Proactive model for Mitigating Internet Denial-of-Service Attacks

Nagesh H.R, K. Chandra Sekaran, Adarsh Rao Kordcal,
Department of Computer Engineering
National Institute of Technology Karnataka, Surathkal, Karnataka, INDIA
hrnagesh2001@rediffmail.com, kch@nitk.ac.in, adarsh_kordcal@rediffmail.com

***Abstract*-Denial-of-Service is one of the most frequent, costly and rapidly growing attacks on the Internet. In a denial of service attack, a malicious user exploits the connectivity of the Internet to cripple the services offered by a victim site, often simply by flooding a victim with many requests. In this paper we have compared the three main architectures already proposed for mitigating the DoS attacks. The comparison is with respect to incremental deployment, traffic analysis, and the attacks on the infrastructure itself. Finally, we combine the strengths of the different proposals to propose a new model for denial of service. Our model uses the concept of active networks to mitigate DoS attacks.**

I. INTODUCTION

A DoS attack can be either a single-source attack, originating at only one host, or a multi-source, where multiple hosts coordinate to flood the victim with a barrage of attack packets. The latter is called a distributed denial of service attack. Sophisticated attack tools that automate the procedure of compromising hosts and launching attacks are readily available on the Internet, and detailed instructions allow even an amateur to use them effectively. Denial-of-Service (DoS) attacks cause significant financial losses.

Launching a denial-of-service (DoS) attack is trivial, but detection and response is a painfully slow and often a manual process. They have caused biggest web sites on the world owned by the most famous E-Commerce companies such as Yahoo, eBay, Amazon, became inaccessible to customers, partners, and users, the financial losses are very huge [1]. On the other hand, if the international terrorist organizations use the DoS/DDoS to attack successfully the web sites or Internet systems of government and military, the results and losses will be disastrous and unimaginable.

II. DENIAL-OF-SERVICE ATTACKS

A. Overview

A DoS attack is an attempt to prevent legitimate users of a service or network resource from accessing that service or resource. DoS attacks usually make use of software to crash or freeze a service or network resource, or bandwidth limits by making use of a flood attack to saturate all bandwidth. They intend to overrun some component of a computer network's resources: bandwidth, memory or CPU so that service to legitimate users is denied. This can be accomplished by sending too many packets at once, malformed packets, or packets requesting complicated and lengthy processing. Once network resources are overwhelmed, it has to ignore some legitimate requests. Therefore, the computer has denied someone the service they have requested.

B. DoS Attack Methods

Smurf Attacks

Smurf attacks are one of the most devastating DoS attacks. In the Smurf (ICMP Packet Magnification) attack, the attacker sends an ICMP echo request (ping) to a broadcast address [11]. The source address of the echo request is the IP address of the victim (uses the IP address of the victim as the return address). After receiving the echo request, all the machines in the broadcast domain send echo replies (responses) to the victim's IP address, as shown in Fig. 1.Victim will be crash or freeze when receiving larger-sized packet flood from many machines. Smurf attack uses bandwidth consumption to disable a victim system's network resources. Smurf attacks can also use UDP echo packets.

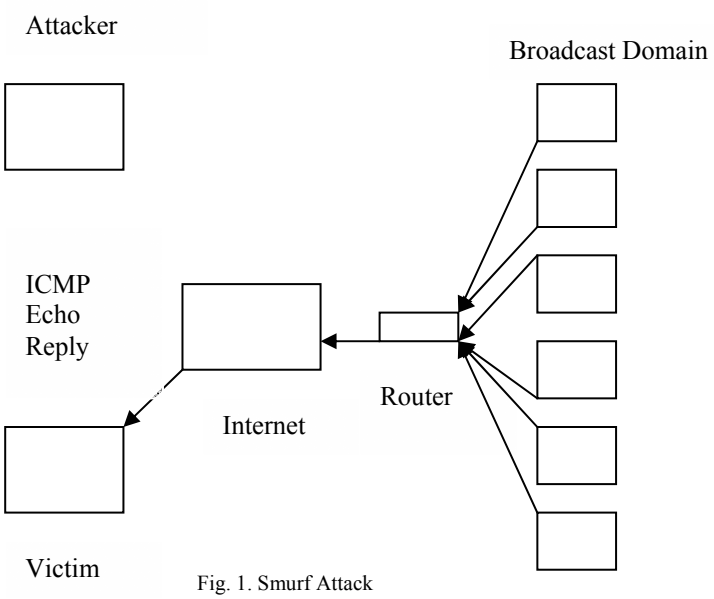

Fig. 1. Smurf Attack

SYN Flood

This method uses resource starvation to achieve the DoS attack. It exploits the vulnerabilities of TCP. Fig. 2 a normal TCP handshake, where a client sends a SYN request to the server, then the server should respond with a ACK/SYN to the client, finally the client sends a final ACK back to the server. But in a SYN flood attack, the attacker sends multiple SYN requests to the victim server with spoofed

International Conference on Information Technology (ITNG'07)
0-7695-2776-0/07 $20.00 © 2007 IEEE

source addresses for the return address. The spoofed addresses are nonexistent on network. The victim server then responds with an ACK/ SYN back to the nonexistent address. Because no address receives this ACK/SYN, the victim server just waits for the ACK from the client.

The ACK never arrives, and the victim server eventually times out. If the attacker sends SYN requests often enough, the victim server's available resources for setting up a connection will be consumed waiting for these bogus ACKs. These resources are usually low in number, so relatively few bogus SYN requests can create a DoS event.

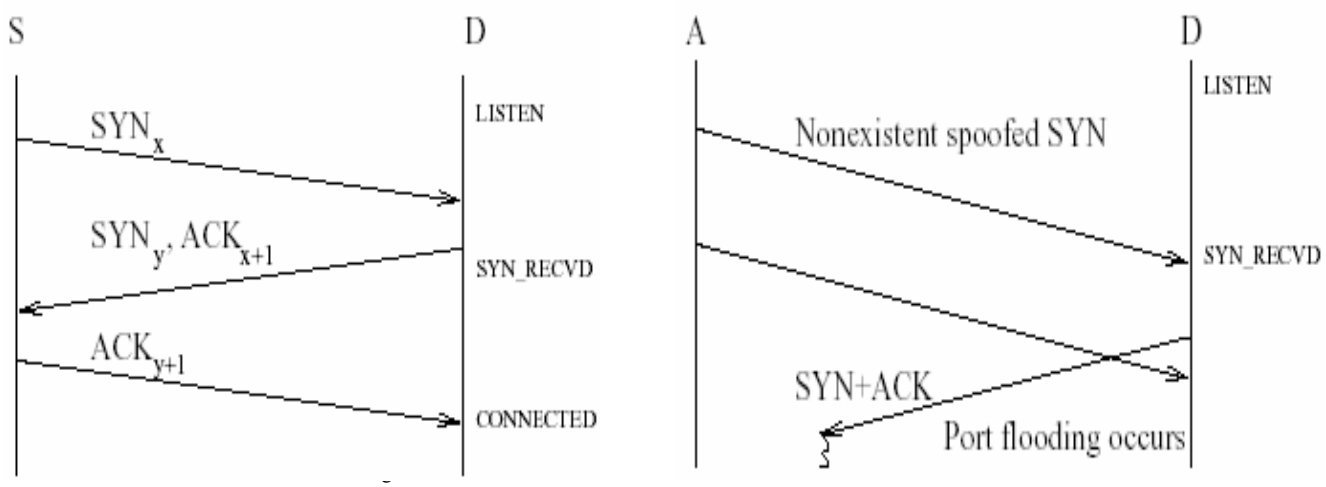

Fig. 2. SYN Flood Attack

DNS Hacking Attacks

Attacks of this type have illustrated the lack of authenticity and integrity of the data held within DNS as well as in the protocols that use host names as an access control mechanism. DNS, being a critical infrastructure, is contacted by all hosts during accessing servers and starting connections [9]. DNS consists of a distributed database that lends to its robustness and also leads to various types of vulnerabilities, which can be categorized into three main types [9]:

Cache Poisoning

Generally, to enhance the process of query response, DNS servers store the common information in a cache. If the DNS server is made to cache bogus information, the attacker can redirect traffic intended for legitimate site to a site under the attacker's control.

Server Compromising

Attackers can compromise a DNS server, thus giving them the ability to modify the data served to the users. These compromised servers can be used for cache poisoning or DoS attacks on some other server.

Spoofing

In this type of attack, the attacker masquerades as a DNS server and feeds the client wrong and/or potentially malicious information. This type of attack can also redirect the traffic to a site under the attacker's control.

DDoS Attacks

DDoS attack is a large-scale, coordinated attack on the availability of Internet services and resources. It launches indirectly the DoS attacks through many compromised computers (they often are called "secondary victims"). The Internet services and resources under the attack are "primary victims". DDoS attack is generally more effective to bring down huge corporate sites than DoS attacks. A typical DDoS attack consists of master, slave, and victim – master being the attacker, slave being the compromised systems and victim of course being the attacker's target.

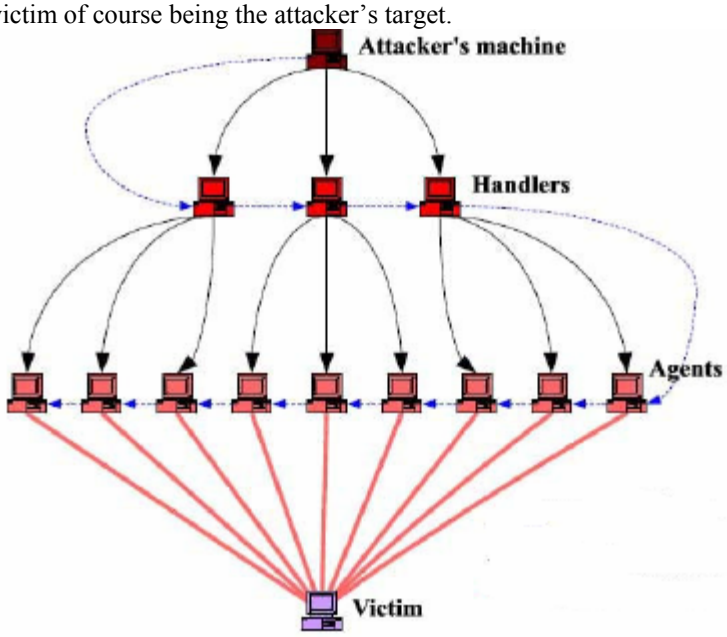

Fig. 3. DDoS Attack Model

Agents report their readiness to the attacker via handlers, compromised machines that will be used to control the attack. This is as shown in Fig. 3.

III. Denial-of-Service Protection Architectures

Denial-of-Service protection architectures can be either proactive (they prevent the attack), or reactive (they react to stop the attack after it has begun). For stopping the attack, they usually require human intervention and cooperation

between different administrative domains. Administrator's use proactive protection to prevent the attack.

This paper deals with three protection architectures viz. Internet Indirection Infrastructure [3], the capabilities [4], and Secure Overlay Services [5]. Each strategy includes some kind of "credential" in each packet. Each strategy also makes changes or additions to the Internet's infrastructure. The infrastructure looks at packets as they cross the network, filtering them based on their credentials. Packets that do not have the proper credentials are dropped and do not reach their destination.

A. Internet Indirection Infrastructure (I3)

If the host can regulate the traffic that it receives, it can prevent itself from being overloaded, thus avoiding DoS attempts. I3's mechanism is to separate the act of sending from the act of receiving a packet. The receiver must agree to get a packet before the packet can be sent to the receiver. This enables an end host the ability to manage its communications.

Separating the act of sending from receiving packets runs contrary to the original design of the network, which sought simplicity and minimal state within the infrastructure. It is this simplicity in fact, that makes the network inherently vulnerable to those who lie and attempt to cheat. I3 accomplishes the separation of sending and receiving by inserting a level of indirection. Packets are not sent directly to their destination.

They are first sent to an intermediary, addressed to a pseudonym for the real IP address. I3 assumes that all hosts will publish their names along with a unique identifier (their pseudonym) in a public directory, just like people publish their name and phone number in the phone book. To send to a particular host, the sender will transmit their data packets to the I3 infrastructure addressed to the public identifier of the destination.

The I3 nodes will then map the identifier to the host's IP address and then forward the packet to its final destination. If a host finds itself receiving more traffic than it can handle, it can tell the I3 infrastructure to stop forwarding the offending traffic. By allowing each host to control its own traffic level, it can stop any attempts to flood its resources and deny its valid clients service

The responsibility for maintaining correct mappings in the overlay between identifiers and IP addresses is left up to each individual host. These mappings are called triggers. Each receiver must send updates to the I3 node storing its trigger reaffirming its value, or the trigger will automatically expire and be deleted.

I3 is an overlay network, which consists of a set of servers that store triggers and forward packets (using IP) between I3 nodes and to end-hosts. Identifiers and triggers have meaning only in this I3 overlay. One of the main challenges in implementing I3 is to efficiently match the identifiers in the packets to those in triggers. This is done by mapping each identifier to a unique I3 node (server), at any given time there is a single I3 node responsible for a given id. When a trigger (id; addr) is inserted, it is stored on the I3 node responsible for id. When a packet is sent to id it is routed by I3 to the node responsible for id, there it is matched against any triggers for that id and forwarded (using IP) to all hosts interested in packets sent to that identifier. To facilitate inexact matching, we require that all id's that agree in the first k bits be stored on the same I3 server. The longest prefix match required for inexact matching can then be executed at a single node. I3 provides a best-effort service like today's Internet. Fig. 4 shows the communication abstractions provided by I3.

I3 provides direct support for communication abstractions, such as mobility, multicast, anycast. Creating a multicast group is equivalent to having all members of the group register triggers with the same identifier. A mobile host that changes its address from R to R' can preserve the end-to-end connectivity by updating its trigger from (id, R) to (id, R'). Conceptually, triggers can be thought of as pointers that point either to receivers or to other triggers.

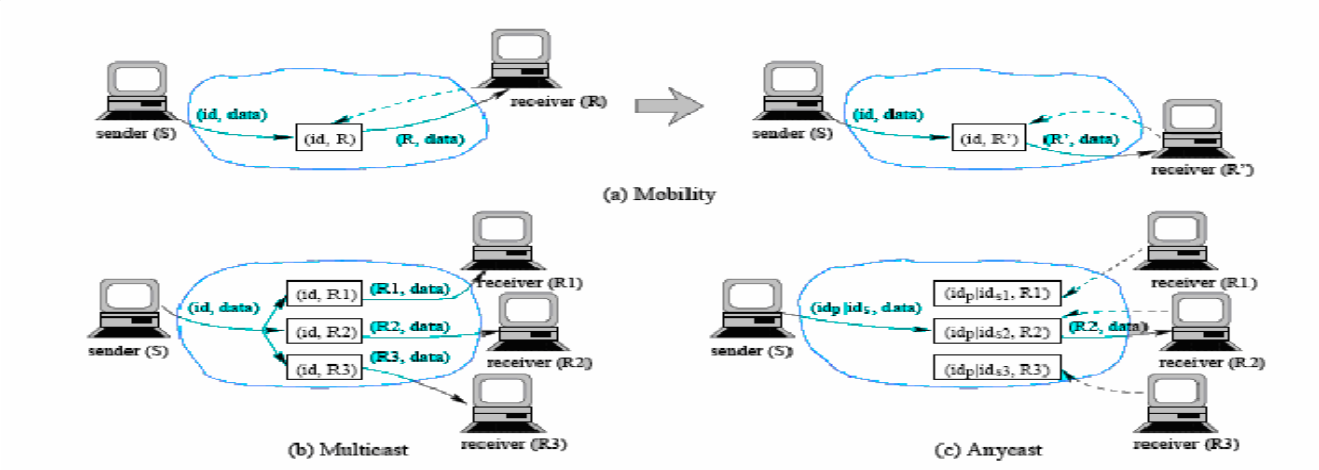

Fig. 4. Communication abstractions provided by I3

(a) Mobility: The change of the receiver's address from R to R' is transparent to the sender. (b) Multicast: Every packet (id, data) is forwarded to each receiver R_i that inserts the trigger (id, Ri). (c) Anycast: The packet matches the trigger of receiver R2. $id_p|id_s$ denotes an identifier of size m, where id_p represents the prefix of the k most significant bits, and id_s represents the suffix of the m - k least significant bits.

B. Capabilities

In this architecture, instead of being able to send anything to anyone at any time, nodes must first obtain "permission to send" from the destination. A receiver provides tokens, or capabilities, to those senders whose traffic it agrees to accept. The senders then include these tokens in packets. This enables verification points distributed around the network to check that traffic has been certified as legitimate by both endpoints and the path in between, and to cleanly discard unauthorized traffic.

Packets addressed to a protected host must have a "capability" included with them. A capability is a permission note from the intended receiver that a specific sender may send a specified amount of traffic at a specified maximum rate to that receiver. Traffic is then sent along its normal paths as determined by the BGP (Border Gateway Protocol).

The capability is validated at each infrastructure server it encounters called a Verification Point (VP), on the path to its destination. If it were found to be invalid, the packet would be dropped. In order to get a capability, a host must communicate with the infrastructure servers called Request-To-Send (RTS) servers. The following is an example interaction between hosts A and B, with A being the initiator of the conversation, and requesting the capabilities to speak to B:

1. Host A sends a packet to its local RTS server ($\propto$), requesting permission to send to host B.
2. $\propto$ Checks to ensure that A has not made too many capability requests. If not, it forwards the request towards host B.
3. B's local RTS server (β) will eventually receive the request from A. If B wishes to communicate with A, as determined by locally specified policies, β will send back a capability.
4. Any RTS or VP server seeing a new capability pass by it, will make a note of the permission.

5. α Receives the capability and passes the information on to A.

Host A now includes the capability in every packet it sends to B. All VPs it passes through verify that the capability is valid before passing it along.

C. Secure Overlay Service

SOS assumes all authorized users are known in advance. Just like I3, SOS uses a distributed overlay network. Authentication is mandatory in SOS. Each sender must authenticate themself using IPSec or other cryptographic authentication protocols. Once authenticated, traffic enters the overlay network and is forwarded according to the distributed hash table protocol to a node called the beacon. In SOS, the receiving host guards itself by employing a simple filtering mechanism. It will only accept packets whose source is one of a small pre-selected subset of the overlay network. The beacon is the only SOS node to know the identity of these selected nodes, called secret servlets. After the beacon receives the packet, it forwards it to a secret servlet. This last hop SOS node will then tunnel the packet to its receiver. The receiver (and its surrounding routers) will check all packets to ensure that their source address is a secret servlet. If not, the packet will be discarded. The SOS architecture is shown in Fig. 5.

The forwarding of a packet within the SOS architecture proceeds through five stages:

1. A source point that is the origin of the traffic forwards a packet to a special overlay node called a SOAP that receives and verifies that the source point has a legitimate communication for the target.
2. The SOAP routes the packet to a special node in the SOS architecture that is easily reached, called the beacon.
3. The beacon forwards the packet to a secret node, called the secret servlet, whose identity is known to only a small subset of participants in the SOS architecture.
4. The secret servlet forwards the packet to the target.
5. The filter around the target stops all traffic from reaching the target except for traffic that is forwarded from a point whose IP address is the secret servlet.

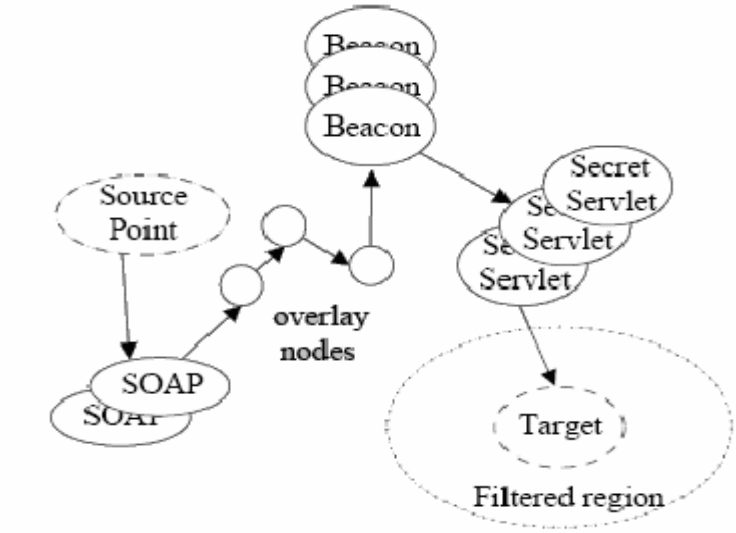

Fig. 5. Secure Overlay Service

IV. Theoretical Analysis of Different Architectures

A. Incremental Deployment Analysis

The solution must allow technical deployment over time. We cannot expect that every ISP or company will implement the new functionality at the same time. The following table lists the merits and demerits of different architectures with respect to incremental deployment.

TABLE I
INCREMENTAL DEPLOYMENT ANALYSIS

Merits of I3, Capabilities, and SOS	
I3	Very simple to add and organize servers
Capabilities	Very simple to add and organize servers Easy to add clients by inserting new Capabilities server at BGP server. No client modification required. Centrally deployed verification points can moderate traffic. Provides protection to the implementers. Strong protection available, if capabilities scheme is enforced on all communicators.
SOS	Very simple to add and organize servers With Local filtering, protection is very strong as there is a limited list of approved senders.
Demerits of I3, Capabilities, and SOS	
I3	All Clients must be modified to use pseudonyms or a proxy must be inserted to capture and modify all traffic. Without near-complete deployment, IP addresses will still be used. Once a critical segment has adopted I3, a server may be able to insist on compliance from the remainder of clients for service. Until then, it is still vulnerable.
SOS	Clients must be modified to send to SOS or a proxy must be inserted.

B. Traffic analysis attacks

An attacker performs a traffic analysis attack when it observes the traffic passing through a router or on an infrastructure link. It can see the packet headers and the packet contents (if they're not encrypted). The following table lists the merits and demerits of different architectures with respect to traffic analysis attacks.

TABLE II
TRAFFIC ANALYSIS

Merits of I3, Capabilities, and SOS	
I3	Recover by switching IP address and creating new triggers.
Capabilities	Attacker with a forged credential can only disrupt one sender/receiver pair. High-rate traffic will be dropped quickly and will not affect the other communicators. Automatic recovery at set intervals is possible.
SOS	Receiver driven recovery by choosing new secret servlets.
Demerits of I3, Capabilities, and SOS	
I3	On path between receiver and I3 node, IP addresses are easy to learn. Receiver can be attacked directly using IP address; it can disrupt all ongoing connections. Can learn the trigger/IP address relation by observing traffic entering and leaving an I3 node.
Capabilities	Close proximity attacker can insert additional traffic to use up permitted bandwidth of all communicators.
SOS	Credential can be clearly observed on the path between SOS and the receiver. Packets forged with this information can overwhelm all connections in progress.

C. Infrastructure Attacks

Attacks on the Internet infrastructure can lead to enormous destruction. Just as an attacker can compromise individual hosts to create a zombie army, it can use other vulnerabilities to compromise infrastructure routers. Once an attacker has complete control over the router, it can observe all information passing through the router as well as modify or delete it. The following table lists the merits and demerits of different architectures with respect to infrastructure attacks.

TABLE III
INFRASTRUCTURE ATTACKS

Merits of I3, Capabilities, and SOS	
I3	Receivers avoid a misbehaving router through trigger selection. No one node is critical to any particular operation. The functionality of the failed node will be provided elsewhere.
Capabilities	Connections can always maintain valid capabilities. RTS servers can filter traffic efficiently as they should only communicate with their local clients and local BGP routers. All other traffic can be discarded. DoS attack requires close physical proximity.
SOS	No one node is critical to any particular operation. The functionality of the failed node will be provided elsewhere. If a beacon is attacked the protocol will choose a new beacon.
Demerits of I3, Capabilities, and SOS	
I3	Malignant routers can amplify traffic, send it to genuine triggers, and overwhelm the destination. Pseudonym not bound to the sender.
Capabilities	Can forge packets with locally stored capabilities. Uses up the bandwidth for each connection. Enable a DoS attack by enlarging all outgoing capabilities from local clients.
SOS	If compromised node is the beacon, it can reveal the secret servlets identities allowing attackers to send unfilterable traffic.

V. Discussion and the Proposed Model for Denial-of-Service Attack

Every model makes some assumptions about the vulnerabilities of the network. A crucial idea made clear by the I3 proposal is that any model should not introduce new vulnerabilities. But, Many of I3's weaknesses arise from the ability of anyone using any identifier. Better infrastructure can be a combination of the strengths of an overlay network and the capabilities infrastructure, with multiple credential generation or authorization points.

This protects against the dangers of a denial of service attack on the infrastructure. A design might include a credential that is valid for any sender, but impose strict limits on its use. Capabilities should have varying length validities. Ideally they should be short lived and new ones should be acquired. They should also be able to be withdrawn in case of an attack, or automatically expire. Better DoS model require trade-offs between overhead and security.

A. Proposed model Architecture using Active Networks

Traditional data networks passively transport bits from one end system to another. Ideally, the user data is transferred opaquely, i.e., the network is insensitive to the bits it carries and they are transferred between end systems without modification [8]. The role of computation within such networks is extremely limited, e.g., header processing in packet-switched networks and signaling in connection-oriented networks. Active Networks break with tradition by allowing the network to perform customized computations on the user data. These networks are "active" in two ways:

1. Switches/routers perform computations on the user data flowing through them.
2. Individuals can inject programs into the network, there by tailoring the node processing to be user and application-specific.

The active networks replace the passive packets of present day architectures with active "capsules". Capsules are the miniature programs that are executed at each router they traverse. This change in architectural perspective, from passive packets to active capsules, simultaneously addresses both of the "active" properties described above. Fig. 6 provides a conceptual view of how an active node might be organized. Bits arriving on incoming links are processed by a mechanism that identifies capsule boundaries, possibly using the framing mechanisms provided by traditional link layer protocols. The capsule's contents are dispatched to a transient execution environment where they can safely be evaluated. The programs are composed of "primitive" instructions that perform basic computations on the capsule contents, and can also invoke external "methods", which may provide access to resources external to the transient environment. The execution of a capsule results in the scheduling of zero or more capsules for transmission on the outgoing links and may change the non-transient state of the node. The transient environment is destroyed when capsule evaluation terminates.

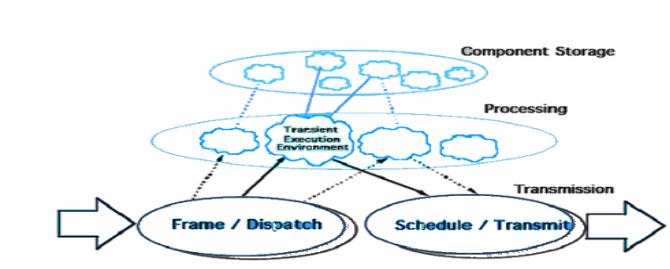

Fig. 6. Active Node Organization

To mitigate DoS/DDoS attacks we can set up an intelligent router assisted by a management station. Whenever a packet arrives at the router, it will ask the management station to see whether the source is using more bandwidth/resources. If it is taking more resources, management station will inform the router not to forward the packet, instead it will be dropped. The code can be deployed in any router using the capsule. The code contains the actual protocol and embedded data.

VI. Conclusion

One of the greatest threats to Internet security and functionality is a denial-of-service attack. Denial-of-Service has proven to be a difficult Internet security problem to solve. In this paper we have compared, and listed the merits and demerits of different solution architectures. We found that the strengths of SOS and Capabilities can be combined together to mitigate the DoS attacks on the Internet. Our model using active networks to mitigate DoS/DDoS attacks can set up an intelligent router assisted by a management station. Whenever a packet arrives at the router, it will ask the management station to see whether the source is using more bandwidth/resources. If it is taking more resources, management station will inform the router not to forward the packet, instead it will be dropped. The code can be deployed in any router using the capsule. The code contains the actual protocol and embedded data. The disadvantage of our model is that it requires additional cost for setting up active nodes.

References

[1] Computer Crime and Intellectual Property Section (CCIPS) of the U.S. Department of Justice

[2] Akamai. http://www.akamai.com.

[3] Stoica I., Adkins D., Zhuang S., Shenker S., and Surana S. "Internet Indirection Infrastructure," *In Proceedings of ACM SIGCOMM*, 2002.

[4] Anderson T., Roscoe T., and Wetherall D. "Preventing internet denial-of-service with capabilities," *In Proc. of Hotnets-II*, Cambridge, MA, Nov. 2003.

[5] Keromytis A., Misra V., and Rubenstein D. "SOS: Secure Overlay Services," *In Proceedings of ACM SIGCOMM*, 2002

[6] Savage S., Wetherall D., Karlin A.R., and Anderson T. "Practical network support for IP traceback," *In SIGCOMM,* 2000.

[7] http://www.sds.lcs.mit.edu/darpa-activenet/

[8] David L. Tennenhouse and David J. Wetherall Telemedia, "Towards an Active Network Architecture Networks and Systems Group," MIT.

[9] Anirban Chakrabarti and G. Manimaran, Iowa State University, "Internet Infrastructure Security: A Taxonomy," *IEEE Network*, November/December 2002

[10] Y. Xiang, Y. Lin, W.L. Lei and S.J. Huang, "Detecting DDOS attack based on network self similarity," *IEEE Proc.-Commun.*, Vol. 151, No. 3, June 2004

[11] Chen, E.Y.; Yonezawa A. "Practical techniques for defending against DDoS attacks," Computer Systems and Applications, 2005. The 3rd ACS/IEEE International Conference-2005.