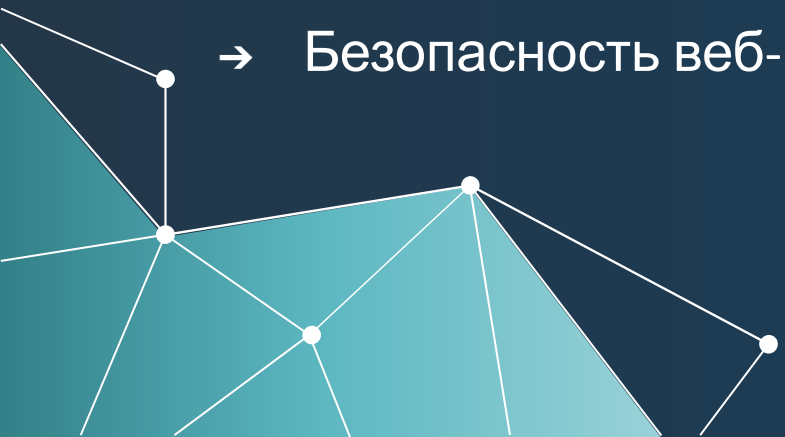
Server Side Programming

Ilya Loshkarev 2024

Серверная разработка ПО

- Основы клиент-серверного взаимодействия
- Разработка веб-приложений
- Разработка веб-API
- Веб-приложения реального времени
- Безопасность веб-приложений



Экзамен

Лабораторные работы - 60 баллов

Индивидуальные задания - 40 баллов

Экзамен - 40 баллов

https://edu.mmcs.sfedu.ru/course/view.php?id=555

Программные средства

dotNET Core, C#

ASP.NET Core 6

Visual Studio 2022

Docker Compose

Основы клиент-серверного взаимодействия

- Сетевая модель OSI
- Уровни стека TCP/IP
- Протокол IP, UDP, TCP
- Протокол HTTP +HTTPS
- Структура HTML документа, порядок работы браузера для отображения страниц
- System.Net, пример элементарного HTTP-сервера

Сетевая модель OSI

На каждом уровне N два объекта обмениваются блоками данных (PDU) с помощью протокола данного уровня на соответствующих устройствах.

Каждый PDU содержит блок служебных данных (SDU), связанный с верхним или нижним протоколом.

Application	HTTP, FTP, POP3, SMTP, WebSocket
Presentation	ASCII, EBCDIC, SSL, gzip
Session	RPC, PAP, L2TP, gRPC
Transport	TCP, UDP, SCTP, Порты
Network	IPv4, IPv6, IPsec, ICMP
Data Link	PPP, IEEE 802.22, Ethernet, DSL, ARP
Physical	Радиоволны, оптоволокно, витая пара и т.д.

Уровни стека TCP/IP

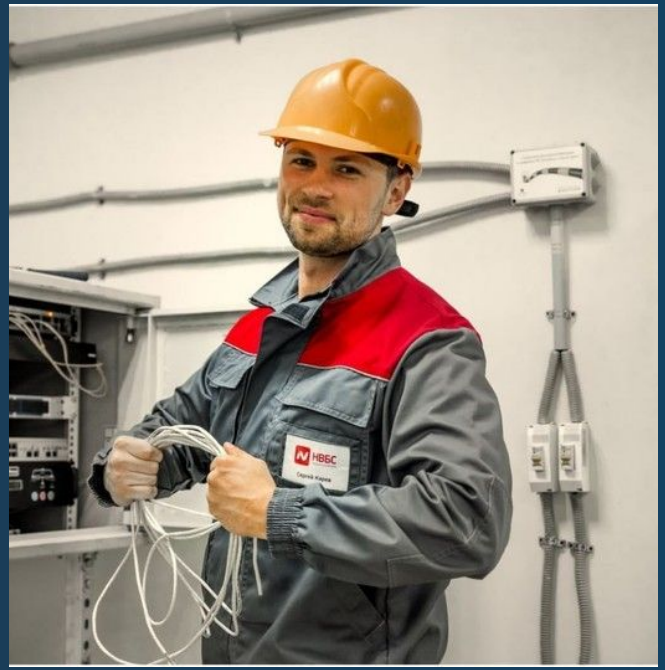
Параллельная система разделения протоколов появившаяся в процессе построения системы Internet в 1970-х

Модели OSI и TCP/IP являются формальной частью стандартов IETF

Application	HTTP, FTP, POP3, SMTP, WebSocket
	ASCII, EBCDIC, SSL, gzip
	RPC, PAP, L2TP, gRPC
Transport	TCP, UDP, SCTP, Порты
Internet	IPv4, IPv6, IPsec, ICMP, ARP?
Link	PPP, IEEE 802.22, Ethernet, DSL, ARP?
	Радиоволны, оптоволокно, витая пара и т.д.

Физический и Канальный Уровни

- Не рассматриваются в рамках данного курса
- Важно учитывать для обеспечения безопасности сети
- Многие уязвимости находятся именно в этих протоколах

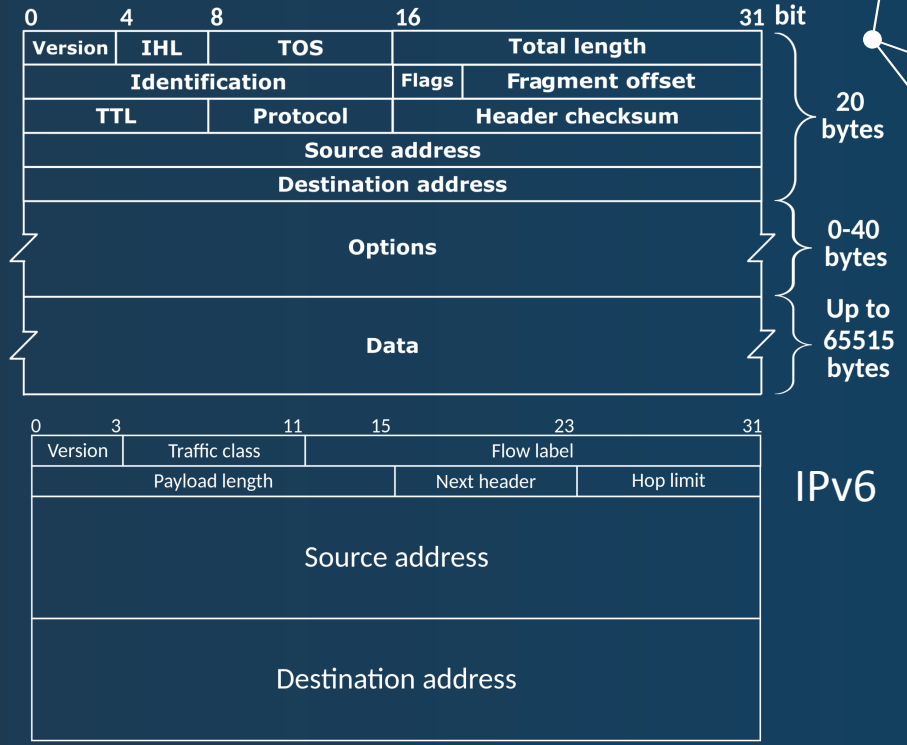


Сетевой протокол IP

Протокол описан в RFC 791 (сентябрь 1981 года)

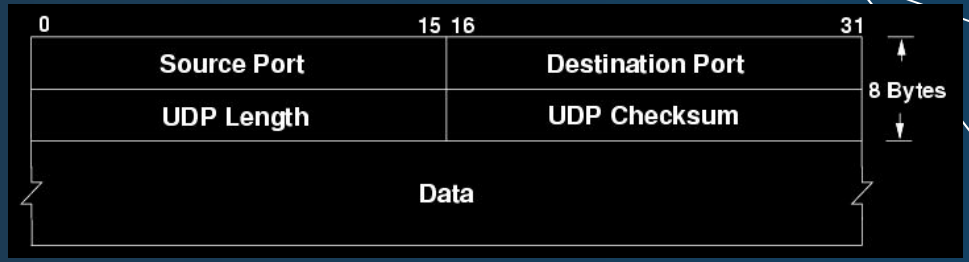
Основная идея -- разделение адреса хостов на “подсети” различных “классов”

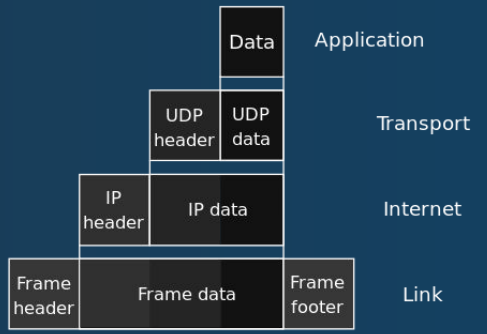
С 1993 применятся бесклассовая система CIDR (RFC 1519)



Транспортный протокол UDP

- Протокол прямой передачи данных
- Является не надежным, но очень быстрым
- Используется для работы интернет систем (DHCP, DNS) и систем реального времени (стриминг, многопользовательские игры)





Транспортный протокол TCP

Протокол с установкой и контролем соединения

Имеет встроенные механизмы контроля ошибок передачи, скорости обработки пакетов и заторов сети

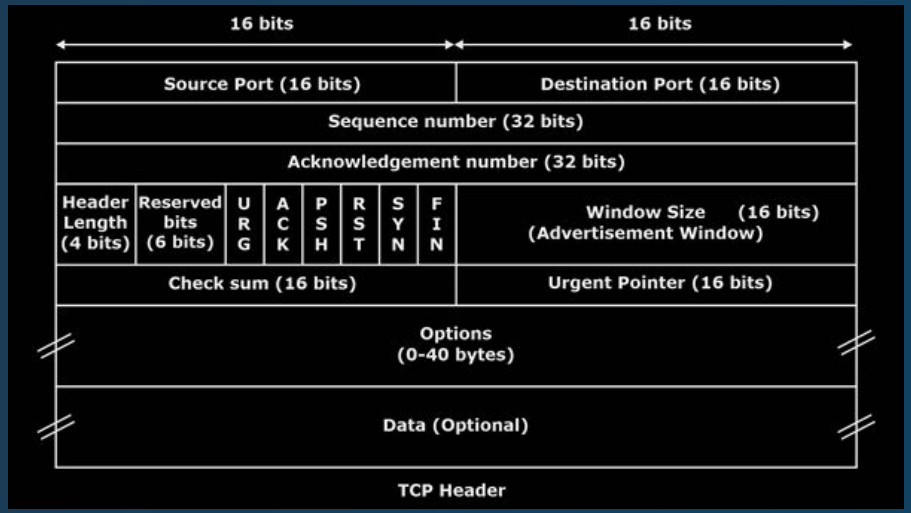
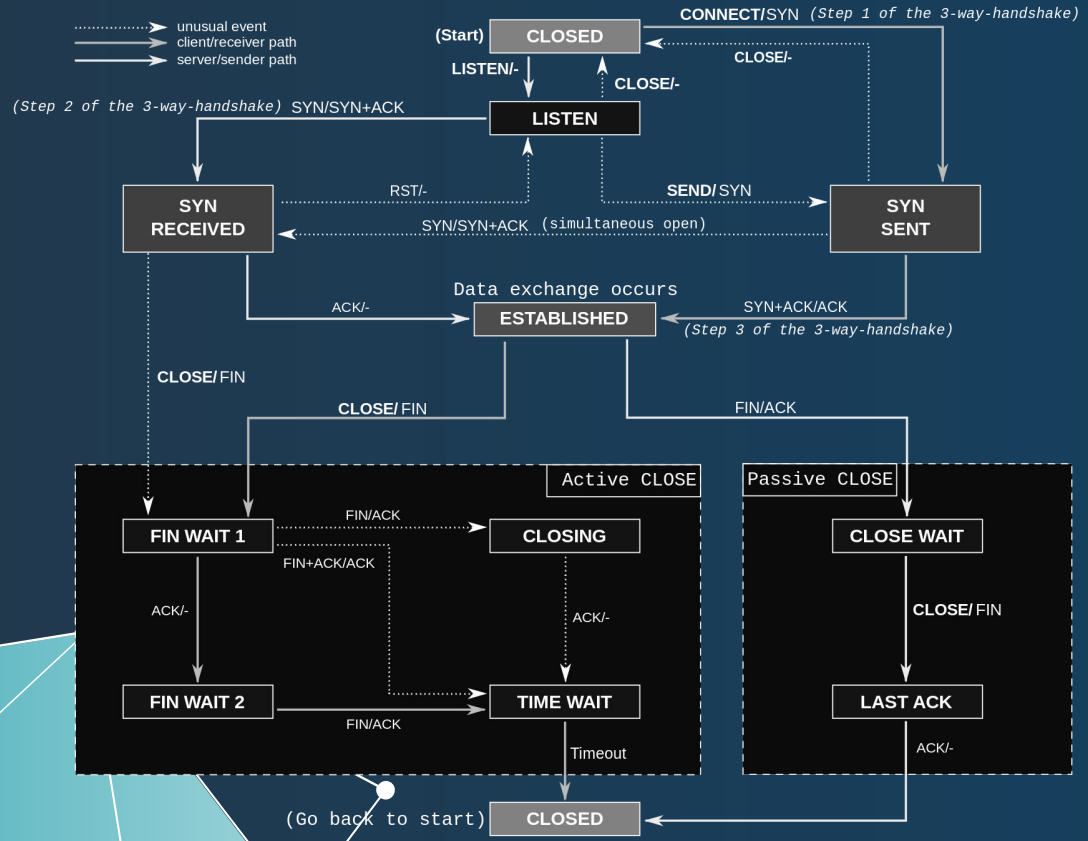
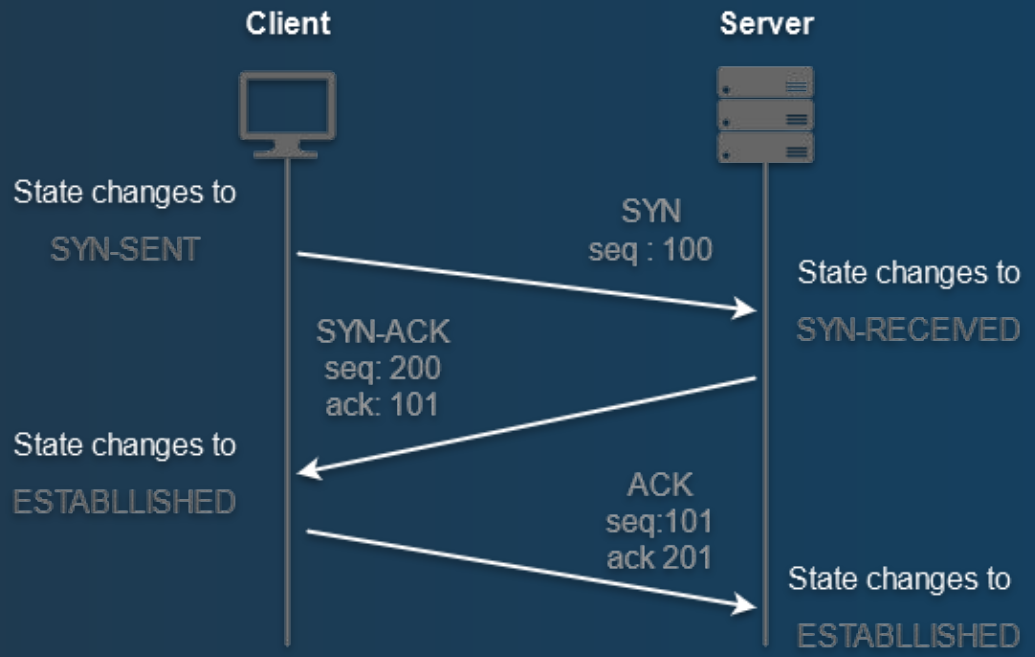


Диаграмма состояний TCP соединения



TCP Handshake

Установка TCP соединения



Протокол HTTP

Текстовый протокол для получения с серверов гипертекстовых документов в формате HTML

Стал универсальным средством взаимодействия между узлами Всемирной паутины

```
GET / HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0
Accept: text/html
Accept-Language: en-GB
Accept-Encoding: gzip
Connection: keep-alive

...
```

Коды HTTP

1XX (informational) -- Запрос получен, продолжаем работу

2XX (successful) -- Запрос был успешно обработан

3XX (redirection) -- Дополнительные действия нужны для обработки запроса

4XX (client error) -- Запрос некорректен и не может быть обработан

5XX (server error) -- Сервер не смог выполнить запрос из-за внутренней ошибки

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005
Content-Type: text/html
Content-Length: 155
Accept-Ranges: bytes
Connection: close

<html>
...
```

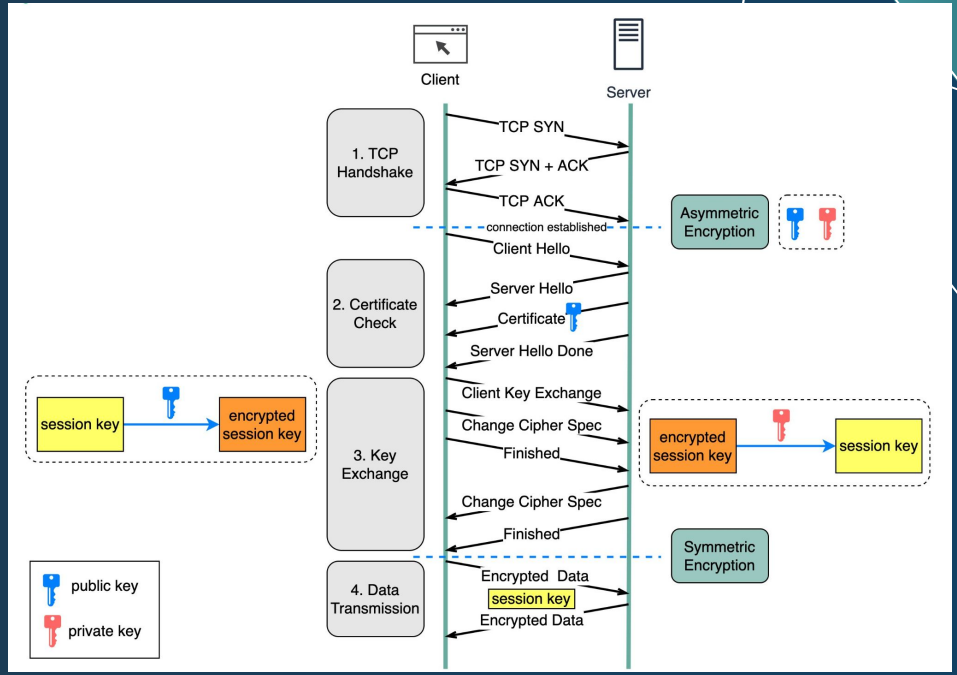
HTTPS, SSL и TLS

HTTPS и SSL разработан Netscape

Работает как обертка-туннель над сообщениями HTTP в HTTPS

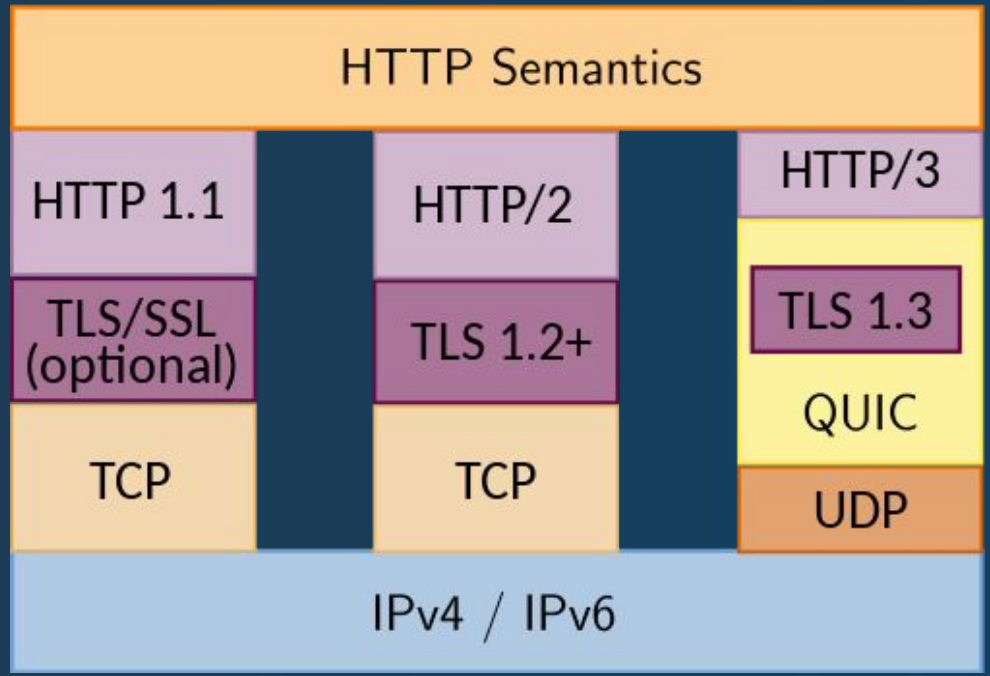
Протокол принят IETF

Переименован в TLS в 1999 году



HTTP /1.1 /2 /3

- Ускорение на уровне протокола, сжатие метаданных
- Возможность переключения протокола после установки соединения
- Возможность обработки нескольких запросов в рамках одного соединения
- Пуш данных с сервера



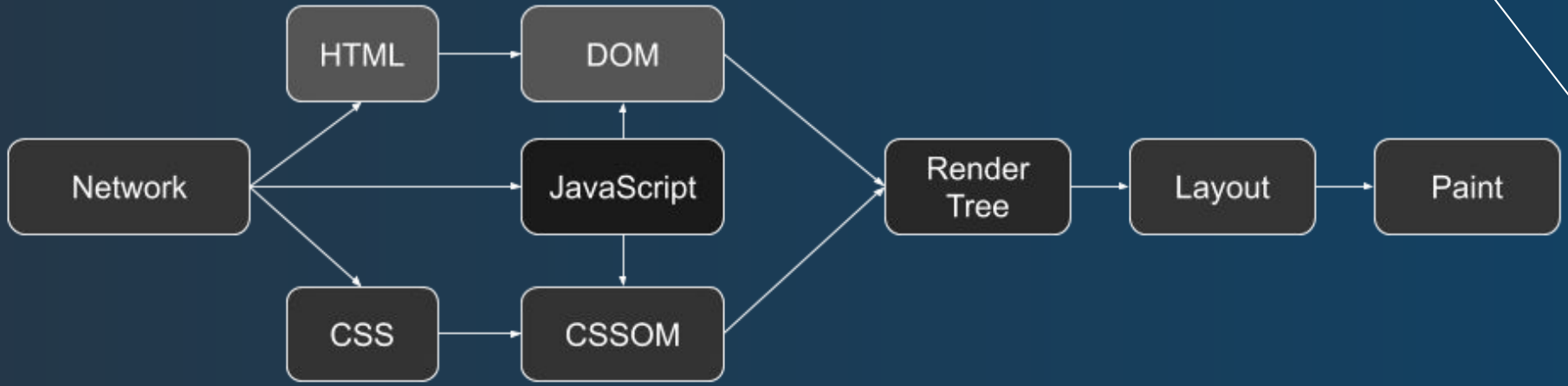
Структура HTML документа

```
<!doctype html>
<html>
  <head>
    <title>Hello</title>
  </head>
  <body>
    <p>Welcome to this example.</p>
  </body>
</html>
```

Tim Berners-Lee

https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML

Браузер, загрузка страниц



System.Net

Содержит основные классы dotNet для сетевого взаимодействия

- HttpClient, HttpListener
- TcpClient, TcpListener
- UdpClient, Dns

https://learn.microsoft.com/ru-ru/dotnet/framework/network-programming/best-practices-for-system-net-classes

Элементарный HTTP-сервер

```
HttpListener listener = new HttpListener();
listener.Prefixes.Add("127.0.0.1");
listener.Start();
Console.WriteLine("Listening...");
while (listener.IsListening) {
    HttpListenerContext context = listener.GetContext();
    HttpListenerRequest request = context.Request;
    HttpListenerResponse response = context.Response;
    string responseString = "<HTML><BODY> Hello world!</BODY></HTML>";
    byte[] buffer = System.Text.Encoding.UTF8.GetBytes(responseString);
    response.ContentLength64 = buffer.Length;
    System.IO.Stream output = response.OutputStream;
    output.Write(buffer, 0, buffer.Length);
    output.Close();
}
listener.Stop();
```