

Uma Abordagem Semântica para Especificação de QoS de Serviços de Comunicação usando Parâmetros de QoE¹

José Cé Júnior, Achilles Prudêncio, Marcelo L. Scheibel, Roberto Willrich, Madalena P. da Silva

Departamento de Informática e Estatística (INE)

Universidade Federal de Santa Catarina (UFSC) – Florianópolis, SC – Brasil

{jcr, achilles, scheibel, willrich, madalena}@inf.ufsc.br

RESUMO

Em diversas operações relacionadas ao gerenciamento de Qualidade de Serviço (QoS) é necessário especificar os níveis de qualidade usando de parâmetros de QoS. A maioria dos trabalhos em QoS adota um conjunto fixo de parâmetros de QoS no nível de rede. Em muitas situações, onde os seres humanos são as aplicações/serviços finais, idealmente, a QoS de rede deve ser especificada usando parâmetros de Qualidade de Experiência (QoE). No entanto, a adoção de parâmetros de QoE para especificar a QoS da rede exige mecanismos eficientes no mapeamento de QoE em parâmetros de QoS de rede. Este artigo propõe um mapeamento de QoS e QoE utilizando uma abordagem de base ontológica que pode ser utilizado durante várias operações relacionadas com o gerenciamento de QoS. O uso da proposta é ilustrada usando um cenário de negociação de serviço de VoIP.

ABSTRACT

In various operations related to Quality of Service (QoS) management is necessary to specify the levels of quality through QoS parameters. Most of the works on QoS adopt a fixed set of QoS parameters at the network level. In many situations where humans are the applications/services end-users, ideally the quality of network services should be specified using Quality of Experience (QoE) parameters. However, the adoption of QoE parameters to specify the network QoS requires efficient mechanisms on QoE mapping into network QoS parameters. This paper proposes a QoE and QoS mapping using an Ontology based approach that can be used during various operations related with QoS management. The use of our proposal is illustrated to support a Voice over IP service negotiation.

Categories and Subject Descriptors

C.2.3 [Computer-Communication networks]: Network Operations – *network management*. I.2.4 [Knowledge Representation Formalisms and Methods]: *Representation languages*.

General Terms

Management, Measurement, Experimentation, Human Factors.

Keywords

Quality of Service, Quality of Experience, Ontology.

1. INTRODUÇÃO

Em qualquer oferta de serviços com QoS (Qualidade de Serviço), existem pelo menos três atores envolvidos: a Provedora de Serviços (NSP – *Network Service Provider*) que oferece o serviço; o Cliente que contrata a NSP; e o usuário, um humano ou um sistema autorizado pelo Cliente que invoca/usa o serviço. Em diversas situações, clientes e usuários necessitam especificar o nível de QoS desejado. Uma destas operações é a negociação do Acordo de Nível de Serviço (SLA - *Service Level Agreement*), que formaliza os termos e as circunstâncias do oferecimento do serviço. Um SLA contém uma lista de Especificação do Nível de Serviço (SLSs - *Service Level Specification*) que permite especificar a QoS de um determinado tráfego também identificado no SLS. Após esta negociação, os usuários dos serviços podem invocar/usar os serviços com QoS.

As soluções atuais de negociação de QoS adotam métricas de desempenho ao nível IP, como OWD (*One Way Delay*), IPDV (*IP Packet Delay Variation*), *Throughput* e PLR (*Packet Loss Rate*), além de adotarem a invocação implícita de serviços. Em muitas situações, expressar a QoS por meio de parâmetros de rede não é natural para um cliente ou usuário do serviço de rede. O entendimento da especificação de QoS é fundamental no processo de negociação/invocação do serviço. Caso contrário, os clientes/usuários podem ficar descontentes.

Diversos trabalhos demonstram que a experiência do usuário é mais importante do que qualquer mecanismo técnico utilizado no interior da rede ([1], [2]). A experiência do usuário pode ser qualificada em termos de parâmetros de QoE (*Quality of Experience*) que é a qualidade percebida de maneira subjetiva pelo usuário. Um exemplo de parâmetro de QoE é o MOS (*Mean Opinion Score*) [3], que pode ser usado para medir a qualidade de voz em aplicações de Voz sobre IP (VoIP), vídeo e TV. Assim um cliente poderia negociar a QoS para o seu serviço usando MOS em vez de parâmetros de rede.

A adoção de parâmetros de QoE durante o processo de negociação e invocação de serviços de rede exige o uso de mecanismos de mapeamento destes parâmetros de QoE em parâmetros de desempenho de rede ([4], [5] e [6]). Este mapeamento é necessário, pois os parâmetros usados pelas NSPs para configurar seus equipamentos de rede são necessariamente no nível de rede ou até mesmo enlace. Dois aspectos dificultam esta operação de mapeamento: existem vários parâmetros e métricas de QoE e QoS para cada tipo de mídia/serviço, e esta lista está em contínua expansão; e diferentes NSPs podem adotar diferentes formas de especificar QoS, de acordo com as suas tecnologias e soluções em termos de oferta de serviço.

¹ A Semantic Approach for QoS Specification of Communication Services using QoE Parameters.

Na área de serviços web, o problema de heterogeneidade de parâmetros de QoS também é um problema durante a seleção e negociação de serviços ([7]). Nos serviços web semânticos, a ontologia fornece um mecanismo para uma especificação que pode ser uma ponte para as diferentes terminologias semânticas e métricas relacionadas com a negociação de serviço. Além disso, as ontologias têm como características a inteligibilidade, interoperabilidade, transparência e extensibilidade.

Em [8] propomos a NetQoSOnt, uma ontologia descrita em OWL [9] que possibilita a interoperabilidade semântica em termos de parâmetros usados na especificação de QoS de serviços de rede. Ela permite expressar relações de equivalência entre especificações de QoS em diferentes níveis, permitindo até mesmo expressar a equivalência entre especificações de QoS no nível de rede e de QoE. Mas como será tratado mais adiante, esta forma de relacionar a QoE com a QoS usando equivalências tem um problema de granularidade.

Este artigo propõe um aprimoramento na ontologia NetQoSOnt, oferecendo recursos para o mapeamento entre especificações de QoE em especificações de QoS nos mais diversos níveis da rede. A solução proposta neste artigo é expressar o mapeamento de especificações de QoS usando regras escritas na linguagem SWRL [10]. Com isso, obtêm-se uma flexibilidade em termos de valores de QoS/QoE e uma melhora quanto à questão da granularidade. O artigo ilustra e testa a solução proposta no domínio de VoIP, onde a qualidade é especificada usando valores de MOS, e uma NSP pode usar a ontologia para inferir quais requisitos de rede são necessários para atender a QoS solicitada.

O restante deste artigo está organizado da seguinte forma. A seção 2 revisa conceitos relacionados a QoS e QoE. A seção 3 apresenta a ontologia NetQoSOnt. A seção 4 apresenta a extensão da NetQoSOnt. A seção 5 apresenta o cenário ilustrativo e de teste. Finalmente, a seção 6 apresenta as conclusões e trabalhos futuros.

2. QOS E QOE

A QoS nos serviços de redes é definida pela ITU-T [11] como o efeito combinado do desempenho do serviço que determina o grau de satisfação do usuário de um serviço. Em geral, as medições e a própria QoS são definidas em termos de parâmetros de desempenho de rede, não em termos de satisfação do usuário. Esta seção apresenta conceitos básicos na área de QoS e QoE.

2.1 QoS na Rede

O gerenciamento da QoS se baseia no SLA firmado entre a NSP e os seus clientes. Como visto anteriormente, um SLA é composto de diversos SLSs, cada um definindo os parâmetros de QoS para um determinado tipo de tráfego ou serviço. Tradicionalmente, cada SLS define o nível de QoS a ser garantido a partir de um conjunto de pares parâmetro-valor (ou faixa de valores).

Devido a falta de padronização em termos de parâmetros de QoS, as NSPs adotam diferentes parâmetros de QoS e métricas, que são dependentes do tipo de solução de QoS e até mesmo nomes comerciais de serviços oferecidos. Por exemplo, no caso da NSP adotar a solução DiffServ, para cada SLS ela deve identificar a Classe de Serviço (CoS) que atende a qualidade especificada.

Uma tentativa de padronização foi feita pelo projeto Tequila [12], que submeteu um Draft IETF propondo as linhas guia para a definição de parâmetros de SLS, incluindo OWD, PLR e PDV. Como a maior parte dos trabalhos, o projeto Tequila adotou parâmetros de qualidade em nível de rede para especificar a QoS.

2.2 QoE

Como citado anteriormente, a QoE é a qualidade percebida de maneira subjetiva pelo usuário. Ela é influenciada por todos os componentes envolvidos na comunicação fim-a-fim, incluindo o próprio usuário, o terminal, a rede e a infra-estrutura do serviço.

Diferentes aplicações ou mídias podem usar diferentes métricas de QoE. Uma destas métricas é o MOS (*Mean Opinion Score*), muito usada para expressar a qualidade em aplicações que envolvam a apresentação de áudio e vídeo, como VoIP, videoconferência e TV sobre IP. O MOS é medido por procedimentos estatísticos usando um grupo de ouvintes. Os ouvintes dão a cada amostra de voz uma classificação que varia de 1 a 5, sendo que esse valor quantifica a qualidade de voz, sendo: 1 – ruim; 2 – pobre; 3 – razoável; 4 – bom; 5 – excelente. Como a obtenção da pontuação MOS é um procedimento de difícil reprodução, na área de VoIP foram propostos alguns métodos objetivos para estimar a qualidade da voz, entre outros o *E-Model* [13] e o *PESQ* [14]. O *E-Model* tem como saída um valor único que varia de 0 a 100, chamado de *FatorR*, que é usado para estimar a percepção da qualidade da voz para o usuário. Conforme apresentado na Tabela 1, os limites de variação dos parâmetros do *FatorR* podem ser mapeados para a pontuação do método MOS e a diferentes em categorias de satisfação do usuário.

Tabela 1. Categoria de transmissão da fala [15]

FatorR	MOS	Satisfação do usuário
$90 \leq R < 100$	4,34 – 4,50	Muito satisfeitos
$80 \leq R < 90$	4,03 – 4,34	Satisfeitos
$70 \leq R < 80$	3,60 – 4,03	Alguns insatisfeitos
$60 \leq R < 70$	3,10 – 3,60	Muitos insatisfeitos
$0 \leq R < 60$	1,00 – 3,10	Quase todos insatisfeitos

Do ponto de vista do usuário, em muitos casos é mais adequado utilizar parâmetros de QoE ou de aplicação do que parâmetros ao nível da rede. Por exemplo, é mais simples o usuário especificar a qualidade de uma aplicação VoIP usando MOS do que usar parâmetros de rede. Os diversos parâmetros de QoE de VoIP medem a qualidade ao nível do usuário do serviço de VoIP. Muitos *softphones* SIP já oferecem recursos para medir e considerar a qualidade [16]. Mas em outras situações, é necessário especificar a qualidade usando parâmetros de rede ou enlace. Portanto, uma solução de QoS ideal deveria oferecer meios de especificar a QoS com transparência em termos de parâmetros.

Do ponto de vista da arquitetura de QoS, a transparência em termos de parâmetros de QoS gera a necessidade de realizar o mapeamento de parâmetros de alto nível em parâmetros que são usados para configurar os serviços de rede ou enlace. Até mesmo o mapeamento em termos de unidades e métricas utilizadas pode ser necessário.

2.3 Mapeamento de QoE em QoS

Para garantir a QoE, é necessário garantir certos limites de desempenho dos serviços de rede. Portanto, os níveis de QoE pretendidos pelos usuários devem ser mapeados em parâmetros de QoS em nível de rede.

Algumas tentativas já existiram de mapear a qualidade subjetiva ao nível de rede, como por exemplo, [4], [5], [6] e [17]. Estas propostas definem como mapear, e não como os sistemas de gerenciamento da QoS podem adquirir o conhecimento de como mapear novos parâmetros de QoS e QoE. Alguns trabalhos, como [5] e [6], propõem soluções para mapeamento usando a qualidade

percebida pelo usuário QoE para QoS, em termos de expressões matemáticas e outras. Em [4] e [18], os autores propõem, respectivamente, uma plataforma baseada em agentes para mapear QoS de rede para QoE e uma ontologia para QoE. No entanto, a ontologia proposta pelos artigos não é usada para resolver, de fato, o mapeamento entre QoS e QoE. Ela foi usada para comunicação entre agentes e o mapeamento é feito via tabelas.

Este artigo utiliza as aplicações de VoIP para ilustrar a aplicação da solução de mapeamento proposta. Em particular, utiliza-se o FatorR que pode ser mapeado para a pontuação do método MOS. A Equação 1, apresentada em [13], define a forma de mapeamento entre o FatorR e MOS.

Equação 1. Conversão de FatorR em MOS [13]

Para $R < 0$:	MOS=1
Para $0 < R < 100$:	$MOS = 1 + 0.035R + R(R-60)(100-R)7.10^{-6}$
Para $R > 100$:	MOS=4.5

Em [17], os autores apresentam os limites de desempenho dos parâmetros de QoS de rede (OWD, PLR, IPDV e vazão) que devem ser garantidos para atingir o valor máximo de MOS dos codecs de voz. Estes limites são apresentados na Tabela 2. Por exemplo, para o G.711 atingir o valor de MOS de 4,5 é necessário que a rede garanta um limite de atraso de 150 ms, uma taxa de perda igual ou inferior a 1,5%, uma variação de atraso de no máximo 40 ms e uma taxa de bits de 64 kbps.

Tabela 2. Parâmetros QoE x parâmetros de QoS [17]

QoS em nível de Usuário				QoS em nível de Rede				
PT	Codec	MOS	Qualidade Subjetiva	FatorR	Atraso (ms)	Perda (%)	Jitter (ms)	Taxa (Kbps)
0	G.711	4,5	Excelente	100	150	1,5	40	64
96 a 127	G.726	4,34	Boa	90	250	3	75	32
15	G.728	4,03	Regular	80	350	15	125	16
18	G.729	3,6	Insatisfatória	70	450	25	225	8
4	G.723	3,1	Ruim	60	600	25	225	6,4

A coluna PT (*Payload Type*) da Tabela 2 é um campo do protocolo RTP (*Real Time Protocol*) que identifica o tipo de mídia transportada no pacote IP. Este campo pode ser verificado pela aplicação receptora para determinar como tratar o dado, passando este para um decodificador adequado. Todos os codecs citados têm uma identificação predeterminada a não ser o codec G.726 que recebe uma identificação através de um valor dinâmico que varia entre 96 a 127, o que em muitos casos recebe a identificação 96 que é o padrão.

Em [19] é proposta uma função matemática, apresentada na Equação 2, que modela a relação entre MOS e os parâmetros de rede usando um conjunto de dados disponibilizados em [20]. A análise dos dados juntamente com o uso de técnicas de ajuste de curvas, derivou a função que ao receber como parâmetros a taxa de perdas de pacotes e o atraso, gera um valor de MOS, demonstrado na Equação 2.

Equação 2. Parâmetros de rede formulados à QoE [19]

$$MOS = T - \alpha p + \beta d - \eta d^2 + \varphi d^3$$

Na Equação 2, os parâmetros são: $\alpha=0.195$, $\beta=2.64 \times 10^{-3}$, $\eta=1.86 \times 10^{-5}$, $\varphi=1.22 \times 10^{-8}$, T representa o MOS do codificador sem perdas e atraso, p é a perda na rede e d representa o atraso.

3. NETQOSONT

Em [8] e [21] nós apresentamos a NetQoSOnt, uma ontologia que possibilita a interoperabilidade semântica em termos de parâmetros usados na especificação de QoS de serviços de rede.

Esta ontologia está sendo desenvolvida utilizando OWL 2.0 [9] e utiliza as características desta nova versão da linguagem OWL para a comparação de parâmetros de QoS através de inferência em ontologias.

Como apresentado na Fig. 1, as classes da NetQoSOnt são organizadas em módulos. Cada módulo engloba classes de uma das quatro camadas da pilha TCP/IP. Adicionalmente, foi acrescentado um módulo para englobar especificações de QoS em nível de usuário e outro com conceitos de base, que são reutilizados por todos os outros módulos. A noção de camadas é muito importante para modelar QoS em redes de computadores, pois os parâmetros de qualidade de uma camada podem depender (e geralmente dependem) das camadas abaixo dele.

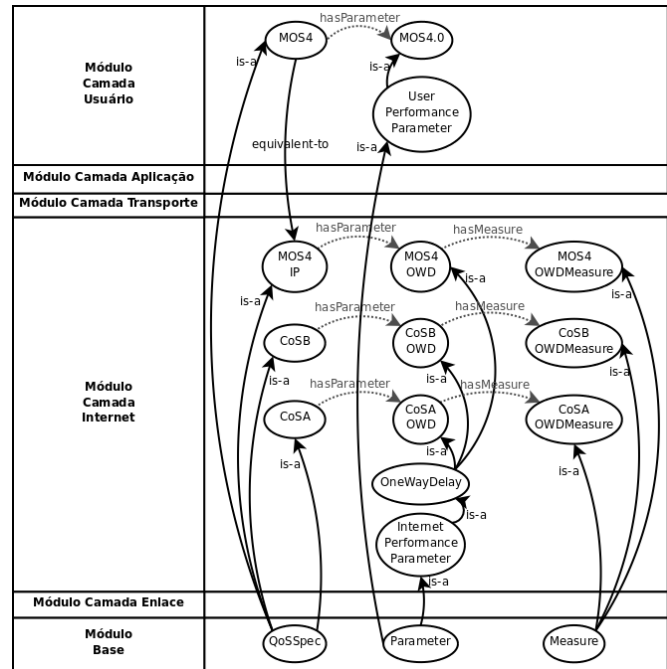


Figura 1. Módulos e alguns conceitos da NetQoSOnt

Um NSP pode utilizar a NetQoSOnt para definir e publicar uma ontologia que descreva a qualidade que oferece, com a ajuda de um especialista de domínio. Além disso, devido a organização modular de NetQoSOnt, a criação de uma interface de edição de ontologias especializadas seria trivial.

3.1 Módulos da NetQoSOnt

O módulo Base é constituído de vários conceitos, dos quais os mais importantes são mostrados na Fig. 1. O conceito *QoS Spec* define a especificação de QoS e pode ser usado para descrever uma CoS de um provedor na Camada de Internet ou os requisitos de um usuário na Camada do Usuário. *Parameter* define os parâmetros de qualidade que descrevem uma especificação, enquanto *Measure* é usado para definir o valor (quantitativo ou qualitativo) que cada parâmetro oferece.

O módulo da Camada de Enlace descreve parâmetros de desempenho e outros conceitos, envolvendo especificamente a camada de enlace.

No módulo da Camada Internet são descritos parâmetros de qualidade e outros conceitos envolvendo especificamente a camada Internet, como *InternetPerformanceParameter*, usado para descrever parâmetros de desempenho da camada Internet.

No módulo da Camada de Transporte são descritos parâmetros de desempenho e outras classes representando conceitos da camada de transporte.

O módulo da Camada de Aplicação e o Módulo de Usuário possuem conceitos de QoS no nível de aplicação e de usuário (QoE). Reutilizando estes conceitos, desenvolvedores de aplicações e organizações podem definir novos conceitos para expressar qualidade nestes níveis. Estas classes podem especificar QoS quantitativa ou qualitativamente e geralmente expressam qualidade fim-a-fim.

3.2 Equivalência entre QoS e Inferência

A Fig. 1 ilustra como NetQoSOnt especifica equivalência entre especificações de QoS. Este exemplo considera que uma organização de normalização em VoIP define faixas de valores de MOS e a QoS que deve ser atendida no nível de rede para garantir a qualidade de voz. Na Fig. 1 são representadas apenas a qualidade $MOS \geq 4$ e o parâmetro de rede atraso (OWD), para facilitar a visualização. Embora não mostradas, outras faixas de MOS e parâmetros de QoS (variação de atraso e taxa de perda de pacotes) foram definidos na ontologia deste exemplo.

Usando NetQoSOnt, a qualidade de voz de $MOS \geq 4$ é especificada pela classe *QoSSpec MOS4* na camada usuário. A classe *QoSSpec MOS4IP* na camada de Internet especifica o nível de QoS que a rede deveria garantir para permitir um $MOS \geq 4$. A Fig. 1 apresenta apenas o parâmetro de atraso *MOS4OWD*. A medida deste atraso é definida em *MOS4OWDMeasure* como sendo $< 100ms$. Para definir a equivalência de QoS entre *MOS4* no nível usuário e *MOS4IP* no nível rede, é usada a relação de *equivalência de conceitos* em ontologias (*equivalent-to*).

Para ilustrar a aplicação de NetQoSOnt, suponha que um usuário solicite $MOS \geq 3.9$ para uma NSP que tenha uma solução de QoS cujo tráfego possa ser mapeado em 3 classes de serviço: *CoSA*, *CoSB* e *BE*. Para utilizar NetQoSOnt, esta NSP deve especificar a qualidade oferecida por estas classes no domínio da provedora. A Fig. 1 ilustra a especificação das QoS oferecidas pelas classes *CoSA* e *CoSB*. Esta figura apresenta unicamente o parâmetro atraso (OWD), considerando-se que a medida deste atraso de *CoSA* é definida em *CoSAOWDMeasure* como sendo $< 50ms$, e analogamente, *CoSB* possui um atraso *CoSBOWD* como sendo $< 200ms$.

Como na ontologia não existe ainda definida uma *QoSSpec* $MOS \geq 3.9$, o sistema proposto em [8] realiza a criação de uma classe $MOS \geq 3.9$. NetQoSOnt modela a comparação de parâmetros como um problema de herança de classes da ontologia, do modo descrito a seguir.

Usando um motor de inferência, as várias subclasses de *QoSSpec* são classificadas em uma hierarquia. Este motor de inferência analisa os intervalos definidos pelas subclasses de *Measure* e como $50 < 100 < 200$, infere que *CoSAOWDMeasure* é subclasse de *MOS4OWDMeasure*, que por sua vez é subclasse de *CoSBOWDMeasure*. Por consequência, infere-se que *CoSAOWD* é subclasse de *MOS4OWD*, que por sua vez é subclasse de *CoSBOWD*. Finalmente, o processo de inferência define que *CoSA* é subclasse de *MOS4IP*, que por sua vez é subclasse de *CoSB*. Desse modo, *CoSA* é inferido como sendo um conceito mais restrito que *MOS4IP* e seu equivalente *MOS4*, o que significa que a oferta de *CoSA* atende aos requisitos de *MOS4* ($MOS \geq 4$). De modo análogo, verifica-se que *MOS4* é subclasse de $MOS \geq 3.9$ (qualidade maior), ou seja, se *MOS4* é atendido,

$MOS \geq 3.9$ também é atendido. Assim, consultando o motor de inferência e descobrindo quais das suas especializações de *QoSSpec* são subclasses das especializações de *QoSSpec* do pedido do cliente, a NSP consegue descobrir que *CoSA* atende à solicitação do usuário.

3.3 Mapeamento de QoE em QoS

A ontologia NetQoSOnt não define regras de mapeamento, e sim relações de equivalência entre especificações de QoS. Isto gera algumas limitações que serão explicadas usando o exemplo descrito em [8]. A primeira limitação é quanto à expressão de mapeamentos entre especificações de QoS para QoE de diferentes níveis. Um exemplo é a relação entre MOS, atraso e perdas definida na Equação 2 (seção 2.3). Atualmente, a NetQoSOnt permite especificar certas faixas de MOS, e os valores absolutos de atraso e perdas de pacotes que a rede deve garantir. Mas outras combinações de valores de atraso e perdas de pacotes também poderiam garantir o mesmo MOS.

A segunda limitação é quanto ao aspecto de otimização do uso de recursos de rede. Como se pode perceber, a especificação do usuário ($MOS \geq 3.9$) foi de certo modo classificada em uma das *QoSSpec* já definidas na ontologia ($MOS \geq 4$), o que pode resultar na necessidade de se garantir QoS de rede acima do pretendido pelo usuário.

Através do exemplo é evidenciado a falta de granularidade na NetQoSOnt. Percebe-se que a organização de normalização, previamente publica uma solução de QoS com valor definido. Antes mesmo de o usuário solicitar um pedido já se encontra na ontologia uma resposta à solicitação. Se o usuário pede um parâmetro de QoE *MOS* com valor ≥ 3.9 , deveria ser criada uma especificação exatamente com esse valor na ontologia da organização. Nota-se então que as solicitações dos usuários são atendidas por CoSs do provedor que deveriam/poderiam atender a outras especificações, o que se entende por falta de granularidade, por conta da organização.

Percebe-se claramente que a proposta da NetQoSOnt de trabalhar com mapeamento na forma de equivalências estáticas não é tão flexível, e pode ser melhorada com o mapeamento dinâmico dos parâmetros.

4. ESTENDENDO NETQOSONT COM REGRAS SWRL

Motivado pelas limitações da NetQoSOnt apontadas anteriormente, nesta seção, a arquitetura da ontologia NetQoSOnt é estendida para permitir o mapeamento dinâmico, em tempo de execução, de especificações de QoS/QoE. A especificação deste mapeamento é feita através de regras SWRL armazenadas em indivíduos na estrutura da ontologia estendida.

A ampliação da ontologia NetQoSOnt apresentada neste artigo resultou em novos conceitos e indivíduos necessários para descrição formalizada do mapeamento de especificações de QoS, incluindo um novo módulo na ontologia, chamado de *Mapeamento de QoS/QoE*. Estes elementos estão ilustrados na Fig. 2, onde, a título ilustrativo, são apresentados apenas conceitos e indivíduos necessários ao mapeamento de QoE/QoS em aplicações VoIP. Por simplificação, esta figura não apresenta alguns dos módulos da NetQoSOnt.

4.1 Novos Conceitos e indivíduos

Em [8] e [21], todas as especificações e requisitos eram classes especializadas de *QoSSpec*. A interpretação é diferente na presente proposta, onde as especificações de QoS da NSP são representadas por indivíduos e os requisitos de QoS dos usuários são classes. Essa variação foi necessária, pois na SWRL padrão não é possível aplicar regras a *classes*, somente a indivíduos.

A Fig. 2 apresenta dois conceitos no módulo camada de aplicação, que são *ApplicationParameter* e *Codec*. *ApplicationParameter* é uma especialização do conceito *Parameter*, e *Codec* é uma especialização de *ApplicationParameter* que é utilizado para especificar os codecs selecionados pelo usuário em uma aplicação VoIP, por exemplo. Os diversos codecs são representados por indivíduos de *Codec*. Nesses indivíduos se encontram os valores do campo *PayloadType* (PT). Os valores são definidos na classe padrão, *Measure*, usando a propriedade de objeto *hasMeasure* e a propriedade de dado *qualityLiteralvalue*. A Tabela 2 apresenta alguns exemplos de valores de PT para os *codecs* mais utilizados em aplicações de VoIP.

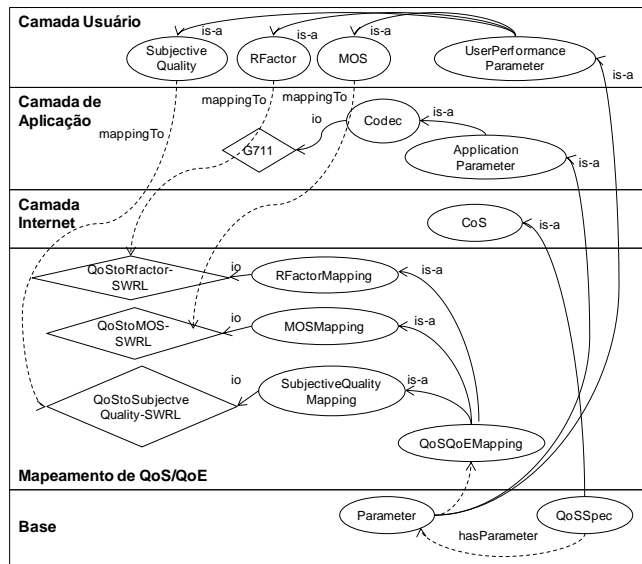


Figura 2. Ontologia de Especificação de QoE

Na NetQoSOnt, o módulo usuário representa recursos relacionados à QoE. Em especial, esta camada define *UserPerformanceParameter*, uma especialização do conceito *Parameter*. A Fig. 2 apresenta exemplos de especializações de *UserPerformanceParameter* na área de VoIP: *MOS*, *RFactor* e *SubjectiveQuality*. O conceito *MOS* permite que o usuário possa especificar a sua QoE através do método subjetivo MOS. O conceito *RFactor* permite que o usuário possa especificar a sua QoE através do método subjetivo FatorR. O conceito *SubjectiveQuality* permite que o usuário possa especificar a sua QoE através do método subjetivo excelente, bom, regular, insatisfatório ou ruim.

Por exemplo, na Fig. 2, as classes de serviço de uma NSP, chamadas CoSA e CoSB, são representados como indivíduos da especialização de *QoSspec CoS*.

Na camada Internet foi definida uma especialização de *QoSSpec* chamada *CoS*. Uma NSP pode representar suas CoS (classes de serviço) criando indivíduos da *QoSSpec CoS*. Em cada um destes indivíduos são definidas as garantias de QoS em termos de

parâmetros da camada de Internet, como atraso, variação de atraso e taxa de perdas.

4.2 Mapeamento de QoS/QoE

Na NetQoSOnt original, as relações de equivalência entre especificações de QoS são criadas de forma manual e estáticas, o que torna pouco flexível o uso de equivalência de conceitos em ontologias. No sentido de se ter uma flexibilidade quanto ao mapeamento dos parâmetros de QoS propõe-se aqui um novo módulo na ontologia NetQoSOnt, chamado módulo de Mapeamento de QoS/QoE, que define formalmente as regras de mapeamento entre QoS/QoE. Estas regras são especificadas usando a linguagem SWRL.

Um novo conceito chamado de *QoSQoEMapping* é usado para explicitar o mapeamento de especificações de QoS/QoE. Este conceito está relacionado com o conceito *Parameter* através de uma propriedade de um objeto chamada *mappingTo*. Essa propriedade relaciona um indivíduo que contém regras de mapeamento com uma subclasse de *Parameter*. Para declarar o mapeamento das especificações de QoS são definidas especializações da classe *QoSQoEMapping*.

A Fig. 2 apresenta três exemplos de especializações de *QoSQoEMapping* definindo regras de mapeamento de especificações de QoS em QoE na área de VoIP: *RFactorMapping*, *MOSMapping* e *SubjectiveQualityMapping*. Também é definido um indivíduo de cada especialização de *QoSQoEMapping*. Estes indivíduos mantêm as regras de mapeamento entre especificações QoS/QoE na sintaxe SWRL. Por exemplo, para os conceitos *MOSMapping*, *RFactorMapping* e *SubjectiveQualityMapping*, tem-se respectivamente, os seguintes indivíduos: *QoSStoMOS-SWRL*, *QoSStoRfactor-SWRL* e *QoSStoSubjectiveQuality-SWRL*. As especializações de *QoSQoEMapping* são criadas não apenas para organizar os indivíduos que contém as regras de mapeamento, mas principalmente para prover flexibilidade na inserção de novas regras de mapeamento, conforme descrito a seguir.

A Fig. 3 apresenta parte das regras de mapeamento inseridas no indivíduo *QoSStoMOS-SWRL*. Elas permitem determinar um valor de MOS a partir dos parâmetros atraso e perda de pacotes, ou seja, elas são uma representação SWRL da Equação 2. Para esta medida, deve-se considerar o codec utilizado. Por uma questão de simplificação, a Fig. 3 apresenta apenas a regra de mapeamento para o codec G.711 (PT igual a 0). Percebe-se que as regras são genéricas para os parâmetros de atraso e perda definidos na NSP.

As regras de mapeamento exemplificadas na Fig. 3 são definidas com *builtins* (relações). Os métodos apresentam argumentos genéricos, como por exemplo “?x”. Por exemplo, “?classesProvedor” representam os indivíduos *CoS* do provedor e *T* o MOS máximo do codec (4.5 para o G.711), como apresentado na Tabela 2. Os argumentos passam valores aos métodos de soma (*swrlb:add*), divisão (*swrlb:divide*), multiplicação (*swrlb:multiply*) e potenciação (*swrlb:pow*). Existem ainda argumentos que representam variáveis do tipo “?valorx”, onde x é um número indicando que a função foi separada em variáveis. Por exemplo, a variável “valor1” recebe a multiplicação do argumento “a” pelo argumento “b”. Para representar a Equação 2 em SWRL foi necessário interpretar a Equação 2 usando variáveis auxiliares (valor1, valor2, etc) que recebem valores de operações executadas pelos métodos e argumentos.

As regras de mapeamento existentes nos indivíduos *QoSFactor-SWRL* e *QoSSubjectiveQuality-SWRL*, usam a Equação 1 e a Tabela 2 para mapear parâmetros de QoE como FatorR e Qualidade Percebida para parâmetros de MOS.

```

Codec(?codec) ^hasMeasure(?PAYLOADCodec,?codec)
^swrlb:equal(?PAYLOADCodec,0.0)->swrlb:add(?T,4.5)
CoS(?classesProvedor)
^hasParameterOWD(?parametroIndividuoAtraso,?classesProvedor)
^hasMeasure(?valorIndividuoAtraso,?parametroIndividuoAtraso)
^hasParameterPLR(?parametroIndividuoPerda,?classesProvedor)
^hasMeasure(?valorIndividuoPerda,?parametroIndividuoPerda)
^swrlb:add(?a,0.195)^swrlb:divide(?b,?valorIndividuoAtraso
100)
->swrlb:multiply(?valor1,?a,?b)
swrlb:add(?c,?T)->swrlb:subtract(?valor2,?c,?valor1)
swrlb:pow(?d,10.0,-3)^swrlb:multiply(?e,2.64,?d)
^swrlb:add(?f,?valorIndividuoPerda,0.0)
->swrlb:multiply(?valor3,?e,?f)
swrlb:add(?g,?valorIndividuoPerda,0.0)
->swrlb:multiply(?valor4,?valorIndividuoPerda,?g)
swrlb:pow(?h,10.0,-5)^swrlb:multiply(?i,1.86,?h)
->swrlb:multiply(?valor5,?i,?valor4)
swrlb:add(?j,?valorIndividuoPerda,0.0)
->swrlb:multiply(?valor6,?j,?j,?j)
swrlb:pow(?k,10.0,-8)^swrlb:multiply(?l,1.22,?k)
->swrlb:multiply(?valor7,?l,?valor6)
swrlb:add(?valor8,?valor2,?valor3)
^swrlb:add(?valor9,?valor5,0.0)
->swrlb:subtract(?valor10,?valor8,?valor9)
swrlb:add(?valor11,?valor10,0.0)
->swrlb:add(?valorFinal,?valor11,?valor7);

```

Figura 3. Regra de Mapeamento em MOS-Mapping

5. CENÁRIO ILUSTRATIVO E DE TESTE

Esta seção apresenta um cenário de uso da ontologia proposta servindo também como cenário de teste. Este cenário ilustra a aplicação da proposta durante o processo de negociação de SLA. No cenário ilustrado na Fig. 4, o cliente inicia a negociação de um SLS para um tráfego VoIP entre a matriz e a filial de sua corporação. O cliente neste caso informa o codec utilizado nas chamadas, G.711, e que ele deseja uma qualidade MOS ≥ 4.34 para estas chamadas.

Para simular o comportamento do provedor e do cliente foi construído um protótipo usando os recursos da linguagem de programação Java, da API OWL 1.0 e 2.0 [22]. Para analisar as inconsistências e realizar a inferência com as ontologias foi usado o motor de inferência *Pellet* [23].

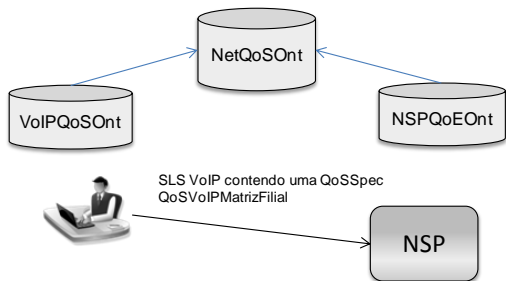


Figura 4. Cenário Ilustrativo

5.1 Ontologias utilizadas

Para este cenário ilustrativo foram definidas duas ontologias, a *VoIPQoEont*, como apresentada na Fig. 2 (com exceção do conceito *CoS*) e a ontologia *NSPQoSont*, que especifica a QoS oferecida pelas *CoSs* da NSP (Fig. 5). As ontologias foram construídas usando a ferramenta *Protégé 4.0* [24].

A ontologia *VoIPQoEont* é usada para especificar os parâmetros de QoE da área de VoIP e regras de mapeamento apresentadas na seção 4.2. Considera-se que *VoIPQoEont* tenha sido definida e publicada por uma organização de padronização de conceitos de

VoIP. No caso, esta suposta organização publicou a ontologia em <http://organizacaoVoIP.org/VoIPQoEont.owl>.

A ontologia *NSPQoSont* representa os conceitos de QoS definidos pela NSP. Esta ontologia é representada na Fig. 5. Por uma questão de simplificação, esta figura apresenta apenas a métrica de desempenho atraso, embora outras métricas de desempenho tais como *jitter* e perda de pacotes foram consideradas. Essa ontologia define três indivíduos *CoS*: *CoSA*, *CoSB* e *CoSBE*. A *CoSA* possui um parâmetro *CoSAPLR* que representa a perda que essa classe oferece. A medida dessa perda é definida em *CoSAPLRMeasure* como sendo $<1\%$. A *CoSA* ainda possui um parâmetro *CoSAOWD* que representa o atraso que essa classe oferece. A medida desse atraso é definida em *CoSAOWDMeasure* como sendo $<140\text{ms}$. A *CoSB* possui um parâmetro *CoSBPLR* que representa a perda que essa classe oferece. A medida dessa perda é definida em *CoSBPLRMeasure* como sendo $<3\%$. A *CoSB* ainda possui um parâmetro *CoSBOWD* que representa o atraso que essa classe oferece. A medida desse atraso é definida em *CoS-BOWDMeasure* como sendo $<250\text{ms}$. A *CoSBE* possui um parâmetro *BEPLR* que representa a perda que essa classe oferece. A medida dessa perda é definida em *BEPLRMeasure* como sendo $<25\%$. A *CoSBE* ainda possui um parâmetro *BEOWD* que representa o atraso que essa classe oferece. A medida desse atraso é definida em *BEOWDMeasure* como sendo $<1000\text{ms}$.

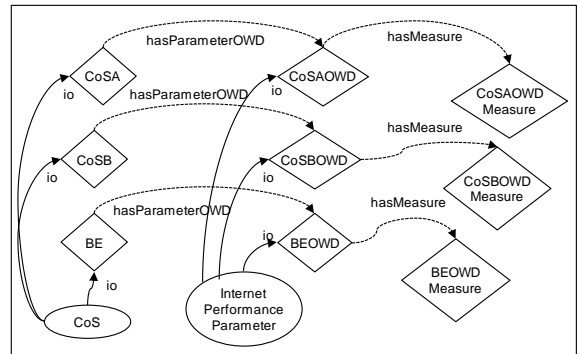


Figura 5. Ontologia NSPQoSont

5.2 Especificando a qualidade desejada

A Fig. 6 apresenta as classes e indivíduos utilizados para representar a QoS negociada no SLS. Este SLS em negociação deve conter uma *QoSSpec*, chamada aqui de *Client1*. Ela tem associados dois parâmetros, o codec G.711 (indivíduo definido na *VoIPQoEont*) considerado para as chamadas VoIP e o parâmetro de QoS. Este último é definido como uma especialização da classe <http://organizacaoVoIP.org/VoIPQoEont.owl#MOS>, chamado *Client1MOS*, que tem uma medida agregada, *Client1FaixaMOS*. Essa medida tem um valor, *qualityLiteralValue double* ≥ 4.34 . Este valor foi declarado no indivíduo *MOSG711ClientMeasure*.

Um módulo do protótipo foi criado para simular a especificação da *QoSSpec* contida no SLS VoIP. A interface do aplicativo permite que o usuário possa expressar a qualidade do serviço desejado através de MOS, FatorR ou Qualidade Percebida. Após a escolha do parâmetro de QoE é feita a inserção dos dados na ontologia da organização pela NSP, ou seja, na ontologia de especificação de QoE, aqui denominada de *VoIPQoEont*. Esses dados são gerados dinamicamente pelo módulo da aplicação.

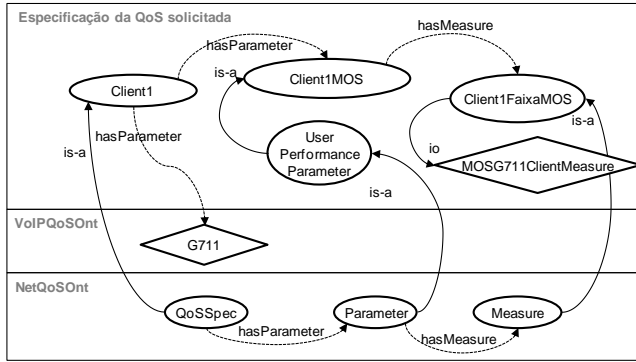


Figura 6. Especificação de QoS Solicitada

5.3 Tratamento da Solicitação do Usuário

Quando da negociação do SLS VoIP, o sistema de negociação de SLA/SLS da NSP deve aceitar/renegociar ou recusar o SLS solicitado. Isto envolve a verificação da disponibilidade de recursos para atender ao pedido. Dentre estas informações, é necessário que a NSP verifique se ela possui uma CoS que atenda a solicitação do usuário. Para isso, ela deve verificar se alguma CoS permite atender a qualidade solicitada pelo usuário $MOS \geq 4.34$.

Considere aqui que a NSP não conheça ainda o conceito de MOS. A proposta deste artigo permite que a NSP consiga importar este conceito e avaliar se uma CoS oferece a qualidade necessária. O procedimento inicia pela importação dos conceitos usados no SLS para a ontologia NSPQoSOnt.

Outro módulo de software foi desenvolvido para simular o sistema de negociação de SLS. Ao receber a requisição do cliente, este módulo deve verificar se a especificação da QoE já é de seu conhecimento. Caso afirmativo, ela já sabe qual CoS atende a solicitação do usuário. Se a NSP ainda não conhece a especificação de QoE, ele usa a URL da organização VoIP (<http://organizacaoVoIP.org/VoIPQoEOnt.owl>) como referência para a importação da ontologia VoIPQoEOnt.

Conhecida a especificação de QoS em negociação, é necessário agora que a NSP compare esta especificação com o nível de qualidade oferecidos pelas suas CoSs. Ou seja, comparar uma especificação de QoS ao nível do usuário com uma especificação de QoS ao nível de rede. Para permitir esta comparação, é necessário mapear as especificações de QoS das CoSs em especificações de QoE usando os mesmos parâmetros de QoE usados pelo cliente. A solução adotada para tal é gerar dinamicamente indivíduos *QoSspec* com parâmetro *MOS* equivalentes a qualidade especificada pela *QoSSpec* que define os parâmetros de desempenho garantidas pelas CoSs. Além dos indivíduos *QoSSpec* são necessários gerar indivíduos *MOS* e *Measure* para expressar o parâmetro da *QoSSpec* da especificação e o seu valor.

Para determinar o MOS mínimo garantido pelas CoSs, o sistema de negociação de SLA deve interpretar as regras de mapeamento da ontologia mantidas no indivíduo *QoSToMOS-SWRL* identificado pela relação *mappingTo* (Fig. 2). Essa relação é responsável pela ligação da solicitação de MOS, especificada na camada de usuário com o indivíduo *QoSToMOS-SWRL*, da classe *MOSMapping*.

Pelo fato da linguagem SWRL não permitir que os valores resultantes das regras possam ser inseridos em ontologias

diretamente como indivíduos, precisa-se de funcionalidades na aplicação para prover esse recurso. Para tal, o módulo do protótipo utiliza um interpretador SWRL, implementado em java, para a análise das regras mantidas no indivíduo *QoSToMOS-SWRL* e a instanciação dos indivíduos. As variáveis que armazenam os resultados das regras têm os valores de MOS resultantes das *CoSA*, *CoSB* e *CoSBE*. A aplicação Java com os recursos da OWL API cria e insere os indivíduos de *QoSSpec* especificando valores para MOS resultantes na ontologia do provedor.

De acordo com os parâmetros de rede publicados para cada CoS da NSP (atraso, perda e jitter), e o conhecimento do tipo de *codec* da especificação, o módulo Java interpreta as regras de mapeamento existente no indivíduo *QoSToMOS-SWRL* através da relação explicada e gera os novos indivíduos. A Fig. 8 apresenta os indivíduos gerados a partir da análise das regras. Os indivíduos *QoSSpec*, *QoSSpecMOSG711CoSA*, *QoSSpecMOSG711CoSB* e *QoSSpecMOSG711BE* representam especificações de QoS usando o parâmetro MOS e considerando o codec G.711 de qualidade equivalente as *QoSSpec* *CoSA*, *CoSB* e *BE*. Ou seja, os níveis de qualidade são equivalentes, apenas expressas usando parâmetros de diferentes camadas de rede. Os indivíduos *MOSG711CoSAMeasure* ($double=4.54$), *MOSG711CoSBMeasure* ($double=4.18$) e *MOSG711CoSABeasure* ($double=0.69$) mantêm os valores máximos de MOS garantidas pelas CoS.

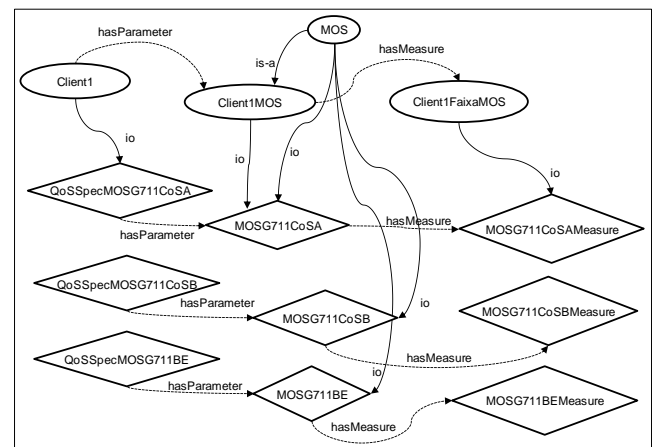


Figura 7. Indivíduos gerados e resultado da inferência

Depois de gerados estes indivíduos, o módulo de negociação de SLS deve usar um motor de inferência para identificar quais das suas CoSs atendem a especificação do usuário. Sendo a *QoSSpec* *Cliente1* a formalização da qualidade desejada pelo cliente, para verificar qual CoS atende a qualidade basta identificar qual dos indivíduos *QoSSpec* gerados é membro da classe *Cliente1*.

Para classificar os membros em classes, o motor de inferência analisa os intervalos definidos pelas subclasses e valores dos indivíduos de *Measure*. Como a classe *Client1FaixaMOS* tem um valor, $double \geq 4.34$, e o indivíduo *MOSG711CoSAMeasure* tem um $double \geq 4.54$, então este indivíduo é inferido como da classe *Client1FaixaMOS*. Assim, *MOSG711CoSA* é indivíduo da classe *Client1MOS* e por consequência *QoSSpecMOSG711CoSA* é indivíduo da classe *Client1*. O que se conclui que CoSA tem limites de desempenho que atendem a solicitação do usuário. Note que esta constatação é automática, obtida pelo protótipo desenvolvido que utiliza o motor de inferência Pellet. Esta operação automática permitiu verificar a aplicabilidade da

ontologia proposta nas operações relacionadas à comparação de especificações de QoS.

Aqui se percebe a flexibilidade obtida pela ontologia proposta. O provedor não conhecendo a especificação de QoE, feita pelo usuário, consegue gerar dinamicamente instâncias na camada de usuário da sua ontologia. Essas instâncias só foram especificadas devido às regras de mapeamento implementadas em indivíduos na organização VoIP.

6. CONCLUSÕES

Este artigo apresentou o aprimoramento na ontologia NetQoSOnt, oferecendo recursos para o mapeamento entre especificações de QoE em especificações de QoS nos mais diversos níveis da rede. Com isso se conseguiu obter uma flexibilidade em termos de valores de QoS/QoE e uma melhora quanto à questão da granularidade.

Com os experimentos realizados, pode-se testar a solução proposta, onde a qualidade pode ser especificada usando valores de MOS, e uma NSP pode usar a ontologia para inferir quais requisitos de rede são necessários para atender a QoS solicitada.

O processo de mapeamento, usando regras na linguagem SWRL, traz flexibilidade aos provedores de serviço e às organizações de normalização de serviços. Flexível, pois novos parâmetros de QoE podem ser definidos e a NSP pode adquirir este conhecimento via a importação desta nova ontologia.

Como trabalho futuro, pretende-se popular a ontologia proposta com novos parâmetros de QoE e a definição de novas regras de mapeamento usando linguagem SWRL. Também se pretende realizar testes em ambientes reais de negociação de SLA/SLS usando tecnologias Web.

7. REFERÊNCIAS

- [1] Kilki, K. 2008. Quality of Experience in Communications Ecosystem. *Journal of Universal Computer Science* 14, 5, 615-624.
- [2] Moorse, A. 2001. Metrics for the Internet Age: Quality of Experience and Quality of Business. 2001. In *Proceedings of the Fifth International Workshop on Performability Modeling of Computer and Communication Systems*, 26-31.
- [3] ITU-T Recommendation P.800. 2003. Methods for subjective determination of transmission quality.
- [4] Siller M.; Woods J., 2006. Using an agent based platform to map quality of service to experience in conventional and active networks. *IEE Proceedings Communications* 153, 6, 828-840.
- [5] Ghinea, C., and Thomas, J. 1999. An approach towards mapping quality of perception to quality of service in multimedia communications. In *Proceedings of the IEEE Workshop on Multimedia Signal Processing*, 497-502.
- [6] Ghinea, G., and Thomas, J. 1999. Quality of perception to quality of service mapping using a dynamically reconfigurable communication system. In *Proceedings of the IEEE Global Telecommunications Conference* 4, 2061-2065.
- [7] Hung, P.C.K., Li, H., and Jeng J.J. 2005. WS-Negotiation: an overview of research issues. *System Sciences*, 2004. In *Proceedings of the 37th Annual Hawaii International Conference*, 33-42.
- [8] Prudêncio, A.C., Willrich, R., Tazi S., and Diaz, M. 2009. Quality of Service Specifications: A Semantic Approach. In *Proceedings of the 8th IEEE International Symposium on Network Computing and Application*, 219-226.
- [9] W3C. 2009. The overview of OWL Web Ontology Language. URL <http://www.w3.org/TR/owl-features/>.
- [10] W3C. 2004. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. URL <http://www.w3.org/Submission/SWRL/>.
- [11] ITU-T. 2002. Network performance objectives for IP-based services, Recommendation Y.1540.
- [12] Goderis, D., et al. 2003. Service Level Specification Semantics and Parameters. URL <http://tools.ietf.org/html/drafttequila-sls-02>.
- [13] ITU-T. 2003. Recommendation G.107 The E-Model, a computational model for use in transmission planning.
- [14] ITU-T. 2001. Recommendation P.862, Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs.
- [15] ITU-T. 2002. Recommendation G.113 Appendix I. Provisional planning values for the equipment impairment factor I_e and packet-loss robustness factor Bpl.
- [16] Farias, C.M et al. 2009. Combinação de pilhas de protocolo para a construção de um softphone SIP. *Anais do XIV Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)*, 178-185.
- [17] ETSI. 2000. TS 101 329-5 v1.1.1. Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 3; End-to-end Quality of Service in TIPHON systems; Part 5: Quality of Service (QoS) measurement methodologies”.
- [18] Gallo, E.; Siller, M.; and Woods, J. 2007. An Ontology for the Quality of Experience framework. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 1540-1544.
- [19] Fujimoto, K., Ata, S., and Murata, M. 2002. Adaptive playout buffer algorithm for enhancing perceived quality of streaming applications. In *Proceedings of the IEEE Global Telecommunications Conference* 1, 2463-2469.
- [20] Savolaine, C. 2001. QoS/VoIP overview. In *Proceedings of the IEEE International Workshop on Communication Quality&Reliability*.
- [21] Willrich, R., Vicente, L.H., Uriarte, R.B., Prudêncio, A.C., and Cé, Júnior J. 2009. Invocação Dinâmica de Serviços com QoS em Sessões Multimídia SIP. In *Proceedings of the 8th Int. Information and Telecommunication Technologies Symposium (I2TS)*.
- [22] OWL API. 2010. The OWL API. URL <http://owlapi.sourceforge.net/>.
- [23] Sirin, E. et al. 2007. Pellet: The Open Source OWL DL Reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5, 2, 51-52.
- [24] Protégé. 2010. The Protégé Ontology Editor and Knowledge Acquisition System. URL <http://protege.stanford.edu/>.