December 03, 2018

Ad network speed comparison tool

Tuna Toksoz

John Dukellis

Ad network speed comparison tool

ABSTRACT

Online ads that take too long to serve lead to users not paying attention to the ad, or even leaving the app. App developers, who want to maximize ad revenue and user engagement, thus have an interest in measuring and optimizing ad-serving times.

This disclosure presents real-time tools to compare latencies of one or more ad networks on a side-by-side basis. App developers can use the tools provided herein to compare ad-network latencies and to optimize the selection of ad networks.

KEYWORDS

Ad network; online ad; mobile ad; ad-serving latency; banner ad

BACKGROUND

Typically, advertising within apps works by calling ad networks serially to determine the one that can fill an ad-slot at a given price. This current procedure to fill ad inventory doesn't account for the length of time taken for the ad to serve or render. Ad-serving latency is a critical part of the user experience, for it is observed that ads that take too long to serve lead to users leaving the app. App developers, who want to maximize ad revenue and user engagement, thus have an interest in measuring and optimizing ad-serving times. Ad-serving latency is particularly important for ad-slots that are filled in real time (not pre-fetched), e.g., banner ads.

DESCRIPTION

This disclosure presents real-time techniques to compare latencies of one or more ad networks on a side-by-side basis. App developers can use the techniques provided herein to compare ad-network latencies and to optimize the selection of ad networks. Per the techniques,

ad-serving latency is made one of the parameters for selecting an ad network, in addition to traditional parameters such as the bid price (value) of the ad slot.

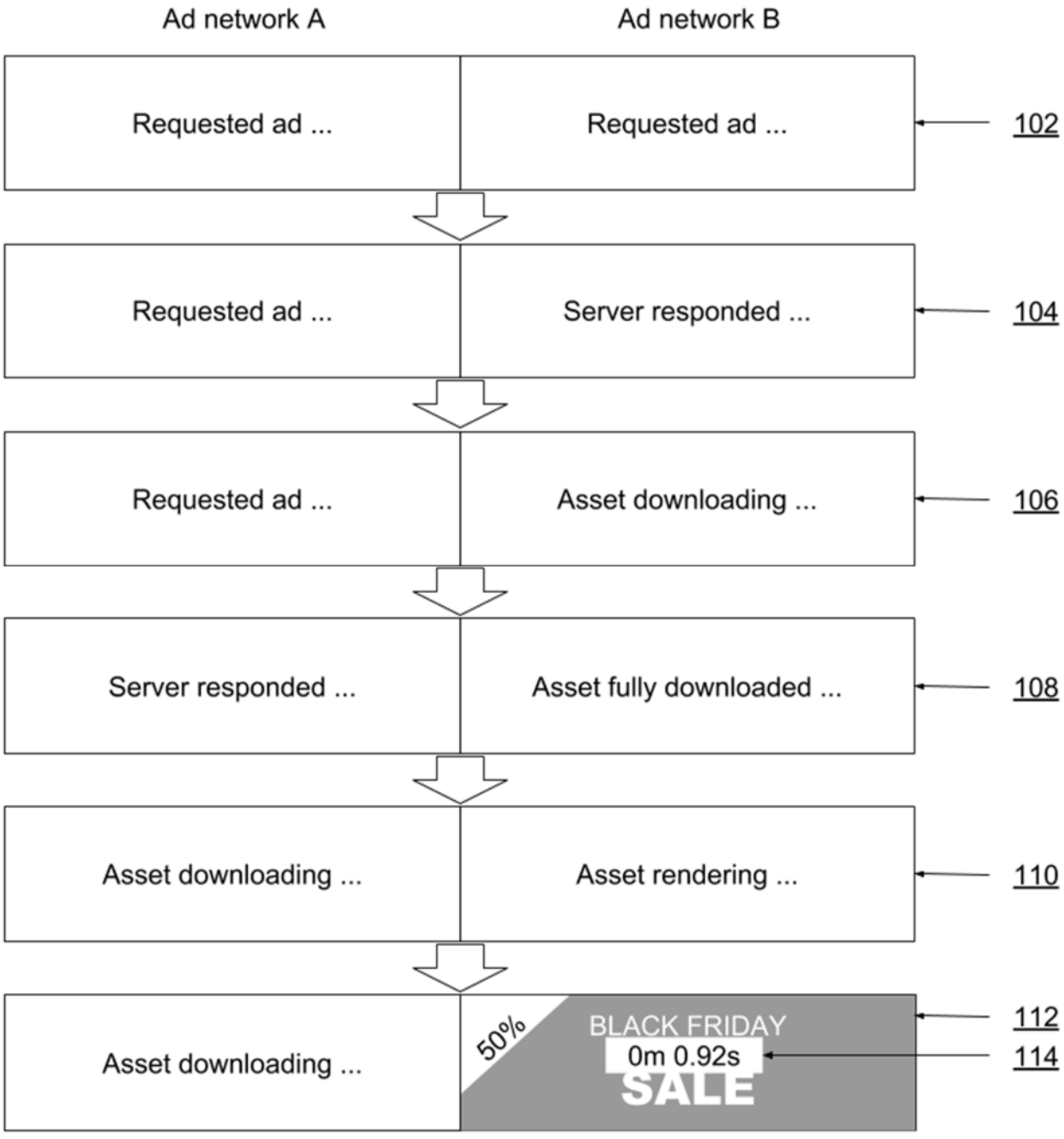

Fig. 1: Side-by-side comparison of ad-network latencies

Fig. 1 illustrates an example of side-by-side comparison of ad-network latencies, per techniques of this disclosure. Multiple networks, e.g., ad-network A and ad-network B, are

simultaneously called to fill in and render parts of a single ad slot (102). For example, the left half of the ad slot renders from ad-network A and the right half of the ad slot renders from ad-network B. The ad slot can be, e.g., a banner slot at the bottom of a page.

Per the techniques, ads from the multiple ad networks are called at exactly the same time, and the developer compares within the app the download speeds by visibly watching the multiple ads as they are called and displayed. A verbose mode enables the display of status messages within the ad slot, e.g., ‘requested ad’, ‘server responded’, ‘no fill’, ‘asset downloading’, ‘asset fully downloaded’, ‘asset rendering’, etc.

Fig. 1, which shows an example evolution in time of the download and rendering of ads from two networks, illustrates how network B outperforms network A by passing through the stages of ‘requested ad’ (102), ‘server responded’ (104), ‘asset downloading’ (106), ‘asset fully downloaded’ (108), and ‘asset rendering’ (112) in the same time as that taken by network A to only proceed from ‘requested ad’ to ‘asset downloading’. The time to finish, from request to full render (114), can also be overlaid on the ad, similar to how a race time finish sticks after an athlete finishes a race.

The controls for the side-by-side comparison tool described herein can be in a software development kit shipped by the ad network to the app developer. Alternately, the controls can be made available through server settings such that a developer can easily see events as they occur in real time. For example, if the developer noticed revenue dropped for a given network, they can turn on the tool to see if the issue is with the ad-serving latency of the network.

Additionally, a developer can run the experiment in the field to get a global picture of ad-serving operations. Often, networks perform properly in one county but not as well in another. In this scenario, the developer can have a status readout in their ad-network account such that they

can see what is happening across the world with users of the app and the different ad networks being used to monetize their ad inventory.

The techniques of this disclosure can be presented as a tool-set provided by an ad network on its ad platform such that app developers and publishers can easily see the latency differences between different ad networks.

CONCLUSION

This disclosure presents real-time tools to compare latencies of one or more ad networks on a side-by-side basis. App developers can use the tools provided herein to compare ad-network latencies and to optimize the selection of ad networks.