

## PROJETO 4 – HandShake

Nesse projeto você deverá modificar seu código de transmissão de dados de maneira a realizar uma comunicação com “handshake”.

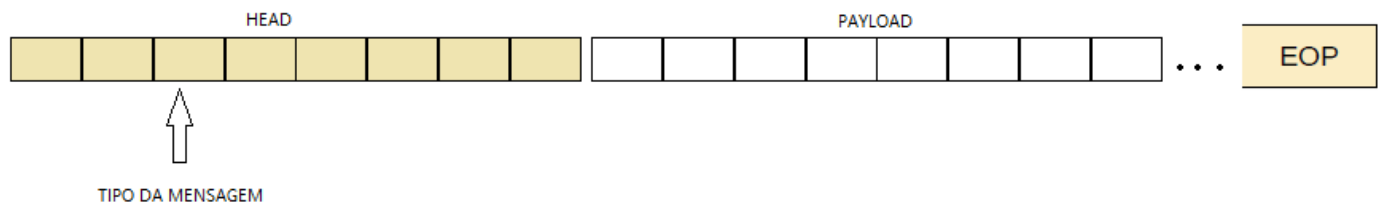
De agora em diante, o head implementado para o envio de um pacote deverá conter, além do número de bytes do payload, o tipo de mensagem enviada. Além disso, mensagens serão enviadas do cliente para o server e do server para o cliente! Durante uma transmissão de dados, 3 etapas irão ocorrer:

1. **Synch**: Client e server verificam se estão prontos para trocarem dados.
2. **Transmissão**. Onde os dados são enviados
3. **Encerramento**. Possibilidades de verificações finais e encerramento da conexão.

A seguir, a descrição de como deve funcionar cada uma das etapas.

### Etapa 1 – Synch

Vamos supor que você irá reservar o terceiro byte do seu head para guardar o tipo de mensagem (parte do seu protocolo). Lembre-se que todo envio de dados deve ser feito através de pacotes que você definiu! **NÃO EXISTIRÁ MAIS TRANSMISSÃO ALGUMA QUE NÃO SEJA FEITA COM PACOTES!**



- 1) Ao iniciar o processo de transmissão, o Client deverá enviar uma mensagem com o tipo 1. Isso indica que está querendo saber se o Server está “ouvindo”.
- 2) Ao receber uma mensagem e verificar que o tipo é igual a 1, o Server percebeu que tem um Client que pode se comunicar. Nesse caso, deve devolver uma mensagem tipo 2. Essa mensagem informará ao Client que o Server está ouvindo e que também pode enviar mensagens.
- 3) Nesse ponto, o Client já sabe que o Server pode ouvi-lo e que ele pode ouvir o Server. Entretanto, o Server ainda não sabe se o Client pode ouvi-lo. Dessa forma, faz-se necessário o envio de uma mensagem tipo 3, do Client para o Server. E assim, o Server passa a saber que foi ouvido pelo Client.

As repostas dadas, tanto por Cliente quanto Servers, para confirmação da capacidade de comunicação são chamadas de mensagens de *acknowledgement*.

As mensagens do tipo synch podem ser enviadas com payload nulo ou com um byte representativo 0x00.



# CAMADA FÍSICA DA COMPUTAÇÃO

ENGENHARIA DA COMPUTAÇÃO - Rodrigo Carareto – 0#07E2/02

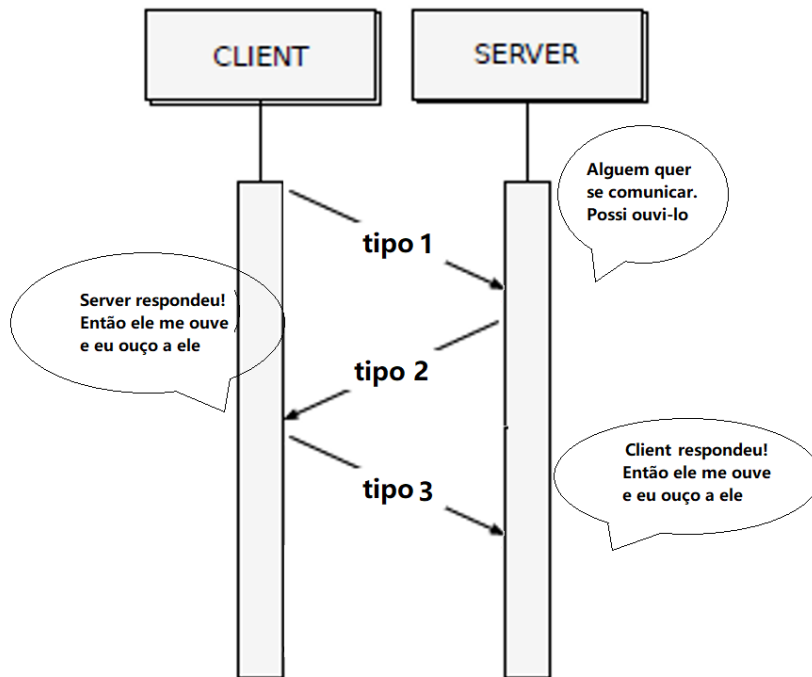


Figura 1 - Etapa Synch

## Tratamento de erro para a etapa 1: Synch

**Erro 1: Não recebimento da mensagem 2.** Ao enviar a mensagem tipo 1, o cliente tem 5 segundos (time out) para receber o tipo 2. Caso a resposta não ocorra, um aviso de erro deve ser mostrado em print e um novo envio da mensagem tipo 1 deve ser feito.

**Erro 2: Não recebimento da mensagem 3.** Ao enviar a mensagem tipo 2, o Server tem 5 segundos (time out) para receber o tipo 3. Caso isso não ocorra, um aviso de erro deve ser mostrado em print e um novo envio da mensagem tipo 2 deve ser feito.

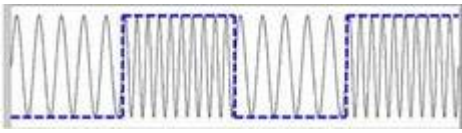
## Etapa 2: Transmissão de dados.

Por enquanto, estamos apenas enviando um único pacote. A etapa de transmissão, então, consiste na transmissão de um único pacote, do Client para o Server. Vamos definir que o pacote enviado com os dados será do **tipo 4**. Ao receber uma mensagem do tipo 4, o Server passa então a saber que se trata de um envio de dados, e não um synch.

Ao receber a mensagem tipo 4, o Server deve fazer as verificações de consistência. Ou seja, se o número de bytes está correto? (futuramente essas verificações serão implementadas com mais rigor).

Se tudo estiver ok, o server retorna uma mensagem de *acknowledgement* (*ack*) do tipo 5. Se houve problemas com o recebimento (número de bytes não confere) deve retornar uma mensagem de *not acknowledgement* (*nack*), que definimos como tipo 6. Nesse caso o Client deverá reenviar o pacote.

As mensagens do tipo ack e nack podem ser enviadas com payload nulo ou com um byte representativo 0x00.



# CAMADA FÍSICA DA COMPUTAÇÃO

ENGENHARIA DA COMPUTAÇÃO - Rodrigo Carareto – 0#07E2/02

## *Tratamento de erro para a etapa 2: transmissão*

**Erro 1: Não recebimento da mensagem 5 ou 6.** Ao enviar a mensagem tipo 4, o Cliente tem 5 segundos (time out) para receber o tipo 5 ou 6. Caso a resposta não ocorra, um aviso de erro deve ser mostrado em print e um novo envio da mensagem tipo 4 deve ser feito.

## Etapa 3: Encerramento.

Qualquer um dos nós (Client ou Server) pode solicitar o encerramento da transmissão (tenha ela sido feita com sucesso ou não). Espera-se que o Client solicite o encerramento apenas após transmitir com sucesso todos os dados (por enquanto, estamos apenas enviando um único pacote). Vamos definir que o pacote enviado para solicitação de encerramento será do tipo 7. Ao receber uma mensagem do tipo 7, o receptor deverá desconectar-se da comunicação estabelecida. O mesmo é feito pelo elemento que enviou o pacote tipo 7.

A mensagem do tipo encerramento pode ser enviada com payload nulo ou com um byte representativo 0x00.

## Avaliação

**Synch:** Funcionando corretamente: 4 pontos.

**Transmissão.** Funcionando corretamente: 4 pontos.

**Encerramento.** Funcionando corretamente: 2 pontos.