Towards Stable and Hybrid UDP-TCP Relay Routing for Streaming and VoIP Services

Salim Mohamed
Electrical and Computer Engineering
Michigan State University
East Lansing, USA
Email: mohame26@msu.edu

Osama Mohammed
Service Delivery and Management
Innovaway
Napoli, Italy
Email: osama.mohammed@it.ibm.com

***Abstract*—Relay or overlay routing for IP networks has been well-documented in past years. However, the implementation cost of relay solutions has not yet been conclusively identified. Dynamic-relay routing relies on periodic probing for enhanced performance while static-relay routing uses less and non-periodic probes to measure latency and packet loss. For both types, there exists considerable research focused on understanding routing dynamics. However, the literature has insufficient exploration of relay attributes, such as stability and mechanisms for reducing the relay probing burden. This paper, in particular, examines relay statistical boundaries and characteristics, such as the number of hops in a minimum delay and relay path or Hop-To-Live (HTL) count inherited from the self-similar model of Internet data. The HTL is introduced in a novel analysis to assist in predicting minimum and stable relay paths while minimizing probing overhead. For doing so, our work is based on analyzing a wide-set on $19,460$ Ping and $14,762$ IPerf paths, respectively, of a network of 140 Planetlab nodes. Further, we briefly evaluated the performance of a new hybrid User Datagram Protocol and Transmission Control Protocol (UDP-TCP) relay streaming over an inexpensive relay selection mechanism managed by a stable HTL modeling. Here, we highlight a preliminary performance of applying a layer-3 and hybrid UDP-TCP streaming a replacement for the current TCP-based stream services, such as YouTube and Voice over IP (VoIP). The main results emphasize the unnecessary repetitive probing burden over the period of 24 hours instead of a careful set of measurements for capturing and predicting relay changes. This work validates this claim by presenting that our implemented HTL-based path estimation predicts stable relay paths for the hybrid UDP-TCP streaming to overcome the high drop-rates caused by the individual TCP or UDP streaming services.**

Keywords-UDP streaming; relay characteristics; Internet measurements.

I. Introduction

A. Overview

Existing relay solutions are completely dynamic while relying on continuous probing when measuring routing changes. In contrast, static relay, routing metrics, such as latency and packet loss are not continuously measured. The ultimate goal of the static relay is reliability so that nodes remain connected. Dynamic relay, however, is used to improve the Quality of Service (QoS) periodically. The study in [11] focused on understanding Time-To-Live (TTL) changes in the IP substrate (underlay) routing. However, here, we introduce and examine the influence of a new relay (overlay) routing metric that defines the number of hops in a minimum delay relay path. For the remainder of this paper, we refer to this metric as called the Hop-To-Live (HTL). Generally, our work is a composition of two parts. The first part is an analysis of the stability characteristics of relay (overlay) routing. Preliminary, in the second part, we performed a performance evaluation of a new streaming scheme called hybrid UDP-TCP streaming. The idea here is new and simply relies on combining for each stream two distinct transport sessions. The first is a conventional TCP session for handling errors and out-of-sequence packets in the stream, and this session is only invoked whenever such an error occurs. The second session is a normal UDP session that carries the major part of the stream. The performance of our hybrid protocol is simply examined by the UDP drop rate, at which the TCP back-end protocol is involved. This evaluation was performed using relay paths determined via our proposed estimation-based path selection model. This model operates based on the HTL characteristics described in the first part. The input of this model is as a set of $19,460$ relay paths connect a 140 Planetlab nodes, and outputs a smaller set of stable relay paths for each end-to-end pair of nodes. The input relay paths represent the relay features used to derive our path estimation model, and all such paths capture only routing changes as we vary Round-Trip-Time (RTT), traffic sizes, and rates.

The main question addressed here is to determine how stable is a measurement-less model of relay paths over $24-$hours so that a single instance of underlay measurement could be reused for estimating new stable paths. The next inquiry is to look into the scale of achieved benefit when implementing our estimation model into real-time scenarios, such as a path serving for the hybrid UDP-TCP protocol.

The motivation behind using hybrid UDP-TCP instead of TCP is slightly similar to the Quick UDP Internet Connections (QUIC) protocol in [24]. QUIC is a multi-purpose application-layer protocol initially designed by Google Langley et al. [25]. Later, in 2013 QUIC was announced publicly for exper-

ISBN: 978-1-61208-786-3

imentation Langley et al. [25], and redesigned by the Internet Engineering Task Force (IETF) while still being an Internet-Draft. QUIC adds the missing reliability feature to UDP at the application-layer as opposed to being implemented at the transport-layer. Meaning, different services now can design their reliability, error-correction, privacy, and security demands at the user-space. In contrast, our hybrid, UDP-TCP does not perform maintain each of these tasks at the transport-layer while simply having UDP to forward the major stream portion, and when an error occurs, the receiving-end notifies its peer via a back-end TCP connection. This paper is neither providing a comparison between the two protocols nor a discussion on the protocol details.

Plantlab is a global and shared research network. However, many routing privileges are disabled to avoid unexpected routing between slices. Researchers at the Planetlab have designed a replacement for the well-known Unix sudo command. Vsys is a method for handling access restrictions to privileged operations. This study used an automated version of Vsys to setup all examined relay paths between end-nodes. However, defining privileges at an arbitrary granularity by filtering data between the host and guest domains is one of the tedious challenges of our work.

Relay routing introduces new implementation-concerns, such as probing overhead (cost) and its processing latency, stability, availability, and sensitivity to underlay routing changes. The lack of privilege at the IP layer in older overlay schemes required relay layers to be implemented at the user-space (application layer) instead of the kernel-space (transport-layer). One example of such schemes is the application-multicast protocol. Recently approaches like Software Defined Networks (SDNs) offer that privilege, but the probing cost remains high. The long-term boundary of our research is to develop a robust, resilient, and inexpensive layer-3 relay protocol to handle the unprecedented demand for streaming services in many circumstances, such as the global pandemics.

B. Dataset

Valuable bandwidth datasets, such as the one used in Jiang et al. [1] are private ones. Moreover, public dataset like CADIA [13] and RIP-NCC [14] do not offer large-scale and demand-based bandwidth traces. Therefore, in order to achieve meaningful bandwidth estimations on a large-scale, we used a global network of 140 nodes. These nodes are distributed across the globe as follows: North America 63.57%, South America 4.29%, Australia 3%, Asia 17.86% and Europe 12.86%. We performed a set of $311,360$ delay traces using Ping and $177,144$ bandwidth measurements via IPerf.

C. Experiments

Ping and IPerf were used to conduct measurements over $19,460$ and $14,762$ end-to-end paths, respectively in a network of $n = 140$ nodes. Ping sends its bulk of packets in four distinct sizes: 0.05, 0.1, 0.25 and 0.5 MBytes. Ping packets were also scheduled in the same order 4 times in 16 experiments. IPerf datagrams were sent at 12 distinct demand-rates as in Table II. Our diverse measurements were used to examine the HTL characteristics, and design a stable HTL-based path estimation. Having a diverse measurement interval as suggested in [11], provides more confidence in capturing possible routing changes.

D. Probing Daemon

The conducted measurements follow the exact probing abstract illustrated in [2]. The nodes were divided into a number of groups. We performed a single measurement in each group g_i where $i \in [1 \rightarrow m]$. Each prober utilizes two loops: The first, is to probe all $n - 1$ nodes, and the second one is to probe unresponsive nodes 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: $\lambda_i(k) = \sum_{j=1}^{n-1} \bar{\epsilon}$ and similarly $\beta_i(k) = \eta \sum_{j=1}^{\theta_i(k)} \bar{\epsilon}$, where $\theta_i(k)$ is the number of unresponsive in the first loop. $\bar{\epsilon}$ is the average probing time in the network. $\eta = 1$ was the average count of re-probing in our case. Our probing scheme used a server-based synchronization to minimize the effect of probing conflict occurs when two daemons or more probe a particular node simultaneously. To reduce such imperfect measurements, we forced daemons to randomly probe all nodes.

Furthermore, we defined the probability of success for reducing the influence of probing conflict on the measurement accuracy. Such probability concerns $n - m$ nodes when no node was targeted simultaneously by more than one prober of m active ones. The probability of success with no conflict in probing was approximated as in [3] [4] by:

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

Here, 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 of success, we can approximate the probability of success in (1) if $m \ll n$ by:

$$\text{Pr(success)} \approx \text{exp}\Big\{ - \frac{m^2}{2(n - m)} \Big\} \tag{2}$$

Solving for success demand equals 70% leads to $m \approx 10$. For our network, $n = 140$, we found $m \approx 10$. Clearly, achieving 100% of success reduces m to one as expected.

E. Contribution

The major contribution of our work is in identifying short and long-term analyses of the minimum RTT relay paths, such as long-term stability as discussed in Section III and the HTL alternation sequences detailed in Section IV and Section V. Further probabilistic relay attributes, such as prevalence are studied in Section VI. For analyzing HTL characteristics, we performed $311,360$ RTT measurements for all paths in a network of 140 nodes. Through extensive analysis, we found that the HTL prevalence shares a similar behavior to the TTL prevalence studied in [11]. Therefore, we conclude that both HTL and RTT are sufficient routing metrics for predicting relay changes and thus, reducing the relay cost. This paper proposes two different schemes for using hybrid UDP-TCP as an alternative for TCP-based streaming and VoIP services and shows the difference in performance between the two schemes.

The remainder of this paper is organized as follows: Section II summarizes the importance of recent relay schemes. Dominant relay redundancy and its type are analyzed in Section VII. The proposed HTL detection scheme is briefed Section VIII. Section IX describes our proposed UDP streaming performance. Section X concludes our paper.

II. Related Research

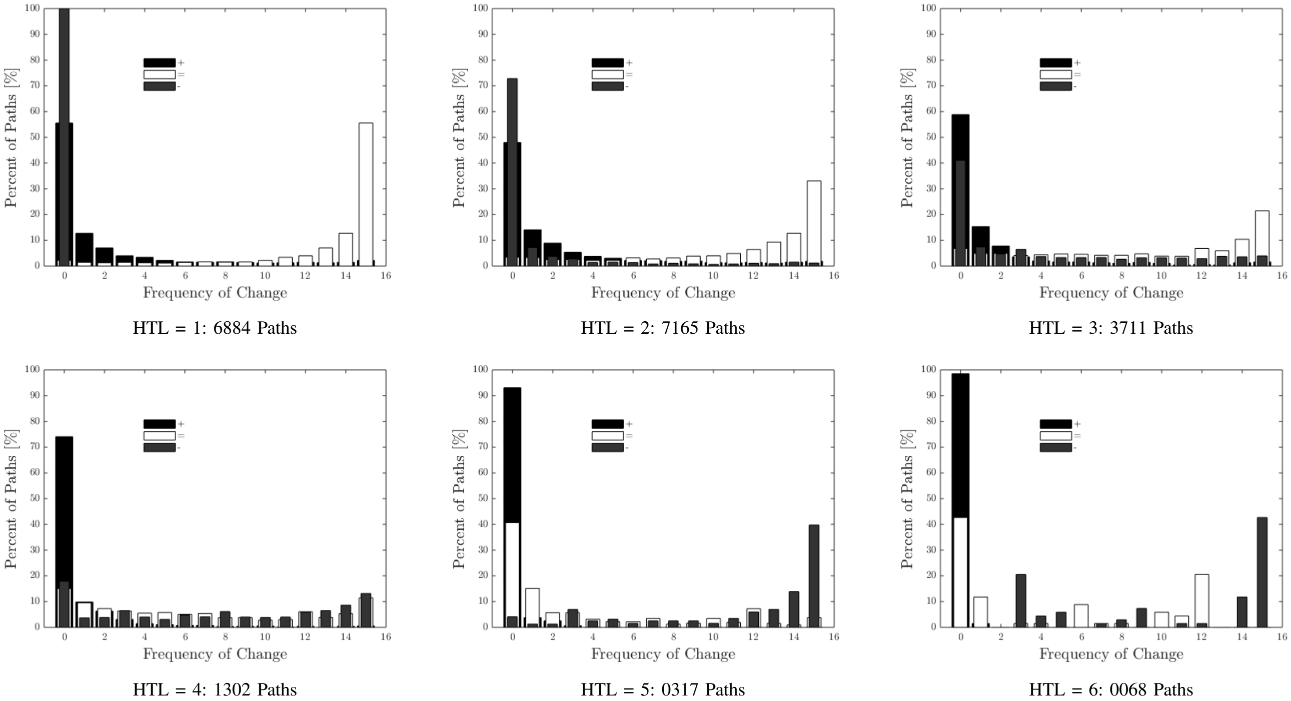
The study in Jiang et al. [1] refers to an estimation-based relay scheme for Skype users. The skew in data density mentioned in Jiang et al. [1] is due to the lack of measurements (samples) for end-pairs, and was replaced by network tomography-based delay estimation. This approach can not be generalized and replace the probing overhead required for achieving a clear view of the network's performance. For a tomography-based estimation, our study is a counterexample, in which we found both the underlay and the relay paths are asymmetric in general. We postponed our symmetry analysis due to page limitations. Therefore, the gain of relay performance using a tomography-based scheme might not achieve the desired QoS for end-users due to the lack of delay symmetry. Researchers in [2] have used same the measurements to construct a Layer-3 forwarding scheme for data transfer at a small-scale. The considered relay paths were selected according to their HTL stability. The difficulty in performing direct probing in a large network studies [3] like [4] to show that it is possible to infer network conditions based on content distribution networks [5] with relaying. [6] is an example of such a scheme. Our study is a specific implementation of [7] and [11], in which the authors focus on examining the basic problem of QoS routing for multimedia applications by finding a path that satisfies multiple constraints. In our study we consider a new direction using UDP as relay protocol instead of TCP for YouTube and VoIP applications like Skype and Viber.

The work in [8] illustrates that IPv4 paths are more stable than IPv6 paths. This motivated us to further examine the stability of the IPv4 relay. In [9], authors highlight the importance of new schemes for predicting underlay RTTs, and that supports our claim for the need of new estimation designs for relay routing. The study in [10] uses relay path stability and symmetry characteristics to overcome the inefficiency of the relay when not considering certain underlay links. The used stability assists in finding more efficient relay paths. In contrast, we used the HTL count stability instead of the entire relay path structure in our study. Paxson in [11] examined end-to-end behaviors due to the different directions of underlay paths, which often exhibit asymmetries. The author characterized the prevalence and persistence of underlay paths. In contrast, we performed a similar analysis for relay paths. In [12], the authors examined path diversity on relay networks. They used 50 Planetlab nodes to conduct Traceroute measurements. They concluded that relay performance gains are limited by the natural diversity of redundant paths between two end-hosts in terms of underlay links, routing infrastructure, administrative control, and location. This motivated our characterization of HTL by analyzing prevalence and redundancy. The study in [17] shows that the mean of per-hop delay between parent and child nodes in a relay tree decreases as the level of the host in the relay tree increases, and this is due to the fact that current underlay routing is not optimized in-terms of delay.

III. HTL Stability

This section provides a complete identification of the HTL changes over the measurement period. The first Ping experiment is considered as a baseline since it contains all possible HTL variations for monitoring the HTL behavior. Here, we present eight HTL sets. Each set represents an average stability measure of a subset of $19,460$ relay paths for a particular HTL, meaning each set contains paths whose HTL equals the set's index. Each path of these sets was randomly traced every $[10 \rightarrow 15]$ minutes. In Figure 1, the $x-$axis indicates the probability of change, by which the paths of a set change over time, for example, $p = 1/15$ indicates a single change over 15 measurements. Each set of relay paths are described by its probability of switch p and switch type, x in (p, x) where x may represent a decrease $-$ in HTL, same $=$ HTL, or increase $+$ in HTL. The subset of paths represented by $(0, +)$ with zero probability switches to longer HTL. The subset $(0, =)$ indicates 100% change either to longer or shorter HTL. Similarly, $(0, -)$ refers to an impossible change to shorter HTL. Before discussing the actual result of this section, let's describe an ideal scenario for a stable HTL of a set of relay paths. The combination $(0, +)$ and $(0, -)$ should be maximized at zero while the $(1, =)$ should peak at one. Therefore, any analysis of this nature should approximate this ideal model in order to conclude that a set of relay paths of a particular HTL is highly stable. The IP routing dynamics, however, forces HTL to deviate in its stability.

From Figure 1, as expected, we found that for all HTL-sets, relay paths follow an exponential decay when switching to higher HTL values. The exponential curve starts to collapse as the set's HTL increases. This means that as the set's HTL increases, relay paths tend not to increase their lengths throughout the observation period. For the sets of HTL counts 2 and 3 hops, we still notice an approximation for the ideal model with a considerable decrease for all combinations. However, for the sets of HTL equal to 3 and 4 hops, the probability of always being at the same HTL is very small, and that is why the $(1, =)$ bar decreases causing a trend of decreasing HTL until peaking at $(1, -)$. By doing an overall



HTL = 1: 6884 Paths

HTL = 2: 7165 Paths

HTL = 3: 3711 Paths

HTL = 4: 1302 Paths

HTL = 5: 0317 Paths

HTL = 6: 0068 Paths

Figure 1. Relay HTL Stability.

analysis of the entire behavior of all HTL sets while focusing on $(p = 0, x)$, we conclude with three important curves. The first curve refers to the $(0, +)$ that starts near 50% in the first set and raises up toward its maximum at the eighth set. This indicates that as HTL increases, paths tend not to change to longer hops. The second curve for $(0, =)$ starts at zero percent, and with positive slope also reaches its maximum in the eighth set. That means relay paths never stay at a fixed length as HTL increases. The third curve for $(0, -)$, however, with negative slope collapsed in the sixth set. That means as HTL $\in [2 \to 6]$ increases, paths tend to reduce their lengths. The HTL stability analysis is summarized as follows: Since paths with HTL ≤ 4 hops are dominant in logical routing, we found that they either prefer to remain at constant HTL or switch to shorter HTL counts. Their tendency to reduce the number of hops is uniform, in particular for paths of HTL ≤ 3 hops. Therefore, the focus should be on HTL $\in [2 \to 4]$ hops when designing stable relay routing. Beyond 4 hops, paths are less stable in maintaining constant HTL.

IV. HTL Frequency Sequence

HTL oscillations occur due to routing changes in IP dynamics. For an underlying path, we argue that with careful path measurements at random intervals that spread over a considerable amount of time, it is possible to observe all available relay paths. This is because External Border Gateway Protocol (EBGP) only exchanges routing advertisements between adjacent Autonomous Systems (ASes), and thus, non-neighboring ASes will have no bearing on EBGP.

Since the failure of an underlay path can last for 225 seconds [18], during such a time on average, we were able to conduct measurements for some outages, and deploy better relay paths that significantly scaled up path performance as in [2]. Using the semi-Markov chain for modeling underlay fluctuations, each state of the chain depends only on a random life-time drawn independently from a state distribution. Therefore, the steady-state probability of a particular state is equal to the average time spent in that state [18]. First, each sequence of hop measurements $\mathrm{M}_s(h)$ is defined as states of a particular relay path. Each state s_i is a representation of the path at a particular granularity. From $\mathrm{M}_s(h)$, we can construct a semi-Markov process. Having $\mathrm{M}_s(h) = 1, 1, 2, 1, 2, 3, 1, 2$ means that each state number represents HTL for its observed relay path. For a an interval between consecutive measurements of 10 minutes, we can perform the following: The possible transitions within $\mathrm{M}_s(h)$ are $1 \to 1, 1 \to 2, 2 \to 1, 2 \to 3, 3 \to 1$. For state 1, its life-time is $\frac{4\times10}{70} = 0.5714$, and its transition $1 \to 2$ occurs with probability 0.75. The probability of a transition $s_1 \to s_2$ is simply defined by: $p(s_1 \to s_2) = |s_1 \to s_2||s_1 \to s_i|^{-1}$. The symbol $|\cdot|$ refers to the number of times a transition occurs. Using the later construction procedure, the semi-Markov chain for the given $\mathrm{M}_s(h)$ will converge on an actual Markov chain as we increase the number of path observations. Using $\mathrm{M}_s(h)$, each underlay path can be mapped onto a transition process composed of possible states of better relay paths. By applying such a model, we can reduce probing frequency or cost by allowing changes within pre-estimated relay paths.

Furthermore, as we vary our probing rates while conducting a measurement sequence, we were interested in answering questions, such as what is the overall HTL miss-rate? How likely is it that a particular HTL will not be observed? To answer our first question, we introduced for each underlay path a second HTL sequence of the most probable relay HTLs so that whenever probing is required, only paths of these lengths will be investigated. This sequence is called frequency sequence $\mathrm{F}_s(h)$. For a given underlay path r, if $\mathrm{M}_s(h) = 3, 3, 5, 2, 4, 5$, then $\mathrm{F}_s(h) = 3, 5, 2, 4$. Finding $\mathrm{F}_s(h)$ for every path in our network shows that only 4.8% of $19,460$ paths with at least one alternative relay path suffer HTL miss(es). The interaction between underlay and relay substrates causes exactly 56.6% of the 4.8% relay paths to suffering from high fluctuations during our measurement period.

During analysis, there were $8,863$ paths whose $\mathrm{F}_s(h)$ sequences demonstrated multi-hop paths (relay was always better), and suffered no HTL miss. Furthermore, an additional $5,824$ paths whose underlay candidates still included in $\mathrm{F}_s(h)$ also suffered no HTL miss. This indicates about 75.4% of paths change their HTL within a stable $\mathrm{F}_s(h)$ during the measurement period. By excluding fixed underlay paths, we found only 949 paths suffered HTL miss(es). Such a number is quite small compared to the total of $19,640$ paths. Figure 2 details the fraction of paths per each HTL miss.

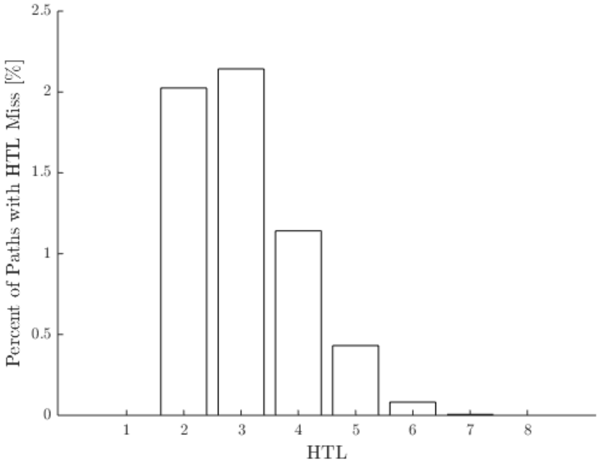


Figure 2. The Overall HTL Miss.

The probing Ping bulks were used in our measurements to create congestion and consequently to stimulate routing changes. However, despite a 2% packet loss on some paths, 2% of such paths of the total underlay paths remained constant as one-hop. The topological location of their ends might be a strong reason for such behaviour. Larger HTLs were less observed, and yet still not common in some relay paths.

V. HTL Transition Sequence

The discussion in Section IV neither investigates the nature of HTL switching, gradual or random, nor how often HTL miss(es) occur. Further, there is no strong evidence that gradual transition indicates path symmetry at the node granularity. Thus, considering time will help studying relay symmetry. The new sequence $\mathrm{T}_s(h)$ takes time into consideration for identifying HTL miss(es) that occur between consecutive measurements. Generally, $\mathrm{M}_s(h)$ is a sub-sequence of $\mathrm{T}_s(h)$, and therefore, $\mathrm{T}_s(h)$ can be generated by placing all the missing HTLs. For example, at the HTL granularity: $\mathrm{M}_s(h) = 1, 3, 1, 2, 4, 1$, $\mathrm{T}_s(h) = 1, \overline{2}, 3, \underline{2}, 1, 2, \overline{3}, 4, \underline{3}, \underline{2}, 1$. The upper bar indicates a miss when switching to higher HTL and vice versa. Note, HTL $= 2$ and 3 are not misses in $\mathrm{M}_s(h)$, but they are in $\mathrm{T}_s(h)$. Using $\mathrm{F}_s(h)$, we can catch missing HTL counts, but not determine neither a probable miss-time nor that common path fluctuations caused a transition miss.

From $\mathrm{T}_s(h)$, if a particular HTL is a *frequent* transition miss, such HTL count is not favorable. For a particular HTL, it can show as a miss in $\mathrm{T}_s(h)$ while not in $\mathrm{F}_s(h)$. For example, $\mathrm{M}_s(h) = 2, 1, 3, 3, 2$, the corresponding $F_s(h) = 2, 3, 1$ and $\mathrm{T}_s(h) = 2, 1, \underline{2}, 3, 3, 2$. Here we consider $\mathrm{T}_s(h)$ as miss-free since the miss chance of HTL $= 2$ is small, 0.2. However, for $\mathrm{M}_s(h) = 4, 3, 3, 3, 3$, $\mathrm{F}_s(h) = 3, 4$ and $\mathrm{T}_s(h) = 4, \underline{2}, 3, 3, 3, 3$, and despite the 0.2 small miss likelihood of HTL $= 2$, we consider such $\mathrm{T}_s(h)$ with a miss since HTL $= 2$ is not in $\mathrm{F}_s(h)$. For HTL = 2 hops, about 90% of our relay paths had no miss(es) during path switching. Figure 3 shows a Cumulative Distribution Function (CDF) breakdown of the transition misses for the most common relay HTLs. From such a result, we can conclude that nature of HTL switching in relay follows a gradual transition rather than a random one.

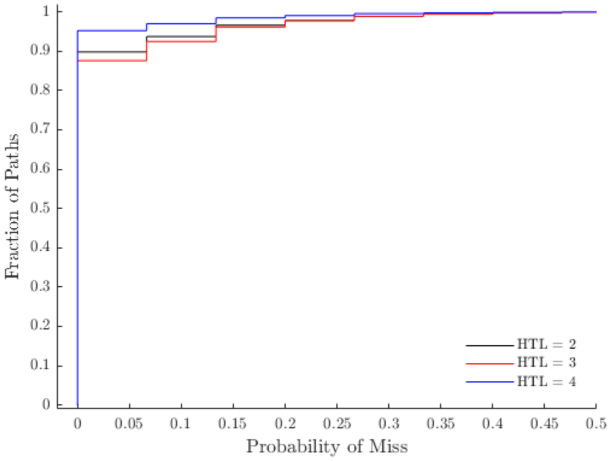


Figure 3. HTL Transition Miss Breakdown.

VI. Dominant Path Prevalence

Let us define for an underlay path r with a set of m underlay measurements: $\mathcal{U}(r) = r_1, r_2, r_2, \ldots r_m$, and another set of shorter delay relay paths: $\mathcal{O}(r) = r^1, r^2, r^1, \ldots, r^m$. Let's also introduce a dominant set $\mathcal{D}_i(r)$, where $i \in [1 \rightarrow x^*]$ as x^* is the possible number of distinct dominant sets we can observe out of m measurements for r. Each $\mathcal{D}_i(r)$ contains a subset of $\mathcal{O}(r)$ of relay paths that appear at a unique frequency of occurrence $\omega_i(r)$. For instance $\mathcal{D}_1(r)$ contains the prevalent relay paths observed at the highest frequency, $\omega_1(r)$, and $\mathcal{D}_2(r)$ encompasses paths with second highest frequency,

$\omega_2(r)$. Generally, the maximum number of dominant sets: $x^* = \arg\max_{x} \sum_{w=1}^{x} w \leq m$ that represent r will occur if each path of m has a unique frequency. Further, we define a source prevalence as a stronger stability measure in addition to our overall HTL stability discussion in Section III. These characteristics are manifested in the relay structure of r over time. Similarly, since the HTL transition sequence discussed in Section IV can be modeled as a Semi-Markov Process where by state represents an HTL count, and each relay path, e.g., r^1 represents a state within the process. Therefore, according to [18], we can model the steady-state likelihood by observing a state r^i to be equal to the time spent in that state.

The prevalence is defined by the steady-state likelihood of the most frequent relay observation of r during the measurement period. Generally, finding the first or second dominant relay path requires careful and frequent analysis of r at the link granularity, g_k. Similarly, in [11], we examined such characteristics at each routing granularity of the following: Node, Autonomous System (AS), city, HTL, RTT in order to determine how stable is a dominant relay path and its prediction accuracy at each granularity. Throughout our discussion, we abbreviate each granularity by g_n, g_a, g_c, g_h, g_d respectively. Using a higher level granularity for instance g_n avoids the analysis burden at g_k for example, and results in quicker estimations for r. Both g_h and g_d have not been examined in related research for estimating routing changes. Furthermore, g_h and g_d are not location dependent as are g_n, g_a and g_c. For demonstration convenience, we analyze the conditional likelihood of r^i as discussed below. For any observed dominant set, we can define $p(\mathcal{D}_i(r)|r) = \frac{\omega_i(r)}{m}$ where $i = 1, 2, \ldots x^*$ as the steady-state likelihood for any relay path in $\mathcal{D}_i(r)$ or prevalence. Here, and again, $\mathcal{D}_i(r)$ considers all relay paths in $\mathcal{O}(r)$ appear with maximum frequency or ω_1. The size of each dominant set $|\mathcal{D}_i(r)|$ represents the prevalence redundancy as explained in Section VII. Figure 4 shows only our cumulative distribution of prevalence of the first dominant set of all paths in our study at each granularity. For example, at g_n, approximately 51%, on the y-axis was dominated by at least one path with a prevalence of 75%, on the x-axis. Surprisingly, our result was very close to [11], in which 49% of underlay paths had prevalence equal to 80%. This indicates two important points: (1) That our measurements and results is strong enough in order to be generalized, and (2) Having a large-scale measurement allows the capturing of stable view of relay behavior. Similar to [11], we find 30% of our measurements are stable or long-lived because they exhibit a prevalence of one. For g_c and g_a, our spread in prevalence is also narrow as expected in [11], because Planetlab is not diverse enough at such granularities.

In contrast to [18], Figure 4 shows path fluctuations at g_a, and implies changes at g_c and vice-verse for both dominant sets, since prevalence at g_c is always strictly below the one at g_a. More strongly, since g_n curve is strictly above g_a, g_c and g_h, with 100% any changes at g_n are as well reflected at any other granularity. This is because the same lack of diversity within Planetlab makes it rare to find nodes belonging to different ASes within the same city or vice versa. In general, having large prevalence medians, 0.75 at g_n, and 0.81 at both g_c and g_a indicates a wide spread in distribution. The prevalence at g_h is strictly less than either of the remaining granularities as expected, and so a change in a path at g_n, g_c or g_a will always be captured by a HTL change. The only missing fact, however, in order to rely on the HTL count for estimating path changes, is the the percentage of error when facing a no change scenario as Section VIII discusses.

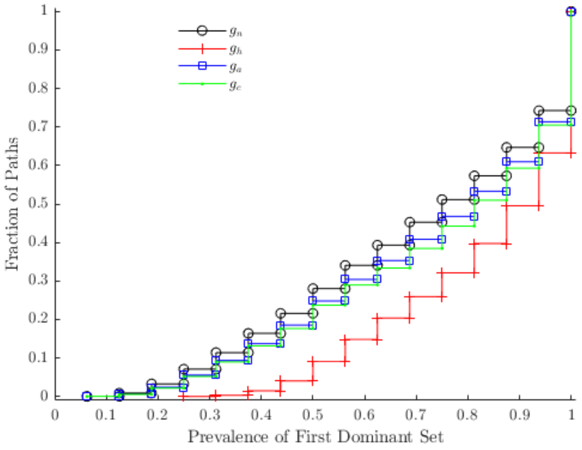


Figure 4. Dominant Path Prevalence.

Similarly, each source of $s \in S = \{1, 2, \ldots N\}$ is associated with an overall prevalence that considers every underlay path r starts from s and in $\mathcal{R}_s = \{r | r : s \rightarrow s' \quad \forall s' \neq s \wedge \exists \mathcal{D}_1(r)\}$. Therefore, source prevalence is calculated as following:

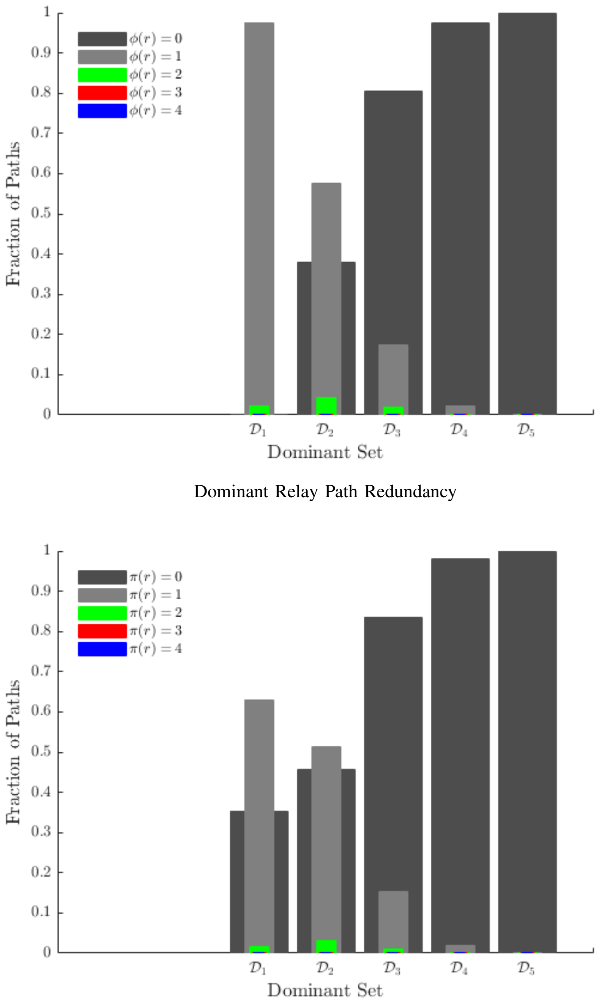
$$p(s|\mathcal{R}_s) = \frac{\sum_{r \in \mathcal{R}_s} \omega_1(r)}{m|\mathcal{R}_s|} \tag{3}$$

The source prevalence in (3) is an average measure of relay routing stability for a source node. Higher $p(s|\mathcal{R}_s)$ indicates a stable dominant relay forwarding via s. Both path and source prevalence are considered as long-term stability characteristics. The source prevalence gives an overall view of the first dominant set of a source s concerning all remaining nodes. The changes near a particular source will affect the prevalence of that source alone [11]. The far path fluctuations in a network will affect all sources not only a particular one [11]. The concept of source prevalence in our study follows that of a similar study [11]. Since underlay routing is not optimized in terms of RTT, oftentimes it is usual to observe that paths out of the same node follow produced disjointed links early on. However, given that our study is an analysis of the shortest relay delay paths, such paths were not expected to be disjoint near their sources, but further into the network. Our analysis shows that on average nodes have 70% of their paths as considerably stable. The prevalence of the second dominant set converges earlier than the first set since being seen as a dominant path reduces the chance of a second dominant one appearing.

Generally, having an underlying path with an extremely small prevalence, for example, the minimum value in our study is 0.0625 indicates a short-lived relay path. However, no relay path showed a minimum below 0.0625 in our study. Low prevalence generally does not exist in underlay routing as [11] confirmed. This is because both intra-domain and inter-domain routing helps to maintain paths with fewer fluctuations. For relay routing, however, the measurement wide view makes such a low prevalence impossible.

VII. Dominant Path Redundancy

Following Section VI, and for reliable relay routing, applications should incorporate a multiplicity of dominant paths, for example as in the dominant set: $|\mathcal{D}_i(r)| = \phi_{\mathcal{D}_i}(r)$. Furthermore an underlay path r with seven relay observations, such as $r^1, r^1, r^2, r^2, r^2, r^1, r^3$ has only two dominant sets with $\phi_{\mathcal{D}_1}(r) = 2$ and $\phi_{\mathcal{D}_2}(r) = 1$. In the dominant set $\mathcal{D}_i$, we were interested in how many paths in $\mathcal{D}_i$ were relay paths. Such a metric $\pi_{\mathcal{D}_i}(r)$ indicates how often relay routing is willing to replace r by better relay paths. For simplicity, we only used HTL to analyze dominant redundancy and type at g_h. Figure 5 shows categorical histograms of both $\phi_{\mathcal{D}_i}(r)$ and $\pi_{\mathcal{D}_i}(r)$, respectively. Clearly, for $\mathcal{D}_i(r)$, $\phi_{\mathcal{D}_i}(r)$ can not be zero as there is always at least one path in $\mathcal{D}_i$. As expected we observed a linear increase in $\phi_{\mathcal{D}_i}(r) = 0$ for $i \in [2 \rightarrow 5]$ as dominant sets tend to be empty for smaller path prevalence. Similarly, $\phi_{\mathcal{D}_i}(r) = 1$ decreases as path prevalence decreases. Note, the summation of the four bars at each dominant set equals one. Figure 5 shows that $\pi_{\mathcal{D}_i}(r)$ follows a similar pattern to $\phi_{\mathcal{D}_i}(r)$ but at different rates. Further, Figure 5 shows that 60% of examined paths have exactly one relay path in their $\mathcal{D}_1$ while 30% have no alternative other than the underlay paths in $\mathcal{D}_1$.



Dominant Relay Path Redundancy

Dominant Relay Path Type

Figure 5. Redundancy and Type of Relay Paths.

VIII. Detection of HTL Changes

The discussion in Section VI shows at g_n, nearly 50% of our $19,460$ examined source-destination paths, are dominated by a single path with prevalence equal to 0.75. The question now is: Can HTL be used by relay nodes in relation to routing changes without any probing overhead? There are many reasons for such a correlation. Detecting relay changes permits nodes to update performance metrics, such as RTT and bandwidth. Therefore, for measurement-based routing, recognizing routing changes is important for future performance estimations.

Current relay systems do not incorporate the hop-count or HTL in their relay headers. This information could be as helpful as the TTL is in the underlay layer. For applications, it could be more important to know the HTL rather than TTL as they could adjust HTL according to their QoS demand. Smaller TTL does not ensure smaller HTL, which means small processing delays of the relay overhead. Furthermore, for end-nodes, a few link changes do not indicate a change in the relay routing as long as such changes still occur between the same relay nodes. Therefore, an additional HTL field should be included in the relay header in order for end-nodes to detect relay changes.

The growth of the Internet might cause routing to exceed the maximum TTL value, 255. Because of the current existence of relay paths with HTL beyond 3 hops in our small network of 140 nodes, we found a widespread distribution in relay link consumption near 70 links, which exceeds the default TTL of 64 Backes et al. [15] and [16]. Hence, relay routing might not improve routing performance in the near future. While forwarding a relay packet via a relay path, the original IP packet is unmodified except for decreasing the TTL. Therefore, for long relay paths, an initial TTL might reach zero before reaching the final destination, and the packet will be dropped. To solve such an issue for the future relay Internet, relay nodes can modify the TTL in the original IP header by allowing nodes to provide a layer-2 update on IP. Via this method relay nodes can access the original IP header to increment the TTL count when necessary. A question raised by such a mechanism is: When should a relay node modify TTL? The relay node increments TTL when the HTL count has not reached zero. There is no need for copying out the original TTL from a selected packet (uniform model) to the outer (relay) IP header before checking and decrementing TTL [20].

The proposed HTL mechanism for detecting relay changes requires including the HTL field in the encapsulated relay header. As a result, relay nodes can determine if a change has occurred without extra probing. The remaining of this section discussed our results for using HTL to detect relay changes. Table I summarizes all associated False-Positive (FP), False-Negative (FN), and total errors when relying on relay routing at g_h for predicting actual relay changes. The FN rate can be reported at every granularity. For instance, when HTL equals 2 hops of two consecutive measurements, corresponding relay paths can be: $r^1 := a \to b \to c$ and $r^2 := a \to d \to c$, although such a change can not be detected at g_h granularity.

The HTL does not result in FP at g_n since any no-change in the nodes involved in a relay path will not be reported as a change by HTL. However, at g_a or g_c HTL can report a FP when HTL changes but the relay path is still constant at those granularities. The FP in our dataset was zero even at g_a and g_c since the Planetlab slice was not dense enough at both granularities.

TABLE I. HTL CHANGE DETECTION

Granularity	FN %	FP %	Error %
g_n	41	0.00	15
g_a	35	0.00	12
g_c	33	0.00	10

IX. HTL-BASED HYBRID UDP-TCP STREAMING

Briefly, this section summarizes our approach to handling UDP-TCP streaming requests. From Figure 6, the controller passes the requester ID to a relay path selection, and receives back all possible stable bandwidth relay paths. The controller then compares the performance of the selector's returned sub-topology in order to choose a relay scheme for the request.

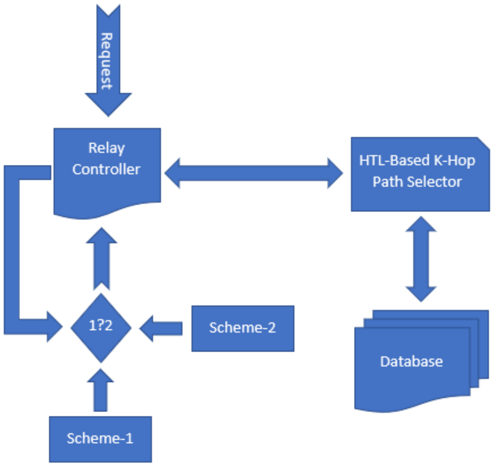


Figure 6. HTL-Based UDP-TCP Request Handling.

A. Bandwidth

Bandwidth is a complex metric measured by tools, such as the listed [13]. None of these tools are robust and scalable. The IPerf is still considered a benchmark in many studies. In our study, IPerf is used to evaluate the UDP bandwidth of all-pairs using 12 rate-demands. We assigned a UDP stream for each rate-demand as detailed in Table II. For each desired rate, all IPerf clients generated the same traffic size for their IPerf servers. The higher rate-demands were attempts to reduce the effect of cross-traffic during our measurements by overloading each examined underlay path with a back-to-back flow of datagrams. Generally, each rate-demand is bounded by the client's network hardware. For ensuring consistency in our measurements, IPerf used its default datagram size of 1500 Bytes as Maximum Transmission Unit (MTU) in order to treat datagrams as packets. However, few clients with multiple interfaces vary their average datagram size due to the distinct hardware of each interface. The average MTU by each node in our Iperf experiments was 1500 Bytes.

TABLE II. UDP RATE-DEMAND AND TRAFFIC

Rate-Demand [Mbps]	Traffic [MBytes]
0.5	0.5
2	4
3	6
5	8
10	10
5	20
5	40
10	80
10	100
10	200
10	400
10	800

For a rate-demand set of link measurements, $\mathcal{L}$, a maximum transmission rate of an egress-interface of a link l is: $r_l^* = \text{arg } \underset{\forall i}{max} \quad r_l^i$, where $i \in \mathcal{L}$. The maximum rate r^* is often called link capacity. The link capacity is always the upper-bound of the available bandwidth $b_l \leq r_l^*$ on a link. Similarly, for path p of $\mathcal{P}$ samples, $r_p^* = \text{arg } max \quad \underset{l \in p}{min} \quad r_l^*$, and consequently, $b_p \leq r_p^*$. Instead of locating a bottleneck with an unnecessary link overhead, an end-to-end bandwidth measurement for a path p should focus on the bottleneck capacity in order to obtain b_p. This study uses IPerf to measure the available bandwidth all-paths conditioned by the given drop-rate threshold τ.

For many applications, an acceptable packet loss in UDP streams is $\%1$. We evaluated our bandwidth under different drop-rates, and examined the change in our measurements as τ varies. For a path p, we defined $b_p = \text{arg } \underset{i}{max} \quad \{r_i | d_p <= \tau\}$ where $i \in \mathcal{P}$. In our study, the number of samples of each path, $|\mathcal{L}| = 12$ as detailed in Table II. However, evaluating b_p is based-on IPerf accuracy but is expensive in probing. We call b_p in our results an expensive bandwidth. Figure 7 shows the cumulative gain of the expensive b_p as τ varies for all $147, 62$ paths. The increase in bandwidth refers to

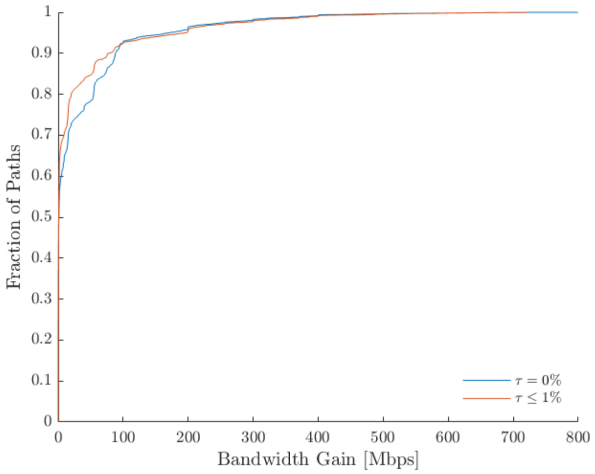


Figure 7. The Expensive Bandwidth Gain.

the difference between the underlay and the relay b_p. The slow CDF convergence indicates more paths gaining more bandwidth using our the relaying scheme outlined above. From Figure 7, we noticed that few paths gained higher relay bandwidths as τ increases. However, the range of $[0 \rightarrow 100]$ Mbps seems to be the dominant bandwidth gain.

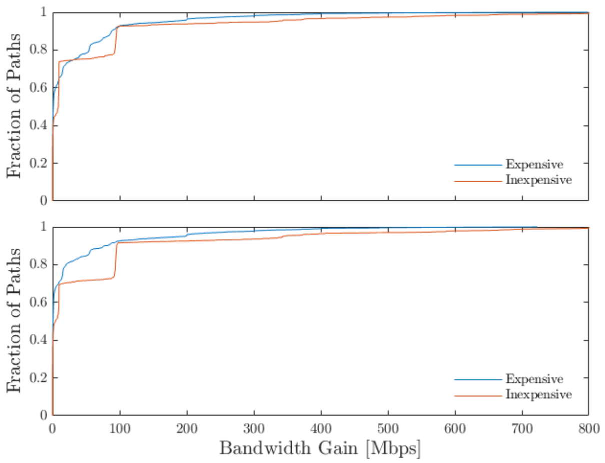


Figure 8. The Expensive vs. Inexpensive Bandwidth Gain (The Upper for $\tau \leq 0$, and Lower for $\tau \leq 1$).

Finding accurate bandwidth estimations requires more careful probing. However, streaming services like YouTube and VoIP applications, such as Skype and Viber might not accommodate such schemes. This paper provides a less expensive probing relaying scheme by choosing a smaller set of demand-rates to evaluate bandwidth. For a Youtube rate-demand, a centralized controller will receive a request from a client with a specific demand, and then probe stable HTL relay paths within the network at the requested demand-rate. This scheme is less expensive in terms of the probing overhead than the previous one. In Figure 8, we compared the performance from Figure 7, in which bandwidth gains were determined for each rate-demand in Table II with only a single set of measurements at a user rate-demand of 800 Mbps.

Figure 8 compares the performance of the expensive scheme in Figure 7, with the inexpensive relay scheme. Our second scheme focused on finding relay paths that offered bandwidth gain and were based on a single rate-demand without the need for the exact bandwidth. However, the expensive scheme used more demand-rates to obtain more accurate bandwidths for all paths before exploring relay gains. Figure 8 indicates that as we reduce our drop-rate demand, or equivalently, increasing τ, we noticed within $[0 \rightarrow 100]$ Mbps, as expected the performance of the inexpensive surpasses its counterpart as the CDF of the later converses earlier. The inexpensive scheme considered only the rate-demand, 800 Mbps, or the maximum possible for a node. However, the actual bandwidth for this node might be below such a high rate-demand. From Figure 8, we concluded that using a less-probing overhead, quickly the inexpensive relay scheme is able to serve UDP-TCP streams at higher rates without exactly determining the available bandwidth for each path. Further, within our examined network, we found that the bandwidth range $[0 \rightarrow 100]$ is the common demand-rate.

B. Hybrid UDP-TCP Success

The use of UDP for streaming instead of TCP is a tradeoff between speed and the handling of the datagram loss. The UDP receiver will discard any duplicate datagrams. There are many studies that analyzed the multi-path [20] [21] or multi-session [22] and [23] TCP performance. For instance, the Multi-Path TCP (MP-TCP) detailed in [19] is a protocol that handles load-balancing and traffic forwarding via multiple paths. The Datagram Congestion Control Protocol (DCCP) described in the Request for Comments (RFC) 4340 can serve as a general congestion-control mechanism for UDP applications to avoid congested links at the IP layer. There are many possible solutions for unreliable connections like UDP streams. First, a joint UDP-TCP connection can solve this issue when the TCP is used as a back-end session for the dropped UDP datagram, while the major portion of the stream still uses UDP. The second solution is to use a similar approach to MP-TCP. A Multi-Path UDP (MP-UDP) allows a UDP sender to duplicate its stream on multiple paths, and the receiver then has a better chance of recovering dropped datagrams. QUIC in [24] duplicates important datagrams but using a single UDP path.

The later solution might introduce duplicate traffic between its receiver and sender as with online-games. For physically compromised client-server connections, we have been trying to determine the likelihood of obtaining a relay path with a minimum drop-rate given a particular rate-demand. Through this path, a UDP client will receive the entire stream without involving TCP to resend incorrect or lost datagrams. Figure 9 shows a relationship between such a likelihood and the rate-demand. The increase in the likelihood of occurs as we raise the rate-demand. This because we find that the current underlay routing is prepared to send UDP streams only at low demand-rates in order to avoid high drop-rates. Therefore, as the demand-rate increases, packets start to experience more loss, and consequently, hybrid relay routing is a suitable solution for such an issue.

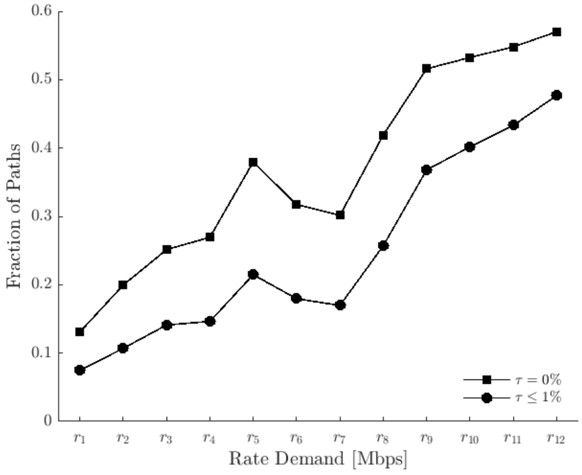


Figure 9. The Likelihood of Successful UDP-TCP Streaming.

X. Conclusion and Future Work

Recently, the speed, at which emerging technologies demand Internet access is higher than the one of developing Internet infrastructure. Relay routing is a principal solution proposed to handle such obstacles at a lower-cost for over a decade. However, relay routing still lacks studies that perform relay characteristics exploration for enhancing performance. This study demonstrates that relay paths are more stable in terms of HTL, and results show that an HTL-based relay path selection assists in reducing the search overhead for better paths as services demand. Furthermore, some services are short-time lived ones, and looking for better paths in large networks might be disadvantageous, especially when the searching time is longer than the service life-time. Hence, our work examines characteristics, such as the relay HTL count to assist in predicting minimum and stable relay paths while minimizing probing overhead. The paper is an attempt for minimizing the propping overhead in overlay schemes and detailing many statistical boundaries for the relay forwarding by analyzing a wide-set of real-time measurements. Our work recommends that an HTL-based relay path prediction is able to determine future paths that reduce drop-rates in streaming services when high transmission-rates are demanded. Further, such an enhancement occurs while reducing probing overhead for minimum delay routing demands by using a single set of measurements. This reduction in probing is a result of the self-similarity model of Internet data.

The study shows that relay paths with HTL ≤ 4 hops are dominant, and they either tend to remain at constant HTL or switch to shorter HTL counts. Their tendency to reduce the number of hops is uniform, Therefore, the focus should be on HTL $\in [2 \rightarrow 4]$ hops when designing stable relay routing. Beyond 4 hops, paths have less stable HTL models. From $19,460$ relay paths, we found $8,863$ paths whose relay option is always better than the underlay routing, and remain within a stable HTL model. An additional $5,824$ paths whose underlay candidates still included also present a miss-free HTL model. Meaning, in total about 75.4% of paths switch their HTLs within stable models during our measurement period. Thus, by excluding constant underlay paths, only 949 paths suffered HTL miss(es). We also concluded that 30% of our measurements are strongly stable or long-lived as they exhibit a prevalence of one. However, regarding the detection of relay changes, the HTL metric was able to detect relay changes with an error of 15%.

For large scale-networks, we find that instead of focusing on exact bandwidth estimates, our second streaming scheme shows that it is highly possible to find other relay paths that can serve a demand-rate quickly. Briefly, we evaluated the performance of a new hybrid UDP-TCP relaying using our HTL-managed relay path selection mechanism. More, precisely, the HTL path modeling was implemented to guide a less-probing path selection for the hybrid UDP-TCP streaming. Here, we simply highlighted a preliminary performance analysis of using the hybrid UDP-TCP streaming carried over layer-3 data-forwarding relay as a replacement for the current TCP-based stream services, such as YouTube and VoIP. Currently, we are expanding our work to design a QUIC counterpart composed of a hybrid UDP-TCP in order to eliminate the introduced QUIC overhead at the user-space while maintaining error-handling, packet security, privacy, and reliability at the kernel-space via the current TCP protocol.

In summary, our analysis emphasizes that a repetitive probing burden for $24-$hours is unnecessary. Instead, a $24-$hours of a careful measurement set is suitable to capture essential path characteristics. The validation of this claim has been justified by presenting that our implemented HTL-based path estimation predicts stable relay paths for the hybrid UDP-TCP streaming to overcome the high drop-rates caused by the individual TCP or UDP streaming.

References

[1] J. Jiang et al., “VIA: Improving Internet Telephony Call Quality Using Predictive Relay Selection,” SIGCOMM, pp. 286–299, 2016.

[2] S. Mohamed, S. Das, S. Biswas, and O. Mohammed, “On The Significance of Layer-3 Traffic Forwarding,” WWIC, Bologna, Italy, pp. 170-181, 2019

[3] J. Kim, A. Chandra and Weissman, “OPEN: Passive Network Performance Estimation for Data-intensive Applications,” Technical Report, pp.8-41, 2008.

[4] A. Su, D. Choffnes, A. Kuzmanovic, and F. Bustamante, “Drafting Behind Akamai, Travelocity-Based Detouring,” In Proceedings of SIGCOMM, pp. 435–446, 2006.

[5] http://www.akamai.com, retrieved: 09-21-2020.

[6] D. Choffnes and F. Bustamante, “On the Effectiveness of Measurement Reuse for Performance-Based Detouring,” INFOCOM, pp. 693-701, 2009.

[7] Z. Wang and J. Crowcroft, “Quality of Service Routing for Supporting Multimedia Applications,” IEEE JSAC, pp. 1228-1234, 1996.

[8] F. Golkar, T. Dreibholz, and A. Kvalbein, “Measuring and Comparing Internet Path Stability in IPv4 and IPv6,” In Proceedings of the 5th IEEE International Conference on the Network of the Future (NoF), pp. 1-5, 2014.

[9] R. Fontugne, J. Mazel, and K. Fukuda, “An Empirical Mixture Model for Large-Scale RTT Measurements,” In Proceedings. IEEE INFOCOM, pp. 2470-2478, 2015.

[10] A. Lareida, D. Meier, T. Bocek, and B. Stiller, “Towards Path Quality Metrics for Overlay Networks,” IEEE 41st Conference on Local Computer Networks, pp. 156–159, 2016.

[11] V. Paxson, “End-to-End Internet Packet Dynamics,” In Proceedings of SIGCOMM, pp. 139-152, 1997.

[12] J. Han, D. Watson, and F. Jahanian, “An Experimental Study of Internet Path Diversity,” IEEE Transactions on Dependable and Secure Computing, pp. 273-288, 2006.

[13] https://www.caida.org, retrieved: 09-21-2020.

[14] https://www.ripe.net, retrieved: 09-21-2020.

[15] M. Backes, “On the Feasibility of TTL-Based Filtering for DRDoS Mitigation,” RAID, pp 303-322, 2016.

[16] “IP Option Numbers: The Current Recommended Default TTL for Internet Protocol is 64 [RFC791] and [RFC1122],” https://www.iana.org/assignments/ip-parameters/ip-parameters.xml, retrieved: 09-21-2020.

[17] S. Fahmy and M. Kwon, “Characterizing Overlay Multicast Networks and Their Costs,” IEEE/ACM Transactions on Networking, pp. 373-386, 2007.

[18] V. Paxson, “Measurements and Analysis of End-to-End Internet Dynamics,” Ph.D. Thesis, pp. 1-386, 1997.

[19] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, “TCP Extensions for Multipath Operation with Multiple Addresses”, Internet-draft, IETF, pp. 1-64, 2011, retrieved: 09-21-2020.

[20] V. Tran, Q. Coninck, B. Hesmans, R. Sadre, and O. Bonaventure, “Observing real Multipath TCP Traffic,” Journal of Computer Communications, pp. 114-122, 2016.

[21] L. Chaufournier, A. Ali-Eldin, and P. Sharma, “Performance Evaluation of Multi-Path TCP for Data Center and Cloud Workloads,” ICPE, pp. 13-24, 2019.

[22] A. Baldini, L. De Carli and F. Risso, “Increasing Performances of TCP Data Transfers Through Multiple Parallel Connections,” ISCC, pp. 630-636, 2009.

[23] T. Nguyen and S. Cheung, “Multimedia Streaming Using Multiple TCP Connections,” PCCC, pp. 215-223, 2005.

[24] J. Iyengar and M. Thomson, “QUIC: A UDP-Based Multiplexed and Secure Transport,” Internet-Draft, https://tools.ietf.org/html/draft-ietf-quic-transport-29, retrieved: 09-21-2020.

[25] A, Langley et al., “The QUIC Transport Protocol: Design and Internet-Scale Deployment,” SIGCOMM, pp. 183-196, 2017.