

Replicação de dados em redes Ad Hoc

Gabriel Oliveira, Antonio Loureiro e Mario Campos

¹Universidade Federal de Minas Gerais, MG, Brazil

Redes móveis ad hoc recentemente vêm atraindo interesse da comunidade científica por sua característica de comunicação não centralizada, comunicação direta entre nodos. Essas terão diversas aplicações em vídeo conferência, operações de resgate e etc. A fim de facilitar a acessibilidade dos dados usam-se técnicas de replicação de dados. Assim, propõe-se duas heurísticas para melhorar a acessibilidade de dados, através de replicação de dados, a hosts móveis em redes ad hoc. Replicação de dados nesses tipos de redes com topologias dinâmicas constitui um grande desafio. Este trabalho expõe os principais problemas relacionados à replicação de dados nas redes em questão: mobilidade dos nós, tempo de resposta e particionamento de rede. São propostas duas heurísticas para melhorar a acessibilidade de dados, através de replicação de dados nos hosts móveis. Simulações são realizadas com o objetivo de validar as abordagens.

Keywords - Redes ad hoc, replicação de dados, acessibilidade de dados

1. Introdução

Uma rede sem fio *ad hoc* é uma coleção de nodos autônomos que coletivamente formam uma rede multi-hop sem qualquer infraestrutura fixa. Essas redes podem ser aplicadas a muitos campos, incluindo a vigilância do campo de batalha, recuperação de desastres e de salvamento de emergência [Valera and Seah 2005] [Ramanathan 2005] [Deshpande and Madden 2004]. Infelizmente, devido às suas características únicas, diferentes desafios significativos de pesquisa surgem em questões de confiabilidade.

Em redes *ad hoc*, os nós movem-se livremente e desconexões ocorrem frequentemente, causando repetidas divisões da rede. Consequentemente, são necessários fundamentos tecnológicos diferentes daqueles usados em redes fixas. Por exemplo, se a rede é dividida em duas redes, devido à migração de *hosts* móveis, *hosts* em uma das duas redes não podem acessar dados armazenados em *hosts* da outra rede. Portanto, acessibilidade de dados em redes *ad hoc* é menor do que em redes convencionais fixas.

Neste tipo de rede é um assunto muito importante a prevenção de deterioração da acessibilidade de dados no ponto da divisão da rede. Uma possível solução é a replicação de itens em *hosts* móveis que não são donos do dado original.

Desta forma, replicação de dados é muito eficaz para melhorar a acessibilidade dos dados. Por outro lado, *hosts* móveis geralmente dispõe de poucos recursos e, portanto é normalmente impossível para *hosts* móveis ter a réplica de todos os itens da rede. Por exemplo, supondo uma situação onde um grupo de pesquisa que realiza uma escavação constrói uma rede *ad hoc* em uma montanha. Os resultados obtidos com a pesquisa consistem em diversos tipos de dados, tais como dados numéricos, fotografias, sons e vídeos. Neste caso, embora seja útil ter os dados que os outros

membros obtiveram, isto torna difícil e custoso (em termos de consumo de energia) para um *host* manter réplicas de todos estes dados.

Este trabalho propõe duas novas heurísticas para o problema de replicação de dados em redes *ad hoc*, a fim de melhorar sua acessibilidade. É assumido um ambiente no qual cada *host* móvel tem espaço de memória limitado para criação de réplicas. O trabalho descreve experimentos de simulação cujos resultados serviram para verificar a eficácia das heurísticas propostas.

O artigo é organizado da seguinte forma. Seção 2 trata de trabalhos relacionados. A Seção 3 trata dos problemas relacionados a replicação de dados. A Seção 4 trata do problema NP-completo mapeado. A Seção 5 trata da heurística proposta. A Seção 6 é dedicada à descrição dos testes e resultados obtidos. A Seção 7 apresenta as conclusões e trabalhos futuros a serem realizados.

2. Trabalhos relacionados

O problema de replicação de dados foi amplamente estudado no contexto de redes cabeadas. [Li and Golin 1999] aborda o problema da colocação de proxies e emprego programação dinâmica para determinar o melhor posicionamento. Eles consideram, no entanto, o caso das redes com uma única árvore topológica. Uma abordagem semelhante é usada por [Cidon and Kutten 2001]. Eles descrevem um algoritmo distribuído que resolve a atribuição de conteúdo eletrônico em uma árvore de distribuição, quando custo de armazenagem é considerada. Uma solução para o problema da colocação, o que minimiza os custos de transferência de dados para topologias de árvore, é também proposto em [Xu and D.L.Lee 2002]. O objetivo do trabalho de [Qiu and Padmanabhan 2001] é minimizar o custo para que os clientes acessem cópias das informações do servidor. Além disso, o máximo número de cópias que podem ser criados é limitado a uma taxa fixa valor.

Nos últimos anos, um grande esforço vem sendo feito em pesquisa para algoritmos de replicação de dados em redes sem-fio *ad hoc*. [Zheng and Wang 2004] propõem um algoritmo dinâmico foi proposto para reduzir o custo total de comunicação. [Nuggehalli and Chiasserini 2003] apresentam um método chamado *POACH*, para alocação de cache eficiente em termos de consumo de energia. [Huang and Chen 2006] analisam o efeito de atribuição de réplica no desempenho de sistemas baseados na mobilidade de grupos. Para lidar com a falta de ponto de acesso fixo [Sailhan and Issarny 2003] introduzem uma estratégia de gestão de cache para coleta de dados do outros pares. A abordagem visa minimizar os custos relativos a energia, especialmente para o problema de cache da Web. Os métodos propostos em [Hara 2003] [Hara and Madria 2006] consistem em muitas técnicas de alocação de réplicas e atualização de dados que podem melhorar a disponibilidade de dados para uma rede particionada.

3. Problemas relacionados à replicação de dados

Os problemas relacionados à replicação de dados em redes *ad hoc* segundo [prasanna 2006] são:

3.1. Consumo de energia

Todos os nodos em redes *ad hoc* são alimentados por baterias. Se um nó com pouca energia é replicado, mas frequentemente tem seus dados acessados, cedo sua energia será drenada e os serviços fornecidos por ele não mais serão disponíveis. Portanto algoritmos de replicação devem replicar dados em nodos com energia suficiente e periodicamente verificar a energia remanescente de cada nodo.

3.2. Mobilidade dos nós

Nós nesse tipo de rede são móveis, o que leva a uma topologia dinâmica. Assim, o algoritmo tem de dar suporte à predição de mobilidade, de tal forma que se um nó se afasta da rede, suas réplicas serão substituídas em outros nós que deverão manter-se na rede por um determinado período de tempo.

3.3. Tempo de resposta

O tempo é um fator crucial em diversas aplicações, tais como, operações de resgate e militares. Por esse motivo os dados são acessados de sua vizinhança ao invés do dono da informação, afim de melhorar o tempo de resposta.

3.4. Partição da rede

Se um nó encontra-se em uma partição e não consegue prover serviços a outros nós em diferentes partições, então os dados compartilhados têm de ser replicados em todas as partições.

3.5. Relocação de réplicas

Este tópico aborda questões relacionadas a quando, onde, quem e como as réplicas são atribuídas. Devido à topologia dinâmica, uma alocação estática de réplicas não é possível. Assim, réplicas serão realocadas dinamicamente para melhorar a disponibilidade de dados.

3.6. Gerenciamento de consistência

Se o dado compartilhado é somente para leitura, o desempenho pode ser melhorado através da sua replicação por todos os nós. Por outro lado, se uma réplica é frequentemente atualizada, outras réplicas se tornam inválidas. Assim, um mecanismo é necessário para gerenciar a consistência dos dados na rede.

4. Heurísticas Propostas

Nesta seção são descritas as duas heurísticas propostas para definir como será realizada a replicação dos dados. A primeira visa a criação de uma vizinhança topológica no qual os dados são compartilhados. A segunda heurística é uma extensão da primeira e tem como objetivo tentar minimizar o problema de partição de rede.

4.1. Primeira Heurística

A primeira heurística consiste em criar uma vizinhança topológica nos moldes de [Kohonen 2001], no qual dados semelhantes, no momento da replicação, estão topologicamente contíguos.

Algorithm *Heurística 1*

Input $G(V, E)$: Grafo de nós

Output $G(V, E)$: Grafo de nós já replicados

```
1  Inicializar Relogio
2  Inicializar TamanhoCachadosns
3  while 1 do
4      bool replicacao=false
5      if Relogio==Tempodeterminado then
6          replicacao=true
7      end if
8      se replicacao for igual a true então replica-se os dados
9      if replicacao=true then
10         for cada nó do
11             for cada nó  $\in$  vizinhança do
12                 Se o nó da vizinhança contém a informação a ser replicada.
13             end for
14             Se a informação a ser replicada já se encontrar na vizinhança, a replica não deve ser feita.
15             Senão replique os dados para um dos vizinhos diretos (escolha randômica), se o nó vizinho escolhido randômico encontrar um nó com espaço de armazenamento igual a TamanhoCachadosns, seleciona-se um novo nó vizinho, por um número de vezes determinado pelo sistema, até encontrar um nó com capacidade senão remove-se o dado do ultimo nó selecionado com a menor taxa de solicitações.
16         end for
17     end if
18 end while
end
```

4.2. Segunda Heurística

A segunda heurística consiste em um refinamento da primeira, com o objetivo de tentar solucionar o problema da *partição da rede*. Essa heurística além da replicação de sua informação pela sua vizinhança, também faz uma cópia de sua informação para uma região afastada da informação original através da distância euclidiana. Assim o nó será replicado quando tiver uma distância superior a um *Threshold* configurado pelo usuário.

Algorithm *Heurística 2*

Input $G(V, E)$: Grafo de nós

Output $G(V, E)$: Grafo de nós já replicados

```
1  Inicializar Relgio
2  Inicializar Threshold
```

```

3  Inicializar TamanhoCachadosns
4  while 1 do
5      bool replicação=false
6      if Relogio==TempoDeterminado then
7          replicação=true
8      end if
9      se replicação for igual a true então replica-se os dados
10     if replicação=true then
11         for cada nó do
12             for cada nó ∈ vizinhança do
13                 Se o nó da vizinhança contém a informação a ser replicada.
14             end for
15             Se a informação a ser replicada já se encontrar na vizinhança, a replica não deve ser feita.
16             Senão replique os dados para um dos vizinhos diretos(escolha randômica), se o nó vizinho escolhido randômico encontrar um nó com espaço de armazenamento igual a TamanhoCachadosns, seleciona-se um novo nó vizinho até encontrar um nó com capacidade senão crie espaço em um nó.

17             Faça um UNICAST a partir do nó que deseja ter sua informação replicada até que chegue em algum nó onde a distância euclidiana
                 $\sqrt{(x_{origem} - x_{nodo_t})^2 + (y_{origem} - y_{nodo_t})^2} \geq Threshold.$ 
18             Então replique o nó.
19         end for
20     end if
21 end while
end

```

5. Teste e Resultados

Os teste foram realizados através de simulações com o uso do simulador *SinalGo* [Sinalgo 2010], desenvolvido pelo grupo de computação distribuída da *ETH Zurich*. O sistema permite uma rápida prototipagem de algoritmos de rede em JAVA. Dentre as opções de simuladores disponíveis, escolheu-se o *SinalGo*, pela curva de aprendizagem muito rápida e por sua independência de plataforma. Já em relação as duas heurísticas propostas foram realizados testes com a segunda heurística proposta, uma vez que ela é uma extensão da primeira.

5.1. Configurações de simulação

Nesta seção as condições do algoritmo testado serão apresentados. O simulador tem um número de parâmetros, que definem o comportamento do sistema. As variáveis são: modo assíncrono, mobilidade, interferência, possibilidade de recebimento de mensagens durante o envio (*canReceiveWhileSending*) e possibilidade de verificação do maximo de memória disponível na rede.

A tabela 1 ilustra os valores dessas variáveis usadas nos testes.

Table 1. Tabela com a configuração das variáveis usados nos testes.

Variável	Valor
modo assíncrono	false
mobilidade	true
interferência	false
canReceiveWhileSending	true
javaVMmaxMem	120

5.2. Resultados Experimentais

Na Figura 1 encontra-se a tela de simulação do sistema proposto. Dentre as facilidades oferecidas pelo simulador estão a possibilidade de ativar uma série de características, tais como, número de nós, mobilidade dos nós, interferência, indicação do número de rounds, zoom, funções implementadas em teclas de atalho e a opção de ativar cada round manualmente ou rodar eternamente.

A Figura 2 representa os primeiros testes realizados com, respectivamente 100. Os nós em amarelo indicam nós no qual informação de replicação de dados já foi realizada com sucesso(já armazena dados de outros nós) e em rosa nós em que ocorreram colisões, o dado a ser replicado já encontrava-se em poder do nodo(já havia sido replicado para o mesmo). Azul indica a condição de estar enviado seu dado a outro nodo, durante a replicação o nodo que envia o dado a ser replicado torna-se azul e o nodo que receberá sua replica torna-se amarelo(não havendo colisão), senão torna-se rosa.

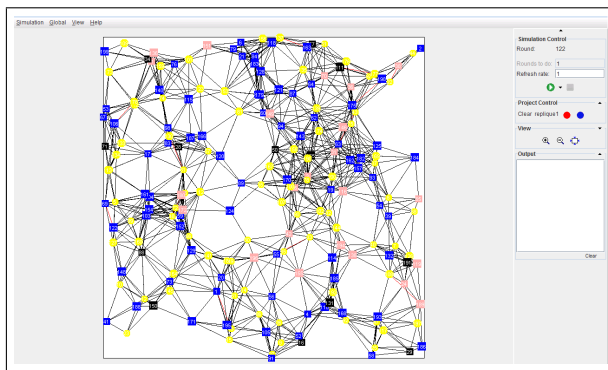


Figure 1. Tela Simulação.

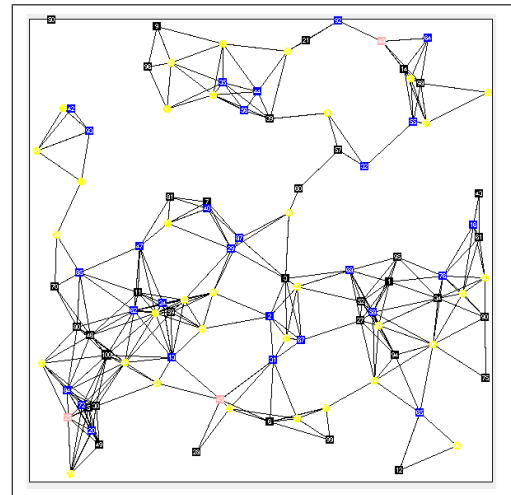


Figure 2. Primeiro teste 100 nós.

5.3. Número de Replicações

Em relação ao número de replicações nos testes realizados, gerou-se a Figura 3. Nos testes foram simulados redes por 1000 *rounds* com tamanho máximo de réplicas suportadas por cada nós de 30, 20 e 15, sendo denominado a unidade de armazenamento de *slot*.

Table 2. Tabela com estatísticas sobre o número de replicações.

Número Slots	Média	devio padrão
15	158.57	37.7
20	182.9	43
30	292	72

A figura 3 ilustra o número de réplicas efetuadas nas simulações realizadas, em azul são o número de réplicas com 30 *slots*, já em vermelho com 20 *slots* e em verde com 15 *slots*, podendo-se notar a convergência do método com o passar do tempo.

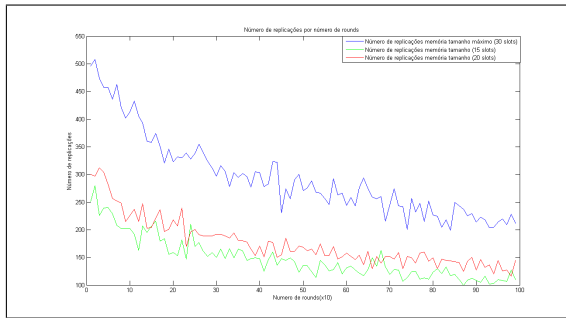


Figure 3. Número de replicações.

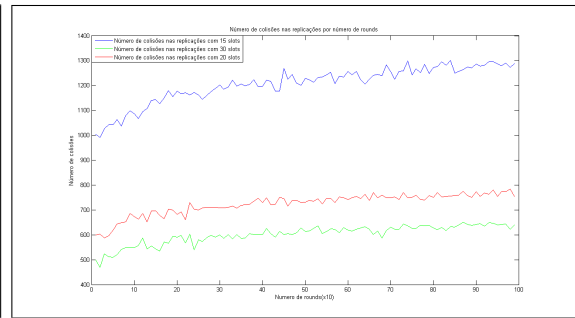


Figure 4. Número de colisões.

A tabela 2 ilustra os valores médios do número de replicações e seus desvios padrões.

5.4. Número de colisões

Em relação às colisões ocorridos nos teste, traduz-se por ações de replicação que encontram os dados a serem replicados já replicados, figura 4.

Nos testes foram simulados redes por 1000 *rounds* com tamanho máximo de réplicas suportadas por cada nós de 30, 20 e 15. A Figura 4 ilustra o número de colisões nas simulações realizadas, em verde são o número de colisões com 30 *slots*, já em vermelho com 20 *slots* e em azul com 15 *slots*, podendo-se notar o aumento do número de colisões com o passar do tempo, tendo como reflexo a distribuição dos dados pela rede com o passar do tempo.

6. Conclusão e trabalhos futuros

As redes sem fio *ad hoc* são vistas com conjuntos de dispositivos com computação e capacidades de comunicação limitados. O principal desafio para as aplicações nessas redes é como lidar com a escassez de recursos e com sua alta mobilidade. Neste trabalho, foram descritos esforços realizados com o intuito de aumentar a acessibilidade dos dados para comunicação de dados em redes sem fio *ad hoc*. Para isso, foram propostas duas heurísticas, sendo a segunda um refinamento da primeira.

As heurísticas propostas consideram os seguintes parâmetros: tempo de resposta, mobilidade dos nós e partição da rede. Resultados obtidos através de simulação validaram a proposta e constatarem sua eficácia.

Dentre os trabalhos futuros destaca-se o aperfeiçoamento da heurística proposta a fim de atacar todos os problemas relacionados.

References

- Cidon, I. and Kutten, S. (2001). Optimal allocation of electronic content. *IEEE INFOCOM*.
- Deshpande, A. and Madden, S. R. (2004). Modeldriven data acquisition in sensor networks. *Proc. Int. Conf. Very Large Database(VLDB), toronto, Canada*.
- Hara, T. (2003). Replica allocation methods in ad hoc networks with data update. *ACM/Baltzer Mobile Newt. Appl*, pages 343–354.
- Hara, T. and Madria, K. (2006). Data replication for improving data accessibility in ad hoc networks. *IEEE Trans. Mobile Comput*, pages 1515–1532.
- Huang, J. and Chen, M. (2006). On the effect of mobility group to data replication in ad hoc networks. *IEEE Trans. Mobile Comput*, pages 492–507.
- Kohonen, T. (2001). *Self-Organizing Maps*. Springer-Verlag New York, Secaucus, NJ, USA.
- Li, B. and Golin, G. (1999). On the optimal placement of web proxies in the internet. *IEEE INFOCOM*.
- Nuggehalli, P. and Chiasserini (2003). Energy efficient caching strategies in ad hoc wireless networks. *Proc. ACM MobiHoc*.
- prasanna, D. (2006). Managing data replication in mobile ad hoc networks database. *International Conference on Collaborative Computing*, pages 1–10.
- Qiu, L. and Padmanabhan, V. (2001). On the placement of web server replicas. *IEEE INFOCOM*, pages 1587–1596.
- Ramanathan, R. (2005). Challenges: a radically new architecture for next generation mobile ad hoc networks. *Proc. ACM MobiCom*, pages 132–139.
- Sailhan, F. and Issarny, V. (2003). Cooperative caching in ad hoc networks. *Proc. Int. conf. Mobile Data Management(MDM)*, pages 13–28.
- Sinalgo (2010). Sinalgo. <http://disco.ethz.ch/projects/sinalgo/index.html>. Simulation framework for testing and validating network algorithms.
- Valera, C. and Seah, W. (2005). Some related article protocol robustness in ad hoc networks through cooperative packet caching and shortest multipath routing. *IEEE Trans. Mobile Comput*, pages 443–457.
- Xu, J. and D.L.Lee (2002). Placement problems for transparent data replication proxy services. *IEEE Journal on Selected Areas in communications*, pages 1383–1398.
- Zheng, J. and Wang, Y. (2004). A dynamic adaptive replica allocation algorithm in mobile ad hoc networks. *Proc. IEEE PerCom*, pages 65–69.