

Uma Arquitetura em Dois Níveis para o Suporte de Comunicação de Tempo-Real em Redes CSMA

Ricardo Moraes, Francisco Vasques e Paulo Portugal

Resumo—Neste artigo uma nova abordagem para o suporte de comunicações de tempo-real (TR) em redes IEEE 802.11 é proposta. Esta abordagem é baseada em uma arquitetura em dois níveis, sendo definido no nível mais baixo (camada MAC) um mecanismo que força a resolução de colisões em favor das estações de TR. No nível superior, um mecanismo de coordenação assegura o acesso livre de colisões entre as estações de TR. Esta arquitetura é capaz de priorizar o tráfego de TR sem a necessidade de controlar o funcionamento das estações não TR.

Abstract—In this paper, a new approach to support real-time (RT) communication in IEEE 802.11 networks is proposed. It is based on a 2-tier architecture. In the lower layer (MAC layer), a forcing collision resolution mechanism enforces a higher priority level to the RT stations. In the upper layer, a coordination mechanism ensures a collision-free access among the set of RT stations. The proposed architecture prioritizes the RT traffic over non-RT traffic, without the need to strictly control the timing behavior of every communication device.

I. INTRODUÇÃO

O interesse em tecnologias de redes sem fio em diversas áreas que exigem um elevado nível de Qualidade de Serviço (QoS), tem crescido nos últimos anos. Os exemplos mais comuns variam de simples aplicações de voz sobre IP (VoIP), até complexos sistemas de controle via redes (*Networked Control Systems* - NCS). Para este tipo de domínio de aplicação, o suporte de comunicações confiáveis é um dos principais requisitos [1]. Por exemplo, em sistemas de automação, os dados referentes ao sistema de controle devem ser periodicamente transferidos entre sensores, controladores e atuadores, obedecendo metas-temporais (*deadlines*) estritas.

Quando os serviços de comunicação sem fio são utilizados para o suporte de aplicações de tempo-real (TR), requisitos específicos de comunicação devem ser considerados, nomeadamente: (i) que o meio físico é essencialmente um *ambiente de comunicação aberto*, ou seja, qualquer estação pode tentar acessá-lo, de forma a estabelecer um canal de comunicação; (ii) que o ambiente de comunicação é altamente susceptível a interferências, não só daqueles sistemas que usam a mesma tecnologia, mas também de outras tecnologias que operam na mesma frequência de transmissão [2]. Então, a taxa de erros por bit (*Bit Error Rate* - BER) não pode ser desconsiderada. Além disso, a infra-estrutura de comunicação de TR é comumente compartilhada com o tráfego não tempo-real (NTR)

[3], o que induz a imprevisibilidade no acesso ao meio de comunicação compartilhado. Conclui-se, portanto, que a carga do sistema não pode ser previsível na etapa de projeto, nem pode ser eficazmente controlada durante o tempo de execução.

Tradicionalmente, a comunicação de TR em redes cabeadas foi garantida através do controle estrito de todos os dispositivos de comunicação. A coexistência de estações de TR com estações NTR em redes cabeadas tem sido possível através de mecanismos de suavização de tráfego aplicados às estações NTR [4]. Infelizmente, esta abordagem não é adequada para ambientes de comunicação sem fio, uma vez que não é possível impor qualquer restrição de tráfego às estações que estão fora da esfera-de-controle da arquitetura de TR.

O restante deste artigo está organizado da seguinte maneira: a seção II apresenta os trabalhos relacionados ao suporte de comunicações de TR em redes IEEE 802.11, a seção III descreve em detalhes a arquitetura proposta. A análise por simulação é apresentada na seção IV e, finalmente, a conclusão é feita na seção V.

II. TRABALHOS RELACIONADOS

Atualmente, as redes padronizadas pelo grupo de trabalho IEEE 802.11, que também são conhecidas como redes WiFi, são o padrão *de facto* em conectividade sem fio para redes locais (WLANs). Há uma grande semelhança entre as redes cabeadas IEEE 802.3 e as redes sem fio IEEE 802.11, uma vez que ambas usam um mecanismo semelhante para gerir o acesso ao meio, o CSMA (*Carrier Sense Multiple Access*). A principal diferença é que o padrão IEEE 802.11 implementa o procedimento CSMA/CA (*CSMA with collision avoidance*), também conhecido como DCF (*Distributed Coordination Function*), ao invés do tradicional CSMA/CD (*CSMA with Collision Detection*) utilizado pelo padrão IEEE 802.3. A emenda IEEE 802.11e foi proposta com o objetivo fornecer diferenciados níveis de QoS para as aplicações, incluindo o transporte de voz e vídeo em WLANs. Este padrão incorpora uma nova função de coordenação, chamada HCF (*Hybrid Coordination Function*), que é usada somente em configurações com QoS¹.

Alguns mecanismos para o suporte à comunicação de TR já estão incluídos nos padrões IEEE 802.11/802.11e. O PCF (*Point Coordination Function*) foi proposto no padrão original IEEE 802.11 como um mecanismo de acesso opcional, onde o coordenador de acesso, normalmente localizado no AP (*Access Point*), implementa um esquema de *polling* centralizado para

Ricardo Moraes, LabSEC, Universidade Federal de Santa Catarina, Florianópolis, Brasil e Instituto de Telecomunicações, Universidade de Aveiro, Aveiro, Portugal, E-mail: rmoraes@inf.ufsc.br.

Francisco Vasques e Paulo Portugal, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal, E-mails: vasques@fe.up.pt e pportugal@fe.up.pt.

¹Para maiores detalhes sobre a descrição dos protocolos IEEE 802.11/802.11e, recomenda-se consultar as normas [5], [6].

a transmissão de dados síncronos. Apesar do mecanismo PCF estar supostamente bem adaptado para suportar aplicações sensíveis ao atraso, a maioria das placas WLAN nunca implementaram o mecanismo, devido a razões de complexidade. O mecanismo HCCA (*HCF controlled channel access*) foi proposto como uma melhoria do PCF. No entanto, estudos preliminares (por exemplo, [7]) já demonstraram que este mecanismo pode não ser adequado para garantir os requisitos específicos das aplicações de TR. Mas, uma série de melhorias está sendo proposta para reduzir o *polling overhead* do HCCA, por exemplo [8].

Além disso, encontra-se na literatura diversas abordagens inovadoras que estão sendo desenvolvidas para oferecer serviços de comunicação de TR em redes sem fio. As principais soluções são baseadas em esquemas de *Polling* [9], [10], *Master-Slave* [11] e *Token Passing* [12], [13]. É digno de destacar a proposta apresentada por Sobrinho e Krishnakumar [14], que adapta o mecanismo EQuB [15] para redes sem fio. O esquema proposto, o *black-burst* (BB), implementa um sistema MAC distribuído em redes *ad hoc*. Este mecanismo exige a substituição do esquema de transmissão aleatório. As estações de TR implementando a abordagem BB tentam acessar o meio de transmissão, após detectá-lo livre por um período igual a t_{med} , menor que o tempo de espera das estações padrão IEEE 802.11. Desta forma, as estações de TR têm prioridade sobre as outras estações. Quando uma estação de TR pretende transmitir, ordena seus direitos de acesso através de pulsos de energia (BBs), imediatamente após detectar o meio livre durante o período de tempo t_{med} . O comprimento dos BBs transmitidos por uma estação de TR é uma função crescente do atraso de contenção experimentado pela estação.

O principal problema das abordagens encontradas na literatura é o requisito de um *ambiente de comunicação fechado* para garantir a comunicação de TR (uma exceção é a abordagem BB que se baseia numa abordagem de induzir a resolução das colisões, em favor das estações de TR). Caso contrário, estas soluções não funcionarão corretamente, uma vez que não são capazes de lidar com as interferências geradas por mensagens enviadas por outras estações (NTR). No entanto, presumir que é possível criar uma zona totalmente livre de tentativas de transmissão de estações NTR em ambientes de comunicação sem fio, não é um pressuposto realista. Pelo que, as abordagens tradicionais para suportar comunicação de TR falham em ambientes de comunicação sem fio.

Portanto, para garantir uma comunicação de TR é necessário mudar das tradicionais abordagens, nas quais o meio físico permitia a consideração de ambientes de comunicação *fechados*, para abordagens que tratam ambientes de comunicação *abertos*. Então, um novo paradigma de comunicação deverá emergir, uma vez que o atual paradigma é ainda baseado em ambientes fechados e totalmente controlados. Assim, as soluções mais promissoras nos cenários de comunicações futuros (*abertos*) são aquelas que permitem a coexistência entre estações de TR e NTR no mesmo domínio de comunicação, garantindo prioridade de tráfego para as estações de TR.

Mais precisamente, consideramos que uma arquitetura estruturada em dois níveis (*2-tier*) é adequada (Figura 1), sendo que: (a) na camada de mais baixo nível (camada MAC), um

mecanismo que força a resolução de colisões (FCR MAC) em favor das estações de TR deve ser definido; (b) na camada superior (*coordination layer*), deve haver um mecanismo de coordenação que assegure o acesso livre de colisões entre o conjunto de estações de TR. Este mecanismo de coordenação pode por exemplo, ser baseado num esquema de passagem de *token* ou num mecanismo TDMA (*Time Division Multiple Access*). Esta abordagem difere das tradicionais, as quais comumente consideram somente a camada de coordenação entre o conjunto de estações de TR (desprezando a existência de estações NTR compartilhando o mesmo meio de comunicação).

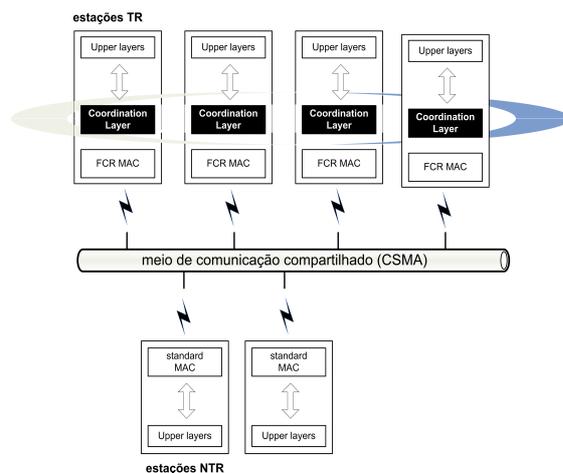


Fig. 1. Estrutura da arquitetura proposta.

A arquitetura VTP-CSMA está sendo proposta no contexto acima especificado. Dentro do nosso conhecimento, a arquitetura VTP-CSMA e a abordagem BB [14], [16] são as únicas soluções que permitem a comunicação de TR em ambientes abertos. A principal desvantagem do BB é que ele obriga a modificação da camada MAC (e possivelmente também de partes da camada PHY), impedindo o uso de COTS hardware. Embora o mecanismo VTP-CSMA também modifique partes da camada MAC, este mecanismo pode ser facilmente implementado em COTS hardware (por exemplo, FPGA) sobre dispositivos padrão IEEE 802.11.

III. A ARQUITETURA VTP-CSMA

A arquitetura VTP-CSMA é baseada no esquema apresentado na Figura 1. O mecanismo de separação de tráfego (FCR MAC) funciona da seguinte maneira: sempre que uma colisão entre uma estação de TR e um conjunto de estações NTR (implementando o padrão IEEE 802.11) ocorre, todas as estações envolvidas, com exceção da estação TR, utilizam os valores padrões definidos na especificação do mecanismo EDCA. Portanto, selecionam um intervalo de *backoff* aleatório de acordo com a categoria de acesso em que a mensagem foi transmitida (*voz*, *vídeo*, *best-effort* ou *background*). Por outro lado, as estações de TR, de agora em diante também referenciadas como estações VTP-CSMA, transferem o seu tráfego na categoria de mais alta prioridade do mecanismo EDCA, ou seja, definindo o *Arbitration Interframe Space*

(AIFS) para: $AIFS[VO] = aSIFSTime + 2 \times aSlotTime$ e os valores das janelas de contenção para $aCWmin[VO] = aCWmax[VO] = 0$.

Portanto, sempre que uma colisão ocorre, qualquer estação VTP-CSMA tenta transmitir suas mensagens no primeiro *slot* disponível do mecanismo EDCA, enquanto todas as outras estações aguardam durante um intervalo de tempo definido pela função de *backoff*. No entanto, se duas ou mais estações VTP-CSMA, simultaneamente, disputassem o meio de acesso, suas mensagens colidiriam e haveria um descarte após o número máximo de tentativas de retransmissão. Esta limitação é a principal razão do mecanismo complementar na camada superior (*coordination layer*), permitindo que múltiplas estações de TR possam coexistir com múltiplas estações NTR. O mecanismo proposto neste artigo é a passagem de *token* virtual (VTP). A principal vantagem deste mecanismo em relação aos tradicionais, é que nesta proposta não há troca de mensagens adicionais para a passagem do *token*. Este mecanismo é baseado num trabalho anterior proposto no contexto das redes Ethernet [17].

O mecanismo VTP considera um processo de grupo G com np membros. Os participantes do grupo são representados como $L = \{NA_1, NA_2, \dots, NA_{np}\}$. Onde NA_i denota a i -ésima estação no grupo G , e também é utilizada como um identificador da estação (ID). Este mecanismo circula um *token* virtual em L . Especificamente, todos os membros do grupo G mantêm um Contador de Acesso (ACo). A i -ésima estação VTP-CSMA *captura* o *token* quando ACo é igual NA_i . Se a estação tem mensagens na fila, transmite-as imediatamente durante um intervalo de tempo definido pela oportunidade de transmissão (TXOP) das estações de TR. O mecanismo de separação de tráfego proposto (FCR MAC) prioriza a mensagem de TR sobre potenciais mensagens NTR, garantindo que a estação VTP-CSMA terá acesso ao meio. No final da TXOP, todas as estações VTP-CSMA incrementam o valor da variável local ACo , passando o *token* para a próxima estação (estação com $NA_i = ACo + 1$)². Sempre que a estação VTP-CSMA que possui o *token* não tem mensagens para transmitir, permite-se que estações NTR possam competir pelo acesso ao meio durante um intervalo de tempo $t_2 = aSIFSTime + 3 \times aSlotTime$. Nos mecanismos EDCA e DCF qualquer estação pode iniciar uma transmissão após $aSIFSTime + 2 \times aSlotTime$. No caso de uma transmissão por uma estação NTR, o valor do ACo também é incrementado no final desta transmissão. Portanto, o mecanismo proposto permite a coexistência de estações TR com estações NTR implementando os padrões IEEE 802.11/802.11e.

A Figura 2 mostra o pseudocódigo do mecanismo VTP-CSMA, que aparece representado por quatro procedimentos: *Initialization*, *Main*, *Transmission* and *Listening*. De acordo com o procedimento *Initialization* (linha 2, Fig. 2), o valor do contador de acesso ACo é inicializado com NA_1 em todas as estações VTP-CSMA. Três variáveis (t_1 , t_2 , t_3) do tipo inteiro são definidas, onde t_1 e t_2 são contadores de *slots*, ao passo que t_3 é um contador de colisões especial usado

²O valor de ACo deve ser incrementado por uma operação "mod", ou seja, $ACo = (ACo \text{ mod } np) + 1$. Portanto, quando refere-se que o valor de ACo é incrementado, executa-se a referida operação "mod".

```

1  ∀ process NAi
2  Initialization: ACo ← NA1; t1 ← 0; t2 ← 0; t3 ← 0;
3  Main: while t3 ≤ RN do
4      if ACo = NAi then
5          {wait AIFS[VO]};
6          NAi executes Transmission;
7      else
8          {wait SIFS};
9          NAi executes Listening;
10     endif
11  end while

```

Fig. 2. O mecanismo VTP-CSMA.

no procedimento de re-inicialização (mecanismo de reset). O procedimento *Main* é executado no início de cada *slot*, onde as diferentes estações VTP-CSMA verificam se o contador de colisão (t_3)³ excedeu o número máximo de tentativas de retransmissão (RN). Sempre que este número é excedido, o procedimento *Initialization* é executado. Em seguida, dependendo do evento ocorrido no *slot* anterior e do valor do ACo , cada estação VTP-CSMA irá executar uma ação específica.

O procedimento *Transmission* somente será executado pela estação VTP-CSMA que possui o direito de transmissão ($ACo = NA_i$). Este procedimento só será inicializado após o meio estar livre durante $AIFS[VO]$, ou seja, o menor valor de AIFS especificado no mecanismo EDCA. Por outro lado, o procedimento *Listening* pode ser inicializado depois do meio estar livre durante SIFS (*Short Interframe Space*).

Cinco estados do canal de transmissão são definidos (determinados no início de cada *slot*):

- 1) *transmission* - Uma ou mais mensagens de outras estações estão sendo transmitidas no canal;
- 2) *successful transmission* - O canal está livre e uma mensagem foi transmitida com sucesso a partir de outras estações no *slot* anterior;
- 3) *continuing idle* - O canal está livre e também estava livre no *slot* anterior;
- 4) *idle after collision* - O canal está livre e houve uma colisão no *slot* anterior;
- 5) *successful*: O canal está livre, "Eu sou a estação que estava transmitindo e a transmissão de uma ou mais mensagens (limitadas pela TXOP) finalizou no *slot* anterior".

De acordo com o estado do canal, definidos acima, e do valor do contador de acesso local ACo , cada estação VTP-CSMA executa uma ação específica. Inicialmente, sempre que a estação VTP-CSMA *captura* o *token* virtual ($ACo = NA_i$), esta estação executará o procedimento *Transmission* (Figura 3). Este procedimento funciona da seguinte maneira. Se a estação VTP-CSMA que possui o *token* tem mensagens para transmitir, a transmissão é iniciada imediatamente. Se uma colisão ocorre, a estação VTP-CSMA incrementa seu con-

³Quando a resolução de colisões inicia-se, isto pode ser uma consequência de três tipos diferentes de colisões: (1) entre estações NTR; (2) entre estações NTR e uma estação VTP-CSMA que possui o *token*; (3) entre estações VTP-CSMA. O primeiro cenário pode ser detectado se o meio de comunicação permanece livre durante o intervalo t_2 . O segundo cenário é facilmente resolvido em favor da estação VTP-CSMA. Finalmente, quando múltiplas estações VTP-CSMA simultaneamente tentam acessar o canal de comunicação significa que os contadores ACo perderam a sincronização, forçando a reinitialização do anel.

tador t_3 e tenta retransmitir suas mensagens até o número máximo definido para retransmissões (RN). Sempre que uma transmissão com sucesso ocorre, a estação VTP-CSMA, que possui o *token*, executará o procedimento *Listening* (Fig. 3, linhas 4-5), onde cada estação VTP-CSMA incrementa o valor ACo , passando o *token* para a próxima estação. Por outro lado, sempre que a estação VTP-CSMA que possui o *token* não tem mensagens para transmitir, o mecanismo permitirá que estações NTR tentem acessar o meio de transmissão durante um período de tempo múltiplo de $aSlotTime$.

```

1 Transmission:
2 if  $NA_i$  has message to be transmitted then
3   start the transmission; wait for transmission to complete;
4   if successful transmission then
5      $t_1 \leftarrow 1$ ;  $t_3 \leftarrow 0$ ; go to Listening;
6   else;
7      $t_3++$ ;
8     if  $t_3 \leq RN$  then
9       go to Transmission;
10    endif
11  endif
12 else
13    $t_2 \leftarrow 2$ ; go to Listening;
14 endif

```

Fig. 3. Procedimento *Transmission*.

Como ilustrado na Figura 2, todas as estações VTP-CSMA que não têm o token ($ACo \neq NA_i$) executam o procedimento *Listening* (Figura 4) e, dependendo do estado do canal, executam as ações específicas: a) *transmission* (linhas 5-7) - todas as estações VTP-CSMA aguardam pelo fim da transmissão e atualizam os valores das variáveis t_1 e t_2 ; b) *successful transmission* (linhas 8-9) - todas as estações VTP-CSMA atualizam t_1 , t_2 e t_3 ; c) *continuing_idle* (linhas 10-17) - todas as estações VTP-CSMA incrementam t_2 e verificam o valor de t_1 . Se $t_1 = 1$, todas as estações VTP-CSMA incrementam o valor de ACo . Ou seja, $t_1 = 1$ e o estado do canal *continuing_idle* significa que uma transmissão com sucesso ocorreu e também sinaliza o final de uma TXOP. Além disso, t_2 será incrementado e, cada vez que o valor de t_2 é maior ou igual a 3, todas as estações VTP-CSMA incrementam o valor de ACo ; d) *idle_after_collision* (linhas 18-19) - todas as estações VTP-CSMA incrementam os valores de t_2 e t_3 .

```

1 Listening:
2 At the beginning of the next slot
3 event  $\leftarrow$  channel event during the last slot;
4 switch (event)
5   case transmission:
6     wait for transmission to complete;
7      $t_1 \leftarrow 0$ ;  $t_2 \leftarrow 0$ ; break;
8   case successful transmission:
9      $t_1 \leftarrow 1$ ;  $t_2 \leftarrow 1$ ;  $t_3 \leftarrow 0$ ; break;
10  case continuing_idle:
11     $t_2++$ ;
12    if  $t_1 = 1$  then
13       $ACo++$ ;  $t_1 \leftarrow 0$ ;  $t_2 \leftarrow 0$ ; break;
14    endif
15    if  $t_2 \geq 3$  then
16       $ACo++$ ;  $t_1 \leftarrow 0$ ;  $t_2 \leftarrow 0$ ;  $t_3 \leftarrow 0$ ; break;
17    endif;
18  case idle_after_collision:
19     $t_2++$ ;  $t_3++$ ; break;
20 end switch

```

Fig. 4. Procedimento *Listening*.

Muito provavelmente, sempre que ocorre uma colisão en-

volvendo apenas estações NTR, os próximos eventos no canal serão seguidos pelo estado *continuing_idle*, devido ao intervalo de backoff selecionado ($CW_{min} = 7$, para a categoria de voz). No entanto, quando uma estação VTP-CSMA está envolvida na colisão, o próximo estado do canal será sempre *transmission* ($CW_{min} = CW_{max} = 0$). Portanto, o mecanismo de reset (procedimento *Initialization*) é ativado sempre que há RN consecutivas colisões, sem observar nenhum tempo livre do canal maior que t_2 ($aSIFSTime + 3 \times aSlotTime$) entre estas colisões.

IV. SIMULAÇÕES

O objetivo das simulações apresentadas nesta seção é comparar o suporte de tráfego de tempo-real (periódico) pela arquitetura VTP-CSMA *versus* o mecanismo EDCA (usando a categoria de maior prioridade - voz) do padrão IEEE 802.11e.

O modelo de simulação da arquitetura VTP-CSMA foi implementado através de *Redes de Petri estocásticas*, desenvolvido anteriormente para avaliar o comportamento temporal do mecanismo EDCA [18]. Utiliza-se um modelo de erros Semi-Markoviano para modelar as interferências no canal, que pode ser representado por dois estados: *Bom* e *Mau*. Assume-se que os erros são independentes, com uma taxa fixa em cada estado. A parametrização do modelo de erros é baseada em [19], onde são apresentados valores representativos para um ambiente industrial. O tempo de permanência em cada um dos estados é distribuído através de uma função *log-normal*. A duração média de permanência no estado *Bom* é de 65ms, e de 10ms no estado *Mau*. O coeficiente de variação (CoV) para o estado *Mau* foi fixado em 10, e em 20 para o estado *Bom*. Os parâmetros representam um canal de comunicação com muitas interferências, onde a probabilidade em regime estacionário de encontrar o canal no estado *Mau* é de aproximadamente 13.3%. Dois conjuntos de simulações são avaliados, diferindo nas respectivas taxas de erros por bit (BER). O primeiro define uma taxa média de BER igual a 10^{-4} , enquanto que o segundo conjunto define que não ocorrem erros.

A. Cenário

O cenário de simulação apresentado neste artigo pretende avaliar o impacto do tráfego NTR (gerado por estações padrão IEEE 802.11e) sobre o tráfego TR. Então, considera-se um ambiente de comunicação aberto, onde estações NTR compartilham o meio de comunicação com um conjunto de estações de TR. As estações de TR transmitem somente tráfego de TR, ao passo que as estações NTR transmitem três tipos de tráfego: voz (VO), vídeo (VI) e *background* (BK).

Basicamente, dois cenários são analisados. O primeiro, chamado *small population*, considera 10 estações NTR operando na mesma frequência de transmissão que 10 estações TR. Com a finalidade de comparar o mecanismo VTP-CSMA com a categoria de maior prioridade do mecanismo EDCA, a transmissão do tráfego de TR é avaliado quando as estações implementam o mecanismo EDCA (TR EDCA) ou VTP-CSMA (TR VTP-CSMA). O segundo, o cenário *large population*, aumenta o número de estações NTR para 40. Cada

estação opera através do modo físico OFDM (*Orthogonal Frequency Division Multiplexing*) com uma taxa de transmissão de dados de 36 Mbps. Todos os demais parâmetros físicos utilizados nas simulações são baseados nos valores definidos na norma IEEE 802.11a PHY [5].

O tráfego de TR é caracterizado por fontes periódicas, com uma pequena variabilidade de *jitter*, modelado por uma distribuição normal com $\sigma/\mu \leq 1\%$ (σ é o desvio padrão e μ é o valor médio esperado). O tráfego NTR é modelado por fontes de tráfego seguindo a distribuição de Poisson. O número máximo de tentativas de transmissão foi definido para 4. O tamanho máximo da fila foi fixado para 50 posições. Todos os outros parâmetros relevantes utilizados na simulação são apresentados na Tabela I.

As mensagens geradas têm tamanho constante. Cada estação TR gera 500 pacotes/s com 45 bytes de dados, o que é equivalente a geração de 1 pacote a cada 2ms. Portanto, cada estação de TR impõe uma carga na rede constante de 180 kbits/s, a qual representa cerca de 0.5% da carga nominal da rede (sem considerar os cabeçalhos ao nível MAC e PHY). Para o caso das estações NTR, a carga imposta na rede varia de 5% até 95% da carga nominal da rede (36 Mbps). Cada estação NTR gera λ pacotes de voz, vídeo and *background* com a mesma taxa de transmissão, de maneira a impor a carga desejada na rede. Finalmente, para as simulações propostas, assume-se que não há mobilidade nem estações “escondidas” (*hidden stations*).

B. Resultados

Todos os resultados foram obtidos usando um intervalo de confiança de 95%. As métricas de desempenho analisadas incluem: taxa de perdas de pacotes, atraso médio e tamanho médio da fila. A taxa de perdas de pacotes representa o percentual de mensagens que são perdidas em cada fluxo de dados. O atraso médio é o tempo necessário para transmitir uma mensagem, medido desde o início de sua geração na camada de aplicação até o final da transmissão na estação receptora. O tamanho médio da fila representa a ocupação média do *buffer* de saída.

As comunicações de TR são normalmente avaliadas considerando cenários de pior caso (*worst-case*). No entanto, quando se utiliza redes com acesso ao meio probabilístico, a análise para o pior caso endereça casos raros. Estes casos são relevantes somente para aplicações críticas, mas altamente pessimistas quando se trata de comunicações típicas. Sabe-se que diversas aplicações são tolerantes no que diz respeito à perda de algumas metas-temporais (*deadlines*). Por exemplo, a transferência de um vídeo pode ser especificada para tolerar uma taxa máxima de perdas de 10%, quando as mensagens “perdidas” estão adequadamente espaçadas. Um outro exemplo relevante de aplicações tolerantes a perdas são os NCSs escalonados de acordo com o modelo (m,k)-firm [20], ou o suporte de aplicações de VoIP, onde um atraso médio abaixo de 150ms e um *jitter* médio < 50ms são aceitáveis para a maioria das aplicações [21]. Em um sistema de comunicação de TR, o atraso médio das mensagens deve ser mantido abaixo de suas metas-temporais, de forma a minimizar a

ocorrência das perdas destas metas. Por outro lado, a taxa de perdas de pacotes deve ser mantida a mais baixa possível. O tamanho médio da fila também é uma métrica importante, uma vez que mostra como o protocolo é capaz de enviar mensagens periódicas. Portanto, o uso de simulação para a análise de comunicação de TR em redes com o meio de acesso probabilístico é bem justificado quando se trata de aplicações tolerantes a perdas.

A Figura 5 ilustra o atraso médio das mensagens, considerando os cenários com e sem erros (*error-free* e *error-prone*). Por uma questão de simplicidade, apenas os valores das estações TR (TR EDCA ou TR VTP-CSMA) são apresentados. A Figura mostra que para uma carga externa na rede acima de 15%, as estações TR VTP-CSMA têm um atraso médio muito menor que as estações TR EDCA. Por exemplo, no cenário *large* para uma carga externa na rede de 55% (*error-free*), as estações VTP-CSMA levam em média 1.412ms para transmitir uma mensagem de TR, enquanto que as estações EDCA levam 18.161ms. Observa-se que, apesar dos valores do atraso médio serem quase constantes (3ms - 5ms) para o cenário com erros, o valor obtido é significativamente maior do que para o caso sem erros (1ms - 1.6ms). Este comportamento pode ser justificado por duas causas. Por um lado, sempre que a variável distribuída *ACo* torna-se inconsistente, mais cedo ou mais tarde, o mecanismo de re-inicialização será executado (linha 2, Fig. 2) para resincronizar o anel virtual. Este mecanismo é eficiente, pois, o atraso médio das mensagens permanecem quase constantes apesar da carga externa imposta por estações NTR. Por outro lado, a taxa de erros é altamente pessimista (40% de probabilidade de erros nos pacotes durante as rajadas de erros).

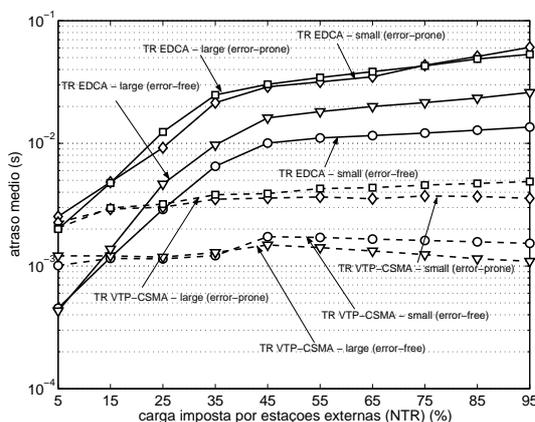


Fig. 5. Atraso médio dos pacotes - Small e Large cenários.

A taxa de perdas de mensagens também foi avaliada para os dois cenários. A Figura 6 apresenta os resultados para o cenário *small*, sendo que resultados similares foram obtidos no cenário *large*. Verifica-se, na Figura, que a taxa de perdas de mensagens das estações TR VTP-CSMA são significativamente menores que as taxas das estações TR EDCA. O principal motivo é que as mensagens geradas por diferentes estações de TR são globalmente serializadas pela arquitetura VTP-CSMA, evitando colisões entre os pacotes de TR. Por

TABELA I
DADOS DA SIMULAÇÃO.

Parâmetros	Estações TR		Estações NTR		
	EDCA	VTP-CSMA	VO	VI	BK
CW_{min}	7	0	7	15	31
CW_{max}	15	0	15	31	1023
AIFS _N	2	2	2	3	7
TXOP (ms)	1.504	1.504	1.504	3.008	0
Tamanho das mensagens - bytes	45	45	160	1280	1600

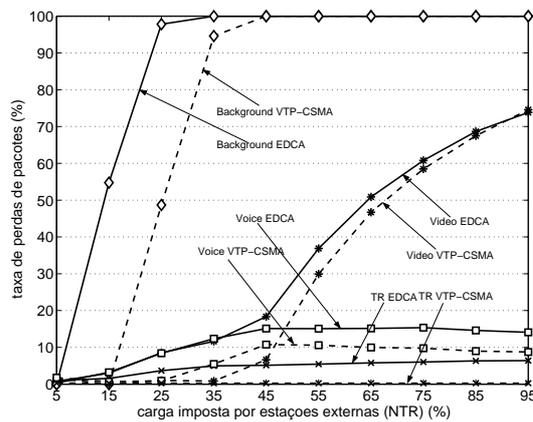


Fig. 6. Taxa de perdas de pacotes - Small population.

outro lado, a Figura 6 também ilustra que as estações TR VTP-CSMA têm uma influência menor sobre o comportamento das outras categorias de acesso do que as estações TR EDCA.

Um outro aspecto relevante em comunicações de TR é associado com o tamanho médio da fila. Constatou-se, que o tamanho médio da fila para as estações TR VTP-CSMA é quase constante, e menor que 1. Este resultado é consistente com o período de geração de pacotes ($2ms$) e o atraso médio ($1ms - 1.6ms$). Para o cenário com erros, para um atraso médio de ($3ms - 5ms$), o tamanho da fila é maior que 1 (haverá perdas frequentes das metas-temporais). Porém, é importante salientar que estes resultados são significativamente melhores que os resultados obtidos pela categoria de acesso de maior prioridade do mecanismo EDCA.

V. CONCLUSÕES

A principal contribuição deste artigo é que a solução proposta considera o pressuposto realista de que o meio de comunicação em redes sem fio é essencialmente aberto, ou seja, a arquitetura VTP-CSMA foi projetada para suportar a coexistência entre estações TR e NTR no mesmo domínio de comunicação.

Os resultados obtidos demonstram que o mecanismo proposto garante uma maior probabilidade de transmissão para as estações VTP-CSMA. Independentemente da carga imposta pelas estações NTR, o atraso médio das mensagens e o tamanho da fila são quase constantes para o tráfego de TR. Portanto, o tráfego NTR tem um impacto insignificante sobre o comportamento temporal do tráfego de TR. O que não acontece para o caso do suporte de comunicação de TR pela categoria de maior prioridade do mecanismo EDCA.

REFERÊNCIAS

- [1] A. Willig, K. Matheus, and A. Wolisz, "Wireless technology in industrial networks," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1130–51, 2005.
- [2] D. Miorandi, E. Uhlemann, S. Vitturi, and A. Willig, "Guest editorial: Special section on wireless technologies in factory and industrial automation, Part I," *IEEE Transactions on Industrial Informatics*, vol. 3, no. 2, pp. 95–98, 2007.
- [3] T. Sauter and F. Vasques, "Editorial: Special section on communication in automation," *IEEE Transactions on Industrial Informatics*, vol. 2, no. 2, pp. 73 – 77, May 2006.
- [4] L. Lo Bello, G. A. Kaczynski, and O. Mirabella, "Improving the real-time behavior of Ethernet networks using traffic smoothing," *IEEE Transactions on Industrial Informatics*, vol. 1, no. 3, pp. 151–161, 2005.
- [5] IEEE 802.11 working group, "IEEE Standard for Information Technology - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," ANSI/IEEE Std 802.11 (R2003), 2003.
- [6] IEEE 802.11e working group, "Amendment 8: Medium Access Control Quality of Service Enhancements," IEEE Std 802.11e-2005, 2005.
- [7] C. Casetti, C.-F. Chiasserini, M. Fiore, and M. Garetto, "Notes on the inefficiency on 802.11e HCCA," Dipartimento de Elettronica, Politecnico di Torino, 2005.
- [8] J. Son, I.-G. Lee, H.-J. Yoo, and S.-C. Park, "An effective polling scheme for IEEE 802.11e," *IEICE Transactions on Communications*, vol. E88-B, no. 12, pp. 4690 – 4693, 2005.
- [9] S.-C. Lo, G. Lee, and W.-T. Chen, "An efficient multipolling mechanism for IEEE 802.11 wireless LANs," *IEEE Transactions on Computers*, vol. 52, no. 6, pp. 764 – 68, 2003.
- [10] S. Lee, K. N. Ha, J. H. Park, and K. C. Lee, "NDIS-based virtual polling algorithm of IEEE 802.11b for guaranteeing the real-time requirements," *IECON (Industrial Electronics Conference)*, pp. 2427–32, 2005.
- [11] D. Miorandi and S. Vitturi, "Analysis of master-slave protocols for real-time industrial communications over IEEE 802.11 WLANs," *2nd IEEE International Conference on Industrial Informatics*, pp. 143 – 148, 2004.
- [12] M. Ergen, D. Lee, R. Sengupta, and P. Varaiya, "WTRP-wireless token ring protocol," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 6, pp. 1863 – 1881, 2004.
- [13] R.-G. Cheng, C.-Y. Wang, L.-H. Liao, and J.-S. Yang, "Ripple: a wireless token-passing protocol for multi-hop wireless mesh networks," *IEEE Communications Letters*, vol. 10, no. 2, pp. 123 – 5, 2006.
- [14] J. Sobrinho and A. Krishnakumar, "Quality-of-service in ad hoc carrier sense multiple access wireless networks," *IEEE J. Sel. Areas Commun*, vol. 17, no. 8, pp. 1353 – 68, 1999.
- [15] J. L. Sobrinho and A. Krishnakumar, "EQuB - Ethernet Quality of Service using black bursts," *Conference on Local Computer Networks*, pp. 286 – 296, 1998.
- [16] G.-H. Hwang and D.-H. Cho, "New access scheme for VoIP packets in IEEE 802.11e wireless LANs," *IEEE Communications Letters*, vol. 9, no. 7, pp. 667 – 669, 2005.
- [17] F. Carreiro, R. Moraes, J. A. Fonseca, and F. Vasques, "Real-time communication in unconstrained shared Ethernet networks: The virtual token-passing approach," *10th IEEE International Conf. on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, pp. 425–32, 2005.
- [18] R. Moraes, P. Portugal, and F. Vasques, "A stochastic petri net model for the simulation analysis of the IEEE 802.11e EDCA communication protocol," *10th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 38 – 45, 2006.
- [19] A. Willig, "Redundancy concepts to increase transmission reliability in wireless industrial LANs," *IEEE Transactions on Industrial Informatics*, vol. 1, no. 3, pp. 173 – 182, 2005.
- [20] M. Hamdaoui and P. Ramanathan, "A dynamic priority assignment technique for streams with (m,k)-firm deadlines," *IEEE Transactions on Computers*, vol. 44, no. 12, pp. 1443 – 51, 1995.
- [21] B. Goode, "Voice over Internet Protocol (VoIP)," *Proceedings of the IEEE*, vol. 90, no. 9, pp. 1495 – 1517, 2002.