用接收的数据信息
cat tcp_snd //查看该应用发送的数据信息
```

对于andr2.2之后的系统版本，采用如下2种命令方式进行获取：

方案一：

1.先找到指定app的PID

adb shell ps



UID	PID	PPID	PGID	NAME	STATE	FLAGS	WORKING SET	USER
u0_a431	7115	3608	1149968	73600	SyS_epoll_	0000000000	S	com.sohu.inputmethod
u0_a37	7164	3607	2450608	75528	SyS_epoll_	0000000000	S	com.google.android.g
u0_a449	7385	3608	1113788	92636	SyS_epoll_	0000000000	S	com.qihoo.appstore
u0_a18	7424	3607	1940548	69336	SyS_epoll_	0000000000	S	com.google.android.g
u0_a449	7706	7031	1041216	17688	inotify_re	0000000000	S	com.qihoo.appstore:f
u0_a533	8119	3608	2132764	171744	SyS_epoll_	0000000000	S	com.wuba
root	8149	2	0	0	worker_thr	0000000000	S	kworker/2:1H
u0_a533	8184	3608	1749004	50956	SyS_epoll_	0000000000	S	com.wuba:downloadpk
u0_a533	8194	3608	1702212	39228	SyS_epoll_	0000000000	S	com.wuba.plugin.dawn
u0_a533	8216	8119	0	0	do_exit	0000000000	Z	58App-Backgroun
u0_a405	9008	3608	1714532	53216	SyS_epoll_	0000000000	S	com.UCMobile:push
u0_a405	9376	3608	1689308	38992	SyS_epoll_	0000000000	S	com.UCMobile
u0_a491	9583	3608	1719796	40516	SyS_epoll_	0000000000	S	com.mogujie:QALSERVI
u0_a420	9852	3608	1872652	45936	SyS_epoll_	0000000000	S	com.tencent.mm:push
u0_a89	9881	3608	1714900	43072	SyS_epoll_	0000000000	S	com.wandoujia.phoeni
u0_a89	9921	3608	1787532	62576	SyS_epoll_	0000000000	S	com.wandoujia.phoeni

2.adb shell cat /proc/Pid/net/dev

```

C:\Users\yu.li>adb shell cat /proc/21896/net/dev
Inter-| Receive
face |bytes    packets  errs drop fifo frame compressed multicast|bytes    packe
ts errs drop fifo colls carrier compressed
rmnet6: 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
rmnet1: 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
rmnet_usb2: 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
lo: 275825 4698 0 0 0 0 0 0 0 275825 46
98 0 0 0 0 0 0 0 0 0 0
rmnet5: 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
rmnet0: 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
rmnet_usb1: 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
rmnet4: 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
sit0: 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
dummy0: 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
usbnet0: 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
wlan0: 67802592 96291 0 93 0 0 0 0 0 14192911 97
976 0 0 0 0 0 0 0 0 0 0
rmnet_usb0: 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
rmnet3: 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
rmnet7: 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0

```

21896是PID: Transmit

wifi

下载字节

发送

这边的wlan0代表wifi 上传下载量标识! 上传下载量单位是字节可以/1024换算成KB

这里可以看到下载的字节数、数据包 和 发送的字节数、数据包

小技巧: wlan0这些值如何初始化0 很简单 你打开手机飞行模式再关掉就清0了

方案二:

ps:uid的获取可以在对应的PID下面去查看status, 里面会查到uid:

```
adb shell cat /proc/pid/status
```

```

shell@android:/ $ cat /proc/21896/status
cat /proc/21896/status
Name:   li.android.apps
State:  S (sleeping)
Tgid:   21896
Pid:    21896
PPid:   175
TracerPid: 0
Uid:    10000 10000 10000 10000
Gid:    10000 10000 10000 10000
FDSize: 256
Groups: 1006 1015 1028 3002 3003
VmPeak: 804544 kB
VmSize: 769492 kB
VmLck:  0 kB
VmPin:  0 kB
VmHWM:  118908 kB
VmRSS:  106408 kB
VmData: 84452 kB
VmStk:  136 kB
VmExe:  8 kB
VmLib:  35536 kB
VmPTE:  274 kB
VmSwap: 0 kB

```

```
adb shell cat /proc/net/xt_qtaguid/stats | grep 10000
```

```

netstat -n -t -p | grep /proc/net/xt_qtaguid/stats
at /proc/net/xt_qtaguid/stats
dx iface acct_tag_hex uid_tag_int cnt_set rx_bytes rx_packets tx_bytes tx_packets
s rx_tcp_bytes rx_tcp_packets rx_udp_bytes rx_udp_packets rx_other_bytes rx_other_packets
tx_tcp_bytes tx_tcp_packets tx_udp_bytes tx_udp_packets tx_other_bytes tx_other_packets
wlan0 0x0 0 0 2030814 15757 888097 11041 1796759 13186 231535 2541 2520 30 657
67 7881 115466 1795 115364 1365
wlan0 0x0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
wlan0 0x0 1000 0 88610 197 47934 589 88382 194 228 3 0 0 47554 584 380 5 0 0
wlan0 0x0 1000 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
wlan0 0x0 9802 0 106008 728 349300 890 106008 728 0 0 0 349300 890 0 0 0 0
wlan0 0x0 9802 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
wlan0 0x0 10000 0 72474 340 90058 674 72108 335 278 4 88 1 89546 666 512 8 0 0
wlan0 0x0 10000 1 2801007 2747 362118 3829 2801007 2747 0 0 0 362118 3829 0
0 0
0 wlan0 0x0 10001 0 479 3 579 5 479 3 0 0 0 579 5 0 0 0 0
1 wlan0 0x0 10001 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 wlan0 0x0 10005 0 156 3 156 3 156 3 0 0 0 156 3 0 0 0 0

```

其中第6和8列为 rx_bytes（接收数据）和tx_bytes（传输数据）包含tcp, udp等所有网络流量传输的统计。
一个uid可能对应多个 进程，所以这两行流量是累加的就求和就行。

/proc/uid_stat/uid/tcp_send 上传流量

/proc/uid_stat/uid/tcp_rcv 下载流量

如下是代码的实现形式：

```

//获取PID
public static String PID(String PackageName) throws IOException {

    String PID=null;

    Runtime runtime = Runtime.getRuntime();

    Process proc = runtime.exec("adb shell ps |grep "+PackageName);

    try {

        if (proc.waitFor() != 0) {

            System.err.println("exit value = " + proc.exitValue());

        }

        BufferedReader in = new BufferedReader(new InputStreamReader(

            proc.getInputStream()));

        StringBuffer stringBuffer = new StringBuffer();

        String line = null;

        while ((line = in.readLine()) != null) {

            stringBuffer.append(line+" ");

        }

        String str1=stringBuffer.toString();

        String str2=str1.substring(str1.indexOf(" "+PackageName)-46,str1.indexOf(" "+PackageName));

        String str3 =str2.substring(0,7);

        str3 = str3.trim();

        PID=str3;

    } catch (InterruptedException e) {

        System.err.println(e);

    } finally{

        try {

            proc.destroy();

        } catch (Exception e2) {

        }

    }

    return PID;

}

```

//获取下载流量

```
public static double GetFlow(String PackageName) throws IOException {  
    double FlowSize=0;  
    String Pid=PID(PackageName);  
    try{  
        Runtime runtime = Runtime.getRuntime();  
        Process proc = runtime.exec("adb shell cat /proc/"+Pid+"/net/dev");  
        try {  
            if (proc.waitFor() != 0) {  
                System.err.println("exit value = " + proc.exitValue());  
            }  
            BufferedReader in = new BufferedReader(new InputStreamReader(  
                proc.getInputStream()));  
            StringBuffer stringBuffer = new StringBuffer();  
            String line = null;  
            while ((line = in.readLine()) != null) {  
                stringBuffer.append(line+" ");  
            }  
            String str1=stringBuffer.toString();  
            String str2=str1.substring(str1.indexOf("wlan0:"),str1.indexOf("wlan0:")+90);  
            String str4=str2.substring(7,16);  
            str4 = str4.trim();  
            int Flow=Integer.parseInt(str4);  
            FlowSize=Flow/1024;  
        } catch (InterruptedException e) {  
            System.err.println(e);  
        } finally{  
            try {  
                proc.destroy();  
            } catch (Exception e2) {  
            }  
        }  
    }  
    catch (Exception StringIndexOutOfBoundsException)  
    {  
    }  
    }  
    return FlowSize;  
}
```

获取每秒下载流量:

```
public static double Flow(String PackageName) throws IOException, InterruptedException  
{  
    double Flow1=GetFlow(PackageName);  
    Thread.sleep(1000);  
    double Flow=GetFlow(PackageName)-Flow1;  
    //System.out.println(GetFlow()-Flow1);  
    return Flow ;  
}
```

写一下上传的流量获取

