

Descrever o handshake implementado

Antes de começar a transferência de dados do pacote é necessário se certificar se o server e o client estão corretamente conectados, para tanto, foi implementado o handshake. Primeiro o client manda um sinal (SYN) para o server que já está previamente à espera do sinal. Em seguida, se o server receber o sinal, ele então, mandará a resposta para o client. Neste caso o sinal enviado é um ACK seguido de SYN. O último sinal é um ACK, que é enviado do client para o server e então os dados são transferidos. Caso o ocorra algum problema no pacote enviado, como tamanho errado do payload, ou pacote vazio, então é enviado o sinal NACK do server para o client.

Descrever os pacotes (SYN, ACK, NACK)

Os sinais SYN, ACK, NACK são formadas da seguinte forma:

| Constante | Tamanho | Tamanho | Comando | EOP |
|-----------|---------|---------|---------|-----|
|-----------|---------|---------|---------|-----|

Quando é mandado um comando, o tamanho do payload, formado pelo segundo e terceiro byte são zerados, uma vez que o tamanho do payload não importa ainda neste momento.

Já para o comando, decidiu-se que o SYN, ACK e o NACK seriam 0x10, 0x11, 0x12 respectivamente.

- SYN

| | | | | |
|------------|------|------|------|-----|
| 0xbb (cte) | 0x00 | 0x00 | 0x10 | EOP |
|------------|------|------|------|-----|

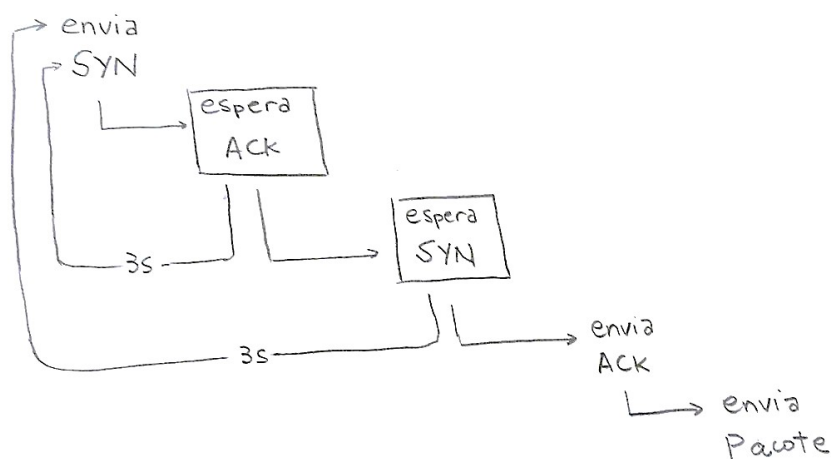
- ACK

| | | | | |
|------------|------|------|------|-----|
| 0xbb (cte) | 0x00 | 0x00 | 0x11 | EOP |
|------------|------|------|------|-----|

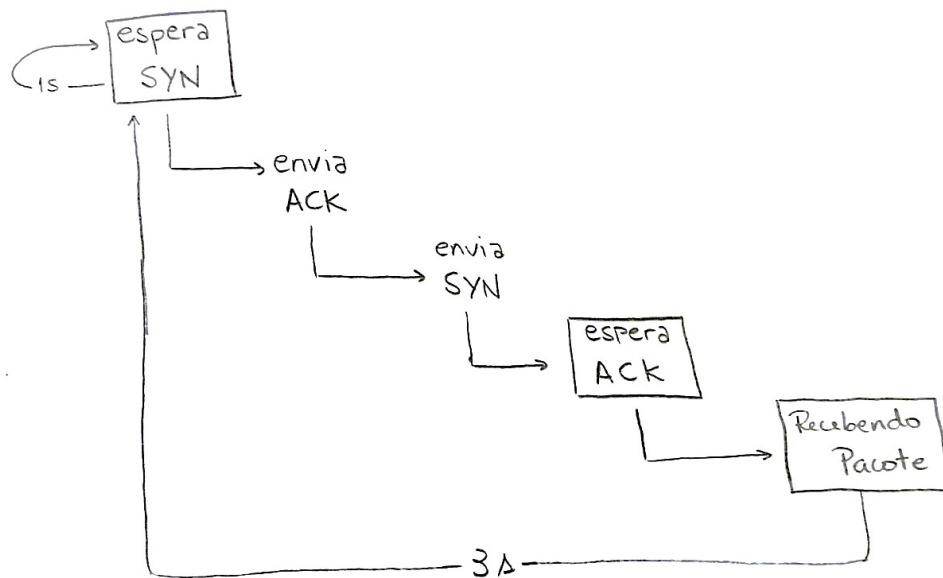
- NACK

| | | | | |
|------------|------|------|------|-----|
| 0xbb (cte) | 0x00 | 0x00 | 0x12 | EOP |
|------------|------|------|------|-----|

Diagrame o envio de pacotes em como uma máquina de estados



Diagrame a recepção de pacotes como uma máquina de estados



Como diferencia pacotes de comando (SYN, ACK, NACK) de pacote de dados

Os pacotes são diferenciados através do quarto byte do head nomeado de type.

Os pacotes de dados estão indicados pelo valor 0x00 enquanto os pacotes de comando estão indicados por 0x10, 0x11 e 0x12 sendo estes SYN, ACK e NACK respectivamente.