Dan KOMOSNY[1], Milan SIMEK[1], Ganeshan KATHIRAVELU[2]

Brno University of Technology (1), Unitec New Zealand (2)

Can Vivaldi Help in IP Geolocation?

Abstract. *The paper deals with IP geolocation based on communication latency measurement. The aim of IP geolocation is to estimate the geographical location of an IP-enabled node. Latency-based IP geolocation methods measure latency from a set of landmarks with the known geographical position to a target with an unknown position. When the latency values are known, the target position is estimated using multilateration. A disadvantage is that for each target's position estimation, a new latency measurement is required. In order to avoid this, it has been proposed to employ a latency prediction method, such as Vivaldi, to predict the latency between a target and a landmark and, thus, reduce the number of latency measurements. In this paper, we investigate this proposal in terms of location accuracy and efficiency. The conclusion of the paper gives an indicative answer about the credibility of Vivaldi for its use in IP geolocation.*

Streszczenie. *W artykule przedstawiono metodę geolokalizacji IP na podstawie pomiarów opóźnień w komunikacji. Działanie algorytmu opiera się na analizie opóźnień przesyłu sygnałów z kilku punktów orientacyjnych o znanym położeniu, do określonego celu. Na tej podstawie, z wykorzystaniem multilateracji, określana jest szukana pozycja. Zastosowano także metodę predykcji opóźnień przesyłu sygnału Vivaldi, w celu ograniczenia ilości każdorazowych estymacji. Algorytm poddano analizie pod względem precyzji i skuteczności lokalizacji. (Badanie zastosowania metody Vivaldi w geolokalizacji IP).*

Keywords: Latency, Prediction, IP, Geolocation, Multilateration, Vivaldi, SOI.
Słowa kluczowe: opóźnienie, predykcja, IP, geolokalizacja, multilateracja, Vivaldi, SOI

Introduction

Knowing one's position has always been an important part of many information systems which offer location services for the users. The position itself can be expressed by many means which have a global or local significance. The most common way to express a position on a global scale is by using geographical longitude and latitude. A local position is expressed using a relation to a specific object or objects. In this paper, we focus on finding the global position of a target.

The paper deals with geolocation of nodes connected to an IP network (for example, computer, PDA, and sensor mote). There are many methods used to obtain the geographical position of an IP node. They can be divided into native and non-native. Native methods use the current features and services provided in IP networks (for example, domain name system, structure of IP address, and communication latency). These approaches are known as *IP geolocation* methods. Non-native methods use techniques not implemented in IP networks. An example is the well-known GPS system. However, GPS requires a line-of-sight path from the target to four or more GPS satelites, which is not true for the majority of IP nodes.

IP geolocation methods can also be divided into passive and active. Passive methods obtain a node's location only by the use of the standard communication in IP networks. For example, the location of the target can be obtained by analysing the domain name of the target or by a standard query to a database which stores location information of the target (a location database). Active IP geolocation methods involve additional communication and processing of the obtained data in order to estimate a target's location. These methods are usually based on a latency measurement from a set of landmarks with the known geographical position to a target with an unknown position. The other way is also possible, a target can estimate its own location by measuring latency to a set of landmarks. When the latency values are known (to a target or to a server which locates the target), they can be transferred to a geographical distance by the use of a latency-to-distance ratio which can have a statical value [14] or it can be evaluated dynamically based on the current transmission conditions in the network [10].

The geographical distances are then used by multilateration to estimate the geographical position of the target as shown in Fig 1. The circles represent the maximum distance of the target's position from each landmark. The radius of

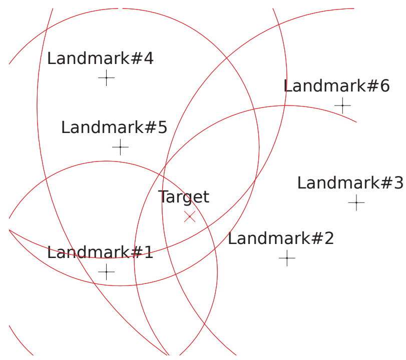

Fig. 1. IP geolocation using latency measurement and multilateration.

the circles is equal to a value derived from the measured latency between the landmark and the target after applying the latency-to-distance conversion ratio [16]. As the circles intersect with each other, they delimit the area of a possible target location. The convention is that the centre of gravity of the resulting area is used to estimate the target's location[1]. The figure shows the case when the circles intersect. Communication latency in the Internet is not constant because it is influenced by many factors, such as routing policies[2], buffering delays, and temporary peak loads. Moreover, communication latency also violates the triangle inequity[3] [28]. Therefore, some of the circles do not need to intersect at all, or, in the other case, their intersections can result in an overestimated area of a target location, which significantly decreases the location accuracy. An example of underestimation caused by non-intersecting circles (Landmarks no. 1,2 and 4) is shown in Fig 2. Fig 3 shows the opposite case where the circles specify the target location very loosely as a result of overestimation.

A disadvantage of IP geolocation based on measuring communication latency is the latency measurement. In large-scale systems, extensive latency measurements can be falsely assessed as a DoS (Denial-of-Service) attack on a target [20]. Therefore, the number of latency measurements

[1] A similar approach can be seen in wireless sensor networks where the radio signal strength RSSI (Received Signal Strength Indicator) is measured and the physical distance derived [21].

[2] An example is routing asymmetry [13].

[3] $|AB| \leq |AC| + |BC|$.

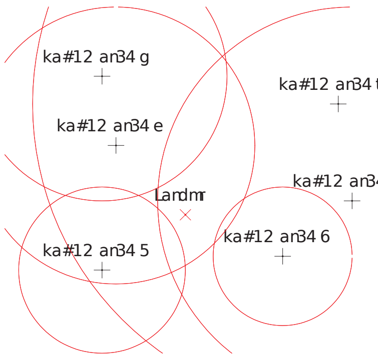

Fig. 2. IP geolocation using latency measurement and multilateration – underestimation.

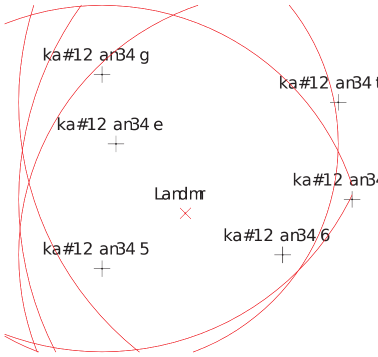

Fig. 3. IP geolocation using latency measurement and multilateration – overestimation.

should be kept low. It has been proposed in the related research that network virtual coordinate systems can be used to predict the latency between a pair of nodes without a real measurement [2, 23, 5]. In these published papers an investigation dealing with the impact of using latency prediction on IP geolocation accuracy and efficiency is missing.

In this paper, we explore the use of a network coordinate system for latency prediction in an IP geolocation system in terms of location accuracy and, also, we study the efficiency of location. For the purpose of the investigation, we implemented SOI (Speed of Internet) [28] which is an IP geolocation technique based on latency measurement and, for latency prediction, we implemented the Vivaldi virtual coordinate system [4]. Vivaldi is generally considered as the representative of network coordinate systems due to its use in real-world applications. We chose SOI since it gives results comparable to the majority of IP geolocation techniques [14].

The paper is organised as follows. Section 2 gives an overview of IP geolocation. Methods used for obtaining a geographic location of an IP node are briefly summarized. Section 3 describes related work. In this section, we focus on latency prediction algorithms working with network coordinate systems. Section 4 introduces the proposed real IP geolocation system which we developed for the purpose of our investigation. Section 5 evaluates the results from the developed system. We discuss the results in terms of location accuracy and efficiency. In section 6 we conclude the paper and give a recommendation about the use of Vivaldi for IP geolocation implemented by SOI.

IP Geolocation Applications and Accuracy

The geographic location of a target can be expressed by many means with different levels of accuracy, such as country, region, city, street or longitude and latitude. The required level of accuracy depends on the application that works with IP node positions. Country-level accuracy is usually sufficient in e-commerce where the knowledge of customer locations helps to improve the service and security. E-commerce uses the known customer's position for presenting goods, services and multimedia related to the location. For example, some international e-shops can ship particular goods only to specific countries due to a legal distribution restriction. Knowing the shopper's country, goods to be displayed can be filtered or a warning message can be displayed when a specific good not meeting the distribution restriction is being ordered. The local currency or the shipping cost estimation without providing the exact postal address can also be shown. Billing security is another example of IP geolocation use in e-commerce. With on-line credit card billing, the geographical distance between the address provided by a customer and the location of the billing transaction can be estimated. If these two places are distinct, there is a higher probability that the card is being misused. MaxMind credit card fraud detection system – minFraud [19] is a fraud detection service which offers this security measure. Also some Video on Demand (VoD) multimedia systems, such as HULU and BBC iPlayer, restrict their services to customers within a specific country [17].

City-level location accuracy is usually required for regional services, such as Internet Protocol television (IPTV). With it different multimedia streams are broadcast to receivers within a specific geographic area (for example regional news). The geographic location of a receiver can also be used for the optimisation of feedback transmission required for the RTP/RTCP protocols [26] which are used for IPTV stream distribution in IP networks [15].

The location of both city and street is used for Voice over IP (VoIP). Emergency calls are automatically directed to a local operator capable of handling them [6]. Hoax and spam calls can be detected when the caller's physical location and the reported location are distinct [27]. A caller in emergency not able to speak can also be located in some degree. In this case, the highest level of accuracy is required (i.e. at least street or block area resolution).

Currently the most used IP geolocation technique is based on querying commercial or freely available geolocation databases. The paper [25] summarises them and evaluates their accuracy. The finding is that IP geolocation databases are very accurate (96–98 %) at the country level with an error range up to 800 Km. Another passive IP geolocation technique queries domain name servers for the LOC parameter which stores the geographical position of the specific domain name server [7]. In our related research, we found that only a few domain name servers support the LOC parameter. We queried 4.5 million servers and we were able to get the location from only 181 of them. These servers were further tested for the ability to respond to an ICMP echo request to measure the latency to them. The finding was that only 143 servers allowed latency measurement [12].

The advantage of passive IP geolocation methods is their speed in obtaining the location information. The manual or automatic filling of the geographic data can lead to wrong locations or missing location records. On the other hand, active IP geolocation techniques based on latency measurements provide a better location accuracy when compared to passive methods. The accuracy of active IP geolocation techniques is compared in [9]. It varies approximately from 40 to 100 Km. The current state-of-the art techniques are accurate to the level of city or region level – Octan [30, 29] (40 Km). The disadvantage of the active IP geolocation methods is the need for an infrastructure (i.e. a set of landmarks with the known position available for running the geolocation algorithm), and the need for targets to respond to the ICMP echo messages. Also, they need a longer time to identify a target position (approximately a couple of minutes). For example, the on-line service based on the Spotter technique [18].

Related Work

The paper investigates an application of the network coordinate systems for latency prediction in IP geolocation. We first describe the related work dealing with latency prediction. Then, we focus on IP geolocation methods using the latency-to-distance conversion.

Network coordinate systems assign each node a position in the chosen space. The distance in the space is then equal to the latency. The aim is to find such coordinates in the space that predict the latency with the lowest error. There are various algorithms used for assigning the coordinates to the nodes. These algorithms typically work with the n-dimensional Euclidian or non-Euclidian (for example, cylindrical, toroidal or spherical coordinate systems) spaces. A n-dimensional Euclidian space is commonly used since it produces good results in latency prediction when compared to non-Euclidian spaces [4, 1].

Global Network Positioning (GNP) is a network coordinate system working with n-dimensional Euclidean spaces [23]. It is based on a set of landmarks that measure latency to each other. Each landmark is assigned a position (coordinates) in the chosen space using an error minimisation function. When a node joins the system, it measures the latencies to a subset of landmarks and it also obtains their coordinates. Using this information it calculates its own coordinates.

The Lighthouses algorithm has been proposed to solve a disadvantage of GNP dealing with the possible unavailability of landmarks in specific areas [24]. Also, it does not require a complex optimisation function as GNP. Instead, simple linear algebra is used. When a node identifies its coordinates it acts as a new landmark. Each node stores the coordinates of the other landmarks. This feature allows a new node to join the system contacting only one node in the system. Each node chooses the landmarks independently of the other nodes. These features make Lighthouses more scalable than GNP.

Vivaldi is probably the most widely used network coordinate system [4, 22]. Unlike the other coordinate systems, Vivaldi has also been implemented in various non-academic applications, such as Azureus or The Bamboo Distributed Hash Table. It does not require any network infrastructure with the landmarks and it is fully decentralised. Due to its scalability, it is suitable for large-scale systems, such us IP geolocation. Vivaldi works with a network of springs. Each spring is placed between a pair of nodes. The current length of the spring between two nodes is equal to the distance between them in the chosen space. The rest length of the spring is equal to the latency between the nodes. Vivaldi changes the length of the springs in a way to minimise the sum of the potential energies of all the springs. In this way, the rest length of the springs is found and, consequently, the latencies between the nodes are known. A possible way how to express the position error (i.e. error of latency prediction) E in the coordinate system[4] is [4]:

$$E = \sum_i \sum_j (L_{ij} - ||\boldsymbol{x}_i - \boldsymbol{x}_j||)^2, \quad (1)$$

where L_{ij} is latency between nodes i and j. $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are the coordinates of nodes i and j respectively. $||\boldsymbol{x}_i - \boldsymbol{x}_j||$ is the distance between nodes i and j in the coordinate system.

The change of the spring lengths is caused by the movement of the nodes in the chosen space. Vivaldi uses time steps for the node's movement. In each time step, all the nodes move in the direction of the spring force placed between the node and its pair. Each step should result in more precise coordinates in the chosen space, and, therefore, better latency prediction. The time steps are repeated until the nodes' coordinates converge to the values that predict latency with the required accuracy.

Vivaldi faces a problem that a local minimum can be found instead of the global one and, consequently, the required prediction accuracy is never reached. In order to eliminate this disadvantage and, also, to reduce the convergence time for obtaining the correct coordinates, Vivaldi uses an adaptive time step for moving the nodes in the chosen space. When a node is in the stage of finding its rough coordinates, a larger time step is applied and the node moves a greater distance. Later, a smaller time step is applied to find more precise coordinates.

A pseudocode describing the Vivaldi algorithm with the adaptive time step is shown in Listing 1 [4]. The entry parameters are the measured latency L_{ij} from node i to node j, the coordinates of node j $\boldsymbol{x_j}$ and its estimated position error e_j. The constants c_e and c_c are used for tuning the Vivaldi algorithm.

\# calculation of the weight using the estimated location error of local node i and remote node j
$w = \frac{e_i}{e_i + e_j}$
\# calculation of the relative error between the latency measured to remote node j and the distance between nodes i and j
$e_s = \frac{| ||\boldsymbol{x_i} - \boldsymbol{x_j}|| - L_{ij} |}{L_{ij}}$
\# update the weighted moving average of the local error
$e_i = e_s \times c_e \times w + e_i \times (1 - c_e \times w)$
\# update the location of node i
$\delta = c_c \times w$
$\boldsymbol{x_i} = \boldsymbol{x_i} + \delta \times (L_{ij} - ||\boldsymbol{x_i} - \boldsymbol{x_j}||) \times \vec{u}(\boldsymbol{x_i} - \boldsymbol{x_j})$

Listing 1. Decentralized Vivaldi with adaptive time step.

First the weight is calculated based on the estimated position error of the local and remote node. Then the local relative error e_i is identified using the weighted moving average. Next, node i moves to its new coordinates $\boldsymbol{x_i}$. The movement distance is set by the difference between the measured latency and the distance between the nodes in the coordinate system $(L_{ij} - ||\boldsymbol{x_i} - \boldsymbol{x_j}||)$ reduced by the calculated value of the adaptive time step δ. Unit vector $\vec{u}(\boldsymbol{x_i} - \boldsymbol{x_j})$ sets the direction of the movement. This algorithm runs on each node in the system.

There are also latency prediction systems that do not use network coordinates. IDMaps [8] and Internet Iso-bar [3] divide the Internet into distinct areas. These systems measure the latency among all the dedicated nodes in the different areas. The latency between any two nodes in the distinct areas is then estimated using the known latency values for

[4] Vivaldi does not require a specific coordinate system.

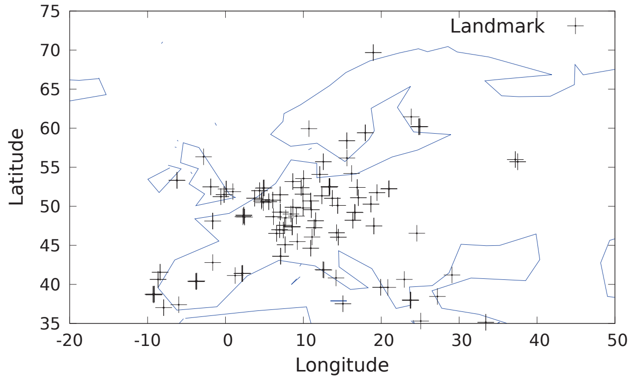

Fig. 4. Location of used PlanetLab landmarks.

the dedicated nodes.

The King method uses a similar methodology [11]. It is based on the idea that each node in the Internet is topologically-close to one or more domain name servers. The latency between any two nodes is approximated by the latency between their close domain name servers. The latency between the domain servers is estimated using the standard recursive domain name queries.

A method for latency-based geolocation that combines distance constraints from a set of landmarks to find a target position is SOI (Speed of Internet) [14]. SOI uses a static latency-to-distance conversion. The value used is 9/4 the speed of light in a vacuum. This constant was derived by using the speed of light in optical cables and networking-related factors, such as data processing, circuitous paths, and buffering delays.

Constraint-Based Geolocation (CBG) is similar to SOI, but it uses a different method for deriving the latency-to-distance ratio [10]. Beside latency measurements between the target and landmarks, it also measures the latency among all the landmarks. Based on the geographical location of the landmarks, CBG defines the latency-to-distance ratio for each landmark using the so called 'best line' as the line that is placed closely below all the measured latencies to the other landmarks. CBG performance in terms of accuracy is similar to SOI [14].

Developed IP Geolocation System

For the purpose of IP geolocation accuracy and efficiency evaluation, when a network coordinate system is used, we developed an IP geolocation system for locating nodes in Europe. Our system is based on the global research network PlanetLab[5]. For location, we used a set of landmarks and a set of targets, both located in Europe. Our landmarks' set consists solely of the PlanetLab nodes. We used the PlanetLab sites location information to assign the location to each landmark. The targets' set consists of the nodes for which we gained the location using personal communication and of the domain name servers supporting the DNS LOC parameter which stores the geographical position of the server.

We verified the location for each on the node used. For the landmarks' set, we compared the location to the delivery address in the PlanetLab site contact field. We considered the nodes where we found a close match between these two items of information. Also, we omitted locations pointing to unlikely places such as in deserted places and the sea. From the total number of 306 PlanetLab nodes at 152 sites in Europe, we successfully verified 244 landmarks. From this

[5]http://www.planet-lab.eu/

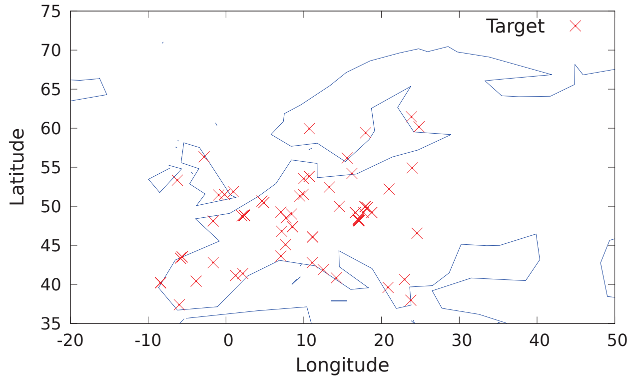

Fig. 5. Location of used targets.

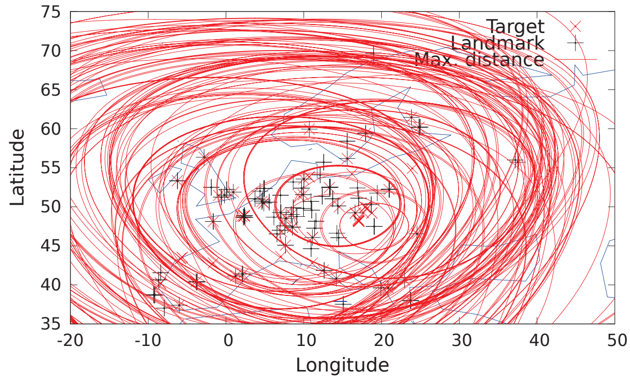

Fig. 6. Great circles around each landmark.

set, a number of the nodes were off-line when we performed the measurement. The landmarks successfully used for location[6] are shown in Fig 4. Our targets' set consisted of 167 targets. We verified the position obtained from the DNS LOC parameter against the country and other codes from the domain names (for example, cities) and we also omitted locations pointing to deserted places. After the verification of the nodes we got 103 targets. The nodes from this set which were accessible during measurement are shown in Fig 5.

We implemented the SOI algorithm and estimated the location of the targets. We spread the measurement across 24 hours to minimize the load on the targets. A sample estimation for a node located in the Slovak Republic is shown in Fig 6. The figure shows the great circles around each used landmark. The radius of each circle was derived from the latency-to-distance calculation. The circles shown do not have a circular shape because we used the World Geodetic System 84 standard to obtain precise distances taking into account the shape of the Earth. Using this approach, we assured a better accuracy than other implementations working with the intersection area of the great circles projected in Cartesian coordinates. The estimated area of the target location given by the intersection of the great circles is shown in Fig 7. We used the centroid of the result polygon as the target's coordinates.

Next, we estimated the latencies using our Vivaldi implementation. As Vivaldi can work with various spaces, we chose the two most accurate ones – 2D Euclidean space and 2D+height space. Other spaces (3D Euclidean or spherical) are also possible, but they give a worse accuracy [4]. 2D+height space extends 2D Euclidian space by adding a new height dimension. The purpose of the height dimension is to project the latency in access networks whereas 2D

[6]At least one successful latency measurement.

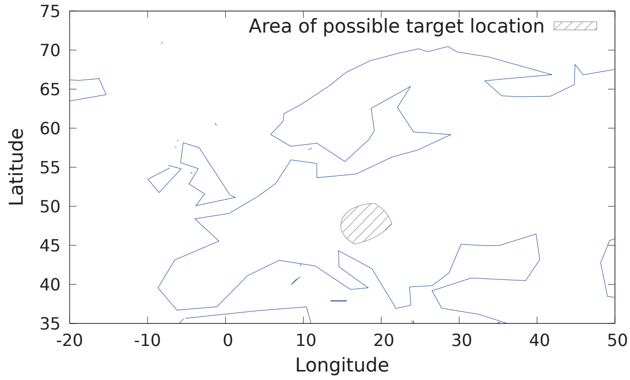

Fig. 7. Estimated area of target location as product of intersections of the great circles.

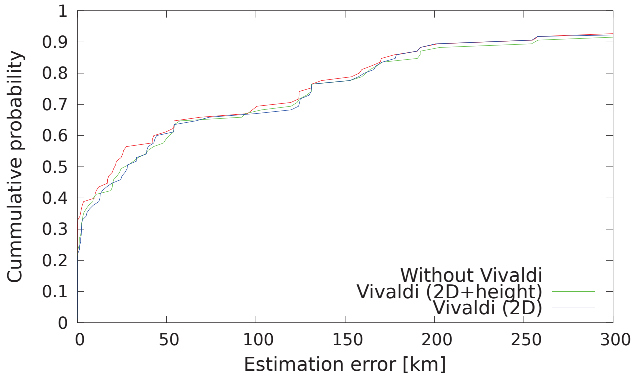

Fig. 8. Estimated location error.

space projects latency in transport networks. Each node has a positive height value which is in contrast with 3D Euclidean space, where the third dimension can have both positive and negative values. The modified equations for 2D+height space are [4]:

$$[x, x_{\mathrm{h}}] - [y, y_{\mathrm{h}}] = [(x - y), x_{\mathrm{h}} + y_{\mathrm{h}}], \tag{2}$$

$$||[x, x_{\mathrm{h}}]|| = ||x|| + x_{\mathrm{h}}, \tag{3}$$

$$\alpha \times [x, x_{\mathrm{h}}] = [\alpha x, \alpha x_{\mathrm{h}}], \tag{4}$$

where $x_{\mathrm{h}}, y_{\mathrm{h}}$ are heights. For 2D+height system the equation expressing the position error in Vivaldi (1) is modified as [4]:

$$E = \sum_i \sum_j (L_{ij} - ||\boldsymbol{x_i} - \boldsymbol{x_j}|| + h_i + h_j)^2. \tag{5}$$

Evaluation and Results Discussion

In this section, we evaluate the results from the IP geolocation system described. First, we focused on location accuracy of SOI without Vivaldi and with Vivaldi being used. We compared the results for two coordinate systems, 2D and 2D+height, as they give the lowest latency prediction error. Fig 8 shows the cumulative distribution function (CDF) for the location relative error, i.e. the difference between the true and estimated location of the targets. It can be seen from the figure, there is no significant difference between SOI without and with Vivaldi used. The second aspect we focused on was the location confidence, i.e. the size of the area where a target could be located. Fig 9 shows the CDF function of the size of the estimated area. We explain the slight difference when SOI with Vivaldi outperforms SOI without Vivaldi by possible underprediction of the latencies (some of the predicted latencies were smaller than the real ones) which, subsequently, resulted in smaller estimated areas. We can conclude that using Vivaldi for latency prediction in IP geolocation does not result in any significant accuracy decrease.

The previous location estimation was done when the system was stable, i.e. Vivaldi found the global minimum of the spring energy function. We also focused on the applicability of Vivaldi considering its stability. We located the targets in a certain number of Vivaldi's working stages before it converted to the right coordinates. We faced a problem that stopping Vivaldi at a specific time did not give the same maximum relative latency prediction error. This is caused by the fact

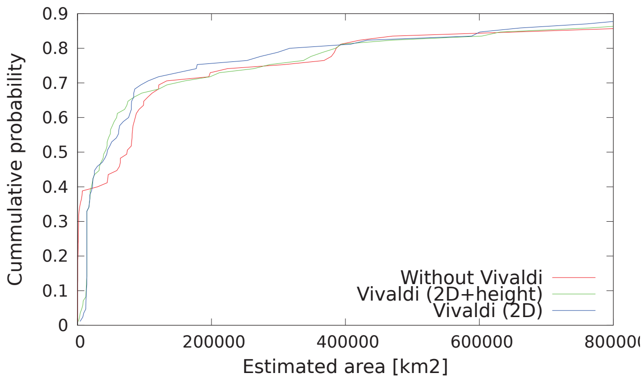

Fig. 9. Estimated area of possible location.

that Vivaldi's performance depends on the current network transmission conditions. We also noticed that using ssh connections to the PlanetLab nodes influences Vivaldi's convergence time a lot (seconds). Considering this issue, instead of the unstable running time of Vivaldi we used the maximum relative latency prediction error which we were monitoring during Vivaldi's convergence. When Vivaldi reached this value, we evaluated the results, which are shown in Fig 10 and Fig 11. Concerning the location estimation error, there are significant differences when Vivaldi is unstable. With a lower prediction error, the location accuracy increases. A similar trend can be seen in the graph (Fig 11) with the size of estimated areas.

We also investigated the location efficiency, i.e. how the used of Vivaldi affects the number underestimations (the number of cases when IP geolocation failed). Table 1 shows the number of underestimations for each maximum relative prediction error when Vivaldi is unstable. As can be seen, the number of underestimations significantly decreases with the lower latency prediction error. We found that with a prediction

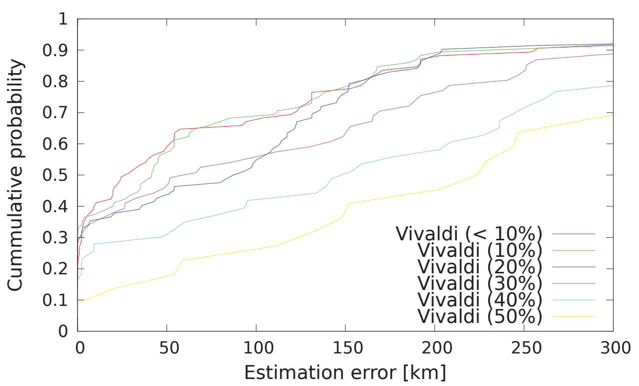

Fig. 10. Estimated location error with unstable Vivaldi.

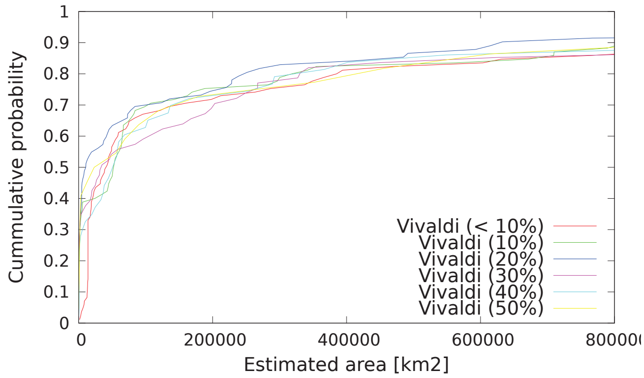

Fig. 11. Estimated area of possible location with unstable Vivaldi.

Vivaldi prediction error	SOI underestimations
50 %	65
40 %	44
30 %	26
20 %	3
10 %	0

Table 1. Number of SOI underestimations with unstable Vivaldi.

error of 10 % our implementation of SOI does not underestimate any target. Therefore, we consider Vivaldi stable for SOI when its relative prediction error is less than 10 %. Taking into account Vivaldi's equation for the latency prediction error

$$e_s = \frac{|||\boldsymbol{x_i} - \boldsymbol{x_j}|| - L_{ij}|}{L_{ij}}, \quad (6)$$

we can conclude that e_s should be less than 10 % for the reliable use of Vivaldi for SOI.

Conclusion

In the paper, we investigated the idea of using a network coordinate system for latency prediction in IP geolocation. For this purpose we implemented an IP geolocation system for locating nodes in Europe. Our system is based mainly on the PlanetLab nodes which acted as the landmarks with the known location. The other set of nodes used, targets, mainly consisted of the DNS servers with the known location. We found the location of the targets using requests to the DNS LOC parameter.

We found that using Vivaldi in IP geolocation (SOI) is possible. First, we investigated Vivaldi in its stable state, i.e. when all the nodes converged to the right coordinates. The results showed that there is no significant difference between the location accuracy with or without Vivaldi being used. Next we addressed the fact that Vivaldi converts to its stable state and may produce wrong latency estimations. We studied how the instability of Vivaldi effected the location accuracy and efficiency. We identified the threshold of prediction error equal or less to 10 % to produce zero underestimations in IP geolocation. Below this threshold, we consider Vivaldi to be reliable to use in IP geolocation with latency-to-distance ratio equal to 9/4 speed of light.

We have also started an investigation of using Vivaldi for location sensor networks' nodes. In sensor networks, several techniques are used for this purpose such as measuring the received signal strength (RSS) or time-of-arrival (TOA). We are focusing on the prediction of these measured values using Vivaldi in order to minimize the number of physical measurements. This could also have a positive effect of lower energy consumption, which is one of the key requirements for a longer sensor networks' operational time.

Acknowledgement: The described research was performed in the laboratories supported by the SIX project; registration number CZ.1.05/2.1.00/03.0072, the operational program Research and Development for Innovation.

REFERENCES

[1] Bruno Abrahao and Robert Kleinberg. On the internet delay space dimensionality. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, IMC '08, pages 157–168, New York, NY, USA, 2008. ACM.

[2] Sharad Agarwal and Jacob R. Lorch. Matchmaking for online games and other latency-sensitive p2p systems. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, SIGCOMM '09, pages 315–326, New York, NY, USA, 2009. ACM.

[3] Yan Chen, Yan Chen, Y H. Katz, and Chris Overton. Internet Iso-bar: A Scalable Overlay Distance Monitoring. *Journal of Computer Resource Management*, 2002.

[4] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris. Vivaldi: A decentralized network coordinate system. In *In SIGCOMM*, pages 15–26, 2004.

[5] Benoit Donnet, Bamba Gueye, and Mohamed Ali Kaafar. A survey on network coordinates systems, design, and security. *IEEE Communication Surveys & Tutorials*, 12(4), October 2010.

[6] Federal Communications Commission. Interconnected VoIP Service; Wireless E911 Location Accuracy Requirements; E911 Requirements for IP-Enabled Service Providers. FR Doc No: 2011-24865, 2011.

[7] T. Fioreze and G. Heijenk. Extending DNS to support geocasting towards VANETs: A proposal. In *Vehicular Networking Conference (VNC)*, pages 271–277. IEEE, 2010.

[8] Paul Francis, Sugih Jamin, Cheng Jin, Yixin Jin, Danny Raz, Yuval Shavitt, and Lixia Zhang. IDMaps: a global internet host distance estimation service. *IEEE/ACM Trans. Netw.*, 9(5):525–540, 2001.

[9] Phillipa Gill, Yashar Ganjali, Bernard Wong, and David Lie. Dude, where's that IP? Circumventing measurement-based IP geolocation. In *Proceedings of the 19th USENIX Security Symposium*, August 2010.

[10] Bamba Gueye, Artur Ziviani, Mark Crovella, and Serge Fdida. Constraint-based geolocation of internet hosts. *IEEE/ACM Trans. Netw.*, 14:1219–1232, December 2006.

[11] Krishna P. Gummadi, Stefan Saroiu, and Steven D. Gribble. King: estimating latency between arbitrary internet end hosts. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, IMW '02, pages 5–18, New York, NY, USA, 2002. ACM.

[12] P. Harth. Geographical location estimation of Internet nodes using communication latency measurement. Master's thesis, Brno University of Technology, 2011.

[13] Yihua He, M. Faloutsos, S. Krishnamurthy, and B. Huffaker. On routing asymmetry in the Internet. In *Global Telecommunications Conference*, volume 2. IEEE, 2005.

[14] Ethan Katz-Bassett, John P. John, Arvind Krishnamurthy, David Wetherall, Thomas Anderson, and Yatin Chawathe. Towards IP geolocation using delay and topology measurements. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, IMC '06, pages 71–84, New York, NY, USA, 2006. ACM.

[15] D. Komosny, P. Moravek, and G. Burget, R.and Kathiravelu. Feedback transmission in large-scale IPTV sessions. In *Proceedings of the 9th international conference on Communications and information technologies*, ISCIT'09, pages 796–801, Piscataway, NJ, USA, 2009. IEEE Press.

[16] O. Krajsa and L. Fojtova. RTT measurement and its dependence on the real geographical distance. In *Proceedings of 34th International Conference on Telecommunications and Signal Processing*. Brno University of Technology, 2011.

[17] Dilip Kumar Krishnappa, Samamon Khemmarat, Lixin Gao, and Michael Zink. On the feasibility of prefetching and caching for online TV services: a measurement study on hulu. In *Proceedings of the 12th international conference on Passive and active measurement*, PAM'11, pages 72–80, Berlin, Heidelberg, 2011. Springer-Verlag.

[18] S. Laki, P. Matray, P. Haga, T. Sebok, I. Csabai, and G. Vattay. Spotter: A model based active geolocation service. In *INFOCOM*, pages 3173–3181. IEEE, 2011.

[19] MaxMind. Fraud detection through IP address reputation and a mutual collaboration network. White paper, 2011.

[20] J. Mirkovic, S. Fahmy, P. Reiher, R. Thomas, A. Hussain, S. Schwab, and C. Ko. Measuring impact of DoS attacks. In *Proceedings of the DETER Community Workshop on Cyber Security Experimentation*, 2006.

[21] P. Moravek, D. Komosny, M. Simek, D. Girbau, and A. Lazaro. Energy analysis of received signal strength localization in wireless sensor networks. *Radioengineering*, 10(4):937–945, 2011.

[22] P. Moravek, D. Komosny, M. Simek, J. Sveda, and T. Handl. Vivaldi and other localization methods. In *In Proceedings of the International Conference on Telecommunications and Signal Processing (TSP).* Asszisztencia Szervező Kft., 2009.

[23] T.S.E. Ng and Hui Zhang. Predicting Internet network distance with coordinates-based approaches. In *Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*, volume 1, pages 170–179. IEEE, 2002.

[24] M. Pias, Crowcroft J., S. Wilbur, T. Harris, and S. Bhatti. Lighthouses for Scalable Distributed Location. In *Peer-to-Peer Systems II*, volume 2735, pages 278–291, 2003.

[25] I. Poese, S. Uhlig, M.i Kaafar, B. Donnet, and B. Gueye. IP geolocation databases: unreliable? *ACM SIGCOMM Computer Communication Review*, 41:53–56, apr 2011.

[26] M. Voznak, M. Halas, B. Borowik, and Z Kocur. Delay model of RTP flows in accordance with M/D/1 and M/D/2 kendall's notation. *International journal of mathematics and computers in simulation*, 5, 2011.

[27] M. Voznak and F. Rezac. The implementation of SPAM over Internet telephony and a defence against this attack. In *Proceedings of the 32nd International Conference TSP*, TSP 09. Asszisztencia Szervező Kft., 2009.

[28] Guohui Wang, Bo Zhang, and T. S. Eugene Ng. Towards network triangle inequality violation aware distributed systems. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, pages 175–188, New York, NY, USA, 2007. ACM.

[29] Yong Wang, Daniel Burgener, Marcel Flores, Aleksandar Kuzmanovic, and Cheng Huang. Towards street-level client-independent IP geolocation. In *Proceedings of the 8th USENIX conference on Networked systems design and implementation*, NSDI'11, pages 27–27, Berkeley, CA, USA, 2011. USENIX Association.

[30] B. Wong, I. Stoyanov, and E. Sirer. Octant: a comprehensive framework for the geolocalization of internet hosts. In *Proceedings of the 4th USENIX conference on Networked systems design & implementation*, NSDI'07, pages 23–23. USENIX Association, 2007.

Authors: *Dan Komosny, Milan Simek, Department of Telecommunications, Brno University of Technology, Purkynova 118, 612 00 BRNO, Czech Republic, email: komosny@feec.vutbr.cz, simek@feec.vutbr.cz,*
Ganeshan KATHIRAVELU, Unitec New Zealand, Private Bag 92025, Auckland, New Zealand, email: kganeshan@unitec.ac.nz.