



REDES DE COMPUTADORES

Camada de Transporte

Transporte orientado para conexão

TCP - *Transmission Control Protocol*

- Para poder fornecer transferência confiável de dados, o TCP conta com os seguintes mecanismos.
 - ✓ **Soma de verificação** - Usada para detectar erros de bits em um pacote transmitido.
 - ✓ **Temporizador** - Usado para controlar a temporização/retransmissão de um pacote, possivelmente porque o pacote (ou seu ACK) foi perdido dentro do canal.



Transporte orientado para conexão

TCP - *Transmission Control Protocol*

- ✓ **Número de sequência** - Usado para numeração sequencial de pacotes de dados que transitam do remetente ao destinatário.
- ✓ **Reconhecimento** - Usado pelo destinatário para avisar o remetente que um pacote ou conjunto de pacotes foi recebido corretamente.
- ✓ **Reconhecimento negativo** - Usado pelo destinatário para avisar o remetente que um pacote não foi recebido corretamente.
- ✓ **Janela, paralelismo** - O remetente pode ficar restrito a enviar somente pacotes com números de sequência que caiam dentro de uma determinada faixa.



Transporte orientado para conexão

A conexão TCP

- Dizemos que o TCP é **orientado para conexão** porque, antes que um processo de aplicação possa começar a enviar dados a outro, os dois processos precisam primeiro se "apresentar".
- Isto é, devem enviar alguns segmentos preliminares um ao outro para estabelecer os parâmetros da transferência de dados.
- A "conexão" TCP não é um circuito TDM ou FDM fim a fim, como acontece em uma rede de comutação de circuitos, pois o estado de conexão reside inteiramente nos dois sistemas finais e não nos elementos intermediários da rede.



Transporte orientado para conexão

A conexão TCP

- Uma conexão TCP provê um **serviço full-duplex**.
- A conexão TCP é sempre **ponto a ponto**.
- Uma vez estabelecida uma conexão TCP, dois processos de aplicação podem enviar dados um para o outro.
- O TCP combina cada porção de dados do cliente com um cabeçalho TCP, formando, assim, **segmentos TCP**.



Transporte orientado para conexão

Estabelecimento da conexão TCP

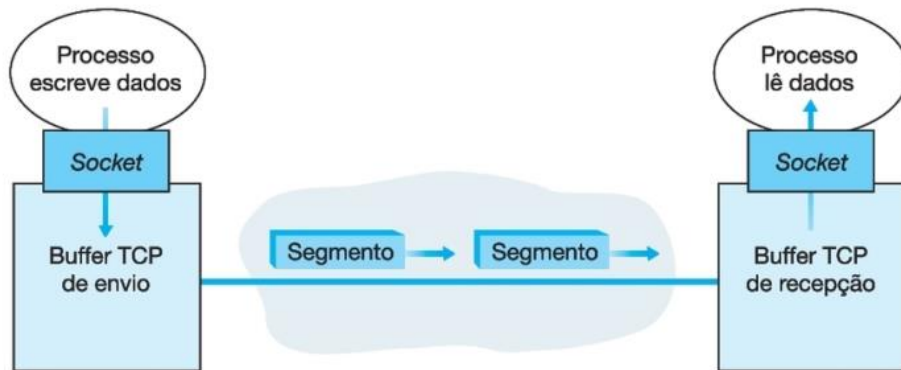
- O host cliente primeiro envia um segmento TCP especial chamado **CONNECTION REQUEST**;
- O servidor responde com um segundo segmento TCP especial, chamado **ACK**;
- O cliente responde novamente com um terceiro segmento especial.
- Como três segmentos são enviados entre dois hosts, esse procedimento de estabelecimento de conexão é muitas vezes denominado a **apresentação de três vias (3-way *handshake*)**.



Transporte orientado para conexão

Estabelecimento da conexão TCP

- Uma vez estabelecida uma conexão TCP, os dois processos podem enviar dados um para o outro.
- O processo cliente passa uma cadeia de dados pelo socket (a porta do processo). O TCP direciona seus dados para o buffer de envio da conexão.



Transporte orientado para conexão

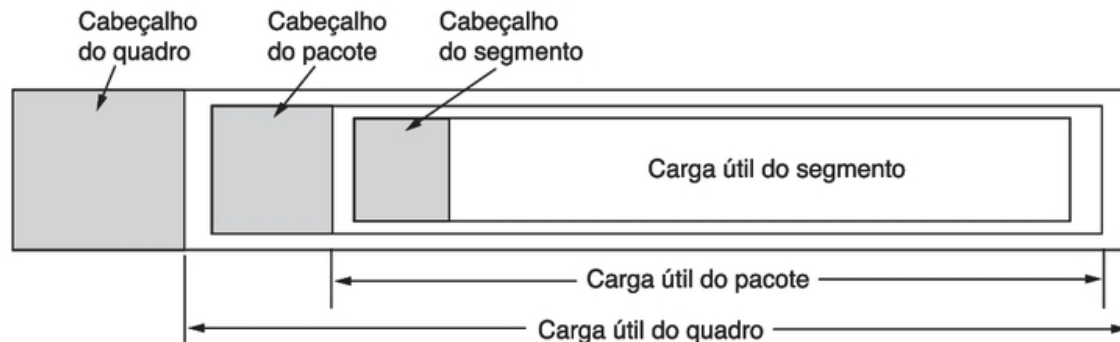
Estabelecimento da conexão TCP

- O TCP retira pedaços de dados do buffer de envio e passa os dados à camada de rede.
- A quantidade máxima de dados que pode ser retirada e colocada em um segmento é limitada pelo **tamanho máximo do segmento** (*maximum segment size* — **MSS**). Que normalmente é menor que 1500 bytes.

Transporte orientado para conexão

Estabelecimento da conexão TCP

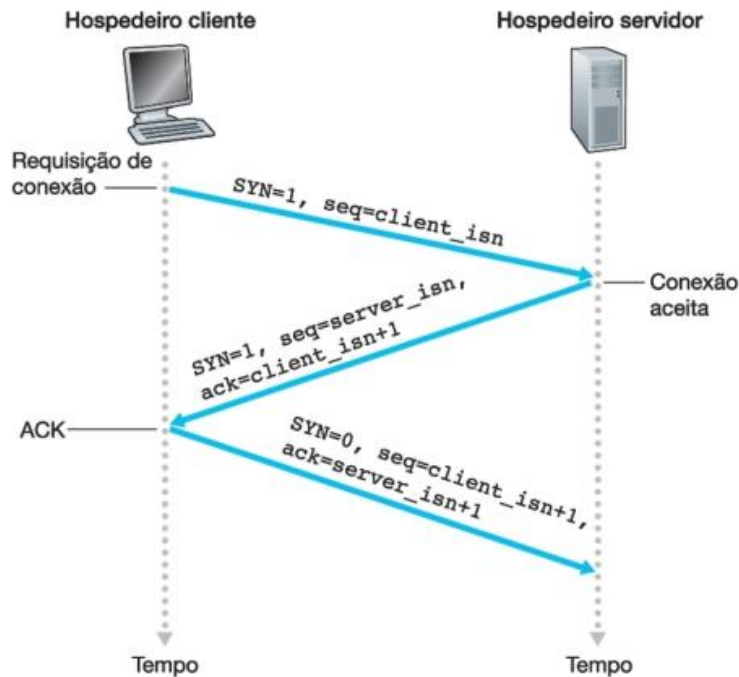
- O TCP combina cada porção de dados com um cabeçalho TCP, formando, assim os segmentos TCP.
- Os segmentos são passados para baixo, para a camada de rede, onde são encapsulados separadamente dentro dos datagramas IP.



Transporte orientado para conexão

10

Campos para estabelecimento da conexão TCP



Campos para estabelecimento da conexão TCP

- **Etapas 1.** O lado cliente primeiro envia um segmento especial ao lado servidor. Esse segmento não contém nenhum dado de camada de aplicação, mas um dos bits de flag no seu cabeçalho, o bit **SYN**, é ajustado para 1. O cliente escolhe ao acaso um número de sequência inicial e o coloca no campo de número de sequência do segmento inicial.
- **Etapas 2.** servidor extrai envia um segmento de aceitação de conexão ao TCP cliente. Esse segmento de aceitação de conexão também não contém nenhum dado de camada de aplicação. O segmento de concessão da conexão às vezes é denominado segmento **SYNACK**.

Campos para estabelecimento da conexão TCP

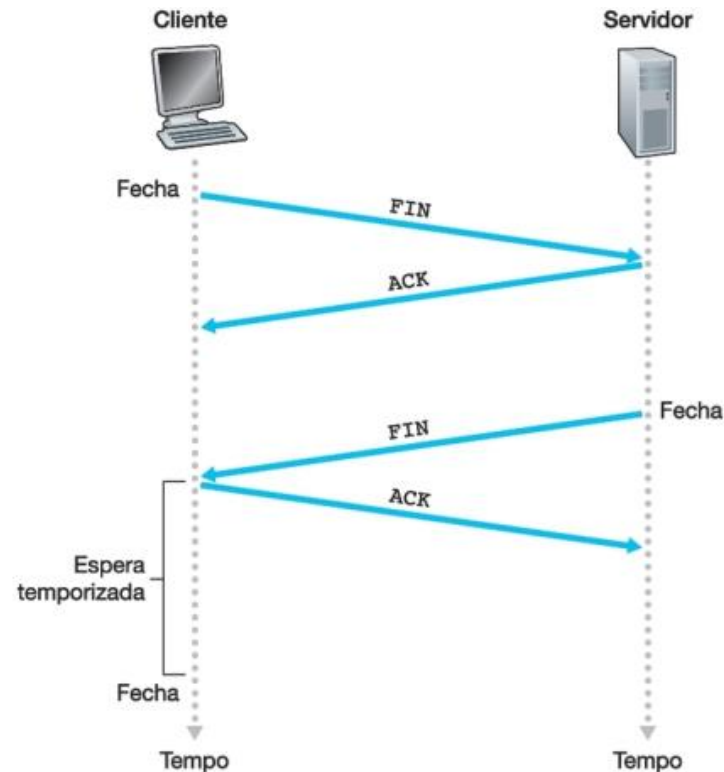
- **Etapa 3.** Ao receber o segmento SYNACK, o cliente envia ao servidor mais um segmento. Este último reconhece o segmento de confirmação da conexão do servidor
- O bit SYN é ajustado para 0, já que a conexão está estabelecida. A terceira etapa da apresentação de três vias pode conduzir os dados cliente/servidor na carga útil do segmento.

Transporte orientado para conexão

13

Encerramento da conexão TCP

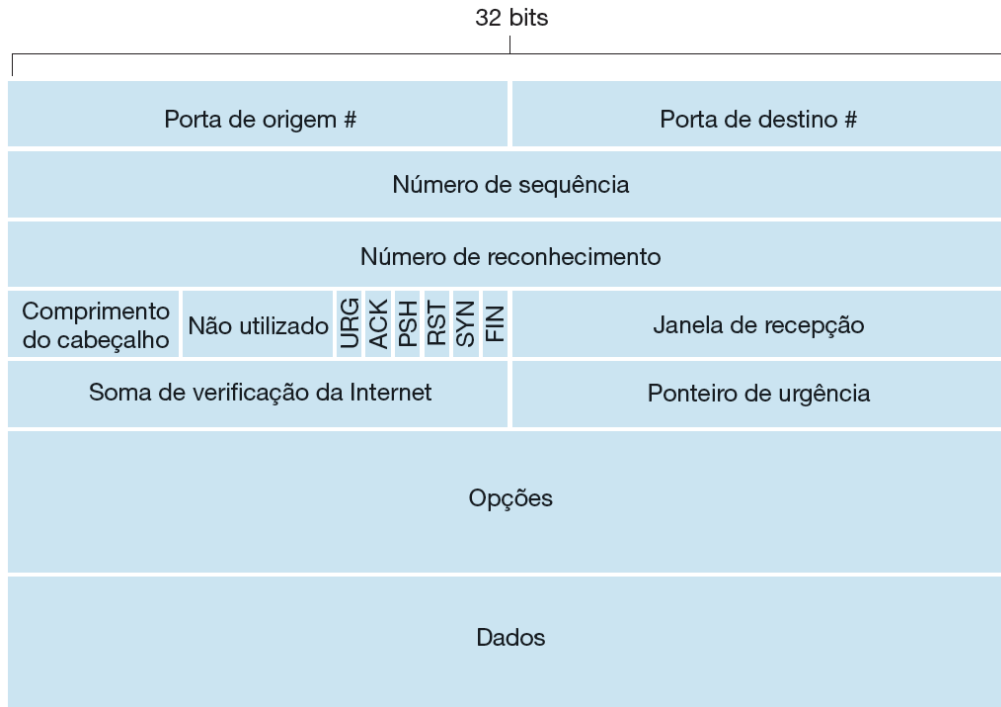
- O cliente enviar um segmento TCP com o bit $FIN = 1$. O cliente espera por um seguimento TCP do servidor.
- Quando recebe esse segmento, o TCP cliente ele espera por outro segmento do servidor com o bit $FIN = 1$.
- Após recebê-lo, o TCP cliente envia um ACK confirmando o encerramento.



Transporte orientado para conexão

14

Estrutura do segmento TCP



Estrutura do segmento TCP

- O cabeçalho TCP tem tipicamente **20 bytes** (12 bytes mais do que o cabeçalho UDP).
- O cabeçalho inclui os seguintes campos:
 1. Número da **porta de origem**, com 16 bits;
 2. Número da **porta de destino**, com 16 bits;
 3. O campo **Soma de verificação**, com 16 bits;
 4. O campo de **janela de recepção**, com de 16 bits: *é usado para controle de fluxo;*



Estrutura do segmento TCP

5. Campo de **comprimento de cabeçalho**, com 4 bits: especifica o comprimento do cabeçalho TCP.
- *O cabeçalho pode ter comprimento variável por causa do **campo de opções TCP**.*
 - *O campo de opções TCP em geral está vazio, de modo que o comprimento do cabeçalho TCP típico é 20 bytes.*
6. O **campo de opções**, de até 32 bits opcional e de comprimento variável: é usado quando um remetente e um destinatário negociam o MSS.



Estrutura do segmento TCP

7. O campo de **flag**, com 6 bits:

- *O bit **ACK** é usado para indicar se o segmento contém um reconhecimento para um segmento que foi recebido com sucesso.*
- *Os bits **RST**, **SYN** e **FIN** são usados para estabelecer e encerrar a conexão.*
- *Marcar o bit **PSH** indica que o destinatário deve passar os dados para a camada superior imediatamente.*
- *Por fim, o bit **URG** é usado para mostrar que há dados nesse segmento que a entidade da camada superior do lado remetente marcou como "urgentes".*

Estrutura do segmento TCP

- O campo **ponteiro de urgência**: O TCP deve informar à entidade da camada superior do lado destinatário quando existem dados urgentes e passar a ela um ponteiro para o final desses dados, indicando onde os dados urgentes terminam.

Observação Importante

- *Embora os campos **PSH,URG** e o ponteiro de dados urgentes sejam previstos, na prática, não são usados.*



Transporte orientado para conexão

Estrutura do segmento TCP

- Dois campos do cabeçalho são parte fundamental do serviço de transferência confiável de dados do TCP:
 - O número de sequência e
 - O número de reconhecimento
- O TCP vê os dados como uma cadeia de bytes não estruturada, mas ordenada.

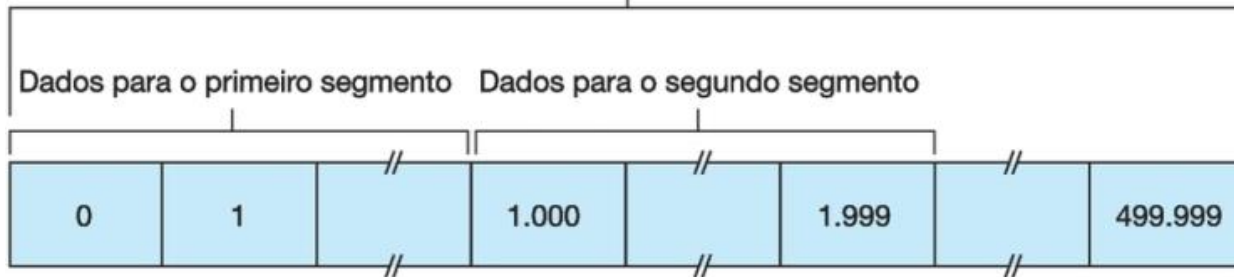
Transporte orientado para conexão

Estrutura do segmento TCP

- **O número de sequência**

Suponha que uma cadeia de dados a ser enviada consista em um arquivo composto de 500 mil bytes, que o MSS seja de 1.000 bytes. *Então o TCP constrói 500 segmentos a partir da cadeia de dados.:*

- *O primeiro recebe o número de sequência 0;*
- *O segundo, o número de sequência 1000;*
- *O terceiro, o número de sequência 2.000, e assim por diante.*



Transporte orientado para conexão

Estrutura do segmento TCP

- Na figura anterior, admitimos que o número de sequência inicial era 0.
- Na verdade, ambos os lados de uma conexão TCP escolhem ao acaso um número de sequência inicial.

Estrutura do segmento TCP

- O número de reconhecimento

Como o TCP é **full-duplex**, portanto o host A pode estar recebendo dados do host B enquanto envia dados ao host B.

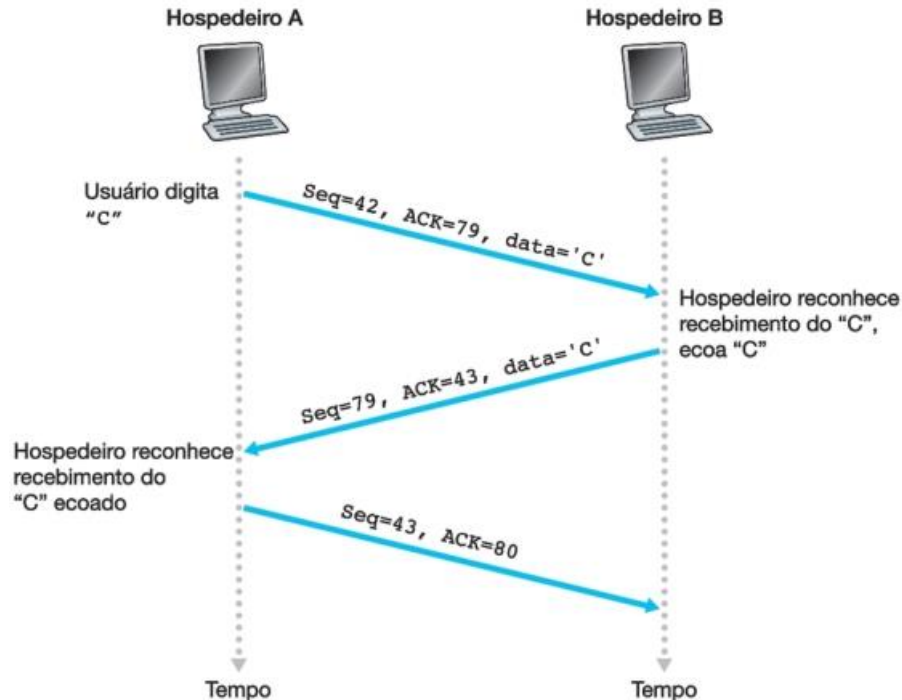
O número de reconhecimento que o host A atribui a seu segmento é o número de sequência do próximo byte que ele estiver aguardando do host B.

Transporte orientado para conexão

Estrutura do segmento TCP

Exemplo

Suponha que o host A tenha recebido do host B todos os bytes numerados de 0 a 535. Assim, ele coloca o número 536 no campo **de número de reconhecimento** do próximo segmento que envia para o host B.



Transporte orientado para conexão

Transferência confiável de dados do TCP

- O TCP cria um serviço de **transferência confiável de dados** sobre o serviço do IP.
- O serviço de transferência garante que a cadeia de bytes recebida é idêntica à cadeia de bytes enviada pelo sistema final que está do outro lado da conexão.

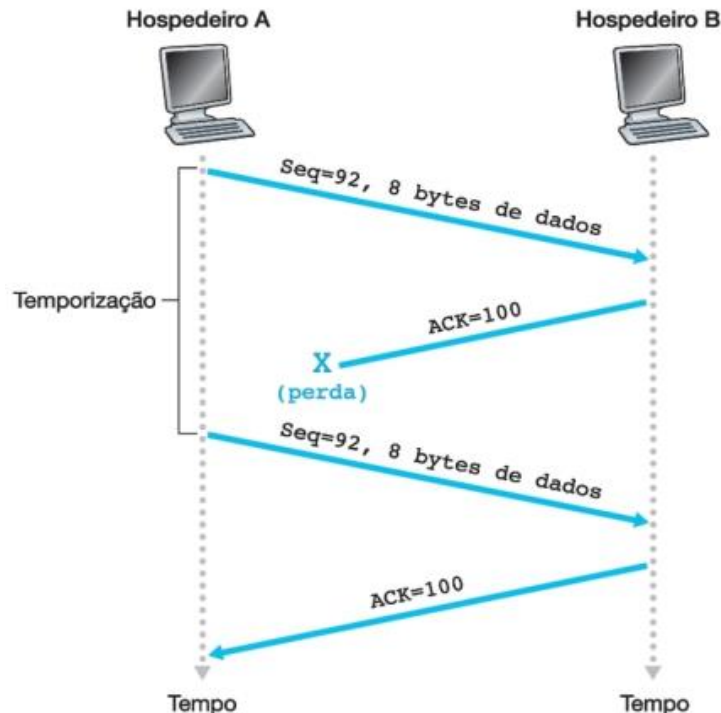
Transporte orientado para conexão

25

Transferência confiável de dados do TCP

Retransmissão devido a um reconhecimento perdido

- Um host A envia um segmento ao host B com numero de seq. 92 e 8 bytes de dados.
- Após enviá-lo, o host A espera por um segmento de B com número de reconhecimento 100.
- Embora o segmento de A seja recebido em B, o reconhecimento de B para A se perde.



Transporte orientado para conexão

Controle de fluxo do TCP

- O TCP provê um serviço de **controle de fluxo** às suas aplicações, para eliminar a possibilidade de o remetente estourar o buffer do destinatário.
- **Controle de fluxo** é um serviço de compatibilização de velocidades.
- O TCP oferece serviço de controle de fluxo fazendo que o remetente mantenha uma variável denominada janela de recepção.

Transporte orientado para conexão

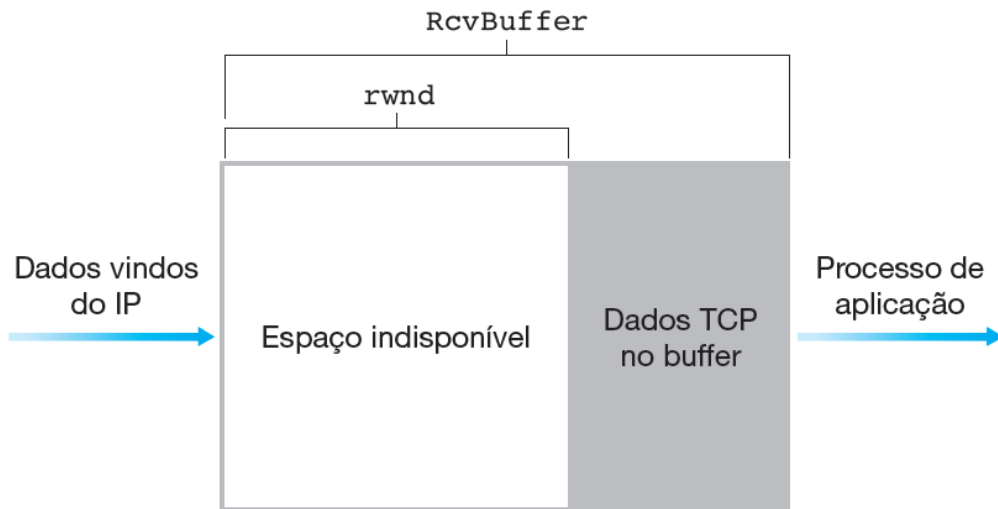
Controle de fluxo do TCP

- A janela de recepção é usada para dar ao remetente uma ideia do espaço de **buffer** livre disponível no destinatário.
- Como o TCP **full-duplex**, o remetente de cada lado da conexão mantém uma janela de recepção distinta.

Transporte orientado para conexão

Controle de fluxo do TCP

- O host receptor aloca um buffer de recepção para cada conexão



Transporte orientado para conexão

Controle de congestionamento TCP

- O TCP usa controle de congestionamento fim-a-fim.
- Isto significa que o remetente limita ou aumenta a taxa de entrega de dados para conexão em função do congestionamento percebido por ele, por isso dizemos que o TCP é **autoregulado**.
- Quando ocorre um evento de perda ou de três ACKs duplicados o remetente reduzirá sua taxa de transmissão à metade.

Transporte orientado para conexão

Controle de congestionamento TCP

- O TCP reconhece que não há congestionamento na rede quando recebe ACKs, então aumentará a taxa de transmissão lentamente a cada tempo de ida e volta.
- Esse comportamento de estar sempre aumentando a taxa de transmissão lentamente e depois reduzindo à metade bruscamente gera um comportamento parecido com dentes de serra, se visualizado graficamente.

