

Novo Esquema de Escalonamento Cross-layer para Redes Multimídia – DQCA-RP

Josias T. Guimarães e José Marcos C. Brito

Resumo – Um aspecto importante nas redes sem fio multimídia é o escalonamento da transmissão dos pacotes gerados pelos terminais da rede. O DQCA (*Distributed Queuing Collision Avoidance*) é um protocolo de acesso ao meio de alto desempenho e estável, que mantém uma vazão alta com qualquer carga de tráfego. Entretanto, ele pode permitir índices altos de atraso e variação de atraso. Neste trabalho propõe-se um esquema de escalonamento *cross-layer*, baseado no DQCA, que visa reduzir o atraso médio, a variação do atraso e as perdas de pacotes de voz, sem queda significativa na vazão de dados. O desempenho do protocolo proposto é comparado, via simulação, com o DQCA e algumas de suas principais variantes.

Palavras chave – *Cross-Layer, DQCA, Escalonamento Oportunista*

Abstract – Scheduling is an important aspect in wireless multimedia networks. DQCA (*Distributed Queuing Collision Avoidance*) is a medium access control protocol for high performance which maintains a high throughput with any traffic load. However, it can allow high levels of delay and jitter. This paper proposes a *cross-layer* scheduling scheme, based on DQCA, which aims to reduce the average delay, jitter and packet loss of voice, without a significant decrease in data throughput. The performance of the proposed protocol is compared, via simulation, with DQCA and some of its main variants.

Keywords – *Cross-Layer, DQCA, Opportunistic scheduling*

I. INTRODUÇÃO

Nos últimos anos tem se observado uma grande expansão do uso das redes sem fio, com novas aplicações sendo incorporadas a estas redes, tornando-as redes multimídia.

Um aspecto importante na implementação das redes multimídia é o provimento de Qualidade de Serviço (QoS) para os diversos tipos de tráfego presentes.

Uma questão preponderante para definir o desempenho da rede e, consequentemente, sua habilidade para prover QoS, é o algoritmo de escalonamento utilizado para definir a ordem de transmissão dos pacotes gerados pelos terminais. O DQCA (*Distributed Queuing Collision Avoidance*) [1] e algumas de suas variantes [2][3] têm sido propostos como alternativas para a implementação desta função nas redes.

No entanto, embora o DQCA e suas variantes possuam elevada vazão, eles podem resultar em atraso e variação de atraso elevados para o tráfego de voz.

Este artigo tem por objetivo propor um novo algoritmo de escalonamento, utilizando *cross-layer*, denominado DQCA-RP (DQCA com reserva periódica). Este algoritmo

emprega no DQCA o conceito de reserva de *time-slots* semelhante ao protocolo PRMA [4], visando principalmente reduzir o atraso e a variação do atraso dos pacotes de voz, sem perda significativa da alta vazão de dados alcançável com o protocolo DQCA e suas variantes.

O restante deste artigo está organizado da seguinte forma: a Seção II resume a operação do protocolo DQCA e algumas de suas variantes; na seção III apresenta-se o protocolo proposto; a seção IV resume o ambiente de simulação utilizado; na seção V são apresentados os resultados da avaliação do desempenho do esquema proposto; finalmente, as conclusões são apresentadas na seção VI.

II. UMA VISÃO DO PROTOCOLO DQCA

O DQCA, proposto em [1], é um protocolo de acesso ao meio baseado no DQRAP (*Distribute Queuing Random Access Protocol*) [5] [6], distribuído, que funciona como um mecanismo de acesso aleatório sob condições de tráfego baixo e, na medida em que a carga de tráfego aumenta, passa suave e automaticamente para um esquema de reserva.

No DQCA o tempo é dividido em quadros que consistem de três campos. O primeiro é o campo de acesso, ou janela de contenção, que é dividido em m *minislots* de controle, durante os quais os nós podem requerer acesso ao canal enviando requisições de acesso (RTS – *Request to send*), sendo que 3 *minislots* de controle são suficientes para garantir o desempenho do sistema próximo do ótimo com qualquer carga de tráfego [7][5]; o segundo campo é o campo de dados (*Data Slot*), para transmissão de dados do usuário, e o terceiro o pacote de realimentação (FBP – *Feedback packet*), que é transmitido pelo *Access Point* (AP) a todos os nós e contém informações necessárias para que os nós gerenciem os estados das filas. O campo FBP contém as CTS (*clear to send*), que informam sobre o estado de cada *minislot* de controle, indicando sucesso, colisão ou vazio, a confirmação (ACK – *acknowledgment*) para o pacote transmitido no campo de dados e informações adicionais que podem ser incluídas dependendo da implementação do protocolo. São inseridos ainda pequenos intervalos SIFS (*short inter frame space*) a cada alternância entre transmissão no sentido *uplink* e *downlink*, ou seja, após o campo de Dados e após o pacote de realimentação.

O DQCA mantém duas filas distribuídas: a fila de transmissão de dados (DTQ – *Data transmission queue*) que é empregada no escalonamento da transmissão de pacotes, e a fila de resolução de colisões (CRQ – *Collision resolution queue*). A cada quadro, o nó na cabeça da DTQ está apto a transmitir seu pacote, enquanto os nós que estiverem na saída

Josias Teixeira Guimarães, Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas Gerais, Barbacena-MG, Brasil, Email: josias.guimaraes@ifsudestemg.edu.br.

José Marcos C. Brito, Instituto Nacional de Telecomunicações, Santa Rita do Sapucaí-MG, Brasil, Email: brito@inatel.br.

da CRQ tentarão resolver a colisão. Dessa forma, os processos de transmissão de dados e de resolução de colisões trabalham em paralelo. Quando um nó tem uma nova mensagem para transmitir, ele observa os estados das filas para verificar se está apto a enviar uma requisição. Para enviar uma requisição ele escolhe aleatoriamente um dos *minislots* de controle e transmite uma RTS. Em cada *minislot* de contenção, um entre três eventos podem ocorrer: nenhuma tentativa de requisição, sucesso na requisição por um único nó, ou requisições simultâneas por mais de um nó, gerando colisão. No caso de sucesso, o nó que enviou a requisição entra na DTQ e aguarda sua vez de transmitir a mensagem; se houver colisão, os nós que enviaram requisições em um mesmo *minislot* entram na mesma posição da CRQ e no próximo quadro aplica-se o algoritmo de resolução de colisões, no qual os nós que estiverem na cabeça da CRQ selecionam aleatoriamente um *minislot* de controle e reenviam uma RTS; os nós que enviarem as RTS com sucesso entram na DTQ e no caso de nova colisão os nós envolvidos entram novamente no final da CRQ e o processo se repete até que a CRQ fique vazia. Para evitar instabilidade, os usuários com mensagens novas só podem enviar requisições se a CRQ estiver vazia.

Mensagens longas são fragmentadas em pacotes menores, que são transmitidos um a um em quadros consecutivos.

Quando a DTQ e a CRQ estão vazias, um controle de acesso semelhante ao *Aloha* toma lugar, com o objetivo de reduzir o atraso de acesso.

As informações relativas às filas não são transmitidas diretamente pelo AP, mas processadas em cada nó ao final de cada quadro, com base em parâmetros alimentados pelo AP através do FBP. As decisões de escalonamento não são tomadas pelo AP, mas cada nó sabe quando transmitir, gerenciando suas posições nas filas distribuídas.

As filas são representadas nos nós por quatro valores inteiros: TQ, RQ, pTQ e pRQ. TQ representa o número de nós na DTQ aguardando para transmissão, enquanto RQ é o número de colisões na CRQ aguardando a resolução. Os parâmetros pTQ e pRQ representam a posição do nó na DTQ e na CRQ, respectivamente [2].

A. Algumas variantes do DQCA

A proposta original do DQCA considerou a disciplina de serviço FIFO (*First-In-First-Out*), na qual o usuário na primeira posição da DTQ tem sempre permissão para transmitir. Entretanto, outros parâmetros podem ser considerados, como o tipo de serviço ou a qualidade do sinal, para definir a ordem de transmissão dos usuários.

Em uma variante, proposta em [3] e aqui denominada de CL1 (*Cross-layer 1*), a DTQ é ordenada pelas taxas de transmissão disponíveis para os terminais, calculada e informada pelo AP com base em uma medida de relação sinal/ruído, de forma que os terminais com melhores condições do enlace têm prioridade na decisão de escalonamento. No caso de terminais com mesma taxa, a prioridade é dada ao terminal que estiver a mais tempo na fila. Este esquema visa principalmente o aumento na vazão e

redução no atraso médio de pacotes, apesar de apresentar problemas de justiça entre os terminais.

Em outra variante, também proposta em [3], denominada de CL2 (*Cross-layer 2*), a DTQ é ordenada por uma função de prioridade virtual, definida pela razão da taxa de transmissão do terminal pela sua posição na fila, Rb/pTQ. A idéia é melhorar a vazão enquanto provê certo nível de justiça, incentivando a transmissão de usuários com taxa de *bits* disponível mais alta, sem desconsiderar os seus tempos de espera na fila.

O esquema Dif1, denominado DQCA+ em [2], prioriza os pacotes de voz sobre os de dados, reduzindo o atraso desses pacotes. Para isso são utilizadas duas filas FIFO, uma para terminais de dados (DTQ) e outra para terminais de voz (VTQ), sendo esta mais prioritária. Neste esquema, terminais de dados podem obter o acesso somente quando a VTQ estiver vazia.

Já o esquema Dif2, proposto em [2] com a denominação de DQCA+CL, também utiliza duas filas, VTQ e DTQ, para o escalonamento das transmissões dos terminais de voz e de dados respectivamente, com prioridade para os de voz. Entretanto, a diferença deste esquema com relação ao Dif1 é que a DTQ é ordenada pelas taxas dos terminais. A utilização desta técnica *cross-layer* permite um aumento da vazão e a redução do atraso médio dos pacotes.

As quatro variantes descritas acima, juntamente com o DQCA básico, serão utilizadas nas comparações de desempenho apresentadas na Seção V.

III. UMA NOVA PROPOSTA DE PROTOCOLO – DQCA-RP

O DQCA-RP implementa algumas modificações no DQCA visando atender aos requisitos de QoS dos terminais de voz. As análises realizadas consideraram apenas tráfego de voz e dados, embora o protocolo possa ser adaptado para atender outros tipos de tráfego.

O mecanismo se baseia na reserva periódica de *time-slots* para os usuários de voz ativos, de acordo com a taxa de geração dos seus pacotes. Assim, quando um terminal se torna ativo e consegue acesso ao canal, ele passa a ter *time-slots* reservados com frequência igual à que os pacotes são gerados; os *time-slots* que não estiverem reservados podem ser utilizados pelos outros terminais da mesma forma que no DQCA básico.

O DQCA-RP mantém a estrutura do quadro do DQCA, o algoritmo de resolução de colisões e o esquema de acesso semelhante ao *Aloha* com carga de tráfego baixa. Como no esquema proposto em [8] e [2], para permitir a distinção do tipo do serviço, as RTS's transportam diferentes sinais, através de diferentes tipos de rajadas predefinidas, que permitem ao AP identificar o tipo de serviço do terminal.

Da mesma forma que nos esquemas Dif1 e Dif2 apresentados, no DQCA-RP é utilizada uma segunda fila de transmissão, VTQ (*Voice Transmission Queue*), para controlar as reservas e transmissões dos terminais de voz ativos. Do mesmo modo que as outras duas filas do DQCA básico, a VTQ também é mantida nos terminais por dois valores inteiros, VQ, que representa o número de terminais

de voz ativos (tamanho da fila), e pVQ, que representa a posição do terminal na fila.

Além disto, no DQCA-RP reserva-se *time-slots* para os terminais de voz ativos, de acordo com a taxa de geração de pacotes destes terminais. Assim, uma diferença básica com o DQCA é que no DQCA-RP o terminal de dados que ocupar a saída da DTQ adia a transmissão de um pacote sempre que o *time-slot* estiver reservado para um terminal de voz.

Para cada terminal de voz ativo, que estiver aguardando na VTQ, é associado um temporizador (*timer*) que indica o instante esperado para a chegada do seu próximo pacote; os valores dos *timer's* dos terminais de voz ativos são mantidos em todos os terminais através de um vetor de comprimento igual ao tamanho da VTQ, que é ordenada por este *timer*.

Sempre que um terminal de voz consegue o acesso, o seu *timer* é inicializado com o instante do início do quadro no qual ele transmitiu a RTS. Após a transmissão de cada pacote, o *timer* do terminal que transmitiu é incrementado pelo valor do intervalo entre chegadas de pacotes de voz, τ , ou recebe zero se for o último pacote da rajada de voz.

Ao receber o pacote de realimentação, cada terminal executa o fluxograma da Figura 1, que pode ser dividido em duas partes: na primeira parte, as atualizações das filas são realizadas com base nos estados dos *minislots*, recebidos através das CTS's. Estas atualizações visam principalmente incluir um terminal na fila de transmissão após uma requisição bem sucedida ou na fila de resolução de colisões, quando houver mais de uma requisição em um *minislot*. Na segunda parte do fluxograma é observado o estado do pacote transmitido no quadro, caso tenha sido transmitido algum, a fim de manter as filas de transmissão, os *timer's* e as reservas de *time-slots*.

A primeira parte do fluxograma é executada para cada CTS, referente a um respectivo *minislots* de contenção, que pode conter o valor 's', indicando sucesso no *minislot* (uma única requisição), 'c' indicando colisão no *minislot* ou 'i' para vazio, ou seja, nenhuma requisição. A seguir descrevemos as ações para cada um dos possíveis *status* associados aos *minislots*:

's' (sucesso) – é verificado o tipo do terminal que enviou a RTS. Se for um terminal de voz, a variável VQ, que indica o tamanho da fila de transmissão de voz, é incrementada de um, o *timer* do terminal que enviou a requisição recebe o instante do início do quadro e, no terminal que enviou a requisição, a variável pVQ recebe o valor de VQ, fazendo com que ele assuma a última posição na fila. Se for um terminal de dados, a variável TQ é incrementada de um e, no terminal que enviou a requisição, a variável pTQ recebe o valor de TQ, assumindo assim a última posição na fila de transmissão de dados.

'c' (colisão) – o valor de RQ, que indica o tamanho da fila de resolução de colisões, é incrementado de um e em cada terminal que transmitiu a requisição o valor de pRQ recebe o valor de RQ, de forma que eles passam a ocupar a última posição na fila de resolução de colisões.

'i' – se não houve nenhuma requisição no *minislot*, então verifica a próxima CTS, se houver, ou passa para a segunda parte do fluxograma ao concluir a última CTS.

Na segunda parte do fluxograma da Figura 1, se não houve transmissão de pacote com sucesso no quadro, o algoritmo é encerrado, caso contrário, verifica-se se o pacote transmitido é o último do terminal, indicado por um *bit* de final de mensagem (*bfm*), a fim de liberar ou não a reserva. No caso de *bfm*=0, se o pacote transmitido foi de um terminal de dados o algoritmo é encerrado, senão o *timer* do terminal de voz que transmitiu o pacote é incrementado com o valor referente ao intervalo entre chegadas de pacotes de voz. Para *bfm*=1, se foi transmitido um pacote de voz, o *timer* e o valor de pVQ do terminal que transmitiu recebem o valor zero e o valor de VQ é decrementado de um; se o pacote transmitido foi de dados, o valor de TQ é decrementado de um e, no terminal que transmitiu o pacote, o valor de pTQ recebe o valor zero. Desta forma, um terminal de voz libera a reserva sempre que deixa de transmitir (fica em silêncio).

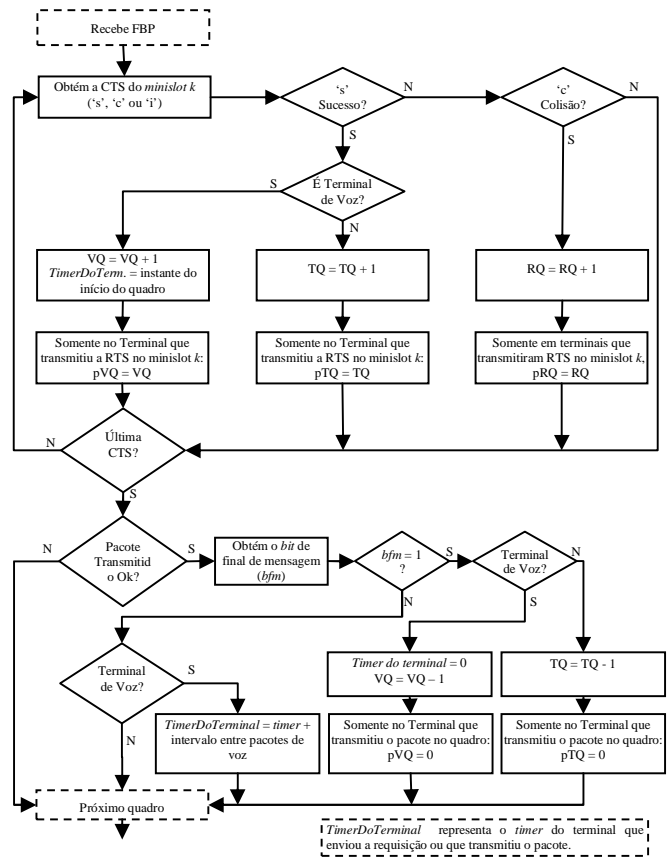


Fig. 1. Fluxograma do processo de manutenção das filas distribuídas e dos *timer's* de terminais de voz

IV. AMBIENTE DE SIMULAÇÃO

Para simular as variações nos estados dos enlaces dos terminais foi considerado um modelo de canal como o apresentado em [9], que propõem um canal de Markov com quatro estados, cada um representando uma das quatro taxas de transmissão disponíveis na camada física: 1, 2, 5.5 e 11 Mbps. As probabilidades de transição entre os diversos estados da cadeia foram fixadas como definido em [9].

O modelo de geração de tráfego de dados segue uma distribuição de Poisson e cada mensagem gerada nos terminais tem o tamanho definido por uma variável aleatória exponencial com média $10 Ld$, onde Ld é o tamanho de cada pacote de dados. Os parâmetros relacionados ao tráfego de dados foram fixados com os mesmos valores de [8] e [2].

O modelo utilizado para a geração de tráfego dos terminais de voz é o modelo ON-OFF, representado através de uma cadeia de Markov de dois estados, na qual os terminais podem estar no estado OFF (silêncio) ou no estado ON (ativo) [8]. A probabilidade de transição do estado ON para o estado OFF é denominada de α , enquanto a probabilidade de transição do estado OFF para o estado ON é denominada de β . O tempo médio que um terminal permanece em cada estado, ON ou OFF, é uma variável aleatória com distribuição exponencial de médias $1/\alpha$ e $1/\beta$ respectivamente [10] e foram definidas, seguindo [11], com os valores 1,004 segundos e 1,587 segundos. A taxa de transmissão, quando no estado ON, é de 13 Kbps.

Os demais parâmetros considerados para a configuração do quadro foram fixados de acordo [8], [2], [9] e [1].

V. RESULTADOS OBTIDOS

Para a comparação do desempenho do protocolo proposto com soluções anteriores desenvolveu-se um simulador, em Linguagem Java, capaz de simular tanto o protocolo proposto quanto os protocolos utilizados como referência para comparação de desempenho (CL1, CL2, Dif1, Dif2 e DQCA).

Os parâmetros de desempenho utilizados para comparação são a vazão para o tráfego de dados e o atraso médio, a variação do atraso (*jitter*) e a perda de pacotes para o tráfego de voz. Todos os parâmetros de desempenho foram determinados, como em [3][2], em função da carga submetida à rede pelos terminais de dados.

Pacotes de voz que tenham sofrido atrasos excessivos são considerados inúteis no processo de decodificação. Assim, em nossas análises, consideramos que os pacotes de voz são descartados (perdidos) quando sofrem atrasos superiores a 300 ms, tendo-se fixado o limite de perdas tolerável em 1%. Assim, para cada protocolo analisado, a simulação foi encerrada se a taxa de perda de pacotes atingiu este patamar.

A Fig. 2 apresenta os resultados de atraso médio de pacotes de voz em função das cargas submetidas, considerando 10 terminais de voz compartilhando o canal. Com os esquemas DQCA, CL1 e CL2 o índice de perdas de pacotes de voz excede o limite de 1% com cargas submetidas a partir de 1,5 Mbps. Observa-se pela figura que o esquema proposto, DQCA-RP, provê o menor índice de atraso médio entre os protocolos estudados, com redução de 88,3% e 82,3% sobre o Dif1 e o Dif2, respectivamente, para altas cargas de tráfego de dados.

Ainda de acordo com a Fig. 2, observa-se que com os esquemas Dif2 e DQCA-RP, o atraso médio cai com o aumento da carga de dados. Este comportamento se deve ao dimensionamento do quadro quando as filas de transmissão estão vazias.

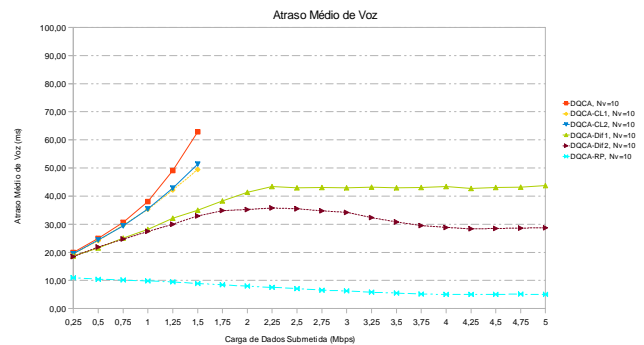


Fig. 2. Atraso médio de voz em função da carga de dados submetida.

Para estimar a variação do atraso para os pacotes de voz utilizou-se como parâmetro o desvio padrão do atraso, que mede a dispersão do atraso em relação a seu valor médio. Assim, quanto maior o desvio padrão, mais disperso em relação à média está o atraso e, conseqüentemente, maior é o *jitter* na rede.

Na Fig. 3 apresenta-se o desvio padrão do atraso dos pacotes de voz, em função da carga de dados, para os protocolos analisados. Observa-se que no DQCA-RP a redução do desvio padrão do atraso destes pacotes chega a 81,4% e 74,6% sobre os valores obtidos com os esquemas Dif1 e Dif2 respectivamente, mantendo-se sempre abaixo de 10 ms independente da carga de dados submetida. A redução no desvio padrão com o aumento da carga de dados se deve ao mesmo fenômeno comentado para a curva de atraso médio.

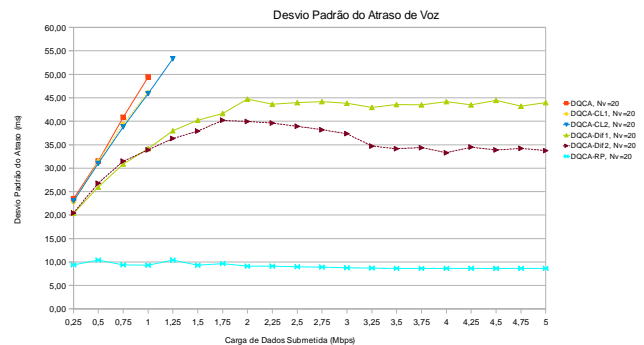


Fig. 3. Desvio padrão do atraso de voz em função da carga de dados submetida

A Fig. 4 apresenta os resultados de perdas de pacotes de voz, em função da carga de dados submetida, considerando 10 terminais de voz compartilhando o canal. Observa-se que, com os esquemas que não priorizam os terminais de voz nas decisões de escalonamento, DQCA, CL1 e CL2, o limite de perdas de pacotes de voz (1%) é alcançado antes da carga de dados submetida chegar a 1,5 Mbps. Os esquemas Dif1 e Dif2 apresentam baixos índices de perdas, com o segundo apresentando um desempenho melhor devido à utilização da técnica de escalonamento oportunista. De todos os protocolos estudados, claramente o DQCA-RP é o que apresenta o melhor desempenho com relação a perdas de

pacotes de voz. Isto se deve à reserva de *time-slots* para os pacotes de voz à medida que eles são gerados nos terminais, o que reduz os atrasos e conseqüentemente os descartes de pacotes.

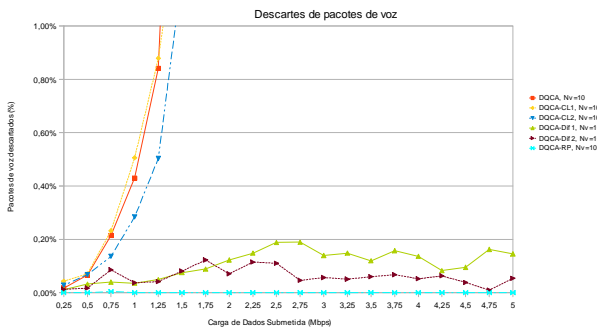


Fig. 4. Descartes de pacotes de voz em função da carga de dados submetida

A Figura 5 apresenta os resultados de vazão obtidos para os protocolos analisados. Novamente, a simulação foi finalizada se a taxa de perdas de pacotes de voz ultrapassou 1%. Na figura são apresentados os resultados de vazão em função das cargas submetidas de dados, com 10 e 20 terminais de voz. Observa-se que nos três esquemas que não diferenciam terminais de voz e de dados na decisão de escalonamento, DQCA, CL1 e CL2, com cargas de dados a partir de 1,5 Mbps os pacotes de voz sofrem perdas acima de 1% do total gerado, violando assim o requisito de QoS desta classe de tráfego. Como o esquema Dif1 prioriza os terminais de voz sobre os de dados, ele não viola o requisito de QoS desses terminais, entretanto, como ele não emprega o conceito de escalonamento oportunista, a vazão de dados não ultrapassa 2,15 e 1,97 Mbps, com 10 e 20 terminais de voz, respectivamente. Nota-se ainda que o esquema Dif2 apresenta o melhor desempenho em vazão de dados. O DQCA-RP apresenta uma vazão 1,8% inferior ao Dif2, o que representa um custo baixo frente aos ganhos obtidos em atraso médio, *jitter* e perdas dos pacotes de voz.

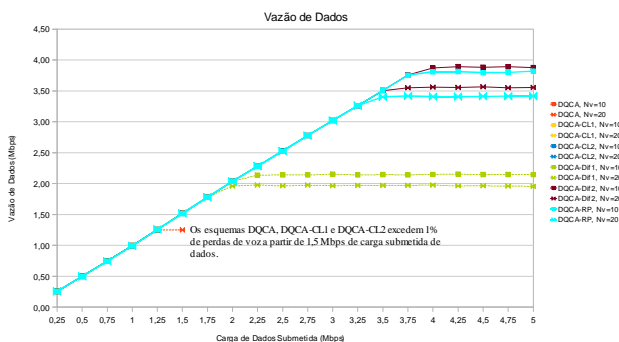


Fig. 5. Vazão de dados em função da carga de dados submetida com 10 e 20 terminais de voz.

VI. CONCLUSÕES

Neste artigo foi proposto um novo protocolo de escalonamento *cross-layer*, oportunista, baseado no DQCA.

O desempenho deste protocolo foi comparado, por meio de simulação, com versões anteriores do DQCA, nas quais ele se baseou.

Os resultados indicaram que, embora o esquema proposto resulte em pequena perda de vazão de dados, ele oferece ganhos significativos para os pacotes de voz, em termos de atraso médio, variação de atraso e taxa de perda de pacotes.

REFERÊNCIAS

- [1] Alonso, L; Ferrús, R. and Agustí, R. WLAN Throughput Improvement via Distributed Queuing MAC, IEEE Commun. Lett., vol. 9, no. 4, pp. 310–12, Apr. 2005.
- [2] Kartsakli, E.; Cateura, A.; Verikoukis, C. and Alonso, L. A Cross-Layer Scheduling Algorithm for DQCA-based WLAN Systems with Heterogeneous Voice-Data Traffic, The 14th IEEE Workshop on Local and Metropolitan Area Networks, 2005. LANMAN 2005, pp. 1–6. doi: 10.1109/LANMAN. 2005.
- [3] Verikoukis, C.; Alonso, J.; Kartsakli, E.; Cateura, A. and Alonso, L. Cross-Layer Enhancement for WLAN Systems Based on a Distributed Queuing MAC Protocol. Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd , vol.3, no.pp. 1293- 1297, 07-10 May 2006.
- [4] Goodman, D. J.; Valenzuela, R. A.; Gayliard, K. T. and Ramamurthi, B. Packet reservation multiple access for local wireless communications. IEEE Transactions on Communications, pp. 885-890, Aug. 1989.
- [5] Xu W. and Campbell, G. DQRAP, A distributed queuing random access protocol for a broadcast channel, Computer Commun. Rev., vol. 23, no. 4, pp. 270–278, Oct. 1993.
- [6] Zhang, X. and Campbell, G. Performance analysis of distributed queuing random access protocol-DQRAP. DQRAP Research Group Rep. 93-1, Illinois Institute of Technology, 1993.
- [7] Xu, W. and Campbell, G. A near perfect stable random access protocol for a broadcast channel, in IEEE Proc. ICC'92, vol. 1, pp. 370–374, 1992.
- [8] Kartsakli, E.; Cateura, A.; Alonso, J.; Verikoukis, C. and Alonso L. Cross-Layer Enhancement for WLAN Systems with Heterogeneous Traffic based on DQCA, IEEE Communications Magazine, vol. 46, no. 6, pp. 60–66, 2008.
- [9] Zárate, J. A.; Kartsakli, E.; Cateura, A.; Verikoukis, C. and Alonso, L. A near-optimum cross-layered distributed queuing protocol for wireless LAN. IEEE Wireless Communication Magazine, Special Issue on MAC protocols for WLAN, vol. 15, no. 1, pp. 48-55, 2008.
- [10] Brady, P. T. A Model for Generating On-Off Speech Patterns in Two-way Conversation, The Bell System Technical Journal, Vol 48, No. 2, pp 2445-2472, 1969.
- [11] ITU-T Recommendation P.59 Artificial Conversational Speech. Disponível em <http://www.itu.int/rec/T-REC-P.59-199303-I>. Acesso em 22 de Agosto de 2011.