Avaliação Progressiva de Redes 5 - Camada de Transporte

Vinícius Takeo Friedrich Kuwaki

14 de Julho de 2020

1 Análise dos Dumps de Tráfego

A seguir serão descritas as comunicações capturadas pelo wireshark em alguns arquivos .dump disponibilizados no Moodle pelo professor.

1.1 Bittorrent

O arquivo de dump do bittorrent, descreve uma comunicação, utilizando o protocolo Bittorrent, para a transmissão de dados entre dois pares. O protocolo de transporte que foi utilizado para a comunicação foi o TCP. A comunicação acontece entre dois pares, para facilitar a nomenclatura, serão chamados de A e B. O A envia um handshake para o B. Que responde com um TCP ACK e um PUSH, requisitando que os dados devem ser enviados imediatamente. Após isso o A informa que os dados podem começar a ser transmitidos. Mas ao receber a confirmação o B estrangula a conexão através de um Unchoke. O A então, tenta pedir a sequência seguinte do fluxo de bytes mas os dumps indicam que o A tentava a todo momento capturar os pacotes, mas nunca conseguia. Ao final, os dados não chegam a ser transmitidos de B para A.

1.2 DNS

Por outro lado, no dump chamado "DNS Questions Answer", é descrito uma operação bem simples de consulta de endereço IP no servidor DNS. O requisitante pede pelo endereço de IP de um site, informando seu URL e o servidor retorna o IP do respectivo endereço de URL. Tudo acontece via UDP, logo não há estabelecimento de conexão nem nada, apenas uma simples requisição e resposta.

1.3 HTTP

No próximo arquivo, o "http", uma conexão TCP é estabelecida entre um "navegador"e um servidor, primeiro a conexão é estabelecida via TCP entre as duas partes. Após estabelecida a conexão com o servidor, o cliente faz um

HTTP GET solicitando o conteúdo da página. O conteúdo é transmitido via TCP para o solicitante, seguindo sempre o mesmo processo, recebe, envia um ACK requisitando o próximo, e assim por diante. Em determinando momento da transmissão dos pacotes, o cliente solicita ao servidor DNS, o IP de um anúncio, que muito provavélmente estava dentro da página web que o mesmo acessou no navegador. O servidor DNS responde com o IP do servidor desse anúncio. O cliente então envia um HTTP GET para esse servidor, que transmite de volta por TCP o anúncio que o cliente requisitou via GET. O cliente termina de baixar os dados do servidor do anuncio e envia um HTTP OK, confirmando que todos os dados foram recebidos. Após isso, a transmissão do restante dos dados que estava acontecendo antes é resumida. Ao final acontece uma interrupção, quando o cliente acaba recebendo um pacote repetido, mas o processo se encerra com todos os dados sendo recebidos com sucesso. O cliente transmite um HTTP OK para informar o servidor de que tudo foi recebido. Após o OK, o cliente informa via TCP que a conexão já pode ser fechada, recebendo um FIN do servidor e encerrando a conexão.

1.4 IPv4

No arquivo "IPv4"por outro lado, descreve uma operação de leitura de e-mail em um cliente Gmail do Google. Através de TCP, o cliente se autentica com o servidor da Google. O servidor responde aceitando a conexão. Após conectado, o servidor envia para o cliente via protocolo SMTP um resposta. Sua confirmação de recebimento é enviada via TCP ACK. O cliente então pergunta qual a versão EHLO o servidor está usando. O servidor então responde que recebeu (via TCP) e envia sua versão para o cliente (via SMTP). O cliente comunica que recebeu (TCP) e depois envia um comando QUIT via SMTP para o servidor. Após isso a conexão do cliente com o servidor é finalizada. O servidor responde em SMTP e o cliente comunica que recebeu, com um TCP ACK. O servidor recebe a comunicação via TCP de que recebeu, e envia um ACK com FIN, finalizando a conexão.

1.5 SSH

O próximo dump examinado é o "SSHv2". A comunicação se inicia com a fonte enviando um SYN para sincronização com o destino, que comunica um SYN e ACK, confirmando que recebeu e sincronizou, todas essa comunicação se deu via TCP. Após isso a fonte informa que recebeu e solicita o envio da primeira sequência de bytes. Após isso, via protocolo SSHv2, o servidor(destino) e o cliente(fonte) informam suas respectivas versões do protocolo que estão utilizando. Após descoberta as versões dos protocolos SSH, a transferência de dados entre as partes acontece quando o servidor envia ao cliente uma chave, via SSHv2. Após, o cliente envia para o servidor três ACK's (por TCP) e uma chave SSHv2. Após, ocorre uma troca de chaves entre o cliente e o servidor utilizando o método Diffie-Hellman para executar a troca de chaves. Após a troca de chaves, acontece a troca de pacotes de forma encriptada entre o servidor e o

cliente via SSHv2. No meio do processo, alguns TCP ACK são enviados para confirmar os recebimentos e requisitar os próximos. Em certo ponto da conexão, (numero 79 no wireshark), o servidor envia um pacote encriptado, mas o cliente pede via TCP um RST para resetar a conexão. Alguns periodos a frente o servidor acaba recebendo três ACK's duplicados, forçando o servidor a resetar a conexão via TCP RST de acordo com a política de controle de congestionamento. Os registros do dump se encerram com o cliente requisitando um reset (TCP RST) para o servidor.

1.6 Telnet

Por fim, analizando o dump no arquivo "telnet", chega-se a conclusão de que a comunicação que acontece entre as duas partes, até pela natureza do protocolo utilizado é uma comunicação com um terminal virtual. A fonte provavelmente está controlando um servidor remoto utilizando o protocolo Telnet. Primeiro uma conexão TCP é estabelecida entre a fonte e o destino. Após estabelecida, a fonte envia um comando via protocolo Telnet que é aceito pelo servidor. O servidor comunica que recebeu (via TCP ACK) e transmite também a resposta utilizando Telnet também. Basicamente o processo se repete várias vezes, uma das partes envia o comando via Telnet, a outra responde, também utilizando Telnet e com a confirmação TCP ACK. Ao final, a fonte requisita o fim da conexão via TCP ACK, e o destino responde confirmando com um TCP FIN e ACK.

2 Analisando a captura de pacotes

Nessa seção, será analisada a captura de pacotes durante o download do aplicativo Discord. A seguir está o gráfico de fluxo de rede:

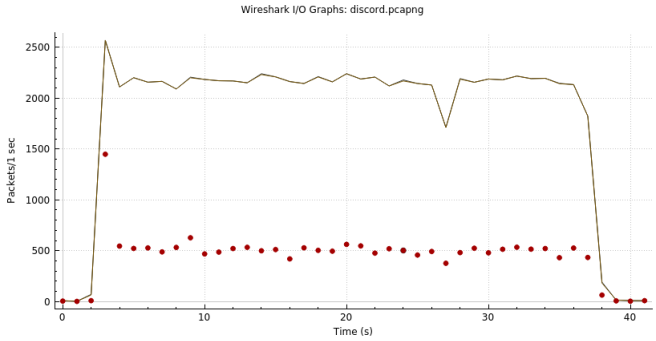

Figura 1: Gráfico do fluxo de rede durante o download da aplicação Discord no formato .deb, disponível nesse link. Os pontos em vermelho representam os pacotes perdidos

2.1 Abertura e fechamento da conexão

A abertura da conexão se deu no instante 55 da Figura 2. Após realizar um filtro buscando SYN e ACK e o IP da fonte (192.168.0.15), constata-se que a conexão é estabelecida nesse instante.

Já o close, acontece no instante 75897 da Figura 3. Na figura é possível ver que no instante 75887, o dispositivo de origem (192.168.0.15) envia a confirmação ACK, requisitando a próxima sequência. O destino (162.159.130.232) então faz um PSH (um push) enviando todo o buffer local, pois todos os dados já foram transmitidos para a origem. No instânte seguinte, a origem confirma que recebeu o último pacote e a conexão então acaba.

2.2 Retransmissões durante a conexão

Um dos momentos em que houve mais retransmissões de pacotes é mostrado na Figura 4. Nela é possível observar que entre o instante 13673 e 13683 houveram quase 5 retransmissões seguidas. Uma possível explicação para isso pode ser o fato de que, o destino (162.159.130.232) ficou aguardando o ACK de algum dos pacotes que fora enviado, mas a origem (192.168.0.15) ainda não havia enviado essa confirmação, ou até mesmo ela pode ter sido perdida no meio do caminho. E como o algoritmo da janela deslizante não avança até que todos os pacotes daquela janela tenha chego, o destino ficou reenviando até que a fonte envie o ACK confimando o recebimento.

2.3 Tamanho máximo das janelas

O tamanho da janela de congestionamento máximo é 24568 de acordo com o dump gerado no momento do download. Esse valor foi informado pelo origem no ACK do handshake 2. Já o tamanho da janela de recebimento pode ser obtido analisando o pico do gráfico, (ver a Figura 1), isto é 2570.

2.4 Relação entre número de sequência e tempo

O número de sequência cresce conforme o tempo para pacotes que foram recebidos com sucesso. Pois a cada ACK, o origem (192.168.0.15) requisita o próximo número da sequência, até que o número de sequência atinja todos os bytes que serão transmitidos, e a conexão ser encerrada.

Referências

TRANSMISSION, C. d. C. no; FILHO, P. C. H. Anatomia do bittorrent.

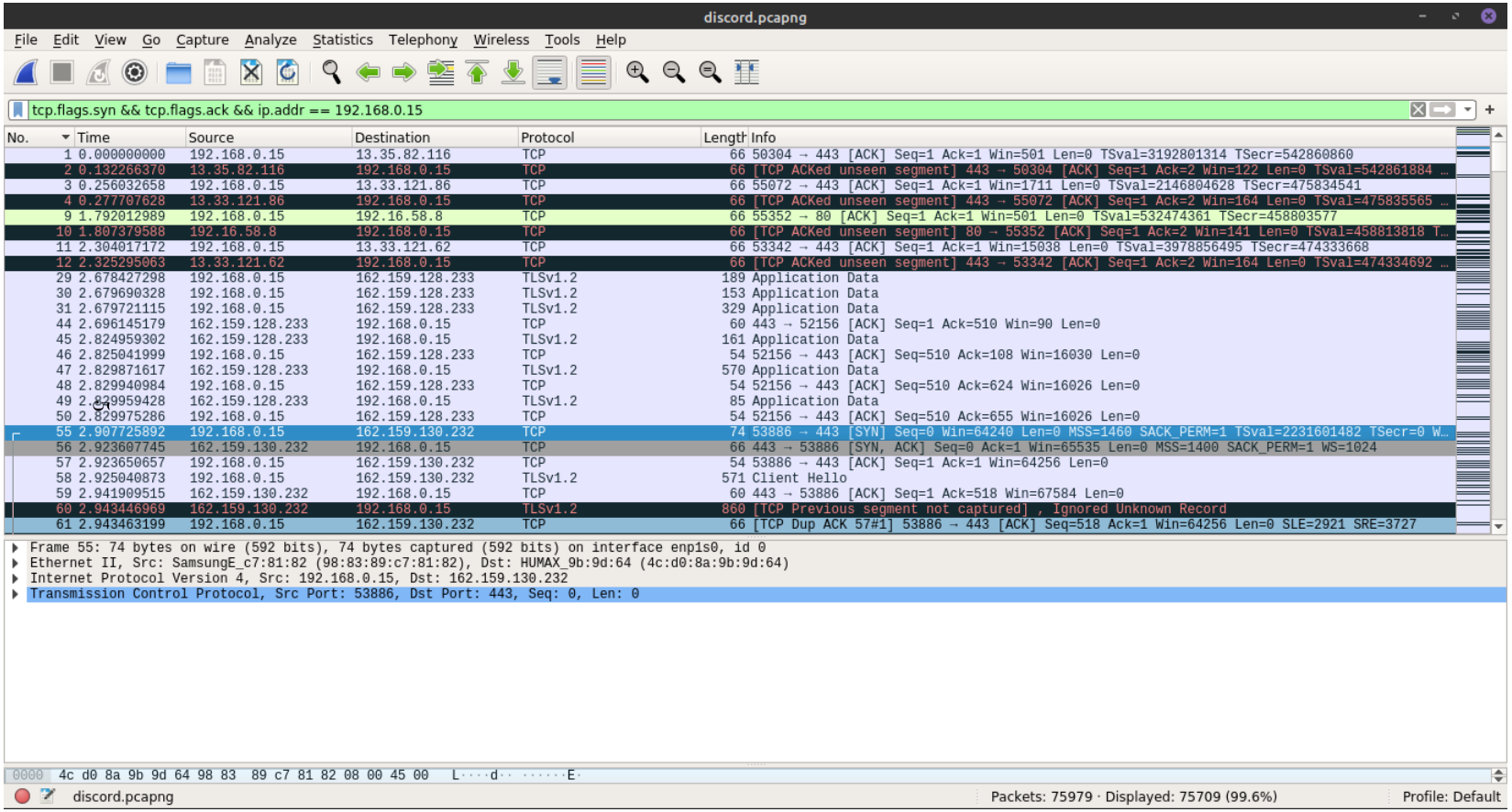

Figura 2: Estabelecimento de conexão para o download

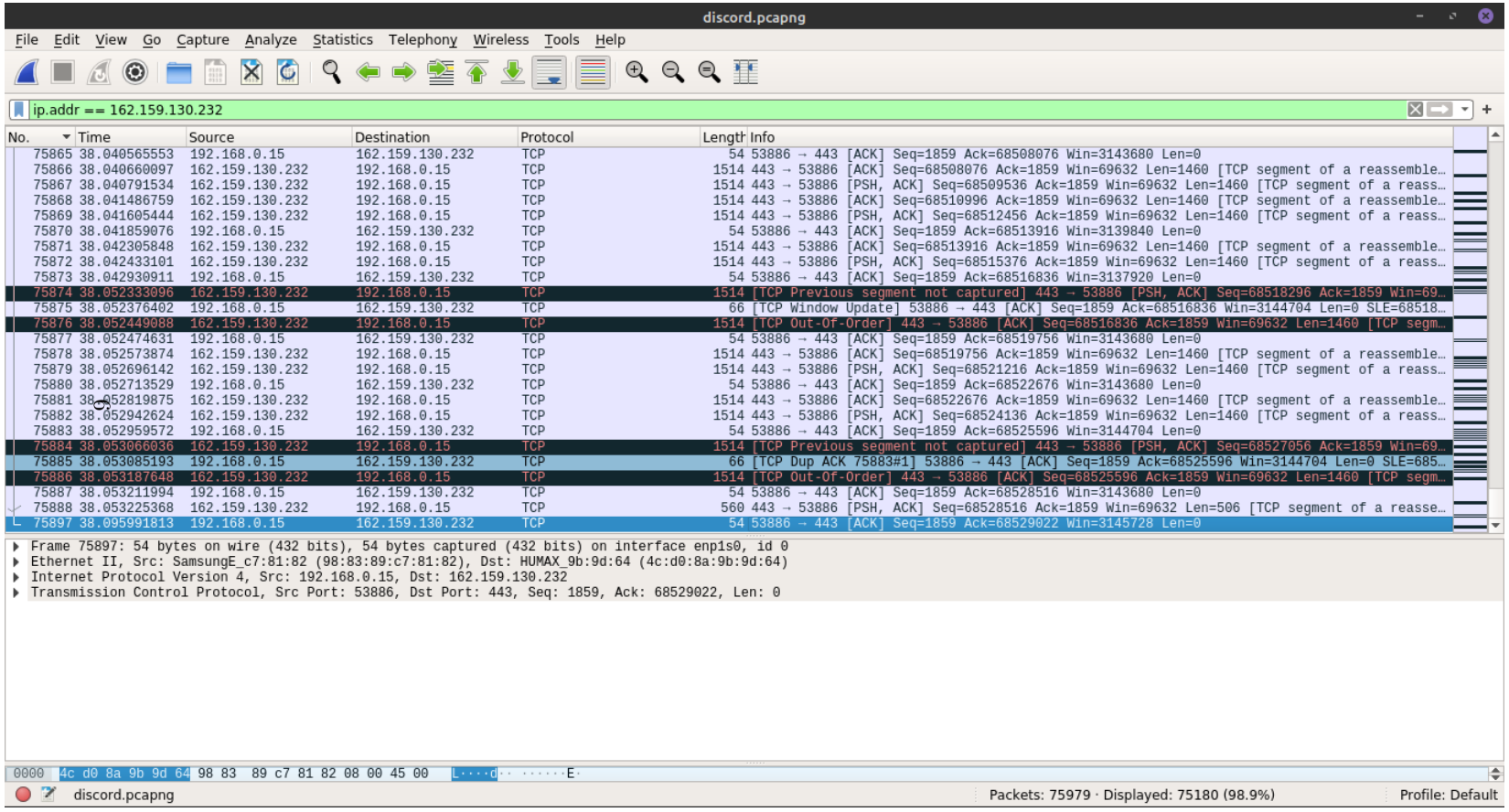

Figura 3: Encerramento da conexão para o download

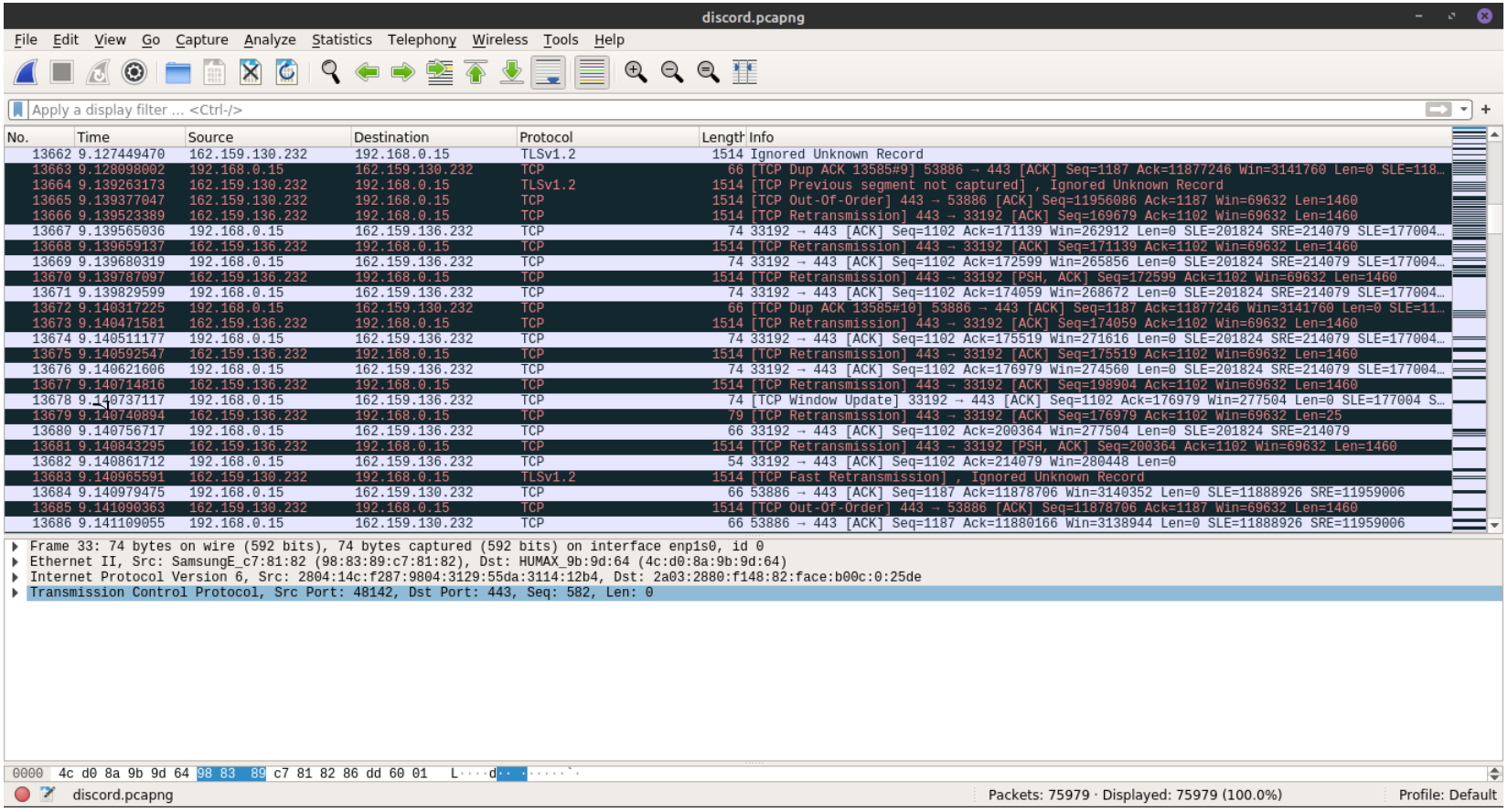

Figura 4: Momento tumultuado na transmissão de pacotes do download, houveram muitas retransmissões nesse período.