

Escalonamento Distribuído de Inundações em Redes Ad Hoc Móveis*

Carlos Henrique Pereira Augusto, José Ferreira de Rezende

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. *Flooding of data and control messages is a widely used mechanism in network applications and protocols, which can have latency requirements for the information delivery. However, this Minimum-Latency Broadcast Scheduling problem is NP-Complete, thus some papers propose approximation algorithms. This work presents simulation results of well-known flooding control mechanism over IEEE 802.11 which shows that simple heuristics can achieve tolerable latencies. Finally, it proposes a distributed scheduling mechanism with partial knowledge of topology and its performance is measured combined with flooding control mechanism in CSMA/CA and TDMA networks. The proposal efficiency is proven by comparison to theoretical limits of centralized algorithms.*

Resumo. *A inundação de mensagens de controle e de dados é um mecanismo bastante empregado por protocolos e aplicações em redes ad hoc, que podem ter requisitos quanto à latência na entrega dessas informações. Entretanto, o problema de Escalonamento de Inundação com Mínima Latência é NP-Completo, levando a propostas de soluções através de algoritmos aproximados. Neste trabalho, são apresentados resultados de simulação de mecanismos convencionais de controle de inundação associados ao acesso ao meio CSMA/CA, indicando que heurísticas simples podem produzir resultados satisfatórios. Finalmente, é proposto um mecanismo de escalonamento distribuído com conhecimento parcial da topologia, o qual é avaliado em conjunto com os mecanismos de controle de inundação em redes CSMA/CA e TDMA, tendo a eficiência comprovada pela comparação com os limites teóricos indicados por algoritmos centralizados.*

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 e ausência de infraestrutura, estes mecanismos tornam-se essenciais, sendo largamente utilizados em protocolos de roteamento [Clausen and Jacquet 2003, Perkins et al. 2003], descoberta de serviços [Lenders et al. 2005, Augusto and de Rezende 2006], e para disseminação ou busca de informação de uma forma geral [Karumanchi et al. 1999].

A forma mais comum de se realizar uma inundação em rede *ad hoc* é chamada de inundação cega (*Blind*). Entretanto, um dos problemas introduzidos é a tempestade

*Este trabalho recebeu recursos da CAPES, CNPq, FAPERJ e FINEP.

de *broadcast* [Ni et al. 1999], que implica em um muitas transmissões redundantes. Se o acesso ao meio for por contenção, como no CSMA/CA, haverá um alto nível de disputa e aumento de colisões, consumindo os já escassos recursos e aumentando o tempo de entrega das mensagens. Sendo o acesso ao meio através de escalonamento no tempo, como no TDMA, poderá haver a necessidade de um grande número de *slots*.

O controle destes efeitos nocivos da tempestade de *broadcast* pode ser realizado através de duas abordagens distintas: através de um processo de seleção dos nós reencaminhadores da inundação, conforme algumas propostas [Haas et al. 2002, Laouiti et al. 2002, Ni et al. 1999, Drabkin et al. 2007, Augusto and de Rezende 2009], que chamaremos de controle de inundação; ou através da escolha da ordem em que os nós reencaminhadores irão realizar suas transmissões [Huang et al. 2007, Gandhi et al. 2008, Yeo et al. 2002, Vergados et al. 2009], chamados de escalonadores. Apesar de ortogonais, as duas abordagens possuem semelhanças, conforme mostrado nos resultados das Seções 3 e 5, pois os algoritmos escalonadores pressupõem implicitamente a escolha de somente alguns nós para o reencaminhamento das mensagens.

Estas duas abordagens possuem objetivos distintos, pois na primeira procura-se reduzir o número de mensagens reencaminhadas, e a solução ótima para este problema é a construção de uma 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 reencaminham as mensagens, quando o número de folhas é máximo, o número de mensagens é mínimo. Entretanto, a implementação desta solução não é prática, uma vez que o problema é NP-difícil [Fujie 2003].

O outro tipo de 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 [Gandhi et al. 2008], e a avaliação deste problema é o foco deste artigo.

Entretanto, uma observação importante é que uma escolha adequada dos nós reencaminhadores pode levar a uma latência da inundação com valores satisfatórios, mesmo não sendo este o problema alvo deste tipo de abordagem. Para mostrar isto, na parte inicial deste trabalho, através de simulação, é medida a latência da inundação quando utilizados mecanismos de controle de inundação associados ao escalonamento distribuído aleatório com contenção do acesso ao meio CSMA/CA do IEEE 802.11.

A partir dos resultados apresentados, são formuladas hipóteses, que fundamentam a proposta de um mecanismo de escalonamento distribuído, baseado em informações locais, com conhecimento limitado da topologia. Este mecanismo é avaliado juntamente com o CSMA/CA e também em uma rede com acesso ao meio do tipo TDMA, mas onde a decisão de utilização do *slot* é totalmente independente e local, implicando que no caso de escolhas incorretas de escalonamento ocorram colisões e perdas de transmissões.

O restante deste artigo está organizado da seguinte maneira. Na Seção 2 são apresentados os trabalhos relacionados. A Seção 3 descreve os resultados de latência em algumas propostas que realizam somente seleção de nós. A Seção 4 apresenta a proposta de escalonamento distribuído, e o algoritmo utilizado. A Seção 5 apresenta os resultados obtidos a partir da implementação da proposta no simulador ns-2 [NS-2 1995], compa-

rando com os valores teóricos de propostas centralizadas. A Seção 6 traz as conclusões, indicando as principais contribuições deste artigo, e sugere alguns trabalhos futuros.

2. Trabalhos Relacionados

O problema de Escalonamento de Inundação com Mínima Latência (MLBS) é um problema NP-Completo [Gandhi et al. 2008], e portanto os algoritmos que solucionam o problema são computacionalmente não-polinomiais. 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 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, a distância em saltos do nó mais distante em relação à fonte da inundação. Desta forma, 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 [Chlamtac and Kutten 1985] e $R\log^2 R/N$ em [Chlamtac and Weinstein 1991]. 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 através da atribuição de um *slot* para cada nó.

O problema MLBS pode ser definido como um problema de coloração de grafos [Huang et al. 2007], 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 [Badia et al. 2008].

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 limitação, quando há necessidade de uma latência limitada para a entrega da inundação, é conveniente a adoção de um acesso ao meio determinístico do tipo TDMA. Entretanto, se faz necessário o uso de um esquema, centralizado ou distribuído, para realizar o escalonamento dos nós. Neste contexto, alguns trabalhos propõem, através de diversas técnicas, mecanismos ou algoritmos com esta finalidade.

Em [Huang et al. 2007], o problema MLBS é tratado através de UDG, e são propostos 3 algoritmos centralizados aproximados chamados BBS, EBS e PBS (*Basic, Enhanced e Pipelined Broadcast Schedule*, respectivamente). Os três algoritmos iniciam a partir da criação da árvore geradora com raiz na fonte da inundação, derivada da busca em largura (BFS) no grafo da rede. Através desta busca 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.

Para o algoritmo BBS, os autores demonstram que o escalonamento está limitado em $24R - 23$ slots. No algoritmo EBS, que apresenta uma melhoria, produzindo não só um escalonamento menor como também uma menor redundância de transmissões, os autores demonstram que está limitado a $16R - 15$. Por último, no algoritmo PBS, que permite

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, é demonstrado um limite assintótico de $R + O(\log R)$ slots. Entretanto, durante a demonstração deste limite assintótico, pode-se observar que a latência é limitada a $R' + O(\log O(R'^3))$, onde R' é R ou $R - 1$. Portanto, com o intuito de comparar os limites de latência, aproximaremos para $R + 3\log R$ o limite deste algoritmo, por não considerar esta demonstração dentro do escopo deste trabalho.

Como exemplo de que o limite assintótico pode ser pouco representativo do limite superior do algoritmo, temos o trabalho em [Gandhi et al. 2008], que também propõe um algoritmo aproximado, com limite assintótico $O(1)$ vezes a latência ótima. O algoritmo também baseia-se na construção do MIS derivado da árvore geradora obtida da BFS no grafo. Apesar de definir uma versão distribuída do algoritmo, ela baseia-se no uso de buscas em largura e em profundidade, custosas em termos de mensagens. E o trabalho em [Huang et al. 2007] indica que a latência deste algoritmo é limitada a $648R$.

Em [Vergados et al. 2009], é apresentado um algoritmo que usa vetor de interferência, que permite identificar rapidamente se um nó interferirá com outro. 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.

[Zhang 2007] apresenta outra abordagem para o problema, através da aplicação de lógica *fuzzy* e redes neurais. Já em [Wu et al. 2005] 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.

Em relação aos mecanismos de controle de inundação, o Blind é a inundação simples, onde cada nó encaminha a mensagem na primeira vez que a recebe. Já no Gossip [Haas et al. 2002], cada nó encaminha a mensagem com uma probabilidade fixa p na primeira vez que a recebe. No Gossip Adaptativo o encaminhamento também ocorre na primeira vez que recebe a mensagem, mas com uma probabilidade p variável, por exemplo calculada como $\frac{\beta}{N}$ conforme proposto no mecanismo RAPID [Drabkin et al. 2007].

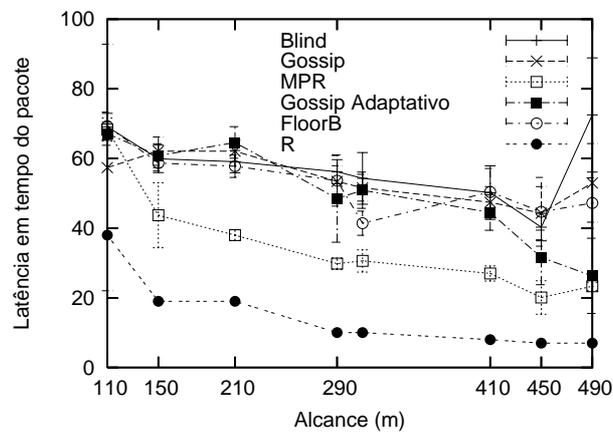
Uma alternativa muito usual é a adoção de MPRs (Multi Point Relays), proposto em [Laouiti et al. 2002]. Esta talvez seja a solução mais adotada para controle de inundação, sendo utilizada no protocolo de roteamento para redes *ad hoc* OLSR [Clausen and Jacquet 2003]. Ela é uma heurística baseada na informação de 2 saltos, obtida através de mensagens de *hello* com a lista de vizinhos. São escolhidos como MPRs os vizinhos de 1 salto que são únicos no alcance de vizinhos de 2 saltos, e em seguida os que permitem o alcance do maior número de vizinhos de 2 saltos.

Finalmente, FloorB, proposto em [Augusto and de Rezende 2009], baseia-se num mecanismo híbrido, que assume um conhecimento da topologia de 2 saltos de forma resumida e utiliza um método probabilístico para o encaminhamento das mensagens de inundação. Para reduzir a representação da vizinhança de 2 saltos, o mecanismo faz uso de filtros de Bloom. Tais filtros possuem propriedades que permitem reduzir a quantidade de informações trocadas entre vizinhos e facilitam a manipulação das informações coletadas.

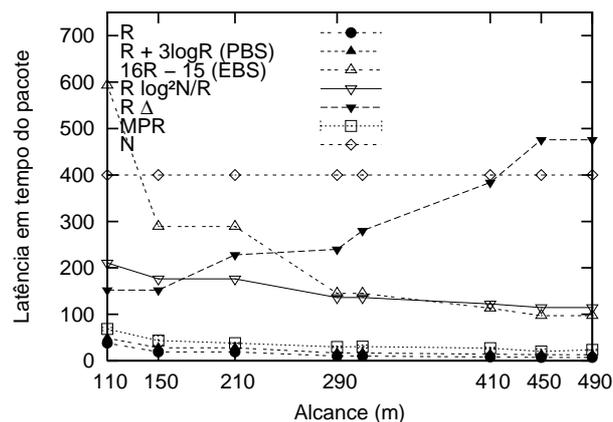
3. Avaliação de Mecanismos de Controle da Inundação

Em [Augusto and de Rezende 2009], é apresentada uma extensa avaliação dos mecanismos de controle de inundação Blind, Gossip, Gossip Adaptativo, MPR e FloorB, quanto à taxa de entrega e reencaminhamentos evitados. Estes mecanismos não realizam escalonamento, e portanto não visam reduzir a latência da inundação. Entretanto, nesta seção apresentamos uma nova avaliação dos mesmos quanto a latência, comparando os resultados com os limites teóricos de algoritmos aproximados e o limite inferior R .

Para esta avaliação, os mecanismos foram implementados no simulador ns-2, e utilizados em 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. Diversos graus de conectividade são obtidos alterando-se os alcances de 110 a 490m. Utilizando-se o padrão IEEE 802.11 com os valores *default* do ns-2, e com 10 rodadas e intervalo de confiança de 95%, obtém-se as latências apresentadas na Figura 1(a).



(a) Latência de mecanismos sem escalonamento



(b) Limites de latência de algoritmos aproximados

Figura 1. Latências de inundação

Por este gráfico, pode-se observar que o mecanismo Blind apresenta uma latência elevada para redes com maior grau. Isto 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.

Para efeitos de comparação, a Figura 1(b) apresenta os limites superiores de alguns algoritmos aproximados, incluindo os limites R e N , além dos resultados anteriores do MPR. Podemos notar que os limites apresentados pelos algoritmos são extremamente conservadores, estando mais próximos do limite superior do que do inferior, exceto pelo algoritmo PBS de [Huang et al. 2007]. Além disto, as latências fornecidas pelos mecanismos de controle de inundação, sem escalonamento, mostram-se pouco distante 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, portanto escalonáveis em menor tempo, que permitem a total cobertura dos vizinhos de 2 saltos, e este processo se reproduz salto-a-salto até cobrir toda a rede. Desta observação, é formulada a primeira hipótese para 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 4.

4. Mecanismo Proposto

O mecanismo proposto, chamado 2BSheld (*2-hop knowledge 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 Id.

A Figura 2 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. 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 nó v_2 é escalonado no *slot* $s + 1$, nó v_1 no *slot* $s + 2$ e o nó v_3 no *slot* $s + 3$.

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

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.

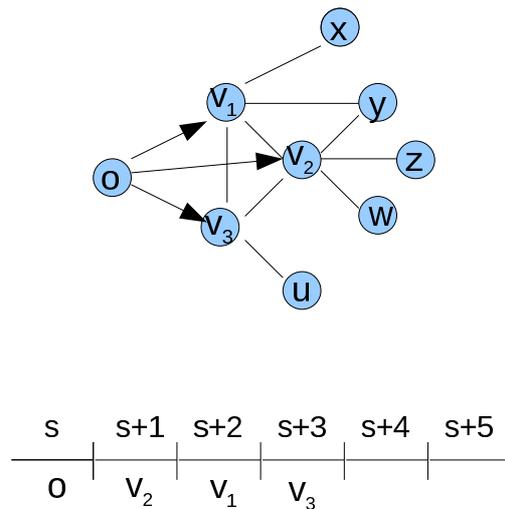


Figura 2. Exemplo de funcionamento do 2Bsheld

Algoritmo 1 2BSheld

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;

N_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** $v \in (N_1^r \cap N_1^e)$, faça:
 - 4: **Se** $|N_1^v - N_1^e| > |N_1^r - N_1^e|$
 - 5: $lowPri ++$
 - 6: **caso contrário**
 - 7: **Se** $|N_1^v - N_1^e| == |N_1^r - N_1^e|$ e $Id(v) < Id(r)$
 - 8: $lowId ++$
 - 9: **fim Se**
 - 10: **fim Para cada**
 - 11: **fim Para cada**
 - 12: $next = s(lowPri, lowId)$
 - 13: escalonar reencaminhamento da inundação em $s + next$
-

Um ponto em aberto neste algoritmo é a função $s(x, y)$. Em princípio, a única restrição é que seja crescente e apresente valores inteiros para x e y inteiros. Seguindo o conceito de desenvolvimento de um mecanismo simples, a primeira função avaliada é $s(x, y) = x + y + 1$. Na Seção 5 outras opções para $s(x, y)$ são avaliadas.

O mecanismo pode ser formalizado através de grafos e conjuntos, considerando a rede sem fio um grafo $G = (E, V)$, onde V é o conjunto de vértices, ou nós da rede, e E o conjunto de arestas, que representam os enlaces entre nós dentro do alcance de transmissão um do outro. 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, e é função do escalonador impedir a transmissão simultânea de nós interferentes entre si, e permitir a daqueles que não se interferem (Hipótese 3). Definimos como $N_1^{e_{i,s}}$ o conjunto dos vizinhos de 1 salto de $e_{i,s}$. O receptor $r \in N_1^{e_{i,s}}$ será um dos j emissores do *slot* seguinte, $e_{j,s+1}$, se $\forall v \in (N_1^{e_{i,s}} \cap N_1^r)$ temos que $|(N_1^v - N_1^{e_{i,s}})| < |(N_1^r - N_1^{e_{i,s}})|$. Além disto, podemos dizer que r será $e_{k,s+n}$, onde $n = |I| + 1$, sendo $I = \{u \in (N_1^r \cap N_1^{e_{i,s}}) \text{ tal que } |(N_1^u - N_1^{e_{i,s}})| > |(N_1^r - N_1^{e_{i,s}})|\}$.

5. Resultados Numéricos

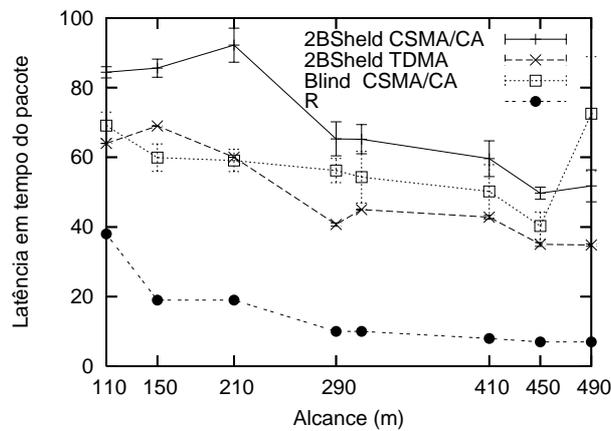


Figura 3. Latências de inundaç o do escalonador em CSMA/CA e TDMA

As avalia es apresentadas nesta se o s o sempre referentes a 10 rodadas com intervalo de confian a de 95%. As medidas de desempenho utilizadas s o a Lat ncia de difus o, calculada como o tempo decorrido entre o in cio do envio de uma mensagem de inunda o e o  ltimo recebimento da mesma, ou seja, os n s que n o recebem a mensagem n o s o considerados, e a Taxa de entrega, calculada como o n mero de recebimentos da mensagem dividido pelo n mero de n s.

A primeira avalia o do mecanismo 2BSheld foi realizada em uma rede em grade id ntica a da Se o 3. O mecanismo, implementado no simulador ns-2,   executado utilizando-se acesso ao meio CSMA/CA e TDMA, em conjunto com o mecanismo Blind. Os resultados s o apresentados na Figura 3 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 mecanismo escalonador acaba provocando uma sincroniza o dos n s, que depois s o novamente escalonados pelo processo de deslocamento por *backoff* e conten o que ocorre no CSMA/CA. Quando o mecanismo Blind   aplicado diretamente ao CSMA/CA, este  ltimo procedimento ocorre logo ap s o recebimento da mensagem, e n o ao in cio de um *slot* sincronizado.

Sobre o TDMA o 2BSheld ainda n o apresenta ganhos significativos de desempenho em rela o ao Blind-CSMA/CA, o que poderia levar a se questionar sua utilidade. Entretanto, os mecanismos de controle de inunda o n o funcionam diretamente sobre o TDMA sem um escalonador, pois a aus ncia de detec o de portadora e *backoff* provoca um excessivo n mero de colis es, e as mensagens n o ultrapassam os primeiros saltos.

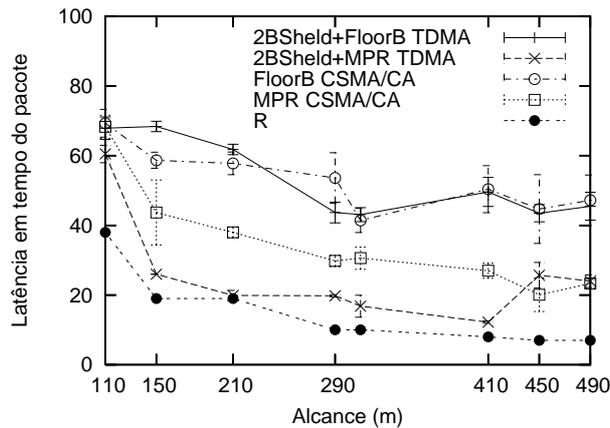


Figura 4. Lat ncias de inunda o do escalonador associado ao FloorB e MPR

Por utilizar a vizinhan a de dois saltos para suas decis es, o 2BSheld   naturalmente associ vel aos mecanismos FloorB e MPR, e os resultados desta associa o entre os mecanismos de controle de inunda o e o mecanismo escalonador s o apresentados na Figura 4. Pode-se observar que n o h  redu o de lat ncia com a associa o ao FloorB, mas h  um ganho com o MPR. Isto ocorre porque o FloorB n o apresenta uma redu o significativa de reencaminhamento de mensagens para este tipo de cen rio, gerando muitas mensagens a serem escalonadas. J  no MPR, com um n mero reduzido de n s a serem escalonados, observa-se a redu o da lat ncia, obtendo valores bem pr ximos de R .

Apesar disto, para redes com maior grau de conectividade, onde o alcance   maior que $410m$, vemos que o escalonador n o consegue a mesma aproxima o do limite te rico R . Isto se deve   forma como a decis o do *slot*   realizada, atrav s da adi o dos valores *lowPri* e *lowId*. Com este processo, um n  v pode ser escalonado em at  N_1^v *slots* a frente, provocando uma grande lat ncia nestas redes com maior grau, principalmente para alcan ar os n s finais, que est o   borda da rede.

Por este motivo, foram realizados testes de desempenho com diversas fun es $s(x, y)$ para se observar as lat ncias obtidas. A Figura 5 apresenta os resultados obtidos para a fun o emp rica $s(x, y) = \text{round}(2 * \log(x + y + 1)) + 1$ que apresentou menores lat ncias para estas redes com maior grau, quando comparados a fun o aditiva original, possivelmente ao custo de um aumento de conflitos de escalonamento, entretanto, sem afetar a taxa de entrega dos mecanismos nestes cen rios¹. O conceito utilizado na escolha desta fun o foi de oferecer sa da 1 para $x = y = 0$, e apresentar valores crescentes com x e y , mas de forma menos acentuada do que na adi o.

Em fun o da observa o acima,   proposto um novo crit rio para atribui o dos *slots*, ou seja, uma nova fun o $s(x, y)$. Neste caso, o conceito utilizado   de que   medida que a mensagem “envelhece” na rede, deve-se procurar escalon -la mais rapidamente. Este conceito pode ser utilizado para determinar um limite de tempo para escalonamento, em fun o de R , ou caso este n o seja conhecido, um limite de tempo em fun o de n mero de *slots*. A decis o deste limiar pode ser da aplica o que realiza a inunda o. Por exemplo, caso a inunda o seja um processo per dico, com alguma informa o de

¹As taxas de entrega obtidas foram superiores a 99%, exceto para alguns mecanismos de controle de inunda o na Figura 1(a) e conforme quando apresentados.

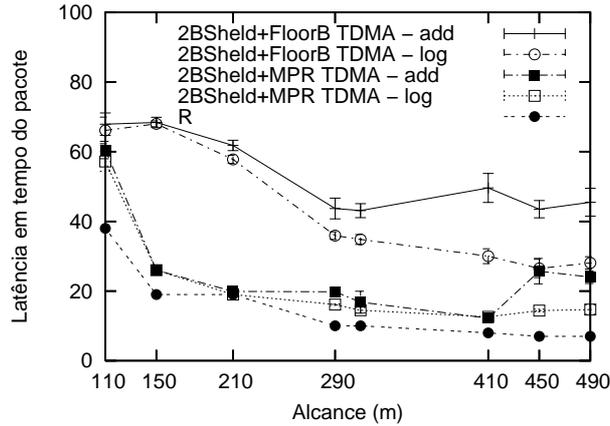


Figura 5. Latências de inundação do escalonador com função log

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.

Desta forma, é possível formalizar uma nova função $s(x, y)$ onde se s é o valor do escalonamento realizado por um nó receptor r para uma mensagem recebida, tem-se que o maior valor possível de s será $s_{max} = |N_1^r|$. Sendo T o limitador de expiração da mensagem, em *slots*, normalmente $T \gg R$, e sabendo que t é o *slot* atual, e h o número de saltos percorridos pela mensagem até chegar a r , temos que $s \leq T - t$, ou, $s_{max} = T - t$.

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, estabelece-se uma proporcionalidade, obtendo $T = \frac{Rt}{h}$ e reescrevendo s_{max} , obtém-se a equação (1).

$$s_{max} = \frac{Rt}{h} - t \quad (1)$$

Estabelecendo uma proporção entre s_{max} calculado na equação (1) e o valor $|N_1^r|$, pode-se admitir a mesma proporcionalidade entre $s(lowPri, lowId) = lowPri + lowId + 1$ e o valor de s ajustado (s'). Então temos (2).

$$s'(lowPri, lowId) = \frac{s(lowPri, lowId) \times (\frac{Rt}{h} - t)}{|N_1^r|} = \frac{(lowPri + lowId + 1)t(R - h)}{h|N_1^r|} \quad (2)$$

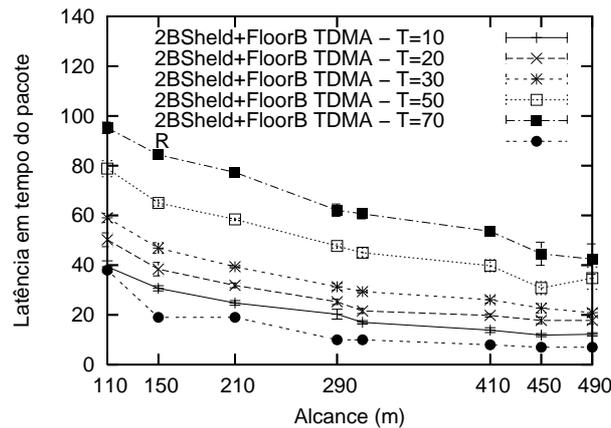
Importante observar que, exceto R , as demais variáveis da equação são obtidas localmente ou da própria mensagem de inundação. Caso não se conheça R , a equação (2) pode ser reescrita facilmente conforme a equação (3), em função de T .

$$s'(lowPri, lowId) = \frac{(lowPri + lowId + 1)(T - t)}{|N_1^r|} \quad (3)$$

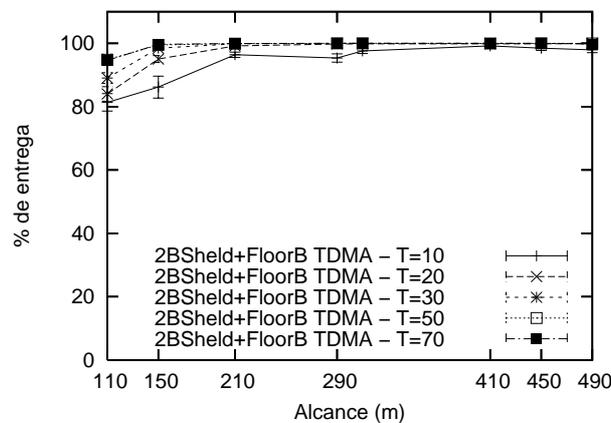
Na Figura 6(a), temos os resultados de latência para o escalonamento realizado com alguns valores de limitador T . Observa-se nesta figura que as latências máximas

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 6(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 $T \gg R$.



(a) Latências de inundação

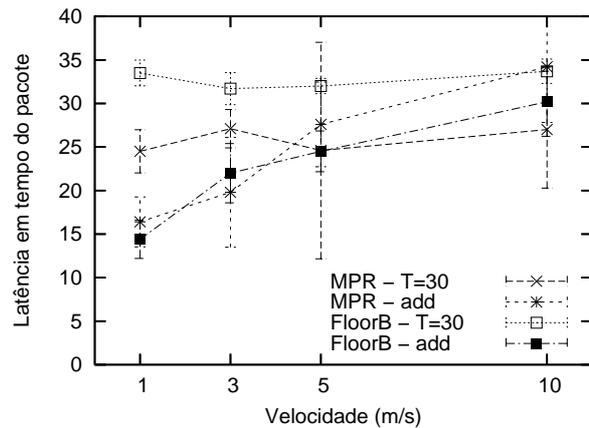


(b) Taxas de entrega

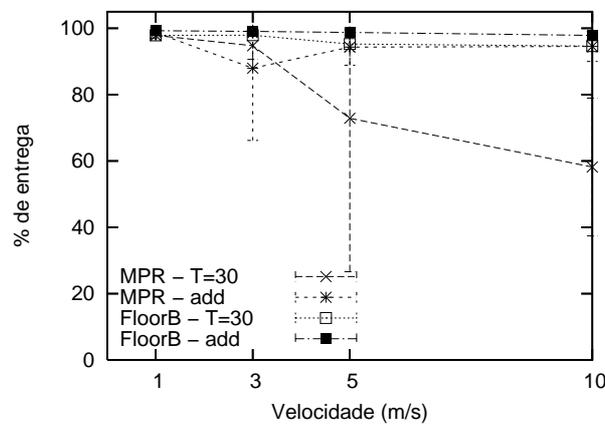
Figura 6. Latência e taxa de entrega com limite T

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 de T podem levar a escalonamentos muito longos. No passo seguinte, o mecanismo é avaliado em um tipo de cenário não abordado pelos trabalhos já citados, aqueles que apresentam mobilidade.

A Figura 7 apresenta os resultados do escalonador em redes móveis, com $T = 30$ e com a função s aditiva. 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 de $0m/s$ ao máximo indicado. Na Figura 7(a) é apresentada a latência do



(a) Latência



(b) Taxa de Entrega

Figura 7. Latência e taxa de entrega de inundação com mobilidade

escalonador associado ao FloorB e ao MPR, na Figura 7(b) temos as taxas de entrega.

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 grande, resultado já apresentado em [Augusto and de Rezende 2009] independente de escalonador. Isto ocorre porque em algumas rodadas a taxa de entrega foi muito baixa, pois a mensagem não ultrapassou os primeiros saltos. Nestes cenários, o nó emissor, devido à mobilidade, perde a vizinhança com os MPRs, e a inundação ocorre antes do mecanismo ser atualizado pelos *Hellos*.

6. Conclusões e Trabalhos Futuros

Este artigo apresentou uma abordagem para o problema de latência mínima para escalonamento de inundação (MLBS). Apesar de ser um problema NP-Completo, com algumas propostas na literatura, este trabalho mostra que algoritmos aproximados existentes apresentam valores limites mais próximos dos limites superiores esperados do que dos limites inferiores. Também foi apresentada a latência de mecanismos comuns de inundação, indi-

cando que ao se realizar uma escolha adequada de nós reencaminhadores, a latência pode ser diminuída consideravelmente.

Diante destas observações, este trabalho 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 desde redes pouco conectadas até redes com grau elevado, ou 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, este trabalho indica valores reais de latência esperados por mecanismos escalonadores de inundação, e estabelece 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 de ser um problema NP-Completo, heurísticas simples permitem obter, sem garantias, resultados próximos ao ótimo, na maior parte dos cenários.

Como trabalhos futuros, apresentam-se duas vertentes principais. A primeira é realizar uma integração entre o escalonador e os mecanismos de controle de inundação MPR e FloorB, de forma que as decisões de um mecanismo impactem no outro (*cross-layer*). A segunda linha de trabalhos futuros consiste em investigar a utilização da Equação 2, considerando o conhecimento prévio de R , e avaliar o mecanismo em ambientes que utilizem modelo físico de forma mais realista, uma vez que a maior parte das propostas de escalonamento se baseiam em modelos de protocolo.

Referências

- Augusto, C. H. P. and de Rezende, J. F. (2006). An adaptive approach to service discovery in ad hoc networks. In *8th IFIP IEEE International Conference on Mobile and Wireless Communications Network, 2006.*, pages 61–75.
- Augusto, C. H. P. and de Rezende, J. F. (2009). Inundação probabilística em redes ad hoc utilizando sumário de vizinhança através de filtros de bloom. In *XXVII Simpósio Brasileiro de Redes de Computadores - SBRC2009.*
- Badia, L., Erta, A., Lenzini, L., and Zorzi, M. (2008). A General Interference-Aware Framework for Joint Routing and Link Scheduling in Wireless Mesh Networks. *IEEE Network*, 22(1):32–38.
- Chlamtac, I. and Kutten, S. (1985). On broadcasting in radio networks—problem analysis and protocol design. *Communications, IEEE Transactions on*, 33(12):1240 – 1246.
- Chlamtac, I. and Weinstein, O. (1991). The wave expansion approach to broadcasting in multihop radio networks. *Communications, IEEE Transactions on*, 39(3):426 –433.
- Clausen, T. and Jacquet, P. (2003). RFC 3626 - Optimized Link State Routing protocol OLSR. *IETF Request for Comments.*
- Drabkin, V., Friedman, R., Kliot, G., and Segal, M. (2007). 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, Washington, DC, USA. IEEE Computer Society.

- Fujie, T. (2003). An exact algorithm for the maximum leaf spanning tree problem. *Computers and Operations Research*, 30(13):1931 – 1944.
- Gandhi, R., Mishra, A., and Parthasarathy, S. (2008). Minimizing broadcast latency and redundancy in ad hoc networks. *IEEE/ACM Trans. Netw.*, 16(4):840–851.
- Haas, Z., Halpern, J. Y., and Li, L. (2002). Gossip-based ad hoc routing. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications*, volume 3, pages 1707 – 1716.
- Huang, S. C.-H., Wan, P.-J., Jia, X., Du, H., and Shang, W. (2007). Minimum-latency broadcast scheduling in wireless ad hoc networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications*. IEEE, pages 733–739.
- Karumanchi, G., Muralidharan, S., and Prakash, R. (1999). Information dissemination in partitionable mobile ad hoc networks. In *Proceedings of IEEE Symposium on Reliable Distributed Systems (SRDS)*, pages 4–13.
- Laouiti, A., Qayyum, A., and Viennot, L. (2002). Multipoint relaying: An efficient technique for flooding in mobile wireless networks. In *35th Annual Hawaii International Conference on System Sciences (HICSS'2002)*.
- Lenders, V., May, M., and Plattner, B. (2005). 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.
- Ni, S.-Y., Tseng, Y.-C., Chen, Y.-S., and Sheu, J.-P. (1999). 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. ACM.
- NS-2 (1995). The network simulator - ns-2.
<http://www.isi.edu/nsnam/ns/> - último acesso em 17/12/2008.
- Perkins, C., Belding-Royer, E., and Das, S. (2003). RFC 3561 - Ad hoc On-demand Distance Vector AODV routing. *IETF Request for Comments*.
- Vergados, D. J., Manolaraki, M.-Y., and Vergados, D. D. (2009). 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.*, pages 394 – 398.
- Wu, X., Sharif, B., Hinton, O., and Tsimenidis, C. (2005). Solving optimum tdma broadcast scheduling in mobile ad hoc networks: a competent permutation genetic algorithm approach. *Communications, IEE Proceedings-*, 152(6):780–788.
- Yeo, J., Lee, H., and Kim, S. (2002). An efficient broadcast scheduling algorithm for tdma ad-hoc networks. *Computers & Operations Research*, 29(13):1793 – 1806.
- Zhang, X.-Z. (2007). Efficient broadcast scheduling based on fuzzy clustering and hop-field network for ad hoc networks. In *International Conference on Machine Learning and Cybernetics, ICMLC 2007*, volume 6, pages 3255–3260.