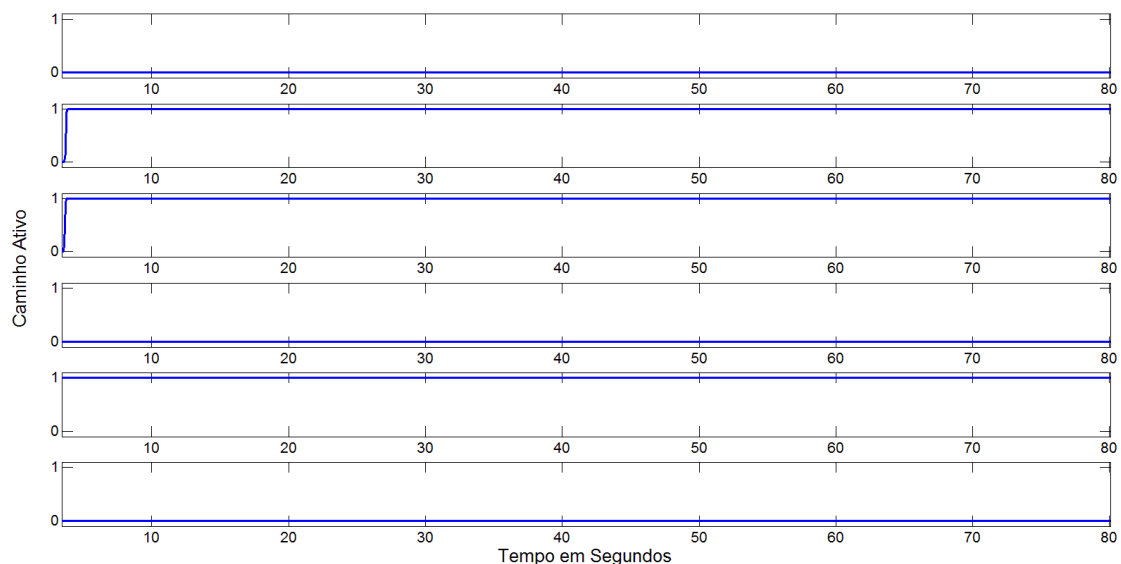


## Testes Multi-Agente:

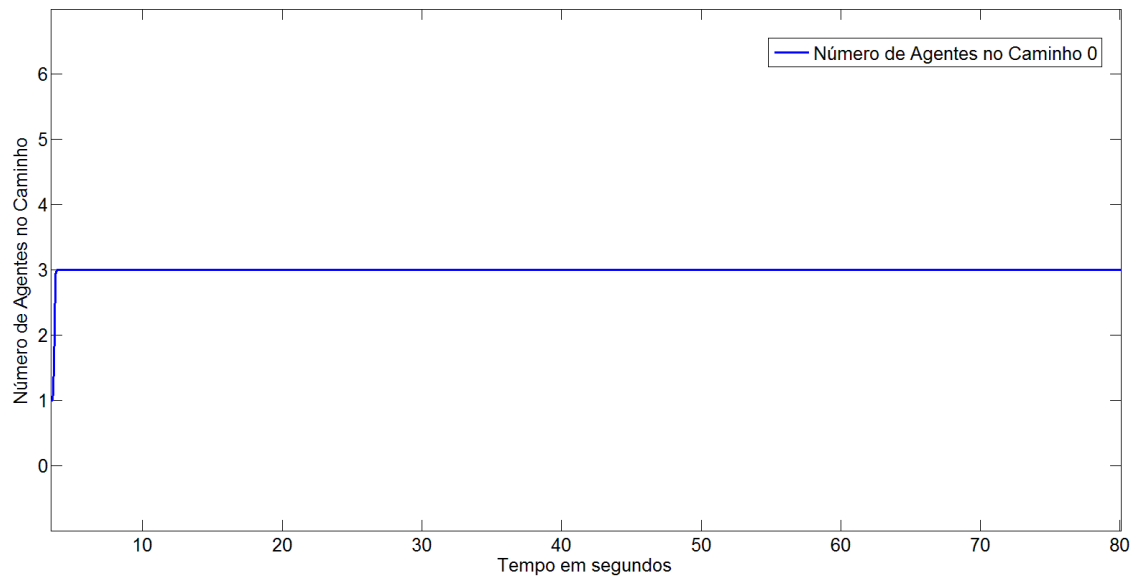
O algoritmo de seleção inteligente de caminho pode melhorar muito a eficiência de uma comunicação *VoIP (Voice over IP)*, por exemplo, uma vez que ele utiliza estratégias para trocar o caminho principal para um de maior qualidade. Porém, é desejável determinar o comportamento de cada estratégia de seleção de caminho para quando múltiplos agentes (ou programas) desejam comunicar-se simultaneamente, dividindo a ocupação entre os caminhos.

Para os testes iniciais, foram utilizados 6 agentes *UDP (User Datagram Protocol) CBR (Constant Bit Rate)* comunicando simultaneamente com o algoritmo de seleção de caminho pelo menor *SRTT (Smoothed Round Trip Time)* ativo, com histerese de 10 ms. Foram realizados testes para várias utilizações de banda diferentes ( $\rho$ ), porque em trabalhos anteriores [1] foi comprovada a instabilidade do sistema multi abrigado, que não consegue distribuir os agentes igualmente entre os caminhos para um  $\rho$  muito elevado (acima de 0.7).

Entende-se como um comportamento adequado uma estabilidade entre os agentes, onde não há muitas trocas de caminho, eles se distribuem entre 3 agentes por um caminho e 3 por outro e o atraso médio entre os agentes seja o mínimo possível. O Gráfico 1 mostra as trocas de caminho por agente e o Gráfico 2 mostra o gráfico entre o número de agentes disputando um mesmo caminho, ambos em uma situação estável, com um  $\rho$  baixo.

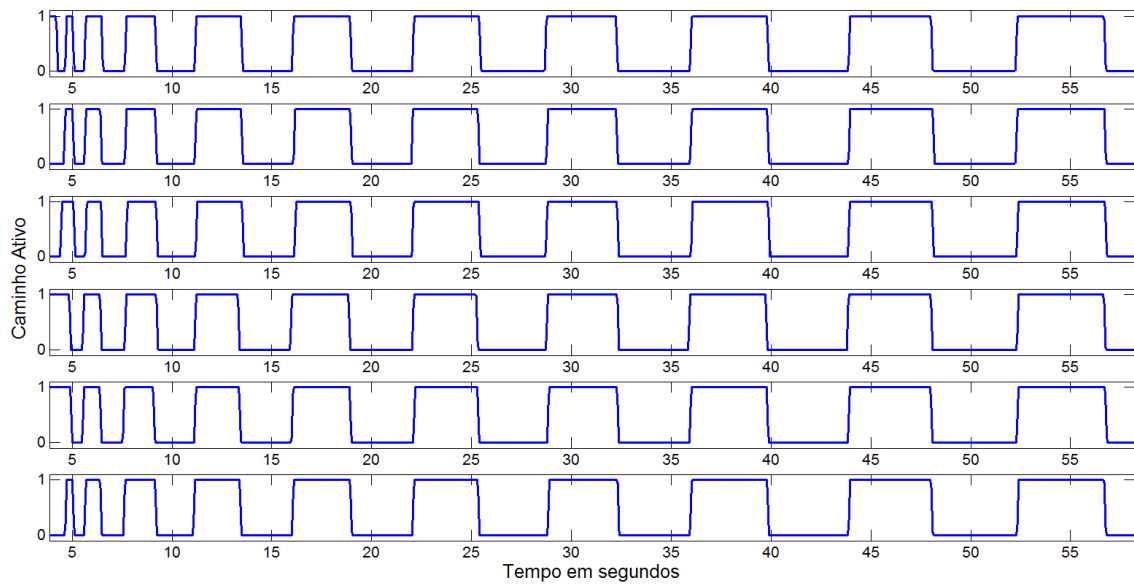


**Gráfico 1: Caminhos Ativos de cada agente ( $\rho=0.72$ )**

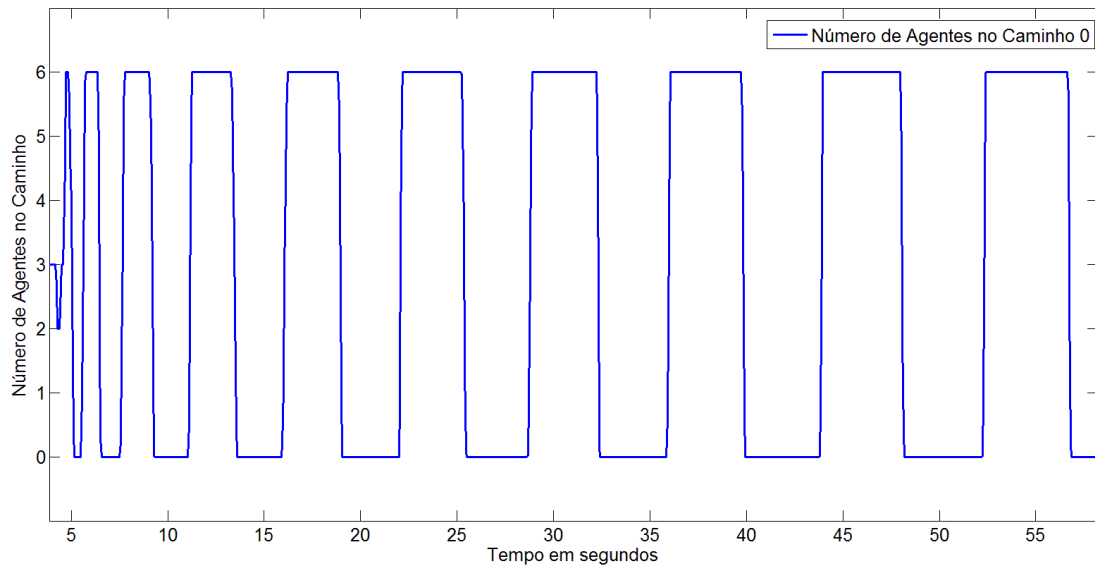


**Gráfico 2: Soma do número de agentes em um caminho ( $\rho=0.72$ )**

Os Gráficos 3 e 4 representam os mesmos dados que os Gráficos 1 e 2, porém em uma situação de instabilidade, com um  $\rho$  muito elevado.



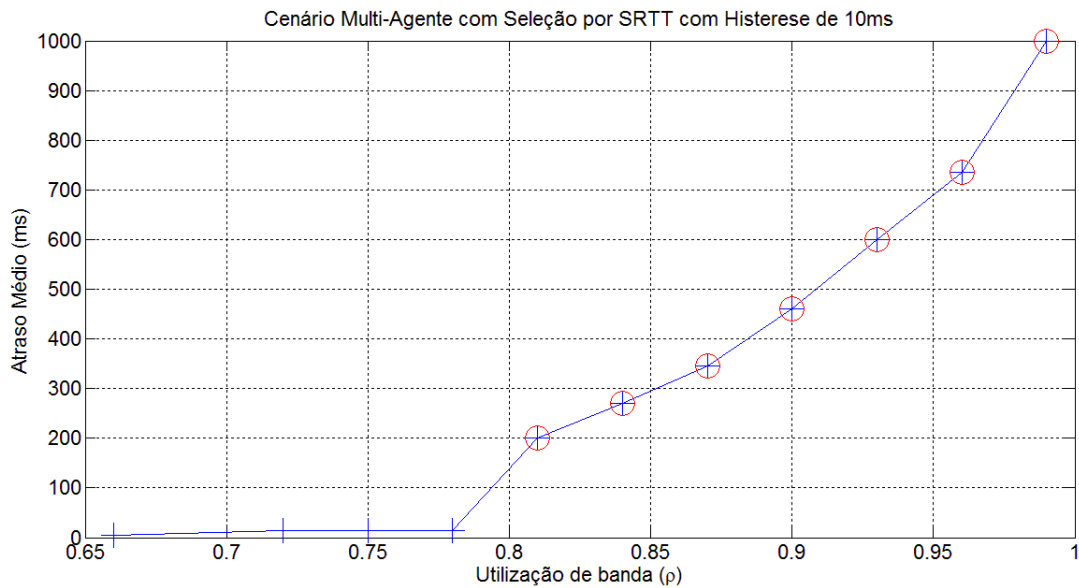
**Gráfico 3: Caminhos Ativos de cada agente ( $\rho=0.99$ )**



**Gráfico 4: Soma do número de agentes em um caminho ( $p=0.99$ )**

Este alto índice de utilização das bandas força um sincronismo entre os agentes, onde todos definem o mesmo caminho ativo, e não se distribuem igualmente entre os caminhos disponíveis. Para tentar evitar o sincronismo, os agentes se iniciam em tempos diferentes, entre 0 e 10s de diferença, seguindo uma distribuição uniforme. Apesar de esta medida não ter evitado o sincronismo, foi optado por continuar os testes desta forma, pois representa uma situação real, onde os agentes podem iniciar em qualquer tempo aleatório.

A instabilidade do sistema ocasiona em um aumento da latência média, devido ao preenchimento das filas dos buffers de apenas um caminho, conforme verificado no Gráfico 5.

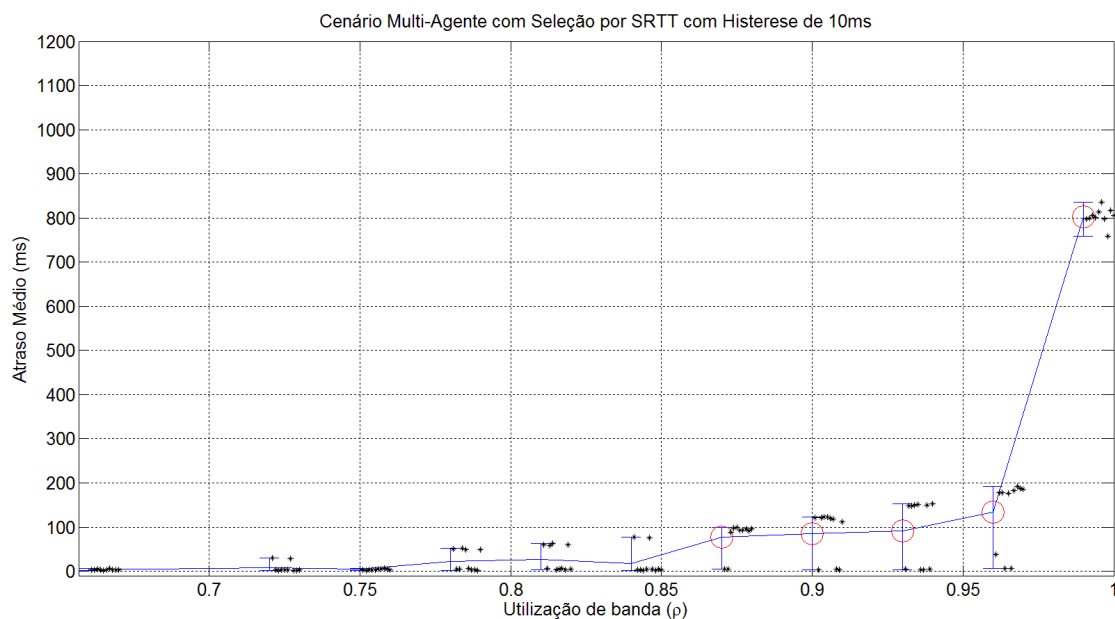


**Gráfico 5: Atraso Médio em função da Utilização de Banda (Histerese de 10ms e Pacotes de 1250B)**

Os círculos em vermelho no Gráfico 5 representam que para a respectiva utilização de banda, o sistema não estabilizou. Desta forma, percebe-se que o atraso aumenta consideravelmente quando o sistema está instável, e é ainda proporcional ao  $\rho$ . Para este teste, foram utilizados pacotes de 1250 bytes.

O objetivo então é analisar diversos aspectos da comunicação multi-caminho, a fim de diminuir ou eliminar esta instabilidade do sistema, seja alterando o algoritmo de seleção de caminho, mudando histereses ou adicionando parâmetros que possam auxiliar nesta tarefa.

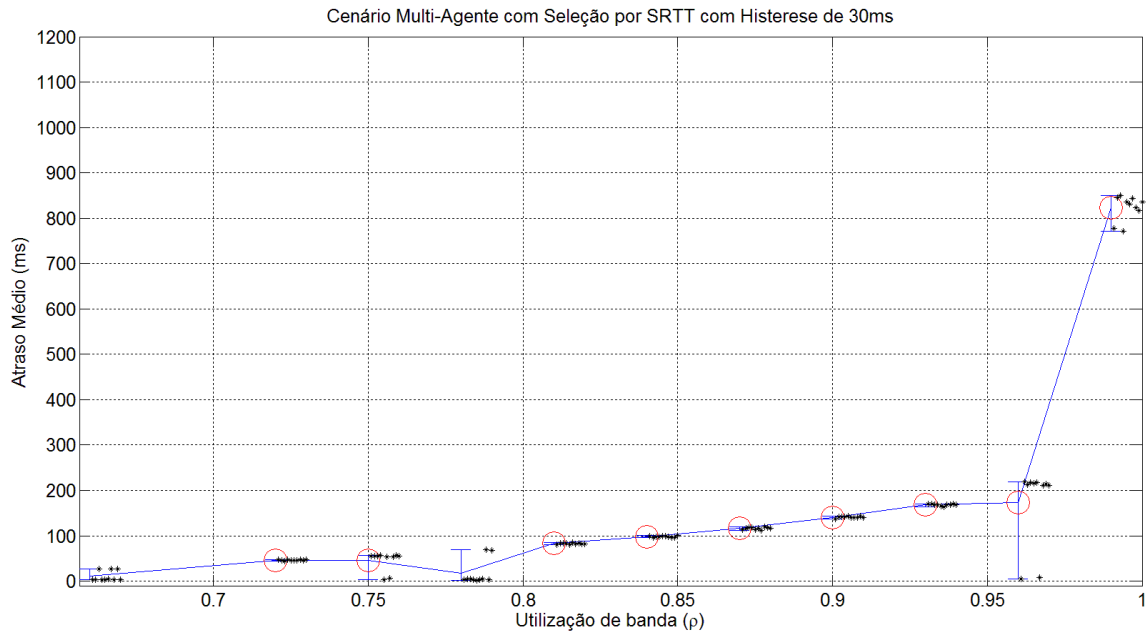
No próximo teste, o tamanho do pacote foi diminuído para 250 Bytes, a fim de representar um cenário mais parecido com o VoIP. Para ilustrar os valores máximos e mínimos obtidos, o Gráfico 6 foi feito com uma barra de erro para mostrar a variação para cada  $\rho$ .



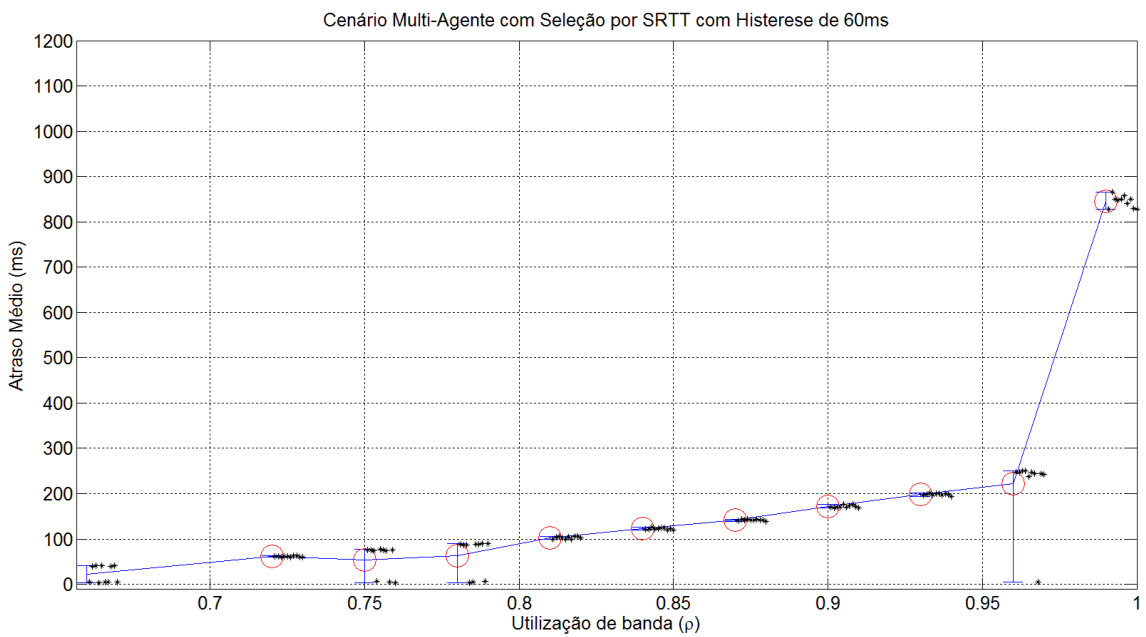
**Gráfico 6: Atraso Médio em função de  $\rho$  (Histerese de 10ms e Pacotes de 250B)**

Observa-se uma melhora de desempenho do sistema para pacotes menores. A variância dos dados também aumentou, uma vez que o sistema estabilizou-se algumas vezes em uma utilização de banda um pouco maior, como é o caso do  $\rho = 0.86, 0.90, 0.93$  e  $0.96$ . Porém, só foram considerados estáveis os sistemas onde mais da metade dos testes esta condição foi apresentada.

Em seguida, procurou-se saber o efeito da mudança da histerese da seleção de caminho sobre a estabilidade do sistema. Para tanto, foram feitos testes com a histerese em 30ms e 60ms. Os resultados estão apresentados nos Gráficos 7 e 8.



**Gráfico 7: Atraso Médio em função de  $\rho$  (Histerese de 30ms e Pacotes de 250B)**

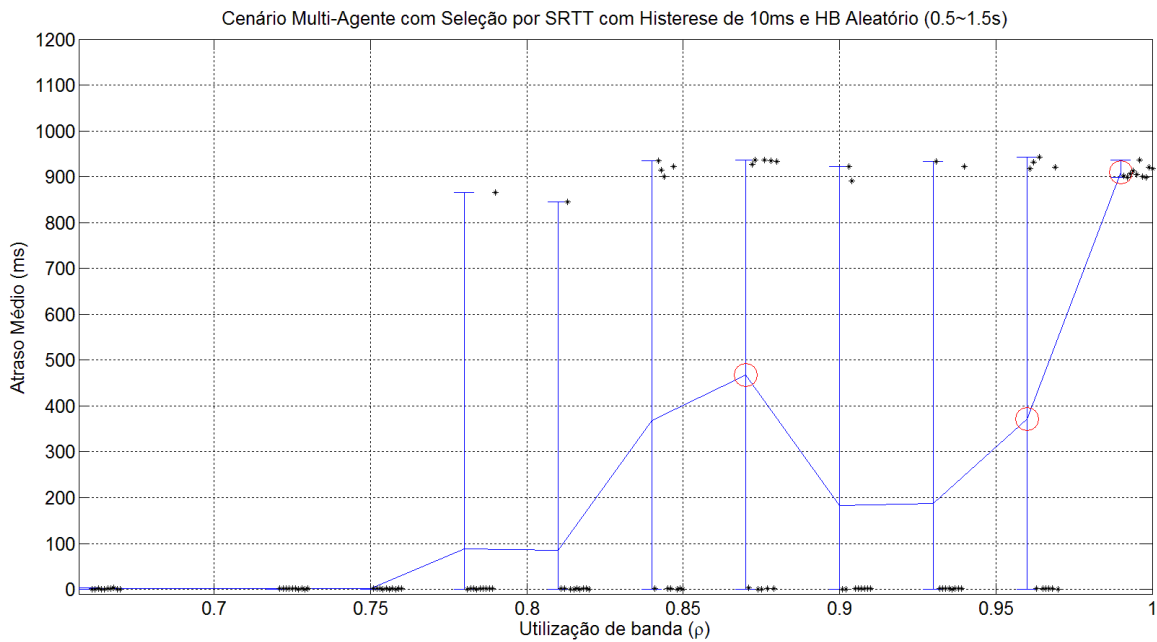


**Gráfico 8: Atraso Médio em função de  $\rho$  (Histerese de 60ms e Pacotes de 250B)**

Mostrou-se que a histerese mais interessante para a estabilidade do sistema é a de 10ms, e que quanto maior a histerese, maior a instabilidade, ou seja, o sistema parece seguir um comportamento linear.

Mais uma medida adotada para tentar evitar o sincronismo entre os agentes foi de mudar o comportamento do HB (*Heartbeat*). Inicialmente, este

estava sendo enviado de forma constante, sempre com o mesmo intervalo entre um pacote e outro. No próximo teste, o intervalo de *Heartbeat* foi configurado inicialmente para um valor aleatório uniforme, entre 0.5s e 1.5s. Este valor foi sorteado apenas uma vez no início da execução do agente, portanto todos os *Heartbeats* são espaçados deste valor. O Gráfico 9 representa o experimento. Como os resultados anteriores foram mais vantajosos com a histerese de 10ms e pacotes de 250 bytes, estes parâmetros foram repetidos.



**Gráfico 9: Atraso Médio em função de  $\rho$  com HB Aleatório**

Do Gráfico 8 para o Gráfico 9, houve uma mudança na forma do envio de HB's. Anteriormente, era enviado um Heartbeat a cada 5 pacotes de dados e o *Heartbeat* era levado em conta no cálculo da banda a ser enviada. A partir do Gráfico 9, o *Heartbeat* passou a ser temporal, com intervalo médio entre eles de 1s na maioria dos casos. Contudo, ele parou de entrar no cálculo da banda a ser enviada, por representar uma porcentagem muito pequena da banda total.

Foi definido o tamanho do pacote de HB como 55 bytes, sendo que apenas 13 são de *payload*. Como são enviados em média 1 Pacote/segundo, temos a banda usada pelo HB:

$$Banda = \frac{55 * 8}{1} = 440 \text{ bits/s}$$

Como a banda mínima enviada pelos agentes é de 110kbps, a porcentagem máxima ocupada pelo HB seria de:

$$\% \text{ de Banda} = \frac{440}{440 + 110000} \cong 0.40\%$$

O que representa como o *Heartbeat* quase não influencia na banda final enviada.

Esta mudança na forma de envio de HB's também é responsável pelo aumento da variabilidade do atraso médio neste caso. Isto acontece pelo fato de que os *Heartbeats* neste teste foram muito menos frequentes do que nos testes anteriores, tornando o algoritmo não muito suscetível à mudanças de caminho, e portanto demorando a detectar a eficiência dos caminhos, aumentando o atraso médio. Entretanto, na maioria dos testes, houve maior estabilidade do que nos testes anteriores. O Gráfico 9 então, pode fornecer então uma impressão um pouco equivocada sobre o experimento, uma vez que nos testes onde o atraso foi perto de 900ms, elevou muito o atraso médio dos 10 testes. O Gráfico 10 foi realizado com os mesmos dados que o Gráfico 9, com a mudança de que neste foi desconsiderado o teste cujo atraso foi o mais elevado.

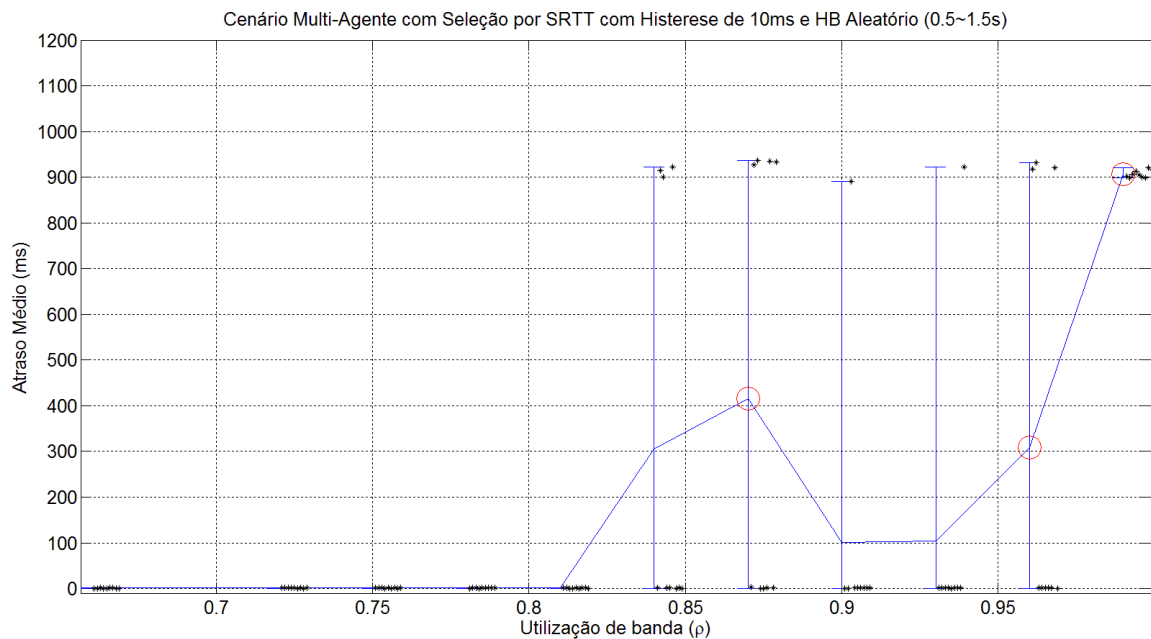
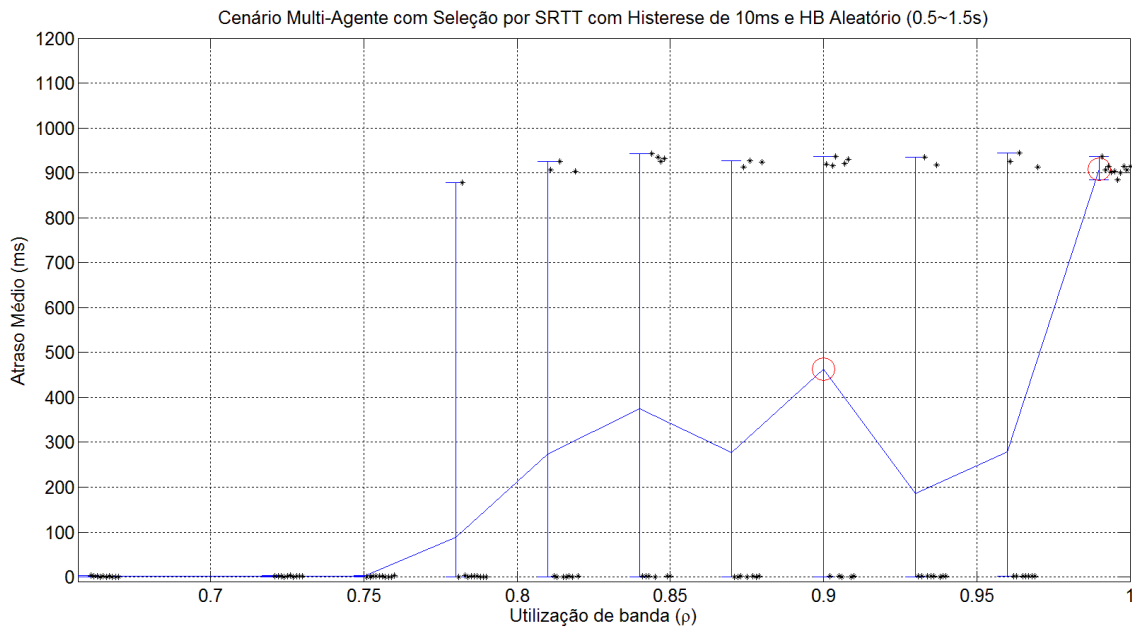


Gráfico 10: HB Aleatório desconsiderando o maior atraso

Já é perceptível a diferença entre os dois gráficos, porém a alta variância ainda permanece para um  $\rho$  mais elevado. A fim de aumentar a aleatoriedade do *Heartbeat*, o próximo teste foi realizado alterando o próximo intervalo de *HB* a cada vez que um HB é enviado.

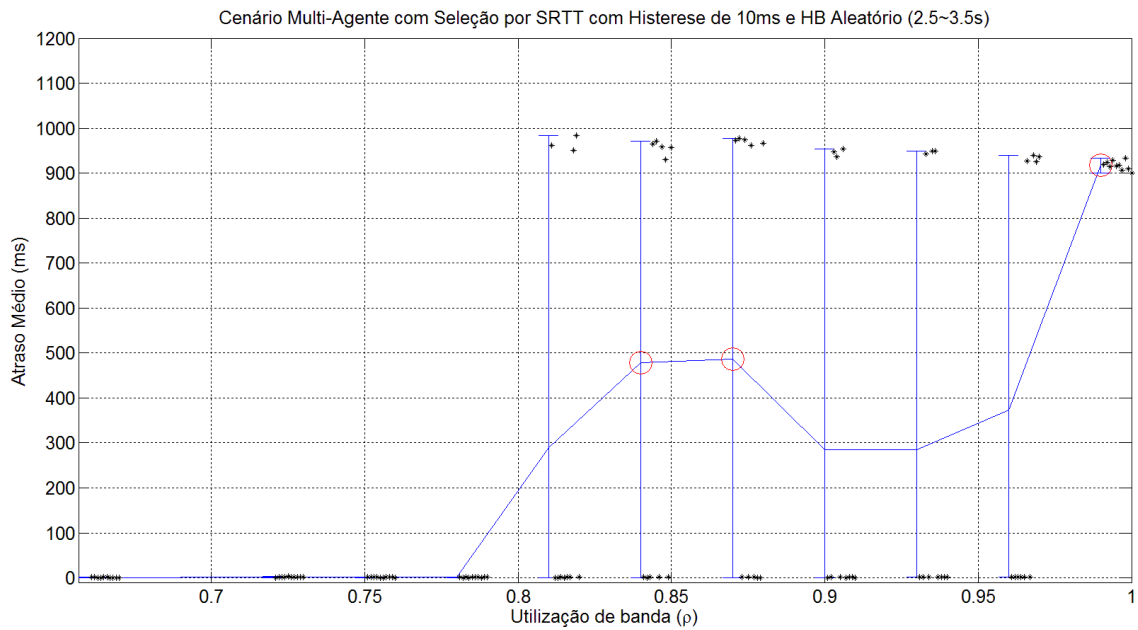




**Gráfico 11: HB Aleatório entre 0.5 a 1.5s**

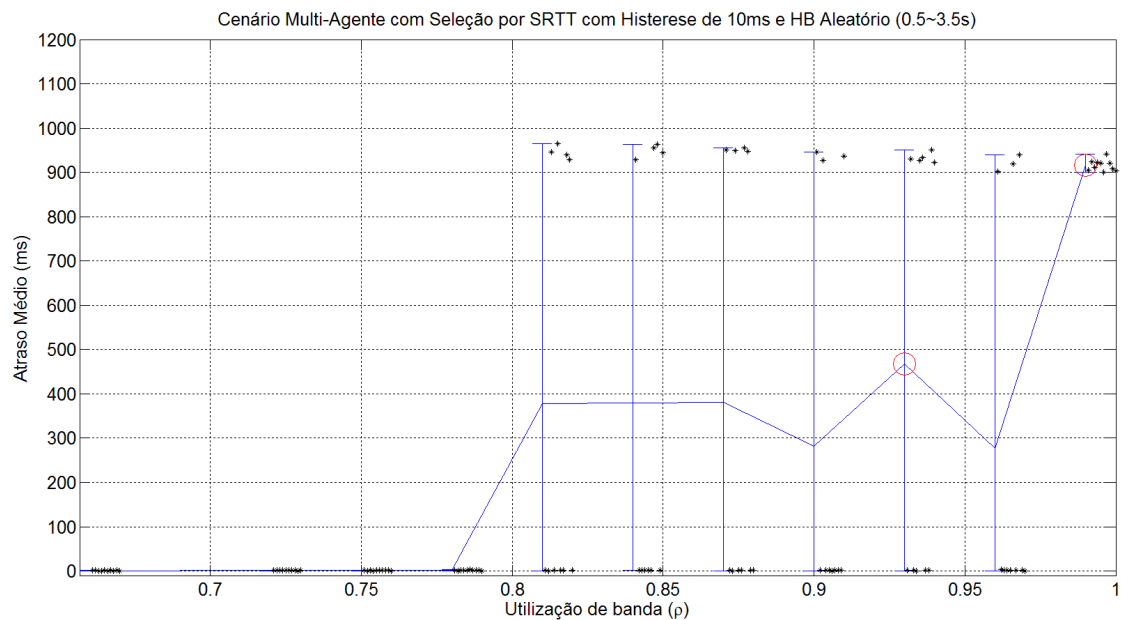
Mesmo com o sorteio do HB a cada envio deste, não houve diferença de desempenho para o método com o sorteio inicial apenas. Por consequência, procurou-se alterar a amplitude e intervalos de aleatoriedade no envio do HB. Caso estas mudanças não causem muitas diferenças nos resultados, outra abordagem deverá ser tomada para eliminar o problema de sincronismo dos agentes.

Primeiramente, foi alterado o intervalo de aleatoriedade para 2.5s até 3.5s, mas também não foram obtidas melhoras no resultado, como mostra o Gráfico 12.



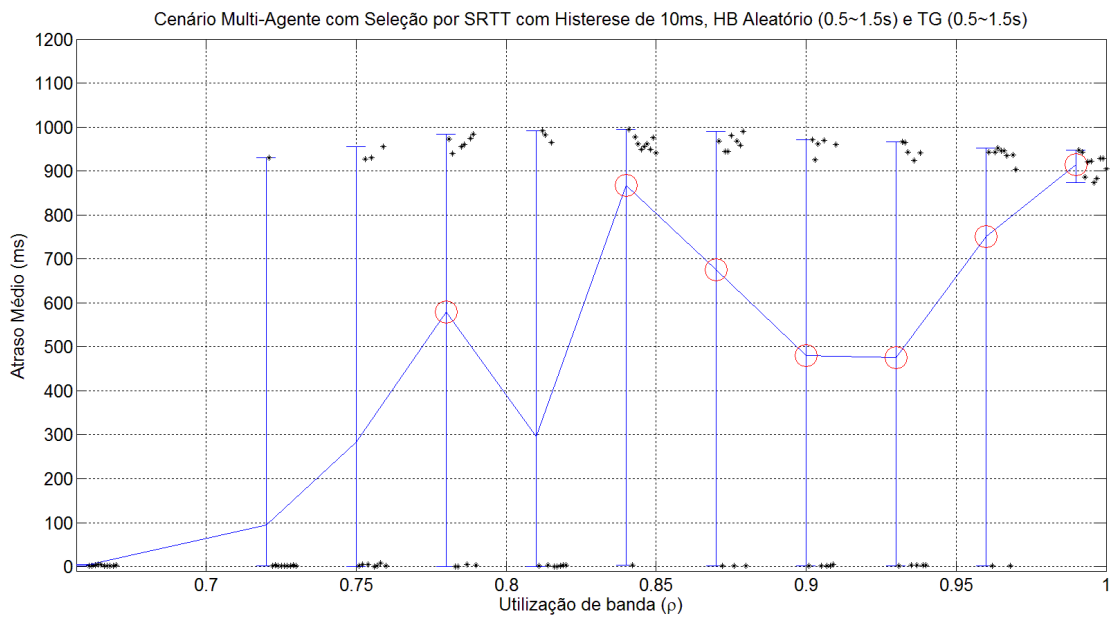
**Gráfico 12: HB Aleatório entre 2.5 e 3.5s**

A ampliação do intervalo de aleatoriedade para 0.5s a 3.5s também obteve índices aceitáveis de estabilidade, mas novamente não houve melhoras significativas de desempenho em comparação com os outros intervalos. Este teste está representado pelo Gráfico 13.



**Gráfico 13: HB Aleatório entre 0.5 e 3.5s**

Outra técnica utilizada em trabalhos anteriores [1] para evitar a instabilidade é a implementação de um Tempo de Guarda (TG). Este tempo de guarda evita que o caminho ativo seja trocado imediatamente após a avaliação de menor latência. Ele aguarda um tempo fixo (ou aleatório) e reavalia se o caminho continua sendo mais vantajoso. Se isto se confirmar o caminho ativo é de fato trocado, senão, nada acontece. Esta implementação foi feita no teste representado pelo Gráfico 14, com um tempo de guarda aleatório uniforme entre 0.5s e 1.5s.



**Gráfico 14: Tempo de Guarda Aleatório entre 0.5 e 1.5s**

Ao contrário do esperado, percebeu-se que o tempo de guarda aleatório juntamente com o HB aleatório aumentou a ineficiência do sistema, sendo que a implementação apenas com o HB aleatório teve menor latência. De próximo, efetuou-se o mesmo teste com o tempo de guarda fixo.

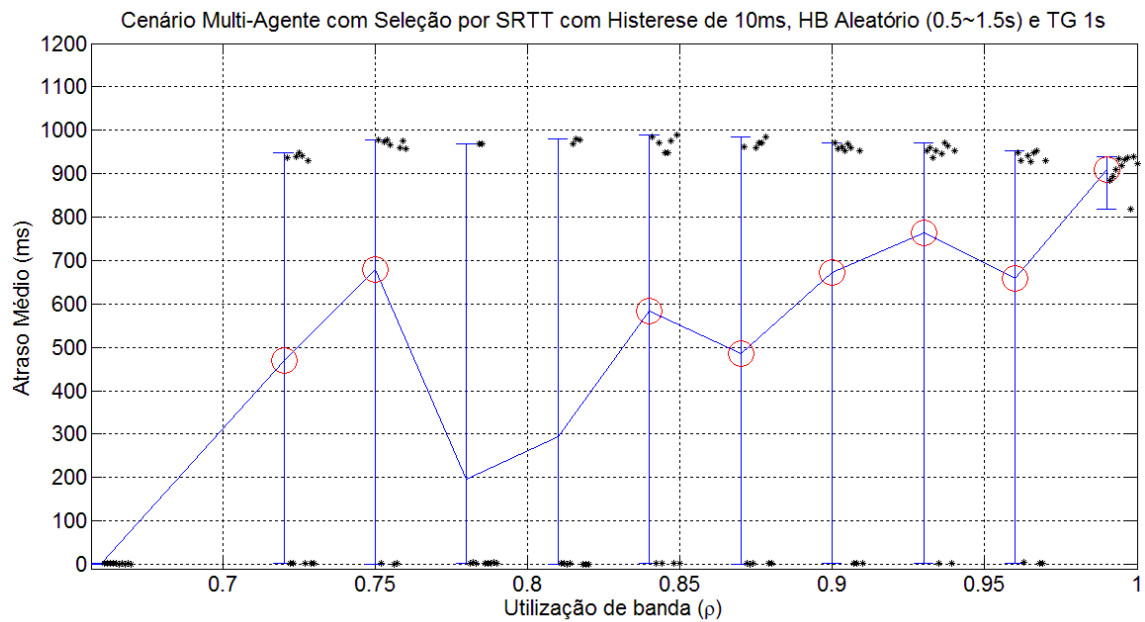


Gráfico 15: Tempo de Guarda de 1s

Novamente, o tempo de guarda não se mostrou muito eficiente, e deixou o sistema se instabilizar até mesmo para um  $\rho$  baixo, como 0.72.

Por fim, procurou-se testar o sistema com o HB aleatório, porém sem nenhuma histerese. Como observado no Gráfico 16, os resultados são muito semelhantes aos obtidos com a histerese de 10ms e por esse motivo os testes foram continuados com histerese, por ser uma abordagem um pouco mais conservadora.

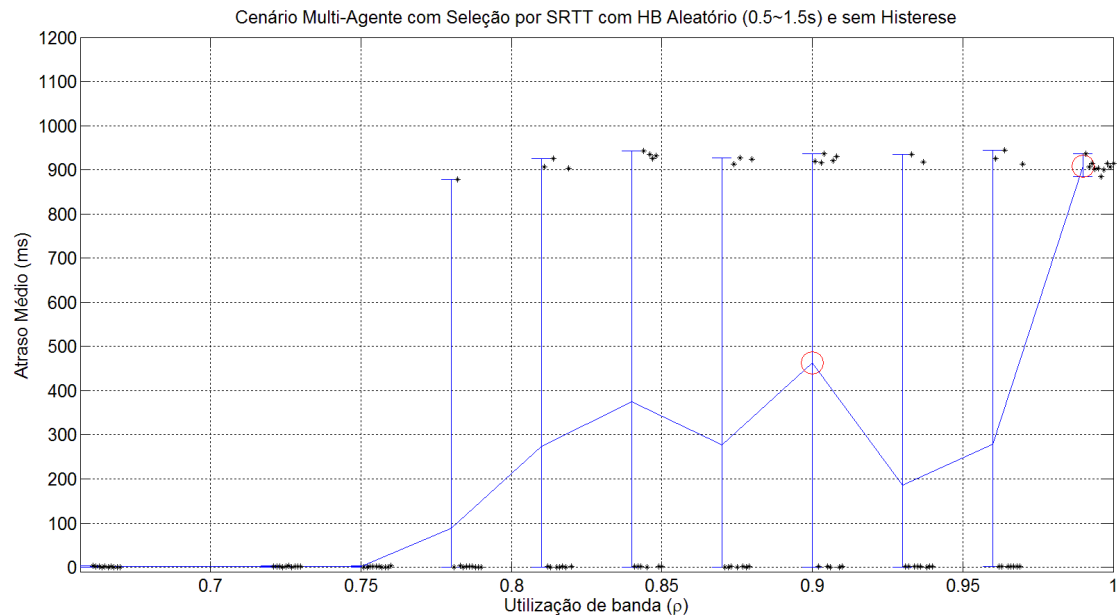


Gráfico 16: Teste sem Histerese

Como forma de representar um comparativo da eficiência de cada método e parâmetro utilizado, a Tabela 1 foi criada indicando o número de vezes em que os agentes se estabilizaram entre os 10 testes realizados.

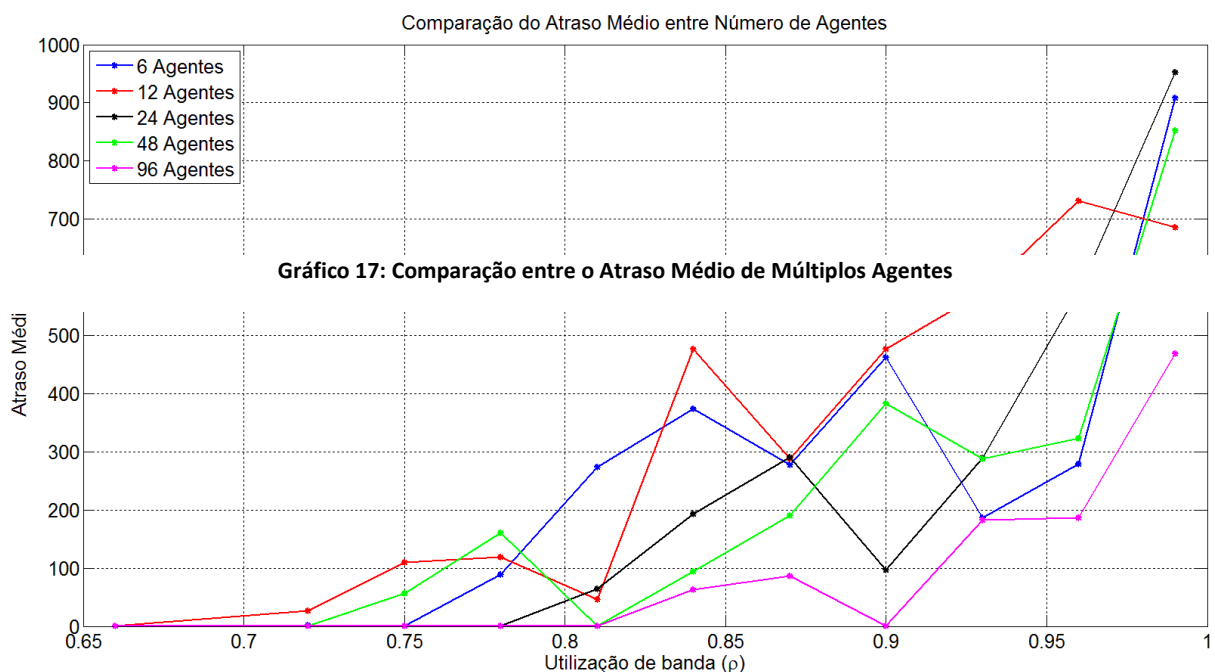
|                  | Histerese<br>60ms | Histerese<br>30ms | Histerese<br>10ms | HB<br>Inicial<br>0.5~1.5s | HB<br>0.5~1.5s | HB<br>2.5~3.5s | HB<br>0.5~3.5s | HB<br>0.5~1.5s<br>TG<br>0.5~1.5s | HB<br>0.5~1.5s<br>TG 1s |
|------------------|-------------------|-------------------|-------------------|---------------------------|----------------|----------------|----------------|----------------------------------|-------------------------|
| $\rho =$<br>0.72 | 0                 | 0                 | 8                 | 10                        | 10             | 10             | 10             | 9                                | 5                       |
| 0.75             | 3                 | 1                 | 10                | 10                        | 10             | 10             | 10             | 7                                | 3                       |
| 0.78             | 3                 | 8                 | 6                 | 9                         | 9              | 10             | 10             | 4                                | 8                       |
| 0.81             | 0                 | 0                 | 6                 | 9                         | 7              | 7              | 6              | 7                                | 7                       |
| 0.84             | 0                 | 0                 | 8                 | 6                         | 6              | 5              | 6              | 1                                | 4                       |
| 0.87             | 0                 | 0                 | 2                 | 5                         | 7              | 5              | 6              | 3                                | 5                       |
| 0.90             | 0                 | 0                 | 3                 | 8                         | 5              | 7              | 7              | 5                                | 3                       |
| 0.93             | 0                 | 0                 | 4                 | 8                         | 8              | 7              | 5              | 5                                | 2                       |
| 0.96             | 1                 | 2                 | 2                 | 6                         | 7              | 6              | 7              | 2                                | 3                       |
| 0.99             | 0                 | 0                 | 0                 | 0                         | 0              | 0              | 0              | 0                                | 0                       |

Tabela 1: Tabela comparativa entre parâmetros

Como forma de analisar a variabilidade dos dados para números de agentes diferentes, foi tornado como padrão o modelo com o *Heartbeat* Aleatório entre 0.5 e 1.5s, já que este foi um dos modelos que apresentou melhor desempenho de estabilidade. A seguir, foi realizado o experimento com o número de agentes variando entre 6, 12, 24, 48 e 96, a fim de mostrar alguma correlação entre estabilidade e número de agentes.

Para conservar os mesmos valores de  $\rho$ , manteve-se como constante os valores de limitação de banda a 500 kbps, e variou-se a banda enviada por cada agente.

O Gráfico 17 mostra o atraso médio de cada agente em função de  $\rho$ . Pela aleatoriedade dos dados mostrados, percebe-se que na maioria dos casos não há muita correlação entre o número de agentes e a estabilidade do sistema. Porém, no último caso, com 96 agentes simultâneos, é possível perceber uma queda considerável no atraso, indicando que os agentes conseguem se acomodar com mais facilidade pelos caminhos conforme aumentamos o número de agentes. Isto pode ser representado como a granularidade de um sistema, como a granularidade de grãos de areia, por exemplo. Quanto menor a granularidade (ou maior número de agentes), mais fácil os grãos de areia se acomodam em um lugar. Entretanto, no sistema em testes, isto não vale em 100% dos casos, e as melhoras reais só começam a aparecer de verdade para um número muito alto de agentes, que dificilmente é alcançado em um caso prático.



Como os resultados de quase todos os testes não apresentaram uma tendência muito definida conforme o valor da utilização de banda, procurou-se aumentar o número de testes. Por se tratar de um comportamento bimodal, os resultados poderiam estar sendo obtidos através de um baixo intervalo de confiança. Como não se sabe a distribuição estatística dos experimentos, não é possível definir a média e o desvio padrão para calcular com qual intervalo de confiança os testes estão sendo feitos. A alternativa foi realizar um número de testes maior, e verificar se a média dos resultados obtidos para 100 testes, por exemplo, é muito diferente para 10 testes.

O Gráfico 18 nos mostra que, independente do número de testes, a média oscila muito perto dos valores já obtidos para 10 testes, ou seja, o intervalo de confiança está razoável da forma que estava sendo feito anteriormente. É visível que para 100 testes, a curva parece mais suavizada, porém os resultados não divergem muito.

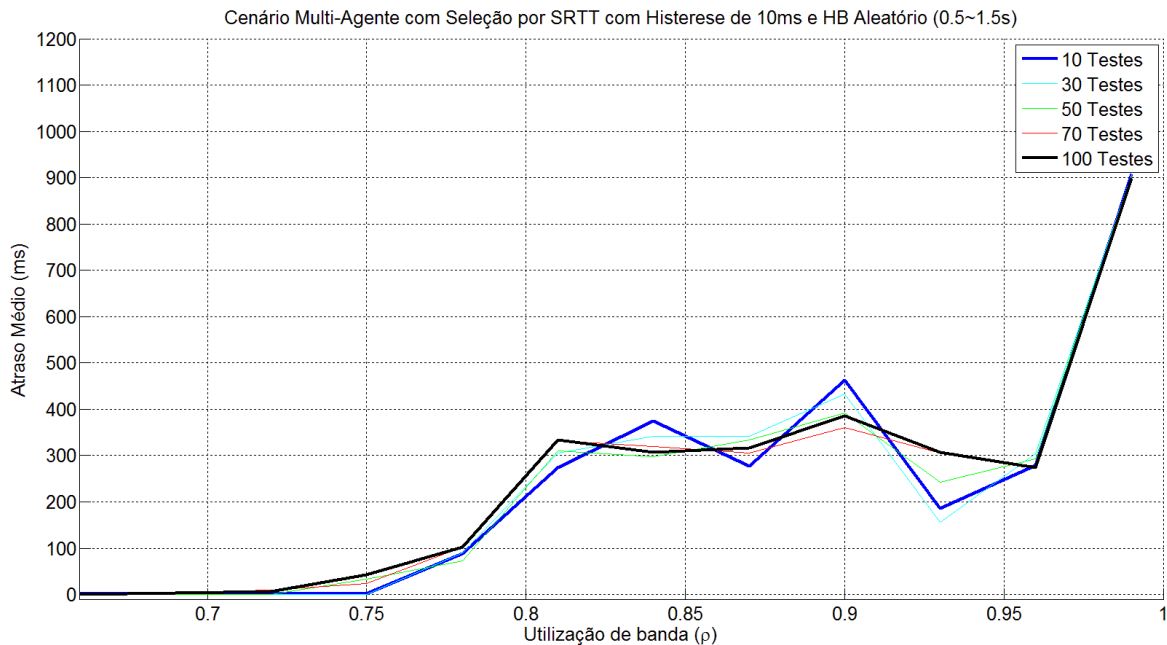


Gráfico 18: Gráfico para diferentes Intervalos de Confiança

Mais um método de seleção de caminho recentemente criado e testado é o *Reactive Delay-centric*, que utiliza-se do indicador MACD (*Moving Average Convergence / Divergence*) [2]. Este indicador é amplamente utilizado na área de compra e venda de ações, por retratar uma possível tendência dos valores. No caso da comunicação multi-abrigada, o valor representa a latência ou o atraso do caminho.

Do cálculo do SRTT, temos que:

$$SRTT_n = (1 - \alpha)SRTT_{n-1} + \alpha RTT_i$$

Equação 1: Cálculo do SRTT

A fim de calcular a tendência do atraso do caminho, são calculados dois SRTT's, um de longo prazo e outro de curto prazo, aqui denominados de  $SRTT_S$  (*short*) e  $SRTT_L$  (*long*). O  $SRTT_S$  utiliza um valor de  $\alpha$  igual a 0.667, e o  $SRTT_L$  um valor para  $\alpha$  de 0.154, conforme sugerido por (TORRES, Alex, 2014).

Com os valores calculados, é definido uma tendência de aumento da latência caso a média de curto prazo ultrapasse a média de longo prazo. Segundo o algoritmo desenvolvido por (TORRES, Alex, 2014), a troca de caminho acontece quando as seguintes condições se tornam verdadeiras:

- O  $SRTT_S$  do caminho secundário seja menor que o do caminho primário.
- O  $SRTT_S$  do caminho primário seja maior que o  $SRTT_L$  do caminho primário (tendência de aumento da latência)
- O  $SRTT_S$  do caminho primário ultrapasse um limiar, definido inicialmente como 150ms.

Seguindo este algoritmo, foi realizado novamente o teste multi-agente, com os resultados exibidos pelo Gráfico 19. Vale notar que por agora se trabalhar com um limiar de troca muito grande e por trabalhar com tendências, o comportamento bimodal muitas vezes deixou de aparecer.

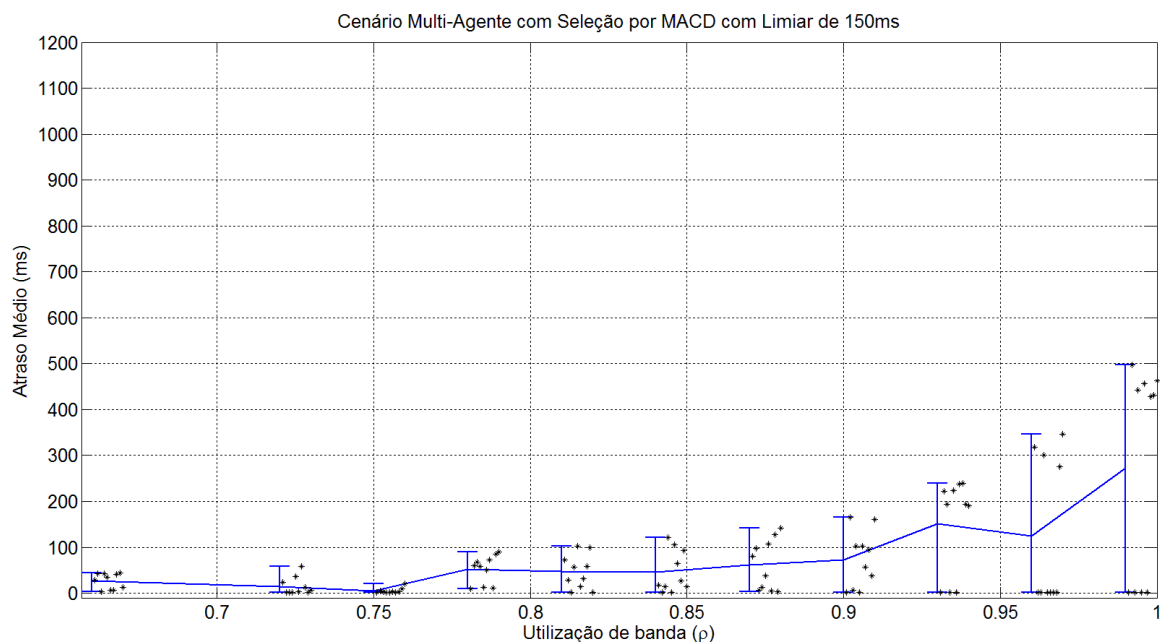


Gráfico 19: Reactive Delay-Centric com Limiar de 150ms

De qualquer forma, é visível a diminuição do atraso médio em relação à outros testes, mesmo que muitas vezes ele não se mantenha abaixo dos milissegundos. A modificação em alguns parâmetros desse método de seleção de caminho pode auxiliar a diminuir os níveis de latência, portanto, a sequência



deste trabalho foi levada nesse aspecto. Como parâmetro inicial, procurou-se alterar o limiar de troca de caminho, inicialmente realizando o teste sem limiar.

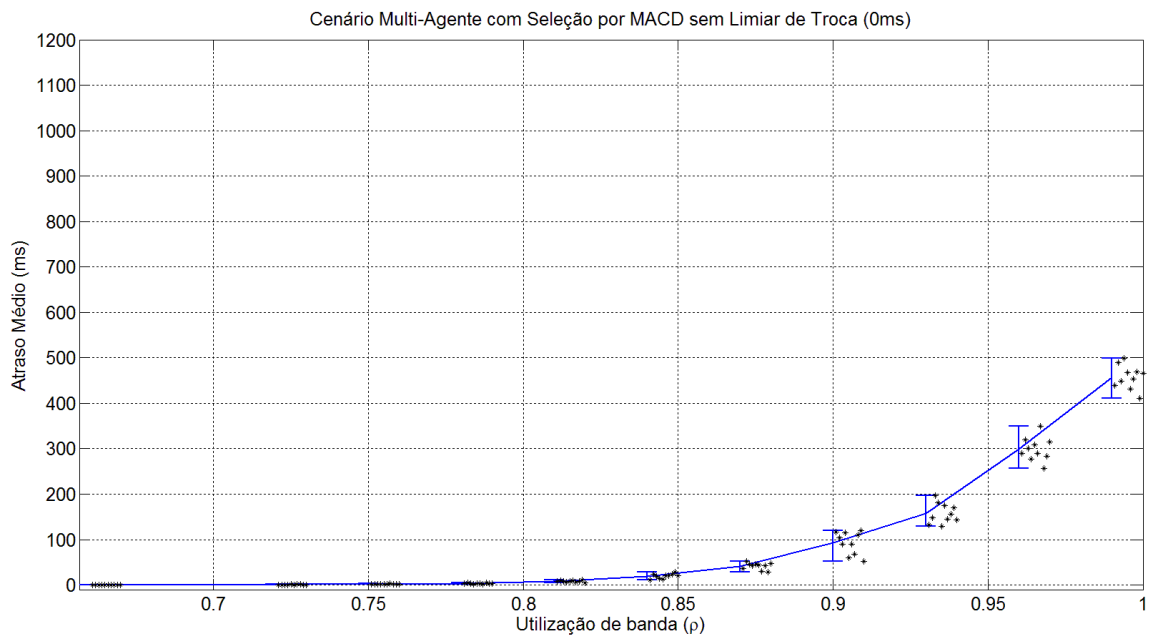


Gráfico 20: Reactive Delay-Centric sem Limiar de Troca

#### Referências Bibliográficas:

[1]:[http://dspace.c3sl.ufpr.br:8080/dspace/bitstream/handle/1884/18581/Dissertacao\\_Igor\\_Gavriloff.pdf?sequence=1](http://dspace.c3sl.ufpr.br:8080/dspace/bitstream/handle/1884/18581/Dissertacao_Igor_Gavriloff.pdf?sequence=1)