COPPE/UFRJ

USANDO PONTOS DE VERIFICAÇÃO PARA RASTREAR ATAQUES DE NEGAÇÃO DE SERVIÇO MACIÇAMENTE DISTRIBUÍDOS

Marcelo Duffles Donato Moreira

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Otto Carlos Muniz Bandeira Duarte

Rio de Janeiro

Setembro de 2009

USANDO PONTOS DE VERIFICAÇÃO PARA RASTREAR ATAQUES DE NEGAÇÃO DE SERVIÇO MACIÇAMENTE DISTRIBUÍDOS

Marcelo Duffles Donato Moreira

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

Prof. Otto Carlos Muniz Bandeira Duarte, Dr.Ing.

Prof. Edmundo Roberto Mauro Madeira, D.Sc.

Prof. Luís Henrique Maciel Kosmalski Costa, Dr.

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2009

Moreira, Marcelo Duffles Donato

Usando Pontos de Verificação para Rastrear Ataques de Negação de Serviço Maciçamente Distribuídos/Marcelo Duffles Donato Moreira. – Rio de Janeiro: UFRJ/COPPE, 2009.

XIV, 65 p.: il.; 29, 7cm.

Orientador: Otto Carlos Muniz Bandeira Duarte

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2009.

Referências Bibliográficas: p. 56 – 60.

1. Identificação de fonte. 2. Rastreamento Inter-domínio. 3. Ataques de Negação de Serviço. I. Duarte, Otto Carlos Muniz Bandeira. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

À minha família e

à Flávia, minha noiva.

Agradecimentos

Agradeço em primeiro lugar a Deus por ter feito bem toda a minha vida.

À minha família, em particular aos meus pais, Julio Cezar e Marli, pelo apoio nos estudos e pelo amor com que me criaram. Aos meus avós Jorge Henrique, Neyla e Gilda, ao meu padrinho André e aos meus irmãos Juliana, Gabriel e Eduardo.

À minha noiva Flávia, por me fazer feliz só pelo fato de estarmos juntos.

À minha comunidade do Caminho Neocatecumenal, por me ajudar a enxergar o amor de Deus em todos os momentos da minha vida.

Ao orientador Otto, por acreditar em mim, pela franqueza das críticas e dos conselhos e pela orientação desse trabalho.

Aos amigos, colegas e ex-colegas do GTA, Kleber, Miguel, Daniel Cunha, Pedro Velloso, Rafael Laufer, Igor Moraes, Carlos Henrique, Marcel, Natalia, Tibério, Reinaldo, Carina, Danilo, Carlo, Diorgenes, Celso, Lyno, Rafael Santos, Rodrigo, Igor Campbell, Pedro Pisa, Hugo e Diogo. Agradeço em especial ao Rafael Laufer, à Natalia e ao Igor Moraes pela contribuição neste trabalho.

A todos os professores e funcionários da COPPE. Agradeço em especial ao professor Luís Henrique Costa pela dedicação e pela atenção dispensada. Agradeço também ao professor Sergio Lima Netto pela ajuda com a teoria da distorção de taxa.

Agradeço aos professores Edmundo Madeira e Luís Henrique Costa pela participação na banca examinadora.

Ao CNPq, CAPES, FUNTTEL, FAPERJ e FINEP pelo financiamento da pesquisa.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

USANDO PONTOS DE VERIFICAÇÃO PARA RASTREAR ATAQUES DE NEGAÇÃO DE SERVIÇO MACIÇAMENTE DISTRIBUÍDOS

Marcelo Duffles Donato Moreira

Setembro/2009

Orientador: Otto Carlos Muniz Bandeira Duarte

Programa: Engenharia Elétrica

Os ataques originados por *botnets* são hoje compostos por milhões de estações de ataque, formando o que se chama de ataque de negação de serviço maciçamente distribuído (*Massively Distributed Denial-of-Service* - MDDoS - *attack*). Esse trabalho mostra que os sistemas de rastreamento existentes não são escaláveis para milhões de atacantes, pois a taxa de erro destes sistemas cresce exponencialmente com o número de atacantes. Argumenta-se que a única abordagem escalável para ataques de MDDoS é o rastreamento por um único pacote, que, por natureza, não depende do recebimento de múltiplos pacotes para reconstruir a rota de ataque. O sistema proposto explora a hierarquia cliente-provedor da Internet no nível de sistemas autônomos com o objetivo de fornecer à vítima as informações de rota mais importantes. Essas informações são usadas na construção de pontos de verificação, que guiam de forma acurada o procedimento de reconstrução de rota em direção ao sistema autônomo de origem do ataque. Além disso, os sistemas autônomos clientes são dispensados da tarefa de marcar pacotes, o que facilita a implantação da proposta. O sistema proposto restringe o resultado da descoberta da origem de um pacote de ataque a menos de dois sistemas autônomos candidatos na média, sendo que um deles é certamente o verdadeiro sistema autônomo no qual o pacote foi originado. Os resultados mostram que a proposta de reconstrução de rota baseada em pontos de verificação é cerca de 1000 vezes mais acurada do que um procedimento ótimo que utiliza a clássica abordagem de reconstrução salto-a-salto.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

USING CHECKPOINTS FOR TRACING MASSIVELY DISTRIBUTED DENIAL-OF-SERVICE ATTACKS

Marcelo Duffles Donato Moreira

September/2009

Advisor: Otto Carlos Muniz Bandeira Duarte

Department: Electrical Engineering

Attacks originated from botnets are now composed of millions of machines, creating what we call a *Massively Distributed Denial-of-Service* (MDDoS) attack. This work shows that the existing traceback systems are not scalable to millions of attackers, because the error rate of these systems grows exponentially with the number of attackers. We argue that the unique approach that scales to MDDoS attacks is the single-packet traceback, which does not rely on multiple received packets to reconstruct the attack path. The proposed scheme exploits the customer-provider hierarchy of the Internet at inter-domain level in order to provide the most important path information to the victim. This information is used to build checkpoints that accurately guide the reconstruction procedure towards the origin autonomous system. Furthermore, our scheme relieves customer autonomous systems of marking packets, which makes feasible its deployment. The proposed system narrows the source of an attack packet down to less than two candidate autonomous systems on average, with one of them always being the real autonomous system from which the packet was originated. The results show that the checkpoint-based reconstruction procedure is 1000x more accurate than an optimal procedure that uses the classical hop-by-hop reconstruction approach.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

AIP	*Accountable Internet Protocol*, p. 7
ASN	*Autonomous System Number*, p. 20
AS	*Autonomous System*, p. 4
BGP	*Border Gateway Protocol*, p. 20, 33
CAIDA	*Cooperative Association for Internet Data Analysis*, p. 24
DDoS	*Distributed Denial of Service*, p. 1
DPM	*Deterministic Packet Marking*, p. 9
FAST	*Fast Autonomous System Traceback*, p. 20
FIT	*Fast Internet Traceback*, p. 15
GBF	*Generalized Bloom Filter*, p. 21
ICMP	*Internet Control Message Protocol*, p. 14
IP	*Internet Protocol*, p. 1
IPv4	*Internet Protocol* versão 4, p. 4
ISP	*Internet Service Provider*, p. 6
MAC	*Message Authentication Code*, p. 2
MDDoS	*Massively Distributed Denial of Service*, p. 3
NAT	*Network Address Translation*, p. 8
P2P	*Peer to Peer*, p. 55

PPM *Probabilistic Packet Marking*, p. 9

SPIE *Source Path Isolation Engine*, p. 19

TOS *Type of Service*, p. 35

TTL *Time to Live*, p. 15

XOR *eXclusive OR*, p. 11

Capítulo 1

Introdução

1.1 Motivação

Os ataques de negação de serviço distribuídos (*Distributed Denial-of-Service* - DDoS - *attacks*) são um dos principais desafios de segurança da Internet atualmente [1]. Os atacantes utilizam redes de ataque, chamadas de *botnets*, compostas por máquinas previamente comprometidas denominadas *bots* ou zumbis. Tipicamente, cada estação de ataque gera certa quantidade de tráfego em direção à vítima e o tráfego agregado é então responsável por exaurir os recursos da vítima, de forma a tornar indisponível o serviço oferecido. Os ataques de DDoS somente ocorrem porque os atacantes são capazes de avariar a vítima e ainda assim permanecer anônimos e, consequentemente, impunes [2]. O IP (*Internet Protocol*) não provê autenticação de fonte, e assim pacotes com endereço de origem forjado podem ser injetados na rede. Tal vulnerabilidade é explorada pelos atacantes para garantir seu anonimato através da técnica de falsificação do endereço de origem (*source address spoofing*). Relatos de ataques recentes indicam o uso de endereços forjados [3, 4, 5]. De fato, a maioria das ferramentas automatizadas de ataque usa endereços de origem forjados para inserir um nível de indireção e/ou aumentar o anonimato do atacante [2]. Além disso, um estudo recente [6] mostra que aproximadamente um quarto das redes da Internet permite a falsificação do endereço de origem. Portanto, ainda que as estações de ataque usem endereços de origem legítimos, não é possível provar a participação de uma estação em um ataque. De fato, dado um pacote de ataque, não se tem como determinar se ele foi originado pela estação que possui o endereço

de origem contido no pacote ou se o endereço de origem foi forjado por uma estação que pertence a uma rede que permite a falsificação do endereço de origem. Logo, um mecanismo de identificação de fonte é necessário para associar um pacote de ataque ao seu verdadeiro emissor.

Alguns trabalhos propostos recentemente tentam solucionar o problema de identificação de fonte usando autenticação criptográfica [7, 8]. No presente trabalho, ao invés de usar primitivas criptográficas para autenticar o endereço de origem de pacotes IP, propõe-se o uso de um sistema de rastreamento para localizar a origem do pacote. A ideia básica de um sistema de rastreamento é reconstruir a rota percorrida pelo pacote a partir de marcações ou estado dos roteadores [9, 10, 11, 12]. O rastreamento IP possui diversas vantagens em relação à abordagem baseada em criptografia. Em primeiro lugar, a sobrecarga (*overhead*) de processamento e de cabeçalho é significativamente menor. A maioria dos sistemas de rastreamento usa apenas de 16 a 25 bits para armazenar eficientemente a informação de marcação em campos pouco usados do cabeçalho IP [13]. A sobrecarga de processamento dos algoritmos de marcação de pacotes é essencialmente mais simples do que o cálculo de assinaturas digitais ou de códigos de autenticação de mensagem (*Message Authentication Codes* - MACs). Além disso, os sistemas de rastreamento não desperdiçam o poder de processamento dos roteadores com a validação da origem de tráfegos legítimos, visto que o procedimento de reconstrução de rota é iniciado somente quando necessário.

Embora o rastreamento de pacotes IP apresente diversas vantagens sobre a abordagem baseada em criptografia, até o momento o rastreamento foi pensado somente como um primeiro passo para a defesa contra ataques de DDoS, e não como um mecanismo de identificação de fonte por pacote. Uma das principais razões para isso é que a maioria dos sistemas de rastreamento requer pelo menos dezenas de pacotes recebidos da mesma origem para poder reconstruir a rota de ataque [14]. O sistema proposto nesse trabalho é capaz de reconstruir cada rota de ataque a partir de um único pacote recebido. Dessa forma, a proposta satisfaz a condição necessária para ser usada como um mecanismo de identificação de fonte por pacote.

Nesse trabalho é considerado um cenário não coberto pelos trabalhos anteriores: ataques compostos por milhões de máquinas enviando tráfego a uma vítima

comum [15], denominados ataques de negação de serviço maciçamente distribuídos (*Massively Distributed Denial-of-Service* - MDDoS - *attacks*). O crescimento das redes de ataque, as *botnets*, fez com que elas se tornassem ainda mais poderosas. Estima-se que uma *botnet* composta por 2 milhões de computadores pessoais é capaz de superar o poder de processamento de 500 dos mais poderosos supercomputadores existentes [16], sendo que há estimativas de que a rede do verme *Storm* pode chegar a 50 milhões de zumbis [15]. Isso torna o combate a ataques de MDDoS um grande desafio. Em primeiro lugar, a dimensão das *botnets* obriga os mecanismos de defesa a serem escaláveis para milhões de atacantes. Isso significa que o estado alocado na rede e a taxa de erro do sistema de defesa não podem crescer com o aumento do número de atacantes. Em segundo lugar, para serem robustos a ataques maciçamente distribuídos, os mecanismos de defesa não podem exigir o recebimento de um número mínimo de pacotes de ataque para funcionar. Devido à dimensão gigantesca das *botnets* atuais, as consequências dos ataques maciçamente distribuídos podem ser devastadoras ainda que cada máquina gere apenas uma pequena quantidade de tráfego de ataque. Mesmo no caso extremo no qual cada estação gera apenas 1 pacote de ataque, o tráfego agregado de milhões de estações de ataque é suficiente para negar o serviço provido por um servidor. Por exemplo, em um ataque de 10 milhões de zumbis, cada qual enviando apenas um único pacote de 1500 octetos, se a vítima receber os 10 milhões de pacotes de ataque dentro de um período de 1 segundo, o tráfego agregado médio recebido pela vítima é de 120 Gb/s.

Devido aos grandes desafios introduzidos pelos ataques de MDDoS, os sistemas de rastreamento propostos na literatura são ineficazes contra tais ataques, pois a taxa de erro desses sistemas cresce exponencialmente com o aumento do número de atacantes, conforme mostrado neste trabalho. Além disso, a exigência de um número mínimo de pacotes recebidos compromete a robustez dos sistemas contra ataques nos quais os atacantes enviam um número suficientemente pequeno de pacotes de ataque para evitar serem rastreados, mas ainda assim avariar gravemente a vítima. Mostra-se nesse trabalho que somente uma solução capaz de rastrear a origem do ataque a partir de um único pacote é robusta e escalável para ataques de MDDoS. Porém, a abordagem de um único pacote requer que a informação de rastreamento seja armazenada no cabeçalho de cada pacote, onde não há espaço suficiente para ar-

mazenar a informação completa da rota de ataque. Com isso, podem ocorrer muitos erros durante o procedimento de reconstrução de rota, levando a uma baixa acurácia do sistema de rastreamento. Essa é a razão fundamental pela qual a grande maioria dos pesquisadores optou pelo uso de múltiplos pacotes ou pelo armazenamento de estado adicional nos roteadores.

1.2 Objetivos

Ao que se sabe, o sistema proposto nesse trabalho é a primeira proposta que permite a identificação da origem de um ataque a partir de um único pacote e também satisfaz requisitos práticos, como nenhum estado armazenado nos roteadores e uma sobrecarga de cabeçalho (25 bits) pequena o suficiente para poder ser alocada no cabeçalho IPv4. Para atingir tais objetivos, considera-se o problema do rastreamento no nível de sistemas autônomos (*Autonomous Systems* - ASes). A estrutura hierárquica da Internet no nível de ASes é explorada para localizar o atacante de forma acurada usando apenas as informações fornecidas pelos ASes atravessados pelo pacote de ataque. É usado um procedimento de reconstrução de rota a fim de rastrear o AS de origem do ataque. A cada passo do procedimento de reconstrução de rota, um AS testa seus vizinhos a fim de determinar por qual deles o pacote passou. Foram identificados dois passos críticos nos quais o elevado número de ASes a serem testados representa um ponto de divergência do algoritmo de reconstrução de rota. Nesses dois passos o número de ASes que devem ser testados é da ordem de milhares, o que pode causar uma elevada taxa de falsos positivos, com muitos ASes sendo incorretamente identificados como pertencentes à rota de ataque. Observou-se que pontos de verificação (*checkpoints*) podem ser estrategicamente posicionados a fim de evitar a divergência do algoritmo de reconstrução. Assim, a novidade da proposta deste trabalho é escolher estrategicamente os ASes que marcam o pacote. Propõe-se um esquema de marcação de pacotes que privilegia os passos críticos e dispensa a participação dos ASes-clientes do procedimento de marcação. Como o uso de pontos de verificação evita a divergência do algoritmo de reconstrução, mostra-se que o atacante é localizado com alta acurácia, mesmo usando poucos bits para o armazenamento da informação de rastreamento.

1.3 Organização do Texto

Este trabalho está organizado da seguinte forma. O Capítulo 2 compara a abordagem proposta de rastreamento sem estado e por um único pacote com os trabalhos relacionados. Mostra-se no Capítulo 3 que o espaço disponível para marcação no cabeçalho IPv4 (25 bits) é insuficiente para representar completamente a informação de rota no nível de ASes. Prova-se que, usando somente 25 bits para marcação e assumindo um procedimento de reconstrução salto-a-salto, mesmo uma solução ótima teria uma baixa acurácia no rastreamento do atacante a partir de um único pacote. No Capítulo 4, são apresentadas algumas características particulares observadas nesse trabalho em relação ao problema de reconstrução de rota no nível de ASes, justificando a solução proposta nesse trabalho. Em seguida, o sistema proposto é apresentado em detalhes. São, então, discutidos pontos importantes, como aspectos práticos, robustez e escalabilidade, comparando as características do sistema proposto com as de outras propostas da literatura. Resultados analíticos e de simulação são apresentados no Capítulo 5 e, finalmente, as conclusões no Capítulo 6. Os resultados mostram que o uso de pontos de verificação é realmente uma estratégia vantajosa para contornar a limitação de espaço para marcação. Considerando o mesmo cenário e a mesma sobrecarga de cabeçalho, o procedimento de reconstrução baseado em pontos de verificação obteve uma acurácia 1000 vezes melhor do que o procedimento clássico que utiliza a abordagem de reconstrução salto-a-salto. Além disso, resultados de simulação numa topologia real da Internet evidenciam a excelente acurácia e alta escalabilidade do sistema proposto. O sistema é capaz de rastrear um número ilimitado de atacantes com menos de 1 falso positivo por atacante, com apenas os ASes não-clientes participando da marcação, o que representa aproximadamente 18,5% de razão de implantação.

Capítulo 2

Trabalhos Relacionados

Esta dissertação de mestrado visa identificar a origem de cada pacote enviado pela rede através do uso de um esquema de marcação de pacotes e reconstrução de rota. Assim, os trabalhos da literatura relacionados ao sistema proposto são mecanismos capazes de identificar a verdadeira origem de cada pacote, o que é denominado de o problema de identificação de fonte.

2.1 Filtragem de Pacotes

A solução mais simples para o problema de identificação de fonte é evitar que pacotes com endereço de origem forjado atravessem a rede. Isso pode ser feito com técnicas de filtragem de pacotes, como a filtragem de ingresso (*ingress filtering*) [17]. Essa técnica se baseia no fato de que pacotes com endereço forjado podem ser filtrados pelo roteador próximo à fonte de tráfego, bastando conhecer a faixa de endereços legítimos que podem chegar numa dada interface de rede. Assim, cada provedor de serviço (*Internet Service Provider* - ISP) filtra voluntariamente o tráfego com endereço de origem forjado originado de dentro de sua rede. A filtragem de ingresso, para ser efetiva, requer uma ampla implantação, mas não há nenhum benefício econômico para que um provedor de serviço passe a adotá-la. A adoção da filtragem de ingresso não se traduz em um incremento de segurança imediato para o ISP que a adotou. De fato, se existir um ISP que não realiza filtragem de ingresso, as estações que pertencem à rede desse ISP podem forjar os endereços de outros ISPs, inclusive daqueles que implementam a filtragem de ingresso. Assim,

apesar da filtragem de ingresso ter sido padronizada como uma das melhores práticas da Internet (*Internet Best Current Practice*) há mais de 9 anos, aproximadamente um quarto das redes da Internet permite a falsificação do endereço de origem [6]. A instalação de filtros no núcleo da rede é também fundamentalmente difícil [18]. Portanto, técnicas de filtragem podem ser um mecanismo complementar e útil, mas não a solução completa para o problema.

2.2 Autenticação Criptográfica

Outra abordagem para o problema da identificação de fonte é autenticar a origem dos pacotes usando mecanismos de criptografia. Os trabalhos Passport [7] e AIP (*Accountable Internet Protocol*) [8] seguem essa abordagem. Esses trabalhos utilizam primitivas criptográficas para evitar o uso de endereços forjados introduzindo um mecanismo para verificar a autenticidade do endereço de origem de um pacote no nível de domínio e estação, respectivamente. O Passport funciona da seguinte maneira. Quando um pacote deixa o seu sistema autônomo (AS) de origem, o roteador de borda insere uma lista de MACs (*Message Authentication Codes*) no cabeçalho do pacote. Cada MAC da lista é calculado usando uma chave secreta compartilhada entre o AS de origem e cada AS do caminho a ser percorrido pelo pacote. Em seguida, quando o pacote entra em um AS do caminho, o roteador de borda desse AS verifica o MAC correspondente usando a chave secreta compartilhada com o AS de origem. Um MAC correto só pode ser produzido pelo AS de origem, que conhece a chave secreta. Assim, o roteador de borda calcula o MAC e compara-o com o MAC contido no pacote. Se os MACs não forem iguais, é sinal de que o endereço de origem do pacote é forjado e o pacote pode ser descartado. O AIP (*Accountable Internet Protocol*) é uma proposta de um novo IP no qual os endereços são autocertificados, isto é, são derivados da chave pública da própria estação, podendo ser verificados por qualquer estação sem a necessidade de uma autoridade de certificação global. Apesar de oferecerem um alto grau de segurança, tanto o Passport quanto o AIP possuem limitações práticas que impedem a sua implantação imediata. O Passport possui uma sobrecarga de cabeçalho de 192 bits, que não podem ser alocados no cabeçalho IPv4. Os endereços autocertificados usados pelo AIP são também incom-

patíveis com a versão atual do IP. Além disso, cálculos de MACs por pacote exigem uma sobrecarga de processamento que limita a capacidade de encaminhamento a, no máximo, poucos gigabits por segundo, considerando o *hardware* dos roteadores atuais [7]. Finalmente, adicionar ao núcleo da rede funções restritivas, como a validação/filtragem dos endereços de origem, prejudica mecanismos importantes como a tradução de endereços de rede (*Network Address Translation* - NAT), o *proxying* e o IP móvel (*Mobile* IP), que usam endereços de origem forjados de forma legítima.

2.3 Identificadores Não-Confiáveis

Outra abordagem proposta recentemente é o uso de identificadores não-confiáveis (*unreliable IDs*) para prover a responsabilização das estações [19]. Com IDs não-confiáveis, como identificadores de nível de aplicação, é possível manter rastros das atividades das estações, bastando associar uma estação aos endereços IP usados ao longo do tempo. Porém, no caso de ataques de negação de serviço, as estações de ataque intencionalmente alteram o endereço de origem dos pacotes para ocultar a identidade do atacante ou para realizar ataques por refletor [2]. Dessa maneira, não se pode garantir a autenticidade dos endereços utilizados e, consequentemente, a associação de identificadores não-confiáveis com tais endereços perde o sentido.

2.4 Rastreamento de Pacotes IP

Uma solução promissora para o problema de identificação de fonte é tornar a rede capaz de rastrear o caminho seguido pelos pacotes até a sua verdadeira origem, o que é conhecido como o problema do rastreamento IP [9]. Essa solução foi proposta recentemente às Nações Unidas com o objetivo de definir métodos para rastrear cada pacote que circula na Internet [20]. As técnicas de rastreamento podem ser divididas em duas classes: técnicas baseadas em marcação de pacotes e técnicas baseadas em auditoria [1]. A ideia básica da marcação de pacotes é fazer com que cada roteador insira informações sobre si mesmo nos pacotes encaminhados. Assim, após receber pacotes suficientes, a vítima pode reconstruir a rota percorrida pelo pacote usando as informações fornecidas pelos roteadores. Já nos esquemas baseados em auditoria, os roteadores armazenam informações sobre os pacotes encaminhados.

Dessa forma, a vítima pode consultar os roteadores para verificar se um dado pacote foi encaminhado ou não pelo roteador consultado. A seguir são apresentados os principais sistemas de rastreamento propostos na literatura.

2.4.1 Marcação de Pacotes

Nos esquemas baseados em marcação de pacotes, alguns campos pouco usados do cabeçalho IP são sobrecarregados a fim de armazenar as marcações dos roteadores. Essa informação de rastreamento poderia ser armazenada no campo de opções do IP, mas isso poderia levar à necessidade de fragmentação do pacote, além de gerar uma sobrecarga de processamento elevada, visto que a adição de opções ao pacote obriga que o encaminhamento do pacote tenha um processamento mais lento (*slow path*). Outra possibilidade seria enviar a informação de rastreamento em um pacote separado, mas isso adicionaria ainda mais sobrecarga de processamento no roteador e reduziria a banda disponível na rede. Portanto, a solução mais eficiente é sobrecarregar campos poucos usados do cabeçalho IP, como o campo de identificação de fragmento (16 bits) [9]. Alguns estudos mostram que menos de 0,25% dos pacotes da Internet são pacotes que sofreram fragmentação [21]. Assim, o impacto negativo da adoção do sistema de rastreamento é pequeno quando comparado aos benefícios trazidos.

O princípio básico da marcação de pacotes é transferir para a vítima a informação de rastreamento necessária para a reconstrução da rota de ataque usando o espaço disponível no cabeçalho IP. A marcação de pacotes pode ser probabilística (*Probabilistic Packet Marking* - PPM) ou determinística (*Deterministic Packet Marking* - DPM). Nos esquemas probabilísticos, cada roteador decide se marca ou não um pacote de acordo com uma dada probabilidade p. Já nos esquemas determinísticos, os roteadores que implementam o sistema de rastreamento sempre marcam os pacotes encaminhados. A seguir são apresentados os esquemas probabilísticos e determinísticos, mostrando suas vantagens e desvantagens.

Marcação de Pacotes Probabilística

A marcação probabilística permite que, após receber um número suficiente de pacotes de ataque, a vítima tenha garantias estatísticas de ter recebido as marcações

de todos os roteadores que compõem a rota de ataque. Tendo recebido todas essas marcações, a rota de ataque pode então ser reconstruída. O cálculo do número de pacotes necessários para receber ao menos uma marcação de cada roteador de uma rota de ataque de tamanho n é equivalente ao cálculo do número médio de tentativas que devem ser feitas para se retirar ao menos um cupom de cada tipo de uma urna com n tipos de cupons, o que é conhecido como o problema do coletor de cupons [9]. Assim, o número médio $E(X)$ de pacotes necessários é limitado superiormente segundo a expressão

$$E(X) \leq \frac{\ln(n)}{p(1-p)^{n-1}}. \tag{2.1}$$

Na prática, os sistemas propostos dividem ainda a marcação de cada roteador em k fragmentos. Nesse caso, para receber todos os k fragmentos de todos os n roteadores que compõem a rota de ataque, o número médio $E(X')$ de pacotes necessários passa a ser [9]

$$E(X') \leq \frac{k.\ln(k.n)}{p(1-p)^{n-1}}. \tag{2.2}$$

Por exemplo, considerando uma rota de 10 saltos e os valores comumente usados $p = 0,04$ e $k = 8$, a Equação 2.2 resulta em aproximadamente 1265 pacotes. A Figura 2.1 ilustra a ordem com que os pacotes são marcados durante a travessia dos pacotes até chegar à vítima. Cada roteador decide se marca ou não o pacote com probabilidade p. Um mesmo pacote pode ser marcado mais de uma vez pelos roteadores da rota de ataque, sendo que as marcações anteriores são sobrescritas pela marcação subsequente. Dessa forma, quanto mais próximo à vítima, maior é a probabilidade de um pacote ser marcado pelo roteador e permanecer intacto até chegar à vítima, conforme ilustrado na figura. A probabilidade de um pacote ser marcado pelo roteador R_i e nenhum outro roteador subsequente sobrescrever essa marcação até que o pacote chegue à vítima é $p(1-p)^{n-i}$. Nota-se também que um pacote pode chegar à vítima sem ter sido marcado por nenhum roteador da rota de ataque. A probabilidade de ocorrer tal evento é $(1-p)^n$. Os pacotes que chegam à vítima sem nenhuma marcação podem ser maliciosamente ajustados pelo atacante de forma a criar erros durante a reconstrução de rota e, assim, prejudicar a acurácia do sistema de rastreamento [22]. Então, para aumentar a robustez do sistema à interferência do atacante, é necessário aumentar o valor de p, mas isso implica um

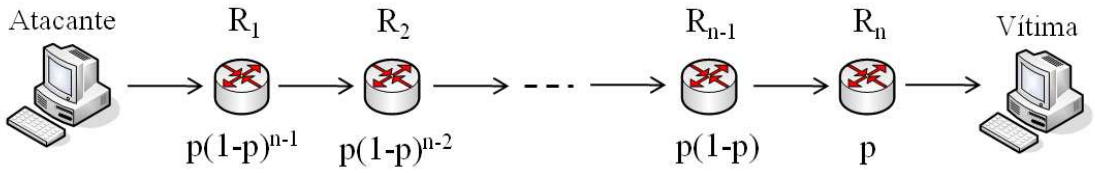


Figura 2.1: Na marcação probabilística, os roteadores marcam pacotes com probabilidade p.

maior número de pacotes necessários para a reconstrução de rota, conforme indica a Equação 2.2. Assim, existe um compromisso entre a interferência do atacante e a quantidade de pacotes necessários para a reconstrução da rota.

Savage *et al.* [9] propuseram um esquema probabilístico de marcação de pacotes no qual os roteadores inserem informações de enlace nos pacotes. A informação de enlace é composta pelos endereços IP dos roteadores conectados pelo enlace. Assim, após receber um número suficiente de pacotes, a vítima dispõe dos endereços de todos os roteadores dos enlaces que compõem a rota de ataque. Além da informação de enlace, a distância do roteador que marcou o pacote até a vítima também é carregada no cabeçalho pacote, conforme ilustrado na Figura 2.2. Ao marcar um pacote, o roteador zera o campo de distância e os roteadores seguintes simplesmente incrementam esse campo ao encaminhar o pacote. Essa informação de distância é usada para ordenar os enlaces durante a reconstrução de rota. O espaço necessário para armazenar a informação de rastreamento é de 69 bits: dois endereços IP de 32 bits para compor a informação de enlace mais 5 bits para o campo de distância[1]. Porém, nesse esquema somente os 16 bits do campo de identificação de fragmento do cabeçalho IP são usados para armazenar a informação de rastreamento. Para reduzir o espaço necessário de 69 para 16 bits, a marcação de cada roteador é dividida em k fragmentos de endereço e também é empregada uma técnica codificação baseada na operação binária de OU exclusivo (*eXclusive OR* - XOR). Conforme pode ser visto na figura, ao marcar um pacote, o roteador insere um dos $k = 8$ fragmentos do seu endereço no campo `informação de enlace`. Consequentemente, esse esquema obtém uma pequena sobrecarga de cabeçalho ao custo de requerer um

[1]Teoricamente, o valor máximo de número de saltos permitido pelo IP é 255. Porém, como praticamente todas rotas da Internet possuem menos de 32 saltos [23], 5 bits são suficientes para representar a informação de distância.

elevado número de pacotes recebidos para recuperar a informação de enlace de toda a rota de ataque. Além disso, considerando um ataque de somente 25 atacantes, o tempo de processamento necessário para reconstruir as rotas de ataque pode chegar a dias e o número de falsos positivos chega a milhares [10].

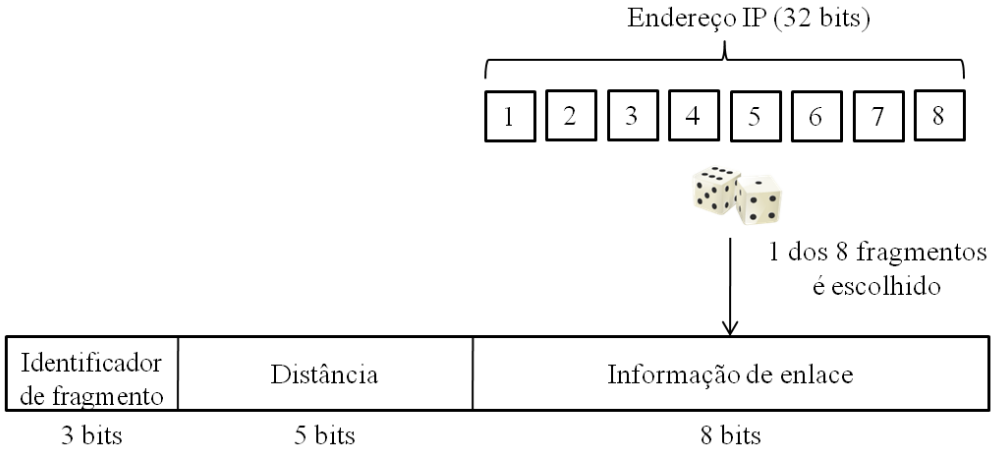
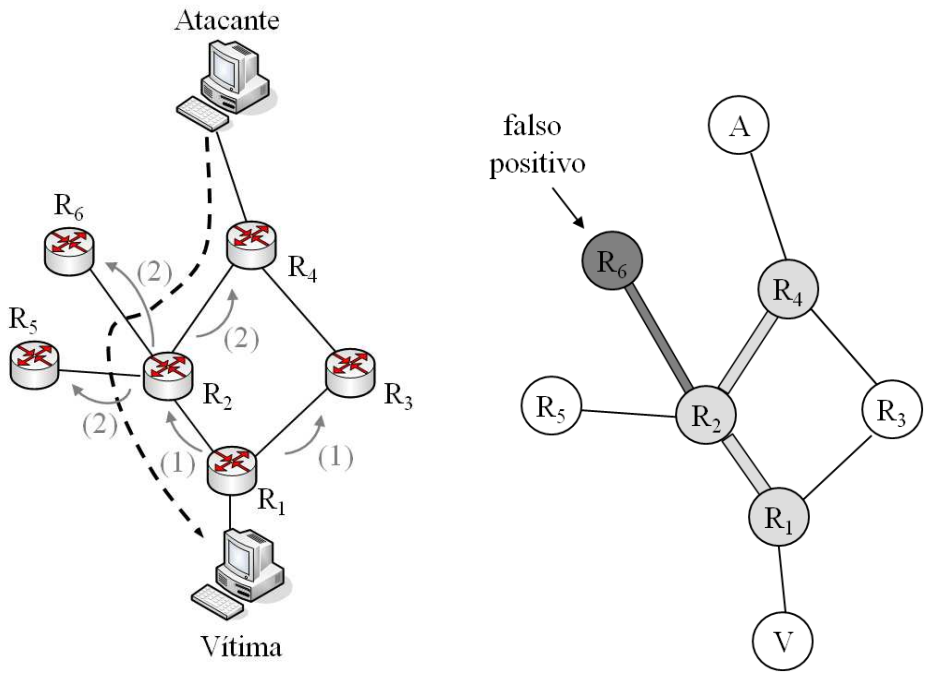


Figura 2.2: Campos de cabeçalho usados para marcação no sistema de Savage *et al.*.

Song e Perrig [10] propuseram um esquema probabilístico aperfeiçoado, assumindo o conhecimento do mapa da topologia. Com essa suposição, não é mais necessário saber o endereço completo dos roteadores para reconstruir a rota de ataque. No esquema de Song e Perrig, a rota de ataque não é mais reconstruída através da recuperação dos endereços dos roteadores pelos quais o pacote passou, mas sim através de uma busca no grafo que representa a topologia de rede. É realizada uma busca em largura[2] partindo do roteador mais próximo à vítima. A Figura 2.3(a) ilustra a reconstrução da rota de ataque (R_4, R_2, R_1). Inicialmente, no passo 1 da figura, são testados os roteadores R_2 e R_3, que são vizinhos de R_1, o roteador mais próximo à vítima. Para decidir por qual vizinho o pacote passou, não é necessário saber o endereço completo de R_2, mas basta saber distingui-lo corretamente dentre os vizinhos testados. Após identificar o roteador correto, R_2, a busca continua testando os vizinhos de cada roteador identificado até chegar ao roteador mais próximo ao atacante. Assim, no passo 2, são testados os roteadores R_4, R_5 e R_6. Finalmente, o roteador R_4 é identificado e o procedimento de reconstrução de rota termina. Chama-se de grafo de reconstrução, o grafo resultante da incorporação dos

[2]A busca em largura, em contraposição à busca em profundidade, testa primeiro os nós-irmãos e depois os nós-filhos.

roteadores identificados pelo procedimento de reconstrução de rota. Formalmente, o grafo de reconstrução é o subgrafo induzido pelo conjunto de vértices que representam os roteadores identificados pelo procedimento de reconstrução de rota. Assim, o grafo de reconstrução desse exemplo contém os vértices R_1, R_2 e R_4 e as arestas que os interligam, conforme ilustrado na Figura 2.3(b). Durante o procedimento de reconstrução de rota, um roteador que não pertence à rota percorrida pelo pacote pode ser incorretamente integrado ao grafo de reconstrução. Esse tipo de erro é chamado de *falso positivo*. Por exemplo, se durante o passo 2 do procedimento de reconstrução o roteador R_6 fosse erroneamente identificado como sendo pertencente à rota de ataque, ele também seria incorporado ao grafo de reconstrução. Como na realidade R_6 não pertence à rota de ataque, então se trata de um falso positivo, conforme mostrado na Figura 2.3(b).



(a) Procedimento de reconstrução. (b) Grafo de reconstrução.

Figura 2.3: Reconstrução de rota através de uma busca no grafo que representa a topologia de rede.

No esquema de Song e Perrig, para marcar os pacotes, os roteadores usam uma função *hash*[3]. A marcação inserida no pacote é o *hash* do endereço IP do roteador. Assim, ao invés de um endereço de 32 bits, a marcação foi reduzida para um iden-

[3]Uma função *hash*, também chamada de função de espalhamento ou função resumo, é definida

tificador de tamanho fixo. Isso elimina a necessidade de dividir o endereço IP em fragmentos. A Figura 2.4 mostra os campos usados para marcação. Assim, como no esquema de Savage *et al.*, 16 bits do cabeçalho IP são sobrecarregados para armazenar a informação de rastreamento. Um campo de distância de 5 bits também é usado. Assim, restam $16 - 5 = 11$ bits para representar o *hash* do endereço IP do roteador. Para reduzir a probabilidade de falso positivo durante a reconstrução de rota, ao invés de uma única função *hash*, são usadas f funções *hash* independentes. A intuição é que a probabilidade de um falso positivo R_a ter o mesmo *hash* que um roteador R_b para uma função *hash* é $1/2^{11}$, mas a probabilidade de R_a ter os mesmos *hashes* que R_b para f funções *hash* independentes é $(1/2^{11})^f = 1/2^{11f}$. Com essa modificação, quando um roteador for marcar o pacote, ele deve escolher uma dentre as f funções *hash* para ser utilizada. Quando a vítima for reconstruir a rota de ataque, ela precisa saber qual função *hash* o roteador usou para marcar um dado pacote. Por isso, os 11 bits são divididos em um campo de 3 bits para armazenar o identificador da função *hash* usada e em outro campo de 8 bits para armazenar o *hash* do endereço do roteador, conforme mostrado na figura.

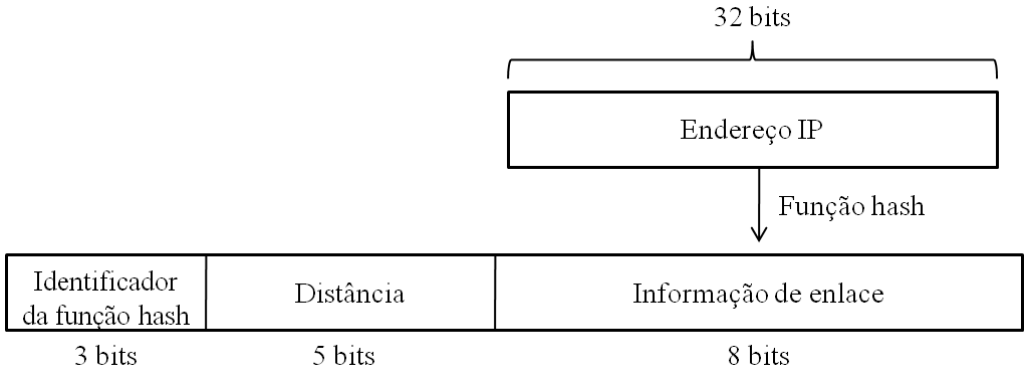


Figura 2.4: No esquema de Song e Perrig, o roteador sorteia uma dentre f funções *hash* para marcar o pacote.

Bellovin *et al.* [24] propuseram um esquema probabilístico que utiliza mensagens ICMP (*Internet Control Message Protocol*) para enviar a informação de rastreamento à vítima. Nesse esquema, cada roteador probabilisticamente seleciona um

como um mapeamento $h : \{0,1\}^* \rightarrow \{0,1\}^n$ de um conjunto de tamanho arbitrário $\{0,1\}^*$ em um conjunto $\{0,1\}^n$ de tamanho fixo (n bits). Nesse caso, a propriedade requerida é que as saídas da função *hash* sejam uniformemente distribuídas ao longo do conjunto-imagem $\{0,1\}^n$.

pacote e cria uma mensagem ICMP contendo informações sobre si mesmo. Essa abordagem de envio de informação de rastreamento fora da banda possui a vantagem de poder usar muito mais bits do que a marcação dentro da banda, que é limitada pelo pequeno espaço disponível para marcação no cabeçalho IP. Portanto, a taxa de falsos positivos e o número de pacotes necessários para a reconstrução de rota são significativamente reduzidos. Entretanto, exigindo uma banda adicional para enviar mensagens ICMP, este esquema pode amplificar os efeitos do ataque de negação de serviço no desempenho da rede. Além disso, a filtragem de mensagens ICMP pode levar à falha do sistema. Por fim, essa abordagem ainda requer um mecanismo de autenticação de mensagens para evitar que mensagens ICMP forjadas atrapalhem a reconstrução de rota.

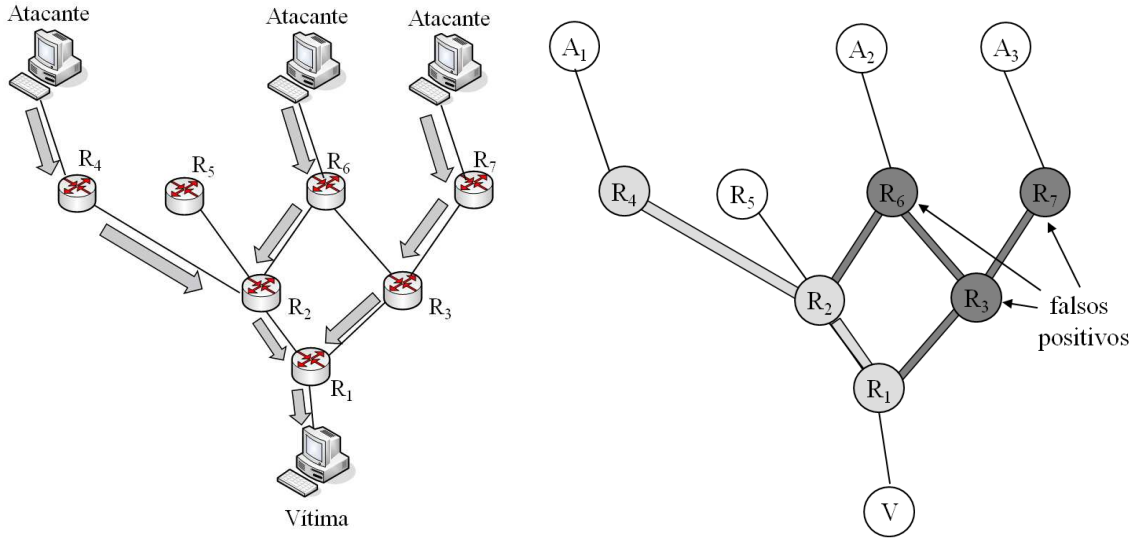
Dean *et al.* [13] propuseram um esquema de marcação probabilística no qual os endereços IP dos roteadores são codificados como um polinômio. A ideia básica é que uma função polinomial de grau g pode ser recuperada a partir da interpolação de $g + 1$ pontos distintos. Assim, cada pacote carrega um ponto de uma função polinomial cujas variáveis desconhecidas são os endereços IP dos roteadores. Após o recebimento de um número suficiente de pacotes, a vítima emprega métodos algébricos para resolver um sistema de equações e recuperar os endereços IP dos roteadores. No entanto, esse esquema não é escalável para um grande número de atacantes, porque o número de pacotes exigidos para a reconstrução de rota de cada atacante depende linearmente do número de atacantes, como mostrado em [14].

Yaar *et al.* [14] propuseram o sistema probabilístico FIT (*Fast Internet Traceback*). A novidade dessa proposta é o armazenamento da informação de distância usando apenas um único bit. A proposta é alterar a semântica do campo TTL (*Time to Live*) do cabeçalho IP. Como o TTL é decrementado por todo roteador da Internet, então esse campo é usado para carregar a informação de distância. Assim, ao marcar um pacote, o roteador sobrescreve os últimos 5 bits do campo TTL com um valor constante globalmente conhecido. A vítima consegue inferir a distância até o roteador que marcou o pacote calculando a diferença entre o valor constante e o valor do TTL recebido. Reduzindo o tamanho do campo de distância de 5 para 1 bit, sobram 4 bits adicionais para armazenar o *hash* do endereço do roteador. Com isso, esse esquema consegue uma redução do número de falsos positivos, permitindo

rastrear até alguns milhares de atacantes. Apesar de considerável, essa melhoria é insuficiente para lidar com ataques maciçamente distribuídos. A observação-chave é que qualquer sistema probabilístico depende do recebimento de múltiplos pacotes para recuperar a informação completa da rota de ataque. O uso de múltiplos pacotes possui um problema de escalabilidade intrínseco, conforme mostrado a seguir.

Embora engenhosos, os esquemas probabilísticos não são escaláveis para ataques de MDDoS. A razão é que tais esquemas não possuem um mecanismo para evitar uma combinação incorreta da informação de rastreamento proveniente de rotas de ataque distintas. De fato, dado um pacote de ataque, não se tem como determinar se ele é proveniente de uma ou outra rota. Sendo assim, a marcação contida no pacote é testada em *todas* as rotas em reconstrução. Para a reconstrução da rota pela qual o pacote realmente passou, essa marcação é benéfica, pois ajudará a identificar corretamente os roteadores dessa rota de ataque. No entanto, para a reconstrução das outras rotas de ataque, essa marcação é maléfica, pois indicará roteadores que não pertencem à rota em reconstrução, gerando falsos positivos. Esse efeito aumenta com o aumento do número de atacantes, comprometendo a escalabilidade do sistema de rastreamento. A Figura 2.5 ilustra esse problema. Na Figura 2.5(a) é apresentado um exemplo de ataque distribuído. Nesse exemplo, há 3 atacantes enviando tráfegos de ataque que seguem por caminhos distintos até chegar à vítima. Como não se tem como determinar por qual rota chegou um dado pacote de ataque, então todas as marcações recebidas são usadas para a reconstrução de todas as rotas de ataque. A Figura 2.5(b) mostra o resultado final da reconstrução da rota (A_1, R_4, R_2, R_1, V). Nota-se que os roteadores R_3, R_6 e R_7 foram incorretamente integrados ao grafo de reconstrução, sendo considerados falsos positivos para o rastreamento do atacante A_1. Além disso, o exemplo representa um caso simplificado, no qual não são considerados os falsos positivos causados pelo uso das marcações das rotas dos atacantes A_2 e A_3 na reconstrução da rota do atacante A_1. Sejam q a probabilidade de ocorrer um falso positivo usando uma única marcação e $\Psi(d)$ o número de marcações distintas com distância d recebidas pela vítima. Assim, considerando as marcações das rotas dos atacantes A_2 e A_3 na reconstrução da rota do atacante A_1, a probabilidade de falso positivo passa a ser $q.\Psi(d)$. Quando há muitos atacantes, a vítima recebe muitas marcações com uma mesma distância d, acarretando um elevado va-

lor de $\Psi(d)$. Portanto, a probabilidade de falso positivo cresce com o aumento do número de atacantes. De fato, conforme é mostrado nos resultados de simulação (Capítulo 5), a acurácia dos sistemas probabilísticos é bastante degradada com o aumento do número de atacantes.



(a) Ataque distribuído. (b) Grafo de reconstrução da rota (A_1, R_4, R_2, R_1, V).

Figura 2.5: Rastreamento do atacante A_1 considerando um ataque distribuído.

Além do problema de escalabilidade, os esquemas probabilísticos são vulneráveis à interferência do atacante por dois motivos. Primeiro porque uma fração dos pacotes recebidos pela vítima não são marcados pelos roteadores, estando, portanto, sob o controle do atacante, que pode marcar o que desejar. O segundo motivo é a exigência de um número mínimo de pacotes recebidos para poder iniciar o procedimento de reconstrução. Dessa forma, os atacantes podem avariar a vítima e ainda assim permanecer não-rastreáveis, bastando enviar um número suficientemente pequeno de pacotes para evitar serem rastreados.

Marcação de Pacotes Determinística

Uma solução simples e eficiente para o problema de identificação de fonte é a marcação de pacotes determinística [25]. Belenky e Ansari observaram que, sendo o objetivo final a identificação do atacante e não da rota completa de ataque, somente o primeiro roteador da rota de ataque precisa marcar o pacote. Assim, a proposta é que o roteador mais próximo do atacante insira no pacote o endereço IP da sua

interface de entrada. Como os 32 bits do endereço IP não cabem dentro do espaço disponível para marcação, o endereço IP é dividido em k fragmentos de endereço, assim como no esquema de Savage *et al.*. Ao marcar o pacote, o roteador escolhe aleatoriamente um dos k fragmentos e o insere no pacote. Após o recebimento de todos os fragmentos, a vítima pode recuperar o endereço do primeiro roteador, bastando agrupar corretamente os fragmentos de endereço recebidos. Considerando um ataque com um único atacante, bastam poucos pacotes recebidos para que a vítima seja capaz de recuperar o endereço do roteador e, assim, o sistema funcionar bem. Porém, considerando um ataque distribuído, ocorre o mesmo problema dos esquemas probabilísticos: os fragmentos do endereço de um atacante são confundidos com os fragmentos dos endereços dos demais atacantes. Para resolver esse problema, na tentativa de tornar o sistema escalável para múltiplos atacantes, são usadas f funções *hash* independentes. Um conjunto de k fragmentos só é aceito como um endereço IP válido se os f *hashes* coincidirem. Nesse caso, conforme mostrado pelos autores, o número esperado de falsos positivos, $E[P]$, é expresso por

$$E[P] = \frac{\left[2^a - 2^a\left(1 - \frac{1}{2^a}\right)^{\frac{N}{E[H]}}\right]^{f.k}}{2^{32(f-1)}}, \tag{2.3}$$

onde $a = 32/k$ é o número de bits de cada fragmento de endereço, N é o número de atacantes e $E[H]$ é o número médio de *hashes* distintos que chegam à vítima, dado por

$$E[H] = 2^d - 2^d\left(1 - \frac{1}{2^d}\right)^N, \tag{2.4}$$

onde d é o tamanho (em bits) da saída das funções *hash*. O custo do uso de múltiplos fragmentos é o aumento da quantidade de informação tranferida à vítima e, consequentemente, o aumento do número de pacotes necessários para o rastreamento. De fato, considerando um ataque com cerca de 2 mil atacantes, o esquema requer o recebimento de pelo menos 55 pacotes da mesma origem para poder recuperar o endereço IP do primeiro roteador. Portanto, da mesma maneira que os sistemas probabilísticos, a abordagem determinística falha na presença de ataques maciçamente distribuídos, porque cada atacante pode enviar poucos pacotes para evitar ser rastreado. Além disso, o número de falsos positivos tem um crescimento abrupto a partir de poucos milhares de atacantes, conforme é mostrado no Capítulo 5. Em

comparação, a proposta desse trabalho é capaz de rastrear um número ilimitado de atacantes com menos de um falso positivo por atacante, usando apenas um único pacote recebido.

2.4.2 Rastreamento Baseado em Auditoria

Snoeren *et al.* [12] introduziram a abordagem de rastreamento baseada em auditoria. A proposta, chamada SPIE (*Source Path Isolation Engine*), utiliza filtros de Bloom [26] para armazenar de forma compacta a informação de auditoria nos roteadores. Durante o encaminhamento de um pacote, alguns dados do pacote são recolhidos e inseridos no filtro de Bloom, gerando uma espécie de "assinatura" do pacote, que pode ser verificada posteriormente. Assim, a vítima pode consultar os roteadores a fim de determinar se um dado pacote passou pelo roteador ou não. O problema dessa abordagem é que, apesar do uso de filtros de Bloom, o estado armazenado nos roteadores aumenta linearmente com o número de pacotes transmitidos. Portanto, a enorme capacidade de armazenamento exigida torna difícil a implantação dessa abordagem na prática, principalmente em enlaces de alta velocidade.

Choi e Dai [27] usam uma abordagem de rastreamento híbrida na qual os roteadores inserem a informação de rastreamento no pacote até que o limite de espaço no cabeçalho IP seja atingido, quando então a informação de rastreamento passa a ser guardada na memória de um roteador e o espaço do cabeçalho IP é novamente liberado para futuras marcações. Outra novidade é que a informação de rastreamento não é composta pelos endereços dos roteadores, mas por códigos que representam os enlaces. Assim, cada roteador mantém uma tabela local que associa cada enlace ao seu código. Os enlaces são representados usando um algoritmo de codificação sem perdas (ver Apêndice A) chamado de código de Huffman, que atribui códigos de acordo com a frequência de ocorrência de cada entrada. Assim, os enlaces mais frequentes são representados por uma palavra de código pequena, enquanto que os enlaces pouco usados são representados por uma palavra de código maior. Os autores mostram que, para uma rota média da Internet (15 saltos), 2 dos 16 roteadores da rota devem armazenar a informação de rastreamento em memória, o que significa que em média 12,5% dos pacotes encaminhados por um roteador são armazenados

em memória.

2.4.3 Rastreamento Inter-domínio

Recentemente, pesquisadores observaram as vantagens do rastreamento inter-domínio [28, 29]. As rotas em nível de sistemas autônomos (*Autonomous Systems* - ASes) são cerca de 5 vezes menores que as rotas no nível de roteadores e o número de ASes (aproximadamente 45 mil) é bem menor do que o número de roteadores (da ordem de dezenas de milhões) e de estações (aproximadamente 1,6 bilhão). Além disso, a informação do roteamento inter-domínio é bastante útil para o rastreamento. Por exemplo, Gao e Ansari [28] usam a informação do atributo `AS_PATH` do protocolo de roteamento BGP (*Border Gateway Protocol*) para inferir a distância de cada AS em relação ao destino do pacote, permitindo assim o projeto de um esquema probabilístico ótimo no qual nenhum pacote chega à vítima sem ser marcado por algum AS da rota de ataque.

Paruchuri e Durresi [29] propuseram o FAST (*Fast Autonomous System Traceback*), um sistema determinístico no qual os 5 primeiros ASes da rota de ataque marcam o pacote[4]. O espaço de marcação é dividido em 5 subcampos para acomodar as 5 marcações permitidas. Um contador também é carregado no pacote para informar qual subcampo um AS deve marcar. A marcação é feita pelo roteador de borda de cada AS. Para notificar a vítima sobre a presença do AS na rota de ataque, é usado o identificador de AS no roteamento inter-domínio, chamado de número de AS (*AS number* - ASN). Ao marcar o pacote, o roteador de borda do AS insere o *hash* do ASN correspondente. Assim como o sistema de Song e Perrig, o FAST realiza uma busca no grafo que representa a topologia de rede para reconstruir a rota de ataque. São também utilizadas múltiplas funções *hash* para reduzir a probabilidade de falso positivo durante a reconstrução de rota. Apesar disso, o sistema FAST não é escalável para ataques maciçamente distribuídos, porque o número de falsos positivos cresce rapidamente com o número de atacantes, conforme mostram os resultados de simulação do Capítulo 5. A razão principal é que, assim como todos os sistemas baseados na marcação de pacotes apresentados anteriormente, o FAST depende do recebimento de múltiplos pacotes para reconstruir a rota de ataque, o

[4] As estatísticas mostram que 99,5% das rotas em nível de ASes possuem menos de 6 saltos [30].

que implica um alto número de erros de reconstrução causados pela combinação incorreta da informação de rastreamento proveniente de rotas de ataque distintas.

2.4.4 Comparação dos Sistemas de Rastreamento

A habilidade de rastrear o atacante a partir de um único pacote é uma condição necessária para (i) escalar para ataques de MDDoS e (ii) usar o sistema de rastreamento como um mecanismo de identificação de fonte por pacote. Entretanto, a maioria dos sistemas de rastreamento requer cerca de milhares de pacotes recebidos da mesma origem para poder reconstruir a rota de ataque [9, 10, 13]. Alguns trabalhos conseguiram reduzir esse número para dezenas de pacotes [14, 25], mas até o presente momento somente os esquemas baseados em auditoria e o esquema proposto por Laufer *et al.* [11] possuem a habilidade de rastrear o atacante a partir de um único pacote. Não obstante, nenhum desses esquemas atende a todos os requisitos que uma solução prática deve satisfazer. Os esquemas baseados em auditoria requerem armazenamento de estado por pacote na infraestrutura de rede, o que não é viável para redes gigabit. Laufer *et al.* deram um passo adiante no desenvolvimento da abordagem de rastreamento por um único pacote ao introduzir o chamado filtro de Bloom generalizado (*Generalized Bloom Filter* - GBF). O GBF permite a reconstrução de rota a partir de um único pacote de forma robusta e eficiente sem armazenar nenhum estado na infraestrutura de rede. Porém, uma sobrecarga de cabeçalho de algumas centenas de bits é necessária para localizar o atacante de forma acurada [11]. O problema do rastreamento por um único pacote é abordado com mais detalhes no Capítulo 3.

A Tabela 2.1 compara os sistemas de rastreamento apresentados nessa seção. Pode-se notar que a maioria dos sistemas propostos requer múltiplos pacotes para funcionar. Por essa razão, conforme argumentado anteriormente, tais sistemas não escalam para ataques maciçamente distribuídos. A única abordagem na qual o número de falsos positivos não cresce com o aumento do número de atacantes é a abordagem de rastreamento por um único pacote, na qual não há erros de reconstrução causados pela combinação incorreta da informação de rastreamento de diferentes atacantes. Pode-se observar que a proposta desse trabalho é a única que escala para ataques de MDDoS e ao mesmo tempo satisfaz requisitos práticos, como

Tabela 2.1: Comparação dos sistemas de rastreamento.

	Pacotes exigidos	**Escala para ataques de MDDoS?**	**Sobrecarga de cabeçalho**	**Estado nos roteadores**
Proposto	1	Sim	25 bits	Nenhum
Laufer *et al.*	1	Sim	192-256 bits	Nenhum
Snoeren *et al.*	1	Sim	Nenhuma	Estado por pacote
Choi e Dai	1	Sim	16 bits	12,5% dos pacotes
Gao e Ansari	6-9	Não	32 bits	Nenhum
Paruchuri e Durresi	8-10	Não	25 bits	Nenhum
Belenky e Ansari	55-130	Não	17 bits	Nenhum
Bellovin *et al.*	Dezenas	Não	Nenhuma	Nenhum
Yaar *et al.*	10-1000	Não	16 bits	Nenhum
Dean *et al.*	Milhares	Não	25 bits	Nenhum
Song e Perrig	500-2000	Não	16 bits	Nenhum
Savage *et al.*	1000-8000	Não	16 bits	Nenhum

nenhum estado armazenado nos roteadores e uma pequena sobrecarga de cabeçalho.

Notação Empregada

Nesse trabalho, considera-se o problema do rastreamento no nível de sistemas autônomos (ASes). Sendo assim, no restante dessa dissertação, é considerada como rota de ataque a sequência de ASes atravessados pelo pacote até chegar ao destino. O AS no qual o pacote de ataque foi originado é denominado AS-atacante ou simplesmente atacante. Da mesma forma, o AS de destino do pacote é chamado de AS-vítima ou simplesmente vítima. A reconstrução de rota é feita segundo o mesmo algoritmo apresentado neste capítulo, só que os nós do grafo agora são ASes, ao invés de roteadores.

Capítulo 3

Solução Ótima para Rastreamento por um Único Pacote

A análise desse capítulo considera uma abordagem de rastreamento sem estado e por um único pacote, isto é, que toda a informação de rastreamento necessária para a reconstrução de rota deve estar contida no cabeçalho de cada pacote. Define-se como solução ótima para o problema do rastreamento sem estado e por um único pacote o esquema que é capaz de reconstruir *todas* as rotas em nível de sistemas autônomos usando o mínimo de espaço de cabeçalho possível. Por isso, a análise a seguir utiliza resultados da teoria da informação [31], especialmente da teoria da distorção de taxa (*rate distortion theory*), a fim de modelar matematicamente o problema. Os conceitos básicos de teoria da informação e as demontrações dos resultados usados nesse capítulo são apresentados no Apêndice A.

3.1 Análise da Sobrecarga de Cabeçalho

A razão fundamental da existência de falsos positivos durante a reconstrução de rota é a insuficiência de espaço disponível para marcação no cabeçalho IP. Se houvesse espaço suficiente para representar cada sistema autônomo (AS) da rota de ataque, então em cada passo do procedimento de reconstrução de rota seria possível identificar inequivocamente, dentre os ASes testados, o AS pelo qual o pacote passou. Porém, com um espaço insuficiente para representar a rota de ataque, é inevitável que dois ou mais ASes possuam uma mesma marcação, gerando falsos positivos no

momento da reconstrução de rota. Portanto, a acurácia da reconstrução de rota depende do espaço disponível para marcação. Quanto mais bits disponíveis, mais informação de rastreamento pode ser representada e, em consequência, os ASes da rota de ataque são identificados com maior acurácia. São apresentadas a seguir duas análises da sobrecarga de cabeçalho requerida para o rastreamento em nível de ASes.

3.1.1 Sobrecarga de Cabeçalho Requerida para Rastreamento sem Erros

A primeira análise realizada avalia a quantidade mínima de espaço para marcação necessária para reconstruir a rota de ataque sem nenhum erro. Para tanto, em cada passo do procedimento de reconstrução de rota, a vítima deve ser capaz de distinguir, dentre os ASes testados, o AS pelo qual o pacote passou. Essa distinção é feita usando a marcação contida no pacote recebido. Assim, o procedimento de marcação deve atribuir um código único para cada vizinho de um dado AS. No primeiro passo do procedimento de reconstrução, são testados N_1 ASes vizinhos e somente um desses é o correto. Logo, são necessários N_1 códigos distintos para determinar inequivocamente o AS correto, isto é, o próximo AS da rota de ataque. Após a identificação desse AS, os vizinhos desse AS são testados no passo seguinte. Novamente, N_2 códigos distintos são necessários para identificar o próximo AS da rota de ataque, e assim por diante. De forma geral, são necessários pelo menos $\lceil log_2(N_i) \rceil$ bits para identificar de forma única um AS dentre N_i ASes vizinhos, considerando uma abordagem de codificação sem perdas. Portanto, se n é o tamanho da rota de ataque, então a quantidade de bits m necessária para representar a rota completa é

$$m = \sum_{i=1}^{n} \lceil \log_2 (N_i) \rceil. \tag{3.1}$$

A fim de estimar o valor de N_i em uma topologia real, foi implementado um simulador de reconstrução de rota. Foi usada uma topologia em nível de ASes construída a partir de dados de medição disponibilizados pela CAIDA (*Cooperative Association for Internet Data Analysis*). Os dados foram obtidos com a infraestrutura de medição Ark (*Archipelago*) [32]. O objetivo foi medir o número de ASes que devem ser testados pelo procedimento de reconstrução em função da distância em relação

Tabela 3.1: Número de testes em cada passo do procedimento de reconstrução.

Distância à vítima (i)	**Número de ASes testados (N_i)**
1 salto	$6,3 \pm 0,8$
2 saltos	$898,5 \pm 16,2$
3 saltos	$1489,6 \pm 11,0$
4 saltos	$194,6 \pm 7,6$
5 saltos	$5,9 \pm 0,8$

à vítima. Considerou-se a reconstrução de todos os possíveis caminhos mais curtos que têm como destino um dado AS. Em cada rodada de simulação, era variado o AS escolhido como destino das rotas. Foram feitas 16.352 rodadas de simulação, uma para cada AS da topologia, e foi calculada a média de todas as rodadas. Durante a reconstrução de cada rota foi medido o número máximo de ASes que devem ser testados em cada passo do procedimento de reconstrução. A razão principal para a escolha do número máximo de AS testados como estimativa para N_i é que a maioria das rotas passam pelos ASes com maior número de vizinhos. De fato, as estatísticas da Internet em nível de ASes mostram que 48% dos caminhos passam pelos 10 ASes com mais de 500 vizinhos [33]. Os resultados são exibidos na Tabela 3.1. A explicação detalhada dos resultados obtidos é feita no Capítulo 4. No momento, o que importam são os valores obtidos como estimativa de N_i.

Em uma abordagem de rastreamento sem estado e por um único pacote, toda a informação de rastreamento deve estar contida no cabeçalho de cada pacote. Assim, de acordo com os resultados mostrados na Tabela 3.1 e a Equação 3.1, a sobrecarga de cabeçalho necessária para a solução ótima é $m = 35$ bits. É importante observar que essa quantidade excede o espaço máximo disponível no cabeçalho IPv4, que é 25 bits, como mostrado por Dean *et al.* [13].

3.1.2 Permitindo uma Taxa de Erro Limitada para Reduzir a Sobrecarga de Cabeçalho

A definição da solução ótima é agora relaxada, passando a permitir uma taxa de erro limitada, D, de forma a tornar possível o projeto de uma solução que use somente os 25 bits disponíveis. Assim, passam a ser admitidos até $D.N_i$ falsos positivos em cada passo do procedimento de reconstrução de rota. Nesse caso, o

valor da sobrecarga de cabeçalho pode ser calculado usando a função de distorção de taxa, que fornece o número mínimo de bits $R(N_i, D)$ necessários para representar N_i ASes, aceitando uma probabilidade de erro máxima igual a D. Assim, a Equação 3.1 se torna

$$\begin{aligned} m' &= \sum_{i=1}^{n} \lceil R(N_i, D) \rceil \\ &\geq \sum_{i=1}^{n} \left\lceil \log_2(N_i) - D.\log_2 D - (1-D).\log_2(1-D) - D.\log_2(N_i - 1) \right\rceil, \end{aligned} \tag{3.2}$$

onde a desigualdade advém da Equação A.12, demonstrada no Apêndice A.

Substituindo os valores da Tabela 3.1 na Equação 3.2, obtém-se o número mínimo de bits m' necessários para representar todas as rotas em nível de ASes em função da distorção D, a taxa de erro máxima permitida durante a reconstrução de rota, conforme pode ser observado na Figura 3.1. Nota-se que a sobrecarga de cabeçalho

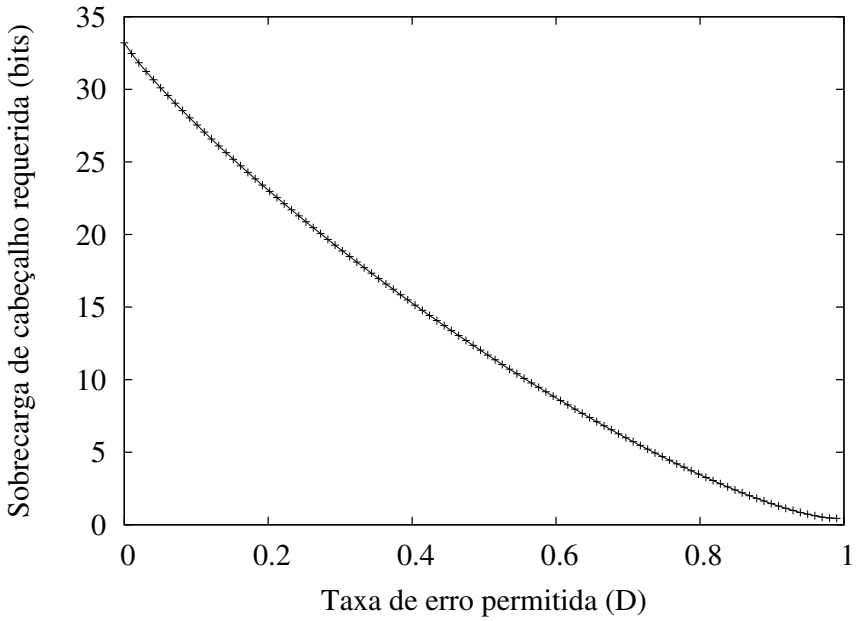


Figura 3.1: Sobrecarga de cabeçalho (m') em função da taxa de erro permitida (D).

pode ser reduzida ao custo de uma maior taxa de erro. A curva da figura mostra que, se uma taxa de erro de 15% for aceita, é possível reconstruir qualquer rota de ataque usando somente 25 bits. Porém, esta taxa de erro é bastante elevada e mesmo um esquema ótimo teria uma baixa acurácia. De fato, mostra-se no Capítulo 5 que, considerando uma taxa de erro de 15%, o procedimento de reconstrução de rota obtém cerca de 800 falsos positivos por atacante. Esse resultado mostra que a alta

taxa de falsos positivos não é uma limitação das soluções de rastreamento propostas, mas é uma característica intrínseca ao problema de representar a informação completa da rota de ataque usando um espaço de cabeçalho insuficiente. Felizmente, mostra-se no próximo capítulo que, considerando o objetivo último de localizar o atacante, não é necessário determinar todos os ASes que pertencem à rota de ataque. Logo, é possível encontrar o atacante de forma acurada usando somente os 25 bits disponíveis.

Capítulo 4

O Sistema Proposto

Nesse capítulo é descrita a proposta dessa dissertação. Antes disso, porém, são apresentadas algumas ideias básicas, que são oriundas da análise do problema da reconstrução de rota em nível de sistemas autônomos (ASes).

4.1 Reconstrução de Rota em Nível de ASes

A análise realizada nos dados e do procedimento de reconstrução de rota em nível de ASes resultou em duas principais descobertas que são apresentadas a seguir.

- **Descoberta 1: A hierarquia cliente-provedor pode ser explorada para localizar de forma acurada o atacante.**

A Figura 4.2 ilustra as particularidades do problema da reconstrução de rota em nível de ASes. Nota-se que, dependendo da distância à vítima, o número de ASes que devem ser testados durante o procedimento de reconstrução de rota varia significativamente. A razão para tal comportamento é que a distribuição do número de vizinhos dos ASes segue uma lei de potência [33]. Assim, a maioria dos ASes possui poucos vizinhos e está localizada na borda da rede, enquanto que alguns poucos ASes possuem um elevado número de vizinhos e estão localizados no núcleo da rede. Logo, quando o procedimento de reconstrução de rota é iniciado pela vítima, há poucos vizinhos a serem testados, visto que se está ainda na borda da rede. Quando se aumenta a distância à vítima, avançando em direção ao núcleo da rede, o número de vizinhos testados aumenta. Da mesma forma, caso se continue a aumentar a

distância à vítima, a borda da rede é atingida novamente e, consequentemente, o número de ASes testados diminui. Esse comportamento é um reflexo da hierarquia cliente-provedor que existe na topologia da Internet em nível de ASes. A relação cliente-provedor é fruto de relações comerciais entre os ASes: um AS-cliente paga ao seu AS-provedor para que este transporte tráfego de/para o AS-cliente. Além disso, os ASes-provedores trocam tráfego entre si para manter a conectividade global da Internet. Em geral, um mesmo AS-provedor possui diversos ASes-clientes, formando o que se chama de hierarquia cliente-provedor. Tal hierarquia é claramente vista nas características dos caminhos em nível de ASes: 62% dos caminhos possuem comprimento de 3 saltos [33]. No padrão de caminho mais comum, mostrado na Figura 4.1, o pacote é originado em um AS-cliente, na borda da rede, sobe para o provedor do AS de origem, entrando no núcleo da rede, depois vai para outro AS-provedor e finalmente chega ao seu destino, geralmente um AS-cliente localizado na borda da rede.

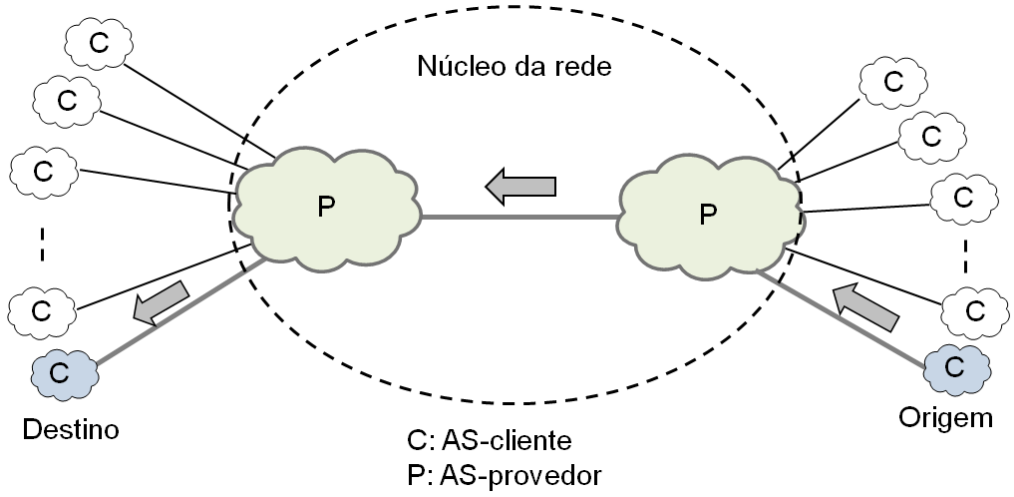


Figura 4.1: A hierarquia cliente-provedor e o padrão de caminho mais comum em nível de ASes.

Como mostrado no Capítulo 3, uma sobrecarga de cabeçalho de 35 bits, que não podem ser alocados no cabeçalho IPv4, é necessária para reconstruir qualquer caminho de ataque sem falsos positivos. Felizmente, foi observado que, usando somente os 25 bits disponíveis no cabeçalho IPv4, o procedimento de reconstrução de rota pode localizar o AS de origem com alta acurácia, desde que seja empregado um esquema de marcação adequado. O esquema de marcação proposto explora a

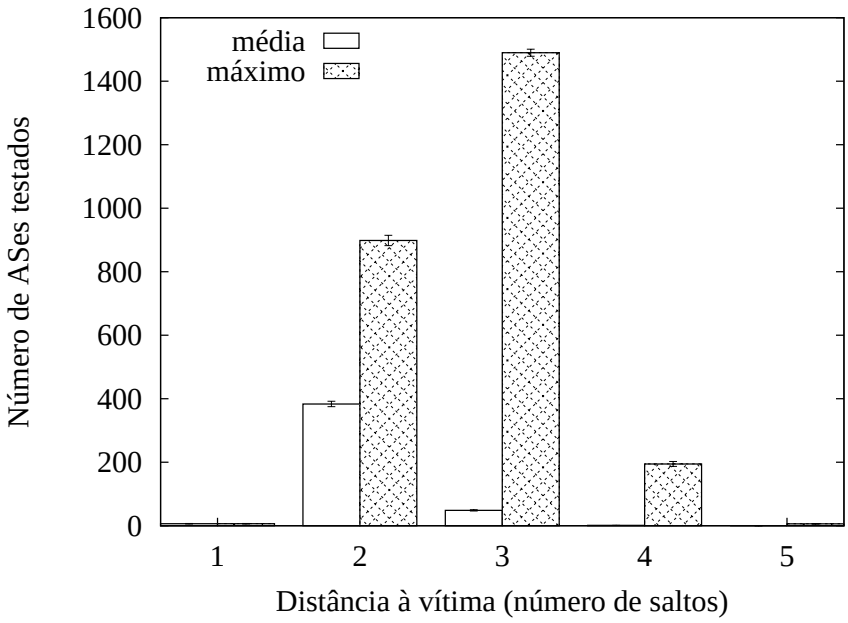


Figura 4.2: Número de ASes que devem ser testados em cada passo do procedimento de reconstrução de rota.

hierarquia cliente-provedor a fim de fornecer à vítima as informações de rota mais importantes. De acordo com as Figuras 4.1 e 4.2, os passos críticos durante a reconstrução de rota ocorrem no segundo e terceiro saltos, nos quais devem ser testados até 898 e 1490 ASes, respectivamente. Há muitos ASes a serem testados devido à concentração de caminhos que passam pelos grandes provedores, chamados de ASes de núcleo. Vale repetir que aproximadamente metade dos caminhos da Internet passam pelos 10 ASes com mais de 500 vizinhos [33]. Assim, partindo da vítima em direção ao atacante, o procedimento de reconstrução consegue identificar o provedor da vítima, entrando no núcleo da rede sem muitas dificuldades. Porém, para identificar o primeiro sistema autônomo (AS) de núcleo atravessado pelo pacote de ataque, é preciso testar quase mil ASes, o que pode resultar em uma elevada taxa de falsos positivos. Por essa razão, o primeiro passo crítico é a identificação do primeiro AS de núcleo da rota de ataque. O segundo passo crítico ocorre quando o procedimento de reconstrução alcança o provedor do AS-atacante e está tentando localizar o AS-atacante. Este é também um passo crítico, porque o atacante deve ser identificado dentre aproximadamente mil ASes.

Observando a existência de dois passos críticos durante a reconstrução de rota, propõe-se um novo esquema de marcação de pacotes determinístico no qual somente

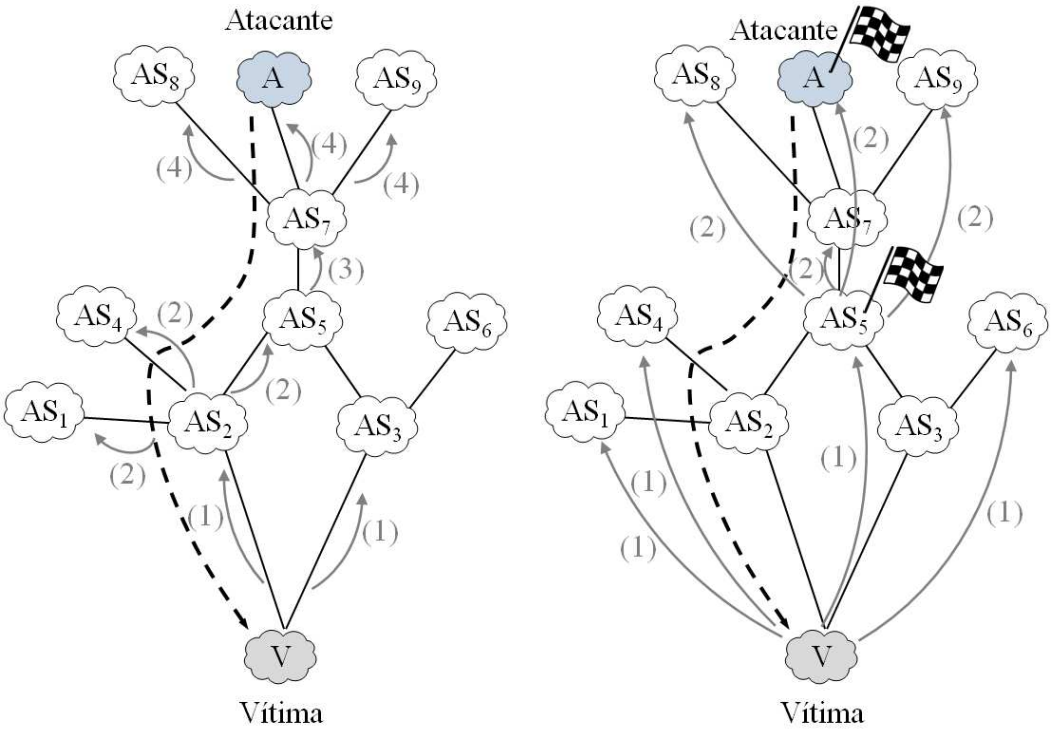
dois ASes marcam o pacote. Essas duas marcações são usadas como pontos de verificação (*checkpoints*) para guiar o procedimento de reconstrução de rota. Os pontos de verificação são estrategicamente posicionados no AS-atacante e no primeiro AS de núcleo da rota de ataque. Assim, em vez de representar a rota de ataque completa, propõe-se marcar apenas alguns ASes e, além disto, alocam-se mais bits do cabeçalho IP para os passos críticos, garantindo que os pontos de verificação sejam alcançados com uma baixa taxa de falsos positivos.

- **Descoberta 2: O atacante pode ser sempre encontrado, mesmo que alguns ASes não marquem o pacote.**

Observou-se também que o sistema autônomo (AS) de origem pode ser identificado ainda que não seja possível determinar todos os ASes percorridos pelo pacote de ataque. É possível pular alguns passos do procedimento de reconstrução de rota e ainda assim encontrar o atacante. Por exemplo, ao invés de testar os ASes que estão a um salto de distância da vítima, pode-se passar diretamente para o teste dos ASes do segundo salto. Neste caso, naturalmente, é necessário realizar mais testes. No procedimento clássico de reconstrução de rota salto-a-salto, testam-se os vizinhos da vítima e identifica-se um ou mais ASes. No salto seguinte, somente os vizinhos dos ASes identificados anteriormente são testados. Por outro lado, no procedimento proposto, o primeiro salto é pulado e *todos* os ASes localizados a d (no exemplo, $d = 2$) saltos da vítima são testados. Assume-se que o valor de d, a distância da vítima até o último AS que marcou o pacote, é conhecido pela vítima. Na Seção 4.2 é explicado como a vítima obtém essa informação no sistema proposto. A Figura 4.3 mostra a reconstrução da rota de ataque (A, AS_7, AS_5, AS_2, V), ilustrando a diferença entre o procedimento clássico e o proposto. No procedimento clássico de reconstrução salto-a-salto, mostrado na Figura 4.3(a), a vítima testa em primeiro lugar os ASes que estão a um salto de distância (AS_2 e AS_3). Considerando que o AS_3 não é um falso positivo, então somente o AS_2 é reconhecido. No próximo passo do procedimento, os ASes do segundo salto, AS_1, AS_4 e AS_5, são testados. Somente o AS_5 é identificado. A seguir, os ASes do terceiro salto são testados. No exemplo, o AS_7 é testado[1] e em seguida integrado ao grafo de reconstrução. Final-

[1]Apesar de o AS_3 ser vizinho do AS_5, o AS_3 não é testado, pois a sua distância à vítima (1 salto) é menor ou igual à distância do AS_5. Essa é uma consequência do uso do algoritmo de busca

mente, os ASes do quarto salto, AS_8, A e AS_9, são testados e o atacante A é então encontrado. Em comparação, o procedimento proposto (Figura 4.3(b)) pula alguns passos do procedimento clássico e realiza testes somente em pontos estratégicos, onde estão posicionados os pontos de verificação. Assim, o procedimento proposto pula o primeiro salto e testa diretamente os ASes que estão a dois saltos da vítima (AS_1, AS_4, AS_5 e AS_6). Nesse momento, o ponto de verificação, que nada mais é do que a marcação contida no pacote recebido pela vítima, é usado para identificar o AS correto, AS_5, com alta acurácia. A seguir, todos os ASes ascendentes do AS_5 (AS_7, AS_8, A e AS_9) são testados. Se a vítima soubesse que o próximo ponto de verificação está localizado a dois saltos do AS_5, ela poderia pular o teste do AS_7, mas é assumido que não se tem essa informação, pois, por opção de projeto, essa informação não é enviada à vítima no sistema proposto. Após o teste do segundo ponto de verificação, o atacante é finalmente encontrado.



(a) Procedimento clássico.

(b) Procedimento proposto.

Figura 4.3: Reconstrução da rota de ataque (A, AS_7, AS_5, AS_2, V).

em largura para a reconstrução de rota.

O procedimento proposto tem como vantagem a redução da sobrecarga de cabeçalho, pois menos ASes são representados no espaço de marcação, ao custo da realização de mais testes. De acordo com a análise apresentada na Seção 3.1.1, a sobrecarga de cabeçalho requerida pelo procedimento clássico para representar a rota de ataque (A, AS_7, AS_5, AS_2, V) é $1 + 2 + 1 + 2 = 6$ bits, enquanto que o procedimento proposto necessita de apenas $2 + 2 = 4$ bits. Nota-se que, nesse exemplo específico, ambos os procedimentos encontram o atacante sem erros, só que o procedimento proposto usa menos bits, pois nem todos os ASes da rota completa de ataque são representados. Por outro lado, em um caso geral, o aumento do número de testes realizados poderia provocar uma alta taxa de falsos positivos, mas na realidade isso não ocorre. A razão é a existência dos pontos de verificação, que evitam a divergência do procedimento de reconstrução nos pontos críticos, garantindo uma baixa taxa de falsos positivos, conforme verificado nos resultados de simulação (Capítulo 5). Ao pular alguns passos do procedimento de reconstrução de rota, a sobrecarga de cabeçalho foi reduzida, mas perdeu-se a capacidade de reconstruir a rota completa de ataque. Não obstante, o atacante pode ainda ser encontrado. Garantindo-se que o procedimento de marcação atribui um código para o atacante, este é certamente reconhecido durante a reconstrução de rota. Portanto, o atacante é sempre encontrado.

4.2 Detalhamento do Sistema Proposto

4.2.1 Definições e Hipóteses

Denomina-se um sistema autônomo (AS) de núcleo qualquer AS que pertence ao núcleo da topologia em nível de ASes, isto é, todos os ASes com nível 0, 1 e 2 mostrados na Tabela 4.1 (adaptada de [34]). No sistema proposto, somente dois ASes estrategicamente escolhidos participam da marcação de pacotes: o provedor do AS-atacante e o primeiro AS de núcleo atravessado pelo pacote de ataque. Propõe-se o uso do protocolo de roteamento inter-domínio BGP (*Border Gateway Protocol*) como o veículo de distribuição da informação de implantação do sistema. Os ASes cooperativos, isto é, os ASes que têm o sistema implantado, anunciam o suporte ao rastreamento em um atributo do BGP nos anúncios de rota. Assim, cada AS

sabe quais ASes são cooperativos e qual a distância entre cada um. Não é exigido que os ASes-clientes sejam cooperativos. Para garantir que o atacante é sempre encontrado, assume-se que os ASes não-clientes são cooperativos. Isto representa somente 18,5% dos ASes da Internet, de acordo com Subramanian *et al.* [34]. Assim, se um AS-cliente não marcar o pacote, o seu provedor o fará em seu lugar. Com isso, além da sua própria marcação, o provedor do atacante irá necessariamente inserir a marcação do atacante no pacote.

Tabela 4.1: Distribuição dos ASes na hierarquia.

Nível	**Porcentagem em relação ao total de ASes**
(0) Núcleo denso	0,2%
(1) Núcleo de trânsito	1,2%
(2) Núcleo externo	8,2%
(3) Pequenos ISPs regionais	8,9%
(4) Clientes	81,5%

4.2.2 Determinando o Primeiro Roteador a Marcar o Pacote

No sistema proposto a marcação é feita pelos roteadores de borda de cada AS logo após o recebimento do pacote pela interface de ingresso. Assim, um roteador de borda deve saber se ele é ou não o primeiro roteador a marcar o pacote. Para isso, cada roteador de borda de um AS cooperativo deve executar o Algoritmo 1, que verifica o outro lado de cada interface de ingresso e retorna "sim", caso o roteador em questão seja o primeiro roteador a marcar os pacotes que chegam pela interface de ingresso testada; e "não", caso contrário. Se a interface de ingresso conecta o roteador a uma subrede do mesmo AS, então o roteador em questão é o primeiro roteador a marcar os pacotes que chegam por essa interface. Caso contrário, a interface de ingresso conecta o AS do roteador a outro AS. Nesse caso, a informação de implantação distribuída pelo BGP é usada para verificar se o AS vizinho não é cooperativo. Em caso positivo, então o vizinho é certamente um AS-cliente, visto que é assumido que todos os ASes não-clientes são cooperativos. Assim, uma vez que o AS vizinho não participa da marcação, o roteador em questão é o primeiro roteador a marcar o pacote. Por outro lado, se o AS vizinho é cooperativo, então

o roteador não pertence ao primeiro AS cooperativo dos caminhos que chegam por essa interface de ingresso. Ressalta-se que esse procedimento é raramente executado, visto que a conectividade inter-domínio não sofre modificações frequentes.

Algoritmo 1 Procedimento para Determinar o Primeiro AS a Marcar o Pacote.

```
Seja R um roteador de borda de um AS cooperativo;
para cada interface de ingresso faça
  se (outro lado da interface conecta R a uma subrede do mesmo AS) então
    se (o roteador vizinho é também um roteador de borda) então
      retornar "não";
    senão
      retornar "sim";
    fim do se
  senão
    se (AS vizinho não é cooperativo) então
      retornar "sim";
    senão
      retornar "não";
    fim do se
  fim do se
fim do para
```

Uma vez que se tem um procedimento para determinar o primeiro roteador a marcar o pacote, então uma solução simples para identificar a origem do pacote seria inserir o número de AS (ASN) do AS de origem no pacote. Porém, o tamanho atual dos ASNs é de 32 bits [35], que não cabem dentro do espaço disponível para marcação no cabeçalho IPv4.

4.2.3 Procedimento de Marcação de Pacotes

Para armazenar a informação de rastreamento, propõe-se sobrecarregar 25 bits do cabeçalho IPv4, como mostrado na Figura 4.4. Conforme argumentado por Dean *et al.* [13], o campo de tipo de serviço (*Type of Service* - TOS) foi projetado para permitir o tratamento especial de tráfego, mas a atribuição de valores arbitrários

a esse campo não produz nenhuma diferença na entrega de pacotes mensurável na prática. Sobrecarregar o bit de fragmento reservado também não causa nenhum efeito nas implementações atuais. Finalmente, embora o campo de identificação de fragmento seja sobrecarregado, no sistema proposto é mantida a compatibilidade com a fragmentação, pois a marcação é a mesma para todos os fragmentos de um pacote. Assim, os fragmentos podem ser reagrupados no destino, visto que todos possuem o mesmo valor no campo de identificação de fragmento [25]. Os campos sobrecarregados do cabeçalho IP foram divididos em três campos. Os campos `Hash 1` e `Hash 2`, de 11 e 12 bits, respectivamente, são usados para acomodar as marcações dos dois ASes que marcam o pacote. É usado também um campo `Controle` de 2 bits.

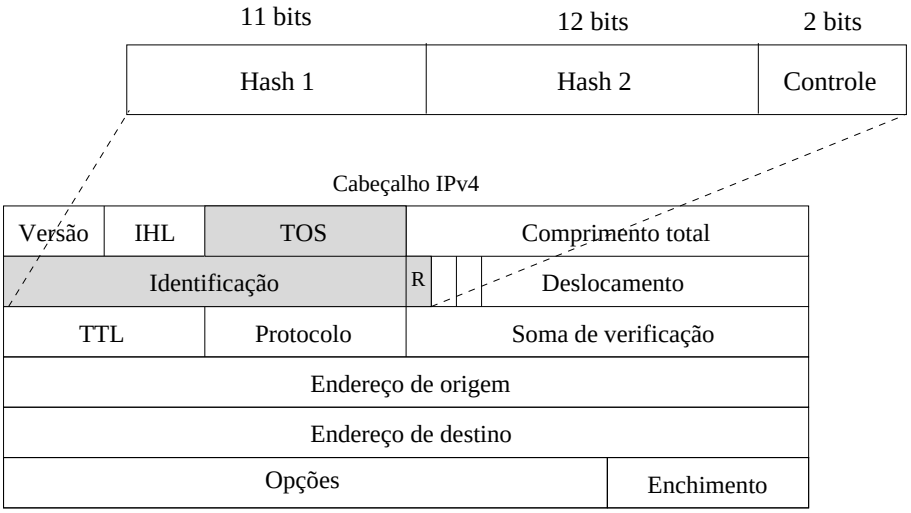


Figura 4.4: Campos sobrecarregados do cabeçalho IPv4 e o seu novo uso no esquema proposto.

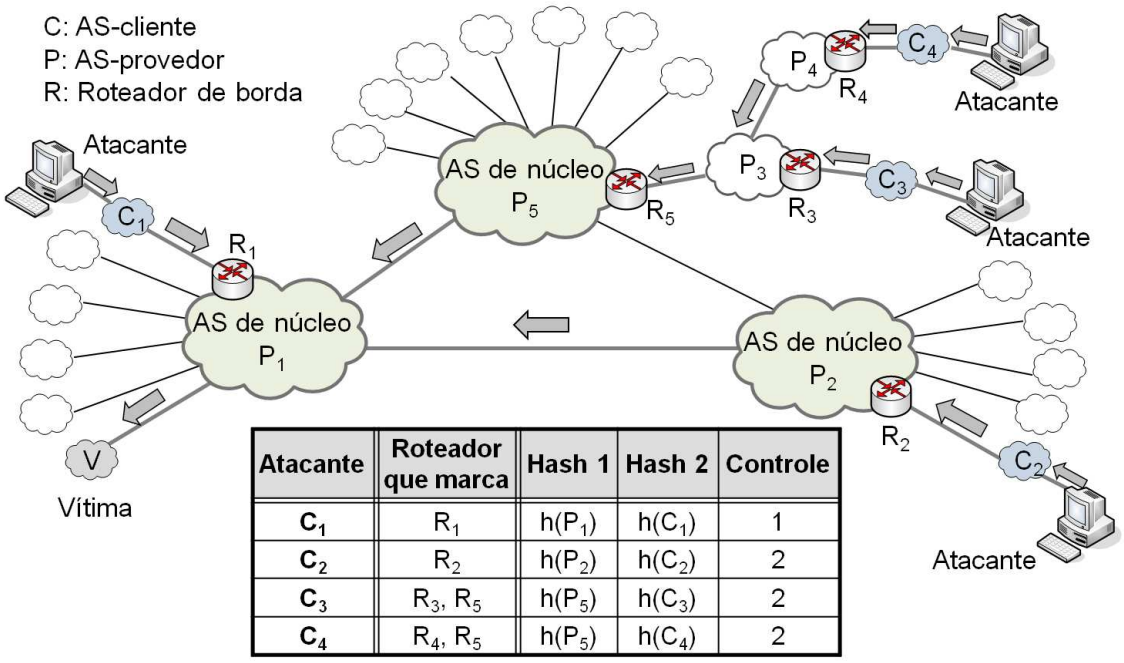
O procedimento de marcação de pacotes proposto é mostrado no Algoritmo 2. O algoritmo é basicamente a inserção do *hash* de um ASN no campo de marcação apropriado. O campo `Controle` é usado para indicar se algum AS de núcleo já marcou o pacote e também para carregar a distância entre o primeiro AS de núcleo da rota de ataque e a vítima. Para apagar a condição inicial do pacote, o primeiro AS cooperativo da rota de ataque coloca o campo `Controle` em zero. O primeiro AS cooperativo pode ser o AS de origem, se este implementa o sistema de rastreamento, ou o seu provedor, que também conhece o ASN do AS de origem. Assim, o primeiro AS cooperativo insere o *hash* de $ASN_{AS\,origem}$ no campo `Hash 2`. Esta inserção é

feita para garantir que a marcação do AS-atacante será inserida, visto que o AS-atacante não precisa participar da marcação. A segunda marcação é feita pelo primeiro AS de núcleo atravessado pelo pacote de ataque. Ao receber um pacote, qualquer AS de núcleo deve analisar o valor do campo `Controle` para verificar se o pacote já foi marcado anteriormente. Se o valor do campo `Controle` for diferente de zero, então algum AS de núcleo já marcou o pacote. Nesse caso, já há duas marcações no pacote e, então, não há nada a ser feito. Por outro lado, se o valor for igual a zero, então o AS atual deve inserir o *hash* do seu ASN no campo `Hash 1` e atualizar o campo `Controle` com a sua distância até a vítima. A distância é calculada contando o número de elementos do atributo do BGP chamado `AS_PATH` [28], que fornece a lista de ASes que precisam ser atravessados para chegar a um dado destino.

Algoritmo 2 Procedimento de Marcação de Pacotes.

Seja h uma função *hash*;
para para cada pacote p **faça**
  se (AS atual = primeiro AS cooperativo) **então**
    $p.hash_2 \leftarrow h(ASN_{AS\,origem})$;
    $p.controle \leftarrow 0$;
  fim do se
  se (AS atual = AS de núcleo) **então**
    se ($p.controle = 0$) **então**
      $p.hash_1 \leftarrow h(ASN_{AS\,atual})$;
      $p.controle \leftarrow$ distância até a vítima;
    fim do se
  fim do se
fim do para

A sobrecarga de processamento introduzida pelo esquema de marcação proposto é bem pequena, porque as marcações podem ser calculadas em avanço. Assim, ao receber o pacote, o roteador de borda do AS que deve marcar o pacote só precisa comparar o conteúdo do campo `Controle` com o valor 0 e inserir valores pré-calculados no pacote, sendo que nenhum cálculo de funções *hash* precisa ser feito a cada pacote recebido.



Atacante	Roteador que marca	Hash 1	Hash 2	Controle
C_1	R_1	$h(P_1)$	$h(C_1)$	1
C_2	R_2	$h(P_2)$	$h(C_2)$	2
C_3	R_3, R_5	$h(P_5)$	$h(C_3)$	2
C_4	R_4, R_5	$h(P_5)$	$h(C_4)$	2

Figura 4.5: Exemplo das marcações recebidas pela vítima após os ataques originados nos ASes-clientes C_1, C_2, C_3, e C_4.

A Figura 4.5 mostra como o procedimento de marcação funciona na prática. A figura apresenta quatro exemplos de ataque, com diferentes tamanhos de rota. Em todos os exemplos, foi considerado que os ASes-clientes não são cooperativos. A rota de ataque mais curta é (C_1, P_1, V). Neste exemplo, o pacote de ataque é originado no AS-cliente C_1, sobe para o AS-provedor P_1, que também é um AS de núcleo, e então alcança a vítima V. De acordo com o Algoritmo 2, o roteador de borda do primeiro AS cooperativo deve inserir o *hash* de $ASN_{AS\,origem}$ no campo `Hash 2`. Dessa forma, o roteador R_1 insere $h(C_1)$ no pacote. A segunda marcação é feita pelo primeiro AS de núcleo atravessado pelo pacote, que também é P_1. Por essa razão, R_1 insere $h(P_1)$ no campo `Hash 1`. R_1 também insere a distância de P_1 até a vítima V, que é 1 salto nesse caso, no campo `Controle`. Outro exemplo de rota de ataque é (C_2, P_2, P_1, V). Novamente, o primeiro AS cooperativo, P_2, é também o primeiro AS de núcleo da rota de ataque. Portanto, o roteador de borda de P_2, R_2, insere $h(P_2)$ e $h(C_2)$ nos campos `Hash 1` e `Hash 2`, respectivamente. O valor 2 é inserido no campo `Controle`, pois a distância entre P_2 e V é de 2 saltos. Após deixar P_2, o pacote chega a P_1, que também é um AS de núcleo. Assim, P_1 deve analisar o valor do campo `Controle`. P_1 então verifica que este valor é diferente de 0 e conclui

que algum AS de núcleo já marcou o pacote. Por isso, P_1 simplesmente encaminha o pacote até a vítima, sem fazer nenhuma modificação. O terceiro exemplo de rota de ataque é (C_3, P_3, P_5, P_1, V). Neste caso, o primeiro AS cooperativo é P_3. Logo, R_3 insere $h(C_3)$ no campo `Hash 2`. Seguindo o Algoritmo 2, R_3 também coloca o campo `Controle` em zero. Em seguida, o pacote chega ao primeiro AS de núcleo, P_5. Assim, o roteador de borda de P_5, R_5, marca o pacote com $h(P_5)$ e atualiza o campo `Controle` com a sua distância até V, que é 2 neste caso. O pacote é então encaminhado ao AS de núcleo P_1, que não faz nada, visto que o valor do campo `Controle` é diferente de 0. Finalmente, o pacote chega à vítima. O último exemplo é de uma rota de ataque de 5 saltos: $(C_4, P_4, P_3, P_5, P_1, V)$. Nesse caso, o primeiro AS cooperativo é P_4. Dessa forma, R_4 insere $h(C_4)$ no campo `Hash 2`. O pacote é então encaminhado para P_3, que não é o primeiro AS cooperativo nem o primeiro AS de núcleo desse caminho. Portanto, P_3 não precisa fazer nada. A seguir, o pacote chega ao primeiro AS de núcleo, que é P_5. Assim, o roteador de borda R_5 marca o pacote com $h(P_5)$ e atualiza o campo `Controle` com sua distância até V, que é 2. O pacote segue então para P_1, que simplesmente o encaminha para a vítima. Após o recebimento do pacote marcado, a vítima pode iniciar o procedimento de reconstrução de rota.

4.2.4 Procedimento de Reconstrução de Rota

O procedimento de reconstrução de rota é descrito pelo Algoritmo 3. A vítima inicia o procedimento verificando o campo `Controle` do pacote de ataque recebido e extraindo a distância d até o primeiro AS de núcleo da rota de ataque. Em seguida, a vítima tenta passar pelos dois passos críticos descritos na Seção 4.1. Para identificar o primeiro AS de núcleo da rota de ataque, a vítima verifica se o *hash* do ASN de cada AS que está localizado a exatamente d saltos da vítima confere com o valor do campo `Hash 1`. Não é necessário assumir o conhecimento do mapa da topologia, visto que a vítima pode inferir a informação de distância a partir dos anúncios BGP recebidos dos ASes vizinhos. Como o campo `Controle` de 2 bits só pode representar até 3 valores de distância, se $d = 3$, os ASes que estão a 4 ou mais saltos de distância da vítima são também testados. No segundo passo do procedimento, o *hash* de cada ascendente dos ASes identificados no primeiro passo são

Algoritmo 3 Procedimento de Reconstrução de Rota.

Seja h uma função $hash$;

Seja p um pacote de ataque recebido pela vítima;

Sejam S_1 e S_2 os conjuntos de ASes identificados no primeiro e segundo saltos;

$d \leftarrow p.controle$;

para cada AS i que está a d (ou mais, se $d = 3$) saltos da vítima **faça**

  se $(h(ASN_i) = p.hash_1)$ **então**

    $S_1 \leftarrow S_1 \cup \{i\}$;

  fim do se

fim do para

para cada AS $j \in S_1$ **faça**

  para cada ascendente a de j **faça**

    se $(h(ASN_a) = p.hash_2)$ **então**

      $S_2 \leftarrow S_2 \cup \{a\}$;

    fim do se

  fim do para

fim do para

retornar $S_1 \cup S_2$;

comparados com o valor contido no campo `Hash 2`. Dessa forma, tem-se a garantia de que o atacante é encontrado mesmo que ele esteja localizado a mais de 2 saltos de distância do primeiro AS de núcleo da rota de ataque. Por exemplo, considerando a rota $(C_4, P_4, P_3, P_5, P_1, V)$, primeiramente a vítima usa o campo `Hash 1` para localizar o AS de núcleo P_5. Após isso, no segundo passo, a vítima testa todos os ASes ascendentes de P_5, isto é, P_3, C_3, P_4, e C_4. Finalmente, o AS-atacante C_4 é encontrado. Portanto, como pelo menos o provedor do atacante certamente marcou o pacote, o procedimento de reconstrução de rota sempre encontra o atacante.

4.3 Discussão

Nessa seção são discutidos alguns aspectos importantes, comparando as características do sistema proposto com as outras propostas existentes na literatura.

4.3.1 Robustez contra a Interferência do Atacante

O esquema proposto é robusto contra dois tipos de interferência do atacante. O primeiro tipo de interferência, à qual estão vulneráveis os esquemas probabilísticos, ocorre quando o atacante ajusta a condição inicial do pacote a fim de gerar falsos positivos durante a reconstrução de rota [22]. No esquema proposto, ao saber que de fato pertence ao primeiro AS cooperativo da rota de ataque, o primeiro roteador a marcar o pacote é capaz de apagar a condição inicial do pacote, eliminando qualquer interferência da condição inicial do pacote. Portanto, o sistema proposto é mais robusto que os esquemas probabilísticos, nos quais uma fração dos pacotes recebidos pela vítima não é marcada por nenhum roteador (Seção 2.4) e, então, está sob controle do atacante. Outro problema de robustez ocorre quando o atacante envia uma pequena quantidade de pacotes de ataques para evitar ser rastreado. Os ataques maciçamente distribuídos agravam esse problema, porque tais ataques provocam efeitos devastadores ainda que cada estação de ataque gere apenas uma pequena quantidade de tráfego. Em um caso extremo, cada estação de ataque pode enviar um único pacote, que não poderá ser rastreado pela maioria dos sistemas de rastreamento. O esquema proposto, por sua vez, segue a abordagem de rastreamento por um único pacote e, dessa forma, é adequado para o combate a ataques

de MDDoS.

4.3.2 Requisitos Práticos

O sistema proposto atende a vários aspectos práticos que são importantes para uma implantação real:

1. não é excedido o espaço disponível para marcação no cabeçalho IP como em [11, 36];
2. não é armazenado estado na infraestrutura de rede como em [12, 37, 27];
3. tanto o procedimento de marcação de pacotes como o procedimento de reconstrução de rota possuem baixo custo de processamento e, portanto, não requerem recursos computacionais excessivos como em [9, 13, 14];
4. não é assumido o conhecimento do mapa da topologia como em [10, 29];
5. o atacante é identificado com alta acurácia, como mostrado no Capítulo 5, enquanto que a maioria dos sistemas propostos possui elevadas taxas de falsos positivos [25, 9, 10, 13, 28, 29];
6. é considerado um cenário de implantação parcial do sistema, dispensando-se a participação dos ASes-clientes, que representam 81,5% do total de ASes da Internet, enquanto que muitos sistemas propostos requerem uma implantação global para funcionar [9, 10, 29, 11, 25];
7. é mantida a compatibilidade com a fragmentação, ao passo que muitas propostas simplesmente ignoram esse aspecto prático [9, 28, 29].

4.3.3 Escalabilidade

O sistema proposto é escalável, porque toda informação necessária para rastrear a origem está contida em um único pacote. Essa premissa é a base de projeto do sistema proposto. A consequência de usar um único pacote para reconstruir a rota de ataque é que a taxa de falsos positivos do sistema proposto é independente do número de atacantes. Por outro lado, a maioria dos sistemas de marcação de

pacotes, devido à escassez de espaço no cabeçalho IP, divide a informação de rastreamento em múltiplos pacotes, o que faz com que a taxa de falsos positivos cresça com o número de atacantes, conforme mostrado na Seção 2.4. Para resolver esse problema de escalabilidade, Goodrich [38] propôs o uso da chamada informação de acoplamento (*linkage information*) para evitar que pacotes oriundos de caminhos distintos sejam incorretamente agrupados. A ideia é atribuir um identificador para cada caminho de forma a permitir que a vítima saiba qual caminho está associado a cada pacote de ataque recebido. Esse mecanismo de acoplamento pacote/caminho não é sempre perfeito e é, em última análise, responsável pelas altas taxas de falsos positivos devidas a erros no processo de agrupamento de pacotes referentes a cada rota de ataque. De fato, assumindo um elevado número de atacantes, são necessários muitos bits para garantir poucos erros de acoplamento pacote/caminho. Considerando que a informação de acoplamento possui b bits, o número médio de caminhos erroneamente acoplados a um dado pacote é $C/2^b$, onde C é o número de caminhos de ataque. Por exemplo, assumindo que cada atacante usa um caminho de ataque distinto e considerando um ataque com 1 milhão de atacantes, são necessários 20 bits de informação de acoplamento para restringir a 1 o número de erros de acoplamento. Nesse caso, praticamente todo o espaço disponível no cabeçalho IP estaria sendo gasto para armazenar a informação de acoplamento, restando pouquíssimos bits para armazenar a informação de rastreamento propriamente dita. Portanto, a melhor opção para reduzir a taxa de falsos positivos é evitar dividir a informação de rastreamento em múltiplos pacotes, isto é, a informação da rota de ataque completa deve estar contida em um único pacote.

4.3.4 Reconstrução de Rota vs. Recuperação de Endereço

O esquema proposto utiliza uma nova abordagem de marcação determinística na qual o atacante é identificado através de um procedimento de reconstrução de rota. Isso permite reduzir a sobrecarga de cabeçalho e, assim, rastrear o atacante a partir de um único pacote, tornando o sistema escalável para ataques de MD-DoS. Originalmente, a marcação determinística de pacotes usa um procedimento de recuperação do endereço do primeiro roteador da rota de ataque [25]. Assim, essa proposta precisa dividir a informação de rastreamento em múltiplos pacotes

para possibilitar a transferência dos 32 bits do endereço IP usando apenas 17 bits para marcação. Consequentemente, a proposta de Belenky e Ansari, apesar de ser determinística, é incapaz de rastrear a origem a partir de um único pacote.

Capítulo 5

Resultados Analíticos e de Simulação

5.1 Resultados Analíticos

5.1.1 Número Esperado de Falsos Positivos

O número de falsos positivos, P, depende do número de ASes que devem ser testados em cada passo do procedimento de reconstrução de rota. Sejam x_1 e x_2 o número de ASes que devem ser testados nos passos 1 e 2, respectivamente; e h_1 e h_2 o comprimento das saídas das funções *hash* dos passos 1 e 2, respectivamente. Assumindo que as funções *hash* são uniformes, então a probabilidade de o *hash* de um AS testado ser igual ao *hash* do primeiro AS de núcleo da rota de ataque é $1/2^{h_1}$. Assim, após x_1 testes, o número esperado de falsos positivos no primeiro passo do procedimento de reconstrução de rota é $x_1/2^{h_1}$ e o número de ASes integrados ao grafo de reconstrução é $(1 + x_1/2^{h_1})$, incluindo o primeiro AS de núcleo da rota de ataque. No segundo passo do procedimento de reconstrução de rota, são testados x_2 ascendentes de cada AS reconhecido no primeiro passo. Logo, o número esperado de falsos positivos, $E[P]$, em todo o procedimento de reconstrução de rota é dado por

$$E[P] = \frac{x_1}{2^{h_1}} + \left(1 + \frac{x_1}{2^{h_1}}\right) * \frac{x_2}{2^{h_2}}. \tag{5.1}$$

5.1.2 Escalabilidade em Relação ao Número de Atacantes

É importante observar que $E[P]$ na Equação 5.1 depende apenas do conjunto de parâmetros $\{x_1, x_2, h_1, h_2\}$. Por conseguinte, o número de falsos positivos não se altera com o aumento do número de atacantes, visto que $E[P]$ não depende do número de atacantes.

5.1.3 Escalabilidade em Relação ao Número de Nós da Topologia

Da Equação 5.1, tem-se que

$$x_1 < 2^{h_1}/2 \ e \ x_2 < 2^{h_2}/3 \ \Rightarrow \ E[P] < 1. \tag{5.2}$$

No sistema proposto $h_1 = 11$ e $h_2 = 12$, o que significa que o número de falsos positivos vai ser menor do que 1, se o número de ASes testados nos passos 1 e 2 for menor do que 1024 e 1366, respectivamente. Este resultado mostra que o sistema proposto é também escalável em relação ao número de nós (ASes) da topologia. Os dados da evolução temporal da topologia em nível de ASes mostram que o número de ASes-clientes aumenta mais rapidamente que o número de ASes de núcleo [39], isto é, o crescimento da topologia preserva a sua hierarquia. Por exemplo, suponha que cada AS de núcleo possui na média 1.000 ASes-clientes. Assim, se são adicionados 1.001 nós à topologia, o número de ASes de núcleo será incrementado de 1 somente. Nota-se também que o conjunto de ASes testados em cada passo do procedimento de reconstrução de rota é composto basicamente por ASes-clientes de um dado AS-provedor. No primeiro passo do procedimento, a maior parte dos ASes testados são clientes do primeiro AS de núcleo da rota de ataque. No segundo passo, a maioria dos ASes testados é composta por clientes do provedor do atacante. Logo, se o crescimento da topologia tiver pouco efeito sobre a razão de número de clientes por provedor, então x_1 e x_2 sofrerão um aumento bem mais lento do que o número geral de nós da topologia. Consequentemente, conclui-se da Equação 5.1 que o aumento do número de ASes causa apenas um ligeiro aumento do número de falsos positivos. É importante frisar que essa é uma propriedade única do sistema proposto, pois ele explora a hierarquia cliente-provedor, o que não é feito pelos demais sistemas da literatura. Essa propriedade é importante, porque os estudos mostram

um crescimento exponencial do número de ASes. É esperado que se atinja o número de 100 mil ASNs alocados em 2015 [35].

5.2 Resultados de Simulação

5.2.1 O Cenário de Simulação

Foi desenvolvido um simulador próprio para analisar o desempenho do sistema proposto. Foi usada uma topologia real da Internet em nível de ASes, obtida em julho de 2009 a partir de dados de medição do projeto Ark (*Archipelago*) [32], composta por 16.352 ASes e 39.346 enlaces. O ataque de negação de serviço maciçamente distribuído, a marcação de pacotes e a reconstrução de rota foram simulados da seguinte maneira. Em primeiro lugar, escolhe-se aleatoriamente a vítima a partir do conjunto de nós da topologia. Em seguida, são definidas as rotas de ataque, que são caminhos sem ciclos que terminam na vítima escolhida. A transmissão dos pacotes de ataque é simulada inserindo-se as marcações apropriadas nos campos de marcação de acordo com os ASes que compõem cada rota de ataque. Uma vez que os pacotes são marcados, o procedimento de reconstrução é iniciado a partir da vítima. Os resultados de simulação apresentados representam a média de 2 mil rodadas de simulação. Para cada resultado medido, foi calculado um intervalo de confiança de 95%, representado nos gráficos por barras de erro verticais.

5.2.2 Sistemas Comparados via Simulação

O sistema proposto é comparado com outros sistemas de rastreamento da literatura. Foram implementados no simulador os seguintes sistemas: o sistema interdomínio FAST [29] e o sistema probabilístico de Song e Perrig [10]. Ao contrário do sistema proposto, esses sistemas não seguem a abordagem de rastreamento por um único pacote. Dessa forma, os ataques simulados foram compostos por um número suficiente de pacotes para cada sistema funcionar adequadamente, isto é, 10 pacotes para o FAST e 1.000 para o esquema probabilístico. Com isso, a vítima recebe todas as possíveis marcações de cada roteador das rotas de ataque. Isso representa o melhor cenário possível para os dois sistemas comparados. O sistema FAST foi

escolhido para comparação porque ele tem algumas características semelhantes ao sistema proposto. Assim, o objetivo foi avaliar o impacto das características que são únicas do sistema proposto. Em primeiro lugar, o FAST é também um sistema determinístico que usa funções *hash* para marcar os pacotes. A diferença é que o FAST usa 5 marcações, ao invés de duas, o que implica em um menor comprimento da saída das funções *hash* (4 bits contra 11 e 12 bits do sistema proposto). Além disso, o FAST é também um sistema inter-domínio, mas que não explora a hierarquia cliente-provedor. Dessa forma, os resultados mostram os benefícios dessa característica do sistema proposto. Uma diferença importante é que o FAST usa 4 funções *hash* diferentes para cada marcação, necessitando de múltiplos pacotes para poder receber todas as marcações de cada AS. Na implementação da reconstrução de rota do FAST no simulador, um AS testado é integrado ao grafo de reconstrução somente se todos os resultados da aplicação das 4 funções *hash* ao ASN do AS testado conferirem com a informação de rastreamento recebida pela vítima. Isso representa o melhor caso para o sistema FAST em termos de número de falsos positivos obtidos. O sistema de Song e Perrig foi implementado no simulador a fim de avaliar o desempenho de um esquema probabilístico. O sistema de Song e Perrig usa funções *hash* com saídas de 8 bits para codificar a informação de enlace. Por ser baseado na abordagem de marcação probabilística, esse sistema requer muito mais pacotes recebidos do que os outros sistemas comparados. Assim, para uma comparação justa, foi considerado um limiar de 5 de 8 *hashes* iguais à marcação recebida pelo vítima para integrar um AS testado ao grafo de reconstrução. Esse limiar corresponde aos 1.000 pacotes recebidos. Um aumento do limiar reduziria o número de falsos positivos desse sistema, mas implicaria também um número ainda maior de pacotes recebidos. A fim de comparar também com um sistema determinístico, é apresentada no gráfico a expressão analítica do número esperado de falsos positivos do sistema de Belenky e Ansari [25], expressa pela Equação 2.3. Considerou-se que o endereço do primeiro roteador foi dividido em $k = 4$ segmentos de $a = 8$ bits e que foram usadas $f = 4$ funções *hash* de $d = 5$ bits cada.

5.2.3 Comparação em Termos de Acurácia e Escalabilidade

A diferença fundamental entre o sistema proposto e os sistemas comparados é que esses outros sistemas dependem do recebimento de múltiplos pacotes para reconstruir a rota de ataque e, portanto, não são escaláveis para ataques de MDDoS, como explicado na Seção 4.3.3. A fim de provar essa afirmação e mostrar que o sistema proposto é escalável para ataques de MDDoS, foi simulado um cenário com 1.000 estações de ataque por AS e foi medida a acurácia em função do número de atacantes. A acurácia é medida em termos do número de falsos positivos. Um sistema de rastreamento ideal encontra o atacante sem nenhum falso positivo durante a reconstrução de rota. Assim, quanto menor o número de falsos positivos, melhor a acurácia com que o verdadeiro atacante é identificado. Os resultados de simulação da acurácia em função do número de atacantes são mostrados na Figura 5.1.

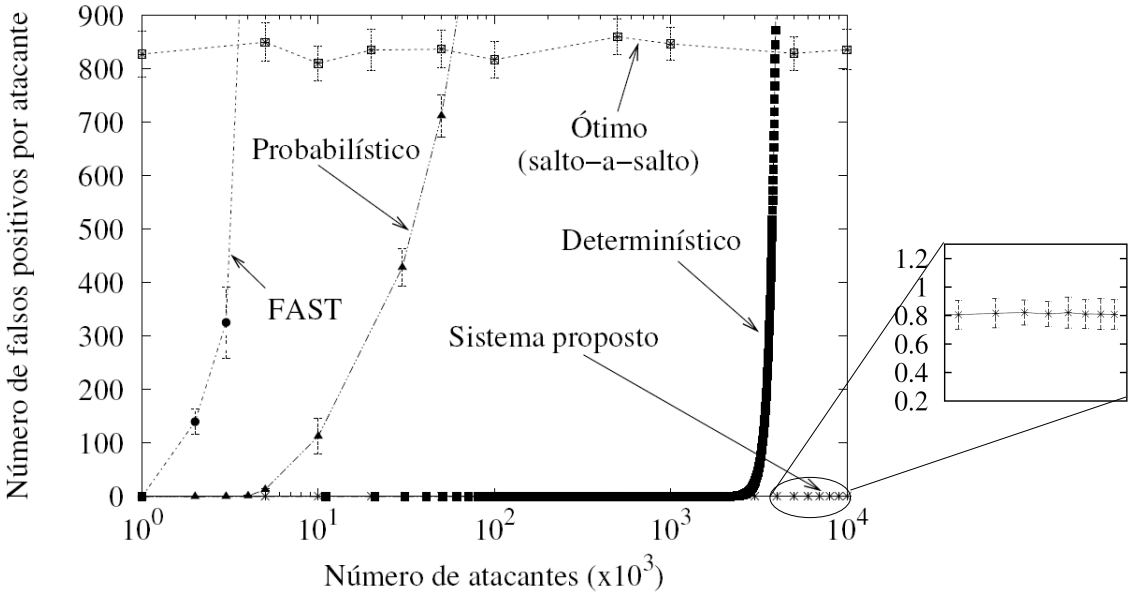


Figura 5.1: Acurácia dos sistemas avaliados, medida em número de falsos positivos por atacante, em função do número de atacantes.

Os resultados confirmam que a acurácia do sistema proposto é constante em relação ao número de atacantes. Além disso, observa-se que o número de falsos positivos é de fato bem pequeno. A figura mostra que o sistema proposto é capaz de rastrear o AS de origem de 10 milhões de atacantes com apenas 0,8 falsos positivos por atacante. Em comparação, o número de falsos positivos dos sistemas FAST, determinístico e probabilístico cresce exponencialmente com o número de

atacantes. Isso se deve principalmente a erros de reconstrução de rota causados pela combinação incorreta da informação de rastreamento proveniente de rotas de ataque distintas. Finalmente, o sistema salto-a-salto ótimo para rastreamento por um único pacote apresentado no Capítulo 3 também foi simulado para observar as diferenças entre o procedimento de reconstrução de rota salto-a-salto e o procedimento proposto, baseado em pontos de verificação, considerando o mesmo cenário e a mesma sobrecarga de cabeçalho (25 bits). A Figura 5.1 mostra que a acurácia do sistema ótimo é também constante, o que confirma que este esquema é também escalável em relação ao número de atacantes. No entanto, a acurácia do sistema salto-a-salto ótimo é cerca de 1.000 vezes pior do que a acurácia do sistema proposto. A razão fundamental é que o sistema ótimo, que segue a abordagem clássica de reconstrução salto-a-salto, tenta representar a rota completa de ataque usando um espaço insuficiente (25 bits), resultando em uma alta taxa de falsos positivos durante a reconstrução de rota. O esquema proposto é mais eficiente por representar somente a informação que realmente importa para localizar o atacante, pois os pontos de verificação foram estrategicamente posicionados nos pontos críticos do procedimento de reconstrução de rota. Portanto, pode-se concluir que o uso de pontos de verificação reduz a informação de rastreamento que precisa ser armazenada no cabeçalho IP e, ao mesmo tempo, mantém uma elevada acurácia.

5.2.4 Avaliação do Efeito do Tamanho da Rota de Ataque

Foi também avaliado o efeito do tamanho da rota de ataque na acurácia do sistema proposto, conforme mostrado na Figura 5.2. Observa-se que o número de falsos positivos cresce com o aumento do tamanho da rota de ataque. Isto ocorre porque, para uma rota maior, mais ASes devem ser testados pelo procedimento de reconstrução de rota, especialmente no segundo passo do algoritmo, no qual são testados todos os ASes ascendentes do AS identificado no primeiro passo. No entanto, mesmo o valor máximo obtido, 1,8 falsos positivos por atacante, é considerado baixo quando comparado ao número de falsos positivos obtidos pelos outros sistemas de rastreamento. Além disso, esse valor máximo não deve aumentar com o crescimento da topologia em nível de ASes, visto que não é esperado um aumento significativo dos tamanhos das rotas em nível de ASes com o passar do tempo [8].

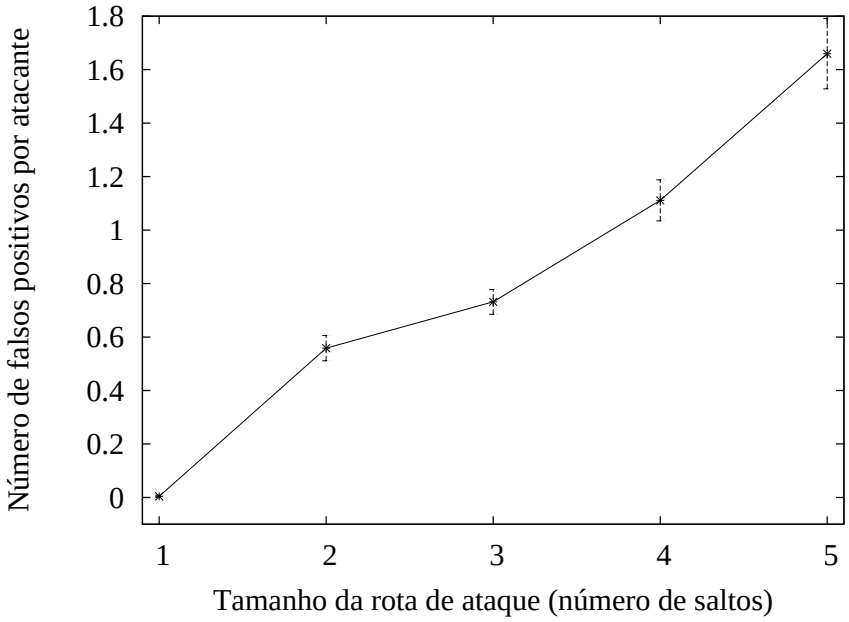


Figura 5.2: Acurácia do sistema proposto em função do tamanho da rota de ataque.

Capítulo 6

Conclusões

Mostra-se nesse trabalho que os sistemas de rastreamento existentes na literatura são ineficazes contra ataques compostos por milhões de estações. Argumenta-se que a melhor opção para rastrear ataques de negação de serviço maciçamente distribuídos (MDDoS) é a abordagem de rastreamento por um único pacote. De fato, os resultados obtidos mostram que as propostas que dependem do recebimento de múltiplos pacotes para poder reconstruir a rota de ataque possuem uma taxa de falsos positivos que cresce rapidamente com o aumento do número de atacantes, comprometendo a escalabilidade do sistema. Por outro lado, a abordagem de rastreamento por um único pacote também possui suas desvantagens. O desafio é satisfazer os requisitos práticos de não armazenar estado na infraestrutura de rede e de usar uma sobrecarga de cabeçalho pequena o suficiente para poder ser alocada no cabeçalho IPv4. Esses requisitos implicam que toda a informação necessária para o rastreamento do atacante deve estar contida em um espaço de cabeçalho insuficiente para armazenar a informação completa da rota de ataque. Para contornar esse problema, propõe-se explorar a estrutura hierárquica da Internet no nível de sistemas autônomos (ASes). É analisado o problema da reconstrução de rotas na Internet no nível de ASes e são obtidas duas descobertas principais: (i) é mostrado que a hierarquia cliente-provedor implica a existência de dois passos críticos durante o procedimento de reconstrução de rota; e (ii) nota-se que o AS de origem pode ser sempre encontrado, mesmo que alguns ASes não participem da marcação de pacotes. Baseando-se nessas duas observações, propõe-se um novo esquema de rastreamento que privilegia os passos críticos da reconstrução de rota e dispensa os

ASes-clientes da tarefa de marcar pacotes. O resultado é que, com somente duas marcações, o AS de origem é encontrado com alta acurácia, a despeito da limitação de espaço disponível para marcação no cabeçalho IPv4. O desempenho do sistema proposto é avaliado numa topologia real e os resultados de simulação confirmam a alta escalabilidade e excelente acurácia do sistema proposto. O sistema é capaz de rastrear um número ilimitado de atacantes com menos de 1 falso positivo por atacante, exigindo apenas a participação dos ASes não-clientes, isto é, apenas 18,5% de razão de implantação.

Tradicionalmente, as propostas de sistemas de rastreamento focam na marcação de pacotes e só então é estudado o problema de reconstruir a rota de ataque usando as marcações fornecidas pelos roteadores. Nesse trabalho, é feita uma mudança de perspectiva. Toma-se o ponto de vista da reconstrução de rota a fim de verificar as informações de rastreamento que são mais importantes para a reconstrução de rota. Assim, pôde-se projetar um esquema de marcação apropriado para a solução do problema da reconstrução de rota. Os resultados obtidos nesse trabalho mostram que essa abordagem é realmente mais eficaz do que a abordagem que foca na marcação de pacotes sem avaliar quais informações são mais importantes para a reconstrução de rota. A principal limitação prática imposta aos sistemas de rastreamento é que o espaço disponível para marcação no cabeçalho IP (25 bits) é insuficiente para representar toda a informação de rastreamento necessária para rastrear o atacante sem nenhum falso positivo. De fato, mostra-se que uma sobrecarga de cabeçalho de 35 bits é necessária para armazenar a informação completa de um caminho em nível de ASes. Com isso, é inevitável a existência de falsos positivos durante a reconstrução de rota. Nesse cenário, a avaliação das informações de rota que são mais importantes é fundamental, pois permite priorizar tais informações em detrimento das informações menos relevantes. Essa é, em última análise, a razão da taxa de falsos positivos do sistema proposto ser bem mais baixa do que a dos sistemas comparados. A estratégia adotada nesse trabalho para priorizar os passos críticos da reconstrução de rota é o uso de pontos de verificação. Os resultados de simulação mostram que essa estratégia é realmente vantajosa para contornar a limitação de espaço para marcação. Considerando o mesmo cenário e a mesma sobrecarga de cabeçalho, o procedimento de reconstrução de rota baseado em pontos de verificação

obteve uma acurácia 1000 vezes melhor do que a solução salto-a-salto ótima.

Um resultado particular do sistema proposto é a escalabilidade em relação ao número de nós da topologia. A observação-chave é que o crescimento da topologia em nível de ASes preserva a sua hierarquia. Como o sistema proposto explora a hierarquia cliente-provedor, mostra-se que ele é escalável em relação ao número de ASes na topologia.

A abordagem de rastreamento por um único pacote permite o uso do sistema de rastreamento como um mecanismo de identificação de fonte por pacote. Além disso, a sobrecarga de processamento introduzida pelo sistema proposto é pequena. Ao marcar o pacote, o roteador de borda do AS só precisa comparar o conteúdo do campo `Controle` com o valor 0 e inserir valores pré-calculados no pacote. Portanto, o sistema proposto é mais eficiente do que os mecanismos de identificação de fonte que utilizam primitivas criptográficas para autenticar o endereço de origem de cada pacote.

Uma questão ainda em aberto é o que fazer com a informação obtida com o sistema de rastreamento. Em geral, propõe-se usar as informações da rota de ataque para filtrar o tráfego de ataque. Outro uso da informação de rastreamento é a responsabilização do atacante. Porém, a informação de rastreamento só permite chegar ao AS de origem do ataque ou ao roteador mais próximo ao atacante. O problema de encontrar o verdadeiro atacante responsável pelo controle da rede de *bots* é chamado de o problema do *stepping stone* [40]. Como o sistema proposto é capaz de rastrear a origem de cada pacote, não é necessário disparar o procedimento de reconstrução de rota para todos os pacotes de ataque recebidos. Basta escolher alguns pacotes e aplicar uma solução para o problema do *stepping stone* a partir dos ASes rastreados. Em ataques por refletor, pode-se realizar novamente o rastreamento a partir da estação-refletora. Além disso, espera-se que o administrador do AS de origem do ataque irá tomar todas as medidas necessárias para não ser acusado de negligência ao permitir que o tráfego de ataque parta da rede sob sua responsabilidade. Assim, o fato de o AS de origem ser identificado é um fator encorajador para a implantação de medidas preventivas e sistemas de rastreamento intra-domínio nos ASes-clientes. Dessa forma, é possível provar a participação de uma estação no ataque, resolvendo finalmente o problema da identificação de fonte.

Vale ressaltar que, apesar da proposta desse trabalho ter detalhes específicos para atender aos requisitos práticos relacionados ao problema do rastreamento na Internet, como a restrição do espaço de marcação a 25 bits, a ideia proposta é mais geral e pode ser utilizada em outros cenários com topologia hierárquica, como sistemas par-a-par (*Peer to Peer* - P2P) hierárquicos [41] e redes complexas [42]. A ideia de pontos de verificação pode ser usada sempre que houver uma informação limitada para buscar objetos em topologias hierárquicas.

Referências Bibliográficas

[1] LAUFER, R. P., MORAES, I. M., VELLOSO, P. B., et al., "Negação de Serviço: Ataques e Contramedidas", Em: *Minicursos do V Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais - SBSeg'2005*, cap. 1, pp. 1–63, Set. 2005.

[2] EHRENKRANZ, T., LI, J., "On the State of IP Spoofing Defense", *ACM Transactions on Internet Technology (TOIT)*, v. 9, n. 2, pp. 1–29, 2009.

[3] MCMILLAN, R., "Porn Site Feud Spawns New DNS Attack", http://www.networkworld.com/news/2009/020509-porn-site-feud-spawns-new.html, Maio 2009.

[4] DEITRICH, D., "Automated Bogon Filtering", http://www.team-cymru.org/documents/bogons-deitrich.pdf, Abril 2006.

[5] MOORE, D., SHANNON, C., BROWN, D. J., et al., "Inferring Internet Denial-of-Service Activity", *ACM Transactions on Computer Systems*, v. 24, n. 2, pp. 115–139, 2006.

[6] BEVERLY, R., BAUER, S., "The Spoofer Project: Inferring the Extent of Source Address Filtering on the Internet". Em: *Proceedings of USENIX Steps to Reducing Unwanted Traffic on the Internet (SRUTI) Workshop*, pp. 53–59, Julho 2005.

[7] LIU, X., LI, A., YANG, X., et al., "Passport: Secure and Adoptable Source Authentication". Em: *5th USENIX Symposium on Network Systems Design and Implementation*, pp. 365–376, Abril 2008.

[8] ANDERSEN, D. G., BALAKRISHNAN, H., FEAMSTER, N., et al., "Accountable Internet protocol (AIP)". Em: *SIGCOMM '08: Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, pp. 339–350, 2008.

[9] SAVAGE, S., WETHERALL, D., KARLIN, A., et al., "Network Support for IP Traceback", *IEEE/ACM Transactions on Networking*, v. 9, n. 3, pp. 226–237, Junho 2001.

[10] SONG, D. X., PERRIG, A., "Advanced and Authenticated Marking Schemes for IP Traceback", *INFOCOM 2001: Proceedings of the 20th IEEE International Conference on Computer Communications*, v. 2, pp. 878–886, 2001.

[11] LAUFER, R. P., VELLOSO, P. B., DE O. CUNHA, D., et al., "Towards Stateless Single-Packet IP Traceback". Em: *LCN '07: Proceedings of the 32nd IEEE Conference on Local Computer Networks*, pp. 548–555, IEEE Computer Society, 2007.

[12] SNOEREN, A. C., PARTRIDGE, C., SANCHEZ, L. A., et al., "Single-Packet IP Traceback", *IEEE/ACM Transactions on Networking*, v. 10, n. 6, pp. 721–734, 2002.

[13] DEAN, D., FRANKLIN, M., STUBBLEFIELD, A., "An Algebraic Approach to IP Traceback", *ACM Transactions on Information and System Security*, v. 5, n. 2, pp. 119–137, 2002.

[14] YAAR, A., PERRIG, A., SONG, D., "FIT: Fast Internet Traceback", *INFOCOM 2005: Proceedings of the 24th IEEE International Conference on Computer Communications*, v. 2, pp. 1395–1406, Março 2005.

[15] SPIESS, K., "Worm 'Storm' Gathers Strength", http://www.neoseeker.com/news/7103-worm-storm-gathers-strength, Set. 2007.

[16] GAUDIN, S., "Storm Worm Botnet More Powerful Than Top Supercomputers", http://www.informationweek.com/news/internet/showArticle.jhtml?articleID=201804528, Set. 2007.

[17] FERGUSON, P., SENIE, D., "RFC 2827: Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing", 2000.

[18] DUAN, Z., YUAN, X., CHANDRASHEKAR, J., "Constructing Inter-Domain Packet Filters to Control IP Spoofing Based on BGP Updates". Em: *INFOCOM 2006: Proceedings of the 25th IEEE International Conference on Computer Communications*, pp. 1–12, Abril 2006.

[19] XIE, Y., YU, F., ABADI, M., "De-anonymizing the Internet Using Unreliable IDs". Em: *SIGCOMM Computer Communication Review*, v. 39, n. 4, pp. 75–86, ACM: New York, NY, USA, 2009.

[20] MCCULLAGH, D., "U.N. Agency Eyes Curbs on Internet Anonymity", http://news.cnet.com/8301-13578_3-10040152-38.html, Set. 2008.

[21] BURCH, H., CHESWICK, B., "Tracing Anonymous Packets to their Approximate Source". Em: *USENIX LISA'00*, pp. 319–327, Nova Orleans, LA, EUA, Dez. 2000.

[22] MOREIRA, M. D. D., LAUFER, R. P., VELLOSO, P. B., et al., "Uma Proposta de Marcação de Pacotes para Rastreamento Robusto a Ataques". Em: *Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais - SBSeg'2007*, Rio de Janeiro, RJ, Brasil, Ago. 2007.

[23] LIU, J., LEE, Z.-J., CHUNG, Y.-C., "Efficient Dynamic Probabilistic Packet Marking for IP Traceback". Em: *IEEE International Conference on Networks - ICON'03*, pp. 475–480, Sydney, Austrália, Set. 2003.

[24] BELLOVIN, S. M., LEECH, M. D., TAYLOR, T., "ICMP Traceback Messages", *Internet Draft: draft-ietf-itrace-04.txt*, Ago. 2003.

[25] BELENKY, A., ANSARI, N., "On Deterministic Packet Marking", *Computer Networks*, v. 51, n. 10, pp. 2677–2700, 2007.

[26] BLOOM, B. H., "Space/Time Trade-offs in Hash Coding with Allowable Errors", *Communications of the ACM*, v. 7, n. 13, pp. 442–426, Julho 1970.

[27] CHOI, K. H., DAI, H. K., "A Marking Scheme Using Huffman Codes for IP Traceback", *International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN'04)*, v. 00, pp. 421, 2004.

[28] GAO, Z., ANSARI, N., "A Practical and Robust Inter-Domain Marking Scheme for IP Traceback", *Computer Networks*, v. 51, n. 3, pp. 732–750, 2007.

[29] PARUCHURI, V., DURRESI, A., BAROLLI, L., "FAST: Fast Autonomous System Traceback". Em: *AINA '07: Proceedings of the 21st International Conference on Advanced Networking and Applications*, pp. 498–505, IEEE Computer Society: Washington, DC, USA, 2007.

[30] MÜHLBAUER, W., FELDMANN, A., MAENNEL, O., et al., "Building an AS-topology Model that Captures Route Diversity", *SIGCOMM Computer Communication Review*, v. 36, n. 4, pp. 195–206, 2006.

[31] COVER, T. M., THOMAS, J. A., *Elements of Information Theory*. 2nd ed. *Wiley Series in Telecommunications*, John Wiley & Sons, Inc., 1991.

[32] HYUN, Y., HUFFAKER, B., ANDERSEN, D., et al., "The IPv4 Routed /24 AS Links Dataset - Jul, 2009", http://www.caida.org/data/active/ipv4_routed_topology_aslinks_dataset.xml.

[33] MAHADEVAN, P., KRIOUKOV, D., FOMENKOV, M., et al., "The Internet AS-level Topology: Three Data Sources and One Definitive Metric", *SIGCOMM Computer Communication Review*, v. 36, n. 1, pp. 17–26, 2006.

[34] SUBRAMANIAN, L., AGARWAL, S., REXFORD, J., et al., "Characterizing the Internet Hierarchy from Multiple Vantage Points". Em: *INFOCOM 2002: Proceedings of the 21st IEEE International Conference on Computer Communications*, v. 2, pp. 618–627, 2002.

[35] HUSTON, G., "32-bit Autonomous System Number Report", http://www.potaroo.net/tools/asn32/, Março 2009.

[36] AL-DUWAIRI, B., DANIELS, T., "Topology Based Packet Marking", *Computer Communications and Networks*, pp. 146–151, Out. 2004.

[37] LI, J., SUNG, M., XU, J., et al., "Large-Scale IP Traceback in High-Speed Internet: Practical Techniques and Theoretical Foundation", *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 115–129, Maio 2004.

[38] GOODRICH, M. T., "Efficient Packet Marking for Large-Scale IP Traceback". Em: *CCS '02: Proceedings of the 9th ACM Conference on Computer and Comunications Security*, pp. 117–126, ACM, 2002.

[39] "The CAIDA AS Relationships Dataset, 2004-2009", http://www.caida.org/data/active/as-relationships.

[40] ZHANG, Y., PAXSON, V., "Detecting Stepping Stones". Em: *Proceedings of the 9th USENIX Security Symposium*, pp. 171–184, 2000.

[41] GARCÉS-ERICE, L., BIERSACK, E., FELBER, P. A., et al., "Hierarchical Peer-to-peer Systems". Em: *Proceedings of ACM/IFIP International Conference on Parallel and Distributed Computing (Euro-Par)*, pp. 643–657, 2003.

[42] NEWMAN, M., "The Structure and Function of Complex Networks", 2003.

Apêndice A

Conceitos e Resultados da Teoria da Informação

A análise apresentada no Capítulo 3, que avalia a sobrecarga de cabeçalho exigida pelo sistema ótimo de rastreamento por um único pacote, utiliza resultados analíticos derivados da teoria da informação [31]. Por isso, nesse apêndice, são apresentados brevemente alguns conceitos básicos da teoria da informação, aplicando os conceitos ao problema do rastreamento. Além disso, são também demonstrados alguns resultados usados no Capítulo 3.

A.1 Codificação sem Perdas

A.1.1 Definição de Codificação

Seja X uma variável aleatória discreta que pode assumir os valores do conjunto $S = \{s_0, s_1, \ldots, s_{N-1}\}$, com probabilidades

$$Pr(X = s_i) = p_i, \ \forall\, i \in \{0, 1, \ldots, N-1\}. \tag{A.1}$$

Define-se codificação como sendo o processo que representa cada elemento s_i por uma palavra de código, geralmente binária. Por exemplo, aplicando a ideia de codificação para o problema do rastreamento, considere que $S = \{AS_1, AS_2, AS_3, AS_4\}$ é o conjunto de vizinhos de um dado AS e que o pacote de ataque possui a mesma probabilidade de ter passado por cada AS_i de S, isto é,

$p_i = Pr(X = AS_i) = 1/4,\ \forall\, i \in \{1, 2, 3, 4\}$. Assim, a Tabela A.1 apresenta um exemplo de código que usa 2 bits para representar cada elemento de S.

Tabela A.1: Exemplo de codificação do conjunto $S = \{AS_1, AS_2, AS_3, AS_4\}$

Elemento	**Palavra de código**
AS_1	00
AS_2	01
AS_3	10
AS_4	11

A.1.2 Função de Entropia

Define-se o comprimento médio τ de um código para o conjunto $\mathrm{S} = \{s_0, s_1, \ldots, s_{N-1}\}$ como

$$\tau \stackrel{\triangle}{=} \sum_{i=0}^{N-1} p_i.l_i, \tag{A.2}$$

onde l_i, geralmente dado em bits, é o comprimento da i-ésima palavra de código. Um resultado bem conhecido é que os comprimentos médios que minimizam τ são dados por $\overset{*}{l_i} = -\log_2(p_i)$, de forma que o número mínimo de bits necessários para representar o conjunto S é

$$H(X) \stackrel{\triangle}{=} \overset{*}{\tau} = \sum_{i=0}^{N-1} p_i.\overset{*}{l_i} = -\sum_{i=0}^{N-1} p_i.\log_2(p_i). \tag{A.3}$$

$H(X)$ é conhecida como a função de entropia, que é uma medida da quantidade de informação de X. Assim, em um processo de codificação sem perdas, qualquer código para S possui um comprimento médio de palavra de código de pelo menos $H(X)$ bits. Um código é dito ótimo se este representa o conjunto S usando $H(X)$ bits em média. Por outro lado, se forem usados menos do que $H(X)$ bits, então há perda de informação no processo de codificação.

Uma variável aleatória discreta é uniforme se $p_i = 1/N,\ \forall\, i \in \{0, 1, \ldots, N-1\}$. Assim, é fácil mostrar que

$$\text{X é uniforme} \Leftrightarrow H(X) = \log_2(N). \tag{A.4}$$

Portanto, para identificar inequivocamente um entre N elementos equiprováveis, são necessários $\log_2(N)$ bits. Esse resultado é usado na Seção 3.1.1.

A.2 Codificação com Perdas

A teoria da distorção de taxa é usada quando se deseja atingir taxas de compressão abaixo da entropia. A ideia básica é aceitar, de forma controlada, erros durante o processo de codificação. Assim, a variável aleatória X é mapeada para $\hat{X}$, que é a representação de S usando $R < H(X)$ bits. O conjunto associado a $\hat{X}$ é $\hat{S}$. A taxa de erro do mapeamento é medida usando uma função de distorção. Nesse trabalho, é considerada a distorção de Hamming

$$d(X, \hat{X}) \triangleq \begin{cases} 0 & \text{se } X = \hat{X}, \\ 1 & \text{se } X \neq \hat{X}. \end{cases} \tag{A.5}$$

O valor esperado da distorção de Hamming é uma probabilidade de erro, visto que $E[d(X, \hat{X})] = Pr(X \neq \hat{X})$.

A.2.1 Função de Distorção de Taxa

Na codificação com perdas, a medida da quantidade de informação é dada pela função de distorção de taxa, que é uma extensão do conceito de entropia. Antes de apresentar a definição da função de distorção de taxa, é necessário introduzir os conceitos de entropia condicional e informação mútua. A entropia condicional, $H(X|\hat{X})$, mede a quantidade de informação que resta acerca de X quando se conhece $\hat{X}$, e é expressa por

$$H(X|\hat{X}) \triangleq \sum_{x \in S} \sum_{\hat{x} \in \hat{S}} p(x, \hat{x}). \log_2 \left(\frac{1}{p(x|\hat{x})} \right), \tag{A.6}$$

onde $p(x, \hat{x}) \triangleq Pr(X = x \, e \, \hat{X} = \hat{x})$ é a distribuição de probabilidade conjunta e $p(x|\hat{x}) \triangleq Pr(X = x \mid \hat{X} = \hat{x})$ é a distribuição de probabilidade condicional. A informação mútua, $I(X, \hat{X})$, entre as variáveis aleatórias X e $\hat{X}$ é definida como a diferença

$$I(X, \hat{X}) \triangleq H(X) - H(X|\hat{X}). \tag{A.7}$$

A informação mútua é, portanto, a quantidade de informação de X que passa a ser conhecida quando se conhece $\hat{X}$. Como $H(X)$ é a quantidade de informação conhecida acerca de X antes do mapeamento de X para $\hat{X}$, então a quantidade de informação conhecida acerca de X após o mapeamento é dada pela informação

mútua $I(X, \hat{X})$. Por essa razão, a função de distorção de taxa, $R(D)$, é dada diretamente pelo valor de $I(X, \hat{X})$, com a restrição de que a probabilidade de erro seja limitada pela constante D, sendo definida como

$$R(D) \stackrel{\triangle}{=} \min_{p(X,\hat{X}) \mid E[d(X,\hat{X})] \leq D} [I(X, \hat{X})], \tag{A.8}$$

onde a minimização é realizada sobre todas as possíveis distribuições de probabilidade conjuntas $p(x, \hat{x})$ para as quais a restrição da probabilidade de erro é satisfeita. A função de distorção de taxa fornece o número mínimo $R(D)$ de bits necessários para representar o conjunto S, aceitando uma probabilidade de erro máxima igual a D. Esse resultado é usado na Seção 3.1.2.

A.2.2 Desigualdade de Fano

Seja X uma variável aleatória discreta que pode assumir N valores equiprováveis. Considerando uma probabilidade de erro máxima, D, no mapeamento de X para $\hat{X}$, a desigualdade de Fano diz que a entropia condicional $H(X|\hat{X})$ é limitada superiormente segundo a expressão

$$H(X|\hat{X}) \leq H(D) + D.\log_2(N-1), \tag{A.9}$$

onde $H(.)$ é a função de entropia binária

$$H(D) = D.\log_2 D + (1-D).\log_2(1-D). \tag{A.10}$$

A.2.3 Função de Distorção de Taxa para N Elementos Equiprováveis

No Capítulo 3, a análise da solução ótima usando somente os 25 bits disponíveis para marcação no cabeçalho IPv4 usa a expressão analítica da função de distorção de taxa para o caso em que X é uniforme. Assim, considerando que X pode assumir

N valores equiprováveis, tem-se

$$
\begin{aligned}
I(X, \hat{X}) &= H(X) - H(X|\hat{X}) && \text{(A.11a)} \\
&= \log_2(N) - H(X|\hat{X}) && \text{(A.11b)} \\
&\geq \log_2(N) - H(D) - D.\log_2(N-1) && \text{(A.11c)} \\
&= \log_2(N) - D.\log_2 D - (1-D).\log_2(1-D) - D.\log_2(N-1). && \text{(A.11d)}
\end{aligned}
$$

O passo A.11a é decorrente da definição de informação mútua (Equação A.7). No passo A.11b foi usada a Equação A.4. O passo A.11c é consequência da desigualdade de Fano (Equação A.9). Finalmente, no passo A.11d, é usada a expressão de $H(D)$ da Equação A.10. A Equação A.11 mostra que $I(X, \hat{X})$ possui um limite inferior. Este limite inferior tem que ser menor ou igual ao valor mínimo de $I(X, \hat{X})$, pois o valor mínimo é o maior dos limites inferiores. Portanto, da definição da função de distorção de taxa (Equação A.8), tem-se que

$$
\begin{aligned}
R(N, D) &= \min_{p(X,\hat{X}) \mid E[d(X,\hat{X})] \leq D} [I(X, \hat{X})] \\
&\geq \log_2(N) - D.\log_2 D - (1-D).\log_2(1-D) - D.\log_2(N-1).
\end{aligned}
\qquad \text{(A.12)}
$$