



# Redes de Computadores II

## Aula 17

### Terminando o Protocolo TCP

**Prof. Elias P. Duarte Jr.**

Universidade Federal do Paraná (UFPR)

Departamento de Informática

[www.inf.ufpr.br/elias/redes](http://www.inf.ufpr.br/elias/redes)

# Sumário da Aula de Hoje

- Hoje continuamos nosso estudo do protocolo TCP
- Melhorando a taxa de transmissão com o algoritmo de Nagle & e a solução de Clark
- Encerramento da Conexão TCP

# Taxa Payload/Controle

- Como dissemos o header do TCP tem (em geral) 20 bytes
- O header do IP tem também 20 bytes
- Total TCP+IP = 40 bytes
- Para transmitir por exemplo 1 <ENTER>
- 1 byte de payload (o <ENTER>) mais 40 bytes de controle
- TCP tenta melhorar esta taxa

# Algoritmo de Nagle

- Evita que o TCP mande muitos pacotes pequenos
- Ideia: vai juntando pequenas quantidades para fazer uma transmissão maior
- O algoritmo é assim:
  - O primeiro byte manda sem espera
  - Depois só transmite mais bytes se:
    - Já tiver chegado o ACK dos anteriores
    - O buffer já estiver  $\frac{1}{2}$  cheio
    - Já tiver MSS bytes para transmitir

# Do Lado do Receptor: Idem!

- Do lado do receptor pode haver o mesmo problema
- A aplicação só lê um byte por vez do buffer
- Neste caso, se não fizer nada, cada leitura causa um ACK
- Lembrando que o ACK do TCP usa *piggybacking*
- Aguarda ter dados para pegar carona
- Na prática: a maior parte vai exclusivo

# Síndrome da Janela Boba

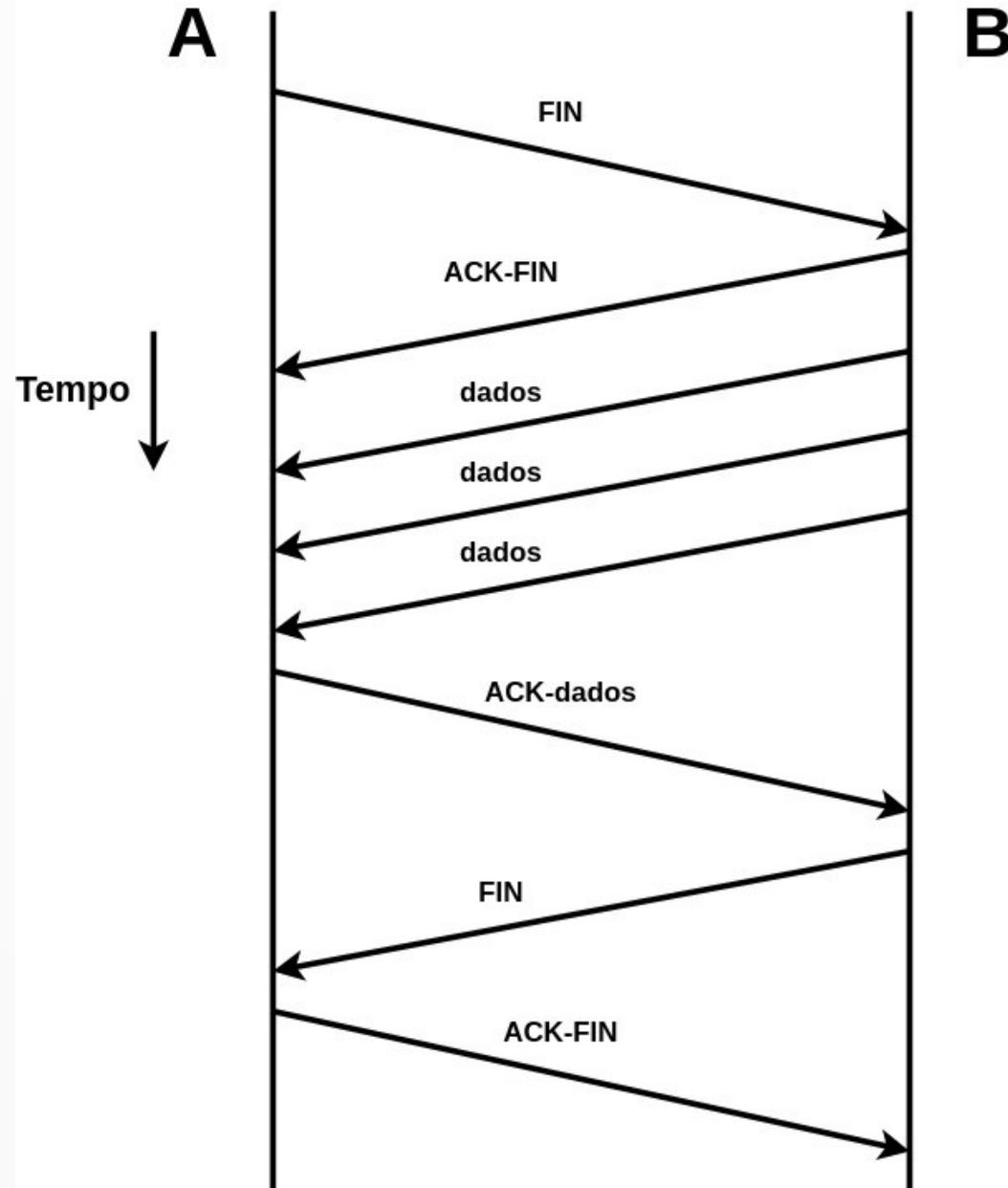
- *Silly Window Syndrome*
- Solução de Clark
  - Não manda um ACK a cada byte retirado
  - Espera ter  $\frac{1}{2}$  do buffer para confirmar
  - Ou MSS bytes para confirmar

# Encerramento da Conexão TCP

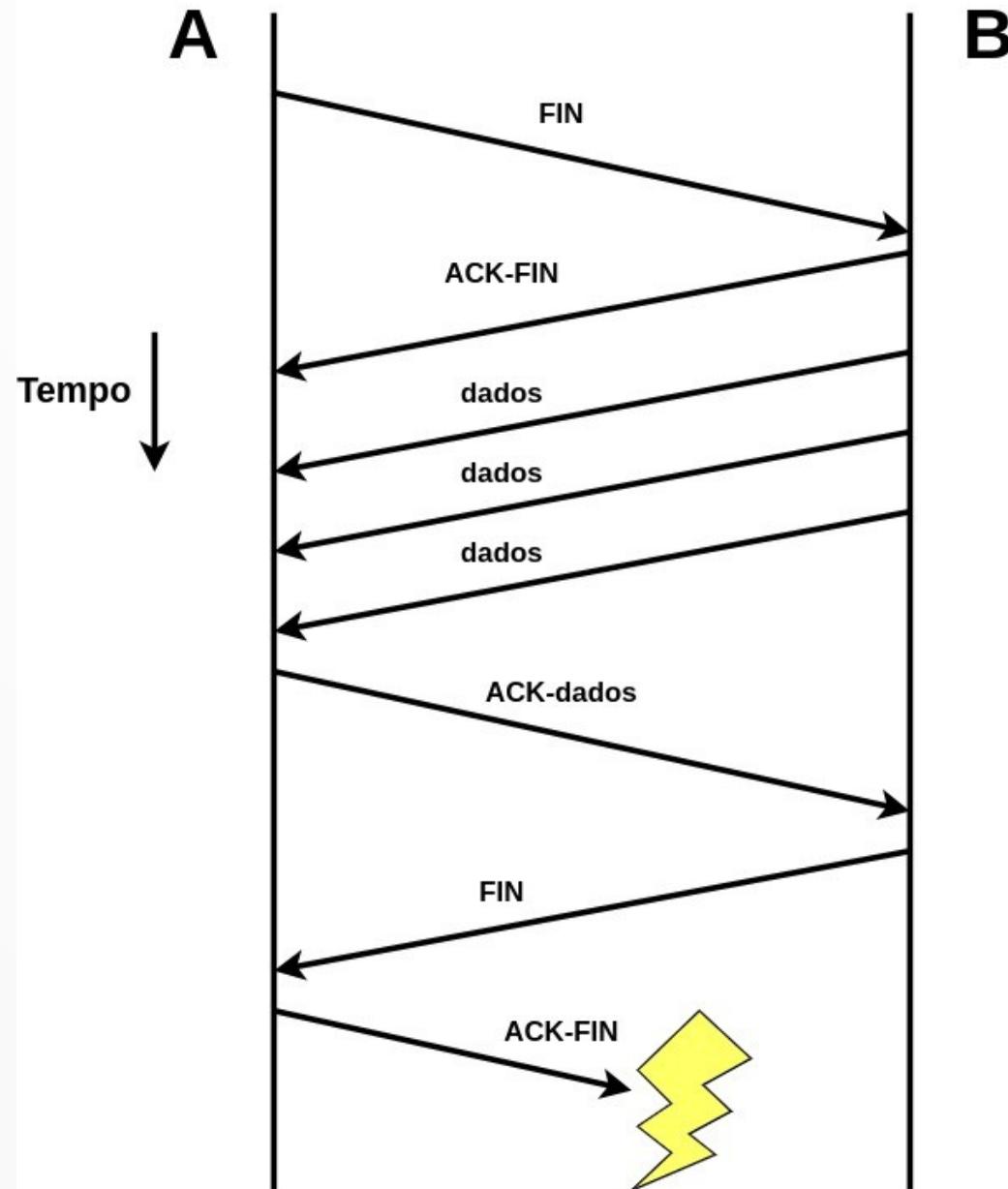
- Apesar da conexão TCP ser *full-duplex* ela é encerrada de cada lado independentemente
- Um dos processos manda um segmento com o flag FIN ligado
- O outro lado deve confirmar, mas pode continuar comunicando!



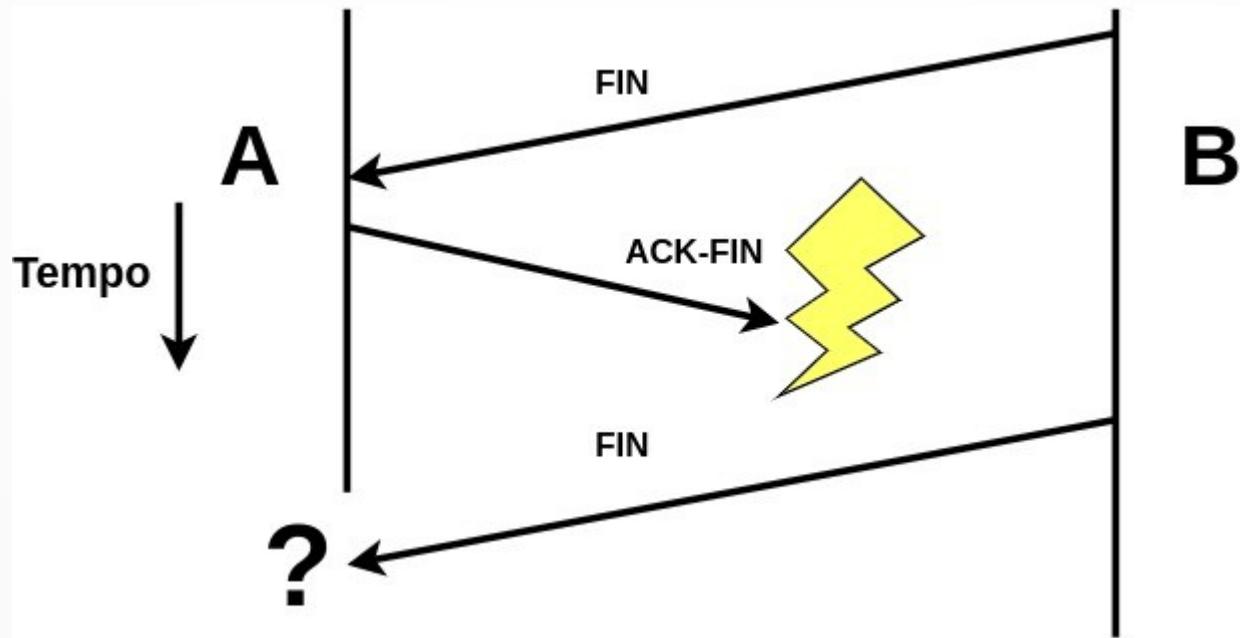
# Encerramento da Conexão TCP



# O que acontece se o último ACK se perde?

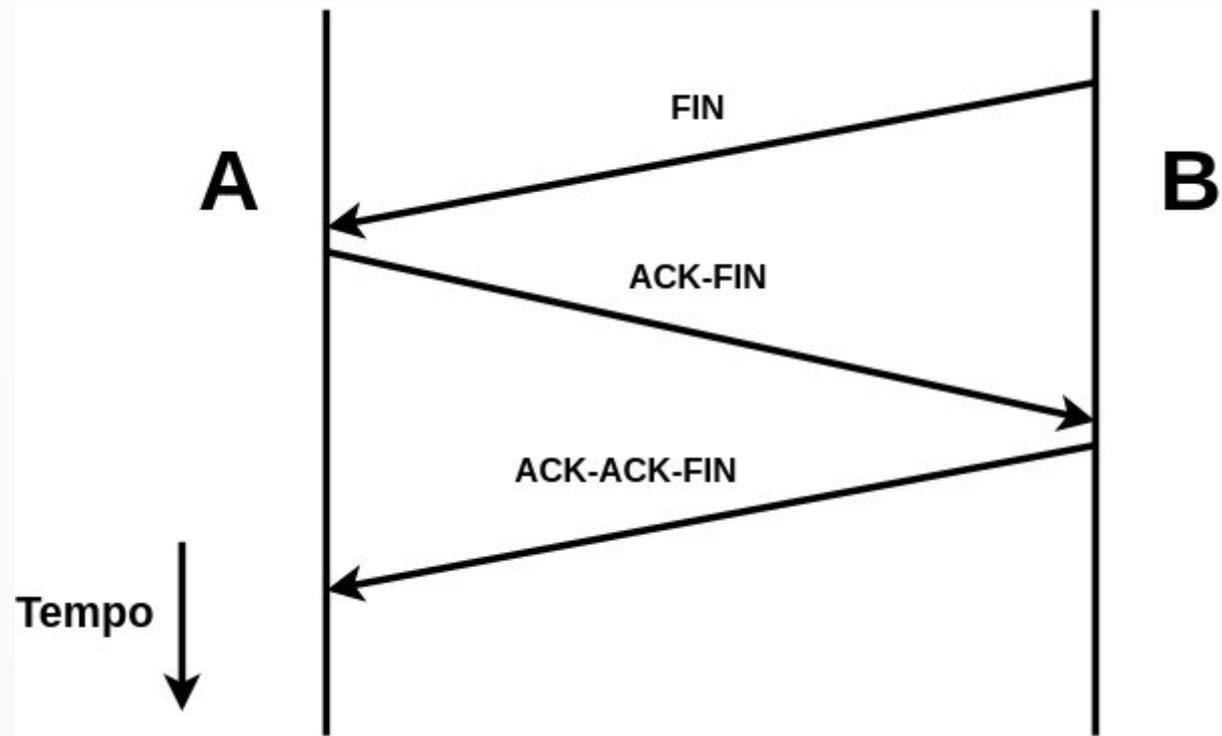


# Último ACK-FIN Se Perde

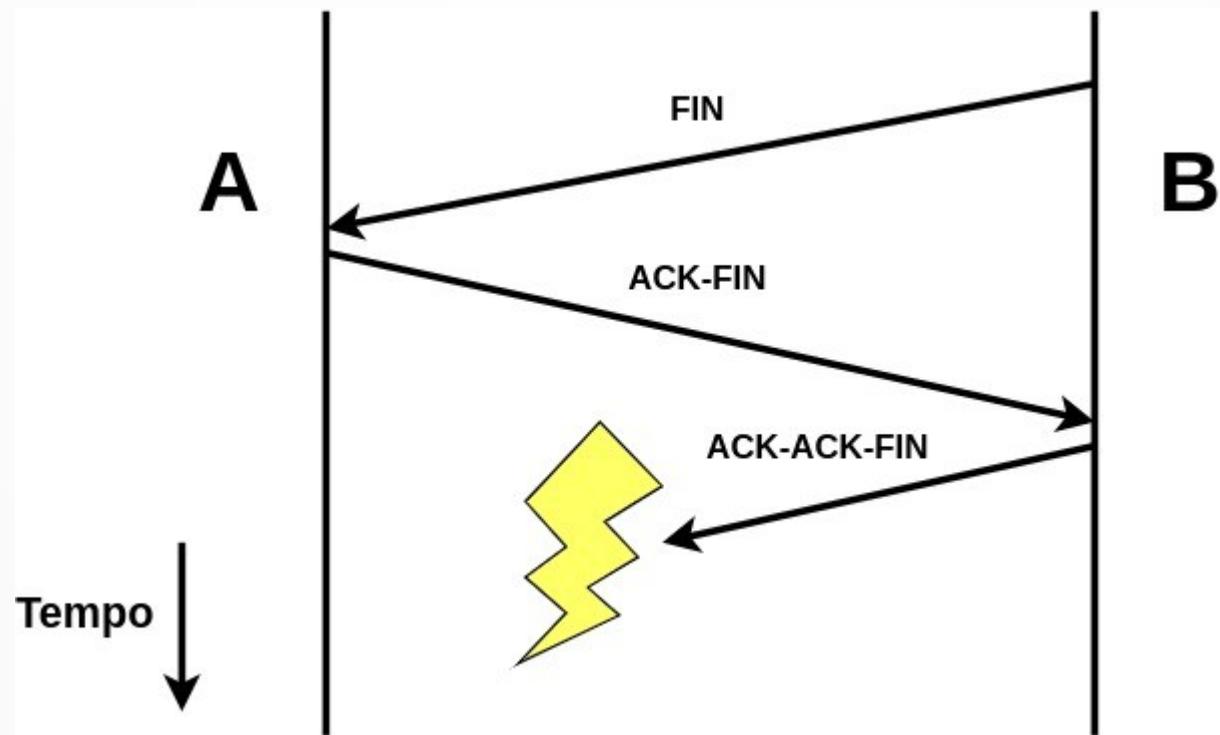


# Tive Uma Ideia: Confirmar o ACK!

- Assim depois de receber o ACK, um ACK-ACK seria gerado



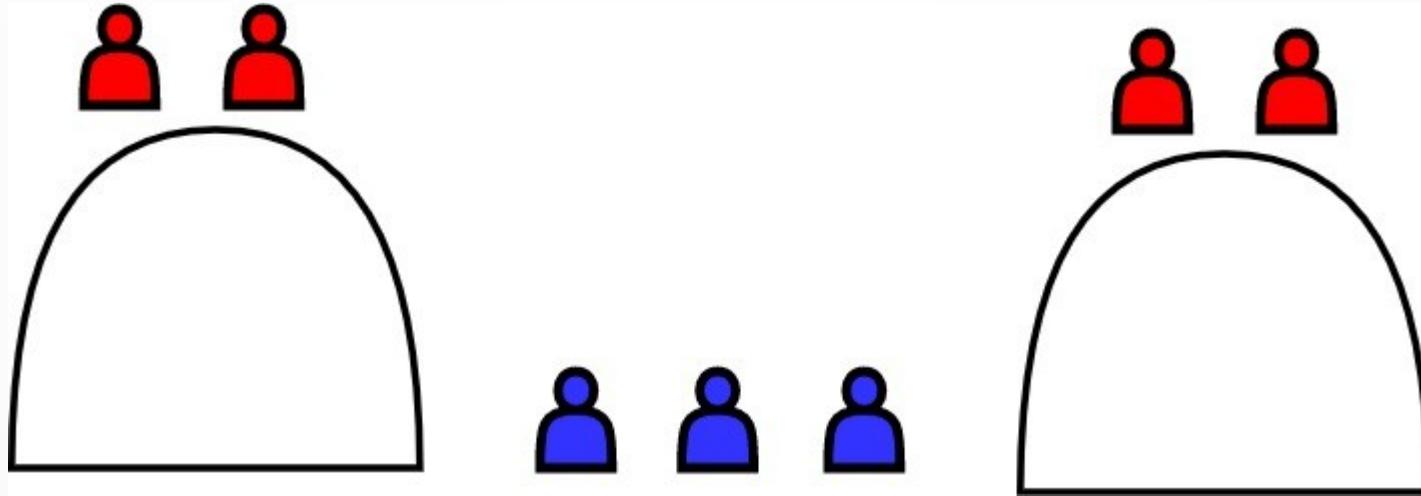
# Mas e se o ACK-ACK-FIN se perde?



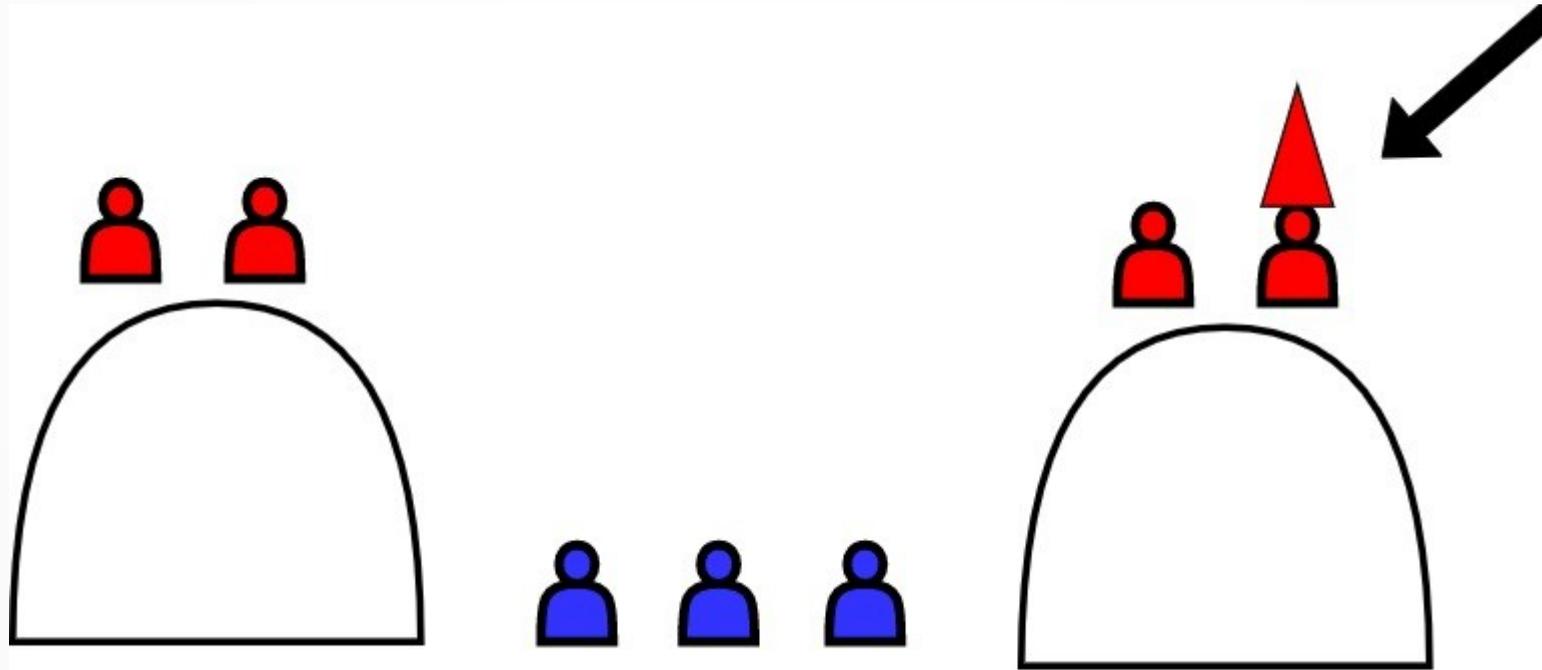
# Tive Uma Ideia: Confirmar o ACK!

- Já sei: vamos gerar um ACK-ACK-ACK
- E se este se perder?
- ACK-ACK-ACK-ACK?
- E se este se perder?
- ACK-ACK-ACK-ACK-ACK?
- E se este se perder?
- ACK-ACK-ACK-ACK-ACK-ACK? 🤖

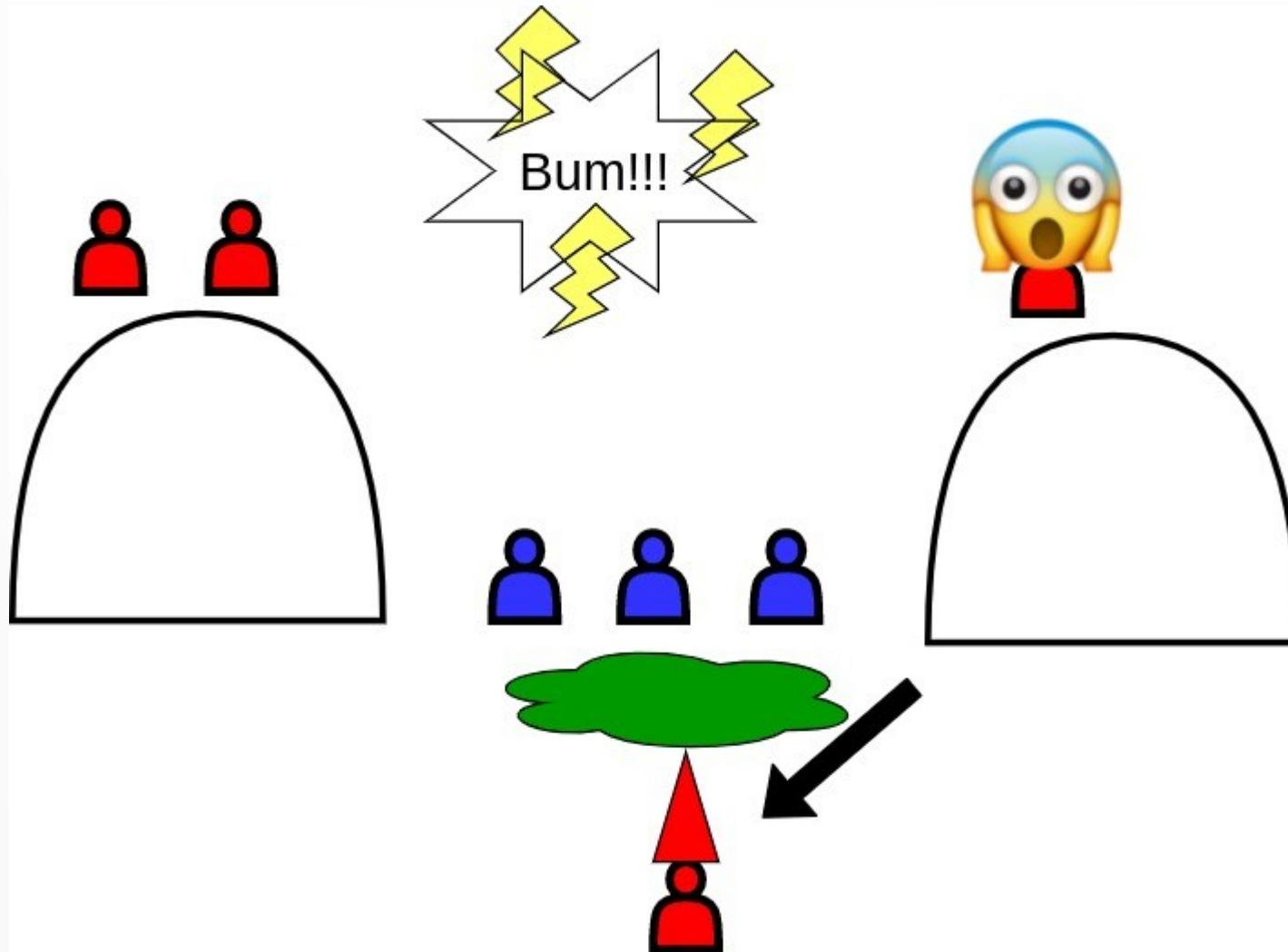
# O Problema dos Dois Exércitos (1)



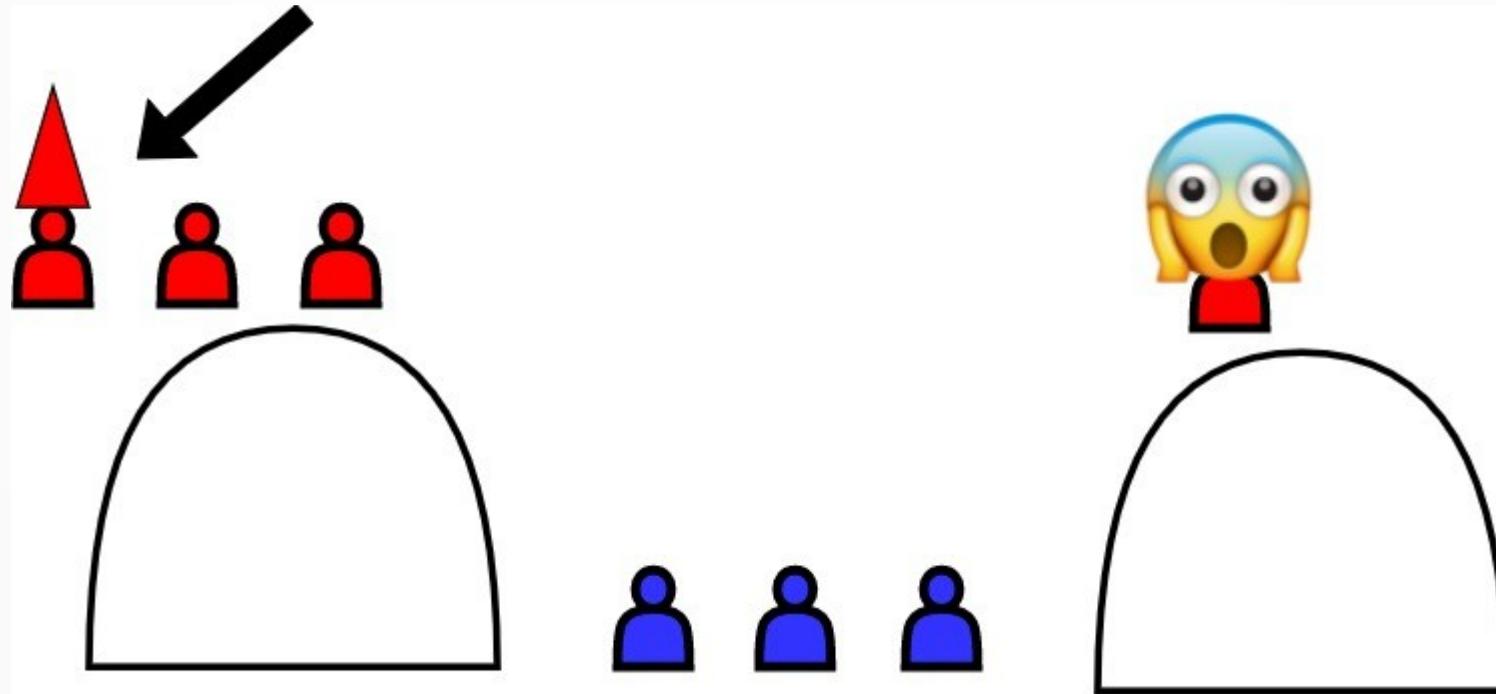
# O Problema dos Dois Exércitos (2)



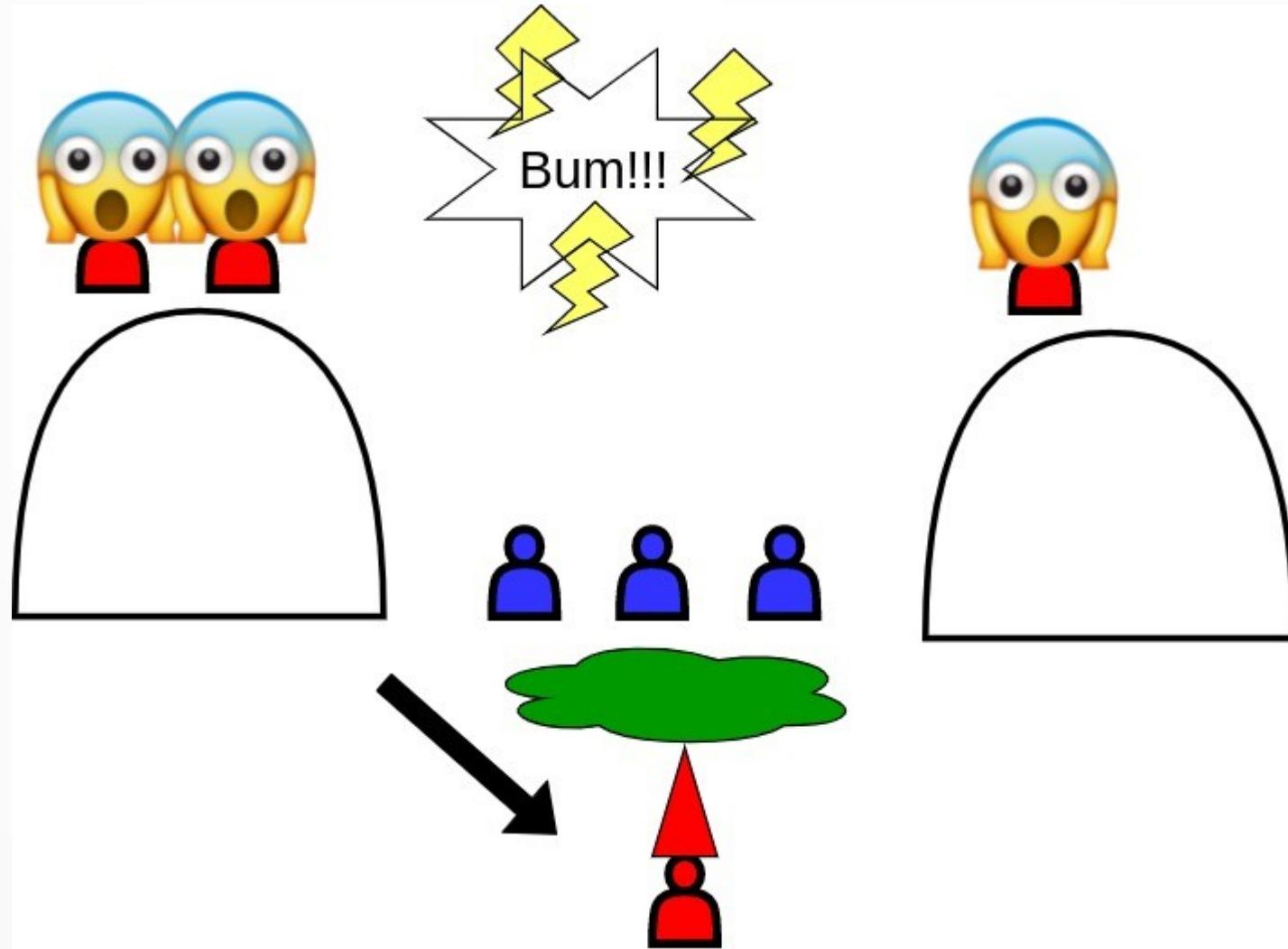
# O Problema dos Dois Exércitos (3)



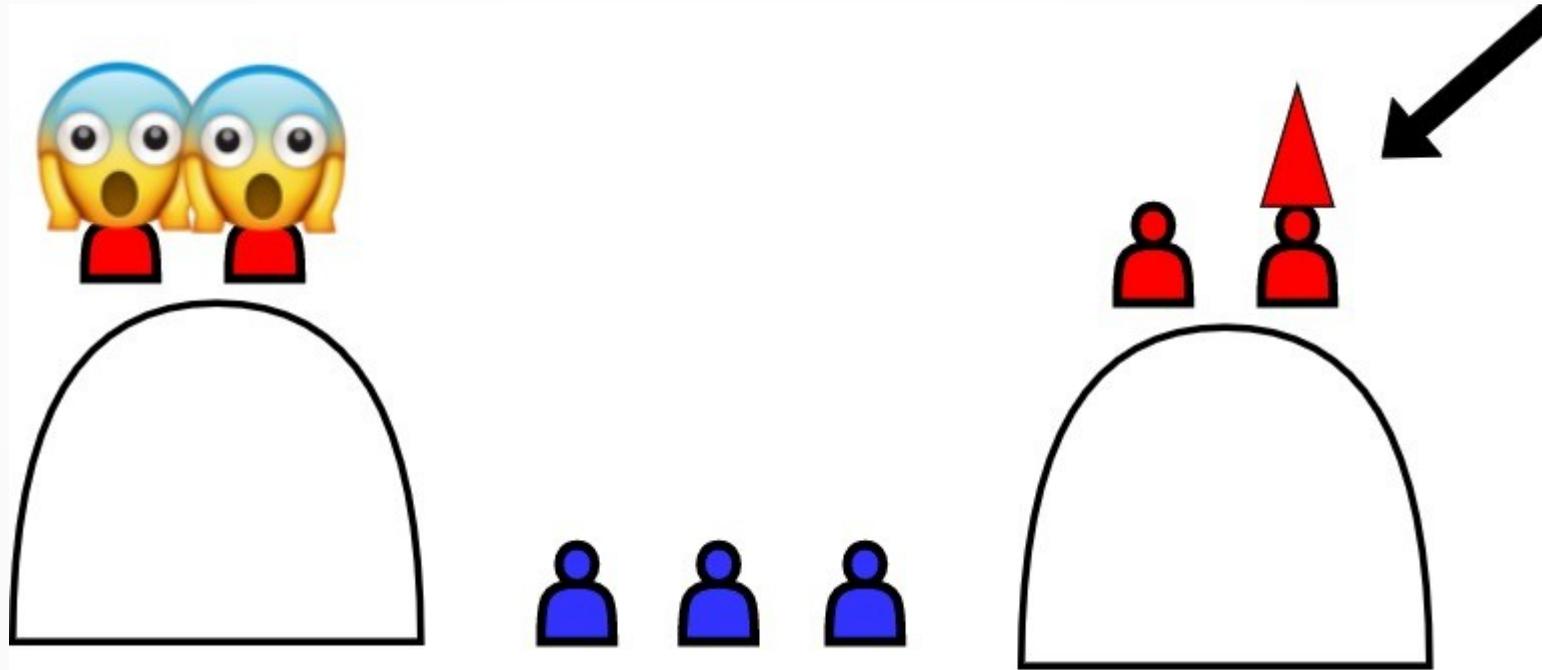
# O Problema dos Dois Exércitos (4)



# O Problema dos Dois Exércitos (5)



# O Problema dos Dois Exércitos (6)



# Um Problema Sem Solução

- Impossibilidade do acordo entre 2 processos que se comunicam sobre um canal de comunicação que perde mensagens
- Ciência da Computação: Tecnologia & Ciência
- Neste caso: ciência → mesmo que a tecnologia avançar tremendamente → não tem solução
- Jamais será possível ter um protocolo de encerramento de conexão TCP que termina corretamente 100% das vezes

# O Que Fazer?

- Em caso de perda do último ACK-FIN
- O processo que transmitiu o último ACK-FIN continua conectado por um período de tempo...
- ... aguardando possíveis retransmissões
- Mas a retransmissão pode acontecer e acontece depois deste período
- Quando chega: não há mais conexão!
- TCP envia RST

# Opcionais TCP: Exemplo

- Um opcional do TCP: confirmação seletiva: SACK
- *Selective Acknowledgments*
- Um ACK do TCP confirma toda a sequência de bytes até o ponto do ACK no fluxo de dados
- Considere que foram transmitidos: 1, 2, 3, 4, 5, 6
- Chegou 1, 3, 4, 5, 6
- Normalmente após a detecção de perda do 2 tudo é retransmitido
- Com o opcional da confirmação seletiva é possível informar lacunas

# Conclusão

- Eficiência com Nagle & Clark
- Encerramento da conexão TCP
- Antes de concluir, reiteramos um ponto importante: o TCP é um protocolo muito eficiente
  - Apesar de tantas funcionalidades
  - Apesar de fazer tantos controles
  - Tem vazão esperada de 90% da vazão nominal da rede

**Obrigado!**

Lembrando: a página da disciplina é:  
<https://www.inf.ufpr.br/elias/redes>