Mitigação de Ataque Low-Rate DoS no Protocolo MPTCP

Danilo Possati, Benevid Felix, Aldri Santos, Michele Nogueira

[1]Centro de Ciência de Segurança Computacional (CCSC)
Universidade Federal do Paraná (UFPR)

{drpossati,bfsilva,aldri,michele}@inf.ufpr.br

Abstract. *Denial of Service (DoS) attacks have become increasingly sophisticated and have caused increasing damage to the services available on the Internet. Low-Rate DoS (LDoS) attack is a type of DoS attack that severely degrades transmissions through bursts of periodic flows while circumventing traditional detection systems. Multipath TCP (MPTCP) is an evolution of TCP that provides multiple subflows on the same connection, improving transmission performance and resiliency. However, LDoS attacks that occur over a bottleneck, shared by MPTCP subflows, can severely affect the multipath transmission. This work investigates such scenario, since it is a worst case, and proposes a technique to provide subflows with random RTOs (Retransmission Timeout) to mitigate LDoS. Results show that random RTOs in subflows decrease the LDoS effects, mainly when the attack occurs in a bottleneck shared by MPTCP subflows.*

Resumo. *Os ataques de negação de serviço (DoS) têm se tornado cada vez mais sofisticados e causado danos crescentes aos serviços disponíveis na Internet. O ataque Low-Rate DoS (LDoS) é um tipo de ataque que degrada severamente as transmissões por meio de rajadas de fluxos periódicas enquanto burla os sistemas tradicionais de detecção. O Multipath TCP (MPTCP) é uma evolução do TCP que provê múltiplos subfluxos em uma mesma conexão, provendo maior desempenho e aumentando a resiliência das transmissões. No entanto, um ataque LDoS direcionado a um gargalo compartilhado pelos subfluxos MPTCP podem afetar drasticamente a transmissão. Este trabalho investiga o comportamento deste cenário por ser considerado o pior caso e propõe uma técnica que provê subfluxos com RTOs (Retransmission Timeout) aleatórios a fim de mitigar o LDoS. Os resultados obtidos por emulação demonstram que a aleatorização do RTO reduz os efeitos do LDoS, especialmente quando o ataque ocorre em um gargalo compartilhado entre os subfluxos MPTCP.*

1. Introdução

A popularização dos dispositivos móveis tem levado a uma maior dependência das pessoas nos serviços disponíveis na Internet. No entanto, esta popularização tem incentivado a evolução das ameaças que afetam a disponibilidade dos serviços presentes na Internet. O ataque de negação de serviço (*Denial-of-Service* – DoS) é uma destas ameaças [Lima et al. 2009, Macedo et al. 2015]. Em geral, a negação do serviço resulta da geração de um grande volume de tráfego na rede a fim de comprometer o servidor ou um enlace, impossibilitando o servidor de responder a todas as solicitações

dos clientes. Devido ao grande volume de tráfego, o DoS pode ser detectado e mitigado [Sharma and Gupta 2018]. Sendo assim, formas mais sofisticadas de provocar a negação de serviço e burlar os mecanismos de detecção e mitigação, como o *Low-Rate DoS* (LDoS) [Kuzmanovic and Knightly 2003], vêm sendo exploradas. Este artigo foca em uma das variantes do LDoS, a qual, explora uma vulnerabilidade no algoritmo de controle de congestionamento de um dos principais protocolos da Internet, o TCP. A variação do LDoS tratada nega silenciosamente o serviço utilizando rajadas de dados periódicas de alta taxa e curta duração. Isto induz a perda de pacotes e faz com que a conexão entre em um processo repetitivo de *timeout* e retransmissão (*Retransmission Timeout* – RTO), diminuindo sucessivamente a janela de congestionamento (CWND) até próximo de zero. As rajadas periódicas e de curta duração do LDoS são de difícil detecção [Chen et al. 2017], uma vez que os períodos (P) do ataque coincidem com os períodos das retransmissões (RTO) dos fluxos, sendo confundidos como tráfego legítimo.

Atualmente a maioria dos dispositivos móveis possui mais de uma interface de comunicação, permitindo a conexão simultânea com diferentes estruturas de redes. Estes dispositivos, denominados *multihomed*, buscam aumentar a taxa de transferência dos dados e tornar as conexões mais confiáveis [Nguyen et al. 2017]. Para suportar estas caracteristicas, o *Multipath TCP* (MPTCP) vem sendo desenvolvido. Ele possibilita o uso simultâneo das interfaces disponíveis por meio da criação de múltiplos subfluxos [Ford et al. 2013]. O MPTCP transmite os dados sobre todos os subfluxos estabelecidos e utiliza um algoritmo de controle de congestionamento acoplado para que possa atender aos princípios do *TCP-Friendly*. O controle de congestionamento do MPTCP tem como objetivo principal melhorar a taxa de transferência e gerenciar o tráfego entre os subfluxos. Uma de suas funções é priorizar os caminhos com menor atraso, direcionando a maior quantidade de tráfego para subfluxos menos congestionados [Raiciu et al. 2011].

No passado, diferentes trabalhos investigaram as consequências do ataque LDoS sobre o TCP [Kuzmanovic and Knightly 2003, Wu et al. 2011, Luo and Yang 2014]. Em [Chen et al. 2017, Bhuyan et al. 2015, Xiang et al. 2011, Chang et al. 2010], os autores introduziram métodos de identificação e mitigação do ataque LDoS no TCP. Por outro lado, em [Luo and Yang 2014], os autores apresentaram formas de intensificar o ataque. As abordagens que visam mitigar o ataque se dividem em técnicas direcionadas aos nós da rede, como roteadores e sistemas de detecção, e técnicas empregadas diretamente nos nós finais, isto é, aplicadas ao TCP. Em relação a esta última, nenhum trabalho abordou o impacto do ataque LDoS sobre o MPTCP [Paasch et al. 2013]. Diferente do TCP, o MPTCP possui múltiplos subfluxos e seu controle de congestionamento direciona o tráfego para os caminhos menos congestionados. Com isto, caso o ataque atinja parcialmente os subfluxos, ele seria facilmente mitigado. No entanto, o pior caso para o MPTCP é se todos os subfluxos compartilham o gargalo sobre ataque. Isto porque o MPTCP utiliza um valor inicial comum de RTO para todos os subfluxos. Assim, o comportamento do ataque seria similar a múltiplos fluxos TCP, abordado nos trabalhos citados acima.

Neste sentido, este trabalho apresenta uma avaliação do MPTCP considerando o cenário mais vulnerável (seu pior caso de uso) e propõe uma técnica para mitigar o ataque LDoS. A técnica de mitigação desenvolvida visa gerenciar dinamicamente a criação de subfluxos com o tempo mínimo de retransmissão (*minRTO*) aleatorizado, a fim de reduzir o número de subfluxos que tenham um RTO que coincida com o período do ataque LDoS.

Os resultados de simulação, usando um cenário emulado com Mininet, mostram que a técnica de mitigação reduz os efeitos do ataque, principalmente para o período de *minRTO* padrão empregado no MPTCP, momento em que o ataque tem mais efeito.

O restante do trabalho está organizado como segue. A Seção 2 discute os trabalhos relacionados. A Seção 3 descreve o ataque LDoS e os detalhes de funcionamento do MPTCP. A Seção 4 detalha a metodologia de avaliação e os resultados do impacto do ataque. A Seção 5 apresenta uma abordagem para mitigação do LDoS sobre o MPTCP e os resultados. Por fim, a Seção 6 apresenta as conclusões e direções futuras.

2. Trabalhos Relacionados

O trabalho de [Kuzmanovic and Knightly 2003] foi o precursor em descrever o ataque LDoS (ou ataque *Shrew*[1]). Os autores descreveram como o ataque obtém sucesso sobre uma vulnerabilidade na definição do temporizador de retransmissões (RTO) do controle de congestionamento do TCP. [Kuzmanovic and Knightly 2006] exploraram o impacto do ataque LDoS com fluxos heterogêneos (diferentes RTTs). Eles avaliaram e compararam a eficiência do ataque em relação a contramedidas aplicadas no roteador (controle de fila) e no TCP. Os autores concluíram que as contramedidas de detecção, direcionadas aos roteadores (RED-PD ou CHOKe) e aplicadas ao TCP, não estão aptas a eliminar os efeitos do ataque LDoS. Os autores demonstraram uma otimização do ataque que ainda pode ser bem severa aos fluxos TCP legítimos, mesmo em um ambiente com fluxos heterogêneos.

[Kuzmanovic and Knightly 2006] destacaram a mitigação como uma importante área de estudo. Nesse sentido, as técnicas empregadas ao longo dos anos para mitigar o ataque LDoS se concentraram principalmente nos roteadores. [Shevtekar et al. 2005] propuseram um sistema de detecção de ataque LDoS para os roteadores de borda. Para detecção do ataque, os autores identificam uma propriedade baseada na diferença dos tempos de chegada dos pacotes de cada fluxo. Segundo os autores, o fluxo de ataque apresenta uma média alta e repete periodicamente, enquanto os demais fluxos não exibem esta propriedade. No entanto, o trabalho não apresenta resultados detalhados sobre a efetividade da proposta. Em [Efstathopoulos 2009], Efstathopoulos complementou o trabalho anterior fazendo uma avaliação experimental do ataque LDoS em redes reais. Os resultados confirmaram o forte impacto do ataque LDoS.

Em [Chang et al. 2010], foi apresentado um mecanismo de filtragem baseado em marcações de prioridade, denominado SAP (*Shrew Attack Protection*). O mecanismo SAP tem como objetivo proteger os fluxos TCP comportados contra os fluxos de ataques LDoS. Contudo, se o atacante troca constantemente as portas de destino, ele pode se evadir do SAP. [Xiang et al. 2011] propuseram duas métricas – entropia generalizada e distância da informação – para a detecção do ataque. As métricas são baseadas em anomalias e não somente identificam previamente os ataques, como também produzem baixas taxas de falsos positivos. [Bhuyan et al. 2015] apresentaram uma discussão empírica sobre as métricas de informação (entropia) utilizadas para detectar os ataques LDDoS. Eles avaliaram as métricas a partir do ponto de vista do atacante em um contexto de detecção para os ataques DDoS com alta e baixa taxa de transferência. Em [Chen et al. 2017], os autores propuseram uma nova métrica de detecção (*Power Spectrum Density Entropy*).

[1]Menção a um pequeno mamífero, conhecido como musaranho, que atacam e devoram animais com o dobro de seu tamanho - http://en.wikipedia.org/wiki/Shrew

Esta métrica é vantajosa pelo fato que os ataques LDoS e os fluxos TCP normais possuem diferentes *Power Spectrum Density*.

No trabalho [Luo and Yang 2014], os autores exploram uma nova variação do ataque LDoS direcionado para o TCP. O ataque usa as deficiências do mecanismo de temporização (*timeout*) e a partida lenta (*Slow Start*) do TCP. Assim como a variação inicial do ataque, o novo ataque visa degradar de forma significativa o TCP, enquanto se evade de sistemas de detecção por consumir uma pequena parte da capacidade da rede. O ataque supera a variação inicial do LDoS mesmo com uma taxa de transferência de dados 47% menor. Ele aumenta a eficiência de degradação sobre a vazão em 45%. Mesmo com mecanismos de defesa, o ataque possui uma taxa de degradação de 11%. O trabalho apontou o mecanismo *slow start* como a principal vulnerabilidade do TCP.

Poucos trabalhos se propuseram a investigar com mais detalhes o impacto do ataque em diferentes sistemas, cenários, protocolos e aplicações. Em [Wu et al. 2011], os autores apresentaram um estudo sucinto sobre dois tipos de ataques DDoS, o Flood DDoS (FDDoS) e o Low-rate DDoS (LDDoS). Os ataques foram estudados conforme seu princípio de geração, mecanismo de utilização, comportamento, assinatura, e desempenho. Contudo, além de limitado ao TCP, o artigo não descreve com detalhes os resultados. No melhor do nosso conhecimento, nenhum trabalho investiga o impacto do ataque sobre protocolos emergentes e com grande aceitação, como o MPTCP. Embora muitos estudos tenham sido feitos sobre o ataque LDoS, ele ainda impõe grandes desafios aos pesquisadores e alguns trabalhos, como [Luo and Yang 2014], mostram a importância de ter soluções para o problema, principalmente no contexto de redes de comunicação e protocolos de transporte.

3. Ataque *Low-Rate* DoS e o TCP Multicaminhos

O protocolo TCP foi projetado para redes cabeadas, em que a ocorrência de falhas na transmissão é muito pequena e uma perda de pacote geralmente indica a existência de um congestionamento. Ao longo dos anos, novos mecanismos de controle de congestionamento foram propostos, especialmente para melhorar o desempenho em redes sem fio e redes de alto desempenho. Exemplos são o TCP CUBIC, TCP RENO, TCP FAST e outros [Duke et al. 2015, Braun et al. 2008]. Atualmente, o controle de congestionamento comumente em uso na Internet aplica o método de prevenção de congestionamento baseado em perdas (*Loss-based Congestion Avoidance* – LCA). O LCA tem como referência a perda de pacotes como indicativo de congestionamento na rede. Um evento de perda resulta em uma redução no tamanho da janela de congestionamento (CWND), diminuindo a carga na rede. Embora seja um mecanismo essencial para controle de fluxos em vigor na Internet, ele possui vulnerabilidades.

Os ataques do tipo Low-Rate DoS provocam constantes reduções da CWND para comprometer o desempenho das transmissões TCP e degradar a qualidade de serviço (QoS) das aplicações que utilizam este protocolo [Kuzmanovic and Knightly 2003]. O ataque LDoS é um tipo especial de ataque de negação de serviço, e sua metodologia consiste em encaminhar periodicamente rajadas de pacotes de curta duração de modo a explorar os mecanismos de controle de congestionamento do TCP. Ele é usualmente caracterizado pelos parâmetros taxa de ataque (T), duração do ataque por período (D) e o período do ataque (P) (Figura 1). A execução do ataque ocorre da seguinte forma. O

atacante define T igual à largura de banda do gargalo e um determinado valor para D, por exemplo, 100ms. O objetivo é inundar a fila dos roteadores com os pacotes de ataque em um curto período de tempo de modo que os pacotes dos fluxos normais sejam descartados. A perda de pacotes força os fluxos TCP entrarem na fase de partida lenta (*slow start*), devido ao estouro do tempo limite de retransmissão (RTO); ou na fase de recuperação rápida, ao receber três reconhecimentos duplicados (Dup-ACK). Independentemente do estado, os fluxos TCP reduzem suas janelas de congestionamento para limitar a taxa de envio de pacotes, resultando na degradação da vazão e da Qualidade de Serviço (QoS).

O controle de congestionamento do TCP opera em duas escalas de tempo tomando como referência o RTT e o RTO. No RTT (a escala de tempo menor), o TCP executa o incremento aditivo e decremento multiplicativo de modo que todos os fluxos TCP transmitam em uma taxa justa quando compartilham um gargalo. Em caso de perdas de pacotes, o TCP opera em uma escala de tempo maior. Ele aguarda expirar o temporizador de retransmissão (RTO), então reduz sua janela de congestionamento para um segmento e reenvia o pacote não reconhecido. Se ocorrerem mais perdas, o RTO é duplicado a cada temporizador esgotado (*timeout*) subsequente. Os algoritmos padrões de controle de congestionamentos utilizam a perda de pacotes como sinal de congestionamento [Paxson et al. 2011].

O valor mínimo e inicial para o RTO é definido como 1 segundo, contudo este valor é constantemente atualizado pelo emissor com base no RTT (*Round-Trip Time*) da conexão. No cálculo do RTO são utilizadas duas variáveis de estado, `SRTT` (*Smoothed Round-Trip Time*) e `RTTVAR` (*Round-Trip Time Variation*). Na primeira aferição R do RTT, estas variáveis são alteradas da seguinte maneira, `SRTT` = R, `RTTVAR` = $R/2$ e o RTO = `SRTT` + $max(G, K * \texttt{RTTVAR})$, sendo $K = 4$ e G a granularidade do relógio. Quando uma medida R' do RTT subsequente é tomada, o emissor atualiza estes valores da seguinte forma, `RTTVAR` = $(1 - \beta) * \texttt{RTTVAR} + \beta * |\texttt{SRTT} - R'|$ e o `SRTT` = $(1 - \alpha) *$ `SRTT` $+ \alpha * R'$, sendo $\alpha = 1/8$ e $\beta = 1/4$. Vale salientar que o `SRTT` usado na atualização do `RTTVAR` é o `SRTT` anterior a sua atualização. Por fim, um novo valor é setado para o RTO = $max(minRTO, \texttt{SRTT} + max(G, k * \texttt{RTTVAR})$ [Paxson et al. 2011].

Como descrito em [Kuzmanovic and Knightly 2003], o ataque Low-Rate DoS explora o mecanismo de retransmissão do TCP por meio de rajadas repetidas com uma alta taxa de fluxo de dados, porém, com curta duração e por períodos fixos, os quais são escolhidos maliciosamente a fim de coincidir com o tempo do RTO. Com uma taxa do ataque na escala de tempo RTT, e suficiente para induzir a perda de pacotes, os pacotes estarão no tempo limite de retransmissão e uma nova tentativa de retransmissão ocorrerá no fim do RTO. Se os momentos do ataque se aproximarem do tempo de RTO, a conexão continuará perdendo pacotes e falhará ao tentar sair do estado de retransmissão. O TCP é

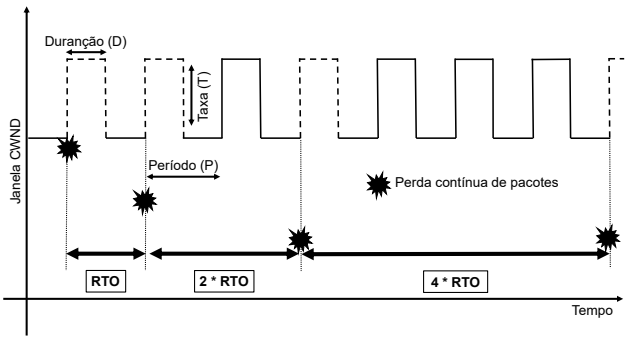


Figura 1. Parâmetros de um ataque LDoS [Chang et al. 2010]

comprometido com sua taxa de transferência se aproximando de zero enquanto o ataque mantém uma taxa média baixa, dificultando a sua detecção. O MPTCP também poderá ser prejudicado por este ataque, pois seus subfluxos se comportam como fluxos TCP e seu algoritmo de controle de congestionamento padrão também toma como base as perdas de pacotes para controle da janela CWND e das retransmissões [Raiciu et al. 2011].

3.1. TCP Multicaminhos (MPTCP)

Esta subseção descreve as características do protocolo MPTCP, bem como seu modo de operação e o funcionamento do seu algoritmo de controle de congestionamento. O protocolo MPTCP [Barré et al. 2011a, Felix et al. 2016] tem como base o protocolo TCP e possibilita a transmissão de dados através de múltiplos caminhos. O MPTCP é o esforço mais recente de padronizar um protocolo de transporte multicaminhos confiável para uso genérico na Internet. Um dos principais objetivos do MPTCP como protocolo de transporte multicaminhos trata-se de obter um desempenho melhor ao oferecido por um único caminho TCP, em termos de vazão e latência [Bonaventure et al. 2012]. O MPTCP também possibilita aumentar a resiliência da comunicação, utilizando os caminhos de forma redundante para persistir a conexão em caso de falhas de um ou outro caminho.

Uma comunicação MPTCP provê a troca de dados bidirecional entre dois nós, não requerendo qualquer mudança na aplicação. O MPTCP possibilita que os nós utilizem diferentes caminhos e diferentes endereços para transmitir pacotes pertencentes a uma mesma conexão. Para a camada de rede, cada subfluxo MPTCP é visto como um fluxo TCP padrão, mas que transportam no cabeçalho TCP um tipo de campo *Option* específico do MPTCP. Todas as operações do MPTCP são sinalizadas através do campo *Option*. Para controlar o envio de pacotes através de diferentes caminhos o MPTCP possui dois níveis de reconhecimento: por subfluxo (*Subflow Sequence Number* – SSN) e por conexão (*Data Sequence Number* – DSN). Os reconhecimentos SSN, utilizados no TCP padrão, são empregados para confirmar o recebimento dos segmentos em cada subfluxo independente do DSN. Os reconhecimentos DSN são utilizados em nível de conexão.

A comunicação entre dois nós utilizando o MPTCP pode ser resumida nos seguintes passos [Ford et al. 2013]. Uma conexão MPTCP inicia com um *three-way handshake*, similar ao TCP. Conforme ilustra a Figura 2, uma conexão MPTCP é estabelecida entre os endereços $A1$ e $B1$ dos nós A e B, respectivamente. Se existem caminhos extras, sub-

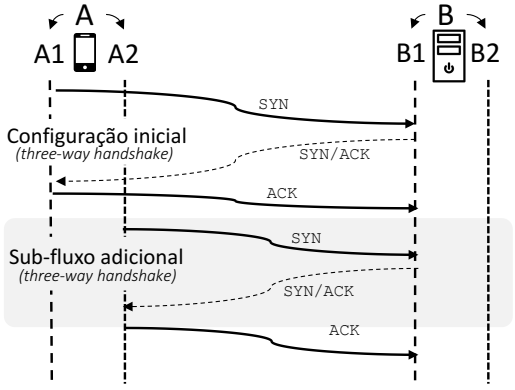


Figura 2. Estabelecimento de subfluxos

fluxos adicionais são criados entre os dois nós e combinados com a conexão, de modo que os múltiplos caminhos sejam transparentes às aplicações. Como ilustrado na Figura 2, um subfluxo adicional é criado entre os endereços $A2$, pertencente ao nó A, e $B1$, pertencente ao nó B. O MPTCP identifica os múltiplos caminhos através dos múltiplos endereços dos nós. A combinação destes endereços permite a criação de caminhos adicionais. No exemplo da Figura 2, outros subfluxos adicionais podem serem criados entre $A1 - B2$ e $A2 - B2$. Estes subfluxos podem ser iniciados tanto pelo nó A, quanto pelo nó B.

A arquitetura funcional do MPTCP [Ford et al. 2011] é composta por quatro entidades, sendo o gerenciador de caminhos [Ford et al. 2013], o escalonador de pacotes, o controle de congestionamento [Raiciu et al. 2011] e a interface de subfluxos. O escalonador, o controle de congestionamento e a interface de subfluxos operam exclusivamente na camada de transporte. O gerenciador de caminhos não se restringe a uma determinada camada, podendo ser implementado fora da camada de transporte. O gerenciador de caminhos se responsabiliza pela descoberta dos caminhos (i.e., subfluxos) entre os dois nós. O escalonador de pacotes, após receber um fluxo de dados da aplicação, realiza suas operações, como segmentar os dados, adicionar um número de sequência em nível de conexão e distribuir os segmentos para os subfluxos conforme política adotada. Um subfluxo, ao receber o segmento, adiciona o seu próprio número de sequência e encaminha o segmento para rede. No destinatário, os dados são ordenados em nível de subfluxo, caso necessário, e passados ao escalonador, que caso necessário também faz o reordenamento em nível de conexão. Por fim, o controle de congestionamento existe como parte do escalonador, realizando, entre outras tarefas, o controle da taxa de envio e a prevenção de congestionamentos por subfluxo.

O mecanismo de controle de congestionamento (CC) é responsável por controlar o fluxo de dados em cada subfluxo através da janela de congestionamento (CWND). Uma das principais tarefas do CC é balancear o tráfego entre os subfluxos, evitando aqueles mais congestionados e garantindo um compartilhamento justo em um gargalo compartilhado entre os subfluxos com outros fluxos (e.g., TCP ou MPTCP) [Barré et al. 2011b]. O CC executado pelo MPTCP tem como base os algoritmos do TCP, como a variante *NewReno*. Todos eles modificam apenas a fase de prevenção de congestionamento (*Congestion Avoidance*), ficando as fases de inicialização lenta (*Slow Start*), retransmissão rápida (*Fast Retransmit*) e recuperação rápida (*Fast Recovery*) inalteradas [Singh et al. 2013]. O CC é executado em nível de subfluxo, usando uma janela CWND individual e uma janela de recebimento (RWND) compartilhada para facilitar a entrega ordenada dos dados. Seus requisitos principais incluem melhorar a vazão, não prejudicar fluxos concorrentes e balancear o congestionamento. A vazão dos subfluxos em uma transferência multicaminhos deve ser, no mínimo, tão eficiente quanto um único fluxo através do melhor caminho. Os subfluxos precisam se comportar como um único fluxo na existência de um gargalo compartilhado. Por fim, o tráfego deve ser direcionado aos caminhos menos congestionados.

A maioria dos algoritmos de CC propostos para o MPTCP aplicam diferentes abordagens às fases de incremento aditivo e decremento multiplicativo. O algoritmo LIA (*Linked Increase Algorithm*), adotado como padrão pelo IETF [Ford et al. 2013], especifica apenas como deve ocorrer o incremento da janela de congestionamento ao receber um ACK, mantendo o decremento padrão do TCP. O crescimento das janelas dos subfluxos são acoplados (Eq. 1). O parâmetro α (Eq. 2) controla a agressividade dos subfluxos de

modo que o incremento da janela não seja superior ao de um fluxo de caminho único com o mesmo tamanho de janela. Nas Eqs. 1 e 2, w_i e RTT_i referem-se ao tamanho da janela de congestionamento e o tempo de ida e volta (RTT) de um subfluxo i, respectivamente. O parâmetro W representa o tamanho total das janelas de congestionamento.

$$w_i = \begin{cases} min(\alpha/W, 1/w_i), & incremento \\ w_i/2, & decremento \end{cases} \quad (1)$$

$$\alpha = W * \frac{max\left(w_i/RTT_i^2\right)}{\left(\sum_i w_i/RTT_i\right)^2} \quad (2)$$

Um dos objetivos do algoritmo de controle de congestionamento é garantir o compartilhamento justo da largura de banda (i.e., princípio de *fairness*) e o uso eficiente dos caminhos. Novos algoritmos de CC foram propostos, como OLIA (*Opportunistic Linked Increase Algorithm*) [Khalili et al. 2012] e BALIA (*Balanced Linked Adaptation Congestion Control Algorithm*) [Ferlin et al. 2016]. No entanto, como o IETF define o LIA como algoritmo padrão, este trabalho propõe inicialmente investigá-lo.

4. Análise do Impacto do LDoS sobre o MPTCP

O protocolo MPTCP utiliza múltiplos subfluxos TCP para transmitir os dados, e caso estes subfluxos compartilhem um gargalo, o ataque LDoS pode ser mais efetivo. Isto ocorre porque o MPTCP utiliza em cada subfluxo o mesmo valor de `minRTO` ($R = 1s$). Conforme ilustra a Figura 3, os subfluxos MPTCP $A1$ e $A2$, provenientes dos endereços do nó Alice, possuem respectivamente R_1 e R_2 como valores de `minRTO`, sendo $R_1 \Leftrightarrow R_2$. Um ataque LDoS que seja direcionado a um gargalo r, compartilhado por $A1$ e $A2$, e que possua um período de ataque (P) que coincida com R_1 e R_2, tem grandes chances de afetar ambos os caminhos com a mesma intensidade e prejudicar a conexão MPTCP como um todo, assim como acontece no protocolo TCP [Kuzmanovic and Knightly 2003].

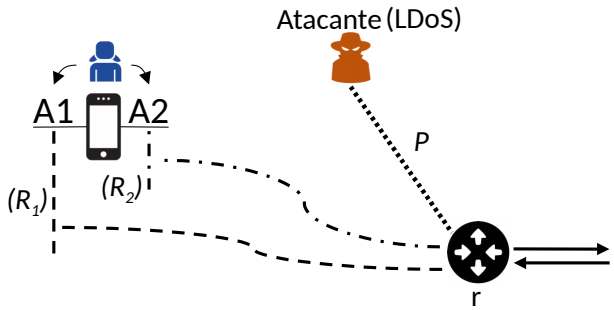


Figura 3. Ataque LDoS sobre subfluxos MPTCP

Para analisar o impacto do ataque LDoS sobre conexões MPTCP com subfluxos compartilhando um gargalo, foram realizados experimentos através do emulador `Mininet` [Antonenko and Smelyanskiy 2013] no cenário ilustrado na Figura 4. Neste cenário, o nó de **Alice** encaminha dados para o nó **Bob**. Ambos possuem duas interfaces de rede $(1, 2)$ e utilizam os respectivos caminhos (subfluxos) $C_{1,1}$ (A1) e $C_{2,2}$ (A2) para transferir uma carga fixa de 2 *Mbytes* de dados. O subfluxo $A1$ atravessa os roteadores

$r1, r3, r4, r5$; e o subfluxo $A2$, os roteadores $r2, r3, r4, r6$, sendo $r3, r4$ um gargalo compartilhado respectivamente por ambos subfluxos. Além disso, o atacante **Mallory** executa um ataque LDoS com pacotes UDP sobre o gargalo. O ataque é avaliado sobre diferentes tempos (períodos), sendo $P = \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5\}$. Em todos os períodos, o ataque possui uma duração D de $150ms$ e uma Taxa T de $1.5Mb/s$.

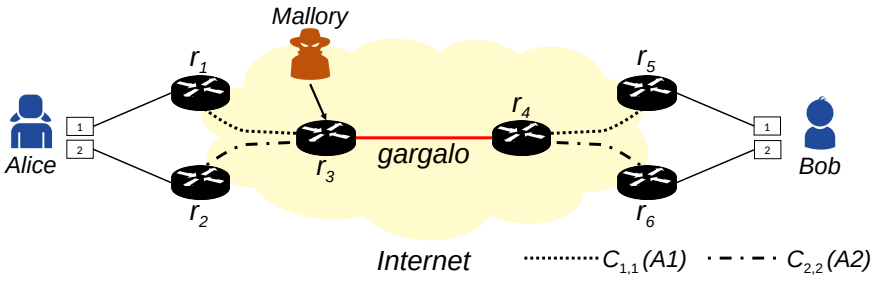


Figura 4. Cenário de avaliação

Adicionalmente, são definidos os parâmetros listados na Tabela 1, como o tamanho da fila, a largura de banda e a latência. Os valores utilizados no cenário são definidos conforme o trabalho [Kuzmanovic and Knightly 2003]. O algoritmo de controle de congestionamento é o LIA, definido como padrão para o MPTCP [Raiciu et al. 2011].

Tabela 1. Tabela de parâmetros

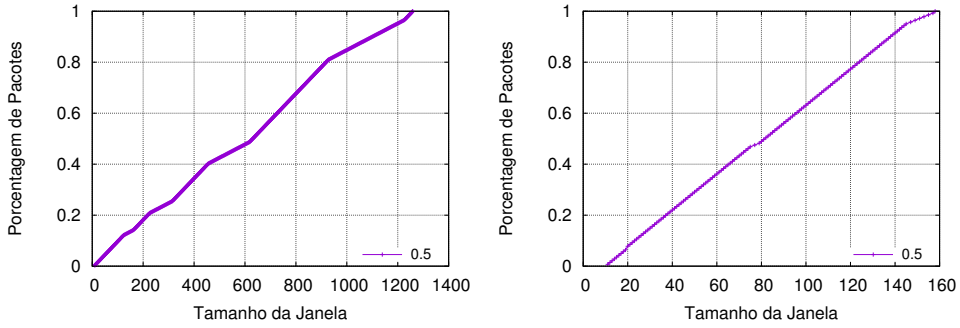
Parâmetros	$r_1 \Longleftrightarrow r_3$	$r_2 \Longleftrightarrow r_3$	$r_3 \Longleftrightarrow r_4$	$r_5 \Longleftrightarrow r_4$	$r_6 \Longleftrightarrow r_4$
fila (# pct)	100	100	1000	100	100
largura banda (Mbit/s)	100	100	1.5	100	100
latência (ms)	2	2	40	2	2

4.1. Resultados

Para verificar o comportamento dos subfluxos no gargalo compartilhado, os experimentos foram executados com os tempos descritos em [Kuzmanovic and Knightly 2003]. A Figura 5 mostra o comportamento dos subfluxos sem a execução do ataque e a Figura 6 o comportamento dos subfluxos sob o ataque. Na Figura 7, o efeito do ataque nos tempos 0.5 e 1.0. Este último considera apenas os períodos mais prejudiciais do ataque LDoS, conforme abordado por [Kuzmanovic and Knightly 2003].

A Figura 5 apresenta a conexão sem ataque. Há um crescimento contínuo na janela de congestionamento em ambos os caminhos e consequentemente os dados são transmitidos em menos tempo. Este comportamento já era esperado, pois não há competição entre os pacotes no gargalo. Também não há motivo para o algoritmo de controle de congestionamento mover os dados de um subfluxo para o outro, por esta razão o subfluxo $A1$, onde tem início a transferência dos dados, alcança um tamanho de janela superior em relação ao subfluxo $A2$. E por não haver diferenças no comportamento da janela entre os períodos, o gráfico da Figura 5 apresenta a transmissão de dados em apenas um período.

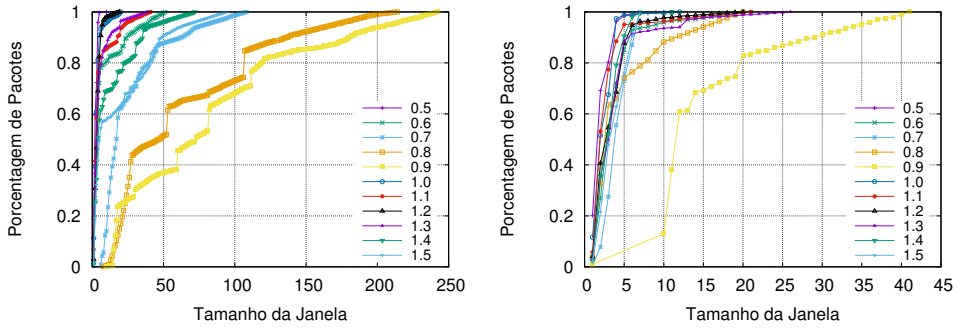
Na Figura 6, é mostrado o efeito do ataque na janela de cada subfluxo. Os períodos de tempo representam os diferentes intervalos entre as rajadas de ataque. Quanto mais próximo do eixo Y mais danoso foi o ataque naquele momento, pois a janela cresce com



a) Subfluxo A1 *b) Subfluxo A2*

Figura 5. Comportamento dos subfluxos sem ataque

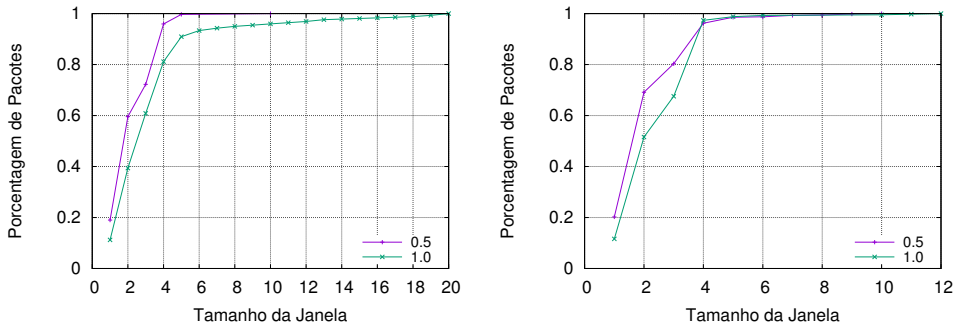
menor frequência. A Figura 6 também demonstra que o efeito do ataque ocorre de forma semelhante nas janelas de ambos os caminhos, visto que possuem o mesmo tempo de espera para a retransmissão. Quando os caminhos tentam retransmitir os pacotes perdidos, no mesmo intervalo de tempo, são confrontados com outra interrupção do ataque que induz a mais perdas de pacotes e consequentemente a um novo período de RTO.



a) Subfluxo A1 *b) Subfluxo A2*

Figura 6. Comportamento dos subfluxos sob ataque

A Figura 7 apresenta o comportamento da janela em dois momentos específicos. Como demonstrado em [Kuzmanovic and Knightly 2003], os tempos 0.5 e 1.0 são os intervalos entre rajadas de ataques mais danosos para o fluxo TCP. A conexão MPTCP



a) Subfluxo A1 *b) Subfluxo A2*

Figura 7. Comportamento dos subfluxos nos tempos 0.5 e 1.0 sob ataque

também foi bastante prejudicada nestes períodos com a janela de congestionamento tendo grande dificuldade de se expandir em ambos os subfluxos. O tamanho final da janela também ficou limitado a menos da metade do tamanho alcançado em períodos onde o ataque surtiu menor efeito, como pode ser visto na Figura 6.

5. Mitigação do Ataque LDoS: Subfluxos MPTCP com RTOs Aleatórios

A proposta de utilizar um valor aleatório para o `minRTO` do MPTCP tem como objetivo prevenir que ambos os subfluxos de uma mesma conexão sejam impactados pelo ataque com uma mesma intensidade. Com isto se espera mitigar os efeitos do ataque em transmissões com o MPTCP. Neste sentido, a estratégia empregada define para cada subfluxo um valor aleatório e distinto para R_1 e R_2, a fim evitar retransmissões em ambos os caminhos que possam coincidir com o período P do ataque. Inspirada em [Kuzmanovic and Knightly 2003], esta proposta é implementada e avaliada no MPTCP. A solução visa beneficiar os subfluxos que usem um valor de *minRTO* que não coincida com P. Isto significa que nem todos os subfluxos serão beneficiados. Aqueles que escolherem um *minRTO* muito próximo do período P de ataque serão prejudicados.

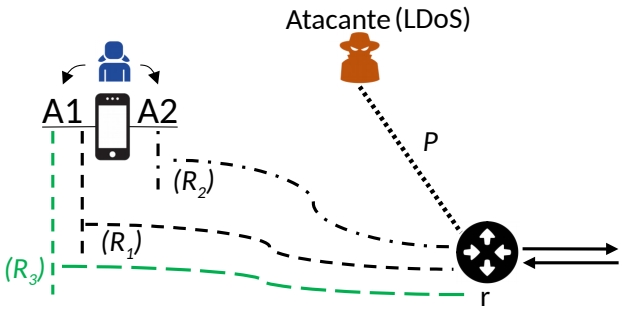
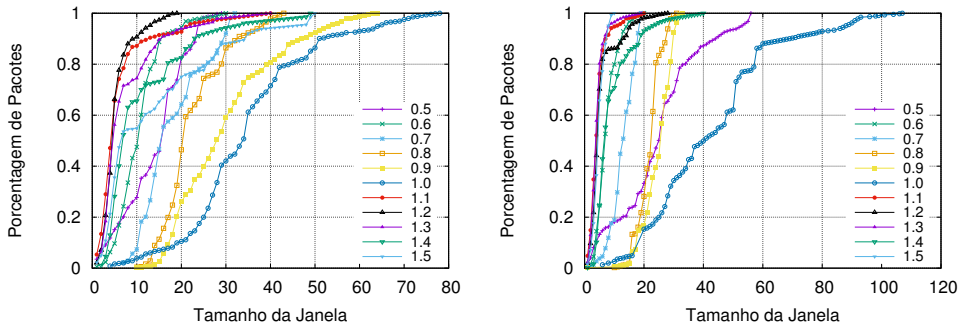


Figura 8. Subfluxos dinâmicos com `minRTO` aleatórios

O MPTCP pode beneficiar cada caminho individualmente, já que permite criar múltiplos subfluxos entre origem e destino, por exemplo, alterando somente as portas lógicas. Isto permite definir um `minRTO` diferente para cada subfluxo e ainda, durante a transmissão, remover e adicionar subfluxos a fim de minimizar o impacto do ataque LDoS. Para ilustrar, dado o cenário da Figura 8, o MPTCP inicia uma conexão definindo diferentes valores para R_1 e R_2, referentes aos subfluxos A_1 e A_2, respectivamente. Caso ocorra um estouro de temporizador (RTO) para o subfluxo A_1, por exemplo, o MPTCP cria um novo subfluxo, alterando as portas lógicas, com um novo valor para $R_3 \neq R_1$.

5.1. Resultados

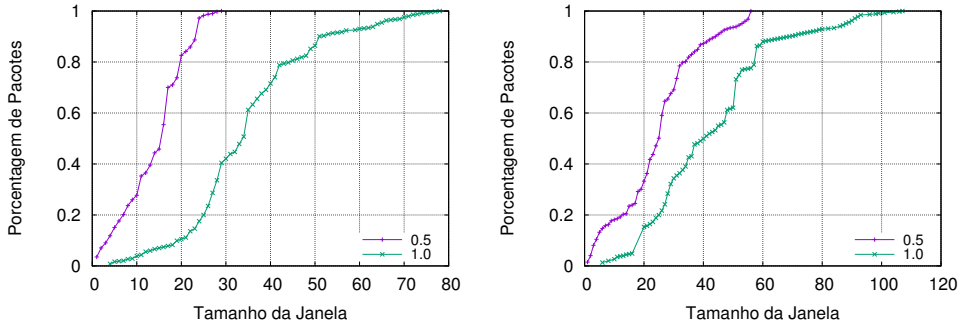
Segundo [Kuzmanovic and Knightly 2003], o sucesso do ataque LDoS está diretamente relacionado com a aproximação entre o tempo do ataque e o RTO do fluxo TCP. Isto faz com que o fluxo continue a perder pacotes e falhe ao sair do estado de *timeout*. Nesta situação os subfluxos MPTCP se comportam de forma semelhante ao TCP. O ataque LDoS afetou significativamente a transferência de dados nos tempos 0.5 e 1.0. Porém, a habilidade de dividir sua conexão em múltiplos subfluxos garante uma vantagem ao MPTCP, porque isso o possibilita variar os valores RTO para cada subfluxo. Assim mesmo que o ataque afete um dos caminhos um outro caminho com valor diferente para o RTO pode passar sem sofrer grandes danos.



a) Subfluxo $A1$ *b) Subfluxo $A2$*

Figura 9. Comportamento dos subfluxos sob ataque e RTOs diferentes entre si

A Figura 9 apresenta o comportamento da janela de cada subfluxo configurado com RTO aleatório. No momento de captura destes dados os RTOs estavam com os valores de 1198 para o subfluxo $A1$ e 1080 para o subfluxo $A2$. A aleatorização dos RTOs busca garantir, não somente diferentes valores em cada caminho, como também fazer com que os tempos de ataque coincidam com os RTOs da menor quantidade de caminhos possíveis. Sendo assim, os tempos 0.5 e 1.0, os quais prejudicaram significativamente as janelas de congestionamento com o RTO de 1000 milissegundos, não obtiveram os mesmos resultados com os valores de RTOs modificados.



a) Subfluxo $A1$ *b) Subfluxo $A2$*

Figura 10. Subfluxos nos tempos 0.5 e 1.0 sob ataque e RTOs diferentes entre si

Com as variações nos valores dos RTOs em cada subfluxo, é possível observar na Figura 9 as alterações no comportamento das janelas em cada momento. As alterações mais perceptíveis são uma melhora na expansão da janela nos tempos 0.5 e 1.0 e o ataque afetando mais intensamente os tempos 0.8 e 0.9. Conforme o valor do RTO de um fluxo é alterado alguns períodos de ataque podem ser mitigados, enquanto outros podem ser beneficiados, mas esta é uma particularidade do experimento que testou ataques em tempos diferentes. No mundo real o ataque procuraria se sincronizar com o RTO configurado para rede. A Figura 10 apresenta o comportamento dos subfluxos nos períodos 0.5 e 1.0 com valores de RTO diferentes em cada caminho. O intuito é demostrar a diferença de pontos de crescimento da janela de congestionamento em relação à Figura 7, em que os tempos de ataque tinham uma maior possibilidade de coincidir com o RTO do caminho.

6. Conclusão

Este trabalho apresentou uma proposta de mitigação do ataque LDoS, bem como uma análise do impacto causado por este ataque no protocolo TCP multicaminhos (MPTCP). Os resultados obtidos demonstraram que as rajadas periódicas do ataque LDoS degrada significativamente o MPTCP. Quando o ataque se direciona a um gargalo compartilhado pelos subfluxos, pode causar tanto dano quanto causaria em uma conexão TCP padrão. A técnica de mitigação empregada, que cria subfluxos com RTO aleatórios, reduziu o impacto do ataque. Isto ocorre porque alguns subfluxos conseguem evadir do ataque devido ao seu RTO não coincidir com o período (P) do ataque. Os resultados mostraram que os tempos de ataque de 0.5 e 1.0 foram os mais prejudiciais para o MPTCP (sem aleatorização) configurados com o RTO de $1s$. E foram estes períodos também que a solução proposta teve um melhor desempenho em mitigar o ataque. Como trabalhos futuros serão avaliados os demais algoritmos de controle de congestionamento do MPTCP.

Referências

Antonenko, V. and Smelyanskiy, R. (2013). Global network modelling based on mininet approach. In *ACM SIGCOMM*, páginas 145–146.

Barré, S., Bonaventure, O., Raiciu, C., and Handley, M. (2011a). Experimenting with Multipath TCP. *ACM SIGCOMM*, 41(4):443–444.

Barré, S., Paasch, C., Bonaventure, O., et al. (2011b). MultiPath TCP-Guidelines for implementers. Technical report, IETF.

Bhuyan, M. H., Bhattacharyya, D., and Kalita, J. K. (2015). An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection. *Elsevier Pattern Recognition Letters*, 51:1–7.

Bonaventure, O., Handley, M., Raiciu, C., et al. (2012). An Overview of Multipath TCP. *The Usenix Magazine*, 37(5):17–23.

Braun, T., Diaz, M., Gabeiras, J. E., and Staub, T. (2008). *End-to-end quality of service over heterogeneous networks*. Springer.

Chang, C.-W., Lee, S., Lin, B., and Wang, J. (2010). The taming of the shrew: mitigating low-rate TCP-targeted attack. *IEEE Transactions on Network and Service Management*, 7(1).

Chen, Z., Pham, T. N. D., Yeo, C. K., Lee, B. S., and Lau, C. T. (2017). FRRED: Fourier robust RED algorithm to detect and mitigate LDoS attacks. In *IEEE ZINC*, páginas 13–17.

Duke, M., Braden, R., Eddy, R., and Blanton, E., Z. A. (2015). TCP Roadmap. RFC 7414, IETF.

Efstathopoulos, P. (2009). Practical study of a defense against low-rate tcp-targeted dos attack. In *IEEE ICITST*, páginas 1–6.

Felix, B., Santos, A., and Nogueira, M. (2016). Reduzindo os Efeitos do Bufferbloat sobre Multi-Caminhos em Redes Sem Fio Heterogêneas. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, páginas 718–731. SBC.

Ferlin, S., Alay, O., Dreibholz, T., Hayes, D. A., and Welzl, M. (2016). Revisiting Congestion Control for Multipath TCP with Shared Bottleneck Detection. In *IEEE INFOCOM*, páginas 1–9.

Ford, A., Raiciu, C., Handley, M., Barre, S., and Iyengar, J. (2011). Architectural Guidelines for Multipath TCP Development. RFC 6182, IETF.

Ford, A., Raiciu, C., Handley, M., and Bonaventure, O. (2013). TCP Extensions for Multipath Operation with Multiple Addresses. RFC 6824, IETF.

Khalili, R., Gast, N., Popovic, M., Upadhyay, U., and Le Boudec, J.-Y. (2012). MPTCP is Not Pareto-Optimal: Performance Issues and a Possible Solution. In *ACM CoNEXT*, páginas 1–12.

Kuzmanovic, A. and Knightly, E. W. (2003). Low-rate tcp-targeted denial of service attacks: the shrew vs. the mice and elephants. In *ACM SIGCOMM*, páginas 75–86.

Kuzmanovic, A. and Knightly, E. W. (2006). Low-rate TCP-targeted denial of service attacks and counter strategies. *IEEE/ACM TON*, 14(4):683–696.

Lima, M. N., Dos Santos, A. L., and Pujolle, G. (2009). A survey of survivability in mobile ad hoc networks. *IEEE Communications Surveys & Tutorials*, 11(1):66–77.

Luo, J. and Yang, X. (2014). The NewShrew attack: A new type of low-rate TCP-Targeted DoS attack. In *IEE ICC*, páginas 713–718.

Macedo, R., Ghamri-Doudane, Y., and Nogueira, M. (2015). Mitigating dos attacks in identity management systems through reorganizations. In *LANOMS 2015*, páginas 27–34.

Nguyen, H. D. D., Phung, C. D., Secci, S., Felix, B., and Nogueira, M. (2017). Can MPTCP secure Internet communications from man-in-the-middle attacks? In *IEEE CNSM*.

Paasch, C., Barré, S., et al. (2013). Multipath TCP in the Linux kernel. *Available from {www. multipath-tcp. org.}*.

Paxson, V., Allman, M., Chu, J., and Sargent, M. (2011). Computing TCP's Retransmission Timer. RFC 6298, IETF.

Raiciu, C., Handly, M., and Wischik, D. (2011). Coupled Congestion Control for Multipath Transport Protocols. RFC 6356, IETF.

Sharma, K. and Gupta, B. (2018). Taxonomy of Distributed Denial of Service (DDoS) Attacks and Defense Mechanisms in Present Era of Smartphone Devices. *IJESMA*, 10(2):58–74.

Shevtekar, A., Anantharam, K., and Ansari, N. (2005). Low rate TCP denial-of-service attack detection at edge routers. *IEEE Communications Letters*, 9(4):363–365.

Singh, A., Xiang, M., Konsgen, A., Goerg, C., and Zaki, Y. (2013). Enhancing Fairness And Congestion Control In Multipath TCP. In *IEEE WMNC*, páginas 1–8.

Wu, Z., Wang, C., and Zeng, H. (2011). Research on the comparison of Flood DDoS and Low-rate DDoS. In *IEE ICMT*, páginas 5503–5506.

Xiang, Y., Li, K., and Zhou, W. (2011). Low-rate DDoS attacks detection and traceback by using new information metrics. *IEEE Transactions on Information Forensics and Security*, 6(2):426–437.