Peer A在开始时对peers中的尝试连接，对于未能连接上的之后是否要再进行尝试连接？

Peer B的configuration中的peers有peer A，那么peer A的configuration中的peers是否一定要有peer B?

潜在的deadlock: connectedPeerList

Invalid protocol的回应，将checkcommand改成switch形式比较好？

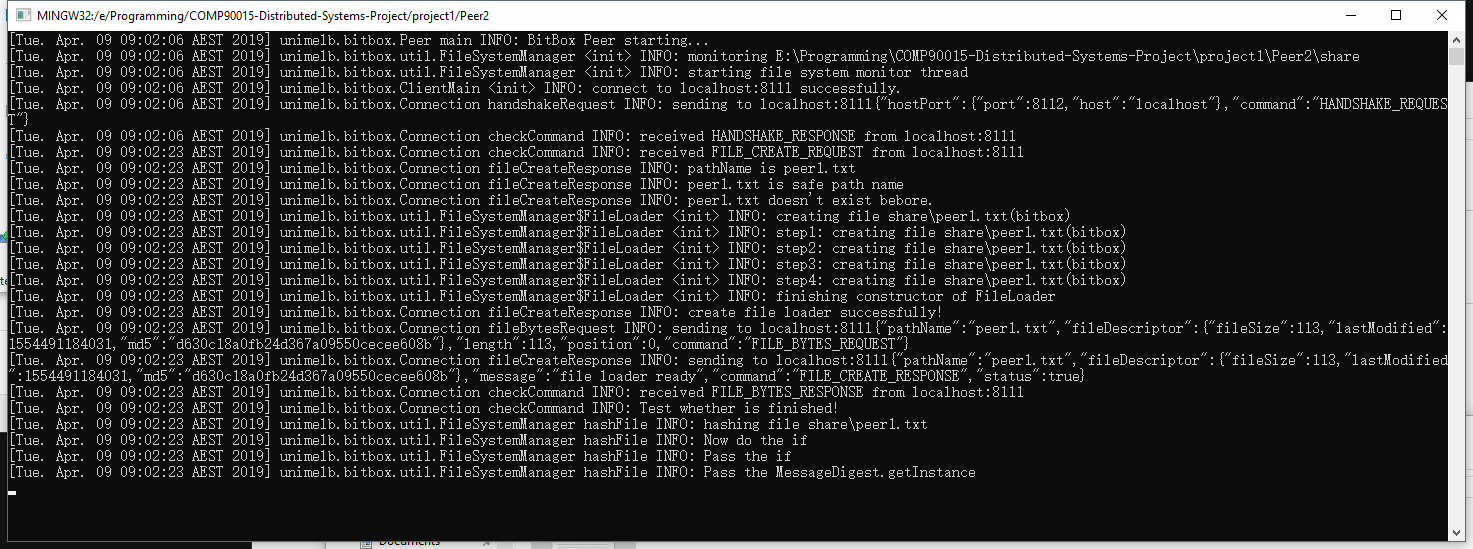
~~Peer A 先启动，PeerB后启动，B连A，A中的同步文件夹会同步到B, 但B中的文件event还不会通知到A。(单连接)。~~

在win下文件夹bug:

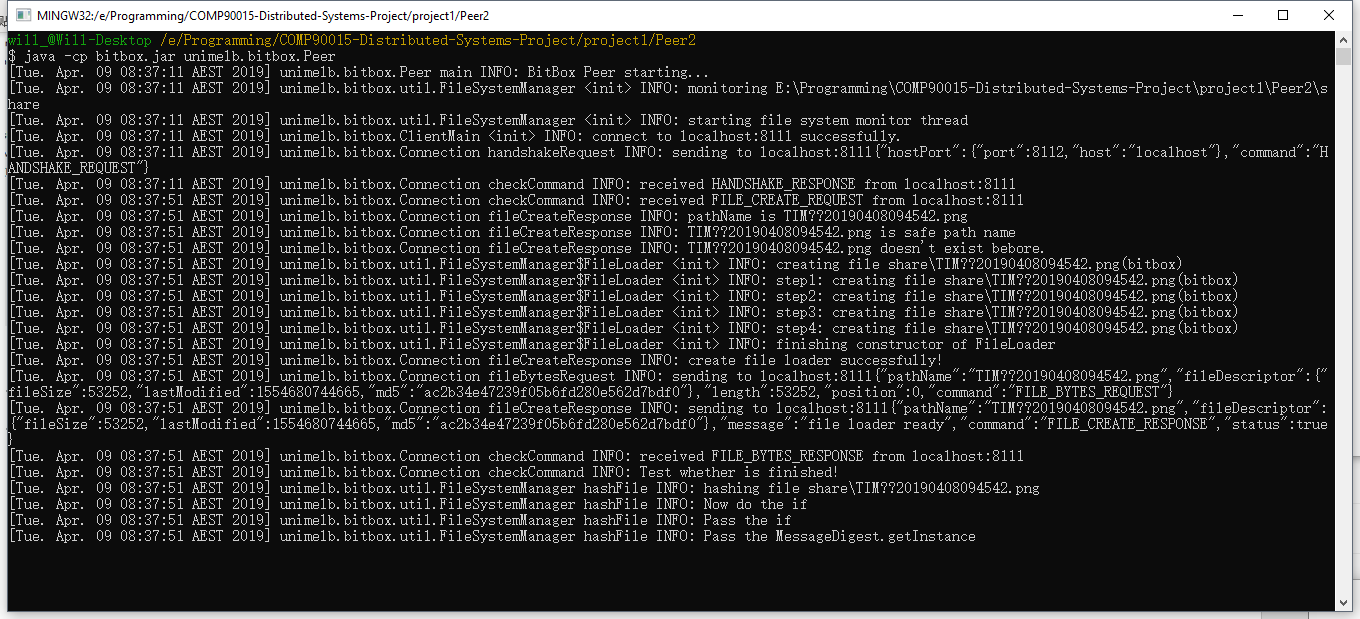
Peer A创建一个空文件再改名，Peer B中会保留默认名的文件夹。

删除有bug，当删除的文件夹中有文件时有bug。

* ~~在win的jdk11下，完成传输文件，但卡在验证。~~

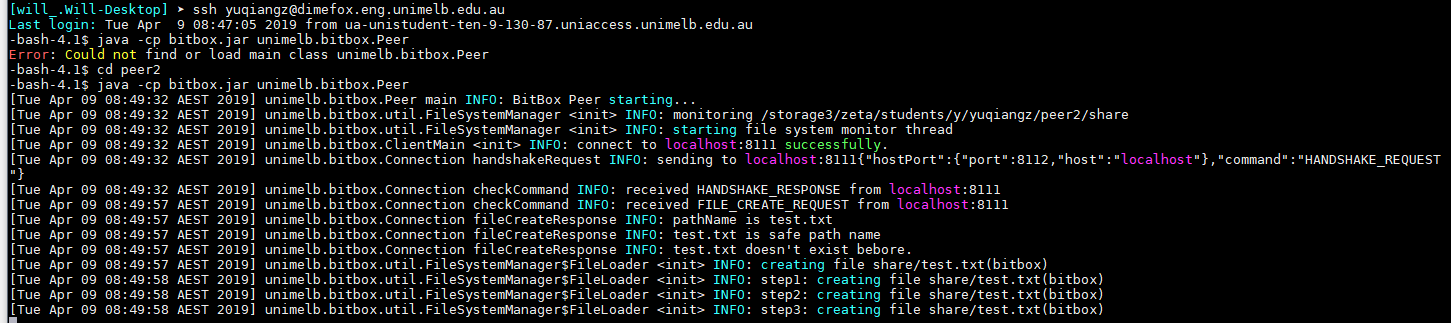


* ~~在win的jdk12下，完成传输文件，但卡在验证-使用win下生成的runnable jar一样~~



~~问题在while ((bytesCount = fis.read(byteArray)) != -1) {~~

* ~~在uni server(jdk1.8)下，仅能创建空文件，没有传输文件数据-使用win下生成的runnable jar一样~~

~~~~

~~卡在lock = channel.lock();~~

~~在连接的一开始进行文件同步~~

~~大文件传输测试~~

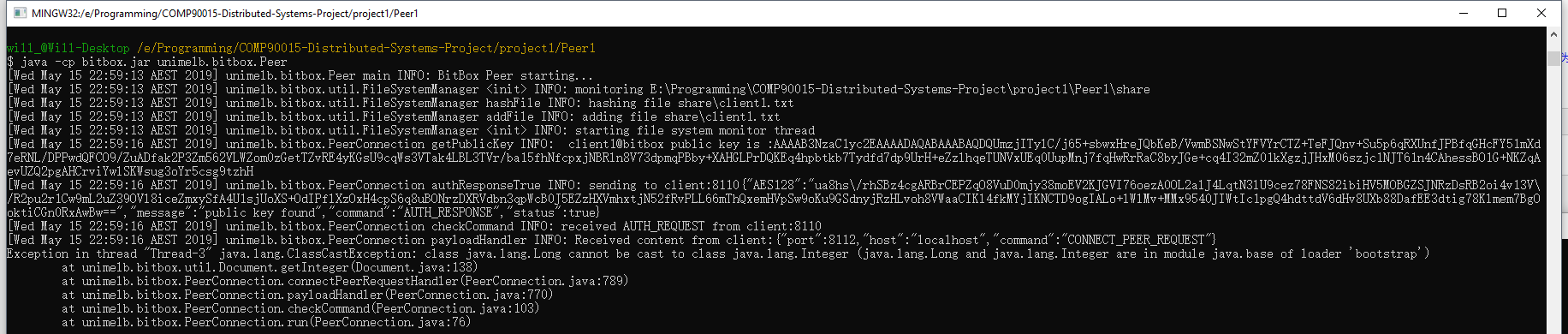
Project2

关于invalid\_protocol好像没有实施完全。

REPORT可对AES的4中不同加密方式进行优劣探讨

发送命令的指示可在sendMessage中使用。

CONNECT\_PEER\_RESPONSE当目标之前已连接该怎么回？



关于Document中的getInteger方法报错，目前解决方法：

String temp = "" + doc.get("port");

int givenPort = Integer.parseInt(temp);

part2：是否要TCP和UDP混用？还是连接的peer都是相同的mode？

BASE64之后的长度变为原来的int encodedLength = (int) (Math.ceil((double)(a.length())/3) \* 4);

UDP并不适合bitbox，一次性的传输限制。

在传一个大文件途中在传其他的会有问题?待测试

ClientPort是client去连接的口，client本身是哪个口未指出。

在UDP下是不知道对方有没有断开链接：现在应对udp模式下连接，断开，再断开会导致invalid protocol，目前方案断开时发送CONNECTION\_REFUSED

在vm上port要在7900 -8000

|  |  |
| --- | --- |
| Send | Want to receive |
| ~~HANDSHANKE\_REQUEST~~ | CONNECTION\_REFUSED, HANDSHAKE\_RESPONSE |
| ~~FILE\_CREATE\_REQUEST~~ | FILE\_CREATE\_RESPONSE |
| ~~FILE\_BYTES\_REQUEST~~ | FILE\_BYTES\_RESPONSE(Position = 0) |
| ~~FILE\_BYTES\_RESPONSE(未完成)~~需修改。 | FILE\_BYTES\_REQUEST |
| ~~FILE\_DELETE\_REQUEST~~ | FILE\_DELETE\_RESPONSE |
| ~~FILE\_MODIFY\_REQUEST~~ | FILE\_MODIFY\_RESPONSE |
| ~~DIRECTORY\_CREATE\_REQUEST~~ | DIRECTORY\_CREATE\_RESPONSE |
| ~~DIRECTORY\_DELETE\_REQUEST~~ | DIRECTORY\_DELETE\_RESPONSE |
| CONNECTION\_REFUSED |  |
| HANDSHAKE\_RESPONSE |  |
| FILE\_CREATE\_RESPONSE | 似乎对丢包无影响？ |
| FILE\_DELETE\_RESPONSE |  |
| FILE\_MODIFY\_RESPONSE |  |
| DIRECTORY\_CREATE\_RESPONSE |  |
| DIRECTORY\_DELETE\_RESPONSE |  |
| INVALID\_PROTOCOL |  |