

CAMADA FÍSICA DA COMPUTAÇÃO

ENGENHARIA DA COMPUTAÇÃO - Rodrigo Carareto

PROJETO 3 – Fragmentação, hand-shake e datagrama

Nesse projeto, sua aplicação que exerce o papel de client deverá enviar um arquivo para a aplicação server. Esse arquivo deverá ser fragmentado e enviado através de “pacotes” (datagramas), handshake e confirmação de recebimento !

Datagrama.

De agora em diante você está **PROIBIDO** de trocar mensagens entre server e client que não sejam um datagrama completo (um pacote). Isso significa que mesmo que queira enviar um único byte, deverá enviar um pacote compondo um datagrama. Para isso vamos considerar o seguinte datagrama:

- HEAD – 12 BYTES - fixo
- PAYLOAD – variável entre 0 e 50 BYTES (pode variar de pacote para pacote)
- EOP – 3 BYTES – fixo (valores de sua livre escolha)

Não importa o tipo de mensagem, sempre será enviada através de um datagrama como definido acima. Tanto de client para server como no sentido oposto.

Handshake

Antes do início do envio da mensagem, o client deve enviar uma mensagem para verificar se o server está “vivo”, pronto para receber o arquivo a ser enviado. O server então deve responder como uma mensagem informando que ele está pronto para receber. Enquanto a mensagem não for recebida pelo cliente, este não começa o envio. Caso o cliente não receba a resposta do servidor dentro de 5 segundos, informando que está pronto para receber o arquivo, o usuário recebe uma mensagem: “Servidor inativo. Tentar novamente? S/N”. Se o usuário escolher “S”, outra mensagem de verificação é enviada ao server. Caso escolha não. Tudo se encerra.

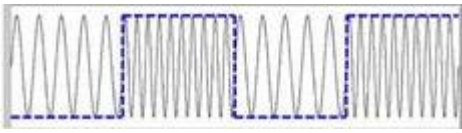
Caso o servidor responda ao cliente em menos de 5 segundos, o cliente deve iniciar a transmissão do arquivo com o envio do primeiro pacote.

Fragmentação

Como seu payload é menor que o arquivo a ser enviado, você deverá enviá-lo em partes (pacotes)! Lembre-se que cada pacote tem que seguir obrigatoriamente a estrutura de seu datagrama! A razão para se dividir uma mensagem em pacotes pode ser devido a uma limitação de hardware (pouco espaço no buffer, por exemplo), ou ainda gestão do tempo de ocupação do canal de comunicação, evitando-se manter a linha ocupada por muito tempo. Outra razão é a de se evitar ter que retransmitir uma mensagem inteira quando acontecer um erro, podendo-se reenviar apenas o fragmento com problema.

Acknowledge / Not acknowlwdge

Durante a transmissão de dados é muito comum a troca de mensagens como confirmação de recebimento de um pacote ou mesmo informando um problema na recepção do pacote. Esse tipo de comunicação gera uma robustez para a transmissão, embora possa afetar a velocidade de transmissão. Existe então um compromisso entre a velocidade de transmissão e a segurança da transmissão no que diz respeito à integridade dos dados.



CAMADA FÍSICA DA COMPUTAÇÃO

ENGENHARIA DA COMPUTAÇÃO - Rodrigo Carareto

FUNCIONALIDADES:

- **Nesse projeto, seu datagrama não pode ultrapassar 65 bytes.** Consequentemente, como já dito antes, seu payload será menor que isso, exigindo uma fragmentação da imagem a ser feita pelo cliente.
- Quando o client enviar um pacote, deve informar obrigatoriamente (em algum espaço do head reservado a isso), o número do pacote e o número total de pacotes que serão transmitidos.
- Ao receber um pacote, o server deve fazer duas verificações: verificar que o número do pacote é 1 a mais que o anterior, ou seja, a ordem está correta. Deve também verificar se o EOP está no local correto (vieram todos os bytes).
- Se tudo estiver ok, o server deve enviar uma mensagem ao cliente para que este envie o próximo pacote.
- Após o envio de um pacote o cliente não envia o próximo enquanto não receber a confirmação do server de recebimento do pacote enviado.
- Se algo estiver errado, o server deve enviar uma mensagem para o cliente solicitando o reenvio do pacote, seja por não ter o payload esperado, ou por não ser o pacote correto.
- Ao receber o último pacote o Server deve ser capaz de reagrupá-los e salvar o arquivo em seu estado original. Após o reagrupamento, o server faz uma última resposta ao cliente, informando que a transmissão foi feita com sucesso.
- Quando houver algum erro de transmissão, o client não precisa corrigir o erro e continuar a transferência de dados. A comunicação deve se encerrar.

ENTREGA:

Você deverá mostrar até 14/09/2023

Conceito C

1Uma transmissão de sucesso com seu server recebendo os pacotes, conferindo e respondendo ao cliente.

Conceito B

- Uma situação onde o server não estava pronto para receber o arquivo por mais de 5 segundos, forçando o cliente a enviar uma outra mensagem de teste. Rode a aplicação do server depois que o cliente já tentou comunicação por umas 2 vezes e mostre que a transmissão se inicia.
- Uma simulação onde o client erre o número do pacote. Mostre a resposta do servidor perante o envio fora de ordem.
- Faça uma simulação onde o tamanho real do payload de um pacote não corresponde ao informado no head. Mostre a resposta do servidor.

Conceito A+

Uma simulação em que se os fios entre os Arduinos forem desconectados e conectados novamente, a transmissão retorna e termina com sucesso.