Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciência

Faculdade de Engenharia

Marco Antonio D'Alessandro Costa

Análise de desempenho de um protocolo BitTorrent ciente de localização em redes corporativas

Rio de Janeiro

2015

Marco Antonio D'Alessandro Costa

Análise de desempenho de um protocolo BitTorrent ciente de localização em redes corporativas

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre em Ciências, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Redes de Telecomunicações

Orientador: Prof. Dr. Marcelo Gonçalves Rubinstein

Rio de Janeiro

2015

CATALOGAÇÃO NA FONTE

UERJ / REDE SIRIUS / BIBLIOTECA CTC/B

C837 Costa, Marco Antonio D'Alessandro.
Análise de desempenho de um protocolo BitTorrent ciente de localização em redes corporativas / Marco Antonio D'Alessandro Costa. - 2015.
68 f.

Orientador: Marcelo Gonçalves Rubinstein.
Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia.

1. Engenharia Eletrônica. 2. Redes corporativas – Dissertações. 3. Informação – Compartilhamento – Dissertações. I. Rubinstein, Marcelo Gonçalves. II. Universidade do Estado do Rio de Janeiro. III. Título.

CDU 004.05

Assinatura

Data

Marco Antonio D'Alessandro Costa

Análise de desempenho de um protocolo BitTorrent ciente de localização em redes corporativas

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre em Ciências, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Redes de Telecomunicações.

Aprovado em 23 de Julho de 2015.

Banca Examinadora:

_____

Prof. Dr. Marcelo Gonçalves Rubinstein (Orientador)

Faculdade de Engenharia - UERJ

_____

Prof. Dr. Igor Monteiro Moraes

Universidade Federal Fluminense - UFF

_____

Prof. Dr. Pedro Braconnot Velloso

Universidade Federal do Rio de Janeiro - UFRJ

_____

Prof. Dr. Rodrigo de Souza Couto

Faculdade de Engenharia – UERJ

Rio de Janeiro

2015

DEDICATÓRIA

Dedico este trabalho a minha família e, em especial, a minha filha Alice que pacientemente aguardou o pai terminar de estudar, fazer as simulações e escrever esta dissertação, antes de ter mais tempo livre para brincar com ela.

AGRADECIMENTOS

Agradeço ao professor Marcelo G. Rubinstein, meu orientador, pelos conhecimentos passados durante a pós-graduação e por todo o auxílio prestado durante as atividades que resultaram nesta dissertação.

Agradeço à minha família pela compreensão e auxílio nesta minha jornada, pois nos vários momentos em que estive ausente sempre tive o apoio necessário.

Agradeço ao Programa de Pós-Graduação em Engenharia Eletrônica da UERJ, pela oportunidade que me deram de fazer parte de um maravilhoso ambiente de pesquisa e, em especial, aos professores Alexandre Sztajnberg e Nival Nunes, pela oportunidade de expandir os meus conhecimentos.

RESUMO

COSTA, M. A. D'A. *Análise de desempenho de um protocolo BitTorrent ciente de localização em redes corporativas*. 2015. 68 f. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2015.

Hoje em dia, distribuições de grandes volumes de dados por redes TCP/IP corporativas trazem problemas como a alta utilização da rede e de servidores, longos períodos para conclusão e maior sensibilidade a falhas na infraestrutura de rede. Estes problemas podem ser reduzidos com utilização de redes par-a-par (P2P). O objetivo desta dissertação é analisar o desempenho do protocolo BitTorrent padrão em redes corporativas e também realizar a análise após uma modificação no comportamento padrão do protocolo BitTorrent. Nesta modificação, o rastreador identifica o endereço IP do par que está solicitando a lista de endereços IP do enxame e envia somente aqueles pertencentes à mesma rede local e ao semeador original, com o objetivo de reduzir o tráfego em redes de longa distância. Em cenários corporativos típicos, as simulações mostraram que a alteração é capaz de reduzir o consumo médio de banda e o tempo médio dos *downloads*, quando comparados ao BitTorrent padrão, além de conferir maior robustez à distribuição em casos de falhas em enlaces de longa distância. As simulações mostraram também que em ambientes mais complexos, com muitos clientes, e onde a restrição de banda em enlaces de longa distância provoca congestionamento e descartes, o desempenho do protocolo BitTorrent padrão pode ser semelhante a uma distribuição em arquitetura cliente-servidor. Neste último caso, a modificação proposta mostrou resultados consistentes de melhoria do desempenho da distribuição.

Palavras-chave: BitTorrent; Redes Corporativas; Desempenho; P2P; Distribuição de conteúdo.

ABSTRACT

COSTA, M. A. D'A. *Performance Analysis of a locality-aware BitTorrent protocol in enterprise networks*. 2015. 68 f. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2015.

Nowadays, distributions of large volumes of data over enterprise TCP/IP networks bring problems such as high network and server utilizations, long periods for completion, and greater sensitivity to flaws in network infrastructure. These problems can be reduced with the use of Peer-to-Peer networks (P2P). The aim of this work is to analyze the performance of the standard BitTorrent protocol in corporate networks and also perform the analysis after a change in the default behavior of the BitTorrent protocol. In this modification, the tracker identifies the peer IP address requesting the list of IP addresses of the swarm and sends only those belonging to the same LAN and to the original seeder, with the aim of reducing traffic on WAN links. In typical enterprise scenarios, the simulations showed that the change is able to reduce the average bandwidth consumption and the average time of downloads compared with standard BitTorrent, and give greater robustness to the distribution in case of failure of WAN links. The simulations also showed that in more complex network environments, with many clients, and where the bandwidth restriction on long distance links causes congestion and packet drops, the performance of standard BitTorrent protocol can be similar than a distribution in client-server architecture. In the latter case, the proposed change showed consistent results in improving distribution performance.

Keywords: BitTorrent; Enterprise Networks; Performance; P2P; Content distribution.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS

ADSL	*Asymmetric Digital Subscriber Line*
ALM	*Application-Level Multicast*
AS	*Autonomous System*
BGP	*Border Gateway Protocol*
BNS	*Biased Neighbor Selection*
BU	*Biased Unchoking*
CDN	*Content Delivery Network*
CPU	*Central Processing Unit*
DDoS	*Distributed Denial of Service*
DNS	*Domain Name System*
HTTP	*Hypertext Transfer Protocol*
IP	*Internet Protocol*
ISP	*Internet Service Provider*
NAT	*Network Address Translation*
P2P	*Peer-to-peer*
PNS	*Network Locality Preference*
PPS	*Piece Distribution Preference*
PS	*Preference Score*
RIR	*Regional Internet Registry*
RTP	*Real-time Transport Protocol*
RTSP	*Real Time Streaming Protocol*
RTT	*Round-Trip Time*
SHA1	*Secure Hash Algorithm*
TCP	*Transmission Control Protocol*
TTL	*Time to Live*
URL	*Uniform Resource Locator*

SUMÁRIO

INTRODUÇÃO

Hoje em dia, é cada vez mais comum a necessidade de transferência de grandes massas de dados (acima de 100 Mbytes) por redes TCP/IP (*Transmission Control Protocol*/*Internet Protocol*), muitas vezes para mais de um cliente simultaneamente. Os dados vão desde arquivos de mídia (áudio e vídeo) até replicações de banco de dados, atualizações de vacinas de programas de antivírus e sistemas operacionais, em que cada arquivo pode ter desde dezenas de kbytes até mais de 10 Mbytes (MICROSOFT, 2015). O destino desses dados pode ser desde grandes servidores localizados em *data centers*, onde são comuns redes Gigabit Ethernet, até microcomputadores pessoais de uso doméstico, que utilizam enlaces via cabo ou ADSL (*Asymmetric Digital Subscriber Line*) para acesso à Internet e redes locais sem fio. Em ambos os exemplos, podem ocorrer problemas específicos devido ao grande volume de dados da transferência, detalhados a seguir:

- **alta utilização dos recursos de rede** – Arquiteturas do tipo cliente-servidor, em geral, têm limitações geradas pelas redes que conectam o cliente ao servidor, principalmente enlaces de longa distância, pois os recursos dessas redes são limitados. O acesso concorrente de vários clientes a um servidor só piora esta situação, provocando lentidão na transferência. Este problema pode ser reduzido balanceando a carga por vários servidores, tanto de maneira centralizada (em apenas um *data center*) ou distribuindo geograficamente os servidores (em vários *data centers*);
- **longo período de tempo para conclusão** – Atualizações de sistemas operacionais e, principalmente, de vacinas de antivírus devem ser realizadas o mais rápido possível, pois uma demora na atualização pode expor um computador, servidor ou até mesmo toda uma rede a vulnerabilidades de segurança, como vírus, *worms* e ataques DDoS (*Distributed Denial-of-Service*);
- **maior sensibilidade a falhas** – Uma arquitetura cliente-servidor com apenas uma fonte de dados (ou com as fontes no mesmo local) pode gerar problemas quanto à disponibilidade em caso de falha do servidor, já que é esperado que o envio dos dados demore um longo período de tempo. Desta maneira, deve se dar preferência por distribuir os servidores geograficamente;
- **necessidade de recuperação dos dados em caso de falha na transferência** – Em caso de alguma falha (do cliente, do servidor ou da rede), todos os dados já

transferidos podem ser perdidos. Então, é necessário que a aplicação que faz a transferência tenha algum recurso para continuar o processo do ponto em que foi interrompido.

Os problemas citados são especialmente impactantes em redes corporativas, onde há uma busca contínua por redução de custos, segurança da informação e alta disponibilidade dos dados. De maneira geral, podemos definir uma rede corporativa como uma rede que atende uma ou mais empresas e é composta por várias redes locais interligadas por enlaces de longa distância ou metropolitanos, que podem ser próprios ou contratados. A topologia de interligação entre as redes locais pode variar, desde uma topologia em estrela a uma topologia de malha, ou até uma combinação de ambas. Usualmente, os enlaces de longa distância têm menos banda e mais latência que as redes locais, então podem se comportar como gargalos na transferência de dados entre as redes locais que compõe a rede corporativa. Além disso, ampliar os circuitos para diminuir possíveis gargalos pode não ser uma opção viável devido ao custo financeiro envolvido. Outra característica é que, em geral, para facilitar a administração da rede corporativa, o endereçamento IP é bem estruturado e dividido em faixas de endereço por região ou localidade. Nas redes corporativas, geralmente se utilizam as faixas de endereçamento privado, por exemplo, a faixa 10.0.0.0/8. Desta maneira, devido a grande quantidade de endereços disponíveis (na faixa 10.0.0.0/8 são mais de 16 milhões de endereços) para serem distribuídos pelas redes locais e de longa distância dentro da rede corporativa, raramente há a necessidade de reutilização de endereços IP, o que torna desnecessária a utilização de NAT (*Network Address Translation*).

Manter os programas atualizados, especialmente os antivírus e sistemas operacionais, é fundamental para segurança de computadores (CERT.BR, 2012). Esta questão ganha importância especial quando se trata de redes corporativas, pois um ataque bem sucedido à rede de computadores de uma empresa pode ter impacto direto no lucro ou na imagem corporativa desta. Assim, o ideal é que a distribuição de cada atualização seja realizada no menor tempo possível, a fim de diminuir as oportunidades de ataque aos computadores.

Deste modo, uma distribuição de dados em uma rede corporativa, feita de maneira ideal, deve ser orientada a utilizar os recursos de rede de maneira eficiente, no menor tempo possível e ser tolerante a falhas.

Há muito tempo, o problema das grandes transferências de dados já é discutido no ambiente de Internet. Como solução, são propostas as redes CDN (*Content Distribution Networks* - Redes de Distribuição de Conteúdo) ou redes Par-a-par (P2P). (PASSARELLA, 2012)

CDN

As CDNs lidam com um problema crucial no uso convencional da Web: como permitir que um editor de conteúdo faça o seu conteúdo ser amplamente disponibilizado na Web. O dimensionamento da capacidade dos enlaces de acesso, da CPU (*Central Processing Unit*) dos servidores, da capacidade de disco, etc. deve levar em consideração a expectativa de popularidade do conteúdo do site. Ainda assim, os editores de conteúdo não podem dimensionar os sites pelos picos de carga, pois isto não é economicamente viável.

As CDNs lidam com este problema replicando o conteúdo em diferentes sites e, possivelmente, em redes de ISPs (*Internet Service Providers*) diferentes. A ideia principal é replicar o conteúdo, e redirecionar as requisições para uma das réplicas de acordo com alguma regra de seleção.

Uma CDN é, tipicamente, implementada e mantida por apenas um operador que tem a posse da rede e cobra dos provedores de conteúdo e/ou dos donos de sites pelos seus serviços. As CDNs são totalmente transparentes aos usuários da Web, já que os usuários são automaticamente (e de maneira transparente) redirecionados para um site replicado, que é considerado o mais apropriado para tornar mais eficiente o acesso ao conteúdo.

Servidores *surrogates* são os nós da CDN onde o conteúdo Web é replicado. Tipicamente, eles são servidores dedicados e cuidadosamente colocados em lugares estratégicos da Internet. A decisão sobre o local onde os *surrogates* serão colocados é crítica para o sucesso da CDN.

O segundo ponto crítico é a decisão de qual conteúdo será replicado. A replicação pode ser de parte do site ou dele todo. O terceiro ponto crítico é identificar quantas cópias deverão ser realizadas e em quantos *surrogates*.

Segundo Passarella (2012), a replicação de conteúdo nas CDNs podem seguir três métodos distintos:

- **envio cooperativo** (*Cooperative push*) – Onde o conteúdo é proativamente replicado nos *surrogates*, antes que os clientes façam a solicitação. A decisão do que e de onde replicar é baseada na identificação do melhor local para as réplicas;

- **busca não-cooperativa** (*Non-cooperative pull*) – Onde, se uma requisição ao *surrogate* não pode ser atendida, ele busca o conteúdo no servidor de origem, faz cache desta informação e envia para o usuário;
- **busca cooperativa** (*Cooperative pull*) – Onde, se uma requisição ao *surrogate* não pode ser atendida, ele primeiro busca os dados em outros *surrogates* onde o conteúdo pode estar armazenado. Depois do conteúdo encontrado, o *surrogate* faz cache desta informação e envia para o usuário.

A decisão do que será replicado proativamente pode seguir as seguintes estratégias:

- **aleatória** – Cada parte do conteúdo e o *surrogate* são definidos de acordo com uma distribuição aleatória uniforme;
- **popularidade** – Cada *surrogate* cria um ranking para os objetos de acordo com o histórico de requisições. Este ranking é usado para selecionar em cada *surrogate* que objetos armazenar;
- **ganancioso local** – Cada *surrogate* cria um ranking para os objetos de acordo com a expectativa de redução de custos que será gerada se o conteúdo estiver localmente replicado. Este ranking é usado para selecionar em cada *surrogate* que objetos armazenar;
- **ganancioso global** – Um ranking é elaborado entre os *surrogates* e, a cada passo, o par (objeto e *surrogate*) selecionado é aquele que provê a maior redução de custo entre os pares.

Redirecionamento de Requisição

O redirecionamento de requisição inclui o conjunto de mecanismos usados por uma CDN para mapear uma requisição de cliente por um objeto da Web para um *surrogate*. O mecanismo deve ser transparente para os usuários finais, isto é, os clientes devem obter as páginas da Web como se fossem as do servidor original.

Um cliente Web faz a requisição por um objeto fornecendo a sua URL (*Uniform Resource Locator*) para o navegador. A requisição é transmitida para um bloco de redirecionamento, que seleciona um *surrogate* onde o objeto requisitado deve estar armazenado. O navegador requisita o objeto do *surrogate* indicado. Os *surrogates* podem estar, ou não, organizados em grupos cooperativos. Se eles estiverem, o mecanismo de indexação é disponibilizado no grupo, mapeando os objetos para suas localizações nos *surrogates* do grupo. Neste caso, o índice é procurado e o *surrogate* que armazena o objeto é identificado. O objeto finalmente é enviado ao navegador. Se o objeto não está disponível em

nenhum *surrogate* do grupo, o navegador é redirecionado ao servidor de origem. Se os *surrogates* não formam grupo, o objeto está no *surrogate* identificador ou ele tem que ser buscado no servidor de origem.

A técnica mais popular de redirecionamento é o redirecionamento por DNS (*Domain Name System*). Este método é utilizado comercialmente pela Akamai. Outro método muito usado é a reescrita de URL.

Apesar de ser uma solução popular na Internet, nem sempre utilizar CDNs em redes corporativas é a melhor solução. Problemas como a falta de suporte especializado e manutenção de servidores em localidades remotas e *offshore* podem prejudicar a escolha de uma CDN para distribuição de dados, pois qualquer problema nos servidores que replicam conteúdo localmente pode deixar a solução indisponível por um longo período de tempo. Neste caso, os clientes teriam que buscar o conteúdo em outras localidades, acabando com a vantagem de distribuir os dados através de um servidor local.

P2P

Uma rede par-a-par (P2P) é formada por pares que dividem igualmente a responsabilidade de prover serviço entre eles de um modo cooperativo. Os pares voluntariamente se juntam a rede e contribuem com recursos. Além do custo do acesso à Internet, não há nenhum outro custo em se juntar a uma rede par-a-par. Do ponto de vista técnico, as redes par-a-par somente requerem que os pares executem um software, e não requerem nenhum serviço do núcleo da Internet além da pilha TCP/IP. As redes par-a-par são, em princípio, auto-escaláveis.

A operação de qualquer sistema de distribuição de conteúdo par-a-par depende de uma rede de pares e das conexões entre eles. Esta rede é formada em cima, e de maneira independente, de uma rede física e é chamada de rede de “sobreposição”. As redes de sobreposição podem ser distinguidas em termos da sua centralização e estrutura de acordo com Androutsellis-Theotokis e Spinellis (2004) e Passarella (2012).

Quanto à centralização, uma rede de sobreposição pode ser:

- **puramente descentralizada** – Todos os pares da rede fazem exatamente as mesmas tarefas, atuando como servidores e clientes, e não há coordenação central para as suas atividades;

- **parcialmente centralizadas** – A base é a mesma das redes puramente descentralizadas, mas alguns nós assumem papéis mais importantes, atuando como índices centrais locais para os arquivos compartilhados pelos pares locais. Estes pares são chamados de "super-pares". Os super-pares não são pontos únicos de falha numa rede par-a-par, já que eles são designados dinamicamente. Se um deles falhar, a rede age para substituir este par por outro;
- **híbrida descentralizada** – Nestes sistemas, há um servidor central que facilita a interação entre pares mantendo diretórios de metadados, descrevendo os arquivos compartilhados que estão armazenados pelos pares. Ainda que as interações fim-a-fim e as trocas de arquivos sejam feitas diretamente entre os pares, os servidores centrais facilitam essas interações através da realização de buscas e identificando os pares que armazenam os arquivos. Nesse tipo de arquitetura, o servidor central é um ponto único de falha.

Quanto à estrutura, uma rede de sobreposição pode ser:

- **não-estruturada** – A localização do conteúdo não tem relação nenhuma com a rede de sobreposição. Numa rede não-estruturada, o conteúdo precisa ser localizado. Os mecanismos de busca podem variar desde métodos de força bruta, como uma "inundação" de requisições na rede, até outras estratégias mais sofisticadas. Sistemas não-estruturados são geralmente mais apropriados para acomodar uma população de pares altamente variável;
- **estruturada** – A topologia da rede de sobreposição é rigidamente controlada e os arquivos (e os ponteiros para eles) são colocados em lugares precisamente especificados. Esses sistemas proveem um mapeamento entre o conteúdo e a localização na forma de uma tabela de roteamento distribuída, de modo que as requisições possam ser roteadas de maneira eficiente até o par que possui o conteúdo desejado. A desvantagem dos sistemas estruturados é que é difícil manter a estrutura necessária para rotear eficientemente as requisições numa população de pares altamente variável;
- **hierárquica** – Neste tipo de rede os pares são organizados em diferentes grupos, que formam redes de sobreposição distintas. Porém há outro grupo, em um primeiro nível, que é formado por um representante de cada grupo. Este tipo de arquitetura é mais flexível que a arquitetura não-estruturada convencional com super-pares, já que cada

grupo pode implementar uma rede de sobreposição diferente (estruturada ou não-estruturada), que não precisa ser igual a rede do primeiro nível.

As redes P2P se diferenciam muito segundo o funcionamento, dependendo da rede de sobreposição escolhida, mas também das decisões de como realizar a descoberta de pares, do que compartilhar e como realizar a transmissão das informações. Ao contrário das CDNs, não precisamos de vários servidores especializados para realizar a distribuição dos dados, o que torna a solução mais robusta para o atendimento a localidades remotas e *offshore*, já que uma eventual falha de *software* ou *hardware* em um ou mais clientes não impactará a distribuição dos demais clientes na localidade, que ainda poderão obter as vantagens da rede P2P. Por estes motivos, a utilização de redes P2P pode ser uma escolha interessante quando em ambientes corporativos há dificuldade de se manter uma infraestrutura distribuída de servidores dedicados a distribuição de dados. A seguir, veremos o funcionamento do protocolo BitTorrent, que é a rede P2P mais utilizada na Internet atualmente e, devido a sua simplicidade de implantação, pode ser utilizada em redes corporativas, como pode ser visto em Van Der Sar (jun. 2010) e Van Der Sar (fev. 2010).

Dando continuidade ao artigo Costa e Rubinstein (2015), o objetivo desta dissertação é analisar o desempenho do protocolo BitTorrent padrão em redes corporativas e também realizar a análise após uma modificação no comportamento padrão do protocolo BitTorrent. A modificação proposta usa o conceito de ciência de localização para reduzir o tráfego nos enlaces de longa distância. Para isso, o rastreador identifica o endereço IP do par que está solicitando a lista de endereços IP do enxame e envia somente aqueles pertencentes à mesma rede local e ao semeador original. Nos cenários corporativos típicos propostos, as simulações mostraram que a alteração é capaz de reduzir o consumo médio de banda dos enlaces de longa distância e também reduzir o tempo médio dos *downloads*, quando comparados ao BitTorrent padrão, além de conferir maior robustez à distribuição em casos de falhas em enlaces de longa distância. As simulações mostraram também que em ambientes mais complexos, com muitos clientes distribuídos por várias redes, e onde a restrição de banda em enlaces de longa distância provoca congestionamento e descartes, o desempenho do protocolo BitTorrent padrão pode ser semelhante a uma distribuição em arquitetura cliente-servidor. Neste último caso, a modificação proposta mostrou resultados consistentes de melhoria do desempenho da distribuição. A redução do consumo médio dos enlaces é interessante, pois reduz a necessidade de ampliação destes, gerando economia para o proprietário da rede. Já a redução do tempo médio de *download* é especialmente interessante quando se trata de atualizações de

sistemas operacionais e antivírus, pois quanto menor de distribuição das atualizações, menor o tempo de exposição dos sistemas a ameaças digitais conhecidas.

A dissertação está organizada da seguinte forma. No Capítulo 1, o funcionamento do protocolo BitTorrent é apresentado. A seguir, diversas estratégias de melhoria do protocolo BitTorrent são detalhadas no Capítulo 2, com foco nas modificações que se utilizam do conceito de localização para tornar a distribuição de conteúdo mais eficiente. No Capítulo 3, são comentadas algumas particularidades da utilização do BitTorrent em redes corporativas, quando se compara com a sua utilização na Internet. Neste capítulo também é apresentada a proposta de modificação do protocolo BitTorrent e é explicado o funcionamento da seleção de pares por sub-redes IP. No Capítulo 4, são detalhados os cenários das simulações que foram realizadas para verificar o desempenho e a robustez do protocolo BitTorrent padrão e do modificado, assim como são apresentados e comentados os resultados das simulações. Depois disso, a dissertação é concluída e sugestões de trabalhos futuros são apresentadas.

1 BITTORRENT

Segundo Cohen (2003), o BitTorrent é “um sistema de distribuição de arquivos que utiliza o método olho-por-olho para buscar uma eficiência à Pareto”. Isto é, o BitTorrent tenta tornar as trocas P2P o mais eficiente possível, sem prejudicar nenhum par individualmente através do método olho-por-olho (a ser apresentado posteriormente). O BitTorrent é a mais popular rede P2P na Internet, correspondendo a 5,03% do tráfego total de Internet cabeada nos EUA, no segundo semestre de 2014, de acordo com Sandvine (2015), ficando apenas atrás dos tráfegos totais do Netflix (32,39%), YouTube (13,25%) e HTTP (*Hypertext Transfer Protocol*) (8,47%).

O protocolo BitTorrent tenta diminuir o impacto ou resolver problemas comuns no funcionamento das redes par-a-par. Por exemplo, os processos de encontrar quais pares tem as partes do arquivo e para onde elas devem ser enviadas podem resultar em um grande *overhead*. Além disso, há o problema de altas taxas de entradas e saídas de pares na rede par-a-par. Os pares raramente se conectam à rede por mais que algumas horas e, frequentemente, apenas por alguns minutos, o que pode causar problemas de justiça, isto é, que alguns pares realizem muito mais *download* do que *upload*. A estratégia para contornar este problema é fazer a taxa de *download* ser proporcional a de *upload* (COHEN, 2003).

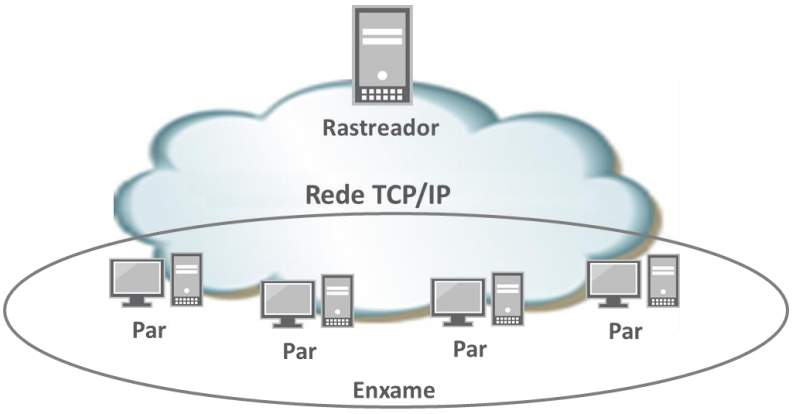
Em um comportamento típico de uma rede par-a-par, o número de pares interessados no *download* aumenta rapidamente após o arquivo ficar disponível. Já o número de pares que tem o arquivo completo para ofertar à rede par-a-par cresce vagarosamente.

1.1 Funcionamento do BitTorrent[1]

A seguir será detalhado, passo a passo, o funcionamento do protocolo BitTorrent. A arquitetura do protocolo está apresentada na Figura 1.

[1] Esta subseção foi baseada em Cohen (2003).

Figura 1 - Arquitetura do protocolo BitTorrent.



Fonte: O autor, 2015.

1.1.1 Publicação do Conteúdo

Para publicar um conteúdo, um arquivo de extensão ".torrent" é colocado em um servidor Web. Este arquivo contém informações sobre o arquivo que será distribuído, como o tamanho, o nome, as informações de *hash* e a URL do rastreador (*tracker*). Os rastreadores são responsáveis por auxiliar os pares que desejam participar da rede do BitTorrent a se encontrarem, em um primeiro momento, e, depois, por manter atualizada a lista dos pares que participam da rede, fornecendo, quando solicitados, uma nova lista para os pares. O conjunto de pares que participam do compartilhamento de um arquivo é chamado de enxame (*swarm*). O enxame se comunica com o rastreador utilizando um protocolo simples baseado em HTTP, em que par que deseja realizar o *download*, envia informação do arquivo desejado e qual porta ele está ouvindo, e o rastreador responde com uma lista de pares que estão realizando o *download* do mesmo arquivo, isto é participando do enxame. Para que o arquivo fique disponível na rede, algum par que possua o arquivo completo deve se conectar ao rastreador. O par que possui o arquivo completo é chamado de semeador (*seeder*), sendo que o primeiro par que possui o arquivo completo a entrar na rede é chamado de semeador original.

1.1.2 Localização de Pares

A responsabilidade do rastreador é estritamente limitada a auxiliar os pares a se encontrarem. Apesar disso, algumas informações sobre as taxas de *upload* e de *download* são enviadas ao rastreador. O algoritmo padrão do rastreador envia para os pares uma lista aleatória com 50 pares conectados, podendo ser menor se houver menos pares na rede. Porém o rastreador pode ser implementado com um algoritmo mais inteligente para gerar a lista de pares, por exemplo, evitando divulgar endereços de semeadores para outros semeadores.

O BitTorrent divide os arquivos em partes de tamanho fixo, tipicamente, 256 kbytes. Cada par informa a todos outros pares conhecidos quais partes ele tem. Para verificar a integridade dos dados, os *hashs* (SHA1 - *Secure Hash Algorithm* 1) de todas as partes do arquivo que será distribuído estão incluídos no arquivo ".torrent", e os pares não informam que têm uma parte até que verifiquem se ela possui o mesmo *hash* contido no arquivo ".torrent".

Quando dois pares se conectam, eles informam entre si as partes que possuem. E assim que o *download* de uma parte é concluído, o par avisa aos outros pares conectados que possui uma nova parte.

Enquanto é possível, os pares continuamente fazem *download* das partes dos pares que eles conhecem. Ocasionalmente, os outros pares podem não possuir as partes desejadas.

1.1.3 Transferência dos Dados

O BitTorrent transfere os dados utilizando conexões TCP. Para se utilizar todo o meio de transmissão e aumentar a velocidade da transferência dos arquivos, é útil que sejam utilizadas conexões TCP em paralelo. O BitTorrent facilita este processo dividindo as partes em blocos, tipicamente de 16 kbytes, e sempre deixando algumas transferências, tipicamente cinco, sendo realizadas ao mesmo tempo.

Toda vez que o *download* de um bloco é completado, uma nova requisição de outro bloco da mesma parte é realizada, a menos que o *download* da parte tenha sido concluído. Neste caso, se inicia o *download* de outra parte.

1.1.4 Seleção de Partes

A seleção das partes numa ordem adequada é muito importante para um bom desempenho da rede BitTorrent. A seguir veremos algumas das estratégias que o protocolo usa para atingir um desempenho mais próximo do ideal na distribuição dos arquivos.

1.1.4.1 Prioridade Estrita

Quando um bloco é requisitado, todos os blocos daquela parte tem que ser requisitados antes de um bloco de outra parte. Desta maneira, as partes são completadas da maneira mais rápida possível.

1.1.4.2 Partes mais Raras Primeiro

Quando os pares selecionam a próxima parte que deve descarregada, eles geralmente selecionam primeiro aquelas partes que tem menos cópias nos pares, numa estratégia conhecida como "*rarest first*" (mais raras primeiro). Com esta técnica se garante que os pares tenham partes que todos queiram e que as partes mais comuns sejam deixadas para depois, evitando que haja um momento em que as partes disponíveis sejam aquelas onde não há mais interesse. Desta maneira, nenhum par completará o *download* do arquivo sem que todas as partes tenham sido distribuídas pelo semeador original. Em algumas situações, o semeador original pode ser tirado do ar (por falha, economia, etc.), e mesmo nestas situações esta estratégia se mostra eficiente, pois replica o mais rápido possível as partes, reduzindo o risco de que o *download* seja inviabilizado.

1.1.4.3 Partes Aleatórias Primeiro

Quando o *download* se inicia, não é utilizada a estratégia *rarest first*. Neste momento, é importante obter uma parte o mais rápido possível para que o novo par do enxame possa contribuir com a transferência para os demais pares. As partes mais raras, geralmente, estão presentes em apenas um par, o que provoca um *download* mais lento do que das outras partes, pois dessas outras partes é possível obter blocos em outros pares. Por esta razão, a primeira parte é selecionada aleatoriamente e, quando é completada, a estratégia muda para a *rarest first*.

1.1.4.4 Fim de Jogo

Algumas vezes uma parte vai ser requisitada a um par com taxas de transferência muito baixas. Isso não é um problema no meio de um *download*, mas pode causar uma demora muito grande no fim da transferência. Para evitar que isso aconteça, uma vez que todos os blocos que um par ainda não possui já estão sendo requisitados, ele realiza requisições destes blocos para todos pares conhecidos que possuem a parte desejada. Ao fim do *download* da parte, ele cancela as transferências ainda pendentes.

1.1.5 Algoritmos de “Estrangulamento”

No BitTorrent, cada par é responsável por tentar maximizar a própria taxa de *download*. Os pares fazem isso através do *download* de qualquer par disponível e decidindo para quem irão enviar através de uma variação da política “olho-por-olho” (*tit-for-tat*). Se os pares decidirem cooperar, eles enviam os dados para os solicitantes. Se decidirem não cooperar, utilizam um mecanismo chamado “estrangulamento” (*choking*), que consiste em se recusar a realizar o *upload*. O *download* pode continuar sendo realizado e a conexão não precisa ser renegociada ao fim do estrangulamento. O algoritmo de estrangulamento não é tecnicamente parte do protocolo BitTorrent, mas necessário para um bom desempenho. Um

bom algoritmo de estrangulamento deve utilizar todos os recursos disponíveis, prover taxas de *download* razoavelmente consistentes para todos e ser de alguma forma resistente a pares que só realizam *download* e não fazem *upload*. Os pares que buscam apenas consumir recursos de uma rede P2P, apenas realizando *download* sem contribuir com *upload* são chamados de *free-riders*.

1.1.5.1 O Algoritmo de Estrangulamento do BitTorrent

Cada par de BitTorrent sempre libera o *upload* (“desestrangula” – *unchokes*) para um número fixo de pares (por padrão, quatro), logo a questão se resume em para quais pares ele vai liberar. Esta abordagem permite que o controle de congestionamento do TCP sature a capacidade do *upload*. A decisão de quais pares liberar é baseada estritamente na taxa de *download* atual. Para calcular a taxa de *download* pode se utilizar na implementação uma média móvel de 20 s. Para evitar que recursos sejam desperdiçados pelo estrangulamento e liberação dos pares, os pares recalculam quem eles devem estrangular uma vez a cada 10 s, e então mantém a decisão até o fim do ciclo. Este período de 10 s é o suficiente para o TCP aumentar a taxa de transmissão até capacidade máxima.

1.1.5.2 Liberação Otimista

Simplesmente fazer *upload* para os pares que fornecem as melhores taxas de *download* pode não ser sempre a melhor estratégia, pois desta maneira podem não ser encontradas conexões com taxas melhores que as atuais. Para solucionar isso, o par BitTorrent deve ter uma “liberação otimista” (“*optimistic unchoking*”) , que é liberar para o *upload* um par independente da taxa de *download* atual. Esta estratégia resolve também o problema que ocorre quando um par novo entra no enxame e ainda não tem nenhuma parte, pois nesta situação a taxa de *upload* do par é nula e o algoritmo de estrangulamento dos outros pares vai mantê-lo bloqueado, mesmo que ele possa transmitir os dados com altas taxas de *upload* assim que conseguir alguma parte do arquivo. Assim, a liberação otimista ajuda a “ativar” novos pares no enxame, entregando dados para eles compartilharem, melhorando o

desempenho do sistema. Em cada par, a liberação otimista é alterada a cada três períodos de estrangulamento (30 s). Este tempo é o suficiente para que o *upload* alcance a sua capacidade total, o *download* tenha reciprocidade e atinja também a capacidade máxima.

1.1.5.3 "Esnobando"

Ocasionalmente, um par BitTorrent será estrangulado por todos os pares que estão enviando dados para ele. Nestes casos, ele continuaria a ter taxas baixas de *download* até que a liberação otimista ache pares melhores. Para mitigar este problema, depois de um minuto sem conseguir nenhuma parte de um par em particular, o BitTorrent assume que ele está sendo "esnobado" por aquele par e não realiza mais *upload* para ele exceto no caso de uma liberação otimista.

1.1.5.4 Apenas *Upload*

Quando um par conclui o *download*, ele não tem mais taxas de *download* para decidir para quais pares ele fará o *upload*. Na implementação padrão, o cliente BitTorrent passa a preferir pares para os quais ele possa enviar os dados nas taxas mais altas. Isto melhora a eficiência na utilização de toda a capacidade de *upload* e dá preferência para os pares que não estejam recebendo dados no momento.

2 PROPOSTAS DE MODIFICAÇÕES NO PROTOCOLO BITTORRENT

Nos últimos anos, foram propostas várias alterações no protocolo BitTorrent com a finalidade de aumentar a sua eficiência, tanto para reduzir a utilização da capacidade da rede, quanto para acelerar a velocidade de *download* dos arquivos. A seguir, veremos algumas destas propostas, mas antes faremos um introdução a uma questão importante no estudo do desempenho do protocolo BitTorrent: o conceito de localização.

2.1 O problema da localização no BitTorrent

Quanto se lida com a distribuição de conteúdo BitTorrent na Internet, um dos principais problemas que ocorrem é a grande quantidade de tráfego nos enlaces inter-ISPs (*Internet Service Provider*s) ou entre os ASs (*Autonomous Systems* – Sistemas Autônomos). Mesmo que dentro de um ISP vários pares estejam realizando o *download* do mesmo conteúdo, isto é, estejam no mesmo enxame, eles não estarão necessariamente conectados entre si. Como consequência, os pares vão realizar desnecessariamente o *download* do conteúdo de pares localizados fora do seu ISP. Os enlaces inter-ISPs geram altos custos para os ISPs, então é interessante que o tráfego nestes enlaces seja baixo para diminuir a necessidade de ampliação de banda. Segundo resultado de estudo apresentado em Oechsner et al. (2009), enxames típicos na Internet tem apenas de 1% a 10% de pares no mesmo AS, o que agrava o problema. O problema se repete em ambientes corporativos, quando os pares podem enviar e receber dados através de enlaces de longa distância, ao invés de dar preferência para pares que estão em uma mesma rede local.

Uma solução para este problema é usar o conceito de localização P2P (BINDAL et al, 2006). É interessante conter o tráfego P2P dentro do próprio ISP, assim minimizando o tráfego inter-ISP. O objetivo de uma política de localização é limitar o número de conexões inter-ISPs. Quanto maior a localidade, menor o número de conexões inter-ISPs. Por padrão, o BitTorrent não aplica nenhuma política de localização e usa uma política aleatória na escolha dos pares que serão informados pelo rastreador (THEORY.ORG, 2015). Podemos estender este conceito para ASs, limitando tráfego inter-ASs, ou para redes locais, limitando o tráfego

pelos enlaces de longa distância. A política de escolha de pares pode ser implementada nos rastreadores ou nos próprios pares. Os sistemas de BitTorrent que se utilizam de alguma informação de localização para tomada de decisão durante o funcionamento (na escolha de pares, por exemplo) são chamados de sistemas de BitTorrent cientes de localização (OECHSNER et al, 2009; LI; XIE, 2010).

Outra maneira de forçar a localização do tráfego é apresentada em Bindal et al. (2006), e consiste em utilizar dispositivos limitadores de tráfego P2P. Estes dispositivos podem identificar o tráfego P2P e manipulá-los. Para cada conteúdo, estes dispositivos rastreiam os pares do ISP que estão fazendo o *download*. Quando os pares se juntam a uma rede para buscar algum conteúdo, o dispositivo intercepta e modifica a resposta do rastreador para o par, substituindo os pares externos ao ISP por pares internos.

Outro fato interessante explorado nos experimentos realizados por Bindal et al. (2006) é que a redução da banda entre os enlaces inter-ISPs acaba forçando que os pares privilegiem as conexões intra-ISP, diminuindo o tráfego entre os ISPs. Isto pode ser explicado pelo funcionamento do mecanismo “olho-por-olho”, que acaba privilegiando a troca de dados entre os pares onde são possíveis as maiores taxas de transmissão de dados. Como neste cenário há gargalos entre os ISPs, o protocolo BitTorrent, utilizando o mecanismo “olho-por-olho”, tem maior probabilidade de manter as conexões intra-ISP do que as inter-ISP, a menos que seja necessário buscar em outro ISP alguma parte que não exista dentro do mesmo ISP.

Em Le Blond, Legout e Darbbous (2011), é estudado o quanto se pode forçar a localidade sem prejudicar o funcionamento do protocolo BitTorrent. No experimento realizado, para controlar o número de conexões inter-ISPs é assumido que o rastreador pode mapear cada par ao seu ISP de origem. O único parâmetro da política de localidade implementada é o número máximo de conexões de saída inter-ISPs, ou seja, de pares conectados pertencentes a diferentes ISPs. O rastreador mantém, para cada ISP, o número de pares fora do ISP que ele retornou como resposta, junto com a identidade dos pares. Quando o rastreador implementa esta política de localidade, ele o faz para todos os pares exceto o semeador original. O tráfego gerado pelo semeador original é desprezível quando comparado ao tráfego agregado do enxame. O artigo citado conclui que podemos forçar a localidade até termos poucas conexões inter-ISPs; no caso, quatro conexões, sem termos problemas de desempenho em relação ao tempo de realização do *download*, mas com uma relevante redução no tráfego inter-ISP.

2.1.1 Liberação Orientada x Seleção Orientada de Pares

Na Seção 2.1, citamos que a política de escolha de pares pode ser implementada nos rastreadores ou nos próprios pares. Em Oechsner et al. (2009), é realizada uma comparação entre as duas estratégias.

Foram propostas duas estratégias: na Liberação Orientada (BU – *Biased Unchoking*) a escolha dos pares para os quais serão realizados os uploads será feita utilizando as informações de localização dos pares vizinhos, na Seleção Orientada de Pares (BNS – *Biased Neighbor Selection*) o rastreador fornece a lista de pares baseada nas informações de localização dos pares.

Ambos algoritmos BNS e BU necessitam de alguma informação sobre a topologia da rede. Se utiliza um valor de localização $L(x,y)$ para todo par de endereços de pares x e y. Para o cálculo de $L(x,y)$ diferentes estratégias podem ser utilizadas. Por exemplo, o número de saltos (de roteamento) ou de ASs entre os pares pode ser usado. Este valor pode ser dado por um serviço ou calculado pelos próprios pares ou pelo rastreador.

No BNS, quando um par de endereço x requer os endereços y dos outros pares do enxame, o rastreador usa a informação contida em $L(x,y)$ para criar a resposta. Como exemplo, se a métrica for de saltos de AS, isto é, quantos ASs serão atravessados para se chegar ao outro par, o rastreador tentará colocar uma fração de pares com $L(x,y) = 0$ na resposta, isto é, pares no mesmo AS. Se não houver pares suficientes para a resposta, o rastreador completa com outros pares que não se enquadram no critério, isto é $L(x,y) > 0$, para evitar a degradação da conectividade do enxame.

Com o BU, o par preferencialmente troca dados com os vizinhos que tem um “bom” valor de $L(x,y)$ e isto é feito através do mecanismo de estrangulamento. No BitTorrent, todos os n pares interessados e estrangulados de um par são liberados de maneira otimista com probabilidade igual a $1/n$. No BU, o valor de $L(x,y)$ é levado em consideração. Dois grupos são gerados com base em um valor de T, fixo, onde um grupo terá $L(x,y) \leq T$, com probabilidade q de algum elemento ser escolhido para ser liberado, e outro grupo $L(x,y) > T$, com probabilidade $1 - q$ de algum elemento ser escolhido para ser liberado. Se algum dos grupos estiver vazio, é escolhido algum par do outro grupo. Como no exemplo de BNS, temos como exemplo de métrica o número de saltos de AS e $T = 0$, isto é, apenas pares dentro do mesmo AS terão preferência, com $q = 1$.

Resultados de simulações realizadas em Oechsner et al. (2009) mostram que a eficiência dos algoritmos depende de vários fatores, mas o BNS e o BU se mostram mais eficientes que o BitTorrent convencional e a combinação dos dois algoritmos (BNS&BU) se mostra com uma eficiência ainda maior, reduzindo ainda mais o tráfego inter-ASs (ou inter-ISPs).

2.1.2 Problema das Partes Raras

Quando utilizamos o algoritmo BU, é dado o privilégio para a liberação otimista de *upload* para os pares dentro do mesmo ISP, porém não é levada em consideração a raridade da parte que vai ser enviada para o ISP de destino. Como no BitTorrent é normal que os pares se desconectem da rede, é interessante que um par que tem partes raras faça o *upload* destas partes o quanto antes, isto é, estes envios tem que ser prioritários para que estas partes se mantenham no enxame mesmo com a desconexão do par. Este cenário se torna mais crítico quando se usa alguma política de localidade como o BU, pois a parte rara pode estar fora do ISP e o *upload* por liberação otimista dará prioridade para as conexões intra-ISP, deixando de enviar a parte rara.

Em Li e Xie (2010) este problema é detalhado e é proposto um mecanismo para solucionar esta questão, o PicBou. Este mecanismo consiste em calcular uma Nota de Preferência (PS – *Preference Score*) a partir do produto de duas outras notas, a Preferência pela Localização da Rede (PNS – *Network Locality Preference*) e a Preferência pela Distribuição das Partes (PPS – *Piece Distribution Preference*). A liberação otimista será dada para o par com a maior nota de PS. Em resumo, a nota PNS é maior quando o par está no mesmo ISP e a nota PPS é maior para os pares que tem partes que não estão presentes no ISP. A estratégia da nota PPS é incentivar que a liberação de *upload* convencional (não-otimista) seja ativada pelo par remoto, propiciando o envio das partes raras ou inexistentes no ISP. Foram realizadas simulações que mostraram que o PicBou consegue ser mais eficiente que o BU e o BNS, reduzindo o tempo médio de *download* e também o tráfego inter-ISP.

2.1.3 A Dependência de infraestrutura

Em geral, os mecanismos propostos para melhorar o problema de tráfego inter-ISP que lidam com a localização dos pares precisam de informações fornecidas por uma infraestrutura externa ou fora da solução padrão do BitTorrent para funcionar; como por exemplo, rastreadores preparados para avaliar a localização dos pares que se conectam ao enxame e fornecer para os clientes BitTorrent listas de pares orientadas a reduzir o tráfego inter-ISP. Porém, a necessidade destas modificações, ou de uma infraestrutura especial, acaba gerando alguns problemas, como a dependência de que alguma entidade ou ISP forneça e faça a manutenção da infraestrutura e a falta de interoperabilidade entre os sistemas de BitTorrent.

Em Ren et al. (2010) é apresentado um sistema BitTorrent ciente de localização, chamado TopBT, que não depende de infraestrutura especial para operar e que pode funcionar em conjunto com os sistemas BitTorrent já existentes. Para obter as informações de localização, o cliente TopBT se utiliza das ferramentas Ping e Traceroute para verificar por quantos e quais roteadores ele tem que passar para atingir os outros pares. O sistema funciona da seguinte forma:

- o cliente TopBT testa os seus pares conectados usando o Ping e o Traceroute. Como o Ping e o Traceroute utilizam diferentes tipos de pacotes e os roteadores no caminho podem filtrar esses pacotes, se utilizam simultaneamente as duas ferramentas para aumentar as chances de sucesso no teste;
- obtém-se o valor de TTL (*Time to Live* – Tempo de Vida) dos pacotes de Ping e Traceroute quando os pacotes de teste retornam. Com este valor é possível se obter o número de saltos de roteamento até o par testado, pois como o valor do TTL na geração do pacote é fixo, subtraindo-se o valor do TTL do pacote recebido do valor do TTL do pacote original, se calcula o número de roteadores que o pacote atravessou até chegar ao seu destino, isto é, o número de saltos de roteamento;
- o próximo passo é calcular o número de saltos de AS. Para isso, primeiro se constrói uma tabela de prefixos de ASs através do *download* de uma tabela de roteamento BGP (*Border Gateway Protocol*). Podem ser usados repositórios públicos para isso. Com o resultado do Traceroute e com a tabela de prefixos, se identifica o AS de cada endereço e, então, por quantos ASs o pacote passa até atingir o par vizinho. A tabela deve ser renovada após algumas semanas, mas o processamento não precisa ser simultâneo ao funcionamento do cliente TopBT.

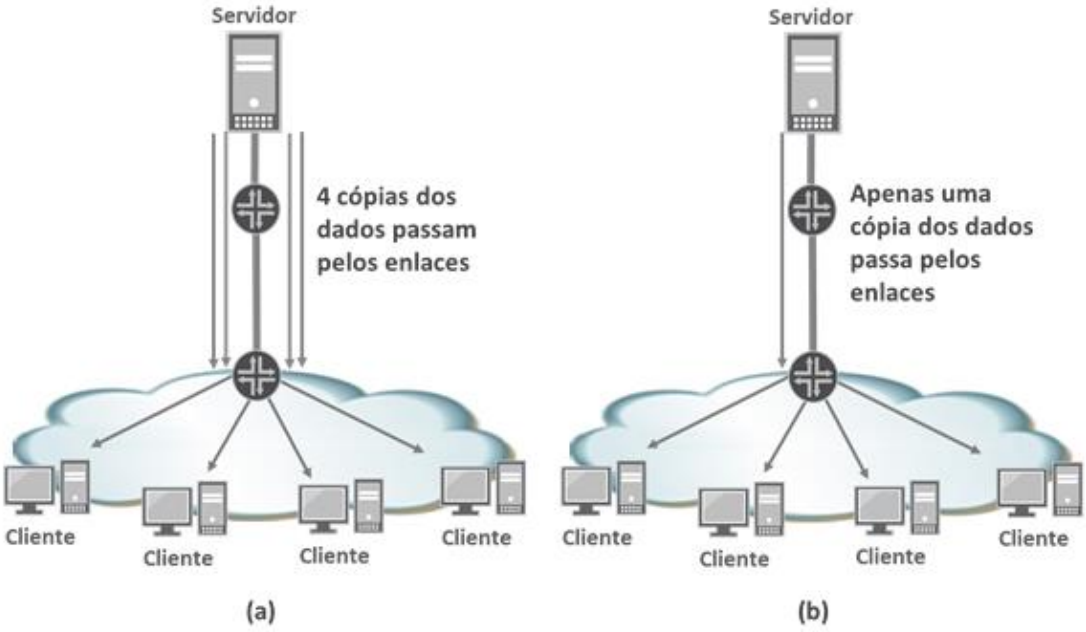
O cliente Top BT monitora as conexões e mede as taxas de *download* (*d*) e *upload* (*l*) entre os pares conectados e também as utiliza para escolher quais pares terão o *upload* liberado. Os valores de saltos de AS (*a*), saltos de roteamento (*l*) e as taxas de *download* (*d*) e *upload* (*l*) são utilizados para calcular a métrica *d/(l x u)* ou *d/(a x u)* para cada par a cada rodada de liberação. Então o cliente TopBT libera o *upload* para os pares com os maiores valores de métrica.

Assim como nos mecanismos anteriores, o TopBT apresenta tráfegos menores inter-ISPs e ainda diminui o tempo de *download*, quando comparado com o BitTorrent convencional.

2.2 BitTorrent e *Multicast*

A utilização do protocolo *multicast* IP nativo pode ser apontada como uma solução para reduzir a redundância de dados nos enlaces em uma distribuição de dados, quando esta distribuição tem origem em único ponto (servidor) para vários destinos (clientes) simultaneamente. Os roteadores com *multicast* IP têm a capacidade de copiar os pacotes que passam por um enlace para outros enlaces onde são identificados clientes que desejam também receber os dados do grupo *multicast* específico. As redes de destino podem ter vários clientes interessados, mas em cada enlace entre o cliente e o servidor passará apenas uma cópia dos dados. A Figura 2 mostra exemplos de transferências com cliente/servidor e *multicast* IP.

Figura 2 - Cliente/Servidor e *multicast*.



Legenda: (a) Em uma transferência convencional na arquitetura Cliente/Servidor, pelos enlaces passam uma cópia do dado para cada cliente. (b) Utilizando o protocolo multicast IP, temos a capacidade de enviar apenas uma cópia e os dados serão replicados, quando necessário, para entregar uma cópia para cada cliente.

Fonte: O autor, 2015.

Apesar de em um primeiro momento o *multicast* se mostrar como uma solução que leva a uma utilização ótima da rede, pois acaba com a redundância de dados, ela não é de fácil implementação na Internet. Na Internet, entre os roteadores e os ASs que possuem administradores diferentes geralmente não temos *multicast* habilitado. E são vários motivos para que isto não aconteça, conforme listado em Pushp e Ranjan (2010):

- muitos roteadores na Internet, por problemas técnicos ou opção dos administradores, não possuem suporte ao protocolo *multicast*;
- o controle de congestionamento não está bem definido para o *multicast*;
- as políticas de cobrança pelo tráfego entre ISPs não estão bem definidas e não há incentivos para colocar *multicast* nas redes.

Por estes motivos, entre outros, quando se busca características de *multicast* na Internet podem ser utilizadas soluções P2P. As soluções *multicast* P2P pertencem ao que se chama de Protocolos *Multicast* no Nível de Aplicação (ALM – *Application-Level Multicast*). Com o ALM se utilizam apenas os clientes finais para criar uma funcionalidade semelhante ao *multicast* IP nativo, com a vantagem de não serem necessárias configurações adicionais no núcleo da rede, apenas o roteamento IP *unicast* é requerido. As soluções ALM utilizam mais banda do que o *multicast* IP nativo, mas têm a vantagem de poderem ser implementadas

rapidamente. De acordo com Passarella (2012), há duas principais estruturas de *multicast* utilizadas no ALM: árvores e malhas. Em uma aplicação que utiliza a estrutura em malha, os pares participantes se comunicam entre si e usam um protocolo de roteamento para entregar o conteúdo aos outros membros do *multicast*. Porém esta estrutura de malha tende a duplicar o conteúdo e necessita de uma lógica adicional de roteamento além da lógica para construir a malha. Desta maneira, as soluções de *multicast* P2P usam preferencialmente estrutura em árvore, utilizando de redes de sobreposição estruturadas. Sistemas ALM podem ter a sua estrutura em árvore montadas resumidamente da seguinte forma: um grupo *multicast* é definido por um ID. Então um nó que será responsável por este ID será chamado a raiz do grupo. Para entrarem no grupo, os nós enviam uma mensagem de *join* para o ID do grupo. A mensagem de *join* é propagada em direção à raiz até que um ponto de ramificação, isto é, um nó que já participe do grupo seja encontrado ou até que a mensagem atinja a raiz. A cada salto na rede de sobreposição, o nó que recebe a mensagem registra o nó que enviou a mensagem como um dos seus ramos e encaminha a mensagem em direção à raiz. Neste momento, o nó que recebeu a mensagem também se registra no grupo. Desta forma, é construída uma árvore que é formada pelos ramos, que vão desde os nós que não possuem ramificações até a raiz. Montada a árvore, as mensagens de *multicast* são encaminhadas ao nó raiz e então distribuídas para os outros nós através da estrutura formada.

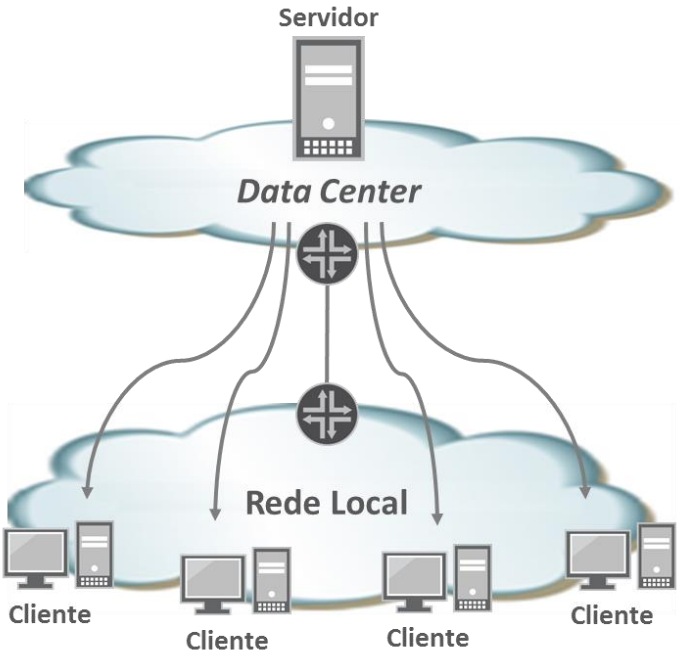
Apesar do *multicast* IP ser de difícil implementação na Internet, em redes corporativas o cenário é diferente. A existência de apenas um administrador (ou apenas poucos administradores) facilita que a configuração dos roteadores seja uniforme e com o *multicast* habilitado. Em redes corporativas, o congestionamento dos enlaces pode ser melhor controlado através de utilização de Qualidade de Serviço (QoS - *Quality of Service*) ou com ampliação de banda sempre que necessário.

Unindo as características das redes P2P e do *multicast* IP foi proposto em Pushp e Ranjan (2010) um protocolo de distribuição de conteúdo chamado HTorrent. Neste protocolo, o envio dos dados é realizado primeiro utilizando *multicast* IP nativo, mas quando há perda de pacotes, eles são retransmitidos utilizando a rede P2P. Este sistema produz ganhos nos tempos de *download* do conteúdo e reduz a utilização da rede. Porém, quando há muita perda de pacotes na transmissão via *multicast*, o HTorrent se mostra menos eficiente que o BitTorrent convencional, gastando mais tempo para o *download* do conteúdo.

3 BITTORRENT EM REDES CORPORATIVAS

O BitTorrent pode ser utilizado com sucesso para melhorar distribuições de dados em redes corporativas, podendo distribuir arquivos de poucos megabytes até alguns gigabytes. A eficiência da utilização das distribuições através de redes P2P como o BitTorrent é apresentada em detalhes em Mundinger, Weber e Weiss (2008). Nesse trabalho, o tempo de transferência teórico de uma solução em arquitetura Cliente/Servidor é comparado com o da arquitetura P2P utilizada pelo BitTorrent. A análise mostra que quanto maior o número de nós e o número de partes de um arquivo, maior será a eficiência do protocolo BitTorrent quando comparada a de uma arquitetura Cliente/Servidor.

Figura 3 - *Downloads* concorrentes.



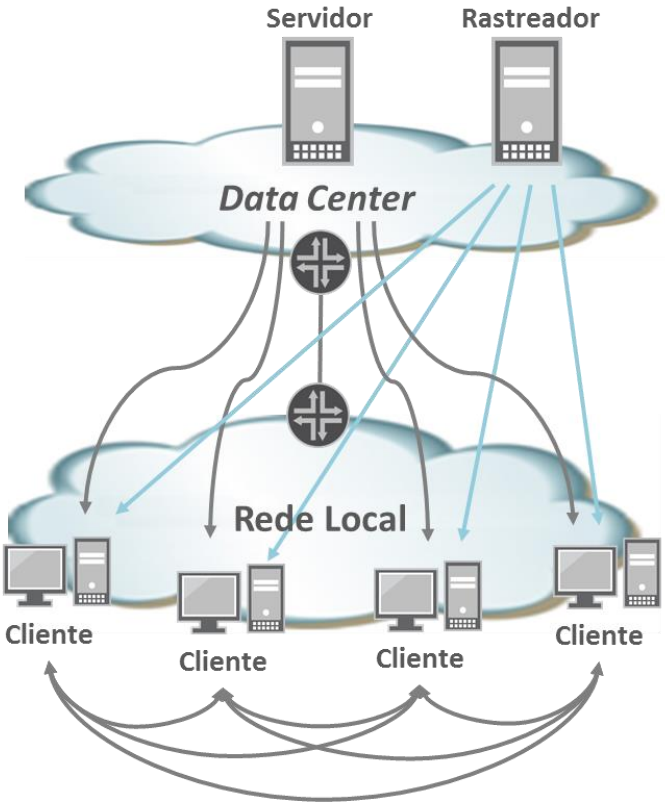
Legenda: *Downloads* concorrentes geram uma carga maior nos servidores e nos enlaces da rede.

Fonte: O autor, 2015.

Uma desvantagem da distribuição de arquivos na arquitetura Cliente/Servidor é citada em Pushp e Ranjan (2010). Por exemplo, na Figura 3, para cada cliente que desejamos enviar o pacote de dados, temos no mínimo uma conexão TCP para o servidor. Cada conexão vai enviar dados pelo enlace que interliga o *data center*, onde o servidor está hospedado, à rede local, onde estão os clientes. Porém, os dados enviados para cada cliente são idênticos, isto é, envia-se uma cópia de cada informação para cada um dos clientes. Esta situação gera desperdício de banda nos enlaces e no servidor.

Na Figura 4, é apresentada uma estratégia mais eficiente utilizando o protocolo BitTorrent. Nela, o servidor que fará o *upload* da distribuição de dados age como um semeador e, com o auxílio de um servidor fazendo o papel de rastreador, envia os dados para os clientes. Porém, assim que os clientes completam o *download* da primeira parte solicitada ao servidor, passam a também ser fornecedores de dados da distribuição para os demais clientes, seguindo as estratégias definidas pelo protocolo BitTorrent. Isto acelera o processo de envio de dados e diminui a utilização do enlace entre a rede local e o *data center*.

Figura 4 - Distribuição com BitTorrent.



Legenda: Com o protocolo BitTorrent, os clientes ajudam a acelerar a distribuição e reduzem o tráfego no enlace entre o *data center* e a rede local.

Fonte: O autor, 2015.

Apesar de o BitTorrent ter sido pensado para um ambiente de Internet, onde não há garantias de disponibilidade e banda, ele pode ser adaptado para ambientes corporativos, onde há mais banda disponível, principalmente em redes locais, baixa latência e os clientes tendem a ser mais confiáveis. Neste caso, até estratégia "olho-por-olho" pode ser repensada, pois pode haver "garantia" que os clientes ficarão na rede por tempo suficiente para realizar o compartilhamento dos dados, pois os *softwares* dos clientes podem ser executados em

segundo plano com controle apenas do administrador da rede, dificultando que os pares se comportem como *free-riders*.

Em Somani et al. (2012), são discutidas algumas técnicas para otimizar as transferências de arquivos em ambientes corporativos, bem como uma avaliação de desempenho do BitTorrent em um cenário deste tipo. Dentre as técnicas possíveis, o artigo destaca alterações no cliente BitTorrent, como revisão da estratégia olho-por-olho e alterações no tamanho das partes dos arquivos, mudança nas estratégias de escolha de pares e até simplificação do protocolo BitTorrent, diminuindo o número de etapas necessárias para a distribuição dos arquivos. Simulações em ambientes similares a *data centers*, com banda de 1 Gbps, latência de poucos milissegundos e nas quais os pares ficam conectados a uma rede BitTorrent durante todo o tempo, mostraram que o volume de dados que cada par contribui para o enxame, isto é, o volume de *upload* que é realizado por cada par, incluindo o semeador original, varia pouco. Cada par do enxame, incluindo o semeador original, contribui com um volume de tráfego que fica entre 75% e 125% do tamanho total do arquivo distribuído. Isto é, o semeador original, apesar de ficar mais tempo na rede com o arquivo completo, não possui uma contribuição muito maior do que as dos outros pares. Este resultado mostra também que, em média, não há par sobrecarregado, o que é interessante em uma rede corporativa, onde os pares que são computadores de uso pessoal podem participar de um enxame sem correr o risco de uma redução significativa no desempenho do processamento, nem de uma alta utilização da sua própria placa de rede.

3.1 Proposta de Seleção de Pares por Sub-Redes IP

Podemos ver em Choffnes e Bustamante (2008) e Xie et al. (2008) propostas de como melhorar o compartilhamento de arquivos por BitTorrent reduzindo o tráfego inter-ISP. Porém estas soluções são complexas e dependem de infraestrutura que vai além daquelas propostas pelo protocolo BitTorrent original.

Na arquitetura P4P, proposta por Xie et al. (2008), são utilizados servidores adicionais nos ISPs, chamados iTrackers, para consolidar informações sobre a rede, como, por exemplo, a ocupação de enlaces. Estas informações são consultadas por outro servidor, chamado appTracker, a quem os clientes também se conectam. Os appTrackers consolidam informações dos iTrackers e, avaliando as necessidades dos clientes, podem, por exemplo,

utilizar estas informações para uma seleção orientada de pares e, então, passar uma lista otimizada de pares ao cliente BitTorrent. Nesta arquitetura, o appTracker substitui o papel do rastreador do BitTorrent padrão.

Já a proposta do *plugin* Ono para clientes BitTorrent, apresentado em Choffnes e Bustamante (2008), se baseia no fato de que podemos nos utilizar de CDNs já existentes, como a da Akamai, para encontrar pares próximos e melhorar o desempenho geral, diminuindo o tráfego P2P. Isto é possível porque quando se fazem consultas por conteúdos em CDNs geralmente as respostas são redirecionamentos para servidores com menor latência possível. Então, se dois ou mais pares recebem respostas semelhantes, isto indica que eles estão próximos entre si e devem, então, privilegiar trocas entre eles.

Na nossa proposta, foi realizada apenas uma modificação simples no rastreador, de modo que quando um cliente BitTorrent solicita uma lista de pares do rastreador, ele verifica quais são os pares que estão na mesma sub-rede e envia uma lista aleatória contendo apenas estes pares da mesma sub-rede do cliente solicitante, com exceção do semeador original, que sempre é enviado para as redes às quais ele não pertence. Esta estratégia de sempre enviar o semeador original para os clientes é semelhante à adotada por Le Blond, Legout e Darbbous (2011), quando ele segrega o tráfego por ISPs. Esta abordagem se aproveita do fato de que, em uma rede corporativa bem estruturada, as faixas de endereço estão relacionadas a localizações geográficas. Sem perda de generalidade, utilizamos para a definição da sub-rede a máscara 255.255.255.0 (/24), logo cada rede local tem uma sub-rede com esta máscara. Porém, há a liberdade de definir a máscara de sub-rede de acordo com o plano de endereçamento da rede corporativa em questão. Assim, sub-redes com máscara 255.255.254.0 (/23) ou 255.255.255.128 (/25), entre outras, podem também se apresentar adequadas. A opção de implementar a modificação com a máscara 255.255.255.0 (/24) foi por se mostrar adequada aos cenários que foram simulados. Com esta alteração proposta, estamos forçando que o tráfego da distribuição se mantenha dentro das redes locais, com exceção do tráfego de controle dos protocolos e do semeador original. Fora a já citada simplicidade, esta abordagem oferece outra vantagem: como o endereço IP é inerente ao cliente, não é necessária configuração adicional no cliente para que protocolo se aproveite da ciência de localização.

Nos capítulos a seguir serão apresentados cenários de redes corporativas que servirão de base para simulações. Estas simulações vão avaliar o desempenho do protocolo BitTorrent com e sem a modificação proposta e, então, os resultados das simulações serão comparados. Para implementação desta proposta podem ser aplicados os seguintes passos no código do rastreador BitTorrent:

1) copia-se integralmente lista de pares original para uma lista temporária;
2) excluem-se da lista temporária as entradas, isto é, pares, que não estão na sub-rede do par solicitante, a menos que seja o semeador original, que é um endereço conhecido pelo rastreador;
3) com a lista temporária limpa, se faz a seleção aleatória de pares;
4) depois da seleção feita e enviada para o par solicitante, a lista temporária é apagada. O processo é repetido a cada nova solicitação de pares que chega ao rastreador.

Outro ponto interessante desta alteração é a importância que o semeador original tem para as redes. A modificação proposta segrega as redes de maneira que, se ocorrer alguma falha e o semeador original sair do enxame sem que haja todas as partes da distribuição em cada uma das redes, a distribuição não será concluída. Em um ambiente corporativo, esse problema pode ser reduzido, pois o semeador original pode estar hospedado em um sistema com alta disponibilidade, com proteção contra falhas de *software*, *hardware* e energia. Ainda assim, uma eventual falha de rede, que ocasione um isolamento de alguma rede do semeador original, pode parar a distribuição nesta rede, se todas as partes ainda não estiverem presentes. No funcionamento padrão do BitTorrent, assim que todas as partes são transmitidas para o enxame, o semeador original não é mais fundamental para que se conclua a distribuição; porém antes disso, a distribuição pode parar caso o semeador original fique isolado.

4 SIMULAÇÕES

Por causa da sua popularidade, a sua popularidade, o BitTorrent foi implementado como aplicação no simulador NS-3 (NS-3, 2014), conforme detalhado em Weingärtner et al. (2012). Devido à existência de vários sistemas de BitTorrent, o modelo proposto não apenas modela um cliente de BitTorrent específico, mas permite facilmente a replicação de um comportamento de um sistema de BitTorrent diferente.

Na verdade, a implementação consiste de dois modelos: um modelo de cliente e um modelo de rastreador de BitTorrent. Há também um componente de controle de enxame (*Swarm Control*), que permite especificar os ajustes iniciais e a atividade dos clientes de BitTorrent durante a simulação, por exemplo, orientando um grupo de pares a iniciar as requisições por um arquivo num certo momento.

Segundo Weingärtner et al. (2012), um dos maiores objetivos do desenho deste modelo de simulação de BitTorrent é permitir simulações em larga escala com centenas ou milhares de clientes. O espaço requerido pelos estados dos pares foi reduzido armazenando as informações redundantes apenas uma vez, como, por exemplo, o arquivo compartilhado. Isto possibilitou reduzir os requisitos de memória das simulações

O modelo proposto apenas implementa as características básicas do protocolo BitTorrent e não suporta nenhuma das mais de 20 extensões propostas ao protocolo.

As simulações foram realizadas em um computador desktop com as seguintes configurações:

- CPU Core i7-3770 Quad-core @ 3,40 GHz (Hyper-threading foi desativado para melhorar o desempenho de processamento por núcleo);
- Memória RAM 16 Gbytes – DDR3 (foram adicionados 48 Gbytes de *swap*);
- Disco SSD 120 Gbytes;
- Sistema operacional Ubuntu 14.04 LTS;
- NS-3 versão 3.20;
- Pacote do BitTorrent VODSim versão 0.3.4.1.

Com o objetivo de estudar o desempenho de uma distribuição de dados (pacote de atualização de sistema operacional, por exemplo) utilizando o protocolo BitTorrent em ambientes corporativos, foram modelados no simulador de redes NS-3 dois cenários típicos de redes corporativas: um simples com duas redes interligadas e outro mais complexo com várias

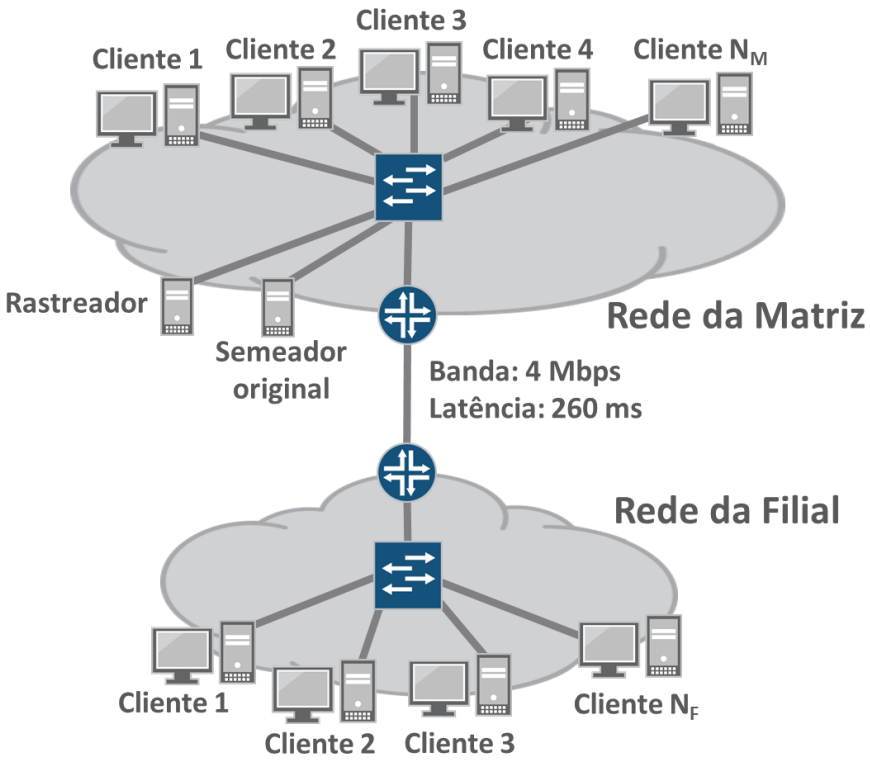
redes. Cada uma destas redes pode ser considerada uma localização, pois elas possuem faixas de endereço identificadas pela máscara 255.255.255.0 (/24). Além disso, ainda foram realizadas simulações no primeiro cenário para avaliar a robustez da modificação proposta. O detalhamento dos cenários, a motivação dos testes, as métricas medidas e resultados serão abordados nas próximas seções.

4.1 Primeiro cenário – Análise de Desempenho

No primeiro cenário foram criadas uma rede local chamada de Matriz e outra chamada de Filial, sendo essas redes conectadas através de um enlace que possui características de pequena banda e alta latência quando comparadas às das redes locais. Além dos testes utilizando o comportamento padrão do rastreador na resposta à solicitação de pares do enxame, isto é, a seleção aleatória dos pares registrados, também foram realizados testes utilizando a seleção de pares por sub-rede no rastreador, conforme apresentada no Capítulo 3.

O cenário modelado no NS-3 está representado na Figura 5. A rede Matriz possui N_m nós onde são executados clientes BitTorrent, mais um semeador original, que também é um cliente, porém já possui o arquivo a ser distribuído. O semeador original faz o papel do servidor onde, a princípio, ficam armazenados os dados que devem ser distribuídos pelo protocolo BitTorrent. O rastreador responsável pela divulgação dos nós que pertencem ao enxame também se encontra na rede Matriz. Todos estes nós e mais um outro que faz o papel de roteador estão conectados a outro, que faz o papel de comutador, formando uma rede local com alta banda (1 Gbps) e baixa latência (menor que 1 ms).

Figura 5 - Cenário simulado com as redes locais Matriz e Filial.



Fonte: O autor, 2015.

A rede Filial possui N_f nós onde também são executados clientes BitTorrent. A princípio, não há nenhum nó que possua o arquivo completo na rede Filial. Os nós da rede Filial devem se registrar no rastreador da rede Matriz para conhecer o enxame. Assim como na rede Matriz, os nós da rede Filial, incluindo um nó roteador, estão conectados a um outro, que faz o papel de comutador, formando uma rede local com alta banda (1 Gbps) e baixa latência (menor que 1 ms).

As redes Matriz e Filial são interligadas através de um enlace entre os nós roteadores. Este enlace possui banda de 4 Mbps e 260 ms de latência. Estes parâmetros de banda e latência foram escolhidos para simular um cenário onde a Filial é um site remoto, atendido por um enlace de longa distância via satélite, e com pouca banda disponível, como um posto de gasolina, mercado ou até uma plataforma de petróleo. O arquivo distribuído possui 10 Mbytes de tamanho. Este tamanho foi escolhido, porque, além de ser um tamanho típico de *patch* de atualização de sistema operacional (MICROSOFT, 2015), tem também é suficiente para a análise do comportamento do BitTorrent no cenário simulado. Todos os nós da Matriz e da Filial se registram no rastreador como uma *flash crowd*, isto é, assim que a simulação começa, todos os nós tentam se registrar ao mesmo tempo no rastreador.

Durante as simulações, o cenário varia da seguinte maneira: é mantido o valor de N_f em 20 nós e o valor de N_m varia, sendo definido como 40, 60, 80 e 100 nós. Para cada valor de N_m, são executadas 10 repetições sem a modificação no rastreador e outras 10 com a modificação.

Para a avaliação de desempenho, utilizaremos as seguintes métricas:

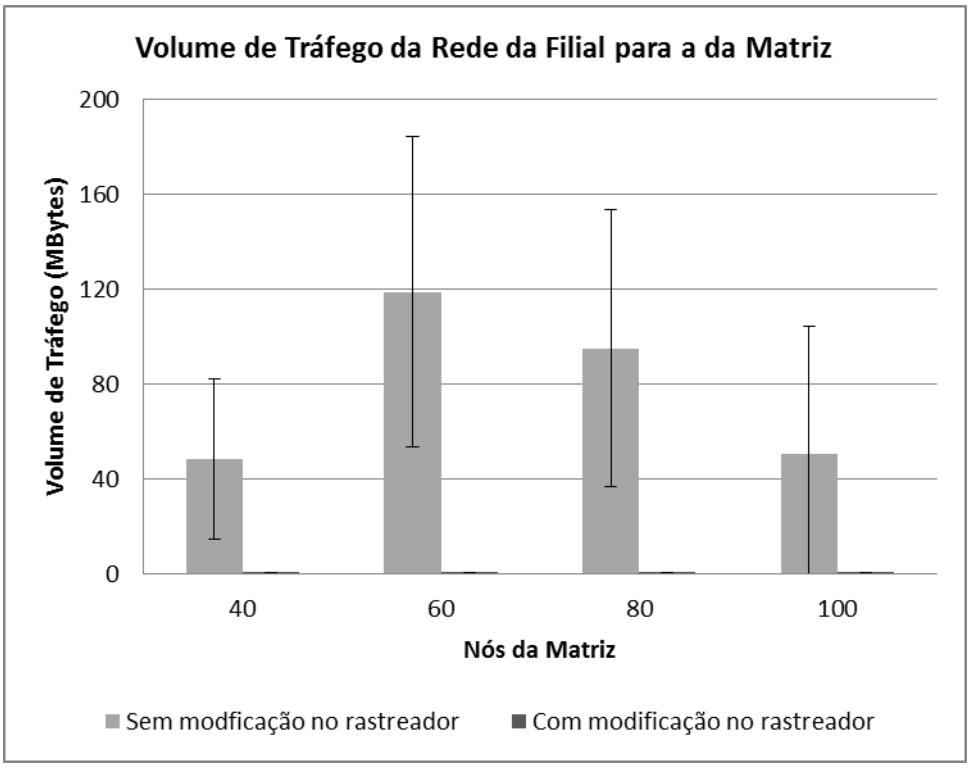
- **tráfego da Matriz para Filial e o tráfego da Filial para a Matriz** (em Mbytes) - É o tráfego que passa pelo enlace de 4 Mbps em cada sentido. Quanto menor, melhor, pois há economia de um recurso caro que é o enlace de longa distância entre a Matriz e a Filial;
- **tempo médio de conclusão do *download*** (em segundos) - Para os nós da Matriz e da Filial, se mede o tempo médio que os nós de cada uma das redes levam para concluir o *download*. Quanto menor, mais eficiente é a distribuição.

Os resultados médios das simulações com e sem a modificação do rastreador são comparados a seguir. Serão também mostrados os intervalos de confiança dos resultados médios com nível de confiança de 90%.

4.1.1 Resultados

Nas Figuras 6 e 7, é apresentado o volume de tráfego médio que transita entre a rede da Matriz e da Filial durante a transferência do arquivo, com e sem a modificação do rastreador.

Figura 6 - Volume de tráfego da rede da Filial para a da Matriz – 4 Mbps.



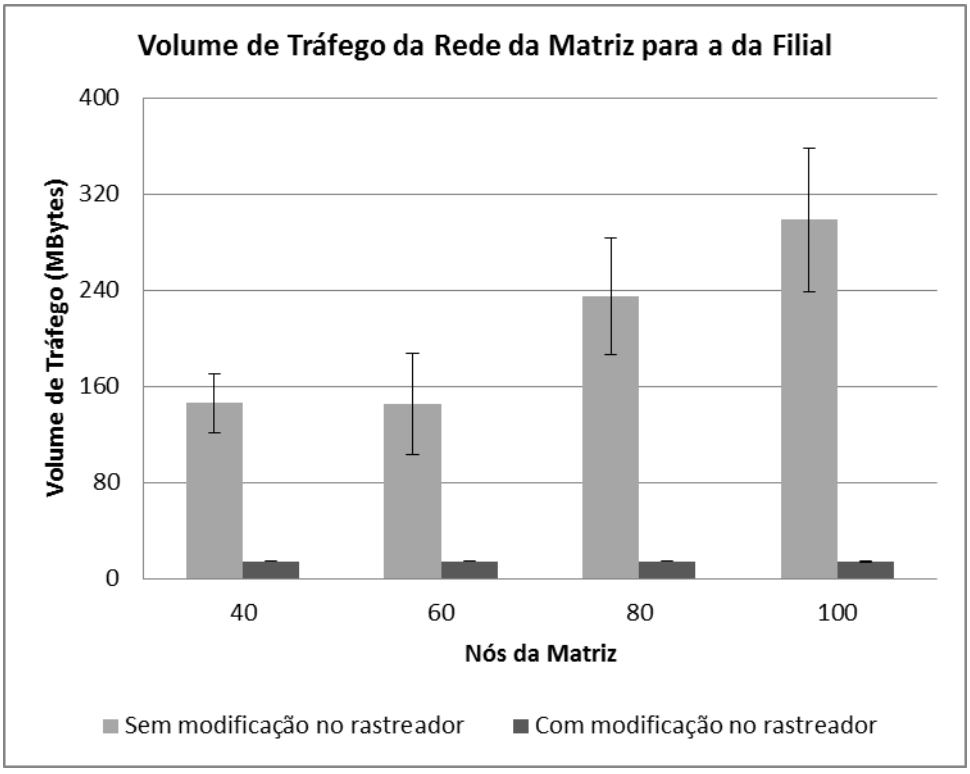
Legenda: Volume de tráfego da rede da Filial para a da Matriz em função da quantidade de nós da Matriz (enlace com banda de 4 Mbps e 260 ms de latência).

Fonte: O autor, 2015.

Na Figura 6, podemos observar que, para o tráfego da Filial para a Matriz, sem a modificação do rastreador, o volume varia muito, produzindo intervalos de confiança grandes, subindo entre 40 e 60 nós, mas com tendência de queda na medida em que se aumenta o número de nós da Matriz. Este comportamento pode ser explicado da seguinte forma: como o rastreador informa para cada nó até 50 pares (THEORY.ORG, 2015) participantes do enxame, com 40 nós temos, geralmente, uma troca rápida na Matriz, já que todos os nós se conhecerão e informarão uns aos outros assim que as partes estiverem completadas, privilegiando a troca local. É importante ressaltar que todos os nós da Matriz vão se registrar no rastreador e estabelecer conexões entre si, antes do primeiro nó da Filial conseguir se registrar no rastreador, devido ao enlace que possui alta latência. Porém, a partir de 60 nós já há uma maior probabilidade de nem todos os nós estarem conectados entre si na Matriz, o que leva a um aumento do tempo médio de *download*. Porém, quanto maior o número de clientes na Matriz, maior é a probabilidade que eles sejam escolhidos na lista de pares aleatória que o rastreador envia aos clientes, tanto para os clientes da Filial, quanto para os clientes da própria Matriz. Então, há uma tendência de que os clientes da Matriz recebam mais solicitações de partes de ambas as redes, Matriz e Filial, na medida em que aumenta a quantidade de clientes

na Matriz. Porém, quando o envio de blocos das partes ocorre dentro da própria rede Matriz, o tempo de *download* de cada parte é menor em comparação com as partes oriundas da rede Filial, pois há maior velocidade na rede local do que no enlace entre as redes. Desta forma, durante um mesmo intervalo de tempo, mais blocos são enviados de dentro da própria rede Matriz do que da rede Filial, reduzindo a contribuição total desta última e diminuindo o volume médio de bytes trafegados no sentido da Filial para a Matriz. Porém, como o processo de formação da lista de pares é aleatório, o volume de tráfego medido varia significativamente a cada rodada do experimento, produzindo grandes intervalos de confiança. Com a modificação no rastreador, o comportamento muda radicalmente. Praticamente, não há envio de informações para a Matriz, com o volume de, aproximadamente, 600 kbytes em média e um intervalo de confiança pequeno. Há apenas tráfego de controle para o rastreador e solicitações para o semeador original.

Figura 7 - Volume de tráfego da rede da Matriz para a da Filial – 4 Mbps.



Legenda: Volume de tráfego da rede da Matriz para a da Filial em função da quantidade de nós da Matriz (enlace com banda de 4 Mbps e 260 ms de latência).

Fonte: O autor, 2015.

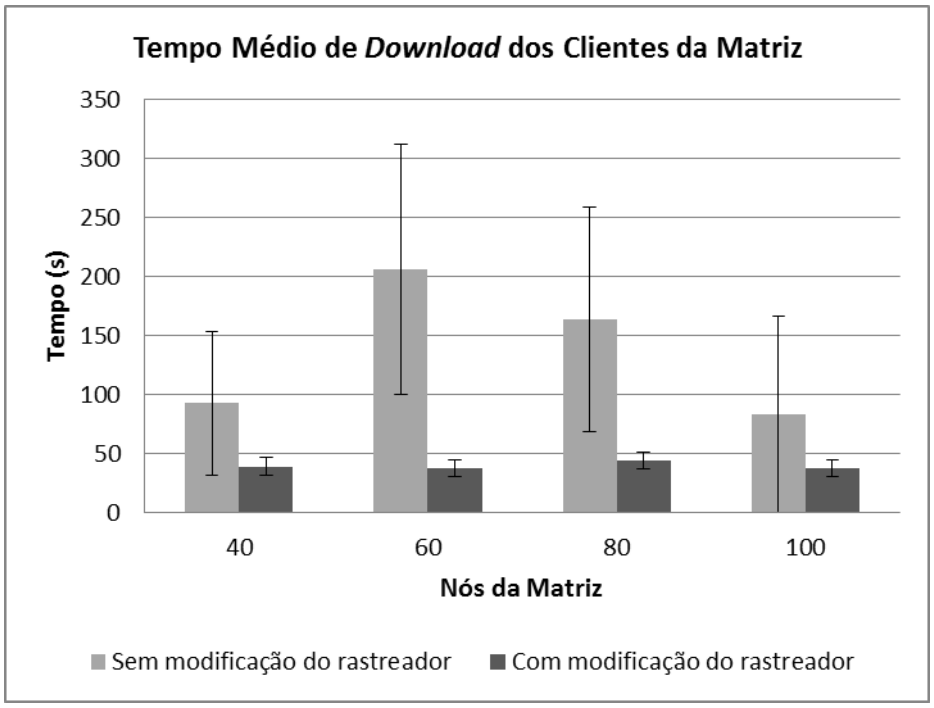
Na Figura 7, confirmamos o comportamento estudado, pois, sem a modificação no rastreador, na medida em que aumentamos o número de clientes na Matriz, há também um aumento no número de solicitações de partes para esta rede pelos clientes da Filial, o que provoca um aumento gradativo do volume médio de tráfego no sentido da Matriz para a Filial.

O efeito da maior velocidade no *download* dos blocos das partes na rede local da Filial não é suficiente para anular o efeito do aumento de solicitações para a rede Matriz. Mais uma vez, devido ao processo aleatório da formação da lista de pares, temos uma significativa variação no intervalo de confiança. Com a modificação no rastreador, os clientes da rede da Filial só conhecem a si mesmos e o semeador original, que fica na Matriz; e apenas os dados deste último são enviados pela rede de interligação entre a Matriz e a Filial. Na média, são enviados, aproximadamente, 14 Mbytes, com *overhead* dos protocolos BitTorrent e TCP/IP, bem menos que os 200 Mbytes necessários, se fosse utilizada a arquitetura cliente-servidor para o envio do arquivo a 20 clientes da Filial. Neste caso, o intervalo de confiança é pequeno.

Nas Figuras 8 e 9, é apresentado o tempo médio necessário para a transferência do arquivo na rede da Matriz e na da Filial, com e sem a modificação do rastreador.

Na Figura 8, vemos o impacto que a transferência de partes dos arquivos pelo enlace que liga as redes Matriz e Filial produz no tempo médio de *download*, pois, sem modificação no rastreador, vemos que a variação é semelhante a que ocorre na Figura 6. Isto é, excluindo a variação entre 40 e 60 nós, o *download* dos arquivos pelos clientes da Matriz fica mais rápido na medida em que há mais clientes na Matriz e, consequentemente, menos dados sendo trafegados pelo enlace entre a Matriz e a Filial, que possui menor banda disponível. Novamente, vemos os grandes intervalos de confiança gerados pelo processo aleatório de criação da lista de pares pelo rastreador. Com a modificação no rastreador, não há envio de partes da Filial para a Matriz e, desta forma, não há gargalos que atrasem significativamente o *download* dos arquivos pelos clientes da Matriz. No gráfico, o tempo médio varia entre 35 e 45 s, com intervalo de confiança pequeno.

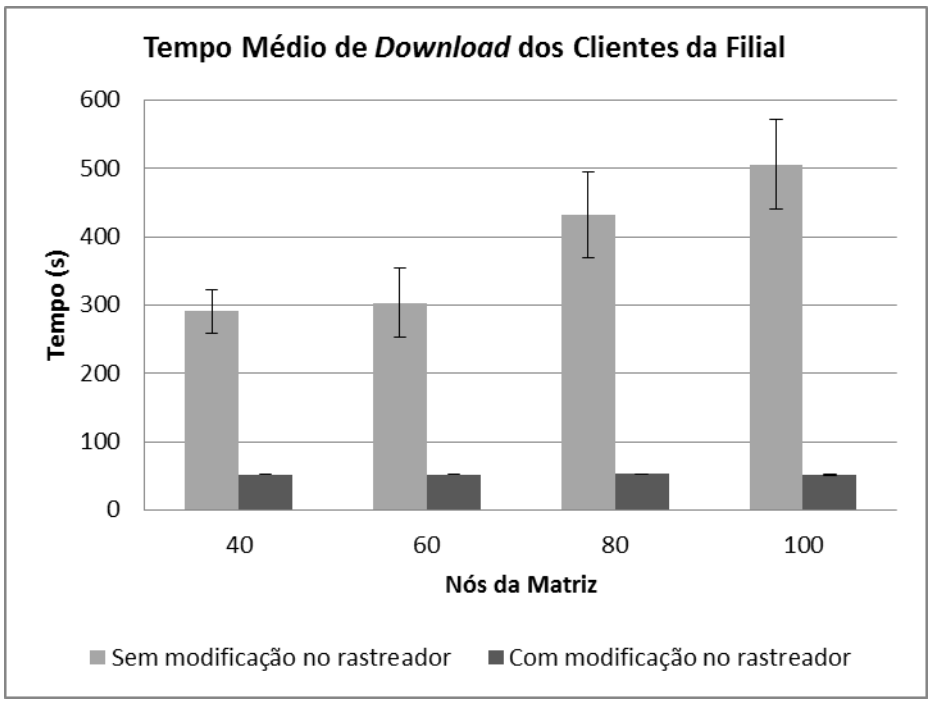
Figura 8 - Tempo médio de download dos clientes da Matriz – 4 Mbps.



Legenda: Tempo médio de *download* dos clientes da Matriz em função da quantidade de nós da Matriz (enlace com banda de 4 Mbps e 260 ms de latência).

Fonte: O autor, 2015.

Figura 9 - Tempo médio de download dos clientes da Filial- 4 Mbps.



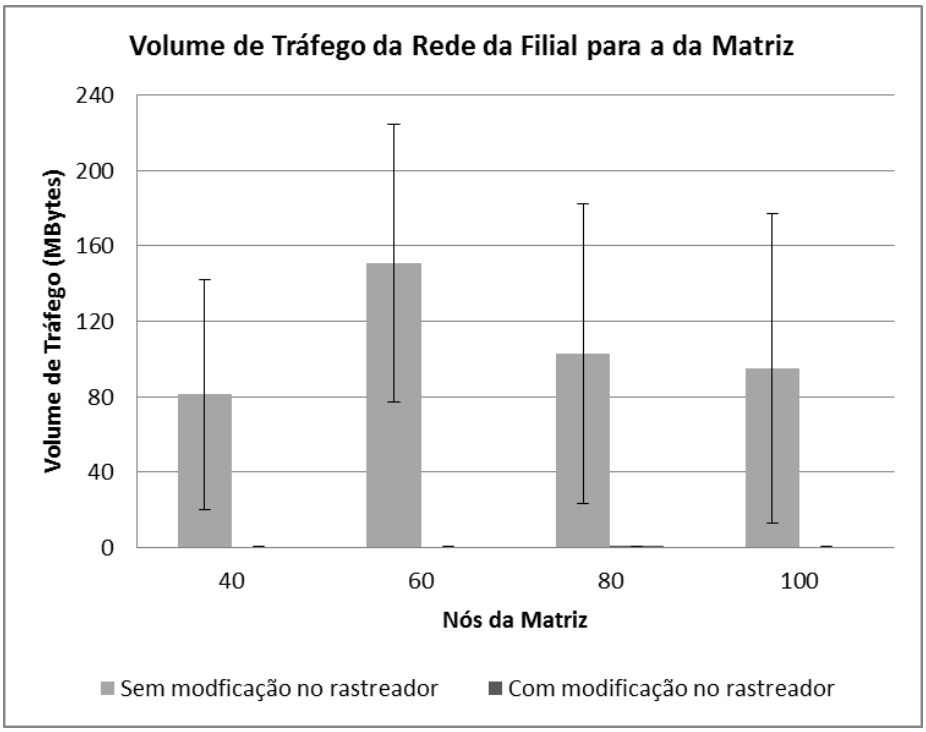
Legenda: Tempo médio de *download* dos clientes da Filial em função da quantidade de nós da Matriz (enlace com banda de 4 Mbps e 260 ms de latência).

Fonte: O autor, 2015.

Na Figura 9, o comportamento é semelhante ao já analisado: sem a modificação no rastreador, o envio de dados pelo enlace entre a Matriz e a Filial aumenta o tempo médio necessário para a conclusão dos *downloads* pelos clientes da Filial. Então, quanto mais clientes houver na Matriz, mais lento será o *download* nos clientes da Filial. Com a modificação no rastreador, o tempo médio de *download* varia entre 50 e 60 s, com intervalo de confiança pequeno.

Como neste cenário utilizamos um enlace de longa distância com banda muita baixa e uma alta latência, podemos ser induzidos a imaginar que, se utilizarmos um enlace com maior banda e menor latência, os resultados possam ser um pouco diferentes e a modificação proposta não seja tão eficiente em melhorar o desempenho da distribuição. Então, para dirimir este tipo de dúvida, repetimos as simulações apenas alterando o enlace, que agora terá 155 Mbps de banda e latência de 20 ms. Os resultados podem ser acompanhados nas figuras a seguir.

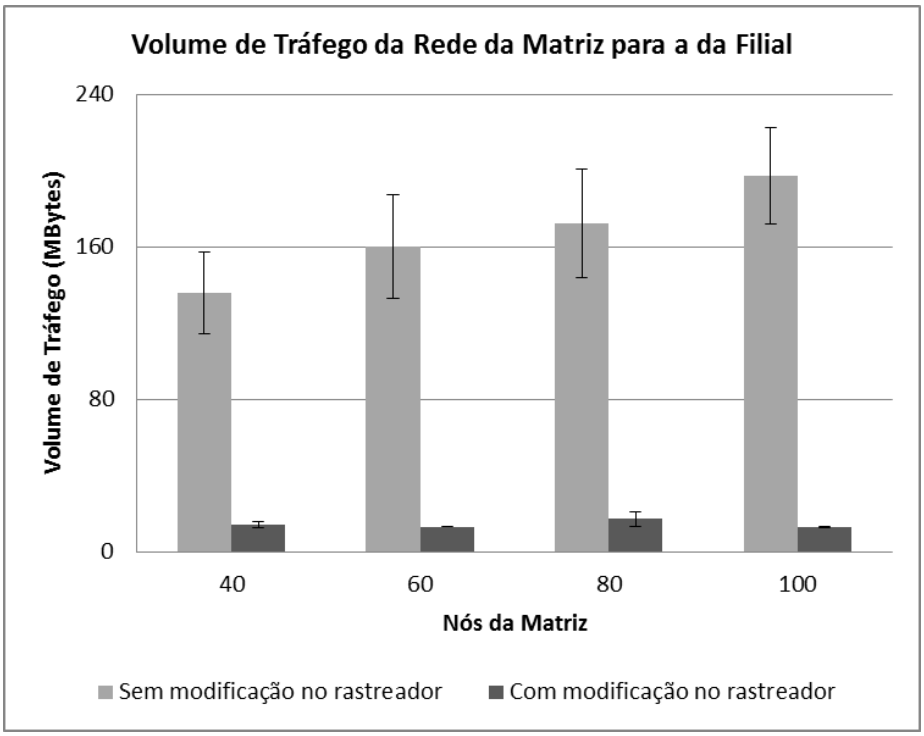
Figura 10 - Volume de tráfego da rede da Filial para a da Matriz – 155 Mbps.



Legenda: Volume de tráfego da rede da Filial para a da Matriz em função da quantidade de nós da Matriz (enlace com banda de 155 Mbps e 20 ms de latência).

Fonte: O autor, 2015.

Figura 11 - Volume de tráfego da rede da Matriz para a da Filial – 155 Mbps.



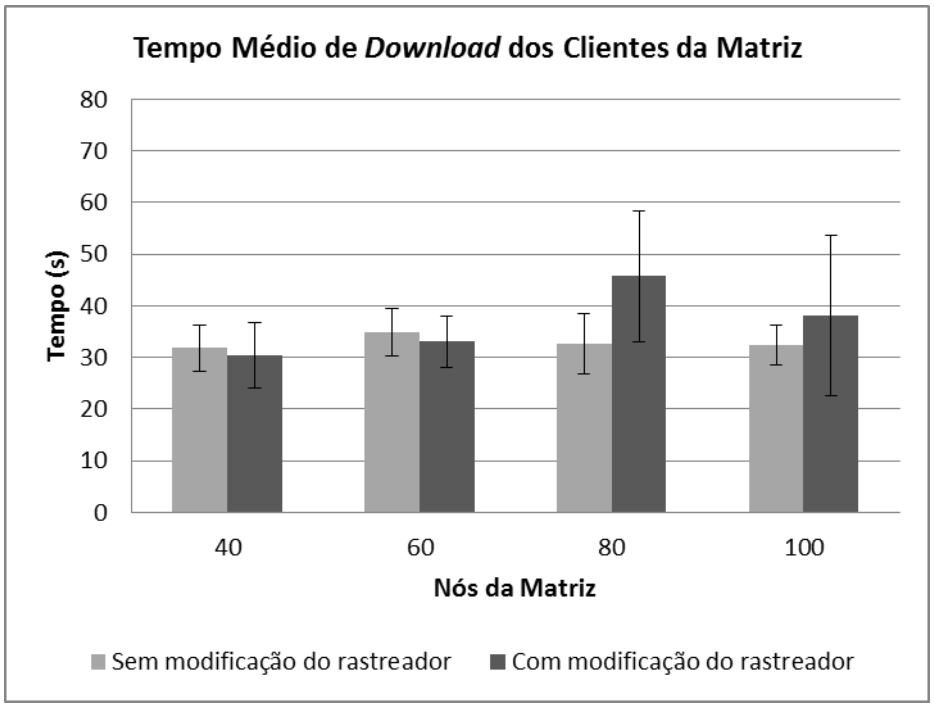
Legenda: Volume de tráfego da rede da Matriz para a da Filial em função da quantidade de nós da Matriz (enlace com banda de 155 Mbps e 20 ms de latência).

Fonte: O autor, 2015.

Comparando as Figuras 6 e 7 com as Figuras 10 e 11, vemos que, sem a modificação no rastreador, temos comportamentos similares à medida que aumentamos o número de nós na Matriz. Porém vemos que volume de tráfego da Rede da Filial para a da Matriz aumenta em todos os casos, quando comparamos o volume para cada quantidade de nós. Isto se explica da seguinte forma: antes, quando se realizava a distribuição, havia uma diferença muito grande de banda entre as redes locais, com 1 Gbps, e o enlace de longa distância, 4 Mbps. Agora, esta diferença foi reduzida, já que estamos utilizando um enlace de longa distância com 155 Mbps. Isto diminui a vantagem que as transferências dos blocos têm na rede local em relação a aquelas feitas pelo enlace de longa distância, onde há menos banda, maior latência e maior possibilidade de congestionamento. Como o envio dos blocos ocorre mais rapidamente, mais deles são enviados durante a distribuição do arquivo, aumentando o volume de tráfego da Rede da Filial para a da Matriz nestas simulações. Já quanto ao volume de tráfego da Rede da Matriz para a da Filial, podemos ver que para 40 e 60 nós na Matriz, o volume é muito próximo das simulações anteriores, porém é significativamente menor (abaixo dos intervalos de confiança) para 80 e 100 nós, o que se justifica pelo menor número de retransmissões necessárias num cenário com maior banda disponível. Observando os

gráficos com a modificação, vemos curvas quase idênticas às simulações anteriores, com apenas o tráfego de controle da Rede da Filial para a da Matriz e com um tráfego de, aproximadamente, 14 Mbytes da Rede da Matriz para a da Filial.

Figura 12 - Tempo médio de download dos clientes da Matriz – 155 Mbps.

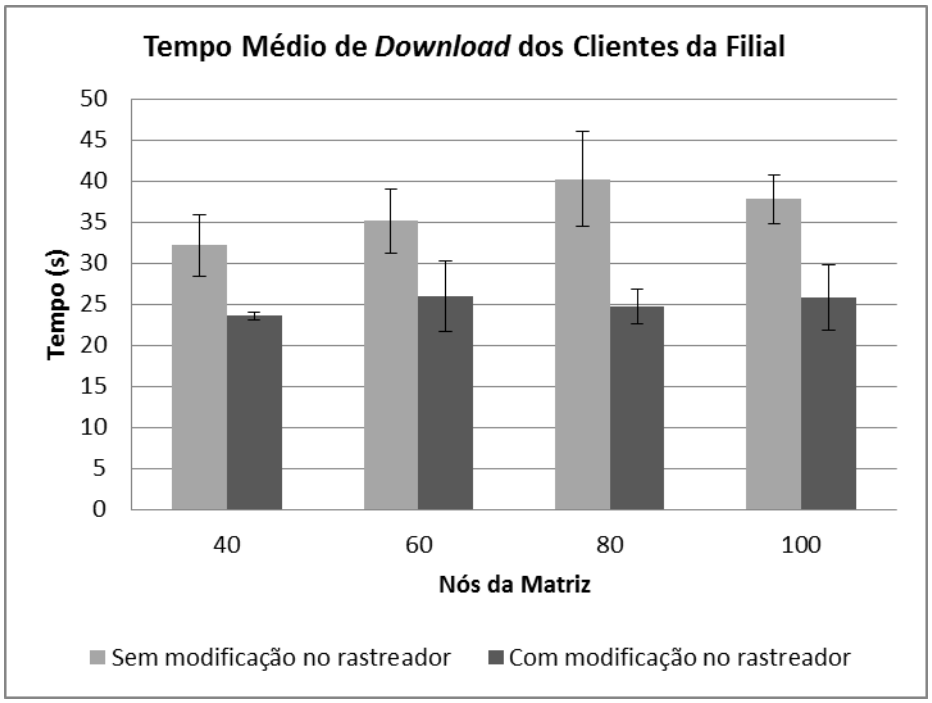


Legenda: Tempo médio de download dos clientes da Matriz em função da quantidade de nós da Matriz (enlace com banda de 155 Mbps e 20 ms de latência).

Fonte: O autor, 2015.

Observando a Figura 12 e comparando com a Figura 8, vemos que, mesmo sem a modificação, os tempos de *download* dos clientes da Matriz caem significativamente, ficando bem próximos, e até com média dos tempos medidos com a modificação. Esta redução é esperada, já que, com o aumento significativo de banda no enlace que interliga as redes, temos uma diminuição do impacto da queda de velocidade da transferência de dados entre os pares da Rede da Matriz e os pares da Rede da Filial. Neste caso, do ponto de vista do tempo de *download* da Rede da Matriz, a modificação já não garante uma melhoria no desempenho da distribuição. Observando a Figura 13 e comparando com a Figura 9, vemos que, sem a modificação, há uma grande redução nos tempos de *download* dos clientes da Filial. Novamente, temos o aumento de banda do enlace entre a Filial e a Matriz contribuindo para a redução destes tempos. Neste caso, do ponto de vista do tempo de *download* da Rede da Filial, ainda há alguma melhoria no desempenho da distribuição, quando se aplica a modificação proposta.

Figura 13 - Tempo médio de download dos clientes da Filial – 155 Mbps.



Legenda: Tempo médio de download dos clientes da Filial em função da quantidade de nós da Matriz (enlace com banda de 155 Mbps e 20 ms de latência).

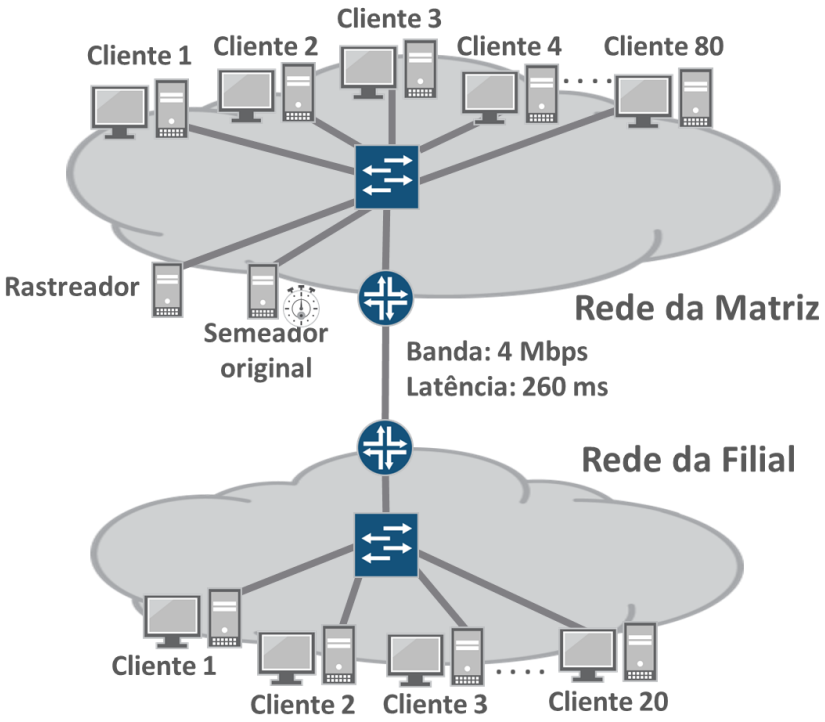
Fonte: O autor, 2015.

Concluímos que com o aumento de banda e a redução da latência no enlace que interliga as redes, temos uma redução nos ganhos de desempenho na métrica de tempo de *download*, porém ainda temos resultados significativos na redução do volume de tráfego que é trocado entre a Rede da Filial e a Rede da Matriz.

4.2 Primeiro cenário – Análise de Robustez

Em continuidade ao estudo do primeiro cenário, estudaremos a robustez da modificação proposta. Assim, realizamos algumas pequenas modificações no primeiro cenário para realizar os testes desejados, conforme apresentado na Figura 14.

Figura 14 - Redes locais Matriz e Filial e o semeador original temporizado.



Fonte: O autor, 2015.

Como pode ser observado, o cenário é semelhante ao anterior, porém fixaremos N_m em 80, deixaremos N_f em 20 e o semeador original será temporizado de maneira a ficar apenas alguns segundos como semeador, sendo realizadas simulações com a temporização igual a 30, 35, 40, 45, 50, 55 e 60 s. O tempo máximo de temporização, 60 s foi escolhido por ser, aproximadamente, o tempo médio de *download* dos clientes da Filial e o tempo mínimo, 30 s, por ser metade do tempo máximo. Como a ausência do semeador original na rede P2P pode levar à situação dos pares nunca concluírem os *downloads*, a simulação será sempre interrompida após 120 s, o dobro do tempo máximo de temporização. Para cada valor de temporização, são executadas 10 repetições sem a modificação e outras 10 com a modificação. O objetivo deste cenário é verificar se a proposta apresenta um comportamento mais robusto em caso de falha da infraestrutura, neste caso específico, do semeador original. Nesta situação, a falha do semeador original, ou até a falha do enlace que liga as redes Matriz e Filal pode provocar um problema que é chamado na literatura de "particionamento do torrent", como explicado em Le Blond, Legout e Darbbous (2011), e ocorre geralmente quando se isolam os clientes uns dos outros, muitas vezes atendendo a alguma política de localização. O particionamento pode provocar uma interrupção na distribuição do arquivo, pois o conjunto de pares que se conhecem não possui o arquivo completo e não tem como buscar as partes faltantes em outros pares.

Para a avaliação de desempenho, utilizaremos a seguinte métrica:

- **quantidade de partes distintas do arquivo distribuídas para a rede da Filial** – Como a proposta de modificação tem como objetivo isolar as redes, é interessante garantir que a modificação não prejudique o envio das partes para a Rede Filial, que está isolada pelo enlace de baixa banda e alta latência. O envio de todas as partes possibilita a conclusão do *download*, mesmo sem o semeador original. Quanto mais partes são enviadas, mais eficiente é a distribuição.

Uma informação importante para a interpretação dos resultados é que o arquivo de 10 Mbytes foi dividido em 306 partes, sendo 305 partes com 32768 bytes e a última parte menor, com 5760 bytes.

Os resultados médios das simulações com e sem a modificação do rastreador serão comparados. Serão também mostrados os intervalos de confiança dos resultados médios com nível de confiança de 90%.

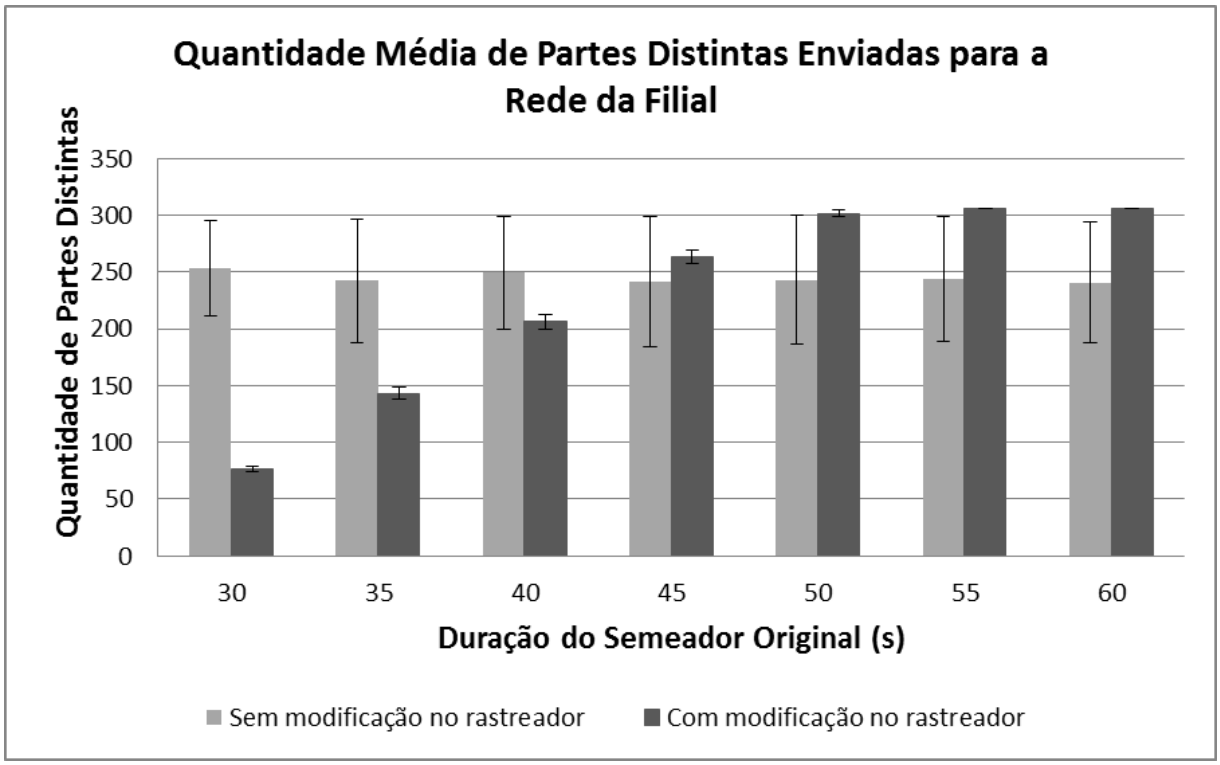
4.2.1 Resultados

Na Figura 15, podemos ver os resultados das simulações com e sem a modificação no rastreador.

O comportamento do envio de partes para a Rede da Filial com a modificação do rastreador mostra que a quantidade de partes distintas enviadas depende diretamente do tempo em que o semeador original permanece operando, o que é coerente, pois, neste caso, ele é o único par que os clientes da Rede da Filial conhecem que tem todas as partes do arquivo e é dele que eles têm que buscar os dados. Para todas as rodadas de teste, a partir de 55 s de funcionamento do semeador original, todas as partes do arquivo conseguem ser transferidas para a Rede da Filial. Podemos então concluir que, neste cenário, depois de 55 s de funcionamento, a Rede da Filial não depende mais do semeador original para concluir o *download* e poderia até ficar isolada, devido a uma falha no enlace, por exemplo, sem que a distribuição ficasse comprometida. Já o comportamento do envio de partes para a Rede da Filial sem a modificação do rastreador mostra a fraca dependência do tempo de duração do semeador original, pois mesmo sem ele o envio de partes continua da Rede da Matriz para a Rede da Filial. A quantidade de partes enviadas, neste cenário, depende do tempo em que a

simulação ficou operando, 120 s. É importante citar que, sem a modificação, independente da temporização do semeador original, em nenhuma das rodadas de simulação o *download* do arquivo foi completado. Neste caso, vemos também que os intervalos de confiança são maiores que nas simulações com a modificação e isto se dá porque a eficiência na transferência das partes depende de decisões aleatórias como a escolha aleatória dos nós pelo rastreador e as decisões de liberação otimista. Com a modificação do rastreador, a eficiência da transferência depende mais da conectividade ao semeador original do que da escolha aleatória de pares, que vai ficar restrita à Rede da Filial, e da liberação otimista.

Figura 15 - Partes distintas enviadas para a Rede da Filial.



Legenda: Quantidade média de partes distintas enviadas para a Rede da Filial, durante 120 s, em função do tempo de duração do semeador original.

Fonte: O autor, 2015.

É interessante notar que, sem a modificação do rastreador, as partes distintas são enviadas para a Rede da Filial sem depender exclusivamente do semeador original, porém uma falha na infraestrutura que interrompa a comunicação entre a Rede da Matriz e a Rede da Filial, mesmo após 120 s de transferência, vai interromper a distribuição na Rede da Filial, pois ela ainda não possui todas as partes do arquivo. Caso ocorra este tipo de falha no cenário com a modificação no rastreador, após 55 s de transferência, a distribuição não é mais interrompida. Podemos concluir que, neste cenário e do ponto de vista da infraestrutura de conexão entre as Redes da Filial e da Matriz, a modificação do rastreador torna a distribuição

mais robusta. Apesar desta afirmação não ser verdade do ponto de vista da infraestrutura do servidor que faz o papel de semeador original, podemos considerar que uma falha em um servidor de conteúdo, que muitas vezes tem soluções de alta disponibilidade, é menos provável do que em um enlace de longa distância, via satélite, onde até as condições climáticas podem impactar no desempenho do circuito.

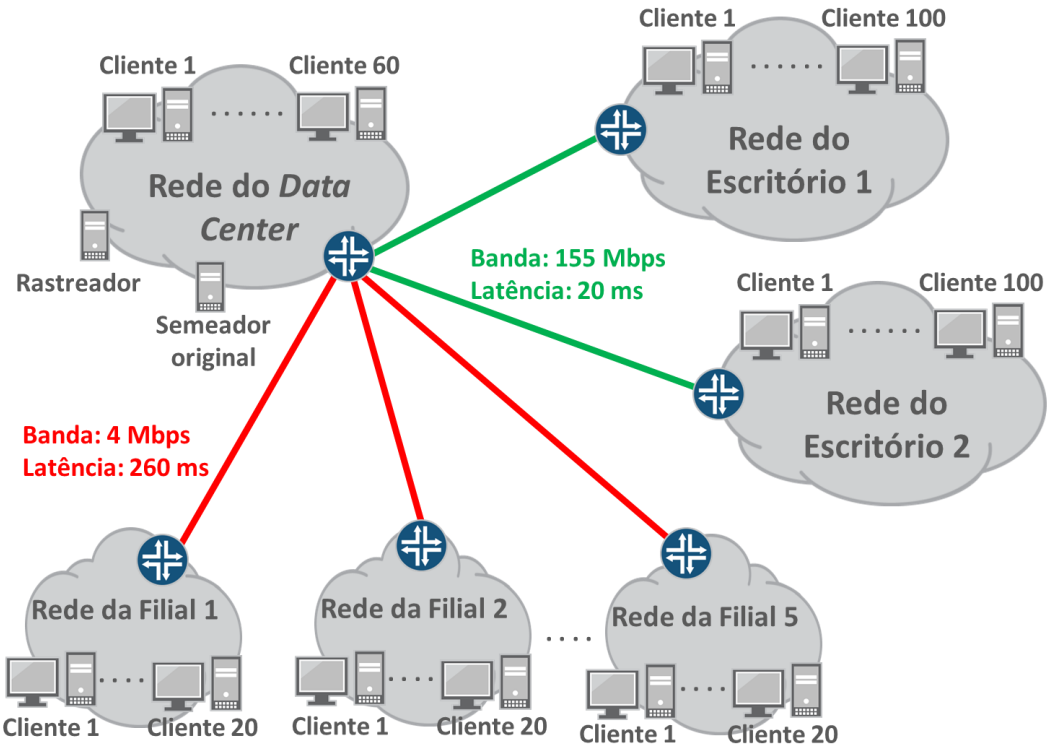
4.3 Segundo cenário

No segundo cenário, estudaremos o comportamento do protocolo BitTorrent padrão e modificado em uma rede mais complexa, que está representada da Figura 16.

Neste cenário, há 3 tipos de redes distintas:

- **Rede do *Data Center*** – Nesta rede, os nós cliente (60), o rastreador, o semeador original e um roteador estão conectados entre si através de um nó comutador, formando uma rede local com alta banda (1 Gbps) e baixa latência (menor que 1 ms). Há uma rede deste tipo no cenário;
- **Rede do Escritório** – Nesta rede os nós clientes (100) e o nó roteador estão conectados entre si através de um nó comutador, formando uma rede local com alta banda (1 Gbps) e baixa latência (menor que 1 ms). O nó roteador também se conecta ao nó roteador da Rede do *Data Center* através de um enlace com 155 Mbps de banda e 20 ms de latência. Há duas redes deste tipo no cenário;
- **Rede da Filial** - Nesta rede os nós cliente (20) e o nó roteador estão conectados entre si através de um nó comutador, formando uma rede local com alta banda (1 Gbps) e baixa latência (menor que 1 ms). O nó roteador também se conecta ao nó roteador da Rede do *Data Center* através de um enlace com 4 Mbps de banda e 260 ms de latência. Há cinco redes deste tipo no cenário.

Figura 16 - Várias redes locais conectadas a um *Data Center*.



Fonte: O autor, 2015.

Esta rede pode representar, por exemplo, uma pequena rede de varejo onde duas sedes regionais (Rede do Escritório) e cinco depósitos ou lojas (Rede da Filial) se conectam a um *Data Center* (Rede do *Data Center*), que possui servidores e alguns computadores para a equipe de suporte do *Data Center*. Com esta modelagem, propomos analisar a distribuição na presença de várias redes (cinco) que possuem enlaces com baixa banda e alta latência (Rede da Filial), juntamente com a presença de algumas redes (duas) que possuem enlaces com mais banda disponível e baixa latência (Rede do Escritório). A relação de cinco para um entre os nós da Rede do Escritório e da Rede da Filial serve para representar que o número de computadores em depósitos e lojas é, em geral, bem menor que o número de computadores em sedes regionais. O número de computadores e servidores na Rede do *Data Center* foi definido como sendo um número intermediário entre a Rede do Escritório e a Rede da Filial.

As análises que serão realizadas de maneira semelhante a do cenário 1, mas, como amostra, serão utilizados os dados de apenas uma das redes de cada tipo, isto é, a Rede do *Data Center*, uma das Redes da Filial e uma das Redes do Escritório. Por motivos de escalabilidade do simulador, para este cenário que possui um maior número de nós clientes, 361, o arquivo distribuído será menor que nas simulações anteriores e possuirá 5 Mbytes de tamanho, pois diferente do que Weingärtner et al. (2012) mostram na curva de consumo de memória, quando se aumentou o número de nós, ocorreu um estouro de memória RAM, não

sendo suficiente a quantidade total de memória disponível no computador, 64 Gbytes, para concluir a simulação deste cenário. Após algumas tentativas de diagnóstico, se encontrou uma solução de contorno reduzindo o tamanho do arquivo distribuído. São executadas 10 repetições sem a modificação no rastreador e outras 10 com a modificação.

Neste cenário, devido ao número de clientes e a topologia de rede proposta, o nó que faz o papel de roteador na Rede do *Data Center* concentrará um grande volume de tráfego. Então, para tentar evitar que haja uma grande quantidade de descartes de pacotes pelo volume instantâneo de tráfego com destinos diferentes a serem roteados pelo nó roteador do *Data Center*, aumentamos o *buffer* de saída dos nós de 100 pacotes, padrão do NS-3, para 500 pacotes.

As métricas utilizadas para avaliação de desempenho neste cenário serão:

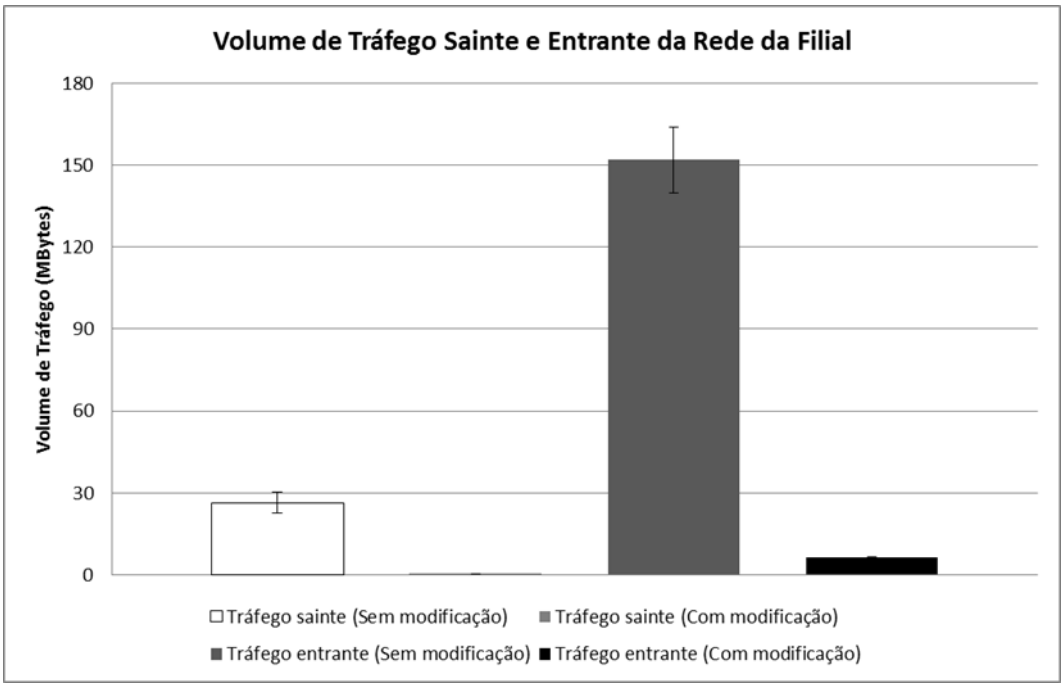
- **"tráfego entrante" na rede e o "tráfego sainte" da rede** (em Mbytes) - É o tráfego que passa pelos enlaces entre os roteadores. Quanto menor, melhor, pois há economia de um recurso caro que são os enlaces de longa distância entre as redes;
- **tempo médio de conclusão do download** (em segundos) - Para os nós das redes, se mede o tempo médio que os nós de cada uma das redes levam para concluir o *download*. Quanto menor, mais eficiente é a distribuição.

Os resultados médios das simulações com e sem a modificação do rastreador serão comparados. Serão também mostrados os intervalos de confiança dos resultados médios com nível de confiança de 90%.

4.3.1 Resultados

Nas Figuras 17, 18 e 19, podemos observar o comportamento da distribuição quanto ao tráfego entrante e sainte das redes.

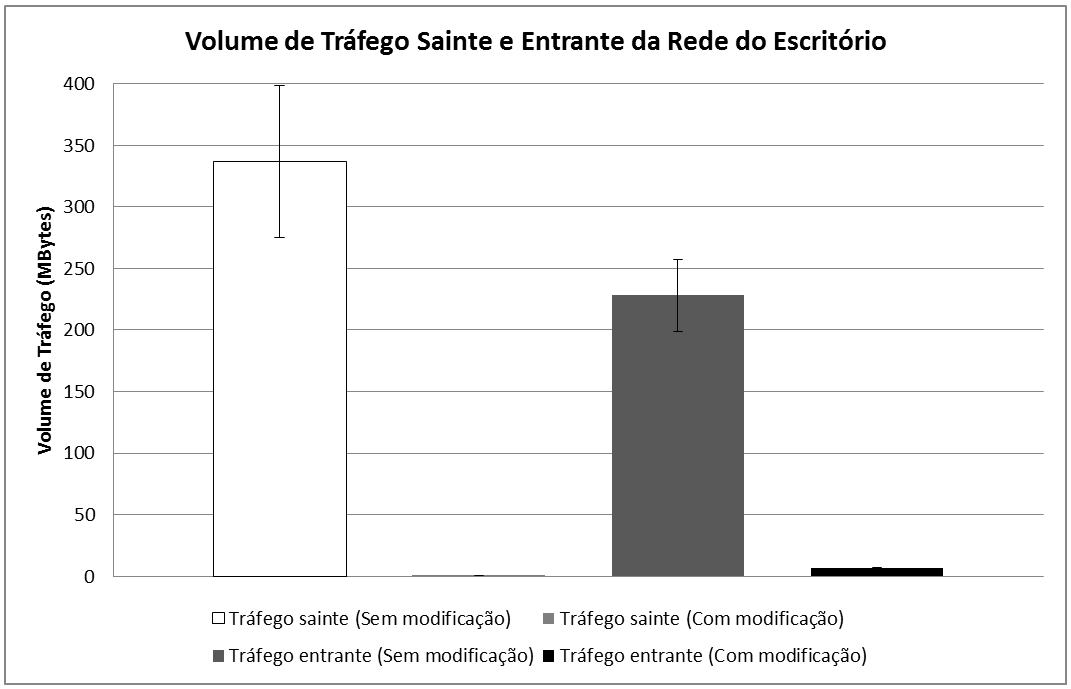
Figura 17 - Volume de tráfego sainte e entrante da Rede da Filial.



Legenda: Volume de tráfego sainte e entrante da Rede da Filial com e sem modificação no rastreador.

Fonte: O autor, 2015.

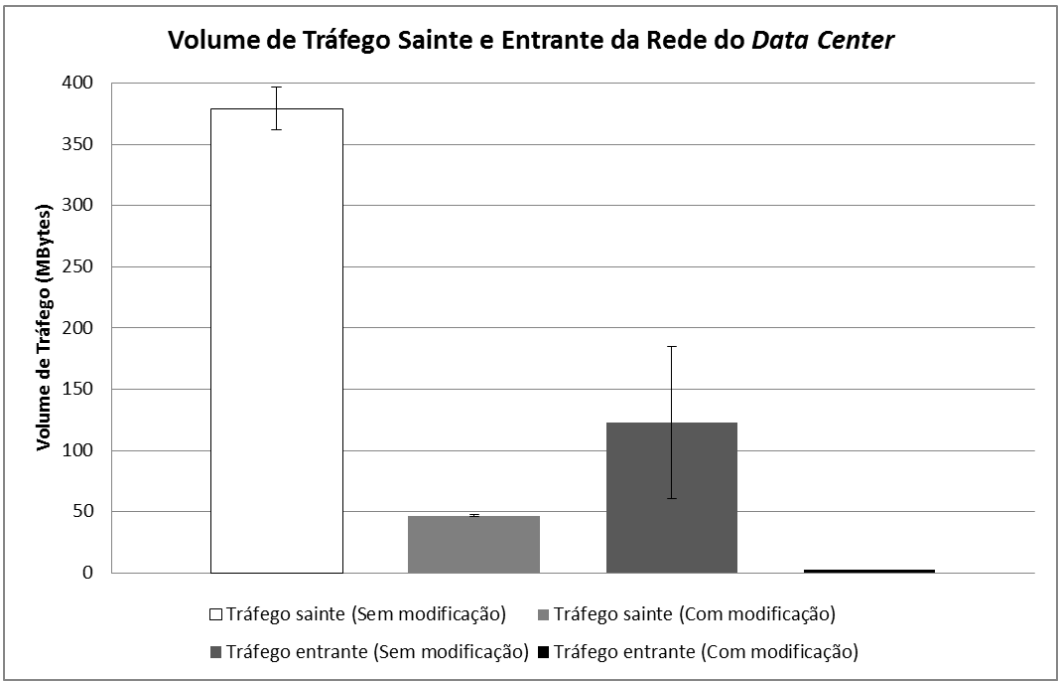
Figura 18 - Volume de tráfego sainte e entrante da Rede do Escritório.



Legenda: Volume de tráfego sainte e entrante da Rede do Escritório com e sem modificação no rastreador**.**

Fonte: O autor, 2015.

Figura 19 - Volume de tráfego sainte e entrante da Rede do *Data Center*.



Legenda: Volume de tráfego sainte e entrante da Rede do *Data Center* com e sem modificação no rastreador.

Fonte: O autor, 2015.

Podemos observar que, em geral, os efeitos da modificação no rastreador são semelhantes aos verificados no primeiro cenário. O tráfego entre as redes, assim como os intervalos de confiança, caem em todas as situações. Para a Rede do Escritório e a Rede da Filial, o tráfego sainte se resume apenas ao tráfego de controle de protocolo. Na Rede do *Data Center*, o tráfego sainte também apresenta a parcela do envio das partes do arquivo pelo semeador original.

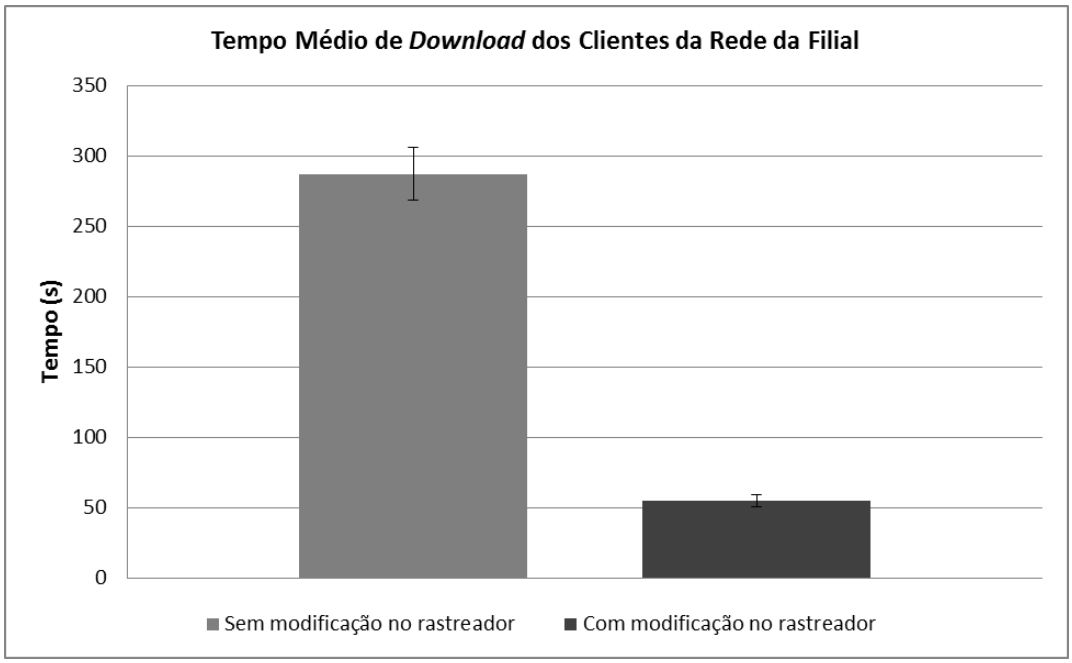
Um resultado que se destaca é o grande volume de tráfego entrante na Rede da Filial sem a modificação, em média, aproximadamente 150 Mbytes. Isto é aproximadamente trinta vezes o tamanho do arquivo completo que está sendo distribuído, 5 MBytes, e 50% a mais que o volume líquido necessário para transferir o arquivo completamente em uma arquitetura cliente-servidor, 100 Mbytes. Este resultado é justificado por dois motivos: primeiro, quanto maior o número de nós na simulação, menor a probabilidade de um cliente na Rede da Filial receber uma lista com vários pares da própria rede. Desta forma, a tendência é que os clientes da Rede da Filial façam mais conexões com clientes de outras redes e busquem deles os blocos das partes do arquivo que está sendo distribuído. Segundo, quando se buscam os blocos das partes do arquivo em clientes fora da Rede da Filial, a tendência é que haja uma maior competição do tráfego pelo enlace de baixa velocidade, provocando congestionamento, descartes e retransmissões, reduzindo ainda mais a eficiência da distribuição. Vemos então

que, neste cenário, para os clientes da Rede da Filial, uma distribuição de dados P2P, utilizando BitTorrent, não é eficiente.

Nas Figuras 18 e 19, podemos ver que, sem a modificação, a Rede do Escritório e a Rede do *Data Center* enviam um grande volume de dados para outras redes, sendo até maior do que o volume que recebem. Este comportamento é normal, já que possuem uma grande quantidade de nós e, no caso da Rede do *Data Center*, há o semeador original e, além disso, elas têm enlaces com menor latência, o que possibilita que se registrem no rastreador antes das Redes da Filial. Então, quando um nó de uma das Redes da Filial é um dos primeiros a consultar o rastreador, há uma grande probabilidade de receber quase a totalidade dos 50 nós de fora da própria rede. No caso de ser o primeiro nó de uma das Redes da Filial, todos os nós serão de fora da própria rede. E mesmo nas outras redes, há sempre a possibilidade de se fazer solicitações por blocos para fora da própria rede. Vemos também que, com a modificação do rastreador, o desempenho melhora também para a Rede do Escritório e para a Rede do *Data Center*. Na Rede do Escritório tivemos uma redução do tráfego médio entrante de 228 Mbytes para 7 Mbytes e uma redução do tráfego médio sainte de 337 Mbytes para 400 kbytes. Na Rede do *Data Center*, tivemos uma redução do tráfego médio entrante de 123 Mbytes para 2 Mbytes, pois o tráfego foi reduzido àquele necessário para o controle dos protocolos, e uma redução do tráfego médio sainte, de 379 Mbytes para 47 Mbytes, já que apenas o semeador original receberá solicitações por blocos.

Nas Figuras 20, 21 e 22, podemos ver o comportamento da distribuição quanto ao tempo médio de *download* do arquivo da distribuição pelos clientes.

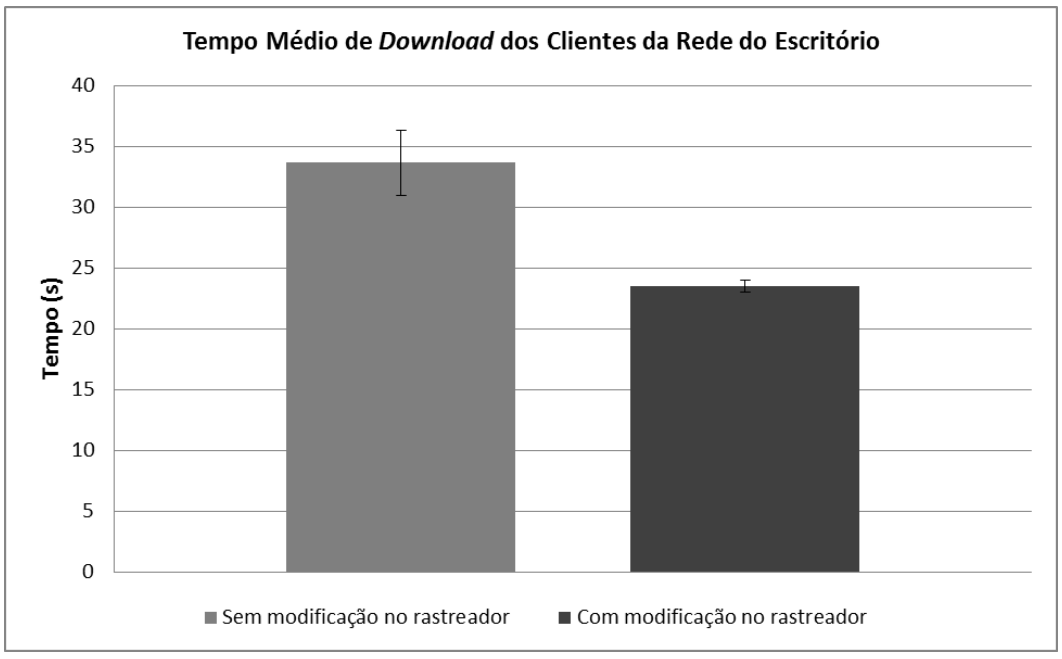
Figura 20 - Tempo médio de *download* dos clientes da Rede da Filial.



Legenda: Tempo médio de *download* dos clientes da Rede da Filial com e sem modificação no rastreador.

Fonte: O autor, 2015.

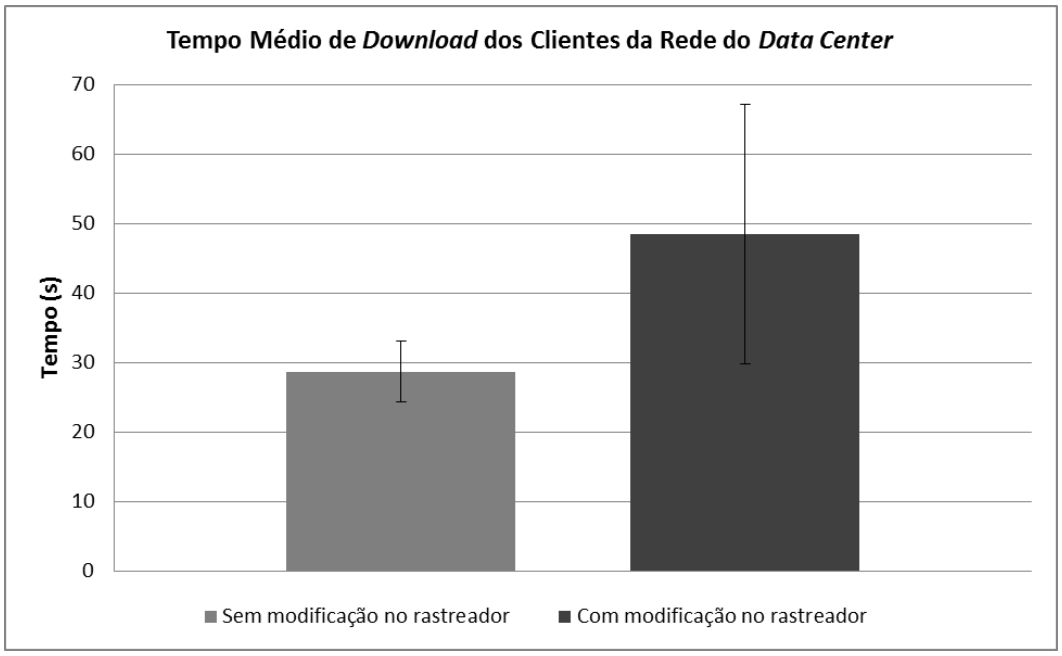
Figura 21 - Tempo médio de *download* dos clientes da Rede do Escritório.



Legenda: Tempo médio de *download* dos clientes da Rede do Escritório com e sem modificação no rastreador.

Fonte: O autor, 2015.

Figura 22 - Tempo médio de *download* dos clientes da Rede do *Data Center*.



Legenda: Tempo médio de *download* dos clientes da Rede do *Data Center* com e sem modificação no rastreador.

Fonte: O autor, 2015.

Mais uma vez, a maior parte dos resultados é semelhante ao do primeiro cenário. O tempo médio de *download* é reduzido tanto para os clientes da Rede da Filial, quanto para os clientes da Rede do Escritório, assim como há uma redução do intervalo de confiança. Porém podemos observar um comportamento peculiar nos clientes da Rede do *Data Center*: com a modificação do rastreador, o tempo médio de *download* aumenta, assim como o intervalo de confiança. Observando os dados brutos, foi visto que, para a maioria das simulações, o tempo médio fica em torno dos 20 s, porém algumas rodadas de testes produzem tempos acima dos 80 s. Este comportamento é justificado pela maior disputa do semeador original no cenário com a modificação no rastreador. Como o rastreador informa até 50 pares quando consultado, temos que na Rede do *Data Center*, todos aqueles nós que se registrarem após o semeador original o conhecerão imediatamente, a menos que mais de 50 pares da Rede do *Data Center* já tenham se registrado. Neste último caso, há uma probabilidade do semeador original não estar presente na lista. Porém, em cada Rede da Filial, todos os pares conhecerão o semeador original, pois, devido à latência do enlace, quando o primeiro cliente se registrar o semeador original já terá se registrado e quando o último cliente for se registrar, haverá apenas 20 para serem informados pelo rastreador. O mesmo fenômeno ocorre para os primeiros 50 nós das duas Redes do Escritório, destacando que os demais 50 pares podem também conhecer o semeador original através da escolha aleatória do rastreador. Podemos dizer então que, pelo

menos, 200 nós conhecerão o semeador original e tentarão contatá-lo. Se houver sucesso na conexão, como ele tem todas as partes do arquivo, será inundado de solicitações de blocos. Então, há a dependência do processo aleatório da liberação otimista do semeador original para a rapidez do *download* nas redes. Neste caso, a proximidade dos clientes da Rede do *Data Center* do semeador original é vantajosa para eles. Este processo aleatório de liberação otimista do semeador original será determinante para o tempo de *download* dos clientes da Rede do *Data Center*, pois se na repetição os nós desta rede forem privilegiados, isto é, forem escolhidos o quanto antes na liberação otimista, o *download* será mais rápido. De outro modo, o tempo de *download* será mais lento.

4.4 Comparação de resultados

No nosso trabalho, focamos em redes corporativas, que como já detalhamos, possuem características diferentes do ambiente de Internet, onde o BitTorrent se tornou popular. Em dois artigos encontramos propostas para melhorar o desempenho do protocolo BitTorrent na Internet utilizando estratégias semelhantes a aquela que foi aqui proposta, ambas utilizando ciência de localização. Porém, nos artigos, foram propostas estratégias para reduzir o tráfego nos enlaces inter-ISPs na Internet, enquanto no nosso trabalho foi proposta uma estratégia para reduzir o tráfego entre as sub-redes em redes corporativas. A seguir comentamos sobre estes trabalhos.

Em Polaczyk e Cholda (2010), podemos ver uma proposta de utilização de informações de endereçamento dos clientes e dos ISPs para que um cliente faça uma seleção orientada de pares com base na lista completa enviada pelo rastreador, através de um *plugin* em um cliente BitTorrent. Com este *plugin*, o cliente BitTorrent consulta bases de dados públicas dos RIRs (*Regional Internet Registrys* - Registros Regional da Internet) para verificar, através dos endereços IP, se os pares enviados pelo rastreador estão próximos. Estes pares então são classificados por critérios de proximidade e é dada preferência pelos pares com a menor distância, ignorando os outros. Porém, na nossa proposta, o rastreador é quem realiza a seleção orientada de pares e ele apenas envia a lista com pares dentro da mesma sub-rede, com exceção do semeador original. Em Le Blond, Legout e Darbbous (2011), são apresentados experimentos que se utilizam do conceito de localização para reduzir as conexões inter-ISP, através da ciência do rastreador sobre quem é o ISP de cada par, e onde as

restrições às redes “externas” também são severas; podendo ser comparáveis com os resultados aqui produzidos. Nesse trabalho, vemos que restrições de conexão que chegam a quatro fluxos inter-ISP produzem uma redução significativa de tráfego inter-ISP, com o resultado chegando próximo do ideal, isto é, apenas uma cópia do arquivo sendo enviada pelo enlace inter-ISP, mas não produzindo um aumento muito significativo no tempo de *download*. Este resultado está coerente com o que vimos durante as simulações no NS-3, porém neste trabalho a restrição não foi feita por conexão e, sim, a um par específico externo à rede, o semeador original. E nos nossos cenários, ainda vimos uma redução no tempo médio de *download*, quando enlaces com pouca banda estão presentes.

CONCLUSÕES

Tornar as distribuições de arquivos mais eficientes em redes corporativas é muito interessante, tanto do ponto de vista econômico, quanto do ponto de vista de segurança de rede. Reduzir o tráfego necessário para realizar a distribuição diminui a ocupação da rede e, consequentemente, também diminui a necessidade de ampliações de infraestrutura de rede e de enlaces de longa distância. A redução do tempo necessário para realizar a distribuição proporciona, por exemplo, que os sistemas operacionais e antivírus estejam atualizados o mais rápido possível, proporcionando mais segurança para as redes e computadores. Podemos tornar as distribuições de arquivo mais eficientes utilizando as redes P2P.

O protocolo P2P mais utilizado na Internet é o BitTorrent. Vários estudos mostram que podemos tornar o protocolo BitTorrent ainda mais eficiente utilizando informações de localização. Na Internet, através de protocolos BitTorrent cientes de localização, podemos reduzir o tráfego entre ISPs, gerando economia e reduzindo o tempo de *download* de arquivos. Em redes corporativas, também podemos nos beneficiar de protocolos BitTorrent cientes de localização.

Neste trabalho, propomos uma modificação simples no protocolo BitTorrent, em que o rastreador passará a levar em conta a sub-rede IP do cliente no momento de gerar a lista que será enviada em resposta à solicitação por pares. No nosso caso, para o rastreador criar a lista de pares em resposta ao cliente, ele faz uma seleção aleatória no conjunto de pares registrados que pertencem a mesma sub-rede do solicitante, sendo a única exceção o semeador original, que sempre participará da seleção aleatória, se já estiver registrado no rastreador.

Então, apresentamos resultados de simulações utilizando o protocolo BitTorrent em dois cenários corporativos para testar o desempenho da modificação proposta e compará-lo ao desempenho do protocolo BitTorrent original. No primeiro cenário, mais simples, com as redes locais conectadas por um enlace de longa distância com pouca banda e alta latência, analisando métricas de desempenho e utilizando o protocolo BitTorrent padrão, concluímos que os resultados das simulações mostraram que na medida que se aumenta o número de clientes na Matriz em relação aos clientes da Filial há uma degradação no desempenho da distribuição. Com uma modificação no rastreador para que ele utilize uma estratégia ciente de localização para gerar a lista de pares, os resultados das simulações mudaram. Os resultados se mostraram mais estáveis, com intervalos de confiança menores, e com melhor eficiência na distribuição do arquivo, atingindo tempos médios de *download* menores e com menor

utilização do enlace de longa distância. Concluímos também que, quando se utiliza um enlace com maior banda e menor latência, com a modificação proposta, ainda temos uma melhora significativa na métrica de volume de tráfego, mas os tempos de *download* ficam mais próximos do protocolo BitTorrent padrão.

Ainda utilizando o primeiro cenário, realizamos um teste de robustez para verificar se a distribuição ficava mais suscetível a falhas de infraestrutura. Analisando os resultados das simulações vimos que, para eventuais falhas no servidor que faz o papel de semeador original, no cenário proposto, utilizar o protocolo BitTorrent padrão pode ser mais vantajoso, pois garante que a Rede da Filial não dependa do semeador original para que seus clientes continuem a participar da distribuição do arquivo. Porém, para eventuais falhas no enlace que liga a Rede da Filial à Rede da Matriz, o protocolo BitTorrent com a modificação no rastreador confere maior robustez, pois consegue realizar a transferência de todas as partes distintas do arquivo com maior rapidez do que o protocolo padrão.

Estudamos também o desempenho da distribuição em um cenário mais complexo, com oito redes conectadas e com uma quantidade maior de clientes BitTorrent. Vimos que, também neste cenário, a modificação no rastreador produziu resultados melhores que o protocolo padrão para as redes remotas. Além disso, vimos que, para redes pequenas e suscetíveis a congestionamento, o desempenho do protocolo BitTorrent padrão para a distribuição de arquivos pode ser igual, ou até pior, do que quando se utiliza uma arquitetura cliente-servidor. Os mecanismos de estrangulamento e liberação não se mostram suficientes para garantir a obtenção das partes do arquivo nos clientes que podem fornecer as taxas mais altas de *upload*, sendo, no caso estudado, aqueles que ficam na mesma rede local. E este pode ser um alerta para quando se utilizar protocolos P2P com redes de sobreposição formadas de maneira aleatória, como o BitTorrent, para distribuir arquivo em cenários corporativos semelhantes aos estudados, pois dependendo dos mecanismos de controle utilizados pelo protocolo, este problema pode também ocorrer, já que não há garantias que os pares dentro da própria rede local se conheçam.

Trabalhos Futuros

Além do funcionamento do protocolo BitTorrent quanto à seleção de pares no rastreador, há outras oportunidades para se estudar o comportamento do protocolo mediante

algumas alterações; por exemplo, alterando as temporizações do protocolo, o número máximo de conexões desejadas e permitidas, o número máximo de pares solicitados pelos clientes, os mecanismos de estrangulamento e liberação, etc. Em um cenário corporativo, pode ser especialmente interessante alterar o comportamento padrão do semeador original com a finalidade de aumentar o desempenho da distribuição do arquivo, já que, provavelmente, ele estará alocado em um servidor dedicado e com mais recursos disponíveis para dedicar à distribuição.

Outra possibilidade para trabalhos futuros é, com mais recursos computacionais disponíveis, aumentar o número de redes e clientes na simulação, mudando algumas características como o enlace e a topologia da rede e, então, analisar o impacto destas variáveis na distribuição.

REFERÊNCIAS

ANDROUTSELLIS-THEOTOKIS, S.; SPINELLIS, D. A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys* (CSUR), v. 36, n. 4, p. 335-371, 2004.

BINDAL, R. et al. Improving traffic locality in BitTorrent via biased neighbor selection. In: *Distributed Computing Systems, 2006. ICDCS 2006. 26th IEEE International Conference on*. IEEE, 2006. p. 66-66.

CHOFFNES, D.; BUSTAMANTE, F. Taming the torrent: a practical approach to reducing cross-ISP traffic in peer-to-peer systems. In: *ACM SIGCOMM Computer Communication Review*. ACM, 2008. p. 363-374.

CERT.BR. Cartilha de Segurança – Segurança de computadores. Junho, 2012. Disponível em < http://cartilha.cert.br/computadores/>. Acesso em: 29 jul. 2015.

COHEN, B. Incentives build robustness in BitTorrent. In: *Workshop on Economics of Peer-to-Peer systems*. 2003. p. 68-72.

COSTA, M. RUBINSTEIN, M. Protocolo BitTorrent Ciente de Localização em Redes Corporativas. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 33, 2015, Vitória, *Workshop de Redes P2P, Dinâmicas, Sociais e Orientadas a Conteúdo (WP2P+2015)*. p. 57-70.
LE BLOND, S.; LEGOUT, A.; DABBOUS, W. Pushing BitTorrent locality to the limit. *Computer Networks*, v. 55, n. 3, p. 541-557, 2011.

LI, Z.; XIE, G. Enhancing content distribution performance of locality-aware BitTorrent systems. In: *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*. IEEE, 2010. p. 1-5.

MUNDINGER, J.; WEBER, R.; WEISS, G. Optimal scheduling of peer-to-peer file dissemination. *Journal of Scheduling*, v. 11, n. 2, p. 105-120, 2008.

MICROSOFT. Descrição das alterações no conteúdo do Software Update Services e do Windows Server Update Services de 2015. Julho, 2015. Disponível em:< https://support.microsoft.com/pt-br/kb/894199>. Acesso em: 29 jul. 2015.

NS-3. NS-3 Consortium. Portal do simulador de redes NS-3. Disponível em: <http://www.nsnam.org/>. Acesso em: 22 nov. 2014.

OECHSNER, S. et al. Pushing the performance of biased neighbor selection through biased unchoking. In: *Peer-to-Peer Computing, 2009. P2P'09. IEEE Ninth International Conference on*. IEEE, 2009. p. 301-310.

PASSARELLA, A. A survey on content-centric technologies for the current Internet: CDN and P2P solutions. *Computer Communications*, v. 35, n. 1, p. 1-32, 2012.

POLACZYK, B.; CHOLDA, P. BitTorrent traffic localization via operator-related information. In: *Communications (ICC), 2010 IEEE International Conference on*. IEEE, 2010. p. 1-5.

PUSHP, S.; RANJAN, P. A practical incentive towards efficient data sharing in large scale enterprise. In: *Networked Computing and Advanced Information Management (NCM), 2010 Sixth International Conference on*. IEEE, 2010. p. 45-50.

REN, S. et al. TopBT: a topology-aware and infrastructure-independent BitTorrent client. In: *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010. p. 1-9.

SANDVINE. Global internet phenomena report – 2H 2014. Disponível em: <https://www.sandvine.com/downloads/general/global-internet-phenomena/2014/2h-2014-global-internet-phenomena-report.pdf>. Acesso em: 26 mai. 2015.

SOMANI, M. et al. BitTorrent for large package distribution in the enterprise environment. In: *Recent Advances in Computing and Software Systems (RACSS), 2012 International Conference on*. IEEE, 2012. p. 281-286.

THEORY.ORG. Bittorrent protocol specification v1.0. Theory.org, Abril, 2015. Disponível em: <https://wiki.theory.org/index.php/BitTorrentSpecification>. Acesso em: 29 jun. 2015.

VAN DER SAR, E. Facebook uses BitTorrent, and they love it. TorrentFreak. Junho, 2010. Disponível em <http://torrentfreak.com/facebook-uses-bittorrent-and-they-love-it-100625/>. Acesso em: 22 nov. 2014.

VAN DER SAR, E. Twitter uses BitTorrent for server deployment. TorrentFreak. Fevereiro, 2010.< http://torrentfreak.com/twitter-uses-bittorrent-for-server-deployment-100210/>. Acesso em:22 nov. 2014.

WEINGÄRTNER, E. et al. Building a modular BitTorrent model for NS-3. In: *Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2012. p. 337-344.

XIE, H. et al. P4P: Provider portal for applications. In: *ACM SIGCOMM Computer Communication Review*. ACM, 2008. p. 351-362.