



## CONTROLE DE INUNDAÇÃO EM REDES AD HOC MÓVEIS

Carlos Henrique Pereira Augusto

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  
Junho de 2011

# CONTROLE DE INUNDAÇÃO EM REDES AD HOC MÓVEIS

Carlos Henrique Pereira Augusto

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. Luís Henrique Maciel Kosmowski Costa, Dr.

---

Prof. Valmir Carneiro Barbosa, Ph.D.

---

Prof. Ronaldo Moreira Salles, Ph.D.

---

Prof. Edmundo Roberto Mauro Madeira, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

JUNHO DE 2011

Augusto, Carlos Henrique Pereira

Controle de Inundação em Redes Ad Hoc Móveis/Carlos Henrique Pereira Augusto. – Rio de Janeiro: UFRJ/COPPE, 2011.

XIV, 72 p.: il.; 29, 7cm.

Orientador: José Ferreira de Rezende

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2011.

Referências Bibliográficas: p. 67 – 72.

1. Redes Ad Hoc. 2. Inundação. 3. Roteamento.  
I. Rezende, José Ferreira de. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica.  
III. Título.

*À minha família, e ao meu  
saudoso irmão Sergio.*

# Agradecimentos

À Lili, minha esposa, companheira e amiga, e aos meus filhos, Lucas e Marcos, pela compreensão pelo tempo subtraído deles, e todo apoio neste longo projeto.

Ao Prof. José Ferreira de Rezende, pela amizade e realismo com que me orientou, sabendo compreender minhas dificuldades e ajudando-me a superá-las.

Aos amigos Marcel, Kleber, Celso, Raphael, André e Coutinho, colegas do GTA, pelo apoio e cumplicidade durante a realização do doutorado. As conversas no café, tanto sobre os conceitos e ideias, quanto sobre as aflições e dificuldades, sempre facilitaram o passo além no trabalho.

Aos amigos Luiz Antônio, José Roberto e Gabriela, do Bacen, pelo incentivo, e ao grande amigo Rafael, também pela compreensão pelo meu afastamento.

Aos meus pais, Gerson e Ednéa, por toda dedicação e esforço ao longo da vida, que me permitiram encontrar meu caminho, e alcançar mais este objetivo.

Ao Banco Central do Brasil, pelo apoio à realização deste trabalho.

A todos aqueles que em algum momento me ajudaram a crescer, e colaboraram para meu aprendizado e minha jornada, em especial às minhas saudosas avós Juracy e Zenith, aos amigos Castañon, Dalle Ore e Veiga, à minha prima Ivanir, e aos meus grandes e saudosos amigos, Pacheco e Sergio.

Finalmente, Àquele que tudo permitiu, doando-me saúde e os meios necessários para buscar alcançar os meus objetivos, Deus.

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

## CONTROLE DE INUNDAÇÃO EM REDES AD HOC MÓVEIS

Carlos Henrique Pereira Augusto

Junho/2011

Orientador: José Ferreira de Rezende

Programa: Engenharia Elétrica

A inundação de mensagens de controle e de dados é um mecanismo bastante empregado por protocolos e aplicações em redes *ad hoc*. Entretanto, tal mecanismo pode causar efeitos nocivos no desempenho dessas redes, gerando um número excessivo de mensagens redundantes e, conseqüentemente, consumindo grandes quantidades de energia e aumentando a latência na entrega destas mensagens. Nesta tese são estudados estes problemas e as soluções existentes na literatura. Também são apresentadas duas propostas, a primeira, chamada de FloorB, que reduz o número de mensagens reencaminhadas através do conhecimento resumido, por filtros de Bloom, da vizinhança de dois saltos, e a segunda, chamada 2BSheld, que realiza um escalonamento das mensagens de inundação de forma distribuída e com conhecimento parcial da topologia, com o objetivo de controlar a latência da inundação. Ambas as propostas são implementadas em ambiente de simulação e comparadas a outras propostas da literatura ou limites teóricos existentes.

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

## FLOOD CONTROL IN MOBILE AD HOC NETWORKS

Carlos Henrique Pereira Augusto

June/2011

Advisor: José Ferreira de Rezende

Department: Electrical Engineering

Flooding of data and control messages is a widely used mechanism in network applications and protocols. However, such mechanism can have harmful effects on the performance of these networks with a drawback effect called broadcast storm, generating an excessive number of redundant messages and hence consuming large amounts of energy and increasing the latency time to message delivery. This thesis studies these problems and solutions in the literature. It also presents two proposals, the first, called FloorB, which reduces the number of forwarded messages through the 2-hop neighborhood knowledge summarized by Bloom filters, and the second, called 2BSheld, which carries a schedule of flooding messages using a partial knowledge of the topology in order to control the latency of flood. Both proposals are implemented in simulation environment and compared to other proposals in the literature or existed theoretical limits.

# Sumário

<b>Lista de Figuras</b>	<b>x</b>
<b>Lista de Tabelas</b>	<b>xii</b>
<b>Lista de Abreviaturas</b>	<b>xiii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Organização do Texto . . . . .	3
<b>2 Trabalhos Relacionados</b>	<b>5</b>
<b>3 Propostas</b>	<b>12</b>
3.1 FloorB - Representação de Vizinhança por Filtros de Bloom . . . . .	12
3.1.1 Vizinhança de 1 e 2 saltos e Reencaminhamento de Mensagem	12
3.1.2 Filtros de Bloom . . . . .	16
3.1.3 FloorB - Algoritmo . . . . .	18
3.1.4 FloorB - Modelo Geométrico . . . . .	20
3.2 2BSheld - Escalonamento de Inundação com Visão e Decisão Local . . . . .	24
3.2.1 Avaliação de Mecanismos de Redução da Inundação . . . . .	24
3.2.2 Mecanismo Proposto . . . . .	28
3.3 Conclusões . . . . .	30
<b>4 Resultados</b>	<b>32</b>
4.1 FloorB . . . . .	32
4.1.1 Implementações . . . . .	32
4.1.2 Simulador próprio . . . . .	36
4.1.3 Simulação no ns-2 . . . . .	45
4.2 2BSheld . . . . .	53
4.2.1 Implementação . . . . .	53
4.2.2 Simulação no ns-2 . . . . .	54
4.3 Conclusões . . . . .	60

<b>5</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>63</b>
5.1	FloorB . . . . .	63
5.2	2BSheld . . . . .	64
5.3	Trabalhos Futuros . . . . .	65
	<b>Referências Bibliográficas</b>	<b>67</b>

# Lista de Figuras

2.1	Exemplo de Funcionamento de MPR. . . . .	8
3.1	Comunicação entre dois nós. . . . .	13
3.2	Conjuntos de Vizinhança. . . . .	14
3.3	Topologia com corte em um nó. . . . .	15
3.4	Funcionamento do filtro de Bloom. . . . .	16
3.5	Probabilidade de falso positivo. . . . .	17
3.6	Descrição geométrica dos conjuntos de vizinhança. . . . .	21
3.7	Valores de $p$ nos modelos geométricos. . . . .	24
3.8	Cenário em grade. . . . .	25
3.9	Latências de inundação. . . . .	27
3.10	Exemplo de funcionamento do 2Bsheld. . . . .	28
4.1	Cenário em grade, sem perdas . . . . .	37
4.2	Distribuição da tarefa de reencaminhamento. . . . .	38
4.3	Cenário em grade, perdas=0,2. . . . .	40
4.4	Cenário com mobilidade RWP, sem perdas. . . . .	42
4.5	Cenário com mobilidade RWP, variação do intervalo entre <i>hellos</i> . . . . .	43
4.6	Cenário com mobilidade Social (Comunidade), sem perdas. . . . .	44
4.7	Cenário com mobilidade Social (Comunidade), perda=0,2. . . . .	45
4.8	Cenário em grade, ns-2. . . . .	46
4.9	Cenário com mobilidade RWP, ns-2. . . . .	47
4.10	Cenário com mobilidade RWP, variação do intervalo entre <i>hellos</i> , ns-2. . . . .	48
4.11	Variação do tamanho do filtro de Bloom. . . . .	49
4.12	Disposição dos nós geradores de carga. . . . .	50
4.13	Cenário com mobilidade RWP, rede com carga variável, ns-2. . . . .	51
4.14	Desempenho do protocolo de roteamento. . . . .	53
4.15	Latências de inundação do escalonador em CSMA/CA e TDMA. . . . .	55
4.16	Latências de inundação do escalonador associado ao FloorB e MPR. . . . .	56
4.17	Latências de inundação do escalonador com função log. . . . .	57
4.18	Latência e taxa de entrega com limite T. . . . .	59

4.19 Escalonamento de inundação com mobilidade - MPR, limite T. . . . .	60
4.20 Escalonamento de inundação com mobilidade - FloorB, limiteT. . . . .	61

# Lista de Tabelas

3.1	Parâmetros da Simulação. . . . .	26
-----	----------------------------------	----

# Lista de Abreviaturas

AODV	<i>Ad hoc On-Demand Distance Vector Routing</i> , p. 1
BBS	<i>Basic Broadcast Schedule</i> , p. 10
CSMA/CA	<i>Carrier Sense Multiple Access with Collision Avoidance</i> , p. 3, 9, 27, 55
EBS	<i>Enhanced Broadcast Schedule</i> , p. 10
IEEE	<i>Institute of Electrical and Electronics Engineers</i> , p. 3, 35, 45, 64
IMC	<i>Iterative Minimal Covering</i> , p. 10
MIS	<i>Maximal Independent Set</i> , p. 10
MLBS	<i>Minimum-Latency Broadcast Scheduling</i> , p. 3, 9, 10, 27, 65
MLST	<i>Maximum Leaf Spanning Tree</i> , p. 5
MPR	<i>Multi Point Relays</i> , p. 3, 7, 8, 19, 34, 35, 38, 39, 41, 42, 44, 47, 64
NP	<i>nondeterministic polynomial time</i> , p. 5
OLSR	<i>Optimized Link State Routing Protocol</i> , p. 1, 7, 35
PBS	<i>Pipelined Broadcast Schedule</i> , p. 10
RAPID	<i>ReliAble ProbabIlistic Dissemination protocol</i> , p. 6, 34
RWP	<i>Random Waypoint</i> , p. 3, 35
SINR	<i>Signal to Interference plus Noise Ratio</i> , p. 9
STDMA	<i>Spatial Time Division Multiple Access</i> , p. 3, 10
TDMA	<i>Time Division Multiple Access</i> , p. 3, 9, 26, 54, 55

TTL      *Time to Live*, p. 52  
UDG      *Unit Disk Graphs*, p. 9, 10, 30

# Capítulo 1

## Introdução

Inundação em redes consiste em um mecanismo ou protocolo com o objetivo de entregar uma mensagem a todos os nós da rede. Em redes *ad hoc*, que apresentam características de topologia dinâmica, escassez de recursos e ausência de infraestrutura, estes mecanismos tornam-se essenciais, por permitirem a coleta ou distribuição de informação na rede, sendo largamente utilizados em protocolos de roteamento [1, 2], descoberta de serviços [3, 4], e para disseminação ou busca de informação de uma forma geral [5].

Entretanto, um dos problemas introduzidos pela inundação é a chamada tempestade de *broadcast* [6], que implica em transmissões redundantes, aumento de colisões, e alto nível de disputa no meio com seu consequente aumento de latência de transmissão, consumindo desta forma os já escassos recursos disponíveis.

O controle destes efeitos nocivos da tempestade de *broadcast* pode ser realizado através de algumas abordagens, que consideramos ortogonais. Uma opção é limitar o próprio uso da inundação, por exemplo, como o trabalho em [7], que evita o uso de inundações ao descobrir previamente e manter armazenadas rotas alternativas. Outro exemplo de limitação do uso da inundação ocorre no emprego do anel expansivo no protocolo de roteamento AODV (*Ad hoc On-Demand Distance Vector Routing Protocol*) [2], que restringe o envio inicial de mensagens de requisição de rotas a somente alguns saltos, não alcançando a rede toda. Solução semelhante é utilizada no mecanismo conhecido como *Fish-eye* [8], que pode ser utilizado em conjunto com o protocolo de roteamento OLSR (*Optimized Link State Routing Protocol*) [1].

Uma segunda opção é estabelecer um processo de seleção dos nós reencaminhadores da inundação, conforme realizado por algumas propostas [6, 9–11]. Este processo consiste basicamente em uma poda de nós, restringindo a tarefa de reencaminhamento a uma parcela dos nós. Chamaremos a este tipo de solução de mecanismos de redução da inundação.

Ainda é possível controlar os efeitos nocivos das tempestades de *broadcast* através da escolha da ordem em que os nós reencaminhadores irão realizar suas transmissões.

sões [12–17], e que chamaremos de mecanismos escalonadores. Nestes mecanismos, o objetivo é escolher uma sequência de reencaminhamento de forma a reduzir a latência obtida, ou seja, o tempo decorrido entre o início da inundação e o recebimento pelo último nó de destino.

Estas três abordagens podem ser aplicadas todas em conjunto, sendo que a primeira, de limitar o uso e a frequência da inundação, é dependente da aplicação que utiliza a inundação e das características da rede [8]. As outras duas abordagens podem ser consideradas de uso geral, e são o foco de trabalho desta tese.

Além disso, estas duas últimas abordagens, apesar de ortogonais, no sentido de que uma delas busca tão somente reduzir o número de mensagens reencaminhadas e a outra visa reduzir o tempo de entrega da inundação na rede, possuem semelhanças, conforme mostrado nos resultados das Seções 3.2.1 e 4.2. A explicação para esta semelhança reside no fato de que, por um lado, os mecanismos de redução da inundação, ao diminuírem o número de retransmissões, também reduzem o tempo de entrega das mensagens, pela diminuição da concorrência, e por outro lado, pelo fato de mecanismos escalonadores eficientes também realizarem um processo de poda de nós com o objetivo de reduzir a latência.

Em relação às soluções de redução de inundação, elas podem ser agrupadas basicamente em probabilísticas e determinísticas. Os mecanismos determinísticos dependem do conhecimento da topologia existente, pelo menos de forma limitada, como por exemplo, da vizinhança de dois saltos. Entretanto, devido à dificuldade na obtenção deste conhecimento, acabam recaindo em algum tipo de inundação ou troca adicional de mensagens, aumentando a sobrecarga de controle.

Já os mecanismos probabilísticos não dependem do conhecimento preciso da topologia, mas apresentam limitações de desempenho quando aplicados a redes que não correspondem às hipóteses probabilísticas iniciais do mecanismo [18], como, por exemplo, quando são aplicados em redes com distribuição não uniforme dos nós.

Assim sendo, na Seção 3.1 apresentamos a proposta chamada FloorB [19, 20], baseada em um mecanismo híbrido, que assume um conhecimento resumido da topologia em até dois saltos, e utiliza um método probabilístico para realizar o reencaminhamento das mensagens de inundação. Para resumir a representação da vizinhança de dois saltos, o mecanismo faz uso de filtros de Bloom. Tais filtros apresentam propriedades que permitem reduzir a quantidade de informações trocadas entre vizinhos e facilitam a manipulação das informações de vizinhança coletadas.

O mecanismo proposto foi avaliado por simulação, comparativamente a outros mecanismos, tais como inundação cega (*Blind*), inundação probabilística simples (*Gossip*), *Gossip Adaptativo* e *MPR (Multi Point Relays)*, tanto em termos de taxas de entrega, como em reencaminhamentos evitados, utilizando-se cenários com rede em grade sem mobilidade, mobilidade do tipo *Random Waypoint (RWP)* e

mobilidade do tipo social [21].

Entretanto, conforme citado, a tempestade de *broadcast* [6] implica em muitas transmissões redundantes, o que pode levar a uma latência elevada para a realização da inundação. Caso a rede utilize um acesso ao meio por contenção, como o CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*), haverá um alto nível de disputa e aumento de colisões, aumentando o tempo de entrega das mensagens. Sendo o acesso ao meio realizado através de escalonamento no tempo, como no TDMA (*Time Division Multiple Access*), poderá haver a necessidade de um grande número de *slots* para realizar todas as transmissões requeridas. A forma mais comum de abordar este problema é através do escalonamento de inundações.

Esta abordagem busca a escolha de uma ordem de reencaminhamento que leve ao Escalonamento de Inundação com Mínima Latência (*Minimum-Latency Broadcast Scheduling - MLBS*) que é um problema NP-Completo [15]. Porém, nesta tese explicitamos uma observação importante, a de que uma escolha adequada dos nós reencaminhadores pode levar a uma latência da inundação com valores satisfatórios, mesmo não sendo este o problema alvo dos mecanismos para redução de inundação. Para mostrar isto, na Seção 3.2.1, através de simulação, é medida a latência da inundação quando utilizados mecanismos de redução de inundação associados ao escalonamento distribuído aleatório com contenção utilizado no método de acesso ao meio CSMA/CA, adotado pelo padrão IEEE 802.11.

A partir dos resultados apresentados, são formuladas hipóteses, que fundamentaram a proposta de um mecanismo de escalonamento distribuído, baseado em informações locais, com conhecimento limitado da topologia [22, 23], chamado 2Bsheld e apresentado no Seção 3.2. Este mecanismo é avaliado associado ao CSMA/CA mas também em uma rede com acesso ao meio do tipo TDMA com reuso espacial (STDMA - *Spatial TDMA*). Entretanto, no mecanismo proposto o processo de escolha do *slot* utilizado é realizado de forma totalmente independente e local, implicando que no caso de escolhas incorretas de escalonamento ocorram colisões e perdas de transmissões.

## 1.1 Organização do Texto

O Capítulo 1 faz uma introdução ao tema, apresentando os aspectos gerais dos problemas já estudados pela literatura. Diversas soluções existentes para o controle de inundação em redes *ad hoc*, são apresentadas no Capítulo 2. O Capítulo 3 apresenta as propostas do mecanismo de redução de inundação e do mecanismo para realizar escalonamento das mensagens de inundação. No Capítulo 4 são apresentados os resultados numéricos, das duas abordagens, obtidos por simulação, assim como toda metodologia empregada nas avaliações. Por fim, o Capítulo 5 apresenta as con-

clusões gerais, e apresenta oportunidades de continuidade ao trabalho de pesquisa realizado nesta tese.

# Capítulo 2

## Trabalhos Relacionados

Uma forma simples de realizar uma inundação em redes *ad hoc* móveis é através da inundação cega (*Blind*) [2–4, 7], que consiste em identificar de forma única cada mensagem de inundação, e cada nó reencaminhar tal mensagem na primeira vez que a recebe. Com esta ideia, cada nó reencaminha a mensagem de inundação uma única vez, e a recebe tantas vezes quantos forem seus vizinhos. Evidentemente, esta não é uma solução ótima em relação ao número de mensagens enviadas, e pode consumir muitos recursos da rede, principalmente se o número de vizinhos for elevado.

Em relação aos mecanismos de redução da inundação, podemos afirmar que a solução ótima para o problema de inundação a partir de um nó, em termos do número de mensagens reencaminhadas, é calcular a MLST - *Maximum Leaf Spanning Tree* - do grafo, e escolher como raiz da árvore o nó origem da inundação, ou seja, obter a árvore geradora do grafo com o maior número de folhas. Esta solução é ótima em termos de número de mensagens reencaminhadas, pois uma vez que as folhas da árvore não precisam reencaminhar as mensagens, quando o número de folhas é máximo, o número de mensagens reencaminhadas é mínimo.

Entretanto, esta solução não é prática, uma vez que o problema é NP-difícil [24], e portanto o algoritmo possui alta complexidade, além de ser de difícil implementação de forma distribuída. Por exemplo, o algoritmo proposto em [24] para obtenção da MLST de um grafo tem como passo inicial a realização de uma busca em largura, que de forma distribuída poderia ser implementada a partir de uma inundação cega. Além disto, a MLST deveria ser calculada por todos os nós, ou divulgada a todos, a cada instante que seja realizada a inundação, por conta da mobilidade.

Para contornar esta dificuldade, diversas soluções são propostas na literatura, procurando obter inundações com boa cobertura da rede, atraso limitado e número reduzido de mensagens reencaminhadas. Por exemplo, em [6], os efeitos da tempestade de *broadcast* e alguns esquemas para redução da inundação, classificados em probabilísticos, baseado em contagem, baseado em distância, baseado na localização e baseado em agrupamento, são avaliados. Os resultados por simulação apresenta-

dos indicam a efetividade dos mecanismos, mas os mesmos não são projetados para se adaptar a redes com características diversas, como por exemplo, redes onde a distribuição dos nós não seja uniforme.

Em [9], são apresentados detalhadamente, em diversos cenários, os conceitos de reencaminhamento probabilístico, chamado Gossip, e outros esquemas auxiliares para melhoria das taxas de entrega. O funcionamento básico do Gossip é semelhante à inundação cega, somente alterando o fato de que a mensagem é reencaminhada na primeira vez que é recebida com uma probabilidade  $p$ , parâmetro do mecanismo. Já em [18], há uma ampla avaliação do mecanismo probabilístico, frente a diversos parâmetros da rede, como mobilidade, carga, densidade de nós e probabilidade de reencaminhamento. Em ambos os trabalhos, não são considerados outros esquemas de redução de inundação, tais como métodos determinísticos.

O mecanismo RAPID (*ReliAble ProbabIlistic Dissemination protocol*), proposto em [11], apresenta um esquema conjugado de Gossip Adaptativo com um esquema semelhante ao baseado em contagem proposto em [6]. No Gossip Adaptativo, a probabilidade de reencaminhamento é calculada como função do número de vizinhos através da fórmula  $\frac{\beta}{V_1}$ , onde  $V_1$  é o número de vizinhos de 1 salto e  $\beta$  é um parâmetro do mecanismo. O aprendizado de vizinhança ocorre pela troca de mensagens de *hello* simples, ou seja, aquelas não precisam carregar nenhuma informação além da identificação do nó que as envia. A associação destes esquemas apresenta uma boa taxa de entrega e grande economia de transmissão em redes com mobilidade não muito elevada, ao custo de introdução de um atraso considerável para a entrega das mensagens.

O trabalho em [25] propõe um Gossip Adaptativo baseado no conhecimento da vizinhança de 1 salto do receptor, obtido por mensagens de *hello* simples, e da vizinhança do emissor, divulgada junto com a mensagem de inundação. Esta proposta apresenta a vantagem de simplificar e reduzir o tamanho das mensagens de *hello*, quando comparado a soluções determinísticas. Apesar de ter muitas características semelhantes à abordagem que propomos, utilizando os conceitos de conjuntos de vizinhos e a relação entre eles, além da correspondência com área de cobertura, esta abordagem apresenta algumas desvantagens, como provocar o crescimento da mensagem de inundação, e também tomar decisões probabilísticas baseadas somente na comparação das vizinhanças do emissor e receptor, não atuando adequadamente em redes com características não homogêneas.

Ainda dentro do conceito de reencaminhamento probabilístico, o trabalho em [26] apresenta uma heurística bastante eficiente, respaldada por um modelo baseado em grafos aleatórios (*Random Graphs*). Nessa proposta, um nó  $u$  decide enviar uma mensagem de inundação para seu vizinho  $v$  com probabilidade  $p(a, b)$ , onde  $a$  e  $b$  são os graus de  $u$  e  $v$ , respectivamente. Esta probabilidade também é dependente

do parâmetro  $\alpha$ , a probabilidade desejada de que um nó receba a mensagem de inundação, e é expressa por:

$$p(a, b) = \frac{1 - (1 - \alpha)^{1/\min\{a,b\}}}{\alpha} \quad (2.1)$$

Entretanto, esta solução difere significativamente daquela utilizada em redes sem fio de um modo geral. Nas transmissões em redes sem fio, considera-se que uma transmissão de um emissor chegará a todos os seus vizinhos, por exemplo, quando se utiliza um modelo do tipo *Unit Disk Graph* (UDG), a transmissão chegará a todos os nós dentro do alcance de transmissão. Desta forma, podemos dizer que em redes sem fio, para reduzir a redundância na inundação, a poda do grafo representativo da rede deve ser feita por nós, enquanto que no trabalho citado ocorre uma poda por arestas, válida para o caso de transmissões em *unicast*. Apesar desta dificuldade de comparabilidade entre as abordagens, existe razoável semelhança desta proposta com a nossa, motivando a sua implementação e avaliação, realizada em alguns cenários no Capítulo 4.

Outra opção para disseminar informação em redes *ad hoc* é utilizar redes sobrepostas [27–29]. Com estas estruturas, é possível transmitir uma mensagem para toda a rede de forma econômica, ao encaminhar as mensagens somente pela rede sobreposta. Entretanto, a construção e manutenção de redes sobrepostas em ambientes móveis, ou muito dinâmicos, é custosa e normalmente baseada em mensagens de inundação, recaindo no problema inicial.

Soluções determinísticas são possíveis, tais como a obtenção de conjuntos dominantes, *backbones* virtuais e outros [30–33]. Entretanto, novamente, a construção e manutenção destas estruturas, e os algoritmos envolvidos, apresentam complexidade elevada, tanto em tempo quanto em mensagens. Deste modo, uma alternativa muito usual é a adoção de MPRs (*Multi Point Relays*), proposta em [10] e avaliada em alguns trabalhos [34–37]. Esta talvez seja a solução mais adotada para a redução de inundação em redes *ad hoc*, sendo a alternativa utilizada no protocolo de roteamento OLSR (*Optimized Link State Routing Protocol*) [1].

Tanto os resultados obtidos nesta tese, quanto os trabalhos citados, indicam que esta solução apresenta altas taxas de reencaminhamentos evitados, e se adapta bem às mais diversas topologias. A seleção de MPRs consiste na aplicação de uma heurística baseada somente na informação de 2 saltos, que é obtida através da troca de mensagens de *hello* que carregam a tabela de vizinhos do nó que a emite, e do algoritmo simplificado abaixo:

O objetivo deste algoritmo de seleção de MPRs é obter um conjunto reduzido de vizinhos de 1 salto que permite alcançar todos os vizinhos de 2 saltos, conforme mostrado na Figura 2.1, onde o nó central  $O$  seria o emissor, e os nós destacados se-

**Algoritmo de seleção de MPR**

- 1: Enviar(receber) hellos com lista de vizinhos
- 2: Computar listas de vizinhos (de 1 e 2 saltos)
- 3: Selecionar como MPR os vizinhos de 1 salto que são únicos no alcance de algum vizinho de 2-saltos
- 4: Remover da lista os vizinhos de 2-saltos que foram cobertos
- 5: Escolher como MPR o vizinho de 1 salto que cobre o maior número de vizinhos de 2-saltos ainda não cobertos
- 6: Remover da lista os vizinhos de 2-saltos que foram cobertos e, caso a lista não esteja vazia, retornar ao passo anterior

**Algoritmo 1: MPR.**

riam os MPRs utilizados para alcançar todos os vizinhos de 2 saltos. Este algoritmo constrói os conjuntos de vizinhos de 1 e de 2 saltos a partir da recepção dos *hellos*, e faz buscas simultâneas nos dois conjuntos, como quando se obtém os vizinhos de 1 salto que são únicos no alcance de algum vizinho de 2 saltos, e por este motivo a complexidade de tempo é  $O(\Delta^2)$ , onde  $\Delta$  é o grau máximo na rede.

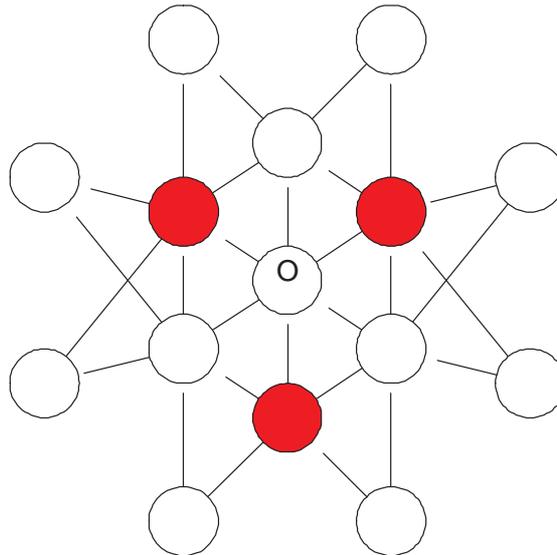


Figura 2.1: Exemplo de Funcionamento de MPR.

Uma observação importante sobre o mecanismo MPR é que à medida que cresce o número de vizinhos de um nó, cresce o número de mensagens de *hello* que ele recebe no mesmo intervalo de tempo, e cresce também o tamanho destas mensagens, pois elas carregam a lista de vizinhos. Desta forma, podemos constatar que neste caso o consumo de banda provocado pelas mensagens de *hello* será proporcional a  $\Delta^2$ . Esta foi uma das motivações para o desenvolvimento da primeira proposta desta tese, conjugada com o fato de que a perda destas mensagens de *hello* geram um grande impacto no desempenho do mecanismo MPR, como mostraremos à frente.

Por outro lado, todas estas soluções de redução da inundação não visam solucionar o problema de controlar a latência. Para isto, é necessário utilizar mecanismos

escalonadores, que executam uma escolha da ordem em que os nós reencaminham as mensagens de inundação. Escalonadores são normalmente associados a um método de acesso ao meio determinístico, como o STDMA, de forma a não depender de temporizações aleatórias e permitir a minimização da latência. Este problema, chamado de Escalonamento de Inundação com Mínima Latência (*Minimum-Latency Broadcast Scheduling - MLBS*) é um problema NP-Completo [15], e portanto os algoritmos que o solucionam são computacionalmente custosos. Soluções de escalonamento através de força bruta, mesmo para redes sem mobilidade, com poucos nós e uma única fonte enviando ou recebendo para alguns destinos, podem se tornar muito custosas pelo esforço computacional demandado [38]. Desta forma, a literatura apresenta algumas abordagens para o problema, tais como algoritmos aproximados, métodos de otimização ou técnicas de inteligência computacional, conforme citados a seguir.

É interessante observar que no problema MLBS é fácil estipular um limite inferior (*lower bound*) que é a latência mínima possível obtida por um algoritmo ótimo em uma rede com topologia adequada. Este limite é proporcional ao raio da rede  $R$ , ou seja, à distância em saltos do nó mais distante em relação à fonte da inundação. Por este motivo, alguns algoritmos propostos na literatura apresentam limites de latência em função de  $R$ , ou de  $R$  e outros parâmetros da rede, como  $R\Delta$  em [12] e  $R \log^2 R/N$  em [13]. Além deste limite inferior, é trivial identificar  $N$ , o número de nós, como um limite superior (*upper bound*), uma vez que sempre será possível realizar um escalonamento que realize a inundação através da atribuição de um *slot* para cada nó.

O problema MLBS pode ser definido como um problema de coloração de grafos [14], onde a solução corresponde ao menor número de cores. Se a rede sem fio for modelada através de *Unit Disk Graphs* (UDG), teremos um modelo de protocolo, onde dois nós não podem ser escalonados simultaneamente caso exista uma aresta entre eles, ou eles possuam pelo menos um vizinho em comum. Outros modelos mais restritivos podem ser utilizados, por exemplo, levando em conta a interferência, também chamado de modelo físico ou baseado em SINR [39].

Em redes que utilizam acesso ao meio do tipo CSMA/CA, o escalonamento é realizado de forma automática e aleatória através da detecção de portadora e do *backoff*. Apesar de eficiente em termos de entrega, o CSMA/CA pode impor atrasos elevados em redes congestionadas, ou na inundação em redes densas ou com grau elevado. Por conta desta restrição, quando há necessidade de uma latência limitada para a entrega da inundação, é conveniente a adoção de um acesso ao meio determinístico, por exemplo, do tipo TDMA. Entretanto, se faz necessário o uso de um esquema, centralizado ou distribuído, para realizar o escalonamento dos nós, ou seja, atribuir qual *slot* TDMA será utilizado pelo nó.

Os algoritmos de escalonamento para inundação têm como objetivo determinar

os nós que serão ativados em cada um dos *slots* de tempo de tal forma que todos os nós tenham pelo menos um enlace escalonado pelo menos uma vez, em um número mínimo de *slots*. Essa sequência mínima de *slots*, que representa o tamanho  $T$  do escalonamento, corresponde ao ciclo de inundação de um método de acesso múltiplo por divisão do tempo e, pelo fato da separação espacial entre os nós permitir a simultaneidade na ativação dos enlaces, é denominado STDMA (*Spatial Time Division Multiple Access*) [38]. Neste contexto, alguns trabalhos propõem, através de diversas técnicas, mecanismos ou algoritmos com esta finalidade.

Em [14], o problema MLBS é tratado através de UDG, e são propostos 3 algoritmos centralizados aproximados chamados BBS (*Basic Broadcast Schedule*), EBS (*Enhanced Broadcast Schedule*) e PBS (*Pipelined Broadcast Schedule*). Os três algoritmos iniciam da mesma forma, a partir da criação da árvore geradora com raiz na fonte da inundação, derivada da busca em largura no grafo da rede. Através da busca em largura também é determinada a distância de todos os nós em relação à fonte e o raio  $R$  da rede. Em seguida é construído o MIS - *Maximal Independent Set* - de  $G$ , induzido pela ordem crescente da profundidade dos nós.

No algoritmo BBS, os autores chamam os nós pertencentes a este MIS de *dominators*, e seus pais na árvore de *connectors*. Somente estes nós reencaminham a inundação, escalonados camada por camada, sendo que na camada zero existe somente a fonte, e na camada  $R$ , nenhum nó é *connector*. Dentro de cada camada, geometricamente, os nós são escalonados de forma a não ocorrerem conflitos. Desta forma, os autores demonstram que o escalonamento produzido por este algoritmo está limitado em  $24R - 23$  *slots*.

O algoritmo EBS já apresenta uma melhoria em relação ao BBS, diferindo na forma como os *connectors* são escolhidos e escalonados. No BBS, todos os nós pais dos *dominators* são escolhidos como *connectors*. No EBS é utilizado o algoritmo IMC (*Iterative Minimal Covering*), produzindo não só um escalonamento menor, como também reduzindo a redundância de transmissões. Os autores demonstram que este algoritmo produz escalonamentos limitados à  $16R - 15$ .

Por último, o algoritmo PBS, após a construção do MIS e seleção dos *dominators*, computa a árvore de menores caminhos da fonte para todos os *dominators*. Em seguida, estabelece através de um algoritmo proposto um ordenamento dos nós por *ranking*, permitindo que os reencaminhamentos ocorram em *pipeline* podendo um nó de uma camada inferior receber e reencaminhar a mensagem antes de um nó de camada superior. Com este algoritmo, o escalonamento fica limitado em  $R + O(\log R)$  *slots*. Entretanto, durante a demonstração deste limite superior, pode-se observar que a latência é limitada a  $R' + O(\log O(R'^3))$ , onde  $R'$  é  $R$  ou  $R - 1$ . Portanto, para efeitos de latência, utilizaremos nas seções seguintes o limite de  $R + \log R^3$  ou  $R + 3 \log R$  para este algoritmo, uma vez que não consideramos a demonstração

deste limite dentro do escopo deste trabalho.

O trabalho em [15] também propõe um algoritmo aproximado, e demonstra que seu limite assintótico é  $O(1)$  vezes a latência ótima. Esse algoritmo também se baseia na construção do MIS derivado da árvore geradora obtida da BFS no grafo. Entretanto, apesar do trabalho definir uma versão distribuída do algoritmo, ela baseia-se no uso de buscas em largura e em profundidade, custosas em termos de mensagens. Além disto, o trabalho em [14] indica que a latência deste algoritmo é limitada em  $648R$ .

Justamente a observação de que limites deste tipo, tais como  $648R$  ou  $24R - 23$ , mesmo sendo considerados para pior caso, podem facilmente ultrapassar o valor de  $N$ , o número de nós da rede, que seria a latência de um escalonamento trivial obtido pela atribuição de um *slot* a cada nó, motivou a elaboração da segunda proposta desta tese, incluindo uma avaliação das latências obtidas pelos mecanismos de redução de inundação.

Entretanto, na literatura existem outras abordagens para este problema, como o trabalho em [17], onde é apresentado um algoritmo que usa vetor de interferência, que permite identificar rapidamente se um nó interferirá com outro. Nesse trabalho o principal objetivo é a execução rápida do algoritmo, de forma a obter um escalonamento com tempo de execução controlável. Não é apresentado limite superior para o escalonamento obtido, somente resultados para escalonamentos quando os nós são ordenados pelo grau ou pelo número de vizinhos de 1 e 2 saltos, com pouca diferença de desempenho entre eles.

Já o trabalho em [40] apresenta outra abordagem para o problema, através da aplicação de lógica *fuzzy* e redes neurais. Em [41], há uma solução através de algoritmos genéticos. Ambos dependem do conhecimento da matriz de conectividade e apresentam resultados de escalonamento somente para 3 redes específicas indicadas na literatura.

# Capítulo 3

## Propostas

Este capítulo apresenta os detalhes das duas propostas que compõem esta tese. Inicialmente, na Seção 3.1, são discutidos os conceitos da proposta do mecanismo de redução da inundação FloorB, em seguida, na Seção 3.2, a proposta do mecanismo de escalonamento de inundação 2BSheld.

### 3.1 FloorB - Representação de Vizinhança por Filtros de Bloom

Nesta seção são apresentados a idéia básica e os conceitos envolvidos na proposta inicialmente descrita em [19]. O mecanismo, denominado FloorB (*Flooding control through neighborhood Representation by Bloom filters*), utiliza um conhecimento resumido da vizinhança de dois saltos e um método probabilístico para o encaminhamento das mensagens.

#### 3.1.1 Vizinhança de 1 e 2 saltos e Reencaminhamento de Mensagem

Um dos conceitos adotados nesta proposta é o de que um nó receptor de uma mensagem de inundação deverá decidir se deve ou não reencaminhar a mensagem baseado no conhecimento aproximado sobre sua vizinhança de 1 e 2 saltos e sobre a vizinhança de 1 salto do nó emissor. A ideia de utilizar somente um conhecimento aproximado advém do fato de que a própria vizinhança é volátil e dinâmica, e o aprendizado sobre ela sofre com interferências, perdas de mensagem e período de atualização. Para definição da proposta, algumas premissas são utilizadas nesta avaliação da vizinhança, que posteriormente serão relaxadas.

Considere uma rede densa com distribuição uniforme dos nós, onde as transmissões são omnidirecionais, com mesma potência, e propagação homogênea, portanto

produzindo uma área de cobertura circular de raio constante. Neste caso, pode-se representar a interação entre dois nós,  $e$  emissor e  $r$  receptor, através da Figura 3.1.

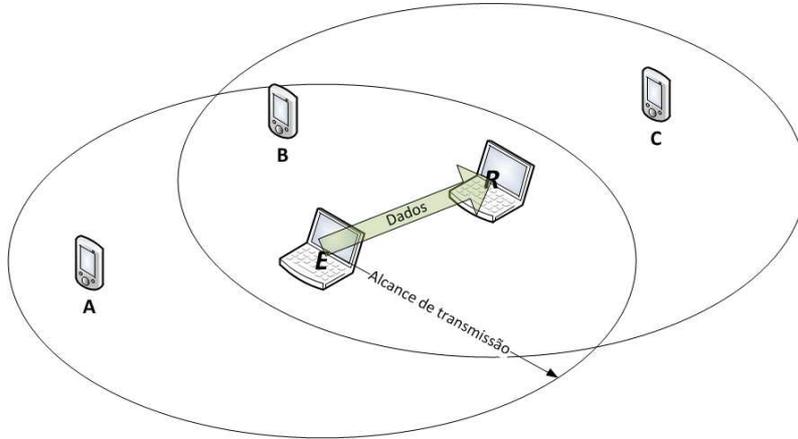


Figura 3.1: Comunicação entre dois nós.

Nesta figura, podemos notar que uma mensagem enviada pelo nó  $e$  é recebida pelo receptor  $r$  e também pelos nós  $a$  e  $b$ , mas não é recebida pelo nó  $c$ . Para a mensagem chegar até ao nó  $c$  é necessário que  $r$  a reencaminhe, e esta decisão pode ser tomada baseada no conhecimento da vizinhança de 1 e 2 saltos, conforme modelado a seguir.

Seja  $V_h(x)$  o conjunto de nós que distam  $h$  saltos do nó  $x$ . Pode-se estabelecer que o conjunto de vizinhos diretos de  $e$  é  $V_1(e)$  (Figura 3.2(a)), e que possui  $|V_1(e)|$  elementos. Analogamente, o conjunto de vizinhos de  $r$  é  $V_1(r)$  (Figura 3.2(b)). Porém, o conjunto de nós que não recebem diretamente uma mensagem enviada por  $e$ , mas podem recebê-la caso  $r$  a reencaminhe, ou seja, o equivalente ao nó  $c$  da Figura 3.1, é dado por  $M = V_1(r) \setminus V_1(e)$  (Figura 3.2(d)). A partir das premissas iniciais, pode-se afirmar que as cardinalidades destes conjuntos são proporcionais às áreas cobertas, conforme descrito em [25].

Com estas considerações, a simples comparação entre a lista de vizinhos do emissor e a lista de vizinhos do receptor fornece uma boa estimativa da necessidade do receptor encaminhar a mensagem ou não. Entretanto, existe uma informação que não é avaliada, uma vez que podem existir nós vizinhos ao emissor, que não são vizinhos do receptor, mas também podem fornecer uma cobertura, ao menos parcial, dos vizinhos do receptor não cobertos pelo emissor, ou seja, alcançar elementos do conjunto  $M$ .

Além disto, pode ocorrer em algumas topologias o fato de que alguns nós que são vizinhos do emissor e receptor simultaneamente, ou seja, pertencentes a  $V_1(e) \cap V_1(r)$  (Figura 3.2(c)), não permitam alcançar nenhum dos nós em  $M$ . Tais nós, portanto, deveriam ser descartados pelo receptor na sua avaliação se ele deve reencaminhar a inundação ou não.

Portanto, o FloorB amplia a avaliação inicial, e a partir da lista completa de vizinhos de cada nó do conjunto  $V_1(r)$  estabelece o conjunto união dos vizinhos de 1 salto dos nós pertencentes a  $M$  como  $U = \{\cup V_1(x) \mid x \in M\}$ , representado na Figura 3.2(e).

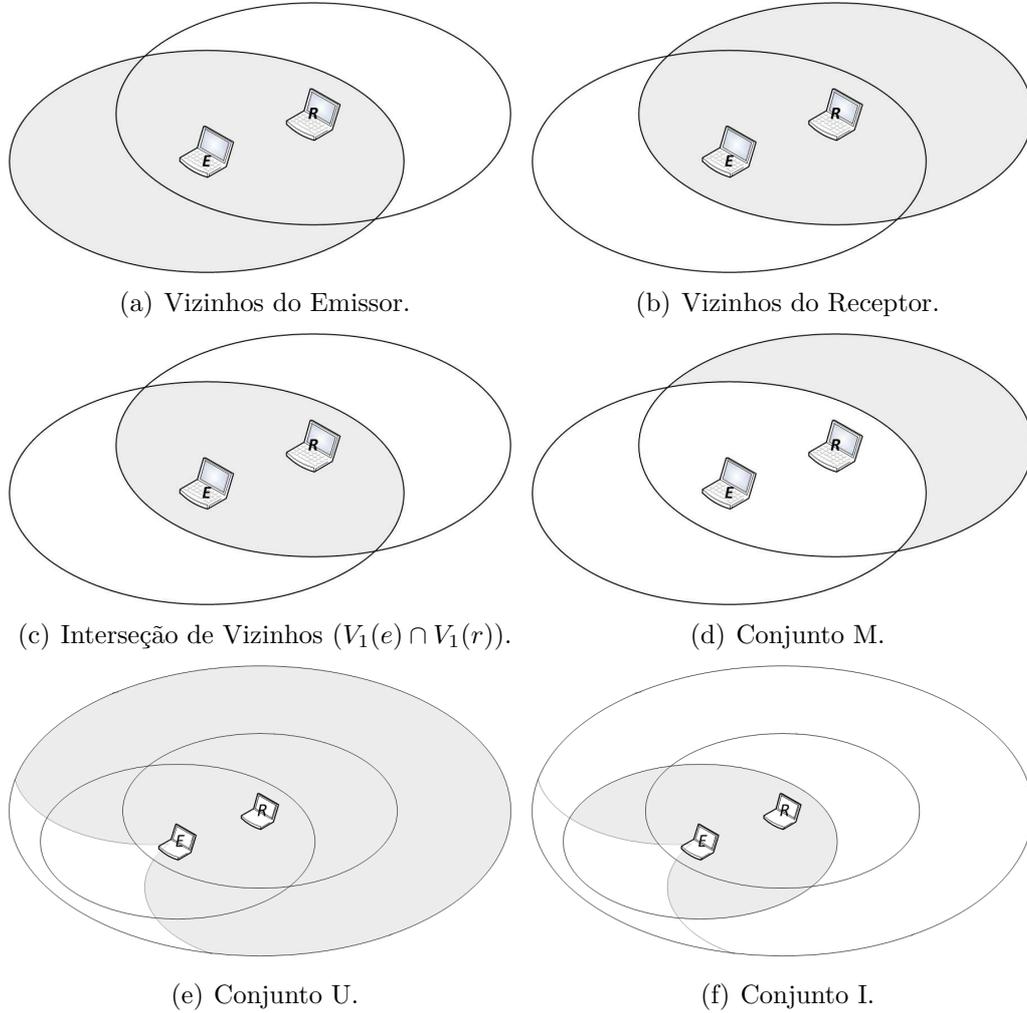


Figura 3.2: Conjuntos de Vizinhaça.

Realizando a interseção deste conjunto  $U$  com os vizinhos de  $e$  obtém-se o conjunto  $I = U \cap V_1(e)$  (Figura 3.2(f)), composto por todos os nós vizinhos de  $e$  que podem alcançar elementos de  $M$ . Com as considerações de distribuição uniforme, rede densa e transmissão omnidirecional, provamos geometricamente na Subseção 3.1.4 que para os nós receptores posicionados no limite do alcance de transmissão do emissor, as cardinalidades destes conjuntos  $I$  e  $M$  se aproximam, e neste caso o nó receptor deve encaminhar a mensagem recebida.

Para relaxar as premissas de distribuição uniforme e rede densa, adotamos um encaminhamento probabilístico, onde a probabilidade de transmissão para receptores que possuem vizinhança muito similar a do transmissor, ou seja,  $|M| \simeq 0$  e  $I \simeq V_1(e)$ , deve ser aproximadamente nula. Quando o receptor está localizado próximo ao limite

do alcance de transmissão do transmissor, a probabilidade de reencaminhamento deve ser aproximadamente 1, e como neste caso  $|M| \simeq |I|$ , utilizamos:

$$p = \frac{|M|}{|I|} \quad (3.1)$$

Na Equação 3.1 é calculada a probabilidade de reencaminhamento da mensagem de inundação em função da cardinalidade dos dois conjuntos  $M$  e  $I$ . Para os casos onde  $|I| = 0$  ou que esta razão é superior a 1, utiliza-se  $p = 1$ .

A principal implicação desta forma de calcular a probabilidade de encaminhamento pela relação entre o número de elementos destes dois conjuntos, comparativamente a outras, é a captura mais precisa de situações excepcionais na topologia da rede. Por exemplo, considere a situação extrema de uma rede composta por dois conjuntos de nós interligados por somente um nó intermediário, como representado na Figura 3.3. Neste caso, o nó  $r$  possui um grande número de vizinhos, o que levaria a uma baixa probabilidade de reencaminhar a mensagem em um mecanismo como o Gossip Adaptativo.

No entanto, os conjuntos  $U$  e  $V_1(e)$  são disjuntos a menos do próprio  $r$ , e portanto, com o FloorB obtém-se  $p = 1$ , conforme desejado. Por raciocínio semelhante, e pela observação de outros cenários, é possível concluir que a obtenção de  $p$  será normalmente adequada, mesmo quando ocorre variação de densidade, topologia específica e redes com alta ou baixa densidade, conforme observado nos resultados apresentados na Seção 3.2.1, ou até mesmo com transmissões não-omnidirecionais ou com obstáculos.

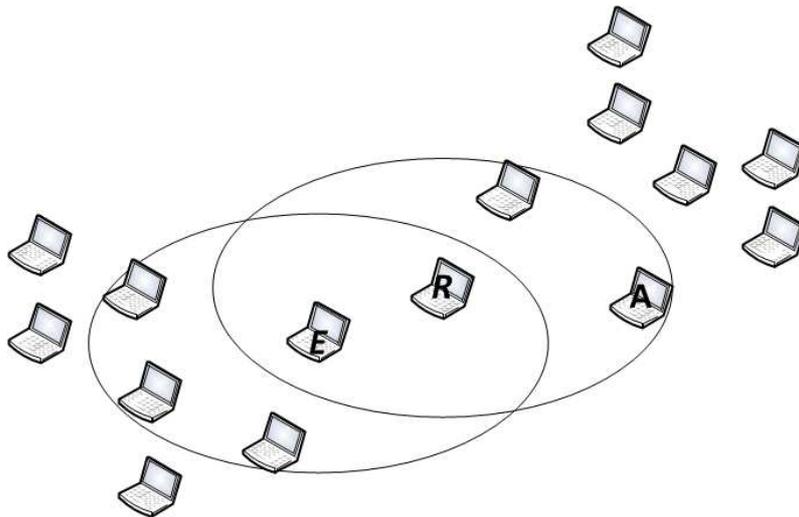


Figura 3.3: Topologia com corte em um nó.

Um questionamento que pode ser feito quanto ao funcionamento deste mecanismo é sobre a obtenção, manutenção e divulgação da tabela de vizinhos de cada nó. Entretanto, este é um procedimento realizado em diversos mecanismos através

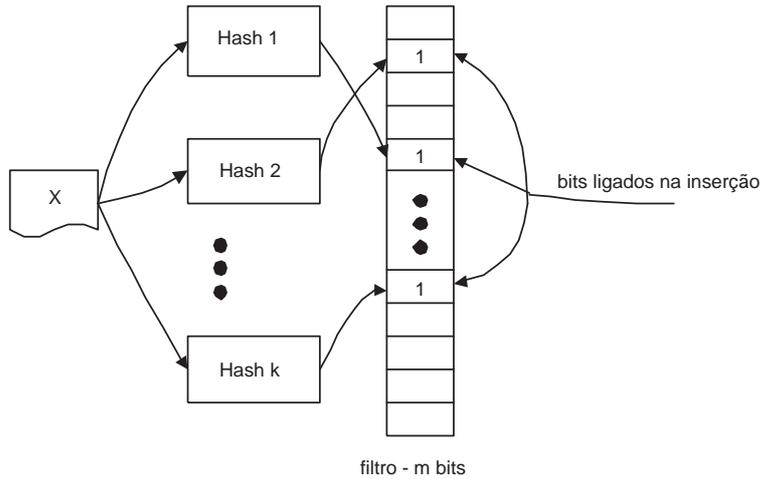


Figura 3.4: Funcionamento do filtro de Bloom.

da troca periódica de mensagens de *hello*. Para se obter a lista de vizinhos de dois saltos ( $V_2(x)$ ) é necessário que cada nó informe sua tabela de vizinhos na mensagem de *hello*, como realizado no mecanismo MPR. Porém, em redes muito densas essas mensagens se tornam extensas e cada nó deverá manter tais listas em memória e processá-las para computar os conjuntos propostos.

Para minimizar este problema, de enviar a lista e obter os conjuntos de vizinhos, utilizando menos recursos, tanto de banda da rede, quanto computacionais, o FloorB utiliza a estrutura de dados Filtros de Bloom, que são descritos na Seção 3.1.2.

### 3.1.2 Filtros de Bloom

Filtros de Bloom [42] são estruturas de dados probabilísticas com espaço reduzido utilizadas para representar um conjunto, e que permitem verificar se um dado pertence ou não a ele de forma rápida. Falsos positivos são possíveis, entretanto com probabilidades controláveis. Os filtros de Bloom têm ampla aplicação em redes de computadores [43, 44].

Um filtro de Bloom é constituído de um vetor de  $m$  bits, e um conjunto de  $k$  funções *hash*, onde cada uma delas mapeia um elemento em uma posição no vetor. Um filtro vazio equivale ao vetor com todos os bits em zero. Para adicionar um elemento ao filtro, aplica-se cada uma das  $k$  funções *hash* ao elemento, obtendo  $k$  posições do vetor que devem ser alteradas para 1, conforme Figura 3.4.

Para consultar se um elemento pertence ao conjunto, calcula-se os  $k$  *hashes* deste elemento, e verifica-se se os bits correspondentes estão ligados no filtro. Caso pelo menos um deles não esteja, tal elemento não pertence ao conjunto. Conforme apresentado em [43], a probabilidade de um falso positivo, ou seja, de que todos os bits correspondentes estejam ligados e de que o elemento não pertença ao conjunto

de  $n$  elementos é dada por

$$P_{false} = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k \quad (3.2)$$

Esta Equação 3.2 é utilizada para obter a Figura 3.5, utilizando-se valores de  $m$  iguais a 128 e 256, e  $k$  igual a 1, 2 ou 3, com número de elementos (vizinhos) variando de 0 a 200.

Nesta figura, podemos observar que os filtros de Bloom com estas configurações apresentam uma probabilidade relativamente baixa de falsos positivos quando o número de vizinhos é inferior a 50, e que a escolha de valores maiores de  $k$  permitem menores probabilidades quando o número de vizinhos é pequeno. Entretanto, com um número maior de vizinhos, acima de 50 com  $m = 128$ , e acima de 100 com  $m = 256$ , valores maiores de  $k$  elevam a probabilidade de falsos positivos.

Por outro lado, a ocorrência destes falsos positivos, quando limitados, não é significativa para o desempenho do mecanismo FloorB, uma vez que ele utiliza uma razão entre o número de elementos de dois conjuntos, e tais erros acabam compensados neste cálculo, conforme mostraremos posteriormente.

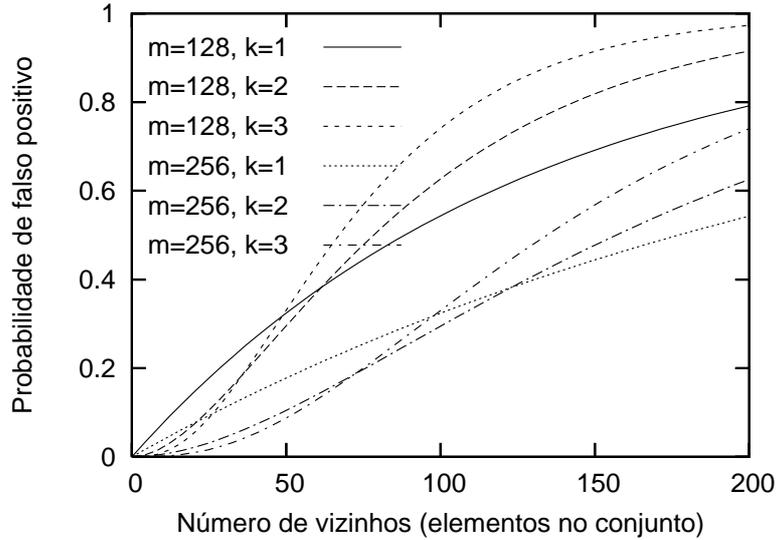


Figura 3.5: Probabilidade de falso positivo.

Também segundo o trabalho em [43], o número de elementos de um conjunto  $A$  pode ser estimado, a partir do filtro correspondente, contando-se o número de bits em 0 ( $\#bits0$ ) no filtro e aplicando a Equação 3.3.

$$|A| = \frac{\log(\frac{\#bits0}{m})}{k \cdot \log(1 - \frac{1}{m})} \quad (3.3)$$

Outra propriedade importante dos filtros de Bloom é a facilidade para se obter o filtro correspondente à união ou interseção de conjuntos, quando se utiliza o mesmo

número de bits  $m$  e as mesmas funções *hash*. O filtro correspondente à união dos conjuntos é obtido pela aplicação simples da operação lógica **OU (OR)** *bit a bit* entre os filtros de cada conjunto. Analogamente, o filtro da interseção pode ser obtido através da operação **E (AND)** sobre os filtros dos conjuntos iniciais, a menos da introdução de alguns falsos positivos. Estas propriedades serão utilizadas pelo FloorB para computar os conjuntos  $U$  e  $I$ , apresentados na Seção 3.1.1, e apresentam uma grande vantagem computacional para o funcionamento do mecanismo.

Podemos notar esta vantagem no caso da obtenção do conjunto  $I$ , onde sem utilizar filtros de Bloom, em termos de algoritmo, seriam necessários dois laços encadeados, onde no externo se percorre um conjunto, por exemplo,  $U$ , e no laço interno o outro conjunto, por exemplo,  $V_1(e)$ , o que leva a uma complexidade de tempo computacional em  $O(\Delta^3)$ . Utilizando filtros de Bloom, esta tarefa se resume a fazer o **E bit a bit** dos filtros, que tem complexidade de tempo em  $O(1)$ .

Em relação às funções *hashes*, seguimos o conceito de *hashes* universais [45] da mesma forma que o trabalho em [46]. Ou seja, as funções *hash* serão da família de funções  $H$ , tais que

$$H = \{h_{a,b}(x) \mid 0 < a < P \text{ e } 0 \leq b < P \text{ e } x < P\} \quad (3.4)$$

sendo que  $P$  é um número primo e que

$$h_{a,b}(x) = [(ax + b) \pmod{P}] \pmod{m} \quad (3.5)$$

Como  $P$  deve ser maior que o maior identificador de nó que será utilizado, e como tanto no nosso simulador próprio quanto no ns-2 estes identificadores são sequenciais até o número de nós utilizado, utilizamos como  $P$  o número de Fermat 4,  $F_4 = 65537$  [47].

### 3.1.3 FloorB - Algoritmo

O algoritmo adotado pelo FloorB para executar as tarefas descritas na Seção 3.1.1, utilizando os conceitos de filtro de Bloom descritos na Seção 3.1.2, é apresentado no Algoritmo 2.

O passo 1 deste algoritmo corresponde à troca periódica de mensagens de *hello*, onde cada nó  $x$  passa a informação  $\text{filtro}(V_1(x))$  para seus vizinhos. O conjunto  $V_1(x)$  é construído localmente pela recepção dos mesmos *hellos*. Em seguida, todas as informações descritas na Seção 3.1.1 são obtidas das propriedades dos filtros de Bloom. Ao receber uma inundação (passo 2), é feita a verificação de pertinência de  $e$  nos filtros armazenados no receptor (passos 3 e 4), e é povoado o conjunto  $M$  (passo 5). Com este conjunto calculado, são obtidos os filtros correspondentes

**Notação:**

$e$  = nó emissor;  $r$  = nó receptor;  
 $filtro(A)$  = filtro de Bloom do conjunto  $A$ ;  
 $V_h(x)$  = conjunto de nós que distam  $h$  saltos do nó  $x$ ;  
 $M = V_1(r) \setminus V_1(e)$ ;  
 $U = \{\bigcup V_1(x) \mid x \in M\}$ ;  
 $I = U \cap V_1(e)$ ;

**Valores iniciais:**

$filtro(U) = filtro(\emptyset)$ ;  
 $M = \emptyset$

**Algoritmo**

- 1: Envio de hellos com filtro de Bloom calculado a partir da lista de vizinhos (recepção de hellos)
- 2: Ao receber uma mensagem de inundação de  $e$ ,
- 3: **Para cada**  $v \in V_1(r)$ , faça:
- 4:     **Se**  $e$  não é vizinho de  $v$  (bits correspondentes não ligados em  $filtro(V_1(v))$ ), faça:
- 5:         Adicionar  $v$  a  $M$
- 6:     **fim Se**
- 7: **fim Para cada**
- 8: **Para cada**  $m$  pertencente a  $M$ , faça:
- 9:      $filtro(U) = filtro(U)$  OU  $filtro(V_1(m))$
- 10: **fim Para cada**
- 11:  $filtro(I) = filtro(U)$  E  $filtro(V_1(e))$
- 12: estimar  $|M|$  e  $|I|$  (equação 3.3)
- 13: calcular  $p$  de reencaminhamento (equação 3.1)
- 14: realizar reencaminhamento com probabilidade  $p$

**Algoritmo 2:** FloorB.

aos conjuntos  $U$  e  $I$  (passos 9 e 11), estimada a cardinalidade dos conjuntos  $M$  e  $I$  (passo 12) e calculada a probabilidade  $p$  (passo 13).

Tal algoritmo apresenta algumas vantagens sobre outros métodos para controlar inundação. Inicialmente, ele produz mensagens de *hello* de tamanho fixo para divulgar a relação de vizinhos. Este tamanho fixo pode ser menor que uma lista completa, quando a rede for densa, uma vez que com o filtro pode-se resumir a vizinhança. Por exemplo, ao utilizar um filtro de 128 bits para relacionar uma vizinhança de 10 nós, teremos uma redução de 60% no tamanho da mensagem, em relação à mensagem completa, quando considerados endereços de 32 bits ( $10 \times 32\text{bits} = 320\text{bits}$ ). Além disto, o algoritmo é mais simples que o adotado pelo MPR, pois percorre uma única vez a lista de vizinhos dos nós, e portanto, sua complexidade de tempo é  $O(\Delta)$ .

Mesmo considerando que as mensagens de *hello* são normalmente curtas e pouco relevantes para o tráfego da rede, a utilização de mensagens de tamanho fixo ou bem definido apresenta algumas vantagens para implementação ou otimização das rotinas que as manipulam, principalmente se parte das tarefas são realizadas por *hardware* específico.

Por outro lado, o algoritmo do FloorB captura características de topologias diversas, o que não ocorre nos mecanismos do Gossip ou Gossip Adaptativo, que têm desempenho reduzido em situação de redes densas mas que possuam algum nó res-

ponsável pela interligação de duas partições da rede, como a topologia apresentada na Figura 3.3.

Na seção 4.1 é apresentado através de simulação o desempenho do FloorB. Apesar da abordagem proposta em [25] não ter sido objeto de comparação em nossas simulações, também foram implementadas outras duas abordagens intuitivas, semelhantes a nossa proposta. Uma, derivada do conceito de *Border Node Retransmission*, apresentado em [25], ou seja, onde nós situados mais próximos ao limite do alcance de transmissão devem possuir maior probabilidade de retransmissão, uma vez que provavelmente possuem menos vizinhos em comum com o transmissor. Esta nossa implementação pressupõe o conhecimento da distância  $d$  entre os nós, e realiza uma transmissão probabilística onde  $p = \frac{d}{R_t}$ , sendo  $R_t$  o alcance de transmissão dos nós. O segundo mecanismo é derivado diretamente de nossa proposta, e utiliza o mesmo conceito, mas de forma diferente, calculando a probabilidade de reencaminhamento como a relação entre as áreas referentes aos conjuntos  $I$  e  $M$ , levando em conta um alcance de transmissão constante  $R_t$ , a distância entre os nós  $d$ , e que tais áreas podem ser obtidas geometricamente pelas interseções de circunferências conforme modelo apresentado na Subseção 3.1.4 a seguir.

### 3.1.4 FloorB - Modelo Geométrico

Conforme sugerido na Subseção 3.1.1, em um método de reencaminhamento probabilístico, a probabilidade de reencaminhamento deve ser proporcional ao número de novos nós que poderão ser alcançados no caso de se realizar o reencaminhamento. A primeira hipótese sugerida é calcular a probabilidade como  $p = d/R_t$ , sendo  $d$  a distância entre os nós e  $R_t$  o alcance de transmissão. Entretanto, considerando uma rede com distribuição uniforme dos nós na área, é possível verificar que o número de nós que receberão a mensagem é proporcional ao quadrado desta relação, ou seja, seria razoável adotar  $p \sim \frac{d^2}{R_t^2}$ .

Um estudo mais acurado deste cálculo de  $p$  pode ser obtido através do determinação geométrica das áreas referentes aos conjuntos  $I$  e  $M$ . Inicialmente, a área referente ao conjunto  $M$  pode ser observada na Figura 3.6, e definida como a diferença entre a área do círculo de alcance e a área da interseção dos dois círculos de raio  $R_t$  e centros  $O_1$  e  $O_2$ , correspondentes à localização dos nós transmissor e receptor. Neste caso temos que  $d$  é a distância entre  $O_1$  e  $O_2$ .

A área da interseção é duas vezes a área do segmento de círculo definido em cada círculo pela corda  $\overline{AB}$  que liga os dois pontos de interseção. Esta área do segmento é igual à área do setor, de ângulo  $\alpha$ , definido pelos dois pontos  $A$  e  $B$ , menos a área do triângulo  $\triangle O_1AB$ .

Como  $\overline{O_1A}$  e  $\overline{O_1B}$  são raios, o triângulo é isósceles, e portanto a área do triângulo

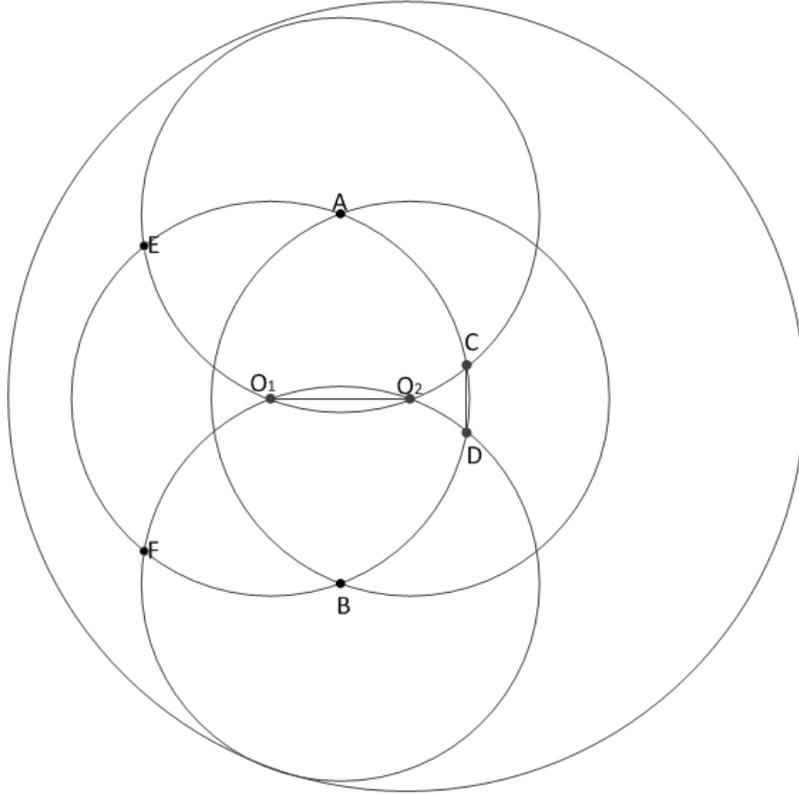


Figura 3.6: Descrição geométrica dos conjuntos de vizinhança.

pode ser calculada como  $\frac{\overline{O_1O_2} \cdot \overline{AB}}{2}$ . Entretanto,  $\overline{O_1O_2} = 2R_t \cos(\alpha/2)$ . E temos também que:

$$\frac{\overline{AB}}{2} = R_t \sin(\alpha/2)$$

Então a área do triângulo  $\triangle O_1AB$  será:

$$S(\triangle O_1AB) = R_t^2 \sin\left(\frac{\alpha}{2}\right) \cos\left(\frac{\alpha}{2}\right) = \frac{R_t^2 \sin(\alpha)}{2} \quad (3.6)$$

Como a área do setor é  $\frac{R_t^2 \alpha}{2}$ , então a área do segmento de círculo  $S(Seg)$  é

$$S(Seg) = \frac{R_t^2}{2} (\alpha - \sin(\alpha)) \quad (3.7)$$

Portanto, a área da interseção,  $S(\cap)$ , será:

$$S(\cap) = R_t^2 (\alpha - \sin(\alpha)) \quad (3.8)$$

Ao mesmo tempo sabemos que a distância dos dois nós é

$$d = \overline{O_1O_2} = 2R_t \cos\left(\frac{\alpha}{2}\right)$$

Então temos

$$\cos\left(\frac{\alpha}{2}\right) = \frac{d}{2R_t}$$

Logo

$$\alpha = 2 \arccos\left(\frac{d}{2R_t}\right)$$

Desta forma é possível calcular a área de interseção  $S(\cap)$  em função de  $d$  e  $R_t$ .

$$S(\cap) = R_t^2(2 \arccos(\frac{d}{2R_t}) - \sin(2 \arccos(\frac{d}{2R_t})))$$

$$S(\cap) = R_t^2(2 \arccos(\frac{d}{2R_t}) - 2 \sin(\arccos(\frac{d}{2R_t})) \cdot \frac{d}{2R_t})$$

$$S(\cap) = R_t^2(2 \arccos(\frac{d}{2R_t}) - \frac{d}{R_t} \sqrt{1 - d^2/4R_t^2})$$

E finalmente, a área referente ao conjunto  $M$  será

$$S(M) = \pi R_t^2 - S(\cap)$$

ou

$$S(M) = \pi R_t^2 - R_t^2(2 \arccos(\frac{d}{2R_t}) - \frac{d}{R_t} \sqrt{1 - d^2/4R_t^2}) \quad (3.9)$$

A dedução da área referente ao conjunto  $I$ ,  $S(I)$ , já demanda uma observação mais criteriosa da Figura 3.6. A primeira observação a ser feita é que devemos obter a área referente ao conjunto  $U$ , o conjunto dos vizinhos de todos os elementos do conjunto  $M$ . Este conjunto  $U$  pode ser obtido primeiramente considerando o círculo de centro  $O_2$  e raio  $2R_t$ . Em seguida, é necessário obter os círculos de raio  $R_t$  com centros em  $A$  e  $B$ . Então podemos observar que a  $S(I)$  corresponde à soma das interseções dos círculos de raio  $R_t$  e centros  $O_1$ ,  $A$  e  $B$ , menos a interseção entre os círculos de centro  $A$  e  $B$  e ainda adicionado o triângulo  $\triangle O_2CD$ , e subtraindo os segmentos de círculo  $O_2C$  e  $O_2D$ , e adicionando o segmento de círculo  $CD$ .

De forma resumida, podemos definir:

- $Seg_1$  = segmento definido pela corda  $\overline{CE}$  = segmento definido pela corda  $\overline{DF}$ , nos círculos de centro  $A$  e  $B$ .
- $Seg_2$  = segmento definido pela corda  $\overline{O_1O_2}$  nos círculos de centro  $A$  e  $B$ .
- $\triangle O_2CD$
- $Seg_3$  = segmento definido pela corda  $\overline{O_2C}$  = segmento definido pela corda  $\overline{O_2D}$ , nos círculos de centro em  $A$  e  $B$ .
- $Seg_4$  = segmento definido pela corda  $\overline{CD}$  no círculo de centro  $O_1$ .

Então teremos que:

$$S(I) = 4S(Seg_1) - 2S(Seg_2) + S(\triangle O_2CD) - 2S(Seg_3) + S(Seg_4) \quad (3.10)$$

A área do segmento  $Seg_1$  corresponde à dedução da Equação 3.7, quando  $d = R_t$ , uma vez que as distâncias de  $A$  e  $B$  para  $O_1$  são o alcance de transmissão. Portanto:

$$S(Seg_1) = \frac{R_t^2}{2}(2 \arccos(\frac{d}{2R_t}) - \frac{d}{R_t} \sqrt{1 - d^2/4R_t^2}) = \frac{R_t^2}{2}(2 \arccos(\frac{1}{2}) - \sqrt{1 - \frac{1}{4}})$$

ou:

$$S(\text{Seg}_1) = \frac{R_t^2}{2} \left( \frac{2\pi}{3} - \frac{\sqrt{3}}{2} \right)$$

Já para o segmento  $\text{Seg}_2$ , temos que a corda definida possui tamanho  $d$ , a distância entre  $O_1$  e  $O_2$ . Utilizando a equação 3.7 reescrita para o ângulo  $\beta$ , definido por  $\widehat{O_1A O_2}$ , temos:

$$S(\text{Seg}_2) = \frac{R_t^2}{2} (\beta - \sin(\beta))$$

E o ângulo  $\beta$  pode ser escrito como:

$$\beta = 2 \arcsin \frac{d}{2R_t}$$

Portanto

$$S(\text{Seg}_2) = \frac{R_t^2}{2} \left( 2 \arcsin \frac{d}{2R_t} - \sin(\beta) \right) = \frac{R_t^2}{2} \left( 2 \arcsin \frac{d}{2R_t} - \frac{d}{R_t} \sqrt{1 - \frac{d^2}{4R_t^2}} \right)$$

Para  $S(\triangle O_2CD)$  começamos observando que o ângulo  $\widehat{O_2AC}$  é central, e portanto o dobro do ângulo  $\widehat{CO_1O_2}$  que é inscrito na mesma circunferência. Por simetria, temos o mesmo resultado para a parte inferior da figura, e constatamos que o ângulo  $\widehat{CO_1D}$  é igual aos ângulos  $\widehat{O_2AC}$  e  $\widehat{O_2BD}$ , e portanto o triângulo  $\triangle O_2CD$  é equilátero, cujo lado chamaremos de  $c$ . Por outro lado o  $\triangle O_1AC$  é equilátero por construção, e portanto, seus ângulos equivalem a  $\frac{\pi}{3}$ . Então podemos notar que  $\widehat{O_2AC} + \beta = \frac{\pi}{3}$ , logo:

$$\widehat{O_2AC} = \frac{\pi}{3} - \beta = \frac{\pi}{3} - 2 \arcsin \frac{d}{2R_t}$$

Como  $c$  é o comprimento de uma corda correspondente ao arco de ângulo  $\widehat{O_2AC}$ , podemos escrever  $c$  como

$$c = 2R_t \sin\left(\frac{\pi}{6} - \arcsin \frac{d}{2R_t}\right)$$

E então podemos escrever

$$S(\triangle O_2CD) = \frac{c^2 \sqrt{3}}{4} = R_t^2 \sin^2\left(\frac{\pi}{6} - \arcsin \frac{d}{2R_t}\right) \frac{\sqrt{3}}{4}$$

Pela dedução de que este triângulo é equilátero, também concluímos que os segmentos  $\text{Seg}_3$  e  $\text{Seg}_4$  são iguais, e correspondem ao segmento do ângulo  $\widehat{O_2AC}$ , logo podemos escrever:

$$S(\text{Seg}_3) = S(\text{Seg}_4) = \frac{R_t^2}{2} \left( \frac{\pi}{3} - 2 \arcsin \frac{d}{2R_t} - \sin\left(\frac{\pi}{3} - 2 \arcsin \frac{d}{2R_t}\right) \right)$$

E finalmente podemos escrever a área correspondente ao conjunto  $I$  como:

$$\begin{aligned} S(I) = & 4 \left[ \frac{R_t^2}{2} \left( \frac{2\pi}{3} - \frac{\sqrt{3}}{2} \right) \right] - \\ & [R_t^2 (2 \arcsin \frac{d}{2R_t} - \frac{d}{R_t} \sqrt{1 - \frac{d^2}{4R_t^2}})] + \\ & R_t^2 \sqrt{3} \left( \sin^2 \left( \frac{\pi}{6} - \arcsin \frac{d}{2R_t} \right) \right) \frac{\sqrt{3}}{4} - \\ & \frac{R_t^2}{2} \left( \frac{\pi}{3} - 2 \arcsin \frac{d}{2R_t} - \sin \left( \frac{\pi}{3} - 2 \arcsin \frac{d}{2R_t} \right) \right) \end{aligned} \tag{3.11}$$

Esta expressão certamente pode ser simplificada, mas tal tarefa foge do escopo

de nosso trabalho. Entretanto, apresentamos na Figura 3.7 os valores obtidos para  $p$ , a probabilidade de reencaminhamento, utilizando as três relações:  $\frac{d}{R_t}$ ,  $\frac{d^2}{R_t^2}$ , e  $\frac{S(M)}{S(I)}$ .

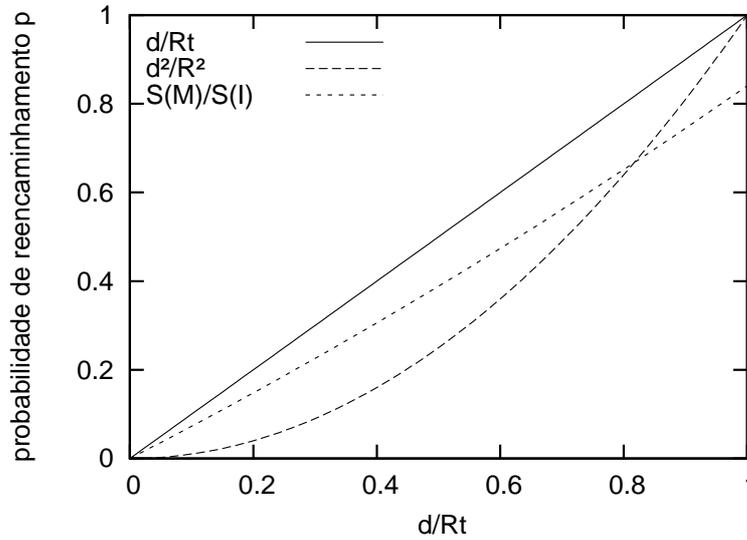


Figura 3.7: Valores de  $p$  nos modelos geométricos.

A primeira observação sobre esta figura é que podemos notar que a relação entre as áreas  $S(M)$  e  $S(I)$  é linear. Esta razão se aproxima de 0 para nós à pequena distância um do outro, e chega a 0,85 quando o receptor se localiza no alcance de transmissão do emissor.

Porém, é muito importante entender que o FloorB não é baseado nesta relação entre áreas, mas sim na relação entre o número estimado de elementos de dois conjuntos,  $M$  e  $I$ , e portanto, se aplica mesmo em situações onde há variação de densidade de nós por área, e também não sofre problemas como efeitos de borda, como ocorre no caso de se usar  $p = \frac{S(M)}{S(I)}$ . Com isto, não é um problema para o funcionamento do FloorB o fato desta relação entre áreas não alcançar o valor 1.

## 3.2 2BSheld - Escalonamento de Inundação com Visão e Decisão Local

Nesta seção, são apresentados os conceitos e as hipóteses utilizados na elaboração da proposta do mecanismo escalonador de inundações baseado no conhecimento de vizinhança de 2 saltos com decisão local chamado de 2BSheld.

### 3.2.1 Avaliação de Mecanismos de Redução da Inundação

Na Seção 4.1 e nos trabalhos em [19] e [20], é apresentada uma extensa avaliação dos mecanismos de redução de inundação Blind, Gossip, Gossip Adaptativo, MPR

e FloorB, quanto à taxa de entrega e reencaminhamentos evitados. Estes mecanismos não realizam escalonamento, conseqüentemente não visam reduzir a latência da inundação. Entretanto, nesta seção apresentamos outra avaliação dos mesmos mecanismos quanto à latência, comparando os resultados com os limites teóricos de algoritmos aproximados e o limite inferior  $R$ .

Para esta avaliação, foi utilizada a implementação dos mecanismos no simulador ns-2 [48], aplicada sobre uma rede em grade com 400 nós, dispostos em 20 linhas e colunas, espaçadas de 100m. A inundação é iniciada pelo nó localizado em um canto da área (Nó 0) e diversos graus de conectividade são obtidos alterando-se os alcances de transmissão desde 110 até 490m, conforme Figura 3.8, utilizando-se o padrão IEEE 802.11 com os valores *default* do ns-2, indicados na Tabela 3.1.

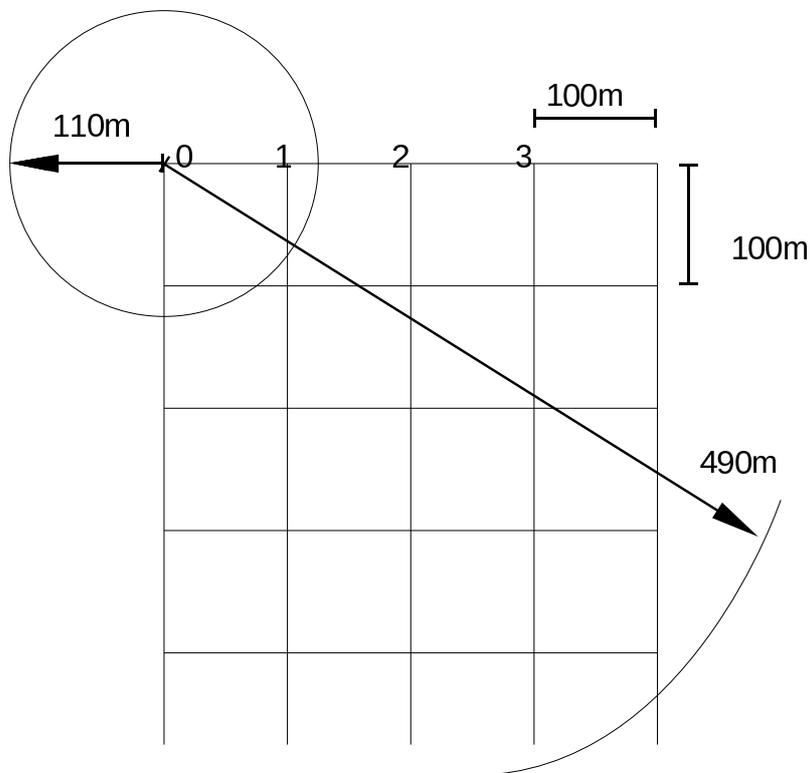


Figura 3.8: Cenário em grade.

Foram realizadas 10 rodadas de simulação, calculando-se a média da latência da inundação e barra de erros com intervalo de confiança de 95%, conforme apresentado na Figura 3.9(a).

A latência é medida como o intervalo de tempo decorrido entre o início da transmissão pela origem da mensagem de inundação e a recepção da mesma mensagem pelo último nó que a recebe, ou seja, os nós que não recebem a mensagem não são considerados.

Para permitir a comparação adequada entre todos os mecanismos, independente do tipo de acesso ao meio utilizado, a latência é normalizada em termos de tempo de

Tabela 3.1: Parâmetros da Simulação.

Propagação	TwoRayGround
MAC	802_11
dataRate_	11Mbps
basicRate_	2Mbps
Potência de transmissão	calculada de acordo com alcance utilizado
CS Threshold	Potência de transmissão / 16
Intervalo entre Hellos	3 s
Envio da inundação	99 s

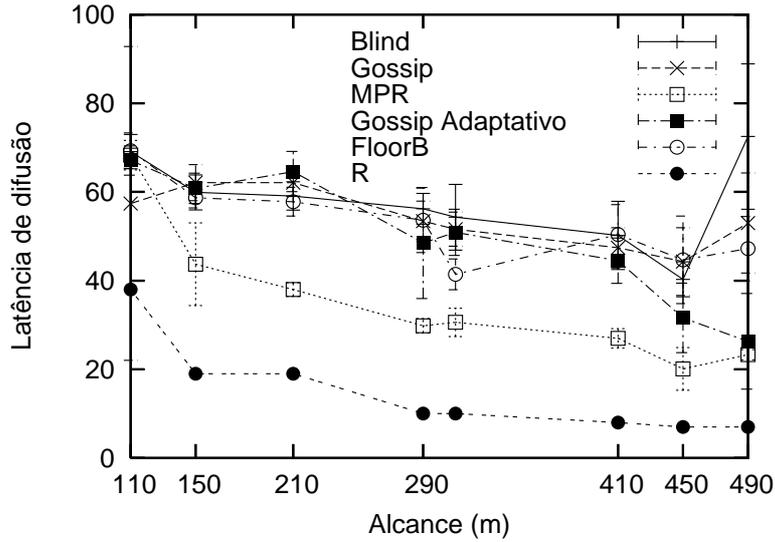
transmissão de um pacote padrão. Desta forma, acréscimos de *backoff* ou contenções do CSMA/CA se refletem em aumento do tempo normalizado para transmissão do pacote. No caso do TDMA, este tempo padrão também é a duração de um *slot*, e portanto será uma medida correspondente de desempenho para este acesso ao meio.

Em cada rodada é realizada uma única inundação após um período de trocas de mensagens nos mecanismos que utilizam *hello*, sendo que tais *hellos* são desabilitados antes do envio da inundação, de forma a não concorrer com a mesma.

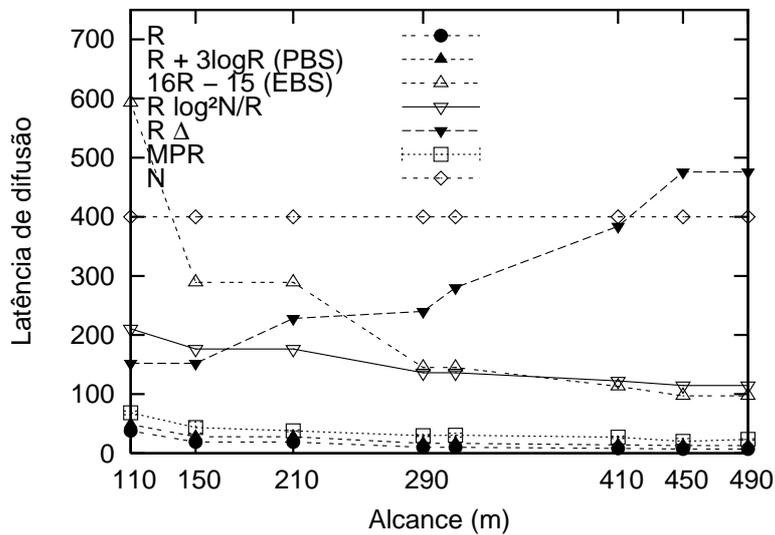
Por este gráfico, pode-se observar que todos os mecanismos não apresentam uma latência muito elevada, e possuem resultados semelhantes. Destaca-se nesta figura o desempenho do mecanismo Blind, que apresenta uma latência elevada para redes com maior grau de conectividade, principalmente alcance de transmissão de  $490m$ , conforme esperado. Também é importante notar a redução da latência apresentada pelo Gossip Adaptativo com alcances maiores, explicada pela redução da entrega deste mecanismo para esses alcances, conforme será apresentado no Capítulo 4. Por último, destacamos o desempenho do MPR, que apresenta resultado melhor que todos os demais.

A latência elevada do Blind ocorre pelo excesso de transmissões e a consequente contenção no acesso ao meio CSMA/CA do 802.11. Já o mecanismo MPR apresenta latência menor mesmo nas redes com maior grau, não ficando distante dos valores limites ( $R$ ). É importante observar que o mecanismo MPR foi projetado para reduzir o número de mensagens reencaminhadas, e não a latência. O bom desempenho é resultado da escolha de poucos nós para reencaminhar, e da escolha de nós que fornecem boa cobertura da vizinhança de 2 saltos.

Para efeitos de comparação, a Figura 3.9(b) apresenta os limites superiores de alguns algoritmos aproximados, incluindo os limites  $R$  e  $N$ , além dos resultados médios do MPR da figura anterior. Podemos notar que neste cenário os limites apresentados pelos algoritmos são extremamente conservadores, estando mais próximos do limite superior do que do inferior, exceto pelo algoritmo PBS de [14]. Além



(a) Latência de mecanismos sem escalonamento



(b) Limites de latência de algoritmos aproximados

Figura 3.9: Latências de inundação.

disto, as latências fornecidas pelos mecanismos de redução de inundação, sem escalonamento, mostram-se pouco distantes do limite inferior, mesmo não tendo sido desenvolvidos com esta finalidade.

Portanto, pode-se supor que uma heurística de redução de nós reencaminhadores, tal como MPR, também permite uma redução na latência final de entrega da inundação. A explicação para isto, neste cenário com CSMA/CA, deve-se à redução da contenção, e também ao fato de que o mecanismo MPR permite uma escolha de poucos nós reencaminhadores, portanto escalonáveis em menor tempo, que permitam a total cobertura dos vizinhos de 2 saltos, e este processo se reproduz salto a salto, até cobrir toda a rede. Desta observação, formulamos a primeira hipótese para fundamentar a nossa proposta de abordagem ao problema MLBS.

*Hipótese 1: Apesar de distintas, as soluções dos problemas MLBS e MLST levam a respostas próximas, e portanto, a solução de um problema oferece uma solução, se não ótima, pelo menos próxima do ótimo, para o outro problema.*

A partir desta Hipótese 1, foi proposto um mecanismo simples e distribuído para escalonamento de inundação em redes *ad hoc* móveis, descrito na Seção 3.2.2.

### 3.2.2 Mecanismo Proposto

O mecanismo proposto, chamado 2BSheld (*2-hop aware Broadcast Scheduling with local decision*), é distribuído e baseia-se no conhecimento da vizinhança de 2 saltos por cada nó da rede. Este conhecimento pode ser obtido através do envio e recebimento de mensagens de *hello* que carregam a lista de vizinhos do nó emissor, como no caso do MPR, ou uma lista resumida através de filtro de Bloom, como no caso do FloorB.

A partir destas listas de vizinhança de 1 e 2 saltos, aplica-se um algoritmo *guloso* simples, onde é escalonado primeiro o nó receptor que permite alcançar o maior número de nós que não receberam a inundação a partir do reencaminhador. Em caso de empate, os vizinhos que empataram são escalonados ordenados de forma crescente pelo seu identificador.

A Figura 3.10 mostra um exemplo de funcionamento do 2BSheld, onde o nó  $o$  inicia a inundação no *slot*  $s$  e os nós  $v_1$ ,  $v_2$  e  $v_3$  são os vizinhos de 1 salto. O nó  $v_2$  alcança 3 novos nós ( $w, y, z$ ), o nó  $v_1$  alcança 2 novos nós ( $x$  e  $y$ ), e o nó  $v_3$  alcança somente 1 novo nó ( $u$ ), portanto o nó  $v_2$  é escalonado no *slot*  $s + 1$ , o nó  $v_1$  no *slot*  $s + 2$  e o nó  $v_3$  no *slot*  $s + 3$ .

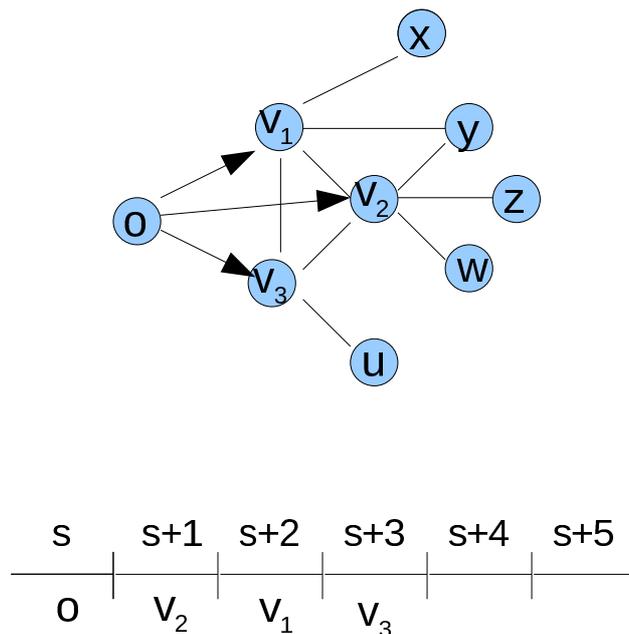


Figura 3.10: Exemplo de funcionamento do 2Bsheld.

Esta heurística baseia-se, além da Hipótese 1, nas Hipóteses 2 e 3 formuladas a seguir, e o algoritmo básico distribuído do 2BSheld é apresentado em Algoritmo 3.

*Hipótese 2: Nós que permitem alcançar um número maior de novos nós com seu reencaminhamento da inundação, permitem também que um maior percentual de nós da rede sejam cobertos pela inundação naquele slot.*

*Hipótese 3: Nós que possuem menos vizinhos em comum com o emissor, devem estar a uma maior distância física dele, e portanto, nós deste tipo e que não são vizinhos entre si, devem estar a uma grande distância, sofrendo menor interferência mútua.*

**Notação:**

$e$  = nó emissor;  $r$  = nó receptor;

$lowPri$  = prioridade do nó por melhor cobertura (menor valor, maior prioridade);

$lowId$  = prioridade do nó por apresentar menor Id;

$V_h(v)$  = conjunto de nós que distam  $h$  saltos do nó  $v$ ;

$Id(x)$  = identificador único do nó  $x$  na rede;

$next$  = número de *slots* a frente para escalonamento

$s(x, y)$  = função empírica para cálculo dos *slots* em função de  $lowPri$  e  $lowId$

---

**Valores iniciais:**  $lowPri = lowId = 0$

---

**Algoritmo**

- 1: Em cada nó  $r$  execute:
- 2:   Ao receber uma mensagem de inundação de  $e$  no *slot*  $s$ , faça:
- 3:     **Para cada**  $v \in (V_1(r) \cap V_1(e))$ , faça:
- 4:       **Se**  $|V_1(v) - V_1(e)| > |V_1(r) - V_1(e)|$
- 5:          $lowPri ++$
- 6:       **caso contrário**
- 7:        **Se**  $|V_1(v) - V_1(e)| == |V_1(r) - V_1(e)|$  e  $Id(v) < Id(r)$
- 8:          $lowId ++$
- 9:        **fim Se**
- 10:     **fim Se**
- 11:    **fim Para cada**
- 12:     $next = s(lowPri, lowId)$
- 13:    escalonar reencaminhamento da inundação em  $s + next$

**Algoritmo 3:** 2BSheld.

O conceito envolvido na Hipótese 2 é o de um algoritmo guloso. No primeiro salto da inundação, a hipótese é claramente válida, uma vez que são escalonados primeiro os nós que permitem alcançar o maior número de vizinhos não cobertos pela transmissão realizada pela fonte. Nos saltos seguintes, serão escalonados primeiro os nós que permitem alcançar o maior número de nós não cobertos pelo salto anterior, ou seja, o nó de onde a mensagem foi recebida. Entretanto, estes nós não cobertos podem já ter recebido a mensagem proveniente de outro reencaminhador. De qualquer forma, ao se escalonar desta forma, entre todos os vizinhos envolvidos na escolha de *slots*, será escalonado primeiro aquele que estenderá a inundação recebida de forma mais ampla na rede.

Já a Hipótese 3 é justificativa para a possibilidade de dois vizinhos de um nó reencaminhador da inundação serem escalonados simultaneamente, desde que não sejam vizinhos entre si. Ao se aplicar o Algoritmo 3, os nós que são escalonados antes são os que possuem mais vizinhos diferentes em relação ao salto anterior, que equivale a menos vizinhos em comum se considerarmos a rede modelada como um grafo tipo UDG. Desta forma, serão escalonados primeiramente os nós mais distantes da origem, e somente serão escalonados simultaneamente nós que não são vizinhos entre si, que pelas características de grafos UDG também serão distantes entre si, portanto, serão menos interferentes mutuamente.

Por outro lado, um ponto em aberto no algoritmo é a função  $s(x, y)$ , uma função empírica utilizada para computar o deslocamento de *slots* (*next*, no passo 12 do Algoritmo 3) que realiza o escalonamento do reencaminhamento, fazendo-se  $x = lowPri$  e  $y = lowId$ . Em princípio, a única restrição para esta função é que seja crescente e apresente valores inteiros quando  $x$  e  $y$  também são inteiros. Seguindo o conceito de desenvolvimento de um mecanismo simples, a primeira função sugerida e avaliada é  $s(x, y) = x + y + 1$ . Durante a avaliação do mecanismo realizada na Seção 4.2 outras opções para essa função são propostas, formalizadas e avaliadas.

O mecanismo proposto pode ser formalizado através de grafos e conjuntos, considerando a rede sem fio um grafo  $G$ . Definindo como  $o$  o vértice origem da inundação, podemos chamar de  $e_{i,s}$  o  $i$ -ésimo vértice a reencaminhar a mensagem de  $o$  no *slot*  $s$ . Os  $i$  nós que encaminham a mensagem no *slot*  $s$ , o fazem simultaneamente. As transmissões simultâneas são possíveis graças ao reuso espacial (STDMA), e é função do escalonador impedi-las quando os nós são interferentes entre si, e permiti-las para aqueles que não se interferem, conforme Hipótese 3. Definimos como  $V_1(e_{i,s})$  o conjunto dos vizinhos de 1 salto de  $e_{i,s}$ . O receptor  $r \in V_1(e_{i,s})$  será um dos  $j$  emissores do *slot* seguinte,  $e_{j,s+1}$ , se  $\forall v \in (V_1(e_{i,s}) \cap V_1(r))$  temos que  $|(V_1(v) - V_1(e_{i,s}))| < |(V_1(r) - V_1(e_{i,s}))|$ . Além disto, podemos dizer que  $r$  será  $e_{k,s+n}$ , onde  $n = |I| + 1$ , sendo  $I = \{u \in (V_1(r) \cap V_1(e_{i,s})) \text{ tal que } |(V_1(u) - V_1(e_{i,s}))| > |(V_1(r) - V_1(e_{i,s}))|\}$ .

### 3.3 Conclusões

Neste capítulo foram apresentados os detalhes que compõem as duas propostas desta tese. Na Seção 3.1 são apresentados os conceitos do mecanismo de redução da inundação FloorB (*Flooding control through neighborhood Representation by Bloom filters*) da seguinte forma:

- na Seção 3.1.1 são apresentados os conjuntos de vizinhança utilizados e como eles podem ser computados.
- na Seção 3.1.2 são apresentados os conceitos e propriedades dos filtros de

Bloom, bem como estes podem ser utilizados para computar os conjuntos utilizados pelo FloorB, reduzindo o tamanho das mensagens de *hello* e o custo computacional para a obtenção destes conjuntos.

- na Seção 3.1.3 é apresentado o algoritmo do FloorB.
- na Seção 3.1.4 é feita uma análise geométrica das áreas correspondentes aos conjuntos utilizados, e demonstrada que assumem um relação linear de acordo com a intuição inicial utilizada na elaboração da proposta do FloorB.

Em seguida, na Seção 3.2, é apresentada a proposta do mecanismo de escalonamento de inundação 2BSheld (*2-hop aware Broadcast Scheduling with local decision*). Para elaboração deste mecanismo escalonador, inicialmente são avaliadas as latências obtidas por mecanismos de redução da inundação (Seção 3.2.1), em seguida formulamos 3 hipóteses, que servem de base para o mecanismo, descrito na Seção 3.2.2.

Nesta descrição do 2BSheld é apresentada uma primeira função  $s$  responsável pelo cálculo do deslocamento de *slots* utilizado no escalonamento. No próximo capítulo são apresentados os resultados obtidos para os dois mecanismos propostos e em relação ao 2BSheld são propostas e avaliadas outras opções para a função de escalonamento  $s$ .

# Capítulo 4

## Resultados

Este capítulo apresenta as metodologias utilizadas e os resultados obtidos nas avaliações das duas propostas desta tese.

### 4.1 FloorB

Esta seção enumera as premissas usadas nas implementações no simulador próprio e no ns-2 [48], descreve os cenários utilizados e avalia os resultados obtidos, para o mecanismo de redução de inundação FloorB.

#### 4.1.1 Implementações

Para avaliar os mecanismos de redução de inundação, foi implementado um simulador em linguagem *tcl* [49], sem a adoção de uma camada de enlace específica, não existindo portanto contenção ou colisão. Neste simulador, cada transmissão é atômica e instantânea, chegando a todos os vizinhos determinados pelo alcance de transmissão. Na recepção, é possível introduzir uma probabilidade de perda de mensagens, tanto de inundação quanto de *hellos*.

A decisão de realizar uma implementação inicial em *tcl* foi motivada por alguns fatores. Inicialmente, esta é uma linguagem de sintaxe simples, interpretada, e com diversas funções para manipulação de listas, permitindo solucionar rapidamente problemas de tratamento de grafos, como manipulação de matrizes de adjacências, listas de vizinhança, e representações de conjuntos, bem mais complicadas de se fazer na linguagem C, por exemplo. Além disto, a possibilidade de abstrair questões de camadas inferiores, tais como modelos de propagação, cálculos de alcances de transmissão, métodos de acesso ao meio, colisões e capturas, permitiriam um entendimento mais preciso do funcionamento dos mecanismos em ambientes ideais, ou seja, sem perdas e interferências. Em relação a utilizar o simulador ns-2, o simulador próprio apresenta ainda vantagem importante quando aplicado a mecanismos mais

teóricos, que utilizam o conhecimento preciso de informações globais que não estão imediatamente disponíveis aos nós no ns-2.

O simulador é modular, permitindo o acréscimo de diversos mecanismos de inundação. Ele consiste basicamente de uma rotina principal que recebe os diversos parâmetros de simulação, incluindo o mecanismo a ser executado, e possui módulos básicos que executam as tarefas gerais de inundação ou controle, tais como calcular a vizinhança em função do alcance selecionado, simular o envio e recebimento da inundação e simular o envio e recebimento de mensagens de *hello*. Outra vantagem do simulador é que as mensagens não são efetivamente enviadas, mas somente alteradas as listas correspondentes a elas nos nós. Por exemplo, quando é executado o procedimento de envio de mensagem, simplesmente é computada a lista de vizinhos do emissor, e preenchida a lista de mensagens recebidas de cada um dos nós obtidos anteriormente. No caso de introdução de perdas, este preenchimento de listas ocorrerá de acordo com a probabilidade selecionada. Desta forma, os seguintes mecanismos foram implementados:

- Blind - cada nó reencaminha a mensagem na primeira vez que a recebe. A implementação corresponde a uma busca em largura, e se a rede for conexa, e não houver perdas, todos os nós receberão a inundação.
- Gossip - cada nó reencaminha a mensagem na primeira vez que a recebe, com probabilidade  $p$ .
- Gossip Adaptativo - cada nó reencaminha a mensagem na primeira vez que a recebe com probabilidade  $p$ , conforme proposto no mecanismo RAPID [11], descrito na Seção 2, mas sem utilizar mecanismo de temporização.
- $d/R_t$  - cada nó reencaminha a mensagem na primeira vez que a recebe com probabilidade  $p = \frac{d}{R_t}$ , a partir do conhecimento exato da distância entre os nós e do alcance de transmissão.
- $S(M)/S(I)$  - cada nó reencaminha a mensagem na primeira vez que a recebe com probabilidade  $p = \frac{S(M)}{S(I)}$ , conforme Equações 3.9 e 3.11, a partir do conhecimento exato da distância entre os nós e do alcance de transmissão.
- Stauffer-Valmir - implementação da heurística proposta em [26]. Este mecanismo baseia-se em uma redução de inundação por arestas e são apresentados somente resultados considerando os modos de economia por nós.
- MPR ideal - cada nó executa o algoritmo de escolha de MPRs no instante que envia (ou reencaminha) a mensagem e o cálculo de MPRs é baseado no conhecimento preciso e imediato do posicionamento dos nós da rede.

- MPR real - cada nó executa o algoritmo de escolha de MPRs no instante que envia (ou reencaminha) a mensagem, mas o cálculo de MPRs é realizado baseado no conhecimento de vizinhança obtido pelo recebimento de *hellos*.
- FloorB - implementação da proposta descrita na Seção 3.1.

Alguns parâmetros dos mecanismos foram previamente estudados, avaliados e escolhidos. Para o Gossip adotou-se  $p = 0,7$ . Este valor é semelhante aos principais valores utilizados na proposta original do Gossip [9], de 0,60, 0,65 e 0,72. Além disto, é importante observar o comportamento do mecanismo, que apresenta economia constante e uma taxa de entrega reduzida em redes esparsas. Outros valores de  $p$  alterariam os resultados obtidos, sem alterar, no entanto, a tendência do mecanismo.

Já para o Gossip Adaptativo, foi selecionado o valor de  $\beta = 6$ , diferente do proposto em [11], onde  $\beta = 2,5$ . Esta escolha foi realizada a partir de simulações e de uma avaliação sobre o modelo apresentado na proposta do RAPID. Devido à equação utilizada para cálculo da probabilidade de reencaminhamento, fazendo-a inversamente proporcional ao número de vizinhos, considerando-se uma rede com distribuição uniforme dos nós na área, ou seja, os valores de  $V_1(x)$  sendo aproximadamente iguais para qualquer nó  $x$ , então  $p_{not}$ , a probabilidade de que nenhum dos nós pertencentes a  $V_1(e)$  reencaminhe a mensagem, será dada por:

$$p_{not} = \left(1 - \frac{\beta}{|V_1(r)|}\right)^{|V_1(r)|} \quad (4.1)$$

A Equação 4.1 possui limite igual a  $e^{-\beta}$  quando o número de vizinhos tende a infinito, fazendo que, com probabilidade igual a  $e^{-\beta}$ , a mensagem já deixe de ser encaminhada logo no primeiro salto, restringindo assim o desempenho deste mecanismo para redes muito densas. É importante notar que esta Equação 4.1 possui uma convergência muito rápida para os limites. Por exemplo, para os valores de  $\beta$  citados, 6 e 2,5, os limites são respectivamente 0,002479, ou uma probabilidade de 0,25% da mensagem não ultrapassar o primeiro salto, e 0,082085, ou uma probabilidade de 8,2% de nenhum dos vizinhos reencaminhá-la. Entretanto, com apenas 20 vizinhos o valor de  $p_{not}$  da Equação 4.1, com  $\beta = 6$ , é aproximadamente de 0,08%, enquanto que com  $\beta = 2,5$  já é de 6,9%.

Para a avaliação do mecanismo Stauffer-Valmir, foi utilizado valor de  $\alpha = 0,99$ , ou seja, uma taxa de entrega desejada de 99% (Equação 2.1).

Os parâmetros do filtro de Bloom do mecanismo FloorB também foram selecionados através de simulações iniciais, e pelo estudo da Figura 3.5. Foram utilizados filtros com 256 bits ( $m = 256$ ) e 2 funções *hash* ( $k = 2$ ), por apresentarem baixas probabilidades de falso positivos para nós com até 50 vizinhos, e também não

representar um tamanho muito elevado para ser colocado nas mensagens de *hello*.

Para avaliar os mecanismos foram utilizados 3 tipos de cenários, enumerados a seguir:

- Grade - nós dispostos em grade, com distância fixa, sem mobilidade, e vizinhança estabelecida pelo alcance de transmissão.
- Aleatório móvel - cenário gerado através da ferramenta **setdest** do ns-2, com nós distribuídos aleatoriamente numa área e com movimentação do tipo *Random Waypoint* (RWP). Foram utilizados os seguintes parâmetros: área de  $800m \times 800m$ , 200 ou 300 nós, alcances de 60, 80, 100 e  $120m$ , conforme indicado, pausa de  $10s$ , e velocidade mínima de  $0m/s$  e máxima conforme indicado a cada gráfico.
- Mobilidade social - cenário gerado através da ferramenta proposta em [21], onde os nós se agrupam em comunidades e se deslocam dentro do seu grupo, seguindo alguma forma de movimentação baseada em modelo social. Foram criados cenários com 100 nós, alcance de  $80m$  e dois grupos em uma área de  $400m \times 400m$ . Estes cenários representam redes semelhantes a apresentada na Figura 3.3, onde dentro do grupo há uma alta densidade, mas a comunicação entre os grupos ocorre somente através de alguns poucos nós na fronteira entre ambos.

Em seguida, os mesmos mecanismos, exceto aqueles de características mais teóricas, tais como MPR ideal,  $d/R_t$ ,  $S(M)/S(I)$ , foram implementados no simulador ns-2, permitindo realizar uma avaliação dos mecanismos quando aplicados sobre uma camada de enlace com características reais, no caso IEEE 802.11. Também não foi implementada a proposta Stauffer-Valmir, pelas questões de comparabilidade já citadas, referentes ao fato desta proposta tratar as transmissões enlace a enlace, de forma equivalente a comunicação *unicast*, e não como o *broadcast* de redes sem fio.

Em relação ao mecanismo MPR, há uma diferença importante em relação às duas implementações. No simulador próprio, o processo de seleção dos MPRs é definido no momento de envio da mensagem de inundação, enquanto a implementação realizada no ns-2 é mais próxima daquela utilizada no OLSR, onde a mensagem de *hello* informa o conjunto de MPRs selecionados.

Na implementação no ns-2, os mecanismos de inundação fazem parte de um agente, que se liga à camada de enlace do nó de forma semelhante a um protocolo de roteamento, mas adotando outra porta de ligação, permitindo o uso em conjunto com os protocolos de roteamento já existentes no ns-2.

As duas medidas de desempenho utilizadas para comparação dos mecanismos foram a taxa de entrega, ou alcançabilidade, expressa por:

$$\text{tx entrega} = \frac{\# \text{ msg recebidas}}{\# \text{ nós na rede} \times \# \text{ msgs enviadas}} \quad (4.2)$$

e a taxa de reencaminhamentos evitados, expressa por:

$$\text{reenc evitados} = 1 - \frac{\# \text{ msg reencaminhadas}}{\# \text{ msg recebidas}} \quad (4.3)$$

Ambas as métricas possuem valores entre 0 e 1, e maiores valores indicam melhores desempenhos. Entretanto, em relação aos reencaminhamentos evitados, uma taxa igual a 1 significa que nenhuma mensagem recebida será reencaminhada, o que levará então que a mensagem fique restrita ao primeiro salto na rede.

Portanto, estas métricas precisam ser sempre observadas em conjunto, uma vez alguns mecanismos podem obter excelentes resultados segundo uma das métricas ao custo de um desempenho ruim na outra.

Além destas medidas, no cenário em grade e com simulador próprio, também foi avaliada a distribuição do número de nós com as diversas taxas de reencaminhamentos realizados, de 0 a 1, em intervalos de 0,1. O objetivo desta medida é obter a distribuição da economia do mecanismo nos diversos nós, sendo melhor o mecanismo que promove uma distribuição mais igualitária, fazendo com que a maior parte dos nós possua a mesma taxa de reencaminhamento.

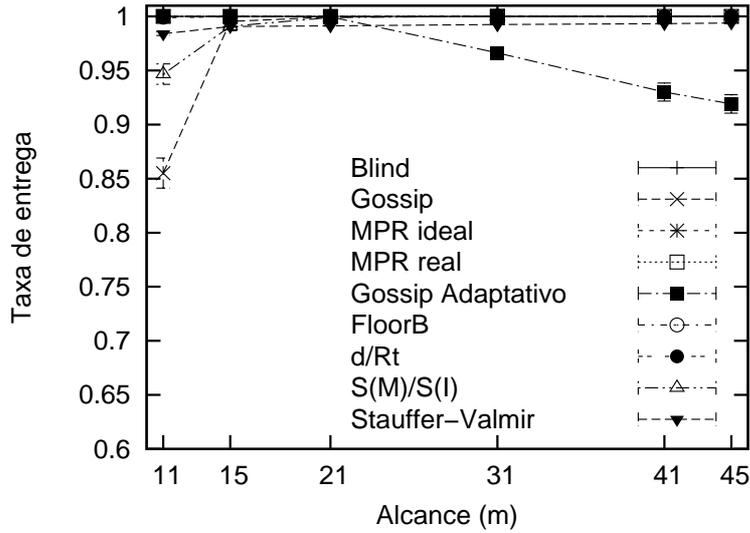
Este fator é importante por duas considerações: mecanismo que sobrecarregue alguns nós com muitos reencaminhamentos estará sujeito a problemas de desempenho no caso de falha deste nó; ao concentrar os reencaminhamentos em alguns nós, caso exista limitação quanto à energia, comum em redes *ad hoc*, mais rapidamente a energia de tal nó se esgotará, levando à falha do mesmo.

### 4.1.2 Simulador próprio

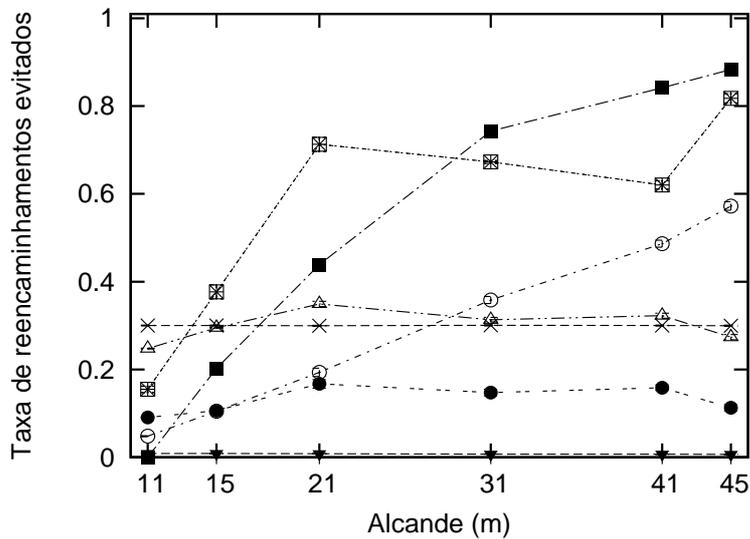
Alguns parâmetros também foram modificados nas diversas avaliações, em particular o alcance dos nós, para alterar o número de vizinhos e a densidade das redes, a velocidade dos nós, para diversos graus de mobilidade, e a taxa de perda na recepção, para simular o comportamento dos mecanismos frente a perdas decorrentes das camadas de enlace (colisão) e física. Em todas as avaliações, 10 fontes iniciam inundações periódicas, realizadas a cada 20s, após um tempo de transiente e aprendizado para os mecanismos de 45s e até o final da simulação em 500s. São calculadas médias e intervalos de confiança de 95%.

Ainda para avaliar a mobilidade, foram realizadas simulações onde foi variado o intervalo entre os envios das mensagens de *hello*. Esta é uma forma indireta de avaliar a robustez dos mecanismos frente à mobilidade.

O primeiro cenário avaliado foi o de rede em grade. Ele é composto de 400 nós



(a) Taxa de Entrega.



(b) Reencaminhamentos Evitados.

Figura 4.1: Cenário em grade, sem perdas

em uma grade de  $20 \times 20$ , equidistantes de  $10m$  cada, sem mobilidade e sem perdas. O alcance de transmissão utilizado foi variado de  $11$  até  $41m$ , possibilitando avaliar a escalabilidade dos mecanismos com o aumento da vizinhança, que variou de  $4$  ( $11m$ ) até  $72$  vizinhos ( $41m$ ). Os resultados obtidos quanto a taxa de entrega podem ser vistos na Figura 4.1(a).

Este é um cenário estático e com distribuição uniforme dos nós, portanto, teoricamente facilmente tratável por todos os mecanismos. Entretanto, pode-se observar que o Gossip apresenta queda na taxa de entrega quando a rede é esparsa, confirmando resultado já esperado e descrito em outros trabalhos [18]. A implementação  $S(M)/S(I)$  apresentou o mesmo problema, mas de forma menos acentuada.

Já o Gossip Adaptativo apresenta o resultado esperado de redução da taxa de

entrega quando a rede se torna densa, conforme apresentado nas explicações sobre a Equação 4.1.

Os mecanismos Blind, MPR ideal e MPR real apresentam taxa de entrega de 100% com todos os alcances, uma vez que a rede é conectada e não há perdas e nem mobilidade. Já o FloorB apresentou taxa de entrega superior a 99,9% em todos os alcances, e o Stauffer-Valmir apresentou taxa de entrega de 99%, conforme configurado.

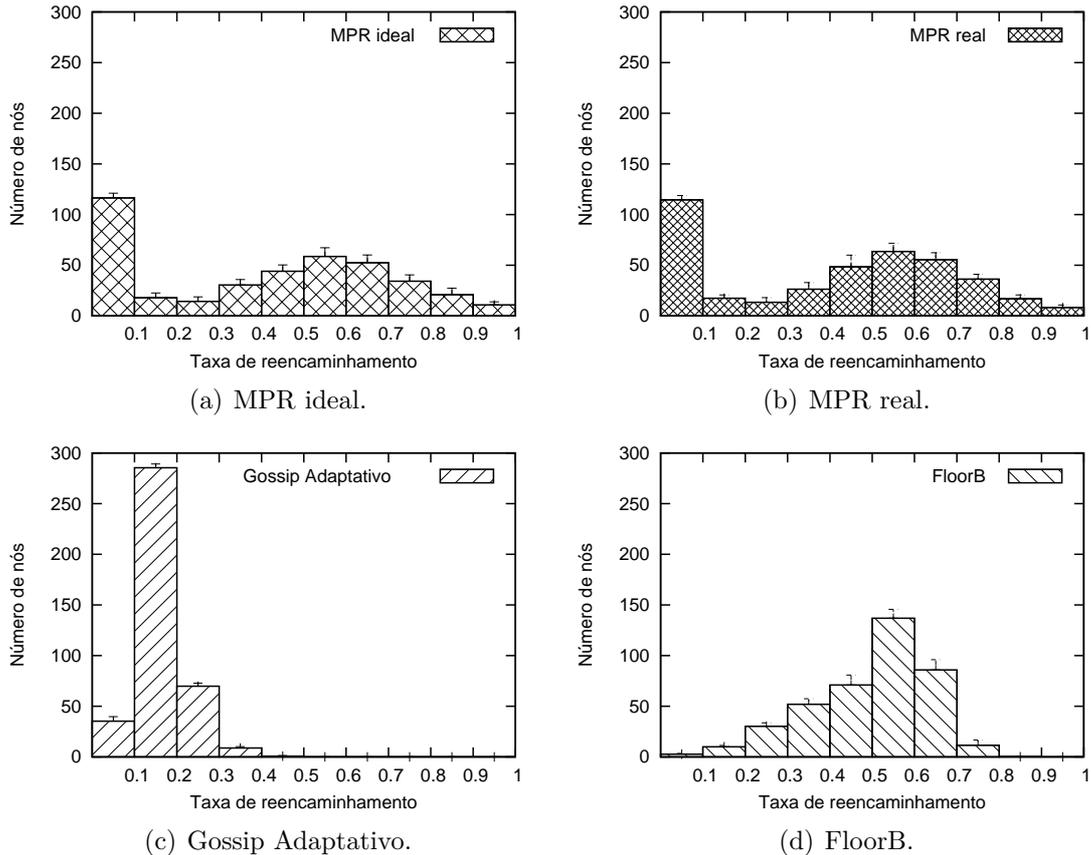


Figura 4.2: Distribuição da tarefa de reencaminhamento.

Na Figura 4.1(b), são mostrados os reencaminhamentos evitados para cada mecanismo. Pode-se notar que para esta medida o Blind e o Gossip têm desempenho constante, equivalente a  $1 - p$ , a probabilidade de não reencaminhar, que no Blind é igual a 0. Os mecanismos  $d/R_t$  e  $S(M)/S(I)$  não apresentam grande economia, tendo resultados abaixo e semelhante ao Gossip, respectivamente. Os mecanismos de MPR, tanto ideal quanto real, apresentaram desempenho superior ao FloorB e ao Gossip Adaptativo, exceto com alcance de  $41m$ , onde o Gossip Adaptativo reencaminhou menos, porém ao custo de uma redução considerável na taxa de entrega. Ainda neste gráfico podemos notar que o mecanismo FloorB é conservador, e não apresenta o mesmo desempenho que o MPR no número de reencaminhamentos evitados, a não ser nas redes muito densas. Entretanto, é justamente nestas redes que o

problema de tempestade de *broadcast* se torna mais grave, ou seja, que o mecanismo deve ser mais atuante.

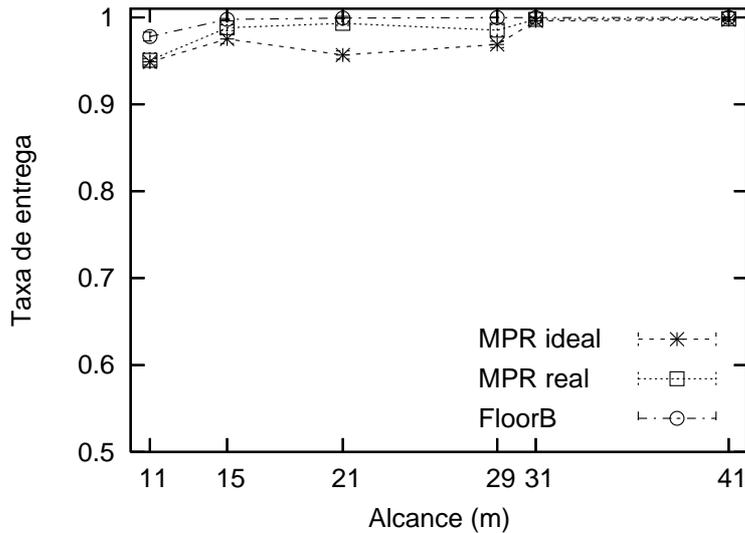
A proposta Stauffer-Valmir apresentou economia quase nula, pois a implementação deste mecanismo faz a decisão de reencaminhamento da mensagem ocorrer enlace a enlace, e o cálculo de reencaminhamentos evitados indica a proporção de nós que recebem a mensagem e não a reencaminham nenhuma vez. Desta forma, podemos concluir que para este tipo de cenário, neste mecanismo quase todos os nós acabam reencaminhando a mensagem de inundação pelo menos uma vez.

Além disto, na Figura 4.2 observa-se a distribuição da quantidade de nós em função das taxas de reencaminhamentos realizados, métrica complementar aos reencaminhamentos evitados apresentado na Figura 4.1(b), para os mecanismos Gossip Adaptativo, MPR ideal, MPR real e FloorB, quando o alcance de transmissão é de  $41m$ . Neste gráfico, pode-se notar que apesar dos mecanismos MPR terem uma boa taxa de reencaminhamentos evitados, isto ocorre à custa de que alguns nós, da ordem de 30, façam mais de 80% de reencaminhamentos das mensagens de inundação. A média de reencaminhamentos do mecanismo MPR torna-se reduzida porque um grande número de nós, da ordem de 120 neste cenário, reencaminham menos de 10% das mensagens de inundação recebidas.

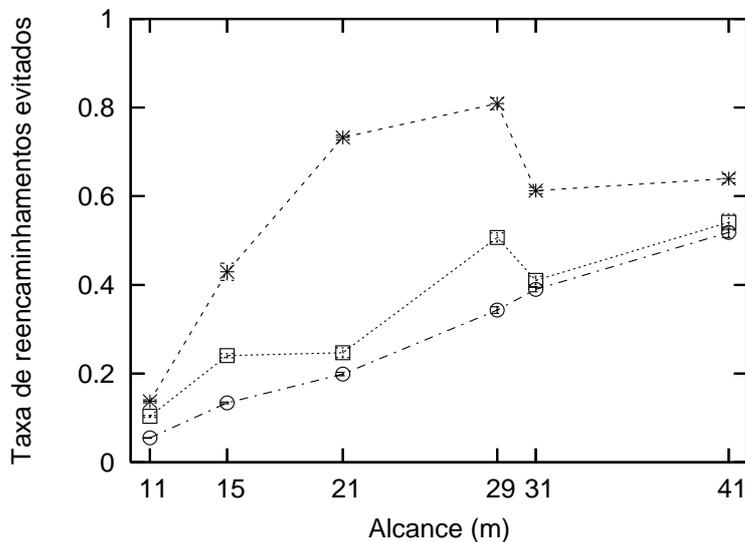
Já os mecanismos Gossip Adaptativo e FloorB apresentam uma distribuição mais equilibrada, não tendo nós com taxas de reencaminhamento superiores a 40% e 80%, respectivamente. Este resultado indica que nestes mecanismos não haverá nós sobrecarregados com a tarefa de reencaminhar quase todas as inundações, enquanto outros permanecem ociosos, sendo que o melhor desempenho foi obtido pelo Gossip Adaptativo, onde quase  $3/4$  dos nós realizaram a mesma tarefa de reencaminhamento, e os demais não tiveram atuação muito diferente. Uma grande vantagem deste comportamento, conforme já citado, é a sobrevida da rede, uma vez que os nós com taxas de reencaminhamento muito alta podem esgotar rapidamente sua energia.

Outro fator importante ao se concentrar o tráfego em poucos nós é que quando perdas ou falhas nas transmissões ocorrem, elas se tornam mais graves, afetando a taxa de entrega, como observado na Figura 4.3, onde é utilizada taxa de perda na recepção de 0,2. Nota-se que no MPR ideal, onde as perdas não influenciam na escolha dos MPRs, há uma redução na taxa de entrega nas redes mais esparsas. Já no MPR real, como as perdas influenciam no aprendizado da vizinhança, a taxa de entrega pouco se altera, mas há uma redução nos reencaminhamentos evitados. O FloorB, neste cenário, apresenta-se mais estável, com melhores taxas de entrega em redes esparsas e taxas de reencaminhamentos evitados em redes densas equivalentes ao MPR real.

Neste gráfico, conforme informado, são apresentadas barras de erro correspon-



(a) Taxa de Entrega.



(b) Reencaminhamentos Evitados.

Figura 4.3: Cenário em grade, perdas=0,2.

dentos a intervalos de confiança de 95%. Entretanto, como há pouca variabilidade entre rodadas, pois os cenários são estáticos, estas barras são muito reduzidas. Por outro lado, neste gráfico, onde foram utilizados alcances de 29 e 31m, fica evidente uma descontinuidade de desempenho em termos de reencaminhamentos evitados, principalmente nos mecanismos MPR. A justificativa para este efeito é o próprio cenário, que apresenta uma descontinuidade em relação a vizinhança. No caso do alcance de 29m, a vizinhança inclui todo o quadrado de duas linhas e duas colunas no entorno do nó. Ao passar o alcance para 31m, são adicionados somente mais dois nós existentes na mesma linha e na mesma coluna, sem acréscimo na diagonal. Isto provoca o seguinte efeito, no primeiro caso, 29m, basta selecionar os nós situados nos cantos do quadrado definido pelo alcance como MPRs, ou seja, bastam 4

MPRs. Quando o alcance passa para  $31m$ , os nós situados nos cantos diagonais não permitem mais alcançar todos os vizinhos de 2 saltos, sendo necessário selecionar também os nós extremos na mesma linha e coluna, totalizando 8 MPRs. A relação não se torna então metade da economia por conta do efeito de borda, que se torna maior com o aumento do alcance.

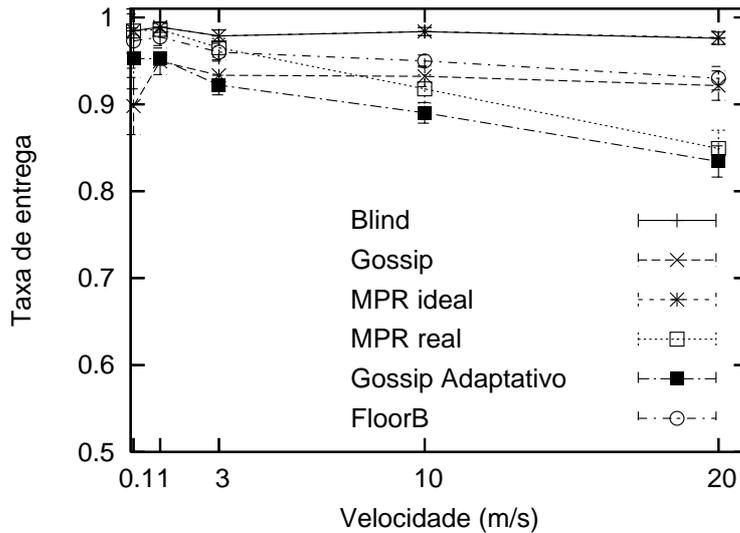
O segundo tipo de cenário avaliado foi o cenário com posicionamento e mobilidade aleatórios do tipo *Random Waypoint*. Para este tipo de cenário, variou-se a velocidade máxima dos nós para se observar o comportamento dos mecanismos de acordo com o grau de mobilidade da rede. Os resultados obtidos podem ser vistos na Figura 4.4 onde observa-se que as taxas de entrega se reduzem para o Gossip Adaptativo e para o MPR real, com o aumento da mobilidade. Isto se deve principalmente ao fato de que o aprendizado de vizinhança responde mais rapidamente na inclusão do que na exclusão de nós.

Quando um novo vizinho entra na área de alcance, basta o recebimento de um *hello* para que o mesmo passe a fazer parte da lista de vizinhos. Entretanto, quando um nó sai do alcance, somente após a perda de um número consecutivo de *hellos* ele sairá da lista. Nestas simulações, foram utilizados intervalos de *hellos* de  $3s$ , e uma perda consecutiva de 3 *hellos* para remoção da lista.

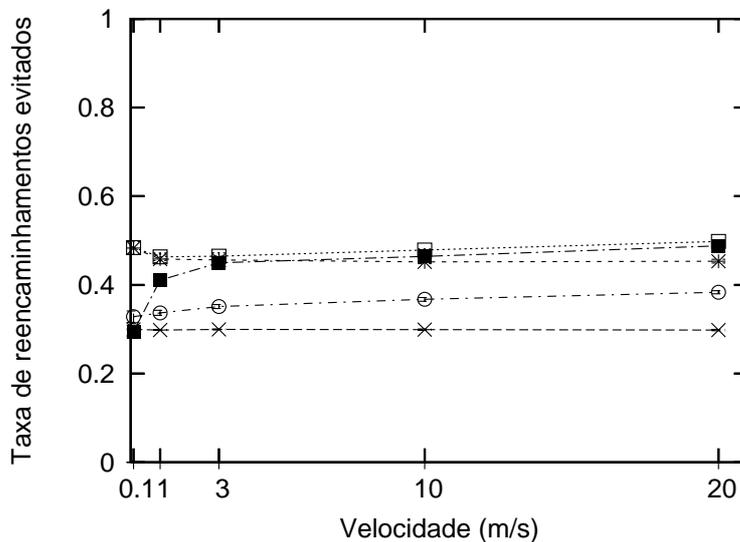
Uma solução possível para que os mecanismos respondam melhor à mobilidade é a redução do intervalo entre *hellos*. Entretanto, isto aumenta o número de mensagens de controle, e pode inclusive provocar o aumento da taxa de perdas, em função do crescimento de colisões. Por outro lado, conforme já citado, o mecanismo FloorB se apresenta mais estável com aumento da mobilidade, uma vez que utiliza uma razão entre o número de elementos em dois conjuntos, e ambos ficam submetidos a erros na contagem de elementos na mesma proporção, praticamente anulando este efeito.

Uma segunda solução seria fazer o processo de entrada e saída na vizinhança ocorrer da mesma forma. Entretanto, fazer a entrada na vizinhança ocorrer somente após o recebimento consecutivo de 3 mensagens de *hello* pode tornar o aprendizado de novos vizinhos muito lento. Adotando a solução contrária, fazendo o nó ser retirado da vizinhança no primeiro *hello* perdido, pode provocar o efeito de qualquer perda de mensagem, devido à colisão ou interferência, gerar mudança na vizinhança, inclusive com redução da taxa de entrega.

Para comprovar esta hipótese de estabilidade com a mobilidade, ainda com cenário aleatório, foram realizadas outras simulações, neste caso com grau de mobilidade constante, de velocidade de  $0m/s$  até  $5m/s$ , e foi variado o intervalo entre *hellos*, desde  $2s$  até  $15s$ . Estes resultados podem ser vistos na Figura 4.5. Observa-se que mesmo com intervalos de  $15s$  o FloorB mantém a taxa de entrega estável. O desempenho do MPR Real com intervalo de  $5s$  nos *hellos* é equivalente ao do FloorB com intervalos de  $10s$  ou  $15s$ , indicando que a proposta tanto é mais robusta a variações



(a) Taxa de Entrega.



(b) Reencaminhamentos Evitados.

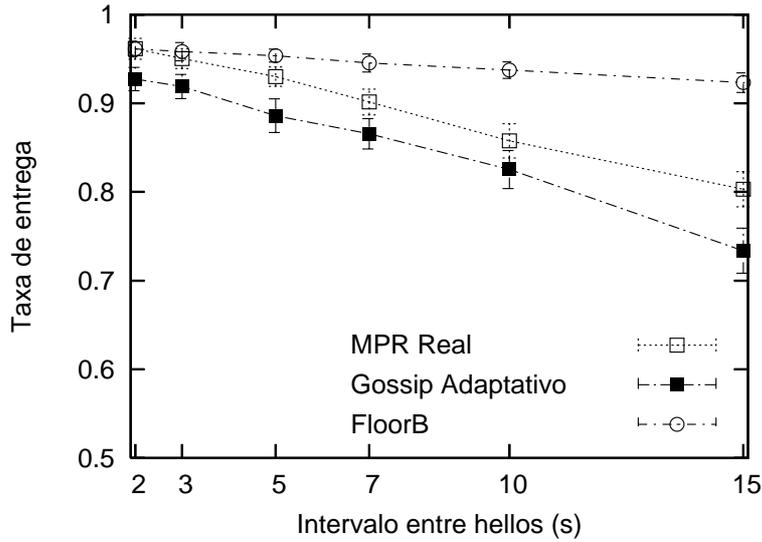
Figura 4.4: Cenário com mobilidade RWP, sem perdas.

de velocidade, como pode ter seu número de mensagens de controle reduzido ao se adotar intervalos maiores no envio dos *hellos*.

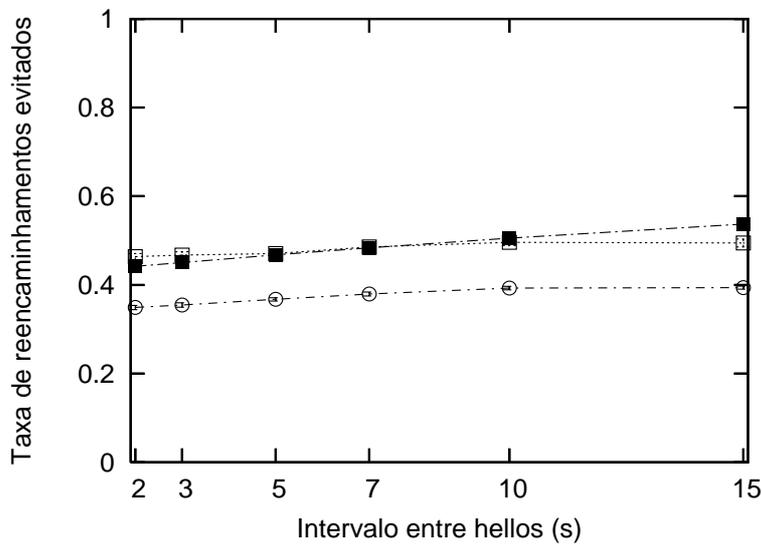
Já quanto aos reencaminhamentos evitados, nota-se nas Figuras 4.4(b) e 4.5(b) que os desempenhos de todos mecanismos são bem semelhantes, com o FloorB apresentando resultado um pouco inferior ao do MPR. Também nota-se que todos os mecanismos não apresentam alteração significativa nesta métrica com o aumento da mobilidade ou do intervalo entre *hellos*.

O terceiro tipo de cenário avaliado consiste de cenário do tipo mobilidade social, onde os nós se movimentam em grupo de acordo com uma predisposição para agrupamento, conforme proposto em [21].

Nestes cenários, existe uma variação muito grande do número de vizinhos dos nós,



(a) Taxa de Entrega.

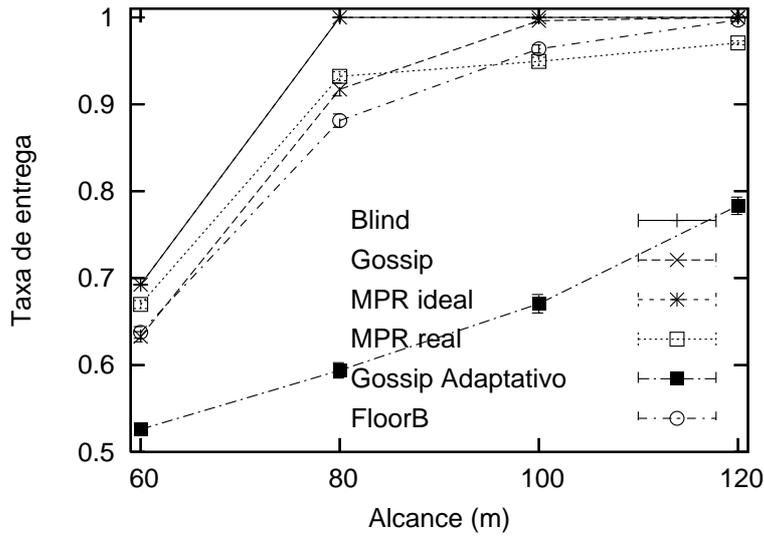


(b) Reencaminhamentos Evitados.

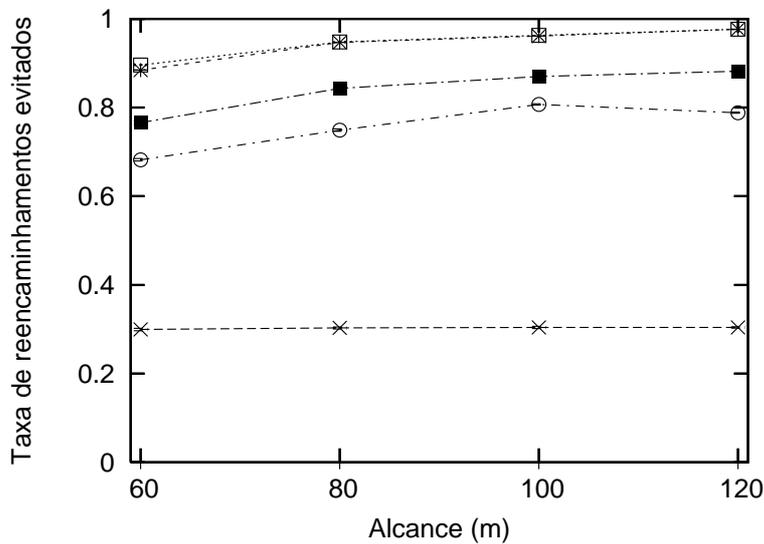
Figura 4.5: Cenário com mobilidade RWP, variação do intervalo entre *hellos*.

de forma semelhante ao apresentado na Figura 3.3, e por este motivo os mecanismos tipo Gossip Adaptativo não obtêm altas taxas de entrega. Conforme a Figura 4.6, como os cenários foram criados considerando um alcance de transmissão de  $80m$ , quando se utiliza um alcance de  $60m$  a rede se torna pouco conectada, e todos os mecanismos apresentam uma taxa de entrega reduzida, mesmo quando não há perdas. É possível observar que as taxas de entrega dos mecanismos MPRs ideal e real e FloorB são semelhantes e próximas do Blind, e novamente o FloorB apresenta-se conservador, com menor número de reencaminhamentos evitados, apesar do alto desempenho de todos.

Entretanto, da mesma forma que no cenário em grade, quando há uma taxa de perdas de 0,2, como observado na Figura 4.7, o desempenho do FloorB é superior



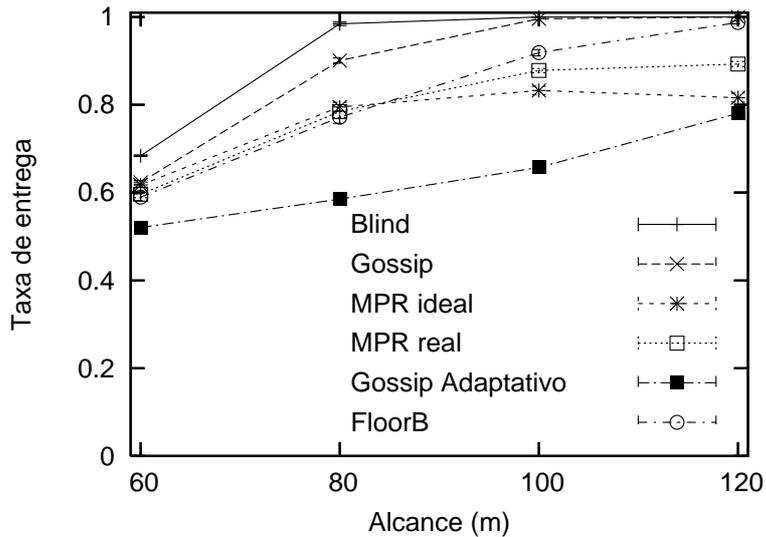
(a) Taxa de Entrega.



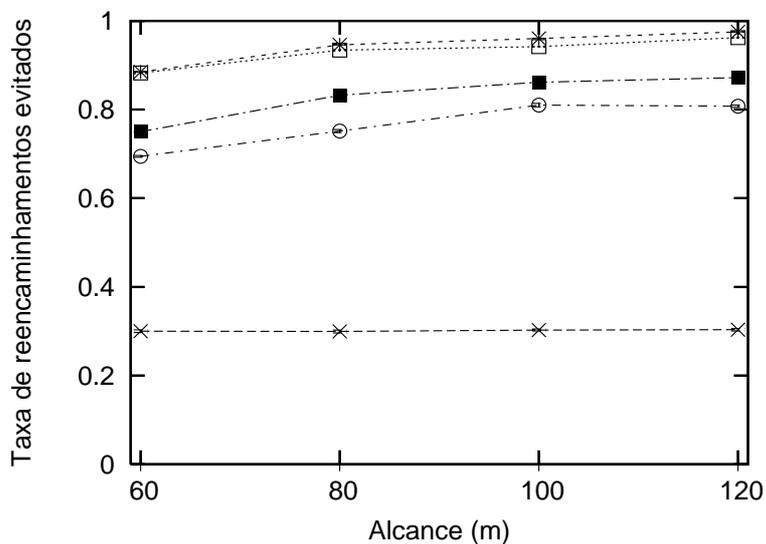
(b) Reencaminhamentos Evitados.

Figura 4.6: Cenário com mobilidade Social (Comunidade), sem perdas.

em termos de taxa de entrega, aproximando-se do Blind, e mantém a taxa de reencaminhamentos evitados. Observa-se também que a introdução de perdas afeta principalmente o mecanismo MPR ideal, que passa a ter desempenho inferior ao MPR real. Isto se deve a escolha mais restrita dos MPRs realizada pelo MPR ideal, que não depende das mensagens de *hello*. Neste caso, ao se ter, por exemplo, um único MPR, de dois ou mais possíveis nós, realizando a ligação dos dois grupos representados pela rede social, as falhas de recepção deste nó representam um impacto maior nas taxas de entrega.



(a) Taxa de Entrega.



(b) Reencaminhamentos Evitados.

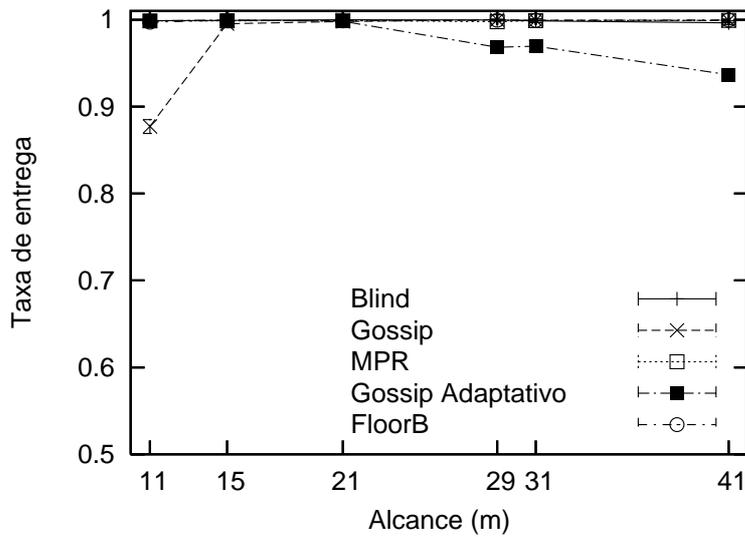
Figura 4.7: Cenário com mobilidade Social (Comunidade), perda=0,2.

### 4.1.3 Simulação no ns-2

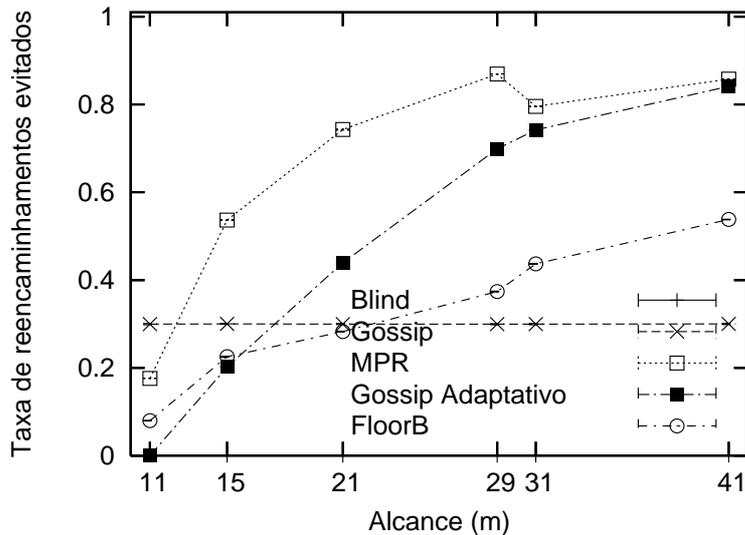
Como continuidade da avaliação, os mecanismos também foram implementados no ns-2 e avaliados com os mesmos cenários em grade e aleatório móvel, utilizando camada de enlace IEEE 802.11, com taxa de transmissão de 11 Mbps. Nesta avaliação, não foram introduzidas perdas, entretanto, elas ocorrem naturalmente devido a interferências e colisões, que se tornam mais intensas em ambientes com vizinhança elevada [50].

Além disto, as questões temporais, tais como sincronização e contenção no acesso ao meio se tornam importantes, uma vez que as transmissões não ocorrem de forma atômica e instantânea. Isto faz com que o tamanho das mensagens interfira no

desempenho dos mecanismos, bem como o próprio uso de *hello* ou não.



(a) Taxa de Entrega.

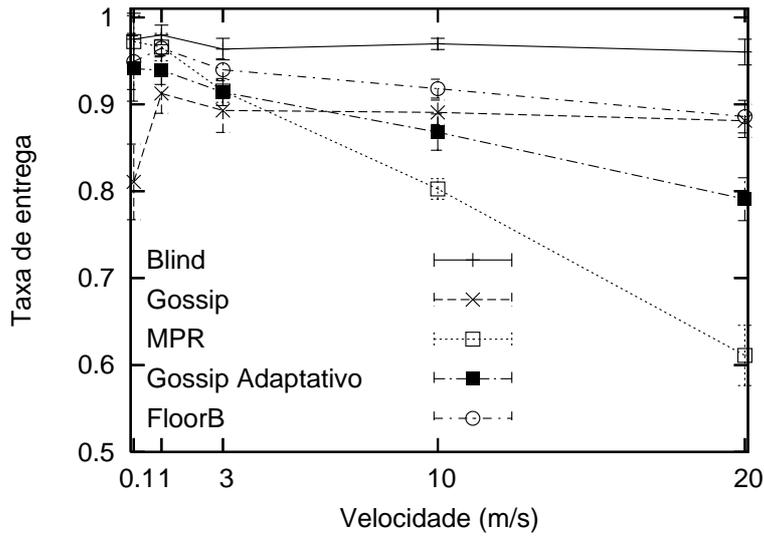


(b) Reencaminhamentos Evitados.

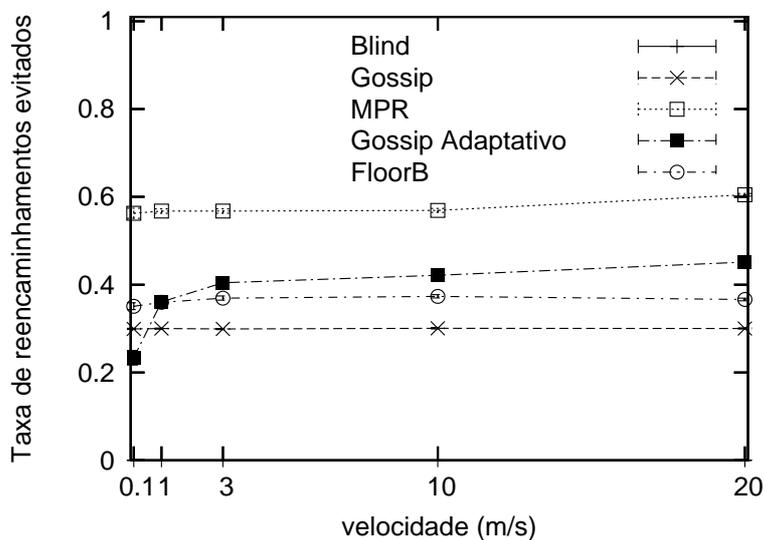
Figura 4.8: Cenário em grade, ns-2.

Com esta implementação no ns-2, pode-se observar na Figura 4.8 o desempenho dos mecanismos no cenário em grades. Nota-se que o resultado é bastante semelhante ao obtido com o simulador próprio, anteriormente apresentado na Figura 4.1, tendo as mesmas tendências. A principal diferença ocorre pelo fato dos mecanismos estarem submetidos a perdas e atrasos nas mensagens.

A Figura 4.9 apresenta o resultado das simulações no ns-2 com cenário móvel aleatório, com 300 nós e alcance de 80m. Observa-se que a taxa de entrega apresentou a mesma tendência obtida com o simulador próprio (Figura 4.4), entretanto, com degradação mais acentuada dos mecanismos MPR e Gossip Adaptativo com o aumento da mobilidade.



(a) Taxa de Entrega.

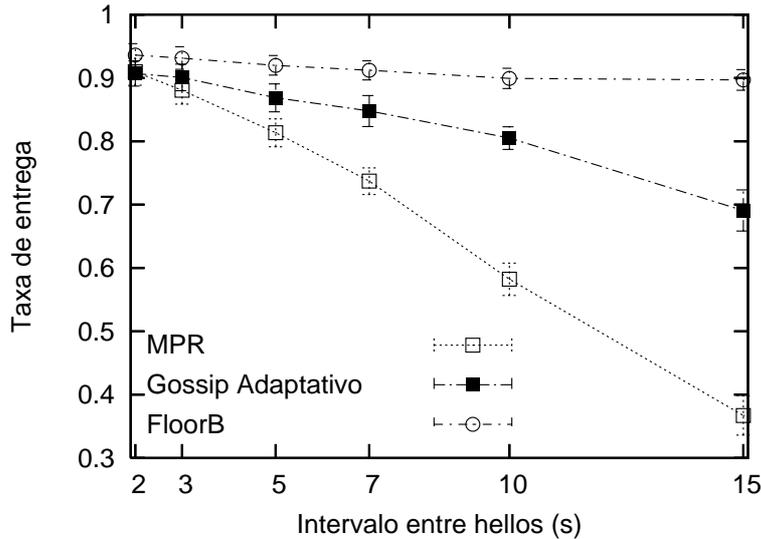


(b) Reencaminhamentos Evitados.

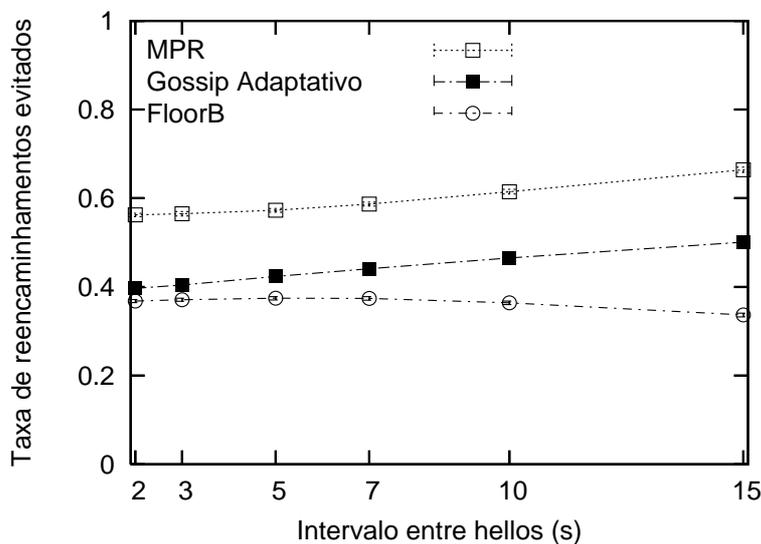
Figura 4.9: Cenário com mobilidade RWP, ns-2.

Pelo mesmo motivo anterior, a avaliação com alteração dos intervalos entre *hellos* foi realizada, e obteve-se novamente a mesma tendência, entretanto, novamente com degradação mais acentuada dos mecanismos MPR e Gossip Adaptativo com o aumento dos intervalos, como se pode observar na Figura 4.10. Nota-se que as taxas de entrega são da ordem de somente 0,4 e 0,7 para os mecanismos MPR e Gossip Adaptativo, respectivamente, quando o intervalo entre *hellos* é de 15s, sensivelmente inferior à taxa de aproximadamente 0,9 obtida pelo FloorB com o mesmo intervalo.

Estes resultados nos fornecem uma validação cruzada das duas implementações, uma vez que ambas apresentam resultados bem semelhantes. É interessante notar que as diversas aproximações e abstrações realizadas no simulador próprio não pro-



(a) Taxa de Entrega.



(b) Reencaminhamentos Evitados.

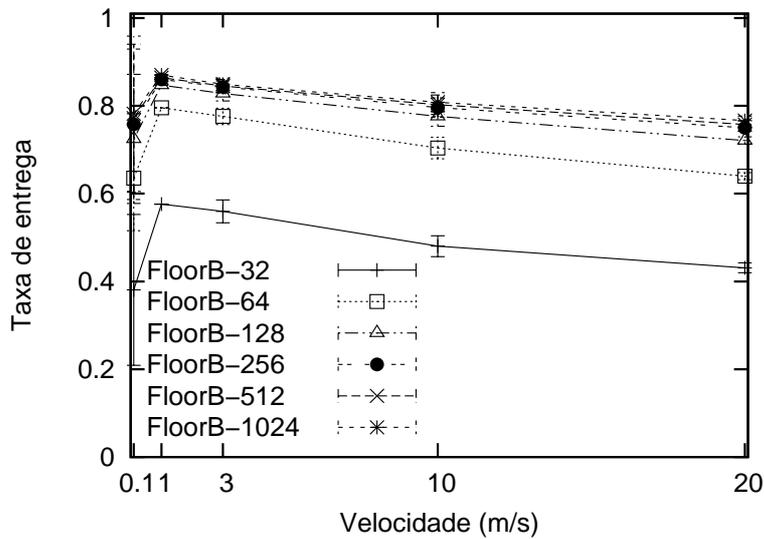
Figura 4.10: Cenário com mobilidade RWP, variação do intervalo entre *hellos*, ns-2.

duzem impacto significativo nos resultados dos mecanismos. Desta forma iniciamos novos tipos de avaliações.

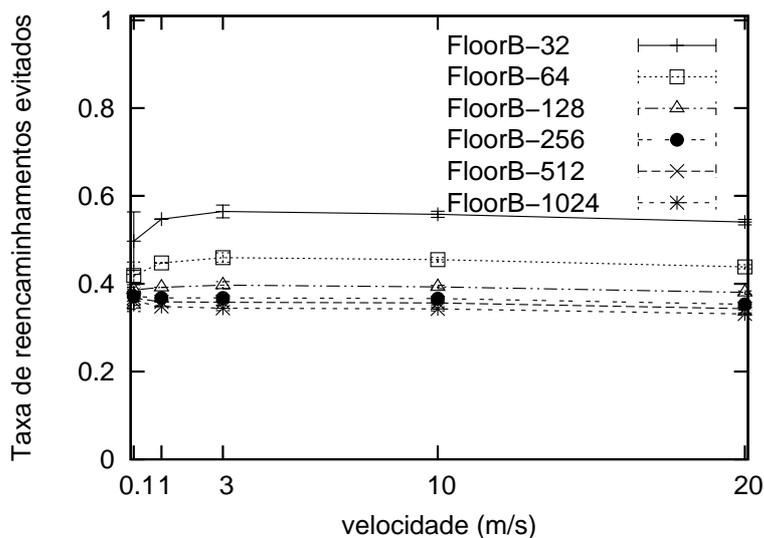
Inicialmente foi estudado o impacto do tamanho do filtro de Bloom no desempenho do FloorB. Foi utilizado um cenário de mobilidade aleatória com 200 nós, com os nós com alcance de transmissão de  $80m$ , e variada a velocidade máxima dos nós. Foram utilizados filtros de 32, 64, 128, 256, 512 e 1024 *bits*, com os resultados de cada um apresentados nas diversas curvas da Figura 4.11.

Observamos que o FloorB com filtro de 32 *bits* apresentou um desempenho inferior. Isto se deve ao tamanho do filtro, insuficiente para representar adequadamente a vizinhança, produzindo uma taxa de falsos positivos exagerada. Com filtros de 64 *bits* ocorreu o mesmo fato, entretanto, de forma menos acentuada. Com filtro de 128

a 1024 *bits* os desempenhos foram muito próximos. Entretanto, redes mais densas podem apresentar problema semelhante se o filtro não possuir tamanho adequado.



(a) Taxa de Entrega.



(b) Reencaminhamentos Evitados.

Figura 4.11: Variação do tamanho do filtro de Bloom.

Ainda com a implementação no ns-2, foi realizada uma nova avaliação, que não era possível ser realizada com o simulador próprio, uma vez que este realiza transmissões instantâneas o que não permite avaliar situações de carga na rede. Para isto, aos cenários aleatórios móveis foram adicionados 16 pares de comunicação do tipo fonte-destino, UDP/CBR, dispostos uniformemente na rede conforme a Figura 4.12. Estes 32 nós não participam das tarefas de inundação, não emitindo, recebendo ou reencaminhando mensagens de controle, só atuando na geração de carga. Da forma como foram dispostos, permitem provocar contenção em toda a área da rede, portanto, concorrendo com todos os *hellos* e mensagens de inundação enviados ou

reencaminhados.

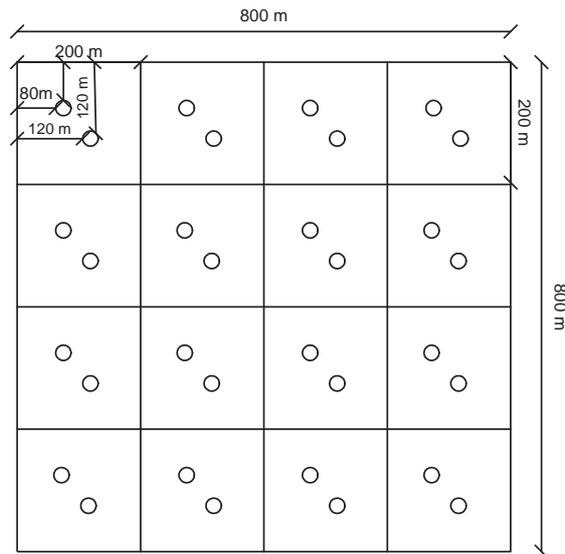


Figura 4.12: Disposição dos nós geradores de carga.

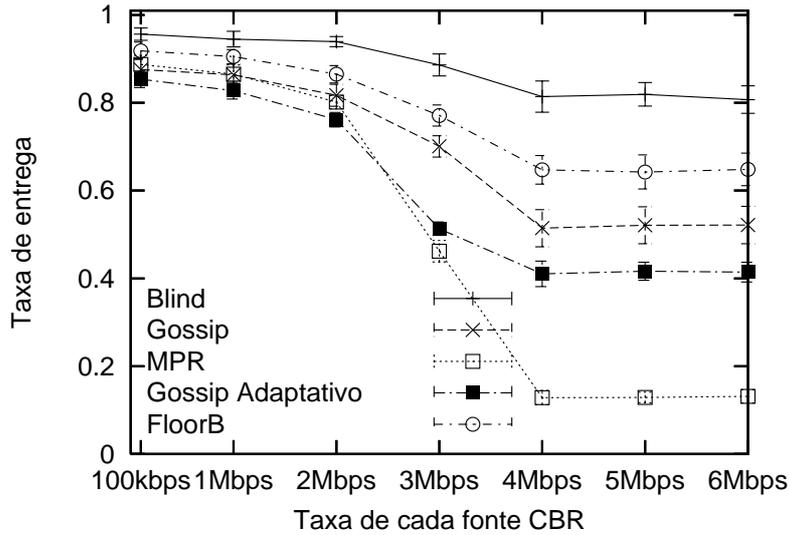
O tráfego CBR gerado inicia-se após 20s de simulação, permitindo a troca inicial de mensagens nos mecanismos, e possui pacote de 1000 *bytes*. A taxa gerada foi variada conforme apresentado na Figura 4.13, desde um valor inicial de 100kbps até 6Mbps correspondentes à saturação da rede.

Pode-se observar pela Figura 4.13(a) que em relação ao comportamento da taxa de entrega dos mecanismos, quando submetidos a carga, o desempenho do Blind foi superior, seguido do FloorB, Gossip, Gossip Adaptativo e MPR, nesta ordem. Em relação aos reencaminhamentos evitados, todos obtiveram desempenho semelhante, exceto o MPR que se apresentou ligeiramente superior aumentando a economia obtida quando há aumento da carga, entretanto, ao custo de reduzir a taxa de entrega.

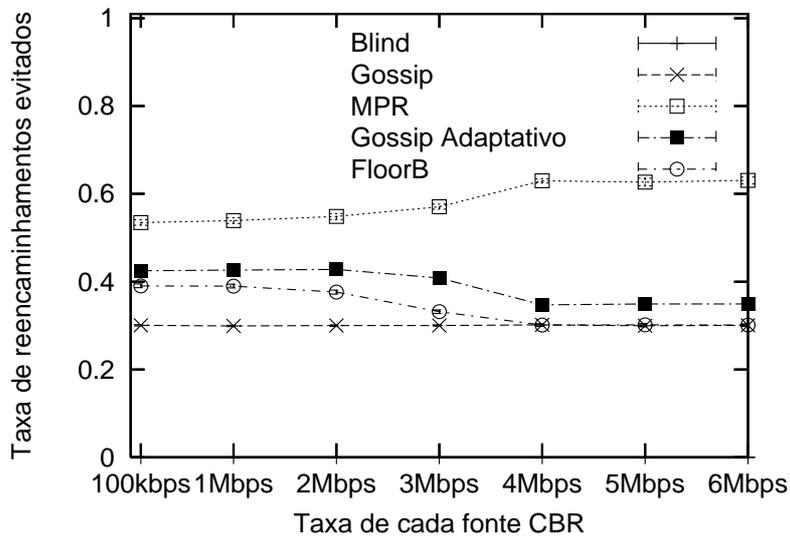
É interessante observar que o FloorB obteve desempenho em entrega superior ao do Gossip, que não é dependente de *hellos*, mesmo obtendo uma taxa de reencaminhamentos evitados igual ou superior ao mesmo. Este resultado indica mais uma vez a robustez do FloorB a perdas nas mensagens de *hello*.

Este desempenho antagônico das métricas utilizadas, observado nesta Figura 4.13 no desempenho do MPR e na Figura 4.1 para o Gossip Adaptativo, onde um mecanismo pode apresentar uma melhoria significativa em uma das métricas, como o MPR, para reencaminhamentos evitados com cargas de 4 a 6 Mbps, mas ao custo de um desempenho muito degradado na outra métrica, nos motivou a realizar uma avaliação integrada dos mecanismos.

Isto foi realizado através da aplicação de mecanismos de inundação em um protocolo de roteamento simples. Foi implementado um protocolo de roteamento do tipo



(a) Taxa de Entrega.



(b) Reencaminhamentos Evitados.

Figura 4.13: Cenário com mobilidade RWP, rede com carga variável, ns-2.

estado do enlace sem pesos, onde cada nó inunda a rede com mensagens periódicas que carregam sua lista de vizinhos. Após receber as mensagens de inundação, o nó aplica o algoritmo de Dijkstra. São computadas todas as rotas da rede, ou rotas de cada nó para todos os demais nós, ou seja, são obtidas  $N \times (N - 1)$  rotas. Para cada mecanismo de redução de inundação são comparados os efeitos sobre o protocolo de roteamento em relação ao número de destinos inalcançáveis, rotas quebradas em nós intermediários, e *loops* de roteamento. Destinos inalcançáveis ocorrem quando um nó não consegue computar uma rota para aquele destino, e portanto esta não é uma métrica acurada para determinar o desempenho do mecanismo, uma vez que não há garantias de que a rede é completamente conectada.

Já as rotas quebradas em nós intermediários oferece uma boa medida do desem-

penho dos mecanismos. Uma quebra intermediária ocorre quando um nó computa uma rota até o destino, mas quando esta rota é percorrida, salto a salto, um nó intermediário identifica o destino como inalcançável. Isto impacta significativamente o desempenho da rede, pois pacotes enviados serão encaminhados da origem até o nó intermediário, e então descartados, uma vez que a rota está quebrada.

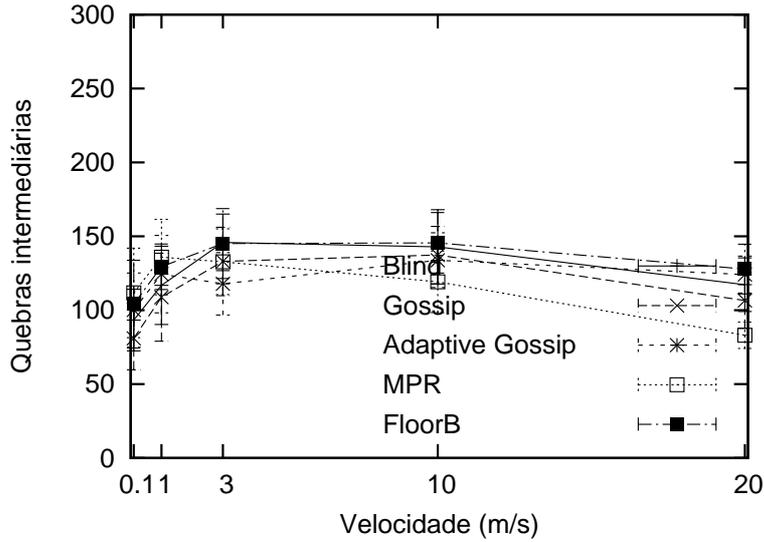
*Loops* de roteamento são problemas graves para desempenho de redes de uma forma geral, pois o pacote encaminhado pode retornar a nós anteriormente percorridos, e somente estratégias como contadores de TTL (*Time to Live*) podem remover estes pacotes em *loop*. Portanto, foi implementada uma verificação de todas as rotas para buscar os possíveis *loops*, seguindo cada rota salto por salto, e verificando se a visão da rede de cada nó percorrido, ou a rota computada, utilizava como próximo salto um nó previamente percorrido pela rota.

Desta forma, foi avaliado o desempenho do protocolo de roteamento quando associado aos mecanismos Blind, Gossip, Gossip Adaptativo, MPR e FloorB. Foi utilizado um cenário com mobilidade do tipo *Random Waypoint*, 200 nós, gerando um total de 39800 rotas, em uma área de 800m por 800m, com alcances de transmissão de 60, 80, 100 e 120m. A Figura 4.14 mostra os destinos inalcançáveis e as quebras intermediárias para alcance de transmissão de 80m. Em todas as simulações não ocorreram *loops*.

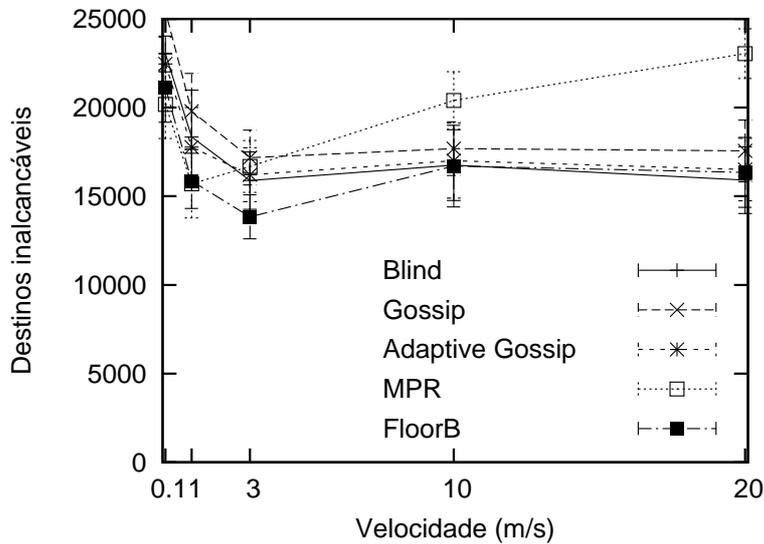
A Figura 4.14(a) apresenta as quebras intermediárias. Em um protocolo ideal é esperado que não existam tais quebras, porque um destino é alcançável ou não. Entretanto, uma mudança de topologia precisa ser propagada para todos os nós, e a perda, ou atraso na difusão, de mensagens de inundação pode provocar visões inconsistentes da topologia da rede. Porém, como podemos notar, o desempenho do protocolo de roteamento foi semelhante para todos os mecanismos de redução da inundação.

Já a Figura 4.14(b) apresenta os destinos inalcançáveis para o mesmo alcance de 80m. Com este alcance, o cenário representa uma rede com baixa conectividade, e existe um grande número de destinos inalcançáveis. Em relação a esta medida de desempenho, o protocolo de roteamento associado ao mecanismo MPR apresenta um desempenho degradado, quando utilizado em topologia com velocidades maiores ou iguais a 10m/s.

As simulações com os demais alcances apresentaram resultados muito semelhantes para todos os mecanismos. Com alcance de 60m a conectividade da rede é muito baixa, com muitos nós desconectados, não fornecendo resultados significativos. Com os alcances maiores, de 100 e 120m, a elevada redundância de caminhos fornecida pela boa conectividade, associado a um número menor de saltos entre os nós mais distantes, também levou a uma situação de desempenho equivalente para o protocolo de roteamento associado a todos os mecanismos.



(a) Quebras intermediárias.



(b) Destino inalcançável.

Figura 4.14: Desempenho do protocolo de roteamento.

## 4.2 2BSheld

Esta seção descreve as ideias utilizadas na implementação do mecanismo escalonador 2BSheld no ns-2, e descreve os cenários utilizados e avalia os resultados obtidos.

### 4.2.1 Implementação

Para avaliar a proposta apresentada na Seção 3.2, foi necessário utilizar um simulador que permitisse a adoção de uma camada de acesso ao meio do tipo TDMA. No simulador ns-2 a implementação de TDMA pressupõe que todos os nós estão dentro do mesmo alcance de transmissão, ou seja, considera uma rede de um único salto, configuração que não atenderia os requisitos de nossa proposta. Seria necessário

então obter um TDMA com reuso espacial.

Como dentro de nossa proposta não havia a necessidade de uma entidade centralizadora para distribuir a informação de *slot* a ser utilizado, e nem de como a sincronização entre nós seria mantida, optamos por desenvolver um acesso do STDMA derivado da implementação do CSMA/CA já existente.

Evidentemente é uma implementação limitada, mas que foi obtida a partir da eliminação da janela de contenção do CSMA/CA, e pela discretização do tempo utilizado para transmitir cada pacote. Isto foi feito através do estabelecimento do tempo necessário para transmitir um pacote padrão, que será o *slot*, e fazer a partir dele que todo o escalonamento de transmissão de pacotes do ns-2 ocorra em múltiplos deste tempo. Esta técnica permitiu a sincronização necessária ao STDMA, e a questão da decisão de *slots* fica em aberto, sendo no caso do mecanismo proposto realizada localmente, mas podendo ser realizada de outras formas de acordo com o escalonador que seja acrescentado.

Do ponto de vista de conflitos, caso o escalonamento não seja correto, este se refletirá, nesta implementação de CSMA/CA modificada, como uma colisão, e a consequente perda do pacote transmitido. Em relação ao CSMA/CA original, a contenção não ocorrerá em função da retirada da janela de contenção, ou seja, o *backoff* utilizado por todos os nós será sempre o mesmo.

As avaliações apresentadas nesta seção serão sempre dentro das condições descritas na Seção 3.2.1 e também parametrizada conforme a Tabela 3.1. As medidas de desempenho utilizadas são a latência de difusão, calculada como o tempo, normalizado pelo tempo de transmissão de um pacote, decorrido entre o início do envio de uma mensagem de inundação e o último recebimento da mesma, ou seja, os nós que não recebem a mensagem não são considerados, e a taxa de entrega, calculada como o número de nós que recebem a mensagem de inundação, entendido que cada mensagem é recebida uma única vez e cópias são descartadas, dividido pelo número total de nós.

Como a latência de difusão é normalizada em termos do tempo de transmissão de um pacote padrão, é possível realizar uma comparação adequada entre todos os mecanismos, mesmo no caso de se utilizar acesso ao meio do tipo CSMA/CA, pois desta forma, os atrasos introduzidos por conta de *backoff* ou contenções se refletem na latência da mesma forma que o aumento de *slots* no acesso ao meio TDMA.

## 4.2.2 Simulação no ns-2

A primeira avaliação do mecanismo 2BSheld foi realizada em uma rede em grade idêntica a da Seção 3.2.1. O mecanismo, implementado no simulador ns-2, é executado utilizando-se ou o acesso ao meio CSMA/CA ou o TDMA, ambos em con-

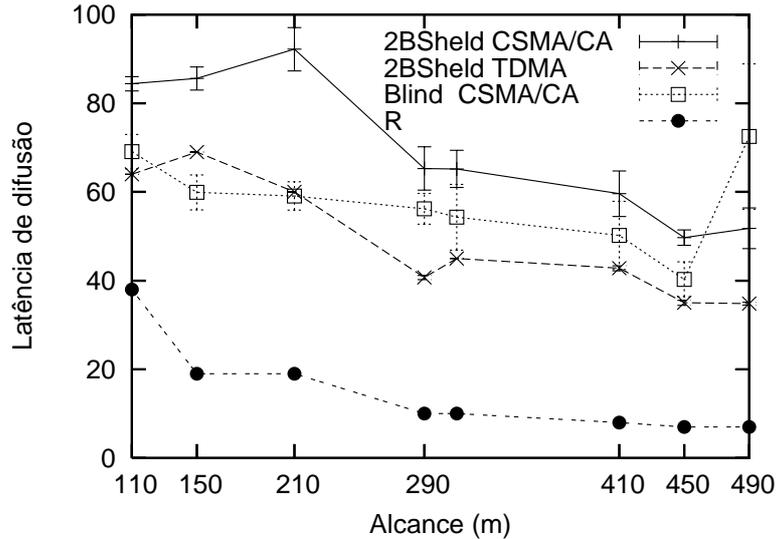


Figura 4.15: Latências de inundação do escalonador em CSMA/CA e TDMA.

junto com o mecanismo Blind. Os resultados são apresentados na Figura 4.15 onde observa-se um desempenho do 2BSheld sobre CSMA/CA pior do que o Blind diretamente sobre este mesmo acesso ao meio.

Isto se deve ao fato de que o mecanismo escalonador acaba provocando um deslocamento no tempo das transmissões dos nós, que depois serão novamente escalonados pelo processo de *backoff* e contenção que ocorre no CSMA/CA. Quando o mecanismo Blind é aplicado diretamente ao CSMA/CA, este último procedimento ocorre logo após o recebimento da mensagem, e os conflitos são reduzidos através do *carrier sense*, onde cada nó aguarda que o meio esteja livre para realizar sua transmissão.

Sobre o TDMA, o 2BSheld ainda não apresenta ganhos significativos de desempenho em relação ao Blind-CSMA/CA, o que poderia levar a se questionar sua utilidade. Entretanto, os mecanismos de redução de inundação não funcionam diretamente sobre o acesso ao meio TDMA sem a atuação de um escalonador.

No TDMA sem um escalonador há um excessivo número de colisões, por conta das ausências de detecção de portadora e *backoff*, e as mensagens não ultrapassam os primeiros saltos. Esta situação foi constatada através de simulações que mostraram que nesta configuração a taxa de entrega é muito reduzida. Neste caso, foi observado que a mensagem de inundação era recebida somente pelos vizinhos de 1 salto do emissor. As mensagens de inundação transmitidas por este nós, simultaneamente no próximo *slot* TDMA, colidiam, e não eram recebidas pelos nós do salto seguinte.

Por utilizar a vizinhança de dois saltos para suas decisões, o 2BSheld é naturalmente associável aos mecanismos FloorB e MPR, e os resultados desta associação entre os mecanismos de redução de inundação e o mecanismo escalonador são apre-

sentados na Figura 4.16. Pode-se observar que não há redução de latência com a associação ao FloorB, mas há um ganho com o MPR. Isto ocorre porque o FloorB não apresenta uma redução significativa de reencaminhamento de mensagens para este tipo de cenário, conforme mostrado na Seção 4.1.2, gerando muitas mensagens a serem escalonadas. Já no MPR, com um número reduzido de nós a serem escalonados, observa-se a redução da latência, obtendo valores bem próximos de  $R$ .

Apesar disto, para redes com maior grau de conectividade, onde o alcance é maior que  $410m$ , vemos que o escalonador não consegue a mesma aproximação do limite teórico  $R$  mesmo associado ao MPR. Isto se deve à forma como a decisão do *slot* é realizada, através da adição dos valores *lowPri* e *lowId*. Com este processo, um nó  $v$  pode ter sua transmissão escalonada em até  $V_1(v)$  *slots* a frente, provocando uma grande latência nestas redes com maior grau, principalmente para alcançar os nós finais, que estão à borda da rede.

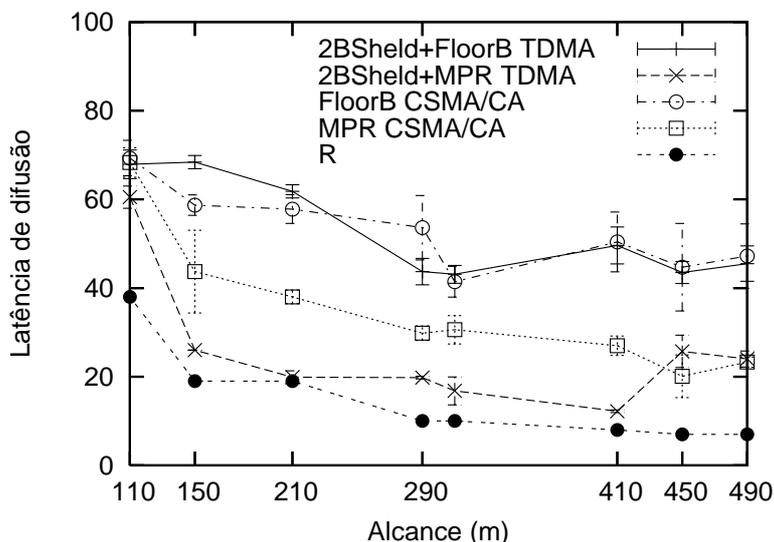


Figura 4.16: Latências de inundaç o do escalonador associado ao FloorB e MPR.

Por este motivo, foram realizados testes de desempenho com diversas funç es  $s(x, y)$  do mecanismo 2Bsheld, para se observar as lat ncias obtidas. Conforme descrito na Seç o 3.2 esta funç o determina qual o deslocamento de *slots* ser  utilizado no escalonamento da mensagem. A Figura 4.17 apresenta os resultados obtidos para a funç o emp rica  $s(x, y) = \text{round}(2 \cdot \log(x + y + 1)) + 1$  que apresentou menores lat ncias para estas redes com maior grau, quando comparados a funç o aditiva original, possivelmente ao custo de um aumento de conflitos de escalonamento, entretanto, sem afetar a taxa de entrega dos mecanismos nestes cen rios<sup>1</sup>. O conceito utilizado na escolha desta funç o foi de oferecer sa da 1 para  $x = y = 0$ , e apresentar

<sup>1</sup>As taxas de entrega obtidas foram superiores a 99%, exceto para alguns mecanismos de reduç o de inundaç o na Figura 3.9(a), que tiveram desempenho de acordo com o apresentado na Seç o 4.1.3, ou quando apresentadas graficamente.

valores crescentes com  $x$  e  $y$ , mas de forma menos acentuada do que na adição.

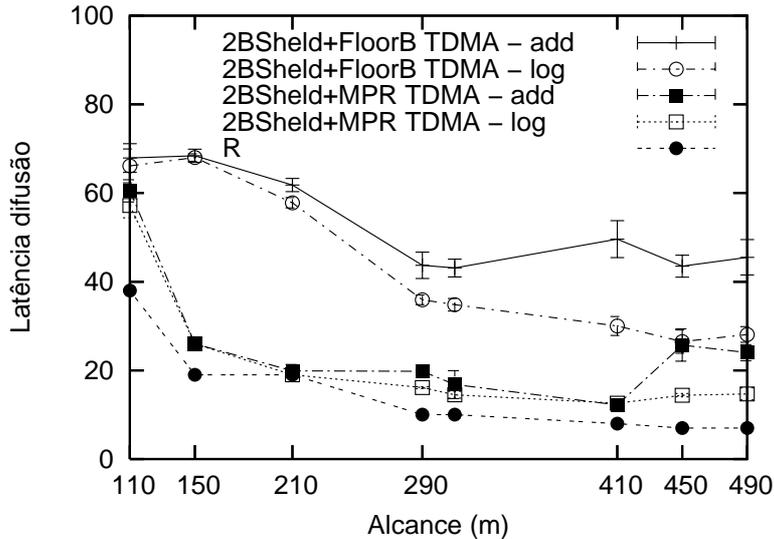


Figura 4.17: Latências de inundação do escalonador com função log.

Em função da observação acima, é proposto um novo critério para atribuição dos *slots*, ou seja, uma nova função  $s(x, y)$ . Neste caso, o conceito utilizado é de que à medida que a mensagem “envelhece” na rede, deve-se procurar escaloná-la mais rapidamente. Este conceito pode ser utilizado para determinar um limite de tempo para escalonamento, em função de  $R$ , ou caso este não seja conhecido, um limite de tempo em função de número de *slots*. A decisão deste limiar pode ser da aplicação que realiza a inundação. Por exemplo, caso a inundação seja um processo periódico, com alguma informação de atualização de estado do nó, não fará sentido receber uma inundação de período anterior ao atual, ou seja, é interessante expirar a mensagem, ou mesmo perdê-la por colisão ou conflito de escalonamento, e não continuar escalonando seu envio após um certo tempo.

Como exemplo deste limite de tempo e de aplicações, podemos citar a própria inundação utilizada pelo protocolo de roteamento OLSR, que é periódica com intervalo de  $5s$ , ou a utilizada em descoberta de serviços na proposta em [3], que possui o mesmo intervalo.

Desta forma, é possível formalizar uma nova função  $s(x, y)$  onde  $s$  é o valor do escalonamento realizado por um nó receptor  $r$  para a mensagem de inundação recebida. Tem-se que o maior valor possível de  $s$  será  $s_{max} = |V_1(r)|$ . Sendo  $T$  o limitador de expiração da mensagem, em *slots*, normalmente teremos  $T \gg R$ , e sabendo que  $t$  é o *slot* atual, e  $h$  o número de saltos percorridos pela mensagem até chegar a  $r$ , será necessário que  $s \leq T - t$ , ou,  $s_{max} = T - t$  para que a mensagem possa ser transmitida antes da expiração.

Entretanto, dentro da visão de  $r$ , após  $t$  *slots* a mensagem percorreu  $h$  saltos. Considerando este o deslocamento médio da mensagem na rede, deseja-se que após

$T$  slots a mensagem tenha percorrido os  $R$  saltos da rede. Portanto, é possível estabelecer uma proporcionalidade, obtendo  $T = \frac{Rt}{h}$  e reescrevendo  $s_{max}$ , obtém-se:

$$s_{max} = \frac{Rt}{h} - t \quad (4.4)$$

Estabelecendo também uma proporção entre  $s_{max}$  calculado na Equação 4.4 e o valor  $|V_1(r)|$ , pode-se admitir a mesma proporcionalidade entre  $s(lowPri, lowId) = lowPri + lowId + 1$  e o valor de  $s$  ajustado ( $s'$ ). Então temos:

$$s'(lowPri, lowId) = \frac{s(lowPri, lowId) \times (\frac{Rt}{h} - t)}{|V_1(r)|} = \frac{(lowPri + lowId + 1)t(R - h)}{h|V_1(r)|} \quad (4.5)$$

É importante observar que, com a exceção de  $R$ , as demais variáveis da Equação 4.5 são obtidas localmente ou da própria mensagem de inundação. Caso não se conheça  $R$ , a Equação 4.5 pode ser reescrita facilmente em função de  $T$ , conforme:

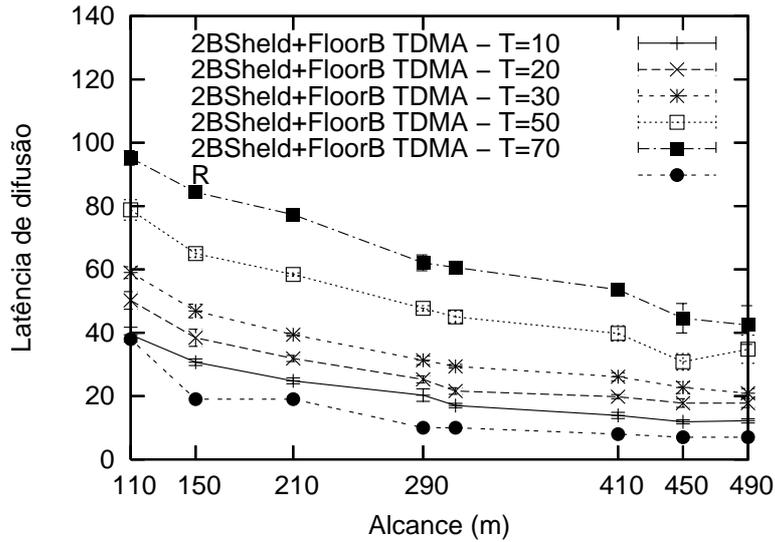
$$s'(lowPri, lowId) = \frac{(lowPri + lowId + 1)(T - t)}{|V_1(r)|} \quad (4.6)$$

Na Figura 4.18(a), temos os resultados de latência para o escalonamento realizado associado ao FloorB e com alguns valores de limitador  $T$ . Observa-se nesta figura que as latências máximas não ficam restritas ao teto indicado. Isto se deve ao fato de que dentro de um *slot*  $s$ , o escalonador irá distribuir o escalonamento de forma que ao último nó a ser escalonado será alocado o *slot*  $T - s$ . Entretanto, para  $s \geq T$ , todo escalonamento será feito para o próximo *slot*, e não ocorrendo conflito ou colisão a mensagem será recebida pelos vizinhos, e novos escalonamentos serão feitos, sucessivamente, sempre para o *slot* seguinte.

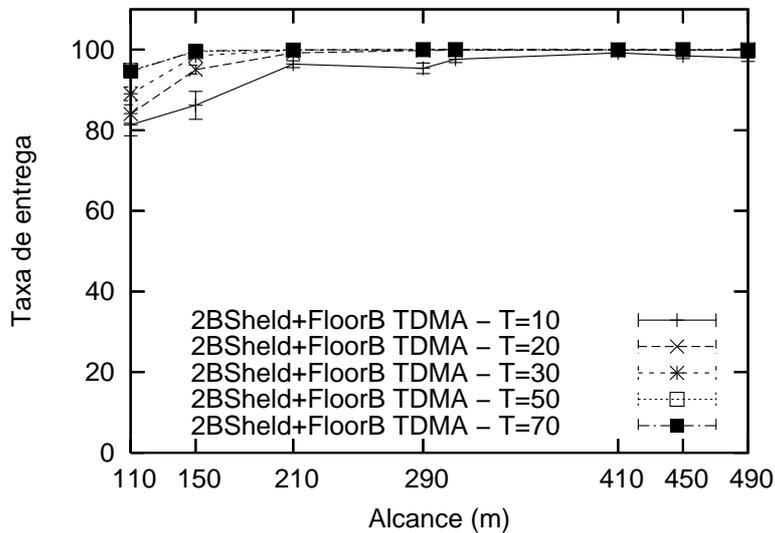
Na Figura 4.18(b), pode-se observar que a adoção do limitador  $T$  permite a redução da latência sem implicar em redução significativa da taxa de entrega quando  $T \gg R$ .

Considerando que a abordagem adotada é simples e não depende de conhecimento da topologia da rede, os resultados apresentados sugerem elevada eficiência nos cenários utilizados. Estes cenários também tornam nítida a influência do limitador  $T$  para o tempo total de escalonamento, indicando que escolhas superestimadas podem levar a escalonamentos muito longos. No passo seguinte, o mecanismo é avaliado em um tipo de cenário ainda não abordado pelos trabalhos já citados que tratam de escalonamento de inundação: aqueles que apresentam mobilidade.

As Figuras 4.19 e 4.20 apresentam os resultados do escalonador em redes móveis, para alguns valores de  $T$ . Nestes cenários, foram colocados 200 nós com alcance de 120 metros em uma área de  $800m \times 800m$ , com mobilidade do tipo *Random Waypoint* e velocidades variando de  $0m/s$  até o máximo indicado. Na Figura 4.19 são apresentadas a latência de difusão e a taxa de entrega do escalonador associado



(a) Latências de inundação



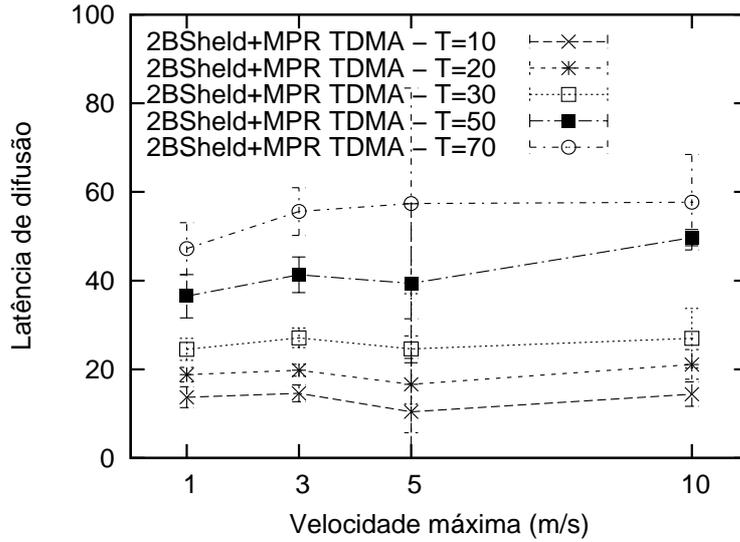
(b) Taxas de entrega

Figura 4.18: Latência e taxa de entrega com limite  $T$ .

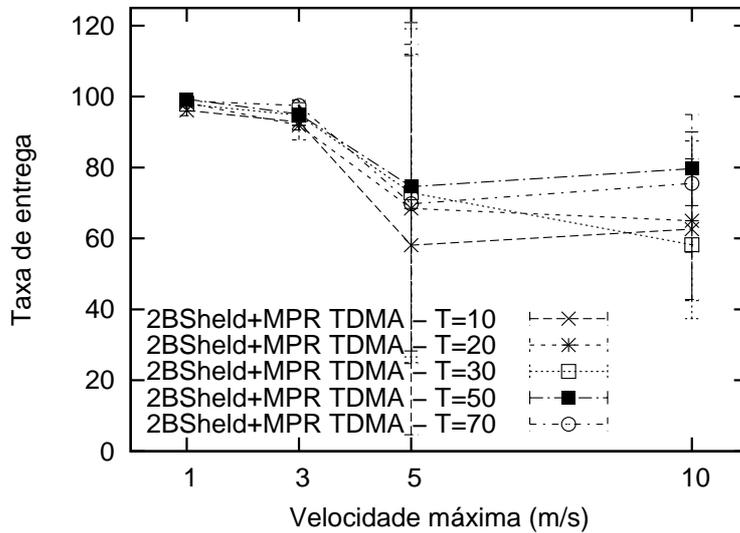
ao MPR, enquanto na Figura 4.20 temos o mesmo resultado com o FloorB.

Observa-se que a latência não sofre grande influência da mobilidade apresentada pela rede quando utilizado limite  $T$ , sendo ligeiramente inferior com o MPR em relação ao FloorB. Entretanto, para altas velocidades o mecanismo MPR se torna menos eficiente que o FloorB em termos de taxa de entrega e também apresentou barra de erros maiores, resultado já apresentado em [19] independente de escalonador.

Isto ocorre porque em algumas rodadas a taxa de entrega foi muito baixa, pois a mensagem de inundação não ultrapassou os primeiros saltos da rede. Nestes cenários, o nó emissor, devido à mobilidade, perde a vizinhança com os MPRs, e a inundação ocorre sem que o mecanismo de controle da inundação tenha sido atualizado pelos *hellos*. Esta situação aleatoriamente foi mais grave com a velocidade de  $5m/s$ . Neste



(a) Latência



(b) Taxa de Entrega

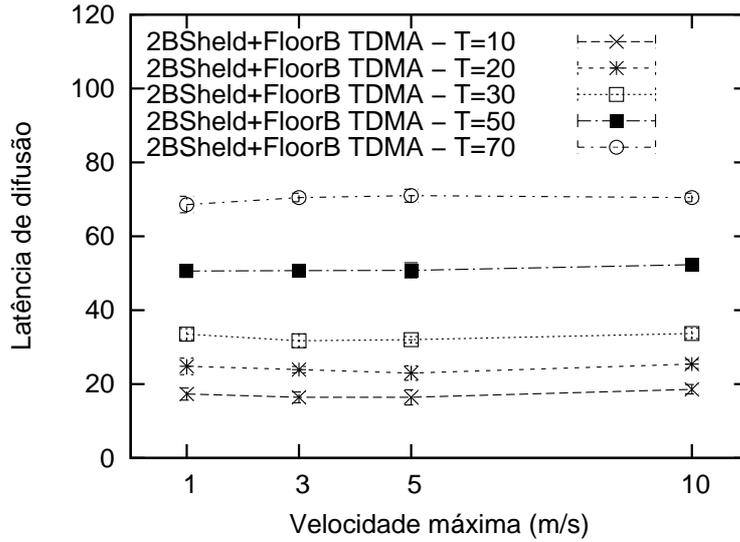
Figura 4.19: Escalonamento de inundação com mobilidade - MPR, limite T.

caso, a barra de erro poderia ser reduzida com o aumento do número de rodadas, mas efetivamente, o aumento do número de rodadas iria mascarar um problema, o de que o MPR tem uma boa média de taxa de entrega, mas eventualmente pode apresentar taxas quase nulas. Portanto, este resultado evidencia que a associação 2BSheld+FloorB é mais estável frente à mobilidade do que 2BSheld+MPR.

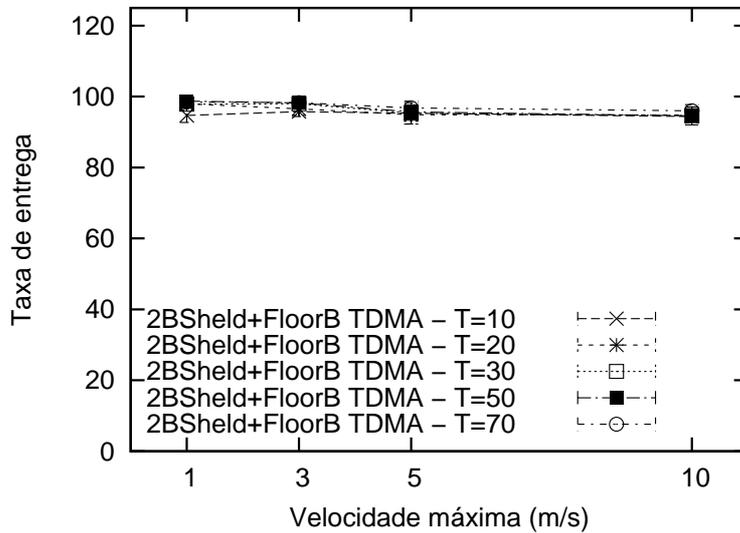
### 4.3 Conclusões

Este capítulo apresentou uma ampla avaliação das propostas desta tese, comparando-as com outros mecanismos ou limites teóricos existentes na literatura.

A Seção 4.1 inicialmente apresenta as opções de implementações utilizadas e em



(a) Latência



(b) Taxa de Entrega

Figura 4.20: Escalonamento de inundação com mobilidade - FloorB, limiteT.

seguida apresenta os resultados de diversos mecanismos de redução de inundação, incluindo a proposta FloorB (*Flooding control through neighborhood Representation by Bloom filters*), nas duas implementações, em simulador próprio e no simulador ns-2, utilizando 3 tipos de cenários: grade, mobilidade aleatória e mobilidade social.

Em seguida, na Seção 4.2, são apresentados os resultados da proposta do mecanismo de escalonamento de inundação 2BSheld (*2-hop aware Broadcast Scheduling with local decision*). Nesta seção são elaboradas e avaliadas mais duas funções de escalonamento do mecanismo. Ainda nesta seção o mecanismo 2BSheld é avaliado em dois tipos de cenários, em grade e com mobilidade, utilizando-o com acesso ao meio CSMA/CA e TDMA, ou conjugado com mecanismos de redução de inundação FloorB ou MPR.

No próximo capítulo são apresentadas as conclusões finais, ressaltadas as principais contribuições desta tese, e apontadas perspectivas para trabalhos futuros.

# Capítulo 5

## Conclusões e Trabalhos Futuros

Neste capítulo apresentamos nossas conclusões, divididas em três partes, a primeira acerca da proposta FloorB, a segunda sobre a proposta 2BSheld, e uma última com considerações finais e perspectivas de trabalhos futuros.

Os trabalhos apresentados nos capítulos anteriores tratam a questão de inundação em redes *ad hoc* móveis, tanto em relação à redução do número de mensagens reencaminhadas, quanto ao controle da latência através de mecanismos escalonados.

O foco desta tese, de forma geral, não foi desenvolver algoritmos ou mecanismos determinísticos com objetivo de alcançar resultados ótimos, mas sim mecanismos que possam atender a requisitos funcionais de taxa de entrega e latência, tratando assim o problema de forma probabilística, como realizado no FloorB, por entender que as questões de mobilidade e propagação levam a incertezas quanto ao real estado da rede em cada instante, e portanto, oferecendo uma caráter probabilístico à visão que cada nó possui da própria rede *ad hoc*.

Dentro desta ótica, diversos mecanismos foram avaliados, e foi possível observar suas limitações, cada qual oferecendo um desempenho de acordo com as características da rede onde foram aplicados.

### 5.1 FloorB

Uma importante contribuição desta tese é a avaliação comparativa entre diversos mecanismos de inundação, alguns determinísticos tais como Blind e MPR, e outros probabilísticos, tais como Gossip, Gossip Adaptativo e FloorB. Nestas avaliações foram utilizados alguns tipos de cenários, incluindo mobilidade do tipo social. Nosso entendimento é o de que não há na literatura avaliações comparativas entre mecanismos determinísticos e probabilísticos, e com cenários com topologias não uniformes.

Além disto, foi proposto um mecanismo de redução de inundação, FloorB, que apresentou diversas vantagens frente a outros mecanismos probabilísticos, como o

Gossip e Gossip Adaptativo.

O FloorB também apresentou algumas vantagens em relação ao mecanismo MPR, tais como:

- Tamanho fixo e reduzido das mensagens de *hello*;
- Algoritmo com complexidade de tempo inferior, em  $O(\Delta)$ ;
- Maior estabilidade na taxa de entrega, tanto em relação à mobilidade quanto às perdas de mensagens.
- Possibilidade de redução no número de mensagens de controle ao se utilizar *hellos* com intervalos maiores.

Ainda em relação ao MPR, apesar do FloorB apresentar taxas de reencaminhamentos evitados ligeiramente inferiores, seu desempenho foi quase equivalente, principalmente em redes com maior densidade, onde os efeitos nocivos da inundação são mais graves.

Outra importante contribuição decorrente desta tese foi o desenvolvimento de um simulador genérico para avaliar mecanismos de redução de inundação em tcl, no qual outros mecanismos podem ser acrescentados e avaliados de forma rápida.

Também foi realizada a adaptação do código para o ns-2, permitindo validar os resultados já obtidos pelo simulador próprio, e verificar o desempenho dos mecanismos quando utilizados sobre uma camada de enlace tal como a do IEEE 802.11, incluindo questões como colisões e a carga a que a rede foi submetida.

Ainda com a implementação no ns-2, foi desenvolvido um protocolo de roteamento do tipo estado do enlace simples, onde as mensagens de inundação com as informações de enlace eram disseminadas na rede com cada um dos mecanismos de redução de inundação, permitindo verificar o impacto de cada mecanismo sobre o funcionamento do protocolo.

O mecanismo proposto, FloorB, se mostrou bastante eficiente em todos os cenários, e sua simplicidade indica que é possível sua implementação em ambiente real.

## 5.2 2BSheld

Esta tese também apresentou uma abordagem para o problema de latência mínima para escalonamento de inundação (MLBS). Apesar de ser um problema NP-Completo, tratado por algumas propostas na literatura, este trabalho mostra que vários algoritmos aproximados existentes apresentam valores limites mais próximos

dos limites superiores esperados do que dos limites inferiores. Também foi apresentada a latência obtida por mecanismos de redução de inundação em alguns cenários, indicando que ao se realizar uma escolha adequada de nós reencaminhadores, a latência pode ser diminuída consideravelmente.

Diante destas observações, esta tese ainda apresenta um mecanismo simples, baseado em um algoritmo “guloso”, que estabelece um escalonamento através de decisões locais. Apesar de não garantir um limite superior e permitir a ocorrência de conflitos, o mecanismo apresentou bom desempenho quando aplicado em cenários desde redes pouco conectadas até redes com grau elevado, ou mesmo em redes com mobilidade variável. Além disto, o mecanismo provê um modo de controlar a duração da inundação na rede, eventualmente reduzindo a taxa de entrega.

Como principal contribuição, nesta parte do trabalho, esta tese indica valores de latência esperados por mecanismos escalonadores de inundação em alguns cenários, em função do tempo de transmissão do pacote e do número de saltos (diâmetro) da rede. Além disso, são estabelecidas algumas hipóteses para a criação de mecanismos realmente distribuídos e baseados em decisões locais. Conforme os resultados, é possível identificar que soluções aproximadas, mesmo utilizando informações globais ou mecanismos complexos, não oferecem ganhos significativos de desempenho, ou seja, apesar do problema ser comprovadamente NP-Completo, a utilização de heurísticas simples permite obter, sem garantias, resultados próximos ao ótimo, em boa parte dos cenários.

Em relação especificamente ao mecanismo, 2BSheld, o fato dele não garantir que não ocorram conflitos, por basear suas decisões em uma visão de 2 saltos da rede, pode ser superada com a adição de algum processo de aleatoriedade e contenção, que mesmo implicando em uma maior latência final, pode ser aceitável.

É importante notar que o mecanismo foi implementado de forma simplificada, sem levar em conta questões fundamentais para mecanismos escalonadores reais em STDMA, como a sincronização entre todos os nós para determinação dos *slots* ou a separação entre o escalonamento das mensagens de controle e inerentes ao próprio mecanismo (*hellos*, por exemplo) e as mensagens de inundação ou outras mensagens de dados em uso na rede. Esta simplificação pode limitar uma aplicação real imediata do 2BSheld da forma como está proposto e implementado, mas não compromete a avaliação de desempenho realizada.

### 5.3 Trabalhos Futuros

Como trabalhos futuros, entendemos que a proposta de redução de inundação FloorB é bastante realística, e pode facilmente ser implementada em conjunto com aplicações que façam uso de inundações, como mecanismos de descoberta de serviços e

protocolos de roteamento para redes *ad hoc*.

O FloorB também pode ter seu desempenho melhorado, principalmente em relação à economia proporcionada pelos reencaminhamentos evitados, ao ser associado a algum dos esquemas de redução de inundação propostos em [6], como por exemplo contagem de mensagens.

Também há uma interessante área de estudos em relação à determinação e divulgação de vizinhança. Ampliar a avaliação do uso de *hellos*, incluindo questões como perdas de mensagens, enlaces assimétricos e controle de potência podem fornecer novas perspectivas para os mecanismos de redução da inundação, inclusive o MPR.

Em relação ao mecanismo escalonador, acreditamos que os conceitos e hipóteses apresentados possam ser mais bem avaliados, aplicando-se um modelo físico e de enlace de forma mais realista, uma vez que a maior parte das propostas de escalonamento se baseia em modelos de protocolo. Acreditamos que estas avaliações sejam uma contribuição importante que pode ser agregada ao trabalho já realizado.

Também vemos como uma importante alternativa para o desenvolvimento do mecanismo escalonador, implementá-lo de forma conjunta, numa abordagem *cross-design*, com os mecanismos de redução de inundação MPR e FloorB, utilizando então de forma mais ampla as informações obtidas e fornecidas por eles.

Para uma implementação mais real e completa do 2BSheld no ns-2, seria importante o desenvolvimento de uma camada de acesso ao meio do tipo STDMA, que segundo nosso entendimento não existe disponível.

# Referências Bibliográficas

- [1] CLAUSEN, T., JACQUET, P. “RFC 3626 - Optimized Link State Routing Protocol OLSR”, *IETF Request for Comments*, out. 2003. Disponível em: <ftp://ftp.rfc-editor.org/in-notes/rfc3626.txt>.
- [2] PERKINS, C., BELDING-ROYER, E., DAS, S. “RFC 3561 - Ad hoc On-Demand Distance Vector AODV Routing”, *IETF Request for Comments*, jul. 2003. Disponível em: <ftp://ftp.rfc-editor.org/in-notes/rfc3561.txt>.
- [3] LENDERS, V., MAY, M., PLATTNER, B. “Service Discovery in Mobile Ad Hoc Networks: A Field Theoretic Approach”. In: *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 120–130, jun. 2005.
- [4] AUGUSTO, C. H. P., DE REZENDE, J. F. “An Adaptive Approach to Service Discovery in Ad Hoc Networks”. In: *8th IFIP IEEE International Conference on Mobile and Wireless Communications Network, 2006.*, pp. 61–75, ago. 2006. Disponível em: <http://www.wcc-2006.org/>.
- [5] KARUMANCHI, G., MURALIDHARAN, S., PRAKASH, R. “Information Dissemination in Partitionable Mobile Ad Hoc Networks”. In: *Proceedings of IEEE Symposium on Reliable Distributed Systems (SRDS)*, pp. 4–13, out. 1999.
- [6] NI, S.-Y., TSENG, Y.-C., CHEN, Y.-S., et al. “The broadcast storm problem in a mobile ad hoc network”. In: *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pp. 151–162, New York, NY, USA, 1999. ACM. ISBN: 1-58113-142-9. doi: <http://doi.acm.org/10.1145/313451.313525>.
- [7] COSTA, L. H. M. K., AMORIM, M. D., FDIDA, S. “Reducing Latency and Overhead of Route Repair with Controlled Flooding”, *ACM/Kluwer Wireless Networks*, v. 10, pp. 347–358, July 2004. ISSN: 1022-0038.

- [8] FAHEEM, Y., ROUGIER, J. L. “Loop avoidance for fish-eye OLSR in sparse wireless mesh networks”. In: *WONS’09 Proceedings of the Sixth international conference on Wireless On-Demand Network Systems and Services*, fev. 2009.
- [9] HAAS, Z., HALPERN, J. Y., LI, L. “Gossip-Based Ad Hoc Routing”. In: *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications*, v. 3, pp. 1707 – 1716, jun. 2002.
- [10] LAOUITI, A., QAYYUM, A., VIENNOT, L. “Multipoint Relaying: An Efficient Technique for Flooding in Mobile Wireless Networks”. In: *35th Annual Hawaii International Conference on System Sciences (HICSS’2002)*, jan. 2002.
- [11] DRABKIN, V., FRIEDMAN, R., KLIOT, G., et al. “RAPID: Reliable Probabilistic Dissemination in Wireless Ad-Hoc Networks”. In: *SRDS ’07: Proceedings of the 26th IEEE International Symposium on Reliable Distributed Systems*, pp. 13–22, Washington, DC, USA, out. 2007. IEEE Computer Society. ISBN: 0-7695-2995-X.
- [12] CHLAMTAC, I., KUTTEN, S. “On Broadcasting in Radio Networks—Problem Analysis and Protocol Design”, *Communications, IEEE Transactions on*, v. 33, n. 12, pp. 1240 – 1246, dec 1985.
- [13] CHLAMTAC, I., WEINSTEIN, O. “The wave expansion approach to broadcasting in multihop radio networks”, *Communications, IEEE Transactions on*, v. 39, n. 3, pp. 426 –433, mar 1991.
- [14] HUANG, S. C.-H., WAN, P.-J., JIA, X., et al. “Minimum-Latency Broadcast Scheduling in Wireless Ad Hoc Networks”. In: *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 733–739, maio 2007.
- [15] GANDHI, R., MISHRA, A., PARTHASARATHY, S. “Minimizing broadcast latency and redundancy in ad hoc networks”, *IEEE/ACM Trans. Netw.*, v. 16, n. 4, pp. 840–851, 2008. ISSN: 1063-6692. doi: <http://dx.doi.org/10.1109/TNET.2007.905588>.
- [16] YEO, J., LEE, H., KIM, S. “An efficient broadcast scheduling algorithm for TDMA ad-hoc networks”, *Computers & Operations Research*, v. 29, n. 13, pp. 1793 – 1806, 2002. ISSN: 0305-0548. doi: DOI:10.1016/S0305-0548(01)00057-0.

- [17] VERGADOS, D. J., MANOLARAKI, M.-Y., VERGADOS, D. D. “Evaluation of broadcast scheduling algorithms for ad-hoc TDMA networks”. In: *1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009. Wireless VITAE 2009.*, pp. 394 – 398, maio 2009.
- [18] YASSEIN, M. B., OULD-KHAOUA, M., PAPANASTASIOU, S. “On the Performance of Probabilistic Flooding in Mobile Ad Hoc Networks”, *ICPADS, 11th International Conference on Parallel and Distributed Systems*, v. 02, pp. 125–129, 2005. ISSN: 1521-9097. doi: <http://doi.ieeecomputersociety.org/10.1109/ICPADS.2005.208>.
- [19] AUGUSTO, C. H. P., DE REZENDE, J. F. “Inundação Probabilística em Redes Ad Hoc Utilizando Sumário de Vizinhança Através de Filtros de Bloom”. In: *XXVII Simpósio Brasileiro de Redes de Computadores - SBRC2009*, maio 2009.
- [20] AUGUSTO, C. H. P., DE REZENDE, J. F. “FloorB: Mecanismo de Controle de Inundação para Redes Ad Hoc Móvel”, *Revista Brasileira de Redes de Computadores e Sistemas Distribuídos (RES D)*, v. 2, n. 1, pp. 35–49, jun. 2009.
- [21] MUSOLESI, M., MASCOLO, C. “A Community based Mobility Model for Ad Hoc Network Research”. In: *Proceedings of the 2nd ACM/SIGMOBILE International Workshop on Multi-hop Ad Hoc Networks: from theory to reality (REALMAN’06)*. ACM Press, May 2006.
- [22] AUGUSTO, C. H. P., DE REZENDE, J. F. “Escalonamento Distribuído de Inundações em Redes Ad Hoc Móveis”. In: *XXVIII Simpósio Brasileiro de Redes de Computadores - SBRC2010*, maio 2010.
- [23] AUGUSTO, C. H. P., DE REZENDE, J. F. “Distributed Broadcast Scheduling in Mobile Ad Hoc Networks”. In: *9th IFIP Annual Mediterranean Ad Hoc Networking Workshop - Med-Hoc-Net2010*, jun. 2010.
- [24] FUJIE, T. “An exact algorithm for the maximum leaf spanning tree problem”, *Computers and Operations Research*, v. 30, n. 13, pp. 1931 – 1944, 2003. ISSN: 0305-0548. doi: DOI:10.1016/S0305-0548(02)00117-X.
- [25] CARTIGNY, J., SIMPLOT, D. “Border node retransmission based probabilistic broadcast protocols in ad-hoc networks”. In: *Telecommunication Systems*, v. 22, pp. 189–204. Kluwer Academic Publishers, jan. 2003.

- [26] STAUFFER, A. O., BARBOSA, V. C. “Probabilistic heuristics for disseminating information in networks”, *IEEE/ACM Trans. Netw.*, v. 15, pp. 425–435, April 2007. ISSN: 1063-6692. doi: <http://dx.doi.org/10.1109/TNET.2007.892877>. Disponível em: <http://dx.doi.org/10.1109/TNET.2007.892877>.
- [27] CALOMME, S., LEDUC, G. “The Critical Neighbourhood Range for Asymptotic Overlay Connectivity in Ad Hoc Networks”, *Ad Hoc & Sensor Wireless Networks*, v. 2, n. 2, pp. 169–187, 2006.
- [28] CALOMME, S., LEDUC, G. “Efficient and Resilient Overlay Topologies over Ad Hoc Networks”. In: *International Workshop on Self-Organizing Systems (IWSOS 2007)*, pp. 44–58, set. 2007.
- [29] CALOMME, S., LEDUC, G. “An overlay maintenance protocol for overlay routing on top of ad hoc networks”. In: *IFIP Networking 2008*, pp. 48–59, maio 2008.
- [30] BASAGNI, S., MASTROGIOVANNI, M., PETRIOLI, C. “A Performance Comparison of Protocols for Clustering and Backbone Formation in Large Scale Ad Hoc Networks”. In: *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, pp. 70–79, out. 2004.
- [31] BASAGNI, S., MASTROGIOVANNI, M., PANCONESI, A., et al. “Localized Protocols for Ad Hoc Clustering and Backbone Formation: A Performance Comparison”. In: *Parallel and Distributed Systems, IEEE Transactions on*, v. 17, pp. 292–306, abr. 2006.
- [32] WAN, P.-J., ALZOUBI, K. M., FRIEDER, O. “Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks”. In: *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, v. 3, pp. 1597–1604, jun. 2002.
- [33] KOZAT, U. C., TASSIULAS, L., KONDYLLIS, G., et al. “Virtual Dynamic Backbone for Mobile Ad hoc Networks”. In: *IEEE International Conference on Communications, 2001. ICC 2001.*, v. 1, pp. 250–255, jun. 2001.
- [34] BUSSON, A., MITTON, N., FLEURY, E. “An analysis of the MPR selection in OLSR”. In: *Spatial Stochastic Modeling of Wireless Networks, SpasWIN 05*, Trento, Italy, April 2005.

- [35] BUSSON, A., MITTON, N., FLEURY, E. “An analysis of the MPR selection in OLSR and consequences.” In: *The fourth Annual Mediterranean Ad Hoc Networking Workshop, MED-HOC-NET 05*, Ile de Porquerolles, France, June 2005.
- [36] NGUYEN, D., MINET, P. “Analysis of MPR Selection in the OLSR Protocol”. In: *AINAW '07: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops*, pp. 887–892, 2007. ISBN: 0-7695-2847-3. doi: <http://dx.doi.org/10.1109/AINAW.2007.94>.
- [37] CAMPISTA, M. E. M., COSTA, L. H. M. K., DUARTE, O. C. M. B. “WPR: A Proactive Routing Protocol Tailored to Wireless Mesh Networks”. In: *IEEE Globecom 2008 Ad Hoc, Sensor and Mesh Networking Symposium (GC'08 AHSN)*, dez. 2008.
- [38] AUGUSTO, C. H. P., CARVALHO, C. B., DA SILVA, M. W. R., et al. “Impacto do Roteamento no Escalonamento de Enlaces em Redes em Malha Sem Fio”. In: *XXVIII Simpósio Brasileiro de Redes de Computadores - SBRC2010*, maio 2010.
- [39] BADIA, L., ERTA, A., LENZINI, L., et al. “A General Interference-Aware Framework for Joint Routing and Link Scheduling in Wireless Mesh Networks”, *IEEE Network*, v. 22, n. 1, pp. 32–38, jan. 2008. doi: 10.1109/MNET.2008.4435906.
- [40] ZHANG, X.-Z. “Efficient Broadcast Scheduling Based on Fuzzy Clustering and Hopfield Network for Ad hoc Networks”. In: *International Conference on Machine Learning and Cybernetics, ICMLC 2007*, v. 6, pp. 3255–3260, ago. 2007.
- [41] WU, X., SHARIF, B., HINTON, O., et al. “Solving optimum TDMA broadcast scheduling in mobile ad hoc networks: a competent permutation genetic algorithm approach”, *Communications, IEE Proceedings-*, v. 152, n. 6, pp. 780–788, dez. 2005.
- [42] BLOOM, B. H. “Spacetime trade-offs in hash coding with allowable errors”. In: *Communications of the ACM*, v. 13, pp. 422–426, New York, NY, USA, 1970. ACM. doi: <http://doi.acm.org/10.1145/362686.362692>.
- [43] MITZENMACHER, M., BRODER, A. “Network Applications of Bloom Filters: (A Survey)”. In: *Internet Mathematics*, v. 1, pp. 485–509, 2004.

- [44] FERNANDES, N. C., MOREIRA, M. D. D., DUARTE, O. C. M. B. “An Efficient Filter-based Addressing Protocol for Autoconfiguration of Mobile Ad Hoc Networks”. In: *IEEE INFOCOM 2009*, abr. 2009.
- [45] CARTER, J. L., WEGMAN, M. N. “Universal classes of hash functions”, *Journal of Computer and System Sciences*, v. 18, n. 2, pp. 143 – 154, 1979. ISSN: 0022-0000. doi: DOI:10.1016/0022-0000(79)90044-8. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0022000079900448>>.
- [46] LAUFER, R. P. *Rastreamento de Pacotes IP contra Ataques de Negação de Serviço*. Tese de Mestrado, COPPE/PEE/UFRJ, Rio de Janeiro, RJ, Brasil, set. 2005.
- [47] WIKIPEDIA. “Fermat number”. 2011. Disponível em: <[http://en.wikipedia.org/wiki/Fermat\\_number](http://en.wikipedia.org/wiki/Fermat_number)>. [http://en.wikipedia.org/wiki/Fermat\\_number](http://en.wikipedia.org/wiki/Fermat_number) - último acesso em 25/05/2011.
- [48] NS-2. “The Network Simulator - ns-2”. 1995. <http://www.isi.edu/nsnam/ns/> - último acesso em 17/12/2008.
- [49] JOHN OUSTERHOUT. “Tcl - Tool Command Language”, <http://www.stanford.edu/ouster/cgi-bin/tclHistory.php>, 1988.
- [50] CARDOSO, K. V., DE REZENDE, J. F. “Adaptação Automática de Taxa em Redes 802.11 Densas”. In: *XXVI Simpósio Brasileiro de Redes de Computadores - SBRC2008*, maio 2008.