



LUCAS JOSÉ DUARTE DE SOUZA

**ALGORITMOS DE CONTROLE DE TAXA
DE TRANSMISSÃO EM REDES 802.11**

LAVRAS – MG

2014

LUCAS JOSÉ DUARTE DE SOUZA

**ALGORITMOS DE CONTROLE DE TAXA DE TRANSMISSÃO EM
REDES 802.11**

Monografia de Graduação apresentada ao
Departamento de Ciência da Computação para
obtenção do título de Bacharel em “Ciência da
Computação”

Orientador

Prof. Dr. Neumar Malheiros

LAVRAS – MG

2014

LUCAS JOSÉ DUARTE DE SOUZA

ALGORITMOS DE CONTROLE DE TAXA DE TRANSMISSÃO EM
REDES 802.11

Monografia de Graduação apresentada ao
Departamento de Ciência da Computação para
obtenção do título de Bacharel em "Ciência da
Computação"

APROVADA em 13 de Fevereiro de 2014.

Prof. Me. Hermes Pimenta de Moraes Junior

UFLA


Prof. Dr. Joaquim Quinteiro Gchôa

UFLA


Prof. Dr. Neumar Malheiros
(Orientador)

LAVRAS – MG

2014

Primeiramente, dedico este trabalho aos meus pais, e espero que estejam orgulhosos por ter conseguido seguir o caminho que tanto me incentivaram e apoiaram. Sem eles não seria possível alcançar este objetivo. Dedico também ao meu orientador, Neumar Malheiros, a qual me deu bastante apoio e suporte durante a realização deste trabalho.

RESUMO

Vários dispositivos possuem interfaces com a tecnologia de redes sem fio IEEE 802.11. Isto faz com que existam ambientes onde muitos dispositivos disputam o acesso ao meio. Dessa forma, é importante que existam mecanismos que otimizem o uso dos recursos da rede, um deles é o controle de taxa transmissão de dados. Esta taxa é o resultado da combinação de esquemas de codificação e modulação do sinal. O dispositivo deve ser capaz de escolher automaticamente a taxa de transmissão ótima, para maximizar a utilização dos recursos da rede. Neste contexto, o problema de controle de taxa de transmissão consiste em ajustar dinamicamente a taxa de transmissão em função das condições do canal com o objetivo de maximizar o desempenho da rede. Neste trabalho, foi realizada uma avaliação de desempenho de vários algoritmos de controle da taxa de transmissão. A avaliação foi realizada por meio de simulações no NS-3 considerando diversos cenários de redes locais sem fio.

Palavras-Chave: Redes de Computadores; 802.11; controle da taxa.

SUMÁRIO

| | | |
|----------|---|-----------|
| 1 | Introdução | 1 |
| 1.1 | Definição do Problema | 1 |
| 1.2 | Objetivos | 2 |
| 1.3 | Motivação | 3 |
| 1.4 | Organização do texto | 3 |
| 2 | Redes sem fio | 5 |
| 2.1 | A Subcamada de Acesso ao Meio (MAC) | 9 |
| 2.2 | Modulação e Taxas de Transmissão | 11 |
| 2.3 | Controle da Taxa de Transmissão | 12 |
| 3 | Algoritmos de Controle da Taxa | 14 |
| 3.1 | Algoritmos baseados em estatística | 14 |
| 3.2 | Algoritmos baseados em qualidade | 20 |
| 3.3 | Algoritmos Híbridos | 23 |
| 3.4 | Considerações finais | 25 |
| 4 | Metodologia | 27 |
| 5 | Implementação do CORA | 30 |
| 5.1 | COgnitive Rate Adaptation | 30 |
| 5.2 | A Implementação | 33 |
| 6 | Avaliação de desempenho | 36 |
| 6.1 | Comportamento das taxas de transmissão nominais | 36 |
| 6.2 | Comportamento do CORA | 39 |
| 6.3 | Comportamento dos algoritmos | 50 |
| 6.4 | Desempenho dos algoritmos | 54 |

| | |
|-------------------------------|-----------|
| 7 Conclusão | 62 |
| A Código fonte do CORA | 66 |

LISTA DE FIGURAS

| | | |
|-----|--|----|
| 2.1 | Classificação das redes sem fio | 6 |
| 2.2 | Estruturas de redes sem fio | 8 |
| 2.3 | Funcionamento do CSMA/CA | 10 |
| 3.1 | Algoritmos baseados em estatística | 14 |
| 3.2 | Exemplo do funcionamento do AARF | 16 |
| 3.3 | Algoritmos baseados em qualidade | 20 |
| 3.4 | Algoritmos Híbridos | 23 |
| 3.5 | Arquitetura do EasiRA | 25 |
| 5.1 | Ciclo de controle do CORA | 31 |
| 5.2 | Diagrama da implementação do CORA | 34 |
| 6.1 | Topologia composta de um AP e uma estação móvel. | 37 |
| 6.2 | Vazão instantânea em função da qualidade do sinal ao longo do tempo para cada taxa de transmissão fixa em redes 802.11a configurados com os parâmetros velocidade - 1.5 m/s - e tamanho de pacotes - 1460 bytes. | 38 |
| 6.3 | Vazão média alcançada pelas taxas de transmissão fixas em redes 802.11a infraestruturadas em cenários com velocidade igual a 1.5m/s. | 39 |
| 6.4 | Vazão agregada média em função do intervalo para diferentes combinações de α e do desvio padrão em uma rede 802.11a infraestruturada. | 41 |
| 6.5 | Vazão agregada média em função do α para diferentes combinações de intervalo e do desvio padrão em uma rede 802.11a infraestruturada. | 42 |
| 6.6 | Vazão agregada média em função do desvio padrão para diferentes combinações de intervalo e de α em uma rede 802.11a infraestruturada. | 43 |

| | | |
|------|--|----|
| 6.7 | Topologia com múltiplas estações e um AP representada por um cenário contendo 6 estações | 45 |
| 6.8 | Vazão agregada média em função do intervalo para diferentes combinações de alpha e do desvio padrão em uma rede 802.11a infraestruturada. | 46 |
| 6.9 | Vazão agregada média em função do alpha para diferentes combinações de intervalo e do desvio padrão em uma rede 802.11a infraestruturada. | 47 |
| 6.10 | Vazão agregada média em função do desvio padrão para diferentes combinações de intervalo e de alpha em uma rede 802.11a infraestruturada. | 48 |
| 6.11 | Controle da taxa em função do tempo realizado pelo ARF em redes 802.11a infraestruturadas com velocidade igual a 1.5 m/s e tamanho de pacote igual a 1460 bytes | 51 |
| 6.12 | Controle da taxa em função do tempo realizado pelo CARA em redes 802.11a infraestruturadas com velocidade igual a 1.5 m/s e tamanho de pacote igual a 1460 bytes | 51 |
| 6.13 | Controle da taxa em função do tempo realizado pelo RRAA em redes 802.11a infraestruturadas com velocidade igual a 1.5 m/s e tamanho de pacote igual a 1460 bytes | 52 |
| 6.14 | Controle da taxa em função do tempo realizado pelo RBAR em redes 802.11a infraestruturadas com velocidade igual a 1.5 m/s e tamanho de pacote igual a 1460 bytes | 52 |
| 6.15 | Controle da taxa em função do tempo realizado pelo Minstrel em redes 802.11a infraestruturadas com velocidade igual a 1.5 m/s e tamanho de pacote igual a 1460 bytes | 52 |

| | | |
|------|--|----|
| 6.16 | Controle da taxa em função do tempo realizado pelo CORA em redes 802.11a infraestruturas com velocidade igual a 1.5 m/s e tamanho de pacote igual a 1460 bytes | 52 |
| 6.17 | Vazão instantânea em função da qualidade do sinal ao longo do tempo para cada algoritmo em redes 802.11a configurados com os parâmetros velocidade - 1.5 m/s - e tamanho de pacotes - 1460 bytes . | 53 |
| 6.18 | Vazão média alcançada pelos algoritmos e taxas de transmissão fixas em redes 802.11a infraestruturas em cenários com velocidade igual a 1.5 m/s. | 55 |
| 6.19 | Taxa de perda de pacotes média alcançada pelos algoritmos e taxas de transmissão fixas em redes 802.11a infraestruturas em cenários com velocidade igual a 1.5 m/s | 56 |
| 6.20 | Taxa de perda de pacote média alcançada pelos algoritmos de controle da taxa em redes 802.11a infraestruturas com múltiplos nós | 57 |
| 6.21 | Vazão agregada média alcançada pelos algoritmos de controle da taxa em redes 802.11a infraestruturas com múltiplos nós | 58 |
| 6.22 | Vazão média por fluxo alcançada pelos algoritmos de controle da taxa em redes 802.11a infraestruturas com múltiplos nós | 59 |
| 6.23 | Nível de justiça dos algoritmos de controle de taxa simulados em redes com múltiplos nós | 60 |

LISTA DE TABELAS

| | | |
|-----|--|----|
| 2.1 | Versões do Padrão 802.11 | 7 |
| 2.2 | Modulações e taxas | 12 |
| 3.1 | Tabela Comparativa | 26 |
| 6.1 | Configurações dos parâmetros de simulação. | 37 |
| 6.2 | Configurações da topologia contendo múltiplos nós comunicando à um AP | 45 |
| 6.3 | Vazão agregada média alcançada por uma amostra das 12 melhores configurações do CORA em uma rede 802.11a infraestruturada em um cenário com 12 estações. | 49 |
| 6.4 | Comparação do ganho percentual do CORA aos demais algoritmos no cenário com velocidade igual a 1.5 m/s e tamanho dos pacotes igual a 1460 bytes | 56 |

1 INTRODUÇÃO

Os dispositivos que acessam a Internet através de redes sem fio já fazem parte do cotidiano das pessoas. Além de computadores, vários outros dispositivos possuem interfaces com esta tecnologia, como impressoras, câmeras fotográficas, dispositivos de armazenamento, celulares, "tablets", etc. Dessa forma, usuários podem se conectar à Internet em diversos lugares como, restaurantes, bares, aeroportos, etc.

Uma justificativa para o crescimento das redes sem fio é a facilidade que usuários encontram ao utilizar dispositivos que implementam tais tecnologias, já que elas não precisam de instalações de cabos e fios e possuem custo financeiro relativamente baixo.

1.1 Definição do Problema

A taxa de transmissão em redes sem fio IEEE 802.11 é de suma importância. Vários computadores e outros dispositivos implementam interfaces com esta tecnologia. Isto faz com que existam ambientes onde muitos dispositivos disputam o acesso ao meio. Dessa forma, é importante que existam mecanismos que otimizam o uso dos recursos da rede de acesso sem fio, um deles é o controle de taxa de transmissão de dados. Esta taxa é o resultado de esquemas de codificação e modulação do sinal de transmissão. Em outras palavras, uma interface de rede sem fio implementa diversos esquemas de modulação. Cada modulação corresponde a uma taxa de transmissão nominal. Isto quer dizer que, na prática, a vazão obtida é menor que a taxa de transmissão nominal, devido à sobrecarga de informações de controle dos protocolos e perda de pacotes resultantes de ruído e interferência no meio de comunicação. A métrica de desempenho denominada vazão pode ser definida como a quantidade de dados efetivamente transmitidos em um intervalo de tempo. Por exemplo, um dispositivo pode utilizar uma taxa nominal de 36Mbps, mas, na prática, obter uma vazão menor, de 30 Mbps.

Esquemas de modulação que resultam em baixas taxas de transmissão são mais robustos em ambientes com alto grau de interferência e por isto resultam em uma área de cobertura maior. Por outro lado, nos esquemas de modulação que resultam em altas taxas, a qualidade do sinal é mais suscetível a ruídos no meio, dessa forma pode acontecer um decremento maior na taxa de dados efetivamente transmitidos.

O dispositivo deve ser capaz de escolher a taxa de transmissão ótima automaticamente (sem a necessidade de intervenção de algum administrador) para maximizar a utilização dos recursos da rede. Ou seja, dada as condições do canal sem fio, existe uma taxa ótima que faz com que a utilização dos recursos da rede sejam maximizados.

Neste contexto, o problema de controle de taxa de transmissão consiste em ajustar dinamicamente a taxa de transmissão em função das condições do canal com o objetivo de maximizar o desempenho da rede.

1.2 Objetivos

O objetivo deste trabalho é estudar os principais algoritmos de adaptação de taxa de transmissão em redes IEEE 802.11 e realizar uma avaliação de desempenho para identificar limitações e vantagens desses algoritmos. Os objetivos específicos destes trabalho são:

- Aprofundar conhecimentos sobre redes de acesso 802.11;
- Aprender a usar ferramentas de simulação para avaliação de desempenho de redes;
- Estudar e avaliar os principais algoritmos para o controle de taxa de transmissão em redes sem fio IEEE 802.11;

- Implementar o algoritmo Cognitive Rate Adaptation (CORA). Este algoritmo será apresentado no Capítulo 5.

1.3 Motivação

O padrão 802.11 já tem sido amplamente utilizado há quase 15 anos. Desde sua primeira publicação, foram lançadas inúmeras versões. Por suas facilidades, o 802.11 está cada vez mais inserido no mercado fazendo com que vários computadores utilizem esta tecnologia. De acordo com a *JiWire* (JIWIRE, 2014), até janeiro de 2014, existiam 870 milhões de *hotspots* no mundo. Esta estatística mostra a importância das redes sem fio, fazendo com que o interesse em investir nesta tecnologia continue crescendo. E, como a cada versão desenvolvida o desempenho é melhorado, é possível que, em um futuro próximo, qualquer tipo de comunicação seja feita com esta tecnologia.

O padrão IEEE 802.11, diz que é obrigatório que qualquer dispositivo utilize algum algoritmo de controle da taxa de transmissão. Mas, como não é dito qual deve ser utilizado, o fabricante fica livre para desenvolver e implementar seu próprio algoritmo de adaptação da taxa. Assim, existem vários algoritmos de controle de taxa. No entanto, estes algoritmos ainda apresentam algumas limitações, que serão descritas no Capítulo 3, de forma que ainda não há uma solução ótima que atenda todas as questões envolvidas. Assim, ainda há melhoramentos a serem feitos nestes algoritmos.

1.4 Organização do texto

Este texto está organizado como descrito a seguir. No Capítulo 2, apresenta-se os conceitos básicos para a compreensão do trabalho. Os assuntos abordados são: conceitos básicos sobre redes de comunicação, redes sem fio, o padrão 802.11, modulação e taxas de transmissão. No Capítulo 3, mostra-se uma revisão bibliográfica

dos principais algoritmos de controle da taxa de transmissão. Estes algoritmos são divididos em três classes: algoritmos baseados em estatística, algoritmos baseados em qualidade e algoritmos híbridos. No Capítulo 4, descreve-se a metodologia aplicada no trabalho. São apresentados o modelo de pesquisa e as atividades propostas. No Capítulo 5, é descrita a implementação do algoritmo CORA. Como o simulador é de código fonte aberto foi possível implementar o CORA sem alterá-lo. No Capítulo 6, é apresentada uma análise de desempenho dos algoritmos de controle da taxa. Foram realizados quatro estudos de casos em redes 802.11.

2 REDES SEM FIO

Neste capítulo, são apresentados conceitos básicos sobre os temas envolvidos neste trabalho. Primeiro, são descritos conceitos gerais sobre redes de comunicação e, em particular, redes sem fio. Por fim, são apresentadas as características principais do padrão 802.11. Os conceitos apresentados neste Capítulo estão de acordo com (KUROSE J. E ROSS, 2011).

Uma rede sem fio pode ser definida como um agrupamento de computadores ou outros dispositivos, conectados sem uso de cabos, que se comunicam através de radiofrequência ou infravermelho (KUROSE J. E ROSS, 2011).

Os serviços de redes sem fio estão em constante crescimento e vem se tornando cada vez mais comuns. Este interesse em redes sem fio se deve ao fato de facilidades como a ausência de cabos e baixos custos de instalação da rede.

Uma estrutura básica de uma rede sem fio deve conter os seguintes elementos: estação base, sistemas finais e padrões de comunicação sem fio. A Estação base é responsável pelo envio e o recebimento de pacotes de um sistema final que está conectado a ela. Torres de celulares e pontos de acesso de uma rede local sem fio são exemplos de uma estação base. Dispositivos conectam-se a uma estação base através de um padrão de comunicação sem fio. Estes padrões apresentam diferentes taxas de transmissão e área de alcance. Alguns deles são: WiFi, WiMax, Bluetooth e etc. Sistemas finais sem fio são dispositivos que executam aplicações finais. Pode ser um "notebook", "smartphone", "tablet", etc.

De acordo com a Figura 2.1, as redes sem fio podem ser classificadas em: WPAN(*Wireless Personal Area Network*), WLAN(*Wireless Local Area Network*), WMAN(*Wireless Metropolitan Network*) e WWAN(*Wireless Wide Area Network*).

As redes WPAN são mais conhecidas como redes pessoais de curta distância. O padrão mais usado para esta rede é o 802.15 mais conhecido como Bluetooth, que é basicamente uma tecnologia de “substituição de fios” e alcança aproximadamente 10 metros. Já as redes WLAN utilizam ondas de rádio para fazer

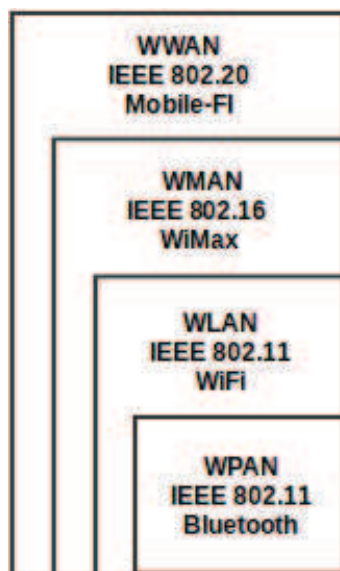


Figura 2.1: Classificação das redes sem fio

conexão à Internet ou entre redes. Neste caso, é usado o padrão 802.11 que é o padrão mais importante e usado em redes locais sem fio. As WMAN e WWAN, que implementa os padrões IEEE 802.16 e IEEE 802.20 respectivamente, normalmente alcançam entre 4 a 10 quilômetros, o que destina estas tecnologias principalmente a operadoras de telecomunicação.

O padrão 802.11 compreende a Subcamada de Acesso ao Meio (MAC) e a Camada Física. A camada MAC especifica a forma como é feito o acesso ao meio e a camada física descreve técnicas que devem ser utilizadas na transmissão de ondas eletromagnéticas (os protocolos nesta camada dependem do enlace).

O padrão 802.11 é dividido em vários outros padrões. A Tabela 2.1 apresenta algum dos principais padrões IEEE 802.11. O padrão 802.11b tem uma taxa de transmissão nominal máxima de 11 Mbps, operando na frequência de 2,4 a 2,485 Ghz. Esta frequência é de uso livre e, sendo assim, outros aparelhos como, forno micro-ondas, telefones, entre outros, podem usar a mesma faixa de frequência. Assim, a rede sem fio que opera neste intervalo pode disputar o espectro de

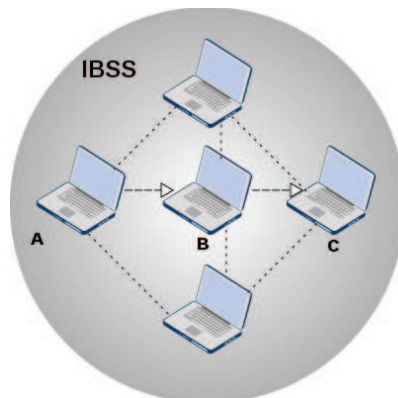
frequência com estes aparelhos. Já as LANs sem fio 802.11a podem operar a taxas e frequências significativamente mais altas, que giram em torno de 5,1Ghz à 5,8Ghz. As LANs sem fio 802.11g, operam na mesma faixa de frequência das LANs 802.11b, porém sua taxa de transmissão pode chegar a 54Mbps (KUROSE J. E ROSS, 2011). O padrão 802.11n pode chegar a uma taxa nominal de 450 Mbps e opera na mesma faixa de frequência que os padrões 802.11b e 802.11g. Este padrão faz uso de um esquema chamado *Multiple-Input Multiple-Output* (MIMO), que aumenta consideravelmente as taxas de transmissão de dados através da combinação de várias antenas.

Tabela 2.1: Versões do Padrão 802.11

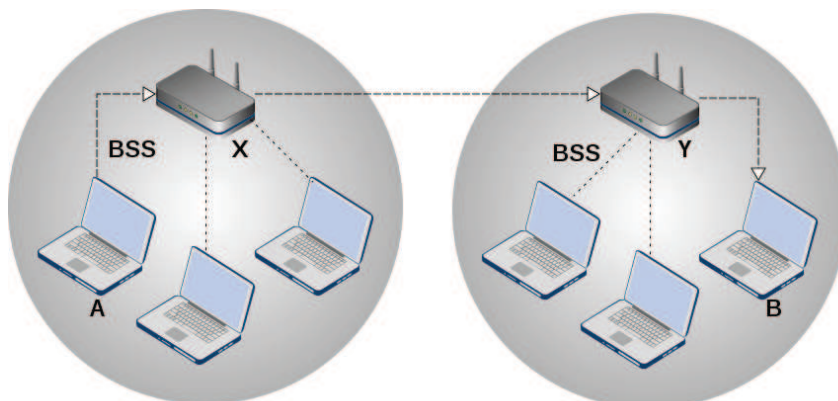
| Padrão 802.11 | Frenquência | Taxa de transmissão de dados |
|---------------|----------------|------------------------------|
| 802.11a | 5 Ghz | até 54 Mbps |
| 802.11b | 2.4 Ghz | até 11 Mbps |
| 802.11g | 2.4 Ghz | até 54 Mbps |
| 802.11n | 2.4 Ghz, 5 Ghz | até 450 Mbps |

Estas versões do padrão 802.11 possuem características parecidas. Todos utilizam o mesmo protocolo de acesso ao meio: o CSMA/CA. Estas versões também tem a capacidade de reduzir a taxa de transmissão para alcançar maiores distâncias. Como ilustrado na Figura 2.2, existem duas arquiteturas de redes sem fio: *Rede Infraestruturada e Rede Ad Hoc*.

Redes infraestruturadas ou BSS(*Basic Service Set*) são aquelas nas quais vários dispositivos estão conectados a um ponto de acesso (*AP - Access Point*), e este é conectado a uma infraestrutura cabeada, possibilitando a comunicação entre sistemas finais. Um exemplo é a infraestrutura de redes de celulares. Já uma rede *ad hoc* ou IBSS (*Independent Basic Service Set*) pode ser formada quando usuários que portam computadores móveis se encontram e trocam dados sem a necessidade do uso de um ponto de acesso. Sendo assim, Redes *ad hoc* são descentralizadas. Independente de qual arquitetura de rede é escolhida, existe suporte a múltiplas taxas de transmissão.



(a) Redes Ad Hoc



(b) Redes Infraestruturadas

Figura 2.2: Estruturas de redes sem fio

Considere o exemplo na Figura 2.2(a) que ilustra uma rede adhoc na qual o dispositivo A envia dados para o C. Para completar o trajeto, o dispositivo A manda estes dados para o dispositivo B que por fim manda os dados para o dispositivo C. Já no exemplo da Figura 2.2(b), o dispositivo A, envia dados para a estação base X, que transmite os dados para a estação Y, que por fim, envia os dados para o dispositivo B. Pode-se observar que em rede ad hoc há retransmissões efetuadas por sistemas finais. Por outro lado, rede infraestruturada não tem esta capacidade, já que os sistemas finais só transmitem dados para uma estação base.

2.1 A Subcamada de Acesso ao Meio (MAC)

O IEEE 802.11 disponibiliza dois métodos para controlar o acesso ao meio: DCF (*Distributed Coordination Function*) e PCF (*Point Coordination Function*). O DCF é obrigatório e é implementado em Redes Infraestruturadas e Redes *ad hoc*. Já o PCF só pode ser utilizado em Redes Infraestruturadas. Ele é opcional e quase não é utilizado pelos fabricantes.

Como várias estações podem transmitir ao mesmo tempo sobre o mesmo canal, é preciso um protocolo de acesso múltiplo para coordenar as ações. Este protocolo é chamado de CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*). Basicamente o DCF é uma implementação do protocolo CSMA/CA. Neste protocolo, cada elemento da rede verifica se o canal está livre antes de transmitir algum quadro e deixa de transmitir quando percebe que o canal está ocupado. Assim, quando o meio fica livre, o dispositivo aguarda um intervalo chamado DIFS (*DCF InterFrame Space*). Após o fim do DIFS, é necessário calcular o *backoff*, que é um valor aleatório. Este valor pode ter sido iniciado em outrora, caso contrário, é necessário calcular um novo valor para ele.

O cálculo do *backoff* é feito da seguinte forma:

$$\textit{backoff} = \textit{random}() \times \textit{SlotTime} \quad (2.1)$$

Sendo *SlotTime* um intervalo de tempo estabelecido pela camada física e *random()* um valor aleatório determinado dentro do intervalo $[0, CW]$. A *CW* (*Contention Window*) é um número determinado dentro do intervalo, $31 \leq CW \leq 1023$. O resultado de *CW* é obtido através da seguinte fórmula, onde *a* é o número de retransmissões:

$$CW = 2^A - 1 \begin{cases} A = 5 + a & \text{se } 0 \leq a \leq 5 \\ A = 10 & \text{se } a > 5 \end{cases} \quad (2.2)$$

Com o tempo do *backoff* definido, é iniciada uma contagem regressiva, que é feita enquanto o meio estiver ocioso. Se o canal estiver ocupado, a contagem feita sobre o valor do *backoff* é parada. Quando o contador atinge zero (isto só acontece caso o meio esteja ocioso), o dispositivo transmite todo o quadro e fica à espera do reconhecimento que é feito através de um quadro de controle denominado ACK (*ACKnowledgment*). O intervalo para o envio do ACK é chamado de SIFS (*Short InterFrame Space*). Caso o emissor não receba o ACK neste intervalo, ele reenvia o quadro, entrando novamente na fase de *backoff* com um valor aleatório maior que o anterior. Se o número de retransmissões exceder um limite máximo, o quadro é considerado como perdido pela camada de enlace. Este esquema está representado na Figura 2.3.

Há também um esquema opcional que ajuda evitar colisões em ambientes congestionados ou com a presença de terminais ocultos. Para evitar este problema, é permitido que uma estação utilize dois quadros de controle: RTS (*Request to Send*) e CTS (*Clear to Send*). Quando um remetente precisa enviar um quadro, ele envia antes um quadro RTS indicando o tempo total para a transmissão. Quando o ponto de acesso recebe o RTS, ele responde em *broadcast* transmitindo o quadro CTS que permite, explicitamente, que o remetente envie o quadro e avisa a todas as outras estações que o meio está ocupado fazendo com que elas não tentem fazer transmissões durante o tempo reservado.



Figura 2.3: Funcionamento do CSMA/CA

2.2 Modulação e Taxas de Transmissão

Os conceitos descritos nesta seção estão de acordo com (CHAVES *et al.*, 2009). Os sinais enviados por um emissor são codificados de acordo com uma técnica de modulação antes de serem transmitidos. As informações são inseridas e transmitidas por uma onda eletromagnética portadora, que está de acordo com as propriedades do meio de transmissão usado. Estes dados quando chegam ao receptor são recuperados através do processo inverso, chamado demodulação.

O sistema final recebe um sinal eletromagnético com degradações devido a ruídos no ambiente. Uma forma de medir o ruído é através da razão da potência do sinal e a potência do ruído (*Signal to Noise Ratio - SNR*). O SNR é medido em decibéis e serve, principalmente, para facilitar que o receptor do sinal extraia o sinal transmitido de um ruído. Outra medida que permite avaliar as condições do canal é a taxa de erros de *bits* (*Bits Error Rate - BER*), que é a probabilidade de um *bit* ser recebido com erro pelo sistema final. Para que se possa utilizar uma taxa de transmissão alta é necessário que a taxa de erros de *bits* seja baixa. Mas como nem sempre existe a possibilidade de se operar a uma taxa alta, existem modulações e codificações que resultam em taxas baixas, mas que tem uma capacidade maior de recuperar *bits* perdidos.

A camada física suporta múltiplas taxas para melhorar o desempenho, lidando melhor com a interferência e ruídos no ambiente. Cada taxa é a combinação de técnicas de espalhamento espectral, da modulação e de um método para correção de erros.

Para calcular a taxa de transmissão, multiplica-se a taxa de codificação, a quantidade de bits por símbolos e a quantidade de símbolos transmitidos por segundo. Para se chegar a este resultado cada taxa de transmissão utiliza algum método para correção de erros e usa uma taxa de codificação que é a quantidade de bits codificados transmitidos para uma determinada quantidade de bits de dados.

Tabela 2.2: Modulações e taxas

| Taxa nominal de bits | Padrão (802.11) | Modulação | Espalhamento Espectral | Bits por Símbolo | Taxa de codificação | Símbolos por segundo |
|----------------------|-----------------|-----------|------------------------|------------------|---------------------|----------------------|
| 1 | b | BPSK | DSSS-BK | 1 | 1/11 | 11×10^6 |
| 2 | b | QPSK | DSSS-BK | 2 | 1/11 | 11×10^6 |
| 5.5 | b | QPSK | DSSS-CCK | 1 | 4/8 | 11×10^6 |
| 6 | a e g | BPSK | OFDM | 1 | 1/2 | 12×10^6 |
| 9 | a e g | BPSK | OFDM | 1 | 3/4 | 12×10^6 |
| 11 | b | QPSK | DSSS-CCK | 2 | 4/8 | 11×10^6 |
| 12 | a e g | QPSK | OFDM | 2 | 1/2 | 12×10^6 |
| 18 | a e g | QPSK | OFDM | 2 | 3/4 | 12×10^6 |
| 24 | a e g | QAM | OFDM | 4 | 1/2 | 12×10^6 |
| 36 | a e g | QAM | OFDM | 4 | 3/4 | 12×10^6 |
| 48 | a e g | QAM | OFDM | 6 | 2/3 | 12×10^6 |
| 54 | a e g | QAM | OFDM | 6 | 3/4 | 12×10^6 |

Assim, a modulação transforma os bits codificados em símbolos que são transmitidos através de uma onda portadora utilizando alguma técnica de espalhamento espectral. Estas técnicas estão relacionadas na Tabela 2.2.

Para os padrões 802.11a, 802.11b e 802.11g existem duas técnicas de modulação: PSK (*Phase Shift Keying*) e QAM (*Quadrature Amplitude Modulation*). A modulação PSK é a técnica mais usada em redes sem fio e é dividida em BPSK (*Binary PSK*) e QPSK (*Quadrature PSK*). A forma mais simples, a BPSK, utiliza duas ondas portadoras. A QPSK apresenta o dobro de eficiência espectral da BPSK, possibilitando um número maior de símbolos e, assim, transmitindo-se mais bits por símbolo. A QAM é uma técnica de modulação baseada em amplitude (como a AM) e fase (como a PSK). Ela permite melhor utilização da largura de banda, em comparação com as modulações QPSK e BPSK.

2.3 Controle da Taxa de Transmissão

Existem múltiplas taxas de transmissão nominal que são geradas pela possibilidade de inúmeras combinações de uma técnica de modulação e uma taxa de codificação. Isto é visto como uma grande vantagem, já que, com um número maior de opções

de taxas existe a flexibilidade para lidar com variações nas condições do meio de transmissão.

Ao longo do tempo e espaço, a propagação das ondas muda por conta da perda gradual da intensidade do sinal, crescimento de ruídos, movimentação dos objetos, concorrência pelo espectro de frequência com outros dispositivos, etc. Dessa forma, o algoritmo de controle da taxa deve encontrar a taxa ótima para maximizar o desempenho da rede de acordo com as condições atuais do canal. Assim, o algoritmo de controle da taxa precisa ter conhecimento sobre as alterações no meio para poder aumentar ou diminuir a taxa quando necessário.

Taxas mais robustas (baixas), fazem com que as transmissões sofram menos interferência do meio, ou seja, a razão entre sinal e ruído (SNR) é maior e a taxa de erro de bits (BER) é menor. Mas, por outro lado, uma taxa de transmissão baixa significa uma vazão de dados limitada.

Já as taxas mais altas podem obter uma vazão maior, mas sofrem mais interferência do meio, diminuindo a razão sinal/ruído e aumentando a taxa de erro de bits. Isto acaba fazendo com que estas taxas tenham uma quantidade menor de dados efetivamente transmitidos quando as condições do meio são muito ruins. Também, à medida que o dispositivo se distancia do ponto de acesso, a razão SNR tende a diminuir, já que o sinal é baixo. Os algoritmos de controle da taxa devem ser capazes de perceber um aumento na BER e diminuir a taxa.

Muitos algoritmos assumem que perdas de quadros (não recebimentos de ACKs) surgem da degradação do canal e servem para indicar a qualidade do canal e conseqüentemente a taxa de transmissão a ser utilizada. Como no padrão 802.11 não há nenhum recurso para que o receptor informe ao emissor a qualidade do canal, muitos algoritmos baseiam-se somente nas perdas detectadas pelo emissor ou na qualidade do sinal (resultado da razão SNR) para tomar decisões e escolher a melhor taxa de transmissão para aquela situação.

3 ALGORITMOS DE CONTROLE DA TAXA

Neste capítulo, é apresentado um estudo detalhado dos principais algoritmos de controle da taxa. Esses algoritmos são divididos em três categorias: *Baseados em estatística*, *Baseados em qualidade* e *híbridos*. Os principais algoritmos de cada categoria são apresentados no decorrer deste capítulo.

3.1 Algoritmos baseados em estatística

Os algoritmos baseados em estatística utilizam informações de perda de quadros para decidir qual esquema de modulação será usado. Em geral, há duas causas possíveis para a perda de quadros: *colisões* e *condições desfavoráveis do canal* (baixo SNR).

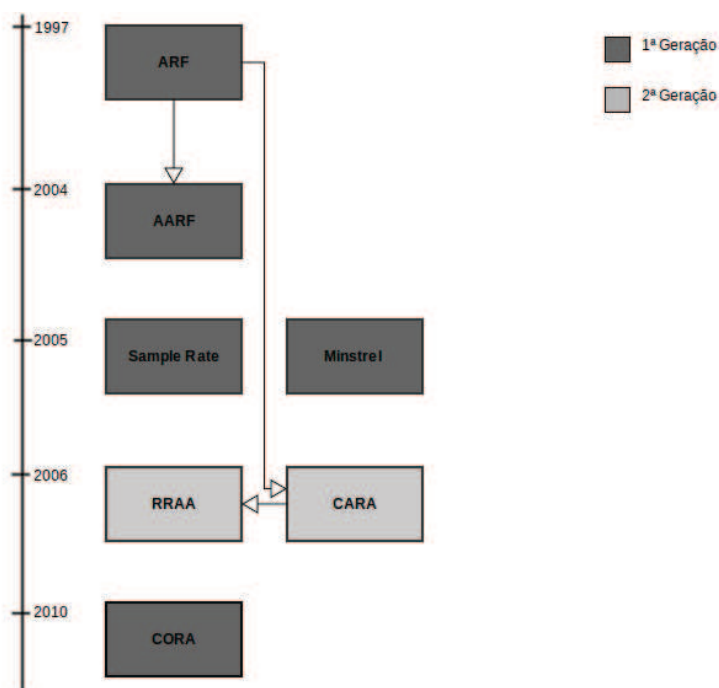


Figura 3.1: Algoritmos baseados em estatística

A primeira geração dos algoritmos baseados em estatística não diferencia as causas, pelas quais, houve perda de quadros (BIAZ; WU, 2008b). Por outro lado, a segunda geração introduz soluções para esta limitação. Na Figura 3.1 são apresentados os principais algoritmos em ordem cronológica. A relação entre eles também é mostrada, ou seja, se o algoritmo é uma extensão ou uma referência de outro. Pode ser visualizado também, quais algoritmos são de primeira ou segunda geração.

A seguir, são descritos detalhadamente os principais algoritmos baseados em estatística.

3.1.1 Auto Rate Fallback

O primeiro algoritmo de controle de taxa de transmissão em redes 802.11 foi o ARF, proposto inicialmente para o padrão WaveLanII (KAMERMAN; MONTEBAN, 1997).

O ARF considera a taxa de 2 Mbps como padrão. A contagem de ACKs permite o algoritmo aumentar ou diminuir a taxa, através de limitantes fixos. Há um temporizador que é iniciado para acompanhar a entrega de ACKs. Quando duas transmissões são enviadas sem sucesso, ou seja, dois ACKs são perdidos, o algoritmo muda para a taxa imediatamente menor. Quando este temporizador expira ou é atingido o número de dez ACKs recebidos com sucesso, o dispositivo muda para a taxa imediatamente superior.

O ARF é um algoritmo simples e intuitivo (BIAZ; WU, 2008b). Mas possui algumas limitações, como:

- Mesmo estando na taxa ótima o ARF pode mudar desnecessariamente para uma taxa acima;
- Ao optar por limitantes fixos, que aumentam ou diminuem a taxa, o algoritmo não é capaz de adaptar eficientemente a uma taxa quando as condições

encontradas no ambiente variam com grande frequência. Isto acontece porque a diferença em relação aos limitantes para aumentar a taxa (10 transmissões enviadas com sucesso) e para diminuir a taxa (2 transmissões enviadas sem sucesso) é grande. Dessa forma, nunca haverá uma sincronia quando a condição do canal varia frequentemente (LACAGE; MANSHAEI; TURLETTI, 2004).

Mesmo sendo um algoritmo com bastante limitações, o ARF é bastante utilizado em diversos dispositivos (SADEGHI *et al.*, 2002).

3.1.2 Adaptive Auto Rate Fallback

O AARF é uma extensão do ARF que melhora uma de suas limitações, a sincronia dos limitantes fixos, descrito na seção anterior (LACAGE; MANSHAEI; TURLETTI, 2004).

O AARF implementa a mesma lógica que o ARF, mas com um diferencial: a variação nos limitantes fixos. A ideia básica do AARF é alterar os limitantes continuamente em tempo de execução. Dessa forma, o algoritmo adapta estes limitantes utilizando um ajuste exponencial.

Os valores padrões para os limitantes fixos são os mesmos usados no ARF. A taxa será aumentada a cada dez transmissões enviadas com sucesso e reduzida a cada duas transmissões enviadas sem sucesso. O limitante é redefinido para o seu valor padrão quando a taxa é reduzida em função de duas transmissões enviadas sem sucesso. Por outro lado, o limitante que incrementa será modificado na seguinte situação: quando houver uma transmissão mal-sucedida, após uma mudança da taxa ou após uma transmissão bem-sucedida, o limitante de incremento será multiplicado por dois.

O seguinte exemplo, ilustrado na Figura 3.2, descreve o algoritmo. Supondo que exista as taxas nominais de 2Mbps, 5Mbps e 11 Mbps. A taxa de 2Mbps está sendo usada. Assumindo valores padrões para os limitantes fixos e

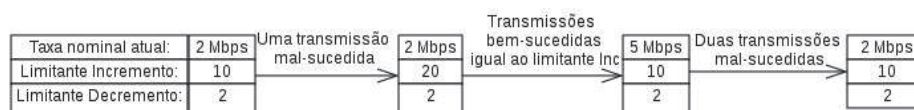


Figura 3.2: Exemplo do funcionamento do AARF

que, em um determinado momento ocorra a primeira transmissão mal-sucedida. Então, o limitante de incremento será multiplicado por dois, ou seja, para conseguir aumentar a taxa será preciso de vinte transmissões enviadas com sucesso. Logo após esta situação, são realizadas vinte transmissões com sucesso e a taxa é aumentada para 5Mbps. Dessa forma, o limitante que incrementa é redefinido para o seu valor padrão. Depois, duas transmissões mal-sucedidas ocorrem. Assim, a taxa volta a ser de 2Mbps.

3.1.3 SampleRate

O SampleRate é um dos mais importantes algoritmos presentes na literatura e é o padrão do driver MadWifi (MORRIS; BICKET; BICKET, 2005)¹. O MadWiFi é um driver do kernel do linux para dispositivos de rede local sem fio.

O objetivo do SampleRate é maximizar a vazão através da escolha de uma taxa que consiga transmitir dados com sucesso no menor atraso. O cálculo para descobrir uma taxa melhor é feito a cada dez segundos. Este é um aspecto fundamental do SampleRate, a maneira como ele envia periodicamente quadros em outras taxas para estimar o tempo médio de transmissão. Para realizar este cálculo o algoritmo transmite um quadro nas demais taxas.

O SampleRate reduz o número de taxas elegíveis eliminando aquelas que nunca vão transmitir dados com desempenho melhor do que a atual. Isto ocorre quando a taxa candidata não consegue ultrapassar a vazão atual de dados.

O algoritmo também pára de enviar o quadro de verificação para uma taxa que tem muitas transmissões mal-sucedidas. Quando o SampleRate encontra uma

¹Informações sobre o MadWiFi estão disponíveis no endereço <http://madwifi-project.org>

taxa com desempenho melhor, ele começa a usá-la. A cada 4 transmissões de quadros enviadas sem sucesso o algoritmo passa a diminuir a taxa até encontrar uma taxa que é capaz de enviar dados.

3.1.4 Minstrel

O Minstrel segue o mesmo raciocínio que o SampleRate, enviando periodicamente quadros em outras taxas (FIETKAU, 2005). Além disso, é uma abordagem disponível no kernel do Linux e no MadWifi.

O funcionamento do Minstrel é simples, podendo se resumir da seguinte forma:

- 10% dos quadros são enviados em taxas escolhidas aleatoriamente. A partir deste envio é criada uma tabela com informações da taxa com a melhor vazão, a taxa com a segunda melhor vazão e uma terceira taxa que é a mais robusta.
- 90% dos quadros são enviados de acordo com esta tabela. Cada retransmissão de um determinado quadro utiliza a ordem apresentada por esta tabela. Se as três taxas falharem é escolhida a menor taxa nominal disponível.

3.1.5 Collision-Aware Rate Adaptation

O CARA é baseado no ARF, uma vez que ele implementa a estratégia de aumentar e diminuir a taxa a partir de limitantes fixos (KIM *et al.*, 2006). O CARA utiliza duas estratégias que permitem distinguir o motivo da perda de quadros: mensagens RTS/CTS e a detecção do CCA (*Clear Channel Assessment*). As mensagens RTS/CTS são obrigatórias no CARA. Por outro lado, a detecção do CCA é opcional, servindo de suporte ao RTS/CTS.

Os limitantes usados pelo CARA são os mesmos usados pelo ARF. A cada dez transmissões enviadas com sucesso a taxa aumenta e a cada duas transmissões enviadas sem sucesso a taxa diminui.

Como a mensagem RTS tem um tamanho pequeno e é transmitido a uma taxa de transmissão robusta a probabilidade de sua transmissão ser mal-sucedida é muito pequena. Dessa forma, qualquer transmissão do RTS enviada sem sucesso é porque ocorreu uma colisão. Mas, quando o RTS é transmitido com sucesso e a transmissão dos dados falha, pode-se deduzir que a perda dos quadros é devido a uma condição do canal ruim. Isto é concluído pois, quando a transmissão do RTS é bem sucedida já havia sido reservado o canal, garantindo que nenhuma colisão ocorreria com a transmissão seguinte.

Após um intervalo de SIFS, um quadro ACK é aguardado. O CCA fica escutando o canal para ver se ele recebe o reconhecimento do ACK indicando que a transmissão foi bem sucedida. Caso o CCA perceba que há uma transmissão em andamento enquanto ele espera o recebimento do ACK, é reportado que houve uma colisão.

Uma das limitações dessa abordagem é o uso de mensagens RTS/CTS que sobrecarregam a rede, reduzindo a vazão. O CARA também não considera o cenário onde um quadro RTS pode ser perdido devido ao nó receptor estar fora do alcance (BIAZ; WU, 2008a).

3.1.6 Robust Rate Adaptation Algorithm

O algoritmo RRAA estima a condição do canal afim de maximizar a vazão através do cálculo do FER (*Frame Error Rate*), que é calculado através da razão entre número de quadros ACKs recebidos e o número de quadros de dados transmitidos durante um intervalo de tempo curto. Além disso, é utilizado um tipo de RTS adaptado para reduzir as colisões (WONG *et al.*, 2006).

O algoritmo é iniciado na taxa mais elevada. A taxa atual é associada a três parâmetros: o tamanho janela de tempo, um limitante de decremento e um limitante de incremento. Quando a janela expira, o FER é calculado. A taxa é reduzida para uma imediatamente menor, caso o FER seja maior que o limitante

de incremento ou é aumentada para uma imediatamente maior, caso o FER seja menor que o limitante de decremento. Se o FER estiver entre os dois limitantes, a taxa não é alterada.

Além disso, o RRAA utiliza um quadro RTS adaptado para reduzir colisões. Um contador é usado para informar quantos quadros serão precedidos do quadro RTS. Caso um quadro não seja enviado com sucesso e não tenha sido precedido por um quadro RTS, conclui-se que o motivo da perda do quadro foi uma colisão. Dessa forma, o contador é incrementado. Se um quadro perdido é precedido por um quadro RTS, ou ocorre um envio com sucesso sem ser precedido por um quadro RTS, o valor do contador é dividido por dois.

Os resultados apresentados pelos autores do algoritmo mostram que o RRAA não envia quadros de sondagem nas demais taxas para escolher taxas futuras. Este algoritmo converge rapidamente quando há mudanças no cenário da rede.

3.2 Algoritmos baseados em qualidade

Os algoritmos abordados nesta seção utilizam uma estratégia para medir a qualidade do canal para escolher a taxa de transmissão. Em geral, esta estratégia é feita através do cálculo da relação entre sinal e ruído. Na Figura 3.3, são mostrados, em ordem cronológica, estes algoritmos.

Existem duas categorias de algoritmos baseados em qualidade: *malha fechada* e *malha aberta*. Os algoritmos de malha fechada escolhem a taxa pela estação emissora, mas, quem indica a qualidade do sinal é a estação receptora. A estação emissora recebe a informação da qualidade do sinal através do RTS/CTS. Já os algoritmos de malha aberta, inferem a qualidade do sinal no emissor, e assim, com esta informação, é escolhido uma taxa no receptor (CARDOSO, 2009). Nas seções a seguir, são detalhados os principais algoritmos.

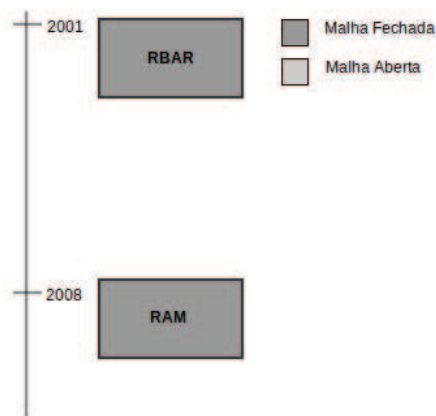


Figura 3.3: Algoritmos baseados em qualidade

3.2.1 Receiver-Based Auto Rate

O RBAR é um algoritmo de malha fechada, ou seja, ele permite que o receptor escolha a taxa apropriada para os dados durante o envio dos quadros RTS/CTS (HOLLAND; VAIDYA; BAHL, 2001).

A implementação deste algoritmo altera o formato dos quadros RTS e CTS, pois em vez de serem enviadas as informações de duração de mensagem para reservar o canal, será carregada a informação de qual taxa será usada e o tamanho dos dados enviado na transmissão. Esta alteração serve para dois propósitos: prover um mecanismo pelo qual o receptor pode comunicar ao emissor a troca da taxa e fornecer às estações vizinhas informações suficientes para calcular a duração do tempo para a transmissão dos dados.

O quadro RTS é enviado para a estação receptora calcular o SNR. Assim, o receptor seleciona a taxa que pode ser suportada pelas condições do canal de acordo com o cálculo da razão entre sinal e ruído. Dessa forma, a estação emissora consegue estimar as condições do canal quando receber o quadro de volta com as informações extraídas na estação receptora.

3.2.2 Rate Adaptation in Mobile Environments

O RAM é um algoritmo que adota uma estratégia de malha fechada, uma vez que o receptor seleciona a taxa para o próximo quadro transmitido (CHEN; GANGWAL; QIAO, 2009). Além disso, o algoritmo foi implementado no driver Madwifi.

O algoritmo é dividido em quatro fases: predição do SNR, escolha da taxa baseado na predição do SNR, *feedback* da taxa de seleção para o emissor e atualização da taxa. As três primeiras fases são realizadas no receptor e a última no emissor.

A fase de predição estima o SNR de forma conservadora, lidando com a alta instabilidade da razão SNR. Para selecionar a taxa, o algoritmo mantém uma tabela *vazão X (taxa, SNR)*. Para cada par (*taxa, SNR*) existe uma vazão relacionada. Com base no valor do SNR, o receptor verifica na tabela a taxa com o melhor desempenho e a seleciona.

Para transmitir as informações de volta para o transmissor é apresentado uma nova abordagem. O algoritmo varia a taxa de transmissão do ACK de uma maneira controlada para indicar a decisão de seleção da taxa de transmissão do próximo quadro. Dessa forma, existe um conjunto de taxas que são eleitas para ser o valor da taxa do ACK a ser transmitido. O ACK será transmitido na taxa mais elevada do conjunto de taxas que é menor ou igual a taxa de transmissão do quadro que está sendo reconhecido.

Após estas fases, é enviado para o transmissor a nova taxa escolhida pelo receptor. A próxima taxa será definida de acordo com o resultado da última transmissão. Se a transmissão atual tiver falhado, a taxa escolhida no receptor será a nova taxa. Se a transmissão atual for bem sucedida, o emissor tem duas opções: Se a taxa atual é maior que 2 Mbps, o emissor depende do feedback do receptor para definir a taxa. Se a taxa atual é igual a 1 ou 2 Mbps o emissor usa uma heurística para definir a taxa.

A tabela que contém informações sobre a vazão X (*taxa, SNR*) funciona bem quando a interferência é fraca ou estável. No entanto, não tem bom desempenho na presença de alto grau na rede.

3.3 Algoritmos Híbridos

Os algoritmos híbridos são aqueles que tentam mesclar as melhores características dos algoritmos baseados em qualidade e dos baseados em histórico. A Figura 3.4 ilustra os principais algoritmos desta classe em ordem cronológica.

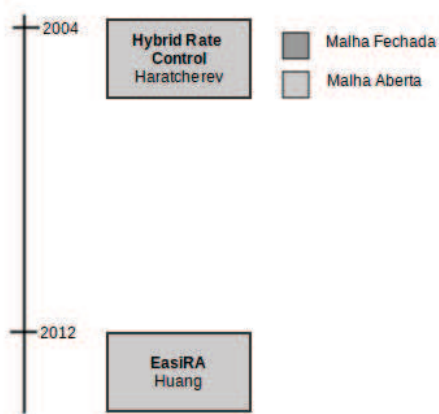


Figura 3.4: Algoritmos Híbridos

3.3.1 Hybrid Rate Control

O Hybrid Rate Control é um algoritmo que não faz transmissões de quadros RTS e CTS, sendo considerado um algoritmo de malha aberta (HARATCHEREV *et al.*, 2004). O objetivo principal do algoritmo é reduzir o atraso e o *jitter* na entrega dos dados.

Para se obter informações sobre as condições do canal, quadros de sondagem são enviados nas taxas adjacentes. O algoritmo mantém informações sobre

as condições do canal baseando-se em estatísticas feitas sobre as transmissões. As informações mantidas são: o FER e a vazão.

A abordagem para o cálculo da vazão é a seguinte: dez por cento dos dados são enviados nas duas taxas adjacentes à atual. No final destas transmissões, o desempenho de cada uma das três taxas é determinado dividindo-se a quantidade de dados transmitidos pelos seus tempos de transmissão acumulados. Por fim, a taxa que proporcionou a maior vazão é escolhida.

Já a abordagem para selecionar uma taxa, considerando o FER, é da seguinte forma: se o FER ultrapassar a taxa de transmissão será decrementada. Se o FER estiver perto de zero, quadros de sondagem são enviados na taxa adjacente maior. Se todos estes quadros forem reconhecidos a taxa é mudada para a que foi sondada. Para evitar que o algoritmo oscile entre duas taxas adjacentes, a ação que incrementa a taxa pode ser suspensa durante algum tempo depois de uma ação que decrementa a taxa.

Alem disso, são enviados quadros de sondagem ACKs para deduzir a qualidade do canal no receptor. Dessa forma, as taxas que não podem melhorar o desempenho são eliminadas.

3.3.2 Hybrid Rate Adaptation Scheme - EasiRA

O EasiRA é um esquema que utiliza três estratégias para escolher a taxa de transmissão: estima a qualidade do canal, utiliza uma estratégia para distinguir o motivo da perda de quadros e combina métodos de seleção aleatório e determinístico para escolher a taxa de transmissão (HUANG *et al.*, 2012). A Figura 3.5 representa a arquitetura do EasiRA.

O EasiRA consiste de dois componentes: um sensor que estima a condição do canal (*LSS - Link Status Sensor*) e um controlador para selecionar a taxa (*RSC - Rate Selection Controller*). O LSS não coleta só informações sobre as condições do canal, mas, também informações sobre o número de quadros enviados



Figura 3.5: Arquitetura do EasiRA

com sucesso, a posição e a velocidade do dispositivo. Dessa forma, o EasiRA converge rapidamente diante de mudanças na condição do canal. Além disso, o LSS diferencia o motivo das perdas de quadros. Para isso, o emissor e o receptor precisam trabalhar cooperativamente. O receptor deve informar ao emissor as mudanças do valor do ESS (*Environmental Signal Strength*) imediatamente. O ESS mede a força do sinal para ajudar a diferenciar o motivo das perdas de dados.

O EasiRA utiliza dois limitantes, que podem ser ajustados, para ajudar na escolha da taxa: α e β . Quando o número de transmissões enviadas sucessivamente com sucesso é maior do que α , o emissor escolhe uma taxa imediatamente superior. Se o número de transmissões enviadas sem sucesso sucessivamente é maior do que β , o emissor seleciona uma taxa imediatamente inferior.

Com base nas informações fornecidas pelo LSS, o RSC decide quando deve aumentar ou diminuir a taxa. Assim, o RSC combina mecanismos de adaptação de taxa aleatórios e determinísticos.

3.4 Considerações finais

Neste capítulo foi apresentado um levantamento dos principais algoritmos de controle da taxa de transmissão abordando conceitos, estratégias usadas, limitações, etc. A tabela 3.1 apresenta um quadro comparativo das principais características dos algoritmos de controle da taxa.

| Algoritmo | Classificação | Métrica utilizada | Diferenciação de perdas | Intervalo | Localização |
|---------------------|---------------|-------------------|-------------------------|-----------------------------|-------------|
| ARF | Estatística | vazão | Não | 2 a 10 quadros | emissor |
| AARF | Estatística | vazão | Não | 2 a (10 * 2x) quadros | transmissor |
| CARA | Estatística | vazão | Sim. RTS/CTS e CCA | 2 a 10 quadros | transmissor |
| CORA | Estatística | configurável | | Configurável (0.1 segundos) | transmissor |
| EasiRA | Híbrido | vazão | Sim. LSS | 1 quadro | ambos |
| Hybrid Rate Control | Híbrido | atraso e jitter | Não | 1 quadro | transmissor |
| RAM | Qualidade | vazão | Não | 1 quadro | receptor |
| RBAR | Qualidade | vazão | Não | 1 quadro | receptor |
| RRAA | Estatística | vazão | Sim. Adaptive-RTS | 5 a 40 quadros | transmissor |
| Sample Rate | Estatística | vazão | Não | 10 segundos | transmissor |

Tabela 3.1: Tabela Comparativa

A coluna *classificação* faz referência às classes de algoritmos existentes. A maioria dos algoritmos são baseados em estatística. Estes algoritmos são simples e possuem tempo de convergência alto. As métricas mais utilizadas são: a vazão, o *jitter* e o atraso. Pode-se fazer a *diferenciação de perdas* por dois motivos: colisão e condição desfavoráveis do canal. Isto pode ser feita através de quadros RTS/CTS adaptados, como é implementado no CARA e o RRAA, ou através de um componente, como é caso do EasiRA. No caso do *intervalo* entre a troca da taxa, as abordagens baseadas em estatística precisam de vários quadros para estimar as condições do canal. Ao contrario das abordagens baseada em qualidade que precisam de apenas um quadro. Em relação à *localização*, pode-se perceber que todos os algoritmos baseados em estatística usam informações do emissor para ajustar a taxa. Já os baseados em qualidade usam informações do receptor. As características do algoritmo CORA serão detalhadas no Capítulo 5.

4 METODOLOGIA

Este trabalho é por natureza uma pesquisa aplicada pois há uma finalidade de aplicação prática, pois tem o objetivo de buscar conhecimento e solucionar problemas. Quanto aos objetivos, é uma pesquisa exploratória com intenção de descobrir. É utilizada uma abordagem quantitativa, uma vez que serão usadas métricas para avaliar o desempenho dos algoritmos.

Os principais procedimentos usados são: *survey* (será apresentado uma revisão bibliográfica das principais soluções que estão na literatura) e estudo de casos múltiplos. O método para coleta de dados baseia-se em experimentação pois serão feitas simulações para avaliar o desempenho dos algoritmos.

Foi realizada uma avaliação de desempenho dos algoritmos. O objetivo desta avaliação foi, através de uma abordagem quantitativa, identificar qual algoritmo apresenta o melhor desempenho em diferentes cenários de rede. Para configuração dos diferentes cenários, foram considerados os seguintes parâmetros: número de nós, grau de mobilidade, densidade e carga de tráfego. O desempenho foi avaliado de acordo com as seguintes métricas: vazão média agregada e taxa de perda de pacotes. Os algoritmos tiveram seus desempenhos avaliados por meio de simulações realizadas no *Network Simulator*¹, que é uma das ferramentas de simulação mais utilizadas pelos pesquisadores na área de redes e sistemas distribuídos. O NS-3 é baseado em eventos discretos. Além disso, este simulador é de código fonte aberto.

Para fazer as simulações foi usado o padrão IEEE 802.11a, que tem oito modulações disponíveis. O 802.11a foi utilizado pois o algoritmo RRAA não apresentava suporte a outros padrões. As métricas utilizadas para medir o desempenho dos cenários são: vazão agregada e a taxa de perda de pacotes. A vazão agregada é calculada de acordo com a equação:

¹Mais informações sobre o simulador podem ser encontradas na página “web” no endereço <http://www.nsnam.org>

$$\sum_{j=1}^f \frac{rxBytes_j}{\Delta t} \quad (4.1)$$

em que f é a quantidade de fluxos, ΔT é o tempo total em que o fluxo envia dados durante o intervalo $[t_{inicial}, t_{final}]$ e $rxBytes$ é a quantidade de bytes efetivamente transmitidos pelo fluxo j .

A taxa de perda de pacotes é calculada através da fórmula:

$$1 - \frac{rxPackets}{txPackets} \quad (4.2)$$

em que $rxPackets$ é o número de pacotes recebidos, $txPackets$ é a quantidade de pacotes transmitidos.

O protocolo de transporte usado é o UDP. O intervalo de confiança utilizado é de 95%. A fórmula utilizada para calcular o intervalo de confiança é:

$$\left(\bar{x} - 1.96 \times \frac{\sigma}{\sqrt{n}}, \bar{x} + 1.96 \times \frac{\sigma}{\sqrt{n}} \right) \quad (4.3)$$

em que \bar{x} é a média, σ é o desvio padrão e n é o total de amostras.

Como somente a análise da vazão agregada média não é satisfatória em cenários que possuem vários nós, foi utilizado o cálculo da justiça (JAIN; CHIU; HAWA, 1984). O objetivo deste cálculo é medir o nível de justiça quanto ao compartilhamento da rede entre as estações. Dessa forma, mostrando se os recursos da rede estão sendo distribuídos entre os nós de forma justa. O nível é medido de 0 a 1. Quanto mais próximo de 1 o nível de justiça é maior, ou seja, a distribuição de recursos na rede esta sendo feita de forma equalitária. Para calcular a justiça é utilizado a seguinte fórmula:

$$J_f = \frac{(\sum_{f=1}^n v_f)^2}{n \times \sum_{f=1}^n v_f^2} \quad (4.4)$$

em que n é o número total de fluxos, v_f é a vazão média do fluxo f .

Os algoritmos analisados são o ARF, o RRAA, o CARA, o Minstrel, o RBAR e o CORA. O RBAR está implementado no simulador em uma versão básica a qual é chamado Ideal.

Assim, esta atividade consistiu na realização das seguintes tarefas:

1. Estudar o simulador;
2. Implementar um algoritmo que não está implementado no simulador;
3. Realizar simulações para avaliar o desempenho destes algoritmos em diferentes cenários;
4. Realizar análise dos resultados.

5 IMPLEMENTAÇÃO DO CORA

O algoritmo escolhido para ser implementado no simulador foi o CORA. Este algoritmo foi escolhido pois não está implementado no NS-3 e por que apresenta uma estratégia diferente dos outros algoritmos. Neste capítulo, será descrito o algoritmo e como foi realizada esta implementação, além das dificuldades encontradas.

5.1 COgnitive Rate Adaptation

O CORA é um algoritmo de primeira geração uma vez que não identifica o motivo de perdas de quadros (CHAVES *et al.*, 2009). Este algoritmo foi desenvolvido utilizando o mecanismo CogProt (MALHEIROS *et al.*, 2010), que é uma abordagem genérica para auto-configuração de protocolos de rede. Neste mecanismo, considera-se os conceitos de redes cognitivas, que podem ser ajustadas dinamicamente com o objetivo de melhorar o desempenho de toda a rede. O *framework* cognitivo CogProt reconfigura parâmetros dos protocolos de comunicação baseando-se em histórico de desempenho do sistema. Dessa forma, faz com que as estações da rede possam ajustar dinamicamente a configuração destes protocolos afim de melhorar o desempenho da rede. O CogProt implementa o processo de auto-configuração através de um ciclo de controle por realimentação que monitora o desempenho e executa ações de reconfiguração em um parâmetro, afim de otimizar o desempenho do sistema. O ciclo de controle consiste em três fases: observação, orientação e decisão/ação.

O CORA implementa o ciclo de controle do CogProt. A Figura 5.1, apresenta o funcionamento do CORA. As três fases do ciclo foram renomeadas. A implementação do CORA funciona da seguinte forma:

Observação: As informações de desempenho de cada taxa são mantidas em uma base de conhecimento local. Nesta fase, o CORA mede o desempenho da

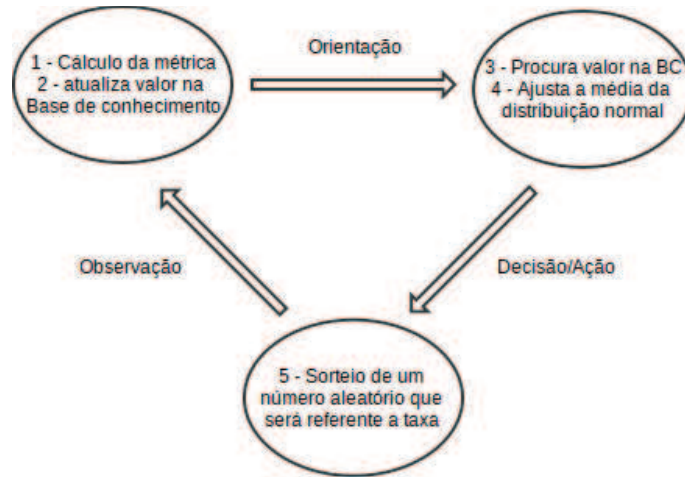


Figura 5.1: Ciclo de controle do CORA

taxa atual. Para isto é medido a vazão de dados. Após este cálculo é atualizado os dados na base de conhecimento através da utilização de uma média exponencial ponderada. Este cálculo é feito da seguinte forma:

$$BC_x = (1 - \alpha) * BC_x + \alpha * M_x \quad (5.1)$$

em que BC_x é o valor calculado através de uma métrica que representa uma taxa x , α é o peso da informação e M_x é o resultado do cálculo feito por uma métrica que representa uma taxa x .

Orientação: Nesta fase, é decidido qual a taxa com o melhor desempenho. O algoritmo analisa a base de conhecimento para escolher a taxa que fornece o melhor desempenho de acordo com alguma métrica de desempenho previamente definida. Em seguida, é feito um ajuste da média μ de uma distribuição normal $\mathcal{N}(\mu, \sigma^2)$. Sendo que, σ é um parâmetro configurável.

Decisão/Ação: Durante este passo, um valor aleatório y é escolhido dentro do intervalo de possíveis taxas elegíveis gerado de acordo com uma distribuição

normal. Após um valor y ser sorteado ele é arredondado para o próximo número. O arredondamento é feito da seguinte forma: $z = y + 0.5$, sendo z um número natural. O valor final de z é referente a uma das n taxas eligíveis. Por fim, a taxa referente a z é escolhida.

Este ciclo de realimentação é repetido por cada um dos elementos da rede em um intervalo I definido. O CORA é executado por cada elemento de rede de forma independente. Além disso, cada elemento tem a sua própria base de conhecimento.

Esta abordagem é descentralizada, pois se baseia apenas em informações disponíveis nos próprios elementos da rede sem necessidade de qualquer comunicação entre os nós.

No algoritmo 1 é apresentado um pseudocódigo do CORA. Na etapa de observação é realizado é calculado o valor da métrica utilizando uma média exponencial ponderada. Em seguida, é atualizado o valor na base de conhecimento. Na etapa de Orientação, é procurado o valor na base de conhecimento que representa a taxa com o melhor desempenho. Após isto, é ajustado a média da distribuição normal. Por fim, na etapa de Decisão, é sorteado um número aleatório referente a uma taxa de transmissão.

Entrada: σ valor referente ao desvio padrão;
 α \leftarrow peso utilizado no cálculo da média;
 BC \leftarrow vetor que representa a base de conhecimento;
 I \leftarrow intervalo entre as iterações

início

$T \leftarrow min$;

while true do

/ Etapa de observação*/*

$M_a \leftarrow M(p)$;

$BC[a] \leftarrow (1 - \alpha) * BC[a] + \alpha * M_a$;

$v_n \leftarrow BC[min]$;

/ Etapa de Orientação*/*

para cada T de min + 1 até max faça

if $BC[t] > V_n$ **then**

$i_n \leftarrow t$;

$V_n \leftarrow BC[t]$;

fim

$\mu \leftarrow i_n$;

/ Etapa de Decisão*/*

$n \leftarrow \lfloor \mathcal{N}(\mu, \sigma^2) + 0.5 \rfloor$;

$T \leftarrow T_n$;

 Sleep(I segundos);

end

fim

Algorithm 1: Cognitive Rate Adaptation

5.2 A Implementação

No anexo A, é apresentado o código fonte do CORA. Todos os parâmetros configuráveis (α , Intervalo e σ) são escolhidos sem qualquer mecanismo de escolha automático. Em outras palavras, a configuração dos parâmetros fica a cargo do usuário. A taxa nominal escolhida para o início de toda simulação é a taxa intermediária disponível. Por exemplo, se o padrão utilizado é o 802.11b a taxa inicial será a de 5.5Mbps. Por outro lado, se for usado o 802.11a ou 802.11g a taxa inicial será de 24Mbps. A classe que dá suporte a implementação dos algoritmos de controle de taxa no NS-3 é a *WifiRemoteStationManager*.

Como já foi dito na seção anterior, o CORA possui um ciclo com três etapas. Para cada etapa deste ciclo foi implementado um método. O método que controla o ciclo se chama *RunAlgorithm*. Este método chama os outros responsáveis por cada etapa do CORA e em seu final é chamado um método, chamado *Schedule*, que ficará responsável por chamar novamente o ciclo no intervalo *I* especificado.

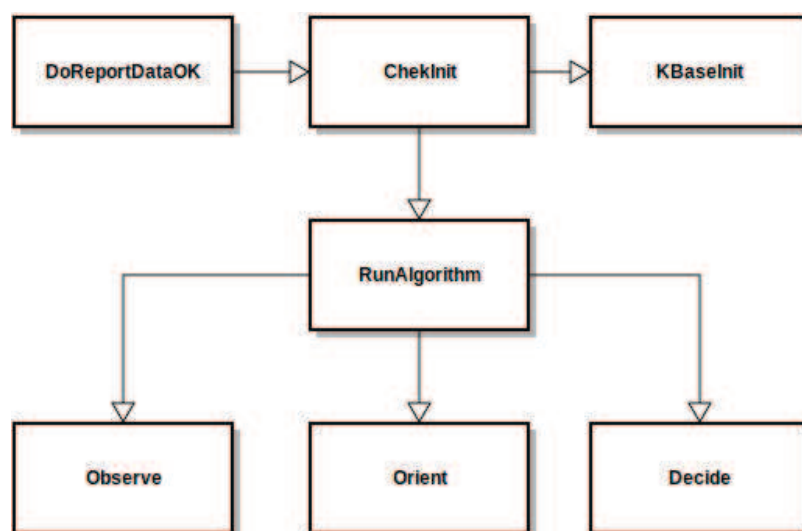


Figura 5.2: Diagrama da implementação do CORA

A Figura 5.2, representa a implementação do CORA no simulador apresentando a relação dos métodos que foram implementados. No método responsável pela observação é feito o cálculo da métrica escolhida e da média exponencial ponderada. Como já foi dito, o framework CogProt não define uma métrica a ser utilizada. A métrica escolhida na implementação é a vazão de dados. Para se calcular a vazão, duas informações são necessárias: a confirmação de um quadro recebido e o tamanho deste quadro. A confirmação do quadro é feita pelo método *DoReportDataOk* e o tamanho do pacote é informado pelo método *ReceivePacket*. O resultado da média exponencial ponderada é guardado na base de conhecimento. A etapa de Orientação, realizada pelo método *Orient*, procura na base de conhe-

cimento o dado com o maior valor. O método que realiza a etapa de decisão, chamado de *Decide*, escolhe uma taxa aleatoriamente, de acordo com a distribuição normal, cujo a média foi escolhida na etapa anterior. O resultado deste cálculo será um número de 1 a n. Este número representará a taxa de transmissão escolhida. Por exemplo, se o padrão 802.11a estiver sendo usado e o número sorteado for igual a 2, então, a taxa equivalente será a de 9Mbps.

6 AVALIAÇÃO DE DESEMPENHO

Neste capítulo, será apresentada a avaliação de desempenho dos algoritmos de controle de taxa. Para este estudo, foram utilizados dois cenários de rede envolvendo redes 802.11a infraestruturada.

Antes de realizar uma comparação do desempenho dos algoritmos, serão apresentados estudos de casos iniciais a partir dos quais foi estudado o comportamento das taxas nominais e do algoritmo CORA. Na seção 6.1, será apresentado o comportamento das taxas de transmissão nominais. Na seção 6.2, é apresentado o comportamento da abordagem CORA utilizando diferentes valores para os seus parâmetros de configuração. A seguir, na Seção 6.3, é analisado o comportamento dos algoritmos. Por fim, na Seção 6.4, é apresentada uma comparação de desempenho dos algoritmos, nos dois cenários de rede, através de tabelas e gráficos comparativos.

6.1 Comportamento das taxas de transmissão nominais

No primeiro estudo de caso proposto, foi analisado o comportamento dos algoritmos e das taxas nominais quando são utilizados em uma rede infraestruturada. Como é visto na Figura 6.1, a comunicação se dá através de uma estação móvel enviando dados para um ponto de acesso, que é mantido fixo. A estação está a 5m do AP e começa a mover imediatamente após o início da simulação, a uma velocidade constante, que pode ser de 1.5 m/s ou 10 m/s. Nos primeiros 50 segundos a estação se desloca 75 metros. Ao alcançar esta distância, a estação começa a se reaproximar do ponto de acesso, voltando a sua posição de origem. Cada simulação tem um tempo total de 100 segundos. A taxa de dados do fluxo é de 25 Mbps. Um resumo das configurações da simulação é visto na Tabela 6.1.

| Parâmetros | Valores |
|-------------------------|------------------------|
| Padrão | 802.11a |
| Tamanho de pacotes | 500, 1000 e 1500 bytes |
| Modelo de propagação | Nakagami |
| Velocidade da estação | 1.5 e 10 m/s |
| Tempo de simulação | 100 segundos |
| Taxa do fluxo | 25 Mbps |
| Protocolo de transporte | UDP |

Tabela 6.1: Configurações dos parâmetros de simulação.



Figura 6.1: Topologia composta de um AP e uma estação móvel.

Para esta topologia foram alterados os seguintes parâmetros: velocidade em que o nó se movimenta (1.5 m/s ou 10 m/s) e o tamanho do pacote enviado (500, 1000 ou 1460 bytes). Dessa forma, são considerados seis cenários diferentes.

Na Figura 6.2 é ilustrado o comportamento das oito taxas nominais para uma única iteração da simulação. Percebe-se que não existe um valor ótimo global para a taxa. A vazão máxima possível é alcançada por taxas diferentes durante todo o período de simulação. Dessa forma, fica evidente a necessidade de se implementar mecanismos que controlam a taxa com o objetivo maximizar a vazão de dados. A taxa de 9Mbps em nenhum instante da simulação consegue maximizar a vazão de dados. Estes comportamentos foram observados em todos os cenários.

A Figura 6.3, apresenta a vazão média alcançada pelas taxas de nominais. Pode-se perceber que em todos os cenários a taxa de 18Mbps tem um desempenho melhor. Por outro lado, as taxas de 6Mbps e 54Mbps apresentam os piores desempenhos. Analisando esta Figura em conjunto com a Figura 6.2, é observado que mesmo não alcançando bons resultados, as taxas nominais mais robustas

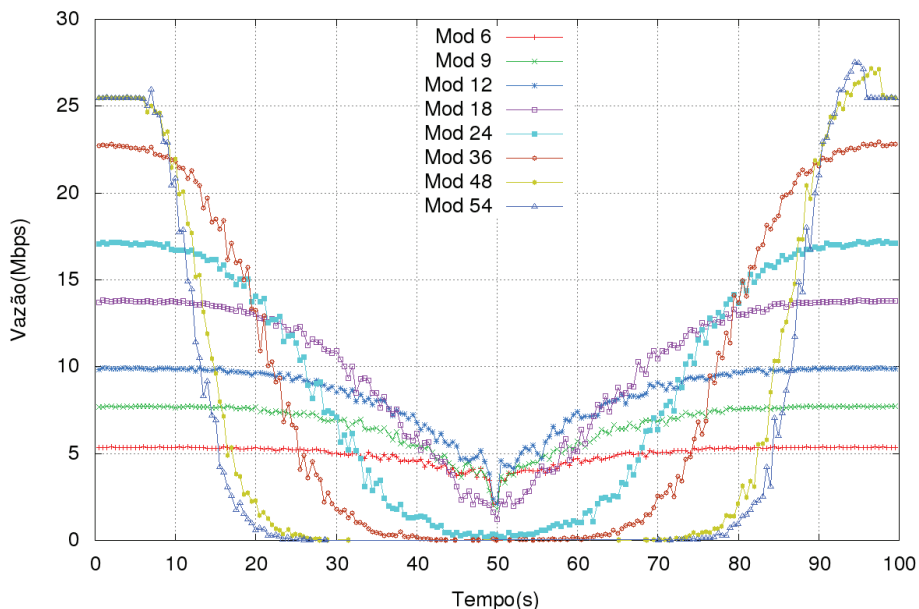


Figura 6.2: Vazão instantânea em função da qualidade do sinal ao longo do tempo para cada taxa de transmissão fixa em redes 802.11a configurados com os parâmetros velocidade - 1.5 m/s - e tamanho de pacotes - 1460 bytes.

(6Mbps e 9Mbps), não apresentam em nenhum momento da simulação uma vazão igual a zero. Por outro lado, em determinado momento da simulação, taxas menos robustas apresentam uma vazão igual a zero. Estes comportamentos são observados quando a estação móvel está mais distante do ponto de acesso. Dessa forma, pode-se concluir que taxas robustas conseguem um raio de cobertura maior.

Quando a estação começa a se distanciar do ponto de acesso, a vazão diminui gradativamente devido a baixa qualidade do sinal. Pode ser visto, para algumas taxas fixas, a partir de um determinado instante, a estação não é mais capaz de enviar dados para o ponto de acesso devido a baixa qualidade de sinal. Por exemplo, a taxa nominal de 54 Mbps, entre 20 a 70 segundos, aproximadamente, não é mais capaz de enviar dados. Assim, reforçando a ideia da utilização de um mecanismo de controle da taxa. Estes comportamentos também acontecem em todos os outros cenários.

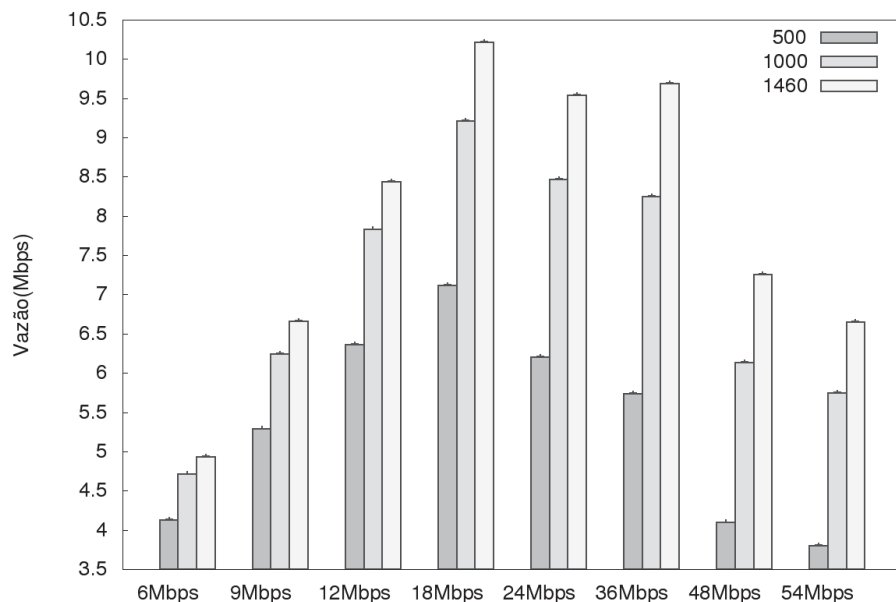


Figura 6.3: Vazão média alcançada pelas taxas de transmissão fixas em redes 802.11a infraestruturadas em cenários com velocidade igual a 1.5m/s.

6.2 Comportamento do CORA

Para avaliar o comportamento do CORA foram utilizados duas topologias: aquela proposta na seção anterior e outra que será apresentada a seguir. Para ambas topologias o CORA foi configurado com os seguintes valores: para o Intervalo I {0.05, 0.1, 0.3}, para o Alpha α {0.3, 0.5, 0.7, 0.9} e o Desvio Padrão σ {0.2, 0.3, 0.5, 0.8}.

6.2.1 Comportamento em uma Rede Infraestruturada

Foi utilizado a topologia apresentada na seção anterior e o cenário utilizado possui 1460 bytes de tamanho de pacotes e 1.5 m/s de velocidade da estação.

Foram realizadas simulações com todas as configurações definidas para os três parâmetros do CORA. A Figura 6.4, apresenta diferentes combinações para o α e o σ em função do I . Como pode ser observado, independente do valor do

intervalo, quando se aumenta o desvio padrão a vazão diminui. Assim, configurações com desvio padrão menor tem o desempenho melhor. Valores menores para o desvio padrão são mais conservadores, tornando-se frequente o sorteio da melhor taxa conhecida observada até o momento ou taxas adjacentes a ela. Isto poderia ser uma desvantagem mas é compensado pelo valor alto de α que faz com que o algoritmos convirja rapidamente. Dessa forma, é possível identificar que valores maiores para o α também melhoram o desempenho. Quando o valor do intervalo é menor o desempenho melhora, pois o número de taxas sorteadas em um determinado instante de tempo será maior que se o intervalo tivesse um valor alto. Conclui-se que, em topologias em que alterações na qualidade do sinal acontecem com frequência é mais interessante ter informações atualizadas em intervalos mais curtos pois há mudanças com frequência na qualidade do canal.

Como é visto na Figura 6.5, a vazão é apresentada em função de α para diferentes combinações de I e σ . Fica mais evidente nestes gráficos, que em configurações em que o α é maior o desempenho é melhor. Além disso, é realçado que quanto menor o σ os resultados alcançados são melhores. Mas quando há uma combinação igual a $\alpha = 0.3$, $I = 0.3$ e $\sigma = 0.2$ o desempenho é baixo, contrariando a ideia de que um σ baixo melhora o desempenho. Isto acontece por que os valores de α e do I não compensam o baixo valor do σ .

Na Figura 6.6, são realizadas combinações entre α e I em função do σ . Como pode ser visto, de forma geral, estes gráficos reforçam a ideia de que valores baixos para o σ produzem resultados melhores.

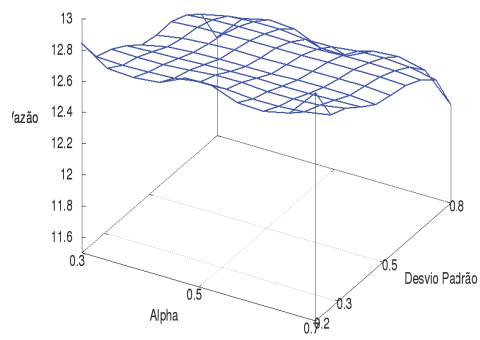
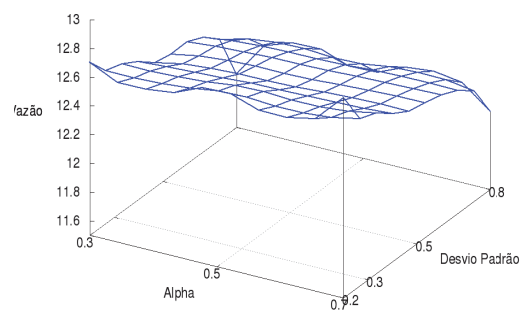
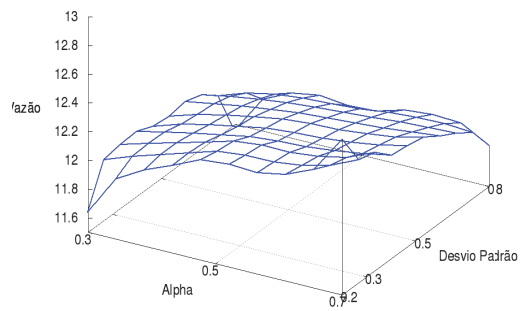
(a) $I = 0.05$ (b) $I = 0.1$ (c) $I = 0.3$

Figura 6.4: Vazão agregada média em função do intervalo para diferentes combinações de α e do desvio padrão em uma rede 802.11a infraestruturada.

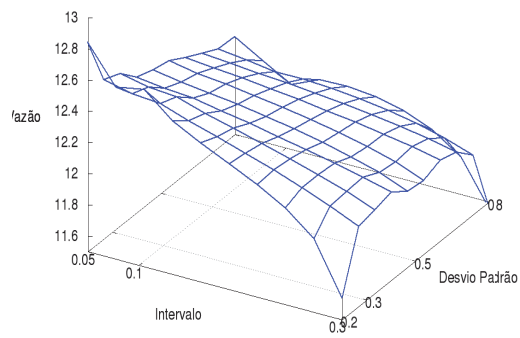
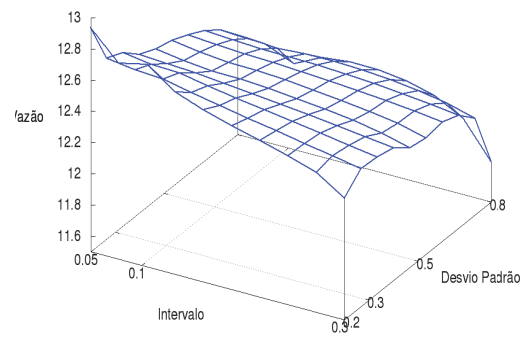
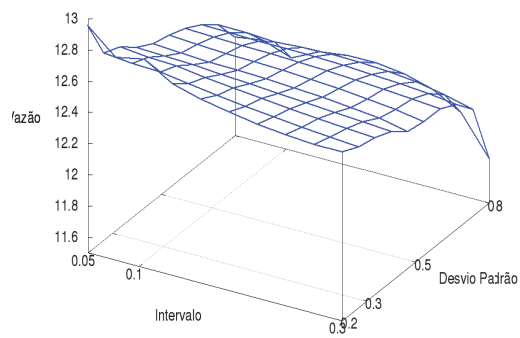
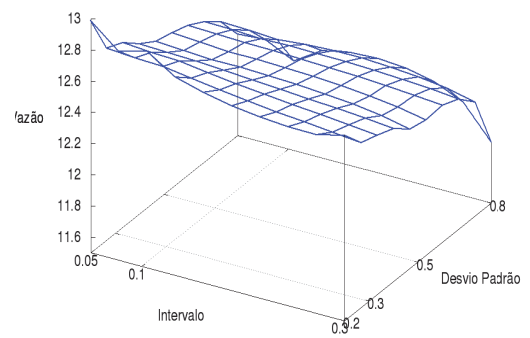
(a) $\alpha = 0.3$ (b) $\alpha = 0.5$ (c) $\alpha = 0.7$ (d) $\alpha = 0.9$

Figura 6.5: Vazão agregada média em função do α para diferentes combinações de intervalo e do desvio padrão em uma rede 802.11a infraestruturada.

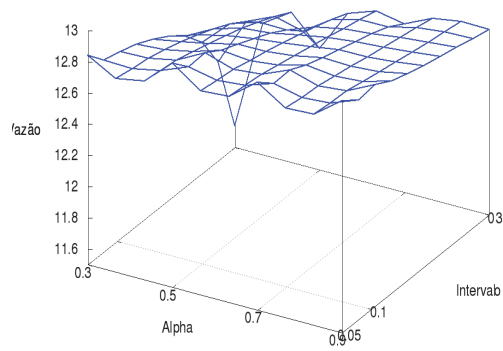
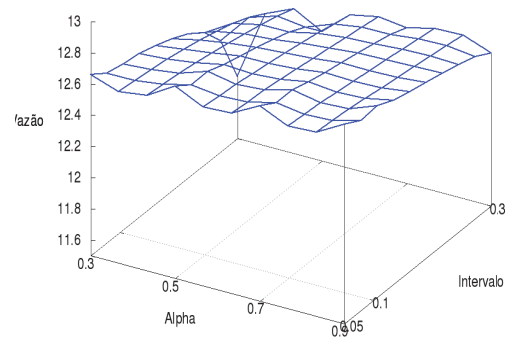
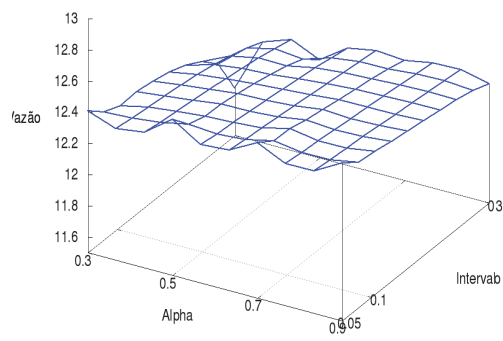
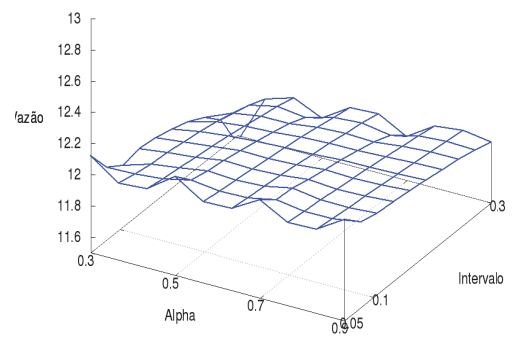
(a) $\sigma = 0.2$ (b) $\sigma = 0.3$ (c) $\sigma = 0.5$ (d) $\sigma = 0.8$

Figura 6.6: Vazão agregada média em função do desvio padrão para diferentes combinações de intervalo e de α em uma rede 802.11a infraestruturada.

A partir destas análises a configuração com a melhor vazão agregada média resultante foi: $I = 0.05$, $\alpha = 0.9$, $\sigma = 0.2$. Em topologias em que há mudanças significativas na condição do canal, configurar o CORA desta forma reflete em:

- agressividade menor ($\sigma = 0.2$). Dessa forma, diminui-se o tempo de convergência, sendo mais conservador, e aumenta a possibilidade de escolher taxas adjacentes àquela que é melhor conhecida no momento (usada como média da distribuição normal).
- importância de resultados recentes ($\alpha = 0.9$). Assim, o algoritmo reage rapidamente quando há mudanças na condição do canal.
- intervalo pequeno ($I = 0.05$). Várias vezes o ciclo é chamado, assim, o algoritmo escolhe mais taxas aleatórias em um determinado instante de tempo reagindo rapidamente a mudanças do canal.

De forma geral, para este tipo de topologia torna-se interessante que o algoritmo reaja rápido, pois, como já foi dito, no decorrer da simulação há mudanças significativas na condição do canal.

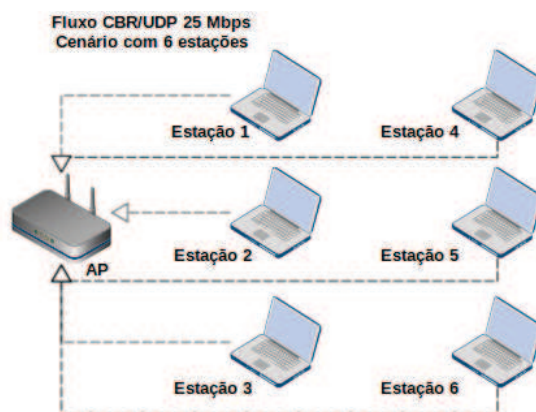
6.2.2 Comportamento com múltiplos nós

A segunda topologia é apresentada na Figura 6.7, a comunicação é feita através de múltiplas estações enviando dados a um AP. Todos os nós desta topologia são mantidos fixos. As estações estão dispostas a uma distância de 5 metros entre elas. O tamanho dos pacotes é de 1460 bytes.

Para esta topologia é alterado o parâmetro quantidade de estações, que pode ser: 6, 12 ou 18. Para estudar o comportamento do CORA foi utilizado um cenário com 12 estações. Os demais cenários possuem resultados com mínimas diferenças. Um resumo dos parâmetros de configuração da simulação são apresentados na Tabela 6.2.

Tabela 6.2: Configurações da topologia contendo múltiplos nós comunicando à um AP

| Parâmetros | Valores |
|-------------------------|----------------|
| Padrão | 802.11a |
| Tamanho de pacotes | 1460 bytes |
| Modelo de propagação | Nakagami |
| Velocidade das estações | 0 m/s |
| tempo de simulação | 100 segundos |
| Taxa de fluxo | 25 Mbps |
| Protocolo de transporte | UDP |
| Quantidade de estações | 6, 12 e 18 nós |

**Figura 6.7:** Topologia com múltiplas estações e um AP representada por um cenário contendo 6 estações

A Figura 6.8, mostra diferentes combinações para o α e o σ em função do I . Pode-se observar que os melhores resultados são vistos quando o intervalo é maior. Como não há uma degradação na qualidade do canal significativa não há a necessidade do ciclo ser chamado diversas vezes em um mesmo período de tempo. Quando o intervalo é menor, pode haver informações não confiáveis pois o ciclo seria chamado mais vezes em um mesmo período de tempo e uma pequena mudança na qualidade do sinal tornaria significativa.

Na Figura 6.9, é apresentado diferentes combinações de σ e de I em função de α . Quanto menor o α maior é a vazão média. Isto acontece pois não há grandes mudanças na condição do canal, então, o algoritmo não necessita de reajir

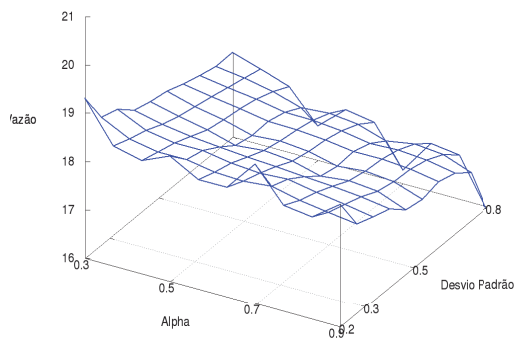
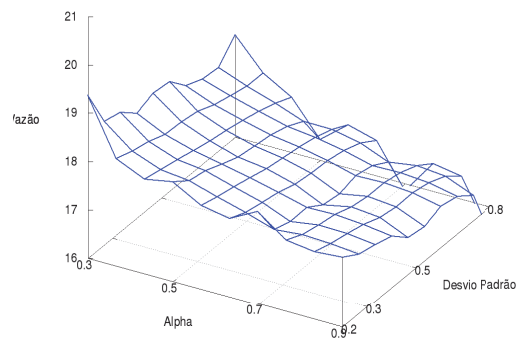
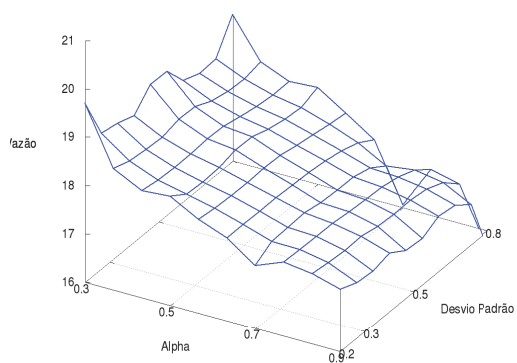
(a) $I = 0.05$ (b) $I = 0.1$ (c) $I = 0.3$

Figura 6.8: Vazão agregada média em função do intervalo para diferentes combinações de alpha e do desvio padrão em uma rede 802.11a infraestruturada.

rapidamente a elas. Dessa forma, valores baixos para α resultam em um compor-

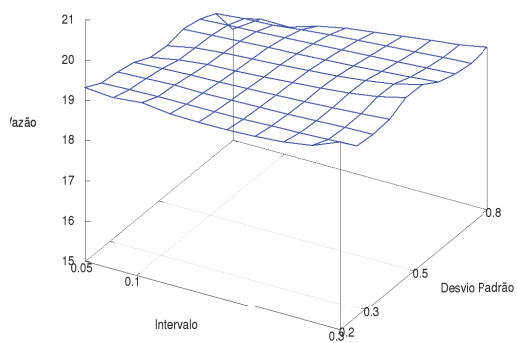
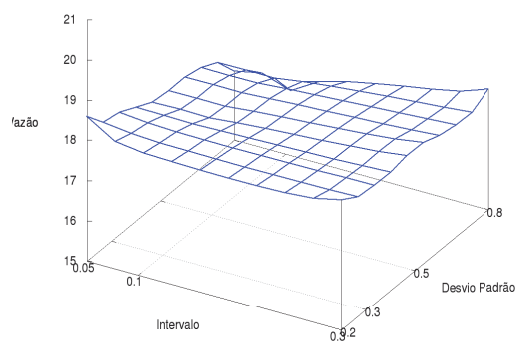
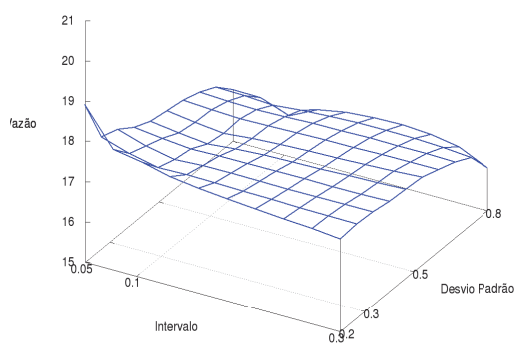
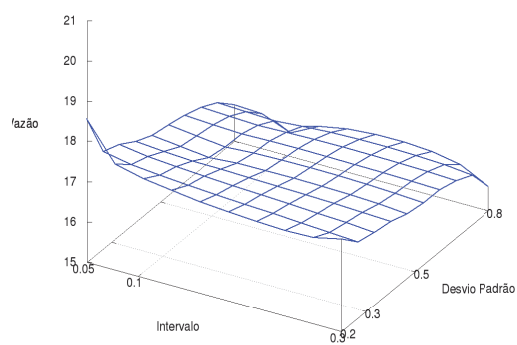
(a) $\alpha = 0.3$ (b) $\alpha = 0.5$ (c) $\alpha = 0.7$ (d) $\alpha = 0.9$

Figura 6.9: Vazão agregada média em função do alpha para diferentes combinações de intervalo e do desvio padrão em uma rede 802.11a infraestruturada.

tamento conservativo, assim, informações antigas são mais influentes. Logo, esta opção torna-se melhor para este tipo de topologia.

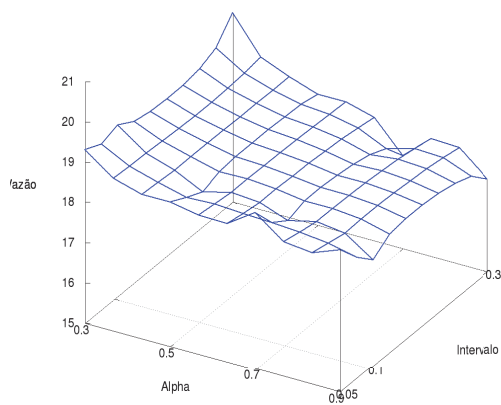
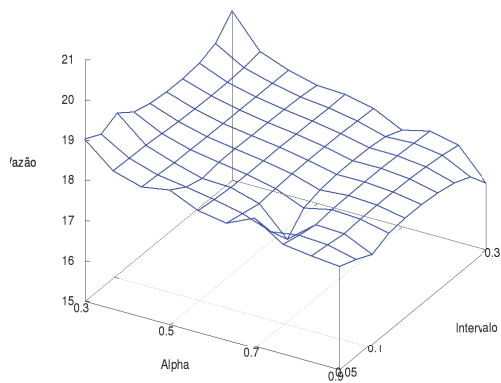
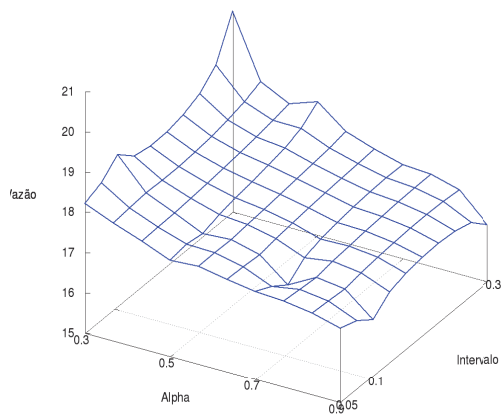
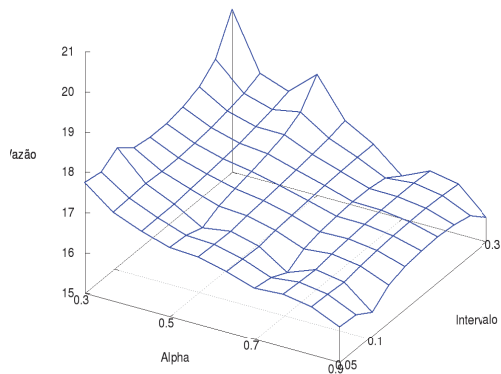
(a) $\sigma = 0.2$ (b) $\sigma = 0.3$ (c) $\sigma = 0.5$ (d) $\sigma = 0.8$

Figura 6.10: Vazão agregada média em função do desvio padrão para diferentes combinações de intervalo e de alpha em uma rede 802.11a infraestruturada.

Na Figura 6.10, é apresentado diferentes combinações de σ e I em função do σ . Percebe-se que, não há uma configuração para o desvio padrão σ que

sobresai-se. Mas de modo geral, valores menores apresentam resultados melhores. Em configurações em que o α é menor e o I é maior, o σ com um valor mediano ($\sigma = 0.5$) apresenta melhores resultados.

Tabela 6.3: Vazão agregada média alcançada por uma amostra das 12 melhores configurações do CORA em uma rede 802.11a infraestruturada em um cenário com 12 estações.

| α | I | σ | Vazão | Intervalo de confiança |
|----------|------|----------|----------|------------------------|
| 0.3 | 0.3 | 0.5 | 19.99488 | 0.6495505572 |
| 0.3 | 0.3 | 0.2 | 19.70406 | 0.8803189952 |
| 0.3 | 0.1 | 0.2 | 19.37306 | 0.60100393 |
| 0.3 | 0.05 | 0.2 | 19.31766 | 0.3947785544 |
| 0.3 | 0.3 | 0.3 | 19.19804 | 0.8662229626 |
| 0.3 | 0.1 | 0.3 | 19.16194 | 0.351748901 |
| 0.3 | 0.3 | 0.8 | 19.03602 | 0.7534343108 |
| 0.3 | 0.05 | 0.3 | 19.02166 | 0.1780742637 |
| 0.3 | 0.1 | 0.5 | 18.98982 | 0.2885641071 |
| 0.7 | 0.05 | 0.2 | 18.91302 | 0.3928255079 |
| 0.5 | 0.05 | 0.2 | 18.59688 | 0.7871587998 |
| 0.9 | 0.05 | 0.2 | 18.57056 | 0.2344202002 |

Com base nestas análises, em topologias em que a condição do canal não é tão degradada e a disputa pelo acesso ao meio é maior, é mais interessante configurar o CORA da seguinte forma:

- agressividade mediana ($\sigma = 0.5$). Para reforçar as análises feitas, a Tabela 6.3 apresenta 12 amostras com as melhores configurações do CORA para esta topologia. Pode-se analisar, que mesmo que a melhor configuração tenha uma agressividade mediana, as próximas 5 melhores configurações tem uma agressividade menor, ou seja, o desvio padrão é 0.2 ou 0.3. De forma geral, valores menores podem ser mais interessantes.
- alpha menor ($\alpha = 0.3$). Informações antigas são mais relevantes pois não há grandes mudanças no canal
- intervalo maior ($I = 0.3$). Como não há uma grande degradação do canal não há necessidade de chamar o ciclo várias vezes.

Para os cenários com 6 e 18 estações a melhor configuração, para ambos, é: intervalo $I = 0.3$, alpha $\alpha = 0.3$ e desvio padrão $\sigma = 0.3$. Estes resultados realçam que valores baixos para o desvio padrão σ podem ser mais interessantes. Comparando estes resultados com o que foi analisado conclui-se que, como não há grandes alterações nas configurações pode-se concluir que as análises são análogas. Dessa forma, pode-se manter as mesmas conclusões que foram obtidas com o cenário contendo 12 estações.

Em geral, neste tipo de topologia torna-se interessante configurações em que a característica é corvegir lentamente pois não há grandes mudanças na condição do canal.

6.3 Comportamento dos algoritmos

Para analisar o comportamento dos algoritmos foi utilizado a mesma topologia e os mesmos cenários da Seção 6.1. Observando em conjunto as Figuras 6.11, 6.12, 6.13, 6.14 e 6.15, cada ponto no gráfico representa a taxa escolhida (eixo y) naquele instante.

O ARF sempre alterna a taxa entre valores próximos, o que piora o desempenho. Isto acontece pois o ARF tenta mudar para uma taxa superior mesmo que aquela taxa não melhore o desempenho da rede. O CARA tem o mesmo princípio do ARF, mas usa mensagens RTS/CTS para identificar se a qualidade do canal é boa. Dessa forma, o CARA sofre dos mesmo problemas do ARF mas com o agravante de ter que enviar o RTS/CTS, e por isto tem um desempenho pior que o ARF. O RRAA faz uso do chamado RTS adaptado, que reage rapidamente a mudanças no canal e não degrada a rede como acontece com o CARA. Dessa forma, o RRAA é mais eficiente que a abordagem CARA. Já o Minstrel perde desempenho pois envia quadros em outras taxas, escolhidas aleatoriamente, fazendo várias escolhas inadequadas para um cenário simples. Dessa forma, ele tem um desempenho pior que o ARF, RRAA, CARA e até que algumas taxas fixas. O CORA

também sorteia diferentes taxas, sendo possível notar que em vários momentos há o uso de taxas inapropriadas para aquele momento. Mas como o cenário é simples e o CORA está configurado com uma agressividade baixa não há um número elevado de taxas inapropriadas sendo escolhidas. Assim, seu desempenho não é afetado. O RBAR tenta sempre escolher a taxa mais alta, por esta razão tem o pior desempenho dentre os algoritmos e de algumas taxas constantes.

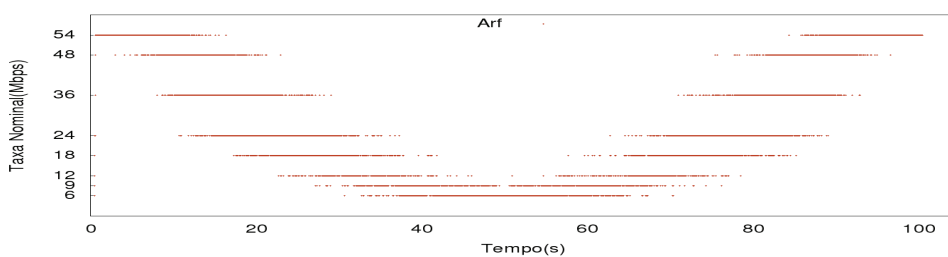


Figura 6.11: Controle da taxa em função do tempo realizado pelo ARF em redes 802.11a infraestruturadas com velocidade igual a 1.5 m/s e tamanho de pacote igual a 1460 bytes

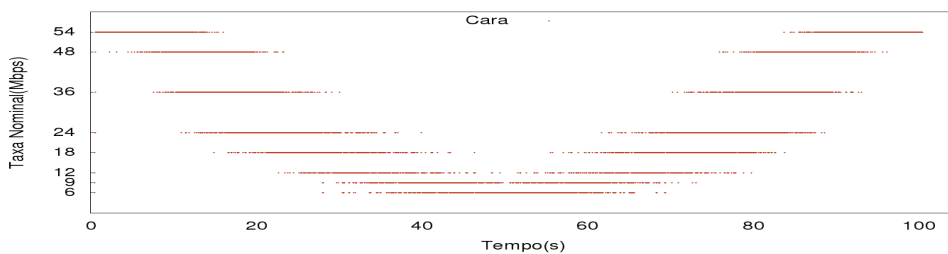


Figura 6.12: Controle da taxa em função do tempo realizado pelo CARA em redes 802.11a infraestruturadas com velocidade igual a 1.5 m/s e tamanho de pacote igual a 1460 bytes

Desempenho dos algoritmos com

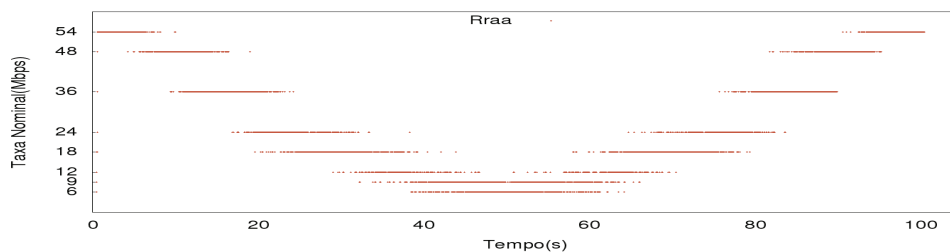


Figura 6.13: Controle da taxa em função do tempo realizado pelo RRAA em redes 802.11a infra-estruturadas com velocidade igual a 1.5 m/s e tamanho de pacote igual a 1460 bytes

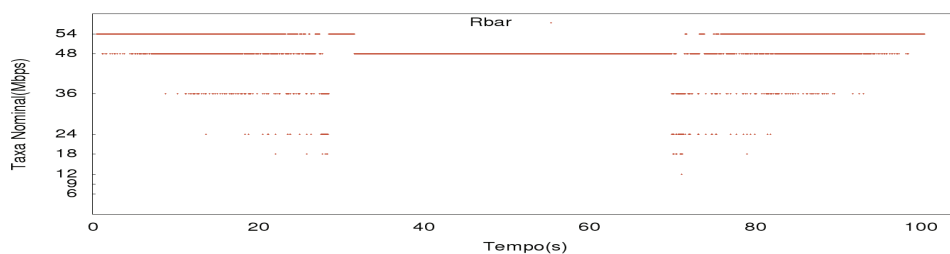


Figura 6.14: Controle da taxa em função do tempo realizado pelo RBAR em redes 802.11a infra-estruturadas com velocidade igual a 1.5 m/s e tamanho de pacote igual a 1460 bytes

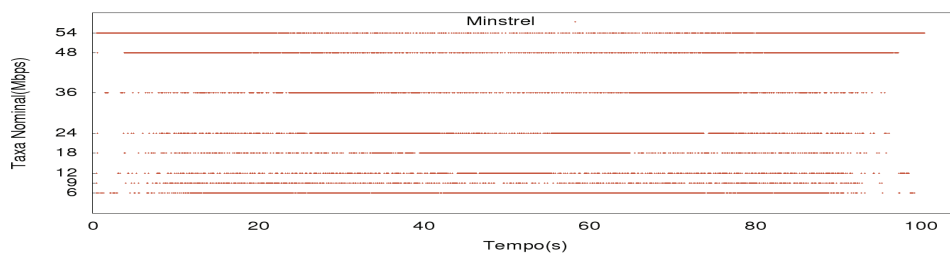


Figura 6.15: Controle da taxa em função do tempo realizado pelo Minstrel em redes 802.11a infra-estruturadas com velocidade igual a 1.5 m/s e tamanho de pacote igual a 1460 bytes

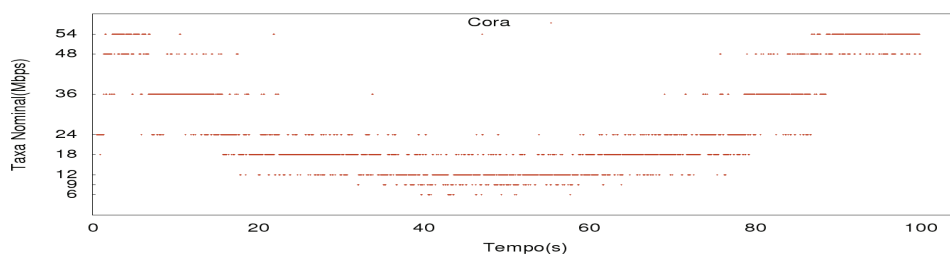


Figura 6.16: Controle da taxa em função do tempo realizado pelo CORA em redes 802.11a infra-estruturadas com velocidade igual a 1.5 m/s e tamanho de pacote igual a 1460 bytes

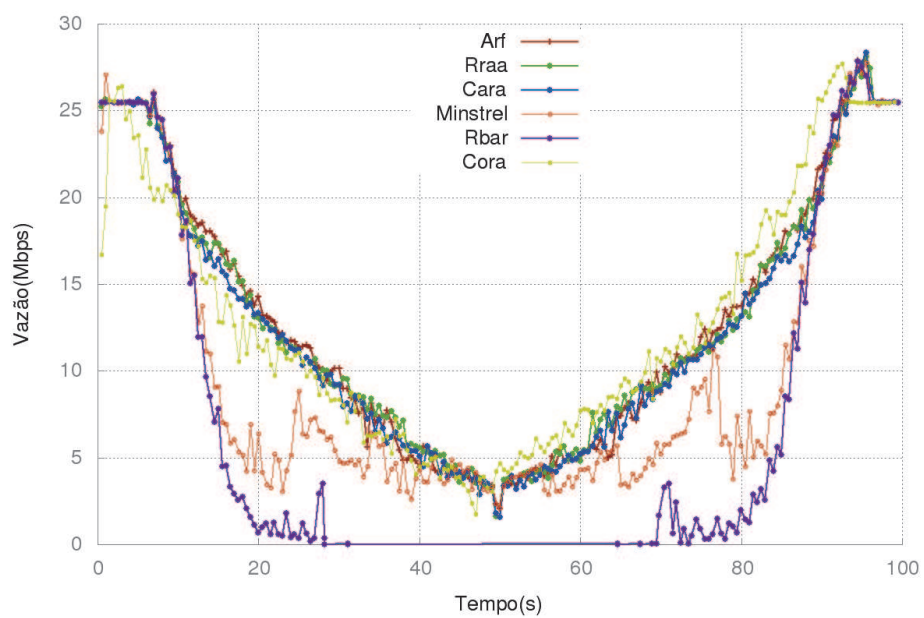


Figura 6.17: Vazão instantânea em função da qualidade do sinal ao longo do tempo para cada algoritmo em redes 802.11a configurados com os parâmetros velocidade - 1.5 m/s - e tamanho de pacotes - 1460 bytes

Como é visto na Figura 6.17, os algoritmos que apresentam desempenho melhor apresentam curvas com menos oscilações que os demais. Percebe-se que, até 15 segundos aproximadamente, todos estão transmitindo sem muitas oscilações, isto acontece porque a estação não se encontra tão distante do ponto de acesso. A partir deste instante, os algoritmos com pior desempenho oscilam muito, diminuindo a vazão no tempo. Dessa forma, isto mostra que estes algoritmos não estão fazendo as escolhas mais apropriadas para esta topologia. Por outro lado, os algoritmos com os melhores desempenhos conseguem aproximar a vazão obtida da vazão ótima. Um comparativo entre as Figuras 6.2 e 6.17, comprova que os algoritmos que apresentam melhores resultados tem um comportamento parecido com as taxas nominais, no momento em que estas maximizam a vazão. Em outras palavras, estes algoritmos convergem para a taxa nominal ótima. Por exemplo, o ARF aos 20 segundos transmite a uma taxa nominal de 24 Mbps. Comparando com a Figura 6.2, percebe-se que aos 20 segundos a taxa nominal que maximiza a vazão é exatamente a de 24 Mbps.

6.4 Desempenho dos algoritmos

Nesta seção, será avaliado o desempenho dos algoritmos de controle nas duas topologias que foram apresentadas nas seções anteriores.

6.4.1 Desempenho em uma Rede Infraestruturada

Foi utilizado a mesma topologia e cenários da Seção 6.1 para analisar o desempenho do algoritmos em redes infraestruturadas. Analisando em conjunto as Figuras 6.18 e 6.19, o CORA apresenta, em quase todos os cenários, uma taxa de perda de pacotes menor e uma vazão agregada média maior que as taxas nominais e os algoritmos de controle. Somente em cenários com o tamanho de pacotes igual a 1000 bytes o ARF leva uma pequena vantagem sobre o CORA, podendo-se concluir através do intervalo de confiança que houve um empate. De forma geral, este

algoritmo apresenta o melhor desempenho em comparação aos demais. O ARF apresenta o segundo melhor desempenho, isto acontece porque o cenário é muito simples e o ARF não utiliza de nenhum mecanismo extra para ajudar na escolha da taxa, não degradando o desempenho. Além dos algoritmos é apresentado as taxas de transmissão fixa com o pior e o melhor desempenho, que são, 6Mbps e 18Mbps, respectivamente. Em comparação aos algoritmos a taxa de 18Mbps consegue alcançar resultados melhores que RBAR e Minstrel. Por outro lado, os demais algoritmos sempre ganham das taxas fixas.

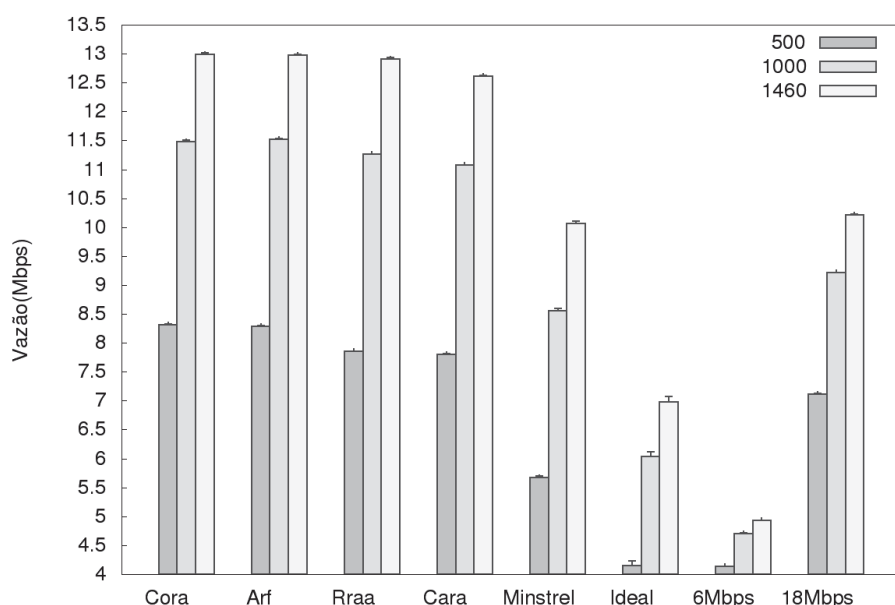


Figura 6.18: Vazão média alcançada pelos algoritmos e taxas de transmissão fixas em redes 802.11a infraestruturas em cenários com velocidade igual a 1.5 m/s.

Na Tabela 6.4, é apresentado o quanto em ganho percentual o CORA é melhor que os demais. Por exemplo, o algoritmo que chega mais próximo do desempenho do CORA é o ARF, sendo 0.12% pior que ele. Por outro lado, o RBAR é o algoritmo com o pior desempenho e em relação ao CORA apresenta um desempenho de 42.27% inferior.

| Algoritmo | Ganho (%) |
|-----------|-----------|
| ARF | 0.12% |
| RRAA | 0.58% |
| CARA | 2.86% |
| Minstrel | 22.49% |
| RBAR | 42.27% |

Tabela 6.4: Comparação do ganho percentual do CORA aos demais algoritmos no cenário com velocidade igual a 1.5 m/s e tamanho dos pacotes igual a 1460 bytes

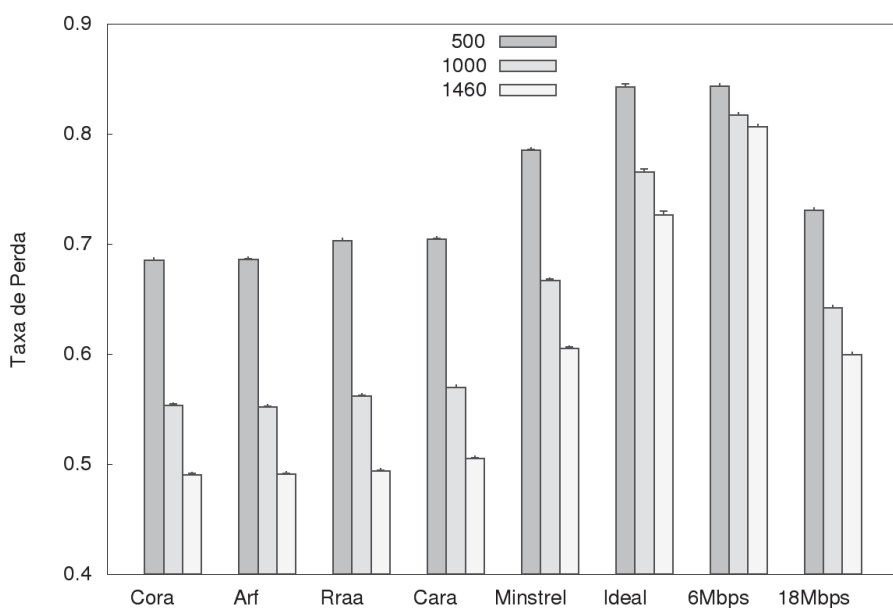


Figura 6.19: Taxa de perda de pacotes média alcançada pelos algoritmos e taxas de transmissão fixas em redes 802.11a infraestruturadas em cenários com velocidade igual a 1.5 m/s

Analisando a Figura 6.19, pode-se perceber que em cenários com pacotes de tamanho menor a taxa de perda é maior. Isto acontece porque há uma necessidade de enviar mais pacotes e quanto maior a quantidade de pacotes enviados na rede maior a possibilidade de perder dados por colisão de pacotes.

Os cenários configurados com a velocidade de 10 m/s alcançaram resultados coerentes em relação aos simulados com a velocidade de 1.5 m/s.

6.4.2 Desempenho com múltiplos nós

Neste estudo de caso será analisado o comportamento dos algoritmos quando utilizados em uma topologia com múltiplos nós disputando o acesso ao meio em uma rede infraestruturada. Esta topologia foi apresentada na Seção 6.2.2.

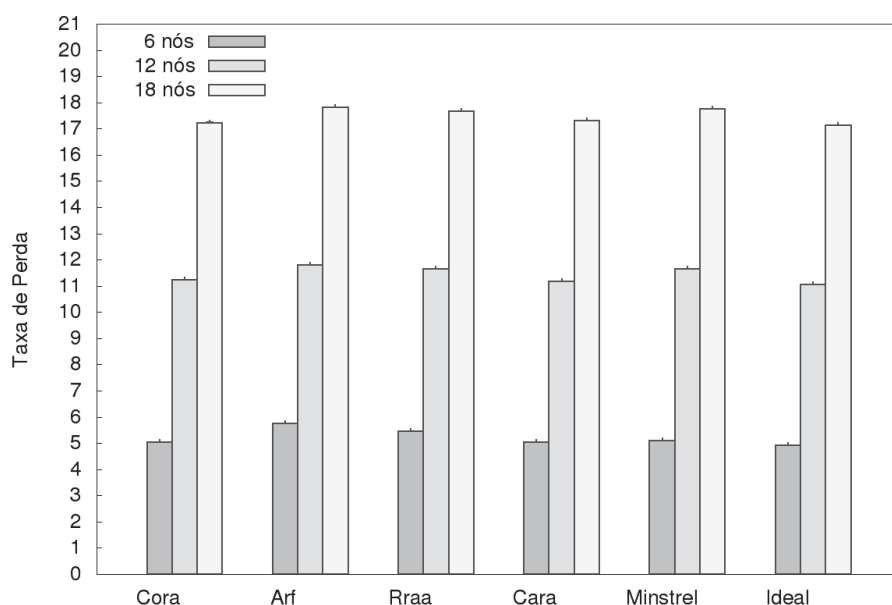


Figura 6.20: Taxa de perda de pacote média alcançada pelos algoritmos de controle da taxa em redes 802.11a infraestruturadas com múltiplos nós

Para estudar o comportamento dos algoritmos com múltiplos nós o parâmetro quantidade de estações, pode ser: 6, 12 ou 18.

Analisando em conjunto as Figuras 6.21 e 6.22, percebe-se que, com o aumento do número de nós a vazão agregada média diminui. Isto acontece pois há um aumento na quantidade de colisões ocorridas no canal. O ARF, o Minstrel e o RRAA tem o desempenho bastante degradado com o aumento do número de estações. Em particular, o RRAA, o CORA e o ARF não apresentam uma perda de desempenho acentuada quando o número de estações é aumentado de 12 para 18. O ARF apresenta o pior desempenho entre os algoritmos comparados. O

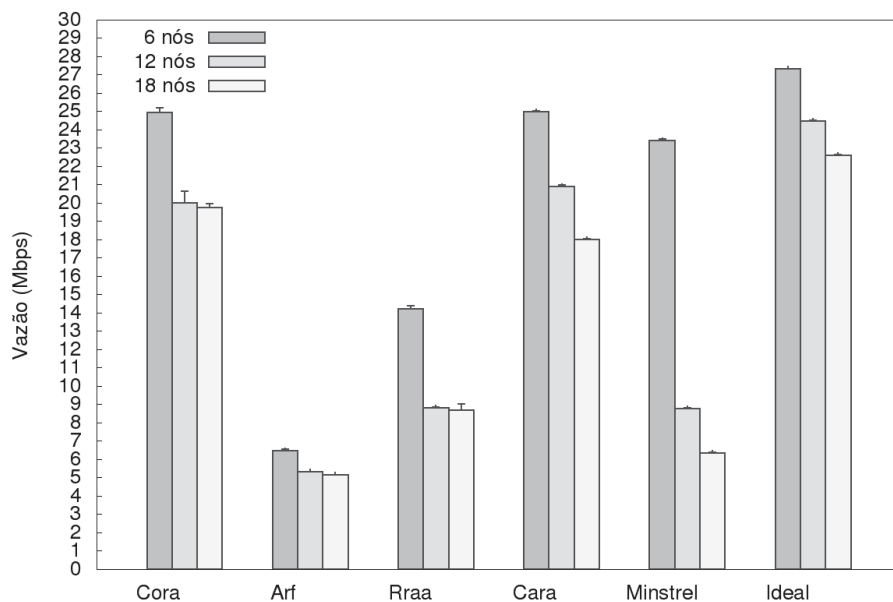
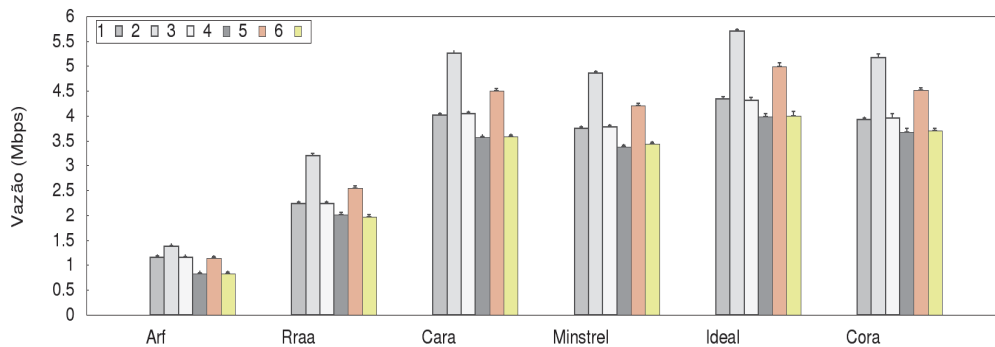
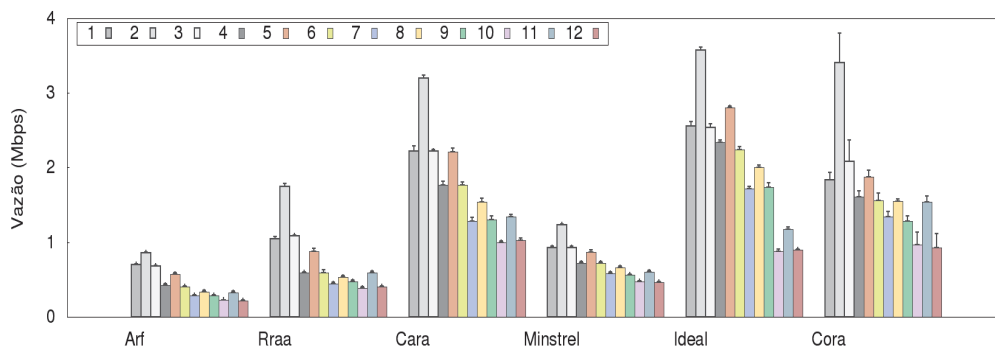


Figura 6.21: Vazão agregada média alcançada pelos algoritmos de controle da taxa em redes 802.11a infraestruturadas com múltiplos nós

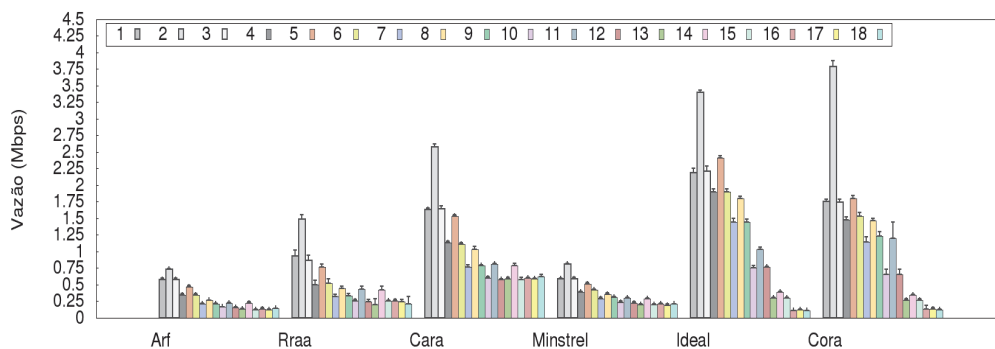
Minstrel tem o seu desempenho degradado por um conjunto de dois fatores: escolher inúmeras taxas aleatórias e por também não possuir nenhum mecanismo que faça escolhas por enlace. Assim, pode-se observar que, com 12 e 18 estações este mecanismo tem um desempenho pior que a abordagem RRAA. Por outro lado, as abordagens RBAR e CARA alcançam bons resultados e não apresentam uma queda brusca de desempenho. Isto acontece devido a utilização de quadros RTS/CTS o que acarreta na redução na quantidade de colisões, e por isto, apresentam melhor desempenho em cenários em que há múltiplos nós. Lembrando que, os quadros RTC/CTS foram modificados na abordagem RBAR para informar as estações vizinhas o tempo de duração da transmissão. Sendo mais um fator para o RBAR ter alcançado resultados melhores. Então, como há uma disputa maior por acesso ao meio nestes ambientes, diminuir a quantidade de colisões nesta topologia torna um fator importante. Estes comportamentos foram observados em (CHAVES *et al.*, 2009), (CARDOSO, 2009) e (RAMACHANDRAN *et al.*, 2007).



(a) 6 estações



(b) 12 estações



(c) 18 estações

Figura 6.22: Vazão média por fluxo alcançada pelos algoritmos de controle da taxa em redes 802.11a infraestruturadas com múltiplos nós

Comparando em conjunto a Tabela 6.3 com a Figura 6.21, é observado que em cenários com 12 estações a melhor configuração para o CORA perde para as abordagens RBAR e CARA. O RBAR tem um desempenho de ≈ 24.5 e o CARA de ≈ 20.9 contra ≈ 20 do CORA. Pode-se observar também que todas as configurações possíveis são melhores que as abordagens ARF, Minstrel e RRAA. Estas observações são análogas em cenários com 6 estações. Quando há um aumento de 12 para 18 estações o CORA tem seu desempenho pouco afetado. Dessa forma, em cenários com 18 estações o CORA alcança resultados melhores que o CARA.

Estas análises são realçadas na Figura 6.20, em que, pode-se verificar que quanto maior o número de nós há uma perda de dados maior pois a disputa pelo acesso ao meio aumenta. Além disso, é realçado que o RBAR tem os melhores resultados pois tem uma taxa de perda de dados média menor que as outras abordagens.

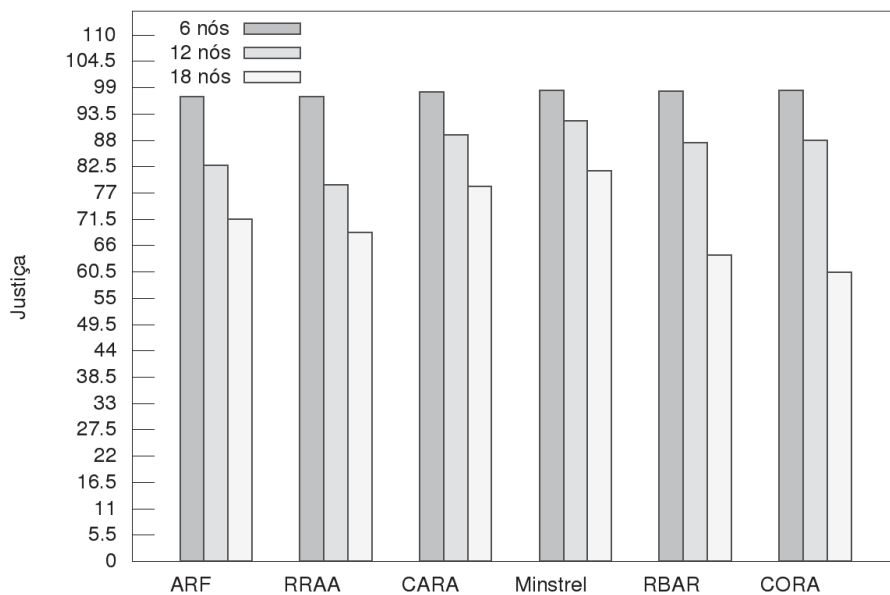


Figura 6.23: Nível de justiça dos algoritmos de controle de taxa simulados em redes com múltiplos nós

Para seis clientes, o Minstrel apresenta uma vazão agregada média inferior em relação ao CARA e o RBAR. Por outro lado, como é visto na Figura 6.23, o grau de justiça alcançado pelo Minstrel é superior à estas duas abordagens, sendo de 98.33% contra 98.09% e 98.23%, respectivamente. Assim, o desempenho geral do Minstrel é afetado pois há uma divisão mais justa da taxa de fluxo. Este comportamento pode ser observado de forma mais acentuada no cenário simulado com 12 e 18 estações. Esta observação pode ser feita também para a abordagem CORA, quando o cenário contém 6 e 12 estações. o CORA não apresenta desempenhos melhores que o RBAR mas mostra um nível de justiça maior. Em um cenário com 18 estações isto não acontece, pois o CORA apresenta um nível de justiça inferior a todos os outros algoritmos. O RBAR também apresenta um comportamento semelhante ao CORA. Mesmo o RBAR apresentando a maior vazão agregada média, ao aumentar o número de estações o índice de justiça diminui acentuadamente. Nas Figuras 6.22b e 6.22c, fica mais evidente os baixos níveis de justiça do CORA e RBAR, com 12 e 18 estações. Pode-se notar que a segunda estação apresenta uma vazão bem maior que as demais. Assim, deduz que os recursos da rede não estão sendo distribuídos de forma equalitária.

7 CONCLUSÃO

Neste trabalho, foi feita a implementação do CORA no simulador NS-3 e uma avaliação de desempenho dos algoritmos de controle da taxa.

Primeiramente, foram estudados conceitos básicos de redes de sem fio, mais precisamente, aqueles relacionados ao padrão 802.11. Estes conceitos são o funcionamento da subcamada MAC, modulação, taxas de transmissão e o controle da taxa.

Em seguida, foi realizado um levantamento bibliográfico dos principais algoritmos de controle da taxa. Dessa forma, foi apresentando o funcionamento de cada algoritmo, suas vantagens e limitações. A seguir, foi feita uma tabela comparativa para mostrar as principais características de cada abordagem.

Após estudar os algoritmos, foi necessário realizar um estudo aprofundado do simulador para implementar o CORA. Após isto, foi feita uma avaliação de desempenho de alguns dos algoritmos estudados. Alguns destes algoritmos já continham sua implementação no simulador.

No primeiro cenário, com apenas uma estação móvel e o ponto de acesso, o CORA obteve um desempenho melhor que os demais algoritmos. Mas, em relação ao ARF, o CORA obteve pouca vantagem. O fator principal para as outras abordagens não alcançarem um desempenho satisfatório é devido ao uso de mecanismos que envolvem o envio de mensagens RTS/CTS ou por tentarem sempre usar uma taxa de transmissão nominal de maior valor. Em cenários com múltiplos nós, foi observado que abordagens que possuem algum tipo de mecanismo que auxilia na escolha da taxa alcançaram resultados melhores. É o caso do CARA e do RBAR que utiliza mensagens RTS/CTS e do Minstrel que envia alguns quadros em diferentes taxas. O CORA também apresenta bons resultados em relação aos demais, perdendo apenas para o RBAR e o Minstrel. Por outro lado, os níveis de justiça alcançados pelo RBAR não são tão bons mesmo sendo o algoritmo que alcançou o melhor desempenho.

Dessa forma, as principais contribuições deste trabalho foram: um levantamento bibliográfico dos principais algoritmos de controle, uma implementação do CORA no simulador e uma avaliação de desempenho destes algoritmos.

Como trabalho futuro, será feita a submissão do código fonte do CORA ao projeto que mantém o NS3 na tentativa de que o algoritmo possa ser incorporado na distribuição oficial do simulador. O código do CORA será aprimorado e modificado para dar suporte a redes *ad hoc* e será incluída uma nova métrica: a taxa de perda de pacotes. Para incluir esta nova métrica é necessário fazer mudanças no método *Observe* e incluir entre os parâmetros configuráveis um atributo para a escolha da métrica, vazão de dados ou taxa de perda.

REFERÊNCIAS BIBLIOGRÁFICAS

BIAZ, S.; WU, S. Era: Effective rate adaptation for wlangs. In: *NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*. [S.l.]: Springer Berlin Heidelberg, 2008. v. 4982, p. 901–913.

BIAZ, S.; WU, S. Rate adaptation algorithms for ieee 802.11 networks: A survey and comparison. In: *Computers and Communications, 2008. ISCC 2008. IEEE Symposium on*. [S.l.: s.n.], 2008. p. 130–136.

CARDOSO, K. V. Controle automático de taxa de transmissão em redes ieee 802.11. *COPPE/PEE/UFRJ*, p. 152, Fevereiro 2009.

CHAVES, L.; MALHEIROS, N.; MADEIRA, E.; GARCIA, I.; KLIAZOVICH, D. A cognitive mechanism for rate adaptation in wireless networks. In: STRASSNER, J.; GHAMRI-DOUDANE, Y. (Ed.). *Modelling Autonomic Communications Environments*. [S.l.]: Springer Berlin Heidelberg, 2009, (Lecture Notes in Computer Science, v. 5844). p. 58–71.

CHEN, X.; GANGWAL, P.; QIAO, D. Practical rate adaptation in mobile environments. In: *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*. [S.l.: s.n.], 2009. p. 1–10.

FIETKAU, F. *Minstrel rate control algorithm*. 2005. Available at <http://linuxwireless.org/en/developers/Documentation/mac80211/RateControl/minstrel>. Disponível em: <http://linuxwireless.org/en/developers/Documentation/mac80211/RateControl/minstrel>.

HARATCHEREV, I.; LANGENDOEN, K.; LAGENDIJK, R.; SIPS, H. Hybrid rate control for ieee 802.11. New York, NY, USA, p. 10–18, 2004.

HOLLAND, G.; VAIDYA, N.; BAHL, P. A rate-adaptive mac protocol for multi-hop wireless networks. In: *Proceedings of the 7th annual international conference on Mobile computing and networking*. [S.l.: s.n.], 2001. p. 236–251.

HUANG, T.; CHEN, H.; ZHANG, Z.; CUI, L. Easira: A hybrid rate adaptation scheme for 802.11 mobile wireless access networks. In: *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*. [S.l.: s.n.], 2012. p. 1520–1525.

JAIN, R. K.; CHIU, D.-M. W.; HAWE, W. R. *A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems*. [S.l.], 1984. Disponível em <http://arxiv.org/abs/cs.NI/9809099>. Disponível em: <http://arxiv.org/abs/cs.NI/9809099>.

JIWIRE. 2014. Disponível em: <http://v4.jiwire.com/search-hotspot-locations.htm>.

KAMERMAN, A.; MONTEBAN, L. Wavelan®-ii: a high-performance wireless lan for the unlicensed band. *Bell Labs Tech. J.*, v. 2, p. 118–133, 1997.

KIM, J.; KIM, S.; CHOI, S.; QIAO, D. Cara: Collision-aware rate adaptation for ieee 802.11 wlans. In: *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*. [S.l.: s.n.], 2006. p. 1–11.

KUROSE J. E ROSS, K. *Computer Networking: A Top-Down Approach*. [S.l.]: Pearson Addison Wesley, 5a edição., 2011.

LACAGE, M.; MANSHAEI, M. H.; TURLETTI, T. Ieee 802.11 rate adaptation: a practical approach. In: *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*. [S.l.: s.n.], 2004. (MSWiM '04), p. 126–134.

MALHEIROS, N.; KLIAZOVICH, D.; GRANELLI, F.; MADEIRA, E.; FONSECA, N. da. A cognitive approach for cross-layer performance management. In: *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*. [S.l.: s.n.], 2010. p. 1–5.

MORRIS, R. T.; BICKET, J. C.; BICKET, J. C. *Bit-rate selection in wireless networks*. [S.l.], 2005.

RAMACHANDRAN, K.; KREMO, H.; GRUTESER, M.; SPASOJEVIC, P.; SESKAR, I. Scalability analysis of rate adaptation techniques in congested ieee 802.11 networks: An orbit testbed comparative study. In: *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*. [S.l.: s.n.], 2007. p. 1–12.

SADEGHI, B.; KANODIA, V.; SABHARWAL, A.; KNIGHTLY, E. Opportunistic media access for multirate ad hoc networks. In: *Proceedings of the 8th annual international conference on Mobile computing and networking*. [S.l.: s.n.], 2002. p. 24–35.

WONG, S. H. Y.; YANG, H.; LU, S.; BHARGHAVAN, V. Robust rate adaptation for 802.11 wireless networks. In: *Proceedings of the 12th annual international conference on Mobile computing and networking*. [S.l.: s.n.], 2006. (MobiCom '06), p. 146–157.

A CÓDIGO FONTE DO CORA

```

/* -- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -- */
/* Author: Lucas Jose Duarte de Souza <lucasjds@computacao.ufla.br>*/
#include "cora-wifi-manager.h"
#include "ns3/assert.h"
#include "ns3/log.h"
#include "ns3/uinteger.h"
#include "wifi-phy.h"
#include "ns3/string.h"
#include "ns3/wifi-module.h"
#include "ns3/core-module.h"
#include "ns3/mesh-module.h"
#include "ns3/network-module.h"
#define Min(a,b) ((a < b) ? a : b)
NS_LOG_COMPONENT_DEFINE ("ns3::CoraWifiManager");
using namespace std;
namespace ns3 {

struct CoraWifiRemoteStation : public WifiRemoteStation{
    bool m_flag;
    bool m_flagTrace;
    double m_contPackets;
    double m_throughput;
    uint32_t m_txRate;
    double m_mean;
};

NS_OBJECT_ENSURE_REGISTERED (CoraWifiManager);
TypeId CoraWifiManager::GetTypeId (void){
    static TypeId tid = TypeId ("ns3::CoraWifiManager")
        .SetParent<WifiRemoteStationManager> ()
        .AddConstructor<CoraWifiManager> ()
        .AddAttribute ("Interval",

```

```

        "Interval of CORA",
        DoubleValue (0.1),
        MakeDoubleAccessor (&CoraWifiManager::m_interval),
        MakeDoubleChecker <double>())
    .AddAttribute ("Alpha",
        "",
        DoubleValue (0.9),
        MakeDoubleAccessor (&CoraWifiManager::m_alpha),
        MakeDoubleChecker <double>())
    .AddAttribute ("StandardDeviation",
        "",
        DoubleValue (0.3),
        MakeDoubleAccessor (&CoraWifiManager::m_standardDeviation),
        MakeDoubleChecker <double>());

    return tid;
}

CoraWifiManager::CoraWifiManager (){
    NS_LOG_FUNCTION (this);
}

CoraWifiManager::~CoraWifiManager (){
    NS_LOG_FUNCTION (this);
}

WifiRemoteStation * CoraWifiManager::DoCreateStation (void) const{
    NS_LOG_FUNCTION (this);
    CoraWifiRemoteStation *station = new CoraWifiRemoteStation ();
    station->m_flag = true;
    station->m_flagTrace = false;
    station->m_contPackets = 0;
    station->m_throughput = 0;
    station->m_txRate = 0;
    station->m_mean = 0;
    return station;
}

```

```

}

void CoraWifiManager::kBaseInit(WifiRemoteStation *st){
    CoraWifiRemoteStation *station = (CoraWifiRemoteStation *)st;
    for (uint32_t i = 0; i < GetNSupported(station); i++)
        m_knowledgeBase[i] = 0;
}

void CoraWifiManager::checkInit(WifiRemoteStation *st){
    CoraWifiRemoteStation *station = (CoraWifiRemoteStation *)st;
    if (station->m_flag && GetNSupported(station) > 1){
        station->m_txRate = GetNSupported(station) / 2;
        m_knowledgeBase = KBase( GetNSupported (station));
        CoraWifiManager::kBaseInit(st);
        ostreamstream oss ;
        oss << "/NodeList/*/DeviceList*/$ns3::WifiNetDevice/Mac/MacRx";
        Config::Connect(oss.str(), MakeCallback(&CoraWifiManager::ReceivePacket, this) );
        Simulator::Schedule(Seconds(m_interval), &CoraWifiManager::RunAlgorithm, this, st);
        station->m_flag = false;
    }
}

void CoraWifiManager::SetupPhy (Ptr<WifiPhy> phy){
    WifiRemoteStationManager::SetupPhy (phy);
}

void CoraWifiManager::DoReportRxOk (WifiRemoteStation *st,
                                     double rxSnr, WifiMode txMode){
    NS_LOG_FUNCTION (this << st << rxSnr << txMode);
}

void CoraWifiManager::DoReportRtsFailed (WifiRemoteStation *station){
    NS_LOG_FUNCTION (this << station);
}

```

```

void CoraWifiManager::DoReportDataFailed (WifiRemoteStation *st){
    NS_LOG_FUNCTION (this << st);
}

void CoraWifiManager::DoReportRtsOk (WifiRemoteStation *st,
                                     double ctsSnr, WifiMode ctsMode, double rtsSnr){
    NS_LOG_FUNCTION (this << st << ctsSnr << ctsMode << rtsSnr);
}

void CoraWifiManager::DoReportDataOk (WifiRemoteStation *st,
                                       double ackSnr, WifiMode ackMode, double dataSnr){
    CoraWifiRemoteStation *station = (CoraWifiRemoteStation *)st;
    NS_LOG_FUNCTION (this << st << ackSnr << ackMode << dataSnr);
    station->m_contPackets++;
    CoraWifiManager::checkInit (st);
}

void CoraWifiManager::DoReportFinalRtsFailed (WifiRemoteStation *station){
    NS_LOG_FUNCTION (this << station);
}

void CoraWifiManager::DoReportFinalDataFailed (WifiRemoteStation *station){
    NS_LOG_FUNCTION (this << station);
}

WifiTxVector CoraWifiManager::DoGetDataTxVector (WifiRemoteStation *st, uint32_t size){
    NS_LOG_FUNCTION (this << st << size);
    CoraWifiRemoteStation *station = (CoraWifiRemoteStation *)st;
    CoraWifiManager::checkInit (st);
    return WifiTxVector (GetSupported (station, station->m_txRate),
                        GetDefaultTxPowerLevel (),    GetLongRetryCount (st),
                        GetShortGuardInterval (st), Min (GetNumberOfReceiveAntennas (st),
                        GetNumberOfTransmitAntennas ()), GetNumberOfTransmitAntennas (st), GetStbc (st));
}

```



```

WifiTxVector CoraWifiManager::DoGetRtsTxVector (WifiRemoteStation *st){
    NS_LOG_FUNCTION (this << st);
    CoraWifiRemoteStation *station = (CoraWifiRemoteStation *) st;
    return WifiTxVector (GetSupported (station, 0),
        GetDefaultTxPowerLevel (), GetShortRetryCount (st), GetShortGuardInterval (st),
        Min (GetNumberOfReceiveAntennas (st), GetNumberOfTransmitAntennas ()),
        GetNumberOfTransmitAntennas (st), GetStbc (st));
}

bool CoraWifiManager::IsLowLatency (void) const{
    NS_LOG_FUNCTION (this);
    return true;
}

void CoraWifiManager::ReceivePacket (string context, Ptr<const Packet> packet){
    m_sizePacket = packet->GetSize ();
}

void CoraWifiManager::RunAlgorithm (WifiRemoteStation *st){
    CoraWifiRemoteStation *station = (CoraWifiRemoteStation *)st;
    CoraWifiManager::Observe (st);
    CoraWifiManager::Orient (st);
    CoraWifiManager::Decide (st);
    station->m_contPackets =0;
    Simulator::Schedule (Seconds (m_interval), &CoraWifiManager::RunAlgorithm, this, st);
    ostream oss ;
    oss << "/NodeList/*/DeviceList*/$ns3::WifiNetDevice/Mac/MacRx";
}

void CoraWifiManager::Observe (WifiRemoteStation *st){
    CoraWifiRemoteStation *station = (CoraWifiRemoteStation *)st;
    double totalMB, totalMbits ,totalBytes;
    totalBytes = station->m_contPackets * m_sizePacket;
    totalMB = totalBytes / 1048576;
    totalMbits = totalMB * 8;
}

```

```

station->m_throughput = totalMbits / m_interval;
m_knowledgeBase[station->m_txRate] = (1 - m_alpha) * m_knowledgeBase[station->m_txRate]
    + m_alpha * station->m_throughput;
}

void CoraWifiManager::Orient(WifiRemoteStation *st){
    CoraWifiRemoteStation *station = (CoraWifiRemoteStation *)st;
    uint32_t higherRate, higherThroughput;
    higherRate = 0;
    higherThroughput = m_knowledgeBase[0];
    for ( uint32_t i =0; i < GetNSupported(station); i++){
        if( m_knowledgeBase[i] > higherThroughput){
            higherRate = i;
            higherThroughput = m_knowledgeBase[i];
        }
    }
    station->m_mean = higherRate;
}

void CoraWifiManager::Decide(WifiRemoteStation *st){
    CoraWifiRemoteStation *station = (CoraWifiRemoteStation *)st;
    Ptr<NormalRandomVariable> nRandom = CreateObject<NormalRandomVariable> ();
    double n;
    nRandom->SetAttribute("Mean", DoubleValue(station->m_mean));
    nRandom->SetAttribute("Variance", DoubleValue(m_standardDeviation));
    n = nRandom->GetValue();
    n += 0.5;
    //act
    if (n < GetNSupported(station) && n >= 0)
        station->m_txRate = n;
}
}

```