**切换说明**

# 一环境搭建

## 下载代码

1. 基站： svn co svn://10.21.1.188/lte\_henb/trunk
2. 终端：svn co svn:// 10.21.1.188/lte\_ue/branches/mac\_phy\_api
   1. **编译和启动**
3. 启动顺序 MME -> henb -> UE
4. 启动MME :目前 MME程序是由python编写，在基站侧代码中，具体路径在 debug/s1ap\_tool/pysctp 下的ctl\_mme.py，运行时在linux终端下访问到mme目录，输入sudo ./ctl\_mme.py即可（默认启动mme为本地地址，端口号为12345），在运行mme时也可带参数，从而指定端口号，例如 **sudo ./ctl\_mme.py 24000** 即说明mme的SCTP端口号被设定为24000；密码是jinjie。
5. 启动源基站：在henb的目录下，在linux终端遍历到/bin下：

* 首先对程序进行编译，在bin目录下输入**make -f ho\_s\_henb.mk**即可完成编译，若需要重新编译所有代码，可先执行**make clean**，然后在执行**make -f ho\_s\_henb.mk**命令。（注意：若每一次对代码修改后，需要重新执行make进行编译）
* 其次代码中包含很多条件编译的宏，在makefile文件中，若有不同的需要可以按需 求添加相应的宏。然后执行**make clean**，再次执行**make -f ho\_s\_henb.mk**编译程序产生可执行文件./ltehenb\_s.host ::1 -p 85677。
* 执行./ltehenb\_s.host ::1 -p 85677

1. 启动目标基站：在henb的目录下，在linux终端遍历到/bin下：

* 首先对程序进行编译，在bin目录下输入**make -f ho\_t\_henb.mk**即可完成编译，若需要重新编译所有代码，可先执行**make clean**，然后在执行**make -f ho\_t\_henb.mk**命令。（注意：若每一次对代码修改后，需要重新执行make进行编译）
* 其次代码中包含很多条件编译的宏，在makefile文件中，若有不同的需要可以按需求添加相应的宏。然后执行**make clean**，再次执行**make -f ho\_t\_henb.mk**编译程序产生ltehenb\_t.host可执行文件
* 执行./ltehenb\_t.host 10.21.1.188 -p 12345

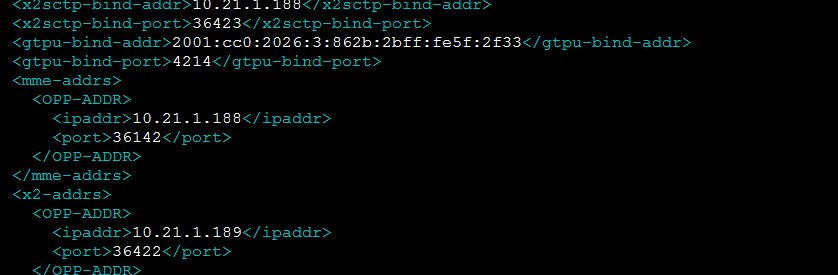
1. 启动终端，进入UE的目录

* 首先对程序进行编译，由于目前终端测代码和基站侧代码采用不同的编译工具，因此编译工具略有不同，如果代码位置发生移动，在UE的目录下包含CMakeCache.txt文件会失效，因此首先要删除这个文件，执行 **rm CMakeCache.txt** 即可。
* 其次删除CMakeCache.txt文件后，运行**cmake .** 生成make工具。
* 然后输入**make clean**命令删除曾经编译好的一些旧文件，之后输入**make**命令编译程序。
* 执行./lteue

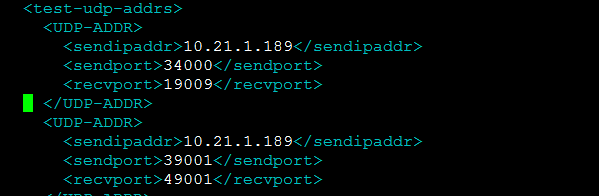
**1.3配置IP或者端口号**

1. 源基站端口号和IP配置（***注意源基站的地址是188， 目的基站是189***）

在bin中的henb.xml文件中配置



在如图出配置x2切换地址和端口号



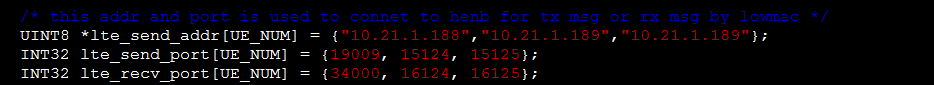
在如图处配置源基站给UE的IP和发送端口号和接收端口号

1. 目标基站和源基站配置端口号的位置一样

*需要说明的是在源基站和UE通信时源基站的发送的IP是UE所在的IP源基站发送的端口号是UE接收源基站的的端口号，源基站接收的端口号是UE发送的端口号。目标基站接收的端口号是UE在执已经改变的端口号具体位置在UE代码中的在ho\_ue.c中alter\_udp\_addr(0, "10.21.1.189", 18012)*

1. 终端配置端口号和IP

在source/main\_src中的test\_ue\_sys.c文件中



在如图上配置IP和端口号

*需要说明的是UE发送的IP地址是基站所在的IP地址，终端发送的端口号是基站接收的端口号，终端接收的端口号是基站发送的端口号*

# 二 现在X2切换需要进一步优化的地方

## 2.1 PDCP状态报告

现在基站侧发送和处理状态报告都只考虑了顺序的数据包丢失问题没有考虑乱序丢包以后的优化可以从这里考虑。具体见36.323协议

## 2.2 转发数据时

现在源基站转发数据时包括两部分数据：一种是源基站没有收到UE反馈ACK的数据包这些数据包必须加上SN号才能转发给UE，而另一种源基站转发的数据是刚从MME接收的数据此时的数据包源基站还没有给其分配SN号所以转发这些数据时没有转发该数据包的SN号。这样导致必须用两个函数才能完成数据转发比较繁琐。因此需要优化，

参考改进思路：

可以整合两个函数为一个函数。具体是当源基站转发完PDCP实体的数据包时，并且开始转发来自MME的数据时，可以在此时给该数据包分配SN号，最终将该数据包和该数据包的SN号一块发个目标基站，这样可以和转发源基站没有收到UE反馈ACK 的数据包一致。

### 2.3一份基站代码同时是源基站又是目标基站问题

现在虽然实现了源基站和目标基站使用一份代码但是此份代码不能在同一时间即是源基站有是目标基站。因此需要进一步优化。