|  |
| --- |
| Sony Ericsson Mobile Communications |
| ANR Example |
|  |

|  |
| --- |
|  |

Contents

[Introduction 3](#_Toc322449386)

[KeyDispatchTimeout1 Keywords: ContentResovler in UI thread 3](#_Toc322449387)

[KeyDispatchTimeout2 Keywords: ContentResolver in AsyncTask onPostExecute, high iowait 4](#_Toc322449388)

[KeyDispatchTimeout3 Keywords: Network stream read in UI thread 5](#_Toc322449389)

[KeyDispatchTimeout4 Keywords: Memory leak/Thread leak 6](#_Toc322449390)

[KeyDispatchTimeout5 Keywords: Call stack normal,CPU starvation 7](#_Toc322449391)

[KeyDispatchTimeout6 Keywords: Deadlock in system process 9](#_Toc322449392)

[KeyDispatchTimeout7 Keywords: Idle state call stack, CPU usages change greatly. 10](#_Toc322449393)

[KeyDispatchTimeout8 Keywords: Call stack runs into native 11](#_Toc322449394)

[KeyDispatchTimeout9 Keywords: Optimize in every way 13](#_Toc322449395)

[BroadcastTimeout1 Keywords: Database operation in broadcast receiver 15](#_Toc322449396)

[BroadcastTimeout2 Keywords: Time consuming work in broadcast receiver 15](#_Toc322449397)

[BroadcastTimeout3 Keywords: OnReceive hasn’t run when timeout 16](#_Toc322449398)

[DMS00926516 Keywords: Blocked in Service onDestroy 17](#_Toc322449399)

[DMS01026315 Keywords: Service causes KeyDispatchTimeOut ANR 18](#_Toc322449400)

[DMS12345678 Keywords: Activity causes BroadcastTimeOut ANR 19](#_Toc322449401)

# Introduction

**This document collects all kind of typical ANR issues and adds simple analysis and remarks contributing to help understanding ANR knowledge and grasping skills on investigating ANR issues.**

**FAT engineers should pay more attention to what will cause ANR issues and apply in work to reduce ANR occurrence rate.**

**Anyway, the most important, understand the nature of ANR.**

**Following issues are from eDream2.1/3.0/4.0. Although 2.1 issues are a little stale, the knowledge involved is not out of date.**

# KeyDispatchTimeout1 Keywords: ContentResovler in UI thread

CPU usage:  
Load: 7.83 / 7.76 / 7.74  
CPU usage from 10806ms to 3166ms ago:  
 **n.conversations: 24%** = 23% user + 0% kernel / faults: 341 minor  
 m.android.phone: 10% = 3% user + 7% kernel / faults: 41 minor  
 system\_server: 7% = 3% user + 3% kernel / faults: 24 minor  
 adbd: 1% = 0% user + 1% kernel  
 app\_process: 1% = 1% user + 0% kernel / faults: 30 minor  
 rild: 0% = 0% user + 0% kernel  
 owstore.android: 0% = 0% user + 0% kernel  
 logcat: 0% = 0% user + 0% kernel  
 d.process.acore: 0% = 0% user + 0% kernel / faults: 1 minor  
 logcat: 0% = 0% user + 0% kernel  
**TOTAL: 100% = 32% user + 14% kernel + 52% iowait + 0% softirq**

DALVIK THREADS:  
""main"" prio=5 tid=3 NATIVE  
| group=""main"" sCount=1 dsCount=0 s=N obj=0x4001b240 self=0xbda8  
| sysTid=1540 nice=0 sched=0/0 cgrp=unknown handle=-1343993184  
at android.os.BinderProxy.transact(Native Method)  
at android.content.ContentProviderProxy.delete(ContentProviderNative.java:507)  
at android.content.**ContentResolver.delete**(ContentResolver.java:709)  
**at com.sonyericsson.conversations.ui.ConversationActivity$27.onClick**(ConversationActivity.java:2767)  
at com.android.internal.app.AlertController$ButtonHandler.handleMessage(AlertController.java:162)  
at android.os.Handler.dispatchMessage(Handler.java:99)  
at android.os.Looper.loop(Looper.java:123)  
at android.app.ActivityThread.main(ActivityThread.java:4370)  
at java.lang.reflect.Method.invokeNative(Native Method)  
at java.lang.reflect.Method.invoke(Method.java:521)  
at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:868)  
at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:626)  
at dalvik.system.NativeStart.main(Native Method)

**This ANR happens when we try to delete messages or entire conversations. The operations involved with database operations are easily to be blocked when IOWait is some high. It should be done in async thread not in the main thread.**

**Fix:   
Move ContentResolver.delete into a separate thread and use a spin progress dialog.**

**Remarks:   
(1) This a typical and most common ANR issue. Placing the example here to stress one idea: ContentResolver is forbidden in main thread.**

# KeyDispatchTimeout2 Keywords: ContentResolver in AsyncTask onPostExecute, high iowait

Process: com.android.email  
Activity: com.android.email/.activity.MessageView  
Subject: keyDispatchingTimedOut  
CPU usage from 2550ms to -2814ms ago:  
5% 187/system\_server: 3.5% user + 1.4% kernel / faults: 86 minor 20 major  
4.4% 1134/com.android.email: 0.7% user + 3.7% kernel / faults: 38 minor 19 major  
4% 372/com.sonyericsson.eventstream: 0.7% user + 3.3% kernel / faults: 6 minor  
1.1% 272/com.android.phone: 0.9% user + 0.1% kernel / faults: 33 minor  
0.9% 252/com.android.systemui: 0.9% user + 0% kernel  
0% 409/com.sonyericsson.eventstream.telephonyplugin: 0% user + 0% kernel / faults: 2 minor  
0.1% 632/com.sonyericsson.devicemonitor: 0.1% user + 0% kernel  
**100% TOTAL: 6.9% user + 8.2% kernel + 84% iowait**  
  
  
----- pid 1134 at 2010-12-17 17:46:51 -----  
Cmd line: com.android.email  
  
DALVIK THREADS:  
(mutexes: tll=0 tsl=0 tscl=0 ghl=0 hwl=0 hwll=0)  
"main" prio=5 tid=1 WAIT  
| group="main" sCount=1 dsCount=0 obj=0x2aaca180 self=0xcf20  
| sysTid=1134 nice=0 sched=0/0 cgrp=[fopen-error:2] handle=1876218976  
at java.lang.Object.wait(Native Method)  
- waiting on <0x2aaca218> (a java.lang.VMThread)  
at java.lang.Thread.parkFor(Thread.java:1424)  
at java.lang.LangAccessImpl.parkFor(LangAccessImpl.java:48)  
at sun.misc.Unsafe.park(Unsafe.java:337)  
at java.util.concurrent.locks.LockSupport.park(LockSupport.java:157)  
at java.util.concurrent.locks.AbstractQueuedSynchronizer.parkAndCheckInterrupt(AbstractQueuedSynchronizer.java:808)  
at java.util.concurrent.locks.AbstractQueuedSynchronizer.acquireQueued(AbstractQueuedSynchronizer.java:841)  
at java.util.concurrent.locks.AbstractQueuedSynchronizer.acquire(AbstractQueuedSynchronizer.java:1171)  
at java.util.concurrent.locks.ReentrantLock$FairSync.lock(ReentrantLock.java:200)  
at java.util.concurrent.locks.ReentrantLock.lock(ReentrantLock.java:261)  
at android.database.sqlite.SQLiteDatabase.lock(SQLiteDatabase.java:378)  
at android.database.sqlite.SQLiteCursor.<init>(SQLiteCursor.java:222)  
at android.database.sqlite.SQLiteDirectCursorDriver.query(SQLiteDirectCursorDriver.java:53)  
at android.database.sqlite.SQLiteDatabase.rawQueryWithFactory(SQLiteDatabase.java:1356)  
at android.database.sqlite.SQLiteDatabase.queryWithFactory(SQLiteDatabase.java:1235)  
at android.database.sqlite.SQLiteDatabase.query(SQLiteDatabase.java:1189)  
at android.database.sqlite.SQLiteDatabase.query(SQLiteDatabase.java:1271)  
at com.android.email.provider.EmailProvider.query(EmailProvider.java:1098)  
at android.content.ContentProvider$Transport.query(ContentProvider.java:187)  
at android.content.**ContentResolver.query**(ContentResolver.java:268)  
at com.android.email.provider.EmailContent$Message.restoreMessageWithId(EmailContent.java:648)  
at com.android.email.Controller.setMessageRead(Controller.java:658)  
at com.android.email.activity.MessageView.onMarkAsRead(MessageView.java:700)  
at com.android.email.activity.MessageView.access$2500(MessageView.java:98)  
at **com.android.email.activity.MessageView$LoadBodyTask.onPostExecute**(MessageView.java:1290)  
at com.android.email.activity.MessageView$LoadBodyTask.onPostExecute(MessageView.java:1255)  
at android.os.AsyncTask.finish(AsyncTask.java:417)  
at android.os.AsyncTask.access$300(AsyncTask.java:127)  
at **android.os.AsyncTask$InternalHandler.handleMessage**(AsyncTask.java:429)  
at android.os.Handler.dispatchMessage(Handler.java:99)  
at android.os.Looper.loop(Looper.java:123)  
at android.app.ActivityThread.main(ActivityThread.java:3652)  
at java.lang.reflect.Method.invokeNative(Native Method)  
at java.lang.reflect.Method.invoke(Method.java:507)  
at com.android.internal.os.ZygoteIn

**IOWait is high which means that many of IO operations are in wait condition. This usually happens when system is too busy on IO. In such condition, database operations will often blocked for some time.**

**Fix:   
Move restoreMessageWithId function call into a new thread.**

**Remarks:  
(1) AsyncTask:** [**http://developer.android.com/reference/android/os/AsyncTask.html**](http://developer.android.com/reference/android/os/AsyncTask.html) **onPostExecute(Result), invoked on the UI thread after the background computation finishes. The result of the background computation is passed to this step as a parameter.  
  
(2) We should be clear about which functions running in the main thread. It’s very important for avoiding ANR.  
  
(3) In project, we often meet with high IOwait issues. There is no very good method to find out why IOWait is high by now. High IOwait** **can be caused by many cases: Yaffs garbage collection/Intensive database operations/ Intensive SDCard access, etc.**

# KeyDispatchTimeout3 Keywords: Network stream read in UI thread

ANR in process: com.sonyericsson.android.mediascape:PhotoViewer (last in com.sonyericsson.android.mediascape:PhotoViewer)  
Annotation: keyDispatchingTimedOut  
CPU usage:  
Load: 6.74 / 6.89 / 6.12  
CPU usage from 8254ms to 3224ms ago:  
ovider.webmedia: 4% = 4% user + 0% kernel / faults: 68 minor  
system\_server: 2% = 1% user + 0% kernel / faults: 18 minor  
re-initialized>: 0% = 0% user + 0% kernel / faults: 50 minor  
events/0: 0% = 0% user + 0% kernel  
**TOTAL: 7% = 6% user + 1% kernel**  
  
DALVIK THREADS:  
""main"" prio=5 tid=3 NATIVE  
| group=""main"" sCount=1 dsCount=0 s=Y obj=0x4001b240 self=0xbda8  
| sysTid=2579 nice=0 sched=0/0 cgrp=unknown handle=-1343993184  
at org.apache.harmony.luni.platform.OSNetworkSystem.receiveStreamImpl(Native Method)  
at org.apache.harmony.luni.platform.**OSNetworkSystem.receiveStream**(OSNetworkSystem.java:478)  
at org.apache.harmony.luni.net.PlainSocketImpl.read(PlainSocketImpl.java:565)  
at org.apache.harmony.luni.net.SocketInputStream.read(SocketInputStream.java:87)  
at org.apache.harmony.luni.internal.net.www.protocol.http.HttpURLConnection$LimitedInputStream.read(HttpURLConnection.java:303)  
at java.io.InputStream.read(InputStream.java:133)  
at java.io.BufferedInputStream.fillbuf(BufferedInputStream.java:157)  
at java.io.BufferedInputStream.read(BufferedInputStream.java:346)  
at android.graphics.BitmapFactory.nativeDecodeStream(Native Method)  
at android.graphics.**BitmapFactory.decodeStream**(BitmapFactory.java:459)  
at com.sonyericsson.android.mediascape.activity.PhotoViewerActivity.**getPreviewImage**(PhotoViewerActivity.java:4465)  
at com.sonyericsson.android.mediascape.activity.PhotoViewerActivity.**dispPreview**(PhotoViewerActivity.java:4406)  
at com.sonyericsson.android.mediascape.activity.PhotoViewerActivity.access$6500(PhotoViewerActivity.java:125)  
at **com.sonyericsson.android.mediascape.activity.PhotoViewerActivity$33$1.run**(PhotoViewerActivity.java:4558)  
at android.os.Handler.handleCallback(Handler.java:587)  
at android.os.Handler.dispatchMessage(Handler.java:92)  
at android.os.Looper.loop(Looper.java:123)  
at android.app.ActivityThread.main(ActivityThread.java:4370)  
at java.lang.reflect.Method.invokeNative(Native Method)  
at java.lang.reflect.Method.invoke(Method.java:521)  
at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:868)  
at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:626)  
at dalvik.system.NativeStart.main(Native Method)

PhotoViewerActivity.java  
private Handler mHandler = new Handler(); // this is in main thread.  
……  
mHandler.post(new Runnable() {   
 public void run() {  
 **dispPreview**(); //this will run in main thread.  
}});

**getPreviewImage**.java  
URL url = new URL(path);   
mPreviewStream = url.openStream();  
previewImage = **BitmapFactory.decodeStream**(mPreviewStream, null, options);

**Fix: Set timeout value for URL connection.   
Timeout for URL connection is necessary from design. But I think putting the job into a separate thread is also a good optimization.  
  
Remarks:   
(1) Handler Knowledge:** [**http://developer.android.com/reference/android/os/Handler.html**](http://developer.android.com/reference/android/os/Handler.html) **(2) public final boolean post (Runnable r)  
Causes the Runnable r to be added to the message queue. The runnable will be run on the thread to which this handler is attached.**

# KeyDispatchTimeout4 Keywords: Memory leak/Thread leak

11-16 21:41:42.560 I/ActivityManager( 1190): ANR in process: android.process.acore (last in android.process.acore)  
11-16 21:41:42.560 I/ActivityManager( 1190): Annotation: keyDispatchingTimedOut  
11-16 21:41:42.560 I/ActivityManager( 1190): CPU usage:  
11-16 21:41:42.560 I/ActivityManager( 1190): Load: 11.5 / 11.1 / 11.09  
11-16 21:41:42.560 I/ActivityManager( 1190): CPU usage from 9046ms to 4018ms ago:  
11-16 21:41:42.560 I/ActivityManager( 1190): **d.process.acore: 98%** = 97% user + 0% kernel / faults: 1134 minor  
11-16 21:41:42.560 I/ActivityManager( 1190): system\_server: 0% = 0% user + 0% kernel / faults: 1 minor  
11-16 21:41:42.560 I/ActivityManager( 1190): adbd: 0% = 0% user + 0% kernel  
11-16 21:41:42.560 I/ActivityManager( 1190): logcat: 0% = 0% user + 0% kernel  
11-16 21:41:42.560 I/ActivityManager( 1190): **TOTAL: 100% = 98% user + 1% kernel**

Cmd line: android.process.acore  
  
DALVIK THREADS:  
"main" prio=5 tid=3 **VMWAIT**  
| group="main" sCount=1 dsCount=0 s=N obj=0x40026240 self=0xbda8  
| sysTid=1815 nice=0 sched=0/0 cgrp=unknown handle=-1344001376  
at dalvik.system.**VMRuntime.trackExternalAllocation**(Native Method)🡺 **Memory shortage makes createBitmap into wait state**at android.graphics.Bitmap.nativeCreate(Native Method)  
at android.graphics.**Bitmap.createBitmap**(Bitmap.java:468)  
at android.view.View.buildDrawingCache(View.java:6324)  
at android.view.View.getDrawingCache(View.java:6178)  
at android.view.ViewGroup.drawChild(ViewGroup.java:1541)  
……  
at com.android.internal.policy.impl.PhoneWindow$DecorView.draw(PhoneWindow.java:1830)  
at android.view.ViewRoot.draw(ViewRoot.java:1349)  
at android.view.ViewRoot.performTraversals(ViewRoot.java:1114)  
at android.view.ViewRoot.handleMessage(ViewRoot.java:1633)  
at android.os.Handler.dispatchMessage(Handler.java:99)  
at android.os.Looper.loop(Looper.java:123)  
at android.app.ActivityThread.main(ActivityThread.java:4370)  
at java.lang.reflect.Method.invokeNative(Native Method)  
at java.lang.reflect.Method.invoke(Method.java:521)  
at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:868)  
at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:626)  
at dalvik.system.NativeStart.main(Native Method)

"Thread-408" prio=5 tid=329 WAIT **// there are over 400 such thread in the log. Thread leak causes memory leak.**| group="main" sCount=1 dsCount=0 s=N obj=0x46910d40 self=0xcd0548  
| sysTid=10602 nice=0 sched=0/0 cgrp=unknown handle=15470792  
 at java.lang.Object.wait(Native Method)  
- waiting on <0x468cd420> (a java.lang.Object)  
at java.lang.Object.wait(Object.java:288)  
at com.sonyericsson.dialer.CallLogContentHelper$UiUpdaterExecutor$1.run(CallLogContentHelper.java:289)  
at java.lang.Thread.run(Thread.java:1096)

Bugreport.txt  
------ PROCRANK ------  
 PID Vss Rss Pss Uss cmdline  
1360 53312K 51812K 38214K **36856K** android.process.acore

\*\* MEMINFO in pid 1360 [android.process.acore] \*\*  
 native dalvik other total  
size: 17036 **23111** N/A 40147  
allocated: 16484 20675 N/A 37159  
free: 296 2436 N/A 2732

**Memory leak will cause ANR. Because when memory is low, normal operations will be blocked for lack of memory or wait GC to collect memory, this will make normal operations taking much more time than before. Low memory will cause system into very bad performance, ANR occurrence rate will increase greatly in this condition. You will see many out of memory issues are along with ANR issues.**

**We should extremely pay attention to resource related functions in callstack, such as “create” “allocate” keywords, that sometime means there is resource leak in the system.**

**Remarks: This is an example, VM heap has 24MB limit. So, 36MB Uss memory is a little too high which means high possibility having memory leak according our experience. Anyway, it’s better to judge with procrank logs which can show the trend of memory usage.**

**Notice: when VM heap has larger size, such as 32MB limit, which makes detecting memory leak harder than before.**

# KeyDispatchTimeout5 Keywords: Call stack normal,CPU starvation

E ActivityManager: ANR in com.android.email (com.android.email/.activity.MessageList)  
E ActivityManager: Reason: keyDispatchingTimedOut  
E ActivityManager: Load: 10.94 / 9.98 / 9.79  
E ActivityManager: CPU usage from 0ms to 7167ms ago:  
E ActivityManager: **93% 368/com.aricent.mtp:** 18% user + 74% kernel / faults: 47 minor  
E ActivityManager: 7.1% 196/system\_server: 5.2% user + 1.8% kernel / faults: 617 minor 7 major  
E ActivityManager: 0.8% 16684/com.android.email: 0.5% user + 0.2% kernel / faults: 60 minor 6 major  
E ActivityManager: 0% 3403/logcat: 0% user + 0% kernel  
E ActivityManager**: 100% TOTAL: 23% user + 76% kernel**  
E ActivityManager: CPU usage from 6624ms to 7141ms later with 99% awake:  
E ActivityManager: **88% 368/com.aricent.mtp**: 19% user + 69% kernel  
E ActivityManager: 5.7% 196/system\_server: 3.8% user + 1.9% kernel / faults: 1 minor  
E ActivityManager: 1.9% 236/InputDispatcher: 1.9% user + 0% kernel  
E ActivityManager: 3.8% 16684/com.android.email: 3.8% user + 0% kernel / faults: 18 minor  
E ActivityManager: 3.8% 16692/Thread-10: 3.8% user + 0% kernel  
E ActivityManager: 1.9% 16699/AsyncTask #1: 1.9% user + 0% kernel  
E ActivityManager: **100% TOTAL: 26% user + 73% Kernel**

----- pid 13854 at 2011-04-16 17:33:08 -----  
Cmd line: com.android.email

DALVIK THREADS:  
(mutexes: tll=0 tsl=0 tscl=0 ghl=0 hwl=0 hwll=0)  
"main" prio=5 tid=1 SUSPENDED  
 | group="main" sCount=1 dsCount=0 obj=0x2aad2248 self=0xcf70  
 | sysTid=13854 nice=0 sched=0/0 cgrp=[fopen-error:2] handle=1876218976  
 at java.lang.String.toString(String.java:~1699)  
 at android.text.BoringLayout.init(BoringLayout.java:171)  
 at android.text.BoringLayout.replaceOrMake(BoringLayout.java:83)  
 at android.widget.TextView.makeNewLayout(TextView.java:5128)  
 at android.widget.TextView.checkForRelayout(TextView.java:5609)  
 at android.widget.TextView.setText(TextView.java:2772)  
 at android.widget.TextView.setText(TextView.java:2640)  
 at android.widget.TextView.setText(TextView.java:2615)  
 at com.android.email.activity.MessageList$MessageListAdapter.bindView(MessageList.java:1726)  
 at android.widget.CursorAdapter.getView(CursorAdapter.java:186)  
 at android.widget.HeaderViewListAdapter.getView(HeaderViewListAdapter.java:220)  
 at android.widget.AbsListView.obtainView(AbsListView.java:1492)  
 at android.widget.ListView.makeAndAddView(ListView.java:1749)  
 at android.widget.ListView.fillSpecific(ListView.java:1294)  
 at android.widget.ListView.layoutChildren(ListView.java:1580)  
 at android.widget.AbsListView.onLayout(AbsListView.java:1343)  
 at android.view.View.layout(View.java:7301)  
 at android.widget.LinearLayout.setChildFrame(LinearLayout.java:1263)  
 at android.widget.LinearLayout.layoutVertical(LinearLayout.java:1137)  
 at android.widget.LinearLayout.onLayout(LinearLayout.java:1051)  
 at android.view.View.layout(View.java:7301)  
 at android.widget.FrameLayout.onLayout(FrameLayout.java:342)  
 at android.view.View.layout(View.java:7301)  
 at android.widget.FrameLayout.onLayout(FrameLayout.java:342)  
 at android.view.View.layout(View.java:7301)  
 at android.view.ViewRoot.performTraversals(ViewRoot.java:1160)  
 at android.view.ViewRoot.handleMessage(ViewRoot.java:1880)  
 at android.os.Handler.dispatchMessage(Handler.java:99)  
 at android.os.Looper.loop(Looper.java:123)  
 at android.app.ActivityThread.main(ActivityThread.java:3701)  
 at java.lang.reflect.Method.invokeNative(Native Method)  
 at java.lang.reflect.Method.invoke(Method.java:507)  
 at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:862)  
 at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:620)  
 at dalvik.system.NativeStart.main(Native Method)

**Through callstack we can’t get valuable info. It seems adding view for a listview which is quite a normal action. We often meet with similar issues which call stack seems normal. In many cases, it’s not caused by the ANR process itself but other outside reason, CPU starvation, low memory, etc.**

**In this case, CPU usage in com.aricent.mtp process is always very high. It’s suspicious. Through investigating, it’s found that an infinite loop will be formed in com.aricent.mtp process in some case. That will cause other processes into CPU starvation condition which means other processes can’t get enough CPU resource. So the normal actions taking little time when CPU is adequate will take much longer time. So, ANR will occur easily.**

**Pick from gerrit**

***“To fix the high cpu usage issue. MTP are calling select() API by specifying the timeout value as 86400 (1 day). The select API is taking reference to timeout variable, and uses the timeout value as count down timer. When the select API returns due to some event or timeout, timeout variable will contain the remaining time to elapse. Since MTP are not resetting the timeout variable inside the loop, the select() API returns immediately and results in infinite loop. The solution is setting the timeout value as NULL to wait indefinitely for an event.”***

# KeyDispatchTimeout6 Keywords: Deadlock in system process

I/ActivityManager( 1222): ANR in process: system (last in system)  
I/ActivityManager( 1222): Annotation: keyDispatchingTimedOut  
I/ActivityManager( 1222): CPU usage:  
I/ActivityManager( 1222): Load: 11.48 / 11.91 / 11.83  
I/ActivityManager( 1222): CPU usage from 10323ms to 31ms ago:  
I/ActivityManager( 1222): system\_server: 1% = 1% user + 0% kernel / faults: 11 minor 11 major  
I/ActivityManager( 1222): yericsson.timer: 1% = 1% user + 0% kernel / faults: 21 minor  
I/ActivityManager( 1222): csson.stopwatch: 1% = 1% user + 0% kernel / faults: 1 minor  
I/ActivityManager( 1222): usb\_mass\_storag: 0% = 0% user + 0% kernel  
I/ActivityManager( 1222): tiwlan\_wq: 0% = 0% user + 0% kernel  
I/ActivityManager( 1222): son.setupwizard: 0% = 0% user + 0% kernel / faults: 4 minor  
I/ActivityManager( 1222): adbd: 0% = 0% user + 0% kernel  
I/ActivityManager( 1222): m.android.phone: 0% = 0% user + 0% kernel  
I/ActivityManager( 1222): app\_process: 0% = 0% user + 0% kernel  
I/ActivityManager( 1222): wpa\_supplicant: 0% = 0% user + 0% kernel / faults: 3 minor  
I/ActivityManager( 1222): ndroid.settings: 0% = 0% user + 0% kernel  
I/ActivityManager( 1222): **TOTAL: 5% = 4% user + 1% kernel 🡺 CPU is low, possibly block caused ANR.**

"**android.server.ServerThread**" prio=5 tid=17 **MONITOR 🡺 Means blocked, we can see through the source code to check**  
| group="main" sCount=1 dsCount=0 s=1 obj=0x458ABC70  
| sysTid=1239 nice=-2 sched=0/0 handle=1291584  
at com.android.server.**NotificationManagerService$1.onPanelRevealed**(NotificationManagerService.java:~290)  
at com.android.server.status.StatusBarService.panelSlightlyVisible(StatusBarService.java:1813)  
at com.android.server.status.StatusBarService.makeExpandedVisible(StatusBarService.java:1088)  
at com.android.server.status.StatusBarService.prepareTracking(StatusBarService.java:1281)  
at com.android.server.status.StatusBarService.animateExpand(StatusBarService.java:1109)  
at com.android.server.status.**StatusBarService$H.handleMessage**(StatusBarService.java:681)  
at android.os.Handler.dispatchMessage(Handler.java:99)  
at android.os.Looper.loop(Looper.java:1511)  
at com.android.server.ServerThread.run(SystemServer.java:487)

"**ActivityManager**" prio=5 tid=19 **MONITOR**  
| group="main" sCount=1 dsCount=0 s=1 obj=0x458B7F78  
| sysTid=1241 nice=-2 sched=0/0 handle=1291824  
at com.android.server.status.**StatusBarService.addPendingOp**(StatusBarService.java:~512)  
at com.android.server.status.StatusBarService.updateIcon(StatusBarService.java:504)  
at com.android.server.**NotificationManagerService.enqueueNotificationWithTag**(NotificationManagerService.java:815)  
at com.android.server.NotificationManagerService.enqueueNotification(NotificationManagerService.java:676)  
at com.android.server.am.ServiceRecord$1.run(ServiceRecord.java:275)  
at android.os.Handler.handleCallback(Handler.java:587)  
at android.os.Handler.dispatchMessage(Handler.java:94)  
at android.os.Looper.loop(Looper.java:1511)  
at com.android.server.am.ActivityManagerService$AThread.run(ActivityManagerService.java:1347)

"**android.server.ServerThread**"

StatusBarService.java  
 private class H extends Handler {  
 public void **handleMessage**(Message m) {  
 **synchronized (mQueue){** animateExpand()

NotificationManagerService.java  
public void **onPanelRevealed**() {  
 **synchronized (mNotificationList)** { ===>blocked

"**ActivityManager**"

NotificationManagerService.java  
public void **enqueueNotificationWithTag**{  
 **synchronized (mNotificationList)** {  
 mStatusBarService.updateIcon(r.statusBarKey, icon, n);

StatusBarService.java  
private void **addPendingOp**(int code, IBinder key, IconData data, NotificationData n, int i) {  
 **synchronized (mQueue)** { ===>blocked

**“android.server.ServerThread” thread occupied mQueue, waiting mNotificationList; “ActivityManager” occupied mNotificationList, waiting mQueue 🡺 DeadLock**

**Possible Fix: (Not applied for it’s a not perfect solution and it’s hard for user to reproduce, and the product is in end phase)**

**Remarks:   
(1) System process’s looper is not in main thread but in “android.server.ServerThread”.  
(2) This is an Google known issue.** [**http://code.google.com/p/android/issues/detail?id=7986#makechanges**](http://code.google.com/p/android/issues/detail?id=7986#makechanges) **(3) This issue is sometimes reported as watchdog issue if watchdog detected it earlier than ANR mechanism.**

**(4) This is an Éclair2.1 issue. In Gingerbread, blocked issue will be more easy to check. The trace will show out which object it’s waiting and which thread is holding the object.**

"Applications Loader" prio=7 tid=13 MONITOR  
| group="main" sCount=1 dsCount=0 obj=0x2b157d28 self=0xaec5f8  
| sysTid=798 nice=10 sched=0/0 cgrp=[fopenerror:2] handle=11454256  
at com.fede.launcher.LauncherModel$ApplicationsLoader.run(LauncherModel.java:~622)  
**-- waiting to lock <0x2affb868> (a com.fede.launcher.LauncherModel) held by threadid=1 (main)**  
at java.lang.Thread.run(Thread.java:1019)

# KeyDispatchTimeout7 Keywords: Idle state call stack, CPU usages change greatly.

04-01 **13:12:11.572** I/InputDispatcher(  220): Application is not responding: Window{2b263310 com.android.email/com.android.email.activity.SplitScreenActivity paused=false}.  5009.8ms since event, 5009.5ms since wait started  
04-01 13:12:11.572 I/WindowManager( 220): Input event dispatching timed out sending to com.android.email/com.android.email.activity.SplitScreenActivity   
04-01 **13:12:14.123** I/Process (  220): Sending signal. PID: 21404 SIG: 3   
04-01 13:12:14.123 I/dalvikvm(21404): threadid=4: reacting to signal 3   
……  
04-01 13:12:15.872 E/ActivityManager(  220): ANR in com.android.email (com.android.email/.activity.SplitScreenActivity)  
04-01 13:12:15.872 E/ActivityManager(  220): Reason: keyDispatchingTimedOut   
04-01 13:12:15.872 E/ActivityManager(  220): Load: 8.68 / 8.37 / 8.53  
04-01 13:12:15.872 E/ActivityManager(  220): CPU usage from 4361ms to 699ms ago:  
04-01 13:12:15.872 E/ActivityManager(  220):   5.5% 21404/com.android.email: 1.3% user + 4.1% kernel / faults: 10 minor  
04-01 13:12:15.872 E/ActivityManager(  220):   4.3% 220/system\_server: 2.7% user + 1.5% kernel / faults: 11 minor 2 major  
04-01 13:12:15.872 E/ActivityManager(  220):   0.9% 52/spi\_qsd.0: 0% user + 0.9% kernel  
04-01 13:12:15.872 E/ActivityManager(  220):   0.5% 65/irq/170-cyttsp-: 0% user + 0.5% kernel  
04-01 13:12:15.872 E/ActivityManager(  220):   0.5% 296/com.android.systemui: 0.5% user + 0% kernel  
04-01 13:12:15.872 E/ActivityManager(  220): **100% TOTAL: 4.8% user + 7.6% kernel + 87% iowait**04-01 13:12:15.872 E/ActivityManager(  220): **CPU usage** **from 3697ms to 4223ms later**:  
04-01 13:12:15.872 E/ActivityManager(  220):   25% 21404/com.android.email: 25% user + 0% kernel / faults: 191 minor  
04-01 13:12:15.872 E/ActivityManager(  220):     16% 21603/\_\_eas(par.hakan: 16% user + 0% kernel  
04-01 13:12:15.872 E/ActivityManager(  220):     7.2% 21406/GC: 7.2% user + 0% kernel  
04-01 13:12:15.872 E/ActivityManager(  220):     1.8% 21409/Compiler: 1.8% user + 0% kernel  
04-01 13:12:15.872 E/ActivityManager(  220):   5.5% 220/system\_server: 0% user + 5.5% kernel / faults: 1 minor  
04-01 13:12:15.872 E/ActivityManager(  220):     5.5% 263/InputDispatcher: 0% user + 5.5% kernel  
04-01 13:12:15.872 E/ActivityManager(  220): **32% TOTAL: 28% user + 3.7% kernel**

----- pid 21404 at 2011-04-01 **13:12:14** -----  
Cmd line: com.android.email

DALVIK THREADS:  
(mutexes: tll=0 tsl=0 tscl=0 ghl=0 hwl=0 hwll=0)  
"main" prio=5 tid=1 NATIVE  
 | group="main" sCount=1 dsCount=0 obj=0x2aad2248 self=0xcf70  
 | sysTid=21404 nice=0 sched=0/0 cgrp=[fopen-error:2] handle=1876218976  
 **at android.os.MessageQueue.nativePollOnce(Native Method)  
 at android.os.MessageQueue.next(MessageQueue.java:119)  
 at android.os.Looper.loop(Looper.java:110)**  
 at android.app.ActivityThread.main(ActivityThread.java:3688)  
 at java.lang.reflect.Method.invokeNative(Native Method)  
 at java.lang.reflect.Method.invoke(Method.java:507)  
 at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:866)  
 at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:624)  
 at dalvik.system.NativeStart.main(Native Method)

**Main thread call stack is in an idle state waiting next message coming. Why ANR ?**

**(1) It was about 3 seconds after ANR occurred that signal is sent to ANR process to get trace.   
(2) “CPU usage from 3697ms to 4223ms later” has zero iowait. That means the iowait has disappeared when 3-4s after ANR occurred, that is, the blocked thread has most likely recovered to normal just after ANR (1-2s,e.g.) So, it was already late when the system got the trace (3s after ANR). The trace is invalid for this case.**

**For such case, the trace has no meaning for solving the issue. We can reproduce to get a good log and then continue to investigate**.

# KeyDispatchTimeout8 Keywords: Call stack runs into native

ActivityManager: ANR in process: com.sonyericsson.mediaplayer (last in com.sonyericsson.mediaplayer)  
ActivityManager: Annotation: keyDispatchingTimedOut  
ActivityManager: CPU usage:  
ActivityManager: Load: 7.09 / 7.53 / 7.35  
ActivityManager: CPU usage from 9100ms to 4067ms ago:  
ActivityManager: android.browser: 14% = 14% user + 0% kernel / faults: 183 minor  
ActivityManager: system\_server: 2% = 1% user + 0% kernel / faults: 38 minor  
ActivityManager: roid.mediascape: 1% = 0% user + 1% kernel / faults: 1 minor  
ActivityManager: adbd: 0% = 0% user + 0% kernel  
ActivityManager: wpa\_supplicant: 0% = 0% user + 0% kernel / faults: 18 minor  
ActivityManager: logcat: 0% = 0% user + 0% kernel / faults: 24 minor  
ActivityManager: **TOTAL: 19% = 15% user + 3% kernel 🡺 CPU is low, ANR is possibly caused by block.**

----- pid 31071 at 2011-02-06 15:42:31 -----  
Cmd line: com.sonyericsson.mediaplayer

DALVIK THREADS:  
"main" prio=5 tid=3 **NATIVE 🡺 Runs into native code, can’t get detailed info from trace, crashdump is needed to investigate further.**   
 | group="main" sCount=1 dsCount=0 s=N obj=0x40023240 self=0xbda8  
 | sysTid=31071 nice=0 sched=0/0 cgrp=unknown handle=-1343993184  
 at android.media.**MediaPlayer.\_reset**(Native Method)  
 at android.media.**MediaPlayer.reset**(MediaPlayer.java:1219)  
 at com.sonyericsson.dlnamedia.impl.stub.MediaPlayerExImpl.reset(PlayerStubFactoryImpl.java:275)  
 at com.sonyericsson.dlnamedia.impl.AudioVideoPlayerImpl.doDeallocate(AudioVideoPlayerImpl.java:680)  
 at com.sonyericsson.dlnamedia.impl.PlayerBase$PlayerStatePrefetched.close(PlayerBase.java:839)  
 at com.sonyericsson.dlnamedia.impl.PlayerBase.close(PlayerBase.java:167)  
 at com.sonyericsson.mediaplayer.wrapper.MediaPlayerWrapper.release(MediaPlayerWrapper.java:370)  
 at com.sonyericsson.mediaplayer.model.VideoControlAdapter.release(VideoControlAdapter.java:314)  
 at com.sonyericsson.mediaplayer.state.StateBase.release(StateBase.java:183)  
 at com.sonyericsson.mediaplayer.state.PlayerStateManager.stop(PlayerStateManager.java:1073)  
 at com.sonyericsson.mediaplayer.service.**VideoLocalAdapter.stop**(VideoLocalAdapter.java:509)  
 at com.sonyericsson.mediaplayer.activity.VideoPlayerActivityBase.onStop(VideoPlayerActivityBase.java:1154)  
 at com.sonyericsson.mediaplayer.activity.**VideoPlayerActivity.onStop**(VideoPlayerActivity.java:321)  
 at android.app.Instrumentation.callActivityOnStop(Instrumentation.java:1169)  
 at android.app.Activity.performStop(Activity.java:3802)  
 at android.app.ActivityThread.performStopActivityInner(ActivityThread.java:3231)  
 at android.app.ActivityThread.handleStopActivity(ActivityThread.java:3279)  
 at android.app.ActivityThread.access$2500(ActivityThread.java:123)  
 at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1887)  
 at android.os.Handler.dispatchMessage(Handler.java:99)  
 at android.os.Looper.loop(Looper.java:123)  
 at android.app.ActivityThread.main(ActivityThread.java:4370)  
 at java.lang.reflect.Method.invokeNative(Native Method)  
 at java.lang.reflect.Method.invoke(Method.java:521)  
 at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:868)  
 at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:626)  
 at dalvik.system.NativeStart.main(Native Method)

**See source code:**

1144 public void onStop() {  
 1150 if ((!mOnPauseBlock) && mPlayerStarted ){  
 1151 if(mPlayerAdapter != null) {  
 1152 // stop  
 1153 mPlayerAdapter.removeListener(mPlayerListener);  
 1154 **mPlayerAdapter.stop();** **🡺 Nothing special, finally invokes MediaPlayer.reset**  
 1185 }

**From the native call stack, we can know it’s doing binder communication. It’s waiting reply from remote process through Binder driver.**

**Native call stack:**

#0  \_\_ioctl () at bionic/libc/arch-arm/syscalls/\_\_ioctl.S:15   
#1  0xafe204c0 in ioctl (fd=<value optimized out>, request=-1091413932) at bionic/libc/bionic/ioctl.c:41  
#2  0xa9d985a6 in **android::IPCThreadState::talkWithDriver** (this=0x11b688, doReceive=<value optimized out>) at frameworks/base/libs/binder/IPCThreadState.cpp:749   
#3  0xa9d98a5c in **android::IPCThreadState::waitForResponse** (this=0x11b688, reply=0xbef258e0, acquireResult=<value optimized out>)    at frameworks/base/libs/binder/IPCThreadState.cpp:614   
#4  0xa9d990b4 in **android::IPCThreadState::transact** (this=0x11b688, handle=<value optimized out>, code=<value optimized out>, data=<value optimized out>, reply=0xbef258e0, flags=16) at frameworks/base/libs/binder/IPCThreadState.cpp:510  
#5  0xa9d95bda in **android::BpBinder::transact** (this=0x41be68, code=11, data=..., reply=0xbef258e0, flags=0) at frameworks/base/libs/binder/BpBinder.cpp:165  
#6  0xab22e4ce in **android::BpMediaPlayer::reset** (this=<value optimized out>) at frameworks/base/media/libmedia/IMediaPlayer.cpp:145   
#7  0xab22b4aa **in android::MediaPlayer::reset** (this=0x4a5fe0) at frameworks/base/media/libmedia/mediaplayer.cpp:478  
#8  0xab304458 in **android\_media\_MediaPlayer\_reset** (env=0xab48, thiz=0x45d52ac0) at frameworks/base/media/jni/android\_media\_MediaPlayer.cpp:389  
#9  0xad00edb8 in dvmPlatformInvoke () at dalvik/vm/arch/arm/CallEABI.S:243  
#10 0xad03893a in dvmCallJNIMethod\_virtualNoRef (args=0x41873b9c, pResult=0xbef25a30, method=0x4197aeb4, self=0xbda8) at dalvik/vm/Jni.c:1603  
#11 0xad013abc in dalvik\_mterp () at dalvik/vm/mterp/out/InterpAsm-armv7-a.S:9381  
#12 0xad01943c in dvmMterpStd (self=<value optimized out>, glue=0xbef25a28) at dalvik/vm/mterp/Mterp.c:107  
#13 0xad018908 in dvmInterpret (self=0xbda8, method=<value optimized out>, pResult=0xbef25ab0) at dalvik/vm/interp/Interp.c:987  
#14 0xad04f134 in dvmInvokeMethod (obj=<value optimized out>, method=0x4190f30c, argList=<value optimized out>, params=<value optimized out>,  returnType=0x4001a400, noAccessCheck=<value optimized out>) at dalvik/vm/interp/Stack.c:750  
#15 0xad0573f0 in Dalvik\_java\_lang\_reflect\_Method\_invokeNative (args=<value optimized out>, pResult=<value optimized out>) at dalvik/vm/native/java\_lang\_reflect\_Method.c:101  
#16 0xad013abc in dalvik\_mterp () at dalvik/vm/mterp/out/InterpAsm-armv7-a.S:9381  
#17 0xad01943c in dvmMterpStd (self=<value optimized out>, glue=0xbef25b48) at dalvik/vm/mterp/Mterp.c:107  
#18 0xad018908 in dvmInterpret (self=0xbda8, method=<value optimized out>, pResult=0xbef25bd8) at dalvik/vm/interp/Interp.c:987  
#19 0xad04f45e in dvmCallMethodV (self=0xbda8, method=0x418dd908, obj=<value optimized out>, fromJni=<value optimized out>, pResult=0xbef25bd8, args=...) at dalvik/vm/interp/Stack.c:534  
#20 0xad03b0c2 in CallStaticVoidMethodV (env=<value optimized out>, jclazz=<value optimized out>, methodID=0x418dd908, args=...) at dalvik/vm/Jni.c:2681  
#21 0xad328b6a in \_JNIEnv::CallStaticVoidMethod (this=<value optimized out>, clazz=<value optimized out>, methodID=0xaac0  at dalvik/libnativehelper/include/nativehelper/jni.h:778  
#22 0xad3298d8 in android::AndroidRuntime::start (this=<value optimized out>, className=<value optimized out>, startSystemServer=<value optimized out>)  at frameworks/base/core/jni/AndroidRuntime.cpp:948

**The remote process which performs real “reset” action is blocked on semaphore\_wait when** **Send\_command.   
So com.sonyericsson.mediaplayer main thread is blocked, ANR happens. The next step is to check why send\_command is blocked.**

**Pid:13284 “PV player” thread, user space call stack:**

#0 \_\_futex\_wait () at bionic/libc/arch-arm/bionic/atomics\_arm.S:168  
#1 0xafe2172a in **sem\_wait** (sem=0x1880fc) at bionic/libc/bionic/semaphore.c:117  
#2 0x82110ffc **in omx\_vdec\_linux::semaphore\_wait** (this=<value optimized out>) at vendor/qcom-opensource/omx/mm-video/8k/vdec/src/omx\_vdec\_linux.cpp:162  
#3 0x821089ea in **omx\_vdec::send\_command** (this=0x1868e8, hComp=<value optimized out>, cmd=<value optimized out>, param1=<value optimized out>, cmdData=0x0) at vendor/qcom-opensource/omx/mm-video/8k/vdec/src/omx\_vdec.cpp:1336  
#4 0x82000e70 **in qc\_omx\_component\_send\_command** (hComp=0x1868ec, cmd=OMX\_CommandStateSet, param1=2, cmdData=0x0) at vendor/qcom-opensource/omx/mm-core/omxcore/src/common/omx\_core\_cmp.cpp:135  
#5 0xa60d2196 in **PVMFOMXBaseDecNode::DoReset** (this=0x16ed00, aCmd=<value optimized out>) at external/opencore/nodes/pvomxbasedecnode/src/pvmf\_omx\_basedec\_node.cpp:5011  
#6 0xa60d053a in PVMFOMXBaseDecNode::ProcessCommand (this=0x16ed00, aCmd=...) at external/opencore/nodes/pvomxbasedecnode/src/pvmf\_omx\_basedec\_node.cpp:6278  
#7 0xa60d3900 in PVMFOMXBaseDecNode::Run (this=0x16ed00) at external/opencore/nodes/pvomxbasedecnode/src/pvmf\_omx\_basedec\_node.cpp:507  
#8 0xa6058102 in OsclExecSchedulerCommonBase::CallRunExec (this=<value optimized out>, pvactive=<value optimized out>) at external/opencore/oscl/oscl/osclproc/src/oscl\_scheduler.cpp:1242  
#9 0xa605837c in OsclExecSchedulerCommonBase::BlockingLoopL (this=0x102ab0) at external/opencore/oscl/oscl/osclproc/src/oscl\_scheduler.cpp:1301  
#10 0xa6058734 in OsclExecSchedulerCommonBase::StartScheduler (this=<value optimized out>, aSignal=<value optimized out>) at external/opencore/oscl/oscl/osclproc/src/oscl\_scheduler.cpp:804  
#11 0xa70dcf82 in PlayerDriver::playerThread (this=<value optimized out>) at external/opencore/android/playerdriver.cpp:1103  
#12 0xa70dd09a in PlayerDriver::startPlayerThread (cookie=<value optimized out>) at external/opencore/android/playerdriver.cpp:1043  
#13 0xa9d1c3ce in thread\_data\_t::trampoline (t=<value optimized out>) at frameworks/base/libs/utils/Threads.cpp:101  
#14 0xafe0fd94 in \_\_thread\_entry (func=0xa9d1c365 <thread\_data\_t::trampoline(thread\_data\_t const\*)>, arg=0xc2470, tls=<value optimized out>) at bionic/libc/bionic/pthread.c:188  
#15 0xafe0f864 in pthread\_create (thread\_out=<value optimized out>, attr=0x78, start\_routine=0xa9d1c365 <thread\_data\_t::trampoline(thread\_data\_t const\*)>, arg=0xc2470) at bionic/libc/bionic/pthread.c:324

**You should be familiar with Binder and MediaPlayer architecture knowledge to find the corresponding thread.**

**Remarks: If traces state is native and we can’t get enough useful info from the traces, then crash dump is needed to see into the native trace. Set crash level to 2 and retest to get it. But be attention that when crash level is set to 2, java uncaught exceptions will also trigger crash dumps. So, it’s not so easy to get the correct crash dump actually especially when it’s not easy to reproduce.**

# KeyDispatchTimeout9 Keywords: Optimize in every way

E/ActivityManager( 211): ANR in com.sonyericsson.android.socialphonebook (com.sonyericsson.android.socialphonebook/.ContactPickMainActivity)  
E/ActivityManager( 211): Reason: keyDispatchingTimedOut  
E/ActivityManager( 211): Load: 8.47 / 3.45 / 1.28  
E/ActivityManager( 211): CPU usage from 4860ms to -935ms ago:  
E/ActivityManager( 211**): 84% 1042/android.process.acore**: 84% user + 0% kernel / faults: 23 minor **🡺 Contacts provider runs in this process, high CPU usually means it’s doing busy work.**E/ActivityManager( 211): 6.5% 211/system\_server: 4.3% user + 2.2% kernel / faults: 458 minor 22 major  
E/ActivityManager( 211): 4.8% 1033/com.sonyericsson.android.socialphonebook: 4.4% user + 0.3% kernel / faults: 76 minor 8 major  
E/ActivityManager( 211): 3.4% 58/spi\_qsd.0: 0% user + 3.4% kernel  
E/ActivityManager( 211): 0.1% 4/events/0: 0% user + 0.1% kernel  
E/ActivityManager( 211): 0% 276/com.android.systemui: 0% user + 0% kernel / faults: 23 minor  
E/ActivityManager( 211): 0% 886/com.sonyericsson.widget.digitalclock: 0% user + 0% kernel / faults: 45 minor  
E/ActivityManager( 211): 0% 913/com.sonyericsson.digitalclockwidget: 0% user + 0% kernel / faults: 6 minor  
E/ActivityManager( 211): **100% TOTAL: 93% user + 6% kernel**E/ActivityManager( 211): CPU usage from 379ms to 904ms later:  
E/ActivityManager( 211): 90% 1042/android.process.acore: 90% user + 0% kernel  
E/ActivityManager( 211): 90% 1052/Binder Thread #: 90% user + 0% kernel  
E/ActivityManager( 211): 3.7% 58/spi\_qsd.0: 0% user + 3.7% kernel  
E/ActivityManager( 211): 3.8% 211/system\_server: 1.9% user + 1.9% kernel / faults: 13 minor  
E/ActivityManager( 211): 3.8% 250/InputDispatcher: 0% user + 3.8% kernel  
E/ActivityManager( 211): 1.1% 4/events/0: 0% user + 1.1% kernel  
E/ActivityManager( 211): 100% TOTAL: 90% user + 9.4% kernel

Cmd line: com.sonyericsson.android.socialphonebook

DALVIK THREADS:  
(mutexes: tll=0 tsl=0 tscl=0 ghl=0 hwl=0 hwll=0)  
"main" prio=5 tid=1 NATIVE  
| group="main" sCount=1 dsCount=0 obj=0x2aad2248 self=0xcf70  
| sysTid=1593 nice=0 sched=0/0 cgrp=[fopen-error:2] handle=1876218976  
at android.os.BinderProxy.transact(Native Method)  
at android.database.BulkCursorProxy.getWindow(BulkCursorNative.java:215)  
at android.database.BulkCursorToCursorAdaptor.onMove(BulkCursorToCursorAdaptor.java:104)  
at android.database.AbstractCursor.moveToPosition(AbstractCursor.java:188)  
at android.database.CursorWrapper.moveToPosition(CursorWrapper.java:187)  
at com.sonyericsson.android.socialphonebook.cursor.SortedCursor.onMove(SortedCursor.java:59)  
at com.sonyericsson.android.socialphonebook.cursor.ContactsCursorSorter$ContactsCursor.onMove(ContactsCursorSorter.java:52)  
at android.database.AbstractCursor.moveToPosition(AbstractCursor.java:188)  
at com.sonyericsson.android.socialphonebook.cursor.CursorWrapper.onMove(CursorWrapper.java:58)  
at android.database.AbstractCursor.moveToPosition(AbstractCursor.java:188)  
at android.database.AbstractCursor.moveToNext(AbstractCursor.java:256)  
at com.sonyericsson.android.socialphonebook.cursor.**ConversationsCursorWrapper**.<init>(ConversationsCursorWrapper.java:49)  
at **com.sonyericsson.android.socialphonebook.ContactPickActivity$ContactItemListAdapter.changeCursor**(ContactPickActivity.java:2810)  
at **com.sonyericsson.android.socialphonebook.cursor.ContactsListFilter.publishResults**(ContactsListFilter.java:176)  
at android.widget.Filter$ResultsHandler.handleMessage(Filter.java:282)  
at android.os.Handler.dispatchMessage(Handler.java:99)  
at android.os.Looper.loop(Looper.java:123)  
at android.app.ActivityThread.main(ActivityThread.java:3701)  
at java.lang.reflect.Method.invokeNative(Native Method)  
at java.lang.reflect.Method.invoke(Method.java:507)  
at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:862)  
at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:620)  
at dalvik.system.NativeStart.main(Native Method)

**This issue occurred when selecting the recipient from contact in message. It can be easily reproduced when user has a large number of contacts with a couple of phone numbers for each. Through test, we found some operations are very heavy in changeCursor method. But move the heavy work to** **a none-UI thread will cause new errors for concurrency operations on cursor.**

**Followings are what we have done to solve the issue.  
(1) Move the heavy work into a none-UI thread.  
(2) Add lock and adjust logic to avoid concurrency errors.   
(3) Get rid of one unnecessary filter.  
(4) Modify search logic(cancel last search operation when search text changes) which will greatly improve the efficiency.**

**Remarks: In many cases, ANRs are performance issues. Moving some sections to a none-UI thread is the simplest but not best method in some cases. And this method is sometimes not available or comfortable because this will bring in extra UI or logic control job.**

**Application developers should make strong and efficient applications but not only implementing functions. Extreme/Stress conditions should be considered and optimized. Application developers should form the consciousness. Optimization is always encouraged if possible.**

# BroadcastTimeout1 Keywords: Database operation in broadcast receiver

Subject: Broadcast of Intent { act=com.android.mms.transaction.MessageStatusReceiver.MESSAGE\_STATUS\_RECEIVED dat=content://sms/5964 **cmp=com.sonyericsson.conversations/com.android.mms.transaction.MessageStatusReceiver** (has extras) }  
  
CPU usage from 0ms to 5040ms later with 99% awake:  
13% 1635/system\_server: 9.1% user + 4.3% kernel / faults: 589 minor 15 major  
4.7% 1731/com.android.phone: 1.1% user + 3.5% kernel / faults: 28 minor 3 major  
2.5% 23066/com.sonyericsson.eventstream: 0.5% user + 1.9% kernel / faults: 64 minor  
0.3% 22798/android.process.acore: 0.3% user + 0% kernel / faults: 39 minor  
0% 108/rild: 0% user + 0% kernel  
0% 1708/com.android.systemui: 0% user + 0% kernel / faults: 33 minor  
0% 2106/com.sonyericsson.digitalclockwidget: 0% user + 0% kernel / faults: 2 minor  
0.1% 22913/com.sonyericsson.eventstream.telephonyplugin: 0.1% user + 0% kernel / faults: 11 minor  
0.1% 23315/com.sonyericsson.conversations: 0% user + 0.1% kernel / faults: 18 minor 6 major  
**100% TOTAL: 12% user + 10% kernel + 76% iowait**  
  
  
----- pid 23315 at 2010-11-23 16:25:09 -----  
Cmd line: com.sonyericsson.conversations  
  
DALVIK THREADS:  
(mutexes: tll=0 tsl=0 tscl=0 ghl=0 hwl=0 hwll=0)  
""main"" prio=5 tid=1 NATIVE  
| group=""main"" sCount=1 dsCount=0 obj=0x2aaca180 self=0xcf28  
| sysTid=23315 nice=0 sched=3/0 cgrp=[fopen-error:2] handle=1876218976  
at android.os.BinderProxy.transact(Native Method)  
at android.content.ContentProviderProxy.update(ContentProviderNative.java:493)  
at android.content.**ContentResolver.update**(ContentResolver.java:724)  
at android.database.sqlite.SqliteWrapper.update(SqliteWrapper.java:79)  
at com.android.mms.transaction.MessageStatusReceiver.updateMessageStatus(MessageStatusReceiver.java:99)  
at **com.android.mms.transaction.MessageStatusReceiver.onReceive**(MessageStatusReceiver.java:63)  
at android.app.ActivityThread.handleReceiver(ActivityThread.java:1776)  
at android.app.ActivityThread.access$2400(ActivityThread.java:121)  
at android.app.ActivityThread$H.handleMessage(ActivityThread.java:985)  
at android.os.Handler.dispatchMessage(Handler.java:99)  
at android.os.Looper.loop(Looper.java:123)  
at android.app.ActivityThread.main(ActivityThread.java:3654)  
at java.lang.reflect.Method.invokeNative(Native Method)  
at java.lang.reflect.Method.invoke(Method.java:507)  
at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:839)  
at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:597)  
at dalvik.system.NativeStart.main(Native Method)

**IOWait is high, so updateMessageStatus is blocked, that caused broadcast timeout.**

**Fix:   
Move operations into a service.**

# BroadcastTimeout2 Keywords: Time consuming work in broadcast receiver

ANR in process: com.sonyericsson.music  
Annotation: Broadcast of Intent { act=android.intent.action.BOOT\_COMPLETED cmp=com.sonyericsson.music/.MusicReceiver }  
CPU usage:  
Load: 10.29 / 2.92 / 1.0  
CPU usage from 7029ms to 74ms ago:  
**debuggerd: 58%** = 5% user + 52% kernel / faults: 104 minor  
ndroid.crashsms: 24% = 17% user + 6% kernel / faults: 11727 minor 1435 major  
**yericsson.music: 10% = 4% user + 6% kernel / faults: 9447 minor 3069 major**  
d.process.acore: 9% = 9% user + 0% kernel / faults: 1532 minor 34 major  
apps.components: 3% = 2% user + 1% kernel / faults: 3369 minor 261 major  
system\_server: 2% = 1% user + 1% kernel / faults: 648 minor 74 major  
sson.homescreen: 0% = 0% user + 0% kernel / faults: 23 minor  
m.android.phone: 0% = 0% user + 0% kernel / faults: 529 minor  
kswapd0: 0% = 0% user + 0% kernel  
rild: 0% = 0% user + 0% kernel / faults: 171 minor 7 major  
iddd: 0% = 0% user + 0% kernel / faults: 36 minor 1 major  
droid.textinput: 0% = 0% user + 0% kernel / faults: 149 minor  
son.semccheckin: 0% = 0% user + 0% kernel / faults: 27 minor  
on.android.fota: 0% = 0% user + 0% kernel / faults: 310 minor  
**TOTAL: 100% = 30% user + 62% kernel + 7% iowait**  
  
DALVIK THREADS:  
"main" prio=5 tid=3 **RUNNABLE 🡺 The thread is still running but time out finally.**  
| group="main" sCount=1 dsCount=0 s=Y obj=0x4001e240 self=0xbdb8  
| sysTid=1832 nice=0 sched=0/0 cgrp=unknown handle=-1344001376  
at java.lang.String.indexOf(String.java:~1125)  
at android.app.ApplicationContext.makeFilename(ApplicationContext.java:1441)  
at android.app.ApplicationContext.getSharedPrefsFile(ApplicationContext.java:309)  
at android.app.ApplicationContext.getSharedPreferences(ApplicationContext.java:315)  
at android.content.ContextWrapper.getSharedPreferences(ContextWrapper.java:146)  
at android.content.ContextWrapper.getSharedPreferences(ContextWrapper.java:146)  
at com.sonyericsson.music.artistbook.PageManager.writePagesToPrefs(PageManager.java:556)  
at com.sonyericsson.music.artistbook.PageManager.**searchForPlugins**(PageManager.java:200)  
**at com.sonyericsson.music.MusicReceiver.onReceive(MusicReceiver.java:65)**  
at android.app.ActivityThread.handleReceiver(ActivityThread.java:2644)  
at android.app.ActivityThread.access$3100(ActivityThread.java:123)  
at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1920)  
at android.os.Handler.dispatchMessage(Handler.java:99)  
at android.os.Looper.loop(Looper.java:123)  
at android.app.ActivityThread.main(ActivityThread.java:4370)  
at java.lang.reflect.Method.invokeNative(Native Method)  
at java.lang.reflect.Method.invoke(Method.java:521)  
at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:868)  
at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:626)  
at dalvik.system.NativeStart.main(Native Method)

**SearchForPlugins is a little time consuming work. When CPU resource it got is limited, it can’t finish in time.**

**Fix:   
Move operations into a service.**

**Remarks: In many cases, we nearly do nothing but start a service to handle the work in broadcast receiver, except that the work is light-weighted and you’re definitely sure it won’t be blocked. We should consider bad system performance chance.**

# BroadcastTimeout3 Keywords: OnReceive hasn’t run when timeout

W/ActivityManager( 216): Timeout of broadcast BroadcastRecord{2b2bdc90 android.net.sip.SIP\_SERVICE\_UP} - receiver=android.os.BinderProxy@2b2b8890, started 10006ms ago  
W/ActivityManager( 216): Receiver during timeout: ResolveInfo{2b1c2558 com.android.phone.SipBroadcastReceiver p=0 o=0 m=0x108000}  
E/ActivityManager( 216): ANR in com.android.phone   
E/ActivityManager( 216): Reason: Broadcast of Intent { act=android.net.sip.SIP\_SERVICE\_UP cmp=**com.android.phone/.SipBroadcastReceiver** }  
E/ActivityManager( 216): Load: 8.63 / 8.46 / 8.55  
E/ActivityManager( 216): CPU usage from 5479ms to 0ms ago:  
E/ActivityManager( 216): **94% 5005/android.process.acore**: 88% user + 6.1% kernel / faults: 1507 minor  
E/ActivityManager( 216): 0.7% 216/system\_server: 0.3% user + 0.3% kernel / faults: 7 minor  
E/ActivityManager( 216): **98% TOTAL: 89% user + 6.3% kernel + 3.2% iowait**E/ActivityManager( 216): CPU usage from 921ms to 1445ms later with 99% awake:  
E/ActivityManager( 216): **92% 5005/android.process.acore**: 86% user + 5.6% kernel / faults: 107 minor  
E/ActivityManager( 216): 88% 5011/Binder Thread #: 84% user + 3.7% kernel  
E/ActivityManager( 216): 3.7% 216/system\_server: 0% user + 3.7% kernel / faults: 2 minor  
E/ActivityManager( 216): 3.7% 237/ActivityManager: 0% user + 3.7% kernel  
E/ActivityManager( 216): 1.8% 1071/com.aricent.mtp: 0% user + 1.8% kernel  
E/ActivityManager( 216): 1.8% 5811/MtpHandlerServi: 0% user + 1.8% kernel  
E/ActivityManager( 216): +0% 14593/flush-179:0: 0% user + 0% kernel  
E/ActivityManager( 216): **100% TOTAL: 88% user + 9.6% kernel + 1.9% iowait**

DALVIK THREADS:  
(mutexes: tll=0 tsl=0 tscl=0 ghl=0 hwl=0 hwll=0)  
"main" prio=5 tid=1 NATIVE

| group="main" sCount=1 dsCount=0 obj=0x2aad2248 self=0xcf70  
 | sysTid=14558 nice=0 sched=0/0 cgrp=[fopen-error:2] handle=1876218976  
 at android.os.BinderProxy.transact(Native Method)  
 at android.database.BulkCursorProxy.close(BulkCursorNative.java:312)  
 at android.database.BulkCursorToCursorAdaptor.close(BulkCursorToCursorAdaptor.java:143)  
 at android.database.**CursorWrapper.close**(CursorWrapper.java:45)  
 at android.content.ContentResolver$CursorWrapperInner.close(ContentResolver.java:1396)  
 at com.android.phone.NotificationMgr$**QueryHandler.onQueryComplete**(NotificationMgr.java:351)  
 at android.content.**AsyncQueryHandler**.handleMessage(AsyncQueryHandler.java:344)  
 at android.os.Handler.dispatchMessage(Handler.java:99)  
 at android.os.Looper.loop(Looper.java:123)  
 at android.app.ActivityThread.main(ActivityThread.java:3701)  
 at java.lang.reflect.Method.invokeNative(Native Method)  
 at java.lang.reflect.Method.invoke(Method.java:507)  
 at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:862)  
 at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:620)  
 at dalvik.system.NativeStart.main(Native Method)

**Main thread is blocked on CursorWrapper.close in onQueryComplete function which runs in the main thread of com.android.phone. Broadcast has no chance to run, then timeout.**

**In this case, android.process.acore occupies too much CPU. So, it’s suspicious. From the log we can know, broadcast timeout is from 09:33:47-09:33.57 (10S). We can check what’s system doing in this period of time. In event logs, from 09:33:47-09:33.57, We find intensive database operations like following.**

**05-30 09:33:47.439 I/db\_sample( 5005): [/data/data/com.android.providers.contacts/databases/contacts2.db....**

**Maybe this is the root cause.**

**Remarks: sendBroadcast(intent) will post a msg to main thread messagequeue of receiver’s hosting process, then looper will get out the msg and call into onReceive function. But if main thread is blocked by some reason, the msg will not be dequeued and onReceive will not be invoked. Then ANR will occur in such case.**

# ServiceTimeout1 Keywords: Blocked in Service onDestroy

W/ActivityManager( 228): Timeout executing service: ServiceRecord{2b133288 com.aricent.mtp/.MtpService}  
E/ActivityManager( 228): ANR in com.aricent.mtp  
E/ActivityManager( 228): Reason: Executing service com.aricent.mtp/.MtpService  
E/ActivityManager( 228): Load: 9.27 / 9.35 / 7.3  
E/ActivityManager( 228): CPU usage from 5070ms to 0ms ago with 99% awake:  
E/ActivityManager( 228): 27% 228/system\_server: 18% user + 9.6% kernel / faults: 96 minor  
E/ActivityManager( 228): 10% 1457/com.android.settings: 10% user + 0.1% kernel / faults: 129 minor  
E/ActivityManager( 228): 2.5% 1449/com.android.commands.monkey: 2.3% user + 0.1% kernel / faults: 23 minor  
E/ActivityManager( 228): 1.1% 822/com.sonyericsson.textinput.uxp: 0.9% user + 0.1% kernel / faults: 57 minor  
E/ActivityManager( 228): 0.3% 1484/com.sonyericsson.music: 0.3% user + 0% kernel  
E/ActivityManager( 228): 0.1% 106/mediaserver: 0% user + 0.1% kernel  
E/ActivityManager( 228): 0% 1438/logcat: 0% user + 0% kernel  
E/ActivityManager( 228): 0% 1467/com.sonyericsson.customization: 0% user + 0% kernel / faults: 4 minor  
E/ActivityManager( 228): **43% TOTAL: 32% user + 10% kernel + 0.3% iowait** **🡺 CPU is not high, ANR is possibly caused by block**  
E/ActivityManager( 228): CPU usage from 386ms to 899ms later:  
……

DALVIK THREADS:

(mutexes: tll=0 tsl=0 tscl=0 ghl=0 hwl=0 hwll=0)  
"main" prio=5 tid=1 NATIVE  
| group="main" sCount=1 dsCount=0 obj=0x2aad2248 self=0xcf70  
| sysTid=1387 nice=0 sched=0/0 cgrp=[fopen-error:2] handle=1876218976  
at **com.aricent.mtp.MtpService.StopMtp**(Native Method)  
**at com.aricent.mtp.MtpService.onDestroy**(MtpService.java:296)  
at android.app.ActivityThread.handleStopService(ActivityThread.java:2073)  
at android.app.ActivityThread.access$2900(ActivityThread.java:121)  
at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1005)  
at android.os.Handler.dispatchMessage(Handler.java:99)  
at android.os.Looper.loop(Looper.java:123)  
at android.app.ActivityThread.main(ActivityThread.java:3688)  
at java.lang.reflect.Method.invokeNative(Native Method)  
at java.lang.reflect.Method.invoke(Method.java:507)  
at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:862)

**Fix:   
*“StopMtp funcion is blocked by read() in main thread. Solution: use ioctl to make read() unblocked.”***

**Remarks: If having crash dump, we can see into native call stacks, then we’ll get the root cause as the fix has said. And, I think we might also get to know the cause by appropriate logs in StopMtp.**

# AffectEachOtherTimeout1 Keywords: Service causes KeyDispatchTimeOut ANR

I/InputDispatcher( 236): Application is not responding: AppWindowToken{2b09cda0 token=HistoryRecord{2b4a3d90 com.sonyericsson.conversations/.ui.ConversationActivity}} - Window{2b235e68 com.sonyericsson.conversations/com.sonyericsson.conversations.ui.ConversationActivity paused=false}. 5006.1ms since event, 5005.6ms since wait started  
  
I/WindowManager( 236): **Input event dispatching timed out sending to com.sonyericsson.conversations/com.sonyericsson.conversations.ui.ConversationActivity**  
E/ActivityManager( 236): CPU usage from 6071ms to 0ms ago:  
E/ActivityManager( 236): 12% 564/com.android.phone: 7.4% user + 5% kernel / faults: 57 minor  
E/ActivityManager( 236): 5.7% 19749/com.sonyericsson.conversations: 4.7% user + 0.9% kernel / faults: 99 minor  
E/ActivityManager( 236): 4.4% 236/system\_server: 3.4% user + 0.9% kernel / faults: 61 minor  
E/ActivityManager( 236): 1.1% 10153/com.sonyericsson.eventstream.telephonyplugin: 0.6% user + 0.4% kernel / faults: 37 minor  
E/ActivityManager( 236): 0% 601/com.google.process.gapps: 0% user + 0% kernel / faults: 27 minor  
E/ActivityManager( 236): 0.3% 10283/app\_process: 0.1% user + 0.1% kernel  
E/ActivityManager( 236): 0.3% 10332/com.sonyericsson.eventstream: 0.3% user + 0% kernel / faults: 5 minor  
E/ActivityManager( 236): **87% TOTAL: 17% user + 8.2% kernel + 61% iowait**  
E/ActivityManager( 236): CPU usage from 1837ms to 2366ms later:  
E/ActivityManager( 236): 26% 236/system\_server: 23% user + 3.8% kernel / faults: 47 minor  
E/ActivityManager( 236): 3.8% 244/SurfaceFlinger: 1.9% user + 1.9% kernel  
E/ActivityManager( 236): 3.8% 263/ActivityManager: 3.8% user + 0% kernel  
E/ActivityManager( 236): 3.8% 482/InputDispatcher: 3.8% user + 0% kernel  
E/ActivityManager( 236): 1.9% 242/Binder Thread #: 1.9% user + 0% kernel  
......  
E/ActivityManager( 236): 100% TOTAL: 47% user + 20% kernel + 30% iowait + 1.8% softirq

Cmd line: com.sonyericsson.conversations  
DALVIK THREADS:  
(mutexes: tll=0 tsl=0 tscl=0 ghl=0 hwl=0 hwll=0)  
"main" prio=5 tid=1 NATIVE  
 | group="main" sCount=1 dsCount=0 obj=0x2aad2248 self=0xcf70  
 | sysTid=19749 nice=0 sched=0/0 cgrp=[fopen-error:2] handle=1876218976  
 at android.os.BinderProxy.transact(Native Method)  
 at android.content.ContentProviderProxy.update(ContentProviderNative.java:493)  
 at android.content.**ContentResolver.update**(ContentResolver.java:730)   
 at com.google.android.mms.util.SqliteWrapper.update(SqliteWrapper.java:91)  
 **at com.android.mms.transaction.****TransactionService.onStart(TransactionService.java:280)  
 at android.app.Service.onStartCommand(Service.java:428)  
 at android.app.ActivityThread.handleServiceArgs(ActivityThread.java:2060)  
 at android.app.ActivityThread.access$2800(ActivityThread.java:121)  
 at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1006)**  
 at android.os.Handler.dispatchMessage(Handler.java:99)  
 at android.os.Looper.loop(Looper.java:123)  
 at android.app.ActivityThread.main(ActivityThread.java:3701)  
 at java.lang.reflect.Method.invokeNative(Native Method)  
 at java.lang.reflect.Method.invoke(Method.java:507)  
 at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:866)  
 at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:624)  
 at dalvik.system.NativeStart.main(Native Method)

**Service runs in the main thread of its hosting process, startService/bindServie will actually enqueue a message on the main thread looper, if there are also activities in the process, the activities and the service will share the looper, so service will affect the activities’ running. For example, if service is running in onStart method, then the activities will not respond to any keyevent at that time. So, keydispatch timeout ANR will occur because of servive doesn’t finish handling in time.**

**In this case, main thread is blocked by database operation in TransactionService.onStart. That caused com.sonyericsson.conversations.ui.ConversationActivity** **KeyDispatchTimeOut ANR.**

**So, although service timeout is 20s, but if there are activities in the hosting process , it’s better finished in 5s.**

# AffectEachOtherTimeout2 Keywords: Activity causes BroadcastTimeOut ANR

ActivityManager(67): **ANR in com.semc.test**  
ActivityManager(67): Reason: Broadcast of Intent { **act=com.semc.test.action.TestBR cmp=com.semc.test/.MyBR** }  
ActivityManager(67): Load: 0.54 / 0.51 / 0.36  
ActivityManager(67): CPU usage from 10255ms to 0ms ago:  
ActivityManager(67): 0.6% 67/system\_server: 0.3% user + 0.2% kernel / faults: 35 minor 1 major  
ActivityManager(67): 0.3% 122/com.android.systemui: 0.1% user + 0.1% kernel / faults: 8 minor  
ActivityManager(67): 0% 40/adbd: 0% user + 0% kernel  
ActivityManager(67): 0% 121/com.android.phone: 0% user + 0% kernel / faults: 7 minor  
ActivityManager(67): 2.3% TOTAL: 0.9% user + 1.2% kernel + 0.1% softirq  
ActivityManager(67): CPU usage from 799ms to 1416ms later:  
ActivityManager(67): 6.6% 67/system\_server: 1.6% user + 5% kernel / faults: 1 minor  
ActivityManager(67): 8.3% 80/ActivityManager: 1.6% user + 6.6% kernel  
ActivityManager(67): 23% TOTAL: 7.6% user + 15% kernel

----- pid 360 at 2011-06-21 10:46:30 -----  
Cmd line: com.semc.test

DALVIK THREADS:  
(mutexes: tll=0 tsl=0 tscl=0 ghl=0 hwl=0 hwll=0)  
"main" prio=5 tid=1 TIMED\_WAIT  
 | group="main" sCount=1 dsCount=0 obj=0x4001f1a8 self=0xce48  
 | sysTid=360 nice=0 sched=0/0 cgrp=default handle=-1345006528  
 | schedstat=( 1477043171 2062931362 192 )  
 at java.lang.VMThread.sleep(Native Method)  
 at java.lang.**Thread.sleep**(Thread.java:1213)  
 at java.lang.Thread.sleep(Thread.java:1195)  
 at **com.semc.test.test\_dispatch$2.onPreferenceClick**(test\_dispatch.java:137)  
 at android.preference.Preference.performClick(Preference.java:812)  
 at android.preference.PreferenceScreen.onItemClick(PreferenceScreen.java:190)  
 at android.widget.AdapterView.performItemClick(AdapterView.java:284)  
 at android.widget.ListView.performItemClick(ListView.java:3513)  
 at android.widget.AbsListView$PerformClick.run(AbsListView.java:1800)  
 at android.os.Handler.handleCallback(Handler.java:587)  
 at android.os.Handler.dispatchMessage(Handler.java:92)  
 at android.os.Looper.loop(Looper.java:123)  
 at android.app.ActivityThread.main(ActivityThread.java:3647)  
 at java.lang.reflect.Method.invokeNative(Native Method)  
 at java.lang.reflect.Method.invoke(Method.java:507)  
 at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:839)  
 at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:597)  
 at dalvik.system.NativeStart.main(Native Method)

**This is a test i made, not a real DMS issue. I want to show how activity affects BroadcastReceiver.**

**In this case, BroadcastReceiver(MyBR) and test\_dispatch Activity are in the same process com.semc.test. I let the main thread go to sleep and then send a broadcast from a sub thread. The BroadcastTimeout ANR occurs after 10s.**

**This example has the root cause with former example BroadcastTimeout3, the onReceive function hasn’t been invoked yet.**