Avaliação Experimental da Escalabilidade de Sistemas P2P e um Novo Algoritmo de Controle de Taxas de Upload

Diego Ximenes Mendes[1], **Edmundo de Souza e Silva**[1]

[1] Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro, RJ - Brasil

{diegoximenes,edmundo}@land.ufrj.br

Abstract. *Recent studies have show that P2P systems are not always scalable, which is evidenced by a phenomenon called "the missing piece syndrome". This syndrome occurs when the vast majority of peers have all pieces of the file being downloaded, except one, common to all those peers. In that scenario the system's performance is compromised. This phenomenon was detected and studied in the literature using analytical models, however, there aren't results indicating that it really occurs in practical situations. In this context this work has two objectives. The first is to capture the occurrence of this syndrome from an experimental setup that employs a real P2P system's protocol (BitTorrent). To our knowledge it is the first experimental study with this purpose. The second is to propose a new upload rate control algorithm to alleviate the problem. Using analytical models we study the advantages and disadvantages of the new strategy.*

Resumo. *Estudos recentes mostram que nem sempre sistemas P2P são escaláveis, o que é evidenciado por um fenômeno chamado de "síndrome do pedaço faltante". Tal síndrome ocorre quando a grande maioria dos peers possui todos os pedaços do arquivo sendo obtido, exceto um deles, comum a todos esses peers. Nesse cenário a vazão do sistema é comprometida. Esse fenômeno foi detectado e estudado na literatura a partir de modelos analíticos. Entretanto, não existem resultados mostrando que o problema ocorre em situações práticas. Nesse contexto este trabalho possui dois objetivos. O primeiro é o de captar a existência dessa síndrome a partir de experimentação utilizando um protocolo real de sistemas P2P (BitTorrent). Do nosso conhecimento essa é a primeira abordagem nesta direção. O segundo é o de propor um novo algoritmo de controle de taxas de upload com o objetivo de aliviar o problema. Através de modelos analíticos avaliamos as vantagens e desvantagens da nova estratégia.*

1. Introdução

Um sistema *Peer-to-Peer* (P2P) é uma arquitetura descentralizada de rede na qual os nós, chamados de *peers*, atuam tanto como consumidores quanto fornecedores de conteúdo. Desse modo um sistema P2P permite a disseminação de dados de maneira eficiente e escalável, sendo atualmente responsável por uma parte considerável do tráfego na Internet.

Em uma aplicação de distribuição de arquivos, ao se incorporar a um *swarm* (conjunto de *peers* interessados em um mesmo conteúdo), um *peer* traz consigo recursos como banda e memória ao sistema. Portanto, a capacidade de transmissão do *swarm*

aumenta com o crescimento da quantidade de *peers*. Porém, devido a própria limitação da dinâmica de um protocolo P2P, essa expansão de recursos pode não corresponder a um aumento proporcional da vazão (taxa na qual os usuários completam os seus downloads). Em geral, um sistema é dito escalável se a sua vazão aumenta linearmente com o crescimento do número de usuários. Por outro lado, um sistema P2P torna-se instável quando a sua vazão é menor do que a taxa de chegada de *peers* ao *swarm*, pois esse caso resulta no crescimento ilimitado da população de *peers* ao longo do tempo.

No trabalho desenvolvido em [Hajek and Zhu 2010], foi detectado, a partir de modelos matemáticos, um fenômeno causador de instabilidade em sistemas P2P chamado de síndrome do pedaço faltante. Tal síndrome é caracterizada quando a grande maioria dos *peers* possui todos os pedaços do arquivo sendo compartilhado, com exceção de um pedaço comum a todos esses *peers*. Nesse cenário, quando dois *peers* se conectam para realizar uma transmissão, com alta probabilidade eles não possuem dados úteis a serem trocados, ou seja, ambos possuem a mesma parte do arquivo obtida. Dessa forma as bandas de upload dos usuários ficam ociosas, e o *publisher* (servidor que possui o arquivo completo) se torna responsável por boa parte das transmissões. Tal situação enfatiza a ideia de que a instabilidade ocorre em ocasiões em que o servidor é o gargalo, isto é, a performance do sistema é altamente dependente da capacidade de serviço do *publisher*. É importante enfatizar que o *publisher* normalmente compartilha seus recursos e, portanto, a taxa dedicada a um único *swarm* pode ser muito baixa em relação à sua capacidade total.

Nesse contexto o presente trabalho possui duas contribuições principais:

- Desenvolvimento de um ambiente de experimentação do protocolo BitTorrent utilizando a plataforma de testes em redes GENI [Duerig et al. 2012, GENI 2015]. A partir desse ambiente foram geradas evidências experimentais da ocorrência da síndrome do pedaço faltante. Do nosso conhecimento, não há na literatura nenhum estudo experimental sobre esse fenômeno.
- Proposta de um novo algoritmo de controle de taxas de upload com o objetivo de aliviar os efeitos dessa síndrome. Foram elaborados modelos analíticos baseados nos trabalhos [Menasché et al. 2012, de Souza e Silva et al. 2014b, de Souza e Silva et al. 2014a], com o intuito de analisar e comparar a performance do novo algoritmo em relação a propostas anteriores.

Este artigo está organizado do seguinte modo. Na seção 2 são apresentados conceitos e resultados recentes na literatura sobre a escalabilidade de sistemas P2P. A seção 3 discorre sobre os estudos experimentais realizados. A seção 4 descreve o novo algoritmo de controle de taxas de upload proposto, assim como o modelo estocástico para avaliar o seu desempenho. Por fim, a seção 5 conclui o trabalho.

2. Conceitos Básicos e Trabalhos Relacionados

Os trabalhos de [Menasché et al. 2012, Hajek and Zhu 2010, Zhu and Hajek 2012, de Souza e Silva et al. 2014b, de Souza e Silva et al. 2014a] tiraram diversas conclusões teóricas a respeito da escalabilidade de sistemas P2P. No modelo de [Hajek and Zhu 2010, Zhu and Hajek 2012] as seguintes suposições foram adotadas: o arquivo sendo disseminado é divido em K pedaços (ou blocos de informação) de tamanhos iguais. Existe um único *publisher* (servidor) sempre online que detém todos os blocos do arquivo, e possui capacidade de upload U pedaços por unidade de tempo. Os *peers* chegam com taxa Poisson (com parâmetro λ *peers* por unidade de tempo). A intervalos de tempo com duração

exponencial e taxa μ, um *peer* seleciona outros aleatoriamente em um *swarm*. Assim que um *peer* é selecionado, um pedaço é escolhido aleatoriamente dentre aqueles pedaços que o potencial receptor não possui e o download é instantâneo. Caso o *peer* selecionado não possua nenhum pedaço ausente no *peer* que o selecionou, uma nova escolha aleatória de *peers* é feita. Esta política de seleção é chamada de *random peer, random useful piece*.

Nos artigos [Menasché et al. 2012, de Souza e Silva et al. 2014b, de Souza e Silva et al. 2014a], o upload de pedaços é feito à taxa μ. Assim que uma transferência de pedaço é concluída, outro *peer* é selecionado (o tempo para realizar uma seleção de *peers* ou pedaço é desprezível). Diversas modificações foram feitas para permitir o estudo de várias outras políticas de seleção. Além disso, estudos tanto de sistemas abertos (taxa de chegadas independente do número de usuários em um *swarm*) quanto de sistemas fechados (tamanho do *swarm* permanece constante) foram feitos. Sistemas fechados são úteis para analisar situações de estresse.

O trabalho de [Hajek and Zhu 2010] foi pioneiro na detecção da síndrome do pedaço faltante para a política de seleção considerada naquele artigo. Foi demostrado que, as políticas de seleção de pedaços *Rarest First Piece* e *Random Useful Piece* possuem a mesma região de estabilidade, e que quando os *peers* e o servidor adotam as políticas *Random Peer*/*Random Useful Piece* o sistema se torna instável quando $\lambda > U$.

No trabalho [Menasché et al. 2012] é feita uma avaliação da região de estabilidade quando o *publisher* adota as estratégias *Most Deprived Peer*/*Rarest First Piece* e os *peers* utilizam *Random Peer*/*Random Useful Piece*. Nos próximos parágrafos é realizada uma análise simplificada desse cenário detalhado em [Menasché et al. 2012], explorando a condição de que $\lambda > U$.

Em um caso de saturação onde a síndrome do pedaço faltante é configurada, um grande número de *peers* possui todos os pedaços com exceção de um pedaço c, chamado então de pedaço faltante. Todos os *peers* que possuem todos os pedaços com exceção do c caracterizam um grupo especial, chamado de *one club*.

Na conjuntura descrita anteriormente, com alta probabilidade o *publisher* sempre serve um *peer* recém chegado ao *swarm* que ainda não possui pedaços. Tal fato acontece pois $\lambda > U$, e o *publisher* com a política *Most Deprived Peer* privilegia os recém chegados, até então desfavorecidos em termos de blocos obtidos. Também é possível perceber que o pedaço c é o mais raro no *swarm*, já que o one club é abundante e todos os *peers* pertencentes a ele possuem os outros $K - 1$ pedaços. Portanto, como o servidor adota a política *Rarest First Piece* o pedaço c será selecionado para transmissão. Seguindo esse raciocínio são definidas duas categorias de *peers*, os *gifted peers* que recebem o pedaço c do servidor imediatamente após entrarem no *swarm*, e os denominados *non-gifted peers* que não recebem esse pedaço inicial.

Além das transmissões analisadas no parágrafo anterior, é possível verificar as conexões de trocas de blocos entre os diferentes grupos de *peers*, e assim concluir resultados sobre a instabilidade a partir de todas essas conexões. Devido a estratégia *Random Peer* adotada pelos *peers* e pela abundância do *one club*, os *gifted-peers* e os *non-gifted peers* terão a todo momento algum *peer* do *one club* para os servir. Desse modo, rapidamente os *non-gifted peers* convergem para se tornarem pertencentes ao *one club*, e os *gifted peers* rapidamente completam os seus downloads e deixam o *swarm*.

Com alta probabilidade, os *non-gifted peers* selecionam algum *peer* do *one club* para realizar o upload, tentativa de transmissão que não tem sucesso devido a não existência de pedaços úteis para a troca. Já os *gifted peers*, com grande probabilidade, selecionam *peers* do *one club* para transmitir o pedaço mais raro c, já que esse é o único bloco útil para a troca.

Como foi descrito, os *gifted peers* permanecem pouco tempo no *swarm*, e consequentemente possuem pouco tempo para disseminar o pedaço c ao *one club*. Em [Menasché et al. 2012] a análise se estende levando em consideração os valores das taxas de upload juntamente com as conexões aqui descritas. Dessa forma foi concluído que nessa situação o sistema é estável quando $\lambda < KU$, e quando o sistema é instável o *one club* cresce indefinidamente.

Considerando os efeitos e as transmissões descritas acima, em [de Souza e Silva et al. 2014b] foi proposta uma estratégia simples mas eficiente para melhorar a performance de sistemas P2P. Essa estratégia consiste na modificação da taxa de upload dos *peers* que possuem $K-1$ pedaços para $0 < \mu' < \mu$. A ideia por detrás dessa estratégia pode ser resumida a seguir. A taxa de upload dos *peers* pertencentes ao *one club* (e consequentemente com $K-1$ pedaços) têm sua taxa diminuída, fazendo com que os *gifted peers* se mantenham por mais tempo no sistema. Desta forma, os *gifted peers* podem fornecer o pedaço mais raro ao *one club* por períodos de tempo mais longos. Em outras palavras, tal estratégia é uma tentativa de controlar o tamanho do *one club* e consequentemente frear a instabilidade. Os resultados apresentados em [de Souza e Silva et al. 2014b] indicam que esse método simples pode gerar um aumento significativo da vazão.

Os modelos de [de Souza e Silva et al. 2014b] mostraram que na medida em que μ' diminui a vazão aumenta. Entretanto, esse aumento não é ilimitado e a vazão cresce enquanto μ' diminui até atingir uma vazão máxima, a partir da qual a vazão passa a diminuir com a redução de μ'. Esse fato é explicado pelo sistema se degenerar a um caso cliente-servidor, onde o *publisher* se torna responsável pela grande maioria das transmissões.

3. Experimentos

Conforme comentado na Seção 2, a síndrome do pedaço faltante tem sido estudada a partir de modelos teóricos e de simulação que são simplificações dos protocolos P2P. O objetivo desta seção é o de constatar a existência desse fenômeno a partir de experimentação utilizando um protocolo real e comumente utilizado na prática.

Foi criado um ambiente de software capaz de criar, gerenciar e analisar experimentos com *swarms* de distribuição de arquivos. Para tal objetivo foi utilizada uma implementação do protocolo BitTorrent chamada libtorrent [libtorrent 2015], e os experimentos foram executados na plataforma de testes em redes GENI [Duerig et al. 2012].

O *Global Environment for Network Innovations* (GENI) consiste de um projeto da *National Science Foundation* (NSF), cuja intenção é prover um laboratório virtual para a exploração de soluções de engenharia em redes e sistemas distribuídos em larga escala. A sua infraestrutura possibilita a utilização de diversos recursos, como por exemplo, PCs físicos e virtuais, roteadores, roteadores programáveis, etc. O GENI é um *testbed* compartilhado onde múltiplos usuários podem executar experimentos simultaneamente sem

que exista interferência entre eles, o que é possível devido a técnicas de virtualização que permitem o isolamento dos recursos de diferentes experimentos.

Como um dos nossos objetivos é experimentar com diferentes parâmetros do protocolo e ainda programar diferentes politicas de seleção de *peers* e pedaços, escolhemos a libtorrent por ser *open-source*. Desta forma, o protocolo pode ser adaptado além de existir uma comunidade ativa de usuários facilitando a troca de informações.

A libtorrent possui uma série de políticas de seleção de *peers*, sendo uma delas baseada na proposta feita em [Chow et al. 2008]. Porém, para os experimentos realizados no presente trabalho, foi necessário o desenvolvimento de adaptações da libtorrent, como por exemplo, a implementação das estratégias *Random Peer* e *Most Deprived Peer*, que não constam na versão original. Para o papel do *tracker* foi utilizado o software opentracker [opentracker 2015], que consiste de um projeto também aberto mas que não necessitou de alterações.

Para os experimentos, utilizou-se de máquinas virtuais por serem recursos mais abundantes na infraestrutura do GENI. No mecanismo elaborado, cada máquina virtual alocada tem associada algumas entidades da dinâmica do protocolo BitTorrent, sendo que todas essas máquinas pertencem a mesma VLAN. Um cliente BitTorrent que faz o papel do *publisher* e o *tracker* possuem cada um uma máquina virtual exclusiva. Já as máquinas virtuais referentes aos *peers* podem estar associadas a mais de um *peer* simultaneamente, onde a quantidade de *peers* por máquina virtual é um dos parâmetros do experimento. A limitação das taxas de upload ocorre através da utilização de procedimentos específicos da própria libtorrent.

Um experimento é iniciado quando todos os usuários entram no *swarm* ao mesmo tempo. Ao entrarem no *swarm*, os *peers* não possuem pedaços do arquivo sendo transmitido, já o *publisher* é o único detentor de todos os pedaços. Assim como nos modelos abordados na Seção 2, um usuário deixa o *swarm* após finalizar o download do arquivo. Além disso, as taxas de upload são idênticas para todos os *peers*.

Conforme descrito anteriormente, em um caso de saturação, os modelos teóricos indicam que a população de *peers* no *swarm* cresce indefinidamente com o tempo. Caso as taxas escolhidas levem a essa situação, seria impraticável realizar tal experimento, já que os recursos físicos são limitados, não permitindo assim o aumento indiscriminado da quantidade de usuários. Portanto, foi adotado o cenário de um sistema fechado, abordagem comum utilizada para estressar um sistema, mantendo a polução finita. Em um sistema fechado, o tamanho da população de usuários (denotada por N) permanece constante ao longo do tempo. Para o tamanho do *swarm* se manter inalterado, quando um *peer* completa o download do arquivo e consequentemente deixa o sistema, outro *peer* sem nenhum pedaço do arquivo entra no *swarm*. Dessa forma, é possível alocar previamente os recursos físicos necessários e suficientes para a realização de um experimento.

A seguir são apresentados resultados de alguns experimentos. A fim de exibir evidências da síndrome do pedaço faltante, as curvas abaixo mostram a fração de *peers* no *swarm* que possuem todos os pedaços, com exceção de um pedaço específico c, em determinado tempo de execução. Dessa forma, é identificada a presença do *one club* ao longo do tempo.

A Figura 1 mostra o resultado de três experimentos que utilizam as mesmas es-

tratégias e parâmetros, com exceção do número de pedaços K. Nas figuras, a taxa U do servidor é inferior à taxa de upload de pedaço. Esse fato não deve ser confundido como sendo baixa a taxa total do servidor que é disponibilizada para transmissão. Apenas indica que o servidor dedica ao swarm em questão uma *fração* de sua taxa nominal de serviço, uma vez que normalmente um servidor é compartilhado entre múltiplos *swarms* simultaneamente.

Como as taxas são definidas em pedaços/segundo, quando as taxas são mantidas e K cresce, o tamanho do novo arquivo aumenta. Portanto, no cenário descrito, diferentes valores de K estão associados a distribuição de arquivos com tamanhos diferentes, porém particionados em pedaços de mesmo tamanho. Essa escolha (μ constante) facilita a comparação e não traz alteração nas conclusões.

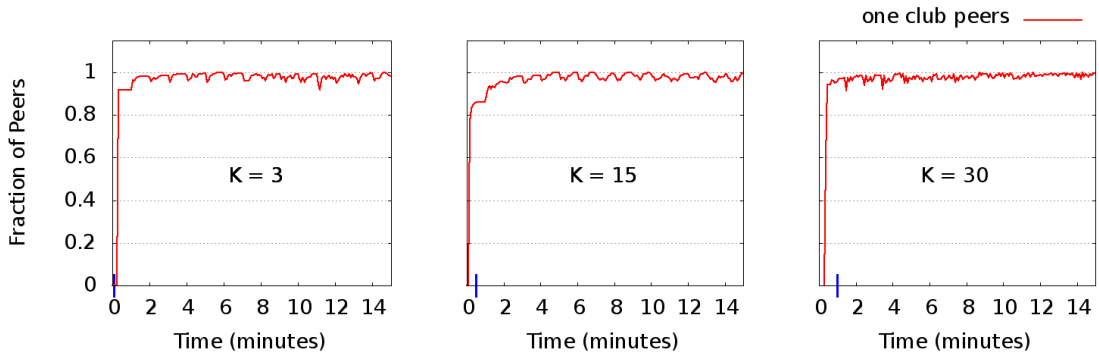

Figura 1. Fração de *peers* pertencentes ao *one club* durante o experimento. Gráficos para valores de K distintos. Estratégias: *peers* adotam *Random Peer*/*Random Useful Piece*, e o *publisher* utiliza *Most Deprived Peer*/*Rarest First Piece*. Parâmetros: $\mathrm{N} = 400$, $\mathrm{U} = 0.5$, $\mu = 10.0$.

Na Figura 1, a curva em vermelho mostra a fração de *peers* no *one club* ao longo do tempo. Já o traço azul indica o tempo teórico esperado para o servidor fazer o upload do arquivo inteiro para apenas um cliente em uma arquitetura cliente-servidor.

É possível notar que a fração de *peers* no *one club* cresce rapidamente com o tempo e aproxima-se de 1 logo após o início do experimento. Na maior parte do tempo a grande maioria dos *peers* pertence ao *one club*. Este é exatamente o que prevê os modelos teóricos para os parâmetros usados. Portanto, a síndrome do pedaço faltante ocorre em um ambiente com protocolo P2P real. Além disso, os gráficos indicam que essa síndrome pode ocorrer para diferentes valores de K.

A Figura 2 apresenta o resultado de três experimentos que utilizam as mesmas estratégias e parâmetros da Figura 1, mas para diferentes valores do tamanho do *swarm* N. Como no caso anterior, a síndrome do pedaço faltante ocorre para diferentes tamanhos de *swarm*. Além disso, é importante perceber que com o aumento do tamanho do *swarm* N, a fração de *peers* no *one club* fica ainda mais próxima de 1.

Observando-se o traço em azul das figuras, é fácil perceber que o número de *peers* no *one club* se aproxima da população do *swarm* em torno do tempo médio necessário para o servidor transferir o arquivo inteiro à um cliente. Como os *peers* no *one club*

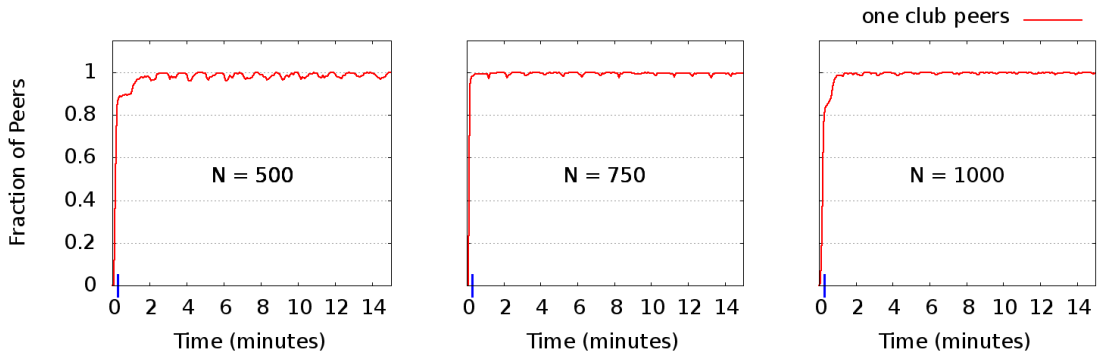

Figura 2. Fração de *peers* pertencentes ao *one club* durante o experimento. Gráficos para valores de $\mathbf{N}$ distintos. Estratégias: *peers* adotam *Random Peer*/*Random Useful Piece*, e o *publisher* utiliza *Most Deprived Peer*/*Rarest First Piece*. Parâmetros: $\mathbf{K = 10}$, $\mathbf{U = 0.5}$, $\boldsymbol{\mu = 10.0}$.

necessariamente tem que possuir $K - 1$ pedaços, o servidor deve, no mínimo, ter tido tempo para transmitir esses $K-1$ pedaços a qualquer *peer* pois todos os *peers* chegam sem nenhum pedaço ao *swarm*. Em outras palavras, pouco depois do servidor transmitir todos os pedaços ao *swarm*, o fenômeno da síndrome do pedaço faltante se instala, desde que os parâmetros do sistemas (população do *swarm*, taxa de upload e capacidade do servidor disponível ao *swarm*) sejam tais que levem à saturação, conforme previsto nos modelos analíticos. Enfatizamos que essas observações foram obtidas a partir de experimentação com o protocolo real.

4. Algoritmo de Controle de Taxas de Upload Baseado nas Raridades dos Pedaços

4.1. Modelos analíticos anteriores

Para a avaliação da performance do protocolo BitTorrent e variações, utilizamos os modelos dos artigos [Menasché et al. 2012, de Souza e Silva et al. 2014b, de Souza e Silva et al. 2014a], porém com algumas modificações para estudar, por exemplo, a estratégia de controle de taxas de upload proposta no presente trabalho.

Relembramos que o arquivo sendo transmitido é divido em K pedaços de tamanhos iguais, o servidor sempre online possui taxa de upload U pedaços por unidade de tempo, e os *peers* possuem taxas de upload homogêneas e iguais à μ pedaços por unidade de tempo. Além disso, para efeito de comparação, mostramos resultados para a estratégia proposta em [de Souza e Silva et al. 2014a] onde peers com $K - 1$ pedaços reduzem a sua taxa a $0 < \mu' < \mu$ pedaços por unidade de tempo. Todas as taxas aqui consideradas pertencem à distribuições exponenciais, e a população do *swarm* é constante (sistema fechado).

Seja o conjunto de pedaços $F = \{1, ..., K\}$ e C o conjunto de subconjuntos de F. Cada elemento do conjunto $S = C \setminus F$ representa uma possível configuração de pedaços que um *peer* pode possuir. Tal configuração é chamada de assinatura, sendo que a cardinalidade do conjunto S é $|S| = 2^K - 1$. O conjunto F não consta em S pelo fato dos *peers* deixarem o *swarm* imediatamente após concluírem o download de todos os

pedaços, não existindo assim a necessidade de se representar essa assinatura no modelo. Devido a simetria do cenário tratado, a escolha do espaço de estados considera o número de *peers* com uma dada assinatura. Seja ω_s o número de *peers* com a assinatura $s \in S$. Um estado $\boldsymbol{\omega}$ é caracterizado pelo número de *peers* com cada assinatura possível, ou seja, $\boldsymbol{\omega} = (w_{s_1}, ..., w_{s_n})$, sendo que $\bigcup_{i=1}^{|S|} s_i = S$ e $s_i \neq s_j \; \forall i \neq j$.

Um *peer* de assinatura s passa a ter uma nova assinatura ao receber um pedaço que ainda não possui. As taxas de transição entre os estados da cadeia são determinadas de acordo com os parâmetros e estratégias de seleção de *peers* e pedaços utilizados. Mesmo fazendo uso das possíveis simetrias dos estados, a cardinalidade do espaço de estados cresce exponencialmente com o aumento de K. Portanto, a solução analítica da cadeia de Markov, ou mesmo uma simulação, só podem ser feitas com um número bastante limitado de pedaços e *peers*. O Apêndice A fornece maiores detalhes do modelo elaborado.

Na Figura 3 são apresentadas três curvas referentes a diferentes valores de μ' quando é utilizada a estratégia de serviço modificado dos *peers* com $K - 1$ pedaços [de Souza e Silva et al. 2014a]. Tais curvas foram obtidas através de soluções analíticas do modelo, e relacionam a vazão com o tamanho do *swarm* N de um sistema P2P fechado.

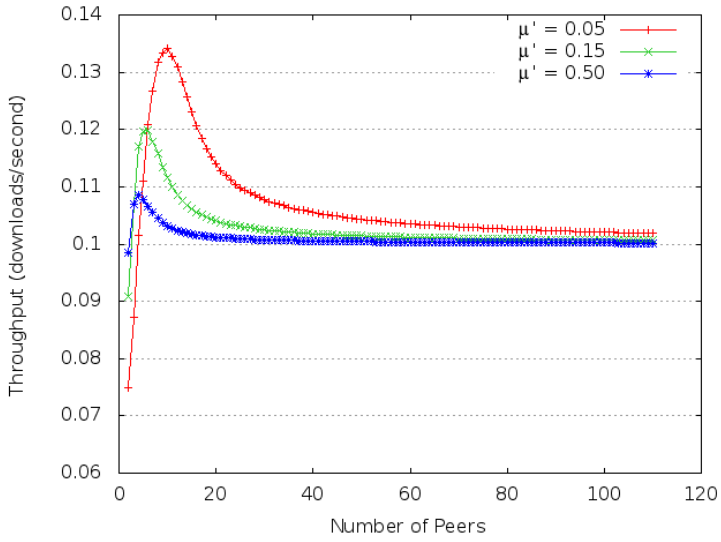

Figura 3. Número de *Peers* x Vazão. *Peers* e o *publisher* adotam as políticas *Most Deprived Peer*/*Rarest Useful Piece*/Serviço Modificado dos *Peers* com $\mathbf{K - 1}$ Pedaços. Parâmetros utilizados: $\mathbf{K = 2}$, $\mathbf{U = 0.1}$, $\boldsymbol{\mu}$= 0.5. Curvas para diferentes valores de μ'.

Assim como identificado em [Menasché et al. 2012, de Souza e Silva et al. 2014b, de Souza e Silva et al. 2014a], o formato específico das curvas acima é recorrente em diversas estratégias adotadas. Para valores pequenos do número de *peers* o *swarm* possui escalabilidade próxima de linear, onde a capacidade de serviço de cada *peer* é utilizada de maneira eficiente. Entretanto, com o crescimento de N, o sistema atinge um ponto de vazão máxima a partir do qual a vazão decresce e converge para um determinado valor. Dessa forma, considerando um número de *peers* muito grande, a inserção ou remoção de *peers* não contribui para o aumento da vazão do

swarm, indicando que os recursos físicos disponíveis não são bem aproveitados.

A partir da Figura 3 e em concordância com [de Souza e Silva et al. 2014a], é possível perceber que quanto menor for μ' melhor é o desempenho obtido. Entretanto, esse crescimento de vazão não é ilimitado com a diminuição de μ' [de Souza e Silva et al. 2014a].

4.2. O Algoritmo proposto

Com o objetivo de melhorar o desempenho de sistemas P2P, nesta seção é proposto um novo algoritmo de controle de taxas de upload em um protocolo P2P de distribuição de arquivos. Além de tentar prevenir a ocorrência da síndrome do pedaço faltante, a nova estratégia procura impedir que a síndrome se agrave caso ela se instale.

O algoritmo tem como estratégia realizar uma distribuição uniforme de pedaços diferentes entre os *peers* de um *swarm* e, desta forma, fazendo com que o *swarm* consiga utilizar sua capacidade de upload de maneira eficiente, evitando a ociosidade dos recursos. Essa eficiência ocorre pois desta forma, com alta probabilidade, quando dois *peers* se conectam para realizar uma transmissão, encontrarão um pedaço útil para a troca.

Nesse contexto, a estratégia proposta diminui consideravelmente as taxas de upload relacionadas a pedaços muito frequentes. Dessa forma, a quantidade de réplicas dos pedaços mais frequentes no *swarm* cresce mais vagarosamente do que o número de cópias dos pedaços menos frequentes. Portanto, pela proposta, o sistema tende a homogeneizar o número de réplicas dos diferentes pedaços no sistema.

Dada a distribuição das quantidades de cópias de cada pedaço do arquivo sendo disseminado é definido o conceito de raridade de um pedaço. Por definição, atribui-se o valor 1 à raridade do pedaço menos frequente no sistema, ou seja, a maior raridade possível é igual a uma unidade. A raridade do segundo pedaço menos frequente será igual a raridade do pedaço com maior raridade menos uma diferença percentual da quantidade de réplicas entre esses dois pedaços, considerando o bloco cuja raridade está sendo calculada como referência. O cálculo da raridade do terceiro pedaço menos frequente será feito de forma semelhante: a raridade do terceiro mais raro é calculada em relação ao segundo mais raro. Dessa forma encontra-se uma indução onde basta ordenar os pedaços pelas suas quantidades de réplicas de maneira crescente e aplicar a relação de raridade. Além dessa indução, é considerado um limite inferior para o valor da raridade, sendo aqui definido por r_{min} tal que $0 \leq r_{min} \leq 1$.

Formalizando os conceitos, seja $cnt[j]$ o número de cópias do pedaço j no *swarm*. Assumir uma permutação $p[1], ..., p[K]$ dos blocos $1, ..., K$ tal que $cnt[p[i]] \leq cnt[p[i+1]]$ para $i = 1, 2, \ldots, K-1$, ou seja, os pedaços são ordenados de maneira crescente em relação à contagem de réplicas. Definindo que $r[j]$ representa a raridade do pedaço j, as raridades de todos os blocos podem ser calculadas da seguinte maneira:

$$r[p[1]] = 1$$

$$r[p[i]] = \max\left(r_{min}, r[p[i-1]] \times \left(1 - \frac{cnt[p[i]] - cnt[p[i-1]]}{\max(cnt[p[i]], 1)}\right)\right), i = 2, \ldots, K$$

Através das definições acima é importante observar que as raridades pertencem

ao intervalo $[0, 1]$ e que as seguintes condições são satisfeitas:

Se $cnt[i] == cnt[j]$ **então** $r[i] == r[j]$
Se $cnt[i] < cnt[j]$ **então** $r[i] \geq r[j]$

Com o valor das raridades dos pedaços calculadas é possível obter as taxas de upload associadas aos blocos. Considerando um caso de taxas homogêneas, e que $taxaUploadPeer[i]$ indica a taxa de serviço dos *peers* ao transmitirem o pedaço i, e $taxaUploadPublisher[i]$ o análogo ao servidor:

$$taxaUploadPeer[i] = \mu \times r[i], \quad i = 1, 2, \ldots, K \tag{1}$$
$$taxaUploadPublisher[i] = U \times r[i], \quad i = 1, 2, \ldots, K \tag{2}$$

Para os *peers*, por exemplo, a nova taxa de upload para um pedaço i qualquer estará sempre no intervalo $r_{min} \times \mu \leq taxaUploadPeer[i] \leq \mu$, garantindo assim a consistência da nova taxa.

Por convenção e com o intuito de manter uma coerência com a política de serviço modificado dos *peers* com $K - 1$ pedaços foi adotado que: $r_{min} = \frac{\mu'}{\mu}$.

Desse modo, pedaços menos raros são distribuídos com uma taxa menor do que pedaços mais raros. Além disso, as taxas são definidas de maneira proporcional à distribuição das quantidades de réplicas no *swarm*. Definindo um limite inferior para as taxas de upload evita-se um cenário onde muitas raridades sejam iguais à zero, que por consequência acarretaria em um número de transmissões baixo, o que pode comprometer a vazão.

Considerando a ocorrência da síndrome do pedaço faltante, o algoritmo fará com que a taxa referente ao pedaço faltante seja a maior possível, ou seja, μ, e as taxas dos outros pedaços sejam a menor possível, ou seja, $\mu \times r_{min} = \mu'$. Dessa maneira quando o sistema estiver saturado, ou seja, quando o *one club* for muito grande, o algoritmo irá atuar da mesma forma como a estratégia do serviço modificado para os *peers* com $K - 1$ pedaços.

Conforme observado em [de Souza e Silva et al. 2014a] a ideia de diminuir a taxa de upload e assim aumentar a vazão pode parecer contra-intuitiva. Entretanto, a ocorrência da síndrome do pedaço faltante implica em uma grande perda de desempenho, e uma vez que tal síndrome se instale ela tende sempre a se agravar. Métodos que evitem a ocorrência dessa síndrome e a rápida remoção dela após a sua instauração devem obter bons resultados em termos de vazão global, mesmo que a princípio, localmente pareça uma estratégia ruim diminuir a taxa de upload. Apesar do algoritmo ter sido definido para adaptar a taxa de upload, a adaptação poderia ter sido feita na taxa de download utilizando os mesmos princípios, bastando perceber que ambos os métodos lidam com os mesmos fatores e objetivos.

A nova política aqui proposta, assim como a do serviço modificado para *peers* com $K - 1$ pedaços de [de Souza e Silva et al. 2014a], possuem um bom apelo em relação à economia de banda do usuário. Outra vantagem da técnica proposta está relacionada com o fato de que ela pode ser aplicada independentemente da escolha das estratégias de

seleção de *peers* e pedaços empregadas.

A principal desvantagem do mecanismo aqui proposto se refere ao *overhead* de informação necessário para que cada cliente tenha acesso às estatísticas dos pedaços de todo o *swarm*. Manter tais estatísticas atualizadas com alta frequência pode ser algo custoso, pois existe a possibilidade de sobrecarga por parte do *tracker*, e pela grande quantidade de tráfego associado aos dados de atualização. Todavia, a fim de contornar tal empecilho, algumas implementações do BitTorrent assumem que cada cliente pode aproximar as estatísticas globais do *swarm* pelas estatísticas locais com relação aos vizinhos desse cliente, diminuindo assim consideravelmente o *overhead*.

A seguir é apresentado a comparação do desempenho da estratégia de serviço modificado para *peers* com $K-1$ pedaços de [de Souza e Silva et al. 2014a] em relação ao algoritmo aqui proposto. Tal performance é comparada através dos resultados do modelo analítico de um sistema fechado anteriormente descrito, e assim analisando a vazão para diferentes valores de N.

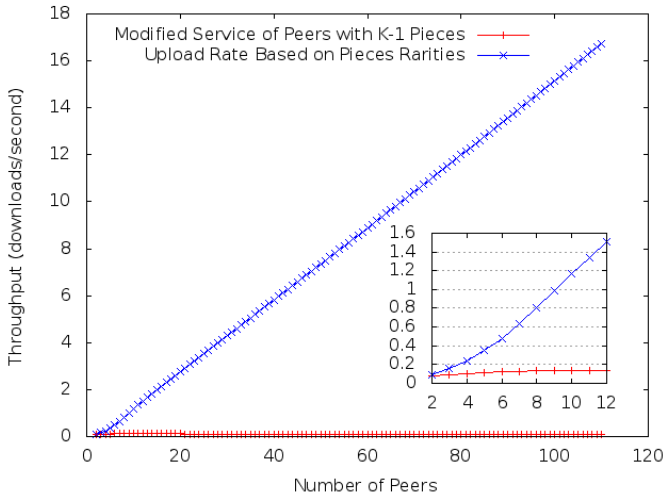

Figura 4. Número de *Peers* x Vazão. *Peers* e o *publisher* adotam as políticas *Most Deprived Peer* *Rarest Useful Piece*. A curva em vermelho indica a utilização do Serviço Modificado dos *Peers* com $\mathbf{K-1}$ Pedaços. Já a azul representa o uso do Algoritmo de Controle de Taxas de Upload Baseado nas Raridades dos Pedaços. Parâmetros utilizados: $\mathbf{K=2}$, $\mathbf{U=0.1}$, $\boldsymbol{\mu}$= 0.5, $\boldsymbol{\mu'}$= 0.05.

Pela Figura 4 é possível observar que o algoritmo proposto possui um desempenho significativamente melhor em relação à política de serviço modificado para *peers* com $K-1$ pedaços. Também é importante verificar que para a situação descrita o algoritmo alcança uma escalabilidade linear. Apesar do resultado inicial promissor, não é possível afirmar se o algoritmo torna o sistema estável ou se haverá saturação com o aumento do tamanho do *swarm*, seguindo o que ocorre com as curvas da Figura 3.

5. Conclusão

Este trabalho é o primeiro a detetar que o fenômeno denominado de síndrome do pedaço faltante pode ocorrer quando um protocolo P2P real (BitTorrent) é utilizado em um ambiente de experimentação. Os diversos experimentos realizados mostram que a síndrome

não é um evento incomum e pode ocorrer com diferentes parâmetros, conforme previsto por modelos analíticos apesar de bastante simplificados. Além disso, os estudos experimentais realizados mostram que, para parâmetros que levam à saturação, a síndrome ocorre rapidamente após o início do *swarm*.

Eliminar ou mesmo diminuir o efeito da síndrome do pedaço faltante é crucial para o bom desempenho de um *swarm*. Desta forma propomos um novo algoritmo de controle de taxa de upload que se mostra promissor através dos modelos estocásticos desenvolvidos. Uma desvantagem do algoritmo é o *overhead* quanto à obtenção das informações necessárias para a escolha de taxas. Entretanto, é possível identificar simplificações para diminuir tal *overhead*. Como trabalho futuro pretende-se estudar o desempenho de tais simplificações utilizando modelos analíticos. Além disso, pretende-se avaliar o algoritmo a partir do ambiente de experimentação utilizado neste trabalho, com o protocolo BitTorrent.

Referências

[BitTorrent.org 2015] BitTorrent.org (2015). The BitTorrent Protocol Specification. `http://www.bittorrent.org/beps/bep_0003.html`. Acessado em Fevereiro de 2015.

[Chow et al. 2008] Chow, A. L. H., Golubchik, L., and Misra, V. (2008). Improving bittorrent: A simple approach. In *Proceedings of the 7th International Conference on Peer-to-peer Systems*, IPTPS'08, pages 8–8, Berkeley, CA, USA. USENIX Association.

[de Souza e Silva et al. 2014a] de Souza e Silva, E., Leão, R. M. M., Menasché, D. S., and Towsley, D. (2014a). Scalability issues in P2P systems. *CoRR*, abs/1405.6228.

[de Souza e Silva et al. 2014b] de Souza e Silva, E., Menasché, D. S., Leão, R. M., and Towsley, D. (2014b). Sobre a capacidade de serviço de sistemas p2p. In *SBRC*, pages 61–74.

[Duerig et al. 2012] Duerig, J., Ricci, R., Stoller, L., Strum, M., Wong, G., Carpenter, C., Fei, Z., Griffioen, J., Nasir, H., Reed, J., and Wu, X. (2012). Getting started with geni: A user tutorial. *SIGCOMM Comput. Commun. Rev.*, 42(1):72–77.

[GENI 2015] GENI (2015). GENI Concepts. `http://groups.geni.net/geni/wiki/GENIConcepts`. Acessado em Fevereiro de 2015.

[Hajek and Zhu 2010] Hajek, B. and Zhu, J. (2010). The missing piece syndrome in peer-to-peer communication. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, pages 1748–1752.

[Kurose and Ross 2013] Kurose, J. and Ross, K. (2013). *Computer Networking A Top-Down Approach*. Pearson, 6 edition.

[Legout et al. 2006] Legout, A., Urvoy-Keller, G., and Michiardi, P. (2006). Rarest first and choke algorithms are enough. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, IMC '06, pages 203–216, New York, NY, USA. ACM.

[libtorrent 2015] libtorrent (2015). libtorrent. `http://www.libtorrent.org/`. Acessado em Fevereiro de 2015.

[Menasché et al. 2010] Menasché, D. S., de Aragão Rocha, A. A., de Souza e Silva, E., Leão, R. M. M., Towsley, D. F., and Venkataramani, A. (2010). Estimating self-sustainability in peer-to-peer swarming systems. *Perform. Eval.*, 67(11):1243–1258.

[Menasché et al. 2011] Menasché, D. S., de Aragão Rocha, A. A., de Souza e Silva, E., Towsley, D., and Leão, R. M. M. (2011). Implications of peer selection strategies by publishers on the performance of P2P swarming systems. *SIGMETRICS Performance Evaluation Review*, 39(3):55–57.

[Menasché et al. 2012] Menasché, D. S., Rocha, A. A., de Souza e Silva, E., Leão, R. M., and Towsley, D. (2012). Stability of peer-to-peer swarming systems. In *SBRC*, pages 161–174.

[opentracker 2015] opentracker (2015). opentracker. `http://erdgeist.org/arts/software/opentracker/`. Acessado em Fevereiro de 2015.

[Stewart 2009] Stewart, W. J. (2009). *Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling*. Princeton University Press.

[Zhu and Hajek 2012] Zhu, J. and Hajek, B. (2012). Stability of a peer-to-peer communication system. *IEEE Transactions on Information Theory*, 58(7):4693–4713.

Apêndice A Modelo Detalhado

Considerando o que foi descrito na seção 4.1, um *peer* de assinatura s passa a ter uma nova assinatura ao receber um pedaço que ainda não possui. A função $T(s,i)$ definida a seguir mapeia qual a nova assinatura de um *peer* quando o mesmo possui assinatura s e recebe o pedaço i quando $i \notin s$.

$$T(s,i) = \begin{cases} s \cup \{i\}, & se\ |s| < K-1 \\ \emptyset, & \text{caso contrário} \end{cases} \tag{3}$$

Sejam $R_S(s,i)$ e $R_P(s,i)$ funções que representam a taxa agregada de transição dos *peers* com assinatura s para $T(s,i)$ fornecida respectivamente pelo servidor e pelo conjunto de *peers* no *swarm*. A taxa de transição total será dada então por $R(s,i) = R_S(s,i) + R_P(s,i)$. As funções R_S e R_P são definidas de acordo com a estratégia de seleção de pedaço e *peers* adotadas. Para facilitar a notação algumas definições são feitas: B_s consiste do conjunto de pedaços úteis a um *peer* com assinatura s e que são menos replicados no *swarm*. Já $B_{s's}$ é o conjunto de pedaços que um *peer* com assinatura s' possui e que um *peer* com assinatura s não possui e são menos replicados no sistema. M representa o conjunto de assinaturas dos *peers* mais desfavorecidos em termos de pedaços baixados no sistema, sendo que apenas assinaturas presentes no *swarm* são consideradas. A seguir se encontram alguns exemplos de definições dessas funções para diferentes políticas abordadas no presente trabalho:

- *Random Peer/Random Useful Piece*

$$R_S(s,i) = \begin{cases} \frac{\omega_s U}{N(K-|s|)}, & se\ i \notin s \\ 0, & \text{caso contrário} \end{cases} \tag{4}$$

$$R_P(s,i) = \begin{cases} \frac{\omega_s \mu}{N-1} \sum\limits_{\substack{s' \in S \\ i \in s'}} \frac{\omega_{s'}}{|s-s'|}, & se\ i \notin s \\ 0, & \text{caso contrário} \end{cases} \tag{5}$$

- *Most Deprived Peer/Rarest First Piece*

$$R_S(s,i) = \begin{cases} \frac{\omega_s U}{|B_s| \sum\limits_{s' \in M} \omega_{s'}}, & se\ i \notin s;\ i \in B_s \\ 0, & \text{caso contrário} \end{cases} \tag{6}$$

$$R_P(s,i) = \begin{cases} \omega_s \mu \sum\limits_{\substack{s' \in S \\ i \in s'}} \frac{\omega_{s'}}{|B_{s's}| \left(\sum\limits_{s'' \in M} \omega_{s''} - \mathbb{1}(s' \in M) \right)}, & se\ i \notin s;\ i \in B_{s's} \\ 0, & \text{caso contrário} \end{cases} \tag{7}$$

Caso alguma política de seleção de taxa de upload seja utilizada basta trocar as taxas μ e U pelas respectivas taxas adaptadas.

A partir das possíveis transições entre assinaturas e as taxas agregadas de transições dos *peers* com determinada assinatura é possível definir quais serão as taxas da matriz de transição entre estados. A fim de facilitar a notação se define o vetor $\boldsymbol{e}_s$ com a mesma dimensão de $\boldsymbol{\omega}$ possuindo todas as coordenadas zero com exceção da coordenada referente à assinatura s que possui valor 1. Se $\omega_s > 0$, a taxa de transição entre os estados $\boldsymbol{\omega}$ e $\boldsymbol{\omega} - \mathbf{e_s} + \mathbf{e_{T(s,i)}}$ será dada por:

$$q(\boldsymbol{\omega}, \boldsymbol{\omega} - \mathbf{e_s} + \mathbf{e_{T(s,i)}}) = R(s,i) \tag{8}$$

As taxas não descritas aqui são definidas como tendo valor zero.

A vazão é calculada a partir do modelo verificando a taxa agregada das transições na cadeia de Markov que representam saídas de *peers* do *swarm*. Considerando $\boldsymbol{\pi}$ o vetor de probabilidades do estado estacionário e $\pi_{\boldsymbol{\omega}}$ a probabilidade estacionária do estado $\boldsymbol{\omega}$, a vazão é definida por:

$$vazão = \sum_{\boldsymbol{\omega} \in \Omega} \sum_{\substack{i=1 \\ \omega_{F \setminus \{i\}} > 0}}^{K} \pi_{\boldsymbol{\omega}} q(\boldsymbol{\omega}, \boldsymbol{\omega} - \mathbf{e_{F \setminus \{i\}}} + \mathbf{e_\emptyset}) \tag{9}$$