Application-level versus Network-level Proximity

Mohammad Malli, Chadi Barakat, Walid Dabbous

Projet Planète, INRIA-Sophia Antipolis, France
E-mail:{mmalli, cbarakat, dabbous}@sophia.inria.fr

Abstract. We motivate in this paper the need for application-level proximity. This proximity is a function of network characteristics that decide on the application performance. Most of existing protocols rely on the network-level proximity as for example the one based on the delay (e.g., the delay closest peer is the best peer to contact). In this paper, we study how much the two proximity definitions differ from each other. The work consists of running extensive measurements over the PlanetLab overlay network and comparing different proximity definitions. Our major observation is that the delay proximity is not always a good predictor of quality and that other network parameters are to be considered as well based on the application requirements. Particulary, the best peer to contact is not always the delay closest one. This can be explained by our other observation, that of the slight correlation of network characteristics with each other.

1 Introduction

The emerging widespread use of Peer-to-Peer (P2P) and overlay networks argues the need to optimize the performance perceived by users at the application level. This amounts to defining a proximity function that evaluates how much two peers are close to each other. The characterization of the proximity helps in identifying the best peer to contact or to take as neighbor[1].

Different functions are introduced in the literature to characterize the proximity of peers, but most of them [7, 6, 13, 15] are based on simple metrics such as the delay, the number of hops and the geographical location. We believe that these metrics are not enough to characterize the proximity given the heterogeneity of the Internet in terms of path characteristics and access link speed, and the diversity of application requirements. Some applications (e.g., transfer of large files and video streaming) are sensitive to other network parameters as the bandwidth and the loss rate. Therefore, the proximity should be defined at the application level taking into consideration the network metrics that decide on the application performance. We propose to do that using a utility function that models the quality perceived by peers at the application level. A peer is closer than another one to some reference peer if it provides a better utility function, even if the path leading to it is longer.

---

[1] This work was supported by Alcatel under grant number ISR/3.04.

The question we ask ourselves is whether the delay proximity is a good approximation of the application-level proximity. We try to answer this question with extensive measurements carried out over the Planetlab platform [12]. To this end, we consider a large set of peers and we measure path characteristics among them. We focus on the delay, the bottleneck bandwidth, the available bandwidth and the loss rate. Then, we consider two typical applications: a file transfer running over the TCP protocol, and an interactive audio service. For both applications, we evaluate the degradation of the performance perceived by peers when they choose their neighbors based on the delay proximity instead of the application-level one. The evaluation is done using the utility functions introduced in [8, 3].

Our main observation is the following. Delay, bandwidth and loss metrics are slightly correlated, which means that, in our setting, one cannot rely on one of these metrics in defining proximity when the application is more sensitive to the others. For example, if one uses the delay to decide on the closest peer to contact for a file transfer, the application performance deteriorates compared to the optimal scenario where neighbors are identified based on the predicted file transfer latency. Furthermore, if one contacts the delay closest peer for an interactive audio service, the speech quality is not as high as that obtained when the peer to contact is the one providing the best predicted speech rating. The same result extends to the other neighbors beyond the closest one.

The paper is structured as follows. Next we present our measurement setup. In Section 3, the correlation of the different measured network characteristics is studied. Section 4 illustrates the difference in performance between the delay based proximity and the application based one for the two typical applications we consider. The paper is concluded in Section 5.

2 Measurement setup

Our experiment consists of real measurements run in February 2005 over the Planetlab platform [12]. Although it is widely used, this platform was proved to be appropriate for measurements [17]. We take 127 Planetlab nodes spread over the Internet and covering America, Europe, and Asia. Forward and reverse paths between each pair of nodes are considered, which leads to 16002 measurements. In the following, we call a Planetlab node a peer. All our results concerning peers are averages over the 127 peers.

We measure the end-to-end characteristics of the paths connecting peers using the *Abing* tool [10]. This tool is based on the packet pair dispersion technique [9]. It consists of sending a total number of 20 probe packet-pairs between the two sides of the measured path. It has the advantage of short measurement time on the order of the second, a rich set of results (e.g, bandwidth in both directions), and a good functioning over Planetlab. The measurement accuracy provided by this tool on Planetlab is quite reasonable compared to other measurement tools [11].

For each unidirectional path between two peers, we measure the round-trip time RTT, the available bandwidth ABw, the bottleneck bandwidth (or capacity) BC, and the packet loss rate P. P is estimated as the ratio of the number of lost and sent packets. BC is the speed of the slowest link along the path. For any of these metrics, say X, we denote by $X(p_i, p_j)$ the value of the metric associated to the path starting from peer p_i and ending at peer p_j.

3 Network proximity definitions

Different definitions were studied in the literature for characterizing the proximity among peers, and hence for selecting the appropriate peer to contact. These definitions can be classified into two main approaches static and dynamic. The difference between these approaches lies in the metric they consider. Static approaches [7, 16] use metrics that change rarely over time as the number of hops, the domain name and the geographical location. Dynamic approaches [6, 16, 13] are based on the measurement of variable network metrics. They mainly focus on the delay and consider it as a measure of closeness of peers; the appropriate peer to contact is often taken as the closest one in the delay space. The focus on the delay is for its low measurement cost (i.e., measurement time , amount of probing bytes). However, its use hides the implicit assumption that the path with the closest peer (in term of delay) has the minimum (or relatively small) loss rate and the maximum (or relatively large) bandwidth.

While we believe that the delay can be an appropriate measure of proximity for some applications (e.g., non greedy delay sensitive applications or those seeking for geographical proximity), it is not clear if it is the right measure to consider for other applications whose quality is a function of diverse network parameters. Greedy applications and multimedia ones are typical candidates for a more enhanced definition of proximity. To answer this question, we use our measurements results and study the correlation among path characteristics. We want to check whether (i) the characteristics are correlated with each other, and (ii) how much a proximity-based ranking of peers using the delay deviates from that using other path characteristics. As we will see in this section, there is a clear low correlation among path characteristics which motivates the need for an enhanced model for proximity. The closest peer in terms of delay is far from being optimal in the bandwidth space or loss rate space, and vice versa.

3.1 Delay vs. bottleneck capacity

Take a peer p and let p_d be the closest peer in terms of delay and p_b the best peer in terms of bottleneck capacity. First, we want to study how much the bottleneck capacity of the path connecting p to p_d, $BC(p, p_d)$, deviates from the largest one measured on the path between p and p_b, $BC(p, p_b)$. Figure 1 draws the complementary cumulative distribution function (CCDF) of the ratio $BC(p, p_d)/BC(p, p_b)$. The curve is calculated over all peers. For a value x on the x-axis, the corresponding value on the y-axis gives the percentage of peers

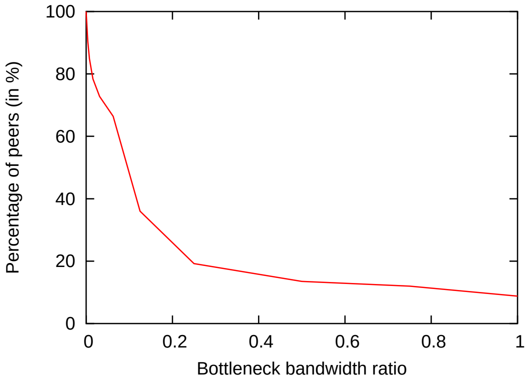

Fig. 1. CCDF of BC on the delay shortest paths

having on their path to the nearest peer a bottleneck bandwidth larger than x times the maximum bottleneck bandwidth.

The figure shows that only (i) 8.8% of peers have the maximum BC on their path to the nearest peer, (ii) 12% have more than 75% the maximum BC, and (iii) 19.2% have more than 25% the maximum BC. This indicates that selecting the best peer in terms of delay leads in most cases to a bottleneck capacity far from the optimal. Applications having a high bandwidth requirement could suffer from this choice.

Now, we generalize our results to the other peers than the closest one. We plot in Figure 2 the bottleneck capacity versus the round trip time for the 16002 paths. Each point in the figure represents one path. The Figure shows that BC does not decrease uniformly when RTT increases. Furthermore, the correlation coefficient between these two variables is small and equal to -0.128[2].

Figure 3 plots the average bottleneck capacity for all peers of rank r in the delay space, r varying from 1 to 126. In other words, for each peer among the 127 peers, we take its neighbor of rank r in the delay space, we measure its bottleneck capacity, then we average this bandwidth over the 127 peers. Again, the figure shows a slow decrease of the bottleneck capacity with the delay-based peer rank. The delay closest peers are far from yielding the best bottleneck capacity.

3.2 Delay vs. available bandwidth

We repeat the same analysis but this time for the delay and the available bandwidth. For a peer p, we denote by p_a the best peer in terms of available bandwidth. Figure 4 shows the CCDF of the ratio $ABw(p,p_d)/ABw(p,p_a)$. This should illustrate how far is the available bandwidth on the delay shortest path from the optimal available bandwidth. The figure is plotted over the 127 peers.

---

[2] One would have expected a coefficient closer to -1 than to 0.

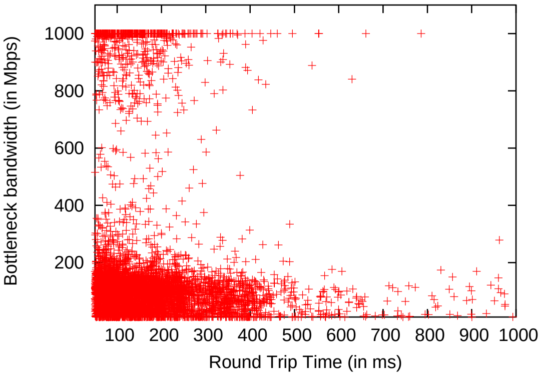

Fig. 2. Variation of BC when RTT increases

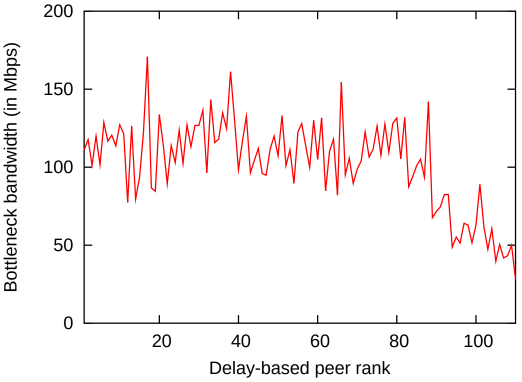

Fig. 3. Variation of BC with the delay-based rank

We can see that only (i) 12% of peers have the maximum ABw on their path with the nearest peer, (ii) 19.2% have more than 75% of the maximum ABw, and (iii) 45.6% have more than 25% of the maximum ABw. Even though these numbers are better than in the bottleneck bandwidth case, the delay is still far from being the proximity metric to use to detect the peer with the maximum available bandwidth.

Figure 5 plots the available bandwidth versus the round trip time for the total 16002 paths. There is no strong correlation between ABw and RTT. In our setting, the two variables are lightly negatively correlated with a coefficient equal to -0.096. Similar result can be observed in Figure 6 where we plot the average available bandwidth for peers of rank r in the delay space, r varying from 1 to 126. We notice that looking at farther and farther peers in the delay space does not lead to an important decrease in the available bandwidth, and so

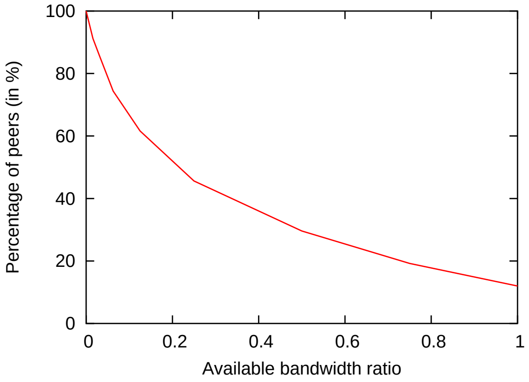

Fig. 4. CCDF of ABw on the delay shortest paths

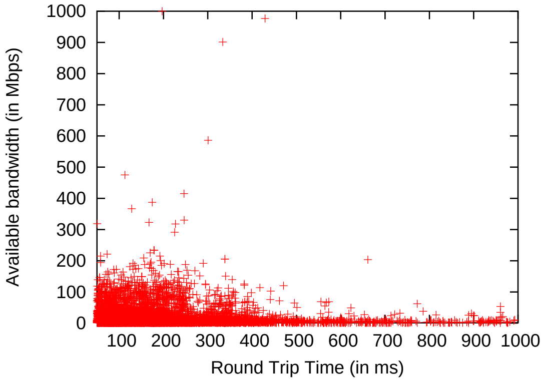

Fig. 5. Variation of ABw when RTT increases

there is a high chance of having the optimal peer from bandwidth point of view located far away (in the delay space) from the peer requesting the service.

3.3 Bottleneck vs. available bandwidth

The bottleneck capacity provides an indication on the maximum performance one can achieve. The available bandwidth indicates how much the network is loaded. It is linked to the bottleneck capacity, but since Internet paths are differently loaded, there should be no reason to think that these two characteristics can replace each other when defining proximity for applications sensitive to the bandwidth. This is what we analyze in this section.

For a peer p, we plot in Figure 7 the percentage of peers having a ratio $ABw(p, p_b)/ABw(p, p_a)$ larger than x, x between 0 and 1. In other words, we check the difference between the available bandwidth on the path having the

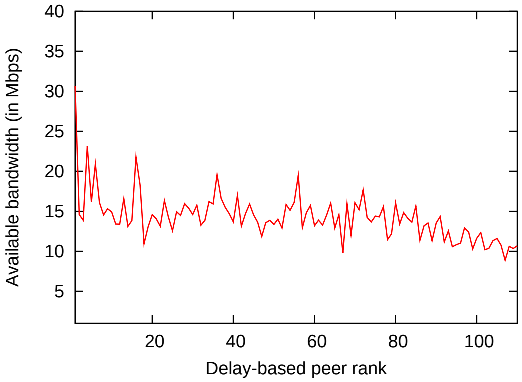

Fig. 6. Variation of ABw with the delay-based rank

maximum bottleneck bandwidth and the maximum available bandwidth. The figure shows that (i) 12.8% of peers have the maximum ABw on the path having the best BC, (ii) 18.4% of peers have more than 75% the maximum ABw, and (iii) 64.8% of peers have more than 25% the maximum ABw. Clearly, selecting the peer with the maximum bottleneck capacity is not equivalent to selecting the one with the maximum available bandwidth, and the error is not negligible.

Then, we study how these two characteristics behave over all peers. We plot in Figure 8 the available bandwidth versus the bottleneck bandwidth for the total 16002 paths. A positive correlation can be seen, which when computed, yields a coefficient equal to 0.475. Figure 9 plots the available bandwidth averaged over all peers having the rank r in the decreasing-order bottleneck bandwidth space. Clearly, the farther a peer in the bottleneck space, the smaller the available bandwidth. But, in spite of this correlation, we suggest not to replace these two metrics in the proximity definition when the application requires one of them. Both need to be considered simultaneously for the proximity definition to be efficient.

3.4 Delay vs. loss rate

Applications are sensitive to the loss rate. We want to check in this section how well a definition of proximity based on delay satisfies the loss rate. We find that all peers have a null loss rate ($P = 0$) on their paths with at least one other peer. To check whether the nearest peer results in the minimum loss rate (i.e., zero), we plot in Figure 10 the cumulative distribution function (CDF) of the loss rate on the path connecting a peer p to its nearest peer p_d. The distribution is computed over the 127 peers. We can see that 87.4% of peers have the minimum loss rate ($P = 0$) on their path to the nearest peer. However, as long as we move away from a peer in the delay space, the loss rate jumps to values on the order of several percents, then it increases slowly. This is illustrated in Figure 11 where

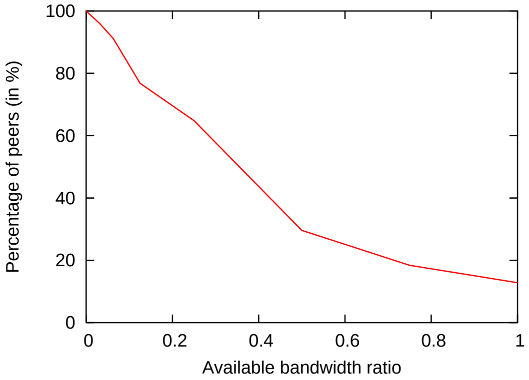

Fig. 7. Percentage of peers having more than the x ratio of maxABw on their maxBC paths

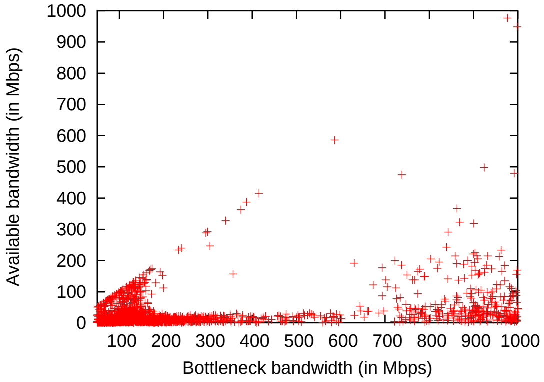

Fig. 8. Variation of ABw when BC increases

we plot the packet loss rate on the path connecting a peer to its neighbor of rank r in the delay space, r changing from 1 to 126. The figure is averaged over the 127 peers.

In our setting, the path with the nearest peer has a minimum loss rate. It seems that it is located in a non-congested neighborhood. Now, when it comes to selecting more than one peer for a certain service sensitive to the loss rate, taking the delay as a metric of proximity stops being efficient, and the loss rate has to be considered as well.

3.5 Load vs. loss rate

Finally, we check the correlation between the network load ($\rho = 1 - ABw/BC$) and the loss rate. Surprisingly, we find these metrics to be lowly correlated,

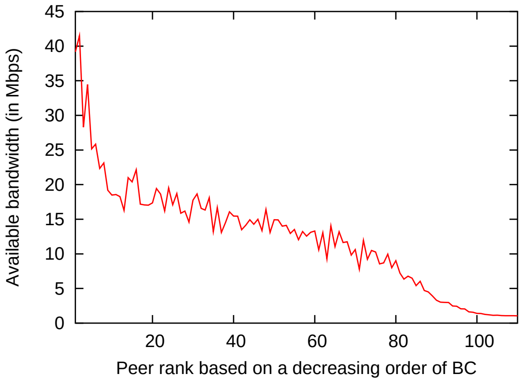

Fig. 9. Variation of ABw with the BC-based rank

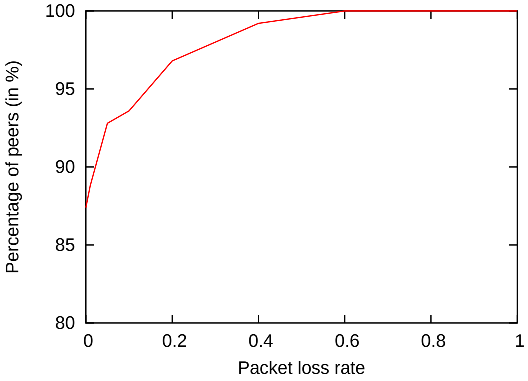

Fig. 10. CDF of loss rate on the delay shortest paths

with a coefficient of correlation equal to 0.0277 in our setting[3]. As in the delay case, we want to check whether the loss rate is satisfied if one takes the load as a proximity metric. Let p_ρ be the peer with the minimum load. We plot in Figure 12 the distribution of $P(p, p_\rho)$ computed over the 127 peers. The figure shows that (i) 66.14% of peers have the minimum loss rate ($P = 0$) on their lowest loaded path, (ii) 79.52% of peers have a loss rate smaller than 0.1, and (iii) 92.12% of peers have a loss rate smaller than 0.5. We complete the analysis by plotting in Figure 13 the packet loss rate as a function of the peer rank in the load space. The line is an average over the 127 peers.

The observation we can make from these figures is that load and loss rate are not highly correlated, and so they need to be both considered simultaneously for an efficient proximity definition (if the application requires both of them).

---

[3] One would have expected these two metrics to be strongly correlated with a coefficient of correlation closer to 1 than to 0.

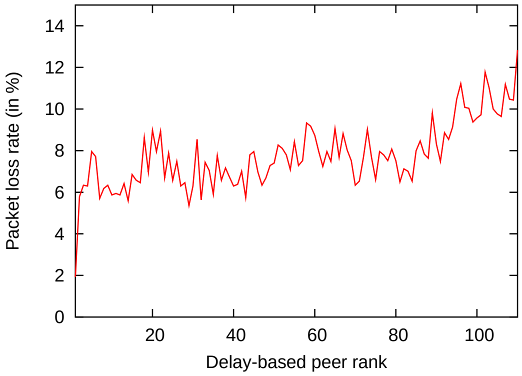

Fig. 11. Variation of P with the delay-based rank

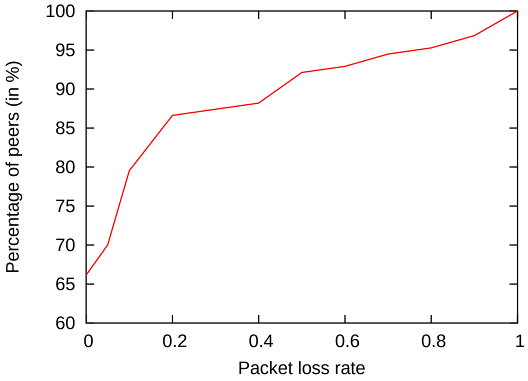

Fig. 12. Percentage of peers having less than the x loss rate on their minimum loaded paths

4 Impact of proximity definition on application performance

The weak correlation among path characteristics pointed out by our measurement results motivates us to compare between the application-level proximity and the network-level proximity. Basically, the proximity can be determined at the network-level by measuring a network parameter as the delay or the bandwidth. But, it can also be determined at the application-level by estimating some utility function that models the application quality such as the transfer time for the file transfer application and the speech rating for interactive audio applications. Such utility function decides on the proximity at the application level.

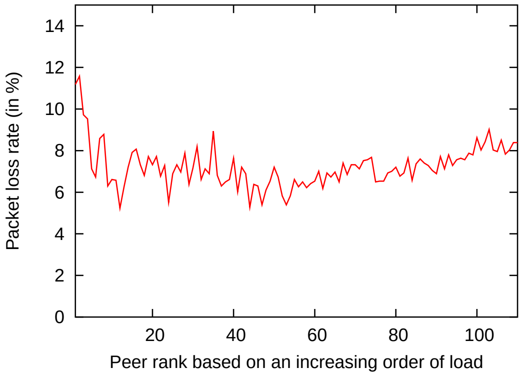

Fig. 13. Variation of P with the load-based proximity

As an example of network proximity definition, we consider the delay based one given its wide use. We check whether the delay proximity is a good approximation of the application-level proximity. To this end, we consider two typical applications, which are: (i) file transfer over TCP protocol, and (2) interactive audio service.

4.1 File transfer over TCP

Firstly, we take the case of file transfer over the TCP protocol. This case can be encountered in the emerging file sharing P2P applications or in the replicated web server context. Applications using TCP are known to form the majority of Internet traffic [4]. For such applications, the optimal peer to select is the one allowing the transfer of the file within the shortest time. We call *latency* the transfer time.

The latency of TCP transfers is known to be a function of diverse network parameters including the available bandwidth, the loss rate, and the round-trip time [14, 8]. The optimal ranking of peers from the standpoint of a certain peer is the one providing an increasing vector of transfer latency. This ranking defines the application-level proximity among peers. Any other ranking results in a different vector and yields a latency increase. We evaluate in this section the degradation of the TCP latency when the delay proximity is used instead of the application-level proximity to perform the ranking of peers from the best to the worst.

To predict the TCP transfer latency, we consider the function PTT (Predicted Transfer Time) that we compute in [8]. This function is the sum of a term that accounts for the slow start phase of TCP and another one that represents the congestion avoidance phase. The function considers the case when a TCP transfer finishes in the slow start with no losses. We omit the window limitation caused by the receiver buffer to allow a better understanding of the impact of path characteristics.

The latency of a TCP transfer depends on the file size. Short transfers are known to be dominated by the slow start phase which is mainly a function of the round-trip time. Long transfers are dominated by the congestion avoidance phase where the available bandwidth and the loss rate figure in addition to the round-trip time. This difference in the sensitivity to network parameters makes interesting the problem of peer ranking for applications using TCP.

The degradation of TCP latency between the delay proximity and the optimal one is computed as follows. Take a peer p and denote the peer having the rank r in the delay space by $p_d(r)$, i.e., the peer having the r-th smallest RTT on its path to p. Denote by $p_o(r)$ the peer having a rank r with the optimal definition based on PTT. Let $PTT(x, y)$ denote the transfer latency between peer x and peer y. We define the *degradation* at rank r as:

$$degradation(r) = \frac{PTT(p, p_d(r)) - PTT(p, p_o(r))}{PTT(p, p_o(r))}. \tag{1}$$

Then, we average the degradation at rank r over all peers p (of number 127). With this degradation function we are able to evaluate how well on average ranking peers based on the delay proximity performs at the application level with respect to the optimal case.

We plot in Figure 14 the transfer time degradation as a function of the rank r for the delay proximity and for different file sizes. The closest 10 peers are considered. The figure shows that the degradation becomes larger when the file size increases, but this degradation decreases for larger r (when we get farther from the peer requesting the transfer). For large files, the degradation can be as high as 150%. For small files, the degradation is small and uniform. This discrepancy between small and large files is due to the different sensitivities the slow start and congestion avoidance phases have to network parameters. Indeed, short transfers are sensitive to the delay, and since the ranking is based on the delay, the degradation is small and we are close to the optimal case. Long transfers are more sensitive to the bandwidth (i.e., bandwidth greedy) and since the bandwidth is uncorrelated with the delay (see previous section), the degradation is large. However, when the rank increases, the transfer time increases in the optimal case and becomes closer to the transfer time obtained when peers are ranked based on the delay. This explains why the delay policy performs better for large rank and file sizes.

4.2 Interactive audio service

We consider now the case of an interactive audio service, where a set of replicated servers are distributed in the network to provide clients with the same audio communication. To serve a client, a central unit is in charge of identifying the server that can provide the best speech quality. As in the case of file transfer, we do not consider the load at end points and subsequently, we ignore the issue of load balancing. This allows to focus on the impact of network path parameters on proximity characterization.

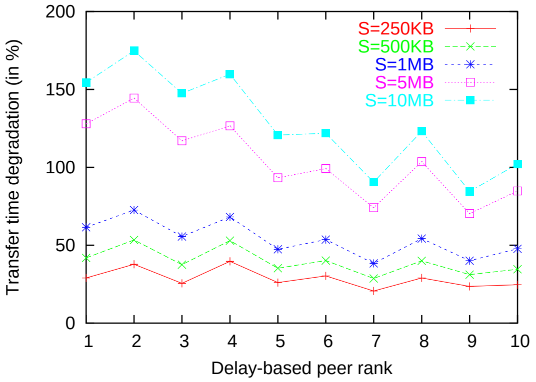

Fig. 14. Transfer time degradation when ranking peers is based on the delay proximity

R interval	Quality of voice rating	MOS
$90 < R < 100$	Best	$4.34 - 4.5$
$80 < R < 90$	High	$4.03 - 4.34$
$70 < R < 80$	Medium	$3.60 - 4.03$
$60 < R < 70$	Low	$3.10 - 3.60$
$50 < R < 60$	Poor	$2.58 - 3.10$

Table 1. R intervals, quality ratings, and the associated MOS

The speech quality suffers mainly from packet loss and delay. The optimal ranking of peers with respect to a certain peer is the one providing a decreased order of the speech quality. Ranking peers in this way defines the application-level proximity. Any other ranking yields a lower speech quality. Our aim is to evaluate how delay-based ranking of peers deviates from the optimal one. This can tell us if the delay-based proximity is a good predictor of the application-level proximity from the standpoint of interactive audio applications.

Speech quality can be characterized subjectively using the Mean Opinion Score (MOS) test. It can also be determined with the E-model, defined by ITU-T G107 [1], which predicts the subjective quality using objective measures (e.g. delay, loss rate). The E-model expresses the audio quality as a rating factor R that accounts for the different transmission parameters having an impact on the conversation. The R-factor calculated by the E-Model ranges from 100 (the best case) to 0 (the worst case). The mapping from the R-factor to the subjective quality and to the MOS score is illustrated in Table 1.

Many papers, as [3], apply the E-model to evaluate the impairments of IP telephony applications and provide expressions for the rating factor R. In this paper, we use the analytical model obtained in [3] and we consider the case of the well known G.711 codec [2]. This model is a reduction of the ITU-T's E-Model.

Let $R(p_i, p_j)$ denote the speech quality rating between peer p_i and peer p_j. According to the E-model proposed in [3], it can be written in the following reduced form:

$$R(p_i, p_j) = R_0(p_i, p_j) - I_d(p_i, p_j) - I_e(p_i, p_j), \tag{2}$$

where $R_0(p_i, p_j)$ is the intrinsic quality of the used codec, $I_d(p_i, p_j)$ is the impairment caused by the end-to-end delay, and $I_e(p_i, p_j)$ is the impairment caused by the end-to-end packet loss.

The end-to-end packet loss process is mainly composed of a network component and another one introduced by the de-jitter buffer (i.e., the buffer used at the end points to compensate jitter). The end-to-end delay is composed of the codec delay component, the delay introduced by the de-jitter buffer , and the network delay component. The codec delay component of G.711 codec is equal to $10 \cdot N$ where N is the number of $10ms$ voice frames packed into a single IP packet. We take $N = 2$ and subsequently set the codec delay to $20ms$ (i.e. encoding and packetization delay). We assume that the de-jitter buffer introduces a $50ms$ delay and a 2% packet loss. These typical values are considered due to the unavailability of a standard reference model for de-jitter buffer implementations. For the network delay component, we take the half of the measured RTT by assuming that the path is symmetric due to the difficulty to measure accurately the one-way delay.

Figure 15 shows the speech quality rating for the r-th closest peer in the delay space and in the optimal space. Recall that the optimal ranking of peers corresponds to a decreased order of the R-factor. The delay ranking of peers is obtained by the increased order of the RTT. The closest 10 peers are considered. We observe in this figure that the application-level proximity provides an optimal ranking with an R-factor around 85, which represents a high audio quality over the 10 closest peers. On the other hand, the delay-based proximity provides a ranking with an R-factor changing between 80 and 60 over the 10 closest peers. This represents a medium audio quality for the 7 delay closest peers and a low quality for the remaining three others.

We conclude that the delay-based proximity is a poor predictor for the interactive audio quality. This stems from the non consideration of the loss rate.

5 Conclusion

We introduce in this paper a new notion of proximity that accounts for path characteristics and application requirements. With extensive measurements over the Planetlab platform, we motivate the need for this notion of proximity by showing that path characteristics are not highly correlated, and so a proximity in one space, say for example the delay, does not automatically lead to a proximity in another space as the bandwidth one. Thus, the proximity needs to be defined as a function of the metrics impacting the application performance. In a future work, we will focus on the deployment of this new definition of proximity and on the evaluation of its gain with other application types.

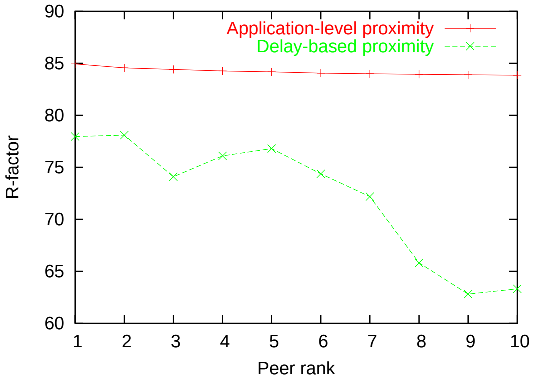

Fig. 15. Speech quality rating when ranking peers is based on the delay and on the optimal proximity

References

1. ITU-T Recommendation G.107: The E-model, a computational model for use in transmission planning, 2000.
2. ITU-T Recommendation G.711: Pulse Codec modulation (PCM) of voice frequencies, 1988.
3. R. Cole, and J. Rosenbluth: Voice over IP Performance Monitoring, ACM SIGCOMM Computer Communication Review, v. 31 , p. 9 - 24, Sigcomm'01, 2001.
4. Cooperative Association for Internet Data Analysis, *http://www.caida.org/*.
5. L. Ding, and R. Goubran: Speech Quality Prediction in VoIP Using the Extended E-Model, IEEE Globecom, 2003.
6. F. Dabek, R. Cox, F. Kaashoek, and R. Morris: Vivaldi: A Decentralized Network Coordinate System, Sigcomm'04, 2004.
7. B. Gueye, A. Ziviani, M. Crovella, and S. Fdida: Constraint-Based Geolocation of Internet Hosts, IMC'04, 2004.
8. M. Malli, C. Barakat, and W. Dabbous: An Efficient Approach for Content Delivery in Overlay Networks, CCNC'05, 2005.
9. J. Navratil, L. Cottrell: ABwE: A Pratical Approach to Available Bandwidth Estimation, PAM'03, April 2003.
10. J. Navratil, L. Cottrell: available at *http://www-iepm.slac.stanford.edu/tools/abing/*, 2004.
11. J. Navratil: available at *http://www.slac.stanford.edu/ jiri/PLANET*, 2004.
12. An open, distributed platform for developing, deploying and accessing planetary-scale network services, see *http://www.planet-lab.org/*.
13. L. Tang, and M. Crovella: Virtual Landmarks for the Internet, IMC'03, 2003.
14. N. Cardwell, S. Savage, T. Anderson: Modeling TCP Latency, Infocom'00, 2000.
15. E. Ng and H. Zhang: Predicting Internet network distance with coordinates-based approaches, IEEE Infocom, 2002.
16. V. Padmanabhan and L. Subramanian: An Investigation of Geographic Mapping Techniques for Internet Hosts, SIGCOMM'01, 2001.
17. L. Peterson, V. Pai, N. Spring, and A. Bavier: Using PlanetLab for Network Research: Myths, Realities, and Best Practices, technical report, 2005.