Impact of Low-Rate TCP-Targeted DoS Attacks on BGP

Ying Zhang
University of Michigan
wingying@umich.edu

Z. Morley Mao
University of Michigan
zmao@umich.edu

Jia Wang
AT&T Labs–Research
jiawang@research.att.com

***Abstract*— Compared to attacks against end hosts, denial of service (DoS) attacks against the Internet infrastructure such as those targeted at routers can be more devastating due to their global impact. We discover that the recently identified low-rate TCP-targeted DoS attacks can have severe impact on BGP. As the interdomain routing protocol on today's Internet, BGP is the critical infrastructure for exchanging reachability information across the global Internet. We demonstrate empirically that BGP routing sessions on the current commercial routers are susceptible to such low-rate attacks launched remotely, leading to session resets and delayed routing convergence, seriously impacting routing stability and network reachability. This is a result of a fundamental weakness with today's deployed routing protocols: there is often no protection in the form of guaranteed bandwidth for routing traffic. Combining analytical models, testbed, and Internet experiments, we thoroughly analyze the effect of such attacks on BGP. We demonstrate the feasibility of launching the attack in a coordinated fashion from wide-area hosts with arbitrarily low-rate individual attack flows, further raising the difficulty of detection. We explore defense solutions by protecting routing traffic using existing router support. Our findings highlight the importance of protecting the Internet infrastructure, in particular control plane packets.**

I. Introduction

There is evidence of increasing occurrences of Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks on the Internet today [1]. Most of the widely known attacks target a single host or multiple hosts within a particular edge network, rather than the Internet infrastructure such as routers inside transit ISP networks. The latter type of attack can be quite devastating. For example, attacks against routers can impact significant amount of traffic, as many networks rely on them to reach other destinations. Thus, it is important to understand attacks against the Internet infrastructure given its critical importance to the well-being of the Internet. In this paper, we focus on examining a particular type of attack against the interdomain routing protocol – the Border Gateway Protocol [2].

The Border Gateway Protocol (BGP), the de facto standard Internet interdomain routing protocol, uses TCP as its transport protocol. A fundamental flaw with routing protocols deployed today is that there is usually no protection in the form of guaranteed bandwidth for routing packets. Thus, congestion of other traffic can adversely impact BGP packets. Recent studies [3], [4] have indicated that congestion can severely impact routing sessions. Thus, any attack that exploits this lack of isolation with an impact on TCP can negatively affect the functioning of BGP.

In this work, we study how the recently identified low-rate TCP-targeted DoS attacks [5] affect BGP. This is the first study that systematically examines the impact of this type of attack on BGP, and we discovered the impact can be quite severe. It has been shown that low-rate TCP attacks can severely degrade TCP throughput by sending pulses of traffic leading to repeated TCP retransmission timeout. Given the fundamental susceptibility of TCP to such low-rate attacks due to its deterministic retransmission timeout mechanism, any application using TCP is vulnerable. In particular, the effect on protocols using TCP within the Internet infrastructure is arguably more severe due to the global scope of the impact. Aside from the potential impact on the throughput of BGP packets, a more critical question is whether such attacks are powerful enough to *reset BGP's routing session* as a result of a sufficiently large number of consecutive packet drops. If the session is reset, it can have serious impact on the Internet in the form of routing instability, unreachable destinations, and traffic performance degradation. Note that one can launch such attacks remotely from end hosts without access to routers nor the ability to send traffic directly to them. Its low-rate nature makes detection inherently difficult. More importantly, the existing best common practice for protecting the Internet routing infrastructure by disallowing access and research proposals such as SBGP [6] are not sufficient to prevent this type of low-rate attack.

In this paper, we study how low-rate TCP targeted DoS attacks can cause session reset and throughput degradation. *We show empirically using testbed experiments that today's routers with default configurations are susceptible to BGP session resets as a result of low-rate TCP-targeted DoS attacks.* We observe that attackers can bring down the targeted BGP session within as little time as 216 seconds. Session reset probability can be as high as 30% with only 42% utilization of the bottleneck link capacity. And when the session is not reset, BGP table transfer duration can be increased from 85 seconds up to an hour with only 27% of the link capacity used.

To fully understand how BGP is vulnerable to such attacks, we also use analytical models to study the probability of session reset and the degree of throughput degradation due to the attack. Using wide-area experiments, we show the ease with which coordinated low-rate attacks can be

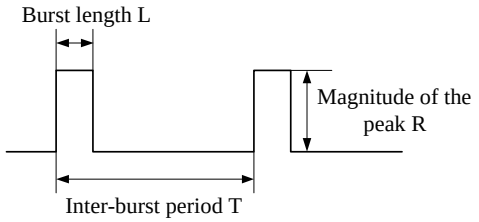


Fig. 1. Notation for low-rate TCP-targeted DoS attacks

launched, resulting in arbitrarily low-rate individual attack flows. This raises the difficulty of attack detection. For attack defense, we suggest several prevention techniques to make such attacks impossible and provide recommendations on configurations supported by routers today.

The rest of the paper is organized as follows. We provide the background by describing the low-rate TCP-targeted DoS attacks and introducing basics about BGP in Section II. Section III discusses impact of low-rate attacks on BGP and key factors in determining vulnerability of BGP. We show using testbed experiments that BGP can be affected by low-rate TCP attacks in Section IV and model the impact of low-rate attacks on BGP in the form of throughput degradation and session reset in Section V. Section VI shows how to launch coordinated low-rate attacks by synchronizing flows across multiple attack hosts. We evaluate defense mechanisms in Section VII and conclude in Section VIII.

II. Background

In this section, we describe low-rate TCP-targeted DoS attacks and the Border Gateway Protocol which is susceptible to such attacks.

A. Low-rate TCP-targeted DoS Attacks

In their seminal work, Kuzmanovic and Knightly showed [5] that TCP's retransmission timeout mechanism can be exploited by maliciously chosen low-rate DoS traffic to throttle TCP flows to a small fraction of their ideal rate. As shown in Figure 1, the low-rate attack consists of periodic, on-off square-wave of traffic bursts with magnitude of the peak R, burst length L, and inter-burst period T.

There are several requirements for the attack to be successful: (i) An integer multiple of the inter-burst period coincides with the minimum retransmission timeout value (minRTO) of TCP. (ii) The magnitude of the attack peak traffic is large enough to cause packet loss. (iii) The burst length is sufficiently long to induce loss. It needs to be longer than roundtrip time (RTT) of TCP flows. When these conditions are satisfied, the aggregate TCP flows sharing the bottleneck link will have close to zero throughput. Even if the inter-burst period takes on other values outside the minRTO range, the throughput can still be severely degraded. The reason is that the TCP retransmission timer repeatedly times out due to loss induced by the attack traffic burst, as the timer value exponentially increases for any given flow sharing the bottleneck link with the attack traffic.

Low-rate attacks are not easy to detect given their overall low average rate despite their high peak rate. Even if detection algorithms focus on high flow rates on short timescales, legitimate bursty TCP flows could be misclassified. One way to defend against such attacks is to randomize the minRTO value; however, this does not completely fully mitigate the attack as shown by Kuzmanovic and Knightly [5]. They also found that even router-assisted mechanisms do not eliminate the attack impact.

There has also been follow-up work on detecting low-rate attacks [7], [8], [9], [10]. Such attacks are harder to detect than traditional brute-force, or flooding style attacks because of their low rate nature. Most of the existing detection algorithms rely on signal analysis. For example, dynamic time warping [7] detects low-rate attacks by matching pattern with obtained attack signatures. Note that none of proposed detection algorithms have been shown to be sufficiently accurate and scalable for deployment in real networks. Furthermore, no known solution exists to effectively mitigate such low-rate attacks. Thus, all applications using TCP are susceptible to degraded performance due to such attacks. In this work, we focus on the Border Gateway Protocol as an important "application" using TCP given its role as the interdomain routing protocol on the Internet.

B. Border Gateway Protocol

The Border Gateway Protocol (BGP) is used as the interdomain routing protocol on today's Internet. In BGP, a routing session is established over a TCP connection between neighboring border routers to exchange reachability information. There are two types of BGP sessions: eBGP and iBGP sessions. The former are between routers within different autonomous systems (ASes) or networks, and usually consist of a single hop, *i.e.*, the two routers are directly connected with a physical link. The latter are within the same AS and can go through multiple router hops. Four types of messages are exchanged over BGP sessions: *Open* (session establishment), *Notification* (errors), *Update* (announcements and withdrawals of routes), and *KeepAlive* (session liveness).

Because BGP is a stateful protocol with no periodic refreshes of routing updates, routing information previously received is assumed to be valid until withdrawn. To ensure connection liveness, KeepAlive messages are exchanged periodically. According to BGP specification [2], each BGP router maintains a *Hold Timer* which limits the maximum amount of time that may elapse between receipt of successive KeepAlive and/or update messages from its neighbor in the BGP session. If the Hold Timer expires, a notification error message is sent and the BGP connection is closed. Upon session reset, all routes previously exchanged are assumed to be unusable and implicitly withdrawn.

There has been increasing concern over BGP's security, as no deployed mechanisms exist to verify the correctness, authenticity, integrity of the routing information exchanged using BGP. Proposed protocols such as SBGP [6], SoBGP [11]

can address some of these issues. Router vendors have also provided protection against commonly known attacks such as TCP RST and SYN flood attacks [12]. However, these proposed and deployed solutions do not address the fundamental problem of the lack of resource isolation of BGP traffic from other traffic. And unlike RST or SYN flood attacks, it is possible to remotely launch resource-based attacks using packets *passing through* the routers without the ability to send packets destined to them.

III. Low-rate DoS Attacks on BGP

Because BGP runs over TCP for reliability, BGP is also be vulnerable to the recently discovered low-rate TCP-targeted DoS attacks. Due to its low-bandwidth property, this type of attack is much more difficult to detect, and thus it is important to understand it thoroughly. In this paper, we focus on investigating the effect of low-rate attacks on a single-hop BGP session. However, the results can be generalized to multihop BGP sessions. Arguably multihop BGP sessions are more susceptible to such attacks as they traverse multiple links, thus more likely to experience congestion.

A. Impact of Attacks on BGP Sessions

The impact on BGP sessions caused by low-rate TCP-targeted DoS attacks are two fold: *throughput degradation* and *session reset*. In the less severe case, the throughput of the BGP update messages can be significantly reduced. However, the rate of BGP updates on average is quite low, except during the table transfer operation upon session establishment. Thus, the impact in the form of rate reduction of BGP traffic is less critical, but can lead to increased convergence delays.

The second type of impact is BGP session reset, which is of a much more severe nature. To reset a BGP session, the induced congestion by attack traffic needs to last sufficiently long to cause the Hold Timer to expire. To monitor the attack success, one can analyze traffic going through the impacted link or the withdrawal of routing updates related to the session. Furthermore, once a BGP session is reset, it is easier to keep the session down as SYN packets are sent less frequently compared to retransmitted data packets.

BGP session reset can lead to severe churn on the Internet's control plane. This not only impacts both routers involved in the BGP session, as each withdraws all the routes previously advertised by its neighbor, but also many other networks on the Internet due to the propagation of routing changes. For example, the number of routes in a default-free router in the core Internet is around 170,000 based on routing data from RouteViews [13]. A significant fraction of the table can be affected upon a BGP session reset. Withdrawing a large number of routes can cause many destination networks to become temporarily unreachable due to inconsistent routing state [14] and a large amount of traffic to become rerouted, which may further lead to congestion due to insufficient capacity.

A recent proposal to mitigate the potential negative impact of short-lived session resets is termed graceful restart [15]. Routers supporting this mechanism will attempt to continue to forward packets using the stale routes. There is, however, an upper bound (by default two or three minutes) on the amount of time a router retains the stale routes to limit the duration of routing inconsistency. Thus, a session reset that lasts sufficiently long time, possibly due to an intense low-rate attack, can still have severe impact on the data plane.

In general, the impact of an eBGP session reset is larger than that of an iBGP session reset because in eBGP sessions routing changes are more likely to propagate across multiple networks and the routing table is usually default-free, thus carrying all the destinations to the Internet. The routes exchanged between two routers in an eBGP session consist of all the routes of their respective customers. Thus, for eBGP sessions between two large ISPs, this number can be quite large. We analyzed routing tables from a tier-1 ISP and found that up to 13% of the routing table can come from a single eBGP session versus only 4% from an iBGP session.

B. Key Factors in Attacking a BGP Session

We study the key factors that determines the vulnerability of BGP to such attacks to illuminate identify solutions.

1. Priority of routing traffic. The fundamental problem that makes BGP vulnerable to low-rate attacks is that router traffic may not be sufficiently protected from congestion caused by other data traffic. Many of the commercial routers today by default use First-In-First-Out (FIFO) or Drop Tail queueing discipline, giving no priority to routing packets. Even in the case where routing data are protected (*e.g.*, through the RED queue management scheme [16]), there are no default policing mechanisms to prevent attack packets from spoofing packets of higher priority. For example, we observed that many routers will mark the routing packets with an IP precedence value of 6 [17]. However, attack packets can also use the same or even higher IP precedence values given the lack of authentication for such values by default. Packet remarking needs to be configured for the protection to be effective. In this work, we illuminate these issues by experimenting with real commercial routers with various configuration settings. Instead of using simulations, we focus on using experiments to validate our conjectures given there are no accurate simulations of BGP on commercial routers.

2. Proprietary router implementation. Router behavior is much less understood compared to that of end hosts due to its proprietary nature and lack of source code access. For example, it is unclear how the TCP stack on commercial routers really behaves. Unlike for end-hosts, critical parameters to the attack such as minRTO are unknown, making successful attacks much more difficult. If minRTO is randomized, it would further reduce the probability of a session reset. Even with known router behavior, depending on its configuration, its dynamic behavior may be quite different compared to the default settings. We mainly focus

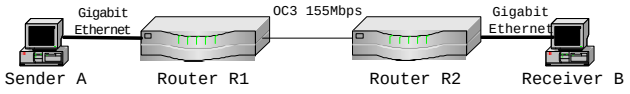


Fig. 2. Lab experiment testbed

on the default settings as most routers are probably set up with such configurations. When we know that the router supports certain features that would help protect against the low-rate attacks, we also examine these features in great detail.

3. Capacity of peering links. In order for low-rate TCP attacks to be successful against BGP routing sessions, the traffic burst needs to be sufficiently powerful to cause enough packet loss, so that the TCP flow of the BGP session enters into retransmission timeout state. This may appear to be difficult to achieve, especially for BGP session involving Internet core backbone links given the heavily overprovisioned core. However, eBGP sessions involve peering links which may not be as well-provisioned compared to links within an ISP backbone. There has been anecdotal evidence that congestion often occurs on peering links. Previous measurement studies such as [18], [19] have shown that some of the bottleneck links of today's Internet paths occur at the boundary between two networks. Links between stub networks and their providers often have much lower speed and these networks often use eBGP to obtain routes. Using data from RouteViews [13], we found 23% of 100,482 eBGP peering sessions belong to the stub networks. Furthermore, it is not necessary that a single attack machine overwhelms the target link. As we show later in Section VI, multiple machines can be used to launch a coordinated attack, as long as they traverse the link involved in the BGP session under attack. In this work, we investigate the necessary conditions and show experimentally how this can be achieved.

IV. Testbed Experiments

In this section, we describe experiments conducted on a local testbed and empirically show that commercial routers can be severely impacted by low-rate TCP-targeted DoS attacks in the form of session resets and degraded table transfer throughput. We first present our experiment setup, and then inferred TCP characteristics and observed BGP parameters of different commercial routers, followed by detailed analysis of the attack impact. Finally, we explain our observations based on the router architecture.

A. Testbed Setup

Our experiment testbed consists of two commercial routers and two PCs shown in Figure 2. The two links connecting the routers and the PCs are full-duplex Gigabit Ethernet. The link between the routers is Packet Over SONET (POS) with 155 Mbps link capacity, serving as the bottleneck link. Note that our experiment testbed closely models the real operational scenario of an eBGP session with two differences. First, we do not model background traffic, effectively making attacks more difficult. The link types are selected to allow traffic from Sender A to Receiver B to overload the link between the two routers. Second, in practice attack hosts are usually several IP hops away from the target link, with more variable and longer delays to the target link. Longer delays do not affect attack effectiveness, but more variable delays can make attacks more difficult to control. We describe later in Section VI how such difficulties can be overcome using coordinated attacks.

The experiment is conducted as follows. A sender program transmits from Sender A UDP-based low-rate attack traffic[1] of the shape shown in Figure 1 with peak rate at 185 Mbps traversing the link between the two routers arriving at Receiver B. The peak rate is set to be 185 Mbps, as it is the lowest rate needed to successfully reset the session with a burst length of 150 ms. With shorter burst length, we observe that the session does not reset even with larger peak rate due to insufficient time to saturate the router buffer to cause congestion.

When the bottleneck link between the two routers becomes congested, we observe attack packets are dropped at both the input and output queue at router R_1. Using default router configurations, locally generated BGP packets from R_1 to R_2 also experience packet loss due to shared router buffer. If one of R_1's BGP packet and its subsequently retransmitted packets are all lost causing the Hold Timer to expire, the BGP session is closed. We will show in Section V the derivation of necessary conditions for session reset.

We experimented with a wide variety of commercial routers using the latest router OS whenever possible from the Schooner testbed [20] and our own lab. They consist of the following types: Cisco 3600, 7200, 7300, 12000 (commonly known as GSRs), and Juniper M10. To study the extent of the phenomenon, the same experiments were performed on all these routers, and similar results were observed. One main difference is that lower-end routers such as Cisco 3600 have smaller buffer compared to more powerful routers such as Cisco GSRs, making them more vulnerable to attacks. Another difference is that the Juniper M10 is found to be more vulnerable due to its larger minRTO and smaller KeepAlive and Hold Timer values. This is explained in more details later in Section V. We emphasize that susceptibility to low-rate DoS attacks is *a general problem with any router* when not configured with ways to prioritize BGP traffic and has a BGP implementation using TCP with a deterministic retransmission timeout mechanism.

In this paper, we use Cisco GSRs with IOS version 12.0 to illustrate our results because they are the most powerful routers we have access to and are commonly used in Internet backbone networks. In particular, the Cisco GSRs used are equipped with Cisco 12410/GRP (R5000 CPU at 200 Mhz) processor, 512 KB L2 cache, and 512 MB memory. The line card on the router is 4 port POS OC-3c/STM-1 Multi

[1]UDP is used as opposed to TCP to precisely control the sending rate. TCP packets, without conforming to congestion control can also be used to avoid detection.

Mode with Engine type 0, a buffer size of 12560 packets for packet sizes matching that of BGP packets.

B. Router Implementation Diversity

Before examining the impact of low-rate attacks on BGP, we first analyze the TCP behavior and default router configuration settings. This is crucial for understanding why commercial routers are vulnerable to low-rate attacks. Similar to end host operating system where a number of variations of TCP protocols have been proposed and deployed [21], router's TCP implementation can also have variations due to differences such as vendors, router series, and RouterOS versions. In our work, TCP related parameters are obtained using software we developed based on the TBIT (TCP Behavior Inference Tool) [21], which infers TCP properties on Web servers. We enhanced it by integrating BGP-related functionality to establish a BGP session with a commercial router. After the session establishment, the tool constructs packets in special ways to infer router's TCP behavior. The most important TCP property inferred is minRTO which can be accurately determined. Besides differences in TCP implementation, routers from different vendors have dissimilar default BGP parameter settings controlling the sending of BGP updates, KeepAlive messages, and session timeout behavior. In this paper, we focus on the default configuration given it is probably the most widely used.[2]

Table I shows two important TCP parameters in determining the attack effectiveness on BGP. The minRTO of Cisco routers is either 300 ms or 600 ms, and that of Juniper M10 is 1 second. Previous work [5] shows that with larger minRTO, the attack inter-burst period can also be larger, as its integer multiple needs to match the minRTO for attacks to be most effective. For a given burst length, a longer inter-burst period reduces the attack rate, increasing attack detection difficulties. Consequently, routers with small minRTO are more difficult to attack without being detected. If the minRTO is sufficiently small, it becomes impossible to precisely control the timing to launch low-rate attacks. However, small minRTOs can cause unnecessary retransmission timeout.

Another important TCP feature that can have impact on how easy it is to keep the BGP session down is the schedule of retransmitting SYN packets upon a loss. Consider Cisco GSR as an example, when the first SYN packet is lost during connection establishment, it will resend it after 2 seconds. If the retransmitted SYN is also lost, the router will resend SYNs successively after 4 seconds, 8 seconds, and 16 seconds until successful transmission. After four retransmission failures, the router will terminate the TCP connection. The router will then wait for 152 seconds before opening a new TCP connection. Thus in order to keep the session down, the attack flow with inter-burst period of 2 seconds is sufficient to drop each SYN packet, which makes the attack flow lower rate and more difficult to detect.

[2]Other configurations except certain protections described in Section VII do not help mitigate the attacks.

Table I also shows the default router BGP configurations for several features relevant to the low-rate attack. Cisco routers use a 60 second KeepAlive Timer and a 180 second Hold Timer by default, while Junipers have smaller default timer values: 30 seconds for KeepAlive and 90 seconds for Hold Timer. We derive in Section V to reset the BGP session, the attacker needs to cause at least 8 consecutive packets to be dropped for Cisco GSR and only 6 for Juniper M10 due to the timer values. Thus, Juniper M10 is more vulnerable to low-rate attacks compared to Cisco GSR. The default queuing algorithm is FIFO instead of RED. RED can help protect routing packets. Graceful restart support provided by Cisco [22] is not enabled by default, but it can help routers tolerate short-lived session down time. There is however a timer limit on the down time before the stale routes are withdrawn, with a default of 2 or 3 minutes.

C. Experiment Results

As described in [5], three key factors determine the attack impact: *burst length*, *inter-burst period*, and *peak magnitude*. Intuitively, longer burst length causes the bottleneck queue to be full for longer duration, leading to larger attack impact. Shorter inter-burst period results in larger probability of dropping BGP packets. The larger the attack peak magnitude, the sooner packets will fill up the router's queue on the bottleneck link. However, attacks with sufficient average rate would no longer be considered low-rate. We discuss how distributed low-rate attacks can be composed in Section VI.

Next, we analyze attack impact to confirm these intuitions and focus on the attack impact on BGP. We use three metrics to measure the BGP performance under attack: session reset probability, time to reset the session, and BGP table transfer duration. In particular, we conduct four sets of experiments. The results reported in this section are obtained by repeating each individual experiment 100 times.

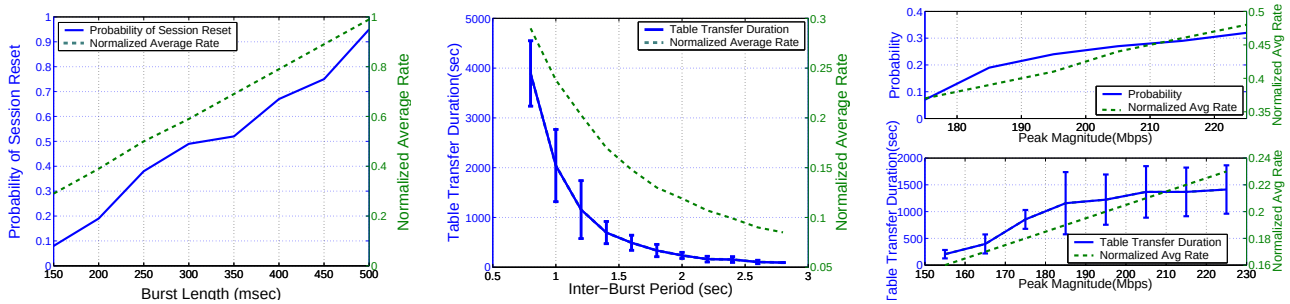
1. Impact of attack burst length on session reset. Attack flows can force the BGP session to reset when attack inter-burst period (or one of its integer multiples) is equal or close to the minRTO value of the TCP implementation on the router. The BGP session will be reset after dropping k consecutively retransmitted packets, where k's value depends on KeepAlive and Hold Timer values configured on the router. In this experiment, we analyze how attack burst length impacts the probability of session reset. The experiment is set up as follows. Router R_1 periodically sends KeepAlive messages to Router R_2. Sender A starts a low-rate attack with traffic destined to Receiver B with a given burst length, a fixed inter-burst period of 600 ms and peak magnitude of 185 Mbps. Figure 3(a) shows as expected that the session reset probability and the average rate of attack flow increase with larger burst length. When the burst length is half of the inter-burst period, the attacker has about 50% chance to reset the session.

2. Impact of attack inter-burst period on BGP table transfer. As observed above, an attack might not reset the session. When the session is not reset, attack flows can

Router type	RouterOS version	TCP properties		BGP/Router parameters (default)			
		minRTO (msec)	SYN retry pattern (sec)	KeepAlive (sec)	Hold Timer (sec)	Queue alg.	Graceful restart timer (sec), range
Cisco 3600	IOS 12.2(25a)	300	2,4,8,16,(2).	60	180	FIFO	Not supported
Cisco 7200	IOS 12.2(28)S3	600	2,4,8,16,(152).	60	180	FIFO	Supported, 120, 1-3600
Cisco 7300	IOS 12.3(3b)	300	2,4,8,16,(152).	60	180	FIFO	Supported, 120, 1-3600
Cisco 12000	IOS 12.0(23)S	600	2,4,8,16,(152).	60	180	FIFO	Supported, 120, 1-3600
Juniper M10	JUNOS[6.0R1.3]	1000	3,6,12,24,(30).	30	90	FIFO	Supported, 180, 1-3600

TABLE I

ROUTER TCP BEHAVIOR, ROUTER AND BGP PARAMETERS.



(a) Attack burst length (R=185 Mbps, T=600 ms)

(b) Inter-burst period (R=185 Mbps, L=200 ms)

(c) Peak magnitude (Top: T=600 ms, L=200 ms) (Bottom: T=1.2 second, L=200 ms)

Fig. 3. Impact of attack traffi c on BGP session reset and table transfer duration: minRTO=600 ms, with Cisco GSRs.

delay updates due to increased queueing and packet loss, resulting in longer BGP convergence delays. We use the BGP table transfer duration as a measure to study the impact of varying inter-burst period of low-rate attacks. In this experiment, we fix the peak magnitude at 185 Mbps and burst length at 200 ms. The smallest inter-burst period is chosen as 800 ms second given minRTO of 600 ms to prevent session reset. We conduct the experiment as follows. First, we load R_1 with a randomly chosen BGP table of 166,527 routing entries obtained from RouteViews [13]. Then, the BGP session between R_1 and R_2 is configured and established. Subsequently within at most 3 seconds of session establishment, Sender A starts the low-rate attack with traffic destined to Receiver B. We record the time when R_2 receives the entire table. Figure 3(b) illustrates the average and standard deviation of BGP table transfer durations with varying inter-burst period. For inter-burst period of 0.8 second and less than 30% average utilization of the link capacity, it takes on average more than one hour to finish transferring the BGP table, which normally lasts only about 85 seconds! As the inter-burst length increases to 1.2 seconds, BGP table transfer still needs 21 minutes to finish on average. However, the impact on BGP table transfer diminishes quickly with increasing inter-burst period.

3. Impact of attack peak magnitude on session reset and table transfer. The required time for attack flows to fill up router's queues depends partly on the attack peak magnitude. To evaluate the impact of peak magnitude on session reset probability, we fix the burst length at 200 ms and the inter-burst period at 600 ms matching the minRTO value. Cisco GSR allocates buffer space in chunks of varying sizes, matching packets of different sizes. We use a fixed attack packet size of 30 bytes, of similar size to the KeepAlive packets, so that they will be placed in the same buffers. We vary the peak magnitude from 175 Mbps to 225 Mbps by changing the sending rate. 175 Mbps is chosen as it is the lowest rate needed to reset the session for our setup. As shown in the top plot of Figure 3(c), the session reset probability increases gradually with larger attack rate. Based on simple calculations, increasing peak magnitude from 175 Mbps to 215 Mbps (or the excess bandwidth relative to the bottleneck link from 20 Mbps to 60 Mbps) reduces the time to fill up the 3 Mbit queue[3] from 150 ms to 50 ms. This effect is equivalent to increasing the burst length by 100 ms, shown in Figure 3(a).

We next analyze the impact of peak magnitude on BGP table transfer duration by fixing the burst length at 200 ms and the inter-burst period at 1.2 seconds, intentionally chosen to be different from the minRTO value to prevent session resets. Attack UDP packet size is set to 1500 bytes, matching the size of BGP packet for table transfer. The attack peak rate varies from 155 Mbps to 225 Mbps. The bottom plot of Figure 3(c) shows as expected that with increasing peak rate, BGP table transfer duration gradually increases.

4. Impact of BGP update behavior on session reset. The BGP session is brought down as soon as the Hold Timer expires. This requires losing all the BGP packets for a duration at least as long as the Hold Timer value. To trigger the first set of packet loss, a BGP packet must encounter congestion possibly induced by the attack traffic. To cause retransmission timeout, the attack flow needs to cause all the packets within one TCP window to be dropped to avoid TCP fast retransmission. If BGP packets are exchanged

[3]3 Mbit is derived from a buffer of 12560 packets with 30 byte packets.

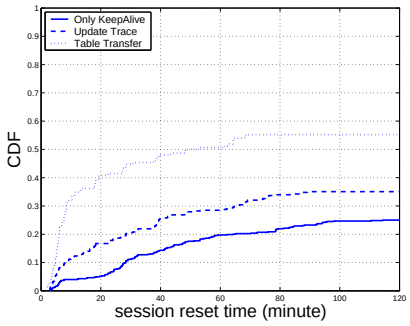


Fig. 4. Attack duration to cause session reset with Cisco GSRs (T=600 ms, L=200 ms, R=185 Mbps, minRTO=600 ms).

infrequently and happen to always miss congestion, it will be impossible to reset the session. Thus, the more frequently BGP messages are exchanged, the more likely the BGP session is reset. In all the above experiments, we focus on the overly conservative scenario where only KeepAlive messages are exchanged. In reality, routers frequently send updates associated with routing changes. Thus, "busier" routers, routers with larger BGP tables, containing more unstable routes, are more likely impacted.

In order to measure the attack impact under different conditions, we examine the attack duration needed for session reset in the following three scenarios with increasing BGP message frequency. (i) Only KeepAlive messages are sent (every 60 seconds). (ii) One typical day's BGP update trace from RouteViews is played back. (iii) One default-free table from RouteViews is transferred. The second scenario is the common case. Figure 4 shows the distribution of the attack duration needed for session reset (cut off at 2 hour time limit). As expected, on average it takes the least time to reset the session for scenario (iii) when updates are most frequently exchanged. Scenario (i) requires on average more time compared to other scenarios. In the best case, it takes only 216 seconds to reset the session.

To summarize, the experiments described above analyze in detail how various parameters of low-rate attacks affect the attack effectiveness on BGP. We observe that increasing burst length and peak magnitude, and reducing inter-burst period increase attack effectiveness. These results confirm the danger that a low-rate attack can reset a BGP session.

D. Router Architecture: Explaining Packet Drops

The router architectures from different vendors and types vary in details; however, packet drops occur whenever any buffer becomes full with the default FIFO queuing. There are usually multiple buffers in a router, traversed by forwarded packets. BGP packets share with attack traffic and other background traffic some of these buffers. More specifically, locally generated BGP packets by the router shares with other packets the buffer space on the output interface. BGP traffic forwarded by the local router, in the case of iBGP sessions, experiences sharing in all these buffers with other traffic. *Thus, protection for BGP traffic needs to be provided at both incoming and outgoing interfaces.*

Category	Variable	Description
Traffic	T	Attack traffic inter-burst period
	L	Attack traffic burst length
	R_{DoS}	Attack traffic peak magnitude
	R_{Other}	Background traffic rate
	R_{BGP}	BGP traffic rate
	Pkt	Average packet size
	k	Number of retransmissions
Bottleneck queue	B	Queue buffer size
	B_0	Size of queue at time 0
	R_{link}	Bottleneck link capacity
Timer values	$minRTO$	Min. retransmission timeout
	$Hold\_Time$	Hold Timer
	$KeepAlive$	Interval to send KeepAlives
	$MinRAdv$	MinRouteAdvertisementInterval

TABLE II
RELEVANT PARAMETERS IN THE MODELS

In our experiments, BGP traffic is associated with a single-hop eBGP session and thus is locally generated. Some attack traffic is observed to be dropped at the input interface, and we observe that BGP packets experience loss at output queues. Under the default configurations, there is no protection for the BGP traffic which thus competes for buffer space with all other traffic. Recent proposals [23], [24] on reducing router buffer size partly to improve delays may endanger BGP packets, as transient congestion either intentional through attacks or unintended by regular traffic can result in BGP packet loss, especially with smaller buffers.

V. Analytical Models

In this section, we present analytical models of the session reset probability and the throughput degradation of BGP under low-rate DoS attacks. We analyze the impact of various attack parameters using our models, confirming our experimental results described in Section IV.

A. Probability of BGP Session Reset

We first study how likely a BGP session reset occurs due to low-rate attacks. BGP packets can be dropped either at the input or output queue due to congestion. We focus on the analysis at the bottleneck queue. In our experiment setting, it is the output queue of router R_1 in Figure 2. The session will be reset, if all the BGP packets within a TCP window and the subsequently retransmitted packets are dropped for a sufficient amount of time, causing the Hold Timer to expire. This occurs before the TCP session is closed due to TCP session timeout, as the TCP timeout value (*e.g.*, 10 minutes) is usually much larger than the Hold Timer value (*e.g.*, 180 seconds). The session reset probability depends on a number of factors, listed in Table II.

We first assume the aggregate background traffic has a constant rate (we relax this constraint later) and the BGP packets are sent uniformly random during time period T. Under low-rate DoS attacks, the total incoming traffic rate is greater than the bottleneck link capacity (*i.e.*, $R_{DoS} + R_{Other} + R_{BGP} > R_{link}$). If the attack traffic

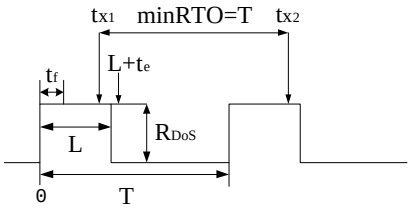


Fig. 5. Low-rate attack illustration when $i \times T = minRTO$.

arrives at time 0, it takes $t_f = \frac{B-B_0}{R_{DoS}+R_{Other}+R_{BGP}-R_{link}}$ to fill up the bottleneck queue. After the attack traffic enters the burst free period at time L, the average transmission time for one packet is $t_e = \frac{Pkt}{R_{link}-R_{Other}}$. For example, using the parameters for earlier experiments shown in Figures 3(a) and (b): B=12560 packets, B_0=0, R_{DoS}=185 Mbps, R_{link}=155 Mbps, R_{Other}=0, Pkt=30 bytes, we obtain t_f=100.5 ms, and t_e=1.54 μs.

Assuming the default FIFO queueing, when the first BGP packet arrives at the bottleneck queue at time t_{x1}, it is dropped if it encounters the attack burst, *i.e.*, $t_f \leq t_{x1} \leq L + t_e$. If t_{x1} can occur at any time during T, then the probability of the first packet drop is $P(1st\ drop) = \frac{L+t_e-t_f}{T}$. Without receiving three duplicate ACKs, the sender will retransmit the packet after $RTO = max\{minRTO, SRTT + max(G, VRTT)\}$, where G denotes the clock granularity with a typical value of less than 100 ms, and the smoothed RTT value $SRTT$ and the RTT variance $VRTT$ are updated each time a new packet is received. We now study several cases with increasing generality:

Scenario 1: $i \times T = minRTO$ ($i \in N$) and no propagation delay. In this case, we assume an integer multiple of the attack inter-burst period matches the minRTO of the target router involved in the BGP session. The first retransmitted packet arrives time $t_{x2} = t_{x1} + minRTO$. As shown in Figure 5, this packet will also be dropped because $iT + t_f \leq t_{x2} \leq iT + L + t_e$. Similarly, the second retransmitted packet arriving at time $t_{x1} + 2minRTO$ is also dropped, and so is the third retransmitted packet arriving at $t_{x1} + 4minRTO$, and so on. Thus, all the retransmitted packets are forced to synchronize with the attack traffic burst and are dropped. The BGP session is reset after the Hold Timer expires. Therefore, the probability of session reset is

$$P(reset) = \frac{L + t_e - t_f}{T} \quad (1)$$

Assume that there are k retransmissions before the BGP session is reset, where the kth retransmitted packet arrives at time $t_{x1} + (2^k - 1)minRTO$. Let $t_{interval}$ denote the time interval between the last received BGP packet and the first dropped BGP packet. This depends on how frequently the BGP messages are sent. The value of k is computed as:

$$k = \lfloor \log_2(\frac{Hold\_Time - t_{interval}}{minRTO} + 1) \rfloor \quad (2)$$

During the attack, if only KeepAlive messages are exchanged, then $t_{interval} = KeepAlive$. If updates are also sent, but subject to per peer based MinRouteAdvertisementInterval timer constraint[4], then $MinRAdv < t_{interval} \leq KeepAlive$. Without the timer constraint, $RTT < t_{interval} \leq KeepAlive$, where RTT is the roundtrip time of the TCP connection for the BGP session.

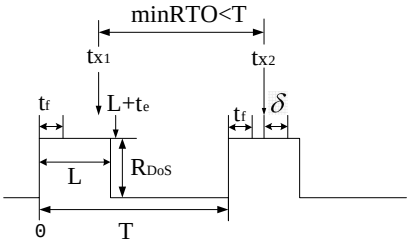


Fig. 6. Low-rate attack illustration when $i \times T = MinRTO + \delta$.

As an example, we use our model to analyze one of our experiments in Section IV where only KeepAlive messages are exchanged. The Cisco GSR has as default setting $Hold\_Timer$=180 seconds, $KeepAlive$=60 seconds. And $minRTO$ is found to be 600 ms. If the most recent BGP packet is received at time 0, then the next packet arrives at about $t_{x1} = KeepAlive$. This packet is expected to be dropped due to congestion. The 7 subsequently retransmitted packets are also dropped at time 60.6 seconds, 61.8 seconds, 64.2 seconds, ..., 136.2 seconds, respectively. In general, the kth retransmitted packet arrives at time $60 + 0.6(1 + 2 + 4 + \ldots + 2^{k-1})$. Before the 8^{th} retransmitted packet arrives at 213 seconds, the BGP session is reset at time 180 seconds.

This matches our observation in Section IV where a total of 8 consecutive packets are dropped before the BGP session is reset. According to our model, Juniper routers are more vulnerable compared to Cisco routers due to its default setting of $Hold\_Time$= 90 seconds, $KeepAlive$=30 seconds and its TCP stack's $minRTO$=1 second. Only 6 consecutive packet drops are needed for Juniper to cause BGP session reset, matching our observations.

So far, we focus on the case when only a single BGP packet is sent during the TCP window. When there are multiple BGP packets sent within the same TCP window, all the packets within that window need to be dropped for the attack to be successful otherwise TCP will enter into the fast-retransmit state instead of the retransmission timeout state. However, as long as all the packets sent within the same TCP window arrive before time $L + t_e$, they are all affected by the attack traffic burst and are dropped.

Scenario 2: $i \times T = minRTO + \delta$ ($i \in N$) and no propagation delay. The attack inter-burst period may not exactly match minRTO due to lack of knowledge or precise timing control, as depicted in Figure 6. When $\delta > 0$, the first retransmitted packet will be dropped if $\delta \leq t_{x1} - t_f$. If all k retransmitted packets are dropped, then $\lfloor \frac{t_{x1}-t_f}{\delta} \rfloor + 1 \geq 2^k$. Similarly, when $\delta < 0$, δ needs to satisfy condition $\lfloor \frac{L+t_e-t_{x1}}{|\delta|} \rfloor + 1 \geq 2^k$ in order for all k retransmitted packets to be dropped. In both cases, the probability of BGP session reset can be modeled as

[4]MinRouteAdvertisementInterval is supposed to be per prefix and per peer, but it is often implemented to be per peer due to reasons of efficiency.

$$P(reset) = \frac{L + t_e - t_f - (2^k - 1)|\delta|}{T} \quad (3)$$

Scenario 3: With propagation delay, $i \times T = minRTO + \Delta$**.** Δ can be modeled as a random variable uniformly distributed in the range [-RTT, RTT], with RTT as the maximum roundtrip time. The probability of BGP session reset is

$$P(reset) = \frac{L + t_e - t_f - \sum_{j=1}^{2^k-1} \Delta_j}{T} \quad (4)$$

If the variance of Δ is large enough such that $\sum_{j=1}^{2^k-1} \Delta_j \geq L + t_e - t_f$, the BGP session is not reset. In fact, this is the rationale behind the defense mechanism based on the randomized minRTO where $minRTO$ within the range [a, b] is equivalent to Δ with large variance.

Scenario 4: Background traffi c has Poisson-distributed inter-arrival time. This is the most general case with the least restrictions. The rate of the background traffic at time t is $R_{t,TCP} = \lambda e^{-\lambda t}$, where λ is the average rate. Then the upper bound of t_f can be derived from the condition $t_f \leq \frac{B - B_0}{R_{DoS} + min\{R_{t,TCP}\} - R_{link}}$. The lower bound of t_e is from the condition $t_e \geq \frac{Pkt}{R_{link} - min\{R_{t,TCP}\}}$. Similar to Equation 1, we have the lower bound of the probability of BGP session reset:

$$P(reset) \geq \frac{L + Min\{t_e\} - Max\{t_f\} - \sum_{j=1}^{2^k-1} \Delta_j}{T} \quad (5)$$

B. Degradation of BGP Throughput

If the attack inter-burst period T or its integer multiple does not approximate the router $MinRTO$ value, the BGP session is not reset. However, the throughput of BGP traffic can have severe degradation leading to increased BGP convergence time. We denote the BGP traffic rate at time t as $R_{t,BGP}$. The average BGP throughput ρ_{BGP} under the DoS attack is:

$$\rho_{BGP} = \frac{\int_0^{t_f} R_{t,BGP} dt + \int_{t_f}^{L+t_e} 0 dt + \int_{L+t_e}^{T} R_{t,BGP} dt}{T} \quad (6)$$

In Equation 6, $\int_0^{t_f} R_{t,BGP} dt$ denotes the amount of BGP traffic passed through during time $[0, t_f]$. $\int_{t_f}^{L+t_e} 0 dt$ describes the fact that all BGP packets are dropped during burst period $[t_f, L + t_e]$. Assume that without the attack flow, all the packets can get through without loss (*i.e.*, $R_{Other} + R_{BGP} < R_{link}$). $\int_{L+t_e}^{T} R_{t,BGP} dt$ represents the amount of BGP traffic passed through during time $[L+t_e, T]$ (*i.e.*, after the attack burst).

According to Equation 6, given a fixed burst length and peak magnitude, a longer inter-burst period results in larger BGP throughput. This matches our observation in Figure 3(b). Similarly, given a fixed burst length and inter-burst period, a larger peak magnitude results in smaller t_f values and hence smaller BGP throughput. This also matches our observation in Figure 3(c).

C. Discussion

During the low-rate attack, background TCP flows may also encounter throughput degradation. Even if the attack inter-burst period does not precisely match the minRTO of background TCP flows, packet loss will be induced to cause background traffic to back off. In such cases, to be successful against BGP sessions, the attack traffic needs to be sufficiently strong (with a large enough attack burst length and peak magnitude) to repeatedly induce packet loss during each attack burst. Our testbed experiments in Section IV do not have background traffic. Adding background TCP flows to the testbed may decrease the initial t_f value; however, with sufficiently strong attack traffic it has little effect on subsequent bursts.

VI. Coordinated Low-rate Attacks

In the previous sections, our focus was on a simplified network setting with two topological advantages from attacker's view point: (i) The network path p from the attacker to a selected destination host (which the attacker is not required to have access to) goes through at least one link l of the BGP session under attack. (ii) The bottleneck link for the path p is link l between the two BGP routers involved in the session. If both conditions are satisfied, the attack host can send sufficient amount of traffic to congest the bottleneck link l, impacting the BGP session. This assumes that there is no rate-limiting occurring at somewhere along p before reaching the bottleneck link. Previous work by Kuzmanovic and Knightly [5] also assumes these conditions to hold in their experiments and simulations. We now explore how attackers can successfully impact a BGP session *without these conditions*. In our case of studying BGP sessions, it can be more difficult to impose these restrictions, as the link of interest may be close to the core network, likely with higher bandwidth compared with the case of attacking end hosts. Nevertheless, recent measurement study [19] indicates that more than 67% of randomly selected paths have bottleneck links between ASes, with more than half of such links having less than 16 Mbps of available bandwidth.

One way to overcome these difficulties is to launch a coordinated attack with multiple attack hosts. Using more than one host lifts the two previously imposed restrictions. Attacker can identify hosts whose network paths for selected destinations go through the links involved in the BGP session. Furthermore, the bottleneck link does not need to be shared, as long as the bandwidth of combined attack flows is sufficient to overload the link. The feasibility of such coordinated low-rate attacks depends on the number of attack machines an attacker controls, the available bandwidth of the target link, and the time synchronization granularity among different machines. Note that this is applicable to both single-hop and multi-hop BGP sessions. The link of interest is the link with the smallest available bandwidth between the two routers involved in a BGP session.

In what follows, we first illustrate two key steps in completing such coordinated low-rate attacks given a link of interest l involved in a BGP session: (i) selecting hosts whose network paths to chosen destinations traverse l. (ii) synchronizing attack traffic sending time so that attack traffic arriving at l will follow the desired square wave pattern. Then, attackers can estimate the number of attack hosts needed based on the available bandwidth of hosts and l, and the average sending rate for each host to avoid detection. At the end of this section, we use wide-area experiments to demonstrate the feasibility of such coordinated attacks.

To ensure that sufficient traffic arrives at the target link, more hosts can be used for the attack and they can send *overlapping traffic bursts*. The overlap can occur both in terms of attack amplitude as well as occurrence in time. Note that each attack host does not need to send traffic to the same destination host. It can pick any host to which traffic traverses the link of interest. As long as the average traffic rate from each individual host remains low, such attacks are very difficult to detect due to their low-rate properties and distributed nature in terms of both source and destination hosts. By increasing the number of attack hosts, each host needs to send very little traffic compared to the required final aggregate attack flow peak rate. Note that we do not discuss how to identify the correct inter-burst period, which should match closely the minRTO value.

A. Selection of Attack Hosts and Destinations

We assume that an attacker has identified a target link l involved in a BGP session between two routers on the Internet.[5] The BGP session can be any of the following: an eBGP session between two ISP peers or a customer and its provider, or an iBGP session within an AS. Here we only focus on eBGP sessions as the impact of such session resets is generally larger compared to iBGP sessions. We highlight the key steps in selecting attack hosts such that for their selected destination hosts the network paths traverse the link l. Our intuition is that we first identify network paths going through the desired AS-level hop, subsequently the IP-level hop. We rely on geographic information based on IP addresses through known mapping techniques [25], [26] to help narrow down the hosts. The BGP session may be attacked from both directions of the target link. Here we illustrate the steps for one direction applicable for the other direction. Routing changes over time; thus, this process of host selection needs to be repeated.

1. Identify the target link's geographic location and AS(es): Given the target link l denoted by its two router interface IP addresses, an attacker can map them to their ASes [27] and the approximate geographic location [25], [26]. We denote the IP addresses as IP_1 and IP_2, and the associated AS numbers as AS_1 and AS_2.

2. Identify host to prefix pairs whose path traverses the link of interest at AS level: We first identify at AS level host to prefix pairs whose network paths traverse the AS_1-AS_2 link. Assume that an attacker is equipped with a set of attack hosts. If BGP data are available from these hosts, destination prefixes whose AS path contains the subpath $[AS_1\ AS_2]$ can be easily identified. Otherwise, public BGP data from sources such as RouteViews [13] and RIPE [28] can be used. By taking advantage of destination-based forwarding, we find prefixes whose AS path contains the following pattern $[AS_x \cdots AS_1\ AS_2 \cdots]$, where AS_x is the upstream provider or the origin AS of the attack host, and "$\cdots$" denotes zero or more ASes.

3. Identify IP-level paths: We now have for each attack host a set of destination prefixes whose AS-level paths possibly go through the AS hop AS_1-AS_2. To select from those such their paths traverse IP_1-IP_2, we traceroute from each attack host a randomly selected IP from each of its destination prefixes to check if l is traversed. To reduce probing overhead, we can further narrow down the attack hosts by selecting those that are geographically close to the link l. This is especially useful if the target link is between two peers, which usually uses the hot-potato or early exit routing strategy.

[5]Similar to the minRTO discovery, we do not discuss how to identify such links, which can be achieved using network topology information.

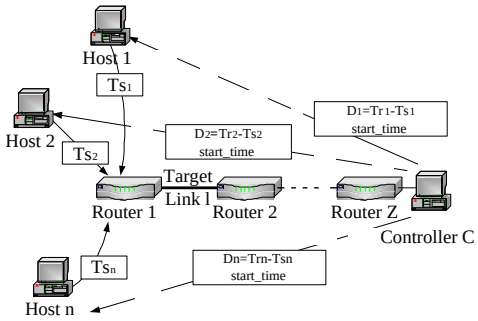


Fig. 7. Coordinated low-rate attacks: attack time synchronization.

B. Time Synchronization

Time synchronization is used to ensure that aggregate attack flows from diverse hosts arriving at the target link l follow the square wave shape for maximal attack efficiency. The differences both in each host's local clock and the network delay from each host to l necessitate time synchronization. Delay variability may limit the synchronization granularity. Our wide-area experiment shows that it is feasible to synchronize within 50 to 100 ms, sufficient for low-rate attacks. Next we describe our algorithm for attack synchronization amongst n attack hosts, as illustrated in Figure 7.

1. Select a reference time: The attacker needs to select one local clock as the reference time for computing the relative time to launch the attack. If the attacker has access to a controller host C, such that network paths from each attack host to C traverse l as shown in Figure 7 and also match the actual attack network path, C's local clock can be used as the reference time. C records the arrival time of packets from each attack host. Alternatively, an attacker can use a router Z instead, such that network paths from each attack

host to Z go through l. ICMP timestamp replies from Z [29], [30] serves as the reference time. If no such router is found, the attacker might find a live destination and send ICMP or IP timestamp request to it directly [31]. Ideally, the host or router serving as a reference time should be close to l, so that the variability in the delays of the network path segment from l to the host or the router will have minimal effect on the time synchronization.

2. Synchronize with the reference time: Once the controller C is identified, each host H_x sends a packet to C, embedding the sending local time Ts_x as the payload. C records the receiving time Tr_x based on its local time. Then C computes the time difference: $D_x = Tr_x - Ts_x$, capturing the difference in both the local clocks and the network one-way delays from H_x to C. After C receives a packet from each host, C obtains n time difference values $D_1, D_2, \ldots, D_n$. The controller C decides to start the attack at time $start\_time$ based on its local clock, allocating sufficient time for the packet to be received by each host. Subsequently, C sends a message to each host H_x with the value of D_x and $start\_time$. Upon receiving the message, host H_x starts attack at time $start\_time - D_x$ based on H_x's local clock. Alternatively, if router ICMP timestamps or host ICMP/IP timestamps are used as the reference time, upon receiving ICMP timestamp replies, the information needs to be aggregated to coordinate the attack starting time.

C. Case Studies: Wide-Area Experiments

Our wide-area experiments show the feasibility of selecting attack hosts and synchronizing coordinated attacks targeted at two types of link: a link between two tier-1 ISPs and a link between a customer and its provider. We use machines from the PlanetLab testbed [32] as candidates for attack hosts. Furthermore, we show a wide-area experiment to successfully reset a local BGP session in our testbed to demonstrate the feasibility of coordinated attacks.

Synchronizing coordinated attacks against a link between two peers: Without the loss of generality, we select our target link to be one between two large tier-1 ISPs with peer-to-peer relationship: ISP_A (AS_A) and ISP_B (AS_B), located in New York City (inferred from the DNS names) with interface IP addresses IP_A and IP_B, obtained from traceroute measurements. For security concerns, we anonymize the identity of the ISPs. We conjecture the presence of a BGP session between these two routers given that the IPs belong to two distinct ASes.

We follow the process outlined above to evaluate the feasibility of synchronizing a coordinated attack against this link.[6] We start with 153 PlanetLab hosts with distinct domains and determine their respective upstream providers using active probing. Using the routing table data from RouteViews [13], we identify prefixes with an AS path contains $[AS_u \cdots AS_A \ AS_B \cdots]$, where AS_u is the upstream provider for the host. After validating through active probing, we narrow down only three PlanetLab sites (two in northeast US and one in UK) whose hosts traverse the link to 19 destination prefixes. We find one PlanetLab host in Singapore that belongs to one of these prefixes to serve as the controller for the reference time.

[6]Note that if these routers implement protection for routing traffi c as we discuss later, the attack will be unsuccessful. Here we only focus on the attack synchronization aspect.

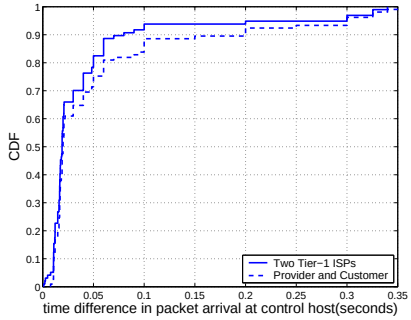


Fig. 8. Time difference for attack synchronization (8 hosts,100 runs).

Probing using the Pathneck [19] tool indicates that the available bandwidth of the target link is only 17.8 Mbps on average. Thus we pick 8 hosts from these three sites to study the synchronization granularity for a coordinated attack. The distribution of time difference between the earliest arriving packet and the latest packet among 8 hosts at the control host for 100 experiments is shown as the curve labeled with "Two Tier-1 ISPs" in Figure 8. More than 80% of the experiments can synchronize within 50 ms, and more than 90% within 100 ms. For attacks with inter-burst period of 600 ms and burst length of 200 ms, this synchronization granularity is sufficient.

Synchronizing coordinated attacks against a link between a provider and a customer: As an example of synchronizing an attack against a BGP session between a provider AS_P and its customer AS_C in the direction from the provider to the customer, we pick a tier-1 ISP and one of its customers peering in New York City with IP addresses obtained from traceroute measurements. Using RouteViews routing tables, we select PlanetLab hosts such that there exist AS paths containing $[AS_u \cdots AS_P \ AS_C \cdots]$, where AS_u is the upstream provider for the host. 35 sites located in U.S., Canada, and Asia are found to have PlanetLab hosts that traverse through the target link. This is much larger compared to the case for the link between two tier-1 ISPs where we found only two nearby sites and one European site, as explained by more path diversity in the core Internet. Two destination prefixes are identified whose paths from these hosts will traverse the target link. Using the Pathneck tool, the available bandwidth of the link is found to be around 22.7 Mbps. Again, we pick 8 hosts, with distinct geographic locations selected from the U.S., Canada, China, and Japan to test the time synchronization granularity. The curve labeled with "Provider and Customer" in Figure 8 shows that close to 90% of the experiments can synchronize within 100 ms, acceptable for the attack described.

Wide-area coordinated attacks against a local BGP

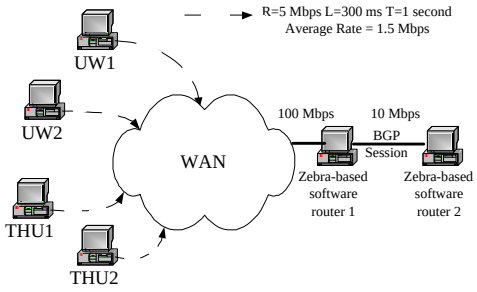


Fig. 9. WAN testbed to reset a local BGP session using 4 remote hosts.

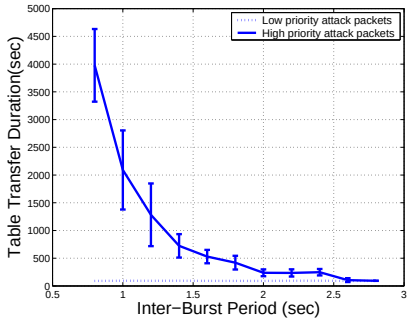


Fig. 10. Impact of inter-burst period on BGP table transfer with WRED enabled (R=185 Mbps, L=200 ms)

session: We have shown above the feasibility of synchronizing coordinated attacks against a core link between two tier-1 ISPs and a link between a tier-1 provider and its customer. We now perform an actual attack against a locally constructed BGP session, which we set up between two commercial PCs running Zebra software [33], with one PC serving as the gateway for the other shown in Figure 9. To increase the attack difficulty, only BGP KeepAlives are exchanged. The link capacity between the two PCs is 10 Mbps. We select four machines: two from University of Washington (UW) and two from Tsinghua University (THU) in China as the attack hosts for the purpose of geographic diversity.[7] Each attack flow has a burst length of 300 ms, an inter-burst period of 1 second, and a peak magnitude of 5 Mbps, thus the average rate is 1.5 Mbps. We conducted 10 experiments, each leading to a session reset. The required time to bring down the session varies from 3.7 minutes to 36.3 minutes with an average of 15 minutes and standard deviation of 11.0.

The above experiments demonstrate the feasibility to remotely reset a BGP session in the core Internet from multiple end hosts. With coordinated attacks, the burst durations can overlap among different flows so that each flow can have a shorter burst length and a longer inter-burst period. The peak magnitude for a single flow could be small as long as there are enough hosts sending traffic to fill up the available bandwidth. We next discuss how this and other low-rate attacks against BGP can be prevented.

VII. Defense Mechanisms

We discussed in Section II prior work on detecting low-rate attacks; however, no known detection techniques exist today that can accurately identify coordinated low-rate attacks given arbitrarily low-rate individual attack flows. Here we focus on prevention mechanisms so that low-rate TCP-targeted DoS attacks cannot impact BGP sessions. To achieve this we enumerate the necessary conditions for such attacks on BGP to be successful. (1) Ability to infer the minRTO value and send low-rate traffic with minRTO as the inter-burst period. (2) Ability to identify the location of the BGP session and send traffic passing through the target link involved in the BGP session. (3) Ability to congest the target link. We describe two general approaches to violate any of the above conditions through hiding the necessary information needed for the attack, and protecting BGP packets from other traffic.

[7]Given this artificial setup, any host can be used, as each host sending traffic destined to *router 2* will traverse the target link.

A. Hiding information

To successfully reset a BGP session, the attacker must know the minRTO value for the TCP stack on the target router. As shown earlier in Section IV, such information can be obtained by experimentally studying commercial routers. Different vendors and router types can have dissimilar minRTOs. Fingerprinting techniques such as nmap [34] or simply trial and error may be used for inference.

There are two ways related to minRTO to thwart the low-rate attack. First, if the minRTO, determining the attack inter-burst period, is small relative to the minimum burst length or smaller than the attack synchronization granularity, it would be impossible to launch low-rate attacks. The burst length needs to be sufficiently long to induce packet drops, and is usually several hundred milliseconds. If the minRTO is 200 ms, for example, the attack can no longer be low-rate.

Another way to mitigate the attack is to randomize the minRTO value as suggested by the prior work [5]. The minRTO value is specified by a range $[a, b]$. A random value within the range is assigned as the minRTO for each flow [5]. Randomization reduces the likelihood that attack flows will hit all consecutive retransmitted packets required to reset the session. However, it does not eliminate the impact on BGP throughput degradation.

Related to hiding minRTO values, another way to prevent low-rate attacks is to conceal network topologies from end-hosts, so that it will be impossible for attackers to identify target links and associated BGP sessions. In fact, many routers in the edge networks already disable ICMP TTL Time Exceeded replies through firewalls. Such replies are needed for traceroute to discover topologies. The deployment of MPLS will also make discovering internal ISP topologies difficult. The disadvantage is that legitimate use to discover topology will also be denied. Related to disabling ICMP TTL Time Exceeded replies, disallowing ICMP timestamp replies would make coordinated attacks more difficult to synchronize.

B. Prioritizing Routing Traffic

A direct approach to defeat low-rate DoS attacks against BGP sessions is to provide bandwidth guarantee for BGP traffic so that it is not affected by congestion caused by other traffic. This approach will protect routing traffic from both intentional attacks, whether brute-force or low-rate, as well as unintended traffic surges. Thus, this is the recommended approach. More importantly, this can be achieved today using existing router support. From a high level, there are two essential components: (1) *Prioritization*: each BGP router and any other router that may forward BGP traffic needs to prioritize BGP traffic at both input and output queues. (2) *Marking*: the edge router must ensure that non-BGP traffic is not marked with the high priority needed to differentiate BGP traffic for prioritization. If differentiation is based on router source IP addresses, source spoofing can be prevented using either ingress filtering or the Generalized TTL Security Mechanism (GTSM) [35].[8]

Next we empirically examine several widely implemented features for traffic differentiation on today's commercial routers in its effectiveness at protecting BGP traffic against low-rate attacks: Random Early Detection (RED) [16], [36], [37], Committed Access Rate (CAR) [38], and class-based policing [39]. These mechanisms either provide prioritized buffer access or link scheduling.

Random Early Detection (RED): RED [16] has been widely implemented in routers to help maintain small queue sizes and prevent TCP synchronization. Weighted RED (WRED) [36] allows traffic differentiation based on IP precedence by setting different RED parameters for each traffic class. In our experiments, we take Cisco recommendations [36] to give lower priority to packets marked with IP precedence of 0 (default) and higher priority to those marked with precedence of 6. Using these configurations, newly arriving high priority packets will not be discarded until all incoming low priority packets are dropped. By default, routers we studied do not enable WRED and mark locally generated BGP packets with a precedence value of 6. WRED is applied only at output queues, and is thus unable to protect routed BGP traffic for iBGP sessions.

We use the same experimental setup as in Figure 2 in Section IV except WRED is enabled for router R1's output queue. Under low-rate attacks configured with R=185 Mbps and L=200 ms, Figure 10 compares the table transfer duration when attack traffic uses IP precedence value of 0 (low priority) versus when it uses IP precedence value of 6 (high priority). In the former case, BGP table transfer is not impacted by the attack. In the latter case, WRED cannot protect BGP traffic resulting in similar performance as in Figure 3(b) without WRED. This illustrates the importance of policing the IP precedence marking on packets, so that attack packets will be treated with lower priority. We also found that WRED can prevent session reset for low-priority attack traffic.

Related to RED and WRED is RED with Preferential Dropping (RED-PD) [37], a flow-based mechanism that uses the packet drop history at the router to detect high-bandwidth flows in times of congestion and preferentially drop packets from these flows. Previous work [5] suggested RED-PD as a possible way to mitigate against such attacks, but concluded that it requires longer timescales to detect malicious attack flows. We use similar ns-2 based simulation environment to study how effective RED-PD can detect low-rate attacks against BGP sessions to thwart session reset attempts.

Previous work [5] shows that with RED-PD-specific parameters of K=3, M=5, RTT=40 ms, and bottleneck capacity of 1.5 Mbps, RED-PD can detect low-rate DoS attacks with R=2 Mbps, L =300 ms, and T=1.1 seconds. Attacks with a lower rate can evade the detection by RED-PD. Using similar RED-PD configurations, our simulations show that given R=2 Mbps and T=600 ms, as long as the burst length L does not exceed 230 ms (corresponding to session reset probability of 33.4%), such low-rate attacks can evade RED-PD detection. Similarly, given L=200 ms, and T=600 ms, as long as the peak rate R does not exceed 2.96 Mbps (corresponding to session reset probability of 32.9%), such low-rate attacks can also evade the RED-PD detection. This shows that RED-PD is not effective enough at preventing BGP session resets under low-rate attacks.

Class-based queueing and traffic marking: Today's routers generally support packet marking and class-based queueing using several criteria. For example, Committed Access Rate (CAR) [38] supported by Cisco routers we studied limits both the input and output transmission rates on an interface based on criteria such as incoming interface, IP precedence, QoS group, or IP access list, and also classifies packets by setting the IP precedence or QoS groups.

In our experiments, we configure CAR on the input interface to reset incoming attack packets to have IP precedence of 0, preventing attack packets from spoofing higher precedence values. We also configure CAR on the output interface to drop the packets with IP precedence of 0 when its burst rate exceeds 100 Mbps. We found that CAR is very effective in isolating BGP packets from attack traffic. The performance is similar to the curve marked with "low-priority attack packets" in Figure 10. Class-based policing [39] is a similar mechanism which we experimented on Cisco routers with the same effectiveness.

In summary, prevention mechanisms described here can be readily implemented in today's routers. Complementary router-supported features, such as Graceful Restart [15] and those proposed by the research community FRTR [40] can help reduce the overhead due to session resets. The existing focus of the network community [41], [42], [43] has been on practices such as preventing unauthorized router access by setting up access control lists and preventing router CPU overload by rate-limiting ICMP replies. However, this is

[8]This works by dropping packets with smaller than expected TTLs based on the number of hops between the two routers involved in a routing session.

not sufficient in protecting routers from remotely launched resource-based attacks such as low-rate attacks described here. We provide here suggestions to protect routing traffic from general DoS attacks including low-rate attacks beyond existing proposals. Operators need to configure routers to provide class-based queueing or prioritized buffer access for BGP traffic marked with higher priority. Edge routers must set up necessary filters to prevent attack packets from spoofing higher priority. Finally, hiding the network topology and infrastructure IP addresses also help protect the network. We recommend that router vendors enable such protection for routing traffic as the default configuration.

VIII. Conclusion

Attacks against Internet infrastructures such as routers can have devastating global impact on network stability and robustness. One fundamental weakness in today's Internet is that the control plane or routing packets by default is not protected from other traffic. Thus, congestion due to data packets as a result of either intentional attacks or unintended traffic bursts can adversely impact routing sessions. In this work, we examine the impact of low-rate TCP-targeted DoS attacks on BGP. Such attacks can be launched remotely without access to routers. Using detailed experimentation and analytical modeling, we show that routers using default configurations are vulnerable to such low-rate attacks. The attacked BGP session can suffer severe impact in the form of session reset and increased convergence delays, resulting in global network instability, unreachable destinations, and data plane performance degradation.

Unlike traditional flooding attacks, low-rate attacks are more stealthy and thus difficult to detect. Moreover, we illustrate coordinated low-rate attacks are feasible from multiple end-hosts, further raising the detection difficulty. As defense mechanisms, we advocate prevention techniques to eliminate the possibility of such low-rate attacks and other related DoS attacks which exploit traffic congestion to impact routing protocols. Fortunately, routers today already support prioritized treatment of routing traffic through scheduling and buffer management, combined with traffic policing to ensure normal traffic cannot spoof higher priority. Our work demonstrates the effectiveness of such solutions to prevent low-rate attacks. We strongly recommend operators to adopt such protection and router vendors to enable such configurations by default to improve the stability and robustness of the Internet.

References

[1] R. Richmond, "Firms Join Forces Against Hackers," *Wall Street Journal*, March 28, 2005.

[2] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)." RFC 1771, March 1995.

[3] J. Cowie, A. Ogielski, and B. Premore, "Internet Worms and Global Routing Instabilities," in *Proc. SPIE*, 2002.

[4] J. Aldridge and A. Vural, "A first look at Saturday's MS-SQL worm as seen by BGP activity recorded by RIS project." RIPE 44 Meeting, January 2003.

[5] A. Kuzmanovic and E. W. Knightly, "Low-Rate TCP-Targeted Denial of Service Attacks (The Shrew vs. the Mice and Elephants)," in *Proc. ACM SIGCOMM*, 2003.

[6] S. Kent, C. Lynn, and K. Seo, "Secure Border Gateway Protocol (Secure-BGP)," *IEEE J. Selected Areas in Communications*, 2000.

[7] H. Sun, J. C. Lui, and D. K. Yau, "Defending Against Low-rate TCP Attacks: Dynamic Detection and Protection," in *Proc. International Conference on Network Protocols*, 2004.

[8] A. Shevtekar, K. Anantharam, and N. Ansari, "Low Rate TCP Denial-of-Service Attack Detection at Edge Routers," *IEEE Communications Letters*, April 2005.

[9] X. Luo and R. K. C. Chang, "On a New Class of Pulsing Denial-of-Service Attacks and the Defense," in *Proc. Network and Distributed System Security Symposium*, 2005.

[10] Y. Chen, Y.-K. Kwok, and K. Hwang, "Filtering Shrew DDoS Attacks Using A New Frequency-Domain Approach," in *Proc. IEEE LCN Workshop on Network Security*, 2005.

[11] J. Ng, "Extensions to BGP to Support Secure Origin BGP (soBGP)." IETF Draft: draft-ng-sobgp-bgp-extensions-01.txt, November 2002.

[12] Cisco Systems, "Cisco Security Advisory: TCP Vulnerabilities in Multiple IOS-Based Cisco Products," 2005.

[13] "University of Oregon Route Views Archive Project." `http://www.routeview.org`.

[14] F. Wang, L. Gao, J. Wang, and J. Qiu, "On Understanding of Transient Interdomain Routing Failures," in *Proc. International Conference on Network Protocols*, 2006.

[15] S. R. Sangli, Y. Rekhter, R. Fernando, J. G. Scudder, and E. Chen, "Graceful Restart Mechanism for BGP." IETF Internet Draft, June 2004.

[16] S. Floyd and V. Jacobson, "Random Early Detection gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, 1993.

[17] P. Almquist, "Type of Service in the Internet Protocol Suite." RFC 1349, July 1992.

[18] A. Akella, S. Seshan, and A. Shaikh, "An Empirical Evaluation of Wide-Area Internet Bottlenecks," in *Proc. Internet Measurement Conference* , 2003.

[19] N. Hu, L. E. Li, Z. M. Mao, P. Steenkiste, and J. Wang, "A Measurement Study of Internet Bottlenecks," in *Proc. IEEE INFOCOM*, 2005.

[20] "Schooner User-Configurable Lab Environment." `http://www.schooner.wail.wisc.edu/index.php3?stayhome=1`.

[21] J. Padhye and S. Floyd, "Identifying the TCP Behavior of Web Servers," in *Proc. ACM SIGCOMM*, 2001.

[22] Cisco Systems, "NSF Awareness," 2005.

[23] G. Appenzeller, N. McKeown, J. Sommers, and P. Barford, "Recent Results on Sizing Router Buffers," in *Network Systems Design Conference*, 2004.

[24] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing Router Buffers," in *Proc. ACM SIGCOMM*, 2004.

[25] V. N. Padmanabhan and L. Subramanian, "An Investigation of Geographic Mapping Techniques for Internet Hosts," in *Proc. ACM SIGCOMM*, 2001.

[26] N. Spring, R. Mahajan, and D. Wetherall, "ISP Topologies with Rocketfuel," in *Proc. ACM SIGCOMM*, 2002.

[27] Z. M. Mao, J. Rexford, J. Wang, and R. Katz, "Towards an Accurate AS-level Traceroute Tool," in *Proc. ACM SIGCOMM*, 2003.

[28] "RIS Raw Data." `http://www.ripe.net/projects/ris/rawdata.html`.

[29] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "User-level Internet Path Diagnosis," in *Proc. ACM Symposium on Operating Systems Principles*, 2003.

[30] K. G. Anagnostakis, M. Greenwald, and R. S. Ryger, "Measuring Network-internal Delays using only Existing Infrastructure," in *Proc. IEEE INFOCOM*, 2003.

[31] "clockdiff." `http://www.linuxforum.com/man/clockdiff.8.php`.

[32] "PlanetLab." `http://www.planet-lab.org`.

[33] "GNU Zebra-routing software." `http://www.zebra.org`.

[34] "Nmap–Network Mapper." `http://www.insecure.org/nmap/`.

[35] V. Gill, J. Heasley, and D. Meyer, "The Generalized TTL Security Mechanism (GTSM)." RFC 3682, February 2004.

[36] Cisco Systems, "Weighted Random Early Detection (WRED)," 1998.
[37] R. Mahajan, S. Floyd, and D. Wetherall, "Controlling High-Bandwidth Flows at the Congested Router," in *Proc. International Conference on Network Protocols*, 2001.
[38] Cisco Systems, "Configuring Committed Access Rate," 2005.
[39] Cisco Systems, "Class-Based Policing," 2006.
[40] L. Wang, D. Massey, K. Patel, and L. Zhang, "FRTR: A Scalable Mechanism for Global Routing Table Consistency," in *Proc. International Conference on Dependable Systems and Networks* , 2004.
[41] Cisco Systems, "NSA/SNAC Router Security Configuration Guide," 2001.
[42] Juniper Networks, "Best common practices for hardening the infrastructure," 2002.
[43] Ryan McDowell, "Implications of Securing Backbone Router Infrastructure." NANOG Meeting, May 2004.