

# Resolvendo o problema de *snapshot* em redes DTN utilizando algoritmos distribuídos

Maurício José da Silva, Ricardo Augusto Rabelo Oliveira  
PPGCC - Programa de Pós-Graduação em Ciência da Computação  
UFOP - Universidade Federal de Ouro Preto  
Ouro Preto, Minas Gerais, Brasil  
email: badriciobq@gmail.com, rrabelo@gmail.com

**Resumo**—Quando falamos em transmissão de dados utilizando uma rede, esperamos que exista um caminho entre a origem e o destino para que a comunicação seja viável. Esta é uma condição para que seja possível utilizar o protocolo de redes TCP/IP. Mas quando pensamos em alguns cenários no mundo real, como por exemplo uma rede interplanetária, não podemos garantir que todos os dispositivos da rede estarão disponíveis a todo momento. Para resolver este problema surgiu uma nova arquitetura de redes conhecida com DTN (*Delay Tolerant Networks*). Neste tipo de rede geralmente os dispositivos não estão disponíveis durante todo o tempo, sendo necessário a utilização de algumas técnicas diferentes para alcançar a origem. Este artigo faz uma abstração desta rede para um grafo temporal, e propõe uma solução em algoritmos distribuídos para resolver o problema de *snapshot* em grafos, visando garantir que a mensagem alcance seu destino mesmo em uma rede que mude a todo momento e que o tempo de propagação desta mudança seja lento.

**Keywords**-grafos, algoritmos, complexidade, *snapshot*, *Delay Tolerant Network*.

## I. INTRODUÇÃO

Sistemas de memória distribuída são sistemas em que múltiplos processadores são interconectados por um canal que permite a comunicação ponto-a-ponto, nos quais os processadores não compartilham fisicamente qualquer tipo de memória. A falta de uma região de memória compartilhada exige que a comunicação entre os processadores seja feita através da troca de mensagens. Como os recursos computacionais são limitados, neste caso, além do limite de memória e de tempo que são exigidos naturalmente por qualquer algoritmo, temos que levar em consideração também o custo que o algoritmo distribuído terá para realizar a troca de mensagens entre os processadores [1].

Em um grafo  $G$  com  $n$  vértices  $V_i$  conectados por  $m$  arestas  $A_j$  onde os nós mudam de posição a todo momento e o tempo de propagação da mudança seja lento, seria necessário construir a árvore geradora do grafo depois de cada mudança para conhecer sua configuração, possibilitando assim a transmissão de informações de forma direta e com o menor custo possível. Ou seja, as mensagens passariam somente pelos nós que estivessem entre a origem e o destino, evitando que a rede fique sobrecarregada com excesso de mensagens.

Outra possibilidade, um pouco mais simplista, seria enviar as mensagens por inundação, onde cada nó que receber a mensagem, a encaminha para todos os seus vizinhos, caso esta não esteja endereçada a ele.

Naturalmente, para que cada vértice  $V_i$  conheça os  $V_{n-1}$  vértices do grafo, é necessário que cada vértice  $V_i$  receba uma mensagem de cada um dos  $V_{n-1}$  vértices do grafo, ou seja,  $k$  mensagens são necessárias para se conhecer o estado global, ou *snapshot*, do grafo. Desta forma, o número de mensagens que trafegarão pelo grafo atingirá proporções exponenciais e sua complexidade é expressa pela fórmula  $k = n.m$ , onde,  $n$  é o número de vértices e  $m$  é o número de arestas.

Aplicando um algoritmo distribuído para resolver o problema do *snapshot*, conseguimos dividir o processamento para os  $V_n$  vértice do grafo possibilitando um processamento mais robusto, e focamos a análise de complexidade somente no algoritmo, levando em consideração não somente a análise de tempo e espaço mas também a análise de troca de mensagens  $k$ .

Ao trabalharmos com a possibilidade do grafo  $G$  não ser conexo durante todo o tempo, tendo pelo menos um vértice  $V_i$  sem nenhuma aresta  $A_j$  que ligue  $V_i$  a  $G$ , uma alternativa de alcançar  $V_i$  em um determinado tempo, considerando que qualquer  $V_i$  será conexo a  $G$  em algum instante de tempo qualquer (grafo temporal) seria a utilização de protocolos sob a arquitetura DTN (*Delay tolerant Network*). Para termos uma visão prática sobre o cenário descrito acima podemos pensar em uma rede interplanetária, que é descrito em maiores detalhes em [3].

Apesar de redes DTN terem sido projetadas com foco em redes interplanetárias, eles têm uma boa aplicação em redes para dispositivos móveis onde as propriedades do TCP/IP geralmente não são satisfeitas, pois os limites de armazenamento são muito restritos e os dispositivos não estão conectados durante todo o tempo. Em [4] é mostrado que é viável a utilização de redes DTN neste ambiente, utilizando as tecnologias de comunicação que são *bluetooth*, Wi-Fi, 3G e interfaces de rádio.

As justificativas para utilizarmos redes DTN são apresentadas na Seção II, para que sejam atingidos os objetivos apresentados na Seção III foi aplicada a metodologia que

pode ser vista na Seção IV. Finalmente a Seção V apresentam os frutos desejados com a realização da pesquisa.

## II. JUSTIFICATIVAS

As redes de computadores utilizam como padrão a pilha de protocolos TCP/IP, que funciona perfeitamente bem para o ambiente pelo qual foi projetada. Porém, quando saímos de suas condições ideais de uso, como por exemplo, redes sem fio entre dispositivos móveis onde se tem restrição de consumo de energia e falta de garantia de conectividade entre o destino e a origem, é necessário utilizarmos outras técnicas de transmissão de dados, o que deu origem a uma nova arquitetura de transmissão, conhecida como DTN.

Esta nova arquitetura de redes prevê a utilização de uma técnica de comutação e de armazenamento persistente de dados em uma nova camada que fica logo abaixo da camada de aplicação, que é denominada camada de agregação ou em inglês *Bundle Layer*. Esta camada é responsável de certa forma por isolar as camadas inferiores da camada de aplicação garantindo interoperabilidade de dispositivos de rede. Redes que utilizam arquitetura DTN tem as seguintes características:

- Atrasos longos: Uma DTN não possui tempo estimado para o pacote chegar ao seu destino.
- Frequentes desconexões: As desconexões podem ocorrer pela mobilidade, que provoca constantes mudanças na topologia da rede.
- Capacidade de Armazenamento: A principal característica das DTNs é a propriedade de “armazenar e enviar” do inglês “*store-and-forward*”. Este processo de transmissão é exemplificado na Figura 1

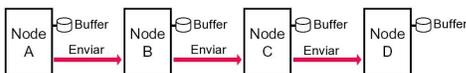


Figura 1. Exemplo de *Store-and-forward* de uma DTN

Levando em consideração que o tempo para a rede se estabilizar é lento e a rota se modifica com muita frequência (características presentes em redes DTN), temos o cenário ideal para aplicarmos um algoritmo distribuído com a finalidade de conhecer o estado global do grafo, pois, para que se chegue ao estado global do grafo é necessário que haja troca de mensagens entre todos os nós da rede. Assim podemos analisar o impacto causado pela ausência de um nó em uma rede DTN, comparando seus resultados com a troca de mensagens para se chegar ao estado global de um grafo onde todos os nós são conexos durante todo o tempo.

As redes DNT tem ampla aplicabilidade em cenários reais, isto se dá pelo fato de que tem se tornado necessário estarmos conectados a uma rede em grande parte do tempo. Como exemplo, temos as redes de dispositivos móveis (celulares, *smartphones*, *ipads* e diversos outros dispositivos

que têm sistema embarcado), além de redes militares, redes interplanetárias e sistemas de rastreamento de animais selvagens.

## III. OBJETIVOS

A partir da análise de complexidade, espera-se implementar algoritmos que se aproximem do limite inferior do problema e provar por meio de simulação que o algoritmo proposto atinge os objetivos esperados, contribuindo para a comunidade disponibilizando os experimentos para posterior análise, e revisando os conceitos estudados na disciplina Projeto e Análise de Algoritmos visando integrá-los futuramente à pesquisa de mestrado.

## IV. METODOLOGIA

Imaginando um cenário onde o algoritmo distribuído faria a troca de mensagens para determinar um estado global do grafo. Estas mensagens seriam enviadas de forma assíncrona, pois não se pode estabelecer limites para a troca destas mensagens.

Descrevendo o cenário a ser analisado, temos um algoritmo distribuído e assíncrono que realiza a troca de mensagens entre os nós gerando eventos, até atingir o estado final e chegar ao estado global do grafo.

Para a realização das simulações foi utilizado o simulador DAJ [5], que é uma ferramenta visual e iterativa para o estudo de algoritmos distribuídos, pois deve-se apresentar explicitamente cada passo da execução do algoritmo e o DAJ se encarrega de exibir o estado de sua execução. Como algoritmos distribuídos geralmente são de difícil compreensão, o DAJ ajuda no estudo destes algoritmos permitindo a criação de *logs* e a reprodução do cenário até que os conceitos sejam compreendidos.

## V. RESULTADOS ESPERADOS

Como resultado esperamos analisar a complexidade dos algoritmos implementados comparando as informações com os resultados obtidos por meio de um ambiente de simulação e estudando o impacto se aplicado em um problema do mundo real. Espera-se propor melhorias em algum protocolo existente ou até mesmo substituir alguma técnica utilizada atualmente.

## REFERÊNCIAS

- [1] C. V. Barbosa, *An Introduction to Distributed Algorithms*, 1st ed. The MIT Press, 1996, ISBN-10: 0262024128.
- [2] S. Burleigh, “Delay-tolerant networking: An approach to interplanetary internet,” *IEEE Communications Magazine*, 2003.
- [3] C. Caini, P. Cornice, R. Firrincieli, and M. Livini, “Dtn meets smartphones: future prospects and tests,” *International Symposium on Wireless Pervasive Computing (ISWPC)*, 2010.
- [4] W. Schreiner, *DAJ - A Toolkit for the Simulation of Distributed Algorithms in Java*, Research Institute for Symbolic Computation (RISC-Linz), <ftp://ftp.risc.unilinz.ac.at/pub/parlab/daj/report/report-main.ps.gz>, October 1998.