

# Busca Exaustiva em Redes P2P

Péricles C. M. Lopes<sup>1</sup>, Ronaldo A. Ferreira<sup>1</sup>

<sup>1</sup>Faculdade de Computação – Universidade Federal de Mato Grosso do Sul (UFMS)  
Caixa Postal 549 – 79.070-900 – Campo Grande – MS – Brasil

pericles@facom.ufms.br, raf@facom.ufms.br

**Abstract.** *Despite numerous efforts in the past few years, efficient complex queries in large-scale P2P networks remain an open and challenging problem. This work presents a hybrid architecture for complex queries in P2P networks that relies on a lightweight network structure, in a predominantly non-structured network, to avoid unnecessary query replication and to speed up query propagation in unstructured P2P networks. In addition to this architecture, this work presents a controlled and exhaustive search protocol that combines structured and non-structured P2P networks features. Extensive simulation results, considering static and dynamic scenarios, show that the proposed architecture, in conjunction with a search protocol, outperforms the best known solution in number of messages, response time, number of hops, and success rate for query resolution. Furthermore, this work shows, analytically, using random trees results, an upper bound on query routing.*

**Palavras-Chave:** *P2P Systems, Distributed Algorithms, Analysis, Simulation*

**Resumo.** *Apesar de inúmeros esforços nos últimos anos, buscas complexas em redes P2P de grande escala permanecem um problema em aberto e desafiador. Este trabalho propõe uma arquitetura híbrida para buscas complexas em redes P2P que utiliza uma leve estrutura, em uma rede predominantemente não estruturada, para evitar replicações desnecessárias e acelerar a propagação de mensagens de busca. Além da arquitetura, também é proposto um protocolo eficiente de busca que combina características de redes P2P estruturadas e não-estruturadas. Resultados extensivos de simulação, considerando cenários estáticos e dinâmicos, mostram que a arquitetura proposta, em conjunto com o protocolo de busca, possui desempenho superior à melhor solução até então conhecida em termos de número de mensagens, tempo de resposta, número de saltos em respostas positivas e taxa de sucesso. Além disso, o trabalho mostra analiticamente, utilizando resultados de árvores aleatórias, o limite superior teórico para o número máximo de pares que uma mensagem de busca deve percorrer para resolver uma consulta.*

**Palavras-Chave:** *Sistemas P2P, Algoritmos Distribuídos, Análise, Simulação*

## 1. Introdução

Nos últimos anos, sistemas par-a-par (P2P - *peer-to-peer*) se tornaram um poderoso paradigma de redes que permite o compartilhamento de vários recursos da Internet de maneira totalmente distribuída. Mais importante, este novo paradigma deu vida a várias outras novas aplicações, tais como mensagens instantâneas, voz sobre IP (Skype), distribuição de vídeos, computação distribuída (SETI@Home) e muitas outras. Além disso, usuários finais se transformaram em grandes produtores de informações que necessitam ser compartilhadas com o resto do mundo. Para manter essas fontes de informações, os usuários ainda necessitam de infraestruturas centralizadas, normalmente mantidas por empresas privadas. Uma solução P2P descentralizada para esse problema ainda não é possível porque, apesar de inúmeros esforços, buscas complexas em redes P2P ainda podem ser consideradas um problema em aberto e desafiador.

O problema de busca em redes P2P estruturadas foi resolvido de forma elegante e eficiente utilizando-se tabelas de dispersão distribuídas (DHT - *Distributed Hash Table*) [Stoica et al. 2001]. Entretanto, a semântica da busca é muito limitada, já que um usuário deve saber o nome exato do objeto para determinar o par responsável por manter a réplica do objeto ou sua referência.

Buscas exaustivas foram introduzidas recentemente para redes P2P [Ferreira et al. 2005, Terpstra et al. 2007] como um método de busca efetivo e confiável. Nesse tipo de busca a semântica é bem mais rica, em comparação com as buscas baseadas em *hashing*, já que algoritmos sofisticados podem ser executados localmente em cada par. Entretanto, as soluções mais eficientes se baseiam em caminhadas aleatórias [Ferreira et al. 2005], ou suas variantes com vários fatores de ramificação [Terpstra et al. 2007], para prover garantias probabilísticas. Apesar da eficiência em algumas aplicações, sabe-se que caminhada aleatória é um método de busca “cego” que pode visitar o mesmo par mais de uma vez, gerando sobrecarga desnecessária, e que não garante a cobertura da rede.

Neste trabalho, com o objetivo de se explorar as melhores características tanto do modelo estruturado como do modelo não estruturado, é proposto e avaliado SplitQuest. Em SplitQuest, os pares possuem identificadores únicos e são posicionados uniformemente em um anel virtual. Além dessa leve estrutura, os pares estabelecem conexões aleatórias entre si, como em uma rede P2P não-estruturada. Essa heterogeneidade das conexões dos pares é explorada de forma inteligente por SplitQuest, que propaga de forma agressiva as mensagens de busca pelos pares com grande número de conexões. Além disso, a estrutura simples e a identificação dos pares permitem que SplitQuest direcione as mensagens de busca para partes específicas da rede, evitando a sobrecarga gerada pela duplicação de mensagens.

SplitQuest foi simulado em diferentes topologias geradas por dados reais e sintéticos, com número variável de pares, em cenários estáticos e dinâmicos e apresentou desempenho superior à melhor solução conhecida para busca exaustiva em relação a tempo de resposta, taxa de sucesso e número de mensagens. O trabalho ainda apresenta uma análise matemática, baseada em resultados de árvores aleatórias, que mostra o limite superior para o número máximo de saltos que uma mensagem de busca é propagada pela rede.

As principais contribuições deste trabalho são: (a) um protocolo de busca exaustiva eficiente que explora a heterogeneidade dos pares para distribuir rapidamente mensagens de busca em redes P2P, evita duplicações desnecessárias de mensagens na rede e atinge altas taxas de recuperação de informação (*recall*); (b) uma análise matemática que prova um limite superior no número máximo de saltos que uma mensagem de busca pode propagar; (c) um novo algoritmo de entrada de pares em uma rede P2P que distribui uniformemente os pares em um anel virtual; (d) resultados extensivos de simulação que comparam SplitQuest favoravelmente com BubbleStorm [Terpstra et al. 2007]—a melhor solução conhecida para buscas exaustivas em redes P2P—em importantes métricas, como número de mensagens, taxa de sucesso e tempo de resposta.

Como resultado deste trabalho, foram publicados os artigos [Lopes and Ferreira 2010a] e [Lopes and Ferreira 2010b] e submetido o artigo [Lopes and Ferreira 2011]. Cabe salientar que o IPTPS é considerado um dos eventos de maior impacto na área, com Comitê de Programa de altíssimo nível e classificação A2 no Qualis da Capes. O SBRC também é um evento bastante seletivo e é considerado o fórum mais importante das áreas de redes e sistemas distribuídos do Brasil.

## 2. O Protocolo SplitQuest

SplitQuest evita a sobrecarga desnecessária de métodos de busca baseados em caminhadas aleatórias (e suas variações) com uma estrutura híbrida que permite que as mensagens de busca

sejam direcionadas para partes específicas e distintas da rede. Em adição às conexões aleatórias, normalmente presentes em redes não estruturadas, SplitQuest dispõe os pares em um anel, ou seja, cada par possui um predecessor e um sucessor identificados com valores no intervalo  $[0, 1]$ .

## 2.1. Replicação de Índices

A replicação de índices em SplitQuest é realizada em um conjunto de pares localizados em um subintervalo contíguo do intervalo  $[0, 1]$ . Quando pares são distribuídos de forma uniforme por todo intervalo, o número esperado de pares em um subintervalo contíguo é proporcional ao tamanho do subintervalo. Conseqüentemente, o problema básico neste caso é determinar o tamanho do subintervalo, ou, mais especificamente, o tamanho do grupo de replicação que minimiza a replicação de índices e mensagens de busca. Quando o valor de  $n$  é conhecido, o tamanho do subintervalo com número esperado de pares igual a  $d$  é dado por  $\frac{d}{n}$ . Seja então  $d$  o tamanho do grupo e  $q$  o número de grupos na rede. O número total de mensagens de replicação e busca de um objeto,  $M = q + d$ , é dado por:  $M = \frac{n}{d} + d$ . É fácil verificar que  $d$  igual a  $\sqrt{n}$  minimiza  $M$ . A Figura 1 ilustra como os pares são organizados em grupos de replicação no intervalo  $[0, 1]$ .

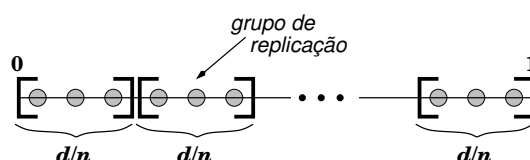


Figura 1. Grupos de replicação em SplitQuest.

## 2.2. Algoritmo de Busca

SplitQuest utiliza dois artifícios para garantir cobertura de rede e evitar duplicação de mensagens: busca por intervalo e atalhos para grupos. Busca por intervalo significa que cada mensagem de busca contém, além da consulta em si, um intervalo  $[X, Y]$  que a busca deve cobrir. Quando um par  $i$  recebe uma mensagem com o intervalo  $[X, Y]$ , ele verifica suas conexões e determina quais vizinhos estão contidos no intervalo recebido. Esses vizinhos são possíveis candidatos para o repasse dessa mensagem de busca. Em seguida, o par  $i$  calcula os identificadores de grupos desses possíveis candidatos e mantém apenas um vizinho para cada grupo. Dependendo dos grupos selecionados, o par  $i$  decompõe o intervalo inicial em subintervalos e envia mensagens de busca contendo esses subintervalos para cada um dos vizinhos selecionados.

A Figura 2 mostra um exemplo simples de propagação de uma mensagem de busca. Apenas um pequeno conjunto de pares e conexões é mostrado. A busca é iniciada em um par **a** que está conectado aos pares **e**, **b** e **g**. O par **a** é inicialmente responsável por cobrir todo o intervalo de  $[0, 1]$ . Assim, este par cria três mensagens com intervalos  $[0, x_4]$ ,  $[x_5, x_8]$  e  $[x_8, 1]$  e as envia para os pares **b**, **e** e **g** respectivamente. O par **b** envia uma mensagem para **c** com intervalo  $[0, x_2]$  e o par **e** envia uma mensagem para **f** com intervalo  $[x_6, x_8]$ . Finalmente, o par **c** envia uma mensagem para **d** com intervalo  $[0, x_1]$  e os pares **b** e **f** são responsáveis por realizar a cobertura dos subintervalos restantes ( $[x_2, x_3]$  e  $[x_6, x_7]$ ).

Outro artifício utilizado por SplitQuest para melhorar o desempenho da busca são os atalhos para grupos. Os pares constroem atalhos (conexões diretas para outros pares) para os dois grupos adjacentes aos grupos a que eles pertencem. Esses atalhos garantem que sempre existe um caminho para um grupo, ou seja, um caminho que a cada novo salto levará a mensagem de busca a cobrir um novo grupo, evitando que uma consulta seja realizada várias vezes em um mesmo grupo.

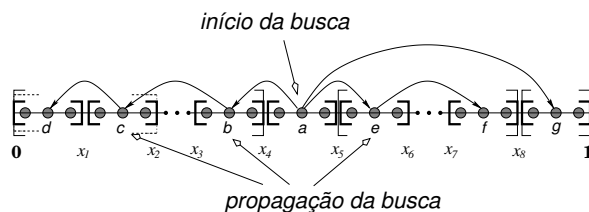


Figura 2. Propagação de uma mensagem de busca em SplitQuest.

### 2.3. Atribuição de Identificadores

Uma questão-chave em SplitQuest é como os identificadores dos pares são gerados. Os identificadores devem ser distribuídos uniformemente no espaço de identificadores para se garantir tamanhos uniformes de grupo. Além disso, os pares não devem depender de um protocolo complexo de roteamento para entrar na rede, como é o caso de sistemas DHT que utilizam funções de dispersão para geração dos identificadores e um protocolo de entrada custoso.

O algoritmo de atribuição de identificadores de SplitQuest se baseia em resultados obtidos por [Naor and Wieder 2007] e também pela extrapolação do “poder de duas escolhas” [Mitzenmacher 2001] para  $k$ -escolhas. Nesse novo algoritmo, o par que deseja se conectar à rede requisita a um *proxy* (um par que já está na rede) que sejam selecionados  $k$  intervalos da rede de maneira aleatória e uniforme. Mais especificamente, o *proxy* seleciona  $k$  pares e os maiores intervalos adjacentes a esses. O identificador do par entrante é o ponto médio do maior segmento. Para se ter uma boa distribuição uniforme, o valor de  $k$  deve ser proporcional a  $\log n$ . A Figura 3(a) mostra a eficiência do algoritmo proposto em distribuir uniformemente os identificadores no intervalo  $[0, 1]$  em comparação com a atribuição de identificadores por funções de dispersão.

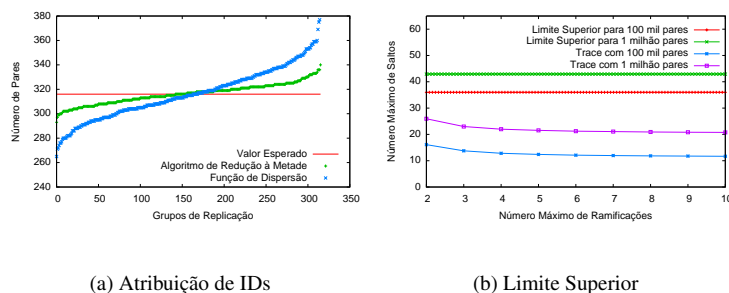
### 2.4. Análise do Protocolo

A estrutura formada pelos pares e suas conexões em SplitQuest pode ser vista como um grafo aleatório. Se forem consideradas as conexões de grupos, montadas como conexões aleatórias durante a busca, pode-se modelar a propagação de mensagens de busca para os diferentes grupos como uma difusão (*broadcast*) em uma árvore aleatória. Em cada passo do algoritmo, os grupos são particionados em função das conexões do par que recebeu a mensagem de busca e do intervalo que este deve cobrir.

Quando uma busca se inicia em um par  $i$ , as mensagens de busca devem visitar um conjunto  $G$  de grupos disjuntos, inicialmente  $G$  é igual ao número total de grupos. O conjunto é dividido em  $R$  subgrupos, que dependem das conexões de  $i$ , com distribuição dada por  $P(R = r) = p_r$ , em que  $p_r$  é a distribuição de probabilidade em  $\{1, 2, \dots\}$ . Condicionado ao evento  $\{R = r\}$ , para  $1 \leq k \leq r$ , um grupo está no  $k$ -ésimo subconjunto com probabilidade  $V_{k,r}$ , em que  $V_r = (V_{k,r}; 1 \leq k \leq r)$  é um vetor de probabilidade em  $\{1, \dots, r\}$ . Se  $N_k$  é a cardinalidade do  $k$ -ésimo subconjunto, então, condicionado ao evento  $\{R = r\}$  e às variáveis aleatórias  $V_{1,r}, \dots, V_{r,r}$ , a distribuição do vetor  $(N_1, \dots, N_r)$  é multinomial com parâmetros  $G-1$  e  $(V_{1,r}, \dots, V_{r,r})$  [Mohamed and Robert 2005].

A análise da altura ( $H_G$ ) da árvore aleatória para uma distribuição geral de  $R$  é difícil e é geralmente obtida por técnicas de análise complexa. Um limite superior mais conservador pode ser obtido quando  $r$  pertence ao conjunto  $\{1, 2\}$ , o que significa que temos no máximo duas ramificações em cada passo da difusão. Devroye [Devroye 1998] mostra que, quando  $r$  pertence a  $\{1, 2\}$ , a altura da árvore é limitada por  $O(\log G)$ , mais especificamente Devroye mostra que  $\lim_{G \rightarrow \infty} P\{H_G > c \log G\} = 0$ , em que  $c$  é uma constante. Um resultado mais forte é derivado em [Devroye 1986], em que é mostrado que  $H_G / \log G \rightarrow 4.31106 \dots$  em probabilidade.

É fácil verificar que o limite superior derivado acima é também um limite superior para o maior número de saltos que uma mensagem de busca é propagada em SplitQuest. Como SplitQuest pode enviar uma mensagem de busca para mais de dois grupos, o limite superior é, de fato, bastante conservador. A Figura 3(b) mostra o número máximo de saltos que uma mensagem de busca percorre em redes com 100 mil e 1 milhão de pares e quando o número máximo de ramificações é incrementado de 2 a 10. A demonstração completa do limite superior pode ser encontrada na Seção 3.5 da dissertação.



(a) Atribuição de IDs

(b) Limite Superior

Figura 3. a) Número de pares em grupos de replicação para tamanho de rede de 100 mil pares. b) Distância máxima de propagação de uma mensagem de busca com diferentes ramificações. As linhas superiores são limites teóricos para 1 milhão e 100 mil pares respectivamente.

### 3. Resultados

Para avaliar SplitQuest, foi desenvolvido um simulador em C++ e realizados vários experimentos para comparar seu desempenho com a melhor solução conhecida até então [Terpstra et al. 2007]. Foram avaliadas três métricas: número de mensagens, taxa de sucesso e latência, em três topologias de rede distintas: regular, power-law e *trace* real (Gnutella) em cenários estáticos e dinâmicos. Em todas as figuras, a coordenada  $x$  representa diferentes tamanhos de rede e a coordenada  $y$  uma medida específica computada como sendo a média de 11 simulações independentes. Para todos os resultados, foram calculados os intervalos de confiança com grau de confiança de 95%.

A Figura 4(a) mostra os resultados obtidos para o número de mensagens quando 5000 documentos são inseridos na rede. O número total de mensagens presentes corresponde à soma dos dados e de mensagens de busca durante toda simulação. É possível observar que, em todas as três topologias e tamanhos de rede, o número de mensagens em SplitQuest é o mesmo (linha única) já que SplitQuest utiliza apenas o tamanho da rede como parâmetro para determinar o número de mensagens inseridas na rede. Dessa forma, esse número de mensagens é independente de topologia e distribuição do grau dos nós dos pares. Além disso seu desempenho com relação ao número de mensagens inseridas na rede superou o BubbleStorm em todos os cenários.

A taxa de sucesso em um cenário dinâmico pode ser vista na Figura 4(b). Pode-se observar que Splitquest atinge altas taxas de sucesso mesmo em cenários dinâmicos. Como SplitQuest direciona a busca para pares não visitados, ele apresenta resultado bem superior a BubbleStorm que utiliza caminhada aleatória.

Outro resultado importante é a baixa latência na resolução de consultas como mostrado na Figura 4(c). Esse resultado se deve à exploração agressiva da heterogeneidade dos pares que faz com que as mensagens sejam distribuídas pela rede de maneira muito mais rápida.

### 4. Conclusão

Este trabalho apresentou SplitQuest, um novo protocolo de busca para redes P2P que se baseia em uma estrutura híbrida para se evitar replicações desnecessárias de mensagens de busca e para

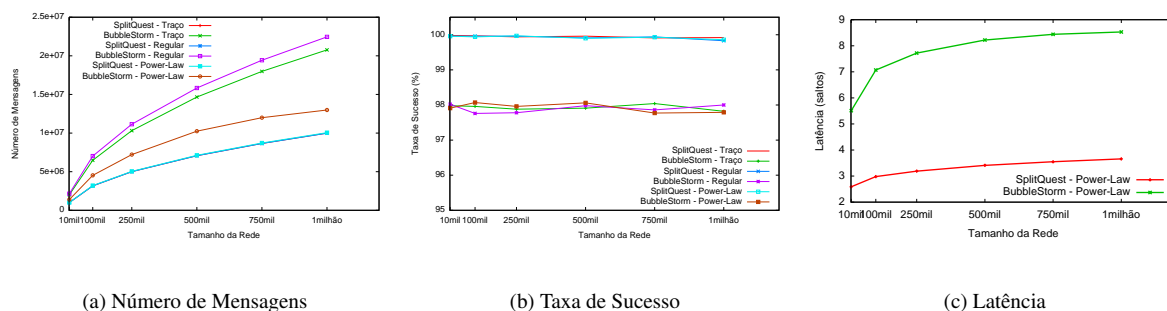


Figura 4. Resultados.

garantir cobertura da rede. Resultados numéricos e análise matemática mostram que SplitQuest é capaz de buscar informações em uma rede P2P de maneira eficiente e que, dessa forma, possui potencial para estimular o desenvolvimento de novas aplicações distribuídas, em particular aplicações que publicam baixas quantidades de dados, tais como microblogs e wikis.

## Referências

- Devroye, L. (1986). A Note on the Height of Binary Search Trees. *Journal of ACM*, 33(3):489–498.
- Devroye, L. (1998). Universal Limit Laws for Depths in Random Trees. *SIAM Journal on Computing*, 28(2):409–432.
- Ferreira, R. A., Ramanathan, M. K., Awan, A., Grama, A., and Jagannathan, S. (2005). Search with Probabilistic Guarantees in Unstructured Peer-to-Peer Networks. In *Proceedings of the 2005 IEEE P2P (P2P'05)*, pages 165–172, Konstanz, Alemanha.
- Lopes, P. and Ferreira, R. A. (2010a). SplitQuest: Controlled and Exhaustive Search in Peer-to-Peer Networks. In *Proceedings of the 2010 Usenix IPTPS (IPTPS'10)*, San Jose, CA.
- Lopes, P. and Ferreira, R. A. (2010b). Uma Arquitetura Híbrida para Buscas Complexas em Redes P2P. In *Anais do SBRC 2010*, Gramado, RS.
- Lopes, P. and Ferreira, R. A. (2011). Exhaustive Search in Peer-to-Peer Networks. *Journal of Parallel and Distributed Computing (JPDC)* - submetido, em avaliação.
- Mitzenmacher, M. (2001). The Power of Two Choices in Randomized Load Balancing. *IEEE Trans. Parallel Distrib. Syst.*, 12(10):1094–1104.
- Mohamed, H. and Robert, P. (2005). A Probabilistic Analysis of Some Tree Algorithms. *The Annals of Applied Probability*, 15(4):2445–2471.
- Naor, M. and Wieder, U. (2007). Novel Architectures for P2P Applications: The Continuous-Discrete Approach. *ACM Transactions on Algorithms (Electronic Edition)*, 3(3).
- Stoica, I., Morris, R., Karger, D., Kaashoek, F., and Balakrishnan, H. (2001). Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. In *Proceedings of the 2001 ACM SIGCOMM (SIGCOMM'01)*, pages 149–160, San Diego, CA.
- Terpstra, W. W., Kangasharju, J., Leng, C., and Buchmann, A. P. (2007). BubbleStorm: Resilient, Probabilistic, and Exhaustive Peer-to-Peer Search. In *Proceedings of the 2007 ACM SIGCOMM (SIGCOMM'07)*, pages 49–60, Tóquio, Japão.