Off by Default!

Hitesh Ballani*, Yatin Chawathe†, Sylvia Ratnasamy†, Timothy Roscoe†, Scott Shenker‡
*Cornell University †Intel-Research ‡ICSI/UC Berkeley

I. Introduction

The original Internet architecture was designed to provide universal reachability; any host can send any amount of traffic (modulo congestion control) to any destination. This blanket openness enabled the Internet to adopt a single, globally routable address space. Unfortunately, today's less trustworthy Internet environment has revealed the downside of such openness—*every* host is vulnerable to attack by *any* other host(s). In the face of mounting security concerns, a primitive set of protective mechanisms (such as firewalls and NATs) that protect the host itself, but not the network leading to the host, have been widely deployed. In addition, the research community is busily producing proposals to address denial-of-service in a more comprehensive fashion [1], [2], [3], [4], [5], [6], [7], [8]. These proposals use various sophisticated architectures and approach the problem from many different perspectives. However, none of them take the simplest and most direct approach: allow each host to explicitly declare to the network routing infrastructure what traffic it wants routed to it.

In this paper, we propose such an approach, and investigate its feasibility. We describe an IP-level control protocol by which endhosts signal, and routers exchange, reachability constraints on different destination prefixes. A router may now forward a packet from host A to host B only if B has explicitly informed the network of its willingness to accept incoming traffic from A. In effect, we're proposing to flip the default constraint on host reachability from "on" to "off". Given current security woes, we believe this more conservative default is appropriate.

Yet it is important to preserve the opportunity for openness. The great strength of the existing "default-on" model is the flexibility it gives applications in their choice of communication models (client-to-server, server-to-server, peer-to-peer) which has been credited with enabling the variety of Internet applications we enjoy today. To preserve this flexibility, our protocol allows hosts to dynamically modify and inform the network of their current reachability constraints; *i.e.*, our conservatism extends only to the network's *default* behavior. On the face of it, requiring the network to dynamically maintain reachability information for every destination would seem to place an intractable burden on routers. Our feasibility analysis suggests that this is not necessarily the case and that a default-off Internet might well be a practical option.

We do not claim that such a default-off approach is sufficient or optimal. On the contrary, the general problem (control over host reachability) is a non-trivial one with a large design space and it's likely too early for any particular approach to claim the prize. Moreover, given the complementary tradeoffs between various solutions, it is quite likely that the "sweet spot" in the design space involves more than one approach. Nonetheless, we hope that exploring an extreme design point will better reveal (and stimulate discussion on) the different options and hence initiate a more principled approach to arriving at the ideal solution.

The remainder of this paper is organized as follows: we describe our goals and proposal for a default-off Internet in Sections II and III, present results from a simple feasibilty study in Section IV and finally discuss related work in Section V.

II. Design Goals and Challenges

We identify three key goals for a default-off network:

a) Off by default: Routers should not forward packets unless explicitly directed to do so by the destination host, in contrast to the current Internet where routers forward packets unless prevented by an operator-configured ACL rule. The off-by-default policy is thus similar to that of typical firewalls, but applied globally to the whole network. A direct consequence is that to receive unsolicited traffic, a host must now *proactively* inform the network of its willingness to do so. As [2] observes, this restriction of traffic to deliberately enabled communication paths raises the bar for attacks on hosts that are not reachable.

b) Explicit expression: Hosts must have a way to explicitly and unambiguously express their reachability, unlike NATs and firewalls which implicitly control a host's reachability by virtue of being in the data path.

c) Flexible constraints: A host should be able to dynamically regulate its reachability along multiple dimensions: who gets to send a host traffic, when, what type (*i.e.*, protocol, port) of traffic, how much, etc. This flexibility is essential to preserve the rich communication models possible today while respecting the administrative boundaries that often define reachability. In this paper, we discuss a limited number of dimensions—temporal (when is a host reachable), spatial (which hosts/prefixes can reach a host, on what ports, and with what protocol), and scope (where a host's reachability is advertised).

To achieve these goals, we propose that hosts signal their first hop routers with their intent to receive packets from other hosts. Routers propagate these as *reachability advertisements* and use this information to forward or drop packets. This naïve approach faces two obvious challenges:

Scalability: If routers were required to maintain reachability state for every host in the network network, our scheme would not scale. We address this in two ways. First, since hosts that are "off" do not issue reachability advertisements and incur no additional state at routers, we maximize the number of hosts that can be treated as "off". To do so, we borrow from Handley and Greenhalgh[2] and arrange that a host that only receives traffic in *response* to its own traffic need not be "on".

Second, we allow routers to *aggregate* reachability advertisements according to available memory. While legitimate packets are always forwarded, aggregation introduces a tradeoff between

the network's effectiveness at limiting unwanted traffic and the size of reachability state needed at routers. More state means less aggregation, and hence unwanted traffic is dropped nearer the source. In other words, we allow the enforcement of default-off policies to be best-effort.

Network dynamics: A naïve implementation of Default-off would couple reachability advertisements with the routing protocol, so that a router advertises a route only if the corresponding host(s) have requested that they be reachable. However, since we expect the reachability of hosts in a domain to be much more dynamic than routes to the domain, this would lead to undesirable routing dynamics. Instead, we avoid the issue of routing dynamics altogether by decoupling reachability maintenance from route computation.

Specifically, routes are computed as they are today, and reachability information for hosts in the prefix is stored in an extension to that prefix's entry in the router forwarding information base (FIB). This keeps the complexity of FIB updates on the order of the number of routable prefixes rather than the much larger number of (possibly aggregated) reachable hosts.

We now present a design for a default-off network that addresses these goals and challenges.

III. Design Description

In our straw-man design for Default-off, when a router receives a packet, it performs a normal route lookup to locate the routing entry for the destination prefix and then checks the associated reachability state, dropping packets that are not explicitly allowed by a reachability entry.

A host explicitly signals reachability to its first-hop router. Routers exchange this state via a *reachability protocol*; this can in some cases be piggybacked on route advertisements. This protocol could be run at both the intra- and inter-domain level. In this paper, we describe and evaluate only the inter-domain scenario; the intra-domain case follows straighforwardly. Thus, we assume border routers exchange *reachability state* for their prefixes with neighbors in other ASes. This state indicates which hosts in a prefix are reachable, and under what constraints. Like BGP, the protocol is incremental, but unlike BGP exchanges between routers are periodic.

"Off" hosts, like those behind NATs, can only receive packets in response to traffic they initiate. As noted above, we adopt the design in [2]: when an "off" host sends a packet, the domain-level path from the client to the server is recorded in the packet header; when the server responds, the packet is routed along the reverse path to the client. The existence of this source route is enough for routers to verify the connection and no router state is needed for such client traffic.

We do not specify in this paper how hosts decide on their reachability, though this should *not* be directly controlled by existing network APIs (e.g., listening on a socket should not automatically make the host reachable). In practice, some combination of administrator policies and user interaction in the host will determine reachability.

We now describe various features of Default-off in detail.

A. Expressing Reachability

Hosts signal reachability to routers by providing the host IP address, a list of *reachability constraints* or *RCs*, and a propagation *scope* (described below). To allow for aggregation of addresses, we extend the IP address to a prefix in reachability advertisements. The general form of advertisements is thus:

[*prefix, prefix-length,* { *RC, RC,*... }, *scope*]

Our current proposal uses two levels of constraints: $\mathbf{RC}_0$ constraints are 3-tuples of destination IP address, protocol, and port, and are used by hosts that wish to be "on" to any and all sources. $\mathbf{RC}_1$ constraints are 4-tuples and are used by hosts that wish to be selectively "on" to specified hosts; they additionally include a list of IP addresses of such sources. Clearly, this initial scheme can be extended. For example, a simple enhancement would include source ports to be specified, or particular flow rates.

The *scope* of an advertisement avoids needless propagation of state when a host wishes to restrict its reachability along topological or administrative boundaries (*e.g.*, a department's internal file server). A simple solution defines a scope of (router or AS-level) hop count that bounds the topological extent of advertisement propagation. Alternatively, one could encode the set of ASes or subnets through which the advertisement can be advertised.

In the limit, scoping could restrict the propagation of a host's reachability advertisement along only those parts of the network that lie on the path from acceptable sources for that destination. However, achieving such fine-grained scoping with full generality is a non-trivial challenge (akin in some sense to scalable multicast routing) and one we leave open for future research. For simplicity, this paper assumes all reachability adverts are globally propagated; incorporating scoping would only improve our performance results.

In addition to scoping, we provide *temporal control* by using standard soft-state techniques to determine the lifetime of a host's reachability advertisement. A host periodically beacons its current reachability, and immediately signals changes to its reachability. To turn "off" altogether, the endhost either sends an explicit withdrawal to its local router or simply ceases its periodic updates and waits for expiry.

B. Encoding Reachability

Encoding the reachability of each host as a straightforward list of constraints clearly leads to excessive router state. Instead, we encode the reachability constraints using Bloom filters [9], trading space for processing in routers. Note that because Bloom filters return false positives, hosts that are "off" may be reported as being "on" and packets to such destinations might arrive at the destination's router before being dropped.

A domain's access router uses k globally known hash functions to encode a host's reachability constraints, using different filters for different constraint types: all constraints of type RC_0 for the host are represented by a bloom filter that encodes all RC_0 three-tuples {destination IP address:destination port:protocol}, and similarly for RC1 constraints. The Bloom filter size must be chosen judiciously to keep probability of false positives at an acceptable level.

C. Aggregating Reachability

To scale in reachability state, a Default-off router aggregates advertisements to fit its memory limitations. There are two levels at which to apply such aggregation. First, we can merge multiple advertisements into one by bitwise OR-ing the corresponding

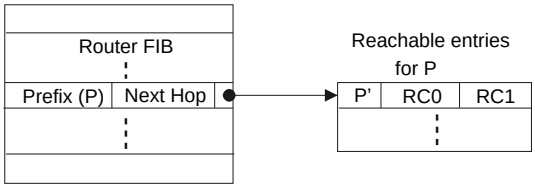


Fig. 1. Conceptual structure of the FIB in a Default-off router. Here, P' $\in$ P

bloom filters, and setting the key for the merged advertisement to the longest common prefix across the aggregated advertisements. Second, we can reduce the size of the bloom filters within a single advertisement, for example shrinking filters by a factor of two by splitting them and performing a bitwise OR of the halves.

As advertisements propagate through the network, each router combines and possibly aggregates new and existing advertisements. This results in higher false positives, meaning more unwanted traffic is allowe further into the network, towards the destination. Unwanted traffic is dropped when it encounters a sufficiently unaggregated filter.

Which advertisements should be aggregated depends on resource constraints at the router, false positives induced by aggregation, and/or the aggregator's relationship with the domain whose reachability state is being aggregated. For example, advertisements from customers might have higher priority than those from a peer provider. Accepting unaggregated advertisements might even be part of SLAs between customers and providers. Our evaluation below use a simple aggregation rule: peer advertisements are always aggregated before customer advertisements; following this, entries to be aggregated are selected at random. Clearly, more sophisticated rules can potentially improve our results. An interesting open question is whether there exists an aggregation rule that achieves an optimal tradeoff between state consumed and false positive rate while respecting policy constraints.

Our proposal effectively turns the network into a global firewall, while the aggregation of advertisements implies that the protection the network offers to a domain drops as the distance from the domain increases. We analyze the quality of this protection in Section IV-B. At the same time, there is an opposite trade off between the protection offered and the extent to which the network is exposed to the dynamics of endhost reachability. The greater the protection, the deeper into the network reachability advertisements must propagate and hence the network is subject to more reachability dynamics. In particular, the time taken for host to transition from "off" to "on" depends on how far its advertisement must propagate before encountering an aggregated reachability entry that already (due to aggregation) had the host marked down as reachable. We analyze this tradeoff in Section IV-C.

D. Packet Forwarding

In addition to performing the standard longest-prefix match before forwarding packets, a router must perform a reachability check. On receiving a packet, a router first checks whether the destination is a path-based address. If so, it immediately forwards the packet based on the path-address. Otherwise, it performs a regular IP lookup in its FIB to locate the next hop and the reachability entry for the destination IP address (see Figure 1)[1]. If no such reachability entry exists, the packet is dropped. Otherwise, the router checks the packet's destination IP address, port and protocol 3-tuple against the reachability entry's RC_0 filter. If the Bloom filter returns a hit, the packet is forwarded otherwise the packet's destination address, port, protocol and source address 4-tuple is checked against RC_1. If that check too fails, the packet is dropped.

E. Discussion

Before proceeding with the evaluation of our design, we briefly note some of the larger questions left unaddressed in our discussion. The first has to do with securing the reachability protocol itself. Because we overlay reachability over existing routing protocols, Default-off inherits the hop-by-hop trust model of current routing and the deployment of more secure routing proposals [10] would apply directly to our scheme too. Similarly, while malicious end hosts may advertise bogus reachability adverts, the damage they can cause should be limited because a router is always free to not aggregate a particular host's advertisement (if, for example, doing so would increase the false positive rate of the RCs) or to simple "upgrade" a host's advertised reachability. Precisely proving the extent of possible damage is however a topic for future work.

Deploying Default-off also merits closer scrutiny in terms of both mechanism and incentives. Note that Default-off should be incrementally deployable by individual ISPs; an ISP can independently deploy Default-off within its local domain with immediate benefit to its direct customers. Indeed, many industry solutions for DoS protection are already on this trajectory although their solutions are based on special-purpose middleboxes [11]. Also open, are the engineering details of how one might best incorporate the Default-off mechanisms into the control and data plane of routers.

Finally, an interesting open question has to do with the interplay between Default-off and the enforcement of organizational policies. On the one hand, default-off allows end users (presumably in conjunction with their administrators) to independently regulate their reachability but on the other our proposal for an explicit signalling of intended reachability appears conducive for systematic policy enforcement.

IV. Feasibility study

Our design from the previous section raises two main performance questions:

- How effective is Default-off at limiting unwanted traffic?
- Can the design handle the dynamics of hosts turning on/off?

This section tries to address these questions. We stress, however, that our results are merely an initial sanity check of the feasibility of our proposal; we defer a more comprehensive evaluation to future work. We start with a brief description of our methodology in Section IV-A and then explore the above questions in Sections IV-B and IV-C respectively.

[1]Locating the reachability advertisement involves doing a longest prefix lookup on the reachability entries associated with the destination prefix. Given that the number of reachability entries is likely very small, we do not imagine the lookup and updating will be expensive, and in fact could likely be trivially handled by storing the prefixes for the reachability entries in TCAM.

Name	Remark	Number
Stub-AS	an AS with no customers	11232
Regional ISPs	an AS with customers and degree < 11	1475
Core-ISPs	rest of the ASs	695

TABLE I

Three categories of ASs based on the number and relationship with neighbors in the AS topology

A. Methodology

Because simulator limitations prevent us from simulating Default-off on a realistic router-level Internet graph, we choose to simulate it over the Internet AS-level topology maps from Subramanian *et al*. [12]. These topologies are annotated with inter-AS relationships (customer-provider or peers) and hence our simulations respect policy in the propagation of routing and reachability advertisements. Table I summarizes the key statistics of our topology, the details of which can be found in [12]. We set the total number of prefixes on the Internet (**P**) to 200,000 [13] and assign these to ASes in our topology.

The crucial usage parameter is **H**, the number of hosts per prefix that signal their intent to be reachable. As described in [2], there are two kinds of reachable hosts: servers and peers. Measurements of P2P traffic in a tier-1 ISP backbone [14] indicate ∼2-3% of observed flows can be attributed to P2P applications from which we approximate that 2-3% of Internet hosts act as peers at any given time.[2] With 600M hosts on the Internet [16], this leads to a total of 6-9M peers or 30-45 peers per prefix. We assume that the number of servers per prefix is small compared to the number of P2P hosts and hence set H=45, the high end of the P2P estimate. As we will see, our results are not very sensitive to slight variations in H.

The crucial technology parameter is the amount of router memory (**T**) available in the data plane to store reachability state. Since our simulations are at the AS-level, not the router level, we cannot accurately model the state held by each individual router and instead adopt two simplified (but hopefully informative) models. In the first (**model 1**), we assume that each domain has a single border router. This is the same as assuming that each border router in the domain holds the same state and has the same amount of available memory. We also assume that this router's available memory T is proportional to the total number of prefixes P; T $= \alpha$ P for some α. Most of our simulations use $\alpha = 3$.

In the second (**model 2**), we merely assume that each AS has sufficient state so that it never needs to aggregate reachability state for its customer prefixes. This appears reasonable since border routers within a single AS are attached to different sets of customers and hence no single router has to hold unaggregated reachability state for all customers of the AS. As mentioned earlier, non-aggregation of customer reachability state may become a standard part of SLAs, and later we argue that this is economically feasible. For this model, when the immediate customers use less than T memory, the rest is devoted to other prefixes. When the immediate customers consume more than T memory, reachability state for all the other prefixes is completely aggregated to one entry each.

[2]Note that this is very likely an overestimate because, in most P2P applications [15], a single peer will initiate multiple flows for a single transfer.

B. Protection

Default-off scales by aggregating reachability advertisements as dictated by available memory at a router. Aggregation introduces false positives, and allows traffic to make some progress towards "off" destinations before being dropped. As described in Section III-A, a reachability advertisement is composed of two components: the prefix (and the prefix length) and the reachability constraints (**RC**). Aggregation of an advertisement can lead to false positives in both components. To factor out the effect due to each, we first consider reachability advertisements as comprised of only prefixes (this is equivalent to merely distinguishing between "on" and "off" hosts) and then consider adding on reachability constraints.

1) Aggregating prefixes: Here, each "on" host's advertisement only includes its IP address represented as a /32 prefix. Using the setup described in Section IV-A, we simulate the propagation (with aggregation) of these reachability advertisements. Once the reachability protocol converges, we route a packet from a random source to a destination host that is "off" (i.e., has not initiated a reachability advertisement) and observe the location at which the packet is dropped. We repeat this for 6 million source-destination pairs.

For models 1 and 2, with $\alpha = 3$ and H=45, Figure 2(a) plots the CDF of the fraction of dropped packets versus the distance (in AS hops) between the destination and the point at which the packets were dropped. To better calibrate our results, we plot four bounding cases: ***At-Source (SRC)***: All unwanted packets are dropped at the source. Note that this is effectively the CDF of path lengths.

Near-Source (N-SRC): All unwanted packets are dropped at the core ISP closest to the source (along the source-to-destination path). This is intended to represent the boundary between the source and the core. Dropping packets here effectively shields both the network core and the destination's access path from unwanted traffic.

Near-Destination (N-DST): All unwanted packets are dropped at the core ISP nearest to the destination. This represents the boundary between destination and core. Here the destination but not the core are shielded from unwanted packets.

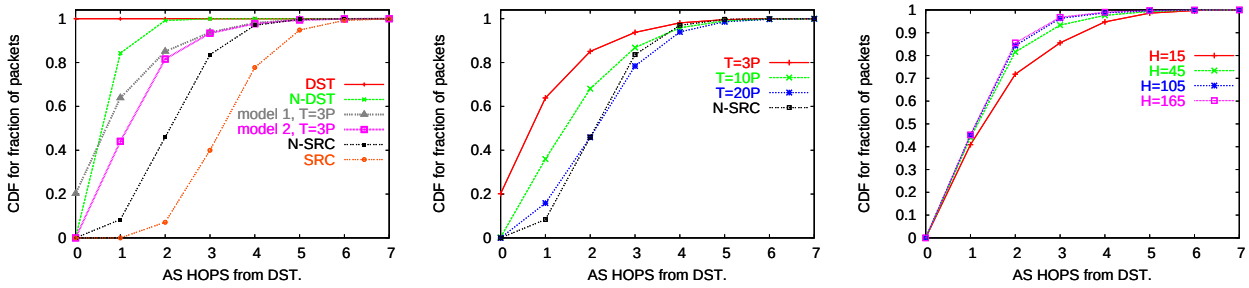
At-Destination (DST): All unwanted packets are dropped at the destination (akin to firewall-based protection).

Even with the more conservative model 1, Default-off can drop most (>80%) of unwanted traffic within the network's core, well away from the destination. With model 2, ∼60% of the packets are dropped 2 or more AS hops away from the destination and the destination's peering link is never choked.

Figures 2(b) and 2(c) show the effect of varying T and H respectively. As can be seen, increasing T leads to better protection while the system scales well with increasing H.

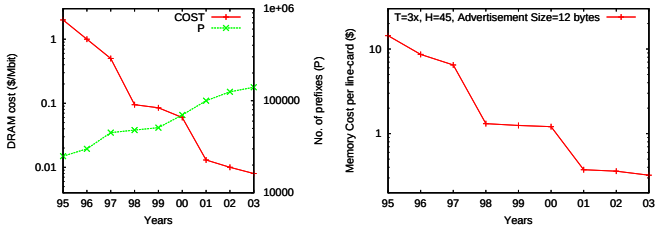
2) Aggregating Bloom Filters: Our simulations so far evaluated the protection offered by the reachable prefix field. The use of bloom filters encoding reachability constraints (RC_0 and RC_1) offer better protection for increased state at routers. Here, we estimate the amount of additional state needed, and then compute the approximate cost of the total state per router.

Instead of assuming "on" hosts are reachable on all ports by everybody, we now assume that each "on" host specifies



(a) (b) (c)

Fig. 2. CDF for the fraction of packets that are dropped a given number of AS hops from the destination with different protection schemes, varying T (H=45, model 1) and varying H (T=3P, model 2)



(a) (b)

Fig. 3. DRAM Cost/Mbit [17], FIB size [18] and the cost of reachability state per line card over the years

5 addresses it wants to be reachable from[3]. This is encoded in RC_1. We also assume that the end-site's reachability router chooses the size of the bloom filter to encode RC_1 so as to ensure that the false positive ratio is less than 1%. This would require ∼10 bits for each constraint inserted into the bloom filter and hence, yield a RC_1 of size 50 bits and an advertisement of ∼12 bytes. Maintaining 600,000 such reachability entries in the routers (i.e. T=3P) would lead to a reachability state of ∼7 MB in the forwarding plane.

Given the state required at typical ISP access routers for tasks such as packet classification and relative to previous proposals that require upto 1Gbit DRAM for FIBs [19], the memory requirements for Default-off appear modest and should not face significant technical barriers. On the contrary, at current prices this would cost about $0.52 per line card for DRAM, and $150 per line card for SRAM. Moreover, technology trends are with us; memory costs are dropping far faster than the rate at which the number of prefixes are rising (Figure 3(a)), and hence the total costs should only decrease over time (Figure 3(b)).

C. Dynamics

Another trade-off introduced by our proposal is the time it takes for a host to turn on versus the update load imposed on the routers. The turn-on time is directly propotional to the distance the advertisement must traverse and the interval at which routers exchange reachability information. For example, the model 2 results shown in section IV-B.1 imply that exchanging reachability state at an interval of 20 seconds would yield an average turn-on time of ∼36 seconds, which seems reasonable. The time to turn off is less critical because the destination stops receiving packets as soon as its first-hop router is notified of the change in the host's reachability. As the corresponding reachability advertisement moves upstream, the drop point moves further away from the destination.

Given that a 20 second inter-advertisement interval leads to an acceptable turn-on time, the question is whether the load this imposes on routers is manageable. Note however that because reachability is computed separately from routing, a reachability event (turning "off" or "on") does not involve recalculating routes and updating the FIB but only involves a longest-prefix match to locate the reachability state for the prefix and then updating it. As mentioned earlier, the reachability state for a prefix is smaller than the router FIB and hence, easier to update. In the worst case scenario, each prefix in the Internet can have at least one reachability event (host turning "on" or "off") every interval, leading to an update rate of 10,000 per second. Existing data structures for FIBs can handle 10,000 routing updates [20], and hence routers could certainly handle the lighter load of updating the reachability database.

Moreover, the fact that unwanted packets in a default-off network can traverse half the network before getting blocked implies that reachability advertisements need to traverse just the other half. For example, figure 2(a) (model 2) shows that 40% of the advertisements only need propogate to the first AS hop and 80% of the advertisements only need to propogate through to two AS hops. Hence, the aggregation of the reachability advertisements reduces the impact of reachability dynamics.

V. Related Work

With regard to controlling traffic to a host, there has been no shortage of proposals from the networking research community. In this paper, our goal has been to investigate the feasibility of a default behavior of non-connectivity in the network. Our approach uses a simple control protocol by which hosts can tell the network what traffic they do want routed to them. The result is a network in which a host's reachability is: (1) flexible, (2) explicitly communicated to the network and, (3) off by default and hence proactively controlled. In what follows, we briefly relate the various prior proposals to Default-off in terms of both its mechanism and its properties but stress that it is difficult to categorically compare across proposals as they vary widely in intent and means.[4] Table II summarizes our discussion.

[3]This implies that each host has 5 reachability constraints; a host saying that it wants to be "on" to all sources for a particular destination port and protocol (RC_0) introduces just one constraint and hence, requires less state

[4]For example, the capabilities-based approach by Yang *et al.* targets more comprehensive protection than (say) pushback, firewalls, or Default-off but requires correspondingly more heavyweight mechanisms.

Proposal	Mechanism	Proactive vs. Reactive	Default	Explicit vs. Implicit	Flexibility
Pushback, AITF	filters	reactive	ON	explicit	yes
Capabilities, SIFF	capabilities	reactive	sig. channel: ON, cap channel: OFF	explicit	yes
i3, Mayday, SOS, etc.	overlays	proactive	overlay: OFF, IP level: ON	explicit	yes
Handley et al.	multiple address spaces	proactive	OFF	implicit	no
Firewalls	middlebox	proactive	IP router: ON, at firewall: OFF	implicit	no
Default-Off	IP reachability protocol	proactive	OFF	explicit	yes

TABLE II

Default-Off properties in comparison to different proposals that allow control over host reachability

A first class of proposals might be termed "reactive": connectivity is still on by default, but in the event of a host detecting that it is under attack it can request that the network prevent the traffic arriving at the host. Perhaps the earliest exposition of reactive DoS defence was Pushback [21], [1] with AITF [8] being a recent refinement of the basic idea. In some sense, Default-off inverts this approach in that default reachability is off and hosts must proactively clear the (unsolicited) traffic they wish to receive.

Alternatively, several researchers have proposed using overlay networks to control the traffic to a destination [22], [6], [23]. These proposals work by effectively requiring that all traffic to the destination be routed through the overlay where sophisticated defenses are easily deployed. Like Default-off, these proposals support explicit and flexible control over host reachability but, because they operate above IP, cannot protect a destination whose IP address is known to attackers. By contrast, Default-off is embedded in the existing routing infrastructure and hence directly controls the network-layer path to the destination.

An interesting class of solutions employ the idea of capabilities [4], [3], [24] to control access to hosts. Under this approach, sources request the destination for permission to send packets via signalling carried in a separate class of traffic. Consequently, the signalling channel must offer open (i.e., default on) access and hence the authors propose the use of rate limiting with fair queuing to secure this open channel. The result is a very different design with different properties. Relative to Default-off, capabilities allow more sophisticated and fine-grained control over connectivity but also incur corresponding greater complexity in both implementation and management. A systematic exploration of the tradeoffs (and possible middle ground) between the approaches is a topic for future work.

In a provocative paper, Handley and Greenhalgh [2] offer a radical solution to the DoS problem: classify each host as either a server or a client, and allow only servers to recieve unsolicited packets. Default-off could be viewed as a relaxation of the Handley and Greenhalgh scheme, which retains its inherent conservatism (and the technique of using source routing for "off" clients) but allows hosts flexibility in their reachability constraints. There is also a distinction to be drawn in terms of mechanism: while Default-off pushes control over reachability into the routing layer, Handley and Greenhalgh's proposal operates at the addressing layer by defining different address spaces for clients and servers.

In conclusion, we compare our work with the most prevalent security mechanisms: firewalls. Default-off takes the basic firewalling notion of blocking all traffic except that explicitly whitelisted, and extends it to be more dynamically controllable by hosts, as well as propagating the whitelists far into the network. Note moreover, that when viewed globally, a firewalled Internet leaves the default "on" at routers only turning it "off" at the destination host's firewall (if one exists at all!). Given current security woes, we believe the more conservative "default-off everywhere" architecture is more appropriate.

REFERENCES

[1] J. Ioannidis and S. M. Bellovin, "Implementing pushback: Router-based defense against DDoS attacks," in *Proc. of Network and Distributed System Security Symposium*, 2002.

[2] M. Handley and A. Greenhalgh, "Steps Towards a DoS-resistant Internet Architecture," in *Proc. of ACM FDNA Workshop*, 2004.

[3] A. Yaar, A. Perrig, and D. Song, "SIFF: A Stateless Internet Flow Filter to Mitigate DDoS Flooding Attacks," in *Proc. of IEE Security and Privacy Symposium*, 2004.

[4] T. Anderson, T. Roscoe, and D. Wetherall, "Preventing Internet denial-of-service with capabilities," in *Proc. of 2nd ACM Hotnets*, 2003.

[5] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet indirection infrastructure," in *Proc. of ACM SIGCOMM*, 2002.

[6] D. Andersen, "Mayday: Distributed filtering for internet services," in *USITS*, 2003.

[7] M. Stiemerling, J. Quittek, and T. Taylor, "MIDCOm Protocol Semantics," June 2004. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-midcom-semantics-08.txt

[8] K. Argyraki and D. R. Cheriton, "Active Internet Traffic Filtering: Real-Time Response to Denial-of-Service Attacks," in *Proc. of USENIX Annual Technical Conference*, 2005.

[9] B. H. Bloom, "Space/Time trade-offs in hash coding with allowable errors," *Communications of ACM*, vol. 13, no. 7, July 1970.

[10] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. H. Katz, "Listen and Whisper: Security Mechanisms for BGP," in *Proc. of NSDI*, 2004.

[11] "Riverhead DDoS mitigation." [Online]. Available: http://www.riverhead.com

[12] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz, "Characterizing the Internet Hierarchy from Multiple Vantage Points," in *Proc. of IEEE Infocom*, 2002.

[13] "Route Views Project Page." [Online]. Available: http://www.route-views.org/

[14] T. Karagiannis, A. Broido, M. Faloutsos, and kc claffy, "Transport Layer Identification of P2P Traffic," in *Proc. of Internet Measurement Conference*, 2004.

[15] B. Cohen, "Incentives Build Robustness in BitTorrent," in *Proc. of Workshop on economics of Peer-to-Peer Systems*, 2003.

[16] "ISC Domain Survey," Jan. 2005. [Online]. Available: http://www.isc.org/index.pl

[17] "ICKnowledge Survey," 2003. [Online]. Available: http://www.icknowledge.com/economics/productcosts4.html

[18] "Geoff Hustons's BGP Report," 2005. [Online]. Available: http://bgp.potaroo.net/

[19] S. Keshav and R. Sharma, "Issues and Trends in Router Design," *IEEE Communications Magazine*, May 1998.

[20] W. Eatherton, G. Varghese, and Z. Dittia, "Tree bitmap: hardware/software ip lookups with incremental updates," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 2, 2004.

[21] R. Mahajan, S. Bellovin, S. Floyd, J. Vern, and S. Shenker, "Controlling high bandwidth aggregates in the network," *ACM SIGCOMM Computer Communications Review, 32(3)*, July 2001.

[22] A. Keromytis, V. Misra, and D. Rubenstein, "SOS: Secure Overlay Services," in *Proc. of ACM SIGCOMM*, 2002.

[23] K. Lakshminarayanan, D. Adkins, A. Perrig, and I. Stoica, "Taming IP packet flooding attacks," in *Proc of 2nd ACM Workshop on Hot Topics in Networks*, 2003.

[24] X. Yang, D. Wetherall, and T. Anderson, "A DoS-limiting Network Architecture," in *Proc. of ACM SIGCOMM*, 2005.