

Redes de Computadores

Prof. Robson de Souza

Aulas 21 e 22

Conteúdo: Camada de Transporte: Princípios da transferência confiável de dados.

A transferência confiável de dados é importante nas camadas de aplicação, transporte e enlace. As características dos canais não confiáveis determinarão a complexidade dos protocolos confiáveis de transferência de dados (rdt).

Transferência confiável sobre canais confiáveis → Em um canal de transferência perfeitamente confiável não há erros de bits e não há perda de pacotes. O transmissor envia os dados para o canal e o receptor lê os dados do canal.

Canal com erros de bit → Nesse caso, o canal pode trocar os valores dos bits num pacote. O checksum pode detectar esses erros. Para recuperar esses erros, podem ser utilizados:

- Reconhecimentos (ACKs): O receptor avisa explicitamente ao transmissor que o pacote foi recebido corretamente.
- Reconhecimentos negativos (NAKs): O receptor avisa explicitamente ao transmissor que o pacote tem erros. Quando o transmissor recebe um NAK, ele reenvia o pacote.

Pode ser que um ACK/NAK seja corrompido, nesse caso o transmissor não sabe o que aconteceu no receptor. O transmissor não pode simplesmente retransmitir, pois pode duplicar os dados.

Para tratar as duplicatas, o transmissor acrescenta um **número de sequência** em cada pacote. O transmissor reenvia o ultimo pacote no caso de um ACK/NAK ser perdido.

O receptor descarta (não passa para a camada de aplicação) pacotes duplicados.

Outro ponto é que o receptor pode não saber se seu último ACK/NAK foi recebido pelo transmissor.

Canais com erros e perdas → Nesse caso, além de trocar os valores dos bits, o canal de transmissão pode também perder pacotes (dados aos ACKs).

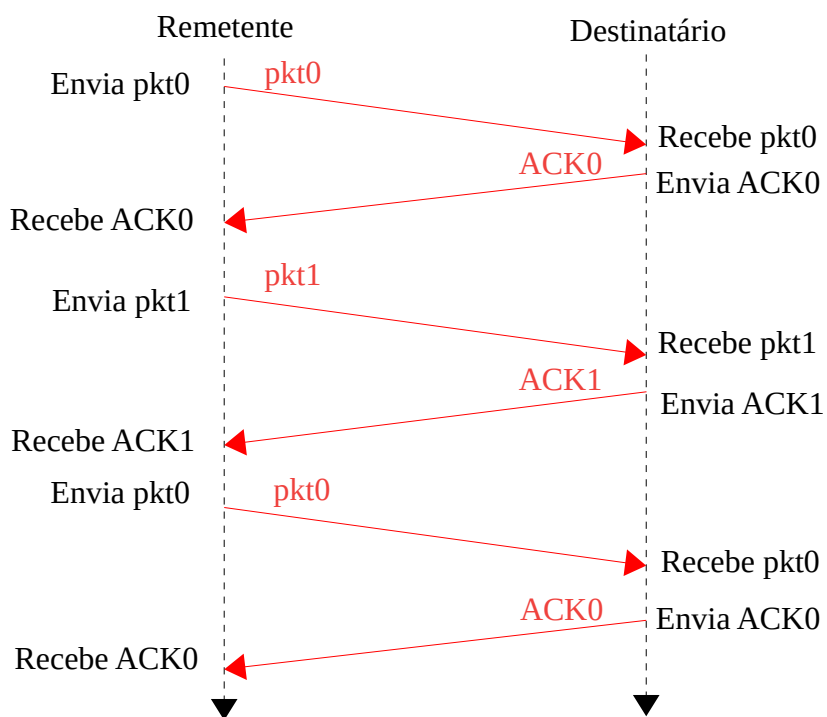
Nesse caso, números de sequência, ACKs, checksum e retransmissões podem ajudar, mas não resolvem o problema. O que pode ser feito nesse caso é fazer com que o transmissor espere um tempo “razoável” pelo ACK, se nenhum ACK for recebido nesse tempo, o transmissor retransmite.

Se o pacote ou ACK estiver apenas atrasado, a retransmissão será duplicada, mas os números de sequência já tratam isso. O receptor deve especificar o número de sequência do pacote sendo reconhecido.

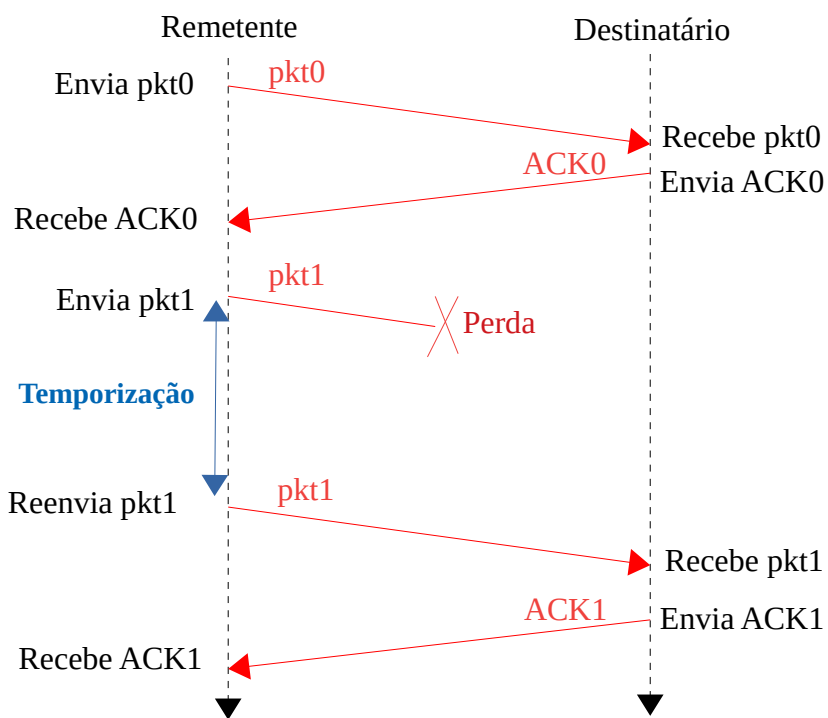
A ideia é utilizar apenas os bits 0 e 1 para identificar os pacotes, pois não vai enviar mais de um.

Exemplo:

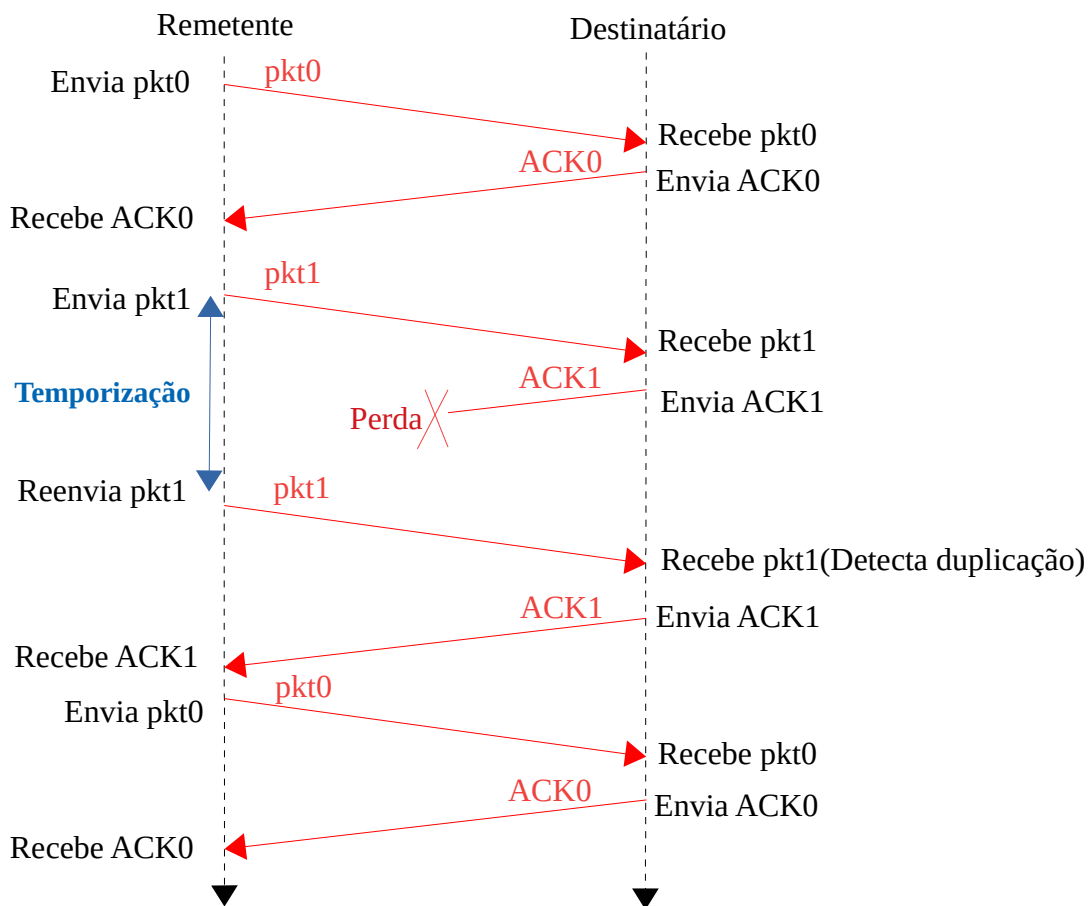
Cenário 1 (Sem perdas):



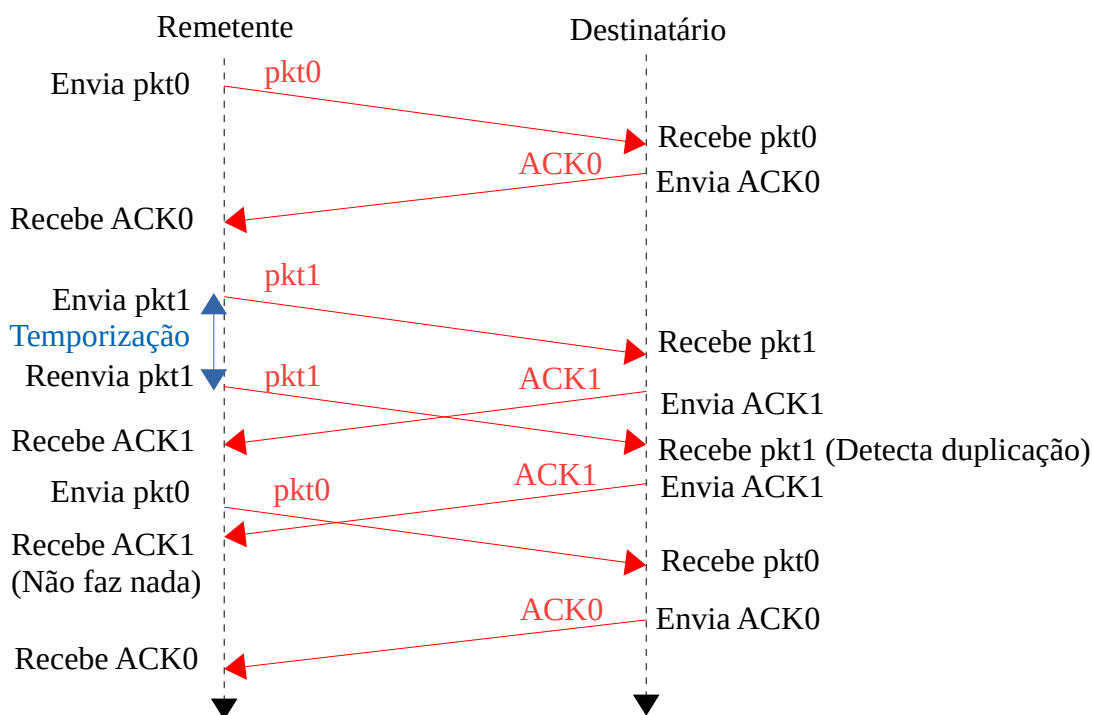
Cenário 2 (Perda de pacote):



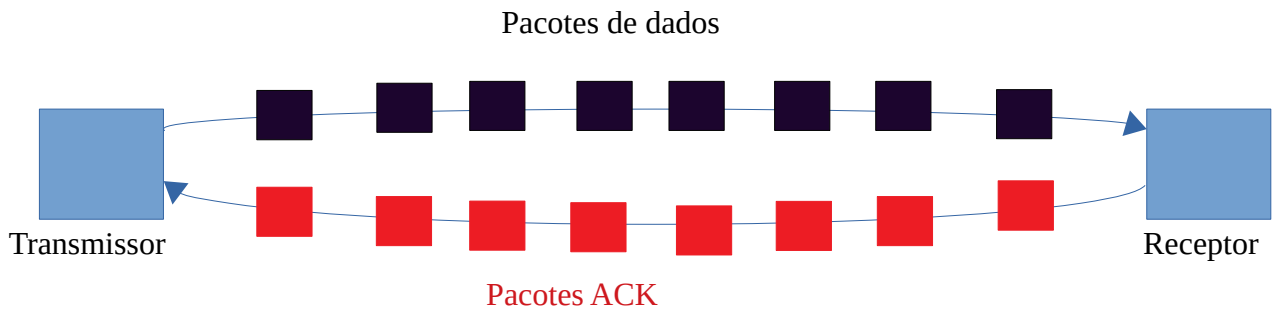
Cenário 3 (Perda de ACK):



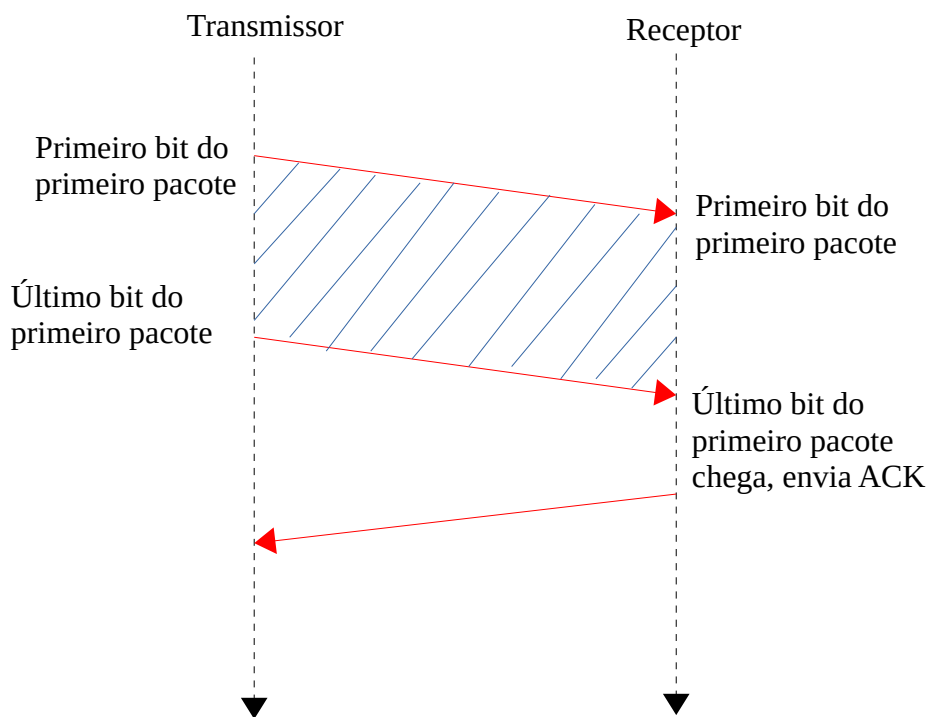
Cenário 4 (Interrupção prematura):



Protocolos com paralelismo → O transmissor envia vários pacotes ao mesmo tempo, todos esperando para serem reconhecidos. Nesse caso, a faixa de números de sequência deve ser aumentada (Não apenas 0 e 1). Deve existir armazenamento no transmissor e/ou no receptor.



Conforme os bits dos pacotes vão chegando, sempre que o último bit de um pacote chegar, é necessário enviar o ACK daquele pacote.



Referências bibliográficas:

TANENBAUM, Andrew. S. Redes de Computadores. São Paulo: Pearson, 5ª Ed. 2011.

KUROSE, James F.; ROSS, Keith W. Redes de Computadores e a Internet – Uma Abordagem Top-Down. São Paulo: Pearson, 6ª Ed. 2013.