

9

Algoritmos de Controle de Admissão de Usuários para Sistemas de Múltiplas Classes

Neste capítulo são apresentados e propostos quatro algoritmos de Controle de Admissão de Chamadas, CAC, para sistemas de múltiplas classes de usuários. Os algoritmos são dinâmicos e distribuídos e tem como objetivo geral atender aos requisitos de qualidade de serviço mantendo elevada a utilização dos recursos do sistema.

Os algoritmos para M classes de usuários propostos neste capítulo são extensões de alguns dos algoritmos de uma classe de usuários, apresentados no capítulo 5. Da mesma forma que os anteriores, os algoritmos de M classes também são testados através do simulador descrito no capítulo 4, seu desempenho é comparado aos resultados dos sistemas com número fixo de canais de guarda, apresentados no capítulo 8.

Além disto, os algoritmos propostos tem seu desempenho testado em simulações que utilizam modelos novos para o tempo de permanência do usuário na célula. Tal trabalho é uma extensão do trabalho feito na referência [54] considerando além da distribuição exponencial, as distribuições constante, uniforme e Weibull para o tempo de permanência do usuário na célula.

Os algoritmos *CAC1M* e *CAC2M*, que são apresentados neste capítulo, foram publicados em [63]. Nesta tese seus resultados são apresentados com maior detalhe do que foi possível apresentar no artigo, incluindo o seu comportamento para outras distribuições que modelam o tempo de permanência do usuário na célula.

Nos algoritmos propostos, o número de canais de guarda em cada setor é definido separadamente para cada classe de usuário. Os usuários novos e usuários em *handoff* da classe γ são chamados de N_γ e H_γ . Cada algoritmo apresenta critérios diferentes para modificar o número de recursos reservados correspondente a sua classe, de uma forma dinâmica, conforme o desempenho obtido pelas medidas $(B_{BN}; B_{FH})_\gamma$ em relação aos requisitos de qualidade de serviço $(Q_{BN}; Q_{FH})_\gamma$. Portanto, todos algoritmos

propostos decidem sobre a admissão de chamadas baseados no desempenho instantâneo de cada setor de uma forma distribuída e dinâmica.

Os algoritmos são definidos para um número genérico M de classes de usuários. Neste capítulo seu desempenho é avaliado para duas classes, $M = 2$. Ou seja, temos quatro tipos de usuários diferentes, cada um com seu requisito de qualidade de serviço. Foram testados dois conjuntos de requisitos de qualidade nas simulações:

$$Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3\%, 1\%, 5\%, 3\%) \quad (9-1)$$

$$Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (4\%, 0.4\%, 5\%, 3\%) \quad (9-2)$$

As classes simuladas apresentam taxas totais de chegada de usuários iguais.

$$\lambda_1 = \lambda_2 \quad (9-3)$$

Cada taxa de chegada de determinada classe é dividida entre usuários novos e em *handoff*.

$$\lambda_1 = \lambda_{N1} + \lambda_{H1} \quad (9-4)$$

$$\lambda_2 = \lambda_{N2} + \lambda_{H2} \quad (9-5)$$

Nas simulações realizadas neste trabalho, a taxa de mobilidade é considerada a mesma para as duas classes.

$$\lambda_\gamma = \lambda_{N\gamma} + \lambda_{H\gamma} = \lambda_{N\gamma} + \theta\lambda_{N\gamma} \quad (9-6)$$

A demanda de recursos do sistema da classe 1 e de um canal (ou uma unidade de banda) e da classe 2 é de dois canais.

$$B_1 = 1 \quad (9-7)$$

$$B_2 = 2 \quad (9-8)$$

Nas seções subseqüentes são apresentados os algoritmos dinâmicos para o controle de admissão de chamadas ao setor, seus objetivos e seus respectivos resultados.

9.1

Algoritmo CAC1M

Este algoritmo busca favorecer as chamadas em *handoff* enquanto a probabilidade de bloqueio de chamadas novas é baixa. Os canais são reservados aos usuários em *handoff* em cada setor, correspondente à cada classe γ , de uma forma dinâmica, ou seja, de acordo com o desempenho do sistema. O algoritmo *CAC1M* é uma evolução de *CAC1*, apresentado no capítulo 5 desta tese. Da mesma forma que sua versão para uma classe, é um algoritmo simples e que se baseia apenas nas falhas de chamadas em *handoff* e na medida do percentual de bloqueio de chamadas novas.

É sabido que a reserva de canais diminui a utilização dos recursos. Com o objetivo de obter alta a utilização dos canais, o setor inicia com número de recursos reservados para os usuários em *handoff*, $R_{H\gamma}$, igual a zero. A cada falha de *handoff* de usuário da classe γ o valor de $R_{H\gamma}$ é incrementado.

Por outro lado, para evitar o aumento excessivo de $R_{H\gamma}$, a cada alocação de usuário novo da classe γ , N_γ , o percentual de bloqueio de chamadas novas é observado. A medida instantânea do percentual de bloqueio de chamadas novas, $B_{BN\gamma}$, é comparada ao seu valor máximo tolerável definido pelo requisito de qualidade de serviço para usuários novos, $Q_{BN\gamma}$. Se a medida $B_{BN\gamma}$ se aproxima do requisito $Q_{BN\gamma}$, o número de canais de guarda, $R_{H\gamma}$, é decrementado.

O parâmetro β_γ define o limiar de proximidade entre a medida de percentual de bloqueio, $B_{BN\gamma}$, e o parâmetro de QoS, $Q_{BN\gamma}$ para a tomada de decisão sobre diminuir o número de canais de guarda, $R_{H\gamma}$.

9.1.1

Descrição do Algoritmo CAC1M

Chegada de chamada em *handoff* da classe γ

- Verifica a disponibilidade de recursos e aloca, ou bloqueia.
- Se houver falha de *handoff*, $R_{H\gamma}$ é incrementado.

Chegada de chamada nova da classe γ

- Verifica se existem canais ordinários e aloca ou bloqueia.
- Registra o sucesso ou falha de alocação na janela de registro correspondente ao tipo de chamada.

- Calcula a medida de bloqueio de novas chamadas, $B_{BN\gamma}$.
- Se $B_{BN\gamma} \geq \beta_\gamma Q_{BN\gamma}$, $R_{H\gamma}$ é decrementado.

O diagrama de blocos do algoritmo proposto CAC1M é apresentado na figura 9.1.

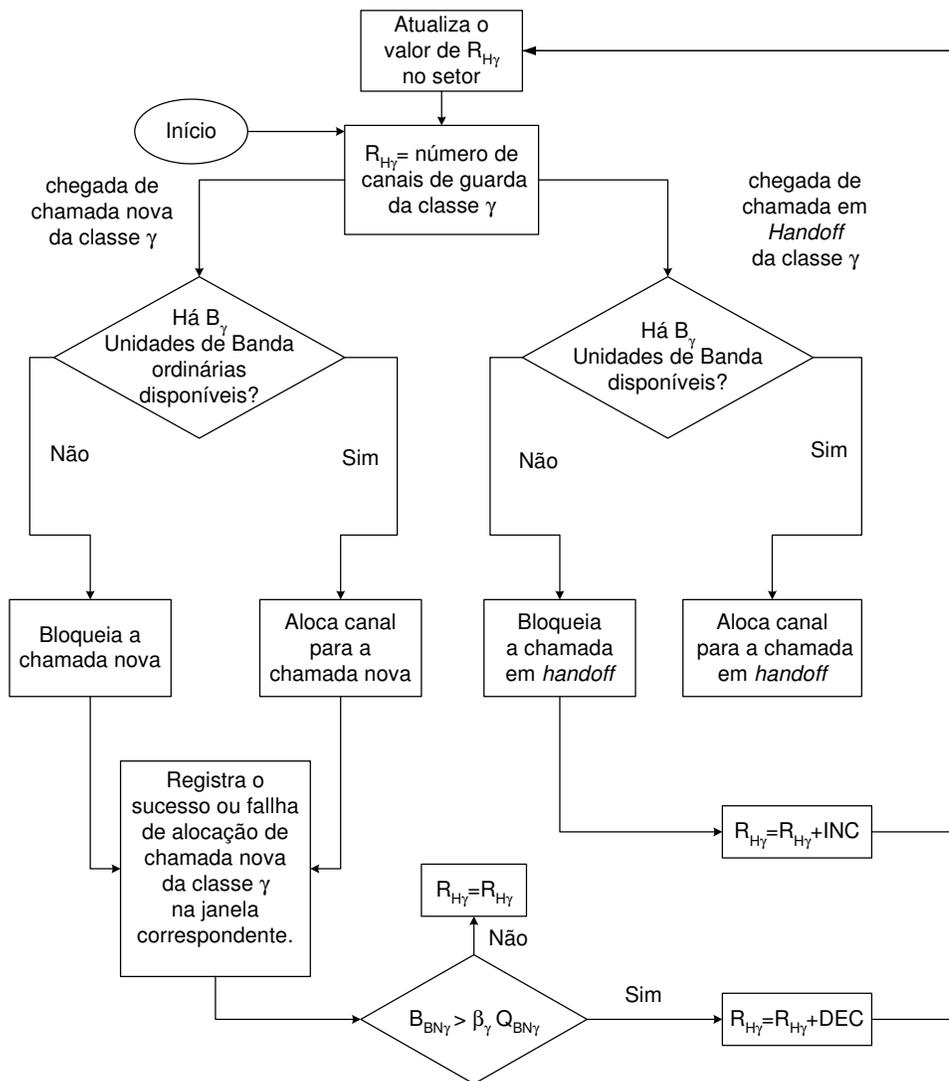


Figura 9.1: Diagrama de blocos do algoritmo CAC1M.

No algoritmo CAC1M os incrementos e decrementos do número de recursos reservados para os usuários em handoff são valores fracionários, onde o parâmetro de decremento, δ , é definido no início da simulação. O objetivo da variação fracionária é ajustar o valor de $R_{H\gamma}$ de uma forma mais suave, acompanhando as modificações do perfil de tráfego no setor.

O requisito de Qualidade de Serviço para Bloqueio de Chamadas Novas da classe γ , $Q_{BN\gamma}$, é definido no início da simulação.

9.1.2

Determinação dos Parâmetros do Algoritmo CAC1M

Nas seções seguintes são apresentados os resultados do desempenho do algoritmo de Controle de Admissão de Canal, *CAC1M*. São testadas variações no comportamento do algoritmo de acordo com os parâmetros de taxa de mobilidade, θ , e do decremento do número de canais de guarda, δ . Os resultados de simulação do algoritmos dinâmico, *CAC1M*, em linguagem C++, são comparados aos resultados analíticos referentes ao número fixo de canais de guarda, obtidos em MATLAB.

Desempenho de CAC1M com δ , Parâmetro de Decremento do Número de Recursos Reservados

No algoritmo CAC1M os incrementos do número de canais de guarda de cada classe são feitos a cada falha de *handoff* da respectiva classe. O valor de incremento é de $+2\delta B_\gamma$, ou seja, uma fração da banda requisitada.

O número de canais de guarda de determinada classe γ é reduzido a cada falha de chamada nova da mesma classe, respeitando a condição de que a medida de bloqueio de chamadas novas, $B_{BN\gamma}$, seja superior a $\beta_\gamma Q_{BN\gamma}$. O valor do decremento é de δB_γ .

A tabela 5.1 apresenta os valores dos incrementos e decrementos do algoritmo *CAC1M*.

Tabela 9.1: Incrementos e decrementos do algoritmo *CAC1M*.

<i>CAC1M</i>	variação de $R_{H\gamma}$
Falha de <i>Handoff</i>	$+2\delta B_\gamma$
$B_{BN\gamma} > \beta_\gamma Q_{BN\gamma}$	$-\delta B_\gamma$

Nas figuras 9.2 e 9.3 são apresentados os resultados do algoritmo CAC1M para vários valores do parâmetro de decremento δ . Os demais parâmetros de simulação são os parâmetros de proximidade dos requisitos de QoS $\beta_1 = 1$ e $\beta_2 = 1$; e os requisitos $(Q_{BN1}; Q_{FH1})$ e $(Q_{BN2}; Q_{FH2})$ iguais a (5%; 3%).

Observa-se nas figuras 9.2 e 9.3 que, quanto menor o valor do decremento δ , maior o número médio de canais de guarda. A curva de $\delta = 0.2$ apresenta decrementos menores que os incrementos e o valor médio do número de canais de guarda tende a ficar cada vez maior, conforme aumenta a intensidade de tráfego de entrada. O valor adotado para o parâmetro de decremento é $\delta = 0.5$ que apresentou melhor desempenho, sem incrementar demais ou de menos.

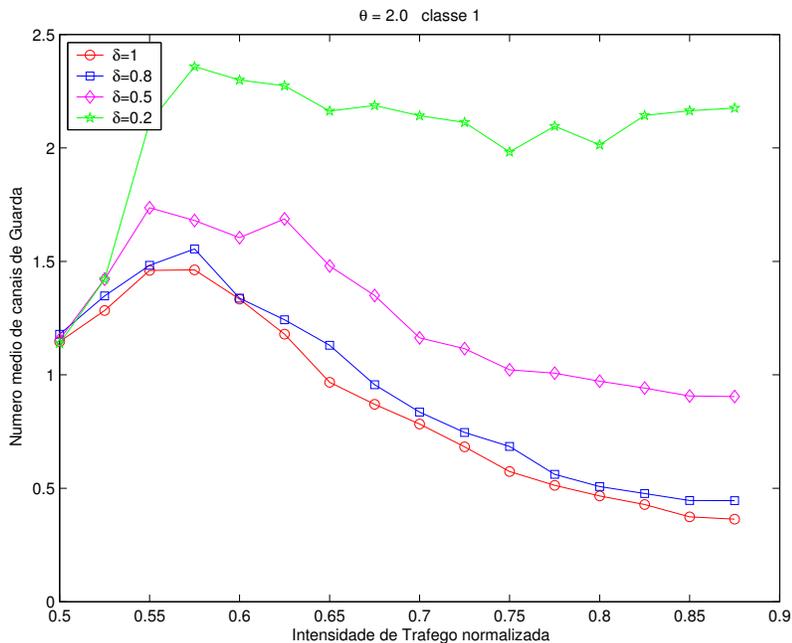


Figura 9.2: Número médio de canais de guarda para a classe 1 versus intensidade total de tráfego de entrada do algoritmo CAC1M para vários valores do decremento δ ; $\beta_1 = 1$; $\beta_2 = 1$; e $(Q_{BN1}; Q_{FH1}) = (Q_{BN2}; Q_{FH2}) = (5\%; 3\%)$.

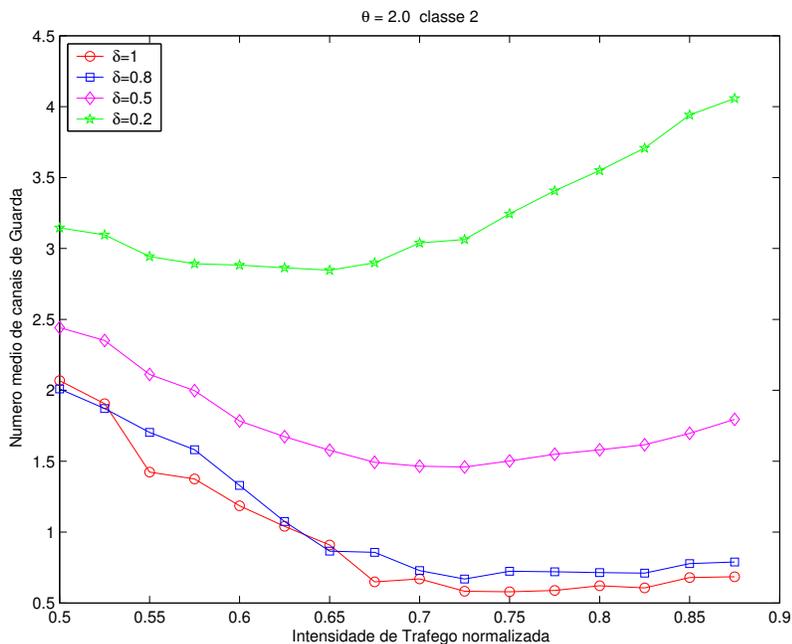


Figura 9.3: Número médio de canais de guarda para a classe 2 versus intensidade total de tráfego de entrada do algoritmo CAC1M para vários valores do decremento δ ; $\beta_1 = 1$; $\beta_2 = 1$; e $(Q_{BN1}; Q_{FH1}) = (Q_{BN2}; Q_{FH2}) = (5\%; 3\%)$.

Em todos os casos analisados, observa-se que o número médio de canais de guarda reservado para a classe 2, cuja banda é $B_2 = 2$, é sempre maior que o número médio de canais de guarda reservado para a classe 1 cuja banda é menor, $B_1 = 1$. Isso se deve ao fato dos acréscimos e decréscimos que são sempre proporcionais ao tamanho da banda de cada classe. Classes maiores tendem a serem mais bloqueadas e precisam de mais unidades de banda para suas chamadas em *handoff*.

9.1.3 Resultados do Algoritmo CAC1M

As figuras 9.4, 9.5, 9.6, 9.5 e 9.8 apresentam o comportamento do algoritmo *CAC1M* no caso de taxa de mobilidade $\theta = 1$ e requisitos de QoS $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, versus intensidade de tráfego de entrada normalizada, ρ .

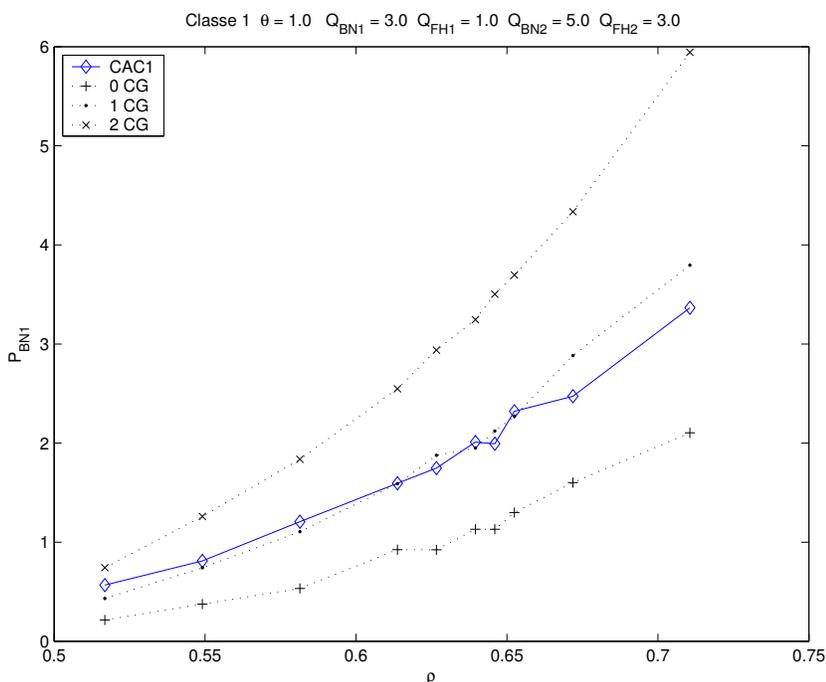


Figura 9.4: Percentual de Bloqueio de chamadas novas da classe 1 do algoritmo CAC1M.

As curvas de percentual de bloqueio de chamadas novas e de falhas de usuários em *handoff* da classe 1 estão nas figuras 9.4 e 9.5. Observa-se que CAC1M se comporta, aproximadamente, em torno da curva de 1CG.

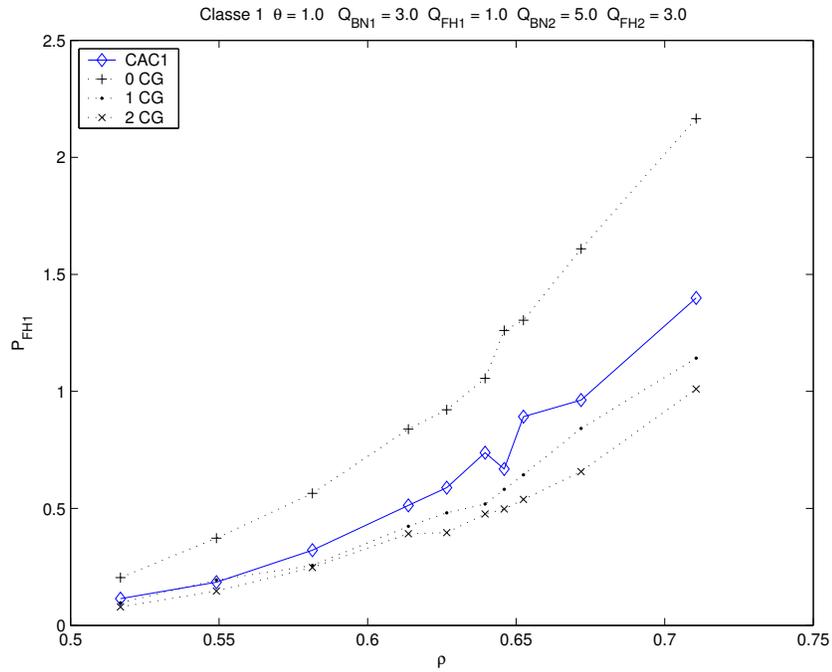


Figura 9.5: Percentual de Falha de chamadas em *handoff* da classe 1 do algoritmo CAC1M.

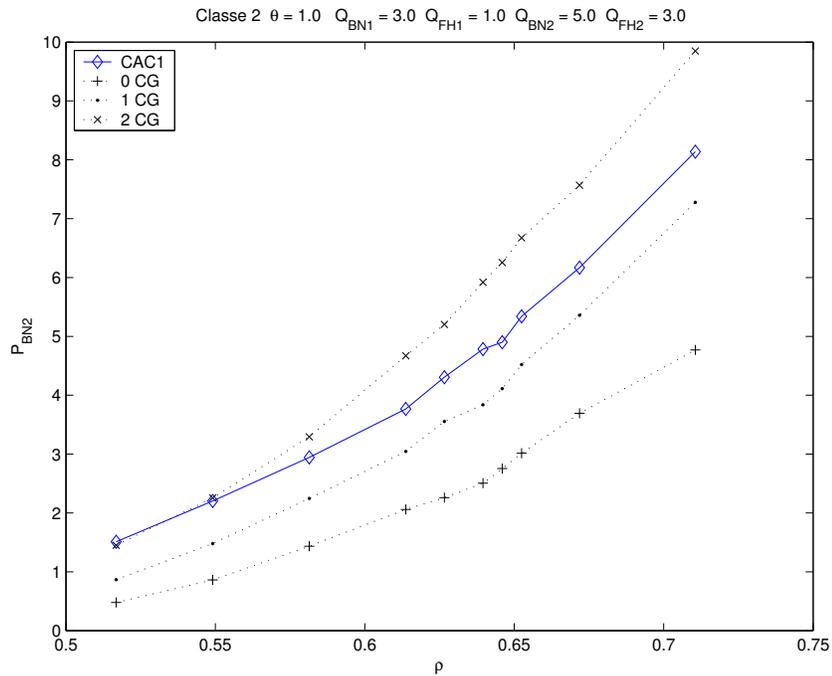


Figura 9.6: Percentual de Bloqueio de chamadas novas da classe 2 do algoritmo CAC1M.

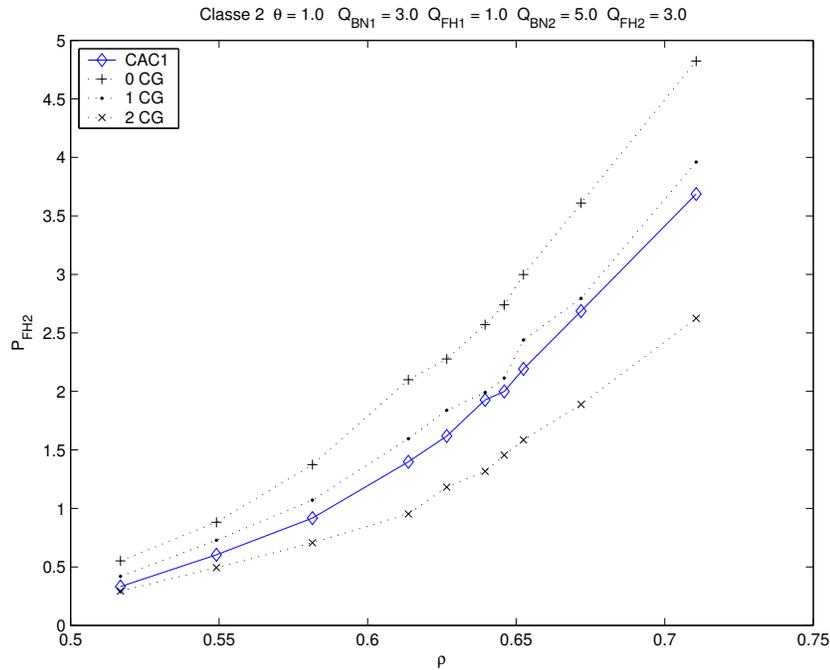


Figura 9.7: Percentual de Falha de chamadas em *handoff* da classe 2 do algoritmo CAC1M.

Para a classe 2, as curvas de percentual de bloqueio de chamadas novas e de falhas de usuários em *handoff* estão nas figuras 9.6 e 9.7. Observa-se que CAC1M se comporta, aproximadamente, em torno de $1,5CG$.

Na figura 9.8 é apresentada a quantidade de recursos reservada para cada um dos tipos de usuário. Observa-se que o algoritmo CAC1M reserva recursos apenas para os usuários em *handoff* de ambas as classes.

Comparando o número de recursos reservados para os usuários em *handoff* de cada classe, nota-se que uma menor quantidade de recursos do sistema foi reservada para os usuários em *handoff* da classe 1; e a classe 2, que requisita mais recursos do sistema, obteve mais recursos reservados para seus usuários em *handoff*.

No final deste capítulo são apresentados diversos resultados do desempenho de todos os algoritmos propostos para M classes em cenários diferentes.

9.2

Algoritmo CAC2M

O algoritmo CAC2M é um aprimoramento de CAC1M onde é introduzido o requisito de QoS relativo às chamadas em *handoff*, $Q_{FH\gamma}$.

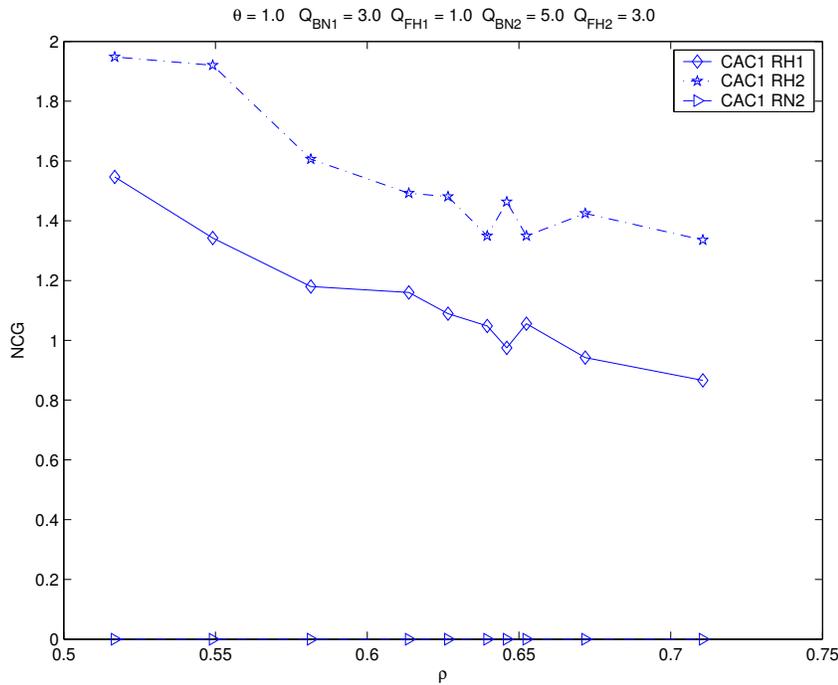


Figura 9.8: Número de Recursos reservados pelo algoritmo CAC1M.

Assim, o controle dos acréscimos do número de recursos reservados é feito pela medida do percentual de bloqueio de novas chamadas, $B_{BN\gamma}$, e pela medida do percentual de falha de chamadas em *handoff* da classe γ , $B_{FH\gamma}$.

9.2.1 Descrição do Algoritmo CAC2M

Chegada de chamada em *handoff* da classe γ

- Verifica se existem recursos disponíveis e aloca, ou bloqueia.
- Registra o sucesso ou falha de alocação na janela de registro correspondente ao tipo de chamada.
- Calcula a medida de falha de chamadas em *handoff*, $B_{FH\gamma}$.
- Em caso de não alocação, se $B_{FH\gamma} \geq \alpha_{\gamma} Q_{FH\gamma}$, $R_{H\gamma}$ é incrementado.

Chegada de chamada nova da classe γ

- Verifica se existem canais ordinários e aloca ou bloqueia.
- Registra o sucesso ou falha de alocação na janela de registro correspondente ao tipo de chamada.
- Calcula a medida de bloqueio de novas chamadas, $B_{BN\gamma}$.

- Em caso de não alocação, se $B_{BN\gamma} \geq \beta_\gamma Q_{BN\gamma}$, $R_{H\gamma}$ é decrementado.

O diagrama de blocos do algoritmo proposto CAC2M é apresentado na figura 9.9.

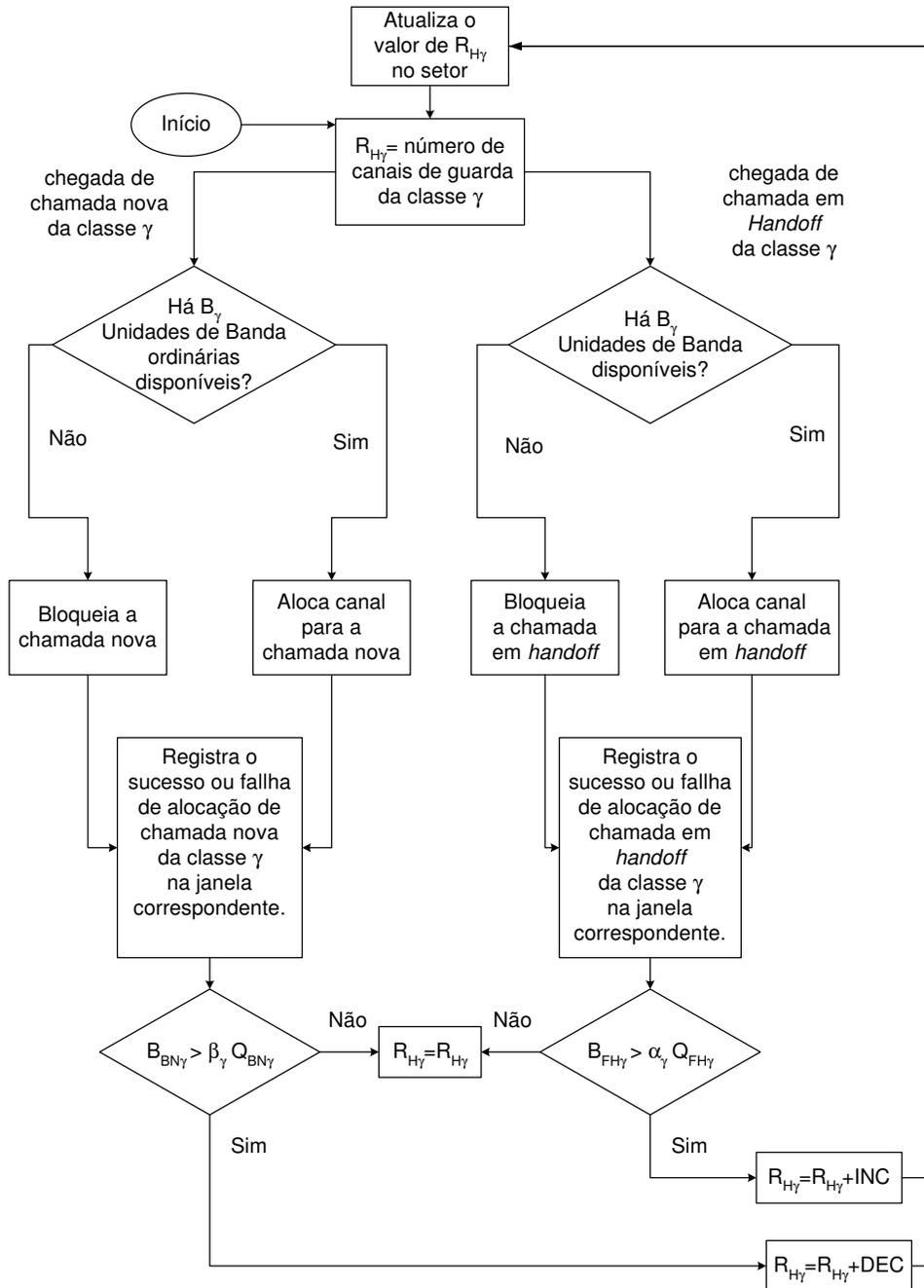


Figura 9.9: Diagrama de blocos do algoritmo CAC2M.

Assim como no algoritmo CAC1M, o algoritmo CAC2M também utiliza incrementos e decrementos do número de recursos reservados para

os usuários em *handoff* como valores fracionários através do parâmetro de decremento, δ que é definido no início da simulação. A tabela 5.4 apresenta os valores dos incrementos e decrementos do algoritmo *CAC2M*.

Tabela 9.2: Incrementos e decrementos do algoritmo *CAC2M*.

<i>CAC2M</i>	variação de $R_{H\gamma}$
$B_{FH\gamma} > \alpha_\gamma Q_{FH\gamma}$	$+\delta B_\gamma$
$B_{BN\gamma} > \beta_\gamma Q_{BN\gamma}$	$-\delta B_\gamma$

9.2.2 Resultados do Algoritmo CAC2M

As figuras 9.10, 9.11, 9.12, 9.11 e 9.14 apresentam o comportamento do algoritmo *CAC1M* no caso de taxa de mobilidade $\theta = 1$ e requisitos de QoS $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, versus intensidade de tráfego de entrada normalizada, ρ .

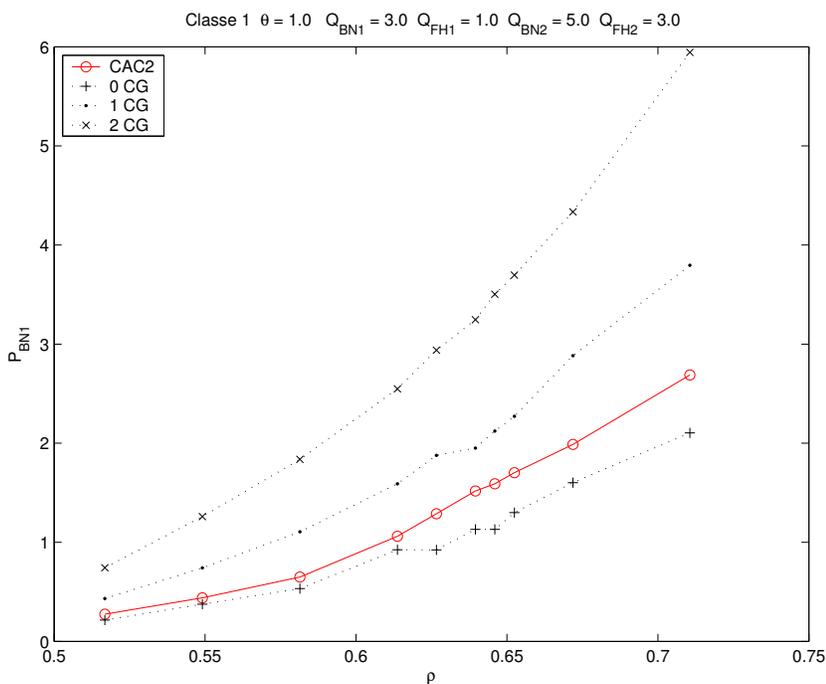


Figura 9.10: Percentual de Bloqueio de chamadas novas da classe 1 do algoritmo *CAC2M*.

Na figura 9.14 é apresentada a quantidade de recursos reservada para cada um dos tipos de usuário. Observa-se que o algoritmo *CAC2M* reserva recursos apenas para os usuários em *handoff* das classes 1 e 2. Para os

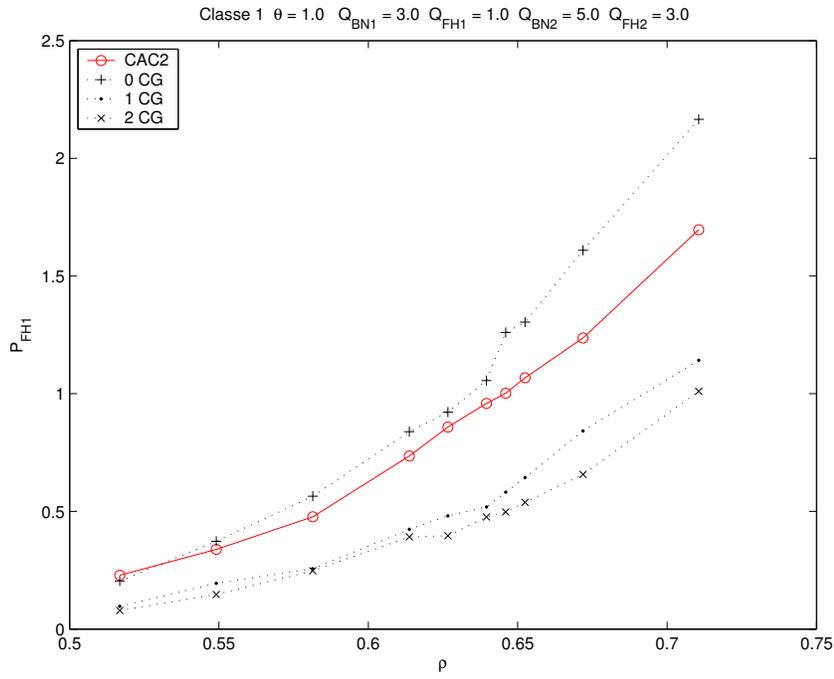


Figura 9.11: Percentual de Falha de chamadas em *handoff* da classe 1 do algoritmo CAC2M.

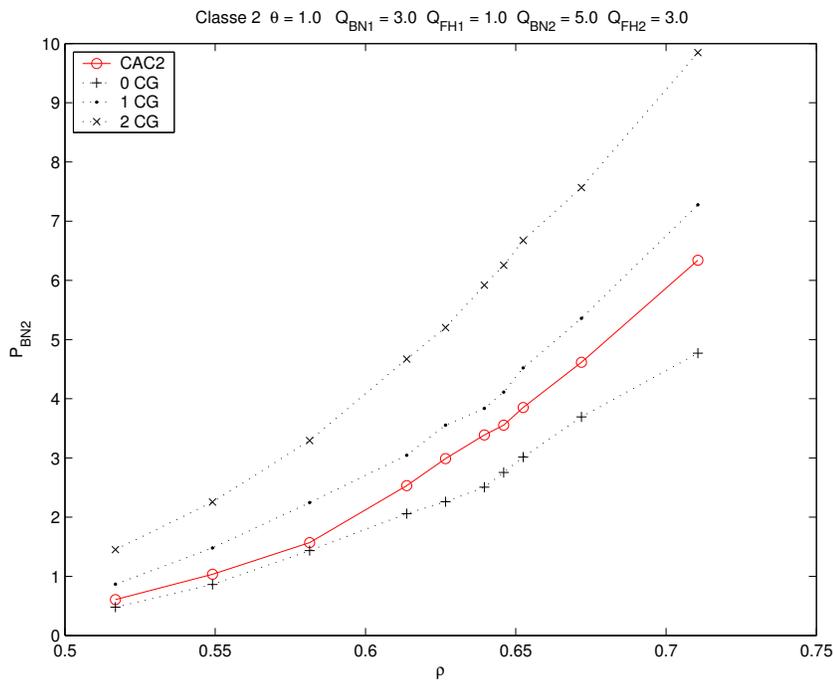


Figura 9.12: Percentual de Bloqueio de chamadas novas da classe 1 do algoritmo CAC2M.

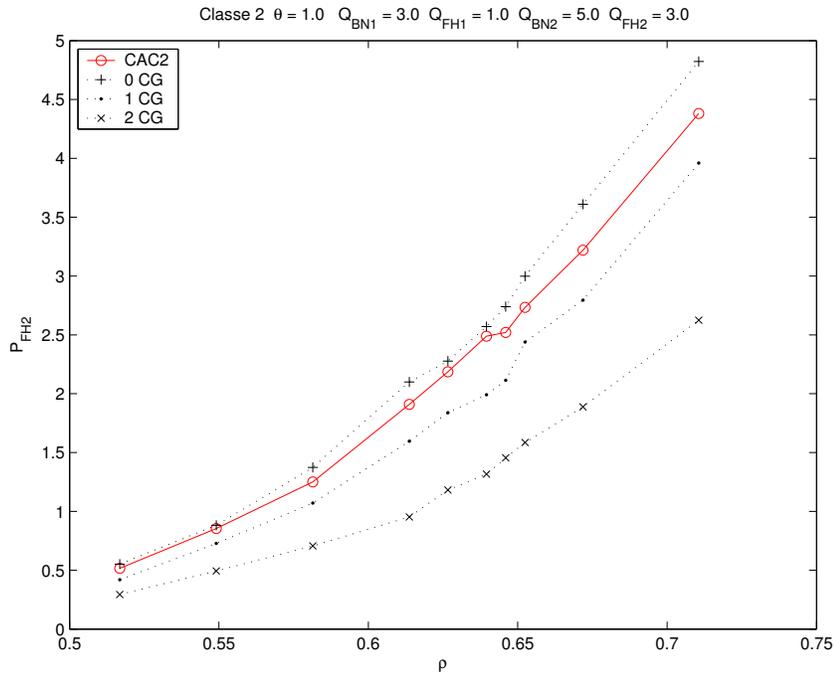


Figura 9.13: Percentual de Falha de chamadas em *handoff* da classe 1 do algoritmo CAC2M.

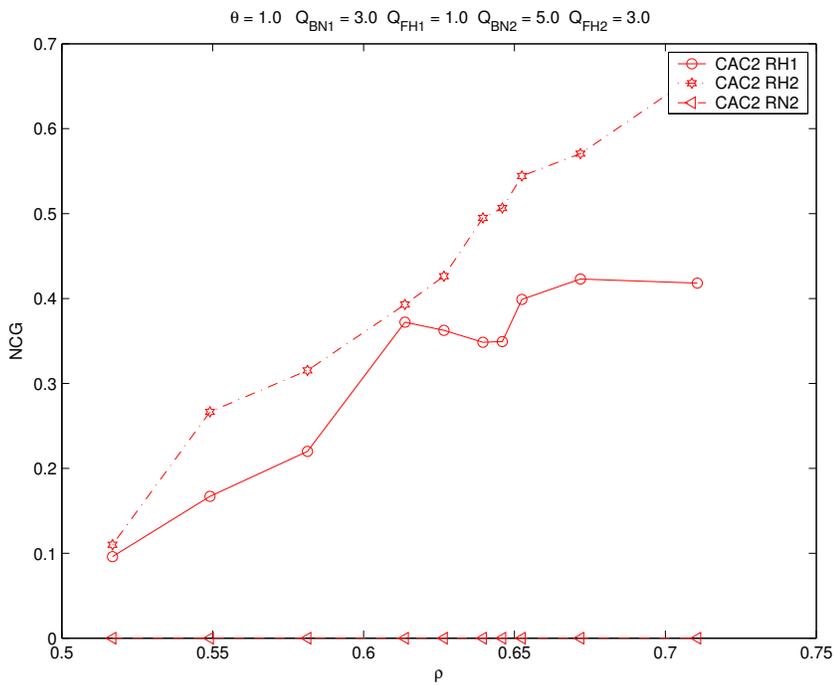


Figura 9.14: Número de Recursos reservados pelo algoritmo CAC2M.

usuários da classe 1, que utiliza menor banda, $B_1 = 1$, é reservada uma quantidade menor de recursos que para usuários das classes 2 cuja banda é $B_2 = 2$. Comparando a figura 9.14 com a figura 9.8, observa-se que a introdução do requisito de QoS para os usuários em *handoff* faz com que o número de recursos reservados para ambas as classes seja menor. O algoritmo CAC1M corrige o número de canais reservados a cada falha de usuário em *handoff*, isto pode fazê-lo reservar demais especialmente no caso de taxa de mobilidade elevada.

9.3

Algoritmo CAC6M

O algoritmo *CAC6M* tem como objetivo fazer com que ambas as restrições de QoS de uma mesma classe deixem de ser atendidas simultaneamente, para um mesmo valor de intensidade de tráfego.

A reserva dos recursos é evitada, enquanto possível, de modo a manter elevada a utilização dos recursos. Ou seja, o algoritmo *CAC6M* não reserva recursos enquanto as medidas de percentual de bloqueio de chamadas e de percentual de falhas de *handoff* estão abaixo dos respectivos requisitos de qualidade de serviço, $(Q_{BN\gamma}; Q_{FH\gamma})$.

Mantendo a filosofia do algoritmo *CAC6* para uma classe, o objetivo de *CAC6M* é fazer com que as duas restrições de QoS de cada classe sejam atendidas simultaneamente. A partir do valor de intensidade de tráfego em que não for mais possível atender à uma restrição de qualidade de serviço de determinada classe, o mesmo deve ocorrer com a outra restrição da mesma classe.

Nas proximidades do valor de intensidade de tráfego de entrada para o qual não é mais possível atender aos requisitos de qualidade de serviço, são feitos ajustes nos valores dos recursos reservados para cada classe. Para permitir a reserva de recursos aos usuários de classes de maior prioridade que iniciam sua chamada, é feita a comparação entre as medidas de bloqueio de usuários de classes diferentes, γ e $\gamma + 1$, para incrementar, ou não, os recursos reservados aos usuários novos da classe $\gamma + 1$.

Portanto, o algoritmo determina a quantidade de recursos reservados de uma forma dinâmica, de acordo com a proximidade do ponto em que não é mais possível o atendimento aos requisitos de QoS. Deve-se observar que o algoritmo *CAC6M*, além de reservar recursos para os usuários em *handoff*, também reserva recursos para os usuários novos de classes de maior

prioridade. O que é uma novidade em relação aos algoritmos *CAC1M* e *CAC2M*.

9.3.1
Descrição do Algoritmo CAC6M

Chegada de chamada em *handoff*, ou nova, da classe γ , verifica se existem recursos e aloca, ou bloqueia; registrando o sucesso, ou falha, de alocação na janela de registro correspondente ao tipo de chamada.

Calcula as medidas de falha de chamadas em *handoff*, $B_{FH\gamma}$, e de bloqueio de chamadas, $B_{BN\gamma}$.

- Se $B_{FH\gamma} \geq Q_{FH\gamma}$ e $B_{BN\gamma} \leq Q_{BN\gamma}$, $R_{H\gamma}$ é incrementado.
- Se $B_{FH\gamma} \leq Q_{FH\gamma}$ e $B_{BN\gamma} \geq Q_{BN\gamma}$, $R_{H\gamma}$ é decrementado.
- Se $B_{BN\gamma} \leq Q_{BN\gamma}$ e $B_{BN(\gamma+1)} \geq Q_{BN(\gamma+1)}$, $R_{N(\gamma+1)}$ é incrementado.
- Se $B_{BN\gamma} \geq Q_{BN\gamma}$ e $B_{BN(\gamma+1)} \leq Q_{BN(\gamma+1)}$, $R_{N(\gamma+1)}$ é decrementado.

9.3.2
Resultados do Algoritmo CAC6M

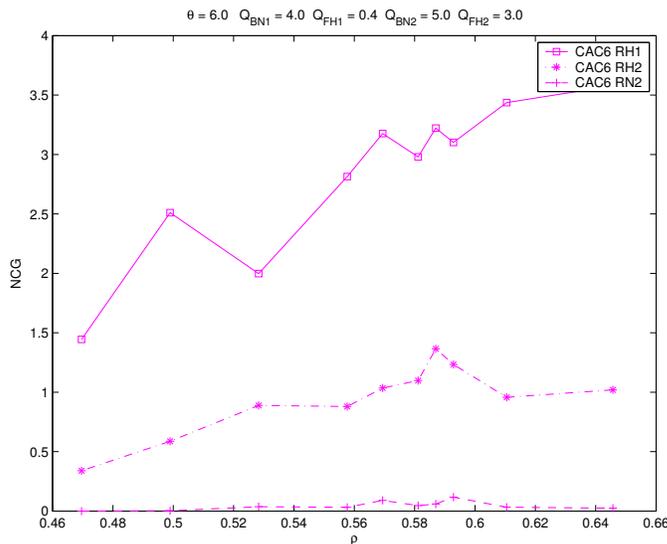


Figura 9.15: Número de Recursos reservados pelo algoritmo CAC6M.

Na figura 9.15 é apresentada a quantidade de recursos reservada pelo algoritmo *CAC6M* para cada um dos tipos de usuário. A taxa

de mobilidade é $\theta = 6$ e os requisitos de QoS são dados pelo vetor $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (4, 0.4, 5, 3)$. Observa-se que o algoritmo *CAC6M* reserva recursos para os usuários em *handoff* das classes 1 e 2, e para os usuários novos da classe 2. A taxa de mobilidade é alta então, devido ao elevado número de usuários em *handoff*, são necessários mais recursos reservados para os usuários H1 e H2. Além disso, como o requisito de usuários em *handoff* da classe 1, $Q_{FH1} = 0.4$, é um percentual muito menor que os demais requisitos, esta classe tende a obter mais recursos reservados.

As figuras 9.16, 9.17, 9.18, 9.19 e 9.20 apresentam o comportamento do algoritmo *CAC6M* no caso de taxa de mobilidade $\theta = 2$ e requisitos de QoS $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, versus intensidade de tráfego de entrada normalizada, ρ .

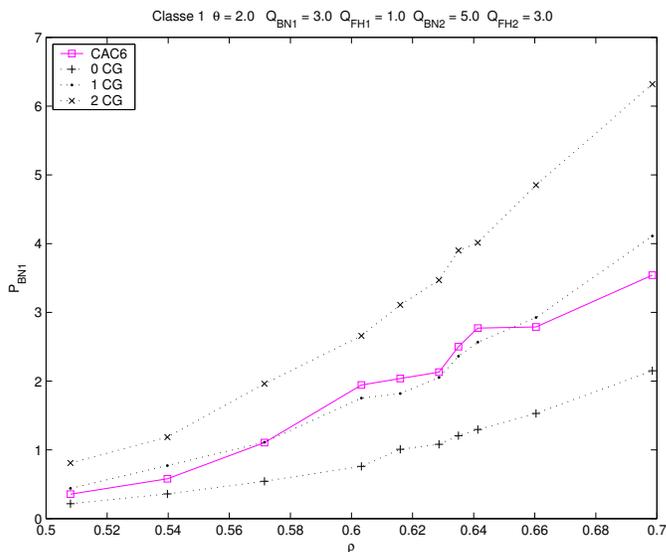


Figura 9.16: Percentual de Bloqueio de chamadas novas da classe 1 do algoritmo CAC6M.

Na figura 9.20 é apresentada a quantidade de recursos reservada para cada um dos tipos de usuário. Observa-se que o algoritmo *CAC6M* reserva recursos para os usuários em *handoff* das classes 1 e 2 e para os usuários novos da classe 2. A reserva para os usuários em *handoff* de classes diferentes é feita de forma independente, por isso as suas curvas se cruzam, eventualmente. Da mesma forma que o algoritmo *CAC6* desenvolvido para uma classe de usuários, este algoritmo busca o número adequado de recursos a ser reservado simplesmente através do ajuste do valor dos recursos nas proximidades do limite de intensidade de tráfego para o qual é possível atender os requisitos de qualidade de serviço.

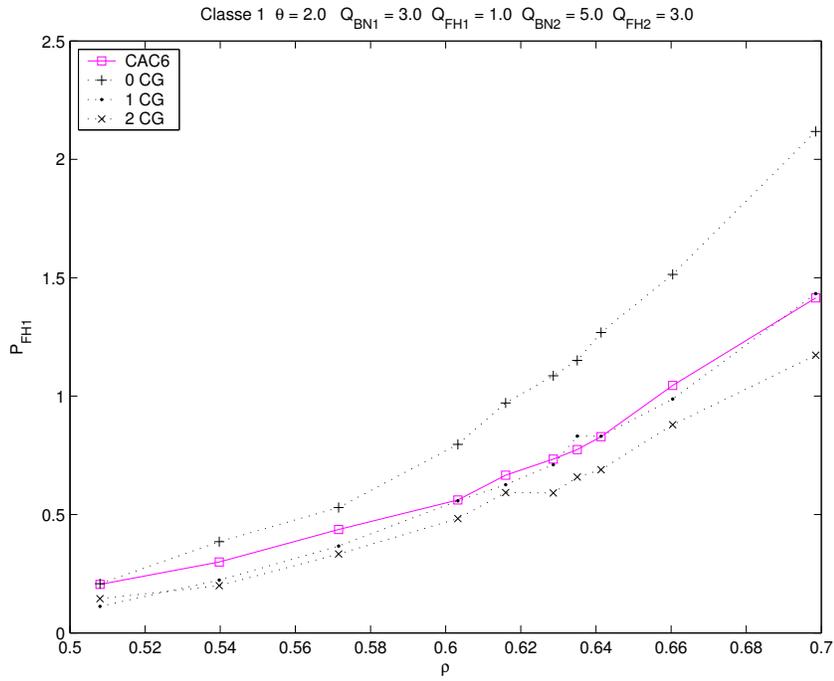


Figura 9.17: Percentual de Falha de chamadas em *handoff* da classe 1 do algoritmo CAC6M.

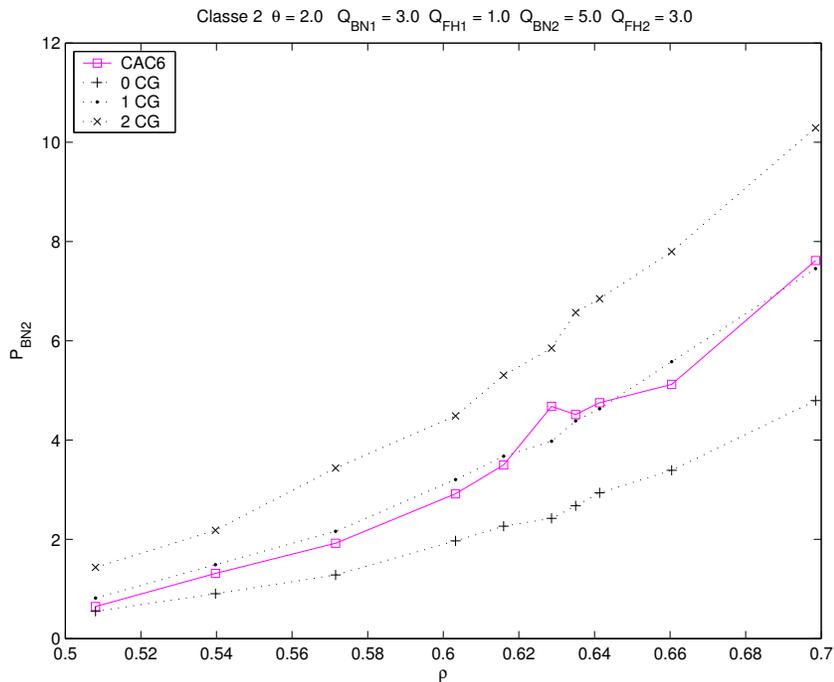


Figura 9.18: Percentual de Bloqueio de chamadas novas da classe 1 do algoritmo CAC6M.

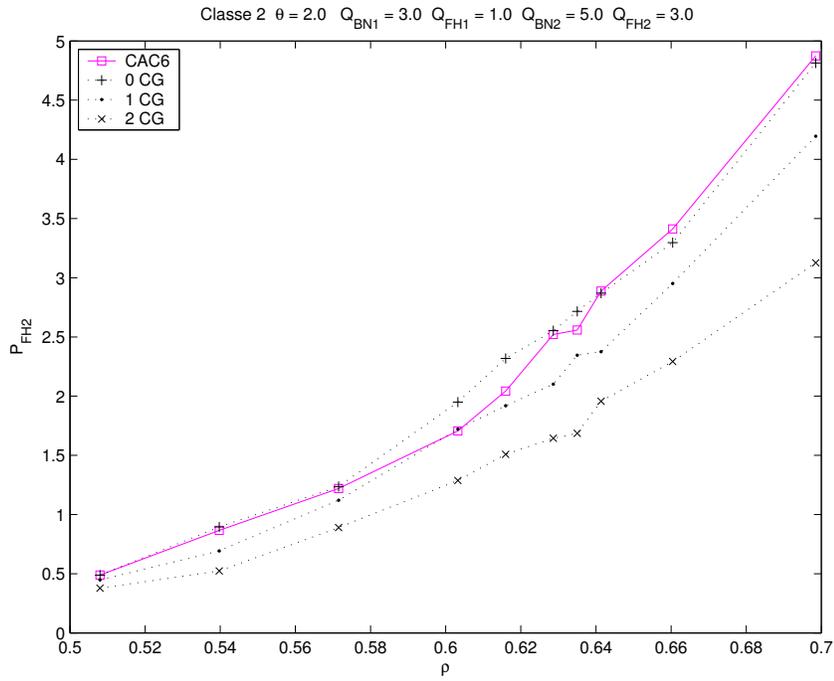


Figura 9.19: Percentual de Falha de chamadas em *handoff* da classe 1 do algoritmo CAC6M.

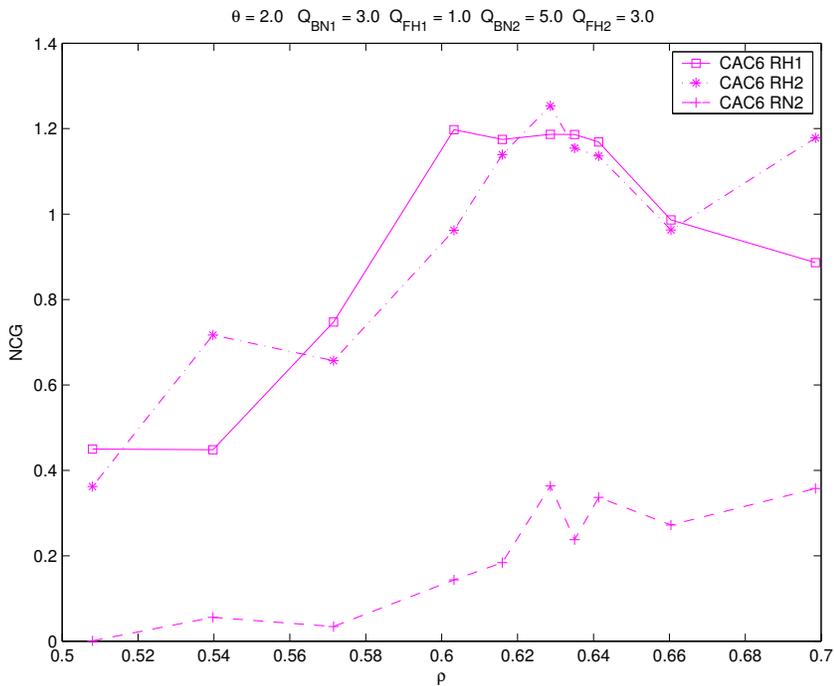


Figura 9.20: Número de Recursos reservados pelo algoritmo CAC6M.

9.4

Algoritmo CAC7M

O objetivo deste algoritmo é trabalhar com o valor ótimo de recursos reservados para cada classe de usuário, de acordo com as condições do tráfego em andamento no sistema e com os requisitos de qualidade de serviço, supondo que o tempo de permanência do usuário na célula e, conseqüentemente, o tempo de retenção do recurso sejam dados pela distribuição exponencial. Este algoritmo é uma extensão de CAC7.

Com base na suposição das distribuições exponenciais para modelar o tempo de duração da chamada e o tempo de permanência do usuário na célula, o tempo de retenção do canal também é dado por uma distribuição exponencial. Desta forma, empregando a teoria das filas, é possível calcular a quantidade de recursos a serem reservados para cada tipo de usuário.

No entanto, este algoritmo é testado para outras distribuições e apresenta bons resultados quando comparado com os algoritmos fixos e com os demais algoritmos apresentados neste trabalho.

9.4.1

Otimização dos Recursos

No caso de duas classes de usuários, há quatro requisitos de QoS, $2M = 4$, e três valores de recursos a serem definidos, $(2M - 1) = 3$.

Para analisar o problema do número ótimo de recursos a ser reservado para cada classe, são definidos dois vetores: \mathbf{Q} , cujas componentes são os requisitos de qualidade de serviço, e \mathbf{R} , cujas componentes são o número de canais de guarda reservados para cada tipo de usuário.

$$\mathbf{Q} = \begin{pmatrix} Q_{BN1} & Q_{FH1} & Q_{BN2} & Q_{FH2} \end{pmatrix} \quad (9-9)$$

$$\mathbf{R} = \begin{pmatrix} 0 & R_{H1} & R_{N2} & R_{H2} \end{pmatrix} \quad (9-10)$$

Na notação empregada neste trabalho, o vetor \mathbf{R} corresponde aos recursos que estão guardados, impedidos de serem utilizados.

Cada tipo de usuário terá acesso ao total de recursos do sistema N_c menos os canais reservados aos outros usuários de sua mesma classe γ , conforme o vetor \mathbf{R}^a definido em (9-11).

$$\mathbf{R}^a = \begin{pmatrix} N_c - R_{H1} & N_c & N_c - R_{H2} & N_c - R_{N2} \end{pmatrix} \quad (9-11)$$

Ou seja, os usuários em *handoff* da classe γ tem acesso à todos os canais menos aos que são os canais reservados aos usuários novos da mesma classe.

$$R_{H\gamma}^a = N_c - R_{N\gamma} \quad (9-12)$$

Da mesma forma, os usuários novos da classe γ tem acesso à todos os canais menos aos que são os canais de guarda dos usuários em *handoff* da mesma classe.

$$R_{N\gamma}^a = N_c - R_{H\gamma} \quad (9-13)$$

Na equação (9-10), atentando para o problema da definição dos possíveis valores de cada componente R_i , do vetor de recursos reservados \mathbf{R} , que atendem aos respectivos requisitos de QoS, nota-se que há uma região limitante para estes valores. Por exemplo, dado o requisito de qualidade de serviço para os usuários em *handoff* da classe 1, Q_{FH1} , há um valor mínimo de recursos reservados para os usuários de *handoff* da classe 1, R_{H1} , considerando que os recursos reservados para os outros tipos de usuários sejam nulos.

Por outro lado, há também três valores máximos de R_{H1} determinados por Q_{BN1} , Q_{BN2} e Q_{FH2} . Neste caso, os limites são observados considerando apenas a guarda de canais para os usuários de *handoff* da classe 1, $H1$. Se considerarmos a reserva de recursos também para as demais classes, a tendência é de que R_{H1} seja maior para garantir o mesmo requisito Q_{FH1} .

Considerando o atendimento simultâneo a todas as restrições, nosso objetivo é encontrar o vetor ideal \mathbf{R} que garanta a maior utilização dos recursos atendendo ao vetor de requisitos de QoS \mathbf{Q} . Deseja-se obter tal vetor para vários valores de intensidade de solicitação de recursos, ρ , e várias taxas de mobilidade, θ . Porém, esta solução nem sempre existe.

O problema pode ser expresso de forma matemática definindo a função f que depende do vetor restrição \mathbf{R} , como sendo o valor unitário menos a utilização total dos recursos do sistema.

$$f(\mathbf{R}) = 1 - U \quad (9-14)$$

Deseja-se encontrar o vetor \mathbf{R} que minimiza a função $f(\mathbf{R})$, sujeito às restrições dadas nas equações (9-15), (9-16) e (9-17) de que o valor das componentes de \mathbf{R} deve estar entre zero e o total de recursos do sistema, N_c . Outro conjunto de restrições advém do fato de que as probabilidades de bloqueio devem ser inferiores aos respectivos requisitos de qualidade de

serviço, conforme as equações (9-18), (9-19), (9-20) e (9-21).

$$0 \leq R_{H1} \leq N_c \quad (9-15)$$

$$0 \leq R_{N2} \leq N_c \quad (9-16)$$

$$0 \leq R_{H2} \leq N_c \quad (9-17)$$

$$P_{BN1} \leq Q_{BN1} \quad (9-18)$$

$$P_{FH1} \leq Q_{FH1} \quad (9-19)$$

$$P_{BN2} \leq Q_{BN2} \quad (9-20)$$

$$P_{FH2} \leq Q_{FH2} \quad (9-21)$$

Utilizando o programa feito para calcular as probabilidades de bloqueio de múltiplas classes de usuário, apresentado no capítulo 8, uma nova rotina em MATLAB foi implementada para, dado os valores máximo das probabilidades de bloqueio, calcular os valores ótimos de recursos a serem reservados para cada classe de usuários. Essa rotina faz uso da função de minimização, “FMINCON”, que utiliza o método do gradiente para buscar o valor mínimo da função em questão, mediante diversas restrições que são os requisitos de qualidade de serviço. A função minimizada é dada por:

$$f(R_{H1}, R_{N2}, R_{H2}) = 1 - U \quad (9-22)$$

A solução (R_{H1}, R_{N2}, R_{H2}) é dada por uma tabela para cada uma das componentes do vetor \mathbf{R} que é diferente de zero. Cada elemento da tabela representa um cenário $(\theta; \rho)$. Os elementos correspondentes aos cenários em que a solução não existe são preenchidos com zero.

No caso de duas classes de usuários, tem-se três tabelas com os valores de R_{H1} , R_{N2} e R_{H2} , para cada vetor de requisitos de QoS, \mathbf{Q} .

As figuras 9.21, 9.22 e 9.23 apresentam os valores de recursos de reserva necessários para atender os requisitos de QoS $\mathbf{Q}(\%) = (3; 1; 5; 3)$ se o tempo de permanência do usuário na célula é dado por uma distribuição exponencial. Os gráficos indicam o número ótimo de recursos para os

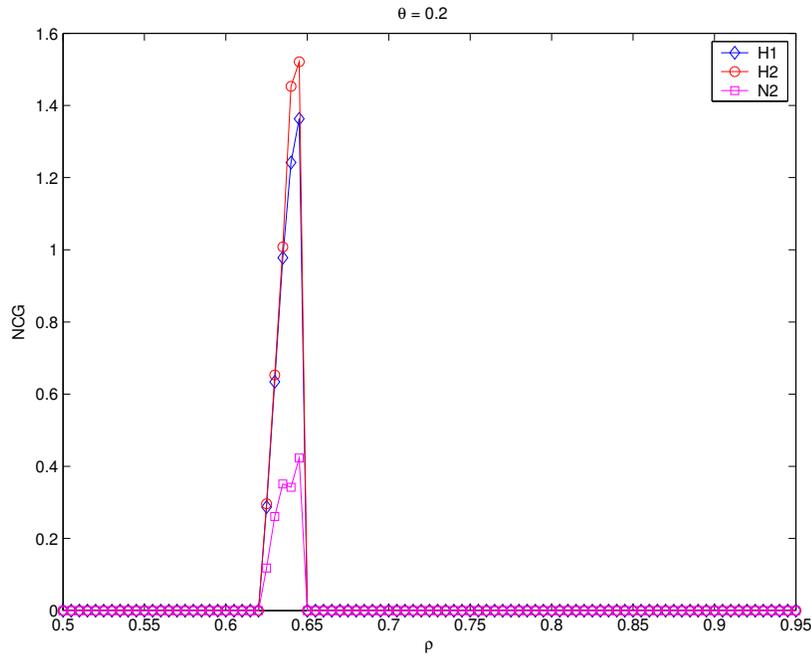


Figura 9.21: Resultados da Reserva de Recursos ótima para os usuários em *handoff* da classe 1, H_1 , em *handoff* da classe 2, H_2 , e novos da classe 2, N_2 , versus intensidade de tráfego normalizada, para mobilidade $\theta = 0,2$ e requisito de QoS $\mathbf{Q}(\%) = (3; 1; 5; 3)$, tempo de permanência exponencial.

usuários em *handoff* da classe 1, H_1 , em *handoff* da classe 2, H_2 , e novos da classe 2, N_2 , versus intensidade de tráfego normalizada, para mobilidade $\theta = 0,2$, $\theta = 2$ e $\theta = 6$, respectivamente.

Na figura 9.21 observa-se que a região de intensidade de tráfego onde é possível obter bons resultados com a reserva de recursos é pequena, $0,620 \leq \rho \leq 0,645$. Para valores de intensidade de tráfego menores que 0,615 o ideal é não reservar recursos para nenhuma classe, o que aumenta a utilização total dos recursos do sistema. Para intensidade de tráfego entre 0,620 e 0,645 há solução não trivial para o valor dos recursos \mathbf{R} que maximiza a utilização dos recursos. Para valores de ρ maiores que 0,650, não há solução possível para o valor do vetor \mathbf{R} , capaz de atender aos requisitos \mathbf{Q} e maximizar a utilização.

Aumentando a taxa de mobilidade, cujos resultados estão nas figuras 9.22 e 9.23, observa-se que a região de intensidade de tráfego onde é possível obter bons resultados com a reserva de recursos é ainda menor. No caso de taxa de mobilidade $\theta = 2$ a região de solução não trivial é $0,625 \leq \rho \leq 0,635$ e para $\theta = 6$ a região é ainda menor: $0,625 \leq \rho \leq 0,630$.

Os gráficos de número de recursos a serem reservados para os usuários H_1 , H_2 e N_2 versus a taxa de mobilidade θ estão nas figuras 9.24 e 9.25, para

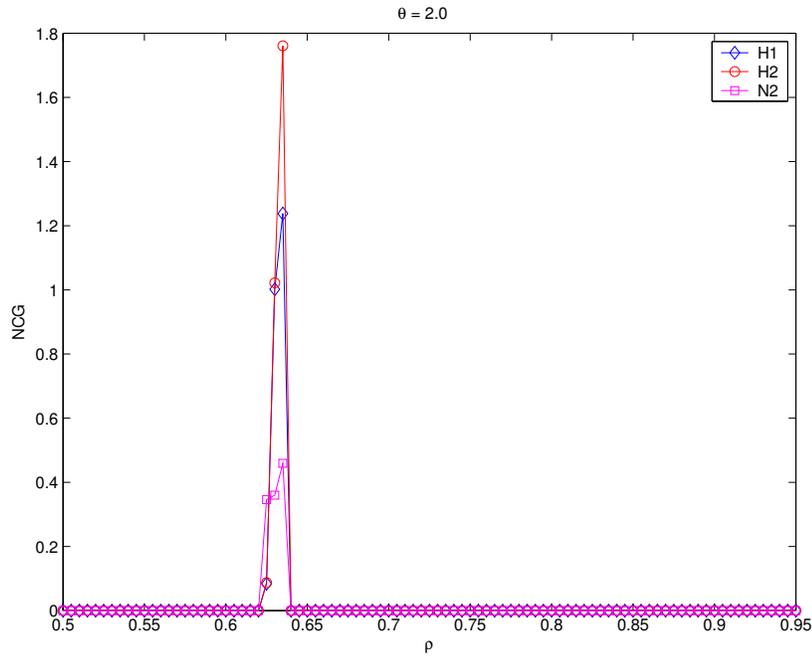


Figura 9.22: Resultados da Reserva de Recursos ótima para os usuários H_1 , H_2 e N_2 , versus intensidade de tráfego normalizada, para mobilidade $\theta = 2$ e requisito de QoS $Q(\%) = (3; 1; 5; 3)$, tempo de permanência exponencial.

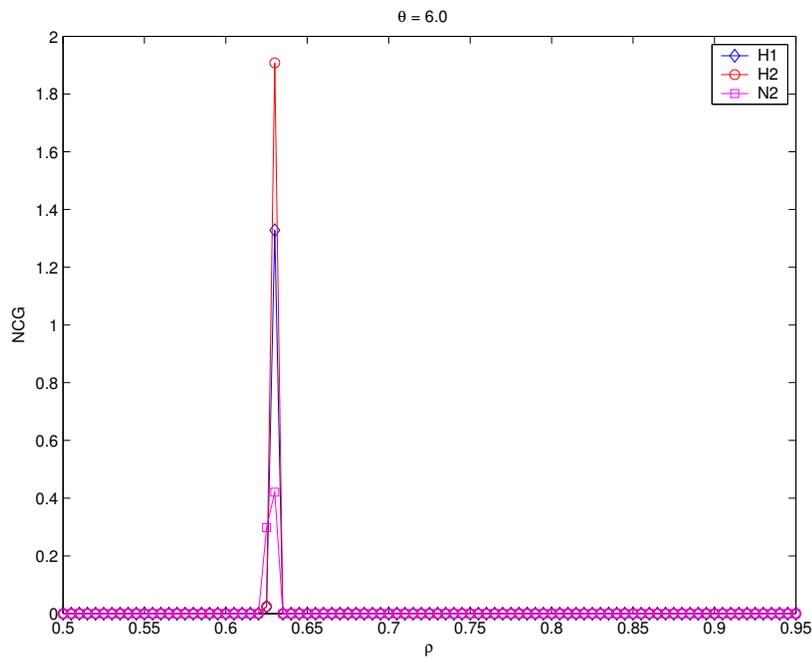


Figura 9.23: Resultados da Reserva de Recursos ótima para os usuários H_1 , H_2 e N_2 , versus intensidade de tráfego normalizada, para mobilidade $\theta = 6$ e requisito de QoS $Q(\%) = (3; 1; 5; 3)$, tempo de permanência exponencial.

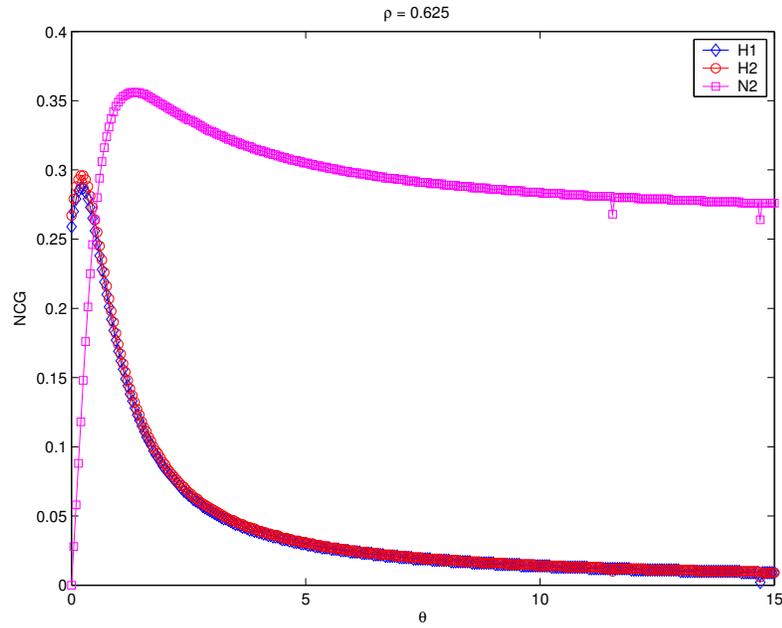


Figura 9.24: Resultados da Reserva de Recursos ótima para os usuários em *handoff* da classe 1, H_1 , em *handoff* da classe 2, H_2 , e novos da classe 2, N_2 , versus taxa de mobilidade, para intensidade de tráfego $\rho = 0,625$ e requisito de QoS $\mathbf{Q}(\%) = (3; 1; 5; 3)$, tempo de permanência exponencial.

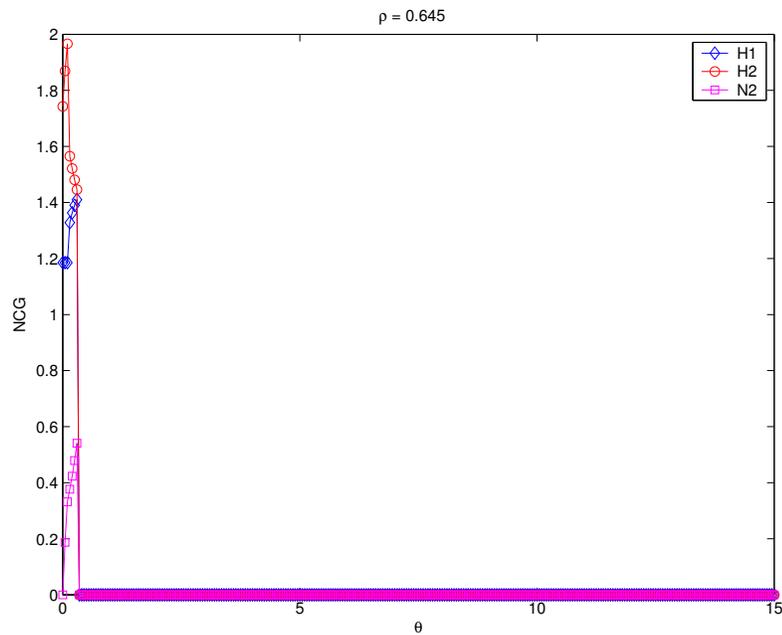


Figura 9.25: Resultados da Reserva de Recursos ótima para os usuários em *handoff* da classe 1, H_1 , em *handoff* da classe 2, H_2 , e novos da classe 2, N_2 , versus taxa de mobilidade, para intensidade de tráfego $\rho = 0,645$ e requisito de QoS $\mathbf{Q}(\%) = (3; 1; 5; 3)$, tempo de permanência exponencial.

a intensidade de tráfego iguais a $\rho = 0.625$ e $\rho = 0.645$, respectivamente. Nota-se que os valores do número de recursos é muito sensível a pequenas mudanças de ρ e θ .

Para ilustrar melhor o problema um novo conjunto de requisitos de qualidade de serviço é testado. As figuras 9.26, 9.27, 9.28, 9.29 e 9.30 apresentam os valores de recursos de reserva necessários para atender os requisitos de QoS $\mathbf{Q}(\%) = (4; 0.4; 5; 3)$ se o tempo de permanência do usuário na célula é dado por uma distribuição exponencial. Os gráficos indicam o número ótimo de recursos para os usuários em *handoff* da classe 1, H_1 , em *handoff* da classe 2, H_2 , e novos da classe 2, N_2 , versus intensidade de tráfego normalizada, para mobilidade $\theta = 0, 2, \theta = 0.5, \theta = 1, \theta = 2$ e $\theta = 6$, respectivamente.

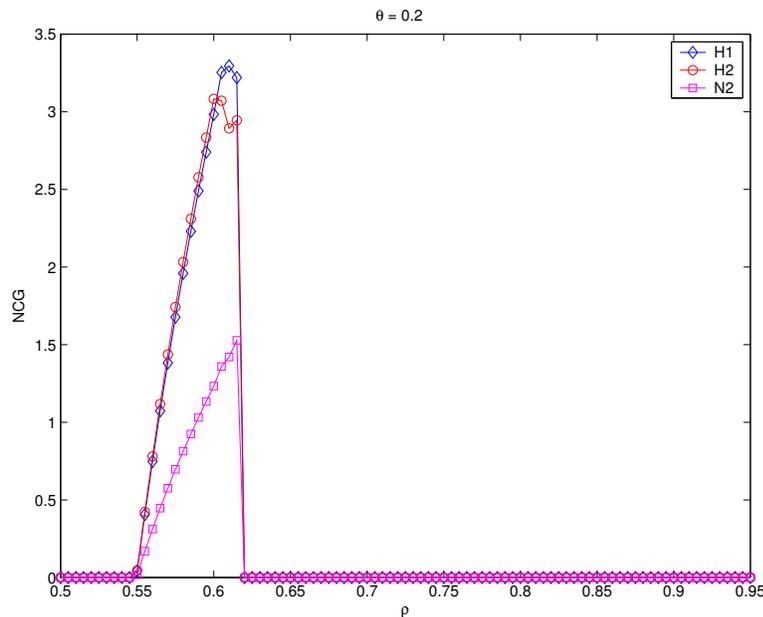


Figura 9.26: Resultados da Reserva de Recursos ótima para os usuários em *handoff* da classe 1, H_1 , em *handoff* da classe 2, H_2 , e novos da classe 2, N_2 , versus intensidade de tráfego normalizada, para mobilidade $\theta = 0, 2$ e requisito de QoS $\mathbf{Q}(\%) = (4; 0.4; 5; 3)$, tempo de permanência exponencial.

Comparando a figura 9.21 com a figura 9.26, observa-se que a região de intensidade de tráfego onde é possível obter bons resultados com a reserva de recursos aumentou do caso $Q\%(3, 1, 5, 3)$ para o caso $Q\%(4, 0.4, 5, 3)$. No caso $Q\%(3, 1, 5, 3)$ a região de solução não-trivial é de $\Delta\rho_1 = 0.02$ e no segundo conjunto de requisitos $Q\%(4, 0.4, 5, 3)$ a região de operação sobe para $\Delta\rho_2 = 0.07$, para uma taxa de mobilidade $\theta = 0.2$. Isso se deve ao maior afastamento entre os requisitos Q_{BN1} e Q_{FH1} no caso $Q\%(4, 0.4, 5, 3)$.

No caso $Q\%(4, 0.4, 5, 3)$, da figura 9.26, para valores de intensidade de tráfego menores que 0.545 o ideal é não reservar recursos para nenhuma classe, o que aumenta a utilização total dos recursos do sistema. Para intensidade de tráfego entre 0.550 e 0.620 há solução não trivial para o valor dos recursos \mathbf{R} que maximiza a utilização dos recursos. Para valores de ρ maiores que 0.625, não há solução possível para o valor do vetor R , capaz de atender aos requisitos Q e maximizar a utilização.

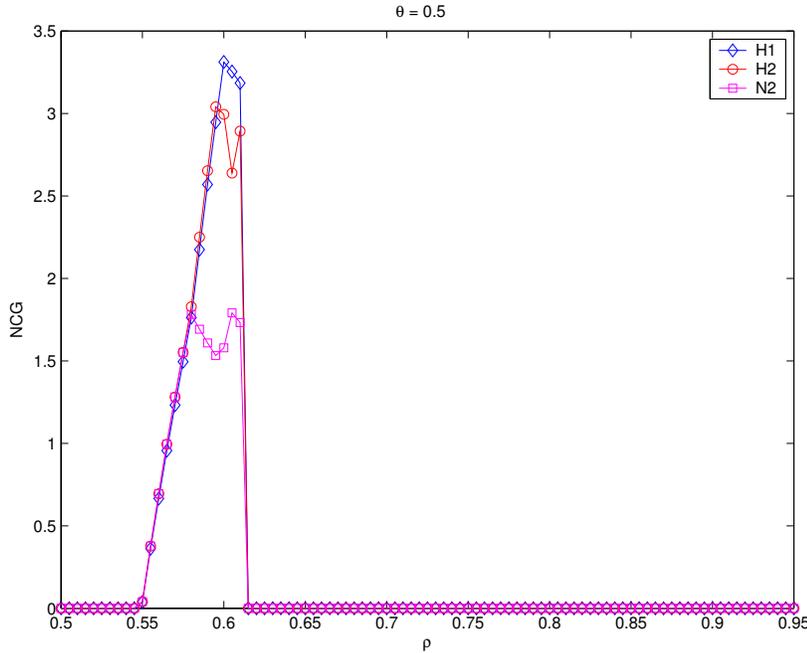


Figura 9.27: Resultados da Reserva de Recursos ótima para os usuários H_1 , H_2 e N_2 , versus intensidade de tráfego normalizada, para mobilidade $\theta = 0.5$ e requisito de QoS $Q(\%) = (4; 0.4; 5; 3)$, tempo de permanência exponencial.

Aumentando a taxa de mobilidade, para o conjunto de requisitos $Q\% = (4, 0.4, 5, 3)$, cujos resultados estão nas figuras 9.27, 9.28, 9.29 e 9.30. Na figura 9.30, a taxa de mobilidade é $\theta = 6$ e $Q\% = (4, 0.4, 5, 3)$, a região de solução não trivial é $0.550 \leq \rho \leq 0.590$, esta região é maior que a encontrada no caso $Q\%(3, 1, 5, 3)$ na figura 9.30. Observa-se que a região de intensidade de tráfego onde é possível obter bons resultados com a reserva de recursos diminui com o aumento da taxa de mobilidade e aumenta com o afastamento entre os requisitos de QoS.

Portanto, a técnica de reserva de recursos com o objetivo de atender aos requisitos de qualidade de serviço torna possível determinar o valor dos recursos a serem reservados de modo a se obter uma utilização máxima, dada a intensidade de solicitação de recursos, ρ , e a taxa de mobilidade, θ .

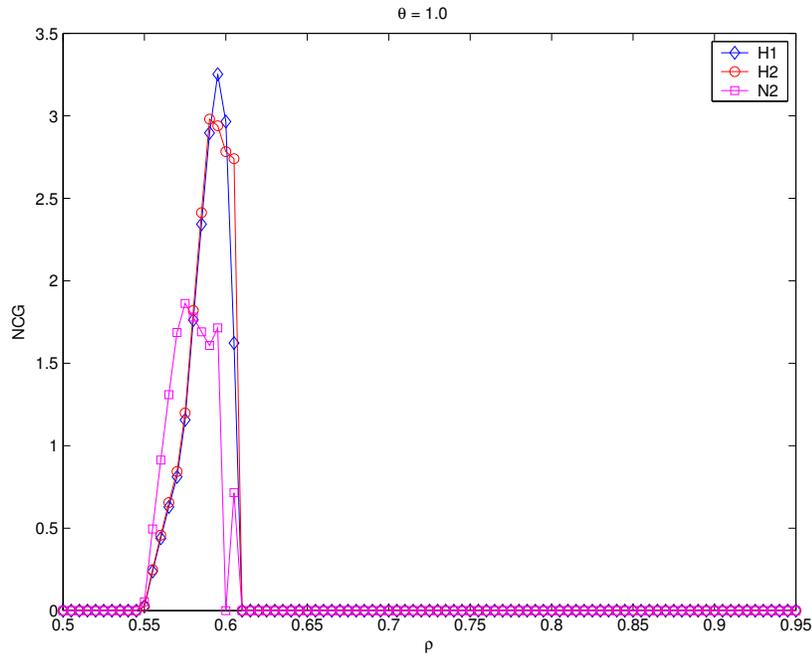


Figura 9.28: Resultados da Reserva de Recursos ótima para os usuários H_1 , H_2 e N_2 , versus intensidade de tráfego normalizada, para mobilidade $\theta = 1$ e requisito de QoS $Q(\%) = (4; 0.4; 5; 3)$, tempo de permanência exponencial.

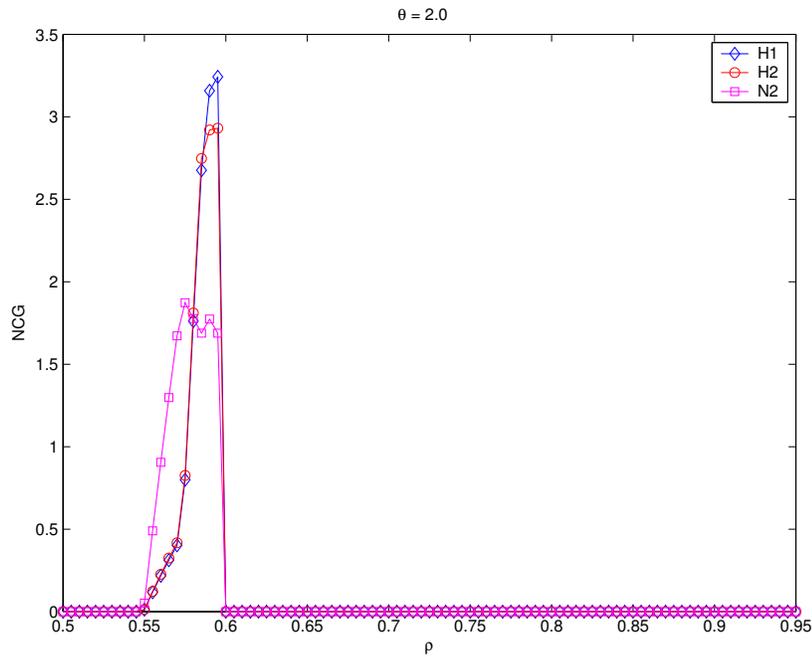


Figura 9.29: Resultados da Reserva de Recursos ótima para os usuários H_1 , H_2 e N_2 , versus intensidade de tráfego normalizada, para mobilidade $\theta = 2$ e requisito de QoS $Q(\%) = (4; 0.4; 5; 3)$, tempo de permanência exponencial.

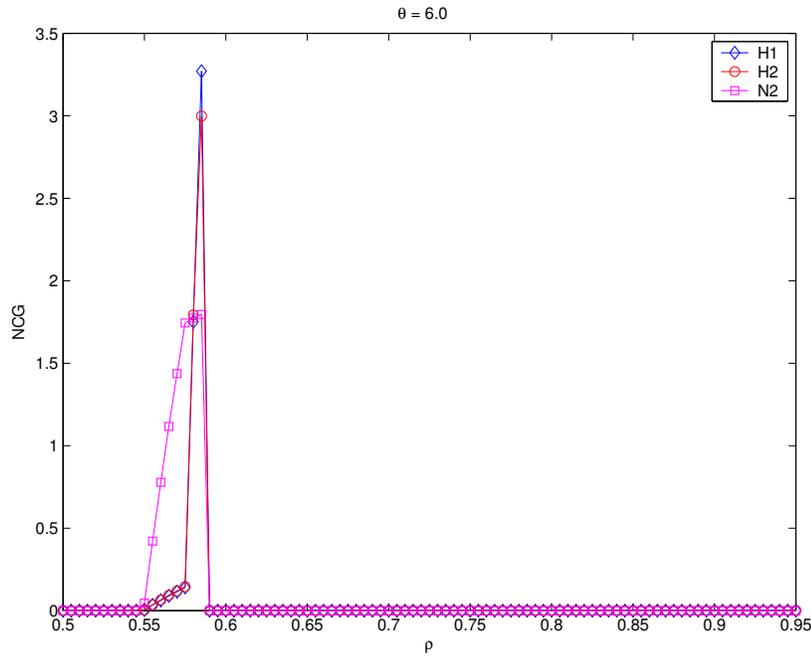


Figura 9.30: Resultados da Reserva de Recursos ótima para os usuários H_1 , H_2 e N_2 , versus intensidade de tráfego normalizada, para mobilidade $\theta = 6$ e requisito de QoS $Q(\%) = (4; 0.4; 5; 3)$, tempo de permanência exponencial.

Dada uma taxa de mobilidade θ , para valores de intensidade de solicitação de recursos baixos, o ideal é não fazer a reserva de recursos. Para uma faixa intermediária de valores de ρ , dependendo do conjunto de requisitos de QoS, $Q\%$, é possível encontrar os valores de (R_{H1}, R_{N2}, R_{H2}) que tornam a utilização máxima apenas até certo valor de intensidade de tráfego de entrada e taxa de mobilidade. Para valores de intensidade de tráfego e taxas de mobilidades superiores, não é mais possível atender aos requisitos de qualidade de serviço.

Assim como no caso de uma classe de usuários, a faixa de valores de ρ em que é vantajoso fazer a reserva de recursos é pequena.

9.4.2 Descrição do Algoritmo CAC7M

O algoritmo *CAC7* trabalha com o valor ótimo de recursos reservados para cada classe calculados pelo programa feito em MATLAB. A solução analítica só é possível para o caso em que se modela o tempo de duração da chamada e tempo de permanência do usuário na célula por distribuições exponenciais. Conseqüentemente, o tempo de retenção do recurso também

e dado por uma distribuição exponencial, o que torna possível a solução analítica por Cadeia de Markov de dimensão M .

O valor ótimo dos recursos é calculado antes do sistema iniciar seu funcionamento, os valores são arrumados em matrizes. Dado o conjunto de requisitos de QoS e o número M de classes de usuários, teremos $2M - 1$ matrizes a serem calculadas, uma para cada tipo de usuário, excluindo os usuários de classe de menor prioridade para a qual nenhum recurso é reservado. Cada elemento da matriz corresponde a uma dada taxa de mobilidade e intensidade de tráfego de entrada, $(\theta; \rho)$ e vetor de requisitos de qualidade de serviço \mathbf{Q} .

A estratégia do algoritmo *CAC7M* é medir as características do tráfego em andamento e estimar o valor de $(\theta; \rho)$. A medida é feita periodicamente e o valor dos recursos a ser reservado para cada classe é corrigido de acordo com a consulta às respectivas matrizes. No caso da solução não existir, para manter a continuidade entre os valores, as matrizes são preenchidas com o valor da solução mais próxima existente.

9.4.3

Resultados do Algoritmo *CAC7M*

As figuras 9.31, 9.32, 9.33, 9.34 e 9.35 apresentam o comportamento do algoritmo *CAC7M* no caso de taxa de mobilidade $\theta = 2$ e requisitos de QoS $\mathbf{Q} = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, versus intensidade de tráfego de entrada normalizada, ρ .

Na figura 9.31 é apresentada a quantidade de recursos reservada para cada um dos tipos de usuário no caso de taxa de mobilidade $\theta = 2$ e requisitos de QoS $\mathbf{Q} = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$. Observa-se que o algoritmo *CAC7M* busca acompanhar a tabela com os valores ideais do número de recursos a ser reservado. Como o valor do número de recursos tem uma mudança muito acentuada com a intensidade de tráfego e com a taxa de mobilidade, sistemas que fossem mais eficientes em medir as características do tráfego em andamento e estimar o valor de $(\theta; \rho)$, fariam um melhor proveito do algoritmo *CAC7M*.

As figuras 9.32 e 9.33, apresentam o comportamento do algoritmo *CAC7M* em relação ao bloqueio de novas chamadas e de chamadas em *handoff* para a classe 1, $\theta = 2$ e $\mathbf{Q}\% = (3, 1, 5, 3)$, versus intensidade de tráfego de entrada normalizada, ρ . Observa-se na figura 9.32, o percentual de bloqueio de chamadas novas se mantém abaixo do requisito $Q_{BN1} = 3\%$. Da mesma forma, na figura 9.33, tem-se que as chamadas em *handoff* da classe

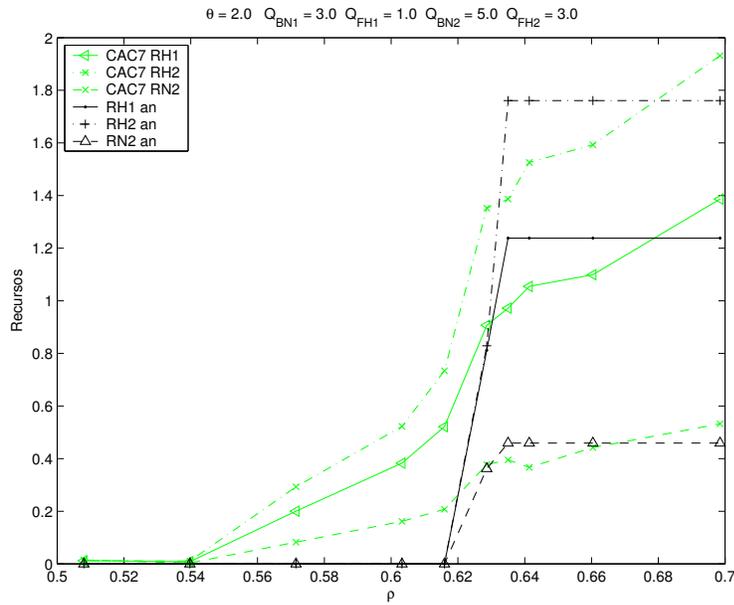


Figura 9.31: Número de Recursos reservados pelo algoritmo CAC7M, $\theta = 2$ e $Q\% = (3, 1, 5, 3)$.

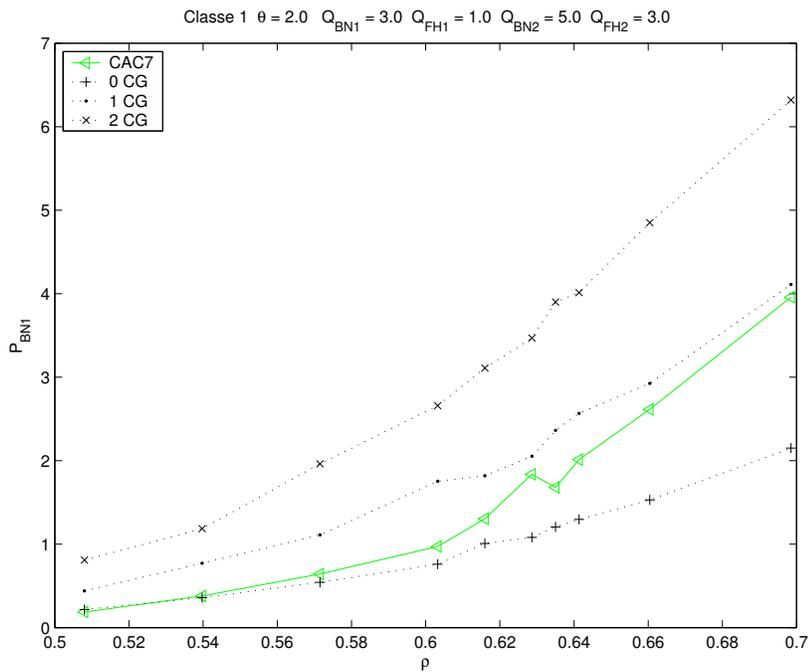


Figura 9.32: Percentual de Bloqueio de chamadas novas da classe 1 do algoritmo CAC7M, $\theta = 2$ e $Q\% = (3, 1, 5, 3)$.

1 são realmente favorecidas pelo algoritmo CAC7, mantendo-se abaixo do requisito $Q_{FH1} = 1\%$ para uma grande quantidade de valores de intensidade

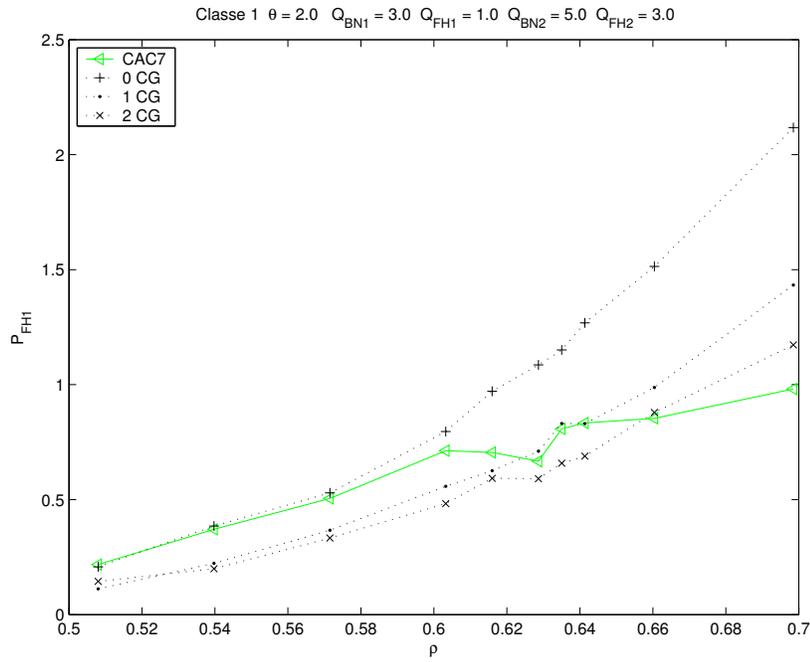


Figura 9.33: Percentual de Falha de chamadas em *handoff* da classe 1 do algoritmo CAC7M, $\theta = 2$ e $Q\% = (3, 1, 5, 3)$.

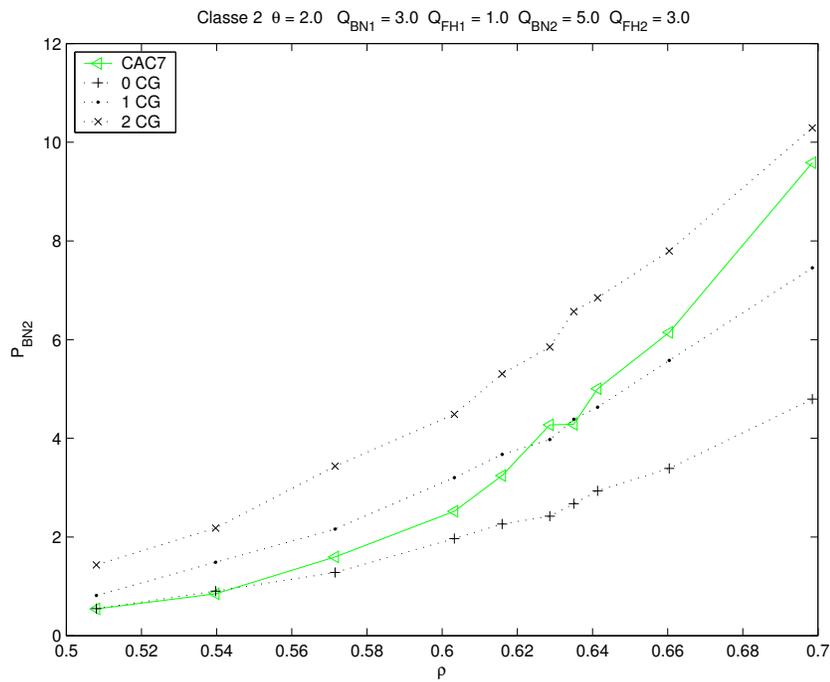


Figura 9.34: Percentual de Bloqueio de chamadas novas da classe 2 do algoritmo CAC7M, $\theta = 2$ e $Q\% = (3, 1, 5, 3)$.

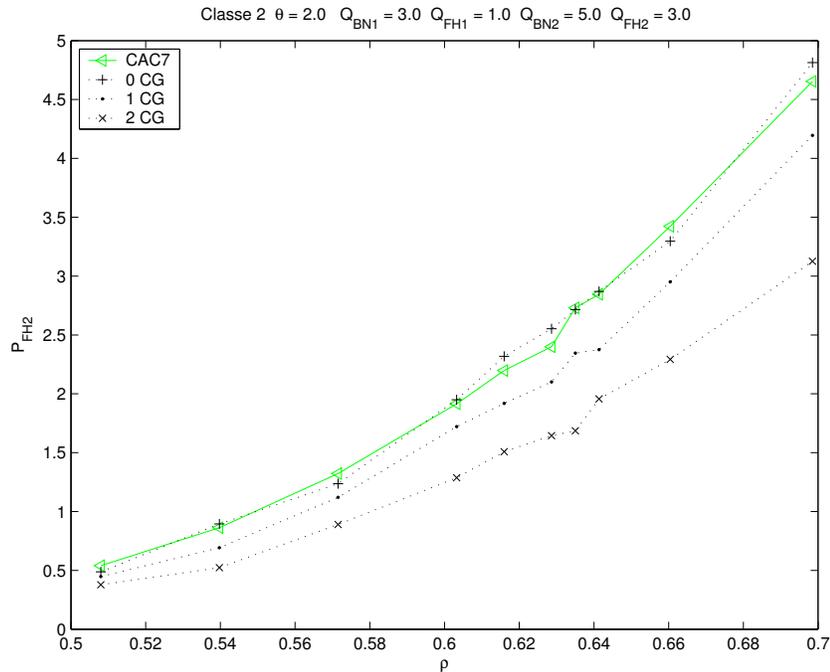


Figura 9.35: Percentual de Falha de chamadas em *handoff* da classe 2 do algoritmo CAC7M, $\theta = 2$ e $Q\% = (3, 1, 5, 3)$.

de solicitação de recursos.

Para a classe 2, as figuras 9.34 e 9.35 demonstram que as curvas de P_{BN2} e P_{FH2} se mantêm abaixo dos respectivos requisitos $Q_{BN2} = 5\%$ e $Q_{FH2} = 3\%$ por uma intensidade de solicitação de recursos de entrada superior a dos algoritmos fixos. Além disto, por manter valores baixos de reserva para valores baixos de intensidade de tráfego, o algoritmo CAC7 tende a aumentar a utilização total dos recursos.

9.5 Comparação do Desempenho dos Algoritmos

Esta seção comparará o desempenho dos algoritmos diante de vários cenários diferentes. Em cada cenário simulado há uma dada distribuição para o tempo de permanência do usuário na célula, uma dada taxa de mobilidade θ e vários valores de intensidade de tráfego ρ .

Da mesma forma que no caso de uma classe de usuários, os parâmetros F e I são utilizados para efetuar a comparação de desempenho no caso de duas classes. Os parâmetros são utilizados para cada classe separadamente.

É utilizado o mesmo modelo para o tráfego das duas classes de usuário.

O tempo de retenção do recurso é dado pelo mínimo entre a distribuição exponencial, que representa a duração da chamada, e a distribuição que modela o tempo de permanência: distribuição constante, distribuição uniforme, distribuição exponencial e distribuição Weibull.

Inicialmente o algoritmo fixo de um canal de guarda é testado para diversas distribuições que modelam o tempo de permanência do usuário na célula. Então, são comparados os algoritmos propostos e os fixos diante de cada uma das quatro distribuições. Finalmente, é verificado como cada algoritmo se porta diante das distribuições propostas para o tempo de permanência do usuário na célula. São testados casos em que a mobilidade é alta, média e baixa e casos em que os requisitos de qualidade de serviço são diferentes.

9.5.1

Parâmetros \mathbf{F} e \mathbf{I} para Comparar o Desempenho

Os algoritmos propostos para duas classes de usuários tem seu comportamento comparado através do parâmetro \mathbf{F} definido em (7-1) e do parâmetro \mathbf{I} definido na equação (5-6). Os dois parâmetros são calculados para cada classe individualmente.

O parâmetro \mathbf{F} considera a proximidade entre cada requisito de QoS e a respectiva Probabilidade de Bloqueio. Os termos de proximidade de chamadas novas e de chamadas em *handoff* são somados, ponderando o termo das chamadas em *handoff* pela taxa de mobilidade θ . Quanto **maior** \mathbf{F} , melhor e quanto maior o valor de ρ antes de \mathbf{F} chegar a zero, melhor.

O parâmetro \mathbf{I} é dado pela razão entre duas intensidades de tráfego máximas atendendo aos requisitos de qualidade de serviço. No numerador tem-se o valor máximo de intensidade de tráfego alcançado pelo algoritmo, ρ_Q^{CAC} , considerando cada uma das distribuições propostas para o tempo de permanência do usuário na célula. O denominador é o valor de intensidade de tráfego máxima obtida analiticamente no caso da distribuição exponencial, ρ_Q^* . Portanto, **o maior valor de \mathbf{I}** representa um melhor desempenho pois indica que o algoritmo alcança uma intensidade de tráfego maior dentro dos limites estabelecidos pelos requisitos de QoS.

Os parâmetros \mathbf{F} e \mathbf{I} utilizados neste capítulo, são os mesmos parâmetros definidos no caso de uma classe de usuários, $M = 1$, ou seja, dois tipos de usuários diferentes (N, H). Neste capítulo, são apresentados os algoritmos de duas classes de usuários, quatro tipos de usuários, ($N1, H1, N2, H2$); no entanto, os parâmetros \mathbf{F} e \mathbf{I} comparam como cada

algoritmo se porta para cada classe de usuários de forma individual, não há uma comparação que envolva o desempenho conjunto para as duas classes.

9.5.2

Comparação do Desempenho do Algoritmo de 1CG de Acordo com a Distribuição para o Tempo de Permanência do Usuário na Célula

Antes de iniciar a comparação entre os algoritmos propostos, esta seção apresenta o comportamento do algoritmo fixo que utiliza um canal de guarda diante das quatro distribuições que modelam o tempo de permanência do usuário na célula. São testados dois casos: alta mobilidade, $\theta = 6$, e baixa mobilidade, $\theta = 0.5$.

Caso Taxa de mobilidade $\theta = 6$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, algoritmo de 1 canal de guarda fixo.

As figuras 9.36, 9.37, 9.38 e 9.39 apresentam as curvas de percentual de bloqueio e de chamadas em *handoff* das classes 1 e 2 do algoritmo fixo de um canal de guarda para os usuários em *handoff* de ambas as classes, 1CG, taxa de mobilidade $\theta = 6$, versus intensidade de tráfego de entrada normalizada, ρ . Observa-se que, assim como no caso de uma classe, observado no capítulo 7, quase não há diferença entre os resultados considerando as diferentes distribuições. O mesmo resultado é observado nas figuras 9.48, 9.49, 9.50 e 9.51 que apresentam os parâmetros F e I das classes 1 e 2.

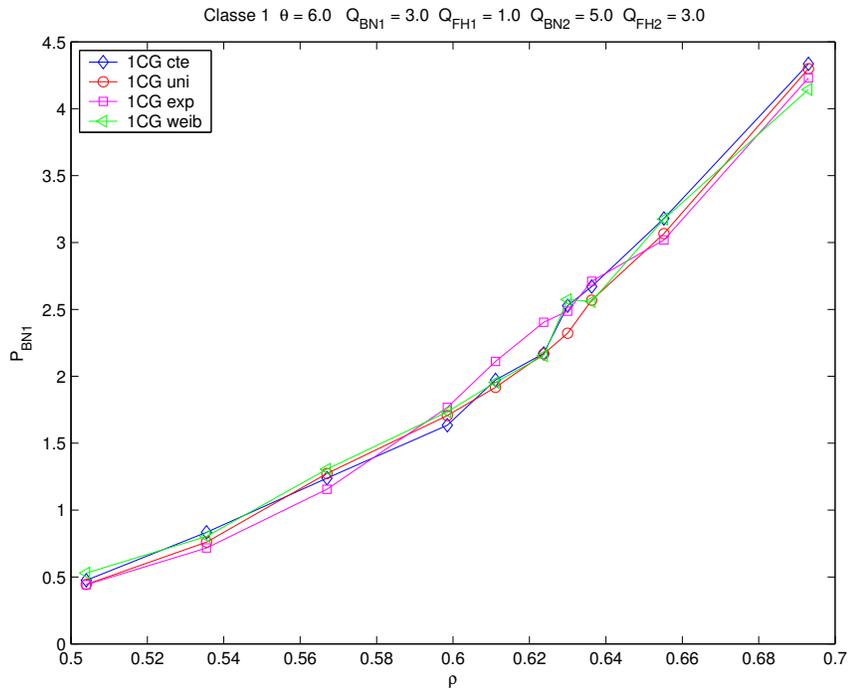


Figura 9.36: Percentual de Bloqueio de chamadas novas da classe 1 do algoritmo de 1CG, taxa de mobilidade $\theta = 6$.

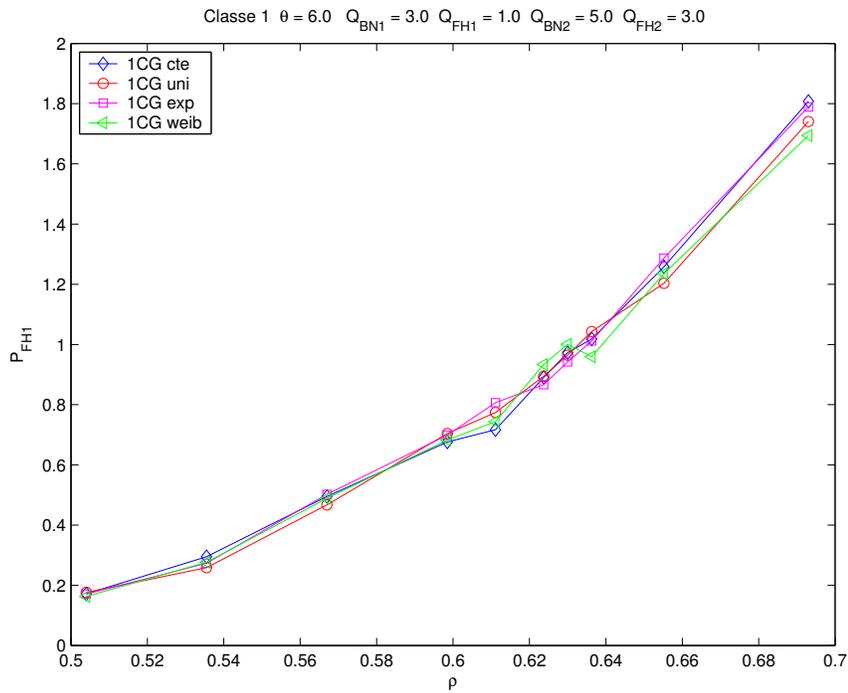


Figura 9.37: Percentual de Falha de chamadas em *handoff* da classe 1 do algoritmo de 1CG, taxa de mobilidade $\theta = 6$.

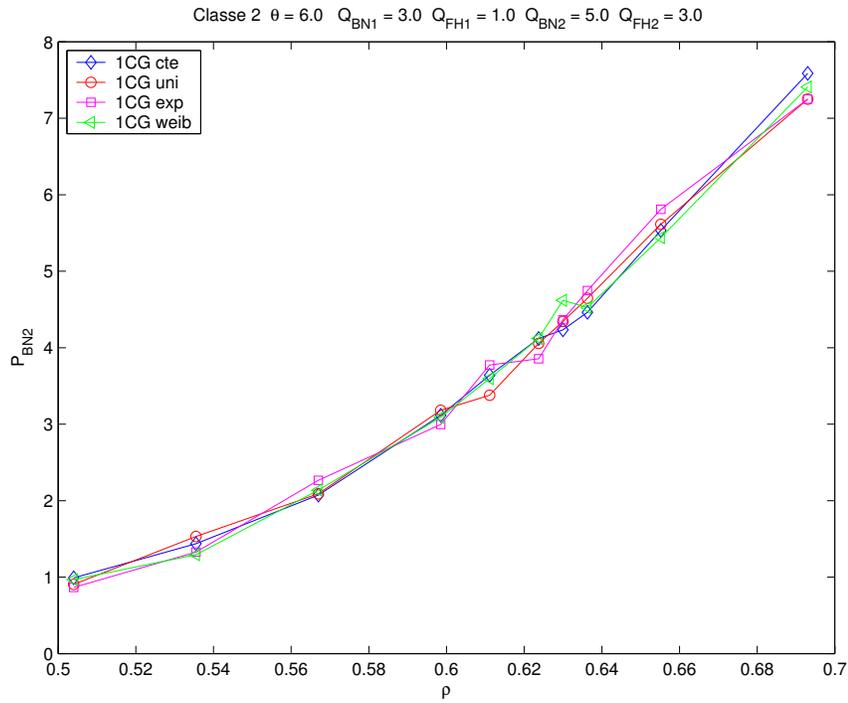


Figura 9.38: Percentual de Bloqueio de chamadas novas da classe 2 do algoritmo de 1CG, taxa de mobilidade $\theta = 6$.

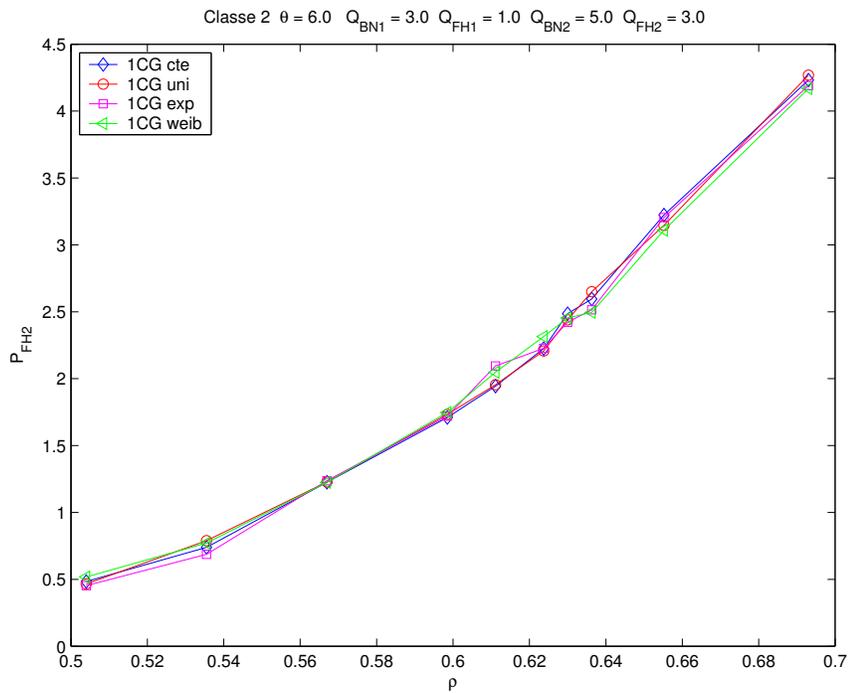


Figura 9.39: Percentual de Falha de chamadas em *handoff* da classe 2 do algoritmo de 1CG, taxa de mobilidade $\theta = 6$.

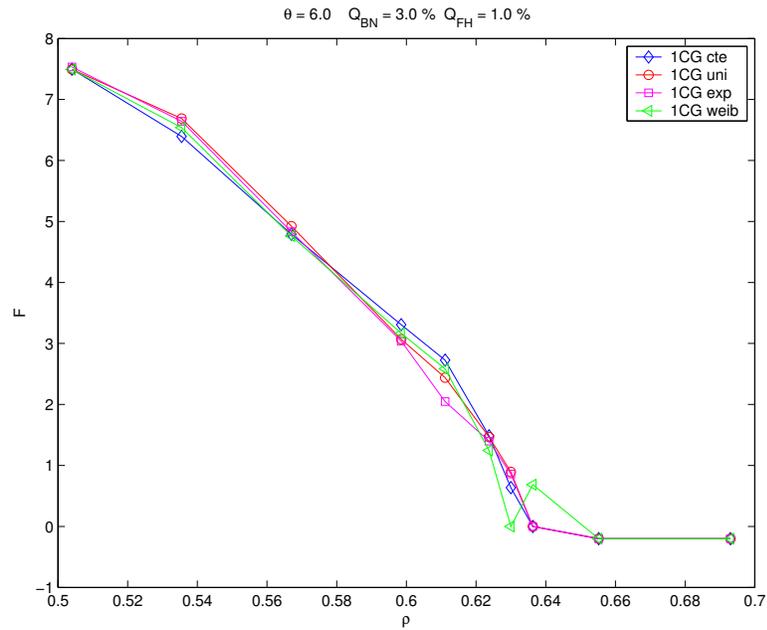


Figura 9.40: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, algoritmo fixo 1CG.

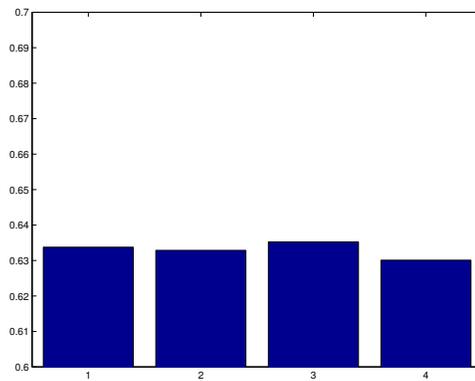


Figura 9.41: Parâmetro I para usuários da classe 1, considerando que o tempo de permanência do usuário na célula é dado pelas Distribuições (1)Constante, (2)Uniforme, (3)Exponencial, (4)Weibull, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, algoritmo fixo 1CG.

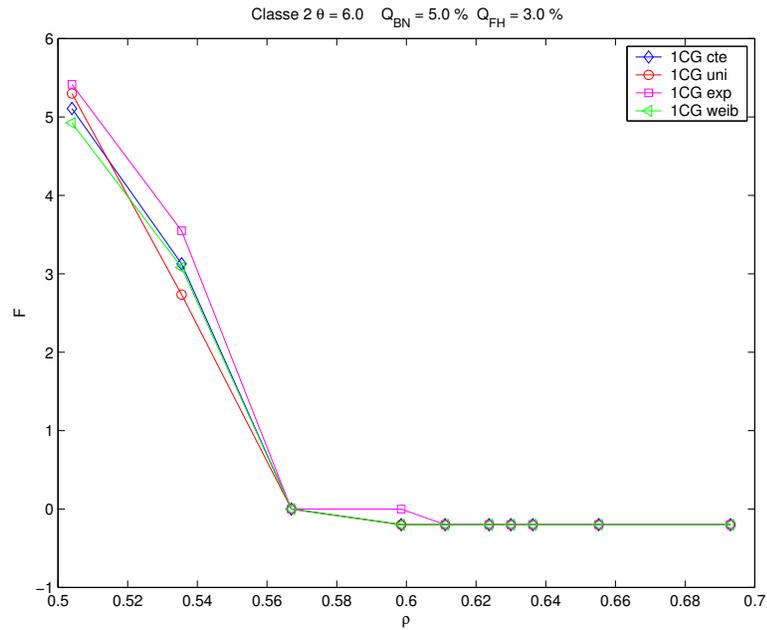


Figura 9.42: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, algoritmo fixo 1CG.

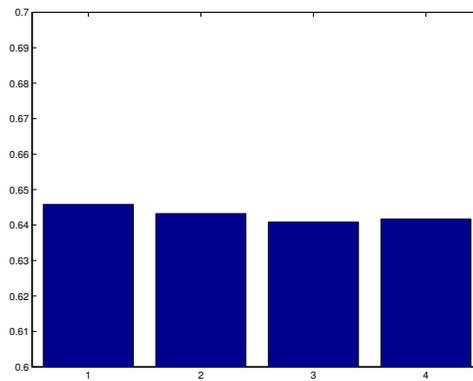


Figura 9.43: Parâmetro I para usuários da classe 2, considerando que o tempo de permanência do usuário na célula é dado pelas Distribuições (1)Constante, (2)Uniforme, (3)Exponencial, (4)Weibull, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, algoritmo fixo 1CG.

Caso Taxa de mobilidade $\theta = 0.5$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, algoritmo de 1 canal de guarda fixo.

Para verificar o comportamento do algoritmo de 1CG fixo, supondo baixa mobilidade e várias distribuições para o tempo de permanência, as figuras 9.44, 9.45, 9.46 e 9.47 apresentam os percentuais de bloqueio versus intensidade de tráfego de entrada normalizada, considerando uma mobilidade baixa no sistema, $\theta = 0.5$. Observa-se que as distribuições uniforme, exponencial, e Weibull apresentam desempenhos percentuais de bloqueio semelhantes. No entanto, no caso em que o tempo de permanência do usuário na célula é representado pela distribuição constante e baixa mobilidade, esta curva se afasta um pouco das demais. O mesmo resultado é observado nas figuras 9.40, 9.41, 9.42 e 9.43 que apresentam os parâmetros F e I das classes 1 e 2. No caso de mobilidade baixa, o valor da constante é maior do que no caso da mobilidade alta, isso faz com que ocorra um número maior de chamadas de longa duração que provocam maiores bloqueios no sistema.

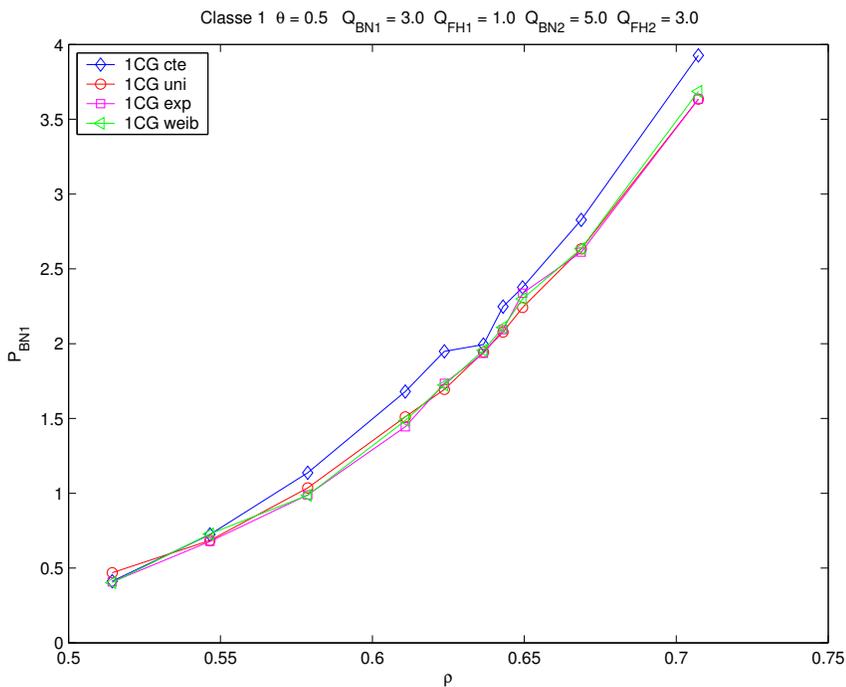


Figura 9.44: Percentual de Bloqueio de chamadas novas da classe 1 do algoritmo de 1CG, taxa de mobilidade $\theta = 0.5$.

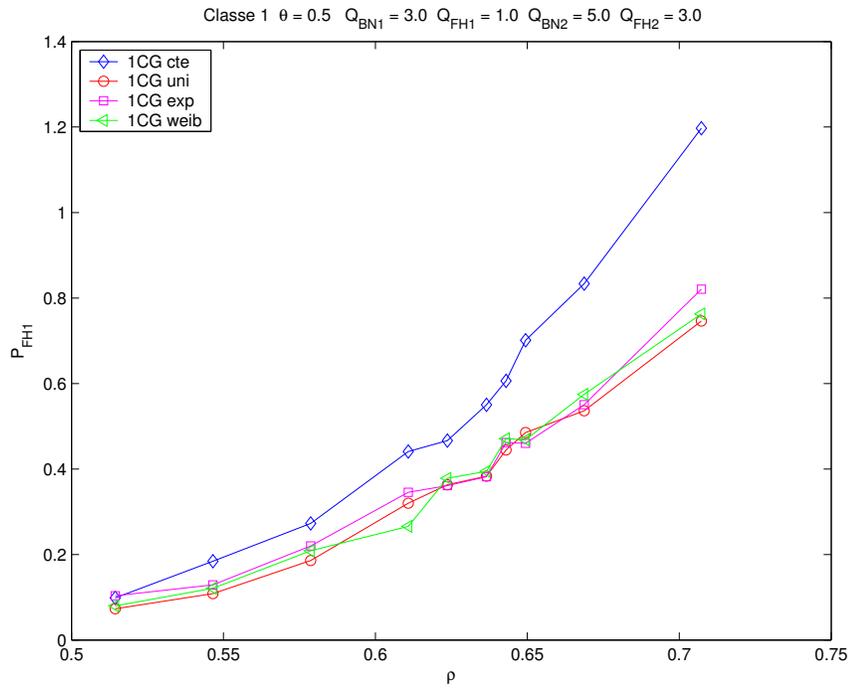


Figura 9.45: Percentual de Falha de chamadas em *handoff* da classe 1 do algoritmo de 1CG, taxa de mobilidade $\theta = 0.5$.

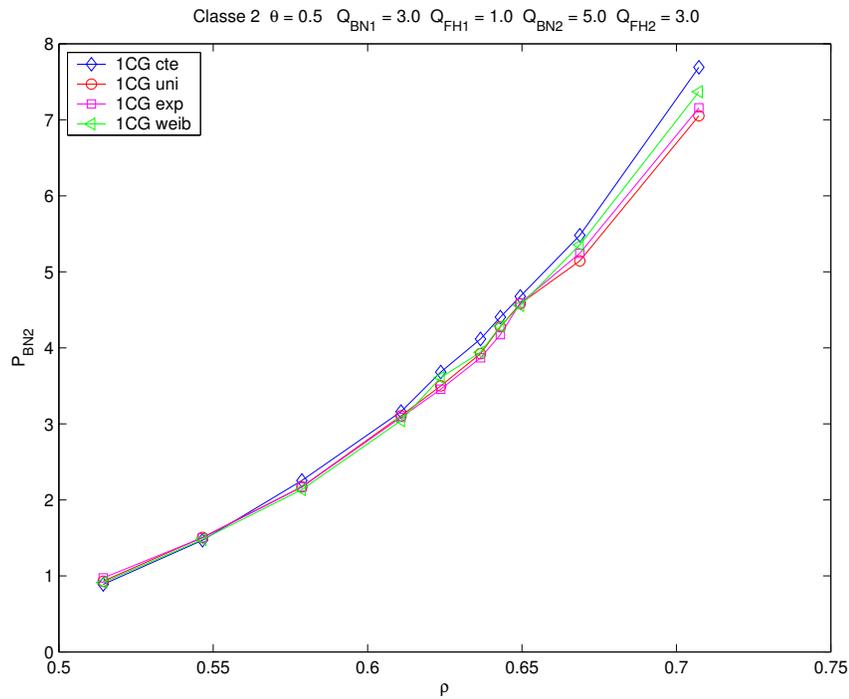


Figura 9.46: Percentual de Bloqueio de chamadas novas da classe 2 do algoritmo de 1CG, taxa de mobilidade $\theta = 0.5$.

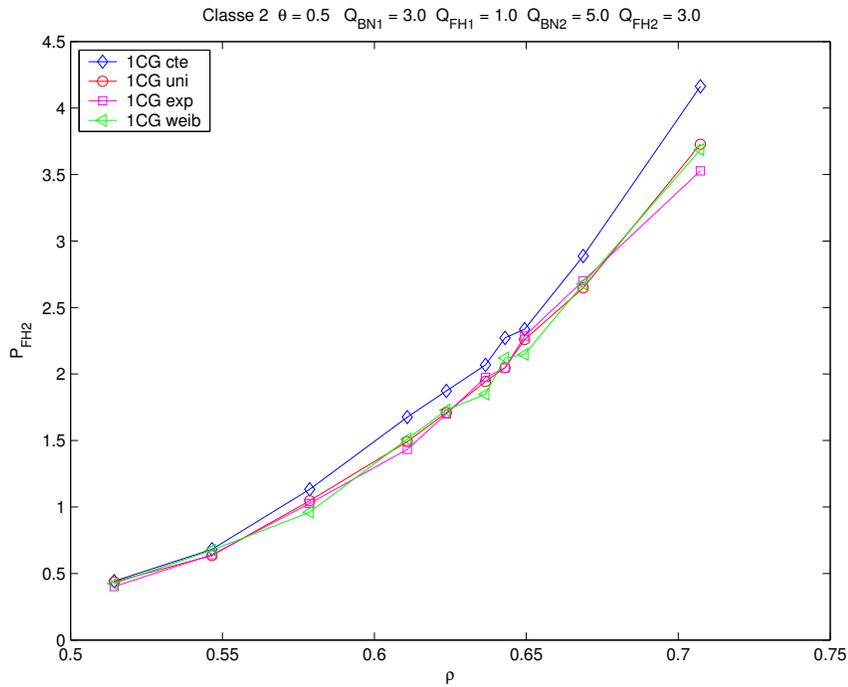


Figura 9.47: Percentual de Falha de chamadas em *handoff* da classe 2 do algoritmo de 1CG, taxa de mobilidade $\theta = 0.5$.

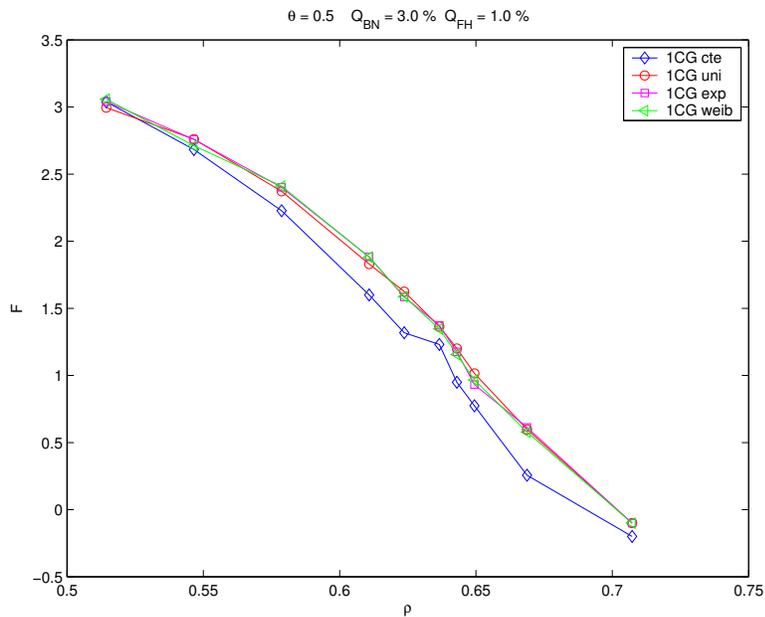


Figura 9.48: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 0.5$, requisito $Q\%(3, 1, 5, 3)$, algoritmo fixo 1CG.

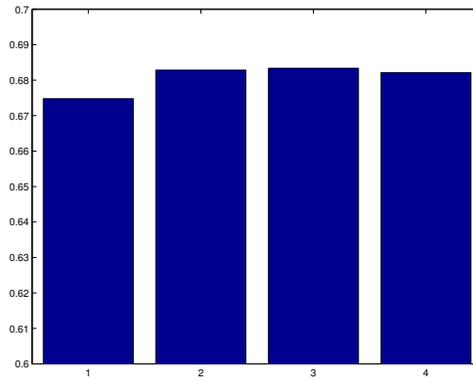


Figura 9.49: Parâmetro I para usuários da classe 1, considerando que o tempo de permanência do usuário na célula é dado pelas Distribuições (1)Constante, (2)Uniforme, (3)Exponencial, (4)Weibull, taxa de mobilidade $\theta = 0.5$, requisito $Q\%(3, 1, 5, 3)$, algoritmo fixo 1CG.

PUC-Rio - Certificação Digital Nº 0024874/CA

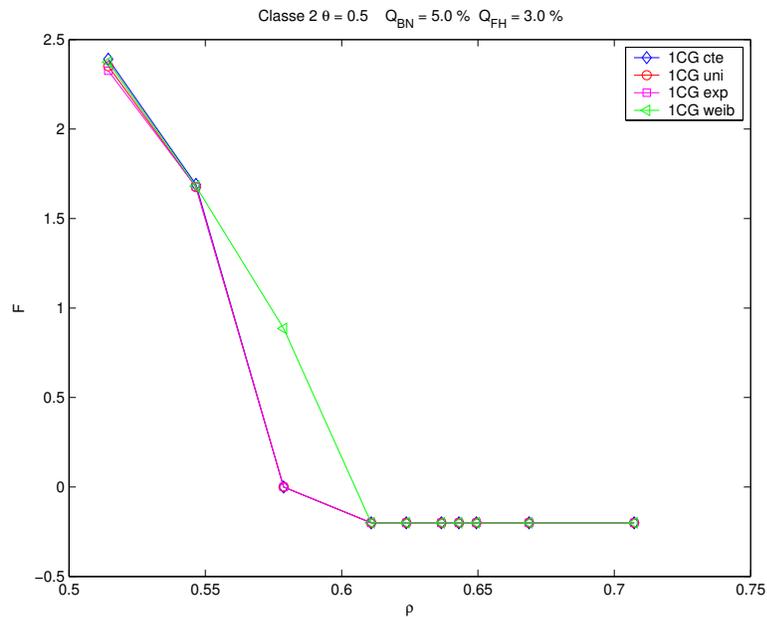


Figura 9.50: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 0.5$, requisito $Q\%(3, 1, 5, 3)$, algoritmo fixo 1CG.

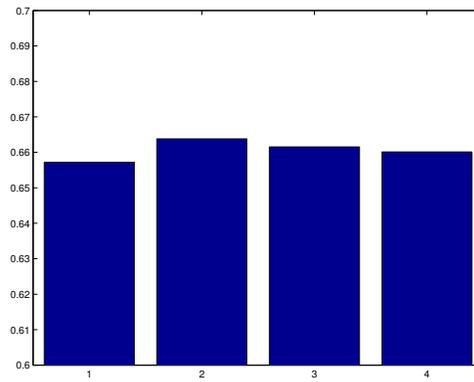


Figura 9.51: Parâmetro I para usuários da classe 2, considerando que o tempo de permanência do usuário na célula é dado pelas Distribuições (1)Constante, (2)Uniforme, (3)Exponencial, (4)Weibull, taxa de mobilidade $\theta = 0.5$, requisito $Q\%(3, 1, 5, 3)$, algoritmo fixo 1CG.

9.5.3

Comparação do Desempenho dos Algoritmos Propostos Considerando Distribuição Exponencial para o Tempo de Permanência do Usuário na Célula

Caso Taxa de mobilidade $\theta = 0.5$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, considerando distribuição exponencial para o tempo de permanência

As figuras 9.52, 9.53, 9.54 e 9.55 que apresentam os parâmetros F e I das classes 1 e 2 do caso $\theta = 0.5$, $Q\% = (3, 1, 5, 3)$, considerando distribuição exponencial para o tempo de permanência do usuário na célula.

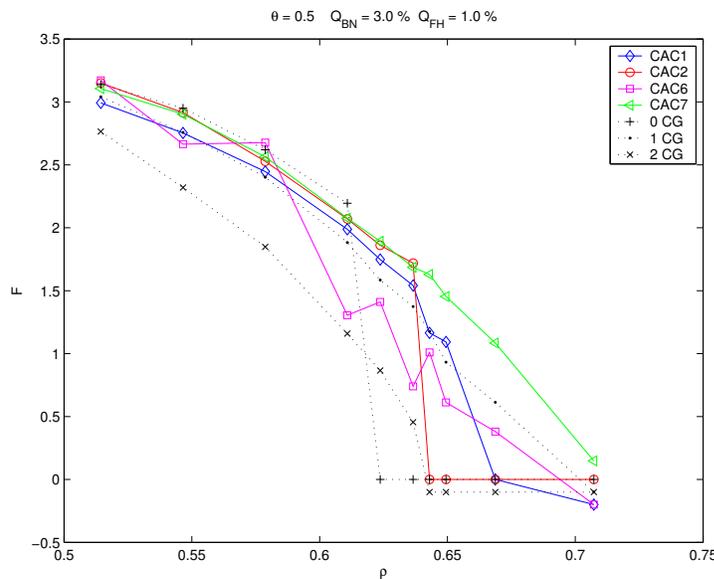


Figura 9.52: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 0.5$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

Observa-se na figura 9.52 que o algoritmo CAC7M apresenta um excelente comportamento para a classe 1, apresentando elevado valor de parâmetro F1. Além disso, nota-se que CAC7M não chega a ultrapassar os valores de Q da classe 1 para os valores de ρ que constam da figura, por isso, o parâmetro I1 não foi traçado na figura 9.53. No entanto, o algoritmo CAC6M apresenta um comportamento não-estável no caso de duas classes, suas curvas tendem a oscilar. A presença de um número maior de restrições $2M = 4$ faz com que as condições para incremento do número de recursos reservados para cada tipo de usuários não são tão coerentes como era no caso de uma classe, $M = 1$ quando só havia duas restrições de QoS.

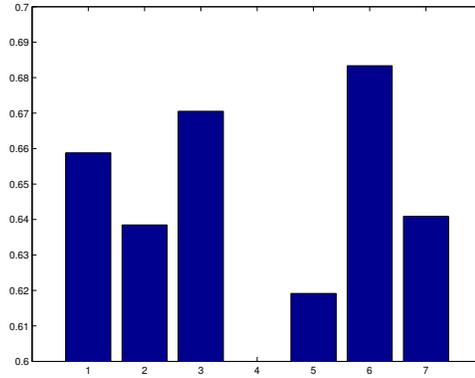


Figura 9.53: Parâmetro I para usuários da classe 1, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 0.5$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

No caso da classe 2, observa-se na figura 9.54 que os quatro algoritmos propostos apresentam percentuais de bloqueio semelhantes.

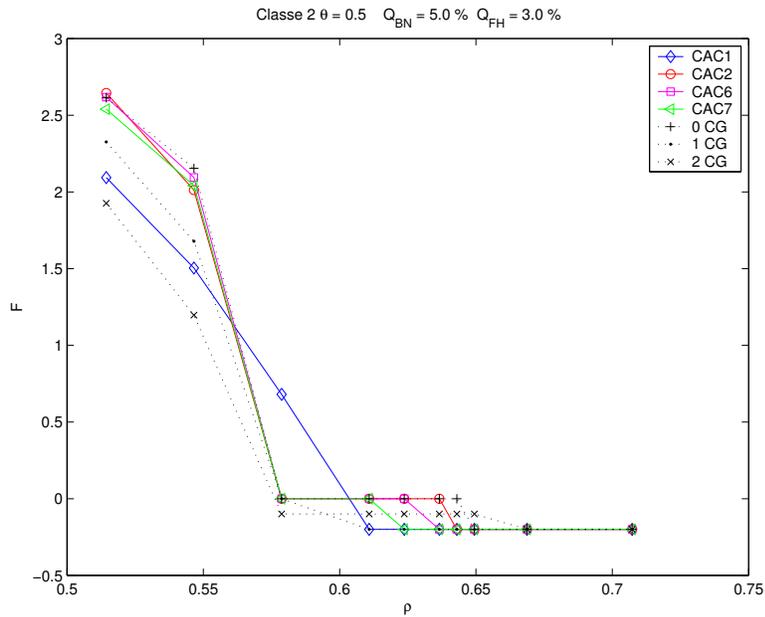


Figura 9.54: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 0.5$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

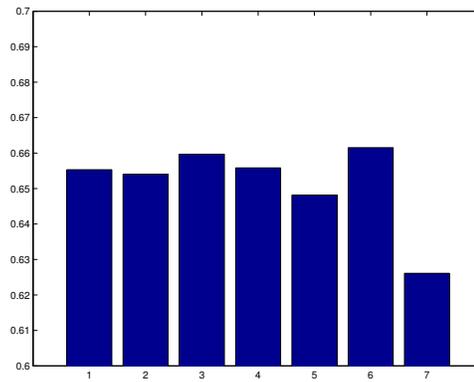


Figura 9.55: Parâmetro I para usuários da classe 2, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 0.5$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

Caso Taxa de mobilidade $\theta = 1$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

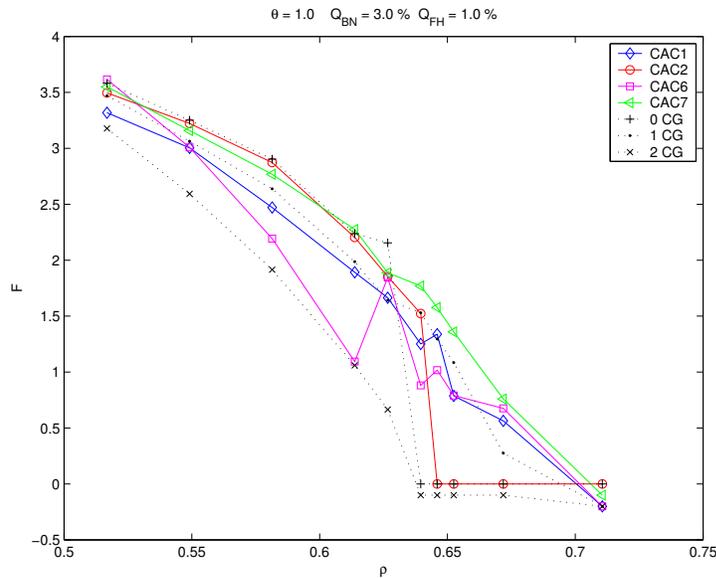


Figura 9.56: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

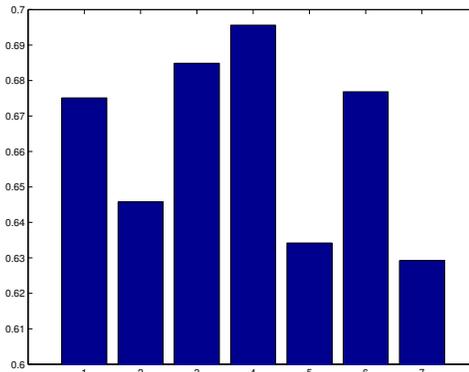


Figura 9.57: Parâmetro I para usuários da classe 1, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

Aumentando a taxa de mobilidade para $\theta = 1$, o algoritmo CAC7 continua apresentando o melhor desempenho para a classe 1 e desempenho equivalente aos demais para a classe 2, conforme as figuras 9.56 e 9.58.

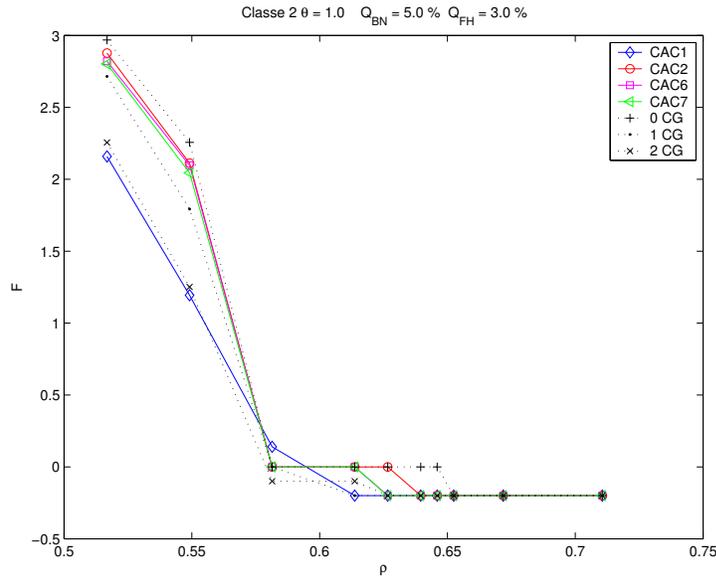


Figura 9.58: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

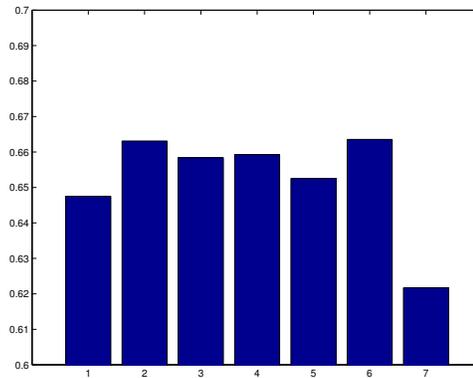


Figura 9.59: Parâmetro I para usuários da classe 2, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

Caso Taxa de mobilidade $\theta = 2$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

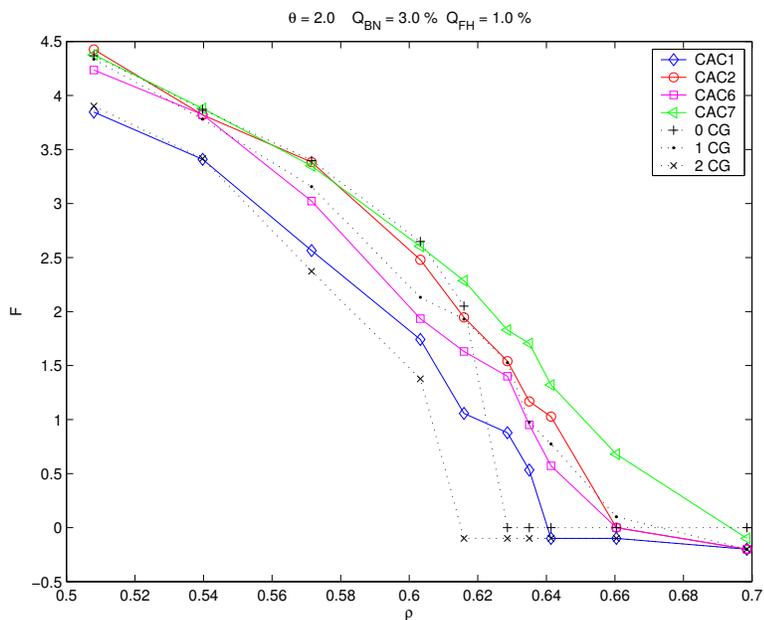


Figura 9.60: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 2$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

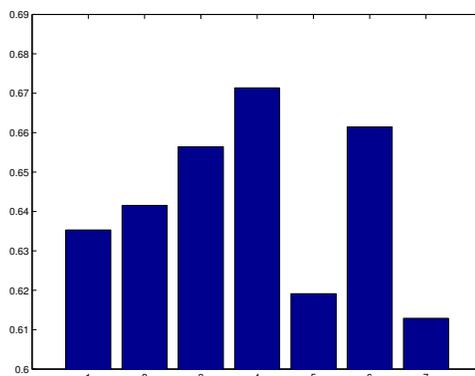


Figura 9.61: Parâmetro I para usuários da classe 1, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 2$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

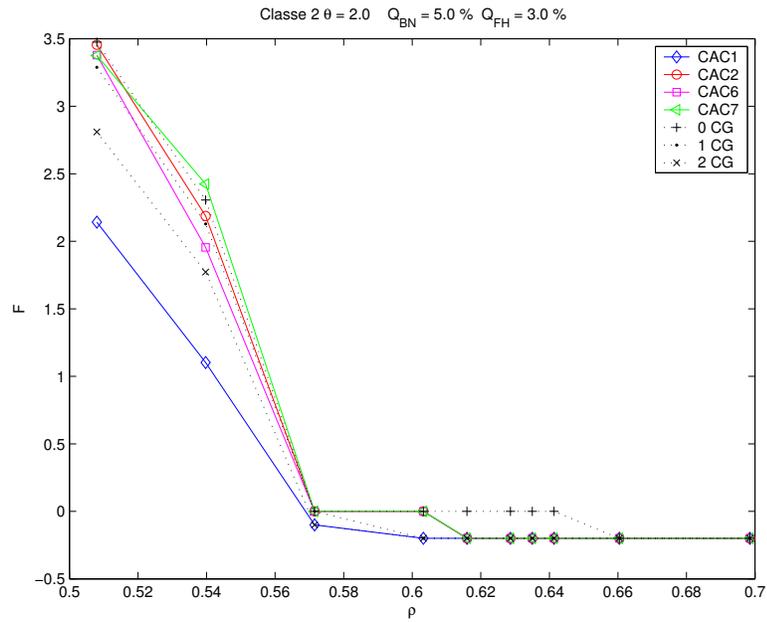


Figura 9.62: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 2$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

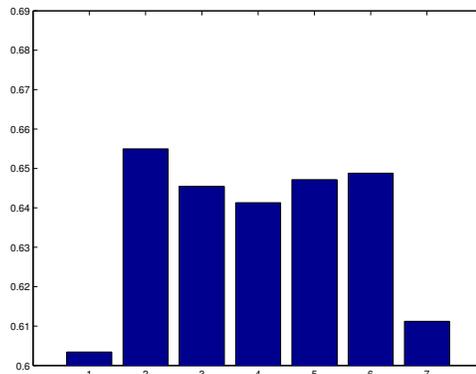


Figura 9.63: Parâmetro I para usuários da classe 2, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 2$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

Caso Taxa de mobilidade $\theta = 6$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

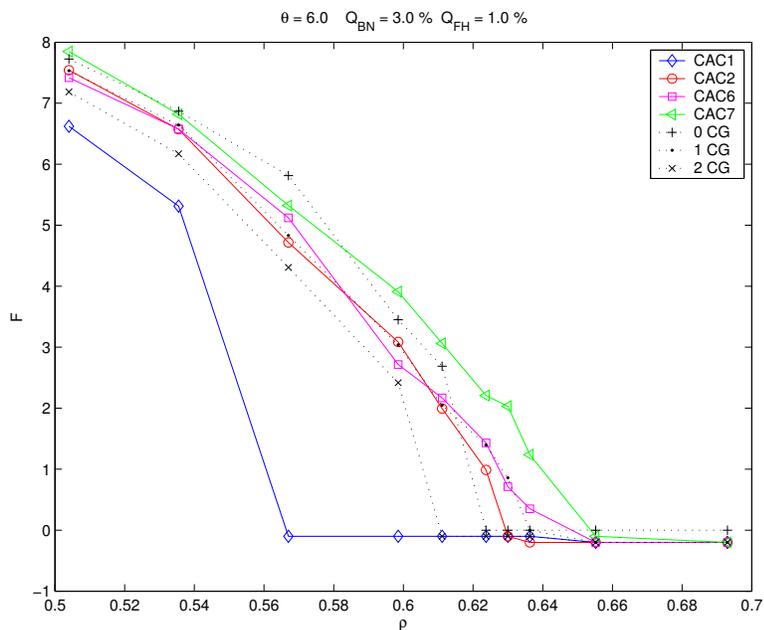


Figura 9.64: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

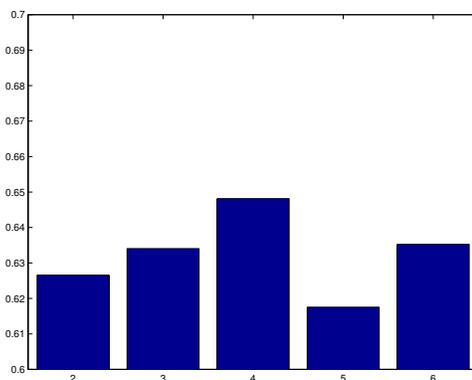


Figura 9.65: Parâmetro I para usuários da classe 1, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

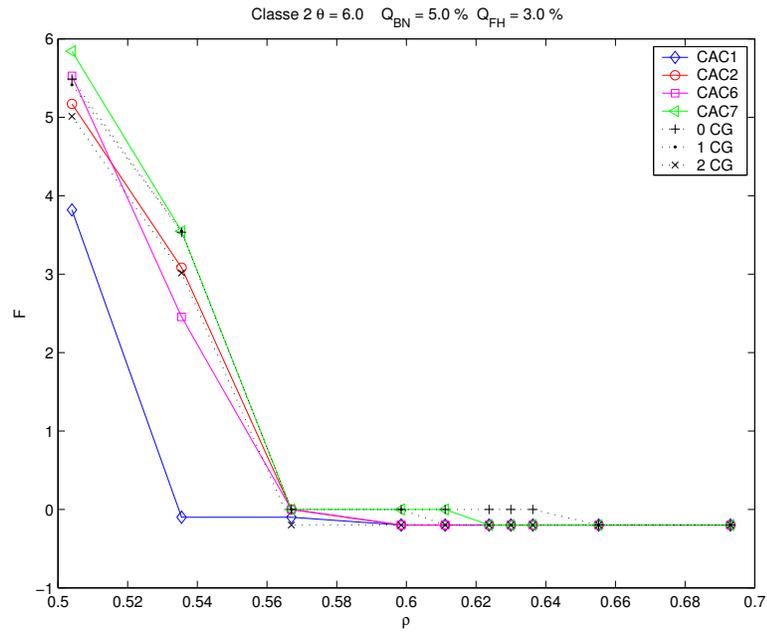


Figura 9.66: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

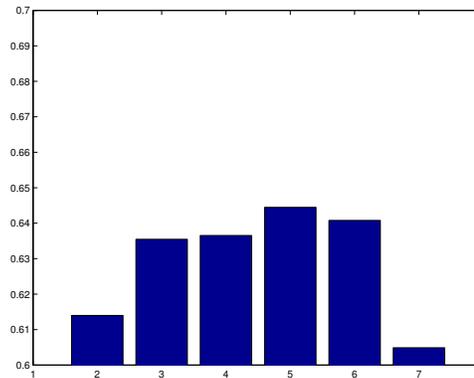


Figura 9.67: Parâmetro I para usuários da classe 2, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

Caso Taxa de mobilidade $\theta = 6$ e requisito de QoS $Q\%(4, 0.4, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

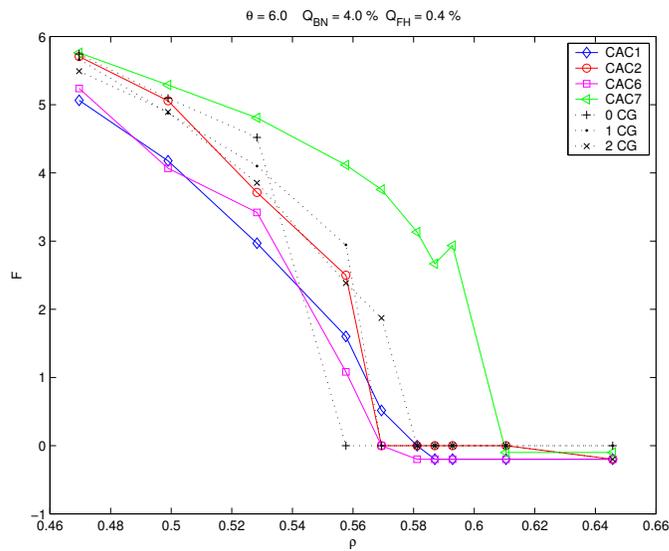


Figura 9.68: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 6$, requisito $Q\%(4, 0.4, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

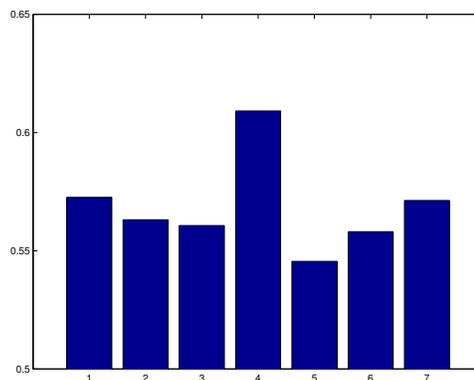


Figura 9.69: Parâmetro I para usuários da classe 1, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 6$, requisito $Q\%(4, 0.4, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

Nos gráficos correspondentes ao caso de distribuição exponencial para o tempo de permanência do usuário na célula, observa-se que o algoritmo CAC7M tende a apresentar um bom desempenho, principalmente para a classe 1. Tal desempenho pode ser especialmente observado no caso em que os requisitos de QoS são $Q\% = (4; 0.4; 5; 3)$ nas figuras 9.68, 9.69, 9.70

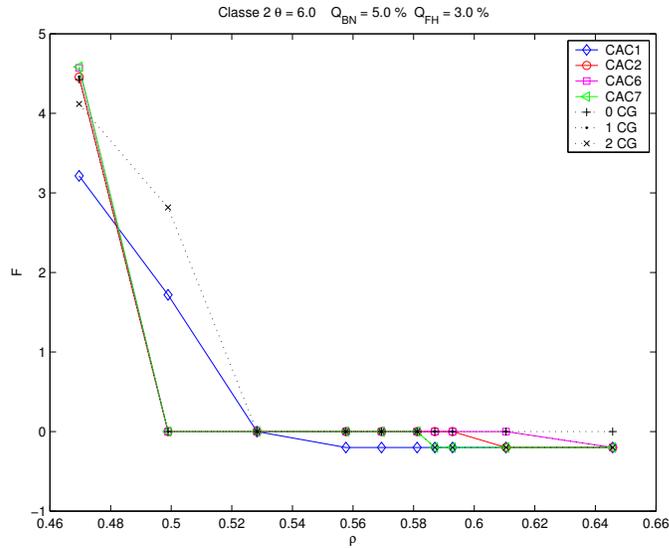


Figura 9.70: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 6$, requisito $Q\%(4, 0.4, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

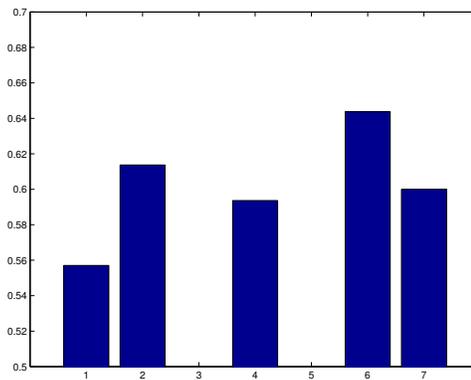


Figura 9.71: Parâmetro I para usuários da classe 1, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 6$, requisito $Q\%(4, 0.4, 5, 3)$, considerando distribuição exponencial para o tempo de permanência.

e 9.71. Os algoritmos CAC1M e CAC2M apresentam bons desempenhos, porém isto não ocorre em todos os cenários. O algoritmo CAC6M, diante do fato de só buscar alterar o número de recursos em faixa restritas, apresenta um comportamento que oscila na região de intensidade de tráfego próxima ao valor ρ_Q .

9.5.4

Comparação do Desempenho dos Algoritmos Propostos Considerando Distribuição Constante para o Tempo de Permanência do Usuário na Célula

Caso Taxa de mobilidade $\theta = 1$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, considerando distribuição constante para o tempo de permanência.

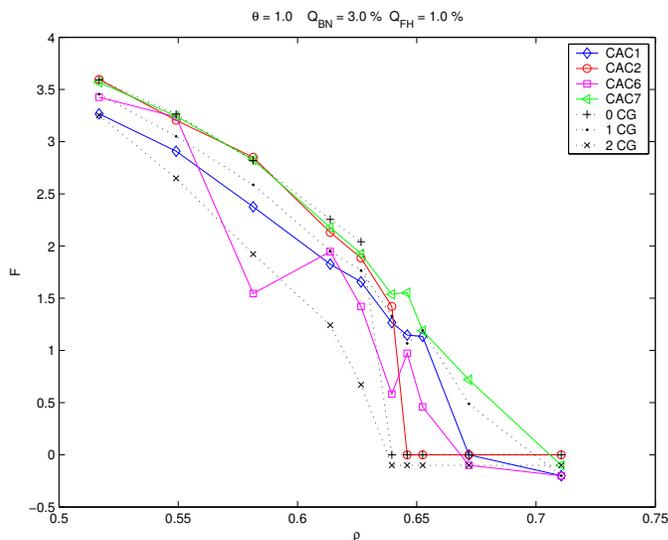


Figura 9.72: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição constante para o tempo de permanência.

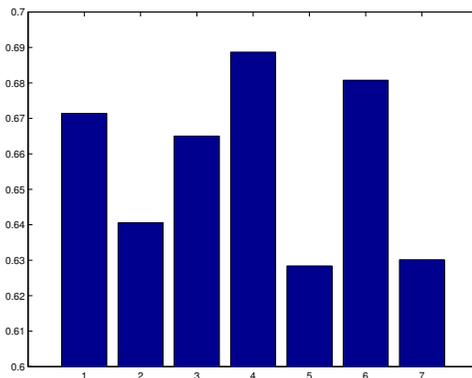


Figura 9.73: Parâmetro I para usuários da classe 1, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição constante para o tempo de permanência.

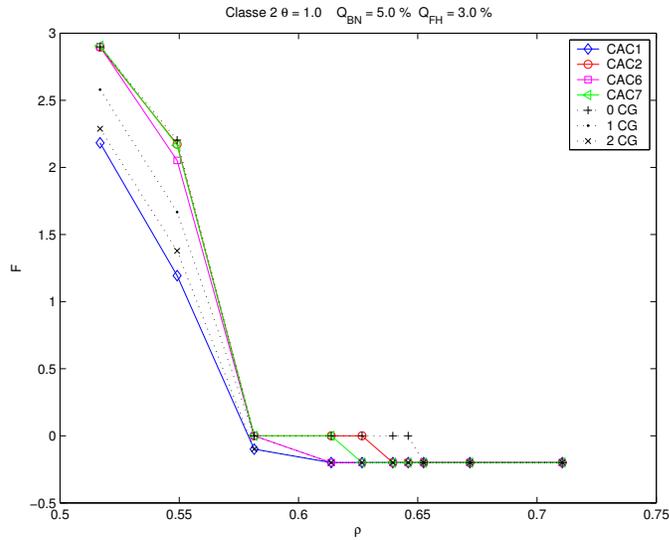


Figura 9.74: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição constante para o tempo de permanência.

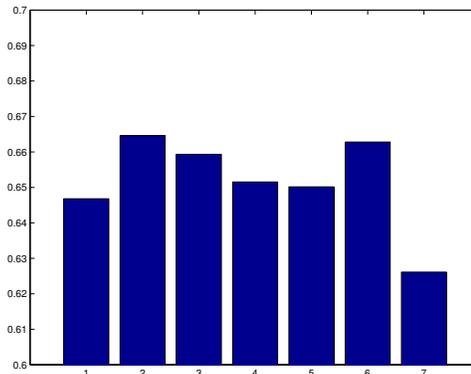


Figura 9.75: Parâmetro I para usuários da classe 2, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição constante para o tempo de permanência.

Neste caso, em que o tempo de permanência do usuário na célula é modelado pela distribuição constante, os dois algoritmos que apresentam melhores desempenhos são CAC7 e CAC2.

Caso Taxa de mobilidade $\theta = 6$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, considerando distribuição constante para o tempo de permanência.

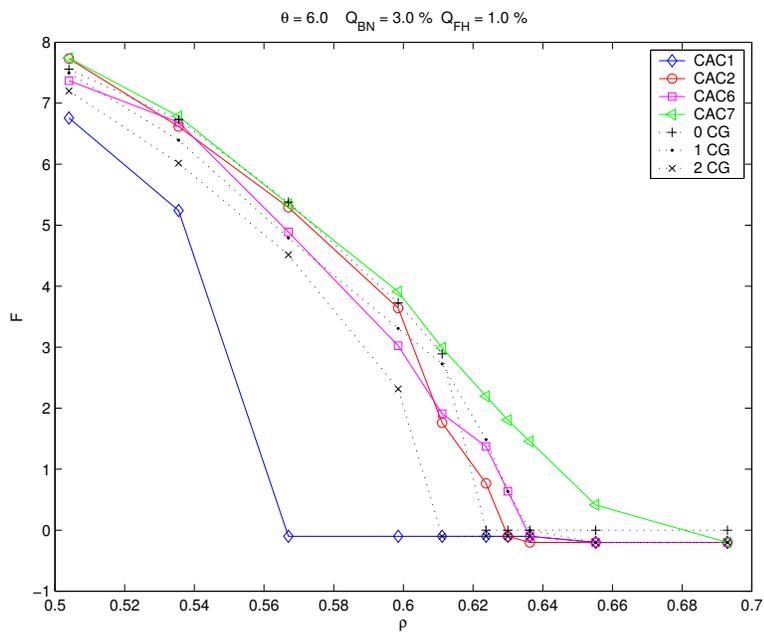


Figura 9.76: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição constante para o tempo de permanência.

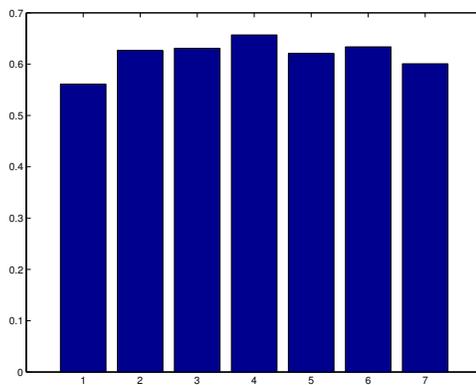


Figura 9.77: Parâmetro I para usuários da classe 1, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição constante para o tempo de permanência.

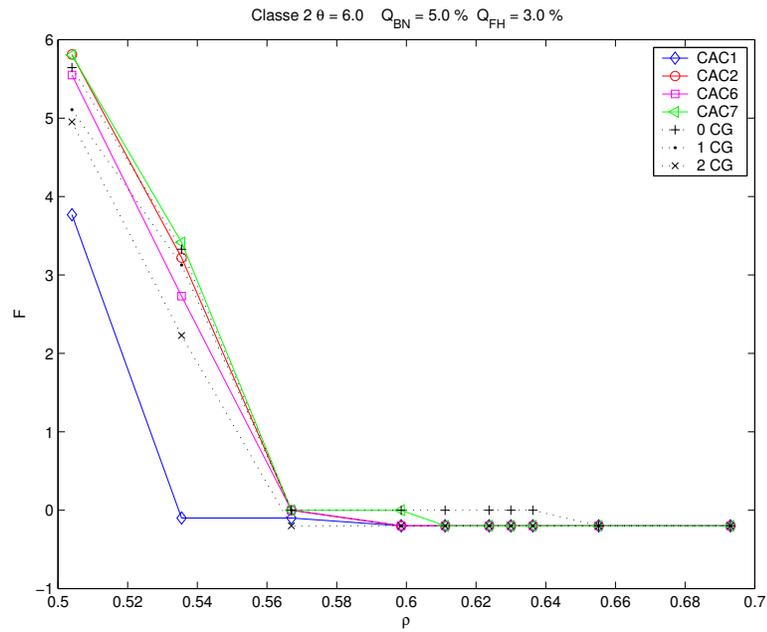


Figura 9.78: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição constante para o tempo de permanência.

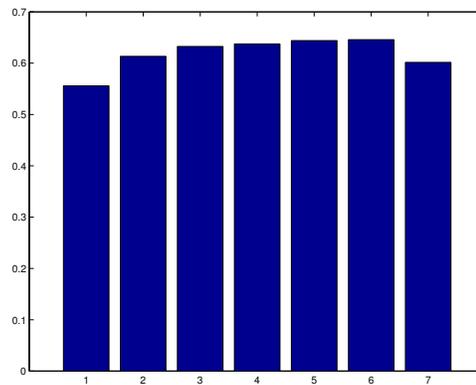


Figura 9.79: Parâmetro I para usuários da classe 2, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição constante para o tempo de permanência.

9.5.5

Comparação do Desempenho dos Algoritmos Propostos Considerando Distribuição Uniforme para o Tempo de Permanência do Usuário na Célula

Caso Taxa de mobilidade $\theta = 1$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, considerando distribuição uniforme para o tempo de permanência.

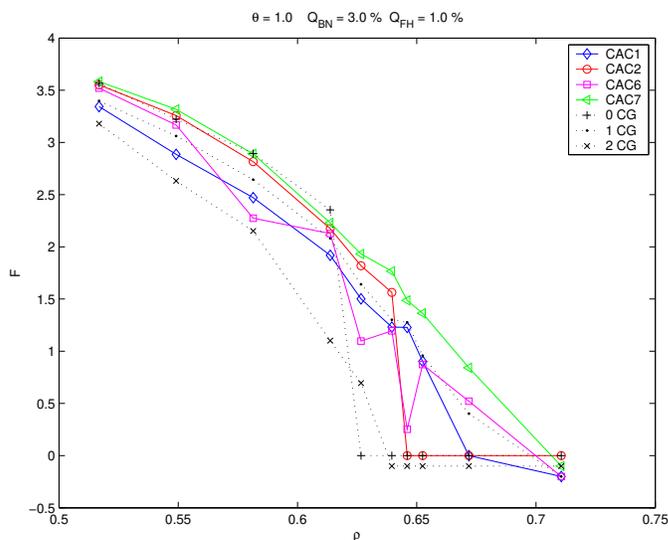


Figura 9.80: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição uniforme para o tempo de permanência.

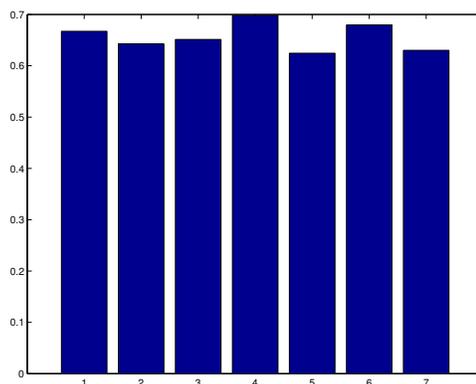


Figura 9.81: Parâmetro I para usuários da classe 1, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição uniforme para o tempo de permanência.

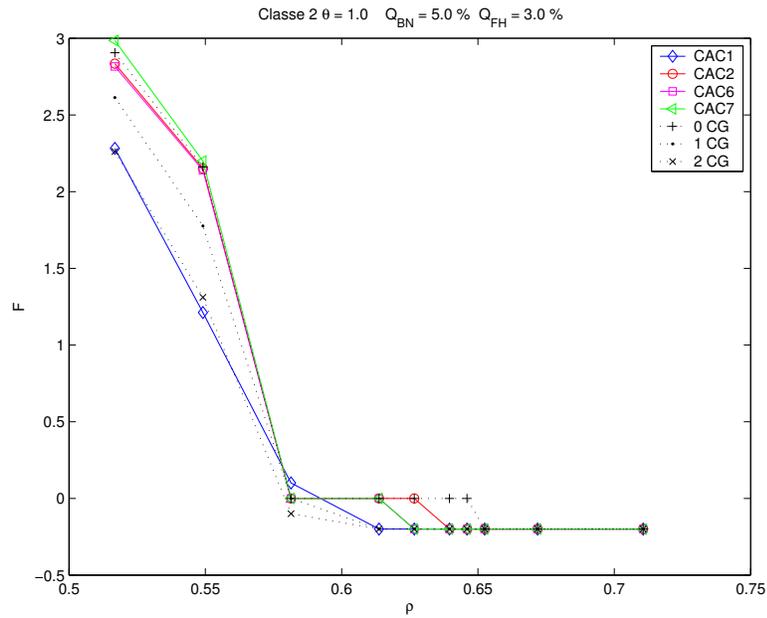


Figura 9.82: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição uniforme para o tempo de permanência.

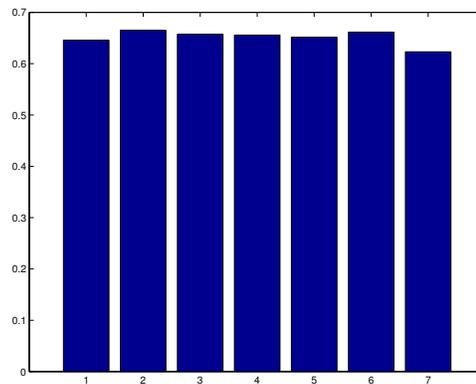


Figura 9.83: Parâmetro I para usuários da classe 2, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição uniforme para o tempo de permanência.

Caso Taxa de mobilidade $\theta = 2$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, considerando distribuição uniforme para o tempo de permanência.

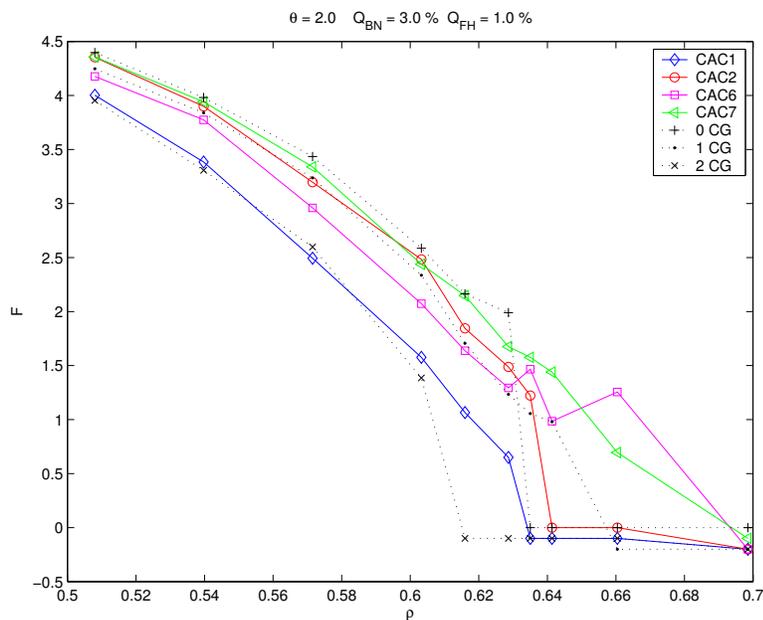


Figura 9.84: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 2$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição uniforme para o tempo de permanência.

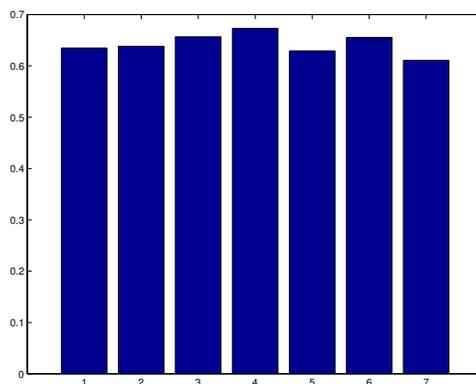


Figura 9.85: Parâmetro I para usuários da classe 1, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 2$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição uniforme para o tempo de permanência.

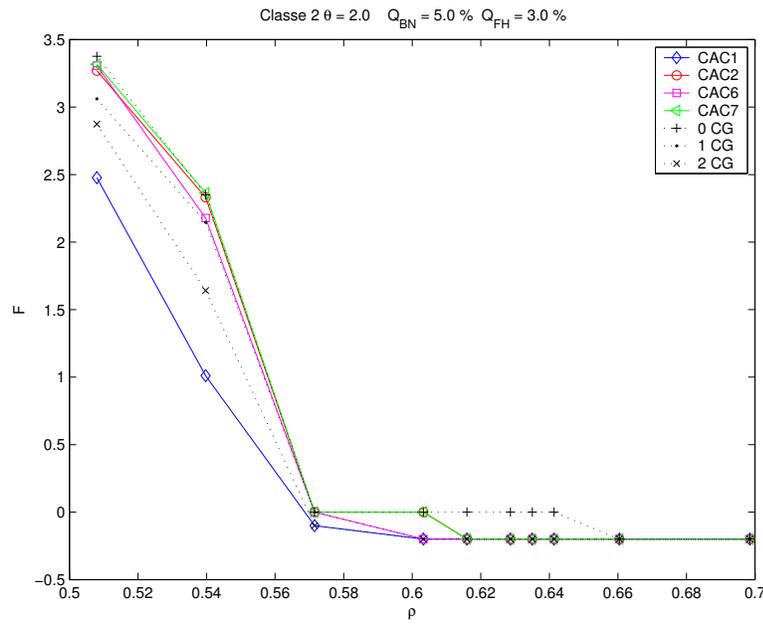


Figura 9.86: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 2$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição uniforme para o tempo de permanência.

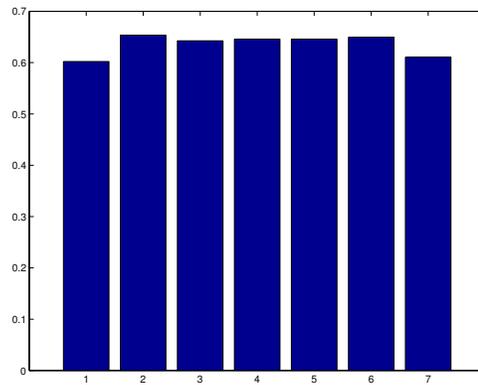


Figura 9.87: Parâmetro I para usuários da classe 2, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 2$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição uniforme para o tempo de permanência.

Caso Taxa de mobilidade $\theta = 6$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, considerando distribuição uniforme para o tempo de permanência.

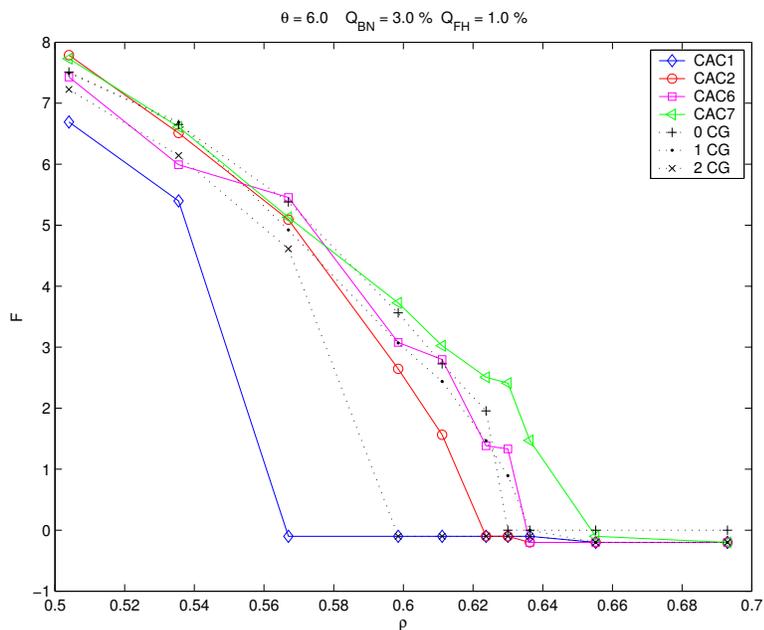


Figura 9.88: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição uniforme para o tempo de permanência.

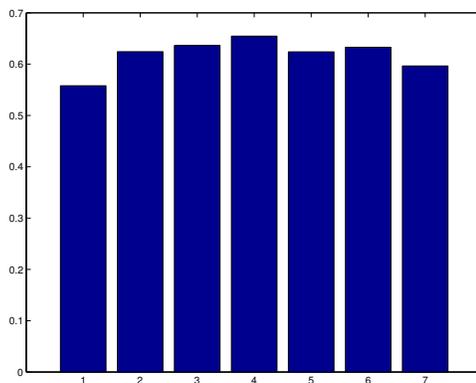


Figura 9.89: Parâmetro I para usuários da classe 1, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição uniforme para o tempo de permanência.

Se o tempo de permanência do usuário na célula é modelado por uma distribuição uniforme, observa-se nas figuras 9.80, 9.81, 9.82 e 9.83 do caso

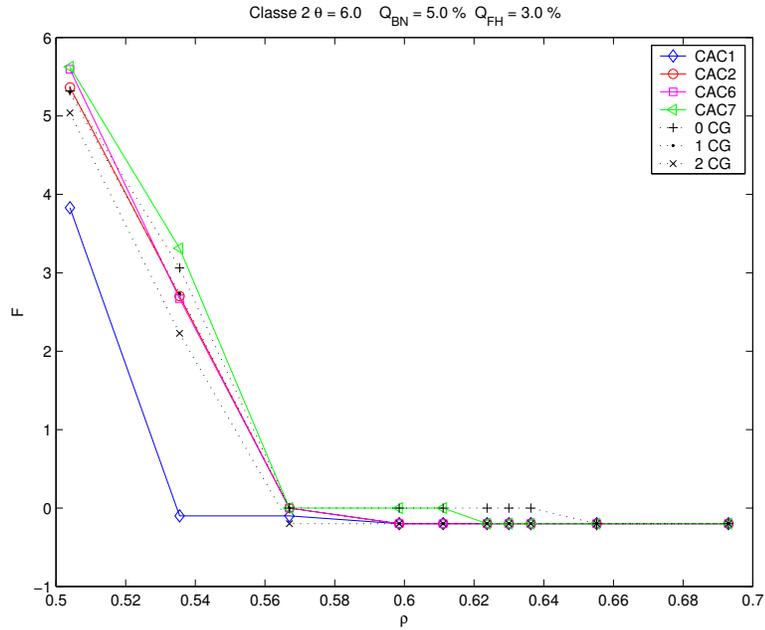


Figura 9.90: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição uniforme para o tempo de permanência.

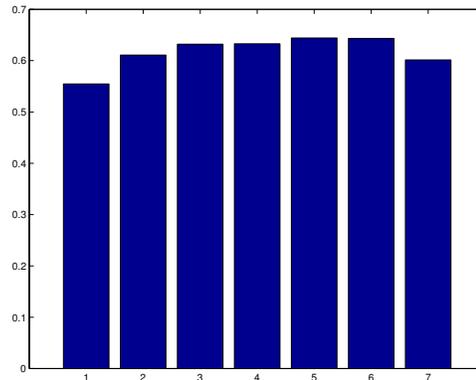


Figura 9.91: Parâmetro I para usuários da classe 2, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição uniforme para o tempo de permanência.

$\theta = 1$; nas figuras 9.84, 9.85, 9.86 e 9.87 do caso $\theta = 2$ e, finalmente nas figuras 9.88, 9.89, 9.90 e 9.91 do caso $\theta = 6$; que o algoritmo CAC7M apresenta o melhor desempenho. Tal resultado é interessante pois o seu projeto original utiliza a distribuição exponencial para a modelagem do tempo de permanência ao invés da distribuição uniforme como no cenário testado.

9.5.6

Comparação do Desempenho dos Algoritmos Propostos Considerando Distribuição Weibull para o Tempo de Permanência do Usuário na Célula

Caso Taxa de mobilidade $\theta = 1$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, considerando distribuição Weibull para o tempo de permanência.

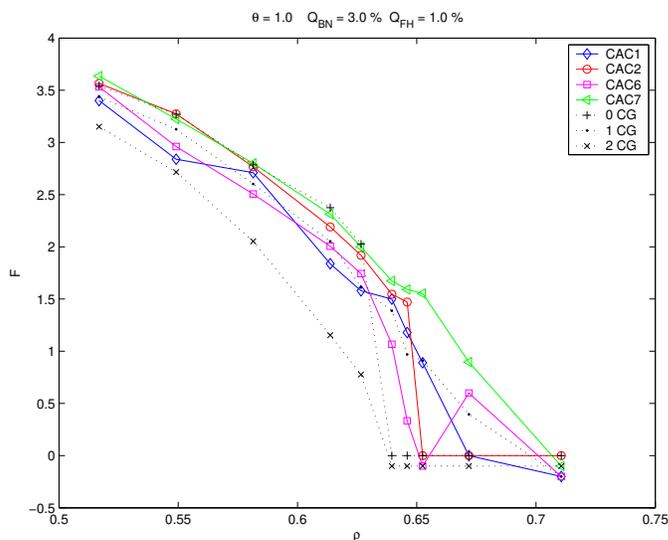


Figura 9.92: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição Weibull para o tempo de permanência.

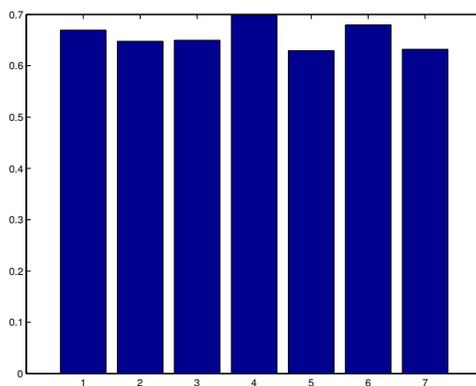


Figura 9.93: Parâmetro I para usuários da classe 1, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição Weibull para o tempo de permanência.

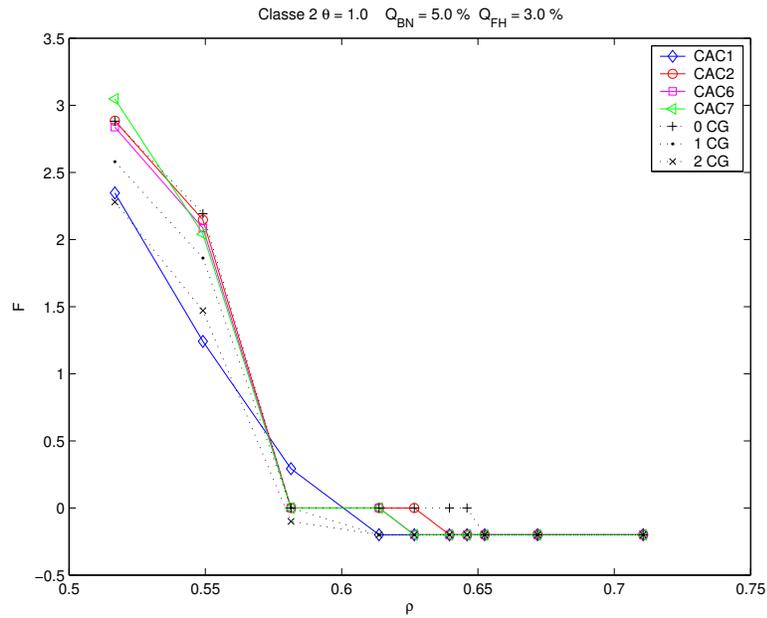


Figura 9.94: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição Weibull para o tempo de permanência.

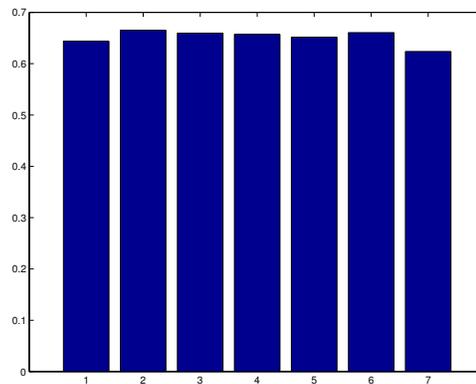


Figura 9.95: Parâmetro I para usuários da classe 2, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição Weibull para o tempo de permanência.

Caso Taxa de mobilidade $\theta = 2$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, considerando distribuição Weibull para o tempo de permanência.

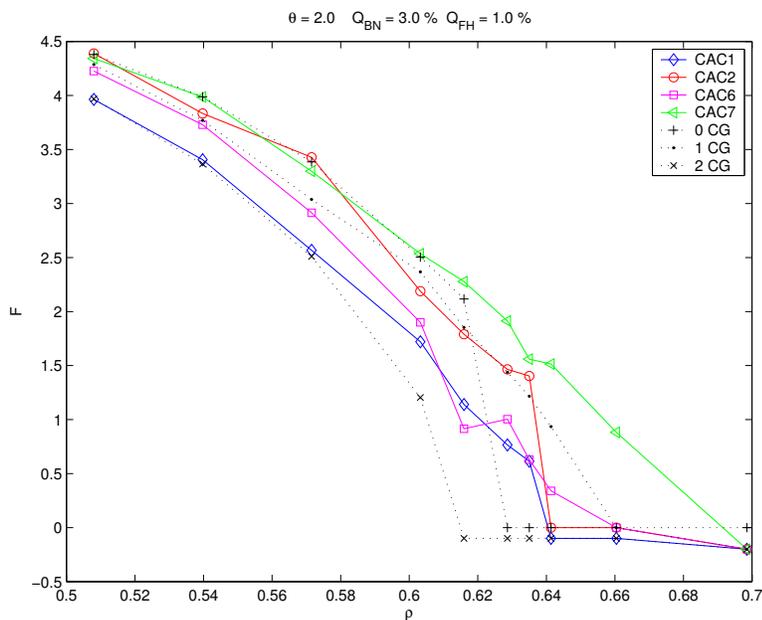


Figura 9.96: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 2$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição Weibull para o tempo de permanência.

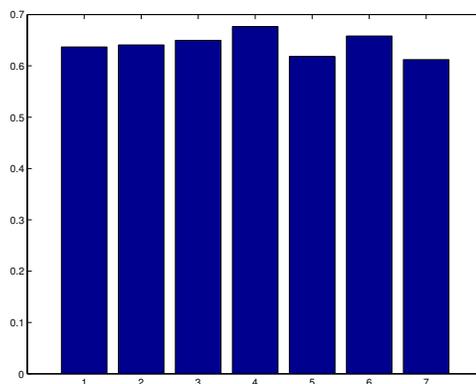


Figura 9.97: Parâmetro I para usuários da classe 1, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 2$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição Weibull para o tempo de permanência.

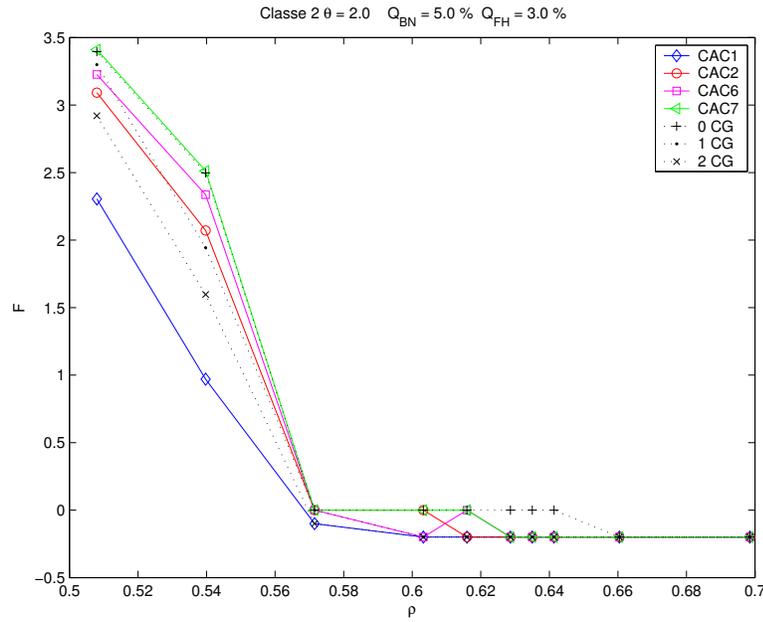


Figura 9.98: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 2$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição Weibull para o tempo de permanência.

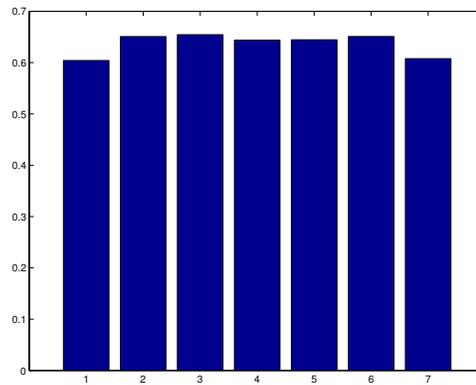


Figura 9.99: Parâmetro I para usuários da classe 2, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 2$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição Weibull para o tempo de permanência.

Caso Taxa de mobilidade $\theta = 6$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, considerando distribuição Weibull para o tempo de permanência.

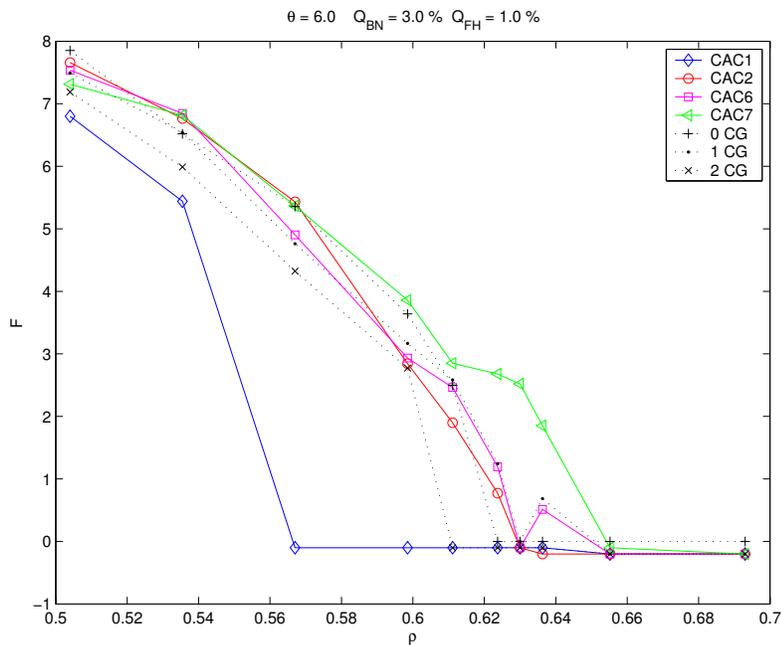


Figura 9.100: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição Weibull para o tempo de permanência.

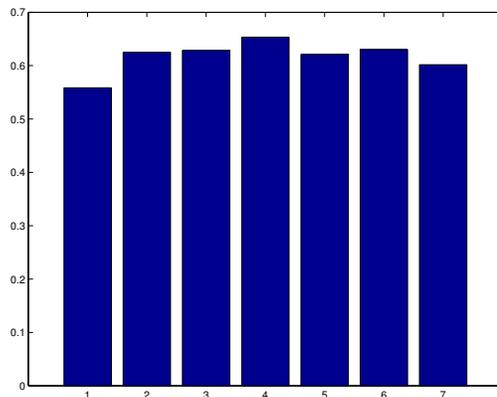


Figura 9.101: Parâmetro I para usuários da classe 1, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição Weibull para o tempo de permanência.

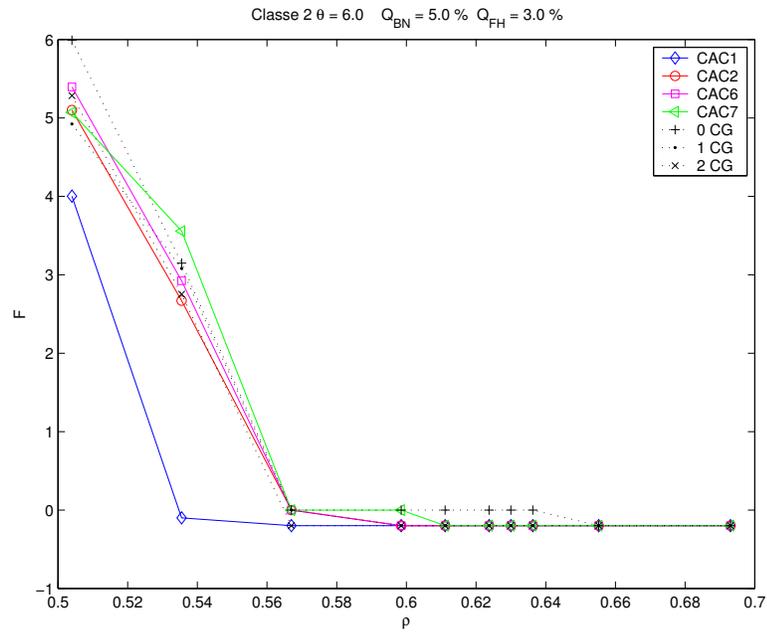


Figura 9.102: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição Weibull para o tempo de permanência.

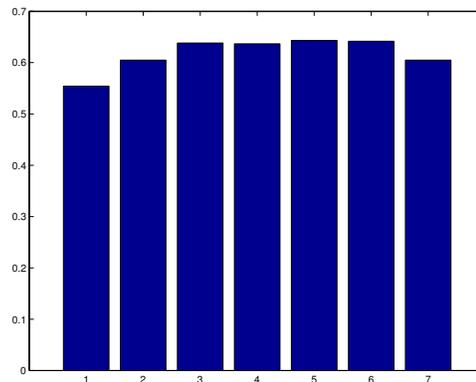


Figura 9.103: Parâmetro I para usuários da classe 2, para algoritmos (1)CAC1, (2)CAC2, (3)CAC6, (4)CAC7, (5)0CG, (6)1CG e (7)2CG, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, considerando distribuição Weibull para o tempo de permanência.

Observa-se que o desempenho do algoritmo CAC7M também é o melhor no caso em que o tempo de permanência é modelado pela distribuição Weibull.

9.5.7

Comportamento do algoritmo CAC1M com a Distribuição que Modela o Tempo de Permanência de usuário na Célula

Caso Taxa de mobilidade $\theta = 1$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, algoritmo CAC1M.

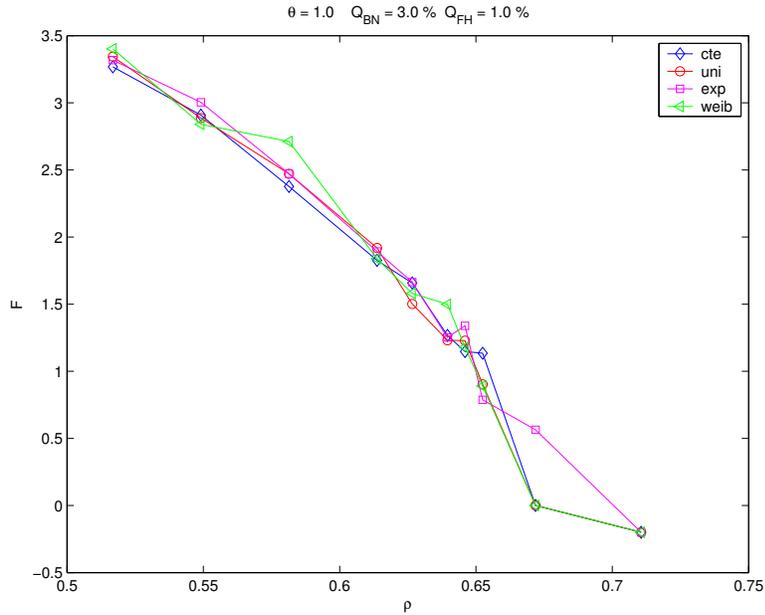


Figura 9.104: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC1M.

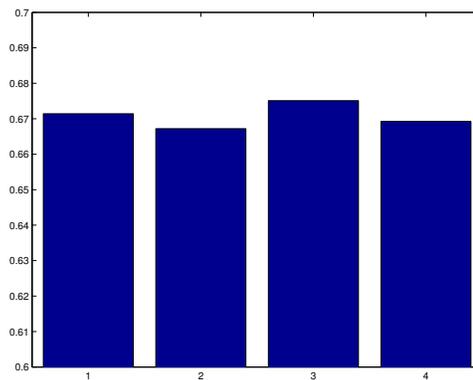


Figura 9.105: Parâmetro I para usuários da classe 1, considerando que o tempo de permanência do usuário na célula é dado pelas Distribuições (1)Constante, (2)Uniforme, (3)Exponencial, (4)Weibull, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC1M.

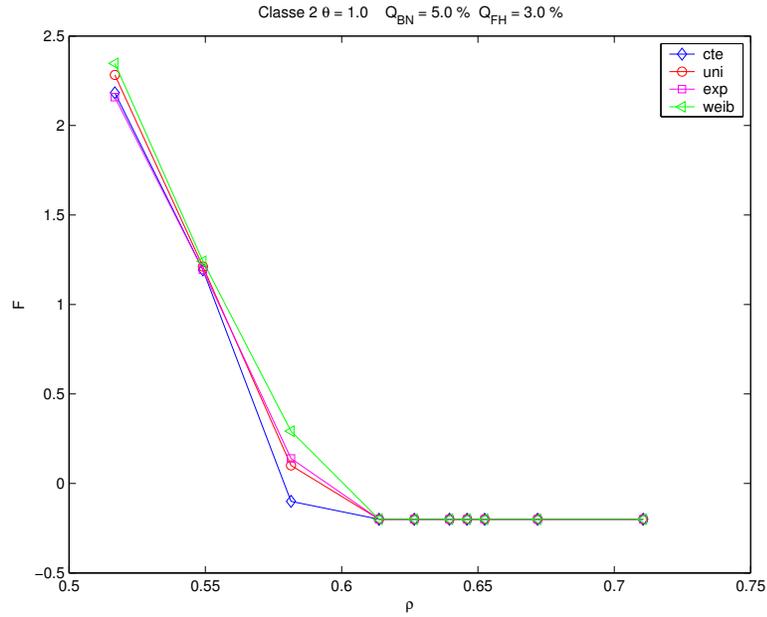


Figura 9.106: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC1M.

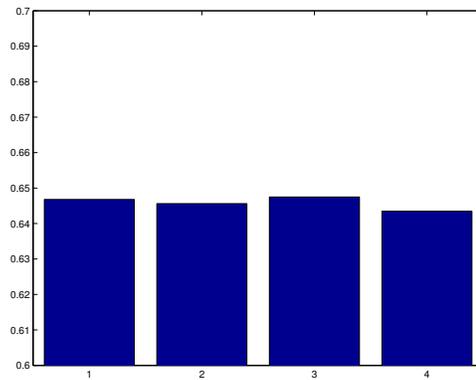


Figura 9.107: Parâmetro I para usuários da classe 2, considerando que o tempo de permanência do usuário na célula é dado pelas Distribuições (1)Constante, (2)Uniforme, (3)Exponencial, (4)Weibull, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC1M.

Caso Taxa de mobilidade $\theta = 6$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, algoritmo CAC1M.

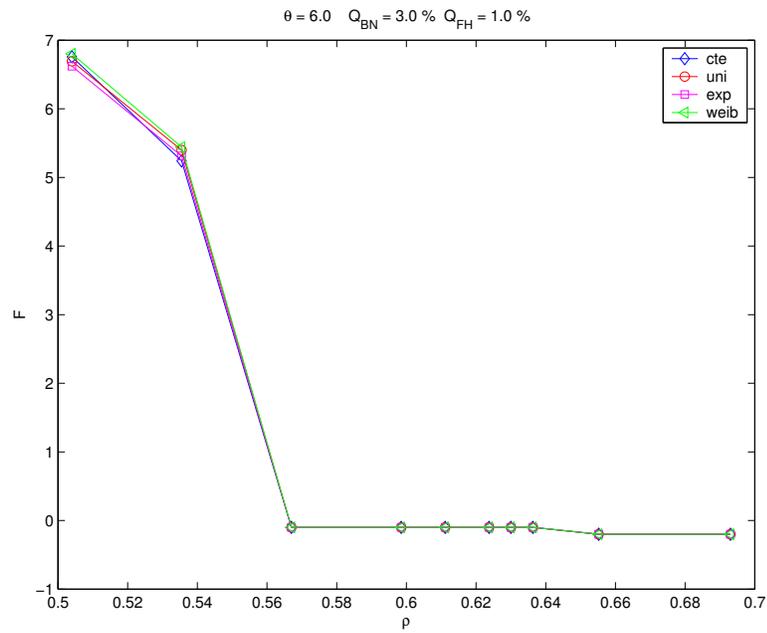


Figura 9.108: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC1M.

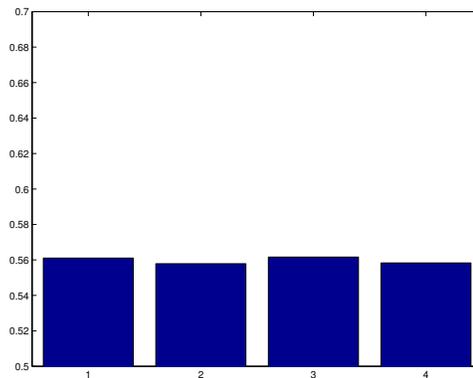


Figura 9.109: Parâmetro I para usuários da classe 1, considerando que o tempo de permanência do usuário na célula é dado pelas Distribuições (1)Constante, (2)Uniforme, (3)Exponencial, (4)Weibull, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC1M.

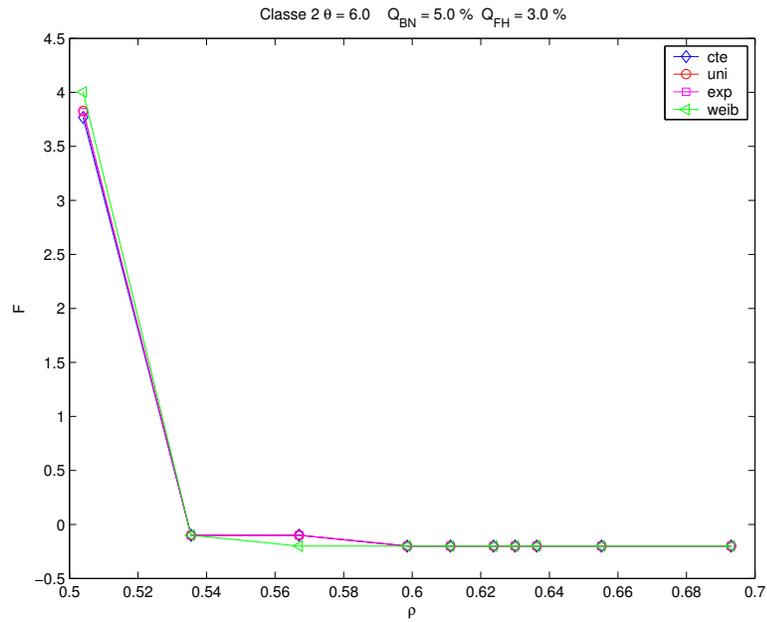


Figura 9.110: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC1M.

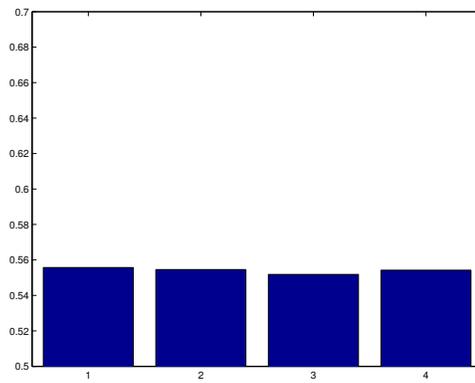


Figura 9.111: Parâmetro I para usuários da classe 2, considerando que o tempo de permanência do usuário na célula é dado pelas Distribuições (1)Constante, (2)Uniforme, (3)Exponencial, (4)Weibull, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC1M.

9.5.8

Comportamento do algoritmo CAC2M com a Distribuição que Modela o Tempo de Permanência de usuário na Célula

Caso Taxa de mobilidade $\theta = 1$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, algoritmo CAC2M.

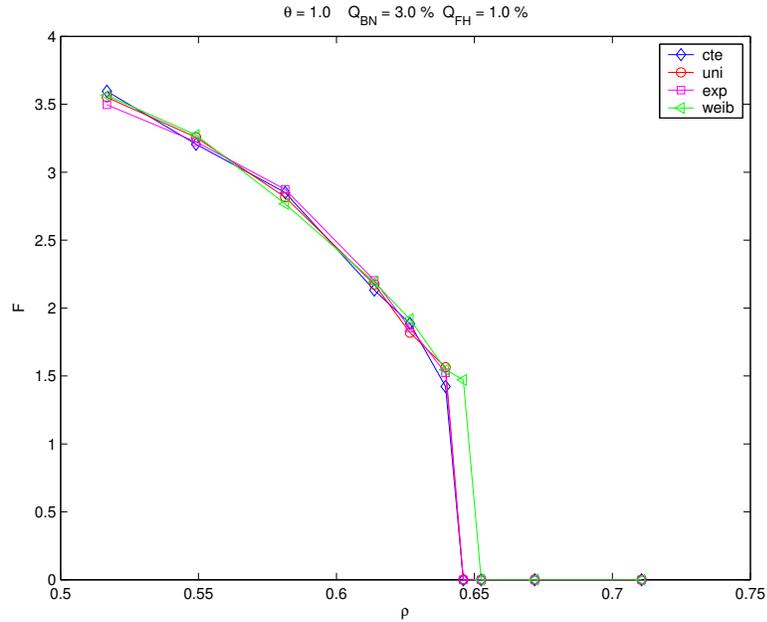


Figura 9.112: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC2M.

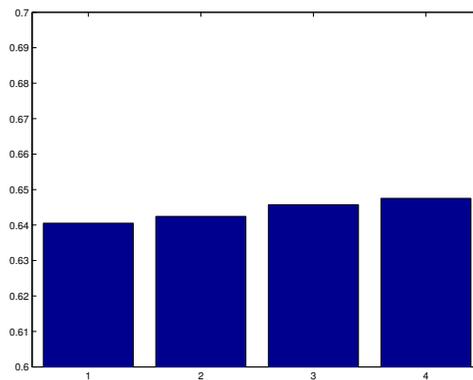


Figura 9.113: Parâmetro I para usuários da classe 1, considerando que o tempo de permanência do usuário na célula é dado pelas Distribuições (1)Constante, (2)Uniforme, (3)Exponencial, (4)Weibull, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC2M.

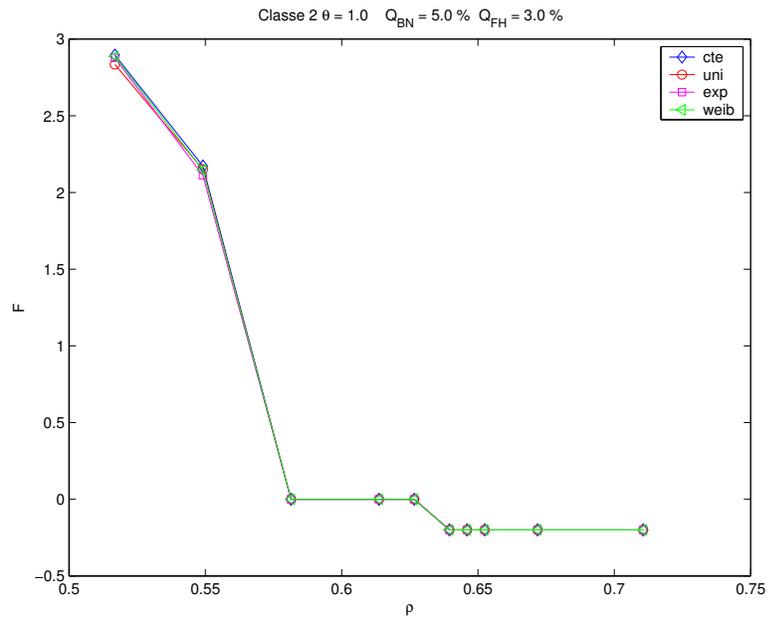


Figura 9.114: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC2M.

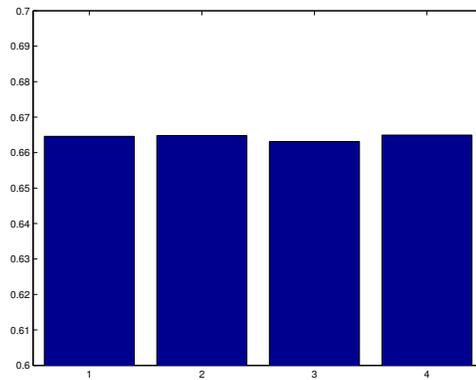


Figura 9.115: Parâmetro I para usuários da classe 2, considerando que o tempo de permanência do usuário na célula é dado pelas Distribuições (1)Constante, (2)Uniforme, (3)Exponencial, (4)Weibull, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC2M.

Caso Taxa de mobilidade $\theta = 6$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, algoritmo CAC2M.

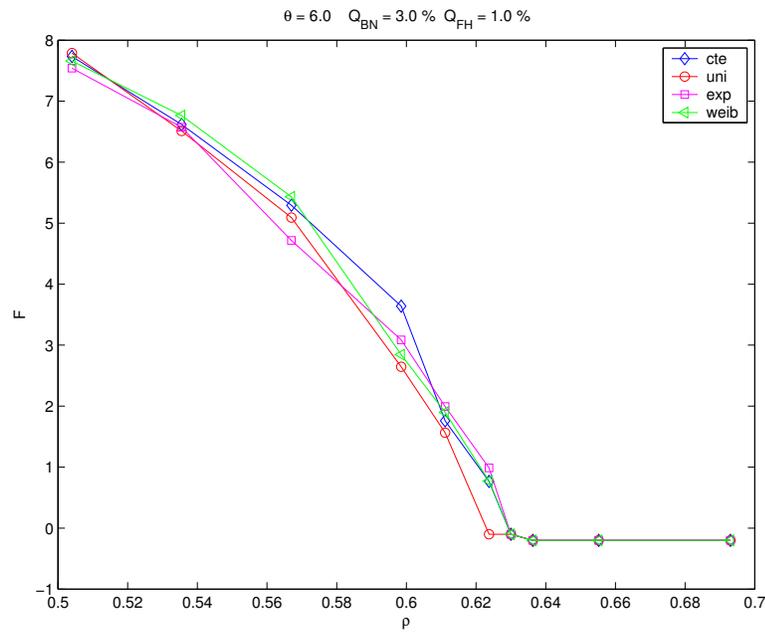


Figura 9.116: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC2M.

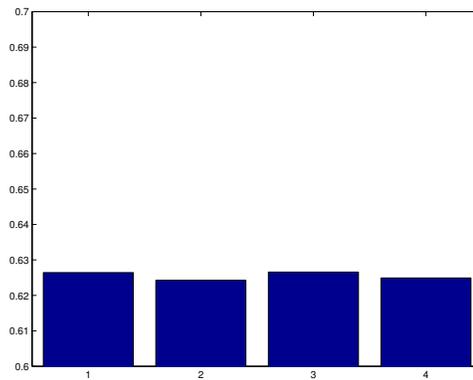


Figura 9.117: Parâmetro I para usuários da classe 1, considerando que o tempo de permanência do usuário na célula é dado pelas Distribuições (1)Constante, (2)Uniforme, (3)Exponencial, (4)Weibull, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC2M.

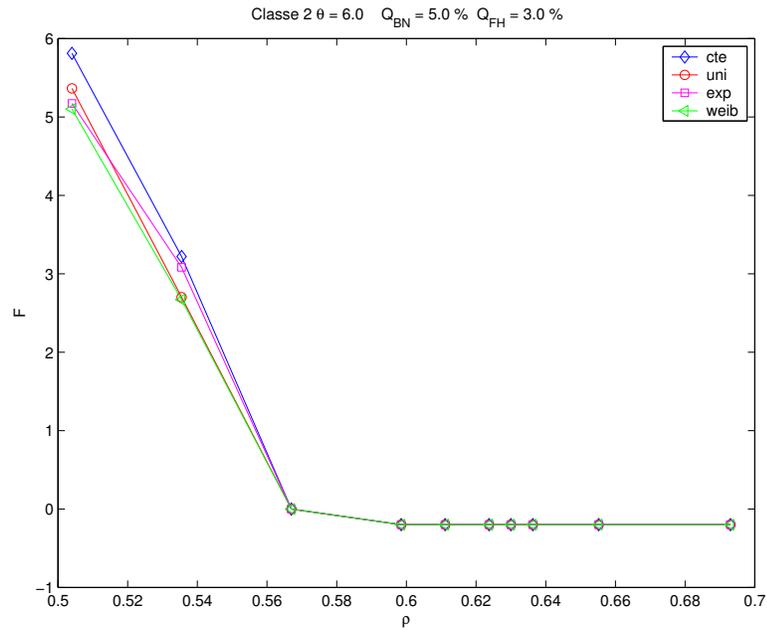


Figura 9.118: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC2M.

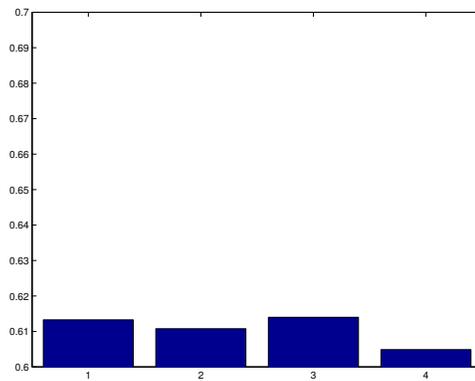


Figura 9.119: Parâmetro I para usuários da classe 2, considerando que o tempo de permanência do usuário na célula é dado pelas Distribuições (1)Constante, (2)Uniforme, (3)Exponencial, (4)Weibull, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC2M.

9.5.9

Comportamento do algoritmo CAC6M com a Distribuição que Modela o Tempo de Permanência de usuário na Célula

Caso Taxa de mobilidade $\theta = 1$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, algoritmo CAC6M.

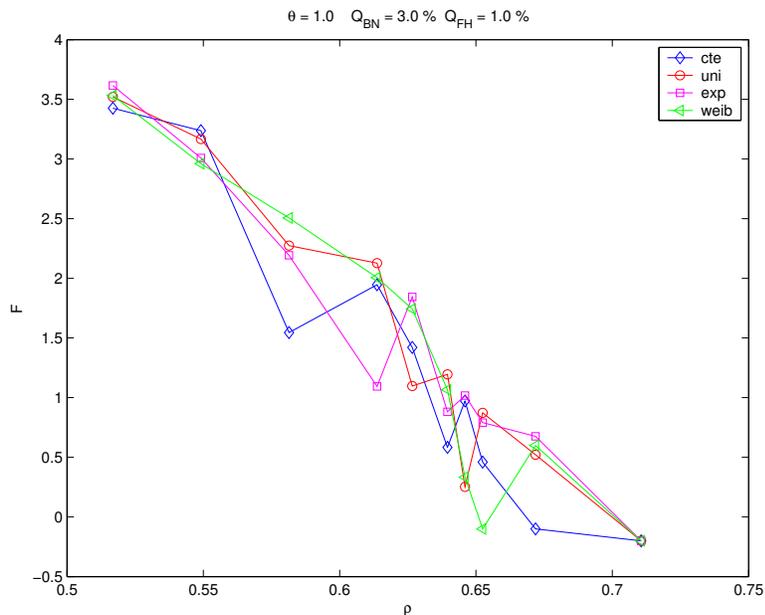


Figura 9.120: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC6M.

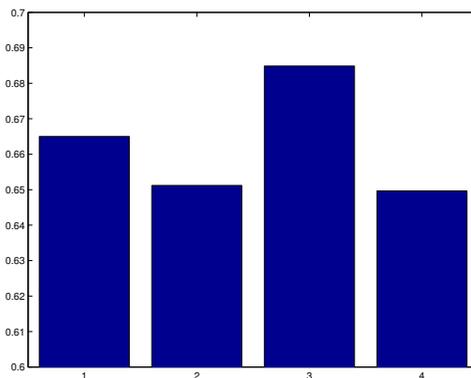


Figura 9.121: Parâmetro I para usuários da classe 1, considerando que o tempo de permanência do usuário na célula é dado pelas Distribuições (1)Constante, (2)Uniforme, (3)Exponencial, (4)Weibull, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC6M.

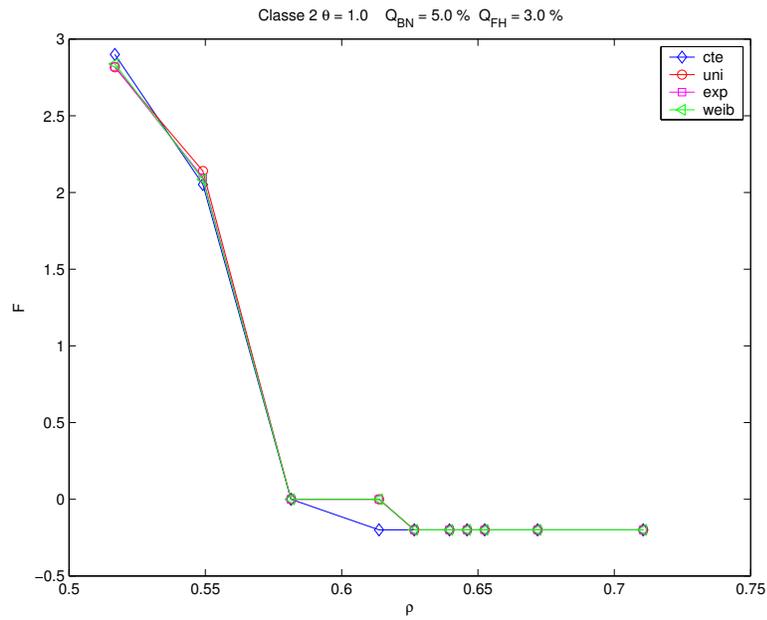


Figura 9.122: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC6M.

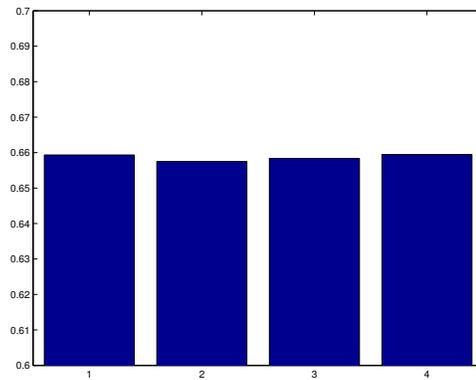


Figura 9.123: Parâmetro I para usuários da classe 2, considerando que o tempo de permanência do usuário na célula é dado pelas Distribuições (1)Constante, (2)Uniforme, (3)Exponencial, (4)Weibull, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC6M.

Caso Taxa de mobilidade $\theta = 6$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, algoritmo CAC6M.

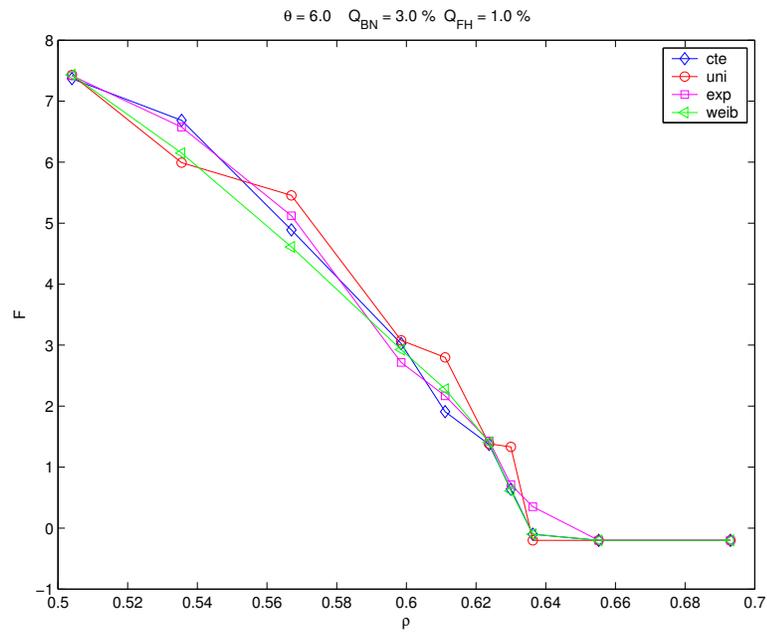


Figura 9.124: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC6M.

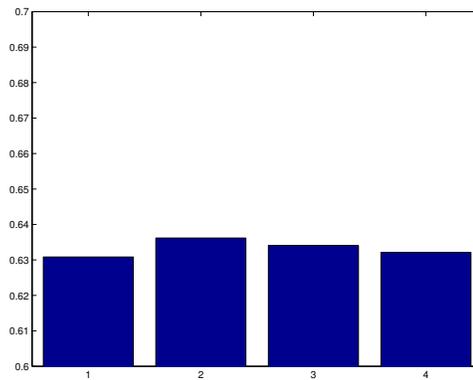


Figura 9.125: Parâmetro I para usuários da classe 1, considerando que o tempo de permanência do usuário na célula é dado pelas Distribuições (1)Constante, (2)Uniforme, (3)Exponencial, (4)Weibull, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC6M.

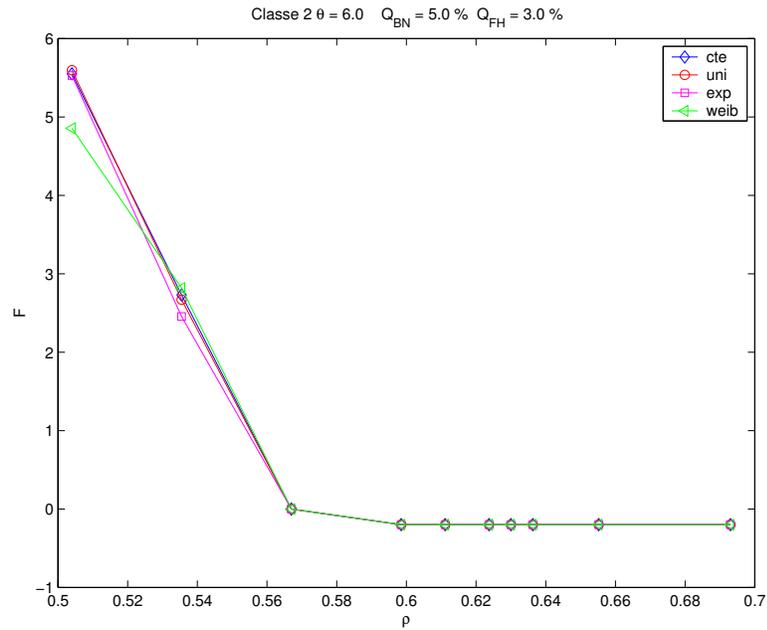


Figura 9.126: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC6M.

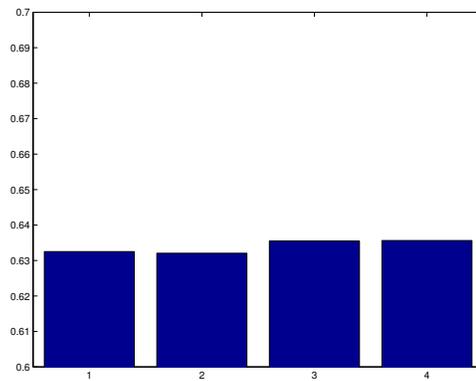


Figura 9.127: Parâmetro I para usuários da classe 2, considerando que o tempo de permanência do usuário na célula é dado pelas Distribuições (1)Constante, (2)Uniforme, (3)Exponencial, (4)Weibull, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC6M.

9.5.10

Comportamento do algoritmo CAC7M com a Distribuição que Modela o Tempo de Permanência de usuário na Célula

Caso Taxa de mobilidade $\theta = 1$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, algoritmo CAC7M.

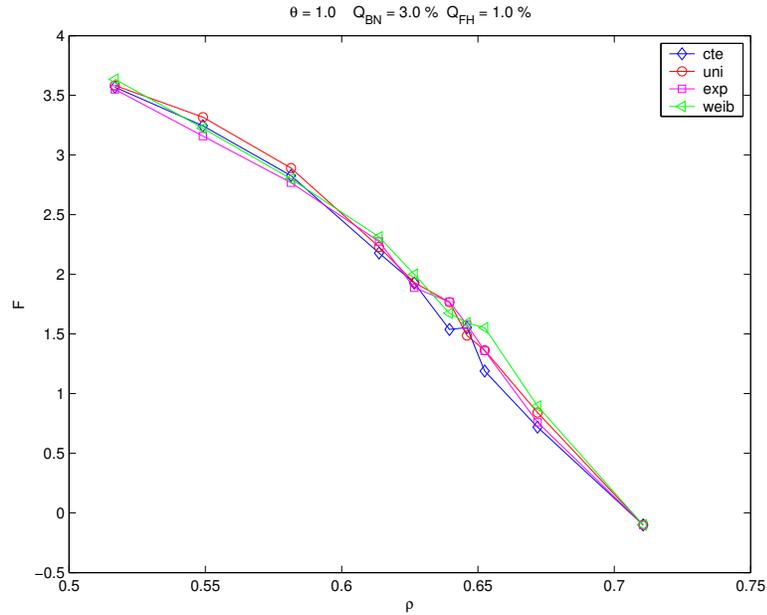


Figura 9.128: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC7M.

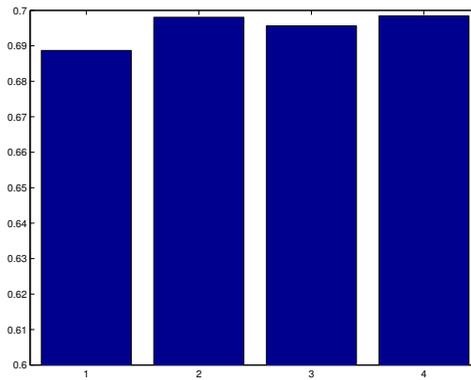


Figura 9.129: Parâmetro I para usuários da classe 1, considerando que o tempo de permanência do usuário na célula é dado pelas Distribuições (1)Constante, (2)Uniforme, (3)Exponencial, (4)Weibull, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC7M.

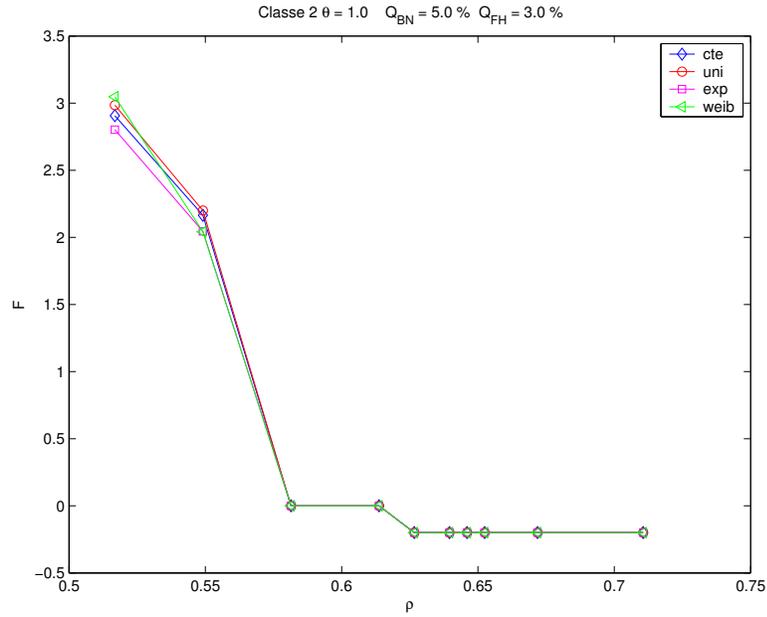


Figura 9.130: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC7M.

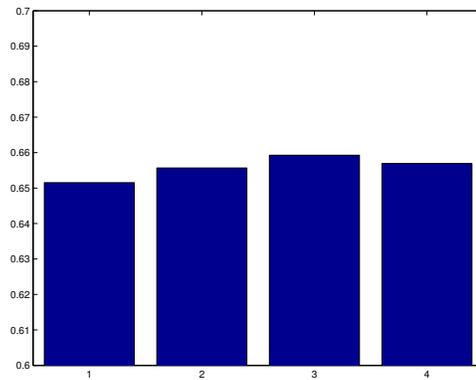


Figura 9.131: Parâmetro I para usuários da classe 2, considerando que o tempo de permanência do usuário na célula é dado pelas Distribuições (1)Constante, (2)Uniforme, (3)Exponencial, (4)Weibull, taxa de mobilidade $\theta = 1$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC7M.

Caso Taxa de mobilidade $\theta = 6$ e requisito de Qualidade de Serviço $Q\% = (Q_{BN1}, Q_{FH1}, Q_{BN2}, Q_{FH2}) = (3, 1, 5, 3)$, algoritmo CAC7M.

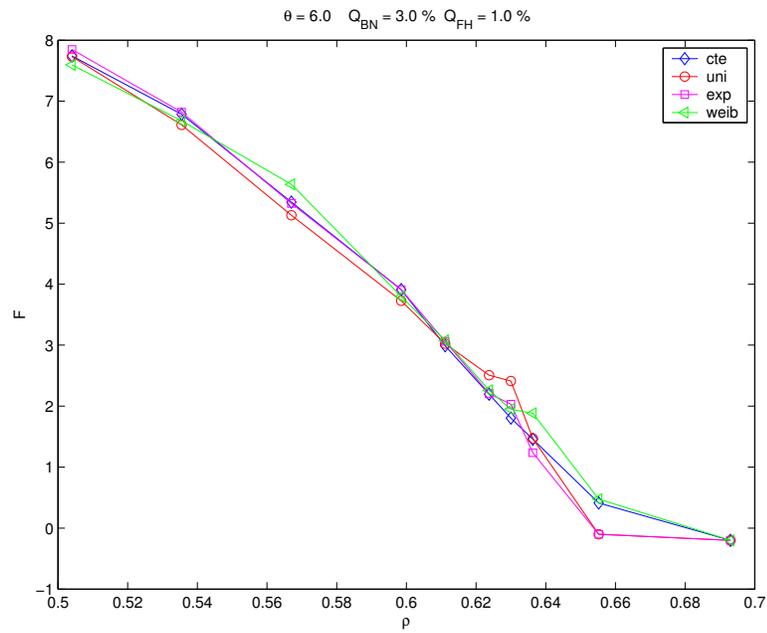


Figura 9.132: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 1, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC7M.

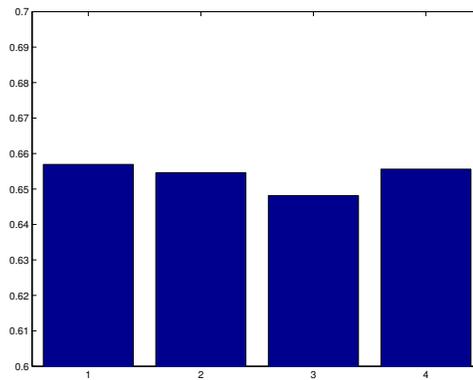


Figura 9.133: Parâmetro I para usuários da classe 1, considerando que o tempo de permanência do usuário na célula é dado pelas Distribuições (1)Constante, (2)Uniforme, (3)Exponencial, (4)Weibull, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC7M.

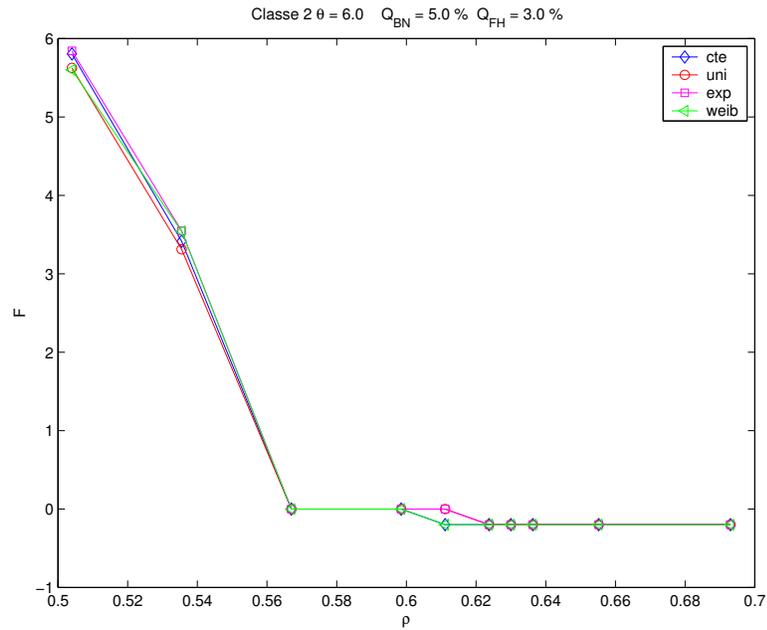


Figura 9.134: Parâmetro F versus intensidade de tráfego de entrada normalizada para usuários da classe 2, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC7M.

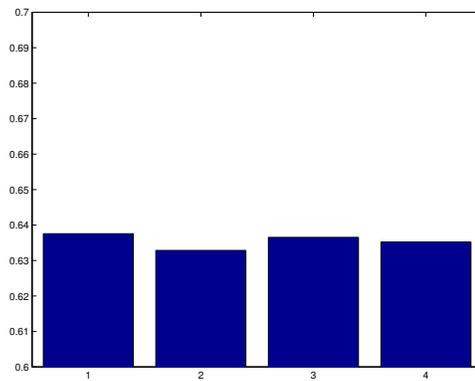


Figura 9.135: Parâmetro I para usuários da classe 2, considerando que o tempo de permanência do usuário na célula é dado pelas Distribuições (1)Constante, (2)Uniforme, (3)Exponencial, (4)Weibull, taxa de mobilidade $\theta = 6$, requisito $Q\%(3, 1, 5, 3)$, algoritmo CAC7M.

9.6 Conclusão

Neste capítulo foram apresentados quatro algoritmos propostos para M classes de usuários. Os algoritmos foram testados no caso de duas classes de usuários, $M = 2$, então são quatro tipos diferentes de usuários: os usuários novos e em *handoff* da classe 1, $(N1, H1)$, e os usuários novos e em *handoff* da classe 2, $(N2, H2)$.

Todos algoritmos propostos para M classes são extensões daqueles definidos para uma classe no capítulo 5, e são dinâmicos e distribuídos.

Uma traço específico dos algoritmos propostos nesta tese é a preocupação simultânea com os $2M$ requisitos de qualidade de serviço, supondo M classes de usuários (novos e em *handoff*).

Outra característica importante das simulações realizadas neste trabalho é que o desempenho de cada classe de usuário é avaliado segundo as medidas de bloqueio obtidas e os respectivos requisitos de qualidade. Cada setor registra a entrada, ou não, de determinado tipo de chamada através de janelas de registro deslizantes descritas no capítulo 4. Os algoritmos dinâmicos de Controle de Admissão de Chamadas, CAC, baseiam-se nos valores instantâneos dos percentuais de bloqueio de novas chamadas e percentuais de falhas de chamadas em *handoff* de cada classe, denominada γ , denotados pelo par $(P_{BN}; P_{FH})_{\gamma}$, cujas medidas são chamadas de $(B_{BN}; B_{FH})_{\gamma}$. A tomada de decisão sobre aceitar ou não determinada chamada nos algoritmos propostos neste trabalho é feita de acordo com a comparação dos valores medidos, $(B_{BN}; B_{FH})_{\gamma}$, com os respectivos requisitos de QoS chamados de $(Q_{BN}; Q_{FH})_{\gamma}$.

Além da variação dos requisitos de qualidade, os algoritmos têm o seu desempenho avaliado de acordo com a taxa de mobilidade, θ , definida em (3-39), que obriga o sistema a se adaptar às características do perfil de tráfego em andamento.

Neste trabalho, em todos os cenários a taxa de chegada de usuários da classe 2 é igual à taxa de chegada de usuários da classe 1. Isso faz com que o sistema tenda a ser utilizado em 66% pela classe 2 que requer $2UB$ e em 33% pela classe 1 que solicita $1UB$. Deve-se notar que tal caso sobrecarrega um pouco o sistema com os usuários da classe 2 que requerem uma quantidade maior de recursos.

Ao aplicar a técnica de reserva de recursos com o objetivo de atender aos requisitos de qualidade de serviço, observou-se que é possível determinar o valor dos recursos a serem reservados de modo a se obter uma utilização

máxima. No entanto, a solução é possível somente até certo valor de intensidade de tráfego de entrada determinado pelas necessidades de banda de cada classe e pelo conjunto de requisitos de QoS. Para valores de intensidade de tráfego superiores, não é mais possível atender aos requisitos de qualidade de serviço.

Observa-se que o algoritmo *CAC7M* busca acompanhar a tabela com os valores ideais do número de recursos a ser reservado. Como o valor do número de recursos varia rapidamente com a intensidade de tráfego e com a taxa de mobilidade, sistemas que fossem mais eficientes na medida dos valores de (θ, ρ) tirariam um melhor proveito do algoritmo *CAC7M*. A medida apurada do perfil de tráfego está fora do escopo deste trabalho.

Nos gráficos correspondentes ao caso de distribuição exponencial para o tempo de permanência do usuário na célula, observa-se que o algoritmo *CAC7M* tende a apresentar um bom desempenho, principalmente para a classe 1. Os algoritmos *CAC1M* e *CAC2M* apresentam bons desempenhos, porém isto não ocorre em todos os cenários. O algoritmo *CAC6M*, diante do fato de só se pretender alterar o número de recursos em faixas restritas, apresenta um comportamento que oscila na região de intensidade de tráfego próxima ao valor ρ_Q .

Outra observação interessante dos resultados apresentados neste capítulo é que o comportamento dos algoritmos não muda muito quando o tempo de permanência do usuário na célula é modelado por distribuições diferentes da exponencial. A distribuição que apresentou comportamento menos próximo ao da exponencial foi a distribuição Weibull. No caso em que a taxa de mobilidade é alta as curvas relativas às distribuições tendem a ficar bem próximas das curvas do caso exponencial. No caso em que a taxa de mobilidade é baixa, a distribuição constante para o tempo de permanência provoca maiores percentuais de bloqueio pois a constante é maior para mobilidade baixa fazendo com que ocorra um maior número de chamadas mais longas em relação ao caso de mobilidade alta.