

Fragmentação e Análise de Desempenho em Redes 802.11

Guilherme M. de Moura¹, José F. de Rezende¹, Guilherme D. Jaime² *

¹Grupo 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>

²LAND
PESC/COPPE/POLI
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, de acordo com o modelo utilizado, que a eficiência do protocolo varia sensivelmente em função do tamanho do fragmento utilizado.*

Palavras-chave: Redes Sem Fio, IEEE 802.11, Otimização.

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 IEEE 802.11 fragmentation technique and propose two adaptive algorithms to optimize channel utilization, according to its conditions. The results show that, depending on the quality, protocol efficiency varies with used fragment size.*

Keywords: Wireless networks, IEEE 802.11, Optimization.

1. Introdução

As redes sem fio ocupam uma posição de destaque atualmente. Com relação aos seus antecessores cabeados, o ganho de mobilidade tornou possível o desenvolvimento de uma ampla gama de aplicações, de forma que diversos dispositivos pessoais, tais como celulares, computadores e portáteis, já são capazes de se comunicar em rede sem utilização de fios. Neste contexto, diferentes padrões e tecnologias surgiram para dar suporte a esse novo paradigma e, dentre todos, o 802.11 do IEEE (*Institute of Electrical and Electronic Engineering*) se destacou. Padronizado em 1997 e posteriormente revisado em

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

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 telefonia celular 2G/3G, aplicações comerciais, residenciais e metropolitanas.

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 acrescenta muita sobrecarga de controle e não oferece meios para obter um tamanho de fragmento que resulte em um aumento de performance do protocolo, dada a qualidade do canal.

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, como no caso do *wireless*. Assim, embora nenhum mecanismo adaptativo tenha sido padronizado, algumas propostas foram feitas com o intuito de otimizar determinadas características do meio, com abordagens heurísticas e modelagens matemáticas, tentando atacar os principais problemas que prejudicam o desempenho do 802.11. Em [H. Sharif, 2000], são propostos dois mecanismos, baseados na janela de contenção do 802.11. Nesse mecanismo, o tamanho do fragmento utilizado varia de forma exponencial, de forma a encontrar o limiar adequado às condições do canal. Em [Tourrilhes, 2000], é proposto um mecanismo que leva em consideração o *dwelling size*, que é o tempo em que uma estação realizando saltos em frequência fica em uma determinada faixa antes de realizar um novo salto. Nesta 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 a latência do meio, 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 usado em [Yeong-Hyeon Kwon e Oh, 2003], obtém o tamanho de fragmento que maximiza a eficiência do protocolo. A segunda abordagem é baseada no mecanismo *additive increase-multiplicative decrease*, usado para o controle da janela deslizante do TCP.

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 mecanismo de fragmentação definido. Na Seção 3, são propostos dois mecanismos adaptativos de fragmentação que buscam otimizar a eficiência do protocolo, baseados na qualidade do canal. Na Seção 4 é apresentado o modelo de erro de *Gilbert-Elliot*, 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 qualquer 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. Só após o período de *backoff* que a estação poderá enviar o quadro. Como o CSMA/CA não detecta colisões na rede, todos os quadros emitidos por um transmissor qualquer 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.

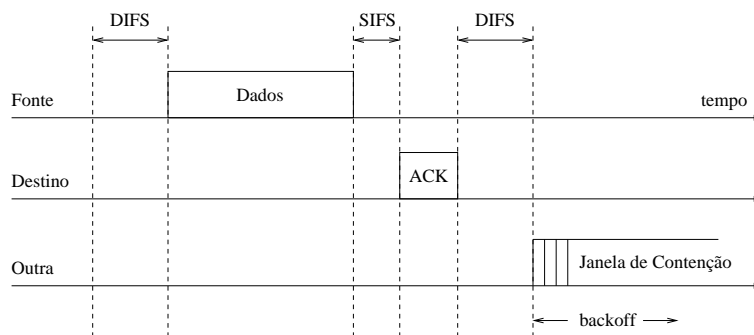


Figura 1: Esquema básico do DCF.

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 o mesmo 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 no campo *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* 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.

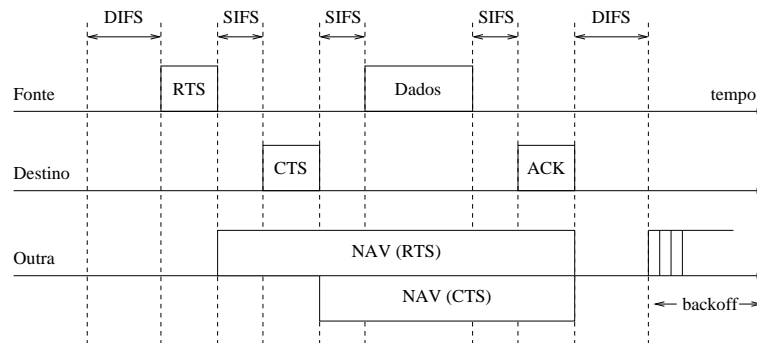


Figura 2: Método de acesso DCF com RTS/CTS.

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. Na figura 3, são ilustradas curvas de vazão da rede, em função da taxa de erro de bit do meio, para varios tamanhos de fragmentos. No receptor, os quadros fragmentados são ordenados e desfragmentados, 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 essa característica devem ser quebrados em blocos não maiores do que `aFragmentationThreshold`, não levando 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 o tempo máximo de atenção do canal, dado por `aMaxTransmitLifetime`, seja alcançado. Neste caso, o transmissor interrompe a comunicação, e descarta o pacote que estava transmitindo. Assim, para transmitir os quadros subseqüentes, o transmissor deve disputar o meio novamente para que uma nova janela de atenção 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

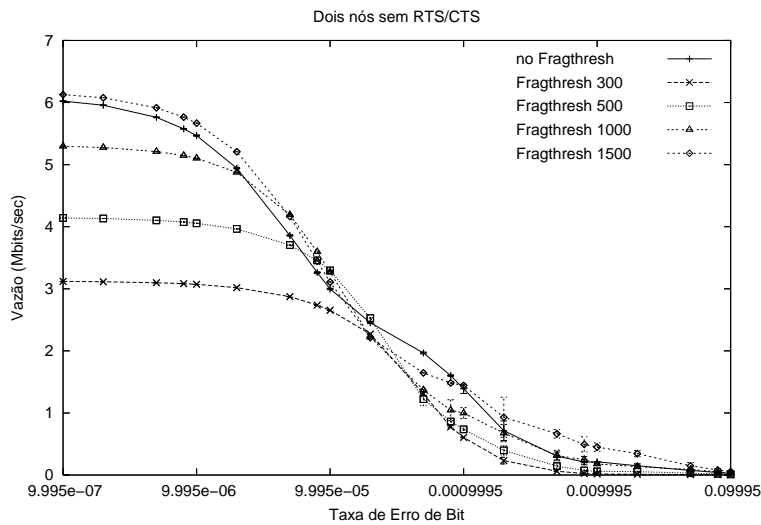


Figura 3: .

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

Neste artigo, são propostos dois mecanismos adaptativos que buscam otimizar a utilização do canal de comunicação em redes 802.11. O objetivo dos mecanismos é encontrar o melhor limiar de fragmentação, ou seja, 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 η , que a partir de uma taxa de erro associada ao canal, calculada por meio de um estimador [Yeong-Hyeon Kwon e Oh, 2003], obtém o tamanho ótimo do fragmento a ser utilizado.

3.1. Função eficiência

A função proposta ilustra matematicamente a eficiência da vazão obtida na rede. Para transmitir um quadro 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:

$$\eta = \frac{\frac{L}{G + \frac{L}{t_d}}}{t_d} \quad (1)$$

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:

$$G = SIFS + T_{preamble} + \frac{S_{header}}{t_b} + SIFS + T_{preamble} + \frac{S_{ack}}{t_b} \quad (2)$$

Tabela 1: Valores obtidos da norma.

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

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 quadro, respectivamente. Na tabela 1, são apresentados os valores necessários para o cálculo de G , obtidos da norma.

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. Considerando a transmissão de um bit como um evento de Bernoulli com probabilidade de perda p_e , pode-se considerar um pacote de dados como uma seqüência de eventos independentes de Bernoulli. Desta forma, a probabilidade de um quadro de tamanho L ser transmitido com sucesso é expressa por:

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

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 dada por p_e como:

$$\eta = (1 - p_e)^L \cdot \left(\frac{\frac{L}{G + \frac{L}{t_d}}}{t_d} \right) \quad (3)$$

Na figura 4, 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 como:

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

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

E 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

$$L^2 + L \cdot t_d \cdot G - t_d \cdot G / p_e = 0 \quad (4)$$

3.2. Algoritmos propostos

O primeiro mecanismo proposto é baseado na função eficiência obtida na subseção anterior. Assim, a medida que transmite os quadros, o transmissor calcula o tamanho ótimo

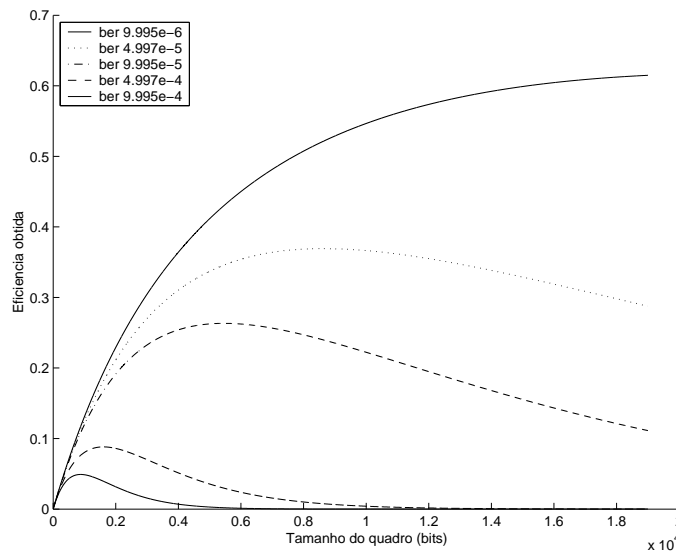


Figura 4: Função eficiência, para várias taxas de erro.

```

Inicio {
  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 5: Pseudo código do primeiro mecanismo.

do fragmento baseado na taxa de erro obtida do estimador utilizado. O cálculo do fragmento é feito periodicamente, e é controlado por um temporizador que regula o período. O pseudo-código abaixo ilustra o mecanismo proposto:

Onde `adapTimeout` é um parâmetro do mecanismo, com o qual programa-se o temporizador.

O segundo mecanismo é baseado no esquema *additive increase-multiplicative decrease*, utilizado para controle da janela deslizante do TCP. Nesse mecanismo, três variáveis são utilizadas: *alfa*, *beta* e *gama*. A variável *alfa* é responsável pelo decréscimo multiplicativo do fragmento. Dessa forma, ao perder um quadro, o transmissor 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 abaixo ilustra o comportamento do mecanismo:

4. Modelo de erros

Nesta seção apresentaremos brevemente o modelo de erros utilizado para avaliação de desempenho das propostas apresentadas na seção anterior.

As simulações do canal de acesso foram realizadas utilizando o modelo do protocolo 802.11 disponível no simulador *NS* e o modelo de erros de transmissão foi implementado neste simulador de acordo com o modelo Gilbert-Elliot [Ebert e Willig, 1999], utilizando as otimizações propostas por *Ebert e Willig* [Ebert e Willig, 1999].

```

Ao perder um quadro {
    fragthreshold = fragthreshold / alfa ;
}

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

```

Figura 6: Pseudo código do segundo mecanismo.

A escolha deste modelo está fundamentada em trabalhos da literatura, como o apresentado em [Haowei Bai e Atiquzzaman, 2003], que mostra 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 (*GOOD*) representa o canal em boas condições de transmissão e o outro estado (*BAD*) representa um período de condições desfavoráveis. Tipicamente, os períodos de permanência no estado (*BAD*) são bem mais curtos do que os do outro estado.

4.1. Modelo de Gilber-Elliot

A Figura 7 ilustra a representação discreta do modelo de *Gilbert-Elliot*. Neste exemplo, os parâmetros P_{gb} e P_{bg} representam as propabilidades de passagem do estado *GOOD* para o estado *BAD* e do estado *BAD* para o estado *GOOD*, a cada transmissão de um *bit*. Além disso, dois parâmetros P_{bad} e P_{good} representam as probabilidades de perda de um bit no estado *BAD* e no estado *GOOD* ($P_{bad} > P_{good}$), respectivamente.

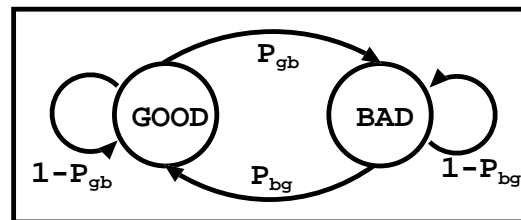


Figura 7: Modelo de Erros de Gilbert-Elliot

A parametrização deste modelo é uma questão amplamente estudada. Os parâmetros utilizados neste trabalho são baseados nos encontrados em [Pravin Bhagwat e Tripathi, 1997] e [Wang e Moayeri., 1995], e foram configurados como definido a seguir:

- Período médio de permanência em *BAD*: $50ms \leq E[Perm_{BAD}] \leq 500ms$
- Período médio de permanência em *GOOD*: $1s \leq E[Perm_{GOOD}] \leq 10s$
- Probabilidade de erro em um bit no estado *BAD*: $P_{BAD} = 10^{-2}$
- Probabilidade de erro em um bit no estado *GOOD*: $P_{GOOD} = 10^{-5}$

Em nossa implementação consideramos a aproximação de que qualquer pacote com pelo menos um *bit* errôneo é detectado como pacote inválido.

Para justificar esta suposição, é importante destacar que o 802.11 utiliza um CRC de 32 *bits* para detecção de erros de transmissão, o que resulta em probabilidade 1 de detecção de qualquer erro de comprimento de *bits* menor ou igual a 33 *bits* e de qualquer erro de seqüência de *bits* de comprimento ímpar. Os demais erros são detectados com probabilidade superior a 0,99999.

O modelo de Gilbert-Elliot é normalmente utilizado para modelar ambientes *in-door* e constitui um caso geral do modelo de Markov de n estados [Wang e Moayeri., 1995]

que é utilizado, também, para ambientes *outdoor*. Logo os testes realizados neste trabalho representam apenas ambientes *indoor*.

4.2. Modelo Otimizado

O grande problema do modelo apresentado na seção anterior é que sua simulação exige que um evento seja gerado para cada *bit* transmitido e dois experimentos de Bernoulli sejam feitos: um para testar se o *bit* é errôneo ou não de acordo com a probabilidade de erro do estado em que o sistema se encontra, e outro para verificar se haverá transição entre os estados do canal. Esta abordagem torna a simulação em nível de pacotes demasiadamente dispendiosa, dado que cada pacote contém, potencialmente, milhares de bits.

O modelo otimizado [Ebert e Willig, 1999] propõe a diminuição desta complexidade através do uso de análise estatística. Duas modificações são inseridas no modelo de Gilbert-Elliot:

- Ao invés de testar se há transições entre estados a cada *bit*, pode-se definir qual é quantidade de *bits* transmitidos até que haja uma transição, como explicado a seguir.

Seja X a variável aleatória que representa o número de *bits* transmitidos até que haja transição entre estados e seja $P[X = i] = (1 - P)^i$ a probabilidade de i *bits* serem transmitidos sem que haja transição entre estados, onde P é a probabilidade de haver mudança de estado a cada *bit* transmitido, sabe-se então, que a variável aleatória X segue a distribuição geométrica. Pode-se facilmente, então, calcular a média de *bits* transmitidos até que ocorra mudança de estado. Por exemplo, em uma implementação contínua, dada a média $E[Perm]$ dos períodos de permanência em cada um dos estados, pode-se facilmente obter a taxa de transição entre eventos através de uma distribuição exponencial de taxa

$$\lambda = \frac{1}{E[Perm]} \quad (5)$$

- Ao invés de testar cada *bit*, pode-se calcular a probabilidade de todo um pacote conter *bits* errôneos: sabendo que a avaliação da probabilidade de erro de cada *bit* também é feita através de experimentos de Bernoulli, pode-se facilmente calcular a probabilidade P_{ok} de não haver nenhum *bit* com erro em um pacote de n *bits* e a probabilidade P_{err} de haver pelo menos 1 *bit* com erro em todo o pacote:

$$P_{ok} = (1 - P)^n, \quad (6)$$

$$P_{err} = 1 - (1 - P)^n \quad (7)$$

onde P é a probabilidade de erro de um *bit* no estado atual.

Através da utilização da equação 7 pode-se realizar apenas um experimento de Bernoulli por pacote para checar se este contém ou não erros. Este teste pode ser feito, em um modelo da camada de enlace do 802.11 no momento em que é disparado o evento que representa o recebimento do último bit de um pacote. Neste caso não são adicionados eventos ao modelo da camada de enlace, minimizando o aumento do tempo de simulação do canal com o modelo de erro implementado.

Evidentemente, pode haver transições entre os estados *GOOD* e *BAD* durante o recebimento de um pacote. Este fenômeno pode ser tratado de duas maneira diferentes: (1) testando a probabilidade de haver erro no pacote a cada transição entre os estados do canal utilizando a equação 7 para os *bits* recebidos durante o tempo de permanência no estado atual, o que adicionaria um evento ao modelo de simulação para cada transição entre

Gilbert-Elliot:

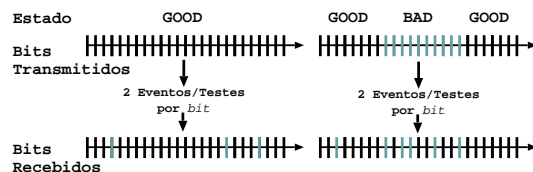


Figura 8: Eventos necessários - Gilbert-Elliot

os estados *GOOD* e *BAD*, ou (2) ao disparo do evento relativo ao recebimento do último *bit* do pacote, através da média $\frac{(1-P)}{1-(1-P)}$ da distribuição geométrica, onde P representa a probabilidade de transição entre os estados.

A Figura 8, ilustra o funcionamento do modelo de Gilbert-Elliot, onde para cada *bit* transmitido, um evento e dois experimentos de Bernoulli são gerados. Já no modelo otimizado, o número de testes por pacote, ilustrado na Figura 9, pode chegar a apenas 1, caso não haja transição entre estados, e permanece sensivelmente menor quando há transições entre estados.

Ebert e Willig:

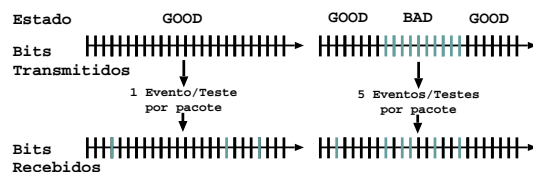


Figura 9: Eventos necessários - Ebert e Willig

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, 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 99%.

Na Figura 10 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. Verificase, 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.

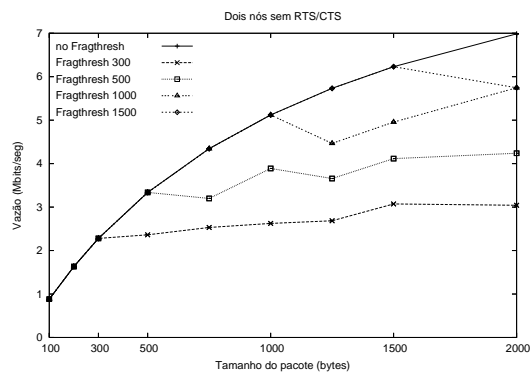


Figura 10: Vazão em função do tamanho do pacote, sem modelo de erro.

Na Figura 11, foi avaliada a vazão da rede em função da taxa de erro do meio. Verifica-se, como esperado, que existe um tamanho de fragmento ótimo que maximiza a vazão da rede para cada intervalo de erro. Assim, quanto pior a qualidade do canal, menor o fragmento cuja a probabilidade de sucesso é ótima.

Figura 11: Vazão em função da taxa de erro, com pacotes de 2000 bytes.

Na Figura ??,

6. Conclusões

O objetivo deste trabalho foi avaliar o mecanismo de fragmentação e sua influência no desempenho das redes *ad-hoc* 802.11. Foram analisados diversos cenários, através de simulações, com experimentos que visavam correlacionar características do 802.11 tais como, colisões, uso do mecanismo RTS/CTS e variação do limiar de fragmentação, utilizado para dividir os quadros em fragmentos. As métricas usadas para avaliar o desempenho foram vazão agregada e número de colisões da rede. Vale ressaltar que os resultados foram obtidos com uma taxa de transmissão de 11 Mbps e uma taxa básica de 2 Mbps, para transmissão de quadros de controle e cabeçalho dos quadros.

Os resultados mostram que a eficiência do protocolo varia sensivelmente com o tamanho do fragmento usado. Assim, em situações aonde a vazão deve ser priorizada,

Figura 12: Comparação dos mecanismos em função da taxa de erro, com pacotes de 2000 bytes.

deve-se utilizar tamanhos grandes de fragmento, enquanto que fragmentos pequenos devem ser utilizados em situações aonde as colisões devem ser evitadas. No último cenário, verificou-se que o mecanismo RTS/CTS diminui a vazão total da rede, embora torne o acesso ao meio mais uniforme e diminua o número de colisões no meio.

6.1. Trabalhos futuros

Referências

- Ebert, J. e Willig, A. (1999). A gilbert-elliot bit error model and the efficient use in packet level simulation.
- 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 Atiquzaman, M. (2003). Error modeling schemes for fading channels in wireless communications: A survey. Em *IEEE-COMSOC*. IEEE.
- 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*.