

TRABALHO DE SIMULAÇÃO

Alunos:

Bruno Bottino Ferreira
Denilson Figueiredo de Sá
Pedro Mendonça Pinto Rocha

Todos os componentes do grupo participaram ativamente da definição do funcionamento do simulador, da implementação do código e da realização do relatório, através de reuniões periódicas, tanto presenciais quanto virtuais.

Sumário

Introdução.....	1
Funcionamento geral do simulador.....	1
Especificações de desenvolvimento e execução.....	1
Métodos de geração de estatísticas e variáveis aleatórias.....	2
Tipos de Eventos e Fila de Eventos.....	2
Estruturas internas utilizadas.....	4
Escolha dos parâmetros.....	4
Especificação da máquina.....	5
Teste de correção	6
Uma máquina a 8Mbps.....	6
Uma máquina a 4Mbps.....	6
Duas máquinas, backoff desligado.....	7
Duas máquinas, sem colisão.....	7
Estimativa da fase transiente.....	8
Código fonte documentado.....	9
Tabelas com os resultados e comentários pertinentes	27
Cenário 1.....	27
Cenário 2.....	29
Cenário 3.....	34
Cenário 4.....	38
Comentários.....	41
Conclusões.....	42

Introdução

Funcionamento geral do simulador

O fluxo de execução do simulador é, em linhas gerais, descrito a seguir. A execução de cada evento é descrita com detalhes na seção que trata dos tipos de eventos.

- As estruturas internas são inicializadas e os eventos iniciais gerados
- O loop principal de rodadas é iniciado. Para cada rodada:
 - Se o critério de parada é satisfeito (todos os ICs pequenos o suficiente), pare
 - Reinicie a coleta de amostras da rodada em cada máquina
 - Retire da fila de eventos e processe “n = eventos por rodada” eventos
 - Se não é a fase transiente, calcule as médias da rodada para cada máquina e adicione às listas de médias globais. Além disso, exiba as médias da rodada na tela
- Os gráficos da evolução das médias são mostrados na tela.

Especificações de desenvolvimento e execução

Foi utilizada a linguagem Python (versão 2.6) para desenvolvimento do simulador, devido à sua alta portabilidade e facilidade de utilização para projetos rápidos, e à disponibilidade de pacotes estatísticos e de plotagem gráfica. Parte do desenvolvimento foi feita em Windows e parte em Linux, e o simulador funciona sem problemas nestes dois ambientes, desde que estejam instaladas as bibliotecas externas necessárias. São elas:

- matplotlib, para *plotagem* dos gráficos;
- numpy, para geração de amostras geométricas;
- scipy, para obtenção dos percentis da distribuição t-student.

Além disso, foram utilizadas os módulos heapq (fila de prioridades), math, random, time e pickle, todos presentes na bibliotecas padrão do Python.

O arquivo simulador.py contém o código da classe Simulador (que é a entidade principal do programa) e estruturas e classes auxiliares. Os arquivos rodar_cenario.py e rodar_cenario_teste.py instanciam o simulador com os parâmetros correspondentes aos cenários pedidos na especificação e alguns cenários utilizados no teste de corretude, respectivamente.

Pode-se editar estes arquivos para incluir mais cenários, definindo o novo cenário de forma análoga aos existentes e adicionando à lista “cenarios”. As funções globais

Exponencial(intervalo_medio) e Deterministica(intervalo) estão disponíveis para definição dos intervalos entre chegadas do cenário. O valor de num_quadros é interpretado como na especificação, ou seja, se ≥ 1 , define uma quantidade determinística de quadros por pensagem; se < 1 , define uma quantidade determinada por uma variável aleatória geométrica cujo parâmetro é igual ao valor passado. Além disso, também pode-se utilizar as funções Deterministica(num_quadros) e Geometrica(p) diretamente para a definição deste parâmetro.

Para executar o simulador em um dado cenário basta chamar este arquivo passando como parâmetro o número do cenário. Por exemplo, o comando

```
python rodar_cenario.py 2
```

irá executar o simulador com os parâmetros relativos ao cenário 2 da especificação.

Métodos de geração de estatísticas e variáveis aleatórias

Foi utilizado o método Batch para geração das rodadas, principalmente pela necessidade de se realizar apenas uma fase transiente, diminuindo o tempo necessário para a execução do programa.

A linguagem Python juntamente com as bibliotecas utilizadas forneceram as funções necessárias para gerar valores de variáveis aleatórias e distribuições:

- Exponencial: na função `random.expovariate(lambda)`
- Geométrica: na função `numpy.random.geometric(p)`
- t-Student: na função `scipy.stats.t.ppf(percent, graus_de_liberdade)`

Para calcular as estatísticas, foi criada uma classe Estatistica que coleta amostras e retorna, a qualquer momento, a média, a variância e o intervalo de confiança das amostras coletadas até então. Além disso, ela permite que a cada amostra adicionada seja associado um “intervalo” para ser desenhado no gráfico junto com a amostra. Sendo assim, esta classe também possui um método responsável por gerar um gráfico com as amostras e respectivos intervalos coletados. Esta classe foi utilizada da seguinte maneira: para cada rodada, são coletadas amostras relativas às variáveis requeridas. Ao fim da rodada, é calculada a média destas amostras. Esta média é adicionada a uma lista global de médias. A média das médias globais (que é o valor final procurado) é adicionada numa outra lista de valores, juntamente com o Intervalo de Confiança atual da média global, de forma a mostrar nos gráficos a evolução de cada média global e seu respectivo IC.

Tipos de Eventos e Fila de Eventos

Foi criada uma classe abstrata Evento da qual todos os outros tipos de eventos herdam. Esta classe apenas possui o protótipo do método “processar” que é chamado para que

seja feito o processamento de um determinado evento, funcionando assim como uma interface (em Python não existe o conceito direto de interface). Os métodos relativos a operações com cada máquina (classe Host) são descritos com detalhes na seção que trata das estruturas internas do simulador. Os seguintes tipos de evento foram utilizados no simulador:

- ChegouMensagem: representa a chegada de uma mensagem da camada superior, formada por uma quantidade arbitrária de quadros. A mensagem é inserida em uma fila de mensagens a serem enviadas por aquela máquina, juntamente com sua respectiva quantidade de quadros. Se a fila estava vazia quando a mensagem chegou, é feita uma tentativa de agendamento de envio (método Host.tentar_enviar). A chamada a Host.tentar_enviar para o caso de fila não vazia será feita no tratamento dos eventos FimDeRecebimento e FimDeEnvio. Além disso, este evento insere na fila de eventos a próxima chegada de mensagem, de acordo com o intervalo entre chegadas definido no cenário.

- InicioDeEnvio: representa o momento em que uma estação começa a transmitir um quadro. A quantidade de quadros da mensagem à frente da fila é decrementada, mas ela ainda não é removida da fila pois o quadro atual pode necessitar ser reenviado. O estado da máquina relativa ao evento é atualizado com a quantidade de tentativas de transmissão do quadro atual e são gerados os eventos FimDeEnvio para este quadro relativo à máquina de origem, e InicioDeRecebimento no hub. Além disso chama Host.checar_jam para verificar ocorrência de colisão.

- FimDeEnvio: representa o momento em que um quadro ou reforço de jam terminou de ser enviado. Este evento pode ter uma flag “cancelado” ligada caso ele seja relativo ao fim de envio de um quadro abortado. Neste caso, o processamento do evento é ignorado. Senão, o método Host.tentar_enviar é chamado para realizar uma tentativa de agendamento de envio, já que este é um momento em que o meio possivelmente se torna livre. Caso este seja um término de envio de quadro e não reforço de jam, são coletadas as estatísticas de TAp e TAm (esta última apenas se for o último quadro da mensagem). Além disso, a quantidade de tentativas de transmissão da máquina é reiniciada para zero e a fila de mensagens a serem enviadas é atualizada (Host.andar_fila).

- InicioDeRecebimento: se a máquina relativa ao evento for o hub, apenas é gerado o InicioDeRecebimento para todas as máquinas. Caso contrário, um contador de uso do meio de recebimento para a máquina é incrementado (se a transmissão não for originária na própria máquina) e é verificado se aconteceu uma colisão (Host.checar_jam).

- FimDeRecebimento: se a máquina relativa ao evento for o hub, apenas é gerado o FimDeRecebimento para todas as máquinas. Caso contrário, um contador de uso do meio de recebimento para a máquina é decrementado (se a transmissão não for originária na própria máquina) e o instante de começo de tempo ocioso é marcado caso o meio tenha começado a ficar vazio. Além disso, o método Host.tentar_enviar é chamado para realizar uma tentativa de agendamento de envio, já que este é um momento em que o meio possivelmente se torna livre.

A fila de eventos no simulador foi implementada como uma fila de prioridades ordenada pelo tempo do evento utilizando a estrutura heapq da linguagem.

Estruturas internas utilizadas

A estrutura básica é a classe Simulador. Ela contém os parâmetros da simulação, assim como algumas variáveis de estado globais da simulação (como o tempo atual) e a coleta das estatísticas globais. Contém também os objetos que representam as máquinas.

A classe Host representa uma estação (máquina) na simulação. Ela contém campos que representam o estado atual da máquina, tais como:

- Tentativas de transmissão do quadro atual
- Contador de uso do meio de recepção
- Flags indicando se a máquina está enviando e se há uma transmissão agendada
- Valores utilizados para contabilizar estatísticas

Relativo ao processamento de eventos (e chamados por estes) esta classe possui também os seguintes métodos:

- `tentar_enviar`: tenta agendar uma transmissão. Caso o meio esteja livre, é calculado o tempo de backoff exponencial (se aplicável). Um evento de `InicioDeEnvio` é então agendado para o máximo entre (tempo atual + tempo do backoff) e (tempo de início de ociosidade + 9.6us). Além disso são calculados alguns valores para posterior utilização para obtenção de estatísticas.
- `andar_fila`: incrementa o próximo quadro a ser enviado na mensagem na frente da fila atualmente. Caso seja o último quadro da mensagem, retira a mensagem da fila, coleta estatísticas para o NCm e reinicia um contador interno de colisões da mensagem atual.
- `checam_jam`: verifica se está havendo colisão no momento. Caso esteja, incrementa um contador interno de colisões, cancela o evento de `FimDeEnvio` do quadro atual, agenda o evento de `FimDeEnvio` do reforço de colisão, e descarta o quadro atual caso já tenha havido 16 tentativas de transmissão.

Escolha dos parâmetros

O tamanho de cada rodada foi definido empiricamente, de forma a obter uma boa qualidade das amostras das rodadas para que a variância se mantivesse baixa e os Intervalos de Confiança diminuíssem com o tempo. O tamanho final definido para as rodadas de todos os cenários foi de 500 000 eventos.

O número de rodadas é determinado ao longo da simulação, de acordo com o critério de parada pedido na especificação, ou seja, o tamanho do Intervalo de Confiança de todas as estatísticas deve ser inferior a 10% do respectivo valor.

Para cálculo do Intervalo de Confiança, utilizamos que:

$$P(\mu - t\text{-student}(1 - \alpha/2) * \sigma / \text{raiz}(n) \leq \mu \leq \mu + t\text{-student}(1 - \alpha/2) * \sigma / \text{raiz}(n)) = 1 - \alpha$$

e

$$1 - \alpha = 0,95 \text{ (dado do problema)}$$

Desenvolvendo:

$$\alpha = 0,5$$

$$t\text{-student}(1 - \alpha/2) = t\text{-student}(1 - 0,025) = t\text{-student}(0,975)$$

Temos que n é o número de rodadas, e σ o desvio padrão entre os valores.

Como utilizamos um método do pacote `scipy` para obtenção dos percentis da `t-Student` com n graus de liberdade, não foi necessário definir um número mínimo de rodadas para utilização do valor assintótico da distribuição.

Especificação da máquina

A máquina utilizada para rodar o simulador possui a seguinte configuração:

Core 2 Duo T5550 (1833 MHz) com 3GB de RAM

Gentoo Linux 64-bit

Python 2.6.5

NumPy 1.4.1

SciPy 0.7.2

matplotlib 0.99.1.1

Os tempos totais de simulação foram:

- Cenário 1: 4 rodadas, 313 segundos simulados em 135,63 segundos reais
- Cenário 2: 24 rodadas, 1652 segundos simulados em 616,15 segundos reais
- Cenário 3: 9 rodadas, 262 segundos simulados em 134,52 segundos reais
- Cenário 4: 6 rodadas, 321 segundos simulados em 153,22 segundos reais

Teste de correção

Para verificar se o programa foi implementado corretamente, foram realizados testes controlados, de forma que os resultados pudessem ser verificados.

Os seguintes cenários foram elaborados:

Uma máquina a 8Mbps

Uma máquina enviando sozinha para a rede.

Parâmetros:

Tempo de chegada de mensagens determinístico de 40 ms entre cada mensagem.

Número de quadros de cada mensagem: 40

Tamanho de 1 quadro: 1000 bytes

Esperado:

Obter uma utilização média da rede de 8Mbps ($40 * 8000 / 40000$).

Obter um tempo de acesso de uma mensagem de $(800 + 9,6) * 39 = 31574,4$.

Resultado:

A média de utilização obtida foi de $0,800090 * 10\text{Mbps} = 8,00090\text{Mbps}$ e o tempo de acesso de uma mensagem foi exatamente 31574,4.

Uma máquina a 4Mbps

Uma máquina enviando sozinha para rede com tempo de chegada de mensagens sendo o dobro do teste 1, e os outros parâmetros iguais.

Esperado:

Obter uma utilização da rede de metade do obtido no teste anterior, 4Mbps, e esperava-se que o tempo de acesso de cada mensagem fosse o mesmo.

Resultado:

A média de utilização obtida foi de $0,400146 * 10\text{Mbps} = 4,00146\text{Mbps}$ e o tempo de acesso de uma mensagem foi de exatamente 31574,4.

Duas máquinas, backoff desligado

Duas máquinas enviando para a rede, mas com algoritmo de backoff desligado, ou seja, quando uma colisão é detectada, uma nova tentativa de envio é feita imediatamente.

Parâmetros:

Tempo de chegada de mensagens determinístico de 200ms por mensagem.

Número de quadros por mensagem: 1

Tamanho do quadro: 1000 bytes

Como queríamos garantir que houvesse colisões, os mesmos parâmetros foram utilizados para as duas máquinas.

Esperado:

Com o algoritmo de backoff desligado, o esperado é que nenhuma máquina conseguisse enviar nenhuma mensagem, e suas estatísticas individuais ficassem zeradas, e a média de utilização da rede é de, aproximadamente

Resultado:

Todas as estatísticas individuais de cada máquina estavam zeradas, indicando que nenhuma delas conseguiu enviar nenhuma mensagem.

Duas máquinas, sem colisão

Duas máquinas enviando para a rede, sem que haja controle de colisão. Neste cenário não estamos interessados em saber se as informações trafegaram corretamente, apenas estamos fazendo uma análise do simulador.

Parâmetros:

Tempo de chegada de mensagens determinístico de 80ms por mensagem.

Número de quadros por mensagem: 40

Tamanho do quadro: 1000 bytes

As mesmas configurações foram utilizadas nas duas máquinas.

Esperado:

Sem controle de colisão, o esperado é que a rede se comporte como se houvesse somente 1 máquina.

Resultado:

Os resultados obtidos foram todos similares aos resultados obtidos no teste 2.

Conclusão dos testes de correção:

Após analisar os resultados de todos os testes efetuados, pode-se observar que foram todos condizentes com os resultados esperados, o que garante que o simulador está funcionando corretamente.

Estimativa da fase transiente

Para estimar a fase transiente, analisamos o comportamento da estatística *Utilização do Ethernet* ao longo da simulação. Com o auxílio de gráficos, pudemos descobrir quantos eventos são necessários para a estabilização dessa estatística. Chegamos aos seguintes tamanhos para a fase transiente de cada cenário:

Cenário 1: 2 500 000 eventos

Cenário 2: 10 000 000 eventos

Cenário 3: 250 000 eventos

Cenário 4: 2 500 000 eventos

Os gráficos usados para estimar a fase transiente, assim como a indicação do tamanho da fase transiente, podem ser visto mais à frente, nas páginas que contêm os resultados de cada cenário.

simulador.py

```
1  # -*- coding: utf-8 -*-
2  # vi:ts=4 sw=4 et
3
4  import heapq
5  import math
6  import matplotlib.pyplot as pyplot
7  import numpy
8  import random
9  import scipy.stats
10 import time
11
12
13 #####
14 # Funções úteis
15
16 def debug_print(string):
17     """Função de impressão, para debug"""
18     #print string
19     pass
20
21 def exibir_legenda():
22     """Exibe a legenda no gráfico"""
23     l = pyplot.legend(loc="best", fancybox=True, shadow=True)
24
25     # Fundo cinza
26     l.get_frame().set_facecolor("0.90")
27
28     # Tamanho do texto
29     for t in l.get_texts():
30         t.set_fontsize("x-small")
31
32 #####
33 # Estruturas auxiliares: Heap de eventos e coleta de estatísticas
34
35 class HeapDeEventos(list):
36     """Implementação de uma fila com prioridades."""
37
38     def adicionar(self, tempo, evento):
39         heapq.heappush(self, (tempo, evento))
40
41     def remover(self):
42         """Retorna uma tupla (tempo, evento)"""
43         return heapq.heappop(self)
44
45 class Estatisticas(object):
46     """Coletor de amostras para geração e plotagem de estatísticas"""
47
48     def __init__(self, titulo=u"Estatísticas"):
49         self.amostras = []
50         self.intervalos = []
51         self.soma_amostras = 0
52         self.soma_quadrados = 0
53         self.num_amostras = 0
54         self.titulo = titulo
55
56     def adicionar_amostra(self, amostra):
57         self.amostras.append(amostra)
58         self.soma_amostras += amostra
59         self.soma_quadrados += amostra*amostra
60         self.num_amostras += 1
61
62     def adicionar_intervalo(self, intervalo):
63         self.intervalos.append(intervalo)
64
65     def plot(self, *args, **kwargs):
66         """Desenha uma linha correspondente às amostras coletadas. Caso
67         os intervalos de confiança tenham sido calculados, desenha
68         também o tamanho dos intervalos."""
69
70         if len(self.amostras) == 0:
71             return
72
73         x = numpy.arange(1, self.num_amostras+1)
74         if len(self.amostras) != len(self.intervalos):
75             plot = pyplot.plot(x, self.amostras, *args, **kwargs)
76         else:
77             pyplot.errorbar(x, self.amostras, yerr=self.intervalos, *args, **kwa
78 rgs)
79
80         #pyplot.title(titulo)
81         #pyplot.show()
82
83     def media(self):
```

simulador.py

```
85         """Retorna a média das amostras"""
86         if self.num_amostras == 0:
87             return 0
88
89         return self.soma_amostras / self.num_amostras
90
91     def variancia(self):
92         """Retorna a variância das amostras"""
93         if self.num_amostras < 2:
94             return 0
95
96         #ref: http://en.wikipedia.org/wiki/Variance#Population_variance_and_samp
le_variance
97         sq = self.soma_quadrados
98         sa = self.soma_amostras
99         n = self.num_amostras
100
101         # Usando abs() para evitar resultado -0.0000 devido a erros de ponto flu
tuante
102         return abs( (sq / (n - 1)) - ((sa * sa) / (n * (n - 1))) )
103
104     def intervalo_de_confianca(self):
105         """Retorna metade do tamanho do intervalo de confiança, ou seja,
106         o intervalo será definido pela média +- o valor retornado por
107         esta função"""
108
109         if self.num_amostras < 2:
110             return 0
111
112         # 1 - 95% = 5%
113         # 5% / 2 = 2.5%
114         # 1 - 2.5% = 97.5% = 0.975
115         t_student_95 = scipy.stats.t.ppf(0.975, self.num_amostras-1)
116
117         return t_student_95 * math.sqrt(self.variancia() / self.num_amostras)
118
119     def precisao_suficiente(self):
120         """Retorna True ou False, indicando se a largura do intervalo de
121         confiança é menor que 10% da média das amostras."""
122
123         if self.num_amostras < 2:
124             return False
125
126         return 2 * self.intervalo_de_confianca() <= self.media() / 10.0
127
128 #####
129 # As funções a seguir são usadas na definição dos parâmetros de cada
130 # Host, na inicialização do simulador.
131 #####
132
133     def Exponencial(intervalo):
134         """Gerador de tempo exponencial com média igual a 'intervalo'."""
135         return lambda: random.expovariate(1.0/intervalo)
136
137     def Deterministica(valor):
138         """Gerador determinístico de tempos ou número de quadros."""
139         return lambda: valor
140
141     def Geometrica(probabilidade):
142         """Gerador de distribuição geométrica."""
143         return lambda: numpy.random.geometric(probabilidade)
144
145 #####
146 # Coisas que encapsulam a interface do Host
147 #####
148
149     class Mensagem(object):
150         """Representa uma mensagem a ser enviada."""
151
152         def __init__(self, rodada, num_quadros):
153             self.rodada = rodada
154             self.num_quadros = num_quadros
155
156     class Host(object):
157         """Representa uma máquina."""
158
159         def __init__(self, hostname, distancia, chegada, num_quadros):
160             """Recebe os parâmetros do host:
161             hostname = Nome da máquina (ignorado pelo simulador)
162             dist = Distância deste host ao hub (medida em metros)
163             chegada = Processo de chegada das mensagens a ser enviadas
164             num_quadros = Distribuição do número de quadros para cada mensagem"""
165
166             self.hostname = hostname
```

simulador.py

```
168         self.distancia = distancia
169         self.chegada = chegada
170         # Estes dois campos serão definidos mais abaixo
171         #self.num_quadros = num_quadros
172         #self.ativo = (chegada and num_quadros)
173
174         if callable(num_quadros) or num_quadros is None:
175             # Se o tipo de distribuição foi definido explicitamente:
176             self.num_quadros = num_quadros
177         else:
178             # Caso contrário, auto-detectar o tipo de distribuição através
179             # do número passado
180             if num_quadros < 1:
181                 # O número passado é uma probabilidade
182                 self.num_quadros = Geometrica(num_quadros)
183             else:
184                 # O número passado é uma quantidade constante
185                 # (determinística) de quadros.
186                 # Cuidado! Não é tratado o caso do número de quadros ser
187                 # fracionário!
188                 self.num_quadros = Deterministica(num_quadros)
189
190         # Um host é considerado ativo se ele gera tráfego. Ou seja, se
191         # ele tem uma distribuição de chegada de mensagens e de número
192         # de quadros de cada mensagem.
193         self.ativo = callable(self.chegada) and callable(self.num_quadros)
194
195     def reset(self):
196         """Faz um "reset" no host, reiniciando todas as estatísticas e o
197         estado do host. Este método deve ser chamado antes de começar a
198         simulação."""
199
200         # Fila de mensagens
201         self.fila = []
202         # Número do próximo quadro (dentro da mensagem atual) a ser enviado
203         self.proximo_quadro = 0
204
205         self.tentativas_de_transmissao = 0
206         self.tempo_considerar_envio_quadro = -1
207         self.tempo_comeco_envio_quadro = -1
208         self.tempo_comeco_envio_mensagem = -1
209         self.uso_do_meio = 0
210         self.tempo_comeco_ocioso = -100000.0
211
212         # Flag para indicar se o host está enviando algum quadro
213         self.enviando = False
214         # Flag para indicar se o host está esperando o tempo do binary backoff
215         self.agendado = False
216         self.contador_colisoes = 0
217
218         self.tap_global = Estatisticas()
219         self.tam_global = Estatisticas()
220         self.ncm_global = Estatisticas()
221         self.vazao_global = Estatisticas()
222
223         self.tap_global_media = Estatisticas()
224         self.tam_global_media = Estatisticas()
225         self.ncm_global_media = Estatisticas()
226         self.vazao_global_media = Estatisticas()
227
228         self.reiniciar_estatisticas()
229
230     def reiniciar_estatisticas(self):
231         """Reinicia as estatísticas no início de uma rodada."""
232
233         self.tap_rodada = Estatisticas()
234         self.tam_rodada = Estatisticas()
235         self.ncm_rodada = Estatisticas()
236
237         self.quadros_com_sucesso = 0
238
239     def precisao_suficiente(self):
240         """Indica se as estatísticas coletadas neste host já possuem a
241         precisão desejada."""
242
243         return (
244             self.tap_global.precisao_suficiente() and
245             self.tam_global.precisao_suficiente() and
246             self.ncm_global.precisao_suficiente() and
247             self.vazao_global.precisao_suficiente()
248         )
249
250     def finalizar_rodada(self, tempo_rodada):
251         """Salva as estatísticas da rodada na estatística global."""
252
```

simulador.py

```
253         self.tap_global.adicionar_amostra(self.tap_rodada.media())
254         self.tap_global_media.adicionar_amostra(self.tap_global.media())
255         self.tap_global_media.adicionar_intervalo(self.tap_global.intervalo_de_c
onfianca())
256
257         self.tam_global.adicionar_amostra(self.tam_rodada.media())
258         self.tam_global_media.adicionar_amostra(self.tam_global.media())
259         self.tam_global_media.adicionar_intervalo(self.tam_global.intervalo_de_c
onfianca())
260
261         self.ncm_global.adicionar_amostra(self.ncm_rodada.media())
262         self.ncm_global_media.adicionar_amostra(self.ncm_global.media())
263         self.ncm_global_media.adicionar_intervalo(self.ncm_global.intervalo_de_c
onfianca())
264
265         self.vazao_global.adicionar_amostra(1000000.0 * self.quadros_com_sucesso
/ tempo_rodada)
266         self.vazao_global_media.adicionar_amostra(self.vazao_global.media())
267         self.vazao_global_media.adicionar_intervalo(self.vazao_global.intervalo_
de_confianca())
268
269     def tentar_enviar(self, simulador):
270         debug_print("tentar_enviar maquina=%s tentativas=%d tco=%f" % (self.host
name, self.tentativas_de_transmissao, self.tempo_comeco_ocioso))
271
272         if len(self.fila) == 0: return # Nada a transmitir
273         if self.enviando: return
274         if self.agendado: return
275
276         if self.uso_do_meio == 0: # Meio livre
277             # Binary backoff
278             k = self.tentativas_de_transmissao
279             if k > 10: k = 10
280             tempo_atraso = random.randint(0, (2 ** k) - 1) * simulador.tempo_fat
ia_backoff
281
282             if simulador.ignorar_backoff:
283                 tempo_atraso = 0
284
285             tempo_envio = max(simulador.tempo_agora + tempo_atraso, self.tempo_c
omeco_ocioso + simulador.tempo_minimo_ocioso)
286             if self.tentativas_de_transmissao == 0:
287                 # Setar tempo de acesso na primeira tentativa somente
288                 self.tempo_considerar_envio_quadro = tempo_envio
289
290             if self.proximo_quadro == 0:
291                 self.tempo_considerar_envio_mensagem = tempo_envio
292
293             debug_print("                agendei para t=%f" % (tempo_envio))
294
295             simulador.eventos.adicionar(
296                 tempo_envio,
297                 InicioDeEnvio(self)
298             )
299             self.agendado = True
300         else:
301             if self.tentativas_de_transmissao == 0:
302                 #setar tempo de acesso na primeira tentativa somente
303                 self.tempo_considerar_envio_quadro = simulador.tempo_agora
304
305                 #evento de ComecoDeEnvio será tratado no FimDeRecebimento
306
307     def andar_fila(self, simulador):
308         self.proximo_quadro += 1
309
310         # Se é o último quadro desta mensagem
311         if self.proximo_quadro == self.fila[0].num_quadros:
312             #estatísticas
313             if self.fila[0].rodada == simulador.rodada_atual:
314                 self.ncm_rodada.adicionar_amostra(1.0 * self.contador_colisoes /
self.fila[0].num_quadros)
315
316                 self.contador_colisoes = 0
317
318                 #mensagem enviada por completo; retira da fila
319                 self.proximo_quadro = 0
320                 self.fila.pop(0)
321
322     def checar_jam(self, simulador):
323         if not simulador.ignorar_colisao and self.uso_do_meio != 0 and self.envi
ando and not self.fim_de_envio.cancelado: #colisao
324             debug_print("    *** COLISAO DETECTADA ***")
325
326             self.contador_colisoes += 1
327
```

simulador.py

```
328         #cancelar FimDeEnvio do quadro
329         self.fim_de_envio.cancelado = True
330
331         #agendar FimDeEnvio do Jam
332         simulador.eventos.adicionar(
333             simulador.tempo_agora + simulador.tempo_reforco_jam,
334             FimDeEnvio(self.fim_de_envio.rodada, maquina = self, sou_jam = T
rue)
335     )
336
337     #verificar quantidade de tentativas de transmissao
338     if self.tentativas_de_transmissao == 16:
339         #descartar quadro
340         self.andar_fila(simulador)
341         #simulador.perdidos += 1
342         #simulador.totais += 1
343
344     #tentará enviar o quadro novamente no FimDeEnvio do jam
345
346     def __getstate__(self):
347         """Este método é chamado pelo módulo pickle. Este método remove
348         as closures e funções lambda, permitindo que o pickle consiga
349         salvar este objeto.
350
351         Pickle é um módulo que permite salvar um objeto em um arquivo e
352         carregá-lo novamente mais tarde. Neste simulador, é útil para
353         poder estudar os dados coletados sem precisar reiniciar a
354         simulação.
355
356         Note que não será possível recomeçar a simulação a partir de um
357         host carregado através do pickle."""
358
359         d = self.__dict__.copy()
360         d["chegada"] = None
361         d["num_quadros"] = None
362         return d
363
364
365     class Hub(object):
366         """Representa um Hub, só serve pra ter um nome e ajudar no Debug."""
367
368         def __init__(self):
369             self.hostname = "Hub"
370
371
372     # valor especial que representa o hub no campo "maquina" dos eventos
373     HUB = Hub()
374
375
376     #####
377     # Eventos
378
379     class Evento(object):
380         """Classe abstrata que representa um evento."""
381
382         def processar(self, simulador):
383             raise NotImplementedError()
384
385
386     class ChegouMensagem(Evento):
387         """Representa a chegada de uma mensagem da camada superior."""
388
389         def __init__(self, rodada, maquina):
390             self.rodada = rodada
391             self.maquina = maquina
392             self.num_quadros = maquina.num_quadros()
393
394         def processar(self, simulador):
395             debug_print("- Evento: ChegouMensagem com %d quadros em t=%f na maquina=%s" % (
396                 self.num_quadros, simulador.tempo_agora, self.maquina.hostname ))
397
398             # Gera o próximo evento
399             simulador.eventos.adicionar(
400                 simulador.tempo_agora + self.maquina.chegada(),
401                 ChegouMensagem(simulador.rodada_atual, self.maquina)
402             )
403
404             fila_vazia = (len(self.maquina.fila) == 0)
405
406             # Adiciona mensagem à fila de envio
407             self.maquina.fila.append(
408                 Mensagem(self.rodada, self.num_quadros)
409             )
410
```

simulador.py

```
411         # A fila estava vazia quando esta mensagem chegou?
412         if fila_vazia:
413             self.maquina.tentar_enviar(simulador)
414
415     class InicioDeEnvio(Evento):
416         """Representa o momento em que uma estação começa a transmitir um
417         quadro."""
418
419         def __init__(self, maquina):
420             self.maquina = maquina
421
422         def processar(self, simulador):
423             debug_print("- Evento: InicioDeEnvio em t=%f maquina=%s quadro=%d" % (
424                 simulador.tempo_agora, self.maquina.hostname, self.maquina.proximo_q
425                 uadro ))
426
427             #verifica mensagem na fila de envio
428             #(ela só será removida de fato no FimDeEnvio com sucesso)
429             mensagem = self.maquina.fila[0]
430
431             #gera evento de FimDeEnvio e o salva
432             fim_de_envio = FimDeEnvio(mensagem.rodada, self.maquina)
433             simulador.eventos.adicionar(
434                 simulador.tempo_agora + simulador.tempo_transmissao_quadro,
435                 fim_de_envio
436             )
437             self.maquina.fim_de_envio = fim_de_envio
438
439             #gera evento de InicioDeRecebimento no hub
440             simulador.eventos.adicionar(
441                 simulador.tempo_agora + (self.maquina.distancia * simulador.tempo_pr
442                 opagacao),
443                 InicioDeRecebimento(mensagem.rodada, HUB, self.maquina)
444             )
445
446             #atualiza estado da máquina
447             self.maquina.tentativas_de_transmissao += 1
448             self.maquina.tempo_comeco_envio_quadro = simulador.tempo_agora
449             self.maquina.enviando = True
450             self.maquina.agendado = False
451
452             self.maquina.checar_jam(simulador)
453
454     class FimDeEnvio(Evento):
455         """Representa o momento em que um quadro ou reforço de jam terminou
456         de ser enviado."""
457
458         def __init__(self, rodada, maquina, sou_jam = False):
459             self.rodada = rodada
460             self.maquina = maquina
461             self.cancelado = False
462             self.sou_jam = sou_jam
463
464         def processar(self, simulador):
465             if self.cancelado: return
466
467             if self.sou_jam:
468                 debug_print("- Evento: FimDeEnvio (Jam) em t=%f na maquina=%s" % (
469                     simulador.tempo_agora, self.maquina.hostname ))
470             else:
471                 debug_print("- Evento: FimDeEnvio (Quadro) em t=%f na maquina=%s" %
472                     (
473                         simulador.tempo_agora, self.maquina.hostname ))
474
475             #gera evento de FimDeRecebimento no hub
476             simulador.eventos.adicionar(
477                 simulador.tempo_agora + (self.maquina.distancia * simulador.tempo_pr
478                 opagacao),
479                 FimDeRecebimento(self.rodada, HUB, self.maquina)
480             )
481
482             if not self.sou_jam:
483                 #debug_print("      TAp = %f" % (self.maquina.tempo_comeco_envio_quad
484                 ro - self.maquina.tempo_considerar_envio_quadro))
485
486                 # Coleta estatísticas (se rodada valida)
487                 if self.rodada == simulador.rodada_atual:
488                     self.maquina.tap_rodada.adicionar_amostra(self.maquina.tempo_com
489                     eco_envio_quadro - self.maquina.tempo_considerar_envio_quadro)
490
491                 # Se é o último quadro desta mensagem
492                 if self.maquina.proximo_quadro+1 == self.maquina.fila[0].num_qua
493                 dros:
```


simulador.py

```
489         self.maquina.tam_rodada.adicionar_amostra(self.maquina.tempo
_comeco_envio_quadro - self.maquina.tempo_considerar_envio_mensagem)
490
491         self.maquina.quadros_com_sucesso += 1
492
493         #reiniciar estado de envio de quadro da máquina
494         self.maquina.tentativas_de_transmissao = 0
495
496         #incrementar proximo quadro a enviar
497         self.maquina.andar_fila(simulador)
498
499         #simulador.totais += 1
500
501
502         #tentar enviar próximo quadro (agendar para daqui a 9.6us)
503         self.maquina.enviando = False
504         self.maquina.tempo_comeco_ocioso = simulador.tempo_agora
505         self.maquina.tentar_enviar(simulador)
506
507
508 class InicioDeRecebimento(Evento):
509     """Representa o momento em que uma máquina começa a receber um
510     quadro."""
511
512     def __init__(self, rodada, maquina, maquina_origem):
513         self.rodada = rodada
514         self.maquina = maquina
515         self.maquina_origem = maquina_origem
516
517     def processar(self, simulador):
518         debug_print("- Evento: InicioDeRecebimento em t=%f na maquina=%s, origem
=%s" % (
519             simulador.tempo_agora, self.maquina.hostname, self.maquina_origem.ho
stname ))
520
521         if self.maquina is HUB:
522             for maquina in simulador.hosts:
523                 #gera evento de InicioDeRecebimento nas maquinas
524                 simulador.eventos.adicionar(
525                     simulador.tempo_agora + (maquina.distancia * simulador.tempo
_propagacao),
526                     InicioDeRecebimento(self.rodada, maquina, self.maquina_orige
m)
527                 )
528         else:
529             if self.maquina != self.maquina_origem:
530                 self.maquina.uso_do_meio += 1
531                 self.maquina.checar_jam(simulador)
532
533             debug_print("          uso do meio agora = %d" % self.maquina.uso_
do_meio)
534
535
536 class FimDeRecebimento(Evento):
537     """Representa o momento em que uma máquina termina de receber um
538     quadro."""
539
540     def __init__(self, rodada, maquina, maquina_origem):
541         self.rodada = rodada
542         self.maquina = maquina
543         self.maquina_origem = maquina_origem
544
545     def processar(self, simulador):
546         debug_print("- Evento: FimDeRecebimento em t=%f na maquina=%s, origem =
%s" % (
547             simulador.tempo_agora, self.maquina.hostname, self.maquina_origem.ho
stname ))
548
549         if self.maquina is HUB:
550             for maquina in simulador.hosts:
551                 #gera evento de FimDeRecebimento nas maquinas
552                 simulador.eventos.adicionar(
553                     simulador.tempo_agora + (maquina.distancia * simulador.tempo
_propagacao),
554                     FimDeRecebimento(self.rodada, maquina, self.maquina_origem)
555                 )
556         else:
557             if self.maquina != self.maquina_origem:
558                 self.maquina.uso_do_meio -= 1
559                 if self.maquina.uso_do_meio == 0:
560                     self.maquina.tempo_comeco_ocioso = simulador.tempo_agora
561
562             debug_print("          uso do meio agora = %d" % self.maquina.uso_
do_meio)
563
```

simulador.py

```
564         self.maquina.tentar_enviar(simulador)
565
566 #####
567
568
569 class Simulador(object):
570     """Classe principal que encapsula um simulador."""
571
572     def __init__(self,
573                 hosts,
574                 eventos_fase_transiente=50000,
575                 eventos_por_rodada=50000,
576                 titulo=u"",
577                 tempo_minimo_ocioso=9.6, # Tempo em que o meio tem que ficar ocioso
578                 entre_transmissões
579                 tempo_transmissao_quadro=800, # 8000 bits/quadro / 10 Mbps = 800mic
580                 roseg/quadro
581                 tempo_propagacao=0.005, # 5 microseg/km = 0.005 microseg/m
582                 tempo_reforco_jam=3.2,
583                 tempo_fatia_backoff=51.2,
584                 automatico)
585         numero_de_rodadas=-1, # Número de rodadas da simulação (-1 para aut
586         omático)
587         ignorar_backoff = False, # Apenas para cenários de teste
588         ignorar_colisao = False # Apenas para cenários de teste
589
590     ):
591         """Recebe todos os parâmetros da simulação."""
592         self.hosts = hosts
593         self.eventos_fase_transiente = eventos_fase_transiente
594         self.eventos_por_rodada = eventos_por_rodada
595         self.titulo = titulo
596
597         self.tempo_minimo_ocioso = tempo_minimo_ocioso
598         self.tempo_transmissao_quadro = tempo_transmissao_quadro
599         self.tempo_propagacao = tempo_propagacao
600         self.tempo_reforco_jam = tempo_reforco_jam
601         self.tempo_fatia_backoff = tempo_fatia_backoff
602         self.numero_de_rodadas = numero_de_rodadas
603         self.ignorar_backoff = ignorar_backoff
604         self.ignorar_colisao = ignorar_colisao
605
606     def start(self):
607         """Prepara o simulador, inicializando algumas variáveis e
608         gerando os eventos iniciais"""
609
610         self.utilizacao_global = Estatisticas()
611         self.utilizacao_global_media = Estatisticas()
612
613         self.utilizacao_total = Estatisticas()
614         self.tempo_ocupado_total = 0
615
616         self.eventos = HeapDeEventos()
617         self.tempo_agora = 0
618
619         for host in self.hosts:
620             host.reset()
621             if host.ativo:
622                 self.eventos.adicionar(
623                     host.chegada(),
624                     ChegouMensagem(0, host)
625                 )
626
627     def run(self):
628         """Executa o loop principal do simulador até conseguir coletar
629         as estatísticas com a precisão desejada, e então desenha alguns
630         gráficos."""
631
632         # Neste simulador, a rodada zero é considerada a fase transiente
633         print "Fase transiente..."
634         self.rodada_atual = 0
635
636         wallclock_comeco_simulacao = time.time()
637
638         # Loop que executa todas as rodadas
639         while True:
640             #for bla in xrange(3): # DEBUG: Roda só 2 rodadas para testar rapidamen
641             te
642
643             # Número de rodadas "automático"
644             if self.numero_de_rodadas < 0:
645                 # Condição de parada:
646                 # - não estou na fase transiente
647                 # - e todos os hosts chegaram à precisão desejada
648                 # - e a utilização do Ethernet chegou à precisão desejada
649                 if self.rodada_atual > 0 \
650                     and all(
```

simulador.py

```

645         host.precisao_suficiente()
646         for host in self.hosts if host.ativo
647     ) and self.utilizacao_global.media.precisao_suficiente():
648         break
649     # Número de rodadas "fixo"
650     else:
651         if self.rodada_atual > self.numero_de_rodadas:
652             break
653
654     # Wallclock é relógio de parede, e armazena o tempo
655     # decorrido no "mundo real"
656     wallclock_comeco_rodada = time.time()
657
658     # Reiniciando estatísticas para a próxima rodada
659     for host in self.hosts:
660         host.reiniciar_estatisticas()
661
662     self.tempo_ocupado_rodada = 0
663     self.tempo_comeco_rodada = self.tempo_agora
664     self.tempo_evento_anterior = self.tempo_agora
665
666     # Quantos eventos devem ser simulados nesta rodada?
667     if self.rodada_atual == 0:
668         eventos_nesta_rodada = self.eventos_fase_transiente
669     else:
670         eventos_nesta_rodada = self.eventos_por_rodada
671
672     # Executa os eventos dentro de uma rodada
673     for iteracao in xrange(eventos_nesta_rodada):
674         # Retirar evento da fila
675         self.tempo_agora, evento = self.eventos.remove()
676
677         # Atualizar estatística de utilização
678         # (coletada apenas para a primeira máquina)
679         if self.hosts[0].enviando or self.hosts[0].uso_do_meio > 0:
680             self.tempo_ocupado_rodada += self.tempo_agora - self.tempo_e
681             self.tempo_ocupado_total += self.tempo_agora - self.tempo_ev
682             ento_anterior
683
684             self.tempo_evento_anterior = self.tempo_agora
685
686             # Processar evento
687             evento.processar(self)
688
689             # Coletar utilização ethernet (de vez em quando)
690             if iteracao % 1000 == 0:
691                 self.utilizacao_total.adicionar_amostra(self.tempo_ocupado_t
692                 otal / self.tempo_agora)
693
694             # Coletar e exibir estatísticas
695             if self.rodada_atual > 0:
696                 tempo_duracao_da_rodada = self.tempo_agora - self.tempo_comeco_r
697                 odada
698
699                 # Coletando estatísticas...
700                 self.utilizacao_global.adicionar_amostra(self.tempo_ocupado_rod
701                 da / tempo_duracao_da_rodada)
702                 self.utilizacao_global.media.adicionar_amostra(self.utilizacao_g
703                 lobal.media())
704                 self.utilizacao_global.media.adicionar_intervalo(self.utilizacao
705                 _global.intervalo_de_confianca())
706
707                 # Coletando estatísticas...
708                 for host in self.hosts:
709                     host.finalizar_rodada(tempo_duracao_da_rodada)
710
711                 # Calculando o tempo real gasto na simulação
712                 wallclock_fim_rodada = time.time()
713                 wallclock_duracao_rodada = wallclock_fim_rodada - wallclock_come
714                 co_rodada
715                 wallclock_duracao_simulacao = wallclock_fim_rodada - wallclock_c
716                 omeço_simulacao
717
718                 print "Rodada %d" % (self.rodada_atual,)
719                 print "-Tempo real: %.2f segundos nesta rodada, %.2f no total" %
720                 (wallclock_duracao_rodada, wallclock_duracao_simulacao)
721                 print "-Tempo simulado: %.2f microseg nesta rodada, %.2f total"
722                 % (tempo_duracao_da_rodada, self.tempo_agora)
723                 print "-Media uso Ether  =%13f | IC +-%13f | %12f na rodada" % (
724                 self.utilizacao_global.media(), self.utilizacao_global.intervalo_de_confianca(),
725                 self.utilizacao_global.amostras[-1])
726                 for i, host in enumerate(self.hosts):
727                     if host.ativo:
728                         print "-Media do TAp(%d)  =%13f | IC +-%13f | %12f na ro

```

simulador.py

```

    dada" % (i+1, host.tap_global.media(), host.tap_global.intervalo_de_confianca(
    ), host.tap_rodada.media())
717         print "-Media do TAm(%d) =%13f | IC +-%13f | %12f na ro
dada" % (i+1, host.tam_global.media(), host.tam_global.intervalo_de_confianca(
    ), host.tam_rodada.media())
718         print "-Media do Ncm(%d) =%13f | IC +-%13f | %12f na ro
dada" % (i+1, host.ncm_global.media(), host.ncm_global.intervalo_de_confianca(
    ), host.ncm_rodada.media())
719         print "-Media da Vazao(%d)=%13f | IC +-%13f | %12f na ro
dada" % (i+1, host.vazao_global.media(), host.vazao_global.intervalo_de_confianc
a(), host.vazao_global.amostras[-1])
720
721         self.rodada_atual += 1
722
723     def exibir_graficos(self):
724         """Exibe na tela os gráficos já gerados."""
725         pyplot.show()
726
727     def salvar_graficos(self, filename, dpi=300, size=(7.4, 10.0)):
728         """Salva em disco os gráficos já gerados. Por padrão, salva a
729         300 DPI no tamanho de uma folha A4 (já desconsiderando as
730         margens)."""
731
732         fig = pyplot.gcf()
733         fig.set_size_inches(size)
734
735         pyplot.savefig(filename, dpi=dpi)
736
737     def gerar_graficos(self, layout="horizontal"):
738         """Gera os 6 gráficos disponíveis, usando layout "horizontal"
739         (para tela) ou "vertical" (para impressão)"""
740
741         graficos = [
742             # (pos_h, pos_v, função)
743             # Onde pos_h é a posição do gráfico no layout horizontal,
744             # e pos_v é a posição do gráfico no layout vertical.
745
746             (1, 1, self.gerar_grafico_tap),
747             (4, 2, self.gerar_grafico_tam),
748             (2, 3, self.gerar_grafico_ncm),
749             (5, 4, self.gerar_grafico_vazao),
750             (3, 5, self.gerar_grafico_utilizacao),
751             (6, 6, self.gerar_grafico_utilizacao_total),
752         ]
753
754         if layout == "horizontal":
755             rows = 2
756             cols = 3
757             pos_index = 0
758         elif layout == "vertical":
759             rows = 3
760             cols = 2
761             pos_index = 1
762         else:
763             raise ValueError("Layout desconhecido: '%s' % layout)
764
765         # Fecha a figura atual (se existir) e cria uma nova
766         pyplot.close()
767         pyplot.figure()
768
769         # Ajustando o espaço vertical entre os gráficos
770         pyplot.subplots_adjust(hspace=0.25)
771
772         # Desenha os 6 gráficos
773         for grafico in graficos:
774             pyplot.subplot(rows, cols, grafico[pos_index])
775             grafico[-1]() # Chama a função (último elemento da tupla)
776
777         pyplot.suptitle(self.titulo, fontsize="large", fontweight="bold")
778
779     def gerar_grafico_tap(self):
780         for host in self.hosts:
781             if host.ativo:
782                 host.tap_global_media.plot(label=host.hostname)
783             #exibir_legenda()
784             pyplot.grid(True)
785             pyplot.title(u"Tap (µs)\ntempo médio de acesso de um quadro", fontsize="
small")
786             pyplot.xlim(0, self.rodada_atual)
787             pyplot.xticks(fontsize="x-small")
788             pyplot.yticks(fontsize="x-small")
789
790     def gerar_grafico_tam(self):
791         for host in self.hosts:
792             if host.ativo:

```

simulador.py

```
793         host.tam_global_media.plot(label=host.hostname)
794         #exibir_legenda()
795         pyplot.grid(True)
796         pyplot.title(u"TAm (μs)\ntempo médio de acesso de uma msg.", fontsize="small")
797     pyplot.xlim(0, self.rodada_atual)
798     pyplot.xticks(fontsize="x-small")
799     pyplot.yticks(fontsize="x-small")
800
801     def gerar_grafico_ncm(self):
802         for host in self.hosts:
803             if host.ativo:
804                 host.ncm_global_media.plot(label=host.hostname)
805                 #exibir_legenda()
806                 pyplot.grid(True)
807                 pyplot.title(u"NCm\nnúm. médio de colisões por quadro", fontsize="small")
808
809     pyplot.xlim(0, self.rodada_atual)
810     pyplot.xticks(fontsize="x-small")
811     pyplot.yticks(fontsize="x-small")
812
813     def gerar_grafico_vazao(self):
814         for host in self.hosts:
815             if host.ativo:
816                 host.vazao_global_media.plot(label=host.hostname)
817                 #exibir_legenda()
818                 pyplot.grid(True)
819                 pyplot.title(u"Vazão média (quadros/segundo)", fontsize="small")
820                 pyplot.xlim(0, self.rodada_atual)
821                 pyplot.xticks(fontsize="x-small")
822                 pyplot.yticks(fontsize="x-small")
823
824     def gerar_grafico_utilizacao(self):
825         self.utilizacao_global_media.plot(label=u"média")
826         self.utilizacao_global.plot(marker="x", color="#C04040", label=u"amostras")
827
828     pyplot.grid(True)
829     pyplot.title(u"Utilização do Ethernet (por rodada)", fontsize="small")
830     pyplot.xlim(0, self.rodada_atual)
831     pyplot.xticks(fontsize="x-small")
832     pyplot.yticks(fontsize="x-small")
833
834     def gerar_grafico_utilizacao_total(self):
835         self.utilizacao_total.plot()
836         pyplot.xscale("log")
837
838         # Marcando o fim da fase transiente
839         pos_x = self.eventos_fase_transiente/1000.0
840         pyplot.axvline(pos_x, color="red")
841         pyplot.annotate(u"Fim da fase transiente", xy=(pos_x, 0.0625),
842                        rotation="vertical", size="x-small", ha="right", va="bottom")
843
844         # Marcando o fim de cada rodada
845         #for rodada in xrange(1,self.rodada_atual):
846         #    pos_x += self.eventos_por_rodada/1000.0
847         #    pyplot.axvline(pos_x, color="green")
848
849     pyplot.grid(True)
850     # pyplot.grid(True, which="both") # which option has been added in matplotlib 1.0.0
851     pyplot.title(u"Utilização do Ethernet (contínua)", fontsize="small")
852     pyplot.xlabel(u"eventos / 1000", fontsize="small");
853     pyplot.xticks(fontsize="x-small")
854     pyplot.yticks(fontsize="x-small")
```

rodar_cenario.py

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3  # vi:ts=4 sw=4 et
4
5  import cPickle as pickle
6  import sys
7
8  import simulador
9  reload(simulador)
10 from simulador import *
11
12
13 def cenario1():
14     maquinas = [
15         Host(
16             hostname = "maq1",
17             distancia = 100,
18             chegada = Deterministica(80 * 1000),
19             num_quadros = 40
20         ),
21         Host(
22             hostname = "maq2",
23             distancia = 80,
24             chegada = Deterministica(80 * 1000),
25             num_quadros = 40
26         ),
27         Host(
28             hostname = "maq3",
29             distancia = 60,
30             chegada = None,
31             num_quadros = None
32         ),
33         Host(
34             hostname = "maq4",
35             distancia = 40,
36             chegada = None,
37             num_quadros = None
38         ),
39     ]
40     simulador = Simulador(
41         hosts = maquinas,
42         eventos_fase_transiente = 2500000,
43         eventos_por_rodada = 500000,
44         titulo = u"Cenário 1"
45     )
46     return simulador
47
48
49 def cenario2():
50     maquinas = [
51         Host(
52             hostname = "maq1",
53             distancia = 100,
54             chegada = Exponencial(80 * 1000),
55             num_quadros = 40
56         ),
57         Host(
58             hostname = "maq2",
59             distancia = 80,
60             chegada = Exponencial(80 * 1000),
61             num_quadros = 40
62         ),
63         Host(
64             hostname = "maq3",
65             distancia = 60,
66             chegada = None,
67             num_quadros = None
68         ),
69         Host(
70             hostname = "maq4",
71             distancia = 40,
72             chegada = None,
73             num_quadros = None
74         ),
75     ]
76     simulador = Simulador(
77         hosts = maquinas,
78         eventos_fase_transiente = 10**7,
79         eventos_por_rodada = 500000,
80         titulo = u"Cenário 2"
81     )
82     return simulador
83
84
85 def cenario3():
```

rodar_cenario.py

```
86     maquinas = [  
87         Host(  
88             hostname = "maq1",  
89             distancia = 100,  
90             chegada = Deterministica(80 * 1000),  
91             num_quadros = 40  
92         ),  
93         Host(  
94             hostname = "maq2",  
95             distancia = 80,  
96             chegada = Deterministica(16 * 1000),  
97             num_quadros = 1  
98         ),  
99         Host(  
100             hostname = "maq3",  
101             distancia = 60,  
102             chegada = Deterministica(16 * 1000),  
103             num_quadros = 1  
104         ),  
105         Host(  
106             hostname = "maq4",  
107             distancia = 40,  
108             chegada = Deterministica(16 * 1000),  
109             num_quadros = 1  
110         ),  
111     ]  
112     simulador = Simulador(  
113         hosts = maquinas,  
114         eventos_fase_transiente = 250000,  
115         eventos_por_rodada = 500000,  
116         titulo = u"Cenário 3"  
117     )  
118     return simulador  
119  
120  
121 def cenario4():  
122     maquinas = [  
123         Host(  
124             hostname = "maq1",  
125             distancia = 100,  
126             chegada = Deterministica(80 * 1000),  
127             num_quadros = 40  
128         ),  
129         Host(  
130             hostname = "maq2",  
131             distancia = 80,  
132             chegada = Exponencial(16 * 1000),  
133             num_quadros = 1  
134         ),  
135         Host(  
136             hostname = "maq3",  
137             distancia = 60,  
138             chegada = Exponencial(16 * 1000),  
139             num_quadros = 1  
140         ),  
141         Host(  
142             hostname = "maq4",  
143             distancia = 40,  
144             chegada = Exponencial(16 * 1000),  
145             num_quadros = 1  
146         ),  
147     ]  
148     simulador = Simulador(  
149         hosts = maquinas,  
150         eventos_fase_transiente = 2500000,  
151         eventos_por_rodada = 500000,  
152         titulo = u"Cenário 4"  
153     )  
154     return simulador  
155  
156  
157 cenarios = {  
158     "1": cenario1,  
159     "2": cenario2,  
160     "3": cenario3,  
161     "4": cenario4,  
162 }  
163  
164  
165 def print_help():  
166     print "Digite: %s <cenário>" % (sys.argv[0],)  
167     print "Cenários disponíveis: " + " ".join(sorted(cenarios.keys()))  
168  
169  
170 def main():
```

rodar_cenario.py

```
171     if len(sys.argv) != 2 or sys.argv[1] not in cenarios:
172         print_help()
173         sys.exit(1)
174
175     id = sys.argv[1]
176     file_prefix = "cenario_%s" % (id,)
177
178     simulador = cenarios[id]()
179     simulador.start()
180     simulador.run()
181
182     # Salvando os resultados num arquivo
183     pickle.dump(simulador, file(file_prefix + ".pickle", "wb"), protocol=2)
184     # Depois, é possível recarregar os resultados usando:
185     # simulador = pickle.load(file("cenario_1.pickle", "rb"))
186     # Depois de carregado, é possível acessar normalmente todos os
187     # membros do objeto simulador, e inclusive gerar novos gráficos.
188     # Só não é possível continuar/recomeçar a simulação.
189
190     # Salvando os gráficos num arquivo
191     simulador.gerar_graficos(layout="vertical")
192     simulador.salvar_graficos(file_prefix + ".png")
193     simulador.salvar_graficos(file_prefix + ".eps")
194     simulador.salvar_graficos(file_prefix + ".svg")
195     # Também é possível salvar em formatos .eps, .ps, .svg, .pdf
196
197     # Exibindo os gráficos na tela
198     simulador.gerar_graficos()
199     simulador.exibir_graficos()
200
201 if __name__ == "__main__":
202     main()
```


rodar_cenario_teste.py

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3  # vi:ts=4 sw=4 et
4
5  import cPickle as pickle
6  import sys
7
8  import simulador
9  reload(simulador)
10 from simulador import *
11
12 #uma máquina sozinha enviando para a rede
13 def testel():
14     maquinas = [
15         Host(
16             hostname = "maq1",
17             distancia = 100,
18             chegada = Deterministica(40 * 1000),
19             num_quadros = 40
20         ),
21         Host(
22             hostname = "maq2",
23             distancia = 80,
24             chegada = None,
25             num_quadros = None
26         ),
27         Host(
28             hostname = "maq3",
29             distancia = 60,
30             chegada = None,
31             num_quadros = None
32         ),
33         Host(
34             hostname = "maq4",
35             distancia = 40,
36             chegada = None,
37             num_quadros = None
38         ),
39     ]
40     simulador = Simulador(
41         hosts = maquinas,
42         eventos_fase_transiente = 500000,
43         eventos_por_rodada = 30000,
44         titulo = u"Cenário de teste 1",
45         numero_de_rodadas = 10
46     )
47     return simulador
48
49 #uma máquina enviando sozinha para a rede com tempo de chegada o dobro do testel
50 def teste2():
51     maquinas = [
52         Host(
53             hostname = "maq1",
54             distancia = 100,
55             chegada = Deterministica(80 * 1000),
56             num_quadros = 40
57         ),
58         Host(
59             hostname = "maq2",
60             distancia = 80,
61             chegada = None,
62             num_quadros = None
63         ),
64         Host(
65             hostname = "maq3",
66             distancia = 60,
67             chegada = None,
68             num_quadros = None
69         ),
70         Host(
71             hostname = "maq4",
72             distancia = 40,
73             chegada = None,
74             num_quadros = None
75         ),
76     ]
77     simulador = Simulador(
78         hosts = maquinas,
79         eventos_fase_transiente = 500000,
80         eventos_por_rodada = 30000,
81         titulo = u"Cenário de teste 2",
82         numero_de_rodadas = 10
83     )
84     return simulador
85
```

rodar_cenario_teste.py

```
86 #duas máquinas enviando para rede, sem backoff
87 def teste3():
88     maquinas = [
89         Host(
90             hostname = "maq1",
91             distancia = 100,
92             chegada = Deterministica(200 * 1000),
93             num_quadros = 1
94         ),
95         Host(
96             hostname = "maq2",
97             distancia = 80,
98             chegada = Deterministica(200 * 1000),
99             num_quadros = 1
100         ),
101         Host(
102             hostname = "maq3",
103             distancia = 60,
104             chegada = None,
105             num_quadros = None
106         ),
107         Host(
108             hostname = "maq4",
109             distancia = 40,
110             chegada = None,
111             num_quadros = None
112         ),
113     ]
114     simulador = Simulador(
115         hosts = maquinas,
116         eventos_fase_transiente = 1000,
117         eventos_por_rodada = 1000,
118         titulo = u"Cenário de teste 3",
119         numero_de_rodadas = 10,
120         ignorar_backoff = True
121     )
122     return simulador
123
124 #duas máquinas enviando para a rede, sem colisão
125 def teste4():
126     maquinas = [
127         Host(
128             hostname = "maq1",
129             distancia = 100,
130             chegada = Deterministica(80 * 1000),
131             num_quadros = 40
132         ),
133         Host(
134             hostname = "maq2",
135             distancia = 80,
136             chegada = Deterministica(80 * 1000),
137             num_quadros = 40
138         ),
139         Host(
140             hostname = "maq3",
141             distancia = 60,
142             chegada = None,
143             num_quadros = None
144         ),
145         Host(
146             hostname = "maq4",
147             distancia = 40,
148             chegada = None,
149             num_quadros = None
150         ),
151     ]
152     simulador = Simulador(
153         hosts = maquinas,
154         eventos_fase_transiente = 500000,
155         eventos_por_rodada = 30000,
156         titulo = u"Cenário de teste 4",
157         numero_de_rodadas = 10,
158         ignorar_colisao = True
159     )
160     return simulador
161
162
163 #dez máquinas enviando para a rede, com utilização média de 0,5mbps cada
164 def teste5():
165     maquinas = [
166         Host(
167             hostname = "maq1",
168             distancia = 100,
169             chegada = Deterministica(320 * 1000),
170             num_quadros = 20
```

rodar_cenario_teste.py

```
171         ),
172         Host(
173             hostname = "maq2",
174             distancia = 80,
175             chegada = Deterministica(320 * 1000),
176             num_quadros = 20
177         ),
178         Host(
179             hostname = "maq3",
180             distancia = 60,
181             chegada = Exponencial(80 * 1000),
182             num_quadros = 5
183         ),
184         Host(
185             hostname = "maq4",
186             distancia = 40,
187             chegada = Exponencial(160 * 1000),
188             num_quadros = 10
189         ),
190         Host(
191             hostname = "maq5",
192             distancia = 30,
193             chegada = Deterministica(80 * 1000),
194             num_quadros = 5
195         ),
196         Host(
197             hostname = "maq6",
198             distancia = 42,
199             chegada = Deterministica(640 * 1000),
200             num_quadros = 40
201         ),
202         Host(
203             hostname = "maq7",
204             distancia = 11,
205             chegada = Exponencial(320 * 1000),
206             num_quadros = 20
207         ),
208         Host(
209             hostname = "maq8",
210             distancia = 70,
211             chegada = Exponencial(160 * 1000),
212             num_quadros = 10
213         ),
214         Host(
215             hostname = "maq9",
216             distancia = 55,
217             chegada = Deterministica(80 * 1000),
218             num_quadros = 5
219         ),
220         Host(
221             hostname = "maq10",
222             distancia = 33,
223             chegada = Exponencial(80 * 1000),
224             num_quadros = 5
225         ),
226     ]
227     simulador = Simulador(
228         hosts = maquinas,
229         eventos_fase_transiente = 10000000,
230         eventos_por_rodada = 800000,
231         titulo = u"Cenário de teste 5"
232     )
233     return simulador
234
235     cenarios = {
236         "1": teste1,
237         "2": teste2,
238         "3": teste3,
239         "4": teste4,
240         "5": teste5,
241     }
242
243
244     def print_help():
245         print "Digite: %s <cenário>" % (sys.argv[0],)
246         print "Cenários disponíveis: " + " ".join(sorted(cenarios.keys()))
247
248
249     def main():
250         if len(sys.argv) != 2 or sys.argv[1] not in cenarios:
251             print_help()
252             sys.exit(1)
253
254         id = sys.argv[1]
255         file_prefix = "cenario_teste_%s" % (id,)
```

rodar_cenario_teste.py

```
256
257     simulador = cenarios[id]()
258     simulador.start()
259     simulador.run()
260
261     # Salvando os resultados num arquivo
262     pickle.dump(simulador, file(file_prefix + ".pickle", "wb"), protocol=2)
263     # Depois, é possível recarregar os resultados usando:
264     #     simulador = pickle.load(file("cenario_1.pickle", "rb"))
265     # Depois de carregado, é possível acessar normalmente todos os
266     # membros do objeto simulador, e inclusive gerar novos gráficos.
267     # Só não é possível continuar/recomeçar a simulação.
268
269     # Salvando os gráficos num arquivo
270     simulador.gerar_graficos(layout="vertical")
271     simulador.salvar_graficos(file_prefix + ".png")
272     simulador.salvar_graficos(file_prefix + ".eps")
273     simulador.salvar_graficos(file_prefix + ".svg")
274     # Também é possível salvar em formatos .eps, .ps, .svg, .pdf
275
276     # Exibindo os gráficos na tela
277     simulador.gerar_graficos()
278     simulador.exibir_graficos()
279
280 if __name__ == "__main__":
281     main()
```

Tabelas com os resultados e comentários pertinentes

Cenário 1

Rodada 1

-Tempo real: 14.49 segundos nesta rodada, 92.26 no total
-Tempo simulado: 34630988.30 microseg nesta rodada, 208720810.80 total
-Media uso Ether = 0.823431 | IC +- 0.000000 | 0.823431 na rodada
-Media do TAp(1) = 870.222487 | IC +- 0.000000 | 870.222487 na rodada
-Media do TAm(1) = 66336.529461 | IC +- 0.000000 | 66336.529461 na rodada
-Media do Ncm(1) = 0.184861 | IC +- 0.000000 | 0.184861 na rodada
-Media da Vazao(1)= 497.473530 | IC +- 0.000000 | 497.473530 na rodada
-Media do TAp(2) = 844.753161 | IC +- 0.000000 | 844.753161 na rodada
-Media do TAm(2) = 65212.163461 | IC +- 0.000000 | 65212.163461 na rodada
-Media do Ncm(2) = 0.188232 | IC +- 0.000000 | 0.188232 na rodada
-Media da Vazao(2)= 493.806294 | IC +- 0.000000 | 493.806294 na rodada

Rodada 2

-Tempo real: 14.32 segundos nesta rodada, 106.59 no total
-Tempo simulado: 34802134.00 microseg nesta rodada, 243522944.80 total
-Media uso Ether = 0.823917 | IC +- 0.006175 | 0.824403 na rodada
-Media do TAp(1) = 869.598373 | IC +- 7.930119 | 868.974259 na rodada
-Media do TAm(1) = 66301.998677 | IC +- 438.755217 | 66267.467893 na rodada
-Media do Ncm(1) = 0.180883 | IC +- 0.050542 | 0.176905 na rodada
-Media da Vazao(1)= 497.327843 | IC +- 1.851132 | 497.182156 na rodada
-Media do TAp(2) = 856.234372 | IC +- 145.882620 | 867.715583 na rodada
-Media do TAm(2) = 65567.538571 | IC +- 4515.468913 | 65922.913682 na rodada
-Media do Ncm(2) = 0.183820 | IC +- 0.056057 | 0.179408 na rodada
-Media da Vazao(2)= 494.632209 | IC +- 10.494239 | 495.458123 na rodada

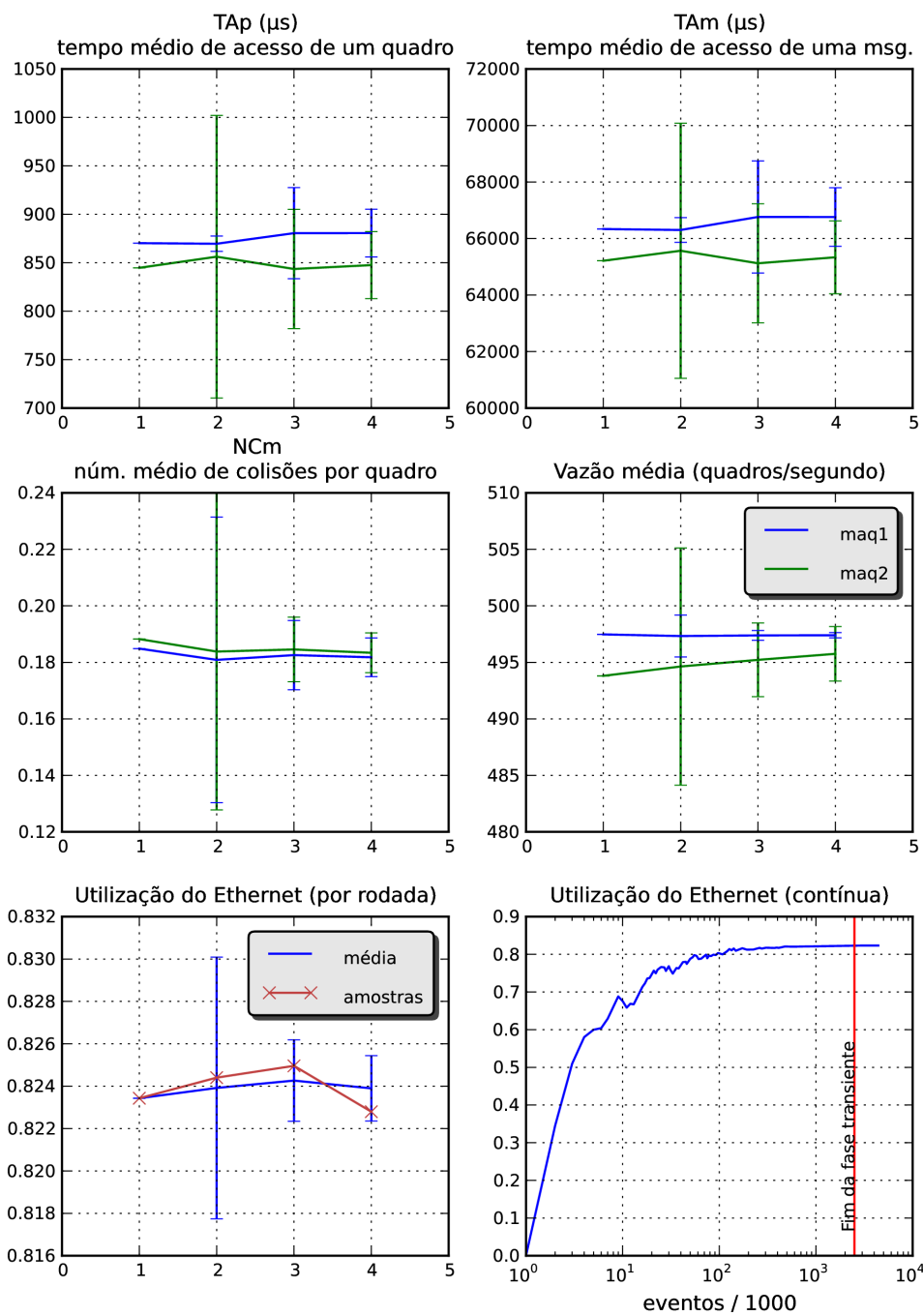
Rodada 3

-Tempo real: 14.81 segundos nesta rodada, 121.41 no total
-Tempo simulado: 34613277.00 microseg nesta rodada, 278136221.80 total
-Media uso Ether = 0.824264 | IC +- 0.001922 | 0.824960 na rodada
-Media do TAp(1) = 880.526263 | IC +- 47.044472 | 902.382044 na rodada
-Media do TAm(1) = 66763.103149 | IC +- 1985.825928 | 67685.312092 na rodada
-Media do Ncm(1) = 0.182565 | IC +- 0.012249 | 0.185930 na rodada
-Media da Vazao(1)= 497.384214 | IC +- 0.435667 | 497.496958 na rodada
-Media do TAp(2) = 843.561866 | IC +- 61.534223 | 818.216855 na rodada
-Media do TAm(2) = 65123.336722 | IC +- 2105.278966 | 64234.933022 na rodada
-Media do Ncm(2) = 0.184562 | IC +- 0.011415 | 0.186047 na rodada
-Media da Vazao(2)= 495.221177 | IC +- 3.260552 | 496.399113 na rodada

Rodada 4

-Tempo real: 14.22 segundos nesta rodada, 135.63 no total
-Tempo simulado: 34870453.60 microseg nesta rodada, 313006675.40 total
-Media uso Ether = 0.823897 | IC +- 0.001541 | 0.822796 na rodada
-Media do TAp(1) = 880.610026 | IC +- 24.606162 | 880.861317 na rodada
-Media do TAm(1) = 66760.907776 | IC +- 1038.629837 | 66754.321658 na rodada
-Media do Ncm(1) = 0.181797 | IC +- 0.006857 | 0.179493 na rodada
-Media da Vazao(1)= 497.398459 | IC +- 0.232324 | 497.441192 na rodada
-Media do TAp(2) = 847.543727 | IC +- 34.587953 | 859.489310 na rodada
-Media do TAm(2) = 65335.330890 | IC +- 1291.335176 | 65971.313394 na rodada
-Media do Ncm(2) = 0.183384 | IC +- 0.007050 | 0.179850 na rodada
-Media da Vazao(2)= 495.754672 | IC +- 2.406376 | 497.355159 na rodada

Cenário 1



Cenário 2

Rodada 1

-Tempo real: 14.20 segundos nesta rodada, 292.86 no total
-Tempo simulado: 39606325.07 microseg nesta rodada, 792220030.21 total
-Media uso Ether = 0.763700 | IC +- 0.000000 | 0.763700 na rodada
-Media do TAp(1) = 540.313308 | IC +- 0.000000 | 540.313308 na rodada
-Media do TAm(1) = 53165.254173 | IC +- 0.000000 | 53165.254173 na rodada
-Media do Ncm(1) = 0.109197 | IC +- 0.000000 | 0.109197 na rodada
-Media da Vazao(1)= 477.751974 | IC +- 0.000000 | 477.751974 na rodada
-Media do TAp(2) = 606.454807 | IC +- 0.000000 | 606.454807 na rodada
-Media do TAm(2) = 55806.044912 | IC +- 0.000000 | 55806.044912 na rodada
-Media do Ncm(2) = 0.114833 | IC +- 0.000000 | 0.114833 na rodada
-Media da Vazao(2)= 454.876840 | IC +- 0.000000 | 454.876840 na rodada

Rodada 2

-Tempo real: 13.10 segundos nesta rodada, 305.96 no total
-Tempo simulado: 38570081.28 microseg nesta rodada, 830790111.49 total
-Media uso Ether = 0.777356 | IC +- 0.173507 | 0.791011 na rodada
-Media do TAp(1) = 628.302712 | IC +- 1118.011381 | 716.292116 na rodada
-Media do TAm(1) = 56667.935968 | IC +- 44505.792010 | 60170.617762 na rodada
-Media do Ncm(1) = 0.110334 | IC +- 0.014447 | 0.111471 na rodada
-Media da Vazao(1)= 459.137385 | IC +- 236.520789 | 440.522795 na rodada
-Media do TAp(2) = 541.155780 | IC +- 829.702807 | 475.856753 na rodada
-Media do TAm(2) = 53190.909748 | IC +- 33228.442817 | 50575.774583 na rodada
-Media do Ncm(2) = 0.103923 | IC +- 0.138634 | 0.093012 na rodada
-Media da Vazao(2)= 490.686504 | IC +- 455.004922 | 526.496168 na rodada

Rodada 3

-Tempo real: 13.69 segundos nesta rodada, 319.65 no total
-Tempo simulado: 37061332.31 microseg nesta rodada, 867851443.81 total
-Media uso Ether = 0.790863 | IC +- 0.067293 | 0.817878 na rodada
-Media do TAp(1) = 621.510286 | IC +- 220.522970 | 607.925433 na rodada
-Media do TAm(1) = 56392.589571 | IC +- 8781.427398 | 55841.896778 na rodada
-Media do Ncm(1) = 0.110128 | IC +- 0.002960 | 0.109716 na rodada
-Media da Vazao(1)= 470.719598 | IC +- 67.983092 | 493.884026 na rodada
-Media do TAp(2) = 583.322555 | IC +- 243.370371 | 667.656103 na rodada
-Media do TAm(2) = 54874.738723 | IC +- 9730.964487 | 58242.396673 na rodada
-Media do Ncm(2) = 0.105285 | IC +- 0.027731 | 0.108011 na rodada
-Media da Vazao(2)= 494.396610 | IC +- 90.377105 | 501.816822 na rodada

Rodada 4

-Tempo real: 15.11 segundos nesta rodada, 334.77 no total
-Tempo simulado: 36862499.89 microseg nesta rodada, 904713943.70 total
-Media uso Ether = 0.799524 | IC +- 0.044703 | 0.825506 na rodada
-Media do TAp(1) = 656.737836 | IC +- 160.844396 | 762.420486 na rodada
-Media do TAm(1) = 57796.132932 | IC +- 6406.635459 | 62006.763016 na rodada
-Media do Ncm(1) = 0.110173 | IC +- 0.001555 | 0.110308 na rodada
-Media da Vazao(1)= 472.008841 | IC +- 35.791766 | 475.876570 na rodada
-Media do TAp(2) = 561.088174 | IC +- 145.631142 | 494.385033 na rodada
-Media do TAm(2) = 53993.107484 | IC +- 5811.547481 | 51348.213767 na rodada
-Media do Ncm(2) = 0.104028 | IC +- 0.015045 | 0.100257 na rodada
-Media da Vazao(2)= 503.066024 | IC +- 54.730969 | 529.074264 na rodada

Rodada 5

-Tempo real: 14.11 segundos nesta rodada, 348.88 no total
-Tempo simulado: 37888684.33 microseg nesta rodada, 942602628.03 total
-Media uso Ether = 0.799196 | IC +- 0.030223 | 0.797883 na rodada
-Media do TAp(1) = 646.481527 | IC +- 112.363206 | 605.456289 na rodada
-Media do TAm(1) = 57393.441018 | IC +- 4471.493783 | 55782.673360 na rodada
-Media do Ncm(1) = 0.111297 | IC +- 0.003293 | 0.115792 na rodada
-Media da Vazao(1)= 472.252745 | IC +- 24.196743 | 473.228361 na rodada
-Media do TAp(2) = 565.056573 | IC +- 99.029073 | 580.930166 na rodada
-Media do TAm(2) = 54146.268557 | IC +- 3950.267415 | 54758.912851 na rodada
-Media do Ncm(2) = 0.105065 | IC +- 0.010567 | 0.109211 na rodada
-Media da Vazao(2)= 502.651600 | IC +- 37.003843 | 500.993907 na rodada

Rodada 6

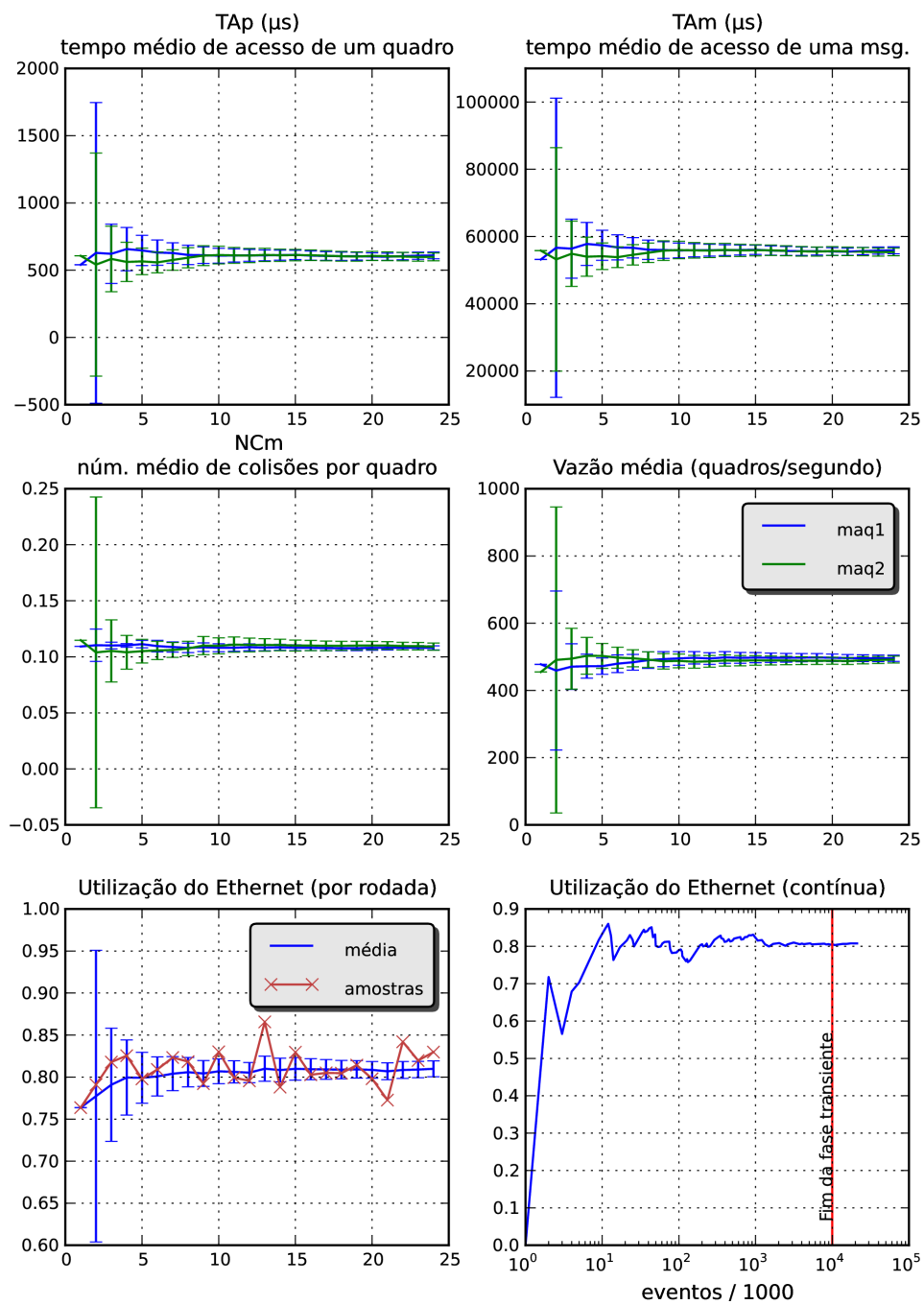
-Tempo real: 15.03 segundos nesta rodada, 363.92 no total
-Tempo simulado: 37571293.78 microseg nesta rodada, 980173921.81 total
-Media uso Ether = 0.800843 | IC +- 0.023236 | 0.809080 na rodada
-Media do TAp(1) = 630.952568 | IC +- 93.853993 | 553.307776 na rodada
-Media do TAm(1) = 56774.404559 | IC +- 3736.081101 | 53679.222263 na rodada
-Media do Ncm(1) = 0.109569 | IC +- 0.005092 | 0.100928 na rodada
-Media da Vazao(1)= 479.567077 | IC +- 26.231738 | 516.138734 na rodada
-Media do TAp(2) = 557.626975 | IC +- 77.259407 | 520.478986 na rodada

-Media do TAm(2) = 53846.644281 | IC +- 3083.956758 | 52348.522897 na rodada
 -Media do Ncm(2) = 0.105697 | IC +- 0.008152 | 0.108860 na rodada
 -Media da Vazao(2) = 497.415906 | IC +- 31.042612 | 471.237432 na rodada
 Rodada 7
 -Tempo real: 13.78 segundos nesta rodada, 377.70 no total
 -Tempo simulado: 36973434.86 microseg nesta rodada, 1017147356.67 total
 -Media uso Ether = 0.804051 | IC +- 0.020275 | 0.823302 na rodada
 -Media do TAp(1) = 627.365408 | IC +- 76.013464 | 605.842449 na rodada
 -Media do TAm(1) = 56646.686174 | IC +- 3021.858626 | 55880.375869 na rodada
 -Media do Ncm(1) = 0.108763 | IC +- 0.004546 | 0.103928 na rodada
 -Media da Vazao(1) = 483.789251 | IC +- 23.496466 | 509.122295 na rodada
 -Media do TAp(2) = 575.459809 | IC +- 75.942460 | 682.456812 na rodada
 -Media do TAm(2) = 54558.655109 | IC +- 3031.640287 | 58830.720077 na rodada
 -Media do Ncm(2) = 0.106079 | IC +- 0.006624 | 0.108371 na rodada
 -Media da Vazao(2) = 495.576073 | IC +- 25.376126 | 484.537075 na rodada
 Rodada 8
 -Tempo real: 13.98 segundos nesta rodada, 391.69 no total
 -Tempo simulado: 37018531.95 microseg nesta rodada, 1054165888.62 total
 -Media uso Ether = 0.805843 | IC +- 0.017489 | 0.818382 na rodada
 -Media do TAp(1) = 612.978721 | IC +- 72.140623 | 512.271908 na rodada
 -Media do TAm(1) = 56069.954439 | IC +- 2873.266172 | 52032.832294 na rodada
 -Media do Ncm(1) = 0.108003 | IC +- 0.004208 | 0.102682 na rodada
 -Media da Vazao(1) = 490.032178 | IC +- 24.588695 | 533.732673 na rodada
 -Media do TAp(2) = 592.193120 | IC +- 74.866807 | 709.326299 na rodada
 -Media do TAm(2) = 55163.705461 | IC +- 2912.774890 | 59399.057927 na rodada
 -Media do Ncm(2) = 0.107456 | IC +- 0.006429 | 0.117093 na rodada
 -Media da Vazao(2) = 491.728342 | IC +- 23.104230 | 464.794229 na rodada
 Rodada 9
 -Tempo real: 13.64 segundos nesta rodada, 405.33 no total
 -Tempo simulado: 37894503.93 microseg nesta rodada, 1092060392.55 total
 -Media uso Ether = 0.804326 | IC +- 0.015443 | 0.792196 na rodada
 -Media do TAp(1) = 611.153383 | IC +- 62.187447 | 596.550679 na rodada
 -Media do TAm(1) = 55995.107643 | IC +- 2477.184131 | 55396.333268 na rodada
 -Media do Ncm(1) = 0.108545 | IC +- 0.003829 | 0.112882 na rodada
 -Media da Vazao(1) = 493.129883 | IC +- 22.321474 | 517.911517 na rodada
 -Media do TAp(2) = 607.964324 | IC +- 73.950470 | 734.133956 na rodada
 -Media do TAm(2) = 55801.834221 | IC +- 2905.363649 | 60906.864298 na rodada
 -Media do Ncm(2) = 0.110022 | IC +- 0.008100 | 0.130556 na rodada
 -Media da Vazao(2) = 486.709140 | IC +- 22.996011 | 446.555522 na rodada
 Rodada 10
 -Tempo real: 14.04 segundos nesta rodada, 419.38 no total
 -Tempo simulado: 36607854.11 microseg nesta rodada, 1128668246.66 total
 -Media uso Ether = 0.806902 | IC +- 0.014749 | 0.830084 na rodada
 -Media do TAp(1) = 607.607165 | IC +- 55.151083 | 575.691201 na rodada
 -Media do TAm(1) = 55854.971083 | IC +- 2196.528147 | 54593.742050 na rodada
 -Media do Ncm(1) = 0.108084 | IC +- 0.003518 | 0.103936 na rodada
 -Media da Vazao(1) = 495.237554 | IC +- 20.157313 | 514.206595 na rodada
 -Media do TAp(2) = 612.375574 | IC +- 65.648505 | 652.076824 na rodada
 -Media do TAm(2) = 55980.666688 | IC +- 2581.125134 | 57590.158897 na rodada
 -Media do Ncm(2) = 0.109826 | IC +- 0.007121 | 0.108059 na rodada
 -Media da Vazao(2) = 487.806235 | IC +- 20.329225 | 497.680087 na rodada
 Rodada 11
 -Tempo real: 13.25 segundos nesta rodada, 432.64 no total
 -Tempo simulado: 37768061.28 microseg nesta rodada, 1166436307.94 total
 -Media uso Ether = 0.806218 | IC +- 0.013229 | 0.799380 na rodada
 -Media do TAp(1) = 609.279900 | IC +- 49.276998 | 626.007251 na rodada
 -Media do TAm(1) = 55919.480660 | IC +- 1962.227992 | 56564.576430 na rodada
 -Media do Ncm(1) = 0.108105 | IC +- 0.003134 | 0.108316 na rodada
 -Media da Vazao(1) = 497.064104 | IC +- 18.414163 | 515.329602 na rodada
 -Media do TAp(2) = 609.935659 | IC +- 58.740446 | 585.536513 na rodada
 -Media do TAm(2) = 55888.082732 | IC +- 2308.840332 | 54962.243166 na rodada
 -Media do Ncm(2) = 0.110866 | IC +- 0.006754 | 0.121264 na rodada
 -Media da Vazao(2) = 485.325794 | IC +- 18.936417 | 460.521388 na rodada
 Rodada 12
 -Tempo real: 13.43 segundos nesta rodada, 446.08 no total
 -Tempo simulado: 38054367.86 microseg nesta rodada, 1204490675.80 total
 -Media uso Ether = 0.805327 | IC +- 0.012089 | 0.795523 na rodada
 -Media do TAp(1) = 609.824400 | IC +- 44.451495 | 615.813907 na rodada
 -Media do TAm(1) = 55939.990090 | IC +- 1770.007020 | 56165.593813 na rodada
 -Media do Ncm(1) = 0.108550 | IC +- 0.002991 | 0.113449 na rodada
 -Media da Vazao(1) = 494.855728 | IC +- 17.301679 | 470.563591 na rodada
 -Media do TAp(2) = 608.488205 | IC +- 53.064684 | 592.566211 na rodada
 -Media do TAm(2) = 55822.013018 | IC +- 2087.059876 | 55095.246170 na rodada
 -Media do Ncm(2) = 0.110588 | IC +- 0.006121 | 0.107526 na rodada

-Media da Vazao(2)= 486.493496 | IC +- 17.268169 | 499.338212 na rodada
Rodada 13
-Tempo real: 13.53 segundos nesta rodada, 459.61 no total
-Tempo simulado: 35083954.34 microseg nesta rodada, 1239574630.15 total
-Media uso Ether = 0.809969 | IC +- 0.014949 | 0.865674 na rodada
-Media do TAp(1) = 610.154396 | IC +- 40.483901 | 614.114340 na rodada
-Media do TAm(1) = 55951.584091 | IC +- 1611.965707 | 56090.712105 na rodada
-Media do Ncm(1) = 0.108231 | IC +- 0.002812 | 0.104396 na rodada
-Media da Vazao(1)= 498.150033 | IC +- 17.312885 | 537.681694 na rodada
-Media do TAp(2) = 613.329466 | IC +- 49.458597 | 671.424591 na rodada
-Media do TAm(2) = 56020.200051 | IC +- 1948.915323 | 58398.444444 na rodada
-Media do Ncm(2) = 0.110551 | IC +- 0.005574 | 0.110110 na rodada
-Media da Vazao(2)= 488.981778 | IC +- 16.632771 | 518.841172 na rodada
Rodada 14
-Tempo real: 14.27 segundos nesta rodada, 473.89 no total
-Tempo simulado: 38377105.62 microseg nesta rodada, 1277951735.77 total
-Media uso Ether = 0.808407 | IC +- 0.014132 | 0.788103 na rodada
-Media do TAp(1) = 611.202631 | IC +- 37.232458 | 624.829693 na rodada
-Media do TAm(1) = 55992.028280 | IC +- 1482.334138 | 56517.802744 na rodada
-Media do Ncm(1) = 0.108472 | IC +- 0.002633 | 0.111605 na rodada
-Media da Vazao(1)= 496.859166 | IC +- 16.135748 | 480.077893 na rodada
-Media do TAp(2) = 610.632346 | IC +- 45.774510 | 575.569796 na rodada
-Media do TAm(2) = 55875.675161 | IC +- 1816.111211 | 53996.851595 na rodada
-Media do Ncm(2) = 0.110539 | IC +- 0.005117 | 0.110390 na rodada
-Media da Vazao(2)= 488.498407 | IC +- 15.304266 | 482.214583 na rodada
Rodada 15
-Tempo real: 13.99 segundos nesta rodada, 487.88 no total
-Tempo simulado: 36729204.67 microseg nesta rodada, 1314680940.44 total
-Media uso Ether = 0.809824 | IC +- 0.013410 | 0.829661 na rodada
-Media do TAp(1) = 613.261874 | IC +- 34.693805 | 642.091279 na rodada
-Media do TAm(1) = 56073.307888 | IC +- 1381.071218 | 57211.222390 na rodada
-Media do Ncm(1) = 0.108065 | IC +- 0.002585 | 0.102366 na rodada
-Media da Vazao(1)= 497.468595 | IC +- 14.970391 | 506.000611 na rodada
-Media do TAp(2) = 611.764599 | IC +- 42.376033 | 627.616131 na rodada
-Media do TAm(2) = 55925.187697 | IC +- 1681.868963 | 56618.363205 na rodada
-Media do Ncm(2) = 0.110022 | IC +- 0.004858 | 0.102778 na rodada
-Media da Vazao(2)= 489.227785 | IC +- 14.230978 | 499.439075 na rodada
Rodada 16
-Tempo real: 14.90 segundos nesta rodada, 502.79 no total
-Tempo simulado: 37725398.89 microseg nesta rodada, 1352406339.33 total
-Media uso Ether = 0.809385 | IC +- 0.012501 | 0.802795 na rodada
-Media do TAp(1) = 608.051377 | IC +- 34.109913 | 529.893910 na rodada
-Media do TAm(1) = 55826.567407 | IC +- 1387.383418 | 52125.460193 na rodada
-Media do Ncm(1) = 0.108069 | IC +- 0.002403 | 0.108136 na rodada
-Media da Vazao(1)= 497.107099 | IC +- 13.937753 | 491.684662 na rodada
-Media do TAp(2) = 611.125138 | IC +- 39.416229 | 601.533222 na rodada
-Media do TAm(2) = 55904.795761 | IC +- 1564.065335 | 55598.916720 na rodada
-Media do Ncm(2) = 0.110058 | IC +- 0.004517 | 0.110598 na rodada
-Media da Vazao(2)= 489.114610 | IC +- 13.231282 | 487.416980 na rodada
Rodada 17
-Tempo real: 13.93 segundos nesta rodada, 516.72 no total
-Tempo simulado: 37778333.58 microseg nesta rodada, 1390184672.91 total
-Media uso Ether = 0.809128 | IC +- 0.011692 | 0.805027 na rodada
-Media do TAp(1) = 605.293722 | IC +- 32.398926 | 561.171246 na rodada
-Media do TAm(1) = 55717.627283 | IC +- 1316.574660 | 53974.585301 na rodada
-Media do Ncm(1) = 0.107792 | IC +- 0.002321 | 0.103351 na rodada
-Media da Vazao(1)= 497.113518 | IC +- 13.021337 | 497.216214 na rodada
-Media do TAp(2) = 607.243278 | IC +- 37.732851 | 545.133515 na rodada
-Media do TAm(2) = 55755.441370 | IC +- 1495.134964 | 53365.771110 na rodada
-Media do Ncm(2) = 0.109809 | IC +- 0.004252 | 0.105828 na rodada
-Media da Vazao(2)= 488.949662 | IC +- 12.366255 | 486.310492 na rodada
Rodada 18
-Tempo real: 13.54 segundos nesta rodada, 530.27 no total
-Tempo simulado: 37688103.19 microseg nesta rodada, 1427872776.10 total
-Media uso Ether = 0.808881 | IC +- 0.010983 | 0.804671 na rodada
-Media do TAp(1) = 603.239036 | IC +- 30.708138 | 568.309381 na rodada
-Media do TAm(1) = 55636.501130 | IC +- 1247.171482 | 54257.356535 na rodada
-Media do Ncm(1) = 0.107661 | IC +- 0.002196 | 0.105432 na rodada
-Media da Vazao(1)= 497.419737 | IC +- 12.235272 | 502.625455 na rodada
-Media do TAp(2) = 603.464417 | IC +- 36.292109 | 539.223793 na rodada
-Media do TAm(2) = 55607.871553 | IC +- 1437.050322 | 53099.184658 na rodada
-Media do Ncm(2) = 0.109851 | IC +- 0.003991 | 0.110564 na rodada
-Media da Vazao(2)= 488.418127 | IC +- 11.657591 | 479.382046 na rodada
Rodada 19

-Tempo real: 15.28 segundos nesta rodada, 545.55 no total
-Tempo simulado: 37218359.21 microseg nesta rodada, 1465091135.31 total
-Media uso Ether = 0.809162 | IC +- 0.010362 | 0.814216 na rodada
-Media do TAp(1) = 603.750307 | IC +- 28.944486 | 612.953187 na rodada
-Media do TAm(1) = 55658.128165 | IC +- 1175.611665 | 56047.414783 na rodada
-Media do Ncm(1) = 0.107625 | IC +- 0.002069 | 0.106977 na rodada
-Media da Vazao(1) = 497.973868 | IC +- 11.583276 | 507.948239 na rodada
-Media do TAp(2) = 601.161338 | IC +- 34.524931 | 559.705907 na rodada
-Media do TAm(2) = 55521.751282 | IC +- 1365.622532 | 53971.586411 na rodada
-Media do Ncm(2) = 0.110017 | IC +- 0.003775 | 0.113000 na rodada
-Media da Vazao(2) = 488.203008 | IC +- 10.989792 | 484.330862 na rodada
Rodada 20
-Tempo real: 14.00 segundos nesta rodada, 559.56 no total
-Tempo simulado: 37815774.09 microseg nesta rodada, 1502906909.40 total
-Media uso Ether = 0.808606 | IC +- 0.009862 | 0.798058 na rodada
-Media do TAp(1) = 601.170943 | IC +- 27.883547 | 552.163019 na rodada
-Media do TAm(1) = 55555.580105 | IC +- 1131.631770 | 53607.166966 na rodada
-Media do Ncm(1) = 0.108041 | IC +- 0.002141 | 0.115954 na rodada
-Media da Vazao(1) = 497.168295 | IC +- 11.076630 | 481.862409 na rodada
-Media do TAp(2) = 605.774621 | IC +- 34.028753 | 693.427001 na rodada
-Media do TAm(2) = 55674.859472 | IC +- 1329.861171 | 58583.915070 na rodada
-Media do Ncm(2) = 0.110118 | IC +- 0.003575 | 0.112045 na rodada
-Media da Vazao(2) = 488.494190 | IC +- 10.404503 | 494.026645 na rodada
Rodada 21
-Tempo real: 13.25 segundos nesta rodada, 572.82 no total
-Tempo simulado: 39206578.33 microseg nesta rodada, 1542113487.73 total
-Media uso Ether = 0.806894 | IC +- 0.010008 | 0.772647 na rodada
-Media do TAp(1) = 599.617499 | IC +- 26.630954 | 568.548621 na rodada
-Media do TAm(1) = 55496.331921 | IC +- 1079.861095 | 54311.368247 na rodada
-Media do Ncm(1) = 0.108063 | IC +- 0.002030 | 0.108512 na rodada
-Media da Vazao(1) = 496.197531 | IC +- 10.693908 | 476.782234 na rodada
-Media do TAp(2) = 603.395723 | IC +- 32.638050 | 555.817759 na rodada
-Media do TAm(2) = 55584.350575 | IC +- 1274.741807 | 53774.172636 na rodada
-Media do Ncm(2) = 0.110114 | IC +- 0.003389 | 0.110033 na rodada
-Media da Vazao(2) = 487.372918 | IC +- 10.136800 | 464.947485 na rodada
Rodada 22
-Tempo real: 15.67 segundos nesta rodada, 588.50 no total
-Tempo simulado: 36142201.65 microseg nesta rodada, 1578255689.38 total
-Media uso Ether = 0.808491 | IC +- 0.010076 | 0.842028 na rodada
-Media do TAp(1) = 604.296570 | IC +- 27.120120 | 702.557066 na rodada
-Media do TAm(1) = 55683.406525 | IC +- 1097.725541 | 59611.973220 na rodada
-Media do Ncm(1) = 0.108099 | IC +- 0.001931 | 0.108840 na rodada
-Media da Vazao(1) = 495.955216 | IC +- 10.177687 | 490.866610 na rodada
-Media do TAp(2) = 601.009231 | IC +- 31.418891 | 550.892892 na rodada
-Media do TAm(2) = 55493.532790 | IC +- 1226.349503 | 53586.359316 na rodada
-Media do Ncm(2) = 0.109615 | IC +- 0.003384 | 0.099129 na rodada
-Media da Vazao(2) = 489.770355 | IC +- 10.849109 | 540.116515 na rodada
Rodada 23
-Tempo real: 13.93 segundos nesta rodada, 602.44 no total
-Tempo simulado: 37009790.00 microseg nesta rodada, 1615265479.38 total
-Media uso Ether = 0.808981 | IC +- 0.009655 | 0.819756 na rodada
-Media do TAp(1) = 607.579644 | IC +- 26.724520 | 679.807277 na rodada
-Media do TAm(1) = 55809.560419 | IC +- 1078.239993 | 58584.946069 na rodada
-Media do Ncm(1) = 0.108193 | IC +- 0.001851 | 0.110268 na rodada
-Media da Vazao(1) = 495.448654 | IC +- 9.755003 | 484.304288 na rodada
-Media do TAp(2) = 597.728955 | IC +- 30.702080 | 525.562902 na rodada
-Media do TAm(2) = 55365.978033 | IC +- 1198.149804 | 52559.773387 na rodada
-Media do Ncm(2) = 0.109464 | IC +- 0.003240 | 0.106157 na rodada
-Media da Vazao(2) = 490.608796 | IC +- 10.483277 | 509.054496 na rodada
Rodada 24
-Tempo real: 13.70 segundos nesta rodada, 616.15 no total
-Tempo simulado: 36747917.78 microseg nesta rodada, 1652013397.16 total
-Media uso Ether = 0.809846 | IC +- 0.009393 | 0.829736 na rodada
-Media do TAp(1) = 609.407891 | IC +- 25.801128 | 651.457565 na rodada
-Media do TAm(1) = 55885.368138 | IC +- 1041.612752 | 57628.945674 na rodada
-Media do Ncm(1) = 0.108000 | IC +- 0.001812 | 0.103548 na rodada
-Media da Vazao(1) = 495.595254 | IC +- 9.321152 | 498.967046 na rodada
-Media do TAp(2) = 598.713884 | IC +- 29.391785 | 621.367240 na rodada
-Media do TAm(2) = 55407.512833 | IC +- 1147.477572 | 56362.813235 na rodada
-Media do Ncm(2) = 0.109075 | IC +- 0.003198 | 0.100105 na rodada
-Media da Vazao(2) = 491.684988 | IC +- 10.256270 | 516.437424 na rodada

Cenário 2



Cenário 3

Rodada 1

-Tempo real: 13.44 segundos nesta rodada, 20.22 no total
 -Tempo simulado: 27855459.80 microseg nesta rodada, 41488417.80 total
 -Media uso Ether = 0.624920 | IC +- 0.000000 | 0.624920 na rodada
 -Media do TAp(1) = 503.027716 | IC +- 0.000000 | 503.027716 na rodada
 -Media do TAm(1) = 51695.508646 | IC +- 0.000000 | 51695.508646 na rodada
 -Media do Ncm(1) = 0.480980 | IC +- 0.000000 | 0.480980 na rodada
 -Media da Vazao(1)= 498.286515 | IC +- 0.000000 | 498.286515 na rodada
 -Media do TAp(2) = 12331.822145 | IC +- 0.000000 | 12331.822145 na rodada
 -Media do TAm(2) = 12331.822145 | IC +- 0.000000 | 12331.822145 na rodada
 -Media do Ncm(2) = 2.558367 | IC +- 0.000000 | 2.558367 na rodada
 -Media da Vazao(2)= 61.926818 | IC +- 0.000000 | 61.926818 na rodada
 -Media do TAp(3) = 11717.065719 | IC +- 0.000000 | 11717.065719 na rodada
 -Media do TAm(3) = 11717.065719 | IC +- 0.000000 | 11717.065719 na rodada
 -Media do Ncm(3) = 2.792507 | IC +- 0.000000 | 2.792507 na rodada
 -Media da Vazao(3)= 61.890919 | IC +- 0.000000 | 61.890919 na rodada
 -Media do TAp(4) = 12641.350116 | IC +- 0.000000 | 12641.350116 na rodada
 -Media do TAm(4) = 12641.350116 | IC +- 0.000000 | 12641.350116 na rodada
 -Media do Ncm(4) = 2.496837 | IC +- 0.000000 | 2.496837 na rodada
 -Media da Vazao(4)= 61.747320 | IC +- 0.000000 | 61.747320 na rodada

Rodada 2

-Tempo real: 13.97 segundos nesta rodada, 34.19 no total
 -Tempo simulado: 27343400.90 microseg nesta rodada, 68831818.70 total
 -Media uso Ether = 0.624753 | IC +- 0.002116 | 0.624587 na rodada
 -Media do TAp(1) = 555.185392 | IC +- 662.726114 | 607.343069 na rodada
 -Media do TAm(1) = 53805.871970 | IC +- 26814.708467 | 55916.235294 na rodada
 -Media do Ncm(1) = 0.471519 | IC +- 0.120207 | 0.462059 na rodada
 -Media da Vazao(1)= 498.453869 | IC +- 2.126432 | 498.621223 na rodada
 -Media do TAp(2) = 12150.446311 | IC +- 2304.598486 | 11969.070476 na rodada
 -Media do TAm(2) = 12150.446311 | IC +- 2304.598486 | 11969.070476 na rodada
 -Media do Ncm(2) = 2.697376 | IC +- 1.766278 | 2.836385 na rodada
 -Media da Vazao(2)= 61.683801 | IC +- 3.087824 | 61.440784 na rodada
 -Media do TAp(3) = 11788.066909 | IC +- 902.155647 | 11859.068098 na rodada
 -Media do TAm(3) = 11788.066909 | IC +- 902.155647 | 11859.068098 na rodada
 -Media do Ncm(3) = 2.831319 | IC +- 0.493145 | 2.870130 na rodada
 -Media da Vazao(3)= 61.610994 | IC +- 3.556785 | 61.331069 na rodada
 -Media do TAp(4) = 12438.428494 | IC +- 2578.363676 | 12235.506872 na rodada
 -Media do TAm(4) = 12438.428494 | IC +- 2578.363676 | 12235.506872 na rodada
 -Media do Ncm(4) = 2.635355 | IC +- 1.760032 | 2.773872 na rodada
 -Media da Vazao(4)= 61.740340 | IC +- 0.088694 | 61.733360 na rodada

Rodada 3

-Tempo real: 13.96 segundos nesta rodada, 48.16 no total
 -Tempo simulado: 27338164.30 microseg nesta rodada, 96169983.00 total
 -Media uso Ether = 0.624542 | IC +- 0.000999 | 0.624119 na rodada
 -Media do TAp(1) = 547.072784 | IC +- 134.186360 | 530.847568 na rodada
 -Media do TAm(1) = 53474.367490 | IC +- 5433.007961 | 52811.358529 na rodada
 -Media do Ncm(1) = 0.477974 | IC +- 0.036380 | 0.490882 na rodada
 -Media da Vazao(1)= 498.224474 | IC +- 1.070990 | 497.765682 na rodada
 -Media do TAp(2) = 12216.076964 | IC +- 531.740929 | 12347.338272 na rodada
 -Media do TAm(2) = 12216.076964 | IC +- 531.740929 | 12347.338272 na rodada
 -Media do Ncm(2) = 2.715264 | IC +- 0.353791 | 2.751040 na rodada
 -Media da Vazao(2)= 61.448210 | IC +- 1.179814 | 60.977028 na rodada
 -Media do TAp(3) = 11846.481174 | IC +- 307.048345 | 11963.309704 na rodada
 -Media do TAm(3) = 11846.481174 | IC +- 307.048345 | 11963.309704 na rodada
 -Media do Ncm(3) = 2.800981 | IC +- 0.162278 | 2.740306 na rodada
 -Media da Vazao(3)= 61.680110 | IC +- 0.756293 | 61.818342 na rodada
 -Media do TAp(4) = 12333.397473 | IC +- 676.998086 | 12123.335430 na rodada
 -Media do TAm(4) = 12333.397473 | IC +- 676.998086 | 12123.335430 na rodada
 -Media do Ncm(4) = 2.674939 | IC +- 0.383941 | 2.754108 na rodada
 -Media da Vazao(4)= 61.705376 | IC +- 0.151434 | 61.635448 na rodada

Rodada 4

-Tempo real: 14.65 segundos nesta rodada, 62.82 no total
 -Tempo simulado: 27654814.10 microseg nesta rodada, 123824797.10 total
 -Media uso Ether = 0.624865 | IC +- 0.001155 | 0.625836 na rodada
 -Media do TAp(1) = 535.482751 | IC +- 79.283159 | 500.712653 na rodada
 -Media do TAm(1) = 53006.502143 | IC +- 3207.990833 | 51602.906105 na rodada
 -Media do Ncm(1) = 0.475377 | IC +- 0.020744 | 0.467587 na rodada
 -Media da Vazao(1)= 498.058994 | IC +- 0.768826 | 497.562556 na rodada
 -Media do TAp(2) = 12125.116637 | IC +- 401.421647 | 11852.235656 na rodada

```

-Media do TAm(2) = 12125.116637 | IC +- 401.421647 | 11852.235656 na rodada
-Media do Ncm(2) = 2.726523 | IC +- 0.188474 | 2.760302 na rodada
-Media da Vazao(2) = 61.526507 | IC +- 0.665466 | 61.761399 na rodada
-Media do TAp(3) = 11926.265667 | IC +- 300.431585 | 12165.619148 na rodada
-Media do TAm(3) = 11926.265667 | IC +- 300.431585 | 12165.619148 na rodada
-Media do Ncm(3) = 2.783712 | IC +- 0.101113 | 2.731905 na rodada
-Media da Vazao(3) = 61.745632 | IC +- 0.447146 | 61.942199 na rodada
-Media do TAp(4) = 12386.595098 | IC +- 392.469430 | 12546.187974 na rodada
-Media do TAm(4) = 12386.595098 | IC +- 392.469430 | 12546.187974 na rodada
-Media do Ncm(4) = 2.660229 | IC +- 0.206190 | 2.616097 na rodada
-Media da Vazao(4) = 61.764582 | IC +- 0.204389 | 61.942199 na rodada
Rodada 5
-Tempo real: 16.12 segundos nesta rodada, 78.95 no total
-Tempo simulado: 27871203.70 microseg nesta rodada, 151696000.80 total
-Media uso Ether = 0.624929 | IC +- 0.000800 | 0.625181 na rodada
-Media do TAp(1) = 532.744204 | IC +- 54.114589 | 521.790014 na rodada
-Media do TAm(1) = 52893.634453 | IC +- 2190.419539 | 52442.163689 na rodada
-Media do Ncm(1) = 0.472722 | IC +- 0.015838 | 0.462104 na rodada
-Media da Vazao(1) = 498.041028 | IC +- 0.521944 | 497.969164 na rodada
-Media do TAp(2) = 12161.925150 | IC +- 289.883598 | 12309.159199 na rodada
-Media do TAm(2) = 12161.925150 | IC +- 289.883598 | 12309.159199 na rodada
-Media do Ncm(2) = 2.707498 | IC +- 0.137885 | 2.631397 na rodada
-Media da Vazao(2) = 61.585222 | IC +- 0.478341 | 61.820078 na rodada
-Media do TAp(3) = 12007.556101 | IC +- 303.576802 | 12332.717838 na rodada
-Media do TAm(3) = 12007.556101 | IC +- 303.576802 | 12332.717838 na rodada
-Media do Ncm(3) = 2.757659 | IC +- 0.099504 | 2.653448 na rodada
-Media da Vazao(3) = 61.746170 | IC +- 0.302175 | 61.748320 na rodada
-Media do TAp(4) = 12380.107537 | IC +- 265.832896 | 12354.157292 na rodada
-Media do TAm(4) = 12380.107537 | IC +- 265.832896 | 12354.157292 na rodada
-Media do Ncm(4) = 2.669414 | IC +- 0.141653 | 2.706153 na rodada
-Media da Vazao(4) = 61.811560 | IC +- 0.189975 | 61.999475 na rodada
Rodada 6
-Tempo real: 13.94 segundos nesta rodada, 92.90 no total
-Tempo simulado: 27328483.60 microseg nesta rodada, 179024484.40 total
-Media uso Ether = 0.624568 | IC +- 0.001107 | 0.622763 na rodada
-Media do TAp(1) = 537.576113 | IC +- 42.752345 | 561.735656 na rodada
-Media do TAm(1) = 53022.308789 | IC +- 1688.574797 | 53665.680473 na rodada
-Media do Ncm(1) = 0.473213 | IC +- 0.012039 | 0.475666 na rodada
-Media da Vazao(1) = 497.561028 | IC +- 1.295431 | 495.161027 na rodada
-Media do TAp(2) = 12134.479380 | IC +- 230.216376 | 11997.250533 na rodada
-Media do TAm(2) = 12134.479380 | IC +- 230.216376 | 11997.250533 na rodada
-Media do Ncm(2) = 2.733290 | IC +- 0.123535 | 2.862251 na rodada
-Media da Vazao(2) = 61.627725 | IC +- 0.377751 | 61.840241 na rodada
-Media do TAp(3) = 12026.413625 | IC +- 234.554552 | 12120.701243 na rodada
-Media do TAm(3) = 12026.413625 | IC +- 234.554552 | 12120.701243 na rodada
-Media do Ncm(3) = 2.756920 | IC +- 0.075245 | 2.753224 na rodada
-Media da Vazao(3) = 61.761848 | IC +- 0.231959 | 61.840241 na rodada
-Media do TAp(4) = 12336.438993 | IC +- 230.184722 | 12118.096272 na rodada
-Media do TAm(4) = 12336.438993 | IC +- 230.184722 | 12118.096272 na rodada
-Media do Ncm(4) = 2.684261 | IC +- 0.113682 | 2.758499 na rodada
-Media da Vazao(4) = 61.816340 | IC +- 0.144137 | 61.840241 na rodada
Rodada 7
-Tempo real: 14.25 segundos nesta rodada, 107.16 no total
-Tempo simulado: 27839939.00 microseg nesta rodada, 206864423.40 total
-Media uso Ether = 0.624541 | IC +- 0.000893 | 0.624384 na rodada
-Media do TAp(1) = 535.745489 | IC +- 34.684473 | 524.761751 na rodada
-Media do TAm(1) = 52956.960395 | IC +- 1367.827060 | 52564.870029 na rodada
-Media do Ncm(1) = 0.472068 | IC +- 0.010082 | 0.465202 na rodada
-Media da Vazao(1) = 497.704354 | IC +- 1.099594 | 498.564311 na rodada
-Media do TAp(2) = 12138.481259 | IC +- 185.466405 | 12162.492530 na rodada
-Media do TAm(2) = 12138.481259 | IC +- 185.466405 | 12162.492530 na rodada
-Media do Ncm(2) = 2.718046 | IC +- 0.106152 | 2.626582 na rodada
-Media da Vazao(2) = 61.685647 | IC +- 0.335324 | 62.033182 na rodada
-Media do TAp(3) = 12008.682210 | IC +- 193.621521 | 11902.293717 na rodada
-Media do TAm(3) = 12008.682210 | IC +- 193.621521 | 11902.293717 na rodada
-Media do Ncm(3) = 2.749653 | IC +- 0.063092 | 2.706052 na rodada
-Media da Vazao(3) = 61.759559 | IC +- 0.186694 | 61.745825 na rodada
-Media do TAp(4) = 12325.946845 | IC +- 186.953440 | 12262.993957 na rodada
-Media do TAm(4) = 12325.946845 | IC +- 186.953440 | 12262.993957 na rodada
-Media do Ncm(4) = 2.674460 | IC +- 0.094549 | 2.615650 na rodada
-Media da Vazao(4) = 61.816530 | IC +- 0.115959 | 61.817664 na rodada
Rodada 8
-Tempo real: 13.65 segundos nesta rodada, 120.82 no total
-Tempo simulado: 27924969.90 microseg nesta rodada, 234789393.30 total

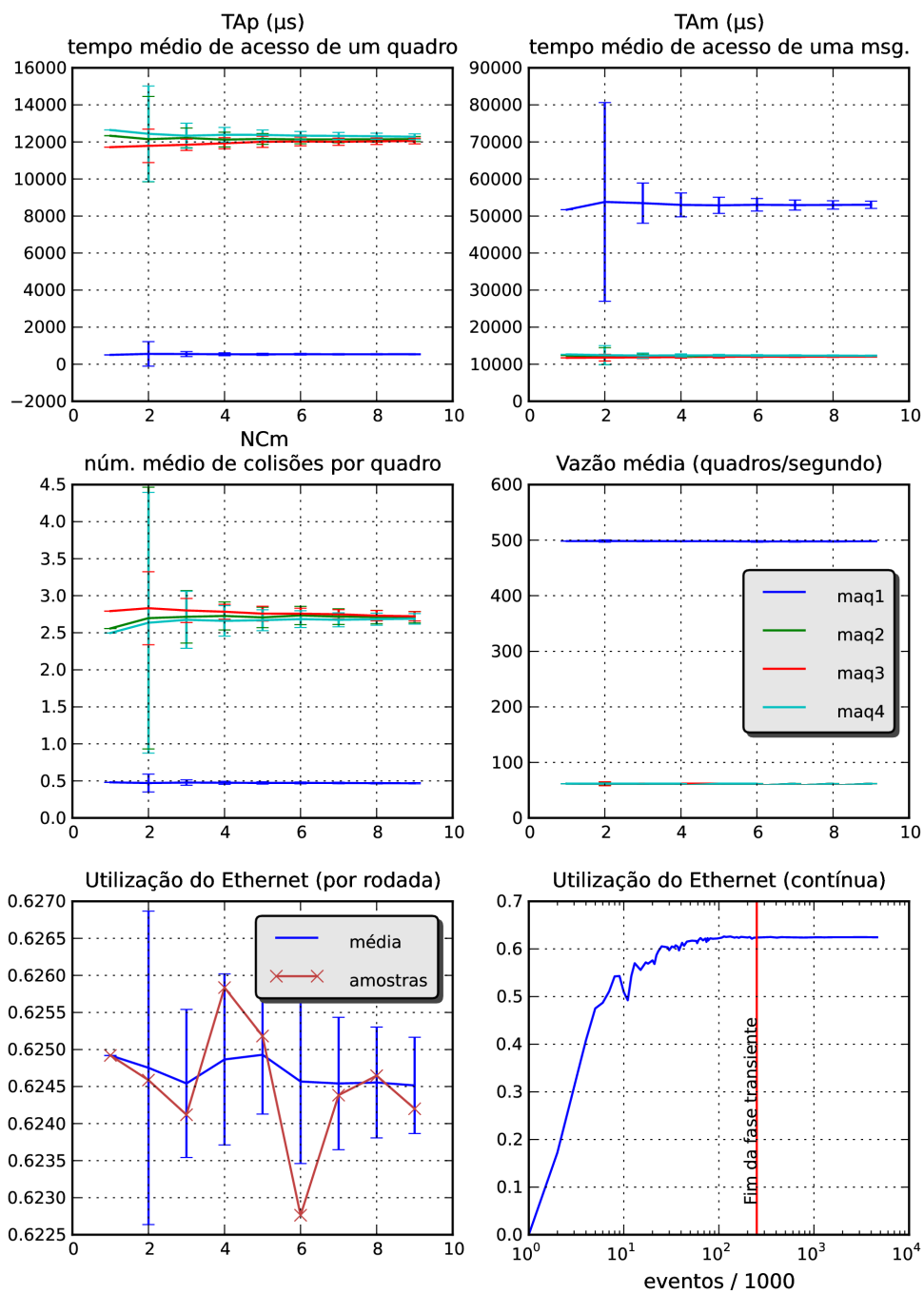
```

```

-Media uso Ether = 0.624555 | IC +- 0.000748 | 0.624646 na rodada
-Media do TAp(1) = 536.354675 | IC +- 29.063215 | 540.618973 na rodada
-Media do TAm(1) = 52987.235209 | IC +- 1146.973071 | 53199.158908 na rodada
-Media do Ncm(1) = 0.470037 | IC +- 0.009709 | 0.455819 na rodada
-Media da Vazao(1)= 497.801135 | IC +- 0.948281 | 498.478604 na rodada
-Media do TAp(2) = 12140.503568 | IC +- 155.290795 | 12154.659734 na rodada
-Media do TAm(2) = 12140.503568 | IC +- 155.290795 | 12154.659734 na rodada
-Media do Ncm(2) = 2.712221 | IC +- 0.089900 | 2.671445 na rodada
-Media da Vazao(2)= 61.723383 | IC +- 0.294477 | 61.987533 na rodada
-Media do TAp(3) = 12034.314559 | IC +- 173.006780 | 12213.741006 na rodada
-Media do TAm(3) = 12034.314559 | IC +- 173.006780 | 12213.741006 na rodada
-Media do Ncm(3) = 2.730344 | IC +- 0.069804 | 2.595183 na rodada
-Media da Vazao(3)= 61.779103 | IC +- 0.162936 | 61.915913 na rodada
-Media do TAp(4) = 12306.251226 | IC +- 163.246036 | 12168.381892 na rodada
-Media do TAm(4) = 12306.251226 | IC +- 163.246036 | 12168.381892 na rodada
-Media do Ncm(4) = 2.682534 | IC +- 0.081399 | 2.739055 na rodada
-Media da Vazao(4)= 61.802095 | IC +- 0.102874 | 61.701051 na rodada
Rodada 9
-Tempo real: 13.69 segundos nesta rodada, 134.52 no total
-Tempo simulado: 27771128.60 microseg nesta rodada, 262560521.90 total
-Media uso Ether = 0.624515 | IC +- 0.000650 | 0.624200 na rodada
-Media do TAp(1) = 537.364387 | IC +- 25.104146 | 545.442088 na rodada
-Media do TAm(1) = 53032.218355 | IC +- 991.897702 | 53392.083526 na rodada
-Media do Ncm(1) = 0.468991 | IC +- 0.008692 | 0.460621 na rodada
-Media da Vazao(1)= 497.863153 | IC +- 0.828017 | 498.359292 na rodada
-Media do TAp(2) = 12155.072146 | IC +- 137.718921 | 12271.620766 na rodada
-Media do TAm(2) = 12155.072146 | IC +- 137.718921 | 12271.620766 na rodada
-Media do Ncm(2) = 2.710664 | IC +- 0.077403 | 2.698211 na rodada
-Media da Vazao(2)= 61.762880 | IC +- 0.269146 | 62.078860 na rodada
-Media do TAp(3) = 12050.411634 | IC +- 153.355440 | 12179.188228 na rodada
-Media do TAm(3) = 12050.411634 | IC +- 153.355440 | 12179.188228 na rodada
-Media do Ncm(3) = 2.722435 | IC +- 0.062745 | 2.659157 na rodada
-Media da Vazao(3)= 61.780402 | IC +- 0.140166 | 61.790791 na rodada
-Media do TAp(4) = 12280.480682 | IC +- 152.459302 | 12074.316336 na rodada
-Media do TAm(4) = 12280.480682 | IC +- 152.459302 | 12074.316336 na rodada
-Media do Ncm(4) = 2.688730 | IC +- 0.071451 | 2.738302 na rodada
-Media da Vazao(4)= 61.792837 | IC +- 0.091016 | 61.718774 na rodada

```

Cenário 3



Cenário 4

Rodada 1

-Tempo real: 13.86 segundos nesta rodada, 81.19 no total
-Tempo simulado: 29395749.30 microseg nesta rodada, 175747644.60 total
-Media uso Ether = 0.619143 | IC +- 0.000000 | 0.619143 na rodada
-Media do TAp(1) = 556.099737 | IC +- 0.000000 | 556.099737 na rodada
-Media do TAm(1) = 53818.389489 | IC +- 0.000000 | 53818.389489 na rodada
-Media do Ncm(1) = 0.539549 | IC +- 0.000000 | 0.539549 na rodada
-Media da Vazao(1)= 498.031190 | IC +- 0.000000 | 498.031190 na rodada
-Media do TAp(2) = 8279.619619 | IC +- 0.000000 | 8279.619619 na rodada
-Media do TAm(2) = 8279.619619 | IC +- 0.000000 | 8279.619619 na rodada
-Media do Ncm(2) = 2.092249 | IC +- 0.000000 | 2.092249 na rodada
-Media da Vazao(2)= 62.015769 | IC +- 0.000000 | 62.015769 na rodada
-Media do TAp(3) = 8877.009163 | IC +- 0.000000 | 8877.009163 na rodada
-Media do TAm(3) = 8877.009163 | IC +- 0.000000 | 8877.009163 na rodada
-Media do Ncm(3) = 2.094267 | IC +- 0.000000 | 2.094267 na rodada
-Media da Vazao(3)= 61.403436 | IC +- 0.000000 | 61.403436 na rodada
-Media do TAp(4) = 8996.631295 | IC +- 0.000000 | 8996.631295 na rodada
-Media do TAm(4) = 8996.631295 | IC +- 0.000000 | 8996.631295 na rodada
-Media do Ncm(4) = 2.057359 | IC +- 0.000000 | 2.057359 na rodada
-Media da Vazao(4)= 62.389973 | IC +- 0.000000 | 62.389973 na rodada

Rodada 2

-Tempo real: 14.05 segundos nesta rodada, 95.24 no total
-Tempo simulado: 29220553.00 microseg nesta rodada, 204968197.60 total
-Media uso Ether = 0.618681 | IC +- 0.005880 | 0.618218 na rodada
-Media do TAp(1) = 554.784269 | IC +- 16.714613 | 553.468800 na rodada
-Media do TAm(1) = 53768.978550 | IC +- 627.825511 | 53719.567611 na rodada
-Media do Ncm(1) = 0.550029 | IC +- 0.133155 | 0.560508 na rodada
-Media da Vazao(1)= 498.275080 | IC +- 3.098920 | 498.518971 na rodada
-Media do TAp(2) = 8580.872063 | IC +- 3827.775233 | 8882.124507 na rodada
-Media do TAm(2) = 8580.872063 | IC +- 3827.775233 | 8882.124507 na rodada
-Media do Ncm(2) = 2.065840 | IC +- 0.335561 | 2.039430 na rodada
-Media da Vazao(2)= 61.996347 | IC +- 0.246785 | 61.976924 na rodada
-Media do TAp(3) = 8603.925307 | IC +- 3469.859382 | 8330.841451 na rodada
-Media do TAm(3) = 8603.925307 | IC +- 3469.859382 | 8330.841451 na rodada
-Media do Ncm(3) = 2.125074 | IC +- 0.391439 | 2.155881 na rodada
-Media da Vazao(3)= 60.954396 | IC +- 5.705584 | 60.505357 na rodada
-Media do TAp(4) = 8982.435546 | IC +- 180.374083 | 8968.239798 na rodada
-Media do TAm(4) = 8982.435546 | IC +- 180.374083 | 8968.239798 na rodada
-Media do Ncm(4) = 2.070093 | IC +- 0.161800 | 2.082827 na rodada
-Media da Vazao(4)= 61.961002 | IC +- 5.450586 | 61.532032 na rodada

Rodada 3

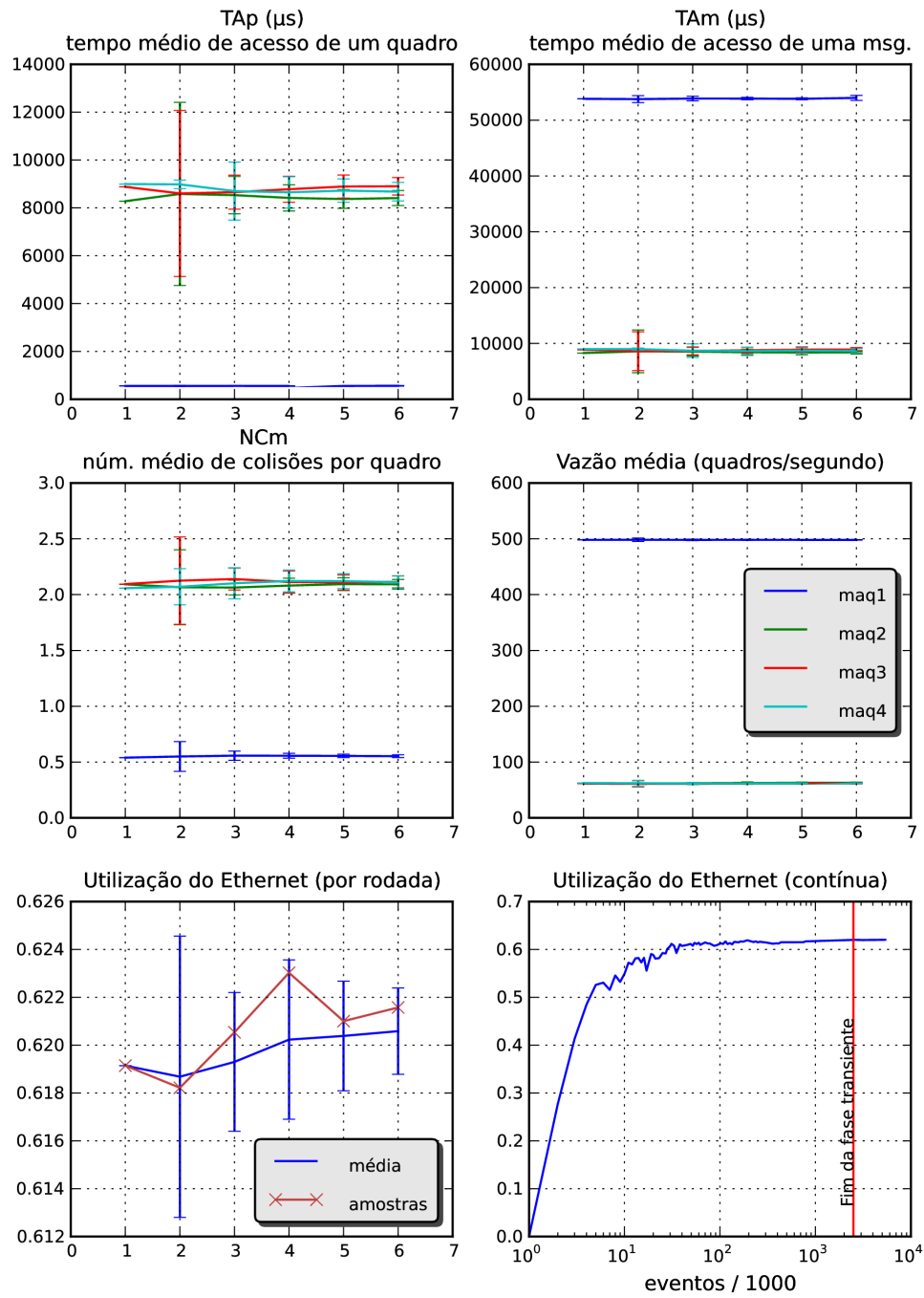
-Tempo real: 14.23 segundos nesta rodada, 109.48 no total
-Tempo simulado: 28913874.30 microseg nesta rodada, 233882071.90 total
-Media uso Ether = 0.619299 | IC +- 0.002900 | 0.620537 na rodada
-Media do TAp(1) = 557.395309 | IC +- 11.700013 | 562.617390 na rodada
-Media do TAm(1) = 53862.627595 | IC +- 421.219754 | 54049.925686 na rodada
-Media do Ncm(1) = 0.557470 | IC +- 0.041266 | 0.572354 na rodada
-Media da Vazao(1)= 497.997670 | IC +- 1.338561 | 497.442849 na rodada
-Media do TAp(2) = 8530.833184 | IC +- 778.707650 | 8430.755427 na rodada
-Media do TAm(2) = 8530.833184 | IC +- 778.707650 | 8430.755427 na rodada
-Media do Ncm(2) = 2.063177 | IC +- 0.066597 | 2.057851 na rodada
-Media da Vazao(2)= 62.162881 | IC +- 0.718160 | 62.495949 na rodada
-Media do TAp(3) = 8655.236357 | IC +- 713.398610 | 8757.858456 na rodada
-Media do TAm(3) = 8655.236357 | IC +- 713.398610 | 8757.858456 na rodada
-Media do Ncm(3) = 2.139499 | IC +- 0.098535 | 2.168350 na rodada
-Media da Vazao(3)= 61.110864 | IC +- 1.302888 | 61.423799 na rodada
-Media do TAp(4) = 8700.912487 | IC +- 1211.809169 | 8137.866370 na rodada
-Media do TAm(4) = 8700.912487 | IC +- 1211.809169 | 8137.866370 na rodada
-Media do Ncm(4) = 2.101593 | IC +- 0.139175 | 2.164593 na rodada
-Media da Vazao(4)= 61.655121 | IC +- 1.693420 | 61.043359 na rodada

Rodada 4

-Tempo real: 14.68 segundos nesta rodada, 124.18 no total
-Tempo simulado: 28866162.10 microseg nesta rodada, 262748234.00 total
-Media uso Ether = 0.620231 | IC +- 0.003331 | 0.623027 na rodada
-Media do TAp(1) = 557.067661 | IC +- 6.207426 | 556.084715 na rodada
-Media do TAm(1) = 53859.459110 | IC +- 220.532690 | 53849.953655 na rodada
-Media do Ncm(1) = 0.556473 | IC +- 0.021815 | 0.553482 na rodada
-Media da Vazao(1)= 498.116481 | IC +- 0.795664 | 498.472916 na rodada
-Media do TAp(2) = 8416.201666 | IC +- 546.768315 | 8072.307111 na rodada

-Media do TAm(2) = 8416.201666 | IC +- 546.768315 | 8072.307111 na rodada
 -Media do Ncm(2) = 2.080859 | IC +- 0.066179 | 2.133905 na rodada
 -Media da Vazao(2) = 62.730987 | IC +- 1.846572 | 64.435306 na rodada
 -Media do TAp(3) = 8778.968040 | IC +- 542.465439 | 9150.163091 na rodada
 -Media do TAm(3) = 8778.968040 | IC +- 542.465439 | 9150.163091 na rodada
 -Media do Ncm(3) = 2.112912 | IC +- 0.099072 | 2.033149 na rodada
 -Media da Vazao(3) = 61.405014 | IC +- 1.157864 | 62.287463 na rodada
 -Media do TAp(4) = 8648.083657 | IC +- 655.708199 | 8489.597164 na rodada
 -Media do TAm(4) = 8648.083657 | IC +- 655.708199 | 8489.597164 na rodada
 -Media do Ncm(4) = 2.122181 | IC +- 0.097936 | 2.183946 na rodada
 -Media da Vazao(4) = 61.700618 | IC +- 0.897432 | 61.837109 na rodada
 Rodada 5
 -Tempo real: 14.41 segundos nesta rodada, 138.60 no total
 -Tempo simulado: 29000929.40 microseg nesta rodada, 291749163.40 total
 -Media uso Ether = 0.620386 | IC +- 0.002292 | 0.621004 na rodada
 -Media do TAp(1) = 556.421129 | IC +- 4.562774 | 553.835003 na rodada
 -Media do TAm(1) = 53821.357322 | IC +- 182.760055 | 53668.950172 na rodada
 -Media do Ncm(1) = 0.555206 | IC +- 0.015156 | 0.550139 na rodada
 -Media da Vazao(1) = 498.041719 | IC +- 0.576367 | 497.742669 na rodada
 -Media do TAp(2) = 8369.393719 | IC +- 391.682364 | 8182.161932 na rodada
 -Media do TAm(2) = 8369.393719 | IC +- 391.682364 | 8182.161932 na rodada
 -Media do Ncm(2) = 2.093892 | IC +- 0.057529 | 2.146025 na rodada
 -Media da Vazao(2) = 62.867142 | IC +- 1.303874 | 63.411761 na rodada
 -Media do TAp(3) = 8891.409639 | IC +- 481.504404 | 9341.176032 na rodada
 -Media do TAm(3) = 8891.409639 | IC +- 481.504404 | 9341.176032 na rodada
 -Media do Ncm(3) = 2.108070 | IC +- 0.068286 | 2.088705 na rodada
 -Media da Vazao(3) = 61.544303 | IC +- 0.872811 | 62.101458 na rodada
 -Media do TAp(4) = 8720.916658 | IC +- 487.073459 | 9012.248662 na rodada
 -Media do TAm(4) = 8720.916658 | IC +- 487.073459 | 9012.248662 na rodada
 -Media do Ncm(4) = 2.120627 | IC +- 0.066323 | 2.114411 na rodada
 -Media da Vazao(4) = 61.822164 | IC +- 0.694033 | 62.308348 na rodada
 Rodada 6
 -Tempo real: 14.61 segundos nesta rodada, 153.22 no total
 -Tempo simulado: 29099523.20 microseg nesta rodada, 320848686.60 total
 -Media uso Ether = 0.620584 | IC +- 0.001806 | 0.621574 na rodada
 -Media do TAp(1) = 560.968882 | IC +- 12.188607 | 583.707645 na rodada
 -Media do TAm(1) = 53988.268798 | IC +- 450.754869 | 54822.826176 na rodada
 -Media do Ncm(1) = 0.552773 | IC +- 0.013053 | 0.540608 na rodada
 -Media da Vazao(1) = 498.020089 | IC +- 0.439242 | 497.911938 na rodada
 -Media do TAp(2) = 8409.478800 | IC +- 313.511978 | 8609.904202 na rodada
 -Media do TAm(2) = 8409.478800 | IC +- 313.511978 | 8609.904202 na rodada
 -Media do Ncm(2) = 2.091908 | IC +- 0.043787 | 2.081985 na rodada
 -Media da Vazao(2) = 62.962196 | IC +- 1.015506 | 63.437466 na rodada
 -Media do TAp(3) = 8901.698090 | IC +- 364.955849 | 8953.140350 na rodada
 -Media do TAm(3) = 8901.698090 | IC +- 364.955849 | 8953.140350 na rodada
 -Media do Ncm(3) = 2.114664 | IC +- 0.054333 | 2.147629 na rodada
 -Media da Vazao(3) = 61.859830 | IC +- 1.045567 | 63.437466 na rodada
 -Media do TAp(4) = 8677.256193 | IC +- 384.931238 | 8458.953869 na rodada
 -Media do TAm(4) = 8677.256193 | IC +- 384.931238 | 8458.953869 na rodada
 -Media do Ncm(4) = 2.111287 | IC +- 0.055590 | 2.064587 na rodada
 -Media da Vazao(4) = 61.925284 | IC +- 0.587821 | 62.440886 na rodada

Cenário 4



Comentários

Pode-se perceber claramente, comparando os resultados do cenário 1 com o cenário 2, a desvantagem causada pelas chegadas sincronizadas de mensagens. O TAp e TAm no cenário 1 são razoavelmente maiores que no cenário 2, devido às colisões inevitáveis no começo da transmissão de cada mensagem, que atrasam o envio do primeiro quadro de cada mensagem e também da mensagem inteira. O NCm também foi afetado devido ao maior número de colisões.

A vazão por sua vez deve se manter aproximadamente constante e independente do funcionamento do sistema, a menos de alguns quadros eventualmente perdidos, desde que o sistema seja estável. Se a vazão nominal for muito maior do que a vazão medida na simulação, significa que o sistema não consegue processar as chegadas tão rápido quanto elas acontecem, causando instabilidade no sistema e estouro das filas de espera. Neste caso, a vazão nominal é de 4Mbps, ou seja, $4\,000\,000/8\,000 = 500$ quadros por segundo, que é pouco maior que a medida, donde pode-se concluir que o sistema é razoavelmente estável.

Por último, o valor esperado para a utilização ethernet é algo pouco superior a 0,8, já que existem duas estações transmitindo a 4Mbps, num total de 8Mbps, que corresponde a 0.8 de utilização em uma rede de 10Mbps. É esperado que este valor seja um pouco maior devido aos *overheads* causados por eventuais colisões. Pode-se observar também que o *overhead* é maior no cenário 1, novamente, devido às colisões forçadas pelo sincronismo de chegada.

Um fenômeno similar ao evidenciado nos cenários 1 e 2 quanto aos tempos de acesso pode ser observado se compararmos os cenários 3 e 4. A chegada de mensagens sincronizadas nas estações 2, 3 e 4 faz aumentar o número médio de colisões (NCm) e aumentar notavelmente o TAp e TAm dessas máquinas. Observa-se, também, um ligeiro aumento na utilização do Ethernet devido ao *overhead* causado pelas colisões. A vazão média, no entanto, não sofreu variação pelo motivo citado anteriormente.

Outro fenômeno notável nos cenários 3 e 4 é a diferença entre os valores de TAm e TAp da máquina 1 para as outras. No caso do TAm, o valor da máquina 1 é muito maior que os outros pois suas mensagens são muito maiores. Por outro lado, o TAp é menor porque a maioria dos quadros das mensagens da máquina 1 encontra o meio livre (já que estes são enviados com maior frequência do que os quadros das outras máquinas).

Finalmente, é evidente que o overhead do Ethernet é bem maior nos cenários 3 e 4 (~0.8Mbps para quatro máquinas) comparado aos 1 e 2 (~0.2Mbps para duas máquinas). Isto é um fato (e desvantagem) conhecido do protocolo CSMA/CD: o overhead do protocolo aumenta exponencialmente com o número de estações. Por isso não se costuma utilizar uma rede muito grande apenas com hubs Ethernet, sua utilização sendo em geral trocada por um switch, que é mais “inteligente” pois trabalha uma camada acima na rede, repassando quadros seletivamente de acordo com o histórico de comunicações observado no ambiente.

Conclusões

Antes do início da implementação, era necessário decidir qual linguagem seria utilizada na construção do simulador. Houve um breve debate se deveria ser escolhida uma linguagem com a qual os membros do grupo estivessem mais familiarizados (C/C++ ou Java) ou se deveria ser escolhida uma linguagem mais moderna (como Python) com mais facilidades. Dentre as facilidades, além das inerentes à linguagem (não precisa compilar, sintaxe mais simples, código mais claro, linguagem menos verbosa), também era possível contar com bibliotecas que auxiliassem no cálculo das variáveis aleatórias e na geração dos gráficos. Acabou-se por decidir por Python, pois, apesar de menos familiarizados, analisou-se que o esforço de se ambientar com a nova linguagem seria menor do que ter de fazer os cálculos e gerar os gráficos nas outras linguagens. Posteriormente, constatou-se que foi uma decisão bem tomada, já que poucas dificuldades foram encontradas durante toda a fase de implementação.

Após a implementação, foram executados os cenários sugeridos pelo professor. Ao rodar as simulações, foi possível obter uma compreensão melhor do funcionamento de uma rede Ethernet. Analisar como o controle de colisão é feito, e as soluções para garantir o envio das mensagens foi bem surpreendente.

Após realizar os testes de correção desligando o controle de colisões, conseguimos perceber que era suficiente para resolver o problema e garantir a troca de mensagens e, para redes pequenas, o *overhead* era baixo o suficiente para não onerar demais a rede. No entanto, em simulações feitas com um grande número de máquinas, pode-se perceber que o *overhead* torna-se cada vez maior, fazendo com que o desempenho da rede caia e a taxa de utilização individual de cada máquina passa a ser baixo, mesmo que a utilização total da rede mantenha-se alta. Mas é importante entender que, por mais oneroso que seja, o controle de colisões é essencial para o funcionamento da rede. Sem ele, a rede provavelmente ficaria inundada de mensagens sobrepostas, gerando um ruído imenso e nenhuma garantia que as outras máquinas conseguiram receber o que foi trafegado.