

UNIVERSIDADE FEDERAL DO AMAZONAS
PRO REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
DEPARTAMENTO DE APOIO À PESQUISA
PROGRAMA INSTITUCIONAL DE APOIO A PESQUISA

UMA SOLUÇÃO PARA O PROBLEMA DO BOOTSTRAP EM SISTEMAS P2P
DE PEQUENO PORTE

Bolsista: Wladimir Barroso Guedes de Araújo Neto, CNPq

MANAUS
2012

UNIVERSIDADE FEDERAL DO AMAZONAS
PRO REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
DEPARTAMENTO DE APOIO À PESQUISA
PROGRAMA INSTITUCIONAL DE APOIO A PESQUISA

RELATÓRIO FINAL
PIB – E – 0143/2011
UMA SOLUÇÃO PARA O PROBLEMA DO BOOTSTRAP EM SISTEMAS P2P
DE PEQUENO PORTE

Bolsista: Wladimir Barroso Guedes de Araújo Neto, CNPq
Orientador: Prof Dr César Augusto Viana Melo

MANAUS
2012

RESUMO DO RELATÓRIO

Redes que utilizam a Arquitetura P2P possuem diversas vantagens em comparação às redes construídas utilizando a Arquitetura Cliente-Servidor. Podem ser citadas como vantagens a escalabilidade, configuração própria e a resiliência à falha dos nós (computadores presentes na rede). Tal característica fez com que essas redes fossem adotadas em diversos serviços de larga-escala na Internet como provedores de conteúdo, transmissão de arquivos e armazenamento de dados. Em uma rede de pequeno porte, tal arquitetura pode ser utilizada para troca direta de conteúdo ou envio de mensagens sem que haja a necessidade de um ponto de coordenação/centralização.

O Problema do Bootstrap, que consiste em inicializar um nó novo na rede fazendo com que este conecte-se aos que já fazem parte dela, continua sendo um desafio à ser superado. Em redes de grande porte, esse problema pode ser contornado utilizando-se servidores dedicados que realizarão a inicialização dos nós na rede. Mas em redes de pequeno porte, tal solução ainda não foi encontrada em sua totalidade e continua sendo foco de estudo.

Neste projeto, realizou-se o estudo da Arquitetura P2P e do Problema do Bootstrap, objetivando mostrar como o Problema do Bootstrap pode ser solucionado no caso de este ocorrer em uma rede de pequeno porte. Com a realização deste estudo, mostrou-se que há duas soluções para o Problema do Bootstrap em redes de pequeno porte, caracterizadas pela utilização de dois protocolos para redes P2P: Brunet e XMPP.

Após o estudo, foi elaborado um Estudo de Caso utilizando o protocolo Brunet para ser utilizado em um cenário que compreende uma rede P2P em computadores utilizando o sistema operacional Ubuntu, com o sistema de transferência de arquivos BitTorrent. Tal Estudo de Caso consiste em uma customização realizada em um cliente BitTorrent, que deve realizar conexões ao protocolo Brunet a fim de que possa inicializar nós na rede contornando o Problema do Bootstrap.

SUMÁRIO

1.INTRODUÇÃO.....	PAG. 5
2.REVISÃO BIBLIOGRÁFICA.....	PAG. 6
3.MÉTODOS UTILIZADOS.....	PAG. 7
4.RESULTADOS E DISCUSSÕES.....	PAG.8
5.CONCLUSÕES.....	PAG. 13
6.REFERÊNCIAS BIBLIOGRÁFICAS.....	PAG. 14
7.CRONOGRAMA EXECUTADO.....	PAG. 15

1. INTRODUÇÃO

Diferentemente de uma arquitetura Cliente-Servidor, em que cada parte desempenha um papel bem definido, em um sistema P2P os pares desempenham tanto o papel de cliente quanto de servidor de dados[1]. Essa característica dos sistemas P2P permitem a implementação de aplicações naturalmente escaláveis, e de grande disponibilidade.

A escalabilidade em sistemas P2P deve-se a contribuição de cada par ao sistema, em termos de processamento e capacidade de armazenamento. Ou seja, quando um par junta-se ao sistema ele deve também desempenhar o papel de servidor, e por isso irá contribuir com uma certa capacidade de armazenamento de informação e de processamento. Ao desempenhar um papel duplo no sistema, os pares compartilham a responsabilidade pelo armazenamento e disponibilização da informação. Dessa forma, se um par falhar os demais devem manter o sistema em funcionamento.

A descentralização dos sistemas P2P mostra-se muito mais vantajosa frente a abordagem Cliente-Servidor. Aplicações Cliente-Servidor estão baseadas em um único ponto (o servidor), que se não estiver disponível comprometerá toda a aplicação. Além disso, o aumento no número de usuário do sistema impõe maior demanda sobre o servidor e requerem, para viabilização da aplicação, um aumento em sua capacidade de processamento.

A despeito dos benefícios da escalabilidade e disponibilidade, a implementação de sistemas P2P é tarefa desafiadora pois, na ausência de uma figura centralizadora, um par que chega ao sistema deve: i) adquirir um identificador global, ii) se utilizar de mecanismo para localização de pares correspondente, e iii) empregar um método que garanta a manutenção de conexões estabelecidas. Tais eventos caracterizam o Problema de Bootstrap.

Em sistemas P2P de grande porte, com dezenas de milhares de pares, os mantenedores desses sistemas têm contornado o problema de bootstrap com a utilização de servidores centrais. Essa é uma solução simples mas que apresenta um custo justificado apenas pela escala dos sistemas em questão. Ou seja, na ausência desses servidores e de um protocolo para solução do bootstrap, aos usuários caberia a resolução do problema, algo que está além do alcance dos usuários casuais.

Por outro lado, em sistemas de pequeno porte, com dezenas de nós, a utilização de servidores centrais se inviabiliza devido o custo envolvido em suas implementações, i.e, a manutenção de um servidor em atividade contínua representam um alto custo para a escala do sistema em questão. Além disso, esse servidor representar um ponto único de falha[2][3], i.e caso ele esteja indisponível todo o sistema estará comprometido.

Na impossibilidade de utilização de servidores, o problema de Bootstrap em sistemas P2P de pequeno porte requer a utilização de um protocolo que permitisse ao par: i) obter um identificador global e realizasse a sua divulgação; ii) encontrar pares correspondentes à sua busca; e, finalmente, iii) realizar a manutenção e/ou restabelecimento de conexões permitindo a continuidade da troca de informação na ocorrência de desconexões.

2.REVISÃO BIBLIOGRÁFICA

O Problema do Bootstrap vem sendo alvo de estudos há alguns anos, por ser uma questão fundamental no estabelecimento de redes que utilizam a arquitetura P2P. Tal problema consiste em como um nó novo consegue entrar em contato com uma rede e tornar-se parte da mesma.

Uma das soluções mais comuns para o Problema consiste na utilização de servidores centrais que resolveriam todos os problemas de conexões entre os nós presentes na rede e o nó que deseja ingressar na mesma. O problema dessa solução é a introdução centralizada de um ponto único de falha, i.e caso este ponto esteja indisponível todo o sistema estará comprometido. Outra abordagem é a utilização de um cache de hosts, como pode ser visto no Sistema de Web Cache Gnutella [10]. Nesse sistema, os clientes (nós) repassam ao cache suas informações de conexão, que por sua vez são passadas à outros nós.

No caso de redes em pequena escala, foco do projeto, as mesmas soluções gerais (utilizadas em redes de larga escala) também podem ser aplicadas, mas as mesmas representam um grande custo relacionado à manutenção e suporte. Dessa forma, soluções alternativas devem ser utilizadas para que a solução do Problema do Bootstrap seja tanto funcional quanto cabível as necessidades do sistema.

Segundo D. I. Wolinsky et al. [2], a solução para o Problema, no contexto de redes de pequeno porte, encontra-se na divisão do mesmo em três partes menores, caracterizadas por Reflexão, Relaying e Rendezvous, que devem ser solucionadas separadamente e depois unidas em uma só aplicação. Dessa forma, garante-se que o Bootstrap possa ocorrer da forma correta, já que cada uma das divisões refere-se aos processos que são obrigatórios para a inicialização de uma rede P2P.

Tendo em vista a aplicação do trabalho de D. I. Wolinsky et al[2], esta foi escolhida para ser utilizada como base para o projeto, por ser a solução que mais se aproxima do que foi proposto neste estudo.

3.MÉTODOS UTILIZADOS

Na primeira parte do projeto, realizou-se o levantamento bibliográfico em duas etapas: Estudo da arquitetura P2P, para entender seu funcionamento; e Estudo das soluções do Problema do Bootstrap, voltadas ao contexto do trabalho.

Durante a primeira etapa estudou-se a arquitetura P2P, em especial as formas de estruturação de uma rede, quais são seus componentes e o que é necessário para que uma rede, utilizando essa arquitetura, possa ser criada e mantida em funcionamento. Realizou-se a montagem de uma rede P2P baseada no protocolo BITTorrent, em laboratório, o que envolveu a instalação de um concentrador de arquivos *.torrent*, conhecido como *tracker* BITTorrent. Os arquivos *.torrent* contêm informações sobre o conteúdo que está sendo compartilhado. Sua finalidade, dentro de uma rede P2P, é mostrar à um cliente BitTorrent onde encontrar o esse conteúdo dentro da rede. Com a instalação do tracker, verificou-se como a rede se comportaria no contexto do projeto, tendo-se, para isso, a transferência de arquivos em uma rede de pequeno porte. A instalação do sistema testado é apresentada na Seção 4.1

Durante a segunda etapa, estudou-se as soluções existentes para o Problema do Bootstrap no contexto do projeto, ou seja, em redes de pequeno porte. Identificou-se de que forma o Problema ocorre e como ele pode ser contornado em rede P2P de pequeno. Tal estudo teve como base o a literatura[2], cujo foco é a solução do problema do Bootstrap em redes de pequeno porte utilizando os protocolos Brunet e XMPP.

A segunda parte do projeto, teve sua execução também realizada em duas etapas. Na primeira etapa conduziu-se o estudo de reconstrução da solução do Problema do Bootstrap para redes de pequeno porte, apresentado no trabalho de D. I. Wolinsky et al.[2], que está baseada na utilização dos protocolos Brunet[5][9] e XMPP[6]. Segundo D.I. Wolinsky, a utilização desses protocolos seria o suficiente para que uma rede P2P possa funcionar tratando completamente o problema do Bootstrap. Tal tratamento é realizado pelo protocolo Brunet, responsável por fazer a recepção de novos nós em uma rede e por realizar a conexão à outros nós de acordo com as requisições de cada nó. Nesse contexto, o protocolo XMPP é utilizado para expandir as capacidades do protocolo Brunet, permitindo que este possa realizar a troca de mensagens entre os nós.

Na segunda etapa, verificou-se de que forma um Estudo de Caso, envolvendo as soluções identificadas na etapa anterior, poderia ser realizado. Estudou-se o protocolo Brunet e como ele deveria ser utilizado para, em conjunto com um tracker BitTorrent, ampliar as capacidades de uma rede P2P. Tal ampliação consiste em agrupar ao sistema de transferência de arquivos BitTorrent as ferramentas de conexão presentes no protocolo Brunet, fazendo assim com que os problemas encontrados durante atividades de transferência, como perda de conexão devido a proteção por parte de Firewall/NATs, sejam contornados.

4.RESULTADOS E DISCUSSÕES

Nesta seção, apresentam-se os resultados obtidos com a realização do projeto. Primeiro, destacam-se os resultados obtidos com a realização do estudo da arquitetura Peer-to-Peer e do Problema do Bootstrap. Posteriormente, os resultados obtidos com o estudo para reconstrução da solução do Problema do Bootstrap escolhida bem como os resultados obtidos com a análise para realização do Estudo de Caso são destacados.

4.1 Estudo da arquitetura Peer-to-Peer (P2P) e do Problema do Bootstrap:

Na primeira etapa do projeto, realizou-se um estudo de como a arquitetura P2P funciona, levando em consideração o escopo do presente projeto. Após o termino do levantamento de dados, foi realizada uma aplicação dos conhecimentos obtidos, tendo-se instalado e utilizado uma rede P2P, baseada no protocolo BitTorrent.

Tal aplicação contou com a instalação de um Tracker BitTorrent (Ilustração 1), chamado XBTiT, utilizado para a transferência de arquivos em rede formada por computadores do laboratório. Nesse estudo, verificou-se o funcionamento de uma rede P2P e avaliou-se que tais redes apresentam vantagens em relação à arquitetura Cliente-Servidor. Configurou-se a rede de forma que esta pudesse funcionar seguindo o modelo de transferências padrão da arquitetura, voltado para redes em larga escala, mas modificando sua estrutura de forma que o modelo pudesse ser aplicado à uma rede local.

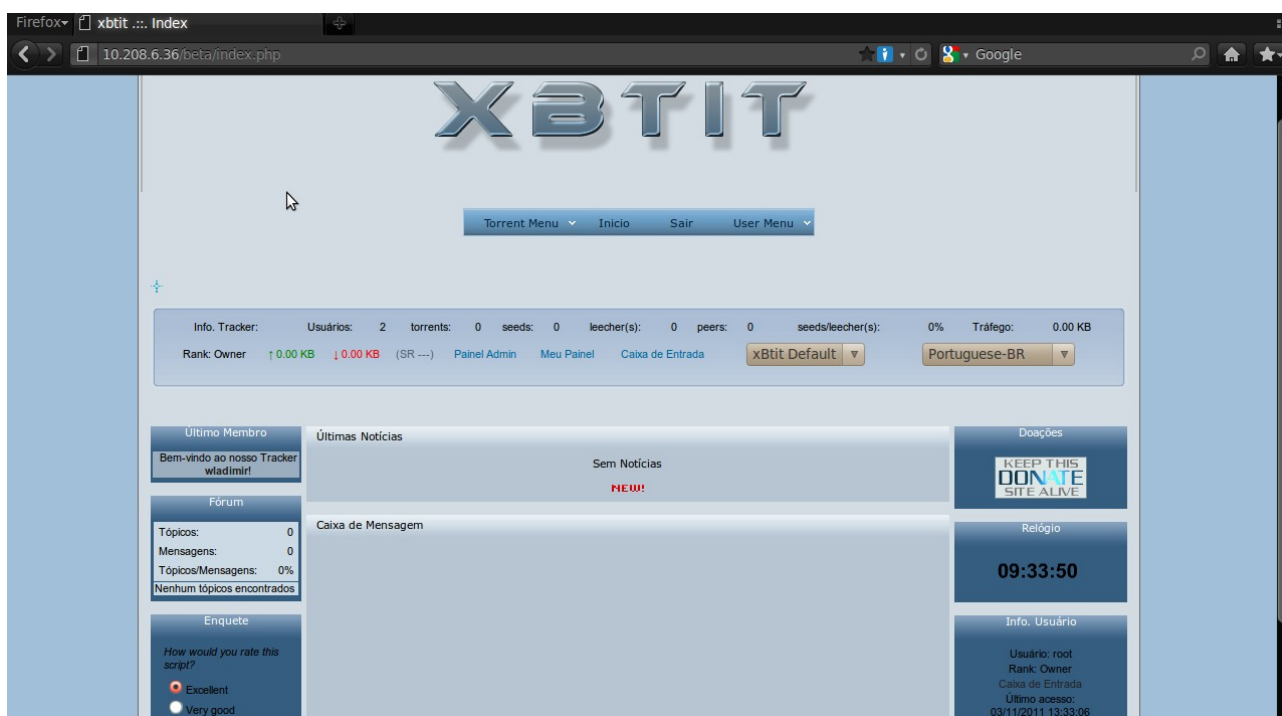


Ilustração 1: Tracker BitTorrent, XBTiT, instalado em um dos computadores do Laboratório.

Na ilustração 1, pode ser visto a estrutura do Tracker BitTorrent. Nesse caso específico, pode ser visto a tela principal do mesmo, onde constam os arquivos presentes no sistema, bem como informações pertinentes ao estado da rede, número de arquivos transferidos, dentre outras informações relacionadas a manutenção do sistema.

Com o Tracker BitTorrent instalado, foram realizados os testes de funcionamento. Tais testes contaram com análises de rede, mostrando se as conexões realizadas entre os computadores eram estáveis, bem como o teste de transferência de arquivos, mostrando como a transferência estava ocorrendo, se havia perda de pacotes ou atraso nas conexões (Ilustrações 2 e 3).

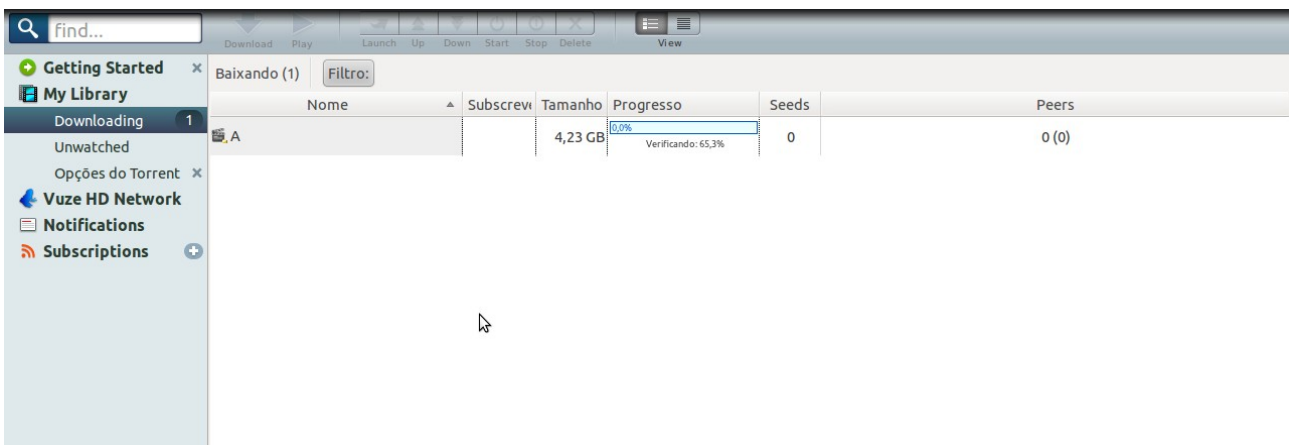


Ilustração 2: Teste de Transferência de Arquivo

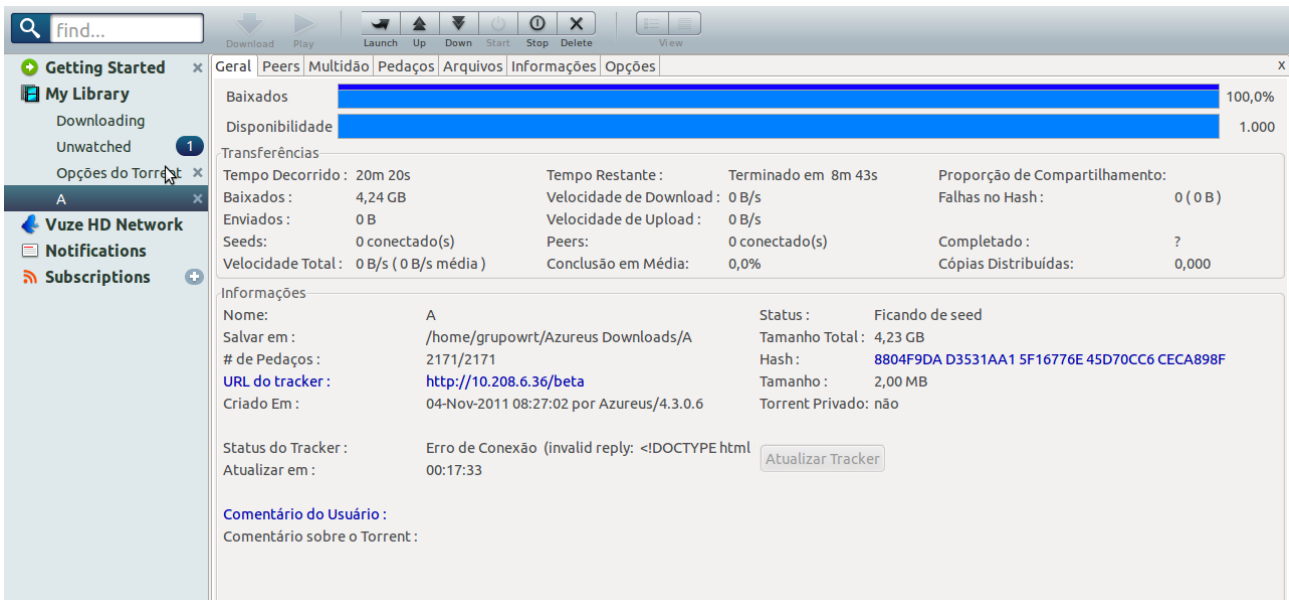


Ilustração 3: Detalhes da Transferência de Arquivos

Nas ilustrações 2 e 3, pode ser visto o processo de transferência de um arquivo, por intermédio do software Vuze, um cliente BitTorrent. Na primeira ilustração, está sendo realizada a transferência, entre dois nós na rede, de um arquivo de vídeo de 4,23GB. Tal transferência foi

completada em 8 minutos e 43 segundos, como pode ser visto na segunda ilustração, à uma taxa de transferência de aproximadamente 8,3MB por segundo.

No teste que envolve a análise de rede, obtivemos um resultado bastante favorável e condizente com nossas expectativas. Devido a estrutura da rede, foi possível realizar conexões tanto diretamente com os computadores conectados por cabos, em uma rede fechada, como realizar conexões sem perda através de rede sem fio.

No teste referente à transferência de arquivos, que pode ser visto anteriormente, o resultado foi igualmente favorável. Graças a topologia da rede (rede local e fechada), as transferências de arquivos ocorreram sem problemas entre os computadores, apresentando velocidade condizente com a esperada para uma rede P2P padrão, onde a velocidade flutua entre 7,5MB/s e 11MB/s.

Na segunda etapa do projeto, foi realizado um estudo tendo como base as soluções já existentes para o Problema do Bootstrap, mais especificamente as soluções voltadas a redes de pequeno porte, foco principal do projeto. A solução descrita no artigo de D. I. Wolinsky et al[2] foi estudada, uma vez que o projeto desenvolvido pelos autores tem o mesmo objetivo do presente projeto. Ou seja, trata-se de uma solução para o Problema do Bootstrap aplicável à redes de pequeno porte.

A solução em [2] apresenta uma forma de contornar o Problema do Bootstrap utilizando dois protocolos adaptados à necessidade da rede utilizada. Um dos protocolos, o XMPP, é utilizado por diversas empresas ao redor do mundo para resolver questões relacionadas à comunicação entre computadores. A principal vantagem que o XMPP traz à uma rede encontra-se nas diversas funções que ele oferece, destacando-se: transferência de mensagens, realização de chamadas de video/audio, serviços de colaboração e transferência de arquivos XML. A Google utiliza o XMPP como protocolo padrão de comunicação em seu sistema de mensagens instantâneas, o Google Talk.

O outro protocolo, Brunet, é utilizado também para resolver problemas de conexões entre computadores, entretanto, diferente do XMPP, este é conhecido e utilizado apenas no meio acadêmico. Como pode ser visto no artigo [9], o Brunet, quando aplicado à uma rede P2P, possibilita à essa rede desempenhar diversas funções que tornam o seu funcionamento mais completo em termos de conexão. Dentre essas funções, podem ser citadas: Serviço de reflexão, caracterizado pela capacidade de cada nó, ao realizar uma conexão, informar ao outro nó como este está visualizando a rede; Serviço de Rendezvous, caracterizado pela utilização de uma DHT (Distributed Hash Table) para armazenar informações pertinentes às requisições de cada nó e; Serviço de Relaying, caracterizado pelo suporte à roteamento recursivo e à utilização de um sistema de conexão indireto chamado Túnel[11].

Como resultado do estudo, foi decidido que, das duas soluções apresentadas, a que deveria ser utilizada no restante do projeto não poderia ser outra além da que se utiliza do protocolo Brunet, visto que esta é a mais acessível em termos de implementação e praticidade de

uso, além do fato de ser largamente difundida no meio acadêmico e de apresentar em sua estrutura os serviços necessários para resolução das três etapas do Problema do Bootstrap.

4.2 Implementação de uma solução para o Problema do Bootstrap:

Na terceira etapa do projeto, foi realizada uma pesquisa sobre aplicações BitTorrent que permitissem customização a fim de que uma solução para o Problema do Bootstrap, em rede de pequena escala, pudesse ser implementada e testada em tais aplicações. Ao final, o desempenho da aplicação já customizada deveria ser medido e comparado com a sua versão original.

Após a pesquisa, foi decidido que seria utilizado no projeto um software de nome Deluge[7], um cliente BitTorrent escrito em Python sob a licença GPL. Entretanto, o código do software deveria ser customizado para receber as funcionalidades presentes no protocolo Brunet.

Depois da escolha do programa a ser utilizado, o passo seguinte da etapa 3 foi estudar o protocolo Brunet. Esse estudo permitiu entender como o seu código, escrito em C#, poderia ser utilizado na customização do cliente BitTorrent Deluge. Além disso, permitiu entender de que forma a inclusão do protocolo Brunet afetaria as atividades de envio e recebimento de pacotes em uma rede P2P. Após o estudo do protocolo Brunet e de sua implementação, a parte prática da etapa 3 do projeto foi iniciada.

Devido à estrutura dos códigos tanto do cliente BitTorrent quanto do protocolo Brunet, tornou-se necessário encontrar uma forma de integrar suas funções. Com o uso de linguagens diferentes, o cliente BitTorrent escrito em Python e o protocolo Brunet escrito em C#, a realização de chamadas diretas, por parte do cliente às funções implementadas no protocolo, é tecnicamente inviável. Para resolver esse problema da conexão entre as linguagens, foi necessária a realização de um estudo da documentação da linguagem Python a fim de encontrar algum meio que permitisse que as chamadas realizadas pelo cliente ao protocolo pudessem ser realizadas.

Após a resolução do problema anterior, foi necessário resolver a questão relacionada ao fato de a linguagem C# não encontrar-se portada para os sistemas operacionais baseados em UNIX. Para solucionar esse problema, foi incorporado ao projeto um Framework chamado Mono[8], responsável por portar as funcionalidades do C# à qualquer outro sistema operacional.

4.3 Estudo de Caso:

Após um mês de testes sobre o cliente BitTorrent, não foi atingido um resultado satisfatório com relação a customização do mesmo. Com a customização, objetivou-se otimizar as funções do protocolo BitTorrent utilizando as ferramentas presentes no protocolo Brunet, que seriam, então, responsáveis por tratar o Problema do Bootstrap no estudo empreendido. Entretanto, mesmo utilizando as plataformas necessárias para a incorporação do código Brunet ao

cliente, o funcionamento não ocorria de acordo com as expectativas iniciais.

Mesmo com a customização do cliente BitTorrent contar com as funções necessárias para que a chamada do protocolo ocorresse, funções essas contidas na própria documentação da linguagem Python, o cliente não conseguia utilizar as funcionalidades presentes no protocolo e acabava por não conseguir localizar nós na rede para que pudesse realizar a troca de arquivos com os mesmos. Como a troca de arquivos não ocorria, os testes referentes a realização dos três passos necessários para resolver o Problema do Bootstrap (Reflexão, Relaying e Rendezvous) não poderiam ser realizados, o que comprometeu a conclusão do Estudo de Caso.

Face as dificuldades enfrentadas, um dos autores do artigo [2], David Wolinsky, foi consultado sobre as funcionalidades presentes no Brunet. Segundo David, o protocolo Brunet foi customizado de forma que somente as conexões entre nós de uma rede P2P fossem viabilizadas, implicando tão somente no atendimento das demandas por um procedimento de Reflexão, Relaying e Rendezvous aplicável ao contexto de redes P2P de pequena escala. Sendo assim, o protocolo Brunet garante apenas que a rede seja construída corretamente, necessitando ser customizado com as ferramentas de outro protocolo que possa garantir também o serviço transferência de dados.

Levando-se em consideração essa lacuna da customização do protocolo Brunet, o Estudo de Caso idealizado para o presente projeto ficou inviabilizado. Realizar uma adaptação do cliente BitTorrent, utilizando outro protocolo, para que a troca de arquivos pudesse ser realizada e aproveitando as ferramentas presentes no Brunet, demandaria um período de tempo maior que o previsto e conhecimento técnico que o autor do projeto não possui.

5. CONCLUSÕES

Com o termino da primeira fase do projeto, em que estudou-se a arquitetura P2P e o Problema do Bootstrap, chegou-se à um ponto onde o entendimento de ambos seria crucial para o bom andamento do projeto. O estudo da arquitetura P2P permitiu a obtenção de conhecimento relacionado ao funcionamento de uma rede P2P. Tal conhecimento permitiu entender o funcionamento de uma rede P2P e a necessidade de se abordar o Problema do Bootstrap.

Tendo realizado o estudo sobre a arquitetura P2P, iniciou-se o estudo do Problema propriamente dito, com o intuito de entender como este ocorre, quais as dificuldades envolvidas em sua resolução e de que formas ele pode ser contornado.

Com o prosseguimento da terceira etapa do projeto, caracterizada por ser a etapa onde a solução do Problema do Bootstrap escolhida seria recriada e aplicada, estudou-se como o protocolo deveria se comportar no escopo do projeto, bem como saber de que forma o cliente escolhido para ser customizado segundo as características do protocolo deveria ser adaptado.

Durante a adaptação foram encontrados alguns obstáculos como a não compatibilidade do protocolo Brunet com o sistema utilizado no projeto, bem como a dificuldade de se utilizar os codigos do cliente BitTorrent e do protocolo em uma mesma aplicação. Para contorná-los, iniciou-se um periodo de estudo sobre o cliente BitTorrent e o protocolo, levando a um subsequente contato com um dos proponentes da solução em estudo.

Os estudos realizados permitiram um entendimento mais aprofundado do funcionamento de ambos os componentes utilizados, mas ainda assim não resolveram os problemas de conexão e execução dos testes. Com as constantes falhas no funcionamento do cliente BitTorrent customizado, encontradas durante os testes de inicialização de uma rede P2P e/ou realização de transferências de arquivos, que comprovassem a resolução do Problema do Bootstrap, a realização do estudo de caso passou a ser questionada.

Segundo D.I. Wolinsky[12], o funcionamento do protocolo abrange realmente a solução para o Problema do Bootstrap mas não da forma como buscavamos aplicar. Diferente de nossa necessidade, o protocolo resumia-se à apenas realizar as conexões entre os nós de uma rede P2P. Para realizar o envio de arquivos seria necessária uma extensão do protocolo.

Com esse conhecimento em mãos, ficou decidido que o projeto deveria ser encerrado pois, embora ainda houvesse a possibilidade de alterar sua estrutura para que outra implementação pudesse ser feita, o tempo necessário para isso não caberia no total dado ao projeto.

6.REFERÊNCIAS BIBLIOGRÁFICAS

[1] Shirky, C. 2000. What is p2p... and what isn't.

<http://www.oreillynet.com/pub/a/p2p/2000/11/24/shirky1-whatisp2p.html>. O'Reilly.

[2] Wolinsky, D. I., St. Juste, P., Boykin, P. O., Figueiredo, R., “Addressing the Bootstrap Problem for Small Relay Networks” in IEEE P2P, 2010.

[3] GauthierDickey, C., Grothoff, C., “Bootstrapping of peer-to-peer network” in Symposium on Applications and the Internet – SAINT, 2008.

[4] Bram Cohen (10-Jan-2008). http://www.bittorrent.org/beps/bep_0003.html *BitTorrent.org*. Recuperado em 20 de Setembro de 2011

[5] Oscar Boykin. - <http://code.google.com/p/brunet/> - Recuperado em 09 de janeiro de 2012

[6] XMPP Standards Foundation - <http://xmpp.org/about-xmpp/history/> - Recuperado em 09 de Janeiro de 2012

[7] Deluge BitTorrent Client – <http://dev.deluge-torrent.org/wiki/About/> - Recuperado em 09 de Maio de 2012

[8] Mono Project – <http://www.mono-project.com/> - Recuperado em 09 de Maio de 2012

[9] Boykin, P.O., Bridgewater, J.S.A., Kong, J.S., Lozev, K.M., Rezaei, B.A. and Roychowdhury, V.P., "A Symphony Conducted by Brunet." ;In Proceedings of CoRR. 2007.

[10] H. Damfpling. 2003 - www.gnucleus.com/gwebcache/specs.html – Recuperado em 10 de Janeiro de 2012

[11] A, Ganguly, et al., "Improving peer connectivity in wide-area overlays of virtual workstations," em HPDC, 2008.

[12] David Isaac Wolinsky, comunicação privada, Março de 2012

