

# Mecanismos de Fragmentação Adaptativa em Redes 802.11

Guilherme M. de Moura<sup>1</sup>, José F. de Rezende<sup>1</sup>, Guilherme D. G. Jaime<sup>2</sup> \*

<sup>1</sup>Grupo de Teleinformática e Automação  
PEE/COPPE/POLI  
Universidade Federal do Rio de Janeiro  
C.P. 68504 - CEP 21945-970  
Rio de Janeiro - RJ - Brasil  
<http://www.gta.ufrj.br>

<sup>2</sup>LAND  
PESC/COPPE  
Universidade Federal do Rio de Janeiro  
Rio de Janeiro - RJ - Brasil  
<http://www.land.ufrj.br>

{gmoura,rezende}@gta.ufrj.br, gdjaime@land.ufrj.br

**Resumo.** O padrão 802.11 do IEEE, protocolo de acesso ao meio utilizado em redes locais sem fio, é muito difundido atualmente. Sua eficiência, dessa forma, é fundamental para que aplicações desenvolvidas nesse ambiente possam ser largamente utilizadas. Neste artigo, foi feita uma análise do esquema de fragmentação utilizado em redes 802.11 e são propostos dois mecanismos adaptativos que tentam otimizar a eficiência do protocolo em função da taxa de perda do canal de comunicação. Os resultados mostram que os mecanismos melhoram a utilização do canal, aumentando a vazão da rede.

**Abstract.** IEEE 802.11 standard for media access control, used in wireless networks, is widely used nowadays. Therefore, protocol efficiency is fundamental so that developed applications can use it with no restrictions. In this article, we analyze the IEEE 802.11 fragmentation technique and propose two adaptive algorithms to optimize protocol efficiency, according to channel's bit error rate. The results show that the mechanisms improve channel utilization, increasing the network throughput.

## 1. Introdução

As redes sem fio ocupam uma posição de destaque atualmente. Diversos dispositivos pessoais, tais como celulares, computadores e portáteis, já são capazes de se comunicar em rede sem utilização de fios. Por permitirem a mobilidade, essas redes tornam possível o desenvolvimento de uma ampla gama de aplicações. Neste contexto, diferentes padrões e tecnologias de rede surgiram para dar suporte a esse novo paradigma. Dentre todos, o 802.11 do IEEE (*Institute of Electrical and Electronic Engineering*) foi o que obteve o maior sucesso comercial. Padronizado em 1997 e posteriormente revisado em 1999 [IEEE, 1999], o 802.11 define características da sub-camada de acesso ao meio, além de diferentes camadas físicas de alcance e taxas variadas, de acordo com a arquitetura da família de padrões IEEE 802. Além disso, devido a sua diversidade em termos de alcance e taxa de transmissão, o 802.11 tem sido utilizado em diversas áreas, tais como

---

\*Este trabalho foi realizado com recursos do CNPq, CAPES, FAPERJ e RNP/FINEP/FUNTTTEL.

aplicações residenciais, comerciais e metropolitanas, e até mesmo em suporte à telefonia celular 2G/3G.

Entretanto, existem vários problemas característicos das redes sem fio que o 802.11 deve contornar, tais como compartilhamento do meio e a alta taxa de erro associada ao canal de transmissão: o ar. Para combater o problema da taxa de erro associada ao meio, o 802.11 define um esquema de fragmentação de quadros. Assim, em situações onde a qualidade do canal é baixa, o transmissor pode obter um melhor desempenho diminuindo o tamanho de seus quadros, ou MSDU's (*MAC Service Data Unit*), para obter uma maior probabilidade de sucesso na transmissão. No entanto, o esquema básico de fragmentação é ineficiente, pois não oferece meios para obter um tamanho de fragmento que resulte em um aumento de performance do protocolo, dada a qualidade do canal, de forma a não acrescentar muita sobrecarga de controle.

Desta forma, a ineficiência da fragmentação impulsiona a pesquisa de técnicas adaptativas que adequem o mecanismo à qualidade do meio, mesmo para canais que variem suas condições em função do tempo. Assim, embora nenhum mecanismo adaptativo tenha sido padronizado, algumas propostas foram feitas, usando abordagens heurísticas e modelagens matemáticas, que tentam atacar os principais problemas que prejudicam o desempenho do protocolo. Em [H. Sharif, 2000], são propostos dois mecanismos, baseados na janela de contenção do 802.11. Nesses mecanismos, o tamanho do fragmento utilizado varia exponencialmente, de forma a encontrar o limiar adequado às condições do canal. Em [Tourrilhes, 2000], é proposto um mecanismo que leva em consideração o *dwell size*, que é o tempo em que uma estação realizando saltos em frequência (*frequency hopping*) fica em uma determinada faixa antes de realizar um novo salto. Nessa abordagem, o tamanho do limiar de fragmentação é variado de acordo com o tempo que a estação ainda dispõe naquela faixa de frequência. Em [Yeong-Hyeon Kwon e Oh, 2003], é proposto um mecanismo que busca minimizar o tempo de transmissão dos quadros, a partir da estimação da taxa de erro associada ao canal.

Neste trabalho são propostos dois mecanismos de fragmentação adaptativa que tentam otimizar a eficiência do protocolo, em função das condições do canal. Na primeira abordagem, é utilizada uma função eficiência do meio, que de posse da probabilidade de erro calculada pelo estimador implementado, obtém o tamanho de fragmento que maximiza a eficiência do protocolo. A segunda abordagem é baseada num mecanismo que ajusta o tamanho do fragmento utilizado, usando um algoritmo AIMD (*additive increase-multiplicative decrease*), sempre que o limite de tempo de transmissão de uma rajada se esgota.

Este trabalho está organizado da seguinte forma. Na Seção 2, é feita uma breve descrição da sub-camada de acesso ao meio do 802.11, além do seu mecanismo de fragmentação. Na Seção 3, são propostos dois mecanismos adaptativos de fragmentação que buscam otimizar a eficiência do protocolo de acordo com as condições do canal. Na Seção 4, é apresentado o modelo de erro de Gilbert-Elliot otimizado, utilizado nas simulações. Na Seção 5, são apresentados os cenários simulados e a comparação dos resultados dos mecanismos propostos. Por fim, na Seção 6, é apresentada a conclusão do trabalho, além de considerações sobre trabalhos futuros.

## **2. Sub-camada de acesso ao meio**

O 802.11 possui dois métodos de acesso ao meio. O método de acesso fundamental, denominado Função de Coordenação Distribuída (*Distributed Coordination Function - DCF*), é distribuído e pode ser utilizado tanto na configuração infra-estruturada quanto na configuração *ad-hoc*. O segundo método de acesso, denominado Função de Coordenação

Centralizada (*Point Coordination Function* - PCF), é opcional e centralizado, podendo ser apenas utilizado em redes estruturadas. Neste trabalho, apenas as redes *ad-hoc* foram analisadas, e, dessa forma, o DCF será explicado melhor nas seguintes subseções.

## 2.1. Função de coordenação distribuída

Nas redes *ad-hoc*, a ausência de um elemento centralizador torna o acesso ao meio uma tarefa complexa. Além disso, como o canal utilizado para propagação do sinal nas redes sem fio é, por natureza, um meio compartilhado, o método de acesso DCF utiliza a técnica de acesso CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*), ou Acesso Múltiplo com Detecção de Portadora Evitando Colisões. Assim, se uma estação quiser transmitir um quadro de dados, deverá esperar um período de tempo DIFS (*DCF Interframe Space*) e, se o meio continuar livre após esse período, um tempo de resguardo, ou *backoff*, deve ser iniciado. Esse período de resguardo é aleatório, variando de 0 até um valor dado pela janela de contenção *CW* (*Contention Window*) vezes o tempo do *slot* (*Slot Time*) de  $20 \mu s$ , e tem o objetivo de minimizar o número de colisões na rede. Após esse período de *backoff*, a estação pode enviar o quadro. Como o CSMA/CA não detecta colisões na rede, todos os quadros emitidos por um transmissor devem ser reconhecidos positivamente com um envio de um ACK. Após receber o quadro de dados, o receptor aguarda então um período de tempo SIFS (*Short Interframe Space*) e envia de volta ao transmissor o ACK relativo ao quadro recebido. A conquista do meio é automaticamente garantida ao receptor, pois o período de tempo SIFS é menor que o DIFS, que todas as estações da rede devem esperar antes de transmitir um quadro de dados. Quando o transmissor finalmente recebe o reconhecimento ACK, ele tem certeza de que a transmissão foi realizada com sucesso. Além disso, o 802.11 define um intervalo de tempo adicional utilizado no modo DCF sempre que uma estação recebe um quadro com erros: o EIFS (*Extended Interframe Space*). O EIFS é maior do que o DIFS e é calculado como:

$$EIFS = DIFS + SIFS + T_{ack}$$

Onde  $T_{ack}$  é o tempo gasto para uma estação enviar o quadro de reconhecimento na taxa básica de transmissão. A Figura 1, ilustra o esquema básico de acesso ao meio do método DCF.

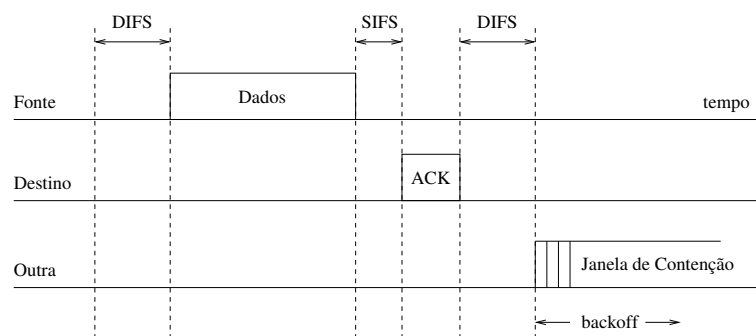
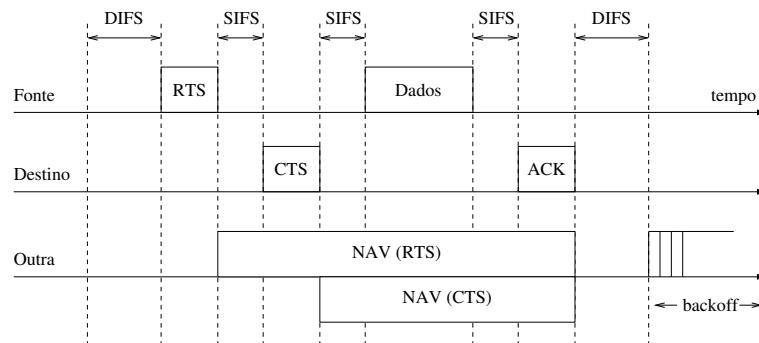


Figura 1: Esquema básico do DCF.

O modo DCF tem ainda um outro meio para evitar colisões, que utiliza dois quadros de controle especiais que fazem a reserva do meio. O primeiro é o RTS (*Request To Send*), que é enviado inicialmente pelo transmissor, após um período DIFS, no momento em que ele deseja iniciar uma comunicação. O RTS é um quadro pequeno (20 bytes), e dessa forma tem menos probabilidade de ser perdido. Além disso, o RTS carrega a duração *Duration/ID* do 802.11 o tempo que a transmissão demorará para ser efetivada.

Assim, todas as estações que captarem o quadro RTS no meio, iniciarão seus temporizadores NAV (*Network Allocation Vector*) com o tempo armazenado no campo *Duration/ID* do RTS. Esse valor indica o período de tempo no qual a estação não poderá transmitir, pois o meio foi reservado para a transmissão em questão. Ao receber o RTS, a estação de destino deverá responder com um CTS (*Clear To Send*), que confirma positivamente o recebimento do RTS ao transmissor e carrega o tempo restante da reserva do meio para as estações que não foram atingidas pelo RTS do transmissor. O mecanismo RTS/CTS, além de evitar colisões no meio, resolve com eficiência o problema do terminal escondido das redes sem fio. Assim, após essa transação inicial, a transmissão pode ser efetivamente realizada. A Figura 2 ilustra o esquema básico de acesso ao meio com o mecanismo RTS/CTS de reserva de meio.



**Figura 2: Método de acesso DCF com RTS/CTS.**

## 2.2. Fragmentação em redes 802.11

O processo de dividir uma MSDU (*MAC Service Data Unit*) em quadros menores é chamado fragmentação. A fragmentação divide o quadro original, aumentando a probabilidade do êxito de uma comunicação em situações onde as condições do canal limitam a transmissão de quadros maiores. No receptor, os quadros fragmentados são ordenados e remontados, para que o processo seja transparente às camadas superiores.

Nas redes 802.11, o processo de fragmentação acontece quando um quadro a ser transmitido tem tamanho superior ao `aFragmentationThreshold`, ou limiar de fragmentação. Os quadros com esta característica devem ser quebrados em blocos não maiores do que `aFragmentationThreshold`, sem levar em consideração o cabeçalho do quadro. Em seguida, os quadros que são obtidos do processo de fragmentação devem ser transmitidos independentemente, com reconhecimentos positivos separados, o que permite que as retransmissões ocorram em nível de fragmento. No entanto, com exceção do primeiro fragmento, o período de tempo necessário para iniciar a transmissão é SIFS, o que garante o monopólio do meio e é chamado de modo rajada (*burst mode*). A transmissão em modo rajada continua até que a transmissão de todos os fragmentos termine ou o tempo máximo de atenção do canal, dado por `aMaxTransmitLifetime`, seja alcançado. Neste último caso, o transmissor interrompe a comunicação e descarta o pacote que estava transmitindo. Assim, após descartar o quadro, o transmissor deve disputar o meio novamente para que uma nova janela de atenção de rajada seja conquistada.

Além disso, o 802.11 permite que o mecanismo de reserva RTS/CTS seja usado para transmissão de fragmentos. Porém, a transação RTS/CTS só ocorre antes do envio do primeiro fragmento. Assim, o RTS reserva o meio para que o primeiro fragmento possa ser transmitido. A partir daí, cada fragmento deverá realizar a reserva de tempo para o fragmento que o sucederá na rajada. Esse processo é chamado de RTS/CTS virtual. No fim da rajada, o reconhecimento do último fragmento deverá carregar 0 no seu campo *Duration/ID*, confirmando o fim da reserva do meio.

### 3. Propostas

Nesta seção, são propostos dois mecanismos adaptativos que buscam otimizar a utilização do canal de comunicação em redes 802.11. O objetivo dos mecanismos é encontrar um limiar de fragmentação, ou o tamanho do fragmento utilizado durante as transmissões, dadas as condições do meio, para obter melhoras na eficiência do protocolo. Para isto, é proposta uma função eficiência  $\eta$ , que a partir de uma taxa de erro associada ao canal, calculada por meio de um estimador implementado, obtém o tamanho ótimo do fragmento a ser utilizado. De forma a diferenciar os mecanismos empregados neste trabalho, a primeira proposta, baseada na função eficiência, será chamada no resto deste artigo de primeira proposta, ou primeiro mecanismo. Da mesma forma, a segunda proposta apresentada, baseada na técnica *additive increase-multiplicative decrease*, será chamada simplesmente de segunda proposta, ou segundo mecanismo.

#### 3.1. Estimador

O estimador implementado baseia-se em uma propriedade característica das modulações digitais, onde a partir de uma relação sinal-ruído, uma modulação e uma codificação utilizada, pode-se estimar a probabilidade de erro de bit do meio. Para as redes 802.11 a taxas a partir de 2 Mbps, que utilizam modulação QPSK (*Quadrature Phase Shift-Keying*), a probabilidade de erro de símbolo é dada por [Haykin, 2000]:

$$P_e \approx \operatorname{erfc} \left( \sqrt{\frac{E}{2N_0}} \right) \quad (1)$$

Onde  $E$  representa a potência de transmissão do símbolo e  $N_0$  a densidade espectral do ruído.

Desta forma, implementou-se um estimador da probabilidade de erro baseado na hipótese da medição da relação sinal-ruído. Na verdade, a cada transmissão de quadro, o transmissor amostra a probabilidade de perda do meio e a armazena. No final de um período de tempo dado por um temporizador implementado na camada de acesso ao meio, o estimador calcula a média da probabilidade de erro de bit e entrega a camada MAC para que esse dado possa ser utilizado.

#### 3.2. Função eficiência

A função proposta ilustra matematicamente a eficiência da vazão obtida na rede. Para transmitir um fragmento de dados de  $L$  bits durante uma rajada em uma rede 802.11, o transmissor deve esperar um período de tempo SIFS. Em seguida, depois da transmissão do fragmento, o receptor envia um reconhecimento positivo ACK, após ter esperado por um período de tempo SIFS. Dessa forma, a função eficiência pode ser definida como:

$$\eta = \frac{\frac{L}{G + \frac{L}{t_d}}}{t_d} \quad (2)$$

Onde  $t_d$  é a taxa de transmissão de dados utilizada e  $G$  é o tempo de sobrecarga ou *overhead* do protocolo, dado pelos tempos de espaçamento entre quadros e os bits de controle (ACK + cabeçalho), que são transmitidos na taxa básica  $t_b$ . Assim, de acordo com a norma,  $G$  pode ser expresso como:

$$G = SIFS + T_{preamble} + \frac{S_{header}}{t_b} + SIFS + T_{preamble} + \frac{S_{ack}}{t_b} \quad (3)$$

Onde  $T_{preamble}$  é o tempo gasto pelo preâmbulo dos quadros, transmitidos na taxa básica, e  $S_{ack}$  e  $S_{header}$  são o tamanho do ACK e do cabeçalho do fragmento, respectivamente. Na

verdade, este cálculo de  $G$  é aproximado, pois deveria levar em consideração a probabilidade de perda do meio e o tempo médio para que uma transmissão pudesse ser realizada. No entanto, este resultado mostrou-se uma boa aproximação em todos os testes realizados e foi mantido, por essa razão, desta forma. Na tabela 1, são apresentados os valores necessários para o cálculo de  $G$ , obtidos da norma.

**Tabela 1: Valores obtidos da norma.**

SIFS	10 $\mu$ s
$T_{preamble}$	192 $\mu$ s
$S_{header}$	272 bits
$S_{ack}$	112 bits

Assim,

$$G = 404.10^{-6} + \frac{384}{t_b}$$

Entretanto, em um canal real, onde a taxa de erro impede que todos os bits transmitidos cheguem com sucesso ao receptor, essa vazão é ainda menor. Para um quadro de dados de tamanho  $L$ , e considerando a transmissão de cada bit como um evento independente de Bernoulli com probabilidade de perda  $p_e$ , a probabilidade sucesso da transmissão é expressa por:

$$P[sucesso] = (1 - p_e)^L \quad (4)$$

Assim, levando em consideração que os bits de controle são transmitidos em uma taxa mais baixa, pode-se considerar a vazão obtida em uma rede com taxa de erro de dados dada por  $p_e$  como:

$$\eta = (1 - p_e)^L \cdot \left( \frac{\frac{L}{G + \frac{L}{t_d}}}{t_d} \right) \quad (5)$$

Na figura 3, são mostradas curvas da eficiência do protocolo para várias taxas de erro do canal, onde pode-se verificar que, para cada erro associado, existe um tamanho de quadro que maximiza a função.

Desta forma, o tamanho do fragmento que otimiza a eficiência do protocolo em função da taxa de erro do canal pode ser calculado analiticamente por derivação simples. Assim, fazendo:

$$\frac{d\eta}{dL} = 0$$

Obtem-se a equação:

$$L^2 \cdot \ln(1 - p_e) + L \cdot t_d \cdot G \cdot \ln(1 - p_e) + t_d \cdot G = 0 \quad (6)$$

Que gera dois valores possíveis para tamanhos de quadro que otimizam a eficiência do protocolo. No entanto pode-se mostrar que, para a equação fornecida, uma das raízes será sempre negativa. Desta forma, apenas um tamanho é obtido para cada probabilidade de erro. Além disso, como para valores pequenos da probabilidade de erro  $\ln(1 - p_e) \approx -p_e$ , obtem-se a equação final do tamanho do fragmento em função da taxa de erro do canal

$$L^2 + L \cdot t_d \cdot G - t_d \cdot G / p_e = 0 \quad (7)$$

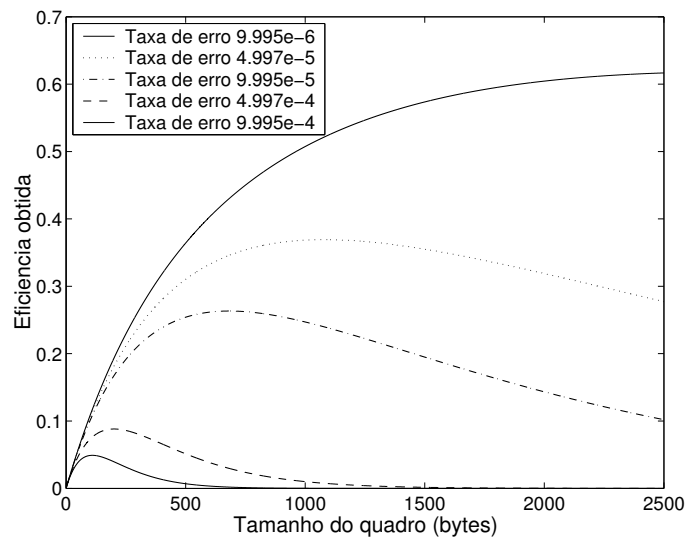


Figura 3: Função eficiência, para várias taxas de erro.

### 3.3. Algoritmos propostos

O primeiro mecanismo proposto é baseado na função eficiência obtida na equação 5 . Assim, à medida que transmite os quadros, o transmissor calcula o tamanho ótimo do fragmento baseado na taxa de erro obtida do estimador utilizado. O cálculo do fragmento é feito periodicamente, e é controlado por um temporizador `adaptiveTimeout` que regula o período. Na Figura 4, é apresentado o pseudo-código que ilustra o mecanismo proposto, onde `adapTimeout` é o parâmetro com o qual programa-se o temporizador.

```

Início {
    Programa temporizador com adapTimeout ;
    Estima probabilidade de erro do periodo ;
}

Quando temporizador estourar {
    Calcula tamanho do fragmento otimo baseado no erro estimado ;
    Volta ao inicio ;
}

```

Figura 4: Pseudo código do primeiro mecanismo.

O segundo mecanismo é baseado no esquema *additive increase-multiplicative decrease*, utilizado no controle da janela de congestionamento do TCP (*Transmission Control Protocol*). Esse mecanismo utiliza o tempo máximo de atenção do meio durante o modo rajada, dado por `aMaxTransmitLifetime`, para tomar suas decisões. Nesse mecanismo, três variáveis são utilizadas: *alfa*, *beta* e *gama*. A variável *alfa* é responsável pelo decrescimento multiplicativo do fragmento. Dessa forma, sempre que o transmissor terminar o modo rajada devido ao estouro do temporizador dado por `aMaxTransmitLifetime`, ele divide o tamanho do fragmento utilizado por *alfa*, para obter o próximo tamanho a ser usado. A variável *beta*, por sua vez, diz quantos quadros consecutivos terão que ser transmitidos com sucesso para que o transmissor incremente o tamanho do fragmento utilizado em *gama*. A figura 5 ilustra o comportamento do mecanismo, onde *fragthreshhold* é o limiar de fragmentação, ou seja, o tamanho dos fragmentos usados durante uma rajada de fragmentos.

```

Quando temporizador de atencao do meio estoura {
    fragthreshold = fragthreshold / alfa ;
}

```

```

Ao transmitir beta quadros consecutivos com sucesso {
    fragthreshold = fragthreshold + gama ;
}

```

**Figura 5: Pseudo código do segundo mecanismo.**

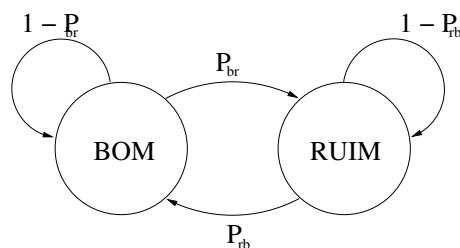
#### 4. Modelo de erros

Nesta seção será apresentado brevemente o modelo de erros utilizado para avaliação do desempenho dos mecanismos propostos.

A escolha do modelo de Gilbert-Elliot está fundamentada em trabalhos da literatura [Haowei Bai e Atiquzzaman, 2003] que mostram que o processo de erros em redes sem fio pode ser aproximado por uma cadeia de Markov discreta de dois estados, onde cada transição representa a transmissão de um bit. Um dos estados, chamado BOM, representa o canal em boas condições de transmissão e o outro estado, RUIM, representa um período de condições desfavoráveis. Tipicamente, os períodos de permanência no estado RUIM são bem mais curtos do que os do outro estado.

##### 4.1. Modelo de Gilbert-Elliot

A Figura 6 ilustra a representação discreta do modelo de Gilbert-Elliot. Neste exemplo, os parâmetros  $P_{br}$  e  $P_{rb}$  representam as probabilidades de passagem do estado BOM para o estado RUIM e do estado RUIM para o estado BOM, a cada transmissão de um bit. Além disso, dois parâmetros  $P_{ruim}$  e  $P_{bom}$  representam as probabilidades de perda de um bit no estado RUIM e no estado BOM, respectivamente. Os parâmetros utilizados neste trabalho são baseados nos encontrados em [Pravin Bhagwat e Tripathi, 1997] e [Wang e Moayeri., 1995], onde para todos os fins, considera-se erro a recepção errônea de pelo menos um bit.



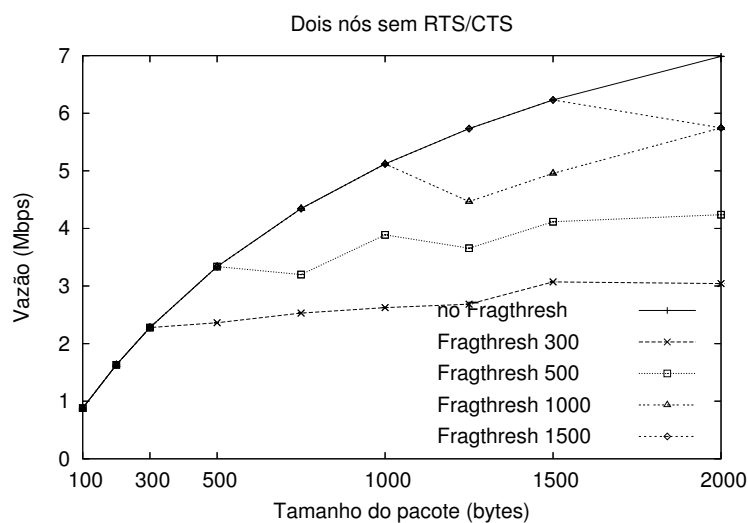
**Figura 6: Representação discreta do modelo de erros de Gilbert-Elliot.**

#### 5. Simulações e resultados

Os testes realizados simulam transmissões em uma rede 802.11b, no ns-2 (*Network Simulator*) versão 2.26. Para isso, foi implementada a parte de fragmentação no módulo do 802.11, além do modelo de erro do canal. Para todos os testes, os valores da taxa básica de transmissão  $t_b$  e da taxa de dados  $t_d$  foram, respectivamente, 2 Mbps e 11 Mbps. O modelo de atenuação utilizado foi o *Two Ray Ground* e o modelo de erro, como já explicado na subseção 4.1, foi o Gilbert-Elliot. Para a transmissão de dados, foi escolhido um tráfego de taxa constante (CBR - *Constant Bit Rate*) a 11 Mbps sobre UDP (*User*

*Datagram Protocol*), e o protocolo de roteamento utilizado foi o AODV (*Ad-hoc On Demand Distance Vector*). Além disso, os valores utilizados para os espaçamentos SIFS e DIFS entre quadros foram, respectivamente,  $10 \mu s$  e  $50 \mu s$ , de acordo com a norma, e o intervalo de confiança é de 95%.

Na Figura 7 foi avaliada, a título de ilustração, a vazão em uma rede de dois nós e sem a utilização do modelo de erro, em função do tamanho do pacote. Verifica-se, de acordo com o esperado, que a vazão da rede aumenta a medida que o limiar de fragmentação cresce. Neste caso, nota-se que o aumento do tamanho do quadro transmitido implica em um maior aproveitamento do meio. No caso onde não se usa fragmentação, os quadros são transmitidos esperando apenas um período DIFS, e por isso a vazão é máxima. Quanto menor o tamanho do fragmento, mais períodos SIFS a fonte deverá esperar antes de concluir a transmissão e, conseqüentemente, menor a vazão da rede.



**Figura 7: Vazão em função do tamanho do pacote, sem modelo de erro.**

Em seguida, avaliou-se a vazão da rede em função da taxa de erro do meio, como ilustrado na Figura 8. Verifica-se, como esperado, que existe um tamanho de fragmento ótimo que maximiza a vazão da rede para cada intervalo de erro. Observa-se também que os tamanhos ótimos obtidos através de simulação da curva de vazão se aproximam dos valores obtidos da função eficiência proposta. A aproximação se deve às considerações feitas na obtenção da função e a granularidade das figuras obtidas. Assim, quanto pior a qualidade do canal, menor o fragmento cuja a probabilidade de sucesso é ótima.

Por último, comparou-se o desempenho dos mecanismos. Para a comparação, implementou-se o mecanismo proposto por [H. Sharif, 2000], referido como terceiro mecanismo, que se baseia na janela de contenção do 802.11 para variar o tamanho dos fragmentos usados na rajada. Além disso, utilizou-se no primeiro mecanismo um valor de 3 segundos para o tamanho da janela de estimação de erro `adaptTimeout` e no segundo mecanismo valores 1.5, 2 e 100 para os parâmetros *alfa*, *beta* e *gama*, respectivamente. Na Figura 9, é apresentada a curva de vazão em função da taxa de erro do meio, onde observa-se que os mecanismos propostos obtêm melhor desempenho.

Para mostrar o ganho obtido com os mecanismos propostos, é ilustrado na Figura 10 o desempenho dos mecanismos, normalizados pela vazão do modo sem fragmentação. Assim, observa-se que o ganho é maior em regiões de alta taxa de erro do meio, onde a perda de quadros é maior e a vazão obtida no modo sem fragmentação é praticamente

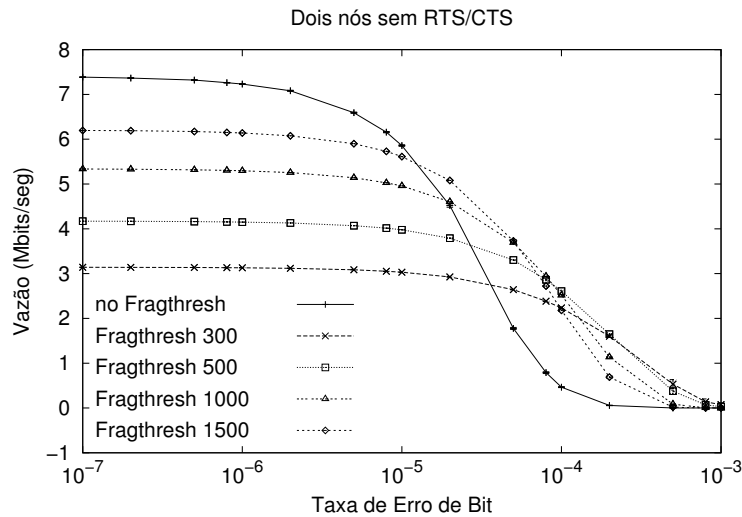


Figura 8: Vazão em função da taxa de erro, com pacotes de 2000 bytes.

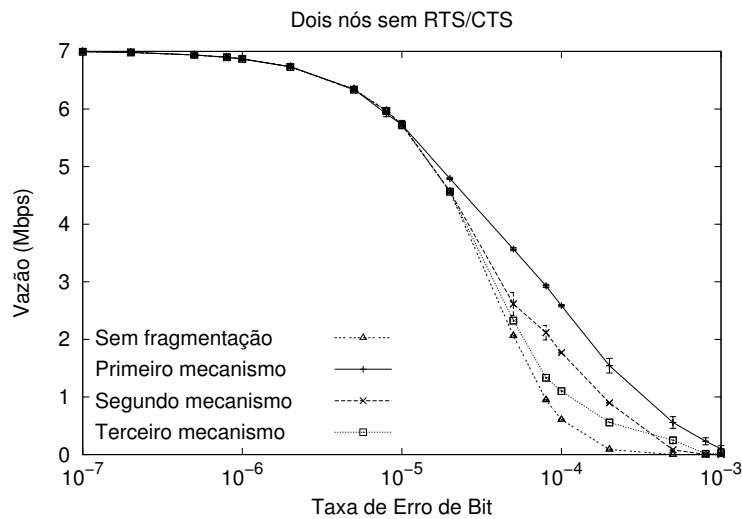


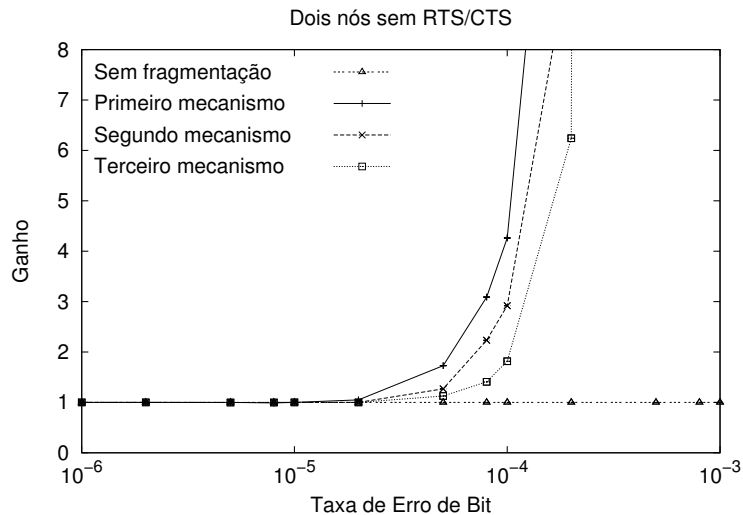
Figura 9: Comparação dos mecanismos em função da taxa de erro.

zero. Assim, os mecanismos propostos mostraram-se mais robustos para utilização em condições mais críticas.

## 6. Conclusões

O objetivo deste trabalho foi propor e avaliar dois mecanismos de fragmentação adaptativa em redes 802.11. O primeiro mecanismo utiliza a função eficiência proposta neste trabalho, que calcula o fragmento ótimo de acordo com a taxa de perda do canal, obtida do estimador implementado. O segundo mecanismo é baseado na técnica *additive increase-multiplicative decrease* para ajuste automático do tamanho de fragmento, e usa parâmetros da camada de acesso ao meio para tomar suas decisões, tais como quadros consecutivos transmitidos com sucesso e tempo máximo para transmitir todos os fragmentos.

Para simular o meio de transmissão das redes 802.11, utilizou-se o modelo de erros de Gilbert-Elliot, utilizado para modelar ambientes *indoor*. Os testes realizados por simulações provam que os mecanismos propostos melhoram a eficiência do protocolo, aumentando a vazão da rede em condições onde a ausência de fragmentação ou mesmo



**Figura 10: Ganho de vazão dos mecanismos, em função da taxa de erro.**

a utilização de fragmentos de tamanho fixo têm um baixo rendimento. A melhoria do desempenho foi mais acentuada em regiões com alta taxa de erro, onde o número de quadros perdidos é significativo. Para regiões com baixa taxa de erro, o desempenho mostrou-se o mesmo do 802.11.

Assim, conclui-se que a utilização dos mecanismos propostos melhora o desempenho das redes 802.11. Para trabalhos futuros, pretende-se avaliar de forma exaustiva a variação dos parâmetros *alfa*, *beta* e *gama* no segundo mecanismo e o compromisso da janela de tempo em que o primeiro mecanismo estima o erro. Neste último caso pretende-se estudar um mecanismo de ajuste automático de janela, que varia em função da variação da taxa de erro do meio.

## Referências

- H. Sharif, S. C. (2000). Adaptive approaches to enhance throughput of IEEE 802.11 wireless LAN with bursty channel. Em *IEEE LCN*.
- Haowei Bai, H. A. e Atiquzzaman, M. (2003). Error modeling schemes for fading channels in wireless communications: A survey. Em *IEEE-COMSOC*. IEEE.
- Haykin, S. (2000). *Communication Systems*. Wiley, quarta edição.
- IEEE (1999). *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE.
- Pravin Bhagwat, Partha Bhattacharya, A. J. e Tripathi, S. K. (1997). Using channel state dependent packet scheduling to improve TCP throughput over wireless LANs. Em *Wireless Networks*, volume 3, (1), páginas 91–102. ACM.
- Tourrilhes, J. (2000). Dwell adaptive fragmentation: how to cope with short dwells required by multimedia wireless LANs. Em *IEEE Globecom*.
- Wang, H. e Moayeri, N. (1995). Finite state Markov channel - a useful model for radio communications channels. Em *IEEE Transactions on Vehicular Technology*, volume 44, (1), páginas 163–171. ACM.
- Yeong-Hyeon Kwon, Dong-Jo Park, S.-M. L. e Oh, M.-K. (2003). Determination of fragmentation size in wireless system. *IEEE*.