

Relatório 04

Cristiano Medeiros Dalbem Mauricio Rodrigues Cruz

26 de outubro de 2011

1 PARTE 1 - Códigos de detecção de erros

1.1 Exercícios

1. Um sistema de telecomandos possui ao todo 8 comandos, cada um constituído de 3 bits. Para maior segurança (ambiente wireless) você resolveu repetir a informação de cada comando 3 vezes em cada transmissão. Pergunta-se:

(a) Qual a distância de Hamming deste conjunto de comandos?

3, pois é necessário que no mínimo um bit (sempre o mesmo) de cada uma das repetições seja alterado para que o comando seja interpretado como outro.

(b) Qual a capacidade de tratamento de erros destes comandos?

Detecção: d erros com $DH = d+1$ \Rightarrow detecta 2 erros.

Correção: e erros com $DH = 2e+1$ \Rightarrow corrige 1 erro.

2. Um determinado sistema utiliza as seguintes palavras de código para enviar suas quatro possíveis informações:

C0	10101
C1	11100
C2	00000
C3	11111

(a) Qual a distância de Hamming mínima do sistema?

2 bits.

- (b) Com os 5 bits disponíveis é possível escolher palavras de códigos que conduzam a uma distância mínima maior que a do sistema cima? Em caso afirmativo dê um exemplo.
Não.
- (c) Para a configuração dada é possível corrigir erros? Por quê?
Não, pois o mínimo de bits de DH para correção de erros é 3 (1 erro).
3. Utilizando o polinômio CRC-CCITT ($x^{16} + x^{12} + x^5 + 1$), gere um código CRC de 16 bits para uma mensagem formada por um bit 1 seguido de quinze bits 0.

$$G(x) = x^{16} + x^{12} + x^5 + 1$$

$$M(x) = x^{15}$$

$$x^r * M(x) \% G(x) = x^{31} \% x^{16} + x^{12} + x^5 + 1 = x^{12} + x^{11} + x^7 + x^4 + x^3$$

$$CRC = 0001100010011000$$

1.2 Experiências

1. O simulador ReedSolomon-Test permite verificar o funcionamento da correção de erros com o Reed-Solomon.

- (a) O que determina o número no campo Fehler Korrigierbar ?

Determina o numero de simbolos que o reed-solomon pode corrigir com um segmento (n) fixo, ajustando apenas o tamanho do segmento de redundancia de maneira a satisfazer:

$$FehlerKorrigierbar = (n - k)/2$$

- (b) Qual é o número máximo de bits (eficiência do algoritmo) que podem ser corrigidos com Fehler Korrigierbar = 3? a) no melhor caso de erros na linha; b) no pior caso de erros.

melhor caso: 12 bits errados divididos em 3 simbolos. Exemplo: xxxx xxxx xxxx 0101

pior caso: 4 bits errados divididos em 4 simbolos distintos. Exemplo: x010 1x00 02x3 053x

- (c) No caso dos CDs, erros em rajadas maiores que a capacidade de correção de cada bloco (frame) são comuns. Que técnica poderíamos utilizar para evitar a perda de dados com rajadas grandes sem ter que aumentar o número de bits de redundância da codificação? Explique.

Técnica de interleaving: reordena os dados de maneira que os erros se distribuam por diferentes pacotes, dessa maneira evitando de estourar o máximo de bits errados que o código permite corrigir em cada símbolo.

2. Baixe o simulador da página da disciplina, leia o help e efetue os testes indicados, utilizando os arquivos como base. Execute passo a passo os comandos mostrados no help. Edite o arquivo manualmente e veja se ele consegue corrigir o número de erros prometido. O relatório deve conter o resultado obtido passo a passo, aumentando o número de símbolos errados até o sistema não recuperar mais os erros. Justifique. Os passos a serem efetuados são:

- RS Input: Utilizamos os valores default de 249 símbolos de entrada e 6 para paridade (máximos de 3 erros corrigíveis).
- Encode: Utilizamos o arquivo de entrada data4.en modificado para casar com a configuração da etapa anterior. Verificamos que o arquivo de saída desta etapa tinha, em comparação com o inicial, os 6 símbolos adicionais no final do stream, que são a codificação (no caso, 2c 7d c8 2b 8 23).
- Insert Errors: ao invés de usar essa opção, inserimos erros manualmente. Testamos com 2, 3 e 4 símbolos com erros.
- Decode: O programa devidamente corrigiu os erros quando estes eram em número de 2 ou 3, mas com 4 ele não conseguiu.
- Compare: essa função mostra exatamente o número de diferenças, quando o algoritmo não consegue corrigir os erros.

2 PARTE 2 - Controle de fluxo

2.1 Debate em Aula

1. O que é controle de fluxo? Como é feito em TCP? E em UDP?

Controle de fluxo é o controle por Software por parte tanto de Cliente quanto Servidor que garantirá nos pacotes enviados a sua chegada, seu ordenamento correto e ausência de réplicas. Em TCP é utilizado Janela Deslizante, e em UDP não há controle de fluxo.

2. O que é janela deslizante? Qual a consequência de uma janela grande demais? E pequena demais?

É uma classe de protocolos que visa melhorar a eficiência de entrega de pacotes mantendo a segurança de protocolos mais ineficientes como o Stop-and-Wait. É necessário otimizar o trade-off de tamanho de janela, pois uma janela pequena demais não aproveitará o paralelismo do envio, enquanto uma grande demais causará reenvio de muitos pacotes no caso de erros.

3. Como relacionar tamanho da janela com banda? Como relacionar RTT com banda?

Uma banda alta é melhor aproveitada com uma janela grande, pois enquanto esperamos o ACK dos pacotes (que depende da latência da rede), podemos ir mandando outros pacotes. Mas isso depende não tanto de uma banda grande ou pequena, mas sim da relação banda/RTT: um RTT alto não prejudica tanto no caso de uma banda pequena, podendo ser usada uma janela menor.

4. Com RTT (Round Trip Time) alto, é mais eficiente tamanho de janela grande ou pequena? Por quê?

Grande, pois enquanto os pacotes estão trafegando de um lado pro outro e os ACKs demoram a chegar, aproveitamos mandando outros pacotes.

5. O que é "tempo de timeout"? Qual o problema para determinar o tempo de timeout (retransmit timer) em redes reais? Qual a solução?

É o tempo de espera pelo ACK de um pacote que enviamos antes de reenviá-lo. O problema é que um timeout muito grande causará muito atraso quando o ACK tiver sido perdido na Rede, enquanto um timeout muito curto causará reenvios desnecessários quando estiver mais rápido que o tempo de ACK leva pra voltar. A solução é usar um timeout pelo menos mais longo que o RTT, o que evitaria o segundo caso.

6. Qual a consequência das perdas num protocolo real, tipo TCP, em termos de controle de fluxo? Outra forma de perguntar: o que o protocolo assume quando tem perdas? Lembre que as perdas acontecem nas duas direções (na transmissão do pacote ou no ACK).

Ele assume que ocorreu congestionamentos na rede, e a consequência é a redução do tamanho da janela de maneira que minimize os erros na próxima transmissão.

2.2 Exercícios

1. Um canal tem uma taxa de 4 kbit/s e um retardo de propagação de 20 ms. Para que faixa de variação de tamanho de quadros a técnica stop-and-wait proporciona uma eficiência de pelo menos 50%? Pode utilizar como base as fórmulas vistas na parte teórica ou no apêndice 7A do livro Data and Computer Communications 7a Edição (ou da sua 6a Edição) - de William Stallings.

$$U = \frac{1}{1 + 2 \frac{t_{propagacao}}{t_{frame}}} = 0.5$$

$$t_{frame} = 2t_{propagacao} = 40ms$$

$$t_{frame} = \frac{tamanho_{frame}}{banda} \Rightarrow tamanho_{frame} = 40ms * 4kb/s = 160kb$$

2. Quadros de 1.000 bits são enviados por um canal de 1 Mbps usando um satélite geoestacionário cujo tempo de propagação a partir da Terra é 270 ms. Qual é a capacidade máxima do canal utilizando os seguintes mecanismos:

(a) Stop-and-wait?

$$\frac{1000bits}{(270 + 270 + 1)} = 1,8Kbits/s$$

(b) Go-back-N com número de sequência de 3 bits?

$$\frac{7000bits}{(270 + 270 + 7)} = 12,79Kbits/s$$

- (c) Go-back-N com número de seqüência de 4 bits?

$$\frac{15000bits}{(270 + 270 + 15)} = 27,02Kbits/s$$

2.3 Experiência

1. Explore o demo Sliding Window em http://www2.rad.com/networks/2004/sliding_window/demo.html. Eventualmente será necessário entrar via web archive (<http://web.archive.org>). Explique apoiado com imagens da execução da tela do computador o funcionamento do demo sliding window. No mínimo, os seguintes itens devem ser explicados:
 - (a) Funcionamento de janela deslizando.

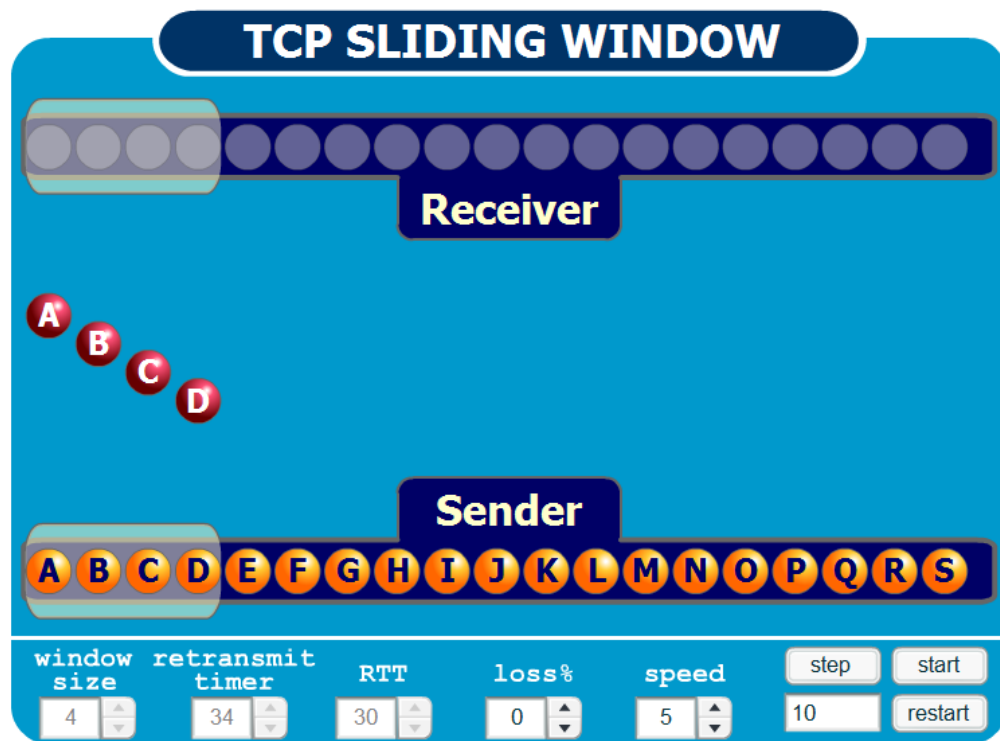


Figura 1: O tamanho da janela define o máximo de pacotes que podem ser enviados antes de parar para esperar por ACKs.

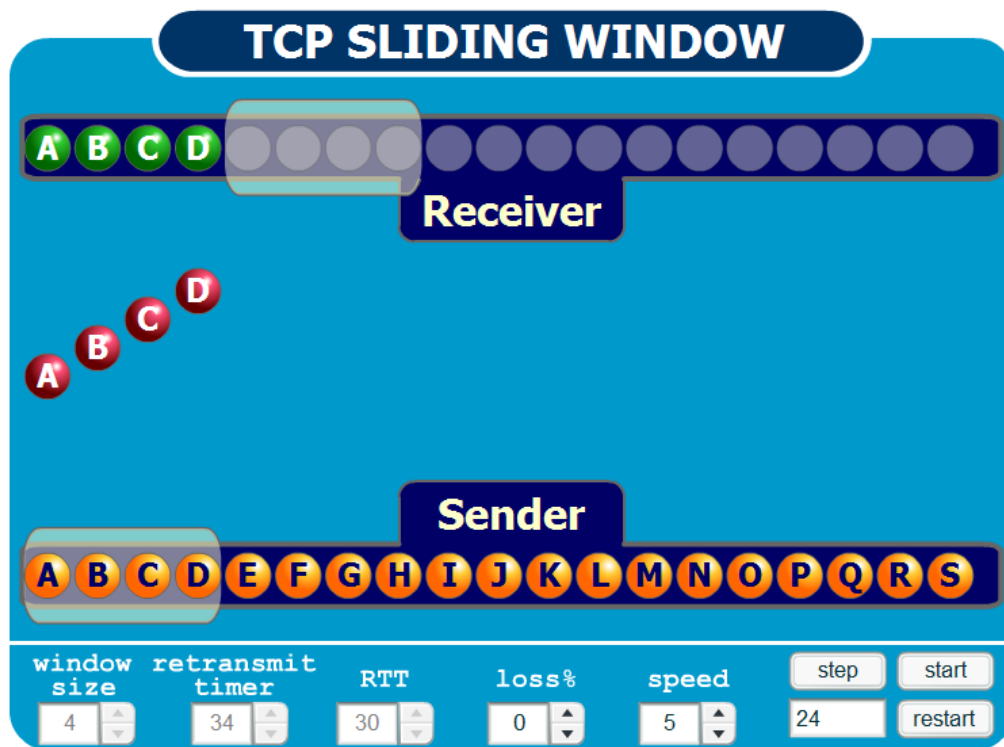


Figura 2: O cliente, que recebe os pacotes, também tem uma janela deslizando, que se move conforme vai recebendo os pacotes.

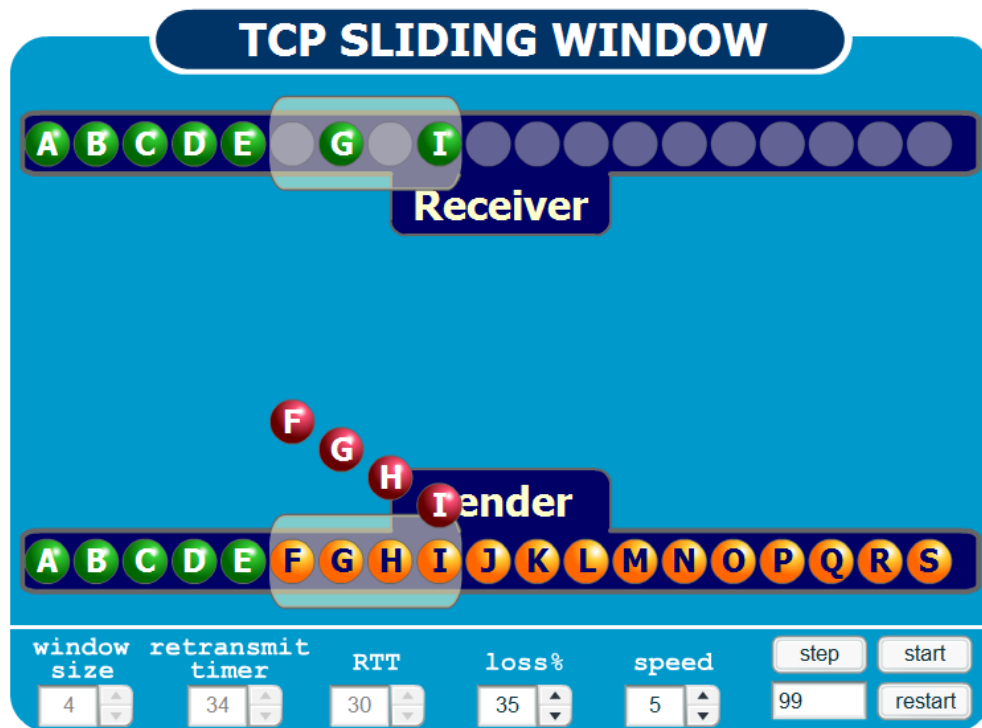


Figura 3: Cada pacote tem um tempo de timeout, que define o quanto ele deve esperar por um ACK antes de ser reenviado. Serve para quando o próprio pacote foi perdido na Rede, então o cliente nunca o receberá, ou mesmo quando o cliente recebeu mas seu ACK foi perdido. Na figura, todos pacotes estão sendo reenviados, pois todos ACKs referentes a eles foram perdidos.

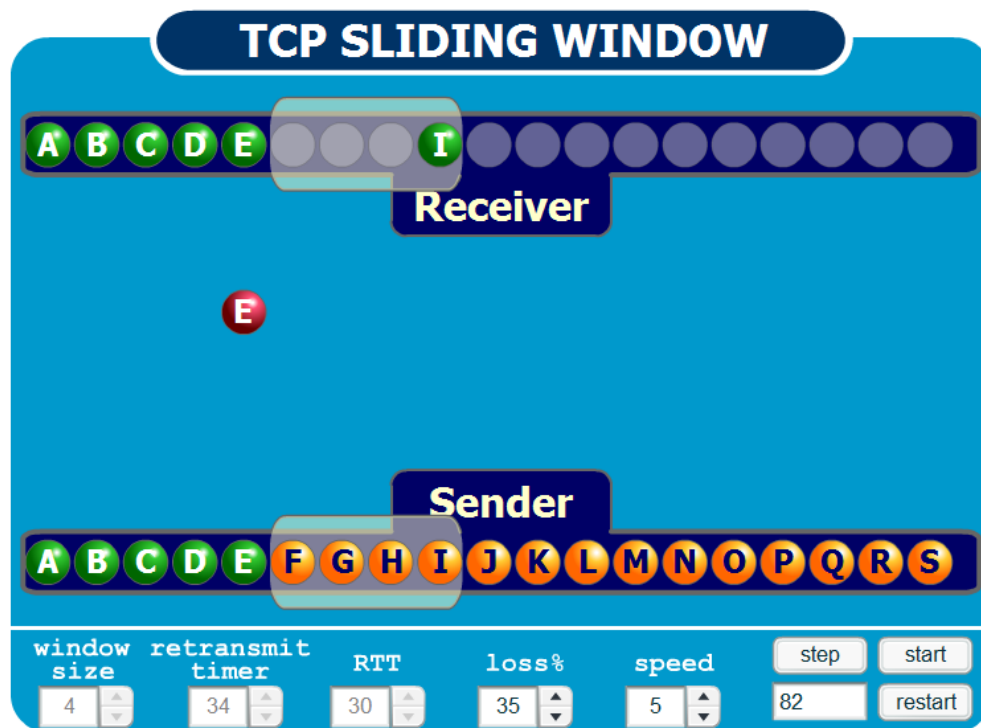


Figura 4: Quando o cliente recebe um pacote adiantado (no caso, I), ou seja, entre o último recebido corretamente e o recebido há lacunas, o cliente envia um ACK referente ao último que foi recebido corretamente (no caso, E), mas tem um buffer que guardará aquele fora da serialização.

- (b) Comente três diferenças desse simulador em relação ao TCP utilizado na WEB.

O simulador não demonstra a capacidade de autogerenciamento do tamanho da janela.

2. Acesse o simulador <http://www.osischool.com/protocol/Tcp/slidingWindow/index.php>. Verifique a simulação. Explique o mecanismo de crescimento da janela deslizante, definindo o slow-start.

O mecanismo serve para autoajustar o tamanho da janela para quando não se conhece a Rede em uso, isto é, não se sabe qual sua banda e sua

taxa de erros. Assim, o mecanismo aumenta gradativamente o tamanho da janela enquanto estiver tudo funcionando bem, isto é, recebendo ACKs de todos pacotes. O crescimento, que é exponencial, ocorrerá até que haja um erro na rede (o que é interpretado como um congestionamento), ou até que um certo treshhold seja alcançado, quando começará a etapa de crescimento linear. Essa nova etapa continuará até que seja detectado congestionamento.