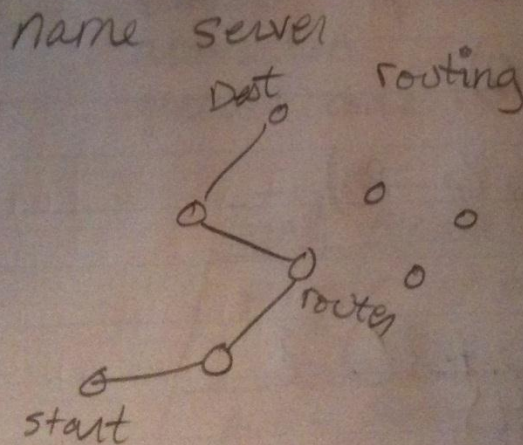


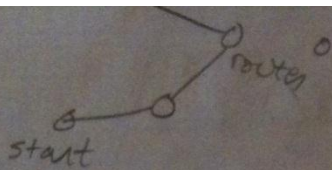
Network Programming

IP Internet Protocol  
port  
protocol

hostname  
www.purdue.edu  
ece/linux23.ecn.purdue.edu



Program



Program

```
main
QApplication app(argc, argv);
ClientSocket sock(argv[1]);
return app.exec();
```

```
class ClientSocket {
    Q_OBJECT
    QTcpSocket *socket;
    QString request();
private slot:
    void reportConnected();
    void reportNetFound();
    void getPage();
    void connectorClosed();
public:
    ClientSocket(QString name);
```

client

```
QTcpServer *S =
    new QTcpServer();
S->listen();
cout << S->serverPort() << endl;
```

```
QApplication app(argv, argc);
ClientSocket sock(argv[1]);
return app.exec();
```

```
class ClientSocket {
    Q_OBJECT
    QTcpSocket *socket;
    QString request();
private slot:
    void reportConnected();
    void reportNotFound();
    void getPage();
    void connectorClosed();
public:
    ClientSocket(string name);
    ~ClientSocket() {
        if (socket) delete socket;
    }
}; // ~
}; // class.
```

```
QTcpServer *s =
    new QTcpServer();
s->listen();
cout << s->serverPort() << endl;
```

// cont ->

cont'd

.client www.purdue.edu  
argv[1]

```
ClientSocket::ClientSocket(string s) {
    socket = new QTcpSocket(); // client
    connect(socket, SIGNAL(connected()), this, SLOT(reportConnected()));
    connect(socket, SIGNAL(readyRead()), this, SLOT(getPage()));
    QString q(s.c_str());
    socket->connectToHost(q, 80);
}
```

```
ClientSocket::reportConnected() {
    QString r = request();
    socket->write(r);
    socket->flush();
}
```

```
}  
ClientSocket::reportConnected() {  
    QString r = request(),  
    socket -> write(r),  
    socket -> flush(),  
}
```

```
QString ClientSocket::request() {  
    QString r = "Host " + "www.purdue.edu";  
    QString url(r),  
    QString s = "GET /HTTP/1.0 \r\n" + url + "\r\n\r\n";  
    return s;  
}
```

```
connect(socket, SIGNAL(readyRead()), this, SLOT(getPage()));  
void ClientSocket::getPage() {  
    int n = socket -> bytesAvailable();  
    char * data = new char[n],  
    cout << data,  
    delete [] data;  
}
```

```
socket -> write(r),  
socket -> flush(),  
}
```

```
QString ClientSocket::request() {  
    QString r = "Host " + "www.purdue.edu";  
    QString url(r),  
    QString s = "GET /HTTP/1.0 \r\n" + url + "\r\n\r\n";  
    return s;  
}
```

```
connect(socket, SIGNAL(readyRead()), this, SLOT(getPage()));  
void ClientSocket::getPage() {  
    int n = socket -> bytesAvailable();  
    char * data = new char[n],  
    cout << data,  
    delete [] data;  
}
```