1. Созданный для пересылки пакет данных (ПД) должен дробиться на блоки приемлемой длины.
2. Каждый блок данных (БлД) содержит следующую информацию:

* Префикс для идентификации начала БлД : **char[]**
* Длина БлД : **quint16**
* Идентификатор транзакции : **Guid**
* Идентификатор БлД : **Guid**
* Статус БлД: начальный, очередной или последний : **quint8**
* Идентификатор предыдущего БлД (если блок начальный, то значение 0) : **Guid**
* Данные : **QByteArray**
* Хэш-сумма : **Hash**

1. БлД должен реализовывать интерфейс **IBinarySerializable**.
2. Передача каждого ПД должна создавать новую транзакцию с генерацией уникального
3. Контроля доставки БлД нет
4. Контроля доставки ПД нет
5. При получении БлД
   1. необходимо проверить префикс на соответствие значению, проверить хэш-сумму на корректность
   2. При получении *начального* БлД
      1. создается транзакция
      2. запоминается ее идентификатор
      3. идентификатор текущего БлД
      4. инициализируется буффер для приема
      5. Транзакция помещается в кэш.
   3. При получении *очередного* БлД
      1. В кэше ищется транзакция с указанным идентификатором
      2. Если транзакция не найдена, то
         1. Создается новая транзакция
         2. Запоминается ее идентификатор
         3. Инициализируется буффер для приема
         4. БлД помещается в кэш транзакции
         5. Идентификатор последнего БлД = 0
         6. Транзакция помещается в кэш
      3. Если *Идентификатор предыдущего БлД* соответствует тому, что хранится в кэше, то
         1. Данные дописываются в буффер
         2. Обновляется *Идентификатор предыдущего БлД*
         3. Осуществляется поиск по кэшу текущей транзакции
      4. Если **НЕ** соответствует, то
         1. БлД запоминается в кэш текущей транзакции
   4. При получении *последнего* БлД
      1. В кэше ищется транзакция с указанным идентификатором
      2. Если транзакция не найдена и значение предыдущего БлД == 0, значит транзакция состоит из 1-го БлД
      3. Если транзакция найдена, то
         1. Если *Идентификатор предыдущего БлД* соответствует тому, что хранится в кэше, то
            1. Данные дописываются в буффер
            2. Формируется сигнал о завершении формирования транзакции
         2. Если **НЕ** соответствует, то
            1. БлД запоминается в кэш текущей транзакции
6. Необходимо удалять подвисшие транзакции из кэша
7. Необходимо ограничивать количество хранимых транзакций
8. Необходимо изолировать спам-узлы

Класс **RPCServer**.

Класс, позволяющий реализовывать RPC-протокол. Он открывает порт на прослушивание и ждет TCP подключение по HTTP-протоколу. При получении запроса он парсится и выбирается путь, который определяет вызываемый метод.

Класс **RPCClient**.

Класс выполняет подключения к каждой известной ноде из известного списка.