IP Options are not an option

Rodrigo Fonseca
George Manning Porter
Randy H. Katz
Scott Shenker
Ion Stoica

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2005-24
http://www.eecs.berkeley.edu/Pubs/TechRpts/2005/EECS-2005-24.html

December 9, 2005

Acknowledgement

We would like to thank Mark Allman for his invaluable contribution to this work.

IP Options are not an option

Rodrigo Fonseca George Porter Randy H. Katz Scott Shenker Ion Stoica

{rfonseca,gporter,randy,shenker,istoica}@cs.berkeley.edu
Computer Science Division – Univ. of California
Berkeley, CA 94720-1776

Abstract

A wide variety of enhancements to the Internet architecture have been proposed over the past several years, many of which require attaching metadata, or state, to packets as they flow through the network. Examples of such extensions are IP traceback[10] and XCP[6]. The IP specification supports an "options" mechanism as an extensible way to couple state with packets. However, as we will show in this paper, IP options are not well supported in the Internet. We make use of the PlanetLab planetary scale network testbed[2] to quantify the fate of IP-option enabled packets in the wide-area. We measured wide-area paths with both standard IP packets and packets with options. We discovered that approximately half of Internet paths drop packets with options, raising serious dependability issues. Surprisingly, our findings indicate that it is feasible to restore support for options in the wide-area. We discovered that the core of the network drops very few options packets, with the vast majority of those drops occurring in edge AS networks. Furthermore, these drops are concentrated in a minority of the ASes.

1 Introduction

As the Internet has evolved, researchers have proposed a variety of extensions to the original Internet Protocol (IP). Many of these extensions, such as IP traceback[10], XCP[6], Quick-start[5], SCORE[11], CETEN[1], and SIFF[13], require storing protocol-specific state in the packet header. Some of these extensions would be beneficial to end-users, and others to ISPs as well, providing features such as DoS protection, enhanced performance, QoS primitives, to name a few.

IP was designed to support such extensibility through the use of "options". An option is a variable-length piece of data that is stored in the IP header and is associated with a particular extension type. IP packets can store multiple options, as long as the total length of all options is no greater than 40 bytes.

Options were designed to be incrementally deployable: a router or an end-host that did not understand a particular option was supposed to ignore that option and simply forward the packet. Unfortunately, this is not what happens in today's Internet. In routers, checking the IP options is an expensive operation, and so many routers choose to drop the packets with options, rather than forwarding them. Some Cisco routers include a "drop options" command which drops all option-enabled packets. The manual page for this command[4] gives a clue as to why such a drastic step is taken:

> *Drop mode filters [options] packets from the network and relieves downstream routers and hosts of the load from options packets.*

As we will show, network operators make use of this drop mode, presumably to prevent their routers' CPU from becoming overloaded by handling options packets. While such a step is locally beneficial, it leads to global dependability problems. Although options are fully standardized as part of RFC 791[9], researchers have anecdotal evidence that they cannot rely on them in practice. This has led them to choose other ways of coupling state with packets, including overloading reserved bits in the IP header. This approach cannot be shared among multiple extensions, and leads to dependability problems of its own.

The lack of options support in the wide-area hampers innovation, since without them, coupling state with packets becomes non-trivial. In this paper, we discuss two common option types: "Record route" and "Timestamp". These options are useful for debugging path and performance issues in the wide area. Unlike traceroute probes, these options can also be used by the destination to infer path properties. Although not presented in this paper, there are some other common options, e.g., Loose source route. As discussed above, there are also a variety of new Internet extensions

that could easily make use of a dependable IP options mechanism.

We are not aware of a previous, comprehensive study of the effect of options on Internet traffic. Thus, we present experimental observations of such traffic on the PlanetLab network. As we will show, options are currently not a reliable way of coupling state with packets in the wide-area. Our results show that approximately 50% of wide-area paths drop options packets. However the situation is not as bad as it seems at first, in that fixing it would not be infeasible. As we will show, surprisingly most option packets are dropped at the edges of the network, rather than in the core. Also, the vast majority of path drops are concentrated in only 15% of the ASes we observed.

We first discuss related work in Section 2. Then, in Section 3, we present our measurement methodology. Our observations are presented in Section 4. Lastly, we present our conclusions in Section 5.

2 Related Work

Space for packet annotations was included in the Internet Protocol version 4[9]. IPv4 includes support for up to 40 bytes of options. This option space is divided into one or more variable length entries each consisting of a type, and possibly length and data fields. IPv6[3] generalizes the use of options to include two distinct sets of headers: hop-by-hop headers and end-to-end headers. Hop-by-hop headers are intended for intermediate routers, and thus are the only ones that affect the core of the network.

The authors of [8] studied the behavior of TCP in the current Internet environment. Their measurement infrastructure included both active and passive measurements taken from web server connections. As part of their work, the found a 66% success rate for the Record Route option, and a 63% success rate for the Timestamp option. They also found that introducing a non-standard option led to only a 30% success rate. Our work differs from theirs in that we attempt to isolate where those drops occur.

3 Methodology

Measurement infrastructure In this work we collected traceroute measurements among PlanetLab [2] nodes, repeating each measurement within a short interval with and without IP options in the probe packets. As our results show, this data allowed us to examine the reachability along these wide-area paths for packets with IP options as compared to normal packets, as well as the impact of options on latency.

We chose one reachable node per PlanetLab site, and ended up with a selection of 160 machines. We then instructed each machine to perform traceroutes to all 160 machines in the list. In the experiments, we used a modified version of traceroute [12] that sent ICMP packets with no options, or with one of the standard Nop, Timestamp (TS), and Record Route (RR) options. We used ICMP packets instead of TCP packets because we encountered some routers with erratic behavior when faced with TCP SYN packets with options (see Section 4). In the remaining of the paper we refer to these as normal traceroute, Nop traceroute, and so forth.

We were able to collect results from 139 of these machines. Of the possible 22,240 paths that resulted, discounting the measurements that did not complete, we were able to get collect data for 21,051, or 94.65% of the total possible pairs. While these are not the complete set of measurements among all possible sites, they represent a significant fraction of the possible paths among these PlanetLab sites. These 21,051 traceroute measurements comprise a total of 2,964,502 ICMP round-trip time measurements involving 7,524 IP addresses.

Autonomous system inference Although the data we collected is based on IP addresses, we chose to map those addresses into Autonomous System (AS) numbers. An AS is a part of the Internet that is under one organization's control. To perform this mapping, we collected measurements from the RouteViews database (http://www.-routeviews.org/). RouteViews collects BGP[7] announcements from various points in the Internet. Using this BGP data, we were able to resolve the AS number of 7,395 of the collected IP addresses (out of a total of 7,524), for a total of 241 different ASes.

We classified an AS as a "Source" AS if any of our traceroute probes originated in that AS. Likewise, an AS that was the destination of any of our probes is a "Destination" AS. "Edge" ASes are the union of those two sets. An AS is considered a "Transit" AS if it is not an edge AS.

Drop location inference To determine the link where a packet with options was dropped we compare the execution with that of the normal traceroute for the same path. If the traceroute with a given option, say Nop, stops at node 'C', and the corresponding normal traceroute (without options) goes through 'C' and the next hop to 'C' is 'D', then we infer that the packet with options was either dropped in the outgoing port of 'C', or in the incoming port of 'D'. When localizing the drop in the AS level path, we say that a drop occurred close to the source (or destination) AS if either end of the link where the drop occurred is part of the source (or destination) AS. For the remainder of the paths, we say that the drop occurred in a 'Transit' AS.

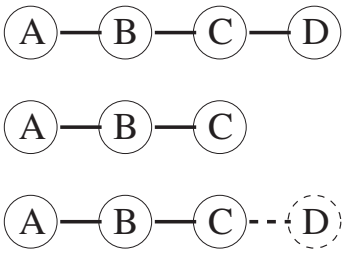


Figure 1. The drop location for options packets is inferred by comparing the partial path to longer paths produced by normal traceroutes from the same source-destination pair

Limitations We chose to deploy our measurement infrastructure on PlanetLab. As such, the transit networks we were able to measure tend to connect nodes in primarily academic institutions. While PlanetLab is a popular testbed often used by the research community, care must be taken in generalizing this data to the Internet as a whole.

We chose to measure paths using end-to-end measurements. Since we had no information on the amount of traffic between those points, we treated every path equally. In reality, some paths carry more traffic than others, and so options support on those paths is more critical.

Lastly, we only consider the effect of options on the underlying wide-area path. We do not measure whether routers in the Internet act on and correctly process the semantics of the options (e.g., adding timestamps, or recording the route of the packet). The scope of this paper is simply to study the dependability of paths carrying options-enabled traffic.

4 Results

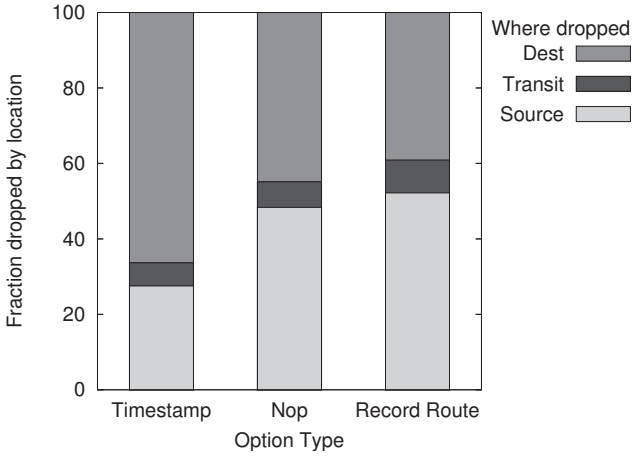
Reachability of IP Options In our measurements, normal traceroute reached the destination in 17,457, or 82.92% of the 21,051 total paths. In the remainder of this section we call these paths *working paths*, and this is the set of paths we consider. For completeness, though, we mention that of the 3,594 paths that did not have normal traceroute reachability, 173 allowed traceroute with the Nop option, 146 with the record route option, and 103 with the Timestamp option.

Table 1 shows the fraction of the working paths that allowed the three types of options. We see that the reachability varies between 34 and 67% of paths, numbers that at first sight make the use of options not an option.

Where the options are dropped Next we look at where along the AS-level path the packets with IP options are dropped. In our analysis below we only include the paths that had at least one transit AS, because we wanted to characterize the behavior of the transit ASes. The thus omitted paths corresponded to between 7 and 9% of the paths with drops. Further, we only included those paths that dropped each type of option, *while allowing the traceroute without options to complete*.

Option	Paths	Success Rate
Normal	17,457	100.00
Nop	11,671	66.86
Timestamp	6,049	34.65
Record Route	9,430	54.02

Table 1. Fraction of working paths that support IP Options



Option	Source	Transit	Destination	Paths w/ drops
Timestamp	27.53%	6.14%	66.33%	8,960
Nop	48.36%	6.77%	44.88%	4,109
Record Route	52.18%	8.73%	39.09%	5,874

Figure 2. Where in the AS paths packets with the different options are dropped: close to the source AS, close to the destination AS, or somewhere else in the middle of the network.

The results, shown in Figure 2, are quite interesting. In accordance with Table 1, Timestamp was the most dropped option, and Nop the least. There is a difference in behavior for the different options, with Timestamp being dropped most frequently close to the destination AS, and Nop and Record Route close to the source AS. What is consistent is the fact that very few of these drops occur at transit ASes: between 85 and 92% of the drops of packets with options occur at the edge ASes. What this result suggests is that the

core of the network seems to allow options to flow, and it is the access ASes that are responsible for the drops that we see. To investigate this further, we now look at how each AS treats packets with options.

Who is dropping the options We now look at the distribution of *path success rate* per AS. As before, we only look at paths that had at least one transit AS. We define path success rate as the number of paths with options that leave the AS divided by the total number of paths that enter the AS. The difference in the two corresponds to the paths that are dropped at the AS. For the source ASes, the paths entering the AS are the paths initiated at the AS, and for the destination ASes, the paths leaving the AS are those that reach the destination within the AS.

Figure 3 shows the CDF for the distribution of path success rate for the three AS groups and option type. Roughly speaking, we found that the ASes broadly fall in two classes: they either drop a large fraction of packets with options, or allow almost all packets to pass through. This is certainly true for the source ASes, as we can see in the curves in Figure 3(a): between 11 and 13% of the ASes allowed less than 2% of the paths through, and 75% of the ASes allowed more than 94% of the paths with options. The median success rate for the sources varies between 98.5% for the Timestamp option and 99.2% for the Nop and RecordRoute options.

For the transit ASes, as we saw previously, there were very few drops. In the corresponding CDFs in Figure 3(b), we see that almost all ASes have a very high success rate. For all options, 95% of the transit ASes allow more than 92% of the paths, and between 87% and 92% of the ASes allow more than 99% of the paths. The success rate is 100% for all options.

The option drop behavior at the destination ASes is qualitatively similar to the source behavior for the Nop and Record Route options, except that for the latter the median success rate drops to about 95%. The support for Timestamp at the destinations, however, is considerably worse: the median success rate is only 54% of paths. 15% of the ASes drop more than 99% of the packets, and only 8% of the ASes that allow all paths through. The remaining 77% of the ASes have almost an even distribution of path success rates, as can be seen in Figure 3(c). We don't know the reason for this, but it shows that the treatment of options varies for the different types.

Figure 4 plots the absolute number of paths in and out of each source and destination AS, for the different options. The number of paths dropped in each AS can be seen as the visible dark section of the bars, the difference between the paths in and out. The number of paths in and out of each AS are generally close to multiples of 160, which is the total number of measurements we did from each source. Some of the ASes clearly have different behavior for different sources or destinations, suggesting that the dropping is actually occurring closer to the source or destination, and not as a global AS policy. An example of this is the first AS listed in Figure 3(a): this AS has 4 different sources in our experiment, and nearly all paths from one of these are dropped, while the opposite is true for the other sources.

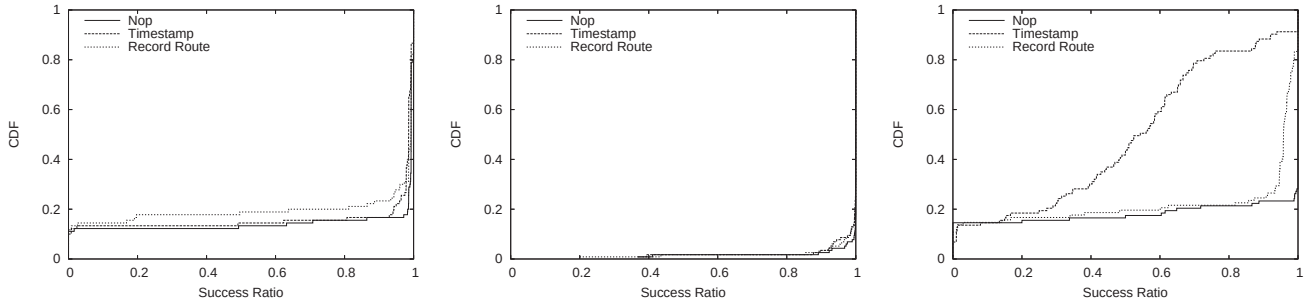
Latency impact of IP Options We studied the effect of options on network latency, since router load is often cited as a motivation for dropping options packets. Figures 5 and 6 summarize our results. We found a measurable, but relatively small, latency penalty for options packets. The first figure plots the ratio of the average one-way latency of options packets versus standard network packets. Values above 1.0 indicate that options packets suffer greater latency than normal packets. Indeed, approximately 90% of cases fall into this category.

Interestingly, for some paths, options packets had less latency than normal packets. This can especially be seen in Figure 6. This figure is a scatter plot of ping times, with normal packet ping times on the X axis, and options-packets on the Y axis. Points above the $y = x$ line indicate that options slow the packet down en route to the destination.

What is striking about this plot is the set of points parallel to, and approximately 200ms below, the X-Y line. After further examination, we isolated these cases to two routers. Probes to these routers without options enabled take approximately 200ms more time to generate a response than options-enabled probes. This is true for multiple sources and for multiple destinations transiting these routers. We tried these probes for several days, and saw consistent results each time.

Anomalous Behaviors As a final result of our empirical study, we came across some examples of option-handling behaviors that were far from expected. The first is the case outlined above in which the generation of ICMP time-exceeded messages for packets without options took more time than for packets with options.

The second example deals with a middlebox that intercepts TCP packets. In a first version of our experiments, we started using traceroute with TCP SYN probing packets. The use of options in these packets triggered what seems to be a bug in a particular middlebox. This box was two hops away from the source of the traceroute, and with a normal SYN packet, the traceroute terminated with two hops, with the latency of the middlebox, but with the IP address of the destination. The middlebox was intercepting the SYN packet and responding with a SYN-ACK, spoofing the source IP address of this response to look like the destination. When we added the Nop option, the traceroute went all the way to the destination, 18 hops away. The surpris-



(a) Source ASes (b) Transit ASes (c) Destination ASes

Figure 3. CDF of path success rate for option packets, for source, transit, and destination ASes

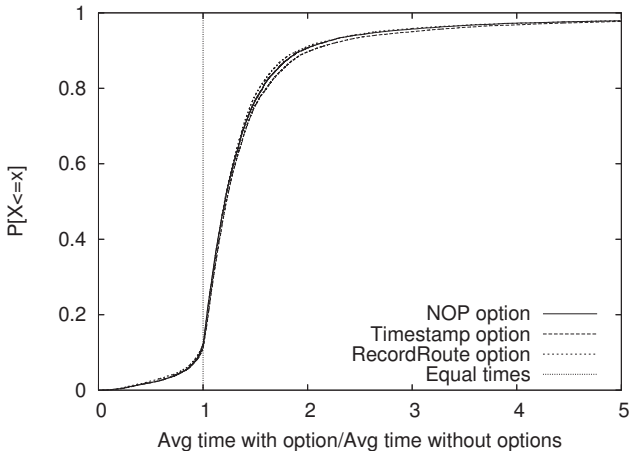


Figure 5. CDFs of ping time with options relative to the ping time without options, for all pairs with at least three measurements.

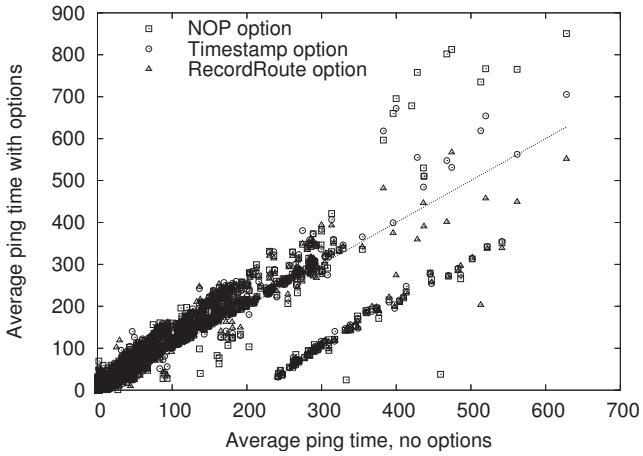


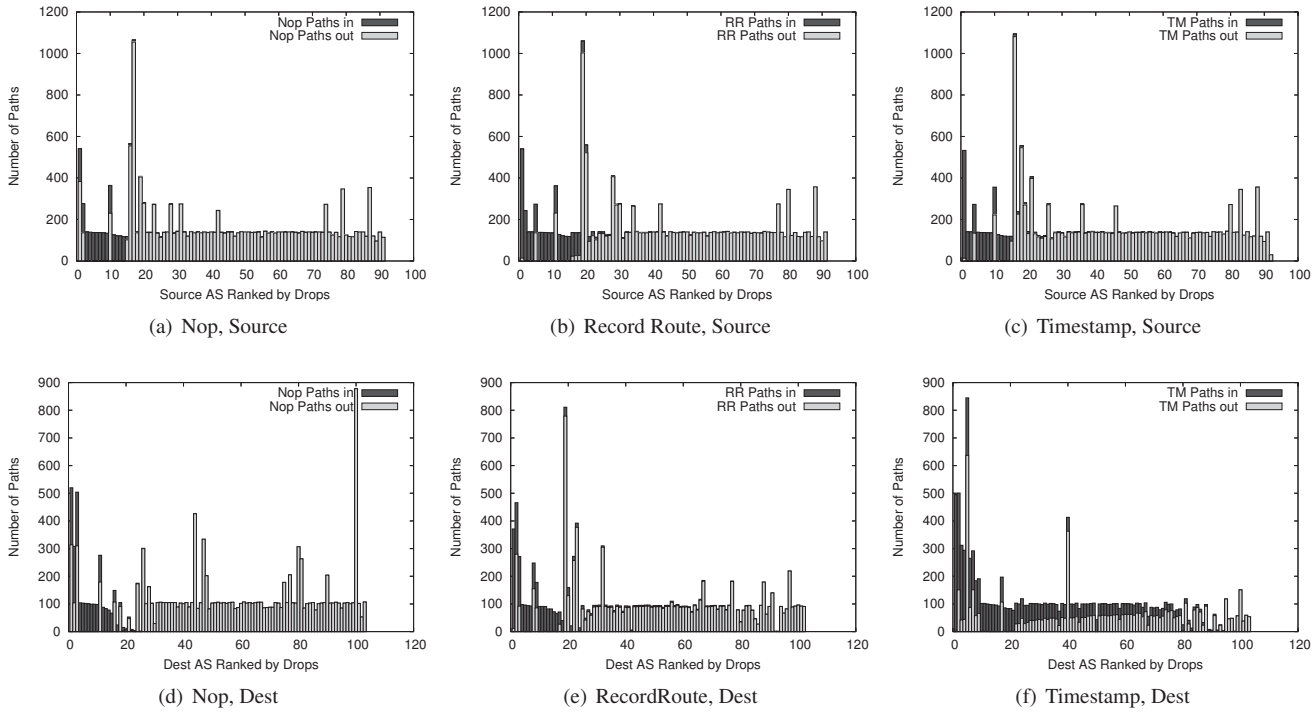
Figure 6. Scatterplot of the ping time with options versus the ping time without options, for all pairs with at least 50 measurements.

ing thing happened when we added the Timestamp option: the traceroute would end with either 5 or 8 hops, at random, with a SYN-ACK apparently from the destination, but with the latency of the same middlebox, two hops away! We believe that this is due to the middlebox TCP interception code not having been tested at all with different kinds of IP options, since the presence of options in the packet shifts the start of the TCP header by the (variable) length of the options space.

What these examples suggest is that apart from dropping packets with options, routers and middleboxes may have different or buggy behavior when dealing with them, which raises questions about the general dependability of options for mission critical applications.

5 Conclusion

We have studied the dependability of IP options-enabled network packets in the Internet. We found that overall, approximately half of Internet paths drop packets with options. Additionally, we found that options cause a measurable, but small, increase in end-to-end latency. A surprising result of our study was that approximately 90% of option-enabled packets drops occurred at the edges of the network, not in the core, and that a small fraction of the ASes are responsible for a vast majority of the drops. Because we were able to pinpoint where options were dropped, our results show that while it is true that IP options are not dependable for wide-area applications to make use of them, fixing the situation would be feasible, and would involve a minority of edge ASes.



(a) Nop, Source (b) Record Route, Source (c) Timestamp, Source

(d) Nop, Dest (e) RecordRoute, Dest (f) Timestamp, Dest

Figure 4. Absolute number of paths into and out of each source and destination ASes. The difference between the two bars (the dark area) is the number of paths that were dropped inside each AS.

Acknowledgments

We would like to thank Mark Allman for his invaluable contributions to this work.

References

[1] Mark Allman. A(nother) technique for improving transport protocol performance in lossy networks, http://-www.icir.org/mallman/talks/ceten-icir-lunch.ps. *ICIR Wednesday Lunch*, April 2004.

[2] Brent Chun, David Culler, Timothy Roscoe, Andy Bavier, Larry Peterson, Mike Wawrzoniak, and Mic Bowman. Planetlab: an overlay testbed for broad-coverage services. *SIGCOMM Comput. Commun. Rev.*, 33(3):3–12, 2003.

[3] S. Deering and R. Hinden. RFC 2460: Internet Protocol, Version 6 (IPv6) specification, December 1998.

[4] Cisco Manual Page: ACL IP Options Selective Drop, http://tinyurl.com/dnkbz.

[5] A Jain, S Floyd, M Allman, and P Sarolahti. Quick-Start for TCP and IP. IETF draft-tsvwg-quickstart-00.txt, May 2005.

[6] Dina Katabi, Mark Handley, and Charlie Rohrs. Congestion control for high bandwidth-delay product networks. In *SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 89–102, New York, NY, USA, 2002. ACM Press.

[7] K. Lougheed and Y. Rekhter. RFC 1267: Border Gateway Protocol 3 (BGP-3), October 1991.

[8] Alberto Medina, Mark Allman, and Sally Floyd. Measuring the evolution of transport protocols in the internet. *SIGCOMM Comput. Commun. Rev.*, 35(2):37–52, 2005.

[9] J. Postel. RFC 791: Internet Protocol, September 1981.

[10] Stefan Savage, David Wetherall, Anna R. Karlin, and Tom Anderson. Practical network support for IP traceback. In *SIGCOMM*, pages 295–306, 2000.

[11] Ion Stoica. *Stateless Core: A Scalable Approach for Quality of Service in the Internet*. PhD thesis, Carnegie Mellon University, December 2000.

[12] tcptraceroute, http://michael.toren.net/code/tcptraceroute/.

[13] Avi Yaar, Adrian Perrig, and Dawn Song. An endhost capability mechanism to mitigate DDoS flooding attacks. In *Proceedings of the IEEE Symposium on Security and Privacy*, May 2004.