Lightweight Super-Peers: Um modelo de Super-Peers para Redes DHT

Marcos Madruga, Thaís Batista, Rodrigo Araújo, Luiz Affonso Guedes

Departamento de Informática e Matemática Aplicada (DIMAp)
Universidade Federal do Rio Grande do Norte (UFRN)
Campus Universitário – Lagoa Nova – 59.072-970 - Natal - RN

madruga@dimap.ufrn.br, thais@ufrnet.br, alf.rodrigo@gmail.com, affonso@dca.ufrn.br

Resumo *Diversas arquiteturas Peer-to-Peer são baseadas em super-peers, visando lidar melhor com a heterogeneidade dos nós existentes nesse tipo de rede. Entretanto, os modelos atuais não são apropriados para redes DHT (Distributed Hash Table), comprometendo algumas de suas propriedades mais interessantes. Este trabalho propõe um tipo especial de super-peer, o lightweight super-peer, que realiza basicamente o roteamento de mensagens, deixando que o processamento das mesmas seja realizado pelos nós clientes. Essa arquitetura reduz as mudanças de topologia na rede, a carga nos super-peers e a quantidade de mensagens nas redes das organizações. Além disso, aproveita melhor os recursos dos nós e resolve o problema do transversal NAT.*

Abstract. *Many Peer-to-Peer architectures are based on super-peers in order to deal with the heterogeneity of the nodes in these networks. However, the current models are not adequate for DHT networks (Distributed Hash Table) and they do not take advantage of some important properties of these networks. This work proposes a special type of super-peer, the lightweight super-peer, that performs the messages routing and leaves the message processing to the client nodes. This architecture reduces the changing in the network topology, the overload of the super-peers and the number of messages in the enterprise internal network. In addition, it takes advantage of the nodes resources and solves the transversal NAT problem.*

1. Introdução

Após o surgimento e a grande popularidade obtida pelas redes *Peer-to-Peer* (P2P) [Steinmetz 2005] através das aplicações de compartilhamento de arquivos, diversas áreas da computação têm mostrado interesse por esse tema, gerando uma proliferação de arquiteturas de redes P2P diferentes. Apesar dessa diversidade, essas redes podem ser classificadas basicamente em não estruturadas e estruturadas [Oram 2001], onde as principais representantes dessa última categoria são as redes DHT (*Distributed Hash Table*) [Ratnasamy 2002]. As redes DHT possuem alta escalabilidade, balanceamento de carga e tolerância a falhas. Além disso, garantem que qualquer dado existente na rede é localizado contatando um número limitado de nós, que pode variar entre diferentes arquiteturas, mas, tipicamente, é inferior a log(N), onde N é o número de nós na rede.

Entretanto, diante do fato de que as redes P2P são compostas por um número elevado de nós com capacidades (processamento, banda de rede, etc) bastante heterogêneas e possuem um comportamento altamente dinâmico (os nós entram e deixam a rede constantemente), novos modelos têm sido propostos para

lidar com essas características, onde nós com melhores recursos desempenham funções especiais [Yang 2001]. Esses nós são chamados *super-peers* e atuam como servidores para um conjunto de clientes e como um nó igual a outros em uma rede de super-peers.

Embora a abordagem de *super-peers* produza uma melhora significativa no desempenho das redes P2P, nos modelos atuais cada *super-peer* assume a responsabilidade sobre os recursos disponibilizados por um conjunto de outros nós. Isso significa que eles respondem às pesquisas endereçadas a esses nós e realizam o roteamento das mensagens circulando na rede. Esse modelo de funcionamento dos *super-peers* dificulta a sua integração com redes DHT, pois compromete algumas das funcionalidades dessas redes, além de não conseguir se beneficiar dessa arquitetura. Ou seja, as características dos modelos de super-peers atuais não foram criadas apoiadas em características das redes DHT. A seguir, citamos os principais problemas identificados nas arquiteturas de super-peers existentes:

- Ao utilizar as arquiteturas de *super-peers* em redes DHT (*Distributed Hash Tables*) [Ratnasamy 2002], apenas os *super-peers* são mapeados para o espaço de identificadores. Perde-se, com essa abordagem, algumas das principais características dessas redes, como, por exemplo, a distribuição homogênea dos recursos entre todos os nós, que é proporcionada pela utilização das funções *hash* [Karger 1997].
- Como a entrada de um novo *super-peer* na rede implica na alteração da sua topologia, acarretando a reconfiguração de outros nós na rede, e o aumento no tempo das pesquisas, os modelos atuais procuram reduzir o número total de *super-peers*. Essa estratégia, entretanto, aumenta o número de nós conectados a cada *super-peer* levando a uma sobrecarga nos mesmos. Tal sobrecarga ocorre porque nos modelos atuais, normalmente, os nós clientes publicam suas informações nos *super-peers*, e estes é que ficam responsáveis por processarem as requisições.
- Os modelos de *super-peers* atuais não são adequados para ambientes onde existe um alto número de nós com boas capacidades, pois, para ter seus recursos efetivamente aproveitados, um nó tem que se tornar *super-peer*, recaindo nos problemas discutidos no item anterior.
- Os *super-peers* roteiam mensagens que não são destinadas nem a ele próprio nem aos nós sobre os quais está responsável. Isso não apenas sobrecarrega o *super-peer*, como gera tráfego desnecessário dentro das redes das organizações. Por organização nos referimos a uma, ou várias redes interconectadas, que estão sob a administração de uma única entidade, como uma empresa, ou uma universidade, por exemplo.
- Os *super-peers* são implementados na camada de aplicação, o que gera sobrecarga para o processo de roteamento, uma vez que as mensagens precisam ser passadas do *kernel* para o espaço do usuário e, posteriormente, (após as decisões de roteamento da rede P2P) retornarem ao *kernel*.

Diante desses fatores e do fato que, devido a sobrecarga imposta aos *super-peers*, não há nenhum interesse por parte dos nós em se tornarem *super-peers* [Singh 2003], neste artigo propomos uma arquitetura P2P que emprega um novo tipo de *super-peer*, chamado *Lightweight Super-peer* (LSP). Essa arquitetura pode ser aplicada a qualquer rede DHT, como Chord [Stoica 2001], ou CAN [Ratnasamy 2001], por

exemplo, mas neste trabalho será apresentada sobre a rede P2P SGrid [Madruga 2006], que também é baseada no modelo DHT e considera a posição dos nós na rede física para criar a topologia lógica da rede. Para resolver os problemas acima citados, o objetivo principal do LSP é separar as funções de roteamento de mensagens e as pesquisas endereçadas aos *nós* sob a responsabilidade do LSP. Para isso, definimos o protocolo NATal (*routing and* NAT *Aplication Layer*) cuja função é rotear (e fazer NAT – *Network Address Translation* [Srisuresh 1999]]) de acordo com dados da camada de aplicação. O NATal permite que o LSP fique encarregado basicamente do roteamento das mensagens e deixe que o processamento das mesmas seja feito pelos nós clientes. Para isso, o NATal especifica que os nós devem informar ao LSP as chaves pelas quais são responsáveis. Dessa forma, o LSP encaminhará diretamente para cada *nó* cliente as mensagens que lhes são destinadas, aliviando a carga no *super-peer* e permitindo que os recursos de cada nó sejam mais efetivamente utilizados. Além disso, mensagens que não são destinadas a nenhum desses nós são roteadas no próprio LSP sem entrar na rede interna da organização. Essas mensagens podem até nem entrar pelo link de acesso à Internet da organização, caso seja utilizado o modelo estendido de roteamento com LSPs operando no modo "*routing only*".

Esse artigo está estruturado da seguinte forma. A seção 2 apresenta brevemente a arquitetura básica do SGrid (sem os *super-peers*). A seção 3 apresenta o modelo LSP e como essa estrutura é aplicada na rede SGrid, além do protocolo NATal. A seção 4 apresenta comentários sobre a implementação do LSP e do NATal e ilustra testes de desempenho. A seção 5 apresenta alguns trabalhos relacionados. A seção 6 contém os comentários finais.

2. SGrid

A arquitetura básica sobre a qual o modelo de *super-peers* é construído consiste de uma rede P2P estruturada chamada SGrid [Madruga 2006]. Essa rede considera a localização física dos nós para organizar sua estrutura lógica e é composta por um espaço bi-dimensional de lados idênticos. O mapeamento dos nós para esse espaço bi-dimensional é baseado no endereço IP. Essa rede automaticamente adapta sua estrutura quando os nós entram ou deixam a mesma. Ela provê uma operação de pesquisa que executa em tempo determinístico, como a maioria das redes P2P estruturadas, resolvendo todas as pesquisas usando, no máximo, O(log N) mensagens para outros nós. O SGrid segue a idéia proposta em CAN (*Content-Addressable Network*) [Ratnasamy 2001], onde um mecanismo de endereçamento escalável suporta eficiente inserção e recuperação de conteúdo. Como em outras redes DHT tanto os nós quanto os dados (chaves) são mapeados para pontos da grade e a determinação do nó responsável por uma chave leva em consideração as distâncias entre esses pontos. A seção 2.1 descreve o SGrid, conforme apresentado em [Madruga 2006], e a seção 2.2 comenta sobre uma extensão recente incorporada a esse modelo para melhorar o posicionamento dos nós.

2.1 Arquitetura do SGrid

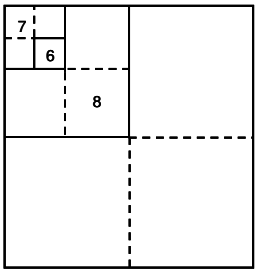
Nesse artigo apresentamos as características do SGrid que são essenciais para o entendimento do modelo de *super-peers*: o método de mapeamento dos nós para o espaço de identificadores, o método de alocação das chaves aos nós, os nós para os quais cada nó mantém apontadores e o algoritmo de pesquisa. Maiores detalhes sobre o SGrid podem ser encontrados em [Madruga 2006]. Evidentemente, além da forma como

essas operações devem ser realizadas, o SGrid especifica também todos os formatos de mensagens que são trocadas entre os nós da rede sendo, portanto, um protocolo de aplicação. O mapeamento dos nós para o espaço bi-dimensional do SGrid é baseado no endereço IP de cada nó. O endereço IP é utilizado para especificar o par (x,y) no espaço de coordenadas para onde o nó é mapeado. Esse ponto servirá como base para determinar as chaves pelas quais o nó é responsável e os demais nós para os quais o nó manterá conexões.

O método de mapeamento dos nós no SGrid é baseado no esquema hierárquico de alocação dos endereços IP. Assumindo que normalmente redes com prefixos IP comuns possuem alguma relação de proximidade, e quanto maior o tamanho desse prefixo comum maior a proximidade, o endereço IP é dividido em *m* grupos de *n* bits. A grade do SGrid é também dividida recursivamente em *m* grupos. Cada grupo do endereço IP, analisados da esquerda para a direita, determinará um quadrante (entre os quatro possíveis) de cada nível da grade. A Figura 1 ilustra a localização do nó com o endereço IP 200.241.86.131 em uma grade de tamanho de lado 256. Uma grade com tamanho de lado 256 tem 8 níveis, uma vez que $\log_2(256) = 8$, e como um endereço IPv4 tem 32 bits, conseqüentemente existem 8 grupos de 4 bits cada. A Figura 1a mostra a divisão do endereço IP em grupos e o quadrante equivalente a cada grupo, considerando que a representação do endereço IP 200.241.86.131 é 11001000.11110001.01010110.10000011. A Figura 1b mostra a representação gráfica para os quadrantes 6,7 e 8 da Figura 1a.

Nível	Bits	Bits (decimal)	Bits (decimal) mod 4	Quadrante
8	1100	12	0	Superior esquerdo
7	1000	8	0	Superior esquerdo
6	1111	15	3	Inferior direito
5	0001	1	1	Superior direito
4	0101	5	1	Superior direito
3	0110	6	2	Inferior esquerdo
2	1000	8	0	Superior esquerdo
1	0011	3	3	Inferior direito

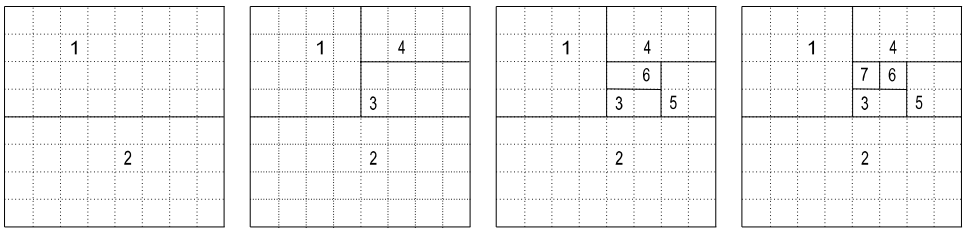
a) Os bits do endereço IP e o mapeamento para os quadrantes.



b) Representação gráfica do mapeamento para os níveis 8, 7 e 6.

Figura 1. Mapeamento relativo aos 12 bits mais à esquerda do endereço IP.

Cada nó é responsável por uma região quadrada ou retangular de chaves. Quando um novo nó entra na rede, o nó responsável pela chave para a qual o novo nó mapeia divide as suas chaves com ele, formando duas regiões menores. Esse processo é ilustrado na Figura 2.



a) Divisão das chaves após a entrada dos nós 1 e 2.

b) Divisão das chaves após a entrada dos nós 3 e 4.

c) Divisão das chaves após a entrada dos nós 5 e 6.

c) Divisão das chaves após a entrada do nó 7.

Figura 2. Divisão do espaço de chaves após a entrada dos nós.

O espaço de identificadores é dividido em níveis hierárquicos, conforme ilustrado na Figura 3 e cada nó mantém apontadores para dois nós em cada nível. Nessa figura, os nós mais escuros indicam os nós para os quais o nó mais claro possui apontadores.

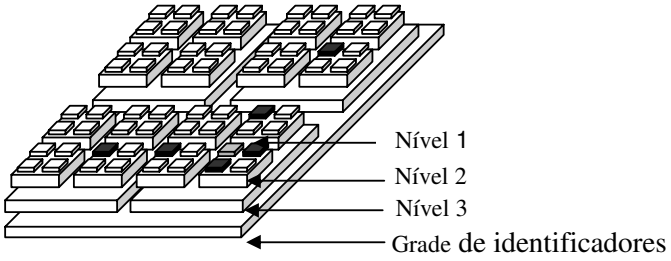


Figura 3. Grade com 3 níveis e os nós apontados em cada nível.

As pesquisas são realizadas calculando o menor quadrado de nível que englobe tanto a chave quanto o nó onde a pesquisa está sendo feita, e repassando-se a consulta para o algum nó vizinho nesse nível para o qual o nó possui um apontador. Na Figura 3 cada região quadrada corresponde a um quadrado de nível. Esse algoritmo proporciona buscas com complexidade O(log N), no pior caso.

2.2 Extensão do SGrid

Dois novos recursos foram adicionados ao SGrid após o modelo apresentado em [Madruga 2006] e brevemente descrito na seção 2.1. Esses recursos tratam de melhorias no mecanismo de posicionamento dos nós e no balanceamento de carga.

Embora o esquema de mapeamento dos endereços IPs do SGrid posicione endereços IPs semelhantes em locais próximos na grade, com a diminuição das faixas de endereços IPs ainda disponíveis, observa-se que, atualmente, mesmo endereços com prefixos comuns podem pertencer a redes completamente diferentes e distantes umas das outras, conforme apresentado em [Freedman 2005]. Para permitir que o SGrid detecte essas situações, uma variação do esquema de mapeamento é utilizada, onde são consideradas as coordenadas geográficas (latitude e longitude) de um nó, e um esquema de propagação de mensagens por difusão. Com essas modificações, ao entrar na rede, cada nó utiliza uma função *hash* para mapear suas coordenadas (latitude e longitude) para um ponto da grade e publica essas informações (seu endereço IP e suas coordenadas) no nó responsável por essa coordenada. Quando um outro nó deseja entrar na rede ele utiliza o mesmo mecanismo para mapear sua latitude/longitude para um ponto da grade e envia uma mensagem para o nó responsável por esse ponto, solicitando a relação de todos os nós que registraram suas coordenadas nesse ponto. Ao receber a mensagem, o referido nó reenvia-a, em difusão, para todos os seus nós vizinhos e assim sucessivamente. Como as respostas dessas mensagens, que produzem uma lista de nós próximos geograficamente, são enviadas ao nó que está entrando na rede, este nó pode calcular o nó que está mais próximo dele. Para isso, basta fazer medições do RTT (r*ound-trip time*) para cada um dos nós retornados, ou um subconjunto deles. O alcance da propagação das mensagens por difusão é controlado por um campo TTL (*Time to live*) , que informa o número máximo de vezes que a mensagem pode ser repassada, semelhante ao Gnutella [Singla 2001], e um campo que indica o número de respostas já obtidas. Esse último campo é incrementado cada vez que o nó que vai repassar a mensagem possuir registros de nós. Se esse campo atingir o valor limite definido, o nó

não reenvia a mensagem. Assim, quanto mais nós existirem na rede, maiores as chances de existirem nós próximos geograficamente e, portanto, menor o número de vezes que a mensagem terá que ser propagada.

Após obter o nó que possui o menor RTT entre os nós geograficamente próximos, o nó que deseja entrar na rede irá comparar esse RTT (RTT-LL) com o RTT para o nó fornecido pelo mapeamento convencional do SGrid, que é obtido apenas pelo mapeamento do endereço IP (RTT-IP), utilizando, o que fornecer o menor RTT, ou seja, o valor retornado pela fórmula: *Min(RTT-LL, RTT-IP).*

3. Lightweight Super-Peers(LSP)

Os três principais objetivos dos *Lightweight super-peers* (LSPs) são: diminuir a carga em cada *super-peer*, permitindo um melhor aproveitamento dos recursos dos nós clientes, diminuir a freqüência com que ocorrem mudanças na topologia da rede P2P, e reduzir o número de mensagens circulando dentro das redes que não são endereçadas a nenhum de seus nós.

As próximas duas seções apresentam a arquitetura do modelo LSP sobre o SGrid e discutem como os dois primeiros objetivos são alcançados. A seção 3.3 comenta sobre o objetivo relacionado à redução do número de mensagens nas redes das organizações. A seção 3.4 apresenta uma modificação feita no SGrid para garantir o balanceamento de carga na presença dos LSPs.

3.1 Redução na carga dos super-peers

A diminuição da carga nos *super-peers* é obtida fazendo com que o LSP deixe de realizar o processamento das mensagens destinadas aos nós internos, e apenas as redirecione para estes nós.

Como o modelo LSP foi criado para redes DHT, para que o LSP possa encaminhar as mensagens para os nós clientes é necessário analisar as informações referentes à chave sendo pesquisada. Essa informação faz parte da camada de aplicação e será utilizada para guiar o processo de reenvio do pacote. Além de fazer parte da rede P2P, mantendo sua tabela de apontadores, o LSP desempenha funções semelhantes às de um roteador, realizando as seguintes operações: (1) roteamento e NAT baseados nos dados da camada de aplicação; (2) interação com os nós da rede interna. O modelo LSP define o protocolo NATal (*Routing and **NAT a**pplication **l**ayer)*, que especifica em que consiste, e como deve ser, essa interação dos nós internos com o LSP e como o roteamento e o NAT devem ser executados pelo mesmo.

O modelo de roteamento proposto é formado por uma arquitetura onde cada nó cliente solicita sua região de chaves ao LSP. Essa região será, portanto, um subconjunto da região de chaves do LSP, de modo que a união das regiões de todos os clientes de um LSP formam uma única região de chaves. Dessa forma, ao receber uma mensagem, o LSP verificará se a mesma refere-se às chaves registradas pelos nós da rede interna. Em caso positivo, a mensagem é encaminhada diretamente ao nó da rede interna responsável pela chave. Em caso negativo, o LSP consulta sua tabela de roteamento e encaminha a mensagem para o nó especificado pela mesma, sem que tal mensagem entre na rede interna da empresa. A Figura 4 ilustra uma rede com três *nós* internos e um LSP, que possui endereço IP 200.1.2.3. Os valores a_x,b_x,c_x,d_x representam a região retangular de chaves pelas quais cada nó é responsável.

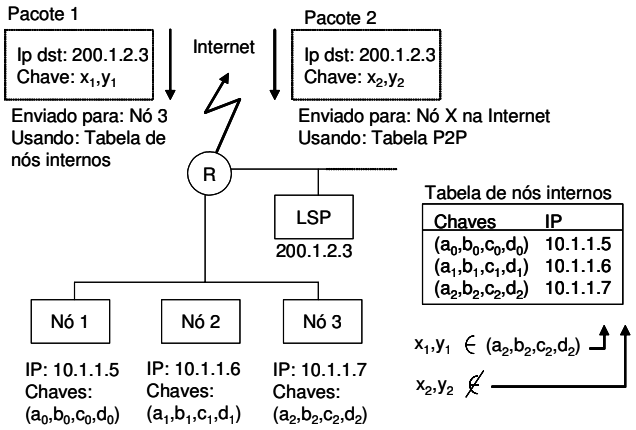


Figura 4. Roteamento utilizando LSP e o protocolo NATal.

Algumas observações são importantes em relação à Figura 4:

- O processo de repasse das mensagens para a rede interna requer a utilização de NAT no endereço de destino. Deve ser utilizado NAT estático para permitir que as máquinas recebam mensagens mesmo sem terem iniciado nenhuma comunicação. O IP a ser utilizado para o NAT é determinado através da tabela com o registro dos nós internos de acordo com a chave pesquisada. Vale ressaltar, que as mensagens do SGrid possuem, como um de seus campos, a chave sendo pesquisada.
- Conforme já citado, o LSP é a única máquina que fará parte da rede P2P global. Embora seja possível utilizar o LSP com endereço IP privado, através do redirecionamento de portas no roteador, o cenário recomendado é que o LSP seja uma máquina com IP público na rede de servidores da organização, tipicamente a DMZ (*demilitarized zone*).
- Sempre que um LSP roteia um pacote ele faz NAT no endereço de origem, colocando o seu endereço nesse campo. Além disso, também coloca seu endereço em um campo específico para essa finalidade, na parte de dados do pacote, caso o referido campo esteja vazio. Esse procedimento, portanto, será feito apenas pelo primeiro LSP no caminho de uma mensagem, ou seja, o LSP da organização onde a mensagem foi gerada. Essa técnica serve para evitar problema de detecção de *ip spoofing* e permitir que o destino final da mensagem possa enviar o retorno para a origem real da mesma.
- O protocolo de transporte utilizado é o UDP e todas as mensagens possuem campos para indicar as coordenadas (x_1,y_1) da chave pesquisada e as coordenadas (x_2,y_2) para qual o nó que fez a pesquisa foi mapeado. Semelhante ao que ocorre com os campos de endereços IP de origem e destino e portas TCP e UDP de origem e destino, que são trocados nas repostas dos pacotes IP-TCP (ou IP-UDP) os campos (x_1,y_1) e (x_2,y_2) são trocados nas mensagens de resposta do SGrid.
- Tanto as pesquisas geradas pelos nós internos, quando as respostas geradas por esses nós, são enviadas através do LSP. Essas mensagens contêm o endereço IP do nó para o qual o LSP deve encaminhar a mensagem. Para o primeiro caso, isso é possível porque os nós internos recebem uma cópia da tabela de roteamento do LSP. Vale ressaltar, que esse comportamento é opcional, ou seja, cada nó interno pode optar por não manter uma cópia da tabela de roteamento, deixando que o LSP tome as decisões de roteamento. Neste caso, contudo, acarreta-se um aumento na carga de processamento do LSP. Diferentemente de soluções de proxy, ou NAT dinâmico, o

LSP não mantém nenhuma informação de estado a respeito dessas mensagens depois que elas são encaminhadas, o que evita sobrecarga na máquina.

- Embora as informações utilizadas para o roteamento, ou seja, as chaves, façam parte da camada de aplicação o LSP não executa como um processo de usuário, mas sim diretamente no kernel do sistema operacional, para melhorar o desempenho. Essa técnica é semelhante ao modelo utilizado pelo *Linux Virtual Server* [Kopper 2005].

Ainda na Figura 4, vemos que ao receber o pacote 1, o roteador o encaminha diretamente ao Nó 3 pois, embora esse pacote esteja endereçado ao IP do LSP, ele se refere a chave (x_1,y_1) que pertence ao conjunto de chaves (a_2,b_2,c_2,d_2) registrado pelo Nó 3. Ao receber o pacote 2, o roteador verifica que o mesmo refere-se a chave (x_2,y_2) que não foi registrada por nenhum nó interno e, portanto, após consultar sua tabela de roteamento P2P, que não é mostrada na Figura 4, o LSP encaminha o pacote para o nó especificado na mesma, que está fora da rede da organização.

3.2 Redução nas mudanças de topologia

Como o único nó a entrar na rede P2P global é o LSP, quando novos nós clientes ligam-se ao LSP, e obtém uma parte da região de chaves do mesmo, nenhuma reconfiguração na nessa rede é necessária. Embora um comportamento semelhante seja observado nos demais modelos de *super-peers*, nesses modelos o aproveitamento dos recursos dos nós clientes é limitado, uma vez que publicam seus recursos nos *super-peers*. No modelo LSP, como as mensagens são entregues aos nós clientes para serem processadas, há um aproveitamento total dos recursos de cada nó.

3.3 Redução na quantidade de mensagens nas redes das organizações

Conforme citado anteriormente, o fato do LSP realizar o roteamento de mensagens evita que mensagens que não são destinadas a nenhum dos nós internos da organização entrem na rede da mesma. Entretanto, como o LSP está na rede de servidores - DMZ, e, portanto, dentro da organização, esse esquema apenas resulta em economia real caso a organização possua unidades dispersas geograficamente e interconectadas por canais de comunicação. Nos modelos distribuídos, como Chord, por exemplo, a mensagem passaria pelos canais de comunicação internos até atingir o nó de destino e depois seria roteada de volta para a Internet.

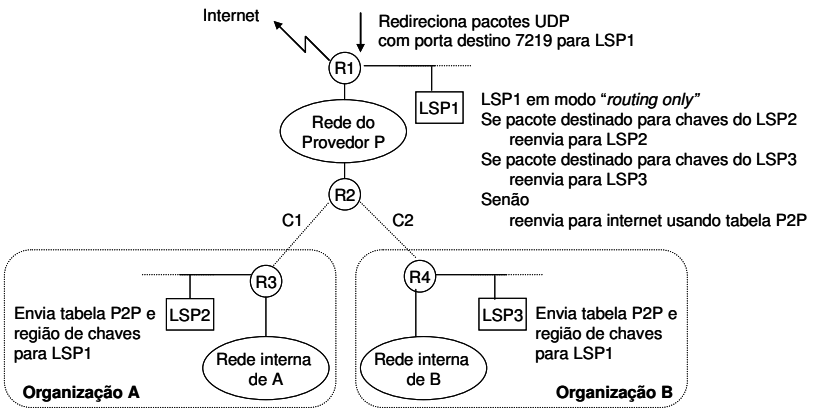


Figura 5. Redução no tráfego de mensagens nas redes das organizações

Na Figura 5 observa-se que o provedor P fornece acesso a Internet para duas organizações, Organização A e Organização B. O acesso à Internet da organização A se dá através do canal de comunicação C1 e o acesso à Internet da organização B se dá através do canal de comunicação C2. O roteador do provedor, R1, é configurado para redirecionar todo tráfego SGrid vindo da Internet para o LSP1. Esse tráfego caracteriza-se por utilizar o protocolo UDP na porta 7219. Esse redirecionamento de portas é uma operação bastante utilizada atualmente e é definida em [Srisuresh 1999]. O LSP1 é um LSP que opera em um modo especial chamado "*routing only*". Quando um LSP está trabalhando nesse modo ele não entra na rede P2P global, tendo como única função realizar o roteamento das mensagens vindas da Internet. Para isso, ele recebe periodicamente as tabelas de roteamento dos LSPs subordinados a ele, além da região de chaves pelas quais cada LSP é responsável. Para a Figura 5, isso significa que LSP2 e LSP3 enviam suas tabelas de roteamento P2P e a região de chaves pelas quais são responsáveis para LSP1. Caso uma mensagem recebida por LSP1 seja referente à região de chaves sob a responsabilidade de LSP2 ou LSP3, LSP1 a encaminha diretamente para o respectivo LSP. Caso a mensagem seja destinada ao endereço IP de LSP2 ou LSP3, mas a chave pesquisada não faça parte da região de chaves do referido LSP, LSP1 utiliza a tabela de roteamento P2P desse LSP, para rotear a mensagem. Como os nós contidos na tabela de roteamento de um LSP são nós externos a rede do LSP, a mensagem será enviada de volta à Internet sem passar pelo canal de comunicação daquela organização. Ainda para a Figura 5, caso LSP1 receba uma mensagem para o endereço IP de LSP3, mas a chave pesquisada não faça parte da região de chaves de LSP3, LSP1 utiliza a tabela de roteamento P2P de LSP3 para rotear a mensagem. Essa operação encaminhará a mensagem de volta a Internet, sem utilizar o canal de comunicação C2.

3.4 Balanceamento de Carga

Uma das fortes características das redes DHT é o balanceamento de carga, que procura deixar cada nó responsável aproximadamente pelo mesmo número de chaves. Ao inserir no SGrid o modelo de super-peers baseado no LSP, onde os nós clientes são atribuídos ao LSP baseando-se apenas na organização a que pertencem, pode acontecer de haver *super-peers* com muitos nós clientes e outros com apenas alguns poucos. Deve-se, portanto, definir um esquema de divisão de chaves que leve em consideração o número de nós clientes que um LSP possui.

Conforme explicado na seção 2, cada nó mantém sua tabela de apontadores para outros nós da rede P2P. Como parte do mecanismo de tolerância a falhas, não abordado nesse artigo, cada nó deve periodicamente enviar uma mensagem para cada um de seus vizinhos para verificar se os mesmos estão operando normalmente. Nessas mensagens é solicitado também que o nó retorne a relação entre o tamanho do espaço de chaves e o número de nós clientes do LSP, ou seja, o número de chaves por nó cliente. Desse modo, um nó que deseje ingressar na rede, escolhe um nó base, usando o mecanismo explicado na seção 2.2, que servirá como referência para a determinação de sua posição na grade. A seguir, envia uma mensagem a esse nó base, perguntando a qual nó deve solicitar seu ingresso na rede, ou seja, o nó ao qual deve ligar-se. A resposta do nó base será ele próprio ou um de seus vizinhos. Para tomar tal decisão, o nó base aplica a seguinte fórmula e retorna o nó para o qual obtiver o maior valor.

Resultado = ((N - Índice_Vizinho) * relação_chaves_nó)

onde:

N é o número de vizinhos que um nó pode possuir, igual para todos os nós, pois depende do tamanho da grade.

Índice_Vizinho é o número do nível ao qual o vizinho se refere, conforme explicado na seção 2. Portanto, quanto menor o nível, mais próximo o vizinho. Quando a fórmula estiver sendo aplicada para o próprio nó base possui valor 0.

relação_chaves_nó é o número médio de chaves pelas quais cada cliente do LSP vizinho é responsável.

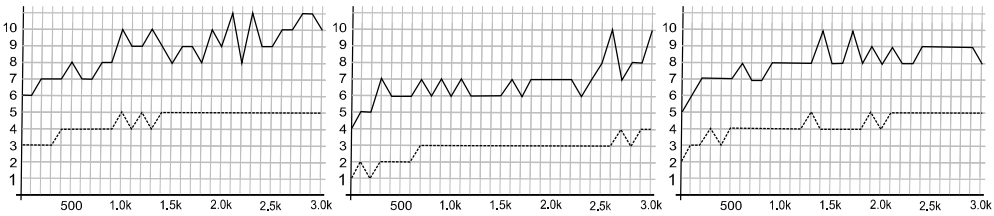
A fórmula acima significa que, ao entrar na rede, um nó pode se ligar a outro nó que está um pouco mais distante que o nó base, mas que possui um número elevado de chaves por nó. Esse método penaliza a relação de proximidade entre os nós para tornar mais equilibrada a divisão de chaves entre eles, fazendo com que cada LSP tenda a ficar responsável por aproximadamente o mesmo número de chaves.

4. Implementação e Avaliação

Todos os componentes do modelo LSP, incluindo protocolo SGrid, protocolo NATal, e os LSPs, foram implementados em C, o que permitiu criar uma rede real e comprovar o correto funcionamento desse modelo. O código do LSP foi implementado como um módulo para o kernel do Linux utilizando o Netfilter [Benvenuti 2005], que permite o registro de funções para manipular cada pacote recebido pela máquina.

Entretanto, como a utilização de uma rede real limita os testes de escalabilidade da rede, foi implementado um simulador, em Java, onde foi criado um ambiente que permite a inserção e remoção de nós na rede, incluindo LSPs, bem como a pesquisa por chaves. Esse ambiente proporciona ainda a geração de gráficos estatísticos que contém, por exemplo, informações a respeito do número de nós visitados em uma pesquisa. Desse modo, é possível comparar o desempenho da rede com e sem a presença dos LSPs.

A Figura 6 ilustra os gráficos gerados por operações de pesquisa, onde as abscissas representam o tamanho da rede, em nós, e as ordenadas o número de nós visitados. A Figura 6a é uma rede sem LSPs. As Figuras 6b e 6c são redes que utilizam LSPs, onde são mostrados o número de nós visitados (média utilizando linha pontilhada e máximo utilizando linha cheia) em uma operação de pesquisa variando o número de nós na rede.



a) Rede sem LSPs b) Rede com 10% de LSPs c) Rede com 50% de LSPs

Figura 6. Pesquisas em redes com e sem a presença de LSPs.

Para o experimento cujo gráfico de pesquisa está ilustrado na Figura 6, as consultas foram realizadas a cada intervalo de 100 novos nós inseridos na rede. O número de consultas foi 300 (trezentos). Para o gráfico da Figura 6b o número de LSPs

existentes na rede é 10% do número de nós total e para o gráfico da Figura 6c esse valor subiu para 50%.

Foi mostrado em [Madruga 2006] e pode-se observar pelo gráfico 6a que no SGrid o número máximo de nós visitados em um processo de pesquisa é O(Log N), mas que, na média, este valor é reduzido pela metade. Em uma rede com 1000 (mil) nós o número máximo nós visitados em uma pesquisa foi 10 (dez), que corresponde a O(Log N), e o número médio de nós visitados foi 5 (cinco), que é igual a O(Log N)/2. Quando se introduz os LSPs, essa relação continua existindo, sendo que N passa a ser o número de LSPs. Desse modo, conforme mostra a Figura 6b, para uma rede com 1000 nós que possui 10% de LSPs, portanto, 100 LSPs, o número máximo de nós visitados em uma pesquisa foi 6 e o número médio foi 3, que são menores que Log(100) e Log(100)/2, respectivamente. Valores semelhantes são obtidos para o experimento da Figura 6c, onde as redes possuem 50% de LSPs. Para a rede com 1000 nós, portanto 500 LSPs, o número máximo de nós visitados em uma pesquisa foi 8 e o número médio foi 4, que são menores que Log(500) e Log(500)/2, respectivamente.

5. Trabalhos Relacionados

Existem diversas arquiteturas P2P baseadas em *super-peers* como, por exemplo, JXTASearch [Waterhouse 2001], FastTack [Leibowitz 2003], Gnutella2 [Singla 2001] e Edutella [Nejdl 2002]. O trabalho apresentado em [Yang 2001] analisa as diferentes opções de projeto que precisam ser consideradas por uma arquitetura desse tipo, tais como: o número de nós ligados a cada *super-peer*, o nível de redundância utilizado, a quantidade de *super-peers* existentes e a topologia da rede formada pelos *super-peers*. Além desses pontos, o mecanismo de roteamento das mensagens entre os *super-peers* é de fundamental importância para o desempenho da rede.

A rede FastTrack, utilizada pelo Kaaza [Leibowitz 2003], e Gnutella 2, utilizam mecanismos de inundação para comunicação entre os *super-peers*, ou seja, cada *super-peer* comunica-se com todos os demais *super-peers*. Isso, evidentemente, leva a problemas de escalabilidade. As redes Edutella organizam os *super-peers* em uma estrutura de hiper-cubo e também utilizam técnicas de *broadcast* para repassar as pesquisas entre os *super-peers*. Entretanto, tal estratégia utiliza índices construídos em XML para limitar a abrangência das pesquisas. JXTASearch constitui a parte relacionada ao mecanismo de pesquisa da arquitetura JXTA [Gong 2001]. JXTA define um conjunto padrão de protocolos para a construção de uma rede *overlay* genérica que seja proveitosa para o desenvolvimento de um grande número de aplicações e serviços. No JXTASearch os dados são publicados pelos nós nos *super-peers* através de descrições em XML, que por sua vez constroem índices utilizando essas informações. A rede entre os *super-peers* utiliza um modelo híbrido entre as redes por inundação e as DHT, sendo chamada de Rede DHT fracamente acoplada, uma vez que permite inconsistência temporária entre os índices mantidos pelos *super-peers.* Para reduzir os problemas decorrentes dessa flexibilidade, o JXTASearch utiliza replicação de dados entre os *super-peers*. Por se basear no modelo DHT, cada *super-peer* é capaz de determinar o *super-peer* exato para onde a busca deve ser encaminhada.

Independentemente das características específicas, nenhuma dessas redes aborda os problemas discutidos nesse trabalho nem as soluções propostas pelo SGrid-LSP: redução na carga dos *super-peers,* redução nas mudanças de topologia da rede e redução no número de mensagens nas redes das organizações. Nessas arquiteturas os nós comuns

não participam da rede diretamente, uma vez que publicam suas informações nos *super-peers*. Isso inviabiliza o balanceamento de carga fornecido pelo modelo DHT e impossibilita um completo aproveitamento dos recursos dos nós clientes, como, por exemplo, capacidade de processamento e armazenamento.

Nessas outras arquiteturas qualquer nó pode realizar a função de *super-peer*, enquanto que no modelo SGrid-LSP essa função precisa ser realizada por um servidor. Entretanto, a própria evolução da Internet tem acontecido apoiando-se nos benefícios da utilização de servidores dedicados para intermediar as comunicações, como é o caso, por exemplo, dos servidores SMTP e proxy HTTP, ressaltando-se, contudo, o perigo de sobrecarga nesses nós. Porém, diferentemente desses servidores, que processam as requisições dos seus clientes e precisam manter informações de estado enquanto atendem as requisições, o LSP dificilmente se tornará um gargalo, pois não processa mensagens, apenas as reenvia, e não mantém nenhuma informação de estado a respeito das comunicações que estão acontecendo.

O problema de comunicação entre máquinas utilizando IPs privados, conhecido como transversal NAT, não é abordado pelas arquiteturas citadas anteriormente, requerendo a utilização de um mecanismo adicional chamado *Hole Punching* [Ford 2003]. Essa técnica, entretanto, requer a utilização de um servidor intermediário além de ser dependente da implementação do NAT. Como na arquitetura do SGrid-LSP os pacotes são sempre enviados para o endereço do LSP, que são públicos, e repassados para os nós internos através da análise dos dados do cabeçalho de aplicação, o problema do transversal NAT não existe.

6. Conclusões

Grande parte das redes *overlay* propostas recentemente têm sido baseadas em *super-peers*. Embora essas arquiteturas tenham se mostrado bastante eficientes, foi mostrado em [Yang 2001] que essa eficiência depende da parametrização correta de certos itens. Nesse artigo, identificamos deficiências no modelo de *super-peers* que não podem ser solucionadas apenas através da parametrização da rede e propomos um modelo de *super-peers* específico para redes DHT, para soluciona-las.

Esse trabalho propõe a separação das funções de roteamento de mensagens e do processamento das mesmas, deixando que os *super-peers* encarreguem-se apenas da primeira delas. Para isso, foi proposto um tipo especial de *super-peer* chamado LSP (*Lightweight Super-Peer*), que realiza o roteamento de mensagens P2P baseado na chave pesquisada, ou seja, em dados da camada de aplicação. Essa operação, que é realizada através do protocolo NATal (*Routing and NAT application layer*), permite que mensagens que não são destinadas a nenhum dos nós da organização sejam roteadas sem entrar na rede da organização e que, mensagens destinadas a algum nó da rede interna, sejam processadas diretamente pelo nó destino, e não pelo *super-peer*, como na maior parte dos modelos que usam esse tipo especial de nó. Esse esquema permite um melhor aproveitamento dos recursos computacionais dos nós clientes dos *super-peers*. Além disso, uma vez que as regiões de chaves de todos os nós de uma organização são adjacentes, esses nós são vistos como um único nó pelos *nós* externos. Desse modo, quando já existir algum nó de uma organização na rede P2P global, a entrada ou saída de outros nós da organização na rede P2P é uma operação local e não precisa ser percebida pelos nós externos. Portanto, essa estratégia reduz as mudanças de topologia na rede P2P global.

As diversas redes P2P existentes requerem mecanismos externos, como, por exemplo, *Hole Punching*, para permitir a comunicação entre máquinas que estão atrás de NAT. No modelo LSP o esquema de roteamento baseado em dados da camada de aplicação elimina esse problema, uma vez que todas as mensagens são sempre endereçadas ao IP do LSP.

Embora, por motivos de espaço, não tenham sido apresentados detalhes de como utilizar o modelo LSP em outras redes DHT, a principal modificação a ser feita nessas redes é fazer os clientes comunicarem-se com o LSP usando o protocolo NATal, para obterem suas regiões de chaves, e enviarem suas pesquisas através desse nó. A rede P2P propriamente dita é formada pelos LSPs, e esses nós utilizariam o próprio protocolo da rede DHT.

Este trabalho apresentou também duas extensões importantes do SGrid: uma que melhora o esquema de posicionamento dos nós baseado no endereço IP, e outra que garante o balanceamento de carga para o modelo LSP. Esta última modificação pode também ser adotada, com as devidas adaptações, quando o LSP for utilizado sobre outras redes DHT.

Os conceitos apresentados nesse trabalho foram validados através de uma implementação que estende a rede P2P SGrid. A implementação do LSP foi desenvolvida em um ambiente Linux usando o NetFilter. Esse ambiente mostrou-se bastante propício para o desenvolvimento do modelo apresentado, uma vez que permite uma fácil integração com o código de roteamento contido no kernel do sistema operacional. Além da implementação real do modelo apresentado nesse artigo, foi desenvolvido um simulador onde pode-se realizar testes em redes com elevados números de nós e comprovar que existe uma melhora no desempenho das pesquisas quando se utilizam LSPs na rede.

Como trabalhos futuros pretende-se reescrever o simulador para integrá-lo a simuladores mais amplamente utilizados pela comunidade científica, como o NS2 e o PeerSim, desenvolver um protocolo multicast na camada de aplicação sobre o SGrid e desenvolver um mecanismo de pesquisa, utilizando o SGrid, que permita buscas flexíveis, como, por exemplo, por palavras-chave, subcadeias e faixas de valores.

7. Referências

Benvenuti, C. (2005) Understanding Linux Network Internals, O'Reilly.

Ford, B., Srisuresh, P., and Kegel, D. (2003) Peer-to-peer (P2P) communication across middleboxes. Internet Draft. http://midcom-p2p.sourceforge.net/draft-ford-midcom-p2p-01.txt, October.

Freedman, M., Vutukuru, M., Feamster, N., and Balakrishnan, H. (2005) Geographic locality of ip prefixes. In Proceedings of ACM Internet Measurement Conference, October.

Gong, L. (2001) JXTA: A network programming environment. IEEE Internet Computing, vol. 5, pp. 88-95.

Karger, D., Lehman, E., Leighton, F., Levine, M., Lewin, D., and Panigrahy, R. (1997) Consistent hashing and random trees: Distributed caching protocols for relieving hot

spots on the World Wide Web. In Proceedings of the 29th Annual ACM Symposium on Theory of Computing, pp. 654–663, El Paso, TX, May.

Kopper, K. (2005) The Linux Enterprise Cluster: Build a Highly Available Cluster with Commodity Hardware and Free Software, No Starch Press.

Leibowitz, N., Ripeanu, M., and Wierzbicki, A. (2003) Deconstructing the Kazaa network, in proceedings of 3 rd IEEE Workshop on Internet Applications, (WIAPP'03), San Jose, California, June.

Madruga, M. Batista, T. Guedes, L. (2006) Uma Arquitetura P2P Baseada na Hierarquia do Endereçamento IP. Artigo aceito para o 24º Simpósio Brasileiro de Redes de Computadores (SBRC 2006), Curitiba, Brasil.

Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmer, M. and T. Risch. (2002) EDUTELLA: a P2P Networking Infrastructure based on RDF. In Proceedings of the 11th International Conference on World Wide Web. Hawaii, USA, May.

Oram, A. (2001) Peer-to-Peer: harnessing the benefits of a disruptive technology. 1st ed., Beijing; Sebastopol, CA: O'Reilly. xv, 432.

Ratnasamy, S., Francis, P., Handley, M., Karp, R. and Shenker, S. (2001) A Scalable Content-addressable Network. In Proceedings of the ACM SIGCOMM '01 Conference, pp. 161-172, San Diego, California, August.

Ratnasamy, S., Shenker, S., and Stoica, I. (2002) Routing algorithms for DHTs: Some open questions. Presented at International Workshop on Peer-to-Peer Systems (IPTPS), pp. 45-52, Cambridge, USA, March.

Singh, S., Ramabhadran, S., Baboescu, F. and Snoeren, (2003) A. The case for service provider deployment of super-peers in peer-to-peer networks, June.

Singla, A. And Rohrs, C. (2001) Ultrapeers: Another Step Towards Gnutella Scalability. http://RFC-Gnutella.sourceforge.net/Proposals/Ultrapeer, December.

Srisuresh, P. and Holdrege, M. (1999) IP network address translator (NAT) terminology and considerations. http://www.ietf.org/rfc/rfc2663.txt, August.

Steinmetz, R., and Wehrle, K. (2005) Peer-to-Peer Systems and Applications. Springer.

Stoica, I., Morris, R., Karger, D., Kaashoek, M. F. e Balakrishnan, H. (2001) "Chord: A scalable peer-to-peer lookup service for internet applications," Proceedings of the 2001 ACM SIGCOMM Conference, pp. 149-160.

Waterhouse, S. (2001) JXTA Search: Distributed Search for Distributed Networks. white paper, Sun Microsystems, Palo Alto, Calif.

Yang, B. and Molina, H.G. (2001) Designing a Super-Peer Network. Proceedings of the 19th International Conference on Data Engineering (ICDE), pp. 49-60, Bangalore, India.