4
Comparação de algoritmos

Este Capítulo destina-se a comparar, através de simulações, o desempenho de dois algoritmos de Controle de Admissão projetados para o ambiente DiffServ. Ambos apresentam soluções originais e bem diversas entre si ao problema da sinalização na arquitetura de Serviços Diferenciados. O primeiro, designado pela sigla GRIP, utiliza pacotes de sondagem (*probe*) e o outro, o mecanismo de passagem de token.

Para análise dos algoritmos, utilizamos os procedimentos usualmente adotados na literatura: são observadas a taxa de descartes e a utilização do enlace, e o principal indicador de desempenho dos algoritmos é a curva "perda-carga" (*loss-load*). Esta curva relaciona a taxa de perda de pacotes com a utilização do enlace.

O objetivo de qualquer mecanismo de Controle de Admissão é maximizar a utilização da rede mantendo os parâmetros de desempenho dentro de níveis aceitáveis. Assim, o algoritmo que apresentar a menor taxa de perda para um mesmo nível de utilização é considerado superior.

4.1
GRIP

O primeiro algoritmo a ser analisado é apresentado em [17] e traz inovações ao conceito de Controle de Admissão Distribuído: cada nó (seja de borda ou núcleo) possui um mecanismo de medição através do qual ele decidirá se está ou não em condições de admitir mais fluxos. A transgressão do paradigma DiffServ é evitada, levando-se em consideração que não há sinalização explícita em nenhum momento do processo de admissão dos fluxos. A esse algoritmo, é dado o nome de **GRIP** (*Gauge&Gate Reservation with*

Independent Probing). A seguir, teremos uma análise mais detalhada desse processo.

4.1.1 Descrição do algoritmo

A proposta do GRIP se baseia em mecanismos de sinalização implícita fim-a-fim através do campo DSCP do cabeçalho dos pacotes IP. Assim, todo o processo de tratamento de fluxos admitidos, de fluxos de sinalização e de tráfego de melhor-esforço (*best-effort*) é realizado através da política de encaminhamento (*Per-Hop Behaviour* - PHB) de cada classe.

São definidas três classes de tráfego, correspondendo a três filas (com diferentes prioridades de atendimento) em cada nó do domínio DiffServ. Estas definições são apresentadas a seguir.

- **Admitido:** são os fluxos que já passaram pelo mecanismo de controle de admissão e foram admitidos. É a fila de maior prioridade;
- **Probe:** é o tráfego de sinalização, que realiza a solicitação de admissão ao domínio. Tem prioridade menor do que a classe de fluxos admitidos;
- **Melhor-esforço:** classe de tráfego não admitido, atendida na medida da disponibilidade dos recursos da rede, sem garantias. É a classe de menor prioridade.

A classificação dos fluxos se dá no nó de entrada do domínio DiffServ, de acordo com a seguinte disciplina: os pacotes pertencentes a fluxos admitidos recebem o código DSCP correspondente a essa classe e são encaminhados com prioridade sobre as outras classes. Cada nó (seja de borda ou núcleo) realiza medições periódicas do volume de tráfego que circula em suas portas, e baseado nesses dados, se estabelece em um dos dois possíveis estados, ADMISSÃO ou REJEIÇÃO, assim definidos.

- ADMISSÃO: indica que a entrada de novos fluxos não resultará em comprometimento dos parâmetros de QoS.
- REJEIÇÃO: caracteriza grande volume de tráfego dos fluxos admitidos, de forma que a admissão de novos fluxos resultaria em degradação de QoS.

Esses estados caracterizam o comportamento do nó na presença de um pacote de probe, ou seja, a sua disponibilidade para aceitar ou não a entrada

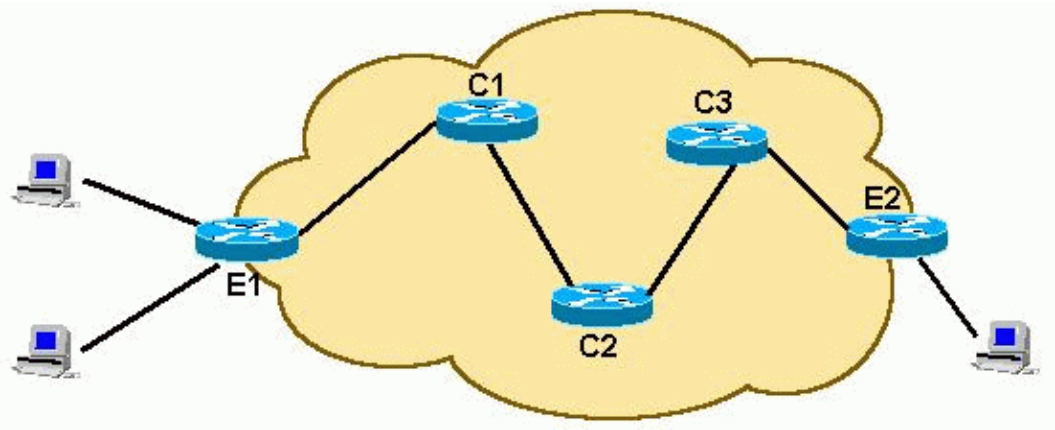

Figure 4.1: Domínio DiffServ

de novos fluxos. Se estiver no estado ADMISSÃO, o nó encaminha os pacotes de probe normalmente. Se estiver no estado REJEIÇÃO, descarta todos os pacotes de probe que chegarem, impedindo assim a admissão de fluxos que passem por este nó.

No caso de um novo usuário solicitar conexão a um nó de borda, este inicia a fase de probe, que segue os seguintes passos, ilustrados com auxílio da Figura 4.1:

1. Se o nó de entrada (E1) estiver no estado ADMISSÃO, envia um pacote de probe ao nó de saída do domínio DS (E2), seguindo a rota definida pelo algoritmo de roteamento e inicia um temporizador;

2. Se algum nó entre E1 e E2 estiver em REJEIÇÃO, o pacote probe será descartado, e eventualmente o temporizador vai chegar ao limite pré-estabelecido que indica que a solicitação foi rejeitada;

3. Se todos os nós entre E1 e E2 estiverem em ADMISSÃO, o pacote de probe chegará a E2 e este enviará um pacote de confirmação na classe ADMITIDO a E1;

4. Recebendo a confirmação, E1 atribui à aplicação solicitante (USUÁRIO) o código DSCP referente à classe ADMITIDO. A partir daí, tem início efetivamente a transmissão de dados do usuário.

A transição de um nó do estado ADMISSÃO para o estado REJEIÇÃO se dá pela comparação entre o volume de tráfego medido em um determinado período de tempo ΔT e um valor-limite calculado previamente, que leva em consideração a caracterização das fontes e o objetivo de desempenho (taxa de perda máxima admissível). Um destes procedimentos de cálculo é proposto por Mitra e Elwalid e utilizado em [30] e [29]. Baseado nesses cálculos, o algoritmo

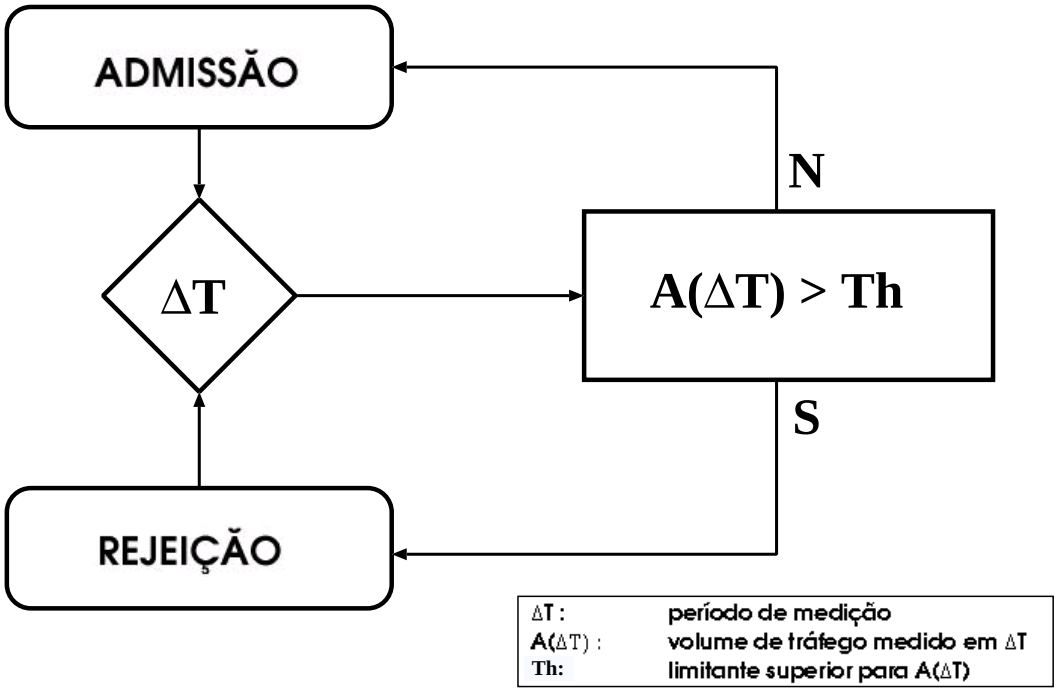

Figure 4.2: Disciplina GRIP

estabelece um limiar que está diretamente associado ao número máximo de usuários que podem ser admitidos.

Ao fim de cada período de medição ΔT, o algoritmo compara o volume de tráfego medido no período com o limiar pré-determinado, dado por Th. Para esta implementação, Th foi definido em função da taxa média (R) das fontes de tráfego através da expressão

$$Th = M.R.\Delta T \tag{4-1}$$

Dessa forma, o parâmetro M, que é o limiar Th normalizado pelo volume médio de cada fluxo, pode ser interpretado como o número médio de fluxos que podem ser admitidos, e funciona como um mecanismo de ajuste da Qualidade de Serviço do sistema: quanto menor o valor de M, menos fluxos serão admitidos, resultando em menor atraso e probabilidade de descarte de pacotes, mas ao mesmo tempo, reduzindo a utilização.

O período das avaliações, referido como janela de tempo, influencia fortemente o desempenho do algoritmo, como será visto mais à frente.

A Figura 4.2 ilustra o funcionamento de um nó sob a disciplina GRIP.

4.1.2 Avaliação de desempenho

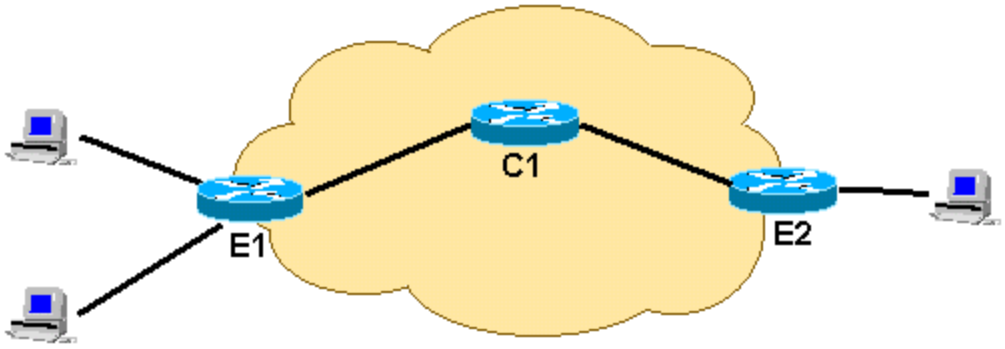

Figure 4.3: Topologia simples utilizada nas simulações

Para avaliação do desempenho do GRIP, os critérios utilizados foram definidos na introdução deste Capítulo. A principal análise será, portanto, referente à capacidade do algoritmo de controlar os níveis de descartes no domínio, buscando sempre a maior utilização possível.

Para cada situação em que é testado o algoritmo, o resultado é comparado com o da situação sem Controle de Admissão. Ou seja: a simulação é repetida com a mesma topologia, as mesmas características das fontes, a mesma taxa de chegada de usuários, porém sem qualquer disciplina controlando a entrada de usuários. Com isto, pode-se quantificar os benefícios do Controle de Admissão.

Obviamente, só faz sentido falarmos em utilizar Controle de Admissão num domínio com escassez de recursos. Assim, as fontes de tráfego utilizadas foram configuradas para introduzir uma carga de tráfego elevada no sistema. Neste caso, a simulação do sistema sem Controle de Admissão indicará alta utilização acompanhada de alta taxa de descartes. O objetivo de implantação do Controle de Admissão é manter a utilização alta, reduzindo a taxa de perda.

Inicialmente, a topologia utilizada é a mais simples possível: um nó de entrada, um nó de núcleo e um nó de saída. Esta topologia está representada na Figura 4.3, onde todos os enlaces têm capacidade de 2 Mbps e buffers com capacidade para 10 pacotes. A escolha dessa topologia se deve ao fato de que os aspectos básicos do desempenho do algoritmo são mais facilmente observados e interpretados numa configuração simples, em um gargalo da rede, por exemplo. A análise de uma topologia um pouco mais complexa será realizada posteriormente.

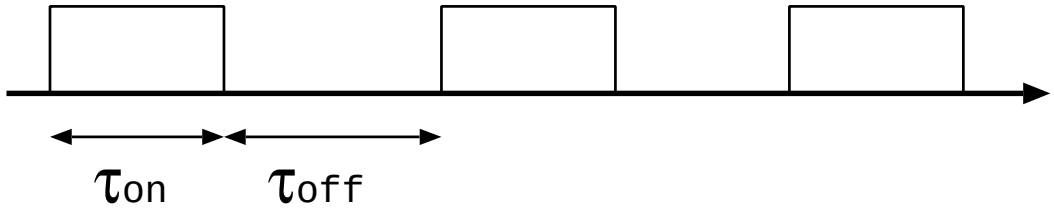

Figure 4.4: Caracterização das fontes ON-OFF

Fontes de tráfego

Foram feitas simulações com três tipos de tráfego: *on-off* periódico, *on-off* exponencial e MPEG. Os modelos das fontes de tráfego são descritos a seguir.

ON-OFF periódico e ON-OFF exponencial

Seguindo a nomenclatura do *ns*, os modelos de fonte denominados *on-off* periódico e exponencial serão referenciados, respectivamente, como CBR e EXP. Na verdade, as fontes *on-off* periódicas não são fontes CBR (*constant bit rate*), mas mantivemos essa nomenclatura para especificar o modelo de fonte de tráfego utlizado nas simulações com o *ns*. Correspondem a fluxos com uma sequência de pacotes de bits como ilustrado na Figura 4.4, onde se observa um período ativo (on) com duração τ_{on} e um período de silêncio (off) com duração τ_{off}. Durante o período ativo, a transmissão é feita a uma taxa de bits constante R_{on} e a taxa média R é dada por

$$R = \frac{R_{on}.\tau_{on}}{\tau_{on} + \tau_{off}} \tag{4-2}$$

As fontes foram configuradas para transmitir a uma taxa média de 50 kbps.

No caso das fontes CBR, $\tau_{on} = \tau_{off}$ e, neste caso, para uma taxa média de 50 kbps, a taxa no período ativo é de 100 kbps. Neste período, transmite-se um único pacote de 120 bytes, o que corresponde a $\tau_{on} = 0.96ms$.

As fontes EXP podem ser vistas como um modelo para transmissão de voz em pacotes com deteção de silêncio. Os parâmetros τ_{on} e τ_{off} são variáveis aleatórias exponenciais de médias 1.004 segundo e 1.587 segundo respectivamente. Estes valores são especificados em [52] como valores representativos de uma conversação. Para simulação são utilizados também pacotes de 120 bytes transmitidos sucessivamente durante o período ativo.

As fontes geram fluxos de acordo com processos de Poisson, ou seja, o intervalo entre o início de dois fluxos consecutivos é uma variável aleatória exponencial de média δ. A duração de cada fluxo é também uma variável exponencial, com a média dada por τ_f. Associando-se este processo a uma fila $M/M/\infty$, pode-se mostrar que o número de fluxos ativos em um determinado instante tem distribuição de probabilidade de Poisson dada por

$$P_k = \frac{A^k}{k!} e^{-A} \tag{4-3}$$

onde $A = \frac{\tau_f}{\delta}$ é a média da distribuição.

Foram especificados os valores $\tau_f = 30s$ e $\delta = 0.2s$, o que corresponde a um número médio de 150 fluxos ativos com uma taxa total média de 7.5 Mbps.

MPEG

Para simular o tráfego MPEG, foi utilizado o *trace* de um trecho do filme *Star Wars* ([43]) com 10 segundos de duração codificado em MPEG com taxa média de 320 kbps, utilizando pacotes de 200 bytes. Fluxos com este mesmo *trace* são gerados com início aleatório de acordo com um processo de Poisson, ou seja, o intervalo entre os inícios de dois fluxos consecutivos é uma variável exponencial com média 1 segundo. O aumento no intervalo médio entre fluxos foi definido para sobrepor uma limitação do *ns* na contagem de grande número de pacotes.

Neste caso, usando-se modelagem mais simples, também se chega a (4-3). Para os valores especificados acima, tem-se um número médio de fluxos igual a 10 em um determinado instante, com taxa média total de 3.2 Mbps.

Na tabela 4.1 estão resumidas as especificações descritas acima.

4.1.3 Simulações

Como há apenas um nó de entrada na topologia da Figura 4.3, e as configurações dos enlaces e dos nós são iguais (capacidade e retardo do enlace, tamanho da fila, etc), o nó E1 faz uma moldagem de tráfego, enfileirando pacotes quando a taxa de chegada for maior que a taxa de atendimento, e descartando pacotes quando a fila estiver cheia. Portanto, o fluxo de dados que chega a C1 está já moldado e este nó não sofre nunca com congestionamento.

Fonte	**Taxa média** (kbps)	**Período ON** (s)	**Período OFF** (s)	**Pacote** (bytes)	**Duração do fluxo** (s)
CBR	50	0.0096 (determ)	0.0096 (determ)	120	30 (média)
EXP	50	1.004 (média)	1.587 (média)	120	30 (média)
MPEG	320	-	-	200	10 (determ)

Table 4.1: Configuração das fontes de tráfego

Assim, o ponto crítico dessa topologia é o nó E1, e por isso, este é o nó monitorado pelo *ns* nas simulações. Os dados aqui apresentados foram obtidos através de medições neste nó e no enlace entre este e C1.

Para comprovarmos a confiabilidade do programa, foram realizados diversos testes. Primeiramente, para definirmos o tempo de simulação a ser utilizado, buscamos repetir várias vezes a mesma simulação, para diversos valores de tempo. O objetivo é determinar uma duração suficientemente grande para garantir que o sistema esteja em estado estacionário.

Seguindo esse processo, obtivemos resultados estáveis para valores de tempo a partir de 2000 segundos. Estes parâmetros correspondem a um certo intervalo de confiança, cujos valores são apresentados no Apêndice C. Consideramos, assim, que esse tempo é suficientemente grande para que os resultados possam ser considerados válidos do ponto de vista da estabilidade do sistema.

Para aumentar a confiabilidade, as simulações foram repetidas 10 vezes com sementes diferentes. Os dados que serviram de base aos gráficos aqui apresentados são a média dos resultados das 10 repetições. Além disso, só consideramos os resultados que apresentaram probabilidade de perda maior que zero.

Os demais testes de validação são apresentados ao longo da seção.

Nas Figuras 4.5 e 4.6 são apresentadas curvas de utilização e perda em função do parâmetro M para diferentes valores da janela de tempo de medição: 0.05 segundo, 0.1 segundo, 0.5 segundo, 1 segundo e 5 segundos. Escolhemos esses valores com o intuito de manter o intervalo de medição próximo do intervalo médio entre usuários (0.2 segundo para fluxos CBR e EXP, e 1 segundo para MPEG).

Como já foi mencionado, o tamanho da janela de medição (ΔT), é um parâmetro que, teoricamente, influi diretamente no desempenho do algoritmo. A princípio, pode-se dizer que a escolha de janelas de medição maiores resulta em dados mais precisos. Porém, períodos de observação longos significam baixa taxa de atualização dos dados utilizados para as decisões de admissão. Como consequência, teremos um esquema de Controle de Admissão lento.

Por outro lado, janelas de medição muito curtas podem resultar em medições equivocadas, considerando que o instante de início dos fluxos e as suas durações são variáveis aleatórias. Essa característica resultaria em alta suscetibilidade do Controle de Admissão às flutuações do tráfego.

Os valores de M foram escolhidos para que as simulações retratem a região crítica em que a soma das taxas médias dos fluxos se aproxima (e até ultrapassa) a capacidade do link. Para as fontes CBR e EXP, de taxa média 50 kbps, M varia de 30 a 50. Como explicado anteriormente, M pode ser intrepertado como o número médio de fluxos que podem ser admitidos simultaneamente. Assim, com M assumindo o valor 40, por exemplo, pode-se entender que o sistema admitirá até 40 fluxos ativos, o que representa uma taxa média de 2 Mbps, que é a capacidade do enlace, e portanto, representa um bom teste à capacidade do esquema de Controle de Admissão. Seguindo essa metodologia, a faixa de valores de M escolhida para as simulações com tráfego MPEG (taxa média 320 kbps) é de 1 a 10.

Podemos observar na Figura 4.5 obtida através de simulações com fontes CBR que a utilização aumenta com M, como esperado. Note-se nessas Figuras o nível de utilização obtido sem Controle de Admissão ("Sem CAC"). Vemos que a janela de 5 segundos apresenta uma ligeira vantagem em relação à utilização do enlace, enquanto os outros valores de janela não mostram grande variação entre si.

A Figura 4.6 apresenta a taxa de perda de pacotes para a situação acima descrita. Nota-se que a janela de 5 segundos leva a taxas de perda bem maiores do que as janelas menores. Podemos explicar esse comportamento da seguinte maneira: com as janelas menores, o mecanismo de Controle de Admissão monitora de maneira mais eficiente a ocupação do enlace, porque atualiza as medições a intervalos menores. Assim sendo, a alta ocupação do enlace em determinado instante de tempo será detectada mais cedo, em relação às outras janelas. Detectando mais cedo o congestionamento (ou a iminência deste), o mecanismo bloqueia a entrada de novos fluxos, resultando em utilização menor. Com um menor número de fluxos ativos, há menos pacotes circulando no domínio e menos descartes.

Considerando como o principal parâmetro de desempenho a probabili-

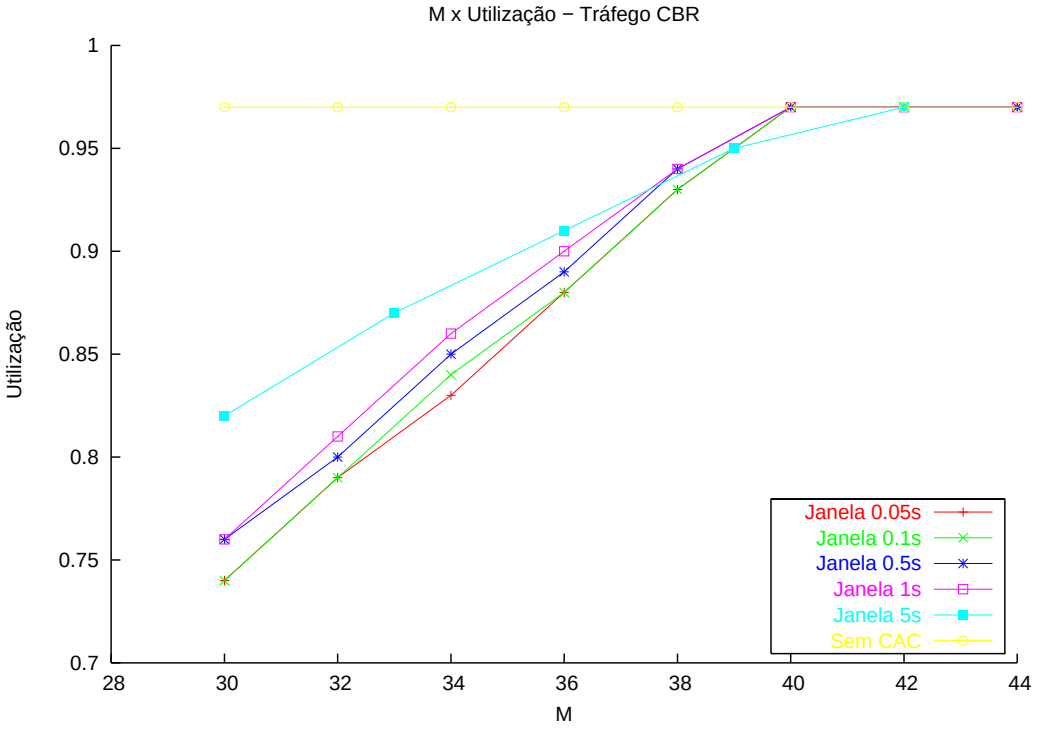

Figure 4.5: GRIP-Utilização do enlace com tráfego CBR - Topologia 1

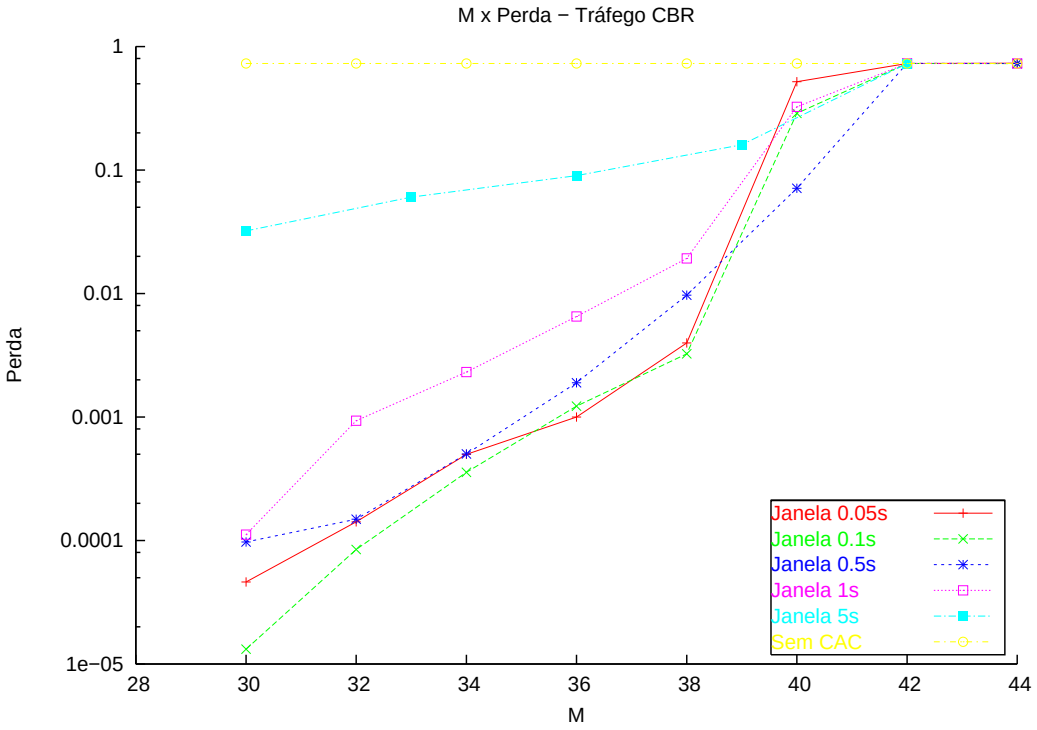

Figure 4.6: GRIP-Taxa de perda de pacotes com tráfego CBR - Topologia 1

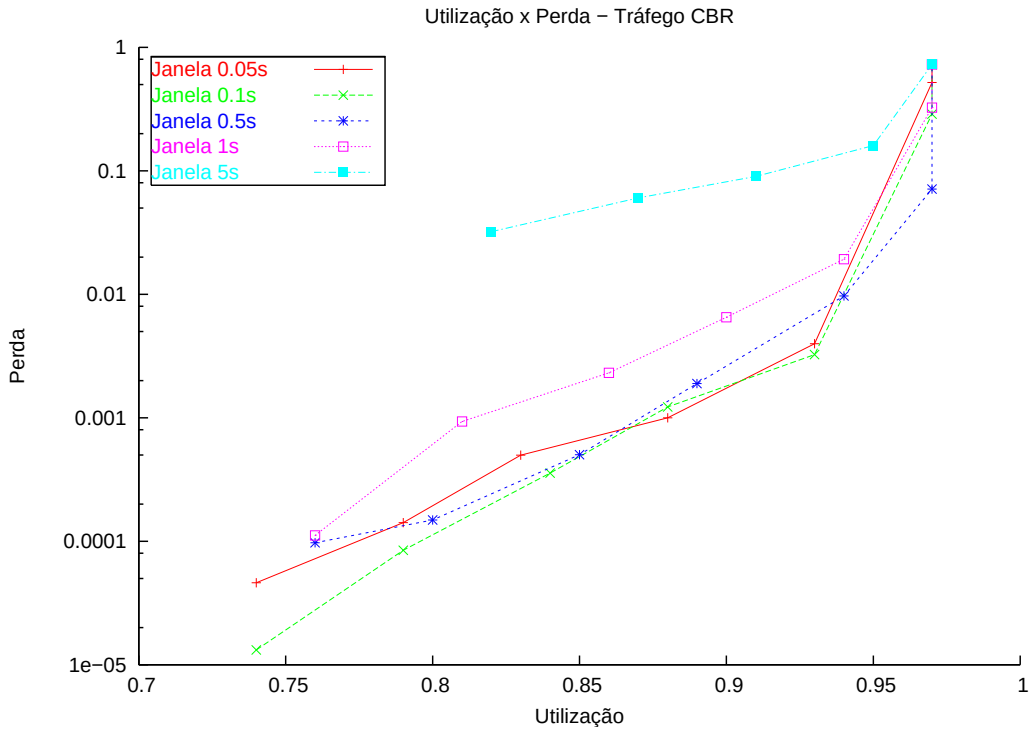

Figure 4.7: Curva perda-carga do GRIP com tráfego CBR - Topologia 1

dade de perda, a melhor configuração do algoritmo de Controle de Admissão é aquela que, uma vez fixado o limitante superior para a probabilidade de perda, oferecer a maior utilização. Essa avaliação é representada pela curva perda-carga, na Figura 4.7.

Nessa Figura, nota-se uma contínua melhora no desempenho do algoritmo com a diminuição da janela, até uma certa estabilização para valores abaixo de 0.5 segundo, indicando uma aproximação da configuração ótima do algoritmo nessas condições de tráfego.

A medição do volume de tráfego com janelas grandes tende a ser imprecisa, pois aproxima as taxas dos fluxos pelas suas taxas médias. No caso do tráfego CBR, esse fator não é crítico, pois a taxa de transmissão é constante. Isso explica o fato de a janela de 5 segundos apresentar a maior utilização. Porém, em tráfegos mais explosivos, como MPEG, observamos a ineficiência dessas janelas.

Um ponto interessante a ser observado em relação ao comportamento do Controle de Admissão é a fração de usuários que foram admitidos em relação ao número de usuários que solicitaram serviço. A Figura 4.8 traz os números de usuários solicitantes e admitidos ao final das simulações do GRIP com tráfego CBR. Esse gráfico nos permite observar a eficiência do mecanismo na sua tarefa de regular a entrada de novos fluxos: o número de solicitações é sempre bastante elevado, no entanto o número de admissões é controlado por M.

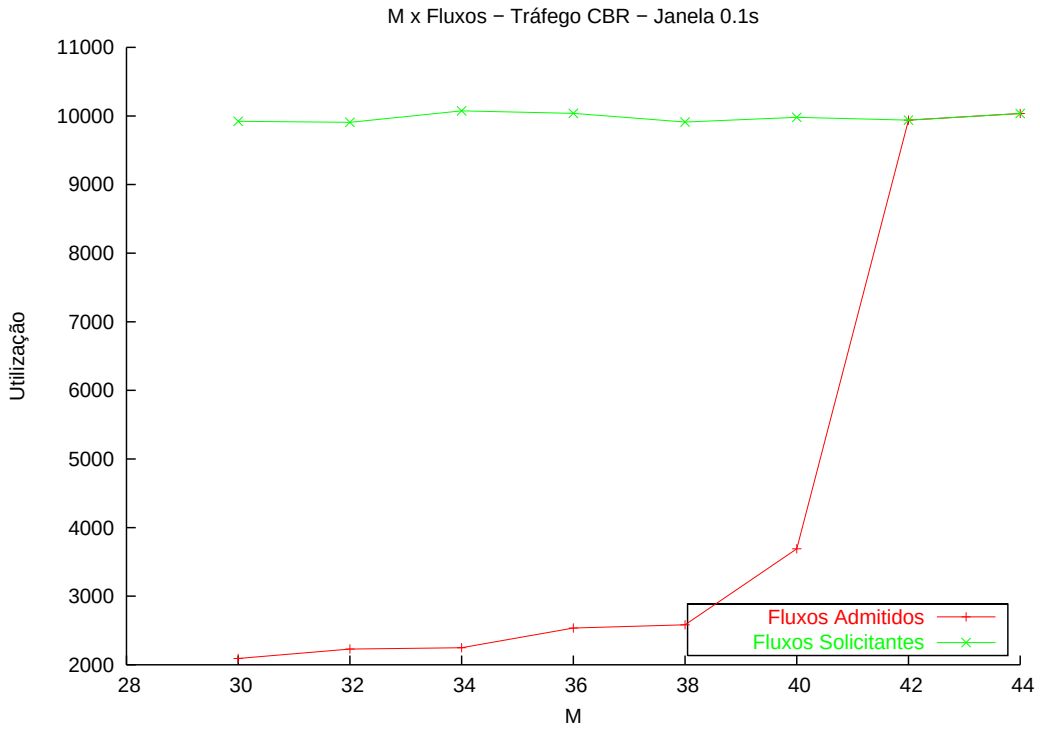

Figure 4.8: Fluxos admitidos pelo GRIP com tráfego CBR - Topologia 1

Lembrando que os fluxos são gerados com intervalo médio de 0.2 segundo, tem-se em média 5 fluxos gerados por segundo e um total de 10000 fluxos gerados no tempo de simulação de 2000 segundos. Como se observa na Figura 4.8, uma grande fração destes é bloqueada até o ponto em que o Controle de Admissão se torna praticamente inefetivo (para M um pouco acima de 40). A razão disto é que para M acima de 40, tem-se

$$\text{Th} > 40.\text{R}.\Delta T$$
$$\text{Th} > 2\text{Mbps}.\Delta T$$

Ou seja, o mecanismo passa a admitir um volume de tráfego em média superior à capacidade do enlace. Assim, para valores de M superiores a 40, as medições são sempre inferiores a Th e os nós estão sempre no estado ADMISSÃO.

Para o tráfego *on-off* exponencial, as simulações estão expostas nas Figuras 4.9, 4.10, 4.11, e 4.12 e para as fontes MPEG, nas Figuras 4.13, 4.14 e 4.15.

Nesta seção, procuramos ressaltar alguns aspectos apontados pelos gráficos e alguns comportamentos característicos. As conclusões e comparações são apresentadas em outra seção.

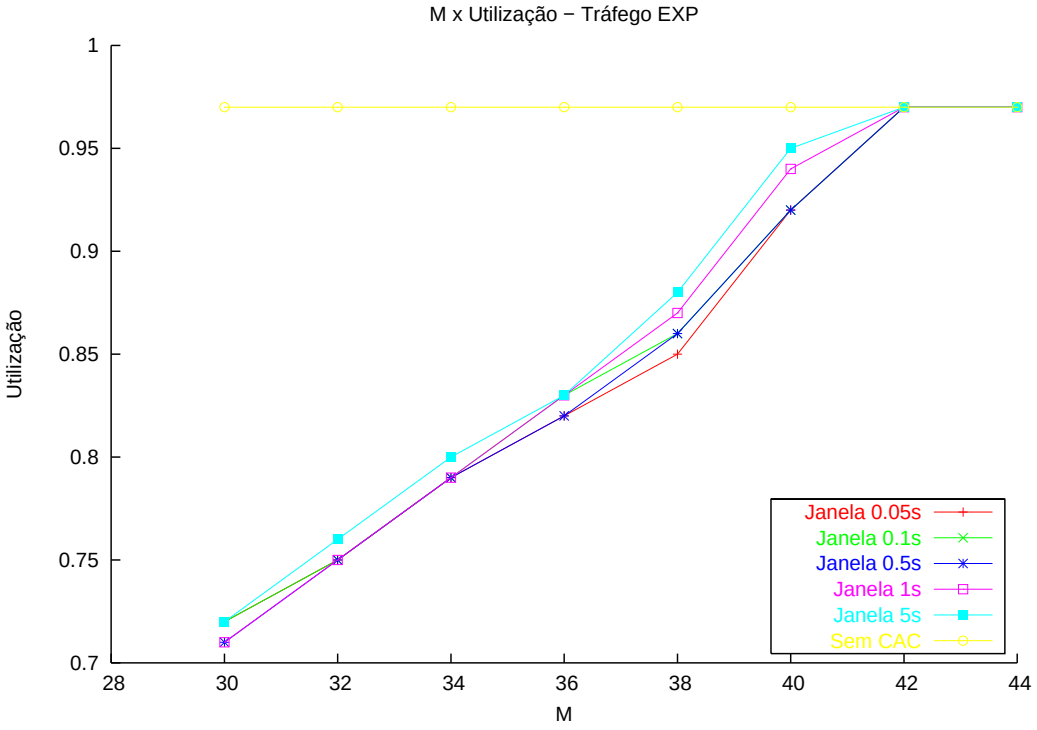

Figure 4.9: GRIP-Utilização do enlace com tráfego EXP - Topologia 1

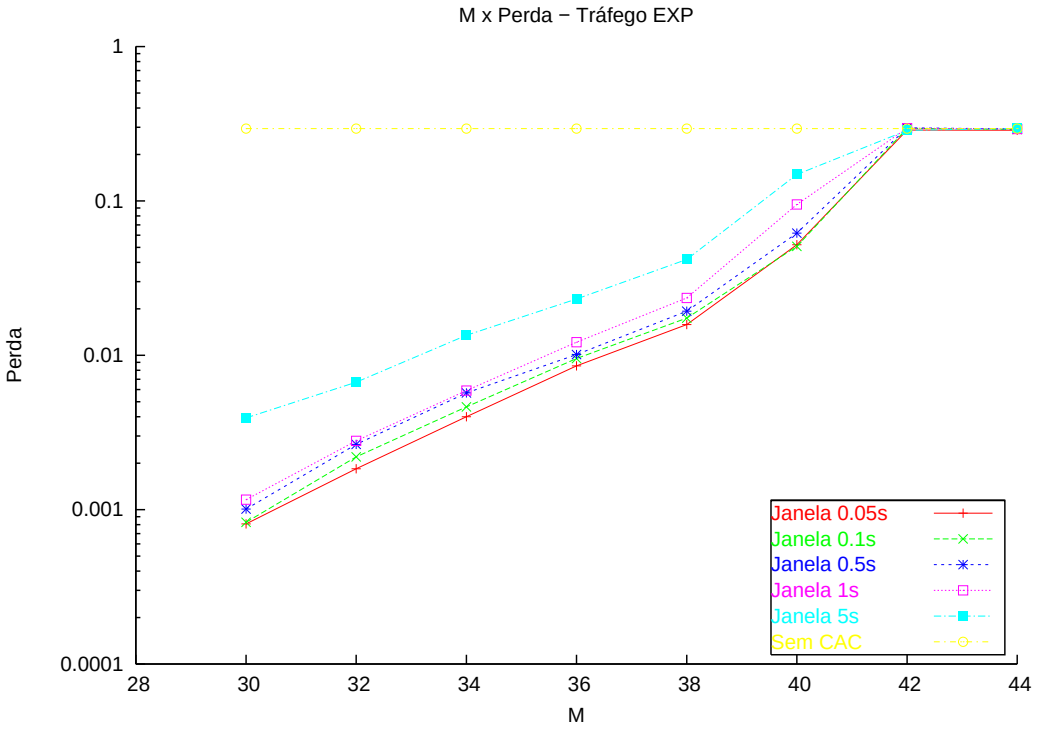

Figure 4.10: GRIP-Taxa de perda de pacotes com tráfego EXP - Topologia 1

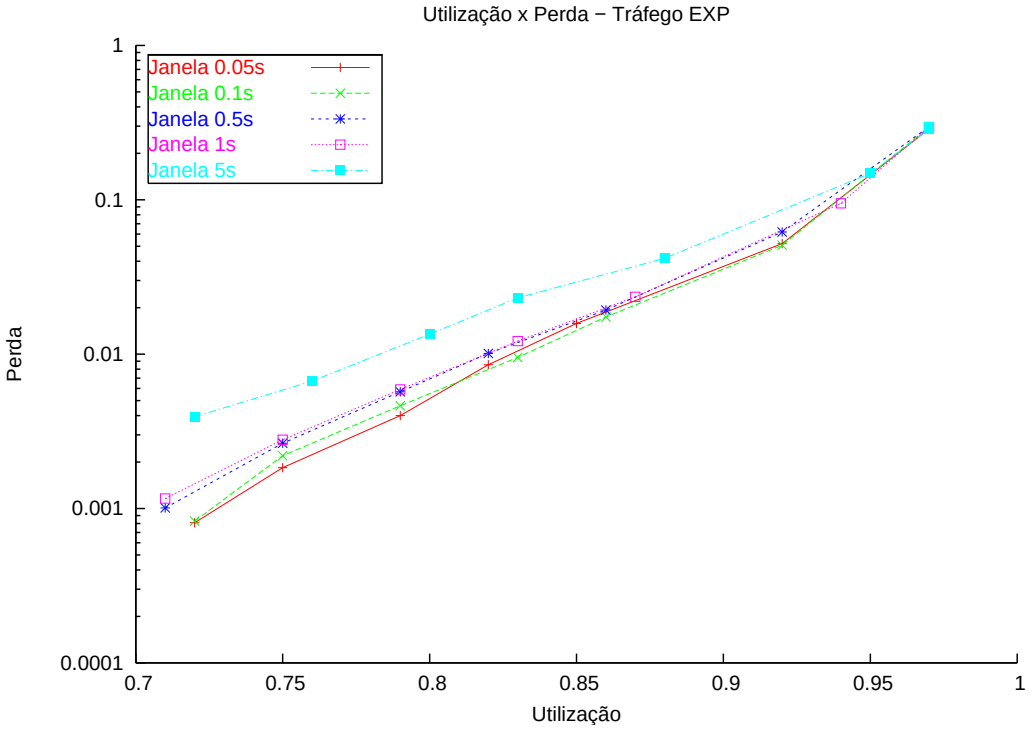

Figure 4.11: Curva perda-carga do GRIP com tráfego EXP - Topologia 1

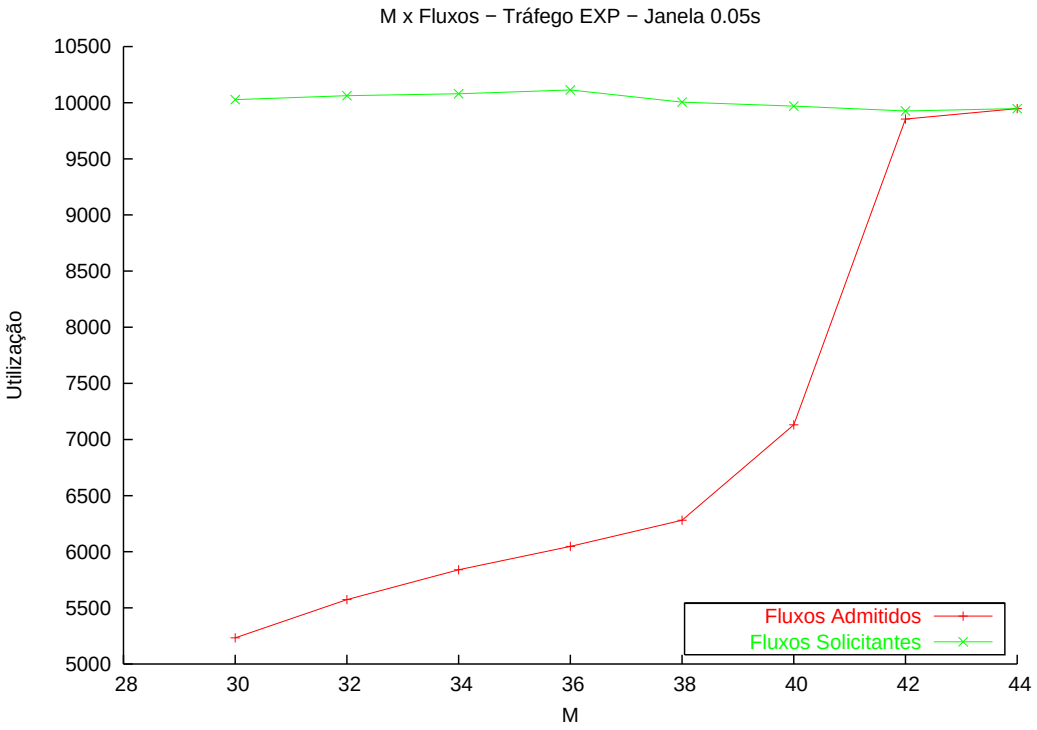

Figure 4.12: Fluxos admitidos pelo GRIP com tráfego EXP - Topologia 1

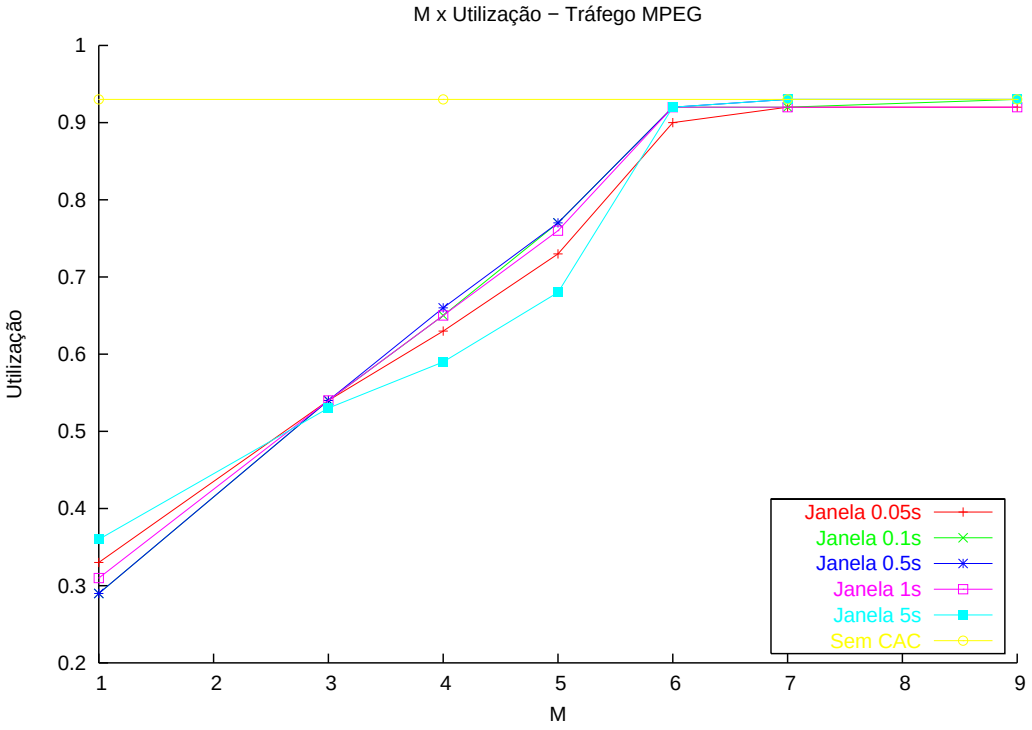

Figure 4.13: GRIP-Utilização do enlace com tráfego MPEG - Topologia 1

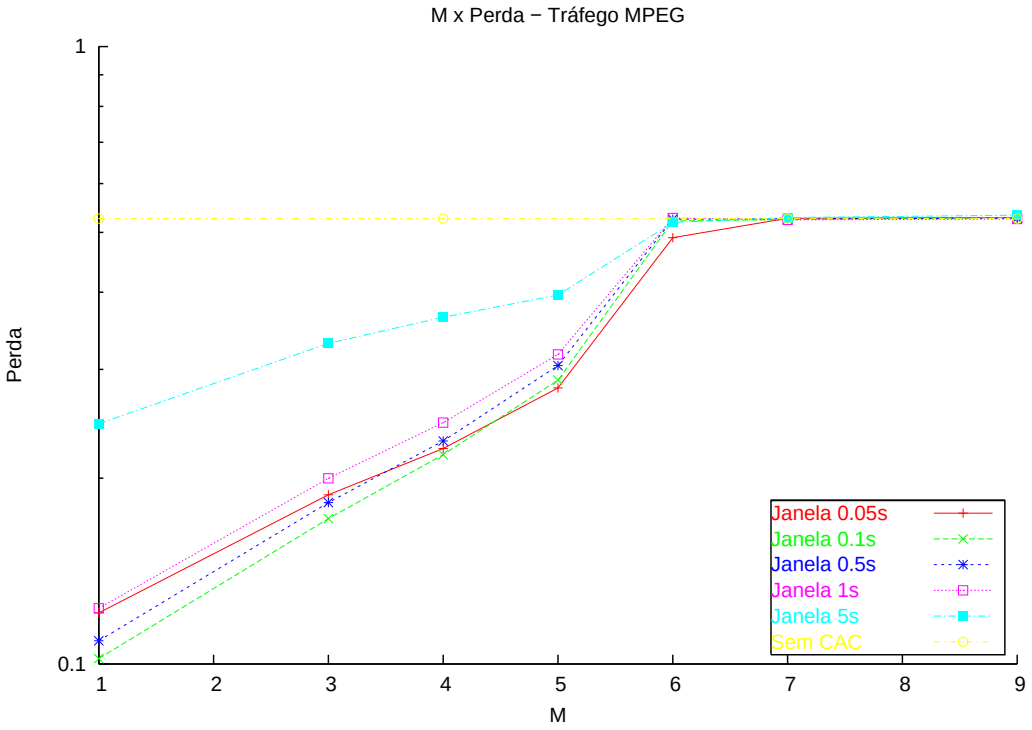

Figure 4.14: GRIP-Taxa de perda de pacotes com tráfego MPEG-Topologia 1

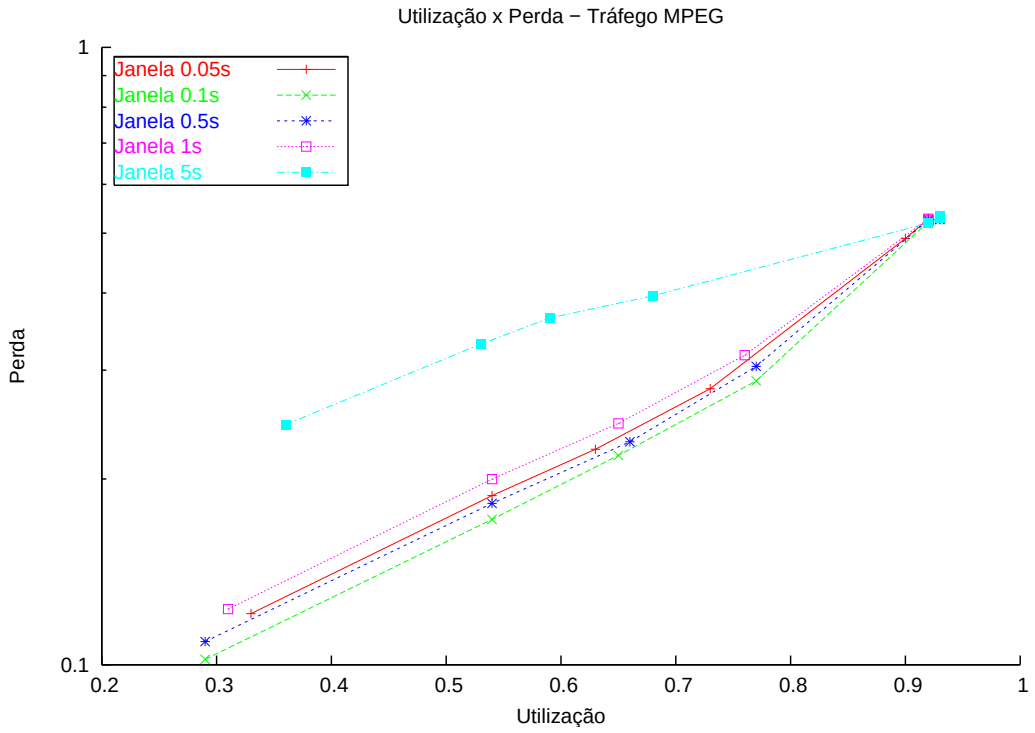

Figure 4.15: Curva perda-carga do GRIP com tráfego MPEG - Topologia 1

Observa-se nas figuras um padrão semelhante ao observado para o tráfego CBR em relação à taxa de perdas. A utilização, entretanto, se mostrou praticamente independente do tamanho da janela. Mas em todos os casos constata-se que as janelas menores são mais adequadas, pois obtém a melhor relação perda vs. carga em todos os casos.

A existência de um valor ótimo do tamanho da janela é sugerida pelos gráficos da Figura 4.15, onde o desempenho melhora à medida que a janela diminui, a partir da janela de 5 segundos. No entanto, o desempenho piora quando a janela diminui de 0.1 segundo para 0.05 segundo. Esse comportamento confirma o que foi dito anteriormente com relação ao tamanho das janelas. Quando o período de amostragem é muito curto, o Controle de Admissão se torna mais suscetível às variações de tráfego. Essa característica se manifestou também no caso CBR, porém com menos intensidade, como pode ser visto na Figura 4.7.

Após essas simulações, buscamos um cenário mais complexo, ilustrado na Figura 4.16, em que dois nós de entrada compartilham um enlace de gargalo em direção ao nó de saída. Todos os nós possuem buffers de 10 pacotes e os enlaces possuem capacidade de 2Mbps. Os nós E1, E2 e C1 possuem módulos de medição configurados com o mesmo limiar (Th), e realizam a transição entre os estados ADMISSÃO e REJEIÇÃO independentemente. Como todos os enlaces têm a mesma capacidade, o enlace C1-E3 alcançará o limiar de utilização

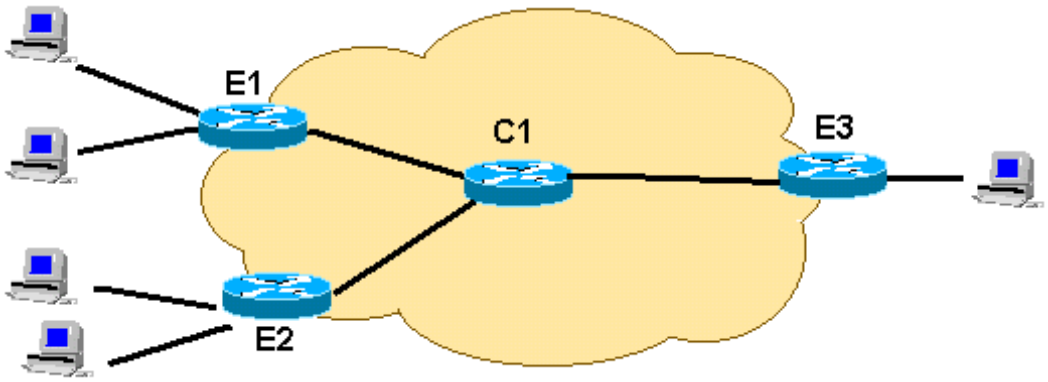

Figure 4.16: Topologia com dois nós de entrada

bem antes dos outros enlaces, logo, caberá ao nó C1 detectar a iminência de congestionamento e bloquear o caminho, evitando assim a admissão de novos fluxos.

As fontes de tráfego conectadas aos nós E1 e E2 possuem exatamente as mesmas características apresentadas anteriormente. Ao contrário da topologia 1, em que o enlace E1-C1 é o gargalo, na topologia 2 o gargalo é o enlace C1-E3, pois este recebe o tráfego proveniente do dois outros links. Logo, este é o enlace que deve ser monitorado. Os resultados aqui apresentados foram obtidos através de medições no nó C1 e no enlace C1-E3. As Figuras 4.17 a 4.25 trazem os resultados das simulações do GRIP na topologia 2.

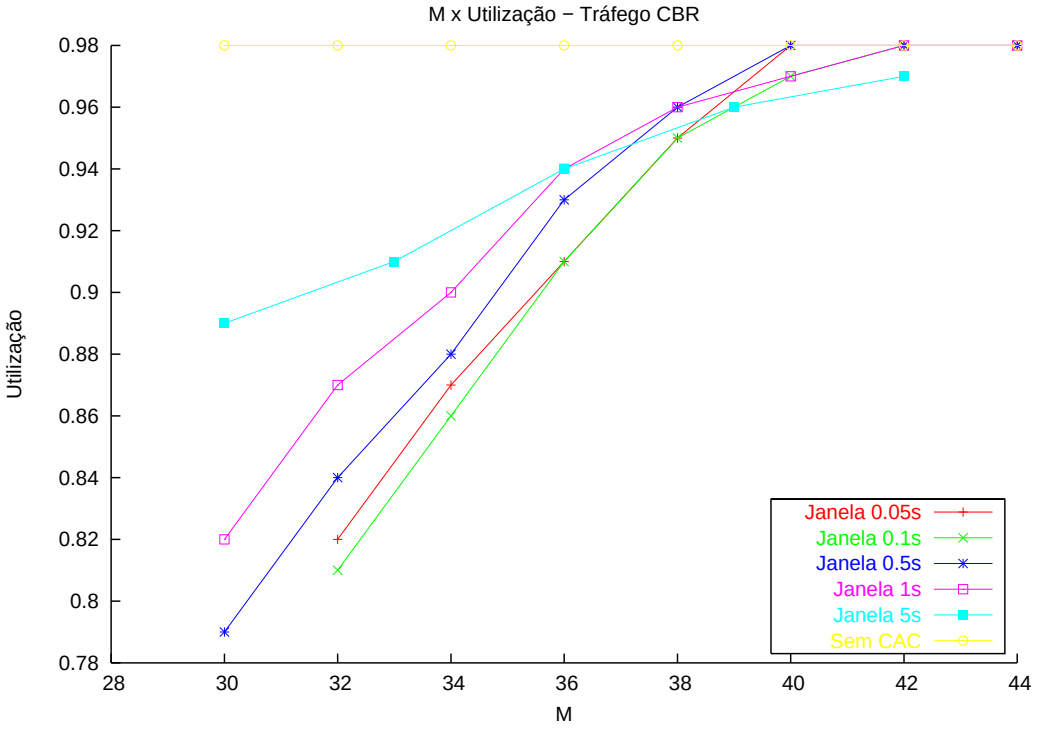

Figure 4.17: GRIP-Utilização do enlace com tráfego CBR - Topologia 2

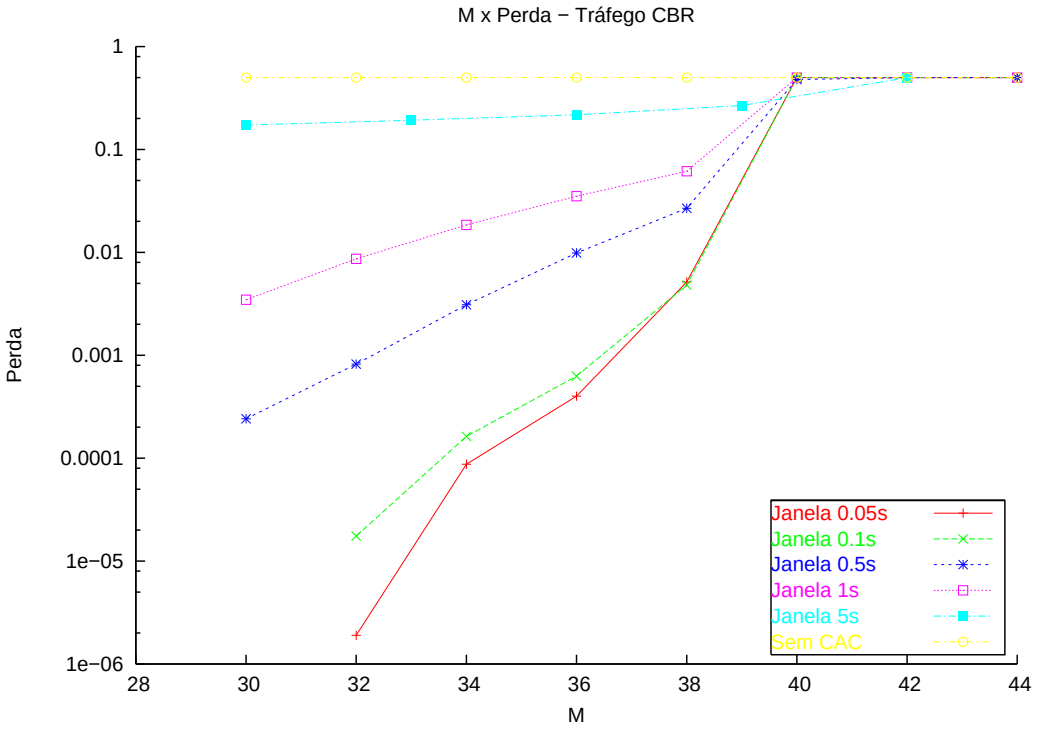

Figure 4.18: GRIP-Taxa de perda de pacotes com tráfego CBR - Topologia 2

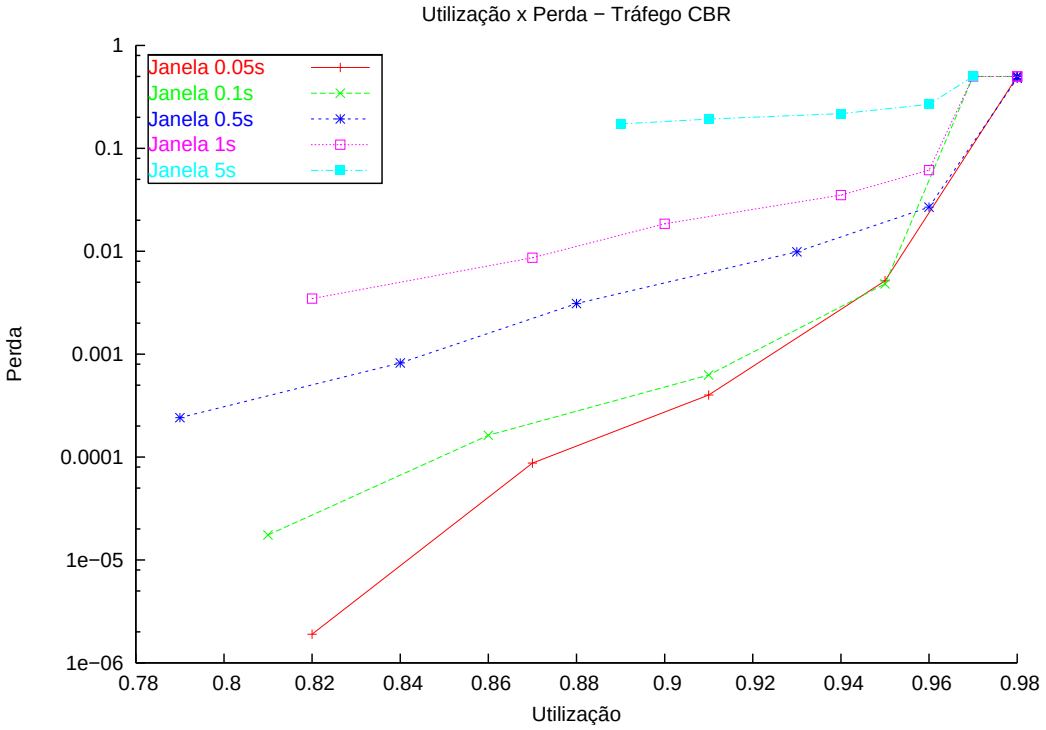

Figure 4.19: Curva perda-carga do GRIP com tráfego CBR - Topologia 2

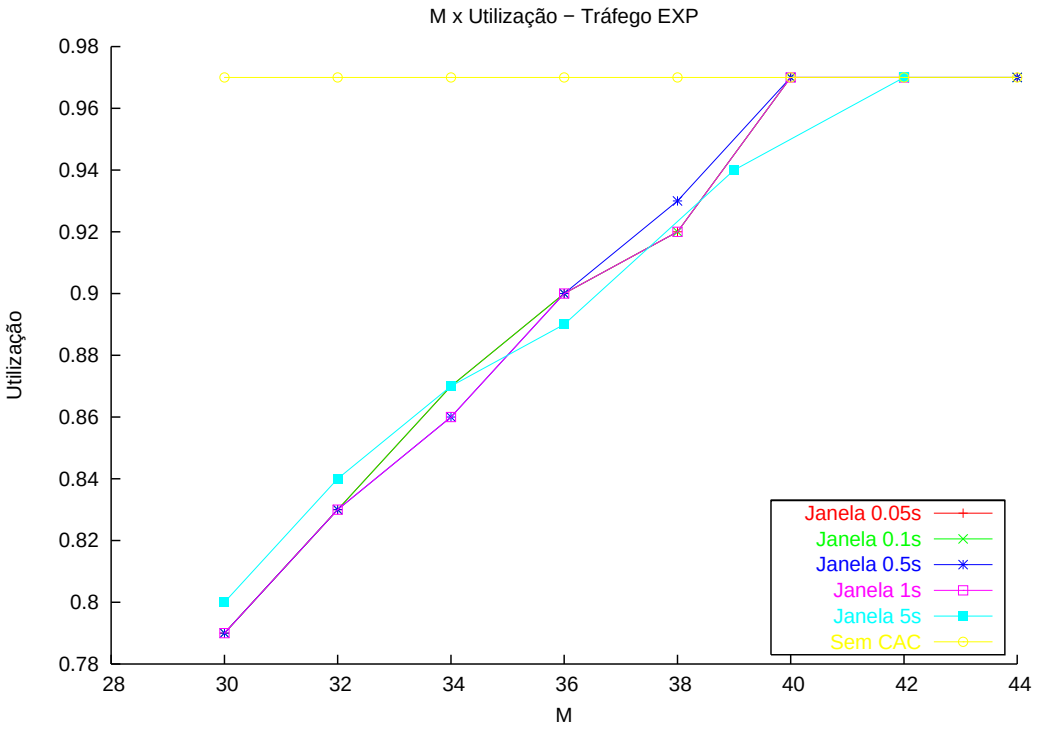

Figure 4.20: GRIP-Utilização do enlace com tráfego EXP - Topologia 2

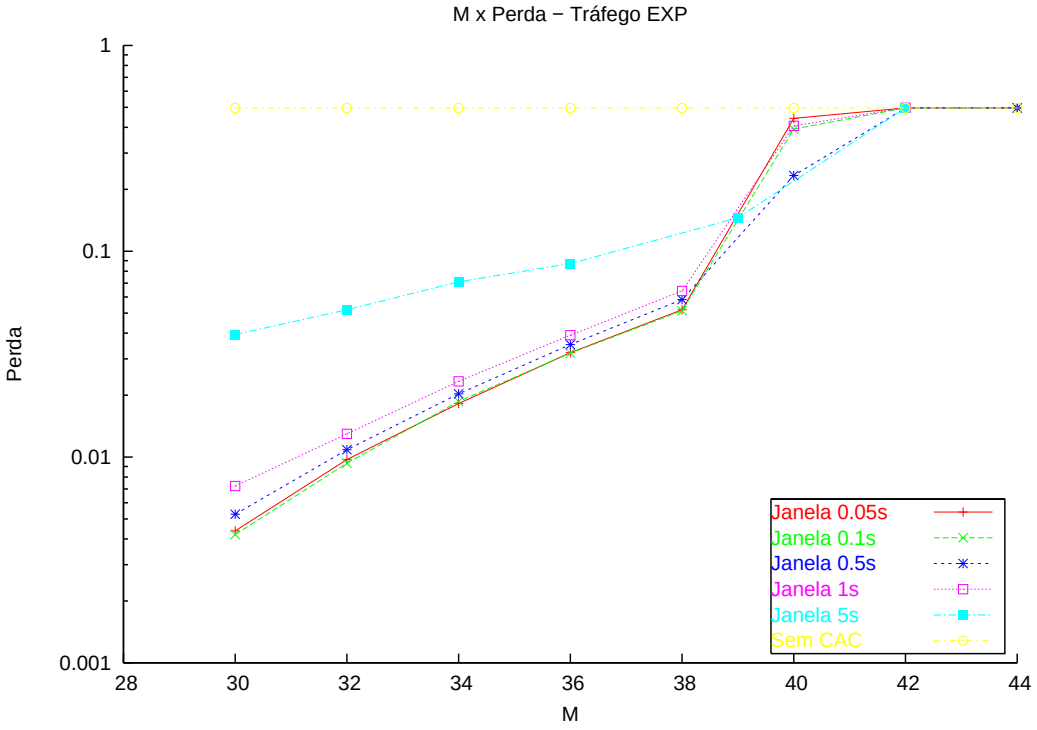

Figure 4.21: GRIP-Taxa de perda de pacotes com tráfego EXP - Topologia 2

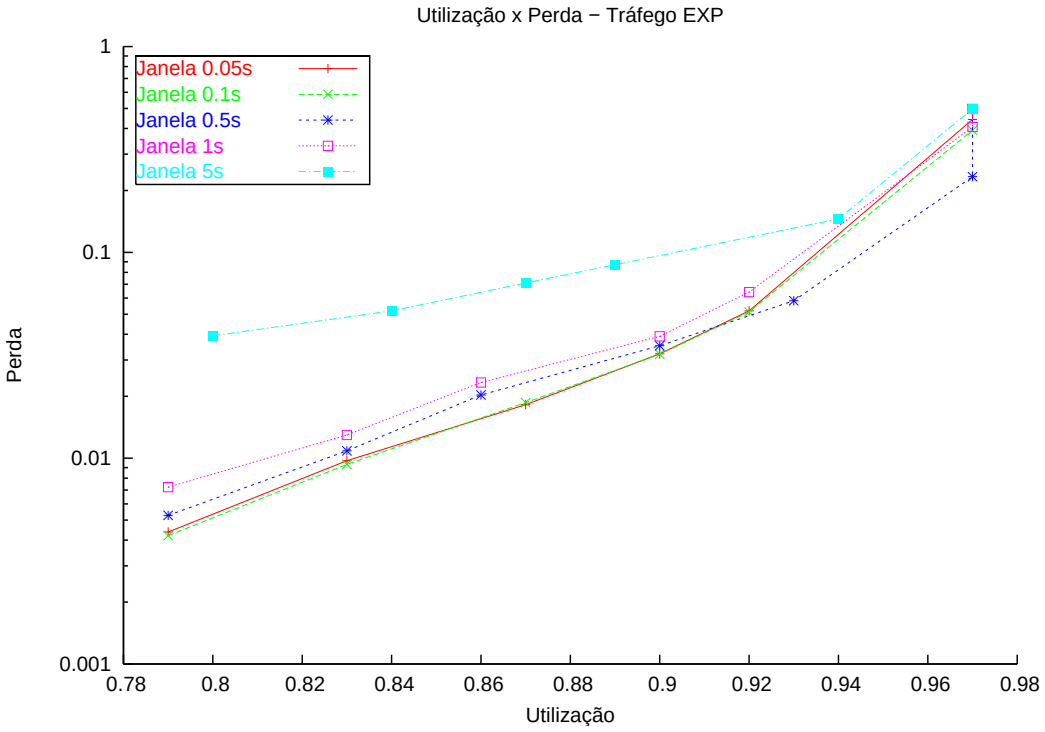

Figure 4.22: Curva perda-carga do GRIP com tráfego EXP - Topologia 2

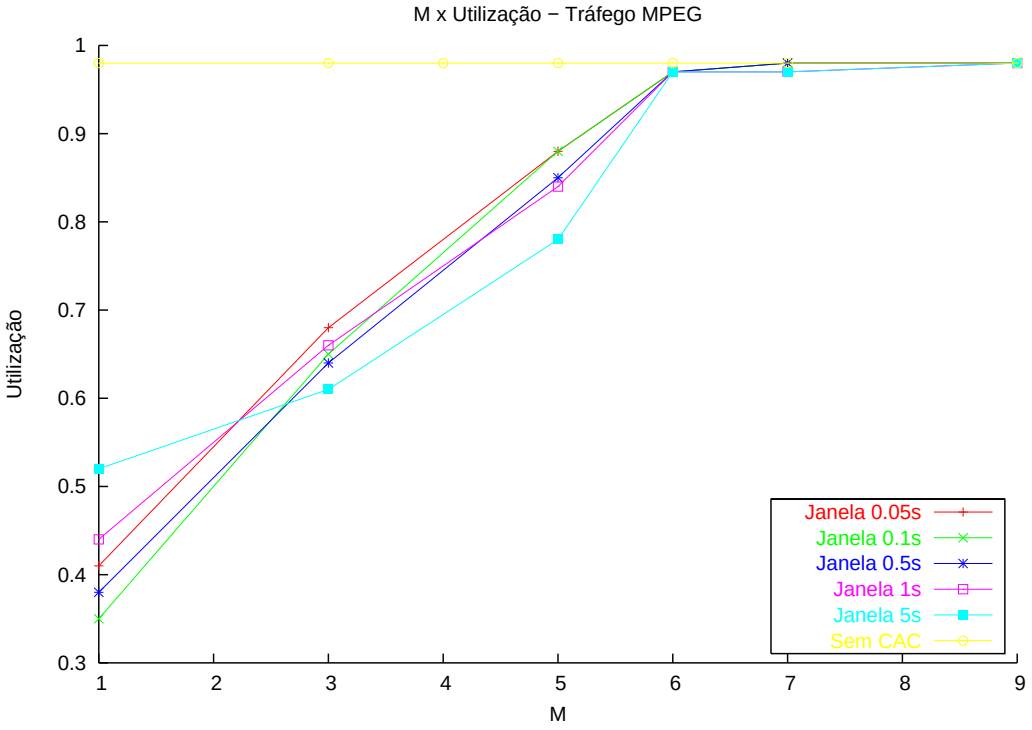

Figure 4.23: GRIP-Utilização do enlace com tráfego MPEG - Topologia 2

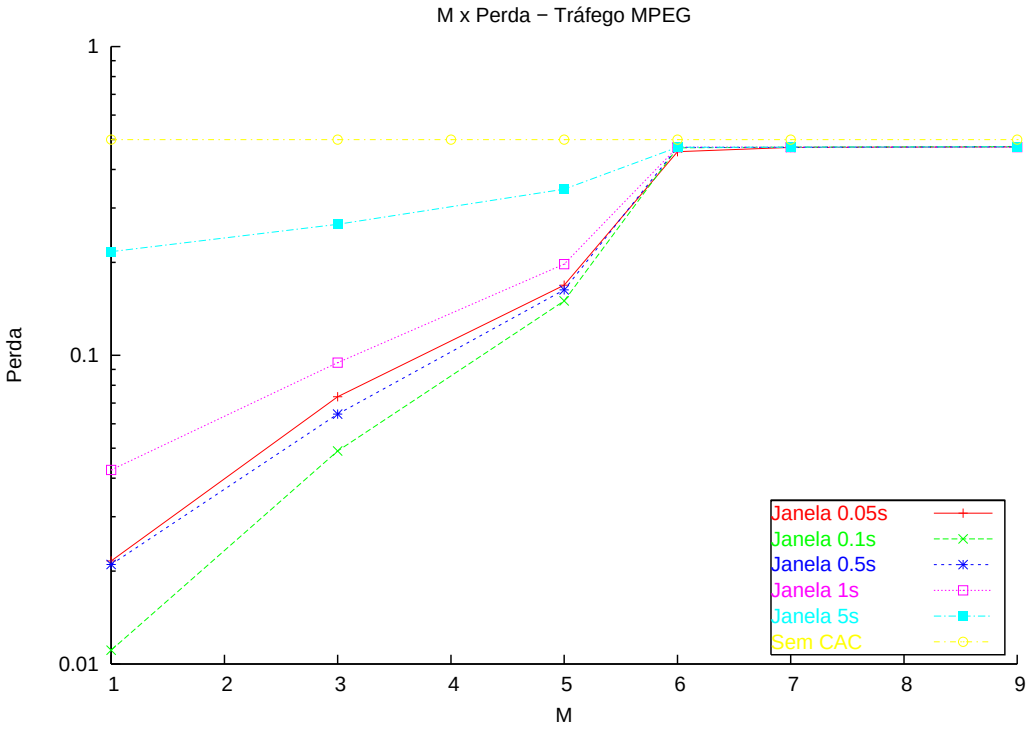

Figure 4.24: GRIP-Taxa de perda de pacotes com tráfego MPEG-Topologia 2

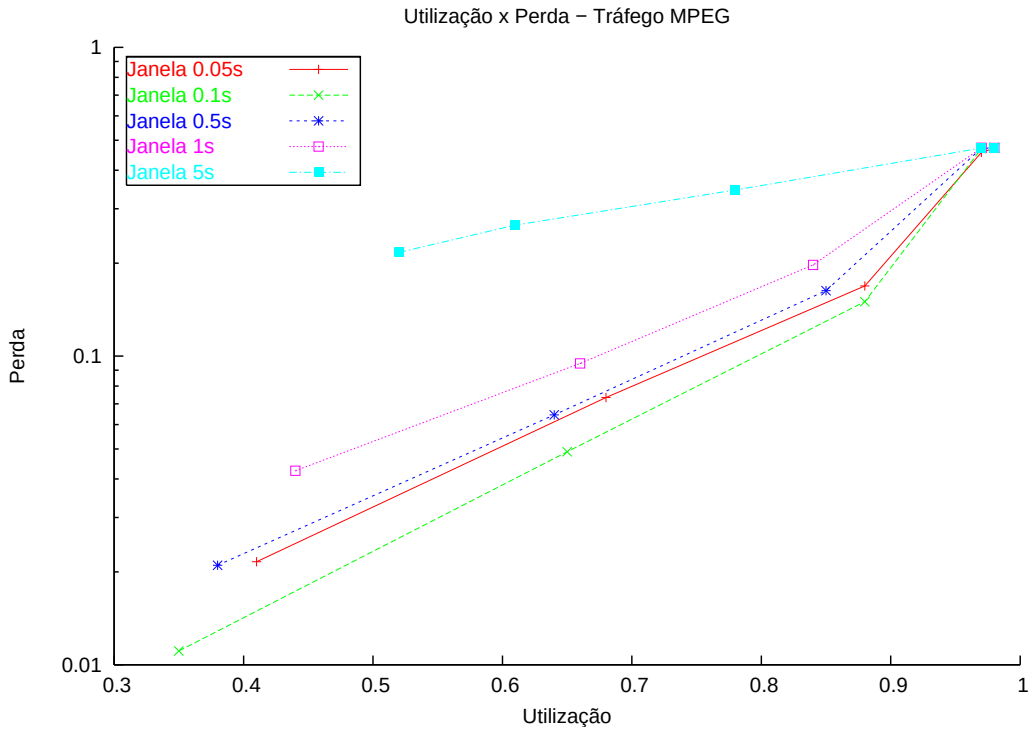

Figure 4.25: Curva perda-carga do GRIP com tráfego MPEG - Topologia 2

Podemos observar algumas características em relação aos resultados obtidos para a topologia 1. A maior modificação no comportamento das curvas perda-carga foi no caso do tráfego CBR. A curva da janela 0.05 segundo permaneceu praticamente inalterada, enquanto todas as outras tiveram um desempenho inferior.

Nos tráfegos EXP e MPEG, o comportamento das curvas não se modifica sensivelmente, entretanto, os valores de taxa de perda são bem maiores.

4.2 Token

O segundo algoritmo a ser analisado utiliza um princípio bem diferente do anterior. Como no GRIP, cada nó do domínio possui um mecanismo de medição e auto-classificação em estado de admissão ou rejeição. No entanto, no algoritmo Token cada nó tem acesso a informações sobre os demais componentes do domínio, e toma a decisão de admissão baseado nesas informações, não sendo, portanto, necessária a sondagem através do pacote de probe.

4.2.1 Descrição do algoritmo

Como foi abordado em capítulos anteriores, a grande dificuldade de se implantar Controle de Admissão num ambiente DiffServ está no mecanismo de sinalização implícita a ser utilizado para tomada da decisão de admissão. Portanto, os mecanismos propostos buscam meios de prover aos nós de borda a informação necessária à decisão da maneira mais eficiente possível sem transgressão do paradigma DiffServ. Muitos desses mecanismos adotam o processo de "*probing*" como recurso de sinalização implícita. É o caso do GRIP.

O algoritmo proposto em [47],[48], traz um processo de admissão bastante diferente do "*probing*", baseado na passagem de um único pacote de sinalização entre os nós da rede. A disciplina de serviço desse mecanismo pode ser comparada à das redes *token ring* (padrão IEEE 802.5), em que cada nó da rede só pode transmitir quando está de posse do *token*.

O algoritmo propriamente dito não é direcionado à arquitetura DiffServ, mas a redes *core-stateless*, ou seja, sem controle do estado dos fluxos nos roteadores de núcleo, porém com reserva de banda e manutenção de estados de fluxos nos nós de borda, o que não acontece no DiffServ. No entanto, como o processo de admissão não gera nenhum tráfego adicional na rede (apenas o próprio *token* circulando), fizemos algumas alterações no algoritmo original para adaptá-lo ao paradigma DiffServ.

O mecanismo proposto em [47],[48], daqui por diante chamado simplesmente de Token, considera a utilização de um protocolo de roteamento como o OSPF ([33]), por exemplo, que permita aos nós de borda conhecer a topologia do domínio. Nesse algoritmo, o *token* circula numa ordem pré-determinada entre todos os roteadores de borda, carregando informações sobre a banda disponível em cada enlace do domínio. Quando chega uma solicitação de conexão a um nó de borda, esta é enfileirada e aguarda a chegada do *token*. Quando ele chega, o Controle de Admissão retira a primeira solicitação da fila e verifica qual o seu modelo de tráfego e quais os enlaces que este fluxo vai percorrer até o nó de destino. Se houver banda disponível em todos os enlaces do caminho, o fluxo é aceito e a banda é reservada, com atualização da tabela do *token*. Esse processo se repete com todas as solicitações que estão enfileiradas, e só depois o *token* é enviado ao nó seguinte.

Por exemplo, suponhamos que, na topologia da Figura 4.16, o nó E1 receba uma solicitação de um usuário que deseja reservar um canal de 2kbps

até o nó E3. Essa solicitação é enfileirada, e quando o *token* chega a E1, apresenta as informações da tabela 4.2.

Enlace	Estado	Banda disponível
$E1 \longrightarrow C1$		2 kbps
$E2 \longrightarrow C1$		3 kbps
$C1 \longrightarrow E3$		7 kbps

Table 4.2: Exemplo de tabela de estados do token

Verificando que há condições de aceitar o fluxo no domínio, o nó E1 aceita o fluxo e atualiza o *token*, reservando a banda solicitada (tabela 4.3), e encaminhando essa informação aos nós seguintes.

Enlace	Estado	Banda disponível
$E1 \longrightarrow C1$		0
$E2 \longrightarrow C1$		3 kbps
$C1 \longrightarrow E3$		5 kbps

Table 4.3: Exemplo de tabela de estados do token atualizada

Quando o usuário terminar de utilizar o serviço, enviará ao nó E1 uma informação de término. O nó E1, então, atualizará o *token* na próxima passagem deste, disponibilizando novamente os 2 kbps que haviam sido reservados nos enlaces $E1 \longrightarrow C1$ e $C1 \longrightarrow E3$.

Em [48], o autor propõe algumas modificações em relação ao esquema original, objetivando dar maior robustez e escalabilidade ao mecanismo.

Para implementação no ambiente DiffServ, são necessárias algumas modificações à proposta original. Aproveitamos a idéia da admissão de usuários submetida à posse do *token*, porém sem o controle individualizado de fluxos. Quando chega uma solicitação a um nó de borda, ele a enfileira, como no processo descrito acima. Quando o *token* chega, o nó verifica a disponibilidade dos enlaces e toma a decisão de admissão, sem reserva de recursos.

Quanto ao comportamento dos roteadores DS de borda, foi implantado um mecanismo semelhante ao do GRIP: os nó monitoram o volume de tráfego em seus enlaces durante o período definido pela janela de medição e comparam este volume com um limiar pré-definido (Th). Enquanto o volume de tráfego estiver abaixo do limiar, o enlace é considerado em condições de admitir novos fluxos. Caso contrário, o enlace é considerado bloqueado. Os nós de borda inserem no *token* a informação da banda ocupada em seu enlace adjacente. No caso da Figura 4.16 e do *token* da tabela 4.2, o nó E1 atualiza a primeira linha da tabela, o nó E2, a segunda linha, e o nó C1, a terceira. Terminada a janela de medição, o nó insere as informações no token e o encaminha ao próximo nó. Portanto, o tempo de permanência do *token* em cada nó é determinado pela janela de medição.

Quando um nó de borda recebe o *token*, ele estima o número de usuários u_i que podem ser admitidos no enlace i através da expressão

$$u_i = \frac{Th - B_i}{R.\Delta T} \tag{4-4}$$

sendo B_i a banda ocupada no enlace i e $R.\Delta T$ a banda equivalente de cada usuário.

Assim, o nó sabe quantos usuários podem ser admitidos em cada enlace e começa a atender às solicitações que estão enfileiradas. Elas são admitidas se os enlaces no seu caminho admitirem novos usuários. Se houverem mais solicitações do que o número de usuários estimado, são recusadas.

É importante observar que não há reserva de recursos, mas simplesmente uma estimativa da ocupação dos enlaces, pois o nó só insere no *token* as informações referentes aos seus enlaces.

Outra modificação incluída no mecanismo se refere ao papel dos nós de núcleo. Originalmente, apenas os nós de borda inseriam informação no *token*, pois estes controlavam a alocação de recursos através da reserva. Utilizando essa nova abordagem, sem reserva de recursos nem controle individualizado de fluxos, é preciso que cada roteador do domínio realize as medições nos enlaces adjacentes e as atualize no *token*. Para tornar mais clara essa necessidade, tomemos como exemplo a topologia da Figura 4.26. Como há apenas um nó de núcleo, os nós de borda podem suprir o token com informações de todos os enlaces do domínio, não sendo portanto necessária a intervenção do nó de núcleo.

No entanto, numa topologia mais complexa como a da Figura 4.27, os nós E1 e E2 só possuem informações referentes a 2 dos 6 enlaces do domínio (apenas os links **a** e **f**). Nesse caso, poderia ainda ser utilizada uma solução

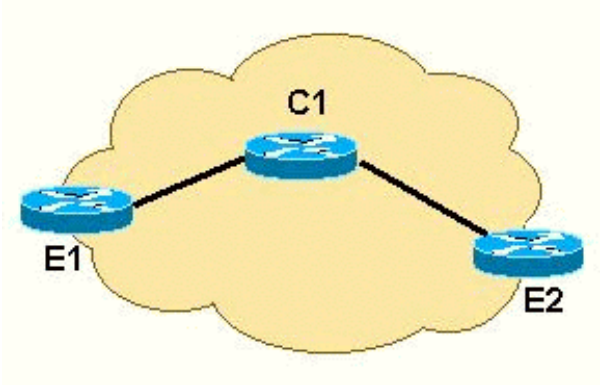

Figure 4.26: Topologia com apenas um nó de núcleo

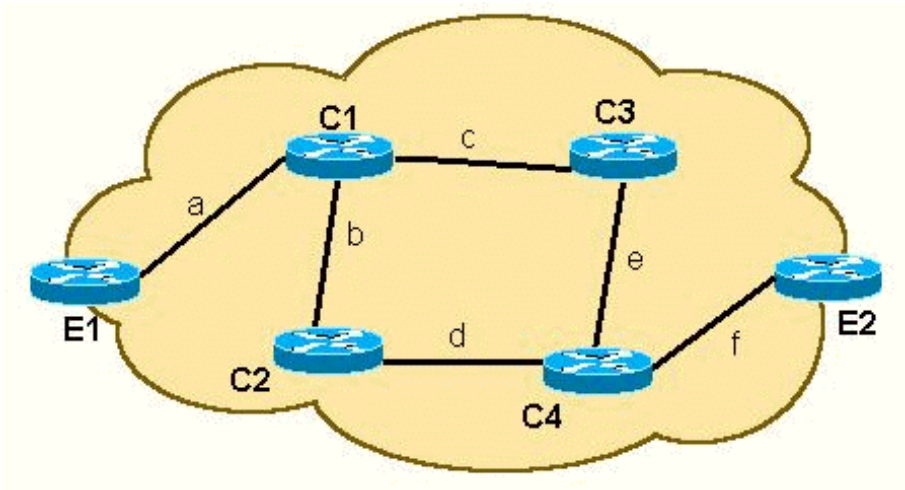

Figure 4.27: Topologia com vários nós de núcleo

como o protocolo OSPF, no entanto o volume de tráfego gerado por essa sinalização comprometeria a escalabilidade do mecanismo. Buscamos então uma solução simples: o *token* circula também entre os nós de núcleo, que inserem as informação referentes ao estado dos seus enlaces (no caso da Figura 4.27, os enlaces **b**, **c**, **d** e **e**).

É importante salientar que nenhuma das medidas implantadas na disciplina do Token representa transgressão ao paradigma DiffServ. Elas apenas permitem que os nós de borda tomem conhecimento das condições dos nós de núcleo sem qualquer mecanismo de sinalização externa.

4.2.2 Avaliação de desempenho

Para avaliação do desempenho do algoritmo, foi utilizada a mesma metodologia da avaliação do GRIP: procuramos determinar a capacidade do algoritmo de controlar o nível de perdas no sistema em situações de congestionamento.

Observamos os mesmos parâmetros de desempenho e utilizamos as mesmas fontes de tráfego, configuradas para impor à rede um volume de tráfego elevado, o que pode ser constatado na comparação dos níveis de serviço do algoritmo com os níveis de serviço obtidos sem a utilização de qualquer mecanismo de Controle de Admissão.

Também foram realizadas simulações para as duas topologias utilizadas anteriormente: a primeira, com apenas um nó de entrada (Topologia 1, ilustrada na Figura 4.3) e a segunda, com dois nós de entrada (Topologia 2, na Figura 4.16).

4.2.3 Resultados

Inicialmente, apresentamos os resultados das simulações do algoritmo Token para a Topologia 1. Utilizando as fontes de tráfego CBR, os resultados obtidos estão expostos nas Figuras 4.28, 4.29, 4.30 e 4.31.

Nas Figuras 4.32 a 4.38 são apresentados os reultados para as simulações com fontes EXP e MPEG.

As observações referentes à comparação do mecanismo Token com o GRIP são feitas na seção 4.3.

Para a Topologia II, os resultados estão nas Figuras 4.39, 4.40, 4.41, 4.42, 4.43, 4.44, 4.45 e 4.46 e 4.47.

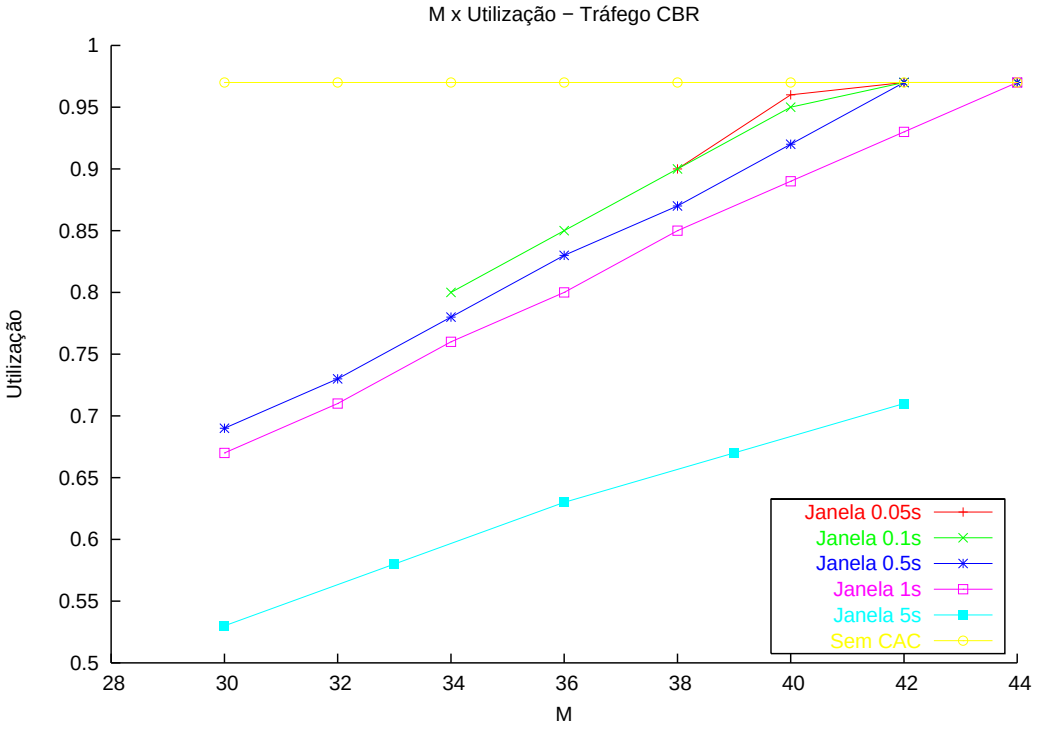

Figure 4.28: Token-Utilização do enlace com tráfego CBR - Topologia 1

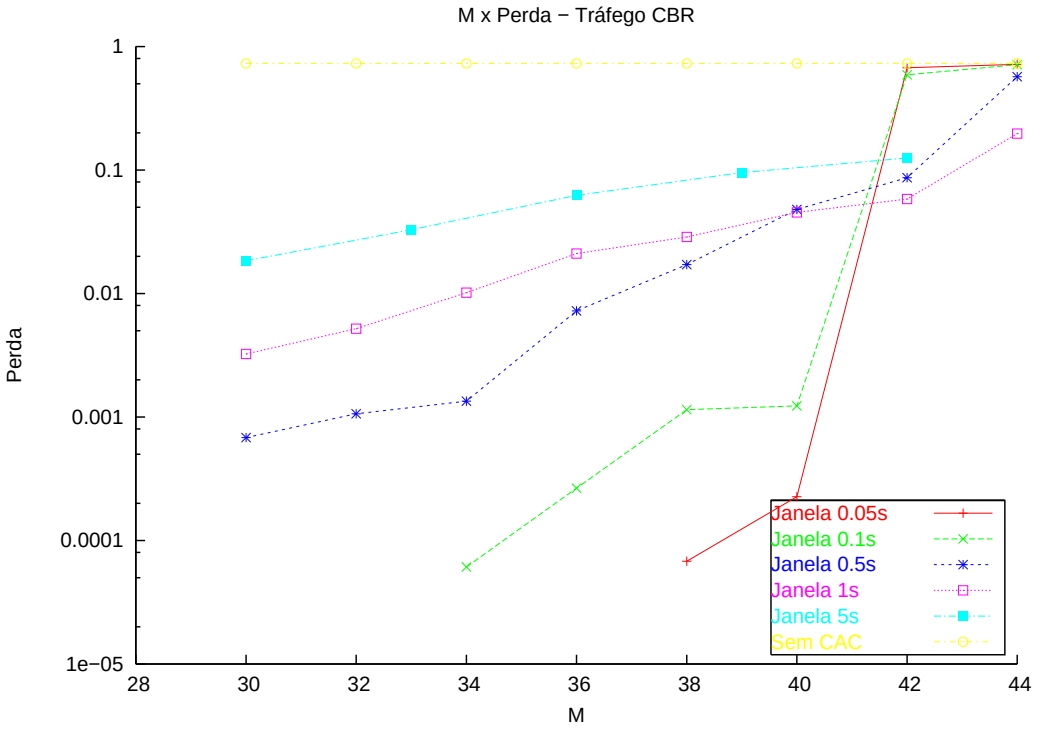

Figure 4.29: Token-Taxa de perda de pacotes com tráfego CBR - Topologia 1

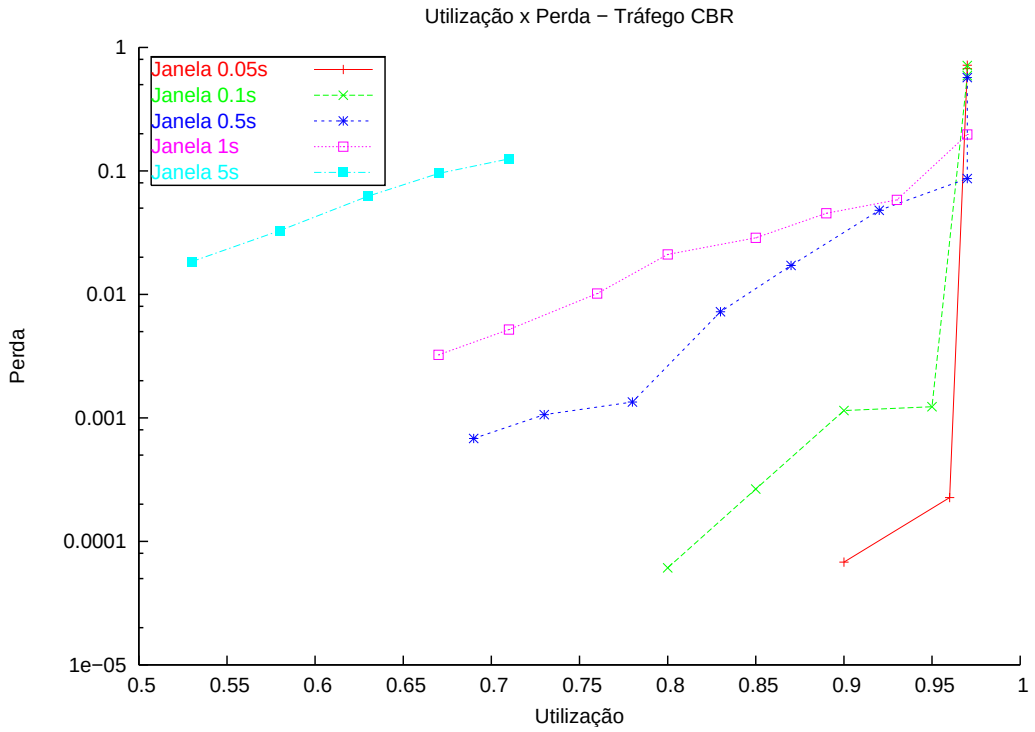

Figure 4.30: Curva perda-carga do Token com tráfego CBR - Topologia 1

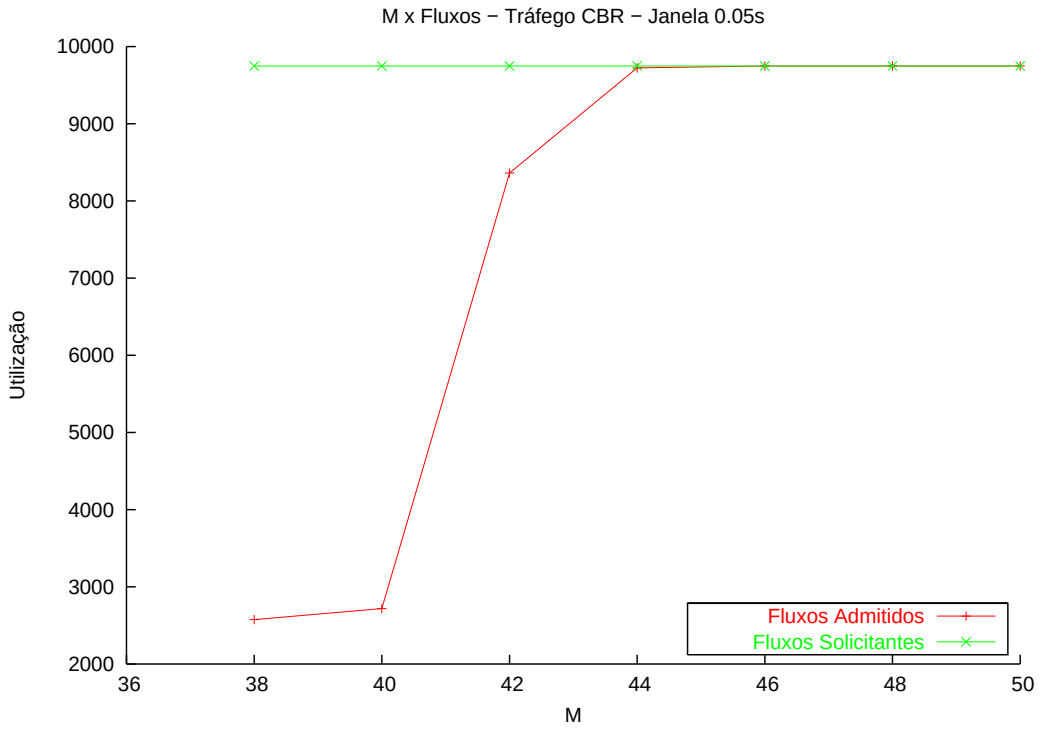

Figure 4.31: Fluxos admitidos pelo Token com tráfego CBR - Topologia 1

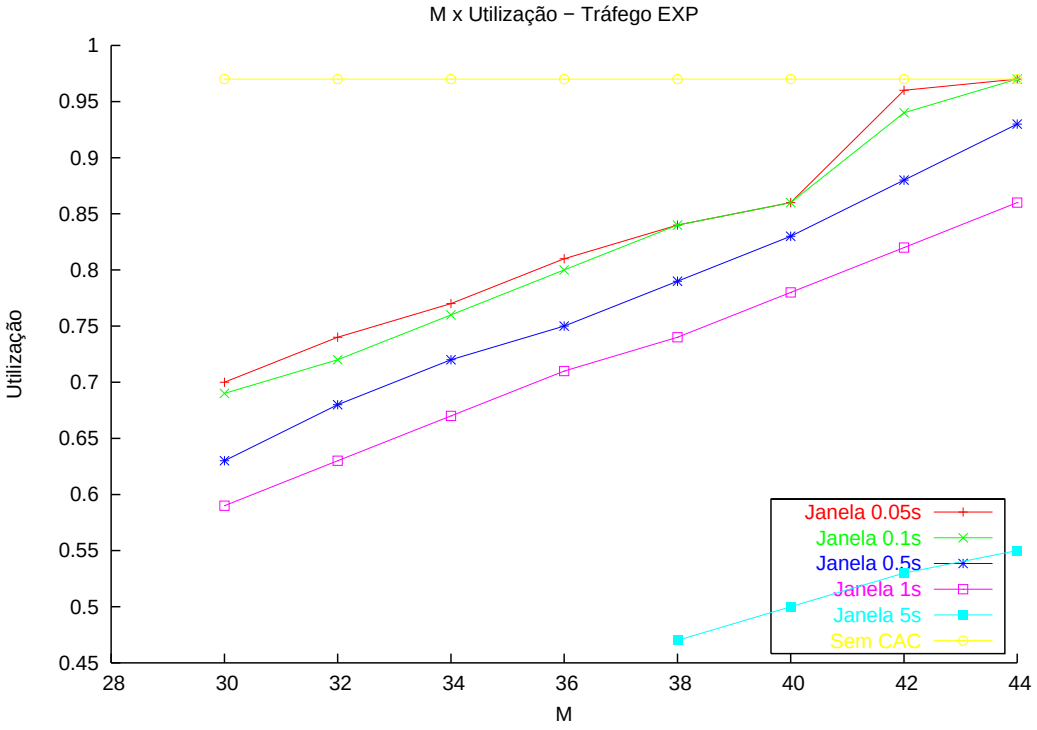

Figure 4.32: Token-Utilização do enlace com tráfego EXP - Topologia 1

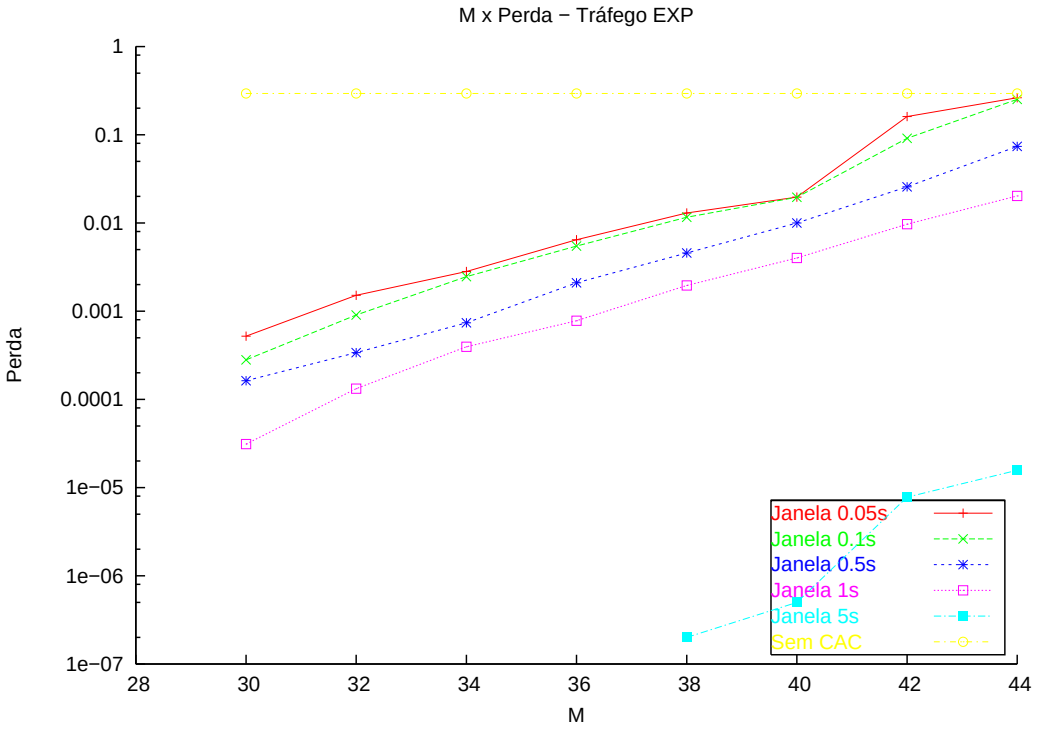

Figure 4.33: Token-Taxa de perda de pacotes com tráfego EXP - Topologia 1

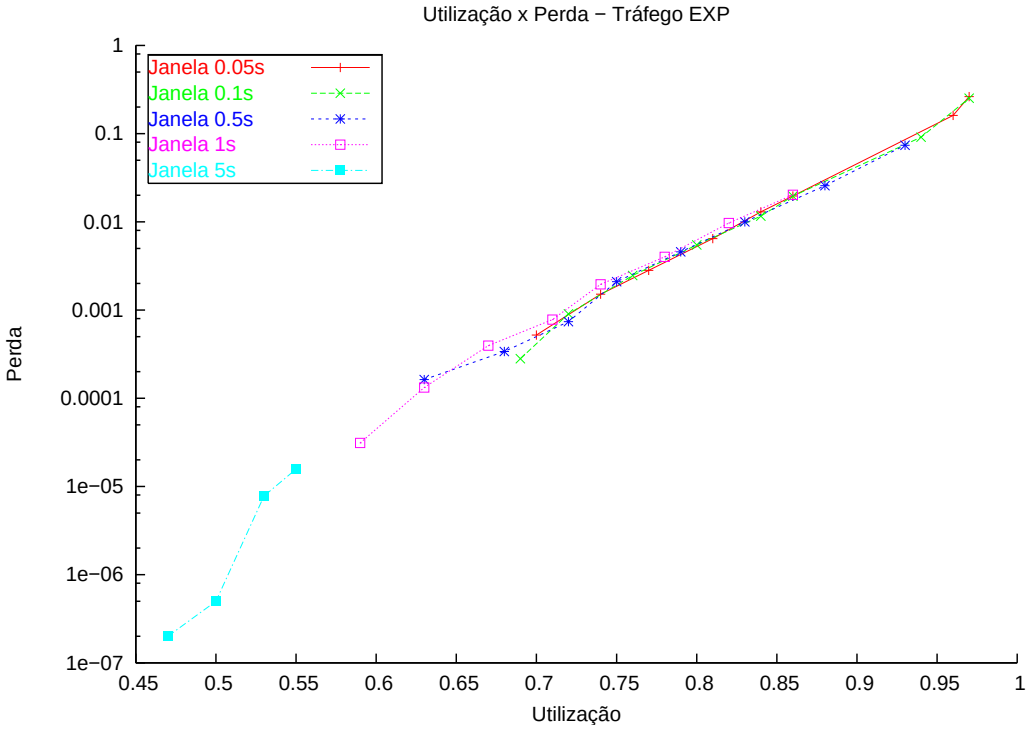

Figure 4.34: Curva perda-carga do Token com tráfego EXP - Topologia 1

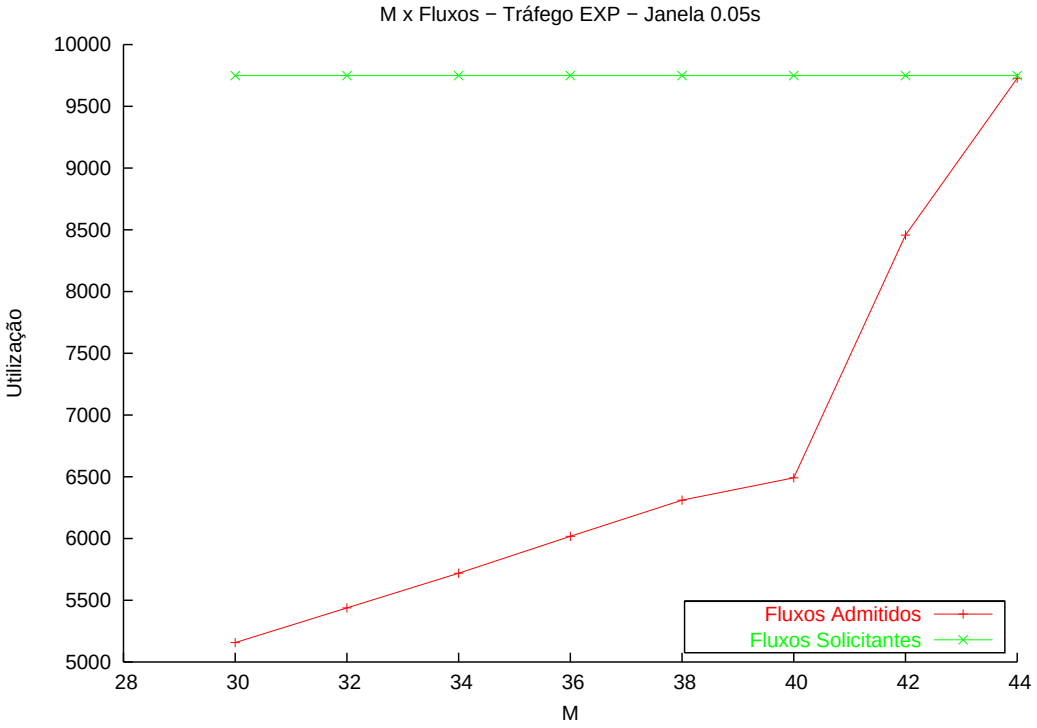

Figure 4.35: Fluxos admitidos pelo Token com tráfego EXP - Topologia 1

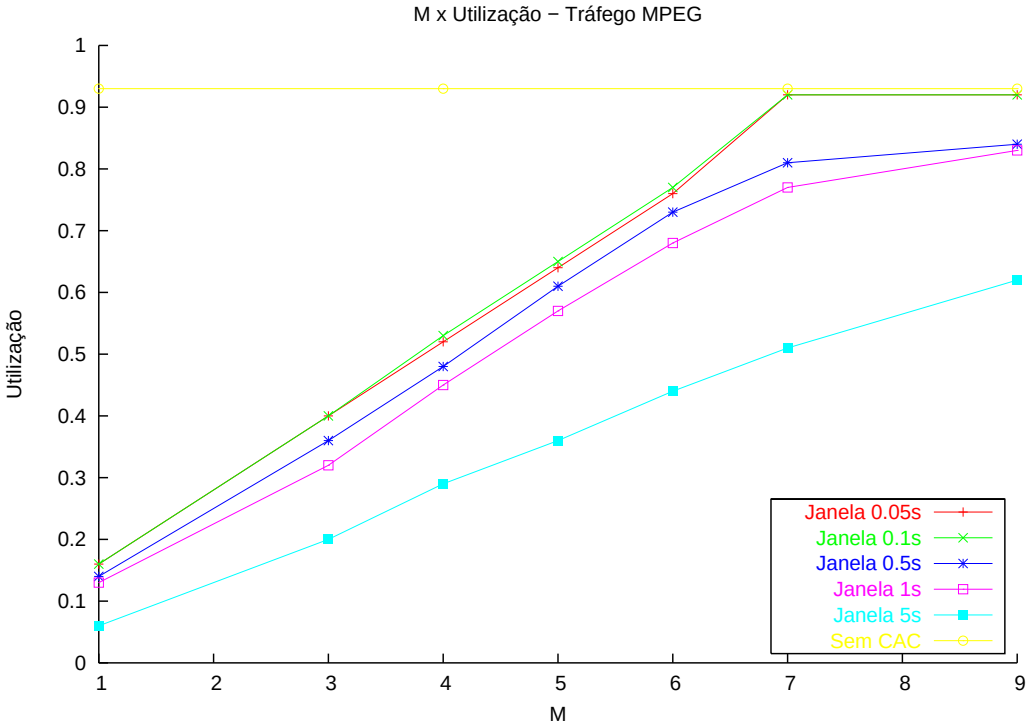

Figure 4.36: Token-Utilização do enlace com tráfego MPEG - Topologia 1

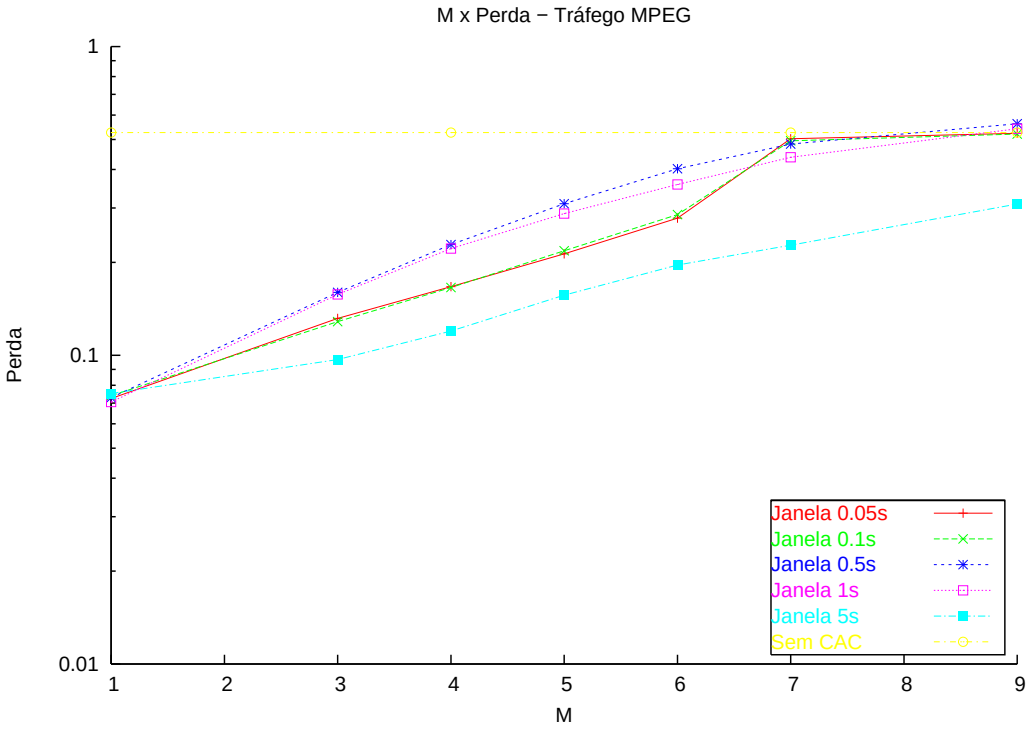

Figure 4.37: Token-Taxa de perda de pacotes com tráfego MPEG-Topologia 1

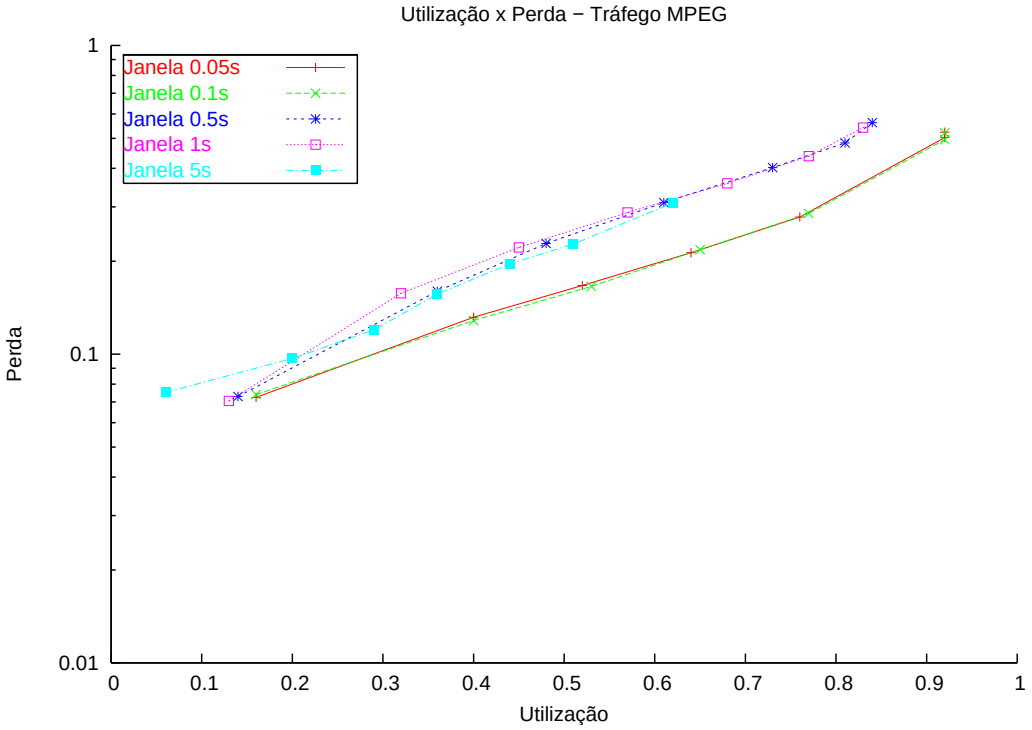

Figure 4.38: Curva perda-carga do Token com tráfego MPEG - Topologia 1

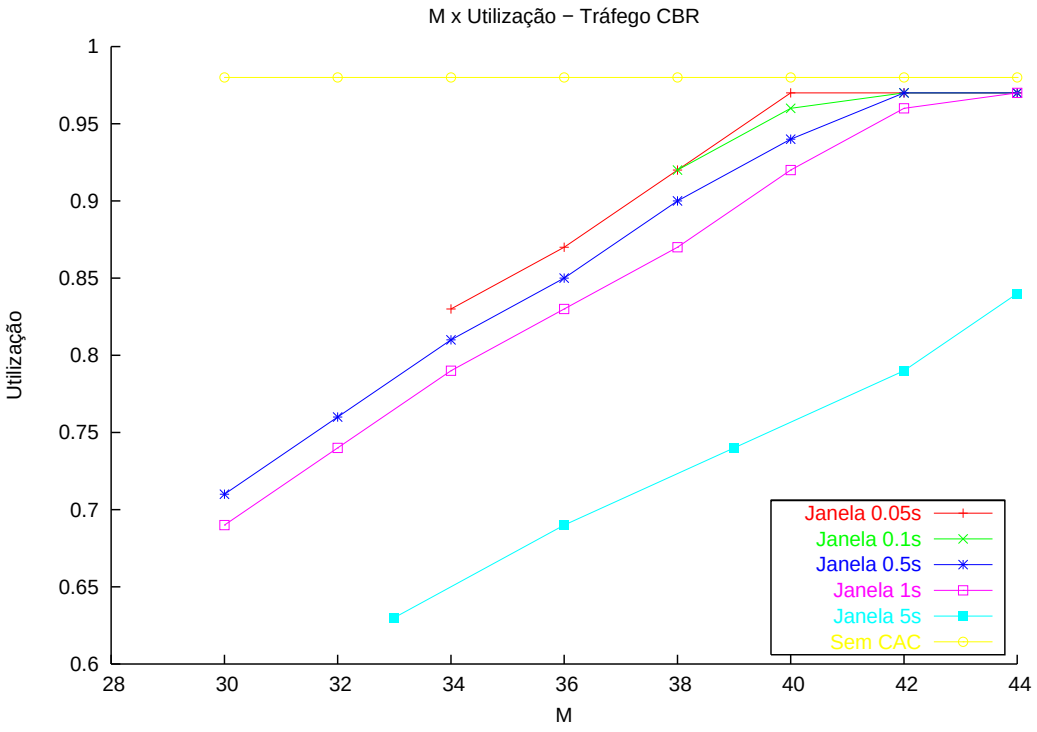

Figure 4.39: Token-Utilização do enlace com tráfego CBR - Topologia 2

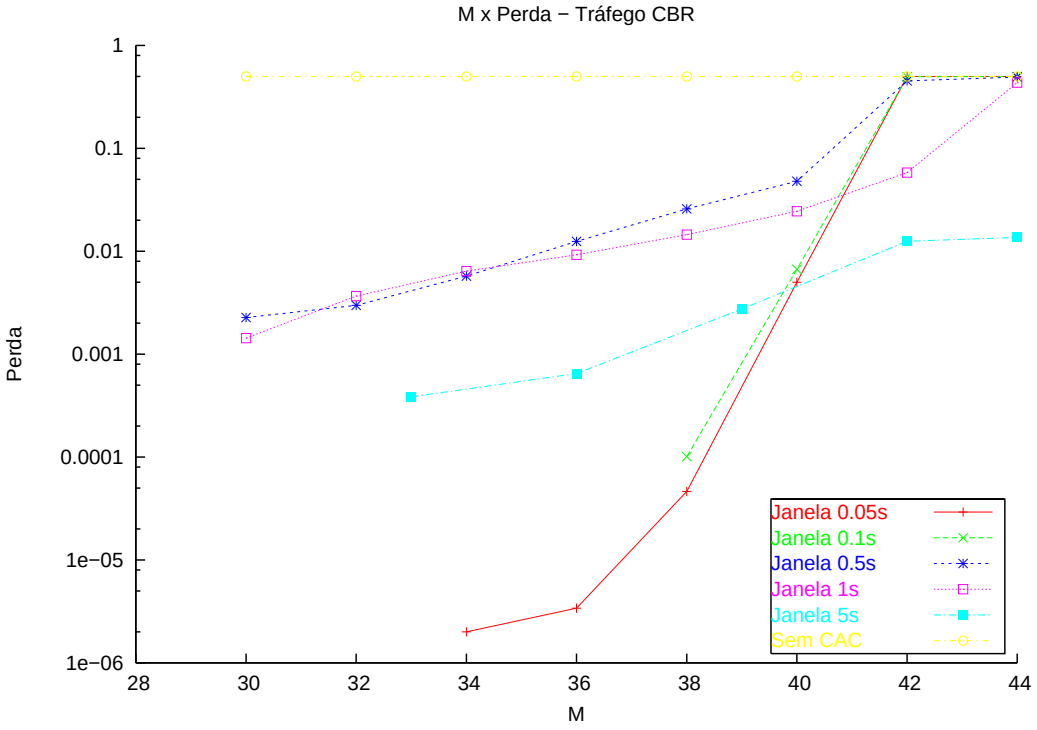

Figure 4.40: Token-Taxa de perda de pacotes com tráfego CBR - Topologia 2

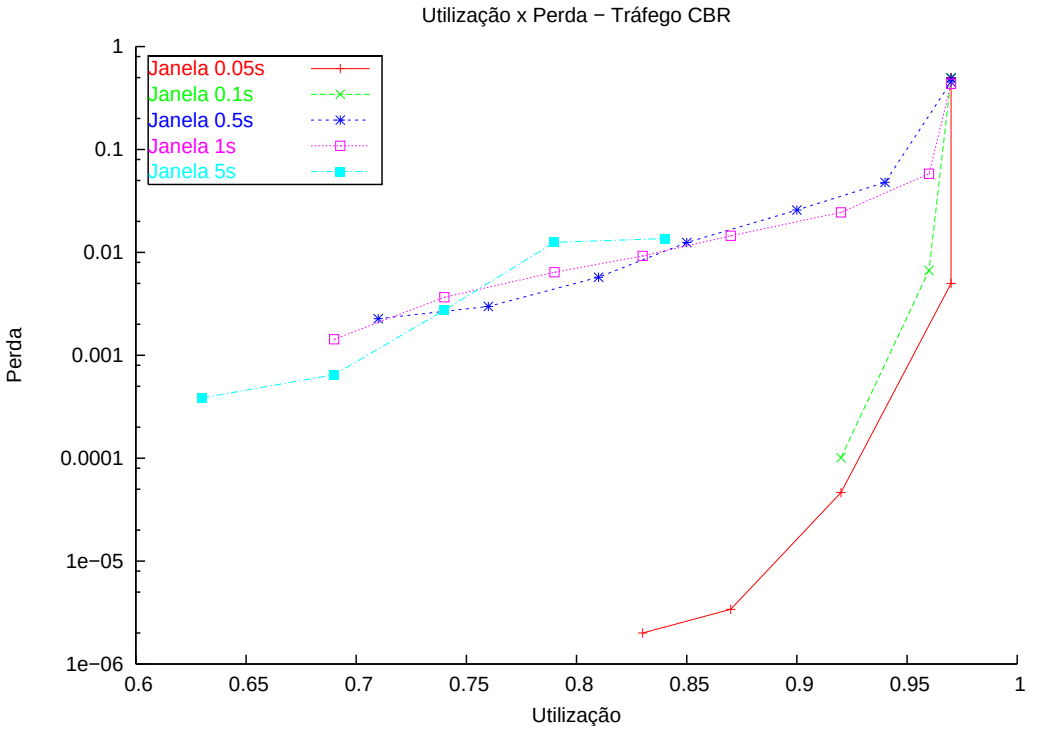

Figure 4.41: Curva perda-carga do Token com tráfego CBR - Topologia 2

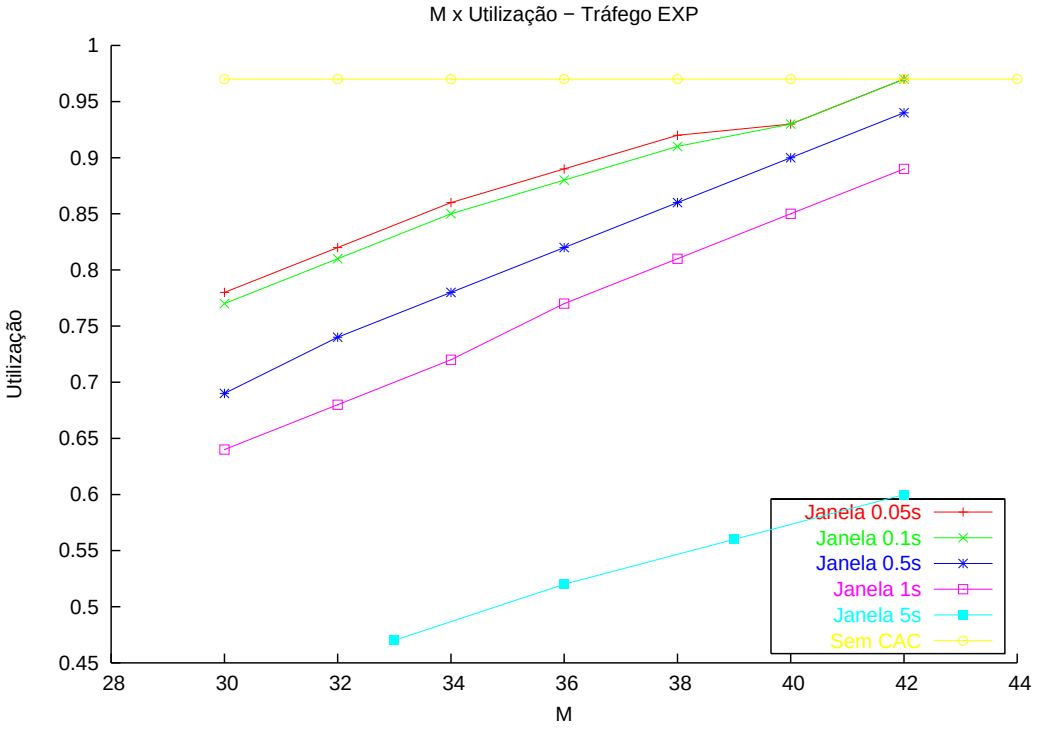

Figure 4.42: Token-Utilização do enlace com tráfego EXP - Topologia 2

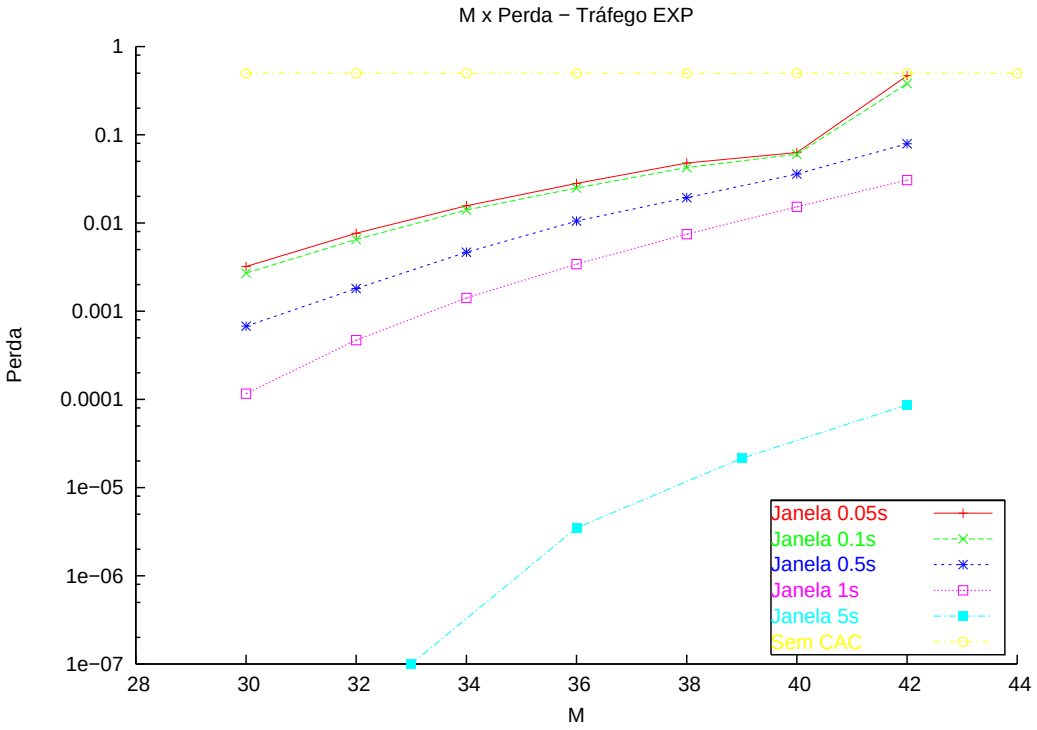

Figure 4.43: Token-Taxa de perda de pacotes com tráfego EXP - Topologia 2

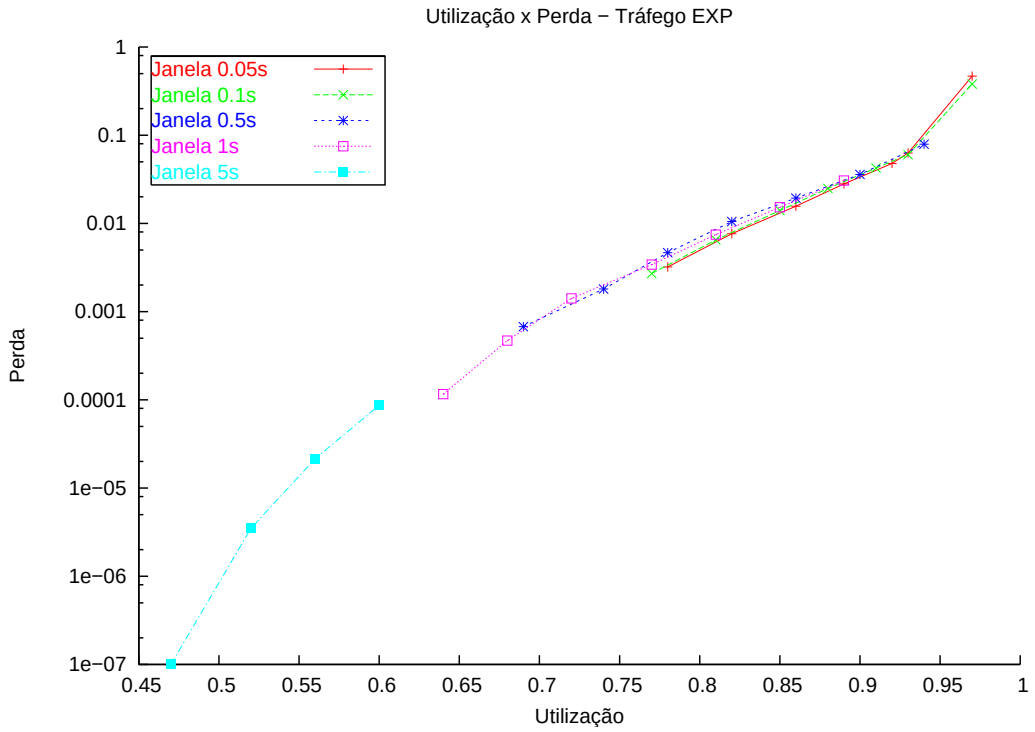

Figure 4.44: Curva perda-carga do Token com tráfego EXP - Topologia 2

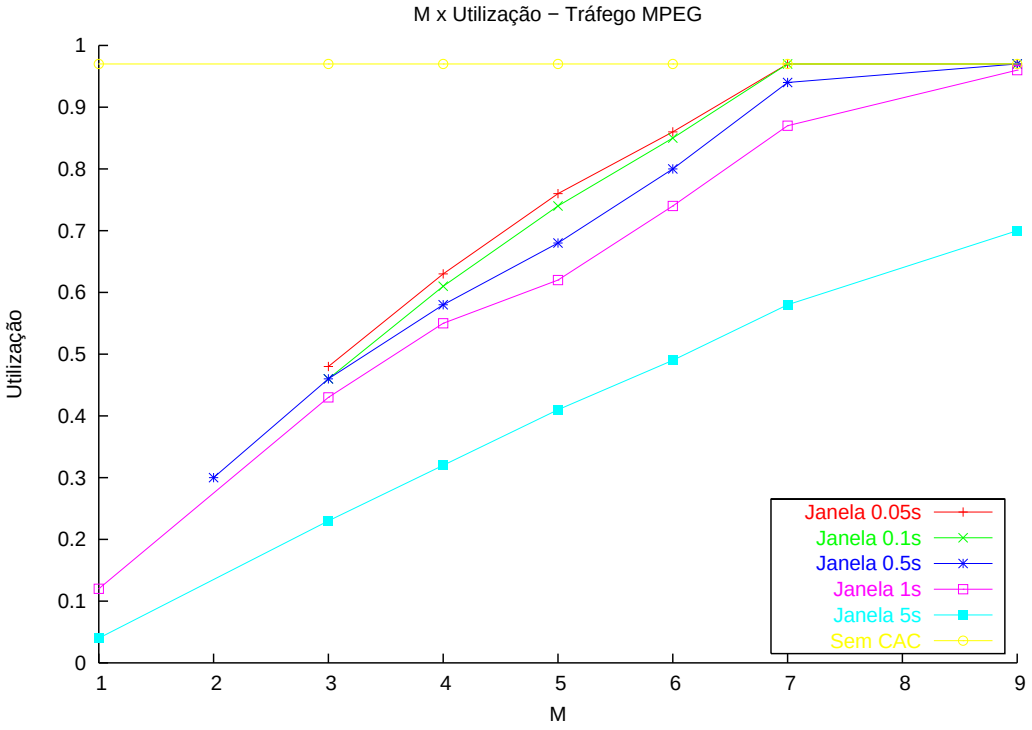

Figure 4.45: Token-Utilização do enlace com tráfego MPEG - Topologia 2

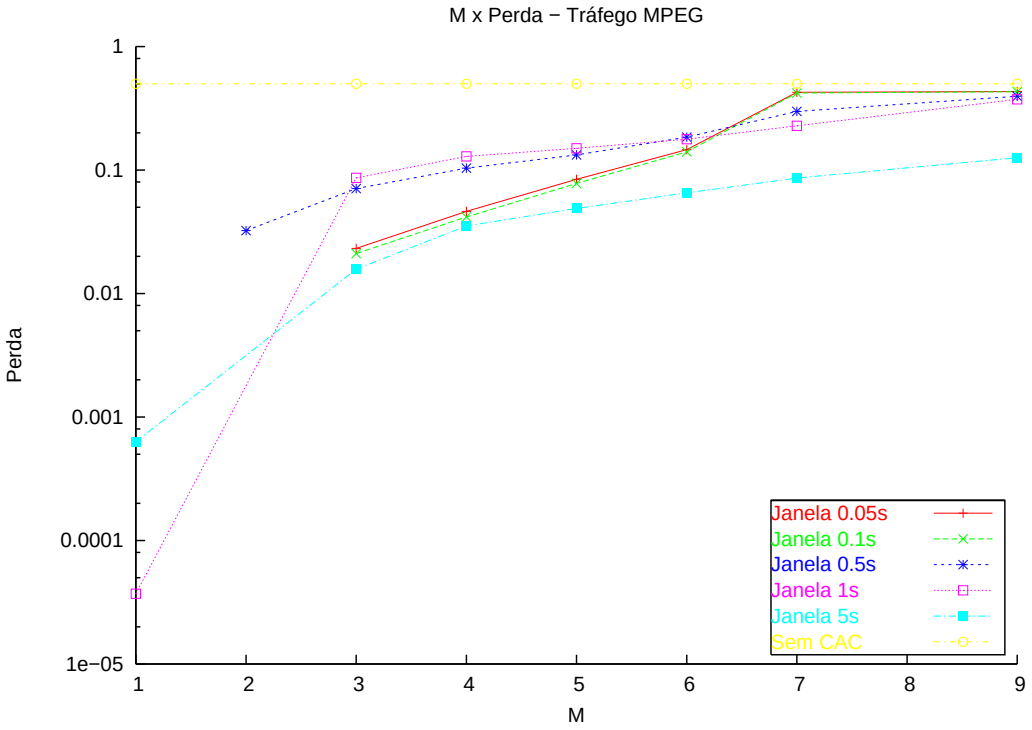

Figure 4.46: Token-Taxa de perda de pacotes com tráfego MPEG-Topologia 2

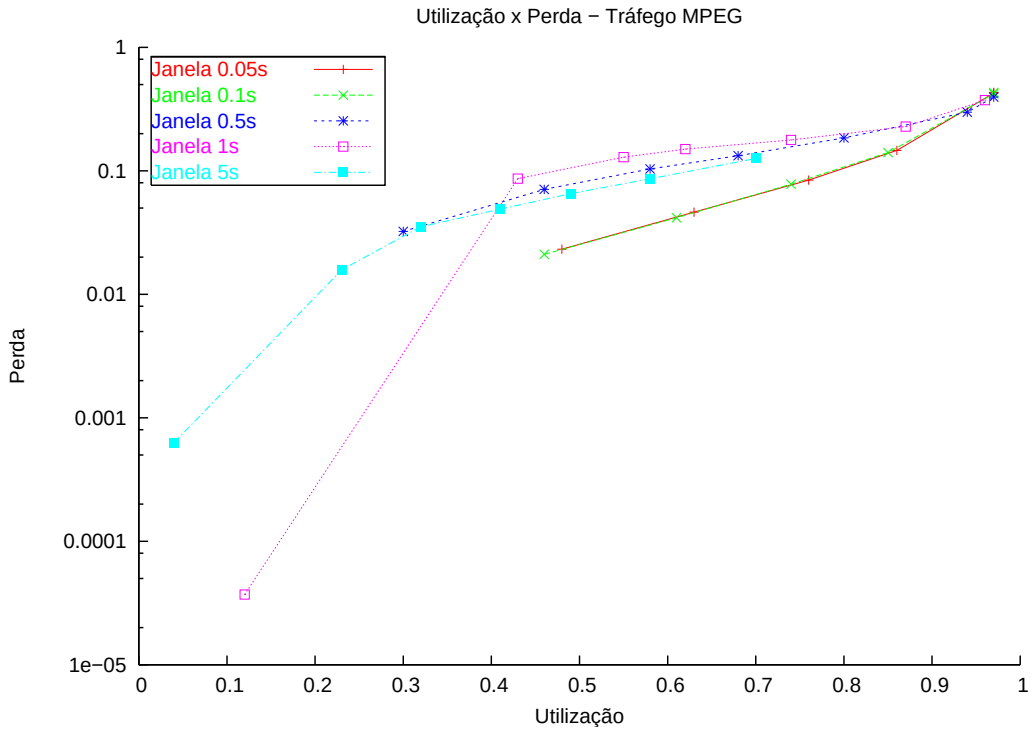

Figure 4.47: Curva perda-carga do Token com tráfego MPEG - Topologia 2

Como nas simulações do GRIP, esses resultados mostraram o desempenho superior das janelas menores em quase todos os casos. Nas simulações com fontes CBR e MPEG, vemos as curvas perda-carga das janelas menores (0.05 segundo e 0.1 segundo) indicando desempenho semelhante entre si e superior às outras janelas. Observando os gráficos de M x Utilização, constatamos que o Token é bem mais sensível ao tamanho da janela do que o GRIP.

No entanto, no caso do tráfego *on-off* exponencial (Figuras 4.34 e 4.44), repetiu-se também o comportamento observado no GRIP: as janelas apresentam um desempenho praticamente idêntico. Vemos nesse caso a curva da janela de 5 segundos em uma faixa de valores bem inferior à dos outros, no entanto, com um comportamento semelhante. Isso se deve à baixa utilização obtida por essa configuração, que é acompanhada de baixa taxa de perda.

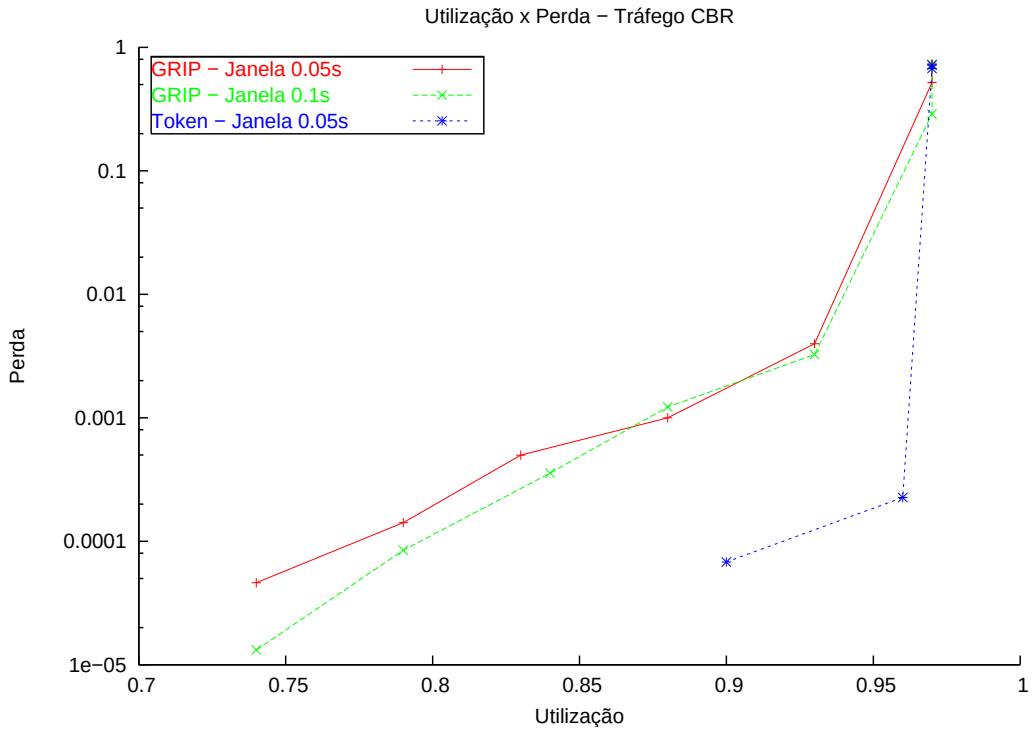

Figure 4.48: Comparação entre GRIP e Token com tráfego CBR utilizando a topologia 1

4.3 Análise comparativa das técnicas

Utilizando a configuração que apresentou os melhores resultados para cada técnica, foram obtidos os resultados expostos a seguir, como comparação entre o GRIP e o Token.

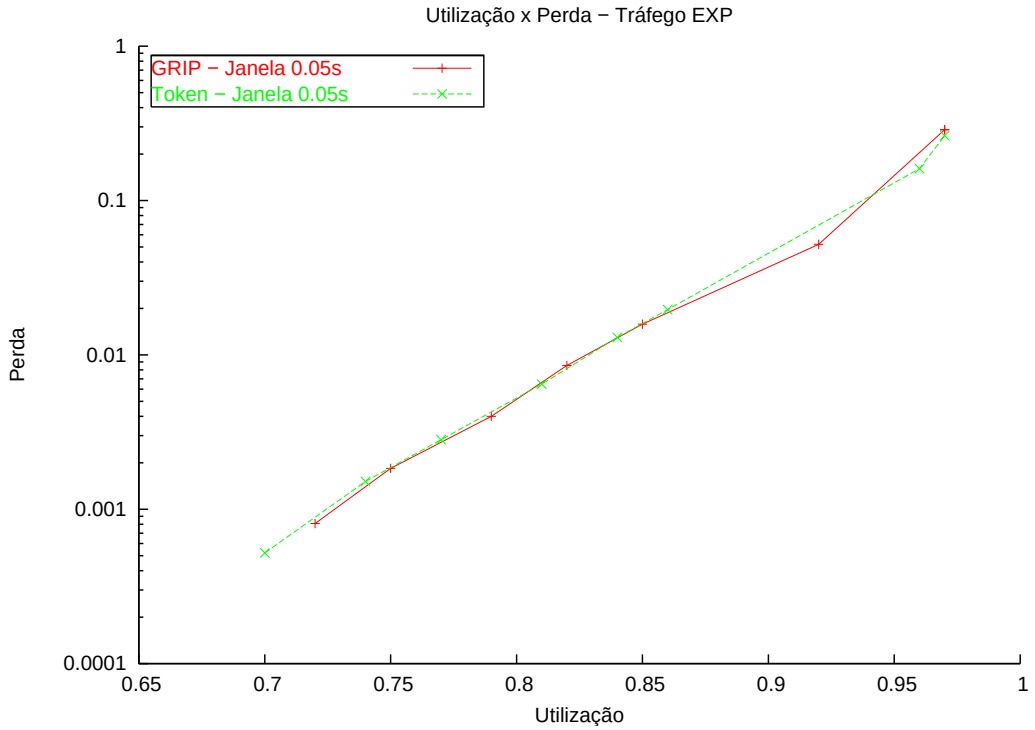

Figure 4.49: Comparação entre GRIP e Token com tráfego EXP utilizando a topologia 1

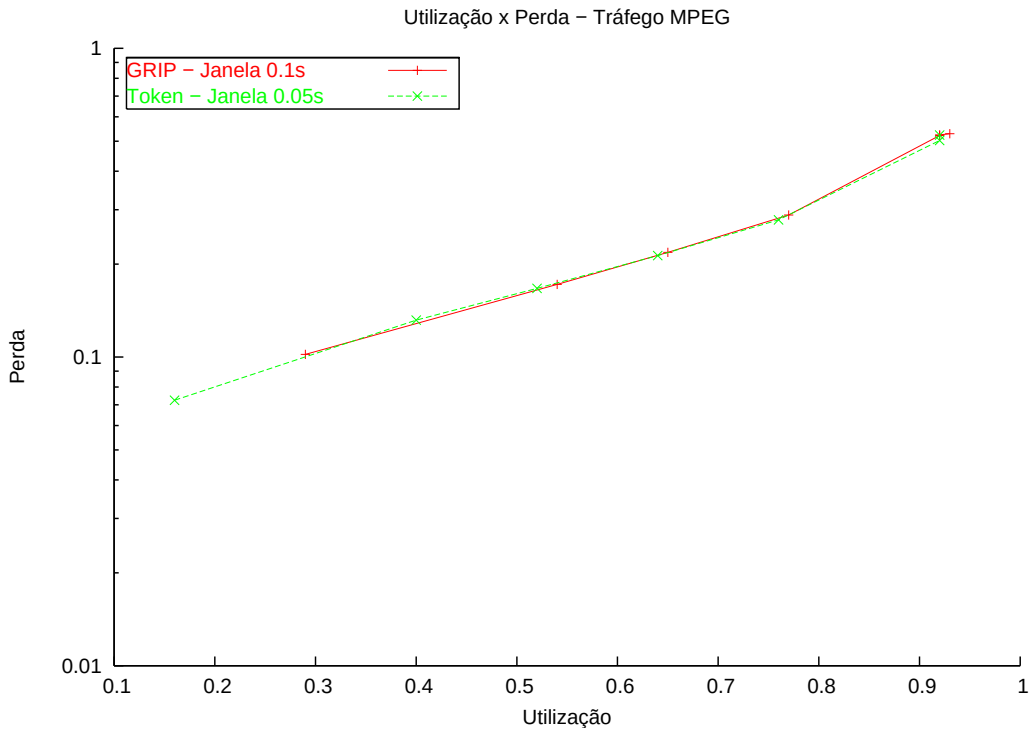

Figure 4.50: Comparação entre GRIP e Token com tráfego MPEG utilizando a topologia 1

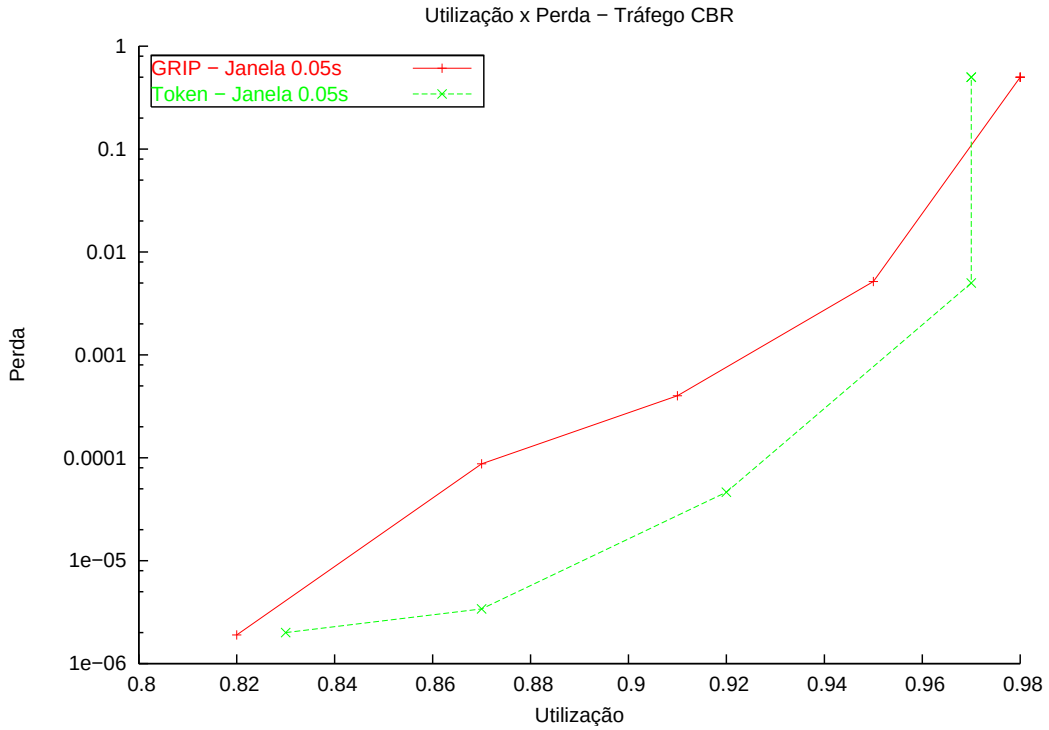

Figure 4.51: Comparação entre GRIP e Token com tráfego CBR utilizando a topologia 2

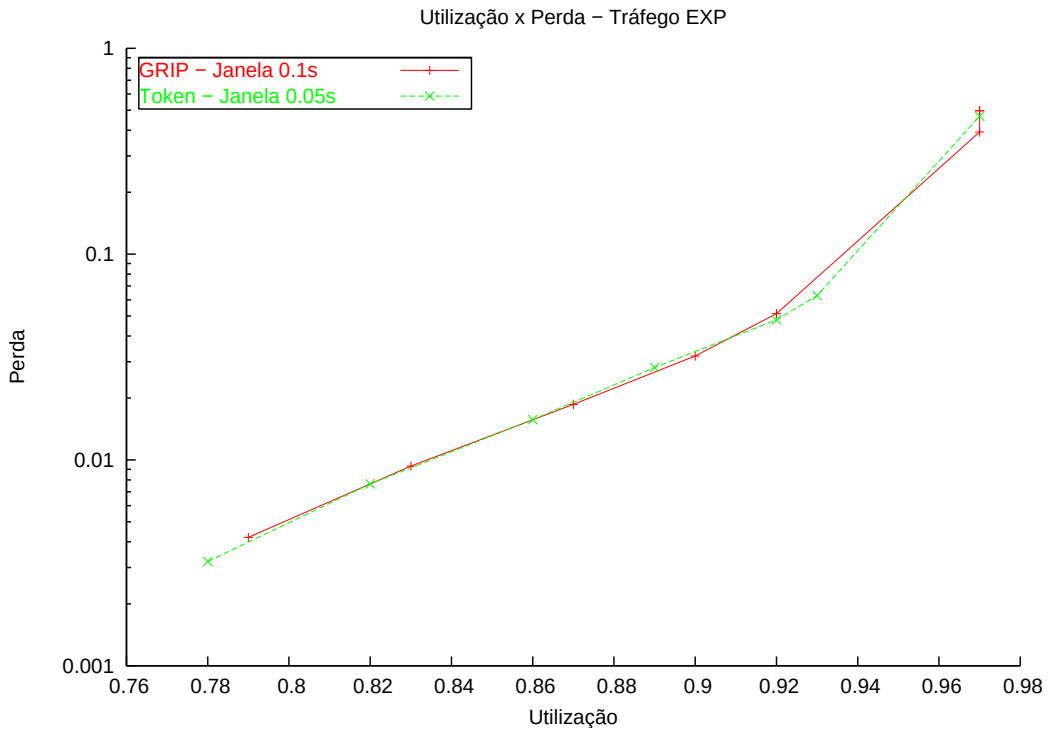

Figure 4.52: Comparação entre GRIP e Token com tráfego EXP utilizando a topologia 2

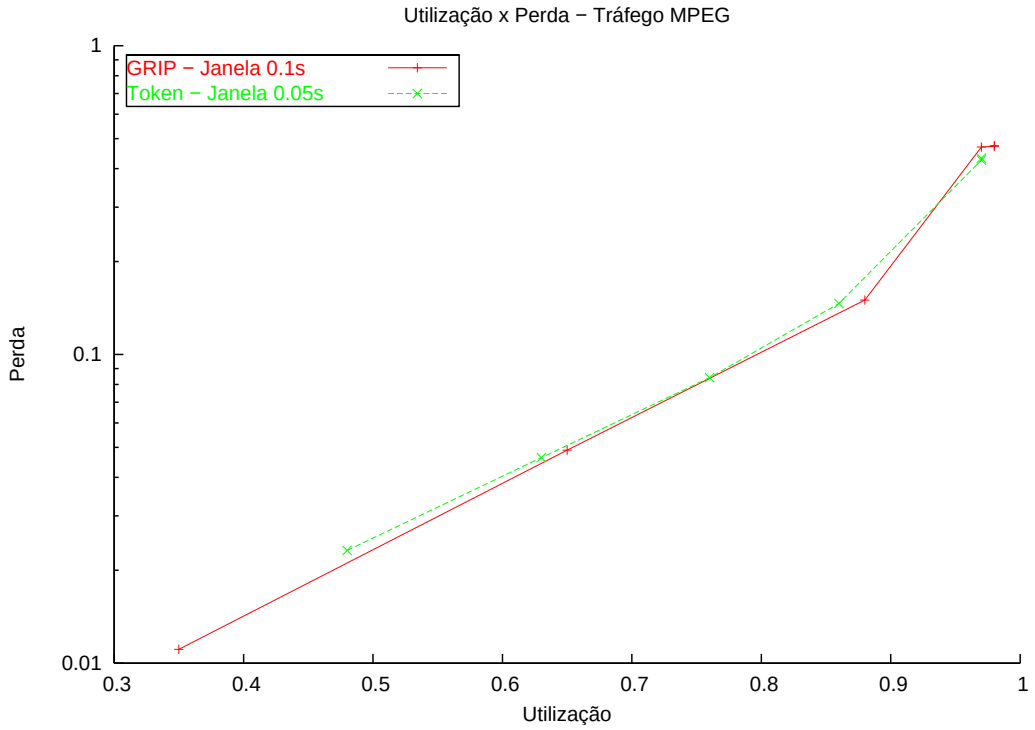

Figure 4.53: Comparação entre GRIP e Token com tráfego MPEG utilizando a topologia 2

Como podemos ver, utilizando a configuração ótima dentre as simuladas nas seções anteriores, o desempenho dos dois algoritmos é bastante semelhante, exceto para o tráfego CBR, onde o Token apresenta um desempenho bem melhor.

No entanto, os algoritmos apresentaram diferente sensibilidade ao tamanho da janela, para cada tipo de tráfego, como comentado a seguir.

- **CBR**: na topologia 1, o Token apresenta uma sensibilidade muito maior ao tamanho da janela, com grandes diferenças do nível de desempenho em cada configuração. Na topologia 2, entretanto, há praticamente dois níveis de desempenho: as curvas das janelas 0.05s e 0.1s bem próximas, e as outras curvas "agrupadas" em outro nível de desempenho, bastante inferior. Já no caso do GRIP, a mudança de topologia resultou em maior diferenciação no desempenho das janelas.
- **EXP**: em ambas as topologias, o efeito de mudança da janela foi praticamente nulo no Token, contrariando as expectativas. No GRIP, houve uma diferença maior da janela de 5 segundos para as outras. O desempenho semelhante da janelas abaixo de 5 segundos nos leva à conclusão de que essa é a melhor curva perda-carga para esse algoritmo, sob essas condições de tráfego.
- **MPEG**: como explicado anteriormente, as curvas perda-carga do GRIP mostram que a redução da janela tem um ponto ótimo a partir do qual o desempenho piora. Algo semelhante acontece no Token, com as curvas das janelas de 0.05s e 0.1s praticamente sobrepostas. Como no caso do tráfego exponencial sob o GRIP, podemos concluir que a janela de 0.1 segundo no GRIP e as janelas de 0.1 segundo e 0.05 segundo no Token representam a melhor configuração dos algoritmos para os padrões de tráfego MPEG examinados.

De uma maneira geral, o Token se mostrou mais sensível ao tamanho da janela, como pode ser observado nos gráficos M x Utilização da seção anterior. Quanto maior a janela, mais "lento" ficou o Token na alocação de usuários. Para comprovar essa característica, observemos as Figuras 4.54 a 4.57. Esses gráficos ilustram a alocação de usuários do Token e do GRIP para duas janelas diferentes (0.05s e 0.5s).

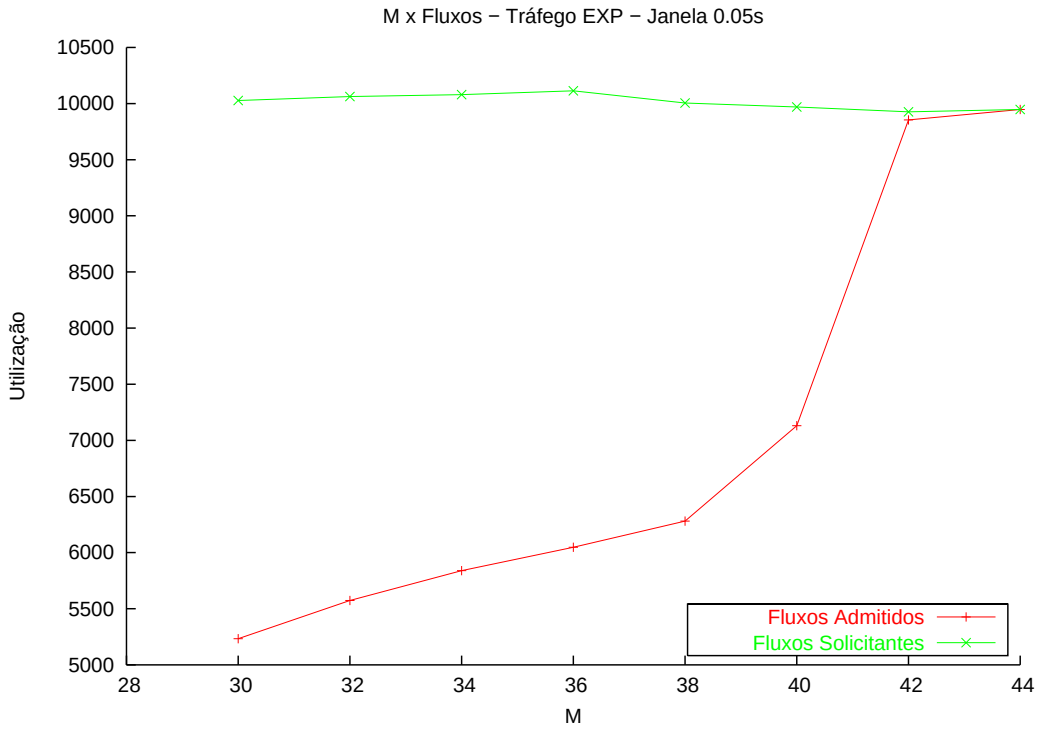

Figure 4.54: Alocação de fluxos no GRIP com janela 0.05s - Topologia 1

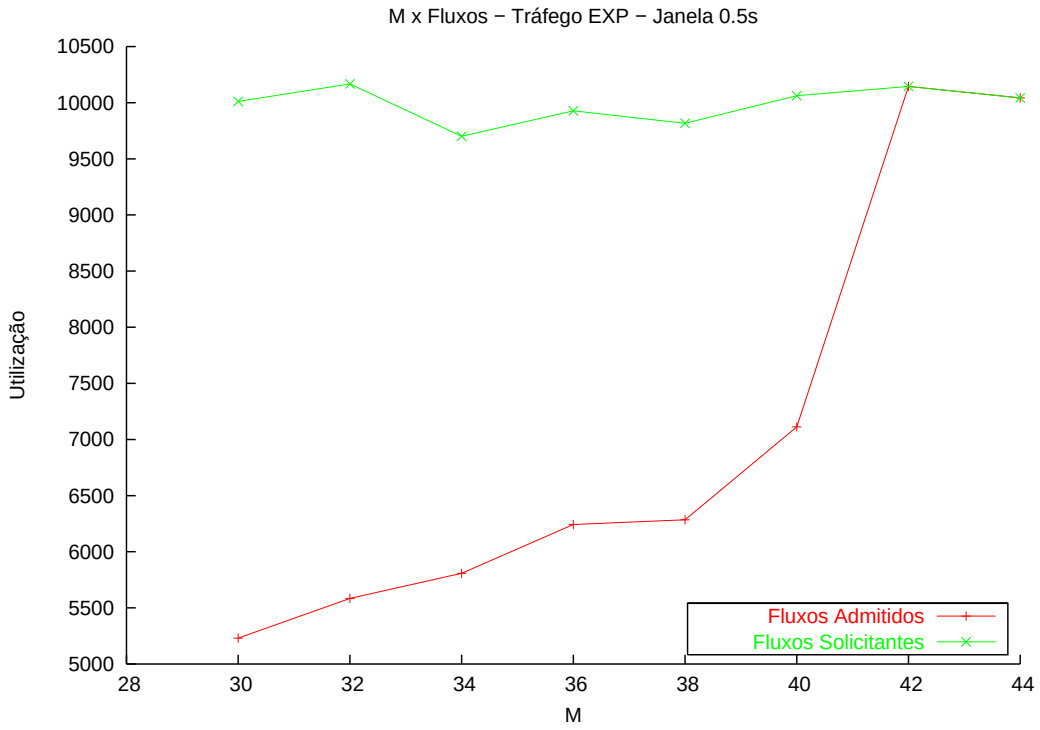

Figure 4.55: Alocação de fluxos no GRIP com janela 0.5s - Topologia 1

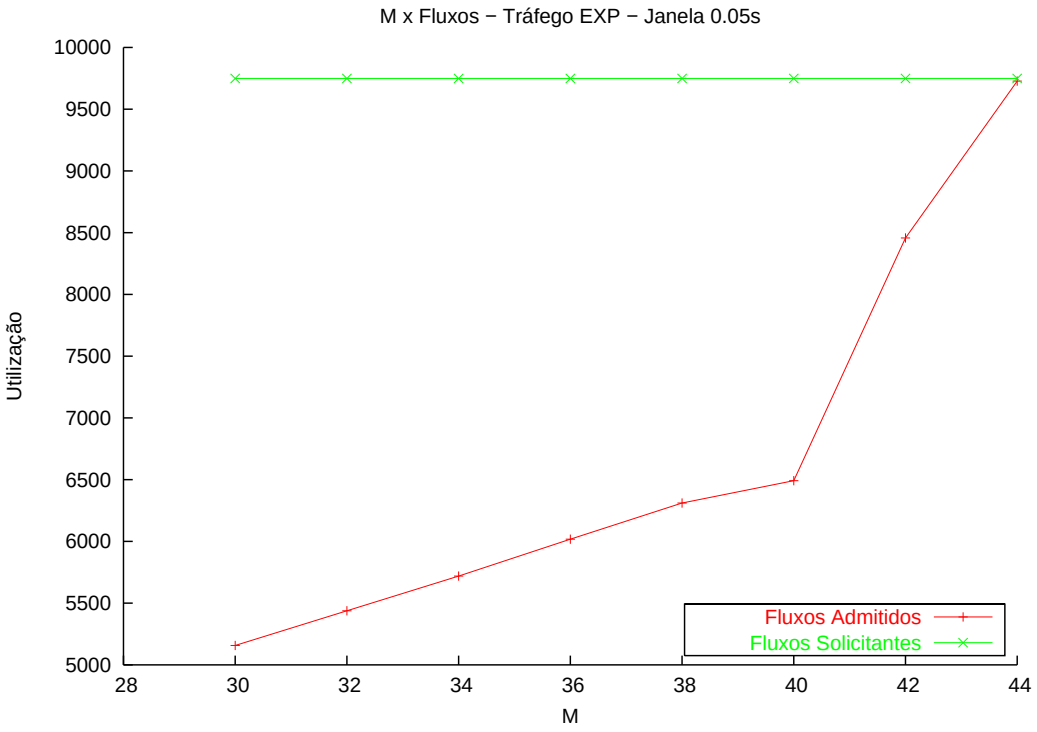

Figure 4.56: Alocação de fluxos no Token com janela 0.05s - Topologia 1

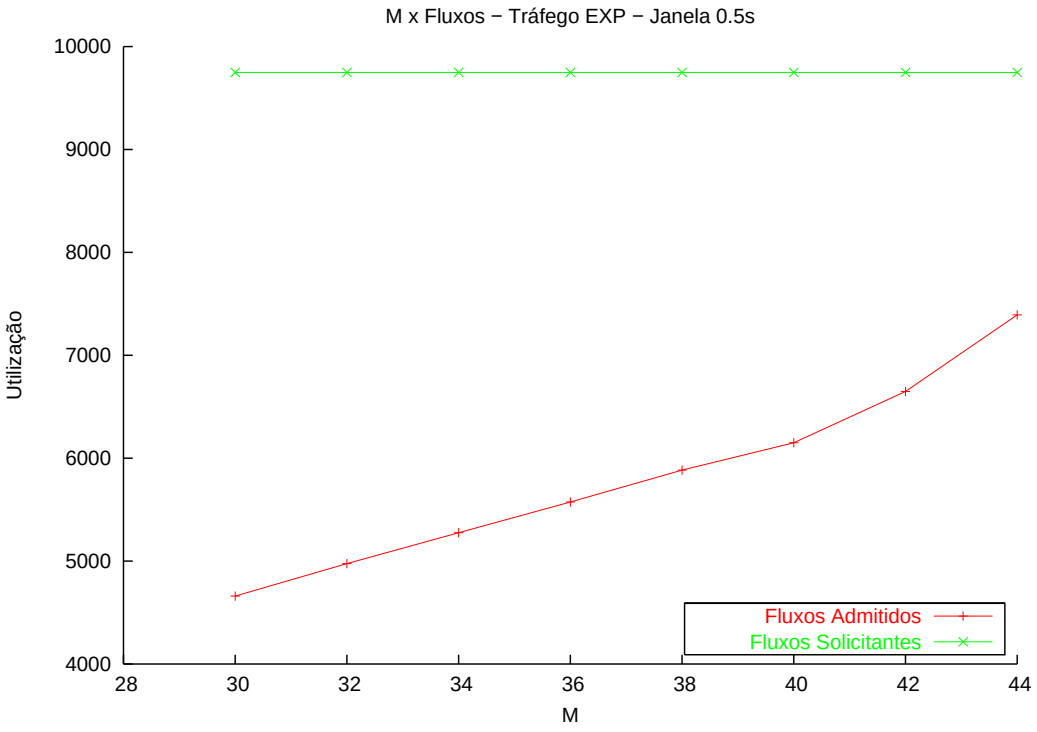

Figure 4.57: Alocação de fluxos no Token com janela 0.5s - Topologia 1

Vemos que o efeito do aumento da janela no GRIP é bem pequeno, enquanto o Token fica bem mais lento com a janela maior. Isso é consequência do esquema de alocação de usuários utilizado pelo Token. Para explicar esse efeito, imaginemos que o sistema se encontra configurado com M maior do que 40. Portanto, Th é maior que a capacidade do enlace. No GRIP, isso significa que cada usuário que chegar ao nó de entrada será admitido, o que pode ser observado nas Figuras 4.54 e 4.55. No token, isso não acontece, porque esse algoritmo restringe o número de usuário que podem ser admitidos, como foi explicado na seção 4.2.1. No caso de $M = 44$, por exemplo, teremos:

$$u = \frac{Th - B}{R.\Delta T}$$

$$Th = M.R.\Delta T$$
$$B = C.\Delta T$$

sendo B o volume de tráfego medido pelo nó de borda. Supondo que o enlace está totalmente ocupado, o volume medido no período ΔT é proporcional a 2Mbps, e assim, temos

$$usuarios = \frac{M.R.\Delta T - C.\Delta T}{R.\Delta T}$$

$$= \frac{M.R - C}{R}$$

$$= \frac{44.50k - 2M}{50k} = 4$$

Ou seja, o algoritmo vai admitir, no máximo, 4 usuários a cada período ΔT, independente do valor de ΔT. Logo, quanto menor for ΔT, mais usuários serão alocados ao longo da simulação.

Uma última análise que podemos fazer é com relação ao tamanho do buffer. Por ser um parâmetro-chave em análises de sistemas de filas, consideramos importante fazer uma breve análise do efeito do tamanho do buffer no desempenho do sistema. Repetimos, então, algumas simulações com o tráfego exponencial utilizando buffer de 40 pacotes e comparamos os resultados com as simulações obtidas anteriormente, com buffer de 10 pacotes. Os resultados estão nas Figuras 4.58 a 4.61.

Observa-se, como era de se esperar, uma queda na taxa de perdas com o aumento do buffer. No entanto, é interessante observar que não houve nenhuma modificação significativa no comportamento das curvas.

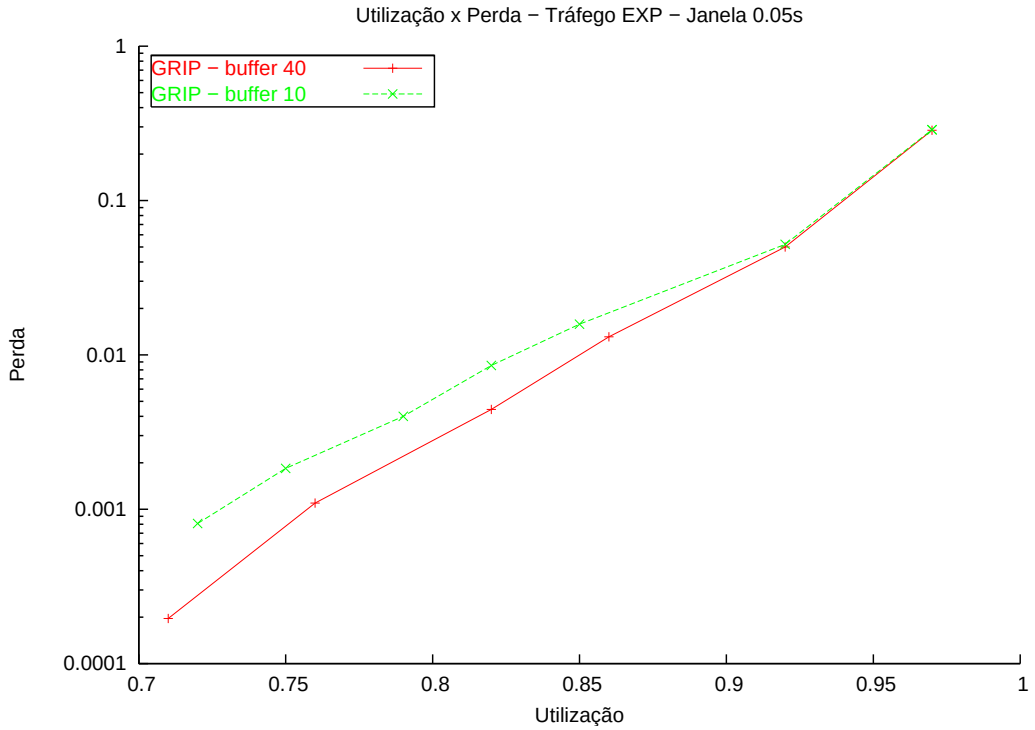

Figure 4.58: Comparação do GRIP entre buffers com tráfego EXP utilizando a topologia 1

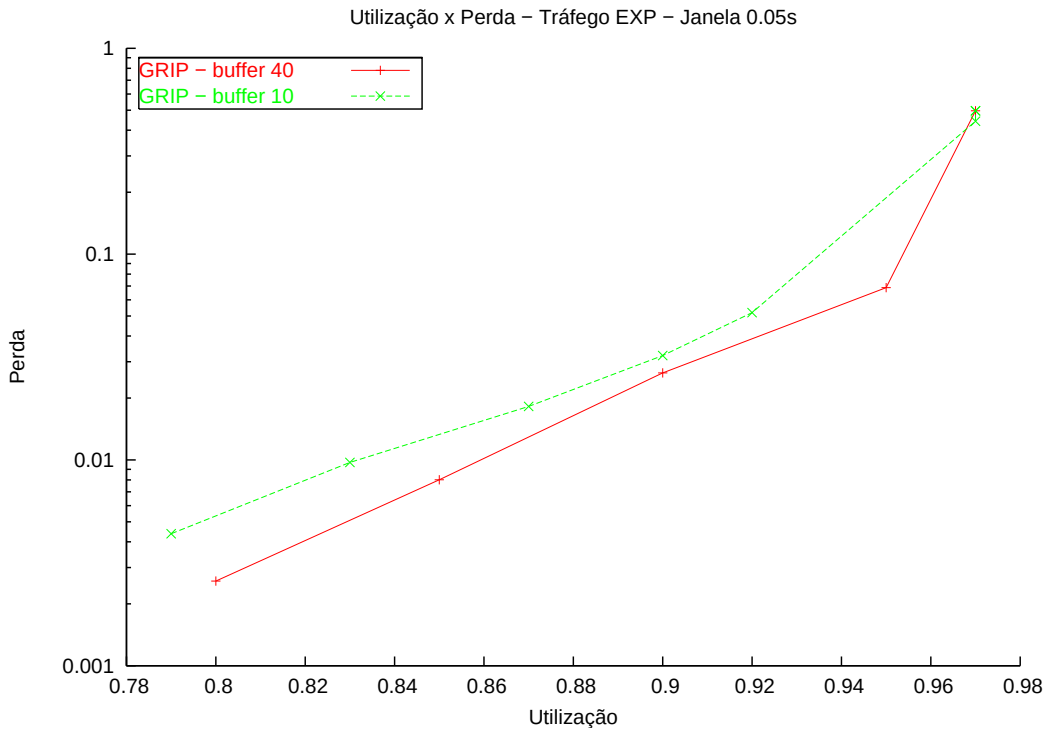

Figure 4.59: Comparação do GRIP entre buffers com tráfego EXP utilizando a topologia 2

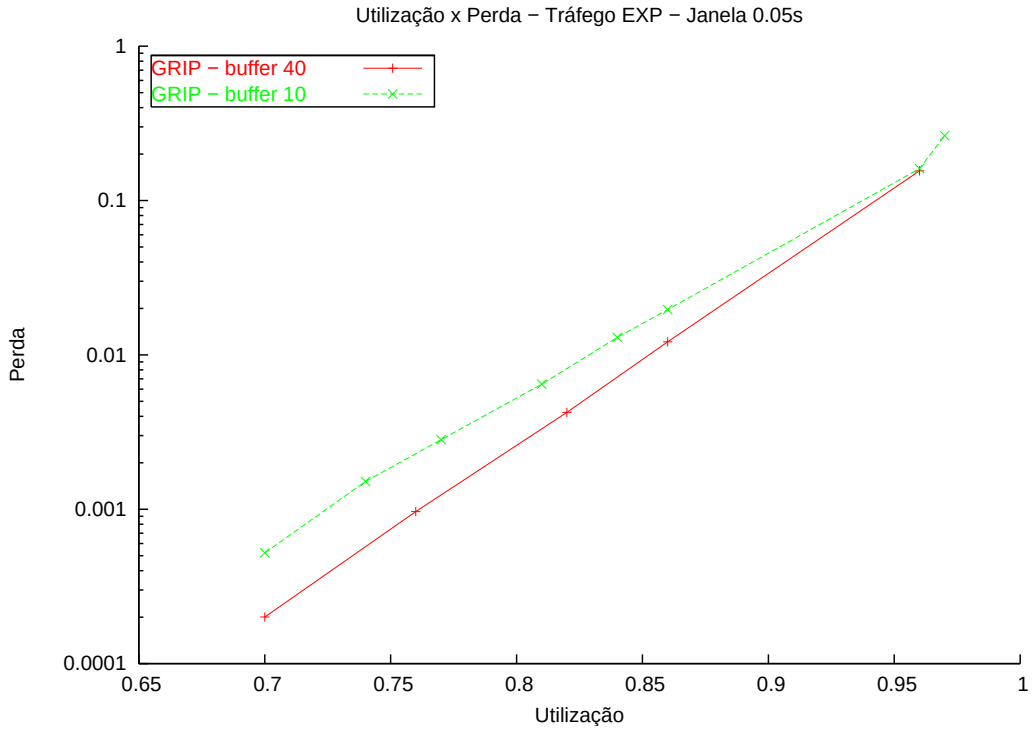

Figure 4.60: Comparação do Token entre buffers com tráfego EXP utilizando a topologia 1

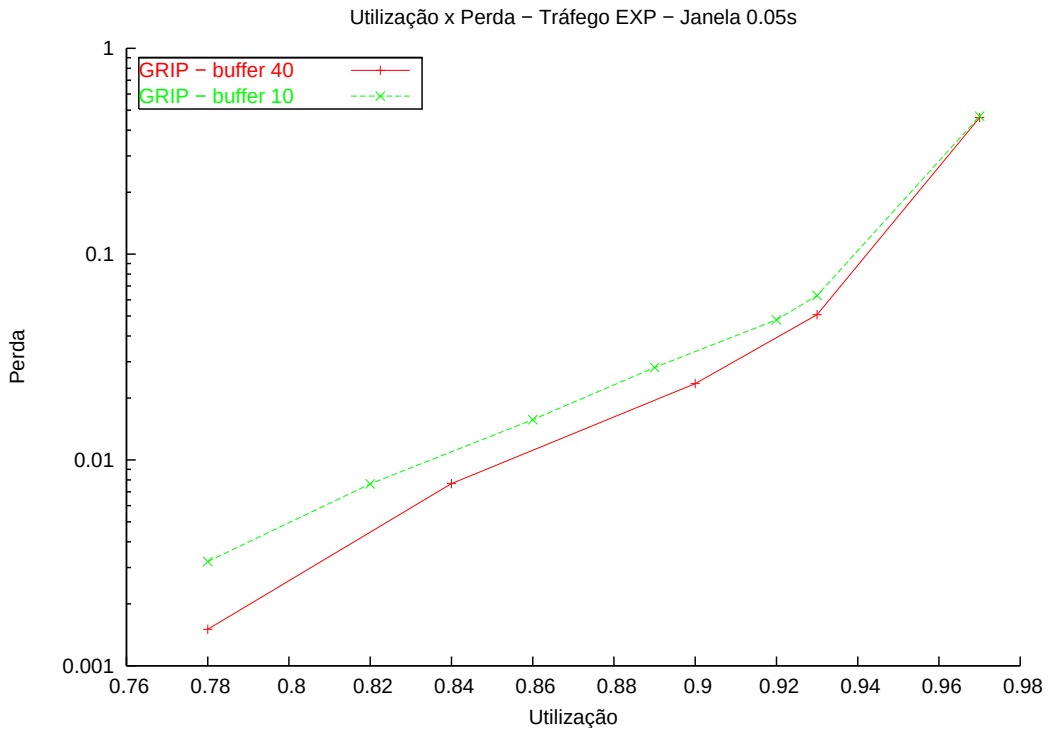

Figure 4.61: Comparação do Token entre buffers com tráfego EXP utilizando a topologia 2