

Resumo

nsQUIC: Uma extensão para Simulação de Protocolo QUIC no NS-3

Renê Cardozo
rene.epcrdz@gmail.com

27 de maio de 2020

Link do artigo: ccsl.ime.usp.br/en/node/429

1 Motivação

O protocolo de transporte TCP tem sido hegemônico nas aplicações de internet, contudo sua definição pela RFC 793 [1] publicada em 1981 tem se tornado obsoleta para a estrutura de redes contemporânea, uma vez que seu foco, à época, trava-se de conexões cabeadas. Este tipo de conexão torna-se cada vez menos frequente devido à ubiquidade dos dispositivos móveis e aplicações de Internet das Coisas.

Apesar de existirem constantes propostas para a melhoria do protocolo, como o TCP Hybla (2002) [2], TCP Westwood (2008) [3] e TCP NewReno (2011) [4], a implementação dessas atualizações é difícil, dado que a implementação do protocolo é realizada diretamente no sistema operacional e para que fossem obtidos resultados significativos e para que ocorresse a disseminação destas modificações, seria necessária um enorme esforço para a modificação de softwares legados e de estrutura complexa. Estes fatores tornam impossível uma atualização frequente dos algoritmos que compõem o protocolo TCP para acompanhar as novas exigências da internet.

Essas limitações foram contornadas a partir da utilização do protocolo UDP na camada de transporte como base para a elaboração de protocolos mais avançados, os quais poderiam ser processados na camada de aplicação, tornando adaptações constantes uma possibilidade viável e, ao mesmo tempo, garantindo as qualidades de conexão características do protocolo TCP, as quais agora passam a ser avaliadas pela camada de aplicação. Dentre os protocolos criados com esse objetivo destaca-se o QUIC (Quick UDP Internet Connections), criado pela Google e voltado diretamente para melhorar a experiência em seu navegador Google Chrome.

O nsQUIC é um módulo de extensão ao simulador de redes NS-3 que permite avaliar o desempenho do protocolo em um ambiente controlado, uma vez que

os resultados disponibilizados pela empresa não possuem um detalhamento do ambiente e das condições em que ocorreram os testes. Além disso, o simulador permite obter resultados sem a necessidade de uma infraestrutura complexa e altamente custosa como a da companhia que produz o protocolo. O código do módulo está em código aberto sob a licença GNU-GPLv2 [5].

Resultados divulgados pela Google em 2015 mostram que o QUIC já era responsável por cerca de 7% do tráfego da Internet, promovendo um aumento significativo na performance, em especial para os casos de streaming de vídeos, onde reduziu em 30% o número de rebuffers necessários no site da empresa, Youtube.

2 nsQUIC

O nsQUIC utiliza a versão 41 do protocolo, bem como a versão 3.27 do NS-3.

O protocolo possui determinadas funcionalidades que aumentam sua performance, como: a conexão em 0-RTT, a qual permite que um servidor possa iniciar uma nova conexão sem necessidade do handshake característico do TCP, caso já tenha estabelecido uma conexão prévia com o cliente; a multiplexação, a qual permite a divisão de pacotes em diferentes streams lógicas, não permitindo que um pacote trave toda a cadeia de pacotes sequentes (head of line blocking); e um controle de congestionamento utilizado para uma melhor estimativa inicial do Round-trip Time e largura de banda, bem como uma melhor sinalização de ACKs [6].

As diferentes ferramentas de compilação de cada projeto, Waf no caso do NS-3, Gn e Ninja para o QUIC, geraram problemas na incorporação dos dois códigos. Estes problemas foram contornados a partir da compilação utilizando diretivas que trocavam a saída do préprocessamento por regras de descrição das dependências do código fonte (g++ -M e g++ -MM) [7].

O segundo passo de adaptação foi trocar a biblioteca de sockets utilizada pelo protocolo, o qual estava programado para fazer uso dos sockets UDP do sistema operacional (Berkeley Sockets - sys/socket.h). As funções que faziam uso desta biblioteca foram então reimplementadas utilizando os sockets definidos para o NS-3.

O paradigma de simulação baseada em eventos do NS-3 conflitava com a implementação para tempo real do QUIC, portanto o código de cliente e servidor do QUIC foi modificado, adotando o esquema de eventos.

O nsQUIC permite a utilização de todos os recursos de simulação já presentes no NS-3, adicionando as peculiaridades para a utilização do protocolo QUIC.

3 Comparações com o TCP

Foram realizadas simulações para comparar o QUIC com as versões Hybla e NewReno do TCP.

Em uma topologia simples, realizando uma transferência de 100MB do servidor para o cliente, com um atraso de 30ms e capacidade de transmissão de 20Mbps, supondo uma conexão "ótima" (sem perda de pacotes), o QUIC apresentou um desempenho um pouco pior que o TCP para poucos dados (1KB), contudo, devido ao seu algoritmo de controle congestionamento mais agressivo, para transferência de 10KB à 1MB o desempenho foi significativamente melhor. De 10MB até 1GB não houve grande diferença entre os protocolos, uma vez que todos conseguiram atingir a largura máxima de banda disponível, sendo o QUIC levemente mais rápido devido à sua vantagem no início da transferência.

Na topologia Dumbbell foram postos dois roteadores conectando os clientes e servidores, o que pode causar perda de pacotes em um cenário real por limitações de buffer dos roteadores. O servidor envia 100MB de dados para o respectivo cliente, sendo a conexão entre nós e roteadores de 10Mbps e atraso de 50ms, e entre roteadores de 10Mbps e atraso de 75ms. O desempenho do QUIC neste cenário mais complexo mostra-se muito superior às duas versões do TCP quando submetidos a uma variação dos atrasos de conexão. Torna-se evidente que o QUIC possui uma melhor estimativa do Round-Trip Time e largura de banda.

4 Referências

1. TRANSMISSION CONTROL PROTOCOL (tools.ietf.org/html/rfc793)
2. TCP Hybla: a TCP Enhancement for Heterogeneous Networks
3. TCPWestwood: End-to-End Congestion Control for Wired/Wireless Networks
4. The NewReno Modification to TCP's Fast Recovery Algorithm (tools.ietf.org/html/rfc3782)
5. Repositório nsQUIC no GitLab (gitlab.com/diegoamc/ns-3-quic-module)
6. QUIC, a multiplexed stream transport over UDP (chromium.org/quic)
7. Manpage g++ (linux.die.net/man/1/g++)