Análise e Contra-Ataque à Poluição e *Whitewashing* em Sistemas P2P de Vídeo ao Vivo.

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

[1]Departamento de Ciência da Computação
Universidade Federal de Juiz de Fora – Juiz de Fora – MG

rafael.barra@ice.ufjf.br

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

Área de Pesquisa: Redes de Computadores
Ano de Ingresso no Programa: 2011

Resumo. *Este artigo analisa o impacto causado por ataques de poluição e whitewashing em sistemas P2P de vídeo ao vivo. Adicionalmente, mecanismos simples e descentralizados de reputação são propostos e implementados em um ambiente de rede real, configurado no PlanetLab. Esse trabalho mostra que ataques de poluição são prejudiciais mesmo em um sistema com um pequeno número de poluidores. Nesse caso, o sistema P2P pode sofrer uma sobrecarga de até 400% com relação a taxa da mídia original. Os mecanismos de reputação propostos e implementados no ambiente do PlanetLab rapidamente bloqueiam ataques de poluição e whitewashing. Para o caso de ataque combinado de poluição e whitewashing, o mecanismo de reputação diminui a sobrecarga no sistema a* 20% *e a perda de chunks em* 3%*.*

Palavras Chave: P2P Streaming, segurança, ataque de poluição, *whitewashing*

1. Introdução

Nos últimos anos, aplicações de vídeo ao vivo sobre a Internet tem atraído a atenção de usuários e pesquisadores na área de redes. A princípio baseadas na arquitetura cliente-servidor, atualmente estas aplicações utilizam a arquitetura P2P (*peer-to-peer*) que é mais resiliente a falhas e mais escalável, sem requerer altos recursos de infraestrutura. Como exemplo da importância destas aplicações, a CNN utilizou uma plataforma P2P para auxiliar na distribuição do conteúdo da posse do presidente Obama em janeiro de 2009. Este evento, considerado um dos maiores na história da Internet, atendeu 1.3 milhões de usuários ao mesmo tempo. Mais da metade dos usuários utilizavam a estrutura P2P para acompanhar a transmissão do evento[1].

A maioria das aplicações P2P de vídeo ao vivo organizam seus *peers* em uma rede sobreposta em forma de malha [Hei et al. 2007]. Nestes sistemas, um *peer* especial codifica o vídeo e os demais requisitam (ou trocam) pedaços de informação para (entre) seus parceiros. Estes pedaços são denominados *chunks*. Durante as trocas de *chunks*, os *peers* podem se comportar de uma maneira maliciosa e oportunista; participantes maliciosos podem tirar vantagem de falhas no sistema visando realizar ataques.

O principal foco deste trabalho é analisar sistemas P2P de vídeo ao vivo em casos onde existem participantes maliciosos infiltrados na troca de *chunks* entre os *peers* legítimos que formam a rede sobreposta. Neste artigo destacam-se os seguintes ataques e comportamentos indesejados [Seibert et al. 2010]:(1) *Poluição de Conteúdo*: participante malicioso altera o conteúdo de um vídeo antes de transmití-lo aos seus vizinhos e; (2) ***Whitewashing***: participante sai e entra no sistema, repetidamente, com uma nova identidade visando evitar penalidades devido a sua má reputação.

Na literatura são encontrados vários esquemas de combate a ataques específicos a sistemas P2P de vídeo ao vivo [Seibert et al. 2010, Borges et al. 2008, Vieira et al. 2009]. No entanto, estas soluções falham em situações onde participantes maliciosos mudam frequentemente suas identidades, ou seja, em cenários com *whitewashing*. A facilidade de se obter uma nova identidade e a dificuldade de caracterizar um *whitewasher* fazem com que este comportamento se torne um desafio.

Neste artigo é analisado o impacto causado por ataques de poluição e *whitewashing* em sistemas P2P de vídeo ao vivo. As principais contribuições são: (1) Através de testes em um ambiente real, esse trabalho mostra que ataques de poluição são prejudiciais mesmo em um sistema com um pequeno número de poluidores. Na presença de um ataque, os participantes podem receber dados poluídos. Mesmo nos casos em que participantes identificam corretamente todos os dados poluídos, os poluidores ainda permanecem atacando o sistema; (2) foi desenvolvido um protótipo de aplicação P2P de vídeo ao vivo com mecanismos de reputação que combatem ataques de poluição e *whitewashing*. Com o objetivo de avaliar os mecanismos propostos, o sistema implementado é testado em um ambiente rede real, em diversas máquinas do PlanetLab [2]. O esquema de reputação proposto rapidamente bloqueia ataques de poluição e *whitewashing*: a sobrecarga no sistema é menor que 20% e a perda menor que 3% quando o mecanismo de reputação é implementado. *Peers* que são identificados

[1] http:www.nytimes.com/external/gigaom/2009/02/07/07gigaom-cnn-inauguration-p2p-stream-a-success-despite-bac-17849.html

[2] http://www.planet-lab.org/

erradamente como poluidores, os *falsos positivos*, se recuperam rapidamente a sua verdadeira reputação. Concluindo, a qualidade de experiência do usuário permanece elevada, mesmo com a aplicação do mecanismo de reputação.

2. Trabalhos Relacionados

Considerando o ataque de poluição, [Hu and Zhao 2010] propõem um esquema que detecta este tipo de ataque, identificando os *peers* atacantes. *Chunks* poluídos são detectados o mais rápido possível. Para identificar os poluidores, é proposto um gerenciamento de confiança. Embora este esquema auxilie na redução de sobrecarga no sistema, é necessária a retransmissão de dados. Em [Lin et al. 2010], os autores mostram, através de resultados de simulação, que o impacto e a eficiência de ataques de poluição independem do número total de *peers* participantes da distribuição de conteúdo. Com a utilização de um *crawler*, os autores em [Haizhou et al. 2011] mostram que um único *peer* poluidor é capaz de comprometer o funcionamento de todo o sistema de transmissão de vídeo ao vivo.

Para os casos com ataque de *whitewashing*, [Oualha and Roudier 2009] propõem uma entidade centralizada responsável por identificar unicamente participantes recém chegados ao sistema. A desvantagem desta abordagem é a centralização de informações em um sistema de natureza distribuída. Feldman *et al.* [Feldman et al. 2006] mostram que esta abordagem afeta a escalabilidade do sistema.

Os autores [Seibert et al. 2010] propõem um mecanismo baseado em reputação para combater ataques de descarte de dados. Este mecanismo considera a experiência individual dos participantes da transmissão de conteúdo, combinada ao depoimento dos demais participantes. Marcando os *peers* recém-chegados como suspeitos e reduzindo os recursos disponíveis para estes *peers*, os autores afirmam que este mecanismo também pode ser estendido para combater o ataque de *whitewashing*. O artigo [Chen et al. 2009] apresenta um modelo que captura as diferenças de comportamento dos *peers whitewashers* e dos *peers* de comportamento colaborativo.

3. Mecanismo de Defesa baseado em Reputação

Neste artigo, propõe-se um modelo de reputação local em que cada *peer* monitora a troca de dados com seus parceiros. Dessa forma, os participantes do sistema são capazes de associar um valor de reputação a cada um de seus parceiros. O objetivo é permitir que cada *peer* identifique e isole parceiros que disseminem conteúdo poluído.

Em uma abordagem de reputação distribuída clássica, cada *peer* associa uma nota a cada parceiro. Essa nota é computada através da experiência individual entre o *peer* e seu respectivo parceiro, e pelo depoimento da rede sobre este parceiro [Borges et al. 2008]. Entretanto, não é claro que explorar o depoimento da rede resulte em um custo-benefício satisfatório para aplicações P2P de vídeo ao vivo.

Um *peer* em um sistema de transmissão ao vivo em P2P tipicamente apresenta muitas interações com seus parceiros em curtos intervalos de tempo. As trocas de dados ocorrem em uma escala de tempo muito menor que as aplicações de compartilhamento de arquivo. Assim, o depoimento da rede sobre um determinado *peer* pode convergir muito lentamente, se comparado a taxa de interações entre 2 *peers*. Finalmente, caso um

determinado p_i queira calcular a reputação de p_j e não tenha muitos parceiros que também sejam parceiros de p_j, o depoimento da rede em relação a p_j pode não ser confiável.

Para evitar estes possíveis problemas, neste artigo é proposto um mecanismo de reputação mais simples, descentralizado, baseado somente na experiência individual que cada *peer* tem em relação a seus parceiros. Este mecanismo será chamado *mecanismo de reputação simples*. Por esse novo mecanismo, cada *peer* p_i periodicamente calcula a reputação de cada parceiro p_j ($R_i[p_j]$).

Mais precisamente, de acordo com a Equação 1, durante cada intervalo de tempo, p_i requisita *r chunks* a p_j. O parceiro p_j pode prover *n* respostas ruins para p_i (onde $0 \leq n \leq r$). Uma resposta é definida como *ruim* quando p_i é forçado a pedir o *chunk* novamente a outro parceiro. A razão n/r representa a qualidade da experiência de p_i em relação a p_j. Se essa razão n/r tem valor acima de um limite T_i^{max}, p_i diminui a reputação local de p_j. Caso contrário, a reputação local de p_j é aumentada.

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

Os parâmetros α_{p_i} e α_{g_i} são respectivamente fatores de penalidade e gratificação. Com o objetivo de rapidamente identificar e penalizar *peers* poluidores, considera-se $\alpha_{p_i} \geq \alpha_{g_i}$. Todos os *peers* recém-chegados no sistema recebem um valor inicial de reputação. Cada *peer* p_i tem um limite de reputação mínima R_i^{min}, com $0 \leq R_i^{min} \leq 1$. Caso $R_i[p_j]$ seja menor que R_i^{min}, p_i remove p_j de sua lista de parceiros.

Por esse esquema de reputação, uma vez que p_i considera um parceiro p_j como poluidor, p_j não terá mais oportunidade de trocar dados com p_i. No modelo de reputação clássico, o depoimento da rede poderia ajudar p_j a se redimir e em um futuro, estar apto a trocar dados com seu parceiro p_i. Nesse sentido, a fim de permitir reabilitação, é proposto um mecanismo que dinamicamente altera o limite mínimo de reputação R_i^{min}.

A ideia chave para alterar R_i^{min} é fazer com que cada *peer* p_i reaja às condições da rede, percebidas através de suas medições locais. Por exemplo, caso p_i infira que a rede está sob ataque, este aumenta o valor de R_i^{min}, penalizando mais rapidamente seus prováveis parceiros poluidores. Caso contrário, este diminui o valor de R_i^{min} para permitir a reabilitação das parcerias punidas anteriormente.

Para mudança dinâmica do valor de R_i^{min}, dois estados são considerados: (1) *calmaria* e; (2) *tempestade*. Na visão de p_i, o sistema está em *calmaria* se percebe um nível de poluição abaixo de um limite pré-definido. Caso contrário, na visão de p_i, o sistema se encontra no estado de *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} no fator de γ_{g_i}; caso contrário, R_i^{min} é aumentado no fator γ_{p_i}. Para que o sistema identifique rapidamente participantes poluidores, $\gamma_{p_i} > \gamma_{g_i}$. Os limites RT_i^{min} e RT_i^{max} são estabelecidos de tal maneira que $0 \leq RT_i^{min} \leq R_i^{min} \leq RT_i^{max} \leq 1$.

$$R_i^{min} = \begin{cases} max(RT_i^{max}, R_i^{min} + \gamma_{p_i}) & \text{se o sistema está no } \textit{estado de tempestade} \\ min(RT_i^{min}, R_i^{min} - \gamma_{g_i}) & \text{se o sistema está no } \textit{estado de calmaria} \end{cases} \quad (2)$$

O estado do sistema é definido na perspectiva local de cada *peer*, baseado somente

nas experiências obtidas em relação a seus parceiros: caso p_i receba uma quantidade de dados poluídos de seus parceiros, a sua visão local é de que o sistema está sofrendo ataque de poluição.

No momento que p_i recebe uma taxa de *chunks* poluídos, o estado do sistema é modificado para *tempestade*, e seu limite mínimo de reputação é aumentado.

Combinado com ataque de poluição, os poluidores podem trocar suas identidades frequentemente. A troca constante de identidade, conhecida como *whitewashing*, tem por objetivo enganar o sistema de reputação. Ataques combinados podem causar grandes danos a qualidade do sistema P2P, e assim, propõe-se um *mecanismo de reputação modificado* (assim denominado para critérios de comparação).

Nesse sentido, pelo *mecanismo de reputação modificado*, participantes recém-chegados ao sistema recebem um baixo valor de reputação inicial. O valor estabelecido para o valor inicial é um valor próximo ao valor limite para que troca de *chunks* aconteçam, ou seja $R_i[p_j] \tilde{=} R_i^{min}$. Para qualquer tentativa de poluição, $R_i[p_j]$ ficará abaixo de R_i^{min} e então p_j será removido de sua lista de parceiros de p_i. Vale ressaltar que apesar da diminuição no valor de reputação na visão de p_i, *peers* recém-chegados podem trocar dados com os demais participantes do sistema.

4. Metodologia e Ambiente de Experimentção

Os mecanismos de reputação propostos (*simples* e *modificado*) foram avaliados em um ambiente real, configurado no PlanetLab. Uma aplicação P2P de vídeo ao vivo baseada em uma rede *mesh* [Hei et al. 2008] foi implementada. Nessa aplicação, foram incorporados os mecanismos de combate a ataques de poluição e *whitewashing*.

O servidor foi instalado numa máquina dedicada na rede do campus, transmitindo 30 minutos de vídeo a uma taxa de 120 kbps. Foram utilizados 133 nós PlanetLab como *peers* do sistema P2P de transmissão de vídeo ao vivo. Desses *peers*, 120 são classificados como *peers bons* e 13 como poluidores (10% do total).

Durante os experimentos, todos os *peers* permanecem ativos até o fim da transmissão. Cada *peer* se conecta no máximo a 18 parceiros. Se um de seus parceiros falha ou deixa o sistema, um *peer* deve requisitar novos parceiros candidatos ao *bootstrap* do sistema. Os poluidores atacam desde o momento em que entram no sistema até o final da transmissão. Durante o ataque, os *peers* poluidores anunciam um mapa de *chunks* completo, forjando ter todos os dados possíveis.

Para indicar um *chunk* como poluído, foi incluído um *bit* no cabeçalho do pacote. Desta forma, é possível identificar um *chunk* como poluído enquanto ele trafega pela rede. Qualquer esquema de verificação de dados (e.g., assinatura baseada em *hash* proposto em [Haridasan and Renesse 2008]) pode ser usado para identificar automaticamente *chunks* poluídos. Sendo assim, neste artigo foi desconsiderado qualquer sobrecarga que possa ser introduzida por essas técnicas de verificação de dados, focando apenas na sobrecarga causada pela retransmissão de *chunks* devido a recepção de conteúdo poluído.

Para o propósito de análise, foram considerados três cenários diferentes. No primeiro caso, os poluidores atacam o sistema durante todo o tempo até o final do experimento, sendo utilizado o *mecanismo de reputação simples*. No segundo caso, os poluidores atacam o sistema seguindo o mesmo comportamento anterior, porém eles saem

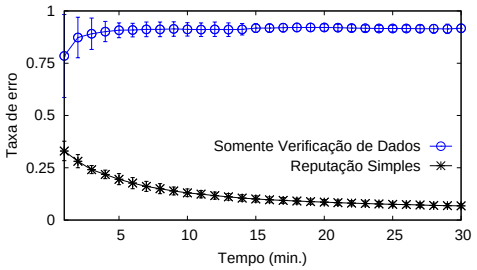

Figura 1. Taxa de erros de *chunks*.

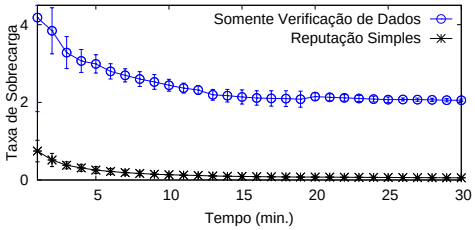

Figura 2. Sobrecarga no sistema.

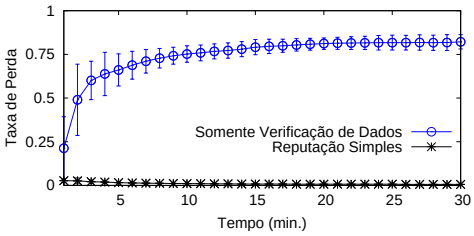

Figura 3. Taxa de perda.

e entram novamente no sistema, aproximadamente a cada 3 minutos, assumindo uma nova identidade (ataque de *whitewashing*). No último caso, é incluído o *mecanismo de reputação modificado* que combate aos *whitewashers*.

5. Resultados

Nesta seção é apresentado os resultados obtidos através dos experimentos realizados no ambiente real configurado no PlanetLab. Todos os resultados apresentam o valor médio da medida calculada e o intervalo de confiaça da realização de 5 rodadas de difusão de um vídeo de 30 minutos através do protótipo da aplicação P2P de vídeo ao vivo.

5.1. Eficiência do Mecanismo de Reputação

Para avaliar a eficiência do mecanismo de reputação proposto neste artigo, três métricas foram avaliadas: a) taxa de *chunks* poluídos recebidos ou *taxa de erro*; b) a sobrecarga no sistema e; c) taxa de *chunks* perdidos (taxa de perda). Essas métricas permitem capturar o dano causado ao sistema e também inferir a qualidade de experiência dos usuários.

A Fig. 1 mostra os resultados no cenário em que os poluidores não realizam o ataque de *whitewashing*. O mecanismo *verificação de dados*, representa o simples descarte de *chunks* reconhecidos como poluídos (por exemplo, através de um esquema de *hash*) e o pedido de retransmissão dos mesmos. O mecanismo *reputação simples* refere-se ao mecanismo de reputação proposto na Seção 3. É possível notar qua a taxa de erros de *chunks* é de quase 100% em um sistema onde não há tentativa de isolar poluidores. O *mecanismo de reputação simples* proposto diminui a taxa de erros de 90% para 6%. No esquema de *verificação de dados*, os *peers* receberam uma grande quantidade de *chunks* poluídos. No pior caso, todo o vídeo é perdido.

Fig. 2 apresenta a sobrecarga no sistema. Em um sistema usando somente a abordagem de verificação e retransmissão de dados, a sobrecarga se estabiliza em torno de 230%. Os *peers* devem ter mais de três vezes a largura de banda necessária em um

sistema sem poluidores. O *mecanismo de reputação simples* reduz a sobrecarga a valores abaixo de 5%.

Finalmente, é apresentado a taxa de perda global do sistema devido a várias requisições por retransmissão. A Fig. 3 mostra que ataques de poluição geram grandes danos em um sistema que usa somente a abordagem de verificação de dados. Neste caso, a taxa de perda de *chunks* alcança valores de até 82%. Como consequência, os *peers* não serão capazes de assistir continuamente ao vídeo, diminuindo a qualidade de experiência. Usando o *mecanismo de reputação simples*, a taxa de perda cai para um valor desprezível, abaixo de 0.7%.

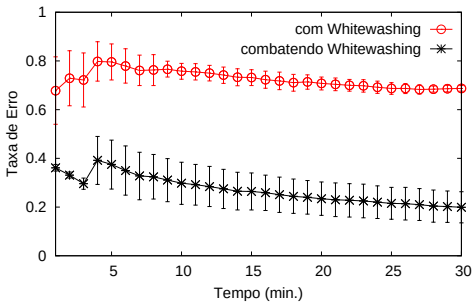

Figura 4. Taxa de erro de *chunks*.

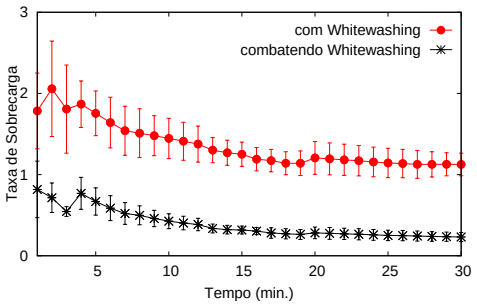

Figura 5. Sobrecarga no sistema.

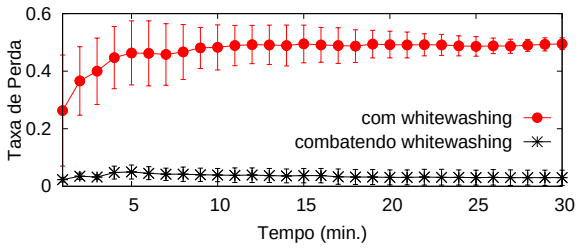

Figura 6. Taxa de Perda.

A Fig. 4 apresenta a taxa de erro em um cenário onde poluidores também realizam ataques de *whitewashing*. Os resultados mostram que, mesmo com o *mecanismo de reputação simples*, *whitewashing* ainda causa uma alta taxa de erro, alcançando quase 70%. Porém, o mecanismo de reputação modificado para lidar com *whitewashing* reduz a taxa de erro de *chunks* a 19%.

A sobrecarga também apresenta uma diferença significante de comportamento do cenário onde os poluidores não fazem *whitewashing*. A Fig. 5 mostra que *whitewashing* é um comportamento indesejado. Utilizando o *mecanismo de reputação simples*, os *peers* experimentam 112% de sobrecarga: os *peers* precisam de mais de dois vezes a largura de banda necessária em um cenário sem ataques. O *mecanismo de reputação modificado* para lidar com *whitewashing* reduziu a sobrecarga para 20%.

De acordo com a Fig. 6, a taxa de perda é menor que 3% quando o *mecanismo de reputação modificado* é implementado para lidar com *whitewashing*. Primeiro, o esquema proposto é fácil e barato de implementar. Segundo, no esquema de reputação proposto não é necessário manter um custoso mecanismo de identificação de *peers* centralizado ou distribuído comumente usado para evitar o ataque de *whitewashing*.

Uma das características importantes do esquema de reputação simples proposto é a recuperação de *peers* classificados como um *peer bom* e que possam ter sido considerados poluidores devido ao funcionamento incorreto da rede, ou seja, os chamados falsos positivos. Para avaliar a robustez dos mecanismos propostos, um *peer* no sistema foi configurado para enviar 10% de seus *chunks* propositalmente como corrompidos (simulando problemas de rede). Para definir os estados de *calmaria* e *tempestade* foi usada a razão *chunks* poluídos/não-poluídos recebidos por cada *peer*.

Foram considerados três valores do parâmetro *lcalm* $= \{0.1, 0.2, 0.3\}$. Para *lcalm* $= 0.1$, 30.2% dos *peers* identificaram o falso poluidor, sendo que 82.8% destes recuperam o falso positivo em aproximadamente 110 segundos. Para o caso *lcalm* $= 0.2$, 28.9% dos *peers* identificaram o falso poluidor, sendo que 73% destes recuperam o falso positivo em um tempo médio de 97 segundos. Por último, para o caso em que *lcalm* $= 0.3$, 27.3% dos *peers* identificam o falso poluidor e 51% voltam a trocar informações. O tempo para recuperação foi, em média, de 72 segundos.

A seguinte discussão pode ser feita com base nos resultados apresentados. Para valores decrescentes do parâmetro *lcalm*, uma maior percentagem de *peers* identificam o falso positivo, bem como o recuperam. Para valores menores de *lcalm*, o período de *tempestade* é mais longo, maior é o valor da reputação mínima R_i^{min}, por consequência, maior a sensibilidade do *peer* a identificar poluição. Poluidores reais são removidos mais rapidamente, atingindo o estado de calmaria. A necessidade de um maior tempo para recuperação de falso positivos está relacionada a dinâmica da mudança de valores da reputação mínima.

Para visualizar o impacto em p_j considerado como falso positivo, a Fig. 7, mostra o total de *chunks* recebidos pelo mesmo durante toda a simulação.. Em todos os casos, após 10 minutos iniciais, p_j recebe, em média, de 2900 a 3000 *chunks* a cada 30 segundos. Este total é o valor esperado de recebimento de *chunks* pelos *peers* que não agem como poluidores. Assim, mesmo p_j sendo considerado como poluidor por um tempo, este não terá problemas ao exibir o vídeo. Este resultado está intimamente ligado à rápida recuperação do mecanismo de reputação implementado.

Concluindo, a Fig. 8 mostra o número de *chunks* recebidos pelo falso positivo p_j, no intervalo entre 15 minutos e 25 minutos. Este intervalo representa o melhor caso da recuperação de falso positivo devido ao mecanismo de reputação proposto. Neste intervalo, o sistema se encontra estável, e p_j recebe quase 100% de *chunks* sem poluição (taxa de *chunks* poluídos não chega a 1%).

5.2. Impacto do Mecanismo de Reputação na Qualidade oferecida aos Usuários

Apesar dos mecanismos propostos combaterem com eficiência o ataque de poluição combinado ou não com o ataque de *whitewashing*, os resultados da Seção 5.1 mostram que, mesmo com os mecanismos implementados, estes ataques ainda causam danos no sistema P2P. Desta forma, é necessário estudar a viabilidade prática do uso dos mesmos, no que tange a qualidade provida aos usuários do sistema.

Para analisar a qualidade de experiência dos usuários quando os mecanismos de reputação propostos são implementados, considera-se a métrica atraso de exibição do vídeo, ou seja, o intervalo de tempo entre criação de *chunks* na origem e exibição dos mesmos nos *peers*. A Fig. 9 mostra o valor da métrica na abordagem onde somente a

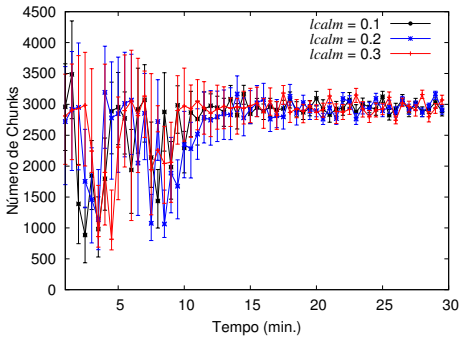

Figura 7. Total de *chunks* recebidos pelo falso positivo.

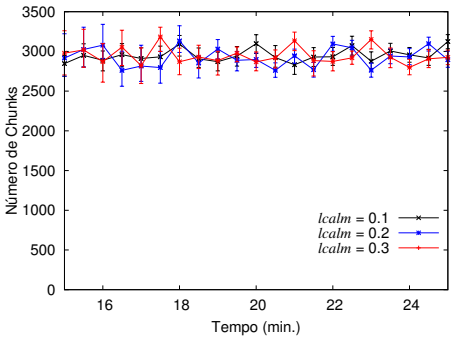

Figura 8. Total de *chunks* recebidos pelo falso positivo.

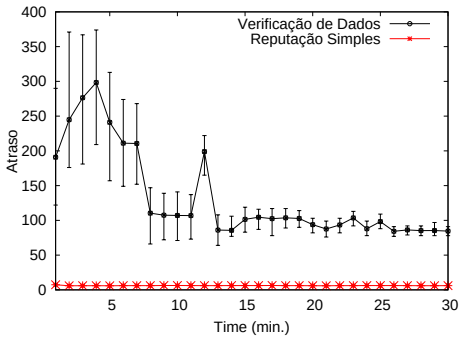

Figura 9. Atrasos - Verificação de Dados e Reputação Simples.

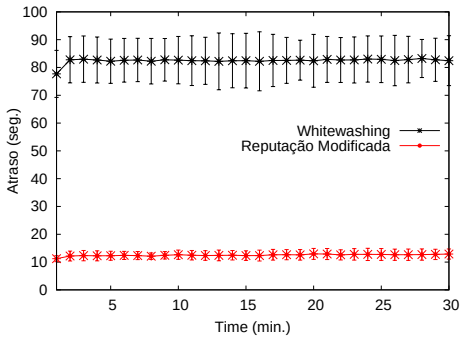

Figura 10. Atrasos - Whitewashing e Reputação Modificada.

verificação de dados é implementada. O atraso médio (considerando todos os *peers* e todos os *chunks* é de mais de 2 minutos (132 segundos). Somando-se as métricas de taxa de erros, perdas e sobrecarga mostradas anteriormente pode-se afirmar que a qualidade de experiência dos usuários é muito ruim. Em contrapartida, a mesma Fig. 9 mostra que o *mecanismo de reputação simples* reduz o atraso médio a apenas 6 segundos.

Finalizando, a Fig. 10 mostra o atraso médio quando os poluidores também realizam o ataque de *whitewashing*. Apesar de aplicar o *mecanismo de reputação simples*, o atraso médio é de quase 1 minuto. Este resultado confirma a necessidade de modificação do mecanismo simples para tratar *whitewashers*. Com a modificação proposta na Seção 3, o atraso médio é reduzido a cerca de 13 segundos, valor aceitável para aplicações desta natureza.

6. Conclusões

Este artigo analisa o impacto negativo da presença de *peers* poluidores em um sistemas P2P de transmissão de vídeo ao vivo. Estes resultados motivam propostas de mecanismos de defesa contra estes participantes, principalmente quando os mesmos agem também como *whitewashers*. Nesta direção foram propostos dois mecanismos de reputação que combatem ataques de poluição e ataques combinados de poluição e *whitewashing*. Suas principais vantagens são a simplicidade de implementação e a característica distribuída : informações necessárias são coletadas localmente em cada participante da difusão.

Para a análise dos mecanismos propostos, foi implementado um sistema real de

difusão de conteúdo ao vivo, e resultados experimentais foram realizados na plataforma PlanetLab. Através das métricas de interesse definidas, o mecanismo de reputação para combater poluição, dimuinui, por exemplo, em até 46 vezes a sobrecarga no sistema. Para os casos com ataque de *whitewashing* a diminuição é de aproximadamente 6 vezes. Os resultados também mostraram a robustez do sistema em recuperar falsos positivos. Adicionalmente, a qualidade de experiência dos usuários permanece satisfatório, mesmo quando os mecanismos de reputação são implementados.

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*, pages 481–484. IEEE.

Chen, J., Lu, H., and Bruda, S. (2009). A solution for whitewashing in p2p systems based on observation preorder. In *IEEE NSWCTC*.

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

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 Renesse, R. V. (2008). Securestream: An intrusion-tolerant protocol for live-streaming dissemination. *Elsevier Computer Communications*, 31(3):563–575.

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

Hei, X., Liu, Y., and Ross, K. W. (2008). Iptv over p2p streaming networks: The mesh-pull approach. *IEEE Communications Magazine*.

Hu, B. and Zhao, H. (2010). Joint pollution detection and attacker identification in peer-to-peer live streaming. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*.

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

Oualha, N. and Roudier, Y. (2009). A game theoretical approach in securing p2p storage against whitewashers. In *18th IEEE WETICE'09*.

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

Vieira, A., Campos, S., and Almeida, J. (2009). Fighting attacks in p2p live streaming. simpler is better. In *IEEE INFOCOM Workshops*.

YU, X. and FUJITA, S. Whitewash-aware reputation management in peer-to-peer file sharing systems. *management*, 6(5):9.