

# 2034-TCP 协议工作原理及实战--零声教育

## 一、TCP 协议基础知识

1、传输控制协议 ( TCP , Transmission Control Protocol ) 是一种面向连接的、可靠的、基于字节流的传输层通信协议。

2、TCP 拥塞控制算法 ( 也称 AIMD 算法 ) 。该算法主要包括四个主要部分：慢启动、拥塞避免、快速重传和快速恢复。

3、TCP 通信必须建立 TCP 连接 ( 客户端和服务端 ) , Qt 提供 QTcpSocket 类和 QTcpServer 类专门用于建立 TCP 通信程序。服务器端用 QTcpServer 监听端口及建立服务器；QTcpSocket 用于建立连接后使用套接字(socket)进行通信。

4、QTcpServer 是从 QObject 继承的类用于服务器建立网络监听，创建网络 socket 连接。QTcpServer 主要接口函数如下：

### QTcpServer Class

The QTcpServer class provides a TCP-based server. More...

Header: `#include <QTcpServer>`

qmake: `QT += network`

Inherits: `QObject`

Inherited By: `QSctpServer`

- List of all members, including inherited members
- Obsolete members

Note: All functions in this class are `reentrant`.

## Public Functions

```
QTcpServer(QObject *parent = nullptr)
virtual ~QTcpServer()
void close()
QString errorString() const
virtual bool hasPendingConnections() const
bool isListening() const
bool listen(const QHostAddress &address = QHostAddress::Any, quint16 port = 0)
int maxPendingConnections() const
virtual QTcpSocket * nextPendingConnection()
void pauseAccepting()
QNetworkProxy proxy() const
void resumeAccepting()
QHostAddress serverAddress() const
QAbstractSocket::SocketError serverError() const
quint16 serverPort() const
void setMaxPendingConnections(int numConnections)
void setProxy(const QNetworkProxy &networkProxy)
bool setSocketDescriptor(qintptr socketDescriptor)
qintptr socketDescriptor() const
bool waitForNewConnection(int msec = 0, bool *timedOut = nullptr)
```

• 31 public functions inherited from QObject

## QAbstractSocket Class

The `QAbstractSocket` class provides the base functionality common to all socket types. [More...](#)

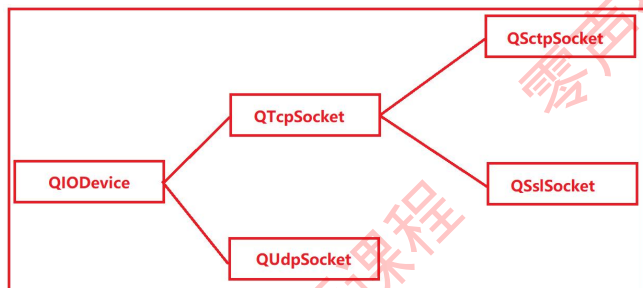
```
Header: #include <QAbstractSocket>
qmake: QT += network
Inherits: QIODevice
Inherited By: QTcpSocket and QUdpSocket
```

- List of all members, including inherited members
- Obsolete members

**Note:** All functions in this class are `reentrant`.

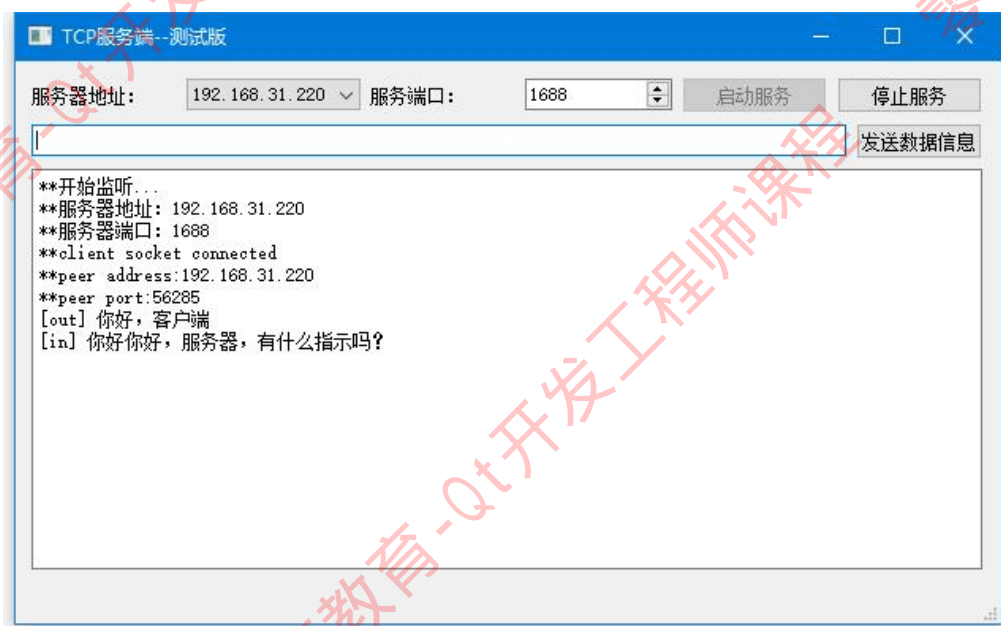
## Public Functions

```
QAbstractSocket(QAbstractSocket::SocketType socketType, QObject *parent)
virtual ~QAbstractSocket()
void abort()
bool bind(const QHostAddress &address, quint16 port = 0, QAbstractSocket::BindMode mode = DefaultForPlatform)
bool bind(quint16 port = 0, QAbstractSocket::BindMode mode = DefaultForPlatform)
virtual void connectToHost(const QString &hostName, quint16 port, QIODevice::OpenMode openMode = ReadWrite,
QAbstractSocket::NetworkLayerProtocol protocol = AnyIPProtocol)
virtual void connectToHost(const QHostAddress &address, quint16 port, QIODevice::OpenMode openMode = ReadWrite)
virtual void disconnectFromHost()
QAbstractSocket::SocketError error() const
bool flush()
bool isValid() const
QHostAddress localAddress() const
quint16 localPort() const
QAbstractSocket::PauseModes pauseMode() const
QHostAddress peerAddress() const
QString peerName() const
quint16 peerPort() const
QNetworkProxy proxy() const
qint64 readBufferSize() const
virtual void resume()
```



## 二、【TCP 应用程序运行结果】

### 1、服务器端



### 2、客户端



### 3、客户端退出或服务器断开运行结果

