

# Formação de clusters em redes P2P por similaridade entre os nós\*

Fabício Murai<sup>1</sup>, Daniel R. Figueiredo<sup>1</sup>

<sup>1</sup>Programa de Engenharia de Sistemas e Computação – COPPE/UFRJ  
Caixa Postal: 68.511 – 21941-972 – Rio de Janeiro, RJ – Brasil

{fabricio,daniel}@land.ufrj.br

**Abstract.** *The tendency of having relationships form among similar vertices has been observed in several real networks. This phenomenon, known as assortative mixing (AM), is present not only in social networks but recently has also been observed in peer-to-peer (P2P) file sharing networks, where similarity is given by the access bandwidth capacity of the peers. Despite the fact that this “clusterization” is a driving force for system scalability, application protocols such as BitTorrent do not explicitly implement mechanisms to achieve this state. In this paper, we propose a simple model to understand which mechanisms within the BitTorrent protocol are responsible for driving the system into high levels of AM. We investigate the influence that system parameters have on the levels of AM achieved by the system. In particular, we show that the “optimistic unchoke” mechanism plays a major role in this process.*

**Resumo.** *A tendência na formação de relacionamentos entre vértices similares vem sendo observada em diversas redes reais. Este fenômeno, conhecido como assortative mixing (AM), está presente não somente em redes sociais, mas também vem sendo observado em redes par-a-par (P2P) para compartilhamento de arquivos, onde a similaridade dos vértices é dada pela largura de banda de acesso dos pares. Essa “clusterização”, apesar de ser responsável pelo bom desempenho do sistema, não está explicitamente refletida no funcionamento dos protocolos destas aplicações, como o BitTorrent. Neste trabalho, propomos um modelo simples para entender quais mecanismos do protocolo BitTorrent levam à ocorrência do AM. Além disso, estudamos a influência dos parâmetros do modelo na taxa de AM atingida pelo sistema. Finalmente, observamos que o mecanismo de “unchoke otimista” tem papel fundamental neste processo.*

## 1. Introdução

Muitas redes que representam sistemas reais possuem características topológicas comuns, mesmo aquelas de origem distintas, tais como redes sociais, tecnológicas, biológicas ou de informação. Dentre as características mais frequentemente observadas, podemos citar a baixa distância média entre os vértices, a alta clusterização e a distribuição do grau dos vértices em cauda longa [Albert and Barabasi 2002, Newman 2003b].

Uma outra característica topológica que vem sendo observada é o relacionamento entre vértices similares (ou diferentes) sob algum aspecto de similaridade, ou seja, ao rotu-

---

\*Este trabalho teve suporte do CNPq.

larmos os vértices com seus respectivos tipos, existe uma tendência maior para relacionamentos entre vértices do mesmo tipo ou de tipos bem diferentes [Newman 2003a]. Em redes sociais, onde a rede representa algum relacionamento entre um conjunto de pessoas, é bem estudado e conhecido o fato de que pessoas tendem a se relacionar com pessoas similares, seja o tipo dado pela etnia, idade ou nacionalidade [J. A. Catania and Fullilove 1992]. Este fenômeno de relacionamento entre pares similares é conhecido por *assortative mixing* (AM).

Assortative mixing também vem sendo observado em redes distintas das redes sociais. Em redes par-a-par (P2P) para compartilhamento de arquivos, tais como BitTorrent e Gnutella, *peers* (vértices) que formam a rede possuem conexões com diferentes larguras de banda. Recentemente, observou-se assortative mixing entre os peers de uma rede BitTorrent quando o tipo do peer é dado pela sua largura de banda [Legout et al. 2007], ou seja, peers tendem a se conectar e trocar informação com peers que possuem largura de banda parecida com a sua própria.

Assim como outras propriedades topológicas, assortative mixing também possui implicações fundamentais na funcionalidade do sistema. No caso de sistemas P2P, como o BitTorrent, a presença de assortative mixing na largura de banda está diretamente relacionado à escalabilidade e robustez do sistema. Em particular, um sistema P2P com alta taxa de assortative mixing é mais eficiente e oferece melhor serviço a seus usuários [Legout et al. 2007]. É então fundamental entender por que tais sistemas atingem altas taxas de assortative mixing.

O protocolo BitTorrent para compartilhamento de arquivos é relativamente complexo, sendo formado por diversos mecanismos que operam simultaneamente. Em essência, o protocolo consiste em um processo dinâmico de troca de peers que tenta maximizar a taxa de download dando em troca taxas de upload. No entanto, não há preferência explícita no protocolo por pares com largura de banda similar. O fenômeno observado é decorrência dos mecanismos que definem o protocolo. Mas que mecanismos são estes? Qual é a influência dos parâmetros do sistema, neste processo, que leva ao assortative mixing?

Neste trabalho, estamos interessados em estudar o processo dinâmico do protocolo BitTorrent para entender quais mecanismos do protocolo são responsáveis por induzir o sistema a um estado com alta taxa de assortative mixing. Em particular, propomos um modelo simples que captura a essência da dinâmica do protocolo BitTorrent. Através de simulações, mostramos que o processo dinâmico, que se baseia apenas em regras locais, leva o sistema de um estado inicial aleatório a um estado com alta taxa de assortative mixing. Estudamos ainda o impacto dos parâmetros do modelo neste processo dinâmico. Entre outras observações, nossos resultados indicam que o *unchoke otimista* é fundamental para que o sistema atinja alta taxa de assortative mixing.

Este trabalho está organizado nas seguintes seções: na seção 2, descrevemos o protocolo BitTorrent e discutimos alguns dos trabalhos relacionados; na seção 3, apresentamos o modelo proposto para capturar a dinâmica do protocolo BitTorrent; na seção 4, apresentamos e discutimos os resultados obtidos; finalmente, a seção 5 apresenta nossas considerações finais.

## 2. Trabalhos relacionados

Na primeira parte desta seção iremos descrever o funcionamento do protocolo BitTorrent. Na segunda parte, iremos tratar de *assortative mixing*, definindo precisamente uma métrica para capturar este fenômeno. Finalmente, na terceira parte apresentamos alguns trabalhos de modelagem e medições do BitTorrent e assortative mixing.

### BitTorrent

O BitTorrent é um protocolo para aplicações par-a-par (P2P) para compartilhamento de arquivos. Um arquivo a ser compartilhado é inicialmente dividido em diversos (milhares) pedaços de tamanho fixo. O protocolo então permite que vários *peers* (cada peer representa um usuário do sistema) façam download de pedaços do arquivo e, simultaneamente, upload desses pedaços uns para os outros. Além disso, o sistema é quase que totalmente distribuído, o que o torna bem escalável. Em geral, os peers que possuem o arquivo completo são chamados de *seeds* e os que possuem nenhuma ou algumas partes são conhecidos como *leechers*. Ao conjunto de peers interessados em um mesmo arquivo dá-se o nome de *swarm*.

Neste trabalho, estamos interessados na dinâmica de conexão entre os leechers. Uma entidade central, o *tracker*, é responsável por informar um peer sobre outros peers interessados no mesmo arquivo. Assim, quando um leecher está interessado em baixar um determinado arquivo, ele primeiro entra em contato com o tracker e recebe uma lista com alguns peers, escolhidos aleatoriamente. O novo leecher estabelece conexões TCP com esses peers e passa a fazer requisições por pedaços do arquivo. No entanto, essas requisições não são imediatamente atendidas: existe um mecanismo de *choking/unchoking* que decide quando um peer irá fazer upload para outro.

O algoritmo de choking tenta capturar a essência do mecanismo *tit-for-tat*, cujo objetivo é fomentar a cooperação entre os peers. Em intervalos fixos, um leecher irá decidir para quais outros irá fazer upload com base na taxa de download a que ele recebe dados de cada um. Os peers que ofereceram as maiores taxas de download nos últimos instantes de tempo serão aqueles que receberão um “unchoke”, ou seja, passarão a receber dados caso tenham pedido algum pedaço do arquivo. O número de peers que recebem unchoke era fixo nas versões antigas do protocolo, geralmente igual a 4. Esse tipo de unchoke é conhecido como **unchoke regular**.

Ao entrar no torrent, um leecher não terá nenhum pedaço do arquivo a oferecer e não poderá ser beneficiado com o unchoke regular de um leecher. No entanto, existe também o **unchoke otimista**, que faz com que um peer selecione aleatoriamente algum peer para fazer upload, dentre aqueles que ele conhece. Assim, um usuário que acabou de entrar na rede pode ser escolhido para receber dados e assim obter seus primeiros pedaços do arquivo.

Além do algoritmo de choking, os peers utilizam a política do *bloco mais raro* para determinar quais blocos solicitar aos seus vizinhos. Uma descrição mais detalhada do protocolo BitTorrent original pode ser encontrada em [Cohen 2003].

### Assortative Mixing

Assim como outras propriedades topológicas, assortative mixing (ou dissortative mixing) vem sendo observado em diversas redes que representam sistemas reais. De maneira ge-

ral, assortative mixing ocorre em uma rede cujos vértices podem ser rotulados de acordo com suas características, quando ela exibe uma tendência por relacionamentos entre vértices similares. Tal fenômeno é bem conhecido e estudado em redes sociais, onde a rede representa relacionamento entre um grupo de pessoas. O exemplo abaixo ilustra bem este conceito.

**Tabela 1. Casais no estudo de [J. A. Catania and Fullilove 1992] tabulados por etnia de cada parceiro.**

Homem	Mulher			
	Negra	Hispânica	Branca	Outro
Negro	506	32	69	26
Hispânico	23	308	114	38
Branco	26	46	599	68
Outro	10	14	47	32

Um estudo realizado em [J. A. Catania and Fullilove 1992] mostra que no processo de formação de casais, existe uma forte associação entre as etnias dos parceiros que formam um casal (ver Tabela 1). Em redes sociais, não somente a etnia das pessoas, mas diversas outras características tais como a idade, a nacionalidade e a classe social, influenciam a escolha de relacionamentos, levando a altas taxas de assortative mixing.

O *Assortative Coefficient* é uma métrica utilizada para quantificar o assortative mixing e foi definido em [Newman 2003a] da seguinte forma. Considere uma rede em que cada vértice está rotulado com seu tipo  $i$ , com  $i = 1, \dots, N$ . Seja  $E_{ij}$  o número de arestas em uma rede que conectam vértices do tipo  $i$  com vértices do tipo  $j$ , para  $i, j = 1, \dots, N$ . Seja  $\mathbf{E}$  a matriz formada pelos elementos  $E_{ij}$ . Normalizando  $\mathbf{E}$  pelo somatório de seus elementos, temos a matriz  $\mathbf{e}$ :

$$\mathbf{e} = \frac{\mathbf{E}}{\sum_{i,j} E_{ij}}$$

O assortative coefficient é então calculado da seguinte maneira:

$$r = \frac{\sum_i e_{ii} - \sum_i a_i b_i}{1 - \sum_i a_i b_i} \quad (1)$$

onde  $a_i = \sum_j e_{ij}$  e  $b_i = \sum_j e_{ji}$ . Repare que o assortative coefficient assume o valor **zero** quando não existe nenhuma preferência por relacionamentos entre os tipos de vértices e valor 1 quando os vértices somente se relacionam com outros do mesmo tipo.

### Modelos e Medições do BitTorrent

Alguns modelos foram propostos na literatura para estudar o comportamento e desempenho do BitTorrent. Entre os mais importantes, está o modelo de fluido proposto em [Qiu and Srikant 2004] para estudar a escalabilidade e desempenho do BitTorrent. Entretanto, este modelo não captura o processo dinâmico de troca de vizinhos, pois trabalha com uma abstração mais genérica do protocolo.

Em um outro trabalho de modelagem, [Neglia et al. 2007] estudam o equilíbrio obtido pelo sistema com relação às conexões entre os peers. Tal estudo caracteriza o

equilíbrio do processo dinâmico utilizando uma abordagem da Teoria dos Jogos. No entanto, este modelo não leva em consideração a evolução do processo dinâmico da troca de peers, assim como ignora aspectos importantes do sistema, como o número constante de conexões ativas e o limite no conhecimento dos peers que participam em um swarm. Estes aspectos serão explicitamente capturados no modelo que propomos e, como veremos, têm impacto fundamental no assortative mixing.

Existem ainda diversos trabalhos de medições do BitTorrent que tentam caracterizar a importância dos mecanismos do protocolo em seu desempenho. Um trabalho influente nesta direção é apresentado em [Legout et al. 2006], onde os autores defendem os algoritmos de choking e bloco mais raro como sendo essenciais para o bom desempenho do sistema.

Em um trabalho mais recente, observou-se que a rede de peers participando de um swarm no BitTorrent possui alta taxa de assortative mixing com relação à largura de banda dos vértices [Legout et al. 2007], ou seja, peers tendem a se conectar e trocar informação com peers que possuem larguras de banda semelhantes a sua. Esta observação é importante pois atesta para a alta escalabilidade do protocolo [Bharambe et al. 2006]. Contudo, uma explicação para este fenômeno, que é decorrência dos mecanismos implementados pelo protocolo, não foi apresentada.

Finalmente, alguns modelos vêm sendo propostos para capturar o assortative mixing observado em redes reais. Em particular, [Newman 2003a] propõe um modelo de rede que exhibe assortative mixing. Entretanto, este modelo assume que a taxa de assortative mixing é determinada externamente (é um parâmetro do modelo), e não consequência de algum processo dinâmico sobre a rede. Em um trabalho mais recente, [Grabowski and Kosiński 2006] propõe um modelo que possui um processo dinâmico sobre a rede capaz de gerar uma alta taxa de assortative mixing. Entretanto, tal modelo não pode ser aplicado a redes BitTorrent, pois: 1- foi elaborado a fim de contemplar características específicas de redes sociais, como laços familiares, representados por conexões que não se desfazem ao longo de todo o processo dinâmico; 2- é um modelo síncrono; 3- considera que as características de um nó são modificadas em função dos vizinhos, o que não ocorre, por exemplo, com a banda de um usuário. O modelo proposto no presente trabalho lida com essas três dificuldades sem perder aplicabilidade à outras redes.

### 3. Modelo proposto

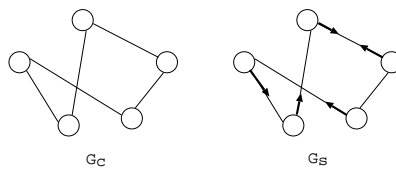
A seguir descreveremos um modelo simplificado para capturar a dinâmica de troca de pares do BitTorrent que leva ao AM.

Quando um leecher  $L$  contacta o tracker para iniciar o download de um arquivo, ele recebe uma lista contendo  $k$  peers escolhidos de forma uniformemente aleatória dentre todos os do swarm. Note que o conhecimento de cada peer define uma rede onde os objetos são os peers propriamente ditos. O conhecimento é um relacionamento bidirecional: se  $i$  conhece  $j$ , então  $j$  conhece  $i$ .

Uma forma natural de representar essa rede é através de um grafo não-direcionado. Por simplicidade, vamos supor que cada peer conhece apenas  $k$  outros peers e, portanto, esta rede pode ser bem representada por um grafo aleatório  $k$ -regular (cada vértice tem grau  $k$ ). A este grafo vamos chamar de “**grafo de conhecimento**” e será representado

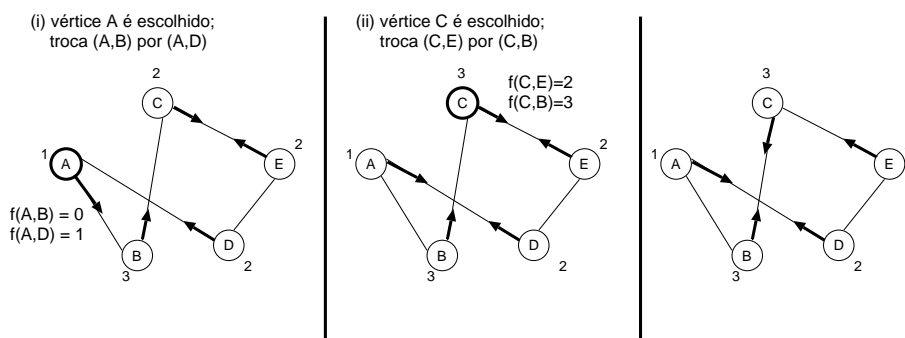
por  $G_c$ . Vamos assumir que nenhum peer sai da rede e, por conseguinte, que esse grafo é estático, isto é, o conhecimento vai se manter imutável ao longo do tempo. Um algoritmo eficiente para geração de um grafo aleatório  $k$ -regular pode ser encontrado em [Steger and Wormald 1999].

No protocolo BitTorrent, o algoritmo de choking/unchoking é executado por todos os peers, conforme descrito na Seção 2. O envio de dados é assimétrico, ou seja,  $i$  estar enviando dados para  $j$  não implica em  $j$  estar enviando dados para  $i$ . Além disso, um peer só pode enviar dados para os peers que ele conhece. Assim, o relacionamento “envio de dados” será codificado no nosso modelo através de uma aresta direcionada, cuja existência está condicionada ao fato dos vértices origem e destino serem vizinhos em  $G_c$ .



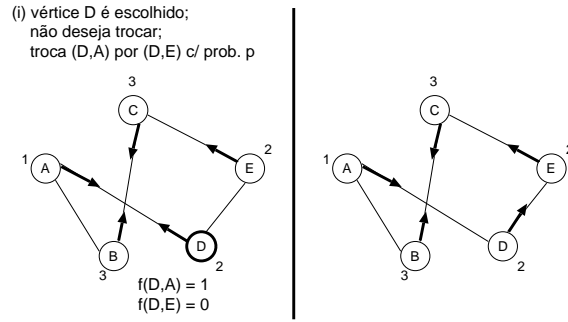
**Figura 1. Exemplo de grafo de conhecimento e grafo de serviço para os parâmetros  $n=5, k=2, c=1$ .**

Seja  $c$  o número de conexões ativas que um peer mantém, ou seja,  $c$  arestas direcionadas representam o fluxo de dados que parte de cada vértice. As conexões ativas no nosso modelo serão representadas por um grafo direcionado  $G_s$ , definido sobre o grafo de conhecimento, conforme mostrado na Figura 1. Este grafo será denominado “**grafo de serviço**”.



**Figura 2. Exemplos de troca regular. Nota-se a preferência pela reciprocidade e pela qualidade de serviço.**

A idéia do algoritmo de choking/unchoking é dar prioridade aos relacionamentos mais valiosos para um peer. No contexto de compartilhamento de arquivos, o relacionamento mais valioso é aquele que é capaz de fornecer mais informação por unidade de tempo. Assim, quando um peer executa esse algoritmo, ele irá enviar mensagens de *unchoke regular* aos peers dos quais conseguiu extrair mais dados. A quantidade de dados fornecida é função, dentre outras coisas, da largura de banda do peer que envia os dados e da largura de banda do peer que os recebe. Mais especificamente, a taxa de transmissão de  $i$  para  $j$  é limitada superiormente por  $\min\{\text{uplink}(i), \text{downlink}(j)\}$ .



**Figura 3. Exemplos de troca otimista. Se o vértice E for o próximo a realizar a troca, ele irá substituir (E,C) por (E,D), aumentando o assortative coefficient.**

No nosso modelo, vamos supor que o uplink e o downlink de um usuário  $i$  são iguais e serão valorados por uma *tag* numérica  $tag(i)$ . Uma *tag* poderá assumir um de  $t$  valores possíveis  $(1, 2, \dots, t)$ . Em uma rede com  $n$  vértices, cada valor de *tag* será atribuído a  $n/t$  vértices. Assumindo que a largura de banda de um peer e de seu vizinho sejam os únicos fatores que influenciam o algoritmo de choking, podemos definir uma função utilidade  $f$  que captura a prioridade de um vizinho para receber o unchoke:

$$f(i, j) = \begin{cases} 0 & \text{se } \nexists(j, i) \in G_s \\ \min\{tag(i), tag(j)\} & \text{caso contrário} \end{cases}$$

No BitTorrent, a cada vez que o algoritmo de choking é executado em um nó,  $c - 1$  peers recebem o unchoke regular e 1 peer recebe o unchoke otimista. Note que todos as conexões ativas antes da execução do algoritmo podem ser substituídas se (i) um conjunto de  $c - 1$  peers fizerem upload para um usuário  $i$  a taxas de transmissões mais elevadas do que os peers que  $i$  está servindo atualmente e (ii) o peer selecionado pelo unchoke otimista não estiver sendo servido ainda. Nosso modelo irá incorporar este mecanismo de troca de arestas em  $G_s$  de maneira simplificada, permitindo que um vértice troque apenas uma aresta de saída a cada execução do algoritmo, ou seja, deixe de fazer upload para um de seus vizinhos e passe a fazer para outro.

A dinâmica do modelo será a seguinte: a cada passo, um vértice  $i$  é aleatoriamente escolhido para realizar a troca de aresta. A conexão com o vértice  $j$  menos valioso dentre os que estão sendo servidos atualmente será fechada e será estabelecida uma conexão com o vértice  $k$  mais valioso dentre os que não estão sendo servidos por  $i$ , desde que  $f(i, k) > f(i, j)$ . Isto é, vamos substituir a aresta  $(i, j)$  pela aresta  $(i, k)$  em  $G_s$ . Formalmente, seja  $j_{min} = \arg \min_j f(i, j) \quad \forall (i, j) \in G_s$  e  $k_{max} = \arg \max_k f(i, k) \quad \forall (i, k) \in G_c$  e  $(i, k) \notin G_s$ . Se  $f(i, j_{min}) < f(i, k_{max})$ , então a aresta direcionada  $(i, j_{min})$  é substituída pela aresta direcionada  $(i, k_{max})$ . Essa troca equivale ao unchoke regular e está ilustrada na Figura 2. Ela será chamada de *troca regular*.

A dinâmica da forma como foi descrita até o momento faria com que as arestas com origem em  $i$  fossem aos poucos encontrando os vértices que possuem arestas para  $i$ . Isso não permitiria que  $i$  pudesse descobrir muitos vértices com a mesma *tag*, pois suas trocas estariam limitadas a “casar arestas” com os vértices que incidem em  $i$ . Como veremos na seção seguinte, a *troca otimista* (equivalente ao unchoke otimista) é fundamental para a ocorrência do AM.

Se ao ser selecionado, um vértice não deseja realizar uma troca regular, ou seja,  $f(i, j_{min}) \geq f(i, k_{max})$ , ele irá fazer uma troca otimista com probabilidade  $p$ , onde  $p$  é um parâmetro do modelo. Nesse caso, ele irá substituir sua aresta  $(i, j_{min})$  por uma aresta  $(i, k) \in G_c$  e  $(i, k) \notin G_s$  escolhida aleatoriamente (ver Figura 3).

Apesar das diversas simplificações, o modelo permite estudar ambientes mais reais e parâmetros mais diversificados. Podemos, por exemplo, levar em consideração a assimetria entre as bandas de download e upload dos peers definindo uma tag como sendo um par (downlink, uplink) e mudando a definição da função acima para representar o novo gargalo de transmissão. A função utilidade pode ainda ser descrita de forma arbitrária para outros tipos de rede, desde que caracterize corretamente a prioridade que alguns objetos têm para estabelecer relacionamentos. Variações na estrutura do grafo do conhecimento e no grafo de serviço, além de distribuição não-uniforme das tags também podem ser capturadas pelo modelo. Vale notar que o modelo apresentado é bem geral e poderia ser prontamente utilizado para o estudo de outras redes.

#### 4. Experimentos e resultados

No conjunto de experimentos que segue, avaliaremos, através de simulação, o *assortative coefficient* do grafo de serviço ao longo do tempo. Serão realizados experimentos variando individualmente alguns parâmetros do modelo a fim de estudar sua influência.

O modelo possui os seguintes parâmetros:

- $n$ : número de vértices
- $k$ : grau do vértice no grafo de conhecimento  $G_c$
- $c$ : grau do vértice no grafo de serviço  $G_s$
- $t$ : número de tags diferentes
- $p$ : probabilidade de troca otimista

Para os experimentos a seguir, os seguintes parâmetros são utilizados a menos que o contrário seja dito:  $n = 1000, k = 50, c = 10, t = 2, p = 0.40$ .

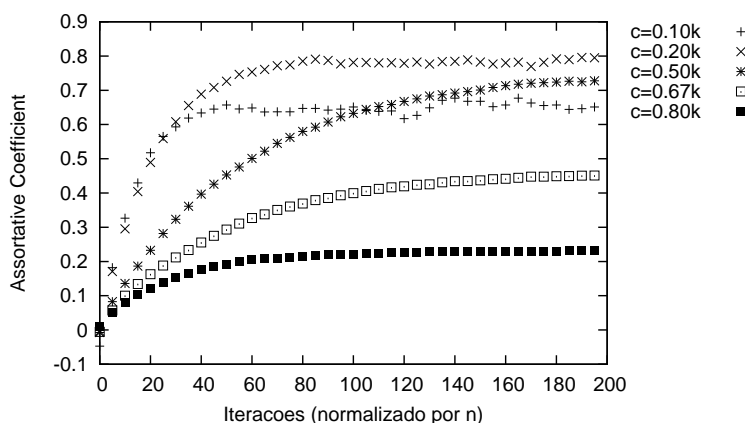
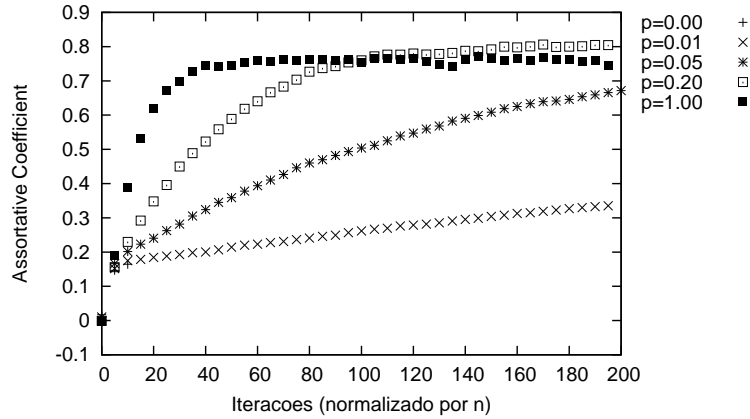


Figura 4. Variação de  $r$  em função de  $c$

Mais do que analisar os valores de  $c$  e  $k$  separadamente, estamos interessados na taxa de AM quando uma fração  $c/k$  dos peers está conectada. Para isso, faremos variar o grau de saída do vértice em  $G_s$ , mantendo fixo o número de vizinhos em  $G_c$ .





**Figura 5. Variação de  $r$  em função de  $p$**

Intuitivamente, diminuindo a fração  $c/k$  de vizinhos para o qual um faz upload, a taxa de AM aumenta, pois é mais fácil encontrar grupos pequenos de vértices com o mesmo tag. Da mesma forma, para valores muito altos de  $c/k$ , a fração de vizinhos com a mesma tag que um vértice passa a ser menor que  $c/k$ , limitando assim o valor do assortative coefficient. No entanto, devemos considerar os efeitos da troca otimista quando  $c$  é muito pequeno, pois neste caso muitas vezes estamos trocando um vértice com a mesma tag por um vértice com tag diferente e, como o número de vértices que recebe upload é pequeno, essa troca irá reduzir significativamente o valor do assortative coefficient.

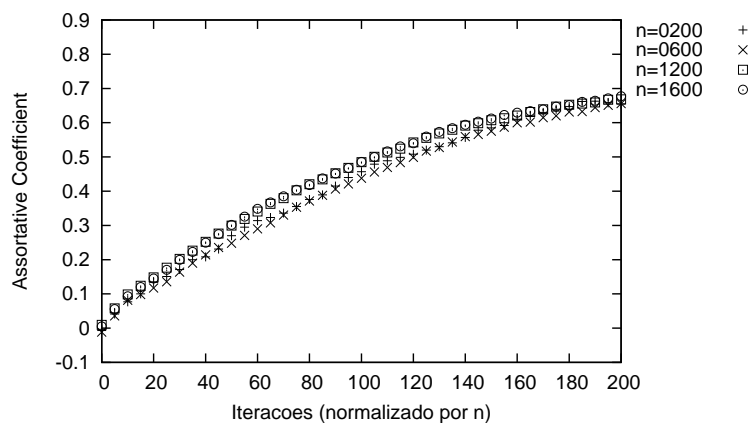
A Figura 4 mostra a evolução do assortative coefficient  $r$  (ver Equação (1)) ao longo do tempo, para diferentes frações  $c/k$ . Note que para  $c = 0.80k, 0.67k, 0.50k, 0.20k$  o valor do assortative coefficient alcançado é estritamente crescente. Porém, para  $c = 0.10k (c = 5)$ , a probabilidade  $p = 0.40$  faz com que: (i) o assortative coefficient alcançado seja menor e (ii) a curva apresente mais variabilidade que as demais. Observe que como  $t = 2$ , é inevitável que para  $c = 0.67k, 0.80k$  alguns peers dêem unchoke em peers que possuem tags diferentes. Na subseção seguinte, apresentamos uma análise que relaciona os parâmetros  $c, k$  e  $t$  ao valor máximo que  $r$  pode assumir.

Conforme discutimos na Seção 3, a troca otimista é fundamental para que um vértice descubra os vizinhos que possuem a mesma tag que ele. Sem esse recurso, o modelo se resumiria praticamente a “casar” arestas direcionadas. Inicialmente, um vértice de  $tag = 2$  que executa a troca otimista substituindo sua aresta “menos valiosa”, irá manter ou aumentar o assortative coefficient, pois quando a taxa de AM é baixa, sua aresta “menos valiosa” provavelmente incide em um vértice de  $tag = 1$ . Por isso, valores altos de  $p$  fazem com que o assortative coefficient cresça mais rapidamente. Por outro lado, depois que  $r$  atinge valores altos, a troca otimista faz com os vértices substituam arestas incidentes a vértices de mesma tag, por arestas que incidem em vértices de tag diferente. Logo, valores altos de  $p$  fazem com que o valor final de  $r$  seja mais baixo. Para verificar esses fatos, observamos o assortative coefficient para diferentes valores de  $p$ .

Os resultados mostrados na Figura 5 devem ser analisados em 2 partes. As curvas mostram a evolução de  $r$  ao longo do tempo, para diferentes valores de  $p$ . Considerando as curvas em que  $p > 0$ , concluímos que quanto maior o  $p$ , mais rápido ocorre a con-

vergência. Entretanto, o valor de  $r$  atingido é menor. A curva  $p = 0.01$ , por exemplo, converge após  $4.2 \times 10^3$  iterações, e o valor de  $r$  nesse instante é 0.883.

Considerando agora  $p = 0$ , observamos que a dinâmica de troca de arestas pára após 14 iterações de cada vértice, obtendo  $r = 0.165$ , ou seja, baixa taxa de AM. Como nesse caso não há unchoke otimista, podemos garantir após passar por todos os vértices que esse sistema jamais será modificado.



**Figura 6. Variação de  $r$  em função de  $n$**

As interações de um vértice realiza com os outros são locais, o que indica que o número de vértices no sistema não influencia no valor do assortative coefficient. Isto foi confirmado nos nossos experimentos (ver Figura 6). Realizamos também experimentos com mais de dois tags e, apesar do fato de menores valores do assortative coefficient terem sido alcançados, as conclusões derivadas foram as mesmas. Omitimos esses resultados por limitações de espaço.

## Resultados analíticos

Para um dado grafo de conhecimento, o valor máximo que o assortative coefficient pode atingir é um número fixo. Note que este coeficiente seria facilmente obtido se as tags dos vizinhos fossem todas conhecidas e houvesse uma preferência explícita pelo fornecimento de serviço a vértices de mesma tag. Ainda assim, existe uma probabilidade não-nula de alcançarmos este coeficiente com o modelo proposto. Sendo o modelo do grafo de conhecimento aleatório, o valor máximo do assortative coefficient é, portanto, uma variável aleatória  $R_{max}$ . Estamos interessados em calcular a esperança dessa variável, isto é,  $E[R_{max}]$ .

A contribuição individual de um vértice para  $R_{max}$  é dada em função das tags dos vizinhos a quem ele está conectado. Seja  $X$  o número de vizinhos com a mesma tag que um certo vértice. Se  $X > c$ , ou seja, maior que o número de uploads que ele faz, então  $R_{max}$  é obtido quando todos os  $c$  uploads desse vértice são feitos para outros de mesma tag. Por outro lado, se  $X < c$ ,  $R_{max}$  é obtido quando  $X$  arestas são dispostas para os vizinhos de mesma tag. Logo, a contribuição de um vértice para o termo  $\sum_i e_{ii}$  presente na Equação (1) é  $1/n \times \min\{X/c, 1\}$ . Pela definição do modelo,  $X$  é uma v.a. com distribuição *binomial*( $k, 1/t$ ), desde que o número de vértices de cada tipo seja maior

que  $k$ . Então, temos:

$$\sum_i e_{ii} = \sum_{i=1}^{c-1} P(X = i) \times \frac{i}{c} + P(X \geq c) \times 1 \quad (2)$$

Como o número de vértices de cada tag é igual a  $n/t$  e o grau de saída é  $c$  para todo vértice, a fração de arestas que partem de vértices de tag  $i$  é  $a_i = 1/t$  para todas as tags. Para calcular  $E[R_{max}]$ , vamos assumir que  $b_i = 1/t$  para todo  $i$ , ou seja, o número de arestas que chegam em vértices de tag  $i$  é igual para todas as tags. Essa suposição é bem realista, pois a função utilidade empregada captura a noção de reciprocidade, fazendo com que o número de arestas incidentes a cada vértice esteja muito próximo de  $c$ . Logo,

$$\begin{aligned} E[R_{max}] &= \frac{\sum_i e_{ii} - \frac{1}{t^2}}{1 - \frac{1}{t^2}} \\ &= \frac{\sum_i e_{ii} \times t^2 - 1}{t^2 - 1} \end{aligned} \quad (3)$$

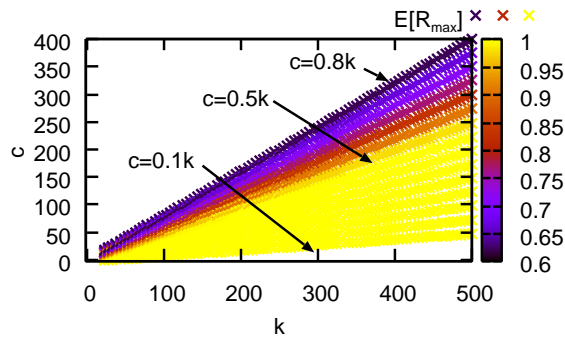


Figura 7. Valores de  $E[R_{max}]$  para  $c=0.10k, 0.15k, \dots, 0.80k$

O gráfico da Equação (3), mostrado na Figura 7, indica uma mudança de comportamento quando  $c < k/2$ , conforme havíamos observado na Figura 4. Neste caso,  $E[R_{max}]$  se aproxima rapidamente de 1. Note que  $E[R_{max}]$  independe de  $n$ .

## 5. Conclusões

Neste trabalho, propusemos um modelo simplificado da dinâmica de troca de conexões entre os peers no protocolo BitTorrent capaz de capturar o fenômeno *assortative mixing*. O modelo não incorpora a preferência dos vértices por outros do mesmo tipo de forma explícita e leva em conta que um vértice só pode manter conexões com um subconjunto dos vértices do grafo, assim como acontece no BitTorrent e em outras redes reais.

Utilizamos o assortative coefficient para quantificar o assortative mixing ao longo do tempo. Analisamos as curvas para a variação individual de alguns parâmetros do modelo. Observamos que o valor máximo desse coeficiente depende apenas do número de vértices que cada nó conhece ( $k$ ), do número de vértices a quem ele oferece upload ( $c$ ) e do número de tags diferentes na rede ( $t$ ). No entanto, o tempo de convergência e o valor final do *assortative coefficient* dependem fortemente da probabilidade de troca

otimista ( $p$ ). Identificamos um compromisso no valor de  $p$ : se a troca otimista ocorre com alta probabilidade, a convergência do processo ocorre mais rapidamente, mas atinge taxas mais baixas de AM.

Conforme observado em trabalhos anteriores [Legout et al. 2006], o unchoke otimista é essencial para o bom desempenho do BitTorrent. A principal contribuição do modelo aqui apresentado foi mostrar que, além disso, o unchoke otimista é também fundamental para a formação de clusters por largura de banda, o que por sua vez contribui para alta escalabilidade desse sistema.

## Referências

- [Albert and Barabasi 2002] Albert, R. and Barabasi, A. L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97.
- [Bharambe et al. 2006] Bharambe, A. R., Herley, C., and Padmanabhan, V. N. (2006). Analyzing and improving a bittorrent network’s performance mechanisms. In *Proceedings of IEEE INFOCOM 2006*, pages 1–12, Barcelona, Spain.
- [Cohen 2003] Cohen, B. (2003). Incentives build robustness in bittorrent. In *Proceedings of the P2PEcon’03*, Berkeley, CA.
- [Grabowski and Kosiński 2006] Grabowski, A. and Kosiński, R. A. (2006). Evolution of a social network: The role of cultural diversity. *Phys. Rev. E*, 73(1):016135.
- [J. A. Catania and Fullilove 1992] J. A. Catania, T. J. Coates, S. K. and Fullilove (1992). The population-based AMEN (AIDS in Multi-Ethnic Neighborhood) study. *Am. J. Public Health*, 82:284–287.
- [Legout et al. 2007] Legout, A., Liogkas, N., Kohler, E., and Zhang, L. (2007). Clustering and sharing incentives in bittorrent systems. In *SIGMETRICS ’07: Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 301–312, New York, NY, USA. ACM.
- [Legout et al. 2006] Legout, A., Urvoy-Keller, G., and Michiardi, P. (2006). Rarest first and choke algorithms are enough. In *IMC ’06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 203–216, New York, NY, USA. ACM.
- [Neglia et al. 2007] Neglia, G., Presti, G. L., Zhang, H., and Towsley, D. (2007). A network formation game approach to study bittorrent tit-for-tat. In *NET-COOP 2007*, pages 13–22.
- [Newman 2003a] Newman, M. E. J. (2003a). Mixing patterns in networks. *Phys. Rev. E*, 67(2):026126.
- [Newman 2003b] Newman, M. E. J. (2003b). The structure and function of complex networks. *SIAM Review*, 45:167–256.
- [Qiu and Srikant 2004] Qiu, D. and Srikant, R. (2004). Modeling and performance analysis of bittorrent-like peer-to-peer networks. In *SIGCOMM ’04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 367–378, New York, NY, USA. ACM.
- [Steger and Wormald 1999] Steger, A. and Wormald, N. C. (1999). Generating random regular graphs quickly. *Comb. Probab. Comput.*, 8(4):377–396.