MARCO AURÉLIO LINS GOMES

P2PRIV-TV - MECANISMO DE PRIVACIDADE EM SISTEMAS IPTV BASEADOS EM REDES BITTORRENT

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Engenharia Elétrica.

São Paulo
2016

MARCO AURÉLIO LINS GOMES

P2PRIV-TV - MECANISMO DE PRIVACIDADE EM SISTEMAS IPTV BASEADOS EM REDES BITTORRENT

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Engenharia Elétrica.

Área de Concentração:

Engenharia da Computação

Orientador:

Prof. Dr. Marcos Antonio Simplicio Junior

São Paulo
2016

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, _____ de _____ de _____

Assinatura do autor: _____

Assinatura do orientador: _____

Catalogação-na-publicação

Gomes, Marco Aurelio Lins
P2PRIV-TV - MECANISMO DE PRIVACIDADE EM SISTEMAS IPTV BASEADOS EM REDES BITTORRENT / M. A. L. Gomes -- versão corr. -- São Paulo, 2016.
82 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Privacidade 2.Segurança 3.Sistemas IPTV 4.Protocolos P2P I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t.

AGRADECIMENTOS

Agradeço primeira e especialmente aos meus pais por todo apoio que me deram e esforço que fizeram em prol da minha formação e educação, sem eles com certeza não teria chegado até aqui.

Agradeço ao Prof. Dr. Marcos Antonio Simplicio Junior pela orientação, apoio e conselhos durante todo o desenvolvimento deste trabalho.

Agradeço aos amigos e familiares pelo apoio e compreensão durante o desenvolvimento deste trabalho.

Agradeço também aos colegas e amigos do LARC - Laboratório de Arquitetura e Redes de Computadores, em especial ao Leonardo, Bruno, Jonas, Ewerton e Cléber por todo o apoio, suporte e conselhos ao longo desse trabalho.

Agradeço à FAPESP pela bolsa concedida, sem a qual não teria conseguido desenvolver este trabalho.

RESUMO

Serviços para entrega de conteúdo multimídia tem se tornado cada vez mais comuns com o advento de conexões mais rápidas à Internet. Porém, esse crescimento na base de usuários consumidores desse tipo de serviços, levam a gargalo de desempenho devido à restrições de recursos de infraestrutura. Para tentar reduzir tais restrições, análises na forma como o conteúdo pode ser entregue foram realizadas e novos mecanismos de entrega de conteúdo surgiram. Um desses mecanismos envolve o uso de redes P2P - Par a Par, no qual o usuário é consumidor e provedor de conteúdo para seus pares. Sistemas IPTV utilizando protocolos P2P estão se tornando populares pelo mundo, devido à descentralização da infraestrutura necessária para entrega do conteúdo, e redução do gargalo de desempenho. Um dos protocolos de rede P2P mais comum é BitTorrent, que é utilizado amplamente pelo mundo, devido à seu método descentralizado de compartilhar os dados. Dos trabalhos desenvolvidos utilizando este método, houve uma necessidade de avaliar os requisitos de segurança para garantir o funcionamento do sistema sem comprometer a infraestrutura ou o usuário. Porém, existem problemas de segurança em aberto, e o objetivo deste trabalho é o de analisar a privacidade do usuário enquanto utiliza o sistema. Para tanto, este trabalho propõe o P2Priv-TV, um mecanismo que garante que o conteúdo que um determinado usuário irá assistir não será de conhecimento dos demais usuários da rede. Este mecanismo é avaliado por meio de emulação para determinar sua viabilidade e eficácia.

ABSTRACT

There are many video streaming services available to users due to broadband connections popularity. These kind of service has increasing numbers of users and that may lead to performance bottlenecks because of shortage of infrastructure resources. To avoid shortages, there are many studies to think in a new way to deliver content and this lead to new deliver methods. One of these methods is the use of P2P (peer to peer) Networks, where the user is content consumer and provider to other peers. BitTorrent is the most popular P2P protocol widely used in the world and there are IPTV systems using this protocol as content delivery mechanism. P2P IPTV systems were developed and security issues were analyzed to ensure that users and system do not been compromised. However, there are other issues, and the goal of this work is to analyze user privacy in the system. This work presents the P2Priv-TV, a mechanism that ensure a user can consume a content without another user in the system knows about it. This mechanism will be analyzed using emulation to analyze feasibility and effectiveness.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1 INTRODUÇÃO

Com a constante melhora na qualidade dos enlaces para conexão à Internet, se torna cada vez mais comum o surgimento de serviços para transmissão e consumo de conteúdo multimídia via rede. Dentre eles, merece especial atenção o serviço conhecido genericamente como IPTV (*Internet Protocol Television*, ou "TV sobre protocolo Internet") (XIAO et al., 2007a; ZEADALLY; MOUSTAFA; SIDDIQUI, 2011), que alia transmissão de conteúdo ao vivo ou sob demanda com capacidades como Internet, *e-mail*, voz sobre IP, entre outros. De fato, IPTV pode ser considerada um dos principais serviços da Internet do Futuro (TRONCO et al., 2010) devido ao seu potencial em revolucionar a indústria de entretenimento e fornecer aplicações personalizadas e com elevada qualidade aos seus espectadores.

Apesar do interesse em soluções de IPTV, elas apresentam um elevado consumo de recursos de rede e requisitos importantes de qualidade de serviço (e.g., baixo atraso), exigindo técnicas que garantam um bom desempenho do sistema mesmo quando se considera um elevado número de usuários simultâneos (HEI; LIU; ROSS, 2008). Neste contexto, o uso de uma arquitetura baseada no paradigma par a par (*peer-to-peer*, ou P2P) pode trazer benefícios interessantes aos provedores do serviço: a estrutura altamente distribuída de sistemas P2P, nos quais os nós participantes consomem e proveem recursos simultaneamente, torna essas redes escaláveis por natureza (LIU et al., 2008). Sua aplicação em serviços de IPTV pode, portanto, aliviar a carga nos servidores de vídeo e na rede como um todo, reduzindo custos com infraestrutura e manutenção da rede, aprimorando o desempenho do sistema e fazendo um uso mais racional dos re-

cursos disponíveis. Estratégias comuns de sistemas de IPTV-P2P para atingir esses objetivos envolvem o uso dos nós da rede para formar uma rede sobreposta (*overlay*) com suporte a *multicast* (LIAO et al., 2006), ou ainda em aproveitar a capacidade ociosa dos nós clientes (e.g., *desktops* ou *set-top-boxes*) para criar unidades de armazenamento temporário (*cache*) e de retransmissão de conteúdo a nós vizinhos. Algumas propostas mais recentes vão além, e consideram a possibilidade de usar as propriedades da rede P2P para prover serviços inovadores e a baixo custo, como exibição deslocada no tempo (*time-shifted TV*) (GALLO et al., 2009; LIU; SIMON, 2010), serviços de adaptação de vídeo na borda da rede (IQBAL; SHIRMOHAMMADI, 2009) e redução de consumo de energia na rede (PUSSEP et al., 2010). Ao mesmo tempo, esta abordagem pode ser tornada atrativa do ponto de vista dos consumidores de serviço: ao perceber que há capacidade ociosa nos nós clientes, os provedores podem utilizá-la e ofertar ao cliente alguma forma de redução de custos, ou possivelmente acesso a conteúdo exclusivo. Dessa forma, o uso mais racional dos recursos devido ao uso de sua capacidade ociosa seria compensado de modo semelhante ao que se propõe em outros tipos de sistemas modernos, como redes elétricas inteligentes (*smart grids*) (FANG et al., 2012). Nesses tipos de redes, a distribuição da energia elétrica entre a geração da energia e o consumidor é feita de forma mais racional, evitando perdas.

1.1 Motivação

O grande potencial da abordagem P2P em promover o uso mais racional dos recursos da rede tem atraído o interesse crescente tanto da academia como da indústria, de modo que o paradigma P2P vem competindo com a tradicional arquitetura cliente-servidor em diversas áreas de aplicação, incluindo IPTV. Este é o caso de sistemas comerciais de IPTV-P2P tais como CoolStreaming[1], PPLive[2], Anysee[3], SopCast[4] e Jo-

[1]http://www.coolstreaming.us
[2]http://www.synacast.com/en/
[3]http://www.anysee.com
[4]http://www.sopcast.org

ost[5] para citar alguns sistemas populares (listas mais extensas podem ser encontradas em (LIU; GUO; LIANG, 2008; XIE et al., 2008; GU et al., 2014)).

Apesar do grande potencial de aplicação de sistemas P2P no contexto de IPTV, um problema com as soluções existentes é que, em sua maioria, não abordam potenciais problemas de segurança resultantes do armazenamento de dados nos equipamentos dos usuários finais ou da comunicação direta entre eles. Isso é preocupante porque a ausência de uma entidade confiável central intermediando as transações efetuadas em sistemas P2P costuma levar a diversas vulnerabilidades de segurança (BARCELLOS; GASPARY, 2006; YU; BUFORD; MERABTI, 2007; GOTTRON; KÖNIG; STEINMETZ, 2010) Assim, sem a implementação dos devidos mecanismos de segurança, pode-se ter diversas vulnerabilidades, como:

- Usuários maliciosos podem acessar o conteúdo armazenado em seus dispositivos e substituí-los, inserindo dados falsos na rede (ataque conhecido como "poluição de conteúdo") (CHRISTIN; WEIGEND; CHUANG, 2005; YANG et al., 2008).

- Em situações em que o usuário final venha a atuar como uma unidade de cache, armazenando conteúdo localmente sem tê-lo adquirido e mesmo sem ter direito de acessá-lo, a ausência de mecanismos de controle de acesso aos dados armazenados pode afetar negativamente os negócios do provedor de conteúdo, pois teremos um usuário acessando um conteúdo sem consentimento do provedor de acesso.

- Sistemas de IPTV-P2P costumam envolver troca de informações entre usuários finais para que seja feita a descoberta do local onde certo conteúdo encontra-se armazenado. Assim, dependendo da forma como é realizado o processo de descoberta, um usuário malicioso pode informar incorretamente a presença de diversos conteúdos populares em certo alvo na tentativa de direcionar diversos

[5] http://www.joost.com

pedidos a esse usuário. Como resultado, a rede da vítima pode ficar congestionada, causando-se, assim, um ataque de negação de serviço distribuído (*Distributed Denial of Service*, ou DDoS) (HARRINGTON; KUWANOE; ZOU, 2007; DAVIS et al., 2008; SUN; TORRES; RAO, 2010).

- Finalmente, o próprio fato dos usuários se comunicarem diretamente pode levar ao indesejável cenário em que eles descobrem o conteúdo acessado uns pelos outros, permitindo-se que qualquer usuário trace o perfil dos clientes da rede, violando sua privacidade (GHEORGHE; CIGNO; MONTRESOR, 2011).

Tais observações indicam a necessidade de se utilizar soluções de segurança que permitam a detecção e prevenção de atividades maliciosas que se aproveitam de vulnerabilidades existentes em sistemas de IPTV baseadas em redes P2P, garantindo que a credibilidade do serviço não seja afetada por usuários mal-intencionados. Uma análise da literatura revela que de fato existe atualmente um número razoavelmente elevado de soluções para prevenir diversos desses ataques, dentre eles o de poluição de dados (CHRISTIN; WEIGEND; CHUANG, 2005; YANG et al., 2008; VIEIRA et al., 2013) e DoS (WU et al., 2011; SU et al., 2012). Por outro lado, foram identificados poucos trabalhos que se concentram no último dos problemas levantados, o de privacidade dos usuários (ISDAL et al., 2010). Mesmo o trabalho identificado como privacidade do usuário, considera que a privacidade está na processo do usuário decidir qual conteúdo será compartilhado para determinados usuários. Neste contexto, o usuário já sabe à priori para quem está enviando o conteúdo, diferentemente do conceito onde origem e destino se desconhecem. Além disso, em sua maioria tais trabalhos são consideravelmente genéricos, não explorando as características intrínsecas a sistemas de IPTV, em especial (1) requisitos de tempo real e (2) capacidade de tirar proveito de tais sistemas envolverem redes privadas, cujos componentes são controlados pelo provedor de conteúdo.

1.2 Objetivo e Justificativa

O presente trabalho tem como objetivo propor uma solução para o problema de ataques contra a *privacidade* de usuários de sistemas IPTV baseados em tecnologias P2P. Especificamente, a solução proposta previne que sejam reveladas a entidades não autorizadas informações sobre qual conteúdo foi acessado por cada usuário do sistema de IPTV. Na prática, apenas o provedor do conteúdo e o próprio usuário que solicita o conteúdo são consideradas entidades autorizadas, sendo os demais usuários do sistema classificados como não autorizados.

Como tecnologia alvo, é dada atenção especial a redes IPTV baseados no popular protocolo BitTorrent (COHEN, 2008) para a troca de conteúdo entre usuários. O interesse neste protocolo específico deve-se à sua utilização com sucesso em sistemas de IPTV existentes na literatura (GALLO et al., 2009; LIU; SIMON, 2010; MIERS, 2012) e também ao amplo suporte do protocolo em termos de documentação e ferramentas de simulação (EVANGELISTA et al., 2011; YANG; ABU-GHAZALEH, 2005; KATSAROS et al., 2009).

1.3 Método

O método adotado no desenvolvimento deste trabalho é a pesquisa aplicada com base na abordagem hipotética-dedutiva, utilizando referências disponíveis na literatura especializada para a definição do problema, seguida da especificação de hipóteses e da análise das mesmas. A proposta de solução é então especificada e implementada na forma de um protótipo para permitir a validação das hipóteses que serviram como base para o projeto da solução.

Como resultado da aplicação desse método, em um primeiro instante o trabalho concentrou-se no estudo do funcionamento de sistemas de fornecimento de conteúdo através de mecanismos P2P, identificando o fluxo envolvido em tais processos, desde

sua requisição ao provedor do serviço, até a entrega do conteúdo ao usuário final.

Em seguida, por meio de análise da literatura especializada, teve-se como foco a identificação das principais vulnerabilidades presentes em sistemas de IPTV-P2P, bem como os métodos que poderiam ser utilizados para saná-las de forma eficaz e eficiente, de modo a não impactar significativamente na qualidade de serviço provida aos usuários finais do sistema. Por meio desse levantamento, foi possível identificar que o problema da privacidade nesses sistemas carece de soluções especializadas, o que levou o presente trabalho a apresentar tal vulnerabilidade como alvo específico.

Definido tal contexto, partiu-se para o estudo de soluções de privacidade em geral, em busca de mecanismos que pudessem ser utilizados como base para criar uma solução voltada ao cenário de serviços de IPTV baseados em P2P. Especificada tal solução, denominada P2Priv-TV, passou-se à construção de um protótipo que permitisse a emulação do sistema e, assim, sua validação em termos de efetividade e desempenho.

1.4 Organização

Este documento é dividido em 6 capítulos.

- O Capítulo 2 apresenta a fundamentação teórica do trabalho, discutindo os principais conceitos e tecnologias envolvidas neste trabalho;
- O Capítulo 3 discute as soluções de segurança que são pertinentes a um sistema IPTV baseado em BitTorrent, assim como soluções de segurança para redes BitTorrent;
- O Capítulo 4 apresenta o P2Priv-TV, mecanismo proposto para fornecer privacidade aos usuários que utilizam sistemas de IPTV baseado em redes BitTorrent;
- O Capítulo 5 analisa em detalhes os cenários que foram montados para avaliar a solução proposta, bem como métricas para analisar os resultados obtidos, as-

sim como discute os resultados experimentais obtidos, considerando diferentes cenários; e

- Finalmente, o Capítulo 6 apresenta as considerações finais sobre este trabalho e temas que podem ser abordados em trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os principais conceitos e tecnologias envolvidos no contexto de sistemas de IPTV baseados em P2P, servindo como base para o entendimento dos conceitos abordados no presente documento. Os conceitos apresentados são: redes BitTorrents, que são um tipo de mecanismo P2P e é utilizado na solução proposta; tópicos de gerenciamento de redes; tradução de endereçamento de redes (NAT); e sistemas IPTV. Essa abordagem tem o objetivo de entender melhor os conceitos utilizados na solução proposta que é apresentada no capítulo 4.

2.1 Redes BitTorrent

O BitTorrent (COHEN, 2008) é um protocolo para distribuição de arquivos, com uma arquitetura na qual a entrega de dados é feita através de comunicação ponto a ponto entre os clientes que desejam receber certo conteúdo. Por ser um protocolo ainda em desenvolvimento, algumas implementações apresentam ligeiras diferenças de outras. Para manter a discussão concisa e coerente, este trabalho considera a implementação utilizada em (EVANGELISTA et al., 2011), aqui escolhida pelo fato do autor do trabalho mencionado ter tido contato direto com os desenvolvedores de aplicações que utilizam o BitTorrent como mecanismo para troca de dados para esclarecer eventuais pontos da especificação do protocolo que geram incompatibilidades entre as aplicações. Portanto, espera-se que tal implementação seja fortemente aderente à especificação do protocolo, o qual é descrito na seção 2.1.1.

2.1.1 Estrutura de dados

Para a distribuição, a fonte do conteúdo cria um arquivo de metadados, denominado arquivo torrent, contendo as seguintes informações (COHEN, 2008):

- URL de um ou mais servidores, denominados *trackers*, que agregam informações sobre os clientes que possuem o conteúdo;
- Informações sobre o conteúdo em si, tais como:
  - Nome do(s) arquivo(s) alvo;
  - O tamanho de cada pedaço do arquivo (unidade base para a troca de dados entre os pares);
  - Número de pedaços no qual o conteúdo foi dividido; e
  - Tamanho total do arquivo.

Tais informações são importantes para identificar o conteúdo e também para estimar o tempo total necessário para obter cada pedaço, calculado com base no tamanho do pedaço e na largura de banda disponível para o cliente descarregar o conteúdo. A variação destas informações pode gerar moficações na forma como o conteúdo é transferido entre os pares. A forma como o conteúdo é transferido será apresentada na seção 2.1.2.

2.1.2 Operação

De acordo com a especificação oficial do BitTorrent (COHEN, 2008), após a criação de cada arquivo de torrent, o arquivo normalmente é enviado para um repositório de armazenamento (tipicamente um servidor web). Esse servidor disponibiliza o arquivo torrent para usuários que desejam ter acesso ao conteúdo, podendo ser facilmente encontrado por meio de ferramentas de busca tradicionais da web. O usuário, ao

descarregar o torrent em seu computador, executa então uma aplicação cliente do BitTorrent que lê o arquivo e, de posse das informações do *tracker*, se comunica com este último para obter informações sobre quais outros clientes possuem o conteúdo completo (denominados *seeders*, ou semeadores) ou apenas uma parte dos seus pedaços (denominados *leechers*, ou sanguessugas).

De posse da lista de usuários com os quais o conteúdo pode ser obtido, a aplicação cliente se comunica diretamente com um subconjunto deles. Cada cliente contactado informa a todos os pares comunicantes quais pedaços do conteúdo ele possui. Em seguida, o cliente interessado em realizar a descarga do conteúdo se conecta aos pares que possuem os pedaços que lhe interessam e inicia a solicitação de tais pedaços. Após a obtenção de cada pedaço, o cliente verifica a integridade dos dados, calculando o *hash* criptográfico (usando SHA-1 (COHEN, 2008)) sobre os dados recebidos e comparando o valor obtido com aquele informado no arquivo de torrent. Dados que não passem nesse teste são descartados; já no caso da verificação ter sucesso, o cliente anuncia aos seus pares a posse daquele pedaço, de modo que outros pares interessados possam obtê-lo. Além da comunicação com os pares, o cliente também se comunica periodicamente com o *tracker*, informando-lhe sobre quanto do conteúdo já foi descarregado, o quanto ainda falta para terminar a descarga e o quanto do conteúdo foi enviado para os demais clientes. Esse ciclo de troca de dados entre o cliente e demais pares da rede se mantém até que a aplicação cliente consiga a totalidade do conteúdo, ou pare a descarga do conteúdo antes de atingir sua totalidade (e.g., por ação do usuário). Ambas as situações de parada são informadas ao *tracker*.

Como o BitTorrent oferece um mecanismo para troca de conteúdo entre usuários da rede, é necessário observar que os usuários têm um tempo variável de permanência na rede. Com isso, existem momentos em que diversos usuários podem prover boa parte dos pedaços do conteúdo que possuem para os demais, assim como há casos em que a disponibilidade de alguns pedaços é baixa. Para tentar evitar esta última situação,

o protocolo implementa um mecanismo chamado *rarest-first* ("mais raros primeiro"), por meio do qual a parte do conteúdo com menor disseminação entre os usuários é preferida ao realizar uma transferência. Esse mecanismo pode, então, garantir uma presença razoavelmente homogênea das partes do conteúdo na rede, facilitando a tarefa de obter acesso a todas elas pelos usuários.

Uma das principais características do protocolo BitTorrent em comparação com protocolos para troca de arquivos centralizados como o FTP (*File Transfer Protocol*, ou Protocolo de Transferência de Arquivos) é que a distribuição de arquivos ocorre de forma distribuída. Assim, evitam-se sobrecargas em um único servidor, aumentando-se a resiliência a falhas do sistema como um todo devido à maior disponibilidade do conteúdo para os usuários. Já quando se compara o BitTorrent com outros mecanismos P2P, a característica chave que diferencia tal protocolo é o mecanismo de prevenção de *free-riders* no sistema (i.e., usuários com comportamento egoísta, que compartilham pouco com outros usuários) (CEBALLOS; GORRICHO, 2006). Isto é feito por meio do mecanismo conhecido como *tit-for-tat* ou "olho-por-olho", pelo qual cada usuário dá preferência a fornecer pedaços a usuários que lhe estejam enviando outros pedaços do conteúdo. Dessa forma, pode-se obter uma maior disponibilidade do conteúdo para os usuários do sistema, pois a colaboração ampla é do interesse de todos.

2.1.3 Uso de BitTorrent em sistemas de IPTV

Existem diversos estudos (GALLO et al., 2009; MIERS, 2012) que utilizam o BitTorrent como base para a construção de sistemas de IPTV-P2P, adaptando-o para atender os requisitos de entrega de conteúdo multimídia ao usuário. Em especial, uma modificação comum na literatura refere-se ao próprio mecanismo de *rarest first*, que deve ser adaptado para dar preferência a pedaços a serem reproduzidos em um futuro próximo, mesmo que não sejam necessariamente os mais raros dentre todos os pedaços existentes. Além disso, como o tamanho dos pedaços pode ser selecionado pelo criador do

torrent, soluções de IPTV costumam adotar valores pequenos (e.g., da ordem de 64 KB (MARFIA et al., 2011)), permitindo que cada um deles seja obtido em um curto intervalo de tempo. Ambas as estratégias têm como objetivo garantir que o *buffer* do aplicativo de reprodução de dados multimídia não se esvazie, de forma que que o conteúdo possa ser reproduzido pelo usuário sem interrupções.

Na seção 2.2 é abordado conceitos de gerenciamento de redes. Estes conceitos são utilizados para que seja possível para o provedor monitorar o funcionamento dos elementos que compõem o sistema.

2.2 Redes Gerenciadas

Uma rede gerenciada, segundo (KLERER, 1988), é uma rede dotada de um conjunto de facilidades para controlar, coordenar e monitorar os recursos utilizados durante as comunicações. Desta forma, os equipamentos da redes podem ser monitorados e configurados para um melhor desempenho durante sua operação.

2.2.1 Abordagens de gerenciamento

O conteúdo do que é normalmente gerenciado em equipamentos de rede, segundo (STANDARDIZATION, 1998), pode ser dividido em áreas funcionais, tais como:

- Gerenciamento de Falhas: os objetivos são a detecção e o relato de falhas, armazenamento de *logs* de eventos para posterior análise, execução de testes de diagnósticos, isolamento e correção de falhas e prevenção de falhas através de monitoramento de taxas de erro;

- Gerenciamento de Desempenho: cujos objetivos são o controle de operação diária da rede, através da análise de elementos estratégicos da mesma (e.g., taxa de transmissão, estado de equipamentos e serviços ativos, vazão e tempo de reposta), identificação de pontos de gargalo e situações que ocasionem problemas

no bom funcionamento da rede, registro de informações para análise posterior, seleção e uso de indicadores para medir desempenho da rede, coleta de informações estatísticas, determinação do desempenho da rede sob condições normais e artificiais de funcionamento e alteração no modo de operação do sistema com o objetivo de melhorar o desempenho da rede;

- Gerenciamento de Contabilização: tem como objetivo controlar a utilização dos recursos da rede, especificando limites e estabelecendo cotas para um determinado usuário ou grupo de usuários, coletar os dados de utilização e realizar a precificação dos recursos utilizados, determinar custos envolvidos na utilização de recursos e emitir relatórios sobre tal utilização e sobre os custos correspondentes;

- Gerenciamento de Configuração: os objetivos são gerenciar o ciclo de vida dos equipamentos e a sua configuração associada, identificar os componentes de hardware e software do sistema para o devido controle, monitorar os componentes do sistema para assegurar conformidade com os requisitos estabelecidos, manter um registro do estadi de cada componente da rede, permitir que a configuração seja flexível para atender demandas especiais, estabelecer os parâmetros de operação da rede, coletar informações e realizar a alteração da configuração atual da rede, além de armazenar as informações relativas à configuração para efeitos de relatório; e

- Gerenciamento de Segurança: tem como objetivos gerenciar os mecanismos e procedimentos que proporcionam proteção aos recursos da rede, armazenar e manipular registros de segurança, garantir a manutenção da política de segurança estabelecida, garantir a proteção dos recursos computacionais e de rede contra vulnerabilidaeds ou violações, monitorar a utilização dos recursos e atividade dos usuários, gerenciar os serviços de segurança, armazenar e tratar *logs* de segurança, emitir relatórios de eventos de segurança e reagir a eventos de ataque.

A solução proposta e apresentada no capítulo 4 faz uso da área de gerenciamento de desempenho. Nesta área, o monitoramento dos elementos de rede são realizados para verificar se há oportunidades de otimização dos recursos presentes na rede, se há elementos que estão em estado de sub-utilização, ou até mesmo verificar se os elementos presentes na rede estão funcionando de acordo limites operacionais pré estabelecidos.

2.2.2 Arquiteturas de gerenciamento e o SNMP

Diversas arquiteturas foram criadas para atender às necessidades de gerenciamento de dispositivos via rede. Dentre eles, podem ser citadas:

- Arquitetura OSI (*Open System Interconnection*) (INFORMATION..., 1994);
- Arquitetura SNMP (*Simple Network Management Protocol*) (CASE et al., 1990);
- Arquitetura TMN (*Telecommunication Management Network*) (INTERNATIONAL TELECOMMUNICATION UNION, 2000);
- Arquitetura WBEM (*Web Based Enterprise Management*) [1];
- Arquitetura IPMI (*Intelligent Platform Management Interface*) (WHITE, 2002);
- Arquitetura de gerenciamento via WEB;
- Arquitetura utilizando CORBA (*Common Object Request Broker Architecture*) [2]; e
- Arquiteturas utilizando conceitos de *web services* e SOA (*Service Oriented Architecture*).

[1] http://www.dmtf.org/standards/wbem
[2] http://www.omg.org/spec/CORBA/3.3

As cinco primeiras arquiteturas citadas foram criadas especificamente para gerenciamento de redes, enquanto as demais se aproveitam de uma determinada tecnologia para criar uma arquitetura de gerenciamento de redes. Por exemplo, para configuração de elementos de rede, é mais comum o uso de uma arquitetura de gerenciamento via WEB, via IPMI, ou ainda através do acesso direto ao equipamento. Dentre as arquiteturas mencionadas, aquela implementada em equipamentos de rede com a finalidade de gerenciamento é a arquitetura SNMP (CASE et al., 1990), usada especialmente para monitorar condições da rede, mas também útil para configurar elementos de interconexão.

A arquitetura SNMP consiste basicamente de uma estrutura hierárquica: um gerente controla as ações e coleta as informações dos demais elementos da rede, denominados agentes, que ficam aguardando requisições do gerente. A comunicação entre o agente e o gerente é normalmente feita utilizando UDP sobre o IP, embora tenha sido adicionado suporte ao TCP a partir da SNMPv2 (SCHOENWAELDER, 2002). O SNMP não limita as informações que podem ser requisitadas a um conjunto pré-definido, mas permite aos arquitetos de sistemas definirem diversos tipos de objetos que podem ser manipulados (i.e., lidos ou escritos), os quais são conhecidos em conjunto como MIBs (*Management Information Bases*). Assim, após receber a requisição, cada agente verifica qual operação foi solicitada e sobre qual MIB essa operação deve ser executada, realizando tal ação e eventualmente enviando como resposta ao gerente o resultado obtido.

Existe ainda um modo especial de operação do agente, que consiste em enviar notificações aos gerentes sem que o mesmo tenha feito a requisição. Esse processo de envio de notificações, conhecidas como *traps*, é aplicado a eventos significativos de determinados objetos, conforme previamente estabelecido pelo administrador da rede.

No contexto do presente trabalho, o uso de redes gerenciadas se faz necessário pois a solução adotada acessa equipamentos de rede e realiza as configurações necessárias

para alterar o comportamento dos mesmos. Assim, assume-se que o provedor tem o devido controle sobre os elementos da rede para prover garantias de qualidade de serviço e também de manipular os elementos que compõem a rede para prover serviços de segurança. Além disso, o protocolo SNMP é utilizado neste trabalho para monitorar os elementos do sistema, permitindo-se avaliar o impacto da solução de segurança proposta sobre o desempenho da rede.

2.3 Privacidade de usuários na rede: Tradução de Endereços (NAT) e TOR

Todo dispositivo que acessa a Internet possui um identificador, que é o seu endereço IP. Essa identificação serve para que os dispositivos consigam se localizar e possam transportar pacotes de um ponto a outro. Porém, o fato de cada endereço IP (em sua versão 4 (DEFENSE ADVANCED RESEARCH PROJECTS AGENCY, 1981)) ser representado por apenas 32 bits permite que um número limitado de dispositivos consigam acessar a Internet: o limite teórico é de 2^{32} (cerca de 4 bilhões) endereços possíveis, embora na prática esse número seja menor devido à existência de endereços especiais e reservados. Assim, surgiu a necessidade de evitar o esgotamento de endereços IPv4 enquanto um novo protocolo com endereço mais longo (neste caso, o IP versão 6, que conta com endereços de 128 bits (DEERING; HINDEN, 1998)) não era desenvolvido. Para atacar esse problema, uma das soluções desenvolvidas pela IETF (*Internet Engineering Task Force*, ou "Força Tarefa de Engenharia da Internet")[3] é o protocolo de Tradução de Endereçamento de Rede (*Network Address Translation* – NAT) (SRISURESH; EGEVANG, 2001).

O NAT é uma forma de permitir que máquinas recebam endereços privados, válidos localmente em suas próprias sub-redes mas não na Internet, possam ainda sim ter seus pacotes roteados para outras redes por meio da Internet. De acordo com (SRI-

[3] www.ietf.org

SURESH; EGEVANG, 2001), isto é feito com a ajuda de um *gateway*, uma máquina da sub-rede que possui acesso à Internet e que gerencia o roteamento do tráfego das máquinas que compõem a sub-rede. Especificamente, essa entidade é responsável por realizar a tradução dos endereços de origem/destino de cada pacote enviado/recebido por uma máquina interna, de modo que tanto a Internet como a rede local enxerguem apenas pacotes com endereços válidos. Assim, o processo realizado em uma rede que utiliza NAT é:

1. A máquina dentro da sub-rede envia uma requisição para um endereço externo à rede local;
2. Essa requisição é roteada normalmente, passando pelo *gateway* da rede;
3. O *gateway* verifica que o endereço IP de destino não se encontra dentro da sub-rede onde está a máquina;
4. O *gateway* então modifica o cabeçalho do pacote, indicando que o IP de origem da requisição é um IP válido na Internet e sob sua responsabilidade;
5. O pacote, depois de modificado, é então enviado pela interface do *gateway* com a Internet para que chegue ao destino;
6. O destino recebe o pacote e, ao processar a requisição, envia a resposta para o endereço de origem nele indicado, ou seja, o endereço especificado pelo *gateway*;
7. O *gateway* recebe o pacote de reposta e confere o endereço para o qual a resposta foi direcionada;
8. Novamente, o *gateway* altera o cabeçalho do pacote de resposta para que o destino aponte para o IP da máquina na sub-rede que é de fato quem fez a requisição;
9. O *gateway* então envia o pacote para a máquina na sub-rede; e

10. A máquina na sub-rede recebe a resposta para sua requisição.

No caso do NAT tradicional (também conhecido como NAT 1:1 (SRISURESH; EGEVANG, 2001)), a alteração no cabeçalho ocorre de forma que o IP privado é substituído por um IP público, da lista de IPs públicos que o *gateway* tem disponível para este tipo de operação, conforme mostra a Figura 1. Dessa forma, a máquina com endereço privado acessa a Internet como se fosse uma máquina com um IP público, tendo suas requisições respondidas normalmente após o estabelecimento de uma sessão com o IP de seu interlocutor. Pode-se inclusive prevenir que uma máquina externa qualquer acesse a máquina interna usando tal IP público, permitindo-se apenas o estabelecimento de sessões de dentro para fora da rede.

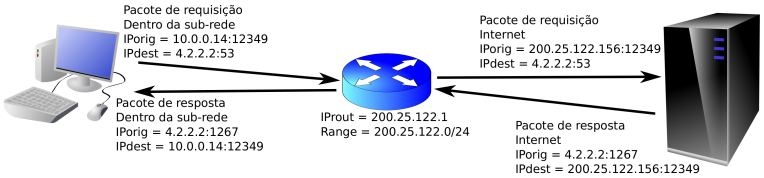

Figura 1: Alteração do cabeçalho de um pacote IP em uma rede configurada com NAT tradicional.

Uma alternativa ao NAT tradicional é o NAPT (*Network Address-Port Translation*, Tradução de Endereço-porta de Redes) (SRISURESH; EGEVANG, 2001), no qual o *gateway* da rede tem apenas o seu próprio IP externo para acesso à Internet ao invés de um conjunto de IPs válidos. Nesse caso, o *gateway* substitui não apenas o endereço de origem do pacote pelo seu próprio endereço IP, mas também a porta de origem por uma de suas portas locais disponíveis. Dessa forma, o destino do pacote entende que a requisição foi feita pelo próprio *gateway*, enviando os pacotes de resposta a ele. Ao receber tal pacote de resposta, o *gateway* faz a tradução inversa, substituindo o IP/porta de destino do pacote pelo endereço IP/porta da máquina interna que de fato enviou a requisição. A Figura 2 ilustra como ocorre a alteração do cabeçalho no caso do NAPT.

Neste trabalho, o NAPT é de especial interesse porque, embora seu objetivo seja prover a conectividade entre dispositivos, ele tem como efeito secundário o mascara-

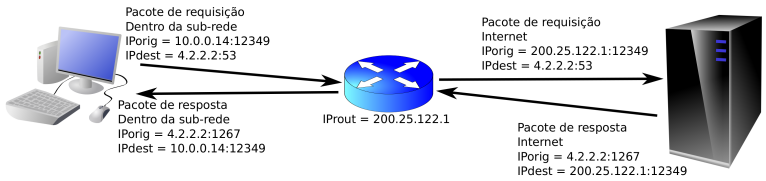

Figura 2: Alteração do cabeçalho de um pacote IP em uma rede configurada com NAPT.

mento do endereço real dos elementos da rede, garantindo maior privacidade na comunicação entre os dispositivos. Cabe notar, entretanto, esta não é a única forma de obter tal serviço de privacidade, embora seja uma das mais eficientes. Por exemplo, o protocolo TOR (DINGLEDINE; MATHEWSON; SYVERSON, 2004a) é um protocolo de privacidade que usa o mascaramento de IPs para ocultar do destino quem é que originou a solicitação. Uma rede que utiliza o protocolo TOR é composta de vários *gateways*, que colaboram para (1) re-roteador os pacotes entre eles, mascarando a cada salto o endereço IP do salto anterior ao substituí-lo com seu próprio endereço, e (2) realizando a (de)cifração dos pacotes recebidos com o objetivo de prover confidencialidade aos dados durante todo o seu trajeto. Assim, o destino enxerga apenas o último *gateway* do caminho, já que é dele o endereço no pacote, enquanto cada salto enxerga apenas os endereços do salto anterior (origem do pacote) e posterior (destino para o qual o pacote deve ser encaminhado). Adicionalmente, usando a extensão do protocolo conhecida como "Serviços Ocultos" [4], pode-se também prevenir que a origem determine o endereço IP do destino. Isso torna o protocolo TOR bastante efetivo para proteger a privacidade dos usuários até mesmo com relação aos roteadores da rede, e não apenas com relação aos pares comunicantes. Por outro lado, devido aos diversos redirecionamentos entre *gatways* e à cifração adicional realizada sobre os dados (BLOND et al., 2013), essa proteção adicional também leva a um custo computacional consideravelmente superior a um simples NAT. Na seção 2.4, é apresentado os conceitos de sistemas IPTV.

[4] https://www.torproject.org/docs/hidden-services.html.en

2.4 IPTV

IPTV, acrônimo para *Internet Protocol Television* ("TV sobre protocolo Internet"), é a tecnologia que provê a transmissão de conteúdo multimídia utilizando a Internet (XIAO et al., 2007b). A análise da literatura revela ao menos três formas razoavelmente distintas de fornecimento de conteúdo multimídia em sistemas de IPTV (GALLO et al., 2009):

- Transmissão tradicional em tempo real: a apresentação do conteúdo é fixa, de modo que o usuário pode reproduzir o conteúdo apenas da forma como ele está sendo transmitido, sem possibilidade de avançar ou retroceder no seu tempo de apresentação (MOL et al., 2009). Se encaixa neste exemplo a maioria das transmissões de eventos esportivos e shows ao vivo pela Internet;

- Video sob demanda: é possível iniciar o vídeo imediatamente após sua requisição. Este caso engloba a maioria dos serviços de locação de vídeo pela Internet, como disponibilizado pelo Netflix[5], Hulu[6] e serviços de streaming de vídeo da Amazon[7];

- Apresentação Deslocada no Tempo ("Time-shifted TV"): uma forma de transmissão na qual é possível retroceder ou avançar na exibição do conteúdo conforme desejado (GALLO et al., 2009; LIU; SIMON, 2010). Essa modalidade de transmissão é embutida nas características das modalidades anteriores, estendendo suas funcionalidades.

Qualquer que seja a modalidade de entrega do conteúdo, isto tradicionalmente é feito por um servidor (ou grupo de servidores) que provê o serviço unidirecionalmente

[5] www.netflix.com
[6] www.hulu.com
[7] www.amazon.com/Instant-Video/b?node=2858778011

para cada usuário, conforme ilustrado na Figura 3. Desta forma, serviços particularmente populares tendem a receber requisições de cada vez mais usuários.

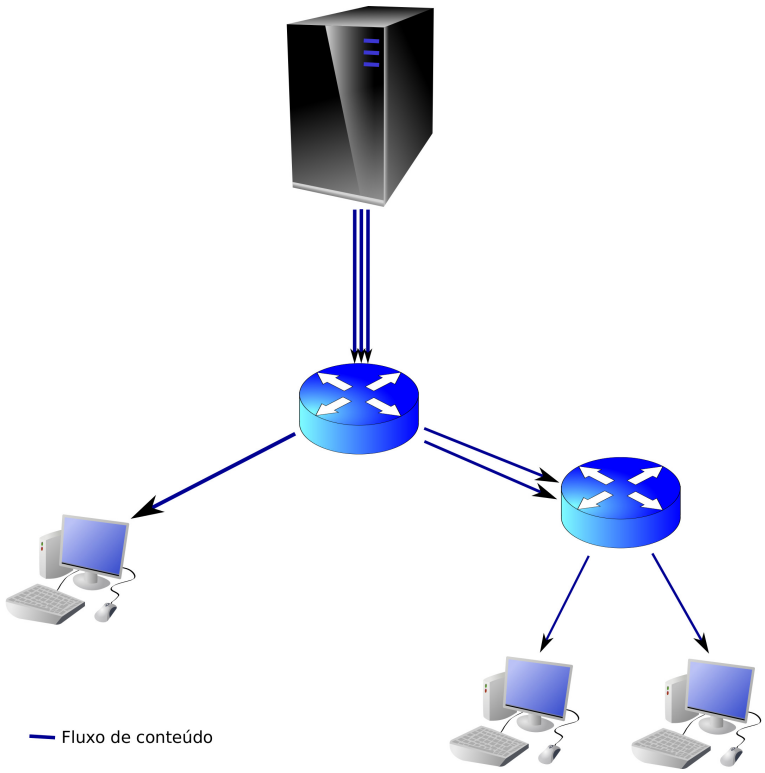

Figura 3: Estrutura de uma rede com solicitações *unicast*.

Potencialmente, isto acaba causando sobrecarga no servidor que provê o conteúdo, consequentemente acarretando na perda da qualidade da experiência para os usuários (HASSAN; NENG; SUAN, 2012; SEYYEDI; AKBARI, 2011).

Para aliviar a carga nos servidores, uma estratégia que utiliza conexões *multicast* ao invés de *unicast* permite atender a múltiplos usuários com um mesmo fluxo multimídia, conforme ilustrado na Figura 4. Apesar desse mecanismo atender bem os casos de transmissão ao vivo de conteúdo, no caso de conteúdo sob demanda a eficiência de tal estratégia depende do agrupamento de usuários em um determinado instante de tempo. Isso leva à necessidade de um compromisso entre a periodicidade de início de cada novo agrupamento e a qualidade de experiência do usuário: quanto maior esse intervalo, menor o número de fluxos total que precisa ser mantido pelo servidor, porém maior o tempo que os usuários são obrigados a aguardar até o início da transmissão do

conteúdo. Além disso, os usuários perdem parte da interatividade que teriam no caso de uma transmissão *unicast*, não podendo facilmente retroceder ou avançar a exibição no tempo de forma arbitrária.

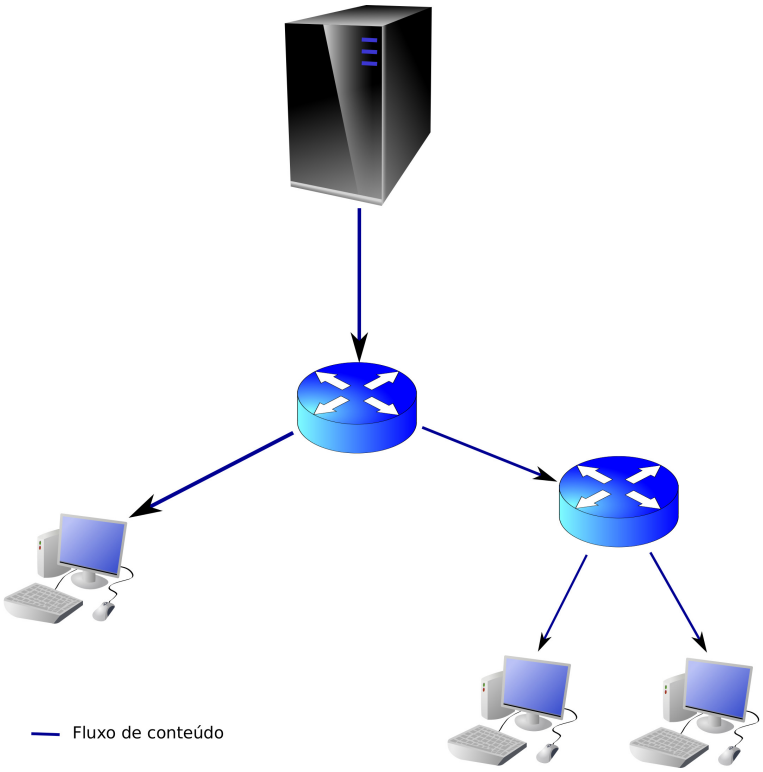

Figura 4: Estrutura de uma rede com solicitações *multicast*.

Finalmente, uma abordagem de maior interesse para este trabalho é a utilização dos recursos do próprio usuário para auxiliar no serviço de provimento dos dados, usando um protocolo P2P para que diferentes usuários obtenham o conteúdo de seus pares ao invés de recorrer o servidor. Especificamente, parte dos dados já consumidos por um usuário podem ser fornecidos a um segundo usuário usando protocolos como o BitTorrent, desde que os dispositivos finais sejam configurados para armazenar localmente tal conteúdo. Assim, contanto que os pedaços do conteúdo sejam obtidos em uma velocidade suficiente para que o *buffer* do aplicativo de reprodução de dados multimídia não se esvazie, a qualidade de experiência dos usuários pode ser assegurada. Isto requer que os nós tenham disponibilidade de recursos computacionais (e.g., capacidade de armazenamento e banda) e sejam também razoavelmente estáveis na

rede (para evitar repetidas desconexões durante a obtenção de pedaços do conteúdo). Além da inerente escalabilidade e disponibilidade de conteúdo provida por esse modelo, o fato dos dados serem melhor espalhados pela rede permite o uso de mecanismos de localidade (MIERS et al., 2010a) para tirar proveito da potencial proximidade entre usuários. Pode-se ainda adotar um modelo de transmissão de conteúdo, que combina a transferência proveniente do próprio servidor com mecanismos de troca de dados via P2P entre usuários, conforme ilustrado na Figura 5.

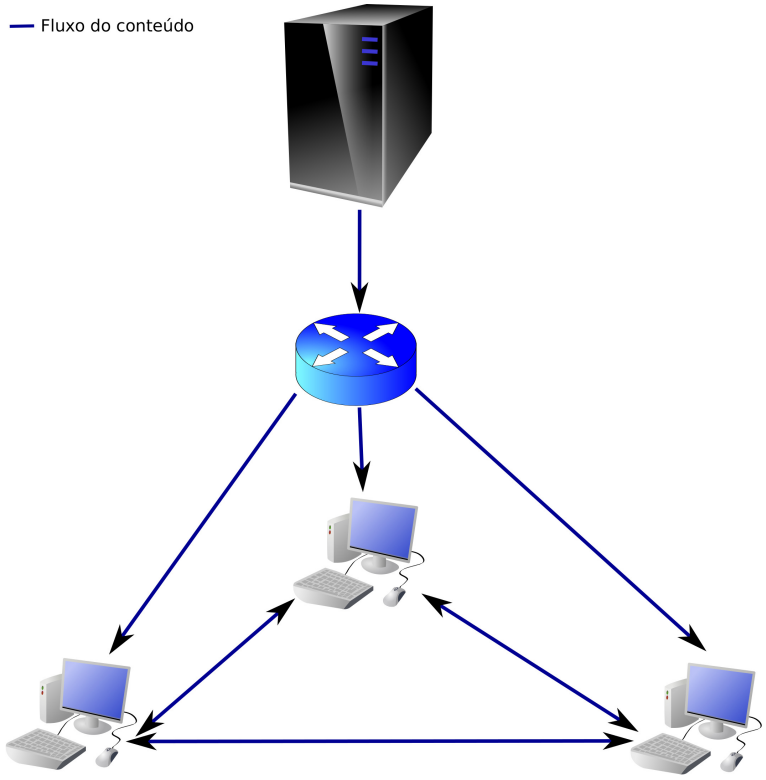

Figura 5: Estrutura de uma rede com troca de dados entre os pares.

A literatura apresenta diversas propostas de sistemas IPTV baseados em P2P para permitir a obtenção de conteúdo sem intervenção direta de servidores (MOL et al., 2009; RODRIGUES; MONTEIRO, 2012; MIERS et al., 2010b; GALLO et al., 2009). Existem também outras propostas que fazem uso de unidades de *cache* colocadas em locais estratégicos da rede de uma forma híbrida (HASSAN; NENG; SUAN, 2012; LU et al., 2011; YIN et al., 2009). Com o uso de unidades de *cache*, a carga para distribuição do conteúdo é balanceada entre as unidades de *cache*.

2.5 Considerações sobre o capítulo

Neste capítulo são abordadas as tecnologias envolvidas neste trabalho. Especificamente, discutiram-se: o BitTorrent, que consiste em um método de entrega de conteúdo para os usuários do sistema; o uso de redes gerenciadas, que permitem controle e configuração dos elementos de rede da infraestrutura do sistema; o conceito de privacidade de usuários e técnicas para obter tal serviço, como tradução de endereços na rede via NAT (que provê privacidade apenas entre usuários finais) e redes TOR (que provê anonimato com relação a todos os nós da rede, inclusive roteadores que compõem sua infraestrutura), sendo que o primeiro método faz parte da solução proposta no próximo capítulo dada a necessidade de não haver privacidade com relação ao provedor de conteúdo; por fim, é abordada a estrutura de um sistema IPTV, bem como os tipos de mecanismos de entrega de conteúdo por estas redes e quais as diferenças entre eles.

Realizado este levantamento, é possível elaborar uma solução que reúna os aspectos destas tecnologias, a qual é apresentada no capítulo 4. No capítulo 3, é apresentado o levantamento dos problemas de segurança observados em sistemas IPTV baseados em redes P2P.

3 REVISÃO DA LITERATURA: SEGURANÇA EM SISTEMAS DE IPTV BASEADOS EM P2P

O capítulo apresenta um levantamento dos problemas de segurança encontrados na literatura. Tais problemas são referentes a sistemas IPTV baseados em redes P2P, com foco em redes BitTorrent. Como já mencionado na Seção 2.1, o uso de um protocolo P2P para obtenção de conteúdo em um sistema IPTV tem como características:

- Redução da carga sobre os servidores de distribuição de conteúdo;
- Possibilidade de distribuir os dados de forma a deixá-los mais próximos (conforme alguma métrica de localidade) ao usuário; e
- Maior resiliência a falhas dada a maior disponibilidade do conteúdo para os usuários.

Por outro lado, esta abordagem também apresenta desafios importantes. Em especial, a participação ativa de clientes no processo de envio de dados permite que usuários maliciosos tentem tirar proveito do sistema, afetando sua segurança de diferentes formas (GHEORGHE; CIGNO; MONTRESOR, 2011), tais como:

- Comprometendo a qualidade dos serviços oferecidos, como por exemplo, alterar o conteúdo entregue, e por consequência, comprometendo a experiência dos usuários;
- tirando proveito da capacidade da rede sem contribuir com seus pares, ou seja,

obter conteúdo dos pares sem necessariamente oferecer conteúdo para os pares; e

- comprometendo a privacidade dos outros usuários da rede, realizando o levantamento de perfil de usuários.

Neste capítulo, são apresentados alguns dos possíveis ataques que podem ser perpetrados por usuários maliciosos na rede, bem como soluções propostas na literatura para tais problemas. A discussão tem como foco vulnerabilidades particularmente sérias no contexto de sistemas de IPTV baseados em P2P, em especial redes construídas usando o protocolo BitTorrent, que é usado para troca de dados neste trabalho. Esta análise, além de apresentar de forma ampla o contexto em que se insere o presente trabalho, também tem como objetivo explicitar o interesse na construção de soluções voltadas a privacidade nessas redes, tema este pouco explorado na literatura especializada. As próximas seções apresentam os tipos de problemas que foram encontradas na literatura.

3.1 Poluição de dados

Em uma rede P2P, o conteúdo a ser distribuído aos usuários fica espalhado entre os pares da rede. Assim, quando um usuário deseja obter um determinado conteúdo, ele primeiro precisa descobrir sua localização, o que é comumente realizado com a ajuda de um diretório central (no caso do BitTorrent, o *tracker*) que indica algumas dentre as múltiplas fontes que possuem o conteúdo desejado. Cada fonte indicada pelo servidor central, ao ser contactada, pode contribuir com uma parte do conteúdo localmente armazenado (e.g., um vídeo requisitado pelo cliente por meio de funcionalidade de VoD). Dessa forma, o conjunto de fontes pode contribuir para acelerar a obtenção da totalidade do conteúdo pelo solicitante. À medida que conteúdos são obtidos, os solicitantes iniciais também passam a atuar como fontes de dados, aumentando desta

forma a disponibilidade do conteúdo na rede (especialmente os mais populares).

Em certos casos, um usuário mal intencionado pode apresentar um comportamento nocivo a este processo colaborativo, enviando um conteúdo diferente do solicitado para seus pares (CHRISTIN; WEIGEND; CHUANG, 2005; YANG et al., 2008). Esta comportamento malicioso acaba prejudicando todos os solicitantes da rede, pois eles não apenas recebem dados que são inconsistentes com os desejados, mas também podem vir a repassar tal conteúdo a outros usuários caso a autenticidade e consistência dos mesmos não seja verificada adequadamente. A Figura 6 ilustra esse cenário, conhecido genericamente como "poluição de dados".

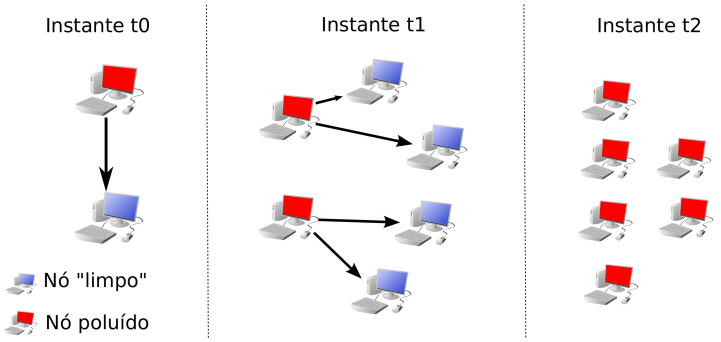

Figura 6: Propagação de conteúdo poluído em uma rede P2P.

Existem na literatura diversos trabalhos que abordam ataques de poluição de dados, apresentando propostas de solução para o problema. Em geral, soluções que podem reduzir a ocorrência de poluição de dados em sistemas P2P envolvem a implementação de mecanismos de verificação de integridade dos dados recebidos, como algoritmos de *hash* criptográfico (CHRISTIN; WEIGEND; CHUANG, 2005; YANG et al., 2008; VIEIRA et al., 2013). Apesar da possibilidade de eficácia desta abordagem em redes BitTorrent em geral, o custo adicional incorrido por sua utilização conflita, de uma forma geral, com o requisito de tempo imposto pelo sistema IPTV. Por esta razão, existem soluções de verificação de integridade de dados especialmente desenvolvidas para o cenário de envios de fluxo multimídia (HABIB et al., 2005; ZHANG; CHEN; SANDHU, 2005). Resumidamente, tais soluções consistem no uso de *hash* probabilístico: ao

invés de verificar o *hash* do conteúdo completo obtido, os usuários podem verificar apenas algumas porções individuais do mesmo. Desse modo, a modificação de parte daquele bloco apresenta uma probabilidade arbitrariamente grande de ser detectada, permitindo um equilíbrio configurável entre o tamanho dos blocos e a segurança provida pelo sistema contra ataques de poluição.

3.2 Operação de Free-riders

Em diversas redes P2P, e em especial em redes BitTorrent, o real benefício está no uso compartilhado de recursos para que todos os membros do sistema consigam obter o conteúdo desejado em um menor intervalo de tempo. Esse beneficio é possível porque cada usuário da rede se comporta não apenas como requisitante de dados, mas também como provedor dos mesmos.

Porém, podem existir elementos na rede que estejam interessados em usar seus recursos computacionais apenas para obter o conteúdo, dedicando pouca (ou mesmo nenhuma) banda e memória para prover dados a seus pares. Se esse comportamento egoísta for adotado por muitos usuários, a disponibilidade do conteúdo na rede acaba sendo comprometida. Além disso, embora o sistema possa funcionar, a eficiência do processo de troca de dados é bastante reduzida, reduzindo a escalabilidade do sistema como um todo. Tais elementos egoístas são conhecidos como *free-riders*, e a detecção e prevenção de tal comportamento é tema de discussão de diversos trabalhos na literatura (WANG et al., 2010; MANOHARAN; GE, 2013; LIANG et al., 2010; JIN et al., 2013; MONTAZERI; AKBARI, 2011).

Conforme brevemente discutido na Seção 2.1, o BitTorrent em si apresenta um mecanismo dedicado a esse propósito, o “tit-for-tat” ou “olho-por-olho”.

Este mecanismo tem por objetivo favorecer os pares que realizam mais contribuições de envio de dados para os demais pares, em detrimento dos que realizam pouca

contribuição, ou até mesmo aquele que não contribuem para a rede. Embora esse mecanismo seja interessante para garantir a colaboração para um determinado conteúdo, esse escopo limitado a um único conteúdo não permite avaliar o comportamento global de cada usuário na rede, algo necessário para aplicar assertivamente programas de bonificação a usuários que colaboram adequadamente. Assim, torna-se interessante a utilização de mecanismos de autenticação do usuário para poder ingressar na rede (COELHO et al., 2011; COELHO, 2011), além de mecanismos de reputação dos mesmos (VIEIRA et al., 2013; TAUHIDUZZAMAN; WANG, 2015). Com estes mecanismos, é possível identificar (e, eventualmente, punir) a ação de usuários egoístas na rede, e tomar medidas para alertar os demais usuários.

3.3 Ataque de Negação de Serviço

No processo de distribuição de conteúdo, é interessante e comum que haja uma distribuição entre as requisições na rede. Isso pode ser feito, por exemplo, por ação direta dos *trackers*, que nas listas de usuários fornecidas tentam evitar a presença repetida de um mesmo nó com o objetivo de não sobrecarregá-lo (WU et al., 2011; SU et al., 2012). Como característica adicional, tais mecanismos também acabam por aumentar a disponibilidade do conteúdo entre os pares da rede, de forma a evitar que poucos nós monopolizem uma determinada parte do conteúdo.

Infelizmente, é possível atacar esse processo de modo a fazer com que diversas solicitações sejam direcionadas a um ou poucos nós. Esse elevado número de solicitações desnecessárias pode vir a sobrecarregar os nós atingidos, afetando a qualidade do serviço percebido pelos usuários afetados. Adicionalmente, as vítimas passam a ter dificuldade em atender requisições legítimas, podendo-se chegar ao caso extremo em que um nó processa apenas requisições maliciosas. Nesta condição, o nó deixa de responder solicitações autênticas, passando a ser considerado indisponível para o resto da rede. A vítima do ataque pode até mesmo vir a ser erroneamente considerada

egoísta por eventuais mecanismos de detecção de usuários que colaboram pouco com o sistema, sofrendo as sanções previstas sem de fato merecer tal tratamento. Se vários nós da rede são atingidos pelo ataque, o sistema como um todo pode vir a ser comprometido e a disponibilidade de conteúdos na rede pode ser afetada. Esse tipo de ataque, conhecido como ataque de negação de serviço, não é incomum em redes P2P (WU et al., 2011; SU et al., 2012), especialmente considerando o elevado volume de tráfego que as mesmas costumam manipular. A Figura 7 ilustra os efeitos desse tipo de ataque contra a disponibilidade de uma sistema P2P.

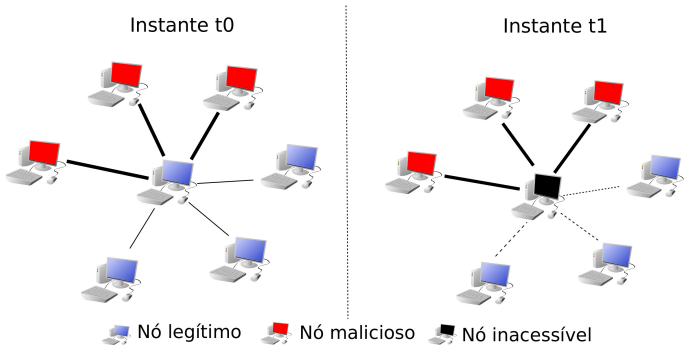

Figura 7: Exemplo de ataque de negação de serviço em uma rede P2P.

Existem na literatura propostas de solução para este tipo de problema, como aquela apresentada em (SU et al., 2012), que se baseia no ranqueamento dos usuários da rede. Dessa forma, é garantido que os usuários maliciosos tenham baixa reputação e assim não prejudiquem o sistema (TAUHIDUZZAMAN; WANG, 2015). Pois usuários com baixa reputação não terão prioridade durante a transferência de dados, e desta forma não poderão prejudicar os demais usuários da rede.

3.4 Privacidade de Usuários

Quando a troca de dados é feita via protocolo P2P, é necessário que os usuários saibam a origem e destino dos dados transmitidos. No caso de um sistema IPTV, isso implica que cada usuário do serviço torne pública suas preferências de consumo de conteúdo, permitindo que os usuários na rede criem perfis de uso uns dos outros, algo

não necessariamente desejado.

Soluções voltadas a prover privacidade a usuários podem ser classificadas em dois tipos principais. O primeiro consiste em o usuário ter a habilidade de especificar que tipo de conteúdo deseja compartilhar com um determinado grupo de usuários. Mais especificamente, esse tipo de solução se concentra na forma como o conteúdo é disponibilizado, permitindo que um usuário crie listas de controle de acesso (*access control lists* – ACLs) para um determinado grupo de usuários (ISDAL et al., 2010). O uso de tal solução implica na necessidade de saber de antemão com quem se deseja compartilhar tais arquivos. Além disso, a privacidade não é total. Afinal, o usuário que recebe o conteúdo sabe para quem o está enviando, assim como o usuário que está recebendo sabe quem é a origem do conteúdo.

Outro tipo de solução para resolver o problema de privacidade dos usuários consiste em inibir a capacidade de um usuário identificar quem está enviando dados para ele, e, analogamente, quem está recebendo dados dele. Esse é o conceito básico apresentado pelas redes TOR (DINGLEDINE; MATHEWSON; SYVERSON, 2004b), protocolo mencionado na Seção 2.3, em que os dados enviados por um usuário passam por um circuito virtual formado por um número configurável de nós, cada qual substituindo o endereço IP do nó anterior. Assim, cada usuário no circuito virtual conhece apenas o nó predecessor e sucessor, impedindo-se que o destino descubra a origem de um determinado pacote. Apesar do amplo sucesso do protocolo TOR para prover navegação segura na web, a utilização específica de redes TOR tem como característica os (possivelmente elevados) atrasos na entrega de pacotes (BLOND et al., 2013). Isto ocorre porque o processo de anonimização característico do TOR envolve usuários espalhados em diversos locais da rede, de modo que a entrada e saída de nós e a heterogeneidade da rede podem levar a circuitos virtuais pouco confiáveis ou eficientes.

3.5 Considerações sobre o capítulo

Do levantamento de problemas realizado e apresentados neste capítulo, observa-se a existência de problemas que podem comprometer vários aspectos do sistema IPTV, como o seu desempenho, o conteúdo entregue aos usuários e a privacidade do usuário. Por outro lado, a análise das soluções propostas para resolver os problemas de segurança observados revela que a maioria das deles se preocupa com problemas relacionados à:

- Integridade do conteúdo transmitido;
- Disponibilidade do conteúdo na rede; e
- Presença de usuários legítimos na rede.

As soluções encontradas na literatura abordando privacidade (e.g., o protocolo TOR) são bastante genéricas, não levando em consideração necessidades inerentes a sistemas de IPTV tais como baixa latência na obtenção de dados. Com o objetivo de preencher tal lacuna, o Capítulo 4 descreve uma solução de privacidade especialmente projetada para uso em sistemas de IPTV construídos usando redes gerenciadas.

4 SOLUÇÃO PROPOSTA: P2PRIV-TV

Este capítulo apresenta uma solução para privacidade dos usuários voltada a redes de IPTV-P2P. A solução proposta, denominada P2Priv-TV, tem como objetivo a inibição da capacidade dos usuários em identificar seus interlocutores na rede, embora possam se comunicar direta e eficientemente usando protocolos P2P para a troca de dados. Ao mesmo tempo, a solução permite ao provedor do serviço de IPTV identificar os usuários que estejam se comunicando, algo importante para que seja possível aplicar mecanismos de bonificação a usuários que contribuem adequadamente com a rede (ou punição para aqueles que não o fazem).

4.1 Descrição de requisitos

A solução proposta leva em consideração as seguintes premissas:

- A rede composta pelo sistema é uma rede *overlay* sobreposta à rede internet que atende os usuários.
- O mecanismo de envio de conteúdo para os usuários é o protocolo BitTorrent.
- Os elementos de rede que compõem a infraestrutura da solução são controlados pelo provedor de IPTV, sendo gerenciáveis por algum tipo de mecanismo de gerenciamento de redes.

Atendidas estas premissas, o sistema precisa atender um conjunto de requisitos

funcionais e não funcionais, os quais são descritos a seguir e guiaram o desenvolvimento da solução proposta.

4.1.1 Requisitos Funcionais

Os requisitos funcionais para a solução proposta são:

- *Privacidade dos usuários*: entregar ao usuário conteúdo multimídia sem que o mesmo consiga identificar o real fornecedor do conteúdo.
- *Verificabilidade por operador*: o operador responsável pelo gerenciamento da infraestrutura do sistema deve ter condições de identificar origem e destino dos dados dentro da rede.
- *Automação*: a implementação do sistema deve ser capaz de acessar os elementos de rede responsáveis pelo roteamento do tráfego entre os usuários que estão obtendo conteúdo de forma automática, sem intervenção de operadores humanos.

4.1.2 Requisitos Não Funcionais

Os requisitos não funcionais para a solução proposta são os seguintes:

- *Eficiência*: a implementação da solução proposta não deve gerar para o usuário perda da Qualidade da Experiência no uso do sistema.
- *Transparência*: o sistema de privacidade deve ser transparente para o usuário final, que não precisará utilizar um cliente BitTorrent modificado para ser compatível a solução de privacidade.
- *Volume de Utilização*: o sistema deve suportar diversos usuários simultaneamente.

4.2 Arquitetura da solução

A arquitetura do P2Priv-TV é inspirada pelo mecanismo de anonimização dos usuários usado pela rede TOR. Especificamente, o comportamento básico da rede TOR consiste no transporte de dados através de usuários que, agindo como roteadores, alteram o cabeçalho dos pacotes de dados. Essa alteração ocorre especificamente no campo que especifica a origem do pacote, de forma que o destino não saiba quem realizou a solicitação. Com isso, o usuário da rede TOR fica anonimizado e o site acessado não possui informações sobre sua localização.

Conforme brevemente discutido na Seção 2.4, uma das características desse mecanismo é o fato do mesmo depender da disponibilidade de usuários com boa capacidade computacional e estáveis na rede. Caso contrário, o desempenho do circuito de anonimização pode ficar bem aquém do desejado em sistemas de IPTV. Por outro lado, em uma rede privada na qual os elementos de rede são controlados pelo provedor de IPTV, a necessidade de recorrer diretamente a usuários para fazer a anonimização dos dados é reduzida: afinal, ao invés de depender de clientes finais (não necessariamente confiáveis) agindo como roteadores, pode-se utilizar os próprios roteadores da rede para essa tarefa. Como os roteadores fazem parte de uma rede *overlay* sobreposta à rede internet, o provedor de conteúdo tem acesso ao mesmo e pode configurá-lo de acordo com suas necessidades. Essa é a ideia básica por trás da solução proposta no presente trabalho, denominada P2Priv-TV, cuja arquitetura envolve os seguintes elementos (veja Figura 8):

- Cliente: é o responsável por receber o conteúdo e exibí-lo ao usuário final. Também compartilha esse conteúdo com os demais pares da rede.

- Pares: são os responsáveis por fornecer conteúdo ao cliente. Podem ser tanto unidades de cache, que são dispositivos de infraestrutura responsáveis por realizar cache dos dados que passam por eles, facilitando a distribuição aos clientes,

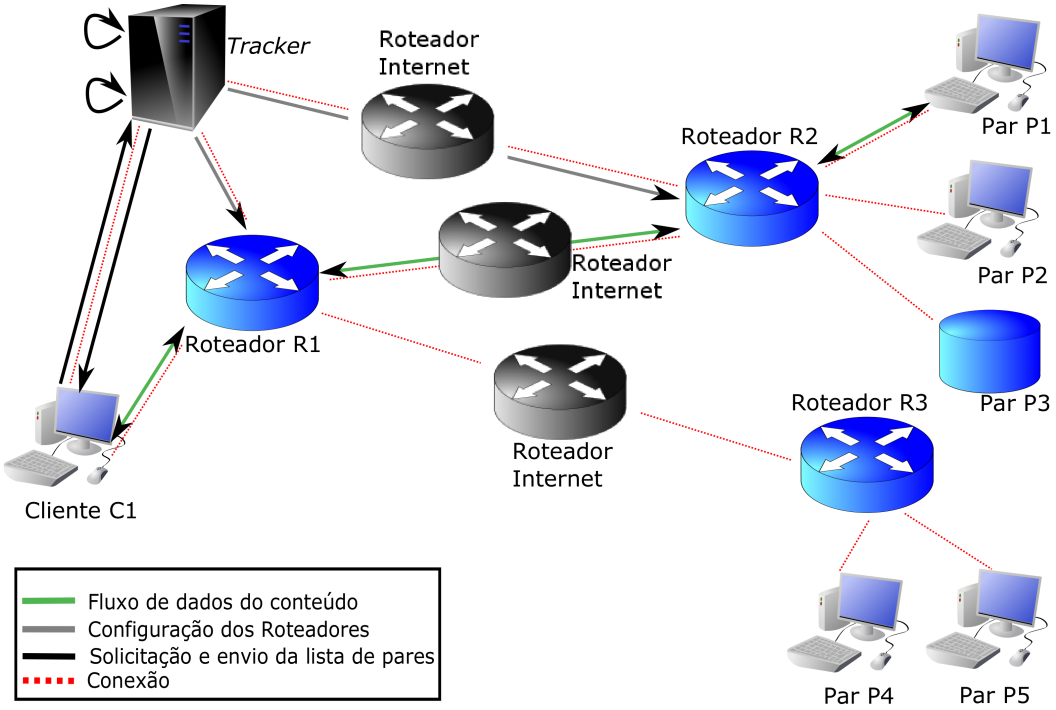

Figura 8: Estrutura do mecanismo proposto pelo P2Priv-TV.

quanto os demais usuários da rede, que proveem o conteúdo já consumido e armazenado localmente.

- *Tracker*: servidor que fornece a lista de pares para o cliente. Dentro deste elemento está inserido o módulo responsável por anonimizar a comunicação entre o cliente e os pares da rede.

- Roteadores: elementos de rede que realizam o direcionamento de tráfego entre os pares e o cliente que deseja o conteúdo, efetuando a tradução de endereços conforme adequado. Os roteadores marcados como Roteadores Internet não fazem parte da solução e estão listados para demonstrar que a solução trata de uma rede *overlay*, que é sobreposta à rede da internet comercial.

4.3 Fluxo de operação

O fluxo de ações do P2Priv-TV durante a troca de conteúdo entre pares pode ser exemplificado na comunicação entre o Cliente C1 e o Par P1, conforme indicado com

números (e sem os roteadores da internet listados) na Figura 9:

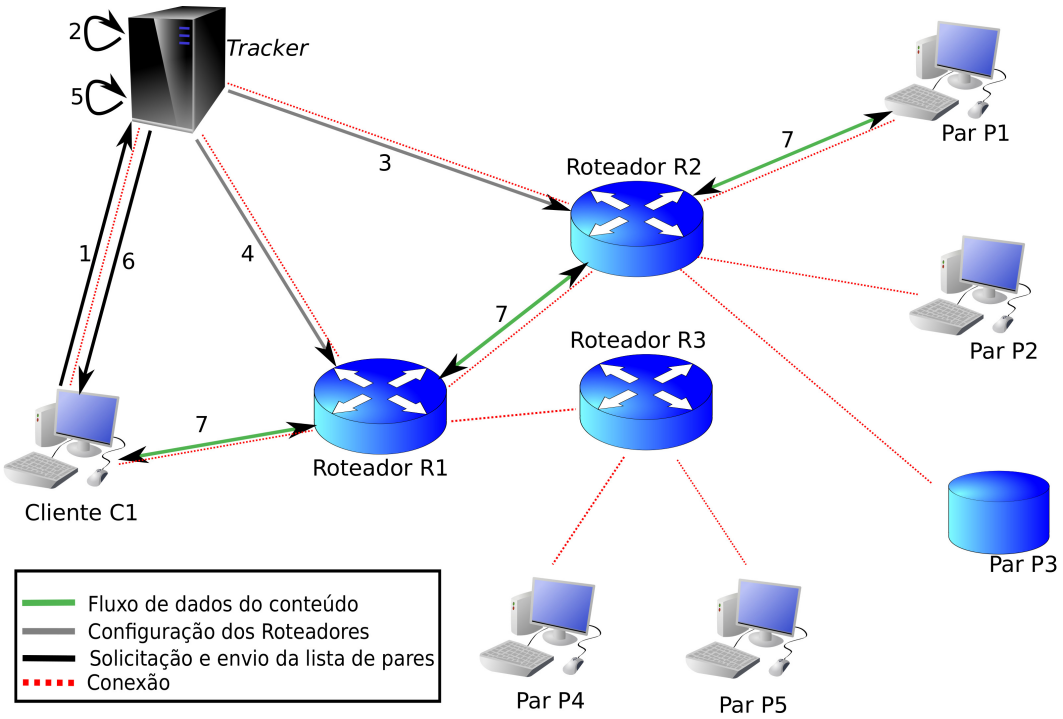

Figura 9: Estrutura do mecanismo proposto pelo P2Priv-TV.

1. O Cliente $C1$, de posse do arquivo torrent, solicita ao *tracker* um determinado conteúdo;

2. O *tracker* gera uma lista, com o conjunto IP e porta (IP_{P1}:P_{P1}, IP_{P2}:P_{P2},etc), dos pares que possuem esse conteúdo, usando os mecanismos de um *tracker* BitTorrent para essa tarefa;

3. O *tracker*, por meio do módulo de anonimização, entra em contato com o roteador mais próximo a cada um dos pares que possuem o conteúdo. O roteador mais próximo aos pares é definido durante a configuração do módulo de anonimização. Na Figura 9, por exemplo, o roteador $R2$ é contatado por estar mais próximo do par $P1$. Por meio dessa comunicação é especificada a regra de NAPT nesses roteadores, que regulam o acesso do cliente $C1$ aos seus pares. A regra NAPT criada em R2 especifica então que toda comunicação vinda do roteador

$R1$ (o mais próximo ao cliente $C1$) com destino ao roteador $R2$ na porta P_{R2} deve ser direcionada para o par $P1$ na porta P_{P1};

4. Além de configurar os roteadores próximos aos pares com os quais o cliente $C1$ se comunica, o *tracker* também entra em contato com o roteador $R1$ para criar a regra NAPT que complementa aquela criada no roteador $R2$. Desta forma, é criada em $R1$ uma regra NAPT especificando que toda comunicação vinda do cliente $C1$ com destino ao roteador $R1$ na porta P_{R1} deve ser direcionada para o roteador $R2$ na porta P_{R2};

5. Tendo configurado os roteadores que farão parte do processo de troca de dados, o módulo de anonimização do *tracker* substitui na lista dos pares gerada no passo 2 todos os endereços e portas pelos equivalentes anonimizados. Por exemplo, o endereço e porta de $P1$ é substituído pelo endereço e porta de $R1$;

6. O *tracker* envia para o cliente $C1$ a lista já anonimizada de pares aos quais ele deve se conectar para obter o conteúdo; e

7. A partir deste momento, a conexão entre o cliente $C1$ e o par $P1$ pode ser estabelecida sem origem e destino se conheçam e de forma transparente, conforme mecanismo do protocolo BitTorrent.

Deste ponto em diante, o cliente pode então fazer as solicitações de download e começar a receber o conteúdo. As conexões são estabelecidas por intermédio dos roteadores próximos a cada par da rede, sem que origem e destino se conheçam e de forma transparente: cada cliente do sistema enxerga o roteador mais próximo como solicitante ou provedor de dados. Para exemplificar como o cabeçalho dos pacotes são afetados nos roteador $R1$ e $R2$, a Tabela 1 mostra o cabeçalho dos pacotes de dados trocados entre o cliente $C1$ e o par $P1$, nos trechos entre $C1$ e $R1$, $R1$ e $R2$, e $R2$ e $P2$:

Dessa forma, pode-se observar que entre $C1$ e $R1$ só há endereços e portas de $C1$

Tabela 1: Cabeçalho dos pacotes de dados de conteúdo trocados entre o cliente $C1$ e o par $P1$

Trecho	$C1 => R1$	$R1 => R2$	$R2 => P1$
Origem	IP_{C1}:P_{C1}	IP_{R1}:P_{C1-R1}	IP_{R2}:P_{R1-R2}
Destino	IP_{R1}:P_{R1}	IP_{R2}:P_{R2}	IP_{P1}:P_{P1}

e $R1$. Da mesma forma, tem entre os elementos $R1$ e $R2$ apenas endereços de $R1$ e $R2$. E finalmente, entre os elementos $R2$ e $P1$, tem apenas endereços de $R2$ e $P1$.

Como mecanismo adicional, após a execução do último passo da Figura 9, o servidor deve verificar periodicamente em cada roteador da rede o uso de cada regra NAPT. Se uma determinada regra estiver em desuso por um tempo superior a um limite pré determinado, a regra NAPT é desfeita nos roteadores envolvidos. Tal mecanismo de remoção de regras NAPT em desuso é interessante para evitar problemas de escalabilidade da solução: com um grande número de regras, o roteador pode sofrer indisponibilidade para criação de novas regras e acabar não atendendo novos usuários. Novamente, este processo é transparente aos usuários: a remoção de uma regra faz com que o usuário não consiga mais alcançar o par respectivo, como se aquele par tivesse se desconectado da rede (e, na realidade, o não uso daquela regra pode indicar que isto tenha de fato ocorrido).

4.4 Considerações sobre o capítulo

Este capítulo apresenta a solução proposta para este trabalho, o P2Priv-TV, detalhando as funcionalidade que ele deve possuir. Em linhas gerais, a solução deve realizar o tráfego de dados entre o solicitante e o fornecedor de conteúdo sem que qualquer das partes possa conhecer um ao outro. Para tanto, a partir do momento em que é solicitado um conteúdo, o cliente realiza a descarga do arquivo torrent e então entra em contato com o *tracker* para obter a lista de pares que possuem o conteúdo (ou partes dele). O *tracker*, ao gerar esta lista, a envia para o módulo de anonimização, que

estabelece o roteamento do tráfego entre roteadores da rede. Esse roteamento é feito de tal maneira que tanto o nó solicitante quanto o nó fornecedor do conteúdo só conhece um dos roteadores responsável por este roteamento, que mais comumente será o roteador mais próximo de si. É também apresentado o fluxo de eventos que ocorrem a partir do momento que o usuário solicita o conteúdo até o momento em que o conteúdo começa a ser exibido.

Com a solução proposta definida, no capítulo a seguir a mesma é avaliada experimentalmente em termos de viabilidade e eficiência, de acordo com os cenários e as métricas ilustrativos. Também são apresentados as ferramentas utilizadas nos cenários criados.

5 ANÁLISE

Para avaliar os mecanismos propostos, o sistema foi emulado considerando-se diferentes cenários, construídos para avaliar diferentes condições às quais o sistema pode vir a ser submetido. O método utilizado nesta avaliação é descrito neste capítulo.

5.1 Cenário de testes

O cenário necessário para avaliação dos mecanismos que compõem o P2Priv-TV consiste em um software que realize a transmissão de conteúdo de vídeo por meio do protocolo BitTorrent, bem como clientes para o consumo desse conteúdo. Os cenários foram compostos por um conjunto de clientes solicitando o conteúdo. Nestes cenários é possível analisar se a inserção da solução proposta causa algum tipo de impacto na reprodução do conteúdo solicitado pelo usuário. Além disso, o cenário utiliza o sistema IPTV com mecanismos de localidade (MIERS, 2012). Tal sistema é composto dos seguintes elementos:

- Um servidor *tracker* principal - MTM (*Main Tracker Module*), que é responsável por gerenciar o tráfego entre os *trackers* secundários;
- Servidores *trackers* secundários - Tr que serão responsáveis por atender às solicitações dos seus respectivos pares para envio de conteúdo;
- Servidores de *proxy* de dados - *proxy* que são responsáveis por serem os primeiros pares da rede a possuir o conteúdo em sua totalidade;

- Servidores de *cache* de dados - *cache* que são responsáveis por se comportar como um par da rede, aumentando a disponibilidade do conteúdo na rede; e
- Pares - *Pm*, onde *m* é novamente um identificador numérico.

Para a implementação da solução e utilização nos cenários de testes, o sistema de localidade não se faz necessário, portanto, não será necessário mais de um servidor *tracker* secundário, e de um *proxy*. Além disso, os servidores de cache não serão utilizados para aumento da disponibilidade do conteúdo na rede. Porém, em testes realizados, o sistema não funciona sem o servidor *tracker* principal desligado. Logo, ele estará ligado durante os testes realizados, mas não terá participação efetiva em nossos testes.

Os cenários montados contêm elementos dos seguintes tipos, que foram instalados em servidores do tipo PC, rodando o sistema operacional GNU/Linux:

- Um servidor *tracker* principal - MTM;
- Um servidor *tracker* secundário - Tr;
- Um servidor de *proxy* de dados - *proxy*;
- Servidores atuando como roteadores de tráfego - *Rn*, onde *n* é um identificador numérico;
- Pares - *Pm*, onde *m* é novamente um identificador numérico.

A quantidade de servidores atuando como roteadores de tráfego e a quantidade de pares varia de acordo com os tipos de cenários emulados. O servidores atuando como roteadores de tráfego rodam o sistema operacional GNU/Linux, e utilizam a ferramenta iptables para realizar o roteamento dos pacotes que chegam à máquina. O servidor *proxy* é usado para fornecer conteúdo para os demais pares da rede, com a característica especial de atuar como um par que sempre possui o conteúdo completo.

Em todos os cenários propostos, os *trackers* principal e secundário, bem como o servidor *proxy* encontram-se em uma sub-rede diferente da sub-rede onde se encontravam os clientes que solicitavam conteúdo. A sub-rede da qual estes servidores fazem parte são gerenciadas por um servidor atuando como roteador de tráfego, $R0$. Os cenários propostos são descritos nas seções 5.1.1 e 5.1.2:

5.1.1 Um par apenas

Este cenário serve para validar o funcionamento de todo o sistema e também para permitir a medição de tempos em um cenário bastante simples, que não envolve conexões entre diversos pares da rede. Conforme ilustra a Figura 10, ele é composto por um par $P1$ e um roteador $R1$, além do roteador $R0$ que conecta o *tracker* (que permite a P1 encontrar o conteúdo) e o servidor *proxy* (que efetivamente fornece o conteúdo a $P1$). Este cenário permite, assim, verificar se o módulo de anonimização configura corretamente as regras entre os roteadores, se os dados são entregues e se o conteúdo obtido do *proxy* é de fato reproduzido pelo par $P1$ e em quanto tempo isso ocorre.

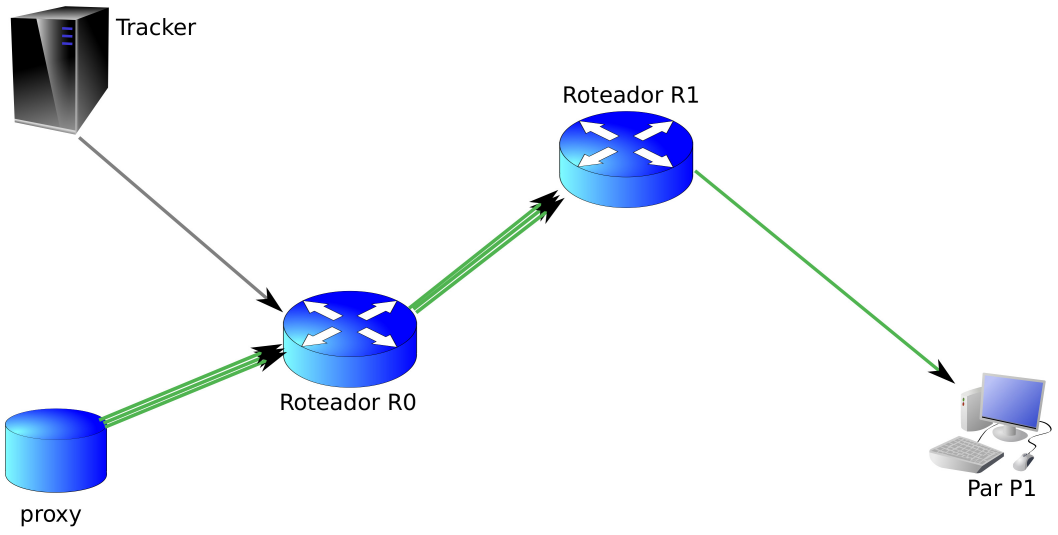

Figura 10: Cenário com apenas um par.

Neste cenário duas sub-redes são utilizadas, a primeira com os servidores (*proxy* e *tracker* principal e secundário) e a outra com o par $P1$. Dessa forma, é necessário que

o tráfego passe pelos roteadores $R0$ e $R1$ para que eles consigam direcionar o tráfego entre as sub-redes.

5.1.2 Um Par por sub-rede

Este cenário é composto por um único par solicitando e fornecendo conteúdo em cada sub-rede, conforme mostrado na Figura 11. Tal arranjo leva a um roteador Rn para cada par Pm, de modo que pode-se esperar que, quando comparado ao cenário descrito na Seção 5.1.1, esta configuração eleve o tempo de entrega dos pacotes de dados.

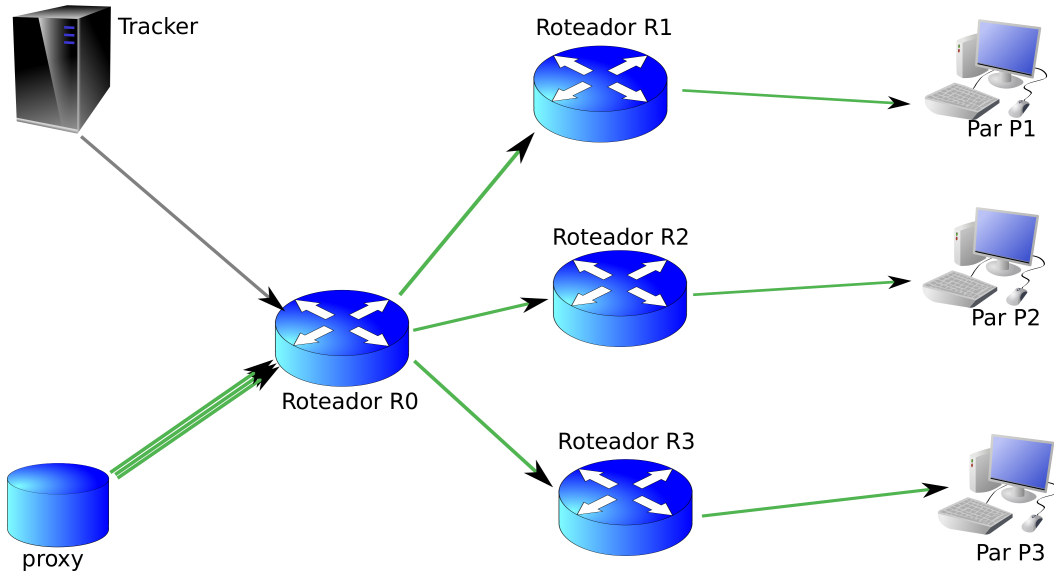

Figura 11: Cenário com apenas um par em cada sub-rede.

Em compensação, espera-se uma distribuição do uso de portas entre os roteadores, ao invés de uma carga concentrada em um único equipamento. Para este cenário foram utilizadas quatro sub-redes, a primeira com os servidores atuando como *tracker* (principal e secundário) e como *proxy*, e as outras três com cada um dos pares $P1$, $P2$ e $P3$.

5.2 Métricas

Para análise do comportamento da solução nos cenários descritos na Seção 5.1, foram utilizadas métricas para avaliar o intervalo de tempo entre a solicitação de conteúdo e o início de sua exibição (estratégia esta inspirada em (MIERS, 2012)), comparando-se situações com e sem a utilização do mecanismos de privacidade proposto. Basicamente, conforme ilustrado na Figura 12, os seguintes intervalos de tempo são medidos: tempo de inicialização de exibição ($t1$ a $t5$), tempo médio de obtenção de cada bloco do conteúdo ($t1$ a $t4$) e tempo médio de configuração do módulo de anonimização ($t1$ a $t2$). Adicionalmente, verifica-se também a ocupação de portas dos roteadores para que seja feita a tradução de endereços. Essas métricas são discutidas com mais detalhes nas Seções 5.2.1, 5.2.2, 5.2.3 e 5.2.4.

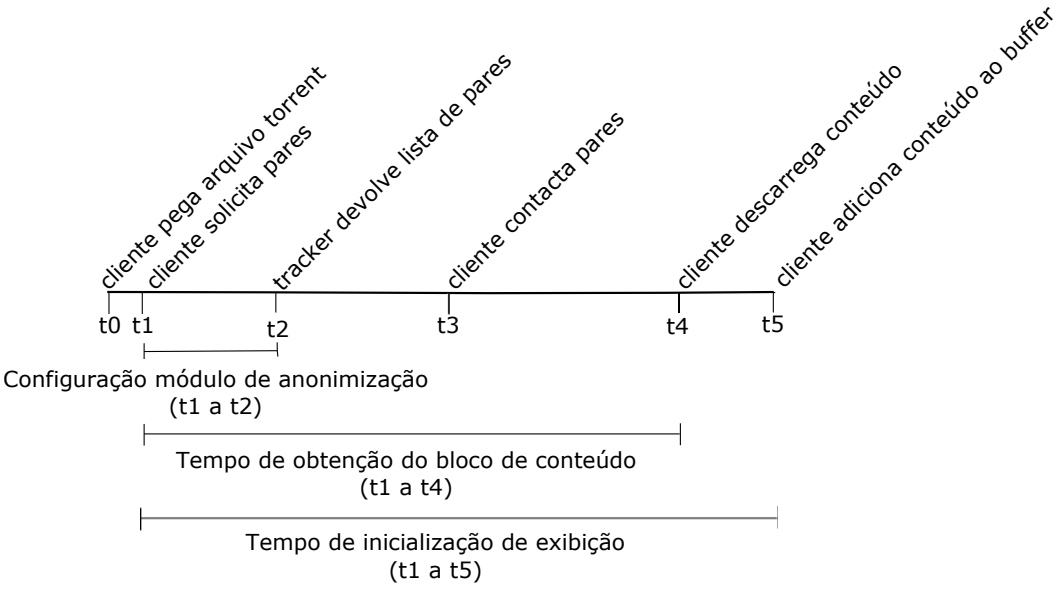

Figura 12: Composição do tempo de inicialização da exibição do conteúdo.

5.2.1 Tempo de inicialização da exibição

É o tempo necessário para o conteúdo ser exibido ao usuário, o qual pode ser medido por meio da análise de arquivos de *log* do sistema. Em sistemas de IPTV, espera-se que este tempo não exceda 5 segundos (GREENGRASS; EVANS; BEGEN, 2009;

MANZATO; FONSECA, 2011). Assim, os atrasos impostos pela configuração de regras de NAPT da solução proposta não devem degradar o desempenho do sistema além desse limite.

5.2.2 Tempo médio para a obtenção de cada bloco de conteúdo

É o tempo necessário para se obter um bloco de conteúdo de dados suficiente para sua exibição, medido pela análise dos carimbos de tempo nos *logs* do sistema. Esta métrica permite avaliar o impacto dos mecanismos de tradução de endereços sobre o desempenho global da rede, e contribui para o tempo total de inicialização mencionado na Seção 5.2.1.

5.2.3 Tempo médio de configuração do módulo de anonimização

É o tempo necessário para que o módulo de anonimização realize o procedimento descrito na Seção 4.2. Neste procedimento, o módulo deve entrar em contato com os roteadores de tráfego para realizar a configuração das regras NAPT. Para fins de otimização, o módulo também verifica se já existe uma regra estabelecida entre origem e destino, com as mesmas portas, caso em que fornece ao *tracker* a rota já estabelecida ao invés de entrar em contato com os roteadores de tráfego. Tal procedimento evita o uso desnecessário de portas para rotas já estabelecidas. Novamente, este intervalo contribui para o tempo total de inicialização medido conforme Seção 5.2.1.

5.2.4 Ocupação de portas lógicas dos roteadores

É a percentagem média e máxima do uso das portas disponíveis nos roteadores da rede, verificados pela análise dos *logs* desses equipamentos. Esta métrica é interessante para avaliar a escalabilidade da solução de privacidade proposta, já que roteadores cujas portas estão todas ocupadas são incapazes de fazer a tradução de endereços para novos usuários.

É importante notar, no entanto, que nos cenários apresentados esta métrica é mencionada principalmente para fins ilustrativos: afinal, o número total de portas utilizadas em um roteador qualquer será sempre diretamente proporcional ao número de clientes conectados àquele roteador, multiplicado pelo número de pares aos quais ele estiver conectado naquele momento. Desta forma, a escalabilidade do sistema é linear, e pode ser melhorada com uma redistribuição dos clientes entre esses roteadores. Além disso, tal abordagem de limitar o número de clientes conectados a um mesmo roteador é esperada em qualquer plataforma real, dado que aceitar um número elevado provavelmente levaria a problemas de desempenho com a própria distribuição de conteúdo multimídia a esses clientes. Portanto, pode-se considerar que a solução de privacidade aqui proposta está alinhada com essa estratégia. Na Seção 5.3, será discutido o processo de emulação.

5.3 Emulação

Para verificar a viabilidade do mecanismo proposto, é necessário implementar uma versão de tal mecanismo e embutí-lo em um sistema de IPTV que realize a transmissão via BitTorrent. Esse tipo de verificação é possível de se realizar utilizando um ambiente de emulação, que oferece uma infraestrutura semelhante à encontrada em uma infraestrutura real, porém em um ambiente controlado. Neste ambiente de emulação, as máquinas presentes são controladas e as alterações de comportamento dos elementos do ambiente podem ser monitoradas e analisadas.

O ambiente de emulação utilizado neste trabalho consiste em uma rede virtual emulada usando servidores próprios. O sistema de IPTV utilizado é o descrito em (MIERS et al., 2010b), que apresenta uma solução de localidade para entrega do conteúdo aos usuários. Este sistema é composto por um conjunto de *trackers* que irão atender às demandas locais de seus clientes priorizando os recursos dentro da mesma localidade. Caso não exista recurso suficiente para atender esta demanda, são acionados recursos

de outras localidades. Os elementos que compõem este sistema são um módulo *tracker* principal, um módulo *tracker* secundário, um módulo *proxy*, que age como uma unidade de *cache* do sistema, e os clientes, que são clientes BitTorrent, usando o jVLC[1] como reprodutor do conteúdo para o usuário.

A implementação do P2Priv-TV neste sistema tem o intuito de agregar funcionalidades ao sistema, oferecendo um sistema de IPTV com mecanismos de privacidade em adição à solução de localidade para entrega de conteúdo aos usuários já disponível.

5.4 Resultados Obtidos

Nesta seção são discutidos os resultados obtidos com o ambiente de emulação construído usando as máquinas virtuais correspondendo aos cenários propostos na Seção 5.1. Além dos resultados obtidos, são discutidos algumas limitações encontradas durante a implementação da solução na Seção 5.5.

5.4.1 Um par apenas

Nesta configuração, o par P1 solicita conteúdo apenas ao servidor *proxy*, que é o detentor inicial de conteúdo. Como há comunicação apenas entre as duas máquinas e na mesma porta, o módulo de anonimização realiza o procedimento descrito em 5.2.3, para otimizar o número de portas ocupadas. Neste caso em específico, tem-se então uma porta ocupada nos dois roteadores que estão próximos ao fornecedor e receptor do conteúdo. Como cada pedaço de conteúdo enviado possui 2 segundos de duração, obtê-los em um tempo menor do que esse é o suficiente para que o *buffer* não seja esvaziado, evitando prejudicar a reprodução do conteúdo para o usuário. Os resultados obtidos nesse cenário são apresentados na Tabela 2:

Analisando os dados obtidos, pode-se observar que a implementação da solução

[1] http://freecode.com/projects/jvlc

Tabela 2: Resultados Obtidos

Métricas	Com a solução	Sem a solução
Tempo de inicialização da exibição	969 ± 54 ms	483 ± 89 ms
Tempo médio para obtenção de cada bloco de conteúdo	1071 ± 337 ms	1077 ± 112 ms
Tempo médio para configuração do módulo de anonimização	522 ± 41 ms	não se aplica
Ocupação de portas dos roteadores	1 porta em R0 e 1 porta em R1	não se aplica

aumentou perceptivelmente o tempo de inicialização da exibição do conteúdo, que chegou a ficar até 100% superior ao tempo em que nenhum serviço de privacidade é provido. Esse atraso é causado basicamente pela configuração do módulo de anonimização, dado que foi observado nas simulações que o tempo de inicialização da exibição é basicamente igual ao tempo para obtenção do primeiro bloco de conteúdo. Embora significativo, tal atraso não chega a ser de fato prejudicial nessa simulação, pois ainda assim fica-se abaixo de 20% do limite máximo de 5 s.

Já o tempo médio para a obtenção do conteúdo, por outro lado, é pouco afetado pelo módulo de anonimização. Afinal, o atraso introduzido pelo mesmo consiste basicamente na tradução de endereços, processo este muito simples e realizado rapidamente pelo iptables. Cabe notar, entretanto, que o protocolo BitTorrent conforme implementado em (MIERS, 2012) apresenta um tempo cerca de 50% menor para obtenção do primeiro bloco do que para os blocos seguintes, o que permite o preenchimento do buffer de exibição rapidamente e um fluxo contínuo de obtenção de pedaços que previne seu esvaziamento. Como resultado, o tempo médio para a obtenção de cada bloco é cerca de duas vezes superior ao tempo para inicialização do conteúdo, independentemente do P2Priv-TV. Desta forma, é necessário que o *buffer* tenha capacidade para armazenar o conteúdo levando em consideração este tempo de obtenção de cada bloco de conteúdo.

Finalmente, dado que este cenário simples envolve apenas dois pares comunicantes, a ocupação de portas nos roteadores mantém-se em 1, tanto com quanto sem o

módulo de privacidade. Portas adicionais se fazem necessárias apenas nos casos em que há comunicação em portas diferentes de um mesmo servidor, ou em servidores diferentes.

5.4.2 Um par por sub-rede

Neste cenário, por decisão de projeto, a entrada dos pares na rede não foi simultânea, mas sequencial, seguindo a ordem $P1$, depois $P2$ e por último $P3$. Tal decisão foi tomada pois o inicialização simultânea dos clientes era afetada por problemas encontrados na ferramenta jVLC, que são discutidos adiante na Seção 5.5. Tal procedimento refletiu na forma como o conteúdo foi compartilhado. Especificamente, o par $P1$, por ter sido o primeiro na rede recebeu conteúdo apenas do *proxy*, e atuou no fornecimento de conteúdo para $P2$ e $P3$. O par $P2$, por sua vez, obteve conteúdo do *proxy* e de $P1$, e forneceu dados apenas para $P3$. Já o par $P3$ recebeu conteúdo de todos os outros nós da rede, não oferecendo dados a qualquer dos seus pares. A Tabela 3 apresenta os resultados experimentais foram obtidos neste cenário:

Tabela 3: Resultados Obtidos

Métricas	Com a solução	Sem a solução
Tempo de inicialização da exibição	2184 ± 1431 ms	537 ± 287 ms
Tempo médio para obtenção de cada bloco de conteúdo	1848 ± 1241 ms	2190 ± 1094 ms
Tempo médio para configuração do módulo de anonimização	587 ± 155 ms	não se aplica
Ocupação de portas dos roteadores	3 portas em R0, 3 portas em R1, 2 portas em R2 e 1 porta em R3	não se aplica

Conforme observado, embora a inicialização da exibição tenha ficado cerca de 400% superior ao cenário em que nenhum mecanismo de privacidade é adotado, o resultado final manteve-se novamente mais de 50% abaixo dos 5s considerados como limite superior aceitável. Neste caso, o maior atraso não se deve direta, mas sim indiretamente ao tempo de configuração do módulo de anonimização. Mais precisamente,

embora o tempo de configuração do módulo de anonimização é bastante semelhante ao cenário descrito na Seção 5.4.1, o processo de obtenção de cada bloco de conteúdo é tal que hajam repetidas trocas de informação para realizar essa configuração. Isso ocorre porque cada bloco de conteúdo é associado a um arquivo torrent próprio, configurado com suas próprias informações relativas a qual *tracker* acessar para obter a lista de pares. Logo, para cada bloco que se deseja obter, é necessário acessar o *tracker* para obter a lista de pares (que envolve o módulo de anonimização), seguido por um processo de escolher de qual par obter o conteúdo. Somente após decidir qual será o par que irá fornecer o conteúdo é que a descarga de dados é iniciada.

O impacto sobre o tempo médio de obtenção de cada bloco é, portanto, semelhante independentemente do uso da solução de privacidade, embora impacte mais no tempo de inicialização da exibição devido à repetição do processo de troca de mensagens para configuração do módulo de anonimização. Embora não tenham sido feitos testes experimentais dessa natureza, a análise dessa situação revela, porém, que ela poderia ser contornada com um maior paralelismo dessas requisições, adiantando-se a requisição de um bloco para mascarar essa latência devido à troca de mensagens de configuração. Portanto, a forma sequencial de pedido de pedaços do conteúdo conforme feita pelo cliente na presente análise pode ser vista como um cenário de pior caso.

Cabe notar, entretanto, que esses tempos só são observados na prática quando o reprodutor de vídeo utilizado na emulação, o jVLC, não apresenta erros. Mais precisamente, observou-se que a ferramenta apresenta instabilidade em sua implementação: no início do processo de descarga do conteúdo, embora o primeiro pacote seja obtido com sucesso, nem sempre as solicitações seguintes são geradas com sucesso, gerando-se uma mensagem de erro. Nesses casos, a aplicação gera um tempo de espera de 5s ao invés de tentar novamente imediatamente, tempo este por si só superior ao máximo considerado aceitável por usuários. Para eliminar esse intervalo artificialmente inserido pelo jVLC nos testes (e que não deveria ocorrer com a correção de tal

comportamento) ele é desconsiderado no testes aqui apresentados: quando tais erros acontecem, a aplicação é reiniciada para que sejam feitas as medições. Também cabe notar que essa instabilidade do jVLC não afeta o tempo de obtenção de pacotes ou o tempo de configuração do módulo de anonimização, dado que ela ocorria apenas logo na obtenção do primeiro pacote, com a descarga do restante do conteúdo ocorrendo normalmente.

Finalmente, conforme mencionado na Seção 5.2.4, a ocupação de portas nos roteadores cresce linearmente com o número de pares comunicantes quando utilizado o módulo de privacidade. Dessa forma, é importante ressaltar a necessidade de observar o número de pares que são gerenciados por um mesmo roteador, para evitar exaustão do número de portas disponíveis.

5.5 Limitações encontradas durante a implementação: iptables

A implementação da solução utilizou a ferramenta iptables[2] para realizar o roteamento dos pacotes de dados entre máquinas em sub-redes diferentes. A razão para adotar tal ferramenta é o fato do iptables ter implementada a funcionalidade de NAT, que é o principal mecanismo utilizado pelo módulo de anonimização. Entretanto, foram observadas algumas limitações com essa ferramenta que impediram a construção de ao menos um cenário adicional de interesse: aquele em que dois ou mais pares comunicantes encontram-se conectados ao mesmo roteador. Neste caso, tal roteador torna-se responsável por prevenir que ambos percebam esta situação de vizinhança ao anonimizar seus respectivos endereços, substituindo-os pelo seu próprio endereço.

Para entender a limitação do iptables em dar suporte a esse cenário, é importante primeiramente discorrer sobre fluxo que o pacote de dados percorre com o uso dessa ferramenta:

[2]http://www.netfilter.org/projects/iptables

- O pacote que chega à máquina destino passa primeiro pela cadeia de regras de pré-roteamento. Para aplicar esse conjunto de regras, é verificado no cabeçalho do pacote se as informações de endereço e porta de origem e de destino correspondem a alguma das regras existentes; caso exista, ela é executada. No caso da solução proposta, é nesta fase que os pacotes têm o seu endereço de destino alterado (originalmente, o endereço de destino é o do próprio roteador). Logo após a cadeia de regras de pré-roteamento, é verificado se o destino do pacote é a pró pria máquina ou alguma outra máquina. Dependendo do resultado, o pacote segue para a cadeia de regras de entrada (pacote com destino à própria máquina), ou segue para a cadeia de regras de redirecionamento (pacote destinado à outra máquina).

- Na cadeia de regras de entrada, é verificado novamente o cabeçalho do pacote em busca de informações que correspondam a alguma regra desta cadeia; dependendo do resultado, o pacote é enviado para tratamento pelo sistema operacional ou então é descartado.

- Já na cadeia de redirecionamento, o pacote é enviado para a cadeia de pós-roteamento. Neste caso, o pacote que é originado pela máquina e os pacotes que são redirecionados têm seus cabeçalhos verificados em busca de informarções que correspondam a alguma das regras. Caso uma regra seja encontrada, o pacote pode ter seu cabeçalho modificado novamente (comumente para alterar o endereço de origem), ser descartado, ou passar sem qualquer modificação.

É esperado que um pacote que trafegue dados do conteúdo solicitado por um cliente (1) chegue na cadeia de pré roteamento, (2) tenha seu endereço de destino alterado para o próximo destino, (3) passe pela cadeia de redirecionamento, e então (4) chegue à cadeia de pós roteamento, onde tem seu endereço de origem alterado para a própria máquina, para assim (5) sair da máquina em direção ao seu destino. Porém existem cenários onde as alterações resultantes modificam o fluxo esperado, e o pacote sai da

cadeia de pré roteamento para a cadeia de entrada, sendo tratado pelo sistema operacional do próprio roteador. O resultado é, então, a simples perda do pacote.

Infelizmente, as condições que levam à mencionada perda de pacotes ocorrem quando se tem mais de um cliente em uma mesma sub-rede, a princípio não foi feita uma implementação desse cenário. Argumenta-se, no entanto, que tal cenário extra não traria uma grande quantidade de informações adicionais com relação àqueles já apresentados, ao menos para efeito de avaliação do impacto da solução sobre o desempenho da rede. A razão é que espera-se que esse cenário leve a: um menor tempo de inicialização da exibição e obtenção dos blocos de conteúdo, dada a proximidade entre os clientes; e a um menor tempo de configuração do módulo de anonimização, já que somente um roteador é afetado.

Por outro lado, é importante discutir o porquê dessa limitação do iptables não impedir a implementação do P2Priv-TV na prática. Especificamente, tais limitações impostas pelo uso de regras NAPT poderiam ser suplantadas com uma ferramenta que atuasse em substituição ao iptables e fosse manipulada pelo módulo de anonimização de modo semelhante. Tal ferramenta operaria como um servidor, de modo que, cada vez que ele recebesse uma requisição em uma determinada porta, redirecionaria o tráfego para outra máquina em outra porta. O módulo de anonimização entraria em contato com tal ferramenta da mesma forma que é feito com o iptables para realização da configuração e depois para monitoramento do uso das portas.

Outra possibilidade de substituição do iptables é o uso de um roteador com suporte à tecnologia de SDN (*software defined network*, ou rede definidas por software) (KREUTZ et al., 2015; ZILBERMAN et al., 2015). Tal tecnologia permite que a função de roteamento de pacotes em equipamentos de rede seja divida em duas: a primeira, o plano de dados, é responsável apenas pelo encaminhamento dos dados, enquanto a segunda, o plano de controle, define como os dado são roteados. A inteligência por trás do plano de controle fica a cargo de uma entidade de rede denominada “controlador

de rede", que no caso do P2Priv-TV ficaria responsável por estabelecer como os dados seriam roteados entre origem e destino, com as regras de anonimização sendo feitas por meio da substituição de cabeçalhos IP de modo equivalente ao NAPT.

5.6 Considerações sobre o capítulo

Neste capítulo foram apresentados os cenários de testes utilizados para analisar de viabilidade e eficiência da solução proposta, bem como as métricas utilizadas para tal análise. Tais métricas têm como foco verificar se, a despeito da inclusão da solução de privacidade, o tempo de inicialização da exibição do conteúdo para o usuário mantém-se dentro do patamar de 5s. Com base nessas métrica e cenários, foi definido um ambiente de emulação para realização de testes experimentais. Os resultados obtidos mostram a viabilidade técnica da solução proposta e um custo adicional aceitável, em termos de: latência, que consiste basicamente em um intervalo adicional de 0.5 s para configurar a recepção de cada pedaço do conteúdo (o qual pode ser parcialmente mascarado ao longo da obtenção do conteúdo caso os pedaços sejam pedidos em paralelo); e uso de portas dos roteadores, que tem crescimento linear com o número de clientes na rede. Também são discutidas algumas limitações encontradas na implementação da solução, em especial relativas ao uso do iptables; como tal mecanismo pode ser substituído, por exemplo, por um programa dedicado ou por mecanismos previstos em redes SDN para tradução de endereços, tais limitações não são consideradas críticas na prática.

6 CONCLUSÕES E TRABALHOS FUTUROS

Tecnologias de P2P, como o bastante popular BitTorrent, têm grande potencial para otimizar o uso de recursos em sistemas de IPTV. Por outro lado, ela também pode trazer problemas de segurança, entre eles a redução da privacidade dos usuários.

Neste cenário, o presente trabalho teve como objetivo propor e analisar um módulo de anonimização leve o suficiente para uso em conjunto com sistemas de IPTV baseado no protocolo BitTorrent. O conceito básico desse módulo de anonimização consiste no uso de roteadores de tráfego para mascarar origem e destino através do uso de regras NAPT. Para isso, o módulo de anonimização (1) recebe a lista de pares que podem fornecer conteúdo para o par solicitante, (2) verificava quais os roteadores de tráfego mais próximos de cada par fornecedor e solicitante do conteúdo, (3) configura esses roteadores com regras NAPT para rotear o fluxo de dados entre os pares comunicantes, e (4) repassa ao par solicitante os endereços IP e de porta do roteador responsável por redirecionar o seu tráfego, para a obtenção do conteúdo. Como resultado, cada nó na rede tem a impressão de comunicar-se apenas com seu próprio roteador de borda, prevenindo obter-se qualquer informação sobre o real usuário solicitando/enviando os pedaços do conteúdo.

A análise experimental da solução proposta em ambiente emulado mostrou que a inserção de tal módulo carrega um custo principalmente em termos de aumento da latência para a obtenção do conteúdo e início de sua exibição. Especificamente, espera-se um aumento da ordem de 0,5 ms no tempo de obtenção de qualquer dado quando

para isso é necessário a (re)configuração dos roteadores para que os mesmos realizem o processo de anonimização; portanto, caso isso possa ser feito apenas uma vez por conteúdo (ou mascarado solicitando-se pedaços de conteúdo com algum grau de paralelismo), esse seria o custo introduzido pela solução proposta em relação a uma arquitetura equivalente sem tal módulo de privacidade. Ainda, no caso específico do cenário de testes utilizado, tal custo não se mostrou impeditivo à reprodução do conteúdo de mídia, dado que o tempo de inicialização da exibição manteve-se abaixo dos 5 segundos considerados aceitáveis por usuários de sistemas de IPTV (GREENGRASS; EVANS; BEGEN, 2009; MANZATO; FONSECA, 2011). Isso não significa que tal desempenho seja necessariamente repetido em um sistema real, mas mostra que a solução tem potencial para uso em tais sistemas mesmo que os tempos obtidos em uma arquitetura de IPTV em produção sejam até 2 vezes superiores ao máximo obtido durante as emulações.

Adicionalmente, existe também um custo em termos de uma maior uso de portas dos roteadores responsáveis pela tradução de endereços na rede. Conforme discutido na Seção 5.2.4, entretanto, essa métrica não parece ser crítica devido ao fato de tal ocupação crescer linearmente com o número de usuários sendo tratados por cada roteador. Portanto, é provável que a escalabilidade da solução seja mais afetada pelo tráfego de conteúdo por um mesmo roteador, que também cresce linearmente com o número de clientes sob sua responsabilidade, do que por essa maior utilização de portas.

Finalmente, com relação a trabalhos futuros, um tópico interessante é o uso de redes definidas por software para a implementação da solução de anonimização aqui proposta. Neste caso, seria possível verificar em que medida tal técnica pode atender às necessidades do sistema e eventualmente suplantar as limitações do iptables discutidas na Seção 5.5. Além disso, a implementação para fins comparativos de outros métodos de anonimização genéricos, como redes TOR (DINGLEDINE; MATHEWSON; SYVERSON, 2004a), pode ser de interesse em trabalhos futuros.

Como resultado indireto da elaboração deste trabalho, foi publicado o artigo (SIMPLICIO et al., 2014), que contém uma proposta de protocolo de detecção de trapaça em jogos de cartas colecionáveis online baseados na abordagem P2P. Além deste, outro trabalho está em desenvolvimento, visando divulgar os resultados obtidos ao final deste trabalho.

REFERÊNCIAS

BARCELLOS, M.; GASPARY, L. *Segurança em Redes P2P: Princípios, Tecnologias e Desafios*. 2006. Minicursos do Simpósio Brasileiro de Redes de Computadores (SBRC).

BLOND, S. L. et al. Towards efficient traffic-analysis resistant anonymity networks. In: ACM. *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*. [S.l.], 2013. p. 303–314.

CASE, J. et al. *RFC 1157: A Simple Network Management Protocol (SNMP)*. [S.l.], May 1990.

CEBALLOS, M.-R.; GORRICHO, J.-L. P2p file sharing analysis for a better performance. In: *Proceedings of the 28th International Conference on Software Engineering*. New York, NY, USA: ACM, 2006. (ICSE '06), p. 941–944. ISBN 1-59593-375-1. Disponível em: <http://doi.acm.org/10.1145/1134285.1134458>.

CHRISTIN, N.; WEIGEND, A. S.; CHUANG, J. Content availability, pollution and poisoning in file sharing peer-to-peer networks. In: *Proc. of the 6th ACM conference on Electronic commerce (EC'05)*. New York, NY, USA: ACM, 2005. (EC '05), p. 68–77. ISBN 1-59593-049-3. Disponível em: <http://doi.acm.org/10.1145/1064009-.1064017>.

COELHO, R. V. Comparação analítica dos esquemas de autenticação em sistemas p2p de live streaming. 2011.

COELHO, R. V. et al. Challenging the feasibility of authentication mechanisms for p2p live streaming. In: ACM. *Proceedings of the 6th Latin America Networking Conference*. [S.l.], 2011. p. 55–63.

COHEN, B. Web Page, *The BitTorrent Protocol Specification*. October 20, 2012 2008. Disponível em: <http://www.bittorrent.org/beps/bep_0003.html>.

DAVIS, C. R. et al. Structured peer-to-peer overlay networks: Ideal botnets command and control infrastructures? In: *European Symposium on Research in Computer Security*. [S.l.: s.n.], 2008. p. 461–480.

DEERING, S.; HINDEN. *Internet Protocol, Version 6 (IPv6) Specification*. [S.l.], December 1998.

DEFENSE ADVANCED RESEARCH PROJECTS AGENCY. *RFC 791: Internet Protocol*. [S.l.], September 1981.

DINGLEDINE, R.; MATHEWSON, N.; SYVERSON, P. Tor: the second-generation Onion Router. In: *Proceedings of the 13th conference on USENIX Security Symposium (SSYM'04)*. Berkeley, CA, USA: USENIX Association, 2004. v. 13, p. 21–21.

____. *Tor: The second-generation onion router*. [S.l.], 2004.

EVANGELISTA, P. et al. Ebitsim: An enhanced bittorrent simulation using omnet++ 4. In: *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2011 IEEE 19th International Symposium on*. [S.l.]: IEEE, 2011. p. 437–440. ISBN 1457704684.

FANG, X. et al. Smart grid—the new and improved power grid: A survey. *Communications Surveys & Tutorials, IEEE*, IEEE, v. 14, n. 4, p. 944–980, 2012.

GALLO, D. et al. A multimedia delivery architecture for IPTV with P2P-based time-shift support. In: *Proceedings of the 6th IEEE Conference on Consumer Communications and Networking Conference*. Piscataway, NJ, USA: IEEE, 2009. (CCNC'09), p. 447–448. ISBN 978-1-4244-2308-8. Disponível em: <http://portal.acm.org/citation.cfm?id = 1700527.1700654>.

GHEORGHE, G.; CIGNO, R. L.; MONTRESOR, A. Security and privacy issues in p2p streaming systems: A survey. *Peer-to-Peer Networking and Applications*, Springer New York, v. 4, p. 75–91, 2011. ISSN 1936-6442.

GOTTRON, C.; KÖNIG, A.; STEINMETZ, R. A survey on security in mobile peer-to-peer architectures: Overlay-based vs. underlay-based approaches. *Future Internet*, v. 2, p. 505–532, 2010.

GREENGRASS, J.; EVANS, J.; BEGEN, A. C. Not all packets are equal, part i: Streaming video coding and sla requirements. *IEEE Internet Computing*, IEEE Computer Society, Los Alamitos, CA, USA, v. 13, n. 1, p. 70–75, 2009. ISSN 1089-7801.

GU, Y. et al. *Survey of P2P Streaming Applications (Internet Draft - v.9)*. [S.l.], October 2014.

HABIB, A. et al. Verifying data integrity in peer-to-peer media streaming. In: *Conference on Multimedia Computing and Networking 2005*. BELLINGHAM: Spie-Int Soc Optical Engineering, 2005. (Proceedings of the Society of Photo-Optical Instrumentation Engineers (Spie), v. 5680), p. 1–12. ISBN 0277-786X 0-8194-5653-5. Disponível em: <<Go to ISI>://WOS:000228692600001>.

HARRINGTON, J.; KUWANOE, C.; ZOU, C. A bittorrent-driven distributed denial-of-service attack. In: *3rd International Conference on Security and Privacy in Communication Networks (SecureComm 2007)*. [S.l.: s.n.], 2007.

HASSAN, M.; NENG, C. K.; SUAN, L. C. Performance analysis of video streaming on different hybrid CDN-P2P infrastructure. In: *IET Int. Conf. on Wireless Communications and Applications (ICWCA 2012)*. [S.l.: s.n.], 2012. p. 1–6.

HEI, X.; LIU, Y.; ROSS, K. IPTV over P2P streaming networks: the mesh-pull approach. *IEEE Communications Magazine*, v. 46, n. 2, p. 86–92, 2008. ISSN 0163-6804.

INFORMATION Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model. [S.l.], 1994. Disponível em: <http://www.iso.org/iso-/catalogue_detail.htm?csnumber=20269>.

INTERNATIONAL TELECOMMUNICATION UNION. *ITU-T M.3010: Principles for a telecommunications management network*. [S.l.], February 2000. Disponível em: <http://www.itu.int/itu-t/recommendations/rec.aspx?rec=4869>.

IQBAL, R.; SHIRMOHAMMADI, S. DAg-stream: Distributed video adaptation for overlay streaming to heterogeneous devices. *Peer-to-Peer Networking and Applications*, v. 2, n. 3, p. 202–216, 2009.

ISDAL, T. et al. Privacy-preserving p2p data sharing with oneswarm. In: ACM. *ACM SIGCOMM Computer Communication Review*. [S.l.], 2010. v. 40, n. 4, p. 111–122.

JIN, Y. et al. Hybrid client-server and peer-to-peer caching systems with selfish peers. In: *Proc. of the IEEE INFOCOM 2013*. [S.l.]: IEEE, 2013. p. 1744–1752. ISBN 978-1-4673-5944-3.

KATSAROS, K. et al. A bittorrent module for the omnet++ simulator. In: IEEE. *Modeling, Analysis & Simulation of Computer and Telecommunication Systems, 2009. MASCOTS'09. IEEE International Symposium on*. [S.l.], 2009. p. 1–10.

KLERER, S. M. The osi management architecture: an overview. *Network, IEEE*, IEEE, v. 2, n. 2, p. 20–29, 1988.

KREUTZ, D. et al. Software-defined networking: A comprehensive survey. *proceedings of the IEEE*, IEEE, v. 103, n. 1, p. 14–76, 2015.

LIANG, C. et al. Incentivized peer-assisted streaming for on-demand services. *IEEE Transactions on Parallel and Distributed Systems*, v. 21, n. 9, p. 1354–1367, 2010. ISSN 1045-9219.

LIAO, X. et al. Anysee: Peer-to-peer live streaming. In: *Proc. of the 25th IEEE International Conference on Computer Communications (INFOCOM'06)*. [S.l.]: IEEE, 2006. p. 1–10.

LIU, J. et al. Opportunities and challenges of peer-to-peer internet video broadcast. *Proc. of the IEEE*, v. 96, n. 1, p. 11–24, Jan. 2008. ISSN 0018-9219.

LIU, Y.; GUO, Y.; LIANG, C. A survey on peer-to-peer video streaming systems. *Peer-to-Peer Networking and Applications*, Springer New York, v. 1, p. 18–28, 2008. ISSN 1936-6442.

LIU, Y.; SIMON, G. Distributed delivery system for time-shifted streaming systems. In: *IEEE 35th Conference on Local Computer Networks (LCN)*. [S.l.]: IEEE, 2010. p. 276–279. ISSN 0742-1303.

LU, Z. et al. Scalable and reliable live streaming service through coordinating CDN and P2P. In: *Parallel and Distributed Systems (ICPADS), 2011 IEEE 17th International Conference on*. [S.l.: s.n.], 2011. p. 581–588. ISSN 1521-9097.

MANOHARAN, S.; GE, T. X. A demerit point strategy to reduce free-riding in bittorrent. *Computer Communications*, v. 36, n. 8, p. 875–880, 2013. ISSN 0140-3664. Disponível em: <<Go to ISI>://WOS:000318466100003>.

MANZATO, D.; FONSECA, N. da. A comparison of channel switching schemes for iptv systems. In: *IEEE International Conference on Communications (ICC 2011)*. [S.l.: s.n.], 2011. p. 1–6. ISSN 1550-3607.

MARFIA, G. et al. Digital Fountains + P2P for future IPTV platforms: A test-bed evaluation. In: *4th IFIP International Conference on New Technologies, Mobility and Security (NTMS 2011)*. [S.l.]: IEEE, 2011. p. 1–5. ISBN 978-1-4244-8704-2.

MIERS, C. et al. I2TS01 – a taxonomy for locality algorithms on peer-to-peer networks. *Revista IEEE América Latina – Latin America Transactions*, v. 8, p. 323–331, August 2010. ISSN 1548-0992.

____. An architecture for P2P locality in managed networks using hierarchical trackers. In: *Proc. of the 6th International Conference on Network and Service Management (CNSM 2010)*. [S.l.: s.n.], 2010. p. 206–213. ISBN 978-1-4244-8910-7.

MIERS, C. C. *Uma arquitetura usando trackers hierárquicos para localidade em redes P2P gerenciadas*. Tese (Doutorado) — University of São Paulo, 2012.

MOL, J. et al. The design and deployment of a BitTorrent live video streaming solution. In: *Proc. of the 11th IEEE International Symposium on Multimedia (ISM'09)*. [S.l.: s.n.], 2009. p. 342–349.

MONTAZERI, A.; AKBARI, B. Mesh based P2P video streaming with a distributed incentive mechanism. In: *Int. Conf. on Information Networking (ICOIN'2011)*. [S.l.: s.n.], 2011. p. 108–113. ISSN 1976-7684.

PUSSEP, K. et al. On energy-awareness for peer-assisted streaming with set-top boxes. In: *6th International Conference on Network and Service Management (CNSM 2010)*. Piscataway, NJ, USA: IEEE Press, 2010. p. 166–173. ISBN 978-1-4244-8908-4.

RODRIGUES, P. L.; MONTEIRO, J. M. Bittorrent based transmission of real-time scalable video over p2p networks. *Sistemas Y Tecnologias De Informacion, Vols 1 and 2*, p. 156–161, 2012.

SCHOENWAELDER, J. *Simple Network Management Protocol (SNMP) over Transmission Control Protocol (TCP) Transport Mapping*. [S.l.], December 2002.

SEYYEDI, S.; AKBARI, B. Hybrid cdn-p2p architectures for live video streaming: Comparative study of connected and unconnected meshes. In: IEEE. *Computer Networks and Distributed Systems (CNDS), 2011 International Symposium on*. [S.l.], 2011. p. 175–180.

SIMPLICIO, M. A. et al. Securetcg: a lightweight cheating-detection protocol for p2p multiplayer online trading card games. *Security and Communication Networks*, Wiley Online Library, v. 7, n. 12, p. 2412–2431, 2014.

SRISURESH, P.; EGEVANG, K. B. *RFC 3022: Traditional IP Network Address Translator (Traditional NAT)*. [S.l.], January 2001.

STANDARDIZATION, I. O. for. *Information technology - Elements of management information related to the OSI Network Layer*. [S.l.], 1998.

SU, M. J. et al. Ddos vulnerability of bittorrent peer exchange extension: Analysis and defense. In: *IEEE International Conference on Communications (ICC)*. NEW YORK: Ieee, 2012. (IEEE International Conference on Communications). ISBN 1550-3607978-1-4577-2053-6. Disponível em: <<Go to ISI>://WOS:000312855701073>.

SUN, X.; TORRES, R.; RAO, S. On the feasibility of exploiting P2P systems to launch DDoS attacks. *Peer-to-Peer Networking and Applications*, Springer New York, v. 3, p. 36–51, 2010. ISSN 1936-6442. 10.1007/s12083-009-0046-6. Disponível em: <http://dx.doi.org/10.1007/s12083-009-0046-6>.

TAUHIDUZZAMAN, M.; WANG, M. Fighting pollution attacks in p2p streaming. *Computer Networks*, Elsevier, v. 79, p. 39–52, 2015.

TRONCO, T. et al. Key issues on future internet. In: *New Network Architectures*. [S.l.]: Springer Berlin / Heidelberg, 2010, (Studies in Computational Intelligence, v. 297). p. 221–236.

VIEIRA, A. et al. SimplyRep: A simple and effective reputation system to fight pollution in P2P live streaming. *Computer Networks*, v. 57, n. 4, p. 1019–1036, 2013. ISSN 1389-1286.

WANG, J. A. et al. Resisting free-riding behavior in bittorrent. *Future Generation Computer Systems-the International Journal of Grid Computing-Theory Methods and Applications*, v. 26, n. 8, p. 1285–1299, 2010. ISSN 0167-739X. Disponível em: <<Go to ISI>://WOS:000281508700022>.

WHITE, A. *Methods and apparatus for diagnosing and correcting faults in computers by a support agent at a remote location*. Google Patents, abr. 2 2002. US Patent 6,367,035. Disponível em: <http://www.google.com/patents/US6367035>.

WU, L. et al. Harnessing the power of bittorrent for distributed denial-of-service attacks. *Security and Communication Networks*, v. 4, n. 8, p. 860–870, 2011. ISSN 1939-0114. Disponível em: <<Go to ISI>://WOS:000293256600006>.

XIAO, Y. et al. Internet protocol television (IPTV): The killer application for the next-generation internet. *Communications Magazine, IEEE*, v. 45, n. 11, p. 126–134, 2007. ISSN 0163-6804.

____. Internet protocol television (iptv): the killer application for the next-generation internet. In: INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. [S.l.], 2007.

XIE, H. et al. *P4P: Explicit Communications for Cooperative Control Between P2P and Network Providers*. [S.l.], 2008. 7 p. Disponível em: <http://www.dcia.info-/documents/P4P_Overview.pdf>.

YANG, S. et al. The content pollution in peer-to-peer live streaming systems: Analysis and implications. In: *Parallel Processing, 2008. ICPP '08. 37th International Conference on*. [S.l.: s.n.], 2008. p. 652–659. ISSN 0190-3918.

YANG, W.; ABU-GHAZALEH, N. Gps: A general peer-to-peer simulator and its use for modeling bittorrent. In: IEEE. *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2005. 13th IEEE International Symposium on*. [S.l.], 2005. p. 425–432.

YIN, H. et al. Design and deployment of a hybrid CDN-P2P system for live video streaming: experiences with LiveSky. In: *Proc. of the 17th ACM international conference on Multimedia (MM'09)*. New York, NY, USA: ACM, 2009. p. 25–34. ISBN 978-1-60558-608-3.

YU, H.; BUFORD, J.; MERABTI, M. Improving messaging security in structured p2p overlay networks. In: *IEEE International Conference on Multimedia and Expo*. [S.l.: s.n.], 2007. p. 408–411.

ZEADALLY, S.; MOUSTAFA, H.; SIDDIQUI, F. Internet protocol television (iptv): architecture, trends, and challenges. *Systems Journal, IEEE*, IEEE, v. 5, n. 4, p. 518–527, 2011.

ZHANG, X. W.; CHEN, S. Q.; SANDHU, R. Enhancing data authenticity and integrity in p2p systems. *Ieee Internet Computing*, v. 9, n. 6, p. 42–49, 2005. ISSN 1089-7801. Disponível em: <<Go to ISI>://WOS:000233118000007>.

ZILBERMAN, N. et al. Reconfigurable network systems and software-defined networking. IEEE, 2015.