On the Estimation of Internet Distances Using Landmarks

Stijn van Langen, Xiaoming Zhou, and Piet Van Mieghem

Abstract— **Many networked applications can benefit from a quick estimation of distances to a large number of hosts. Landmark schemes provide such estimates based on distances in a hyperspace in which the hosts are embedded. We evaluate a simple landmark scheme using empirical data for both delay and hopcount. We investigate various choices for the hyperspace distance function. We also compare the measurements with simulations on random graphs with minimal link weight routing.**

Keywords— **Landmarks, distance estimation, delay, hopcount.**

I. Introduction

In various types of networking applications there is a need for quickly determining how far nodes are separated from each other. An example is when a client has to select the nearest from a set of equivalent servers. Similarly, optimization of overlay networks like peer-to-peer networks often requires that nodes connect to peers in their neighborhood. Usually "sepa-ration" or "neighborhood" is defined in terms of the network delay experienced in a packet exchange, but we will use the general term "network distance" to include other quantities as well. In particular, besides the delay we will study the hopcount.

Actively measuring network distances between many pairs of nodes imposes a large load of probe packets on the network. A delay measurement requires at least a few packet exchanges to filter out noise. A hopcount measurement using for instance traceroute is even more expensive. Mapping-techniques based on landmarks or beacons attempt to solve this problem[1][2]. In these techniques each of the N nodes measures its distance to a small set of M well-known landmark servers. By conceiving the results as the components of a vector, each node is embedded in a M-dimensional hyperspace. Now it is assumed that the distance between two network nodes can be estimated by computing the (e.g. Euclidean) distance between their respective coordinate vectors in the hyperspace. The assumption implies that one can estimate the entries of a large $N \times N$ distance matrix when knowing only M rows or columns. It is important to stress that the method is entirely heuristic; the only a priori basis is the intuition that nearby nodes in the network are likely to have "nearby" coordinate vectors. Purely the fact that it seems possible in practice justifies its use.

The authors are with Delft University of Technology, Faculty of Electrical Engineering, Mathematics, and Computer Science, P.O. Box 5031, 2600 GA Delft. Email: {S.vanLangen, X.Zhou, P.VanMieghem}@ewi.tudelft.nl

In this paper we evaluate landmark-based distance estimation using various hyperspace distance functions. Our empirical evaluation is based on data from the RIPE NCC TTM project. This data-set contains measurements of both delay and hopcount between a set of about 60 test-boxes. The network delay directly influences the user experience in many applications. There can be an indirect influence as well because the TCP throughput is roughly inversely proportional to the round-trip time. The hopcount, on the other hand, is usually not a network distance of direct relevance to the user. It is however an important quantity from a network-centric point of view. When nodes connect to peers which are separated by a small number of IP hops, the amount of traffic traversing network boundaries can potentially be limited. This argument of "segregating traffic by topology" has also been given in [2], but a detailed account of the efficacy still has to be made.

II. Related work

A closely related work is the GeoPing approach. GeoPing was originally intended to infer geographical locations of nodes from delay measurements to a set of landmarks [1], but later it was also used to infer the delay itself [3]. In the latter form it coincides with the method studied by us, except that they just use the Euclidean hyperspace distance. A main conclusion of our work is that for predicting the delay, the Euclidean distance function is not the optimal choice. A minor difference is that they use the median delay in their later paper, whereas we use the minimum delay.

Guyton and Schwartz [2] studied the problem of selecting the nearest Internet servers in terms of hopcount. They compare various methods including (among several methods requiring router-support) the landmark-based scheme. They use the "triangulation" hyperspace distance function D_A which was first proposed in [8]. The focus in [2] is on the cost of the methods in terms of the number of packets exchanged. The effectiveness of "triangulation" is addressed as well, but using a different measure than we do. We do not consider costs and focus on the landmark-method, for which we evaluate the effectiveness using various hyperspace distance functions. Our analysis confirms that D_A is a satisfactory choice for the hopcount. An added value of our experiment is that we can compare the effectiveness of hopcount and delay estimation on the same set of nodes.

Global Network Positioning (GNP) by Ng and Zhang [7] is a different estimation scheme which is also based on landmarks. In this approach not only the peers but also the landmarks are embedded in the hyperspace, based on the measured inter-landmark distances. The assignment of coordinates to landmarks and other nodes is cast in the form of a minimization problem where the objective function quantifies the errors in the distance estimates. This minimization should be solved online, which imposes a computational load on nodes and landmarks. The authors provide an extensive comparison between GNP, the "triangulation heuristics" D_+, D_∞, and D_A that we consider in this paper, and IDMaps (see below). They concluded that D_+ is the best triangulation heuristic for the delay. This is confirmed by our analysis. GNP is more accurate, at the cost of a higher computation and communication complexity. Lighthouse [5] and SCoLE [6] improve on GNP in the sense that the role of the landmarks is progressively decentralized.

IDMaps [4] is an infrastructure for estimating distances. Their "tracers" are servers similar to our landmarks, except that they actively measure their distances towards other tracers and towards domains on the Internet.

Various overlay networks use distance estimates to construct efficient topologies. Ref. [9] describes the use of a landmark scheme in a distributed hash-table named CAN.

III. Problem description and definitions

Let $d(A, B)$ be the network distance between node A and B. We assume that $d(A, B) \geq 0$. Note that our "network distance" is not a mathematical distance function. In particular the triangle inequalities may not hold.

The M landmark servers are denoted as $L_1, ..., L_M$. To each node X_i we assign a coordinate vector

$$\boldsymbol{x}_i \equiv (\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, ..., \boldsymbol{x}_{iM})$$

where $\boldsymbol{x}_{ik} = d(X_i, L_k)$ is the measured distance between the node i and the landmark k. The hyperspace distance between the coordinate vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is written as $D(\boldsymbol{x}_i, \boldsymbol{x}_j)$, to distinguish it from the network distance d. The central question of our paper is: To which extent is $D(\boldsymbol{x}_i, \boldsymbol{x}_j)$ correlated with $d(X_i, X_j)$? We answer this question for the following functionals:

$$\begin{aligned}
D_q(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \left(\sum_{k=1}^{M} (\boldsymbol{x}_{ik} - \boldsymbol{x}_{jk})^q \right)^{1/q}, q > 0 \\
D_+(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \min_{k=1,...,M} (\boldsymbol{x}_{ik} + \boldsymbol{x}_{jk}) \\
D_A(\boldsymbol{x}_i, \boldsymbol{x}_j) &= (D_+ + D_\infty)/2 \\
D_G(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \sqrt{D_+ D_\infty}
\end{aligned}$$

The Holder q-norm D_q generalizes the Euclidean distance D_2 which was used in [3]. For $q = \infty$, D_q reduces to $D_\infty(\boldsymbol{x}_i, \boldsymbol{x}_j) = \max_{k=1,...,M} |\boldsymbol{x}_{ik} - \boldsymbol{x}_{jk}|$. The functions D_∞ and D_+ play a special role in the sense that they are a lower and upper bound for the network distance d, assuming that d satisfies the triangle inequalities. Based on this observation, the arithmetic average D_A of D_∞ and D_+ has been proposed as a hyperspace distance [8][2]. We also include the geometric mean D_G. If d satisfies the triangle inequalities we have

$$D_\infty \leq D_G \leq D_A \leq D_+$$

IV. Experiment

Our data are provided by RIPE NCC TTM (Test Traffic Measurement) project. The TTM is a measurement infrastructure designed and run by RIPE NCC as a commercial service offered to ISP's. The TTM infrastructure consists of approximately 60 measurement boxes scattered over Europe (and a few in the US and Asia). Between each pair of measurement boxes, IP packets of a fixed length (100 bytes), called probe-packets, are continuously transmitted with interarrival times of about 40 seconds, resulting in a total of about 2160 probe-packets per day. The sending measurement box generates an accurate time-stamp synchronized via GPS in each probe-packet, while the receiving measurement box reads the GPS-time of the probe-packet on arrival. The end-to-end delay is defined as the difference between these two time-stamps and has an accuracy of 10 μs. The hopcount is measured every 6 minutes using traceroute. We have analyzed the data collected by TTM on May 15, 2003, at which time there were 58 active boxes, where 48 hosts are located in Europe, 7 in the US, and 1 in Japan, Australia and New Zealand. The map in Figure 1 shows the geographical distribution of the test-boxes.

For each sender-destination pair we computed the minimum end-to-end delay over 24 hours (that is approximately 2160 probe-packets), in order to know the congestion-free delay. The RIPE TTM differs from other infrastructures like PingER and AMP in that it measures one-way delays rather than RTTs. In this paper however we do not care about asymmetry, and we always consider the symmetrized network distance, defined as the sum of the network distances in both directions. For the delay, the result can be considered as a round-trip time. For the hopcount we considered only the most dominant path during 24 hours. We omitted pairs for which the delay or hopcount in one of the directions is missing, leaving in total 1503 pairs and 1024 within Europe. These measurements provide us with the network distance matrix $d(A, B)$.

Of all the test-boxes, 10 are assigned the status of landmark. We chose them randomly except that 6 are located in the EU, 1 is located in the Asia and 3 are located in the US. The remaining test-boxes are referred to as ”peers”. We repeated the experiment with $M = 5$ respectively $M = 15$ landmarks.

A. Delay

In Figure 2 we plot the data for the delay, using 10 landmarks. The figure shows six panels, corresponding with six different hyperspace norms: D_1,

Fig. 1. Location of the RIPE test-boxes

D_2, D_∞, D_+, D_A, and D_G. Each data point corresponds to a pair of peers. On the horizontal axis is their actual network distance $d(A, B)$, on the vertical axis their distance predicted by the landmark scheme, that is their hyperspace distance $D(\boldsymbol{x}_A, \boldsymbol{x}_B)$. The lines shown are the least-squares fitted line, and the diagonal $D = d$. The inset also shows the linear correlation coefficient,

$$r(D, d) = \frac{cov(D, d)}{\sqrt{var(D) \cdot var(d)}}.$$

We repeated the experiment using less a smaller (5) respectively larger (15) number of landmarks. We also repeated the experiment on the subset of test-boxes (landmarks and peers) which are all in Europe, again with 5, 10, and 15 landmarks. The results are summarized in Table I. Each row corresponds to a different hyperspace distance function; each column corresponds to a different number of landmarks, selected from the complete set of test-boxes ("All") or from those in Europe only ("Eur"). The first number in each cell is the linear correlation coefficient $r(D, d)$. It makes sense to study also Spearman's rank correlation coefficient

$$r'(D, d) \equiv r(rank(D), rank(d))$$

quantifying the ability to predict which nodes are closest irrespective of the actual values of the distance. The rank correlation is given as the second number in each table cell.

The D_+ hyperspace distance gives the highest correlations, in particular higher than for the Euclidean distance D_2. The results for D_A and D_G are almost as good as D_+.

B. Hopcount

We now turn to the results for the hopcount. Figure 3 shows the results for 10 landmarks. The complete results including those for $M = 5$ and $M = 15$ landmarks are summarized in Table II with the same notation as Table I.

The plots and the table both show that for the hopcount, as compared to the delay, the correlation between estimated and actual distance is smaller. The functional D_A shows the largest correlation. Another interesting observation about the hopcount is that the majority of data-points in the scatterplot for D_∞ (D_+) is located below (above) the diagonal. This means that in practice these lower and upper bounds derived by assuming the triangle inequalities are satified more often than not. Note that a violation of only one of the inequalities for one of the triangles $X_i X_j L_k$ is sufficient for a violation of either the lower bound (i.e. $d < D_\infty$) or the upper bound ($d > D_+$).

V. On network distance triangulation

We have seen that although the delay and hopcount do not by construction satisfy the triangle inequalities, a prediction scheme based on the triangle inequalities can be relatively successful. Apparently these network distances are sufficiently correlated between triplets of nodes. In this section we try to shed some light on the possible origin of such correlations. The most obvious origin is the fact that a communication network is embedded in the real, Euclidean world. Even when network distances are only weakly correlated to geographical distance, the network distance matrix will bear some reminiscence of the geographical triangle inequalities.

Geographical embedding is however not a necessary condition for correlations in the network distance matrix. Consider an arbitrary network topology where the paths are sought to minimize the sum of given link weights. Note that in such a scenario the path weights obey the triangle inequalities by construction, whereas the hopcount does not. We did some simulations in order to see to which extent both types of network distances can be predicted using landmarks.

For sake of simplicity we simulated random graphs with i.i.d. link weights w. We applied landmark estimation to both the path weight and the hopcount, for the six hyperspace distances defined before. In particular we generated random graphs of 2000 nodes and link probability $p = 0.01$. Each link weight w is drawn from a polynomial distribution,

$$\Pr[w \leq x] = x^\alpha \cdot 1_{x \in [0,1]}$$

where α is a positive real number. For large values of α the link weights are nearly constant (and equal to unity), so the path weight becomes identical to the hopcount. For $\alpha = 1$ the link weight distribution is uniform. For small values of α the fluctuations of the link weights are much larger than their average. As representative values for these regimes we consider $\alpha = 0.1$, 1, and 10. For each value of α, 10^4 random graphs were generated, and in each of them, 10 random nodes were elected as landmarks, and 10 others were randomly chosen as "peers". In Table III we summarize the results. The column names "hops" and "weights" correspond with the choice of the network distance being estimated.

The following observations are made from these data:

• The correlations for "hops" are always smaller than for "weights". This was more or less expected, in view of the fact that the (minimized) path weights satisfy the triangle inequalities, while their hopcounts do not. For larger values of α the difference between hops and weights will disappear, and both will satisfy the triangle inequalities. (The results shown for "hops" and $\alpha = 10$ are somewhat anomalous for the network size studied because the hopcount is only a small integer.)
• For $\alpha = 0.1$ and "weights" the correlations are nearly perfect. Also the correlations for the hopcount have reached a moderately high level. This can be understood because for sufficiently small α, all shortest paths in the graph lie on a single spanning tree [10]. As a consequence, the triangle inequalities are satisfied not only for path weights but also for the hopcount.

In conclusion, the hopcount on minimal weight paths satisfies the triangle inequalities both in the limit of weak and strong link weight fluctuations. This result is actually quite robust and independent from the graph topology and details of the link weight distribution. In the Internet path selection is obviously more complicated since routing policies play an important role. It remains an interesting question to model the effect of these policies on network distance matrices.

VI. Conclusions

We evaluated a landmark-based estimation scheme using real data for the delay and hopcount between Internet hosts. In particular we studied which hyperspace distance function correlates best with the actual distances between hosts. From our experiments and simulations we could draw the following conclusions:
• The best results for the delay estimation are obtained using D_+.
• The best results for the hopcount estimation are obtained using D_A.
• Hopcount is generally harder to estimate than delay.
• Theoretically, we found that even if a network is not embedded in a Euclidean space, and routes are based on link weight minimization, the hopcount distance matrix can be sufficiently correlated to support landmark-based estimation.

Acknowledgements

We are grateful to Henk Uijterwaal (RIPE) for the use of the RIPE measurement system. Part of this work has been funded by the NWO SAID project.

References

[1] V. N. Padmanabhan and L. Subramanian, *An Investigation of Geographic Mapping Techniques for Internet Hosts*, Proc. ACM Sigcomm, San Diego, CA, USA (2001).
[2] J. D. Guyton and M. F. Schwartz, *Locating Nearby Copies of Replicated Internet Servers*, Proc. ACM Sigcomm, Cambridge, MA, USA (1995).
[3] K. Lakshminarayanan and V. N. Padmanabhan, *Network Performance of Broadband Hosts: Measurements & Implications*, Microsoft Technical Report (2003).
[4] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, *IDMaps: A Global Internet Host Distance Estimation Service*, IEEE/ACM Transactions on Networking 9, 525 (2001).
[5] M. Pias, J. Crowcroft, S. Wilbur, T. Harris, and S. Bhatti, *Lighthouses for Scalable Distributed Location*, Proc. of the 2nd International Workshop on Peer-to-Peer Systems, Berkeley, CA, USA (2003).
[6] M. Szymaniak, G. Pierre, and M. van Steen, Scalable Cooperative Latency Estimation, preprint (2003)
[7] T. S. E. Ng and H. Zhang, *Towards Global Network Positioning*, Proc. ACM Sigcomm Internet Measurement Workshop, San Francisco, CA (2001); *Predicting Internet Network Distance with Coordinates-Based Approaches*, Proc. IEEE Infocom, New York, NY, USA (2002).
[8] S. M. Hotz, unpublished (Technical Report, University of Southern California, 1994; PhD thesis 1996).
[9] S. Ratnasamy, M. Handley, R. Karp, S. Shenker, *Topology-Aware Overlay Construction and Server Selection*, Proc. IEEE Infocom, New York, NY, USA (2002).
[10] L. A. Braunstein, S. V. Buldyrev, R. Cohen, S. Havlin, and H. E. Stanley, *Optimal Paths in Disordered Complex Networks*, Phys. Rev. Lett. 91, 168701 (2002).

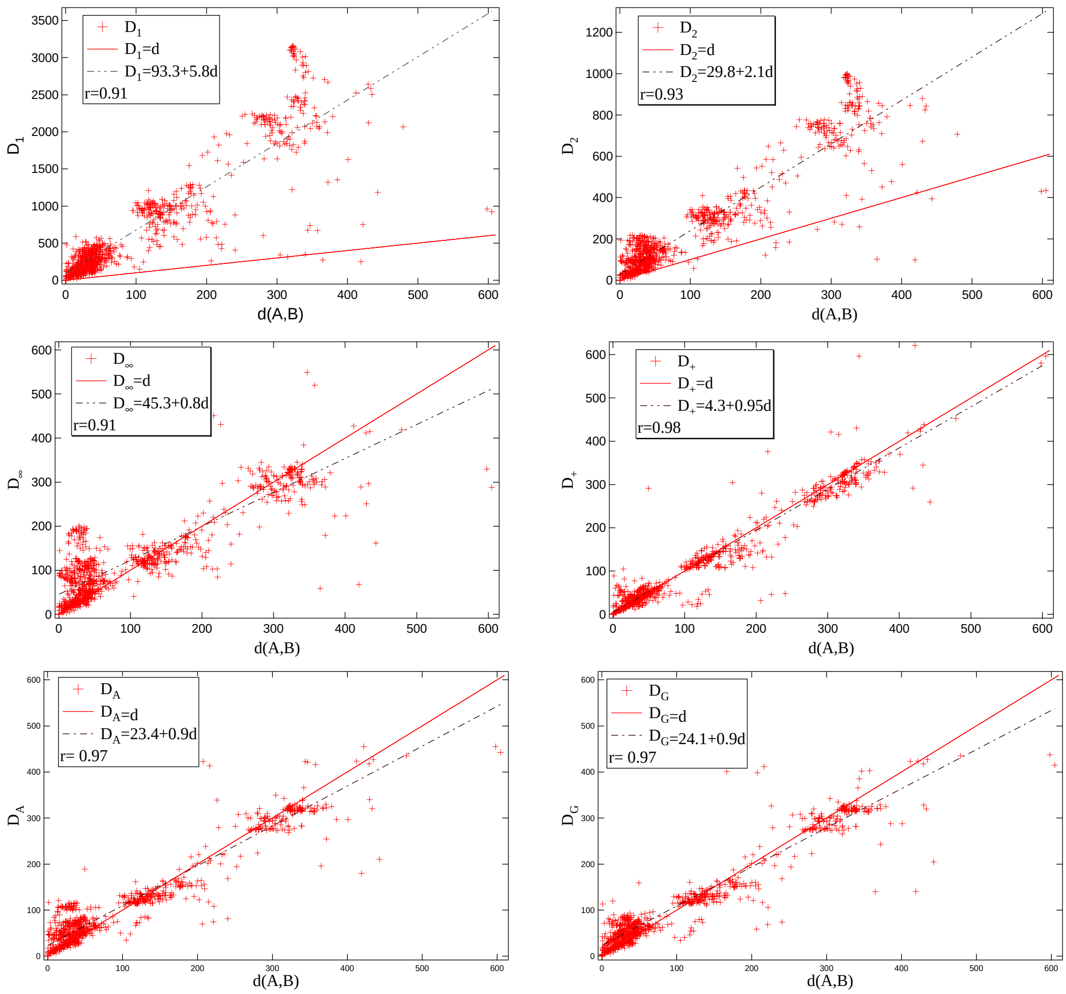

Fig. 2. Scatterplots for the delay measurements. Shown is the estimated (i.e. hyperspace) distance versus the measured distance, for six hyperspace distance functions. The number of landmarks is10. The plotted lines are the diagonal, $D = d$, and a least-squares fit.

	5 (All)	10 (All)	15 (All)	5 (Eur)	10 (Eur)	15 (Eur)
D_1	0.86, 0.91	0.91, 0.88	0.89, 0.91	0.76, 0.75	0.89, 0.84	0.64, 0.80
D_2	0.90, 0.91	0.93, 0.86	0.96, 0.90	0.78, 0.74	0.90, 0.80	0.83, 0.74
D_∞	0.93, 0.90	0.91, 0.77	0.88, 0.84	0.65, 0.70	0.70, 0.75	0.91, 0.70
D_+	0.96, 0.90	0.98, 0.93	0.99, 0.93	0.96, 0.80	0.96, 0.87	0.93, 0.84
D_A	0.95, 0.93	0.97, 0.86	0.98, 0.90	0.79, 0.77	0.81, 0.82	0.91, 0.76
D_G	0.94, 0.93	0.97, 0.90	0.98, 0.91	0.81, 0.81	0.77, 0.82	0.94, 0.83

TABLE I

Correlation coefficients for the delay data.

	5 (All)	10 (All)	15 (All)	5 (Eur)	10 (Eur)	15 (Eur)
D_1	0.23, 0.19	0.45, 0.46	0.51, 0.53	0.39, 0.40	0.45, 0.46	0.57, 0.60
D_2	0.29, 0.28	0.48, 0.47	0.54, 0.54	0.40, 0.37	0.46, 0.46	0.57, 0.60
D_∞	0.35, 0.37	0.41, 0.40	0.47, 0.45	0.40, 0.35	0.38, 0.35	0.46, 0.44
D_+	0.67, 0.65	0.71, 0.67	0.79, 0.78	0.64, 0.58	0.65, 0.60	0.70, 0.62
D_A	0.67, 0.71	0.78, 0.74	0.81, 0.80	0.66, 0.61	0.70, 0.65	0.75, 0.70
D_G	0.54, 0.58	0.71, 0.67	0.76, 0.74	0.55, 0.50	0.62, 0.58	0.68, 0.65

TABLE II

Correlation coefficients for the hopcount data.

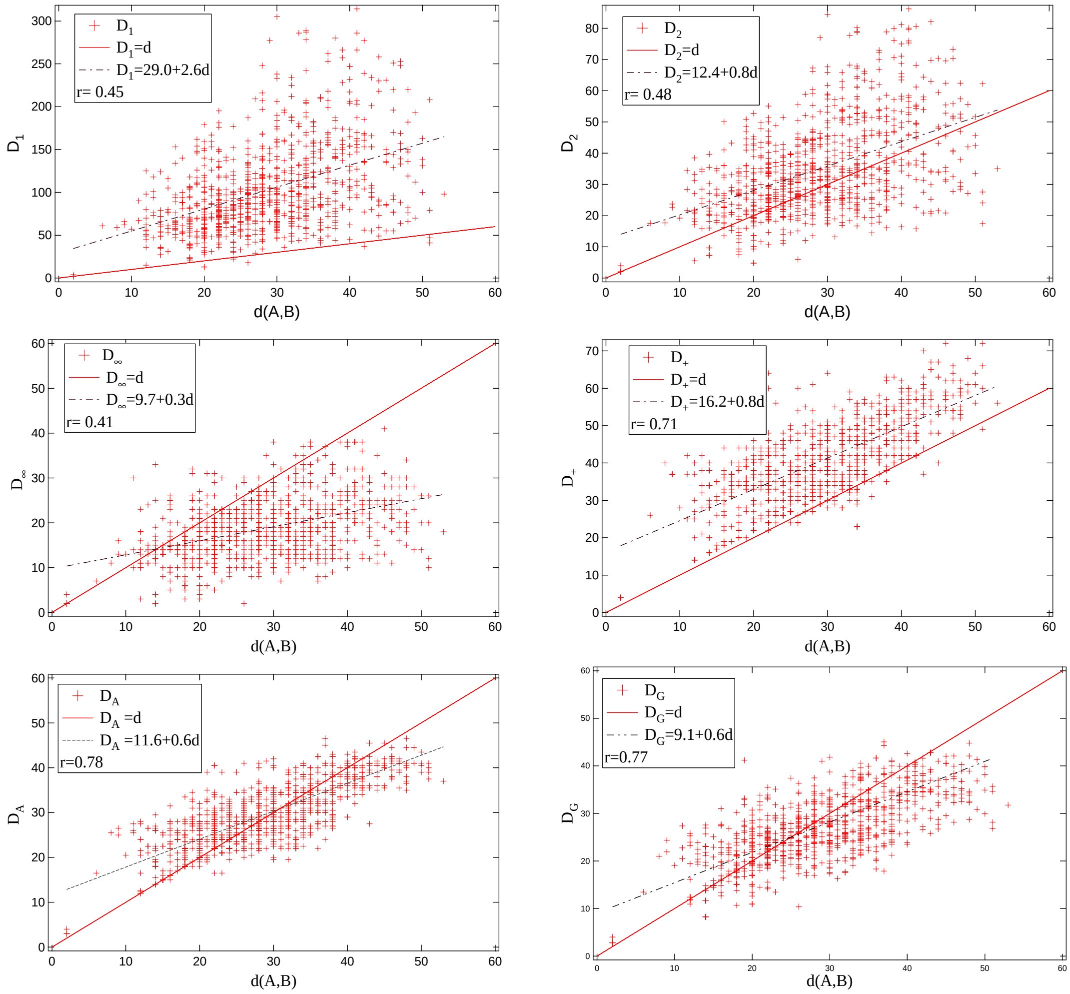

Fig. 3. Scatterplots for the hopcount. The number of landmarks is 10.

	$\alpha = 0.1$		$\alpha = 1$		$\alpha = 10$	
	hops	**weights**	**hops**	**weights**	**hops**	**weights**
D_1	0.60, 0.54	0.97, 0.98	0.13, 0.13	0.92, 0.86	-0.10, -0.10	0.39, 0.33
D_2	0.60, 0.58	0.97, 0.99	0.14, 0.14	0.93, 0.87	-0.10, -0.10	0.47, 0.39
D_∞	0.60, 0.57	0.99, 0.99	0.16, 0.16	0.92, 0.87	0.12, 0.10	0.50, 0.41
D_+	0.60, 0.60	0.97, 0.99	0.20, 0.19	0.98, 0.94	0.32, 0.31	0.65, 0.43
D_A	0.70, 0.71	0.99, 0.99	0.25, 0.24	0.98, 0.95	0.35, 0.33	0.70, 0.51
D_G	0.70, 0.71	0.99, 1.00	0.25, 0.24	0.97, 0.94	0.27, 0.32	0.70, 0.49

TABLE III

Correlation coefficients for the random graph simulations.