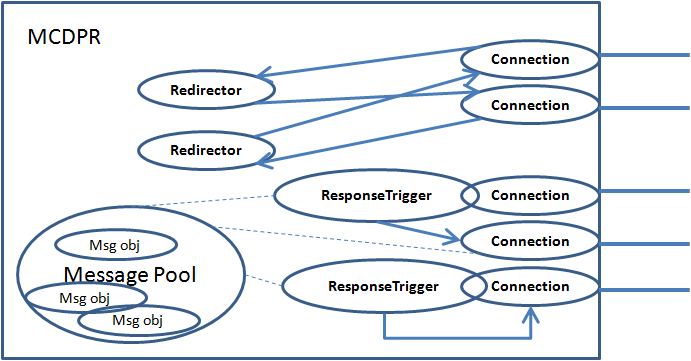
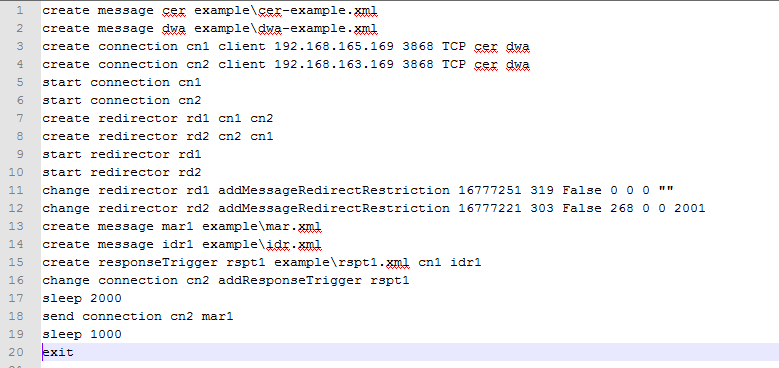
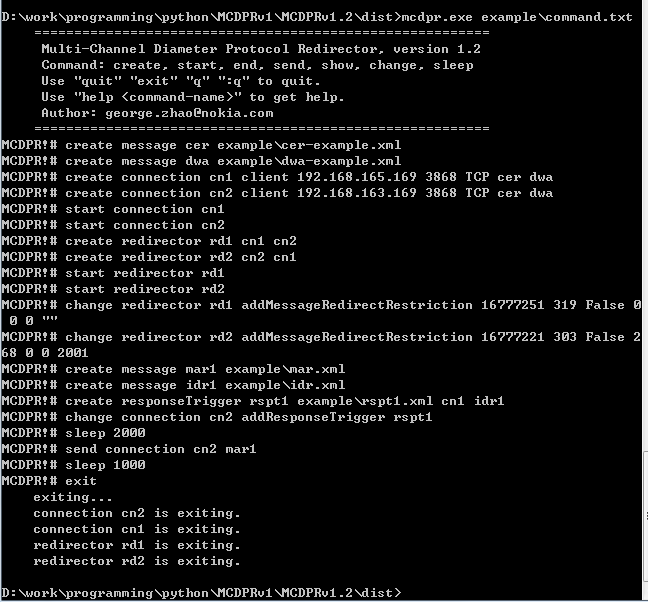
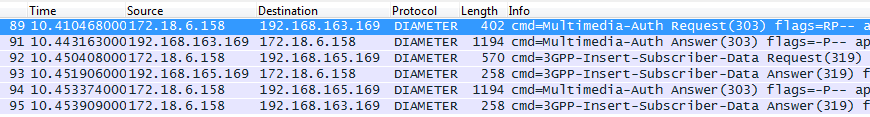
**Multi-Connection Diameter Protocol Redirector 1.5 Guide2**

# MCDPR进阶 – MCDPR内部工作原理

1. MCDPR是一个有限的状态机，它内部的各个部件，根据自己的配置，进行工作。部件之间的关系可以通过下图了解。
2. 根据上图，我们应该这样理解MCDPR：
   1. Connection用于连接外部的Diameter Peer。
   2. Redirector可以单向的在两个Connection之间转发消息，并且可以设定原则来禁止转发某些消息。
   3. ResponseTrigger依附于某一个Connection，当这个Connection上有消息到来的时候，ResponseTrigger进行判断，所有Match的情况，ResponseTrigger会给其他的连接，当然也可以给自己这个连接发送一个消息。
   4. 虚线代表了使用的关系，ResponseTrigger, Connection, 都可以直接使用Message对象，并且将他们发送出去。

# MCDPR进阶 – 示例文件分析

1. 在example目录里面有几个文件，其中的command.txt内容如下：这个示例并没有太多的实用价值，它的主要目的是向大家展现如何使用MCDPR，并且MCDPR内部组件的工作原理。
2. 下面来解释下这个command.txt。
   1. 建立一个消息对象cer，使用example目录下的cer-example.xml来建立。
   2. 建立一个消息对象dwa，使用example目录下的dwa-example.xml来建立。
   3. 建立一个连接对象cn1，参数如图上所示。
   4. 建立一个连接对象cn2，参数如图上所示。
   5. 运行cn1
   6. 运行cn2
   7. 建立一个redirector，让它转发从cn1来的所有消息到cn2
   8. 建立一个redirector，让它转发从cn2来的所有消息到cn1
   9. 运行rd1
   10. 运行rd2
   11. 为rd1加入一个禁止转发原则。当application id是16777251，当commandcode是319，是一个Answer的时候，该消息被禁止转发。
   12. 为rd2加入一个禁止转发原则。当application id是16777221，当commandcode是303，是一个Answer的时候，且AVP code是268的值是2001的时候，该消息被禁止转发。
   13. 建立一个消息对象mar1，使用example\mar.xml建立
   14. 建立一个消息对象idr1，使用example\idr.xml建立
   15. 建立一个responseTrigger对象rspt1，使用example\rspt1.xml建立，如果匹配成功，则在cn1上发送idr1这个消息对象。
   16. 将rspt1依附在cn2上
   17. 休息2秒
   18. 在cn2上发送mar1消息对象。
   19. 休息1秒
   20. 退出
3. 可能已经被观察到，这里的第11和第12步已经被注释掉了，原因在于这两步对当前这个脚本的影响很大，我们后面会分析到。
4. 先来看看脚本被运行后的效果。
5. 下面是wireshark的结果。



可以看到一共有6条消息，这是为什么呢，一起来分析下：

#89，这条消息就是第18步在cn2上发送的mar1。

#91，cn2对端的HSS回复了一条MAA。

#92，rspt1的匹配命中了，rspt1在依附的时候定义的配置是，如果命中则在cn1上发送消息idr1。

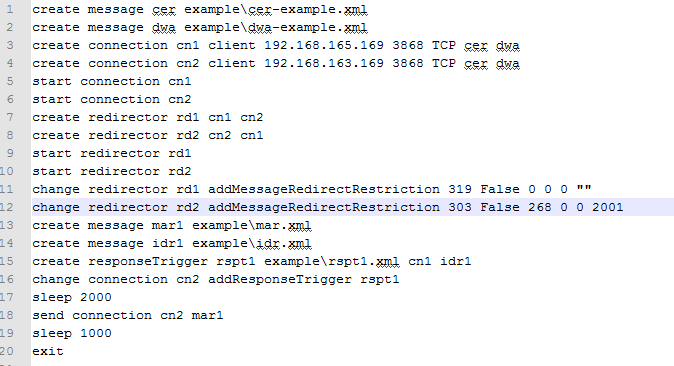
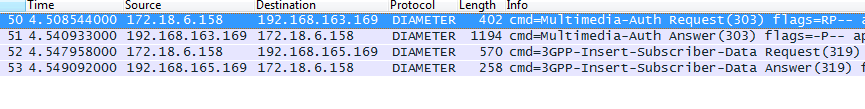
#93，收到cn1对端的HSS回复的一条IDA。

#94，由于有rd2的存在，所以#91的MAA被转发给了cn1

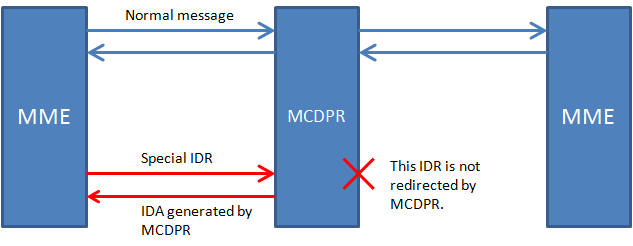
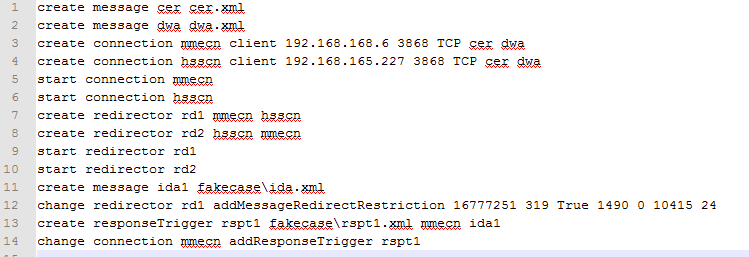
#95，由于有rd1的存在，所以#93的IDA被转发给了cn2

Redirector和responseTrigger相互独立的工作，效果都是正确的。

但是，很明显，在一般的测试中，我们都希望类似于#94和#95这样的消息终止在模拟器上就可以了。但是由于有redirector的存在，类似#94，#95这样的Answer消息是会被转发的，所以我们必须配置禁止转发原则来禁止他们的转发。

1. 我们将11行和12行的注释关掉。command.txt内容如下图：
2. 可以看到由于有禁止转发原则的存在，wireshark的内容有所变化：只有4条消息，而之前的MAA和IDA都不被转发了。

# MCDPR进阶 – 实际案例分析

1. 由于环境的问题，作者目前很难找到环境来测试一个真正的例子了，所以在这里设计了一个问题（之前做过类似的），并且写了一个命令脚本（并未经过测试），供大家参考。
2. 问题是一个异常处理的case：MME和HSS连接，手机作为VOLTE客户端注册，并且可以打电话，在某一个特定的IDR到达的时候，我们需要响应一个带有错误码的IDA。
3. 思路：
   1. 将MCDPR放置在MME和HSS之间，并和他们都建立连接。
   2. 首先要保障除了那个特定的IDR到达的情况下，其他所有的Diameter消息都可以在MME和HSS之间交互。
   3. 侦测到特别的IDR，则根据需要组织带有错误码的IDA返回。
   4. 这个特别的IDR，不被转发到HSS上。
4. 下面是一个示意的脚本，注意这个脚本没有经过测试，只是示意而已。
5. 下面是一个解释：
   1. 建立一个消息对象cer，使用example目录下的cer.xml来建立。
   2. 建立一个消息对象dwa，使用example目录下的dwa.xml来建立。
   3. 建立一个连接对象mmecn，参数如图上所示。
   4. 建立一个连接对象hsscn，参数如图上所示。
   5. 运行mmecn
   6. 运行hsscn
   7. 建立一个redirector，让它转发从mmecn来的所有消息到hsscn
   8. 建立一个redirector，让它转发从hsscn来的所有消息到mmecn
   9. 运行rd1
   10. 运行rd2
   11. 建立一个消息ida1，这就是我们用于回复那个特别的IDR而准备的answer。
   12. 建立一个禁止转发原则，application id为16777251，commandcode为319, AVP是10415的1490的话（IDR-Flag），且如果这个AVP的值为24，则不进行转发。这个正是我们所说的那个特别的IDR。
   13. 建立一个responseTrigger，这个Trigger当收到特别的IDR时被触发。
   14. 将这个Trigger附着在mmecn上。
6. 上面的思路是用禁止转发原则来禁止特别的IDR转发到HSS，然后用responseTrigger来产生一个来自于MCDPR的IDA发送给MME。
7. 类似的case有很多，大都是利用responseTrigger和禁止转发原则来灵活搭建。

# 结束语

MCDPR的设计相对来说比较灵活，通过对内部组件的灵活搭建，可以实现很多不同的功能，如果大家在使用过程中有任何好的成功案例，希望大家可以分享到[george.zhao@nokia.com](mailto:george.zhao@nokia.com)，多谢。

由于时间的仓促，问题很多，还有很多功能没有完整。如果有对MCDPR感兴趣，并且使用后发现问题的朋友，可以发送邮件到[george.zhao@nokia.com](mailto:george.zhao@nokia.com)来询问。