

HIRE: Hidden Inter-packet Red-shift Effect

Francesco Ciaccia*[†], Ivan Romero[†], René Serral-Gracià*, Mario Nemirovsky[†]

*Universitat Politècnica de Catalunya (UPC), Barcelona, Spain {fciaccia, rserral}@ac.upc.edu

[†]Clevernet Inc., San Francisco, USA {fciaccia, iromero, mario}@clevernet.io

Abstract—Over the years, different techniques have been proposed to detect bottleneck bandwidth and available bandwidth of an end-to-end path. However, to the author’s knowledge, no work has been conducted on detecting which link or node on the path could be the narrow link. In this paper, we present a novel technique based on packet pairs dispersion analysis, whose objective is twofold: first, it allows to estimate the narrow link capacity using a new approach which takes into account both inter-packet time and packet propagation delay. Its second objective is to induce the specific hop in the end-to-end path which represents the narrow link. This is achieved by injecting packets trains with intermediate TTL-expiring packets which decrease the train rate when they cross the narrow link (red-shift effect). We validate our approach in simulations showing the tool robustness in very complex scenarios.

Index Terms—Capacity estimation, active probing, bottleneck determination, hop determination, narrow link determination

I. INTRODUCTION

In today’s Internet, massive amounts of data are being moved due to data hungry applications, for example multimedia content streaming. Such demand, combined with a reduction in cost and the augmented reliability of high speed broadband access, result in new challenges for the future Internet. Even though the Internet core is largely over-provisioned, parts of the network can still represent a bottleneck in presence of cross-traffic coming from multiple sources, especially in access links. End-to-end capacity determination, together with available bandwidth estimation, has always been a key research topic in the networking community [1]–[5]. Such measurements are useful to improve bandwidth allocation, packet scheduling, and congestion control, both in endpoints and routers. Capacity estimation is of the uttermost importance in modern systems that rely on dynamic algorithms for resources allocation, as, for example, 5G mmWave networks. Such environments, and wireless environment in general, present considerable challenges as the resources allocated to the different network devices can frequently vary. Capacity estimation techniques can also be effectively exploited to verify network sizing and planning in enterprises WAN or even detecting Service Provider throttling in residential connections.

In this paper we propose a novel approach to the capacity estimation problem. Our technique improves state of the art analysis of packet pair dispersion, introducing the concept of *packet pair dispersion delay*. The novel concept of *hidden packets* is also introduced. It consists in sending packets with smartly crafted TTL, to later improve the accuracy of the estimation with a series of machine learning approaches.

As a consequence, this technique provides a twofold benefit; on the one hand it allows to compute the link with the smallest available capacity on the path, namely, the narrow link [2]. On the other hand, it provides the specific hop in the network where such narrow link is located, giving a very accurate picture of the network status and topology. An important feature of the proposed contribution is that it can be exploited end-to-end, while requiring a modest amount of network and computational resources, making it suitable for real world usage. We leave as an important part of our future work the implementation of the proposed solution in a real environment. We validate both the accuracy and the feasibility of our algorithm through simulation of environments with realistic traffic loads and cross-traffic events, where we obtain an estimation error for the narrow link capacity of less than 1% in most scenarios.

The rest of the article is structured as follows: Section II delve deeper into the state of the art solutions and shows how our approach improves them; Section III provides the theoretical assumptions on which HIRE approach is built; Section IV describes the algorithm implemented in HIRE to estimate the narrow link capacity; Section V explains the packet red-shift effect and its applications in narrow link location; Section VI shows simulation results validating our approach both in capacity estimation and bottleneck location. Finally, Section VII summarizes the approach and its results with some final remarks.

II. RELATED WORK

Literature regarding packet pair dispersion and capacity estimation is extensive. As of today many schemes for the narrow link capacity estimation have been proposed [1]–[4]. Most models, either based on active probing or passive measurements, measure the dispersion between packet pairs or packet trains. Packet pairs dispersion has been proven to be a noisy measurement [2], [4], [6]. The effort of most works in this area have been focusing on signal processing and de-noising of the intern-packet rate distribution derived from packet pairs dispersion. Few approaches derive available bandwidth measurements by looking at signatures in the queuing delay generated by probe trains emitted with specific patterns [7], [8]. In this work we propose a novel approach to capacity estimation that takes into account both packet dispersion and propagation delay on a tri-dimensional plane. Such approach, combined with state of the art machine learning techniques, provides very precise estimations for the narrow link capacity.

Our study also addresses the narrow link position location on the end-to-end path. Few techniques such as the one presented in [9] provides a solution to determine the position of the currently congested bottleneck links but not the position of the narrow link as well as its nominal capacity. Their technique uses the rate of the ICMP Time Exceeded responses from expiring packets to evaluate the congestion status of a given bottleneck. However, studies such as [10] demonstrate how router manufacturers and Internet Providers usually rate limit ICMP traffic, making measurements based on ICMP replies untimely and unreliable. HIRE on the other hand takes advantage of TTL expiring packets injected in between probing pairs and their packet dispersion-delay analysis to locate the narrow link and estimate its capacity. When adding *hidden* expiring intermediate packets, the probing pairs change their rate when crossing the narrow link, causing a *red-shift* effect that allows for narrow link location. Such a measurement is dependent from precise packet time-stamping which is often a problem, especially in current SDN virtualized environments. Opposed to that, our approach simplifies deployment, easing the tool adoption. The approach for narrow link capacity estimation and location is validated in simulations, showing the tool robustness in very competitive environments, with irregular cross-traffic patterns occupying up to 90% of the end-to-end path.

III. THEORETICAL BASES

A. Smallest Link Capacity Set

Given an end-to-end path, we represent it as an oriented weighted acyclic graph from a source to a destination node. Links weights are the links capacities. We define the Smallest Link Capacity Set (SLCS) as the set of all the links of an end-to-end path which capacity value constitute the narrow link of a complete-path-subset; a complete-path-subset is any sub-segment of the end-to-end path that includes the source node. If in a complete-path-subset multiple consecutive links have the same capacity value, they are considered as a single link. Figure 1 shows a 7-hop path with different link capacities (500, 500, 400, 600, 300, 800, 600 Mbps) from source to destination. Thus, the SLCS is $\{O_1, O_2, O_3\}$ where $O_1 = (300Mbps, link_5)$, $O_2 = (400Mbps, link_3)$, $O_3 = (500Mbps, link_1)$. End-to-end RTT is 280ms. HIRE is able to estimate the capacity value of all the link in the SLCS, starting with the end-to-end narrow link. It achieves this by actively generating packet pair probes and using a new type of packet pair dispersion analysis.

B. Packet pair dispersion

Packet pair dispersion is the basis of many network inference framework. To obtain such measurement, pairs of probe packets of a given size Z are sent over an end-to-end path between two end-points at a given rate R . To be able to measure the narrow link capacity value C_n , R must be higher than C_n . The dispersion of a packet pair τ is the time interval between the instant the last bit of the first packet and the

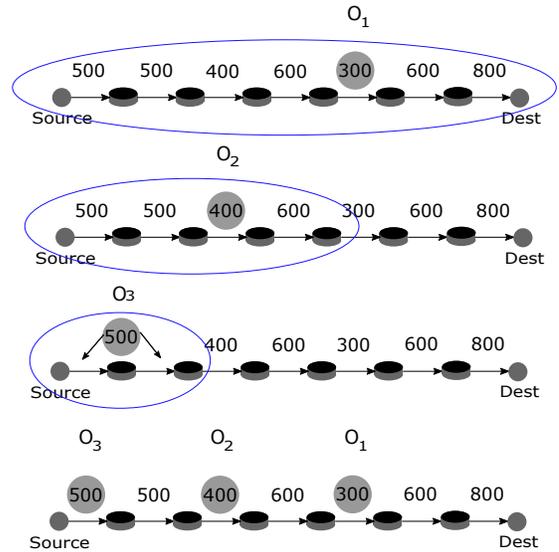


Fig. 1: Smallest Link Capacity Set - SLCS. Capacities are expressed in Mbps.

instant the last bit of the second packet of the probing pair is received at the destination. We derive the *inter-packet rate* as:

$$IP_{rate} = \frac{Z}{\tau} \quad (1)$$

Typically, bandwidth estimation tools collect a certain amount of measurements derived from Equation 1 either from actively probing the network or by means of passive measurements. They then build a histogram representing its distribution during the measurement period. Various techniques have been tested throughout the years to remove noise from these distributions and extrapolate the end-to-end available bandwidth or its narrow link capacity. HIRE does so with a new representation that takes into account not only the packet pair dispersion but also the propagation delay that each of the pair packets incur, and finally estimate the narrow link capacity based on three distributions: the packet pair dispersion and each packet delay distributions.

C. Packet pair dispersion-delay

Consider a N -hop path defined by two sequences: capacities sequence $\mathbf{C} = \{C_0, C_1, \dots, C_N\}$, and delays $\mathbf{D} = \{D_0, D_1, \dots, D_N\}$. In an end-to-end context, we can only get measures on the sender and the receiver end points. When a packet pair reaches the receiver, we can compute the one-way delay (OWD) of each packet of the probing pair: OWD_1 for the first packet received, and OWD_2 for the second packet. When packets traverse a network their propagation delay is affected by different factors: the physical propagation delay in the transmission medium and the processing delay at each hop (serialization, routing, etc...) constitute the deterministic delay, D_{det} . Packets can also incur in buffering delay caused by capacity impairment between links that we define as D_{buffer} . Finally, packets can be delayed due to possible congestion along the path which we define as the stochastic delay D_{sto} .

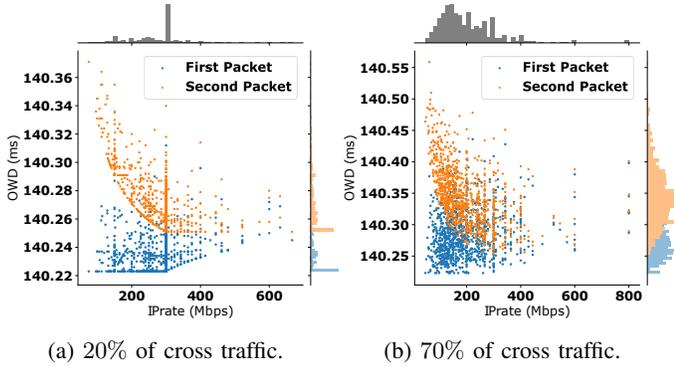


Fig. 2: Packet pair delay dispersion representation under different network conditions. Cross-traffic is one-hop persistent.

As a result, each probing pair packet OWD is obtained as $OWD = \sum_{i=0}^N D_i$. where the delay at each hop i , D_i is given by: $D_i = D_{det,i} + D_{buffer,i} + D_{sto,i}$.

Every time a pair of packets is sent across the end-to-end path, it can be affected by cross traffic (stochastic delay D_{sto}) and/or by limitations of a link capacity (buffering delay D_{buffer}). These two factors can increase the propagation delay of either one or both packets of the pair, also affecting the relative distance τ at which the pair was sent from the source causing variations in the IP_{rate} measured in the receiver. The three values of OWD_1 , OWD_2 , and IP_{rate} configure the *packet pair dispersion-delay (ppdd)* triplet that we use to find the narrow link capacity of an end-to-end path so that each $ppdd_i = (OWD_{1,i}, OWD_{2,i}, IP_{rate,i})$. Figure 2a shows a representation for the metric triad. Blue dots represent the first packet and orange dots the second packet. On the y-axis we plot the OWD of each probe, while on the x-axis is the IP_{rate} of the each probing pair. On the top and right side of the plot are shown the histograms of IP_{rate} , OWD_1 , and OWD_2 respectively. The measurements are obtained from a simulation generated in the ns-2 simulator. The simulated topology is the one shown in Figure 1. 1000 ICMP packet pair probes of size $Z = 1500B$ and an $IP_{rate} = 1Gbps$ are generated. We generate cross-traffic along the path as UDP packets of 1500B. The generation pattern follows a Pareto distribution with a shape value $\alpha = 1.9$. Cross traffic is generated by 16 sources-destination pairs between *each* link (following [2] nomenclature cross-traffic is *one-hop persistent*). In Figure 2a packet pair triplets (IP_{rate} , OWD_1 , and OWD_2) are plotted in a scenario with an average 20% of each link capacity occupied by cross-traffic.

In the absence of cross traffic the only factor adding delay to the packet pair is the capacity limit of the narrow link. When this happens, only the second packet of the packet pair is affected, resulting in an increased value for OWD_2 . This is, $OWD_2 = D_{det} + D_{buffer}$ and $OWD_1 = D_{det}$, where D_{det} is made up with deterministic delays and D_{buffer} is the buffering delay caused by the capacity limitations of the bottleneck. In Figure 3, the blue dashed line represents the minimum OWD value acquired by the first packet. These

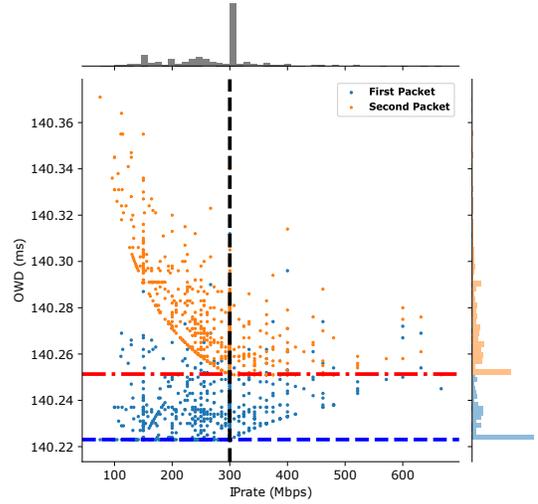


Fig. 3: Red dashed line represents $OWD_2 = OWD_{det} + D_{buffer}$. Blue dashed line is $OWD_1 = OWD_{det}$, not affected by additional delays. Vertical black dashed line is the capacity of the bottleneck on the path, O_1 .

packets are not affected by buffering, so that $OWD_1 = D_{det}$. The red dashed line represents the minimum OWD value acquired by the second packet of the probing pair, which is affected by buffering at the bottleneck due to the capacity limit, giving $OWD_2 = D_{det} + D_{buffer}$. Taking into account that the narrow link capacity is $300Mbps$, which corresponds to an inter-packet of $40\mu s$ with $1500B$ of packet size, and the inter-packet sent is $12\mu s$ (1Gbps), it is possible to observe that the distance between the blue dashed line and the red dashed line corresponds to $D_{buffer} = 28\mu s$. The sum of the distance between the blue and red dashed lines and the inter-packet sent is equal to the inter-packet of the narrow link.

It is worth noting that the maximum value of IP_{rate} achieved by the packets laying over the blue dashed line is the narrow link capacity. Similarly, the minimum value in IP_{rate} achieved by the second packets laying over the red line is the narrow link capacity. Both values coincide over the vertical black dashed line, which is the capacity of the narrow link on the path, O_1 . We distinguish three zones in the relative behavior of the packet pair dispersion and delay: when the IP_{rate} is below the narrow link capacity C_n , D_{sto} due to cross-traffic has a major impact on the second packet of the probing pair, causing a dilation of the packet pair dispersion, and so a lower IP_{rate} . For IP_{rate} values exactly at C_n , D_{sto} affects both packets in the same way, causing either none or a constant incremental offset in the delay. IP_{rate} values bigger than C_n are due to cross-traffic in the links posterior to the narrow link. In this case the first packet of the pair tends to reduce its relative distance in respect to the first packet, causing a reduction in the measured packet pair dispersion. It is worth noting the shape of the packet pair dispersion-delay for the two packets: when $IP_{rate} < C_n$ the second packet **ppdd** distribution follows a concave form; for $IP_{rate} > C_n$ the second packet follows a linear form. The first packet follows a linear form for $IP_{rate} < C_n$ and a convex form

for $IP_{rate} > C_n$. This behavior will be the focus of a future work on the technique to improve its results.

While in the scenario shown in Figure 2a the capacity mode could be detected visually and numerically from the regular packet dispersion, the situation changes in presence of higher level of cross-traffic. In Figure 2b we show a scenario with 70% of cross traffic. To be noted that in nowadays networks such amount of cross-traffic in a one-hop persistent fashion is hardly found. We simulate such a scenario to recreate noisy conditions that can be found in real-world environments which are related to the difficulties involved in performing precise measurements, especially in time-stamping. In such a scenario the capacity mode is not identifiable with a simple visual inspection anymore. Also the OWD distributions vary substantially. The overall **ppdd** distributions shapes do not follow the pattern described previously in a clear way. In Section IV we define a heuristic approach that, combined with state of the art machine learning techniques, allows to identify the capacity mode in the IP_{rate} distribution with a high level of confidence even in noisy environments.

IV. NARROW LINK CAPACITY DETERMINATION

To determine the capacity of the narrow link we take into account the *ppdd* triplets distributions, reducing the area of interest around the capacity of the narrow link. To achieve so, we find a set of *ppdd*, the S_{ppdd} , so that its triads OWD_1 and OWD_2 values are concentrated in the low part of their distributions. We define the three distributions as \mathcal{B}_{IPr} , \mathcal{B}_{owd1} , \mathcal{B}_{owd2} for the IP_{rate} , OWD_1 , and OWD_2 measurements respectively. To select relevant packet pair probes measurements we pick a quantile value q_x from \mathcal{B}_{owd1} that can be statistically close to the linear form that \mathcal{B}_{owd1} describes when the respective pair IP_{rate} values is lower than C_n . We find experimentally that when $x = 30$, the quantile q_x provides a good approximation over a wide range of scenarios, up to 90% of cross-traffic. We then look for a low quantile q_k in \mathcal{B}_{owd2} . We define the S_{ppdd} set as:

$$S_{ppdd} = \{ppdd_i \mid OWD_{1,i} < q_{30}(\mathcal{B}_{owd1}) \text{ and } OWD_{2,i} < q_k(\mathcal{B}_{owd2})\} \quad (2)$$

where $OWD_{1,i}, OWD_{2,i} \in ppdd_i$. The percentile q_k is found iteratively with unitary increments of k until it satisfies the condition of Equation 3:

$$\{k \mid len(S_{ppdd}) > \alpha\} \quad (3)$$

where α is the minimum number of packets that we find to be representative to build the statistics. For this work we find $\alpha = 5$ experimentally. The selection \mathcal{B}_{sel} of the \mathcal{B}_{IPr} distribution is finally extracted from S_{ppdd} :

$$\mathcal{B}_{sel} = \{IP_{rate,i} \mid IP_{rate,i} \in S_{ppdd}\} \quad (4)$$

Once \mathcal{B}_{sel} is obtained as in Equation 4 we apply a kernel density estimation (KDE) transformation over it. KDE is a method to estimate the probability density function of a

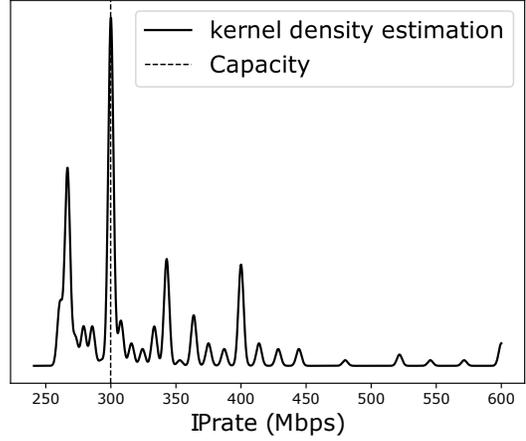


Fig. 4: Capacity determination with Kernel Density Estimation based on the \mathcal{B}_{sel} selection obtained with the proposed heuristic.

random variable obtained from samples of such distribution. The main control parameters for KDE methods are the kernel type and bandwidth. For this study we use a gaussian kernel with bandwidth set to 2. Further optimizations could be obtained with a fine tuning of the KDE bandwidth. Once obtained the KDE continuous distribution, we can search for its global maximum and use it as narrow link capacity estimation. In Figure 4 this technique is applied to the case shown in Figure 2b. With 70% of cross-traffic finding the capacity mode by simply inspecting the \mathcal{B}_{IPr} distribution would have been very difficult. Thanks to the HIRE approach we provide the exact value of the narrow link capacity even in such a noisy environment.

V. HIDDEN RED-SHIFT EFFECT

In this section we define a technique that will be applied both for locating the narrow link position on the path and to estimate the capacity of the rest of links belonging to the SLCS. The technique consists in injecting *hidden* packets H_p between the probing pairs generating packet probing trains. All train packets have size Z . The additional H_p packets are said to be hidden as they are generated with a Time To Live (TTL) value in their corresponding IP header field that will cause their expiration during transit, before reaching the destination. The first and last packet of the train are generated with an $\mathbf{IPS} \equiv IP_{rate,sent} = Z/\tau$. In absence of cross-traffic along the path the dilation effect caused by the hidden packets over the probing pair dispersion measured in the receiver ($IP_{rate,recv}$ that we call \mathbf{IPR}) is proportional to the amount of injected hidden packets and the links queueing delay D_{buffer} added up to the node where the hidden packets expired. Taking advantage of this effect, we can evaluate the differences in the IPR behavior varying iteratively the TTL value of the hidden packets. This enables both narrow link location and SLCS capacity estimation.

A. Narrow link location

Figure 5 shows an example that summarizes the HIRE technique. In this case, four iterations are needed to determine

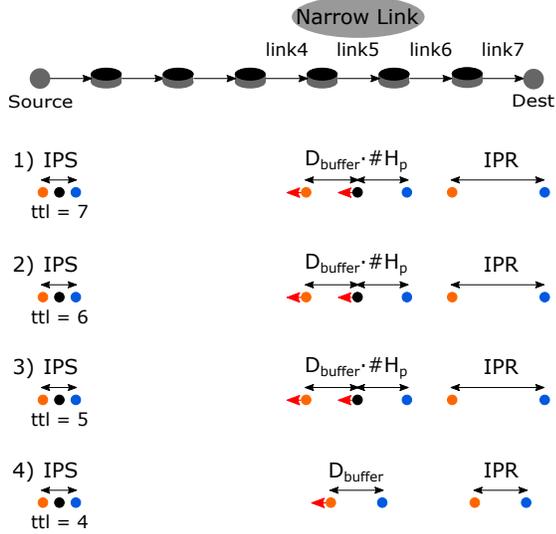


Fig. 5: Schematic example of the Hidden Inter-packet Red-shift Effect (HIRE) over a non-congested path.

the narrow link. Three packets are sent: blue (first packet pair probe), black (hidden packet), and orange (second packet pair probe). Blue and orange have a TTL value big enough to reach the destination. Black packets are the hidden packets and have different TTL at each iteration. Three packets are sent with an IPS rate value between the blue and orange packets equivalent to the narrow link C_n capacity. Additionally, a hidden packet with $TTL = 7$ is sent between the blue and orange packets. When the train of packets reaches the narrow link at hop-5, both orange and black packets suffer a red-shift effect due to D_{buffer} . As a result the inter-packet time between all packets of the train increases. At the same time OWD_1 and OWD_{H_p} increase. When the probing train reaches hop-7 the hidden packet is dropped leaving only the blue and orange packets. The packets are finally captured at the destination with a rate IPR. The process is repeated iteratively reducing the TTL value. Up to $TTL = 5$ the IPR value measured in the destination will be similar. When the hidden packet is generated with a $TTL = 4$, it will expire when reaching hop-4 just before the narrow link. The IPR measured in the destination will increase. Only the orange packet suffers the *red-shift effect* resulting in an increase in the packet dispersion equal to D_{buffer} . To detect this change of behavior we repeat the described process generating a number of probing trains N_t per each TTL value with a fixed amount of hidden packets $\#H_p$. In Figure 6 we apply a heuristic to the 20% cross-traffic scenario to locate the narrow link position. Per each group of $N_{t,i}$ of TTL_i we compute the mean of the IPR values measured by each train. We then use a simple heuristic to detect the behavior change between the groups post and pre narrow link. We start adding the computed mean values to a distribution and compute its standard deviation. If, when adding a new mean value of the TTL_i group, the standard deviation increases more than a given threshold γ , we detect a behavior change and select TTL_{i-1} as the location of the

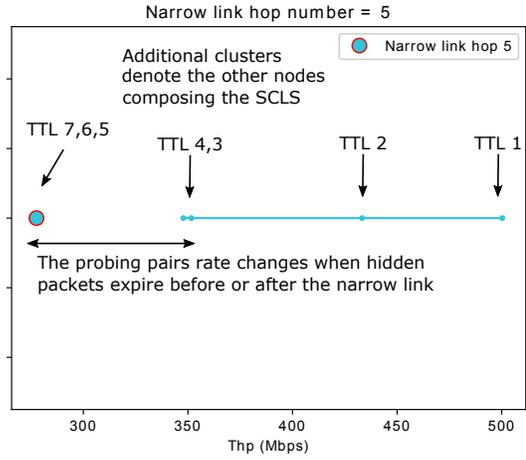


Fig. 6: HIRE applied with $N_t = 20$ and $\#H_p = 5$. The narrow link position is detected when the probing train dispersion behavior changes due to hidden packets expiration.

narrow link. For this study we use $\gamma = 0.2$.

B. SLCS capacity determination

Thanks to the hidden red-shift effect, HIRE is able to determine the capacity of the rest of links of the Smallest Link Capacity Set, SLCS. To achieve it, we need to evaluate first the narrow link capacity and its location. Once we obtain them, we generate another probing session with $IPS = C_n$. The hidden packets should also be generated so that they expire before reaching the narrow link. In this way we guarantee that the probing trains will be forwarded through the narrow link without being buffered there. However they will see a dilation in their dispersion causing the measured IPR to follow Equation 5:

$$IPR = \frac{Z}{(\tau + D_{buffer}) \cdot (\#H_p + 1)} \quad (5)$$

It is then possible to estimate the capacity $C_{n,i}$ of the node i as $C_{n,i} = IPR_i \cdot (\#H_{p,i} + 1)$.

Figure 7 shows this technique applied to estimate the capacity of the second SLCS link of the topology of Figure 1, indicated as O_2 . The simulation environment has 20% of cross-traffic. Probing pair packets are generated with $IPS = 300Mbps$. Five intermediate hidden packets of 1500B are injected in between each pair. Finally, $TTL_{H_p} = 3$ so that the hidden packets will expire right after O_2 but before reaching O_1 . The value provided is exactly the capacity of the O_2 link.

VI. EVALUATION

We provide an evaluation of the HIRE narrow link capacity estimation performance in a variety of simulation scenarios. In Table I we show the narrow link capacity estimation C_n obtained from measurements generated in the topology described in Figure 1. As explained in Section III-C, cross-traffic is generated in form of UDP traffic, following a Pareto random distribution. 16 sources of cross-traffic inject traffic over each link of the path, in a one-hop persistent fashion. The

VII. CONCLUSIONS

In this paper we presented HIRE, a new active probing approach for estimating the narrow link capacity of an end-to-end path based on probing pairs. Its main contributions include a new way of interpreting the packet pair dispersion in conjunction with each pair packet propagation delay that we call the *packet pair dispersion delay* - **ppdd**. We also present the *hidden packets red-shift effect* which consists in injecting TTL expiring packets in between of probing pairs to cause a rate reduction when crossing specific links along the path. Based on this two new concepts we develop different techniques to estimate the narrow link capacity, as well as its location along the path. HIRE can even provide the capacity estimation of other links previous to the narrow link. We validate HIRE in a simulation environment with realistic characteristics. In scenarios with up to 70% random cross-traffic, HIRE provides exact narrow link estimations, with an error lower than 1%. We prove its estimation precision even in spite of very high level of cross-traffic (up to 90%) while using very few probes, with a negligible estimation error of less than 6%.

ACKNOWLEDGMENT

This work was supported by the Industrial PhD grant 2015DI023 of AGAUR and Gencat and the project Efficient Smart Multi Connected Networks co-financed by the Spanish Ministry of Ciencia Innovación y Universidades with reference RTC-2017-6655-7, The Spanish Agenda Estatal de Investigación and the European Regional Development Fund (FEDER).

REFERENCES

- [1] A. B. Downey, "Using pathchar to estimate internet link characteristics," *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 4, pp. 241–250, 1999.
- [2] C. Dovrolis, P. Ramanathan, and D. Moore, "What do packet dispersion techniques measure?" in *Proceedings IEEE INFOCOM 2001*. IEEE, 2001.
- [3] R. Kapoor, L.-J. Chen, L. Lao, M. Gerla, and M. Y. Sanadidi, "Capprobe: A simple and accurate capacity estimation technique," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 67–78, 2004.
- [4] N. Kagami, R. I. T. da Costa Filho, and L. P. Gaspar, "Capest: Offloading network capacity and available bandwidth estimation to programmable data planes," *IEEE Transactions on Network and Service Management*, 2019.
- [5] J. Strauss, D. Katabi, F. Kaashoek, and F. Kaashoek, "A measurement study of available bandwidth estimation tools," in *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*.
- [6] C. Dovrolis, P. Ramanathan, and D. Moore, "Packet-dispersion techniques and a capacity-estimation methodology," *IEEE/ACM Transactions On Networking*, vol. 12, no. 6, pp. 963–977, 2004.
- [7] V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil, and L. Cottrell, "pathchirp: Efficient available bandwidth estimation for network paths," in *Passive and active measurement workshop*, 2003.
- [8] A. K. Paul, A. Tachibana, and T. Hasegawa, "An enhanced available bandwidth estimation technique for an end-to-end network path," *IEEE Transactions on Network and Service Management*, vol. 13, no. 4, pp. 768–781, 2016.
- [9] N. Hu, L. Li, Z. M. Mao, P. Steenkiste, and J. Wang, "Locating internet bottlenecks: Algorithms, measurements, and implications," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 41–54, 2004.
- [10] H. Guo and J. Heidemann, "Detecting icmp rate limiting in the internet," in *International Conference on Passive and Active Network Measurement*. Springer, 2018, pp. 3–17.

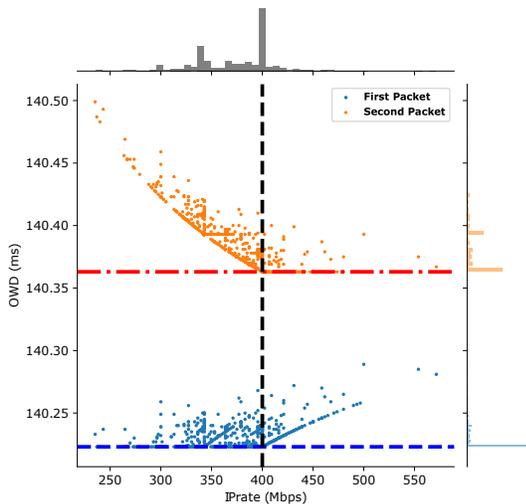


Fig. 7: Capacity determination of the O_2 link of the SLCS.

probing pairs are ICMP Echo Request packets of fixed size $Z = 1500B$. To obtain our estimation we measure OWD_1 , OWD_2 , and IP_{rate} in the destination node.

In Table Ia we show the estimation results obtained with increasing values of cross-traffic when generating 1000 ICMP probes of 1500B. The first column $C_{n,f}$ is with cross-traffic packets of fixed size of 1500B. The column $C_{n,r}$ shows the results when the cross-traffic packets size varies following a uniform random distribution with values between 40B and 1500B. As shown, the estimations are almost always exact, independently from the network conditions. We see a slight underestimation in presence of 80% and 90% random cross-traffic, where HIRE estimation has an error of less than 3%.

In Table Ib we show the technique results when reducing the number of probes generated from 1000 down to only 50. The cross-traffic amount is 70%, a rather high value. The worst estimation is when using only 50 probes; the result produced has a marginal error of approximately 6% against a reduction of 95% of the amount of probing traffic generated.

CT %	$C_{n,f}$	$C_{n,r}$
10%	300.00	300.00
20%	300.00	300.00
30%	300.01	300.00
40%	300.01	300.01
50%	300.05	299.99
60%	300.08	300.00
70%	300.02	300.03
80%	300.09	295.02
90%	299.97	292.66

(a) C_n estimation with increasing amount of cross-traffic.

#probes	$C_{n,f}$	$C_{n,r}$
1000	300.02	300.06
500	299.92	286.13
100	299.90	284.98
50	299.99	282.39

(b) C_n estimation when reducing the amount of probing pairs.

TABLE I: HIRE narrow link capacity estimation summary. Nominal C_n is 300Mbps. $C_{n,f}$ is the estimation obtained with fixed cross-traffic packet size of 1500B, while $C_{n,r}$ with variable size cross-traffic packets.