MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA
CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES – CÂMPUS SÃO JOSÉ

RELATÓRIO TÉCNICO

LABORATÓRIO WIRESHARK E TCPDUMP (PARTE 1 E 2)

Arthur Cadore Matuella Barcella

TAREFA:

Desvendando o HTTP com Wireshark

Fonte base: Wireshark - HTTP

Objetivos

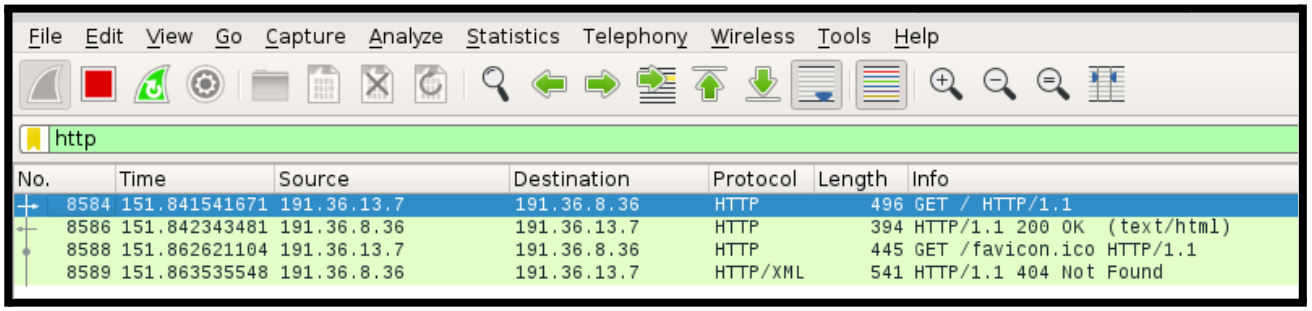
- Baseado na pequena introdução ao Wireshark estamos prontos para utilizar o mesmo para investigar protocolos em operação.
- Explorar vários aspectos do protocolo HTTP:
    1. A interação básica GET/resposta do HTTP.
    2. A interação manual GET/resposta do HTTP utilizando o telnet.
    3. Diferenciação do comportamento das versões 1.0 e 1.1 do protocolo HTTP.

A Interação Básica GET/Resposta do HTTP

1. Vamos iniciar a nossa exploração do HTTP baixando um arquivo em HTML simples - bastante pequeno, que não contém objetos incluídos. Faça o seguinte:
    1. inicie o navegador;
    2. limpe o cache do mesmo (teclas de atalho para o Google Chrome: **Ctrl + Shift + Del**);
    3. inicie o Wireshark, como descrito no **Ferramentas básicas**;
    4. inicie a captura de pacotes;
    5. digite o seguinte URL no navegador http://redes.sj.ifsc.edu.br/;
    6. pare a captura de pacotes;
    7. digite "http" (somente as letras, sem as aspas) na caixa de texto de especificação do filtro de exibição, de tal forma que apenas as mensagens HTTP capturadas serão exibidas na janela de listagem de pacotes. (Só estamos interessados em HTTP desta vez, e não desejamos ver todos os pacotes

05/09/2022

DESENVOLVIMENTO



1. **O seu navegador executa HTTP 1.0 ou 1.1?**

   Está utilizando HTTP 1.1

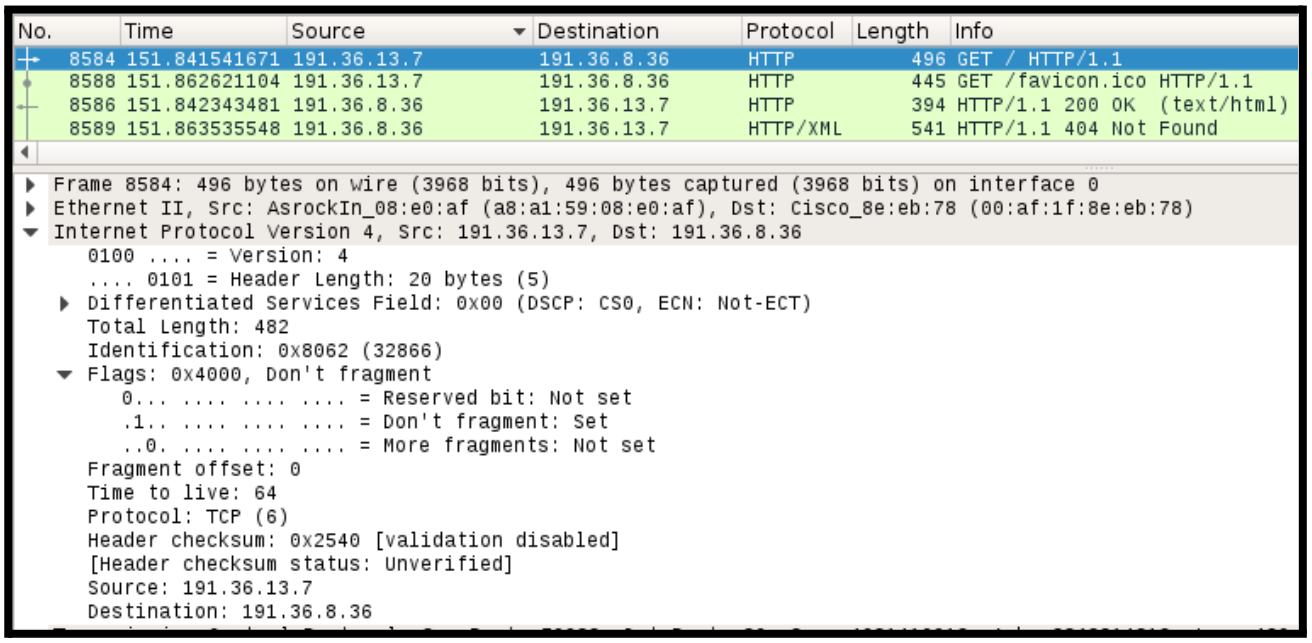
2. **Qual a versão de HTTP do servidor?**

   A resposta do servidor também se encontra em HTTP 1.1

3. **Quais idiomas (se algum) o seu navegador indica ao servidor que pode aceitar?**

No cabeçalho HTTP <u>apenas de solicitação</u> o parâmetro "Accept-Language" é responsável pela informação. Neste caso está indicando PT-BR (português Brasil).

4. Qual o endereço IP do seu computador?

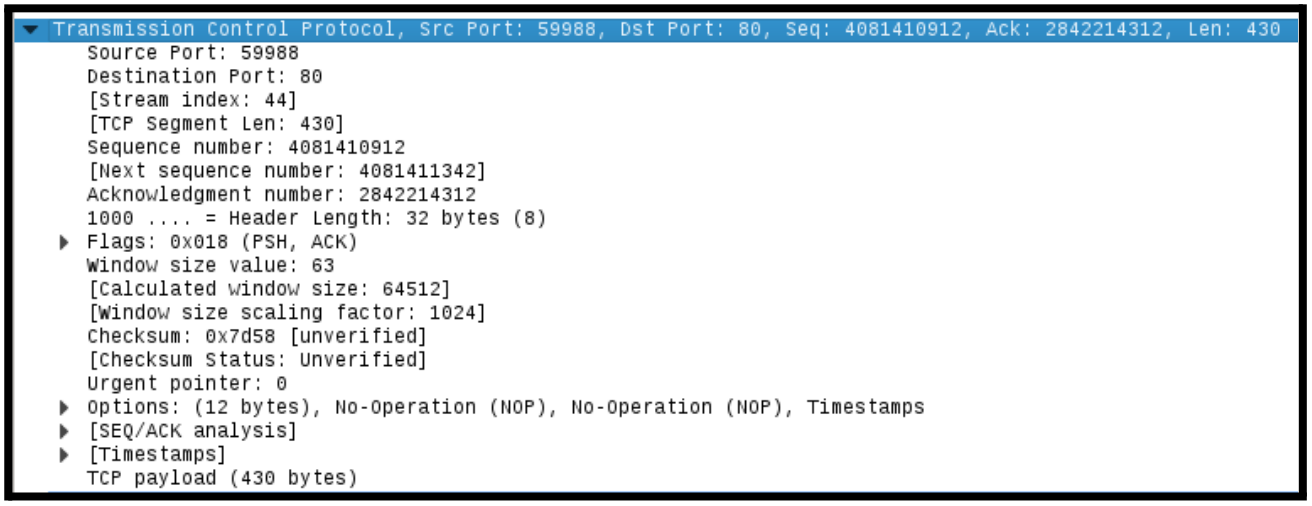


O meu computador (solicitante), é o endereço IP de origem no pacote selecionado acima, portanto: 191.36.13.7

5. E do servidor redes.sj.ifsc.edu.br?

Na mesma captura exibida acima, o endereço IP de destino é o do servidor, portanto 191.36.8.36.

6. Qual o número da porta utilizada no seu computador?

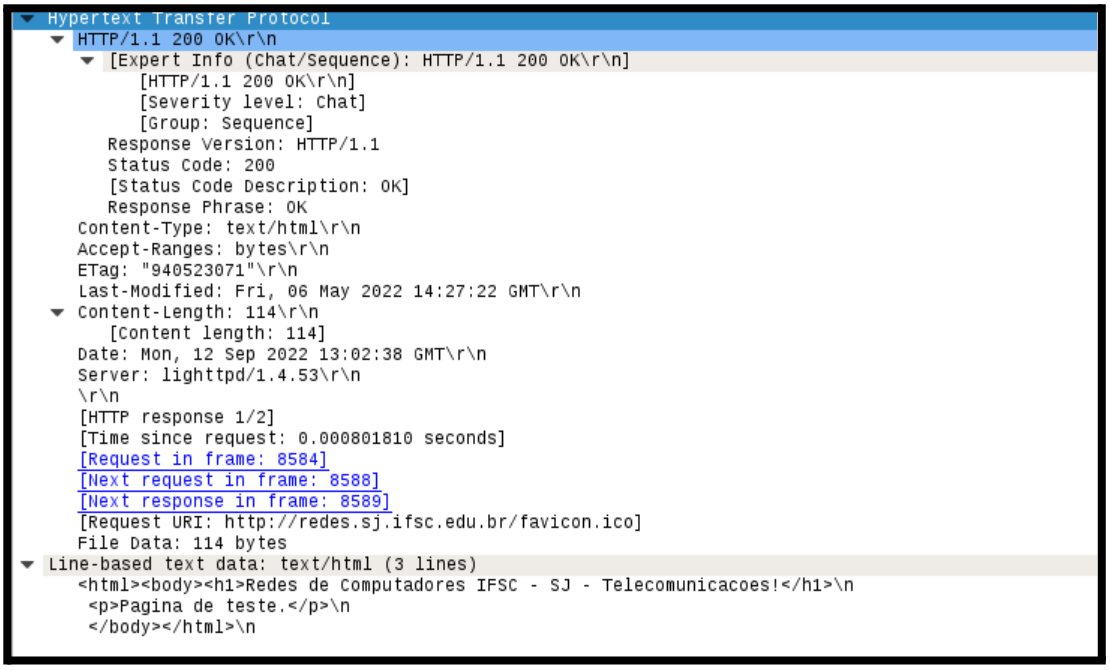


A porta de origem (da máquina que estou utilizando) é: 59988

7. **E do servidor redes.sj.ifsc.edu.br?**

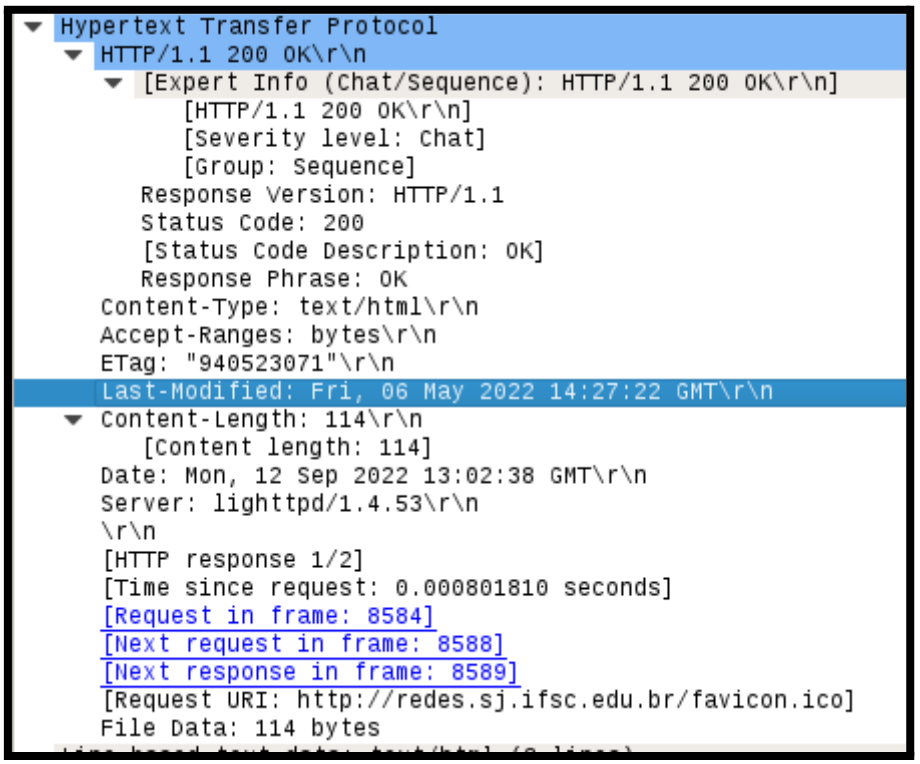
A porta de destino (do servidor) é: 80

8. **Qual o código de status retornado do servidor para o seu navegador?**



Código de status (recebido na resposta pelo servidor): 200

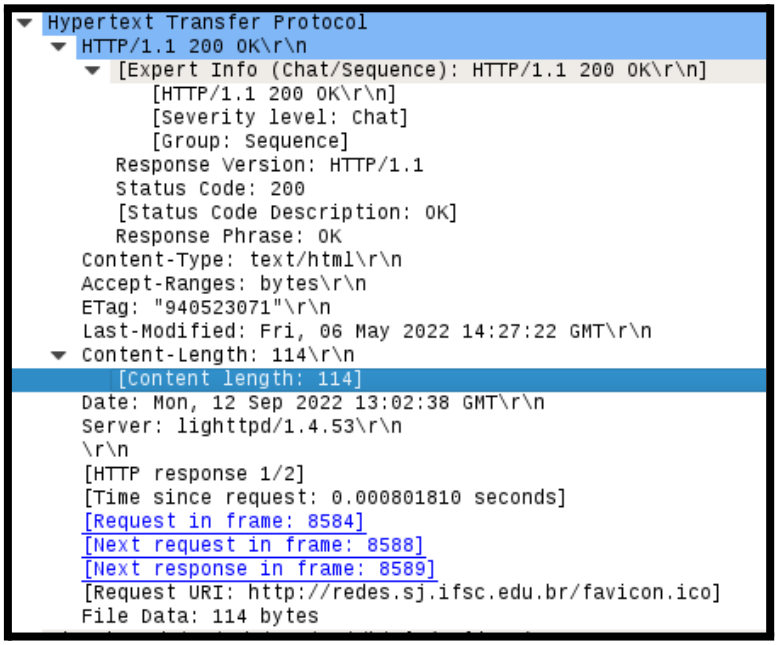
9. **Quando o arquivo em HTML que você baixou foi modificado no servidor pela última vez?**



Última modificação na sexta, 06 de Maio de 2022 às 14:27:22.

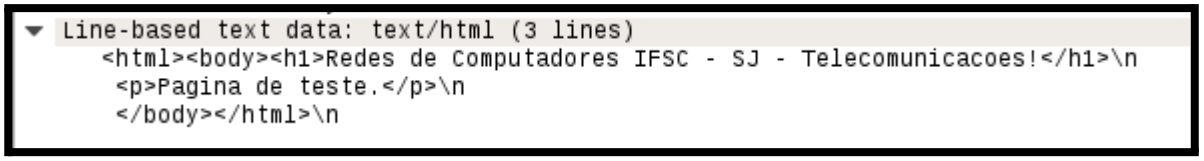
10. **Quantos bytes de conteúdo são baixados pelo seu navegador?**

114 bytes foram enviados (dentro do pacote HTTP), retirando os outros cabeçalhos usados para transmitir o pacote pela rede:



11. **Encontre a mensagem Redes de Computadores IFSC - SJ - Telecomunicacoes! - Página de teste. Onde (em qual campo) encontra-se?**

    Encontra-se dentro do body do pacote HTTP:



12. **Qual a diferença entre os endereços IP e porta de origem e destino entre a mensagem GET e a de resposta do HTTP?**

    Os endereços são opostos, pois o pacote enviado pela máquina tinha os seus endereços nos campos de origem, enquanto que na resposta, estes valores são colocados no destino do pacote.

---

1. **Identifique a página html que foi enviada como resposta. Respeita o protocolo HTTP?**

```
aluno: ~$ telnet -4 redes.sj.ifsc.edu.br 80
Trying 191.36.8.36...
Connected to redes.sj.ifsc.edu.br.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.0 200 OK
Content-Type: text/html
Accept-Ranges: bytes
ETag: "940523071"
Last-Modified: Fri, 06 May 2022 14:27:22 GMT
Content-Length: 114
Connection: close
Date: Mon, 12 Sep 2022 13:31:56 GMT
Server: lighttpd/1.4.53

<html><body><h1>Redes de Computadores IFSC - SJ - Telecomunicacoes!</h1>
 <p>Pagina de teste.</p>
 </body></html>
Connection closed by foreign host.
```

Respeita o cabeçalho visto na teoria.

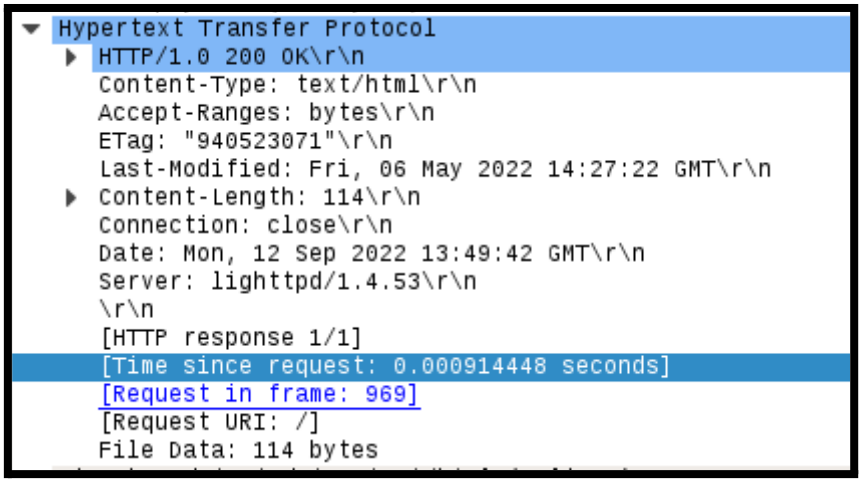
2. No Wireshark compare o resultado das execuções desses comandos com o que se viu nas capturas Wireshark com acesso pelo navegador. Qual a diferença em cada caso?

```
aluno: ~$ telnet -4 redes.sj.ifsc.edu.br 80
Trying 191.36.8.36...
Connected to redes.sj.ifsc.edu.br.
Escape character is '^]'.
GET / HTTP/1.0
```

```
▾ Hypertext Transfer Protocol
  ▾ GET / HTTP/1.1\r\n
    ▸ [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /
      Request Version: HTTP/1.1
    Host: redes.sj.ifsc.edu.br\r\n
    Connection: keep-alive\r\n
    Pragma: no-cache\r\n
    Cache-Control: no-cache\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.5060.134 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: pt-BR,pt;q=0.9\r\n
    \r\n
    [Full request URI: http://redes.sj.ifsc.edu.br/]
    [HTTP request 1/18]
    [Response in frame: 18630]
    [Next request in frame: 18641]
```

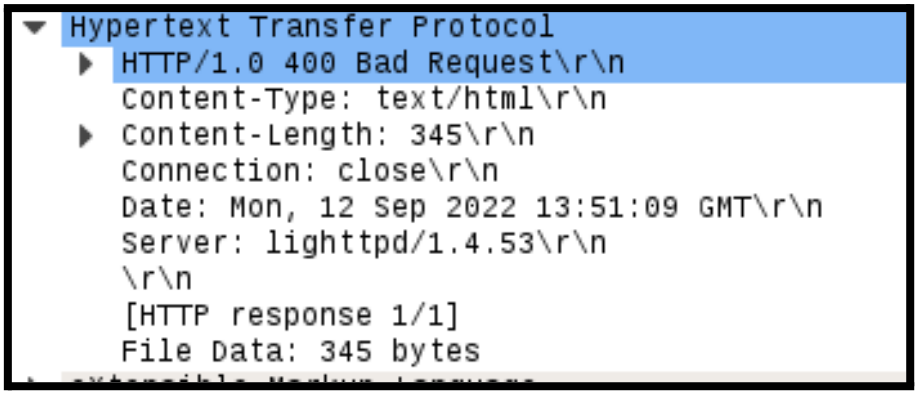
3. **Quanto tempo levou para fechar a conexão (após o duplo Enter)?**

O valor do tempo de conexão está selecionado na imagem abaixo:



4. **Refaça um pedido em que o recurso é inexistente no servidor (ex: página html com nome/URL inexistente). Observe a resposta. Qual é o código da mensagem recebida?**

Neste caso o código da mensagem é 400, ou “bad request”:

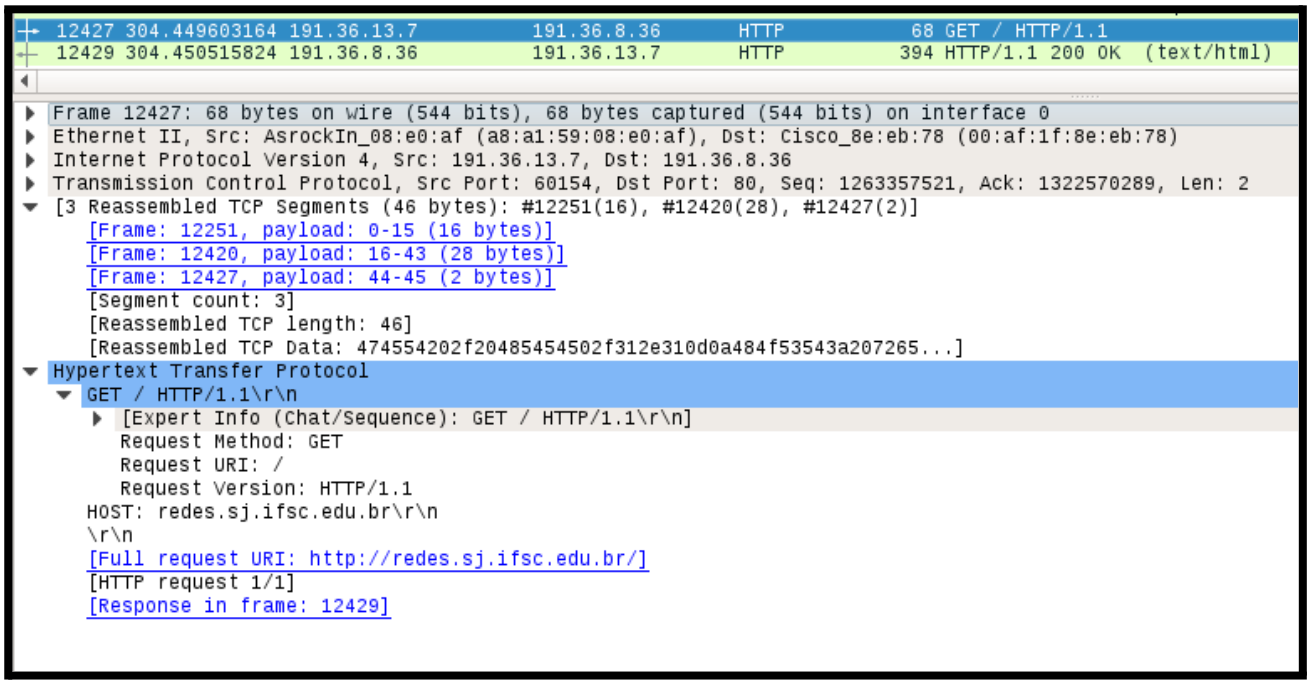


1. **Identifique a página html que foi enviada como resposta. Respeita o protocolo HTTP?**

   Respeita o cabeçalho visto na teoria.

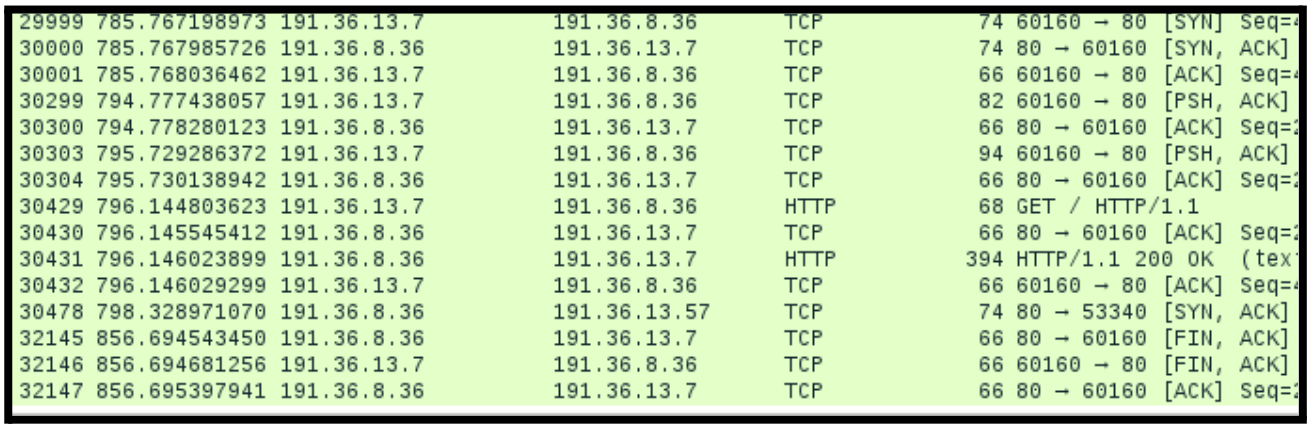
```
Connection closed by foreign host.
aluno: ~$ telnet -4 redes.sj.ifsc.edu.br 80
Trying 191.36.8.36...
Connected to redes.sj.ifsc.edu.br.
Escape character is '^]'.
GET / HTTP/1.1
HOST: redes.sj.ifsc.edu.br

HTTP/1.1 200 OK
Content-Type: text/html
Accept-Ranges: bytes
ETag: "940523071"
Last-Modified: Fri, 06 May 2022 14:27:22 GMT
Content-Length: 114
Date: Mon, 12 Sep 2022 13:54:18 GMT
Server: lighttpd/1.4.53

<html><body><h1>Redes de Computadores IFSC - SJ - Telecomunicacoes!</h1>
 <p>Pagina de teste.</p>
 </body></html>
```

2. **No Wireshark compare o resultado das execuções desses comandos com o que se viu nas capturas Wireshark com acesso pelo navegador. Qual a diferença em cada caso?**



3. Quanto tempo levou para fechar a conexão (após o duplo Enter)?

785 - 856 = 71 segundos



4. **Refaça um pedido em que o recurso é inexistente no servidor (ex: página html com nome/URL inexistente). Observe a resposta. Qual é o código da mensagem recebida?**

Resposta do servidor: 404 (Not Found), recurso não encontrado.

```
Trying 191.36.8.36...
Connected to redes.sj.ifsc.edu.br.
Escape character is '^]'.
GET /gg_teste HTTP/1.1
HOST: redes.sj.ifsc.edu.br

HTTP/1.1 404 Not Found
Content-Type: text/html
Content-Length: 341
Date: Mon, 12 Sep 2022 14:09:28 GMT
Server: lighttpd/1.4.53

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
         "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
 <head>
  <title>404 Not Found</title>
 </head>
 <body>
  <h1>404 Not Found</h1>
 </body>
</html>
```

4. O que explica a diferença de tempo para fechamento de conexão entre as versões HTTP 1.0 e 1.1?

No HTTP 1.1 a conexão é mantida durante um período para que sejam feitas solicitações sucessivas, neste caso o tempo para fechamento da conexão aumenta.

5. **Descreva qual seria o procedimento para o download de dois objetos, via telnet, nos protocolos HTTP 1.0 e 1.1?**

Para a versão 1.0, seria necessário realizar duas vezes o comando, um para cada solicitação.

Para a versão 1.1, basta apenas abrir a conexão com o comando telnet, e então realizar as duas solicitações em sequência.

Desvendando o HTTP com Wireshark, parte 2

Objetivos

- Explorar vários aspectos do protocolo HTTP:
  1. A requisição condicional.
  2. Formatos de mensagens HTTP.
  3. Os processos e protocolos envolvidos ao baixar arquivos grandes em HTML.
  4. Os processos envolvidos ao baixar arquivos em HTML com objetos incluídos.

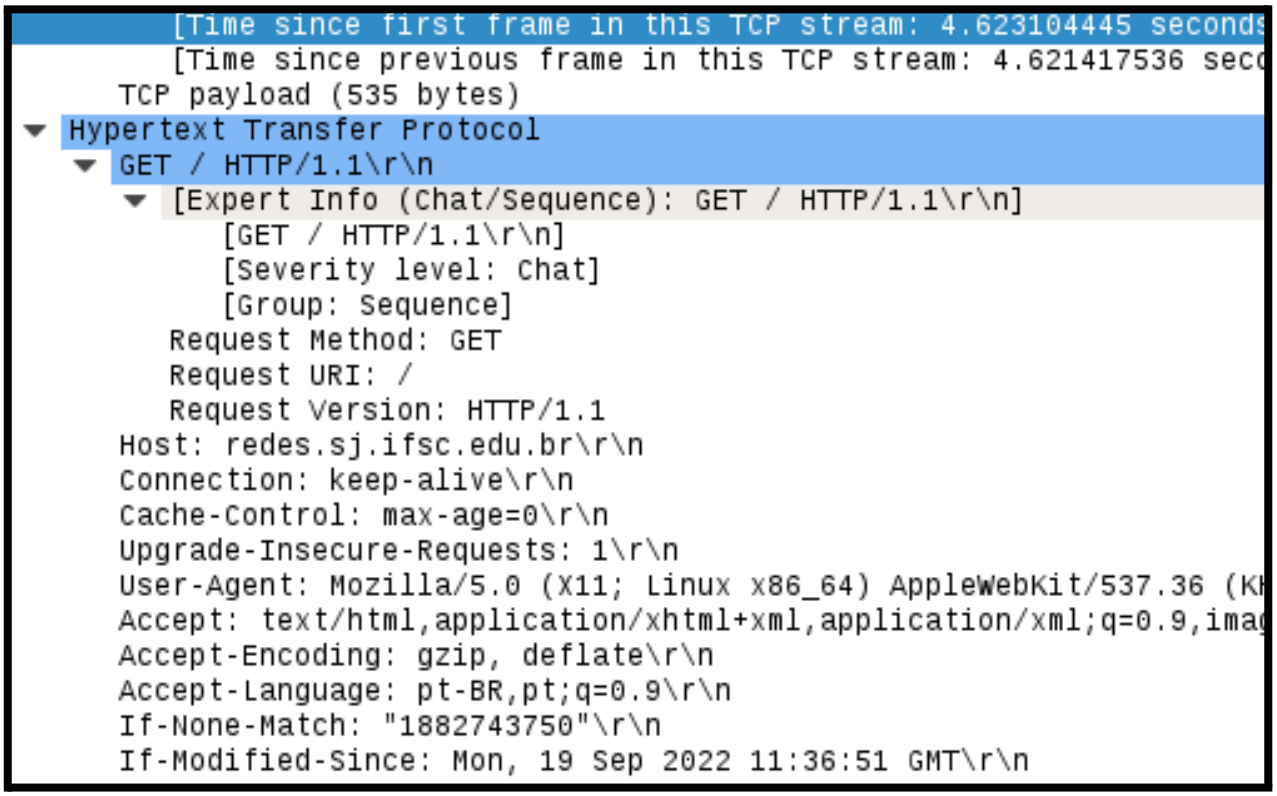
A Interação HTTP GET Condicional/Resposta

1. A maioria dos navegadores web tem um cache (seção 2.2.6 do livro) e, desta forma, realizam GET condicional quando baixam um objeto HTTP. Execute os seguintes passos:
   1. Inicie o navegador web;
   2. Limpe o cache do seu navegador (**Ctrl + Shift + Del**);
   3. Inicie o Wireshark;
   4. Digite o URL no navegador http://redes.sj.ifsc.edu.br. Seu navegador deve exibir um arquivo em HTML muito simples com duas linhas;
   5. Pressione o botão "refresh" no navegador (ou digite o URL novamente);
   6. No Wireshark pare a captura de pacotes e digite "http" na caixa de texto de especificação de filtro, para que apenas as mensagens HTTP sejam apresentadas na janela de listagem de pacotes.

Responda às seguintes questões (desconsidere a requisição e resposta (erro) da mensagem Favlcon):

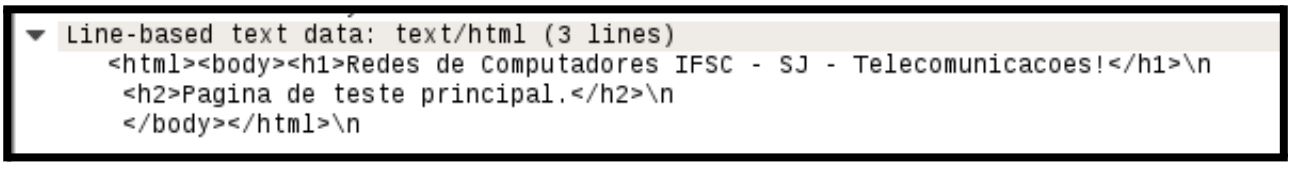
1) **Inspecione o conteúdo da primeira mensagem - HTTP GET - do seu navegador para o servidor redes.sj.ifsc.edu.br. Você vê uma linha "If-Modified-Since"?**

R: O campo "if-modified-since" não foi identificado no HTTP GET apenas na resposta dada pelo servidor.



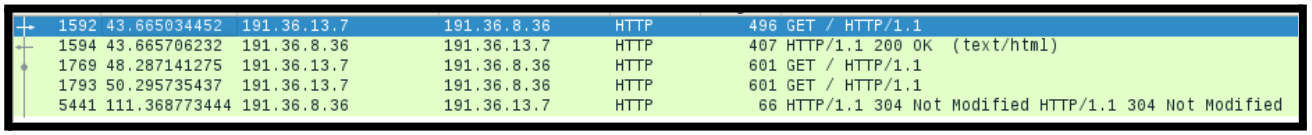
2) Inspecione o conteúdo da resposta do servidor, segunda mensagem. O servidor retornou explicitamente o conteúdo do arquivo? Como você pode dizer isso?

R: Sim, o conteúdo solicitado encontra-se no texto/html (seção body do HTTP), dentro do pacote enviado pelo servidor.



3) Agora inspecione o conteúdo da terceira mensagem - HTTP GET - do seu navegador para o servidor. Você vê uma linha "If-Modified-Since"? Caso a resposta seja afirmativa, qual informação segue o cabeçalho "If-Modified-Since"?

R: Sim, é a informação de quando ele recebeu o arquivo dado pelo servidor pela última vez, para ver se ocorreu modificação.



1592	43.665034452	191.36.13.7	191.36.8.36	HTTP	496 GET / HTTP/1.1
1594	43.665706232	191.36.8.36	191.36.13.7	HTTP	407 HTTP/1.1 200 OK (text/html)
1769	48.287141275	191.36.13.7	191.36.8.36	HTTP	601 GET / HTTP/1.1
1793	50.295735437	191.36.13.7	191.36.8.36	HTTP	601 GET / HTTP/1.1
5441	111.368773444	191.36.8.36	191.36.13.7	HTTP	66 HTTP/1.1 304 Not Modified HTTP/1.1 304 Not Modified

4) **Qual é o código de status e a frase retornada do servidor na resposta à segunda mensagem HTTP GET? É diferente do código de retorno da primeira mensagem?**

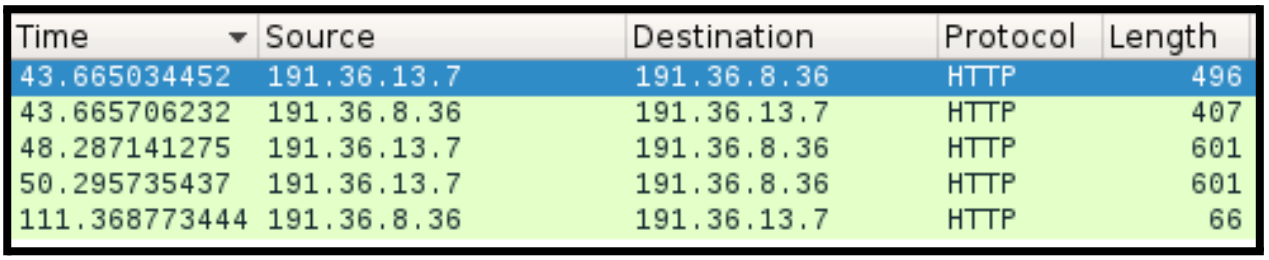
R: Código 304, a mensagem é “Not Modified”.

5) **Na segunda resposta, o servidor retornou explicitamente o conteúdo do arquivo? Explique.**

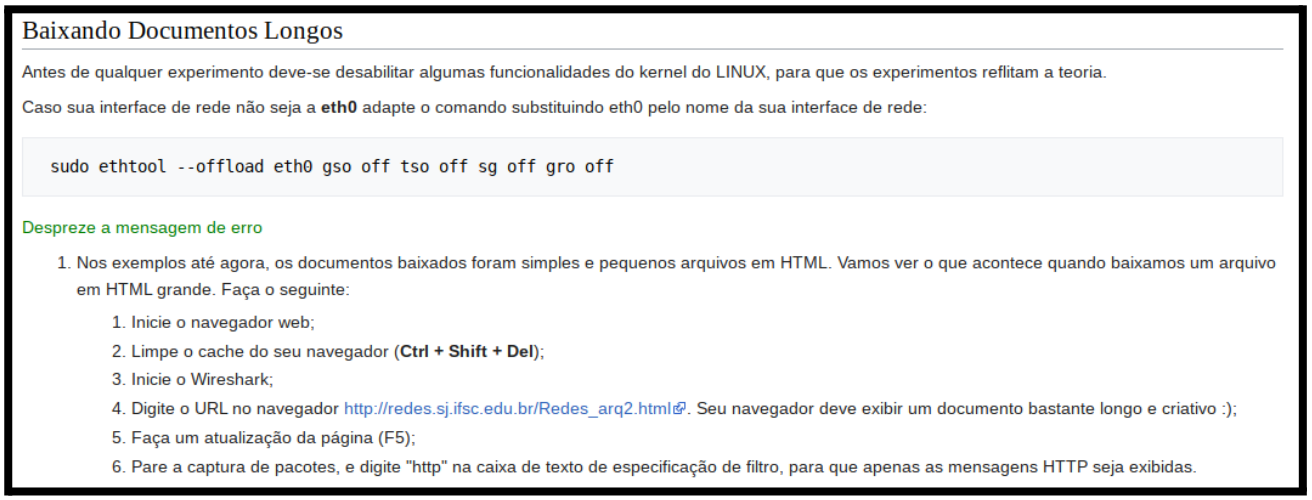
R: Não, no segundo retorno, o servidor retornou informando que a informação solicitada não foi alterada e portanto não será enviada.

6) **Qual o tamanho da primeira e segunda mensagem de retorno (respostas) do servidor?**

R: 407 e 66 respectivamente, conforme a imagem.



Time	Source	Destination	Protocol	Length
43.665034452	191.36.13.7	191.36.8.36	HTTP	496
43.665706232	191.36.8.36	191.36.13.7	HTTP	407
48.287141275	191.36.13.7	191.36.8.36	HTTP	601
50.295735437	191.36.13.7	191.36.8.36	HTTP	601
111.368773444	191.36.8.36	191.36.13.7	HTTP	66



Baixando Documentos Longos

Antes de qualquer experimento deve-se desabilitar algumas funcionalidades do kernel do LINUX, para que os experimentos reflitam a teoria.

Caso sua interface de rede não seja a **eth0** adapte o comando substituindo eth0 pelo nome da sua interface de rede:

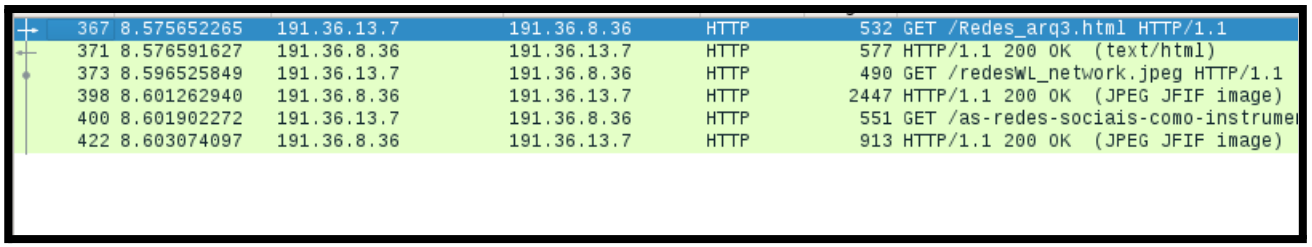
```
sudo ethtool --offload eth0 gso off tso off sg off gro off
```

Despreze a mensagem de erro

1. Nos exemplos até agora, os documentos baixados foram simples e pequenos arquivos em HTML. Vamos ver o que acontece quando baixamos um arquivo em HTML grande. Faça o seguinte:
    1. Inicie o navegador web;
    2. Limpe o cache do seu navegador (**Ctrl + Shift + Del**);
    3. Inicie o Wireshark;
    4. Digite o URL no navegador http://redes.sj.ifsc.edu.br/Redes_arq2.html. Seu navegador deve exibir um documento bastante longo e criativo :);
    5. Faça um atualização da página (F5);
    6. Pare a captura de pacotes, e digite "http" na caixa de texto de especificação de filtro, para que apenas as mensagens HTTP seja exibidas.

Responda às seguintes questões:

1) Quantas mensagens HTTP GET foram enviadas pelo seu navegador?



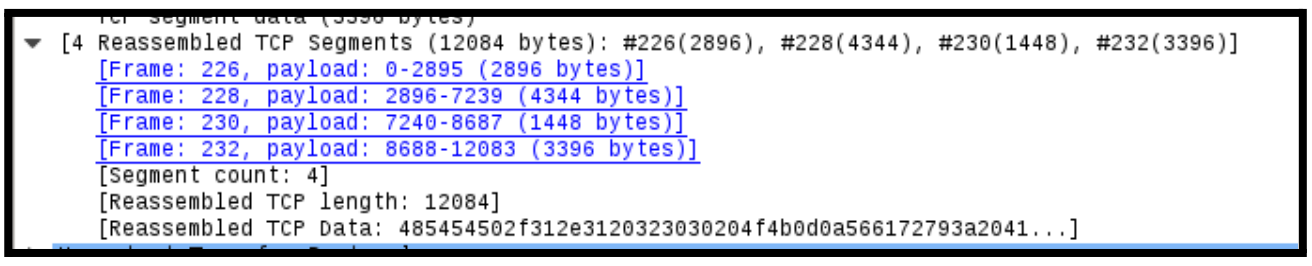
367	8.575652265	191.36.13.7	191.36.8.36	HTTP	532 GET /Redes_arq3.html HTTP/1.1
371	8.576591627	191.36.8.36	191.36.13.7	HTTP	577 HTTP/1.1 200 OK (text/html)
373	8.596525849	191.36.13.7	191.36.8.36	HTTP	490 GET /redesWL_network.jpeg HTTP/1.1
398	8.601262940	191.36.8.36	191.36.13.7	HTTP	2447 HTTP/1.1 200 OK (JPEG JFIF image)
400	8.601902272	191.36.13.7	191.36.8.36	HTTP	551 GET /as-redes-sociais-como-instrume
422	8.603074097	191.36.8.36	191.36.13.7	HTTP	913 HTTP/1.1 200 OK (JPEG JFIF image)

R: Foram enviadas, no total, 3 solicitações GET HTTP.

2) Quantas respostas HTTP sua máquina recebeu?

R: Foram recebidas, no total, 3 respostas (HTTP response).

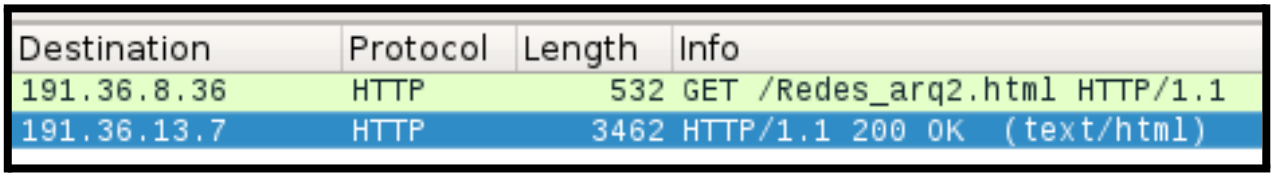
3) Quantos segmentos TCP foram necessários para carregar a resposta?



```
[4 Reassembled TCP Segments (12084 bytes): #226(2896), #228(4344), #230(1448), #232(3396)]
    [Frame: 226, payload: 0-2895 (2896 bytes)]
    [Frame: 228, payload: 2896-7239 (4344 bytes)]
    [Frame: 230, payload: 7240-8687 (1448 bytes)]
    [Frame: 232, payload: 8688-12083 (3396 bytes)]
    [Segment count: 4]
    [Reassembled TCP length: 12084]
    [Reassembled TCP Data: 485454502f312e3120323030204f4b0d0a566172793a2041...]
```

4) **Qual é o código de status e a frase associada com a resposta à mensagem HTTP GET? Obs:Observe os campos do cabeçalho de uma resposta HTTP.**

R: O código da resposta é: 200 OK.



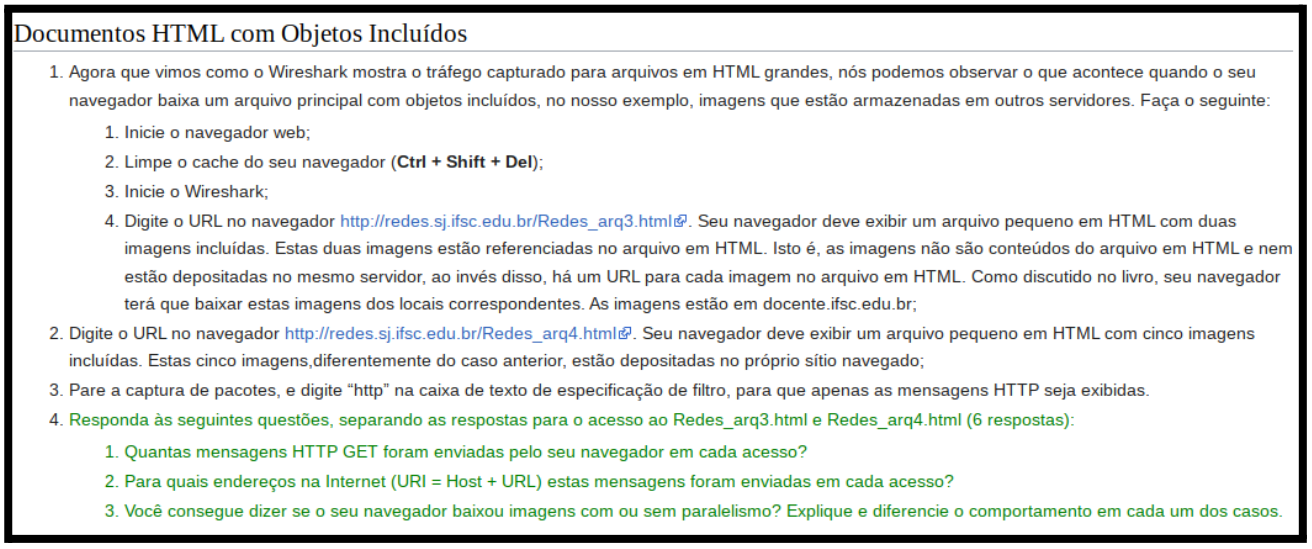
Destination	Protocol	Length	Info
191.36.8.36	HTTP	532	GET /Redes_arq2.html HTTP/1.1
191.36.13.7	HTTP	3462	HTTP/1.1 200 OK (text/html)

No segundo GET realizado, quantos segmentos TCP foram necessários para obtenção da resposta do servidor?

R: 9 no total, não foi retirada captura, pois o wireshark não conseguiu registrar o pacote.

5) **O que explica a diferença entre a primeira e segunda requisições?**

R: o MTU foi alterado para o padrão Ethernet de 1500 Bytes.



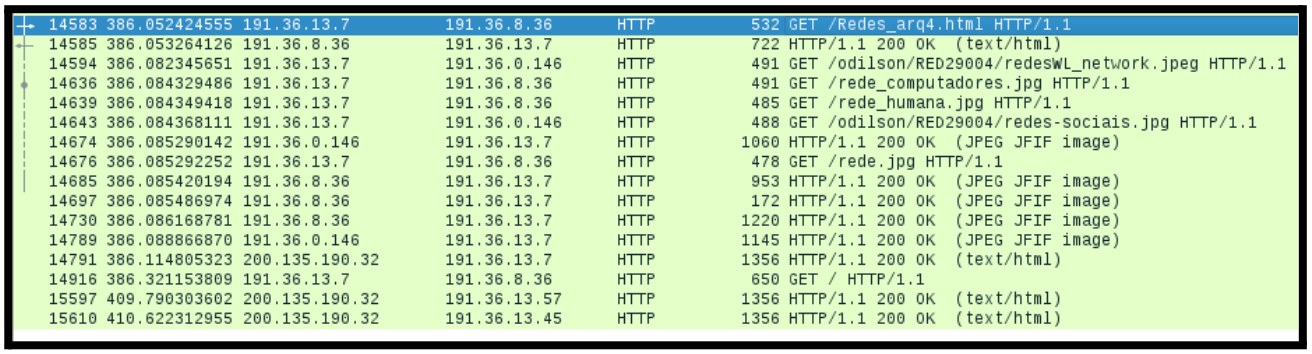
Documentos HTML com Objetos Incluídos

1. Agora que vimos como o Wireshark mostra o tráfego capturado para arquivos em HTML grandes, nós podemos observar o que acontece quando o seu navegador baixa um arquivo principal com objetos incluídos, no nosso exemplo, imagens que estão armazenadas em outros servidores. Faça o seguinte:
    1. Inicie o navegador web;
    2. Limpe o cache do seu navegador (**Ctrl + Shift + Del**);
    3. Inicie o Wireshark;
    4. Digite o URL no navegador http://redes.sj.ifsc.edu.br/Redes_arq3.html. Seu navegador deve exibir um arquivo pequeno em HTML com duas imagens incluídas. Estas duas imagens estão referenciadas no arquivo em HTML. Isto é, as imagens não são conteúdos do arquivo em HTML e nem estão depositadas no mesmo servidor, ao invés disso, há um URL para cada imagem no arquivo em HTML. Como discutido no livro, seu navegador terá que baixar estas imagens dos locais correspondentes. As imagens estão em docente.ifsc.edu.br;
2. Digite o URL no navegador http://redes.sj.ifsc.edu.br/Redes_arq4.html. Seu navegador deve exibir um arquivo pequeno em HTML com cinco imagens incluídas. Estas cinco imagens,diferentemente do caso anterior, estão depositadas no próprio sítio navegado;
3. Pare a captura de pacotes, e digite "http" na caixa de texto de especificação de filtro, para que apenas as mensagens HTTP seja exibidas.
4. Responda às seguintes questões, separando as respostas para o acesso ao Redes_arq3.html e Redes_arq4.html (6 respostas):
    1. Quantas mensagens HTTP GET foram enviadas pelo seu navegador em cada acesso?
    2. Para quais endereços na Internet (URI = Host + URL) estas mensagens foram enviadas em cada acesso?
    3. Você consegue dizer se o seu navegador baixou imagens com ou sem paralelismo? Explique e diferencie o comportamento em cada um dos casos.

Responda às seguintes questões, separando as respostas para o acesso ao Redes_arq3.html e Redes_arq4.html (6 respostas):

1) **Quantas mensagens HTTP GET foram enviadas pelo seu navegador em cada acesso?**

R: Foram enviadas 3 requisições no redes_arq3, e 7 requisições no redes_arq4.



```
14583 386.052424555 191.36.13.7     191.36.8.36      HTTP    532 GET /Redes_arq4.html HTTP/1.1
14585 386.053264126 191.36.8.36     191.36.13.7      HTTP    722 HTTP/1.1 200 OK  (text/html)
14594 386.082345651 191.36.13.7     191.36.0.146     HTTP    491 GET /odilson/RED29004/redesWL_network.jpeg HTTP/1.1
14636 386.084329486 191.36.13.7     191.36.8.36      HTTP    491 GET /rede_computadores.jpg HTTP/1.1
14639 386.084349418 191.36.13.7     191.36.8.36      HTTP    485 GET /rede_humana.jpg HTTP/1.1
14643 386.084368111 191.36.13.7     191.36.0.146     HTTP    488 GET /odilson/RED29004/redes-sociais.jpg HTTP/1.1
14674 386.085290142 191.36.0.146    191.36.13.7      HTTP   1060 HTTP/1.1 200 OK  (JPEG JFIF image)
14676 386.085292252 191.36.13.7     191.36.8.36      HTTP    478 GET /rede.jpg HTTP/1.1
14685 386.085420194 191.36.8.36     191.36.13.7      HTTP    953 HTTP/1.1 200 OK  (JPEG JFIF image)
14697 386.085486974 191.36.8.36     191.36.13.7      HTTP    172 HTTP/1.1 200 OK  (JPEG JFIF image)
14730 386.086168781 191.36.8.36     191.36.13.7      HTTP   1220 HTTP/1.1 200 OK  (JPEG JFIF image)
14789 386.088866870 191.36.0.146    191.36.13.7      HTTP   1145 HTTP/1.1 200 OK  (JPEG JFIF image)
14791 386.114805323 200.135.190.32  191.36.13.7      HTTP   1356 HTTP/1.1 200 OK  (text/html)
14916 386.321153809 191.36.13.7     191.36.8.36      HTTP    650 GET / HTTP/1.1
15597 409.790303602 200.135.190.32  191.36.13.57     HTTP   1356 HTTP/1.1 200 OK  (text/html)
15610 410.622312955 200.135.190.32  191.36.13.45     HTTP   1356 HTTP/1.1 200 OK  (text/html)
```

2) **Para quais endereços na Internet (URI = Host + URL) estas mensagens foram enviadas em cada acesso?**

R: para o endereço do servidor 191.36.8.36 (dentro do diretório especificado em cada requisição (última coluna na captura acima).

3) **Você consegue dizer se o seu navegador baixou imagens com ou sem paralelismo? Explique e diferencie o comportamento em cada um dos casos.**

R: Sim, as solicitações HTTP GET foram enviadas em sequência (4 GETs em sequência), o retorno também foi feito em sequência.