Análise do Impacto de Ataques de Poluição Combinado com *Whitewashing* em Sistemas P2P de *Live Streaming*

Rafael Barra de Almeida, Alex Borges Vieira, Ana Paula Couto Silva

[1]Instituto de Ciências Exatas – Departamento de Ciência da Computação
Universidade Federal de Juiz de Fora (UFJF)
Caixa Postal 36.036-900 – Juiz de Fora – MG – Brazil

rafael.barra@ice.ufjf.br,{alex.borges,anapaula.silva}@ufjf.edu.br

Abstract. *P2P live streaming has become a popular method of transmitting live content. Due to its popularity, it may be the target of attacks and opportunistic users behavior. These attacks can lower transmission quality or even make users abandon the system. In this article, we address attacks in a P2P system. More precisely, we analyze a reputation system in a mesh based P2P system under a pollution attack. During the attack, malicious users collude and also change their identities (whitewashing) often trying to cheat the reputation system. We have tested the reputation system on PlanetLab, which confirms it efficiency during an attack without whitewashing. Ours results are twofold: first we show that just verifying package integrity is not an effective protection. In this case, peers have to ask again for the polluted data, generating a significant data transmission overhead. Second, a simple reputation mechanism is effective against pollution attack and peers collusion, but when malicious peers do whitewashing, the mechanism becomes ineffective. In this case, we note up to 180% of overhead due pollution and retransmission.*

Resumo. *Transmissões ao vivo em P2P tornaram-se uma maneira popular para transmissão de eventos ao vivo. Dada essa popularidade, esses sistemas podem ser alvo de ataques e comportamentos oportunistas. Esses ataques podem diminuir a qualidade da transmissão e até mesmo, fazer com que os usuários abandonem o sistema. Neste artigo são tratados ataques a sistemas de transmissão ao vivo em P2P. Mais precisamente, é analisado um sistema de reputação em uma rede P2P baseada em mesh que sofre um ataque de poluição. Durante o ataque, participantes maliciosos conluem e também trocam de identidade (whitewashing) para tentar enganar o sistema de reputação. O sistema de reputação foi testado no PlanetLab e os resultados confirmam sua eficiência em ataques sem whitewashing. Os resultados apresentados se dividem em duas vertentes: primeiro, simplesmente checar os dados e pedir retransmissão não é uma medida eficiente de proteção. Nesse caso, os participantes irão experimentar uma sobrecarga significativa devido às retransmissões. Segundo, o sistema de reputação simples é eficiente contra ataques de poluição e conluio, mas quando há whitewashing, ele se torna ineficiente. Nesse caso, nota-se uma sobrecarga de até 180% devido aos dados poluídos e retransmissões.*

Palavras Chave: P2P Streaming, segurança, ataque, sobrecarga

1. Introdução

Transmissões de vídeo ao vivo na Internet são uma das aplicações mais importantes atualmente [Haizhou et al. 2011]. Grandes emissoras de TV pelo mundo têm investido nesse tipo de aplicação. Por exemplo, a CNN utilizou uma plataforma P2P para auxiliar as redes de distribuição de conteúdo durante a transmissão do discurso de posse do presidente Barack Obama. Esse evento, tido com um dos maiores de transmissão de vídeo na história da Internet, contou com 1.3 milhões de usuários simultâneos dos quais mais da metade dependeu da estrutura P2P [The New York Times 2009].

Tradicionalmente, a arquitetura mais utilizada para distribuição de vídeo é a cliente-servidor. Porém, essa abordagem sofre com a grande necessidade de recursos para lidar com os dados gerados por esse tipo de aplicação. Nesse contexto, a arquitetura P2P (*peer-to-peer* ou ponto-a-ponto) parece ser a mais adequada, pois permite lidar com falhas de rede e aumento da quantidade de usuários sem a necessidade de recursos elevados.

Vários sistemas P2P para transmissão de vídeo ao vivo organizados em malha têm sido implantados com sucesso e são os mais utilizados atualmente [Hei et al. 2007, Hei et al. 2008]. Porém, os participantes desses sistemas se aproveitam de falhas para realizar vários tipos de ataques ao mesmo. Por isso é importante que haja mecanismos que forneçam confiabilidade na transmissão de dados entre os *peers*. Atualmente, um dos mecanismos mais usados para computar confiabilidade é a reputação. Com esse mecanismo os sistemas computam a confiabilidade de maneira distribuída, levando em conta a experiência de cada *peer* e/ou a experiência global do sistema [Marti and Garcia-Molina 2006].

Entre os ataques e comportamentos indesejados se destacam [Seibert et al. 2010]:

1. **Poluição de conteúdo**: quando *peers* maliciosos alteram parte do conteúdo de um vídeo ao transmiti-lo aos seus vizinhos.
2. **Egoísmo (*Free-Riding*)**: quando um *peer* malicioso recebe dados dos seus vizinhos mas não os encaminham para os demais participantes, não contribuindo com o sistema.
3. **Descarte de Dados (*Data dropping*)**: um participante atrai vários parceiros com promessas de servi-los, porém, nega as requisições.
4. ***Whitewashing***: quando um *peer* sai e entra repetidamente no sistema usando uma nova identidade para evitar sofrer as penalidades devido a mal comportamento.

Algumas tentativas foram bem sucedidas ao combater os tipos de ataques citados anteriormente [Borges et al. 2008], [Seibert et al. 2010]. Porém, essas soluções ainda podem falhar quando o nó malicioso faz *whitewashing*. Devido à facilidade de se conseguir nova identificação [Friedman and Resnick 1999] e à grande dificuldade de se identificar um *whitewasher*, esse comportamento se torna difícil de lidar em sistemas P2P baseados em reputação [Chen et al. 2009]. Poucas soluções para esse comportamento são encontradas na literatura. Algumas serão discutidas na seção 2.

Nesse trabalho, analisa-se o impacto de ataques de poluição quando combinado com *whitewashing* em sistemas P2P de streaming. Foi desenvolvido um protótipo de aplicação que foi executado no PlanetLab, o que permitiu uma avaliação do sistema em um ambiente real. O sistema foi testado em três cenários diferentes. Inicialmente foi verificado o comportamento do sistema na presença de participantes maliciosos que

disseminam poluição durante toda a transmissão. Com esse cenário é possível ver os danos causados pelo ataque de poluição ao sistema. No segundo cenário foi inserido o mecanismo de defesa, baseado em reputação local. Nesse cenário, é avaliada a eficiência desse mecanismo em isolar os parceiros maliciosos e conter a disseminação de conteúdo poluído. Em um terceiro cenário os poluidores vão também fazer *whitewashing*. Dessa maneira será possível ver se esse comportamento realmente causa danos ao sistema.

Resultados mostram que com apenas 10% dos participantes maliciosos o sistema chega a apresentar 96% de dados poluídos e podem gerar uma sobrecarga de aproximadamente 230% na rede. Com o mecanismo de defesa ativo a poluição e a sobrecarga são reduzidas a menos de 4%. Porém, quando os participantes maliciosos fazem *whitewashing* o mecanismo de defesa adotado se torna ineficaz. Nesse caso a poluição observada no sistema chega a 84% e a retransmissão gerada a mais de 100% após 1 hora de execução.

O restante desse trabalho é organizado da seguinte forma: a Seção 3 mostra como a análise do sistema foi feita, apresentando o funcionamento do sistema, o mecanismo de defesa utilizado, a forma como é feito o *whitewashing* e os cenários testados. A Seção 5 mostra através de gráficos os valores obtidos dos experimentos.

2. Trabalhos relacionados

Na tentativa de combater ataques de poluição e *whitewashing* em sistemas P2P de *streaming* duas principais abordagens tem sido implementadas:

A primeira faz uso de uma entidade centralizada de confiança, que é responsável por associar fortes identidades ligadas a informações únicas dos *peers* recém-chegados no sistema. Essa abordagem, além de introduzir um ponto único de falha, ainda vai de encontro à natureza descentralizada dos sistemas P2P [Oualha and Roudier 2009]. A segunda abordagem impõe uma penalidade a todos os recém-chegados no sistema. Contudo, foi mostrado em [Feldman et al. 2006] que essa abordagem pode afetar negativamente a escalabilidade do sistema, já que *whitewashing* é um comportamento que não pode ser observado e, sendo assim, a penalidade é aplicada a todos os *peers* recém-chegados. Além disso, o desempenho do sistema também seria reduzido no caso em que houvesse muita rotatividade dos *peers*.

Em [Seibert et al. 2010] os autores detalham alguns dos compromissos feitos implicitamente pelos participantes enquanto trocam dados. Mostram que, quando esses compromissos não são aplicados de forma explícita, eles podem ser explorados por nós maliciosos para conduzir ataques ao sistema. É proposto ainda um mecanismo baseado em reputação que leva em conta a experiência individual de cada participante e o depoimento global do sistema para combater ataques de *data dropping* em sistemas P2P de *streaming*. Por marcar todos os novos nós como suspeitos, reduzindo os recursos disponíveis a esses nós, os autores acreditam que tal mecanismo também desencoraje os participantes a praticar *whitewashing*. Em [Chen et al. 2009] é apresentado um modelo que tenta capturar as diferenças de comportamento entre *whitewashers* e os demais *peers* baseando-se na observação do sistema no tempo. Os autores afirmam que cada *peer* deve manter uma sequência de ações similares mesmo quando saem e entram novamente no sistema. Sendo assim poderiam ser identificados e toda tentativa de conexão com algum outro *peer* do sistema seria negada. Em [Haizhou et al. 2011],

com o apoio de um *crawler* foi feita uma avaliação do impacto da poluição no PPLive levando em consideração a evolução dinâmica dos participantes assistindo a um canal, o tempo de vida dos usuários e a frequência de chegada e saída de *peers* no sistema. Os autores mostraram que um simples poluidor é capaz de comprometer todo sistema. Em [Lin et al. 2010], com o auxílio de um simulador, os autores mostram que o impacto e a efetividade dos ataques de poluição não são dependentes do tamanho da rede, mas sim da estabilidade e da disponibilidade de largura de banda dos poluidores e da origem. Em [Dhungel et al. 2007], é apresentado um experimento, no qual um poluidor é inserido em um sistema real. Os resultados obtidos mostram que ataques de poluição podem comprometer um sistema P2P de transmissão de vídeo ao vivo. Além disso, são sugeridas algumas técnicas possíveis para verificar a integridade do fluxo de mídia distribuído.

3. Metodologia e Cenários de Experimentação

Foi desenvolvido um protótipo de aplicação baseado em sistemas P2P organizados em malha que foi executado no PlanetLab com uma abordagem semelhante a do SopCast[1], onde os nós são organizados em malha e cada um requisita ou envia dados a seus parceiros. Essa abordagem permite ao sistema ser escalável e resistente a falhas [Hei et al. 2008]. Nesse tipo de sistema cada nó faz *broadcast* periodicamente informando a seus vizinhos quais *chunks* possui. Ao receber uma requisição alguns participantes maliciosos podem alterar propositalmente o conteúdo dos *chunks* antes de enviá-los a seus parceiros, fazendo com que estes recebam dados que não correspondem aos requisitados. Dessa maneira se configura o ataque de poluição. Na ocorrência desse tipo de ataque, os nós "honestos", caso não identifiquem esses dados como poluídos, podem encaminhá-los inocentemente aos seus parceiros, contribuindo assim para a disseminação do conteúdo poluído. Esse cenário pode reduzir drasticamente o desempenho do sistema.

Nesse trabalho não é diferenciada a poluição de conteúdo gerada intencionalmente da poluição gerada por acidente, como por exemplo, quando um participante tem problemas de transmissão que corrompem os dados ou quando um pacote chega atrasado, tudo isso é considerado poluição. Com isso é possível abranger mais situações reais de uma rede, ao mesmo tempo em que evita usuários com problemas de rede que comprometem suas transmissões.

O sistema simulado contém um nó especial chamado origem ou servidor, que é o responsável por gerar o *streaming* de vídeo e dividi-lo em *chunks*. Quando um *peer* entra no sistema ele inicialmente contacta uma unidade centralizada chamada *bootstrap*. Este, por sua vez, responde ao recém-chegado com uma lista de participantes que tem interesse no mesmo *streaming* de vídeo, além da posição atual do vídeo na origem. O *peer* recém-chegado irá então tentar se conectar com os *peers* presentes na lista recebida do *bootsrap* para formar sua lista de vizinhos. A partir desse momento o *peer* pode fazer download e upload de acordo com seus interesses.

O desempenho do sistema será medido em função da quantidade de *chunks* poluídos presentes e pela retransmissão gerada pela necessidade de novas requisições ao receber dados poluídos.

[1]Uma das aplicações mais populares de P2P streaming

Foram realizados testes com 120 nós do PlanetLab, todos entrando no sistema ao mesmo tempo. Cada teste durou 60 minutos. O servidor de vídeo e o *bootstrap* estão em uma máquina dedicada na rede do campus da UFJF. Cada participante se conecta a 18 vizinhos escolhidos aleatoriamente a partir da lista recebida do *bootstrap* (chamados de ativos). Caso o tamanho da lista seja maior que o número máximo de vizinhos, os excedentes ficam em uma lista de que pode ser usada caso algum *peer* da lista de ativos falhe. A cada segundo os participantes enviam uma mensagem de verificação para seus vizinhos e para o *bootstrap*. Essa mensagem é usada para indicar que o participante ainda está "vivo" além de carregar o seu mapa de chunks. O *bootstrap* responde a essas mensagens com uma nova lista de *peers* e a posição atual do vídeo na origem.

Foi incluído um marcador no cabeçalho dos pacotes poluídos para que seja possível identificá-los quando são transmitidos através da rede. Em um cenário real, qualquer esquema de verificação de dados proposto (e.g. assinatura baseada em *hash*) [Haridasan and van Renesse 2007] poderia ser usado para identificar automaticamente os *chunks* poluídos. Assim, nesse trabalho, é ignorado qualquer sobrecarga que possa ser gerada por essas de técnicas de verificação, concentrando-se na sobrecarga causada pela retransmissão de *chunks* devido a recepção de conteúdo poluído.

Foi verificado um sistema onde existem 10% de nós maliciosos que disseminam dados poluídos durante todo o tempo de vida, chamaremos este de cenário 1, onde não há nenhum mecanismo de defesa ativo e nem a pratica de *whitewashing* pelos poluidores. Nesse cenário, os poluidores forjam ter todos os *chunks* passando o seu mapa de *chunks* completo para seus parceiros. Quando recebem uma requisição de *chunk* enviam um dado que não condiz com o pedido. Com esse cenário é possível ver o impacto do ataque de poluição no sistema.

Num segundo cenário verificamos o comportamento do sistema quando o mecanismo de defesa está ativo. Todos os participantes que não são poluidores calculam a reputação de seus parceiros usando o modelo de reputação local apresentado em 4. Quando um parceiro é identificado como poluidor ele é removido da lista de vizinhos e toda tentativa de conexão feita por esse parceiro é negada.

Por fim, verificamos as condições do sistema quando os participantes poluidores também praticam *whitewashing* periodicamente. Nesse cenário o sistema de defesa está sempre ativo. Com isso é possível verificar se o do modelo de reputação local ainda é capaz de combater efetivamente o ataque de poluição quando existem poluidores que fazem *whitewashing*. Em todos os casos existem 10% de nós maliciosos que vão poluir durante todo o tempo de vida do sistema.

4. Mecanismo de Defesa: Reputação

Nesse trabalho é utilizado um modelo de reputação local, onde cada participante monitora a troca de dados com cada parceiro para computar sua reputação. O objetivo é permitir que cada participante identifique os parceiros que disseminam conteúdo poluído e possa isolá-los. No modelo de reputação global, os participantes calculam a reputação de seus parceiros baseados em dois componentes: a experiência individual e o depoimento da rede [Borges et al. 2008]. Não há certeza de que usar o depoimento da rede para computar a reputação de um parceiro tem um bom custo-benefício para aplicações P2P de *streaming*. Por algum motivo, em pequenos intervalos de tempo, um participante

pode se comportar de maneiras muito diferentes com diferentes parceiros, enviando dados poluídos para somente alguns deles. Sendo assim, esperar que a rede convirja para uma opinião consistente pode demorar muito. Dado que atualmente as taxas de transmissão ao vivo excedem 5 *chunks* por segundo, um *chunk* danificado ou poluído se espalha muito rápido em uma transmissão P2P ao vivo, impactando assim muito mais na qualidade da *streaming* do que em uma aplicação de compartilhamento de arquivos. Além disso, um participante p_i pode não ter muitos parceiros que compartilham uma parceria com um dado participante p_j. Assim, o depoimento da rede no p_j pode não ser confiável.

Assim, nesse trabalho será usado um mecanismo de reputação mais simples, descentralizado que se baseia somente na experiência individual que cada participante tem com seus parceiros para calcular a reputação. Mais precisamente, em intervalos regulares de tempo um participante p_i computa a reputação de cada parceiro p_j, $R_i[p_j]$, como sua experiência individual com p_j medida naquele intervalo de tempo, que é computado de acordo com a Equação 1. Durante cada intervalo de tempo p_i requer *r chunks* a p_j. O parceiro p_j pode prover *n* respostas ruins para p_i (onde $0 \leq n \leq r$). Aqui uma resposta ruim é qualquer uma que force p_i a pedir o dado novamente para outro parceiro na rede P2P. por exemplo, uma resposta com dado poluído ou uma não resposta de p_j. A razão n/r dá a qualidade da experiência de p_i com p_j.

Se a razão n/r está abaixo do limite T_i^{max}, p_i diminui a reputação local de p_j. Caso contrário, aumenta sua pontuação. A reputação de p_j em p_i, $R_i[p_j]$, é atualizada como:

$$R_i[p_j] = \begin{cases} max(0, R_i[p_j] - \alpha_{p_i} * (1 + n/r)^{y_i}) & \text{se } n/r > \mathrm{T}_i^{max} \\ min(1, R_i[p_j] + \alpha_{g_i} * (1 - n/r)) & \text{caso contrário} \end{cases} \quad (1)$$

onde α_{p_i} e α_{g_i} são fatores de penalidade e recompensa, respectivamente. Foi feito $\alpha_{p_i} \geq \alpha_{g_i}$ para identificar e penalizar rapidamente os poluidores. Dessa maneira, um participante é recompensado linearmente e punido exponencialmente.

Todo participante recém-chegado recebe uma reputação inicial e, além disso, todos os participantes tem uma reputação mínima R_i^{min} ($0 \leq R_i^{min} \leq 1$): se $R_i[p_j]$ cai abaixo de R_i^{min}, p_i remove p_j de sua lista de parceiros.

O principal problema de se basear somente na experiência individual para computar a reputação é que uma vez que o participante p_i considera o parceiro p_j como um poluidor devido a baixa reputação, p_i não terá mais a oportunidade de trocar dados com com p_j novamente. Isto é, do ponto de vista de p_i, p_j será banido e assim nunca será capaz de se reabilitar. No modelo de reputação global, o depoimento da rede poderia ajudar um participante a se reabilitar junto a um velho parceiro. Essa é uma propriedade interessante, já que alguns participantes podem encaminhar conteúdo poluído.

Por isso, é usado um esquema para permitir reabilitação de participantes que se baseia somente na experiência individual. A ideia é mudar dinamicamente a reputação mínima de cada participante p_i em reação às condições da rede, de acordo com o percebido por cada participante. Se p_i sente que a rede está sob ataque, ele aumenta a reputação mínima R_i^{min} assim penalizando seus maus parceiros mais rapidamente. Caso contrário, ele diminui R_i^{min} para permitir parcerias com participantes previamente punidos. Isso é feito independentemente por todos os participantes.

Mais precisamente, são definidos dois estados do sistema, chamados *calmaria* e *tempestade*. O sistema está no estado calmaria na perspectiva do participante p_i, se p_i acredita que não existe poluidor no sistema. Caso contrário, na perspectiva de p_i, o sistema está em tempestade. A cada intervalo de tempo, p_i verifica o estado do sistema e atualiza R_i^{min} de acordo com a Equação 2. Se o sistema está em calmaria p_i diminui seu limite local R_i^{min} por γ_{g_i}; caso contrário R_i^{min} é aumentado por γ_{p_i}. Foi feito $\gamma_{p_i} > \gamma_{g_i}$ de modo a reagir mais rápido a poluidores. os limites RT_i^{min} e RT_i^{max} foram definidos de maneira que $0 \leq RT_i^{min} \leq R_i^{min} \leq RT_i^{max} \leq 1$.

$$R_i^{min} = \begin{cases} max(RT_i^{max}, R_i^{min} + \gamma_{p_i}) & \text{se } \textit{estado tempestade} \\ min(RT_i^{min}, R_i^{min} - \gamma_{g_i}) & \text{se } \textit{estado de calmaria} \end{cases} \quad (2)$$

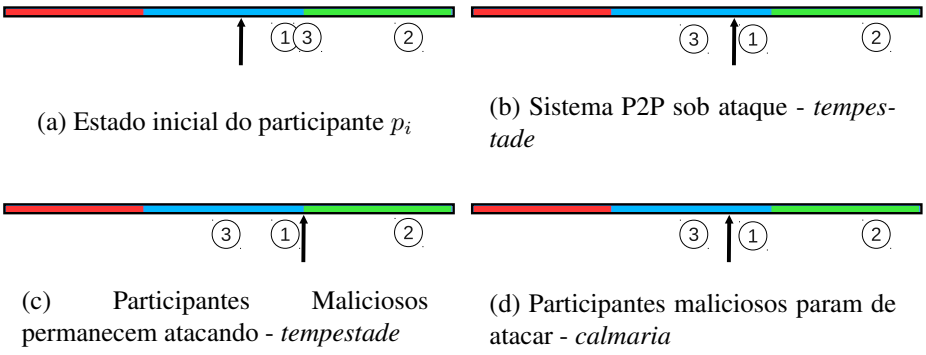
O estado do sistema é definido a partir de cada participante, baseado somente nas suas experiências com seus parceiros. Isto é, se o participante p_i recebe qualquer dado poluído de algum parceiro ele suspeita que o sistema está sob ataque de poluição: sua visão local do estado do sistema é marcada como em tempestade, e ele reage a isso ajustando o limite R_i^{min} de acordo. De maneira similar, se todos os *chunks* recebidos são legítimos, p_i percebe o sistema como em um estado de calmaria e assim reduz R_i^{min}.

A Figura 1 mostra como um participante reage a mudanças no estado do sistema. Na figura, a barra representa a escala de reputação de valores baixos (esquerda) para altos (direita). A seta representa o valor atual de R_i^{min} enquanto que os números 1, 2 e 3, indicam a reputação atual em p_i de três parceiros (referenciados como p_1, p_2 e p_3). Note que, na Figura 4, todos os parceiros estão localizados à direita da seta, indicando que suas reputações estão acima do limite mínimo. Assim que p_i recebe qualquer *chunk* poluído, ele coloca o sistema no estado de tempestade, aumentando seu limite mínimo de reputação, como mostrado na Figura 4. Já que a reputação local de p_3 cai abaixo do mínimo, p_i entende que p_3 é um poluidor, o remove de sua lista de parceiros e pára de interagir com ele. Depois disso, p_i continua recebendo dados poluídos e, novamente, aumenta R_i^{min}, como mostrado na Figura 4. Agora, p_1 também é também é removido da lista de parceiros de p_i, já que sua reputação está abaixo do limite mínimo. Após isso, p_i pára de receber dados poluídos, assim muda o estado do sistema para calmaria e diminui R_i^{min}. O novo valor de R_i^{min} permite p_i aceitar p_1 como parceiro novamente. Note que, mudanças na R_i^{min}, de acordo com a Equação 2, são realizadas independentemente das mudanças nas reputações locais de cada parceiro, definida na Equação 1.

5. Resultados

Os resultados apresentados têm como parâmetros os valores mostrados na Tabela 1. Em todos os experimentos realizados os participantes poluidores permanecem no sistema poluindo durante todo o tempo. Todos os participantes entram no sistema ao mesmo tempo no início do experimento. Para sincronizar essa entrada de nós, foi usado o CRON do Linux. Os resultados foram coletados a cada 5 minutos e os valores apresentados são médias de 5 execuções no Planet Lab de cada cenário.

Inicialmente são apresentados resultados para o cenário onde só existe poluição e nenhum esquema de defesa. Nesse cenário, é possível observar o impacto de ataques de poluição em um sistema real. A Figura 2 apresenta a proporção de dados



(a) Estado inicial do participante p_i

(b) Sistema P2P sob ataque - *tempestade*

(c) Participantes Maliciosos permanecem atacando - *tempestade*

(d) Participantes maliciosos param de atacar - *calmaria*

Figura 1. Limite Dinâmico da Reputação do Participante p_i (seta indica R_i^{min}, 1, 2 e 3 são parceiros de p_i e suas posições na barra indicam suas atuais reputações com p_i).

Tabela 1. Parâmetros de Execução

Parâmetro	Valor
Número de Participantes	120
Tempo de Execução	1 hora
Limite (n/r)	0.5
Reputação Inicial (n/r)	0.7
Reputação Mínima	0.4
Y	2
α_g	0.3
α_p	0.6

poluídos percebidos pelos usuários da rede P2P; e a sobrecarga gerada pelos pedidos de retransmissão. Conforme pode ser observado, com apenas 10% de poluidores, a proporção de dados poluídos no sistema chega a mais de 96%. A retransmissão observada ficou próxima de 230%. Assim, em um sistema sem nenhum mecanismo de defesa, seria necessário mais que o triplo da banda de rede necessária em um cenário sem ataques.

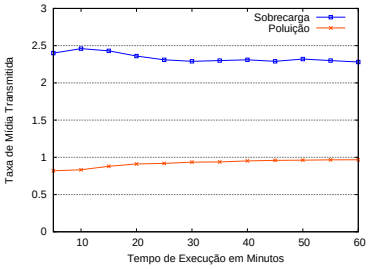


Figura 2. Cenário sem nenhum mecanismo de defesa contra ataques.

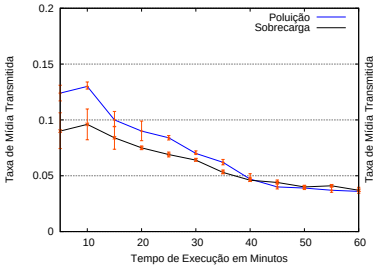


Figura 3. Cenário com reputação implementado e sem ataque de *whitewashing*.

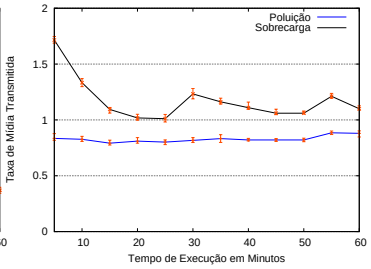


Figura 4. Cenário com reputação implementado e com ataque de *whitewashing*

A Figura 3 apresenta os resultados do segundo cenário implementado, onde o há um mecanismo de defesa. O objetivo desse cenário é verificar se o sistema de reputação é capaz de isolar os participantes maliciosos do sistema, e assim, conter a disseminação de conteúdo poluído. Os resultados encontrados mostram que a proporção de dados poluídos percebidos no sistema P2P e a sobrecarga gerada pelas novas requisições de *chunks* foram

reduzidas a menos de 4%. Mesmo sob influência de variáveis externas como atrasos na rede entre os participantes do PlanetLab, esses resultados demonstram que o sistema de reputação é eficiente sob ataques de poluição e conluio dos atacantes. Neste gráfico a escala foi reduzida para um melhor entendimento do mesmo. Além disso, foi inserido o intervalo de confiança para cada valor para um grau de confiança de 95%. É possível ver que os intervalos são bastante pequenos.

Os resultados do terceiro cenário, onde os poluidores fazem *whitewashing*, são apresentados na Figura 4, juntamente com os intervalos de confiança para cada valor para um grau de confiança de 95%. Nesse cenário, o sistema de reputação sofre com as constantes trocas de identidades. Há uma alta proporção de *chunks* poluídos e retransmissão. Apesar da ineficiência do mecanismo de reputação, com cerca de 110% de sobrecarga em média, ele ainda apresenta resultados consideráveis se comparado a um sistema sem nenhuma proteção.

Concluindo, os resultados mostram que o mecanismo de defesa baseado em reputação local é bastante eficaz ao combater a poluição e conluio no sistema de transmissão ao vivo P2P. Porém, quando os participantes poluidores fazem também *whitewashing* esse mecanismo de defesa se torna ineficaz.

6. Conclusões e Trabalhos Futuros

Este trabalho analisa o impacto de ataques de poluição quando combinado com *whitewashing* em sistemas P2P de streaming. Foi desenvolvido um protótipo de aplicação, executado no PlanetLab, que permite a avaliação do sistema em um ambiente real.

Os resultados mostram que ataques são um problema de fato. A maior parte dos sistemas comerciais não apresenta mecanismos de proteção e nesse caso, um pequeno número de participantes maliciosos podem conseguir um grande efeito em seus ataques. Os experimentos realizados mostram que, com apenas 10% dos participantes maliciosos o sistema de transmissão ao vivo em P2P chega a apresentar 96% de dados poluídos e podem gerar uma retransmissão de aproximadamente 230% na rede. Ou seja, nesse caso, os participantes do sistema P2P devem ter mais de 3 vezes a banda de rede necessária para um sistema sem ataques.

O sistema de reputação analisado consegue mitigar os efeitos de ataque de poluição, mesmo quando há conluio entre os atacantes. Nesse caso, a sobrecarga observada é inferior a 4% da mídia original. Porém, quando os participantes maliciosos realizam *whistewashing*, o mecanismo de defesa torna-se ineficaz. Nesse caso a poluição observada no sistema chega a 84% e a sobrecarga gerada a mais de 100%.

Como trabalho futuro, destaca-se a criação de mecanismos auxiliares para combater os efeitos gerados pelo ataque de *whistewashing*. Nesse caso, deseja-se criar medidas simples e eficientes pois a principal solução adotada na literatura, a criação de mecanismos de identificação confiáveis, por vezes é complicada e custosa. Nesse sentido, serão exploradas maneiras de criar a reputação inicial de um usuário, baseado no seu histórico de parcerias. Essa modificação poderá iniciar participantes confiáveis com boa reputação e dar aos novatos uma reputação mínima. Assim, ao iniciar qualquer tentativa de ataque, os poluidores serão prontamente penalizados e bloqueados do sistema P2P.

Referências

Borges, A., Almeida, J., and Campos, S. (2008). Fighting pollution in p2p live streaming systems. In *Multimedia and Expo, 2008 IEEE International Conference on*.

Chen, J., Lu, H., and Bruda, S. (2009). A solution for whitewashing in p2p systems based on observation preorder. In *2009 International Conference on Networks Security, Wireless Communications and Trusted Computing*, pages 547–550. IEEE.

Dhungel, P., Hei, X., Ross, K., and Saxena, N. (2007). The pollution attack in p2p live video streaming: measurement results and defenses. In *Proceedings of the 2007 workshop on Peer-to-peer streaming and IP-TV*, pages 323–328. ACM.

Feldman, M., Papadimitriou, C., Chuang, J., and Stoica, I. (2006). Free-riding and whitewashing in peer-to-peer systems. *Selected Areas in Communications, IEEE Journal on*, 24(5):1010–1019.

Friedman, E. and Resnick, P. (1999). The social cost of cheap pseudonyms. *Ann Arbor*, 1001:48109–1092.

Haizhou, W., Xingshu, C., and Wenxian, W. (2011). A measurement study of polluting a large-scale p2p iptv system. In *College of Computer Science, Sichuan University, Chengdu 610065, P. R. China*.

Haridasan, M. and van Renesse, R. (2007). Securestream: An intrusion-tolerant protocol for live-streaming dissemination. In *Journal of Computer Communications. Special issue on Foundation of Peer-to-Peer Computing*. Elsevier.

Hei, X., Liang, C., Liang, J., Liu, Y., and Ross, K. (2007). A measurement study of a large-scale p2p iptv system. *Multimedia, IEEE Transactions on*, 9(8):1672–1687.

Hei, X., Liu, Y., and Ross, K. (2008). Iptv over p2p streaming networks: the mesh-pull approach. *Communications Magazine, IEEE*, 46(2):86–92.

Lin, E., de Castro, D., Wang, M., and Aycock, J. (2010). Spoim: A close look at pollution attacks in p2p live streaming. In *Quality of Service (IWQoS), 2010 18th International Workshop on*, pages 1–9. IEEE.

Marti, S. and Garcia-Molina, H. (2006). Taxonomy of trust: Categorizing p2p reputation systems. *Computer Networks*, 50(4):472–484.

Oualha, N. and Roudier, Y. (2009). A game theoretical approach in securing p2p storage against whitewashers. In *Enabling Technologies: Infrastructures for Collaborative Enterprises, 2009. WETICE'09. 18th IEEE International Workshops on*.

Seibert, J., Sun, X., Nita-Rotaru, C., and Rao, S. (2010). Towards securing data delivery in peer-to-peer streaming. In *Communication Systems and Networks (COMSNETS), 2010 Second International Conference on*, pages 1–10. IEEE.

The New York Times (2009). CNN: Inauguration P2P Stream a Success, Despite Backlash.