Time is Money: Header Bidding and Latency

Waqar Aqeel and cSpeed team
waqeel@cs.duke.edu,cspeed.net/team

ABSTRACT

The Internet is now latency-bound for most purposes. While the effect of latency has been studied in applications such as search engines, e-commerce, gaming, and virtual reality, digital advertising has not received this attention. Billions of ad slots are sold in programmatic auctions every day through Real-Time Bidding and Header Bidding. Header Bidding is particularly sensitive to latency as the auction typically takes place inside the user's browser. To quantify this effect, we propose a measurement study on the relationships of websites, ad exchanges, bid prices, and latency. We also present the design and implementation of our measurement tool as a web browser extension that leverages the increased visibility of client side auctions to report bid prices and fine-grained timing data. From this study, we hope to establish that lower latency leads to higher revenues for publishers, and better bid confidence for advertisers.

1. INTRODUCTION

Internet throughput has steadily increased to the point where it is not the bottleneck anymore. Latency is the primary performance indicator in this scenario. Besides degrading user experience, high latency adversely affects online businesses as well. Google quantified this effect on web searches and found that a 400 ms increase in latency decreases number of searches per user by 0.74% [5]. In the e-commerce domain, CDNs present reduction in latency as one of their key value propositions, citing, for example, a 100 ms increase in latency causing 1% loss in sales per user for Amazon [2]. This has prompted recent work on ambitious infrastructural revamp of the Internet to reduce latency to its theoretical limit [12].

Digital advertising, especially realtime auctions are also latency-sensitive. In the US alone, this is an $83 billion industry [6] that has been consistently showing double-digit growth for two decades. Latency plays a crucial role in Real-Time Bidding (RTB). In fact, Google AdX, one of the largest ad exchanges, requires that all bidders for an ad auction respond within a deadline of 120ms [8] after it sends them the bid request. Leaving the industry recommended latency buffer of 20 ms, bid computation, and data fetch time of 40 ms [10], it only leaves 60 ms for the round trip time between a bidder and Google AdX. Also, the exchange has only four trading locations: US East Coast, US West Coast, Amsterdam, and Hong Kong. Given the current state of Internet latency [4], this is a severe restriction on bidder location and connection quality. Accordingly, Google AdX recommends that bidders peer directly with them or colocate to reduce latency and latency volatility.

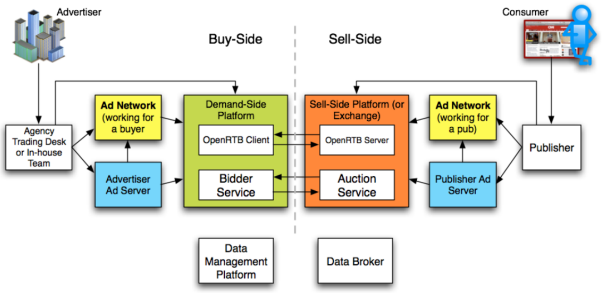

Figure 1: The OpenRTB Ecosystem, reproduced

Latency becomes even more central when it comes to in-browser auctions, such as in Header Bidding. Since user browsers cannot be colocated with bidders, publishers have to set very long deadlines (500 ms to 3000 ms) to make sure maximum number of bidders can meet the deadline. Clearly, latency and related considerations have a profound effect on the advertisement infrastructure and operations. To quantify this effect, we develop a measurement tool that studies real auctions happening in volunteers' browsers and reports on live bid timing and pricing data.

In this paper, we present a brief overview of online ad technology, RTB, and Header Bidding in § 2. In § 3, we set the goals for this measurement study and outline what we aim to find out. § 4 details the design and implementation of our measurement tool. Lastly, we conclude in § 6.

2. BACKGROUND

From 1994 to 2009, the online advertising industry was dominated by privately negotiated premium contracts and ad networks. These ad networks connected a large number of advertisers and publishers. Inventory transactions were made in bulk without much regard to identities and profiles of users that consumed those impressions. Real-time bid-

#	Step in auction	Time since navigation
1	Browser fetches HTML content of the page	6ms
2	Javascript in the HTML header tells Prebid which SSPs to contact	7ms
3	Prebid calls given SSPs in parallel and sends them tracking information such as cookies	20ms
4	SSPs conduction auction with their registered DSPs and return winning bids to the Prebid	400ms
5	Prebid hears from all requested SSPs or preconfigured timeout expires	500ms
6	Prebid sends ad request to the ad server along with bids it received from SSPs	600ms
7	Ad server determines which line item to serve and tells the browser	650ms
8	Browser fetches the ad creative and makes trackback call to the winning SSP	700ms

Table 1: Step in the auction and ad display process and time elapsed since user navigated to page

ding arrived in 2009 as a game changing technology [7] that reduced the transaction granularity level to a single impression. With background information on every user, advertisers were in complete control of exactly how much they want to spend on any given impression. This technology evolution, and continued growth in industry size enabled specialization and many new roles were introduced to the advertising landscape. Figure 1 shows the RTB ecosystem as depicted in the OpenRTB specification [9]. The interested reader should refer to [14] for a more detailed account of the evolution and technology behind RTB.

Here, we define some terminology that will be useful for our discussion:

Ad Exchange is the marketplace where the auction for each ad impression is held individually. Publishers or SSPs, and advertisers or DSPs connect to the ad exchange to participate in the auction.

Demand Side Platform (DSP) helps advertisers manage their integrations with multiple ad exchanges. DSPs typically bid in realtime on behalf of advertisers. This reduces the technological and infrastructural burden on the advertisers.

Ad Server/Publisher Ad Server helps publishers manage ad slots, and advertising campaigns. The most popular ad server is Google's DoubleClick for Publishers, which follows the waterfall model for configured demand sources.

Supply Side Platform (SSP) allows publishers to connect to multiple ad exchanges and DSPs, optimizing for yield and revenue. This reduces the technological burden on the publisher.

Cost-Per-Mille (CPM) is the cost model typically used for ad auctions. It is the amount the advertiser pays for 1000 impressions. Other goal-driven cost modeling might also be in place, such as Cost Per Click (CPC) and Cost Per Action (CPA), but we are only interested in the impression cost.

2.1 Header Bidding

Header Bidding arose from the inherent unfairness of the waterfall model that ad servers have employed for years. By design of the waterfall, some buyers, whether it be privately negotiated buyers or ad exchanges such as Google's AdX, get the first look on any inventory. If they are able to meet the floor price, potential buyers down the waterfall, that might have been willing to bid higher for the impression, do not get the chance to bid. This reduces incentive for advertisers to join these exchanges since they know they will only be offered leftover stock referred to as remnant. It is also bad for publishers, because they lose revenue when they settle for the first buyer in the waterfall that matches the floor price without offering others the chance to bid higher.

Header Bidding overcomes these downsides by holding parallel auctions inside the user's browser. This allows the inventory to be offered at multiple exchanges simultaneously, and every SSP gets 'first look' at the inventory. It also maximizes publisher revenue.

In 2009, AppNexus, an ad tech company, launched Prebid as an alternative to the waterfall [1] and essentially pioneered Header Bidding. Prebid is a wrapper that allows publishers to easily integrate with multiple SSPs without having to write all the handling code themselves. Since Prebid's launch, HB has been adopted by 75% of the Alexa top 1 million publishers, and 56% of all client-side wrappers are Prebid-based [11]. Owing to its popularity, we focus on how Prebid operates and use the same for the measurements in this study.

Table 1 outlines the steps that take place when the user loads a webpage that uses Prebid. Publishers want this entire process to complete as quickly as possible to keep page load times low and deliver ads to their visitors quickly. A large number of TLS connections are established with multiple ad exchanges and the ad server, and multiple HTTPS request-response cycles take place. This makes the process highly latency sensitive. As such, the publisher has to set the timeout and the number of SSPs to solicit very carefully. Too many exchanges and too high a timeout, and the webpage is stuck waiting for bids from SSPs; too few exchanges and too

slow a timeout, and the publishers loses revenue. As we see in the coming sections, low reputation publishers compromise user experience in this tradeoff and set long deadlines to maximize their revenue.

3. MEASUREMENTS

By their design, Header Bidding and Prebid provide us with unprecedented visibility into the SSP infrastructure and operations. From inside the user's browser, we can determine the user's IP address, number and type of ad slots on the webpage, IP addresses of the SSPs Prebid contacts, detailed timing data on the network requests and responses that follow, CPM for each bid received, and whether a bid won an auction after surviving the ad server waterfall. Using this visibility, we intend to make measurements that will answer the following interesting questions:

SSP locations and AS numbers: How geographically distributed are the SSPs that a user has to send bid requests to? A sparser distribution means that latencies would be higher and SSPs could benefit from investing in infrastructure. AS number distribution implies network administration diversity. An SSP could be deployed at 8 different locations, yet all controlled by the same cloud provider.

Client location and connection graph: How far, geographically, does a client typically reach out to contact an SSP? For example, a user in the Midwest US having to contact an SSP on the East Coast indicates a lack of penetration and low chances of reach.

Domain and number of bidders contacted: Do high reputation websites contact fewer SSPs to keep page load time low? More SSPs contact usually translates to higher pageload times because of both network and browser processing latency. Reputation can be determined the same way search engines determine link-worthiness (can use senderscore.org).

Prebid timeout and pageload time: What is the relationship between Prebid timeout and pageload time? Keeping all other factors constant, how does varying the Prebid timeout affect the number of bids timed out?

Domain and average bid CPM: What is the relationship between domain reputation, as determined by senderscore.org using social methods [13], its Alexa rank, number of ad slots on the homepage, and average bid CPM that it receives.

Response time and bid CPM: When an SSP has more compute time to determine CPM, does it affect the CPM amount? Longer compute time will presumably allow the DSP to fetch more data on the user and bid more accurately.

Number of timed out bids: How many bids are too late for the Prebid timeout? Would these bids have won their respective auctions if they had arrived in time? Would publisher revenue rise if the network had lower latency?

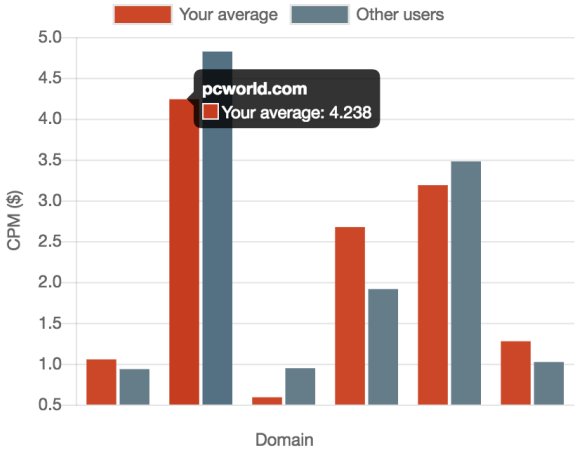

Figure 2: Per slot average bid price for user vs. global average

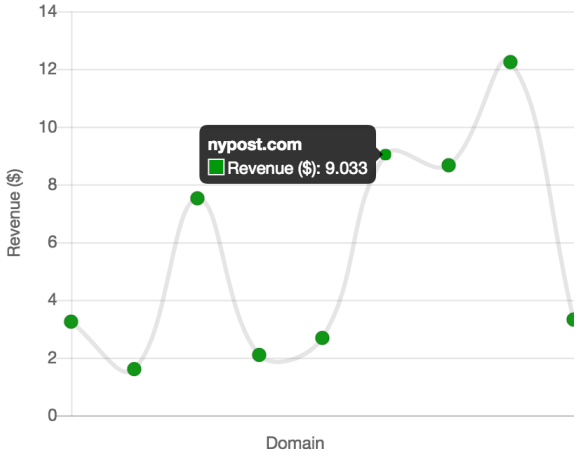

Figure 3: Revenue publisher earned from user's visits

Tuning Prebid timeout: Do publishers configure the Prebid timeout based on user or network attributes, such as location, user visit frequency, observed pageload times?

Number of SSPs and CPM: How does soliciting bids from more SSPs affect the revenue for the publisher? Offering inventory at more auctions should result in higher CPM per ad slot.

SSPs and latency: How do different SSPs compare when it comes to latency for reaching an average end user? Does denser geographical distribution of an SSP offer a competitive edge over other SSPs?

4. DESIGN AND IMPLEMENTATION

As a client-side header bidding wrapper, Prebid is delivered as a JavaScript library that is integrated into each HTML webpage of a publisher. The publisher's custom JavaScript

Figure 4: Timing view for a sample webpage

invokes Prebid's functions to tell it about the ad slots in the page, which SSPs to contact, and the timeout. As such, we can also use Prebid's functions to find out about the incoming bid amounts and round trip times.

We developed our measurement tool as a browser extension using the WebExtensions API that all major browsers including Chrome, Firefox, and Edge support. The WebExtensions API divides an extension into three modules: (i) **the background script** is a persistent script launched at window open, independent of any tabs (ii) **the popup script** is launched when the user clicks the action button of an extension, and terminates when the associated popup closes (iii) **the content script** lives inside the webpage and has access to the DOM of the page. Communication is supported among these modules through message passing.

We use the background script to listen to tab open, close, navigate, and pageload events, and trigger the content script from there. The content script uses its DOM access to query the Prebid library for received bids, their timings, and whether they won the auction for the slot. The background page then uploads some of this data to our collection server.

Bid prices and winning bids are stored in the WebExtensions storage.local area. When the action button is clicked, the popup script plots the per slot average bid prices this user has received against the global averages fetched from the collection server. It also plots the revenue different publishers have earned from this user by summing up the CPM for bids that won their auctions. Graphs as shown in Figure 2 and Figure 3 are generated.

4.1 Timing data

To conduct measurements as outlined in section **??**, we need these attributes of bid requests and responses:

- IP addresses of SSPs contacted
- Request round trip time as seen on the network
- Request initiation and finish time as seen by Prebid. These will include the time Prebid takes to prepare requests and parse responses, and also any competing JavaScript events that the engine has to handle
- Breakdown of response time into DNS lookup, TCP connection setup, TLS handshake, Time to first byte (TTFB), and content download

No one source of timing data allows us to capture all this information, so we use 3 different sources: (i) **Prebid** gives us a timestamp of when it started preparing the request and when it finished parsing the response. This is important as a bid response is considered timed out even if it arrives on the network before the deadline has passed, but Prebid finishes processing it only after. (ii) **browser.network** provides us the wire view of the network. It also gives us the IP addresses of ad exchange servers so we can geolocate them. (iii) **HTML5 Performance.timing API** provides us the breakdown of response time into domain lookup, connection setup etc. The browser does not record this breakdown unless the $Timing - Allow - Origin$ header is set on the response. To circumvent this, we listen for incoming web responses for known ad exchange URL patterns and set the $Timing - Allow - Origin$ header.

The timing graph for a typical webpage visit looks like the one shown in Figure 4. Here, dfp is the ad server request, for which Prebid does not record timing. Some ad exchanges escape detection from the Performance.timing API because deep iframe embedding. In this example, all bid responses have finished Prebid processing before the timeout is triggered and the ad server is loaded. This means that no bids timed out.

5. ACKNOWLEDGEMENTS

We would like to thank AppNexus for their browser extension HeaderBid Expert and their open source list of SSPs and respective URL patterns [3]. These were instrumental in the understanding and implementation of our measurement tool.

6. CONCLUSION

Digital advertisement is an $83 billion industry constantly undergoing technical innovation and change. The most recent trend, Header Bidding, makes the ad auctions highly sensitive to latency by putting the user's browser at the center of the auction. Despite the popularity of this trend, it's infrastructural composition and network behavior has not received attention from the community.

We thus present a measurement plan that investigates the geographic and administrative outlook of the header bidding inftrastructure. We present the design and implementation of a measurement tool that leverages the visibility Header Bidding provides to quantify the effect of latency on publisher revenue and other concerns. With future work, we hope to prove that reducing network latency will benefit publishers, advertisers and all kinds of brokers in the middle. Although we only measure Header Bidding, we believe our measurements can be extrapolated to Real-Time Bidding, and thus give credence to the economic viability of recent ambitious efforts in designing an Internet backbone with lower latency.

7. REFERENCES

[1] AdExchanger. The rise of 'header bidding' and the end of the publisher waterfall. https://adexchanger.com/publishers/the-rise-of-header-bidding-and-the-end-of-the-publisher-waterfall/, 2015.

[2] Akamai. Akamai "10for10". https://www.akamai.com/us/en/multimedia/documents/brochure/akamai-10for10-brochure.pdf, July 2015.

[3] AppNexus. header-bidder-expert. https://github.com/prebid/header-bidder-expert/blob/master/src/js/definitions/calls.js, Jan. 2018.

[4] I. N. Bozkurt, A. Aguirre, B. Chandrasekaran, P. B. Godfrey, G. Laughlin, B. Maggs, and A. Singla. Why is the internet so slow?! In M. A. Kaafar, S. Uhlig, and J. Amann, editors, *Passive and Active Measurement*, pages 173–187, Cham, 2017. Springer International Publishing.

[5] J. Brutlag. Speed Matters for Google Web Search. http://goo.gl/vJq1lx, 2009.

[6] eMarketer. Us ad spending: emarketer's updated estimates and forecast for 2017. https://www.emarketer.com/Report/US-Ad-Spending-eMarketers-Updated-Estimates-Forecast-2017/2002134, Sept. 2017.

[7] Google. The arrival of real-time bidding. https://static.googleusercontent.com/media/www.google.com/en//doubleclick/pdfs/Google-White-Paper-The-Arrival-of-Real-Time-Bidding-July-2011.pdf, 2011.

[8] Google. Latency restrictions and peering. https://developers.google.com/ad-exchange/rtb/peer-guide, Dec. 2017.

[9] I. T. Lab. Openrtb api specification version 2.3.1. https://www.iab.com/wp-content/uploads/2015/05/OpenRTB_API_Specification_Version_2_3_1.pdf, June 2015.

[10] P. Pandey and P. Muthukumar. Real-time ad impression bids using dynamodb. https://aws.amazon.com/blogs/aws/real-time-ad-impression-bids-using-dynamodb/, Apr. 2013.

[11] ServerBid. Header bidding industry index (hbix). https://www.serverbid.com/hbix/, 2018.

[12] A. Singla, B. Chandrasekaran, P. B. Godfrey, and B. Maggs. The internet at the speed of light. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, HotNets-XIII, pages 1:1–1:7, New York, NY, USA, 2014. ACM.

[13] M. Tavakolifard and K. C. Almeroth. Social computing: an intersection of recommender systems, trust/reputation systems, and social networks. *IEEE Network*, 26(4):53–58, July 2012.

[14] S. Yuan, J. Wang, and X. Zhao. Real-time bidding for online advertising: Measurement and analysis. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, ADKDD '13, pages 3:1–3:8, New York, NY, USA, 2013. ACM.