

FloorB: Mecanismo de Controle de Inundação para Redes Ad Hoc Móveis

Carlos Henrique Pereira Augusto¹, José Ferreira de Rezende¹

¹GTA - PEE - COPPE – Universidade Federal do Rio de Janeiro (UFRJ)

Caixa Postal 68.504 – 21.945-970

Rio de Janeiro – RJ – Brasil

{chenrique | rezende}@gta.ufrj.br

Abstract

Inundação é um mecanismo fundamental no funcionamento de 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. Em função disto, existem diversas propostas para minimizar esses efeitos, que por outro lado têm baixa eficiência pelas próprias características das redes ad hoc, como falta de infra-estrutura e mobilidade dos nós. Para suplantar este problema, este artigo propõe o mecanismo FloorB que controla a inundação através do conhecimento resumido, por filtros de Bloom, da vizinhança de até dois saltos. A eficiência de tal mecanismo, comparativamente a outras propostas determinísticas ou probabilísticas, é avaliada através de simulação.

Keywords: Redes ad hoc, Inundação, Filtros de Bloom

1. INTRODUÇÃO

Nas redes *ad hoc*, por suas características de topologia dinâmica, escassez de recursos e ausência de infra-estrutura, o processo de inundação se torna um mecanismo fundamental, por permitir a coleta, distribuição ou disseminação de informação na rede, sendo largamente utilizado em protocolos de roteamento [10, 25], descoberta de serviços [19, 15], e para disseminação ou busca de informação de uma forma geral [16]. Entretanto, um dos problemas introduzidos pela inundação é a chamada tempestade de *broadcast* [23], que implica em transmissões redundantes, alto nível de disputa no meio e aumento de colisões, consumindo desta forma os já escassos recursos disponíveis.

Para controlar estes efeitos nocivos da inundação, diversas soluções foram propostas na literatura [14, 18, 23, 9, 11]. Estas soluções podem ser agrupadas basicamente em probabilísticas e determinísticas. Os mecanismos determinísticos dependem do conhecimento da topologia existente, pelo menos de forma limitada, por exemplo da vizinhança de dois saltos. Entretanto, por apresentarem dificuldade na obtenção deste conhecimento, acabam recaindo em algum tipo de inundação ou troca adicional de mensagens, aumentando a sobrecarga de controle.

Por outro lado, os mecanismos probabilísticos não dependem do conhecimento da topologia, mas apresentam limitações de desempenho quando aplicados a redes que não correspondem às hipóteses probabilísticas iniciais do mecanismo [27], por exemplo, quando são aplicados em redes com distribuição não uniforme dos nós.

Assim sendo, a proposta descrita neste artigo, chamada FloorB, inicialmente apresentada em [1], baseia-se em um mecanismo híbrido, que assume um conhecimento da topologia em até dois saltos, de forma resumida, 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 é avaliado por simulação, comparativamente a outros mecanismos, tais como inundação cega (*Blind*), inundação probabilística simples (Gossip), Gossip Adaptativo e MPR, 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* e mobilidade do tipo social.

Para apresentar a proposta, o restante do texto está organizado da seguinte maneira. Na Seção 2, são apresentados os trabalhos relacionados. A Seção 3 descreve a proposta e alguns conceitos utilizados na sua elaboração, tais como a avaliação de vizinhança de 1 e 2 saltos e os conceitos de filtros de Bloom, e no final descreve o algoritmo utilizado. A Seção 4 enumera as premissas usadas no desenvolvimento do simulador próprio, utilizado nas primeiras avaliações, descreve os cenários de simulação e discorre sobre os resultados obtidos, e finalmente apresenta os resultados obtidos a partir da implementação no simulador ns-2 [24]. A Seção 5 traz as conclusões, indicando as principais contribuições deste artigo, e sugere alguns trabalhos futuros.

2. TRABALHOS RELACIONADOS

A solução ótima para o problema de inundação eficiente a partir de um nó é calcular a MLST - *Maximum Leaf Spanning Tree* - do grafo, escolhendo como raiz 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 será mínimo.

Entretanto, esta solução não é prática, uma vez que o problema é NP-difícil [13], e portanto o algoritmo possui alta complexidade, além de ser de difícil implementação de forma distribuída. Por exemplo, o algoritmo proposto em [13] para obtenção da MLST de um grafo tem como passo inicial a realização de uma busca em largura, que de forma distribuída deve ser implementada como uma inundação cega. Além disto, a MLST deveria ser calculada a cada instante que seja realizada a inundação, por conta da mobilidade, e para cada fonte que inicie a inundação.

Para solucionar 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 [23], os efeitos da tempestade de *broadcast* e alguns esquemas para controle de inundação, classificados em probabilísticos, baseado em contagem, baseado em distância, baseado na localização e baseado em agrupamento, são avaliados. Os resultados por simulação apresentados 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 [14], são apresentados detalhadamente, em diversos cenários, os conceitos do reencaminhamento probabilístico, chamado Gossip, e outros esquemas auxiliares para melhoria das taxas de entrega. O funcionamento bá-

sico 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 [27], 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 controle de inundação, tais como métodos determinísticos.

O mecanismo RAPID, proposto em [11], apresenta um esquema conjugado de Gossip Adaptativo com um esquema semelhante ao baseado em contagem de [23]. No Gossip Adaptativo, a probabilidade de reencaminhamento é calculada como função do número de vizinhos através da fórmula $\frac{\beta}{N}$, onde N é o número de vizinhos e β é um parâmetro do mecanismo. O aprendizado de vizinhança ocorre pela troca de mensagens de *hello* simples. 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 [9] propõe um Gossip Adaptativo baseado no conhecimento de 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*, entretanto, provoca o crescimento da mensagem de inundação, e também toma decisões probabilísticas baseadas somente na comparação das vizinhanças do emissor e receptor, não atuando adequadamente em redes com características não homogêneas.

Outra alternativa para disseminar informação em redes *ad hoc* é utilizar redes sobrepostas [7]. 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 e outros [3, 2, 26, 17]. 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 [18] e avaliada em alguns trabalhos [5, 6, 22]. Esta talvez seja a solução mais adotada para o controle de inundação, sendo a alternativa utilizada no protocolo de roteamento para redes *ad hoc* OLSR [10].

Tanto os resultados obtidos neste trabalho, quanto os trabalhos citados, indicam que esta solução apresenta altas taxas de reencaminhamentos evitados, adaptando-se

bem as 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:

Algoritmo de seleção de MPR

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

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. 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 Δ é o grau máximo na rede.

3. FLOORB - REPRESENTAÇÃO DE VIZINHANÇA POR FILTROS DE BLOOM

Nesta seção, é apresentada a idéia básica e os conceitos envolvidos na proposta. O mecanismo, denominado FloorB (*Flooding control through neighborhood Representation by Bloom filters*), utiliza um conhecimento limitado da vizinhança de dois saltos e um método probabilístico para o encaminhamento das mensagens.

3.1. VIZINHANÇA DE 1 E 2 SALTOS E REENCAMINHAMENTO DE MENSAGEM

Um dos conceitos adotados na proposta é o de obter um conhecimento limitado sobre a vizinhança de 1 e 2 saltos do nó receptor, e sobre a vizinhança de 1 salto do nó emissor. Algumas premissas são utilizadas nesta avaliação, que são posteriormente 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 1.

Seja $N^h(x)$ o conjunto de nós que distam h saltos do nó x , pode-se estabelecer que o conjunto de vizinhos diretos de e é $N^1(e)$ (Figura 2(a)), que possui $|N^1(e)|$ elementos. Analogamente, o conjunto de vizinhos de r é $N^1(r)$ (Figura 2(b)). Porém, o conjunto de nós que

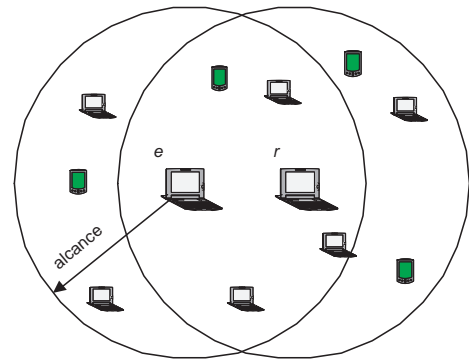


Figura 1. Comunicação entre dois nós

não recebem diretamente uma mensagem enviada por e , mas podem recebê-la caso r a reencaminhe, é dado por $M = N^1(r) - (N^1(e) \cap N^1(r))$ (Figura 2(d)). A partir das premissas iniciais, pode-se afirmar que as cardinalidades destes conjuntos são proporcionais às áreas cobertas, conforme descrito em [9].

Com estas considerações, a simples comparação entre a lista de vizinhos do emissor e a lista de vizinhos do receptor fornece uma 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 do 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 $N^1(e) \cap N^1(r)$ (Figura 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 $N^1(r)$ estabelece o conjunto união dos vizinhos de 1 salto dos nós pertencentes a M como $U = \{\bigcup N^1(x) \mid x \in M\}$, representado na Figura 2(e).

Realizando a intersecção deste conjunto U com os vizinhos de e obtém-se o conjunto $I = U \cap N^1(e)$, 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, podemos ver que para os nós receptores posicionados no limite do alcance de transmissão do emissor, as cardinalidades destes conjuntos I e M são muito próximas, 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,

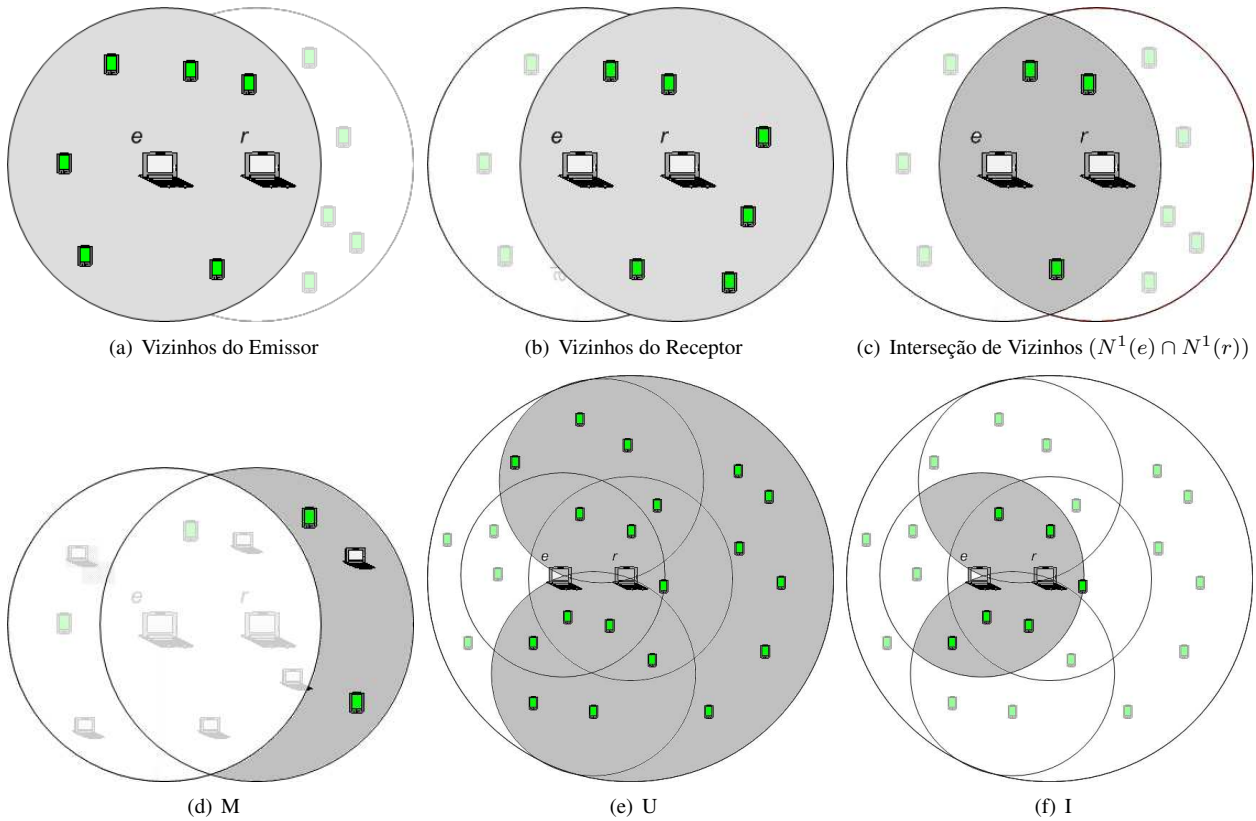


Figura 2. Conjuntos de Vizinhança

onde a probabilidade de transmissão para receptores que possuem vizinhança muito similar a do transmissor, ou seja $|M| \simeq 0$ e $I \simeq N^1(e)$, deve ser aproximadamente nula e a probabilidade de transmissão para receptores no limite de alcance, ou seja $|M| \simeq |I|$, deve ser aproximadamente 1, e portanto foi utilizada a Equação 1.

$$p = \frac{|M|}{|I|} \quad (1)$$

Para os casos onde $|I| = 0$ ou que esta razão é superior a 1, utiliza-se $p = 1$. Já nos casos onde $|M| = 0$, adota-se, de forma conservadora, a Equação 2.

$$p = \frac{0.333}{(|N^1(e) \cap N^1(r)|)^2} \quad (2)$$

Esta Equação 2 é aplicada no caso onde não existem novos nós que recebam a mensagem, caso ela seja reencaminhada, e portanto deveria-se esperar uma probabilidade nula. Entretanto, a equação apresenta um decaimento mais agressivo do que o adotado no Gossip Adaptativo, ao utilizar o quadrado do número de nós na interseção, e também ao adotar um valor de probabilidade inicial menor que 1 ($\beta < 1$). Com a adoção desta segunda equação, pretende-se ter uma probabilidade residual de reencaminhamento para aqueles nós que não possuem vizinhos não

cobertos. Este procedimento pode reduzir a economia obtida pelo mecanismo, porém permitindo a ocorrência de reencaminhamentos em situações onde ainda não há um aprendizado adequado da vizinhança, tornando o mecanismo mais ágil e robusto.

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. Neste caso, o nó b possui um grande número de vizinhos, o que levaria a uma baixa probabilidade de reencaminhar a mensagem no Gossip Adaptativo.

No entanto, os conjuntos U e $N^1(e)$ são disjuntos, e portanto obtém-se $p = 1$, conforme desejado. Por raciocínio semelhante, e pela observação de outros cenários, é possível concluir que a obtenção de p será sempre adequada, mesmo quando ocorre variação de densidade, topologia específica, redes com alta ou baixa densidade, ou transmissões não-omnidirecionais ou com obstáculos, conforme observado nos resultados apresentados na Seção 4.

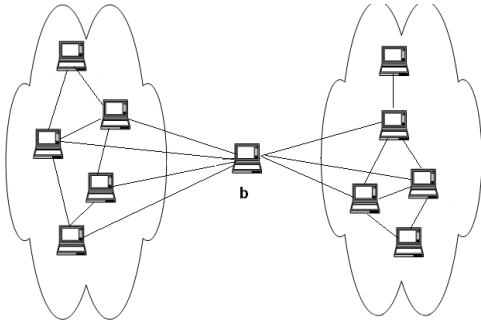


Figura 3. Topologia com corte em um nó

Um questionamento que pode ser feito quanto a este 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 da troca periódica de mensagens de *hello*. Para se obter a lista de vizinhos de dois saltos ($N^2(x)$) 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, de banda da rede e computacionais, o FloorB realiza uma aplicação de Filtros de Bloom, que são descritos na Seção 3.2.

3.2. FILTROS DE BLOOM

Filtros de Bloom [4] 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 [20, 12].

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 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 [20], 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 pela Equação 3, apresentada na Figura 5, para valores de m iguais a 128 e 256, e para k igual a 1, 2 ou 3, com número de elementos (vizinhos) variando de 0 a 200.

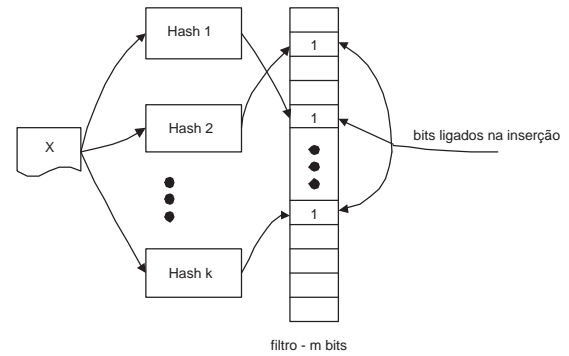


Figura 4. Funcionamento do filtro de Bloom

Nesta figura, podemos observar que os filtros de Bloom com estas configurações apresentam uma probabilidade de falso positivo relativamente baixa 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, mas ao custo de probabilidades mais elevadas quando há aumento deste número. Entretanto, estes falsos positivos, quando limitados a baixos valores, não são significativos 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.

$$P_{false} = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k \quad (3)$$

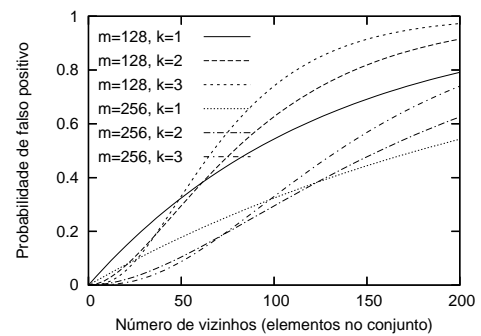


Figura 5. Probabilidade de falso positivo

Também segundo o trabalho em [20], o número de elementos de um conjunto $|S|$ pode ser estimado, a partir do filtro correspondente, contando-se o número de bits em 0 ($\#bits_0$) no filtro e aplicando a Equação 4.

$$|S| = \frac{\log\left(\frac{\#bits_0}{m}\right)}{k \log\left(1 - \frac{1}{m}\right)} \quad (4)$$

Outra propriedade importante dos filtros de Bloom é a facilidade para se obter o filtro correspondente à união ou intersecção 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. Analogamente, a intersecção é obtida através da operação **E (AND)**. Estas propriedades serão utilizadas pelo FloorB para computar os conjuntos U e I , apresentados na Seção 3.1.

3.3. FLOORB - ALGORITMO

O algoritmo adotado pelo FloorB para executar as tarefas descritas na Seção 3.1, utilizando os conceitos de filtro de Bloom descritos na Seção 3.2, é apresentado a seguir.

Algoritmo do FloorB

Notação:

e = nó emissor; r = nó receptor;
 $filtro(A)$ = filtro de Bloom do conjunto A ;
 $N^h(x)$ = conjunto de nós que distam h saltos do nó x ;
 $M = N^1(r) - (N^1(e) \cap N^1(r))$;
 $U = \{\bigcup N^1(x) | x \in M\}$;
 $I = U \cap N^1(e)$;

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** $v \in N^1(r)$, faça:
- 4: **Se** e não é vizinho de v (bits correspondentes não ligados em $filtro(N^1(v))$), 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 $filtro(N^1(m))$
- 10: **fim Para cada**
- 11: $filtro(I) = filtro(U)$ E $filtro(N^1(e))$
- 12: estimar $|M|$ e $|I|$ (equação 4)
- 13: calcular p de reencaminhamento (equação 1 ou equação 2)
- 14: realizar reencaminhamento com probabilidade p

O passo 1 deste algoritmo corresponde à troca periódica de mensagens de *hello*, onde cada nó x passa a informação $filtro(N^1(x))$ para seus vizinhos. O conjunto $N^1(x)$ é construído localmente pela recepção dos mesmos *hellos*. Em seguida, todas as informações descritas na Seção 3.1 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 populado o conjunto M (passo 5). Com este conjunto calculado, são obtidos os filtros correspondentes 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 um 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 $O(\Delta)$.

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 responsável pela interligação de duas partições da rede, como a topologia apresentada na Figura 3.

4. AVALIAÇÃO

Esta seção enumera as premissas usadas no desenvolvimento do simulador próprio, descreve os cenários utilizados e avalia os resultados obtidos.

4.1. SIMULADOR

Para avaliar o controle de inundação realizado pela proposta, foi implementado um simulador em linguagem *tcl*, sem a adoção de uma camada de enlace específica, não existindo portanto contenção ou colisão. 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*. O simulador é modular, permitindo o acréscimo de diversos mecanismos de inundação, e 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 .
- 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*.
- 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.
- FloorB - implementação da proposta descrita na Seção 3.

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 [14], 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 a equação utilizada para cálculo da probabilidade, fazendo-a inversamente proporcional ao número de vizinhos, ocorre que a probabilidade de que nenhum do nós pertencentes a $N^1(e)$ reencaminhe a mensagem será dada pela Equação 5, uma vez que em redes com distribuição uniforme os valores de $N^1(e)$ e $N^1(r)$ serão aproximadamente iguais.

$$p_{not} = \left(1 - \frac{\beta}{|N^1(r)|}\right)^{|N^1(e)|} \quad (5)$$

A Equação 5 possui limite igual a $e^{-\beta}$ quando o número de vizinhos tende a infinito, fazendo que com probabilidade $e^{-\beta}$ a mensagem já deixe de ser encaminhada logo no primeiro salto, restringindo o desempenho deste mecanismo para redes muito densas. É importante notar que esta Equação 5 possui uma convergência muito rápida para os limites, e por exemplo, com 20 vizinhos o valor de p_{not} , com $\beta = 6$, é aproximadamente de 0,08%, enquanto que com $\beta = 2,5$ é de 6,9%.

Os parâmetros do filtro de Bloom do mecanismo FloorB também foram através de simulações iniciais, e pelo estudo da Figura 5. Foram utilizados filtros com 256 bits ($m = 256$) e 2 funções *hash* ($k = 2$), por apresentarem baixas probabilidades de falso positivos para redes com poucos nós, e também não possuir um tamanho muito elevado para ser colocado nas mensagens.

Para avaliar os mecanismos propostos 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 [24], com nós distribuídos aleatoriamente numa área e com movimentação do tipo *random waypoint*. Foram utilizados os seguintes parâmetros: área de $800m \times 800m$, 300 nós, alcance de $80m$, pausa de $10s$, e velocidade mínima de $0m/s$.
- 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, 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 MPR real, foram implementados no simulador ns-2 [24], permitindo realizar uma avaliação quando os mecanismos são aplicados sobre uma camada de enlace com características reais, no caso IEEE 802.11. Uma diferença importante na implementação do mecanismo MPR no ns-2 ocorre no processo de seleção dos MPRs. No simulador próprio, esta seleção é definida no momento de envio da mensagem de inundação, enquanto a implementação realizada no ns-2 é mais próxima do mecanismo de MPR do OLSR, onde a mensagem de *hello* informa o conjunto de MPRs selecionados.

4.2. RESULTADOS

As duas medidas de desempenho utilizadas para comparação dos mecanismos foram a taxa de entrega, ou alcançabilidade, expressa pela Equação 6 e a taxa de reencaminhamentos evitados, expressa pela Equação 7.

$$tx \text{ entrega} = \frac{\# \text{ msg recebidas}}{\# \text{ nós na rede} \times \# \text{ msgs enviadas}} \quad (6)$$

$$reenc \text{ evitados} = 1 - \frac{\# \text{ msg reencaminhadas}}{\# \text{ msg recebidas}} \quad (7)$$

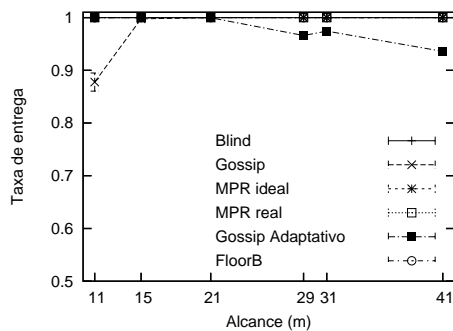
Além destas medidas, em alguns cenários também foi avaliada a distribuição do número de nós com as diversas taxas de reencaminhamentos realizados, de 0 a 1, em

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 possuam a mesma taxa de reencaminhamento.

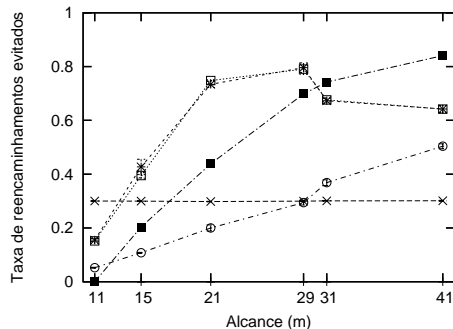
Este fator é importante por duas considerações: mecanismo que sobrecarregue alguns nós com muitos reencaminhamentos estará sujeito a problemas de desempenho no caso de falha deste nó; ao concentrar os reencaminhamentos em alguns nós, caso exista limitação quanto a energia, comum em redes *ad hoc*, mais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

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 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 as mensagens de *hello* enviadas. Esta é uma forma indireta de avaliar a robustez dos mecanismos frente à mobilidade.



(a) Taxa de Entrega



(b) Reencaminhamentos Evitados

Figura 6. Cenário em grade, sem perdas

O primeiro cenário avaliado foi o de rede em grade. Ele é composto de 400 nós em uma grade de 20×20 , 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 6(a).

Este é um cenário estático e com distribuição uniforme dos nós, e 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 [27].

Já o Gossip Adaptativo apresenta o resultado esperado de redução da taxa de entrega quando a rede se torna densa. Isto é devido a Equação 5, conforme já citado.

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.

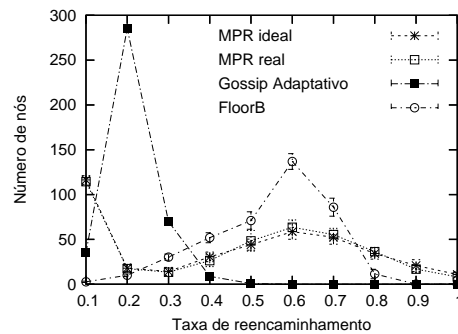


Figura 7. Cenário em grade, sem perdas

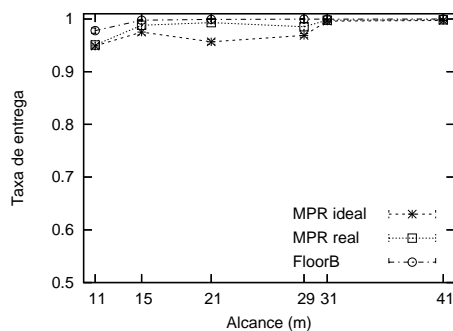
Na Figura 6(b), são mostrados os reencaminhamentos evitados para cada mecanismo. Pode-se notar que para esta medida o Blind e o Gossip têm desempenho constante, equivalente a $1 - p$, a probabilidade de não reencaminhar, que no Blind é igual a 0. Os mecanismos de MPR, tanto ideal quanto real, apresentaram 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.

Por este gráfico pode-se 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.

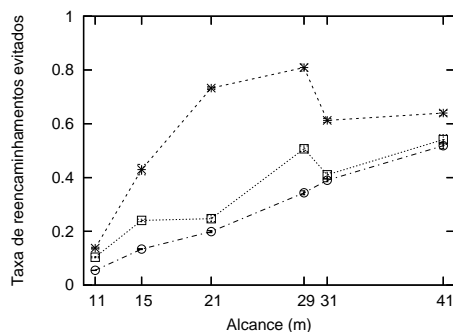
Além disto, na Figura 7 observa-se a distribuição da

quantidade de nós em função das taxas de reencaminhamentos realizados, 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 façam mais de 90% de reencaminhamentos por mensagem 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 50% e 90%, respectivamente. Este resultado indica que nestes mecanismos não haverá nós sobrecarregados com a tarefa de reencaminhar quase todas as inundações, enquanto outros permanecem ociosos. Uma grande vantagem deste comportamento, conforme já citado, é a sobrevivência da rede, uma vez que os nós com taxas de reencaminhamento muito alta podem esgotar rapidamente sua energia.



(a) Taxa de Entrega



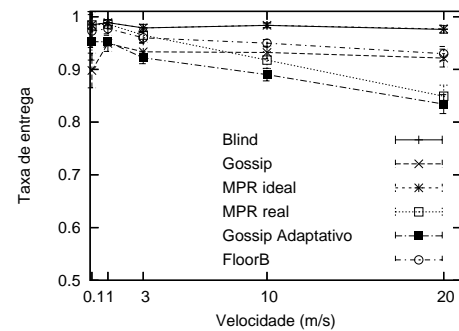
(b) Reencaminhamentos Evitados

Figura 8. Cenário em grade, perdas=0,2

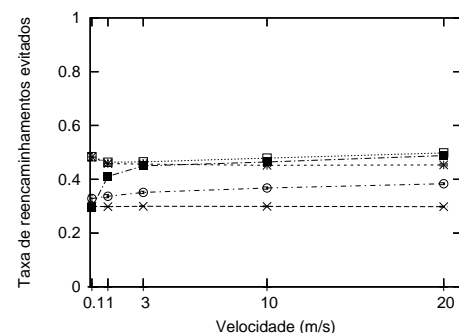
Outro fator é que ao concentrar o tráfego em poucos nós, ao ocorrerem perdas, elas se tornam mais graves, afetando a taxa de entrega, como observado na Figura 8, onde é utilizada taxa de perda na recepção de 0,2. Nota-

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.

O segundo tipo de cenário avaliado foi o cenário com posicionamento e mobilidade aleatórios. 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 9 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.



(a) Taxa de Entrega



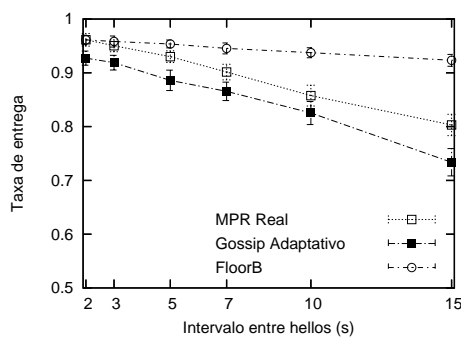
(b) Reencaminhamentos Evitados

Figura 9. Cenário com mobilidade Random Waypoint, sem perdas

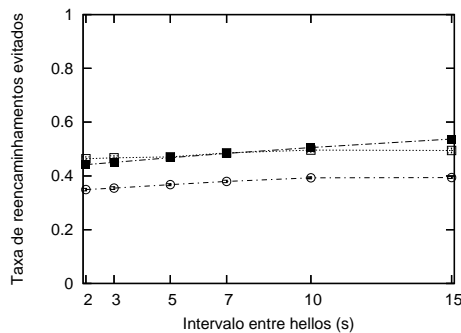
Quando um novo vizinho entra na área de alcance, basta o recebimento de um *hello* para que o mesmo passe a fazer parte da lista de vizinhos. Entretanto, quando um nó sai do alcance, somente após a perda de um número consecutivo de *hellos* ele sairá da lista. Nestas simulações, foram utilizados intervalos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



(a) Taxa de Entrega



(b) Reencaminhamentos Evitados

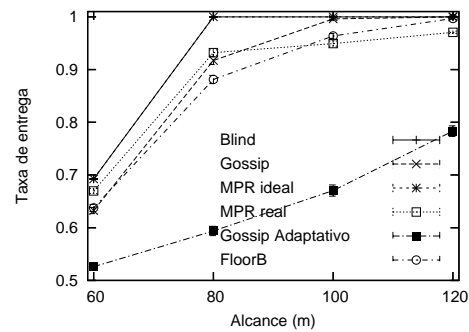
Figura 10. Cenário com mobilidade Random Waypoint, variação do intervalo entre *hellos*

Para comprovar esta hipótese, 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 10. 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 de velocidade, como pode ter seu número de mensagens de controle reduzido ao se adotar intervalos maiores no envio dos *hellos*.

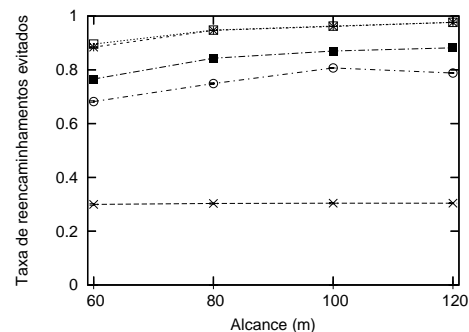
Já quanto aos reencaminhamentos evitados, nota-se

nas Figuras 9(b) e 10(b) que os desempenhos de todos mecanismos são bem semelhantes, com o FloorB apresentando resultado um pouco inferior. 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].



(a) Taxa de Entrega

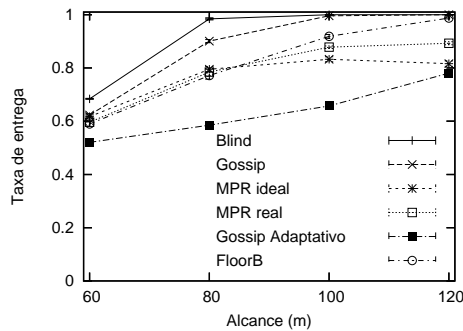


(b) Reencaminhamentos Evitados

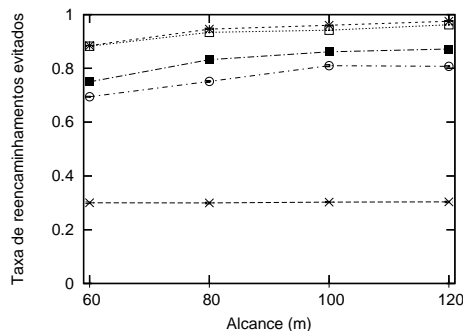
Figura 11. Cenário com mobilidade Social (Comunidade), sem perdas

Nestes cenários, existe uma variação muito grande do número de vizinhos dos nós, de forma semelhante ao apresentado na Figura 3, e por este motivo os mecanismos tipo Gossip Adaptativo não obtêm altas taxas de entrega. Conforme a Figura 11, quando não há perdas, as taxas de entrega dos mecanismos MPRs ideal e real e FloorB são semelhantes, 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 na Figura 12, o desempenho do FloorB é superior 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



(a) Taxa de Entrega



(b) Reencaminhamentos Evitados

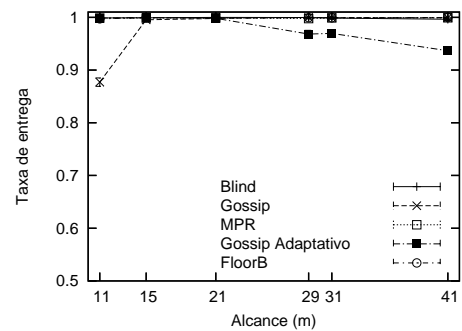
Figura 12. Cenário com mobilidade Social (Comunidade), perda=0,2

real. Isto se deve a escolha mais restrita dos MPRs realizada pelo MPR ideal, que não depende das mensagens de *hello*. Ao ter, por exemplo, um único MPR realizando a ligação dos dois grupos representados pela rede social, as falhas de recepção deste nó representam um impacto maior nas taxas de entrega.

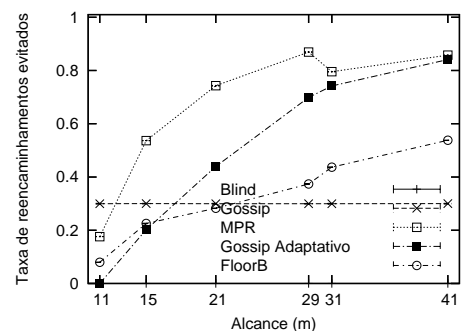
Como continuidade da avaliação, os mesmos mecanismos também foram implementados no ns-2 e avaliados com os mesmos cenários em grade e aleatório móvel, utilizando camada de enlace IEEE802.11, com taxa de transmissão de 11 Mbps. Nesta avaliação, não foram introduzidas perdas, entretanto, elas ocorrem naturalmente devido a interferências e colisões, que se tornam mais intensas em ambientes com vizinhança elevada [8].

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.

Com esta implementação no ns-2, pode-se observar na Figura 13 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 6, tendo as mesmas tendências. A principal diferença ocorre pelo fato dos mecanismos estarem submetidos a perdas e atrasos nas mensagens.

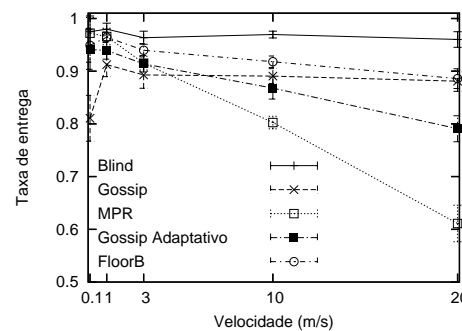


(a) Taxa de Entrega

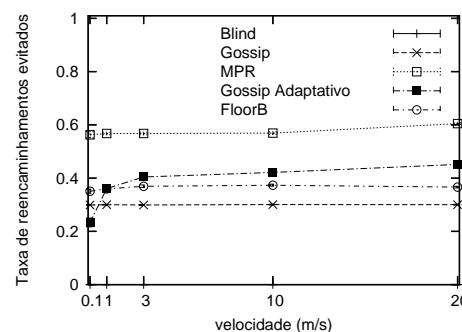


(b) Reencaminhamentos Evitados

Figura 13. Cenário em grade, ns-2



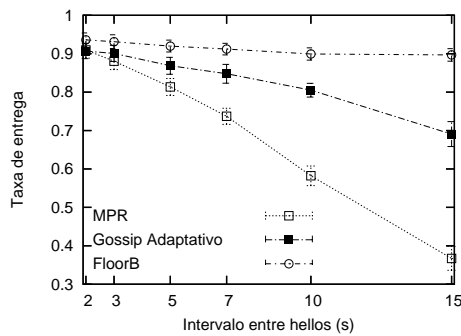
(a) Taxa de Entrega



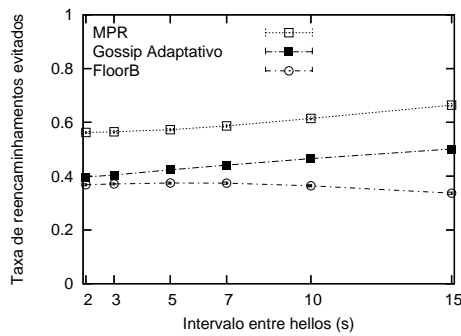
(b) Reencaminhamentos Evitados

Figura 14. Cenário com mobilidade Random Waypoint, ns-2

A Figura 14 apresenta o resultado das simulações no ns-2 com cenário móvel aleatório. Observa-se que a taxa de entrega apresentou a mesma tendência obtida com o simulador próprio (Figura 9), entretanto, com degradação mais acentuada dos mecanismos MPR e Gossip Adaptativo com o aumento da mobilidade.



(a) Taxa de Entrega



(b) Reencaminhamentos Evitados

Figura 15. Cenário com mobilidade Random Waypoint, variação do intervalo entre *hellos*, ns-2

Pelo mesmo motivo anterior, a avaliação com alteração entre 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 pode-se observar na Figura 15. 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.

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 e portanto não permite avaliar situações de carga na rede. Para isto, foi adicionado aos cenários aleatórios móveis 16 pares de comunicação do tipo fonte-destino, UDP/CBR, dispostos uniformemente na rede conforme a Figura 16. Estes 32 nós não participam das tarefas de inundação, não emitindo, recebendo

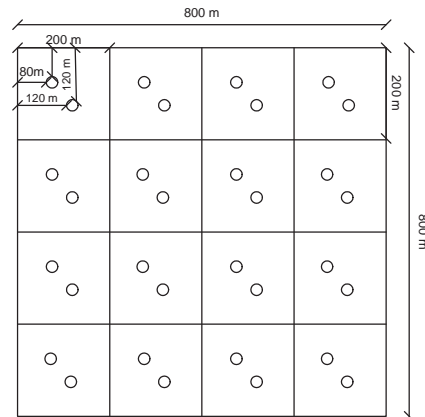
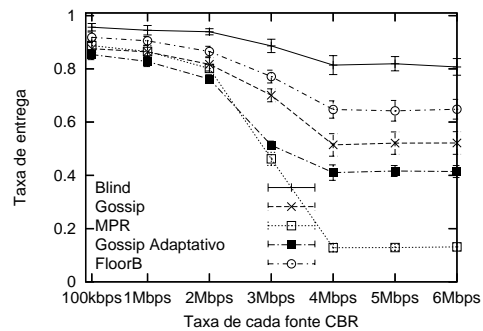
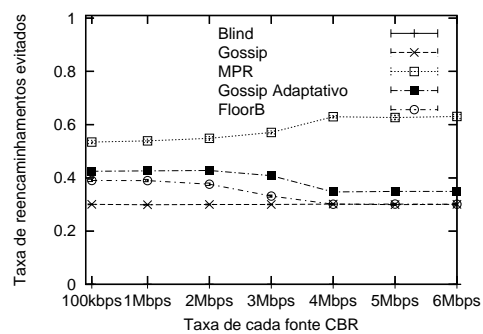


Figura 16. Disposição dos nós geradores de carga

ou reencaminhando mensagens, só atuando na geração de carga. Da forma como foram dispostos, permitem provocar contenção em toda a área da rede, concorrendo portanto com todos os *hellos* e mensagens de inundação enviados ou reencaminhados.



(a) Taxa de Entrega



(b) Reencaminhamentos Evitados

Figura 17. Cenário com mobilidade Random Waypoint, variação do intervalo entre *hellos*, ns-2

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 17, desde um valor inicial de 100kbps até 6Mbps correspondentes à sa-

turação da rede.

Pode-se observar pela Figura 17(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*.

5. CONCLUSÃO

Uma importante contribuição deste trabalho 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. No entendimento dos autores, 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 controle 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 a mobilidade quanto a perdas de mensagens.
- Possibilidade de redução no número de mensagens de controle ao 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. Por outro lado, ainda se encontra em estudo o acréscimo de um mecanismo de contagem de mensagens, semelhante ao adotado em [11], com o intuito de melhorar o desempenho em termos de economia, sem alterar a taxa de entrega do FloorB.

Outra importante contribuição deste trabalho é o desenvolvimento de um simulador genérico, no qual outros

mecanismos podem ser acrescentados. Como trabalho futuro, está prevista a inclusão do mecanismo sugerido em [9]. Também foi realizada a conversão do código para o ns-2, permitindo validar os resultados já obtidos, e verificar o desempenho dos mecanismos quando utilizados sobre uma camada de enlace tal como a do IEEE802.11.

Como trabalho futuro, estuda-se a implementação de um protocolo de roteamento genérico, do tipo estado de enlace, e aplicar todos os mecanismos implementados para a execução da inundação das mensagens de tal protocolo, avaliando os efeitos provocados pelos desempenhos dos mecanismos às rotas calculadas pelo protocolo de roteamento.

Referências

- [1] Carlos Henrique Pereira Augusto and José Ferreira de Rezende. Inundação probabilística em redes ad hoc utilizando sumário de vizinhança através de filtros de bloom. In *XXVII Simpósio Brasileiro de Redes de Computadores - SBRC2009*, May 2009.
- [2] Stefano Basagni, Michele Mastrogiovanni, Alessandro Panconesi, and Chiara Petrioli. Localized protocols for ad hoc clustering and backbone formation: A performance comparison. In *Parallel and Distributed Systems, IEEE Transactions on*, volume 17, pages 292–306, April 2006.
- [3] Stefano Basagni, Michele Mastrogiovanni, and Chiara Petrioli. 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*, pages 70–79, October 2004.
- [4] Burton H Bloom. Spacetime trade-offs in hash coding with allowable errors. In *Communications of the ACM*, volume 13, pages 422–426, New York, NY, USA, 1970. ACM.
- [5] A. Busson, N. Mitton, and E. Fleury. An analysis of the MPR selection in OLSR. In *Spatial Stochastic Modeling of Wireless Networks, SpasWIN 05*, Trento, Italy, April 2005.
- [6] A. Busson, N. Mitton, and E. Fleury. 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.
- [7] Sandrine Calomme and Guy Leduc. Efficient and resilient overlay topologies over ad hoc networks. In *International Workshop on Self-Organizing Systems (IWSOS 2007)*, pages 44–58, September 2007.

- [8] Kleber Vieira Cardoso and José Ferreira de Rezende. Adaptação automática de taxa em redes 802.11 densas. In *XXVI Simpósio Brasileiro de Redes de Computadores - SBRC2008*, May 2008.
- [9] Julien Cartigny and David Simplot. Border node retransmission based probabilistic broadcast protocols in ad-hoc networks. In *Telecommunication Systems*, volume 22, pages 189–204. Kluwer Academic Publishers, January 2003.
- [10] T. Clausen and P. Jacquet. RFC 3626 - Optimized Link State Routing protocol OLSR. *IETF Request for Comments*, October 2003.
- [11] Vadim Drabkin, Roy Friedman, Gabriel Kliot, and Marc Segal. Rapid: Reliable probabilistic dissemination in wireless ad-hoc networks. In *SRDS '07: Proceedings of the 26th IEEE International Symposium on Reliable Distributed Systems*, pages 13–22, October 2007.
- [12] Natalia Castro Fernandes, Marcelo Duffles Donato Moreira, and Otto Carlos Muniz Bandeira Duarte. An efficient filter-based addressing protocol for autoconfiguration of mobile ad hoc networks. In *IEEE INFOCOM 2009*, April 2009.
- [13] Tetsuya Fujie. An exact algorithm for the Maximum Leaf Spanning Tree problem. *Computers and Operations Research*, 30(13):1931 – 1944, 2003.
- [14] Zygmunt Haas, Joseph Y. Halpern, and Li Li. Gossip-based ad hoc routing. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications*, volume 3, pages 1707 – 1716, June 2002.
- [15] Sumi Helal, Nitin Desai, Varun Verma, and Choonhwa Lee. Konark - a service discovery and delivery protocol for ad-hoc networks. In *Third IEEE Conference on Wireless Communications and Networking (WCNC)*, volume 3, pages 2107–2113, March 2003.
- [16] Gautham Karumanchi, Srinivasan Muralidharan, and Ravi Prakash. Information dissemination in partitionable mobile ad hoc networks. In *Proceedings of IEEE Symposium on Reliable Distributed Systems (SRDS)*, pages 4–13, October 1999.
- [17] Ulas C. Kozat, Leandros Tassioulas, George Kondylis, Bo Ryu, and Mahesh K. Marina. Virtual dynamic backbone for mobile ad hoc networks. In *IEEE International Conference on Communications, 2001. ICC 2001.*, volume 1, pages 250–255, June 2001.
- [18] Anis Laouiti, Amir Qayyum, and Laurent Viennot. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. In *35th Annual Hawaii International Conference on System Sciences (HICSS'2002)*, January 2002.
- [19] Vincent Lenders, Martin May, and Bernhard Plattner. 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)*, pages 120–130, June 2005.
- [20] Michael Mitzenmacher and Andrei Broder. Network applications of bloom filters: (a survey). In *Internet Mathematics*, volume 1, pages 485–509, 2004.
- [21] Mirco Musolesi and Cecilia Mascolo. 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] Dang Nguyen and Pascale Minet. Analysis of MPR selection in the OLSR protocol. In *AINAW '07: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops*, pages 887–892, 2007.
- [23] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. 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*, pages 151–162, New York, NY, USA, 1999. ACM.
- [24] NS-2. The network simulator - ns-2, 1995. <http://www.isi.edu/nsnam/ns/> - último acesso em 17/12/2008.
- [25] C. Perkins, E. Belding-Royer, and S. Das. RFC 3561 - Ad hoc On-demand Distance Vector AODV routing. *IETF Request for Comments*, July 2003.
- [26] Peng-Jun Wan, Khaled M. Alzoubi, and Ophir Frieder. 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*, volume 3, pages 1597–1604, June 2002.
- [27] M. Bani Yassein, M. Ould-Khaoua, and S. Papanastasiou. On the performance of probabilistic flooding in mobile ad hoc networks. *ICPADS, 11th International Conference on Parallel and Distributed Systems*, 02:125–129, 2005.