Forwarding Accountability: A Challenging Necessity of the Future Data Plane

Christos Pappas, Raphael M. Reischuk, Adrian Perrig

Forwarding Accountability: A Challenging Necessity of the Future Data Plane

Christos Pappas, Raphael M. Reischuk, and Adrian Perrig

ETH Zürich
{pappasch,reischuk,adrian.perrig}@inf.ethz.ch

Abstract. Forwarding accountability mechanisms pinpoint the sending/forwarding properties of traffic to the entities that send and forward the traffic along a path. In this paper, we take flooding attacks as a use case and describe a proposal to hold senders accountable for the sending rates of their flows. Furthermore, we describe the corresponding challenges, potential solutions, and briefly present the literature in the area of forwarding accountability.

1 Introduction

The Internet started out as a small-scale network among scientists, and turned into a global-scale network for business and private communication alike. At the heart of this success story lies the best-effort delivery service of the network layer – a fundamental property of the Internet architecture. The network does neither provide guaranteed delivery of traffic, nor guaranteed quality of service. The simplicity of this design enabled multiple services and protocols to evolve on top of the minimalistic network layer.

Another property of the Internet architecture is the lack of feedback about the fate of packets. Functionality that detects whether and when packets were delivered is pushed to the end points without aid from the network. However, contrary to best-effort delivery, this design principle has raised trouble with respect to network security. To counter the problems of this design principle, we consider *forwarding accountability* a necessity for the future data plane.

In general, accountability mechanisms associate state and actions to entities, rendering misbehavior detectable, provable, and non-repudiable. Forwarding accountability in particular, associates the sending/forwarding properties of traffic (e.g., latency, bandwidth) to the entity that sends/forwards the traffic (e.g., a host, a router, a switch, or even an Autonomous System), constructing verifiable evidence about how traffic is sent/forwarded. This verifiable information can then be used by users or legal authorities to make informed decisions in cases of misbehavior or poorly performing networks entities.

In this paper, we highlight the importance of forwarding accountability by describing how it would aid in solving two burning issues for the networking community: network neutrality violations and flooding attacks.

Network-neutrality violations. Network neutrality has become an increasingly hot subject in the networking community. Internet service providers (ISPs)

have been accused of blocking [2, 1] and slowing down traffic from specific content providers [4].

Consider the dispute between Netflix and Comcast [3]. Netflix – backed by the media and Internet activists – accused Comcast of deliberately slowing down its video traffic, causing an unacceptable quality of experience for the customers. Comcast denied the blame and attributed the problem to the inability of Netflix's direct ISPs to handle the amount of traffic.

An accountable data plane could alleviate many concerns raised by the neutrality debate [11]. As the exposed forwarding information would be trustworthy, the Internet community would obtain much richer feedback on how "neutral" an ISP really is. Verifiable information could be combined with other higher-level information (e.g., Service Level Agreements) to make an informed judgement about ISP's practices.

Flooding attacks. In recent months, we have observed an increase in the frequency and intensity of flooding attacks rooted in misconfigured or vulnerable Internet services: in February 2014, attackers used misconfigured time synchronization servers to attack Cloudflare with a peak of 400 Gbps. For 2015, Akamai reports a 116.5% increase in total DDoS attacks and a 42.8% increase in the average attack duration compared to the previous year [5].

In an ideal Internet, users could enjoy the benefits of an accountable forwarding plane. Receivers could specify a traffic profile that sources need to adhere to, drawing a clear line for benign traffic and enabling misbehavior detection. In case of traffic profile violations, receivers could provide proofs of misbehavior to the origin and transit ISPs, and ISPs could ensure compliance for misbehaving hosts through traffic shaping.

We provide an example to demonstrate the virtues of forwarding accountability. Consider the topology depicted in Figure 1 and assume that a web server is located in AS_n. We assume that an attacker launches a reflection attack against the server by exploiting the NTP protocol running on vulnerable servers in AS_0. Specifically, the attacker fakes the victim's source IP address and sends NTP commands to the servers within AS_0. Due to traffic amplification, the NTP servers generate traffic that overpowers the victim's resources.

With forwarding accountability in place, each packet is associated with a proof that can later remind every AS on the path that it forwarded the traffic. When the web server reports the attack to the ASes on the path by providing the per-packet proofs, the ASes can acknowledge or deny that they forwarded the malicious traffic. Based on the feedback from the ASes, it is possible to detect misbehavior and narrow down its location on the path. In our example, it becomes clear that the NTP servers in AS_0 sourced the malicious traffic. AS_0 can then drop or deprioritize AS_0's traffic and thus protect the victim web server and the other networks on the path.

Contributions. This paper's focus is on forwarding accountability with respect to flooding attacks. We extend our recent work, FAIR [12], by outlining an ACcountability-based Ddos Protection framework – ACDP – to hold the sending hosts accountable for the sending rates of their originated flows. Furthermore,

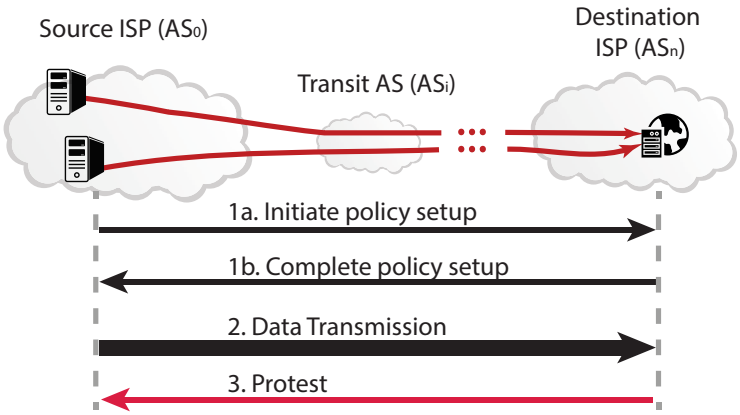

Fig. 1. ACDP Operation.

we describe the corresponding challenges and present the literature in the area of forwarding accountability.

2 Overview of ACDP

We provide a high-level overview of ACDP. In ACDP, communication proceeds in the following three stages (Figure 1).

- **Stage 1 (Setup):** Source and destination hosts set up a sending policy that dictates the sending rate for a specific flow between them.
- **Stage 2 (Transmission):** The source sends its traffic to the destination. Each AS on the path (including the source's and destination's ISP) inscribes information in the packet headers, which serves as a reminder to itself that it has forwarded the packets.
- **Stage 3 (Protest):** If the destination host detects a sending-rate violation, it proceeds to the protest phase and hands the sending policy together with the packet headers to its own ISP. The ISP then contacts other ASes on the path by providing the aggregated proof; the proof eventually identifies the adversary.

2.1 Setup (Stage 1)

Before sending the actual data, communicating hosts set up a sending policy. The sending policy specifies the sending properties that a host should apply to its outgoing traffic towards the communicating peer. The sending properties can be formally expressed through the Token Bucket [8] parameters (the average sending rate, the maximum burst size, and the measurement interval). We consider bidirectional communication channels and thus, both communicating hosts indicate their preferred sending properties. Specifically, the sending policy is constructed as follows:

1. The source[1] initiates the policy setup and constructs a policy packet. It inscribes the sending properties that the destination should adhere to when sending traffic to the source.
2. Each AS on the path (including the host's ISP) indicates its presence on the path by inscribing its identifier in the policy packet. However, it does not interfere with the policy details.
3. The destination completes the policy by filling in its own desired sending properties that the source should adhere to. Then, it sends the policy packet back to the source.
4. Similar to step 2., each AS on the path back to the source – possibly different from the outbound path – indicates its presence.

We assume that communicating hosts sign their information with their private key, in order to make the policy non-repudiable. Furthermore, each AS uses a secret key to protect the integrity of its own information, so that it can later remind itself – without keeping state locally – that it witnessed the corresponding policy.

2.2 Transmission (Stage 2)

With the sending policy in place, hosts start exchanging traffic under the restrictions of the sending policy. We describe the data-plane operations performed by the source, transit, and destination ASes. These operations are applied to each packet.

Source AS. The border routers of the source AS inscribe the Autonomous System Number (ASN), a timestamp, and a sequence number. The timestamp is included to calculate the sending rate of the source in the next stage. Furthermore, in conjunction with the sequence number, it serves as a protection against packet replay from transit ASes. The ASN points to the AS that forwards a packet and constructs a trace of the AS path together with the ASNs of the next on-path ASes. Finally, the inscribed information is protected with a short MAC in the packet, computed with the secret key of the source AS.

Transit and Destination ASes. Each border router of the transit and destination ASes, performs the following operations:

1. The router verifies that the source's timestamp does not deviate from the local time. If the check fails, the packet gets dropped. This check ensures that the source AS does not collude with a customer host in order to conceal an attack by reporting false timestamps.
2. The router inscribes its own information in the packet: its ASN, a short nonce, and a MAC computed over the inscribed information. The nonce serves as protection against replaying the MAC of the AS.

[1] We refer to the host that initiates the connection as the source; and to its communicating peer as the destination.

The destination AS forwards the packet to the eventual recipient, who monitors each flow in order to detect policy violations. In case of a violation, the destination sends the received packets and the policy packet to its own ISP. The ISP proceeds to the protest phase, representing its customer.

2.3 Protest (Stage 3)

In the third stage, destination ASes provide proofs of misbehavior to the source AS and the other transit ASes. It is an offline procedure of at most two rounds.

In the first round, the destination AS sends the policy packet and the packet headers it received from its customer to the source AS. The policy packet contains the transmission properties, and the packet headers contain evidence about the actual transmission properties by the source. The source AS examines the evidence (i.e., verifies its own MAC inscribed in the packets) and approves or rejects the complaint. If the source AS approves the complaint, it can take measures against its misbehaving customer. However, a non-cooperating AS or a replay attack from a transit AS can lead to a rejected complaint in the first round.

A rejection in the first round leads to the second round. The destination AS sends the same information to all ASes on the path. They examine the evidence in the same way as the source AS and approve or reject the complaint. Based on the approvals and rejections, the ASes can determine the root of the problem, because each complaint is accepted at least by the benign cooperating ASes adjacent to the destination, as shown in FAIR [12].

3 Challenges

In this section, we discuss the major challenges related to our proposal. We sketch potential solutions and list the open problems to encourage future research with respect to deployment and performance.

3.1 Deployment Challenges

The first challenge is the required modification of hosts. With ACDP, end hosts have to perform additional functionality compared to the legacy communication paradigm (e.g., under TCP or UDP). Namely, end hosts have to perform a policy setup before a connection starts transmitting data. A change in the network stack of the host's operating system is an unrealistic requirement and we believe that this task can be delegated to a gateway between the host and its ISP. Typically, hosts connect to the Internet through their ISP-provided routers, which act as middleboxes and usually perform additional tasks (e.g., acting as a Network Address Translators or a firewall). Requiring middleboxes to interfere and perform the additional functionality keeps the hosts unmodified and provides a smoother deployment path.

A second challenge is the upgrade of the AS infrastructure required in order to inscribe the additional information in the packet headers; specifically, the MAC computation requires a hardware implementation of a cryptographic engine. Although it is impossible to circumvent this requirement, hardware cryptographic engines are readily available for commodity processors [10]. Since the required technology exists at a low price, the required upgrades would not incur a high procurement cost for ISPs.

The third challenge is a viable business model that provides incentives to ISPs to adopt such a mechanism. We anticipate that security-concerned customers (especially enterprise networks) will be interested in buying service from an ISP that handles and forwards its customers' complaints to the sources of misbehavior. Hence, competition would be the key to promote Accountability-as-a-Service [7]. However, a thorough economic analysis is required to explore the viability of such a security service.

3.2 Performance Challenges

The additional functionality required for ACDP introduces overhead with respect to processing, latency, and bandwidth.

A border router of an AS has to inscribe additional information, which includes the computation of a MAC. We conducted an experiment on one 10 GbE NIC port of a commodity server machine, simulating the required processing, and found that there was not a substantial drop in throughput. Specifically, for 64 byte packets (the minimum packet size, i.e., the maximum packet rate) the switch forwards at 95% of the line-rate. For 128-byte packets and larger, the switch achieves line-rate performance. The initial results indicate that forwarding performance would not suffer from such an accountability framework.

Another performance issue is the increase in latency for communicating hosts. Before the actual communication starts, end hosts must establish the sending policy, which translates to one Round-Trip-Time (RTT). This overhead can be significant for latency-sensitive applications (e.g., video streaming). More extensive research is required to optimize this aspect, but potential solutions include piggybacking the policy packet on the first data packets and embedding a default policy in the DNS records.

Our proposal comes with an increased packet size that leads to bandwidth overhead. The length increase is inevitable, but certain measures can limit the introduced overhead. For instance, a short MAC (4 bits) per ISP is enough to enable misbehavior detection in the context of flooding attacks [12].

4 Related Work

We present the main proposals in the area of forwarding accountability to date.

Goldberg et al. [9] propose end-to-end path quality monitoring in the presence of adversaries. Specifically, an alarm is raised when packet loss and delay exceed a given threshold. The proposal leverages secure sampling, which allows

end points to coordinate their measurements of loss and delay when an on-path adversary delays or drops packets. An alternative protocol uses a sketch to exchange loss measurements securely and efficiently in adversarial scenarios, accompanied by a theoretical analysis about their accuracy vs. overhead tradeoffs. In addition, these protocols make sensible assumptions for networking environments: no symmetric paths, no processing at forwarding devices, and configurable storage overhead based on accuracy target. However, these protocols do not provide granular performance reports for smaller path segments and do not localize misbehavior.

FAIR [12] is a forwarding accountability mechanism that pushes stricter security policies to ISPs. The source and destination ASes set up a communication channel with a corresponding sending policy, which can specify sending properties (e.g., average sending rate) or forbid abnormal packet headers used for malicious activity (e.g., Christmas tree packets). Transit ASes on the path mark packets and in case of policy violations, the packets are used as a proof of misbehavior. FAIR comes with an implementation that introduces low bandwidth overhead and can switch packets at a line-rate of 120 Gbps. However, the proposal does not allow proving misbehavior at the granularity of flows, and thus cannot be used to identify individual misbehaving flows or hosts.

AudIt [6] proposes an accountability interface, provided by ISPs, that gives loss and delay feedback to the traffic sources. The framework relies on statistics reports from ISPs, without requiring complicated key establishment. However, the proposal is based on aggregation of flow information, and thus ISPs can hide their lies since they report mean values.

References

1. FCC Fines Telecom that Blocked Vonage VoIP Calls. http://bit.ly/1MokIA4 (Mar 2005)
2. AT&T Faces Formal FCC Complaint for Blocking Cellular FaceTime Use. http://bit.ly/1JYNxpt (Sep 2012)
3. Comcast vs. Netflix: Is This Really About Net Neutrality? http://cnet.co/T6JuPP (May 2014)
4. Netflix Performance on Verizon and Comcast Has Been Dropping for Months. http://bit.ly/1URc8zR (Feb 2014)
5. Akamai: Q1 State of the Internet - Security Report. "http://bit.ly/1RhrFWs" (2015)
6. Argyraki, K., Maniatis, P., Irzak, O., Ashish, S., Shenker, S.: Loss and Delay Accountability for the Internet. In: Proceedings of ICNP (Oct 2007)
7. Bender, A., Spring, N., Levin, D., Bhattacharjee, B.: Accountability as a service. In: Proceedings of the 3rd USENIX Workshop on Steps to Reducing Unwanted Traffic on the Internet. pp. 5:1–5:6. SRUTI'07, USENIX Association, Berkeley, CA, USA (2007), http://dl.acm.org/citation.cfm?id=1361436.1361441
8. Cisco: Cisco Policing and Shaping Overview. "http://bit.ly/1HOHr9V" (May 2015)
9. Goldberg, S., Xiao, D., Tromer, E., Barak, B., Rexford, J.: Path-quality Monitoring in the Presence of Adversaries. In: Proceedings of ACM SIGMETRICS (2008)

10. Gueron, S.: Intel Advanced Encryption Standard (AES) New Instruction Set. "https://software.intel.com/sites/default/files/article/165683/aes-wp-2012-09-22-v01.pdf" (Mar 2010)
11. Pappas, C., Argyraki, K., Bechtold, S., Adrian, P.: Transparency Instead of Neutrality. In: Proceedings of ACM HotNets (Nov 2015)
12. Pappas, C., Reischuk, R.M., Perrig, A.: FAIR: Forwarding Accountability for Internet Reputability. In: Proceedings of IEEE ICNP (Nov 2015)