```
/**
 * OpenCV video streaming over TCP/IP
 * Client: Receives video from server and display it
 * by Steve Tuenkam
 */

#include "opencv2/opencv.hpp"
#include <sys/socket.h>
#include <arpa/inet.h>
#include <unistd.h>

using namespace cv;


int main(int argc, char** argv)
{

    //--------------------------------------------------------
    //networking stuff: socket , connect
    //--------------------------------------------------------
    int         sokt;
    char*       serverIP;
    int         serverPort;

    if (argc < 3) {
           std::cerr << "Usage: cv_video_cli <serverIP> <serverPort> " << std::endl;
    }

    serverIP   = argv[1];
    serverPort = atoi(argv[2]);

    struct  sockaddr_in serverAddr;
    socklen_t           addrLen = sizeof(struct sockaddr_in);

    if ((sokt = socket(PF_INET, SOCK_STREAM, 0)) < 0) {
        std::cerr << "socket() failed" << std::endl;
    }

    serverAddr.sin_family = PF_INET;
    serverAddr.sin_addr.s_addr = inet_addr(serverIP);
    serverAddr.sin_port = htons(serverPort);

    if (connect(sokt, (sockaddr*)&serverAddr, addrLen) < 0) {
        std::cerr << "connect() failed!" << std::endl;
    }
```

```
    //-------------------------------------------------------------
    //OpenCV Code
    //-------------------------------------------------------------

    Mat img;
    img = Mat::zeros(480 , 640, CV_8UC1);
    int imgSize = img.total() * img.elemSize();
    uchar *iptr = img.data;
    int bytes = 0;
    int key;

    //make img continuos
    if ( ! img.isContinuous() ) {
          img = img.clone();
    }

    std::cout << "Image Size:" << imgSize << std::endl;


    namedWindow("CV Video Client",1);

    while (key != 'q') {

        if ((bytes = recv(sokt, iptr, imgSize , MSG_WAITALL)) == -1) {
            std::cerr << "recv failed, received bytes = " << bytes << std::endl;
        }

        cv::imshow("CV Video Client", img);

        if (key = cv::waitKey(10) >= 0) break;
    }

    close(sokt);

    return 0;
}
```