

# Um Algoritmo de Difusão Baseado na Troca de Resumos\*

João Ludovico and Hugo Miranda

Universidade de Lisboa  
Faculdade de Ciências  
LaSIGE

**Resumo** A difusão é uma operação muito utilizada em redes não infra-estruturadas. Contudo, alguns algoritmos de difusão utilizam uma grande quantidade de retransmissões supérfluas, com custos evidentes na ocupação do meio e nos recursos energéticos dos dispositivos. A diminuição do número de retransmissões é um desafio considerável, uma vez que se procuram algoritmos completamente descentralizados e adequados para dispositivos onde os recursos computacionais são limitados.

Neste artigo é apresentado um algoritmo de difusão cujo funcionamento assenta em estimativas, locais a cada nó, sobre a disposição geográfica dos vizinhos. A partir destas estimativas, cada nó selecciona um conjunto de vizinhos que acredita encontrarem-se em regiões suficientemente distantes para assegurarem a continuação da difusão em todas as direcções da rede. Embora o algoritmo tente estimar localizações geográficas, assume que os dispositivos não dispõem de um serviço de localização geográfica (GPS). Adicionalmente, o algoritmo é escalável, na medida em que não faz depender o número de retransmissões de cada mensagem difundida do número de vizinhos e faz crescer o estado mantido em cada nó linearmente com o número de vizinhos directos.

## 1 Introdução

As redes ad hoc sem fios são caracterizadas pela ausência de uma infra-estrutura que ligue e forneça serviços aos vários nós que formam a rede. Nestas redes, um nó só comunica directamente com nós que estejam dentro do seu raio de transmissão. Deste modo, uma mensagem que se pretende que seja distribuída por todos os nós tem de ser propagada sucessivamente pelos participantes ao longo da rede. A difusão de mensagens é usada por vários serviços, como por exemplo, serviços de publicador/subscritor, de descoberta de rotas e descoberta de recursos.

A concretização usual de difusão é através de um algoritmo de *inundação*, em que a mensagem é retransmitida por todos os nós quando a recebem pela primeira vez. A inundação tem uma elevada taxa de entrega mas, por outro lado,

---

\* O trabalho descrito neste artigo foi parcialmente suportado pela Fundação para a Ciência e Tecnologia (FCT) através do projecto PTDC/EIAEIA/103751/2008 - PATI

leva a um grande consumo de largura de banda e à redução do tempo de vida dos nós, uma vez que o envio de mensagens contribui significativamente para o consumo da bateria [5]. Adicionalmente, durante uma inundação ocorre um elevado número de colisões, resultado da utilização de um meio partilhado. Tendo em conta estes problemas, o desafio de criar algoritmos de difusão eficientes, ou seja, com uma taxa de entrega elevada, é a redução do número de retransmissões, que por sua vez irá reduzir o consumo de largura de banda e a taxa de colisões.

A selecção do conjunto mínimo de nós que assegure a entrega a todos os participantes seria simplificada se os nós dispusessem de um serviço de localização geográfica (por exemplo, GPS). Contudo, este artigo assume um modelo de sistema mais genérico, em que os nós não dispõem de um serviço de localização, tornando-o adequado a um maior conjunto de aplicações de redes não infra-estruturadas como as redes de sensores e as redes dentro de edifícios. Com estas limitações pretende-se determinar um conjunto de transmissores tão reduzido quanto possível mas que assegure a cobertura total da rede. Uma aproximação possível à determinação deste conjunto consiste em fazer com que cada nó selecione como retransmissores alguns dos seus vizinhos, idealmente aqueles que se encontrem mais distantes entre si, por forma a que as suas retransmissões não sejam redundantes por cobrirem os mesmos nós deixando outros a descoberto.

Este artigo apresenta e avalia um algoritmo de difusão que permite a cada nó estimar a proximidade dos seus vizinhos de modo a reduzir retransmissões redundantes. A proximidade dos vizinhos é estimada sem recorrer a um serviço de localização geográfica, utilizando apenas mensagens de sinalização trocadas entre os nós e que agregam informação sobre a vizinhança, fazendo com que nós vizinhos apresentem agregações semelhantes. O algoritmo de difusão utiliza depois uma função de comparação sobre a informação agregada de forma a seleccionar o conjunto de vizinhos que deverá retransmitir a mensagem por forma a assegurar a sua propagação.

O artigo está organizado da forma indicada em seguida. A secção 2 descreve o trabalho relacionado em algoritmos de difusão. O algoritmo proposto é apresentado na secção 3 e avaliado na secção 4. A secção 5 conclui o artigo.

## 2 Trabalho Relacionado

É possível encontrar na literatura variadas propostas de algoritmos de difusão de mensagens para redes não infra-estruturadas. Esta secção apresenta apenas aquelas que são mais relevantes no âmbito do trabalho desenvolvido. Um estudo mais completo pode ser encontrado por exemplo em [7].

Nos algoritmos não cientes de contexto, a decisão por parte de um dispositivo de retransmitir ou não retransmitir uma mensagem advém apenas da informação com que foi configurado inicialmente. Um exemplo destes algoritmos é a difusão probabilista [8], em que cada nó retransmite uma mensagem recebida pela primeira vez com uma probabilidade  $p$ . O algoritmo de inundação pode ser considerado o caso particular da difusão probabilista em que  $p = 1$ .

Nos algoritmos probabilistas, o número de retransmissões de cada mensagem pode ser aumentado ou reduzido fazendo variar a probabilidade de retransmissão. Contudo, uma definição adequada de um valor para  $p$  depende das características da vizinhança de cada nó. Em regiões com uma elevada densidade de dispositivos, um número reduzido de retransmissões e portanto, um valor baixo de  $p$  é suficiente para atingir uma elevada taxa de entrega. Em cenários cuja densidade é baixa, o valor de  $p$  tem de ser mais elevado. Infelizmente, os algoritmos não cientes do contexto não dispõem dos mecanismos necessários à regulação de  $p$  em função das características do ambiente, obrigando à utilização de valores conservadores que tendem em muitos casos a resultar num desperdício de recursos dos dispositivos.

Nos algoritmos cientes do contexto o dispositivo inclui na sua tomada de decisão informação relativa ao meio em que se encontra. Esta informação pode ser, por exemplo, relativa à localização geográfica dos vizinhos, ao tráfego de dados na rede ou a sua distância aos outros dispositivos.

No caso da localização geográfica os dispositivos partilham um sistema de coordenadas comum e obtêm a sua posição nesse sistema (por exemplo GPS). Esta informação é depois utilizada para os dispositivos decidirem se retransmitem a mensagem. Por exemplo em [6], cada retransmissor difunde juntamente com a mensagem as 6 coordenadas alvo onde se deverão encontrar os próximos retransmissores e que resultarão numa cobertura óptima da região. A retransmissão é depois assegurada pelos nós mais próximos dessas coordenadas. Outros exemplos de algoritmos que usam esta abordagem podem ser encontrados em [3,10,11]. Salienta-se no entanto, que a discussão destas aproximações está fora do âmbito deste artigo, uma vez que requerem a disponibilidade de um sistema de posicionamento geográfico nos dispositivos participantes.

Nos algoritmos dependentes do tráfego, os dispositivos tomam a decisão de retransmitir ou não retransmitir a mensagem de acordo com o número de cópias recebidas. Estes algoritmos seguem o princípio de que a utilidade de uma retransmissão (medida pelo número de novos nós que recebem a mensagem) diminui com o número de retransmissões numa dada vizinhança [12]. Num destes algoritmos [12], os dispositivos atrasam a retransmissão de uma mensagem por um período aleatório durante o qual contam o número de cópias escutadas. O número mínimo de cópias escutadas que previne a retransmissão é fixado à partida e independente da envolvente a cada nó. Em alternativa, no Dynamic Counter-Based broadcast (DCB) [1] cada dispositivo ajusta o limite de cópias escutadas em função de informação obtida dos seus vizinhos directos. Assim, se o dispositivo tiver muitos vizinhos vai considerar que a rede é densa e vai ter um limite mais reduzido.

Ao fazer depender o tempo de espera de um gerador de números aleatórios, os algoritmos dependentes do tráfego mantêm a aleatoriedade na escolha dos nós que retransmitem. Esta é uma solução sub-óptima uma vez que os nós seleccionados para retransmissão podem encontrar-se bastante próximos dos retransmissores anteriores. Esta proximidade pode diminuir a utilidade da retransmissão, uma vez que o espaço coberto pela nova retransmissão será, em grande parte, seme-

lhante à retransmissão anterior. O algoritmo PAMPA [9] é um exemplo da classe de algoritmos que inclui a distância entre os nós na decisão de retransmissão. O PAMPA aborda este problema complementando os algoritmos dependentes do tráfego com a preferência de retransmissão dos nós mais distantes. Para tal, o PAMPA faz depender o tempo de espera da força de sinal com que as mensagens são recebidas, de forma a que dispositivos mais distantes do emissor tenham um menor tempo de espera e, conseqüentemente, retransmitam primeiro. Contudo, o processo de selecção de retransmissores do PAMPA ignora particularidades na topologia da rede, que são acentuadas pela baixa redundância do PAMPA e que, em alguns casos excepcionais, reduzem a taxa de entrega [4].

### 3 The Digest Exchange Broadcast Algorithm (DEBA)

O Digest Exchange Broadcast Algorithm (DEBA) reduz o número de retransmissões de uma mensagem delegando no nó que origina a mensagem e em cada retransmissor a selecção do conjunto de vizinhos que continuarão a propagação. De entre os seus vizinhos, cada nó selecciona um conjunto que presume representar localizações geograficamente distintas, por forma a assegurar a continuação da disseminação em todas as direcções. Contudo, a localização dos vizinhos pode apenas ser estimada, uma vez que é assumido que todos os nós: *i*) utilizam antenas omnidireccionais (e portanto não permitem perceber a direcção de onde a mensagem foi recebida) e, *ii*) não dispõem de um serviço de localização geográfica (por exemplo GPS). Assim, o DEBA impõe exclusivamente como restrição a bidireccionalidade da comunicação, o que o torna adequado para a maioria dos cenários antecipados para as redes não infra-estruturadas, como as redes de sensores, ad hoc ou veiculares.

A estimativa da localização relativa dos dispositivos é realizada pela associação a cada nó de uma identificação independente da localização, gerada de forma descentralizada. Contudo, o formato desta identificação permite que cada nó a agregue com a dos seus vizinhos mais próximos. Ou seja, cada nó partilhará com os seus vizinhos uma parte da identificação agregada, mas (na maior parte dos casos) não a totalidade, uma vez que cada nó terá vizinhos que não são comuns. Durante a difusão, cada nó procura seleccionar vizinhos com agregações suficientemente distintas para evitar difusões redundantes na mesma região. As subsecções seguintes detalham o DEBA, apresentando primeiro uma descrição abstracta e posteriormente uma concretização concebida para atender aos baixos recursos e largura de banda dos dispositivos móveis.

#### 3.1 Estruturas de Dados

O algoritmo DEBA é totalmente descentralizado. Adicionalmente, o estado que deve ser mantido em cada nó é limitado pelo número de vizinhos directos (a 1 salto) o que torna o algoritmo escalável. O algoritmo assume que cada nó  $n$  tem um identificador único ( $E_n$ ), que pode por exemplo ser o seu endereço de rede ou MAC. Adicionalmente, cada nó  $n$ , mantém ainda as seguintes estruturas de dados.

*Vector Local* O vector local de um nó  $n$  ( $VL_n$ ) é um conjunto de  $k$  inteiros seleccionados aleatoriamente do intervalo  $[0, \max_I]$ . O DEBA é ortogonal ao mecanismo utilizado para a geração de  $VL_n$ , requerendo apenas que a probabilidade de dois quaisquer nós seleccionarem o mesmo conjunto seja baixa. Um gerador de números aleatórios ou uma sequência de operações de dispersão (*hash*) aplicadas a  $E_n$  são exemplos de funções que podem ser utilizadas.

*Vector de Vizinhança* Um vector de vizinhança de um nó  $n$  ( $VV_n$ ) é a união de  $VL_n$  com os vectores locais de todos os seus vizinhos directos. No âmbito do vector de vizinhança, não é feita qualquer associação entre os seus elementos e o nó que contribuiu com esses elementos para o vector. De notar que o vector de vizinhança não é imutável, sendo refrescoado com a aprendizagem dos vectores locais dos seus vizinhos. Contudo, numa rede onde os dispositivos não se movem, os vectores de vizinhança estabilizam após algumas iterações.

O mecanismo utilizado para disseminar vectores de vizinhança é ortogonal ao DEBA. As concretizações do algoritmo podem associar os vectores a mensagens de dados ou utilizar uma mensagem específica, difundida periodicamente para a rede.

*Tabela de Vizinhança* A Tabela de Vizinhança ( $TV_n$ ) de um nó  $n$  é uma associação entre cada vizinho  $v$  a um salto de  $n$  e o vector de vizinhança anunciado por  $v$ . A tabela serve dois propósitos. Por um lado, facilita a geração de  $VV_n$ , que desta forma consiste apenas na união de todos os elementos de  $TV_n$ . Por outro, oferece ao nó uma estimativa da proximidade dos seus vizinhos, pela comparação dos elementos em comum nos vectores de vizinhança de cada um dos seus vizinhos.

De modo a ilustrar os conceitos anteriores, a Fig. 1 apresenta um subconjunto de uma rede composta por 4 nós. Os nós  $A$  e  $D$  têm conectividade entre si e com o nó  $B$ , o nó  $C$  tem apenas conectividade com o nó  $B$ . As estruturas  $VL$ ,  $VV$  e  $TV$  junto a cada nó apresentam os valores após a estabilização do algoritmo. Por clareza da apresentação, os membros de  $VV_n$  que pertencem simultaneamente a  $VL_n$  estão assinalados com um asterisco (\*), denotando desta forma que devem ser propagados pelos receptores de  $VV_n$  uma vez que pertencem a um vizinho directo do receptor.

Da figura pode-se observar que  $VV_A = VV_D$  o que sugere ao nó  $B$  que  $A$  e  $D$  se encontram próximos. Analogamente,  $TV_B$  mostra uma clara diferença entre estes vectores e  $VV_C$ , o que sugere que os primeiros não se encontram no raio de alcance do último. Por sua vez, os nós  $A$  e  $D$ , podem aperceber-se da sua proximidade e da diferença do vector de vizinhança de  $B$ , por este último conter os elementos em  $VL_C$ , o que não acontece com  $VV_A$  e  $VV_D$ .

### 3.2 Algoritmo de Difusão

O cabeçalho de cada mensagem difundida utilizando o DEBA contém um identificador único de mensagem gerado pelo emissor (que pode ser obtido por exemplo

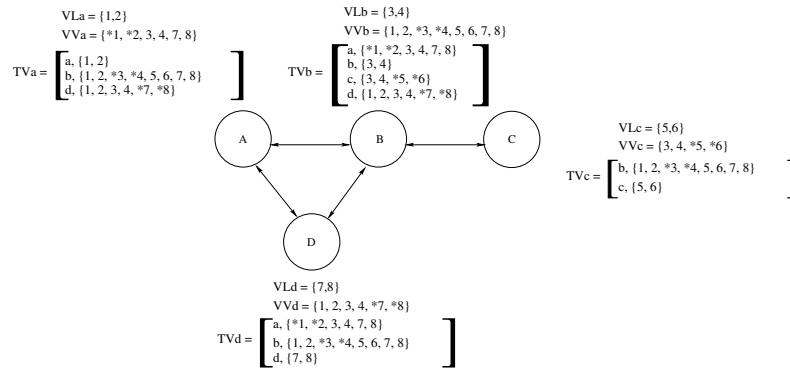


Figura 1. Estruturas de dados utilizadas pelo algoritmo DEBA

pela concatenação de  $E_n$  com um número de série local ao emissor). O identificador único de mensagem permite aos nós identificar duplicados, que são descartados. Esta funcionalidade assegura igualmente a terminação da propagação das mensagens, dispensando estimativas sobre o número máximo de saltos que a mensagem irá necessitar para cobrir a totalidade da rede e portanto um campo do tipo *time-to-live* ( $TTL$ ). Adicionalmente, o cabeçalho da mensagem contém uma lista dos endereços seleccionados para retransmitir a mensagem. Esta lista é preenchida por cada nó que transmite a mensagem, com endereços de nós que constam da sua Tabela de Vizinhança, ou seja, dos seus vizinhos directos. Cada nó transmite a mensagem no máximo uma vez, utilizando o endereço de difusão do nível de ligação de dados. Assim, e a menos de colisões, a mensagem é recebida por todos os vizinhos do transmissor, independentemente de virem ou não a retransmitir a mensagem.

A função de selecção dos vizinhos que retransmitirão cada mensagem é a parte central do algoritmo DEBA e é executada independentemente por cada nó que tenha sido seleccionado para retransmitir a mensagem. Para um nó  $n$ , a função recebe como argumentos  $TV_n$  e a lista de vizinhos seleccionados pelo nó do qual a mensagem foi recebida. A função retorna a lista dos vizinhos seleccionados. Genericamente, a função pode ser decomposta em dois passos: *i*) ordenação de  $TV_n$ , *ii*) sequencialmente, para cada nó em  $TV_n$ , decidir se o nó é ou não seleccionado.

O critério de ordenação de  $TV_n$  influencia o algoritmo uma vez que os primeiros elementos analisados terão preferência de selecção sobre os restantes. Assim, o passo *i*) ordena  $TV_n$  por ordem decrescente da dimensão do vector de vizinhança. Este critério é justificado por cada nó gerar um vector local com o mesmo número de inteiros e portanto, a um Vector de Vizinhança com mais elementos corresponder um nó com um maior número de vizinhos, ou seja, com uma maior cobertura da retransmissão.

Para uma mensagem  $m$ , o passo *ii*) compara sequencialmente o vector de vizinhança de cada um dos elementos de  $TV_n$  com cada um dos elementos de um conjunto  $C_n^m$  composto pelos vectores de vizinhança de:

- o nó do qual foi recebida a mensagem  $m$ ;
- os nós que constam da lista de endereços em  $m$  e que constem também de  $TV_n$ ;
- nós entretanto já seleccionados para retransmissão em iterações anteriores do passo *ii*).

Para que um nó seja seleccionado, tem de se mostrar suficientemente distinto na comparação com todos os elementos de  $C_n^m$ , caso em que o nó passa a integrar  $C_n^m$  para as comparações com os nós seguintes. As comparações podem seguir diferentes critérios. Sejam  $VV_a \in TV_n, VV_b \in C_n^m$  dois vectores de vizinhança em comparação, neste artigo, são experimentadas e avaliadas as seguintes quatro métricas:

**rácio total, determinista (RTD)** quociente do número de inteiros em comum entre os dois vectores pelo número total de inteiros na união dos dois vectores, ou seja,  $RTD(a, b) = \frac{|VV_a \cap VV_b|}{|VV_a \cup VV_b|}$ . Os vectores são considerados semelhantes se o quociente for superior a uma constante pré-definida.

**rácio total, probabilista (RTP)** como em RTD, contudo, o vector é considerado distinto se RTP for inferior a uma constante pré-definida. Caso contrário, a semelhança é determinada por uma função probabilista que decresce linearmente com o valor de RTP.

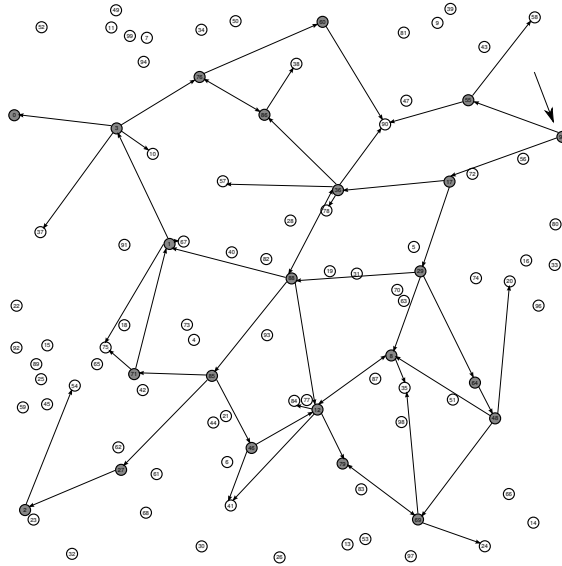
**rácio mínimo, determinista (RMD)** quociente do número de inteiros em comum entre os dois vectores pela dimensão do menor vector, ou seja,  $RMD(a, b) = \frac{|VV_a \cap VV_b|}{\min\{|VV_a|, |VV_b|\}}$ . Os vectores são considerados semelhantes se o quociente for superior a uma constante pré-definida.

**rácio mínimo, probabilista (RMP)** como em RMD, contudo, o vector é considerado distinto se RMP for inferior a uma constante pré-definida. Caso contrário, a semelhança é determinada por uma função probabilista que decresce linearmente com o valor de RMP.

De notar que qualquer das métricas resulta trivialmente na exclusão de nós que estejam já presentes em  $C_n^m$  uma vez que, para qualquer nó  $n$ ,  $RTD(n, n) = RMD(n, n) = 1$ . A operação de retransmissão termina com a incorporação dos endereços dos nós seleccionados no cabeçalho da mensagem.

*Exemplo* Para ilustrar a execução do algoritmo de disseminação, a Fig. 2 apresenta a disseminação de uma mensagem numa rede com 100 nós utilizando a função **RMD** no cenário de  $1000m \times 1000m$  descrito em detalhe na Sec. 4. A disseminação da mensagem foi iniciada no nó assinalado no canto superior direito. Os destinos das setas indicam os nós seleccionados para retransmissão pelo nó onde a seta tem origem.

A figura permite observar experimentalmente as boas capacidades de distribuição espacial das retransmissões obtidas pelo DEBA. Os ângulos abertos



**Figura 2.** Exemplo da propagação de uma mensagem utilizando o DEBA

formados pelas setas com origem no mesmo nó sugerem que na maioria dos casos da disseminação estudada não ocorrem pedidos de retransmissão para nós próximos. A figura demonstra ainda a existência de alguma redundância nos pedidos de retransmissão, observável pelos conjuntos de setas com chegada a um mesmo nó e por algumas setas bidireccionais, que indicam que dois nós se seleccionam mutuamente. A análise a estes casos demonstrou que resultam do processamento concorrente da mesma mensagem por vários nós, que seleccionam o mesmo dispositivo para retransmissão. Estes casos são resolvidos trivialmente pelo algoritmo uma vez que cada nó não retransmite cada mensagem mais de uma vez.

### 3.3 Concretização

Um dos desafios do DEBA é conseguir um equilíbrio entre a redução da probabilidade de repetição de vectores locais, a dimensão das tabelas e dos vectores de vizinhança e a complexidade computacional das operações a realizar.

Uma probabilidade elevada de repetição de vectores locais reduz a eficiência do algoritmo por fornecer informações incorrectas sobre a proximidade dos nós. Por essa razão, importa alargar tanto quanto possível o número de combinações disponível em  $VL_n$ , ditado pelas constantes  $max_I$  e  $k$ . Os restantes factores apelam à utilização de valores pequenos em  $k$ , por forma a diminuir a memória



ocupada por  $TV_n$ , a largura de banda e energia consumidas com a retransmissão de  $VV_n$  e o número de comparações realizadas para selecção dos retransmissores.

A concretização inspira-se no conceito de *Bloom Filter* [2], um vector de bits de tamanho fixo, nos quais são representados com o valor 1 apenas os bits correspondentes aos inteiros que se pretende assinalar. De entre as características que tornam os bloom filters apelativos encontra-se a reduzida complexidade computacional das operações de união e intersecção de conjuntos, conseguidas utilizando os operadores de disjunção e conjunção bit-a-bit, nativos na maioria dos processadores actuais. Adicionalmente, a utilização de *bloom filters* fixa uma dimensão para os vectores de vizinhança, que se torna independente do número de vizinhos de cada dispositivo. Para a avaliação do algoritmo, foram definidas as constantes  $\max_I = 1023, k = 8$ . Ou seja, cada vector local  $VL_n$  é composto por 8 inteiros seleccionados aleatoriamente do intervalo  $[0, 1023]$ , a que corresponderá um *Bloom Filter* com 1024 bits (128 bytes), dos quais 8 terão o valor 1.

Contudo, os *bloom filters* não permitem a contabilização do número de saltos que tem de ser associada a cada um dos bits. A transmissão dessa informação é realizada acrescentando à mensagem que contém o vector de vizinhança um segundo vector de bits com a mesma dimensão. Neste vector, cada posição  $i$  terá o valor 1 caso o bit  $i$  do vector de vizinhança deva ser propagado pelos receptores e o valor 0 caso contrário. Por analogia à Fig. 1, este vector terá activos os bits correspondentes às posições assinaladas com asterisco. A dimensão final das mensagens de anúncio dos vectores de vizinhança é assim de 256 bytes. De notar que estas constantes podem ser alteradas para atender a restrições externas, por exemplo uma dimensão máxima das tramas ao nível de ligação de dados ou uma grande quantidade de vizinhos que aumente a probabilidade de repetição de vectores locais.

## 4 Avaliação

Nesta secção são apresentados e discutidos os resultados do DEBA utilizando as métricas descritas na Sec. 3. Os valores utilizados para estas métricas podem ser observados na Tab. 1. Os resultados são comparados com os obtidos utilizando dois algoritmos de controlo, a *inundação* e a *difusão cega*. No algoritmo de *inundação* todos os nós retransmitem após receberem a mensagem pela primeira vez. A inundação apresenta na maioria dos casos a cobertura máxima possível da rede, permitindo distinguir ineficiências dos algoritmos avaliados de incapacidade física de entrega das mensagens resultante da ocorrência de partições na rede.

**Tabela 1.** Valores utilizados nas métricas

Métrica	Constante	Métrica	Mínimo	Máximo	Função
RTD	0.65	RTP	0.6	0.7	$(RTD(a, b) - 0.6) \times 10$
RMD	0.85	RMP	0.8	0.9	$(RMD(a, b) - 0.8) \times 10$

Para confirmar a relevância das funções utilizadas no DEBA na taxa de entrega utiliza-se um algoritmo de *difusão cega*. Neste algoritmo as funções estudadas para o DEBA são substituídas por uma outra que selecciona aleatoriamente os retransmissores. De forma a assegurar condições comparáveis, a função selecciona como retransmissores 25% dos vizinhos do nó, ou seja, uma taxa de retransmissão comparável à obtida pelas restantes funções em redes com uma dimensão intermédia (cf. Fig. 4).

Os testes foram realizados utilizando o simulador de redes *ns-2* v. 2.34. Os nós foram configurados para simularem a interface radio Lucent Wave-LAN DSSS a 914 MHz utilizando o protocolo IEEE 802.11 a 2 Mb/s, com um raio de transmissão de 250 metros. O modelo de propagação utilizado foi o Two-Ray Ground, que assegura a entrega das mensagens a menos da ocorrência de colisões, o que permite abstrair o desempenho dos algoritmos de factores externos, como as interferências.

Em cada teste, 100 dispositivos são distribuídos aleatoriamente em redes com dimensões de  $500m \times 250m$ ,  $500m \times 500m$ ,  $1000m \times 500m$ ,  $1000m \times 1000m$ ,  $2000m \times 1000m$  e  $2000m \times 2000m$  a que correspondem a taxas de ocupação entre  $1250m^2/nó$  e  $40000m^2/nó$ . Em cada dimensão de rede foram realizados 100 testes, com disposições distintas dos nós. Cada teste consiste na difusão de 100 mensagens, iniciadas por nós distintos. As mensagens têm um tamanho fixo de 682 bytes, o que inclui os cabeçalhos e hipotéticos dados da aplicação. Um tamanho fixo de mensagem torna a avaliação ortogonal ao formato utilizado pelos vectores locais dos nós. Cada teste tem a duração de 1500 segundos. Os primeiros 390 constituem uma fase de aquecimento, com os nós a trocarem periodicamente os seus vectores de vizinhança. Os resultados apresentados reportam os últimos 1100 segundos, durante os quais são difundidas as mensagens. Os nós permanecem imóveis ao longo das simulações, emulando desta forma uma rede de sensores.

A vantagem das funções estudadas sobre aproximações probabilistas é visível na Fig. 3, que compara a taxa de entrega, ou seja, a proporção média de nós que receberam cada mensagem. Na figura, as barras de erro representam a distância do valor médio ao desvio padrão. Ao observarmos os resultados da difusão cega, constata-se que a sua taxa de entrega se afasta significativamente das restantes com o aumento da área. Este facto é atribuído a escolhas incorrectas de retransmissores que impedem a propagação para algumas regiões da rede. Pelo contrário, o DEBA acompanha as taxas de entrega da inundação em todas as densidades, sendo ligeiramente mais penalizado nas maiores áreas, ou seja, onde ocorrem partições. A figura mostra ainda um desempenho consistentemente inferior das funções de rácio mínimo.

A Fig. 4 mostra que a ligeira penalização da taxa de entrega do DEBA é compensada por uma significativa redução do número de retransmissões, com reflexos na autonomia dos dispositivos e na largura de banda consumida. Na figura, as barras de erro representam a distância do valor médio ao desvio padrão. Como desejável, o DEBA acompanha o aumento da área com um aumento da taxa de retransmissão, ou seja, a proporção média de nós que retransmitem cada

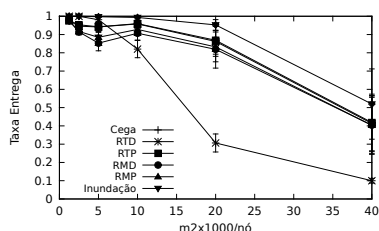


Figura 3. Taxa de entrega

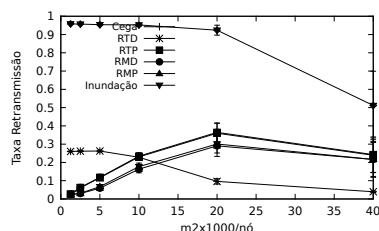


Figura 4. Taxa de retransmissão

mensagem. Isto acontece porque os dispositivos estão mais afastados uns dos outros e portanto são necessárias mais retransmissões para cobrir o mesmo número de dispositivos. Os decréscimos da taxa de retransmissão com o crescimento da área são justificados, no caso da inundação e do DEBA pela impossibilidade de atingir os dispositivos devido à existência de partições e pela baixa taxa de entrega, que impede a continuação da propagação, no caso da difusão cega.

As figuras mostram ainda que não existe diferença significativa entre as funções RTD e RTP e entre RMD e RMP. Isto é justificado porque, no caso das funções probabilistas, à medida que o resultado da função de comparação se aproxima do valor máximo, a probabilidade de um dispositivo ser seleccionado diminui. Adicionalmente, esta semelhança sugere que, na generalidade dos casos, as funções estudadas para o DEBA são resilientes a falhas de alguns nós seleccionados.

É de notar, no entanto, que o desempenho do DEBA está dependente das funções de comparação utilizadas e dos parâmetros utilizados na definição dos vectores locais dos nós. Em casos limite, em que ocorra grande semelhança de vectores locais e portanto de vectores de vizinhança, as funções de comparação podem não seleccionar nenhum dispositivo para continuar a propagação.

## 5 Conclusão e Trabalho Futuro

A disseminação de mensagens é uma operação crucial para diversos algoritmos utilizados em redes não infraestruturadas. Este artigo apresentou o Digest Exchange Broadcast Algorithm (DEBA), um algoritmo de difusão que tem como objectivo a diminuição do número de mensagens transmitidas na rede, mantendo uma taxa de entrega elevada. Para tal, o DEBA selecciona um conjunto mínimo de vizinhos responsáveis por continuar a propagação. Na impossibilidade de obter estimativas precisas da topologia, a selecção é feita utilizando um modelo de vector local de cada nó que pode ser agregado com os dos vizinhos, gerando assim uma noção de proximidade entre os dispositivos. Os resultados da avaliação mostram que este modelo apresenta uma cobertura comparável com algoritmos de força bruta, como a inundação, mas com um número significativamente inferior de retransmissões.

Como trabalho futuro, pretendemos continuar a investigar mecanismos capazes de melhorar as taxas de entrega. Neste âmbito, serão investigadas novas funções de comparação e aprofundado o estudo dos parâmetros que caracterizam a geração dos vectores locais e o seu impacto na taxa de entrega.

## Referências

1. Al-Humoud, S.O., Mackenzie, L.M., Vanderbauwhede, W.: Dynamic counter-based broadcast in manets. In: Proc. of the 4th ACM work. on Performance monitoring and measurement of heterogeneous wireless and wired networks (PM2HW2N '09). pp. 84–88. ACM (2009)
2. Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. *Commun. ACM* 13, 422–426 (1970)
3. Durresti, A., Paruchuri, V.K., Iyengar, S.S., Kannan, R.: Optimized broadcast protocol for sensor networks. *IEEE Trans. Comput.* 54, 1013–1024 (2005)
4. Ellis, C., Miranda, H., Taïani, F.: Count on me: lightweight ad-hoc broadcasting in heterogeneous topologies. In: Proc. of the Int'l Work. on Middleware for Pervasive Mobile and Embedded Computing (M-PAC '09). pp. 1–6. ACM (2009)
5. Feeney, L.M., Nilsson, M.: Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In: Proc. of the 20th Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM 2001). vol. 3, pp. 1548–1557. IEEE (2001)
6. Garbinato, B., Holzer, A., Vessaz, F.: Six-shot broadcast: A context-aware algorithm for efficient message diffusion in MANETs. In: Proc. of the On the Move to Meaningful Internet Systems Conf. (OTM 2008). Lecture Notes in Computer Science, vol. 5331, pp. 625–638. Springer (2008)
7. Garbinato, B., Holzer, A., Vessaz, F.: Context-aware broadcasting approaches in mobile ad hoc networks. *Computer Networks* 54(7), 1210 – 1228 (2010)
8. Haas, Z.J., Halpern, J.Y., Li, L.: Gossip-based ad hoc routing. *IEEE/ACM Trans. Netw.* 14, 479–491 (2006)
9. Miranda, H., Leggio, S., Rodrigues, L., Raatikainen, K.: A power-aware broadcasting algorithm. In: Proc. of The 17th Annual IEEE Int'l Symp. on Personal, Indoor and Mobile Radio Communications (PIMRC'06) (2006)
10. Pham, N.D., Choo, H.: Efficient location-aided flooding scheme initiated by receiver for manets. In: Proc. of the 3rd Int'l Conf. on Ubiquitous Information Management and Communication (ICUIMC '09). pp. 194–199. ACM (2009)
11. Sun, M.T., Lai, T.H.: Location aided broadcast in wireless ad hoc network systems. In: Proc. of Wireless Communications and Networking Conf. (WCNC 2002). pp. 597—602 (2002)
12. Tseng, Y.C., Ni, S.Y., Chen, Y.S., Sheu, J.P.: The broadcast storm problem in a mobile ad hoc network. *Wireless Networks* 8, 153–167 (2002)