

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**  
**Федеральное государственное бюджетное образовательное учреждение**  
**высшего образования**  
**«Рязанский государственный радиотехнический университет имени В.Ф.**  
**Уткина»**

**Кафедра САПР ВС**

**Отчет о лабораторной работе №1:**  
**«Потоковые сокет»**

Выполнили:

ст. гр.846

Степура М. Н.

Немцев А. Е.

Проверил:

Шибанов В. А.

Псоянц В. Г.

Рязань 2021

**Цель работы:** Знакомство с работой потоковых транспортных протоколов.

## Практическая часть.

### Серверная часть кода:

```
using (Socket socket = new Socket(AddressFamily.InterNetwork, SocketType.Stream,
ProtocolType.Tcp))
{
    try
    {
        socket.Bind(ip);
        socket.Listen(10);
        control.PrintLine("Сервер включен...");
    }
    catch (Exception ex)
    {
        control.PrintLine(ex.Message);
    }

    bool isExit = true;

    while (true)
    {
        var handler = socket.Accept();

        if (isExit)
        {
            control.PrintLine(handler.LocalEndPoint.ToString() + " (" +
DateTime.Now.ToShortTimeString() + ") - Подключился");

            isExit = !isExit;
        }

        var builder = new StringBuilder();
        byte[] data = new byte[256];

        try
        {
            do
            {
                int bytes = handler.Receive(data);
                builder.Append(Encoding.Unicode.GetString(data, 0, bytes));
            }
            while (handler.Available > 0);

            if (builder.ToString().ToUpper() == "QUIT" ||
builder.ToString().ToUpper() == "ВЫХОД")
            {
                control.PrintLine(handler.LocalEndPoint.ToString() + " (" +
DateTime.Now.ToShortTimeString() + ") - Отключился");

                isExit = !isExit;
                handler.Shutdown(SocketShutdown.Both);
                handler.Close();
            }
            else
            {
                control.PrintLine(handler.LocalEndPoint.ToString() + " (" +
DateTime.Now.ToShortTimeString() + "): " + builder.ToString());
            }
        }
    }
}
```

```

        string message = "Ваше сообщение доставлено";
        data = Encoding.Unicode.GetBytes(message);
        handler.Send(data);
    }
}
catch (Exception ex)
{
    control.PrintLine(ex.Message);
    handler.Shutdown(SocketShutdown.Both);
    handler.Close();
    isExit = !isExit;
}
}
}

```

## Клиентская часть кода:

```

if (!int.TryParse(ConfigurationManager.AppSettings["ServerPort"], out var port))
{
    Console.WriteLine("Неверный Port");
}
else
{
    var ip = new
IPEndPoint(IPAddress.Parse(ConfigurationManager.AppSettings["ServerIP"]), port);
    while (true)
    {
        using (var socket = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp))
        {
            try
            {
                socket.Connect(ip);

                Console.Write("Введите сообщение: ");
                string message = Console.ReadLine();

                byte[] data = Encoding.Unicode.GetBytes(message);
                socket.Send(data);

                if (message.ToUpper() == "QUIT" || message.ToUpper() == "ВЫХОД")
                {
                    socket.Shutdown(SocketShutdown.Both);
                    socket.Close();
                    break;
                }

                data = new byte[256];
                var builder = new StringBuilder();

                do
                {
                    int bytes = socket.Receive(data, data.Length, 0);
                    builder.Append(Encoding.Unicode.GetString(data, 0, bytes));
                }
                while (socket.Available > 0);

                Console.WriteLine("Ответ сервера: " + builder.ToString());
            }

```

```
        catch (Exception ex)
        {
            Console.WriteLine(ex.Message);
            continue;
        }
    }
}
```

**Вывод:** Мы познакомились с работой потоковых транспортных протоколов.