Provider Provisioned Overlay Networks and Their Utility in DoS Defense

Jinu Kurian
Dept. of Computer Science
University of Texas at Dallas
Richardson, Texas - 75083
Email: jinuk@student.utdallas.edu

Kamil Sarac
Dept. of Computer Science
University of Texas at Dallas
Richardson, Texas - 75083
Email: ksarac@utdallas.edu

***Abstract*—The current overlay deployment model supports minimal or no involvement by ISPs in overlay deployment and operation. This model rules out a richer set of interactions between the native and overlay layers and therefore sacrifices the potential performance gains that can be realized through such interactions. In this paper, we present a new overlay deployment model (PON) where the ISPs are actively involved in the deployment and operation of overlay networks. Because of ISP involvement, the PON model can better support some of the more difficult overlay-based applications. The PON model also establishes a solid business model which provides incentives for the deployment of PON overlays. To demonstrate the utility of the PON model we consider one of the more difficult applications, denial of service (DoS) defense, and describe a PON overlay which provides DoS defense as a value-added service. We call this overlay FONet and describe the FONet architecture and its functional overview. Finally, we evaluate the FONet architecture in a controlled laboratory environment to illustrate its effectiveness in providing the DoS resistant services proposed.**

I. Introduction

Recently, overlay networks have been proposed for a wide variety of applications. These include multicast [1], QoS [2], [3], resilient routing [4], DoS defense [5], [6] etc. One common feature among the disparate overlay networks that have been proposed is that they are largely independent from ISPs for their operation. This operational independence has been largely influential in the aforesaid popularity and has been considered vital for some overlay applications (e.g., resilient routing services [4]). Based on their level of independence from the ISPs there are two models for overlay deployment in the Internet today: (1) the P2P model (not to be confused with P2P overlays like Gnutella, Kazaa etc) where the ISPs have no involvement in the deployment of overlays at all (e.g., RON [4]) and are usually unaware of the overlay presence, and (2) the service overlay (SON) model where the ISP may provide some support (e.g., ISPs may provide support for the deployment of QoS provisioning overlays [2], [3]) to an overlay service provider (OSP) for the initial deployment of the overlay network.

In this paper, we argue that this independence while well suited for some applications, does not necessarily extend to *all* possible applications. To support this premise, we propose a new model of overlay operation that derives explicit support from the underlying ISPs for their deployment and operation. We call this model the provider provisioned overlay (PON) model. In the PON model, overlays will be built in a collaborative manner by a number of participating ISPs and OSPs. ISPs deploy PON nodes in their domains and offer them to OSPs. ISPs may also provide additional support for the operation of the overlay to the OSP and charge the OSP for the resources and support provided. The OSP will lease a number of PON nodes from multiple ISPs and deploy overlay-based applications on top of them. It offers these applications as value-added services to interested networked application servers (NAS servers) and charges them for the service. Depending on the type of service offered, NAS servers may in turn charge their end users for the value-added service. This PON model as described above has several advantages over existing deployment models (more details in the next subsection) and can better support many of the overlay applications in the Internet today. As an example of the latter, we will consider a well researched overlay application, denial of service (DoS) defense, and consider how existing architectures can be improved upon with the new deployment model (see Section II).

A. The PON Overlay Model

The PON model as described above has some unique characteristics that provide many advantages over the existing overlay deployment models.

Deploying end-to-end services: In the Internet today, the deployment of end-to-end services (for example QoS, multicast, IP traceback) have proven to be notoriously difficult. The lack of global cooperation and incentives for deployment are cited [7] as the main reasons behind this difficulty. In PON, the presence of the OSP to coordinate between different ISPs solves this coordination problem in the deployment of end-to-end services. As discussed in the next paragraph, it also provides incentives for deployment. One of the more difficult end-to-end challenges in the Internet today is end-to-end security and DoS resistance. Although overlay based solutions [5], [6] have been proposed to provide some measure of relief, the scope of these solutions is limited due to the lack of involvement of ISPs and a mechanism to coordinate between ISPs. In the rest of the paper we will primarily consider how the PON model can be applied to build an overlay based architecture (called FONet) for DoS defense. Before going into the details of FONet, we will consider some other advantages of PON in general.

ISP, OSP and end user friendliness: The PON overlay is deployed in a federated manner with multiple ISPs leasing out their local PON nodes to the OSP. This avoids the need

for a single entity (the OSP) to spend potentially prohibitive amounts of money and resources to deploy overlay nodes across multiple domains. By charging the OSP for the resources and support provided, the ISP can gain added revenue from the overlay. This ISP friendliness provides an incentive to ISPs to deploy PON overlays in their networks. It also helps ISPs to be aware of the overlay traffic demands and account for it in their traffic engineering (TE) settings. The PON model also empowers the end user of the value-added service. In the current Internet model, there is no guarantee provided to the end user in the inter-domain scale. In the PON model, the users are provided an end-to-end guarantee by the OSP for the value-added services they receive.

Building better overlay architectures: The PON model allows for the active involvement of the ISPs during overlay operation and deployment. This enables PON overlays to better support some of the applications considered by traditional overlays. Routing overlays [4] can be enhanced with locally available routing information to reduce probing costs and to provide best routes. During the construction of routing overlays, it has been shown that an awareness of the underlying native layer topology is important for overlays with better overall performance and failure resistance [8], [9]. In probing based routing overlays, costly oscillations may occur if the overlay and native layers are unaware of each other. The involvement of ISPs in overlay operation can help avoid such oscillations by provisioning for overlay traffic in traffic engineering calculations. QoS provisioning overlays [3] can be enhanced with MPLS based path protection and bandwidth guarantees, packet marking and labeling at the edges of the domains and selection of best nodes and routes for building the overlay. As we mentioned before, DoS defense through overlays is another application which has received a lot of attention in the research community in recent years [5], [6], [10]. We will concentrate on this application and our PON based solution (FONet) for the rest of the paper.

The rest of the paper is organized as follows. Section II describes the the state-of-the art in DoS defense solutions, the FONet architecture and its design rationale. Section III gives a functional overview of the architecture. Section IV gives experimental results from evaluating the FONet architecture in a representative testbed. Finally, Section V concludes the paper.

II. FONet: A PON Overlay for DoS resistant Communication

In FONet we aim to design an architecture that builds on existing approaches in overlay based DoS defense [5], [6]. Through the active involvement of the ISPs in the deployment and functioning of the FONet overlay, we are able to create an architecture that has comparatively lesser resource redundancy; is intrinsically protected against attacks; is scalable to a large number of domains, end users and NAS servers; and can provide different classes of DoS-protection services to suit varied end-user and NAS server requirements. Before looking at the FONet architecture in more detail, we will first consider some of the related work in the area.

A. State-of-the-art in DoS Defense Mechanisms

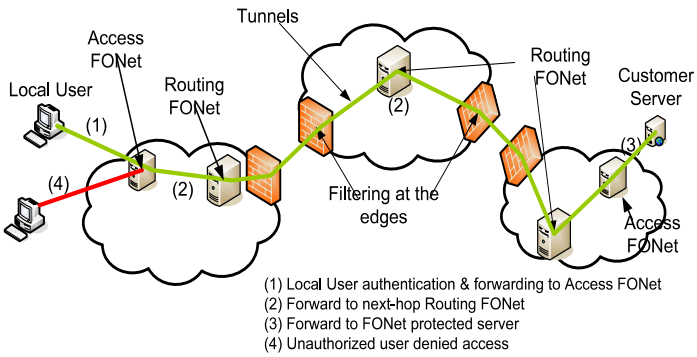
Practical DoS defense solutions have been a long standing problem in the Internet. We will consider some of the most recent solutions proposed.

Solutions which have a goal of eliminating the vulnerabilities that make DoS attacks possible in the Internet are an intriguing area of development. Some of these methods redesign the Internet to remove the vulnerabilities entirely [11], [12]. Others mask the vulnerabilities and thereby provide DoS resistance to compliant end systems [5], [6], [10]. Based on the type of protection service provided, we can classify some of these solutions into three categories:

1) Strict Protection Service (SPS): In SPS, the protected node is open only to traffic from well known or previously authorized users. SOS [5] and Mayday [6] are examples of methods which provide SPS. In these methods, users are authenticated at an access point and their traffic is routed through a set of overlay nodes to the target. Unauthenticated traffic is dropped at the access point or at filters deployed around the protected node. In the existing models to provide SPS service, the overlay by itself is vulnerable to attack. So, the overlay design has to rely on circuitous routing mechanisms and location hiding to protect itself. While mostly effective, this method also introduces a significant overhead in the end-to-end response time for the user. Additionally, the resource redundancy required to build these overlays limits their scalability.
2) Partial Protection Service (PPS): In PPS, the protected node is open to both authorized and unauthorized users. Authenticated user traffic is given preferential treatment in the network to the unauthenticated traffic. TVA [12] and SIFF [13] are examples of methods which provide PPS. In these methods, authorized traffic carries a token to distinguish it from unauthorized traffic in the network. Existing PPS methods may require significant modifications to the network core to be deployed. Their effectiveness however makes them a suitable solution for the next-generation Internet.
3) Basic Protection Service (BPS): In BPS, the protected node has no prior knowledge of the users. The objective here is to distinguish between human users and automated tools (bots) which may be used to launch DoS attacks. WebSOS [14] and the i3-based approach [10] are examples of methods which provide BPS. In these methods, lightweight authenticators like GTTs or client puzzles are administered to ensure that only human users are allowed access to the NAS server. BPS by itself offers only a limited protection and access control. It needs to be used in conjunction with other schemes for effective protection.

While all of the solutions described above are effective, they consider DoS defense from a ”one size fits all” perspective.

Fig. 1. FONet architecture and operation



It is plain to see that not every NAS server (and end user) in the Internet has the same security requirements, the same security budgets or the same service to provide. It is also not currently feasible to provide complete security to every user in the Internet. In FONet we aim to design a DoS protection mechanism that is able to provide different classes of services to suit the varied requirements of different users and NAS servers who are security aware and are willing to pay for the added protection.

B. FONet Architecture

In developing the FONet architecture, we aim to meet some important basic requirements of overlay based solutions. The objective behind these requirements is to ensure that the protected NAS server can continue to serve its legitimate users even while it is under attack. These requirements are: 1) there should be a filter ring around the target which can distinguish between legitimate (authenticated) and illegitimate (possibly attack traffic), 2) there should be a guaranteed path to the filter ring from the traffic source, and 3) there should be a strong authentication mechanism at the traffic source which restricts access to the overlay network only to authorized users.

In addition to the basic requirements described above, we seek to incorporate certain enhancements due to the PON model to improve the overall security of the architecture while simplifying its operation and extending its functionality:

- The architecture by itself is made secure. This avoids the need for circuitous routing to protect overlay traffic. This also has an effect of reducing redundancy and improving performance.
- The architecture should be able to provide SPS, BPS, and PPS services within a single framework in a large scale. This allows the architecture to serve the protection needs of different types of NAS servers and their users.

Based on the requirements listed above, we have carefully designed the FONet overlay to meet these requirements. The FONet architecture is practical, compact, and can provide DoS resistant communication effectively in a large scale. At the highest level, the FONet architecture consists of two components (Figure 1):

FONet overlay network: FONet overlay nodes are high end server machines capable of handling large amounts of Internet traffic. Two types of FONet nodes may be present in a domain (1) Access FONet nodes and (2) Routing FONet nodes. Access FONet nodes serve as the entry/exit point to the FONet overlay for traffic from within the local domain. They are made accessible only to local users and the local Routing FONet nodes. Routing FONet nodes forward user data over the overlay network towards the remote NAS server. They are made accessible only to the local Access FONet nodes in their domain and Routing FONet nodes in the neighboring domains. The traffic between FONet nodes is tunneled using an appropriate tunneling mechanism. Many options are available for this including MPLS based tunnels, SSH/IPSec based tunnels, leased lines etc. MPLS for example can provide DoS resistance, bandwidth guarantees and path protection services to the FONet traffic.

Filtering support: Filtering support is required from the ISPs to ensure that the FONet nodes by themselves are not open to attack. For Access FONet nodes, null routing can be used to drop all traffic from outside the domain at the domain boundaries. In the case of Routing FONet nodes, all external traffic except from neighboring Routing FONet nodes needs to be filtered. Access Control lists (ACLs) can be deployed at the border routers to selectively filter traffic to the Routing FONet. In the case of MPLS based tunnels, traffic can be filtered out based on the appropriate MPLS label of the neighboring Routing FONet node.

III. Functional Overview

Figure 1 shows the overlay architecture and its functional overview. From a user perspective, if she is accessing a PPS or SPS protected NAS server, she is required to pre-register with the NAS server to obtain an authentication token. Once registered, the user can then contact the nearest Access FONet in her domain where she is authenticated before being allowed access into the FONet overlay. In the BPS case, the authentication service will consist of a Turing test or client puzzle. Once authenticated, user traffic is routed through the overlay network hop-by-hop until it reaches the remote NAS server's Access FONet and finally to the NAS server.

A. User Authentication

The objective of user authentication is to securely authenticate remote and distributed users of all remote NAS servers while maintaining minimal user information at the Access FONet nodes and without requiring that all authentication requests be forwarded to the remote NAS server for approval. User authentication has three steps, 1) pre-registration (for SPS and PPS) to obtain an authentication token, 2) initial authentication at the user-site Access FONet node with the authentication token, and 3) session maintenance after the initial authentication.

Pre-registration: The purpose of pre-registration is to obtain a token for a desired domain that the user wishes to access through the FONet overlay. The user contacts its nearest Access FONet node (the user can use a DNS like service for example fonet.foo.com) which will mediate the access request to the remote domain (the user should have previously

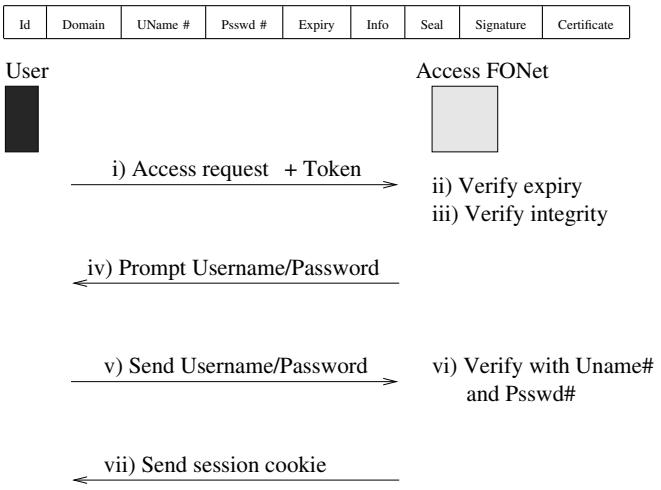


Fig. 2. Token and Initial user authentication.

registered for the FONet service). By using simple measures like SYN cookies to prevent spoofing, and lightweight authentication techniques like GTTs at the Access FONet, the pre-registration step can be protected against DoS attacks. Additionally, the remote NAS server and Access FONet nodes can give priority to registered users over new registrations in its queuing and processing. If the user is allowed to successfully register, a token is created with the user's information and returned to the mediating Access FONet to be installed in the user's machine as a cookie. This token is verified during authentication at the Access FONet nodes to allow further access to the user.

The structure of the token installed during pre-registration is shown in Figure 2. We will consider each field in turn. The Id field specifies the Access FONet at which the token is valid. Domain specifies the domain(s) for which the token has been issued for and is valid for. The Uname# and Psswd# are keyed-hashes of the user's registration name and password provided during initial registration. The key used for the hash is the public key of the mediating Access FONet which initially forwarded the user request. Expiry contains an explicit (unencrypted) expiration time for the token. The Info field is an optional field that may be used by the NAS server to include added information (e.g., user specific data like IP or MAC address, user restrictions, QoS service class, etc) about the user. These six fields form the Data field of the token. The Data field is hashed and keyed with the private key of the mediating Access FONet to form the Seal of the token. At the Access FONet, the Data field is again hashed and signed with the private key of the Access FONet to form the Signature. Finally, the public key of the mediating Access FONet is signed with a trusted certificate authority (a trusted third party like a commercial CA) and forms the last field. This public key can be used by an Access FONet for remote authentication to validate the Signature field in the token. It is signed by a trusted certificate authority to ensure its validity as a digital certificate.

The token has some important properties that make it secure and non-transferable. The Domain field binds the token to a single domain. This ensures that the user cannot reuse a token for other NAS sites it has not registered for. Even if the token is intercepted by an adversary, it is not useful without the username and password of the user. If required, the token can also be bound to a single user machine (for example using a combination of fields like BIOS checksum, MAC address, operating system version etc) which ensures that it cannot be distributed to create an attack. Finally, the Seal field ensures that the token has not been altered in any manner by the user or an attacker. Since the Seal includes the Expiry and Id fields as part of the hash, the user cannot modify the token to extend its validity or reuse an expired/revoked token.

Initial user authentication: User authentication (Figure 2) consists of the user-site Access FONet node retrieving the token from the user and verifying it based on user entered information. In all three cases; SPS, BPS or PPS, the first check is always a GTT. This prevents the possibility of compromised users forming botnets. A large scale attack would require human intervention and a distinct username/password (which cannot be obtained from the token) which makes them implausible.

Based on whether the user is local or remote, the authentication procedure varies slightly. We consider each in turn. For local user authentication, the user sends the its access request along with the token (the token can be installed as a cookie so that the browser fetches it automatically) to its local Access FONet node. Once the Access FONet has the token, it does the following checks. Except for the Id check, failure in any of the other checks results in an authentication failure. First, it checks to see if the Id in the token is for itself. If not then it goes to remote user authentication. It then checks if the remote domain requested matches the Domain in the token. If the domain matches, it checks a locally maintained black list for the domain to see if the user has had its token revoked. This black list can be maintained with minimal information with the Uname# in the token. Next it checks the Expiry to verifiy that token is current and has not been tampered with by recalculating the Seal. Finally, it inquires the user for a username/password combination. The user provided username and password are hashed and verified with the Uname# and Psswd# fields in the token. Remote authentication is similar to local authentication after the user request is forwarded to the installing FONet node. More details can be found in [15].

Now that we have described the FONet architecture and its functional overview, in the next section we will present our experimental results which validate the design choices.

IV. Evaluations

To evaluate the protection services offered by the FONet architecture, we conducted experiments in a laboratory environment. The purpose of our experiments was to emulate the protection services offered by FONet by flow isolation through the overlay and filtering of unwanted traffic. For this purpose, both SPS and PPS cases are emulated and the protection

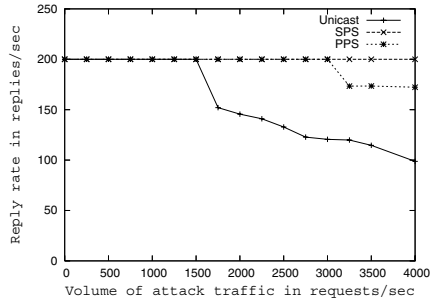


Fig. 4(a). Avg reply rate.

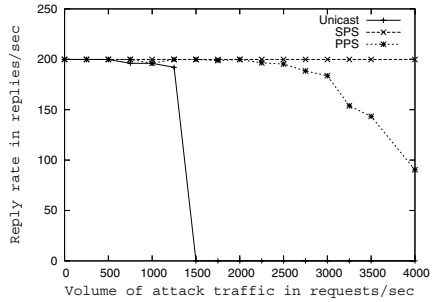


Fig. 4(b). Worst case reply rates.

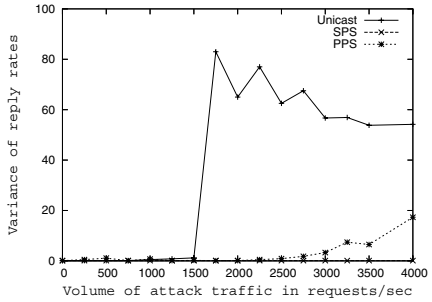


Fig. 4(c). Variance of reply rates.

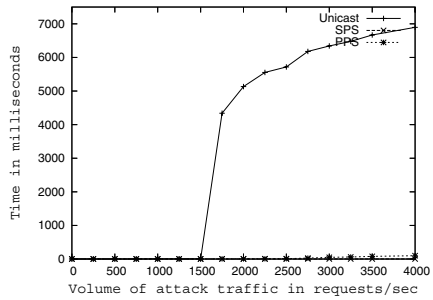


Fig. 4(d). Avg time for one request

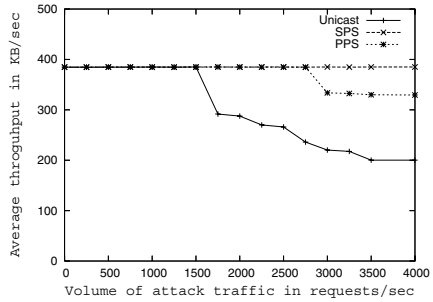


Fig. 4(e). Throughput.

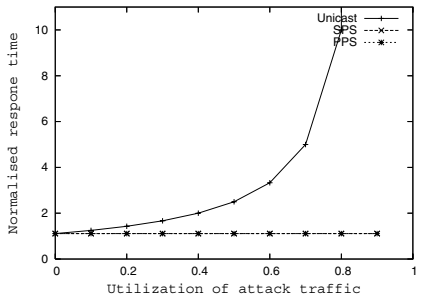


Fig. 4(f). Theoretical perspective.

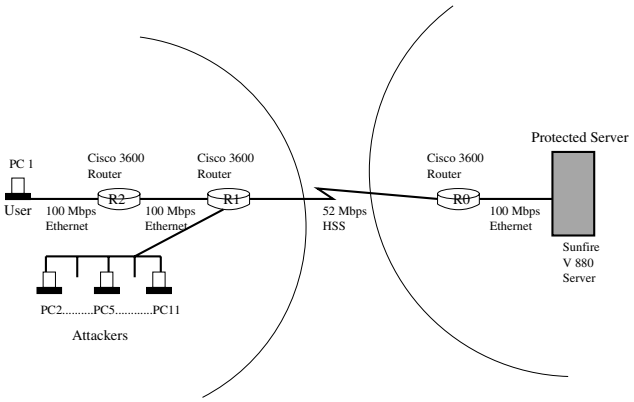


Fig. 3. Experimental topology.

offered in each case is compared with an unprotected NAS server. For the SPS case this amounts to filtering all traffic to the protected NAS server, except that from the overlay. For the PPS case, both overlay and native traffic is allowed to the NAS server but priority is given to the overlay traffic.

Our setup (Figure 3) consists of 3 Cisco 3600 routers, 10 Linux PCs as hosts and attacking nodes, and a Sunfire V880 server running an unmodified Apache webserver hosting a 190.8 Kb html file. The legitimate clients have a constant 200 http requests/sec rate while the attack traffic is varied between 250 to 4000 requests/sec. Httperf [16] is used to simulate both the attack traffic and the legitimate user requests. FONet nodes are emulated on top of the two routers R0 and R2. R2 serves as the Access/Routing FONet node of the remote domain while R0 serves as the Access/Routing FONet node in the server's domain.

For the SPS case, R0 is configured with an access control list which will limit all traffic to the server except for traffic from the (overlay) legitimate users. Here, R0 serves as a border router for the server's local domain and filters out unauthorized traffic to the overlay node. Only authorized traffic from R2 is allowed access to the server. For the PPS case, the legitimate user traffic is provided a guaranteed 10% of the total bandwidth to the server. This is required to emulate the DoS resistant tunnels between FONet nodes in the architecture. Finally at the server, legitimate users are given a 5 to 1 priority over unknown user traffic. Since the PPS server aims to provide guaranteed service to its authorized users only, the priority queuing ensures that legitimate users are served before unknown users via unicast and attack traffic. The BPS case is similar to PPS provided the server uses priority service for overlay traffic, so we don't consider it further. In all cases, experiments are conducted twice for each attack rate for a duration of 30 minutes per run.

Response times under attack: Figures 4(a), 4(b) and 4(c) show the performance of the server as average, worst case and variance of the reply rates from the server for varying attack rates. The average and worst case reply rates are a direct indicator of the performance of the server under load. Worst case response rates show the effect the attack on the server over time. As the attack prolongs, the server gets more and more bogged down and eventually as seen in the unicast case, stops responding completely. Both SPS and PPS cases show a marked difference in their reply rates, demonstrating their effectiveness under attack. The variance shows the jitter in server performance under attack. Variance rates for the SPS and PPS cases are smaller because their performance unlike

in the unicast case is consistently good.

User's perspective: Figure 4(d) shows the performance of the server from a single legitimate user's perspective. The figure shows the time required by the server to complete a single HTTP GET request from the user for the webpage (including connection establishment, request and transfer time). From the user's perspective, the results show that the user will experience lesser delays and smaller response times from the server when it is under attack as opposed to the unicast case. The variance of request completion times was also measured (not displayed) to measure the jitter as experienced by the user. Smaller variance in response times were observed for SPS and PPS again showing that the user experiences a more consistent performance from the server even under attack.

Throughput: Figure 4(e) shows the average throughput of the server for legitimate requests (also referred to as goodput). Goodput is a direct indicator of the overall performance of the server throughout the duration of the attack. The SPS and PPS cases maintain a high goodput while the unicast case shows a drastic reduction in its goodput under attack.

Response time (theoretical) : To validate our experimental results, we further analyze all three cases from a queuing networks perspective. Our objective in this analysis is to estimate the response times experienced by an end user in all three cases. Assume that the server has an exponential service rate of μ requests/sec. From the server's perspective there are three (poisson) arrival streams to itself, the legitimate users arrive at a rate λ_l requests/sec and the unknown users and attack traffic arrives at a combined rate of $\lambda_b + \lambda_c = \lambda_a$ requests/sec (by the additive property of Poisson streams). The analysis here is for an infinite capacity server, it easily extends to a finite capacity case, so we avoid the discussion here.

For the unprotected server, there is no differentiation between arrival streams. The total arrival rate for all traffic is Poisson with rate $\lambda_l + \lambda_a$ requests/sec. The expected response time for legitimate users is thus $E[\tau] = \frac{1}{\mu-\lambda_l-\lambda_a}$ seconds. For the SPS case, the only traffic arriving at the server is authorized traffic at λ_l requests/sec. So, the expected response time of the server $E[\tau]$ reduces to $E[\tau] = \frac{1}{\mu-\lambda_l}$ seconds. For the PPS/BPS cases, we assume that the server is non-preemptive priority based with processing priority given to the legitimate users. Assuming that the server has the same service rate μ for both classes of customers, the expected response time of the legitimate traffic is as for the SPS case, $E[\tau] = \frac{1}{\mu-\lambda_l}$ seconds.

Figure 4(f) plots the normalized response times for varying levels of attack traffic utilization $\rho_a = \frac{\lambda_a}{\mu}$. ρ_a is varied from 0.1 to 0.9. Higher values of ρ_a are ignored to allow for a stable system. The legitimate users are assumed to have a uniform arrival rate for $\rho_l = \frac{\lambda_l}{\mu} = 0.1$.

The experimental and theoretical results validate the feasibility of the FONet architecture and its ability to support multiple DoS resistant services. FONet protected NAS servers show a drastic difference in their resistance to attack when compared to unprotected NAS servers. The SPS case affords maximum protection but is the most restrictive. The PPS and BPS cases offer a good balance between security and functional restrictions.

V. Conclusion

In this paper we described the PON approach for overlay deployment and its advantages over existing deployment models for many applications. To demonstrate the feasibility of the PON approach, we developed FONet, an overlay based architecture to provide DoS resistant communication services in the Internet. We experimentally and theoretically evaluate the services provided by FONet to validate our architectural design. Our results show that the FONet architecture can provide effective protection against DoS attacks for NAS servers. In particular the results show that the SPS service offers nearly perfect protection to DoS attacks while the PPS service offers significantly higher protection to DoS attacks when compared to the unprotected case. Currently, we are working on some other issues including routing, partial deployment, and compromised nodes. A more detailed discussion and possible solutions to some of these issues can be found in the accompanying technical report [15].

References

[1] Y.-H. Chu and S. G. Rao, "A case for end system multicast," in *Proceedings of ACM SIGMETRICS*, Santa Clara, CA, USA, June 2000.

[2] Z. Li and P. Mohapatra, "QRON: QoS-aware Routing in Overlay Networks," *IEEE JSAC: Special Issue on Recent Advances on Service Overlay Networks*, vol. 22, no. 1, pp. 29–40, January 2004.

[3] Z. Duan, Z.-L. Zhang, and Y. T. Hou, "Service overlay networks: SLAs, QoS, and bandwidth provisioning," *IEEE/ACM TON*, vol. 11, no. 6, pp. 870 – 883, December 2003.

[4] D. Andersen and H. Balakrishnan, "Resilient Overlay Networks," in *Proceedings of 18th ACM SOSP*, Banff, Canada, October 2001.

[5] A. Keromytis, V. Misra, and D. Rubenstein, "SOS: An architecture for mitigating DDoS attacks," *IEEE JSAC: Special Issue on Service Overlay Networks*, vol. 22, no. 1, pp. 176–188, January 2004.

[6] D. Andersen, "Mayday: Distributed filtering for Internet services," in *4th USENIX SITS*, Seattle, WA, USA, March 2003.

[7] L. Peterson, S. Shenker, and J. Turner, "Overcoming the Internet Impasse through virtualization," in *Proceedings of the ACM HOTNETS III*, San Diego, CA, USA, November 2004.

[8] Z. Li and P. Mohapatra, " The Impact of Topology on Overlay Routing Service," in *Proceedings of IEEE INFOCOM*, Hong Kong, China, March 2004.

[9] J. Han and D. Watson, "Topology Aware Overlay Networks," in *Proceedings of IEEE INFOCOM*, Miami, FL, USA, March 2005.

[10] D. Adkins, K. Lakshminarayanan, A. Perrig, and I. Stoica, "Taming IP Packet Flooding Attacks," in *Proceedings of Second ACM HOTNETS II*, Cambridge, MA, USA, November 2003.

[11] T. Anderson, T. Roscoe, and D. Wetherall, "Preventing Internet denial-of-service with capabilities," in *Proceedings of ACM HOTNETS II*, Cambridge, MA USA, November 2003.

[12] X. Yang and D. Wetherall, "A DoS-limiting Network Architecture," in *Proceedings ACM SIGCOMM*, Philladelphia, PA, USA, August 2005.

[13] A. Yaar, A. Perrig, and D. Song, "SIFF: A Stateless Internet Flow Filter to Mitigate DDoS Flooding Attacks," in *Proceedings of IEEE SSP*, Oakland, CA, USA, May 2004.

[14] W. Morein, A. Stavrou, D. Cook, A. Keromytis, V. Misra, and D. Rubenstein, "Using graphic turing tests to counter automated DDoS attacks against web servers," in *Proceedings of the 10th ACM ICCS*, Washington, DC, USA, October 2003.

[15] J. Kurian and K. Sarac, "Provider provisioned overlay networks and their utility in dos defense," University of Texas at Dallas, Tech. Rep., March 2007.

[16] D. Mosberger and T. Jin, "Httperf: A Tool for Measuring Web Server Performance," in *Proceedings of the First WISP*, Madison, WI, USA, June 1998.