



Resolução:

1) O diagrama temporal TCP04.a esquematiza uma evolução possível: A submete ao nível transporte quatro mensagens,  $M_0$  a  $M_3$ . Para efeitos de comunicação com B, A embebe-os em *segmentos*. Ademais de outros *campos*, cada segmento detém um campo *Numeração*, um campo *Dados* e um campo *Check-sum*.

Numeração	Outros campos	Dados	Check-sum
-----------	---------------	-------	-----------

No tocante a  $M_0$ , A encapsula-o num segmento,  $S_0$ ; em termos práticos, o campo *Dados* será preenchido com os bits de  $M_0$ , o campo *Numeração* é posto a 0, e no campo *Check-sum* são vertidos os bits apropriados ao algoritmo em vigor para detecção de erro... Após o que despacha  $S_0$  para B... Ao chegar a B, o segmento é testado e, não tendo ocorrido erros,  $M_0$  é extraído do campo *Dados*, e passada ao nível Utilizador - e é devolvido a A o correspondente *Ack*. Quando este chegar a A, o nível transporte pode dar início à transmissão da mensagem seguinte,  $M_1$ ; o procedimento é análogo...

Realce-se que a numeração dos segmentos é feita em módulo-2, isto é: conquanto estejam em jogo *quatro* mensagens, são utilizados apenas *dois* valores para distinguir os segmentos, no caso os valores 0 e 1.

No que toca à troca das duas primeiras mensagens, isso não dá azo a confusões: o protocolo utilizado é *Stop-and-wait*, isto é, *dá-se início ao envio de  $M_1$  apenas depois de ter sido recebido o Ack correspondente a  $M_0$* ... Mas essa é uma situação ideal: não há erros...

Que acontecerá no mundo real, em que há erros? Existem duas situações possíveis:

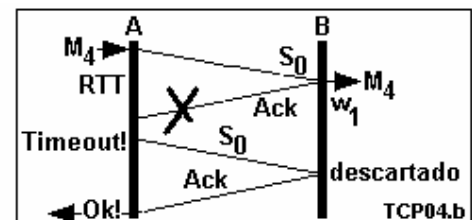
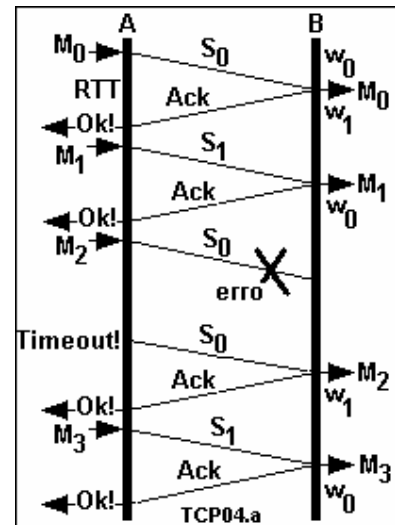
1. uma delas é a ocorrência de erros/perda no *segmento transmitido de A para B*, seja aquele,  $S_0$ , que transporta  $M_2$ ; por aplicação do algoritmo acima assinalado, há alguma probabilidade de B chegar a detectar que o segmento está errado - pelo que: B não devolverá *Ack!* consequentemente, A não receberá *Ack* - e isso levanta uma questão: como evitar que A fique *indefinidamente* à espera?! A resposta é: dotando A de um *relógio*: imediatamente após o envio do último bit de um segmento,  $S_0$  ou  $S_1$ , A dispara um relógio, por um certo período, seja  $T_{\text{timeout}}$ . Se o *Ack* chegar antes de ter decorrido esse período, esse relógio será cancelado; caso contrário, A deverá admitir que o *Ack* não virá, pelo que deverá tomar as medidas apropriadas, nomeadamente: *retransmitir* o (último) segmento enviado. O diagrama temporal contempla este caso: aquando do envio de  $M_2$ , o *timeout* expira, e  $S_0$  é retransmitido; à segunda tentativa, não ocorrem erros de comunicação, e o *Ack* vem a ser devolvido; mas, se voltara a acontecer erros de comunicação, o *timeout* teria voltado a expirar - e A teria que voltar a retransmitir  $S_0$  - como regra, até um número máximo de vezes, após o que seria porventura aconselhável empreender outras medidas (pode acontecer que, por ex., B esteja *power-off*...).

2. uma outra situação possível é a ocorrência de erros/perda no *Ack devolvido de B para A*; considere-se aquele que corresponde ao segmento,  $S_0$ , que transporta  $M_4$ , cfr figura TCP04.b; de novo, A não receberá um *Ack* válido! Pelo que o *timeout* irá disparar - com a sabida consequência: A irá retransmitir  $S_0$ , isto é: B, que já recebeu  $S_0$ , irá recebê-lo outra vez... Quer dizer: existe uma ameaça de *duplicação de informação* em B. Como superá-la? A resposta é: dotando B de uma *variável de estado*,  $w$ , que pode assumir dois valores, 0 e 1, que especifica *qual o segmento que B está aguardando*. No arranque,  $w=0$ , isto é, está-se aguardando o segmento  $S_0$ . Logo após a chegada deste, aquela variável evolui para  $w=1$ ... À medida que forem chegando segmentos válidos cujo campo *Numeração* coincide com o valor dessa variável de estado, assim ela muda de valor... E que acontece se isso não acontecer, como é o caso da retransmissão do segmento  $S_0$  que transporta  $M_4$  - que vai atingir B quando  $w=1$ ? O seguinte: B *descarta* o campo *Dados* desse segmento, mas *devolve Ack*...Desse modo, não chega a haver duplicação de informação em B, e A é alertado para o facto de  $M_4$  já estar em B...

Abra-se um parêntesis para sublinhar: a  $w$ , dá-se o nome de *aresta inferior da janela de recepção*. No caso presente, ela tem dimensão 1 (i.e., em *Stop-and-wait*, B está esperando no máximo *um* segmento de cada vez)

Com a adição de um relógio e da variável de recepção  $w$ , o protocolo fica satisfazendo a especificação, as mensagens  $M_0, \dots, M_4$  estão sendo entregues por ordem... pelo que está ainda por responder a questão enunciada: Onde está a falha do protocolo?

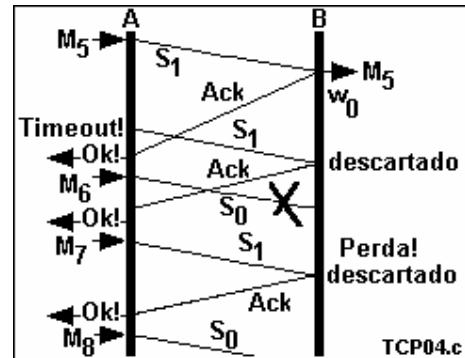
Antes de mais, uma pergunta: Quanto deve ser  $T_{\text{timeout}}$ ? Obviamente, deverá ser suficientemente grande para cobrir o tempo que demora até A receber o *Ack* devolvido por B (senão, existiria o risco de A perder tempo a retransmitir um



segmento bem recebido - apenas porque não esperou o tempo suficiente para receber o Ack devolvido a propósito). Seja  $RTT$  o tempo que medeia entre o instante em que último bit de um segmento foi despachado para o canal de comunicação e o instante em que se acaba de receber o correspondente Ack. Então,  $T_{timeout}$  deverá ser não inferior a  $RTT$ . Mas não deve ser *muito* maior do que  $RTT$ : se, efectivamente, um segmento tem que se retransmitir, porque a sua última tentativa de transmissão se gorou, convém, por mor de eficiência, não hesitar tempo demasiado...

Considere-se então o seguinte cenário: em determinado momento, avalia-se a quanto monta  $RTT$  - e fixa-se  $T_{timeout}$  algo superior a  $RTT$ . Ora pode acontecer que, algum tempo após, em virtude de maior tráfego na rede de comunicação,  $RTT$  advenha bem maior... Pelo que o relógio vem a disparar antes de ter sido recebido o Ack correspondente a um segmento bem recebido. O diagrama temporal TCP04.c esquematiza esse cenário.

O segmento,  $S_1$ , que transporta  $M_5$ , chega a B, é aceite, e é devolvido o correspondente Ack... Este, todavia, atrasa-se a chegar a A; pelo que, decorridos  $T_{timeout}$ , o relógio expira... A retransmite  $S_1$ , a que B responde com novo Ack... Entretanto, o Ack atrasado chega enfim a A; este de imediato dá início à transmissão de  $M_6$ , mediante  $S_0$ ... Logo após, chega o Ack correspondente à recepção do segmento que havia sido retransmitido... A, crendo, erroneamente, que tal Ack diz respeito a  $S_0$ , passa de imediato a enviar a Mensagem  $M_7$ , mediante  $S_1$ ... Se, eventualmente,  $S_0$  se corromper, B ainda permanecerá aguardando com  $w=0$  quando  $S_1$  o atingir - pelo que se descartará dele... O leitor poderá verificar que as mensagens  $M_6$  e  $M_7$  não virão a ser entregues ao Utilizador em B: o protocolo falhará...



2. A segunda alínea supõe um canal que pode *re-ordenar* pacotes... Quer dizer, A pode enviar dois segmentos, seja  $S_0$  -  $S_1$ , e a B chegarem esses dois segmentos mas por ordem inversa:  $S_1$  -  $S_0$ . Terá isso alguma relevância quando a filosofia em jogo é *StopAndWait* - isto é, precisamente quando A pode enviar  $S_1$  somente depois de ter recebido o *Acknowledge* relativo a  $S_0$  - o que parece afastar a eventualidade de  $S_1$  chegar antes de  $S_0$ ? De facto, tem: o último exemplo esquematizado testemunha exactamente o envio de um segmento,  $S_0$ , sem antes A ter recebido o *Acknowledge* correspondente ao segmento que o precedeu,  $S_1$ .

Para a resolução da segunda alínea convém então desenhar um diagrama temporal envolvendo uma situação similar: o disparo de um *timeout* imediatamente seguido da recepção de um Ack atrasado. Naquele que é apresentado em TCP05, assumiu-se que o módulo de numeração é 3. Nele,

- a mensagem  $M_0$  é comunicada de A para B, sem percalços: ela é encapsulada no segmento  $S_0$ , após o que segue o seu caminho até B; este devolve um Ack numerado,  $Ack_1$ , - cujo número (1) é exactamente o valor com que ficou a aresta inferior da sua janela de recepção, i.e., é o número do segmento que B ficou aguardando...

- já com  $M_1$ , transportada em  $S_1$ , vem a existir um contratempo: a correspondente *confirmação de boa recepção*,  $Ack_2$ , atrasa-se...

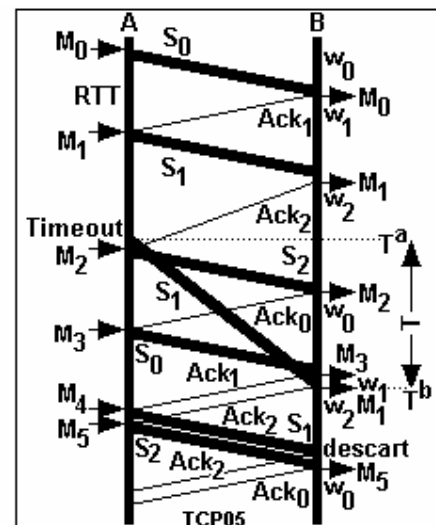
- pelo que o *timeout* vem a expirar:  $M_1$  é retransmitida, de novo encapsulada no segmento  $S_1$ ;

- logo após o envio do *primeiro bit* de  $S_1$ , chega o primeiro bit do Ack atrasado;

- pelo que o receptor pode dar  $M_1$  por bem recebida e desencadear a comunicação de  $M_2$ , mediante  $S_2$ ; B replicará com  $Ack_0$ ;

- sucessivamente, na recepção de  $Ack_0$ ,  $Ack_1$ ,  $Ack_2$ ,... A envia as mensagens  $M_3$ ,  $M_4$  e  $M_5$ , suportadas respectivamente nos segmentos  $S_0$ ,  $S_1$ ,  $S_2$ ... No que se segue, representar-se-á por  $S_i(M_j)$  o segmento  $S_i$  veiculando a Mensagem  $M_j$ .

Não se pode olvidar, entretanto, o segmento,  $S_1(M_1)$ , que havia sido transmitido aquando do expirar do *timeout*: em algum momento - no prazo máximo de  $T$  segs - ele irá findar o seu percurso pela rede... Que acontecerá se chegar a B antes de  $S_2(M_2)$ ? O seguinte: como B já está esperando  $S_2$ , ( $w=2$ ),  $S_1(M_1)$  será *descartado*; e o mesmo lhe sucederá se chegar logo após  $S_2$  (porque então encontrará  $w=0$ ). Mas sorte diferente terá se fizer a sua aparição após  $S_0$  ter chegado a B - isto é, quando B já estiver esperando um segmento numerado precisamente 1! B aceitará como *válido* esse segmento - e passará ao Utilizador uma nova cópia de  $M_1$ ! Em contrapartida, descartará o segmento,  $S_1$ , que transporta  $M_4$ ,...



Este funcionamento é péssimo. A sua causa é óbvia: se, quando  $S_1(M_1)$  chegasse a B, a janela de recepção fôra outro valor que não 1, ele seria descartado! Para o lograr, bastaria escolher um módulo de numeração *maior*. Se ele fôsse 4, por ex., então B ainda estaria com  $w=0$  e descartaria esse  $S_1$  atrasado. A não ser, é claro, que  $S_1(M_1)$  se atrasasse ainda mais... Claramente, o módulo de numeração a escolher depende do atraso sofrido por  $S_1$  na rede... Ele deve ser escolhido por forma a que o avanço da aresta inferior da janela de recepção,  $w$ , para  $w=1$ , deve ser feito após  $S_1(M_1)$  chegar, ou seja, o primeiro bit de  $S_0(M_3)$  deve chegar a B após o primeiro bit de  $S_1(M_1)$ !

Sendo  $R$  pkt/seg o débito máximo, então o tempo que demora a transmitir/receber um pacote é de  $T_{xmt}=1/R$  seg/pkt. Seja  $T^a$  o instante em que é enviado o primeiro bit de  $S_1(M_1)$ , e  $T^b$  o instante em que é recebido o último bit de  $S_1(M_1)$ ; conforme ao enunciado,  $T^b=T^a+T$ . Seja  $M$  o módulo de numeração.

O instante em que chega a B o primeiro bit de  $S_1(M_1)$  é  $T_{M1}=T^b-T_{xmt}$ .

O instante em que é recebido o primeiro bit de  $S_0(M_3)$  é  $T_{M3}=T^a+T_{xmt}+T_{prog}+[T_{xmt}+RTT]*(M-2)$ .

Para um bom funcionamento,  $T_{M1} < T_{M3}$ ; admitindo  $T_{prop} \approx RTT/2$ , deduz-se, por manipulações alébricas triviais,

$$T^b-T_{xmt} < T^a+T_{xmt}+RTT/2+[T_{xmt}+RTT]*(M-2)$$

$$T < 2T_{xmt}+RTT/2+[T_{xmt}+RTT]*(M-2)$$

$$[T-2T_{xmt}-RTT/2]/(T_{xmt}+RTT)+2 < M$$

$$M > [T+3RTT/2]/(1/R+RTT)$$

O termo à direita na inequação acima é uma função *decrecente com RTT*: o seu valor máximo é obtido com  $RTT=0$ ; por substituição, obtém-se  $M > T/(1/R)$  ou seja, enfim,  $M > RT$ .

5. © [2007/09] Considere um protocolo *stop and wait* para o envio de pacotes de dados de uma estação A para uma estação B. A especificação do protocolo impõe apenas que todos os pacotes de dados enviados por A sejam eventualmente recebidos em B, e que B saiba identificar pacotes de dados duplicados, para assim os descartar.

5.1. Suponha que o canal de comunicação bi-direccional que une a estação A à estação B pode corromper ou perder pacotes mas que, quando não são perdidos, os entrega sequencialmente ao destinatário. Os pacotes de dados a enviar de A para B são numerados módulo 4. Admita que os Acks enviados de B para A não são numerados; A envia sucessivamente quatro pacotes; Imagine uma situação - e represente-a por uma diagrama temporal - em que o primeiro e segundo pacotes são bem recebidos, mas o terceiro pacote não vem a ser bem recebido. O que acontece aos seguintes?

5.2. Suponha que o canal de comunicação bi-direccional que une a estação A à estação B pode corromper, perder e re-ordenar pacotes. Os pacotes de dados a enviar de A para B e os respectivos Acks são numerados módulo 4. A estação A transmite a um débito máximo de  $R$  pacotes/seg. Qual o atraso máximo de um *pacote* no canal de comunicação? e qual o atraso máximo de um *Ack*?

$R_1$ :  $T_{AntesDeAckDeS1}$ ,  $S_2$  perde-se

$R_2$ :  $T_{Pkt}=2 RTT + RTT/2 + 4 * 1/R$ ;  $T_{Ack}=2 RTT + RTT/2 + 3 * 1/R$

em que  $T_{Pkt/Ack}$ : atraso máximo entre o envio do 1º bit dum bloco e a recepção do seu último bit.

(Sugestões: a)  $T_{AntesDeAckDeS0}$ , re-envia  $S_0$ , que demora  $T_{Pkt}$  até B, aí chegando logo após  $S_3$ ; b)  $T_{AntesDeAckDeS0}$ , re-envia  $S_0$ , e o respectivo  $Ack_1$  demora  $T_{Ack}$  a chegar a A, confirmando a recepção do pacote  $S_0$  - que, entretanto se perdeu...)

### Go Back N

6. Indique uma vantagem e uma desvantagem do protocolo "*Go-Back-N*" (Voltar-Atrás-N) face ao protocolo "*Selective Repeat*" (de Repetição Selectiva).
7. Indique, justificando, se a seguinte afirmação é verdadeira, ou falsa: "Num protocolo de transmissão fiável "*Go-Back-N*" (Voltar-Atrás-N), a utilização da linha aumenta sempre que se aumenta a dimensão da janela do emissor".
- R: Falsa: a utilização da linha aumenta apenas até 100%. A partir daí, já não adianta aumentar mais a dimensão da janela do emissor
8. Indique duas formas de um protocolo de transmissão fiável "*Go-Back-N*" (Voltar Atrás N), funcionando na presença de erros de transmissão, recuperar da perda de uma confirmação (ACK).
- R: i) *timeout* no emissor, por falta da confirmação; ii) para um fluxo contínuo de pacotes de informação, um ACK cumulativo de um pacote já confirmado.

9. [09E1.1] Considere dois computadores que comunicam através do protocolo *Go-Back-N*. No que se segue, assuma a seguinte notação:  $N_{\min}$  = menor número do pacote enviado mas ainda não confirmado;  $N_{\max}$  = número do próximo pacote a ser aceite do nível superior (i.e. próximo pacote a ser transmitido);  $W$  = tamanho da janela do emissor. Num dado instante verifica-se que, no emissor,  $N_{\min} = 7$ ,  $N_{\max} = 19$  e  $W = 19$ . Diga que acções executa o emissor quando:

9.1. O nível superior (a aplicação) solicita o envio de 5 pacotes novos; [Ex: 0.5]

9.2. O emissor recebe um *ACK* com numeração 2; [Ex: 0.5]

9.3. Expira o temporizador. [Ex: 0.5]

R1: Envia os pacotes, numerando-os 19 a 23, *inclusive* (A Janela actual estende-se de 7 a  $25 = 7 + 19 - 1$ )

R2: Descarta esse *Ack* (Considera-o um *Ack* 'duplicado' atrasado)

R3: Retransmite todos os pacotes, dos nºs 7 a 18, *inclusive* - ou até 23, se já enviou os 5 pacotes novos (O mecanismo *Go-back-N* significa "voltar atrás e retransmitir"). Em qualquer caso, re-armar o temporizador.

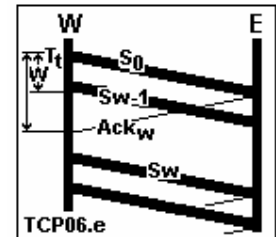
10. [08T1.5] Considere uma comunicação entre dois computadores, gerida por um protocolo *GoBackN*, com atraso de propagação extremo-a-extremo de 100 ms. Admita que o débito máximo permitido é de 10 Mbps e que se usam pacotes com tamanho 500 bytes. Admita também que podem ocorrer erros e perdas mas a rede entrega os pacotes todos por ordem. Além disso, considere desprezáveis os tempos de transmissão dos *ACK*s. Qual o tamanho mínimo para a janela do emissor que permite uma eficiência máxima na comunicação de, pelo menos, 75% ? Nesse caso, indique, justificando, um espaço de numeração possível para os pacotes.

R: 1) 376, pois (cfr fig TCP06.e)

Eficiência =  $0,75 \leq T_{xmt} * w / (T_{xmt} * 2 * 100)$  com  $T_{xmt} = 500 * 8 / (10 * 10^6) \text{ seg} = 0,4 \text{ mseg}$

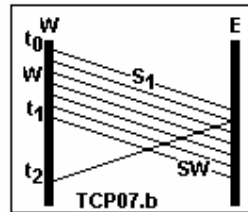
→  $w_{\min} \geq 200,4 * 0,75 / 0,4 = 375,75$  ou  $w_{\min} = 376$

2) Espaço de numeração: 0, 1, 2, ..., 376



11. [07T1] Considere uma comunicação entre dois computadores,  $W$  e  $E$ , gerida por um protocolo *GoBackN*.

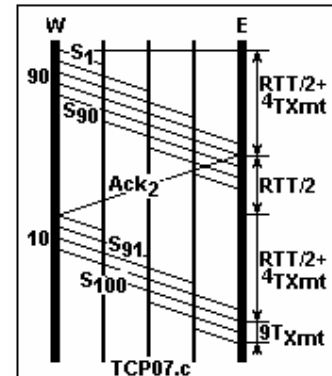
Admita que não há erros nem perdas de pacotes. No instante  $t_0$ ,  $W$  começa a enviar uma Janela de tramas. Admita que  $W$  tem sempre tramas para enviar e que o último bit da primeira trama chega ao destino exactamente no instante em que no emissor se esgota a janela - e, por conseguinte, o emissor interrompe a transferência. Assumindo que a ligação entre  $W$  e  $E$  é directa (isto é, não existem quaisquer comutadores entre  $W$  e  $E$ ) e que, da operação do protocolo resulta uma eficiência de, no máximo, 60 %, diga justificando: qual o número mínimo de bits necessários para representar o número de sequência das tramas? Exprima  $RTT$  em função do tempo de transmissão de cada Trama,  $T_{xmt}$ .



R: O diagrama temporal [TCP07.b] esquematiza a situação.

efic =  $W / [W + (W - 1)] = 0,6 \rightarrow W = 3 \rightarrow \text{Módulo}_{GoBackN} = W + 1 = 4 \rightarrow \text{Nbits} = \log_2 4 = 2$

$RTT = 4 * T_{xmt}$



12. [07T1] Considere agora que entre os dois computadores  $W$  e  $E$  existem 3 nós de comutação e que  $W$  envia um ficheiro de  $1 \times 10^6$  bit, dividido em pacotes de  $10 \times 10^3$  bit, usando o protocolo *GoBackN*. Admita que a janela de emissão de  $W$  é agora de 90 pacotes, os tempos de processamento nos nós e nos extremos são desprezáveis, os pacotes de confirmação têm tamanho desprezável e que o atraso de propagação total entre  $W$  e  $E$  é de 50 ms (portanto o atraso total de propagação de ida-e-volta,  $RTT$ , é de 100 ms). A capacidade disponível ao longo do percurso é de 10 Mbps. Determine o tempo que decorre entre o início da transmissão em  $W$  até à recepção completa do ficheiro em  $E$ .

R: O diagrama temporal [TCP07.c] esquematiza a situação

$N_{pkts} = 10^6 / (10 * 10^3) = 10^2 = 90 + 10$ ;

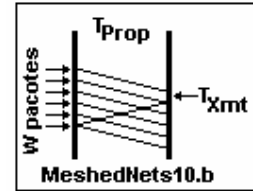
$T_{xmt} = 10 * 10^3 / (10 * 10^6) = 10^{-3}$ ; Latência =  $104 + (10 * 3) + 50 = 167 \text{ mseg}$ .

- 13.[06E3] Considere um caminho de 1000 Km com atraso de propagação de 5  $\mu$ s/Km, sobre o qual se consegue transmitir a um débito máximo de 1 Mbit/s. Suponha que usa um algoritmo de janela deslizante do tipo *Go-Back-N*. Sabendo que cada pacote tem no total 1000 bits, qual a dimensão mínima da janela que conduz, na ausência de erros, a uma eficiência de utilização de 100 % ? Indique, justificando, qual o intervalo de numeração das tramas que deverá ser usado. [Despreze a dimensão dos pacotes de confirmação].

$$R: W = (T_{xmt} + T_{prop} + T_{prog}) / T_{xmt}, \quad c / T_{prog} = 1000 * 5 * 10^{-6} = 5 * 10^{-3} \text{ e } T_{xmt} = 1000 / 10^6 = 10^{-3}$$

$$\rightarrow W = (1 + 5 + 5) / 1 = 11 \rightarrow \text{Mód}_{Numer} \geq (11 + 1).$$

Na prática, o campo de numeração precisa de 4 bits =  $\log_2 12 \rightarrow$  numera-se 0, ..., 15, 0, ...



- 14.[08E3.10] Considere uma ligação entre dois computadores, **W** e **E**, gerida por um protocolo *GoBackN*. Admita que o canal garante a entrega sequencial dos pacotes, mas que eles podem chegar corrompidos. Suponha que a janela de transmissão em **W** e **E** tem dimensão 3. Para confirmar a boa recepção dos pacotes recorre-se a dois métodos: i) pacotes específicos *Ack*; ou ii) procede-se a *piggyback* em pacotes de dados. Em dado momento, constata-se que, em **W**:

- o último pacote enviado por **W** (e único ainda por confirmar) veiculava o número de sequência 300;
- o último pacote recebido por **W**, aceite e confirmado, veiculava o número de sequência 250.

14.1. Quais os números de sequência dos próximos pacotes que, de momento, **W** está autorizado a enviar?

14.2. Quais os números de *acknowledgement* passíveis de ocorrer em pacotes que venham a ser recebidos em **W**?

14.3. Quais os números de sequência passíveis de ocorrer em pacotes de dados que **W** venha a receber?

R1.: 301 e 302; quando expirar o relógio, pode também re-enviar o 300;

R2.: 300 e 301.

R3.: 248, 249, 250, 251, 252 e 253.

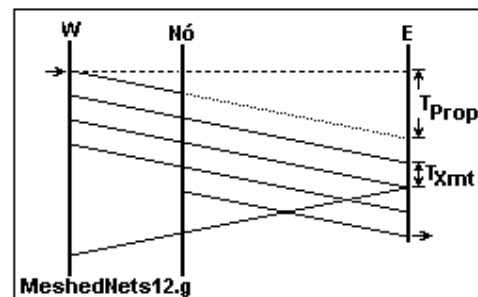
- 15.[09T1.5] Considere a transmissão de uma mensagem entre dois computadores, **W** e **E**, através de um único nó de comutação de pacotes. Os tempos de propagação nas linhas que interligam **W** e **E** ao nó são, respectivamente, 2 e 4 ms. Admita que elas têm a mesma capacidade, **C** bps. Admita ainda que aquela mensagem é repartida em três pacotes, cada qual de comprimento total 1500 *bytes*.

15.1. Supondo que o emissor envia os pacotes em rajada e não ocorrem erros, qual deve ser a capacidade mínima que as linhas devem ter para que o tempo total de transmissão (i.e., o tempo entre o envio do primeiro bit por **W** e a recepção do último bit em **E**) não exceda 30 ms? Considere desprezáveis quaisquer tempos de processamento.

15.2. Suponha agora que o canal entre **W** e **E** está sujeito a erros e que o controlo é feito usando o protocolo *Go-Back-N*. Admita que o tempo de transmissão de um pacote,  $T_{xmt}$ , é 1 ms. Qual o tamanho mínimo da janela do emissor para que a eficiência máxima na linha de saída de **W** seja, pelo menos, 60%? Nestas condições proponha um espaço de numeração para os pacotes.

$$R: 1) [1500 * 8] / [(30 - 2 - 4) * 10^{-3} / 4] = 2 \text{ Mbps (cfr MeshedNets12.g)}$$

$$2) (w * 1) / (1 + 1 + 2 * 6) \geq 0,6 \rightarrow w \geq 8,4 \rightarrow w = 9; \text{ e Espaço de Numeração: } \{0, 1, 2, 3, \dots, 9\}$$



### Selective Repeat

- 16.Ⓢ Considere os protocolos *GoBackN* e *SelectiveRepeat*, desenhados para um canal que garante entrega sequencial de pacotes. Suponha que o espaço para numeração de pacotes tem dimensão **K**, isto é, os pacotes são numerados módulo **K**. Quais as dimensões máximas das janelas?

Resolução:

A resolução de problemas anteriores acentuou já que a situação mais crítica ocorre quando o *timeout* dispara, com a conhecida reacção de **A**: retransmitir o segmento *ainda não confirmado mais antigo*. A situação é crítica porque, quando reage, **A** não sabe se o segmento *ainda por confirmar* foi ou não bem recebido por **B**. Quais as consequências desta "ignorância"? a propósito, convém esquematizar o diagrama temporal respeitante à situação: envio de uma janela

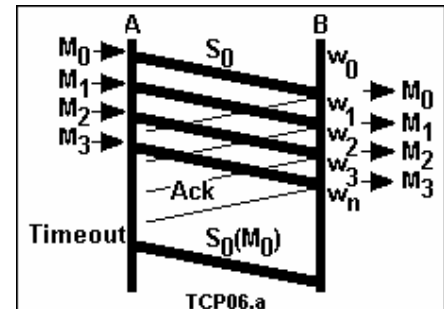
completa, de A para B, cuja transmissão é bem sucedida - seguido de um *timeout* (porque os Acks devolvidos se perdem ou se atrasam)...

Nos diagramas temporais apresentados, assumiu-se que o tamanho da *janela de transmissão* é  $W_T=4$ , e que de início, a aresta inferior da janela de recepção de B está a  $w=0$

### 1. GoBackN

O receptor tem *janela de recepção de dimensão*  $W_R=1$ , i.e.: B está disponível para receber *um* segmento apenas ( $S_0$ ). À medida que vão chegando os vários segmentos, "roda" sucessivamente para  $w=1, w=2, w=3, w=n$ . Qual deverá ser o último valor,  $n$ ? claramente, existem duas alternativas: 4, ou 0. A questão é saber qual delas é a correcta...

A decisão é ditada pelo comportamento de A: quando ocorre o *timeout*, ele retransmite a mensagem  $M_0$ , embebida no segmento  $S_0$ , cfr figura TCP06.a. Se  $w$  estivesse a 0 quando esse segmento chegasse a B, ele seria aceite como *válido* e o seu conteúdo,  $M_0$ , passado ao Utilizador: haveria *duplicação* da informação! Pelo que nesse momento  $w$  não deverá estar a 0: há que escolher a alternativa  $n=4$ ; por conseguinte, para numerar os segmentos há que dispôr de pelo menos 5 (cinco) identificadores diferentes: o módulo de numeração deve ser pelo menos 5 - ou, mais geralmente,  $K \geq W_T + 1$  ou  $W_T \leq K - 1$ .

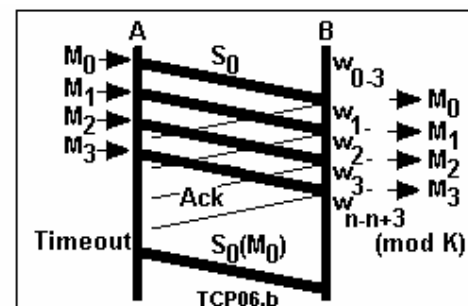


### 2. SelectiveRepeat

O receptor tem *janela de recepção de dimensão*  $W_R=4$ . B está disponível para receber os segmentos  $S_0, S_1, S_2$  e  $S_3$ . Depois, à medida que vão chegando os vários segmentos, "roda" sucessivamente para  $w=1, w=2, w=3, w=n$ . Quando  $w=1$ , é tentador afirmar que B ficou disponível para receber os segmentos  $S_1$  a  $S_4$ ; mas tal seria estabelecer *a priori* a necessidade de pelo menos 5 identificadores, o que está por demonstrar... Se bastassem 4 identificadores apenas, B teria ficado disponível para receber os segmentos  $S_1, S_2, S_3$  e  $S_0$ . Ambas as situações ficam cobertas afirmando que, quando a aresta inferior da janela de recepção é  $n$ , B está disponível para receber os segmentos cujos identificadores pertençam ao conjunto  $W_n = \{n, n+1 \text{ (módulo } K), \dots, n+3 \text{ (módulo } K)\}$  em que cada termo  $n+i \text{ (módulo } K)$  significa o resto da divisão de  $n+i$  por  $K$ .

Qual deverá ser o último valor,  $n$ ? claramente, existem duas alternativas: 4, ou 0. A questão é saber qual delas é a correcta...

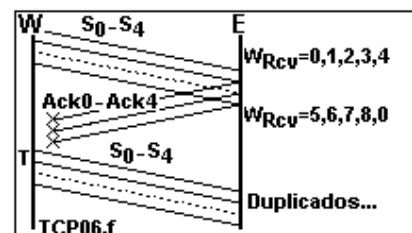
A decisão é novamente ditada pelo comportamento de A: quando ocorre o *timeout*, ele retransmite a mensagem  $M_0$ , embebida no segmento  $S_0$ , cfr figura TCP06.b. Se 0 fizesse parte de  $W_n$ , então, quando esse segmento chegasse a B, ele seria aceite como *válido* e o seu conteúdo,  $M_0$ , passado ao Utilizador: haveria *duplicação* da informação! Pelo que nesse momento  $W_n$  não deverá conter 0: há que escolher a alternativa  $W_n = \{4, 5, 6, 7\}$ ; por conseguinte, para numerar os segmentos deverão existir pelo menos 8 (oito) identificadores diferentes: o módulo de numeração deve ser pelo menos 8 - ou, mais geralmente,  $K \geq 2 * W_T$  ou  $W_T \leq K / 2$ .



- 17.[08E1.3]Suponha um protocolo *Selective Repeat* em que o tamanho da janela é 5 e em que as tramas têm numeração modulo-9. Admita que o canal de comunicação entre emissor e receptor não troca a ordem das tramas. Justifique se este protocolo conduz ou não a um funcionamento correcto e, no caso negativo, apresente um exemplo de mau funcionamento e uma solução para resolver o problema.

R: Em *Selective Repeat*, o Tamanho da Janela não pode exceder metade do Módulo de Numeração. Portanto, o protocolo dado não funciona correctamente. Solução: diminuir o tamanho da janela, para 4, ou aumentar o módulo, para 10.

Na fig TCP06.f, exemplifica-se um acidente de percurso: W transmite 5 Segmentos; E recebe-os, roda a janela (ficando a aguardar os 5 Segmentos seguintes), e devolve os correspondentes Acks. Estes perdem-se todos. Em W, expira o *Timeout* de retransmissão do 1º Segmento. Ele é retransmitido e aceite: obteve-se um duplicado!



18. [2007/09] Considere um protocolo de transmissão fiável, funcionando na presença de erros de transmissão, mas sobre uma rede que entrega os pacotes todos por ordem. Caso se pretenda poder ter 30 pacotes transmitidos em *pipeline*, à espera de confirmação, indique justificadamente quantos números de sequência distintos, no mínimo, têm de ser utilizados na numeração dos pacotes, por forma a que o protocolo funcione correctamente para os seguintes casos:

18.1. Protocolo "Go-Back-N" (Voltar Atrás N).

18.2. Protocolo *Selective-Repeat* (Retransmissão Selectiva).

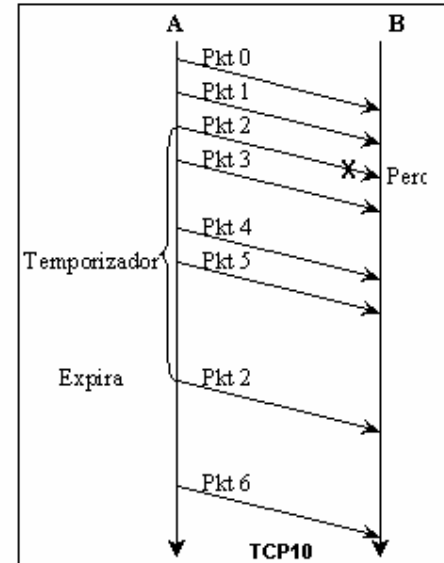
R: GoBackN= 31; SelectiveRepeat=60

19. Considere o diagrama temporal da figura TCP10, que ilustra os pacotes enviados por um protocolo de transporte a enviar dados de A para B, mas onde não estão ilustradas as respostas de B. No início, A tem dados a enviar suficientes para encher 7 pacotes. Em toda a comunicação, apenas o pacote 2 é perdido.

19.1. Sabendo que o protocolo de transporte é um dos seguintes protocolos: *stop-and-wait* (parar e esperar), *Go-Back-N* (voltar atrás N) ou *Selective-Repeat* (Retransmissão Selectiva), que protocolo de transporte está a ser utilizado?

19.2. Qual a dimensão da janela do emissor?

19.3. Complete o diagrama temporal com os pacotes enviados de B para A, admitindo que B confirma imediatamente todos os pacotes que recebe e que as confirmações incluem o número do pacote que B pretende confirmar

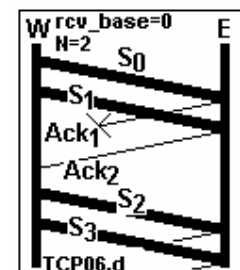


20. [08T1.4] Que acções deve um emissor do protocolo *Selective Repeat* executar quando este recebe um pacote de confirmação (ACK) contendo o número de sequência *rcv\_base+N*, em que *rcv\_base* representa o menor número do pacote enviado mas até ao momento ainda não confirmado pelo receptor. Diga também que acções deverá um emissor do protocolo *Go-Back-N* executar perante o mesmo evento e apresente uma situação possível num diagrama temporal onde se verifique o cenário descrito (sómente para o protocolo *Go-Back-N*).

R: O número K em Ack pode significar: ou i) é *cumulativo*, i.e., o receptor recebeu *todos* os pacotes até ao de número K-1, *inclusivé*, ou ii) é *individual*, i.e., o receptor recebeu o pacote K.

GBN com Ack *cumulativo*: o emissor considera bem recebidos os pacotes enviados com números *rcv\_base*, *rcv\_base+1*, ... até *rcv\_base+N-1*, *inclusivé*. A janela do emissor roda de N: se estava esgotada, pode transmitir mais N pacotes (cfr fig TCP06.d). Cancela o *timeout* de retransmissão; mas, se há algum pacote por confirmar, re-arma-o.

SR com ack *individual*: o emissor considera recebido o pacote de número *rcv\_base+N*; cancela o *timeout* de retransmissão *desse* pacote. Se N for zero, roda a janela de 1; se já tiver recebido os acks dos pacotes de números *rcv\_base+1*, ... *rcv\_base+i*, roda a Janela de um total de *i+1*: pode transmitir os *i+1* pacotes seguintes.



21. Considere um caminho de 5000 Km de comprimento com atraso de propagação igual a 5  $\mu$ seg/Km, e sobre o qual consegue transmitir a um débito máximo de 100 Mbps. Suponha que usa um algoritmo de janela deslizante para controlo de erros e controlo de fluxo. Cada pacote tem 1000 bits. Se usar *Go-Back-N*, qual a dimensão mínima da janela, em número de pacotes, que garante uma eficiência de utilização do caminho de 100%? E se usar *Selective-Repeat*?

Resolução:

Conforme ao método seguido na resolução de problemas similares, a primeira etapa é: *desenhar um diagrama temporal que esquematize fielmente a situação descrita*. O diagrama temporal correspondente à situação descrita encontra-se esquematizado em TCP07. Nele, foi tido em consideração o seguinte:



- no instante  $T_0$ , A desencadeia o envio do segmento  $S_0$ , bit após bit;

- no instante  $T_1$ , finda o envio desse segmento; o tempo nisso despendido,  $T_{xmt}=T_1-T_0$ , dito *tempo de transmissão*, é dado por  $1000/(100 \cdot 10^6)=10\mu\text{seg}$ ;

- logo após, sem qualquer interrupção, A desencadeia o envio do segmento seguinte,  $S_1$ ;

- este "ciclo" de envio contínuo de segmentos prossegue até A ter completado a transmissão dos  $w$  segmentos cobertos pela janela de transmissão (por mor de comodidade de desenho, no diagrama admitiu-se que ela tem dimensão  $w=2$ ; pelo que o último segmento a ser transmitido é  $S_1$ , cuja transmissão cessa no instante  $T_2$ );

- transmitido o último segmento que faz parte dessa janela, A aguarda que lhe chegue, do outro extremo, B, a *confirmação de boa recepção* de, ao menos, um segmento...

- entretanto,  $S_0$  propaga-se... O seu primeiro bit chega a B no instante  $T_6$ ; e o seu último bit chega a B no instante  $T_7$ ; de imediato, B testa o segmento chegado - e, não detectando a presença de erros de transmissão, devolve a *confirmação de boa recepção*, sob a forma de um segmento,  $Ack_1$ ;

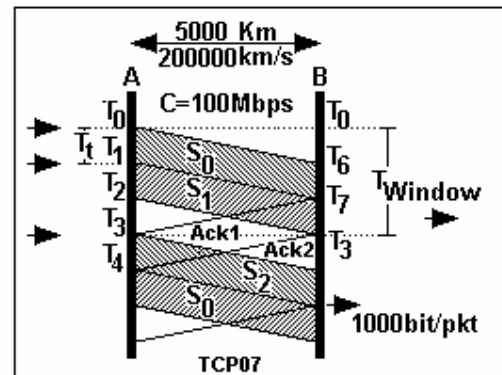
- no instante  $T_3$ , esse segmento,  $Ack_1$ , chega a A; de imediato, A desencadeia o envio do segmento seguinte,  $S_2$ ; pouco depois, em  $T_4$ , chega nova confirmação de recepção,  $Ack_2$ , e A envia o segmento seguinte... e assim continua até esgotar de novo a janela de transmissão...

Do diagrama temporal, é patente que o comportamento genérico de A envolve duas "fases": uma primeira, entre  $T_0$  e  $T_2$ , de envio de uma janela completa de *informação*, e uma outra, entre  $T_2$  e  $T_3$ , de *pausa*. Em ordem a uma utilização *eficiente* dos recursos de transmissão disponíveis, convém que a duração da primeira fase seja bem maior que a duração da segunda; o óptimo seria, aliás, que esta última tivesse duração nula... Por eficiência de utilização do caminho designa-se precisamente a *fracção de tempo* que o caminho é de facto utilizado para transmitir informação *útil*; no caso, essa fracção volta-se em:  $(T_2 - T_0)/(T_3 - T_0)$ . O problema reduz-se então a determinar quanto tempo medeia entre  $T_0$  e  $T_2$  e entre  $T_0$  e  $T_3$  - o que, a partir do diagrama temporal, se volta em simples *geometria euclidiana*:

O período de tempo entre  $T_0$  e  $T_2$  mede-se sobre a *vertical em A*: é  $w \cdot T_{xmt}$ . Quanto ao período de tempo entre  $T_0$  e  $T_3$ , mede-se sobre a *vertical em B*: ele será o somatório dos tempos parciais  $[T_0 \text{ a } T_6] + [T_6 \text{ a } T_7] + [T_7 \text{ a } T_3]$ , que se volta em  $T_{\text{Propag}} + T_{xmt} + T_{\text{Propag}} = 2 \cdot 5000 \cdot 5 + 10 = 50010 \mu\text{seg}$ .

A eficiência em causa é, por conseguinte,  $w \cdot 10 / 50010$ ; igualando esta expressão a 1 (o objectivo é que a eficiência seja de 100%), deduz-se  $w=50010/10=5001$ . De facto, o leitor poderá verificar que, se a janela de transmissão for superior ao valor encontrado, ainda assim a eficiência continuará a ser 100%; isso corresponderá à *chegada de Ack*, *acontecer antes de terminada a transmissão da janela*: A não chega nunca a ingressar na fase de "pausa"! Esse valor encontrado é, pois a janela *mínima*.

Notas: Na resolução acima, considerou-se desprezável o tempo de transmissão de um Ack; se essa assumption não for válida, então, ao tempo  $T_3-T_0$ , haverá que adicionar  $T_{\text{Ack}} = \text{Comprimento}_{\text{Ack}} / (100 \cdot 10^6)$ . A solução é a mesma, quer se esteja em *Go-Back-N* ou *Selective-Repeat*; a diferença ocorreria apenas na etiquetagem do diagrama: após o envio de  $S_2$ , (que é respondido com  $Ack_2$ ), dever-se-ia, em *Selective-Repeat*, enviar  $S_3$ ...

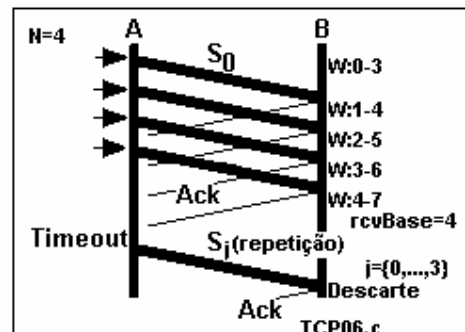


22. Repita o problema anterior, mas admitindo que usa *Selective-Repeat*, e basta uma eficiência de utilização da linha de 80%.

23. [07T1] Que deve um receptor do protocolo *Selective Repeat* fazer quando recebe um pacote com número contido entre  $[rcv\_base-N, rcv\_base-1]$ , em que  $rcv\_base$  representa o menor número do pacote ainda não entregue ao nível acima. Justifique, através de um diagrama temporal, como é possível este evento ocorrer. Explique também se é possível este evento ocorrer com o protocolo TCP.

R: O diagrama temporal [TCP06.c] esquematiza o evento, para  $N=4$  e  $rcvBase=4$ . Por cada Pacote bem recebido, é devolvido o pertinente Ack (se se receber  $S_2$ , envia-se:  $Ack_2$  em *Selective Repeat* e  $Ack_3$  em TCP). O evento pode ocorrer se *expirar ao menos um dos Relógios dos últimos N Pacotes já enviados*; causas podem ser:

- o respectivo *Timeout* é "curto demais" ou:





27.[10T1.2] Considere um cenário em que dois computadores, *A* e *E*, comunicam através da Internet usando um protocolo de Repetição Selectiva (*Selective-Repeat ARQ*). Suponha que não há troca de ordem dos pacotes. Suponha também que o tamanho da janela usado pelo emissor e receptor é 5 e que a numeração das tramas é feita com um campo de 3 bits. Diga se este protocolo terá ou não um funcionamento correcto (justificando no caso afirmativo ou, caso contrário, apresentando um exemplo de mau funcionamento).

R: O protocolo dado não funciona correctamente. Justificação: em Selective Repeat, o Módulo de Numeração não pode ser inferior ao dobro do Tamanho da Janela – o que no caso significa que não pode ser inferior a 10. Acontece que o campo de numeração é de 3-bits: permite apenas  $2^3=8$  números de sequência. Solução: diminuir o tamanho da janela, para 4, ou aumentar o tamanho do campo de numeração, para 4-bits.

Eis um acidente de percurso: *A* transmite 5 Segmentos  $\{S_0-S_4\}$ ; *E* recebe-os, roda a janela (ficando a aguardar os 5 Segmentos seguintes,  $\{S_5-S_9\}$ ), e devolve os correspondentes Acks,  $\{Ack_0-Ack_4\}$ . Perde-se  $Ack_0$ . Em *W*, expira o *Timeout* de retransmissão do 1º Segmento  $\{S_0\}$ . Ele é retransmitido e aceite: obteve-se um duplicado!

28.[10E2.2] Considere uma ligação entre *W* e *E*, gerida por um protocolo de janela deslizante “*selective repeat*”. Em dado momento, observa-se que *W* tem 5 blocos para transmitir, a janela de transmissão é 3 e a janela de recepção é 2. Na subsequente troca de segmentos, perdem-se o 1º bloco e o *acknowledgement* da retransmissão do 1º bloco. A partir daí não há mais perdas nem surgem mais dados novos para transmitir.

28.1. Qual o estado da janela do emissor após a recepção de cada *acknowledgement*?

28.2. Quais os pacotes que chegam ao receptor e não podem ser logo entregues à aplicação?

28.3. Qual a menor gama de números de sequência que pode ser usada?

28.4. Seja  $RTT=3$  ms,  $T_{Xmt}=1$  ms (tempo de transmissão de um bloco) e  $T_{Out}=5$  ms (intervalo do temporizador). Qual o tempo que decorre entre o início da transmissão do 1º bit e a recepção do *acknowledgement* que confirma a recepção do último bit por *E*?

R1: cfr o diagrama temporal em TCP06.j: mantém-se em  $\{0-2\}$  até à confirmação de  $S_0$ , em que transita para  $\{3-0\}$ ; na recepção dos dois seguintes Acks, transita sucessivamente para  $\{4-1\}$  e  $\{0-2\}$ .

R2: O terceiro pacote a ser transmitido: a janela de recepção não o cobre nesse momento ( $t_2$ );

R3:  $\{0$  a  $4\}$  (Repare-se: em  $t_{2+}$  a janela não pode ainda cobrir o número ‘0’, pois  $S_0$  pode vir a ser – como sucede no ex. – retransmitido; o pior cenário ocorre quando  $S_0$ ,  $S_1$  e  $S_2$  são todos aceites, e são devolvidos os correspondentes Acks, mas eles perdem-se todos: é necessário que a janela de recepção fique de tal modo que o receptor venha a rejeitar eventuais pacotes numerados  $\{0,1$  ou  $2\}$  – para não haver duplicados; isso consegue-se se ele rodar a janela para  $\{3-4\}$ , implicando ao menos 5 números de sequência).

R4:  $T_{Xmt}+T_{Out}+T_{Xmt}+T_{Out}+T_{Xmt}+RTT+2*T_{Xmt}+RTT=5*T_{Xmt}+2*T_{Out}+2*RTT=5+10+3=21$  ms

