Trabalho 3 - Redes de Computadores I

Gustavo Kundlatsch (17102810)

16 de Junho de 2019

Resumo

O objetivo desse trabalho é obter conhecimentos práticos no uso do software Wireshark, que analisa o tráfego de rede, e o organiza de acordo com protocolos, IPs envolvidos ou diversos outros filtros que o usuário pode selecionar. Através desse aplicativo podemos controlar o tráfego de uma rede e monitorar a entrada e saída de dados do computador, através de diversos protocolos, ou a entrada e saída de dados da rede à qual o computador está ligado.

1 Introdução

Nesse trabalho vamos utilizar o software Wireshark para identificar pacotes realizando conexões, transferência de dados e a desconexão, analisando as camadas de aplicação, rede e transporte. Os pacotes analisados na demonstração utilizaram o protocolo TCP e HTTP. O trabalho foi realizado utilizando um computador pessoal, com um processador Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz, 8GB de memória RAM DDR3 e uma placa de vídeo NVIDIA GeForce 930M. A rede em que o computador estava conectado era a rede pessoal, de uma república (uma casa contendo seis estudantes, com diversos dispositivos cada, como computadores, notebooks, celulares e tablets).

2 Funcionamento

2.1 Conexão

No processo de estabelecer uma conexão, o objetivo normalmente é estabelecer uma ligação entre o usuário final e o servidor. O primeiro passo para isso acontecer é o usuário enviar um pacote TCP com a flag [SYN], com o objetivo de estabelecer a conexão. Caso o servidor possa aceitar a conexão, ele irá responder com um pacote com a flag [SYN, ACK] para satisfazer a mensagem anterior. Quando o usuário recebe esse pacote, ele reponde com um pacote [ACK] para satisfazer a mensagem do servidor. Se todos os passos funcionarem, a conexão será bem estabelecida.

2.2 Transferência de Dados

Para explicar o funcionamento da transferência de dados, vamos conceituar algumas coisas primeiro. TCP/IP é um conjunto de protocolos. Diferente do formato OSI que é dividido em sete camadas, os protocolos TCP/IP está equivalentemente organizado em quatro camadas apenas. Quando dois programas se comunicam através da internet, por exemplo, eles estão em contato direto com a primeira camada TCP/IP: Aplicação (mesmo nome da primeira camada OSI).

Nessa camada de aplicação, existem diversos protocolos com diversos objetivos. O foco desse trabalho é explorar a troca de dados HTTP (Hypertext Transfer Protocol). Todo protocolo é criado com o objetivo de padronizar algo que pode ser feito de diversas maneiras. No caso do HTTP, o objetivo é padronizar a transferência de hypermídia, como textos, imagens e sons. Conceitualmente, o hipertexto é uma ligação cujo objetivo é facilitar a navegação dos usuários, através de um texto que pode ser composto de diversas palavras, imagens ou até mesmo sons que, ao serem clicados, direcionam o usuário para outra página onde se esclarece com mais precisão o assunto do link abordado.

Depois de ocorrer o processamento dessa requisição, a camada de aplicação chama a camada de transporte através da porta 80 (esse valor é específico do protocolo HTTP, cada protocolo possui uma porta própria), indicando que tipo de conteúdo será enviado. Em seguida, a camada de transporte deve dividir os dados em pacotes e ordená-los. É aqui que acontece o uso do TCP (Transmission Control Protocol), que tem como objetivo encapsular as informações de controle da seguinte maneira: número da porta de origem, número da porta destino, número de sequência (Seq), soma de verificação (ACK). Após ser encapsulado em forma de cabeçalho de acordo com o TCP, a camada Internet adere os endereços IPs da origem e destino do pacote, para finalmente os pacotes saírem para a Interface com a Rede. Essa camada depende do tipo de rede em que o computador está conectado. É muito comum a rede ser do tipo Ethernet, que adiciona dados no começo dos pacotes e finalmente os convertem em impulsos elétricos.

2.3 Desconexão

Quando ou o usuário final ou o servidor querer terminar a conexão, ele deve enviar um pacote TCP com a flag [FIN, ACK]. Caso nenhum erro aconteça, a contra parte irá devolver uma confirmação com uma flag [ACK] num pacote TCP e, então, deve enviar o seu próprio pacote com a flag [FIN, ACK] e esperar a resposta [ACK] correspondente.

3 Experimento

Nessa seção vamos desenvolver como foram realizados os experimentos (testes) com o Wireshark para estudar o funcionamento das operações descritas na seção anterior. Para o desenvolvimento do experimento, foi utilizado o site

http://wombocombo.com.br/, um blog de notícias de e-sports, que não utiliza SSL (protocolo HTTPS), permitindo visualizar os pacotes.

3.1 Conexão

A conexão é dividida nas três etapas anteriormente descritas. Nas imagens abaixo, as legendas descrevem o processo que está sendo realizado/analisado em cada momento.

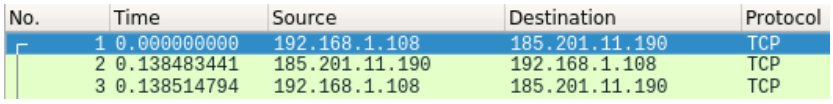


Figura 1: Aqui mostramos os IPs que estão se conectando, e o protocolo que está sendo utilizado

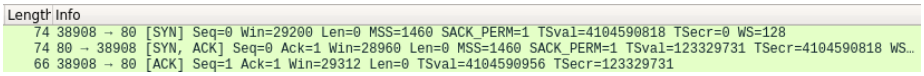


Figura 2: Informações das operações, mostrando o funcionamento de cada uma das etapas, com as flags [SYN], [SYN, ACK] e [ACK] aparecendo em ordem

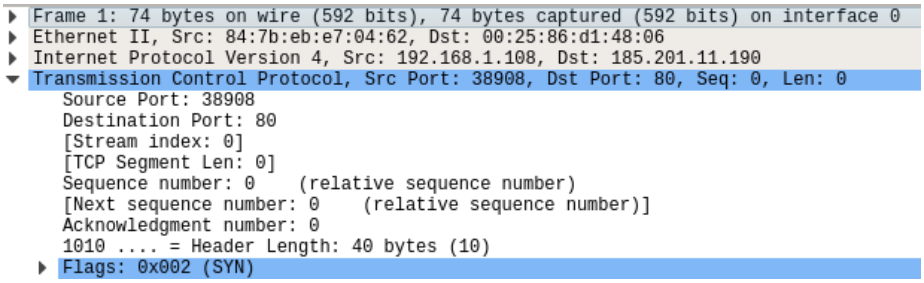


Figura 3: Aqui temos o primeiro pacote aberto, e temos um TCP saíndo do computador indo para o servidor. O pacote leva a flag SYN. Depois disso, é esperado que o servidor responda com as flags SYN, ACK, sendo que o ACK deve ser igual ao SYN do pacote 1, ou seja, '1'. A porta de destino é 80, como esperado, pois o esperado é um HTTP.

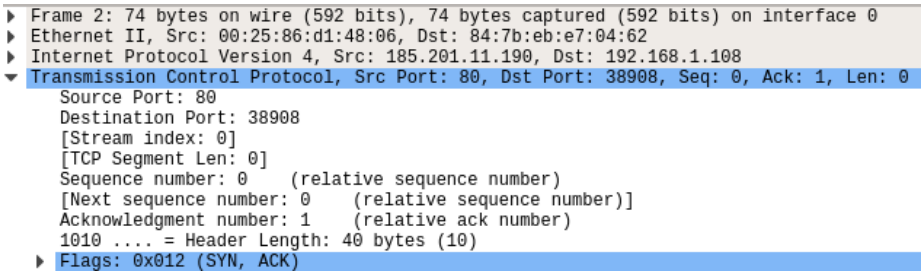


Figura 4: O pacote 2 é a resposta do servidor dizendo o esperado. Tudo que resta é enviar um pacote para o servidor com ACK igual ao do pacote 15, ou seja, '1'.

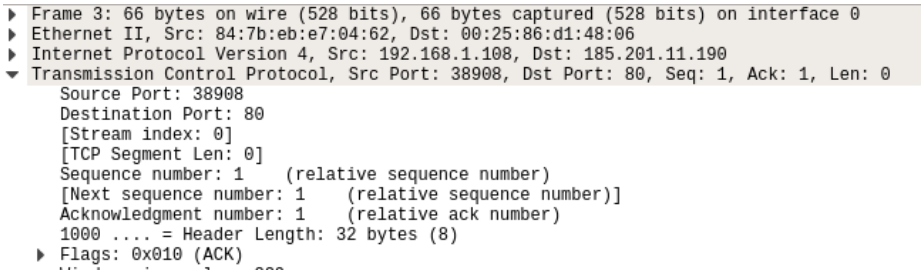


Figura 5: Por fim temos o resultado esperado: ACK = 1, SEQ = 1. A conexão foi bem sucedida.

3.2 Transferência de Dados

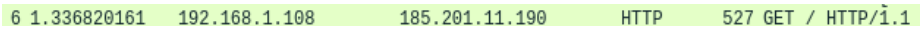


Figura 6: Essa linha é um pedido do computador para que o servidor forneça a página acessada (homepage).

Figura 7: Esta imagem mostra os detalhes do pacote, mostrando o método GET. Ao lado de get aparece uma barra, porque o endereço acessado foi a homepage do site. Caso fosse outra página, haveria o caminho até ela. Depois da barra, aparece a versão do protocolo, nesse caso 1.1.

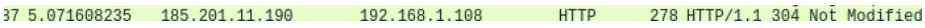


Figura 8: O servidor pode ter diversas respostas, o código 200, por exemplo, é o código de sucesso. Nesse caso, o retorno foi um código que significa "Não alterado". Esse código serve para motivos de cache, ou seja, a versão da última solicitação do mesmo tipo não sofreu mudanças, então a versão em cache pode ser reaproveitada. Isso acontece pois esse é um site que é frequentemente visitado no computador utilizado para o experimento, ou seja, a versão mais recento do site já estava em chache.

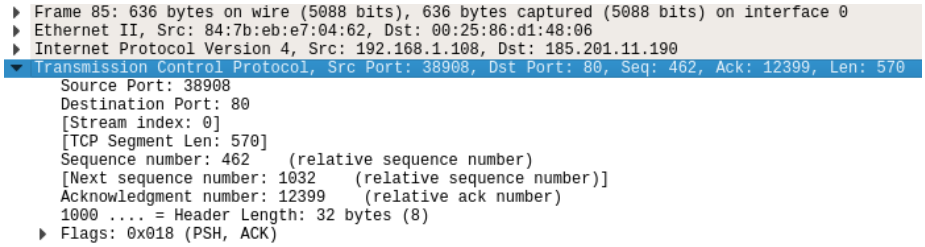
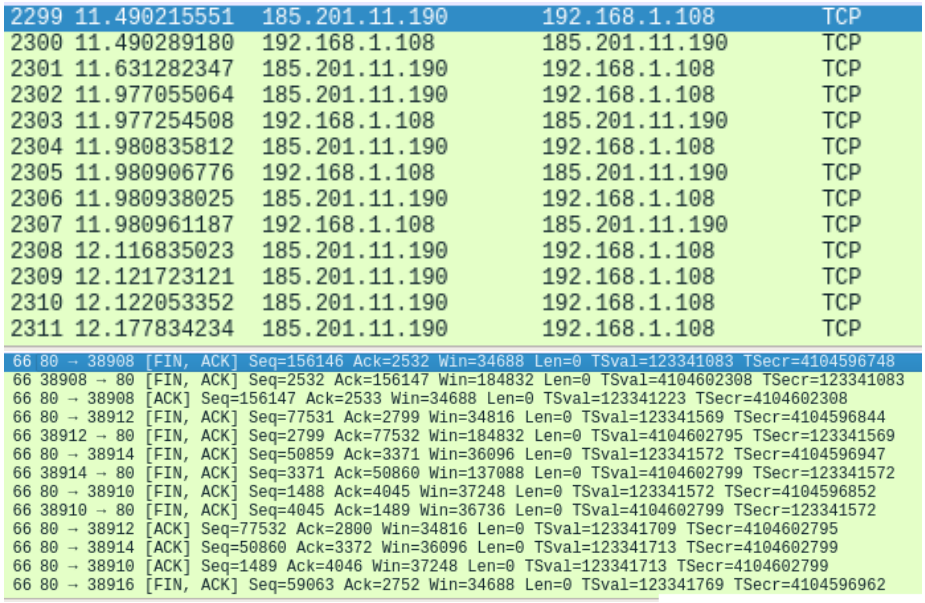


Figura 9: Aqui vemos mais informações do pacote, como portas de origem e destino, e o número de sequência e o próximo número de sequência. Esse tipo de informação é mais ou menos relevante de acordo com o tipo da aplicação que está transmitindo. No caso de um servidor de um site de streaming de vídeo, por exemplo, o número de sequência e o próximo número de sequência se tornam bastante relevantes para manter a consistência do vídeo que está sendo transmitido.

3.3 Desconexão



2299	11.490215551	185.201.11.190	192.168.1.108	TCP
2300	11.490289180	192.168.1.108	185.201.11.190	TCP
2301	11.631282347	185.201.11.190	192.168.1.108	TCP
2302	11.977055064	185.201.11.190	192.168.1.108	TCP
2303	11.977254508	192.168.1.108	185.201.11.190	TCP
2304	11.980835812	185.201.11.190	192.168.1.108	TCP
2305	11.980906776	192.168.1.108	185.201.11.190	TCP
2306	11.980938025	185.201.11.190	192.168.1.108	TCP
2307	11.980961187	192.168.1.108	185.201.11.190	TCP
2308	12.116835023	185.201.11.190	192.168.1.108	TCP
2309	12.121723121	185.201.11.190	192.168.1.108	TCP
2310	12.122053352	185.201.11.190	192.168.1.108	TCP
2311	12.177834234	185.201.11.190	192.168.1.108	TCP

```
66 80 → 38908 [FIN, ACK] Seq=156146 Ack=2532 Win=34688 Len=0 TSval=123341083 TSecr=4104596748
66 38908 → 80 [FIN, ACK] Seq=2532 Ack=156147 Win=184832 Len=0 TSval=4104602308 TSecr=123341083
66 80 → 38908 [ACK] Seq=156147 Ack=2533 Win=34688 Len=0 TSval=123341223 TSecr=4104602308
66 80 → 38912 [FIN, ACK] Seq=77531 Ack=2799 Win=34816 Len=0 TSval=123341569 TSecr=4104596844
66 38912 → 80 [FIN, ACK] Seq=2799 Ack=77532 Win=184832 Len=0 TSval=4104602795 TSecr=123341569
66 80 → 38914 [FIN, ACK] Seq=50859 Ack=3371 Win=36096 Len=0 TSval=123341572 TSecr=4104596947
66 38914 → 80 [FIN, ACK] Seq=3371 Ack=50860 Win=137088 Len=0 TSval=4104602799 TSecr=123341572
66 80 → 38910 [FIN, ACK] Seq=1488 Ack=4045 Win=37248 Len=0 TSval=123341572 TSecr=4104596852
66 38910 → 80 [FIN, ACK] Seq=4045 Ack=1489 Win=36736 Len=0 TSval=4104602799 TSecr=123341572
66 80 → 38912 [ACK] Seq=77532 Ack=2800 Win=34816 Len=0 TSval=123341709 TSecr=4104602795
66 80 → 38914 [ACK] Seq=50860 Ack=3372 Win=36096 Len=0 TSval=123341713 TSecr=4104602799
66 80 → 38910 [ACK] Seq=1489 Ack=4046 Win=37248 Len=0 TSval=123341713 TSecr=4104602799
66 80 → 38916 [FIN, ACK] Seq=59063 Ack=2752 Win=34688 Len=0 TSval=123341769 TSecr=4104596962
```

Figura 10: Nestas duas imagem, que são uma complemento da outra (a primeira ficaria do lado esquerdo, e a segunda no direito, na visualização do Wireshark) temos uma série de pedidos de desconexão. Como foi explicado anteriormente, quem quiser fazer a desconexão (nesse caso o servidor) envia um pacote TCP com a flag [FIN/ACK]. Como não ocorreu nenhum erro, a contra parte (computador) devolveu uma confirmação com uma flag [ACK] num pacote TCP (pacote 2301) e, então, deve enviou o seu próprio pacote com a flag [FIN/ACK], que teve como resposta um pacote [ACK] do servidor, confirmando a desconexão.

4 Conclusão

Ao decorrer do experimento, foi possível aprender na prática conceitos que haviam sido trabalhados em sala de aula, de maneira bastante direta e que pode render várias visões de como a rede funciona na vida real. Durante a análise da terceira etapa do experimento, a desconexão, houveram alguns problemas em identificar quais pacotes estavam fazendo quais operações, devido a distância entre a última ação (a transferência de dados) e a desconexão, além de várias desconexões aparentemente estarem ocorrendo juntas. Tirando isso, foi possível realizar o experimento sem problemas, gerando análises concisas daquilo que foi observado.