

# Controle de Admissão *Fuzzy* baseado em Limites Dinâmicos de Congestionamento Para Redes de Celulares

Anna Izabel J. Tostes, Fátima F.L.P. Duarte-Figueiredo, Luis E. Zárate

<sup>1</sup>Mestrado em Informática – Pontifícia Universidade Católica de Minas Gerais (PUCMG)  
Belo Horizonte – MG – Brasil

annatostes@gmail.com, {fatimafig, zarate}@pucminas.br

**Abstract.** *One challenge of mobile networks is ensuring Quality of Service (QoS), measured by availability and users' satisfaction. A key to the success of QoS is Call Admission Control (CAC). One example is CAC-RD, based on resources Reservation and network Diagnosis. CAC-RD blocks lower-priority applications using static utilization thresholds. This static characteristic prevents the maximum utilization of network resources, under certain traffic conditions. This paper proposes CAC-RDF, an innovative strategy of dynamic calls' blocking through fuzzy logic, according to the level of network congestion. CAC-RDF improves CAC-RD's resources distribution ensuring adequate QoS.*

**Resumo.** *Um desafio das redes de celulares é a garantia de Qualidade de Serviço (QoS), medida pela disponibilidade e satisfação dos usuários. Um ponto nevrálgico para o sucesso de QoS é o Controle de Admissão de Chamadas (CAC). Um exemplo é o CAC-RD, baseado em Reserva de recursos e Diagnóstico da rede. O CAC-RD bloqueia chamadas menos prioritárias por limites estáticos de utilização. Esse caráter estático impede o máximo aproveitamento dos recursos da rede, em certas condições de tráfego. Este artigo propõe o CAC-RDF, uma estratégia inovadora de bloqueio dinâmico de chamadas através de lógica fuzzy, de acordo com o nível de congestionamento da rede. O CAC-RDF melhora a distribuição de recursos do CAC-RD garantindo QoS adequada.*

## 1. Introdução

As redes de celulares da terceira geração (3G) oferecem diversos serviços, tais como a navegação na web, streaming de vídeo, transmissão de imagens, além de chamadas de voz e SMS (*Short Message Service*). Nessas redes, a Qualidade de Serviço (QoS) é mensurada não apenas pela disponibilidade de recursos, mas também pela satisfação dos usuários. Como serviços diferentes possuem requisitos de qualidade diferentes, essas redes precisam de políticas que garantam QoS adequada para cada tipo de aplicação. Este trabalho aborda chamadas no contexto de demanda de qualquer serviço (voz e dados). Controle de Admissão de Chamadas (CAC) é um ponto nevrálgico para o sucesso de QoS em redes de celulares. O CAC decide sobre a aceitação ou o bloqueio de uma nova chamada, de acordo com a disponibilidade de recursos da rede. A relevância de se ter um CAC é evitar a aceitação indiscriminada de serviços e o congestionamento da rede.

Existem, na literatura, propostas de CACs para redes 3G UMTS (*Universal Mobile Telecommunication System*). Um exemplo é o CAC-RD [Tostes et al. 2010,

Storck et al. 2008], baseado em Reserva de recursos e Diagnóstico da rede. O CAC-RD bloqueia chamadas menos prioritárias, a partir de limites estáticos de utilização da rede. Serviços *background* são bloqueados a partir de 40% de utilização, mesmo quando a rede não está congestionada, no CAC-RD. Esse é o problema abordado por este trabalho: o desperdício de recursos da rede ocasionado quando o CAC-RD bloqueia novas chamadas, seguindo limites estáticos.

Como solução, este trabalho propõe o CAC-RD *Fuzzy* (CAC-RDF): um CAC-RD com o bloqueio dinâmico de chamadas menos prioritárias, de acordo com congestionamento da rede. A estratégia inovadora de utilizar lógica *fuzzy* para bloquear dinamicamente de chamadas é a originalidade deste trabalho. Se a rede não estiver congestionada, qualquer nova aplicação é aceita. Caso a rede esteja pré-congestionada (tendendo ao congestionamento eminente), chamadas menos prioritárias são bloqueadas, de acordo com o nível de Qualidade de Serviço (QoS) atendido para as chamadas já estabelecidas. Em caso de congestionamento, o CAC-RDF não aceita nenhuma nova chamada até que recursos sejam liberados. O desafio foi saber como dinamizar os limites de aceitação e quando bloquear as chamadas menos prioritárias de acordo com esses limites. Para isso, foi utilizada a lógica *fuzzy* (LF), que possibilita a criação de um mecanismo qualitativo, através de regras representadas naturalmente, com base em descrições linguísticas (notações como baixo, médio e alto). O CAC-RDF possui dois objetivos: maior aceitação de chamadas prioritárias que o CAC-RD e melhor distribuição dos recursos disponíveis na rede. Simulações com o CAC-RD e CAC-RDF foram realizadas para comparar o desempenho dos limites estáticos com os dinâmicos, no módulo de simulação E-UMTS [SEACORN 2004], desenvolvido no NS-2<sup>1</sup>. Os resultados mostraram que o CAC-RDF melhora a distribuição de recursos do CAC-RD, garantindo QoS adequada.

Este trabalho está organizado em cinco seções. A seção 2 explica a especificação de QoS para as redes UMTS. A seção 3 apresenta controles de admissão e trabalhos relacionados. A seção 4 descreve o CAC-RDF com os limites dinâmicos. A seção 5 mostra as simulações realizadas. A seção 6 conclui este trabalho.

## 2. Redes UMTS e Qualidade de Serviço

A rede 3G mais utilizada no mundo é a rede UMTS (*Universal Mobile Telecommunication System*) [3GPP 2001]. Uma extensão é a E-UMTS (*Enhanced UMTS*), que inclui HSPA (*High Speed Packet Access*) na sua arquitetura, com largura de banda de até 15 Mbps.

A Qualidade de Serviço (QoS) se relaciona ao grau de satisfação dos usuários de uma rede [Steinmetz and Wolf 1997]. Em redes UMTS, os tipos de chamadas (aplicações) são agrupadas em quatro classes de serviços: (1) *conversational*; (2) *streaming*; (3) *interactive*; e (4) *background*. Os parâmetros de QoS são o atraso, o jitter e a vazão por usuário. Para se garantir uma QoS adequada, as redes UMTS necessitam de técnicas para avaliar parâmetros de QoS de forma a obtê-los com valores ideais. A tabela 1 mostra os requisitos de qualidade para esses parâmetros, especificadas pelo 3GPP em [3GPP 2001]. Neste trabalho, foram selecionados os serviços mais sensíveis de cada classe. Para conversa de voz (1), o limite máximo de jitter é 1ms. Para vídeo (2), o atraso máximo deve ser de 10s. Para navegação na web (3), o atraso tolerado é entre 2 e 4s. Para SMS (4), o limite de atraso deve ser 30s.

<sup>1</sup>O NS-2 (*Network Simulator – versão 2*) está disponível em <http://www.isi.edu/nsnam/ns/>.

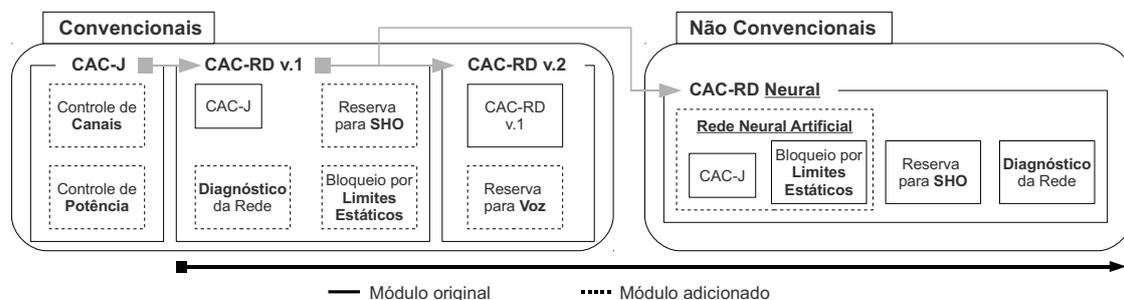
**Tabela 1. Exigências de QoS na rede UMTS.**

#	Classe	Aplicação	Demanda	Atraso		Jitter	Vazão/Usuário
1	<i>Conv.</i> <sup>1</sup>	Conversa de voz	Alta	Preferencial	< 150ms	< 1 ms	4–13 Kbps
				Limite	< 400ms		
2	<i>Stre.</i>	Video <i>One-way</i>	Baixa	Limite	< 10s	< 2s	32–384 Kbps
3	<i>Inte.</i> <sup>2</sup>	Navegação na Web	Alta	Limite	< 4s <sup>3</sup>		
4	<i>Back.</i>	SMS	Média	Limite	< 30s	–	2,8 Kbps

### 3. Controles de Admissão e Trabalhos Relacionados

Os Controles de Admissão de Chamadas (CACs) são mecanismos de QoS que decidem a aceitação de novas chamadas na rede. CACs bem projetados implicam em um serviço de telefonia celular altamente eficiente, sem queda de conexões e/ou não disponibilidade de serviços [Tostes and Duarte-Figueiredo 2009]. A ideia é que CACs saibam quando aceitar e rejeitar chamadas, aproveitando, da melhor forma, os recursos da rede.

De forma geral, os artigos de CACs podem ser classificados em: (1) CACs convencionais e (2) CACs não convencionais. Os CACs convencionais utilizam técnicas tradicionais [Ghaderi and Boutaba 2006], como controle de canais e de potência (e.g., o CAC-J [Antoniou 2004]), priorização de recursos para chamadas mais sensíveis a QoS, reserva de recursos da rede e bloqueio de chamadas, usando limites estáticos (CAC-RD) ou dinâmicos. As vantagens dos CACs convencionais são a alta eficiência, a baixa complexidade e o baixo *overhead*. Os CACs não convencionais, além de possuírem as vantagens dos convencionais, podem ser adaptativos e estáveis. Eles utilizam a inteligência computacional incorporadas às suas técnicas através do uso de redes neurais [Hiramatsu 1989, Tostes 2010] ou lógica *fuzzy* (LF) [Pedrycz and Vasilakos 2000]. A LF trabalha com incertezas do sistema. Isso permite que os CACs não convencionais administrem as requisições, qualitativamente, através de regras representadas com base em descrições linguísticas simples (e.g., se QoS = ruim, então rede = congestionada). Em [Ascia et al. 1997], foi proposto um CAC projetado para tomar decisões de aceitação de acordo com flutuações de tráfego, prevista pelo mecanismo de LF. As variáveis do mecanismo de LF remetem a três informações da rede: (1) congestionamento, (2) qualidade e (3) capacidade da rede. Essa proposta diferencia deste trabalho por não abordar os níveis de QoS para cada classe de serviço e não discriminar a aceitação de chamadas de acordo com a prioridade da requisição, fatores que são importantes na distribuição de carga na rede e que foram utilizados neste trabalho.

**Figura 1. Evolução da pesquisa com o CAC-RD.**

Em nossos trabalhos anteriores, os CACs foram desenvolvidos em três etapas, conforme mostra a figura 1. Nas duas primeiras etapas, foram propostos dois CACs con-

vencionais: (1) CAC-RD v.1, evolução do CAC-J de Antoniou (2004) com a reserva de recursos para *soft handover* (SHO) e um diagnóstico da rede [Tostes et al. 2010]; e (2) CAC-RD v.2, que diminui o bloqueio de chamadas de voz através da reserva recursos também para chamadas de voz [Tostes and Duarte-Figueiredo 2009]. Na terceira fase foi proposto o CAC não convencional CAC-RD Neural [Tostes et al. 2009]. Ele permite a expansão do cenário de simulação com o comportamento do CAC-RD v.1 devido à substituição dos seguintes módulos do CAC-RD v.1 por uma rede neural artificial: bloqueio por limites estáticos, controle de canais e de potência.

Como mostra a figura, o CAC-RD possui os seguintes módulos: bloqueios por limites estáticos; reserva de canais; diagnóstico da rede; e controle de canais e de potência (CAC-J). O módulo de bloqueio por limites estáticos rejeita chamadas menos prioritárias quando a utilização da rede atinge determinados limites. Se a utilização da rede estiver em até 40%, todas as chamadas são aceitas. Caso a utilização esteja entre 40% e 50%, chamadas de não tempo real (*background*) são bloqueadas. Entre 50% e 65% de utilização, chamadas das classes *interactive* e *background* são bloqueadas. Entre 65% e 75%, chamadas da classe *streaming* são rejeitadas. Acima de 75% de utilização, todas as novas chamadas são bloqueadas, inclusive as da classe *conversational*. Em [Tostes et al. 2010], foi avaliado o desempenho do CAC-RD em relação ao CAC-J, através de simulações conduzidas no NS-2. Os resultados demonstram que, em geral, o CAC-RD reduziu em 40% os bloqueios de SHO e em 11% os bloqueios de chamadas de voz (classe *conversational*). Como o problema do CAC-RD é um possível desperdício dos recursos da rede, um dos trabalhos futuros apresentados em [Storck 2007] foi a utilização de limites dinâmicos.

#### 4. CAC-RD Fuzzy

Conforme mencionado, este trabalho propõe o CAC-RD Fuzzy (CAC-RDF): uma evolução do CAC-RD com bloqueios dinâmicos de chamadas menos prioritárias por meio de lógica *fuzzy* (LF). Seu compromisso é a garantia do melhor aproveitamento dos recursos da rede com a priorização de serviços. O objetivo do CAC-RDF é evitar o desperdício de recursos do CAC-RD quando a rede não está congestionada. Caso a rede esteja altamente disponível (descongestionada), qualquer nova chamada é aceita. Caso contrário, chamadas menos prioritárias são bloqueadas de acordo com a utilização de limites dinâmicos, e não estáticos, como era no CAC-RD. Esse controle proporciona à rede maior adaptabilidade e estabilidade na sua tomada de decisão.

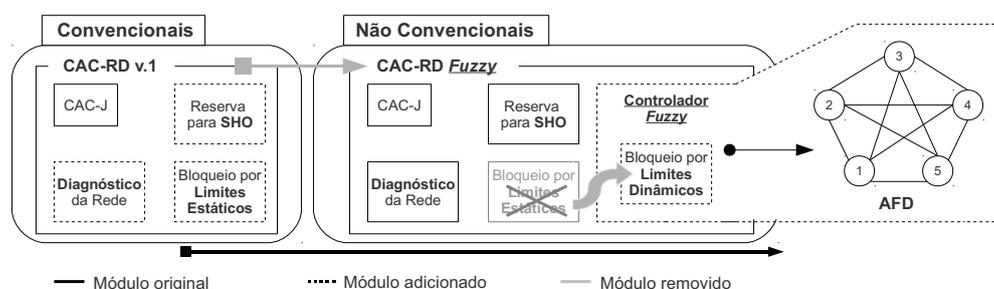


Figura 2. Módulos do CAC-RD Fuzzy.

Assim como no CAC-RD, a arquitetura do CAC-RDF é dividida em módulos, como mostra a figura 2. Todos os módulos do CAC-RD foram mantidos, exceto o módulo

de bloqueio por limites estáticos. Esse módulo foi substituído pelo módulo de bloqueio por limites dinâmicos. Esse módulo é um mecanismo de LF, representado na figura pelo autômato finito determinístico (AFD), que analisa a rede e decide quais classes devem ser bloqueadas. A partir do nível de garantia de QoS, ele define o estado da rede: (1) todas as chamadas são aceitas (AA); (2) bloqueia chamadas *background* (BB); (3) bloqueia chamadas *background* e *interactive* (BI); (4) bloqueia chamadas *background*, *interactive* e *streaming* (BS); e (5) todas as chamadas são bloqueadas (BC).

Quatro etapas foram seguidas para criar o mecanismo de LF. Na etapa 1, foram definidas recomendações *fuzzy* para a criação do mecanismo de LF (Representação *Fuzzy*). Na etapa 2, foi escolhida uma biblioteca de LF que integrasse com o NS-2, onde foi desenvolvido o CAC-RD. Na etapa 3, foi analisado o desempenho da rede E-UMTS para se verificar se os níveis de QoS definidos pelo 3GPP seriam atendidos ou se a rede entrava em congestionamento. Na etapa 4, o mecanismo de LF foi modelado de acordo com as recomendações da representação *fuzzy* desenvolvidas na etapa 1.

#### 4.1. Etapas 1 e 2: Representação *Fuzzy* e Biblioteca de Lógica *Fuzzy*

Conforme dito acima, as duas primeiras etapas deste trabalho envolveram a criação das recomendações da representação *fuzzy* de CAC e a seleção da biblioteca de LF. A representação *fuzzy* foi estabelecida de acordo com os conceitos de LF. O resultado da representação é o mecanismo de LF, responsável pela dinamicidade dos limites do CAC.

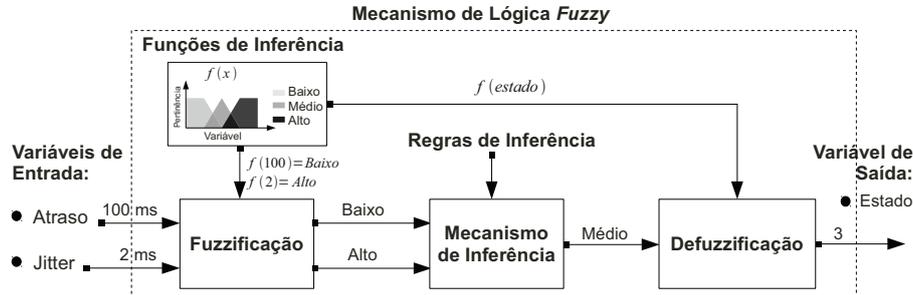


Figura 3. Funcionamento da representação *fuzzy*.

O funcionamento do mecanismo de LF é representado pela figura 3. O primeiro passo é a definição das variáveis de entrada e saída e suas respectivas funções de inferência. Em seguida, acontece o processo de fuzzificação, que transforma os valores reais de entrada (no exemplo da figura, 100 ms e 2 ms) em valores linguísticos (na figura, baixo e alto) a partir da sua função de inferência específica ( $f(x)$ , sendo  $x$  a variável). No terceiro passo, o mecanismo de inferência deduz a variável de saída a partir das entradas linguísticas e das regras de inferência, definidas por um especialista. O último passo consiste em traduzir o valor linguístico de saída (na figura, médio) para valor real (na figura, estado 3) a partir de um algoritmo de defuzzificação que consulta as funções de inferência.

A linguagem utilizada para empregar a LF em um sistema é a Linguagem de Controle *Fuzzy* (FCL), uma linguagem específica de domínio desenvolvida pelo *International Electrotechnical Commission* (IEC) 61131-7 [Commission 2009]. Utilizada para modelar o mecanismo de LF, sua vantagem é a manutenção: qualquer modificação que precise ser realizada na modelagem do controle pode ser feita com a alteração de um arquivo FCL.

Neste trabalho, a FCL será utilizada na representação *fuzzy* do CAC. Como biblioteca de LF, foi escolhido o *jFuzzyLogic*, uma biblioteca de código aberto, gratuito e portátil<sup>2</sup>.

A motivação para essa representação *fuzzy* está no desafio dos CACs não convencionais de integrar a Computação Móvel e a Inteligência Computacional. A complexidade da modelagem *fuzzy* exige recomendações de boas práticas para o estabelecimento de um CAC *Fuzzy*. Seguem os sete passos da representação *fuzzy* definidos neste trabalho: (1) definir o número de limites do CAC e suas respectivas classes de serviços; (2) escolher as variáveis linguísticas (variáveis de entrada e de saída); (3) definir as funções de inferência das variáveis linguísticas; (4) definir as regras de inferência para o mecanismo de LF; (5) escolher algoritmo de defuzzificação; (6) implementar o mecanismo de LF como módulo no CAC; (7) testar a dinamicidade dos limites.

#### 4.2. Etapa 3: Análise de Desempenho da Rede UMTS

A etapa 3 consiste em analisar a rede UMTS, através de simulações com o CAC-J, para se determinar o domínio dos parâmetros de QoS, por classe de serviço e BS (*Base Station*) a fim de se identificar cenários (pontos) de congestionamento. Simulações foram conduzidas no módulo E-UMTS, desenvolvido no NS-2. O ambiente de simulação foi o cenário urbano [Antoniu 2004] com os parâmetros mostrados na tabela 2. Como Antoniu (2004) definiu que o cenário urbano ideal deve apresentar até 200 usuários em seu estudo sobre a cobertura e a capacidade do módulo E-UMTS (com os mesmos parâmetros adotados neste trabalho), foram simulados 210 usuários, 100% ativos em 1 célula, durante 200s de simulação, com 309 chamadas requisitadas com diferentes tempos de duração.

**Tabela 2. Parâmetros adotados para o cenário de simulação urbano.**

Parâmetros do Cenário Urbano	
Modelo de propagação de rádio	<i>Hata - COST 231</i>
Probabilidade do usuário estar ativo	<i>Busy Hour Call Attempts (BHCA)</i>
Ambiente de operação	<i>Outdoor</i>
Modelo de mobilidade	<i>Gauss-Markov</i> : velocidade de 50 km/h ou aleatória
Limites estáticos do CAC-RD	(40%, 50%, 65%, 75%)
Topologia	1 célula → BS tri-setorial (120°/sector)
<i>Conversational</i> (12,2 kbps)	42% de usuários
<i>Streaming</i> (768 kbps)	16% de usuários
<i>Interactive</i> (384 kbps)	18,50% de usuários
<i>Background</i> (144 kbps)	23,50% de usuários

Os resultados demonstraram pontos de congestionamento para as classes *conversational* (todo tempo), *streaming* (após 40s) e *interactive* (após 30s), mostrados pelos gráficos da figura 4. A partir da identificação desse congestionamento, as variáveis linguísticas da representação *fuzzy* do CAC-RD foram definidas.

#### 4.3. Etapa 4: Modelagem do Mecanismo de LF

A etapa 4 deste trabalho foi a aplicação dos sete passos da representação *fuzzy* no módulo de bloqueios do CAC-RD, tornando-o dinâmico. O novo módulo se adapta ao estado do tráfego da rede. Ele foi implementado no *Radio Resource Management* (RRM), componente da rede UMTS que faz a gerência dos recursos de rádio da rede e é responsável por responder à diversas BSs sobre diversos pedidos de conexão (chamadas).

<sup>2</sup>Disponível em <http://sourceforge.net/projects/jfuzzylogic/>

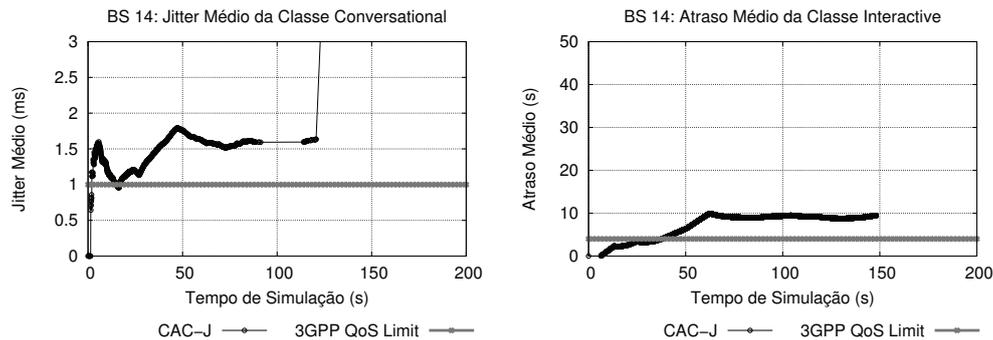


Figura 4. Desempenho da QoS para cada classe de serviço na BS 14.

No primeiro passo, foram definidos os limites de bloqueios do mecanismo de LF. Como o CAC-RD possui quatro limites, um para cada classe de serviço, a mesma quantidade de limites foi utilizada no mecanismo de LF. Foi mantida a mesma prioridade entre as classes do CAC-RD, partindo da classe mais prioritária: (1) *conversational*; (2) *streaming*; (3) *interactive*; e (4) *background*.

Os passos 2 e 3 da representação podem ser visualizados na tabela 3. O passo 2 é a seleção das variáveis linguísticas do mecanismo de LF. As variáveis de entrada escolhidas foram atraso de *conversational* (CD); jitter de *conversational* (CJ); atraso de *streaming* (SD); jitter de *streaming* (SJ); atraso de *interactive* (ID). A escolha dessas variáveis foi para garantir que, se a rede não consegue garantir QoS fim-a-fim para uma classe de serviço, o CAC não deve mais aceitar chamadas dessa classe, já que elas também não terão qualidade garantida. Quanto à variável de saída, para manter a mesma ideia do módulo do CAC-RD, foi escolhido o estado de decisão do módulo. Essa variável possui os mesmos cinco estados de aceitação do módulo do CAC-RD: (1) todas as chamadas são aceitas (AA); (2) bloqueia chamadas *background* (BB); (3) bloqueia chamadas *background* e *interactive* (BI); (4) bloqueia chamadas *background*, *interactive* e *streaming* (BS); e (5) todas as chamadas são bloqueadas (BC). O passo 3 envolve a definição das funções de pertinência para cada variável. As funções de pertinência definem valores linguísticos para valores determinísticos de um universo de discurso. Cada variável linguística assume um rótulo *\_baixo*, *médio* ou *alto*, atribuído pela função de pertinência, de acordo com o seu valor determinístico. Esse grau de pertinência varia de zero a um. Por exemplo, se uma variável possui grau de pertinência 0,7 para o rótulo “baixo” e de 0,3 para o rótulo “médio”, significa que ela é mais baixa do que média.

O passo 4 envolve a definição das regras de inferência, feitas por um especialista. Ele define regras SE-ENTÃO que determinam o comportamento do mecanismo de LF. Nesse ponto, são definidas as relações entre as variáveis linguísticas. As regras do mecanismo de LF, definidas pelos autores, são apresentadas na tabela 4. Na representação *fuzzy* do CAC-RD, foram criadas 27 regras de inferência, apresentadas na tabela 4. A ideia dessas regras é bloquear as chamadas menos prioritárias, de acordo com o nível de qualidade das classes de serviço. Caso uma variável de entrada seja média, é bloqueada uma classe menos prioritária. Como na regra 2 o atraso de *interactive* é médio, a saída é BB (bloqueia *background*). Caso contrário, duas classes menos prioritárias são bloqueadas (BI).

O passo 5 é a escolha do algoritmo de defuzzificação. A defuzzificação calcula um valor escalar para a saída, a partir da área no gráfico da função de pertinência da variável

**Tabela 3. Funções de inferência.**

Var. <sup>1</sup>	Função de Inferência
CD	$\mu_B(x) = \begin{cases} 1, & \text{se } x \leq 100 \\ -0,01x + 2, & \text{se } 100 < x \leq 200 \end{cases}$
	$\mu_M(x) = \begin{cases} 0,01x - 1, & \text{se } 100 \leq x \leq 200 \\ 1, & \text{se } 200 < x \leq 300 \\ -0,01x + 4, & \text{se } 300 < x \leq 400 \end{cases}$
	$\mu_A(x) = \begin{cases} 0,01x - 3, & \text{se } 300 \leq x \leq 400 \\ 1, & \text{se } 400 < x \leq 450 \end{cases}$
CJ	$\mu_B(x) = \begin{cases} -0,5x + 1, & \text{se } x \leq 2 \\ 0,5x, & \text{se } x > 2 \end{cases}$
	$\mu_A(x) = \begin{cases} -0,5x + 1, & \text{se } x \leq 2 \\ 0,5x, & \text{se } x > 2 \end{cases}$
SD	$\mu_B(x) = \begin{cases} 1, & \text{se } x \leq 1 \\ -0,5x + 1,5, & \text{se } 1 < x \leq 3 \end{cases}$
	$\mu_M(x) = \begin{cases} 0,5x - 0,5, & \text{se } 1 \leq x \leq 3 \\ 1, & \text{se } 3 < x \leq 6 \\ -0,25x + 2,5, & \text{se } 6 < x \leq 10 \end{cases}$
	$\mu_A(x) = \begin{cases} 0,25x - 1,5, & \text{se } 6 \leq x \leq 10 \\ 1, & \text{se } 10 < x \leq 15 \end{cases}$
SJ	$\mu_B(x) = \begin{cases} -0,67x + 1, & \text{se } x \leq 1,5 \\ 0,67x, & \text{se } x > 1,5 \end{cases}$
	$\mu_M(x) = \begin{cases} -x + 2,5, & \text{se } 1,5 < x \leq 2,5 \\ x - 1,5, & \text{se } 1,5 \leq x \leq 2,5 \end{cases}$
	$\mu_A(x) = \begin{cases} 1, & \text{se } 1,5 < x \leq 4 \\ x - 1,5, & \text{se } 1,5 \leq x \leq 2,5 \end{cases}$
ID	$\mu_B(x) = \begin{cases} 1, & \text{se } x \leq 0,5 \\ -0,67x + 1,33, & \text{se } 0,5 < x \leq 2 \end{cases}$
	$\mu_M(x) = \begin{cases} 0,67x - 0,33, & \text{se } 0,5 \leq x \leq 2 \\ -0,5x + 2, & \text{se } 2 < x \leq 4 \end{cases}$
	$\mu_A(x) = \begin{cases} 0,5x - 1, & \text{se } 2 \leq x \leq 4 \\ 1, & \text{se } 4 < x \leq 6 \end{cases}$
Saída <sup>2</sup>	$\mu_{AA}(x) = \begin{cases} 1, & \text{se } x \leq 5 \\ -0,2x + 2, & \text{se } 5 < x \leq 10 \end{cases}$
	$\mu_{BB}(x) = \begin{cases} 0,2x - 1, & \text{se } 5 \leq x \leq 10 \\ -0,2x + 3, & \text{se } 10 < x \leq 15 \end{cases}$
	$\mu_{BI}(x) = \begin{cases} 0,2x - 2, & \text{se } 10 \leq x \leq 15 \\ -0,2x + 4, & \text{se } 15 < x \leq 20 \end{cases}$
	$\mu_{BS}(x) = \begin{cases} 0,2x - 3, & \text{se } 15 \leq x \leq 20 \\ -0,2x + 5, & \text{se } 20 < x \leq 25 \end{cases}$
	$\mu_{BC}(x) = \begin{cases} 0,2x - 4, & \text{se } 20 \leq x \leq 25 \\ 1, & \text{se } 25 < x \leq 30 \end{cases}$
	$\mu_{BC}(x) = \begin{cases} 0,2x - 4, & \text{se } 20 \leq x \leq 25 \\ 1, & \text{se } 25 < x \leq 30 \end{cases}$

Legenda: B - baixo; M - médio; A - alto; AA - aceita tudo; BB - bloqueia *background*; BI - bloqueia *background* e *interactive*; BS - bloqueia *background*, *interactive* e *streaming*; BC - bloqueia todas as chamadas.

**Tabela 4. Regras de inferência.**

Regras de inferência <sup>3</sup>					
CD	CJ	SD	SJ	ID	Saída
(B ∨ B) ∧ (B ∨ B) ∧ B					→ AA
(B ∨ B) ∧ (B ∨ B) ∧ M					→ BB
(B ∨ B) ∧ (B ∨ B) ∧ A					→ BI
(B ∨ B) ∧ (M ∨ M) ∧ B					→ BB
(B ∨ B) ∧ (M ∨ M) ∧ M					→ BI
(B ∨ B) ∧ (M ∨ M) ∧ A					→ BS
(B ∨ B) ∧ (A ∨ A) ∧ B					→ BB
(B ∨ B) ∧ (A ∨ A) ∧ M					→ BI
(B ∨ B) ∧ (A ∨ A) ∧ A					→ BS
(M ∨ A) ∧ (B ∨ B) ∧ B					→ AA
(M ∨ A) ∧ (B ∨ B) ∧ M					→ BB
(M ∨ A) ∧ (B ∨ B) ∧ A					→ BI
(M ∨ A) ∧ (M ∨ M) ∧ B					→ BB
(M ∨ A) ∧ (M ∨ M) ∧ M					→ BI
(M ∨ A) ∧ (M ∨ M) ∧ A					→ BS
(M ∨ A) ∧ (A ∨ A) ∧ B					→ BS
(M ∨ A) ∧ (A ∨ A) ∧ M					→ BS
(M ∨ A) ∧ (A ∨ A) ∧ A					→ BC
(A ∨ A) ∧ (B ∨ B) ∧ B					→ BC
(A ∨ A) ∧ (B ∨ B) ∧ M					→ BC
(A ∨ A) ∧ (B ∨ B) ∧ A					→ BC
(A ∨ A) ∧ (M ∨ M) ∧ B					→ BC
(A ∨ A) ∧ (M ∨ M) ∧ M					→ BC
(A ∨ A) ∧ (M ∨ M) ∧ A					→ BC
(A ∨ A) ∧ (A ∨ A) ∧ B					→ BC
(A ∨ A) ∧ (A ∨ A) ∧ M					→ BC
(A ∨ A) ∧ (A ∨ A) ∧ A					→ BC

<sup>1</sup> Variáveis de entrada: atraso de *conversational* (CD); jitter de *conversational* (CJ); atraso de *streaming* (SD); jitter de *streaming* (SJ); atraso de *interactive* (ID).

<sup>2</sup> Variável de saída: estado da BS.

<sup>3</sup> Regra: se (CD ou CJ) e (SD ou SJ) e ID, então Saída.

de saída, estabelecida pelo mecanismo de inferência (base de conhecimento). A literatura apresenta vários métodos de defuzzificação [Rao and Rao 1995]: o método do critério máximo, do centro de gravidade, da média dos máximos, do *singleton*, do centro de área, do mínimo dos máximos e do máximo dos máximos. Neste trabalho, será utilizado o centro de gravidade, que escolhe o valor escalar como o centro de massa da área. Isto é, o valor escalar que divide a área na metade. Esse valor escalar é a saída defuzzificada do mecanismo de LF, para as referidas entradas, correspondente ao estado da BS.

O passo 6 é implementar o mecanismo de LF, isto é, o CAC-RD *Fuzzy*, que é a representação *fuzzy* do CAC-RD. Conforme dito, foi utilizada a biblioteca *jFuzzyLogic*. Para integrar o *jFuzzyLogic* (Java) com o RRM do NS-2 (C++), foi utilizado o *Java Native*

*Interface* (JNI), versão 1.4<sup>3</sup>. A implementação do mecanismo de LF foi dividida em duas partes: uma no RRM – o CAC consulta o estado da BS pelo mecanismo de LF; outra na biblioteca *jFuzzyLogic*, com a modelagem do mecanismo de LF escrita em FCL.

O último passo do procedimento é a realização da validação do mecanismo de LF. Para se validar uma saída, os valores de entrada devem ser relacionados à saída esperada. Por exemplo, deve-se entrar com valores baixos para obter uma saída de valor baixo (AA – aceita todas as chamadas). O mesmo vale para valores médios e altos. Se, nos três casos, o mecanismo de LF tiver retornado a saída esperada, o mesmo foi validado. Caso contrário, sua modelagem deve ser revista. Para facilitar esse teste, foi desenvolvida uma ferramenta gráfica de validação. As entradas definidas pelo usuário passam pelo mecanismo de LF, gerando a saída do mecanismo de LF e os gráficos com as funções de pertinência. A tabela 5 apresenta os valores testados para as variáveis linguísticas de entrada e de saída. Por exemplo, para a linha 1, foi verificado, para os valores linguísticos das variáveis de entrada, se a variável de saída (estado) estava de acordo com as regras linguísticas. Pela regra 1, para todos rótulos “baixo” o estado de saída deve ser AA – estado 1 (estado esperado – EX – igual ao obtido – OB). Foram utilizados como casos de teste valores das variáveis linguísticas em cada rótulo (baixo, médio, alto). Para todos os casos, o estado de saída obtido (coluna Estado:OB) foi o esperado (coluna Estado:EX) pelas regras.

**Tabela 5. Validação do mecanismo de LF.**

#	CD	CJ	SD	SJ	ID	Estado		#	CD	CJ	SD	SJ	ID	Estado	
						EX	OB							EX	OB
1	100	0	0	0	0	1	1	22	0	0	0	0	0,75	1	1
2	150	0	0	0	0	1	1	23	0	0	0	0	1,5	2	2
3	250	0	0	0	0	1	1	24	0	0	0	0	3,1	3	3
4	317	0	0	0	0	2	2	25	0	0	0	0	4	3	3
5	358	0	0	0	0	3	3	26	100	0,4	0	0	0	2	2
6	0	0,4	0	0	0	2	2	27	113	0	10	0	0	2	2
7	0	1,2	0	0	0	3	3	28	113	0	0	2,5	0	2	2
8	0	2	0	0	0	3	3	29	0	0,2	10	0	0	2	2
9	0	0	1	0	0	1	1	30	0	0,2	0	2,5	0	2	2
10	0	0	1,75	0	0	1	1	31	100	0	0	0	1,5	2	2
11	0	0	3	0	0	1	1	32	0	0,4	0	0	1,5	2	2
12	0	0	5	0	0	1	1	33	325	0	0	0	1,5	3	3
13	0	0	8	0	0	1	1	34	0	0,5	0	0	1,5	3	3
14	0	0	10	0	0	1	1	35	0	0	10	0	1,5	2	2
15	0	0	0	0,5	0	1	1	36	0	0	10	0	3,1	3	3
16	0	0	0	0,75	0	1	1	37	0	0	0	2,5	1,5	2	2
17	0	0	0	1	0	1	1	38	0	0	0	2,5	3,1	3	3
18	0	0	0	1,5	0	1	1	39	0	0	5	2,5	1,5	3	3
19	0	0	0	2	0	1	1	40	0	0	5	2	3	4	4
20	0	0	0	2,5	0	1	1	41	300	0,4	9	2	3,7	5	5
21	0	0	0	0	0,5	1	1	42	300	0,4	10	2	3,6	5	5

## 5. Simulações e Análises

Para avaliar este trabalho, foram realizadas simulações com o CAC-RD e com o CAC-RDF para comparação dos módulos de limites estáticos e dinâmicos. Para avaliar apenas

<sup>3</sup>Disponível em: <http://java.sun.com/j2se/1.4.2/docs/guide/jni/>

o desempenho dos limites, o módulo de reserva de canais foi desativado tanto no CAC-RD quanto no CAC-RDF. Simulações foram conduzidas no cenário de congestionamento apresentado pela análise de desempenho (210 usuários em 1 célula durante 200s).

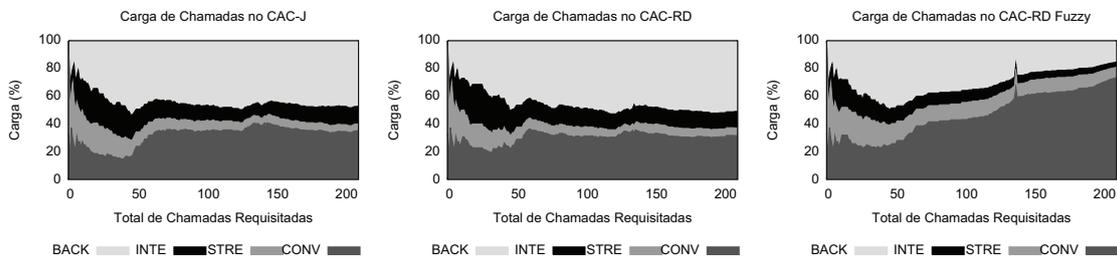


Figura 5. Comparação de Carga entre CAC-J, CAC-RD e CAC-RDF.

A primeira análise realizada foi quanto à distribuição da carga de cada classe de serviço nas BSs. A figura 5 mostra essa distribuição para o CAC-J, o CAC-RD e o CAC-RDF. Percebe-se que o CAC-RDF aceitou mais chamadas prioritárias do que o CAC-RD e o CAC-J, devido à dinamicidade nos limites. No CAC-RDF, menos chamadas *background* e *interactive* (classes menos prioritárias) e mais chamadas *streaming* e *conversational* (classes mais prioritárias) foram aceitas. Esse comportamento foi detectado nas três BSs simuladas. Além disso, foram aceitas mais chamadas no CAC-RDF, de acordo com a carga ocupada pela classe (vazão requisitada). *Conversational* foi a classe mais aceita.

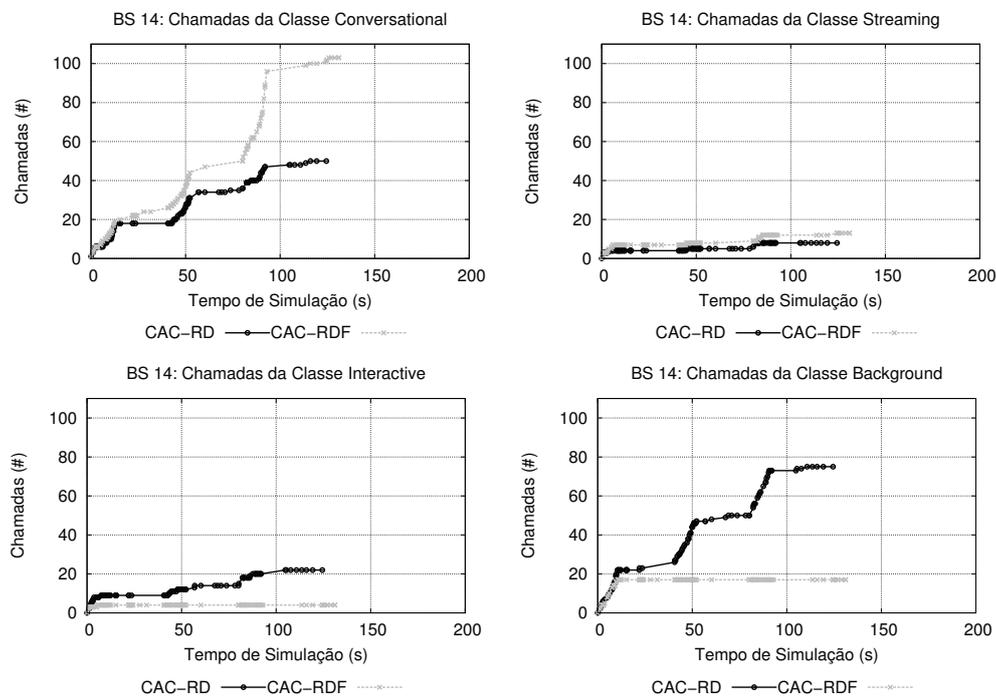


Figura 6. Cobertura (chamadas) para cada classe de serviço na BS 14.

A distribuição da carga refletiu o comportamento do CAC ao longo do tempo, conforme mostra a figura 6 (BS 14). Os gráficos mostram que, no CAC-RDF, menos chamadas das classes menos prioritárias foram aceitas para maior aceitação das prioritárias. Logo, mais chamadas de *conversational* e *streaming* foram aceitas no CAC-RDF.

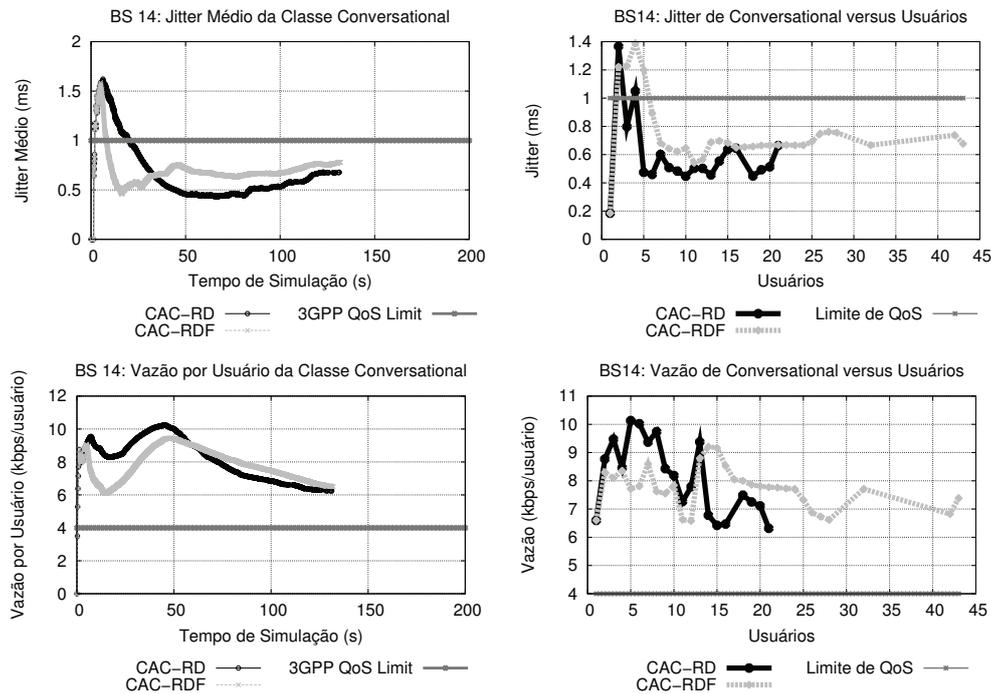


Figura 7. Parâmetros de QoS para *conversational*.

Uma comparação de níveis de QoS foi feita para o CAC-RD e o CAC-RDF. Os parâmetros explicados a seguir são os que representaram variações. A figura 7 mostra os gráficos de jitter e vazão por usuário para a classe *conversational*. Os gráficos da esquerda mostram os parâmetros de QoS pelo tempo de simulação enquanto os da direita mostram por usuários aceitos da classe. Observa-se que, a partir de 30s, os dois CACs mantiveram jitter aceitável para essa classe. Quanto à vazão, o limite do 3GPP ( $> 4$  kbps) foi respeitado em ambas as simulações. O mesmo comportamento percebido para os gráficos de QoS por tempo de simulação pode ser observado nos gráficos por usuários aceitos da classe, mas com maior aceitação de usuários da classe *conversational* no CAC-RDF.

Para *streaming*, a figura 8 mostram os gráficos de atraso e vazão por usuário, respectivamente. O atraso foi mantido para ambos os CACs próximo ao limite de 10s. A vazão por usuário permaneceu baixa, mas acima do limite de 32 Kbps. O mesmo comportamento é observado tanto pelo tempo de simulação quanto pelos usuários aceitos. Contudo, o gráfico do atraso por tempo de simulação não mostra diferença entre o CAC-RD e o CAC-RDF. Já o gráfico de atraso por usuários aceitos mostra uma redução do atraso entre 2 e 3 usuários aceitos para o CAC-RDF, enquanto com o CAC-RD o atraso aumenta e ultrapassa o limite de QoS.

Para *interactive*, os gráficos de atraso e vazão por usuário são mostrados na figura 9, por tempo de simulação (esquerda) e por usuários aceitos (direita). O mesmo comportamento dos gráficos por tempo de simulação é percebido nos gráficos por usuários aceitos. Nos gráficos por tempo de simulação, o atraso permaneceu abaixo do limite, apenas para o CAC-RDF. Isso porque o tráfego que aumentou o atraso até ultrapassar o limite de qualidade (apenas no CAC-RD). Já a vazão por usuário, quando houve tráfego, foi superior no CAC-RDF do que no CAC-RD.

Para *background*, são apresentados os gráficos da figura 10. O mesmo comporta-

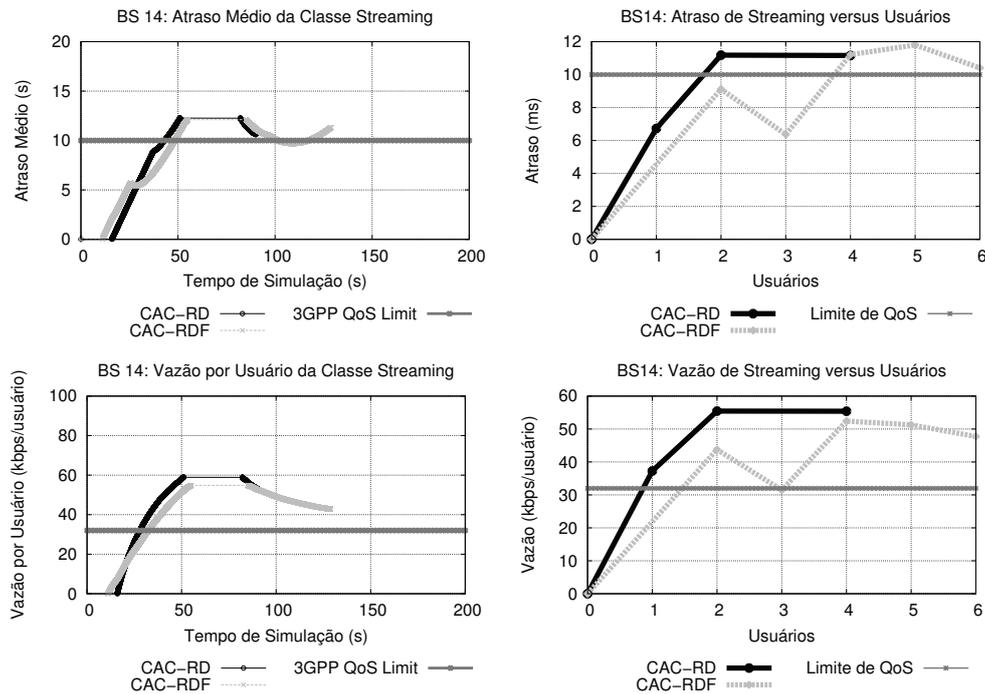


Figura 8. Parâmetros de QoS para *streaming*.

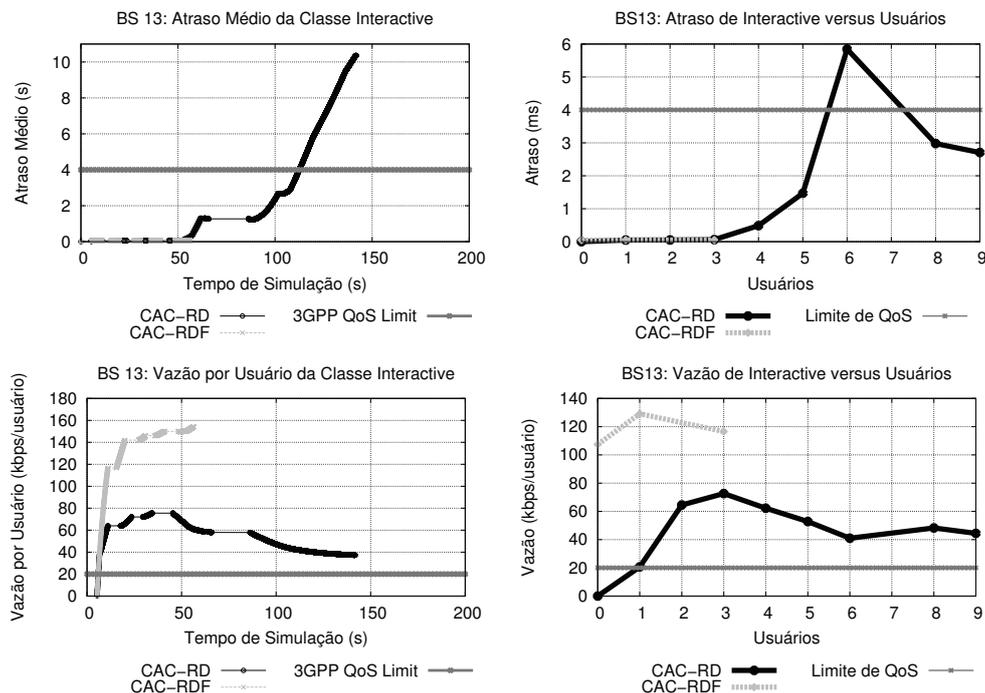


Figura 9. Parâmetros de QoS para *interactive*.

mento dos gráficos por tempo de simulação é percebido nos gráficos por usuários aceitos. Quanto ao atraso, assim como na simulação de congestionamento, a QoS foi garantida para essa classe (atraso < 30s). Em relação à vazão por usuário, o CAC-RDF apresentou maior taxa para essa classe do que o CAC-RD (exceto pelo tempo de término do tráfego).

Pelos resultados com o CAC-RDF, percebe-se que houve recursos que não foram aproveitados pelo CAC-RD e poderiam ser melhor aproveitados para as classes prioritá-

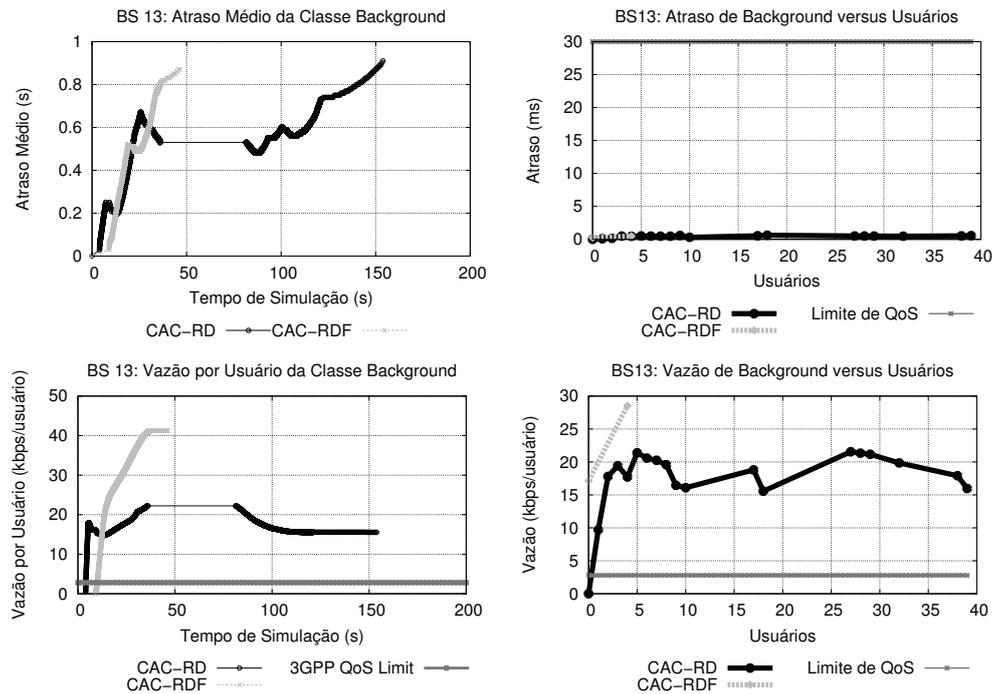


Figura 10. Parâmetros de QoS para *background*.

Tabela 6. Intervalo de 95% de confiança do CAC-RD e do CAC-RD Fuzzy.

Classe	CAC-RD		CAC-RD Fuzzy	
	Atraso/Jitter	Vazão	Atraso/Jitter	Vazão
<i>Conv.</i>	[0,38 – 0,99]	[7,23 – 9,51]	[0,53 – 0,83]	[6,69 – 8,69]
<i>Strea.</i>	[3,97 – 11,29]	[24,68 – 57,20]	[5,17 – 11,22]	[24,48 – 53,23]
<i>Inte.</i>	[0,40 – 6,94]	[37,29 – 60,05]	[0,048 – 0,052]	[92,41 – 163,82]
<i>Back.</i>	[0,38 – 0,66]	[14,90 – 19,69]	[0,28 – 0,68]	[18,10 – 38,91]

rias. Logo, o CAC-RDF foi mais eficaz e adaptativo que o CAC-RD. A análise estatística dos gráficos encontra-se na tabela 6. O intervalo de confiança dos parâmetros de QoS analisados foram calculados com 95% de confiança.

## 6. Conclusão

O CAC-RD é um controle de admissão com bloqueios estáticos de chamadas menos prioritárias, reserva de recursos e diagnóstico da rede, proposto em trabalhos anteriores. Conforme dito, o problema do CAC-RD é o caráter estático de bloqueios de chamadas, que pode causar desperdício de recursos. Em solução a esse problema, este trabalho propõe o CAC-RD Fuzzy (CAC-RDF): uma estratégia inovadora de bloqueios dinâmicos de chamadas menos prioritárias com LF. O diferencial desse CAC é a dinamização dos limites, com LF, de acordo com a QoS para bloquear as chamadas menos prioritárias.

Para analisar o CAC-RDF, simulações com os controles CAC-RDF e CAC-RD foram realizadas. Os resultados mostraram que o congestionamento devido à aceitação de chamadas foi melhor controlado pelo CAC-RDF do que pelo CAC-RD. Quando a QoS está em níveis satisfatórios para todas as classes, o CAC-RDF aceita todas as chamadas. Já o CAC-RD, bloqueia chamadas *background* após 40% de utilização, independentemente da QoS das outras chamadas. Este trabalho demonstrou que a LF contribui na estabilidade dos CACs. Em resumo, o CAC-RDF apresentou vantagens em relação ao CAC-RD:

melhor aproveitamento dos recursos da rede; priorização total de recursos; e estabilidade dos CACs. Caso a modelagem *fuzzy* não seja bem feita, o mecanismo de LF pode não funcionar conforme o desejado, produzindo um desempenho até pior do que o esperado. Por isso, é importante seguir um procedimento de representação *fuzzy*. Como trabalhos futuros, este trabalho sugere a aplicação da representação *fuzzy* em outros CACs, a simulação de novos cenários e redes (redes 4G, LTE) com o CAC-RDF e com o módulo de reserva de canais, comparação com outros CACs *fuzzy* e sua análise computacional.

## Referências

- 3GPP (2001). TS 25.211: Technical specification group services and system aspects: Services and service capabilities (release 1999).
- Antoniou, J. (2004). A system level simulator for enhanced UMTS coverage and capacity planning. Master's thesis, University of Cyprus, Department of Computer Science.
- Ascia, G., Catania, V., Ficili, G., Palazzo, S., and Panno, D. (1997). A VLSI fuzzy expert system for real-time traffic control in ATM networks. *Fuzzy Systems, IEEE Transactions on*, 5(1):20–31.
- Commission, I. E. (2009). IEC 61131-7: Fuzzy control programming released.
- Ghaderi, M. and Boutaba, R. (2006). Call admission control in mobile cellular networks: a comprehensive survey. *Wireless Comm. and Mobile Computing*, 6(1):69–93.
- Hiramatsu, A. (1989). ATM communications network control by neural network. In *International Joint Conference on Neural Networks*, volume 1, pages 259–266, USA.
- Pedrycz, W. and Vasilakos, A. (2000). *Computational Intelligence in Telecommunications Networks*. CRC Press, Inc., Boca Raton, FL, USA.
- Rao, V. B. and Rao, H. (1995). *C++, neural networks and fuzzy logic (2nd ed.)*. MIS:Press, New York, NY, USA.
- SEACORN (2004). Enhanced UMTS system level simulator. Acessado em 25.02.2008.
- Steinmetz, R. and Wolf, L. C. (1997). Quality of service: Where are we. In *Proc. 5th IWQOS'97*, pages 211–222, New York, USA. IEEE Press.
- Storck, C. R. (2007). CAC-RD: Controle de admissão de chamadas para redes UMTS. Master's thesis, PUC Minas, Mestrado em Informática, Brasil.
- Storck, C. R., Tostes, A. I. J., and Duarte-Figueiredo, F. L. P. (2008). CAC-RD: Controle de admissão de chamadas para redes UMTS. *SBRC 2008*.
- Tostes, A. I. J. (2010). Representações neural e fuzzy de controle de admissão de chamadas para redes UMTS. Master's thesis, PUC Minas, Mestrado em Informática.
- Tostes, A. I. J. and Duarte-Figueiredo, F. L. P. (2009). Simulação e análise de controles de admissão de chamadas para redes móveis de terceira geração 3G. In *CTIC/SBC*.
- Tostes, A. I. J., Duarte-Figueiredo, F. L. P., and Zárate, L. E. (2009). Controle de admissão neural para simulação de grandes cenários em redes 3G UMTS. In *WGRS/SBRC*.
- Tostes, A. I. J., Storck, C. R., and de L. P. Duarte-Figueiredo, F. (2010). CAC-RD: an UMTS call admission control. *Journal of Telecommunication Systems*.