

Qt Networking

Miroslav Jaroš

Pb173 Crypto

11. 4. 2016

Agenda

- 1 Networks in general
- 2 Networks in Qt
 - Basic concept
 - Signals and Slots
 - Demo

Computer networks

- Connects computers for information exchange
- Computers communicates using specified protocol
 - Protocol uses each other
 - Http works over TCP
- ISO/OSI model defines 7 layers
- TCP/IP – Internet

TCP/IP I

- IP Internet protocol
 - Addressing computers over network
 - Works on 3rd layer in ISO/OSI
 - IPv4 32 bit address commonly written in 4 digital numbers
127.0.0.1
 - IPv6 128 bit address commonly written as 8 hexadecimal numbers
::1
 - Also adds many other features for peer to peer communication

TCP/IP II

- TCP Transmission control protocol
 - Reliable and ordered delivery
 - Works on 4th layer of ISO/OSI
 - Every connection is quadruple
SOURCE_HOST, SOURCE_PORT, TARGET_HOST, TARGET_PORT
 - Data are delivered complete and in same order as they were transmitted
 - You can't use TCP for broadcasting

TCP/IP III

- UDP User datagram protocol
 - Nonreliable delivery
 - Connection is identified as in TCP but there is no control of validity
 - Best-effort protocol, there is no retransmission or ordering of data
 - On the other hand UDP is in many cases much faster than TCP due to almost nonexistent messaging overhead
 - Can be used for broadcasting, multicasting, anycasting, and unicasting

Networks in programming

- Networking is service provided by OS
- Any network actions are accessible via OS API (Win 32, POSIX)
- Therefore network code is strongly depends on target platform
- Privileged ports
- Nondeterminism: you can't tell when or on which route will be your message delivered
- Parallelism: Many of network events can happen simultaneously

Networks and Qt I

- Creates multiplatform network interface and tools
- All network classes and functions are contained in module network
- Just add `Qt += network` in your `.pro` file
- It contains classes to handle Tcp and Udp connections, SSL connections and even websockets

Networks and Qt II

- Most important classes are:
 - `QTcpSocket` handles single Tcp connection
 - `QTcpServer` creates listening socket and handles new incoming connections
 - `QUdpSocket` handles udp connections – there's no need for `UdpServer`
 - `QSslSocket` handles encrypted connections

QtNetwork and Signals

- You should be probably aware of Signals and Slots concept
- QtNetwork uses signals and slots in same manner as the rest of library
- Sockets emits signals that induces the change of state or any event that happens on socket
- The most important for you are
 - `readyRead` Data were delivered to socket and are ready to be read
 - `disconnected` Socket host has disconnected

Demo

- Simple Client-Server architecture
- Can be found here

`https://github.com/mijaros/QtNetworks.git`

Links

- <http://doc.qt.io> **Qt documentation**
- <http://doc.qt.io/qt-4.8/signalsandslots.html>
Introduction to signals and slots principle
- <http://doc.qt.io/qt-5/qtnetwork-index.html>
Network module documentation

The end

Thank you for attention