1. hiveNetworkCommon::hiveIsAdapterAvailable()要求提供一个字符串作为输入参数，对于常见的UDP，TCP，IB等网络，需要在相应的工程内部定义一个常量字符串来表明相应的网络，否则用户可能不知道字符串和网络的对应关系（例如IB对应的字符串是“mellanox”，谁敢保证所有用户都知道这个字符串对应的是IB？）；
2. TestCase-0700不完善，需要增加如下三个验证：a)是否有普通的网卡？b)是否有无线网卡？3)hiveNetworkCommon::hiveIsAdapterVailable()如果输入一个错误的字符串，是否返回false？（否则我可以怀疑你这个函数的实现就是不论什么字符串都返回true）；
3. 增加TestCase-0700的文档到Confluence；
4. hiveNetworkCommon::CPhysicalNode是否有必要暴露给dll外部，目前在TCP和IB中使用到hiveNetworkCommon::CPhysicalNode的地方，都是为了判断其是否包含某个logical node，而hiveNetworkCommon提供了hiveFindLogicalNodeByProtocal接口，因此实际可以不把hiveNetworkCommon::CPhysicalNode暴露给dll外部；
5. TestCase-0701设计不合理，假设我定义了一个IB和TCP节点，但并未将TCP节点设计为IB节点，测试用例仍然可以通过；
6. 增加TestCase-0701的文档到Confluence；
7. 增加TestCase-0702的文档到Confluence；
8. 增加TestCase-0703的文档到Confluence；
9. 在TestCase-0703的基础上增加一个测试用例，两个节点互发IB数据；
10. 在TestCase-0703的基础上增加一个测试用例，包含三个IB节点，其中两个IB节点向另外一个IB节点发送数据；
11. 在TestCase-0703的基础上增加一个测试用例，包含三个IB节点，两两互发IB数据；
12. TestCase-701在debug x64下无法正常退出；
13. 在TestCase-0702的基础上增加一个测试用例，验证IB发包的数据正确性；
14. TestCase-0703有内存泄漏，但在多线程模式下，以前的方法无法定位到内存泄漏点，需要使用Intel Parallel Studio中的工具来进行定位；
15. 在CIBPacket::extractSourceIP()中，在根据ib\_wc\_t来获取远端的LID时，使用的是vWorkCompletion->recv.ud.remote\_lid，这里为什么是ud而不是rd？需要确认目前使用IB时的数据传输方式，以及这个传输方式是如何确定？
16. 在IB下增加使用RDMA的数据传输方式，并和现有模式进行效率比较；
17. 增加TCP下的异步传输方式，并和同步方式进行效率比较；
18. 目前每建立一个连接，就会创建若干个queue pair，要保证一个节点在和其他多个节点建立连接的时候，创建的queue pair数目不超过adapter允许的最大queue pair数目；