

Diagnosing Path Inflation of Mobile Client Traffic

Kyriakos Zarifis¹, Tobias Flach¹, Srikanth Nori¹, David Choffnes², Ramesh Govindan¹, Ethan Katz-Bassett¹, Z. Morley Mao³, and Matt Welsh⁴

¹ University of Southern California ² Northeastern University
³ University of Michigan ⁴ Google Inc.

Abstract. As mobile Internet becomes more popular, carriers and content providers must engineer their topologies, routing configurations, and server deployments to maintain good performance for users of mobile devices. Understanding the impact of Internet topology and routing on mobile users requires broad, longitudinal network measurements conducted from mobile devices. In this work, we are the first to use such a view to quantify and understand the causes of geographically circuitous routes from mobile clients using 1.5 years of measurements from devices on 4 US carriers. We identify the key elements that can affect the Internet routes taken by traffic from mobile users (client location, server locations, carrier topology, carrier/content-provider peering). We then develop a methodology to diagnose the specific cause for inflated routes. Although we observe that the evolution of some carrier networks improves performance in some regions, we also observe many clients - even in major metropolitan areas - that continue to take geographically circuitous routes to content providers, due to limitations in the current topologies.

1 Introduction

As mobile Internet becomes more popular, carriers and content providers must engineer their topologies, routing configurations, and server deployments to maintain good performance for users of mobile devices. A key challenge is that performance changes over space and time, as users move with their devices and providers evolve their topologies. Thus, understanding the impact of Internet topology and routing on mobile users requires broad, longitudinal network measurements from mobile devices.

In this work, we are the first to identify and quantify the performance impact of several causes for inflated Internet routes taken by mobile clients, based on a 901K measurement dataset gathered from mobile devices during 18 months. In particular, we isolate cases in which the distance traveled along a network path is significantly longer than the direct geodesic distance between endpoints. Our analysis focuses on performance with respect to Google, a large, popular content provider that peers widely with ISPs and hosts servers in many locations worldwide. This rich connectivity allows us to expose the topology of carrier networks as well as inefficiencies in current routing. We constrain our analysis to devices located in the US, where our dataset is densest.

Our key results are as follows: First, we find that path inflation is endemic: in the last quarter of 2011 (Q4 2011), we observe substantial path inflation in at least 47% of measurements from devices, covering three out of four major US carriers. While the average

fraction of samples experiencing path inflation dropped over the subsequent year, we find that one fifth of our samples continue to exhibit inflation. Second, we classify root causes for path inflation and develop an algorithm for identifying them. Specifically, we identify whether the root cause is due to the mobile carrier’s topology, the peering between the carrier and Google, and/or the mapping of mobile clients to Google servers. Third, we calculate the impact of this path inflation on network latencies, which are important for interactive workloads typical in the mobile environment. We show that the impact on end-to-end latency varies significantly depending on the carrier and device location, and that it changes over time as topologies evolve. We estimate that additional propagation delay can range from at least 5-50ms, which is significant for service providers [4]. We show that addressing the source of inflation can reduce download times by hundreds of milliseconds. We argue that it will become increasingly important to optimize routing as last-mile delays in mobile networks improve and the relative impact of inflation becomes larger. Last, we make our dataset publicly available and provide an online tool for visualizing our network performance data.

2 Background and Related Work

Background. As Internet-connected mobile devices proliferate, we need to understand factors affecting Internet service performance from mobile devices. In this paper, we focus on two factors: the carrier topology, and the routing choices and peering arrangements that mobile carriers and service providers use to provide access to the Internet.

The device’s carrier network can have multiple Internet *ingress points* — locations where the carrier’s access network connects to the Internet. The carrier’s network may also connect with a Web service provider at a *peering point* — a location where these two networks exchange traffic and routes. The *Domain Name System (DNS) resolvers* from (generally) the carrier and the service provider combine to direct the client to a server for the service by resolving the name of the service to a server IP address.

Idealized Operation. This paper focuses on Google as the service provider. To understand how mobile devices access Google’s services, we make the following assumptions about how Google maps clients to servers to minimize latency. First, Google has globally distributed servers, forming a network that peers with Internet service provider networks widely and densely [2, 5]. Second, Google uses DNS to direct clients (in our case, mobile devices) to topologically nearby servers. Last, Google can accurately map mobile clients to their DNS resolvers [6]. Since its network’s rich infrastructure aims at reducing client latency, Google is an excellent case study to understand how carrier topology and routing choices align with Google’s efforts to improve client performance.

We use Fig. 1 to illustrate the ideal case of a mobile device connecting to a Google server. A mobile device uses DNS to look up `www.google.com`. Google’s resolver returns an optimal Google destination based on a resolver-server mapping. Traffic from the device traverses the carrier’s access network, entering the Internet through an ingress point. Ideally, this ingress point is near the mobile device’s location. The traffic enters Google’s network through a nearby peering point and is routed to the server.

In this paper, we identify significant deviations from this idealized behavior. Specifically, we are interested in *metro-level path inflation* [10], where traffic from a mobile

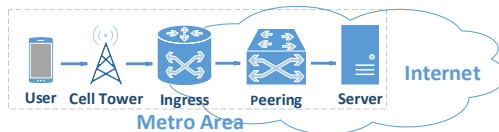


Fig. 1. Optimal routing for mobile clients.

client to a Google server exits the metropolitan (henceforth metro) area even though Google has a presence there. This metro-level inflation impacts performance by increasing latency.

Example Inflation. The carrier topology determines where traffic from mobile hosts enters the carrier network. Prior work has suggested that mobile carriers have relatively few ingress points [11]. Therefore, traffic from a mobile client in the Los Angeles area may enter the carrier’s backbone network in San Francisco because the carrier does not have an Internet ingress in Los Angeles. If the destination service has a server in Los Angeles, this topology design can add significant latency compared to having an ingress in LA. Routing configurations and peering arrangements can also cause path inflation. As providers move services to servers located closer to clients, the location where carriers peer with a provider’s network may significantly affect performance. For instance, if a carrier has ingress points in Seattle and San Francisco, but peers with a provider only in San Francisco, it may route Seattle traffic to San Francisco even if the provider has a presence in Seattle.

Related Work. Research showed 10 years ago that interdomain routes suffer from path inflation particularly due to infrastructure limitations like peering points only at select locations, but also due to routing policies [8]. In recent work, researchers investigated reasons for suboptimal performance of clients of Google’s CDN, showing that clients in the same geographical area can experience much different latencies to Google’s servers [4, 12]. Cellular networks present new challenges and opportunities for studying path inflation. One study demonstrates differences in metro-area mobile performance but does not investigate the root causes [7]. Other work shows that routing over suboptimal paths due to lack of nearby ingress points causes a 45% increase in RTT latency because of the additional distance traveled, compared to idealized routing [1]. We show how topologies and path inflation have evolved, and that ingress point location is only one of several the factors that can affect performance.

3 Dataset

Data Collected. Our data consists of network measurements (ping, traceroute, HTTP GET, UDP bursts and DNS lookups) issued from Speedometer, an internal Android app developed by Google and deployed on thousands of volunteer devices. Speedometer conducts approximately 20-25 measurements every five minutes, as long as the device has sufficient remaining battery life (80%) and is connected to a cellular network.¹

Our analysis focuses on measurements toward Google servers including 310K traceroutes, 300K pings and 350K DNS lookups issued in three three-month periods (2011

¹ The app source is available at: <https://github.com/Mobiperf/Speedometer>

Q4, 2012 Q2 and Q4). We focus on measurements issued by devices in the US, where the majority of users are located, with a particular density of measurements in areas with large Google offices. All users running the app have consented to sharing collected data in an anonymized form.² Some fields are stripped (e.g. device IP addresses, IDs), others are replaced by hash values (e.g. HTTP URLs). Location data is anonymized to the center of a region that contains at least 1000 users and is larger than 1 km^2 .

The above measurements are part of a dataset that we published to a Google Cloud Storage bucket and released under the Creative Commons Zero license³. We also provide *Mobile Performance Maps*, a visualization tool to navigate parts of the dataset, understand network performance, and supplement the analysis in this paper: <http://mpm.cs.usc.edu>.

Finding Ingress Points. We assume that an ingress point serves clients in a particular geographic region and as such an ingress point’s IP address should not appear in traceroutes of clients which have a different ingress point closer to them. To determine the serving range and approximate location of the ingress points we extracted carrier-dependent features from the traceroutes such that clients observing the same features are clustered together (and therefore served by the same ingress point) while yielding a high location correlation. For example, the majority of Sprint clients observed the IP address pattern $66.1.x.200$ as the first hop where a particular x -value is only observed in a limited geographic area. Thus, we treat each x -value as a separate ingress point. For AT&T clients we achieved the best clustering by grouping clients based on the first hop with a public IP address observed in the traceroute. Finally, for T-Mobile and Verizon we clustered based on the most common carrier address seen by clients located in a 1-by-1 degree latitude/longitude grid. Unless DNS hostnames reveal the ingress point location, we approximate it by using the centroid of the locations of the clients it serves.

Finding Peering Points. To infer peering locations between the carriers and Google, we identified for each path the last hop before entering Google’s network, and the first hop inside it (identified by an IP address from Google’s blocks). Using location hints in the hostnames of those hop pairs, we infer peering locations for each carrier [9]. In cases where the carrier does not peer with Google (*i.e.*, sends traffic through a transit AS), we use the ingress to Google’s network as the inferred peering location.

4 A Taxonomy of Inflated Routes

Types of Path Inflation. Table 1 shows, for traceroutes in our dataset from the four largest mobile carriers in the US, the fraction of routes that incurred a metro-level path inflation as described earlier. For three of the four carriers, *more than half of all traceroutes to Google* experienced a metro-level deviation in Q4 2011. Further, nearly *all* measurements from AT&T customers traversed inflated paths to Google. Note that these results are biased toward locations of users in our dataset and are not intended to be generalized. Nevertheless, at a high-level, this table shows that metro-level deviations occur in routes from the four major carriers, even though Google deploys servers around the

² Google’s privacy and legal teams reviewed and approved data anonymization and release.

³ <http://commondatastorage.googleapis.com/speedometer/README.txt>

	AT&T	Sprint	T-Mobile	Verizon
Q4 2011	0.98	0.10	0.65	0.47
Q2 2012	0.98	0.21	0.25	0.15
Q4 2012	0.00	0.21	0.20	0.38

Table 1. Fraction of traceroutes from major US carriers that show metro-level inflation.

world to serve nearby clients [4]. However, we also observe that the fraction of paths experiencing metro-level inflation decreases significantly over the subsequent 12 months. As we will show, we can directly link some of these improvements to the topological expansion of carriers.

In the rest of the paper, we examine path inflation to understand its causes and to explore what measures carriers have adopted to reduce or eliminate it. We begin by characterizing the different types of metro-level inflations we see in our dataset. We split the end-to-end path into three logical parts: client to carrier ingress point (Carrier Access), carrier ingress point to service provider ingress point (Interdomain), and service provider ingress point to destination server (Provider Backbone). Then we define the following observed traffic patterns of inflated routes:

Carrier Access Inflation. Traffic from a client in metro area **L** (Local) enters the carrier’s backbone in metro area **R** (Remote), and is directed to a Google server in **R**.

Interdomain Inflation. Traffic from a client in area **L** enters the carrier’s backbone in **L**, then enters Google’s network in area **R** and is directed to a Google server there.

Carrier Access-Interdomain Inflation. Traffic from a client in metro area **L** enters the carrier’s backbone in metro area **R**, then enters Google’s network back in area **L** and is directed to a Google server there.

Provider Backbone Inflation. Traffic from a client in area **L** enters the carrier’s backbone and Google’s network in area **L**, but is directed to a Google server in a different area **R**. In all cases, Google servers are known to exist in both metro areas **L** and **R**.

Possible Causes of Path Inflation. If a carrier lacks sufficient ingress points from its cellular network to the Internet, it can cause *Carrier Access Inflation*. For example, if a carrier has no Internet ingress points in metro area **L**, it must send the traffic from **L** to another area **R** (Fig. 2, user B). If a carrier’s access network ingresses into the Internet in metro-area **L**, a lack of peering between the mobile carrier and Google in metro-area **L** causes traffic to leave the metro area, resulting in *Interdomain Inflation* (Fig. 2, user C). If a carrier has too few ingresses and lacks peering near its ingresses, we may observe *Carrier Access-Interdomain Inflation*. In this case a carrier, lacking ingress in area **L**, hauls traffic to a remote area **R**, where it lacks peering with Google. A peering point exists in area **L**, so traffic returns there to enter Google’s network. Though a provider like Google has servers in most major metropolitan areas, it can still experience *Provider Backbone Inflation* if either Google or the mobile carrier groups together clients in diverse regions when making routing decisions. In this case, Google directs at least some of the clients to distant servers. Google may also route a fraction of traffic long distances across its backbone for measurement or other purposes.

Identifying root causes. We run one or more of the following checks, depending on the inflated part(s) of the path, to perform root cause analysis (illustrated in Fig. 3).

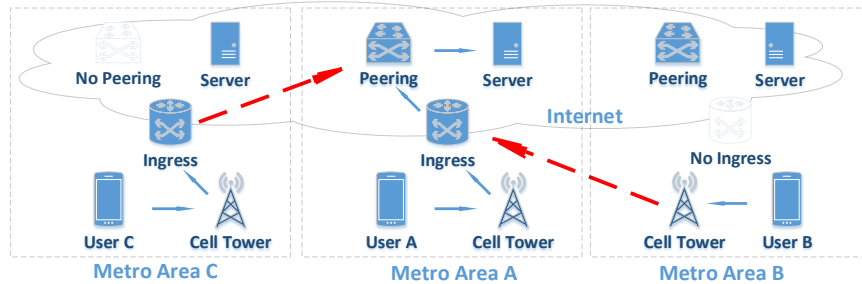


Fig. 2. Different ways a client can be directed to a server. User A is the ideal case, where the traffic never leaves a geographical area. User B and C’s traffic suffers path inflation, due to lack of ingress point and peering point respectively.

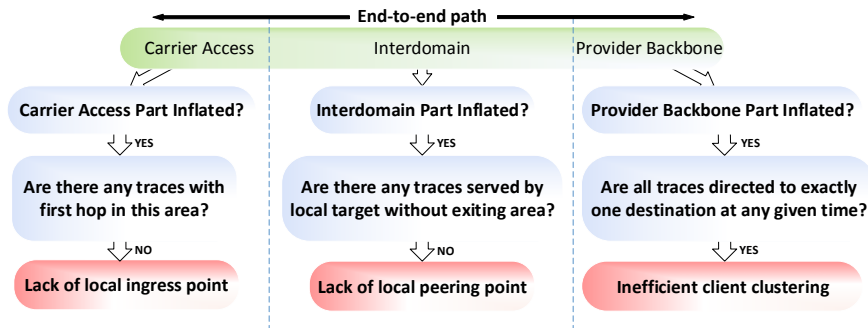


Fig. 3. Root cause analysis for metro-level inflation.

Examining Carrier Access Inflation. For inflated carrier access paths, we determine whether the problem is the lack of an available nearby ingress point. To do so, we examine the first public IP addresses for other traceroutes issued by clients of the same carrier in the same area. If none of those addresses are in the client’s metro area, we conclude there is a **lack of available local ingress**.

Examining Interdomain Inflation. For paths inflated between the carrier ingress point and the ingress to Google’s network, we determine whether it is due to a lack of peering near the carrier’s ingress point. We check whether any traceroutes from the same carrier enter Google’s network in that metro area, implying that a local peering exists. If no such traceroutes exist, we infer a **lack of local peering**.

Examining Provider Backbone Inflation. For paths inflated inside Google’s network, we check for inefficient mappings of clients to servers. We look for groups of clients from different metro areas all getting directed to servers at either one or the other area for some period, possibly flapping between the two areas over time. If we observe that behavior, we infer **inefficient client/resolver clustering**.

A small number of traceroutes (< 2%) experienced inflated paths but did not fit any of the above root causes. These could be explained by load balancing, persistent incorrect mapping of a client to a resolver/server, or a response to network outages.

5 Results

We first present examples of the three dominant root causes for metro-level inflation. We then show aggregate results from our inflation analysis, its potential impact on latency, and the evolution of causes of path inflation over time.

Case studies. For each root cause, we now present one example. For each example, we describe what the traceroutes show, what the diagnosis was, and note the estimated performance hit, ranging from 7-72% extra propagation delay. We constrain our analysis to the period between late 2011 and mid 2012, where the dataset is sufficiently dense.

Lack of ingress point. We observe that all traceroutes to Google from AT&T clients in the NYC area enter the public Internet via an ingress point in Chicago. Thus, Google directs these New York clients to a server in the Chicago area, even though it is not the server geographically closest to the clients. These Chicago servers are approximately 1074km further from the clients than the New York servers are, leading to an expected minimum additional round-trip latency of 16ms (7% overhead) [3].

Lack of peering. We observe AT&T peering with Google near San Francisco (SF),⁴ but not near Los Angeles (LA) or Seattle. Therefore, Google directs clients in those two areas to servers in SF rather than in their local metros. While our data in these regions become sparse after mid 2012, we verified that this inflation persists for clients from LA in Q2 2013. The observed median RTT for Seattle users served by servers in SF is 90ms. Since those servers are 1089km farther away from the servers nearest to the Seattle users, they experience a delay inflation of at least 16ms (21%). As a result, loading even a simple website like the Google homepage⁵ requires an additional 160ms.

Coarse client-server mapping granularity or Inefficient client/resolver clustering. We observe a behavior for Verizon clients that suggests that Google is jointly directing clients in Seattle and SF. At any given time, traffic from both areas was directed towards the same Google servers, either in the Seattle or in the SF area, therefore exhibiting suboptimal performance for some distant clients. Figure 4 illustrates this behavior over a 2-month period. Normally, users served by servers in their metro area observe a median RTT of 22ms and 45ms for SF and Seattle respectively. However, when users in one area served by servers in the other area (indicated by the filled pattern in the figure), the additional 1089km one-way distance adds an extra 16ms delay (an overhead of 72% and 35% for SF and Seattle users respectively).

Other. A small number of traceroutes (less than 2%) exhibited inflated routes but we excluded them from our analysis: First, for some of the traceroutes, our dataset was too sparse to draw strong conclusions. For example, all 151 traceroutes in 2011 from Verizon clients near Miami go to a server near Washington, D.C., but that number of measurements could represent a single user, and so we avoid drawing broader conclusions. Second, we excluded measurements if we observed traceroutes with similar parameters (same carrier, metro region, and time period) which do not exhibit inflation.

Inflation Breakdown by Root Cause. In this section, we show aggregated statistics of some of the observed anomalies that cause performance degradation. We focus on Q4

⁴ For the granularity of our analysis, we treat all locations in the Bay Area as equivalent.

⁵ Downloading the mobile page version required approximately 10 RTTs in test runs.

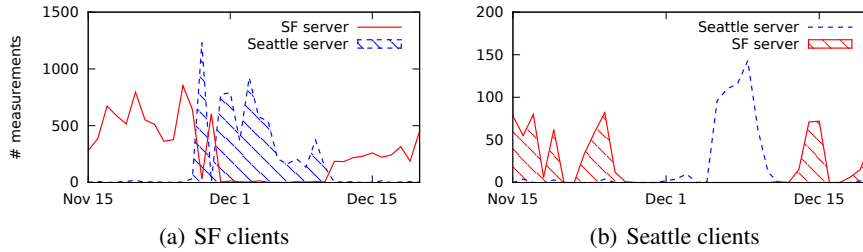


Fig. 4. Server selection flapping due to coarse client-server mapping. Dashed areas denote measurements where the client was directed to a remote server.

	Closest Server	Count	Fraction Inflated	I	P	D	Extra Dst. (km)	Extra RTT (ms)	Extra PLT (ms)
AT&T	SF	7759	1.00	x	x		4200	31.5	315
	Seattle	303	1.00		x		2106	15.8	158
	NYC	2720	1.00	x			2148	16.1	161
Verizon	SF	20528	0.30			x	2178	16.3	163
	Seattle	2435	0.33			x	1974	14.8	148
	NYC	7029	0.98				694	5.2	52

Table 2. Overall results for two carriers for 2011 Q4. The table shows what fraction of all traceroutes from clients in three different locations presented a deviation, cause of the deviation (I = Ingress, P = Peering, D = DNS/clustering), extra distance traveled, extra round trip time (RTT), and extra page load time (PLT) when accessing the Google homepage.

2011 for AT&T and Verizon Wireless, and we select three densely populated geographic locations (SF, New York and Seattle) where there exist Google servers and we have sufficient data points. For all the measurements issued from those areas, we quantify the fraction of metro-level inflations and determine the root cause.

We observed inflated routes from all three regions for both carriers (Table 2). Most of the traceroutes from Verizon clients in the NYC area went to servers near Washington, D.C., but we were unable to discern the exact cause. This represents a small geographic detour and may not impact performance in practice. Verizon clients from the Seattle and SF metro were routed together, possibly as a result of using the same DNS resolvers, as described in our case study above.

For all traces from AT&T clients in the NYC area, the first public AT&T hop is in Chicago, indicating a lack of a closer ingress point. AT&T clients from the SF area were all served by a nearby Google server. However, traffic went from SF to Seattle before returning to the server in SF. In the traceoutes, the first public IP address was always from an AT&T router in Seattle, suggesting a lack of an ingress point near SF, and increasing the RTT by at least 31ms for all traffic. This behavior progressively disappeared in early 2012, with the observed appearance of an AT&T ingress point in the SF area. An informal discussion with the carrier confirms initial deployment of this ingress in 2011. Note that traceroutes from clients in Seattle were also routed to Google targets in the SF area. Though Seattle traffic ingressed locally, AT&T routed it to SF before handing it to Google’s network, indicating a lack of peering in Seattle and explaining why traffic from SF clients returned to SF after detouring to Seattle.

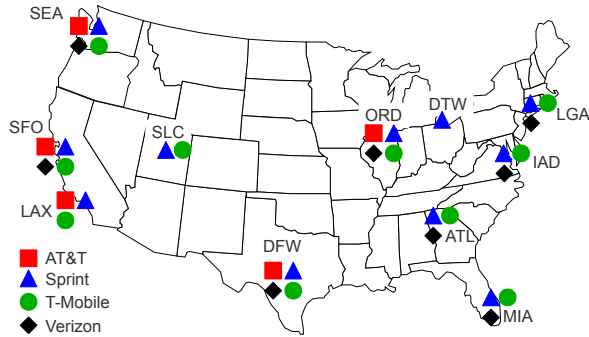


Fig. 5. Observed ingress points for major US carriers.

Evolution of Root Causes. As suggested above, carriers’ topologies have evolved over time. Since our dataset is skewed towards some regions, we cannot enumerate the complete evolution of carrier topology and routing configuration, but can provide insight into why we see fewer path inflation instances over time for some carriers.

Ingress Points. Figure 5 maps the observed ingress points. While our dataset is limited, we can still see indications of improvements. An earlier study [11] found 4-6 ingress points per carrier, whereas our results indicate that some carriers doubled this figure. Specifically, we noticed the appearance of AT&T ingress points in SF and LA, and of at least one Sprint ingress point in LA during the measurement period.

Peering points. Table 3 summarizes the peering points that we observe. In 2011, most traceroutes from Sprint users in LA are directed to Google servers in Texas or SF. In measurements from Q2 2012, we observed an additional peering point between Sprint and Google near LA. Around the same time, we observe that Google started directing Sprint’s LA clients to LA servers.

Carrier	Peering locations (2011 Q4)	(2012 Q2)	(2012 Q4)
AT&T	dfw, hou, msp, ord, pdx, sat, sfo, sjc	+ atl, cmh	+ den
Sprint	ash, atl, dfw, lga, ord, sea, sfo, sjc	+ bur, lax, lgb	
T-Mobile	dfw, iad, lax, lga, msp, sea, sfo	+ mil	+ mia
Verizon	atl, dal, dfw, hou, iad, lax, ord, scl, sea, sfo, sjc		+ ash, mia

Table 3. Observed peering locations between carriers and Google.

6 Path Inflation Today

Our measurements show that many instances of path inflation in the US disappeared over time. However, in addition to the persistent lack of AT&T peering in the LA area mentioned earlier, we see evidence for inflated paths in other regions of the world (from Q3 2013 measurement data). For example, clients of Nawras in Oman are directed to servers in Paris, France instead of closer servers in New Delhi, India. This increases the round trip distance by over 7000km, and may be related to a lack of high-speed routes to the servers in India. We also see instances of path inflation in regions with well-developed infrastructure. E-Plus clients in southern Germany are delegated to Paris or

Hamburg servers instead of a close-by server in Munich, and Movistar clients in Spain are directed to servers in London instead of local servers in Madrid. These instances suggest that path inflation is likely to be a persistent problem in many parts of the globe, and motivate the design of a continuous measurement infrastructure for identifying instances of path inflation, and diagnosing their root causes.

7 Conclusions

This paper took a first look into diagnosing path inflation for mobile client traffic, using a large collection of longitudinal measurements gathered by smartphones located in diverse regions and carrier networks. We provided a taxonomy of causes for path inflation, identified the reasons behind observed cases, and quantified their impact. We found that a lack of carrier ingress points or provider peering points can cause lengthy detours, but, in general, routes improve as carrier and provider topologies evolve. Our dataset is publicly available at <http://mpm.cs.usc.edu> and our ongoing work includes developing techniques for automatic detection of evolving topology issues.

References

1. Dong, W., Ge, Z., Lee, S.: 3G Meets the Internet: Understanding the Performance of Hierarchical Routing in 3G Networks. In: ITC (2011)
2. Gill, P., Arlitt, M.F., Li, Z., Mahanti, A.: The Flattening Internet Topology: Natural Evolution, Unsightly Barnacles or Contrived Collapse? In: PAM (2008)
3. Katz-Bassett, E., John, J.P., Krishnamurthy, A., Wetherall, D., Anderson, T., Chawathe, Y.: Towards IP geolocation using delay and topology measurements. In: IMC (2006)
4. Krishnan, R., Madhyastha, H.V., Srinivasan, S., Jain, S., Krishnamurthy, A., Anderson, T., Gao, J.: Moving Beyond End-to-End Path Information to Optimize CDN Performance. In: IMC (2009)
5. Labovitz, C., Iekel-Johnson, S., McPherson, D., Oberheide, J., Jahanian, F.: Internet inter-domain traffic. In: SIGCOMM (2010)
6. Mao, Z.M., Cranor, C.D., Douglis, F., Rabinovich, M., Spatscheck, O., Wang, J.: A Precise and Efficient Evaluation of the Proximity Between Web Clients and Their Local DNS Servers. In: USENIX ATC (2002)
7. Sommers, J., Barford, P.: Cell vs. WiFi: on the performance of metro area mobile connections. In: IMC (2012)
8. Spring, N.T., Mahajan, R., Anderson, T.E.: The causes of path inflation. In: SIGCOMM (2003)
9. Spring, N.T., Mahajan, R., Wetherall, D., Anderson, T.E.: Measuring ISP topologies with Rocketfuel. *IEEE/ACM Trans. Netw.* 12(1) (2004)
10. Tangmunarunkit, H., Govindan, R., Shenker, S., Estrin, D.: The Impact of Routing Policy on Internet Paths. In: INFOCOM (2001)
11. Xu, Q., Huang, J., Wang, Z., Qian, F., Gerber, A., Mao, Z.M.: Cellular data network infrastructure characterization and implication on mobile content placement. In: SIGMETRICS (2011)
12. Zhu, Y., Helsley, B., Rexford, J., Siganporia, A., Srinivasan, S.: LatLong: Diagnosing Wide-Area Latency Changes for CDNs. *IEEE TNSM* 9(3) (2012)