The Location/Identity Split is Useful for Middleboxes, Too

Michael Walfish and Hari Balakrishnan
MIT Computer Science and Artificial Intelligence Laboratory

1 Introduction

In this short paper, we observe that the location/identity split, an architectural distinction that is most often invoked to support mobile and multi-homed hosts (*e.g.*, [8, 10]), also permits middleboxes, such as NATs and firewalls, to be coherently incorporated into the Internet architecture. This observation arose in the context of our work on the *Delegation-Oriented Architecture (DOA)* [11]. We first give an overview of DOA, then describe an example middlebox under the DOA framework, and then wander into architectural musings.

2 Background

For detail on why middleboxes and the Internet architecture are not in harmony, see [2] and [3, 6, 7, 9].

DOA is an incremental extension to the Internet architecture that coherently accommodates network-level intermediaries like NATs and firewalls. DOA is based on two main ideas. First, all entities have *unique identifiers*, separate from their locations, and packets carry these identifiers. Second, DOA allows senders and receivers to express that one or more intermediaries should process packets en route to a destination. This *delegation* lets DOA embrace middleboxes as first-class network citizens that are explicitly invoked and need not be physically interposed in front of the hosts they service.

The mechanics of DOA are as follows. DOA's components are depicted in Figure 1. End-hosts receive identifiers called EIDs. As in HIP, source and destination EIDs are carried in packets in a header (shown in Figure 2) after the IP header and before the transport header. The destination EID field can hold a logical stack of identifiers (as in i3). An EID is the secure hash of a public key (as in HIP). To communicate with an end-host identified by an EID e, a prospective peer p resolves e to either an IP address or else to another EID, f (which would ultimately be resolved to an IP address, perhaps through additional resolutions); more specifically, an EID resolves to a data structure called the `erecord`, depicted in Figure 3. p sets the destination EID field in the packet to e and the destination IP address field to the ultimate outcome of the resolution. DOA requires an EID resolution infrastructure and envisions using distributed hash tables (DHTs) for this purpose. See [11] for a detailed description of DOA.

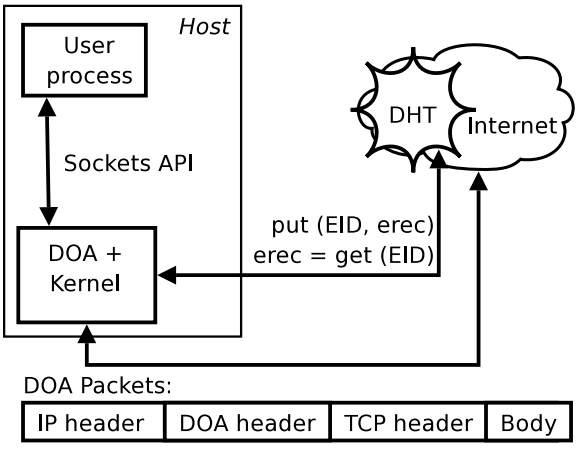

Figure 1: High-level view of DOA design.

3 Example: Remote Packet Filter (RPF)

To motivate the power of the location/identity split in supporting middleboxes, we start with an example: a remote packet filter (RPF) that gives similar functions to today's firewalls but need not be physically interposed in front of the hosts for which it provides filtering service. The RPF is a basic application of DOA's mechanisms; it is depicted in Figure 4.

A user (or representative of the user, *e.g.*, corporate IT

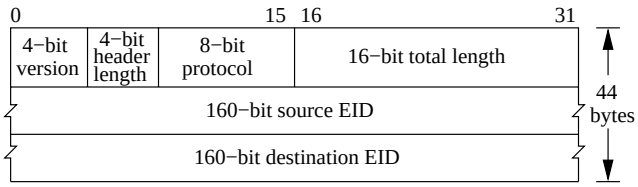

Figure 2: Example DOA header with no stacked identifiers.

EID: 0x345ba4d...
Target: EID$^+$ or IP address
Hint: *e.g.*, IP address
TTL: time-to-live, a caching hint

Figure 3: The `erecord`. An EID resolves to an `erecord`.

staff) wanting remote firewall service creates a mapping in the EID resolution infrastructure from the end-host's EID, *e*, to the RPF's EID, *f* (or to the RPF's IP address, but then if that IP address changes, the resolution of *e* will be incorrect). This end-host expresses its actual network location either by (1) communicating directly with the RPF and telling it about the association between *e* and *i* or (2) putting *i* in the Hint field in the `erecord`.

When a sender attempts to contact *e*, it first looks up *e* in the EID resolution infrastructure, sees that *e* maps to *f*, and then further resolves *f* to an IP address (which might involve intermediate resolution steps, depending on whether the RPF itself has delegates). In the simple case in which *f* resolves directly to an IP address *j*, the sender forms IP packets with destination address *j* and destination EID *e*. Note that *f* must be in the *stack* of identifiers since the host given by *j* may actually be the RPF's *delegate* rather than the RPF itself, and the host given by *j* thus needs *f* to determine where next to send the packet.

When receiving IP packets, the RPF extracts the destination EID *e* from the DOA header, looks up the set of rules associated with *e*, and finally applies these rules; examples of such rules are filters to block or accept traffic based on IP- or transport-layer fields. The result is "passing" or "failing" a packet. When packets "fail", the RPF drops the packet. The RPF attests that packets "passed" by inserting into the packet a MAC (keyed with a secret shared between RPF and end-host) taken over the packet. The RPF then rewrites the packet's destination IP address and sends the packet to the end-host, which applies a single rule: redoing the MAC computation and testing whether the result matches the MAC in the packet. The end-host ignores packets that fail this test; thus, only packets that have been vetted by the RPF are processed by the end-host's networking and application software. The MAC is carried in a DOA security header, which extends the standard DOA header.

The RPF depends on both of DOA's core mechanisms: first, because of unique host identifiers, the RPF has a way (namely the destination EID field) to distinguish among hosts, allowing it to apply host-specific rules and then send the processed packet to the correct destination. Second, the delegation primitive is what allows the RPF to be invoked in the first place.

4 More Generally . . .

As the RPF example shows, the separation between location and identity gives rise to the following high-level architectural properties:

- The architecture creates per-host identities, which lets hosts refer to themselves and pass to other hosts a handle that these other hosts can use to direct packets back to the original host. This property follows from the existence of EIDs.
- The architecture allows hosts to express "to reach me, send your packets there." (The architecture also allows hosts to express, "to reach me, send your packets to these intermediaries in sequence.") This property follows from fact that EIDs are resolved.
- The architecture lets intermediaries know where next to send the packet. This property follows from the fact that EIDs (or EID stacks) are carried in packets.

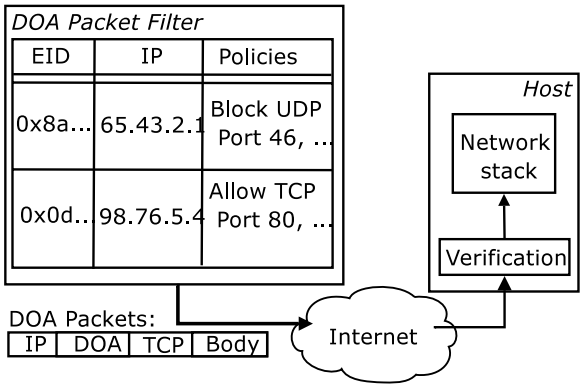

Figure 4: Packet filtering under DOA using delegation. End-hosts apply a simple verification rule, not a collection of them.

The above three properties might seem trivial, but together they lead to coherent support for intermediaries. Indeed, their absence in the current Internet architecture is one of the reasons why middleboxes must today either be on-path or else require application-specific machinery (*e.g.*, VPNs, whose interfaces differ across vendors) to be off-path.

5 Why Doesn't DOA Use . . .?

The previous section implies, nebulously, why the location/identity separation—which is mostly what DOA accomplishes—is *sufficient* to coherently support middleboxes. In this section, we briefly consider hypothetical alternatives to DOA with the goal of showing why the location/identity separation is *necessary* to coherently support middleboxes.

IPv6 addresses. Could moving the Internet to IPv6 address some of the motivations for DOA? IPv6 addresses are globally unique, which addresses one reason for today's NATs. However, they are still location-dependent and thus are not a good choice for persistent host identifiers. There are at least two reasons to require that host identifiers be topology-independent: (1) Hosts should be able to pass around handles to themselves

without worrying that those handles could be invalidated by a location change and (2) As in the RPF example, hosts might wish to map their identifiers to topology-independent identifiers for their delegates. If the target of the EID resolution is itself topology-dependent, then the EID owner has to update a mapping whenever its delegate changes location.

DNS names. One possibility would be to claim that the status quo (or the status quo augmented with IPv6) is enough and that today's human-friendly DNS names are handles that hosts can pass around to each other. We disagree that the status quo is sufficient, for two reasons: (1) intermediaries cannot use DNS names to figure out where next to send packets that arrive at these intermediaries since the DNS names are not carried in the packet and (2) again because the DNS name is not carried in the packet, the receiver cannot resolve the source's identifier to determine how to send a packet back to the sender, so the packet must traverse the same intermediaries on the way back to the sender. This requirement is inflexible and conflicts with our goal of flexible invocation of intermediaries.

(One could imagine DNS names being carried in packets, along the lines of architectures like TRIAD [5] and IPNL [4]; these proposals incorporate a location/identity split but do so with identities that are human-readable, in contrast to the flat identifiers that carry cryptographic meaning, as are used by HIP and by DOA.)

6 Conclusion

Coherently supporting intermediaries requires that packets carry location-independent identifiers and that these identifiers resolve to actual locations or other identifiers. These requirements are a restatement of the location/identity split. In other words, we have argued that the location/identity split is necessary for coherently supporting intermediaries.

We conclude with two open questions that are at the intersection of HIP's focus and DOA's focus:

- *Mobility:* We argued that the location/identity split helps intermediaries but we did not detail a solution to mobility, and, indeed, the DOA design and implementation do not allow in-band changes to the binding between identity and location. What would it take to extend DOA to mobile scenarios where hosts *and* intermediaries could move during a connection?

- *Application-level intermediaries:* DOA focuses on network-level intermediaries. Can the location/identity separation, combined with the architectural observations made by HIP and by DOA, result in a coherent architecture for application-level intermediaries? The ideas in [1] are promising but inchoate.

References

[1] H. Balakrishnan, K. Lakshminarayanan, S. Ratnasamy, S. Shenker, I. Stoica, and M. Walfish. A layered naming architecture for the Internet. In *ACM SIGCOMM*, Portland, OR, Aug. 2004.

[2] B. Carpenter and S. Brim. Middleboxes: Taxonomy and issues, Feb 2002. RFC 3234.

[3] B. Ford, P. Srisuresh, and D. Kegel. Peer-to-peer (P2P) communication across middleboxes, Oct. 2003. Internet draft `draft-ford-midcom-p2p-01.txt` (Work in progress).

[4] P. Francis and R. Gummadi. IPNL: A NAT-extended Internet architecture. In *ACM SIGCOMM*, San Diego, CA, Aug. 2001.

[5] M. Gritter and D. R. Cheriton. TRIAD: A new next-generation Internet architecture, http://www-dsg.stanford.edu/triad/, July 2000.

[6] T. Hain. Architectural implications of NAT, Nov. 2000. RFC 2993.

[7] K. Moore. Things that NATs break, http://www.cs.utk.edu/~moore/opinions/what-nats-break.html, as of May 2004.

[8] R. Moskowitz and P. Nikander. Host identity protocol architecture, Sep 2003. draft-moskowitz-hip-arch-05, IETF draft (Work in Progress).

[9] P. Srisuresh and M. Holdrege. IP network address translator (NAT) terminology and considerations, Aug. 1999. RFC 2663.

[10] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet Indirection Infrastructure. In *ACM SIGCOMM*, Pittsburgh, PA, Aug. 2002.

[11] M. Walfish, J. Stribling, M. Krohn, H. Balakrishnan, R. Morris, and S. Shenker. Middleboxes no longer considered harmful. In *Proc. OSDI 2004*, San Francisco, CA, Dec. 2004.