CCNcheck: um mecanismo de mitigação para poluição de conteúdos em Redes Centradas em Conteúdo

Igor C. G. Ribeiro*, Flávio de Q. Guimarães*
Célio V. N. Albuquerque*, Antonio A. de A. Rocha

Instituto de Computação
Universidade Federal Fluminense (UFF)
Niterói, RJ – Brasil

{iribeiro,fqueiroz,celio,arocha}@ic.uff.br

Abstract. *Content-Centric Networking is a proposal for the future Internet, where the data is identified and requested based on its name. Any node that stores a content with the same name can answer to the user. To ensure the data's integrity and authenticity, all contents in the network are digitally signed by its publishers. Nevertheless, make the routers to check the signature of all contents imposes a significative processing overhead. For this reason, the signature verification is optional for routers, and it is not executed by default. Based on this behavior, malicious producers can create polluted versions of contents, therefore reducing their availability. To mitigate this problem, we propose CCNcheck, a mechanism that imposes a probabilistic signature verification by routers. We concluded that CCNchek allows consumers, in specific scenarios, to recover a greater number of valid contents and helps to reduce the waste of network resources used to forward polluted contents.*

Resumo. *A Rede Centrada em Conteúdo constitui uma proposta para a Internet do futuro, onde os dados são identificados e requisitados pelo nome. Qualquer repositório que armazene um conteúdo com este nome pode responder ao usuário. Para garantir a integridade e autenticidade, os conteúdos são assinados por seus publicadores. Apesar disso, verificar a assinatura de todos os conteúdos nos roteadores introduz um overhead de processamento significativo. Assim, a verificação de assinaturas pelos roteadores é opcional e por padrão não é realizada. Como consequência, publicadores maliciosos podem criar versões poluídas dos conteúdos, reduzindo assim sua disponibilidade. Para mitigar este problema, neste trabalho é proposto o CCNcheck, um mecanismo que realiza a verificação probabilística de assinaturas de conteúdos pelos roteadores. Como resultado, o CCNcheck permite, em cenários específicos, que os consumidores recuperem uma quantidade maior de conteúdos válidos e reduz o desperdício de recursos usados no encaminhamento de conteúdos poluídos.*

1. Introdução

Com a proposta das Redes Centradas em Conteúdo (*Content Centric Networks* - CCN) [Jacobson et al. 2009], a requisição e recuperação de dados deixam de ser orientadas à localização e se tornam orientadas ao conteúdo em si. Apesar de ser utilizado em

*Laboratório MídiaCom

uma nova proposta para a arquitetura da rede, esse paradigma não é realmente novo. As redes *Peer-to-Peer* (P2P), por exemplo, já permitem que conteúdos sejam buscados e requisitados na rede, independentemente de sua localização. A inovação trazida pela CCN é a implementação deste paradigma na camada de rede, e não na camada de aplicação. Por essa razão, mesmo que a CCN ainda não esteja implementada em larga escala, é possível supor que alguns dos problemas enfrentados nas redes P2P também ocorrerão na CCN.

Com a popularização das redes P2P ocorrida na última década, a disponibilidade de diversos tipos de arquivos foi radicalmente ampliada. Com isso, se tornou fácil recuperar músicas e vídeos protegidos por *copyright*, impactando negativamente no faturamento das produtoras. Por sua vez, tais produtoras contra-atacaram através de processos legais, como no caso do Napster [BBC]. Entretanto, com a evolução da tecnologia das redes P2P, os sistemas de busca de conteúdos se tornaram distribuídos e, por consequência, mais difíceis de serem paralisados. Por outro lado, trabalhos como [Lou and Hwang 2009] sugerem que a inserção de versões poluídas de conteúdos protegidos por *copyright* podem levar a uma redução da disponibilidade das versões legítimas de tais conteúdos.

Na CCN, a disseminação de conteúdos poluídos tem o potencial de alcançar níveis ainda maiores do que aqueles das redes P2P, devido a realização de *cache* de conteúdos por todos os nós da rede. Apesar disso, pouco tem sido discutido sobre essa ameaça. Este trabalho apresenta o CCNcheck e constitui um dos primeiros esforços no combate à poluição de conteúdos na CCN. Este mecanismo faz com que os roteadores da rede verifiquem, probabilisticamente, assinaturas de conteúdos, descartando aqueles considerados poluídos. Como resultado, o CCNcheck objetiva possibilitar que os usuários recuperem uma fração maior de conteúdos legítimos e consequentemente reduzir o desperdício de recursos da rede utilizados no encaminhamento de conteúdos poluídos. Neste contexto, a principal contribuição deste trabalho é a proposta, implementação e avaliação do CCNCheck para a redução da disseminação de conteúdos poluídos na CCN.

Este trabalho está organizado da seguinte forma: a Seção 3 fornece uma visão geral sobre a arquitetura da CCN. Na Seção 4, é apresentado o CCNcheck, um mecanismo para a redução da disseminação de conteúdos poluídos na rede. A Seção 5 descreve as simulações utilizadas para avaliar o mecanismo proposto. Os resultados obtidos são discutidos na Seção 6. Na Seção 2 são considerados alguns trabalhos relacionados. Por fim, na Seção 7 são apresentadas as conclusões e os trabalhos futuros.

2. Trabalhos Relacionados

Por ser uma tecnologia nova e ainda não implementada em larga escala, muitas das questões relevantes à CCN ainda estão sendo amadurecidas. São exemplos disso, problemas de segurança como: ataques de negação de serviço [Compagno and Tsudik 2012] e a violação da privacidade de usuários [Arianfar et al. 2011], [DiBenedetto et al. 2012]. Entretanto, diferentemente da maioria dessas áreas de pesquisa, o problema da poluição de conteúdos na CCN tem recebido pouca atenção da comunidade científica.

Na realidade, até onde foi possível investigar, apenas o trabalho proposto por [Gasti et al. 2012] aborda este problema. Neste trabalho, os autores sugerem que a disseminação de conteúdos poluídos na CCN pode levar à negação de serviço, já que um consumidor pode não conseguir recuperar uma versão legítima do conteúdo desejado. Além disso, também são propostas duas possíveis contramedidas baseadas na

verificação probabilística de assinaturas. Na primeira, cada roteador verifica um conjunto aleatório de conteúdos armazenados em seu *cache*. A segunda contramedida requer que a carga de verificação de conteúdos seja distribuída entre todos os roteadores pertencentes ao mesmo domínio administrativo. Ambas as propostas abordam somente a verificação de conteúdos já armazenados em *cache*. O CCNcheck, por outro lado, propõe a verificação dos conteúdos como parte do processo de decisão de encaminhamento. Assim, o CCNcheck pode ser visto como um complemento aos mecanismos propostos em [Gasti et al. 2012], podendo ser utilizados em conjunto.

Apesar da curta literatura diretamente relacionada à poluição de conteúdos na CCN, o mesmo não ocorre em relação às redes P2P - *Peer-to-Peer*, onde existem diversos trabalhos abordando o assunto. Apesar dessas duas arquiteturas possuírem diferenças de implementação, ambas compartilham o mesmo paradigma: o conteúdo é o ente mais importante e não sua localização física. Assim, é possível investigar os trabalhos que abordam a poluição de conteúdos em redes P2P no contexto da CCN.

Visando compreender melhor a natureza e a magnitude da poluição de conteúdos em sistemas P2P de compartilhamento de arquivos, [Liang et al. 2005] apresentam um estudo baseado no KaZaA, o sistema P2P de compartilhamento de arquivos mais popular na época. Como resultado, observou-se que para músicas recentes, mais de 50% das cópias eram poluídas e que essa poluição era intencional. Na CCN, todos os nós compartilham conteúdos, inclusive os roteadores. Por esta razão, se os roteadores não verificarem a assinatura dos conteúdos, então é de se esperar que a disseminação de conteúdos poluídos na CCN será ainda maior do que nas redes P2P, onde somente os sistemas finais compartilham conteúdos.

Em [Lee et al. 2006], é proposto um modelo que leva em conta o fator humano no processo de disseminação de conteúdos. Através de uma pesquisa realizada com universitários, percebeu-se que os usuários nem sempre reconhecem se um conteúdo é ou não poluído, podendo então auxiliar, de maneira não intencional, na disseminação de conteúdos poluídos na rede. A partir deste trabalho, é possível concluir que para maximizar o efeito da poluição de conteúdos, os poluidores precisam ser eficazes em impedir que os usuários identifiquem um conteúdo como poluído. Além disso, também foi mostrado que ataques realizados com base em conteúdos populares podem quadruplicar a carga de tráfego na rede. Na CCN, devido ao processo de verificação da assinatura dos conteúdos, os usuários nunca falham ao determinar se um conteúdo recebido é ou não uma versão poluída[1].

Para combater a poluição de conteúdos nas redes P2P, foram realizadas diversas propostas, onde a maioria estabelece um sistema de reputação para os conteúdos ou para os usuários. Por serem muito mais próximos do seu paradigma orientado ao conteúdo, sistemas de reputação baseados no conteúdo seriam mais interessantes e adequados à CCN. Neste contexto, é possível citar os sistemas Credence [Walsh and Sirer 2005], que utiliza votos positivos e negativos, e inforanking [FOTIOU et al. 2010], baseado somente em votos positivos. É importante notar que estes mecanismos de reputação pertencem à camada de aplicação, enquanto o CCNcheck é uma abordagem baseado na camada de rede. Nada impede, portanto, que o CCNcheck seja usado em conjunto com algum desses

[1]Na realidade, devido às colisões de *hash*, existe uma probabilidade desprezível, porém não nula, de um conteúdo poluído ser identificado como legítimo.

mecanismos.

3. Visão Geral da Rede Centrada em Conteúdo

A Rede Centrada em Conteúdo [Jacobson et al. 2012] propõem uma nova arquitetura para a Internet, tornando a rede mais próxima das necessidades atuais dos usuários, tanto em relação a segurança quanto na maneira de se recuperar conteúdos. A CCN segue um paradigma onde o conteúdo é o ente mais importante e não sua localização na rede. Isso significa que, para requisitar um conteúdo, não é necessário saber em que servidor ele está armazenado, mas somente o nome que o identifica. Cabe então à rede localizar o conteúdo baseando-se apenas em seu nome.

Na CCN, os roteadores utilizam os nomes de conteúdo para consultarem a tabela de encaminhamento e saberem por quais interfaces um determinado pacote pode ser encaminhado. Para que seja possível interpretar e extrair informações desses nomes, os roteadores precisam conhecer o padrão de nomeação utilizado. No padrão adotado pela CCN, conteúdos são nomeados utilizando-se uma estrutura semelhante às *Uniform Resource Locator* (URLs). Por exemplo, o video da primeira aula da disciplina Redes de Computadores da Universidade Federal Fluminense poderia ser nomeado da seguinte maneira: `/br.uff.ic/redes/aulas/video1`.

A recuperação de conteúdos na CCN é baseada no modelo de requisição/resposta, onde entidades, conhecidas como publicadores, disponibilizam conteúdos na rede e os clientes, conhecidos como consumidores, requisitam esses conteúdos. Assim, para recuperar determinado conteúdo, foram definidos dois tipos de pacote, um para transportar a requisição e outro para o conteúdo propriamente dito. Esses pacotes recebem o nome de pacote de interesse, doravante chamado apenas de interesse, e pacote de dados, respectivamente.

Ao se propagarem na rede, os interesses deixam ”rastros” nos roteadores. Esses rastros são, na verdade, entradas na *Pending Interest Table* (PIT), que associam interfaces de entrada a um nome de conteúdo. Utilizando essas informações, os pacotes de dados podem ser encaminhados através do caminho reverso do interesse que os originou. Por conta desta característica, se mais de um interesse for recebido para o mesmo conteúdo, só há a necessidade de encaminhar o primeiro deles para o próximo salto. Os interesses ficam “aguardando” na PIT o retorno do conteúdo. Para aumentar a disponibilidade dos dados e reduzir o tempo de resposta, todos os nós da rede realizam *cache* de conteúdos e utilizam uma estrutura conhecida como *Content Store* (CS), para armazená-los. Mais informações sobre a CCN, assim como sobre outras arquiteturas baseadas no mesmo paradigma, podem ser encontradas em [Brito et al. 2012].

Para garantir a integridade e autenticidade dos conteúdos, a CCN requer que toda entidade que tenha a intenção de ser um publicador de conteúdos possua um par de chaves assimétricas, uma pública e uma privada. Para publicar um conteúdo na rede, o publicador deve gerar uma assinatura digital do conteúdo e fornecê-la para o consumidor, juntamente ao conteúdo em si, através de um pacote de dados. Uma vez que os conteúdos são requisitados pelo nome, é necessário estabelecer uma associação forte entre este nome e o conteúdo que ele identifica. Na CCN esta associação forte é criada através da assinatura não somente do conteúdo, mas sim do mapeamento entre o conteúdo e o seu nome. Uma maneira de implementar esse mecanismo seria gerar o *hash* criptográfico do conteúdo e do

seu nome, concatenar os dois e em seguida cifrar essa concatenação com a chave privada do publicador. Assim, ao receber um conteúdo, o consumidor também precisa obter a chave pública do publicador para verificar a assinatura presente no pacote de dados. Uma análise mais completa sobre os mecanismos de segurança da CCN, suas vulnerabilidades e possíveis ataques, pode ser encontrada em [Ribeiro et al. 2012].

Por padrão, a assinatura dos conteúdos é verificada apenas pelos consumidores, e não pelos roteadores da rede. Essa característica é suficiente para permitir que os usuários não sejam vítimas de dados contendo *malwares*. Entretanto, se os consumidores sempre receberem conteúdos inválidos, então é possível dizer que o serviço ao qual a CCN se presta estará sendo negado a eles. Por outro lado, impor a verificação da assinatura de todos os conteúdos aos roteadores causaria um *overhead* considerável aos mesmos. Por essa razão, esta abordagem é considerada inviável [Gasti et al. 2012]. A Seção 4 aborda o CCNcheck, um mecanismo que adota uma abordagem intermediária entre checar assinaturas somente nos consumidores e checar a assinatura de todos os conteúdos nos roteadores.

4. CCNcheck

Este trabalho propõe o CCNcheck, um mecanismo que provê a verificação probabilística de assinaturas de conteúdos pelos roteadores da rede. Seu principal objetivo é aumentar a disponibilidade de conteúdos legítimos e reduzir o desperdício de recursos introduzido pelo encaminhamento de conteúdos poluídos até o consumidor. O valor da probabilidade de verificação deve ser definido com cuidado para garantir um balanceamento eficiente entre o *overhead* de verificação e a eficiência na redução da disseminação de conteúdos poluídos. O CCNcheck é flexível e permite que sejam definidas estratégias específicas para cada cenário. Por exemplo, é possível definir que todos os roteadores da rede utilizem o mesmo valor de probabilidade, ou que os roteadores de borda verifiquem assinaturas com uma probabilidade maior do que os demais, ou ainda ajustar dinamicamente o valor da probabilidade de verificação de conteúdos.

4.1. Definição do Ataque

No contexto deste trabalho, publicadores maliciosos podem realizar os seguintes tipos de ataque de poluição de conteúdos:

1. Renomeação: os conteúdos são publicados com um nome diferente do original.
2. Corrupção: os conteúdos são publicados com seus *bits*, ou metadados, alterados. Dessa forma, os consumidores podem receber conteúdos inúteis como resposta às suas requisições;
3. Falsificação: Neste caso existem duas possibilidades. Na primeira, os conteúdos são publicados com uma chave pública que não está associada à chave privada utilizada para assiná-los. A segunda alternativa é publicar conteúdos com uma chave pública que não esteja associada a entidade do mundo real esperada pelo consumidor.

Em qualquer um desses casos, o objetivo do atacante não é enganar o consumidor, já que este pode facilmente verificar a assinatura dos conteúdos e detectar possíveis violações. O objetivo destes ataques é reduzir a disponibilidade de certos conteúdos populares através da disseminação de suas versões poluídas. Por consequência, os consu-

midores podem ser impedidos (ou ao menos atrapalhados) de obter uma versão válida do conteúdo desejado.

4.2. Implementação

A CCN permite que as decisões de encaminhamento de interesses sejam tomadas salto a salto, com base na política de encaminhamento utilizada, definida em sua camada de estratégia. Por outro lado, o processo de encaminhamento de pacotes de dados é definido na estrutura interna da CCN e é único para todos os nós da rede. Com base nisso, propõe-se que o CCNcheck seja implementado como uma política de encaminhamento de pacotes de dados, residente na camada de estratégia, assim como as políticas de encaminhamento de interesses.

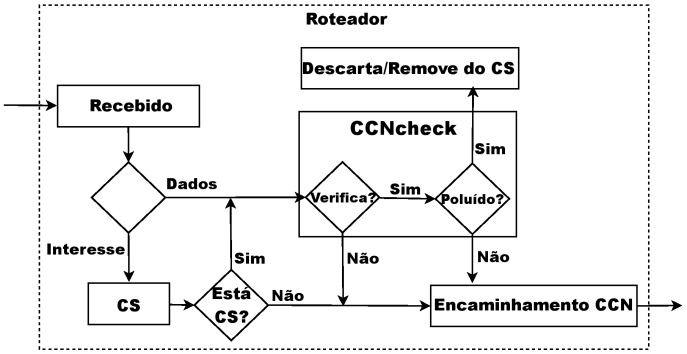


Figura 1: Papel do CCNcheck no processo de encaminhamento de pacotes.

A Figura 1 resume o funcionamento do CCNckeck. Quando um pacote é recebido, seu cabeçalho é analisado para determinar se ele é um interesse ou um pacote de dados. Se o roteador possuir o conteúdo requisitado em seu *CS*, então este é direcionado para o CCNcheck antes de ser encaminhado. O mesmo ocorre quando o roteador recebe um pacote de dados de um de seus vizinhos. Dentro do módulo do CCNcheck, a assinatura do conteúdo pode ou não ser verificada, de acordo com a probabilidade de verificação estabelecida. Se a assinatura não for verificada, ou se ela for verificada e o conteúdo for legítimo, o pacote de dados é encaminhado como de costume pela CCN. Caso o conteúdo esteja poluído, ele é removido do *CS*, se estiver no *cache*, e é descartado em seguida.

Após o pacote de dados ser recebido pelo módulo do CCNcheck, o Algoritmo 1 é executado. Como entrada, são esperados dois parâmetros, a assinatura da associação entre o conteúdo e seu nome e a chave pública do publicador. Após o processamento, é informado se o conteúdo está ou não poluído. Na linha 2, a função $GetProbability$ retorna a probabilidade de verificação de assinaturas para o nó em questão. Essa função pode ser implementada para retornar um valor estático, ou calcular um valor dinâmico, de acordo com as características da rede.

O corpo da instrução IF (linha 6) deve ser executado de acordo com a probabilidade p. Por essa razão, $p = P_r[r < k]$, ou seja, a probabilidade da variável r ser menor que a variável k é igual a p. A variável r representa um valor inteiro aleatório sorteado uniformemente a partir do intervalo $[0, MAX]$. A variável k é calculada da seguinte forma: seja $P_r[r = i]$ a probabilidade de r ser igual ao valor i. Como r é uma variável aleatória uniforme, $P_r[r = i] = 1/(MAX + 1), \forall i$. Assim, $P_r[r < k]$ é dada por

Algoritmo 1: ProbabilisticCheck

Data: Assinatura do conteúdo: $Sign(C, N)$.
Data: Chave pública do publicador: K_{PUB}.
Result: $True$, se o conteúdo for legítimo e $False$ caso esteja poluído ou não tenha sido verificado.

```
1  begin
2      p ⟵ GetProbability
3      k ⟵ p · (MAX + 1)
4      r ⟵ rand()
5      result ⟵ True
6      if r < k then
7          signatureOk ⟵ CheckSignature(K_PUB, Sign(C, N))
8          publicKeyOk ⟵ CheckPublicKey(K_PUB)
9          result ⟵ signatureOk and publicKeyOk
10     end
11     return result
12 end
```

$P_r[r < k] = P_r[r = 0] + P_r[r = 1] + ... + P_r[r = k - 1] = k \cdot 1/(MAX + 1)$. Logo, $k = p \cdot (MAX + 1)$. A função $CheckSignature$ verifica a assinatura propriamente dita e a função $CheckPublicKey$ verifica se a chave pública do publicador realmente pertence a entidade do mundo real esperada. Se ambas as verificações retornarem $True$, então conclui-se que o conteúdo é legítimo. Caso contrário, o conteúdo é identificado como poluído e o pacote de dados será descartado pelo CCNcheck. Se um conteúdo não é verificado, ele é considerado legítimo.

5. Avaliação

Para avaliar a eficiência do CCNcheck, foram realizadas simulações utilizando o simulador NS3 [NS3], através do módulo NDNSim [Afanasyev et al. 2012], que implementa a pilha de protocolos da CCN. Todas as simulações realizadas podem ser descritas em função do comportamento dos consumidores (*CO*), publicadores legítimos (*PL*), publicadores maliciosos (*PM*) e dos roteadores (*R*). As próximas seções descrevem esses comportamentos e também os quatro cenários de simulação utilizados.

5.1. Comportamento dos Publicadores

Inicialmente, foi preciso definir de que maneira seria implementada a publicação de conteúdos legítimos. A CCN não define um padrão para tal e por isso é possível escolher a maneira que melhor se encaixa em cada caso. O NDNSim possui três opções para este propósito:

1. as entradas na FIB de cada roteador devem ser inseridas manualmente, sendo equivalente ao uso de rotas estáticas;
2. uma entidade global é responsável por calcular as rotas e popular as FIBs de todos os roteadores da rede, atuando como um protocolo de roteamento centralizado;
3. os interesses inundam a rede em busca de um nó que possua o conteúdo requisitado.

Claramente a primeira opção não é nem prática nem escalável, o que limitaria bastante a densidade das topologias utilizadas nas simulações. A segunda opção seria interessante, porém ela é implementada de forma a não permitir que um interesse seja encaminhado por mais de uma interface em um roteador. Dessa maneira, ou o interesse seria sempre encaminhado até um publicador legítimo ou até um malicioso, o que não seria adequado à análise da poluição de conteúdos na CCN. Dessa forma, optou-se por utilizar a terceira opção, ou seja, a inundação de pacotes de interesse.

Os *PLs* somente respondem a interesses cujo nome de conteúdo esteja dentro de seu prefixo de atuação. Por exemplo, definindo-se o prefixo */br.uff.ic* para *PL*, faria com que ele respondesse com um pacote de dados para */br.uff.ic/videos/1*, mas não para */br.ufrj.if/videos/1*.

Por outro lado, nenhum prefixo de nome de conteúdo é atribuído aos *PMs*. Ao invés disso, esses publicadores são configurados para responder a qualquer interesse recebido. Uma vez que os interesses inundam a rede, contanto que exista um caminho entre *CO* e os *PMs*, sempre haverá uma resposta maliciosa para uma requisição de conteúdo. Todos os pacotes de dados enviados pelos publicadores legítimos e maliciosos possuem tamanho de 1024 bytes.

5.2. Comportamento do Consumidor

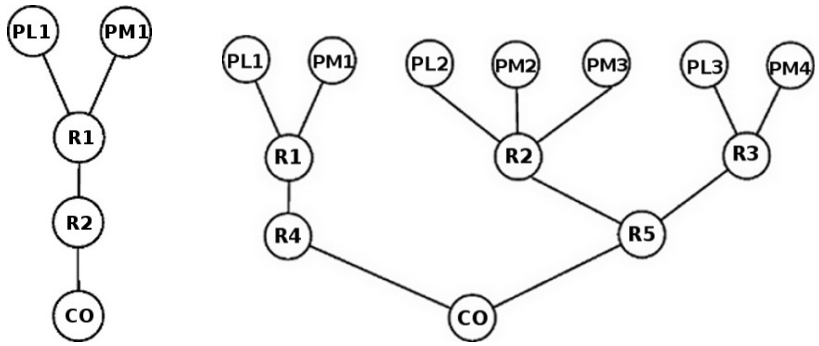
Em todas as simulações realizadas, o *CO* está sempre interessado em 20 conteúdos diferentes, que são requisitados a uma taxa de *10 interesses/s*. Para cada interesse enviado, CO inicia um temporizador e retransmite o interesse quando ocorre o estouro deste. O valor desse temporizador é definido dinamicamente, de acordo com o RTT estimado calculado pelo consumidor. Quando um conteúdo é recebido, *CO* verifica sua assinatura e o descarta se este constituir uma versão poluída. Na prática, receber um conteúdo poluído é equivalente a não ter recebido conteúdo algum. Uma vez que os interesses inundam a rede, pode ser que após receber um conteúdo poluído *CO* também receba um conteúdo legítimo oriundo de outro caminho. Por essa razão, conteúdos não são novamente requisitados imediatamente após a recepção de uma versão poluída, mas somente após o estouro do temporizador. Por fim, para evitar que interesses sejam retransmitidos eternamente caso *CO* nunca consiga obter uma versão legítima de determinado conteúdo, *CO* foi configurado para realizar no máximo 10 retransmissões para cada conteúdo requisitado.

5.3. Comportamento dos Roteadores

Em todos os roteadores, o CCNcheck é configurado com a mesma probabilidade p de verificação de assinaturas. Os pacotes de interesse recebidos são tratados da maneira usual estabelecida pela CCN (Seção 3). Os pacotes de dados são encaminhados para o módulo do CCNcheck, que procede como explicado na Seção 4.

5.4. Cenários de Simulação

As Figuras 2a e 2b ilustram dois cenários mais simples utilizados nas simulações. A Figura 2a mostra um cenário onde existem dois publicadores, um malicioso e um legítimo, um consumidor e dois roteadores. É importante perceber que, neste cenário, a única ramificação existente é a que liga o roteador $R1$ aos dois publicadores. A Figura 2b apresenta um cenário com uma topologia mais ramificada, contendo 3 publicadores legítimos,



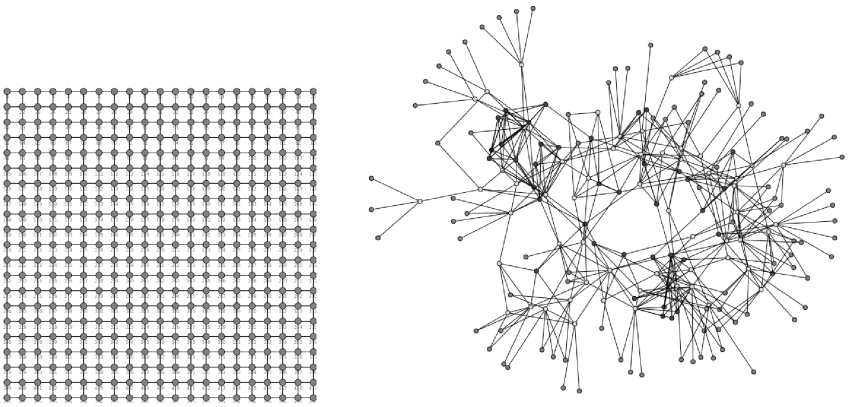
(a) Cenário 1 **(b) Cenário 2**

Figura 2: Topologias utilizadas para avaliar o comportamento do CCNcheck diante de caminhos com múltiplos saltos e ramificações. Na figura, PL são publicadores legítimos, PM são publicadores maliciosos, R são roteadores e CO são consumidores.

4 publicadores maliciosos, 1 consumidor e 5 roteadores. Apesar do número de nós ser maior nesse cenário, comparado ao do cenário da Figura 2a, o número de saltos percorridos por todos os pacotes de dados nesses dois cenários é o mesmo (igual a 3).

Para avaliar o CCNcheck em topologias mais complexas, foram utilizados os cenários da Figura 3. O Cenário 3 (Figura 3a) representa uma topologia em grade com 21 linhas e 21 colunas, totalizando 441 nós e 840 ligações entre eles. Neste cenário, *PM* ocupa a posição $(1, 10)$, *PL* ocupa a posição $(1, 12)$ e *CO* ocupa a posição $(21, 11)$.

O Cenário 4 (Figura 3b) constitui uma topologia baseada no ISP Exodus, obtida através do mapeador de topologias Rocketfuel [Spring et al. 2002]. Não foram fixadas posições para os publicadores e o consumidor. Estas foram definidas aleatoriamente durante as simulações.



(a) Cenario 3 **(b) Cenário 4**

Figura 3: Cenários utilizados para avaliar o comportamento do CCNcheck em topologias que contêm múltiplos saltos e ramificações.

5.5. Parâmetros de Simulação

Por padrão, o NDNSim utiliza um limite de tempo, definido pelo usuário, como critério de parada para as simulações. Para garantir que todos os eventos relacionados ao proces-

samento e encaminhamento de pacotes de interesse e dados sejam executados durante a simulação, foi definido um limite de tempo de 10000 segundos.

Para avaliar o efeito do aumento da probabilidade de verificação na eficiência do CCNcheck, foram determinados 10 valores para p, $\{0.0, 0.01, 0.02, ..., 0.1\}$, e cada um deles foi avaliado nos 4 cenários de simulação descritos na Seção 5.4. O objetivo de avaliar valores de p até no máximo 10%, é tentar reduzir a disseminação de conteúdos poluídos sem aumentar exageradamente o *overhead* introduzido pela verificação de assinaturas nos roteadores.

As simulações foram executadas de duas formas diferentes. Para os Cenários 1, 2 e 3, foram realizadas 500 rodadas de simulação para cada um dos dez valores de p analisados, totalizando 5000 execuções. Já no Cenário 4, foram geradas 50 posições aleatórias para as entidades *PM*, *PL* e *CO*. Em cada uma dessas posições foram executadas 300 rodadas de simulação para cada um dos dez valores de p, totalizando 150000 execuções.

6. Análise dos Resultados

Para avaliar o CCNckeck, foram propostas duas métricas que resumem bem seus objetivos: a porcentagem de conteúdos poluídos recuperados pelo consumidor e a quantidade de pacotes de dados trocados na rede. A primeira mostra como o CCNcheck traz benefícios ao usuário final, enquanto a segunda ilustra os benefícios proporcionados à infraestrutura de rede.

6.1. Porcentagem de Conteúdos Poluídos Recuperados pelo Consumidor

Em cada rodada de simulação, esta métrica foi calculada dividindo-se o total de conteúdos poluídos pelo total de conteúdos recebidos pelo consumidor. Em seguida, foi calculada a média e o desvio padrão considerando todas as rodadas de simulação. Esse procedimento foi repetido para todos os valores de p analisados e em cada um dos cenários de simulação descritos na Seção 5.4. A Figura 4 resume os resultados obtidos.

Como pode ser observado, com exceção do Cenário 4, a utilização do CCNcheck com $p = 0.0$ [2] faz com que um total de 90% (Cenário 1), 92% (Cenáio 2) e 89% (Cenário 3) dos conteúdos recuperados sejam poluídos. Quando um conteúdo poluído é recebido pelo consumidor, todos os roteadores que participaram do seu processo de encaminhamento terão armazenado uma cópia deste conteúdo em seus respectivos *CSs*. O Consumidor então descartará a versão poluída e requisitará novamente o conteúdo. Entretanto, como o(s) roteador(es) de primeiro salto possuem a versão poluída em seu *CS*, o consumidor nunca obterá uma versão válida deste conteúdo. A cada nova retransmissão, uma nova versão poluída será obtida.

No Cenário 4 (Figura 4d), esse efeito também ocorre, mas ele não é refletido nos resultados como nos outros cenários de simulação. Isto porque para $p = 0.0$, na maioria das posições aleatórias utilizadas no Cenário 4, todos ou nenhum dos conteúdos recuperados eram poluídos. Além disso, o número de posições onde 100% dos conteúdos recebidos eram poluídos é proporcional ao número de posições onde essa fração era de 0%. Por essa razão, na média, em torno de 53% dos conteúdos recuperados em todas as posições aleatórias eram poluídos.

[2]Configurar o CCNcheck com $p = 0.0$ é equivalente a desabilitar sua funcionalidade.

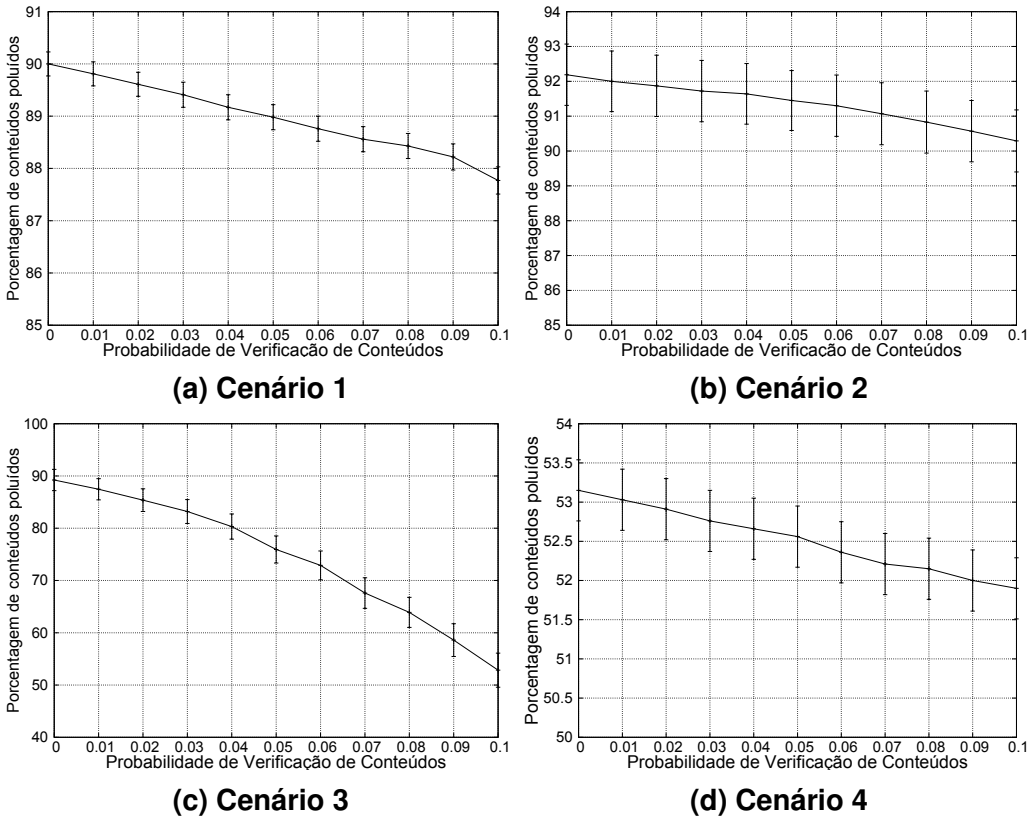


Figura 4: Porcentagem média de conteúdos poluídos recebidos pelo consumidor.

Comparando-se os resultados obtidos para os Cenários 1 e 2 (Figuras 4a, 4b), é possível perceber que para todos os valores de p, a fração de conteúdos poluídos é menor no Cenário 1. Essa diferença deve-se à relação entre o número de saltos em um caminho e o número de ramos conectados ao consumidor. Suponha que o consumidor desista de recuperar um conteúdo caso uma versão poluída do mesmo seja obtida. Suponha também que $R1$ e $R2$, no Cenário 2a, sejam substituídos por n roteadores ligados em série. Neste caso, a probabilidade do consumidor obter um conteúdo poluído é dada pela Equação 1. Logo, aumentando-se o número de saltos em um caminho entre o consumidor e um publicador malicioso, $P_r[CO]$ reduz exponencialmente. Por outro lado, se novos ramos forem adicionados ao consumidor, assim como ocorre no Cenário 2, então $P_r[CO]$ pode ser calculada pela Equação 2, onde n é o número de ramos conectados ao consumidor e $P_r[V_i]$ é a probabilidade do i-ésimo vizinho do consumidor receber um conteúdo poluído. A partir da Equação 2 é possível concluir que $P_r[CO]$ pode aumentar, diminuir ou se manter constante quando aumenta-se o número de ramos, dependendo da quantidade de conteúdos poluídos se propagando nos ramos adicionados. No Cenário 2, a quantidade de conteúdos poluídos que atravessam o ramo do roteador $R5$ é maior do que no ramo do roteador $R4$. Por essa razão, a porcentagem de conteúdos poluídos é maior no Cenário 2 que no Cenário 1.

$$P_r[CO] = \frac{1}{2} \cdot (1-p)^n \tag{1}$$

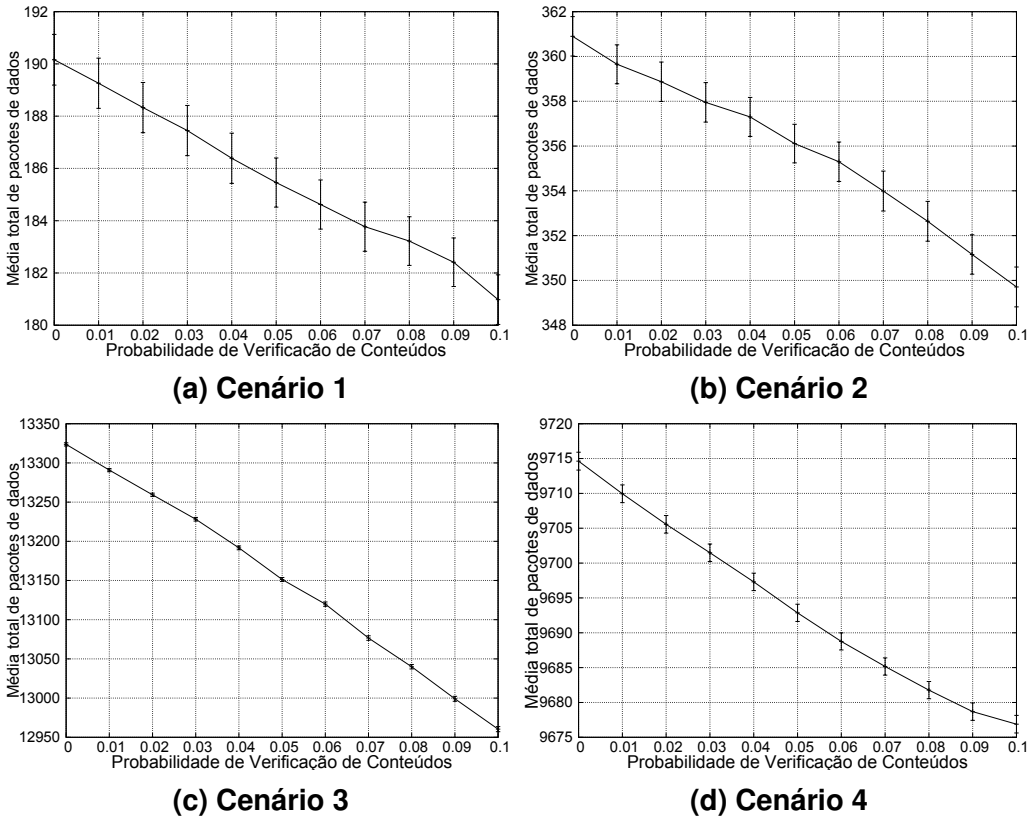


Figura 5: Média do total de pacotes de dados trocados na rede.

$$P_r[CO] = \sum_{i=1}^{n} P_r[V_i] \cdot (1-p) \cdot \frac{1}{n} \tag{2}$$

A eficiência do CCNcheck é claramente maximizada no Cenário 3. Apesar de existirem muitos caminhos neste cenário, todos eles possuem um grande número de saltos (no mínimo 21). Já no Cenário 4, os caminhos possuem uma pequena quantidade de saltos. Consequentemente, a fração de conteúdos poluídos recebidos pelo consumidor é mais baixa no Cenário 3 se comparada aos outros cenários.

6.2. Quantidade de pacotes de dados trocados

O encaminhamento de conteúdos poluídos produz diversas mensagens desnecessárias na rede, gerando desperdício de recursos. Com o uso do CCNcheck, cada vez que um conteúdo poluído é verificado por um roteador ele deixa de ser encaminhado, permitindo que os recursos outrora desperdiçados, passem a ser usados para realizar trabalho útil. Com isso, além do CCNcheck aumentar a quantidade de conteúdos válidos recebidos pelo consumidor, ele também contribui para tornar a rede em si mais eficiente.

A Figura 5 mostra o comportamento dessa métrica em todos os cenários de simulação avaliados. Devido a topologia simples dos Cenários 1 e 2, o número de pacotes de dados trocados é sempre bem menor do que nos Cenários 3 e 4. Ao observar a Figura 5, é possível concluir que a redução do número de mensagens proporcionada pelo CCNckeck é pequena. Entretanto, é preciso perceber que quando o roteador descarta uma

versão poluída de um conteúdo, uma versão legítima (ou poluída) pode ser recebida posteriormente. Assim, essa nova versão pode substituir aquela descartada no cômputo do total de pacotes de dados trocados na rede. Por essa razão, aumentando-se a probabilidade de verificação, aumenta-se também a quantidade de conteúdos legítimos se propagando na rede, como pode ser observado pelo aumento da fração de conteúdos legítimos recuperados pelo consumidor exibido na Figura 4. Assim, ainda que o total de pacotes de dados trocados não reduza tanto, o desperdício de recursos com o encaminhamento de conteúdos poluídos é reduzido.

7. Conclusão e Trabalhos Futuros

Estes trabalho apresenta o CCNcheck, um mecanismo que permite que os roteadores verifiquem as assinaturas de uma parcela aleatória dos conteúdos trocados na rede. Como resultado, é mostrado que o CCNcheck consegue reduzir o total de conteúdos poluídos recuperados pelos consumidores, fazendo com que eles consigam recuperar uma maior quantidade de conteúdos legítimos. Além disso, o CCNcheck também reduz a carga de tráfego gerada para encaminhar conteúdos poluídos, permitindo que os recursos da rede sejam melhor aproveitados no transporte de dados úteis.

Apesar do CCNcheck já apresentar resultados positivos, sua eficiência se mostrou dependente da topologia de rede utilizada. Quanto maior o número de saltos, maior a chance do conteúdo poluído ser descartado no caminho. Por essa razão, ao invés de utilizar probabilidades estáticas iguais para todos os roteadores da rede, o valor de p poderia ser calculado independentemente em cada roteador, baseando-se em sua posição física na rede e em fatores como a fração de conteúdos poluídos recebidos até o momento. Além disso, os roteadores de borda conectados aos publicadores possuem uma carga de tráfego de pacotes de dados menor do que os roteadores do núcleo da rede. Por essa razão, esses roteadores poderiam suportar uma probabilidade de verificação mais alta, fazendo com que somente uma fração pequena de conteúdo poluído seja propagada para a rede. Essas modificações têm o potencial de produzir resultados interessantes e serão analisadas em trabalhos futuros.

Referências

Afanasyev, A., Moiseenko, I., and Zhang, L. (2012). ndnSIM: NDN simulator for NS-3. Relatório Técnico NDN-0005, NDN.

Arianfar, S., Koponen, T., Raghavan, B., and Shenker, S. (2011). On Preserving Privacy in Content-Oriented Networks. In *ACM SIGCOMM Workshop on Information-Centric Networking*, pages 19–24.

BBC. Processo legal contra o Napster. http://news.bbc.co.uk/2/hi/business/1166651.stm (Acessado em 09/09/2013).

Brito, G. M., Velloso, P. B., and Moraes, I. M. (2012). Redes Orientadas a Conteúdo: Um Novo Paradigma para a Internet. In *Minicurso - Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuidos - SBRC*, pages 211–264.

Compagno, A. C. M. G. P. and Tsudik, G. (2012). NDN Interest Flooding Attacks and Countermeasures. In *ACSAC 2012*.

DiBenedetto, S., Gasti, P., Tsudik, G., and Uzun, E. (2012). ANDaNA: Anonymous Named Data Networking Application. In *NDSS 2012*.

FOTIOU, N., MARIAS, G. F., and POLYZOS, G. C. (2010). Information Ranking in Content-Centric Networks. In *Future Network and Mobile Summit, 2010*, pages 1–7.

Gasti, P., Tsudik, G., Uzun, E., and Zhang, L. (2012). DoS & DDoS in Named-Data Networking. Online: http://arxiv.org/pdf/1208.0952.pdf.

Jacobson, V., Smetters, D. K., Thornton, J. D., and Plass, M. F. (2009). Networking named content. In *International Conference on emerging Networking Experiments and Technologies - CoNEXT*.

Jacobson, V., Thornton, J. D., Plass, M., Briggs, N., Braynard, R., and Smetters, D. K. (2012). Networking Named Content. *Communications of the ACM*, 55(1):117–124.

Lee, U., Choi, M., Cho, J., Sanadidi, M. Y., and Gerla, M. (2006). Understanding Pollution Dynamics in P2P File Sharing. In *5th International Workshop on Peer-to-Peer Systems (IPTPS'06)*.

Liang, J., Kumar, R., Xi, Y., and Ross, K. W. (2005). Pollution in P2P File Sharing Systems. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1174–1185.

Lou, X. and Hwang, K. (2009). Collusive Piracy Prevention in P2P Content Delivery Networks. *IEEE Transactions on Computers*, 58(7):970–983.

NS3. NS-3 Simulator. http://www.nsnam.org (Acessado em 09/09/2013).

Ribeiro, I. C. G., Guimarães, F. Q., Kazienko, J., Rocha, A. A. A., Velloso, P. B., Moraes, I. M., and De Albuquerque, C. V. (2012). Segurança em Redes Centradas em Conteúdo: Vulnerabilidades, Ataques e Contramedidas. *Minicurso - Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais - SBSeg*, pages 101–150.

Spring, N., Mahajan, R., and Wetherall, D. (2002). Measuring ISP topologies with rocketfuel. In *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 133–145. ACM SIGCOMM.

Walsh, K. and Sirer, E. G. (2005). Fighting Peer-to-Peer SPAM and Decoys with Object Reputation. In *Workshop on Economics of peer-to-peer systems*, pages 138–143. ACM SIGCOMM.

