

Trasmissione/ricezione via TCP con i Socket su server remoto reale

E' il caso di provare le due applicazioni appena descritte in un contesto reale, cioè, facendo girare l'applicazione server su un PC remoto e non in locale.

Ho modificato leggermente il codice del server affinché venisse visualizzato l'indirizzo IP dell'applicazione client al momento della avvenuta trasmissione di un messaggio. Questo è il codice:

```
namespace SocketServer
{
    class Program
    {
        static void Main(string[] args)
        {
            EseguiServer();
            Console.ReadKey();
        }

        public static void EseguiServer()
        {
            Socket listener = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
            try
            {
                listener.Bind(new IPEndPoint(IPAddress.Any, 3306));
                listener.Listen(10);

                Console.WriteLine("Attesa di connessioni ... \n");
                byte[] bytes = new Byte[1024];
                string data = null;

                Socket clientSocket;
                clientSocket = listener.Accept();

                while (true)
                {
                    int numByte = clientSocket.Receive(bytes);
                    data = Encoding.ASCII.GetString(bytes, 0, numByte);

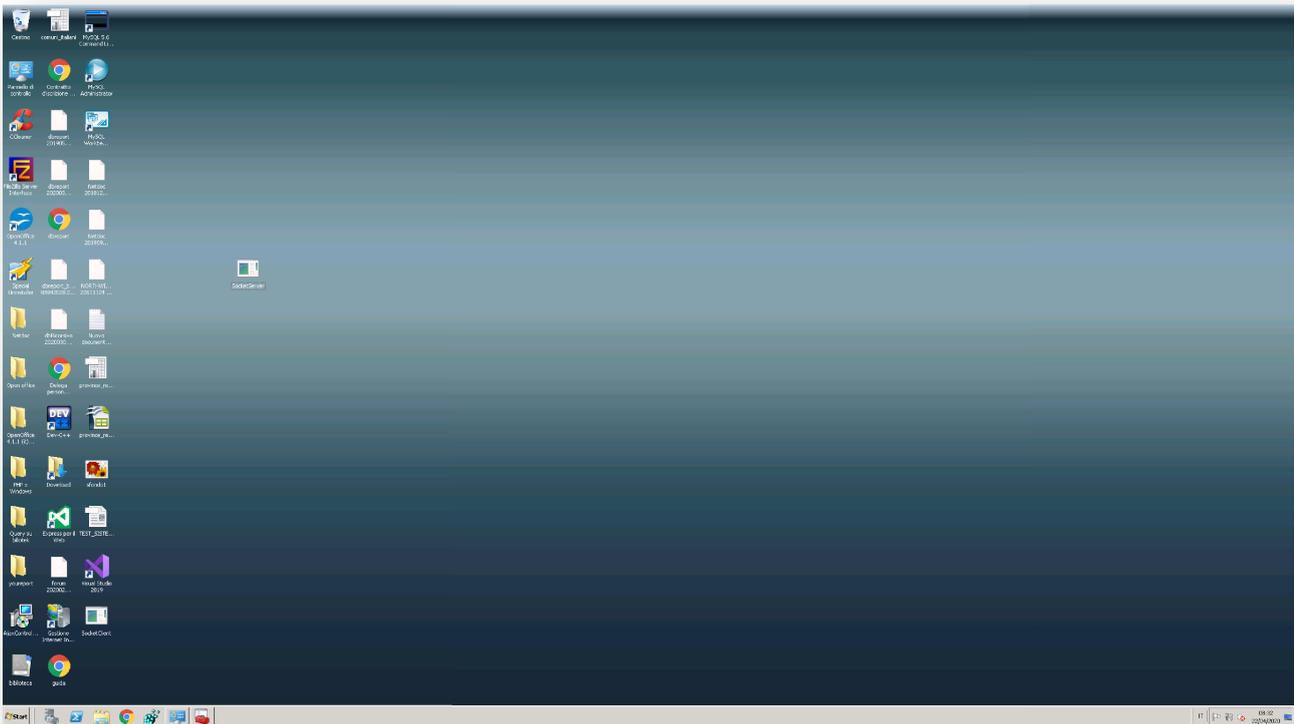
                    if (data.IndexOf("@") > -1)
                    {
                        clientSocket.Shutdown(SocketShutdown.Both);
                        clientSocket.Close();
                        listener.Close();
                        break;
                    }
                    else
                    {
                        Console.WriteLine("Testo ricevuto -> {0} da {1} \n", data,
                            IPAddress.Parse(((IPEndPoint)clientSocket.RemoteEndPoint).Address.ToString()));
                        byte[] message = Encoding.ASCII.GetBytes("OK messaggio ricevuto!!! \n");
                        clientSocket.Send(message);
                        numByte = 0;
                    }
                }
            }
            catch (Exception e)
            {
                Console.WriteLine(e.ToString());
            }
        }
    }
}
```

Questo codice verrà compilato in modalità **Release**, cioè in maniera definitiva e l'eseguibile sarà fatto girare sul server web del nostro laboratorio.

Ma prima di tutto bisogna mettere le mani sul router della scuola affinché possa inoltrare i messaggi provenienti dal client al server.

Questa procedura è necessaria perché il router non è a conoscenza della porta di ascolto dell'applicazione server e quindi è necessario 'informarlo' che tutti quei messaggi indirizzati al proprio indirizzo IP pubblico (79.7.41.27) e sulla porta '3306', devono essere inoltrati all'host (server web con indirizzo IP, interno alla rete dell'Istituto, 172.16.1.181) su cui gira l'applicazione server che resta in attesa di connessioni da parte del client su quella porta.

Per accedere alle impostazioni del router della scuola, mi sono connesso via Desktop Remoto al server web del laboratorio, eccolo:

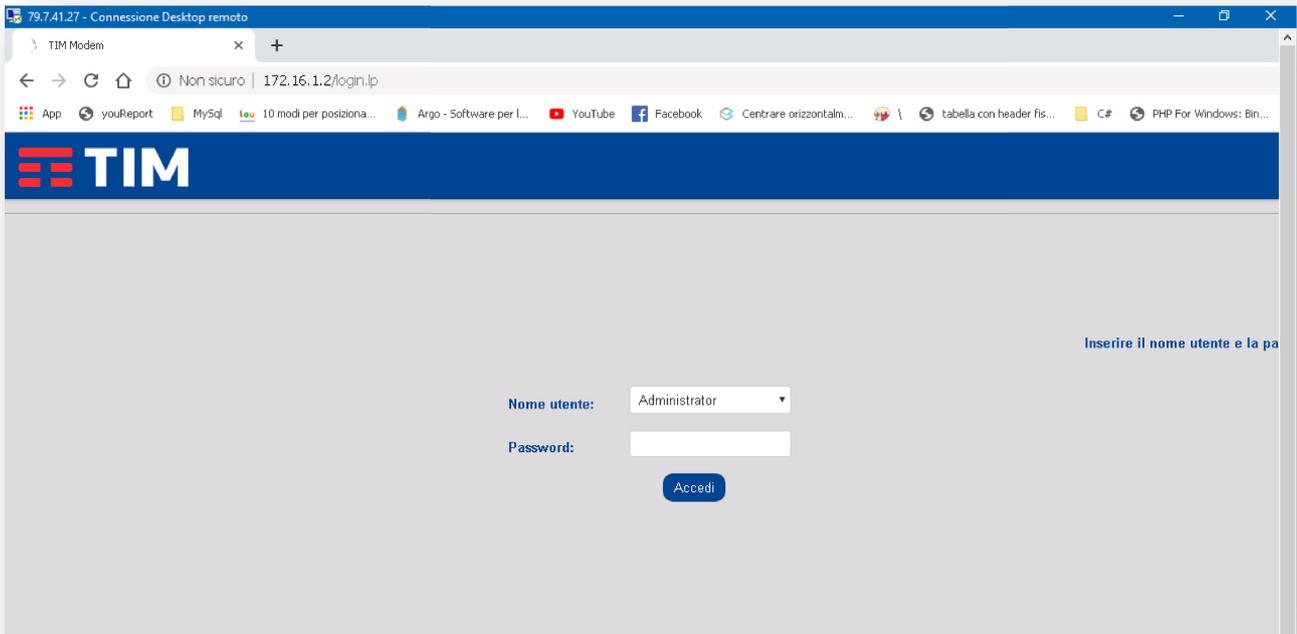


Poi ho lanciato il browser per connettermi al router che ha come indirizzo IP

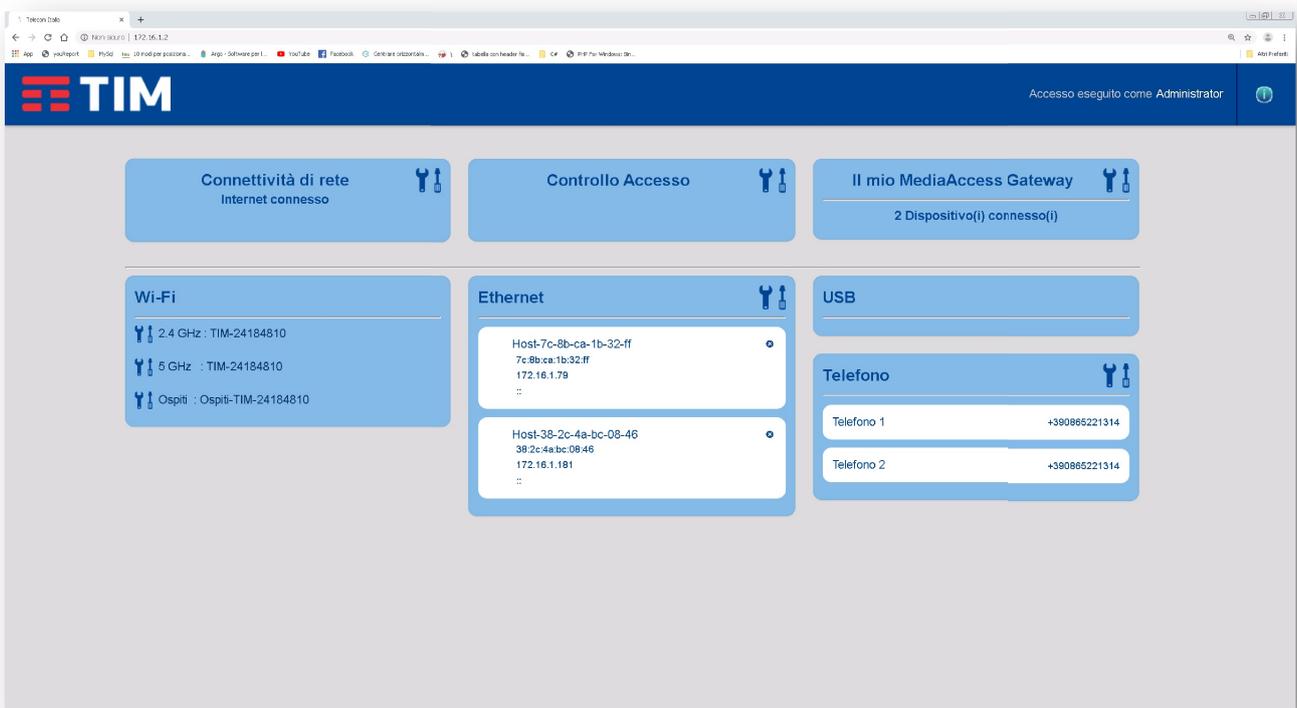
172.16.1.2

che rappresenta il **'GATEWAY'** impostato su tutte le schede di rete che accedono ad internet tramite questo router.

In figura l'accesso:



Dopo aver digitato la password di accesso, sono dentro:



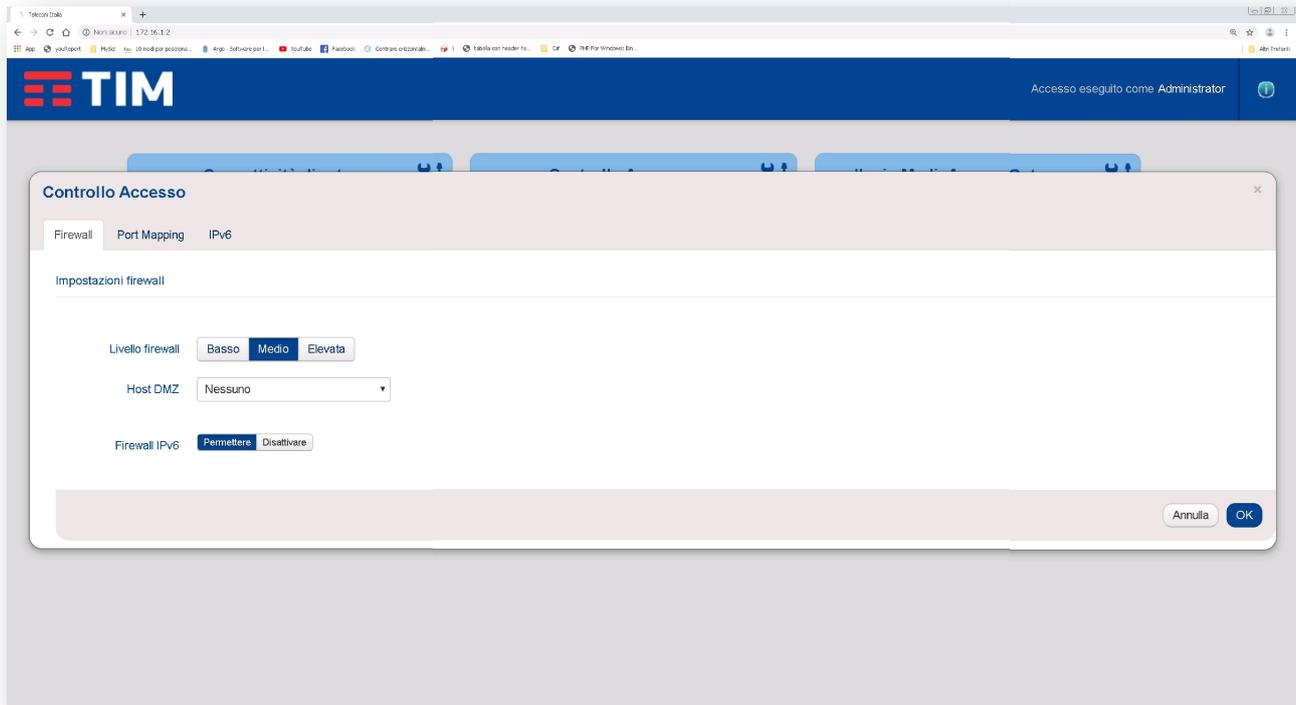
Questa è la schermata dove posso modificare tutte le impostazioni. Ma quella che ci interessa, riguarda il riquadro

“Controllo Accesso”

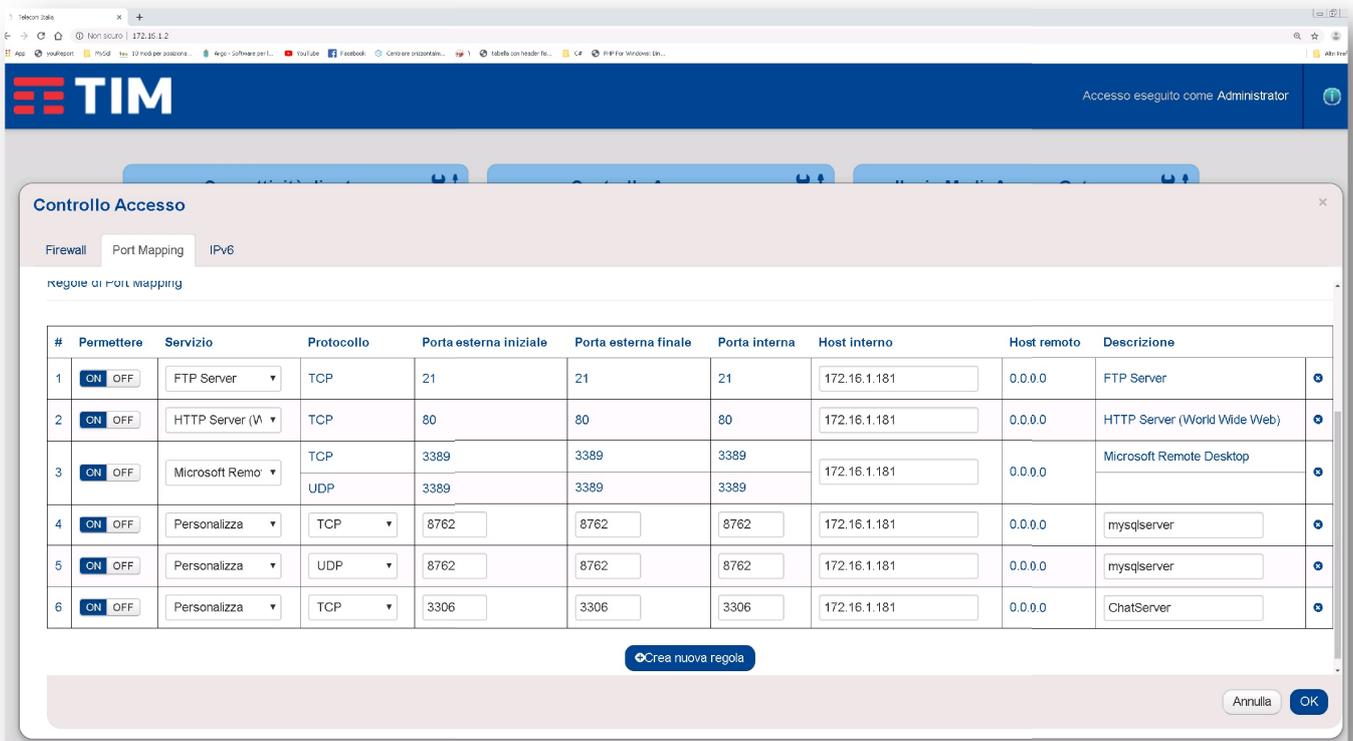
ed è lì che ho la possibilità di effettuare le operazioni di cui ho bisogno.

T.P.S.I.T.- Trasmissione di messaggi in rete tramite protocollo TCP e UDP in C#

Clicco sul riquadro ed entro:



Nella schermata, si notano tre schede: FIREWALL, Port Mapping e IPv6, a noi interessa quella del “Port Mapping”, cioè la mappatura delle porte del router, eccola:



In questa schermata si notano tutte le porte aperte sul router e tutte quante sono riferite all'host

172.16.1.181

che è il server web del laboratorio. Si notano due 'well known port', la **21** e **80** che si riferiscono rispettivamente al protocollo **FTP**⁶ e **http**⁷.

Poi ci sono delle porte, a parte quella dedicata al 'Desktop Remoto di Microsoft' che tutti i router hanno per default, ci sono quelle che ho impostato per il server MySQL sotto la Descrizione 'mysqlserver' nel Servizio 'Personalizza' con protocollo TCP e UDP, per l'accesso in remoto al server di database.

L'ultima porta è quella relativa al socket server che resta in ascolto sulla porta '3306' con la Descrizione 'ChatServer' con protocollo TCP.

La procedura di impostazione di tutte queste porte sul router viene chiamata

NAT⁸

Network Address Iranslation

Nell'ambito delle reti informatiche, il *network address translation* (abbreviato in NAT e traducibile in italiano con "traduzione degli indirizzi di rete") è un meccanismo che permette di modificare l'indirizzo IP dei pacchetti in transito attraverso apparati di rete – come router o firewall – all'interno di una comunicazione in corso tra due o più nodi della rete.

L'operazione di NAT è disponibile anche su molti router casalinghi, ma non tutti. Purtroppo alcune aziende telefoniche come Fastweb, Vodafone, Linkem ed altre, non consentono l'accesso al router e per abilitare delle porte bisogna farlo tramite un operatore dell'azienda che può connettersi in remoto allo stesso router.

Dette e fatte tutte queste operazioni, ho copiato la versione '*Release*' del server socket sul server di laboratorio.

Purtroppo però non basta al corretto funzionamento dell'applicazione. Infatti il sistema operativo per comunicare con il nostro server socket, fa uso anch'esso di porte e queste porte sono casuali cioè vengono create a runtime ogni volta che viene lanciata l'applicazione. E qui che entra in gioco il Firewall il quale nega la comunicazione su queste porte. Quindi è necessario aprire il Firewall e abilitare tutte le porte con una nuova regola. Ovviamente terminata l'esperienza si dovrà eliminare questa regola per una questione di sicurezza.

Detto questo, proviamo a trasmettere un messaggio.

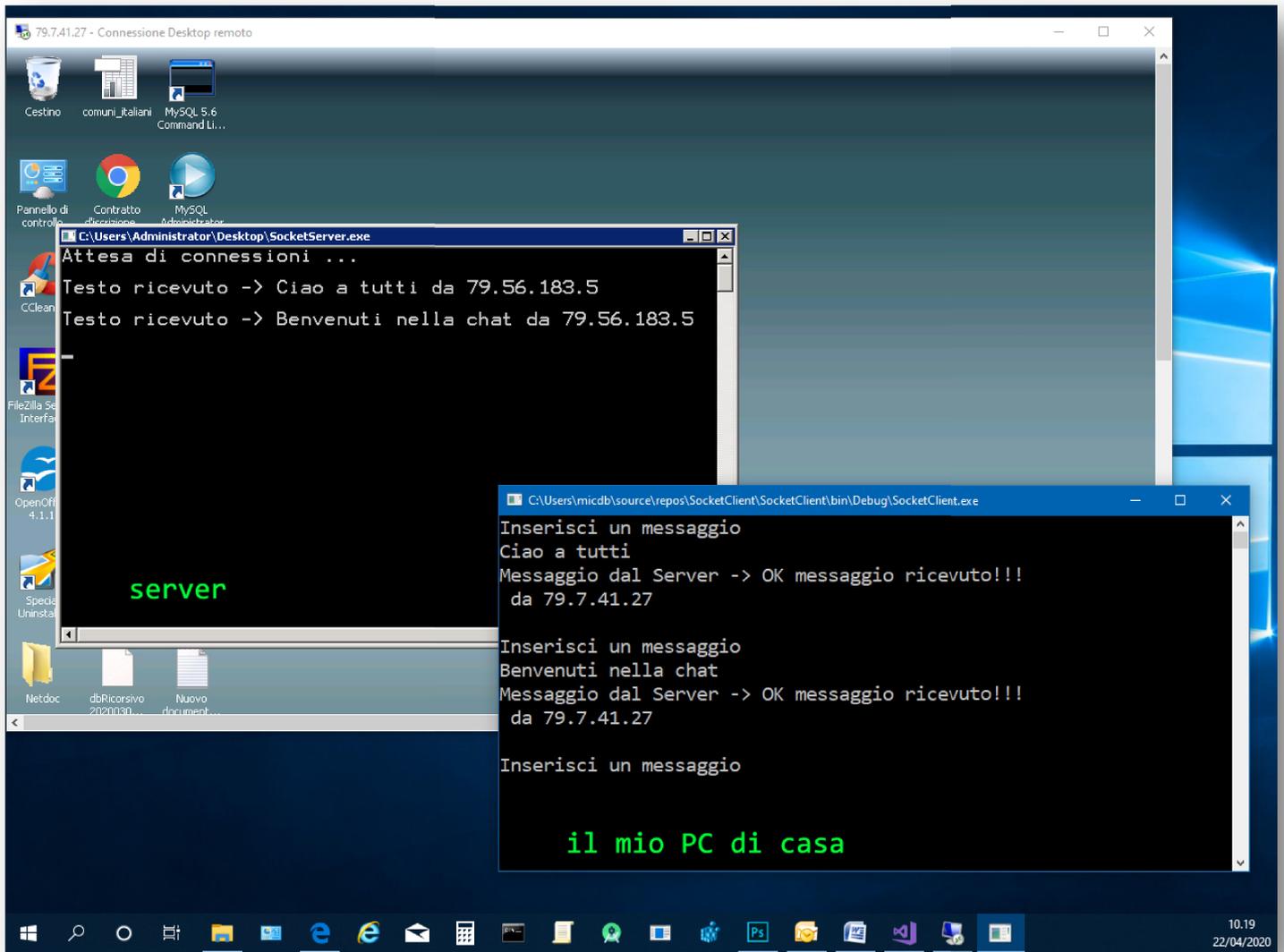
Per prima cosa bisogna che il server socket sia avviato per primo, quindi, sempre tramite Desktop Remoto, lancio il server socket che resterà in ascolto sulla porta nattata in precedenza. Poi sul mio PC lancio il socket client ed incomincio a trasmettere alcuni messaggi.

In figura una prova di esecuzione:

⁶ <https://www.computersec.it/2019/03/28/protocollo-ftp/>

⁷ https://it.wikipedia.org/wiki/Hypertext_Transfer_Protocol

⁸ https://it.wikipedia.org/wiki/Network_address_translation



Come potete notare la ricezione di messaggi da parte del socket server funziona e al messaggio ricevuto viene visualizzato l'IP del mittente che, in questo caso, è l'indirizzo pubblico del mio router **79.56.183.5** che è diverso da quello che avevo nei giorni precedenti in quanto esso è dinamico e cambia ogni volta che lo accendo.

Mentre il socket client che è stato eseguito sul mio PC, ad ogni messaggio trasmesso riceve la 'notifica' della ricezione e l'IP del router della scuola che è appunto il **79.7.41.27**.