



PROBLEMA DE RENDEZVOUS EM REDES DE RÁDIOS COGNITIVOS

Raphael Melo Guedes

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Elétrica.

Orientador: José Ferreira de Rezende

Rio de Janeiro
Setembro de 2015

PROBLEMA DE RENDEZVOUS EM REDES DE RÁDIOS COGNITIVOS

Raphael Melo Guedes

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. José Ferreira de Rezende, Dr.

Prof. Aloysio de Castro Pinto Pedroza, Dr.

Prof. Flávia Coimbra Delicato, D.Sc.

Prof. Ricardo Campanha Carrano, D.Sc.

Prof. Ronaldo Moreira Salles, Ph.D.

Prof. Sidney Cunha de Lucena, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2015

Guedes, Raphael Melo

Problema de Rendezvous em Redes de Rádios Cognitivos/Raphael Melo Guedes. – Rio de Janeiro: UFRJ/COPPE, 2015.

XIII, 86 p.: il.; 29, 7cm.

Orientador: José Ferreira de Rezende

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2015.

Referências Bibliográficas: p. 78 – 82.

1. Redes Sem-fio. 2. Rádio Cognitivo. 3. Canal de controle. 4. *Rendezvous*. I. Rezende, José Ferreira de. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Agradecimentos

Agradeço aos meus pais Antônio e Lucia, minhas irmãs Ana e Camile, ao Sebastião e a meu sobrinho Bruno pelo apoio.

Aos colegas do LAND: André, Carlos, Coutinho, Marcel, Gaspare, Guilherme Domingues, Guilherme Iecker, Thársus, Pedro Eusébio e todos os demais; pela ajuda, paciência e amizade.

Agradeço à Carolina pela grande ajuda em diversas situações, pelo apoio e inúmeros conselhos.

Aos amigos de faculdade: Daniel, Diego e Felipe que me ajudaram muito.

Aos profs. do LAND em especial ao professor José Rezende pela orientação e ajuda.

E claro agradeço ao Criador.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

PROBLEMA DE RENDEZVOUS EM REDES DE RÁDIOS COGNITIVOS

Raphael Melo Guedes

Setembro/2015

Orientador: José Ferreira de Rezende

Programa: Engenharia Elétrica

Os rádios cognitivos são uma solução promissora ao problema da escassez de espectro. Estes dispositivos podem acessar o espectro de forma oportunista a fim de tirar vantagem de faixas de frequências, isto é canais, não utilizados por usuários licenciados enquanto não os interfere. Este modelo de acesso oportunista exige que os rádios cognitivos alternem entre os canais, à procura de novas oportunidades e contatos com seus vizinhos para troca de informações necessárias ao gerenciamento do espectro. Este processo de busca, conhecido como *rendezvous*, é denominado *blind rendezvous* quando o rádio não utiliza uma solução centralizada.

Várias soluções têm sido propostas para lidar com o problema de *rendezvous*. Um grupo das soluções recorre a um canal, chamado Canal de Controle Comum (CCC), com o objetivo de funcionar como um “local” de encontro para os rádios. Outra abordagem se vale do próprio funcionamento dos rádios cognitivos, ou seja, o salto de canais para realizar o *rendezvous* entre os rádios. No entanto, ambas soluções apresentam deficiências.

Para o caso do CCC, temos o problema de gargalo do canal de controle quando um número grande de rádios acessa este único canal, além da dificuldade de definir um único canal de controle livre para todos os rádios da rede. No caso, de uso de salto de canais as soluções de *rendezvous* carecem de um meio para realizarem transmissões destinadas a múltiplos dispositivos, por exemplo transmissões em *broadcast*. Além disso, pouco se é comentado sobre o funcionamento da rede após o primeiro encontro dos rádios. Assim, este trabalho propõe soluções aos problemas citados, nos dois grupos de abordagens, e apresenta um protocolo de acesso ao meio aplicado após o primeiro encontro dos rádios.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

RENDEZVOUS PROBLEM IN COGNITIVE RADIO NETWORKS

Raphael Melo Guedes

September/2015

Advisor: José Ferreira de Rezende

Department: Electrical Engineering

Cognitive radios are a promising solution for the spectrum scarcity problem. These radios access the spectrum in an opportunistic manner in order to take advantage of frequency bands, i.e., channels, left unused by primary users while not interfering with them. This opportunistic access model requires cognitive radios to switch among channels searching for new opportunities and making contact with their neighbors to exchange the information needed for spectrum management. This search process, known as rendezvous, is termed blind rendezvous when it does not use a centralized solution.

Several solutions have been proposed to deal with the rendezvous problem. A group of solutions uses a channel called Common Control Channel (CCC), in order to act as a “meeting place” for the radios. Another approach relies on the proper functioning of the cognitive radios, i.e., the channel hopping to make the rendezvous among them. However, both groups of solutions have shortcomings.

In the case of CCC, there is a problem called channel bottleneck problem, when a large number of radios access the same single channel simultaneously. In addition, to set a single available control channel for all network radios is a difficult task. In the case of channel hopping solutions, most of them do not deal with broadcast transmissions, that is, when a cognitive radio wants to exchange information with all its neighboring nodes. Also, most works do not comment much about functioning of the network after the first meeting of radios. Then, this work proposes solutions to the cited problems in both groups of approaches, and presents a media access protocol applied after the first meeting of radios.

Sumário

Lista de Figuras	ix
Lista de Tabelas	xi
Lista de Abreviaturas	xii
1 Introdução	1
1.1 Organização do Texto	5
2 Rendezvous em Rádios Cognitivos	6
2.1 Classificação de Soluções para o Problema de <i>Rendezvous</i>	9
2.2 Múltiplos Canais de Controle	11
2.3 Sem Canal de controle	13
2.3.1 Sincronismo de Início de Sequências	15
2.3.2 Sincronismo de <i>Slot</i>	16
2.3.3 Exemplos de Soluções para <i>Rendezvous</i> que Usam Sequência de Salto de Canais	17
2.4 Deficiências do CCC e Protocolos Baseados em Sequência	21
3 Múltiplos Canais de Controle em um Ambiente de Grupo de Rádios	24
3.1 Múltiplos Canais de Controle e Sequência de Salto de Canais	24
3.2 Máquina de Aprendizagem <i>Reinforcement Learning</i>	26
3.3 Modelagem do problema	28
3.3.1 Algoritmo de <i>Reinforcement Learning</i>	31
3.3.2 Modelagem RL-Ciclo	32
3.3.3 Modelagem RL-Seq	33
3.4 Ambiente de Simulação	34
3.4.1 Resultados da Simulação	35
3.5 Conclusões do Capítulo	39

4	Transmissões em <i>Broadcast</i> Para Soluções Baseadas Em Sequência de Salto de Canais	41
4.1	Motivação para Sequência de <i>Broadcast</i>	41
4.2	Formalização da Sequência de Salto de Canais e Sequência de <i>Broadcast</i>	44
4.3	Algoritmos para Sequência de <i>broadcast</i>	47
4.4	Modelo de Simulação e Resultados	49
4.5	Conclusões do Capítulo	56
5	Regras de Funcionamento para uma Rede de Rádio Cognitivos	59
5.1	Motivação para o Conjunto de Regras	59
5.2	Regras de Funcionamento	62
5.3	Modelo do Sistema	63
5.4	Ambiente de Simulação	65
5.5	Resultados da Simulação	66
5.6	Conclusões do Capítulo	71
6	Conclusões e Trabalhos Futuros	74
	Referências Bibliográficas	78
A	<i>Modular Clock</i> formulação	83

Lista de Figuras

2.1	USs podem operar em diferentes canais independentemente.	6
2.2	Procedimento básico de <i>rendezvous</i> - canal para negociação (CCC) e envio de dados usando um canal de dados.	8
2.3	Exemplo de sequência de salto de canais entre os Rádios A e B.	9
2.4	Classificação de soluções para o problema de <i>rendezvous</i>	10
2.5	Sistemas síncronos - sincronismo de início de sequências.	16
2.6	Exemplo de duas sequências sem sincronismo de <i>slots</i>	17
2.7	Exemplo de sequências a partir do conjunto de <i>quorums</i>	18
2.8	Ciclo de <i>Rendezvous</i>	22
3.1	<i>Q-table</i> - exemplo de <i>exploitation</i> e exploração.	28
3.2	Método de funcionamento.	29
3.3	Cenário de simulação.	34
3.4	Nós UPs com atividade <i>On-Off</i>	36
3.5	UPs trocam de canais em 30% da simulação.	37
3.6	Tempo de convergência medido em períodos de controle.	38
3.7	Taxa de entrega com aumento do nós UPs.	39
4.1	Cenário de simulação.	50
4.2	MTTR para os 2 algoritmos com 8 RCs vizinhos.	53
4.3	MTTR - com e sem uso da proposta	54
4.4	Redução de MTTR.	55
4.5	ATTR - com e sem uso da proposta	56
4.6	Redução de ATTR.	57
4.7	Taxa de sucesso - com e sem uso da proposta	58
5.1	Diagrama de funcionamento para uma rede de rádios cognitivos.	62
5.2	Modelo de atividade nos canais.	64
5.3	Controle de acesso.	65
5.4	Taxa de Entrega do Tráfego <i>Unicast</i> sem Carga de <i>Broadcast</i>	68
5.5	Taxa de Entrega do Tráfego <i>Unicast</i> com Carga de <i>Broadcast</i>	69
5.6	Taxa de Entrega do Tráfego <i>Broadcast</i>	70

5.7	Atraso Médio do Tráfego <i>Broadcast</i>	71
5.8	Influência do Número de Canais.	72
5.9	Atraso médio de transmissões em <i>broadcast</i>	73
A.1	Ilustração de $2p_i$ “ <i>time-out</i> ” do algoritmo <i>mclock</i> que garante, que os rádios têm intervalos de tempo de pelo menos p_i <i>slots</i> com r_i fixo. . .	86

Lista de Tabelas

- 2.1 Características de diferentes protocolos baseadas em sequência. 20
- 2.2 Métricas de carga e TTR médio de diferentes algoritmos de *rendezvous*. 20

Lista de Abreviaturas

A-MOCH	<i>Asynchronous Maximum Overlapping Channel Hopping</i> , p. 13
ACK	<i>Acknowledge</i> , p. 15
ACLA	<i>Actor Critic Learning Automaton</i> , p. 26
AMRCC	<i>Adaptive Multiple Rendezvous Control Channel</i> , p. 19
ATTR	<i>Average Time-To-Rendezvous</i> , p. 52
CBR	<i>Constant Bit Rate</i> , p. 66
CCC	<i>Common Control Channel</i> , p. 7
CRN	<i>Cognitive Radio Networks</i> , p. 6
CRNs	<i>Cognitive Radio Networks</i> , p. 1
CWM	<i>Cognitive Wireless Mesh</i> , p. 12, 24
DSA	<i>Dynamic Spectrum Access</i> , p. 1
ETCH	<i>Efficient Channel Hopping for Communication Rendezvous</i> , p. 18
Jump-stay	<i>Jump-Stay Based Channel-hopping Algorithm</i> , p. 19
MAC	<i>Medium Access Control</i> , p. 10
MTTR	<i>Maximum Time-To-Rendezvous</i> , p. 14, 51
OSS	<i>Opportunistic Spectrum Sharing</i> , p. 6
RC	Rádio Cognitivo, p. 1
RL	Reinforcement Learning, p. 25
RTS	<i>Request To Send</i> , p. 14
SARSA	<i>State-Action-Reward-State-Action</i> , p. 26

SINR	<i>Signal-to-Interference-plus-Noise Ratio</i> , p. 65
SSCH	<i>Slotted Seeded Channel Hopping</i> , p. 19
SeqR	<i>Sequence-Based Rendezvous</i> , p. 18
TDMA	<i>Time Division Multiple Access</i> , p. 30
TTR	<i>Time-To-Rendezvous</i> , p. 47
TTR	<i>Time-To-Rendezvous</i> , p. 8
UP	Usuário Primário, p. 1
US	Usuário Secundário, p. 1
mclock	<i>Modular Clock</i> , p. 17

Capítulo 1

Introdução

Estudos contando com diversas medições revelaram que a maioria do espectro de frequências **licenciado**, ou seja, bandas de frequências acessadas por dispositivos que detêm licença de uso, está subutilizado ou não utilizado na maior parte do tempo [1]. Em contraste, o espectro **não licenciado**, o qual pode ser utilizado por qualquer dispositivo, apresenta-se superlotado, consequência do crescimento exponencial do número de dispositivos sem-fio [2, 3]. A discrepância na atribuição e utilização do espectro de frequências sugere uma solução onde **usuários não licenciados** tenham permissão para acessar o espectro que atualmente não está ocupado por algum **usuário licenciado** [1, 4], desde que este acesso oportunista não interfira nas transmissões dos **usuários licenciados**. Esta solução aponta para o emprego de um dispositivo, chamado Rádio Cognitivo (RC) [5], capaz de realizar acessos dinâmicos ao espectro de frequências (*Dynamic Spectrum Access* - DSA).

Um RC, conforme definido em [6], é um sistema de comunicação sem-fio que pode modificar seu comportamento, ou seja, adaptar seus próprios parâmetros operacionais de transmissão (tipo de modulação por exemplo) com base na percepção do ambiente ao seu redor [7, 8]. Em particular, o RC tem sido vislumbrado como o principal dispositivo capacitado a realizar o acesso dinâmico ao espectro de frequências, sendo assim capaz de utilizar de forma oportunista frequências que estejam ociosas.

Em uma rede de rádios cognitivos (*Cognitive Radio Networks* - CRNs), o proprietário de um canal licenciado é referido como um usuário primário (UP), enquanto os demais usuários do canal são denominados usuários secundários (USs) ou usuários cognitivos. Cada US está equipado com um ou mais rádios cognitivos, capazes de identificar de forma oportunista porções livres do espectro, isto é, canais ociosos, saltando entre eles sem causar interferência sobre alguma transmissão de um UP [2]. Por questões de simplicidade, usaremos o termo usuários secundários (USs) e rádios cognitivos (RCs) de forma indiferente.

A princípio, existem ambientes que se adequam à ideia de CRNs de forma mais

imediate, como é o caso de se explorar o uso de bandas de TV. Nestas, determinadas regiões apresentam um grande número de canais de televisão inativos/desativados para transmissões pelas emissoras locais [9], tornando mais simples para o RC a utilização destes canais livres sem a preocupação de interferência sobre algum usuário primário.

Ao contrário deste cenário, classificado como **estático**, assumimos neste trabalho um ambiente **dinâmico** [10] que compreende a entrada e saída de UPs e USs, dentro de um certo número de canais disponíveis. Assim, para o acesso oportunista a algum canal, os USs devem sensoriar previamente o canal selecionado e verificar se este está livre de transmissões de algum UP. Se um canal não está ocupado por algum UP, pode estar livre para dois ou mais USs o utilizarem e nele estabelecerem um enlace de comunicação. Como os rádios não possuem informações sobre os demais USs e a escolha de qual canal livre a se utilizar é aleatória, dois USs selecionarem simultaneamente um mesmo canal livre pode se tornar uma tarefa demorada [7].

Deste modo, o atual trabalho propõe-se a examinar como os rádios cognitivos inicialmente encontram-se uns com outros, dentro de um extenso espectro disponível, em constante mudança e com múltiplos UPs, o que é denominado problema de *rendezvous* [7].

Rendezvous, ou encontro, é o processo pelo qual dois ou mais rádios, que estão operando no mesmo canal e dentro do alcance um do outro, obtêm informação da existência um do outro, e a partir daí podem estabelecer um enlace de comunicação [2]. O processo de *rendezvous* é uma etapa fundamental e essencial no funcionamento de uma rede de rádios cognitivos, pois trata-se do primeiro passo para o início de uma comunicação entre os rádios [7]. O foco principal do problema é quando não existem controladores centralizados que dão suporte aos RCs, informando por exemplo quais canais estão livres de UPs, e quando as informações que os rádios inicialmente possuem sobre os outros rádios é nula, caracterizando o chamado *blind rendezvous*.

Uma das soluções ao problema do *rendezvous* é recorrer a uma faixa de frequência ou canal reservado chamado **canal de controle**. Este canal de controle funciona como um “local” de encontro para os rádios, facilitando o *rendezvous*. Cada rádio, conhecendo este canal, escuta e/ou envia mensagens através dele com a finalidade de informar a outros RCs sua existência. Pelo canal de controle, os RCs podem compartilhar dados de sensoriamento, conforme um sensoriamento cooperativo, ou seja, pode-se melhorar o desempenho na detecção de canais livres combinando as observações dos demais RCs, espacialmente dispersos, para uma tomada de decisão combinada mais precisa do que a partir de decisões individuais [11]. Além disso, pode-se transmitir informações de roteamento, coordenação de acesso ao espectro, etc.

Contudo, tratando-se de uma rede que possibilita DSA (*Dynamic Spectrum Access*), estabelecer/reservar um canal de controle não é a melhor escolha. Devido à disponibilidade dinâmica dos canais e seu compartilhamento com usuários primários, qualquer canal escolhido como canal de controle pode vir a ser utilizado no futuro por algum usuário licenciado, e desta forma permanecer inacessível durante longos períodos de tempo. Além disso, o uso de um único canal de controle pode dar origem ao problema de saturação do canal de controle [12–14]. A saturação do canal de controle é o problema ocasionado quando um grande número de acessos simultâneos são realizados a um canal físico que torna-se um gargalo, além de representar um ponto de falha para a rede [15].

Assim, alguns trabalhos contornam o problema de *rendezvous* sem adotar um canal de controle. Estes trabalhos empregam a técnica de *channel hopping* aliada à criação de uma ou mais sequências de saltos de canais e são chamados de **protocolos baseados em sequência de salto de canais**. As sequências, estabelecidas na forma de uma sucessão de pares do tipo (*slots*, canal), informam ao rádio qual canal deve ser utilizado a cada *slot* de tempo. Cada sequência possui um tamanho fixo em número de *slots*, na qual ao término dos *slots* retorna-se ao primeiro *slot* conforme um ciclo. Em geral, as sequências de saltos de canais criadas possuem propriedades que garantem encontros entre dois rádios quaisquer da rede.

O objetivo principal das sequências é diversificar a alocação de canais, para minimizar o impacto da atividade dos usuários primários sobre os RCs, e desta forma contornar as dificuldades de se ter apenas um canal físico como canal de controle. Isso além de promover o encontro entre os RCs em um tempo finito, visto que as sequências de salto criadas apresentam propriedades que garantem, em um tempo máximo, que dois RCs quaisquer possam se encontrar em um mesmo canal.

Apesar disso, o uso de sequência de salto de canais apresenta algumas deficiências que ao se utilizar um único canal de controle não ocorrem. Um dos maiores problemas, quando se trata de sequências de salto de canais, ao invés do uso de um único canal de controle, está relacionado à recepção de mensagens de *broadcast*. Isto é, a recepção de mensagens destinadas a qualquer componente da rede enviadas com o intuito de informar algo a todos os USs. Ao se utilizar um único canal de controle, este tipo de recepção ocorre de maneira natural, visto que os rádios escutam o mesmo canal em questão no mesmo instante, ao contrário do que ocorre com o uso de sequência de salto de canais, onde os encontros entre os rádios tendem a ser distribuídos pelos *slots* da sequência. Deste modo, este tipo de solução carece de um meio no qual mensagens de controle ou mensagens sobre a comunicação entre rádios vizinhos sejam trocadas.

A falta de informação relacionada à comunicação entre rádios vizinhos, ou um meio que a possibilite, pode ser considerada a maior deficiência dos tipos de soluções

que aplicam sequências de salto de canais [16]. Esta deficiência pode originar o problema de terminais escondidos, chamado de terminais escondidos multicanais (*multi-channel hidden terminal problem*) ou problema de “surdez”, devido ao uso de diferentes canais os rádios por não detectarem a presença e não terem informação um do outro, acabam transmitindo e causando colisões. O problema de terminais escondidos multicanais pode causar interferências prejudiciais às transmissões atuais, e sua principal causa é a falta de informações sobre o estado atual da rede [16].

Este problema surge pois os USs operam em diferentes canais de forma independente, e desta forma dois USs podem realizar transmissões em um canal, enquanto no mesmo instante outros dois USs transmitem em um outro canal. Devido a questões do próprio uso da sequência de salto de canais, pois envolve o chaveamento de canais, podem ocorrer situações em que um rádio não consegue detectar transmissões de outro rádio e vice-versa. Esta impossibilidade na detecção, aliada ao conhecimento das transmissões ocorridas apenas no atual canal de operação e não das transmissões em outros canais disponíveis, pode levar a colisão entre as transmissões dos rádios. Neste atual trabalho, apesar de citarmos este problema não abordamos proposta para o caso de terminais escondidos multicanais.

Ou seja, as duas formas de solução ao problema do *rendezvous*, seja adotando um canal de controle ou usando uma sequência de salto de canais, apresentam deficiências. Assim, o objetivo deste trabalho é estudar e propor soluções a alguns problemas citados.

Primeiramente, exploramos a ideia do uso do canal de controle com foco na dificuldade de se conseguir um único canal como canal de controle para todos os USs componentes da rede. Assim, partimos da ideia de canal de controle, porém voltada a uma organização dos rádios em grupos. A partir do grupo, um rádio escolhido como líder elege mais de um canal que pode ser usado como canal de controle. Nesta parte, utilizamos um “canal de controle lógico” dentro de cada grupo de rádios ao invés de um único canal de controle físico. Desta forma, mesclamos o uso da abordagem de canal de controle com a ideia de sequência de salto de canais. Através dos resultados, pode-se constatar que o uso da sequência de salto de canais em conjunto com o grupo de rádios forneceu um desempenho satisfatório para a taxa de entrega de mensagens de controle, mesmo em cenários com maior atividade de UPs.

Na segunda parte, focamos nas soluções que trabalham com sequência de salto de canais, mais especificamente abordamos o problema de envio de mensagens de *broadcast* em soluções que implementam a sequência de salto de canais. Para este tipo de transmissão, grande parte dos trabalhos adota a replicação da mensagem a cada rádio encontrado, até que todos os rádios próximos recebam a mesma mensagem em instantes de tempo diferentes. Diferentemente desses trabalhos, propomos um

mecanismo que constrói uma sequência para o envio de mensagens de *broadcast* a partir das sequências de salto de canais dos vizinhos do rádio emissor. Os resultados das simulações mostraram que o mecanismo proporcionou ganhos de desempenho em termos da taxa de sucesso, ou seja, recebimento das mensagens em *broadcast*, assim como melhora no tempo médio de recebimento.

Por fim, propomos e avaliamos um conjunto de regras de funcionamento, conforme um protocolo de acesso ao meio em uma rede de rádios cognitivos (CRNs). Na literatura, trabalhos que tratam de *rendezvous*, principalmente os que empregam sequência de salto de canais, estão mais relacionados ao primeiro encontro entre os rádios, e assim pouco comentam sobre o funcionamento da rede após o primeiro encontro. Desta maneira, definimos um conjunto de regras, partindo da ideia de uso de sequência de salto de canais, a qual será empregada não somente para o primeiro encontro mas também para todo o funcionamento da rede. Estas regras definem as ações dos rádios em cada estado de operação, seja em modo de transmissão de mensagens de *broadcast*, *unicast* ou em modo de recepção (*idle*). Verificamos pelos resultados que o conjunto de regras que estabeleceu o uso de diferentes sequências de salto de canais sob demanda aliviou o problema de acesso ao meio compartilhado, mostrando-se uma boa estratégia.

1.1 Organização do Texto

No Capítulo 2, os principais conceitos e propostas ao problema de *rendezvous* serão apresentados. Logo após, no Capítulo 3, apresentamos uma proposta que emprega o uso de canal de controle em um ambiente onde os rádios são organizados em grupos. No capítulo seguinte, será apresentada a proposta para transmissões em *broadcast* para soluções que utilizam sequência de salto de canais, seguida da análise dos resultados obtidos a partir de simulações. O Capítulo 5, apresenta a proposta das regras de funcionamento para uma rede de rádios cognitivos, bem como a comparação entre o uso das regras propostas e o caso de se utilizar apenas um canal de controle. Por fim, o Capítulo 6 apresenta as conclusões finais, destacando as principais contribuições da tese e apresentando possíveis trabalhos futuros.

Capítulo 2

Rendezvous em Rádios Cognitivos

Os Rádios Cognitivos (RCs) têm sido reconhecidos como uma tecnologia promissora ao problema de subutilização do espectro. Sua premissa apoia-se no compartilhamento oportunista do espectro (*Opportunistic Spectrum Sharing* - OSS) , onde usuários não licenciados (usuário secundários - USs) devem se abster de transmitir em canais onde sinais de usuários licenciados (usuário primários - UPs) sejam detectados [17].

Logo, em uma rede de rádios cognitivos (*Cognitive Radio Networks* - CRN), a disponibilidade de canais é determinada pela atividade de usuários primários, que muda dinamicamente em frequência (canal), espaço e tempo. Como consequência, o conjunto de canais disponíveis para cada usuário secundário também se altera dinamicamente [16]. Desta forma, em um determinado instante, USs podem operar em diferentes canais independentemente, conforme Figura 2.1 [16].

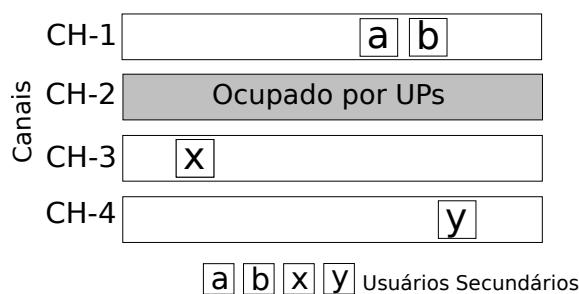


Figura 2.1: USs podem operar em diferentes canais independentemente.

Antes da transmissão de dados começar, dois USs que desejam se comunicar devem se encontrar em algum canal comum, livre para ambos, e estabelecer um enlace de comunicação [18], o que é chamado de *rendezvous*.

No entanto, o *rendezvous* em uma rede de rádios cognitivos nem sempre é trivial. O modelo de acesso oportunista define que cada rádio selecione um canal, dos que estão momentaneamente livres de UPs, para utilizar. Como essa escolha é individual,

cada rádio pode selecionar um canal diferente para uso. Além disso, os rádios podem nem sequer estarem cientes da presença uns dos outros antes do primeiro encontro, ou seja, a vizinhança dos rádios pode ser desconhecida.

Outro fato que dificulta o *rendezvous* está relacionado ao conjunto de canais disponíveis para o rádio. O conjunto de canais disponíveis para cada rádio, também chamado de “**visão do espectro**”, pode se alterar dinamicamente de acordo com a atividade dos UPs e seu alcance. Desta forma, o conjunto de canais disponíveis para um rádio em um determinado instante pode ser diferente do de outro rádio cognitivo, caracterizando uma “**visão heterogênea do espectro**” [2].

Essa heterogeneidade na disponibilidade de canais pode ser causada pela posição relativa entre RCs e usuários primários [7, 18], por variações espaço-temporais na disponibilidade do espectro, bem como pelas condições de interferência e/ou bloqueio [19]. Em geral, se o conjunto de canais disponíveis é o mesmo para todos os usuários, ou seja, mesma “**visão do espectro**”, estamos em um cenário chamado **simétrico**. Em cenários **assimétricos**, um conjunto diferente de canais pode estar disponível para diferentes usuários secundários.

O grau de atividade dos usuários primários também dificulta o *rendezvous*. Como primordialmente, os RCs devem evitar causar interferência sobre os usuários primários, cenários onde os usuários primários fazem uso dinâmico do espectro, e os RCs são equipados com apenas uma interface de rádio, forçam estes últimos a constantemente mudar seu canal de operação na procura por novas oportunidades de canais livres. A estes cenários, onde temos a entrada e saída de rádios e a atividade de usuários primários é intensa, denominamos de cenários **dinâmicos**.

Em contrapartida, nos cenários **estáticos**, o uso de canais por usuários licenciados praticamente não se altera no tempo. Como exemplo de cenário estático, podemos citar o uso de frequência de canais de TV [9], onde a utilização do recurso pelo usuário licenciado, neste caso a emissora de televisão, é bem controlada e não se altera em curtos períodos de tempo.

Assim, o grande desafio é proporcionar o *rendezvous* em cenários **assimétricos** e **dinâmicos**, pois em cenários simétricos todos os usuários podem simplesmente permanecer no canal de menor índice para atingir o *rendezvous* [20]. Chamamos de índices a identificação que relaciona uma frequência física a um canal e, neste caso, a identificação utilizada por cada rádio cognitivo é a mesma [7].

Atualmente, existem duas abordagens bem conhecidas para possibilitar o *rendezvous* em uma rede de RCs: **1)** Usando um canal dedicado chamado **Canal de Controle Comum** (*Common Control Channel - CCC*), e **2)** Usando a técnica de salto de canais (*channel hopping*) aliada ao uso de uma sequência de salto de canais [16], chamados de **Protocolos Baseados em Sequência de Salto de Canais**.

O Canal de Controle Comum funciona como um “local” de encontro para os RCs facilitando o *rendezvous*. Desta forma, os rádios primeiro se encontram neste canal e, a partir dele, agendam qual será o canal de transmissão de dados entre o par de RCs. Além disso, os rádios transmitem através deste canal informações para melhorar a eficiência no uso do espectro, tais como: compartilhamento de dados de sensoriamento, difusão de informações de roteamento, coordenação do acesso ao espectro, etc [21].

Nesta abordagem, conforme Figura 2.2, o período de funcionamento está dividido em dois intervalos: intervalo de negociação, ou de controle, e intervalo de dados (*data*). Quando um US quer iniciar a comunicação, ele primeiro utiliza o CCC durante o intervalo de controle, e tenta negociar com o vizinho (receptor) pretendido. Depois de negociar através do CCC, a transmissão de dados poderá se realizar durante o intervalo de dados através de outro canal disponível, conhecido como canal de dados. Assim, esta abordagem garante que todos os USs possam se encontrar em um CCC durante o intervalo de controle [16].

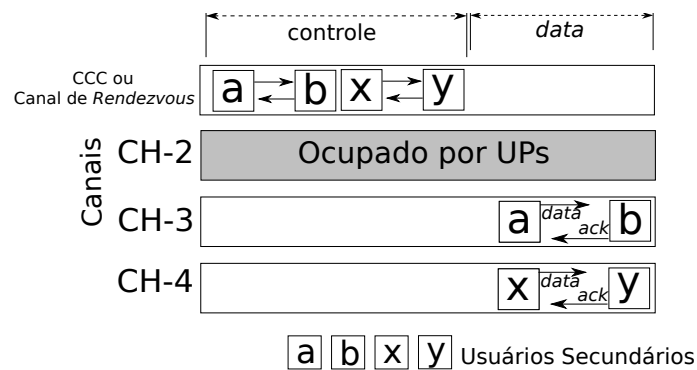


Figura 2.2: Procedimento básico de *rendezvous* - canal para negociação (CCC) e envio de dados usando um canal de dados.

O outro grupo de soluções, chamado protocolos baseados em sequência, se aproveita da própria forma de funcionamento dos RCs, ou seja, no salto de canais para proporcionar o *rendezvous*. Cada solução define uma sequência de salto de canais que almeja o *rendezvous* em um tempo finito. A diferença entre as soluções se dá na quantidade de informações necessárias como parâmetros de entrada para a determinação da sequência de saltos, ao tempo para o encontro (*Time-To-Rendezvous* - TTR), ao tamanho da sequência, o número de sequências geradas, entre outras. Conforme Figura 2.3 [16], podemos verificar duas sequências de salto de canais entre os Rádios A e B. Pode-se constatar que o TTR para estas duas sequências foi de 6 *slots* e que o *rendezvous* ocorreu no canal 4, o qual pode ser chamado de canal de *rendezvous*.

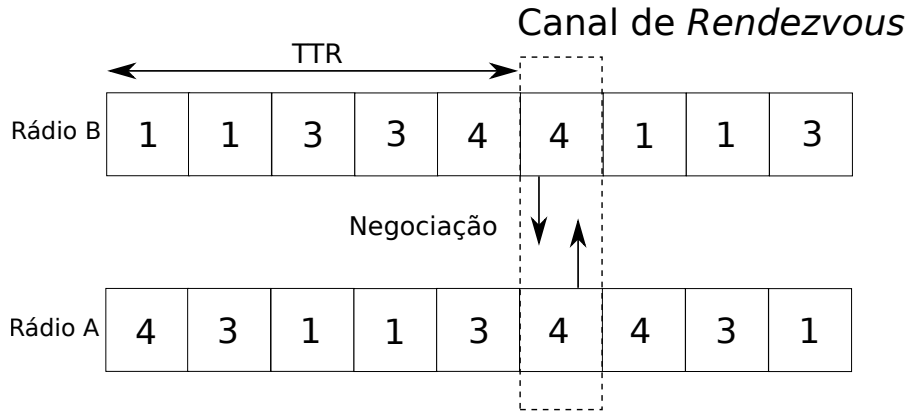


Figura 2.3: Exemplo de sequência de salto de canais entre os Rádios A e B.

2.1 Classificação de Soluções para o Problema de *Rendezvous*

Muitos trabalhos foram publicados a respeito de *rendezvous* em rádios cognitivos. Dentre eles, [7, 21] destacam-se por apresentar diversas soluções da literatura e levantar problemas existentes. Além disso, ambos propõem formas de classificação para as soluções. Adotaremos no trabalho atual o tipo de classificação apresentado em [7].

Na Figura 2.4, conforme [7], os trabalhos se dividem em dois ramos: **Sistemas de *Rendezvous* Assistidos** e **Sistemas de *Rendezvous* Não Assistidos**. Sob um sistema **Assistido**, um controlador centralizado direciona os rádios cognitivos para os canais disponíveis. Este servidor opera sobre um bem conhecido e dedicado canal de controle.

Embora este sistema seja simples de ser implementado e proveja maior controle, não é escalável, flexível ou robusto. Se um único canal de controle é utilizado, ele pode se tornar um gargalo para o sistema e, caso novos rádios ingressem ao sistema, a capacidade de coordená-los pode se tornar difícil. Finalmente, ter uma arquitetura centralizada pode criar um ponto único de falha [7].

No outro ramo, ***Rendezvous* Não Assistido**, os próprios rádios são encarregados de encontrar um canal comum. A partir deste ramo, existem três amplas categorias de técnicas: os rádios podem usar um **Único Canal de Controle**, **Múltiplos Canais de Controle**, ou **Nenhum Canal de Controle**.

Sob um esquema de **Canal de Controle Único**, os rádios que procuram encontro utilizam um único e bem conhecido canal de controle. Embora isto dispense a necessidade de um controlador centralizado, ainda incorre em sobrecarga (*overhead*) e pode agir como um único ponto de falha [7].

Sob **Múltiplos Canais de Controle**, os rádios devem primeiro se encontrar em

um dos canais de controle para que se inicie a negociação de qual canal realmente será utilizado como canal de controle. Nesta classificação, os RCs podem formar grupos (*clusters*), reduzindo a necessidade de se encontrar um único canal livre para todos os rádios da rede. No entanto, ainda há sobrecarga na criação e manutenção dos grupos e, na localização e identificação dos possíveis canais de controle [7].

Se **Nenhum Canal de Controle** está disponível, os rádios devem descobrir uma maneira de se encontrar com os demais rádios por conta própria, o que é caracterizado como *blind rendezvous* [7].

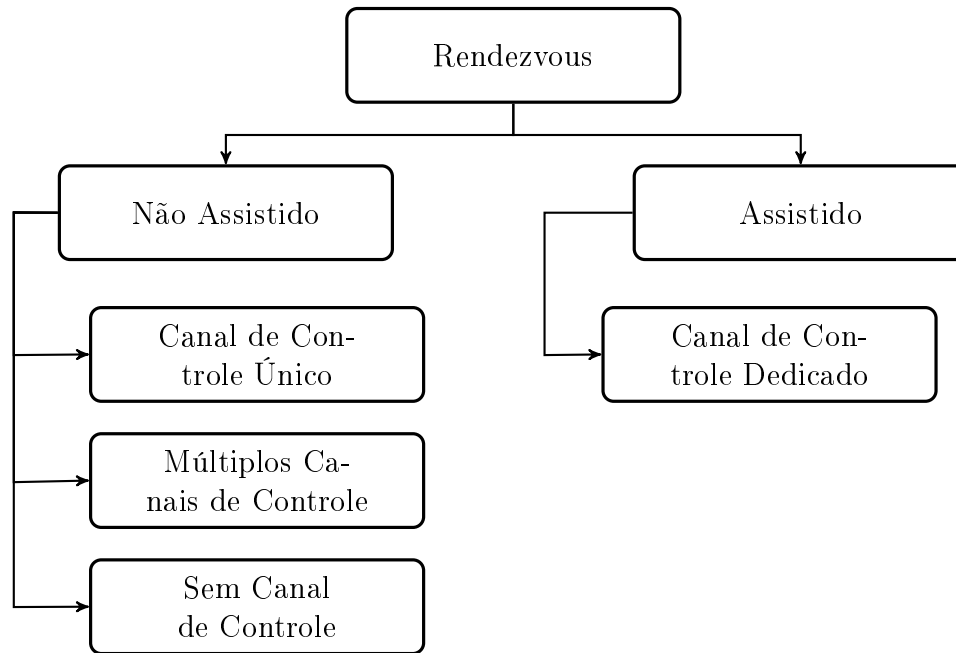


Figura 2.4: Classificação de soluções para o problema de *rendezvous*.

Canais de controle também podem ser adotados em redes sem infraestrutura, tais como redes móveis *ad hoc*. Desta forma, grande parte dos trabalhos em protocolos de controle de acesso ao meio (*Medium Access Control* - MAC) para redes com multicanais assume que cada nó (componente da rede) é equipado com dois rádios [7, 22]. A ideia é que um rádio (ou interface) monitore constantemente um canal de controle dedicado e utilize esse canal para reservar as transmissões em um dos vários canais de dados em potencial. Após a indicação da reserva de algum canal através do canal de controle, o segundo rádio sintoniza para este canal de dados reservado e depois ocorre a troca de dados [7].

Embora o uso de um único canal de controle simplifique o processo de encontro e pareça razoável mesmo em uma rede sem infraestrutura, existe uma preocupação quanto à potencial contenção do canal de controle sob alta carga de acessos, caso vários rádios, todos próximos uns dos outros, desejem negociar o uso do espectro

disponível e compartilhado.

Outro ponto importante é a existência de políticas que garantam que nenhum rádio tenha vantagem, sobre os demais, de acessar o canal de controle. Por esta razão, a proposta do IEEE 802.22 é não adotar um Canal de Controle Comum, e sim selecionar um canal para este uso dentre os canais livres disponíveis [7, 23].

Nos casos de **Rendezvous Não Assistido** e **Sem Canal de Controle**, caracterizados como *blind rendezvous*, todos os canais vagos estão potencialmente disponíveis para a troca de dados e controle. Os rádios são responsáveis por verificar quais canais estão livres e, em seguida, tentar estabelecer um enlace de comunicação em um desses canais. Por estas razões, soluções para *blind rendezvous* permitem que os rádios cognitivos possam se tornar verdadeiramente autônomos, não contando com infraestrutura ou disponibilidade exclusiva de algum canal. Resolver efetivamente o problema de *blind rendezvous* requer a capacidade de garantir um encontro em um período de tempo razoável, sob uma variedade de condições e requisitos do sistema [7].

Neste trabalho, daremos mais atenção às soluções do tipo **Não Assistido** com **Múltiplos Canais de Controle** e **Sem Canal de Controle**, desta forma apresentaremos a seguir alguns trabalhos da literatura dentro destes grupos.

2.2 Múltiplos Canais de Controle

Normalmente, trabalhos nesta classificação recorrem à formação de grupos para selecionar o canal de controle. Estes trabalhos seguem a ideia que, independentemente da distribuição do uso de frequências (canais) entre os rádios de uma rede de RCs, tem-se uma maior chance de se ter um canal comumente disponível para RCs que estejam próximos do que um canal disponível para toda a rede [21]. Os RCs podem formar grupos através de uma coordenação entre rádios vizinhos, que com base nas informações coletadas, elegem o melhor canal para ser selecionado como canal de controle.

Um dos desafios nestes trabalhos vêm da comunicação intergrupos, que requer a troca de mensagens entre grupos de rádios vizinhos com diferentes canais de controle. Diferentes técnicas foram desenvolvidas para a atribuição do canal de controle, que podem ser divididas em duas categorias: coordenação entre vizinhos e uso de agrupamento (*clustering*) [24, 25].

Clustering é uma técnica comum em redes sem-fio distribuídas, onde os rádios são divididos em grupos com base em algoritmos de formação. Um membro do grupo é eleito como líder (*clusterhead*), e atua como entidade central para a coordenação do grupo. Como resultado, o rádio líder seleciona um canal geralmente disponível para todos os membros do grupo como canal de controle. Os trabalhos nesta classificação

se diferenciam quanto à formação dos grupos e como é alocado o canal de controle. Uma característica é que a atividade de UPs, devido à interferência, pode forçar a mudança de canal de controle. Isto afeta diretamente a manutenção e reconfiguração dos grupos. Assim, a eficiência na formação e sua reconfiguração apresenta um grande impacto sobre o estabelecimento do canal de controle [21].

Os autores em [24] propõem uma estrutura de grupo entre nós vizinhos de um salto com CCC selecionado pelo nó líder de cada grupo. Os grupos são formados pelo processo de descoberta de vizinhos, através da varredura de canais e *broadcast* de *beacons* (sondas). O líder é quem inicia o envio de sondas para formar o *cluster* e também quem decide qual será o CCC, porém nem sempre o agrupamento e a escolha do CCC é ótima. Assim, para minimizar o número de agrupamentos e CCCs correspondentes, um algoritmo de otimização de *clusters*, com base no problema de encontrar um conjunto mínimo dominante foi proposto em [26]. No entanto, este processo de reconfiguração incorre em grande sobrecarga (*overhead*) de controle entre os grupos. Além disso, o grau de eficiência em reagir à seleção de um novo canal de controle quando o atual passa a ser utilizado por UPs depende dos algoritmos de formação de grupos e dos algoritmos de seleção do canal de controle.

Em redes do tipo Malha Cognitiva (*Cognitive Wireless Mesh - CWM*) [27], o agrupamento dos rádios em *clusters* facilita o gerenciamento e aumenta a probabilidade de que exista um canal livre para controle. Este tipo de rede tem por definição o fato de ser autogerenciável, logo o acesso dinâmico ao espectro é possibilitado pela troca de informações através do canal de controle. Consequentemente, mensagens perdidas no canal de controle podem prejudicar severamente a rede, pois um US, ao perder alguma mensagem de controle, pode ficar sem a indicação de qual canal será utilizado em alguma comunicação futura.

Alguns trabalhos selecionam o canal de controle depois de formados os *clusters*, enquanto outros formam grupos de acordo com a disponibilidade de canais em comum. Esta última abordagem torna-se mais robusta à atividade de UPs devido à sua capacidade de reagrupamento.

Uma dificuldade relacionada a esta categoria de soluções é a comunicação inter-grupos. Assim, foi proposta em [28] uma forma de seleção de canais para controle utilizando um esquema de envio e recebimento de tons. Este procedimento de tons permitiu que rádios, comuns a dois grupos, conseguissem se comunicar e trocar mensagens de controle.

Outro ponto é que trabalhos que tratam deste problema, em geral, adotam cenários cujo alcance dos usuários primários se estende sobre todos os usuários secundários, ou seja, a “**visão da espectro**” é a mesma para todos os USs com relação à detecção de UPs. Isto nem sempre é verdade em decorrência de uma série de fatores, como por exemplo erros de detecção e alcance dos usuários primários.

Um esquema de coordenação distribuída foi proposto em [29], onde formam-se grupos de acordo com os diferentes canais livres para os rádios da rede. Após a descoberta de vizinhança inicial, cada RC conhece quais os canais disponíveis de seus vizinhos e assim seleciona um canal para controle a partir de um sistema de votos. Cada RC vota em um canal, geralmente disponível para o maior número de vizinhos. As informações de voto são trocadas através de transmissões em *broadcast*. Além disso, utiliza-se, no início de cada quadro MAC, uma janela de controle dedicada, através do qual RCs ligados a grupos diferentes podem enviar e receber mensagens de controle de grupos distintos.

Existem outros trabalhos que recorrem à inteligência de enxame para a atribuição do CCC. O trabalho [30] se destaca por ser o primeiro a aplicar um algoritmo baseado em inteligência de enxame, em que uma maior eficiência é alcançada se os trabalhadores (*workers*) especializados executarem tarefas em paralelo. Em resposta à detecção de atividade de UPs, o melhor canal escolhido pela maioria de RCs é utilizado como o CCC. Além disso, a cobertura é melhorada através da redução do número de opções de canais que podem ser de controle, diminuindo assim o atraso na escolha de um novo canal de controle, caso no atual seja detectado algum tráfego de mensagens de UPs. Porém a manutenção da proposta incorre em elevada sobrecarga, especialmente se a atividade de UPs for dinâmica.

2.3 Sem Canal de controle

Nesta abordagem, o problema do *rendezvous* em rádios cognitivos é tratado utilizando a técnica de salto de canais (*channel hopping*) guiada pelo uso de uma sequência. Estas soluções são conhecidas também por **protocolos baseados em sequência de salto de canais** e seus objetivos são eliminar a necessidade de um CCC, como também transpor alguns inconvenientes vindos do uso de um canal de controle.

Os trabalhos da literatura se diferenciam pelo emprego de diferentes estratégias na construção das sequências de saltos [2, 7, 15, 31–34], assim como pelo número de sequências criadas e as informações necessárias para sua construção. Dependendo da solução podem-se ter n sequências e cada rádio utiliza apenas uma delas ou seleciona uma e depois de um período de uso troca aleatoriamente para outra sequência. Em algumas soluções pode-se ter apenas uma única sequência utilizada por todos os rádios. Existem casos em que tem-se mais que uma sequência para cada rádio, como por exemplo o A-MOCH (*Asynchronous Maximum Overlapping Channel Hopping*) [15, 17], que dependendo do estado, transmitindo ou *idle*, o rádio seleciona a sequência mais apropriada. Além disso, por propriedade, as sequências de salto de canais garantem um tempo máximo para o encontro entre dois rádios, estimado em

número de *slots*, denominado *Maximum Time-To-Rendezvous* (MTTR).

Outra característica que diferencia as soluções está relacionada à quantidade mínima do número de sobreposições entre duas sequências quaisquer garantida pela solução. Existem trabalhos onde garante-se apenas uma sobreposição em um canal enquanto outros garantem que o número mínimo de sobreposições é igual ao número de canais disponíveis, ou seja, no mínimo duas sequências terão sobreposições em cada canal pelo menos uma vez desde que todas as sequências sejam percorridas. Como forma geral, cada sequência de salto de canais deve ser capaz de apresentar sobreposição com todas as outras sequências periodicamente, o que neste caso representa que cada sequência deve ser capaz de permitir *rendezvous* com as demais sequências criadas. Além disto, devido às propriedades das redes DSA em que a disponibilidade dos canais é dinâmica, qualquer sequência deve conter todos os canais que o RC possa utilizar [35].

Soluções baseadas em sequência de salto de canais podem ser categorizadas como **síncronas** [15, 33] ou **assíncronas** [2, 7, 32], conforme os RCs iniciam suas sequências de saltos. Quando o instante de início das sequências é comum a todos os rádios, o algoritmo é classificado como **síncrono**. A seguir, na Subseção 2.3.1 exemplificamos a diferença entre estas duas categorias.

Como característica geral, as soluções existentes indicam que o tempo é dividido em *slots*, desta forma a sequência de canais, determinada na forma de um par (*slot*, canal), define qual canal deve-se utilizar na duração de cada *slot*. Quando dois RCs saltam para o mesmo canal, e estão no alcance um do outro, no mesmo instante de tempo, ocorre o *rendezvous*. Salvo casos onde as condições anteriores são satisfeitas, porém o canal onde ocorreria o *rendezvous* já está em uso por algum usuário primário. Os autores em [7] identificam cinco modelos diferentes em que o *rendezvous* pode ocorrer, de acordo com as capacidades do sistema, políticas de uso do espectro e condições do meio.

De forma mais específica, quando dois USs encontram-se em um canal comum, eles precisam trocar informações de controle necessárias para indicar o encontro. O procedimento básico, da descoberta ou encontro de um rádio vizinho, pode ser descrito da seguinte forma: quando o US muda para um canal disponível, ele verifica (escuta) o canal, com a finalidade de perceber a presença de UPs e/ou outras transmissões de USs. Se o canal apresenta-se livre, o US transmite um *beacon* em *broadcast*. O *beacon* é uma mensagem de sonda, uma espécie de sinalizador, o qual pode ser uma mensagem curta ou apenas um sinal que indique a presença de um US. Diferentes tipos de sinalizadores podem ser usados, tal como um *beacon* em [7], RTS (*Request To Send*) em [36], e um pedido de conexão [37]. Independente do sinalizador, o propósito de transmiti-lo é revelar a presença de um US que está disposto a se comunicar. Deste modo, se um receptor no canal atual recebe este sinalizador,

ele pode responder com uma mensagem de confirmação (*Acknowledge* - ACK) indicando ao emissor do sinalizador que existe um US presente no canal. O *rendezvous* é realizado e finalizado uma vez que estes dois usuários tenham trocado estes pacotes de controle (neste exemplo, *beacon* e ACK). Caso contrário, ou seja o emissor do *beacon* não receba nenhuma resposta, este irá saltar para o próximo canal, seguindo a sequência de salto de canais, e transmitir a mensagem de *beacon* novamente até que receba alguma resposta. Após o *rendezvous*, os USs podem negociar e selecionar qual canal de dados os dois passarão a utilizar [16] em transmissões futuras.

A Figura 2.3 ilustra a operação de um protocolo baseado em sequência. Os USs A e B, após operarem conforme a sequência de salto de canais própria, podem realizar o *rendezvous* quando ambos estão sintonizados no canal 4 no mesmo instante, e assim os dois passam a ter informações da existência um do outro.

Vale ressaltar que para o primeiro passo do processo, ou seja, verificar se o meio esta livre de UPs, diversas técnicas podem ser empregadas. Por exemplo, detecção de energia, cicloestacionariedade entre outras, conforme em [38]. Admitimos neste trabalho que os USs conseguem diferenciar transmissões de USs e UPs em decorrência de algum indicativo, tal como um preâmbulo característico, presente apenas no início das transmissões dos USs.

A prática de uso de uma sequência de salto de canais minimiza o impacto da atividade de UPs sobre os USs. Como os rádios não se tornam reféns de um único canal, por exemplo CCC, a variação na alocação dos canais ao longo do tempo torna o RC mais robusto à atividade de UPs. Além disso, essa abordagem permite o reuso espacial de canais, pois possibilita a comunicação simultânea de RCs na mesma vizinhança através da atribuição de sequências que sejam ortogonais entre si, mesmo que esta ortogonalidade não seja total entre as sequências envolvidas.

2.3.1 Sincronismo de Início de Sequências

Conforme comentado, os trabalhos relacionados a sequências de salto de canais podem ser classificados como sistemas **síncronos** ou **assíncronos** quando comparados à necessidade de sincronismo de início das sequências.

Trabalhos classificados como **síncronos** criam sequências que são válidas apenas quando todos os rádios iniciam suas respectivas sequências no mesmo instante. Em sistemas **assíncronos**, as sequências garantem encontro com os demais USs mesmo que o início das sequências seja qualquer. Ou seja, os rádios podem ingressar na rede em instantes de tempo diferentes que mesmo assim as sequências criadas garantem um encontro em tempo finito. A Figura 2.5 ilustra uma solução que necessita de um sincronismo de início de sequências. Neste exemplo, apenas se os rádios estiverem com o início de sequências sincronizado o *rendezvous* será garantido. Caso o início

das sequências seja defasado de um *slot*, por exemplo o nó B ingressa na rede no *slot* t_1 , o *rendezvous* entre os dois rádios nunca ocorrerá.

Podemos notar que uma coordenação entre USs é difícil de se conseguir antes do primeiro *rendezvous*, assim sistemas síncronos assumem a existência de uma coordenação prévia entre os rádios, ou seja, trabalhos nesta classificação assumem um sincronismo de relógio global entre todos os rádios, o que pode ser um tarefa custosa.



Figura 2.5: Sistemas síncronos - sincronismo de início de sequências.

2.3.2 Sincronismo de *Slot*

Além do sincronismo de início de sequências, podemos pensar na necessidade de termos um sincronismo de *slots* entre as sequências, representado na Figura 2.6 [39]. Nesta figura, S_i e U_i desde $i = 0$ até $i = T + 1$, representam duas sequências, no caso sequência S e U , enquanto os subíndices informam o *slot*.

Sem um sincronismo de *slot*, os limites de *slot* entre duas sequências dos rádios A e B, por exemplo, podem apresentar um desvio, representado por θ na Figura 2.6.

Se considerarmos t como o tempo mínimo necessário para se estabelecer um enlace de comunicação entre dois rádios, o problema do sincronismo de *slot* pode ser resolvido aumentando o tempo de *slot* para $t' = 2t$ [39].

Na Figura 2.6, se a medida de desalinhamento entre os *slots* dos nós A e B for representada por $\delta = nt' + \theta$ ($0 \leq \theta < t'$), dois casos podem ser considerados:

- **Caso 1)** Para $0 \leq \theta < t$, cada *slot*, a contar pelo início do *slot* de B, se sobrepõe ao correspondente de A durante $(t' - \theta)$, que é tempo suficiente para estabelecer um enlace de comunicação, se eles estiverem em um canal em comum.

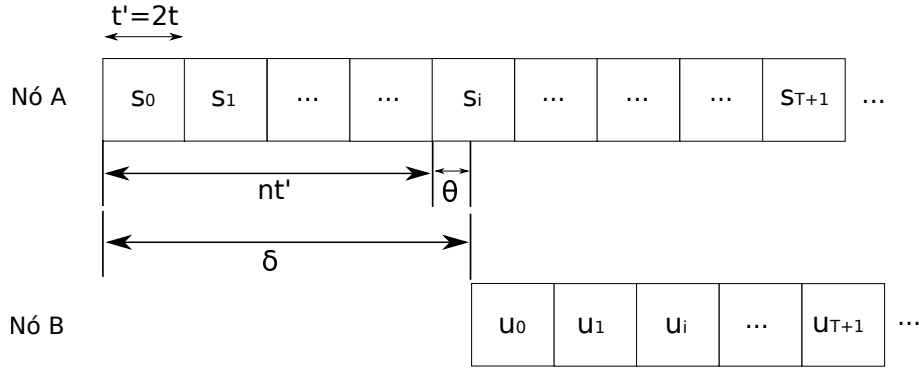


Figura 2.6: Exemplo de duas sequências sem sincronismo de *slots*.

- **Caso 2)** Para $t \leq \theta < t'$, cada *slot*, a contar pelo início do *slot* de B, se sobrepõe ao próximo *slot* de A durante θ .

Em ambos os casos, o tempo de duração de um *slot* dobrado para $2t$ assegura que os dois rádios permanecerão por pelo menos o tempo t suficiente para o *rendezvous*.

2.3.3 Exemplos de Soluções para *Rendezvous* que Usam Sequência de Salto de Canais

Um exemplo de algoritmo de *rendezvous* assíncrono é o *Modular Clock (mclock)* [7]. Este algoritmo usa aritmética modular e números primos para garantir um tempo máximo de *rendezvous* (MTTR) que seja satisfatório. O algoritmo começa pela seleção de um canal de forma aleatória, a partir da lista de canais disponíveis. Após cada intervalo de tempo (*slot*), o RC salta $r \bmod p$ canais percorrendo esta lista, onde p é o menor número primo maior que o número de canais e r é um valor escolhido aleatoriamente dentro do intervalo $[0, p)$. Após $2 \times p$ *slots*, um novo valor de r é escolhido aleatoriamente [40]. Obrigatoriamente, o número de canais deve ser um número primo para que a propriedade de tempo de encontro da solução seja garantida. Ao fim do trabalho, após a conclusão, temos um Anexo, onde segue a dedução matemática do algoritmo *mclock*.

Em [15], é apresentado um esquema baseado em sistema de *quoruns* para construir a sequência de salto de canais. O objetivo é aumentar o número de sobreposições entre as múltiplas sequências. Um *quorum* é definido como um elemento de um sistema S que satisfaz a propriedade de interseção: $p \cap q \neq \emptyset, \forall p, q \in S$ [21]. Por exemplo, a partir de $Z = \{0, 1, 2\}$ pode-se formar o conjunto de *quoruns* $S = \{\{0, 1\}, \{0, 2\}, \{1, 2\}\}$ e cada elemento de S dá origem a uma sequência, conforme a Figura 2.7. Assim temos u, v e w que são sequências que apresentam sobreposição de canais entre elas e cada sequência pode ser atribuída a um rádio como uma sequência de salto de canais. Na figura, os campos em branco repre-

sentam casos em que o canal será atribuído aleatoriamente. Este sistema funciona adequadamente desde que exista um sincronismo global de início de sequência entre os rádios.

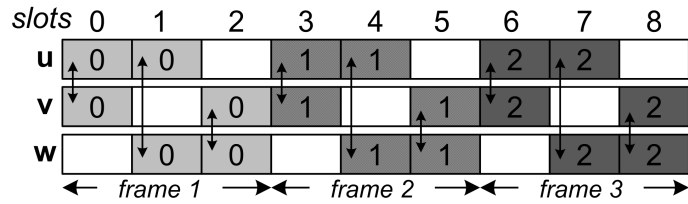


Figura 2.7: Exemplo de sequências a partir do conjunto de *quoruns*.

Outro algoritmo é o ETCH (*Efficient Channel Hopping for Communication Rendezvous*)[35]. Este algoritmo consiste de três partes: escalonamento, atribuição de canais e execução da sequência. Para conseguir uma melhor média de TTR, o ETCH constrói um conjunto de $2N$ sequências, contendo $2N - 1$ slots, onde N é o número de canais, de forma que cada sequência apresente sobreposição de canais com as demais em diferentes slots. Para isso, as sequências são construídas em dois passos: no primeiro passo, gera-se $2N - 1$ escalonamentos e, no segundo passo, é feita a atribuição de canais. Ao fim do processo, todos os rádios dispõem de um conjunto de sequências iguais. Na execução, primeiramente, o rádio seleciona aleatoriamente uma sequência que irá seguir, depois de passar por todos os slots desta sequência, ele realiza uma seleção aleatória entre as sequências restantes, e passa a utilizar esta sequência recém escolhida [35]. Nesta solução, o conjunto de sequências gerado é o mesmo para todos os rádios da rede. A razão para o rádio re-selecionar uma sequência de salto de canais, quando termina de utilizar alguma, é para se certificar de que qualquer par de rádios possa ser capaz de se encontrar em diferentes canais.

Existe também um método chamado SYN-MAC [21] onde o padrão de salto de canais segue uma sequência em *round robin*. Todos os RCs são sincronizados e chaveiam para um canal em um tempo pré-definido. A cada período, um novo canal é selecionado conforme uma sequência de canais. Caso o canal selecionado esteja em uso por algum UP, aguarda-se a próxima oportunidade e seleciona-se o próximo canal.

Conforme o algoritmo SeqR (*Sequence-Based Rendezvous*) [32], constroem-se as sequências de salto de canais da seguinte maneira. Primeiro, seleciona-se uma permutação de um conjunto de canais, que possui n elementos. Após, esta permutação é repetida N vezes e intercalada com seus próprios elementos. Por exemplo, se a permutação escolhida para $N = 3$ for $\{2, 0, 1\}$, a sequência inicial será o primeiro elemento da permutação, em seguida, o conjunto de permutação seguido do segundo elemento, seguindo-se da permutação outra vez, e assim por diante. Assim, a sequên-

cia resultante para este exemplo será $\{\underline{2}, 2, 0, 1, \underline{0}, 2, 0, 1, \underline{1}, 2, 0, 1\}$. Cada sequência gerada pelo SeqR possui período igual a $N \times (N + 1)$ slots. Esta solução, apesar de assíncrona, pode apresentar tempos de TTR grandes devido a seu próprio período, que com o aumento do número de canais disponíveis cresce bastante.

No algoritmo SSCH (*Slotted Seeded Channel Hopping*) [33], cada rádio utiliza um ou múltiplos pares de (canal, *seed*) para determinar suas sequências de saltos. No par, “canal” representa o canal inicial da sequência e “*seed*” é a taxa na qual o RC salta pelos canais da sequência. São permitidas $N - 1$ *seeds*, onde N é o número de canais. Cada período de sequência inclui um *slot* chamado *slot* de paridade, no qual o valor da *seed* determina qual será o canal utilizado neste *slot* de paridade. Assim, todos os rádios com a mesma *seed* garantem pelo menos um encontro neste *slot* de paridade. Para cada par (canal, *seed*), o período da sequência é dado por $N + 1$ slots e os rádios se encontram exatamente uma vez dentro de um período em que utilizam *seeds* diferentes.

Na solução AMRCC (*Adaptive Multiple Rendezvous Control Channel*) [34] cada rádio, na construção da sequência de saltos, recorre a uma tabela de classificação de canais ordenada (*ranking table*), montada a partir do resultado de sensoriamento de canais. Esta ordenação segue dos canais onde foi detectada menor atividade de UPs para os canais com maior atividade. Os autores partem da ideia que o resultado de sensoriamento de atividade de UPs para rádios vizinhos são os mesmos, com alta probabilidade. Isto implica em uma alta correlação entre as tabelas de classificação de canais entre rádios vizinhos e, conseqüentemente, uma alta correlação entre suas sequências de saltos. Após a construção da tabela de classificação de canais, o rádio gera uma sequência pseudo-aleatória que contém valores inteiros no intervalo $[1, C]$, onde C é o número de canais. A sequência pseudo-aleatória é inclinada (*biased*) para os valores mais baixos da tabela, ou seja, quanto menor o valor, com maior frequência o canal correspondente aparece na sequência. O trabalho propõe duas variantes do esquema: **i)** *basic* AMRCC, e **ii)** *enhanced* AMRCC, que se diferenciam na forma de calcular o número de ocorrências de cada canal na sequência, usando uma função linear decrescente, ou função parabólica decrescente, respectivamente.

A solução *Jump-stay* (*Jump-Stay Based Channel-hopping Algorithm*) [2] opera em duas fases de funcionamento: uma fase “*jump*” e outra fase “*stay*”. A ideia básica é a de gerar a sequência de salto de canais em rodadas, onde cada rodada é composta por um padrão “*jump*” e um padrão “*stay*”. Na fase “*jump*” o rádio pode saltar pelos canais disponíveis, enquanto na fase de “*stay*” o rádio permanece em um canal específico. Para gerar os dois padrões, o rádio é obrigado a selecionar três parâmetros de antemão: P , que é o menor número primo maior que M (número de canais), um número diferente de zero r_0 gerado a partir de $[1, M]$ e um índice i_0 a partir de $[1, M]$. Em cada rodada, temos o padrão “*jump*” e subsequente o padrão

“stay” por $2P$ slots e P slots, respectivamente. Ou seja, cada rodada leva $3P$ slots de tempo total. No padrão “jump”, o rádio começa com o índice i_0 e mantém o salto de canais até $[1, P]$ com passos determinados pelo tamanho r_0 , calculados por operações de módulo de P . No subsequente padrão “stay”, o rádio permanece no canal r_0 . Por exemplo, supõem-se $M = 4$, $r_0 = 1$ e $i_0 = 3$. Com P igual a 5 que é o menor número primo maior que 4. A completa sequência de saltos gerada em uma rodada é: $\{3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 1, 1, 1, 1, 1\}$ a parte em itálico refere-se ao padrão “jump” enquanto os 5 últimos números representam o padrão “stay” [2].

Outras características sobre alguns protocolos baseados em sequência de saltos estão resumidas na Tabela 2.1 a seguir:

Protocolos	Sequências distintas para TX e RX	Mesma visão do espectro	Síncrono	Atividade de UPs
SeqR	não	sim	não	não
SSCH	não	sim	sim	não
<i>mclock</i>	não	sim	não	não
<i>quorum</i>	sim ¹	sim	sim ²	não
ETCH	não	sim	sim/não	não
SYN-MAC	não	não	sim	não
AMRCC	não	sim	não	sim
<i>Jump-stay</i>	não	sim	sim/não	não

Tabela 2.1: Características de diferentes protocolos baseadas em sequência.

Na Tabela 2.1, a coluna “Mesma visão do espectro” indica se os usuários primários possuem alcance de cobertura sobre todos os usuários secundários. A coluna “Atividade de UPs” está relacionada ao mecanismo de atribuição de sequências utilizar alguma informação sobre a atividade de UPs ao criar sua sequência de saltos.

Algoritmos	carga	TTR médio
SeqR	$1/N$	$(N^2 + N)/2$
SSCH	$1/(N - 1)$	$(N+1)/2$
<i>mclock</i>	-	$2P^2/(P-1)$
<i>quorum</i>	$2/3$	$3/2$
ETCH	$1/N$	$(2N-1)/2$
SYN-MAC	$1/N$	$N/2$
AMRCC	$2/N+1$ ou $1/\sum_{i=2}^N(N-i+1)^2$	$N(N+1)/4$ ou $\sum_{i=1}^N(N-i+1)^2/2$
<i>Jump-stay</i>	-	$5P/3 + 11/3 + 1/(N-1)$

Tabela 2.2: Métricas de carga e TTR médio de diferentes algoritmos de *rendezvous*.

¹Existem dois modelos, um assíncrono e outro síncrono o valor refere-se ao caso assíncrono. Caso seja a solução síncrono a mesma sequência é utilizada para transmitir e receber.

²Na tabela colocamos a classificação apenas do modelo síncrono conforme em [15].

A Tabela 2.2 apresenta os valores de duas métricas relacionadas à sequência de salto de canais. Nesta tabela, o valor de N indica o número de canais disponíveis.

A métrica “carga”, introduzida por [15], representa o número de vezes que qualquer canal pode aparecer na sequência de saltos. Na Tabela 2.2, os valores de carga e TTR médio apresentados para o *quorum* referem-se ao modelo *quorum* síncrono conforme [15]. Para esta métrica, quanto menor o valor melhor, pois indica uma melhor distribuição no uso de canais. Se uma grande proporção de rádios vizinhos encontram-se no mesmo canal, pode acontecer o mesmo congestionamento no caso de uso de apenas um CCC [15]. Embora apresentada, esta métrica “carga” não será avaliada durante as simulações realizadas neste trabalho. Alguns valores de “carga” foram deduzidos enquanto outros foram retirados de [2].

Para o AMRCC (*Adaptive Multiple Rendezvous Control Channel*) [34] apresentamos duas medidas, pois existem dois tipos de AMRCC, o chamado *basic* e o *enhanced*, cujos valores seguem respectivamente na Tabela 2.2.

O valor P , presente nas métricas do *Jump-stay* [2] e *mclock*, indicam o menor número primo, maior ou igual ao número de canais N . O valor de TTR médio apresentado para o *Jump-stay* trata-se do modelo síncrono do algoritmo.

2.4 Deficiências do CCC e Protocolos Baseados em Sequência

Conforme apresentado no Capítulo 1, ambas soluções, seja utilizando CCC ou protocolos baseados em sequência apresentam deficiências.

A principal desvantagem de utilizar a abordagem do CCC é que ela é susceptível às atividades de UPs. Quando um UP passa a utilizar o CCC, todos os USs devem adiar suas transmissões no CCC, e desocupar o canal imediatamente. Além disso, o conjunto de canais disponíveis na rede de RC, incluindo o CCC, muda dinamicamente, o que dificulta o estabelecimento de um canal de controle sempre disponível para todos os USs.

Além disso, o princípio de abordagem do CCC força todos os USs transmitirem todos os pacotes de controle através do CCC. Assim, a taxa de colisão de pacotes de controle é alta quando o número de usuários na rede é grande, uma vez que todos os usuários utilizam apenas este canal para negociação. Este procedimento, obviamente, degrada o funcionamento da rede, uma vez que as transmissões de dados dependem do sucesso de negociações no CCC. Consequentemente, os autores de [41] sugeriram ajustar a largura de banda do CCC para reduzir a possibilidade de saturação. Por fim, usar um CCC estático cria uma vulnerabilidade de segurança, uma vez que se pode incentivar adversários a lançarem ataques neste canal de controle.

Os protocolos baseados em sequência de salto de canais apresentam como vantagem o atraso na descoberta de vizinhos. Dependendo do valor do TTR entre duas sequências, o encontro entre dois rádios pode ser elevado. Porém, a maior deficiência dos protocolos baseados em sequência é a falta de uma forma de avisar/informar todos os rádios vizinhos de um rádio emissor, ou seja, existe a carência de um meio onde um nó possa comunicar e receber atualizações sobre comunicações entre rádios vizinhos. Esta deficiência pode ser vista como a dificuldade de se enviar uma mensagem em *broadcast* em uma rede onde os rádios utilizam salto de canais, pois os encontros tendem a ser distribuídos pelos *slots* das sequências. Essa falta de informação sobre as comunicações de rádios vizinhos pode causar problemas de terminal escondido, denominado terminal escondido multicanal. Atualmente pelo nosso conhecimento, grande parte dos protocolos baseados em sequência sofrem deste problema.

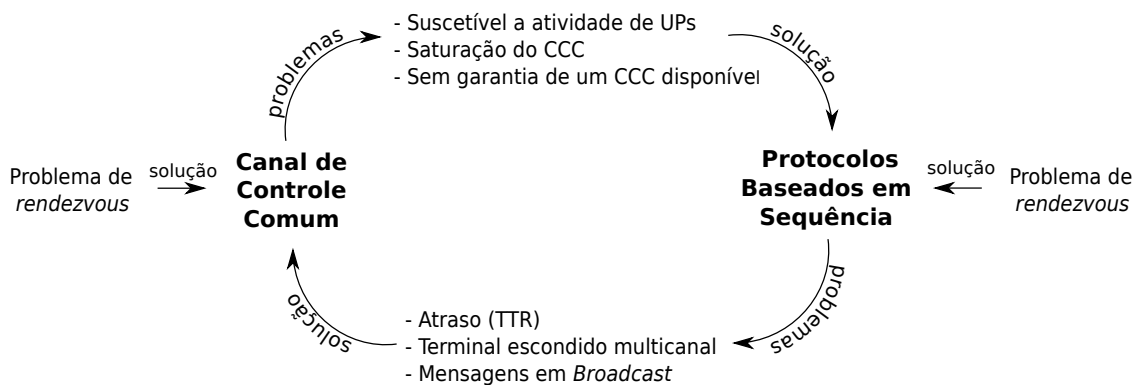


Figura 2.8: Ciclo de *Rendezvous*.

A abordagem de canal de controle proporciona protocolos simples e eficientes. Possui a vantagem adicional de eliminar os processos de descoberta de vizinhança, e causa menores atrasos no acesso ao canal, no entanto seu funcionamento sofre de problemas de saturação do canal de controle e necessita da disponibilidade de um canal livre para todos os rádios. Caso o CCC venha a ficar indisponível, por exemplo devido ao aparecimento de algum usuário primário, o funcionamento da rede torna-se comprometido. Por outro lado, protocolos baseados em sequência podem compensar os problemas devido à atividade de UPs que a abordagem CCC apresenta, pois apresentam uma maior robustez a esta atividade. Além disso, o problema de saturação de algum canal causado pelo número grande de acessos tende a ser amenizado, devido aos encontros não necessariamente ocorrerem no mesmo *slot*. No entanto, protocolos baseados em sequência sofrem do problema do terminal escondido multicanal devido à deficiência de recepção de mensagens de controle que informem sobre as comunicações de rádios vizinhos. Ou seja, o tipo informação que pode ser comunicado através de transmissões em *broadcast*. Esta deficiência não

existe quando se utiliza um CCC, pois uma informação pode ser disseminada para todos os rádios de forma natural pelo canal de controle. Ou seja, as deficiências de uma solução levam à utilização da outra, isto cria o chamado Círculo de *Rendezvous*, conforme Figura 2.8.

Desta forma, propomos na primeira parte deste trabalho uma solução aplicada à abordagem de uso de canal de controle, mais especificamente à categoria de **Múltiplos Canais de Controle**. Esta proposta aplica a ideia de múltiplos canais de controle dentro de um ambiente de grupos de rádios. Pode-se pensar também que ela mescla a utilização de grupos de rádios e uso de sequência de salto de canais, e seu objetivo é deixar o meio por onde trafegam as mensagens de controle mais imune à influência da atividade de UPs.

Na segunda e terceira partes, nos concentramos nos **protocolos baseados em sequências de salto de canais** pois este tipo de abordagem apresenta, no geral, mais vantagens do que quando se utiliza um CCC, além de possibilitar uma maior independência e robustez na forma de funcionamento dos rádios cognitivos. Assim, propomos uma solução ao problema de recepção de mensagens de *broadcast* em um cenário que aplica o uso de sequência de salto de canais, e após apresentamos um conjunto de regras de funcionamento para um rede de rádios cognitivos, a ser utilizado não apenas no primeiro encontro entre os rádios e sim em todo funcionamento da rede. Esta última proposta pode ser vista como um protocolo de acesso ao meio.

Capítulo 3

Múltiplos Canais de Controle em um Ambiente de Grupo de Rádios

Nesta primeira parte do trabalho, propomos a utilização de uma sequência de salto de canais dentro de uma formação de grupos de rádios. A ideia é que dentro de um grupo de rádios ao invés de se utilizar um único canal para a troca informações de controle, os rádios possam recorrer a uma sequência de canais. Esta sequência é estabelecida a partir de uma máquina de aprendizagem e tem a finalidade de tornar os rádios menos susceptíveis à atividade de UPs. Além disso, a maioria dos trabalhos que utilizam a ideia de agrupamento dos rádios assume que todos os USs da rede possuem a mesma “**visão do espectro**” e assim enxergam os mesmos canais livres, o que nem sempre é verdade. Este fato dificulta selecionar um único canal como controle para o grupo de rádios.

3.1 Múltiplos Canais de Controle e Sequência de Salto de Canais

Conforme apresentado no Capítulo 2, dentre as formas de solução ao problema de *rendezvous*, em que os rádios usam canais compartilhados com UPs, temos: o uso de CCC (*Commum Control Channel*) combinado ao agrupamento dos rádios, no caso de **Múltiplos Canais de Controle**, e a opção de o uso de **sequência de salto de canais**.

Em geral, soluções de **Múltiplos Canais de Controle** são empregadas em redes do tipo CWM (*Cognitive Wireless Mesh*), onde os rádios formam grupos e dentro de cada grupo reservam um mesmo canal para controle, o que facilita seu gerenciamento. Estes trabalhos utilizam a ideia de que ao se formarem grupos, aumentam-se as chances de que exista um canal livre, para controle, entre os rádios do grupo. Visto que existir um único canal de controle (CCC) para todos os rádios

da rede é mais difícil em um ambiente DSA.

Porém, existem situações em que, em função da disposição dos UPs e suas atividades, nem todos os componentes do grupo possuem canais livres em comum. Normalmente, trabalhos nesta área não levam em consideração que os rádios, mesmo próximos, podem ter “**visões do espectro**” diferentes e assim canais disponíveis diferentes. Isto decorre de diferentes influências espaciais e/ou temporais dos UPs.

Desta forma, propomos ao invés de se empregar um único canal como de controle dentro de um grupo (*cluster*), podermos utilizar mais de um canal em sequência para a transmissão das mensagens de controle [42]. Ou seja, mesclamos duas categorias de soluções ao problema de *rendezvous*, a de **Múltiplos Canais de Controle** e uso de **sequência de salto de canais**, aumentando desta forma as chances de estabelecer um meio de comunicação entre os rádios para transmissão de mensagens de controle. Além disso, este tipo de abordagem torna os USs mais robustos à atividade dos UPs, frente ao caso de um único canal de controle.

No entanto, conforme já informado, a ideia de uso de sequências de salto de canais, quando aplicada ao ambiente de RCs, deve possuir algumas ressalvas quanto à sequência de transmissão utilizada. Surge uma dificuldade que é descobrir uma sequência de salto de canais, em que os rádios possam receber e enviar mensagens entre os próprios componentes do grupo sem interferir nos usuários licenciados que estejam ao seu alcance. Além disso, em decorrência da atividade de UPs, uma sequência de canais utilizada pode deixar de ser a melhor no decorrer do tempo, com isso os USs devem identificar quando uma sequência deve ser atualizada para uma sequência melhor.

Para a questão da definição da sequência de salto de canais dentro do grupo, não utilizamos uma solução da literatura, mas propomos o uso de uma máquina de aprendizagem, chamada *Reinforcement Learning* (RL). Desta maneira, a máquina de aprendizagem é encarregada de descobrir uma sequência de salto de canais que leve a um melhor desempenho na taxa de entrega de mensagens de controle entre os rádios de um mesmo grupo. Escolhemos a máquina de aprendizagem RL, pois por característica ela possui uma complexidade computacional baixa, o que a torna atrativa para o uso embarcado em RCs.

Na Seção 3.2 apresentaremos a ferramenta RL e, após, mostraremos como modelamos o problema, seguido das soluções propostas, os resultados obtidos através de simulação e por fim a conclusão do capítulo.

3.2 Máquina de Aprendizagem *Reinforcement Learning*

Na área da ciência da computação, *Reinforcement Learning* é um tipo de máquina de aprendizagem que preocupa-se com a maneira com que um **agente** realiza **ações** em um **ambiente**, de modo que alguma métrica, referida como **recompensa**, seja maximizada. Resumidamente, em RL temos agentes que, ao inspecionarem um ambiente, realizam alguma ação que se reflete em uma métrica, também coletada do ambiente, chamada de recompensa ou custo. A partir do valor da recompensa coletada, o agente percebe se realizou uma boa ou má escolha.

O RL adota uma abordagem simples com uso de poucos parâmetros, e desta forma a complexidade envolvida na modelagem do meio pode ser minimizada [43]. Em contrapartida a esta facilidade de adaptação e baixa complexidade computacional, esta ferramenta pode possuir o inconveniente de ter uma convergência lenta.

Esta máquina de aprendizagem possui diversos algoritmos como SARSA (*State-Action-Reward-State-Action*), ACLA (*Actor Critic Learning Automaton*) [44], *Q-learning* [45] etc. A principal diferença destes algoritmos se dá na forma em que as ações são contabilizadas em recompensa, com pesos dados às ações, ao histórico de ações e/ou expectativa de recompensas futuras. Apresentaremos em mais detalhes o algoritmo do tipo *Q-learning* que foi o utilizado para definir a sequência de salto de canais. Utilizamos o algoritmo do tipo *Q-learning* pois dentre os estudados é o que necessita de menos parâmetros na modelagem do problema.

Q-learning é um algoritmo *on-line* do tipo RL que busca determinar uma política/decisão ótima, sem uma modelagem detalhada do ambiente. Este algoritmo possui alguns parâmetros apresentados a seguir:

- **Épocas de decisão** \rightarrow denotadas por $t \in T$, $T = \{1, 2, \dots\}$. Indicam quando o algoritmo entra em execução;
- **Estados** $\rightarrow s \in S$. São próprios da modelagem de cada problema;
- **Ações** $\rightarrow a \in A$. Representam as decisões que podem ser realizadas;
- **Recompensas** $\rightarrow r_t(s_t, a_t)$. Relativas a um par estado-ação, é o indicativo do quão boa ou não foi a **ação** realizada;
- **Q-value** $\rightarrow Q_t(s_t, a_t)$. Indica o valor que é atualizado através da coleta de **recompensas**;
- **Q-table** \rightarrow Tabela onde os **estados** são as linhas e as colunas são as **ações**, e onde os **Q-value's** são armazenados. Assim, possui $S \times A$ entradas.

Cada agente mantém um *Q-value* $Q_t(s_t, a_t)$ para todas as ações possíveis em sua *Q-table*. O *Q-value* estima o nível de recompensa para o par estado-ação, assim mudanças nos *Q-value's* levam a mudanças nas ações dos agentes.

A cada **Época de decisão** t , o agente observa seu estado atual (linha) e escolhe uma ação (coluna) em sua *Q-table*. Posteriormente à execução da ação, o agente recebe uma recompensa r_t relativa a esta ação a_t selecionada.

Com o valor da recompensa, o agente atualiza seu respectivo *Q-value* no tempo $t + 1$ conforme:

$$Q_{t+1}(s, a) \leftarrow (1 - \alpha) \times Q_t(s_t, a_t) + \alpha r_{t+1}(s_t, a_t) \quad (3.1)$$

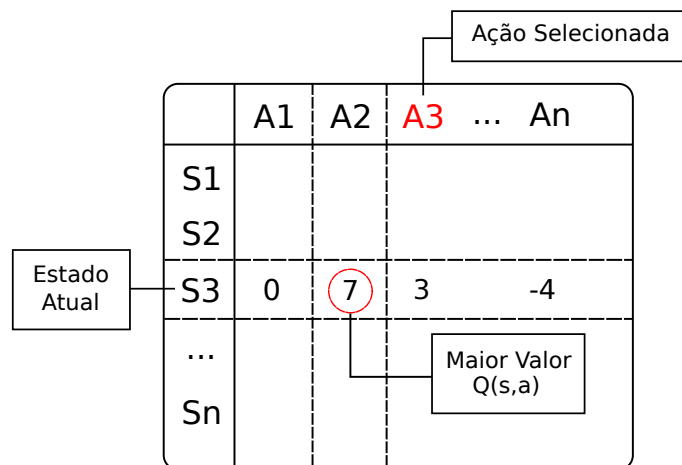
Na Equação 3.1, α é a **taxa de aprendizagem**; logo, maiores valores de α indicam maior peso a valores recentes de recompensa. Esta equação de atualização foi a mesma utilizada em [43].

O *Q-learning* inicia com toda a *Q-table* zerada e a cada instante o agente seleciona uma ação baseada em uma estratégia de exploração. Uma estratégia comumente usada é a *ϵ -greedy* [45], que seleciona uma ação gananciosa, $\arg \max_a Q(s, a)$, com alta probabilidade e, ocasionalmente, com uma pequena probabilidade seleciona uma ação aleatória. Isso tenta garantir que todas as ações e seus efeitos sejam experimentados [46].

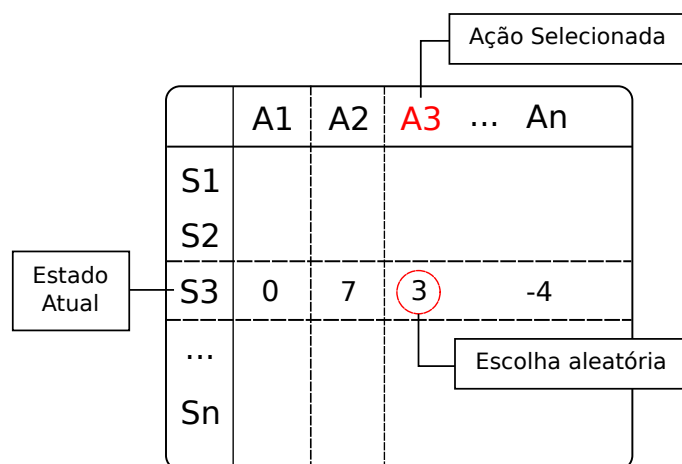
Na Figura 3.1³ temos um exemplo de *Q-table* com disposição dos **Estados** na coluna à esquerda e, as possíveis **Ações** na linha horizontal superior. No funcionamento da máquina de aprendizagem e conseqüentemente preenchimento da *Q-table*, temos por exemplo: o **Agente** (rádio) no **Estado** atual, neste exemplo S3. Em S3 o **Agente** pode selecionar uma **Ação** com uma probabilidade de $(1 - \epsilon)$. Esta probabilidade representa o caso do **Agente** optar por uma estratégia de *exploitation*, que indica que no **Estado** S3 o **Agente** irá pesquisar sobre a linha correspondente ao **Estado** atual, e selecionará a **Ação** que até então recebeu maior recompensa por ter sido selecionada no passado, no caso a **Ação** A2. Com probabilidade ϵ o **Agente** pode realizar a estratégia de *exploração* (*exploration*). Durante a *exploração* o **Agente** irá selecionar uma **Ação** de forma aleatória, desta forma o **Agente** poderá experimentar outras **Ações** e/ou **Estados** menos visitados ou ainda não visitados. O objetivo da *exploração* é fazer com que o **Agente** tente todas as **Ações** e **Estados**, pois pode ocorrer do **Agente** ter encontrado um ótimo local ao invés de um ótimo global.

Na seção seguinte, apresentaremos como utilizamos o RL para tratar do problema da definição da sequência de salto de canais em uma topologia de grupos.

³Tabela adaptada de fontes da Internet.



(a) *Exploitation*



(b) *Exploração*

Figura 3.1: *Q-table* - exemplo de *exploitation* e exploração.

3.3 Modelagem do problema

Conforme apresentado, a proposta é mesclar a solução de salto de canais em uma formação em *cluster*, utilizando a ferramenta de RL para encontrar a sequência de saltos de canais que será utilizada para os rádios trocarem mensagens de controle.

Na modelagem, consideramos três tipos de nós, a saber: nós UPs, nós USs comuns e nó US de controle. A modelagem será aplicada exclusivamente aos nós USs e, conforme temos diferentes USs, seus comportamentos também serão distintos.

Antes dos nós iniciarem todo o procedimento, que será apresentado a seguir, é necessário que exista uma coordenação entre eles, justamente para definir qual dentre os nós será o US de controle. Pode-se utilizar qualquer solução de **Múltiplos Canais de Controle** desde que a solução empregue o uso de um nó que seja líder do grupo de rádios.

Assumimos que todos os nós possuem apenas uma interface de rede; desta maneira, os nós USs devem alternar seu tipo de funcionamento. Por um intervalo de

tempo, o nó deve se preocupar com as mensagens de controle e, em outros instantes, pode trocar dados com outro US. Assim, definimos dois períodos, um chamado de **Período de Controle** e outro de **Período de Utilização** ou dados. Este procedimento é necessário devido aos USs não poderem utilizar permanentemente um canal escolhido, pois suas transmissões podem interferir em algum nó UP que até então estava fora de atividade, seja por questões espaciais ou temporais. O método entrará em execução durante o **Período de Controle**. Assim nesta etapa de **Período de Controle**, a máquina de aprendizagem passa a funcionar, desde a **Época de decisão**, onde ocorre a escolha da **Ação**, de acordo com o **Estado** em que o rádio se encontra, até a coleta de **Recompensa**. Este funcionamento se aplica aos USs e ao US de controle. Assim, nestes rádios a sequência de salto de canais é definida e avaliada pela coleta de **Recompensas**, no caso recebimento das mensagens de controle e ACKs para os USs e US de controle, respectivamente. Na Figura 3.2, apresentamos o comportamento do nó US de controle.

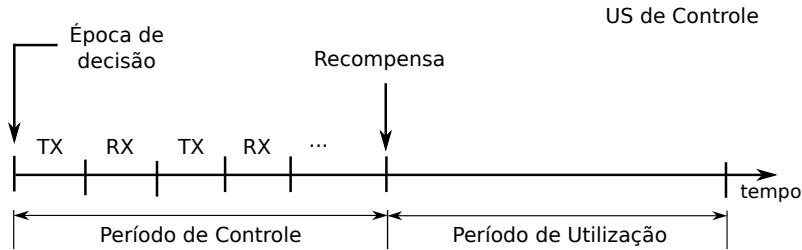


Figura 3.2: Método de funcionamento.

A transmissão da mensagem de controle é realizada pelo US de controle, enquanto os nós USs comuns confirmam com ACK o recebimento desta mensagem. No início do período de controle (Época de decisão), a primeira atividade do nó é recorrer a algum mecanismo para decidir qual será seu canal de controle. Na parte TX do período de controle é que o nó irá realmente transmitir pelo canal escolhido. Na parte RX, o nó US de controle reservará um tempo para o recebimento de ACKs de seus vizinhos. Este recebimento deve ser sincronizado, com um *slot* de tempo reservado para cada nó US vizinho do nó US de controle. Este procedimento é para evitar que ocorra uma colisão de ACKs. O comportamento dos nós US comuns é o inverso. Na Época de decisão, eles também recorrem ao mecanismo de decisão, porém para determinar em qual canal eles irão “ouvir”. Depois, no tempo reservado, os USs comuns passam para a etapa em que cada um irá transmitir seu ACK no instante apropriado. Este procedimento se repete a cada conjunto TX-RX da Figura 3.2. Assim, durante a Época de decisão, o nó US de controle terá uma sequência de canais em que ele irá transmitir, enquanto os USs comuns terão uma sequência de canais onde eles irão “ouvir” a mensagem de controle. No Período de Controle todos

os canais disponíveis podem ser utilizados. Estes canais utilizados neste período são definidos pelo uso de algum mecanismo, neste caso o RL.

Este procedimento pode ser repetido até que uma sequência seja estabelecida e, depois disso, a comunicação de controle possa ser transmitida adotando esta sequência. Como consequência, o tamanho da sequência indicará a quantidade de conjuntos TX-RX.

Conforme dito anteriormente, antes dos nós iniciarem todo este procedimento proposto é necessário que exista uma coordenação entre eles. Esta coordenação é para justamente definir qual dentre os nós será o US de controle, e além disso, concluir a formação do *cluster*. Existem diversas possibilidades de se chegar a tal configuração. Como primeiro passo, pode ser feita uma descoberta de vizinhança para cada nó ter a indicação de quem são seus vizinhos. A dificuldade é que até então não existe um canal de controle em que os nós possam se comunicar afim de obterem tal informação, ou seja, o *rendezvous* entre os rádios ainda não ocorreu.

Como solução, cada nó pode varrer o meio e, ao primeiro canal livre de atividade de UPs, o nó opte por transmitir uma mensagem como um *hello*, informando sua identificação, ou opte por apenas ouvir o meio. Todos os nós realizam semelhante procedimento; caso nenhum nó receba *hello* algum, pode ter ocorrido de todos terem optado por ouvir o meio. Logo, repete-se o procedimento até que algum nó escolha transmitir. Ao primeiro recebimento de algum *hello*, este nó receptor responde com um ACK em oportunidade breve. No ACK, o nó informa quantos vizinhos ele tem conhecimento até o momento. Assim, cada nó verifica se existe algum outro nó com maior quantidade de vizinhos que ele próprio. Desta forma, o nó mais indicado a ser o US de controle é o que possui maior número de vizinhos. Assim que um nó descobre que é o que possui mais vizinhos, ele notifica os demais do grupo que será o US de controle. Neste procedimento, podem ocorrer colisões de mensagens pois nenhum nó está sincronizado; logo, a eleição do líder pode ser demorada.

Como o nó US de controle sabe a quantidade de vizinhos que possui, ele pode construir um esquema de transmissão e recepção do tipo TDMA (*Time Division Multiple Access*), e informar quando cada nó terá a oportunidade de TX e RX. Esta ordem pode ser atribuída pelo número de identificação de cada nó. Para o bom funcionamento é preciso uma sincronização, assim, a cada mensagem trocada entre os nós, já no período de utilização do mecanismo de decisão (decisão do US de controle), pode-se incluir informações de *clock* que auxiliem no sincronismo. Existem outras maneiras de eleição de líder como em [47], onde por exemplo, inicia-se por um algoritmo de cobertura mínima (*Minimal Spanning Tree*) seguido por um outro de eleição de líder, que opte pelo nó que tenha o grau mais elevado.

Conforme informado anteriormente, qualquer meio ou solução de formação de grupos de rádios pode ser aplicado, desde que exista um nó líder de cada grupo. A

proposta desta parte do trabalho está relacionada à formação da sequência de salto de canais dentro do grupo. Assim, propomos um mecanismo de decisão do tipo RL para determinar a ordem de canais a se utilizar na sequência. A forma como foi implementado este mecanismo de decisão será apresentada na subseção a seguir.

3.3.1 Algoritmo de *Reinforcement Learning*

A modelagem do problema consiste em o que definiremos como estados, ações, recompensas e como atualizaremos os valores da *Q-table*. Realizamos duas modelagens chamadas RL-Ciclo e RL-Seq, onde apresentaremos em cada uma as ações e estados definidos.

A recompensa para o nó US de controle está relacionada à quantidade de ACKs recebida, enquanto que para os nós US comuns equivale à quantidade de mensagens de controle recebidas. A atualização da *Q-table* segue a Equação 3.1 apresentada anteriormente e ambos nós USs utilizam o Algoritmo 1 a cada Época de Decisão.

No início, todos os pares estado-ação da *Q-table* são completados com zeros. Realizada esta fase de inicialização de variáveis, entramos na fase de aprendizado. Nesta fase, em decorrência do valor de ε e b sorteado, o nó realizará exploração ou não. Caso opte por exploração uma ação aleatória será selecionada dentre as possíveis no estado atual. Caso não se realize exploração, o nó no estado atual buscará pelo maior valor até então preenchido de *Q-value*, e assim optará por esta ação. Se todos os campos forem zero, escolhe-se uma ação aleatoriamente.

Selecionar exploração ou não, e depois o maior valor até então obtido, ocorre devido a utilização da estratégia ε -greedy. Após, executa-se a ação, recolhe-se a recompensa adquirida e atualiza-se o valor da *Q-table*. Ao fim, adicionamos um teste para agilizar a convergência, caso o *Q-value* obtido seja menor que um limiar δ , passamos o valor de *Q-value* para zero. O procedimento de aprendizagem é

repetido durante todo período de funcionamento.

Algoritmo 1: *Q-learning* com estratégia ε -greedy.

```
1 Inicialização das variáveis:  $\varepsilon$ ,  $\alpha$  e  $s_t$ ;  
2 begin  
3    $t = 0$   
4   for para cada  $s \in S$ ,  $a \in A$  do  
5     Inicializo Q-value para todas as ações  $a_t$  de cada estado  $s_t$  com  
6     zeros  
7   end  
8 Aprendizado:  
9 repeat  
10   Gero um número aleatório  $b$  entre 0 e 1  
11   if ( $b < \varepsilon$ ) then  
12     Seleciono uma ação  $a_t$  aleatoriamente  $\rightarrow$  Exploração  
13   else  
14     Escolho ação  $a_t$  que possua o maior Q-value para o estado atual ou  
15     caso todas as ações sejam zero, seleciono alguma aleatoriamente  
16   Executo ação  $a_t$   
17   Verifico recompensa  $r(s_t, a_t)$  ganha pela ação  
18   Atualizo entrada na Q-table conforme:  
19    $Q_{t+1}(s, a) \leftarrow (1 - \alpha) \times Q_t(s_t, a_t) + \alpha r_{t+1}(s_t, a_t)$   
20     if ( $Q_{t+1}(s, a) < \delta$ ) then  $Q_{t+1}(s, a) \leftarrow 0$   
21   e atualizo estado atual  $s_t = s_{t+1}$   
22 until durante período de funcionamento ;
```

3.3.2 Modelagem RL-Ciclo

De forma intuitiva, a primeira maneira de modelagem que podemos seguir é o estado representar uma sequência de canais. Porém, se modelarmos desta forma, o número de combinações pode ser grande em decorrência do tamanho estabelecido da sequência e o número de canais a se utilizar. Por exemplo, para uma sequência de tamanho a com b canais disponíveis, teríamos b^a possibilidades, o que pode tornar a convergência do algoritmo lenta.

Assim, modelamos cada estado como um canal, e não como uma sequência de canais. Por exemplo: para uma sequência de tamanho 3 com 4 canais disponíveis, como um possível funcionamento para o nó US de controle, temos:

1. Nó está no canal 1, olha sua *Q-table*, passa para o canal 2, TX no canal 2;

2. Nó está no canal 2, olha sua *Q-table*, passa para o canal 3, TX no canal 3;
3. Nó está no canal 3, olha sua *Q-table*, passa para o canal 4, TX no canal 4.

Assim, sua sequência de transmissão foi: 2, 3 e 4. Após a realização de toda a sequência é que o nó irá contabilizar o quão proveitoso foi a transmissão nestes canais; porém a recompensa não é devida à sequência escolhida, e sim pela decisão de cada canal, pois a atualização se dá no par estado-ação. Por exemplo, para a escolha do primeiro canal da sequência do exemplo anterior: estava no canal 1, passou para o 2, TX no canal 2 e recebeu 3 ACKs por esta ação. Para o par estado-ação (canal 1 e TX no canal 2), a entrada na *Q-table* será atualizada ao equivalente de 3 ACKs. Desta forma, a sequência é decidida a cada passo pela consulta à *Q-table*. Ao fim, contabiliza-se o quão boa foi esta escolha. Assim, diminuimos o número de sequências testadas e a convergência tende a ser mais rápida. Se o nó escolhesse uma sequência de início e contabilizasse as recompensas devido a esta sequência, ele teria que varrer uma quantidade grande de combinações.

A recompensa para o US de controle é a quantidade de ACKs recebida por utilizar a sequência de salto de canais indicada pelo RL, enquanto que para os USs a recompensa é a quantidade de mensagens de controle recebidas ao se seguir a sequência indicada pelo RL.

3.3.3 Modelagem RL-Seq

Na modelagem RL-Seq, o estado equivale ao ordenamento na sequência, e a ação equivale a qual canal utilizar. Assim, a primeira linha da *Q-table* refere-se à primeira posição da sequência, a segunda linha indica a segunda posição da sequência etc. As colunas representam os canais que podem ser usados. Em qualquer linha (ordenamento) pode-se selecionar qualquer canal. Inicialmente recorre-se sempre à primeira linha, e escolhe-se um canal que será o primeiro canal da sequência de transmissão. Depois, passa-se obrigatoriamente à segunda linha, novamente elege-se um canal que representará o canal da segunda posição da sequência, e assim por diante, até que toda a sequência seja preenchida. Quando atingimos a última linha da tabela retornamos à primeira.

Em ambas as modelagens, quando o nó US de controle seleciona um canal para transmitir e, no instante de enviar sua mensagem verifica que existe algum UP em atividade neste canal, o nó perde sua chance de transmitir e marca o respectivo *Q-value* com zero, independente do valor prévio, isto é para refletir uma escolha que não foi adequada.

No funcionamento do RL-Seq as recompensas são as mesmas indicadas anteriormente no RL-Ciclo.

Na próxima seção, apresentaremos os cenários utilizados para testar os dois mecanismos propostos na escolha da ordem de chaveamento de canais.

3.4 Ambiente de Simulação

Com a finalidade de avaliar o método de funcionamento proposto, aliado aos mecanismos de RL-Ciclo e RL-Seq, desenvolvemos um simulador que usa os recursos básicos do NS-2 [48], que é um simulador a eventos discretos. A proposta foi avaliada em dois cenários, um conforme a Figura 3.3 e outro onde a disposição dos nós UPs foi aleatória.

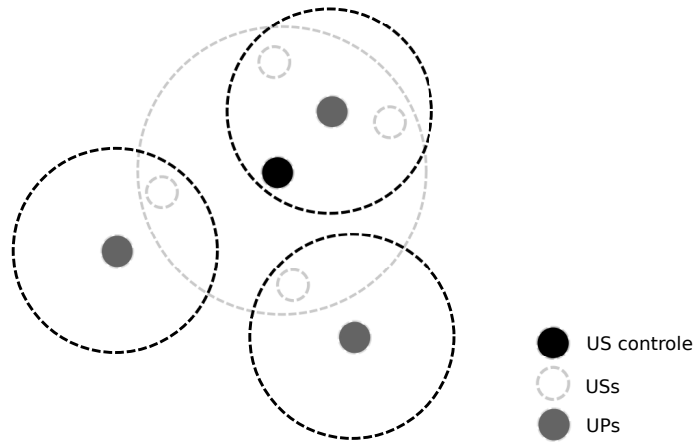


Figura 3.3: Cenário de simulação.

Para o cenário da Figura 3.3 temos 3 nós UPs, 4 nós USs e 1 nó US controle. Trata-se de um exemplo geral em que a atividade dos nós UPs deve-se a questões temporais, onde seus posicionamentos e atividades influenciam na comunicação dos nós. O nó US de controle está posicionado dentro do alcance de comunicação dos outros nós USs, e todos os nós USs são interferidos por algum UP. Assim, neste cenário, pode ocorrer o caso em que o nó US de controle envia alguma mensagem de controle, que é recebida por algum US, porém este nó US receptor pode estar impossibilitado de responder com um ACK naquele canal. Este é um caso de falso negativo para o nó US de controle, enquanto que para o nó US comum, o recebimento é visto como recompensa. Consideramos que as mensagens de controle podem ser recebidas mesmo nesta situação devido a questões de captura do meio. Vale destacar que neste cenário temos USs com “visões de espectro heterogêneas”, visto que os USs não estão no alcance de todos os UPs. Conforme ressaltado no início do capítulo, muitos trabalhos adotam um cenário onde todos os USs podem perceber a atividade de UPs, o que facilita a decisão do grupo de nós sobre qual canal a se utilizar, mas nem sempre se aplica por questões de propagação. A seguir apresentaremos os

resultados obtidos nas simulações.

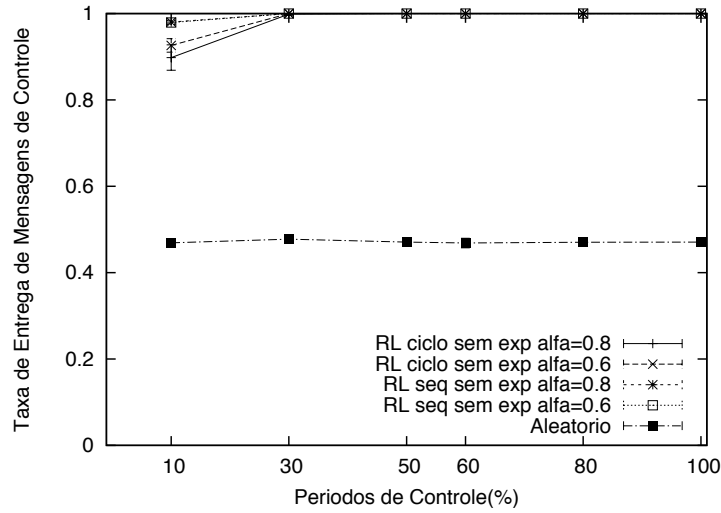
3.4.1 Resultados da Simulação

Todos os resultados são médias de 30 experimentos e os gráficos possuem intervalo de confiança de 95%. Cada experimento possui 1000 períodos de controle. O eixo x indica a porcentagem de períodos de controle até o instante da coleta de informação. A taxa de entrega contabiliza quantos nós receberam alguma mensagem no período de controle, e cada valor no gráfico representa a média da taxa de entrega entre o ponto atual e o ponto anterior, ou seja, os valores de taxa de entrega não foram acumulados.

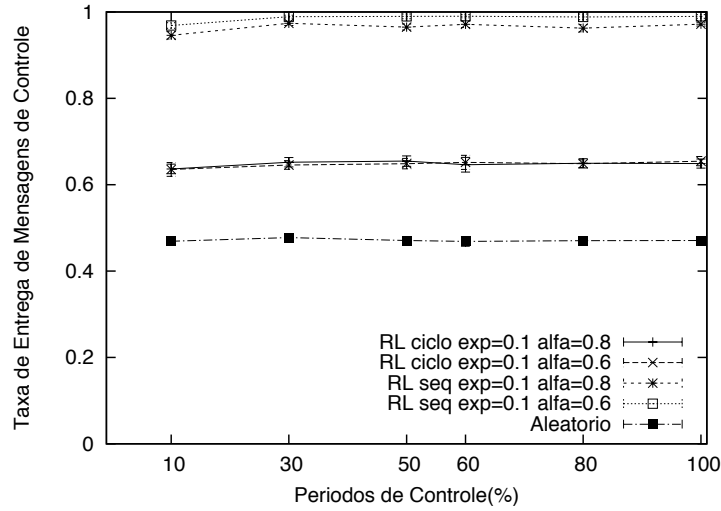
Realizamos quatro experimentos de teste. No primeiro a atividade dos nós UPs foi fixa, ou seja, uma vez definido os canais dos UPs eles não se alteraram durante o período de simulação. No segundo, a atividade do nós UPs seguiu uma distribuição exponencial, com um parâmetro λ_1 que indicou o tempo médio que o nó pôde permanecer em atividade e um λ_2 que foi relativo ao tempo médio de desligamento. Este modelo de atividade ON-OFF de UPs é amplamente utilizado na literatura e é uma boa aproximação para atividade UPs em ambientes dinâmicos. No terceiro caso, quando a simulação atingiu 30% dos períodos de controle, a configuração de canais dos nós UPs modificou-se aleatoriamente. Desta forma, pretendemos verificar o tempo de recuperação dos mecanismos RL. Em um quarto experimento, alteramos a quantidade de UPs e seus posicionamentos foram aleatórios, justamente para testarmos condições em que a comunicação poderia ser impossibilitada devido à atividade de UPs. Comparamos o uso de RL-Ciclo, RL-Seq e o caso aleatório. No caso aleatório na Época de decisão o agente selecionou aleatoriamente uma sequência de salto. O objetivo é mostrar a importância de se definir uma sequência de salto de canais no instante de envio das mensagens de controle, e também confirmar que nem sempre qualquer sequência se adequa. Nas simulações não contabilizamos perdas de mensagens devido a colisões e problemas no meio de comunicação, contudo nenhuma transmissão de US foi realizada caso no canal corrente existisse atividade de algum UP.

A Figura 3.4 apresenta os resultados com uma configuração onde os nós possuem uma sequência de tamanho 3 e podem utilizar 3 canais diferentes. Nesta configuração, cada UP inicia com um canal diferente dos outros UPs e possui o comportamento de estar em atividade ou não no canal.

Simulamos os mecanismos com valores de α (taxa de aprendizagem) sendo 0.6 ou 0.8, e examinamos o caso do nó explorar com 10% de probabilidade ou não explorar (“sem exp” nos gráficos). O comportamento dos nós UPs, entre estar ativo ou não, seguiu uma variável aleatória exponencial. Neste teste, os nós não trocavam de



(a) Sem exploração

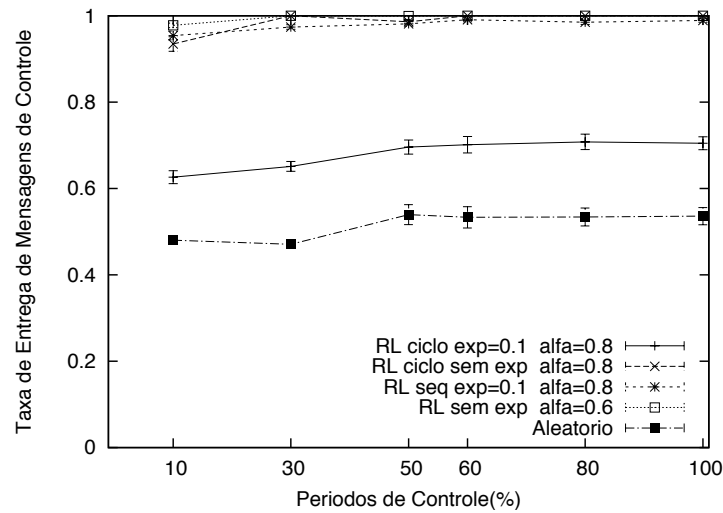


(b) Com exploração

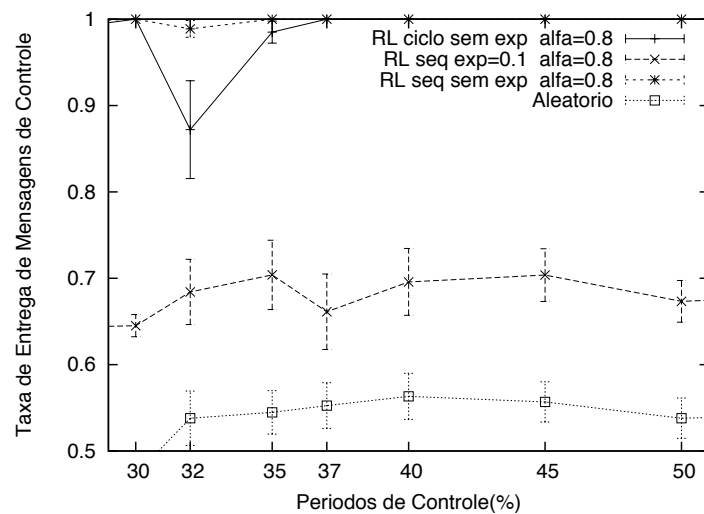
Figura 3.4: Nós UPs com atividade *On-Off*.

canal, apenas suspendiam sua atividade no canal que estavam utilizando e depois retornavam a reutilizá-lo. Usamos $\lambda_2 > \lambda_1$ assim o tempo médio que o nó permaneceu em atividade foi maior que o tempo médio que o nó esteve desligado. Verificamos uma convergência rápida nos dois mecanismos RLs propostos. Com 30% dos períodos de controle os nós já convergem para a sequência de chaveamento de canais. No caso com exploração, Figura 3.4 (b), em ambos modelos a taxa de entrega decresce, porém no caso do RL-Ciclo a perda é bem mais significativa. Os mecanismos, ao perceberem um canal livre, que antes estava em uso, passam a utilizá-lo, e quando o nó UP retorna a sua atividade, este canal deve ser liberado. Logo, os nós alternam neste comportamento, por isso a taxa de entrega decresce. O RL-Ciclo é o mais prejudicado, pois ao explorar ele passa a tentar não só um novo canal mas um novo ciclo e isto dificulta sua recuperação. Também foram realizados testes alterando a

porcentagem de exploração em 20% e, nestes casos, o comportamento dos gráficos foi similar ao caso de 10%. Embora não apresentado, os gráficos com atividade de UPs fixa obtiveram comportamento semelhante. Os valores de taxa de entrega já apresentam um valor elevado, mesmo em 10% de períodos de controle, devido a um período inicial de aquecimento (*warm-up*) para preenchimento da *Q-table* que inicia com os campos (*Q-values*) zerados.



(a) Período geral de simulação



(b) Período de recuperação

Figura 3.5: UPs trocam de canais em 30% da simulação.

A Figura 3.5 apresenta o caso em que, decorridos 30% dos períodos de controle, os nós UPs trocam aleatoriamente de canais. Neste período, os nós já possuem valores na *Q-table* que indicam os saltos de canais a seguir. Porém com a mudança de canais pelos UPs, pode-se tornar necessária a busca por uma nova sequência adequada. No gráfico da Figura 3.5 (a), temos a visão geral do período de simulação. Na Figura 3.5 (b), analisamos apenas um trecho do período com a finalidade de mostrar

o quanto em média os mecanismos levaram para se recuperar em número de períodos de controle. Podemos notar que o RL-Seq recupera-se rápido, cerca de 5 períodos de controle, enquanto o RL-Ciclo demora um pouco mais. No caso da exploração, o mais prejudicado foi o RL-Ciclo, cuja razão já foi explicada anteriormente.

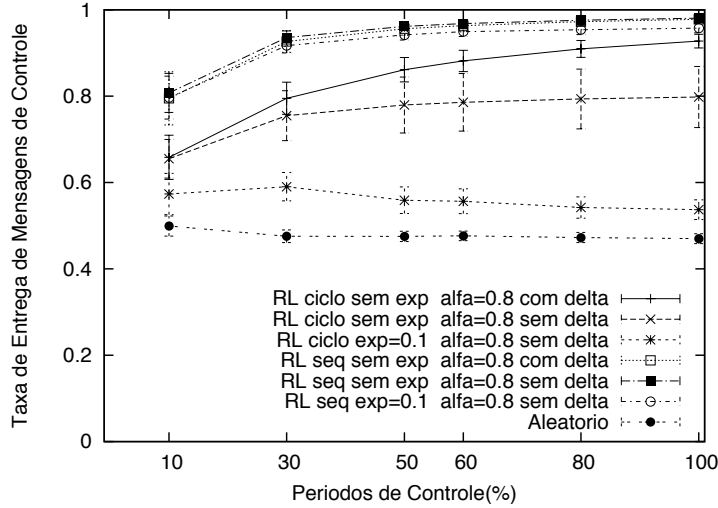
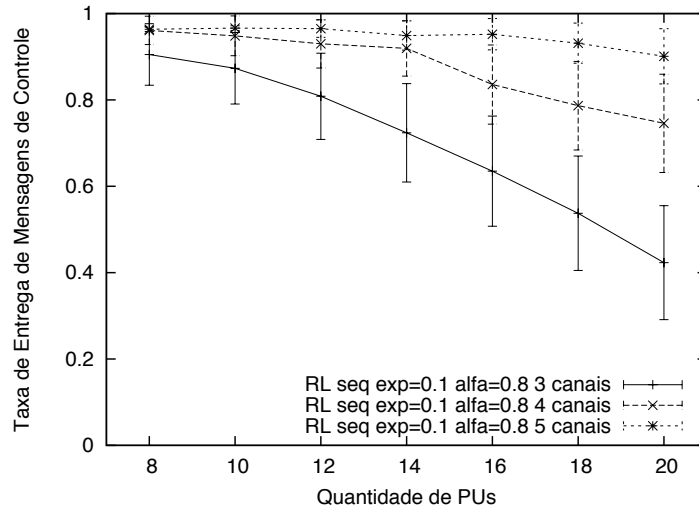


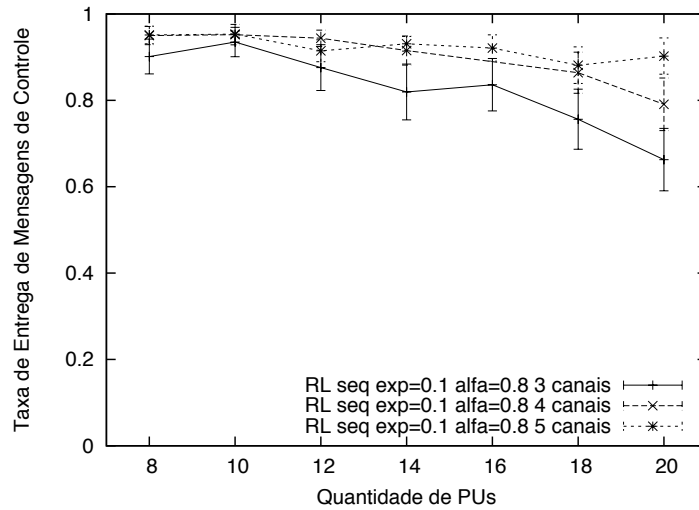
Figura 3.6: Tempo de convergência medido em períodos de controle.

Para o gráfico da Figura 3.6 mostramos apenas o número de períodos de controle de 1000 para 100, assim o gráfico equivale aos instantes de 0 até 10% dos gráficos anteriores. Testamos também a influência do valor de δ na convergência dos métodos. Na legenda, quando indicamos o método “com delta”, é o equivalente a um limiar que abaixo deste a entrada da *Q-table* foi atualizada para zero. Enquanto que nas legendas “sem delta”, o valor foi modificado a partir do funcionamento da equação de atualização. Pode-se perceber que o modelo mais influenciado pelo uso do δ foi o RL-Ciclo, pois desta forma um ciclo escolhido anteriormente, que não era um dos melhores, foi abandonado mais rapidamente.

A Figura 3.7 apresenta os resultados obtidos no cenário onde o posicionamento dos nós UPs foi aleatório. Este cenário possui uma área quadrada de 150 metros de lado, com cada nó tendo alcance de transmissão de 45 metros. Para este teste aumentamos a quantidade de experimentos de 30 para 60 rodadas. No cálculo da taxa de entrega final foi descontado um período inicial de 40% de períodos de controle. No eixo x variamos a quantidade de UPs por teste; assim, podemos perceber que conforme aumentamos a quantidade de UPs, a taxa de entrega decaiu, nos dois gráficos, justamente porque existiram configurações em que o nó US de controle ficou impossibilitado de transmitir mensagens de controle. Conforme aumentamos a quantidade de canais, cresceu a possibilidade do nó US de controle conseguir algum canal disponível para transmitir aos nós USs comuns, daí uma taxa de entrega maior. Na Figura 3.7 (b), a taxa de entrega foi maior devido às oportunidades de



(a) Atividade de UPs fixa



(b) Atividade de UPs *On-Off*

Figura 3.7: Taxa de entrega com aumento do nós UPs.

algun canal ficar livre devido a suspensão da atividade de algum nó UP.

3.5 Conclusões do Capítulo

Neste capítulo, discutimos o uso de *Reinforcement Learning* (RL) em conjunto a solução de sequência de salto de canais aplicada a uma formação de *cluster*, onde a tarefa de encontrar a sequência de saltos, utilizada como “canal de controle”, foi atribuída ao RL. Com esta finalidade, modelamos dois tipos de mecanismos chamados, RL-Ciclo e RL-Seq.

Na avaliação por simulação variamos o parâmetro de ϵ que indica uma maior possibilidade do rádio tentar explorar novos pares de ações e estados. Esta medida está relacionada a tentativa de descobrir um ótimo global caso o rádio até então tenha

encontrado um ótimo local. Modificamos também os valores de α que indicam um maior peso a ações do passado e ações do presente.

Uma outra métrica a ser considerada durante as simulações e avaliações das propostas RL-ciclo e RL-Seq foi o tempo de convergência. Embora esta métrica não tenha sido avaliada propriamente durante todo o período de simulação, a Figura 3.6 avalia este tempo no período inicial de funcionamento.

Outra métrica possível de se utilizar é a latência na recepção de mensagens. Esta métrica não foi avaliada pois na simulação a recepção das mensagens se deu de forma atômica, isto em decorrência do simulador utilizar os recursos básicos do NS-2, como por exemplo o agendamento de eventos discretos e gerador de números aleatórios. Porém a latência está relacionada a duração do **Período de controle**, que justamente se inicia com a seleção da sequência e termina com a coleta de recompensas devido seu uso no envio e recepção das mensagens.

Pelos resultados obtidos pode-se constatar que com o emprego dos mecanismos de RL propostos, em especial o RL-Seq, os nós conseguiram descobrir sequências de saltos de canais que forneceram um desempenho satisfatório para a taxa de entrega de mensagens de controle, mesmo em ambientes com maior atividade de nós licenciados.

Esta solução do grupo de nós utilizar uma sequência de salto de canais ao invés de um único canal como canal de controle, garantiu aos rádios uma robustez às atividades de UPs. Este resultado pode ser visto pela Figura 3.7 que relaciona a taxa de entrega pela quantidade de UPs na rede, e quantidade de canais disponíveis para uso. Pelos gráficos temos que quanto menor o número de canais disponíveis, para uso, menor foi a taxa de entrega. Além disso, o uso do RL permitiu uma maior autonomia aos nós quanto à atualização da sequência de salto de canais a se utilizar.

Capítulo 4

Transmissões em *Broadcast* Para Soluções Baseadas Em Sequência de Salto de Canais

Nesta segunda parte do trabalho, focamos nas soluções de **protocolos baseados em sequência de salto de canais**, que conforme abordado nos Capítulos 1 e 2, possui como uma de suas deficiências a recepção de mensagens em *broadcast*. Assim, neste capítulo, apresentaremos a motivação e proposta da criação de uma sequência de *broadcast*, que pode ser utilizada conjuntamente com qualquer solução que use uma sequência de salto de canais. Mostraremos a formalização de uma sequência de salto de canais em conjunto com a formalização da sequência de *broadcast*. Ao fim, apresentaremos os resultados, a partir de simulações, do uso da sequência de *broadcast* frente à abordagem tradicional empregada em soluções baseadas em sequências de salto de canais.

4.1 Motivação para Sequência de *Broadcast*

Como já apresentado, o problema de *rendezvous* em RCs pode ser mitigado através da criação de uma sequência de saltos de canais, que determina a ordem com que cada rádio visita os canais disponíveis. Diversas soluções que aplicam a ideia de sequência de salto de canais, classificadas como **protocolos baseados em sequência de salto de canais** foram publicadas, as quais se diferem principalmente quanto à forma de criação das sequências, quanto ao tamanho das sequências e o tempo máximo previsto para o encontro.

Em geral, quanto mais informação se possui sobre os demais rádios da rede, mais simples e rápido torna-se o *rendezvous*. Em casos mais extremos de falta de informação, o *rendezvous* torna-se mais difícil e é conhecido como *blind rendezvous*.

Vale ressaltar que classificamos como *blind rendezvous* trabalhos em que os rádios não possuem informações sobre USs vizinhos, tais como a quantidade de vizinhos de um rádio, canais livres dos nós vizinhos, etc. Porém se alguma informação for agregada posteriormente, desde que coletada pelo próprio rádio, o trabalho continua sendo caracterizado como *blind rendezvous*.

Conforme capítulos anteriores, uma das maiores deficiências dos **protocolos baseados em sequência de salto de canais** é a falta de um meio que possibilite que alguma informação seja disseminada para todos os rádios, ou mesmo para um grupo de rádios, de forma rápida. Como solução, propomos realizar um *rendezvous* entre múltiplos usuários que pode ser conseguido através de uma transmissão em *broadcast*. Propomos a construção de uma sequência de *broadcast* a ser utilizada pelo rádio quando este desejar transmitir uma mensagem de *broadcast*.

Esta sequência de *broadcast* pode ser utilizada para transmitir aos rádios vizinhos do rádio emissor, chamado de *broadcaster*, informações de controle sobre a rede de RCs, assim como informações sobre comunicações entre rádios vizinhos.

A troca de informações relacionadas a comunicações entre rádios vizinhos pode evitar problemas de terminais escondidos, como também pode melhorar o aproveitamento na utilização do espectro, dado que esta informação pode auxiliar em uma coordenação no uso de canais entre transmissores e receptores que estão dentro do mesmo alcance. No entanto, a maioria dos protocolos baseados em sequência não tratam diretamente da questão de *rendezvous* entre múltiplos dispositivos ou transmissões em *broadcast*.

Como primeira opção de solução pode-se, após os rádios terem se encontrado a partir do uso de sequência de salto de canais, adotar um canal de controle comum aos nós da rede. A partir do canal de controle, informações sobre as comunicações existentes poderiam ser informadas, o que auxiliaria os rádios. Porém, este tipo de solução apresenta deficiências. A principal está relacionada ao problema de gargalo do canal de controle, que ocorre quando vários rádios transmitem pelo canal ao mesmo tempo.

Da literatura, o trabalho [2] usa uma abordagem para transmissão de mensagens de *broadcast* em um cenário multissalto e multiusuário. Neste trabalho, os autores propõem a ideia de aplicar diversos encontros entre pares de USs, de forma a conseguir um *rendezvous* entre todos os nós da rede. Dois USs após terem se encontrado, em um canal comum disponível para ambos, trocam os parâmetros que a solução utiliza para criar a sequência de salto de canais, neste caso uma tupla de 3 valores. Rádios que utilizam os mesmos parâmetros (a mesma tupla) criam sequências de salto de canais idênticas. Além disso, existe um critério de comparação e desempate entre as tuplas. Quando dois USs se encontram, o usuário que possui a tupla com menores parâmetros a atualiza para a tupla com parâmetros maiores. Logo,

dois rádios que se encontram, após a aplicação do critério de desempate, passam a utilizar a mesma sequência de salto de canais. Este processo é repetido até que todos os rádios passam a seguir a mesma sequência originada pela tupla de valores maiores. Por fim, já que todos os nós utilizam a mesma sequência, uma transmissão em *broadcast* pode ser realizada.

Uma das desvantagens desta solução é o tempo de convergência para a sequência única. Pode-se demorar a convergir, uma vez que não se sabe qual nó possui a tupla de valores maiores, nem a ordem de encontro entre os nós. Assim, alguns nós podem ter que atualizar sua sequência mais que uma vez, visto que podem encontrar outro nó que possua parâmetros de tupla maiores do que a do último nó encontrado até então.

De outra forma, pode-se pensar em transmissões em *broadcast* periódicas. Assim, pode-se definir uma sequência de salto de canais comum entre todos os dispositivos. Em seguida, os dispositivos mudam temporariamente em intervalos regulares de tempo para essa sequência. Os rádios que possuem informações de controle entram em modo de transmissão, enquanto os demais permanecem em modo de recepção.

Esta alternativa possui desvantagens, relacionadas com a definição do intervalo de tempo estipulado para estas transmissões em *broadcast* periódicas. Intervalos muito curtos iriam resultar em um desperdício de largura de banda, enquanto intervalos mais longos resultariam em uma maior latência para realizar a transmissão. Outra desvantagem é a contenção que pode ocorrer quando dispositivos diferentes precisam realizar uma transmissão e estão no mesmo canal.

Por estas razões, consideramos que transmissões em *broadcast* realizadas sob demanda são mais convenientes. Para tal, uma forma de se proceder é transmitir repetidamente uma mesma mensagem de *broadcast*, por todos os *slots* de uma sequência, a fim de realizar vários encontros com os nós vizinhos de um modo “um para um” [33], com o objetivo de emular um *rendezvous* de “um para n ”.

Neste capítulo, propomos um mecanismo para executar transmissões em *broadcast* em uma rede de rádios cognitivos que usa sequência de salto de canais. Este mecanismo visa facilitar transmissões em *broadcast* no modo “um para um”, levando em conta as condições heterogêneas do espectro entre os usuários secundários e o uso dinâmico do espectro realizado pelos usuários primários. O mecanismo proposto tem como principal característica o fato de ser independente, ou seja, ele pode ser utilizado em combinação com qualquer protocolo baseado em sequência de salto de canais.

Antes de apresentarmos o mecanismo, apresentaremos a seguir a formalização de uma sequência de salto de canais e a formalização de uma sequência de *broadcast*.

4.2 Formalização da Sequência de Salto de Canais e Sequência de *Broadcast*

Formalmente, consideramos que a rede de rádios cognitivos consiste de K ($K \geq 2$) rádios coexistindo com UPs na mesma área geográfica. Além disso, o espectro de frequências licenciado pode ser dividido em M ($M > 1$) canais ortogonais que não se sobrepõem. Assumimos que esses canais são unicamente indexados, como: $0, 1, 2, \dots, M - 1$ e os índices são bem conhecidos por cada usuário da rede [15].

O conjunto de potenciais canais disponíveis aos rádios é indicado por:

$$\mathcal{C} = \{c_0, c_1, \dots, c_n, \dots, c_{M-1}\}$$

Onde c_n indica o n -ésimo canal que usualmente chamamos de canal n por conveniência ($n = 0, 1, 2, \dots, M - 1$). Cada rádio é equipado com uma única interface, que pode acessar canais em \mathcal{C} de maneira oportunista.

Devido à atividade de UPs, nem sempre todos os canais em \mathcal{C} estarão livres. Assim, o conjunto $\mathcal{C}_i \subseteq \mathcal{C}$ indica os canais disponíveis para o rádio i , ($i = 1, 2, \dots, K$).

Além disso, o sistema foi modelado como **assimétrico**, ou seja, os canais livres podem não ser os mesmos para todos os rádios. Desta forma, dois rádios na mesma rede podem ter canais disponíveis diferentes, o que por vezes é dito como ter “**visões heterogêneas**” do espectro de frequências. Assim, pode ocorrer de: $\mathcal{C}_i \neq \mathcal{C}_j$, para $1 \leq i, j \leq K$.

O tempo foi particionado em *slots*, e a cada *slot* cada rádio salta para um canal seguindo uma sequência \mathbf{s} , na tentativa de se encontrar com seus potenciais vizinhos.

Formalmente uma sequência de salto de canais \mathbf{s} de período T *slots*, pode ser definida como [17]:

$$\mathbf{s} = \{s_0, s_1, \dots, s_i, \dots, s_{(T-1)}\}$$

Onde $s_i \in [0, M - 1]$ representa o índice do canal na sequência \mathbf{s} no i -ésimo *slot* da sequência de salto de canais.

Dadas duas sequências \mathbf{u} e \mathbf{v} de período T , se $\exists i \in [0, T - 1]$ tal que $u_i = v_i = h$ onde $h \in [0, M - 1]$, dizemos que \mathbf{u} e \mathbf{v} podem se encontrar no i -ésimo *slot* no canal h . Desta maneira o i -ésimo *slot* é chamado de *slot de rendezvous* e o canal h é chamado de canal de *rendezvous* entre \mathbf{u} e \mathbf{v} .

Na prática, conforme apresentado no Capítulo 2, a realização do *rendezvous* e o estabelecimento de um enlace de comunicação envolve mais processos em um intervalo de um *slot*, incluindo o recebimento de *beacons* no canal disponível, aplicação de mecanismo de *handshaking*, etc. Por motivos de foco na questão da sequência de salto de canais, assumimos que o *rendezvous* é dito completo quando dois ou mais

rádios saltam para o mesmo canal disponível, estão no alcance um do outro e, o canal está livre de transmissões de UPs.

A partir da sequência de salto de canais de todos os nós vizinhos de um salto de um nó, queremos construir uma sequência chamada sequência de *broadcast*. A sequência de *broadcast* será utilizada quando um nó, chamado *broadcaster*, desejar enviar uma mensagem de controle para seus vizinhos de um salto. Como, a princípio, cada nó possui sua própria vizinhança, assim cada nó terá sua própria sequência de *broadcast*.

Desejamos que a sequência de *broadcast* possua sobreposição, com todas ou com a maior quantidade de sequências de salto de canais dos nós vizinhos de um salto do nó emissor da mensagem de *broadcast*.

Assim como em [15], chamamos de $I_j(\mathbf{u}, \mathbf{v})$ a função que indica quando o canal j é um canal de *rendezvous* entre duas sequências \mathbf{u} e \mathbf{v} , isto é:

$$I_j(\mathbf{u}, \mathbf{v}) = \begin{cases} 1, & \text{se } \exists i \in [0, T - 1] \mid u_i = v_i = j, \\ 0, & \text{outros casos} \end{cases}$$

Representando por V_k o conjunto de sequências dos nós vizinhos de um salto do nó k . Temos V_k como:

$$V_k = \{\mathbf{u}, \mathbf{v}, \mathbf{s}, \dots\}$$

Desta maneira, cada índice de V_k indica a sequência de salto de canais de um vizinho de um salto do nó k .

A seguinte função $W_k(\mathbf{b}, \mathbf{w})$ representa a sobreposição de uma sequência \mathbf{b} , que representa uma sequência candidata a ser uma sequência de *broadcast* do nó k emissor com qualquer sequência \mathbf{w} , com $\mathbf{w} \in V_k$, em qualquer canal j .

$$W_k(\mathbf{b}, \mathbf{w}) = \begin{cases} 1, & \text{se } \sum_{j=0}^{(M-1)} I_j(\mathbf{b}, \mathbf{w}) \geq 1 \forall \mathbf{w} / \mathbf{w} \in V_k \\ 0, & \text{outros casos} \end{cases}$$

Se representarmos por $Z_k(\mathbf{b})$ a função que computa o número de sobreposições da sequência \mathbf{b} com todas as sequências contidas em V_k , onde $Z_k(\mathbf{b})$ é dada por:

$$Z_k(\mathbf{b}) = \sum_{\mathbf{w} \in V_k} W_k(\mathbf{b}, \mathbf{w})$$

Idealmente, se

$$Z_k(\mathbf{b}) = |V_k|$$

a sequência \mathbf{b} atinge todos os vizinhos de um salto do nó k , e será a sequência de *broadcast* do nó k . Caso contrário, outra sequência \mathbf{b}' que tiver maior sobreposição com as demais sequências em V_k será a sequência de *broadcast* do nó k .

As funções apresentadas anteriormente ($I_j(\cdot)$ e $W_k(\cdot)$), representam apenas casos

de soluções voltadas para sistemas **síncronos**. Ou seja, soluções que assumem e necessitam de um sincronismo de início de sequência entre as sequências de todos os rádios.

Assim, para representar um sistema **assíncrono**, dada uma sequência \mathbf{u} conforme em [17] usamos a função $rotate(\mathbf{u}, n)$ que indica uma rotação cíclica de n slots através da sequência \mathbf{u} , isto é:

$$rotate(\mathbf{u}, n) = \{\mathbf{v}_g | \mathbf{v}_g = \mathbf{u}_{(g+n) \bmod T}, g \in [0, T - 1]\}$$

Onde n é um número inteiro não negativo. Por exemplo, dado $\mathbf{u} = \{0, 1, 2\}$ e $T = 3$, $rotate(\mathbf{u}, 2) = \{2, 0, 1\}$

Como para o caso **assíncrono**, os nós podem iniciar suas sequências em instantes diferentes, podemos definir o conjunto \mathbf{S} que contém o conjunto de sequências originadas a partir da sequência \mathbf{s} rotacionada por l , conforme a função $rotate(\cdot)$, onde $0 \leq l \leq T - 1$

$$\mathbf{S} = \{rotate(\mathbf{s}, 0), rotate(\mathbf{s}, 1), rotate(\mathbf{s}, l), \dots, rotate(\mathbf{s}, T - 1)\}$$

Desta forma, cada índice de conjunto \mathbf{S} representa a sequência \mathbf{s} rotacionada conforme o valor de l . Pode-se notar que o primeiro índice de \mathbf{S} representa a própria sequência \mathbf{s} , pois está rotacionada de 0 slots.

Podemos redefinir a função $I_j(\mathbf{u}, \mathbf{v})$ a valer 1 quando $\exists i \in [0, T - 1]$ quando $rotate(\mathbf{u}, l) = rotate(\mathbf{v}, f)$ mesmo com $l \neq f$ e $rotate(\mathbf{u}, l) \in \mathbf{U}$ e $rotate(\mathbf{v}, f) \in \mathbf{V}$. Sendo \mathbf{U} e \mathbf{V} o conjunto das sequências \mathbf{u} e \mathbf{v} rotacionadas.

O mesmo para a função $W_k(\cdot)$, pode-se pensar em estar relacionada ao conjunto

$$\mathcal{V}_k = \{\mathbf{S}, \mathbf{U}, \mathbf{V}, \dots\}$$

Este conjunto \mathcal{V}_k representa o conjunto de sequências dos vizinhos de um salto do nó emissor k rotacionadas de qualquer valor l , com $0 \leq l \leq T - 1$, onde $W_k(\cdot)$ irá valer 1 apenas quando existir um casamento de pelo menos um slot entre a sequência \mathbf{b} com todas as rotações de uma mesma sequência de um nó vizinho, em qualquer canal. Para a função $Z_k(\mathbf{b})$ que contabiliza a quantidade de sobreposições entre a sequência \mathbf{b} , candidata a sequência de *broadcast*, temos a mesma abordagem, porém relacionada ao conjunto \mathcal{V}_k .

Para encontramos uma sequência \mathbf{b} de *broadcast* definida anteriormente propomos dois algoritmos, apresentados a seguir.

4.3 Algoritmos para Sequência de *broadcast*

Os algoritmos propostos estabelecem a forma de criação de uma sequência de salto de canais denominada sequência de *broadcast*. Esta sequência possui o objetivo de minimizar o tempo de *rendezvous* (*Time-To-Rendezvous* - TTR) entre o nó *broadcaster* e todos os seus vizinhos de um salto. O mecanismo de uso desta sequência baseia-se na retransmissão da mesma mensagem de *broadcast* por todos os *slots* da sequência de *broadcast*.

Além disso, o mecanismo assume que todos os usuários secundários utilizam um modelo baseado em papéis (*role-based model*) [7], além do mesmo protocolo de sequências de salto de canais, ou seja a mesma solução de *rendezvous*. No modelo de papéis, considera-se que os nós podem assumir diferentes papéis (funções) e em cada papel utilizam-se diferentes sequências de salto de canais, neste caso: funções de *broadcaster* (emissor de mensagem de *broadcast*) ou receptor de mensagens. Assim, os nós ociosos, ou seja, nós que atualmente não têm nada a transmitir, usam uma sequência de salto de canais chamada **sequência de repouso**. Quando um nó deseja transmitir alguma mensagem de *broadcast*, ele muda para a sequência apropriada, e retorna para a sequência de repouso após realizar sua tarefa. Esta sequência de repouso é individual para cada nó e é a sequência determinada pela solução de *rendezvous* utilizada.

Consideramos que cada usuário secundário na rede executa *rendezvous* periódicos de modo “um para um” com todos seus vizinhos de um salto, no qual trocam informações sobre suas sequências de repouso e informações sobre a disponibilidade de canais. Cada nó transmite os parâmetros da solução de *rendezvous* utilizada, de maneira que a sua sequência de repouso possa ser reproduzida pelo nó que recebe estes parâmetros. Os rádios também informam aos seus vizinhos a sua perspectiva de atividade de UPs sobre cada canal. Mais precisamente, os rádios informam a probabilidade de cada canal estar livre de usuários primários (p_{idle}). Esta informação permite que os nós criem sua sequência de *broadcast* de acordo com Algoritmos 2 e 3.

Propomos dois algoritmos que geram uma sequência de *broadcast* a partir das sequências de repouso e informações de disponibilidade de canais, denominados *Greedy Simple* e *Greedy p_{idle}* . O tamanho da sequência de *broadcast* criada, ou seja o seu número de *slots*, é o mesmo da sequência de repouso utilizada pelos USs. O objetivo dos algoritmos é obter uma sequência de *broadcast* que permita que o nó *broadcaster* realize *rendezvous* com o maior número possível de vizinhos de um salto, em um tempo menor se comparado a vários encontros individuais do nó *broadcaster* com cada vizinho. Ambos os algoritmos varrem todas as sequências de repouso dos nós vizinhos do nó *broadcaster*, verificando qual canal deve ser utilizado a cada *slot*

Algoritmo 2: Seleção de sequência de *Broadcast* usando *Greedy Simple*.

```
1 input: RADIOS, nó Broadcaster, SEQSIZE, CANAIS, SEQ(RADIOS)
2 init sequence = { }
3 init reachedNodes =  $\emptyset$ 
4 while ( $|sequence| < SEQSIZE$ ) do
5   selecione canal C onde nó Broadcaster alcança o maior número de
   vizinhos  $\notin reachedNodes$ 
6   acrescente C a sequence
7   adicione todos os vizinhos alcançados a reachedNodes
8   if (reachedNodes == RADIOS) then
9      $reachedNodes = \emptyset$ 
10 retorna sequence
```

da sequência de *broadcast*. Esta verificação indica o canal que permite *rendezvous* com o maior número de nós vizinhos de uma forma “gulosa”. A cada iteração, caso um nó já tenha sido alcançado (encontrado), sua sequência de repouso não é mais levada em consideração. Este processo se repete até que todos os nós sejam atingidos. Se todos os nós são alcançados antes da sequência de *broadcast* atingir o mesmo tamanho da sequência de repouso, o algoritmo tenta mais uma vez se encontrar com todos os nós até que a sequência chegue ao tamanho necessário, quando o algoritmo termina.

Os algoritmos começam a divergir quando existe um empate na escolha de qual melhor canal utilizar no *slot* da sequência de *broadcast*, isto é, quando existem dois ou mais canais que permitem atingir o mesmo número de nós vizinhos. Desta forma, o algoritmo *Greedy Simple* simplesmente desempata pelo canal com o menor número de identificação (ID), enquanto que o algoritmo *Greedy p_{idle}* usa a estimativa de cada vizinho sobre os canais mais ociosos para selecionar o canal que é mais provável de estar livre para a maioria dos nós. A informação de *p_{idle}* indica a estimativa de um canal estar livre de algum UP. Esta estimativa é calculada por cada US depois de feita uma série de amostragens a cada canal, pois antes de uma transmissão o US deve sensorar o canal para evitar uma interferência com algum UP. Assim, esta estimativa reflete a probabilidade de um canal estar livre, onde cada US possui sua própria estimativa para cada canal.

Além disso, no algoritmo *Greedy Simple* um nó é considerado como atingido quando a sua sequência de repouso tem um *slot* de sobreposição com a sequência de *broadcast*, isto é, as duas sequências têm o mesmo canal no mesmo *slot*. Por outro lado, o algoritmo *Greedy p_{idle}* só considera um vizinho como alcançado, e não o leva mais em consideração ao escolher a sequência de *broadcast*, quando a soma do valor de *p_{idle}* do nó para aquele *slot* ultrapassa 1.0.

Quando um nó secundário utiliza uma sequência de *broadcast* para transmitir

Algoritmo 3: Seleção de sequência de *Broadcast* usando *Greedy P_{idle}* .

```
1 input: RADIOS, nó Broadcaster, SEQSIZE, CANAIS,  
   SEQ(RADIOS),  $P_{idle}(nó, canal)$   
2 init sequence = { }  
3 init reachedNodes =  $\emptyset$   
4 for (nó in RADIOS) do  
5   | init  $P_{idle}acumulada(nó) = 0.0$   
6 while ( $|sequence| < SEQSIZE$ ) do  
7   | selecione canal C onde nó Broadcaster alcança o maior número  
   | esperado de vizinhos  $\notin reachedNodes$   
8   | for (nó in vizinhos alcançados) do  
9     |  $P_{idle}acumulada(nó) = P_{idle}acumulada(nó) + P_{idle}(nó, C)$   
10    | if ( $P_{idle}acumulada(nó) \geq 1.0$ ) then  
11    |   | adicione nó a reachedNodes  
12  | acrescente C a sequence  
13  | if (reachedNodes == RADIOS) then  
14  |   | reachedNodes =  $\emptyset$   
15 retorna sequence
```

uma mensagem, ele transmite no dado canal e no determinado *slot* somente se aquele canal estiver livre da atividade de UPs. Caso contrário, o nó aguarda o *slot* seguinte, repetindo este procedimento até que ele percorra toda a sequência de *broadcast*. No entanto, por causa da atividade de UPs, nem sempre a mensagem de *broadcast* atingirá todos os nós caso a sequência seja seguida uma única vez, assim a sequência de *broadcast* pode ser reutilizada repetidamente, num esforço de aumentar as chances de atingir todos os vizinhos do nó *broadcaster*.

4.4 Modelo de Simulação e Resultados

O modelo de simulação implementado emula o comportamento de um nó enviando mensagens de *broadcast* periódicas para seus vizinhos, de acordo com um protocolo baseado em sequência de salto de canais e uso de papéis (*role-based model*). O nó pode ter dois papéis, o papel de *broadcaster* ou receptor de mensagens, onde temos apenas um nó com a função de *broadcaster* a cada rodada da simulação. Por esse motivo, o cenário de simulação, conforme Figura 4.1, consiste em $X + 1$ usuários secundários, com um nó (nó B) transmitindo mensagens de *broadcast* de maneira que seus X vizinhos ajam como nós receptores. Em cada simulação, os X nós receptores utilizam sequências de repouso individuais dadas pelas soluções de *rendezvous*: *modular clock* (*mclock*), *Sequence-based Rendezvous* (*seqr*), *Quorum-based Channel Hopping* (*quorum*) ou *Slotted Seeded Channel Hopping* (*ssch*), onde a forma de

funcionamento e as características das soluções de *rendezvous* implementadas foram apresentadas no Capítulo 2.

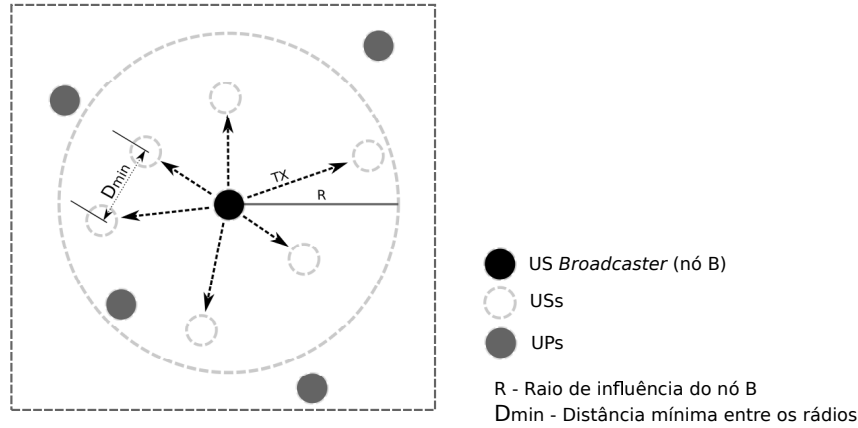


Figura 4.1: Cenário de simulação.

Cada um dos X nós receptores tem sua própria sequência de repouso, a qual é obtida através da seleção de parâmetros iniciais da solução de *rendezvous* em uso na simulação.

Utilizamos um simulador próprio em TCL que gera as sequências de repouso dos nós receptores seguindo a solução de *rendezvous* selecionada, além de criar a sequência de *broadcast* para o nó *broadcaster* de acordo com os dois algoritmos propostos. O simulador avança a passos discretos no tempo, de acordo com os *slots* das sequências de salto de canais, chaveia o nó para o canal indicado, computa a mensagem de *broadcast* a ser enviada e verifica se o canal atual do nó de *broadcast* está em uso por algum UP. Esta verificação impede que o nó *broadcaster* envie alguma mensagem através do canal atual que cause interferência sobre algum UP. Caso o canal esteja ocupado, o nó *broadcaster* aguarda até o próximo *slot* e verifica uma nova oportunidade de transmissão.

Realizamos dois tipos de simulações, conforme o nó *broadcaster* usa ou negligencia a informação privilegiada sobre a sequência de saltos dos nós vizinhos e as disponibilidades de cada canal. Nos resultados, os casos “sem o uso da proposta” indicam a avaliação quando as informações sobre a sequência de saltos dos nós vizinhos são negligenciadas. No início da simulação, assumimos que o nó B já trocou estas informações privilegiadas. Ao utilizar essas informações privilegiadas, o nó B pode criar a sequência de *broadcast* de dois modos diferentes, usando um dos algoritmos propostos na Seção 4.3: *Greedy Simple* (**s**imple) ou *Greedy p_{idle}* (**p**Idle).

Caso contrário, em que este conhecimento prévio é negligenciado, o nó B utiliza para uma transmissão em *broadcast* a mesma sequência dada pelas soluções de *rendezvous*, ou seja, a sua sequência de repouso. Isto é o melhor que o nó B pode fazer sem qualquer conhecimento da sequência de saltos de seus vizinhos, uma vez que

esta abordagem assegura, pelo menos, as boas propriedades do *rendezvous* um a um da solução em uso.

Cada canal é utilizado por um usuário primário, cujo comportamento segue um modelo ON-OFF. A duração de cada estado é distribuída exponencialmente com média de μ_{ON} e μ_{OFF} , respectivamente. Assim, a probabilidade do canal j estar ocioso é dada por $p_{inativo} = \frac{\mu_{OFF,j}}{\mu_{ON,j} + \mu_{OFF,j}}$. Essas probabilidades ociosas são geradas aleatoriamente para cada canal e são distribuídas uniformemente entre 0.0 e 1.0.

Utilizamos o seguinte modelo, a fim de determinar a influência de UPs sobre a comunicação dos nós secundários. Inicialmente, o nó B é posicionado no centro de um círculo de raio R , que representa a sua área de comunicação. Seus X vizinhos são então uniformemente posicionados dentro desse círculo, mantendo uma distância mínima de D_{min} entre si. Em seguida, N nós primários, um para cada canal, são dispostos de forma aleatória dentro de uma área de quadrado concêntrico ao círculo, Figura 4.1. A distância mínima D_{min} também foi mantida entre eles. Todos os nós secundários dentro do alcance R de um determinado nó primário i irão sensorar o canal j com sua correspondente probabilidade de uso e os demais nós vão perceber esse canal com $p_{idle} = 1.0$, ou seja o canal estará livre. Consideramos que cada nó secundário é capaz de estimar a taxa com que UPs utilizam cada canal através de operações periódicas de detecção nos canais, como em [49]. Assim, a estimativa pode ser feita através de sensores periódicos a cada canal, este sensoramento representa um procedimento de amostragem em um canal para descobrir seu estado a cada instante desta amostragem. A amostra corresponde ao valor 1 ou 0 conforme o estado do canal, 1 para ocupado e 0 para livre. Desta forma, o processo produz uma sequência binária para cada canal que representa os períodos de uso do canal por UPs, no caso de 1, e períodos de oportunidades no caso de zeros. Recorre-se a um sensoramento periódico, com um período de sensoramento e tempo de sensoramento definido, onde o tempo de sensoramento é a quantidade de tempo que se permanece no canal para se realizar uma simples medição de modo a permitir um nível de detecção desejável.

Devido à influência dinâmica dos usuários primários, não há garantia de que, mesmo ao final de uma sequência de *broadcast*, todos os vizinhos do nó recebam a mensagem de *broadcast*. Portanto, uma solução possível seria estender o tempo em que o nó B permanece no papel de *broadcaster*. Para avaliar esta ideia, nós introduzimos um novo parâmetro nas simulações chamado de número de sequências de *broadcast*, que indica quantas vezes o nó B irá repetir a sequência de *broadcast*.

Uma métrica importante para avaliar as soluções de sequências de salto de canais, que é medida nas simulações, é a tempo máximo para *rendezvous* (*Maximum Time-To-Rendezvous* - MTTR). Esta métrica é constituída pelo tempo máximo, em número de *slots*, necessário para dois nós alcançarem o *rendezvous*. No caso de *bro-*

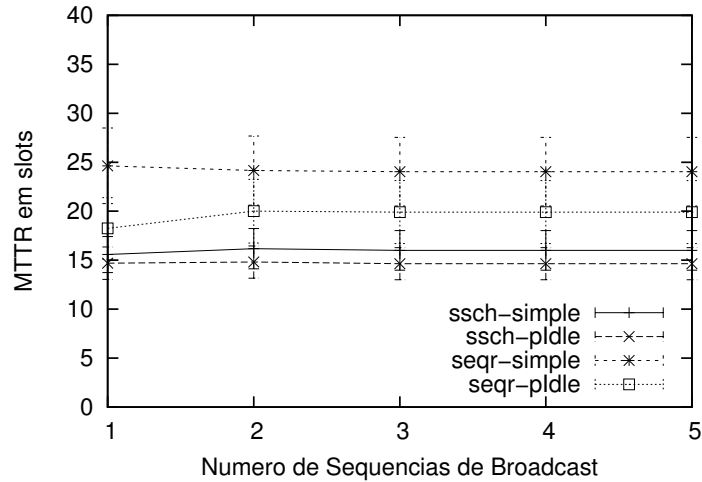
adcast, o MTTR é definido como o tempo máximo que o nó *broadcaster* necessita para encontrar com todos os seus nós vizinhos de um salto em uma simulação. Além do MTTR, também avaliamos o tempo médio para *rendezvous* (*Average Time-To-Rendezvous* - ATTR), que de forma análoga ao MTTR, é definido como o tempo médio necessário para o *rendezvous* do nó *broadcaster* com todos os seus nós vizinhos de um salto.

Uma vez que assumimos que a transmissão de *broadcast* não é totalmente confiável, ou seja, algumas transmissões não chegam a todos os nós vizinhos, apresentamos uma nova métrica para avaliar o *rendezvous* em *broadcast*, nomeada taxa de sucesso. Esta métrica provê o percentual de *rendezvous* em transmissões de *broadcast* em que todos os vizinhos foram encontrados.

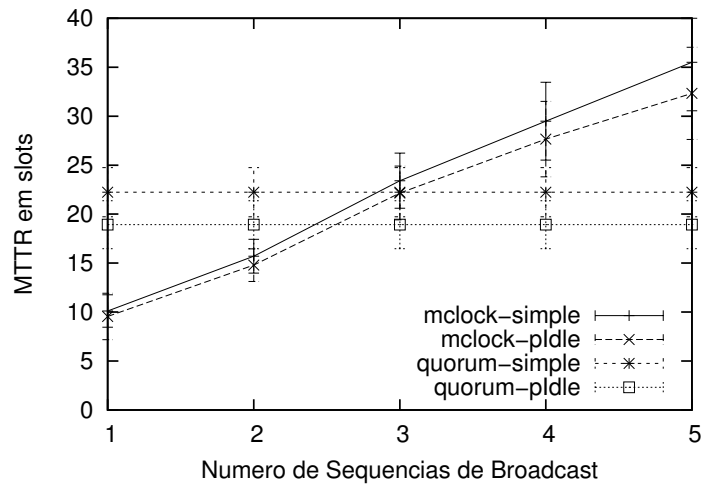
Os resultados da simulação, apresentados nesta seção, foram obtidos em cenários com: $N = 13$ canais, $\mu_{off} = 5$ slots, $R = 250$ metros, e $D_{min} = 20$ metros. Os valores destes parâmetros foram definidos depois de uma série de testes, os quais envolviam tempo de simulação e escolha dos valores de μ_{off} e μ_{on} , pois dentro do tempo de simulação, os valores de μ_{off} e μ_{on} devem ser suficientes para apresentarem uma soma de probabilidade igual a 100%. Variamos o número de sequências de *broadcast* de 1 até 5, e variou-se X (número de vizinhos do nó B) de 2 a 8. Em cada simulação, o nó B envia 1000 mensagens de *broadcast*, ou seja, o nó B assume o papel de *broadcaster* 1000 vezes. Todos os resultados apresentados nos gráficos representam a média de 100 rodadas de simulação com intervalos de confiança de 95%.

As Figuras 4.2(a) e 4.2(b) mostram o desempenho de ambos os algoritmos propostos, `pIdle` e `simple`, para as soluções de *rendezvous* testadas, em termos de tempo máximo para *rendezvous* (MTTR) para chegar a todos os vizinhos de um salto. Pode-se ver que o algoritmo `pIdle` supera o `simple` uma vez que leva em conta a atividade dos UPs. Esta diferença foi observada em todos os resultados obtidos. Por esta razão, a partir de então apresentaremos somente os resultados obtidos com o algoritmo `pIdle` para avaliar o desempenho da proposta. No entanto, pode-se notar que o desempenho de `simple` é próximo do obtido pelo outro algoritmo e tem a vantagem de não requerer a estimativa da ocupação do canal a partir da perspectiva de cada nó vizinho.

Pelos resultados nas Figuras 4.3(a) e 4.3(c), verificamos o MTTR quando as soluções de *rendezvous* utilizam ou não a proposta de sequência de *broadcast*. Pode-se notar que o `seqr` tem os maiores valores de MTTR, mesmo quando apenas uma ciclo de sequência de *broadcast* é utilizado. Seu fraco desempenho deve-se ao fato de que esta solução gera sequências de salto de canais com elevado número de slots, o que acaba gerando tempos de MTTR altos. As demais soluções apresentam valores de MTTR inferiores a 100 slots para todos os parâmetros utilizados.



(a) SSCH e SeqR



(b) Quorum e Modular Clock

Figura 4.2: MTTR para os 2 algoritmos com 8 RCs vizinhos.

As Figuras 4.3(b) e 4.3(d) apresentam os resultados de MTTR quando se utiliza a sequência de *broadcast*. Podemos observar que o desempenho de todas as soluções de *rendezvous* melhoram ao se utilizar o algoritmo proposto. Conforme Figura 4.3(b) o *mclock* apresenta MTTRs piores com o aumento do número de ciclos de sequência de *broadcast*, isto ocorre porque o aumento deste número de ciclos torna a taxa de sucesso de *broadcast* deste mecanismo mais elevada, como pode ser visto na Figura 4.7, o que conduz à inclusão de MTTRs mais elevados nos resultados.

Os resultados da Figura 4.4 mostram o percentual de redução de MTTR alcançado quando se combina a proposta com cada solução de *rendezvous*. Verificamos que o *seqr*, o qual tem o pior desempenho absoluto, quando comparado com os outros esquemas, é o que tem seu desempenho melhorado ainda mais quando combinado com a proposta. Para todas as soluções de *rendezvous* testadas, a menor

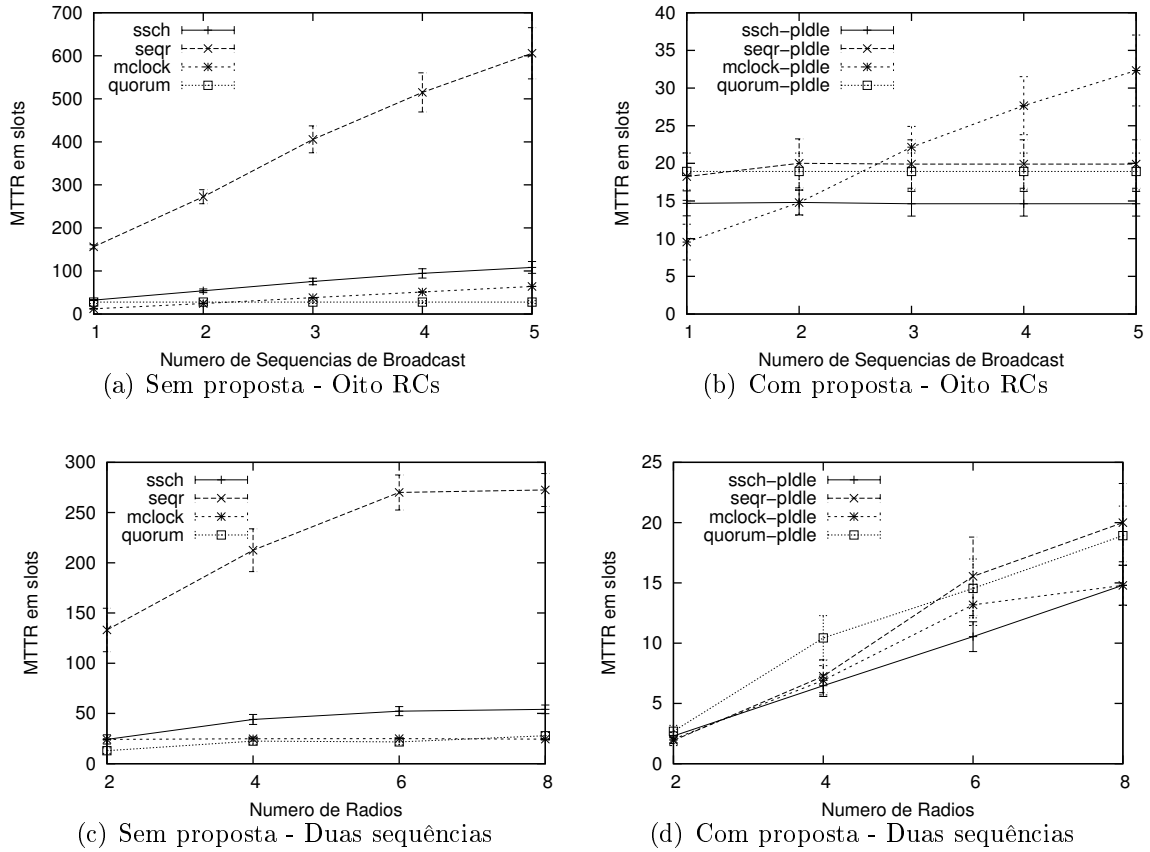


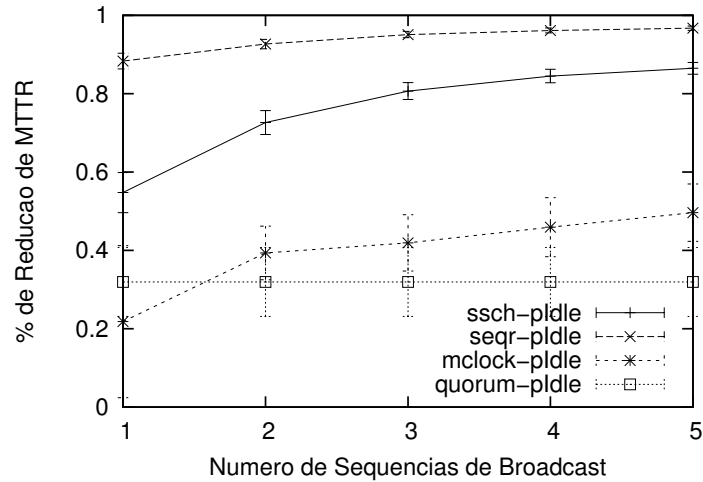
Figura 4.3: MTTR - com e sem uso da proposta.

percentagem de redução foi de cerca de 30%.

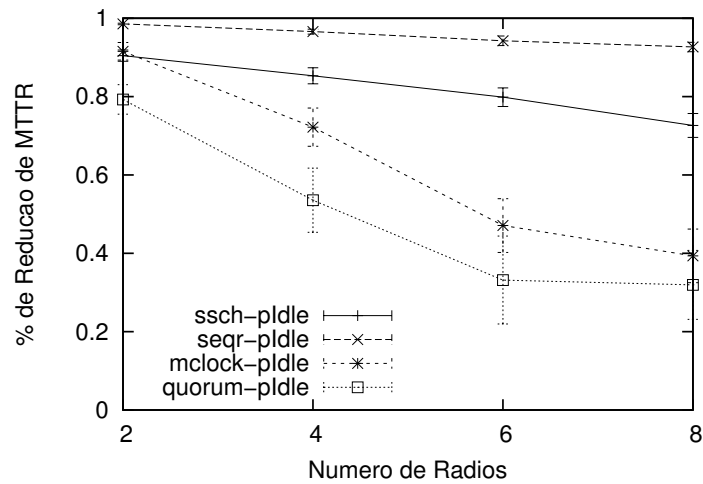
Conforme Figuras 4.5(a) e 4.5(c), temos o tempo médio para o *rendezvous* (ATTR) quando as soluções de *rendezvous* não utilizam a seqüência de *broadcast*. O comportamento mostrado é semelhante ao da Figura 4.3, em que o *seqr* apresenta o pior desempenho, enquanto que os melhores resultados são alcançados pelo *quorum* e *mclock*. Na Figura 4.5(a), pode ser visto que a ATTR para o *mclock* também piora à medida que o número de seqüências de *broadcast* utilizadas aumenta. Isto ocorre pelo mesmo motivo do MTTR, ou seja valores mais altos de ATTR são incluídos nos cálculos.

As Figuras 4.5(b) e 4.5(d), apresentam os resultados do valor de ATTR ao se utilizar a proposta. Pode-se notar que o desempenho de todas as soluções de *rendezvous* são beneficiadas com a utilização do algoritmo proposto. Pela Figura 4.5(b), podemos ver que o *mclock* possui ATTRs mais altos quando o número de ciclos da seqüência de *broadcast* aumenta. Isto acontece porque o aumento do número de ciclos torna a taxa de sucesso do mecanismo de transmissão maior, tal como mostrado na Figura 4.7. Isso faz a inclusão de ATTRs mais elevados nos resultados, o qual apenas considera casos de transmissão bem sucedidos.

Os resultados da Figura 4.6 apresentam o percentual de redução de ATTR cau-



(a) Oito RCs



(b) Duas sequências de *broadcast*

Figura 4.4: Redução de MTTR.

sado pelo uso da proposta juntamente com cada solução de *rendezvous*. Podemos constatar que o **seqr**, que tem o pior desempenho absoluto quando comparado com os outros, e é o que se beneficia mais quando combinado com a proposta. Para todas as soluções de *rendezvous*, o menor percentual de redução foi de cerca de 50%.

As Figuras 4.7(a) e 4.7(c) mostram a taxa de sucesso de *broadcast* dos sistemas de *rendezvous* sem a proposta pIdle. O esquema que apresenta a pior taxa de entrega é o **mclock**, com o seu melhor valor em torno de 80%. O **quorum** tem 100% de taxa de entrega mesmo sem usar nossa proposta. Isto se deve ao fato do esquema ter um elevado grau de sobreposição entre as sequências de cada nó. Todos os esquemas alcançam melhores taxas de sucesso quando se aumenta o número de ciclos de sequências de *broadcast*. Na Figura 4.7(c), observamos que a taxa de sucesso piora com o aumento do número de rádios, pois torna-se mais difícil a entrega da

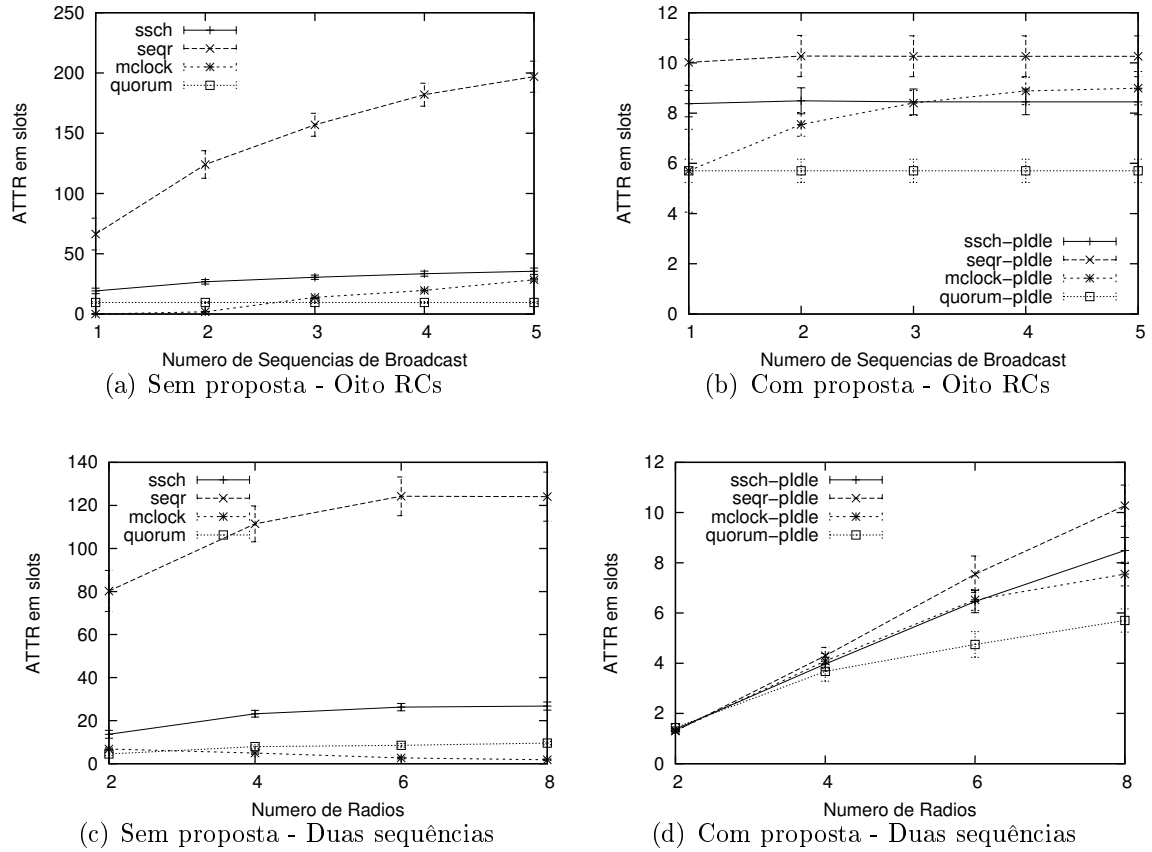


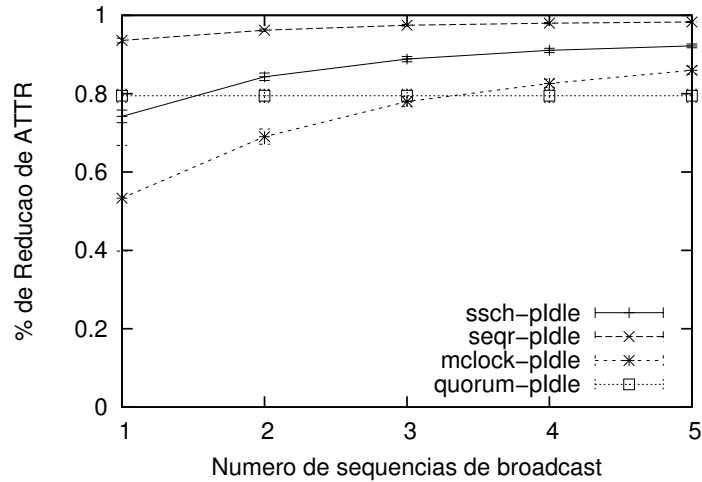
Figura 4.5: ATTR - com e sem uso da proposta.

mensagem de *broadcast* para todos os nós. Mais uma vez, a maior sobreposição entre as sequências do *quorum* permitiu manter uma taxa de entrega elevada, mesmo com mais rádios.

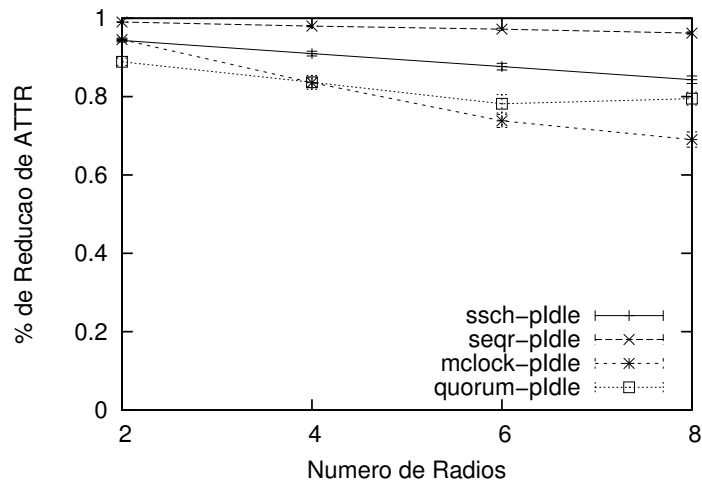
A partir das Figuras 4.7(b) e 4.7(d), temos a taxa de sucesso de *broadcast* em diferentes soluções de *rendezvous*, quando utilizado em combinação com a proposta pIdle. Com exceção do *quorum*, que já apresenta 100% de sucesso sem a nossa proposta, todos os esquemas de *rendezvous* melhoraram consideravelmente. O *ssch* e *seqr* atingiram 100% de taxas de entrega com uso da proposta e mesmo o *mclock*, que não aumentou sua taxa de sucesso para 100%, apresentou melhorias significativas.

4.5 Conclusões do Capítulo

Transmissões em *broadcast* em soluções de *rendezvous* usando salto de canais não é um problema trivial. Quando a comunicação em *broadcast* envolve a divulgação de mensagens de controle em uma rede de rádio cognitivos, o problema é ainda mais difícil devido à atividade dos UPs. Uma solução para este problema seria que todos os RCs utilizassem a mesma sequência de salto de canais (sequência única) no



(a) Oito RCs



(b) Duas sequências de *broadcast*

Figura 4.6: Redução de ATTR.

envio de uma mensagem de *broadcast*. No entanto, esta abordagem de utilizar uma sequência única pode apresentar a desvantagem de contenção no acesso ao meio, que ocorre quando temos uma grande quantidade de tentativas de transmissões simultâneas através do mesmo canal. Este problema de sequência única será melhor verificado no capítulo seguinte. Portanto, uma solução alternativa quando os nós têm sequências diferentes, é o nó *broadcaster* retransmitir mensagens em cada *slot* de sua própria sequência de saltos de modo que alcance todos os nós vizinhos. No entanto, quando a sequência do nó emissor não possui sobreposição com as sequências de seus vizinhos, e a atividade de UPs não é levada em conta, esta estratégia pode proporcionar um atraso até ocorrer o *rendezvous* entre o nó e seus vizinhos, além de adicionar sobrecarga.

Neste capítulo, propomos dois algoritmos para a construção de uma sequência de

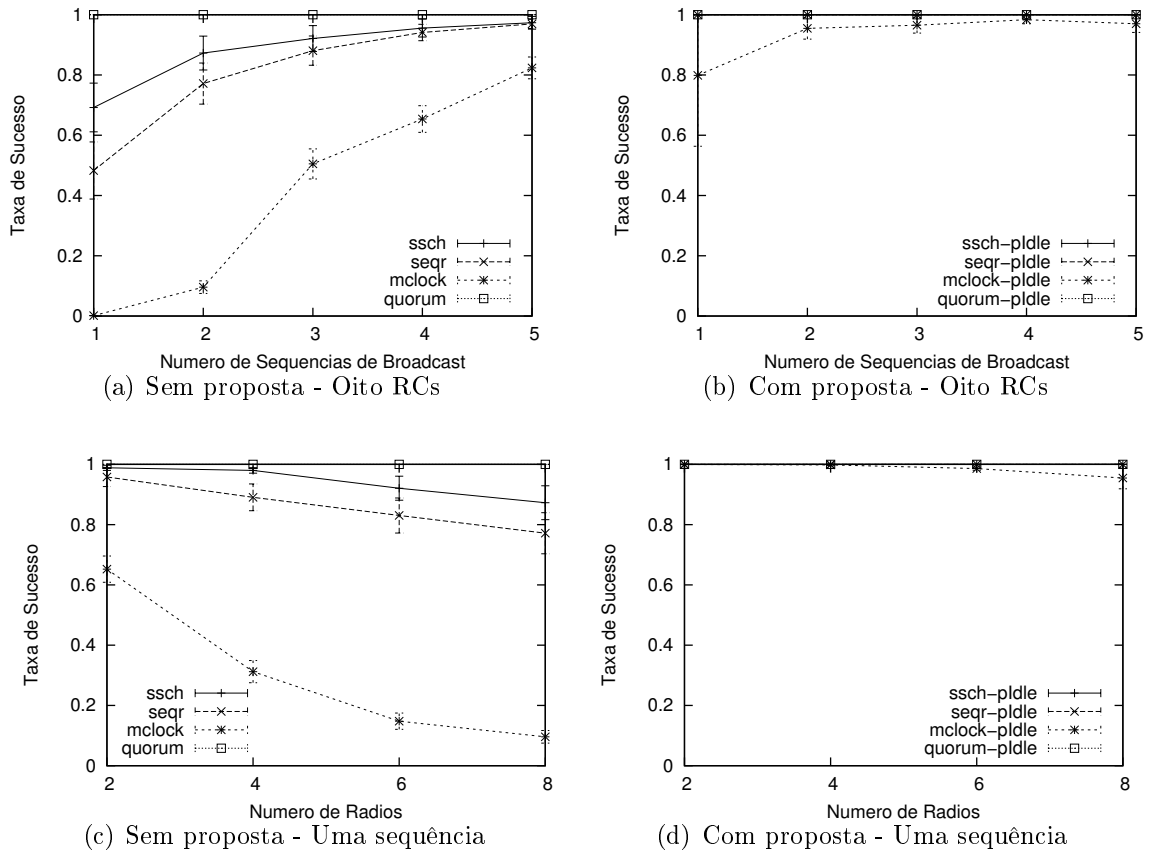


Figura 4.7: Taxa de sucesso - com e sem uso da proposta.

broadcast que leva em conta o número de sobreposição de canais e a atividade de UPs em cada canal. Estes algoritmos têm a grande vantagem de ser ao mesmo tempo simples e independentes da solução de *rendezvous* adotada. Os algoritmos foram avaliados em termos da redução máxima atingida no tempo de *rendezvous*, quando comparado com o caso no qual o nó *broadcaster* utiliza sua própria sequência de saltos não modificada para efetuar a transmissão. A utilização da própria sequência de saltos para transmissão em *broadcast* é o melhor que as soluções da literatura podem fazer.

Os resultados obtidos pelo uso das propostas mostram que ambos algoritmos proporcionaram ganhos de desempenho em termos destas métricas e também em termos do número de vizinhos alcançados, ou seja, a taxa de sucesso.

Capítulo 5

Regras de Funcionamento para uma Rede de Rádio Cognitivos

As soluções para *rendezvous*, em principal as que utilizam sequência de salto de canais, estão voltadas à resolução do primeiro encontro entre os rádios, de forma que o funcionamento das CRNs após este primeiro encontro dos rádios não é discutido na literatura.

Neste capítulo, propomos um conjunto de regras aplicadas aos rádios cognitivos. Este conjunto de regras pode ser visto como um protocolo de acesso ao meio. O objetivo destas regras é aliviar o problema de acesso ao meio e auxiliar no funcionamento de uma rede de rádios cognitivos. Estas regras indicam a cada rádio, sob demanda, o uso de diferentes sequências de canais de acordo com o estado que o rádio se encontra: transmitindo *unicast*, transmitindo *broadcast* ou em modo de recepção. Este conjunto de regras, utiliza a ideia da sequência de *broadcast* apresentada no capítulo anterior.

A proposta foi implementada no NS-2 em um cenário de redes em malha de rádios cognitivos onde cada rádio pôde gerar transmissões em *unicast* e *broadcast*. Esta proposta foi comparada a forma de funcionamento mais intuitiva, que é o caso de uso de um único canal. Pelos resultados pôde-se verificar o problema de gargalo quando este único canal foi utilizado por diversos rádios ao mesmo tempo.

5.1 Motivação para o Conjunto de Regras

Conforme apresentado no Capítulo 2, temos dois grandes grupos de soluções para o problema de *rendezvous* em uma rede de rádios cognitivos. A solução do canal de controle comum (CCC), onde temos um canal reservado para mensagens de controle através do qual os rádios escutam e/ou transmitem mensagens que facilitam o encontro entre os rádios, e outro grupo de soluções que cria uma sequência de saltos

de canais que indica a ordem que os rádios devem chavear pelos canais disponíveis. As sequências de saltos de canais são construídas de forma que permitem o encontro entre dois RCs quaisquer em um tempo finito.

Uma característica importante e comum às soluções de *rendezvous* é que grande parte delas está relacionada a propor uma solução ao primeiro encontro entre os RCs. Trabalhos, principalmente que usam sequência de salto de canais, não comentam ou propõem uma forma de funcionamento da rede posterior ao primeiro encontro. Assim, este atual capítulo aponta uma proposta de um protocolo de acesso ao meio, que indica uma forma de funcionamento dos RCs após o primeiro encontro.

No caso de se utilizar um CCC, a forma mais simples de funcionamento é manter este canal de controle ativo para permitir a transmissão de mensagens de controle importantes para os nós da rede. Porém, esta solução apresenta deficiências, como por exemplo o problema do gargalo do canal de controle [12–14].

Para o caso de se utilizar uma sequência de salto de canais, depois do primeiro encontro temos duas opções: **1)** os nós podem definir um canal de controle o qual passam a utilizar, porém desta forma temos as mesmas deficiências de um CCC ou **2)** mantém-se o uso de sequência de salto de canais.

Dentre as possibilidades, manter o uso de sequência de salto de canais é o mais vantajoso, visto que ao se seguir uma sequência de canais os rádios ficam menos vulneráveis à atividade de UPs, pois uma vez que não permanecem a todo instante em um único canal ficam menos susceptíveis à atividade dinâmica do UPs. Além disso, soluções que criam sequências que não necessitam de sincronismo de início de sequência, soluções assíncronas, possuem propriedades que permitem encontros em tempo finito mesmo que alguns rádios ingressem à rede em tempos distintos. Ou seja, o uso de sequências de salto de canais, como funcionamento de uma CRNs, permite mais facilmente que novos nós ingressem a rede, e desta forma representa uma boa estratégia.

Assim, como uma solução para o funcionamento da rede de rádios cognitivos após o primeiro encontro, pode-se pensar em uma convergência de todos os nós para uma **sequência única**. Esta sequência única pode ser conseguida, por exemplo, a partir da solução para transmissões de mensagens de *broadcast* proposta em [2]. Nesta solução, os nós convergem para uma sequência única, utilizando um esquema de comparação e desempate entre os parâmetros necessários que geram a sequência de saltos de canais. Desta maneira, após um encontro, os rádios envolvidos seguem a mesma sequência de salto de canais, neste caso, o rádio que perdeu na comparação de parâmetros passa a utilizar os parâmetros do nó que venceu. Ao fim, após todos os nós terem se encontrado, todos seguem a mesma sequência de salto de canais, pois passam a utilizar o mesmo conjunto de parâmetros que dá origem a sequência.

O tempo de convergência para esta sequência única pode ser grande, mas uma

vez conseguida a convergência, esta tende a se manter, a menos que um novo nó ingresse na rede, e possua parâmetros (relacionados à solução de sequência de saltos adotada) maiores que os atuais que geraram a sequência em uso corrente.

A solução de usar uma sequência única, doravante denominada de **solução intuitiva**, tem como vantagem permitir o envio imediato de uma mensagem a todos os nós. Além disso, possui a vantagem de facilitar o ingresso de novos rádios e mantê-los mais imunes à atividade de UPs. No entanto, o uso dessa sequência única se aplicada a todas as transmissões *unicast* da rede pode provocar o problema de contenção no acesso ao meio, o que pode limitar a capacidade da rede.

Podemos definir que as transmissões *unicast* não necessariamente serão realizadas através desta sequência única, porém se imaginarmos esta sequência única como uma espécie de “canal de controle lógico”, as transmissões *unicast* podem ser informadas aos demais rádios através desta sequência única, o que pode provocar problema de contenção ao meio.

Assim, propomos uma forma de funcionamento da rede, que pode ser vista como um protocolo de acesso ao meio, aplicada após já ter ocorrido o *rendezvous* de um nó com seus vizinhos de um salto. Devido às características relevantes apresentadas vindas do uso da sequências de salto de canais, definimos que os nós permanecem utilizando a sequência de salto de canais adotada para o primeiro encontro. A **solução proposta** utiliza a ideia dos nós assumirem diferentes papéis (*roles*) no decorrer do tempo, os quais são associados ao uso de diferentes sequências de saltos de canais. Assim, existe um conjunto de regras que, de acordo com o estado do nó, seja em modo de transmissão de mensagens de *unicast*, transmissão de mensagens de *broadcast* ou em modo de recepção, instruí o nó a que sequência de salto de canais utilizar. Estas regras podem estar contidas no próprio *firmware* dos rádios, do mesmo modo que os algoritmos de formação das sequências de salto de canais.

Nos resultados, e daqui por diante, quando nos referenciarmos à **solução proposta** queremos indicar o uso deste conjunto de regras proposto e **solução intuitiva** ao caso de se utilizar uma sequência única.

A **solução proposta** e a **solução intuitiva** foram implementadas no simulador NS-2, onde foi necessário incluir: um modelo de interferência físico baseado na SINR (*Signal-to-Interference-plus-Noise Ratio*) [50], um novo agente gerador de tráfego em *broadcast*, a possibilidade dos nós saltarem de canal segundo diferentes sequências de saltos especificadas, soluções de *rendezvous* para a geração dessas sequências, o algoritmo (*Greedy p_{idle}*) apresentado no Capítulo 4 para a geração da sequência de *broadcast*, e a coleta das métricas de desempenho para a comparação entre as duas soluções. A seguir, apresentamos o conjunto de regras do protocolo de acesso ao meio proposto.

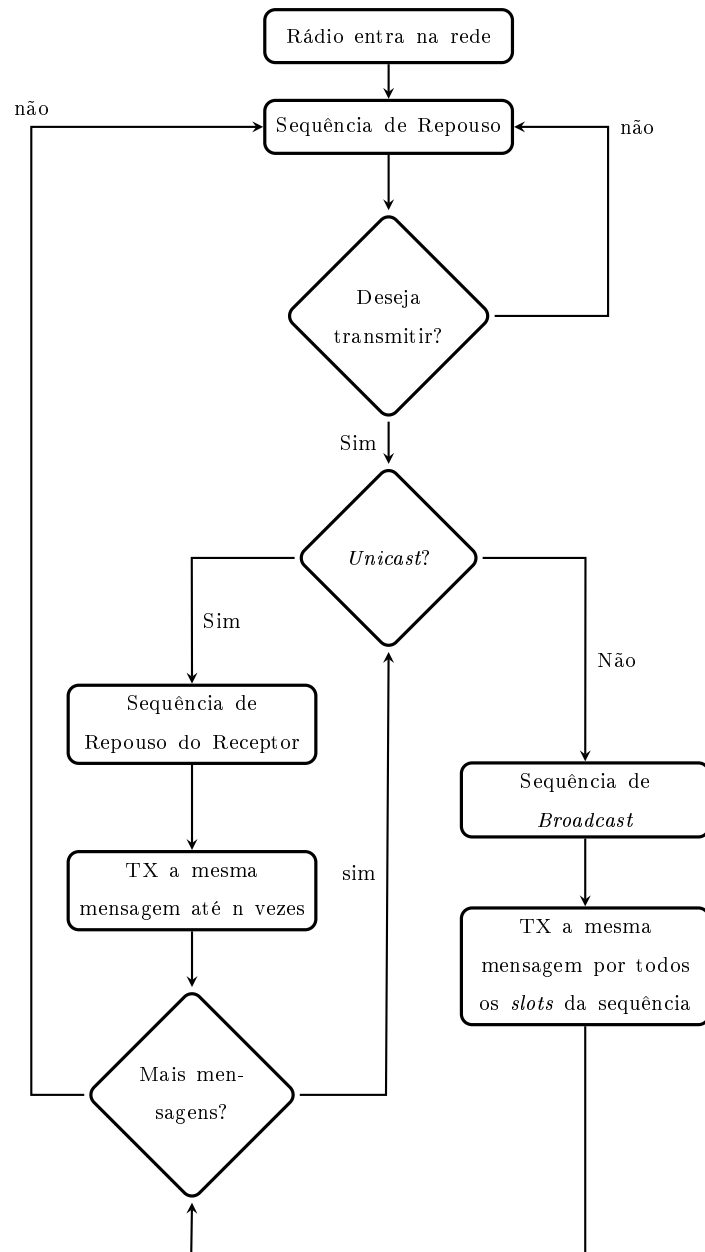


Figura 5.1: Diagrama de funcionamento para uma rede de rádios cognitivos.

5.2 Regras de Funcionamento

O uso do mecanismo de saltos de canais pelos USs consiste em todos os nós saltarem entre canais, evitando aqueles canais ocupados pelos UPs, seguindo uma sequência de saltos definida por uma solução de *rendezvous*. Toda vez que ocorre um encontro entre dois nós, mensagens de controle ou de dados podem ser trocadas entre eles.

Como explicado anteriormente, a maioria das soluções de *rendezvous* propostas na literatura não propõem uma forma de funcionamento depois do primeiro encontro.

Assim a forma de funcionamento proposto, apresentada na Figura 5.1, consiste na definição de uma política de papéis/estado, no qual cada nó mapeia o uso de

uma sequência diferente em decorrência do seu objetivo. Desta forma, definimos três possíveis estados para o nó: repouso ou **modo de recepção**, **modo de transmissão em *unicast*** e **modo de transmissão em *broadcast***. De acordo com o estado de funcionamento do nó, ele decide por uma sequência mais apropriada que evite ou cause menos competição no meio. Assim, o nó utiliza a sequência denominada **sequência de repouso** quando não possui nada a enviar, ou seja, em modo de repouso. A **sequência de repouso** é aquela gerada pela solução de *rendezvous* utilizada. Quando o nó deseja enviar uma mensagem de *broadcast*, ele utiliza a sua **sequência de *broadcast***, quando deseja enviar uma mensagem *unicast* o nó transmissor chaveia para a **sequência de repouso** do nó de destino.

O problema de transmissões em *broadcast* pode ser resolvido conforme apresentado no Capítulo 4, onde seleciona-se de maneira “gulosa” os canais utilizados de forma a relacionar o conhecimento das sequências dos nós vizinhos e a estimativa de uso do canal pelos UPs.

A intenção desta estratégia baseada em papéis é aliviar o acesso ao meio, através da atribuição de sequências distintas. Diferentemente do caso de uma única sequência, cada nó utilizará uma sequência de *broadcast* própria originada a partir de sua vizinhança. Outra vantagem é de não criar a necessidade de uma negociação prévia pacote a pacote para uma transmissão *unicast*, pois na necessidade de envio de uma mensagem deste tipo, o nó transmissor chaveia para a sequência de repouso do nó receptor desejado, sequência esta esperada e válida para o modo de recepção. Esta abordagem reduz a necessidade de troca de informações, ou seja, mensagens de controle para um agendamento de alguma transmissão futura entre o transmissor e receptor.

A seguir, apresentamos o modelo do sistema utilizado e os resultados da comparação de desempenho entre a nossa proposta e a solução intuitiva.

5.3 Modelo do Sistema

No modelo de rede de rádios cognitivos adotado neste trabalho, consideramos um cenário com N_s USs e N_p UPs, ambos utilizando uma faixa do espectro licenciado composta por \mathcal{C} canais, onde $\mathcal{C} = \{c_0, c_1, \dots, c_n, \dots, c_{M-1}\}$ conforme apresentado no Capítulo 4. Os USs acessam os \mathcal{C} canais utilizando técnicas de salto de canais, onde o tempo é dividido em *slots* de duração τ . Durante cada *slot*, um único canal c_i ($i = 1, 2, \dots, M - 1$) é utilizado pelo USs, o qual mantém sua única interface de rádio sintonizada neste canal. A duração τ de um *slot* é dada pelo tempo necessário para a transmissão de um pacote de dados seguida pela mensagem de confirmação ACK. As mensagens de dados enviadas pelos USs possuem tamanho fixo.

Neste modelo, assumimos que os UPs apenas mudam de estado nas transições

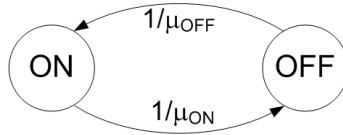


Figura 5.2: Modelo de atividade nos canais.

entre os *slots* e não trocam de canal durante sua operação. Assim, modelamos o padrão de atividade de cada canal $c_i \in \mathcal{C}$ de acordo com um modelo de transição de dois estados, como apresentado na Figura 5.2. Os estados **ON** e **OFF** representam respectivamente os períodos de tempo em que o canal está ocupado, ou seja, com atividade de UPs, ou livre (UPs ociosos). Este modelo **ON-OFF** é amplamente adotado na literatura e é uma boa aproximação para muitos cenários e tecnologias legadas [51].

Os canais utilizados em cada *slot* de tempo pelo USs são representados pela sequência de saltos de canal $\mathbf{s} = \{s_0, s_1, \dots, s_i, \dots, s_{(T-1)}\}$, onde T é o tamanho da sequência. Assumimos que os USs estão sincronizados, ou seja, as trocas de canal são simultâneas e as sequências de saltos estão alinhadas, o que indica um sincronismo de *slots*. Esta afirmação representa um detalhe de implementação o qual poderia ser diferente caso aplicássemos a ideia de duplicar o tempo τ , conforme apresentado no Capítulo 2.

A comunicação entre dois USs se torna possível quando estes estão no mesmo canal, apesar de utilizarem sequências distintas, no mesmo instante e alcance e não existir UP ativo no canal.

A fim de evitar colisões de pacotes, utilizamos um mecanismo de controle de acesso ao meio do tipo CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) aplicados aos USs. Assim, no início de cada *slot*, os USs que desejam transmitir devem escutar o meio livre por um intervalo de tempo escolhido aleatoriamente. Caso o meio permaneça livre durante todo o intervalo, a transmissão é realizada. Caso outra transmissão seja detectada durante este intervalo, o US adia a transmissão do pacote para o próximo *slot*, onde o transmissor novamente disputará o acesso ao meio, conforme Figura 5.3.

No caso de transmissões de pacotes *unicast*, o US transmissor aguarda por um reconhecimento (ACK) do destinatário. Caso o reconhecimento não chegue, o US disputará novamente o meio nos próximos *slots* para realizar novas tentativas de retransmissão do pacote, até que um certo número de tentativas seja atingido. Neste caso, o pacote será descartado. Nenhum controle de erro é usado na transmissão de pacotes em *broadcast*. Em resumo, utilizando este modelo CSMA/CA em *slots*, os USs disputam o acesso aos \mathcal{C} canais para a transmissão de pacotes em *unicast* e *broadcast*. Para a transmissão destes dois tipos de mensagem, os USs comutam

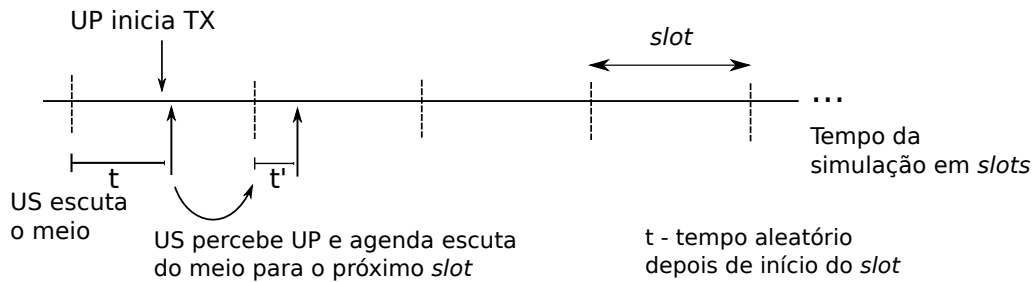


Figura 5.3: Controle de acesso.

entre papéis, de acordo com o funcionamento descrito.

5.4 Ambiente de Simulação

O simulador utilizado para a avaliação e comparação das soluções foi o NS-2, sendo necessário, para isto, implementar um conjunto de extensões. Primeiramente, um modelo de interferência físico baseado na SINR (*Signal-to-Interference-plus-Noise Ratio*) [50] foi implementado para ser usado com a camada MAC 802.11 padrão do simulador. Este modelo permite computar a interferência acumulada sofrida por uma transmissão, com a qual é possível calcular a SINR de um determinado quadro e, por conseguinte, a probabilidade dele ser recebido com sucesso.

Em seguida, a camada MAC 802.11 foi modificada para permitir que os nós trocassem de canal a cada *slot*, segundo diferentes sequências de salto de canais, e se comportassem de acordo com o modelo CSMA/CA em *slots*, como detalhado na Seção 5.3.

Para a geração das sequências de saltos, algumas soluções de *rendezvous* foram implementados (*quorum*, ETCH, *mclock*, etc.), incluindo o algoritmo proposto para a geração da sequência de *broadcast*. Implementou-se também um novo agente gerador de tráfego em *broadcast*, que gera pacotes periódicos com intervalos de tempo $\mu \pm jitter$. O valor de *jitter* foi acrescentado com a finalidade de diminuir a quantidade de rádios que tenham mensagem de *broadcast* a enviar no mesmo instante. Os pacotes em *broadcast* gerados são recebidos pelo próprio agente, o qual coleta e armazena estatísticas por vizinho a respeito dos pacotes recebidos.

O cenário utilizado nos experimentos foi uma rede em malha sem-fio de múltiplos saltos. Para a geração das topologias foi utilizado um modelo de posicionamento que garante a conectividade entre os nós, evitando que o grau máximo entre todos os nós ultrapasse um determinado valor. Assim, seguindo esse modelo, um nó ao ser disposto aleatoriamente em uma determinada área, somente será mantido se estiver dentro da área de cobertura de pelo menos um outro nó e obedecer o grau máximo de vizinhança estipulado.

Os usuários primários presentes nos cenários têm a atividade regida por um modelo ON-OFF, assim como descrito na Seção 5.3. O tempo médio de permanência nos estados ON e OFF foram calculados através de variáveis aleatórias exponenciais com médias μ_{ON} e μ_{OFF} , respectivamente. Os UPs possuíam um raio de influência, denominado R_I . Usuários secundários dentro do raio de influência de algum UP ficam impossibilitados de transmitir e receber pacotes naquele canal enquanto o mesmo encontra-se no estado ON. Vale destacar que, neste modelo, a probabilidade do canal estar livre para um US é dada por $p_{Idle} = \frac{\mu_{OFF}}{\mu_{ON} + \mu_{OFF}}$.

Todos os nós da rede em malha sem-fio geram tráfego *broadcast* de acordo com uma determinada carga. Vale ressaltar que esse tráfego é de apenas um salto, ou seja, ele é gerado a partir de um nó para todos os seus vizinhos, não sendo reencaminhado por esses nós receptores. Da mesma forma, o tráfego *unicast* é gerado por todos os nós, tendo como destino um dos vizinhos de cada nó, o qual é escolhido aleatoriamente. Isto devido a criação da sequência de *broadcast* levar em conta os nós vizinhos de um salto e, à princípio, o nó tem conhecimento apenas das sequências de repouso de seus vizinhos de um salto.

O tráfego gerado foi do tipo CBR (*Constant Bit Rate*), cuja carga pôde ser controlada pela variação do intervalo entre a geração dos pacotes. Como todo o tráfego gerado é de apenas um único salto, os resultados não são influenciados pelo protocolo de roteamento utilizado, mas apenas pela carga imposta à rede, a qual é variada durante a avaliação de desempenho para ambos os tráfegos.

5.5 Resultados da Simulação

Em todas as simulações realizadas, 25 USs foram dispostos em uma área quadrada de 1000 metros de lado, seguindo o modelo de posicionamento descrito na seção anterior. O grau de vizinhança máximo utilizado para a geração das topologias foi igual a 6 vizinhos por nó. Testamos esta geração com outros valores de grau, porém durante a geração da disposição dos nós pela área existe um teste de validação sobre o posicionamento, e quando o cenário gerado falha na verificação, uma nova tentativa de disposição é realizada, caso ocorra um número de tentativas sucessivas sem sucesso (parametrizado através do *script*) o cenário até então montado é abandonado e volta-se ao início das tentativas. Valores de grau de vizinhança maiores implicaram em mais tentativas ou mesmo casos onde o número de tentativas máximo foi atingido sem resultado. O grau igual a 6 pela área quadrada de 1000 metros de lado com 25 nós atendeu ao teste de posicionamento de maneira satisfatória.

O número de UPs foi igual ao número de canais disponíveis. Cada UP foi disposto aleatoriamente dentro da área, e possuiu um raio de influência R_I igual a 250 metros, este valor de raio foi definido pois, para a área do cenário, possibilitou casos que

nem todos os USs estivessem sobre o alcance de todos os UPs. O modelo ON-OFF de atividade dos UPs foi parametrizado pela fixação do parâmetro μ_{OFF} em 20 *slots* e a posterior escolha aleatória da probabilidade de cada UP estar ocioso (p_{Idle}) dentro do intervalo $(0, 1)$. Com estes valores foi possível calcular o parâmetro μ_{ON} de cada UP através da fórmula apresentada na seção anterior. A escolha do tempo de μ_{OFF} foi feita em casamento com a quantidade de *slots* definida para o tempo total de simulação, de modo que a soma da probabilidade para os estados de OFF e ON resultassem em 100%.

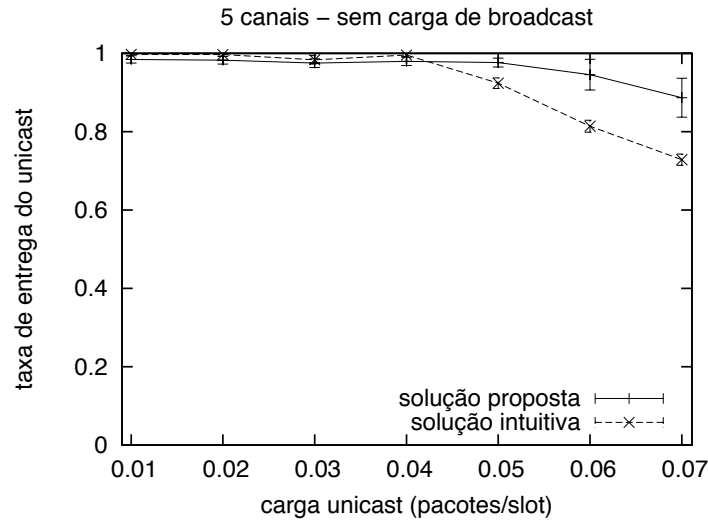
Cada experimento foi executado por 50000 *slots* de tempo no qual avaliamos a taxa de entrega do tráfego *unicast* e a taxa de entrega e o atraso, em número de *slots*, do tráfego de *broadcast*. Essas métricas de desempenho estão em função da carga *unicast*, pelo número de canais disponíveis para os USs. Cada valor apresentado nos gráficos equivale à média de 30 rodadas, onde as barras de erro correspondem ao intervalo de confiança de 95%.

As sequências de saltos de canais utilizadas na solução proposta foram atribuídas a partir da solução *mclock* [7]. Outras soluções, como *quorum* e ETCH, foram utilizados nas simulações, porém os resultados foram muito próximos, e por isso foram omitidos.

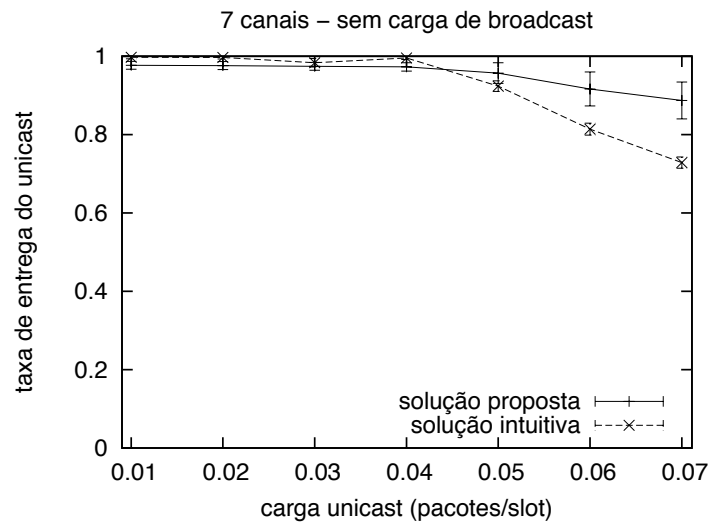
A carga de tráfego *unicast* é mostrada na abscissa de todos os gráficos em termos do número de pacotes por *slot*, esta carga foi variada de 0.01 até 0.07. Esta carga está relacionada à taxa de pacotes criada por cada nó de acordo com o tempo de simulação do experimento. Valores de carga maiores que 0.07 implicaram em um tempo maior de simulação, por isso limitamos a carga até 0.07. A carga do tráfego *broadcast*, μ , assumiu três valores distintos: 0 (sem carga), 500 (alta carga) e 1000 (média carga). Esses valores correspondem ao intervalo entre geração de pacotes *broadcast*, em número de *slots*. O valor do *jitter* somado ao intervalo μ foi sorteado no intervalo $[-0.05\mu, +0.05\mu]$, para aliviar a situação de dois ou mais USs gerarem pacotes de *broadcast* no mesmo instante.

Os gráficos da Figura 5.4 mostram a taxa de entrega do tráfego *unicast* para sequências de 5 e 7 canais quando nenhum tráfego *broadcast* foi gerado. Os resultados demonstram que a solução proposta apresenta um desempenho superior com o aumento da carga, pois consegue aliviar o compartilhamento no meio pelo fato de cada nó utilizar diferentes sequências de canais a cada instante. Neste caso, como não tínhamos tráfego de *broadcast*, todas as sequências utilizadas correspondem às sequências de repouso dos nós. O aumento do número de canais de 5 (Fig. 5.4(a)) para 7 (Fig. 5.4(b)) teve pouca influência nesse resultado. Geramos gráficos também com 11 canais, porém seus valores tiveram a mesma tendência dos resultados de 5 e 7 canais, por isso o omitimos.

Nos próximos dois conjuntos de resultados (Figuras 5.5 e 5.6), analisamos a



(a) 5 canais

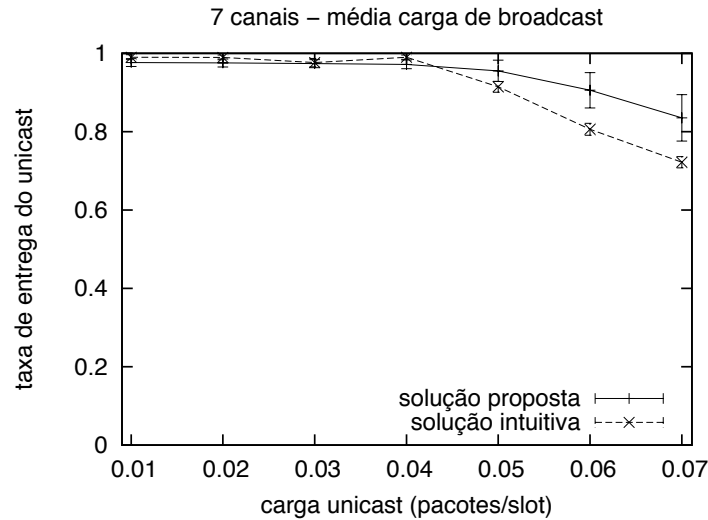


(b) 7 canais

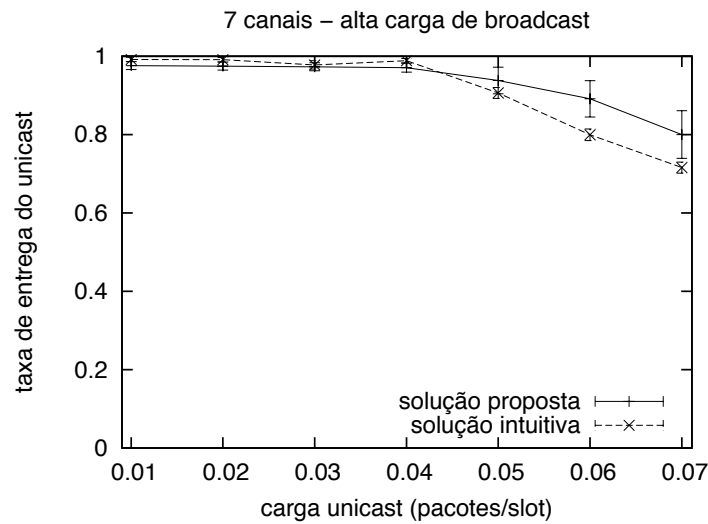
Figura 5.4: Taxa de Entrega do Tráfego *Unicast* sem Carga de *Broadcast*.

influência do tráfego *broadcast*, em conjunto com o tráfego *unicast*. Com relação ao tráfego *unicast* (Figura 5.5), podemos notar que o acréscimo do tráfego *broadcast* teve um impacto reduzido na taxa de entrega do primeiro. No entanto, podemos ver que a solução proposta foi afetada. Isso se deve ao fato de uma única mensagem de *broadcast* ter de ser retransmitida em todos os *slots* da sequência de *broadcast*, o que não acontece no caso da solução intuitiva. Esta característica também afeta a recepção de pacotes *unicast*, já que vizinhos de um US têm maior dificuldade de encontrá-lo em sua sequência de repouso quando ele está utilizando a sequência de *broadcast*. No entanto, de maneira geral a solução proposta continua sendo superior à solução intuitiva nessa métrica pelo fato de diminuir o compartilhamento do meio nos diversos canais, aumentando a capacidade da rede em escoar o tráfego gerado.

Pelos gráficos das Figuras 5.7(a) e 5.7(b), podemos verificar que, apesar da so-



(a) 7 canais - média carga

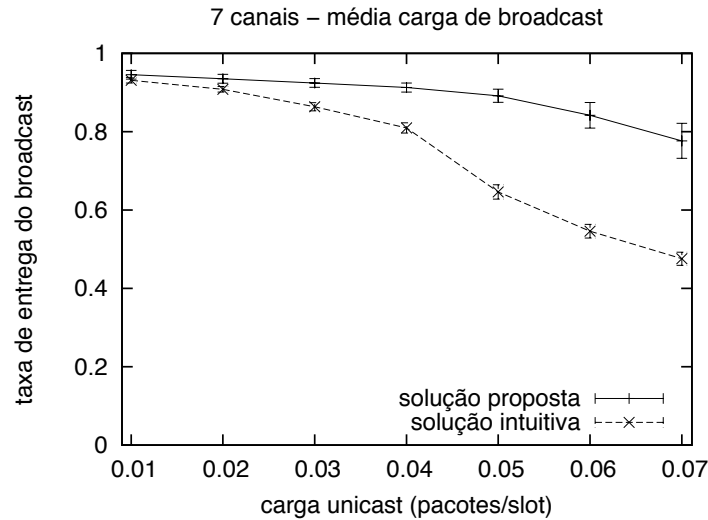


(b) 7 canais - alta carga

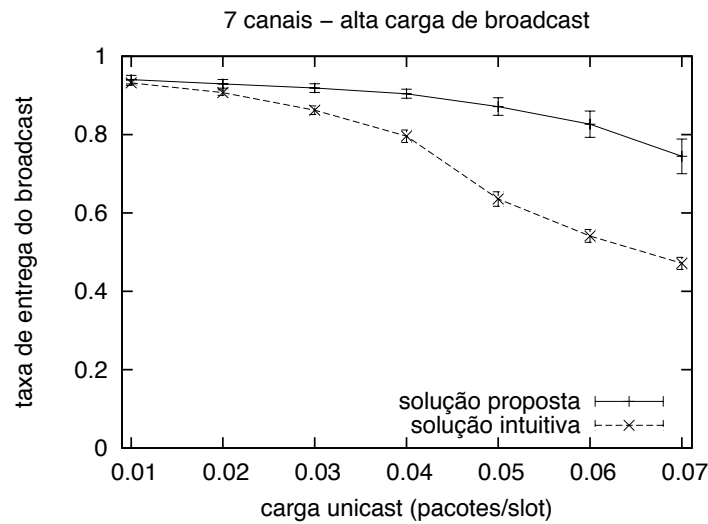
Figura 5.5: Taxa de Entrega do Tráfego *Unicast* com Carga de *Broadcast*.

lução intuitiva permitir que todos os seus vizinhos sejam alcançados por uma única transmissão, ela apresenta um pior desempenho em termos do atraso médio das mensagens *broadcast*. Quando a carga *unicast* ultrapassa um determinado valor, em torno de 0.04 pacotes/*slot* por nó, o atraso médio dessas mensagens cresce de forma acentuada, assim como a sua taxa de entrega, como evidenciado pelas Figuras 5.6(a) e 5.6(b). Portanto, a solução proposta favorece o tráfego *broadcast* mesmo em situações de alta carga na rede, aumentando a taxa de entrega e diminuindo o atraso médio na entrega das mensagens.

Os gráficos da Figura 5.8 mostram as três métricas de desempenho para a solução proposta em função da carga *unicast* com um número crescente de canais. A partir dos gráficos podemos notar que, quanto maior o número de canais, maior é a taxa de entrega do tráfego *broadcast*. No entanto, esse aumento no número de canais afeta



(a) 7 canais - média carga

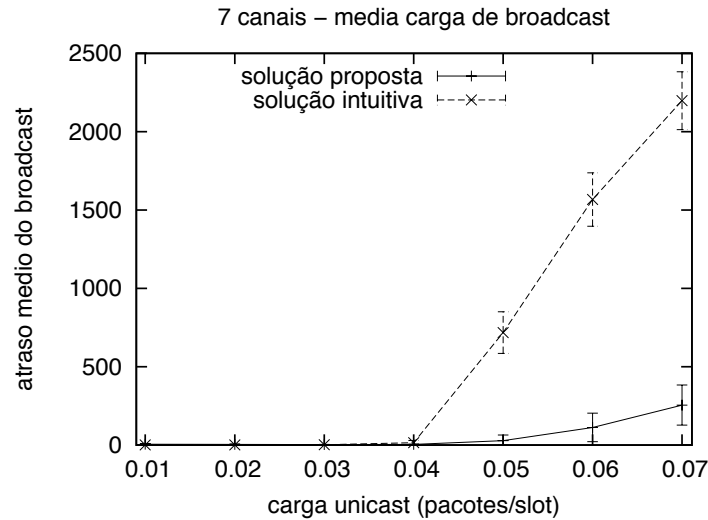


(b) 7 canais - alta carga

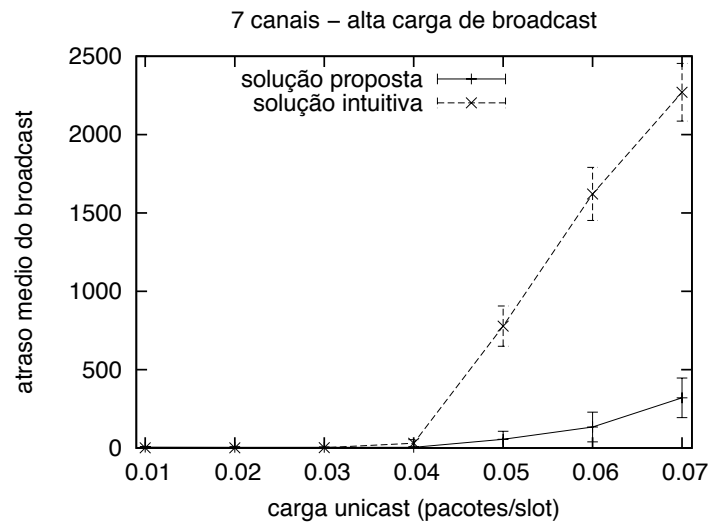
Figura 5.6: Taxa de Entrega do Tráfego *Broadcast*.

negativamente a taxa de entrega do tráfego *unicast*, ou seja, a capacidade disponível na rede. A explicação desse fenômeno é a mesma dos resultados das Figuras 5.5 e 5.6. Com o aumento do número de canais, maior será o tamanho da sequência de *broadcast*, e mais tempo o US permanecerá fora da sequência de repouso e terá dificuldade em receber pacotes *unicast*. Desta forma, existe um compromisso com relação ao número de canais a serem utilizados nas sequências de saltos da solução proposta.

Na Figura 5.9, temos o atraso médio das mensagens de *broadcast* da solução proposta, porém agora analisado sobre a influência do número de canais para montagem da sequência de *broadcast*. Podemos observar que para uma quantidade maior de canais, neste caso 11 canais, o atraso se apresenta em média maior, mesmo para cargas mais baixas de mensagens *unicast*. Isto se deve justamente pelo tamanho



(a) 7 canais - média carga



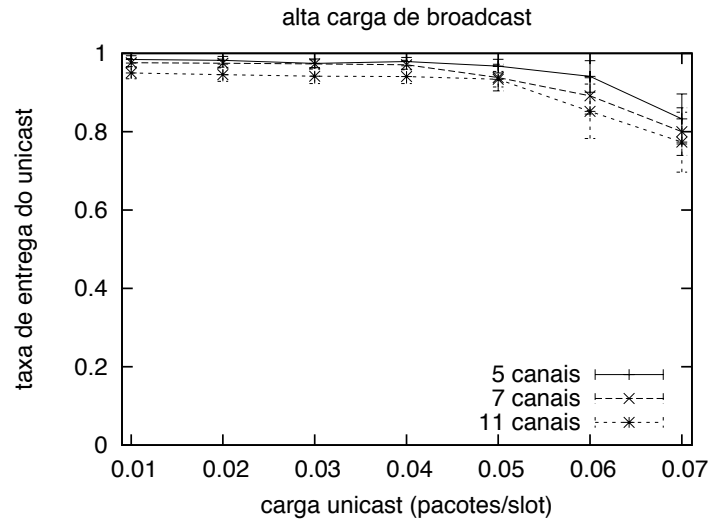
(b) 7 canais - alta carga

Figura 5.7: Atraso Médio do Tráfego *Broadcast*.

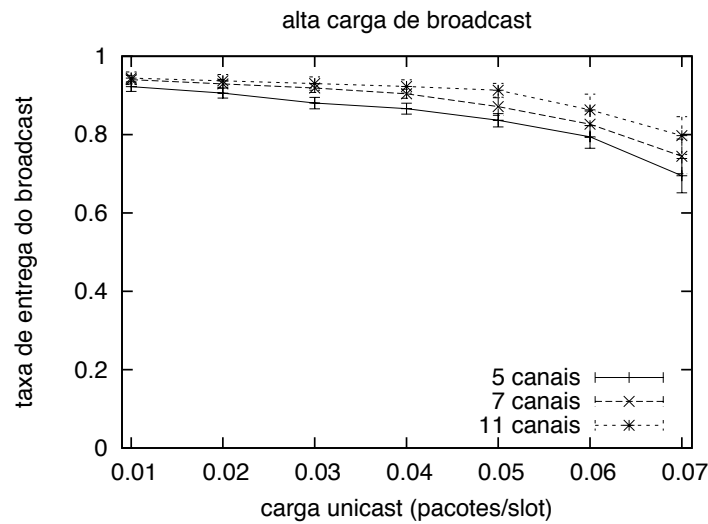
da sequência ser maior quando temos mais canais disponíveis para utilizar, e assim mais tempo o rádio permanece na sequência de *broadcast* repetindo a mesma mensagem em todos os *slots* da sequência. Isto aumenta as chances de entrega, e faz com que mensagens que levaram um tempo maior para serem entregues passem a ser contabilizadas na média do atraso, daí seu maior valor na média.

5.6 Conclusões do Capítulo

Neste capítulo, propomos regras de acesso ao meio que utilizam alguma solução de *rendezvous* e a criação de uma sequência de *broadcast*. Estas regras podem ser vistas como um protocolo de acesso ao meio. Nesta proposta, a partir de um critério de tipo de mensagem e/ou destinatário cria-se uma abordagem de papéis/estado que



(a) taxa de entrega *unicast*



(b) taxa de entrega *broadcast*

Figura 5.8: Influência do Número de Canais.

define diferentes sequências de saltos de canais a fim de facilitar o compartilhamento do meio. Este conjunto de regras de acesso foi comparado ao caso mais intuitivo de uso de apenas uma mesma sequência de canais, que a princípio é uma boa opção, porém não atende bem a situações onde o acesso ao meio é muito exigido.

A partir dos resultados pode-se constatar que o conjunto de regras empregado demonstrou uma melhora na taxa de entrega de mensagens, seja de *broadcast* quanto de *unicast*, ao mesmo tempo em que gerou uma redução no atraso médio de recepção das mensagens para múltiplos usuários. Assim, o uso do conjunto de regras representou uma boa estratégia na forma de acesso ao meio, sendo uma boa opção a se seguir como forma de funcionamento da rede.

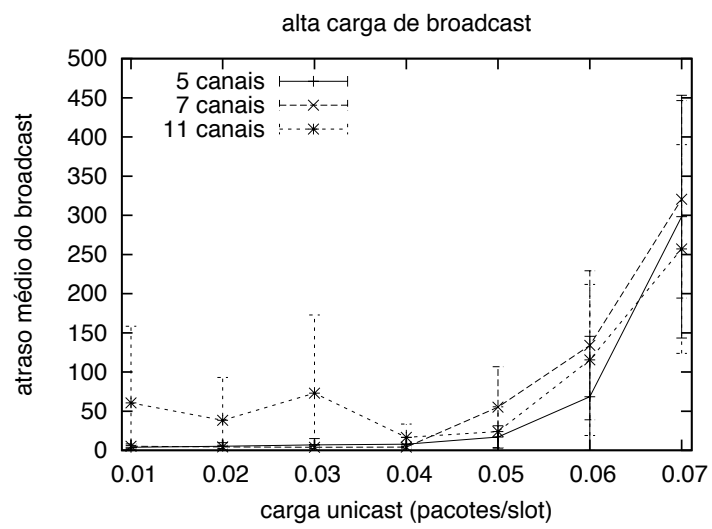


Figura 5.9: Atraso médio de transmissões em *broadcast*.

Capítulo 6

Conclusões e Trabalhos Futuros

Nos últimos anos, diversos trabalhos de pesquisa foram relacionados ao desenvolvimento de rádios cognitivos (RCs), uma vez que este tem sido reconhecido como uma maneira de utilizar o espectro de frequências de forma eficiente. Em uma rede de rádios cognitivos, os chamados usuários secundários (USs) estão autorizados a utilizar faixas de frequência do espectro licenciado de forma oportunista, desde que não causem interferência nos usuários primários (UPs) ou usuários licenciados do canal.

Nestas redes, antes que a transmissão de dados se inicie, dois USs que desejam se comunicar devem se encontrar em algum canal comum, livre para ambos, e estabelecer um enlace de comunicação [18], o que é chamado de *rendezvous*.

O *rendezvous* em uma rede de rádios cognitivos é uma etapa fundamental para o funcionamento da rede. No entanto, nem sempre é uma tarefa trivial, uma vez que o acesso ao espectro é realizado de forma dinâmica cada rádio pode selecionar um canal, dos que estão momentaneamente livres de UPs, para utilizar. Como essa escolha é individual, cada rádio pode escolher um canal diferente para uso. Além disso, os rádios podem nem sequer estarem cientes da presença uns dos outros antes do primeiro encontro.

Logo, o problema de *rendezvous* torna-se uma tarefa difícil dependendo da quantidade de informações que os rádios possuem sobre a rede e/ou sobre os outros rádios vizinhos. Geralmente, este problema é atacado através de dois grupos de abordagens bem conhecidos.

O primeiro grupo de abordagem considera o uso de um canal de comunicação exclusivo entre os USs, chamado Canal de Controle Comum (CCC). Pelo canal de controle, os USs podem se encontrar e trocar informações que auxiliam no funcionamento da rede. Porém devido a fatores como a atividade dinâmica dos UPs, nem sempre pode existir um canal livre que sirva simultaneamente a todos os USs. Além disso, o CCC pode apresentar o problema de gargalo do canal de controle quando diversos acessos são realizados ao canal ao mesmo tempo. No entanto, esta

abordagem permite que uma mensagem de controle seja enviada a todos os USs de maneira natural.

A outra abordagem da literatura, chamada sequência de salto de canais, propõe que os rádios trabalhem com a possibilidade de saltar de um canal para outro guiados através de uma sequência. Esta sequência de canais garante um encontro entre dois RCs em um tempo finito. Esta abordagem é mais imune a atividade de UPs, uma vez que os USs não se apoiam em um único canal para transmitirem mensagens de controle conforme o CCC. No entanto, neste tipo de abordagem transmissões em *broadcast* acabam não sendo tratadas ou mesmo as soluções propostas são ineficientes.

Como vemos, ambos grupos de soluções apresentam deficiências. Como característica, as deficiências de um CCC são solucionadas a partir da utilização de uma sequência de salto de canais e, por conseguinte, os problemas de uma sequência de salto de canais são solucionados com o uso de um CCC.

Desta maneira, as duas abordagens originam um ciclo, chamado Ciclo de *Rendezvous*. Neste ciclo as deficiências de abordagens que aplicam um CCC possuem como solução uma abordagem que aplique uma sequência de salto de canais e vice-versa. Grande parte das soluções propostas na literatura, ou sofrem de uma falta de disponibilidade de um CCC, e/ou problema de transmissões em *broadcast*.

Assim na primeira parte do trabalho, propomos uma solução de múltiplos canais de controle em conjunto com o uso de sequência de salto de canais. Desta forma, um grupo de rádios passou a utilizar vários canais para transmitir mensagens de controle ao invés de um único canal, deixando os rádios mais imunes à atividade de UPs. Utilizamos uma ferramenta computacional chamada *Reinforcement Learning* (RL), onde o RL tinha a tarefa de encontrar uma sequência de saltos de canais que atingisse uma taxa de entrega de mensagens de controle alta.

Pelos resultados obtidos, através de simulações, pode-se constatar que com o emprego desta abordagem, os rádios conseguiram descobrir sequências de saltos de canais que forneceram um bom desempenho na taxa de entrega de mensagens de controle, mesmo em ambientes com maior atividade de UPs.

Na segunda parte do trabalho, propomos dois algoritmos para a construção de uma sequência de salto de canais relacionada a transmissões em *broadcast*. Estes algoritmos levam em conta o número de canais que se sobrepõem e à atividade de cada UP em cada canal disponível. Ambas propostas são simples e agnósticas quanto a qual solução de sequência de salto de canais utilizada.

Estes dois algoritmos foram avaliados em termos da redução máxima e média de tempo de encontro, quando comparados com o caso em que o nó *broadcaster*, emissor da mensagens de *broadcast*, utiliza a sua própria sequência de salto de canais não modificada para executar a transmissão em *broadcast*. Os resultados das

simulações mostram que ambos os algoritmos proporcionam ganhos de desempenho nas métricas, além de um aumento no número de vizinhos alcançados, ou seja, uma maior taxa de sucesso.

Outra característica, comum aos trabalhos que tratam do problema de *rendez-vous* é que muitos estão relacionados a solução do primeiro encontro, sendo a forma de manutenção da rede e seu funcionamento pouco comentado.

Desta forma, na última parte do trabalho propomos um conjunto de regras de funcionamento que guiam os nós na forma de acesso ao meio. Nesta proposta, a partir de um critério de tipo de mensagem e/ou destinatário cria-se uma abordagem por papéis (*roles*) que define diferentes sequências de saltos de canais a fim de facilitar o compartilhamento do meio. Em resumo, os nós ao transmitirem uma mensagem *unicast* passam a seguir a sequência de salto de canais empregada pelo nó destinatário da mensagem. Caso não exista nada a transmitir, ou seja o nó em modo de recepção, utiliza a sequência de salto de canais própria. Quando o nó deseja enviar uma mensagem em *broadcast*, este passa a seguir a sua sequência de *broadcast*. A sequência de *broadcast* utilizada foi a mesma apresentada na segunda parte do trabalho.

Este conjunto de regras de funcionamento, que pode ser visto como um protocolo de acesso ao meio, foi comparado ao caso mais intuitivo de uso de apenas um canal ou uma mesma sequência de canais, que a princípio é uma boa opção, porém não atende bem a situações onde o acesso ao meio é muito exigido. A partir dos resultados pôde-se constatar que o conjunto de regras empregado demonstrou uma melhora na taxa de entrega de mensagens, tanto de *broadcast* quanto *unicast*, ao mesmo tempo em que gerou uma redução no atraso médio na recepção das mensagens de *broadcast*.

Como trabalhos futuros, pode-se propor, por exemplo, uma forma de criar uma sequência de salto de canais que utilize a informação de disponibilidade de canais quanto ao uso dos UPs. Em nossas pesquisas, verificamos que poucos trabalhos nesta área consideram esta informação, conforme pode-se notar a partir da Tabela 2.1 do Capítulo 2. Pode-se também modificar algum algoritmo da literatura, acrescentando tal informação, e verificando seus resultados, porém tentando preservar as propriedades do trabalho original.

Outra questão é com relação a uma troca de parâmetros dinâmica das soluções de sequência de salto de canais, onde a partir do uso de alguma informação, a solução opte por gerar uma nova sequência de saltos, agilizando o tempo de encontro e melhorando a questão de interferência sobre os UPs.

Esta possibilidade de troca de sequência pode ser disparada por alguma ferramenta de inteligência artificial, pois uma grande dificuldade é definir os instantes de troca e geração da nova sequência. Uma troca de parâmetros constante pode piorar a taxa de entrega entre os nós, enquanto pode diminuir a interferência sobre os UPs.

Esta mudança de parâmetros pode estar relacionada com uma estimativa de uso de canais pelos UPs, desta forma, sendo pró-ativa à seleção de canais.

Outro ponto interessante é tentar definir sequências que aumentem o reuso espacial. Assumindo que o nó já se encontrou com seus vizinhos, sabe suas sequências, sabe também a probabilidade de uso dos canais dos nós vizinhos, assim pode existir alguma forma de coordenação ou uso de sequências que aumente o reuso de canais entre um nó e seus vizinhos.

Todos os resultados apresentados foram obtidos a partir de simulação. Um ponto que pode ser interessante, como uma validação, é uma implementação em ambiente real. Uma dificuldade é a quantidade de nós empregados, visto que para o problema abordado o mais correto seria ter uma quantidade razoável de rádios. Além disso, esta tarefa exigiria a resolução de outros problemas, como detecção e diferenciação entre nós secundários e primários.

Referências Bibliográficas

- [1] FCC Spectrum Policy Task Force. “Report of the spectrum efficiency group”. 2002.
- [2] LIN, Z., LIU, H., CHU, X., et al. “Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio networks”. In: *Proceedings of IEEE INFOCOM*, pp. 2444–2452, 2011. doi: 10.1109/INFOCOM.2011.5935066.
- [3] AKYILDIZ, I. F., LEE, W.-Y., VURAN, M. C., et al. “NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey”, *Computer Networks: The Int. J. of Comp. and Telecom. Networking*, v. 50, pp. 2127–2159, September 2006.
- [4] SHEN, L. M. C.-C., RYU, B. “Single-Radio Adaptive Channel Algorithm for Spectrum Agile Wireless Ad Hoc Networks”. In: *Proceedings of IEEE DySPAN*, 2007.
- [5] MITOLA, J. *Cognitive radio: Model-based competence for software radios*. Tese de Doutorado, Dept. of Teleinformatics, KTH, 1999.
- [6] HAYKIN, S. “Cognitive radio: brain-empowered wireless communications”, *IEEE J. Select. Areas Commun*, v. 23, n. 2, pp. 201–220, fev. 2005.
- [7] THEIS, N. C., THOMAS, R. W., DASILVA, L. A. “Rendezvous for cognitive radios”, *IEEE Transactions on Mobile Computing*, v. 10, n. 2, pp. 216–227, fev. 2011. ISSN: 1536-1233. doi: 10.1109/TMC.2010.60.
- [8] GALINDO-SERRANO, A., GIUPPONI, L., BLASCO, P., et al. “Learning from Experts in Cognitive Radio Networks: The Docitive Paradigm”. In: *5th International Conference on Cognitive Radio Oriented Wireless Networks and Communications, (CROWNCOM 2010), Cannes (France)*, 2010.
- [9] STEVENSON, C., CHOUINARD, G., LEI, Z., et al. “IEEE 802.22: The first cognitive radio wireless regional area network standard”, *Communications*

Magazine, IEEE, v. 47, n. 1, pp. 130–138, January 2009. ISSN: 0163-6804. doi: 10.1109/MCOM.2009.4752688.

- [10] KHALIFE, H., MALOUCH, N., FDIDA, S. “Multihop cognitive radio networks: to route or not to route”, *Network, IEEE*, v. 23, n. 4, pp. 20–25, July 2009. ISSN: 0890-8044. doi: 10.1109/MNET.2009.5191142.
- [11] AKYILDIZ, I. F., LO, B. F., BALAKRISHNAN, R. “Cooperative spectrum sensing in cognitive radio networks: A survey”, *Physical Communication*, v. 4, pp. 40–62, 2011.
- [12] SO, J., VAIDYA, N. H. “Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver”. In: *Proceedings of ACM MobiHoc*, pp. 222–233, New York, NY, USA, 2004. ACM. ISBN: 1-58113-849-0. doi: 10.1145/989459.989487.
- [13] MO, J., SO, H.-S. W., WALRAND, J. “Comparison of multi-channel MAC protocols”. In: *Proceedings of ACM MSWiM*, pp. 209–218, New York, NY, USA, 2005. ACM. ISBN: 1-59593-188-0. doi: 10.1145/1089444.1089480.
- [14] LUO, T., MOTANI, M., SRINIVASAN, V. “CAM-MAC: A cooperative asynchronous multi-channel mac protocol for ad hoc networks”. In: *Proceedings of IEEE BROADNETS*, pp. 1–10, oct. 2006. doi: 10.1109/BROADNETS.2006.4374373.
- [15] BIAN, K., PARK, J.-M., CHEN, R. “Control Channel Establishment in Cognitive Radio Networks using Channel Hopping”, *IEEE Journal on Selected Areas in Communications*, v. 29, n. 4, pp. 689–703, april 2011. ISSN: 0733-8716. doi: 10.1109/JSAC.2011.110403.
- [16] HTIKE, Z., HONG, C. S., LEE, S. “The Life Cycle of the Rendezvous Problem of Cognitive Radio Ad Hoc Networks: A Survey”, *Journal of Computing Science and Engineering*, v. 7, pp. 81–88, June 2013.
- [17] BIAN, K., PARK, J.-M. “Asynchronous channel hopping for establishing rendezvous in cognitive radio networks”. In: *Proceedings of IEEE INFOCOM*, 2011.
- [18] SHIN, J., YANG, D., KIM, C. “A Channel Rendezvous Scheme for Cognitive Radio Networks”, *Communications Letters, IEEE*, v. 14, n. 10, pp. 954–956, October 2010. ISSN: 1089-7798. doi: 10.1109/LCOMM.2010.091010.100904.

- [19] KRUNZ, M. “Rendezvous Protocols for Spectrum-agile Wireless Networks”.
http://www2.engr.arizona.edu/~krunz/Rendezvous_Protocols.htm
 - último acesso em 30/03/2015.
- [20] GANDHI, R., WANG, C.-C., HU, Y. C. “Fast Rendezvous for Multiple Clients for Cognitive Radios Using Coordinated Channel Hopping”. In: *Proceedings of IEEE SECON*, pp. 1–9, 2012.
- [21] LO, B. F. “A survey of common control channel design in cognitive radio networks”, *Elsevier Physical Communication*, v. 4, pp. 26–39, 2011.
- [22] MO, J., SO, H. S., WARLAND, J. “Comparison of Multichannel MAC Protocols”, *IEEE Transactions Mobile Computing*, v. 7, pp. 50–65, June 2008.
- [23] XIAO, Y., HU, F. “Cognitive Radio Networks”, *CRC Press*, 2008.
- [24] CHEN, T., ZHANG, H., MAGGIO, G. M., et al. “CogMesh: A Cluster-based Cognitive Radio Network”. In: *IEEE*), jul. 2007.
- [25] LAZOS, L., LIU, S., KRUNZ, M. “Spectrum Opportunity-based Control Channel Assignment in Cognitive Radio Networks”. In: *Proceedings of IEEE SECON*, pp. 1–9, 2009.
- [26] CHEN, T., ZHANG, H., MAGGIO, G. M., et al. “Topology Management in CogMesh: A Cluster-based Cognitive Radio Mesh Network”. In: *Proceedings of IEEE ICC*, pp. 6516–6521, 2007.
- [27] TAO CHEN, H. Z., KATZ, M. D. *Cloud Networking Formation in CogMesh Environment*. Relatório técnico, Zhejiang University, China. VTT, Finland, 2009.
- [28] GUEDES, R. M., DE REZENDE, J. F. “Seleção Dinâmica e Distribuída de Canais de Controle em Rádios Cognitivos”. In: *I Workshop on Broadband Access Networks (WRA 2012) SBRC 2012*, may 2012.
- [29] ZHAO, J., ZHENG, H., YANG, G.-H. “Distributed Coordination in Dynamic Spectrum Allocation Networks”. In: *Proceedings of IEEE DySPAN*, pp. 259–268, nov. 2005.
- [30] CHEN, T., ZHANG, H., KATZ, M., et al. “Swarm Intelligence Based Dynamic Control Channel in CogMesh”. In: *Proceedings of IEEE ICC*, pp. 123–128, 2008.

- [31] SILVIUS, M. D., GE, F., YOUNG, A., et al. “Smart Radio: Spectrum Access for First Responders”. In: *Proceedings of SPIE Defense and Security Conference*, mar. 2008.
- [32] DASILVA, L. A., GUERREIRO, I. “Sequence-Based Rendezvous for Dynamic Spectrum Access”. In: *Proceedings of IEEE DySPAN*, pp. 1–7, out. 2008. doi: 10.1109/DYSPAN.2008.52.
- [33] BAHL, P., CHANDRA, R., DUNAGAN, J. “SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks”. In: *Proceedings of ACM MobiCom*, pp. 216–230, New York, NY, USA, 2004. ACM. ISBN: 1-58113-868-7. doi: 10.1145/1023720.1023742.
- [34] CORMIO, C., CHOWDHURY, K. R. “Common control channel design for cognitive radio wireless ad hoc networks using adaptive frequency hopping”, *Ad Hoc Networks*, v. 8, n. 4, pp. 430–438, 2010. ISSN: 1570-8705. doi: 10.1016/j.adhoc.2009.10.004.
- [35] ZHANG, Y., LI, Q., YU, G., et al. “ETCH: Efficient Channel Hopping for communication rendezvous in dynamic spectrum access networks”. In: *Proceedings of IEEE INFOCOM*, pp. 2471–2479, 2011. doi: 10.1109/INFOCOM.2011.5935070.
- [36] ALTAMIMI, M., NAIK, K., SHEN, X. S. “Parallel link rendezvous in ad hoc cognitive radio networks”. In: *Proceedings of IEEE Global Telecommunications Conference*, pp. 1–6, 2010.
- [37] PU, D., WYGLINSKI, A. M., MCLERNON, M. “A frequency rendezvous approach for decentralized dynamic spectrum access networks”. In: *4th International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, pp. 1–6, 2009.
- [38] AXELL, E., LEUS, G., LARSSON, E., et al. “Spectrum Sensing for Cognitive Radio : State-of-the-Art and Recent Advances”, *Signal Processing Magazine, IEEE*, v. 29, n. 3, pp. 101–116, May 2012. ISSN: 1053-5888. doi: 10.1109/MSP.2012.2183771.
- [39] YANG, D., SHIN, J., KIM, C. “Deterministic Rendezvous Scheme in Multi-channel Access Networks”, *IET Electronics Letters*, v. 46, Sep 2010.
- [40] GUEDES, R. M., DA SILVA, M. W. R., COUTINHO, P. S., et al. “Agnostic Broadcast Rendezvous for Cognitive Radio Networks Using Channel Hopping”. In: *Proceedings of IEEE LCN*, pp. 647–654, october 2012.

- [41] MA, L., HAN, X., SHEN, C. C. “Dynamic open spectrum sharing MAC protocol for wireless ad hoc networks”. In: *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 203–213, 2005.
- [42] GUEDES, R. M., DE REZENDE, J. F. “Seleção Dinâmica de Canais de Controle em Rádios Cognitivos Utilizando Reinforcement Learning”. In: *Proceedings of SBRC 2011*, pp. 75–88, may 2011.
- [43] YAU, K.-L. A., KOMISARCZUK, P., TEAL, P. D. “Applications of Reinforcement Learning to Cognitive Radio Networks”. In: *Communications Workshops (ICC), 2010 IEEE International Conference*, jul. 2010.
- [44] WIERING, M. A., VAN HASSELT, H. *Ensemble Algorithms in Reinforcement Learning*. Relatório técnico, University of Groningen and Utrecht University, Netherlands.
- [45] SUTTON, R. S., BARTO, A. G. “Reinforcement Learning: An Introduction”. A Bradford Book The MIT Press Cambridge, Massachusetts London, England, 1998.
- [46] KOK, J. R., VLASSIS, N. “Collaborative Multiagent Reinforcement Learning by Payoff Propagation”, *J. Mach. Learn. Res.*, v. 7, pp. 1789–1828, December 2006. ISSN: 1532-4435.
- [47] BERGE, C. “The Theory of Graphs and its Applications”. In: *Greenwood Press Reprint; New edition edition*, 1982.
- [48] NS-2. “The Network Simulator - ns-2”. <http://www.isi.edu/nsnam/ns/> - último acesso em 07/12/2012.
- [49] KIM, H., SHIN, K. G. “Efficient Discovery of Spectrum Opportunities with MAC-Layer Sensing in Cognitive Radio Networks”, *IEEE Transactions on Mobile Computing*, v. 7, n. 5, pp. 533–545, maio 2008.
- [50] BRAR, G., BLOUGH, D. M., SANTI, P. “Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks”. In: *Proceedings of ACM MobiCom*, pp. 2–13, New York, NY, USA, 2006. ACM. doi: 10.1145/1161089.1161092.
- [51] WELLENS, M., RIIHIJÄRVI, J., MÄHÖNEN, P. “Empirical time and frequency domain models of spectrum use”, *Physical Communication*, v. 2, n. 1-2, pp. 10–32, 2009. ISSN: 1874-4907. doi: 10.1016/j.phycom.2009.03.001.

Apêndice A

Modular Clock formulação

Neste anexo, apresentamos o algoritmo e a prova matemática do funcionamento da solução *mclock*. Todo o conteúdo deste anexo foi retirado de [7].

Modular Clock é um algoritmo de *rendezvous* que usa número primos e aritmética modular para garantir o TTR (*Time To Rendezvous*) entre os rádios que utilizam a sequência de salto de canais.

Os autores adotaram a seguinte notação para os parâmetros que o *mclock* utiliza:

- r_i taxa com que o rádio i salta entre os canais. A cada *slot* de tempo, o rádio salta avançando r_i canais dentre o conjunto de canais disponíveis, retornando conforme um ciclo (aplicando aritmética modular) quando acalça um canal maior que um limiar.
- t indica o tempo do sistema em *slots*
- p_i é o número primo escolhido pelo rádio i
- m_i quantidade de canais disponíveis para o rádio i

No Algoritmo 4, temos o pseudo código do funcionamento do *modular clock*. O algoritmo inicia pela seleção de um canal aleatoriamente dentre o conjunto de canais disponíveis. Depois de cada *slot* de tempo, o índice j_i é incrementado por $r_i \bmod p_i$. O módulo de p_i é o menor número primo maior ou igual ao número de canais disponíveis m_i . Se o índice do canal resultante j_i está em $[0, m_i)$, então o rádio salta para o canal c_{i,j_i} , ou seja, o canal com índice j_i dentro do conjunto de canais disponíveis para o rádio i .

Pode existir um “gap” entre m_i e p_i , caso o índice resultante seja maior que $m_i - 1$. Neste caso, o índice é remapeado através da função *mod* entre 0 e $m_i - 1$ (inclusive) e o rádio move-se para este canal (note que o índice j_i não é alterado neste caso).

Se o encontro não for alcançado após $2p_i$ slots de tempo, um novo valor de r_i é escolhido aleatoriamente dentro do intervalo $[0, p_i)$. O algoritmo termina em qualquer instante quando o encontro é alcançado. Note que todas as variáveis listadas no Algoritmo 4 são relacionadas ao rádio que está executando o algoritmo, e não a valores globais.

Algoritmo 4: Algoritmo do *Modular Clock*

```

1 Observa-se  $m_i$ , o número de canais disponíveis
2 Calcula-se  $p_i$ , o próximo número primo maior que  $m_i$ 
3  $j_i^0 = rand[0, m_i)$ 
4 while not rendezvous do
5     escolha  $r_i$  de  $[0, p_i)$  aleatoriamente
6     for  $t = 0$  até  $2p_i$  do
7          $j_i^{t+1} = (j_i^t + r_i) \bmod (p_i)$ 
8         if  $j_i^{t+1} < m_i$  then
9              $c = c_{i, j_i^{t+1}}$ 
10        else
11             $c = c_{i, (j_i^{t+1} \bmod (m_i))}$ 
12        Tente o rendezvous no canal  $c$ 

```

Observe que entre as linhas do **for** no algoritmo, o índice do canal do rádio i que está no passo t pode ser expressado pela equação:

$$j_i^t = t \cdot r_i + j_i^0 \bmod (p_i), \quad (\text{A.1})$$

onde j_i^0 é o índice do canal inicial do rádio i . Os subíndices são utilizados para distinguir entre os diferentes rádios que utilizam o algoritmo *mclock*. Além disso, observa-se que, quando os índices do canal estão em $[m_i, p_i)$, o algoritmo consistentemente remapeia esses canais para $[0, m_i)$. Assim, este passo da análise refere-se a canais que existem em $[0, p_i)$.

Assumindo que todos os rádios observam o mesmo conjunto de canais e as identificações destes canais são as mesmas. Podemos dizer que: $\bar{m} = m = m_i$ e também $p_1 = p_2$.

Teorema 1: No modelo quando dois rádios executam o algoritmo *mclock* com $r_1 \neq r_2$, o encontro entre eles ocorrerá em um intervalo de tempo menor que p_i .

Prova: Sem perda de generalidade, pode-se iniciar o algoritmo no ponto em que os dois rádios começam a tentar o encontro. Definindo-se $t = 0$ a ser este momento e, os índices de canais neste momento, j_1^0 e j_2^0 , são distribuídos aleatoriamente dentro de $[0, m)$, a convergência irá ocorrer quando:

$$t \cdot r_1 + j_1^0 \bmod(p_1) \equiv t \cdot r_2 + j_2^0 \bmod(p_2). \quad (\text{A.2})$$

Definiu-se $\delta_r = r_1 - r_2$ e $\delta_j = j_1^0 - j_2^0$. Assim, a Equação A.2 pode ser reescrita como:

$$t \cdot \delta_r \equiv -\delta_j \bmod(p_i). \quad (\text{A.3})$$

Com isso, dois pontos podem ser observados:

1. Com $r_1 \neq r_2$ e $r_i \in [0, p_i)$, isto significa que $\delta_r \in (-p_i, 0) \cup (0, p_i)$.
2. O inverso multiplicativo modular (i.e., a^{-1} onde $a \cdot a^{-1} \bmod(p) \equiv 1$) de qualquer número a e módulo p existe se e somente se a e p são coprimos, onde coprimo é definido no sentido do maior divisor comum (mdc) entre a e p ser igual a 1.

Combinando o primeiro ponto com o fato de que p_i é primo, sabe-se que $\text{mdc}(\delta_r, p_i) = 1$. A partir do segundo ponto, sabe-se que existe um inverso multiplicativo modular $\delta_r \bmod(p_i)$ que será chamado de δ_r^{-1} . Tendo mostrado que δ_r^{-1} existe, multiplica-se ambos os lados de A.3 por ele, resultando em:

$$t \equiv -\delta_j \cdot \delta_r^{-1} \bmod(p_i). \quad (\text{A.4})$$

Desde que o lado direito de A.4 seja um valor em $[0, p_i)$ e que o valor de t esteja em $[0, p_i)$, pode-se provar que a convergência irá ocorrer em $t < p_i$ passos.

Proposição 1: Quando dois rádios executam o algoritmo *mclock* simplesmente seguindo A.1 com $r_1 = r_2$ e $j_1^0 \neq j_2^0$, o encontro não ocorrerá.

Prova: Usamos a mesma notação da prova do **Teorema 1**, e provando por contradição.

Assume-se que:

$$t \cdot \delta_r \equiv -\delta_j \bmod(p_i). \quad (\text{A.5})$$

Desde que $j_i^0 \in [0, p_i)$ com $j_1^0 \neq j_2^0$, isto significa que $\delta_j \in (-p_i, 0) \cup (0, p_i)$. Além disso, desde que $r_1 = r_2$, isto significa que $\delta_r = 0$. Assim, A.5 pode ser reescrita como:

$$0 \equiv -\delta_j \bmod(p_i). \quad (\text{A.6})$$

No entanto, uma vez que $\delta_j \in (-p_i, 0) \cup (0, p_i)$, o lado direito da equação está em $(0, p_i)$ e a igualdade não é mantida. Assim, contrariou-se a hipótese de base.

Outra maneira de olhar para a **Proposição 1** é a seguinte: uma vez que o valor de δ_r é 0, as sequências mantêm a mesma diferença de índice de canal (δ_j) quando iniciam o algoritmo.

Para corrigir isso, o algoritmo *mclock* tem um recurso de “*time-out*” na linha 6. Ao esperar até $2p_i$ intervalos de tempo antes do tempo limite para mudar os valores de r_i , os rádios podem assegurar que tiveram pelo menos p_i slots sem que outro rádio tenha mudado seu r_i . Por exemplo, suponha que o Rádio 1 inicie sua sequência no tempo de *slot* 0, enquanto que o Rádio 2 inicia a sua sequência no tempo de *slot* de $p_i - \varepsilon$. Se o Rádio 1 aguarda até $t = 2p_i$ para alterar o valor de r_i , então eles vão ter mantido fixo os valores de r_i por $p_i + \varepsilon$ intervalos de tempo, o que garante pelo **Teorema 1** que terá tempo suficiente para ocorrer o encontro. A Figura A.1 ilustra essa ideia.

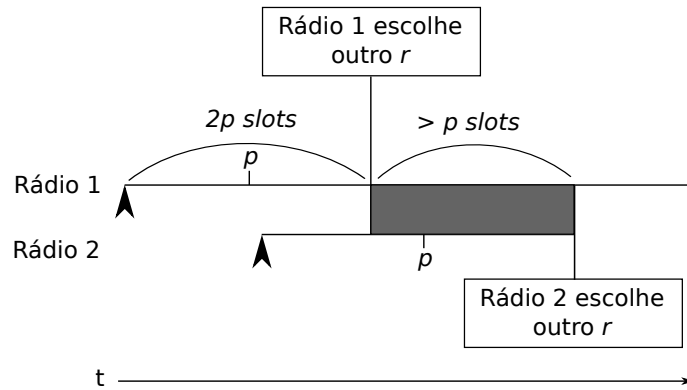


Figura A.1: Ilustração de $2p_i$ “*time-out*” do algoritmo *mclock* que garante, que os rádios têm intervalos de tempo de pelo menos p_i slots com r_i fixo.

Todas as informações e deduções apresentadas referem-se ao tipo de *mclock* chamado modelo compartilhado (*shared model*). Existe uma modificação do *mclock* para permitir encontros, sem que a condição $\bar{m} = m = m_i$ seja satisfeita. Não apresentamos neste anexo esta modificação e dedução pois utilizamos no trabalho o *mclock* que assume a condição de $\bar{m} = m = m_i$.