Compact neighbor discovery
(A bandwidth defense through bandwidth optimization)

Pars Mutaf
INRIA
Email: pars.mutaf@inria.fr

Claude Castelluccia
INRIA
Email: claude.castelluccia@inria.fr

***Abstract*— We present a stateless defense against the Neighbor Discovery Denial-of-Service (ND-DoS) attack in IPv6. The ND-DoS attack consists of remotely flooding a target subnet with bogus packets destined for random interface identifiers; a different one for each malicious packet. The 128-bit IPv6 address reserves its 64 low-order bits for the interface ID. Consequently, the malicious packets are very likely to fall on previously unresolved addresses and the target access router (or leaf router) is obligated to resolve these addresses by sending neighbor solicitation packets.**

Neighbor solicitation packets are link layer multicast (or broadcast), and hence also forwarded by bridges. As a consequence, the attack may consume important bandwidth in subnets with wireless bridges, or access points. This problem is particularly important in the presence of mobile IPv6 devices that expect incoming sessions from the Internet. In this case, address resolution is crucial for the access router to reliably deliver incoming sessions to idle mobile devices with unknown MAC addresses.

We propose a novel neighbor solicitation technique using Bloom filters. Multiple IPv6 addresses (bogus or real) that are waiting in the access router's address resolution queue are compactly represented using a Bloom filter. By broadcasting a single neighbor solicitation message that carries the Bloom filter, multiple IPv6 addresses are concurrently solicited. Legitimate neighbor solicitation triggering packets are not denied service. An on-link host can detect its address in the received Bloom filter and return its MAC address to the access router.

A bandwidth gain around 40 can be achieved in all cells of the target subnet. This approach that we call *Compact Neighbor Discovery (CND)* is the first bandwidth DoS defense that we are aware of to employ a bandwidth optimization.

I. Introduction

DENIAL-OF-SERVICE attacks that consume network resources continue to threaten the Internet. The freedom offered by the best effort and connectionless IP routing service is unfortunately exploited for DoS (Denial-of-Service) attacks that flood victim sites with bogus packets and consume the network resources. During the last decade, the Internet community had to fight against these attacks through research, standardization, emergency response and training. Nevertheless, attackers constantly search for new vulnerabilities for mounting new attacks and making as many victims as possible. Attackers have demonstrated deep knowledge about IP functions and their design rational. Any vulnerability is exploited. In an evolving Internet, DoS defense therefore requires thorough risk assessment. The potential threats that may emerge in the future must be identified and countermeasures must be developed where possible and before a threat becomes reality.

A. The Neighbor Discovery DoS attack

A flooding threat was recently discovered in the IPv6 Neighbor Discovery (ND) [1] protocol; the successor of ARP [2] in IPv4. The IETF SEND (SEcure Neighbor Discovery) working group analyzed the potential security flaws of ND and published the assessed risks in [3]. Most of them were on-link threats, i.e. exploitable by on-link attackers located in the same subnet as the victim(s). A notable example is the neighbor advertisement/solicitation spoofing attack. In ND, nodes on the same link use *neighbor solicitation* and *neighbor advertisement* messages to create bindings between IP addresses and MAC addresses. These entries are held in a data structure called *neighbor cache*. The neighbor advertisement spoofing attack consists of impersonating a victim node and modifying its neighbor cache entry held by other nodes. This attack causes packets for the victim, both hosts and routers, to be sent to some other link layer address. In its simplest form this attack makes the victim unreachable.

The SEND working group addressed these on-link threats using cryptographic techniques e.g. authentication, authorization and proof of address ownership [4], [5]. One particular flaw, however, was a flooding-based DoS attack mountable by any Internet node. This threat fell out of the working group's principal scope. The subject flooding threat is referred to as the *Neighbor Discovery DoS attack* and is at the focus of this paper.

The ND-DoS attack consists of flooding a target subnet with bogus packets destined for random interface identifiers; a different one for each malicious packet. The 128-bit IPv6 address reserves its 64 low-order bits for the interface ID. Consequently, the malicious packets are very likely to fall on previously unresolved addresses and the target access router (or leaf router) is obligated to resolve these addresses by sending neighbor solicitation packets. 2^{64} is a huge number. For any reasonable subnet size, the number of neighbor cache entries in access router memory will have no practical significance in front of ND-DoS. In theory, for example 5,000 neighbor solicitations per second can be remotely triggered during more than 1 million centuries (i.e. $\frac{2^{64}}{5{,}000}$ seconds) and without reusing a given interface ID.

One of the resources being attacked is the conceptual neighbor cache, which will be filled with attempts to resolve IPv6 addresses having a valid prefix but invalid interface ID. The SEND working group concluded that this impact can

be trivially defeated through efficient cache management, i.e. by restricting the amount of state reserved for unresolved solicitations [3].

B. Bandwidth cost of ND-DoS

In this paper we are rather focused on the bandwidth cost of the ND-DoS attack. In response to each malicious packet, the target access router is forced to link layer multicast a neighbor solicitation. This consumes bandwidth in the whole network since link layer multicast packets are also forwarded by bridges.

We recognize that the attack is unlikely to saturate a high speed target LAN operating at 100Mbps, if the ND-DoS packets must traverse a congested core Internet, or lower speed Internet links. However, wireless MAC protocols that operate at lower speeds may significantly suffer from it. An attacking node may saturate the wireless edges of the Internet using ND-DoS packets that are easily routed by the high speed core Internet routers. Basic 802.11 offers 1 or 2 Mbps transmission in the 2.4 GHz band using either frequency hopping spread spectrum (FHSS) or direct sequence spread spectrum (DSSS). 802.11b, an extension to 802.11, provides 11 Mbps transmission with a fallback to 5.5, 2 and 1 Mbps in the 2.4GHz band. 802.11b uses only DSSS. In theory 802.11a provides up to 54 Mbps in the 5GHz band by using an orthogonal frequency division multiplexing encoding scheme rather than FHSS or DSSS. However, it has a smaller transmission range than 802.11b. New wireless MAC protocols are also being standardized. 802.20 will support peak data rates per user around 1 Mbps in the 3GHz band. Another MAC-layer technology 802.16 is intended to support individual channel data rates of from 2Mbps to 155Mbps, but aimed at much lower speed user mobility models than 802.20.

Wireless technology will probably evolve and transmission speeds will increase in the future. However, the core Internet will probably profit from new technologies as well. Consequently, the gap between wireless transmission speeds and DoS flooding rates may persist. It is therefore necessary to counter the ND-DoS threat through protocol design.

Wireless MAC protocols attract our interest also because mobile IP devices will expect incoming real-time sessions from the Internet [6]. In this case, neighbor discovery upon incoming packet is crucial for reliable incoming session delivery to idle mobile devices with unknown MAC address.

C. A stateful defense (previous work)

A possible defense may consists of replacing address resolution by a stateful mechanism. This is briefly mentioned in [3]. In this approach, access to the link is restricted to "registered" nodes, and the access router keeps track of nodes that have registered for access on the link. When the access router receives a packet destined for an unregistered IPv6 address, it simply drops the packet. While this approach can effectively defeat the ND-DoS attack, it will probably bring its own problems. For example, the access router cannot determine whether an incoming packet is ND-DoS packet or the corresponding entry was lost (due to reboot with loss of state, temporary memory outage or access router replacement). This approach may also suffer from race conditions. For example, an incoming legitimate packet would be lost, if the destination host could not yet register. Note that the original design of ND (and its predecessor ARP) defeats these problems by employing a truly reliable protocol that can resolve unknown IP addresses.

D. Proposal overview

In this paper we propose an alternative approach using Bloom filters [7]. Our proposal is resistant to loss of state, hence it can complement the above approach for a loss of state scenario, or replace it. We integrate a bandwidth optimization into standard neighbor discovery. A bandwidth gain of about 40 can be achieved. This implies that in front of an ND-DoS attack that triggers for example 5,000 neighbor solicitation packets per second, 92.9Kbps will be consumed instead of 3.7Mbps in theory. Consequently, modern wireless access protocols such as 802.11b or similar protocols that would otherwise be very vulnerable, can easily resist the bandwidth impacts of the attack. This is achieved by compactly representing the IPv6 addresses (bogus or real) in the access router's address resolution queue using a Bloom filter. By broadcasting a single neighbor solicitation message that carries the Bloom filter, multiple IPv6 addresses are concurrently solicited. Legitimate neighbor solicitation triggering packets are not denied service. An on-link host can detect its address in the received Bloom filter and return its MAC address to the access router.

This approach that we call *Compact Neighbor Discovery (CND)* is the first bandwidth DoS defense that we are aware of to employ a bandwidth optimization.

The rest of this paper is organized as follows: Section II describes the details of Bloom filters and Compact Neighbor Discovery, Section III provides an analysis, Section IV provides experimentation results, Section V presents discussions and finally Section VI concludes the paper.

II. Compact Neighbor Discovery

A. Neighbor discovery terminology

In IPv6, nodes on the same link use the *Neighbor Discovery (ND)* protocol to discover each other's presence, to determine other nodes' link layer addresses, find routers and to maintain reachability information about paths to active neighbors [1]. IPv6 neighbor discovery is defined as part of ICMPv6 [8].

In ND, nodes on the same link use *neighbor solicitation* and *neighbor advertisement* messages to create bindings between IP addresses and MAC addresses (i.e. for resolving IP addresses). These entries are held in a data structure called *neighbor cache*.

A neighbor cache entry may be in various states. For the purposes of these paper, we are mostly interested in INCOMPLETE and REACHABLE states. Before soliciting a *target address*, the soliciting node creates a neighbor cache

entry in INCOMPLETE state, which transits to REACHABLE state upon receipt of the destination's neighbor advertisement.

In order to limit the storage needed for the neighbor cache entries, a node may garbage collect old entries. [1] suggests that implementations should insure that sufficient space is always present to hold the working set of active entries. A small cache may result in an excessive number of neighbor discovery messages if entries are discarded and rebuilt in quick succession. Policies that remove entries that have not been used in some time (e.g., ten minutes or more), are recommended.

B. Bloom filters

A Bloom filter [7], named after its inventor Burton Bloom, is a randomized data structure that allows for compact representation of a set $A = \{a_1, a_2, ..., a_n\}$, using a m-bit vector (called Bloom filter), and supports membership queries.

The procedure requires k uniform and independent hash functions $h_1(), h_2(), ..., h_k()$, where $1 \leq h_i() \leq m$. First all bits of the bit vector are set to 0. Then for each element $a_i \in A$ the bit positions $h_1(a_i), h_2(a_i), ..., h_k(a_i)$ are set to 1 (a particular bit may be set more than once). The resulting bit vector, called a Bloom filter, represents all members of the set A.

In order to check whether $b \in A$, the bit positions $h_1(b), h_2(b), ..., h_k(b)$ of the Bloom filter are checked. If any of them is 0, then b is certainly not an element of A. Otherwise, $b \in A$. However, there is a small probability that all bits at positions $h_1(b), h_2(b), ..., h_k(b)$ are set although $b \notin A$. This is called a *false positive*. The false positive probability depends on the Bloom filter size m, the number of elements n that are inserted into the Bloom filter and the number k of hash functions.

The false positive probability and the optimal number of hash functions that minimizes it are well-known:

After inserting n elements to a bit vector of m bits, the probability that a particular bit is still 0 is $(1 - 1/m)^{kn}$. The probability of false positive in this situation is

$$F = \left(1 - (1 - \frac{1}{m})^{kn}\right)^k \simeq (1 - e^{-kn/m})^k$$

and minimized for

$$k_{opt} = (ln2) \times \frac{m}{n} \quad (1)$$

in which case it becomes

$$F = (0.6185)^{m/n} \quad (2)$$

Throughout the paper we will frequently refer to Eq.2. I.e. in this paper the optimal number of hash functions (that minimizes the false positive probability) is always used for any $\frac{m}{n}$ ratio.

C. Protocol description

Compact Neighbor Discovery (CND) replaces the 128-bit target address of a standard neighbor solicitation packet by a m-bit Bloom filter. The packet also carries the optimal number k_{opt} of hash functions calculated (in advance) regarding Eq.1, and the Bloom filter size.

Let n be the number of IPv6 addresses waiting in the access router's address resolution queue. If $n = 1$ and time has come to send a neighbor solicitation, standard neighbor discovery is applied. Otherwise, the set of n IPv6 addresses are inserted by the access router into a m-bit Bloom filter. I.e., for each IPv6 address IP_i in the queue, the bit positions $h_1(IP_i), h_2(IP_i), ..., h_k(IP_i)$ of the m-bit bit vector are set to 1. The resulting packet is sent to the all-nodes multicast address (i.e., link layer broadcast address).

Upon receipt of a CND neighbor solicitation packet, an on-link node IP_j performs a Bloom filter membership query by checking the k bit positions $h_1(IP_j), h_2(IP_j), ..., h_k(IP_j)$ of the Bloom filter (the Bloom filter bit positions are computed only once per configured IPv6 address). If any of them is 0, then IP_j is certainly not being solicited. Otherwise, IP_j is being solicited and should respond with a neighbor advertisement packet. The CND procedure is illustrated in Figure 1. CND introduces a small *unnecessary neighbor advertisement* probability F due to false positives. A node IP_y may receive a CND neighbor solicitation packet with set Bloom filter bits at positions $h_1(IP_y), h_2(IP_y), ..., h_k(IP_y)$, although its address is not being solicited. In this case the node IP_y will send an unnecessary neighbor advertisement.

Staying in line with the original design principles of ND and ARP, we assert that CND cannot avoid the unnecessary neighbor advertisements. For example, a host that recently sent a neighbor advertisement to its access router, may find a second solicitation suspicious (i.e. probably false positive occurred during Bloom filter query). However, there is no reliable way for the host to determine whether its neighbor cache entry (if any) was not lost or prematurely garbage collected by the access router. The host should reply to the neighbor solicitation, otherwise an incoming packet or session may be missed.

Fortunately the unnecessary neighbor advertisement rate can be very much reduced. Bloom filters allow important false positive rate reductions, which can be further improved in the case of CND as we describe in the following section.

III. ANALYSIS

A. Notation

- n - the number of concurrently solicited IPv6 addresses.
- m - Bloom filter size (bits).
- J - Neighbor solicitation triggering packets per second received by the access router (from anywhere in the Internet). This represents the total rate of malicious and legitimate packets that trigger neighbor solicitations. A large J is the result of an ND-DoS attack.

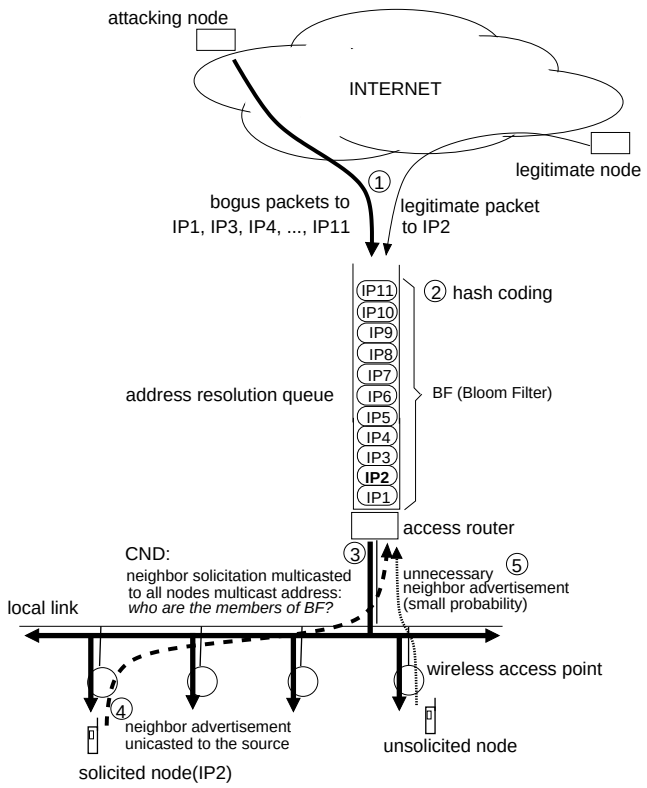


Fig. 1. Example CND procedure. In this example $n = 11$. The access router cannot differentiate the ND-DoS packets from legitimate ones. The defense employed by CND is therefore based on a stateless bandwidth optimization, rather than trying to reject malicious packets. The access router solicits all IPv6 addresses (bogus or real) in the queue using a compact Bloom filter presentation, which reduces the damages.

- G - The bandwidth gain of CND compared to ND.
- F - False positive, or unnecessary neighbor advertisement, probability upon receipt of a broadcast CND neighbor solicitation packet.
- $F_{allowable}$ - Allowable false positive rate per host. In practice, we will fix F to an allowable false positive rate of our choice. This is denoted $F_{allowable}$, and for a given Bloom filter size, it limits the number of concurrently solicited addresses.
- n_{max} - the maximum number of IPv6 addresses that can be solicited as a function of $F_{allowable}$ and m.
- G_{max} - the maximum bandwidth gain that can be obtained as a function of $F_{allowable}$ and m.
- r_{ns} - Neighbor solicitation rate of the target access router (neighbor solicitation packets per second).
- r_{una} - Unnecessary neighbor advertisement rate of a host in the target subnet (neighbor advertisement packets per second).
- p - Number of hosts per cell.

B. Unnecessary neighbor advertisement rate

In order to avoid address resolution queue overflow, the access router needs to set

$$r_{ns} = \left\lceil \frac{J}{n} \right\rceil$$

At first look, we would define the unnecessary neighbor advertisement rate of a given node in the target subnet as

$$r_{una} = \left\lceil \frac{J}{n} \right\rceil \times F$$

unnecessary neighbor advertisements per time unit.

However, there exists a simple optimization which further reduces the unnecessary neighbor advertisement rate. This optimization consists of employing $c = \lceil \frac{J}{n} \rceil$ different queues, and grouping the target addresses regarding their $log_2(c)$ low-order bits. These bits form what we can call a *group ID*. Only the addresses with the same group ID are inserted into the same Bloom filter. Each CND neighbor solicitation packet is attached the group ID which corresponds to the addresses inserted in the carried Bloom filter. Then, assuming uniformly distributed target addresses, a given on-link host receives one CND neighbor solicitation packet that carries its group ID, out of c. No membership query is made for an unmatching group number, which divides the average false positive rate by c, in which case we can define

$$r_{una} = F$$

unnecessary neighbor advertisements per time unit. This result is surprising in that the false positive rate per time unit does not depend on J. As we will later show, the maximum n is bounded by other parameters. Paradoxically we take the advantage of this limitation by creating $c = \lceil \frac{J}{n} \rceil$ groups and divide the unnecessary neighbor advertisement rate by c. The larger the $\lceil \frac{J}{n} \rceil$ ratio the more frequent must be r_{ns}, which would normally increase the unnecessary neighbor advertisement rate. However, the same $\lceil \frac{J}{n} \rceil$ ratio also offers a false positive rate reduction possibility, which cancels the effect of $\lceil \frac{J}{n} \rceil$.

C. Bandwidth gain

CND is not dependent on MAC layer specifics. However, for analytical convenience (i.e. readability of our equations), we need the exact packet sizes and therefore this section builds on a case study of 802.11b. Some aspects of the following analysis are only theoretical. For example, we will vary the number of terminals per 802.11b cell between 30 and 300. By current practice 802.11b is not used with that large cells (more than $p = 30$ is not common), however this analysis will be needed for evaluating CND's sensitivity to cell sizes. The same analysis can easily be adapted to other MAC protocols by using the appropriate MAC header size.

The 802.11b header is 28 bytes long, the IPv6 header is 40 bytes long, and a standard neighbor solicitation/advertisement

packet is 24 bytes long, which gives a total of 736 bits. A CND neighbor solicitation packet is 608+m bits long[1].

Using ND, the access router must set $r_{ns} = J$ in order to avoid address resolution queue overflow. Consequently, the attack will consume

$$B_{ND} = J \times 736$$

bits per second in each cell of the target subnet.

Using CND, the access router can set $r_{ns} = \frac{J}{n}$ since n addresses are concurrently resolved. However, each host in the target subnet will send $r_{una} = F$ unnecessary neighbor advertisements per second as previously evaluated. Thus, a total of

$$B_{CND} = \frac{J}{n} \times (608 + m) + 736\ p\ F$$

bits per second will be consumed in each cell of the target subnet. The bandwidth gain ($G = \frac{B_{ND}}{B_{CND}}$) is

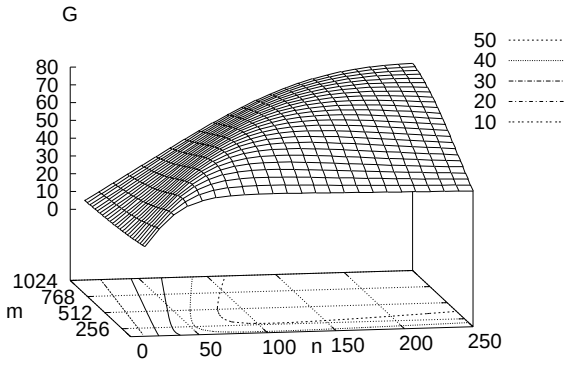
$$G = \frac{736}{\frac{608+m}{n} + \frac{736pF}{J}} \quad (3)$$

Figure 2 illustrates the bandwidth gain of CND, for different cell sizes. Small $\frac{m}{n}$ ratio results in high false positive rate which cancels the bandwidth gain of CND. This effect is amplified as the cell sizes are increased. It can be noted that there is an n value that maximizes the bandwidth gain for a given Bloom filter size. However, in practice, n has a smaller upper bound that depends on the "allowable false positive rate" per host, that we denote $F_{allowable}$. The transmission of unnecessary neighbor advertisements may represent important power drain on battery powered devices. Therefore, we assert that $F_{allowable}$ is small, for example 0.001. From Eq. 2, it can easily be shown that the maximum number of addresses that an access router can concurrently solicit is bounded by

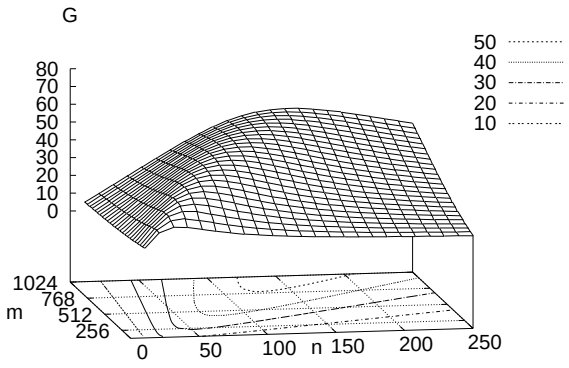
$$n_{max} = \frac{m \times log(0.6185)}{log(F_{allowable})} \quad (4)$$

I.e., given a Bloom filter size and allowable false positive rate, there is an upper bound on the number of addresses that can be concurrently solicited. Figure 3 shows the bandwidth gain replotted under $F \leq 0.001$ constraint (for $p = 30$ only). On the illustrated surface that resembles a triangle, CND is practical i.e. it does not represent important false positive overhead to battery powered hosts. For larger cell sizes e.g. $p = 100$ and $p = 300$, we obtained approximately the same CND surface. This is due to the fact that F is very small and J is very large. We have an extremely small $\frac{F}{J}$ ratio; in the order of 10^{-6} or less. For a reasonable cell size e.g. not considerably larger than $p = 300$, we have $\frac{608+m}{n} \gg \frac{736pF}{J}$. Thus, the constant $\frac{736pF}{J}$ has negligible contribution to Eq. 3 and can be omitted. This approximation assumes reasonable Bloom filter sizes (i.e. not very large ones), which limits n_{max}.

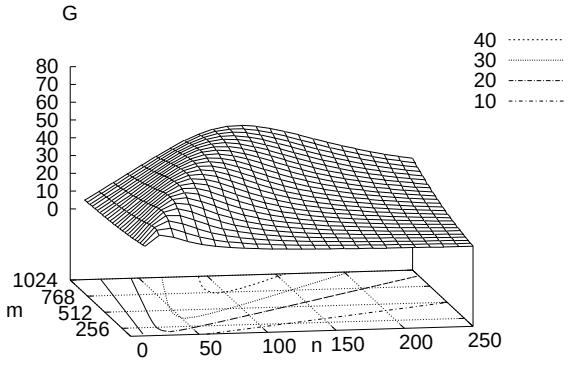
[1]This does not take into account the bits consumed for representing k, m and c. In practice, the standard neighbor solicitation message reserves 32 bits for future use, which are already taken into account in this analysis. For analysis simplicity we assume that some reserved bits are used for CND.



(a) p=30



(b) p=100



(c) p=300

Fig. 2. The bandwidth gain per cell in the attacked subnet (J=1000).

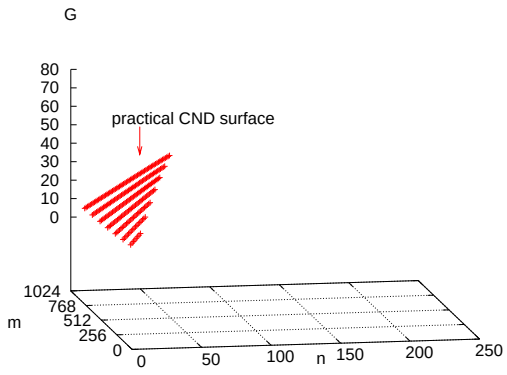


Fig. 3. Bandwidth gain per cell in the attacked subnet, replotted under $F \leq 0.001$ constraint. This surface was plotted for $p = 30$. With $p = 100$ and $p = 300$ approximately the same result was obtained (hence, not shown). This surface is roughly a triangle; $\frac{\partial G}{\partial n}$ is practically constant and equal to $\frac{736}{608+m}$. The reader may notice that this triangle is common to the three surfaces illustrated in Figure 2.

This requirement is automatically met since the CND neighbor solicitation packet size is limited by the MTU which is 1500 bytes. Regarding our calculations, this approximation yields 1.55% error with the largest possible Bloom filter, a moderate attack rate $J = 1,000$, $F_{allowable} = 0.001$ and a large cell size $p = 300$. Thus, we can simplify our bandwidth gain formulation as

$$G \simeq \frac{736n}{608 + m} \tag{5}$$

to which approximately corresponds the CND surface illustrated in Figure 3 (plotted under $F \leq 0.001$ constraint).

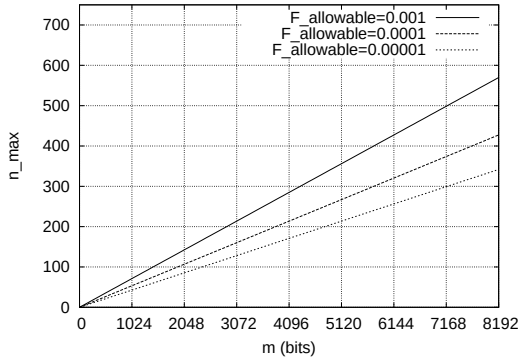
D. Maximum bandwidth gain

From performance evaluation standpoint we are also interested in the maximum bandwidth gain (or, bandwidth protection) that can be obtained under energy constraints. By replacing n by n_{max} in Eq. 5, we obtain the maximum bandwidth gain of CND as a function of m and $F_{allowable}$

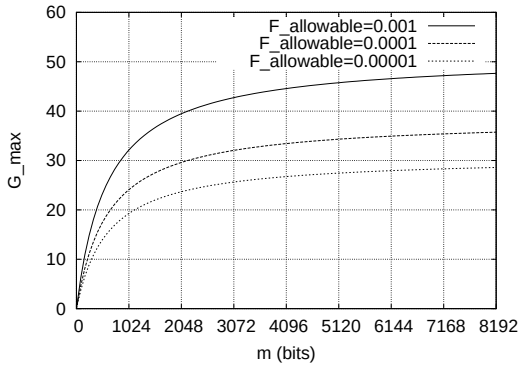
$$G_{max} = \frac{-153.57\ m}{log_{10}(F_{allowable})(608 + m)} \tag{6}$$

The maximum bandwidth gain formulation assumes that the necessary number of addresses n_{max} are always available in the address resolution queue. During a ND-DoS attack this condition is automatically met since neighbor solicitation triggering packets arrive at a very high rate. The number of queued IPv6 addresses quickly reach n_{max}, without introducing important CND delay to legitimate packets.

Figure 4 shows the n_{max} and the corresponding G_{max} values as a function of m and $F_{allowable}$. Regarding these curves, the following important conclusion is drawn: the constraint on $F_{allowable}$ should not be exaggerated. A too small $F_{allowable}$ unnecessarily reduces the number of addresses that can be



(a)



(b)

Fig. 4. The maximum number of hosts that can be concurrently solicited as a function of $F_{allowable}$ and m, and the corresponding bandwidth gain.

concurrently solicited, hence the bandwidth gain of CND. Table I shows the expected unnecessary neighbor advertisement rate of a host. Without justification, $F_{allowable} = 0.001$ seems a reasonable choice. This setting offers important bandwidth protection without significant false positive overhead on battery powered devices.

This issue has also bearings on the hash computation cost paid by the access router. When under ND-DoS attack, the access router computes $k_{opt} \times J$ hashes per second; k_{opt} hash functions are applied to the destination IPv6 address of each ND-DoS packet. From Eq. 1 and Eq. 2, it is easily shown that

$$k_{opt} = (ln2) \times \frac{log(F_{allowable})}{log(0.6185)} \tag{7}$$

It can be noted that an unnecessarily small $F_{allowable}$ will increase k_{opt}. Modern hash functions run at very high speeds, and in our case the hash input is a 128-bit IPv6 address which is relatively small. However, there is no point in unnecessarily increasing the hash computation cost for an exaggerated constraint on $F_{allowable}$.

$F_{allowable}$	r_{una}
10^{-3}	1 per 16 minutes
10^{-4}	1 per 2.8 hours
10^{-5}	less than 1 per day
10^{-6}	less than 1 per 11 days

TABLE I

EXPECTED UNNECESSARY NEIGHBOR ADVERTISEMENT RATE OF A HOST (WHILE ITS SUBNET IS UNDER ATTACK).

		CND bw		
J	ND bw	$m = 128$	$m = 1024$	$m = 2048$
1,000	736Kbps	81.8Kbps	22.7Kbps	18.6Kbps
5,000	3.7Mbps	408.9Kbps	113.4Kbps	92.9Kbps
10,000	7.4Mbps	817.8Kbps	226.7Kbps	185.8Kbps

TABLE II

EXAMPLE ND-DOS ATTACK RATES AND EXPECTED DAMAGES WITH ND AND CND ($F_{allowable} = 0.001$).

E. Estimated bandwidth consumption

The estimated ND-DoS damages with and without CND are shown in Table II. Bloom filters are quite efficient. The smallest Bloom filter that we define replaces the 128-bit IPv6 address of the standard neighbor solicitation message. A 128-bit Bloom filter allows to concurrently solicit 9 addresses under $F_{allowable} = 0.001$ constraint, which offers important bandwidth savings. A 128-bit Bloom filter saves most of the bandwidth offered by 802.11b in front of a serious ND-DoS rate that results in $J = 10,000$. Better protection is obtained using larger Bloom filters. On the other hand, we note that increasing the Bloom filter size from 1024 to 2048 bits will have no significant benefit for 802.11b (the difference is about 30Kbps when $J = 10,000$). The Bloom filter size becomes less important beyond a certain point, which can also be observed in Figure 4-b.

In the following section we show by experiments that large Bloom filters are rather useful for saving CPU cycles in access points with limited CPU.

IV. EXPERIMENTS

In the previous analysis we assumed that wireless access points can keep pace with the transmission speed of their MAC protocol. Consequently, we concluded that very large Bloom filters have no significant benefit (as observed in Figure 4-b.). This result changes however when the bottleneck is the access point CPU rather than wireless bandwidth.

We built a testbed in order to measure the *neighbor solicitation throughput* of a commodity 802.11b wireless access point. By "neighbor solicitation throughput", we mean the number of addresses per second that can be solicited through an 802.11b access point. We denote it T (addresses per second). The testbed consists of 2 machines and a 802.11b access point. An access router emulator (under ND-DoS attack) generates broadcast neighbor solicitation packets at a high rate. These packets are forwarded by the access point, and counted by a 802.11b terminal. We measure the number of neighbor solicitation packets that the access point can forward per second, and denote it Z.

We observed that the access point can forward at most $Z \simeq 1,500$ standard neighbor solicitation packets per second, which corresponds to $1,500 \times 736$ = 1,1Mbps. Our access point's CPU cannot keep pace with 802.11b bandwidth and less ND-DoS resistant than in theory. Note that a standard neighbor solicitation message can solicit one IPv6 address, and hence using standard ND we have $T = Z$, i.e. in our case $T = 1,500$ addresses per second can be solicited using standard ND.

Next we measured Z with CND which however defines larger neighbor solicitation packets, due to its Bloom filter overhead. We observed that the access point can forward a smaller number of CND neighbor solicitation packets per second. However, each CND neighbor solicitation packet solicits many addresses. The results are shown in Table III. The CND(m) notation is used for different Bloom filter sizes and $F_{allowable} = 0.001$ is set. The larger the Bloom filter size, the better is the neighbor solicitation throughput and performance increase is important as shown in Figure 5. Clearly, higher neighbor solicitation throughput implies smaller CPU cost per solicited address. The larger the Bloom filter size, the more access point CPU cycles will be left to active sessions.

	Z	n_{max}	$T = Z \times n_{max}$
ND	1,500	-	1,500 addr/s
CND(128)	1,500	9	13,500 addr/s
CND(1024)	1,200	71	85,200 addr/s
CND(2048)	900	142	127,800 addr/s
CND(4096)	600	284	170,400 addr/s
CND(8192)	400	568	227,200 addr/s

TABLE III

NEIGHBOR SOLICITATION THROUGHPUT OF AN ACCESS POINT (IN ADDRESSES/SECOND).

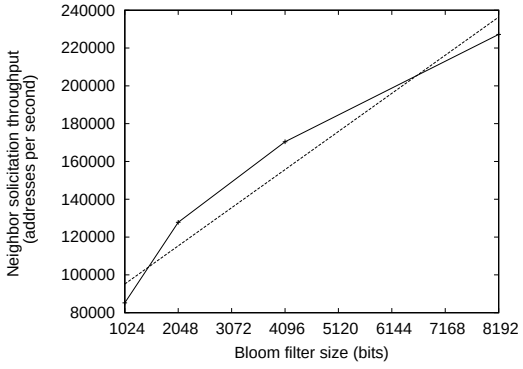


Fig. 5. Measured neighbor solicitation throughput of an access point, with CND, as a function of Bloom filter size. Higher throughput implies lower CPU cost per solicited address.

Another important conclusion of this analysis is that faster wireless MAC protocols such as 802.11a may also need CND defense in practice. 802.11a provides up to 54 Mbps which

may be difficult to consume with a remotely triggered ND-DoS attack. If however, the access point has a limited CPU, CND defense may help.

V. Discussion: router performance issues

Inspired from [9], we have implemented Bloom filters using the cryptographic hash function MD5 [10]. MD5 is a message digest algorithm that produces a 16-byte uniform output, which we divide into 8 different 16-bit hash results. $8 \times x$ hash outputs are obtained by concatenating the input IPv6 address with x different values. By reducing the results to modulo m, different Bloom filter bit positions are obtained. 16-bit hash output allows up to 65536 different bit positions, which is considered more than enough since the Bloom filter size is limited by MTU (with 1500 bytes MTU, the maximum Bloom filter size is 11,392 bits). Note also that, all nodes, routers and hosts, must use the same ordered set of hash functions (the first k_{opt} functions of the set will be applied to an IPv6 address). Therefore, the concatenated values and their order must be well-known.

Under $F_{allowable} = 0.001$ constraint which requires $k_{opt} = 9$ (by Eq. 7), MD5 is called twice for each neighbor solicitation triggering packet. When input a 16-byte IPv6 address, MD5 as implemented in [10] takes on the average about 6 microseconds on a 863.735MHz Intel Pentium III. In these conditions, in theory, it is possible to cope with $J = \frac{1}{(6\times2)10^{-6}} = 83,333$ neighbor solicitation triggering packets per second.

The total number of instructions and memory accesses per neighbor solicitation triggering packet will depend on the internal details of router architecture and neighbor discovery implementation. One neighbor cache lookup is probably unavoidable for an incoming packet. If the destination address has no neighbor cache entry, CPU cycles will be consumed for creating a neighbor cache entry in INCOMPLETE state and soliciting the address.

Neighbor cache lookup cost during ND-DoS attack may also be reduced using Bloom filters. Similar techniques have been proposed for speeding IP lookup in Internet routers and name lookup in distributed systems [11], [12]. In our case, the access router can represent the IPv6 addresses that have neighbor cache entries in its subnet using one or more Bloom filters. Upon incoming packet, the access router can check the destination's Bloom filter membership. If the destination IPv6 address is known i.e. it was coded into a Bloom filter, the access router can proceed to fetch the destination MAC address from neighbor cache (otherwise neighbor solicitation will be initiated). Provided that hash computation is faster than memory access, the compact Bloom filter presentation may be preferred when the access router is under ND-DoS attack (i.e. when most of the incoming packets are destined for addresses without neighbor cache). In this optimization false positives will lead to a small rate of unnecessary neighbor cache lookup. This has no harm except the time consumed for a misleading Bloom filter query. The unnecessary neighbor cache lookup rate that is avoided under ND-DoS attack can outweigh the cost of a small positive rate. Note also that some neighbor cache entries will expire over time. An aged Bloom filter that is overly populated with IPv6 addresses that are no longer used can be reset and reconstructed with the actual IPv6 addresses found in the neighbor cache.

When under ND-DoS attack, if the access router has difficulties in creating J neighbor cache entries per second, the INCOMPLETE state of the neighbor cache may be skipped. In this case, upon receipt of n neighbor solicitation triggering packets, the access router will not create neighbor cache entries in INCOMPLETE state (nor buffer the packets) but solicit the destination IPv6 addresses using a single CND packet. If one (or more) of the solicited addresses was real, the access router will receive a neighbor advertisement from the destination host and create a neighbor cache entry in REACHABLE state. Having resolved the destination address, the access router can route the incoming packet next time i.e. when retransmitted by the source.

Beyond a certain ND-DoS attack rate, some neighbor solicitation triggering packets will need to be randomly dropped by the access router; first because the bandwidth gain of CND is not unlimited, and also for the router's own protection. Legitimate packets will be retransmitted and may not be dropped next time, and the destination address can be resolved. Once an address is resolved, the packets destined to it will not enter the address resolution queue. CND will reduce the rate of dropped neighbor solicitation triggering packets, making incoming sessions more likely to succeed. However, this optimization has a limit.

VI. Conclusion

We have represented a stateless defense against the Neighbor Discovery Denial-of-Service (ND-DoS) attack in IPv6. ND-DoS is a flooding based attack that can trigger an outstanding rate of link-layer multicast (or, broadcast) packets in a target subnet. This attack is possibly the most serious bandwidth threat for future IPv6 subnets with wireless access points and mobile IP devices that expect incoming sessions from the Internet.

In flooding-based DoS attacks, attacking nodes may employ IP spoofing, i.e. malicious packets may have randomly changing source IP addresses. This technique is known to hide the attack's origin. IP spoofing also makes difficult (if not impossible) for a victim to distinguish malicious packets from legitimate ones. Network ingress filtering is an effective solution against DoS attacks that employ IP spoofing[13]. In this approach, by early dropping the packets with topologically incorrect source addresses, IP spoofing packets are filtered at the source and never reach the target network. Unfortunately, network ingress filtering is rarely activated by the ISPs although modern routers implement it. Consequently, IP traceback mechanisms have been proposed for tracing IP spoofing packets towards their origin ([14], [15] to cite a few examples). Ingress filtering and IP traceback have a wide scope of defense in that they could counter any flooding-based attack that deploys IP spoofing, e.g. SYN-flooding[16], Smurf[17].

We have rather proposed a bandwidth defense through bandwidth optimization; a defense that is specific to ND-DoS. Thus, the offered ND-DoS protection is not dependent on ingress filtering or IP traceback which may never find world wide deployment, and which would be useless against malicious packets with topologically correct IP addresses.

We have integrated a bandwidth optimization into standard neighbor discovery. The proposed optimization that we call *Compact Neighbor Discovery (CND)*, is a novel neighbor solicitation technique that uses Bloom filters. Multiple IPv6 addresses (bogus or real) in the access router's address resolution queue are compactly represented using a Bloom filter. By broadcasting a single neighbor solicitation message that carries the Bloom filter, multiple IPv6 addresses are concurrently solicited. Legitimate neighbor solicitation triggering packets are not denied service, since Bloom filters support membership queries. An on-link host can detect its address in the received Bloom filter and return its MAC address to the access router. Bloom filters yield a small false positive probability. Therefore in CND, the hosts in the target subnet send unnecessary neighbor advertisements at low rates in response to neighbor solicitation messages that solicit other nodes.

In this paper we have addressed the signaling impacts of ND-DoS, which we considered most important and least trivial. We focused our attention on the 802.11b model, although CND does not depend on MAC layer specifics. We have shown that a bandwidth gain around 40 can be achieved in all cells of the target subnet. This is a promising result. It shows that, with CND assistance, modern wireless MAC protocols can cope with serious ND-DoS attacks. We have shown by experiments that by reducing the number of neighbor solicitation packets, CND can also save access point CPU time. Hence faster MAC protocols may also profit from it, if access points (or bridges in general) have CPU limitations.

ACKNOWLEDGMENT

We are grateful to Alper Yegin and James Kempf for helpful discussions on ND-DoS during a summer internship in 2002 at DoCoMo Labs, CA. The ND-DoS flaw was unmasked by Alper Yegin who was also first to mention a stateful defense possibility and its potential problems in a loss of state scenario.

We would like to thank Venkat Anantharam for suggesting

$$c = \left\lceil \frac{J}{n} \right\rceil$$

Thanks Aurelien Francillon and Christoph Neumann for excellent discussions on many details of this paper.

Finally, we would like to thank the anonymous reviewers for helping improve the paper.

REFERENCES

[1] T. Narten, E. Nordmark, and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)," RFC 2461, IETF, December 1998.

[2] D. Plummer, "An Ethernet Address Resolution Protocol," RFC 826, IETF, November 1982.

[3] P. Nikander et al., "IPv6 Neighbor Discovery (ND) Trust Models and Threats," RFC 3756, IETF, May 2004.

[4] T. Aura, "Cryptographically Generated Addresses (CGA)," draft-ietf-send-cga-06, Internet Draft (work in progress), October 2004.

[5] J. Arkko et al., "SEcure Neighbor Discovery (SEND)," draft-ietf-send-ndopt-05, Internet Draft (work in progress), April 2004.

[6] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," RFC 3775, IETF, June 2004.

[7] B. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, July 1970.

[8] A. Conta and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification," RFC 2463, IETF, December 1998.

[9] L. Fan, P. Cao, J. Aldeida, and A. Broder, "Summary Cache: A scalable wide-area Web cache sharing protocol.," in *Proceedings of SIGCOMM'98 Conference*, October 1998, vol. 28, pp. 254–265, Corrected version available at URL: http://www.cs.wisc.edu/~cao/papers/summarycache.html.

[10] R. Rivest, "The MD5 Message-Digest Algorithm," RFC 1321, IETF, April 1992.

[11] S. Dharmapurikar et al, "Longest Prefix Matching Using Bloom Filters," in *Proceedings of SIGCOMM'03 Conference*, Karlsruhe (Germany), August 2003.

[12] M. C. Little et al, "Using bloom filters to speed-up name lookup in distributed systems," *Computer Journal*, vol. 45, no. 6, pp. 645–652, 2002.

[13] P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," RFC 2267, IETF, January 1998.

[14] S. Bellovin et al., "ICMP Traceback Messages," Internet Draft, work in progress, IETF, February 2003.

[15] Savage et al., S., "Practical Network Support for IP Traceback," Technical report uw-cse-2000-02-01, department of computer science and engineering, university of washington, http://www.cs.washington.edu/homes/savage/traceback.html.

[16] Computer Emergence and Response Team (CERT), "CERT Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks," http://www.cert.org/advisories/ca-1996-21.html, Last revised November 2000.

[17] Computer Emergence and Response Team (CERT), "CERT Advisory CA-1998-01 Smurf IP Denial-of-Service Attacks," http://www.cert.org/advisories/ca-1998-01.html, Last revised March 2000.