

## Trabalho de Avaliação e Desempenho (V1.0)

Primeiro Semestre de 2010

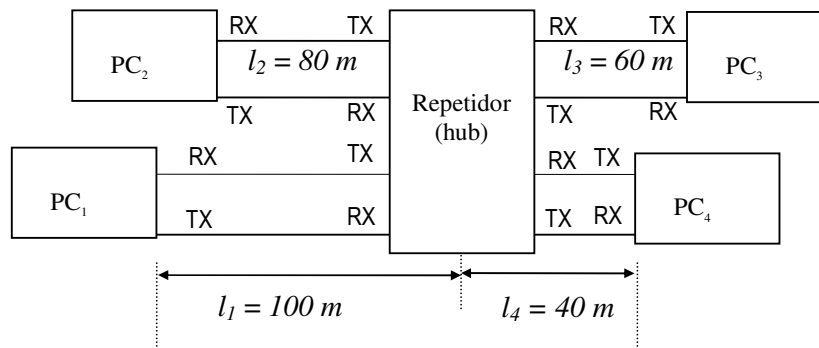
05/2010

### 1 Objetivo

O aluno deverá simular o comportamento de 4 estações acessando uma rede Ethernet a 10 Mbps, usando uma simulação orientada a eventos discretos em tempo contínuo.

### 2 Arquitetura a ser simulada

Quatro PCs estão conectados a um repetidor (hub) com distâncias variadas e iguais a  $l_1=100m$ ,  $l_2=80m$ ,  $l_3=60m$  e  $l_4=40m$ .



A propagação elétrica no meio físico é assumida de  $5 \mu s/km$ .

A conexão de cada estação é feita com dois pares trançados, um para transmissão (tx) e outro para recepção (rx). Qualquer sinal transmitido por uma estação em tx é recebido pelo rx correspondente no hub e retransmitido pelo hub em todos os tx. Todas as estações, inclusive a que transmitiu, irão receber em seu rx o sinal retransmitido, após o atraso de propagação devido.

Uma estação detecta colisão se o sinal recebido não é o mesmo que ela está enviando ou se sinais superpostos existem na sua porta rx (existem várias formas de detectar colisão por hardware e não nos interessa este detalhe no momento). Vamos assumir que a detecção de colisão é feita na estação em tempo ZERO. Uma estação não colide consigo mesma, apesar de receber sua própria transmissão.

Uma estação transmitindo, sempre está com seu circuito de detecção de colisão ativado. A estação, após colisão, reage adequadamente suspendendo sua transmissão e gerando um sinal de reforço de colisão, como veremos mais adiante.

Observe que devido à assimetria da topologia, a transmissão de uma estação será sentida nas outras estações em instantes diferentes no tempo, devido aos atrasos de propagação variados.

#### 2.1 Tamanho das Mensagens

Quadros são de tamanho fixo de 1.000 bytes, incluindo todos os overheads possíveis. Uma mensagem (informação do nível de rede sendo enviada para a interface de rede) pode ter a necessidade de ser quebrada em vários quadros para transmissão.

A pmf do número de quadros  $N$  em cada mensagem pode ser escolhida como geométrica ou determinística. Se  $0 < p < 1$ , a distribuição é geométrica, com  $P(N=k) = (1-p)^{k-1} p$ ,  $k > 0$ , e média  $1/p$ . Se  $p \geq 1$ , o número de quadros em cada mensagem é fixo e igual a  $p$ .

A distribuição do número de quadros por mensagem para a estação  $PC_i$  é dada pelo parâmetro  $p_i$ . Por exemplo, se a entrada for  $p_1=10$  e  $p_2=0,1$ , então a estação  $PC_1$  gerará mensagens de tamanho fixo com 10 quadros e a estação  $PC_2$  gerará mensagens com número de quadros geometricamente distribuído com média 10. Em ambos os casos o tamanho do quadro é sempre fixo e igual a 1.000 bytes.

## 2.2 Processo de Chegada de Mensagens

A taxa de chegada de mensagens em cada estação poderá ser determinística (D) ou exponencial (E). A escolha de uma ou outra opção deverá ser feita na entrada do programa, para cada uma das estações, de forma independente. O aluno escolherá a forma mais adequada de fornecer este dado. O intervalo médio entre chegadas, definido em segundos, será  $A_i$  para a estação  $PC_i$ .

No caso de estações operarem de forma determinística, vamos assumir uma sincronização inicial de tempo entre elas. Assim, se for dado  $A_1=A_2=A$ , determinístico para ambas as estações, isto significará que ambos os PCs receberão mensagens simultaneamente, a cada  $A$  segundos.

Se a taxa de chegada na estação  $PC_i$  for exponencial com intervalo médio  $A_i$  entre chegadas, então a taxa da exponencial será  $1/A_i$ .

## 2.3 Funcionamento da rede Ethernet

O Ethernet a ser simulado é um repetidor que opera com o protocolo CSMA/CD (carrier sense multiple access with collision detection), 1-persistente, observando um tempo mínimo de 9,6  $\mu$ s (equivalente a 96 bits transmitidos no meio) entre quadros. A detecção de atividade (meio ocupado) ou colisão ocorre na conexão de cada estação com o meio físico. Vamos assumir que os estados de meio ocupado e colisão detectada são acionados em tempo zero (atraso da lógica desprezível). O protocolo funciona da forma seguinte:

- a) Se a estação tem algum quadro para transmitir, ela sente o meio.
- b) Se no instante de tx o meio está livre, a estação checa se já passou 9,6  $\mu$ s (equivalente a tx de 96 bits) desde o final da última transmissão (final de uma transmissão com sucesso ou final de reforço de colisão), que pode ter sido gerada por ela própria, eventualmente. Se já passou, ela inicia a transmissão imediatamente.  
**Se o atraso ainda está pendente, a estação espera a contagem do atraso obrigatório e, então, transmite independentemente do estado do meio (ocupado ou não) naquele novo instante. O meio livre já havia sido sentido anteriormente, e por isso a estação, findo o atraso, sempre transmite.**
  - Se durante a transmissão do quadro ocorre colisão, a estação aborta a transmissão em andamento, transmite um sinal de reforço de colisão por 3,2  $\mu$ s (equivalente a tx de 32 bits) e, após este atraso, escolhe um instante aleatório para retransmissão, segundo o algoritmo Binary Backoff (veja abaixo), e repete o processo.
- c) Se no instante de tx o meio está ocupado, a estação persiste sentindo o meio até que o final de transmissão seja detectado (meio livre). **Então, após esperar um atraso de 9,6  $\mu$ s, a estação sempre transmite, independentemente do estado do meio (ocupado ou não) no início da transmissão.**
  - Se durante a transmissão do quadro ocorre colisão, a estação aborta a transmissão em andamento, transmite um sinal de reforço de colisão por 3,2  $\mu$ s (equivalente a tx de 32 bits) e, após este atraso, escolhe um instante aleatório para retransmissão, segundo o algoritmo Binary Backoff (veja abaixo), e repete o processo. Veja que o comportamento ao detectar colisão é o mesmo.

Se uma estação for a única a transmitir na rede, seus quadros são necessariamente transmitidos com um intervalo de  $9,6 \mu\text{s}$  entre eles.

O reforço de colisão pode ser visto como uma extensão do tempo de transmissão do quadro abortado. O final da última transmissão, então, poderá ser entendido como o final de transmissão de um quadro abortado ou o final de uma transmissão com sucesso.

Uma estação só processa um quadro por vez. Um quadro é transmitido com sucesso quando não há detecção de colisão durante sua transmissão.

### 2.3.1 Algoritmo Binary Backoff

O tempo é dividido em intervalos ou fatias (slots) de  $51,2 \mu\text{s}$ , equivalentes à transmissão de 512 bits a 10 Mbps.

Após  $k$  colisões, a estação escolhe aleatoriamente um atraso entre 0 e  $2^k - 1$  intervalos para retransmitir. Quando o número de colisões sobe além de 10, continua-se a usar  $k=10$ , ou seja, 1024 possibilidades de atraso para retransmissão. Se o número de colisões consecutivas chega a 16, o quadro é imediatamente descartado após a detecção da 16ª colisão.

A submissão do quadro descartado para uma nova transmissão é uma tarefa relegada a protocolos de nível superior, que não serão simulados neste trabalho.

Para dar um exemplo, imagine que o quadro na estação  $PC_1$  já sofreu 1 colisão. Quando este quadro começa sua segunda tentativa de transmissão, ele colide com a primeira tentativa de transmissão de um quadro da estação  $PC_2$ . Cada uma das estações ao detectar colisão gera o reforço de colisão e escolhe o próximo instante de retransmissão segundo o algoritmo de colisão.

A estação  $PC_1$ , com  $k=2$ , irá escolher aleatoriamente um atraso dentre as seguintes quatro possibilidades:

- $0 \mu\text{s}$  (tenta transmitir imediatamente), com probabilidade  $\frac{1}{4}$
- $51,2 \mu\text{s}$ , com probabilidade  $\frac{1}{4}$
- $102,4 \mu\text{s}$ , com probabilidade  $\frac{1}{4}$
- $153,6 \mu\text{s}$ , com probabilidade  $\frac{1}{4}$

A estação  $PC_2$ , por seu lado, só sofreu uma colisão, e irá escolher aleatoriamente um atraso dentre os seguintes:

- $0 \mu\text{s}$  (tenta transmitir imediatamente), com probabilidade  $\frac{1}{2}$
- $51,2 \mu\text{s}$ , com probabilidade  $\frac{1}{2}$

Caso alguma das estações decida transmitir imediatamente, ela deve observar se já se passaram  $9,6 \mu\text{s}$  de inatividade após a detecção de meio livre. Caso contrário, ela aguarda o término deste atraso e, então, transmite, independentemente do estado do meio neste instante.

Procure ter certeza de que entendeu o comportamento do Ethernet, pois este comportamento é de fato o que o padrão especifica. Estaremos testando o comportamento de um hub ou repetidor Ethernet.

## 3 Medidas de desempenho a serem coletadas

### 3.1 $T_{Ap}(i)$ = tempo de acesso de um quadro na estação $i$

*$T_{Ap}(i)$  é medido como o tempo transcorrido entre o primeiro instante em que o quadro é considerado para transmissão, até o início de sua transmissão com sucesso. Observe que, se um quadro é considerado para transmissão e o meio está ocupado, o tempo de acesso já está sendo contado. O tempo de acesso só será zero quando a estação tentar transmitir pela primeira vez e o meio estiver livre. Entretanto, se uma estação tem vários quadros para transmitir, o intervalo de  $9,6 \mu\text{s}$ , que tem que ser*

observado após a transmissão com sucesso do quadro anterior, não é contado no tempo de acesso do quadro atual, pois só após este atraso obrigatório é que a estação poderá considerar o quadro para transmissão. Quadros descartados não contribuem para o tempo de acesso médio dos quadros, mas somente os quadros transmitidos com sucesso. Estamos interessados em obter  $E[TA_p(i)]$ .

### 3.2 $TAm(i)$ = tempo de acesso de uma mensagem na estação $i$

$TAm(i)$  é o intervalo entre o instante em que o primeiro quadro da mensagem é considerado para transmissão até o instante de início da transmissão com sucesso do último quadro da mensagem. Se o último quadro de uma mensagem for descartado, o tempo de acesso dessa mensagem fica indeterminado e não deve ser considerado nas estatísticas. Caso um quadro intermediário seja descartado, o tempo de acesso da mensagem continuará a ser calculado de acordo com a definição, e, para efeitos deste trabalho, será computado no tempo de acesso médio das mensagens. Qualquer mensagem que tenha pelo menos um quadro descartado será considerada descartada. Estamos interessados em obter  $E[TAm(i)]$ .

### 3.3 $NCm(i)$ = número médio de colisões por quadro na estação $i$

$NCm(i)$  é obtido calculando o número de colisões dos quadros de uma mensagem dividido pelo número de quadros da mensagem. Como são transmitidas várias mensagens por estação, a média é obtida fazendo o cálculo para cada mensagem transmitida e tirando a média. Estamos interessados em obter  $E[NCm(i)]$ .

### 3.4 Utilização do Ethernet

A utilização do Ethernet é definida como sendo a relação entre o tempo que o meio está ocupado com alguma transmissão (quadro com sucesso, ou colisão ou reforço de colisão) e o tempo total de simulação, desprezando o tempo da fase transiente, naturalmente. **A utilização deve ser medida na estação 1 para referência.**

### 3.5 Vazão ( $i$ ) da estação $i$

**Vazão( $i$ )** é definida como sendo a relação entre o número de quadros transmitidos com sucesso na estação e o tempo de simulação, desconsiderando a fase transiente (quadros transmitidos na fase transiente não são computados e o tempo de simulação não inclui o tempo da fase transiente).

## 4 Especificação da Simulação

### 4.1 Estatísticas a Serem Coletadas

Deverão ser calculados  $E[TA_p(i)]$ ,  $E[TAm(i)]$ ,  $E[NCm(i)]$ , vazão( $i$ ) e utilização do Ethernet com intervalos de confiança de 95%. O intervalo de confiança não deverá ser superior a 10% do valor estimado, ou seja,  $U(\alpha) - L(\alpha) < 0,10 \mu$ .

### 4.2 Cenários

- 4.2.1 Cenário 1: estação 1 e estação 2 transmitem mensagens de 40 quadros de 1000 bytes, totalizando 4 Mbps por estação, sincronizadas (mensagens chegam nas duas estações ao mesmo tempo).

Parâmetros:  $A_1 = A_2 = 80 \text{ ms}$ , ambos determinísticos;  $p_1 = p_2 = 40$ ; estações 3 e 4 sem tráfego.

- 4.2.2 Cenário 2: estação 1 e estação 2 transmitem mensagens de 40 quadros de 1000 bytes, totalizando 4 Mbps por estação, chegada de mensagens Poisson.

Parâmetros:  $A_1 = A_2 = 80 \text{ ms}$  exponencial;  $p_1 = p_2 = 40$ ; estações 3 e 4 sem tráfego.

- 4.2.3 Cenário 3: estação 1 injetando tráfego determinístico (mensagens de 40 quadros) e estações 2, 3 e 4 gerando tráfego de mensagens com um único quadro na taxa de 0,5 Mbps por estação também determinístico – todas sincronizadas na geração)

Parâmetros:  $A_1 = 80 \text{ ms}$ , determinístico;  $p_1 = 40$ ;  $A_2=A_3=A_4 = 16 \text{ ms}$ , todos determinísticos;  $p_2 = p_3 = p_4 = 1$ .

- 4.2.4 Cenário 4: estação 1 injetando tráfego determinístico (mensagens de 40 quadros) e estações 2, 3 e 4 gerando tráfego de mensagens com um único quadro na taxa de 0,5 Mbps Poisson por estação.

Parâmetros:  $A_1 = 80 \text{ ms}$ , determinístico;  $p_1 = 40$ ;  $A_2=A_3=A_4 = 16 \text{ ms}$ , todos exponenciais;  $p_2 = p_3 = p_4 = 1$ .

## 5 Relatório

O relatório deverá conter necessariamente as seções abaixo, na seqüência apresentada. A introdução deve descrever com detalhes a filosofia de programação do simulador, com enumeração das estruturas principais e facilidades de programação utilizadas. Esta introdução deve propiciar uma visão completa do funcionamento do simulador, englobando o fluxo de controle e de dados. O relatório deverá ser entregue via e-mail.

### 5.1 Introdução (10%)

Descreva com detalhes a implementação do simulador, explicando:

- **a participação de cada membro nas etapas de construção do simulador, da geração dos resultados e elaboração da documentação. Esta informação deve constar da página de rosto do trabalho e é mandatória. Sem ela o trabalho não será avaliado;**
- o funcionamento geral do simulador;
- as estruturas internas utilizadas;
- como foi feita a coleta das estatísticas;
- como foi implementada a lista de eventos, quais os eventos escolhidos, e qualquer outra informação pertinente;
- a forma de geração das variáveis aleatórias envolvidas (facilidades de geração de número aleatório da linguagem utilizada);
- o tipo de linguagem utilizada; o aluno poderá usar qualquer linguagem para a programação do simulador. Os resultados gráficos com os intervalos de confiança poderão ser impressos utilizando qualquer pacote disponível.
- o método para obtenção dos intervalos de confiança (replicação, batch ou regenerativo), informando sobre a escolha das sementes e outras providências;
- a escolha dos parâmetros utilizados (número de fregueses coletados por rodada, número de rodadas, etc) para cada cenário (resumo do que será apresentado como resultado)
- a máquina que usou para rodar a simulação e tempo gasto na execução completa da simulação;
- outras informações pertinentes.

### **5.2    *Teste de Correção (10%)***

Nesta seção você descreverá os testes de correção que foram efetuados para garantir o funcionamento perfeito do simulador. Descreva estes procedimentos de teste. Algumas dicas estão na apostila.

Os testes de correção são importantíssimos, pois determinam a confiança que você terá nos resultados. De uma maneira geral, procure ter plena certeza de que o programa está executando o que você quer simular. Inserção de eventos na fila de controle pode ser fonte de erro. Tenha certeza de que a geração das amostras está correta.

Para algumas situações é possível ter o resultado analítico para a simulação. Você poderá usar este resultado e mostrar o pleno funcionamento do simulador. Por exemplo, imagine a situação em que só uma estação transmite. Faça a estação gerar tráfego determinístico e comprove os valores obtidos com o esperado. Faça outros testes para comprovar que a coleta de estatística está funcionando corretamente.

### **5.3    *Estimativa da fase transiente (10%)***

Nesta seção você descreverá como a fase transiente foi estimada para os diversos valores de  $\rho$ . Se você usou uma heurística, apresente resultados quantitativos que justificam sua escolha. Se você usou algum resultado analítico, mostre como seu estado inicial do sistema para os diferentes valores de utilização foi determinado. Lembre-se que a fase transiente poderá ser mais crítica para valores mais altos de utilização.

### **5.4    *Listagem documentada do programa e executável (10%)***

Independentemente da introdução, o corpo do programa deve ser extensamente documentado, já que nosso intuito principal é exercitar um trabalho de construção de simulador com garantia de exatidão e facilidade de compreensão da programação. No momento não estamos interessados no aspecto eficiência, que poderia vir após uma fase de otimização. Mas estamos interessados em obter um índice (número de rodadas versus total de fregueses (fase transiente incluída) o mais baixo possível.

A documentação do fonte deverá ser feita com rigor, explicando cada subrotina ou passo da programação. Os comentários são fundamentais e devem vir inseridos no texto do programa. Lembre-se que terei que ler todos os programas dos alunos e um programa mal documentado receberá avaliação baixa.

**O executável para PC só deverá ser entregue se solicitado.**

### **5.5    *Tabelas com os resultados e comentários pertinentes (50%)***

Não deixe de indicar todos os parâmetros usados na obtenção dos resultados apresentados, comentar estes resultados e dar sua interpretação para eles.

Relacione os índices alcançados e aponte as condições em que obteve o melhor índice.

Comente os resultados obtidos, comparando-os entre si. Observe as situações desfavoráveis criadas pelo sincronismo de chegada e pela operação do algoritmo de retransmissão.

Se você achar interessante rodar o simulador em outras condições de tráfego para evidenciar alguma suspeita de comportamento das estações, este resultado complementar pode ser apresentado. Entretanto, não deixe de executar e comentar as simulações solicitadas.

### **5.6    *Conclusões (10%)***

Coloque aqui seus comentários finais. Descreva dificuldades encontradas, otimizações feitas, e outras conclusões que você tirou do trabalho. Comente o que poderia ser melhor, tempo de execução do seu programa, e outros comentários que você julgar relevante.

Cada uma das seções terá seu peso na avaliação. Portanto, não deixe que falte nenhuma seção no seu relatório.

## **IMPORTANTE**

---

**A data limite para a entrega do relatório será 15/07/2010, quinta-feira.**

**O trabalho deverá ser feito em grupo, preferencialmente. Cada grupo poderá ter no máximo 5 pessoas e sua composição deverá ser informada até quinta-feira, 13/05/2010, via e-mail. Alunos que estiverem sem grupo terão que fazer individualmente o trabalho, o que não é recomendado.**

**Todo trabalho deve ter uma única lista de eventos. Pode ser usada qualquer linguagem, mas o trabalho deve gerar um código objeto capaz de rodar em Windows ou Linux.**

**Na entrega, os simuladores serão rodados com diferentes valores de entrada para teste e aferição. Os códigos fontes devem ser próprios de cada grupo e o grupo deve participar de todas as etapas do processo de construção, documentação e exercício do simulador para os diferentes cenários solicitados.**

**ALTERAÇÕES DE GRUPO A POSTERIORI SÓ COM AUTORIZAÇÃO DO PROFESSOR.**

---