Multipath Routing, Congestion Avoidance and DDoS Resistance

Kari Visala
Helsinki Institute for Information Technology HIIT / Aalto University School of Science and Technology
kari.visala@hiit.fi

Abstract

In open networks, especially in the global Internet consisting of thousands of independent subnetworks with different policies, a portion of the nodes must be considered malicious. A distributed denial of service (DDoS) attack targets a victim using multiple colluding nodes scattered around the network, that try to deplete the resources of the target by unwanted messages. Because the number of unprotected systems attached to Internet is large, botnets have grown to consist of millions of nodes and they are actively used to attack businesses, organizations, and countries. The Internet has become a critical part of our infrastructure and therefore a solution is needed as we become more and more dependent on it. Multipath routing and congestion control have been suggested as possible solutions to this problem. In this paper, we explain how these techniques could be used as DDoS prevention measure as well as discuss their advantages and disadvantages by taking into account various factors, such as costs of deployability, effectiveness, scalability and others.

1 Introduction

DDoS attacks [14] are possible against all types of networks such as wireline Internet, ad hoc networks, and overlays. The aim of the attack is to deplete the victim of some resources such as bandwidth, memory, or processing power by sending malicious requests to the target. A DDoS attack can be launched from multiple nodes in the network and the attackers can be distributed similarly to the legitimate users of the targetted service. For example, a large botnet that has infected the machines of ordinary end-users, can be used for a coordinated flooding of packets towards the victim from multiple sources. The traffic can be hard to distinguish from flash crowd effect [21] that stems from the Zipf [2] popularity distribution of many types of content. In addition, the unwanted traffic can be masqueraded as legitimate protocol messages and only the service itself can distinguish which packets belong to the attack, possibly after some analysis. Because of this, it is very hard for the core network itself to react to unwanted traffic without additional information. We analyze and classify different types of DDoS attacks based on literature in Section 2. The problem of DDoS is a serious one in the current Internet and requires countermeasures as we show in Section 3, where we review the existing measurements about the current state of the DDoS traffic in the Internet.

The problem of DDoS can be even more difficult for overlay networks. Overlays have also been suggested as the solution to the DDoS problem [21, 4]. Even though they are easy to deploy and do not need involvement from the Internet service providers (ISPs), these approaches are vulnerable to DoS attack vectors from the unsecured underlay such as an IP network [23]. This is a fundamental limitation that cannot be changed by the algorithms used on top of the overlay. Also, overlay solutions are bound to have some overhead compared to the use of the underlaying network.

Because Internet, based on the TCP/IP protocol suite, was not designed upfront for a large number of hostile nodes attached to it, anybody can freely send unwanted traffic to any network attached to the Internet simply by knowing the IP address of the target if no additional techniques are deployed. This shortcoming has been partially addressed, as an afterthought, by middleboxes such as NAT and firewalls. For example, if the source address of an IP packet is checked on route, it is not possible to spoof it by the sender. However, because the network should not interpret the content of the packets [20], firewalls can be implemented only relatively close to the protected network, which may not be enough to defend against massive DDoS attacks. Also, no complex algorithms in line-speed core routers are possible and memory operating at the required speeds is expensive. There is also the question of what incentives the tier-1 operators have to prevent unwanted traffic, because that would require investments on their part and reduce their income, which is a function of the transited packets. Due to these problems, some clean-slate networking technologies have been proposed. For example, PSIRP [13], approaches the problem by introducing special *forwarding identifiers* that act as capabilities for end-to-end network routes. The sender must first perform a succesful rendezvous on a *slow path* in order to receive the capability to use the *fast path* resources. However, the rendezvous function uses an overlay, that still must solve the DDoS problem on that level, thought the problem is easier as no unwanted underlay traffic is possible.

Many approaches to countering DDoS attacks have been proposed [16]. For example, the source of each packet can be checked at routers to avoid source address spoofing and make the attackers accountable for their actions. However, since many DDoS attacks are launched by botnets from even millions of victim nodes, the accountability does not prevent attacks originating from these nodes directly. There are also lots of solutions for protecting the end-user computers from worms and intrusion, such as automatic security patches, firewalls, and virus protection, but since the nodes attached to the Internet are numerous and heterogeneous, this approach will never fully prevent attacks. In addition, it does not prevent attacks from nodes owned by malicious

users. Firewalls and middleboxes can be used to filter traffic in the network, but they require state to be stored and are typically deployable only near the victim's network, which makes them less effective as the firewall itself can become the bottleneck for the DDoS attack converging from all over the network. Resources can be automatically replicated and accounted based on demand in data centers, but this requires investment from the service provider. In the case of large data centers this may not be a large problem if the costs are distributed between multiple services. However, this solution does not help home users and somewhat sacrifices the original fully distributed philosophy of the Internet. Some attacks can be mitigated by better protocol design [6]. For example, state on the server side should be avoided and standardized behaviour can allow middleboxes to intercept packets. This solution, however, does not prevent simple packet flooding attacks and cannot be applied to all applications. Clean-slate approaches can use more drastic methods such as source routing based on capabilities [13] but they have to be first deployed widely and may require changes in applications.

In this paper, we analyze, how multipath routing [8] and congestion control or rate-limiting [4, p. 134] based solutions could be used to address the DDoS problem and explain some of the proposed approaches. Our general point of view is that routing and congestion control can be seen as the two sides of the same resource allocation problem of link capacity division to multiple flows: Routing is basically about adding new links to the route by starting from empty set of resources. On the other hand, congestion control can be seen as a substraction of resources from the case where the sender floods all outgoing links. Both of these lead in the general case to the idea of using multiple paths through the network to carry packets of the same flow. This mechanism can be optimized to maximize rates for all flows based on some fairness measure, take into account application QoS requirements, adhere to some routing policies etc. We explain the basic idea of multipath routing in Section 6 and how congestion control is related to the problem in Section 5 and go through some of the current technologies and analyze their strengths and weaknesses as a DDoS defenses.

No security mechanisms are without some drawbacks. They add complexity of the implementation, may cause fluctuations in the network, and may be incentive incompatible with some of the stakeholders. We categorize these challenges in Section 4. Concluding remarks are made in Section 7.

2 Distributed Denial of Service Attacks

The goal of a DDoS attack is to prevent the legitimate use of a service by deploying multiple nodes in the network to send unwanted traffic. In a typical case, attackers send packets to the server and deplete its resources such as network bandwidth, processor time, or memory. There are other types of attacks, such as using malformed packets to break protocol semantics, but in this paper we restrict our scope to handling large quantities of extraneous packets, because it is the most difficult problem from the perspective of the network. The attacks that are not based on amount of traffic can be handled on the application level by better protocol design and are outside the scope of this paper. It should be noted that these packets are not always targeted towards the service itself, but for example, their goal might be to prevent the service from using an outside resources to perform its task by consuming resources of the network. We consider (multi-path) routing and congestion control to operate on the network layer even though congestion control is implemented in TCP in the Internet Protocol Suite. Basically, we consider functionality related to end-to-end resource management to be about network layer and of interest here.

In [5], remote denial of service attacks were classified as either network device level, OS level, application level, data flood, or protocol feature attack. We are mostly interested in network device level and data flood attacks as the other problems can be tackled at different level. In [16] a more comprehensive classification of DDoS attack and defense mechanisms was made. On the top level, the types of attack were partitioned into the following categories:

- **Classification by degree of autonomy**, that is divided to *manual*, *semi-automatic*, or *automatic*. The (semi-)automatic methods could be further classified by their *communication mechanism* (direct, indirect), *host scanning strategy* (random, hitlist, signpost, permutation, local subnet), *vulnerability scanning strategy* (horizontal, vertical, coordinated, stealthy), and *propagation mechanism* (central, back-chaining, autonomous).
- **Classification by exploited weakness**, that is either *semantic* or *brute-force*.
- **Classification by source address validity**, that is either *spoofed* or *valid*. The spoofed mechanisms could be further divided into routable or non-routable based on address routability or random, subnet, enroute, fixed based on spoofing technique.
- **Classification by possibility of characterization** and if it is characterizable, then whether the traffic is filterable or non-filterable.
- **Classification by attack rate dynamics**, which is either constant, increasing, or fluctuating rate.
- **Classification by the impact on the victim**, which is either *disruptive* (self-, human- or non-recoverable) or *degrading*.
- **Classification by victim type**, which is application, host, resource, network, or infrastructure.
- **Classification by persistence of agent set**, that can be constant or variable.

The categorization for either known or expected defense mechanisms in [16] is summarized below:

- **Classification by activity level**, which was divided to *preventive* and *reactive*. Preventive defense mechanisms were further partitioned to *attack prevention*

(system and protocol security) and *DoS prevention* (resource accounting and multiplication). Reactive methods were split to either *classification by attack detection strategy* (pattern, anomaly or third-party) or *classification by attack response strategy* (agent identification, rate-limiting, filtering, or reconfiguration).

- **Classification by cooperation degree**, that can be *autonomous*, *cooperative*, or *interdependent*.
- **Classification by deployment location**, that can be *victim network*, *intermediate network*, or *source network*.
- **Classification by attack response strategy**, which had the following subcategories: *agent identification*, *rate-limiting*, *filtering*, and *reconfiguration*.

In this paper, we consider brute force attacks that target host, network, or infrastructure victim without restricting other attack type classifications above. The defense mechanims covered here fall into reactive methods, both rate-limiting and reconfiguration. They are interdependent as they require deployment at multiple networks and thus, the deployment location is intermediate network.

3 Internet Measurements

DDoS attacks are not just a theoretical possibility, but they have become commonplace in the current Internet. The prevalence of DDoS attacks globally in the Internet was given a conservative estimate in [17] using so called *backscatter analysis* of traffic from a large enough set of IP addresses from traces gathered from 2001 to 2004. 68700 attacks were detected against 34700 distinct IP addresses. The targets of attack included large companies like Amazon and Hotmail, ISPs, and individual dialup connections. In 2002, an attack towards the infrastructure of the Internet was launched without any specific target service [16] using the root DNS servers as the target. The motivation for attacks ranges from mischief to religious, ethnic, or political reasons to commercial gain. The intensity of some of the attacks were over 100000 packets per second. We believe that it is safe to assume that the number and intensity of current attacks is considerably larger because of the growing number of nodes and traffic attached to the Internet and the increased importance of Internet to all areas of society.

4 Challenges

In addition to many technical challenges, the prevention of DDoS has to be weighted against other competing goals. Even a single stakeholder can have multiple evaluation criterion that are partly contradictory and in the global Internet the problem becomes extremely complex because of the different stakeholders involved. In fact, it has been argued that Internet architecture should be designed based on the concept of *tussle* [3]. This means that possible points of contention are identified and instead of fixing the balance between different goals top-down, the architecture should allow the parameters to find their values based on the external game between agents related to the system. Below we categorize the main challenges to the solution to DDoS and in the following sections discuss how the multipath routing and congestion control based approaches answer to these facets of the problem:

- *Technical* challenges include scalability of the solution to the future number of users, nodes, applications, and traffic in the Internet. Basically it follows from this requirement that functionality must be distributed and intelligence and resources are not always collocated. In addition the solution must efficient, that is, the overhead incurred has a cost to the users in terms of latency, bandwidth, memory, and processing power. The solution must be compatible with the current technologies and the basic architecture of the Internet. For example, it must be assumed that the core routers forward packets at line-speeds of tens of Gigabits per second, which requires expensive, high speed memory for the routing tables. All proposed solutions must also be control-theoretically stable.
- *Ease of use*, by which we mean that the security mechanism should not hinder the normal use of the system. There is evidence [9] that many vulnerabilities in the Internet, such as the protection of end-user computers, can in fact result from the rational behaviour of the users even though security solutions to these problems exist.
- *Architectural* constraints, such as the *end-to-end principle* (E2E) [20, 1], which basically states that the network itself should only have minimal functionality that is required to efficient utilization of the invested resources, and rest of the features should be implemented on higher layers at endpoints to be more flexible. However, this goal as such is not in direct conflict with the goal of the DDoS prevention because protecting the network resources themselves from malicious users is required for the efficient use of the network for legitimate users. On the other hand, because of E2E, the network should not interpret application-specific information inside packets and it follows that the network cannot decide which packets are legitimate based on their content.
- *Trust* challenge means that each architecture places some assumptions about trust relationship between resource owners. For example, policy decision points must reside at trusted nodes, which limits what kind of distribution strategies for security functionality can be used.
- *Complexity* of the technology should be minimized to keep if flexible and evolvable to the changing needs of stakeholders. It can be said that one of the key features behind the success of IP protocol has been its simplicity that has allowed it to be run on top of everything and everything be run top of it.
- *Deployability* in an incremental way is almost a prerequisite for a new technology to be adopted in the Internet that consists of about 3000 ISPs and roughly 30000 ASes.

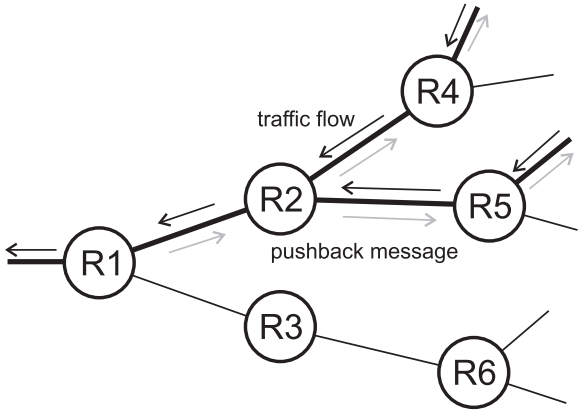

Figure 1: Router R1 sends pushback messages upstream to limit rate closer to sources.

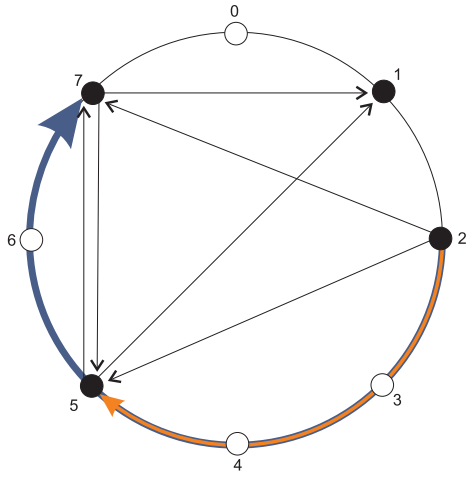

Figure 2: A simple Chord ring example with 3-bit address space and four nodes.

- *Incentive compatibility* means that for the technology to be deployed, each participating entity must have some incentive to switch to use it. For example, in [19] the incentives of tier-1 operators to deploy caching is questioned as it would reduce their profits. In the case of DDoS, transit operators are compensated to transfer also malicious packets and the competition is limited, which may limit where the DDoS prevention functionality can be placed in the network.

5 Congestion Control

The goal of congestion control is to avoid congestion from collapsing the whole network throughput and divide the link resources optimally based on some fairness metrics such as *max-min fairness* or *proportional fairness* [12], which also maximizes transit operator profits when individual users are modelled as simple price takers. In the Internet Protocol Suite, the congestion control is solved by TCP implementation at the endpoints in a distributed way based on congestion signalling from routers. The signaling is typically just the information whether the packet was dropped or not and the sender controls its sending rate based on feedback from the receiver.

End-to-end congestion in itself is not enough to prevent DDoS attacks because the congested link capacity is fairly divided between the flows sharing the link and legitimate flows are given the same or lower priority than the attackers, because the attackers can also circumvent TCP congestion control algorithm implemented at the end nodes. In the future, if routers contain large caches [10] that can store traffic for a long time period, the need for an end-to-end congestion is reduced as the congestion collapse is not anymore possible since the packets are not dropped but cached. However, this does not yet optimize the network resource utilization and latency can grow without bound in the case of congestion.

5.1 Congestion Control as a DDoS Defense

An *aggregate-based congestion control* (ACC) for detecting and controlling high bandwidth flow aggregates generated by DDoS attacks and flash crowds was proposed in [15]. Conventional flow-based protections such as fair queuing are not suitable for this problem as the problem arises from the aggregate of a large number of flows converging from different parts of the network. ACC supports two modes of operation: *local* and *pushback*. The local ACC identifies and controls the throughput of the aggregate at a single router and pushback method can, in addition, request adjacent upstream routers to reduce the rate of the problematic aggreagate. For example, in Figure **??**, the router R1 sends pushback messages upstream toward sources of the heavy traffic flow. Pushback method requires more from the network, but is more effective as the unwanted traffic could in principle be eliminated early in the upstream so that the congested link allows legitimate traffic to flow freely. The ACC mechanism is assumed to be used in conjunction with end-to-end congestion control and is activated only when a problematic aggregate is detected by a sustained congestion at a link. The packets are bundled in aggregates based on source or destination address, address prefix, or protocol type. The actual rate limit used for the identified aggregate can be based on the remaining capacity that should provide a predefined quality of service for the legitimate flows.

ACC cannot always identify the problematic aggregate from other traffic and therefore the rate limit should never be zero. In addition the pushback mechanism requires ACC to be deployed to multiple routers in the network and if domain boundary is crossed, it is uncertain whether the upstream has incentives to implement the pushback protocol. The identification mechanism can also have false positives. The rate-limiter is implemented on the forwarding fast path, which means that it is limited in the per-flow processing and storage use, which can make the system less scalable. Local ACC is deployable but not necessarily incentive compatible at tier-1 ASes as they are compensated for unwanted traffic too. There are no trust issues as each domain can control fully its router hardware. If the malicious traffic is distributed uniformly across the inbound links, the pushback mechanism does not help. In addition the pushback method may overcompensate in case of flash crowd.

For some peer-to-peer overlays, a more powerful rate limiting based method can be used [4, p. 134]. When using

Chord [22] *distributed hash table* (DHT), key-value pairs are stored to DHT nodes based on the one-way hash of the key. That is, legitimate content is randomly and statistically evenly distributed to other nodes that are organized randomly in a ring covering the address space. An example Chord ring is shown in Figure **??**. Chord routing algorithm uses $O(logn)$ routing hops to reach a target node using exponentially longer finger links to other nodes in a Chord overlay of n nodes. Queries for a given key are routed recursively toward the node that is responsible for the part of the address space of the hash of the key. Because the destination of the query messages should be random in the case that all users are innocent, we can start from the assumption that the legitimate traffic in each Chord node should follow as closely as possible the probabilities derived from the lengths of address space segments served by each outgoing finger link. This is a very powerful property and by adding an *admission limit* for the external queries to a node and a *forwarding limit* to queries that have been routed at least one hop in the Chord overlay that force the actual traffic distribution to follow the expected distribution of the legitimate traffic. Basically, those nodes that flood the network are prevented early on in the overlay routing from depleting the resources in an unbalanced way. This is even robust if there are some malicious DHT nodes because the forwarding limit can be applied at every nodes.

If the number of attackers is much larger than the number of legitimate users, then the access of innocent users to the service may be slowed down by malicious queries. In PSIRP, [13] this mechanism was used to secure the control plane of the network, which was implemented as a hierarchical version of Chord. Data plane was secured using Bloom filter based source forwarding headers in packets [11]. When not used in a clean slate aproach such as PSIRP with additional security measures on the underlay network, peer-to-peer networks are always vulnerable to DDoS attacks via the underlay unless these messages are somehow filtered from the network. DHTs have obvious deployability, incentive compatibility, and trust issues as they distributed all over the network and trust relationships between all pairs of nodes, for example based on bilateral agreements, are not practical. On the other hand, DHTs are scalable because they distribute the load.

6 Multipath Routing

If it was possible to use multiple different routes through a network, a more robust and efficient communication could be achieved [8]. For example, it would be possible to avoid congested paths or choose the used path based on specific application needs. This is a realistic goal in the Internet as studies have shown that typically multiple independent routes can be found between end nodes [8]. Multipath routing can be done at different levels of granularity: IP address prefix, TCP flow, or individual packets. However, it should be noted that TCP congestion control algorithm implicitly assumes that packets flow the same route for a certain time period in order to properly use the link capacities available.

In intradomain routing, a *link-state* algorithm is used, where each router shares the whole map of the local network and has a full control over which routes the packets follow. Routers use destination based hop-by-hop forwarding and multiple paths could be achieved, for example, using one of the following methods:

- By configuring a tunnel through multiple routers forming a logical link, that the packets follow. The tunnel could be configured using MPLS labels below IP and the routers could easily be changed simply by modifying the tunnel configuration at routers.
- By explicit routing, which is implemented by source forwarding and storing the whole path in the packet headers.
- By splitting traffic at each router in some proportion, which can increase the utilization of network resources. The ratio for split can be determined by a network management system.
- Alternatively, packets could be sent along all possible paths, but this wastes bandwidth.

In Internet interdomain routing, BGP *path-vector* protocol is used to advertise paths toward IP prefixes. Each path contains the list of ASes that are on the route, which makes it possible to check whether two paths share domains on the AS level. The different routes should be as independent as possible to fully reap the benefits of multipath routing. However, typically only single, preferred path is advertised to customers and peers toward a certain address prefix. This limits the use of BGP for multipath routing in the current Internet or at least it should be modified to advertise more routes to increase the number of choices available to customer networks. On the other hand, stub networks are often multihomed, which makes the originating domain a natural place to implement the path splitting on domain level. In addition, a *deflection point* or tunneling could be used for inter-domain multipath routing in general case. Another option is that sources tag packets to indicate that an alternative path should be used, but this requires more cooperation from the networks.

6.1 Multipath DDoS Prevention

Multipath routing can be used to mitigate DDoS, because the attacker must divide its resources to all available paths toward the victim. Each legitimate user can use a congestion control algorithm for each path simultaneously and balance the traffic to available routes with different sending rates for each. This approach is, however, limited by the fact that typically the convergence of the DDoS trafic must happen near the victim at some point and even if the multiple paths available to a client are disjoint in the core network, they may join near the destination, which considerably reduces the effectiveness of this approach. In addition, when using congestion control in parallel for two routes and splitting the traffic, care must be taken to avoid unstable coupled control loops on the sender side.

In addition, multipath routing was used in [18] to reinforce the security of VoIP calls together with secret sharing scheme in order to provide confidentiality of communication by not

sending all information along a single path. However, their goal was not to prevent DDoS using this approach.

One problem related to multipath routing is operator incentives and the question of who has the control over route selection. The current Internet is largely based on bilateral contracts between ASes and it is generally assumed that technologies that require a large number of ASes to agree on something, are very difficult to deploy. However, stub ASes are increasingly paying to multiple domains for *multihoming* and the added benefits of multipath routing can eventually produce the incentives to provide path selection services for demanding customers [8]. An alternative, clean-slate approach was presented in NIRA architecture [24], where the full control over route selection is given to the endpoints. This is made scalable by splitting the domain level network topology information to *upgraphs* of each node and all valley-free [7] routes can be constructed by joining two upgraphs. Because Internet is mostly built on bilateral contracts between ASes, the valley-free routes contain almost all BGP policy compliant routes even though BGP is capable of describing much more exotic policies. However, the full power of BGP is probably not possible to use for source controlled routing as it is not scalable because of the exponential number of possible paths as a function of the number of domains. In addition to scalability, most approaches to multipath routing require additional computational or storage overhead both on control and data planes. Multipath routing also adds complexity to the network architecture. The ease of use and trust issues are similar to both single path and multipath routing.

7 Conclusion

In this paper, we explained the problem of DDoS attacks, the importance of this problem in the current Internet, and briefly introduced the different approaches to both the attack and defense. We analysed the multitude of challenges posed for a solution to DDoS in a realistic environment. Then we introduced two possible approaches to DDoS defense: multipath routing and congestion control and discussed how they can answer to the aforementioned challenges.

References

[1] M. Blumenthal and D. Clark. Rethinking the design of the Internet: The end to end arguments vs. the brave new world. *ACM Transactions on Internet Technology*, 1(1):70–109, Aug. 2001.

[2] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System. In *ACM SIGCOMM IMC'07*, Oct. 2007.

[3] D. Clark, J. Wroclawski, K. Sollins, and R. Braden. Tussle in Cyberspace: Defining Tomorrow's Internet. *IEEE/ACM Transactions on Networking*, 13(3):462–475, June 2005.

[4] N. Daswani. *Denial-of-service (dos) attacks and commerce infrastructure in peer-to-peer networks (draft)*. PhD thesis, Stanford, Jan. 2005.

[5] C. Douligeris and A. Mitrokotsa. DDoS attacks and defense mechanisms: classification and state-of-the-art. *Computer Networks*, (44):643–666, 2004.

[6] P. Eronen. Denial of service in public key protocols, 2000.

[7] L. Gao. On Inferring Autonomous System Relationships in the Internet. *IEEE/ACM Transactions on Networking*, 9(6):733–745, Dec. 2001.

[8] J. He and J. Rexford. Towards Internet-wide Multipath Routing. *IEEE Network magazine*, 22(2):16–21, 2008.

[9] C. Herley. So Long, And No Thanks for the Externalities: The Rational Rejection of Security Advice by Users. In *NSPW'09*.

[10] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard. Networking Named Content. In *ACM CoNEXT 2009*, 2009.

[11] P. Jokela, A. Zahemszky, C. Esteve, S. Arianfar, and P. Nikander. LIPSIN: Line speed Publish/Subscribe Inter-Networking. In *SIGCOMM'09*, 2009.

[12] F. Kelly. Charging and Rate Control for Elastic Traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.

[13] D. Lagutin, K. Visala, A. Zahemszky, T. Burbridge, and G. F. Marias. Roles and Security in a Publish/Subscribe Network Architecture. In *ISCC'10*, 2010.

[14] F. Lau, S. H. Rubin, M. H. Smith, and L. Trajkovic. Distributed Denial of Service Attacks. In *IEEE International Conference on Systems, Man, and Cybernetics*, volume 3, pages 2275–2280, 2000.

[15] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker. Controlling high bandwidth aggregates in the network. *Computer Communication Review*, 32(3):62–74, 2002.

[16] J. Mirkovic and P. Reiher. A Taxonomy of DDoS Attack and DDoS Defense Mechanisms. *ACM SIGCOMM Computer Communications Review*, 34(2), Apr. 2004.

[17] D. Moore, C. Shannon, D. J. Brown, G. Voelker, and S. Savage. Inferring Internet Denial-of-Service Activity. *ACM Transactions on Computer Systems (TOCS)*, 24(2):115–139, May 2006.

[18] R. Nishimura, S.-i. Abe, N. Fujita, and Y. Suzuki. Reinforcement of VoIP Security with Multipath Routing and Secret Sharing Scheme. *Journal of Information Hiding and Multimedia Signal Processing*, 2010.

[19] J. Rajahalme, M. Särelä, P. Nikander, and S. Tarkoma. Incentive-Compatible Caching and Peering in Data-Oriented Networks. In *ReArch'08*. ACM, 2008.

[20] J. Saltzer, D. Reed, and D. Clark. End-to-end arguments in system design. *ACM Transactions on Computer Systems*, 2(4):277–288, November 1984.

[21] A. Stavrou and A. Keromytis. Countering DDoS Attacks with Multi-Path Overlay Networks. *IA newsletter*, 9(4):26–30, 2006.

[22] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, F. M. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: a Scalable Peer-to-Peer Lookup Protocol for Internet Applications. *IEEE/ACM Trans. Netw.*, 11(1):17–32, Feb. 2003.

[23] G. Urdaneta, G. Pierre, and M. v. Steen. A Survey of DHT Security Techniques. *ACM Computing Surveys*, 2009.

[24] X. Yang, D. Clark, and A. W. Berger. NIRA: A New Inter-Domain Routing Architecture. 15(4):775–788, 2007.