

Desenvolvimento de aplicações adaptativas para redes IEEE 802.11

Arlindo Flávio da Conceição¹, Fabio Kon¹

¹Grupo de Sistemas Distribuídos – Departamento de Ciência da Computação
Instituto de Matemática e Estatística – Universidade de São Paulo (DCC - IME/USP)
<http://gsd.ime.usp.br>

{arlindo,kon}@ime.usp.br

Abstract. *This work explores the development of adaptive applications for widely used IEEE 802.11 wireless networks. We describe the most challenging problems of programming for these networks and present useful techniques to solve them. We propose a novel adaptive algorithm that improves the efficiency of datagram streaming over IEEE 802.11 networks. It uses the signal quality information to adapt the transmission and, by doing this, improves the network utilization. In addition, this work presents the methods to interact, at the application layer, with IEEE 802.11 interfaces in the Linux and Windows operational systems.*

Resumo. *Devido ao crescente uso do protocolo IEEE 802.11 nos dias atuais, este trabalho explora o desenvolvimento de aplicações adaptativas para redes IEEE 802.11. Com essa finalidade, apontamos os principais novos desafios e problemas relacionados ao desenvolvimento de aplicações para essas redes, assim como técnicas úteis para resolvê-los. Apresentamos um novo algoritmo adaptativo baseado na qualidade do sinal para adaptar a transmissão de fluxos contínuos às condições da conexão, aumentando assim a eficiência de utilização da rede. Além disso, descrevemos os métodos para, na camada de aplicação, interagir com as interfaces IEEE 802.11 em sistemas Linux e Windows.*

1. Introdução

As redes sem fio IEEE 802.11 (Wi-Fi) firmaram-se como o padrão *de facto* para redes locais sem fio por oferecerem mobilidade e alta capacidade de vazão a custos relativamente baixos. Trazem, porém, novos desafios ao desenvolvimento de aplicações, como a ocorrência de migrações (*handoffs*) e a alta variabilidade da qualidade da conexão.

Apesar das diferenças entre as redes sem fio e as redes cabeadas tradicionais, do ponto de vista do desenvolvimento de aplicações, as redes sem fio IEEE 802.11 têm sido muitas vezes tratadas como redes Ethernet. Além disso, os protocolos UDP e TCP, que geralmente são utilizados na camada de transporte, não foram projetados levando em consideração as características das redes sem fio, mas sim das redes Ethernet cabeadas. A fim de adequar esses protocolos às redes sem fio, foram feitas propostas para a otimização de TCP [Balakrishnan et al. 1997, Pilosof et al. 2003] e de UDP [Larzon et al. 2004], mas essas otimizações, em geral, não foram amplamente adotadas.

Dado que as abstrações providas por TCP e UDP não são plenamente capazes de lidar com as peculiaridades das redes IEEE 802.11, uma das alternativas é a implementação de técnicas adaptativas na camada de aplicação. Deste modo, este artigo tem dois

objetivos: prover uma referência aos desenvolvedores interessados em desenvolver aplicações adaptativas para redes IEEE 802.11 e, principalmente, apresentar uma estratégia adaptativa que, implementada na camada de aplicação, seja capaz de aumentar a eficiência de transmissão em redes IEEE 802.11. Para cumprir esses objetivos, o conteúdo aqui exposto está organizado da seguinte maneira: a Seção 2 discute os trabalhos relacionados e contextualiza este trabalho em relação a essas referências. A Seção 3 apresenta os novos desafios e problemas referentes ao desenvolvimento para redes IEEE 802.11. A Seção 4 descreve algumas práticas e técnicas úteis para a solução desses problemas. A Seção 5 introduz os métodos que podem ser utilizados para interagir com as interfaces de rede sem fio a partir da camada de aplicação e, sobretudo, mostra como podem ser obtidas e interpretadas as informações sobre a qualidade de sinal e de conexão em redes IEEE 802.11. A Seção 6 descreve os princípios e a implementação de uma estratégia adaptativa que, baseada na qualidade do sinal, aumenta significativamente a eficiência das transmissões sobre redes IEEE 802.11. Por fim, a Seção 7 traz as considerações finais deste trabalho.

2. Trabalhos relacionados

O número de trabalhos que abordaram IEEE 802.11 a partir da camada de aplicação é relativamente pequeno, pois a maioria concentra-se em camadas inferiores; por exemplo, na otimização da camada MAC [Tan and Guttag 2005], dos protocolos de transporte [Pilosof et al. 2003], ou na caracterização do tráfego [Muñoz et al. 2001].

Ao observar as pesquisas realizadas pode-se notar uma lacuna na qual se esquece que é na camada de aplicação que residem as maiores possibilidades para adaptação, pois é nela que se podem aplicar técnicas adaptativas baseadas nos requisitos das aplicações, o que não poderia ser feito nos elementos de rede e protocolos subjacentes, sem prejuízo à generalidade dos mesmos. Contudo, olhando para o passado recente, é possível ver algumas razões para a escassez de pesquisa nessa área. A principal foi, sem dúvida, a limitada capacidade de interação entre a aplicação e as interfaces de rede. Mas essa limitação foi muito reduzida com as novas versões de NDIS (*Network Driver Interface Specification*) e com a incorporação ao núcleo do Linux de ferramentas para redes sem fio (*Wireless Tools*). Outra razão para a lacuna, mas que ainda persiste, é a dificuldade de condução de experimentos práticos sobre redes IEEE 802.11, especialmente daqueles relacionados à mobilidade, pois a alta variabilidade intrínseca às redes sem fio dificulta a execução de experimentos representativos e reprodutíveis.

A fim de tentar sanar essa lacuna, este trabalho concentra-se na camada de aplicação, apontando não apenas os principais problemas vinculados à programação para redes IEEE 802.11, mas também algumas técnicas que podem ser úteis à sua solução. Veremos que, apesar de ainda ser difícil conduzir pesquisas empíricas em redes sem fio, é possível implementar estratégias adaptativas capazes de otimizar o desempenho das aplicações executadas sobre redes IEEE 802.11.

3. Novos desafios impostos pelas redes IEEE 802.11

Os novos desafios para o desenvolvimento de aplicações adaptativas trazidos pelas redes IEEE 802.11 podem ser classificados em três grupos:

3.1. Problemas intrínsecos às redes sem fio

As redes IEEE 802.11 estão sujeitas aos problemas intrínsecos das redes sem fio, dentre os quais estão a maior susceptibilidade a interferências externas, o desvanecimento do sinal devido à mobilidade e à distância ao ponto de acesso, a propagação do sinal por dois ou mais caminhos (efeito *multipath fading*) e a existência de nós invisíveis (*hidden nodes*) [Schiller 2000].

Uma das razões para a maior susceptibilidade a interferências externas é o fato de outros dispositivos eletrônicos, que vão desde fornos microondas até interfaces Bluetooth, usarem a mesma frequência de rádio que IEEE 802.11. Além dos dispositivos eletrônicos, existem também outros fatores capazes de interferir na qualidade de uma conexão sem fio, sendo alguns deles obviamente não gerenciáveis pela camada de aplicação, como a orientação e alturas das antenas, o enfraquecimento do sinal devido a sobreposição de objetos, a umidade do ar etc. Desse modo, a qualidade do sinal nessas redes — devido ao grande número de fontes de interferência — pode variar substancialmente e em curtos intervalos de tempo, o que dificulta o projeto de algoritmos adaptativos.

Além disso, em redes sem fio — devido a existência de *hidden nodes* — é inviável aplicar detecção de colisão (CSMA/CD), por isso IEEE 802.11 usa CSMA/CA, que tem implicações na latência e taxa de perdas. Outro problema que pode ser considerado intrínseco às redes sem fio são os *handoffs*. Presentes apenas nas redes sem fio, os *handoffs* podem acarretar períodos de desconexão da ordem de alguns segundos [Ramani and Savage 2005].

3.2. Problemas intrínsecos às redes IEEE 802.11

Os mecanismos IEEE 802.11¹ também dificultam o projeto de aplicações adaptativas. Por exemplo, o mecanismo de *Automatic Repeat reQuest* (ARQ) reenvia um quadro de transmissão caso a sua entrega não tenha sido confirmada [Gast 2002]. Em suma, devido à atuação do ARQ, um quadro é reenviado pelo menos quatro vezes antes que a sua perda seja notificada ao protocolo de nível superior. Desse modo, ao usar retransmissões para melhorar a confiabilidade da transmissão, IEEE 802.11 faz com que as taxas de perda de pacotes não reflitam fielmente a real condição do canal. Na prática, observamos que, em redes IEEE 802.11, a qualidade da conexão já se encontra bastante deteriorada quando as perdas de pacotes começam a ser percebidas.

Assim como o mecanismo de ARQ, outros mecanismos internos ao protocolo IEEE 802.11 afetam fortemente as métricas de adaptação. Dentre eles podemos citar: *Automatic Rate Control* (ARC), *Power Saving Mode* (PSM) e *Request to Send / Clear to Send* (RTS/CTS). Por exemplo, o mecanismo de RTS/CTS, ao tentar eliminar o problema de *hidden-nodes*, agrega alta variação à latência. PSM, por sua vez, pode gerar períodos de inatividade que podem ser interpretados como congestionamentos pelo protocolo TCP. Por fim, o mecanismo de ARC é o pivô de dois problemas: a má distribuição da capacidade de transmissão em redes IEEE 802.11 [Tan and Gutttag 2005] e a redução da

¹Cabe dizer que não contestamos a importância dos mecanismos adaptativos de IEEE 802.11, pois eles foram e são fundamentais, cada um a seu modo, para a ampla utilização dessas redes. Mas é importante chamar a atenção para as conseqüências negativas que esses mecanismos trazem para o projeto de aplicações adaptativas para redes IEEE 802.11.

capacidade de vazão total do sistema em função da degradação da qualidade de sinal de apenas uma unidade móvel [Heusse et al. 2003].

3.3. Problemas relacionados à diversidade de implementação

Outro desafio atualmente imposto pelas redes IEEE 802.11 é a diversidade de comportamento entre dispositivos Wi-Fi, o que advém do uso, tanto de dispositivos IEEE 802.11 de diferentes fabricantes, quanto de diferentes configurações. Por questões de simplicidade, mas também para não invadir o espaço de diferenciação competitiva entre os fabricantes de dispositivos de rede, o padrão IEEE 802.11 deixa em aberto alguns pontos da especificação; uma parte destes, porém, possui forte influência no comportamento das transmissões sobre essas redes. Por exemplo, o padrão IEEE 802.11 não especifica como o mecanismo de ARC deve escolher a modulação (taxa de transmissão) a ser usada em caso de degradação do sinal. Ao não definir quais, nem como essas métricas de adaptação devem ser usadas pelo mecanismo de ARC, o padrão IEEE 802.11 delega aos fabricantes essa definição; sendo assim, cada fabricante define os algoritmos e limiares de adaptação segundo as peculiaridades de seus dispositivos; ou seja, mediante a degradação do sinal, mecanismos ARC de dispositivos de fabricantes diferentes atuam de maneira diversa e, em função disso, experimentam diferentes taxas de perda de pacotes, latência, *jitter* etc.

O padrão IEEE 802.11 também deixa em aberto como as interfaces de rede (*Network Interface Cards*, ou NICs) reportam a qualidade do sinal. NICs de diferentes fabricantes, quando submetidos a uma mesma condição de conexão, podem reportar diferentes níveis de qualidade de sinal. Por exemplo, os NICs IEEE 802.11 dos fabricantes Atheros, Symbol e Cisco usam escalas diferentes para reportar a qualidade do sinal: de 0 a 60, de 0 a 31 e de 0 a 100, respectivamente. Além disso, o valor médio da qualidade do sinal nesses NICs (30, 16 e 50, respectivamente) corresponde a diferentes valores de força do sinal (-65 dBm, -70 dBm e -58 dBm, respectivamente). Outro fator que também contribui para a diversidade de comportamentos em IEEE 802.11 é a existência de variantes interoperantes do padrão; por exemplo, NICs IEEE 802.11g podem acessar redes 802.11b e g, mas submetidos a redes com capacidades de vazão diferentes.

Desse modo, o uso de diferentes NICs e configurações implica a utilização de conexões com diferentes taxas de perda de pacotes, latência, *jitter*, qualidade do sinal etc. Essa diversidade de comportamentos, por sua vez, dificulta o desenvolvimento de aplicações adaptativas, pois a interpretação das métricas de adaptação pode variar em função dos ambientes utilizados. Resta observar que a diversidade de implementação é um problema necessariamente exclusivo das redes IEEE 802.11: afinal, também existem diversos fabricantes para interfaces *Ethernet*. Mas ela é mais perceptível nas redes IEEE 802.11, pois a mobilidade leva a uma maior interação entre interfaces de diferentes fabricantes, com possibilidade de interação entre interfaces diferentes durante uma única conexão.

Portanto, no desenvolvimento para IEEE 802.11 existem novos problemas que — por não serem devidamente tratados pelas abstrações de transporte — são percebidos na camada de aplicação através de:

- maiores taxas de perda de pacotes;
- maior incidência de perda de pacotes em rajadas;
- maior variação na latência;

- maior incidência de entrega de pacotes em rajada;
- variações abruptas na qualidade da conexão;
- períodos de desconexão;
- menor previsibilidade da conexão.

4. Desenvolvimento para redes IEEE 802.11: práticas recomendadas e técnicas úteis

A seguir, apontamos algumas recomendações e técnicas que devem ser consideradas no desenvolvimento de aplicações para redes IEEE 802.11.

4.1. Tamanhos dos pacotes

A primeira recomendação para o desenvolvimento de aplicações para redes IEEE 802.11 é relativa ao tamanho dos pacotes. De maneira geral, a utilização de pacotes pequenos implica menores taxas de erros, mas, por outro lado, reduz a eficiência de utilização da rede devido à maior inserção de sobrecargas. Estas advêm, por exemplo, da transmissão de um maior número de cabeçalhos e do maior número de troca de mensagens de confirmação de entrega de quadros (mensagens ACK). Além disso, é importante que os pacotes não sejam maiores do que os limites de fragmentação da camada *Ethernet*, que é de 1.500 octetos. Ou seja, os pacotes UDP e TCP não devem ser maiores do que 1.472 e 1.460 octetos, respectivamente. Ultrapassar esses limites de tamanho implica um número maior de fragmentações e desfragmentações, o que deve ser evitado. Além disso, escolha do tamanho ideal dos pacotes também depende fortemente da natureza da aplicação [Ikkurthy and Labrador 2002].

4.2. Novas métricas de adaptação

Em redes IEEE 802.11, além das métricas de adaptação tradicionais, tais como a taxa de perda de pacotes e a latência, existem novos fatores a serem monitorados e que também podem ser utilizados como métricas de adaptação. Dentre essas novas métricas destacam-se: a qualidade do sinal e o número de retransmissões executadas pela camada MAC. A qualidade do sinal, como veremos adiante, oferece bons indicadores sobre a necessidade de adaptação e pode também indicar a necessidade de execução de *handoffs*. Por sua vez, o número de retransmissões executadas pela camada MAC complementa com vantagens a informação de perda de pacotes, pois esta última, como foi mencionado anteriormente, é mascarada pelas retransmissões automáticas.

Outro fator importante a ser monitorado é a taxa de conexão. Dispositivos IEEE 802.11 — diferentemente das interfaces Ethernet — podem conectar-se a diferentes taxas de transmissão. A escolha da taxa mais apropriada é feita pelo mecanismo de ARC, que pode reduzir significativamente a capacidade de transmissão sem qualquer notificação à aplicação. Em redes IEEE 802.11b, por exemplo, a taxa de conexão pode ser reduzida de 11 Mbps para 5,5 Mbps, o que diminui a vazão efetiva de 5,6 Mbps para 1,6 Mbps, aproximadamente. Portanto, as aplicações devem estar preparadas para reduções abruptas na capacidade de vazão. Mas atualmente, apesar do impacto da atuação do mecanismo de ARC nas aplicações, não há métodos simples para a monitoração da taxa de transmissão. Em outro trabalho propomos um mecanismo para auxiliar a disponibilização das escolhas feitas pelo mecanismo ARC à camada de aplicação [da Conceição and Kon 2006].

A importância de monitorar as taxas de transmissão reside também na interpretação confiável das métricas de adaptação, pois as mesmas taxas de erro não podem ser interpretadas da mesma forma, quando obtidas usando-se diferentes taxas de transmissão; essa mesma lógica aplica-se também a outras métricas de adaptação, tais como a latência e a vazão.

Também é importante observar que diferentes ambientes de execução (computadores, NICs, pontos de acesso e as respectivas configurações) podem implicar diferentes comportamentos de perdas de pacotes e de latência. A latência, por exemplo, pode variar significativamente após a execução de um *handoff*. Sendo assim, algoritmos adaptativos robustos não devem usar limiares de adaptação fixos (*hard coded*), mas devem valer-se de mecanismos de calibração capazes de determinar dinamicamente os limiares de adaptação.

A carga total na rede seria uma métrica importante para o controle de admissão em redes IEEE 802.11, porém esta métrica não está disponível para as aplicações. Nas redes IEEE 802.11, a capacidade dos pontos de acesso é compartilhada pelas unidades móveis, de tal sorte que uma unidade móvel pode exceder a capacidade do sistema e, dessa forma, interferir negativamente no tráfego de outras unidades [da Conceição and Kon 2003]. Na ausência de métodos eficientes para controle de admissão de fluxos, é preciso que as aplicações façam uso moderado dos recursos de rede; por exemplo, adotando estratégias incrementais de utilização dos recursos. Isso se aplica especialmente aos sistemas baseados em UDP que, por não possuírem mecanismos integrados para controle de congestionamentos, podem facilmente exceder a capacidade de transmissão da rede.

4.3. Recuperação de erros

Devido às razões expostas anteriormente, as redes IEEE 802.11 estão sujeitas a taxas maiores de perdas de pacotes do que as redes cabeadas tradicionais, mas o impacto dessas perdas pode ser minimizado pela utilização, na camada de aplicação, de técnicas para recuperação de erros, tais como: *Automatic Repeat reQuest* (ARQ) e *Forward Error Correction* (FEC). Os mecanismos de ARQ [Sinha and Papadopoulos 2004] baseiam-se no reenvio de pacotes² e os de FEC, por sua vez, transmitem informação redundante nos pacotes, de tal forma que, perdido um deles, este pode ser reconstruído [Muñoz et al. 2001].

Em termos de uso de banda, FEC costuma ser menos eficiente do que ARQ, mas em geral é capaz de recuperar as perdas mais rapidamente do que ARQ. Nas redes IEEE 802.11, entretanto, é comum que a perda de pacotes ocorra em rajada (em seqüência), especialmente quando a conexão já se encontra bastante deteriorada; nessas situações, ARQ é mais adequado do que FEC [Vandalore et al. 2001], pois, ao se perderem pacotes em seqüência, normalmente perdem-se também as informações usadas para reconstruí-los. As técnicas para minimização das perdas de pacotes podem ser aplicadas a todos os pacotes da transmissão, ou podem ser aplicadas apenas para evitar a perda dos pacotes considerados mais importantes. Por exemplo, na transmissão de fluxos de vídeo H.264, normalmente feita sobre UDP, existem pacotes do tipo *parameter set* que são essenciais para a decodificação do vídeo. Recomenda-se que a transmissão desses pacotes seja protegida [Etoh and Yoshimura 2005], quer pela aplicação de FEC, de ARQ, pela simples duplicação de envio desses pacotes, quer pelo uso de canais confiáveis.

²A própria camada MAC de IEEE 802.11, ao fazer retransmissões, implementa um mecanismo de ARQ.

4.4. Períodos de desconexão

Uma das principais diferenças de IEEE 802.11 em relação às outras redes é a maior incidência de períodos de desconexão, que podem ocorrer por várias razões, tais como *handoffs*, interferências, sombreamento etc. E eles normalmente ocorrem de maneira abrupta, sem que haja tempo hábil para o acionamento de mecanismos capazes de notificá-las ou de amenizar seus efeitos. Tomemos como exemplo uma aplicação de vídeo sob demanda em que uma unidade móvel requisita um vídeo a um servidor na Internet. Nessas aplicações, os fluxos de vídeo são geralmente implementados usando-se UDP e, em geral, os servidores não possuem mecanismos adaptativos para interromper a entrega dos quadros de vídeo quando a unidade móvel estiver experimentando períodos curtos de desconexão. Sem tais mecanismos, os quadros de vídeo enviados pelo servidor continuarão chegando ao ponto de acesso que, por sua vez, tentará transmiti-los para a unidade móvel — mas obviamente sem sucesso. Isso acarreta, antes de tudo, o desperdício da capacidade do ponto de acesso, o que pode afetar negativamente a transmissão para outras unidades móveis por ele servidas. Em suma, as aplicações para redes IEEE 802.11 precisam implementar mecanismos capazes de lidar com períodos de desconexão. Isso pode ser feito através de mecanismos para operação desconectada, ou mesmo através da implementação de mecanismos mais simples — mas muito úteis — como a troca de mensagens *KeepAlive*, que permitem verificar constantemente a conectividade de uma unidade móvel. Uma vantagem adicional de implementar as mensagens *KeepAlive* é que elas podem também carregar informações úteis à adaptação, como por exemplo a qualidade da conexão, servindo desse modo como uma forma de sinalização.

Na Seção 6, apresentaremos a implementação de um novo algoritmo que ilustra o uso das mensagens *KeepAlive*, mas antes, vejamos rapidamente como pode ser obtida, entre outras informações, a qualidade do sinal em redes IEEE 802.11.

5. Interação entre aplicações e interfaces IEEE 802.11

É importante lembrar que as conexões IEEE 802.11 não precisam ser necessariamente tratadas como caixas-pretas: diversos NICs disponíveis no mercado possibilitam a alteração de suas configurações a partir da camada de aplicação, permitindo, entre outras coisas, a alteração de parâmetros como o limiar de fragmentação e o uso do mecanismo RTS/CTS.

5.1. Em sistemas Windows: *Network Driver Interface Specification (NDIS)*

Nos sistemas operacionais da família Windows, a forma mais simples para interagir com os NICs a partir da camada de aplicação³ é através de NDIS. Para cada tecnologia de rede, tais como Ethernet ou IEEE 802.11, NDIS especifica um conjunto de *Object Identifiers* (OIDs), cujo formato é *OID_XXX_YYY*, onde *XXX* é a tecnologia de rede e *YYY* é o objeto desejado. Por exemplo, *OID_802_3_YYY* representa o objeto *YYY* das redes Ethernet e *OID_802_11_RSSI* representa a qualidade do sinal (*Received Signal Strength Indication* ou *RSSI*) de um NIC para redes IEEE 802.11. A lista completa de OIDs pode ser encontrada em <http://msdn.microsoft.com>. Contudo,

³Apesar de ser executado no espaço do núcleo do sistema operacional, NDIS pode ser acessado pela camada de aplicação através de chamadas *IOCTL* ao *driver* *NDISPROT*, que se encontra disponível na Web.

NDIS define três classes de OIDs: obrigatórios, recomendados e opcionais. Os OIDs obrigatórios são aqueles considerados essenciais à interação com o NIC, como por exemplo `OID_802_11_RSSI`. Os OIDs recomendados são aqueles considerados importantes, mas não essenciais, como por exemplo `OID_802_11_STATISTICS`, que fornece estatísticas sobre a conexão. Por fim, existem também os OIDs opcionais, como a requisição `OID_802_11_RTS_THRESHOLD`, que permite alterar o valor do *RTS Threshold*. A principal implicação da classificação dos OIDs nesses três grupos é que nem todo OID funciona para todo NIC. Por exemplo, em nossos experimentos, dentre quatro NICs testados, a requisição de `OID_802_11_STATISTICS`, cuja implementação não é obrigatória, funcionou em apenas um deles.

A melhor fonte de informação sobre como usar o NDIS são os códigos-fonte da *Wireless Research API* (WRAPI), que estão disponíveis em <http://ramp.ucsd.edu/pawn/wrapi>; porém essa biblioteca pode não funcionar corretamente nas versões mais novas do sistema operacional Windows. Os passos necessários para o uso de NDIS e um programa-exemplo, escrito em linguagem C, que chama o objeto `OID_802_11_RSSI`, podem ser encontrados em <http://gsd.ime.usp.br/software/WirelessMultimedia/ndis>. A partir desse programa exemplo, do código fonte de WRAPI e da documentação de NDIS contendo a funcionalidade de cada OID, é relativamente simples criar suas próprias DLLs para interação com as interfaces de rede. Para as futuras versões dos sistemas operacionais da Microsoft, por exemplo o Windows Vista, algumas das funções de NDIS serão preferencialmente realizadas usando-se *Native Wi-Fi*.

5.2. Em sistemas Linux: /proc/net/wireless

Nos sistemas baseados em Linux deve-se usar as ferramentas para redes sem fio (*Wireless Tools*⁴). Em particular as aplicações `iwconfig` e `iwspy` e a biblioteca `iwlib.o`, que pode ser usada de dentro das aplicações. As principais informações sobre a qualidade da conexão sem fio, contudo, estão resumidas no pseudo-arquivo `/proc/net/wireless`, cujo formato é apresentado na Figura 1.

Inter- face	sta- tus	Quality			Discarded packets					Missed beacon	WE 16
		link	level	noise	nwid	crypt	frag	retry	misc		
wlan0:	0000	100	224	0	0	0	0	6	122829	0	
wlan1:	0000	55.	218.	163.	0	0	0	1	0	0	

Figura 1. O pseudo-arquivo `/proc/net/wireless`.

A Figura 1 mostra as informações referentes a duas interfaces IEEE 802.11, `wlan0` e `wlan1`, e as respectivas estatísticas sobre a qualidade da conexão. Para detalhes sobre o significado de cada informação fornecida no pseudo-arquivo `/proc/net/wireless`, consulte o manual da aplicação `iwconfig`. No contexto deste trabalho, os parâmetros mais relevantes são:

- **link:** qualidade do sinal;
- **level:** força do sinal (*signal strength*);
- **noise:** força do ruído;
- **retry:** número de retransmissões.

⁴Vide http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html.

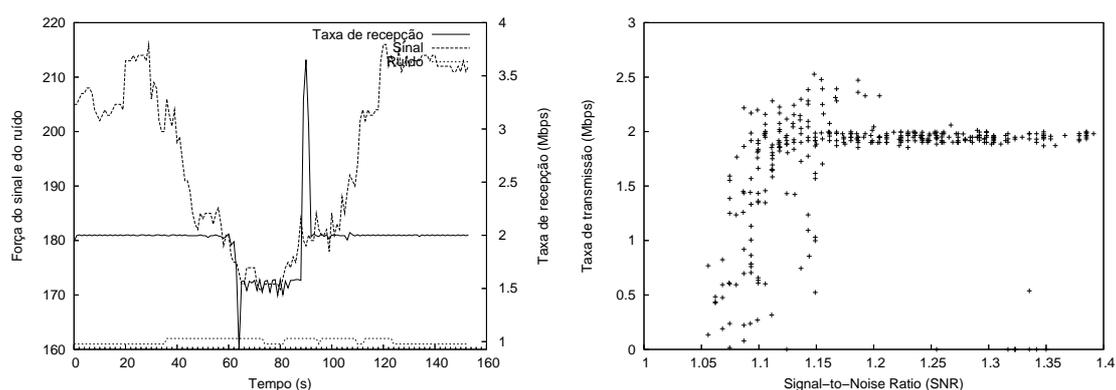
Na Figura 1, observe que os NICs reportam a qualidade do sinal de maneiras distintas; wlan0 reporta a qualidade como um indicador relativo (100%) e wlan1 a relação entre a força do sinal e do ruído (ou seja, $\text{link}=\text{level}-\text{noise}$). Em suma, os valores de link, level e noise obedecem a uma aritmética de oito bits, podendo assumir valores entre 0 e 255; porém, cada fabricante estabelece como os valores de qualidade de sinal são informados. Na próxima seção, usamos essas informações para implementar um novo algoritmo adaptativo para otimização de transmissões sobre redes IEEE 802.11.

6. Adaptação de fluxos contínuos baseada na qualidade do sinal

Vimos nas seções anteriores os principais desafios para o desenvolvimento de aplicações adaptativas para redes IEEE 802.11, as principais técnicas para ajudar nesse desenvolvimento e as formas para obtenção de informações sobre a qualidade das conexões sem fio. Esta seção, por sua vez, apresenta uma nova estratégia adaptativa que, baseada na qualidade do sinal das redes IEEE 802.11, adequou os fluxos de dados à capacidade da rede, aumentando assim a eficiência das transmissões. Mas, antes de apresentar os algoritmos, é necessário entender o comportamento das transmissões sobre redes IEEE 802.11 em condições de mobilidade.

A Figura 2 apresenta o comportamento de um fluxo contínuo enviado de uma máquina servidora para a unidade móvel através de um ponto de acesso, na presença de mobilidade. Ou seja, durante o experimento, a unidade móvel é levada de um ponto próximo do ponto de acesso para um ponto mais afastado e trazida de volta. Da Figura 2(a) podem-se extrair algumas observações iniciais sobre o comportamento de um fluxo e de seus indicadores em função da mobilidade:

- a qualidade do sinal varia significativamente;
- o ruído apresenta pequenas variações, e;
- a capacidade de recepção é afetada pela mobilidade.



(a) Comportamento do sinal e da taxa de transmissão

(b) Relação entre o sinal e a taxa de transmissão

Figura 2. Qualidade do sinal e taxa de transmissão na presença de mobilidade.

No experimento apresentado na Figura 2(a), a taxa de recepção, que é inicialmente de 2 Mbps, é reduzida quando a unidade móvel é afastada do ponto de acesso. Essa redução deve-se à atuação do mecanismo de ARC do ponto de acesso, que reduz a taxa

de conexão ao identificar a degradação do sinal em função da mobilidade. Quando o ponto de acesso escolhe transmitir usando uma modulação mais robusta e, portanto, de menor capacidade de vazão, a taxa de recepção de 2 Mbps já não é mais alcançada pela unidade móvel. Usando como indicador da qualidade do sinal a razão entre a força do sinal e do ruído (*Signal-to-Noise Ratio* ou SNR), a Figura 2(b) oferece uma visualização da relação entre a qualidade do sinal e a taxa de recepção efetivamente alcançada pela unidade móvel.

O gráfico apresentado na Figura 2(b) pode ser dividido em três partes: a primeira, à direita, onde o valor de SNR é alto e a recepção normalmente alcança 2 Mbps. A segunda, ao centro, onde a taxa de recepção é mais instável, mas, em média, ainda alcança 2 Mbps. Outra, por fim, à esquerda, onde o valor de SNR é baixo e a taxa de recepção raramente atinge 2 Mbps. Baseados nesses três patamares de operação, podem ser definidos três limiares de adaptação α , β e γ , onde: α é o menor valor de SNR, de tal modo que a taxa de transmissão desejada é alcançada; β é o menor valor de SNR, de forma que a taxa de transmissão é aceitável; e γ é o menor valor de SNR, para o qual ainda se observa transmissão. Por exemplo, analisando-se a Figura 2(b), α , β e γ poderiam ser 1.15, 1.1 e 1.05, respectivamente.

6.1. Estratégia adaptativa baseada nos limiares de adaptação α , β e γ

Partindo da observação de que não se atingem taxas elevadas de transmissão quando a qualidade do sinal está degradada, os valores de SNR, α e β podem ser usados em um algoritmo adaptativo da seguinte maneira:

- **nível 0:** se $SNR \geq \alpha$, a transmissão continua normalmente, pois a transmissão deve estar transcorrendo normalmente;
- **nível 1:** se $\alpha > SNR \geq \beta$, o servidor deve transmitir de maneira conservadora. Um exemplo de estratégia conservadora é reduzir o tamanho dos pacotes, mas sem reduzir a taxa de transmissão, pois dessa forma reduz-se a probabilidade de perda de pacotes [Schiller 2000];
- **nível 2:** se $\beta > SNR \geq \gamma$, o servidor deve reduzir a taxa de transmissão, pois, como mostra a Figura 2(b), nessas condições a transmissão raramente atinge altas taxas de transmissão;
- **nível 3:** se $SNR < \gamma$, a transmissão é interrompida até que SNR volte a ser maior ou igual a γ , pois, para valores de SNR menores do que γ , a rede sem fio normalmente estará inoperante.

A estratégia adaptativa é organizada em níveis para que os clientes possam definir os valores de α , β e γ em função da qualidade de sua recepção. Dessa forma, não é preciso reportar ao servidor a qualidade do sinal e os limiares de adaptação; basta enviar o nível de operação. Em situações de uso real do algoritmo, as estratégias apropriadas para cada nível de operação devem ser implementadas em função da aplicação e da natureza dos dados. Por exemplo, para aplicações de transmissão de fluxos contínuos de vídeo, as estratégias poderiam ser: no nível 1, o envio redundante dos pacotes mais importantes ou o uso de códigos corretores; no nível 2, a redução da qualidade de compressão e, conseqüentemente, da taxa de transmissão ou o descarte de quadros; no nível 3, resta pouco a

fazer além de interromper a transmissão, mas pode-se ainda tentar notificar o usuário das condições de conexão ou tentar, quando disponíveis, outras formas de conexão.

Os valores de α , β e γ podem ser encontrados de maneira analítica e com intervenção humana, como fizemos em caráter ilustrativo na Seção 6, ou podem ainda ser descobertos automaticamente.

6.1.1. Descoberta automática de α , β e γ

A definição automática de γ é trivial; para fazê-lo, basta armazenar o menor valor de SNR dentre os valores de SNR medidos a cada recepção de pacotes. Os valores de α e β podem ser definidos a partir da correlação entre a taxa de perda de pacotes e o valor de SNR. Podemos definir α e β como os menores valores de SNR, de tal modo que as taxas de perdas de pacotes sejam, por exemplo, inferiores a 1% e 25%, respectivamente. Para fazer essa correlação, a cada segundo de transmissão, medimos a Taxa de Perdas de Pacotes (TPP) e o valor de SNR e atualizamos α e β segundo o pseudocódigo apresentado na Figura 3.

```
se [(TPP < 1%) e (SNR <  $\alpha$ )] então
   $\alpha \leftarrow \alpha - [(\alpha - SNR)/16]$ 
fim se
se [(TPP < 25%) e (SNR <  $\beta$ )] então
   $\beta \leftarrow \beta - [(\beta - SNR)/16]$ 
fim se
```

Figura 3. Algoritmo para atualização de α e β .

Os valores de α e β são inicializados com os primeiros valores obtidos de SNR. A fim de aumentar a estabilidade do algoritmo, os valores de TPP e SNR utilizados para atualização de α e β são, na verdade, a média das últimas cinco medições de TPP e SNR. Para evitar que casos excepcionais afetem demasiadamente os valores de α e β , a diferença entre os coeficientes e o SNR é amortizada; em nossa implementação, usamos um fator de amortização 16.

Realizamos experimentos com diferentes valores de amortização. Se o fator de amortização for pequeno demais, por exemplo 2, a convergência de α e β é mais rápida, mas, em contrapartida, os valores de α e β absorvem casos excepcionais na transmissão (casos em que valores excepcionalmente baixos de SNR conseguem — mesmo que por curtos espaços de tempo — taxas altas de transmissão). Se o fator de amortização for grande demais, a convergência dos valores de α e β torna-se lenta.

A amortização também poderia ser implementada usando-se, ao invés de apenas um fator de amortização fixo, funções mais complexas — em que a convergência dos valores de α e β fosse rápida e as exceções fossem evitadas. Em nossos experimentos, avaliamos a utilização de diferentes funções de amortização, testamos funções lineares e não lineares. No entanto, a estratégia de usar um fator de amortização de valor 16 mostrou-se mais estável e de aplicação mais geral, além de ser mais simples.

A Figura 4 demonstra o processo de busca automática de α e β . Nesta execução,

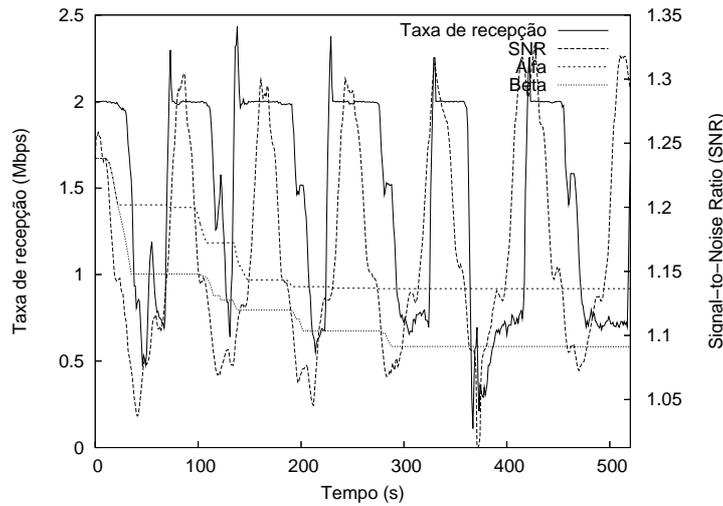


Figura 4. Convergência de α e β (TPPs 1% e 25% e fator de amortização 16).

usamos o algoritmo apresentado na Figura 3 (com fator de amortização 16) para correlacionar os valores de α e β às taxas de perda de pacotes de 1% e 25%, respectivamente. Na figura, a variação do valor de SNR ilustra a mobilidade presente no experimento, que foi a razão principal das quedas na taxa de recepção. A figura mostra ainda que os valores de α e β convergem para 1,13 e 1,09, respectivamente. Observe que a transmissão sofre quedas bruscas na taxa de recepção e que essas quedas ocorrem — não por acaso — quando o valor de SNR cai ao patamar de α . Repare também que o instante da queda na taxa de transmissão e o instante em que SNR “cruza” α aproximam-se conforme o algoritmo ajusta os valores de α . Em outras palavras, a aproximação de SNR do valor de α é um indício forte de problemas futuros na conexão. Note também que a transmissão normalmente está comprometida quando SNR cai ao patamar de β .

Resta mencionar que, dado que a função usada para atualização de α e β é monotonicamente decrescente, é preciso estabelecer um critério de parada para o algoritmo; caso contrário, a acuidade dos limiares será deteriorada pela — sutil mas contínua — absorção dos casos excepcionais que podem ocorrer ao longo da transmissão. Recomendamos que se use apenas a estabilidade do valor de α como critério de parada; pois, em nossos experimentos, verificamos que normalmente pode-se parar o algoritmo quando as atualizações de α forem da ordem de 1% do seu valor absoluto. Além disso, usamos apenas α no critério de parada, pois na faixa de operação de β a transmissão é menos estável, o que dificulta a definição de um critério de convergência.

Contudo, apesar da utilidade da determinação automática dos limiares de adaptação, notamos que na maioria das situações práticas os valores para α , β e γ precisam ser determinados apenas uma vez para cada ambiente de execução. Após terem sido determinados esses limiares de adaptação, uma nova busca só se faz necessária caso ocorram mudanças no ambiente, tais como a execução de *handoffs* ou o uso de um NIC diferente.

6.2. Implementação

Na implementação da nova estratégia adaptativa proposta neste trabalho, a unidade móvel monitora os valores de SNR e (a cada segundo) notifica ao servidor o nível de operação. O servidor, por sua vez, ajusta a taxa de transmissão de acordo com o nível de opera-

ção. Para aumentar a estabilidade das adaptações (histerese), o servidor altera o modo de transmissão apenas após receber duas notificações consecutivas contendo o mesmo valor de operação. O mecanismo de troca de mensagens entre o servidor e a unidade móvel é implementado usando-se datagramas UDP. Escolhemos usar UDP e não TCP, como seria natural para mensagens de controle, devido ao comportamento normalmente bloqueante das chamadas TCP.

Entretanto, se $SNR < \gamma$, como podemos notificar o servidor? Quando o valor de SNR é menor do que γ , a rede provavelmente está inoperante. Sendo assim, é preciso implementar no servidor mecanismos que permitam adaptar os fluxos mesmo durante esses períodos de desconexão. Desse modo, implementamos um mecanismo de *timeout* em que, caso o servidor não receba nenhuma notificação do cliente durante três segundos, o servidor assume o nível três de operação e interrompe a transmissão, até que uma nova notificação de operação seja recebida.

Em suma, a unidade móvel periodicamente envia datagramas UDP notificando o seu nível de operação e o servidor interrompe a transmissão caso não receba essas notificações. As estratégias de adaptação implementadas foram: no nível de operação 0, o servidor transmite 125 pacotes de 2.000 octetos⁵ por segundo (totalizando 2 Mbps de taxa de transmissão); no nível 1, 250 pacotes de 1.000 octetos por segundo (ainda 2 Mbps); no nível 2, 125 pacotes de 1.000 octetos por segundo (a taxa de transmissão é reduzida para 1 Mbps); e, no nível 3, como dito anteriormente, a transmissão é interrompida.

6.3. Ambiente experimental

Os experimentos apresentados a seguir foram executados em um ambiente composto por: um computador servidor (Dual-Pentium III, com 2 processadores de 1 GHz e 512 MB de memória; sistema operacional Debian, distribuição Woody), um computador portátil (Pentium III 800 MHz com 128 MB de memória; sistema operacional Debian, distribuição Sarge) e um ponto de acesso IEEE 802.11 (Samsung, modelo SWL-4000 AP). A conectividade entre o computador móvel e o ponto de acesso foi realizada utilizando-se, principalmente, um cartão PCMCIA IEEE 802.11b (Lucent, modelo WaveLAN Silver PC24E-H-FC, *chipset* Orinoco). Também fizemos avaliações utilizando diferentes NICs e pontos de acesso, inclusive avaliamos redes 802.11g, mas, para maior consistência dos resultados, todos os experimentos apresentados neste artigo foram obtidos usando-se o ambiente experimental acima descrito. Apesar da dificuldade para realizar experimentos práticos sobre questões de mobilidade, nossos testes foram conduzidos exaustivamente: os resultados apresentados foram obtidos de pelo menos 12 execuções dos algoritmos, sendo que todas as execuções foram consistentes entre si.

6.4. Desempenho do algoritmo adaptativo

A Figura 5 mostra a comparação entre um fluxo de dados adaptativo e outro não adaptativo. Na execução do algoritmo adaptativo, foram usados valores de $\alpha = 1.13$ e $\beta = 1.09$. A figura mostra que, durante a fase inicial do experimento, aproximadamente até os 30s,

⁵Como mencionado na Seção 4.1, o tamanho dos pacotes, quando possível, não deve superar 1.472 octetos. Entretanto, para gerar fluxos de 2 Mbps, ou mais, tais como os necessários para transmissão de TV Digital, é preciso utilizar pacotes maiores do que esse valor; por essa razão utilizamos pacotes de 2.000 octetos. Além disso, a utilização de valores “redondos” simplifica apresentação dos algoritmos, sem, no entanto, afetar a sua generalidade.

ambos os algoritmos alcançam a taxa de recepção de 2 Mbps. Aos 30s, porém, os algoritmos começam a sofrer os efeitos da mobilidade e da queda na qualidade do sinal. Mas, durante essa fase, o algoritmo adaptativo opera uma série de adaptações com o objetivo de reduzir o desperdício da capacidade da rede. Quando a qualidade da conexão volta a ser alta, entretanto, uma questão mostra-se fundamental para a diferença de comportamento entre os algoritmos: o algoritmo adaptativo retoma mais rapidamente as taxas normais de recepção do que o algoritmo não adaptativo.

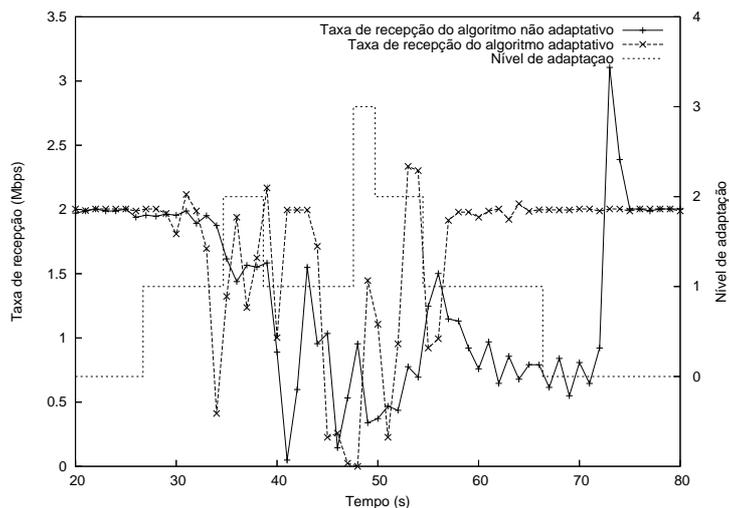


Figura 5. Comparação entre o algoritmo adaptativo e o não adaptativo.

O algoritmo adaptativo recupera-se mais rapidamente do que o não adaptativo por duas razões. Primeiro, porque o algoritmo não adaptativo mantém a taxa de transmissão em 2 Mbps e, desse modo, sobrecarrega a capacidade de transmissão do ponto de acesso. Segundo, e mais importante, as adaptações executadas pelo algoritmo adaptativo reduzem as perdas de pacotes, evitando assim a necessidade de atuação do mecanismo ARC e, conseqüentemente, a redução da taxa de transmissão. É importante lembrar que as métricas usadas pelo mecanismo de ARC para adaptação da taxa de transmissão baseiam-se principalmente na qualidade da transmissão e da recepção em termos de perda de quadros e pacotes, número de mensagens ACK perdidas, vazão etc. [Haratcherev et al. 2005]; e não na qualidade do sinal. A qualidade do sinal é apenas uma métrica auxiliar para o mecanismo de ARC, porque é difícil estabelecer uma relação apenas entre a qualidade do sinal e a respectiva taxa ótima de transmissão; afinal, isso depende também de outros fatores, como a sensibilidade dos NICs usados, o tamanho dos pacotes e o volume de dados trafegados. Portanto, ao adaptar os fluxos, mesmo não interferindo diretamente na força do sinal, reduzimos a necessidade de atuação dos mecanismos de ARC.

Além disso, a fim de evitar excessivas alterações na taxa de transmissão, que poderiam levar à instabilidade do sistema, o mecanismo de ARC aguarda certo período de tempo antes de retomar as taxas mais altas de transmissão [Haratcherev et al. 2005] e, por essa razão, o fluxo não adaptativo continua sofrendo os efeitos da mobilidade, mesmo já estando próximo ao ponto de acesso.

A Tabela 1 resume os resultados obtidos pelos algoritmos adaptativo e não adaptativo. O cliente do algoritmo adaptativo recebeu efetivamente mais dados do que o não adaptativo, mesmo tendo o servidor enviado mais dados para o algoritmo não adaptativo

<i>Algoritmo</i>	<i>Transmitido</i>	<i>Recebido</i>	<i>Taxa Média</i>	<i>Eficiência</i>
Não adaptativo	1.848 Mbits	1.390 Mbits	1,505 Mbps	0,75
Adaptativo	1.659 Mbits	1.604 Mbits	1,736 Mbps	0,96

Tabela 1. Comparação entre o algoritmo adaptativo e o não adaptativo.

do que para o adaptativo. Em números, o algoritmo adaptativo alcançou 96% de eficiência⁶ contra apenas 75% do algoritmo não adaptativo. Além disso, atingiu taxas médias de recepção 15% maiores do que o algoritmo não adaptativo (1,736 Mbps contra 1,505 Mbps). Ademais, poderíamos traçar cenários em que essa diferença seria ainda mais acentuada, como por exemplo experimentos sujeitos a longos períodos de desconexão.

7. Conclusões

Neste artigo apresentamos os principais problemas relacionados ao desenvolvimento de aplicações adaptativas para redes IEEE 802.11, assim como as técnicas que podem ser usadas para resolver esses problemas.

A contribuição central deste trabalho, contudo, foi a proposta de uma nova estratégia adaptativa que, ao adequar as transmissões à qualidade do sinal, reduz o desperdício de recursos da rede e, conseqüentemente, aumenta a eficiência das transmissões. Destacamos que o algoritmo adaptativo proposto obteve 96% de eficiência nas transmissões sujeitas à mobilidade, enquanto transmissões não adaptativas atingiram apenas 75% de eficiência. Além disso, a estratégia adaptativa proposta é suficientemente genérica para ser aplicada em diferentes ambientes de redes sem fio, sendo que grande parte dessa generalidade deve-se ao algoritmo para a descoberta automática dos limiares de adaptação α , β e γ .

As redes IEEE 802.11, até o presente momento, foram erroneamente tratadas como redes *Ethernet*; isso deve-se principalmente ao fato de que não existiam formas padronizadas de interação entre as aplicações e as interfaces para essas redes. Entretanto, esta pesquisa demonstrou, por meio de implementações concretas, que as peculiaridades das redes IEEE 802.11 podem — e por vezes devem — ser tratadas pelos desenvolvedores na camada de aplicação. Alguns servidores de fluxos já implementam estratégias adaptativas, porém estas estratégias são inadequadas às redes sem fio; é preciso adequar essas implementações. Ademais, este trabalho também foi elaborado com o objetivo de servir como referência para o desenvolvimento de aplicações cada vez mais adequadas às novas realidades trazidas pelas redes sem fio e, em especial, pelas redes IEEE 802.11. Desse modo, esperamos incentivar a pesquisa e o desenvolvimento de técnicas adaptativas para redes IEEE 802.11, cujo aprimoramento nos parece um tema urgente.

Referências

Balakrishnan, H., Padmanabhan, V. N., Seshan, S., and Katz, R. H. (1997). A comparison of mechanisms for improving TCP performance over wireless links. *IEEE/ACM Transactions on Networking (TON)*, 5(6):756–769.

⁶Eficiência é definida como a quantidade de dados transmitida dividida pela quantidade de dados recebida. Desse modo, a eficiência é tão melhor quanto mais próxima for de 1.

- da Conceição, A. F. and Kon, F. (2003). Adaptação de fluxos contínuos UDP sobre redes IEEE 802.11b. In *V Workshop de comunicação sem fio e computação móvel (WCSF)*, pages 91–101, São Lourenço-MG, Brazil.
- da Conceição, A. F. and Kon, F. (2006). O uso de pares de pacotes para monitoração da taxa de transmissão e da capacidade de vazão em redes IEEE 802.11. In *24th Brazilian Symposium on Computer Networks (SBRC)*, Curitiba-PR, Brazil.
- Etoh, M. and Yoshimura, T. (2005). Advances in wireless video delivery. *Proceedings of the IEEE*, 93(1):111–122.
- Gast, M. S. (2002). *802.11 Wireless Networks. The definitive guide*. O'Reilly Media.
- Haratcherev, I., Taal, J., Langendoen, K., Lagendijk, R., and Sips, H. (2005). Automatic IEEE 802.11 rate control for streaming applications. *Wireless Communications and Mobile Computing*, 5:421–427. Special issue on Radio Link and Transport Protocol Engineering for Future-Generation Wireless Mobile Data Networks.
- Heusse, M., Rousseau, F., Berger-Sabbatel, G., and Duda, A. (2003). Performance anomaly of 802.11b. In *22nd Conference on Computer Communications (INFOCOM)*, volume 2, pages 836–843, San Francisco, USA.
- Ikkurthy, P. and Labrador, M. A. (2002). Characterization of MPEG-4 traffic over IEEE 802.11b wireless LANs. In *Annual IEEE Conference on Local Computer Networks (LCN)*, volume 27, pages 421–430, Tampa, USA.
- Larzon, L.-A., Degermark, M., Pink, S., Jonsson (Ed.), L.-E., and Fairhurst (Ed.), G. (2004). RFC 3828 - The Lightweight User Datagram Protocol (UDP-Lite).
- Muñoz, L., Garcia, M., Choque, J., and Mähönen, R. A. P. (2001). Optimizing internet flows over IEEE 802.11b wireless local area networks: A performance-enhancing proxy based on forward error correction. *IEEE Communication Magazine*, 39(12):60–67.
- Pilosof, S., Ramjee, R., Raz, D., Shavitt, Y., and Sinha, P. (2003). Understanding TCP fairness over wireless LAN. In *22nd Conference on Computer Communications (INFOCOM)*, volume 2, pages 863–872, San Francisco, USA.
- Ramani, I. and Savage, S. (2005). SyncScan: Practical fast handoff for 802.11 infrastructure networks. In *24th Conference on Computer Communications (INFOCOM)*, Miami, USA.
- Schiller, J. (2000). *Mobile Communications*. Addison Wesley.
- Sinha, R. and Papadopoulos, C. (2004). An adaptive multiple retransmission technique for continuous media streams. In *14th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, pages 16–21, Kinsale, Ireland.
- Tan, G. and Guttag, J. (2005). The 802.11 MAC protocol leads to inefficient equilibria. In *24th Conference on Computer Communications (INFOCOM)*, Miami, USA.
- Vandalore, B., chi Feng, W., and Raj Jain, S. F. (2001). A survey of application layer techniques for adaptive streaming of multimedia. *Real-Time Imaging*, 7(3):221–235.