

**Poison Reverse é suficiente para garantir que não aconteça o problema de contagem para o infinito.**  
Poison Reverse ajuda a mitigar, mas não é suficiente para eliminar o problema de contagem ao infinito sozinho. Outros mecanismos adicionais são necessários como por exemplo o Split Horizon e o uso de uma métrica máxima são necessários para complementar o Poison Reverse e melhorar a estabilidade da rede.

**Considere um sistema de framing onde se usa bit stuffing para o escape das flags de início de frame**

Indique as vantagens e desvantagens entre ter uma flag mais curta ou uma flag mais longa.

**Flags Curtas:**

- Vantagens:** Menos overhead na transmissão de dados; mais rápido para identificar o início e o fim dos frames.
- Desvantagens:** Maior probabilidade de ocorrência de padrões de bits semelhantes dentro dos dados, aumentando a necessidade de bit stuffing.

**Flags Longas:**

- Vantagens:** Menor probabilidade de ocorrência acidental dentro dos dados, reduzindo a necessidade de bit stuffing.
- Desvantagens:** Maior overhead; mais bits precisam ser transmitidos para identificar os frames.

Se a flag tiver um número de bits múltiplos de 8, será melhor usar bit stuffing ou byte stuffing? Justifique a sua resposta.

Bit Stuffing vs. Byte Stuffing para Flags de Múltiplos de 8 Bits

Se a flag tiver um número de bits múltiplos de 8:  
Byte Stuffing é melhor.

**Justificação:**

- Byte Stuffing:** Trabalha com bytes inteiros, alinhando-se naturalmente com flags que são múltiplos de 8 bits. Isso simplifica o processamento e reduz a complexidade de inserção e remoção de bits adicionais.
- Bit Stuffing:** Introduce bits individuais, o que pode ser mais complexo e menos eficiente para flags de comprimento múltiplo de 8 bits, pois cada byte precisa ser verificado e possivelmente alterado.

**3. Considere dois hosts de rede A e B ligados por um canal de 10 Mbps. A está a enviar pacotes de 1300 bytes para B.**

**Foi executado um comando 'ping' de A para B, devolvendo o seguinte output:**

\$ ping www.google.com

PING B (w.x.y.z) 56(84) bytes of data.

64 bytes from B (w.x.y.z): icmp\_seq=1 ttl=118 time=34.7 ms  
64 bytes from B (w.x.y.z): icmp\_seq=2 ttl=118 time=34.2 ms  
64 bytes from B (w.x.y.z): icmp\_seq=3 ttl=118 time=34.0 ms

--- B ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2003ms  
rtt min/avg/max/mdev = 33.977/34.282/34.685/0.297 ms

**Desprezando o tempo de transmissão ACK, responda às seguintes questões:**

**(a) Qual é o número máximo de pacotes por segundo que A consegue transmitir para B, se não for usado qualquer protocolo de transporte?**

O tempo de transmissão é dado pela fórmula Tempo de Transmissão = Tamanho do Pacote / Largura de Banda. Aqui, o tamanho do pacote é 1300 bytes (ou 10400 bits) e a largura de banda é 10 Mbps. Portanto, o tempo de transmissão é 10400 bits / 10<sup>6</sup> bits/segundo = 0.00104 segundos (ou 1.04 ms). O número máximo de pacotes por segundo que A pode transmitir para B é o inverso do tempo de transmissão, ou seja, 1 / 0.00104 ≈ 961 pacotes/segundo.

**(b) Se for usado o protocolo Stop&Wait, qual é o número máximo de pacotes por segundo que A consegue enviar para B?**

No protocolo Stop&Wait, cada pacote é seguido por um ACK antes que o próximo pacote seja enviado. Portanto, o tempo total para cada pacote é o tempo de transmissão mais o tempo de ida e volta (RTT). O RTT é dado como aproximadamente 34 ms no output do ping. Portanto, o tempo total para cada pacote é 1.04 ms (tempo de transmissão) + 2\*34 ms (RTT) = 69.04 ms. O número máximo de pacotes por segundo que A pode enviar para B sob o protocolo Stop&Wait é o inverso deste tempo total, ou seja, 1 / 0.06904 ≈ 14 pacotes/segundo.

**(c) Qual é a taxa de utilização do canal, nas condições da alínea anterior?**

A taxa de utilização do canal sob as condições do protocolo Stop&Wait é dada pela fórmula Taxa de Utilização = Tempo de Transmissão / (Tempo de Transmissão + RTT). Usando os valores obtidos anteriormente para o tempo de transmissão e RTT, a taxa de utilização do canal é 1.04 ms / 69.04 ms ≈ 0.015, ou 1.5%.

**(e) Proponha um timeout adequado para este canal.**

Um timeout adequado para este canal poderia ser ligeiramente acima do tempo máximo de ida e volta observado, a fim de levar em conta variações no atraso, garantindo uma comunicação eficiente. O RTT máximo das estatísticas do ping é de aproximadamente 34.7 ms; portanto, um valor um pouco mais alto pode ser considerado como um timeout. Por exemplo, poderíamos usar um valor de 40 ms para ter uma margem de segurança.

**4. Considere dois hosts de rede A e B ligados por um canal de 300Kbps e com um tempo de propagação entre extremidades de 75 milissegundos. A está a enviar pacotes de 10000 bits de comprimento para B.**

**(a) Qual é o número máximo de pacotes por segundo que A consegue transmitir para B (continuamente)?**

Transmission Time = Packet Size / Bandwidth  
  
10000 bits / 300000 bits/s = 0.033s  
  
1 / Transmission Time = 30 pacotes/s

**(b) Qual é o número máximo de pacotes por segundo que A consegue transmitir para B, usando o protocolo Stop&Wait?**

75ms = 0.075s  
  
Transmission Time + 2 \* Propagation Time = 0.033s + 2 \* 0.075s = 0.183s  
  
1 / (Transmission Time + 2 \* Propagation Time) = 1/0.183s ≈ 5 pacotes/s

**(c) Qual é a taxa de utilização do canal nas condições da alínea anterior?**

RTT = 2 \* Propagation Time  
  
Usage Rate = Transmission Time / (Transmission Time + RTT) =  
0.033s / (0.033s + 0.15s) ≈ 0.18 = 18%

**5. Considere dois hosts de rede A e B ligados por um canal de 1Gbps. A está a enviar para B pacotes de 1500 bits de comprimento.**

**Foi executado um comando 'ping' de A para B, devolvendo o seguinte output:**

\$ ping B  
PING B (1.2.3.4) 56(84) bytes of data.  
64 bytes from 1.2.3.4 (1.2.3.4): icmp\_seq=1 ttl=30 time=250 ms  
64 bytes from 1.2.3.4 (1.2.3.4): icmp\_seq=2 ttl=30 time=259 ms  
64 bytes from 1.2.3.4 (1.2.3.4): icmp\_seq=3 ttl=30 time=265 ms

--- B ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2003ms  
rtt min/avg/max/mdev = 250.822/261.134/265.468/3.066 ms

**Desprezando o tempo de transmissão ACK e considerando a velocidade teórica de propagação da fibra ótica (~ 200.000Km/s), responda às seguintes questões:**

**(a) Qual é o número máximo de pacotes por segundo que A consegue transmitir para B, usando o protocolo Stop&Wait?**

O número máximo de pacotes por segundo que A consegue transmitir para B, usando o protocolo Stop&Wait, é determinado pelo tempo de ida e volta (RTT). No protocolo Stop&Wait, cada pacote deve ser reconhecido antes do próximo pacote ser enviado. Portanto, o número máximo de pacotes por segundo é 1/RTT. Considerando o RTT médio de 261.134 ms (ou 0.261134 segundos), o número máximo de pacotes por segundo é aproximadamente 1/0.261134 ≈ 3.828 pacotes por segundo.

**(b) Proponha um tamanho de janela para usar o protocolo Sliding Window. Haverá vantagem em usar um protocolo deste tipo? Justifique.**

O tamanho da janela para o protocolo Sliding Window deve ser escolhido de forma a maximizar a utilização do canal. Uma boa escolha seria um tamanho de janela igual ao produto da largura de banda e o RTT (BW \* RTT), também conhecido como produto Bandwidth-Delay. Considerando uma largura de banda de 1 Gbps e um RTT médio de 261.134 ms, o tamanho da janela seria de aproximadamente 261.134 Mbits. Usar um protocolo deste tipo tem a vantagem de permitir múltiplos pacotes em trânsito ao mesmo tempo, aumentando a utilização do canal e melhorando o throughput.

**(d) A determinado momento da ligação, A recebe um pacote TCP com Advertised Window = 0. Qual o motivo de isto acontecer, e o que deve A fazer nesta situação?**

Se A recebe um pacote TCP com Advertised Window = 0, isso significa que B está temporariamente incapaz de receber mais dados. A deve parar de enviar dados e começar a enviar sondagens de janela (window probes) para verificar quando B está pronto para receber dados novamente.

**(e) Assuma que está numa equipa, a implementar um protocolo de aplicação para funcionar num serviço entre A e B. O seu colega diz que não deve enviar mensagens (ao nível da aplicação) com mais de 1460 bytes. Concorda com ele? Justifique.**

Concordo com o seu colega. O tamanho máximo de um pacote TCP/IP é geralmente limitado a 1500 bytes devido à limitação do tamanho máximo de transmissão (MTU) na maioria das redes Ethernet. No entanto, o cabeçalho TCP/IP ocupa 40 bytes, deixando 1460 bytes para os dados da aplicação. Portanto, para evitar a fragmentação de pacotes, é aconselhável limitar o tamanho das mensagens da aplicação a 1460 bytes.

Packet Transit Time = Transmission Time + Propagation Time

Transmission Time = Packet Size / Bandwidth

Propagation Time = Channel Length / Propagation Speed( ~ 200.000Km/s)

Usage Rate = Transmission Time / ( Transmission Time + RTT ) Goodput = Packet Size / RTT (Packet size can be the full size of the window in bits)

**Considere dois hosts de rede A e B ligados por um canal de 1Gbps e com um tempo de propagação entre extremidades de 80 milissegundos. A está a enviar pacotes com 1500 bytes de comprimento para B.**

**Qual é o número máximo de pacotes por segundo que A consegue transmitir para B, usando o protocolo Selective Repeat com uma janela de tamanho 2?**

Cálculos:

- Tamanho do pacote: 1500 bytes = 1500 \* 8 bits = 12,000 bits
- Largura de banda do canal: 1 Gbps = 1,000,000,000 bits por segundo
- Tempo de propagação: 80 ms = 0.08 segundos
- Janela de tamanho 2: Protocolo Selective Repeat permite enviar 2 pacotes antes de esperar por um ACK.

Tempo total para enviar e receber um pacote:

Tempo de transmissão (Tt) =  $\frac{\text{Tamanho do pacote}}{\text{Largura de banda}} = \frac{12,000 \text{ bits}}{1,000,000,000 \text{ bits/segundo}} = 0.000012 \text{ segundos}$

Tempo de ida e volta (RTT):

RTT = 2 × Tempo de propagação = 2 × 0.08 segundos = 0.16 segundos

Número máximo de pacotes na janela:

Janela de tamanho = 2

Dado que o tempo de transmissão é muito pequeno em comparação com o tempo de ida e volta (RTT), o fator limitante é a RTT.

Pacotes transmitidos por segundo:

Taxa de pacotes =  $\frac{\text{Janela}}{\text{RTT}} = \frac{2}{0.16} = 12.5 \text{ pacotes por segundo}$

**Qual é a taxa de utilização do canal nas condições da alínea anterior?**

Tempo total para transmitir 2 pacotes:

Tempo total = 2 × (Tempo de transmissão + RTT)

Mas, como a transmissão é contínua no Selective Repeat (não espera-se a RTT completa para cada pacote individualmente, mas sim por janelas), consideramos apenas o RTT para a janela de 2 pacotes.

Utilização do canal:

Utilização =  $\frac{\text{Tempo de transmissão total dos 2 pacotes}}{\text{RTT}}$

Utilização =  $\frac{2 \times 0.000012 \text{ segundos}}{0.16 \text{ segundos}} = 0.00015 = 0.015\%$

**Um dos hosts recebeu quatro ACKS relativos ao mesmo pacote. Indique uma razão que possa ter levado isto a acontecer.**

Uma razão comum para este evento é a retransmissão devido à perda de pacotes ou atrasos significativos na rede. Pode acontecer que o receptor não tenha recebido o ACK para um determinado pacote (por causa de perda ou atraso), e o remetente retransmite o pacote várias vezes. Quando o receptor finalmente recebe todas as retransmissões, ele envia múltiplos ACKs em resposta.

**Enviar pacotes menores:**

Pacotes menores podem ser transmitidos mais rapidamente, aumentando a taxa de transferência efetiva.

Aumentar a janela de envio: Usar janelas de envio maiores do que os outros hosts, enviando mais pacotes antes de esperar por ACKs.

Manipular temporizadores:

Reduzir os temporizadores de retransmissão, enviando pacotes repetidamente mais rapidamente do que os outros hosts.

Considere os três algoritmos de Routing vistos nas aulas: Inundação, Link state e Bellman-Pord.

1. Inundação (Flooding)

Ponto Forte:

A inundação garante que os pacotes alcancem todos os nós na rede, independentemente da topologia ou falhas nos links, já que os pacotes são enviados por todos os caminhos possíveis.

Ponto Fraco:

A inundação gera uma quantidade massiva de tráfego redundante, o que pode sobrecarregar a rede, especialmente em redes grandes, levando a congestionamento e desperdício de largura de banda.

2. Link State

Ponto Forte:

O protocolo de estado de enlace (Link State) permite que cada roteador tenha uma visão completa da rede, o que resulta em escolhas de rotas mais eficientes e rápidas. Os algoritmos de roteamento de estado de enlace, como o OSPF (Open Shortest Path First), são muito eficazes em redes estáticas ou com mudanças lentas.

Ponto Fraco:

A implementação e manutenção de um algoritmo de estado de enlace podem ser complexas e exigem maior capacidade de processamento e memória nos roteadores, especialmente em redes grandes.

3. Bellman-Ford (Distance Vector)

Ponto Forte:

O algoritmo de vetor de distância (Distance Vector) é mais simples de implementar e requer menos capacidade de processamento em comparação com o estado de enlace. Cada roteador apenas precisa saber a distância para seus vizinhos diretos.

Ponto Fraco:

O algoritmo Bellman-Ford pode demorar para convergir, especialmente em redes grandes, e é suscetível a problemas de roteamento em loop e à contagem até o infinito.

Para o edifício CLAV da Universidade de Évora, o Link State poderia ser o melhor dos três algoritmos.

1. O edifício parece ter uma infraestrutura complexa com salas de aula, gabinetes de trabalho e corredores, o que sugere uma topologia de rede possivelmente complexa.

2. O Link State permite que cada ponto de acesso WiFi tenha uma visão completa da topologia da rede, o que é útil em ambientes complexos para encontrar as melhores rotas.

3. A capacidade do Link State para rápido recalculo das rotas é vantajosa em um ambiente dinâmico como uma universidade, onde dispositivos podem entrar e sair da rede frequentemente.

A rede 192.168.1.0/29 não poderá ter mais do que 12 hosts diferentes.

Falso. A rede 192.168.1.0/29 suporta até 6 hosts, pois a máscara /29 indica 29 bits fixos, deixando apenas 3 bits para endereços de host.

Considerando que é possível encapsular qualquer protocolo em outro, dê um exemplo em que seria útil encapsular IP sobre TCP.

Encapsular IP sobre TCP não é um procedimento padrão, pois o TCP é normalmente encapsulado dentro do IP para transporte de dados. No entanto, um exemplo teórico onde isso poderia ser útil seria em uma situação de tunelamento, onde o IP seria usado para transportar dados através de uma conexão TCP estabelecida, como em uma VPN (Virtual Private Network) que utiliza TCP para criar um túnel seguro através da Internet.

Comente a afirmação: “Numa rede WiFi é bastante útil utilizar encaminhamento por inundação, desde que seja com aprendizagem pelo caminho inverso.”

O encaminhamento por inundação numa rede WiFi pode ser útil em cenários específicos, como em redes ad hoc ou em situações de descoberta inicial de rotas. A aprendizagem pelo caminho inverso ajuda a reduzir o tráfego desnecessário, memorizando as rotas eficientes para futuras comunicações. No entanto, deve ser usado com cautela para evitar sobrecarga na rede devido ao excesso de pacotes sendo transmitidos.

Considere dois hosts de rede A e B ligados por um canal de 25Mbps e com um tempo de propagação entre extremidades de 35 milissegundos. A está a enviar pacotes de 1400 bytes de comprimento para B.

(a)Qual é o número máximo de pacotes por segundo que A consegue transmitir para B, usando o protocolo Stop&Wait?

(a) Número máximo de pacotes por segundo que A consegue transmitir para B usando Stop&Wait

Transmission Time (T<sub>sub-trans</sub>):

$$T_{trans} = \frac{\text{Packet Size}}{\text{Bandwidth}} = \frac{1400 \times 8 \text{ bits}}{25 \times 10^6 \text{ bps}} = \frac{11200}{25000000} \approx 0.000448 \text{ segundos}$$

Propagation Time (T<sub>sub-prop</sub>):

$$T_{prop} = 35 \text{ ms} = 0.035 \text{ segundos}$$

Round-Trip Time (RTT):

$$RTT = 2 \times T_{prop} = 2 \times 0.035 = 0.07 \text{ segundos}$$

Total Time for Stop&Wait (T<sub>sub-total</sub>):

$$T_{total} = T_{trans} + RTT = 0.000448 + 0.07 \approx 0.070448 \text{ segundos}$$

Pacotes por segundo:

$$\text{Pacotes por segundo} = \frac{1}{T_{total}} \approx \frac{1}{0.070448} \approx 14.2 \text{ pacotes por segundo}$$

(b) Qual é a taxa de utilização do canal nas condições da alínea anterior?

(b) Taxa de utilização do canal nas condições da alínea anterior

Uso do canal:

$$\text{Usage Rate} = \frac{T_{trans}}{T_{trans} + RTT} = \frac{0.000448}{0.000448 + 0.07} \approx \frac{0.000448}{0.070448} \approx 0.00636$$

Ou seja, a taxa de utilização do canal é cerca de 0.64%.

(c) Mantendo as condições anteriores, que tamanho de janela usaria para um protocolo Go-Back-N ?

(c) Tamanho da janela para um protocolo Go-Back-N

Para o protocolo Go-Back-N, o tamanho da janela N deve ser tal que mantenha o canal sempre ocupado. O valor de N pode ser determinado pela fórmula:

$$N \geq \frac{RTT}{T_{trans}}$$

Calculando:

$$N \geq \frac{0.07}{0.000448} \approx 156.25$$

Portanto, o tamanho da janela deve ser pelo menos 157 pacotes.

(d) E para Selective Repeat? Justifique

(d) Tamanho da janela para Selective Repeat

Para o protocolo Selective Repeat, o tamanho da janela W é geralmente metade do valor utilizado para o Go-Back-N, devido ao controle mais preciso de erros:

$$W = \frac{N + 1}{2} = \frac{157 + 1}{2} = 79$$

Portanto, o tamanho da janela para Selective Repeat deve ser pelo menos 79 pacotes.

(e) Proponha um tempo de timeout adequado para este canal. Explique porque seria adequado.

(e) Proposta de um tempo de timeout adequado para este canal

O tempo de timeout (T<sub>timeout</sub>) deve ser maior que o Round-Trip Time (RTT) para assegurar que pacotes perdidos ou ACKs atrasados não causem retransmissões desnecessárias.

Um valor seguro seria:

$$T_{timeout} \approx 1.5 \times RTT$$

Calculando:

$$T_{timeout} \approx 1.5 \times 0.07 = 0.105 \text{ segundos}$$

Portanto, um tempo de timeout de aproximadamente 0.105 segundos seria adequado para este canal. Esse valor garante que o transmissor aguarde tempo suficiente para receber um ACK antes de retransmitir o pacote, reduzindo a probabilidade de retransmissões desnecessárias devido a pequenos atrasos no canal.

Numa rede Wi-Fi todos os hosts têm acesso a todos os pacotes transmitidos.

Falsa. Justificação: Em redes Wi-Fi, embora o meio físico de transmissão seja compartilhado, os pacotes não são necessariamente acessíveis a todos os hosts. Protocolos de segurança, como WPA2, criptografam os dados transmitidos, tornando-os acessíveis apenas ao destinatário correto. Além disso, técnicas como o uso de pontos de acesso e roteadores controlam o fluxo de pacotes para destinos específicos.

A rede 192.168.1.0/26 poderá ter mais do que 8 hosts diferentes.

Verdadeira. Justificação: Um prefixo /26 na notação CIDR (Classless Inter-Domain Routing) indica que os primeiros 26 bits são usados para identificar a rede, restando 6 bits para identificar os hosts. Isso resulta em:

$$2^6 = 64 \text{ endereços IP}$$

Desses 64 endereços, dois são reservados: um para o endereço de rede (192.168.1.0) e outro para o endereço de broadcast (192.168.1.63). Portanto, a rede 192.168.1.0/26 pode suportar até 62 hosts utilizáveis (64 - 2 = 62), o que é muito mais do que 8 hosts.

Considere dois hosts de rede A e B ligados por um canal de 1Gbps e com um tempo de propagação entre extremidades de 10 milissegundos. A está a enviar pacotes de 1400 bytes de comprimento para B.

(a) Qual é o número máximo de pacotes por segundo que A consegue transmitir para B, usando o protocolo Stop&Wait?

(a) Número máximo de pacotes por segundo que A consegue transmitir para B usando o protocolo Stop&Wait

Transmission Time (T<sub>sub-trans</sub>):

$$T_{trans} = \frac{\text{Packet Size}}{\text{Bandwidth}} = \frac{1400 \times 8 \text{ bits}}{1 \times 10^9 \text{ bps}} = \frac{11200}{1 \times 10^9} = 0.0000112 \text{ segundos}$$

Propagation Time (T<sub>sub-prop</sub>):

$$T_{prop} = 10 \text{ ms} = 0.01 \text{ segundos}$$

Round-Trip Time (RTT):

$$RTT = 2 \times T_{prop} = 2 \times 0.01 = 0.02 \text{ segundos}$$

Total Time for Stop&Wait (T<sub>sub-total</sub>):

$$T_{total} = T_{trans} + RTT = 0.0000112 + 0.02 = 0.0200112 \text{ segundos}$$

Pacotes por segundo:

$$\text{Pacotes por segundo} = \frac{1}{T_{total}} = \frac{1}{0.0200112} \approx 49.97 \text{ pacotes por segundo}$$

(b) Qual é a taxa de utilização do canal nas condições da alínea anterior?

(b) Taxa de utilização do canal nas condições da alínea anterior

Uso do canal:

$$\text{Usage Rate} = \frac{T_{trans}}{T_{trans} + RTT} = \frac{0.0000112}{0.0000112 + 0.02} = \frac{0.0000112}{0.0200112} \approx 0.000559$$

Ou seja, a taxa de utilização do canal é cerca de 0.056%.

(c) Mantendo as condições anteriores, que tamanho de janela usaria para um protocolo Go-Back-N ?

(c) Tamanho da janela para um protocolo Go-Back-N

Para o protocolo Go-Back-N, o tamanho da janela N deve ser tal que mantenha o canal sempre ocupado. O valor de N pode ser determinado pela fórmula:

$$N \geq \frac{RTT}{T_{trans}}$$

Calculando:

$$N \geq \frac{0.02}{0.0000112} \approx 1785.71$$

Portanto, o tamanho da janela deve ser pelo menos 1786 pacotes.

(d) Qual será o comprimento deste canal (em Km)?

(d) Comprimento deste canal (em Km)

Sabemos que:

$$T_{prop} = \frac{\text{Channel Length}}{\text{Propagation Speed}}$$

Rearranjando para encontrar o comprimento do canal:

$$\text{Channel Length} = T_{prop} \times \text{Propagation Speed}$$

Usando a velocidade de propagação de aproximadamente 200,000 km/s:

$$\text{Channel Length} = 0.01 \text{ segundos} \times 200,000 \text{ km/s} = 2000 \text{ km}$$

Portanto, o comprimento do canal é de 2000 km.

(e) Proponha um tempo de timeout adequado para este canal. Explique porque seria adequado.

(e) Proposta de um tempo de timeout adequado para este canal

O tempo de timeout (T<sub>timeout</sub>) deve ser maior que o Round-Trip Time (RTT) para assegurar que pacotes perdidos ou ACKs atrasados não causem retransmissões desnecessárias.

Um valor seguro seria:

$$T_{timeout} \approx 1.5 \times RTT$$

Calculando:

$$T_{timeout} \approx 1.5 \times 0.02 = 0.03 \text{ segundos}$$

Portanto, um tempo de timeout de aproximadamente 0.03 segundos seria adequado para este canal. Esse valor garante que o transmissor aguarde tempo suficiente para receber um ACK antes de retransmitir o pacote, reduzindo a probabilidade de retransmissões desnecessárias devido a pequenos atrasos no canal.