Technical Disclosure Commons

Defensive Publications Series

November 14, 2018

Improving monetization by selecting ad network to optimize latency

Tuna Toksoz

John Dukellis

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

Toksoz, Tuna and Dukellis, John, "Improving monetization by selecting ad network to optimize latency", Technical Disclosure Commons, (November 14, 2018)
https://www.tdcommons.org/dpubs_series/1649

This work is licensed under a Creative Commons Attribution 4.0 License.
This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

Improving monetization by selecting ad network to optimize latency

ABSTRACT

Web page or applications sometimes load slowly due to delays incurred in loading on-page or in-app advertisements. Such slow loading ads can lead to users becoming inattentive to the ad, and user attrition, thus reducing revenue and monetizable value of the property for the website owner or application developer.

This disclosure provides techniques that enable web publishers and app developers to compare ad networks for latency and to optimize the selection of ad networks from which an advertisement is inserted. The techniques also enable comparison of the cost-per-mille revenue derived from an ad network with the loss due to attrition caused the latency of that ad network. Such comparison enables accurate setting of bidding price for ads originating from an ad network.

KEYWORDS

CPM; user attrition; in-app advertising; online advertising; ad network; latency; ad auction; app monetization; ad auction

BACKGROUND

Web page or applications sometimes load slowly due to delays incurred in loading on-page or in-app advertisements. Such slow loading ads can lead to users becoming inattentive to the ad, and user attrition, thus reducing revenue and monetizable value of the property for the website owner or application developer.

DESCRIPTION

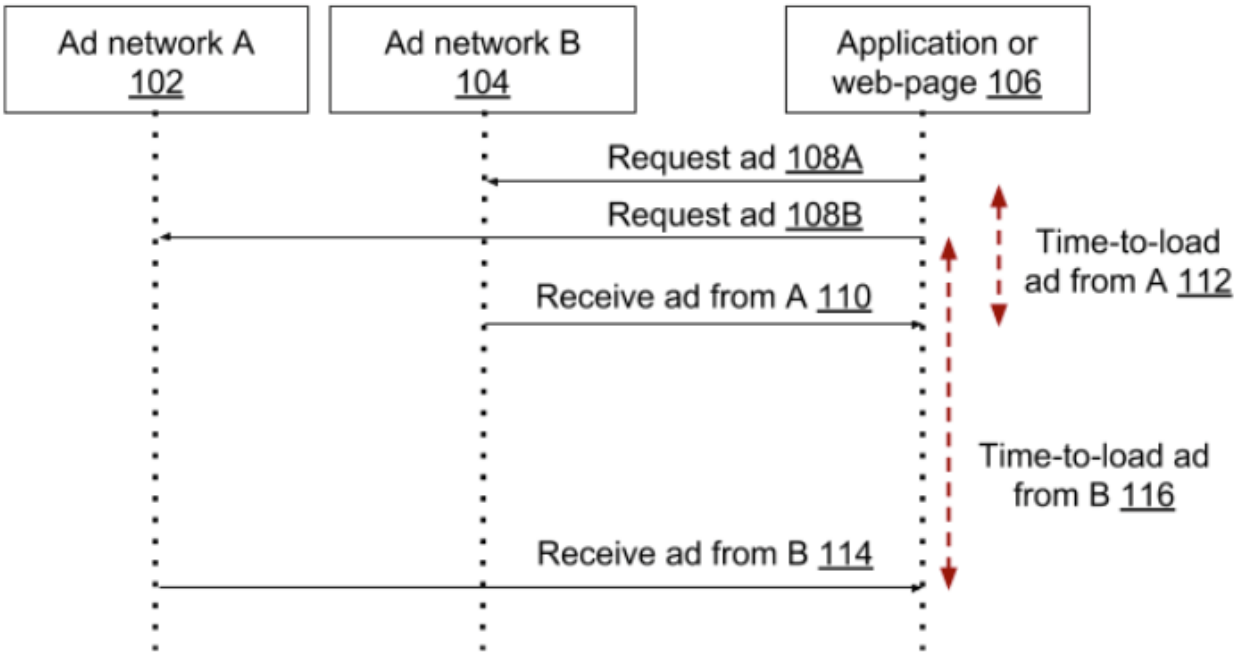

Fig. 1: Comparing latencies of different ad networks

Fig. 1 illustrates comparing the latencies of different ad networks, per techniques of this disclosure. An application of web-page (106) requests (108A, 108B) advertisements from two or ad networks (e.g., 102, 104), denoted A and B, at nearly the same time. Ads may be requested from any number of ad networks.

An ad network A responds with ad (110), and the latency of this response is determined (112). The latency includes the time between the initial request for ad and the time to fully download and display the ad asset. The latency thus includes the time to the first cached event which might trigger the start of an ad. In a similar manner, another ad network B responds with ad (114), and the latency of the response is determined (116). The latency of ads from different ad networks is reported to the app or web developer. The app or web developer can take a variety of actions based on this information. For example,

- The app or web developer may institute a policy that excludes ad networks that exceed a threshold latency for more than a threshold fraction of ad requests.
- The app or web developer may adjust pricing such that ads from a slower ad network are presented with a higher bidding price.
- The app or web developer may maintain multiple ad networks to optimize on-time availability of an ad.

The rationale for each of the above actions is explained below.

Excluding ad networks that exceed latency

The rationale for excluding ad networks that do not meet latency thresholds is that users often leave if advertisements don't appear (or pages/apps don't load completely) within a certain time. A user that leaves is a significant negative event for the application or web publisher. The techniques of this disclosure track and report latencies of an ad network both at an individual app level and across multiple apps. The techniques also provide analytics data that enable development of an attrition-versus-delay graph similar to the one illustrated in Fig. 2.

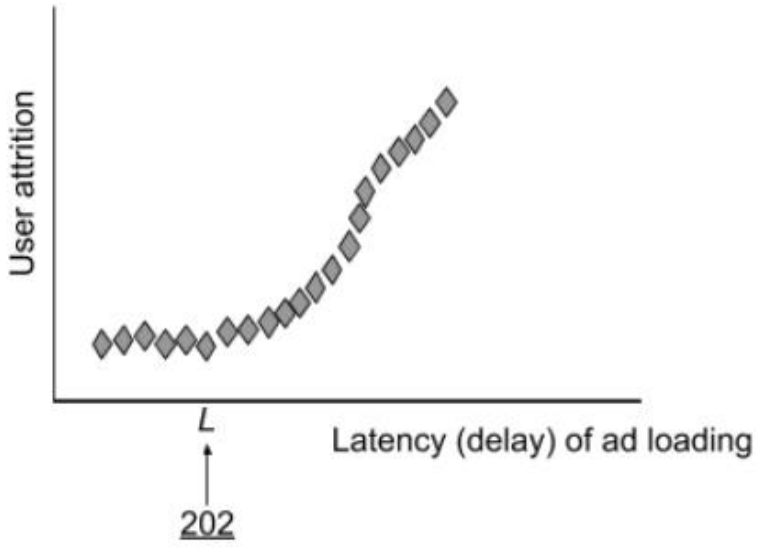

Fig 2: User attrition versus ad-loading latency

Attrition-versus-latency graphs, e.g., as illustrated in Fig. 2, enable a web publisher or app developer to determine the amount of delay that drives users to attrition during the ad wait or load time, and thereby establish latency thresholds for ad networks. In Fig. 2, for example, user attrition is subdued until the latency reaches *L* units of time (202), at which point attrition increases relatively rapidly. A web publisher may then stipulate that an ad network provide ads with a latency less than *L* seconds at least, e.g., 30%, of the time.

Adjusting pricing based on ad network latency

The rationale for adjusting bidding price of an ad based on the latency of the ad network is that the audience, and therefore, the revenue of the web publisher or application, declines with latency. A certain amount of latency, even if relatively high, may be acceptable if the advertiser compensates by offering a higher CPM (bid price) for an ad. A high-latency network, e.g., a network that has latency on average three seconds higher than its competitor, may continue placing ads on the application by paying a higher price, e.g., 100% higher bid price, for ads routed through that ad network. The price adjustment can be automated and dynamic, e.g., price adjustment may be based on current overall average CPMs, with sliding marked-up pricing for ads that breach the latency threshold.

Multiple ad networks to serve ads

The rationale for an application or web developer to use multiple ad networks is redundancy, since the chances of having an available ad to insert are higher in comparison to using a single ad network. The likelihood of an ad being provided on-time by a particular ad network is computed based on historical responses, the country of the user, the type of the ad (e.g., video or static-image), etc. The computed likelihood is used to decide whether to run ad requests from multiple ad networks. Another technique to increase ad availability is for the

publisher or application developer to have on hand one or more replacement ads. In the event that a requested ad is not received within a given time threshold from any of the ad networks, the readily available on-hand ad is displayed. The on-hand ad(s) can be renewed or replaced periodically.

The techniques of obtaining and comparing latencies from multiple ad networks, and of setting latency thresholds, adjusting bid prices or using multiple ad networks based on latency, can be implemented as a software development kit (SDK). The SDK can be used by web and app developers. The techniques of this disclosure enable publishers and application developers to improve earnings by serving ads in available inventory locations and by reducing user attrition caused by slow ads.

CONCLUSION

This disclosure provides techniques that enable web publishers and app developers to compare ad networks for latency and therefrom to optimize selection of ad networks. The techniques also enable comparison of the cost-per-mille revenue derived from an ad network with the attrition caused by the latency of the ad network. Such comparison enables accurate setting of bidding price for ads originating from an ad network.