



Pioneering Technologies  
for a Better Internet

Cs3, Inc.

5777 W. Century Blvd.  
Suite 1185  
Los Angeles, CA 90045-5600

Phone: 310-337-3013  
Fax: 310-337-3012  
Email: info@cs3-inc.com

# *A Fair Service Approach to Defending Against Packet Flooding Attacks*

**Patented**

Donald Cohen  
K. Narayanaswamy

## **Abstract**

Using research funding from the Defense Advanced Research Projects Agency (DARPA), and a grant from the California Technology Investment Partnership (CAL TIP), Cs3 is in the process of implementing a defense (on top of both IPv6 and IPv4) against packet flooding and related distributed denial of service (DDoS) attacks on the Internet. The ideas underlying this defense are now patent pending. The approach is amenable to implementation on different router platforms. It is incremental and scalable, and shows the potential to resolve this important network security problem.

## **CONTENTS**

- 1. Background: Distributed Denial of Service Attacks
- 2. Technical Problems Underlying Packet Flooding Attacks
- 3. Existing Solutions and Their Inadequacies
- 4. Proposed Changes to Infrastructure
  - 4.1. The Fair Service Approach
  - 4.2. Cooperating Neighborhoods and "Visible Sources"
- 5. Processing at the Destination
  - 5.1. Expected and Unexpected Packets
  - 5.2. Defenses Against Other Related Attacks
  - 5.3. Rate Limiting
- 6. Analysis of the Defense
  - 6.1. Assessment of Incremental Benefits
  - 6.2. Effectiveness of the Fair Service Defense
  - 6.3. Limits on Cooperation
- 7. Implementation Status
- 8. Endnotes

---

# 1. Background: Distributed Denial of Service Attacks

The Internet is increasingly vulnerable to a new kind of security problem - Distributed Denial of Service (DDoS) attacks. One common type of DDoS attack is packet flooding, where the victim's communication bandwidth is filled with traffic from the attacker (or an army of "slaves" over which he has gained control), thereby preventing the communication the victim really wants. These attacks can be devastating:

- The most infamous of the DDoS attacks occurred in February 2000, when it has been estimated that a 4-hour attack on popular sites like CNN, Yahoo, and eBay caused a total of \$1 billion in economic impact (*Source: Yankee Group*).
- More recently, during the week of January 22, 2001, Microsoft's vast site was shut down for hours by a DDoS attack, causing them lost revenue and tarnishing their reputation.

[CERT](#), a federally funded computer and network security response team, and the FBI have both issued several pointed advisories about the potential for [increases in denial of service attacks](#) and the growing threat that these attacks pose to e-business and e-commerce.

With funding from DARPA, the leading-edge research sponsor within the US Department of Defense, Cs3, a small R&D company in Los Angeles, has been investigating distributed network intrusion detection and response techniques and tools. Cs3 has developed an innovative solution to the denial of service problem that is scalable, reliable, and incrementally implementable within large-scale networks. With the help of follow-on DARPA funds and a CALTIP grant, implementation of this defense has been undertaken. The implementation works with both IPv4 and IPv6. A patent application for this defense has also been filed.

This white paper describes the Cs3 solution to distributed, coordinated denial of service attacks based on packet flooding.

## 2. Technical Problems Underlying Packet Flooding Attacks

The primary objective of the Cs3 defense is to thwart packet flooding attacks, where an attacker tries to disrupt the victim's communication by using up all of his bandwidth. A secondary objective is to defend against a related class of attacks where the attacker tries to use up some other resource, such as http (web) service.

Most organizations think of security as a characteristic of a particular site. This view may have merit for some problems, such as intrusion detection and virus protection, but a site cannot unilaterally defend itself against packet flooding DDoS attacks. In this case much of the damage is already done before the site can remedy the situation. In particular, the packets that the site wants to get from other places (such as its customers) do not arrive due to congestion in the network. This problem has to be fixed in the network that delivers packets to the victim.

In today's Internet, when a site realizes that it is under attack the only solution is to physically contact the places that are forwarding the attack packets and ask them to stop. Clearly it would be worthwhile to automate this process. However the first step is to find out what those places are. In the current infrastructure, the victim can only tell which ISPs (if he has more than one) are forwarding the attacking packets to him. He has to ask those ISPs to figure out who is forwarding the packets to them, and so on. Clearly, this step could also benefit from automation.

Yet another problem is that it may not even be obvious that a packet flooding attack is in progress. Even in principle, there is no defense against an attack that cannot be distinguished from an overload of normal traffic. If an attack consists of traffic that is easily distinguished from the normal traffic, then the distinguishing characteristics can be used to filter out the attack traffic. This suggests that attackers will try to send traffic that looks normal. Of course, if the attacker really acts like a normal user then his attack will not have much effect on the service he is trying to deny to other users. In order to overload the victim he has to act like a larger number of normal users than the victim can handle.

In this case, the attacker can be distinguished from normal users by the rate of his communication. But that only works if the victim can accurately identify the source of a packet -- an operation that is not supported by the Internet today. The attacker can forge different source addresses on a large number of packets, thereby making them appear to come from a large number of different places. There are also other ways to make attacks appear to be large amounts of normal activity. We will discuss some of these later. For examples of popular attacks the interested reader may wish to see:

- [The "trinoo" attack program](#)
- [The "tribe flood network" attack program](#)
- [The "stacheldraht" attack program](#)

### 3. Existing Solutions and Their Inadequacies

At present there is no deployed and fully automated solution to this problem. Traditional network security products, such as firewalls and intrusion detection systems, as we point out above, do not address this particular security problem. There are numerous "best practices" procedures that site administrators can undertake to minimize the impact of DDoS attacks. Examples of these actions include the following:

- [CERT's advisories](#) specifically about DDoS attacks. CERT suggests simple, common-sense approaches -- for example, that every installation should protect its own machines to prevent them from being used to attack third parties.
- [WWW Security FAQ: Securing Against Denial of Service Attacks](#)
- [Cisco's recommended measures](#) (both forensics and preventive) in reaction to DDoS attacks

While the above are certainly useful to some degree (to date, we must mention, a very small degree), they do not constitute a practical, reliable, automated defense against denial of service attacks.

The norm at critical sites is that, once an attack is noticed, manual procedures are carried out to communicate with upstream routers and their owners to isolate the traffic coming from the attacker. As we will see, Cs3's approach can be viewed, in part, as automation of some of these manual steps.

Finally, a number of recent startup companies have advocated an approach to the DDoS problem that we refer to as "Smart Filtering". These companies recognize that it is too late to solve the problem when the packets arrive at their destination. While we are not familiar with all the details of these proprietary approaches (because they have not been published), it seems that these approaches work through intelligent, rule-based analysis of patterns and rates of the traffic flowing through ISPs. The problem is that this approach can, at best, only recognize attacks that have been seen and analyzed before. The result is likely to be similar to what we see today in virus scanning software. The defenses are always trying to keep up with (and always a little behind) the attackers. Further, any analysis that makes use of the contents of packets is likely to fail as encryption becomes more widespread.

Cs3's fair service approach, which we describe next, is not susceptible to these problems. It is conceptually simpler and can be defeated only if the attackers control a considerable part of the infrastructure (in which case NO defense is adequate).

## 4. Proposed Changes to Infrastructure

The packet flooding problem cannot be solved locally. It requires cooperation among many machines in the network. The Cs3 solution is divided into two parts. One is in the network infrastructure that forwards packets from source to destination. The other is at the endpoints of the communication and is discussed below.

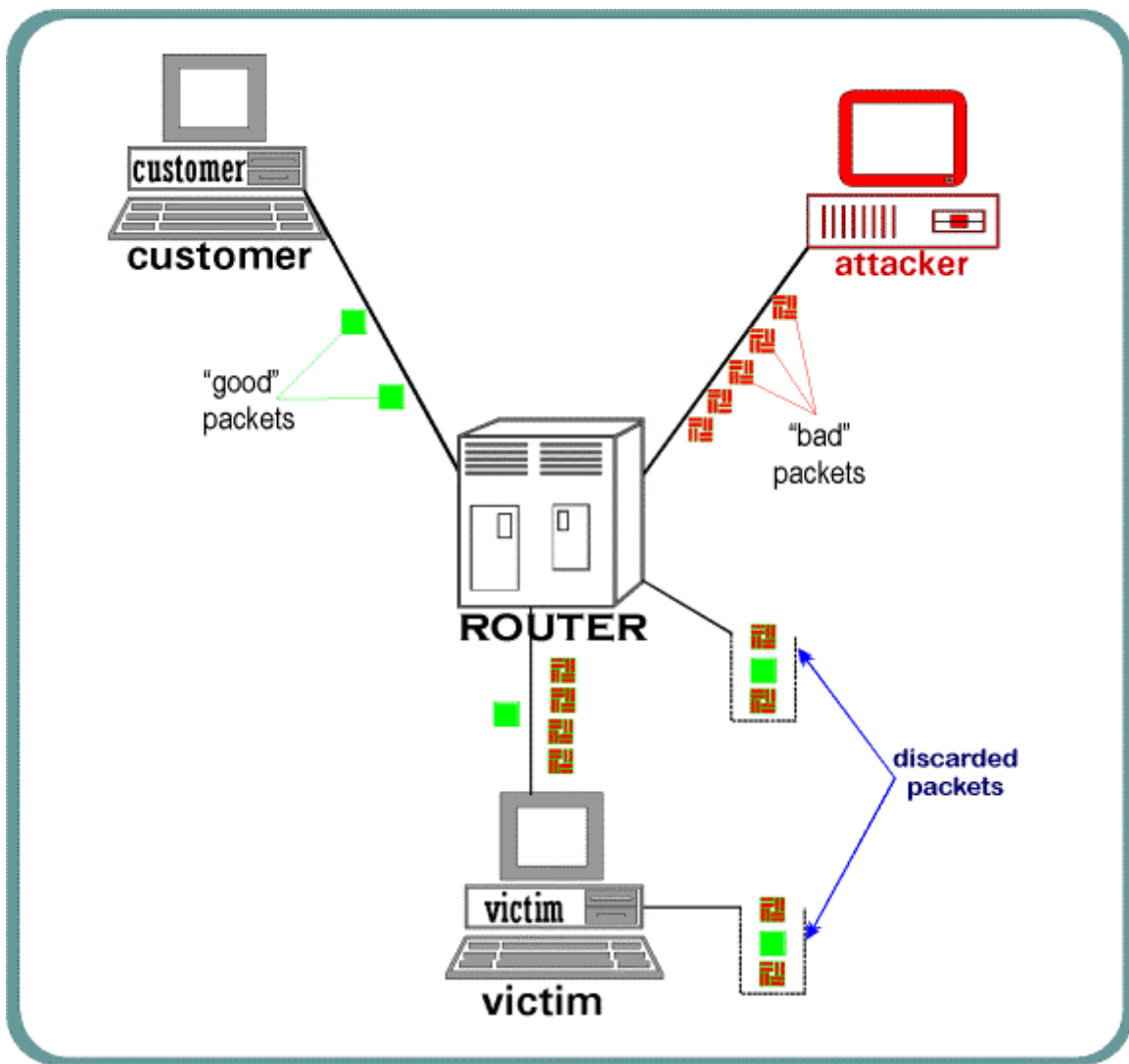
### 4.1. The Fair Service Approach

Our goal is not actually to prevent the attacker from sending his traffic, but to prevent that traffic from interfering with that of other users. The attacker wants to claim a very large part of a shared resource, in this case communication bandwidth to the victim. Our approach is to allocate that resource in a fair manner among the users requesting service. The attacker can therefore claim as much as a normal user, but he cannot exclude other users.

We expect legitimate communication to be "well-behaved" in the sense that a given sender, S, never sends more than a small amount of data to a given receiver, R, without R indicating (by return communication) a willingness to accept more. Most communication in the Internet follows the TCP protocol, which makes it well behaved in this sense. In general, users who communicate in a well-behaved manner end up requesting "reasonable" amounts of service. Attackers tend not to be well-behaved and request "unreasonable" amounts. Fair service will tend to punish the unreasonable behavior.

Note that services requiring high bandwidth for extended periods on demand are really incompatible with public networks. That is precisely because they are acting like attackers. It makes sense to use such services in a private network, or to use them in a public network only when demand is low.

The most straight forward service allocation mechanism, often called "first come first served", is also the most common implementation of the Internet's current "best effort" service philosophy. Unfortunately, this is highly susceptible to attack. Our explanations below will refer to packets as "bad" if they are sent by an attacker and "good" if sent by a legitimate user. Of course, it should be understood that the distinction between good and bad, and for that matter between attacker and user, are informal.



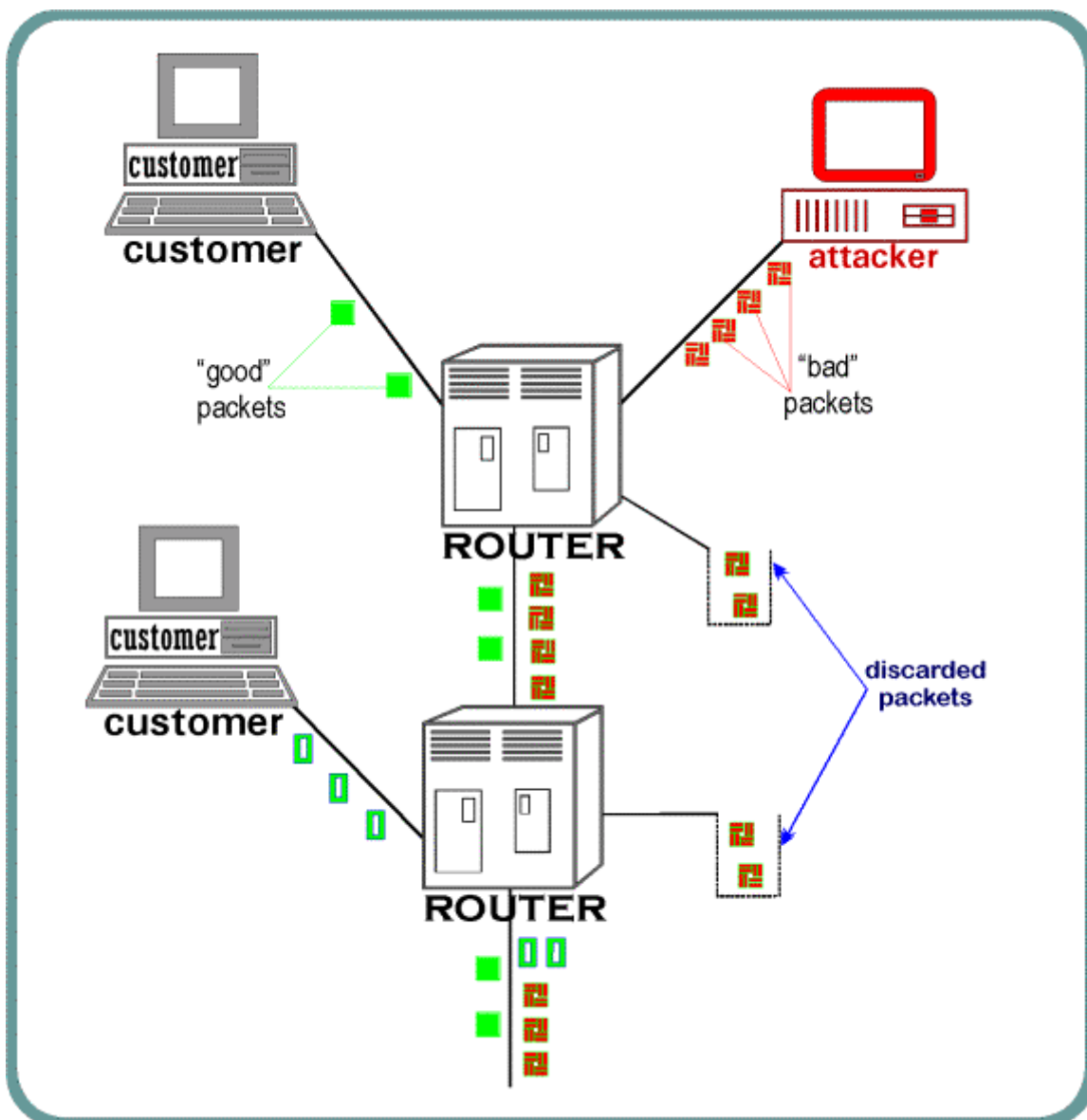
*Figure 1. A Sample Attack Scenario*

Figure 1 shows an attacker sending a large number of packets to a victim. If the router and victim can actually process all of the packets they receive then this merely wastes the otherwise unused time of the router and victim. On the other hand, if the router or victim cannot process the packets fast enough, then some will be lost. Since most of the packets are bad, most of the lost packets are likely to be bad, but it is also likely that some will be good. Of course, the attacker does not mind that his bad packets are lost -- he just wants some of the customer's good packets to be lost.

If we could really distinguish good packets from bad, the denial of service defense would be trivial. We could simply discard the bad packets immediately! We have not solved (nor do we believe it necessary or possible to solve) the problem of divining good or bad intent from packets.

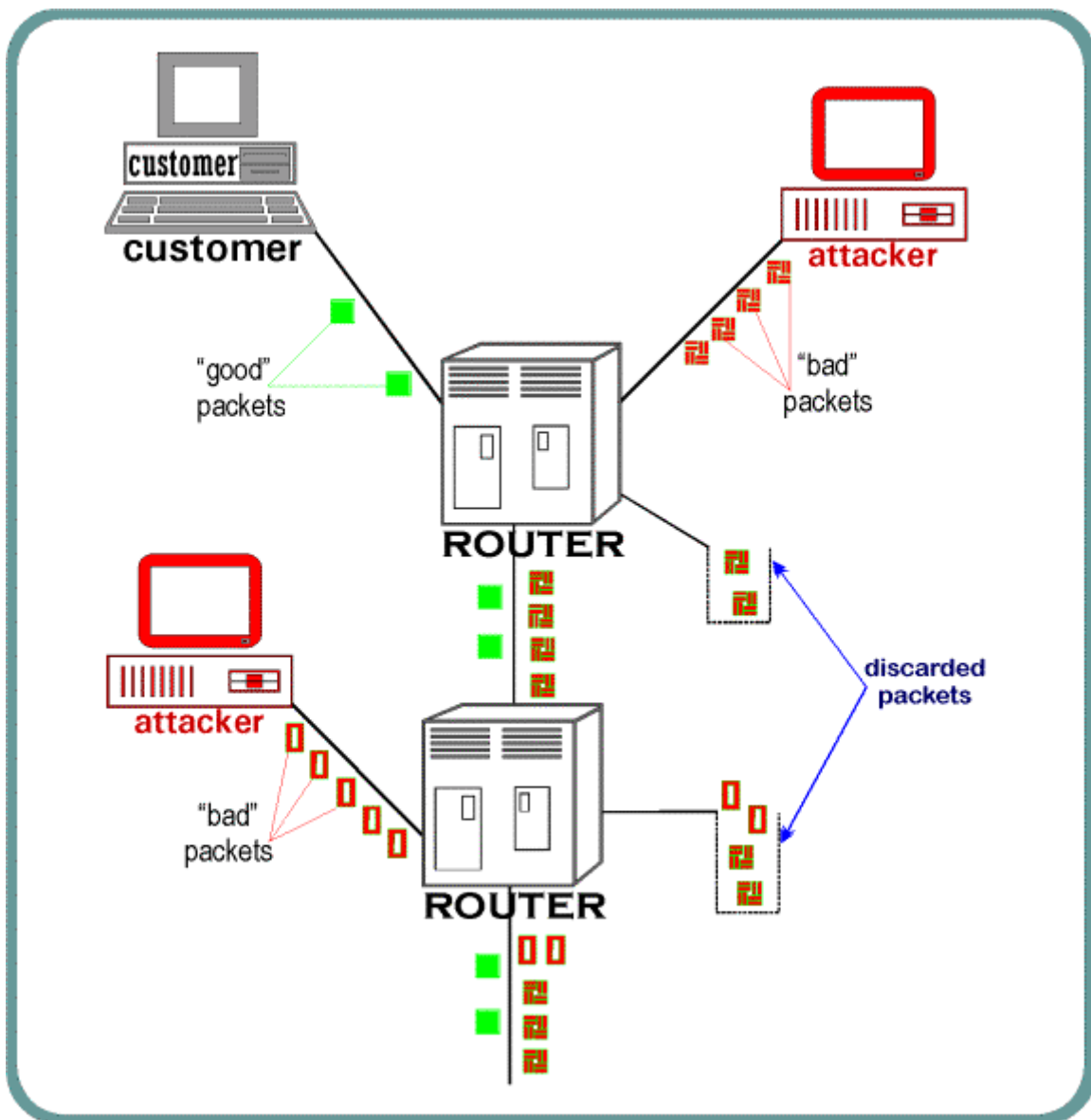
The Cs3 defense requires the router to provide "fair service" to all packet senders. Fair service, in this context, means that the service is divided evenly among all who want it. When one requestor stops requesting then the remaining service is divided evenly among the rest, and so on. Thus if we have one pie to share and three requestors who request  $1/4$ ,  $1/2$  and  $3/4$  of it, then they receive  $1/4$ ,  $3/8$  and  $3/8$  respectively. We do not actually require strict fairness. Other allocation policies, such as "weighted fairness" are also reasonable. In this case, the good and bad packets arrive from different links, so the router could get the desired result by allocating its service fairly among the links from which it receives

packets. If the router did this, then all of the good packets would be forwarded and only some of the bad packets would be lost.



*Figure 2. Local Fairness is Not Sufficient*

Unfortunately, fair service among the directly connected links is not good enough in general. In Figure 2 the upper router forwards all of the good packets to the lower router. However, if the lower router only offers fair service to its two inputs, it will have to discard some of the packets from the upper router, including some of the packets from the customer. In order for the lower router to forward all of the customer packets it needs some way to distinguish them from those of the attacker. This is what our defense provides. The lower router will be able to offer fair service to all three of the machines sending packets in Figure 2. The result, as illustrated in the Figure, is that it will forward all of the customer packets and drop only packets from the attacker.



*Figure 3. Multiple Attackers*

It is worth pointing out that this scheme is effective against multiple attackers. Figure 3 is similar to Figure 2 but the second customer has been replaced with an attacker. In this case all of the customer packets are delivered and the remaining bandwidth is divided evenly between the attackers.

## 4.2. Cooperating Neighborhoods and "Visible Sources"

The Cs3 denial of service defense is distributed among a set of interconnected, cooperating machines called a *"cooperating neighborhood"*. The quality of the defense is related to the size of the cooperating neighborhood. A larger neighborhood provides better defense. We would like one neighborhood to encompass the entire Internet, but here we describe the general case which includes smaller neighborhoods. Within a neighborhood, it is possible to reliably trace the paths of packets.

IP packets, by their nature, cannot be reliably associated with the "users" among whom service would ideally be allocated. A close approximation would be the actual source of a packet, but this cannot be reliably determined from the packet, as mentioned earlier. The current Internet infrastructure provides access to a tiny amount of data related to this source, the link on which the packet arrived. Our additions

provide more, its path through the cooperating neighborhood. The main motivations and features of the resulting protocol, called Path Enhanced IP (PEIP), are described in a separate White Paper entitled [IP Changes to Eliminate Source Forgery](#).

If the cooperating neighborhood is large then this is a lot of information about the source, if the neighborhood is small then the information is not quite as much. The best available approximation to the source of a packet is the link through which it entered the cooperating neighborhood. We refer to the location from which a packet entered the neighborhood (typically a link connected to a cooperating router) as the "visible source" of the packet. The defense allocates service in a fair (or otherwise reasonable) manner among these places.

Each visible source may be shared by many actual users. An attack will reduce service to the users sharing the same visible source as the attacker. The advantage of a large neighborhood is a large number of visible sources, each shared by fewer users, so an attack will deny service to fewer users. For a user who wants to communicate with a particular machine, it is best to be in the cooperating neighborhood of that machine because no attacker from another machine can then deny him service. Conversely, an attacker wishing to deny service to as many users as possible would prefer to share an entry point into the cooperating neighborhood with as many users as possible.

## 5. Processing at the Destination

So far we have concentrated on what routers have to do in order to deliver all of the "good" packets to their destination. At this point it is necessary to admit that the routers are not omniscient. They deliver only an approximation to the set of packets that the destinations would ideally process. The destinations know more than the routers about the costs and benefits of processing different packets. It is on the basis of these that the destination should allocate its service. However, the visible source of a packet is very useful in this determination. In particular, within a given type of packet, it often makes sense to allocate service fairly by visible source.

Below we describe how we expect destinations to handle the packets they receive, i.e. to determine which will be discarded and the order in which the others will be processed. It makes sense to think of a destination as a single host connected to the Internet, but in many cases, all the hosts at a site communicate with places outside the site through a firewall. In these cases it is advantageous to do much or all of this processing at the firewall. We will describe the processing as if it is done at a firewall, but it should be clear how the same ideas apply to an individual host.

### 5.1. Expected and Unexpected packets

Above we defined well behaved communication and mentioned that TCP communication is well behaved. We now extend that definition to describe "expected" packets. When the receiver, R, replies with an indication that it is ready for the sender, S, to send more data, the data that S sends in response is expected by R. In order to be expected this data must also be identifiable with high probability as the response. That is, it should be difficult for an attacker without seeing the reply from R to make up a packet that will be expected by R. TCP communication normally results in expected packets. In particular, when a TCP packet from R to S indicates (by its ack bit, acknowledgment and window) that it is ready for S to reply, a limited number of packets from S to R with the indicated sequence numbers are to be considered expected.

Most of the packets sent in today's Internet are expected. Expected packets are generally to be favored over unexpected packets by hosts and firewalls. Of course, an attacker might use your ftp server to download huge files just to use up your bandwidth. We do not consider it the job of the packet flooding



defense to decide which file transfers are valuable and which are attacks. If your firewall and ftp server are otherwise configured to accept the connection then we assume the packets on that connection are good. The defense will, however, prevent floods of random TCP packets that are not part of any real connection. These packets do not qualify as "expected".

The firewall will immediately forward the packets that are expected by those connections passing through it. The connections themselves effectively limit the rate with their "ack" packets. The firewall will allow only a limited number of packets to pass through in one direction without an "ack" in the other. Other protocols that a site wants to process at high rate and are similarly self limiting can be treated in a way similar to TCP. A well behaved user can send a lot of data at a high rate, but only to someone who has given him permission to send a lot of data at a high rate..

Now we come to the unexpected packets. These include non-TCP packets, such as icmp (for instance, ping), TCP "syn" packets used to open TCP connections, and all packets that would be intentionally discarded (such as those with bad IP checksums or ttl=0). We may include IP fragments in the unexpected class. In general, any packets expected to be expensive to process, such as processing fragments and breaking packets into fragments, should be relegated to a limited rate queue so as to thwart attacks that try to use resources by sending lots of expensive packets.

Unexpected packets are not necessarily bad. However, they are expected to arrive at a relatively low rate, and it is reasonable to limit the rate at which they are processed. In fact, different types of unexpected packets should be sent to different queues with different processing rates. Each queue is then subject to fair scheduling by visible source. Therefore the well behaved customers are very likely to get service for their unexpected packets.

### 5.3. Defenses Against Other Related Attacks

This brings us to the defense against related denial of service attacks. Suppose a site runs a public web server. An attacker, realizing that he cannot deny service by simply sending arbitrary packets, decides to try to monopolize the web server by starting a large number of processes that retrieve web pages from the server. It would be reasonable to limit the number of TCP connections allowed to the server at a time. In all likelihood the server already does this.

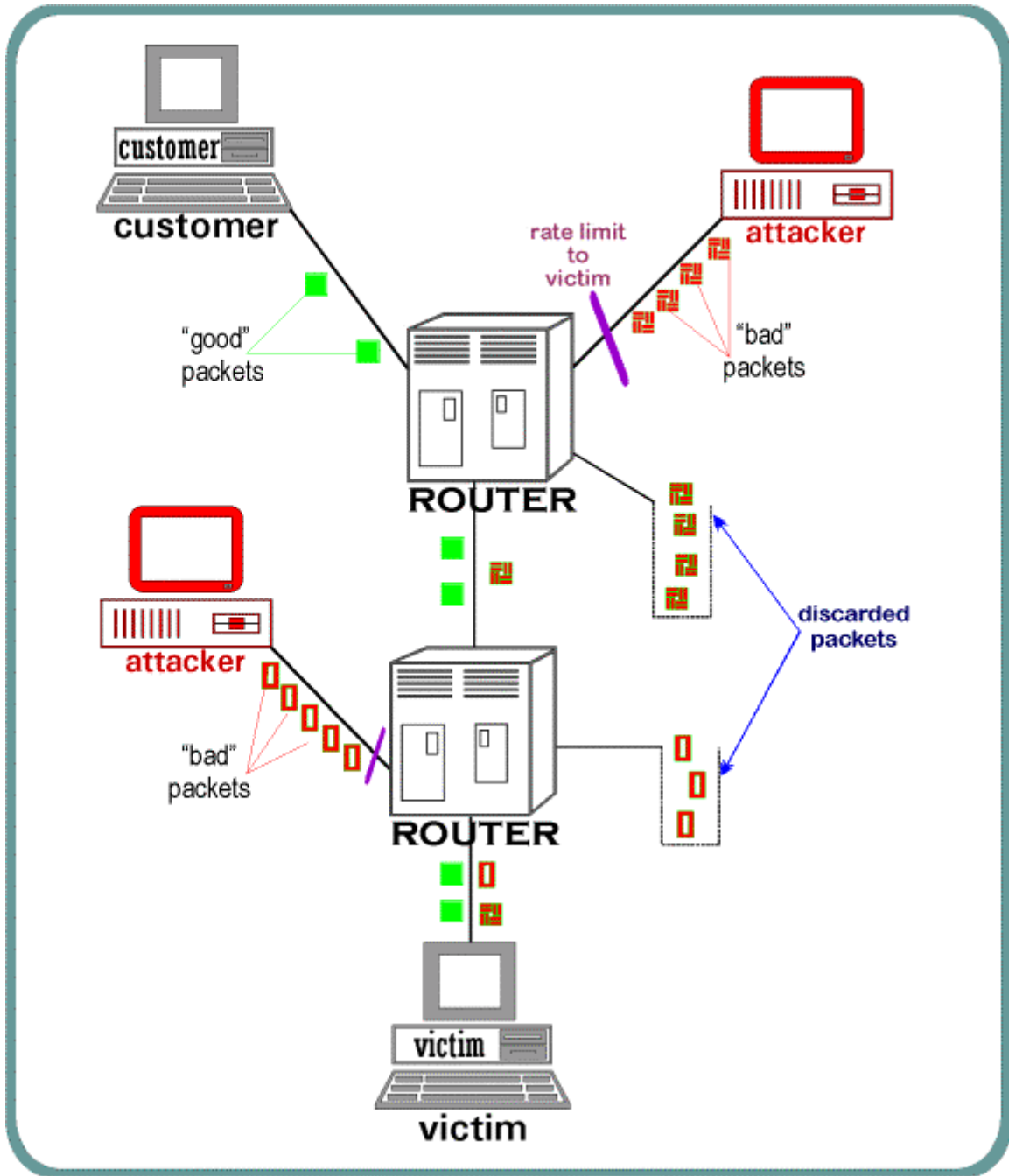
The problem is that the attacker will use up that limit so others cannot get in. However, suppose the firewall limits the rate at which it forwards the syn packets that open these connections, and suppose it allocates that limit fairly to visible sources. Even though the attacker sends many more syn packets than all of the real customers combined, the real customers are all served if possible and the only requests discarded are those from the attacker. See the section on [limits of the defense](#) for more on this subject.

### 5.4. Rate Limiting

The last piece of the defense is that cooperating routers may limit the rate at which they forward packets from certain visible sources to sites that request these limits. This requires more work from the routers and also raises various problems, such as how much traffic will be needed to control rate limits and how to prevent attacks based on this feature. Those issues are beyond the scope of this paper.

The set of packets that are discarded or significantly delayed by a site using the fair service approach is a very good approximation of the set of packets that the site would prefer not to have been sent in the first place. We view this as the best computable approximation to "bad" packets. Visible sources that send a large number of such packets (especially if they send a small number of good packets) are prime candidates for rate limiting. The section on [limits of the defense](#) discusses how well the defense works

with and without use of rate limiting. Readers may be surprised at the effectiveness of fair service without rate limiting.



*Figure 4. Result of Rate Limiting*

Figure 4 depicts the situation after the victim limits the rate from the visible sources that send it bad packets at high rates. Each limit applies to packets with a particular visible source (path) and a destination address in a particular range (the victim site). The first difference is that the upper router is no longer forwarding packets at a high rate from the upper attacker. The link between the two routers is therefore available for other traffic. The second difference is that the lower router is no longer forwarding packets at a high rate from the lower attacker. The link between the lower router and the victim is therefore no longer filled with bad packets from either attacker.

## 6. Analysis of the Defense

### 6.1. Assessment of Incremental Benefits

It is true that larger cooperating neighborhoods afford better defenses against packet flooding attacks. However, installing our defense even at a single site can produce benefits. The only case where our defense does not help at all is a machine with only one neighbor that is not cooperating. This would be the case for a single host connected to an ISP that does not cooperate. A machine with two non-cooperating neighbors, however, can distinguish two visible sources. [\[Endnote 1\]](#) This allows it to defend itself against a single attacker, or multiple attackers that communicate through the same neighbor. Of course, such an attack also affects all of the legitimate communication through that neighbor. The victim cannot reduce the flow of bad packets in this case. It benefits by knowing which packets to ignore, and of course, by being able to communicate through the other neighbor.

Next, consider a corporate network that adopts our defense internally, but has no cooperating neighbors outside the network. The first point is that service can no longer be denied between places inside the network by attacks from outside. This is already a significant benefit. Next, consider the ability of the public to access any public servers inside the corporate network. The analysis above applies in this case. Just think of the entire corporate network as one site with a number of non-cooperating neighbors. However in this case the number of non-cooperating neighbors is larger, probably proportional to the size of the perimeter of the corporate network.

The case of an ISP is particularly interesting because of a property of our defense that has not yet been explicitly mentioned. The firewall component is symmetric. That is, there is no difference between "inside" and "outside". The firewall therefore protects the outside from packet flooding attacks from the inside as well as vice versa. [\[Endnote 2\]](#), This is a reason that upstream providers would want the ISP to use the defense, and therefore an incentive for an ISP to adopt the defense even if it has no cooperating neighbors. Furthermore, if its customers are connected to it by different links, the defense prevents them from attacking each other. If the ISP is connected to multiple upstream providers, then both it and its customers benefit as above from its ability to distinguish which packets come from which of those providers. Of course, once an ISP adopts the defense, both its customers and upstream providers gain by cooperating with it.

Overall, the quality of the defense is related directly to the size of the cooperating neighborhood. A complete quantitative analysis is beyond the scope of this document. However, a relatively simple theoretical argument follows. An attack, at worst, causes the loss of all communication from the visible source(s) of the attack. It is conceivable that traffic from different places might be of different inherent value to the victim. In that case, one can assign a value to each visible source, namely the sum of the values of the communication from all places that communicate through that visible source.. A reasonable measure of the cost of an attack is the sum of the values of the visible sources affected by the attack. Conversely, one might describe as the value of the defense the sum of the values of all the other visible sources, since that is additional value that the attack would have cost without the defense.

### 6.2. Effectiveness of the Fair Service Defense

In principle there is no defense against an attack that cannot be distinguished from an overload of normal traffic. This has important consequences. For instance, a public server that is saturated by a small number of customers can be attacked successfully by a small number of attackers doing whatever customers do. Our visible sources can be viewed as a way of distinguishing between many normal customers and one attacker impersonating many normal customers. This aspect of the defense is at least partly defeated by

distributed attacks. If you can attack from  $n$  visible sources you can claim  $n$  shares of service. At that point the question becomes how many shares is too many. As pointed out by the example above, this depends on the service as well as the server and the infrastructure.

We offer as an example an analysis of a small web server. We imagine for this example that we have a large cooperating neighborhood. The effects of neighborhood size were considered above. Suppose the server input and output bandwidth are each 1Mbps (125KB/sec). The server is meant to serve at most 5 pages/sec, each page being 25KB (which seems about average). One might expect that such a small operation would be easy to attack.

We start with the infrastructure (the routers) which is supposed to deliver all of the good packets to the server and deliver the responses back to the clients. For a typical web server the bandwidth requirement is very different in these two different directions. A normal flooding attack would try to send packets to the server. Fortunately for the defense, a normal client does not have to send very much to the server: a few packets to make the request and a short ack packet for every few thousand bytes sent to it. This comes to about 1KB (about half in the initial request and half in subsequent acks) from the client to the server. This means that the network could deliver data from 125 clients per second. Of course, we expect only a few real clients per second. But even with 100 attackers (or one attacker attacking from 100 places) this would not at all affect the ability of the infrastructure to deliver the good packets from the clients to the server. Even when under attack from 1000 places a customer would be able to send data to the server at a rate that would allow the web page to be downloaded in 8 seconds, slower than desired, but not horrible by today's standards. In other words, the small amount of communication from client to server in this case makes the fair service defense very effective. Rate limiting could be used to further improve matters if necessary, but we delay discussion of that.

The second problem is what the server will do when under attack. We will assume that the server has the ability to process all of the incoming packets at least enough to provide fair service. Otherwise it should be connected to a slower link. In this example we would consider the server to be under unusually heavy load if there were four real customers per second asking for service. In that case one might imagine that one attacker would fill the capacity and that 10 or 100 attackers would effectively deny all service. However a typical web server has an advantage that we have not yet exploited. A real customer is not expected to make a request every second, or even every minute. If the server is willing to devote a little storage to the task, it can allocate its effort over a considerable time, say, hours or days. Suppose the server keeps track of how many requests it has served from each visible source in the last hour. It always serves those that have so far received the least service. In that case we would find that an attack coming from 100 places could indeed deny service, but only for 100 seconds per hour (far less if there were not a heavy load of real customers). Of course, if real customers tend to get two pages at a time then the same attack denies service for twice as long.

There is another possible attack that might not be so obvious as the two above. That is the communication from the server to the client. The fact that the server sends data at a relatively high rate to the client suggests that this communication might be vulnerable. This is not (to our knowledge) a very common attack. If the objective is to attack *all* of the traffic from the server, the attacker would have to flood all of the routers near the server with packets going in the same direction as the packets sent by the server. This would be very difficult unless the attacker had control of a large number of machines very close to the server. A more practical objective would be to attack all of the traffic to one particular client, or to all of the clients in a particular ISP. Suppose a small ISP has 100 customers who share a 10Mbps connection. A normal customer might reasonably expect to get 1Mbps during relatively infrequent large downloads. Suppose this ISP is attacked from 1000 different places sending it packets at the highest possible rate. One customer of the ISP now tries to download a web page. The web server represents one of about 1000

visible sources trying to send packets. Fair service gives it only about 1/1000 of the total available bandwidth, or 10Kbps. That may be sufficient for some purposes but it is far inferior to what the customer is paying for. He wants to be able to download at high bandwidth. (Of course, that is still a lot of trouble just to attack a small ISP!)

We now consider how rate limiting would help. First, note that the attack above must consist almost entirely of unexpected packets. If the ISP is using our defense then its firewall knows this and also knows the visible sources of these packets. In fact, it is probably already discarding most of these packets since it should be forwarding unexpected packets at a rate far less than the total available bandwidth. It could therefore ask the routers in the neighborhood to further reduce the rate at which they forward packets from those visible sources. Unfortunately, in this case it would have to ask for limits to be placed on a large number of visible sources. However, if this could be done it would result in more bandwidth available for the server sending good packets to the customer.

### 6.3. Limits on Cooperation

Cooperation between neighbors requires a certain amount of trust. Of course, it also takes a certain amount of confidence to allow someone to connect his machine to yours, so we do not think this is a big barrier to adoption of our defense. We assume the owners of neighboring machines already know and trust each other. Ideally a similar relationship should hold between owners of all the machines in a large cooperating neighborhood. However, it is not reasonable to adopt a transitive model of trust. That is, you should not trust your neighbor's neighbor just because your neighbor does.

As an example of the problem, suppose an attacker gains control of a router in the cooperating neighborhood. He can then send packets that appear to come from a huge number of different visible sources that he trusts. If others view those visible sources as each deserving a fair share of their total resources then the attacker can claim a large share of those resources. We recommend that each machine identify in its own configuration the set of visible sources in the cooperating neighborhood that it considers to be deserving of fair service. We will call these "shareholders".

Allocating resources among this set of shareholders does not prevent the use of the rest of the forwarding path in allocating the resources within one shareholder. In fact, this is recommended. The reason for this recommendation is that, otherwise, the shareholders act like the immediate neighbors in Figure 1. That is, one attacker can fill the queue for one shareholder and deny service to all other visible places forwarding packets through the same shareholder. It is likely that there will be many more shareholders than immediate neighbors, so the impact is not as bad as sharing service only among immediate neighbors. But, by using the rest of the path within a shareholder we get a benefit analogous to the benefit allocating service to all visible sources rather than just immediate neighbors. A router outside the perimeter of shareholders can then deny service only to those visible sources that forward packets through the same shareholder. A router closer than that, of course, can deny service to all visible sources that forward packets through that router by simply not forwarding those packets. It can also affect other traffic by claiming the entire share of the places forwarding through it.

## 7. Implementation Status

We have started the implementation of a router prototype in Linux. The firewall implementation will follow, also in Linux. The Linux version of the defense is expected to be available for general use later this spring.

The prototype router only really involves two algorithms, one to implement the protocol for recording the paths along which packets are forwarded and one for using fair service to visible source for queuing. The

latter is considerably more complex than the former. However each has the desired complexity. Both have small constant time cost per packet. The forwarding algorithm uses, in addition to the space for the packet itself, a relatively large but constant amount of space (similar to SFQ in Linux or how we imagine WFQ works in Cisco routers). Therefore, these algorithms are practical for large, fast routers, and could be implemented in hardware or software.

---

## 8. Endnotes

[back to reference](#)

[1] In order for two different machines to count as different neighbors the machine receiving packets must be able to tell which packets come from which machine. In this sense, different neighbors are different visible sources, so it is better to have more neighbors even if they do not cooperate.

[back to reference](#)

[2] It is reasonable for an ISP to act as a firewall if the TCP connections that pass through it do not also send packets by other routes that do not pass through the ISP (in fact through the same firewall machine). A multihomed customer who regularly uses the ISP to forward only a subset of the packets of a given connection will suffer because the firewall will not recognize that some of those packets are expected.