

Priorização de chamadas de voz em redes em malha sem fio

Clayton Reis da Silva e Célio V. N. Albuquerque

Resumo—O trabalho propõe a priorização de um serviço de VoIP sobre redes em malha sem fio (redes mesh). Redes mesh estendem o alcance de redes sem fio tradicionais através de comunicação em múltiplos saltos. VoIP tem a vantagem de ser um serviço de baixo custo, potencialmente com menor qualidade que chamadas tradicionais de voz. Existe a necessidade da investigação do impacto das características de desempenho da rede mesh, como atraso e perdas de pacote altamente variáveis nas comunicações de voz, pois estas redes não foram criadas para comunicação multimídia. Testes reais foram executados para testar ferramentas de diferenciação de serviço.

Palavras-Chave—Redes Mesh, VoIP, Controle de Tráfego, Qualidade de Serviço.

Abstract—This article proposes the analysis of a VoIP service over wireless mesh networks. Mesh networks extend the range of traditional wireless networks by multihop communication. VoIP has low cost service advantage, but may also result on lower quality than traditional voice calls. There is a need to research the impact of the mesh networks performance characteristics, such as high variability on delay and packet loss in voice communications, since such networks have not been created to support multimedia communication. Real experiments were performed to test quality of service configurations.

Keywords—Mesh Networks, VoIP, Traffic Control, Quality over Service.

I. INTRODUÇÃO

Por volta de 1970 que a idéia de VoIP (*Voice over Internet Protocol*) começou a ser discutida, porém as chamadas VoIP iniciais eram de baixa qualidade. Assim, o sistema convencional PSTN (*Public Switched Telephone Network*), mais confiável, era o mais usado.

Com a popularização massiva da Internet a partir dos anos 90, e com o investimento na infra-estrutura da rede IP, o VoIP tornou-se um serviço viável para a transmissão de voz, por causa do baixo custo das ligações, principalmente de longas distâncias.

O trabalho propõe a implementação e análise de um serviço de voz sobre IP sobre a rede em malha sem fio do Projeto ReMesh [1], situada no Campus da Praia Vermelha, onde se situa o Instituto de Computação da UFF. O serviço de VoIP sobre mesh possibilitará a comunicação com a rede telefônica através de terminais móveis. Como os serviços de rede em malha sem fio e VoIP são de baixo custo e fácil implementação, esta é uma pesquisa muito importante para, em um futuro próximo, serem implantados. Desta forma, em

uma cidade coberta por uma rede mesh, torna-se possível efetuar ligações telefônicas entre pontos na mesma rede, com um custo baixo.

No entanto, existe a necessidade de se investigar o impacto das características de desempenho da rede mesh, como perda de pacotes, vazão e atraso altamente variáveis na qualidade das comunicações de voz, visto que as redes IP não foram criadas, inicialmente, para comunicação multimídia (voz e/ou vídeo, por exemplo), o que dificulta a implementação desse tipo de serviço.

O objetivo desse trabalho é estudar as características e problemas de chamadas de voz, a fim de implementar através da ferramenta Controle de Tráfego (TC) [6], diferenciação de qualidade de serviço em ligações realizadas entre dois ou mais nós da rede, independente do estado atual da mesma.

Para atingir esse objetivo, vários cenários foram analisados, para verificar os passos de realização de uma chamada, a qualidade máxima que ela poderia chegar em um cenário real e os tipos de pacotes que trafegam na rede durante a realização da mesma. Foi gerado tráfego concorrente, de modo a simular uma rede em plena atividade, como uma rede ativa comum. E a ferramenta proposta foi implantada na rede, para gerar qualidade do serviço de voz.

O restante deste texto está organizado da seguinte forma: A Seção 2 explica o atual estado da arte, com breves explicações sobre redes mesh e voz sobre redes IP. Trata de apontar o problema atual de implementação desse serviço, e os pontos que podem ser atacados. A Seção 3 explica a solução proposta para resolver o problema, através de controle de tráfego. A Seção 4 fala dos cenários que serão propostos para os testes. A Seção 5 mostra os resultados dos testes, discussão e conclusão de cada um, além de uma análise geral dos resultados. A Seção 6 conclui o trabalho, e destaca o que pode ser feito para melhorá-lo.

II. ESTADO ATUAL

A. Redes mesh IEEE 802.11

As redes em malha sem fio (redes mesh) consistem de roteadores e clientes mesh, onde os nós são fixos e constituem o backbone da rede. Roteadores mesh podem ser configurados como *gateways*, que são os nós responsáveis pela comunicação com outras redes (por exemplo, a Internet). Na Figura 1 pode-se ver um exemplo de rede mesh, composta por clientes mesh, roteadores e dois *gateways*.

Nesse tipo de rede, os nós se comunicam por múltiplos saltos, em modo ad-hoc, sem a necessidade de um nó centralizador. Desta forma, todos os nós da rede estão em um

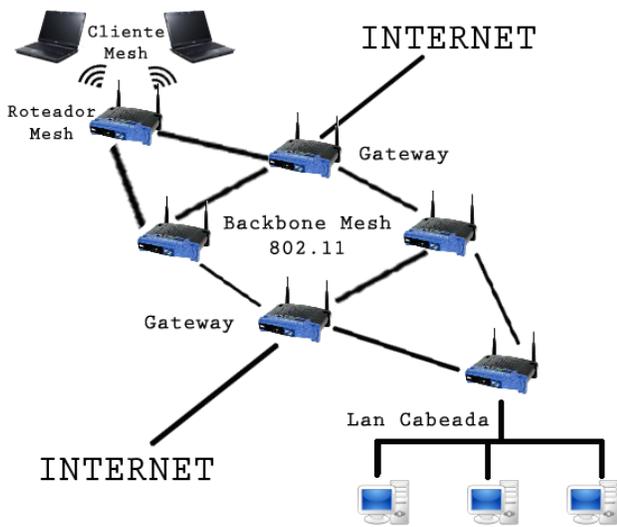


Fig. 1. Exemplo de rede mesh

mesmo nível de hierarquia. Assim, pacotes são encaminhados de nó a nó até que atinja o destino, permitindo inclusive que dois clientes se comuniquem de forma direta. Esse tipo de rede ainda tem como características importantes ser auto-configurável, ter boa tolerância a falhas e suportar mobilidade. Ou seja, a queda ou movimento de um nó não compromete a rede como um todo. Além disso, redes ad-hoc são altamente escaláveis, visto que cada novo nó da rede é potencialmente um roteador mesh, podendo contribuir com o encaminhamento e desta forma proporcionando escalabilidade.

B. Transmissão de voz sobre redes IP

VoIP (*Voice over Internet Protocol*) é um serviço de realização de chamadas sobre redes IP. Nele, a voz é transformada de sinal analógico em digital e empacotada, a fim de ser transmitida pela rede. O serviço também trata de sinalização para estabelecimento das chamadas fim-a-fim.

Os serviços VoIP têm como grande vantagem o seu baixo custo. Porém, seus problemas em relação à qualidade podem acabar inutilizando esse serviço, em detrimento da telefonia tradicional.

O primeiro fator que dificulta a implementação de um serviço de VoIP é que as redes IP comuns não oferecem garantia nenhuma de QoS (*Quality of Service* ou Qualidade de Serviço), pois são baseadas no modelo de melhor esforço (*Best effort*). Em outras palavras, significa que a rede faz seu melhor esforço para transportar cada pacote do emissor ao destino o mais rapidamente possível, porém não garante nada acerca de atraso na entrega do mesmo, ou do *jitter* (variação do atraso) de um conjunto de pacotes. Essa falta de garantia torna uma aplicação de tempo real em um serviço de baixa qualidade.

Outro fator que dificulta a implementação desse serviço é que redes IP foram projetadas originalmente para o tráfego de aplicações elásticas, ou seja, o atraso não é crucial, como transferência de arquivos e *e-mail*. Essas aplicações têm como

necessidade crítica que a mensagem seja transferida fim-a-fim, sem comprometimento com valores de atraso e *jitter*. Aplicações de tempo real, como VoIP e vídeo, suportam uma certa porcentagem de perda (diferente das aplicações de dados), mas são bastante sensíveis a atraso e *jitter*.

Mais um fator que dificulta a implementação de VoIP em qualquer tipo de rede é a concorrência de outros fluxos. Por causa de subutilização de recursos, não se pode alocar um canal (ou um caminho) só para transmissão de voz, como é o modelo padrão de telefonia convencional. A concorrência com outros tráfegos é exemplificada na Figura 2.

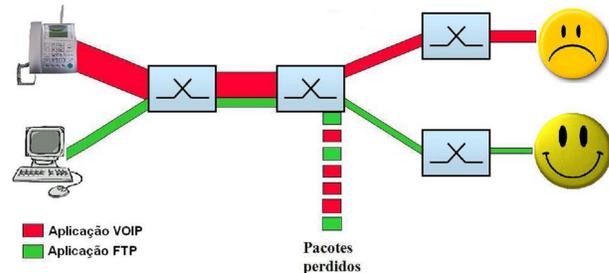


Fig. 2. Problema de concorrência de tráfego

Como se pode ver na figura, uma aplicação de telefonia tem qualidade degradada porque está compartilhando o meio com outras aplicações. No exemplo, rajadas de pacotes chegando ao buffer do roteador compartilhado faz com que o mesmo encha, e pacotes são descartados a fim de esvaziá-lo. A aplicação FTP, pelos mecanismos de recuperação de perdas e tratamento de erros no TCP, é bem sucedida. A aplicação de voz, por não ter nenhuma garantia de qualidade e recuperação de pacotes, será mal sucedida. O ponto atacado na proposta desse trabalho é priorizar aplicações de voz, de modo que não sofram tanta interferência de outros tráfegos concorrentes na rede.

Outros fatores, mais específicos às redes mesh, são problemas associados a transmissão através de múltiplos saltos, interferência de outros dispositivos sem fio e desvanecimento do sinal. Um grande número de saltos entre fonte e destino introduz um grande atraso na rede. O fator interferência é muito preocupante, pois a banda dos padrões 802.11b/g (os mais utilizados atualmente) é compartilhada com vários outros dispositivos, como por exemplo, forno microondas, dispositivos *Bluetooth* e telefones sem fio. Outras redes 802.11 podem causar interferência. Além disso, os próprios nós da rede podem causar interferência ao canal em outros nós. O fator desvanecimento ocorre devido a propriedades físicas dos sinais eletromagnéticos usados na transmissão. O capítulo 3 de [2] descreve os tipos de interferência que uma rede 802.11 está sujeita, sobre o desvanecimento que o sinal sofre com o tempo, distância ou obstáculos, e descreve como o desvanecimento e a interferência degradam a qualidade de uma transmissão.

Os fatores que são determinantes na qualidade de voz são atraso, *jitter*, e perda de pacotes.

- Atraso: é a medida de tempo entre a fala do emissor e a chegada da mesma para o receptor. Seu grande valor ocasiona falta de sincronização entre os usuários e prejudica a interatividade. É o grande problema de

estabelecimento de serviços de voz, pois como dito antes, redes IP não oferecem nenhuma garantia em relação a esse fator.

- *Jitter*: é a variação da taxa de chegada de pacotes, introduzida pela variação do atraso na rede. O seu efeito pode ser controlado com o uso de *playout buffers*[10].
- Perda de pacotes: normalmente os pacotes são perdidos nos roteadores da rede, quando seu buffer enche e eles começam a descartar pacotes. Dependendo do *codec*, perdas maiores que 2% não são recomendadas [3]. Uma porcentagem muito grande de perdas pode causar sensação de “picotamento” da voz, e grandes períodos inexistentes de silêncio.

O modo de medir a qualidade de voz entre chamadas recomendada pelo ITU-T P.800 [4] é a pontuação de opinião média ou MOS (*Mean Opinion Score*). Nele os usuários pontuam a qualidade de uma chamada telefônica em uma faixa entre 1(ruim) e 5(excelente). Os fatores que são levados em conta para o cálculo do MOS são o atraso, o *jitter*, a porcentagem de perda de pacotes e o codificador de voz (*codec*) utilizado. Valores de MOS acima de 3.6 são aceitáveis [5], de modo que a partir do momento que ficar abaixo desse limiar, o usuário poderá sentir problemas na comunicação.

III. PROPOSTA

O controle de tráfego [6] através da ferramenta *traffic control* (TC) do Linux funciona através de algoritmos que manipulam como os dados são transmitidos para as respectivas interfaces de rede. Esses algoritmos são responsáveis por enfileirar, priorizar, descartar e oferecer garantias a um determinado tráfego. A utilização de controle de tráfego permite solucionar problemas de gerência de redes limitando, por exemplo, a taxa de transmissão de pacotes.

O TC permite criar filas de disciplinas de escalonamento de pacotes (*Queueing Disciplines - qdisc*) para gerenciar a largura da banda em uma interface. Uma *qdisc* é um algoritmo utilizado para gerenciar as filas de uma interface do dispositivo. Com ela é possível alterar o tratamento dos pacotes de entrada ou saída (geralmente o segundo caso). As *qdiscs* podem ser com subdivisão de classes internas (*classful*) ou sem a subdivisão (*classless*).

Em [6] pode-se ver os vários tipos de disciplina que o TC implementa. Nesse trabalho, dois tipos de disciplina foram estudados, o mecanismo de enfileiramento HTB (*Hierarchical Token Bucket*) e o mecanismo de policiamento TBF (*Token Bucket Filter*), a primeira *classful* e a segunda *classless*.

O HTB é uma disciplina de controle de tráfego que permite divisão do tráfego em classes que podem ser priorizadas de acordo com o tipo de conexão. Ele funciona de forma hierárquica e controla os pacotes que saem de uma determinada interface de rede. O HTB ajuda a controlar o uso da banda de saída de um enlace. Ele assegura que a quantidade de serviço fornecida a cada classe é, no mínimo, a quantidade que esta pede, e no máximo o teto (*ceil*) configurado para a classe. Pelo mecanismo de *borrowing* (empréstimo), quando uma classe usa menos do que a quantidade a que tem direito, a largura de banda subutilizada é distribuída pelas outras classes.

Ele permite o uso de um enlace físico para criar vários enlaces lógicos mais lentos e por esse enlaces lógicos diferentes fazer fluir diferentes tipos de tráfego. Deve-se especificar como dividir o enlace físico em enlaces lógicos e que enlace lógico tratará que tipo de pacote de saída da interface.

O TBF é uma disciplina de enfileiramento precisa e com baixa necessidade de processamento, onde os pacotes somente saem da fila enquanto não passarem de certa taxa pré-estabelecida, com a possibilidade de ter pequenas rajadas definidas pelo usuário. A implementação do TBF consiste em um buffer (bucket) que contém uma série de “créditos”(tokens) que permitem ou não que um pacote ou um conjunto deles saiam da interface de rede. Essa cadência de tokens é o que define a vazão da fila, quanto mais tokens, maior o buffer. A principal característica do TBF é a possibilidade da policiamento e suavização do tráfego da rede, ou seja, quando o número de bytes for maior do que a quantidade de tokens disponíveis não será possível que todos os pacotes saiam da interface quando desejam, provocando atrasos e descartes de pacotes.

IV. AVALIAÇÃO DE DESEMPENHO

Para realização dos testes, foi utilizada a versão da ferramenta *callgen* da UFAM [7]. O *callgen* é um gerador automático de chamadas na rede entre dois pontos que usa para sinalização o protocolo H.323 [8] e permite escolha de *codec* e a realização de chamadas simultâneas. Ao fim das chamadas ele gera um relatório contendo informações importantes, como atraso, porcentagem de perdas e *jitter*. O *codec* utilizado para realização dos testes foi o G.723.1, na taxa de 6.3kbit/s [12], que tem um MOS máximo teórico de 3.9.

O primeiro passo para criação dos filtros foi a captura de pacotes da rede durante a realização de uma chamada de voz, por meio da ferramenta *wireshark*. Nessa captura pôde-se verificar os tipos de pacote que trafegam na rede durante a chamada, principalmente os pacotes de voz do G.723.1 (UDP, 198 bytes), pacotes de controle RTCP (*Real-Time Control Protocol - UDP*, de 138 ou 142 bytes) e pacotes de controle de rotas do protocolo OLSR [13] (*Optimized Link State Route - UDP*, porta 698) usado na rede mesh. Essas informações foram utilizadas para a realização da filtragem dos pacotes nos testes com o TC.

A rede utilizada para a realização dos testes foi a rede interna do Instituto de Computação da UFF, cuja topologia está apresentada na Figura 3. Cada ponto é um roteador sem fio configurado em modo *ad-hoc*. Cada linha é a qualidade OLSR-ML do enlace entre os pontos. ML (*Minimum Loss*) [1] é uma extensão do OLSR desenvolvida pelo projeto Remesh.

Foram realizados 4 testes. A estratégia de testes adotada foi a seguinte:

- Teste 1: Rede livre sem tráfego concorrente, chamada estabelecida entre pontos 8 e 16 da rede.
- Teste 2: Rede com tráfego TCP concorrente entre os pontos 16 e 27, sem TC configurado, com chamada estabelecida entre pontos 8 e 16 da rede.
- Teste 3: Rede com tráfego TCP concorrente entre os pontos 16 e 27, com TC configurado em todos os roteadores

com divisão de hierarquia em 2 classes, utilizando HTB. Chamada estabelecida entre pontos 8 e 16 da rede.

- Teste 4: Rede com tráfego TCP concorrente entre os pontos 16 e 27, com TC configurado em todos os roteadores com divisão de hierarquia em 3 classes, utilizando HTB e TBF. Chamada estabelecida entre pontos 8 e 16 da rede.

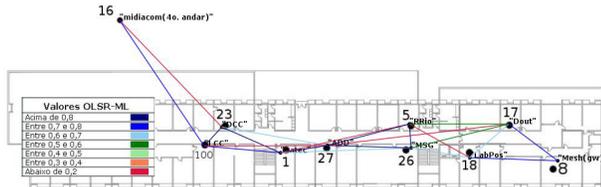


Fig. 3. Topologia da rede dos testes

Para os testes 1 e 3 foram realizadas 3 baterias de 5 chamadas não-simultâneas cada, com um total de 15 chamadas para cada teste. Para os testes 2 e 4 foram realizadas 4 baterias de 5 chamadas não-simultâneas cada, com um total de 20 chamadas para cada teste. Foi escolhido um total de 15 chamadas para que um teste com resultados discrepantes não influenciasse na média geral. Os testes foram separados em 3 baterias por causa que se deseja realizar as chamadas em um horário que haja o mínimo de interferência possível na rede, nesse caso em horário noturno. Como cada chamada tem duração de 10 minutos, não pôde ser realizada 15 chamadas consecutivas em apenas uma noite para cada teste. Foi realizada uma bateria a mais nos testes 2 e 4 porque algumas chamadas deles não conseguiram ser estabelecidas. Assim, para esses testes não terem menos de 15 chamadas, foi realizada em ambos uma bateria a mais.

O objetivos dos testes é com o Teste 1, analisar a capacidade máxima da rede, em termos de MOS, atraso, *jitter* e perda de pacotes. No Teste 2, verificar como que um tráfego concorrente pode depreciar a qualidade de uma chamada de voz, visto que temos no mínimo um link compartilhado entre os fluxos de voz e de dados (esse último gerado através da ferramenta *iperf*). Os testes 3 e 4 servem para mostrar como uma ferramenta que provê diferenciação de tráfego pode gerar qualidade e estabilidade nas chamadas. A diferença dos testes 3 e 4 está no modo de atacar o problema.

O *script* utilizando apenas HTB para o Teste 3 trata de separar os tráfego em apenas duas classes, o que será priorizado e o que não será priorizado. Na primeira classe situam-se os pacotes de voz do G.723.1, pacotes de controle RTCP e pacotes de controle OLSR que cuidam do roteamento da rede. O esquema da divisão de classes do Teste 3 está demonstrado na Figura 4.

O *script* utilizando HTB e TBF para o Teste 4 trata de separar os tráfego em três classes. A primeira classe é para pacotes de voz do *codec* G.723.1, que será a classe de mais alta prioridade. Nessa classe está sendo usada a disciplina de enfileiramento TBF, em vez da FIFO (*First In First Out*) utilizada por *default* quando nenhuma disciplina é definida, onde simplesmente o primeiro pacote a chegar no *buffer* é o primeiro pacote a sair. O TBF é usado para delimitar a permanência máxima de um pacote na fila por 90ms, no caso

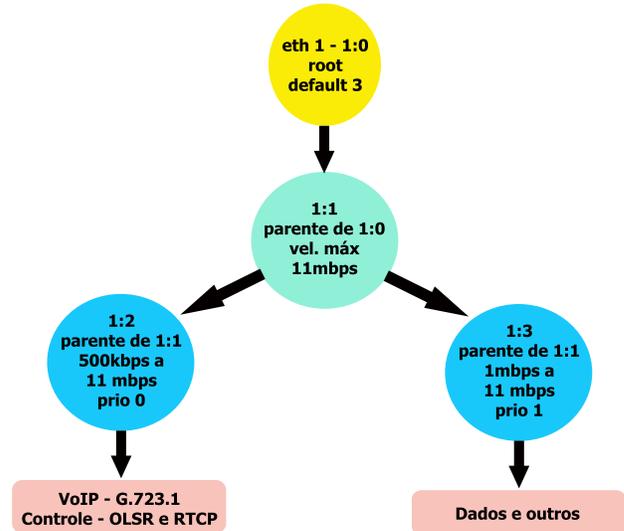


Fig. 4. Teste 3, QoS com duas classes

desse *script*. A segunda classe é para pacotes de controle RTCP e OLSR. E a terceira classe, de menor prioridade, abrange todos os outros tipos de pacotes que não forem filtrados pelas classes anteriores. O esquema de divisão de classes do Teste 4 está demonstrado na Figura 5.

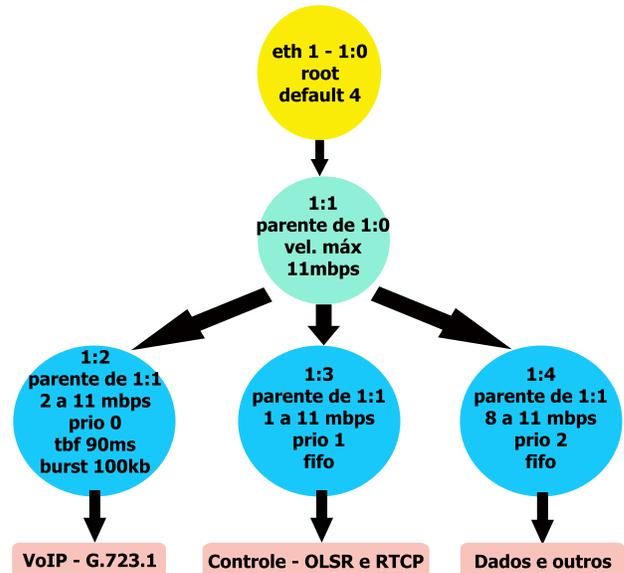


Fig. 5. Teste 4, QoS com três classes

V. RESULTADOS

Os fatores analisados dos logs gerados pelo *callgen* foram *jitter*, atraso e perda de pacotes. A partir deles, foi calculado o MOS de cada chamada a partir do modelo E [11].

Um fator interessante, não mostrado nos gráficos, é que em todos os testes, o atraso médio da rede foi constante na faixa de 168 a 170 ms. Percebe-se então que o atraso médio da rede, nesse caso, é fixo. Esta pode ser uma característica da própria rede.

A. Análise dos Resultados

Para cada gráfico a seguir, o intervalo de confiança de cada medida é de 95%.

A Figura 6 mostra os valores médios de MOS e seu intervalo de confiança para cada cenário.

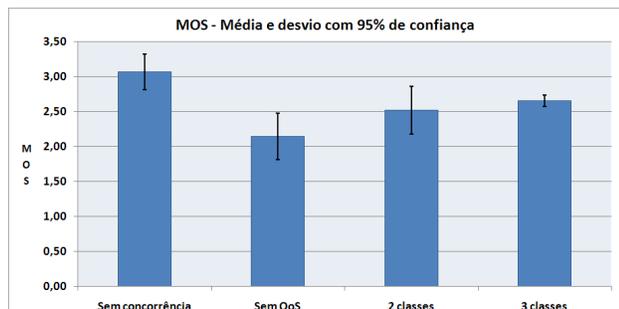


Fig. 6. Média e intervalo de confiança de MOS em cada cenário

O primeiro fator a notar é que o MOS médio máximo da rede é de 3,07, de acordo com o teste 1, sem tráfego concorrente, que seria a situação ideal. É um valor inaceitável comercialmente. Porém, o MOS máximo teórico do codificador utilizado é de 3,9, que é um valor baixo. A mudança de codificador, utilizando por exemplo o G.711 [9], codificador de alta qualidade recomendado pelo que codifica a uma taxa de 64kbps e tem um MOS teórico de 4,5, poderia melhorar a qualidade das chamadas. O que seria necessário mudar nos scripts de controle de tráfego seria o tipo de pacote a ser priorizado, a partir do tamanho e/ou porta utilizada, para ser de acordo com o codificador escolhido. Para efeito de registro, o MOS máximo atingido em uma chamada foi de 3,51; e tiveram chamadas com MOS baixíssimo, como por exemplo 1,63, o que não se deve a problemas na própria comunicação de voz, e sim provavelmente a interferências externas, que redes mesh sofrem bastante.

O Teste 2 mostra, com um valor médio de MOS das chamadas de 2,14, que a simples concorrência de um tráfego apenas pela banda já é o bastante para degradar significativamente a qualidade da chamada. Apenas duas chamadas tiveram o valor de MOS acima de 3, por exemplo. Os próximos gráficos mostrarão o porquê desse valor de MOS ter sido tão baixo.

Os testes 3 e 4 ilustram como o TC, independente do tipo de disciplina utilizada, faz com que a qualidade da chamada melhore, apesar de não chegar ao máximo possível. O teste 4, utilizando HTB com três classes e TBF, foi mais estável que o teste 3, que utiliza apenas HTB com duas classes. Apesar da média do teste 4 ter sido maior, foi o teste 3 que teve uma chamada que se aproximou mais do ideal para esse cenário, com MOS de 3,41.

A Figura 7 mostra os valores médios de *jitter* e seu desvio padrão para cada cenário.

Esse resultado demonstra que o modo que o problema que foi atacado com o TC melhora pouco nesse quesito. O *jitter* do teste de rede livre foi de 37,2 ms, que é um valor baixo. O baixo intervalo de confiança demonstra que em todas as chamadas o *jitter* foi estável.

No Teste 2 o *jitter* chegou a valores muito altos, com média de 101,6 ms. Mais uma prova que apenas um tráfego

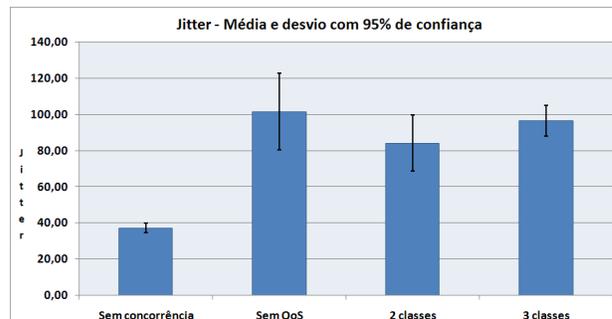


Fig. 7. Média e intervalo de confiança de *jitter* em cada cenário

concorrente faz com que a qualidade de uma chamada de voz abaixe muito.

Esperava-se que o uso de TC ajudasse a abaixar o *jitter*. O resultado demonstrado pelo Teste 3, com *jitter* médio de 84,2 ms, foi acima do esperado. Em vários casos o valor do mesmo passou de 100 ms. Para tentar sanar esse problema que foi elaborado o Teste 4, e estudado o uso de TBF para fazer com que o pacote de voz ficasse na fila por no máximo 90 ms. O resultado do Teste 4, apesar de ser mais estável com o desvio mais baixo dos testes de rede concorrente, teve um *jitter* médio de 96,5 ms. Mas muitos testes tiveram o valor de *jitter* em 90 ms. Fatores externos podem ter feito com que o seu valor em alguns testes chegassem a 130 ms, o que fez com que a média como um todo subisse.

Um caso a estudar é se abaixar o valor de permanência do pacote na fila TBF pode fazer com que o *jitter* abaixe, porém com cuidado de ver o impacto na taxa de perdas de pacotes (pois após o valor do período de permanência expirar, pacotes começam a ser descartados), fator que será analisado no gráfico a seguir.

A Figura 8 mostra os valores médios de porcentagem de perda de pacotes e seu intervalo de confiança para cada cenário.

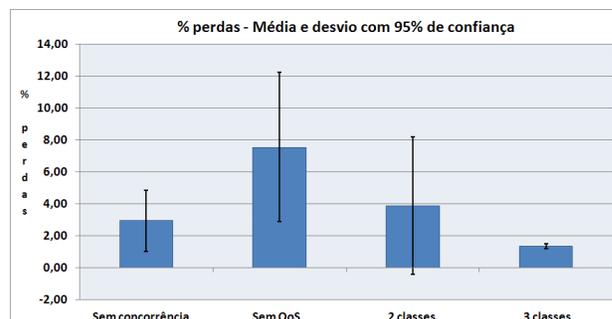


Fig. 8. Média e intervalo de confiança de porcentagem de perda de pacotes em cada cenário

O Teste 1 tem média de porcentagem de perdas de 2,94%. Em várias chamadas, o valor foi entre 1% e 3%. Em um teste, a porcentagem foi de 15% e em outro de 7%. Como a única concorrência no teste de rede livre era de pacotes de controle, o que não influencia muito nesse caso, esses valores pontuais nesses testes são de interferências externas na rede.

O resultado da porcentagem de perda do Teste 2 é mais uma prova de como a concorrência de tráfego faz com que a

qualidade da chamada de voz diminua. A média foi de 7,54%. Em vários testes a perda foi superior a 10% dos pacotes, chegando a um caso de uma chamada ter porcentagem de perda de 40%. Essa alta perda se deve ao fato de que vários pacotes chegando ao buffer faça com que ele encha, pacotes são descartados para esvaziá-lo. Essa situação foi abordada na Figura 2.

A simples separação em filas com uso de TC fez com que o valor médio da porcentagem de perda de pacotes caísse. No Teste 3 foi de 4%, e no Teste 4 foi de 1,34%.

No Teste 3 (exceto em duas chamadas que tiveram anormalidade de perdas de 6% e 29,7%) e no Teste 4 todas as chamadas tiveram porcentagem de perda de pacotes abaixo de 2%. Esse resultado foi esperado, pois está de acordo com o que foi citado na Seção 2, de o ideal de perdas ser abaixo de 2% em uma transmissão. A separação em mais classes (Teste 4) fez com que essa porcentagem de perdas fosse mais estável, pois mesmo excluindo as chamadas do Teste 3 que tiveram anormalidade, a média deles é igual ao Teste 4 com um desvio maior. O fato da porcentagem de perdas ser em torno de 1% foi mais comum no Teste 4.

B. Análise geral dos testes

Pôde-se perceber que o Teste 4 foi o mais estável, no fator MOS e no fator perdas, em ambos se aproximando do Teste 1. O que chamou a atenção em ambos cenários foi o baixo desvio nos resultados.

O Teste 2 mostra claramente como que um tráfego concorrente pode deteriorar a qualidade de uma chamada de voz. Houve várias chamadas em que o MOS aproximou-se de 1, o pior valor possível, com perdas de até 40% e valores de *jitter* de 186ms. Essas variações fora do comum são porque redes sem fio sofrem interferência de fatores externos nos roteadores. Por isso houve essa quantidade exaustiva de testes.

O Teste 3, apesar de apresentar alguns valores melhores que o 4, na média é abaixo. E ele é fixo, não há o que mudar no *script*. O Teste 4, ainda pode-se balancear o valor de latência no roteador, assim na tentativa de abaixar o *jitter* (apesar de estável, é alto), com uma tendência de descarte maior de pacotes. Deve-se analisar o fator *jitter* x porcentagem de perdas, de um modo que chegue a um valor de MOS aceitável.

Um resultado importante é que, independente do caso do Teste 3 (QoS com 2 classes) ou Teste 4 (QoS com três classes), o TC é uma ferramenta válida para aumentar o valor de MOS para perto do máximo possível da rede, dando uma maior qualidade à chamada, independente de haver ou não tráfego concorrente na rede.

Um resultado esperado foi que o HTB auxiliou na diminuição da taxa de perda de pacotes, garantindo ao UDP (protocolo de transporte sobre o qual chamadas de voz geralmente rodam) uma boa confiabilidade com relação à taxa de perdas e erros.

do controle de tráfego não ter feito com que a qualidade da chamada chegasse ao máximo possível da rede, ele ajudou a mesma a subir a um nível próximo do máximo. O resultado da baixa perda de pacotes é bastante útil, visto que a mesma foi menor até que na situação ideal, sem tráfegos concorrentes. Para resolver o problema do alto *jitter*, podem ser utilizadas outras soluções existentes, como por exemplo buffer através do uso de *playout* adaptativo no receptor [10].

AGRADECIMENTOS

Agradecemos à CNPq por ter financiado esse projeto, e aos componentes do grupo MIDIACOM da UFF por terem auxiliado em várias etapas dessa pesquisa, com sugestões e discussões.

REFERÊNCIAS

- [1] D.C. Muchaluat-Saade, C. Albuquerque, L.C. Schara Magalhães, D. Passos, J. Duarte e R. Valle, *Redes em Malha: Solução de Baixo Custo para Popularização do Acesso à Internet no Brasil*. XXV Simpósio Brasileiro de Telecomunicações (SBRT'07), Recife, PE, 03 a 06 de setembro de 2007.
- [2] E. Tavares, "*Um Estudo de Voz sobre IP em Redes em Malha 802.11*", Dissertação de Mestrado em Engenharia de Telecomunicações - TET/UFF, concluída em 15 de abril de 2008.
- [3] C. Shim, L. Xie, B. Zhang, C. Sloane, "*How delay and packet loss impact voice quality in VoIP*", Technical Report, Qovia, Inc., dezembro de 2003.
- [4] ITU-T P.800 recommendation, "*Methods for Subjective Determination of Transmission Quality*", disponível em <http://www.itu.int/rec/T-REC-P.800-199608-I/>, acessado em maio de 2009.
- [5] A. Markopoulou, F. A. Tobagi e M. J. Karam, "*Assessment of VoIP quality over internet backbones*". INFOCOM, New York City, NY, Junho de 2002.
- [6] LARTC, "*Linux Advanced Routing & Traffic Control*", Disponível em <http://lartc.org>, acessado em maio de 2009.
- [7] Página do Grupo de Redes de Computadores e Multimídia (GRCM) da Universidade Federal do Amazonas (UFAM), <http://grcm.dcc.ufam.edu.br/index.php/Principal>, acessado em maio de 2009.
- [8] ITU-T Recommendation H.323, "*Packet-based multimedia communications systems*". Genève, junho de 2006.
- [9] ITU-T Recommendation G.711, "*Pulse Code Modulation (PCM) of Voice Frequencies*", Disponível em <http://www.itu.int/rec/T-REC-G.711-198811-I/en>, acessado em maio de 2009.
- [10] R. Ramjee, J. Kurose, D. Towsley, H. Schulzrinne, "*Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks*", IEEE Infocom 94.
- [11] ITU-T Recommendation G.107, "*The E-model: a computational model for use in transmission planning*", Disponível em <http://www.itu.int/rec/T-REC-G.107/en>, acessado em maio de 2009.
- [12] ITU-T Recommendation G.723.1, "*General Aspects of Digital Transmission Systems - Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbit/s*", março de 1996.
- [13] <http://www.olsr.org>, "*Optimized Link Route State*", acessado em junho de 2009.

VI. CONCLUSÃO

Neste trabalho, testes reais mostram que a concorrência de outros fluxos na rede faz com que a qualidade de uma chamada de voz possa ser bastante deteriorada. Apesar do uso