Universidade Tecnológica Federal do Paraná – Câmpus Cornélio Procópio

Engenharia de Software

ES63H – Redes de Computadores

Profa. Natássya B. F. da Silva

Implementação de entrega ordenada para o protocolo UDP

Artur Massaro Gonzaga

Leonardo de Oliveira Flores

Luiz Henrique Barros Teixeira

Cornélio Procópio, PR

07 de Outubro de 2018

Artur Massaro Gonzaga
Leonardo de Oliveira Flores
Luiz Henrique Barros Teixeira

Implementação de entrega ordenada para o protocolo UDP

Relatório técnico apresentado como parte dos requisitos de avaliação da disciplina de Redes de Computadores.

Universidade Tecnológica Federal do Paraná – Câmpus Cornélio Procópio

Engenharia de Software

ES63H – Redes de Computadores

Profa. Natássya B. F. da Silva

Cornélio Procópio, PR

07 de Outubro de 2018

Resumo

O uso de redes de comunicação no século XXI é indispensável, visto que seu vasto uso no dia a dia tanto na área comercial, como em bancos, companhias aéreas e hotéis, mas também no uso pessoal, como em smartphones, tablets e notebooks, esconde por trás de toda a facilidade para o uso destas redes uma série de protocolos que garantem a sua usabilidade. Atualmente há dois principais protocolos de comunicação, sendo eles o TCP e UDP.O TCP se trata de um protocolo mais completo, com mais funcionalidades enquanto o UDP,um protocolo para comunicação rápida,o que gera eventualmente a necessidade de se implementar funcionalidades a mais, sendo exemplificado neste trabalho a implementação e a necessidade de entrega ordenada no protocolo UDP.

Palavras-chaves: Protocolo de comunicação. UDP. Entrega ordenada.

Lista de abreviaturas e siglas

UDP	User Datagram Protocol
TCP	Transmission Control Protocol
RTP	Real Time Protocol
RTCP	Real-Time Transport Control Protocol

Sumário

1 HISTÓRICO

A princípio, o protocolo TCP descrito no trabalho original de "Vint Cerf e Bob Kahn(1974)[UDP,]", deveria ser capaz de prover comunicação e transporte de pacotes por toda a internet, havendo desde a entrega sequenciada de dados até o serviço de datagrama por meio do uso direto do serviço básico de rede, o que ocasionaria em eventuais perdas, corrompimento e reordenação de pacotes. Devido aos problemas ocasionais e baixo desempenho para grandes pacotes de dados, houve a necessidade da criação de um protocolo simplificado. Este protocolo haveria apenas o endereçamento e roteamento de pacotes, sendo este um protocolo eficiente e simples. A partir da utilização, permite a personalização e implementação do serviço de acordo com as necessidades do usuário. Esse protocolo recebeu o nome de "*User Datagram Protocol*" ou simplesmente UDP.

2 INTRODUÇÃO

Pela simplicidade do protocolo UDP, ele conta com uma taxa de transferência de dados superior ao TCP, o que é um grande atrativo para diversos serviços, principalmente os que precisam transmitir grandes pacotes de dados no menor tempo possível, como é o caso de serviços de streaming ou jogos on-line.

No envio de dados pela Internet, é comum a divisão de grandes dados em pequenos pacotes para a otimização do envio, porém muitas vezes esses pacotes são enviados e recebidos em ordens diferentes, o que pode ocasionar em falhas e corrompimento dos dados caso não haja um tratamento apropriado para corrigir este tipo de falha.

2.1 OBJETIVO

É necessário a implementação de um serviço de entrega ordenada para garantir a qualidade do sistema, assim evitar os problemas com arquivos corrompidos ou com falhas em qualquer tipo de serviço. Este documento tem como objetivo apresentar soluções de implementação de entrega ordenada com o uso do protocolo UDP.

3 APLICAÇÕES

Como já citado anteriormente, o protocolo UDP se comparado com o TCP, nota-se uma disparidade grande em desempenho e devido a este fator, ele é comumente encontrado em serviços de transmissão de dados (streaming), como Streaming de vídeo, Jogos multiplayer e Voip [ITSTILLWORKS,].

Apesar dos serviços citados acima necessitarem de um protocolo de transmissão que seja eficiente em velocidade, há também a necessidade de qualidade da transmissão, e levando em consideração a simplicidade do UDP, estas empresas optam pela utilização deste, porém para suprir a demanda, algumas funcionalidades "extras" devem ser implementadas.

Diversos serviços extras podem ser implementados no protocolo UDP conforme a necessidade de cada empresa ou usuário, porém uma função frequentemente utilizada é a ordenação de pacotes, visto que há a necessidade de que cada pacote enviado seja entregue na ordem em que é enviado, para, por exemplo, evitar quadros dessincronizados durante um streaming de vídeo.

4 VANTAGENS E DESVANTAGENS

O protocolo de comunicação UDP, como já citado anteriormente, possui uma superioridade notória em relação ao desempenho na entrega de pacotes, devido à sua simplicidade, muitas vezes é necessário a implementação de serviços extras para a adequação ao caso de uso, Nestes casos podem ser implementados alem da entrega ordenada, serviços como RTCP sendo assim, há um gasto extra em tempo de desenvolvimento e complexidade do projeto.

Exemplo de serviço é o uso de "*Real Time Protocol*" ou RTP, que possibilita a transmissão de dados em tempo real, garantindo sua entrega ordena, é comumente utilizado em junção com o RTCP, para monitoramento da rede, podendo assim adequar a taxa de transmissão para cada cliente, garantindo um streaming de dados mais fluida.

As desvantagens são a necessidade dessa implementação de serviços extras para que não hajam erros na transmissão de dados, como o corrompimento de um pacote ou a simples ineficiência da transmissão de dados, já que o propósito principal do protocolo é a sua simplicidade enquanto um processo efetivamente rápido.

5 TÉCNICAS E FERRAMENTAS DE IMPLEMENTAÇÃO

5.1 Implementação original

A implementação do projeto será feita em duas partes, sendo elas: cliente e servidor. A primeira tem como função a separação dos pacotes, neste caso para que seja feita a implementação da ordenação de pacotes será necessário que o servidor separe os dados em pequenas partes(Pacotes) de 32 bits, os quais 8bits será destinado ao funcionamento do sistema e o restante destinado à mensagem e enviar separadamente cada uma ao servidor, que ao receber irá concatenar com o pacote anterior e enviar um mensagem de confirmação, permitindo assim o envio de um novo pacote, até que o cliente envie uma mensagem final sinalizando que todos os pacotes já foram enviados.

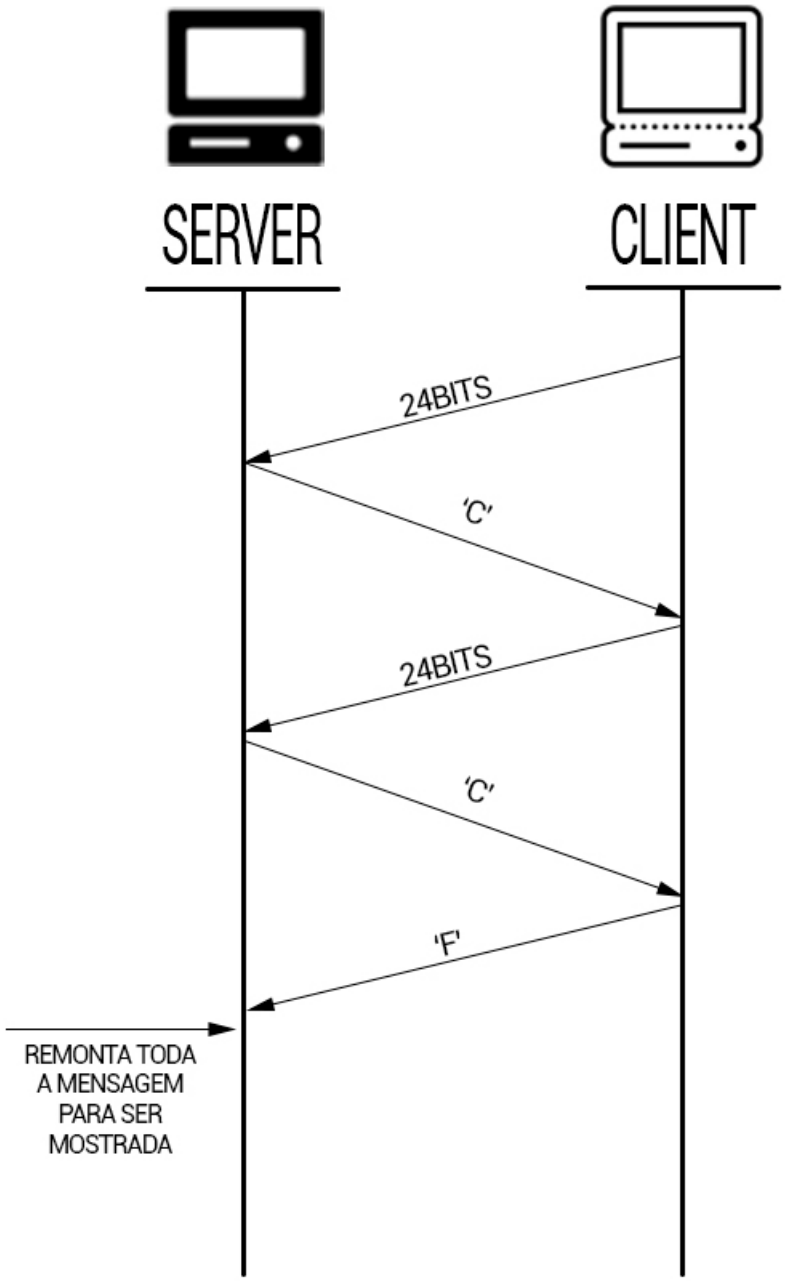

Figura 1: O diagrama acima ilustra o funcionamento do sistema

Será usado para a comunicação o serviço de sockets que tem a função de prover a comunicação entre dois ou mais aplicativos, sendo possível assim a implementação do protocolo UDP.

Este será desenvolvido em ambiente linux utilizando a linguagem C e para que seja possível a utilização das funções de sockets será usado a biblioteca socket.h, nativa do linux. Para minimizar exerforços foi itiluzado como base o codigo retirado de "UDP...,

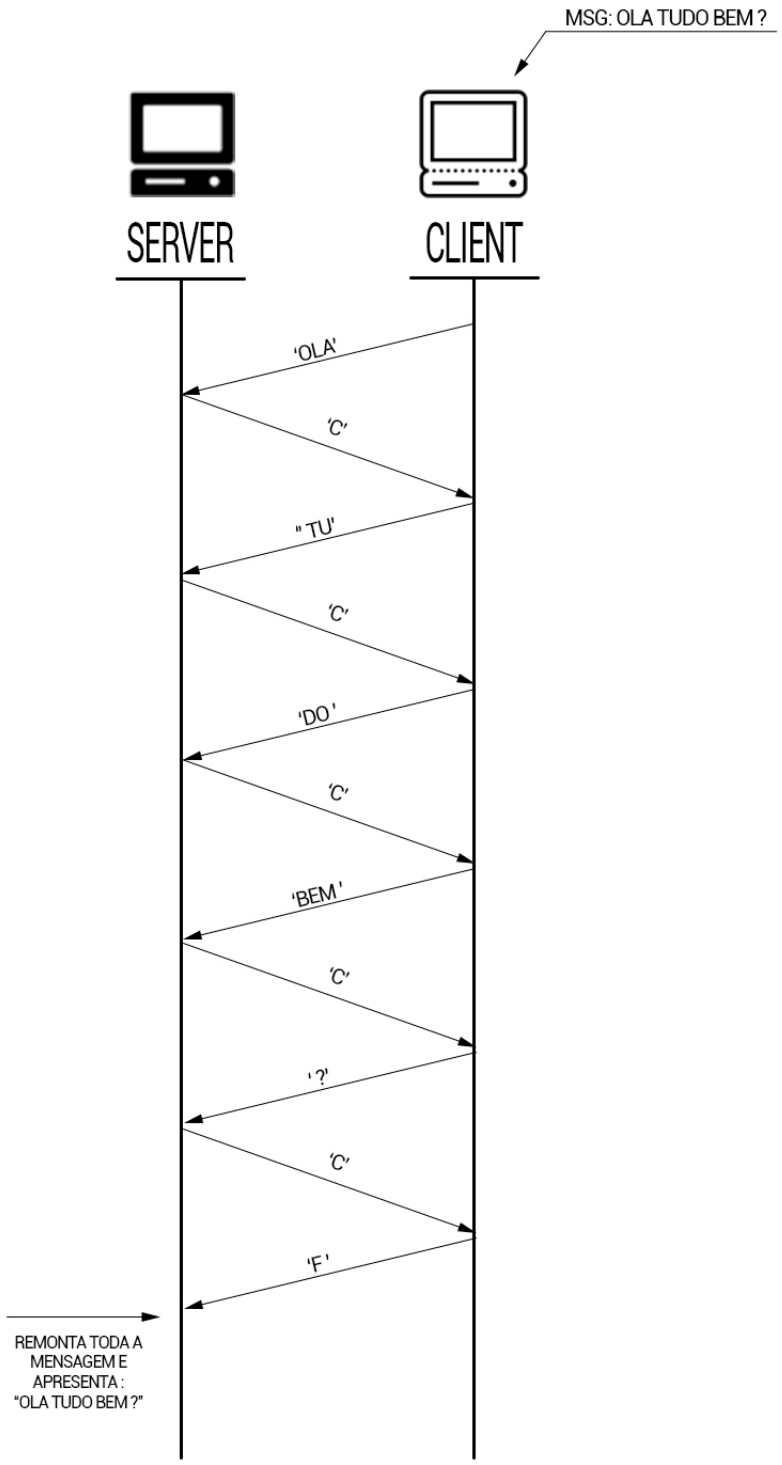

Figura 2: O diagrama acima exemplifica o funcionamento do sistema

5.1.1 Tratamento de erros

Caso a conexão seja perdida ou haja outros problemas com a comunicação no momento da troca de pacotes a mensagem ficara incompleta, sendo assim há a necessidade de se implementar parte do código para este determinado fim.

Seu funcionamento será feita da seguinte forma: ao separar o dado a ser enviado em pequenos pacotes de 32 bits, parte deste pacote(8Bits iniciais) serão destinados ao envio de um carácter de confirmação, o qual ao ser recebido pelo servidor verificara qual ação a ser tomada.

Neste programa o carácter inicial por ser de 3 opções, 'c','r','f', o qual serão responsáveis por tratar o pacote em caso de confirmação, reenvio e finalização, respectivamente.

Ao enviar um pacote o cliente aguarda uma resposta do servidor, caso este exceda o seu tempo limite será considerado como perda de pacote, fazendo assim com que o ultimo pacote seja enviado novamente ao servidor, caso haja perda de mais de 10 pacotes a conexão será encerrada, evitando assim o envio exagerado de pacotes.

Ao receber um pacote repetido o servidor confere se aquele já foi adicionado a mensagem integral, caso tenha sido é desconsiderado caso contrario é adicionado à mensagem.

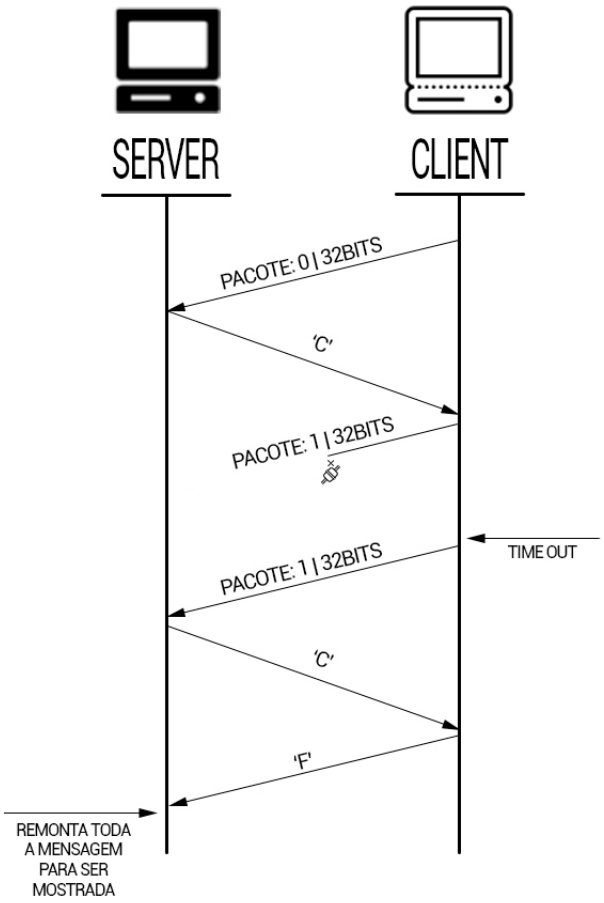

Figura 3: O diagrama acima exemplifica o funcionamento do sistema em caso de perda do pacote antes de chegar ao servidor

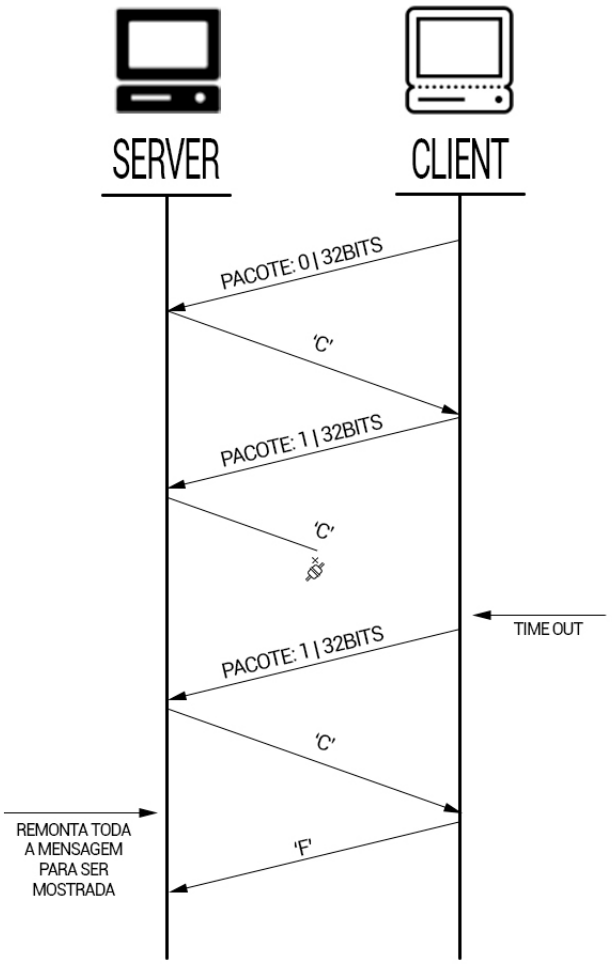

Figura 4: O diagrama acima exemplifica o funcionamento do sistema em caso de perda do pacote após chegar ao servidor

5.2 Implementação Extra

Este projeto foi desenvolvido no intuito de exemplificar o funcionamento e utilização de um servidor UDP, com a funcionalidade de entrega ordenada implementada em RTP.

Foi utilizado para o desenvolvimento o ambiente linux, instalado um servidor Wowza, em suas configurações padrão que, através da interface UDP com RTP, possibilita a execução do streaming de vídeo on demand.

Para facilitar a utilização e apresentação, foram desenvolvidas duas aplicações:

uma desktop em Java, que ao ser executada inicia o servidor e possibilita ao usuário a escolha de qual arquivo deseja fazer o streaming, onde após a escolha, o arquivo é movido para a pasta padrão do servidor e seu nome é alterado para o padrão utilizado em nossa aplicação.

A outra parte da aplicação é a interface mobile, onde apenas um botão habilita e inicia a transmissão do conteúdo disponível no servidor via RTP/RTSP através de seu endereço ip, o qual por padrão está configurado para 192.168.100.17, porta: 1935.

Referências

ENSURING packet order in UDP. https://stackoverflow.com/questions/3745115/ensuring-packet-order-in-udp. Accessed: 2018-10-07. Nenhuma citação no texto.

ITSTILLWORKS. https://itstillworks.com/applications-use-udp-protocol-7434472.html. Accessed: 2018-12-04. Citado na página 7.

POSTEL, J. *User datagram protocol*. [S.l.], 1980. Nenhuma citação no texto.

PROTOCOLO TCP e Protocolo UDP. http://mikeicorli.blogspot.com/2012/05/normal-0-21-false-false-false-pt-x-none.html. Accessed: 2018-10-07. Nenhuma citação no texto.

PROTOCOLOS udp e tcp em redes ad-hoc. http://www.teleco.com.br/tutoriais/tutorialprotocolo/pagina_3.asp. Accessed: 2018-10-07. Nenhuma citação no texto.

REDES de computadores/Multiplexação e demultiplexação. https://pt.wikibooks.org/wiki/Redes_de_computadores/Multiplexaç~ao_e_demultiplexaç~ao. Accessed: 2018-10-07. Nenhuma citação no texto.

SOCKETS em C (WinSock). http://www.professordiovani.com.br/sd/Sockets_cplus.htm. Accessed: 2018-10-07. Nenhuma citação no texto.

SYSTEM Calls or Bust. https://www.gta.ufrj.br/ensino/eel878/sockets/syscalls.html#socket. Accessed: 2018-10-07. Nenhuma citação no texto.

TANENBAUM, A. S. *Redes de computadores*. 5. ed. [S.l.]: Pearson. Nenhuma citação no texto.

TRANSMISSION Control Protocol. https://pt.wikipedia.org/wiki/Transmission_Control_Protocol. Accessed: 2018-10-07. Nenhuma citação no texto.

UDP. https://pt.wikipedia.org/wiki/User_Datagram_Protocol. Accessed: 2018-10-07. Citado na página 5.

"UDP Server-Client implementation in C". https://www.geeksforgeeks.org/udp-server-client-implementation-c/. Accessed: 2018-12-03. Citado na página 11.

UDP Server-Client implementation in C. https://www.geeksforgeeks.org/udp-server-client-implementation-c/. Accessed: 2018-10-07. Nenhuma citação no texto.