Improving the quality of interdomain paths by using IP tunnels and the DNS

Olivier Bonaventure*, Cédric de Launois*, Bruno Quoitin* and Marcelo Yannuzzi†

* CSE Dept, Université catholique de Louvain (UCL), Belgium
{bonaventure,delaunois,quoitin}@info.ucl.ac.be
†Universitat Politècnica de Catalunya, Barcelona, Catalunya, Spain
yannuzzi@ac.upc.edu

***Abstract*— We propose an incrementally deployable extension to the current Internet architecture that provides a more accurate selection of the interdomain paths without requiring any change to the BGP messages. Our architecture relies on a few basic principles. First, each border router computes its coordinates by using a network coordinate system. Second, we use the DNS to store information about the border routers that are able to reach each prefix as well as their coordinates. Third, BGP routers use this DNS information to establish IP tunnels to reach important destination prefixes by using the best path towards this prefix. As an example, we show by using real measurements that our architecture allows multihomed stub ASes to reduce the delay of their interdomain paths.**

I. Introduction

In today's Internet, interdomain traffic is experiencing a growing demand for highly efficient and cost effective mechanisms to improve its end-to-end performance [9], [8]. To accomplish this, a common practice among stub Autonomous Systems (ASs) is to use multiple providers [1]. This practice known as multihoming offers several benefits to these ASs, especially from the resiliency viewpoint [2]. For instance, stubs that connect to multiple providers expect a larger path diversity. Furthermore, they would like that the paths with the best quality be used to send and receive traffic. Various quality metrics can be used depending on the applications: low delay, high bandwidth, low jitter, low loss rate, etc. However, BGP was designed to provide reachability and to allow domains to locally apply route selection policies. BGP does not currently carry QoS metrics and BGP routers do not always select the paths with the best "quality" [17].

To improve the quality of the interdomain paths, we propose an incremental change to the Internet architecture. Our architecture uses interdomain tunnels that are established based on information about the BGP routers. This information is distributed by using the DNS. The main advantage of our architecture is that a few ASes can start to use it without any cooperation with the transit providers since it does not require any change to the BGP messages.

This paper is organized as follows. In section II we discuss the factors that affect the quality of the paths selected by BGP. We then propose our incrementally deployable architecture in section III. Finally, we use measurement-driven simulations in section IV to show the benefits of our architecture. Finally, we review the related work in section V.

This work was partially supported by the Walloon Government under the TOTEM project (http://totem.info.ucl.ac.be), by a grant from FRIA (Fonds pour la Formation à la recherche dans l'Industrie et dans l'Agriculture, Belgium), by the MCyT (Spanish Ministry of Science and Technology) under contract FEDER-TIC2002-04344-C02-02, and by the E-Next network of Excellence (http://www.ist-e-next.net.

II. Motivation

As discussed above, multihoming does not always lead to improved performance as the BGP decision process does not take any QoS metric into account. In order to evaluate the importance of this problem, we performed a simulation study of the delays along the paths between multihomed sites. The simulation is based on real delay measurements made during May 2004 between 58 active test boxes from the RIPE NCC Test Traffic Measurements Service [13]. The test boxes are scattered over Europe and a few are located in the US, Australia, New Zealand and Japan. Each test box is equipped with a GPS clock so that one-way delays between each pair of boxes can be measured accurately (within 10μs). More than 2000 probes are performed per day and per test box pair. The interval between two consecutive probes is randomized according to a Poisson distribution, as recommended in [5].

To simulate multihoming, we follow a methodology similar to the one used in [3], [4]. We select a few RIPE nodes in the same metropolitan area, and consider them as the border routers of a single virtual multihomed network. This method actually models multihoming where the provider-dependent prefixes advertised by the virtual site are aggregated by its providers. A total of 13 multihomed sites are emulated by this method, a number similar to the study of Akella et al. on multihoming [4]. In our study, 10 sites are dual-homed, 1 is 3-homed, 1 is 4-homed and a last one has 8 providers. One multihomed site is located in the US, one in Japan, and the others in Europe.

Figure 1 shows an analysis of delays between the border routers of the 13 multihomed sites. The figure shows, for each pair of multihomed sites, the range of delays on the available paths. On the x-axis, we show the pairs of dual-homed stubs in decreasing order of their best delay. On the y-axis, we show the median delay on the available paths for the corresponding pairs of stubs, as well as the lowest and highest delays. We observe that for many site-site pairs, there are large variations in the measured delays. Differences larger than 100ms between the best and worst delays are frequent. Due to the performance-blind selection of paths performed by BGP, the worst path could be selected, leading to a delay that can sometimes be tremendously larger than the delay of the best available paths.

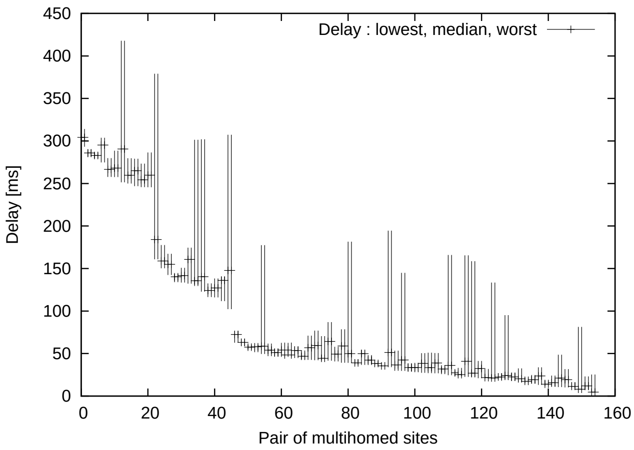

Fig. 1. Delays on the available paths between 13 multihomed sites.

In addition to this, a lot of interdomain paths are hidden to the multihomed sites, decreasing the freedom of choosing alternative paths such as lower delay paths. This is due to BGP decision process where each router distributes only a single best route towards each prefix. Therefore, single-homed stubs receive a single route towards any destination while dual-homed stubs receive at most two routes.

One way to solve this problem in the case of multihomed stubs is to leverage the diversity of Internet paths by relying on the routes towards the prefixes of the providers of the destination domain. Figure 2 shows a simple example of two stubs: *AS2* is a single homed stub that uses *P5* to access the Internet while *AS1* is a multihomed stub connected through providers *P1*, *P2* and *P3*. With BGP, *AS2* will only know a single path towards *AS1*: the path through *P5* and *P2*. However, if we look at the routes known by *AS2* to reach the providers of *AS1*, we observe that there are 2 alternative paths. The first one goes through *P5* and *P1* and the other one through *P5*, *P2* and *P3*. The number of possible paths towards the destination domain is equal to the number of providers of *AS2* times the number of providers of *AS1*.

In order to show the potential benefit of exploiting the routes towards the providers of the destination domain, we performed a simulation based on real BGP routing tables collected by the RouteViews project [23]. The study was performed on a routing table collected on December 1st, 2004. The routing table contained 5750380 routes received from 34 different peers. In the simulation, we only considered the 32 peers that announced a full routing table, i.e. more than 140.000 routes.

Among all the received routes, we identified, based on the AS-paths, 6402 multihomed stubs. These multihomed stubs originated 29575 different prefixes. We then considered all the 496 pairs of RouteViews peers. For each pair of peers, we simulated a dual-homed stub domain connected to the peers. For each simulated stub, we counted the number of different paths learned through BGP towards all the considered destination prefixes. We consider that two paths are different if at least the provider in the source AS or the provider in

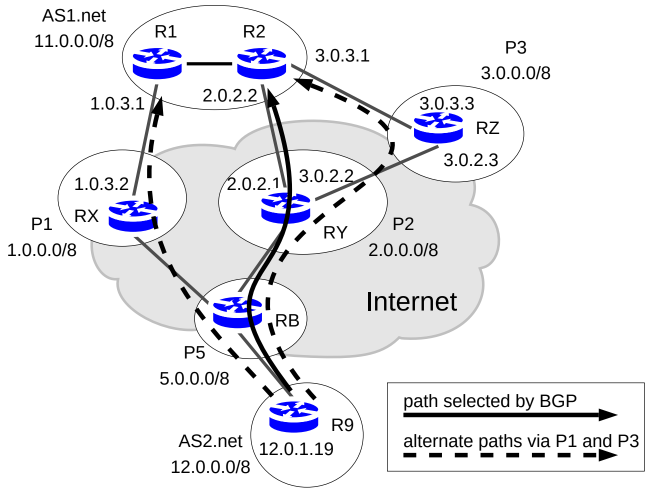

Fig. 2. Using the BGP routes towards the providers of the destination domain, in order to increase path diversity.

the destination AS are different. Note that if two paths are different, that does not mean that they are completely disjoint.

We show the results of our simulations in figure 3. The figure shows the distribution of the number of different paths available with BGP towards the destination domain and towards the providers of the destination AS, for all the destination prefixes. On the x-axis, we show the number of different paths available and on the y-axis, the number of prefixes that could be reached with the corresponding number of paths. The number of available paths is an average over the 496 simulated dual-homed stubs. We do not show the variance since it is very low.

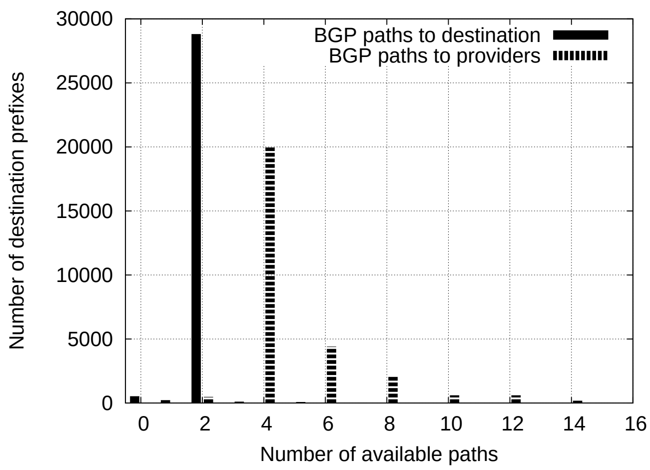

Fig. 3. Path diversity when multihoming to RouteViews peers.

When looking at the BGP paths towards the destination AS, the number of distinct paths is comprised between 0 and 2. If there is no path, that means that the destination prefixes cannot be reached. This fortunately occurs for only a small subset of the RouteViews dataset. This is probably due to the filters used by some ISPs. If there is only one path, that means that the destination prefix was not reachable through one of the providers. But most of the time, the destination prefixes were reachable through both providers. The number of available BGP paths cannot be more than 2 since the simulated stubs only receive one route for each destination prefix from each provider. Moreover, it is frequent that these paths merge at the same provider of the destination AS. The path diversity is thus low with BGP even if there are two different paths most of the time.

If we look at the routes towards the providers of the destination AS, the path diversity increases a lot. Most destination prefixes (67 %) are reachable through at least 4 different paths. There is also a significant number of destination ASes (30 %) that are reachable through more than 4 paths due to some destination stubs being more than dual-homed. The reason for the large majority of the destination prefixes having an even number of different paths is that the source stub is dual-homed. The simulations show that using the routes towards the providers of the destination domain brings out a lot of new paths.

III. A NEW INTERDOMAIN ARCHITECTURE

From the simulations described in the previous section, we know that BGP suffers from two drawbacks from a quality of service viewpoint. First, a BGP router advertises a single path towards each destination. Second, there is no QoS metric inside the BGP advertisements. Although several BGP extensions have been proposed to address those problems [31], [10], [29], deploying them on the global Internet would be difficult.

To avoid changing anything to the BGP messages, we first note that when a stub AS is connected to a provider, the IP addresses used on the stub-provider link usually belong to the provider's prefix. For example, in figure 2 the IP address of router $R1$ on the link with provider $P1$ is `1.0.3.1` and belongs to $P1$'s CIDR block. With this in mind, we note that to reach $AS1$, there are three possible entry routers : `1.0.3.1`, `2.0.2.2` and `3.0.3.1`. In this figure, $AS1$ advertises its own IP prefix (`11.0.0.0/8`) to its providers. Each distant router will select the best path to reach `11.0.0.0/8` among the BGP paths learned from its peers. For example, in figure 2, $AS2$ would learn path `P5:P2:AS1`. From its BGP routing table for prefix `11.0.0.0/8`, router $R9$ does not know that there are alternate paths via $P3$ and $P1$. However, as shown in [17], the length of the BGP `AS Path` is not always a good indication of the quality of an interdomain path. Thus, the paths via $P1$ or $P3$ may have a lower delay than the path selected by BGP. It should be noted that although the BGP table of router RB does not provide a path to reach $AS1$ via $P3$ and $P1$, it contains at least one path to reach the prefix that belongs to those providers. For example, in figure 2, packets sent by $AS2$ to reach address `3.0.0.0` will follow the `P5:P2:P3` path. To use this path to reach $AS1$, router $R9$ could encapsulate its packets inside a GRE, IP-in-IP or IPSec tunnel with destination `3.0.3.1`.

To be able to establish the required interdomain tunnel, a BGP router in the source AS must determine the IP addresses of the entry border routers in the destination AS. A first solution to distribute those IP addresses would be to rely on BGP and add to the BGP advertisements sent by a source AS, a list of extended communities [25] containing the IP addresses of the candidate tunnel endpoints in the source AS. This could be done by defining a new type of extended communities. Unfortunately, not all BGP routers in the Internet support this BGP attribute and furthermore some transit ASes strip this attribute when distributing BGP advertisements. Thus, instead of changing BGP, we propose to distribute the information about the tunnel endpoints by using the DNS. The main advantage of the DNS is that thanks to its distributed nature and the extensible format of the DNS resource records, it is easy to add new attributes to the DNS and to deploy them incrementally. Furthermore, ISPs are now starting to deploy secure extensions to the DNS [14].

We use the reverse DNS and add a type of resource records (RR) : the *TUNNEL* DNS RR. There is one type of tunnel RR for each supported type of IP tunnel. A tunnel RR for an IP-in-IP tunnel will contain the name of a border router that can act as a tunnel tail-end in the destination AS. In the case of GRE or IPSec tunnels, additional parameters would be placed inside the tunnel RR. Several tunnel RR can be associated with each IP prefix in the reverse DNS. For example, in $AS1$'s DNS server, three tunnel RR (`r1.as1.net`, `r2a.as1.net` and `r2b.as1.net`) would be associated to `0.0.0.11.in-addr.arpa`. In addition to the tunnel RR, we propose to add in the DNS an Address Prefix List (APL) RR as defined in [19]. This APL RR is used to indicate the prefixes that can be reached via the tunnel endpoints indicated in the tunnel RR. Figure 4 shows a sample configuration of $AS1$'s DNS server.

The source AS, $AS2$ in figure 2, needs to determine the best path to reach the destination prefix. For this, two solutions are possible. The first one is to perform active measurements as done by some commercial products [9], [8]. Unfortunately, this approach is not scalable since the number of paths that must be probed increases quadratically with the number of ASs present in the architecture[1]. The cost of sending those probes can be justified when the source AS sends a large amount of traffic to the destination AS, but not for all paths.

However, as shown in the previous section, there can be several paths with a low delay towards a destination and a few paths with a much higher delay. Thus, in practice, the main issue is often to ensure that a path with a long delay is not selected by the source AS. To avoid selecting such paths, we rely on a modified version[2] of the Vivaldi algorithm [11].

We use the improved Vivaldi coordinate system on the AS border routers. Each border router sends probes to a few tens of distant border routers. Based on the delay measurements, each border router computes its coordinates and dynamically updates the AS' DNS server. DNS extensions such as [30] can

[1] If $E_p(i)$ (resp. $E_p(j)$) is the number of possible tunnel tail-ends (resp. head-ends) in the destination (resp. source) AS and N the number of ASs, then $\sum_{i=0}^{N-1}\sum_{j=i+1}^{N-1} E_p(i).E_p(j)$ paths must be actively probed.

[2] Due to space limitations, we cannot describe those modifi cations in details in this paper. Basically, we changed the Vivaldi algorithm to ensure that it always converges and have validated our modifi cations based on the RIPE delay measurements. A description of these changes is available at `http://www.info.ucl.ac.be/people/delaunoi/svivaldi`.

```
;
; AS1's reverse DNS server
;
0.0.0.11.IN-ADDR.ARPA.   IN APL ( 1:11.0.0.0/8 )
                         IN TUNNEL ( IP:R1.AS1.NET
                                     IP:R2A.AS1.NET
                                     IP:R2B.AS1.NET )
;
; AS1.NET
;
R1.AS1.NET.             IN A 1.0.3.1
                        IN COORD 1:9:12:1       ; X:Y:H:E
R2A.AS1.NET.            IN A 2.0.2.2
                        IN COORD 5:23:6:2.5     ; X:Y:H:E
R2B.AS1.NET.            IN A 3.0.3.1
                        IN COORD 6:20:2:1.5     ; X:Y:H:E
;
; AS2's reverse DNS server
;
0.0.0.12.IN-ADDR.ARPA.  IN APL ( 1:12.0.0.0/8 )
                        IN TUNNELSRC ( IP:R9.AS2.NET )
19.1.0.12.IN-ADDR.ARPA. IN APL (1:12.0.0.0/8)
;
; AS2.NET
;
R9.AS2.NET.             IN A 12.0.1.19
                        IN COORD 32:20:4:1.3    ; X:Y:H:E
```

Fig. 4. Sample DNS configuration for $AS1$ and $AS2$

be used to allow a router to securely update its DNS records. Thus, the DNS server can contain up-to-date coordinates for all the entry border routers inside its AS. Those coordinates are encoded inside the *COORD* resource record. It contains the coordinates $(x, y, height)$ and an error estimation (e) computed using our version of the Vivaldi algorithm [11]. Figure 4 shows a DNS server configured by assuming such Vivaldi coordinates.

The main advantage of the coordinate system is that the euclidean distance between the coordinates of two routers is a good prediction of the round-trip-time between the two routers in the Internet. Furthermore, if each border router probes $\overline{K}$ neighbors on average, then only $\overline{K}.N$ paths must be probed, a much lower overhead than with active probing.

In our architecture, when a border router needs to select the path with the lowest delay to reach a destination, it queries the DNS to determine the border routers of the destination AS and their coordinates. If the lowest delay path was learned via BGP, this path can be used. Otherwise, the border router will establish a tunnel to reach a border router of the destination AS via the best path. In a small stub AS, a single tunnel will probably be used, but nothing in our architecture prevents a large site from establishing several interdomain tunnels to reach a given destination.

It should be noted that the flexibility of the DNS allows to provide other types of information to aid in the selection of interdomain paths or the establishment of interdomain tunnels. For example, a destination AS could provide a DNS RR indicating the available bandwidth on its ingress links to favor the selection of the less loaded ingress link. Another possibility would be to indicate a preference for one of its links over others.

To be accepted by ISPs, this utilization of interdomain tunnels should not cause new security issues. Today, the current practice to avoid IP spoofing attacks is to rely on ingress filtering [7]. Our tunnel-based architecture can be made as secure as the current Internet architecture. Consider in figure 2 that malicious host `17.12.9.1` sends IP packets with source address `12.0.1.1` inside an IP tunnel towards `1.0.3.1`. When those packets arrive at router $R1$, this router should be able to verify whether `17.12.9.1` is allowed to encapsulate packets with source IP addresses inside the `12.0.0.0/8` prefix. To perform this verification, we propose to dynamically install filters similar to those discussed in [21] on each entry border router upon reception of encapsulated packets. When router $R1$ receives the first encapsulated packet from a distant router, it should query the reverse DNS to obtain the list of IP prefixes that are upstream of this router. This list of prefixes can be encoded by using a APL resource record as defined in [19]. To avoid fake APL RRs, we propose to require that each AS using tunnels encodes inside the reverse DNS for its own prefixes the list of IP addresses that are allowed to initiate IP tunnels as a *TUNNELSRC* DNS RR. For example, when router $R1$ receives the first encapsulated packet from `12.0.1.19` ($R9$), it queries the reverse DNS for the APL RR. Then, router $R1$ queries the *TUNNELSRC* RR associated to `0.0.0.12.in-addr.arpa`. The DNS response indicates that `r9.as2.net` is a valid tunnel source for prefix `12.0.0.0/8`. Note that an additional security measure would be to use the DNSSEC security extensions to cryptographically sign all the DNS records used. Some DNS servers already support those extensions [14].

IV. Performance evaluation

To evaluate the performance of our proposed architecture, we use the RIPE dataset discussed in section II. We compute the coordinates of each node and their evolution against time by replaying the full set of delay measurements over the month. About 300 millions RTT probes were used. The algorithm used to compute the coordinates of a RIPE node is the improved version of the Vivaldi distributed algorithm [11].

Unfortunately, since the BGP routing tables of the RIPE test boxes are not available, we cannot compare the path selected using coordinates with the path that would have been selected by BGP. However, it has been shown that BGP path lengths are not correlated with their performances [17].

For a given pair of multihomed sites, we take the $M \times N$ paths between the M source and N destination border routers. A tunnel is established between the closest source and destination border routers according to their coordinates.

In figure 5, we compare the delay of the path selected using synthetic coordinates with the average delay over all paths, and the worst delay among all paths.

On the x-axis, we show the relative difference $(\delta_{selected} - \delta_{lowest})/\delta_{lowest}$ between the delay of the selected path and the lowest delay among all paths. On the y-axis, we show $f(x)$, the fraction of pairs of multihomed sites for which a relative difference lower than x is observed. We can see that we are able to select the path with the lowest delay about 40% of the time, and that we find a path with a delay at most 20% worse than the lowest delay for about 80% of the pairs of multihomed sites. It should be noted that most RIPE nodes are located in Europe, hence low delays and high relative differences between them are common. The distribution for the path selected using coordinates is computed regularly over time. The 5^{th} and 95^{th} percentiles show that the distribution does not vary much over time.

The coordinates of the entry border routers are regularly updated over the month in order to match the delays observed

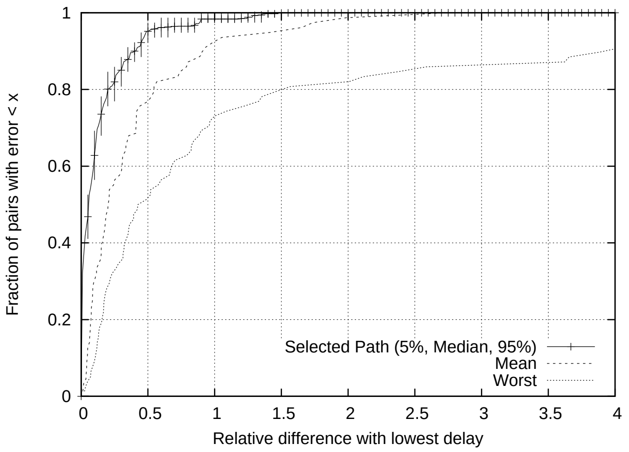

Fig. 5. The cumulative distribution of the relative difference between the delay of the best path and the delay of the path selected using coordinates.

with the neighbors. It is sometimes needed to reestablish a tunnel when the currently selected border routers are no longer the closest ones according to their coordinates. We have evaluated how many times a tunnel is reestablished, per multihomed site, and per day. This number appears to be less than one tunnel change per day in average. Due to space limitations, we cannot report this evaluation here.

V. Related work

The closest approach to using interdomain tunnels for leveraging better Internet performances is Detour [26]. However, the Detour approach assumes that the endsystems will be able to locate the appropriate Detour router. In our approach, tunnels are established between the domain border routers and we rely on the DNS to exchange information between domains. Another approach requiring changes to the endsytems is the utilization of endsystem-based overlay networks such as RON [6]. The idea of explicitly routing traffic through tunnels, based on measurements, has been studied in [16], at the intradomain level.

Our approach has similarities with IPv6 multihoming solutions (see [12] and references therein). With IPv6 multihoming, each endsystem receives several IPv6 addresses, one per provider. By selecting the address that it uses to reach a destination, each host can indirectly select the interdomain path to be used. This approach is unfortunately difficult with IPv4 due to the limited IPv4 address space. Our architecture has the advantage of being deployable today.

Several commercial multihoming techniques have also been proposed recently, but few details are available about their operation [22], [15]. Those devices typically rely on active probing or use NAT (Network Address Translation) and are focused on small enterprise networks. [28] proposes a BGP-based optimization solution. However, it is only aimed at outbound traffic and it relies on active measurements.

In addition, there are also proposals to bring changes to interdomain routing. For instance, [1] and [24] considered the use of a separate protocol to carry control information and [20] proposes to introduce negotiation between ISPs. Unfortunately, to be used, those protocols and mechanisms must be supported by all transit ASes. This requires changes to potentially all BGP routers in the global Internet. The utilization of interdomain MPLS tunnels suffers from a similar drawback. Our approach only needs a cooperation between the source and the destination AS. Another recent proposal considers the use of network-capabilities to enable loose source routing and apply policies at the forwarding level instead of the routing-level [27]. More drastic changes to the Internet architecture were proposed in [32], [18].

VI. Conclusion and further work

In this paper, we proposed an incrementally deployable extension to the current Internet architecture that provides an accurate selection of the interdomain paths without requiring any change to the BGP messages. Our architecture relies on a few basic principles. First, each border router computes its coordinates by using a network coordinate system. Second, we use the DNS to store information about the border routers that are able to reach each prefix as well as their coordinates. Third, BGP routers use this DNS information to establish IP tunnels to reach important destination prefixes by using the best path in terms of delay towards this prefix. As an example, we have shown by using real measurements that our architecture allows multihomed stub ASes to reduce the delay of their interdomain paths. This is a significant concern for the deployment of services such as Voice or Video over IP in the global Internet.

This combined utilization of IP tunnels with the DNS can be used to provide other types of services. For example, a stub AS could use such tunnels to load-balance the traffic on its access link to reduce congestion and a small transit AS could terminate the tunnels on behalf of its customers …

We are currently evaluating the performance of the proposed architecture in more details in a simulation environment and we intend to implement the proposed architecture on an open-source router platform and a DNS server.

Acknowledgments

We thank the RIPE NCC for providing the Test Traffic Measurements Service and allowing us to use the collected raw data.

References

[1] S. Agarwal, C. Chuah, and R. Katz. OPCA: Robust Interdomain Policy Routing and Traffic Control. In *Proceedings of the 6th International Conference on Open Architecture and Network Programming, IEEE OpenArch*, April 2003.

[2] A. Akella et al. A measurement-based analysis of multihoming. In *Proceedings ACM SIGCOMM'03*, August 2003.

[3] A. Akella, B. Maggs, A. Seshan, A. Shaikh, and R. Sitaraman. A Measurement-based Analysis of Multihoming. In *Proceedings of ACM SIGCOMM*, Karlsruhe, Germany, August 2003.

[4] A. Akella, S. Seshan, and A. Shaikh. Multihoming Performance Benefits: An Experimental Evaluation of Practical Enterprise Strategies. In *Proceedings of USENIX Annual Technical Conference*, Boston, MA, 2004.

[5] G. Almes, S. Kalidindi, and M. Zekauskas. A round-trip delay metric for IPPM. RFC 2681, IETF, September 1999.

[6] D. Andersen, H. Balakrishnan, M. Kaashoek, and R. Morris. Resilient overlay networks. In *SOSP 2001*, 2001.

[7] F. Baker and P. Savola. Ingress filtering for multihomed networks. RFC3704, March 2004.

[8] J. Bartlett. Optimizing multi-homed connections. *Business Communications Review*, 32(1):22–27, January 2002.

[9] Cisco Systems, Inc. Cisco IOS Optimized Edge Routing. http://www.cisco.com/warp/public/732/Tech/routing/oer/, November 2004.

[10] G. Cristallo and C. Jacquenet. Providing quality of service indication by the BGP-4 protocol : the QoS_NLRI attribute. Internet draft, draft-jacquenet-qos-nlri-03.txt, work in progress, July 2001.

[11] F. Dabek, F. Kaashoek, and R. Morris. Vivaldi: A decentralized network coordinate system. In *Proceedings of ACM SIGCOMM'04*, Portland, Oregon, USA, August 2004.

[12] C. de Launois, B. Quoitin, and O. Bonaventure. Leveraging Network Performances with IPv6 Multihoming and Multiple Provider-Dependent Aggregatable Prefixes. In *To appear in proc. of QoSIP 2005*, February 2005.

[13] F.Georgatos et al. Providing Active Measurements as a Regular Service for ISP's. In *Proceedings of PAM'01*, Amsterdam, April 2001. http://www.ripe.net/ttm.

[14] M. Gieben. DNSSEC. *IP Protocol Journal*, 7(2), June 2004.

[15] F. Guo, J. Chen, W. Li, and T. Chiueh. Experiences in Building a Multihoming Load Balancing System. In *Proceedings of IEEE INFOCOM*, March 2004.

[16] T. Guven, C. Kommareddy, R. La, M.A. Shayman, and B. Bhattacharjee. Measurement Based Optimal Multi-path Routing. In *Proceedings of IEEE INFOCOM*, March 2004.

[17] B. Huffaker, M. Fomenkov, D. Plummer, D. Moore, and K. Claffy. Distance Metrics in the Internet. In *Proc. of IEEE International Telecommunications Symposium (ITS)*, September 2002.

[18] H. Tahilramani Kaur, S. Kalyanaraman, A. Weiss, S. Kanwar, and A. Gandhi. Bananas: an evolutionary framework for explicit and multipath routing in the internet. In *Proceedings of the ACM SIGCOMM, FDNA*, pages 277–288, 2003.

[19] P. Koch. A DNS RR Type for Lists of Address Prefixes (APL RR). RFC3123, June 2001.

[20] R. Mahajan, D. Wetherall, and T. Anderson. Towards coordinated interdomain traffic engineering. In *ACM SIGCOMM HotNets' 2004*, November 2004.

[21] P. Marques, N. Sheth, R. Raszuk, B. Greene, and D. McPherson. Dissemination of flow specification rules. Internet draft, draft-marques-idr-flow-spec-02.txt, work in progress, December 2004.

[22] P. Morrissey. Mapping out the Best Route. *Network Computing, http://www.nwc.com*, December 2003.

[23] University of Oregon. Routeviews project. http://www.routeviews.org, 2004.

[24] P. Pan, E. Hahne, and H. Schulzrinne. BGRP: A tree-based aggregation protocol for inter-domain reservations. *Journal of Communications and Networks*, 2(2), June 2000.

[25] S. Sangli, D. Tappan, and Y. Rekhter. BGP extended communities attribute. Internet draft, draft-ietf-idr-bgp-ext-communities-07.txt, work in progress, April 2004.

[26] S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, and J. Zahorjan. Detour: Informed internet routing and transport. *IEEE Micro*, 19(1):50–59, 1999.

[27] J. Snoeren and B. Raghavan. Decoupling Policy from Mechanism in Internet Routing. In *Proceedings of ACM Hotnets-II*, January 2004.

[28] Cisco Systems. Cisco Optimized Edge Routing. http://www.cisco.com/warp/public/732/Tech/routing/oer, May 2004.

[29] D. Walton, D. Cook, A. Retana, and J. Scudder. Advertisement of Multiple Paths in BGP. Internet draft, draft-walton-bgp-add-paths-01.txt, work in progress, November 2002.

[30] B. Wellington. Secure Domain Name System (DNS) Dynamic Update. RFC3007, November 2000.

[31] L. Xiao, K. Lui, J. Wang, and K. Nahrstedt. QoS extensions to BGP. In *ICNP 2002*, Paris, France, November 2002.

[32] X. Yang. Nira: a new internet routing architecture. In *Proceedings of the ACM SIGCOMM, FDNA*, pages 301–312, 2003.