On the Significance of Layer-3 Traffic Forwarding

Salim Mohamed[1(✉)], Saptarshi Das[1], Subir Biswas[1], and Osama Mohammed[2]

[1] Electrical and Computer Engineering, Michigan State University, East Lansing, USA
mohame26@msu.edu, {dassapta,sbiswas}@egr.msu.edu
[2] Service Delivery and Management, Innovaway, Napoli, Italy
osama.mohammed@it.ibm.com

Abstract. Designing performance-enhanced and large-scale overlay networks over the conventional IP substrate encounters different implementation obstacles put in place by Internet Service Providers (ISPs). These include lack of proper privileges and restrictive routing policies that prevent the overlay services from being deployed easily. The evolution of Software Defined Networks (SDNs), however, helps to address these concerns by simplifying the mechanism for overlay router design. In this paper, we have included an analysis of 18,906 delay traces from a network of 138 hosts. Our main aim was to demonstrate the rich existence of IP overlay paths that can be leveraged to significantly enhance Internet routing performance. We try to make the case for using layer-3 forwarding minimum delay overlay paths by demonstrating superior performance in this approach compared to existing overlay designs which work mostly at the TCP and application layers. In particular, the study was conducted to benefit applications that are sensitive to end-to-end delay and throughput. This paper presents a specific analysis of end-to-end delay in order to enhance TCP performance. The current work aims at increasing throughput and reducing file transfer time via overlay while maintaining simplicity and preserving all TCP characteristics. The results of this study show that the use of the shortest delay paths between physically disjoint node pairs can benefit TCP throughput and minimize file transfer time by orders of magnitude. The ultimate objective behind this study is to develop a reliable and scalable over-lay design for file transfers that require high transmission rates.

Keywords: Overlay routing · Delay characteristics · TCP performance

1 Introduction

Background: Traditional Internet connections are established via a set of underlying routing protocols at the IP substrate. On a consistent basis, overlay routing can provide better end-to-end performance such as faster download times, lesser stream re-buffering [8] or higher throughput for end-users. Overlay routing achieves this via exploring non-congested paths that are not necessarily discovered by best-effort Internet routing protocols. This flexibility helps to

Published by Springer Nature Switzerland AG 2019
M. Di Felice et al. (Eds.): WWIC 2019, LNCS 11618, pp. 170–181, 2019.
https://doi.org/10.1007/978-3-030-30523-9_14

direct traffic away from congested segments of the network and subsequently improves performance. Overlay paths can be chosen based on different parameters including lower delay, drop rate, or higher bandwidth along the path. The main goal of this work is to improve layer-3 TCP performance using minimum delay overlay routing.

The following key questions have being investigated in our analysis. For a given physical network topology, are there overlay paths that can provide lower delay compared to the best-effort paths offered by the current Internet protocols? How stable such paths are? What is the additional resource burden for such overlay paths that are generally longer than the best-effort paths? What improvement can be achieved by choosing such overlay paths? By answering these questions, our approach has achieved significant TCP throughput gain and file transfer delay reduction when comparing some underlay to overlay paths.

Methodology: Here, we describe the used platform, performed experiments and our designed procedure for conducting measurements. Platform: Networking testbeds such as Planetlab and Emulab are distinct. Planetlab, for instance, connects global clusters, from which a user can subscribe to a set of nodes in a slice. Our measurements are conducted using 140 nodes distributed according across the globe: North America 63.57%, South America 4.29%, Australia 3%, Asia 17.86% and Europe 12.86%. Experiments: Traceroute and Ping were used to conduct $18,906$ and $19,460$ end-to-end measurements respectively in a network of $n = 140$ nodes. Traceroute performed short-spaced measurements, i.e., $[5 \rightarrow 10]$ min interval. Ping sends bulk of packets of four distinct sizes, i.e., $0.05, 0.1, 0.25$ and 0.5 megabytes. These loads were scheduled in same order 4 times, i.e., a result of 16 experiments. Having a diverse measurements interval as suggested in [1], provides more confidence in capturing possible routing changes. Probing daemon: Throughout measurement period, all experiments followed exact probing abstraction illustrated as follows: The probing complexity is $\mathrm{O}(m)$, where m is the average number of allowed probers in a subset of nodes, i.e., group. In our case, we allowed one probing per group g_i where $i \in [1 \rightarrow m]$. Generally, maximum probing time t_i among all groups is an experiment running time. The probing time is considered at our server-side to represent the difference between a request time and response time from a daemon. Each daemon iterates into two loops: The first, is to probe all $n-1$ nodes, and the second loop is for re-probing unresponsive nodes once again. The actual group time is: $t_i = \sum_{k=1}^{|g_i|} \lambda_i(k) + \beta_i(k)$ as $\lambda_i(k)$ and $\beta_i(k)$ are probing loops times. These times can be determined as: $\lambda_i(k) = \sum_{j=1}^{n-1} \bar{\tau}$ and similarly $\beta_i(k) = \eta \sum_{j=1}^{\theta_i(k)} \bar{\tau}$, where $\theta_i(k)$ is number of nodes failed to be probed in first loop, $\bar{\tau}$ is average probing time in network and $\eta = 1$ is an average count of re-probing in our case. Our probing scheme used a server-based synchronization for each daemon, and afterward each daemon sends back its conducted measurements to our server. To minimize the effect of imperfect measurements occur when a large number of daemons probe a particular node simultaneously, we forced each daemon to probe at a time few randomly-selected destinations until covering all nodes. The possible consequence without such a procedure would be that the observed

delays to any destination can be influenced by our probing packets. Despite such randomness, we defined for successful measurement, a probability of success as described below. Probability of success: To reduce the influence of probing packets on the actual routing, such probability concerns for $n - m$ nodes, there is no node to be targeted simultaneously by all m probers. Determining such a probability as a function of m is identical to many existing problems such as birthday-paradox and withdraw with replacement. Solutions of different variations are in [4,5]. Her, we just compute the probability of success as a function of the number of groups used in probing. The probability of success, i.e., no collision in probing as in [4] is:

$$\Pr(\text{success}) = \prod_{i=1}^{m-1} \left(1 - \frac{i}{n-m}\right)$$

Recall, n and m are the numbers of nodes and groups respectively. Practically, m represents the number of active probers that can probe the network within a particular time. Therefore, m must be chosen carefully to satisfy desired success probability. Due to tedious computation when solving for an exact solution for m that satisfies a given demand for success, we can approximate the probability of success if $m \ll n$ by:

$$\Pr(\text{success}) = \exp\left\{ - \frac{m^2}{2(n-m)} \right\}$$

Solving for success demand equals 70% leads to $m = \frac{1}{2}\sqrt{(2.8534n + 0.50887)} - 0.35667$. For our network $n = 140$, we find $m \approx 10$. Clearly, achieving 100% of success reduces m to one as expected, i.e., the probability of success vanishes as the number of concurrent probers increases.

Contributions: The main contributions of this paper are as follows: First, we experimentally demonstrate that in many situations, there exists a rich topology of overlay paths that can provide lower round-trip-time (RTT) comparing to the best-effort underlay paths. Second, we characterize these overlay paths in terms of their benefit, temporal stability and the link consumption, i.e., time-to-live (TTL). Finally, we demonstrate that such lower-delay overlay paths can eventually be leveraged to improve application layer performance. We demonstrate the latter using FTP as a target application while the transfer time and throughput were considered as performance metrics at the application layer. Packets in the segmented TCP approach suffer from additional delay at the application stack of every overlay node, and that requires a robust store-forward buffering design [13] in order to minimize the re-transmission times. Furthermore, buffers are required to be coupled with back-pressure schemes to avoid flooding the slowest hop at higher rates. On the other hand, this paper proposed a slightly different approach to eliminate the previous and cost. The implemented IP-layer forwarding preserves all the current properties of the end-to-end TCP protocols except redirecting traffic through other paths that experience the minimum delay among all possible.

2 Related Research

This work can be depicted as an experimental prototype to [3] for content delivery applications. The dominant Internet traffic is served over the connection-oriented transport protocol TCP [6]. Our approach, however, is not dedicated to TCP applications, but we analyzed its performance using TCP as a possible worst-case scenario for the following reasons. TCP's feedback control always starts with a small congestion window that impacts the number of packets in flight. Further, the three-way handshaking signals add new additional round trip times to path latency [6]. Thus, overlay paths must remain constant over long periods of time and must change only when the QoS suffers degradation. In our study, the seven examined overlay paths were reused over a period of 5-hour by a rate of 50 times per hour.

Based on the argument mentioned above, our approach is applicable for live and high-quality video streaming such as Netflix, Voice-over-IP like Skype. The VoIP design in [2] can be viewed as an application of our approach. The authors in [2] provide no details about their overlay structure that improves call quality by 45% to be close from an oracle-based solution, i.e., all metrics for all paths are known. The whitepaper [5] on Akamai's overlay and the study in [6] are further instances of the use of overlay schemes to reduce the content retrieve-time in Data Centers Networks (DCNs) which can be a possible use-case for the overlay design outlined in our study. Most of the literature on cloud services using overlay as demonstrated in [5–8] and [9] do not specify their internal overlay design which we have done in the current study.

The study in [3] proposes a combined bandwidth and delay aware routing scheme for enhancing internet QoS. Their emulated SDN controller takes bandwidth and latency measurements from a separate monitoring entity and pre-calculates appropriate routes for each pair of nodes in advance. Our study, in contrast, performs an experimental overlay topology that can serve high-rate content delivery applications using the single metric of delay calculated using active probing measurements. Interestingly, our design also has achieved higher throughput and delay reductions over the set of examined overlay routes and by a large magnitude. On average, in the semi-heterogeneous topology discussed in [3], the results show that there was 22% of bandwidth increase and [4 → 15] RTT reduction.

3 Results and Analysis

This section summarizes our results and analysis as follows: Subsect. 3.1 shows the real time achieved TCP performance. This gain has been achieved after implementing the proposed overlay on seven long-distance overlay paths that forward 5-megabytes content between their end-nodes. In Subsect. 3.2, we provide an abstract characterization for the existence of possible minimum delay overlay paths between all available source-destination pairs, and further evaluate the stability of this characterization across the network. Subsection 3.3 discusses the

variation of the delay reduction between sources. For overlay networks that compete for small end-to-end delay, there is always a trade-off between the desired delay reduction and the burden of utilizing links extensively. Subsection 3.4 explores the total link consumption via both the underlying substrate and the obtained overlay paths. In Subsect. 3.5, however, we show the trade-off between the link consumption and the end-to-end delay. Subsection 3.6 illustrates the distribution of the RTT reduction at the node granularity.

3.1 Overlay and TCP Performance

The real-time implementation of the proposed single session and minimum end-to-end delay overlay routing was an essential stage in providing some insight about its importance and usability. Following the previous analysis, seven overlay pairs have been selected and configured to transfer the 5-megabyte file in between. We argue that although this number is small due to restrictions imposed at the IP layer by Planetlab and the no-flexibility using [1], our achieved performance via these paths can be considered as a generalization when using the same design. The comparison in Fig. 1 shows variation in the TCP throughput using the best effort IP routing and the proposed overlay design for the random set of overlay paths with different delay benefits. Regarding the TCP throughput, there is a considerable improvement for p_1, an increase of a magnitude for p_2, p_3, p_4 and p_6 while maintaining slightly the same underlay throughput for p_5 and p_7. The reason behind maintaining the same underlay throughput for p_5 and p_7 is that the overlays of these two paths originally provide small delay reductions, and so was the throughput improvement. The main reasons behind this leading performance of some paths are as follows: First, since the achieved throughput is inversely proportional to RTT, these minimum delay paths experienced less delay so that for any given TCP congestion window the throughput is maximized. From Fig. 1, the time required to transfer the 5-megabyte file as a second measure for TCP performance shows similar behavior to the corresponding

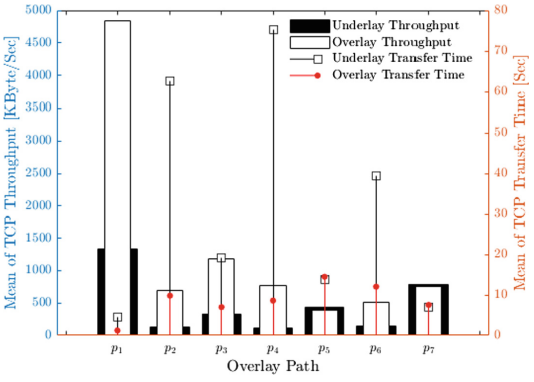

Fig. 1. TCP performance

throughput. In general, the transfer time has been reduced significantly except for p_5 and p_7 that provide minor delay reductions. This simple design overcomes the multi-segment and the multi-path forwarding by not requiring extensive packet buffering and reordering at the receiver end. Second, our design also has no packet encapsulation at the application layer.

3.2 Existence and Stability of Better Overlay Paths

Ideally, overlay paths are expected to be clustered around smaller RTTs when compared with the underlay RTTs that can be seen distributed across the entire delay range. Therefore, the overlay delay distribution curve should be clearly shifted to the left of the underlay distribution along the entire range, and its peaks raised up as much as possible at the smallest delay values of the range. In contrast, the two curves are expected to be aligned on each other as the RTT increases. Figure 2 confirms this description by showing the behavior of this distribution over the 24-hour measurement period. By considering the most commonly observed round-trip-times and excluding all delays of the one-hop overlay paths, the plot shows that there is only 2.8% of underlay paths out of 18,906 with 40 ms end-to-end delay. The overlay routing, however, reduces the end-to-end delay to about 25 ms for 2.6% of the paths and raises the number of paths that experience 40 ms to 4% instead. Overlay paths competing for shorter delays exist up to the 500 ms mark after which overlay routing starts yielding to underlay as their pdfs are perfectly aligned.

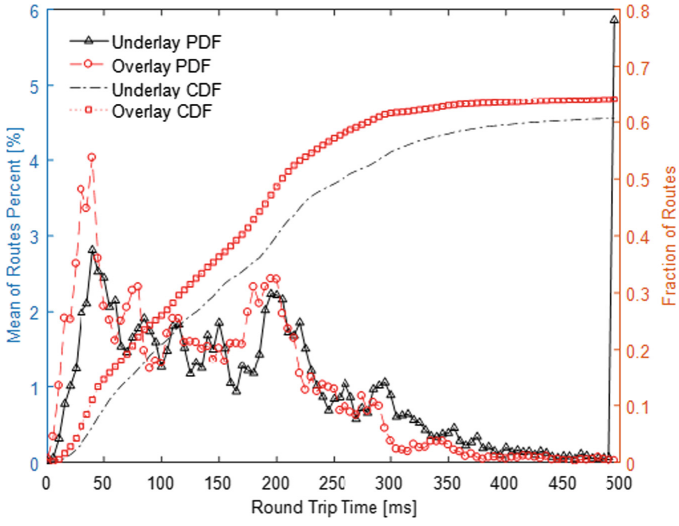

Fig. 2. Underlay and overlay RTT distributions

The range between [10 → 500] ms is commonly observed and solved by the proposed overlay, as the percent of overlay paths collapses outside this range. In this analysis, we only considered overlay paths whose end-to-end delays are less than 500 ms, and that is why the CDF curves do not approach one. The CDFs also show that for 250 ms delay, 49% and 40% are an overlay and underlay paths, respectively. We found that two hops overlay paths, however, can provide distinct delay reduction for 35% of the 18, 906 end-to-end connections, i.e., more than the current IP routing can perform.

3.3 RTT Variations in Benefit

Overlay controllers should be able to enforce routing decisions based on benefit change. We decided to partition the entire overlay design into four routing sets. The set S_1 represents overlay paths that can reduce the physical end-to-end delay by more than 100 ms. S_2 accommodates paths for delay reduction within the range [10 → 100] ms. Failures indicate either one of two different scenarios. The first scenario is that a physical path has no better overlay alternative, while the second one is that for a physical failure that cannot be resolved even by the overlay routing. These two scenarios are represented by the sets S_3 and S_4 respectively. Figure 3 above shows the variation of this reduction between different nodes. Overlay benefit is indicated by the degree, to which the overlay distribution is shifted to the left of the underlay. It is evident that higher overlay benefit is achieved for nodes where the overlay or underlay distributions are toward the higher RTT values. This range of RTT reduction is further analyzed in Subsect. 3.5. We argue that our large-scale measurements represent this range reasonably enough for wide area networks because of the heterogeneous

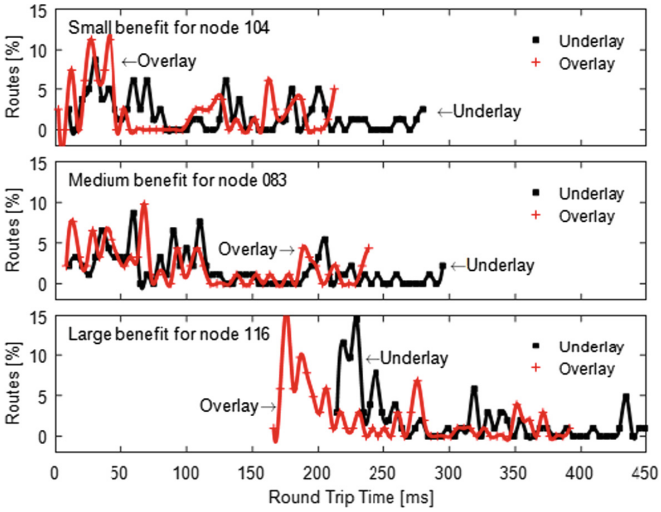

Fig. 3. Variations of RTT benefit

resources in Planetlab, and the underlying routing will start to be minimum beyond this range. By looking at the relationship between S_2 and S_3, we can infer the following. As $|i_{s_2}| - |i_{s_3}|$ increases, the overlay benefit increases, and therefore, compared to what we described in Subsect. 3.2, the RTT distribution will be further shifted left for the nodes with large overlay benefit. Nodes with small benefit from overlay will have almost two identical RTT curves. The analyzed traceroute experiments have shown identical behavior between the benefit curves, and Fig. 3 shows only random nodes from the first experiment.

3.4 Burden of Link Consumption

There is always a trade-off between minimizing the end-to-end delay and the number of links, i.e., network resources, used to achieve this objective. To understand such behavior, Fig. 4 shows the averaged link consumption for both the underlay and the overlay topologies. The underlay link consumption is be approximated by $\mu = 16.5$ and $\sigma = 4.7$ while overlay corresponds to $\mu = 33.7$ and $\sigma = 14$, where μ and σ are the mean and standard deviation of link distributions respectively. Surprisingly, from this result, there still exist overlay pairs that communicate under minimized end-to-end delay, while maintaining the common range of the underlay link consumption that is below 30 links. The jump at 30 links for the underlying consumption represents both underlay paths of 30 links and all failed paths.

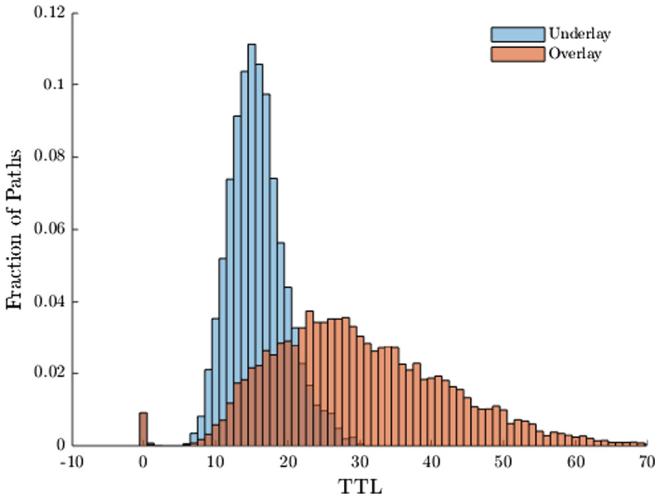

Fig. 4. Underlay and overlay TTL distributions

There is also strong stability in the underlay distribution over 24 h compared to the over-lay distribution since the link consumption over 24 h does not deviate much from their means that are represented vertically on the two curves. This

indicates that any delay aware overlay routing can more greedy but sensitive to delay changes than the traditional IP protocols, and that is because of its global view about the network delay.

3.5 Link Consumption and RTT Benefit

Section 3.4 defined the amount of overlap between the link consumptions of both the underlay and overlay paths. In here, however, we relate the previous link characteristic to the end-to-end delay reduction. For a given end-to-end path p, we refer by $\delta_{rtt}(p)$ to the delay difference in RTT between the underlay and the overlay paths $r : i \rightarrow j$. That is $\delta_{rtt}(p) = \underline{d}(p) - \overline{d}(p)$, where $\underline{d}(p))$ is the underlay and $\overline{d}(p)$ is the overlay RTT. Based on this definition, $\delta_{rtt} \geq 0$ with equality if the overlay path is just one hop, i.e., the same as the underlying path. Therefore, when $\delta_{rtt}(p) > 0$, that means the overlay path is at least two hops in length: $\delta_{hop}(p) = \overline{h}(p) - \underline{h}(p) \geq 2$ hops. The $\overline{h}(p)$ represents the length in hops of the overlay path while $\underline{h}(p)$ represents the underlay path and is always equal to one. Similarly, at the link granularity, the introduced $\delta_{link}(p)$ on the other hand, can be either positive or negative. When $\delta_{link}(p) = 0$ it does not necessarily imply that for the given connection its overlay and underlay paths are identical, simply because an overlay path of at least two hops can use in total the same number of links used by the corresponding underlying one hop path. In this analysis, we only considered the overlay paths that are not in any of following underlay-overlay combinations: failure-recovery, failure-failure or recovery-failure, where the last one is related to the underlying paths that have minimum RTT. That is why the cumulative CDF in Fig. 5 above does not approach one. This figure shows

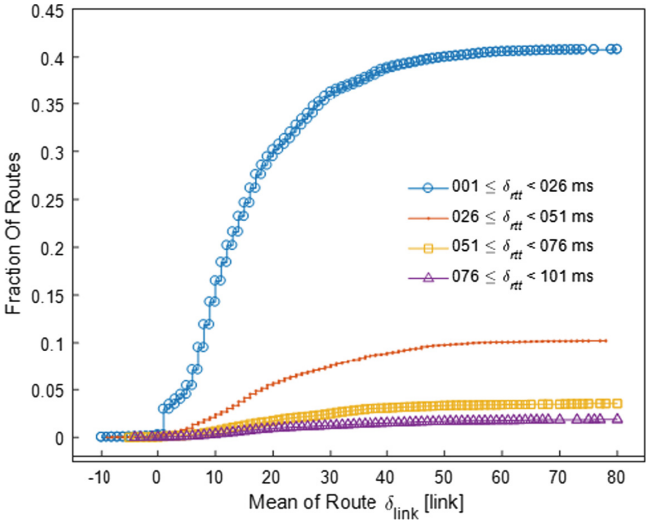

Fig. 5. Delay benefit vs. link consumption

a constrained $\delta_{link}(p)$ by a desired delay reduction, i.e., $\delta_{rtt}(p)$. From Fig. 5, we can conclude that the $[1 \rightarrow 25]$ ms is the range, on which we can maximize the performance of the overlay routing at the smallest link consumption.

Ideally, as the configuration of the overlay hops consumes less number of hops, the expected link consumption should not be high. Our analysis also has looked at the relationship between the three metrics: delay reduction, link and hop consumption across the entire network. The result in Fig. 6 shows that throughout the entire measurement period on average, the overlay topology uses two hops of an average link consumption within $[10 \rightarrow 15]$ links in order to achieve an average delay reduction of 30 ms.

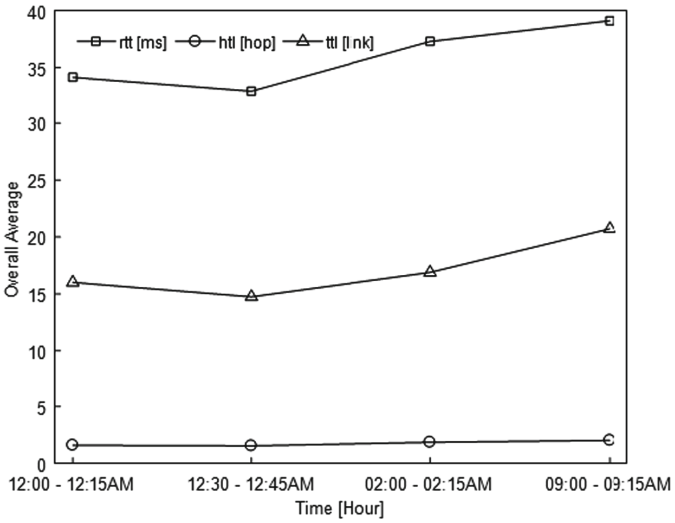

Fig. 6. Link and hop consumption vs. Delay

3.6 Node RTT Benefit

The previous Sect. 3.5 discussed δ_{rtt} only at the path granularity. The expectation at the node granularity is that the aggregated $\delta_{rtt}(n_i)$ for nodes that act as sources will show a peak at the most shared reduction between them. Since we have only considered the overlay set S_2, this lump will reside somewhere in the range $[1 \rightarrow 100]$. In a network of N nodes, and for a node i in particular, we define the average delay reduction per node as:

$$\delta_{rtt}(n_i) = \frac{1}{N-1} \sum_{j \neq i} \delta rtt(p : n_i \rightarrow n_j)$$

Fig. 7 depicts the complete picture of $\delta_{rtt}(n_i)$ of the first traceroute experiment peaked at 25 ms. This means that the averaged delay reduction range, at which all sources are functioning, is $[22 \rightarrow 27]$. This, of course, does not mean that the

overall benefit from implementing an overlay will be only within that range, but on average, this is the trend at the source granularity.

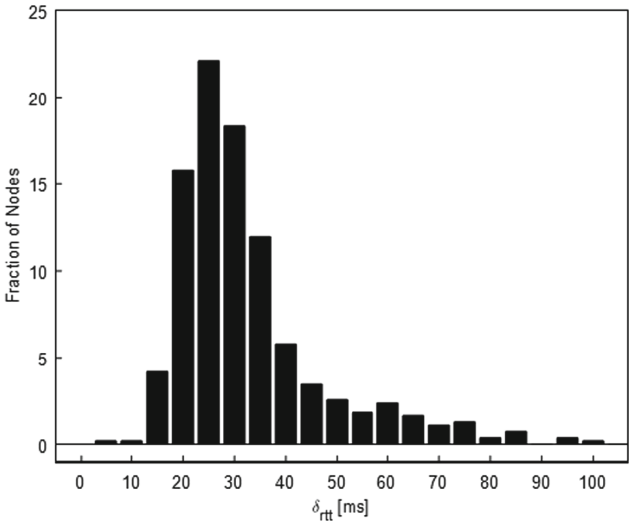

Fig. 7. Distribution of node delay benefit

4 Conclusion

This study analyzed the behavior of single session overlay routes that minimize the end-to-end delay for all possible TCP pairs in the network. Our analysis has been derived for the most observed and common range to reduce the delay of $[1 \rightarrow 100]$ ms. The study has confirmed the existence of a considerable number of overlay routes that can minimize the underlay round-trip times by the mentioned reduction range for the majority of nodes. The advantage of this overlay design varies from node to node, based on their topological positions. The study summarizes the trade-off between minimizing the end-to-end delay and the consumption of the network resources by showing the lack of delay reduction using fewer links than the underlying paradigm. In the future, we will investigate the difference in reduction between this minimum delay overlay routing and another overlay design that will optimize the overall network delay, while being less greedy in consuming network resources.

References

1. Freire, C., Quereilhac, A., Turletti, T., Dabbous, W.: Automated Deploy-ment and Customization of Routing Overlays on PlanetLab, TRIDENTCOM (2012)
2. Jiang, J., et al.: VIA: improving internet telephony call quality using predictive relay selection. In: Proceedings of ACM SIGCOMM (2016)

3. U-chupala, P., et al.: Application oriented bandwidth and latency aware routing with OpenFlow network. In: Proceedings of The 6th IEEE International Conference on Cloud Computing Technology and Science (CloudCom) (2014)
4. Lee, S., Banerjee, S., Sharma, P., Yalagandula, P., Basu, S.: Bandwidth-aware routing in overlay networks. In: Proceedings of IEEE-INFOCOM (2008)
5. Habib, S., Bokhari, F.S., Khan, S.U.: Routing techniques in data center networks. In: Khan, S.U., Zomaya, A.Y. (eds.) Handbook on Data Centers, pp. 507–532. Springer, New York (2015). https://doi.org/10.1007/978-1-4939-2092-1_16
6. Sitaraman, R., Kasbekar, M., Lichtenstein, W., Jain, M.: Overlay Networks: An Akamai Perspective. Advanced Content Delivery, Streaming and Cloud Services. John Wiley & Sons, New Jersey (2014)
7. Cheng, F., Sitaraman, R., Torres, M.: End-user mapping: next generation request routing for content delivery. In: Proceedings of ACM SIGCOMM (2015)
8. Maggs, B., Sitaraman, R.: Algorithmic nuggets in content delivery. In Proceedings of ACM SIGCOMM Computer Communication (2015)
9. Liu, Y., Gu, Y., Zhang, H., Gong, W., Towsley, D.: Application level relay for high-bandwidth data transport, GridNets (2004)